(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2024년02월27일
(11) 등록번호 10-2641272
(24) 등록일자 2024년02월22일

(51) 국제특허분류(Int. Cl.)
G06T 19/00 (2011.01) *G06F 3/01* (2006.01)
G06F 3/0346 (2013.01) *G06F 3/147* (2006.01)
G06T 15/00 (2006.01) *G06T 15/04* (2011.01)
G06T 15/20 (2011.01) *G06T 17/20* (2006.01)
H04N 5/14 (2023.01)
(52) CPC특허분류
G06T 19/006 (2013.01)
G06F 3/011 (2022.02)
(21) 출원번호 10-2021-7012646
(22) 출원일자(국제) 2019년09월27일
심사청구일자 2022년09월08일
(85) 번역문제출일자 2021년04월27일
(65) 공개번호 10-2021-0071015
(43) 공개일자 2021년06월15일
(86) 국제출원번호 PCT/US2019/053528
(87) 국제공개번호 WO 2020/076526
국제공개일자 2020년04월16일
(30) 우선권주장
16/155,686 2018년10월09일 미국(US)
(56) 선행기술조사문헌
JP2017004516 A*
US20170213388 A1*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
밸브 코포레이션
미국 워싱턴 (우편번호 98004) 벨뷰 엔이 포쓰 스트리트 10400 스위트 1400
(72) 발명자
블라초스, 알렉스
미국 98004 워싱턴주 벨뷰 노스이스트 4번 스트리트 10400
레이바이, 아론
미국 98004 워싱턴주 벨뷰 노스이스트 4번 스트리트 10400
(74) 대리인
양영준, 김연송, 백만기

전체 청구항 수 : 총 24 항 심사관 : 옥윤철

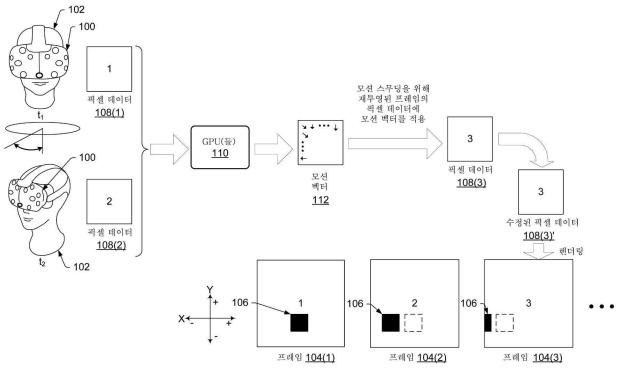
(54) 발명의 명칭 **재투영된 프레임을 위한 모션 스무딩**

(57) 요 약

본원에서는 저더를 완화하는 방식으로 이동하거나 움직이는 객체의 모션을 처리하기 위한 디스플레이 시스템의 모션 스무딩 기술이 설명된다. 예를 들어, 2개의 이전에 렌더링된 프레임과 연관된 제1 픽셀 데이터 및 제2 픽셀 데이터는 입력으로서 그래픽 처리 유닛(GPU)에 제공될 수 있다. GPU의 비디오 인코더는 입력 픽셀 데이터를 처리

(뒷면에 계속)

대 표 도



하여 재투영된 제3 픽셀 데이터를 수정하는 데 사용되는 모션 벡터의 어레이를 생성할 수 있다. 재투영된 프레임에 대한 수정된 제3 픽셀 데이터는 이동하거나 움직이는 객체의 저더를 완화하는 방식으로 헤드 마운티드 디스플레이(HMD)와 같은 디스플레이에 렌더링하기 위해 "모션 스무딩"된다.

(52) CPC특허분류
G06F 3/0346 (2013.01)
G06F 3/147 (2013.01)
G06T 15/005 (2013.01)
G06T 15/04 (2013.01)
G06T 15/205 (2013.01)
G06T 17/20 (2013.01)
H04N 5/145 (2013.01)
G09G 2320/10 (2013.01)
G09G 2354/00 (2013.01)

명 세 서

청구범위

청구항 1

방법으로서,

헤드 마운티드 디스플레이(HMD)의 그래픽 처리 유닛(GPU)에 대한 입력으로서,

애플리케이션으로부터 수신된 제1 프레임과 연관된 제1 픽셀 데이터; 및

상기 애플리케이션으로부터 수신된 제2 프레임과 연관된 제2 픽셀 데이터를 제공하고,

상기 제2 프레임은 가장 최근에 렌더링된 프레임이고, 상기 제1 프레임은 상기 제2 프레임 이전에 렌더링된, 단계;

상기 GPU로부터, 상기 제1 픽셀 데이터와 상기 제2 픽셀 데이터 사이의 비교에 적어도 부분적으로 기초하여 상기 GPU의 비디오 인코더에 의해 생성된 모션 벡터의 어레이를 수신하는 단계;

상기 제2 픽셀 데이터에 적어도 부분적으로 기초하여 재투영된 프레임에 대한 제3 픽셀 데이터를 생성하는 단계;

상기 재투영된 프레임에 대한 렌더 메쉬를 생성하는 단계로서, 상기 렌더 메쉬는 상기 모션 벡터의 어레이의 해상도와 정합하는 해상도에서 복수의 정점을 갖는 테셀레이션형 메쉬를 포함하는, 단계;

상기 모션 벡터의 어레이 중에서 비제로 모션 벡터에 적어도 부분적으로 기초하여, 상기 비제로 모션 벡터의 방향으로 상기 복수의 정점의 정점을 이동된 정점으로서 상기 렌더 메쉬 내의 상이한 위치로 이동시키는 단계;

상기 렌더 메쉬의 이동된 정점에 따라 상기 제3 픽셀 데이터를 수정하여 상기 재투영된 프레임에 대한 수정된 제3 픽셀 데이터를 획득하는 단계; 및

상기 수정된 제3 픽셀 데이터에 적어도 부분적으로 기초하여 상기 재투영된 프레임을 상기 HMD의 디스플레이 패널에 렌더링하는 단계를 포함하는, 방법.

청구항 2

제1항에 있어서, 상기 제1 픽셀 데이터는 상기 제2 픽셀 데이터가 프레임 버퍼로 출력되기 전에 상기 프레임 버퍼로 출력되어 상기 제1 프레임이 상기 제2 프레임 이전에 렌더링되게 하고, 상기 제2 픽셀 데이터는 상기 수정된 제3 픽셀 데이터가 상기 프레임 버퍼로 출력되기 전에 상기 프레임 버퍼로 출력되어 상기 제2 프레임이 상기 재투영된 프레임 이전에 렌더링되게 하는, 방법.

청구항 3

제1항에 있어서, 상기 HMD의 GPU에 대한 입력으로서 상기 제1 픽셀 데이터 및 상기 제2 픽셀 데이터를 제공하는 단계는, 상기 HMD의 GPU에 대한 입력으로서, 상기 디스플레이 패널의 하나 이상의 에지에서 픽셀의 서브셋 이외의 픽셀에 각각 대응하는 상기 제1 픽셀 데이터의 일부 및 상기 제2 픽셀 데이터의 일부를 제공하는 단계를 포함하는, 방법.

청구항 4

제1항에 있어서, 상기 제1 픽셀 데이터의 일부 및 상기 제2 픽셀 데이터의 일부는 각각 상기 디스플레이 패널의 주변에서 렌더링된 패널 마스크에 대응하는 데이터를 나타내고, 상기 방법은,

상기 HMD의 헤드 트랙킹 모듈에 의해 제공된 회전 데이터에 기초하여, 상기 HMD가 상기 제1 프레임의 렌더링과 상기 제2 프레임의 렌더링 사이에서 제1 방향으로부터 제2 방향으로 회전된 것으로 결정하는 단계;

상기 제1 픽셀 데이터를 수정하여 상기 제1 프레임의 장면을 상기 HMD의 제2 방향과 정렬하는 수정된 제1 픽셀 데이터를 획득하는 단계로서, 상기 수정된 제1 픽셀 데이터의 일부는 상기 패널 마스크에 대응하는 데이터를 나타내는, 단계;

상기 제2 픽셀 데이터를 수정하여 상기 제2 프레임의 장면을 상기 HMD의 제1 방향과 정렬하는 수정된 제2 픽셀 데이터를 획득하는 단계로서, 상기 수정된 제2 픽셀 데이터의 일부는 상기 패널 마스크에 대응하는 데이터를 나타내는, 단계; 및

상기 수정된 제1 픽셀 데이터의 부분을 상기 수정된 제2 픽셀 데이터의 부분과 조합하여 상기 패널 마스크에 대응하는 공통 픽셀 값을 갖는 픽셀의 서브셋을 결정하는 단계를 더 포함하고,

상기 HMD의 GPU에 대한 입력으로서 상기 제1 픽셀 데이터 및 상기 제2 픽셀 데이터를 제공하는 단계는, 상기 HMD의 GPU에 대한 입력으로서, 상기 픽셀의 서브셋 이외의 픽셀에 각각 대응하는 상기 제1 픽셀 데이터의 특정 부분 및 상기 제2 픽셀 데이터의 특정 부분을 제공하는 단계를 포함하는, 방법.

청구항 5

방법으로서,

그래픽 처리 유닛(GPU)에 대한 입력으로서,

애플리케이션으로부터 수신된 제1 프레임과 연관된 제1 픽셀 데이터로서, 상기 제1 프레임은 이전에 렌더링된, 제1 픽셀 데이터; 및

상기 애플리케이션으로부터 수신된 제2 프레임과 연관된 제2 픽셀 데이터를 제공하고, 상기 제2 프레임은 상기 제1 프레임 이후에 이전에 렌더링된, 단계;

상기 GPU로부터, 상기 제1 픽셀 데이터 및 상기 제2 픽셀 데이터에 적어도 부분적으로 기초하여 상기 GPU의 비디오 인코더에 의해 생성된 모션 벡터의 어레이를 수신하는 단계;

상기 모션 벡터의 어레이에 오버레이된 감쇠 텍스처를 생성하는 단계;

상기 감쇠 텍스처를 사용하여 상기 모션 벡터의 어레이의 하나 이상의 모션 벡터의 크기를 감소시켜 모션 벡터의 수정된 어레이를 획득하는 단계;

상기 제2 픽셀 데이터에 적어도 부분적으로 기초하여 재투영된 프레임에 대한 제3 픽셀 데이터를 생성하는 단계;

상기 모션 벡터의 수정된 어레이에 적어도 부분적으로 기초하여 상기 제3 픽셀 데이터를 수정해서 상기 재투영된 프레임에 대한 수정된 제3 픽셀 데이터를 획득하는 단계; 및

상기 수정된 제3 픽셀 데이터에 적어도 부분적으로 기초하여 상기 재투영된 프레임을 디스플레이에 렌더링하는 단계를 포함하는, 방법.

청구항 6

제5항에 있어서, 상기 제1 픽셀 데이터는 상기 제2 픽셀 데이터가 프레임 버퍼로 출력되기 전에 상기 프레임 버퍼로 출력되어 상기 제1 프레임이 상기 제2 프레임 이전에 렌더링되게 하고, 상기 제2 픽셀 데이터는 상기 수정된 제3 픽셀 데이터가 상기 프레임 버퍼로 출력되기 전에 상기 프레임 버퍼로 출력되어 상기 제2 프레임이 상기 재투영된 프레임 이전에 렌더링되게 하는, 방법.

청구항 7

제5항에 있어서,

상기 재투영된 프레임에 대한 렌더 메쉬를 생성하는 단계로서, 상기 렌더 메쉬는 복수의 정점을 갖는 테셀레이션형 메쉬를 포함하는, 단계; 및

상기 모션 벡터의 수정된 어레이 중에서 비제로 모션 벡터에 적어도 부분적으로 기초하여, 상기 복수의 정점의 정점을 이동된 정점으로서 상기 렌더 메쉬 내의 상이한 위치로 이동시키는 단계를 더 포함하고,

상기 모션 벡터의 수정된 어레이에 적어도 부분적으로 기초하여 상기 제3 픽셀 데이터를 수정하는 단계는 상기 렌더 메쉬의 이동된 정점에 따라 상기 제3 픽셀 데이터의 픽셀 값을 이동시켜 상기 재투영된 프레임에 대한 수정된 제3 픽셀 데이터를 획득하는 단계를 포함하는, 방법.

청구항 8

제5항에 있어서, 상기 디스플레이는 헤드 마운티드 디스플레이(HMD)이고, 상기 방법은,

상기 HMD의 헤드 트랙킹 모듈에 의해 제공된 회전 데이터에 기초하여, 상기 HMD가 상기 제1 프레임의 렌더링과 상기 제2 프레임의 렌더링 사이에서 제1 방향으로부터 제2 방향으로 회전된 것으로 결정하는 단계; 및

다음 중 적어도 하나를 수정하는 단계를 더 포함하고,

상기 제1 프레임의 장면을 상기 HMD의 제2 방향과 정렬하는 수정된 제1 픽셀 데이터를 획득하기 위한 상기 제1 픽셀 데이터; 또는

상기 제2 프레임의 장면을 상기 HMD의 제1 방향과 정렬하는 수정된 제2 픽셀 데이터를 획득하기 위한 상기 제2 픽셀 데이터;

상기 HMD의 GPU에 대한 입력으로서 상기 제1 픽셀 데이터 및 상기 제2 픽셀 데이터를 제공하는 단계는, 상기 HMD의 GPU에 대한 입력으로서, 상기 수정된 제1 픽셀 데이터 또는 상기 수정된 제2 픽셀 데이터 중 적어도 하나를 제공하는 단계를 포함하는, 방법.

청구항 9

제5항에 있어서, 상기 GPU에 대한 입력으로서 상기 제1 픽셀 데이터 및 상기 제2 픽셀 데이터를 제공하는 단계는, 상기 GPU에 대한 입력으로서, 상기 디스플레이의 하나 이상의 에지에서 픽셀의 서브셋 이외의 픽셀에 각각 대응하는 상기 제1 픽셀 데이터의 일부 및 상기 제2 픽셀 데이터의 일부를 제공하는 단계를 포함하는, 방법.

청구항 10

제5항에 있어서, 상기 디스플레이는 헤드 마운티드 디스플레이(HMD)이고, 상기 제1 픽셀 데이터의 일부 및 상기 제2 픽셀 데이터의 일부는 각각 상기 HMD의 디스플레이 패널의 주변에서 렌더링된 패널 마스크에 대응하는 데이터를 나타내고, 상기 방법은,

상기 HMD의 헤드 트랙킹 모듈에 의해 제공된 회전 데이터에 기초하여, 상기 HMD가 상기 제1 프레임의 렌더링과 상기 제2 프레임의 렌더링 사이에서 제1 방향으로부터 제2 방향으로 회전된 것으로 결정하는 단계;

상기 제1 픽셀 데이터를 수정하여 상기 제1 프레임의 장면을 상기 HMD의 제2 방향과 정렬하는 수정된 제1 픽셀 데이터를 획득하는 단계로서, 상기 수정된 제1 픽셀 데이터의 일부는 상기 패널 마스크에 대응하는 데이터를 나타내는, 단계;

상기 제2 픽셀 데이터를 수정하여 상기 제2 프레임의 장면을 상기 HMD의 제1 방향과 정렬하는 수정된 제2 픽셀 데이터를 획득하는 단계로서, 상기 수정된 제2 픽셀 데이터의 일부는 상기 패널 마스크에 대응하는 데이터를 나타내는, 단계; 및

상기 수정된 제1 픽셀 데이터의 부분을 상기 수정된 제2 픽셀 데이터의 부분과 조합하여 상기 패널 마스크에 대응하는 공통 픽셀 값을 갖는 픽셀의 서브셋을 결정하는 단계를 더 포함하고,

상기 HMD의 GPU에 대한 입력으로서 상기 제1 픽셀 데이터 및 상기 제2 픽셀 데이터를 제공하는 단계는, 상기 HMD의 GPU에 대한 입력으로서, 상기 픽셀의 서브셋 이외의 픽셀에 각각 대응하는 상기 제1 픽셀 데이터의 특정 부분 및 상기 제2 픽셀 데이터의 특정 부분을 제공하는 단계를 포함하는, 방법.

청구항 11

제5항에 있어서,

상기 모션 벡터의 어레이의 크기를 임계 크기와 비교하는 단계;

상기 모션 벡터의 어레이의 모션 벡터의 서브셋이 상기 임계 크기보다 큰 크기를 갖는 것으로 결정하는 단계; 및

상기 모션 벡터의 서브셋의 크기를 감소시켜 상기 모션 벡터의 수정된 어레이를 획득하는 단계를 더 포함하는, 방법.

청구항 12

제5항에 있어서, 상기 제1 프레임 및 상기 제2 프레임은 일련의 프레임에서 연속적으로 렌더링되는, 방법.

청구항 13

제5항에 있어서, 상기 제1 픽셀 데이터는 제1 크로마 데이터를 포함하고 상기 제2 픽셀 데이터는 제2 크로마 데이터를 포함하고, 상기 모션 벡터의 어레이는 상기 제1 크로마 데이터 및 상기 제2 크로마 데이터에 적어도 부분적으로 기초하여 상기 GPU의 비디오 인코더에 의해 생성된 모션 벡터의 제1 어레이이고, 상기 방법은,

상기 GPU에 대한 제2 입력으로서,

상기 제1 프레임과 연관된 제1 루마 데이터; 및

상기 제2 프레임과 연관된 제2 루마 데이터를 제공하는 단계;

상기 GPU로부터, 상기 제1 루마 데이터 및 상기 제2 루마 데이터에 적어도 부분적으로 기초하여 상기 GPU의 비디오 인코더에 의해 생성된 모션 벡터의 제2 어레이를 수신하는 단계; 및

상기 모션 벡터의 제1 어레이 또는 상기 모션 벡터의 제2 어레이 중 하나를 상기 모션 벡터의 제1 어레이 및 상기 모션 벡터의 제2 어레이의 분석에 기초하여 모션 벡터의 선택된 어레이로서 선택하는 단계를 더 포함하고,

상기 제3 픽셀 데이터를 수정하는 데 사용되는 상기 모션 벡터의 수정된 어레이는 상기 모션 벡터의 선택된 어레이에 기초하는, 방법.

청구항 14

제5항에 있어서, 상기 모션 벡터의 어레이는 모션 벡터의 제1 어레이이고, 상기 방법은,

상기 제1 프레임을 제1 회전량만큼 회전시켜 제1 회전된 프레임을 획득하는 단계;

상기 제2 프레임을 제2 회전량만큼 회전시켜 제2 회전된 프레임을 획득하는 단계;

상기 GPU에 대한 제2 입력으로서,

상기 제1 회전된 프레임과 연관된 제3 픽셀 데이터; 및

상기 제2 회전된 프레임과 연관된 제4 픽셀 데이터를 제공하는 단계;

상기 GPU로부터, 상기 제3 픽셀 데이터 및 상기 제4 픽셀 데이터에 적어도 부분적으로 기초하여 모션 벡터의 제2 어레이를 수신하는 단계;

상기 모션 벡터의 제1 어레이 및 상기 모션 벡터의 제2 어레이에서 모션 벡터의 방향 또는 크기 중 적어도 하나의 차이를 결정하는 단계; 및

상기 차이에 적어도 부분적으로 기초하여 모션 벡터의 최종 어레이를 생성하는 단계를 더 포함하고,

상기 제3 픽셀 데이터를 수정하는 데 사용되는 상기 모션 벡터의 수정된 어레이는 상기 모션 벡터의 최종 어레이에 기초하는, 방법.

청구항 15

제5항에 있어서, 상기 모션 벡터의 어레이는 제1 해상도에서 모션 벡터의 제1 어레이이고, 상기 방법은,

상기 모션 벡터의 제1 어레이에 적어도 부분적으로 기초하여 상기 디스플레이에 걸쳐 있는 이미지 영역의 제1 부분에서 모션의 부재를 검출하는 단계;

상기 모션 벡터의 제1 어레이에 적어도 부분적으로 기초하여 상기 이미지 영역의 제2 부분에서 객체의 모션을 검출하는 단계;

상기 GPU에 대한 추가 입력으로서,

상기 이미지 영역의 제1 부분에 대응하는 상기 제1 픽셀 데이터의 제1 부분;

상기 이미지 영역의 제1 부분에 대응하는 상기 제2 픽셀 데이터의 제1 부분;

상기 이미지 영역의 제2 부분에 대응하는 상기 제1 픽셀 데이터의 제2 부분; 및

상기 이미지 영역의 제2 부분에 대응하는 상기 제2 픽셀 데이터의 제2 부분을 제공하는 단계;

상기 GPU로부터, 상기 제1 픽셀 데이터의 제1 부분 및 상기 제2 픽셀 데이터의 제1 부분에 적어도 부분적으로 기초하여 상기 GPU의 비디오 인코더에 의해 생성된 모션 벡터의 제2 어레이를 수신하는 단계로서, 상기 모션 벡터의 제2 어레이는 상기 제1 해상도보다 높은 해상도인 제2 해상도에서 생성되는, 단계; 및

상기 GPU로부터, 상기 제1 픽셀 데이터의 제2 부분 및 상기 제2 픽셀 데이터의 제2 부분에 적어도 부분적으로 기초하여 상기 GPU의 비디오 인코더에 의해 생성된 모션 벡터의 제3 어레이를 수신하는 단계를 더 포함하고, 상기 모션 벡터의 제3 어레이는 상기 제2 해상도보다 낮은 해상도인 제3 해상도에서 생성되고,

상기 제3 픽셀 데이터를 수정하는 데 사용되는 상기 모션 벡터의 수정된 어레이는 상기 모션 벡터의 제2 어레이 및 상기 모션 벡터의 제3 어레이에 기초하는, 방법.

청구항 16

제5항에 있어서, 상기 제3 픽셀 데이터를 수정하는 단계는 깊이 버퍼를 이용하여 상기 수정된 제3 픽셀 데이터의 프레임 버퍼에 대한 픽셀 값의 최종 세트를 결정하는 단계를 더 포함하는, 방법.

청구항 17

시스템으로서,

디스플레이;

그래픽 처리 유닛(GPU); 및

로직을 포함하고, 상기 로직은,

상기 GPU에 대한 입력으로서,

애플리케이션으로부터 수신된 제1 프레임과 연관된 제1 픽셀 데이터로서, 상기 제1 프레임은 이전에 렌더링된, 제1 픽셀 데이터; 및

상기 애플리케이션으로부터 수신된 제2 프레임과 연관된 제2 픽셀 데이터를 제공하도록 구성되고, 상기 제2 프레임은 상기 제1 프레임 이후에 이전에 렌더링되고;

상기 GPU로부터, 상기 제1 픽셀 데이터 및 상기 제2 픽셀 데이터에 적어도 부분적으로 기초하여 상기 GPU의 비디오 인코더에 의해 생성된 모션 벡터의 어레이를 수신하도록 구성되고;

상기 제2 픽셀 데이터에 적어도 부분적으로 기초하여, 재투영된 프레임에 대한 제3 픽셀 데이터를 생성하도록 구성되고;

상기 재투영된 프레임에 대한 렌더 메쉬로서, 상기 모션 벡터의 어레이의 해상도와 정합하는 해상도에서 복수의 정점을 갖는 렌더 메쉬를 생성하도록 구성되고;

상기 모션 벡터의 어레이 중에서 비제로 모션 벡터에 적어도 부분적으로 기초하여, 상기 복수의 정점의 정점을 이동된 정점으로서 상기 렌더 메쉬 내의 상이한 위치로 이동시키도록 구성되고;

상기 렌더 메쉬의 이동된 정점에 적어도 부분적으로 기초하여 상기 제3 픽셀 데이터를 수정해서 상기 재투영된 프레임에 대한 수정된 제3 픽셀 데이터를 획득하도록 구성되고;

상기 수정된 제3 픽셀 데이터에 적어도 부분적으로 기초하여 상기 재투영된 프레임을 상기 디스플레이에 렌더링하도록 구성되는, 시스템.

청구항 18

제17항에 있어서, 상기 로직은,

상기 제2 픽셀 데이터가 프레임 버퍼로 출력되기 전에 상기 제1 픽셀 데이터를 상기 프레임 버퍼로 출력하여 상기 제1 프레임이 상기 제2 프레임 이전에 렌더링되게 하도록 더 구성되고;

상기 수정된 제3 픽셀 데이터가 상기 프레임 버퍼로 출력되기 전에 상기 제2 픽셀 데이터를 상기 프레임 버퍼로

출력하여 상기 제2 프레임이 상기 렌더링된 프레임 이전에 렌더링되게 하도록 더 구성되는, 시스템.

청구항 19

제17항에 있어서,

상기 렌더 메쉬는 상기 정점을 포함하는 상기 복수의 정점을 갖는 테셀레이션형 메쉬를 포함하고;

상기 정점은 (i) 상기 비제로 모션 벡터의 방향으로 및 (ii) 상기 비제로 모션 벡터의 크기에 대응하는 양만큼 이동되고;

상기 렌더 메쉬의 이동된 정점에 적어도 부분적으로 기초하여 상기 제3 픽셀 데이터를 수정하는 단계는 상기 렌더 메쉬의 이동된 정점에 따라 상기 제3 픽셀 데이터의 픽셀 값을 이동시켜 상기 재투영된 프레임에 대한 수정된 제3 픽셀 데이터를 획득하는 단계를 포함하는, 시스템.

청구항 20

제17항에 있어서, 상기 디스플레이는 헤드 마운티드 디스플레이(HMD)이고, 상기 로직은,

상기 HMD의 헤드 트랙킹 모듈에 의해 제공된 회전 데이터에 기초하여, 상기 HMD가 상기 제1 프레임의 렌더링과 상기 제2 프레임의 렌더링 사이에서 제1 방향으로부터 제2 방향으로 회전된 것으로 결정하도록 더 구성되고;

다음 중 적어도 하나를 수정하도록 더 구성되고,

상기 제1 프레임의 장면을 상기 HMD의 제2 방향과 정렬하는 수정된 제1 픽셀 데이터를 획득하기 위한 상기 제1 픽셀 데이터; 또는

상기 제2 프레임의 장면을 상기 HMD의 제1 방향과 정렬하는 수정된 제2 픽셀 데이터를 획득하기 위한 상기 제2 픽셀 데이터;

상기 HMD의 GPU에 대한 입력으로서 상기 제1 픽셀 데이터 및 상기 제2 픽셀 데이터를 제공하는 단계는, 상기 HMD의 GPU에 대한 입력으로서, 상기 수정된 제1 픽셀 데이터 또는 상기 수정된 제2 픽셀 데이터 중 적어도 하나를 제공하는 단계를 포함하는, 시스템.

청구항 21

제17항에 있어서, 상기 로직은,

상기 제1 픽셀 데이터와 상기 제2 픽셀 데이터 사이의 컬러 픽셀 값을 비교하여 상기 모션 벡터의 어레이에서 개별 모션 벡터에 대응하는 개별 색상 차이 값을 결정하도록 더 구성되고;

임계 색상 차이보다 작은 색상 차이 값과 연관된 모션 벡터의 서브셋을 결정하도록 더 구성되고;

상기 모션 벡터의 서브셋의 크기를 제로로 감소시켜 모션 벡터의 수정된 어레이를 획득하도록 더 구성되고,

상기 비제로 모션 벡터에 적어도 부분적으로 기초하여 상기 렌더 메쉬의 정점을 이동시키는 단계는 상기 모션 벡터의 수정된 어레이 중에서 비제로 모션 벡터에 기초하여 상기 제3 픽셀 데이터를 수정하는 단계를 포함하는, 시스템.

청구항 22

제17항에 있어서, 상기 제1 픽셀 데이터는 제1 크로마 데이터를 포함하고 상기 제2 픽셀 데이터는 제2 크로마 데이터를 포함하고, 상기 모션 벡터의 어레이는 상기 제1 크로마 데이터 및 상기 제2 크로마 데이터에 적어도 부분적으로 기초하여 상기 GPU의 비디오 인코더에 의해 생성된 모션 벡터의 제1 어레이이고, 상기 로직은,

상기 GPU에 대한 제2 입력으로서,

상기 제1 프레임과 연관된 제1 루마 데이터; 및

상기 제2 프레임과 연관된 제2 루마 데이터를 제공하도록 더 구성되고;

상기 GPU로부터, 상기 제1 루마 데이터 및 상기 제2 루마 데이터에 적어도 부분적으로 기초하여 상기 GPU의 비디오 인코더에 의해 생성된 모션 벡터의 제2 어레이를 수신하도록 더 구성되고;

상기 모션 벡터의 제1 어레이 또는 상기 모션 벡터의 제2 어레이 중 하나를 상기 모션 벡터의 제1 어레이 및 상기 모션 벡터의 제2 어레이의 분석에 기초하여 모션 벡터의 선택된 어레이로서 선택하도록 더 구성되고,

상기 렌더 메쉬의 정점을 이동시키는 데 사용되는 상기 비제로 모션 벡터는 상기 모션 벡터의 선택된 어레이 중에서 비제로 모션 벡터를 포함하는, 시스템.

