



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2021년05월31일
(11) 등록번호 10-2258446
(24) 등록일자 2021년05월25일

- (51) 국제특허분류(Int. Cl.)
H04N 13/156 (2018.01) H04N 13/111 (2018.01)
H04N 13/178 (2018.01) H04N 21/218 (2011.01)
H04N 21/81 (2011.01)
- (52) CPC특허분류
H04N 13/156 (2018.05)
H04N 13/111 (2018.05)
- (21) 출원번호 10-2019-0076607
- (22) 출원일자 2019년06월26일
심사청구일자 2019년06월26일
- (65) 공개번호 10-2020-0006918
- (43) 공개일자 2020년01월21일
- (30) 우선권주장
1020180080574 2018년07월11일 대한민국(KR)
- (56) 선행기술조사문헌
KR1020180001777 A*
KR1020180028299 A*
KR1020180029473 A*
*는 심사관에 의하여 인용된 문헌

- (73) 특허권자
엘지전자 주식회사
서울특별시 영등포구 여의대로 128 (여의도동)
- (72) 발명자
허혜정
서울시 서초구 양재대로11길 19 LG전자 특허센터
오세진
서울시 서초구 양재대로11길 19 LG전자 특허센터
이장원
서울시 서초구 양재대로11길 19 LG전자 특허센터
- (74) 대리인
특허법인(유한)케이비케이

전체 청구항 수 : 총 19 항

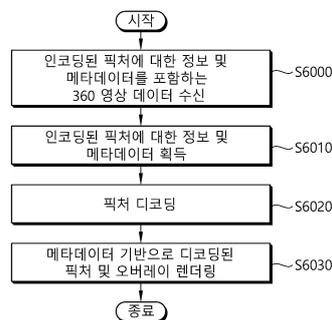
심사관 : 옥윤철

(54) 발명의 명칭 360 비디오 시스템에서 오버레이 처리 방법 및 그 장치

(57) 요약

본 발명은 360 비디오 수신 장치에 의하여 수행되는 360 비디오 데이터 처리 방법으로서, 360 영상 데이터를 수신하는 단계, 상기 360 영상 데이터로부터 인코딩된 픽처에 대한 정보 및 메타데이터를 획득하는 단계, 상기 인코딩된 픽처에 대한 정보를 기반으로 픽처를 디코딩하는 단계 및 상기 메타데이터를 기반으로 디코딩된 픽처 및 오버레이를 렌더링하는 단계를 포함하고, 상기 메타데이터는 오버레이 관련 메타데이터를 포함하고, 상기 오버레이 관련 메타데이터를 기반으로 상기 오버레이를 렌더링하고, 상기 오버레이 관련 메타데이터는 상기 오버레이의 그룹 정보를 포함하는 것을 특징으로 한다.

대표도



(52) CPC특허분류

H04N 13/178 (2018.05)

H04N 21/21805 (2013.01)

H04N 21/816 (2013.01)

명세서

청구범위

청구항 1

360도 비디오 데이터를 수신하는 단계, 상기 360도 비디오 데이터는 하나 또는 그 이상의 픽처들 및 메타데이터를 포함함;

상기 하나 또는 그 이상의 픽처들을 디코딩하는 단계; 및

상기 메타데이터에 기초하여 상기 디코딩된 하나 또는 그 이상의 픽처들 및 오버레이 미디어를 렌더링하는 단계를 포함하고,

상기 오버레이 미디어는 상기 디코딩된 하나 또는 그 이상의 픽처들 위에 렌더링(rendered over)되고,

상기 메타데이터는 상기 오버레이 미디어와 관련된 정보를 포함하고,

상기 오버레이 미디어와 관련된 정보는 상기 오버레이 미디어를 구성하는 방법을 나타내는 오버레이 알파 컴포지션(overlay alpha composition) 정보를 포함하는,

360도 비디오 데이터 처리 방법.

청구항 2

제1항에 있어서,

상기 오버레이 미디어와 관련된 정보는,

스위칭이 가능한 오버레이 미디어들에 대한 정보를 포함하고,

상기 스위칭이 가능한 오버레이 미디어들에 대한 정보는 상기 스위칭이 가능한 오버레이 미디어들에 대한 식별 정보를 포함하고,

상기 오버레이 미디어와 관련된 정보는,

현재 액티브(currently active) 상태의 오버레이 미디어에 대한 식별 정보를 더 포함하는,

360도 비디오 데이터 처리 방법.

청구항 3

제1항에 있어서,

상기 오버레이 미디어와 관련된 정보는 상기 오버레이 미디어에 대한 인터랙션(interaction)을 나타내는 인터랙션 정보를 더 포함하고,

상기 인터랙션 정보는 상기 오버레이 미디어가 사용자에게 의해 켜져있는지(on) 또는 꺼져있는지(off) 여부를 나타내는 스위치 온 오프 플래그(switch on off flag), 상기 오버레이 미디어의 투명도(opacity)가 상기 사용자에게 의해 변경될 수 있는지 여부를 나타내는 변경 투명 플래그(change opacity flag), 상기 오버레이 미디어의 위치가 상기 사용자에게 의해 변경될 수 있는지 여부를 나타내는 위치 플래그(position flag), 상기 오버레이 미디어의 뎁스(depth)가 상기 사용자에게 의해 변경될 수 있는지 여부를 나타내는 뎁스 플래그(depth flag), 상기 오버레이 미디어의 스케일이 상기 사용자에게 의해 변경될 수 있는지 여부를 나타내는 리사이즈 플래그(resize flag), 및 상기 오버레이 미디어가 상기 사용자에게 의해 회전될 수 있는지 여부를 나타내는 로테이션 플래그(rotation flag)를 더 포함하는,

360도 비디오 데이터 처리 방법.

청구항 4

삭제

청구항 5

제1항에 있어서,

상기 오버레이 미디어와 관련된 정보는 상기 오버레이 미디어와 함께 렌더링될 메인 미디어를 지시하는 정보를 포함하고,

상기 오버레이 미디어와 함께 렌더링될 메인 미디어를 지시하는 정보는 grouping_type 필드를 가지는 EntityToGroupBox에 의해 지시되고,

상기 디코딩된 픽처는 상기 메인 미디어를 포함하는 것을 특징으로 하는,

360도 비디오 데이터 처리 방법.

청구항 6

제1항에 있어서,

상기 오버레이 미디어가 배경 미디어가 포함된 트랙(track) 또는 이미지 아이템(image item) 내에 포함되는 경우, 오버레이 소스 영역은 상기 배경 미디어의 패킹된 리전과 오버랩되지 않는,

360도 비디오 데이터 처리 방법.

청구항 7

제1항에 있어서,

상기 메타데이터는,

스위칭이 가능한 복수의 메인 미디어에 대한 정보를 더 포함하는 것을 특징으로 하는,

360도 비디오 데이터 처리 방법.

청구항 8

제1항에 있어서,

상기 360도 비디오 데이터는 복수의 트랙을 포함하고,

상기 메타데이터는 그룹 정보에 의해 지시되는 엔티티 그룹 내의 제1 트랙이 메인 미디어를 포함하는지를 지시하는 제1 플래그 및 상기 엔티티 그룹 내의 제2 트랙이 오버레이 미디어를 포함하는지를 지시하는 제2 플래그를 더 포함하는 것을 특징으로 하는,

360도 비디오 데이터 처리 방법.

청구항 9

제1항에 있어서,

상기 오버레이 미디어를 구성하는 방법은 소스 오버(source over) 타입에 기초하여 결정되는,

360도 비디오 데이터 처리 방법.

청구항 10

제9항에 있어서,

상기 오버레이 알파 컴포지션 정보의 알파 블렌딩 모드(alpha blending mode)는,

소스 오버(source over)를 포함하는 것을 특징으로 하는,

360도 비디오 데이터 처리 방법.

청구항 11

제10항에 있어서,

상기 소스 오버 타입에 기초하여 상기 오버레이 미디어를 구성하는 경우, 상기 오버레이 미디어는 상기 오버레이 미디어의 알파(alpha) 값 및 상기 오버레이 미디어의 픽셀(pixel) 값에 기초하여 상기 디코딩된 하나 또는 그 이상의 픽처들 위에 렌더링되는,

360도 비디오 데이터 처리 방법.

청구항 12

제1항에 있어서,

상기 오버레이 미디어와 관련된 정보는 상기 오버레이 미디어가 렌더링되는 스피어 영역(sphere region)을 나타내는 정보를 포함하고, 상기 스피어 영역은 두 개의 아지무스 원(azimuth circle)과 두 개의 엘리베이션 원(elevation circle)에 기초하여 결정되는,

360도 비디오 데이터 처리 방법.

청구항 13

제12항에 있어서,

상기 오버레이 미디어와 관련된 정보는 상기 스피어 영역의 중심 위치를 나타내는 정보, 상기 스피어 영역의 범위를 나타내는 정보, 및 상기 중심 위치로부터 상기 스피어 영역의 중심 위치를 향한 축을 기준으로 기울어진(tilt) 정도를 나타내는 틸트 정보를 포함하는,

360도 비디오 데이터 처리 방법.

청구항 14

제1항에 있어서,

상기 메타데이터는 상기 오버레이 미디어의 스태틱(static) 메타데이터를 포함하고,

상기 오버레이 미디어의 스태틱 메타데이터는 OverlayConfigBox에 저장되고,

상기 OverlayConfigBox는 SchemeTypeBox 내의 scheme_type 필드 값이 podv인 경우, ProjectedOmniVideoBox에 포함되는,

360도 비디오 데이터 처리 방법.

청구항 15

제1항에 있어서,

상기 메타데이터는 이미지(image) 내에 포함된 상기 오버레이 미디어에 대한 정보를 포함하고,

상기 오버레이 미디어에 대한 정보는 OverlayConfigProperty에 저장되고,

상기 OverlayConfigProperty는 ItemPropertyContainerBox에 포함되는,

360도 비디오 데이터 처리 방법.

청구항 16

하나 또는 그 이상의 픽처들을 획득하는 단계;

상기 하나 또는 그 이상의 픽처들을 인코딩하는 단계; 및

상기 하나 또는 그 이상의 픽처들 및 메타데이터를 포함하는 360도 비디오 데이터를 전송하는 단계,

상기 메타데이터는 360도 비디오 데이터 처리 장치에서 오버레이 미디어를 상기 하나 또는 그 이상의 픽처들 위에 렌더링(rendered over)하기 위한, 상기 오버레이 미디어와 관련된 정보를 포함하고,

상기 오버레이 미디어와 관련된 정보는 상기 오버레이 미디어를 구성하는 방법을 나타내는 오버레이 알파 컴포지션(overlay alpha composition) 정보를 포함하는,

360도 비디오 데이터 전송 방법.

청구항 17

제16항에 있어서,

상기 오버레이 미디어와 관련된 정보는 상기 오버레이 미디어와 연관된 스피어 리전(sphere region)을 나타내는 정보를 더 포함하는,

360도 비디오 데이터 전송 방법.

청구항 18

삭제

청구항 19

제16항에 있어서,

상기 오버레이 미디어가 배경 미디어가 포함된 트랙(track) 또는 이미지 아이템(image item) 내에 포함되는 경우, 오버레이 소스 영역은 상기 배경 미디어의 패킹된 리전과 오버랩되지 않는,

360도 비디오 데이터 전송 방법.

청구항 20

360도 비디오 데이터를 수신하는 수신부, 상기 360도 비디오 데이터는 하나 또는 그 이상의 픽처들 및 메타데이터를 포함함;

상기 하나 또는 그 이상의 픽처들을 디코딩하는 디코더; 및

상기 메타데이터에 기초하여 상기 디코딩된 하나 또는 그 이상의 픽처들 및 오버레이 미디어를 렌더링하는 렌더러, 상기 오버레이 미디어는 상기 디코딩된 하나 또는 그 이상의 픽처들 위에 렌더링(rendered over)되고,

상기 메타데이터는 상기 오버레이 미디어와 관련된 정보를 포함하고,

상기 오버레이 미디어와 관련된 정보는 상기 오버레이 미디어를 구성하는 방법을 나타내는 오버레이 알파 컴포지션(overlay alpha composition) 정보를 포함하는,

360도 비디오 데이터 처리 장치.

청구항 21

하나 또는 그 이상의 픽처들을 획득하는 획득부;

상기 하나 또는 그 이상의 픽처들을 인코딩하는 인코더; 및

상기 하나 또는 그 이상의 픽처들 및 메타데이터를 포함하는 360도 비디오 데이터를 전송하는 전송부,

상기 메타데이터는 360도 비디오 데이터 처리 장치에서 오버레이 미디어를 상기 하나 또는 그 이상의 픽처들 위에 렌더링(rendered over)하기 위한, 상기 오버레이 미디어와 관련된 정보를 포함하고,

상기 오버레이 미디어와 관련된 정보는 상기 오버레이 미디어를 구성하는 방법을 나타내는 오버레이 알파 컴포지션(overlay alpha composition) 정보를 포함하는, 360도 비디오 데이터 전송 장치.

발명의 설명

기술 분야

[0001] 본 발명은 360 비디오에 관한 것으로, 보다 상세하게는 360 비디오 시스템에서 오버레이 처리 방법 및 그 장치에 관한 것이다.

배경 기술

[0002] VR(Virtual Reality) 시스템은 사용자에게 전자적으로 투영된 환경 내에 있는 것 같은 감각을 제공한다. AR(Augmented Reality, AR) 시스템은 현실의 이미지나 배경에 3차원 가상 이미지를 중첩하여, 사용자에게 가상과 현실이 혼합된 환경 내에 있는 것 같은 감각을 제공한다. VR 또는 AR을 제공하기 위한 시스템은 더 고화질의 이미지들과, 공간적인 음향을 제공하기 위하여 더 개선될 수 있다. VR 또는 AR 시스템은 사용자가 인터랙티브하게 VR 또는 AR 콘텐츠를 소비할 수 있도록 할 수 있다.

발명의 내용

해결하려는 과제

- [0003] 본 발명의 기술적 과제는 360 비디오 데이터 처리 방법 및 장치를 제공함에 있다.
- [0004] 본 발명의 다른 기술적 과제는 360 비디오 데이터에 대한 메타데이터를 전송하는 방법 및 장치를 제공함에 있다.
- [0005] 본 발명의 또 다른 기술적 과제는 360 비디오에 대한 오버레이 처리 방법 및 장치를 제공함에 있다.
- [0006] 본 발명의 또 다른 기술적 과제는 360 비디오에 대한 오버레이를 위한 메타데이터를 전송하는 방법 및 장치를 제공함에 있다.

과제의 해결 수단

- [0007] 본 발명의 일 실시예에 따르면, 360 비디오 수신 장치에 의하여 수행되는 360 비디오 데이터 처리 방법을 제공한다. 상기 방법은 360 영상 데이터를 수신하는 단계, 상기 360 영상 데이터로부터 인코딩된 픽처에 대한 정보 및 메타데이터를 획득하는 단계, 상기 인코딩된 픽처에 대한 정보를 기반으로 픽처를 디코딩하는 단계, 상기 메타데이터를 기반으로 디코딩된 픽처 및 오버레이를 렌더링하는 단계를 포함하고, 상기 메타데이터는 오버레이 관련 메타데이터를 포함하고, 상기 오버레이 관련 메타데이터를 기반으로 상기 오버레이를 렌더링하고, 상기 오버레이 관련 메타데이터는 상기 오버레이의 그룹 정보를 포함하는 것을 특징으로 한다.
- [0008] 본 발명의 다른 실시예에 따르면, 360 비디오 전송 장치에 의하여 수행되는 360 영상 데이터 처리 방법을 제공한다. 상기 방법은 360 영상을 획득하는 단계, 상기 360 영상을 처리하여 픽처를 도출하는 단계, 상기 360 영상에 관한 메타데이터를 생성하는 단계, 상기 픽처를 인코딩하는 단계 및 상기 인코딩된 픽처 및 상기 메타데이터에 대하여 저장 또는 전송을 위한 처리를 수행하는 단계를 포함하되, 상기 메타데이터는 오버레이 관련 메타데이터를 포함하고, 상기 오버레이 관련 메타데이터는 상기 오버레이의 그룹 정보를 포함하는 것을 특징으로 한다.
- [0009] 본 발명의 또 다른 일 실시예에 따르면, 360 비디오 수신 장치가 제공된다. 상기 360 비디오 수신 장치는 360 영상 데이터를 수신하고, 상기 360 영상 데이터로부터 인코딩된 픽처에 대한 정보 및 메타데이터를 획득하는 수신처리부, 상기 인코딩된 픽처에 대한 정보를 기반으로 픽처를 디코딩 데이터 디코더 및 상기 메타데이터를 기반으로 디코딩된 픽처 및 오버레이를 렌더링하는 렌더러를 포함하되, 상기 메타데이터는 오버레이 관련 메타데이터를 포함하고, 상기 렌더러는 오버레이 관련 메타데이터를 기반으로 상기 오버레이를 렌더링하고, 상기 오버레이 관련 메타데이터는 상기 오버레이의 그룹 정보를 포함하는 것을 특징으로 한다.
- [0010] 본 발명의 또 다른 일 실시예에 따르면, 360 비디오 전송 장치가 제공된다. 상기 360 비디오 전송 장치는 360 영상을 획득하는 데이터 입력부, 상기 360 영상을 처리하여 픽처를 도출하는 프로젝션 처리부, 상기 360 영상에 관한 메타데이터를 생성하는 메타데이터 처리부, 상기 픽처를 인코딩하는 데이터 인코더 및 상기 인코딩된 픽처 및 상기 메타데이터에 대하여 저장 또는 전송을 위한 처리를 수행하는 전송 처리부를 포함하되, 상기 메타데이터는 오버레이 관련 메타데이터를 포함하고, 상기 오버레이 관련 메타데이터는 상기 오버레이의 그룹 정보를 포함하는 것을 특징으로 한다.

발명의 효과

- [0011] 본 발명에 따르면 지상파 방송망과 인터넷 망을 사용하는 차세대 하이브리드 방송을 지원하는 환경에서 VR 콘텐츠 (360 콘텐츠)를 효율적으로 전송할 수 있다.
- [0012] 본 발명에 따르면 사용자의 360 콘텐츠 소비에 있어서, 인터랙티브 경험(interactive experience)를 제공하기 위한 방안을 제안할 수 있다.

- [0013] 본 발명에 따르면 사용자의 360 콘텐츠 소비에 있어서, 360 콘텐츠 제작자가 의도하는 바가 정확히 반영되도록 시그널링 하는 방안을 제안할 수 있다.
- [0014] 본 발명에 따르면 360 콘텐츠 전달에 있어, 효율적으로 전송 캐패시터를 늘리고, 필요한 정보가 전달될 수 있도록 하는 방안을 제안할 수 있다.
- [0015] 본 발명에 따르면 360 비디오에 오버레이를 효율적으로 제공할 수 있고, 사용자 시각에 기반한 부가 정보를 효율적으로 표시할 수 있다.
- [0016] 본 발명에 따르면 360 비디오에 대한 오버레이를 통하여 특정 타겟과의 링크를 제공할 수 있다.
- [0017] 본 발명에 따르면 오버레이를 통하여 효율적으로 화면 전환 또는 부가 정보 제공을 위한 링크를 제공할 수 있다.
- [0018] 본 발명에 따르면 ISOBMFF(ISO base media file format) 등 ISO(International Organization for Standardization) 기반 미디어 파일 포맷을 통하여 효율적으로 360도 비디오 데이터에 대한 시그널링 정보를 저장 및 전송할 수 있다.
- [0019] 본 발명에 따르면 DASH(Dynamic Adaptive Streaming over HTTP) 등의 HTTP(HyperText Transfer Protocol) 기반 적응형 스트리밍을 통하여 360도 비디오 데이터에 대한 시그널링 정보를 전송할 수 있다.
- [0020] 본 발명에 따르면 SEI (Supplemental enhancement information) 메시지 혹은 VUI (Video Usability Information)를 통하여 360도 비디오 데이터에 대한 시그널링 정보를 저장 및 전송할 수 있고, 이를 통하여 전체적인 전송 효율을 향상시킬 수 있다.

도면의 간단한 설명

- [0021] 도 1은 본 발명에 따른 360 비디오 제공을 위한 전체 아키텍처를 도시한 도면이다.
- 도 2 및 도 3은 본 발명의 일 실시예에 따른 미디어 파일의 구조를 도시한 도면이다.
- 도 4는 DASH 기반 적응형 스트리밍 모델의 전반적인 동작의 일 예를 나타낸다.
- 도 5는 본 발명이 적용될 수 있는 360 비디오 전송 장치의 구성을 개략적으로 설명하는 도면이다.
- 도 6은 본 발명이 적용될 수 있는 360 비디오 수신 장치의 구성을 개략적으로 설명하는 도면이다.
- 도 7은 본 발명의 3D 공간을 설명하기 위한 비행기 주축(Aircraft Principal Axes) 개념을 도시한 도면이다.
- 도 8는 360 비디오의 처리 과정 및 프로젝션 포맷에 따른 리전별 패킹 과정이 적용된 2D 이미지를 예시적으로 나타낸다.
- 도 9a 내지 9b는 본 발명에 따른 프로젝션 포맷들을 예시적으로 나타낸다.
- 도 10a 및 10b는 본 발명의 일 실시예에 따른 타일(Tile)을 도시한 도면이다.
- 도 11은 본 발명의 일 실시예에 따른 360도 비디오 관련 메타데이터의 일 예를 나타낸다.
- 도 12는 뷰포인트, 뷰잉 포지션, 뷰잉 오리엔테이션의 개념을 개략적으로 나타낸다.
- 도 13은 본 발명에 따른 3DoF+ 비디오 제공을 위한 아키텍처의 예를 개략적으로 도시한 도면이다.
- 도 14a 및 14b는 3DoF+ 엔드 투 엔드 시스템 아키텍처의 예이다.
- 도 15는 FLUS(Framework for Live Uplink Streaming) 아키텍처의 예를 개략적으로 나타낸다.
- 도 16은 3DoF+ 송신단에서의 구성을 개략적으로 나타낸다.
- 도 17은 3DoF+ 수신단에서의 구성을 개략적으로 나타낸다.
- 도 18은 360 비디오의 오버레이를 나타낸 모습의 예이다.
- 도 19는 오버레이 미디어 트랙 상에 오버레이 메타데이터 시그널링의 예를 나타낸다.
- 도 20은 VR 미디어 파일 내에 오버레이 트랙의 구성을 나타낸 예이다.