청구항 23

제17항에 있어서, 상기 로직은,

모션 벡터의 제1 수정된 어레이를 획득하기 위해 상기 모션 벡터의 어레이에 스칼라 중간값 필터; 또는

상기 모션 벡터의 수정된 어레이를 획득하기 위해 상기 모션 벡터의 어레이에 평균-최대 블러 필터 중 적어도 하나를 적용하도록 더 구성되고,

상기 비제로 모션 벡터에 적어도 부분적으로 기초하여 상기 렌더 메쉬의 정점을 이동시키는 단계는 상기 모션 벡터의 수정된 어레이 중에서 비제로 모션 벡터에 기초하여 상기 제3 픽셀 데이터를 수정하는 단계를 포함하는, 시스템.

청구항 24

제5항에 있어서,

상기 제1 픽셀 데이터와 상기 제2 픽셀 데이터 사이의 컬러 픽셀 값을 비교하여 상기 모션 벡터의 어레이에서 개별 모션 벡터에 대응하는 개별 색상 차이 값을 결정하는 단계;

임계 색상 차이보다 작은 색상 차이 값과 연관된 모션 벡터의 서브셋을 결정하는 단계; 및

상기 모션 벡터의 서브셋의 크기를 제로로 감소시켜 상기 모션 벡터의 수정된 어레이를 획득하는 단계를 더 포함하는, 방법.

발명의 설명

기 술 분 야

[0001] 관련 출원의 교차 참조

[0002] 본 출원은 2018년 10월 9일자에 "MOTION SMOOTHING FOR RE-PROJECTED FRAMES"이라는 명칭으로 출원된 미국 특허 출원 제16/155,686호에 대한 우선권을 주장하는 PCT 출원으로서, 이는 그 전체가 본원에 참조로서 원용된다.

배 경 기 술

[0003] 가상 현실(VR) 시스템은 비디오 게임 산업의 내외부 모두에서 사용된다. VR 헤드셋에 내장된 것과 같은 VR 시스템용 디스플레이는 통상적으로 VR 애플리케이션에 적합한 최소 재생률로 작동한다. 예를 들어, 90헤르츠(Hz)는 VR 디스플레이의 일반적인 재생률이다. "라이브 렌더링" 시나리오에서, 비디오 게임과 같은 그래픽 기반 애플리케이션은 디스플레이의 재생률과 정합하는 프레임률로 렌더링하기 위한 프레임을 출력하며, 이는 애플리케이션으로부터 수신된 새 프레임(본원에서는 "실제 프레임"으로 지칭됨)이 화면을 재생할 때마다 표시된다는 것을 의미한다. 이러한 라이브 렌더링 시나리오는 종종 애플리케이션이 "프레임률에 도달하는" 것으로 지칭된다.

[0004] 실제로, 애플리케이션은 다양한 이유로 인해 항상 프레임률에 도달하는 것은 아니다. 예를 들어, 애플리케이션은 간헐적으로 프레임을 낮출 수 있고, 및/또는 애플리케이션은 일시적으로 더 느린 비율(예를 들어, 이상적인 프레임률이 초당 90프레임인 경우 초당 45프레임)로 프레임을 출력할 수 있다. 애플리케이션이 프레임률에 도달하지 않는 상황에서, "회전 전용 재투영"이라는 기술을 사용하여 누락된 프레임을 사용자의 머리 회전을 처리하는 방식으로 재투영된 프레임으로 대체할 수 있어, 마치 애플리케이션이 프레임률에 도달하는 것처럼 사용자에게 나타나게 할 수 있다. 예를 들어, 재투영 없이, 애플리케이션으로부터의 프레임률이 불충분하면 게임 내 스터터링 또는 히칭이 발생할 수 있다. 사용자가 가상 환경에 완전히 몰입되는 VR 애플리케이션에서, 애플리케이션이 프레임률에 도달하지 못하고 누락된 프레임을 보상하기 위한 재투영이 없으면 사용자는 메스꺼워질 수 있다. 따라서, 재투영은 애플리케이션이 프레임률에 도달하지 않을 때 더 나은 사용자 경험을 제공하는 기술이다. 애플리케이션이 이상적인 프레임률의 절반(예를 들어, 초당 90프레임이 이상적인 프레임률인 경우 초당 45프레

임)으로 프레임을 출력하는 예를 고려한다. 이러한 예에서, 다른 모든 프레임은 가장 최근에 렌더링된 실제 프레임으로부터 픽셀 데이터를 사용하여 재투영되어 사용자의 현재 머리 방향으로 재투영된 장면과 정합하도록 (예를 들어, 회전 및 재투영 계산을 통해) 장면을 변환하는 재투영된 프레임을 생성할 수 있다. 이를 통해, 재투영된 프레임이 프레임률에 도달하지 못한 애플리케이션을 보상하는 데 사용되는 경우에도, 사용자의 머리 회전에 대해 예상되는 방식으로 장면이 움직이는 것처럼 사용자에게 보인다.

[0005] 회전 전용 재투영은 게임 내 스터터링 또는 히칭을 방지하지만, 적어도 낮은 지속성 디스플레이를 사용하는 VR 시스템(예를 들어, 디스플레이가 프레임 시간의 극히 일부 동안 조명되는 경우)에서 머리의 회전 동안 그 자체의 원치 않는 시각적 아티팩트를 생성한다. 예를 들어, 회전 전용 재투영은 머리 회전을 처리하지만, 장면에서 프레임 사이에서 이동하거나 움직이는 가상 객체를 처리하지 않는다. 이로 인해 "저더"라는 원치 않는 시각적 아티팩트가 이동하거나 움직이는 객체와 관련하여 발생할 수 있다. 저더로 인해 사용자는 이동체(예를 들어, 화면을 가로지르는 총알이나 볼)가 두 위치 사이에서 프레임 간 튀는(또는 그 자체로부터 분리되는) 것처럼 보이는 경우에 "이중 고스팅 효과"를 인지한다. 따라서, 재투영이 사용되는 동안 사용자가 머리를 회전시키면, 장면에서 이동하거나 움직이는 객체가 흔들릴 것이다.

[0006] 본원에는 이러한 시스템 및 다른 시스템을 개선하고 향상시키기 위한 기술적 해결책이 제공된다.

도면의 간단한 설명

[0007] 상세한 설명을 첨부 도면을 참조하여 설명한다. 도면에서, 참조 부호의 맨 왼쪽 숫자(들)는 참조 부호가 처음에 나타나는 도면을 식별한다. 서로 다른 도면에서 동일한 참조 부호의 사용은 유사하거나 동일한 구성요소 또는 특징을 나타낸다.

도 1은 헤드 마운티드 디스플레이(HMD)와 같은 디스플레이에 프레임을 렌더링하기 위해 재투영이 사용될 때 장면에서 이동하거나 움직이는 객체를 처리하기 위한 예시적인 모션 스무딩 기술을 나타낸 도면이다.

도 2는 예시적인 렌더 메쉬 및 재투영된 프레임을 위한 모션 스무딩에 렌더 메쉬가 사용될 수 있는 방법을 나타낸 도면이다.

도 3은 본원에 개시된 실시예에 따른 모션 스무딩 기술의 일부로서 그래픽 처리 유닛(GPU)에 의해 생성된 모션 벡터를 사용하여 재투영된 프레임을 생성하기 위한 예시적인 프로세스의 흐름도이다.

도 4는 본원에 개시된 실시예에 따른 픽셀 데이터가 모션 벡터 추정을 위해 GPU에 입력되기 전에 이전에 렌더링된 프레임의 픽셀 데이터를 정렬하기 위한 예시적인 프로세스의 흐름도이다.

도 5는 본원에 개시된 실시예에 따른 이전에 렌더링된 프레임의 일부 픽셀 데이터를 배제하고 픽셀 데이터의 잔여 부분을 모션 벡터 추정을 위해 GPU에 입력으로서 제공하기 위한 예시적인 프로세스의 흐름도이다.

도 6은 본원에 개시된 실시예에 따른 이전에 렌더링된 프레임의 픽셀 데이터를 정렬하고, 이전에 렌더링된 프레임의 픽셀 데이터의 패널 마스크부를 배제하고, 픽셀 데이터의 잔여 부분을 모션 벡터 추정을 위해 GPU에 입력으로서 제공하기 위한 예시적인 프로세스의 흐름도이다.

도 7은 본원에 개시된 실시예에 따른 모션 스무딩 기술의 일부로서 재투영된 프레임을 생성하기 위해 사용되기 전에 GPU로부터 출력되는 모션 벡터를 임계화하기 위한 예시적인 프로세스의 흐름도이다.

도 8은 본원에 개시된 실시예에 따른 모션 스무딩 기술의 일부로서 재투영된 프레임을 생성하기 위해 사용되기 전에 GPU로부터 출력되는 모션 벡터를 감쇠하기 위한 예시적인 프로세스의 흐름도이다.

도 9는 본원에 개시된 실시예에 따른 모션 스무딩 기술의 일부로서 재투영된 프레임을 생성하기 위해 사용되기 전에 GPU로부터 출력되는 모션 벡터를 감쇠하는 데 사용될 수 있는 감쇠 텍스처를 생성하기 위한 예시적인 프로세스의 흐름도이다.

도 10은 본원에 개시된 실시예에 따른 모션 스무딩 기술의 일부로서 재투영된 프레임을 생성하기 위해 모션 벡터의 결과 세트가 사용되기 전에 색상 변화가 거의 또는 전혀 없는 영역에 대응하는 모션 벡터를 제로화하기 위한 예시적인 프로세스의 흐름도이다.

도 11은 본원에 개시된 실시예에 따른 모션 스무딩 기술의 일부로서 재투영된 프레임을 생성하기 위해 모션 벡터의 결과 세트가 사용되기 전에 하나 이상의 필터를 사용하여 모션 벡터 필드를 "클린업"하기 위한 예시적인 프로세스(1100)의 흐름도이다.

도 12는 본원에 개시된 실시예에 따른 픽셀 데이터가 모션 벡터 추정을 위해 GPU에 입력되기 전에 이전에 렌더링된 프레임을 회전시키기 위한 예시적인 프로세스의 흐름도이다.

도 13은 본원에 개시된 실시예에 따른 이전에 렌더링된 프레임의 크로마 데이터 및 루마 데이터에 기초하여 생성되는 모션 벡터의 어레이 사이에서 선택하기 위한 예시적인 프로세스의 흐름도이다.

도 14는 본원에 개시된 실시예에 따른 모션 벡터의 다수 어레이를 획득하고, 어레이 간의 차이를 결정하고, 모션 스무딩을 위해 결정된 차이에 기초하여 모션 벡터의 최종 어레이를 생성하기 위한 예시적인 프로세스의 흐름도이다.

도 15는 본원에 개시된 실시예에 따른 이미지 영역의 상이한 부분에 대해 상이한 해상도에서 모션 벡터의 다수 어레이를 획득하기 위한 예시적인 프로세스의 흐름도이다.

도 16은 본원에 개시된 기술이 구현될 수 있는 VR 헤드셋과 같은 웨어러블 장치의 예시적인 구성요소를 도시하고 있다.

발명을 실시하기 위한 구체적인 내용

[0008] 본원에서는 특히 헤드 마운티드 디스플레이(HMD)와 같은 디스플레이에 프레임을 렌더링하기 위해 재투영이 사용될 때 장면에서 이동하거나 움직이는 객체를 처리하기 위한 예시적인 모션 스무딩 기술을 설명한다. 본원에 설명된 모션 스무딩 기술은, 이동하거나 움직이는 객체를 위한 전술한 저더 아티팩트와 같은, 이동하거나 움직이는 객체에 대한 원치 않는 시각적 아티팩트를 완화한다. HMD는 디스플레이에 이미지를 렌더링할 때 개시된 모션 스무딩 기술을 구현할 수 있는 예시적인 유형의 디스플레이이다. HMD는 사용자를 가상 현실(VR) 환경 또는 증강 현실(AR) 환경에 몰입시키기 위해 사용자에 의해 착용될 수 있다. HMD의 하나 이상의 디스플레이 패널은 애플리케이션(예를 들어, 비디오 게임)에 의해 출력되는 프레임에 기초하여 이미지를 렌더링하며, 이러한 이미지는 HMD에 포함된 광학 장치를 통해 사용자가 볼 수 있어, 마치 사용자가 VR 또는 AR 환경에 몰입된 것처럼 사용자가 이미지를 인지하게 한다.

[0009] 전술한 바와 같이, HMD는 프레임률에 도달하지 못하는 애플리케이션을 보상하기 위해 "재투영"이라는 기술을 이용할 수 있다. 예를 들어, 재투영된 프레임은 이상적인 프레임률을 달성하기 위해 실제 프레임 사이에서 렌더링될 수 있으며, 각 재투영된 프레임은 애플리케이션으로부터 수신된 최근 렌더링된 실제 프레임(예를 들어, 가장 최근에 렌더링된 실제 프레임)으로부터 픽셀 데이터를 사용하여 생성될 수 있다. 재투영된 프레임에서, 이전의 실제 프레임에서 렌더링된 장면은 사용자의 머리 회전을 처리하는 방식으로 (예를 들어, 회전 및 재투영 계산을 통해) 변환된다.

[0010] 본원에서는 프레임 간에 이동하거나 움직이는 객체의 모션을 추가로 처리하는 재투영된 프레임을 생성하는 데 사용되는 모션 스무딩 기술을 설명한다. 본원에 설명된 모션 스무딩 기술은 모션 벡터의 형태로 다수 프레임에 걸쳐 객체의 모션(예를 들어, 방향 및 크기)을 추정하기 위해 컴퓨터 비전 알고리즘을 사용한다. 모션 벡터는 그래픽 처리 유닛(GPU)의 비디오 인코더(예를 들어, 비디오 인코드 칩)를 사용하여 획득될 수 있다. 예를 들어, GPU의 비디오 인코더는 다수의 이전에 렌더링된 프레임의 픽셀 데이터를 분석하여 모션 벡터의 어레이를 생성할 수 있고, 이러한 모션 벡터는 이동하거나 움직이는 객체를 처리하는 방식으로 재투영된 프레임의 픽셀 데이터를 수정하는 데 사용할 수 있다. 달리 말하면, GPU로부터 수신된 모션 벡터는 이동체의 저더가 완화되도록 렌더링될 재투영된 프레임에 객체가 위치되어야 하는 장소를 (이전에 렌더링된 프레임에서 객체의 모션으로부터) 외삽할 수 있다.

[0011] 예시적인 모션 스무딩 프로세스에서, 제1 프레임 및 제2 프레임과 각각 연관된 제1 픽셀 데이터 및 제2 픽셀 데이터는 그래픽 처리 유닛(GPU)에 입력으로서 제공될 수 있다. 제1 프레임 및 제2 프레임은 애플리케이션으로부터 수신되어 이전에 사전 렌더링될 수 있다(예를 들어, 애플리케이션으로부터 가장 최근에 렌더링된 2개의 실제 프레임). GPU의 비디오 인코더는 입력 픽셀 데이터를 처리하여 모션 벡터의 어레이를 생성할 수 있다. 제2 프레임과 연관된 제2 픽셀 데이터에 적어도 부분적으로 기초하여 재투영된 프레임(향후 렌더링될 프레임)에 대해 제3 픽셀 데이터가 생성될 수 있으며, 이러한 제3 픽셀 데이터는 GPU로부터 수신된 모션 벡터의 어레이에 적어도 부분적으로 기초하여 추가로 수정될 수 있다. 재투영된 프레임에 대한 이러한 수정된 제3 픽셀 데이터는 장면에서 이동하거나 움직이는 객체에 대한 임의의 저더를 제거하지 않더라도 완화하기 위해 "모션 스무딩"된다. 그 후, 모션 스무딩된 재투영된 프레임은 수정된 제3 픽셀 데이터에 적어도 부분적으로 기초하여 디스플레이에 렌더링되어, 애플리케이션이 프레임률에 도달하지 못하는 경우에도 객체가 예상되는 방식으로 장면 내에서 이동하거나 움직이는 경우에 보다 사실적인 관측 환경을 제공할 수 있다. 따라서, GPU로부터 수신된 모션 벡터의 어레

이는 향후 프레임(재투영된 프레임)으로 외삽하기 위해 사용될 수 있는 데, 이는 수정된 제3 픽셀 데이터가 제1 픽셀 데이터 및 제2 픽셀 데이터를 프레임 버퍼로 출력한 후에 프레임 버퍼로 출력되기 때문이다. 이러한 방식으로, 제1 프레임은 제2 프레임 이전에 렌더링되고 제2 프레임은 재투영된 프레임 이전에 렌더링된다. 이러한 외삽 기술은 프레임 간 보간과 대조될 수 있고, 본원에 설명된 기술 및 시스템이, 프레임 간 보간을 위해 모션 벡터를 사용하는 것과 대조적으로, 향후 프레임으로 외삽하기 위해 GPU로부터 수신된 모션 벡터를 사용하는 것과 관련됨을 이해해야 한다.

[0012] 또한, 본원에서는 본원에 개시된 기술 및 프로세스를 구현하도록 구성된 디스플레이 시스템(예를 들어, HMD)을 포함하는 시스템과 같은 시스템뿐만 아니라, 본원에 개시된 기술 및 프로세스를 구현하기 위한 컴퓨터 실행 가능 명령을 저장하는 비일시적 컴퓨터 판독 가능 매체를 개시한다. 본원에 개시된 기술 및 시스템이, 예로서, 비디오 게임 애플리케이션, 특히 VR 게임 애플리케이션의 컨텍스트에서 논의되지만, 본원에 설명된 기술 및 시스템은, 제한 없이, 비VR 애플리케이션(예를 들어, AR 애플리케이션) 및/또는 산업 기계 애플리케이션, 방어 애플리케이션, 로봇 애플리케이션 등과 같은 비게임 애플리케이션을 포함하는 다른 애플리케이션에 이점을 제공할 수 있음을 이해해야 한다.

[0013] 도 1은 헤드 마운티드 디스플레이(HMD)(100)와 같은 디스플레이에 프레임을 렌더링하기 위해 재투영이 사용될 때 장면에서 이동하거나 움직이는 객체를 처리하기 위한 예시적인 모션 스무딩 기술을 나타낸 도면이다. HMD(100)가 개시된 모션 스무딩 기술을 구현할 수 있는 예시적인 "디스플레이 시스템"으로서 본원에 제시되었지만, "디스플레이 시스템"의 다른 유형 및/또는 구현은 본원에 설명된 모션 스무딩 기술을 구현할 수 있음을 이해해야 한다. 따라서, HMD가 단지 본원에서 설명된 기술을 구현하기 위한 예시적인 유형의 디스플레이일지라도, 본원에서 "HMD"에 대한 어떠한 언급도 "디스플레이"또는 "디스플레이 시스템"이라는 용어로 대체될 수 있음을 이해해야 한다.

[0014] 도 1의 예에서 HMD(100)는 스테레오 쌍의 디스플레이 패널의 좌측 디스플레이 패널 및 우측 디스플레이 패널과 같은 다수의 디스플레이 패널을 포함할 수 있다. 이러한 디스플레이 패널은 HMD(100)를 착용한 사용자(102)가 관측할 수 있는 일련의 이미지 프레임(이하, "프레임"으로 지칭됨)을 표현하는 데 사용될 수 있다. 2개의 패널로 구성된 HMD(100)에 대한 예가 설명되었지만, HMD(100)는 단일 디스플레이 패널 또는 2개보다 많은 디스플레이 패널을 포함할 수 있음을 이해해야 한다. 따라서, 본원에 사용된 바와 같은 "디스플레이 패널"이란 용어는 2-패널 HMD(100)의 한 쌍의 디스플레이 패널의 어느 하나의 디스플레이 패널을 지칭할 수 있거나, 또는 임의 개수의 디스플레이 패널을 갖는 HMD(100)(예를 들어, 단일 패널 HMD(100) 또는 다중 패널 HMD(100))의 단일 디스플레이 패널을 지칭할 수 있다. 2-패널 HMD(100)에서, 스테레오 프레임 버퍼는 예를 들어 HMD(100)의 양자의 디스플레이 패널에서 2160 x 1200 픽셀(예를 들어, 디스플레이 패널당 1080 x 1200 픽셀)을 렌더링할 수 있다.

[0015] HMD(100)는 HMD(100)의 디스플레이 패널(들)에 프레임을 표현하는 동안 발광 요소를 이용하여 광을 방출하는 발광 디스플레이와 같은 임의의 적절한 유형의 디스플레이 기술을 이용하는 디스플레이 패널을 포함할 수 있다. 일 예로서, HMD(100)의 디스플레이 패널은 액정 디스플레이(LCD), 유기 발광 다이오드(OLED) 디스플레이, 무기 발광 다이오드(ILED) 디스플레이, 또는 HMD 애플리케이션에 적합한 디스플레이 기술을 이용하는 임의의 다른 적절한 유형의 디스플레이를 포함할 수 있다.

[0016] HMD(100)는, VR 게임 시스템과 함께 사용하기 위한 것과 같은, VR 시스템에서 사용하기 위한 VR 헤드셋을 나타낼 수 있다. 그러나, HMD(100)는 추가적으로 또는 대안적으로 AR 애플리케이션에서 사용하기 위한 AR 헤드셋으로 구현될 수 있다. AR에서 사용자(102)는 실제 환경에 오버레이된 가상 객체를 보는 반면, VR에서 사용자(102)는 실제 환경을 보지 못하지만 HMD(100)의 디스플레이 패널 및 광학 장치(예를 들어, 렌즈)를 통해 인지되는 가상 환경에 완전히 몰입된다. 본원에 설명된 예는 주로 VR 기반 HMD(100)에 관련되지만, HMD(100)는 VR 애플리케이션에서의 구현으로 제한되지 않는다는 점을 이해해야 한다.

[0017] 일반적으로, HMD(100) 자체와 같은 컴퓨팅 장치 또는 HMD(100)와 연관되고 결합된 컴퓨팅 장치(예를 들어, 개인용 컴퓨터(PC), 게임 콘솔 등)에서 실행되는 애플리케이션은 HMD(100)의 디스플레이 패널(들)에 궁극적으로 렌더링되는 일련의 프레임(104)을 출력하도록 구성될 수 있다. 일부 실시예에서, 프레임(104)이 HMD(100)의 디스플레이 패널에 렌더링되기 전에 목표에 렌더링될 수 있도록 오프 스크린 렌더링이 이용된다. 따라서, 본원에서 사용된 바와 같은 "렌더링"은 디스플레이와 다른 목표에 대한 렌더링 및 디스플레이 자체에 렌더링하기 전에, 및/또는 디스플레이에 렌더링(예를 들어, 다른 목표에 대한 오프 스크린 렌더링 없이 또는 그 이후)하는 것을 포함할 수 있다.

[0018] 도 1의 예는 일련의 프레임(104)으로서 순차적으로 렌더링될 3개의 예시적인 프레임(104(1), 104(2), 및

104(3))을 나타낸다. 도 1의 예에서, 객체(106)는 프레임(104(1)-104(3))의 코스에 걸쳐 좌측 방향(즉, 우측에서 좌측으로)으로 장면을 가로질러 이동하는 것으로 도시되어 있다. 프레임(104(2) 및 104(3))에서 객체(106)의 점선 윤곽선은 이전에 렌더링된 프레임(104)에서 객체(106)가 위치되었던 장소를 나타낸다. 여기서, 프레임(104(1))이 먼저 렌더링된 후에 프레임(104(2))이 두 번째로 렌더링된 다음 프레임(104(3))이 세 번째로 렌더링된다.

[0019] 일련의 프레임(104)에서 프레임(104) 중 적어도 일부는 비디오 게임 애플리케이션과 같은 애플리케이션 또는 임의의 다른 유형의 그래픽 기반 애플리케이션으로부터 출력된다는 점에서 "실제" 프레임일 수 있다. 애플리케이션은 개별 프레임(104)을 렌더링하기 위해 픽셀 데이터(108)를 프레임 버퍼에 출력하는 그래픽 파이프라인에서 실행될 수 있다. 일부 실시예에서, 픽셀 데이터(108)는 픽셀당 값(예를 들어, 컬러 값)의 2차원 어레이를 포함할 수 있다. 일부 실시예에서, 픽셀 데이터(108)는 깊이 값, 휘도 값 등과 같은 추가 데이터 또는 메타데이터를 더 포함한다. 일부 실시예에서, 픽셀 데이터(108)는 단일 세트의 컬러 및 알파 값(예를 들어, 적색 채널에 대한 하나의 컬러 값, 녹색 채널에 대한 하나의 컬러 값, 청색 채널에 대한 하나의 컬러 값, 및 하나 이상의 알파 채널에 대한 하나 이상의 값)으로 표현되는 각 픽셀에 대한 데이터를 포함할 수 있다. 이러한 픽셀 데이터(108)는 원하는 시각 효과로 HMD(100)의 디스플레이 패널(들)에 이미지를 표현하기 위해 프레임 버퍼로 출력될 수 있다.

[0020] 런타임 동안, HMD(100)의 헤드 트래킹 모듈은 사용자(102)의 머리 위치/포즈에 따른 일련의 프레임(104)에서 다음 프레임(104)을 렌더링하는 방법에 대해 애플리케이션에 알리기 위해 애플리케이션에 제공되는 HMD(100)의 위치 및 포즈에 대한 데이터를 생성할 수 있다. 이를 통해 애플리케이션은 사용자(104)가 객체(이동체(106)와 같은 정적 및 이동 개체 모두)를 포함하는 가상 환경을 둘러보고 있다고 믿게 하는 방식으로 HMD(100)에 이미지를 렌더링하기 위한 픽셀 데이터(108)를 출력할 수 있다. 애플리케이션이 프레임률에 도달하는 경우에 사용자(102)의 머리 움직임과 함께 예상되는 방식으로 정적 및 이동 객체 모두 장면 내에서 움직이는 것으로 인지된다. 본원에 설명된 모션 스무딩 기술은 이동체에 대해 유사한 시각적 인지를 달성하도록 프레임률에 도달하지 못하는 애플리케이션을 보상하는 방식이다.

[0021] 도 1의 예에서, 제1 프레임(104(1))은 애플리케이션으로부터 수신되는 제1 "실제"프레임을 나타낼 수 있고, 제2 프레임(104(2))은 애플리케이션으로부터 수신되고 제1 프레임(104)(1) 이후에 렌더링된 제2 "실제" 프레임을 나타낼 수 있고, 제3 프레임(104(3))은 제2 프레임(104(2))과 연관된 픽셀 데이터(108(2))로부터 생성되는 재투영된 프레임을 나타낼 수 있다. 그러므로, 도 1의 "제3 프레임(104(3))"은 본원에서 때때로 "재투영된 프레임(104(3))"으로 지칭된다. HMD(100)는 일련의 프레임(104)을 처리할 수 있으며, 제1 프레임(104(1)) 및 제2 프레임(104(2))을 렌더링한 후에, HMD(100)의 로직은, 그래픽 처리 유닛(GPU)(110)(예를 들어, HMD(100)의 GPU(110))에 대한 입력으로서, 제1 프레임(104(1))과 연관된 제1 픽셀 데이터(108(1)) 및 제2 프레임(104(2))과 연관된 제2 픽셀 데이터(108(2))를 제공하도록 구성될 수 있다.

[0022] GPU(110)의 비디오 인코더(예를 들어, 비디오 인코드 칩)는 GPU(110)에 입력된 제1 픽셀 데이터(108(1)) 및 제2 픽셀 데이터(108(2))에 기초하여 모션 벡터(112)의 어레이를 생성할 수 있다. 모션 벡터(112)의 어레이를 생성하기 위해, GPU(110)의 비디오 인코더는 입력으로서 제공된 각 프레임(104)의 픽셀 데이터(108) 사이의 픽셀당 값(예를 들어, 휘도 값)을 비교할 수 있다. 추가적으로 또는 대안적으로, GPU(110)의 비디오 인코더는 입력으로서 제공된 각 프레임(104)의 픽셀 데이터(108) 사이에서 매크로블록(예를 들어, 16픽셀의 블록(즉, 4 x 4 픽셀 매크로블록), 64픽셀의 블록(즉, 8 x 8 픽셀 매크로블록))을 비교할 수 있다. 이러한 방식으로, GPU(110)는 임의의 적절한 해상도에서 한 쌍의 프레임(104) 사이에서 픽셀 데이터(108)의 부분을 비교할 수 있다. 일부 실시예에서, 입력 픽셀 데이터(108(1) 및 108(2))는 다운 샘플링된 프레임을 GPU(110)에 입력하기 위해 더 낮은 해상도로 다운 샘플링된다.