- 도 21은 오버레이 미디어 트랙 상에 오버레이 메타데이터 시그널링의 다른 예를 나타낸다.
- 도 22는 파일 #1의 경우 가능한 4가지의 오버레이 미디어 패킹 구성을 나타낸 예이다.
- 도 23은 파일 #1의 경우 트랙 내에 구조를 나타낸 예이다.
- 도 24는 텍스처 아틀라스를 생성하는 방법의 순서도를 나타낸 예이다.
- 도 25는 텍스처 아틀라스를 생성하는 모습을 나타낸 예이다.
- 도 26은 VR 미디어의 리전별 패킹을 설명하기 위한 도면이다.
- 도 27은 오버레이 미디어의 리전별 패킹 방법의 순서도를 나타낸 예이다.
- 도 28은 오버레이 미디어의 리전별 패킹 모습을 나타낸 예이다.
- 도 29는 파일 #2인 경우 오버레이 미디어 패킹의 구성을 나타낸 예이다.
- 도 30은 파일 #2인 경우 VR 미디어 트랙이 VR 미디어의 일부분과 오버레이 미디어로 패킹되는 모습을 나타낸 예이다.
- 도 31은 파일 #2인 경우 VR 미디어 트랙이 VR 미디어와 오버레이 미디어로 패킹되는 모습을 나타낸 예이다.
- *도 32는 오버레이 프로젝션 지원 방법을 설명하는 순서도의 예이다.
- 도 33은 오버레이 미디어 패킹 및 프로젝션에 관한 메타데이터 시그널링 예를 나타낸다.
- 도 34는 오버레이 미디어 패킹 및 프로젝션에 관한 메타데이터 시그널링 다른 예를 나타낸다.
- 도 35a 및 도 35b는 VR 미디어 트랙 및 오버레이 미디어 트랙의 그룹핑 및 링킹 예를 나타낸다.
- 도 36은 파일 #1인 경우 오버레이 메타데이터 트랙을 나타낸 예이다.
- 도 37a 내지 도 37c는 오버레이를 배치할 위치를 나타낸 예이다.
- 도 38은 오버레이가 뷰포트 상에 배치되는 경우에 대한 예이다.
- 도 39는 오버레이가 스피어 상에 배치되는 경우에 대한 예이다.
- 도 40은 오버레이가 스피어 내부의 3차원 공간 상에 배치되는 경우에 대한 예이다.
- 도 41은 오버레이가 스피어 내부의 3차원 공간 상에 존재하는 경우, 오버레이의 위치/크기/회전을 나타낸다.
- 도 42는 오버레이 렌더링 속성의 예를 나타낸다.
- 도 43은 오버레이 미셀레이니어스(miscellaneous)의 예를 나타낸다.
- 도 44는 뷰포트 내에 이동 가능한 공간을 나타낸 예이다.
- 도 45는 VFC 알고리즘을 설명하기 위한 예이다.
- *도 46은 오버레이 인터렉션 제공 방법을 설명하는 순서도의 예이다.
- 도 47은 동적 오버레이 메타데이터의 구성 예를 나타낸다.
- 도 48은 동적 오버레이 메타데이터 트랙과 오버레이 미디어 트랙 링크 시그널링 예를 나타낸다.
- 도 49는 오버레이 메타데이터와 관련 오버레이 미디어의 링킹 예를 나타낸다.
- 도 50은 추천 뷰포트 오버레이 예를 나타낸다.
- 도 51은 'ovrc' 트랙 참조의 예를 나타낸다.
- 도 52은 메타데이터 트랙 그룹핑 예를 나타낸다.
- 도 53는 VR 미디어 상에 배치된 오버레이를 지원하는 송신기의 아키텍처 예를 나타낸다.
- 도 54은 VR 미디어 상에 배치된 오버레이를 지원하는 송신기의 아키텍처 예를 나타낸다.
- 도 55는 오버레이 미디어 트랙 상에 오버레이 메타데이터 시그널링의 또 다른 예를 나타낸다.

- 도 56은 오버레이 미디어 패킹, 프로젝션과 디폴트 렌더링 시그널링 예들을 나타낸다.
- 도 57은 오버레이 미디어 패킹, 프로젝션과 디폴트 렌더링 시그널링 다른 예들을 나타낸다.
- 도 58은 VR 미디어 트랙, 오버레이 미디어 트랙 및 오버레이 미디어 아이템의 그룹핑 예를 나타낸다.
- 도 59는 본 발명에 따른 360 비디오 전송 장치에 의한 360 비디오 데이터 처리 방법을 개략적으로 나타낸다.
- 도 60은 본 발명에 따른 360 비디오 수신 장치에 의한 360 비디오 데이터 처리 방법을 개략적으로 나타낸다.
- 도 61은 본 발명의 실시예들을 지원할 수 있는 장치를 예시적으로 나타낸다.
- 도 62는 본 발명의 기술적 특징이 적용될 수 있는 5G 사용 시나리오의 예를 나타낸다.
- 도 63은 본 발명의 일 실시 예에 따른 서비스 시스템을 나타낸다.

발명을 실시하기 위한 구체적인 내용

- [0022] 본 발명은 다양한 변경을 가할 수 있고 여러 가지 실시예를 가질 수 있는 바, 특정 실시예들을 도면에 예시하고 상세하게 설명하고자 한다. 그러나, 이는 본 발명을 특정 실시예에 한정하려고 하는 것이 아니다. 본 명세서에서 상용하는 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로, 본 발명의 기술적 사상을 한정하려는 의도로 사용되는 것은 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 명세서에서 "포함하다" 또는 "가지다" 등의 용어는 명세서상에 기재된 특징, 숫자, 단계, 동작, 구성 요소, 부품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성 요소, 부품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.
- [0023] 한편, 본 발명에서 설명되는 도면상의 각 구성들은 서로 다른 특징적인 기능들에 관한 설명의 편의를 위해 독립적으로 도시된 것으로서, 각 구성들이 서로 별개의 하드웨어나 별개의 소프트웨어로 구현된다는 것을 의미하지는 않는다. 예컨대, 각 구성 중 두 개 이상의 구성이 합쳐져 하나의 구성을 이룰 수도 있고, 하나의 구성이 복수의 구성으로 나뉘어질 수도 있다. 각 구성이 통합 및/또는 분리된 실시예도 본 발명의 본질에서 벗어나지 않는 한 본 발명의 권리범위에 포함된다.
- [0024] 이하, 첨부한 도면들을 참조하여, 본 발명의 바람직한 실시예를 보다 상세하게 설명하고자 한다. 이하, 도면상의 동일한 구성 요소에 대해서는 동일한 참조 부호를 사용하고 동일한 구성 요소에 대해서 중복된 설명은 생략될 수 있다.
- [0025] 도 1은 본 발명에 따른 360 비디오 제공을 위한 전체 아키텍처를 도시한 도면이다.
- [0026] 본 발명은 사용자에게 가상현실 (Virtual Reality, VR)을 제공하기 위하여, 360 콘텐츠를 제공하는 방안을 제안한다. VR이란 실제 또는 가상의 환경을 복제(replicates) 하기 위한 기술 내지는 그 환경을 의미할 수 있다. VR은 인공적으로 사용자에게 감각적 경험을 제공하며, 이를 통해 사용자는 전자적으로 프로젝션된 환경에 있는 것과 같은 경험을 할 수 있다.
- [0027] 360 콘텐츠는 VR을 구현, 제공하기 위한 콘텐츠 전반을 의미하며, 360 비디오 및/또는 360 오디오를 포함할 수 있다. 360 비디오는 VR을 제공하기 위해 필요한, 동시에 모든 방향(360도)으로 캡처되거나 재생되는 비디오 내지 이미지 콘텐츠를 의미할 수 있다. 이하, 360 비디오라 함은 360도 비디오를 의미할 수 있다. 360 비디오는 3D 모델에 따라 다양한 형태의 3D 공간 상에 나타내어지는 비디오 내지 이미지를 의미할 수 있으며, 예를 들어 360 비디오는 구형면(Spherical surface) 상에 나타내어질 수 있다. 360 오디오 역시 VR을 제공하기 위한 오디오 콘텐츠를로서, 음향 발생지가 3차원의 특정 공간상에 위치하는 것으로 인지될 수 있는, 공간적(Spatial) 오디오 콘텐츠를 의미할 수 있다. 360 콘텐츠는 생성, 처리되어 사용자들로 전송될 수 있으며, 사용자들은 360 콘텐츠를 이용하여 VR 경험을 소비할 수 있다. 360 비디오는 전방향(omnidirectional) 비디오라고 불릴 수 있고, 360 이미지는 전방향 이미지라고 불릴 수 있다.
- [0028] 본 발명은 특히 360 비디오를 효과적으로 제공하는 방안을 제안한다. 360 비디오를 제공하기 위하여, 먼저 하나 이상의 카메라를 통해 360 비디오가 캡처될 수 있다. 캡처된 360 비디오는 일련의 과정을 거쳐 전송되고, 수신 측에서는 수신된 데이터를 다시 원래의 360 비디오로 가공하여 렌더링할 수 있다. 이를 통해 360 비디오가 사용자에게 제공될 수 있다.
- [0029] 구체적으로 360 비디오 제공을 위한 전체의 과정은 캡처 과정(process), 준비 과정, 전송 과정, 프로세싱 과정,

렌더링 과정 및/또는 피드백 과정을 포함할 수 있다.

- [0030] 캡처 과정은 하나 이상의 카메라를 통하여 복수개의 시점 각각에 대한 이미지 또는 비디오를 캡처하는 과정을 의미할 수 있다. 캡처 과정에 의해 도시된 도 1의 (110)과 같은 이미지/비디오 데이터가 생성될 수 있다. 도시된 도 1의 (110)의 각 평면은 각 시점에 대한 이미지/비디오를 의미할 수 있다. 이 캡처된 복수개의 이미지/비디오를 로(raw) 데이터라 할 수도 있다. 캡처 과정에서 캡처와 관련된 메타데이터가 생성될 수 있다.
- [0031] 이 캡처를 위하여 VR 을 위한 특수한 카메라가 사용될 수 있다. 실시예에 따라 컴퓨터로 생성된 가상의 공간에 대한 360 비디오를 제공하고자 하는 경우, 실제 카메라를 통한 캡처가 수행되지 않을 수 있다. 이 경우 단순히 관련 데이터가 생성되는 과정으로 해당 캡처 과정이 같음될 수 있다.
- [0032] 준비 과정은 캡처된 이미지/비디오 및 캡처 과정에서 발생한 메타데이터를 처리하는 과정일 수 있다. 캡처된 이미지/비디오는 이 준비 과정에서, 스티칭 과정, 프로젝션 과정, 리전별 패킹 과정(Region-wise Packing) 및/또는 인코딩 과정 등을 거칠 수 있다.
- [0033] 먼저 각각의 이미지/비디오가 스티칭(Stitching) 과정을 거칠 수 있다. 스티칭 과정은 각각의 캡처된 이미지/비디오들을 연결하여 하나의 파노라마 이미지/비디오 또는 구형의 이미지/비디오를 만드는 과정일 수 있다.
- [0034] 이 후, 스티칭된 이미지/비디오는 프로젝션(Projection) 과정을 거칠 수 있다. 프로젝션 과정에서, 스티칭된 이미지/비디오는 2D 이미지 상에 프로젝션될 수 있다. 이 2D 이미지는 문맥에 따라 2D 이미지 프레임으로 불릴 수도 있다. 2D 이미지로 프로젝션하는 것을 2D 이미지로 매핑한다고 표현할 수도 있다. 프로젝션된 이미지/비디오 데이터는 도시된 도 1의 (120)과 같은 2D 이미지의 형태가 될 수 있다.
- [0035] 2D 이미지 상에 프로젝션된 비디오 데이터는 비디오 코딩 효율 등을 높이기 위하여 리전별 패킹 과정(Region-wise Packing)을 거칠 수 있다. 리전별 패킹이란, 2D 이미지 상에 프로젝션된 비디오 데이터를 리전(Region) 별로 나누어 처리를 가하는 과정을 의미할 수 있다. 여기서 리전(Region)이란, 360 비디오 데이터가 프로젝션된 2D 이미지가 나누어진 영역을 의미할 수 있다. 이 리전들은, 실시예에 따라, 2D 이미지를 균등하게 나누어 구분되거나, 임의로 나누어져 구분될 수 있다. 또한 실시예에 따라 리전들은, 프로젝션 스킴에 따라 구분될 수도 있다. 리전별 패킹 과정은 선택적(optional) 과정으로써, 준비 과정에서 생략될 수 있다.
- [0036] 실시예에 따라 이 처리 과정은, 비디오 코딩 효율을 높이기 위해, 각 리전을 회전한다거나 2D 이미지 상에서 재배열하는 과정을 포함할 수 있다. 예를 들어, 리전들을 회전하여 리전들의 특정 변들이 서로 근접하여 위치되도록 함으로써, 코딩 시의 효율이 높아지게 할 수 있다.
- [0037] 실시예에 따라 이 처리 과정은, 360 비디오상의 영역별로 레졸루션(resolution)을 차등화하기 위하여, 특정 리전에 대한 레졸루션을 높인다거나, 낮추는 과정을 포함할 수 있다. 예를 들어, 360 비디오 상에서 상대적으로 더 중요한 영역에 해당하는 리전들은, 다른 리전들보다 레졸루션을 높게할 수 있다. 2D 이미지 상에 프로젝션된 비디오 데이터 또는 리전별 패킹된 비디오 데이터는 비디오 코덱을 통한 인코딩 과정을 거칠 수 있다.
- [0038] 실시예에 따라 준비 과정은 부가적으로 에디팅(editing) 과정 등을 더 포함할 수 있다. 이 에디팅 과정에서 프로젝션 전후의 이미지/비디오 데이터들에 대한 편집 등이 더 수행될 수 있다. 준비 과정에서도 마찬가지로, 스티칭/프로젝션/인코딩/에디팅 등에 대한 메타데이터가 생성될 수 있다. 또한 2D 이미지 상에 프로젝션된 비디오 데이터들의 초기 시점, 혹은 ROI(Region of Interest) 등에 관한 메타데이터가 생성될 수 있다.
- [0039] 전송 과정은 준비 과정을 거친 이미지/비디오 데이터 및 메타데이터들을 처리하여 전송하는 과정일 수 있다. 전송을 위해 임의의 전송 프로토콜에 따른 처리가 수행될 수 있다. 전송을 위한 처리를 마친 데이터들은 방송망 및/또는 브로드밴드를 통해 전달될 수 있다. 이 데이터들은 온 디맨드(On Demand) 방식으로 수신측으로 전달될 수도 있다. 수신측에서는 다양한 경로를 통해 해당 데이터를 수신할 수 있다.
- [0040] 프로세싱 과정은 수신한 데이터를 디코딩하고, 프로젝션되어 있는 이미지/비디오 데이터를 3D 모델 상에 리-프로젝션(Re-projection)하는 과정을 의미할 수 있다. 이 과정에서 2D 이미지들 상에 프로젝션되어 있는 이미지/비디오 데이터가 3D 공간 상으로 리-프로젝션될 수 있다. 이 과정을 문맥에 따라 매핑, 프로젝션이라고 부를 수도 있다. 이 때 매핑되는 3D 공간은 3D 모델에 따라 다른 형태를 가질 수 있다. 예를 들어 3D 모델에는 구형(Sphere), 큐브(Cube), 실린더(Cylinder) 또는 피라미드(Pyramid)가 있을 수 있다.
- [0041] 실시예에 따라 프로세싱 과정은 부가적으로 에디팅(editing) 과정, 업 스케일링(up scaling) 과정 등을 더 포함할 수 있다. 이 에디팅 과정에서 리-프로젝션 전후의 이미지/비디오 데이터에 대한 편집 등이 더 수행될 수 있다. 이미지/비디오 데이터가 축소되어 있는 경우 업 스케일링 과정에서 샘플들의 업 스케일링을 통해 그 크기를