[0023] GPU(110)로부터 출력된 모션 벡터(112)의 어레이는 본원에서 때때로 "모션 벡터 필드(112)"로 지칭된다. 이러한 모션 벡터 필드(112)는 임의의 적절한 해상도에서 출력될 수 있고 및/또는 임의의 적절한 해상도로 다운 샘플링/업 샘플링될 수 있다. 예를 들어, 모션 벡터(112)의 어레이는 픽셀당 단일 모션 벡터(112), 픽셀의 그룹당 단일 모션 벡터(112)(예를 들어, 4 x 4 매크로블록, 8 x 8 매크로블록, 픽셀의 임의의 형상 패치 등에 대한 하나의 모션 벡터(112)), 또는 심지어 주어진 프레임(104)에 대한 모든 픽셀을 위한 단일 모션 벡터(112)를 포함할 수 있다. 일반적으로, 모션 벡터(112)는 방향에 대한 X 및 Y 성분 및 크기(통상적으로 2D 화살표의 길이로 표시됨)를 갖는 2차원(2D) 화살표이다. 모션 벡터(112)의 크기는 X 및 Y 성분 방향 모두에서 픽셀 수와 같은 임의의 적절한 측정 단위로 명시될 수 있다.

[0024] GPU(110)의 비디오 인코더에 의한 입력 픽셀 데이터(108)의 비교에 기초하여, 제2 프레임(104(2))의 일부가 제1

프레임(104)(1)의 일부와 유사한(예를 들어, 임계 휘도 값 내에서) 경우 및 각 프레임(104)의 유사한 부분이 거리(예를 들어, X 및 Y 성분 방향에서 픽셀 수)만큼 오프셋된 경우, 이는 모션 벡터(112)의 어레이에 포함되는 모션 벡터(112)로 표현될 수 있다. 프레임(104(1) 및 104(2))에서 객체(106)에 대응하는 픽셀 값이 유사하다고 GPU(110)의 비디오 인코더에 의해 결정되는(예를 들어, 픽셀 데이터(108)의 픽셀 값에 기초하여 일부 유사성 메트릭을 만족하는 정합 부분) 예를 고려한다. 이러한 객체(106)에 대한 모션 벡터(112)는 객체(106)의 이동과 동일한 방향을 가리키는 방향을 가질 수 있거나, 또는 모션 벡터(112)는 객체(106)의 이동 방향과 반대 방향을 가리킬 수 있다. 다시 말해서, 모션 벡터(112)는 후속 프레임(104(2))에서 객체(106)의 위치로부터 오프셋되는 이전 프레임(104(1))에서 객체(106)의 위치를 다시 참조하는 방향을 가리킬 수 있다. 따라서, 모션 벡터(112)는 제2 프레임(104(2))의 좌표로부터 제1 프레임(104(1))의 좌표까지 오프셋을 제공한다. 오프셋은 객체(106)와 같은 이동하거나 움직이는 객체에 대해 제1 프레임(104(1))의 이미지로부터 제2 프레임(104(2))의 이미지로의 변환을 설명한다.

[0025] 전술한 바와 같이, 도 1의 제3 프레임(104(3))은 재투영된 프레임을 나타낼 수 있는 데, 이는 제3 프레임(104(3))에 대한 픽셀 데이터(108(3))가 이전에 렌더링된 실제 프레임(이 경우, 제2 프레임(104(2)))과 연관된 픽셀 데이터(108(2))로부터 파생될 수 있음을 의미한다. 다시 말해서, 재투영된 프레임(104(3))에 대한 픽셀 데이터(108(3))는 애플리케이션(예를 들어, 비디오 게임 애플리케이션)에서 수신되지 않지만, 애플리케이션 생성 프레임의 픽셀 데이터(108)로부터 생성되고 애플리케이션이 프레임률에 도달하지 않을 때 누락된 프레임의 간격을 "채우는" 데 사용된다. 이 경우, 재투영된 프레임(104(3))에 대한 픽셀 데이터(108(3))는 제2 프레임(104(2))과 연관된 픽셀 데이터(108(2))로부터 생성되는 데, 이는 제2 프레임(104(2))이 도 1의 예에서 재투영된 프레임(104(3)) 이전에 가장 최근에 렌더링된 프레임(104)이기 때문이다. 일부 실시예에서, 회전 및 재투영 변환은 제2 프레임(104(2))과 연관된 제2 픽셀 데이터(108(2))를 수정하기 위해 계산되고 사용되어, 제2 프레임(104(2))이 렌더링된 이후 HMD(100)의 회전을 처리하는 방식으로 제2 프레임(104(2))에서 렌더링된 장면을 효과적으로 회전, 번역, 및/또는 이와 달리 이동시키는 재투영된 프레임에 대한 제3 픽셀 데이터(108(3))를 생성할 수 있다. 예를 들어, 사용자(102)는 재투영된 프레임(104(3))에 대한 제3 픽셀 데이터(108(3))를 생성 시에 처리되는 제2 프레임(104(2))이 렌더링될 때부터 자신의 머리를 회전시킬 수 있어, 이러한 머리 움직임에 따라 장면이 제시된다.

[0026] 그 후, 도 1의 모션 스무딩 기술은 모션 벡터(112)의 어레이에 적어도 부분적으로 기초하여 제3 픽셀 데이터(108(3))를 수정하여 재투영된 프레임(104(3))에 대한 수정된 제3 픽셀 데이터(108(3)')를 획득한다. 일부 실시예에서, 제3 픽셀 데이터(108(3))를 수정된 제3 픽셀 데이터(108(3)')로의 이러한 수정은 특정 픽셀 또는 픽셀의 그룹에 대응하는 비제로 모션 벡터(112)에 기초하여 제3 픽셀 데이터(108(3))의 픽셀 값을 상이한 위치로 이동하는 단계를 포함한다. 이동은 방향으로 그리고 양만큼 이루어질 수 있다(예를 들어, 수평(+/-) 및 수직(+/-) 방향으로 다수의 픽셀을 이동). 그 후, 재투영된 프레임(104(3))은 수정된 제3 픽셀 데이터(108(3)')에 적어도 부분적으로 기초하여 디스플레이(예를 들어, HMD(100)의 디스플레이 패널)에 렌더링된다. 따라서, 모션 벡터(112)(이전에 렌더링된 실제 프레임(104(1) 및 104(2))의 픽셀 데이터(108)로부터 생성된 모션 벡터(112))에 기초하여 수정된 재투영된 프레임(104(3))은 객체(106)를 예상된 위치에 렌더링하기 위해 "모션 스무딩"된다.

[0027] 도 2는 예시적인 렌더 메쉬(200) 및 재투영된 프레임을 위한 모션 스무딩에 렌더 메쉬(200)가 사용될 수 있는 방법을 나타낸 도면이다. 예를 들어, 렌더 메쉬(200)는 GPU(110)에 의해 출력된 모션 벡터(112)의 어레이에 기초하여 도 1의 재투영된 프레임(104(3))에 대한 제3 픽셀 데이터(108(3))를 수정하는 데 사용될 수 있다. 이러한 예에서, HMD(100)의 로직은 렌더 메쉬(200)를 생성할 수 있고, 렌더 메쉬(200)의 정점(202)은 모션 벡터(112)의 어레이에 따라 이동될 수 있다. 예를 들어, 정점(202)은 비제로 모션 벡터(112)의 방향으로 그리고 비제로 모션 벡터(112)의 크기에 대응하는 양만큼 이동될 수 있다. 예를 들어, 일 예로 도 1에 도시된 좌향 이동체(106)를 취하면, 모션 벡터(112)는 렌더 메쉬(200)의 컨텍스트에서 적용되어 정점(202)을 좌향(또는 음의 X) 방향으로 특정 개수의 픽셀(모션 벡터(112)의 크기에 대응) 이동시킬 수 있다.

[0028] 렌더 메쉬(200)는 복수의 정점(202(1), 202(2),…, 202(N))(총괄적으로 202)을 갖는 테셀레이션형 메쉬로 도시되어 있다. 렌더 메쉬(200)의 테셀레이션은 임의의 적절한 기하학적 패턴일 수 있다. 도 2의 예시적인 렌더 메쉬(200)는 삼각형(204)의 반복 패턴으로 도시되어 있지만, 사각형(때때로 "쿼드"로 지칭됨), 육각형(예를 들어, 허니콤 패턴의 경우) 등을 포함하나 이에 제한되지 않는 임의의 적절한 기하학적 형상이 렌더 메쉬(200)에 대해 사용될 수 있다. 이러한 예에서, 사각형(또는 쿼드)의 하단 좌측 코너로부터 사각형(또는 쿼드)의 상단 우측 코너까지의 대각선은 특정 방향을 갖는 반복적인 삼각형(204)의 렌더 메쉬(200)를 생성하기 위해 사용된다. 렌더 메쉬(200)에 대한 다른 방향은, 도 2에 도시된 방향 대신에, 사각형(또는 쿼드)의 상단 좌측 코너로부터 사각형

(또는 쿼드)의 하단 우측 코너까지의 대각선을 사용하여 사각형(또는 쿼드)을 삼각형(204)으로 분할함으로써 생성될 수 있다. 일부 실시예에서, 예를 들어 사각형(또는 쿼드)의 하단 좌측 코너로부터 사각형(또는 쿼드)의 상단 우측 코너까지의 대각선을 사용하여 다른 모든 사각형을 분할하고 사각형(또는 쿼드)의 상단 좌측 코너로부터 사각형(또는 쿼드)의 하단 우측 코너까지의 대각선을 사용하여 그 사이의 사각형을 분할함으로써, 이러한 상이한 방향의 혼합이 단일 렌더 메쉬(200)에 대해 사용될 수도 있다. 일부 실시예에서, HMD(100)의 로직은 GPU(110)로부터 수신된 모션 벡터 필드(112)에 기초하여 렌더 메쉬(200)의 생성 시에 사용하기 위해 이러한 다수 방향 중 어느 방향을 동적으로 결정하도록 구성될 수 있다. 이는 렌더 메쉬(200)에서 기하학적 형상(예를 들어, 삼각형(204))에 대한 최상의 방향을 선택하기 위해 수행될 수 있어 가장 스무딩하게 보이는 모션 스무딩된 이미지를 생성한다.

[0029] 렌더 메쉬(200)는 또한 임의의 적절한 해상도에서 생성될 수 있다. 예를 들어, 가장 높은 해상도의 렌더 메쉬(200)는 픽셀당 두 인접한 삼각형(204)일 수 있으며, 각 사각형(또는 쿼드)은 단일 픽셀에 매핑된다. 더 낮은 해상도는 16 픽셀의 그룹과 같이 픽셀의 그룹당 두 인접한 삼각형(204)일 수 있다. 대안적으로, 픽셀은 임의의 적절한 해상도에서 렌더 메쉬(200)의 정점(202)에 매핑될 수 있다. 예를 들어, 각 정점(202)은 가장 높은 해상도에서 단일 픽셀과 연관될 수 있거나, 각 정점(202)은 더 낮은 해상도에서 16 픽셀의 그룹과 같은 픽셀의 그룹과 연관될 수 있다. 일부 실시예에서, 렌더 메쉬(200)의 해상도는 모션 벡터(112)의 어레이의 해상도와 동일한 해상도이어서 모션 벡터(112)의 어레이에서 단일 모션 벡터(112)는 정점(202) 또는 사각형(또는)에 매핑된다(예를 들어, 2개의 인접한 삼각형(204)). 렌더 메쉬(200)와 모션 벡터(112)의 어레이 사이의 정합 해상도의 달성은 다양한 방식으로, 예를 들어 GPU(110)로부터 렌더 메쉬(200)의의 해상도와 정합하는 특정 해상도에서 모션 벡터(112)의 어레이를 요청하고, 모션 벡터(112)의 어레이를 다운 샘플링 또는 업 샘플링하여 렌더 메쉬(200)로서의 해상도와 정합하고, 또는 GPU(110)에 의해 출력되는 모션 벡터(112)의 어레이의 해상도와 정합하는 해상도에서 렌더 메쉬(200)를 생성함으로써 이루어질 수 있다.

[0030] 도 2는 4개의 비제로 모션 벡터(112(1), 112(2), 112(3), 및 112(4))가 렌더 메쉬(200)의 4개의 정점(202)에 대응하는 예를 도시하고 있다. 이러한 4개의 모션 벡터(112(1)-(4))는 GPU(110)가 도 1에 도시된 이동체(106)에 기초하여 검출한 모션 벡터(112)에 대응할 수 있다. 따라서, 예시적인 모션 벡터(112(1)-(4))는 객체(106)의 방향성 모션에 대응하는 좌향 방향을 가리킬 수 있지만, 전술한 바와 같이 방향성은 (예를 들어, 우향 방향으로) 객체(106)의 방향성 모션과 반대일 수 있다. 모션 벡터(112)의 방향성은 원하는 방향으로 픽셀 데이터(108)를 수정하기 위해 모션 스무딩 알고리즘에서 고려될 수 있다. 모션 벡터(112)의 필드가 4개의 예시적인 모션 벡터(112(1)-(4))를 포함하고 모션 벡터(112)의 어레이에서 모든 잔여 모션 벡터(112)가 제로 벡터인 기본 예를 고려한다. 이러한 예에서, 재투영된 프레임(104(3))에 대한 픽셀 데이터(108(3))는, 비제로 모션 벡터(112(1)-(4))에 기초하여, 비제로 모션 벡터(112(1)-(4))에 대응하는 정점(202)을 (도 2의 하단에 도시된) 이동된 정점(206(1), 206(2), 206(3), 및 206(4))으로서 렌더 메쉬(200) 내의 상이한 위치로 이동시킴으로써 수정될 수 있다. 도 2의 하단은 모션 스무딩이 적용된 후의 렌더 메쉬(200)를 도시한 것으로, 이동된 정점(206(1)-(4))은 모션 스무딩 전의 정점(202)의 위치에 비해 렌더 메쉬(200) 내의 상이한 위치에 있다. 모션 벡터(112)가 이러한 방식으로 적용될 때, 이동된 정점(206(1)-(4))은 예를 들어 렌더 메쉬(200)에서 기하학적 형상(예를 들어, 삼각형(204)) 중 특정 형상을 신장하거나 둘러쌈으로써 렌더 메쉬(200)의 하나 이상의 부분이 왜곡된다. 도 2의 예에서, 도 2의 하단에 도시된 바와 같이 삼각형(204) 중 일부는 모션 스무딩의 결과로 신장되어 신장된 삼각형(208)을 생성한다. 이동된 정점(206(1)-(4))에 대응하는 (재투영된 프레임(104(3))에 대한 픽셀 데이터(108(3))의) 픽셀 값은 렌더 메쉬(200) 내에서 이동된 정점(206(1)-(4))의 위치에 대응하는 상이한 픽셀 위치에서 렌더링된다. 이동된 정점(206(1)-(4))과 비이동된 정점(202) 사이의 픽셀 위치는 (예를 들어, 이동된 정점(206(1)-(4))과 비이동된 정점(202) 사이의 픽셀 값을 보간하는 것과 같은 구배를 적용함으로써) 블렌딩될 수 있다. 일부 실시예에서, 깊이 버퍼는 재투영된 프레임(104(3))에 대한 수정된 픽셀 데이터(108(3)')의 프레임 버퍼에 출력되는 픽셀 값의 최종 세트를 결정하는 데 이용될 수 있다. 즉, 모션 벡터(112)를 렌더 메쉬(200)에 적용한 결과, 이동된 정점(206(1)-(4))에 대응하는 이미지의 위치에 다수의 픽셀 값이 있을 수 있다. 이 경우, "더 가까운"(더 작은) 깊이 값과 연관된 어느 픽셀 값이든 "더 먼"(더 큰) 깊이 값과 연관된 해당 위치에서 다른 픽셀 값을 렌더링하는 대신 렌더링될 수 있다.

[0031] 본원에 설명된 프로세스는 하드웨어, 소프트웨어, 펌웨어 또는 이들의 조합(즉, 로직)으로 구현될 수 있는 일련의 작업을 나타내는 논리 흐름 그래프의 블록 집합으로서 예시된다. 소프트웨어의 경우, 블록은 하나 이상의 프로세서에 의해 실행될 때 언급된 작업을 수행하는 컴퓨터 실행 가능 명령을 나타낸다. 일반적으로, 컴퓨터 실행 가능 명령은 특정 기능을 수행하거나 또는 특정 추상 데이터 유형을 구현하는 루틴, 프로그램, 객체, 구성요소, 데이터 구조 등을 포함한다. 작업이 설명되는 순서는 제한적으로 해석되지 않으며, 설명된 블록의 임의의 수는

임의의 순서로 그리고/또는 병렬로 결합되어 방법을 구현할 수 있다.

[0032] 도 3은 본원에 개시된 실시예에 따른 모션 스무딩 기술의 일부로서 GPU에 의해 생성된 모션 벡터를 사용하여 재투영된 프레임을 생성하기 위한 예시적인 프로세스(300)의 흐름도이다. 논의를 위해, 프로세스(300)는 이전 도면을 참조하여 설명된다.

[0033] 302에서, 디스플레이 시스템의 로직(예를 들어, HMD(100)의 로직)은 GPU(110)에 대한 입력으로서 이전에 렌더링된 프레임(104)과 연관된 픽셀 데이터(108)를 제공할 수 있다. 예를 들어, 가장 최근에 렌더링된 2개의 프레임(104)과 연관된 픽셀 데이터(108)는 GPU(110)(예를 들어, HMD(100)의 GPU(110))에 입력으로서 제공될 수 있다. 이러한 프레임(104)은 이전에 렌더링된 제1 프레임(104(1)) 및 제1 프레임(104(1)) 이후에 이전에 렌더링된 제2 프레임(104(2))과 같은 애플리케이션(예를 들어, 비디오 게임 애플리케이션)으로부터 수신된 실제 프레임일 수 있다. 따라서, 제2 프레임(104(2))은 애플리케이션으로부터 수신된 가장 최근에 렌더링된 프레임을 나타낼 수 있고, 제1 프레임(104(1)) 및 제2 프레임(104(2))은 일련의 프레임(104)에서 연속적으로 렌더링될 수 있지만, 블록 302에서 입력으로서 제공된 픽셀 데이터(108)는 연속적으로 렌더링된 프레임에 대한 픽셀 데이터(108)일 필요는 없다. 예를 들어, 중간 프레임(들)(104)은 제1 프레임(104(1))과 제2 프레임(104(2)) 사이에 렌더링될 수 있고, 블록 302에서 입력으로서 제공된 픽셀 데이터(108)는 제1 프레임(104(1)) 및 제2 프레임(104(2))과 관련될 수 있다.

[0034] 304에서, 모션 벡터(112)의 어레이는 GPU(110)로부터 수신될 수 있다. 블록 304에서 수신된 모션 벡터(112)의 어레이는 제1 프레임(104(1))과 연관된 제1 픽셀 데이터(108(1)) 및 제2 프레임(104(2))과 연관된 제2 픽셀 데이터(108(2))에 적어도 부분적으로 기초하여(예를 들어, 제1 픽셀 데이터(108(1))와 제2 픽셀 데이터(108(2)) 간의 비교에 기초하여) GPU(110)의 비디오 인코더에 의해 생성될 수 있다. GPU(110)의 비디오 인코더는, 예를 들어 비교된 픽셀 값 간의 차이가 임계 차이보다 작은지를 결정함으로써, 픽셀 값(또는 픽셀 값의 그룹) 간의 유사성을 찾는 적절한 컴퓨터 비전 및/또는 비디오 인코딩 알고리즘을 사용하도록 구성될 수 있다. 이러한 유사성 메트릭 내의 무엇이든 두 프레임(104) 사이에서 정합 픽셀 데이터(108)인 것으로 간주될 수 있다.

[0035] 306에서, 로직은 제2 프레임(104(2))의 제2 픽셀 데이터(108(2))에 적어도 부분적으로 기초하여 재투영된 프레임(104(3))에 대한 제3 픽셀 데이터(108(3))를 생성할 수 있다. 이 경우, 제2 프레임(104(2))은 재투영된 프레임(104(3)) 직전에 렌더링된 프레임(104)을 나타낸다.

[0036] 308에서, 로직은 모션 벡터(112)의 어레이에 적어도 부분적으로 기초하여 제3 픽셀 데이터(108(3))를 수정해서 재투영된 프레임(104(3))에 대한 수정된 제3 픽셀 데이터(108(3)')를 획득할 수 있다. 서브블록 310 및 312에 의해 도시된 바와 같이, 재투영된 프레임(104(3))에 대한 픽셀 데이터(108(3))의 수정은 렌더 메쉬(200)를 이용할 수 있다.

[0037] 따라서, 310에서, 로직은 재투영된 프레임(104(3))에 대한 렌더 메쉬(200)를 생성할 수 있다. 렌더 메쉬(200)는 복수의 정점(202)을 갖는 테셀레이션형 메쉬를 포함할 수 있다. 일부 실시예에서, 렌더 메쉬(200)의 해상도는 모션 벡터(112)와 렌더 메쉬(200)의 "요소"(예를 들어, 렌더 메쉬(200)의 정점(202), 렌더 메쉬(200)의 사각형(또는 쿼드) 등과 같은 "요소") 사이에 일대일 대응이 있도록 모션 벡터(112)의 어레이의 해상도와 정합할 수 있다. 모션 벡터 필드(112)와 렌더 메쉬(200) 사이의 정합 해상도를 획득하는 단계는 GPU(110)가 모션 벡터 필드(112)를 특정 해상도에서 출력하도록 요청하는 단계, 모션 벡터 필드(112)의 해상도를 다운 샘플링 또는 업 샘플링하는 단계, 및/또는 GPU(110)에 의해 출력된 모션 벡터 필드(112)의 해상도와 정합하는 해상도에서 렌더 메쉬(200)를 생성하는 단계와 같은 본원에 설명된 임의의 기술을 포함할 수 있다.

[0038] 312에서, 로직은 렌더 메쉬(200)의 (복수의 정점(202) 중) 정점(202)을 이동된 정점(206)으로서 렌더 메쉬(200) 내의 상이한 위치로 이동시킬 수 있다. 정점(202)은 (i) 비제로 모션 벡터(112)의 방향으로 그리고 (ii) 비제로 모션 벡터(112)의 크기에 대응하는 양만큼 이동될 수 있다. 따라서, 블록 308에서 제3 픽셀 데이터(108(3))의 수정은, 예를 들어 제3 픽셀 데이터(108(3))의 픽셀 값을 이동 정점(206)에 따라 이동시켜 재투영된 프레임(104(3))의 수정된 제3 픽셀 데이터(108(3)')를 획득함으로써, 렌더 메쉬(200)의 이동된 정점(206)에 따라 이루어질 수 있다. 예를 들어, 제3 픽셀 데이터(108(3))의 픽셀 값은 렌더 메쉬(200) 내의 이동된 정점(206)에 따라 수정된 제3 픽셀 데이터(108(3)') 내의 새로운 위치로 좌측으로 4 픽셀, 및 상향으로 4 픽셀 이동될 수 있다.

[0039] 일부 실시예에서, 다수의 모션 벡터 필드(112)는 블록 302에서 입력된 이전에 렌더링된 프레임(104)의 상이한 세트에 기초하여 블록 304에서 수신될 수 있고, 추가 모션 관련 파라미터는 재투영된 프레임에 대해 블록 308에서 모션 스무딩에서 사용하기 위해 다수의 모션 벡터 필드(112)에 기초하여 결정될 수 있다. 예를 들어, 블록

304에서 수신된 모션 벡터(112)의 어레이를 생성하는 이전의 두 프레임(104(1) 및 104(2))에 추가하여, 도 3의 알고리즘은, 일부 실시예에서, 예를 들어 제1 프레임(104(1)) 및 제1 프레임(104(1)) 이전에 렌더링된 "제0" 프레임(104(0))을 GPU(110)에 대한 입력으로서 제공하고 해당 다른 쌍의 입력 프레임(104)에 기초하여 모션 벡터(112)의 추가 어레이를 수신함으로써, 하나 이상의 추가 프레임으로 되돌아갈 수 있다. 그 후, 모션 벡터의 다수 어레이는 프레임 간 이동하는 객체의 가속과 유사한 모션 관련 파라미터를 결정하기 위해 비교될 수 있으며, 이러한 모션 관련 파라미터는, 예를 들어 제3 픽셀 데이터(108(3))에 적용되는 최종 모션 벡터(112)의 크기를 수정(예를 들어, 증가/감소)하여 모션 스무딩 조정 측면에서 픽셀 값을 다소 이동시킴으로써, 블록 308에서 적용될 수 있다.

[0040] 314에서, 로직은 재투영된 프레임(104(3))을 수정된 제3 픽셀 데이터(108(3)')에 적어도 부분적으로 기초하여 디스플레이(예를 들어, HMD(100)의 디스플레이 패널(들))에 렌더링할 수 있다. 블록 314에서 렌더링되는 결과적인 재투영된 프레임(104(3))은 GPU(110)로부터 수신된 모션 벡터(112)에 따라 제3 픽셀 데이터(108(3))의 수정에 의해 "모션 스무딩"된다. GPU(110)는 하나 이상의 GPU(110)를 나타낼 수 있음을 이해해야 한다. 예를 들어, 다수의 GPU(110)는 HMD(100)의 스테레오 디스플레이 패널에 주어진 프레임(104)을 렌더링하기 위해 이용될 수 있고, 이러한 GPU(110)에 입력되는 프레임(104)의 픽셀 데이터(108)는 그에 따라 분할될 수 있다(예를 들어, 픽셀 데이터(108)의 좌측 절반은 제1 GPU(110)에 입력으로서 제공될 수 있고, 픽셀 데이터(108)의 우측 절반은 제2 GPU(110)에 입력으로서 제공될 수 있다).

[0041] 도 4는 본원에 개시된 실시예에 따른 픽셀 데이터가 모션 벡터 추정을 위해 GPU에 입력되기 전에 이전에 렌더링된 프레임의 픽셀 데이터를 정렬하기 위한 예시적인 프로세스(400)의 흐름도이다. 논의를 위해, 프로세스(400)는 이전 도면을 참조하여 설명된다. 또한, 도 3 및 4에서 오프 페이지 참조 "A"로 도시된 바와 같이, 프로세스(400)는 도 3의 블록 304에서의 동작 이전에 수행되는 동작을 나타낼 수 있고, 프로세스(400)는 일부 실시예에서 블록 304-314의 동작을 계속할 수 있다.

[0042] 402에서, HMD(100)의 로직은, HMD(100)의 헤드 트래킹 모듈에 의해 제공된 회전 데이터에 기초하여, 픽셀 데이터(108)가 GPU(110)에 입력으로서 제공될 이전에 렌더링된 프레임(104)을 렌더링하는 사이에서 HMD(100)가 제1 방향으로부터 제2 방향으로 회전한 것으로 결정할 수 있다. 이는 사용자(102)가, 제1 프레임(104(1)) 및 제2 프레임(104(2))을 렌더링하는 시간에 대응하는 것으로 여겨지는, 시간 t1과 시간 t2 사이에 우향 방향으로 머리를 회전한 것으로 나타낸 도 1의 예에 도시되어 있다. 연속적인 프레임(104) 사이에서 도 1에 도시된 회전량은 예시를 위해 과장되어 있다.

[0043] 404에서, 이전에 렌더링된 프레임(104)은 해당 프레임(104)에 대한 픽셀 데이터(108)가 GPU(110)에 입력으로서 제공되기 전에 정렬될 수 있다. 이는 두 프레임(104(1) 및 104(2))을 렌더링하는 과정에서 객체(고정 및 이동체 둘 모두)가 위치 사이를 이동하게 할 수 있는 프레임(104) 사이의 HMD(100)의 이동으로 인해 이루어진다. 하나의 프레임(104(1))을 다른 프레임(104(2))과 정렬하거나 그 반대로 정렬함으로써, 장면 내의 특정 정적 객체를 나타내는 픽셀 데이터(108)는 일반적으로 두 프레임(104(1) 및 104(2)) 사이의 동일한 위치로 이동될 수 있어, 정적 객체에 대응하는 픽셀 데이터(108)는 GPU(110)의 비디오 인코더에 의해 이동체로 오인되지 않는다. 블록 404에서의 정렬은 (i) 제2 프레임(104(2))을 렌더링할 때에 제1 프레임(104(1))의 장면을 HMD(100)의 제2 방향과 정렬하는 수정된 제1 픽셀 데이터(408(1))를 획득하기 위한 (제1 프레임(104(1))과 연관된) 제1 픽셀 데이터(108(1)), 또는 (ii) 제1 프레임(104(1))을 렌더링할 때에 제2 프레임(104(2))의 장면을 HMD(100)의 제1 방향과 정렬하는 수정된 제2 픽셀 데이터를 획득하기 위한 (제2 프레임(104(2))과 연관된) 제2 픽셀 데이터(108(2)) 중 적어도 하나를 수정하는 단계를 포함할 수 있다. 도 4의 하단의 도면은 HMD(100)의 제2 방향과 제1 프레임(104(1))의 장면을 정렬하는 수정된 제1 픽셀 데이터(408(1))를 획득하기 위해 제1 픽셀 데이터(108(1))가 수정되는 전자의 경우를 도시하고 있다. 그러나, 제1 픽셀 데이터(108(1)) 또는 제2 픽셀 데이터(108(2))는 정렬 목적을 위해 수정될 수 있음을 이해해야 한다.

[0044] 406에서, 로직은 GPU(110)에 대한 입력으로서 이전에 렌더링된 프레임(104) 중 하나의 수정된 픽셀 데이터 및 다른 프레임(104)의 원래 픽셀 데이터(108)를 제공할 수 있다. 도 4의 도면에 나타낸 예는 수정된 제1 픽셀 데이터(408(1)) 및 원래의 제2 픽셀 데이터(108(2))를 GPU(110)에 대한 입력으로서 제공하는 것을 도시하고 있다. 전술한 바와 같이, 프로세스(400)는 블록 406으로부터 프로세스(300)의 블록 304로 계속될 수 있다(오프 페이지 참조 "A"로 도시됨). 따라서, 모션 벡터(112)의 어레이는 블록 406에서 GPU(110)에 대한 입력으로서 제공되는 픽셀 데이터에 기초하여 블록 304에서 GPU(110)로부터 수신될 수 있고, 도 3의 모션 스무딩 알고리즘의 잔여 동작은 모션 스무딩되는 재투영된 프레임(104(3))을 렌더링하기 위해 수행될 수 있다.

[0045] 도 5는 본원에 개시된 실시예에 따른 이전에 렌더링된 프레임의 일부 픽셀 데이터를 배제하고 픽셀 데이터의 잔여 부분을 모션 벡터 추정을 위해 GPU에 입력으로서 제공하기 위한 예시적인 프로세스(500)의 흐름도이다. 논의를 위해, 프로세스(500)는 이전 도면을 참조하여 설명된다. 또한, 도 3 및 5에서 오프 페이지 참조 "A"로 도시된 바와 같이, 프로세스(500)는 도 3의 블록 304에서의 동작 이전에 수행되는 동작을 나타낼 수 있고, 프로세스(500)는 일부 실시예에서 블록 304-314의 동작을 계속할 수 있다.

[0046] 502에서, HMD(100)의 로직은, HMD(100)의 헤드 트래킹 모듈에 의해 제공된 회전 데이터에 기초하여, GPU(110)에 입력으로서 제공될 이전에 렌더링된 프레임(104)을 렌더링하는 사이에서 HMD(100)가 제1 방향으로부터 제2 방향으로 회전한 것으로 결정할 수 있다. 이전에 렌더링된 프레임(104)은 도 1에 도시된 제1 프레임(104(1)) 및 제2 프레임(104(2))일 수 있다.

[0047] 504에서, 로직은 GPU(110)에 대한 입력으로서 이용 가능한 픽셀 데이터(108)의 일부를 제공할 수 있다. GPU(110)에 입력으로서 제공되는 이용 가능한 픽셀 데이터(108)의 부분은 예를 들어 제1 픽셀 데이터(108(1))의 부분(506(1)) 및 제2 픽셀 데이터(108(1))의 부분(506(2))을 포함할 수 있으며, 이들 각각은 HMD(100)의 디스플레이 패널의 하나 이상의 에지에서 픽셀의 서브셋 이외의 픽셀에 대응한다. 예를 들어, 도 5에 도시된 바와 같이, GPU(110)에 입력으로서 제공된 픽셀 데이터(108)의 부분(506)은 디스플레이 패널의 좌우 에지에서 픽셀 데이터(108)의 잔여 부분을 배제한다(도 5에서 흑색으로 도시됨). 다시 말해서, 디스플레이 패널의 좌우 에지에서의 픽셀 데이터(108)는 GPU(110)에 제공되지 않으므로 GPU(110)는 디스플레이 패널의 좌우 에지 사이의 픽셀 데이터(108)의 중앙부(506)에 모션 추정 노력을 집중시킨다. 양 및/또는 음의 수직 방향으로 HMD(100) 회전의 경우, 픽셀 데이터(108)의 배제된 부분은 이미지의 상단 및 하단 에지일 수 있다. 어쨌든, 이미지의 에지(들)에서 픽셀 데이터(108)의 일부를 제외하고 픽셀 데이터(108)의 잔여 부분(506)을 GPU(110)에 배타적으로 제공하면 이미지의 에지(들)에서 효과적으로 제로 모션 벡터(112)가 되고, 이는 그렇지 않으면 이동체(106)의 특성이 아닌 이상점 모션 벡터(112)를 초래할 수 있는 이미지의 에지 근처에서 원치 않는 시각적 아티팩트가 나타나는 상황에서 유용할 수 있다. 전술한 바와 같이, 프로세스(500)는 블록 504로부터 프로세스(300)의 블록 304로 계속될 수 있으며, 이 경우 모션 벡터(112)의 어레이는 블록 504에서 GPU(110)에 대한 입력으로서 제공되는 픽셀 데이터에 기초하여 GPU(110)로부터 수신되고, 도 3의 모션 스무딩 알고리즘은 모션 스무딩되는 재투영된 프레임(104(3))을 렌더링하기 위해 수행될 수 있다.

[0048] 도 6은 본원에 개시된 실시예에 따른 이전에 렌더링된 프레임의 픽셀 데이터를 정렬하고, 이전에 렌더링된 프레임의 픽셀 데이터의 패널 마스크부를 배제하고, 픽셀 데이터의 잔여 부분을 모션 벡터 추정을 위해 GPU에 입력으로서 제공하기 위한 예시적인 프로세스(600)의 흐름도이다. 논의를 위해, 프로세스(600)는 이전 도면을 참조하여 설명된다. 또한, 도 3 및 6에서 오프 페이지 참조 "A"로 도시된 바와 같이, 프로세스(600)는 도 3의 블록 304에서의 동작 이전에 수행되는 동작을 나타낼 수 있고, 프로세스(600)는 일부 실시예에서 블록 304-314의 동작을 계속할 수 있다.

[0049] 602에서, HMD(100)의 로직은, HMD(100)의 헤드 트래킹 모듈에 의해 제공된 회전 데이터에 기초하여, GPU(110)에 입력으로서 제공될 이전에 렌더링된 프레임(104)을 렌더링하는 사이에서 HMD(100)가 제1 방향으로부터 제2 방향으로 회전한 것으로 결정할 수 있다. 이전에 렌더링된 프레임은 도 1에 도시된 제1 프레임(104(1)) 및 제2 프레임(104(2))일 수 있다.

[0050] 604에서, 로직은 제1 프레임(104(1))과 연관된 제1 픽셀 데이터(108(1))를 수정하여 HMD(100)의 제2 방향과 제1 프레임(104(1))의 장면을 정렬하는 수정된 제1 픽셀 데이터(608(1))를 획득할 수 있다. 도 6의 예에서, 제1 픽셀 데이터(108(1))의 일부는 각 프레임에서 HMD(100)의 디스플레이 패널의 주변에서 렌더링된 패널 마스크(흑색으로 도시됨)에 대응하는 데이터를 나타낸다. 따라서, 수정된 제1 픽셀 데이터(608(1))의 패널 마스크 부분(614(1))(즉, 도 6에서 흑색으로 도시된 부분)은 수정된 제1 픽셀 데이터(608(1))의 패널 마스크에 대응하는 데이터를 나타낸다.

[0051] 606에서, 로직은 제2 프레임(104(2))과 연관된 제2 픽셀 데이터(108(2))를 수정하여 HMD(100)의 제1 방향과 제2 프레임의 장면을 정렬하는 수정된 제2 픽셀 데이터(608(2))를 획득할 수 있다. 다시, 제2 픽셀 데이터(108(2))의 일부는 각 프레임에서 HMD(100)의 디스플레이 패널의 주변에서 렌더링된 패널 마스크(흑색으로 도시됨)에 대응하는 데이터를 나타낸다. 따라서, 수정된 제2 픽셀 데이터(608(2))의 패널 마스크 부분(614(2))(즉, 도 6에서 흑색으로 도시된 부분)은 수정된 제2 픽셀 데이터(608(2))의 패널 마스크에 대응하는 데이터를 나타낸다.

[0052] 610에서, 로직은 수정된 제1 픽셀 데이터(608(1))의 패널 마스크 부분(614(1))을 수정된 제2 픽셀 데이터(608(2))의 패널 마스크 부분(614(2))과 조합하여 패널 마스크(616)에 대응하는 공통 값을 갖는 픽셀의 서브셋

을 결정할 수 있다. 이는 일종의 벤 다이어그램으로 생각될 수 있는 데, 이 경우 패널 마스크(616)에 대응하는 픽셀의 서브셋은 수정된 제1 픽셀 데이터(608(1)) 및 수정된 제2 픽셀 데이터(608(2))에서 수정된 패널 마스크의 조합된 버전이다.

[0053] 612에서, 로직은, GPU(110)에 대한 입력으로서, 제1 픽셀 데이터(108(1))의 특정 부분(예를 들어, 중앙부) 및 제2 픽셀 데이터(108(2))의 특정 부분(예를 들어, 중앙부)을 제공할 수 있으며, 이들 각각은 패널 마스크(616)에 대응하는 픽셀의 서브셋 이외의 픽셀에 대응한다. 이를 통해 GPU(110)는 패널 마스크(616) 픽셀에 의해 커버되는 각각의 이전에 렌더링된 프레임에서 픽셀 데이터(108)의 일부에 대한 모션 추정을 무시할 수 있고, GPU(110)는 모션 추정 노력을 각 프레임(104)에 대한 픽셀 데이터(108)의 중앙의 비패널 마스크 부분에 집중시킬 수 있다. 전술한 바와 같이, 프로세스(600)는 블록 612로부터 프로세스(300)의 블록 304로 계속될 수 있으며, 이 경우 모션 벡터(112)의 어레이는 블록 612에서 GPU(110)에 대한 입력으로서 제공되는 픽셀 데이터에 기초하여 GPU(110)로부터 수신되고, 도 3의 모션 스무딩 알고리즘은 모션 스무딩되는 재투영된 프레임(104(3))을 렌더링하기 위해 수행될 수 있다.

[0054] 도 7은 본원에 개시된 실시예에 따른 모션 스무딩 기술의 일부로서 재투영된 프레임을 생성하기 위해 사용되기 전에 GPU로부터 출력되는 모션 벡터를 임계화하기 위한 예시적인 프로세스(700)의 흐름도이다. 논의를 위해, 프로세스(700)는 이전 도면을 참조하여 설명된다.

[0055] 702에서, 디스플레이 시스템의 로직(예를 들어, HMD(100)의 로직)은 GPU(110)에 대한 입력으로서 이전에 렌더링된 프레임(104)과 연관된 픽셀 데이터(108)를 제공할 수 있다. 블록 702에서 수행된 동작(들)은 프로세스(300)의 블록 302와 관련하여 설명된 것과 유사할 수 있다.

[0056] 704에서, 로직은 모션 벡터(112)의 어레이를 GPU(110)로부터 수신할 수 있다. 블록 704에서 수행된 동작(들)은 프로세스(300)의 블록 304와 관련하여 설명된 것과 유사할 수 있다.

[0057] 706에서, 로직은 모션 벡터(112)의 어레이에서 개별 모션 벡터의 크기(또는 길이)를 제1 임계 크기와 비교하여 제1 임계 크기보다 큰 크기를 갖는 모션 벡터(112)의 제1 서브셋을 결정할 수 있다. 제1 임계 크기는, 가상 객체(106)의 이동 또는 움직임 이외의 것인 프레임 간의 변화를 나타낼 수 있는, 큰 크기의 이상점 모션 벡터(112)의 이러한 서브셋의 영향을 완화하는 데 이용될 수 있다. 비정상적으로 큰 크기의 모션 벡터는 다양한 이유로 발생할 수 있다. GPU(110)는 제1 프레임(104(1))의 하단 좌측에서의 픽셀과 충분히 유사할 제2 프레임(104(2))의 상단 우측에서의 픽셀을 찾고, 모션 벡터(112)가 프레임(104) 사이의 가상 객체(106)의 이동 또는 움직임을 나타내지 않더라도 어레이에서 다른 모션 벡터(112)에 비해 상대적으로 큰 크기를 갖는 결과적인 모션 벡터(112)를 출력할 수 있다. 일부 경우에, 비디오 게임을 통해 사용자는 장면이 프레임(104) 사이에서 급격하게 변경되는 다른 위치로 텔레포팅하여, 큰 모션 벡터(112)가 GPU(110)의 비디오 인코더에 의해 생성되게 할 수 있다. 이러한 및 다른 경우에, 이러한 큰 모션 벡터(112)를 임계화하는 것이 유용할 수 있다.

[0058] 707에서, 다수의 이전에 렌더링된 프레임(104)이 제1 프레임(104(1)) 및 제2 프레임(104(2))을 포함할 때, 로직은 제1 프레임(104(1))이 렌더링된 제1 시간과 제2 프레임(104(2))이 렌더링된 제2 시간 사이의 기간(또는 간격)를 결정할 수 있다.

[0059] 709에서, 블록 706에서의 비교에 사용되는 제1 임계 크기는 제1 시간과 제2 시간 사이의 기간에 적어도 부분적으로 기초하여 선택될 수 있다. 예시적인 예에서, 단위 시간당 사용자의 관점으로부터 이동도로 측정된 제1 임계 크기는 두 프레임(104(1) 및 104(2)) 사이의 11.1 ms 시간당 6도의 이동이다. 따라서, 두 프레임(104(1) 및 104(2))이 시간 상 더 오래 떨어져 있을수록, 제1 임계 크기가 더 커지고, 이에 따라 모션 벡터(112)가 (크기상) 더 길어질 수 있다. 이는 특정 속도를 초과하는 모션이 있으면 모션 스무딩이 그다지 효과적이지 않을 수 있고 실제로 (예를 들어, 장면의 일부를 "스위미"하게 보이게 함으로써) 부정적인 시각적 아티팩트를 유발할 수 있기 때문이다.

[0060] 708에서, 로직은 제1 임계 크기보다 큰 크기를 갖는 것으로 결정된 모션 벡터(112)의 제1 서브셋의 크기를 감소시켜 그 크기는 제1 임계 크기에서 캡핑될 수 있다. 이는 모션 벡터(112)의 제1 수정된 어레이를 생성한다. 다시 말해서, 제1 임계 크기를 초과하는 해당 모션 벡터(112)의 경우, 로직은 해당 모션 벡터(112)의 크기를 제1 임계 크기로 캡핑(또는 제한)하도록 구성되어, 모션 벡터(112)의 제1 수정된 어레이는 제1 임계 크기 이하인 크기를 포함하고 제1 임계 크기보다 큰 어떠한 크기도 포함하지 않는다. 일부 실시예에서, 모션 벡터(112)의 크기를 제1 임계 크기에서 캡핑하고 모션 스무딩을 위해 모든 비제로 모션 벡터를 사용하는 대신, 로직은 제1 임계 크기를 충족하거나 초과하는 크기를 갖는 모션 벡터(112)의 제1 서브셋을 폐기하여 모션 스무딩에서 전혀 사용

되지 않을 수 있다.

[0061] 710에서, 로직은 모션 벡터(112)의 어레이에서 개별 모션 벡터의 크기(또는 길이)를 제2 임계 크기와 비교하여 제2 임계 크기보다 작은 크기를 갖는 모션 벡터(112)의 제2 서브셋을 결정할 수 있다. 제2 임계 크기는, 사용자의 일정한 머리 움직임 및/또는 정확하지만 절대 제로 모션의 정도까지는 정확하지 않은 헤드 트랙킹에 의해 종종 야기되는 프레임 간의 변화를 나타낼 수 있는, 작은 크기의 이상점 모션 벡터(112)의 이러한 서브셋의 영향을 완화하는 데 이용될 수 있다. 이 때문에, GPU(110)의 출력은 제로 길이 모션 벡터(112)를 거의 제공하지 않는다. 오히려, GPU(110)의 출력이 전술한 바와 같이 주변 헤드 모션 및/또는 트래킹 지터로 인해 상당한 양의 노이즈를 갖는 것이 더 일반적이다. 다시 말해서, 두 연속 프레임 사이의 픽셀이 거의 100% 일치하지 않는다. 예시적인 예에서, 모션 픽셀로 측정된 제2 임계 크기는 2 모션 픽셀의 임계치이다. 이러한 예에서, 크기(또는 길이)가 2 모션 픽셀 미만인 임의의 모션 벡터는 작은 크기 이상점 벡터로 간주된다.

[0062] 712에서, 로직은 제2 임계 크기(예를 들어, 2 모션 픽셀 미만의 길이를 갖는 작은 크기의 이상점 모션 벡터)보다 작은 크기를 갖는 것으로 결정된 모션 벡터(112)의 제2 서브셋의 크기를 제로의 길이/크기로 감소시킬 수 있다. 이는 제2 임계 크기보다 작은 크기를 갖는 어떠한 모션 벡터도 포함하지 않는 모션 벡터(112)의 제2 수정된 어레이를 생성한다. 이는 본원에서 때때로 모션 벡터(112)에 작은 데드 존을 적용하는 것으로 지칭된다.

[0063] 714에서, 로직은 제2 임계치보다 작은 크기를 갖는 어떠한 모션 벡터(112)도 없이 제1 임계 크기에서 캡핑된 크기를 갖는 모션 벡터(112)의 제2 수정된 어레이에 기초하여 재투영된 프레임에 대한 픽셀 데이터(108(3))를 수정할 수 있다. 일부 실시예에서, 예를 들어 모션 벡터의 제1 서브셋이 제1 임계 크기에서 캡핑되는 대신 폐기될 때, 모션 벡터(112)의 폐기된 제1 서브셋 이외의 모션 벡터의 잔여 서브셋(112)은 블록 714에서 재투영된 프레임에 대한 픽셀 데이터(108(3))를 수정하는 데 사용된다.. 픽셀 데이터(108(3))의 수정은 본원에 설명된 임의의 기술(예를 들어, 프로세스(300)의 블록 308-312를 참조하여 설명된 기술)을 포함할 수 있다.

[0064] 716에서, 로직은 재투영된 프레임(104(3))에 대한 수정된 픽셀 데이터(108(3))에 적어도 부분적으로 기초하여 디스플레이(예를 들어, HMD(100)의 디스플레이 패널(들))에 재투영된 프레임(104(3))을 렌더링할 수 있다. 블록 716에서 수행된 동작(들)은 프로세스(300)의 블록 314와 관련하여 설명된 것과 유사할 수 있다.

[0065] 도 8은 본원에 개시된 실시예에 따른 모션 스무딩 기술의 일부로서 재투영된 프레임을 생성하기 위해 사용되기 전에 GPU로부터 출력되는 모션 벡터를 감쇠하기 위한 예시적인 프로세스(800)의 흐름도이다. 논의를 위해, 프로세스(800)는 이전 도면을 참조하여 설명된다.

[0066] 802에서, 디스플레이 시스템의 로직(예를 들어, HMD(100)의 로직)은 GPU(110)에 대한 입력으로서 이전에 렌더링된 프레임(104)과 연관된 픽셀 데이터(108)를 제공할 수 있다. 블록 802에서 수행된 동작(들)은 프로세스(300)의 블록 302와 관련하여 설명된 것과 유사할 수 있다.

[0067] 804에서, 로직은 모션 벡터(112)의 어레이를 GPU(110)로부터 수신할 수 있다. 블록 804에서 수행된 동작(들)은 프로세스(300)의 블록 304와 관련하여 설명된 것과 유사할 수 있다.

[0068] 806에서, 로직은 (예를 들어, 해당 모션 벡터(112)의 품질에 대한 신뢰도가 낮은 경우) 모션 벡터(112) 중 개별적인 것의 크기를 감쇠(예를 들어, 축소, 감소 등)할 목적으로 모션 벡터(112)의 어레이에 오버레이된 감쇠 텍스처를 생성할 수 있다. 감쇠 텍스처는 복수의 텍셀(예를 들어, 텍셀의 그리드)을 포함하도록 임의의 적절한 해상도에서 생성될 수 있다. 감쇠 텍스처의 해상도는 다수의 모션 벡터가 감쇠 텍스처의 개별 텍셀 내에 있도록 모션 벡터(112)의 어레이의 해상도보다 낮은 해상도일 수 있다. 일 예에서, 감쇠 텍스처는 사용자의 시야(FOV) 중앙에서의 텍셀이 약 6도 x 6도(수평 및 수직)가 되는 해상도를 가질 수 있다. 텍셀당 해상도(수평 및 수직)는 텍셀당 도수가 감쇠 텍스처에 걸쳐 일정하지 않을 수 있기 때문에 대략적일 수 있다. 예를 들어, HMD(100)의 FOV의 영역은 대략 100도 x 100도(예를 들어, 플러스 또는 마이너스 약 10도)에 걸쳐 있을 수 있다. 감쇠 텍스처는 비선형 투영 매트릭스에 기초하여 생성될 수 있으므로, 해상도는 감쇠 텍스처의 텍셀마다 가변되지만 해상도는 대략 6-10도 x 6-10도인 텍셀을 제공한다. 또한, 감쇠 텍스처의 각 텍셀에는 (예를 들어, 0.0 및 1.0의 범위 내에서) 감쇠 값이 할당될 수 있다. 주어진 텍셀에 할당된 감쇠 값은 해당 텍셀에서 모션 벡터(112)의 크기가 감소(축소)되는 양을 제어한다. 예를 들어, 1.0의 감쇠 값은 감쇠가 없는 것에 해당할 수 있으며, 이에 따라 1.0의 감쇠 값을 모션 벡터(112)에 적용하는 것은 모션 벡터(112)가 크기 측면에서 있는 그대로(예를 들어, 감소되지 않고, 축소되지 않는 등) 남아 있음을 의미한다. 그러나, 0.0의 감쇠 값은 전체 감쇠에 해당할 수 있으며, 이에 따라 0.0의 감쇠 값을 모션 벡터(112)에 적용하는 것은 모션 벡터(112)가 크기 측면에서 제로로 감소됨을 의미한다. 0.0과 1.0 사이의 감쇠 값은 부분 감쇠에 해당할 수 있으며, 이에 따라 0.5의 감쇠 값을 모션

벡터(112)에 적용하는 것은 모션 벡터가 원래 길이(크기)의 50%(절반)로 감소됨을 의미한다. 따라서, 감쇠 텍스처는 모션 벡터(112)의 크기를 감소(축소)시킬 수 있지만, 모션 벡터(112)의 크기를 증가(늘릴)시킬 수는 없다.

[0069] 도 8은 2개의 연속적으로 렌더링된 프레임(104) 사이의 화면의 하단에서의 휘도(또는 색상)의 급격한 변화에 기초하여 블록 808에서 생성될 수 있는 예시적인 감쇠 텍스처(807)를 도시하고 있다. 예를 들어, 비디오 게임을 플레이하는 동안, HMD(100)의 사용자는 다음 렌더링된 프레임(104)에서 화면의 하단 절반에 큰 청색 메쉬가 나타나게 하는 핸드헬드 제어기 상의 버튼을 선택할 수 있다. 이러한 예시적인 시나리오에서, GPU(110)는, 이러한 프레임(104) 및 이전에 렌더링된 프레임(104)을 입력으로서 처리할 때, 화면의 하단 절반에 있는 큰 청색 메쉬에서의 픽셀의 휘도 값과 최상으로 정합하는 휘도 값으로 장면의 상단에 렌더링되는 스카이에서 픽셀을 찾을 수 있다. 이러한 결과로, GPU(110)는 갑자기 크기가 매우 크고 장면에서 실제 모션을 나타내지 않는 장면의 하단에 모션 벡터(112)를 갖는 모션 벡터 필드(112)를 출력한다. 예시적인 감쇠 텍스처(807)에서, 백색 텍셀은 1.0의 감쇠 값을 나타낼 수 있는 반면, 흑색 텍셀은 0.0의 감쇠 값을 나타낼 수 있고 회색 텍셀의 음영은 배타적으로 0.0과 1.0 사이의 감쇠 값을 나타낼 수 있다.

[0070] 808에서, 로직은 블록 806에서 생성된 감쇠 텍스처를 사용하여 모션 벡터(112)의 어레이에서 개별 모션 벡터의 크기를 감소(예를 들어, 축소)시킬 수 있다(예를 들어, 감쇠 텍스처의 감쇠 값을 사용하여 감쇠 값에 대응하는 양만큼 모션 벡터(112)의 크기를 감소시킴). 이는 감쇠 텍스처에서 1.0 미만의 적어도 하나의 감쇠 값을 가정하여 모션 벡터(112)의 수정된 어레이를 생성한다. 목적은 프레임(104) 사이의 임계량보다 크게 변경된 모션 벡터(112)를 축소하는 것으로, 이는 이러한 모션 벡터(112)가 장면 내의 실제 모션을 나타내지 않으며 모션을 나타내는 것으로 오인되지 않게 나타낼 가능성이 크다. 감쇠 텍스처는 해당 영역의 모션 벡터가 단일 프레임에서(크기 측면에서) 극단적인 화면의 영역을 찾는 데 사용되며, 이는 모션 벡터가 실제 모션을 나타내는 것으로 신뢰할 수 없거나, 또는 장면 내에서 실제 모션을 나타내는 경우에 이러한 모션은 모션 스무딩이 어차피 눈에 띄는 영향을 미치기에는 너무 빠르며 해당 모션 벡터를 사용하여 모션 스무딩을 시도하는 것보다 제로 모션 벡터를 갖는 것이 바람직함을 나타낸다. 현실적으로, 프레임 간 모션이 있을 때마다, 디스플레이의 재생률이 통상적으로 90 Hz 정도이고 프레임 사이의 짧은 시간 내에 무엇이든 매우 멀리 이동하기가 어렵기 때문에 모션은 통상적으로 이전 프레임과 크게 다르지 않다. 따라서, 감쇠 텍스처에 의해 감쇠된 해당 극단적인 모션 벡터(112)는 종종 실제 모션을 나타내지 않는다.

[0071] 서브블록 809에 의해 도시된 바와 같이, 일부 실시예에서, 모션 벡터의 크기를 감소시키기 위해 블록 808에서 감쇠 텍스처를 사용하는 단계는 감쇠 텍스처를 사용하기 전에 감쇠 텍스처에 최소 N x N 필터를 적용하여 모션 벡터의 크기를 감소시키는 단계를 포함할 수 있다. 여기서, N은 임의의 적절한 수일 수 있다. N = 3은 최소 3 x 3 필터(즉, 3 x 3 텍셀 블록)를 사용한다. N = 3의 예를 사용하여, 블록 809에서 감쇠 텍스처에 최소 3 x 3 필터를 적용할 때, 로직은, 감쇠 텍스처(807)로부터 텍셀 기준으로, 해당 텍셀을 중심으로 한 각 3 x 3 텍셀 블록(예를 들어, 해당 텍셀 및 8개의 인접 텍셀을 포함하는 텍셀 블록)에서 9개의 감쇠 값 중 최소 값을 페칭할 수 있다. 블록 809에서 이러한 최소 N x N 필터를 적용하면, 통상적으로 비제로 감쇠 값이 할당되지만 제로의 감쇠 값이 할당된 인접한 텍셀을 갖는 감쇠 텍스처의 임의의 대응하는 텍셀은 해당 대응하는 텍셀의 대응하는 모션 벡터에 제로의 감쇠 값을 적용할 것이기 때문에 더 많은 모션 벡터(112)가 제로화되어, 감쇠 텍스처(807)에서 대응하는 텍셀에 비제로 감쇠 값이 할당됨에도 불구하고 대응하는 텍셀에서의 모션 벡터를 제로로 감소시킨다. 다시 말해서, 감쇠 텍스처의 주어진 텍셀에서 모션 벡터(112)는 인접 텍셀에 제로의 감쇠 값이 할당되면 제로화되고, 이는 인접 텍셀이 수많은 큰 크기의 모션 벡터를 갖는다는 것을 의미한다. 도 8은 감쇠 텍스처(807)에 최소 N x N 필터를 적용한 결과인 예시적인 감쇠 텍스처(811)를 도시하고 있다. 결과적인 감쇠 텍스처(811)는 감쇠 텍스처(807)보다 더 많은 흑색 텍셀을 포함하며, 이는 결과적인 감쇠 텍스처(811)를 사용하여 더 많은 모션 벡터가 제로화됨을 의미한다.