확대할 수 있다. 필요한 경우, 다운 스케일링을 통해 사이즈를 축소하는 작업이 수행될 수도 있다.

- [0042] 렌더링 과정은 3D 공간상에 리-프로젝션된 이미지/비디오 데이터를 렌더링하고 디스플레이하는 과정을 의미할 수 있다. 표현에 따라 리-프로젝션과 렌더링을 합쳐 3D 모델 상에 렌더링한다 라고 표현할 수도 있다. 3D 모델 상에 리-프로젝션된(또는 3D 모델 상으로 렌더링된) 이미지/비디오는 도시된 도 1의 (130)과 같은 형태를 가질 수 있다. 도시된 도 1의 (130)은 구형(Sphere)의 3D 모델에 리-프로젝션된 경우이다. 사용자는 VR 디스플레이 등을 통하여 렌더링된 이미지/비디오의 일부 영역을 볼 수 있다. 이 때 사용자가 보게되는 영역은 도시된 도 1의 (140)과 같은 형태일 수 있다.
- [0043] 피드백 과정은 디스플레이 과정에서 획득될 수 있는 다양한 피드백 정보들을 송신측으로 전달하는 과정을 의미할 수 있다. 피드백 과정을 통해 360 비디오 소비에 있어 인터랙티비티(Interactivity)가 제공될 수 있다. 실시예에 따라, 피드백 과정에서 헤드 오리엔테이션(Head Orientation) 정보, 사용자가 현재 보고 있는 영역을 나타내는 뷰포트(Viewport) 정보 등이 송신측으로 전달될 수 있다. 실시예에 따라, 사용자는 VR 환경 상에 구현된 것들과 상호작용할 수도 있는데, 이 경우 그 상호작용과 관련된 정보가 피드백 과정에서 송신측 내지 서비스 프로바이더 측으로 전달될 수도 있다. 실시예에 따라 피드백 과정은 수행되지 않을 수도 있다.
- [0044] 헤드 오리엔테이션 정보는 사용자의 머리 위치, 각도, 움직임 등에 대한 정보를 의미할 수 있다. 이 정보를 기반으로 사용자가 현재 360 비디오 내에서 보고 있는 영역에 대한 정보, 즉 뷰포트 정보가 계산될 수 있다.
- [0045] 뷰포트 정보는 현재 사용자가 360 비디오에서 보고 있는 영역에 대한 정보일 수 있다. 이를 통해 게이즈 분석(Gaze Analysis) 이 수행되어, 사용자가 어떠한 방식으로 360 비디오를 소비하는지, 360 비디오의 어느 영역을 얼마나 응시하는지 등을 확인할 수도 있다. 게이즈 분석은 수신측에서 수행되어 송신측으로 피드백 채널을 통해 전달될 수도 있다. VR 디스플레이 등의 장치는 사용자의 머리 위치/방향, 장치가 지원하는 수직(vertical) 혹은 수평(horizontal) FOV(Field Of View) 정보 등에 근거하여 뷰포트 영역을 추출할 수 있다.
- [0046] 실시예에 따라, 전송한 피드백 정보는 송신측으로 전달되는 것 뿐만아니라, 수신측에서 소비될 수도 있다. 즉, 전송한 피드백 정보를 이용하여 수신측의 디코딩, 리-프로젝션, 렌더링 과정 등이 수행될 수 있다. 예를 들어, 헤드 오리엔테이션 정보 및/또는 뷰포트 정보를 이용하여 현재 사용자가 보고 있는 영역에 대한 360 비디오만 우선적으로 디코딩 및 렌더링될 수도 있다.
- [0047] 여기서 뷰포트(viewport) 내지 뷰포트 영역이란, 사용자가 360 비디오에서 보고 있는 영역을 의미할 수 있다. 시점(viewpoint)은 사용자가 360 비디오에서 보고 있는 지점으로서, 뷰포트 영역의 정중앙 지점을 의미할 수 있다. 즉, 뷰포트는 시점을 중심으로 한 영역인데, 그 영역이 차지하는 크기 형태 등은 후술할 FOV(Field Of View)에 의해 결정될 수 있다.
- [0048] 전송한 360 비디오 제공을 위한 전체 아키텍처 내에서, 캡처/프로젝션/인코딩/전송/디코딩/리-프로젝션/렌더링의 일련의 과정을 거쳐게 되는 이미지/비디오 데이터들을 360 비디오 데이터라 부를 수 있다. 360 비디오 데이터라는 용어는 또한 이러한 이미지/비디오 데이터들과 관련되는 메타데이터 내지 시그널링 정보를 포함하는 개념으로 쓰일 수도 있다.
- [0049] 상술한 오디오 또는 비디오 등의 미디어 데이터를 저장하고 전송하기 위하여, 정형화된 미디어 파일 포맷이 정의될 수 있다. 실시예에 따라 미디어 파일은 ISO BMFF(ISO base media file format)를 기반으로 한 파일 포맷을 가질 수 있다.
- [0050] 도 2 및 도 3은 본 발명의 일 실시예에 따른 미디어 파일의 구조를 도시한 도면이다.
- [0051] 본 발명에 따른 미디어 파일은 적어도 하나 이상의 박스를 포함할 수 있다. 여기서 박스(box)는 미디어 데이터 또는 미디어 데이터에 관련된 메타데이터 등을 포함하는 데이터 블록 내지 오브젝트일 수 있다. 박스들은 서로 계층적 구조를 이룰 수 있으며, 이에 따라 데이터들이 분류되어 미디어 파일이 대용량 미디어 데이터의 저장 및/또는 전송에 적합한 형태를 띄게 될 수 있다. 또한 미디어 파일은, 사용자가 미디어 콘텐츠의 특정지점으로 이동하는 등, 미디어 정보에 접근하는데 있어 용이한 구조를 가질 수 있다.
- [0052] 본 발명에 따른 미디어 파일은 ftyp 박스, moov 박스 및/또는 mdat 박스를 포함할 수 있다.
- [0053] ftyp 박스(파일 타입 박스)는 해당 미디어 파일에 대한 파일 타입 또는 호환성 관련 정보를 제공할 수 있다. ftyp 박스는 해당 미디어 파일의 미디어 데이터에 대한 구성 버전 정보를 포함할 수 있다. 복호기는 ftyp 박스를 참조하여 해당 미디어 파일을 구분할 수 있다.