[0072] 810에서, 로직은 감쇠 텍스처에서 1.0 미만의 감쇠 값에 대해 대략 감쇠된 크기의 일부를 갖는 모션 벡터(112)의 수정된 어레이에 기초하여 재투영된 프레임에 대한 픽셀 데이터(108(3))를 수정할 수 있다. 픽셀 데이터(108(3))의 수정은 본원에 설명된 임의의 기술(예를 들어, 프로세스(300)의 블록 308-312를 참조하여 설명된 기술)을 포함할 수 있다.

[0073] 812에서, 로직은 재투영된 프레임(104(3))에 대한 수정된 픽셀 데이터(108(3))에 적어도 부분적으로 기초하여 디스플레이(예를 들어, HMD(100)의 디스플레이 패널(들))에 재투영된 프레임(104(3))을 렌더링할 수 있다. 블록 812에서 수행된 동작(들)은 프로세스(300)의 블록 314와 관련하여 설명된 것과 유사할 수 있다.

[0074] 도 9는 본원에 개시된 실시예에 따른 도 8의 프로세스(800)를 참조하여 설명된 감쇠 텍스처와 같은 감쇠 텍스처

를 생성하기 위한 예시적인 프로세스(900)의 흐름도이다. 따라서, 프로세스(900)는 프로세스(800)의 블록 806에서 감쇠 텍스처를 생성하기 위해 수행될 수 있는 동작의 서브프로세스를 나타낼 수 있다. 논의를 위해, 프로세스(900)는 이전 도면을 참조하여 설명된다.

[0075] 902에서, GPU(110)의 비디오 인코더에 의해 생성된 모션 벡터(112)의 어레이를 수신한 후, 디스플레이 시스템의 로직(예를 들어, HMD(100)의 로직)은 모션 벡터(112)의 다수의 이전에 획득된 어레이에 적어도 부분적으로 기초하여 감쇠 텍스처의 각 텍셀에 대한 델타 벡터의 세트를 결정할 수 있다. 예를 들어, 모션 벡터(112)의 어레이의 이력은 일련의 프레임이 처리됨에 따라 유지될 수 있고, 모션 벡터(112)의 마지막 두 어레이(예를 들어, 모션 벡터(112)의 가장 최근 두 어레이)가 비교되어 블록 902에서 델타 벡터의 세트(예를 들어, 2개의 어레이에서 한 쌍의 대응하는 모션 벡터 사이의 차이(예를 들어, 크기 측면에서)를 나타내는 차이 벡터)를 결정할 수 있다.

[0076] 904에서, 로직은 임계 크기보다 큰 크기를 갖는 감쇠 텍스처의 각 텍셀에서 델타 벡터의 백분율을 계산할 수 있다. 예를 들어, 블록 904에서 로직은 감쇠 텍스처의 주어진 텍셀에서 델타 벡터를 평가하고, 텍셀에서 "이상점" 벡터를 임계 크기보다 큰 델타(예를 들어, 크기)를 갖는 해당 델타 벡터로 식별하고, 텍셀에서 이상점 벡터의 백분율을 계산할 수 있다. 프레임당(또는 11.1 ms 시간당) (사용자의 관점으로부터) 약 3도의 이동/모션의 임계치와 같은 임의의 적절한 임계치는 블록 904에서 "이상점" 벡터를 식별할 목적으로 사용될 수 있다. 다시 말해서, 블록 904에서 사용된 임계치는, 도 7의 프로세스(700)의 블록 707 및 709를 참조하여 전술한 바와 같이, 제1 프레임(104(1))이 렌더링된 제1 시간과 제2 프레임(104(2))이 렌더링된 제2 시간 사이의 기간(또는 간격)에 기초할 수 있다.

[0077] 906에서, 로직은, 감쇠 텍스처의 각 텍셀에 대해, "이상점" 델타 벡터인 텍셀에서 델타 벡터의 백분율이 25%와 같은 임계 백분율을 충족하거나 초과하는지 여부를 결정할 수 있다. "이상점" 델타 벡터인 텍셀에서 델타 벡터의 백분율이 블록 906에서 임계 백분율을 충족하거나 초과하면, 프로세스(900)는 블록 906에서 블록 908까지의 "YES" 경로를 따를 수 있다.

[0078] 908에서, 로직은 임계 백분율을 충족하거나 초과하는 다수의 "이상점" 델타 벡터를 갖는 텍셀에 대해 감쇠 값을 제로로 계산할 수 있다. 즉, 로직은 주어진 텍셀에 대한 감쇠 값을, 해당 텍셀에 대한 델타 벡터의 특정 양(예를 들어, 25% 이상)이 최대 길이보다 큰 경우에, 제로로 계산할 수 있다. 프로세스(800)에 설명된 바와 같이 제로의 감쇠 값을 주어진 텍셀 내의 모션 벡터(112)에 적용하면 이러한 모션 벡터(112)의 크기를 제로로 감소시킨다(예를 들어, 해당 텍셀 내의 모션 벡터를 완전히 감쇠시킴).

[0079] 906에서, "이상점" 델타 벡터의 임계 백분율 미만을 갖는 텍셀의 경우, 프로세스(900)는 블록 906에서 블록 910까지의 "NO" 경로를 따를 수 있으며, 이 경우 로직은 블록 904에서 계산된 백분율에 적어도 부분적으로 기초하는 감쇠 값으로서 해당 텍셀에 대한 감쇠 값을 계산할 수 있다. 예를 들어, 주어진 텍셀에 대한 임계 크기보다 큰 크기를 갖는 델타 벡터의 백분율이 임계 백분율(예를 들어, 25%)보다 작은 경우, 감쇠 값은 임계 백분율보다 작은 값의 범위 내에서(예를 들어, 0%-25%의 범위 내에서) 계산된 백분율에 선형으로 매핑되는 값으로 설정될 수 있다. 블록 910에서 계산된 감쇠 값은 (감쇠 텍스처가 모션 벡터 필드(112)에 적용될 때) 제로화되지 않은 잔여 벡터에서 노이즈를 줄이는 데 효과적이다.

[0080] 912에서, 계산된 감쇠 값은 이전에 렌더링된 프레임(104)에 기초하여 해당 텍셀에 대한 기존 감쇠 값과 비교될 수 있다. 감쇠 값이 블록 912에서 기존 값보다 작으면, 프로세스(900)는 블록 912에서 블록 914까지의 "YES" 경로를 따를 수 있으며, 이 경우 텍셀에 대한 감쇠 값은 블록 908 또는 910에서 계산된 감쇠 값으로 설정된다. 예를 들어, 이전 프레임에서 감쇠 텍스처의 해당 텍셀에 대한 감쇠가 없는 경우, 기존 감쇠 값은 1.0과 같을 수 있고, 블록 908 또는 910에서 계산된 새로운 감쇠 값이 1.0 미만인 경우, 이러한 예에서 감쇠 값은 해당하는 새롭게 계산된 감쇠 값으로 설정되어 주어진 텍셀에서 모션 벡터를 감쇠시킨다.

[0081] 912에서, 계산된 감쇠 값이 해당 텍셀에 대한 기존 값보다 작지 않은 경우, 프로세스(900)는 블록 912에서 블록 916까지의 "NO" 경로를 따를 수 있으며, 이 경우 기존 감쇠 값은 감쇠 없음에 대응하는 최대 감쇠 값(예를 들어, 1.0)을 향해 시간이 지남에 따라 증가된다. 예를 들어, 일부 상황에서 프레임 간 변경되는 불규칙한 모션 벡터가 발생할 수 있으므로, 새로운 감쇠 값은 감쇠 텍스처의 텍셀에 대한 이전(구) 감쇠 값을 초과하여 "블렌딩"될 수 있으며, 이는 계산된 감쇠 값이 해당 기간 동안 텍셀의 기존 감쇠 값보다 작지 않다고 가정하면 감쇠 값이 일정 기간 동안(예를 들어, 1초 동안) 점진적으로 증가하여 감쇠가 없음을 의미한다.

[0082] 도 10은 본원에 개시된 실시예에 따른 모션 스무딩 기술의 일부로서 재투영된 프레임을 생성하기 위해 모션 벡터의 결과 세트가 사용되기 전에 색상 변화가 거의 또는 전혀 없는 영역에 대응하는 모션 벡터를 제로화하기 위

한 예시적인 프로세스(1000)의 흐름도이다. 논의를 위해, 프로세스(1000)는 이전 도면을 참조하여 설명된다.

[0083] 1002에서, 디스플레이 시스템의 로직(예를 들어, HMD(100)의 로직)은 GPU(110)에 대한 입력으로서 이전에 렌더링된 프레임(104)과 연관된 픽셀 데이터(108)를 제공할 수 있다. 블록 1002에서 수행된 동작(들)은 프로세스(300)의 블록 302와 관련하여 설명된 것과 유사할 수 있다.

[0084] 1004에서, 로직은 모션 벡터(112)의 어레이를 GPU(110)로부터 수신할 수 있다. 블록 1004에서 수행된 동작(들)은 프로세스(300)의 블록 304와 관련하여 설명된 것과 유사할 수 있다.

[0085] 1006에서, 로직은 GPU(110)에 입력된 다수의 이전에 렌더링된 프레임(104) 사이의 (RGB 벡터의 측면에서) 컬러 픽셀 값을 비교하여 모션 벡터(112)의 어레이에서 모션 벡터당 색상 차이 값을 결정할 수 있다. 이러한 색상 차이 값은 임계 색상 차이와 비교되어, 해당 색상 차이 값이 임계 색상 차이보다 작기 때문에 프레임 간의 색상 변화가 거의 또는 전혀 없는 것과 연관된 모션 벡터(112)의 서브셋을 결정할 수 있다. 일부 실시예에서, 로직은 RGB 픽셀 값을 벡터로서 취급한다. 그 후, 색상 차이(예를 들어, 델타 벡터)를 결정하기 위해 이전에 렌더링된 프레임(104) 사이의 대응하는 RGB 벡터가 비교된다. 모션 벡터(112)의 어레이에서 각 모션 벡터(112)는 픽셀 블록(예를 들어, 8 x 8 픽셀 블록)에 대응할 수 있기 때문에, 이러한 색상 비교는 프레임(104) 사이의 8 x 8 픽셀 블록에서 RGB 픽셀 값의 일부 또는 전부를 비교함으로써 모션 벡터를 기준으로 수행될 수 있다. 일부 실시예에서, 로직은 프레임(104) 사이의 색상 변화의 양(예를 들어, 색상 차이 값)을 결정하기 위해 단일 모션 벡터(112)에 대응하는 픽셀의 각 8 x 8 픽셀 블록에서 픽셀의 모든 행 및 픽셀의 다른 모든 열에 대한 색상 값을 비교하고, 색상 변화량은 특정 모션 벡터(112)와 연관된다. 따라서, 주어진 모션 벡터(112)에 대응하는 픽셀 블록과 관련하여, 프레임 간의 색상 비교는 프로세싱 리소스를 보존하기 위해 대응하는 픽셀 블록에서 픽셀의 일부(예를 들어, 25%)를 비교하는 단계를 포함할 수 있다.

[0086] 1008에서, 로직은 색상 차이 값이 임계 색상 차이보다 작은 모션 벡터(112)의 서브셋의 크기를 제로의 길이/크기로 감소시킬 수 있다. 이는 이전에 렌더링된 프레임 사이에 색상 변화가 거의 또는 전혀 없는(때로는 제로 모션 색상 차이로 지칭됨) 장면의 영역에서 어떠한 모션 벡터도 포함하지 않는 모션 벡터(112)의 수정된 어레이를 생성한다. 이는, 예를 들어 텍스트가 단색 배경 전방에 있을 때, 에지 주변에서 텍스트가 흐릿하거나 구불구불하게 표시되는 것을 방지하는데 도움을 줄 수 있다. 또한, 장면의 조명이나 기타 사용자 인터페이스 요소가 점차 밝아지거나 어두워질 때마다 시각적으로 산만해지는 아티팩트가 나타나는 것을 방지하는 데 도움을 줄 수 있다.

[0087] 1010에서, 로직은 모션 벡터(112)의 수정된 어레이에 기초하여 재투영된 프레임에 대한 픽셀 데이터(108(3))를 수정할 수 있으며, 이는 이전에 렌더링된 프레임(104) 사이에서 색상이 효과적으로 동일한 영역에서 더 이상 모션 벡터(112)를 포함하지 않는다. 픽셀 데이터(108(3))의 수정은 본원에 설명된 임의의 기술(예를 들어, 프로세스(300)의 블록 308-312를 참조하여 설명된 기술)을 포함할 수 있다.

[0088] 1012에서, 로직은 재투영된 프레임(104(3))에 대한 수정된 픽셀 데이터(108(3))에 적어도 부분적으로 기초하여 디스플레이(예를 들어, HMD(100)의 디스플레이 패널(들))에 재투영된 프레임(104(3))을 렌더링할 수 있다. 블록 1012에서 수행된 동작(들)은 프로세스(300)의 블록 314와 관련하여 설명된 것과 유사할 수 있다.

[0089] 도 11은 본원에 개시된 실시예에 따른 모션 스무딩 기술의 일부로서 재투영된 프레임을 생성하기 위해 모션 벡터의 결과 세트가 사용되기 전에 하나 이상의 필터를 사용하여 모션 벡터 필드를 "클린업"하기 위한 예시적인 프로세스(1100)의 흐름도이다. 논의를 위해, 프로세스(1100)는 이전 도면을 참조하여 설명된다.

[0090] 1102에서, 디스플레이 시스템의 로직(예를 들어, HMD(100)의 로직)은 GPU(110)에 대한 입력으로서 이전에 렌더링된 프레임(104)과 연관된 픽셀 데이터(108)를 제공할 수 있다. 블록 1102에서 수행된 동작(들)은 프로세스(300)의 블록 302와 관련하여 설명된 것과 유사할 수 있다.

[0091] 1104에서, 로직은 모션 벡터(112)의 어레이를 GPU(110)로부터 수신할 수 있다. 블록 1104에서 수행된 동작(들)은 프로세스(300)의 블록 304와 관련하여 설명된 것과 유사할 수 있다.

[0092] 1106에서, 로직은 모션 벡터(112)의 어레이에 N x N 스칼라 중간값 필터를 적용하여 모션 벡터(112)의 제1 수정된 어레이를 획득할 수 있다. 여기서 N은 5 x 5 스칼라 중간값 필터를 적용할 수 있는 N = 5와 같은 임의의 적절한 수일 수 있다. N = 5인 예에서, 5 x 5 스칼라 중간값 필터는 선택된 모션 벡터를 둘러싼 5 x 5 세트의 모션 벡터(112)를 살펴봄으로써 모션 벡터(112)의 어레이의 각 모션 벡터에 적용된다. 선택된 모션 벡터의 경우, 선택된 벡터를 둘러싼 5 x 5 영역에는 25개의 모션 벡터가 있다. 선택된 모션 벡터에 적용된 스칼라 중간값 필터는 (25개의 x 성분 값 중에서) 중간 x 성분 값 및 (25개의 y 성분 값 중에서) 중간 y 성분 값을 계산한 후,

중간 x 성분 값과 중간 y 성분 값을 조합하여 스칼라 중간값 벡터를 획득하고, 선택된 모션 벡터는 스칼라 중간값 벡터로 대체될 수 있다. 이러한 프로세스는 모션 벡터(112)의 어레이의 각 벡터에 대해 반복된다. 다시, N에 대한 임의의 적절한 값을 사용할 수 있으며, N = 5는 단지 예일뿐이다. 블록 1106에서 스칼라 중간값 필터를 적용하면 모션 벡터 필드(112)에서 노이즈를 감소시키고 이상점 모션 벡터의 제거를 돕는다. 블록 1106에서 스칼라 중간값 필터를 사용하면 또한 더욱 계산 집약적인 더 복잡한 거리 알고리즘에 의존하는 벡터 중간값 필터와 같은 더 복잡한 필터를 적용하는 것보다 계산적으로 더 빠르다.

[0093] 1108에서, 로직은 모션 벡터(112)의 제1 수정된 어레이에 M x M 평균-최대 블러 필터를 적용하여 모션 벡터(112)의 제2 수정된 어레이를 획득할 수 있다. 여기서, M은 3 x 3 평균-최대 블러 필터를 적용할 수 있는 M = 3과 같은 임의의 적절한 수일 수 있다. 평균-최대 블러 필터는 2개의 개별 필터(평균 벡터 필터 및 최대 길이 벡터 필터)를 독립적으로 적용한 후, 해당하는 2개의 독립적으로 적용된 필터에서 결과적인 벡터의 벡터 평균을 가져온다. M = 3을 사용하는 예에서, 3 x 3 평균-최대 블러 필터는 선택된 모션 벡터를 둘러싼 3 x 3 세트의 모션 벡터(112)를 살펴봄으로써 모션 벡터(112)의 어레이의 각 모션 벡터에 적용된다. 선택된 모션 벡터의 경우, 선택된 벡터를 둘러싼 3 x 3 영역에는 9개의 모션 벡터가 있다. 먼저, 평균 벡터 필터가 선택한 모션 벡터에 적용되어, (9개의 x 성분 값 중에서) 평균 x 성분 값 및 (9개의 y 성분 값 중에서) 평균 y 성분 값을 계산한 후에 평균 x 성분 값과 평균 y 성분 값을 조합하여 평균 벡터를 획득한다. 이어서, 최대 길이 벡터 필터가 선택된 모션 벡터에 독립적으로 적용되어, (9개의 모션 벡터 중에서) 가장 긴 벡터를 결정한다. 그 후, 평균 벡터와 가장 긴 벡터의 벡터 평균을 취하여 결과적인 벡터를 결정하고, 선택된 모션 벡터를 결과적인 벡터로 대체할 수 있다. 이러한 프로세스는 모션 벡터(112)의 어레이의 각 벡터에 대해 반복된다. 다시, M에 대한 임의의 적절한 값을 사용할 수 있으며, M = 3은 단지 예일뿐이다. 블록 1108에서 평균-최대 블러 필터를 적용하면 모션 벡터 그룹의 서브영역에서 불연속성을 제거하는 스무딩된 모션 벡터 필드를 제공한다. 일부 실시예에서, 블록 1108에서 평균-최대 블러 필터를 적용하는 단계는 본원에 설명된 바와 같이 모션 벡터 필드(112)가 모션 스무딩을 위해 사용되기 전에 모션 벡터 필드(112)에 대한 최종 "클린업" 단계이다. 예를 들어, 모션 벡터의 임계화, 감쇠 텍스처의 사용, 색상 차이가 거의 또는 전혀 없는 영역에서 벡터의 제로화 등과 같이 본원에 설명된 다른 기술과 조합하여 사용되는 경우, 이러한 다양한 동작은 평균-최대 블러 필터를 적용하기 이전에 수행될 수 있다.

[0094] 1110에서, 로직은 필터링된 모션 벡터를 갖는 모션 벡터(112)의 수정된 어레이에 기초하여 재투영된 프레임에 대한 픽셀 데이터(108(3))를 수정할 수 있다. 픽셀 데이터(108(3))의 수정은 본원에 설명된 임의의 기술(예를 들어, 프로세스(300)의 블록 308-312를 참조하여 설명된 기술)을 포함할 수 있다.

[0095] 1112에서, 로직은 재투영된 프레임(104(3))에 대한 수정된 픽셀 데이터(108(3))에 적어도 부분적으로 기초하여 디스플레이(예를 들어, HMD(100)의 디스플레이 패널(들))에 재투영된 프레임(104(3))을 렌더링할 수 있다. 블록 1112에서 수행된 동작(들)은 프로세스(300)의 블록 314와 관련하여 설명된 것과 유사할 수 있다.

[0096] 도 7-11의 프로세스에서 설명된 다양한 기술(GPU(110)에 의해 출력되는 모션 벡터 필드를 수정하기 위한 다양한 기술)이 임의의 조합 및/또는 임의의 순서로 사용될 수 있음을 이해해야 한다. 예를 들어, 도 7을 참조하여 설명된 기술은 도 8 및 9를 참조하여 설명된 기술과 조합하여 사용될 수 있고, 해당 기술의 일부 또는 전부는 도 10을 참조하여 설명된 기술과 조합하여 사용될 수 있고, 해당 기술의 일부 또는 전부는 도 11을 참조하여 설명된 기술과 조합하여 사용될 수 있다. 즉, 최적화된 모션 벡터 필드를 획득하는 강력한 기술은 모션 벡터 필드의 강력한 수정에서 함께 도 7-11을 참조하여 설명된 기술의 전부를 이용할 수 있다.

[0097] 도 12는 본원에 개시된 실시예에 따른 픽셀 데이터가 모션 벡터 추정을 위해 GPU에 입력되기 전에 이전에 렌더링된 프레임을 회전시키기 위한 예시적인 프로세스(1200)의 흐름도이다. 논의를 위해, 프로세스(1200)는 이전 도면을 참조하여 설명된다.

[0098] 1202에서, 디스플레이 시스템의 로직(예를 들어, HMD(100)의 로직)은 픽셀 데이터(108)가 GPU(110)에 입력으로서 제공될 다수의 이전에 렌더링된 프레임(104)의 제1 프레임(104(1))을 제1 회전된 프레임(1204(1))을 획득하기 위한 회전량만큼 회전시킬 수 있다.

[0099] 1206에서, 로직은 다수의 이전에 렌더링된 프레임(104) 중 제2 프레임(104(2))을 제2 회전된 프레임(1204(2))을 획득하기 위한 회전량만큼 회전시킬 수 있다.

[0100] 1208에서, 로직은 GPU(110)에 대한 입력으로서 회전된 프레임(1204(1) 및 1204(2))의 픽셀 데이터를 제공할 수 있다. 프로세스(1200)는 블록 1208로부터 프로세스(300)의 블록 304로 계속될 수 있다(오프 페이지 참조 "A"로 도시됨). 따라서, 모션 벡터(112)의 어레이는 블록 1208에서 GPU(110)에 대한 입력으로서 제공되는 픽셀 데이터

에 기초하여 블록 304에서 GPU(110)로부터 수신될 수 있고, 도 3의 모션 스무딩 알고리즘의 잔여 동작은 모션 스무딩되는 재투영된 프레임(104(3))을 렌더링하기 위해 수행될 수 있다.

[0101] 프로세스(1200)는 이동하거나 움직이는 객체를 진정으로 대표하지 않는 "이상한" 모션 벡터를 야기하는 장면에서 수평 및/또는 수직 아키텍처 에지의 효과를 완화할 수 있다. 다수의 이전에 렌더링된 프레임(104(1) 및 104(2))은 블록 1202 및 1206에서 동일한 회전량만큼 회전될 수 있거나, 또는 각 프레임(104(1) 및 104(2))은 블록 1202 및 1206에서 각각 상이한 회전량만큼 회전될 수 있다. 일부 실시예에서, 블록 1202 및 1206에서 각 프레임의 회전량은 미리 정의된다(예를 들어, 입력 프레임(104(1) 및 104(2))을 45도로 회전). 프레임(104(1) 및 104(2))을 45도 회전하면 장면에서 수평 및 수직 에지가 대각선 방향이 될 수 있으며, 그 결과 이러한 대각선 에지에 의해 GPU(110)의 비디오 인코더에 의해 생성되는 "이상한" 모션 벡터만큼 많이 발생하지 않을 수 있다. 일부 실시예에서, 블록 1202 및 1206에서 각 프레임의 회전량은 각 프레임(104)에 대한 임의의 회전량일 수 있다. 이는 45도로 정렬될 수 있는 임의의 에지를 일시적으로 은닉할 수 있다.

[0102] 도 13은 본원에 개시된 실시예에 따른 이전에 렌더링된 프레임의 크로마 데이터 및 루마 데이터에 기초하여 생성되는 모션 벡터의 어레이 사이에서 선택하기 위한 예시적인 프로세스(1300)의 흐름도이다. 논의를 위해, 프로세스(1300)는 이전 도면을 참조하여 설명된다. 또한, 도 3 및 13에서 오프 페이지 참조 "B"로 도시된 바와 같이, 프로세스(1300)는 도 3의 블록 306에서의 동작 이전에 수행되는 동작을 나타낼 수 있고, 프로세스(1300)는 일부 실시예에서 블록 306-314의 동작을 계속할 수 있다.

[0103] 1302에서, 디스플레이 시스템의 로직(예를 들어, HMD(100)의 로직)은 GPU(110)에 대한 입력으로서 이전에 렌더링된 프레임(104)과 연관된 루마 데이터를 제공할 수 있다. 예를 들어, 제1 프레임(104(1))과 연관된 제1 픽셀 데이터(108(1))는 제1 루마 데이터, 제1 크로마 데이터, 및/또는 추가 유형의 데이터를 포함할 수 있다. 유사하게, 제2 프레임(104(2))과 연관된 제2 픽셀 데이터(108(2))는 제2 루마 데이터, 제2 크로마 데이터 등과 같은 유사한 유형의 데이터를 포함할 수 있다. 따라서, 제1 루마 데이터 및 제2 루마 데이터는 블록 1302에서 GPU(110)에 입력으로서 제공될 수 있다.

[0104] 1304에서, 로직은, GPU(110)로부터, 제1 루마 데이터 및 제2 루마 데이터에 적어도 부분적으로 기초하여 GPU(110)의 비디오 인코더에 의해 생성된 모션 벡터(112)의 제1 어레이를 수신할 수 있다.

[0105] 1306에서, 로직은 제1 프레임(104(1))과 연관된 제1 크로마 데이터 및 제2 프레임(104(2))과 연관된 제2 크로마 데이터를 GPU(110)에 대한 입력으로서 제공할 수 있다.

[0106] 1308에서, 로직은, GPU(110)로부터, 제1 크로마 데이터 및 제2 크로마 데이터에 적어도 부분적으로 기초하여 GPU(110)의 비디오 인코더에 의해 생성된 모션 벡터(112)의 제2 어레이를 수신할 수 있다.

[0107] 1310에서, 로직은 모션 벡터(112)의 제1 어레이 또는 모션 벡터(112)의 제2 어레이 중 하나를 모션 벡터(112)의 제1 어레이 및 모션 벡터(112)의 제2 어레이의 분석에 기초하여 모션 벡터의 선택된 어레이로서 선택할 수 있다. 예를 들어, 선택 알고리즘은 모션 벡터(112)의 각 필드에서 모션 벡터(112)의 크기 및/또는 방향에 기초하여 예를 들어 가장 불규칙한 것을 선택할 수 있다. 예를 들어, 밝은 섬광(예를 들어, 가상 폭발)이 제2 프레임(104(2))에서 발생할 때, 루마 데이터는 GPU(110)가 일부 임계 크기를 초과하는 크기를 갖는 모션 벡터(112)를 출력하게 할 수 있는 반면, 동일한 이전에 렌더링된 프레임(104)에 대한 크로마 데이터는 GPU(110)가 극단적인 큰 크기를 갖는 그러한 모션 벡터를 포함하지 않는 모션 벡터(112)를 출력하게 할 수 있다. 이 경우, 크로마 데이터로부터 생성된 모션 벡터(112)의 어레이는 블록 1310에서 선택될 수 있다. 이는 선택 알고리즘의 일 예일 뿐이며 다른 것이 이용될 수 있다. 전술한 바와 같이, 프로세스(1300)는 블록 1310으로부터 프로세스(300)의 블록 306으로 계속될 수 있으며, 이 경우 블록 1310으로부터의 모션 벡터(112)의 선택된 어레이에 기초하여 재투영된 프레임(104(3))에 대한 픽셀 데이터(108(3))가 수정된다.

[0108] 도 14는 본원에 개시된 실시예에 따른 모션 벡터의 다수 어레이를 획득하고, 어레이 간의 차이를 결정하고, 모션 스무딩을 위해 결정된 차이에 기초하여 모션 벡터의 최종 어레이를 생성하기 위한 예시적인 프로세스(1400)의 흐름도이다. 논의를 위해, 프로세스(1400)는 이전 도면을 참조하여 설명된다. 또한, 도 3 및 14에서 오프 페이지 참조 "B"로 도시된 바와 같이, 프로세스(1400)는 도 3의 블록 306에서의 동작 이전에 수행되는 동작을 나타낼 수 있고, 프로세스(1400)는 일부 실시예에서 블록 306-314의 동작을 계속할 수 있다.

[0109] 1402에서, 디스플레이 시스템의 로직(예를 들어, HMD(100)의 로직)은 GPU(110)에 대한 입력으로서 이전에 렌더링된 프레임(104)과 연관된 제1 픽셀 데이터(108(1))를 제공할 수 있다. 블록 1402에서 수행된 동작(들)은 프로세스(300)의 블록 302와 관련하여 설명된 것과 유사할 수 있다.

[0110] 1404에서, 로직은 블록 1402에서 입력으로서 제공된 제1 픽셀 데이터(108(1))에 기초하여 GPU(110)로부터 모션 벡터의 제1 어레이(112(1))를 수신할 수 있다.

[0111] 1406에서, 로직은 GPU(110)에 대한 입력으로서 이전에 렌더링된 프레임(104)과 연관된 제2 픽셀 데이터(108)를 제공할 수 있다.

[0112] 1408에서, 로직은, GPU(110)로부터, 제2 픽셀 데이터(108(2))에 적어도 부분적으로 기초하여 모션 벡터의 제2 어레이(112(2))를 수신할 수 있다.

[0113] 1410에서, 로직은 제1 어레이(112(1))와 제2 어레이(112(2)) 사이의 모션 벡터(112)의 방향 및/또는 크기의 차이를 결정할 수 있다. 예를 들어, 제1 어레이(112(1))의 모션 벡터(112)와 제2 어레이(112(2))의 대응하는 모션 벡터(112) 사이를 비교하여, 모션 벡터가 모션 벡터의 방향 또는 크기 중 어느 하나 또는 둘 모두의 측면에서 상이한 경우에, 그렇다면 서로 다른 양을 결정할 수 있다.

[0114] 1412에서, 로직은 블록 1410에서 결정된 바와 같이 모션 벡터의 제1 어레이(112(1)) 또는 모션 벡터의 제2 어레이(112(2)) 사이의 차이에 적어도 부분적으로 기초하여 모션 벡터(112)의 최종 어레이를 생성할 수 있다. 예를 들어, 모션 벡터(112)의 최종 어레이는 제1 어레이(112(1)) 또는 제2 어레이(112(2))의 개별 모션 벡터를 감쇠시키거나 또는 각 어레이(112(1) 및 112(2))에서 대응하는 모션 벡터의 평균을 나타내는 평균 모션 벡터를 감쇠시켜 생성되는 모션 벡터의 세트를 나타낼 수 있다. 전술한 바와 같이, 프로세스(1400)는 블록 1412로부터 프로세스(300)의 블록 306으로 계속될 수 있으며, 이 경우 블록 1412에서 생성된 모션 벡터(112)의 최종 어레이에 기초하여 재투영된 프레임(104(3))에 대한 픽셀 데이터(108(3))가 수정된다.

[0115] 프로세스(1400)가 사용될 수 있는 방법에 대한 예시적인 예로서, 모션 벡터의 제1 어레이(112(1))가 원래("y-up") 방향으로 입력 프레임(104(1) 및 104(2))의 픽셀 데이터(108)에 기초하여 블록 1404에서 수신되고, 그리고 모션 벡터의 제2 어레이(112(2))가 45도 회전된 입력 프레임(104(1) 및 104(2))의 픽셀 데이터(108)에 기초하여 블록 1408에서 수신되는 예를 고려한다. 입력 이미지 프레임(104)을 회전시키는 이러한 개념은 도 10을 참조하여 설명되었다. 따라서, 이러한 예시적인 예에서, 제1 프레임(104(1)) 및 제2 프레임(104(2))에 대한 픽셀 데이터(108)는 블록 1402에서 입력으로서 제공되어 모션 벡터의 제1 어레이(112(1))를 생성할 수 있고, 제1 회전된 프레임(1004(1)) 및 제2 회전된 프레임(1004(2))에 대한 픽셀 데이터(108)는 블록 1406에서 입력으로서 제공되어 모션 벡터의 제2 어레이(112(2))를 생성할 수 있다. 이러한 예시적인 예에서, 모션 벡터의 두 어레이(112(1) 및 112(2))는 (예를 들어, 제2 어레이(112(2))를 역방향으로 45도 회전시킴으로써) 재정렬된 후에 비교되어 양자의 어레이(112(1) 및 112(2))에서 대응하는 벡터 간의 각도 차이(방향) 및/또는 길이 차이(크기)를 살펴봄으로써 불일치를 결정할 수 있다. (예를 들어, 장면에서 객체를 이동하거나 움직이는) 실제 모션은 회전된 입력 프레임(104)에 대해 모션 벡터의 매우 유사한 어레이를 생성할 수 있는 반면, 부정확하게 검출된 모션은 모션 벡터의 매우 상이한 어레이를 생성할 수 있는 경우일 수 있다. 이러한 예시적인 예에서, 모션 벡터(112)의 최종 어레이는 2개의 정규화된 벡터의 내적에 의해 각 어레이(112(1) 및 112(2))에서 2개의 대응하는 벡터의 평균 길이를 감쇠함으로써 블록 1412에서 생성될 수 있다. 일부 실시예에서, 도 8 및 9를 참조하여 설명된 것과 같은 감쇠 텍스처는 최종 어레이를 생성하기 위해 블록 1412에서 이용될 수 있다. 예를 들어, 프로세스(900)의 블록 902에서 결정된 델타 벡터의 세트는 블록 1404에서 수신된 모션 벡터의 제1 어레이(112(1))와 블록 1408에서 수신된 모션 벡터의 제2 어레이(112(2)) 사이의 델타에 기초할 수 있고, 결과적인 감쇠 텍스처는 블록 1412에서 사용되어 제1 어레이(112(1)), 제2 어레이(112(2)), 또는 두 어레이(112(1) 및 112(2)) 사이의 벡터의 조합(예를 들어, 평균)을 감쇠시킬 수 있다.

[0116] 프로세스(1400)가 사용될 수 있는 방법에 대한 또 다른 예시적인 예로서, 다수의 이전에 렌더링된 프레임(104)의 입력 텍스처에 대해 제1 밉 레벨(밉맵 레벨)이 생성되고, 그리고 이러한 제1 밉 레벨에 대한 대응하는 픽셀 데이터(108)가 모션 벡터의 제1 어레이(112(1))를 생성하기 위해 블록 1402에서 GPU(110)에 입력으로서 제공되는 예를 고려한다. 한편, 다수의 이전에 렌더링된 프레임(104)의 입력 텍스처에 대해 제2 밉 레벨이 생성될 수 있고, 이러한 제2 밉 레벨에 대한 대응하는 픽셀 데이터(108)는 모션 벡터의 제2 어레이(112(2))를 생성하기 위해 블록 1406에서 GPU(110)에 입력으로서 제공될 수 있다. 이는 모션 벡터의 대응하는 어레이의 임의의 개수를 생성하기 위해 임의의 수의 밉 레벨에 대해 수행될 수도 있다. 일부 실시예에서, 모션 벡터(112)의 3개 또는 4개의 어레이가 블록 1410 이전에 수신되도록 입력 텍스처의 3개 또는 4개의 밉 레벨이 생성될 수 있다. 일부 실시예에서, 각 밉 레벨은 이전 밉 레벨의 폭 및 높이의 절반(예를 들어, 면적의 25%)이다. 블록 1410에서, 이러한 예시적인 예에서, 상이한 밉 레벨로부터 생성된 어레이 쌍(112) 간의 차이가 결정될 수 있고, 블록 1412에서, 결정된 차이에 기초하여 모션 벡터(112)의 최종 어레이가 생성될 수 있다. 예를 들어, 로직은 상이한

밉 레벨에 대해 모션 벡터(112)의 다수 어레이에 걸쳐 변칙을 검출할 수 있고, 변칙적인 모션 벡터는 감쇠될 수 있다(예를 들어, 제로로 감소). 모션 벡터(112)의 상이한 어레이를 생성하기 위해 입력 프레임(104)의 상이한 밉 레벨을 사용하는 이러한 예는 장면에 반복 패턴(예를 들어, 벽지)의 넓은 영역이 있고, GPU(110)의 비디오 인코더가 이와 달리 이동하거나 움직이는 객체가 없더라도 모션 벡터 해상도와 관련된 반복 패턴의 주파수로 인해 다수의 모션이 있다고 생각하는 여러 큰 크기의 모션 벡터를 생성할 수 있는 상황에 도움이 될 수 있다. 따라서, 프로세스(1400)는, 상이한 밉 레벨에 대한 모션 벡터(112)의 어레이를 생성하는 데 사용될 때, 반복 패턴의 상이한 주파수에 걸쳐 변칙의 검출을 도울 수 있고, 검출될 때 감쇠 텍스처를 사용하여 해당 모션 벡터(112)를 감쇠(예를 들어, 제로화)할 수 있다.

[0117] 프로세스(1400)가 사용될 수 있는 방법에 대한 또 다른 예로서, 다수 어레이(112(1) 및 112(2))가 서로 다른 해상도에서 획득되고, 블록 1410에서 제1 어레이(112(1))의 단일 모션 벡터를 제2 어레이(112(2))의 다수의 대응하는 모션 벡터와(예를 들어, 제2 어레이(112(2))의 다수의 대응하는 모션 벡터의 방향 및/또는 크기 측면에서 평균과) 비교하여 차이가 결정되는 예를 고려한다. 상이한 해상도에서 모션 벡터(112)의 어레이를 사용하면 블록 1412에서 모션 벡터의 최종 어레이를 생성하는 데 유용한 차이를 산출할 수 있다.

[0118] 도 15는 본원에 개시된 실시예에 따른 이미지 영역의 상이한 부분에 대해 상이한 해상도에서 모션 벡터의 다수 어레이를 획득하기 위한 예시적인 프로세스(1500)의 흐름도이다. 논의를 위해, 프로세스(1500)는 이전 도면을 참조하여 설명된다. 또한, 도 3 및 15에서 오프 페이지 참조 "B"로 도시된 바와 같이, 프로세스(1500)는 도 3의 블록 306에서의 동작 이전에 수행되는 동작을 나타낼 수 있고, 프로세스(1500)는 일부 실시예에서 블록 306-314의 동작을 계속할 수 있다.

[0119] 1502에서, 디스플레이 시스템의 로직(예를 들어, HMD(100)의 로직)은 GPU(110)에 대한 입력으로서 이전에 렌더링된 프레임(104)과 연관된 픽셀 데이터(108)를 제공할 수 있다. 블록 1502에서 수행된 동작(들)은 프로세스(300)의 블록 302와 관련하여 설명된 것과 유사할 수 있다.

[0120] 1504에서, 로직은 GPU(110)로부터 모션 벡터의 제1 어레이(112(1))를 수신할 수 있으며, 모션 벡터의 제1 어레이(112(1))는 제1 해상도에서 수신되거나 또는 이와 달리 제1 해상도로 업 샘플링 또는 다운 샘플링된다.

[0121] 1506에서, 로직은 모션 벡터의 제1 어레이(112(1))에 적어도 부분적으로 기초하여 디스플레이에 걸쳐 있는 이미지 영역(1505)의 제1 부분(1505(A))에서 모션의 부재를 검출할 수 있고, 모션 벡터의 제1 어레이(112(1))에 적어도 부분적으로 기초하여 이미지 영역(1505)의 제2 부분(1505(B))에서 객체(106)의 모션을 검출할 수 있다. 예를 들어, 비제로 모션 벡터는 이미지 영역(1505)의 우측 절반에서 검출될 수 있고, 이미지 영역(1505)의 좌측 절반은 임의의 비제로 모션 벡터 없이 제로 값의 모션 벡터를 포함할 수 있다.

[0122] 1508에서, 로직은 (제1 프레임(104(1))과 연관된) 제1 픽셀 데이터(108(1))의 제1 부분 및 (제2 프레임(104(2))과 연관된) 제2 픽셀 데이터(108(2))의 제1 부분을 GPU(110)에 대한 입력으로서 제공할 수 있다. 제1 픽셀 데이터(108(1))의 제1 부분 및 제2 픽셀 데이터(108(2))의 제1 부분 각각은 블록 1506에서 모션의 부재가 검출된 이미지 영역(1505)의 제1 부분(1505(A))에 대응할 수 있다.

[0123] 1510에서, 로직은, GPU(110)로부터, 제1 픽셀 데이터(108(1))의 제1 부분 및 제2 픽셀 데이터(108(2))의 제1 부분에 적어도 부분적으로 기초하여 GPU(110)의 비디오 인코더에 의해 생성된 모션 벡터의 제2 어레이(112(2))를 수신할 수 있다. 이러한 모션 벡터의 제2 어레이(112(2))는 모션 벡터의 제1 어레이(112(1))의 제1 해상도보다 높은 해상도인 제2 해상도에서 생성될 수 있다. 모션의 부재가 검출된 이미지 영역(1505)의 제1 부분(1505(A))에 대해 더 높은 해상도의 모션 벡터 필드(112(2))를 획득하는 것은 더 높은 해상도에서의 모션 벡터 필드(112(2))가 대규모 이동이 검출되지 않은 이미지 영역(1505)의 제1 부분(1505(A))에서 소규모 이동을 검출하는 데 도움이 될 수 있다는 개념에 기초할 수 있다.

[0124] 1512에서, 로직은 (제1 프레임(104(1))과 연관된) 제1 픽셀 데이터(108(1))의 제2 부분 및 (제2 프레임(104(2))과 연관된) 제2 픽셀 데이터(108(2))의 제2 부분을 GPU(110)에 대한 입력으로서 제공할 수 있다. 제1 픽셀 데이터(108(1))의 제2 부분 및 제2 픽셀 데이터(108(2))의 제2 부분 각각은 블록 1506에서 객체(106)의 모션이 검출된 이미지 영역(1505)의 제2 부분(1505(B))에 대응할 수 있다.

[0125] 1514에서, 로직은, GPU(110)로부터, 제1 픽셀 데이터(108(1))의 제2 부분 및 제2 픽셀 데이터(108(2))의 제2 부분에 적어도 부분적으로 기초하여 GPU(110)의 비디오 인코더에 의해 생성된 모션 벡터의 제3 어레이(112(3))를 수신할 수 있다. 이러한 모션 벡터의 제3 어레이(112(3))는 모션 벡터의 제2 어레이(112(2))의 제2 해상도보다 낮은 해상도인 제3 해상도에서 생성될 수 있다. 모션이 검출된 이미지 영역(1005)의 제2 부분(1005(B))에 대

해 상대적으로 더 낮은 해상도의 모션 벡터 필드(112(3))를 획득하는 것은 더 높은 해상도의 모션 벡터 필드가 대규모 이동이 제1 모션 벡터 필드(112(1))의 제1 해상도에서 이미 검출된 이미지 영역(1505)의 제2 부분(1505(B))에서 이동을 검출할 필요가 없을 수 있다는 개념에 기초할 수 있다.

[0126] 1516에서, 로직은 모션 스무딩을 위해 모션 벡터의 제2 어레이(112(2)) 및 모션 벡터의 제3 어레이(112(2))를 사용할 수 있다. 프로세스(1500)는 블록 1516으로부터 프로세스(300)의 블록 306으로 계속될 수 있으며, 이 경우 블록 1516으로부터의 모션 벡터(112)에 기초하여 재투영된 프레임(104(3))에 대한 픽셀 데이터(108(3))가 수정된다. 일 예에서, 모션 스무딩을 위해 블록 1516에서 모션 벡터의 다수 어레이를 사용하는 단계는 프로세스(1400)의 블록 1412를 참조하여 설명된 동작을 포함할 수 있다.

[0127] 도 16은 본원에 개시된 실시예에 따른 VR 헤드셋과 같은 HMD(1600)의 예시적인 구성요소가 내장될 수 있음을 도시하고 있다. HMD(1600)는 이전 도면에서 참조된 HMD(100)와 동일하거나 유사할 수 있으며, 따라서 도 16에 도시된 HMD(1600)의 구성요소는 HMD(100)에서 구현될 수 있다. HMD(1600)는 사용자(102)에 의해(예를 들어, 사용자(102)의 머리에) 착용될 독립형 장치로서 구현될 수 있다. 일부 실시예에서, HMD(1600)는 사용자(102)가 사용자(102)의 머리 둘레에 맞는 크기로 이루어진 고정 메커니즘(예를 들어, 조정식 밴드)을 사용하여 자신의 머리에 HMD(1600)를 고정하도록 하는 것과 같은 헤드 장착형일 수 있다. 일부 실시예에서, HMD(1600)는 근안 또는 니어-투-아이 디스플레이(들)를 포함하는 가상 현실(VR) 또는 증강 현실(AR) 헤드셋을 포함한다. 이와 같이, "웨어러블 장치", "웨어러블 전자 장치", "VR 헤드셋", "AR 헤드셋" 및 "헤드 마운티드 디스플레이(HMD)"라는 용어는 도 16의 장치(1600)를 지칭하기 위해 본원에서 상호 교환적으로 사용될 수 있다. 그러나, 이러한 유형의 장치는 HMD(1600)의 예일 뿐이며, HMD(1600)는 다양한 다른 폼 팩터로 구현될 수 있음을 이해해야 한다.

[0128] 예시된 구현예에서, HMD(1600)는 하나 이상의 프로세서(1602) 및 메모리(1604)(예를 들어, 컴퓨터 판독 가능 매체(1604))를 포함한다. 일부 구현예에서, 프로세서(들)(1602)는 중앙 처리 유닛(CPU), 그래픽 처리 유닛(들)(GPU)(110), CPU 및 GPU(110) 둘 모두, 마이크로프로세서, 디지털 신호 프로세서 또는 당업계에 알려진 기타 처리 장치 또는 구성요소를 포함할 수 있다. 대안적으로 또는 추가적으로, 본원에 기능적으로 설명된 것은 적어도 부분적으로 하나 이상의 하드웨어 로직 구성요소에 의해 수행될 수 있다. 예를 들어, 비제한적으로, 사용될 수 있는 하드웨어 로직 구성요소의 예시적 유형으로는 필드 프로그래머블 게이트 어레이(FPGA), 주문형 집적 회로(ASIC), 주문형 표준 제품(ASSP), 시스템 온 칩 시스템(SOC), 복잡한 프로그래머블 로직 장치(CPLD) 등을 포함한다. 추가적으로, 프로세서(들)(1602)의 각각은 프로그램 모듈, 프로그램 데이터, 및/또는 하나 이상의 운영 체제를 저장할 수도 있는 자체 로컬 메모리를 가질 수 있다.

[0129] 메모리(1604)는 컴퓨터 판독 가능 명령, 데이터 구조, 프로그램 모듈 또는 기타 데이터와 같은 정보의 저장을 위한 임의의 방법 또는 기술로 구현된 휘발성 및 비휘발성 메모리, 착탈식 및 비착탈식 매체를 포함할 수 있다. 이러한 메모리는 RAM, ROM, EEPROM, 플래시 메모리 또는 기타 메모리 기술, CD-ROM, 디지털 다용도 디스크(DVD) 또는 기타 광학 스토리지, 자기 카세트, 자기 테이프, 자기 디스크 스토리지 또는 기타 자기 저장 장치, RAID 저장 시스템 또는 원하는 정보를 저장하는 데 사용될 수 있고 컴퓨팅 장치에 의해 액세스될 수 있는 임의의 기타 매체를 포함하지만, 이에 제한되는 것은 아니다. 메모리(1604)는 컴퓨터 판독 가능 저장 매체("CRSM")로서 구현될 수 있으며, 이는 메모리(1602)에 저장된 명령을 실행하기 위해 프로세서(들)(1602)에 의해 액세스 가능한 임의의 이용 가능한 물리적 매체일 수 있다. 하나의 기본 구현예에서, CRSM은 랜덤 액세스 메모리("RAM") 및 플래시 메모리를 포함할 수 있다. 다른 구현예에서, CRSM은 리드 온리 메모리("ROM"), 전기적으로 소거 가능한 프로그래머블 리드 온리 메모리("EEPROM") 또는 원하는 정보를 저장하는 데 사용될 수 있고 프로세서(들)(1602)에 의해 처리될 수 있는 임의의 다른 유형의 매체를 포함할 수 있지만, 이에 제한되는 것은 아니다.

[0130] 일반적으로, HMD(1600)는 본원에 설명된 기술, 기능 및/또는 동작을 구현하도록 구성되는 로직(예를 들어, 소프트웨어, 하드웨어 및/또는 펌웨어 등)을 포함할 수 있다. 컴퓨터 판독 가능 매체(1604)는 본원에 설명된 기술, 기능 및/또는 동작을 수행하기 위해 프로세서(들)(1602)에서 실행하도록 구성될 수 있는 명령, 데이터 저장소 등과 같은 다양한 모듈을 포함하는 것으로 도시되어 있다. 몇 가지 예시적인 기능 모듈이 컴퓨터 판독 가능 매체(1604)에 저장되고 프로세서(들)(1602)에서 실행 가능한 것으로 도시되어 있지만, 동일한 기능이 하드웨어, 펌웨어에서 또는 시스템 온 칩(SOC) 및/또는 기타 로직으로서 대안적으로 구현될 수 있다.

[0131] 운영 체제 모듈(1606)은 다른 모듈의 이익을 위해 HMD(1600) 내에서 그리고 이에 결합된 하드웨어를 관리하도록 구성될 수 있다. 추가로, 일부 경우에, HMD(1600)는 메모리(1604)에 저장되거나 그렇지 않으면 HMD(1600)에 액세스할 수 있는 하나 이상의 애플리케이션(1608)을 포함할 수 있다. 이러한 구현예에서, 애플리케이션(들)(1608)은 게임 애플리케이션(1610)을 포함한다. 그러나, HMD(1600)는 임의 개수 또는 유형의 애플리케이션

을 포함할 수 있으며, 본원에 도시된 특정 예에 제한되지 않는다. 게임 애플리케이션(1610)은 사용자(102)에 의해 플레이 가능한 비디오 기반 대화형 게임(예를 들어, VR 게임)의 게임플레이를 시작하고, HMD(1600)의 디스플레이 패널에 렌더링될 프레임(예를 들어, 실제 프레임(104))을 출력하도록 구성될 수 있다. HMD(1600)의 다른 로직과 조합된 컴포지터(1611)는 모션 스무딩된 재투영된 프레임(104)을 렌더링하기 위해 본원에 설명된 모션 스무딩 기술을 수행하도록 구성될 수 있다.

[0132] 일반적으로, HMD(1600)는 입력 장치(1612) 및 출력 장치(1614)를 갖는다. 입력 장치(1612)는 제어 버튼을 포함할 수 있다. 일부 구현예에서, 하나 이상의 마이크는 사용자 음성 입력과 같은 오디오 입력을 수신하기 위한 입력 장치(1612)로서 기능할 수 있다. 일부 구현예에서, 하나 이상의 카메라 또는 다른 유형의 센서(예를 들어, 관성 측정 유닛(IMU))은 사용자(102)의 핸드 및/또는 헤드 모션과 같은 제스처 입력을 수신하는 입력 장치(1612)로서 기능할 수 있다. 일부 실시예에서, 추가 입력 장치(1612)는 키보드, 키패드, 마우스, 터치 스크린, 조이스틱 등의 형태로 제공될 수 있다. 다른 실시예에서, HMD(1600)는 키보드, 키패드 또는 기타 유사한 형태의 기계적 입력을 생략할 수 있다. 대신, HMD(1600)는 입력 장치(1612), 네트워크 인터페이스(무선 또는 유선 기반), 전력 및 처리/메모리 기능의 비교적 단순한 형태로 구현될 수 있다. 예를 들어, 하나 이상의 입력 구성 요소의 제한된 세트(예를 들어, 구성을 시작하기 위한 전용 버튼, 전원 온/오프 등)는 이후에 HMD(1600)가 사용될 수 있도록 사용될 수 있다. 일 구현예에서, 입력 장치(들)(1612)는 볼륨 증가/감소를 위한 기본 볼륨 제어 버튼(들)뿐만 아니라 전원 및 리셋 버튼과 같은 제어 메커니즘을 포함할 수 있다.

[0133] 출력 장치(1614)는 디스플레이(들)(1616)를 포함할 수 있으며, 이는 하나 또는 다수의 디스플레이 패널(예를 들어, 스테레오 쌍의 디스플레이 패널)을 포함할 수 있다. 출력 장치(1614)는, 제한 없이, 조명 요소(예를 들어, LED), 햅틱 감각을 생성하기 위한 진동기, 스피커(들)(예를 들어, 헤드폰) 등을 더 포함할 수 있다. 예를 들어, 전원이 켜져 있을 때와 같은 상태를 나타내는 간단한 조명 요소(예를 들어, LED)가 있을 수도 있다.

[0134] HMD(1600)는 네트워크에 대한 무선 연결을 용이하게 하기 위해 안테나(1620)에 연결된 무선 유닛(1618)을 더 포함할 수 있다. 무선 유닛(1618)은 Wi-Fi, 블루투스, 무선 주파수(RF) 등과 같은 다양한 무선 기술 중 하나 이상을 구현할 수 있다. HMD(1600)는 네트워크, 연결된 주변 장치(PC, 게임 콘솔 등을 포함함) 또는 다른 무선 네트워크와 통신하는 플러그-인 네트워크 장치에 대한 유선 연결을 용이하게 하기 위한 물리적 포트를 더 포함할 수 있음을 이해해야 한다.

[0135] HMD(1600)는 하나 이상의 광학 요소를 사용하여 전자 디스플레이(들)(1616)로부터 사용자의 눈(들)으로 광을 보내는 광학 서브시스템(1622)을 더 포함할 수 있다. 광학 서브시스템(1622)은, 제한 없이, 예를 들어 조리개, 렌즈(예를 들어, 프레넬 렌즈, 볼록 렌즈, 오목 렌즈 등), 필터 등을 포함하는 다양한 유형 및 서로 다른 광학 요소의 조합을 포함할 수 있다. 일부 실시예에서, 광학 서브시스템(1622) 내의 하나 이상의 광학 요소는 반사 방지 코팅과 같은 하나 이상의 코팅을 가질 수 있다. 광학 서브시스템(1622)에 의한 이미지 광의 확대는 전자 디스플레이(들)(1616)가 물리적으로 더 작고, 더 가볍고, 더 큰 디스플레이보다 더 적은 전력을 소모하게 한다. 추가로, 이미지 광의 확대는 표시된 콘텐츠(예를 들어, 이미지)의 시야(FOV)를 증가시킬 수 있다. 예를 들어, 표시된 콘텐츠의 FOV는 표시된 콘텐츠가 거의 모든(예를 들어, 120 내지 150 도 대각선), 일부 경우에는 사용자의 FOV 모두를 사용하여 제시되도록 한다. AR 애플리케이션은 더 좁은 FOV(예를 들어, 약 40도 FOV)를 가질 수 있다. 광학 서브시스템(1622)은, 제한 없이, 배럴 왜곡, 핀쿠션 왜곡, 길이방향 색수차, 가로 색수차, 구면 수차, 코마 수차, 필드 곡률, 난시 등과 같은 하나 이상의 광학 오류를 수정하도록 설계될 수 있다. 일부 실시예에서, 디스플레이를 위해 전자 디스플레이(들)(1616)에 제공된 콘텐츠는 사전 왜곡되고, 광학 서브시스템(1622)은 콘텐츠에 기초하여 생성된 전자 디스플레이(들)(1616)로부터 이미지 광을 수신할 때 왜곡을 보정한다.