- [0054] moov 박스(무비 박스)는 해당 미디어 파일의 미디어 데이터에 대한 메타 데이터를 포함하는 박스일 수 있다. moov 박스는 모든 메타 데이터들을 위한 컨테이너 역할을 할 수 있다. moov 박스는 메타 데이터 관련 박스들 중 최상위 계층의 박스일 수 있다. 실시예에 따라 moov 박스는 미디어 파일 내에 하나만 존재할 수 있다.
- [0055] mdat 박스(미디어 데이터 박스)는 해당 미디어 파일의 실제 미디어 데이터들을 담는 박스일 수 있다. 미디어 데이터들은 오디오 샘플 및/또는 비디오 샘플들을 포함할 수 있는데, mdat 박스는 이러한 미디어 샘플들을 담는 컨테이너 역할을 할 수 있다.
- [0056] 실시예에 따라 전술한 moov 박스는 mvhd 박스, trak 박스 및/또는 mvex 박스 등을 하위 박스로서 더 포함할 수 있다.
- [0057] mvhd 박스(무비 헤더 박스)는 해당 미디어 파일에 포함되는 미디어 데이터의 미디어 프리젠테이션 관련 정보를 포함할 수 있다. 즉, mvhd 박스는 해당 미디어 프리젠테이션의 미디어 생성시간, 변경시간, 시간규격, 기간 등의 정보를 포함할 수 있다.
- [0058] trak 박스(트랙 박스)는 해당 미디어 데이터의 트랙에 관련된 정보를 제공할 수 있다. trak 박스는 오디오 트랙 또는 비디오 트랙에 대한 스트림 관련 정보, 프리젠테이션 관련 정보, 액세스 관련 정보 등의 정보를 포함할 수 있다. Trak 박스는 트랙의 개수에 따라 복수개 존재할 수 있다.
- [0059] trak 박스는 실시예에 따라 tkhd 박스(트랙 헤더 박스)를 하위 박스로서 더 포함할 수 있다. tkhd 박스는 trak 박스가 나타내는 해당 트랙에 대한 정보를 포함할 수 있다. tkhd 박스는 해당 트랙의 생성시간, 변경시간, 트랙 식별자 등의 정보를 포함할 수 있다.
- [0060] mvex 박스(무비 익스텐드 박스)는 해당 미디어 파일에 후술할 moof 박스가 있을 수 있음을 지시할 수 있다. 특정 트랙의 모든 미디어 샘플들을 알기 위해서, moof 박스들이 스캔되어야할 수 있다.
- [0061] 본 발명에 따른 미디어 파일은, 실시예에 따라, 복수개의 프래그먼트로 나뉘어질 수 있다(200). 이를 통해 미디어 파일이 분할되어 저장되거나 전송될 수 있다. 미디어 파일의 미디어 데이터들(mdat 박스)은 복수개의 프래그먼트로 나뉘어지고, 각각의 프래그먼트는 moof 박스와 나뉘어진 mdat 박스를 포함할 수 있다. 실시예에 따라 프래그먼트들을 활용하기 위해서는 ftyp 박스 및/또는 moov 박스의 정보가 필요할 수 있다.
- [0062] moof 박스(무비 프래그먼트 박스)는 해당 프래그먼트의 미디어 데이터에 대한 메타 데이터를 제공할 수 있다. moof 박스는 해당 프래그먼트의 메타데이터 관련 박스들 중 최상위 계층의 박스일 수 있다.
- [0063] mdat 박스(미디어 데이터 박스)는 전술한 바와 같이 실제 미디어 데이터를 포함할 수 있다. 이 mdat 박스는 각각의 해당 프래그먼트에 해당하는 미디어 데이터들의 미디어 샘플들을 포함할 수 있다.
- [0064] 실시예에 따라 전술한 moof 박스는 mfhd 박스 및/또는 traf 박스 등을 하위 박스로서 더 포함할 수 있다.
- [0065] mfhd 박스(무비 프래그먼트 헤더 박스)는 분할된 복수개의 프래그먼트들 간의 연관성과 관련한 정보들을 포함할 수 있다. mfhd 박스는 시퀀스 넘버(sequence number)를 포함하여, 해당 프래그먼트의 미디어 데이터가 분할된 몇 번째 데이터인지를 나타낼 수 있다. 또한, mfhd 박스를 이용하여 분할된 데이터 중 누락된 것은 없는지 여부가 확인될 수 있다.
- [0066] traf 박스(트랙 프래그먼트 박스)는 해당 트랙 프래그먼트에 대한 정보를 포함할 수 있다. traf 박스는 해당 프래그먼트에 포함되는 분할된 트랙 프래그먼트에 대한 메타데이터를 제공할 수 있다. traf 박스는 해당 트랙 프래그먼트 내의 미디어 샘플들이 복호화/재생될 수 있도록 메타데이터를 제공할 수 있다. traf 박스는 트랙 프래그먼트의 개수에 따라 복수개 존재할 수 있다.
- [0067] 실시예에 따라 전술한 traf 박스는 tfhd 박스 및/또는 trun 박스 등을 하위 박스로서 더 포함할 수 있다.
- [0068] tfhd 박스(트랙 프래그먼트 헤더 박스)는 해당 트랙 프래그먼트의 헤더 정보를 포함할 수 있다. tfhd 박스는 전술한 traf 박스가 나타내는 트랙 프래그먼트의 미디어 샘플들에 대하여, 기본적인 샘플크기, 기간, 오프셋, 식별자 등의 정보를 제공할 수 있다.
- [0069] trun 박스(트랙 프래그먼트 런 박스)는 해당 트랙 프래그먼트 관련 정보를 포함할 수 있다. trun 박스는 미디어 샘플별 기간, 크기, 재생시점 등과 같은 정보를 포함할 수 있다.
- [0070] 전술한 미디어 파일 내지 미디어 파일의 프래그먼트들은 세그먼트들로 처리되어 전송될 수 있다. 세그먼트에는 초기화 세그먼트(initialization segment) 및/또는 미디어 세그먼트(media segment) 가 있을 수 있다.

- [0071] 도시된 실시예(210)의 파일은, 미디어 데이터는 제외하고 미디어 디코더의 초기화와 관련된 정보 등을 포함하는 파일일 수 있다. 이 파일은 예를 들어 전술한 초기화 세그먼트에 해당할 수 있다. 초기화 세그먼트는 전술한 ftyp 박스 및/또는 moov 박스를 포함할 수 있다.
- [0072] 도시된 실시예(220)의 파일은, 전술한 프래그먼트를 포함하는 파일일 수 있다. 이 파일은 예를 들어 전술한 미디어 세그먼트에 해당할 수 있다. 미디어 세그먼트는 전술한 moof 박스 및/또는 mdat 박스를 포함할 수 있다. 또한, 미디어 세그먼트는 styp 박스 및/또는 sidx 박스를 더 포함할 수 있다.
- [0073] styp 박스(세그먼트 타입 박스)는 분할된 프래그먼트의 미디어 데이터를 식별하기 위한 정보를 제공할 수 있다. styp 박스는 분할된 프래그먼트에 대해, 전술한 ftyp 박스와 같은 역할을 수행할 수 있다. 실시예에 따라 styp 박스는 ftyp 박스와 동일한 포맷을 가질 수 있다.
- [0074] sidx 박스(세그먼트 인덱스 박스)는 분할된 프래그먼트에 대한 인덱스를 나타내는 정보를 제공할 수 있다. 이를 통해 해당 분할된 프래그먼트가 몇번째 프래그먼트인지가 지시될 수 있다.
- [0075] 실시예(230)에 따라 ssix 박스가 더 포함될 수 있는데, ssix 박스(서브 세그먼트 인덱스 박스)는 세그먼트가 서브 세그먼트로 더 나뉘어지는 경우에 있어, 그 서브 세그먼트의 인덱스를 나타내는 정보를 제공할 수 있다.
- [0076] 미디어 파일 내의 박스들은, 도시된 실시예(250)와 같은 박스 내지 풀 박스(FullBox) 형태를 기반으로, 더 확장된 정보들을 포함할 수 있다. 이 실시예에서 size 필드, largesize 필드는 해당 박스의 길이를 바이트 단위로 나타낼 수 있다. version 필드는 해당 박스 포맷의 버전을 나타낼 수 있다. Type 필드는 해당 박스의 타입 내지 식별자를 나타낼 수 있다. flags 필드는 해당 박스와 관련된 플래그 등을 나타낼 수 있다.
- [0077] 도 4는 DASH 기반 적응형 스트리밍 모델의 전반적인 동작의 일 예를 나타낸다. 도시된 실시예(400)에 따른 DASH 기반 적응형 스트리밍 모델은, HTTP 서버와 DASH 클라이언트 간의 동작을 기술하고 있다. 여기서 DASH(Dynamic Adaptive Streaming over HTTP)는, HTTP 기반 적응형 스트리밍을 지원하기 위한 프로토콜로서, 네트워크 상황에 따라 동적으로 스트리밍을 지원할 수 있다. 이에 따라 AV 콘텐츠 재생이 끊임없이 제공될 수 있다.
- [0078] 먼저 DASH 클라이언트는 MPD를 획득할 수 있다. MPD는 HTTP 서버 등의 서비스 프로바이더로부터 전달될 수 있다. DASH 클라이언트는 MPD에 기술된 세그먼트에의 접근 정보를 이용하여 서버로 해당 세그먼트들을 요청할 수 있다. 여기서 이 요청은 네트워크 상태를 반영하여 수행될 수 있다.
- [0079] DASH 클라이언트는 해당 세그먼트를 획득한 후, 이를 미디어 엔진에서 처리하여 화면에 디스플레이할 수 있다. DASH 클라이언트는 재생 시간 및/또는 네트워크 상황 등을 실시간으로 반영하여, 필요한 세그먼트를 요청, 획득할 수 있다(Adaptive Streaming). 이를 통해 콘텐츠가 끊임없이 재생될 수 있다.
- [0080] MPD(Media Presentation Description)는 DASH 클라이언트로 하여금 세그먼트를 동적으로 획득할 수 있도록 하기 위한 상세 정보를 포함하는 파일로서 XML 형태로 표현될 수 있다.
- [0081] DASH 클라이언트 컨트롤러(DASH Client Controller)는 네트워크 상황을 반영하여 MPD 및/또는 세그먼트를 요청하는 커맨드를 생성할 수 있다. 또한, 이 컨트롤러는 획득된 정보를 미디어 엔진 등등의 내부 블락에서 사용할 수 있도록 제어할 수 있다.
- [0082] MPD 파서(Parser)는 획득한 MPD를 실시간으로 파싱할 수 있다. 이를 통해, DASH 클라이언트 컨트롤러는 필요한 세그먼트를 획득할 수 있는 커맨드를 생성할 수 있게 될 수 있다.
- [0083] 세그먼트 파서(Parser)는 획득한 세그먼트를 실시간으로 파싱할 수 있다. 세그먼트에 포함된 정보들에 따라 미디어 엔진 등의 내부 블락들은 특정 동작을 수행할 수 있다.
- [0084] HTTP 클라이언트는 필요한 MPD 및/또는 세그먼트 등을 HTTP 서버에 요청할 수 있다. 또한 HTTP 클라이언트는 서버로부터 획득한 MPD 및/또는 세그먼트들을 MPD 파서 또는 세그먼트 파서로 전달할 수 있다.
- [0085] 미디어 엔진(Media Engine)은 세그먼트에 포함된 미디어 데이터를 이용하여 콘텐츠를 화면상에 표시할 수 있다. 이 때, MPD의 정보들이 활용될 수 있다.
- [0086] DASH 데이터 모델은 계층적 구조(410)를 가질 수 있다. 미디어 프리젠테이션은 MPD에 의해 기술될 수 있다. MPD는 미디어 프리젠테이션을 만드는 복수개의 구간(Period)들의 시간적인 시퀀스를 기술할 수 있다. 피리어드는 미디어 콘텐츠의 한 구간을 나타낼 수 있다.
- [0087] 한 구간에서, 데이터들은 어댑테이션 셋들에 포함될 수 있다. 어댑테이션 셋은 서로 교환될 수 있는 복수개의

미디어 콘텐츠 컴포넌트들의 집합일 수 있다. 어댑테이션은 레프리젠테이션들의 집합을 포함할 수 있다. 레프리젠테이션은 미디어 콘텐츠 컴포넌트에 해당할 수 있다. 한 레프리젠테이션 내에서, 콘텐츠는 복수개의 세그먼트들로 시간적으로 나뉘어질 수 있다. 이는 적절한 접근성과 전달(delivery)을 위함일 수 있다. 각각의 세그먼트에 접근하기 위해서 각 세그먼트의 URL이 제공될 수 있다.