[0136] HMD(1600)는 모션, 위치 및 방향 데이터를 생성하는 데 사용되는 센서와 같은 하나 이상의 센서(1624)를 더 포함할 수 있다. 이러한 센서(1624)는 자이로스코프, 가속도계, 자력계, 비디오 카메라, 컬러 센서, 또는 기타 모션, 위치 및 방향 센서일 수 있거나 이들을 포함할 수 있다. 센서(1624)는 또한 모션, 위치 및 방향 데이터를 생성하기 위해 카메라 또는 컬러 센서에 의해 외부에서 볼 수 있는 일련의 능동 또는 수동 마커와 같은 센서 서브 부분을 포함할 수 있다. 예를 들어, VR 헤드셋은 외부 카메라로 보거나 광(예를 들어, 적외선 또는 가시광선)에 의해 조명될 때, 모션, 위치 및 방향 데이터를 생성하기 위해 소프트웨어에 의한 해석을 위해 하나 이상의 참조 지점을 제공할 수 있는 반사기 또는 광(예를 들어, 적외선 또는 가시광선)과 같은 여러 마커를 외부에 포함할 수 있다. HMD(1600)는 HMD(1600)의 환경에서 기지국에 의해 투영되거나 브로드캐스트되는 광(예를 들어, 적외선 또는 가시광선)에 민감한 광 센서를 포함할 수 있다.

[0137] 일 예에서, 센서(들)(1624)는 관성 측정 유닛(IMU)(1626)을 포함할 수 있다. IMU(1126)는 가속도계, 자이로스코

프, 자력계 및/또는 모션 검출, IMU(1626)와 관련된 오류 수정에 적합한 기타 센서, 또는 이들의 일부 조합으로부터 수신된 측정 신호를 기반으로 캘리브레이션 데이터를 생성하는 전자 장치일 수 있다. 측정 신호에 기초하여 IMU(1626)와 같은 모션 기반 센서는 HMD(1600)의 초기 위치에 대한 HMD(1600)의 추정 위치를 나타내는 캘리브레이션 데이터를 생성할 수 있다. 예를 들어, 다수의 가속도계는 병진이동 모션(전진/후진, 위/아래, 좌/우)을 측정할 수 있고 다수의 자이로스코프는 회전 모션(예를 들어, 피치, 요 및 롤)을 측정할 수 있다. 예를 들어, IMU(1626)는 측정 신호를 빠르게 샘플링하고 샘플링된 데이터로부터 HMD(1600)의 추정 위치를 계산할 수 있다. 예를 들어, IMU(1626)는 속도 벡터를 추정하기 위해 시간에 따라 가속도계로부터 수신된 측정 신호를 통합할 수 있고, HMD(1600) 상의 기준 지점의 추정 위치를 결정하기 위해 시간에 따라 속도 벡터를 통합할 수 있다. 기준 지점은 HMD(1600)의 위치를 설명하는 데 사용될 수 있는 지점이다. 기준 지점은 일반적으로 공간 내의 지점으로 정의될 수 있지만, 다양한 실시예에서, 기준 지점은 HMD(1600) 내의 지점(예를 들어, IMU(1626)의 중심)로 정의된다. 대안적으로, IMU(1626)는 샘플링된 측정 신호를 캘리브레이션 데이터를 결정하는 외부 콘솔(또는 기타 컴퓨팅 장치)에 제공한다.

[0138] 센서(1624)는 센서 데이터를 고속으로 제공하기 위해 비교적 고주파수에서 작동할 수 있다. 예를 들어, 센서 데이터는 1000 Hz(또는 1 밀리세컨드마다 1개의 센서 판독)의 속도로 생성될 수 있다. 이러한 방식으로, 초당 1,000회 판독이 이루어진다. 센서가 이러한 속도(또는 더 빠른 속도)로 이 정도의 데이터를 생성할 때, 모션을 예측하는 데 사용되는 데이터 세트는 수십 밀리세컨드 정도의 비교적 짧은 시간 기간 동안에도 상당히 크다.

[0139] 언급된 바와 같이, 일부 실시예에서, 센서(1624)는 3D 공간에서 HMD(1600)의 위치 및/또는 방향, 포즈 등을 추적하기 위해 HMD(1600)의 환경에서 기지국에 의해 방출되는 광에 민감한 광 센서를 포함할 수 있다. 위치 및/또는 방향의 계산은 광 펄스의 타이밍 특성 및 센서(1624)에 의해 검출된 광의 존재 또는 부재에 기초할 수 있다.

[0140] HMD(1600)는 아이 트래킹 모듈(1628)을 더 포함할 수 있다. HMD(1600) 내부의 카메라 또는 기타 광학 센서는 사용자 눈의 이미지 정보를 포착할 수 있으며, 아이 트래킹 모듈(1628)은 포착된 정보를 사용하여 비틀림 및 회전(즉, 롤, 피치 및 요)의 크기 및 각 눈에 대한 응시 방향을 포함하는, (예를 들어, 왜곡 조정을 위한) HMD(1600)에 대한 각 눈의 동공 간 거리, 안간 거리, 3차원(3D) 위치를 결정할 수 있다. 일 예에서, 적외선은 HMD(1600) 내에서 방출되어 각 눈으로부터 반사된다. 반사된 광은 HMD(1600)의 카메라에 의해 수신되거나 검출되고 분석되어 각 눈에 의해 반사된 적외선 광의 변화에서 눈의 회전을 추출한다. 사용자(102)의 눈을 추적하는 많은 방법이 아이 트래킹 모듈(1628)에 의해 사용될 수 있다. 따라서, 아이 트래킹 모듈(1628)은 각 눈의 최대 6개의 자유도(즉, 3D 위치, 롤, 피치 및 요)를 추적할 수 있으며, 적어도 추적된 양의 서브셋은 사용자(102)의 두 눈으로부터 결합되어 응시점(즉, 사용자가 보고 있는 가상 장면의 3D 위치 또는 지점)을 추정할 수 있다. 예를 들어, 아이 트래킹 모듈(1628)은 이전 측정으로부터의 정보, 사용자(102) 머리의 위치를 식별하는 측정, 및 전자 디스플레이(들)(1616)에 의해 제시된 장면을 설명하는 3D 정보를 통합할 수 있다. 따라서, 사용자(102) 눈의 위치 및 방향에 대한 정보는 사용자(102)가 보고 있는 HMD(1600)에 의해 제시된 가상 장면 내의 응시점을 결정하는 데 사용된다.

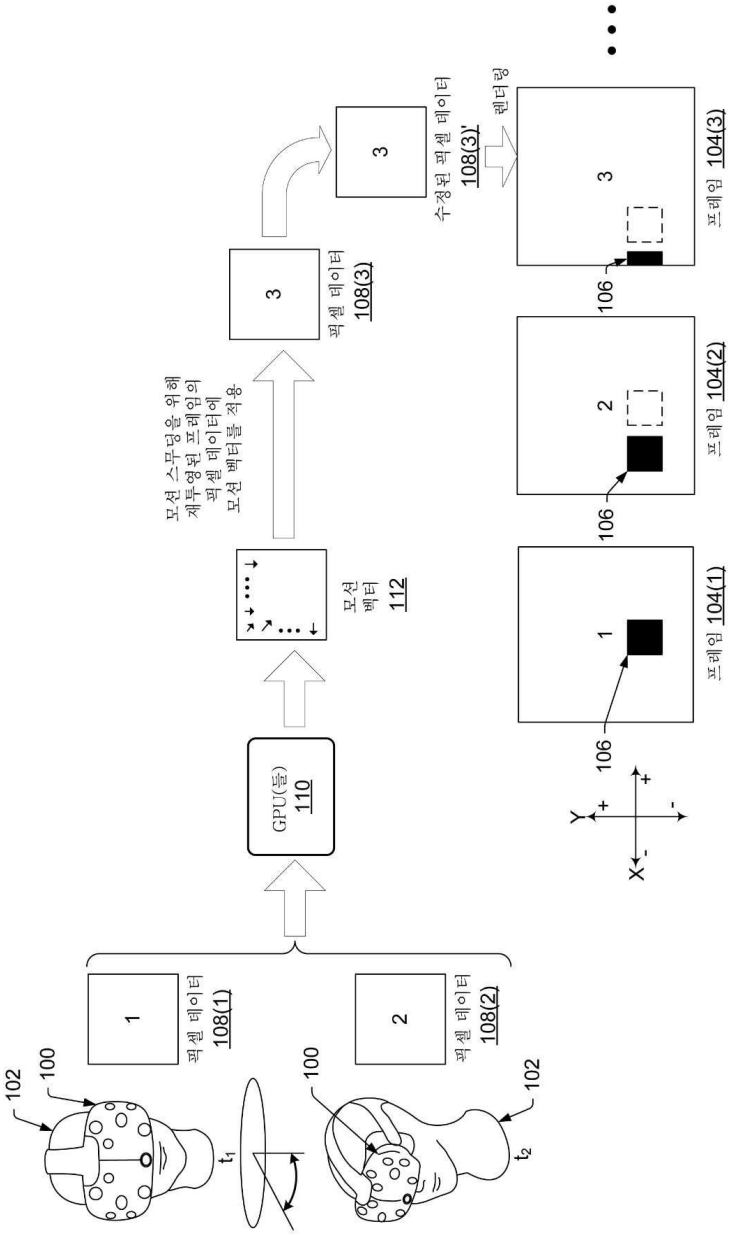
[0141] HMD(1600)는 헤드 트래킹 모듈(1630)을 더 포함할 수 있다. 헤드 트래킹 모듈(1630)은 전술한 바와 같이 사용자(102)의 머리 회전을 포함하는 헤드 모션을 추적하기 위해 센서(1624) 중 하나 이상을 이용할 수 있다. 예를 들어, 헤드 트래킹 모듈(1630)은 HMD(1600)의 최대 6개의 자유도(즉, 3D 위치, 롤, 피치 및 요)를 추적할 수 있다. 이러한 계산은 일련의 프레임(104)의 모든 프레임(104)에서 이루어질 수 있으므로 애플리케이션(1608)은 머리 위치 및 방향에 따라 다음 프레임(104)(재투영된 프레임(104)의 경우에도)에서 장면을 렌더링하는 방법을 결정할 수 있다. 일부 실시예에서, 헤드 트래킹 모듈(1630)은 현재 및/또는 이전 데이터에 기초하여 HMD(1600)의 향후 위치 및/또는 방향을 예측하도록 구성된다. 이는 사용자(102)가 디스플레이(들)(1616) 상의 광(및 이에 따른 이미지)을 실제로 보기 전에 애플리케이션이 프레임(104)을 렌더링하도록 요청되기 때문이다. 따라서, 다음 프레임(104)은 프레임(104)을 렌더링하기 전에 대략 25 내지 30 밀리세컨드(ms)와 같은 초기 시점에서 만들어진 헤드 위치 및/또는 방향의 이러한 미래 예측에 기초하여 렌더링될 수 있다. 헤드 트래킹 모듈(1630)에 의해 제공되는 회전 데이터는 임의의 적절한 측정 유닛에서 HMD(1600)의 HMD(1600) 회전 방향 및 회전량 둘 다를 결정하는 데 사용될 수 있다. 예를 들어, 회전 방향은 좌, 우, 상, 하에 대응하는 양 또는 음의 수평 및 양 또는 음의 수직 방향으로 단순화 및 출력될 수 있다. 회전량은 각도, 라디안 등으로 할 수 있다. 각속도는 HMD(1600)의 회전 속도를 결정하기 위해 계산될 수 있다.

[0142] 기술 요지는 구조적 특징에 특정적인 언어로 설명되었지만, 첨부된 청구범위에 정의된 기술 요지가 반드시 기술된 특정 특징에 제한되지는 않는다는 것을 이해해야 한다. 오히려, 특정 특징은 청구 범위를 구현하는 예시적인

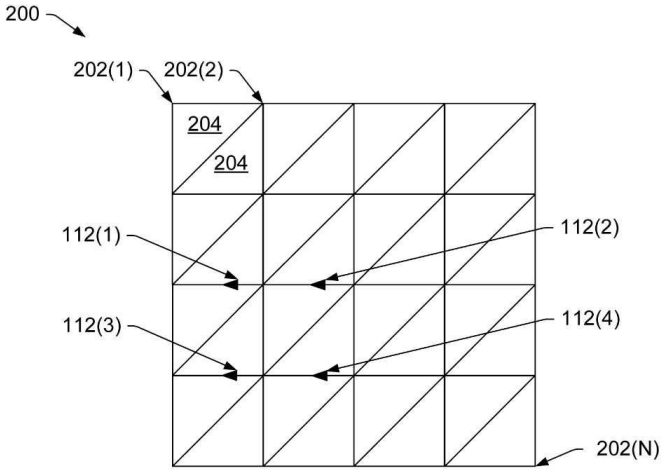
형태로 개시된다.

도면

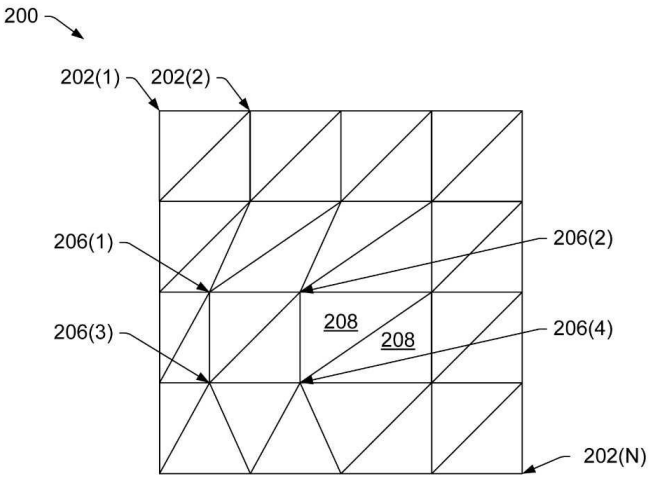
도면1



도면2

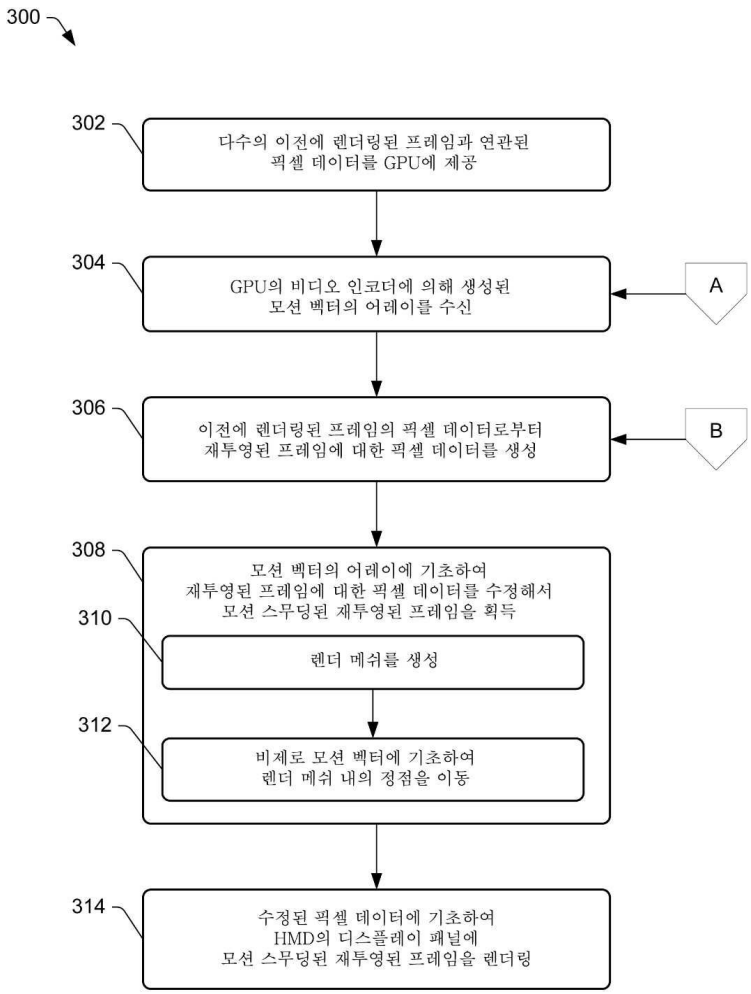


모션 스무딩 전

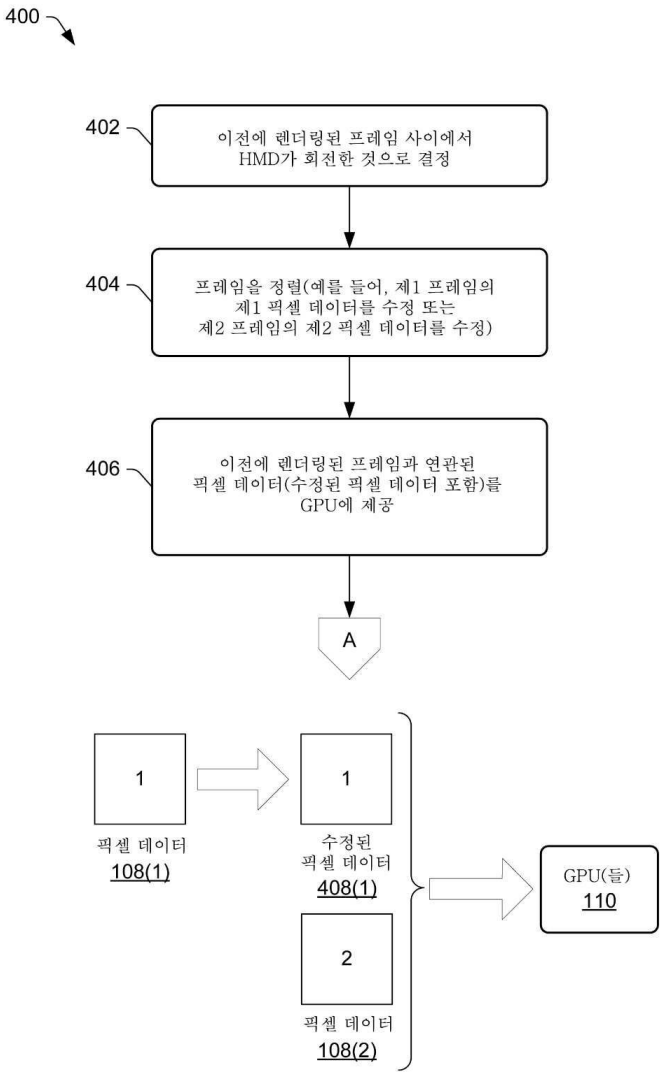


모션 스무딩 후

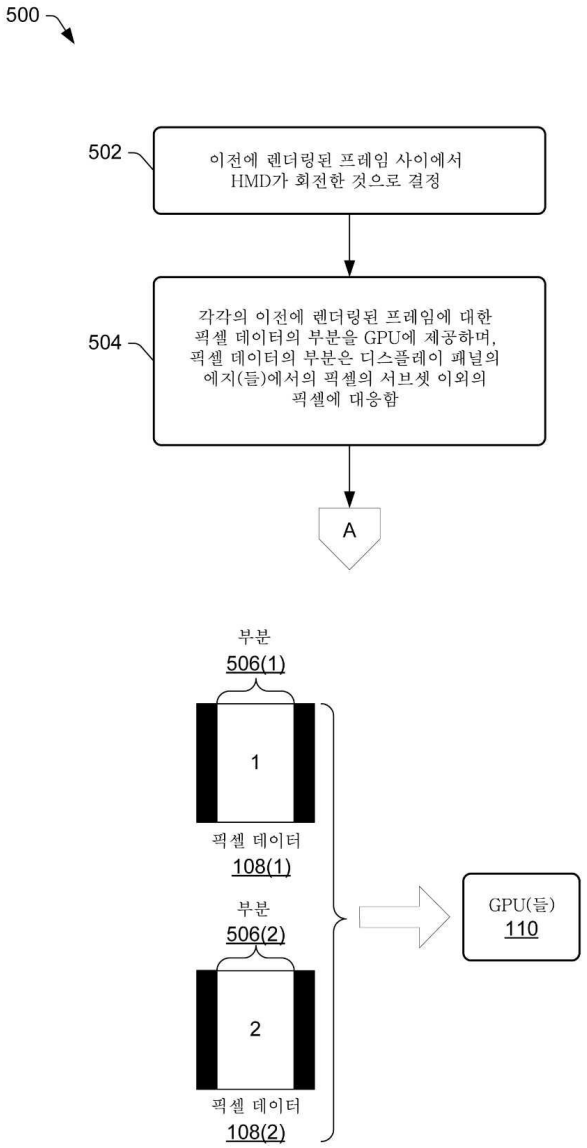
도면3


300
302
다수의 이전에 렌더링된 프레임과 연관된
픽셀 데이터를 GPU에 제공
304
GPU의 비디오 인코더에 의해 생성된
모션 벡터의 어레이를 수신
A
306
이전에 렌더링된 프레임의 픽셀 데이터로부터
재투영된 프레임에 대한 픽셀 데이터를 생성
B
308
모션 벡터의 어레이에 기초하여
재투영된 프레임에 대한 픽셀 데이터를 수정해서
모션 스무딩된 재투영된 프레임을 획득
310
렌더 메쉬를 생성
312
비제로 모션 벡터에 기초하여
렌더 메쉬 내의 정점을 이동
314
수정된 픽셀 데이터에 기초하여
HMD의 디스플레이 패널에
모션 스무딩된 재투영된 프레임을 렌더링

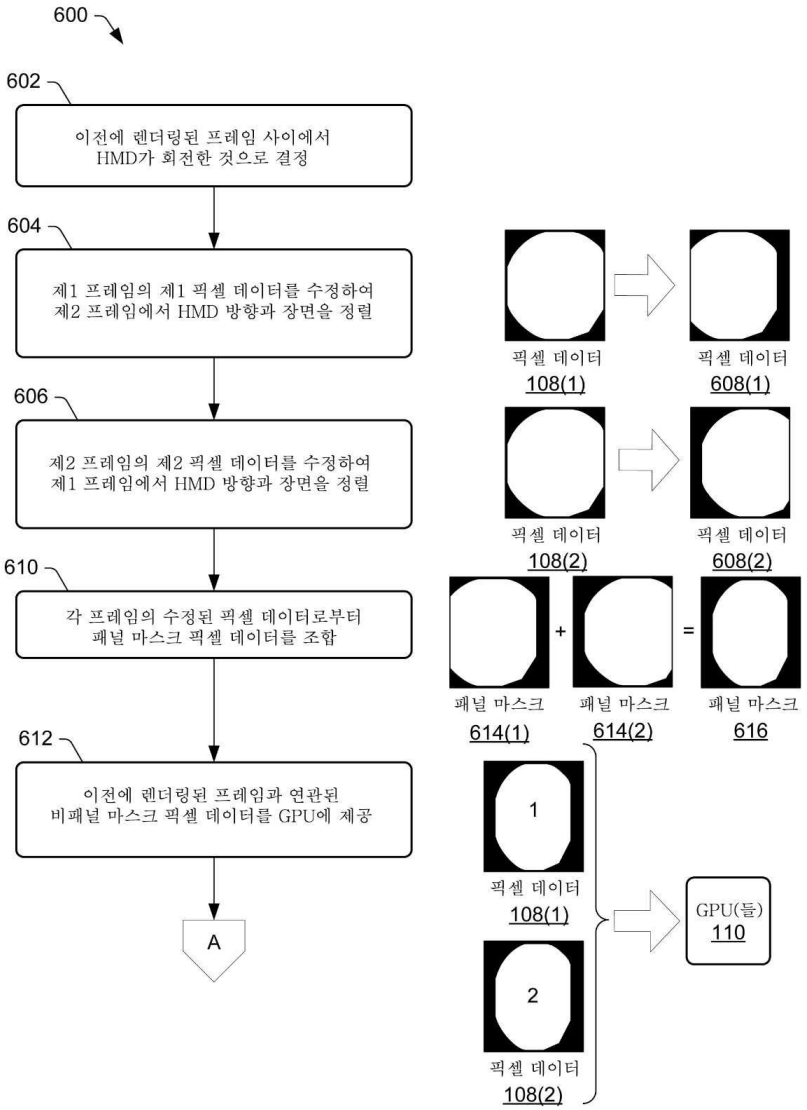
도면4


400
402
이전에 렌더링된 프레임 사이에서
HMD가 회전한 것으로 결정
404
프레임을 정렬(예를 들어, 제1 프레임의
제1 픽셀 데이터를 수정 또는
제2 프레임의 제2 픽셀 데이터를 수정)
406
이전에 렌더링된 프레임과 연관된
픽셀 데이터(수정된 픽셀 데이터 포함)를
GPU에 제공
A
1
픽셀 데이터
108(1)
1
수정된
픽셀 데이터
408(1)
2
픽셀 데이터
108(2)
GPU(들)
110

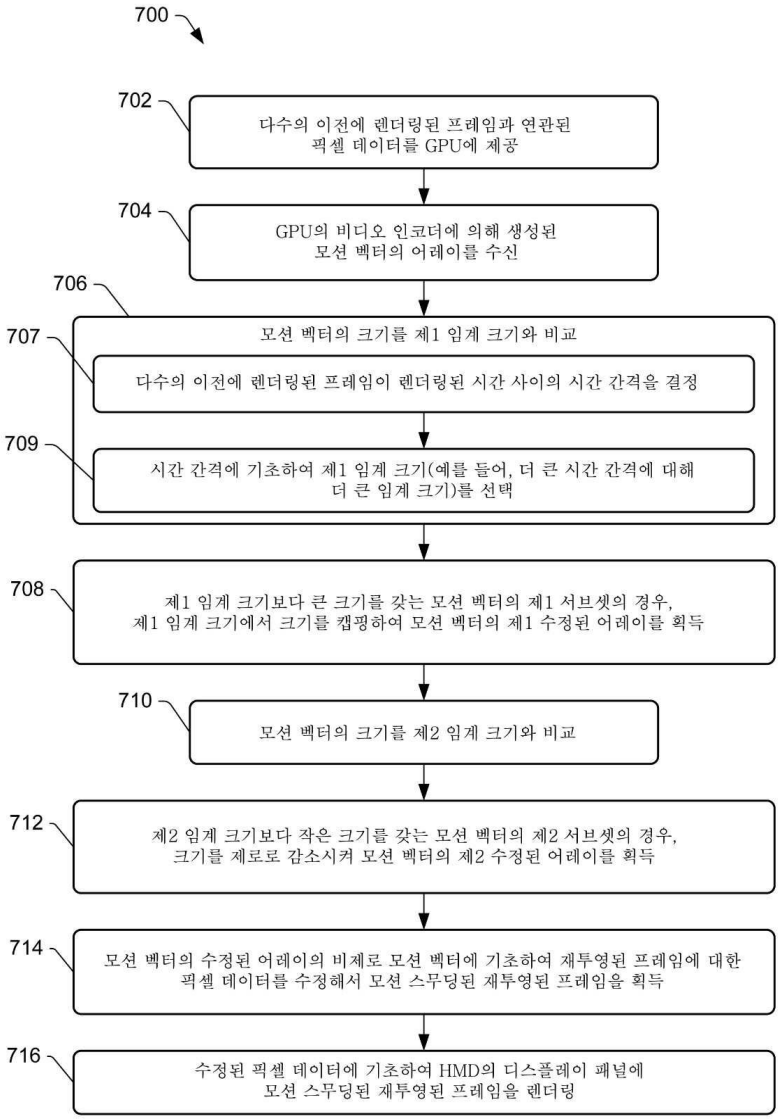
도면5


500
502
이전에 렌더링된 프레임 사이에서 HMD가 회전한 것으로 결정
504
각각의 이전에 렌더링된 프레임에 대한 픽셀 데이터의 부분을 GPU에 제공하며, 픽셀 데이터의 부분은 디스플레이 패널의 에지(들)에서의 픽셀의 서브셋 이외의 픽셀에 대응함
A
부분
506(1)
1
픽셀 데이터
108(1)
부분
506(2)
2
픽셀 데이터
108(2)
GPU(들)
110

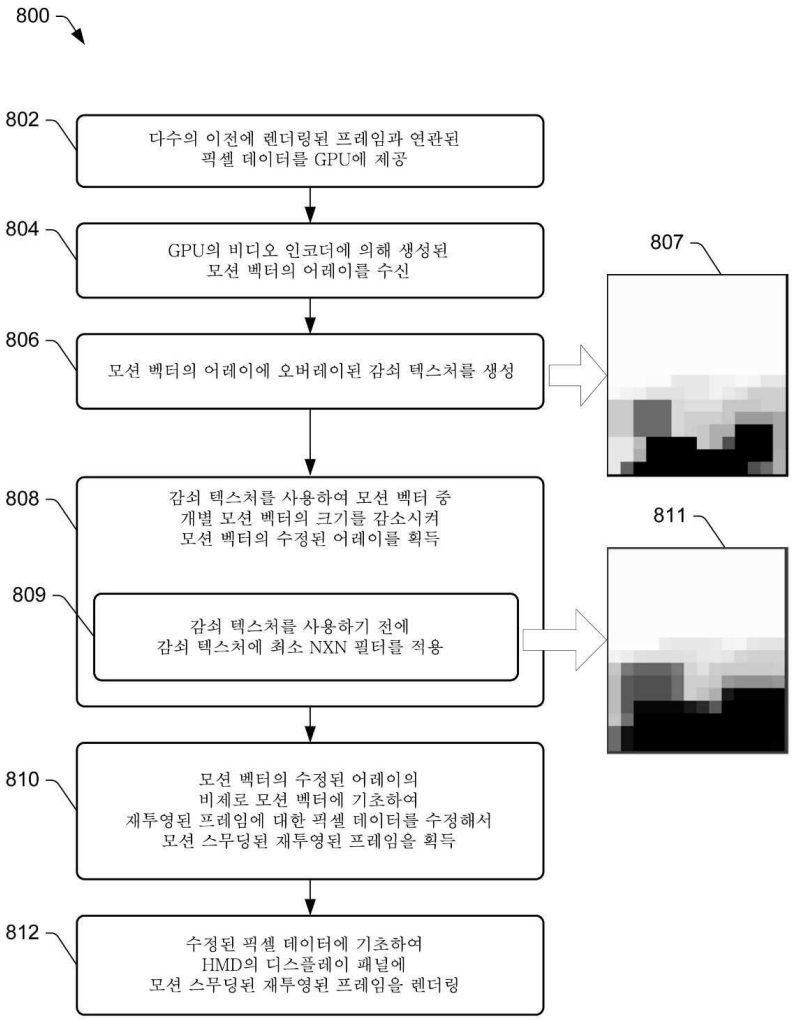
도면6


600
602
이전에 렌더링된 프레임 사이에서 HMD가 회전한 것으로 결정
604
제1 프레임의 제1 픽셀 데이터를 수정하여 제2 프레임에서 HMD 방향과 장면을 정렬
606
제2 프레임의 제2 픽셀 데이터를 수정하여 제1 프레임에서 HMD 방향과 장면을 정렬
610
각 프레임의 수정된 픽셀 데이터로부터 패널 마스크 픽셀 데이터를 조합
612
이전에 렌더링된 프레임과 연관된 비패널 마스크 픽셀 데이터를 GPU에 제공
A
픽셀 데이터
108(1)
픽셀 데이터
608(1)
픽셀 데이터
108(2)
픽셀 데이터
608(2)
+
=
패널 마스크
614(1)
패널 마스크
614(2)
패널 마스크
616
1
픽셀 데이터
108(1)
2
픽셀 데이터
108(2)
GPU(들)
110

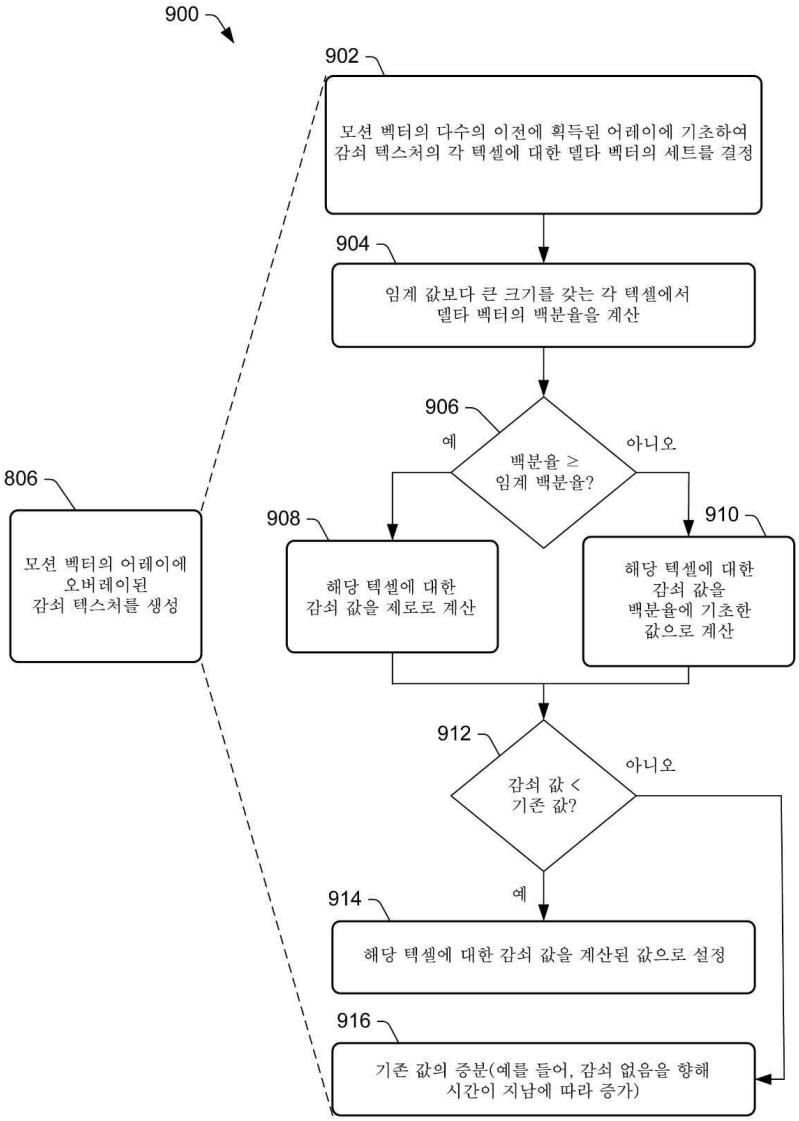
도면7


700
702
다수의 이전에 렌더링된 프레임과 연관된 픽셀 데이터를 GPU에 제공
704
GPU의 비디오 인코더에 의해 생성된 모션 벡터의 어레이를 수신
706
모션 벡터의 크기를 제1 임계 크기와 비교
707
다수의 이전에 렌더링된 프레임이 렌더링된 시간 사이의 시간 간격을 결정
709
시간 간격에 기초하여 제1 임계 크기(예를 들어, 더 큰 시간 간격에 대해 더 큰 임계 크기)를 선택
708
제1 임계 크기보다 큰 크기를 갖는 모션 벡터의 제1 서브셋의 경우, 제1 임계 크기에서 크기를 캡핑하여 모션 벡터의 제1 수정된 어레이를 획득
710
모션 벡터의 크기를 제2 임계 크기와 비교
712
제2 임계 크기보다 작은 크기를 갖는 모션 벡터의 제2 서브셋의 경우, 크기를 제로로 감소시켜 모션 벡터의 제2 수정된 어레이를 획득
714
모션 벡터의 수정된 어레이의 비제로 모션 벡터에 기초하여 재투영된 프레임에 대한 픽셀 데이터를 수정해서 모션 스무딩된 재투영된 프레임을 획득
716
수정된 픽셀 데이터에 기초하여 HMD의 디스플레이 패널에 모션 스무딩된 재투영된 프레임을 렌더링

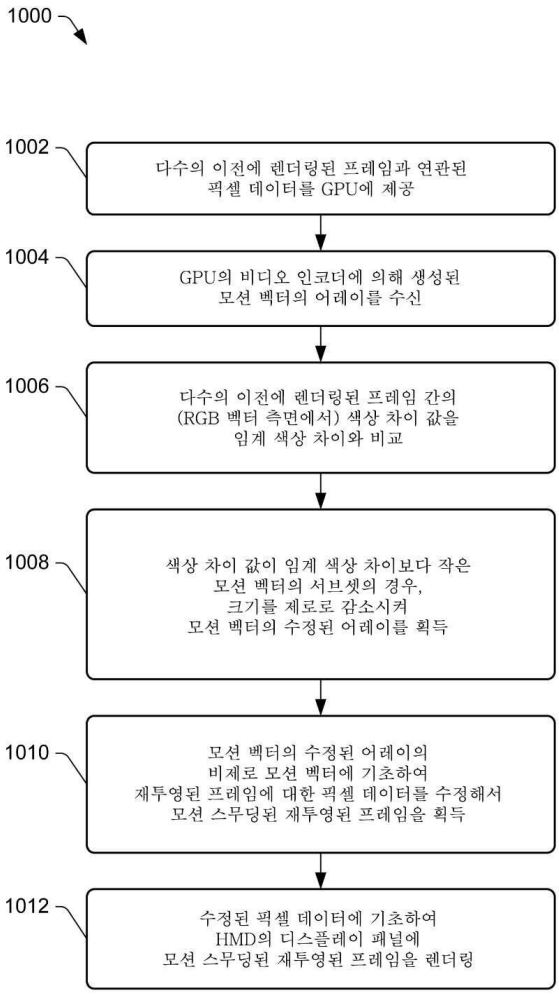
도면8


800
802
다수의 이전에 렌더링된 프레임과 연관된
픽셀 데이터를 GPU에 제공
804
GPU의 비디오 인코더에 의해 생성된
모션 벡터의 어레이를 수신
806
모션 벡터의 어레이에 오버레이된 감쇠 텍스처를 생성
807
808
감쇠 텍스처를 사용하여 모션 벡터 중
개별 모션 벡터의 크기를 감소시켜
모션 벡터의 수정된 어레이를 획득
809
감쇠 텍스처를 사용하기 전에
감쇠 텍스처에 최소 NXN 필터를 적용
811
810
모션 벡터의 수정된 어레이의
비제로 모션 벡터에 기초하여
재투영된 프레임에 대한 픽셀 데이터를 수정해서
모션 스무딩된 재투영된 프레임을 획득
812
수정된 픽셀 데이터에 기초하여
HMD의 디스플레이 패널에
모션 스무딩된 재투영된 프레임을 렌더링

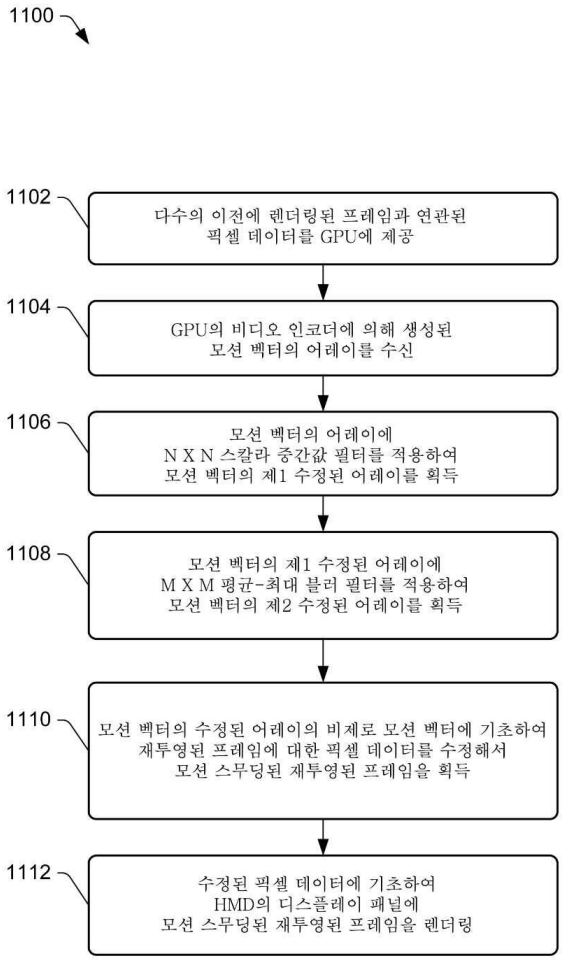
도면9


900
902
모션 벡터의 다수의 이전에 획득된 어레이에 기초하여 감쇠 텍스처의 각 텍셀에 대한 델타 벡터의 세트를 결정
904
임계 값보다 큰 크기를 갖는 각 텍셀에서 델타 벡터의 백분율을 계산
906
백분율 ≥ 임계 백분율?
예
아니오
806
모션 벡터의 어레이에 오버레이된 감쇠 텍스처를 생성
908
해당 텍셀에 대한 감쇠 값을 제로로 계산
910
해당 텍셀에 대한 감쇠 값을 백분율에 기초한 값으로 계산
912
감쇠 값 < 기존 값?
아니오
예
914
해당 텍셀에 대한 감쇠 값을 계산된 값으로 설정
916
기존 값의 증분(예를 들어, 감쇠 없음을 향해 시간이 지남에 따라 증가)

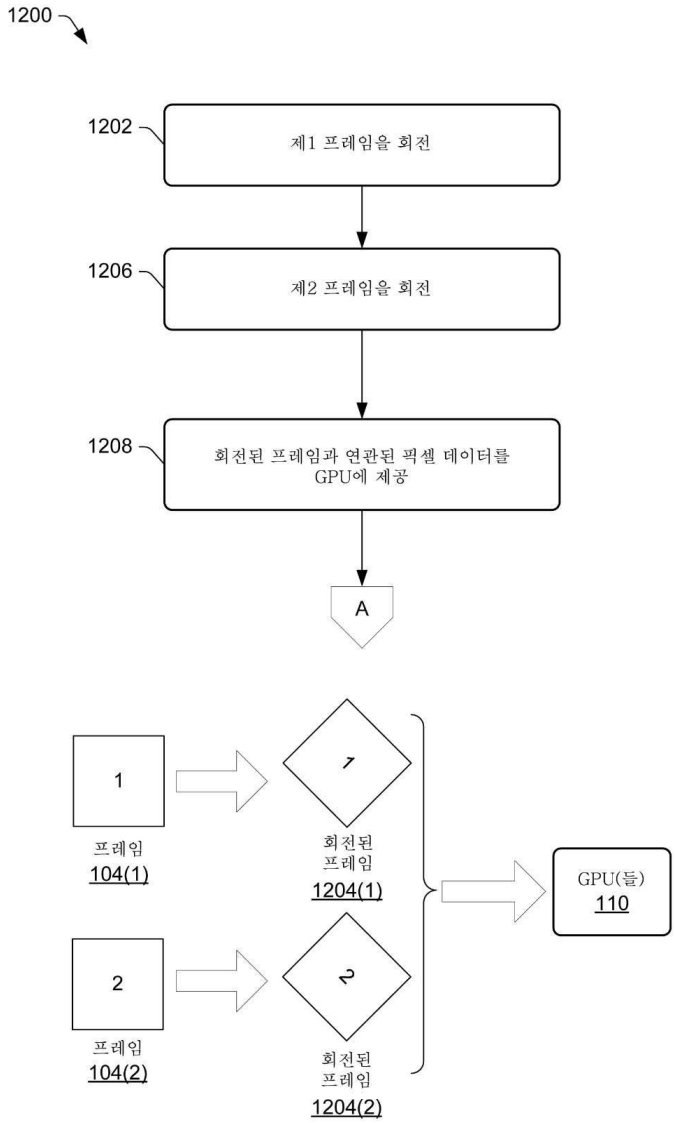
도면10


1000
1002
다수의 이전에 렌더링된 프레임과 연관된 픽셀 데이터를 GPU에 제공
1004
GPU의 비디오 인코더에 의해 생성된 모션 벡터의 어레이를 수신
1006
다수의 이전에 렌더링된 프레임 간의 (RGB 벡터 측면에서) 색상 차이 값을 임계 색상 차이와 비교
1008
색상 차이 값이 임계 색상 차이보다 작은 모션 벡터의 서브셋의 경우, 크기를 제로로 감소시켜 모션 벡터의 수정된 어레이를 획득
1010
모션 벡터의 수정된 어레이의 비제로 모션 벡터에 기초하여 재투영된 프레임에 대한 픽셀 데이터를 수정해서 모션 스무딩된 재투영된 프레임을 획득
1012
수정된 픽셀 데이터에 기초하여 HMD의 디스플레이 패널에 모션 스무딩된 재투영된 프레임을 렌더링

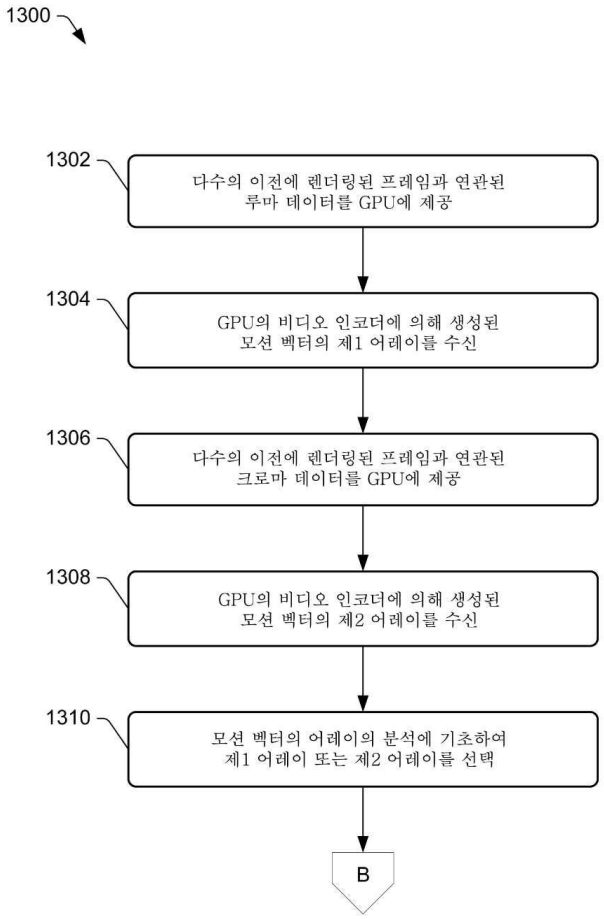
도면11


1100
1102
다수의 이전에 렌더링된 프레임과 연관된
픽셀 데이터를 GPU에 제공
1104
GPU의 비디오 인코더에 의해 생성된
모션 벡터의 어레이를 수신
1106
모션 벡터의 어레이에
N X N 스칼라 중간값 필터를 적용하여
모션 벡터의 제1 수정된 어레이를 획득
1108
모션 벡터의 제1 수정된 어레이에
M X M 평균-최대 블러 필터를 적용하여
모션 벡터의 제2 수정된 어레이를 획득
1110
모션 벡터의 수정된 어레이의 비제로 모션 벡터에 기초하여
재투영된 프레임에 대한 픽셀 데이터를 수정해서
모션 스무딩된 재투영된 프레임을 획득
1112
수정된 픽셀 데이터에 기초하여
HMD의 디스플레이 패널에
모션 스무딩된 재투영된 프레임을 렌더링

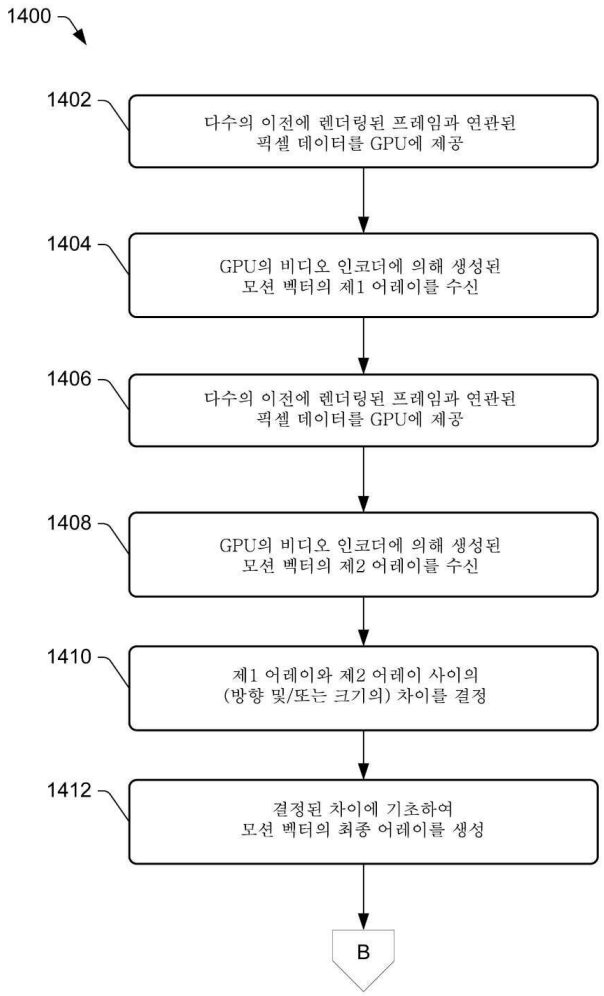
도면12


1200
1202
제1 프레임을 회전
1206
제2 프레임을 회전
1208
회전된 프레임과 연관된 픽셀 데이터를 GPU에 제공
A
1
프레임
104(1)
1'
회전된
프레임
1204(1)
2
프레임
104(2)
2'
회전된
프레임
1204(2)
GPU(들)
110

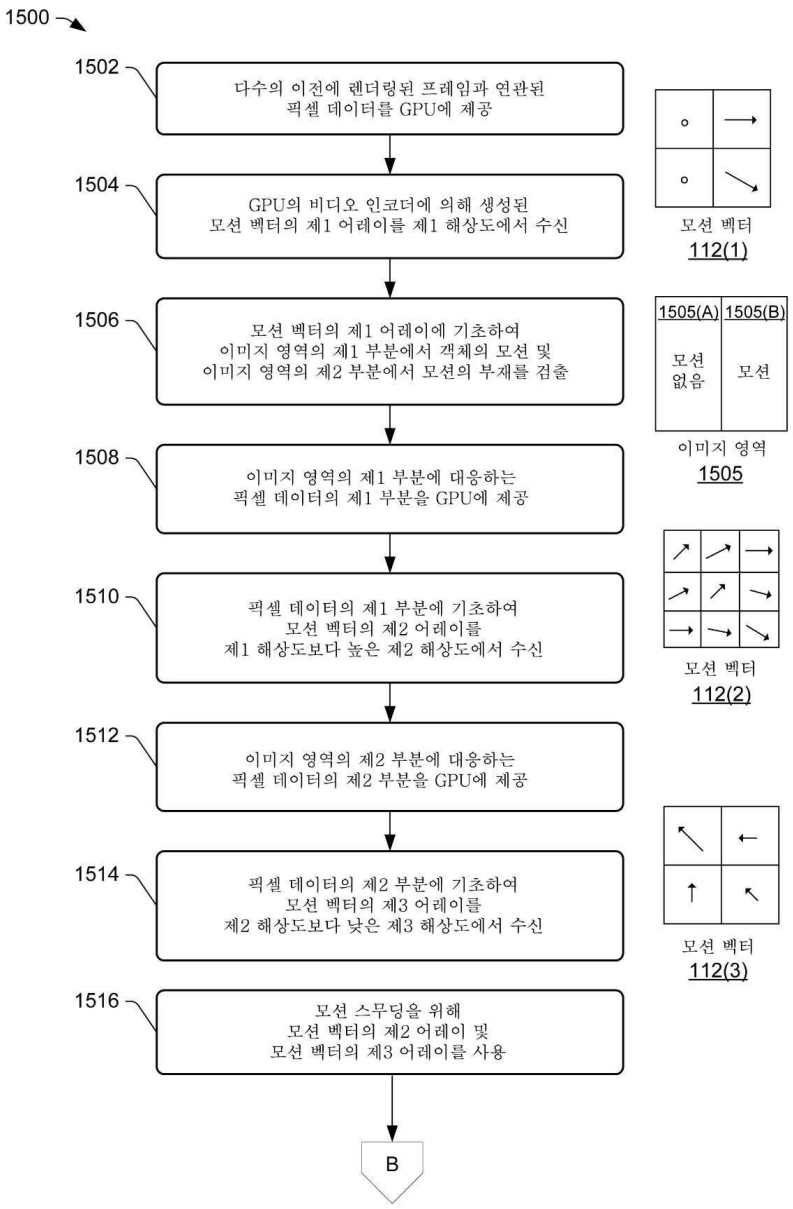
도면13


1300
1302
다수의 이전에 렌더링된 프레임과 연관된 루마 데이터를 GPU에 제공
1304
GPU의 비디오 인코더에 의해 생성된 모션 벡터의 제1 어레이를 수신
1306
다수의 이전에 렌더링된 프레임과 연관된 크로마 데이터를 GPU에 제공
1308
GPU의 비디오 인코더에 의해 생성된 모션 벡터의 제2 어레이를 수신
1310
모션 벡터의 어레이의 분석에 기초하여 제1 어레이 또는 제2 어레이를 선택
B

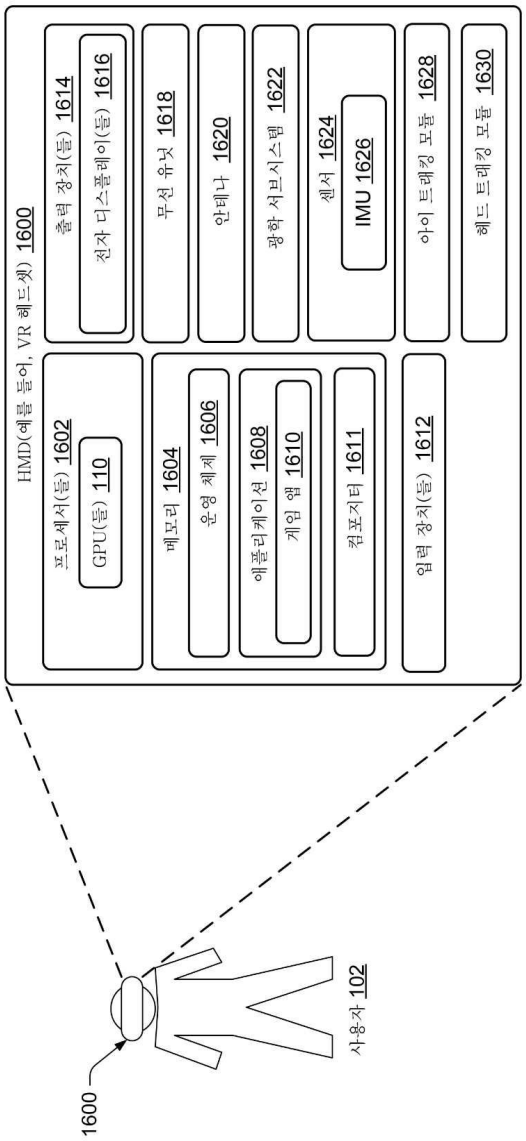
도면14


1400
1402
다수의 이전에 렌더링된 프레임과 연관된 픽셀 데이터를 GPU에 제공
1404
GPU의 비디오 인코더에 의해 생성된 모션 벡터의 제1 어레이를 수신
1406
다수의 이전에 렌더링된 프레임과 연관된 픽셀 데이터를 GPU에 제공
1408
GPU의 비디오 인코더에 의해 생성된 모션 벡터의 제2 어레이를 수신
1410
제1 어레이와 제2 어레이 사이의 (방향 및/또는 크기의) 차이를 결정
1412
결정된 차이에 기초하여 모션 벡터의 최종 어레이를 생성
B

도면15


1500
1502
다수의 이전에 렌더링된 프레임과 연관된 픽셀 데이터를 GPU에 제공
1504
GPU의 비디오 인코더에 의해 생성된 모션 벡터의 제1 어레이를 제1 해상도에서 수신
1506
모션 벡터의 제1 어레이에 기초하여 이미지 영역의 제1 부분에서 객체의 모션 및 이미지 영역의 제2 부분에서 모션의 부재를 검출
1508
이미지 영역의 제1 부분에 대응하는 픽셀 데이터의 제1 부분을 GPU에 제공
1510
픽셀 데이터의 제1 부분에 기초하여 모션 벡터의 제2 어레이를 제1 해상도보다 높은 제2 해상도에서 수신
1512
이미지 영역의 제2 부분에 대응하는 픽셀 데이터의 제2 부분을 GPU에 제공
1514
픽셀 데이터의 제2 부분에 기초하여 모션 벡터의 제3 어레이를 제2 해상도보다 낮은 제3 해상도에서 수신
1516
모션 스무딩을 위해 모션 벡터의 제2 어레이 및 모션 벡터의 제3 어레이를 사용
B
모션 벡터
112(1)
1505(A)
1505(B)
모션 없음
모션
이미지 영역
1505
모션 벡터
112(2)
모션 벡터
112(3)

도면16


HMD(예를 들어, VR 헤드셋) 1600
프로세서(들) 1602
GPU(들) 110
메모리 1604
운영 체제 1606
애플리케이션 1608
게임 앱 1610
컴포지터 1611
입력 장치(들) 1612
출력 장치(들) 1614
전자 디스플레이(들) 1616
무선 유닛 1618
안테나 1620
광학 서브시스템 1622
센서 1624
IMU 1626
아이 트래킹 모듈 1628
헤드 트래킹 모듈 1630
사용자 102
1600