- [0088] MPD는 미디어 프리젠테이션에 관련된 정보들을 제공할 수 있고, 피리어드 엘리먼트, 어댑테이션 셋 엘리먼트, 레프리젠테이션 엘리먼트는 각각 해당 피리어드, 어댑테이션 셋, 레프리젠테이션에 대해서 기술할 수 있다. 레프리젠테이션은 서브 레프리젠테이션들로 나뉘어질 수 있는데, 서브 레프리젠테이션 엘리먼트는 해당 서브 레프리젠테이션에 대해서 기술할 수 있다.
- [0089] 여기서 공통(Common) 속성/엘리먼트들이 정의될 수 있는데, 이들은 어댑테이션 셋, 레프리젠테이션, 서브 레프리젠테이션 등에 적용될 수 (포함될 수) 있다. 공통 속성/엘리먼트 중에는 에센셜 프로퍼티(EssentialProperty) 및/또는 서플멘탈 프로퍼티(SupplementalProperty)가 있을 수 있다.
- [0090] 에센셜 프로퍼티는 해당 미디어 프리젠테이션 관련 데이터를 처리함에 있어서 필수적이라고 여겨지는 엘리먼트들을 포함하는 정보일 수 있다. 서플멘탈 프로퍼티는 해당 미디어 프리젠테이션 관련 데이터를 처리함에 있어서 사용될 수도 있는 엘리먼트들을 포함하는 정보일 수 있다. 실시예에 따라 후술할 디스크립터들은, MPD를 통해 전달되는 경우, 에센셜 프로퍼티 및/또는 서플멘탈 프로퍼티 내에 정의되어 전달될 수 있다.
- [0091] 도 5는 본 발명이 적용될 수 있는 360 비디오 전송 장치의 구성을 개략적으로 설명하는 도면이다.
- [0092] 본 발명에 따른 360 비디오 전송 장치는 전송한 준비 과정 내지 전송 과정에 관련된 동작들을 수행할 수 있다. 360 비디오 전송 장치는 데이터 입력부, 스티처(Stitcher), 프로젝션 처리부, 리전별 패킹 처리부(도시되지 않음), 메타데이터 처리부, (송신측) 피드백 처리부, 데이터 인코더, 인캡슐레이션 처리부, 전송 처리부 및/또는 전송부를 내/외부 엘리먼트로서 포함할 수 있다.
- [0093] 데이터 입력부는 캡처된 각 시점별 이미지/비디오 들을 입력받을 수 있다. 이 시점별 이미지/비디오들은 하나 이상의 카메라들에 의해 캡처된 이미지/비디오들일 수 있다. 또한 데이터 입력부는 캡처 과정에서 발생된 메타데이터를 입력받을 수 있다. 데이터 입력부는 입력된 시점별 이미지/비디오들을 스티처로 전달하고, 캡처 과정의 메타데이터를 시그널링 처리부로 전달할 수 있다.
- [0094] 스티처는 캡처된 시점별 이미지/비디오들에 대한 스티칭 작업을 수행할 수 있다. 스티처는 스티칭된 360 비디오 데이터를 프로젝션 처리부로 전달할 수 있다. 스티처는 필요한 경우 메타데이터 처리부로부터 필요한 메타데이터를 전달받아 스티칭 작업에 이용할 수 있다. 스티처는 스티칭 과정에서 발생된 메타데이터를 메타데이터 처리부로 전달할 수 있다. 스티칭 과정의 메타데이터에는 스티칭이 수행되었는지 여부, 스티칭 타입 등의 정보들이 있을 수 있다.
- [0095] 프로젝션 처리부는 스티칭된 360 비디오 데이터를 2D 이미지 상에 프로젝션할 수 있다. 프로젝션 처리부는 다양한 스킴(scheme)에 따라 프로젝션을 수행할 수 있는데, 이에 대해서는 후술한다. 프로젝션 처리부는 각 시점별 360 비디오 데이터의 해당 텍스(depth)를 고려하여 매핑을 수행할 수 있다. 프로젝션 처리부는 필요한 경우 메타데이터 처리부로부터 프로젝션에 필요한 메타데이터를 전달받아 프로젝션 작업에 이용할 수 있다. 프로젝션 처리부는 프로젝션 과정에서 발생된 메타데이터를 메타데이터 처리부로 전달할 수 있다. 프로젝션 처리부의 메타데이터에는 프로젝션 스킴의 종류 등이 있을 수 있다.
- [0096] 리전별 패킹 처리부(도시되지 않음)는 전송한 리전별 패킹 과정을 수행할 수 있다. 즉, 리전별 패킹 처리부는 프로젝션된 360 비디오 데이터를 리전별로 나누고, 각 리전들을 회전, 재배열하거나, 각 리전의 레졸루션을 변경하는 등의 처리를 수행할 수 있다. 전송한 바와 같이 리전별 패킹 과정은 선택적(optional) 과정이며, 리전별 패킹이 수행되지 않는 경우, 리전별 패킹 처리부는 생략될 수 있다. 리전별 패킹 처리부는 필요한 경우 메타데이터 처리부로부터 리전별 패킹에 필요한 메타데이터를 전달받아 리전별 패킹 작업에 이용할 수 있다. 리전별 패킹 처리부는 리전별 패킹 과정에서 발생된 메타데이터를 메타데이터 처리부로 전달할 수 있다. 리전별 패킹 처리부의 메타데이터에는 각 리전의 회전 정도, 사이즈 등이 있을 수 있다.
- [0097] 전송한 스티처, 프로젝션 처리부 및/또는 리전별 패킹 처리부는 실시예에 따라 하나의 하드웨어 컴포넌트에서 수행될 수도 있다.
- [0098] 메타데이터 처리부는 캡처 과정, 스티칭 과정, 프로젝션 과정, 리전별 패킹 과정, 인코딩 과정, 인캡슐레이션 과정 및/또는 전송을 위한 처리 과정에서 발생할 수 있는 메타데이터들을 처리할 수 있다. 메타데이터 처리부는

이러한 메타데이터들을 이용하여 360 비디오 관련 메타데이터를 생성할 수 있다. 실시예에 따라 메타데이터 처리부는 360 비디오 관련 메타데이터를 시그널링 테이블의 형태로 생성할 수도 있다. 시그널링 문맥에 따라 360 비디오 관련 메타데이터는 메타데이터 또는 360 비디오 관련 시그널링 정보라 불릴 수도 있다. 또한 메타데이터 처리부는 획득하거나 생성한 메타데이터들을 필요에 따라 360 비디오 전송 장치의 내부 요소들에 전달할 수 있다. 메타데이터 처리부는 360 비디오 관련 메타데이터가 수신측으로 전송될 수 있도록 데이터 인코더, 인캡슐레이션 처리부 및/또는 전송 처리부에 전달할 수 있다.

- [0099] 데이터 인코더는 2D 이미지 상에 프로젝션된 360 비디오 데이터 및/또는 리전별 패키징된 360 비디오 데이터를 인코딩할 수 있다. 360 비디오 데이터는 다양한 포맷으로 인코딩될 수 있다.
- [0100] 인캡슐레이션 처리부는 인코딩된 360 비디오 데이터 및/또는 360 비디오 관련 메타데이터를 파일 등의 형태로 인캡슐레이션할 수 있다. 여기서 360 비디오 관련 메타데이터는 전송한 메타데이터 처리부로부터 전달받은 것일 수 있다. 인캡슐레이션 처리부는 해당 데이터들을 ISOBMFF, CFF 등의 파일 포맷으로 인캡슐레이션하거나, 기타 DASH 세그먼트 등의 형태로 처리할 수 있다. 인캡슐레이션 처리부는 실시예에 따라 360 비디오 관련 메타데이터를 파일 포맷 상에 포함시킬 수 있다. 360 비디오 관련 메타데이터는 예를 들어 ISOBMFF 파일 포맷 상의 다양한 레벨의 박스(box)에 포함되거나 파일 내에서 별도의 트랙내의 데이터로 포함될 수 있다. 실시예에 따라, 인캡슐레이션 처리부는 360 비디오 관련 메타데이터 자체를 파일로 인캡슐레이션할 수 있다.
- [0101] 전송 처리부는 파일 포맷에 따라 인캡슐레이션된 360 비디오 데이터에 전송을 위한 처리를 가할 수 있다. 전송 처리부는 임의의 전송 프로토콜에 따라 360 비디오 데이터를 처리할 수 있다. 전송을 위한 처리에는 방송망을 통한 전달을 위한 처리, 브로드밴드를 통한 전달을 위한 처리를 포함할 수 있다. 실시예에 따라 전송 처리부는 360 비디오 데이터뿐만 아니라, 메타데이터 처리부로부터 360 비디오 관련 메타데이터를 전달받아, 이 것에 전송을 위한 처리를 가할 수도 있다.
- [0102] 전송부는 전송 처리된 360 비디오 데이터 및/또는 360 비디오 관련 메타데이터를 방송망 및/또는 브로드밴드를 통해 전송할 수 있다. 전송부는 방송망을 통한 전송을 위한 요소 및/또는 브로드밴드를 통한 전송을 위한 요소를 포함할 수 있다.
- [0103] 본 발명에 따른 360 비디오 전송 장치의 일 실시예에 의하면, 360 비디오 전송 장치는 데이터 저장부(도시되지 않음)를 내/외부 요소로서 더 포함할 수 있다. 데이터 저장부는 인코딩된 360 비디오 데이터 및/또는 360 비디오 관련 메타데이터를 전송 처리부로 전달하기 전에 저장하고 있을 수 있다. 이 데이터들이 저장되는 형태는 ISOBMFF 등의 파일 형태일 수 있다. 실시간으로 360 비디오를 전송하는 경우에는 데이터 저장부가 필요하지 않을 수 있으나, 온 디맨드, NRT(Non Real Time), 브로드밴드 등을 통해 전달하는 경우에는 인캡슐레이션된 360 데이터가 데이터 저장부에 일정 기간 저장되었다가 전송될 수도 있다.
- [0104] 본 발명에 따른 360 비디오 전송 장치의 다른 실시예에 의하면, 360 비디오 전송 장치는 (송신측) 피드백 처리부 및/또는 네트워크 인터페이스(도시되지 않음)를 내/외부 요소로서 더 포함할 수 있다. 네트워크 인터페이스는 본 발명에 따른 360 비디오 수신 장치로부터 피드백 정보를 전달받고, 이를 송신측 피드백 처리부로 전달할 수 있다. 송신측 피드백 처리부는 피드백 정보를 스티처, 프로젝션 처리부, 리전별 패키징 처리부, 데이터 인코더, 인캡슐레이션 처리부, 메타데이터 처리부 및/또는 전송 처리부로 전달할 수 있다. 실시예에 따라 피드백 정보는 메타데이터 처리부에 일단 전달된 후, 다시 각 내부 요소들로 전달될 수 있다. 피드백 정보를 전달받은 내부 요소들은 이 후의 360 비디오 데이터의 처리에 피드백 정보를 반영할 수 있다.
- [0105] 본 발명에 따른 360 비디오 전송 장치의 또 다른 실시예에 의하면, 리전별 패키징 처리부는 각 리전을 회전하여 2D 이미지 상에 매핑할 수 있다. 이 때 각 리전들은 서로 다른 방향, 서로 다른 각도로 회전되어 2D 이미지 상에 매핑될 수 있다. 리전의 회전은 360 비디오 데이터가 구형의 면 상에서 프로젝션 전에 인접했던 부분, 스티칭된 부분 등을 고려하여 수행될 수 있다. 리전의 회전에 관한 정보들, 즉 회전 방향, 각도 등은 360 비디오 관련 메타데이터에 의해 시그널링될 수 있다. 본 발명에 따른 360 비디오 전송 장치의 또 다른 실시예에 의하면, 데이터 인코더는 각 리전 별로 다르게 인코딩을 수행할 수 있다. 데이터 인코더는 특정 리전은 높은 퀄리티로, 다른 리전은 낮은 퀄리티로 인코딩을 수행할 수 있다. 송신측 피드백 처리부는 360 비디오 수신 장치로부터 전달받은 피드백 정보를 데이터 인코더로 전달하여, 데이터 인코더가 리전별 차등화된 인코딩 방법을 사용하도록 할 수 있다. 예를 들어 송신측 피드백 처리부는 수신측으로부터 전달받은 뷰포트 정보를 데이터 인코더로 전달할 수 있다. 데이터 인코더는 뷰포트 정보가 지시하는 영역을 포함하는 리전들에 대해 다른 리전들보다 더 높은 퀄리티(UHD 등)로 인코딩을 수행할 수 있다.

- [0106] 본 발명에 따른 360 비디오 전송 장치의 또 다른 실시예에 의하면, 전송 처리부는 각 리전 별로 다르게 전송을 위한 처리를 수행할 수 있다. 전송 처리부는 리전 별로 다른 전송 파라미터(모듈레이션 오더, 코드 레이트 등)를 적용하여, 각 리전 별로 전달되는 데이터의 강건성(robustness)을 다르게 할 수 있다.
- [0107] 이 때, 송신측 피드백 처리부는 360 비디오 수신 장치로부터 전달받은 피드백 정보를 전송 처리부로 전달하여, 전송 처리부가 리전별 차등화된 전송 처리를 수행하도록 할 수 있다. 예를 들어 송신측 피드백 처리부는 수신측 으로부터 전달받은 뷰포트 정보를 전송 처리부로 전달할 수 있다. 전송 처리부는 해당 뷰포트 정보가 지시하는 영역을 포함하는 리전들에 대해 다른 리전들보다 더 높은 강건성을 가지도록 전송 처리를 수행할 수 있다.
- [0108] 전술한 본 발명에 따른 360 비디오 전송 장치의 내/외부 엘리먼트들은 하드웨어로 구현되는 하드웨어 엘리먼트 들일 수 있다. 실시예에 따라 내/외부 엘리먼트들은 변경, 생략되거나 다른 엘리먼트로 대체, 통합될 수 있다. 실시예에 따라 부가 엘리먼트들이 360 비디오 전송 장치에 추가될 수도 있다.
- [0109] 도 6은 본 발명이 적용될 수 있는 360 비디오 수신 장치의 구성을 개략적으로 설명하는 도면이다.
- [0110] 본 발명에 따른 360 비디오 수신 장치는 전술한 프로세싱 과정 및/또는 렌더링 과정에 관련된 동작들을 수행할 수 있다. 360 비디오 수신 장치는 수신부, 수신 처리부, 디캡슐레이션 처리부, 데이터 디코더, 메타데이터 파서, (수신측) 피드백 처리부, 리-프로젝션 처리부 및/또는 렌더러를 내/외부 엘리먼트로서 포함할 수 있다. 한편, 시그널링 파서는 메타데이터 파서라고 불릴 수 있다.
- [0111] 수신부는 본 발명에 따른 360 비디오 전송 장치가 전송한 360 비디오 데이터를 수신할 수 있다. 전송되는 채널 에 따라 수신부는 방송망을 통하여 360 비디오 데이터를 수신할 수도 있고, 브로드밴드를 통하여 360 비디오 데이터를 수신할 수도 있다.
- [0112] 수신 처리부는 수신된 360 비디오 데이터에 대해 전송 프로토콜에 따른 처리를 수행할 수 있다. 전송측에서 전송을 위한 처리가 수행된 것에 대응되도록, 수신 처리부는 전술한 전송 처리부의 역과정을 수행할 수 있다. 수신 처리부는 획득한 360 비디오 데이터는 디캡슐레이션 처리부로 전달하고, 획득한 360 비디오 관련 메타데이터 는 메타데이터 파서로 전달할 수 있다. 수신 처리부가 획득하는 360 비디오 관련 메타데이터는 시그널링 데이터 의 형태일 수 있다.
- [0113] 디캡슐레이션 처리부는 수신 처리부로부터 전달받은 파일 형태의 360 비디오 데이터를 디캡슐레이션할 수 있다. 디캡슐레이션 처리부는 ISOBMFF 등에 따른 파일들을 디캡슐레이션하여, 360 비디오 데이터 내지 360 비디오 관련 메타데이터를 획득할 수 있다. 획득된 360 비디오 데이터는 데이터 디코더로, 획득된 360 비디오 관련 메타 데이터는 메타데이터 파서로 전달할 수 있다. 디캡슐레이션 처리부가 획득하는 360 비디오 관련 메타데이터는 파일 포맷 내의 박스 혹은 트랙 형태일 수 있다. 디캡슐레이션 처리부는 필요한 경우 메타데이터 파서로부터 디 캡슐레이션에 필요한 메타데이터를 전달받을 수도 있다.
- [0114] 데이터 디코더는 360 비디오 데이터에 대한 디코딩을 수행할 수 있다. 데이터 디코더는 메타데이터 파서로부터 디코딩에 필요한 메타데이터를 전달받을 수도 있다. 데이터 디코딩 과정에서 획득된 360 비디오 관련 메타데이터 는 메타데이터 파서로 전달될 수도 있다.
- [0115] 메타데이터 파서는 360 비디오 관련 메타데이터에 대한 파싱/디코딩을 수행할 수 있다. 메타데이터 파서는 획득 한 메타데이터를 데이터 디캡슐레이션 처리부, 데이터 디코더, 리-프로젝션 처리부 및/또는 렌더러로 전달할 수 있다.
- [0116] 리-프로젝션 처리부는 디코딩된 360 비디오 데이터에 대하여 리-프로젝션을 수행할 수 있다. 리-프로젝션 처리 부는 360 비디오 데이터를 3D 공간으로 리-프로젝션할 수 있다. 3D 공간은 사용되는 3D 모델에 따라 다른 형태 를 가질 수 있다. 리-프로젝션 처리부는 메타데이터 파서로부터 리-프로젝션에 필요한 메타데이터를 전달받을 수도 있다. 예를 들어 리-프로젝션 처리부는 사용되는 3D 모델의 타입 및 그 세부 정보에 대한 정보를 메타데이 터 파서로부터 전달받을 수 있다. 실시예에 따라 리-프로젝션 처리부는 리-프로젝션에 필요한 메타데이터를 이 용하여, 3D 공간 상의 특정 영역에 해당하는 360 비디오 데이터만을 3D 공간으로 리-프로젝션할 수도 있다.
- [0117] 렌더러는 리-프로젝션된 360 비디오 데이터를 렌더링할 수 있다. 전술한 바와 같이 360 비디오 데이터가 3D 공 간상에 렌더링된다고 표현할 수도 있는데, 이처럼 두 과정이 한번에 일어나는 경우 리-프로젝션 처리부와 렌더 러는 통합되어, 렌더러에서 이 과정들이 모두 진행될 수 있다. 실시예에 따라 렌더러는 사용자의 시점 정보에 따라 사용자가 보고 있는 부분만을 렌더링할 수도 있다.
- [0118] 사용자는 VR 디스플레이 등을 통하여 렌더링된 360 비디오의 일부 영역을 볼 수 있다. VR 디스플레이는 360 비

디오를 재생하는 장치로서, 360 비디오 수신 장치에 포함될 수도 있고(tethered), 별도의 장치로서 360 비디오 수신 장치에 연결될 수도 있다(un-tethered).

- [0119] 본 발명에 따른 360 비디오 수신 장치의 일 실시예에 의하면, 360 비디오 수신 장치는 (수신측) 피드백 처리부 및/또는 네트워크 인터페이스(도시되지 않음)를 내/외부 요소로서 더 포함할 수 있다. 수신측 피드백 처리부는 렌더러, 리-프로젝션 처리부, 데이터 디코더, 디캡슐레이션 처리부 및/또는 VR 디스플레이로부터 피드백 정보를 획득하여 처리할 수 있다. 피드백 정보는 뷰포트 정보, 헤드 오리엔테이션 정보, 게이즈(Gaze) 정보 등을 포함할 수 있다. 네트워크 인터페이스는 피드백 정보를 수신측 피드백 처리부로부터 전달받고, 이를 360 비디오 전송 장치로 전송할 수 있다.
- [0120] 전술한 바와 같이, 피드백 정보는 송신측으로 전달되는 것 뿐아니라, 수신측에서 소비될 수도 있다. 수신측 피드백 처리부는 획득한 피드백 정보를 360 비디오 수신 장치의 내부 요소들로 전달하여, 렌더링 등의 과정에 반영되게 할 수 있다. 수신측 피드백 처리부는 피드백 정보를 렌더러, 리-프로젝션 처리부, 데이터 디코더 및/또는 디캡슐레이션 처리부로 전달할 수 있다. 예를 들어, 렌더러는 피드백 정보를 활용하여 사용자가 보고 있는 영역을 우선적으로 렌더링할 수 있다. 또한 디캡슐레이션 처리부, 데이터 디코더 등은 사용자가 보고 있는 영역 내지 보게될 영역을 우선적으로 디캡슐레이션, 디코딩할 수 있다.
- [0121] 전술한 본 발명에 따른 360 비디오 수신 장치의 내/외부 요소들은 하드웨어로 구현되는 하드웨어 요소들일 수 있다. 실시예에 따라 내/외부 요소들은 변경, 생략되거나 다른 요소로 대체, 통합될 수 있다. 실시예에 따라 부가 요소들이 360 비디오 수신 장치에 추가될 수도 있다.
- [0122] 본 발명의 또 다른 관점은 360 비디오를 전송하는 방법 및 360 비디오를 수신하는 방법과 관련될 수 있다. 본 발명에 따른 360 비디오를 전송/수신하는 방법은, 각각 전술한 본 발명에 따른 360 비디오 전송/수신 장치 또는 그 장치의 실시예들에 의해 수행될 수 있다.
- [0123] 전술한 본 발명에 따른 360 비디오 전송/수신 장치, 전송/수신 방법의 각각의 실시예 및 그 내/외부 요소들 각각의 실시예들을 서로 조합될 수 있다. 예를 들어 프로젝션 처리부의 실시예들과, 데이터 인코더의 실시예들은 서로 조합되어, 그 경우의 수만큼의 360 비디오 전송 장치의 실시예들을 만들어 낼 수 있다. 이렇게 조합된 실시예들 역시 본 발명의 범위에 포함된다.
- [0124] 도 7은 본 발명의 3D 공간을 설명하기 위한 비행기 주축(Aircraft Principal Axes) 개념을 도시한 도면이다. 본 발명에서, 3D 공간에서의 특정 지점, 위치, 방향, 간격, 영역 등을 표현하기 위하여 비행기 주축 개념이 사용될 수 있다. 즉, 본 발명에서 프로젝션 전 또는 리-프로젝션 후의 3D 공간에 대해 기술하고, 그에 대한 시그널링을 수행하기 위하여 비행기 주축 개념이 사용될 수 있다. 실시예에 따라 X, Y, Z 축 개념 또는 구형 좌표계를 이용한 방법이 사용될 수도 있다.
- [0125] 비행기는 3 차원으로 자유롭게 회전할 수 있다. 3차원을 이루는 축을 각각 피치(pitch) 축, 요(yaw) 축 및 롤(roll) 축이라고 한다. 본 명세서에서 이 들을 줄여서 pitch, yaw, roll 내지 pitch 방향, yaw 방향, roll 방향이라고 표현할 수도 있다.
- [0126] Pitch 축은 비행기의 앞코가 위/아래로 회전하는 방향의 기준이 되는 축을 의미할 수 있다. 도시된 비행기 주축 개념에서 pitch 축은 비행기의 날개에서 날개로 이어지는 축을 의미할 수 있다.
- [0127] Yaw 축은 비행기의 앞코가 좌/우로 회전하는 방향의 기준이 되는 축을 의미할 수 있다. 도시된 비행기 주축 개념에서 yaw 축은 비행기의 위에서 아래로 이어지는 축을 의미할 수 있다. Roll 축은 도시된 비행기 주축 개념에서 비행기의 앞코에서 꼬리로 이어지는 축으로서, roll 방향의 회전이란 roll 축을 기준으로 한 회전을 의미할 수 있다. 전술한 바와 같이, pitch, yaw, roll 개념을 통해 본 발명에서의 3D 공간이 기술될 수 있다.
- [0128] 한편, 상술한 내용과 같이 2D 이미지 상에 프로젝션된 비디오 데이터는 비디오 코딩 효율 등을 높이기 위하여 리전별 패킹 과정(Region-wise Packing)이 수행될 수 있다. 상기 리전별 패킹 과정은 2D 이미지 상에 프로젝션된 비디오 데이터를 리전(Region) 별로 나누어 처리를 가하는 과정을 의미할 수 있다. 상기 리전(Region)은 360 비디오 데이터가 프로젝션된 2D 이미지가 나누어진 영역을 나타낼 수 있고, 상기 2D 이미지가 나뉘어진 리전들은 프로젝션 스킴에 따라 구분될 수도 있다. 여기서, 상기 2D 이미지는 비디오 프레임(video frame) 또는 프레임(frame)이라고 불릴 수 있다.
- [0129] 이와 관련하여 본 발명에서는 프로젝션 스킴에 따른 상기 리전별 패킹 과정에 대한 메타데이터들 및 상기 메타데이터들의 시그널링 방법을 제안한다. 상기 메타데이터들을 기반으로 상기 리전별 패킹 과정은 보다 효율적으

로 수행될 수 있다.

[0130] 도 8는 360 비디오의 처리 과정 및 프로젝션 포맷에 따른 리전별 패킹 과정이 적용된 2D 이미지를 예시적으로 나타낸다. 도 8의 (a)는 입력된 360 비디오 데이터의 처리 과정을 나타낼 수 있다. 도 8의 (a)를 참조하면 입력된 시점의 360 비디오 데이터는 다양한 프로젝션 스킴에 따라서 3D 프로젝션 구조에 스티칭 및 프로젝션될 수 있고, 상기 3D 프로젝션 구조에 프로젝션된 360 비디오 데이터는 2D 이미지로 나타낼 수 있다. 즉, 상기 360 비디오 데이터는 스티칭될 수 있고, 상기 2D 이미지로 프로젝션될 수 있다. 상기 360 비디오 데이터가 프로젝션된 2D 이미지는 프로젝션된 프레임(projected frame)이라고 나타낼 수 있다. 또한, 상기 프로젝션된 프레임은 전술한 리전별 패킹 과정이 수행될 수 있다. 즉, 상기 프로젝션된 프레임 상의 프로젝션된 360 비디오 데이터를 포함하는 영역을 리전들로 나누고, 각 리전들을 회전, 재배열하거나, 각 리전의 레졸루션을 변경하는 등의 처리가 수행될 수 있다. 다시 말해, 상기 리전별 패킹 과정은 상기 프로젝션된 프레임을 하나 이상의 패킹된 프레임(packed frame)으로 맵핑하는 과정을 나타낼 수 있다. 상기 리전별 패킹 과정의 수행은 선택적(optional)일 수 있고, 상기 리전별 패킹 과정이 적용되지 않는 경우, 상기 패킹된 프레임과 상기 프로젝션된 프레임은 동일할 수 있다. 상기 리전별 패킹 과정이 적용되는 경우, 상기 프로젝션된 프레임의 각 리전은 상기 패킹된 프레임의 리전에 맵핑될 수 있고, 상기 프로젝션된 프레임의 각 리전이 맵핑되는 상기 패킹된 프레임의 리전의 위치, 모양 및 크기를 나타내는 메타데이터가 도출될 수 있다.

[0131] 도 8의 (b) 및 (c)는 상기 프로젝션된 프레임의 각 리전이 상기 패킹된 프레임의 리전에 맵핑되는 예들을 나타낼 수 있다. 도 8의 (b)를 참조하면 상기 360 비디오 데이터는 파노라믹(panoramic) 프로젝션 스킴(projection scheme)에 따라서 2D 이미지(또는 프레임)에 프로젝션될 수 있다. 상기 프로젝션된 프레임의 상단면(top) 리전, 중단면(middle) 리전 및 하단면(bottom) 리전은 리전별 패킹 과정이 적용되어 우측의 도면과 같이 재배열될 수 있다. 여기서, 상기 상단면 리전은 2D 이미지 상에서 상기 파노라마의 상단면을 나타내는 리전(region)일 수 있고, 상기 중단면 리전은 2D 이미지 상에서 상기 파노라마의 중단면을 나타내는 리전일 수 있고, 상기 하단면 리전은 2D 이미지 상에서 상기 파노라마의 하단면을 나타내는 리전일 수 있다. 또한, 도 8의 (c)를 참조하면 상기 360 비디오 데이터는 큐빅(cubic) 프로젝션 스킴에 따라서 2D 이미지(또는 프레임)에 프로젝션될 수 있다. 상기 프로젝션된 프레임의 앞면(front) 리전, 뒷면(back) 리전, 윗면(top) 리전, 바닥면(bottom) 리전, 우측옆면(right) 리전 및 좌측옆면(left) 리전은 리전별 패킹 과정이 적용되어 우측의 도면과 같이 재배열될 수 있다. 여기서, 상기 앞면 리전은 2D 이미지 상에서 상기 큐브의 앞면을 나타내는 리전(region)일 수 있고, 상기 뒷면 리전은 2D 이미지 상에서 상기 큐브의 뒷면을 나타내는 리전일 수 있다. 또한, 여기서, 상기 윗면 리전은 2D 이미지 상에서 상기 큐브의 윗면을 나타내는 리전일 수 있고, 상기 바닥면 리전은 2D 이미지 상에서 상기 큐브의 바닥면을 나타내는 리전일 수 있다. 또한, 여기서, 상기 우측옆면 리전은 2D 이미지 상에서 상기 큐브의 우측옆면을 나타내는 리전일 수 있고, 상기 좌측옆면 리전은 2D 이미지 상에서 상기 큐브의 좌측옆면을 나타내는 리전일 수 있다.

[0132] 도 8의 (d)는 상기 360 비디오 데이터가 프로젝션될 수 있는 다양한 3D 프로젝션 포맷들을 나타낼 수 있다. 도 8의 (d)를 참조하면 상기 3D 프로젝션 포맷들은 사면체(tetrahedron), 큐브(cube), 팔면체(octahedron), 이십면체(dodecahedron), 이십면체(icosahedron)를 포함할 수 있다. 도 8의 (d)에 도시된 2D 프로젝션(2D projection)들은 상기 3D 프로젝션 포맷에 프로젝션된 360 비디오 데이터를 2D 이미지로 나타낸 프로젝션된 프레임(projected frame)들을 나타낼 수 있다.

[0133] 상기 프로젝션 포맷들은 예시로서, 본 발명에 따르면 다음과 다양한 프로젝션 포맷(또는 프로젝션 스킴)들 중 일부 또는 전부가 사용될 수 있다. 360 비디오에 대하여 어떤 프로젝션 포맷이 사용되었는지는 예를 들어 메타데이터의 프로젝션 포맷 필드를 통하여 지시될 수 있다.

[0134] 도 9a 내지 9b는 본 발명에 따른 프로젝션 포맷들을 예시적으로 나타낸다.

[0135] 도 9a의 (a)는 등정방향 프로젝션 포맷을 나타낼 수 있다. 등정방향 프로젝션 포맷이 사용되는 경우, 구형 면 상의 $(r, \theta_0, 0)$ 즉, $\theta = \theta_0, \phi = 0$ 인 점과 2D 이미지의 중앙 픽셀이 매핑될 수 있다. 또한, 앞면 카메라(front camera)의 주점(principal point)를 구형 면의 $(r, 0, 0)$ 지점으로 가정할 수 있다. 또한, $\phi_0 = 0$ 으로 고정될 수 있다. 따라서, XY 좌표계로 변환된 값 (x, y) 는 다음의 수학적식을 통하여 2D 이미지 상에 (X, Y) 픽셀로 변환될 수 있다.

수학식 1

$$X = K_x * x + X_O = K_x * (\theta - \theta_0) * r + X_O$$

$$Y = -K_y * y - Y_O$$

[0136]

[0137]

또한, 2D 이미지의 좌상단 픽셀을 XY 좌표계의 (0,0)에 위치시키는 경우, x축에 대한 오프셋 값 및 y축에 대한 오프셋 값은 다음의 수학식을 통하여 나타낼 수 있다.

수학식 2

$$X_O = K_x * \pi * r$$

$$Y_O = -K_y * \pi / 2 * r$$

[0138]

[0139]

이를 이용하여 XY 좌표계로의 변환식을 다시 쓰면 다음과 같을 수 있다.

수학식 3

$$X = K_x * x + X_O = K_x * (\pi + \theta - \theta_0) * r$$

$$Y = -K_y * y - Y_O = K_y * (\pi / 2 - \phi) * r$$

[0140]

[0141]

예를 들어 $\theta_0 = 0$ 인 경우, 즉 2D 이미지의 중앙 픽셀이 구형 면 상의 $\theta = 0$ 인 데이터를 가리키는 경우, 구형 면은 (0,0)을 기준으로 2D 이미지 상에서 가로길이(width) = $2K_x \pi r$ 이고 세로길이(height) = $K_y \pi r$ 인 영역에 매핑될 수 있다. 구형 면 상에서 $\phi = \pi / 2$ 인 데이터는 2D 이미지 상의 윗쪽 변 전체에 매핑될 수 있다. 또한, 구형 면 상에서 (r, $\pi / 2$, 0) 인 데이터는 2D 이미지 상의 ($3\pi K_x r / 2$, $\pi K_y r / 2$) 인 점에 매핑될 수 있다.

[0142]

수신 측에서는, 2D 이미지 상의 360 비디오 데이터를 구형 면 상으로 리-프로젝션할 수 있다. 이를 변환식으로 쓰면 다음의 수학식과 같을 수 있다.

수학식 4

$$\theta = \theta_0 + X / K_x * r - \pi$$

$$\phi = \pi / 2 - Y / K_y * r$$

[0143]

[0144]

예를 들어 2D 이미지 상에서 XY 좌표값이 ($K_x \pi r$, 0) 인 픽셀은 구형 면 상의 $\theta = \theta_0$, $\phi = \pi / 2$ 인 점으로 리-프로젝션될 수 있다.

[0145]

도 9a의 (b)는 큐브 프로젝션 포맷을 나타낼 수 있다. 예를 들어 스티칭된 360 비디오 데이터는 구형의 면 상에 나타내어질 수 있다. 프로젝션 처리부는 이러한 360 비디오 데이터를 큐브(Cube, 정육면체) 형태로 나누어 2D 이미지 상에 프로젝션할 수 있다. 구형의 면 상의 360 비디오 데이터는 큐브의 각 면에 대응되어, 2D 이미지 상에 도 9a의 (b) 좌측 또는 (b) 우측에 도시된 것과 같이 프로젝션될 수 있다.

- [0146] 도 9a의 (c)는 실린더형 프로젝션 포맷을 나타낼 수 있다. 스티칭된 360 비디오 데이터가 구형의 면 상에 나타내어질 수 있다고 가정할 때, 프로젝션 처리부는 이러한 360 비디오 데이터를 실린더(Cylinder) 형태로 나누어 2D 이미지 상에 프로젝션할 수 있다. 구형의 면 상의 360 비디오 데이터는 실린더의 옆면(side)과 윗면(top), 바닥면(bottom)에 각각 대응되어, 2D 이미지 상에 도 8A의 (c) 좌측 또는 (c) 우측에 도시된 것과 같이 프로젝션될 수 있다.
- [0147] 도 9a의 (d)는 타일-기반 프로젝션 포맷을 나타낼 수 있다. 타일-기반(Tile-based) 프로젝션 스킴이 쓰이는 경우, 전술한 프로젝션 처리부는 구형 면 상의 360 비디오 데이터를, 도 9a의 (d)에 도시된 것과 같이 하나 이상의 세부 영역으로 나누어 2D 이미지 상에 프로젝션할 수 있다. 상기 세부 영역은 타일이라고 불릴 수 있다.
- [0148] 도 9b의 (e)는 피라미드 프로젝션 포맷을 나타낼 수 있다. 스티칭된 360 비디오 데이터가 구형의 면 상에 나타내어질 수 있다고 가정할 때, 프로젝션 처리부는 이러한 360 비디오 데이터를 피라미드 형태로 보고, 각 면을 나누어 2D 이미지 상에 프로젝션할 수 있다. 구형의 면 상의 360 비디오 데이터는 피라미드의 바닥면(front), 피라미드의 4방향의 옆면(Left top, Left bottom, Right top, Right bottom)에 각각 대응되어, 2D 이미지 상에 도 8의 (e) 좌측 또는 (e) 우측에 도시된 것과 같이 프로젝션될 수 있다. 여기서, 상기 바닥면은 정면을 바라보는 카메라가 획득한 데이터를 포함하는 영역일 수 있다.
- [0149] 도 9b의 (f)는 파노라믹 프로젝션 포맷을 나타낼 수 있다. 파노라믹 프로젝션 스포맷이 사용되는 경우, 전술한 프로젝션 처리부는, 도 9b의 (f)에 도시된 것과 같이 구형 면 상의 360 비디오 데이터 중 옆면 만을 2D 이미지 상에 프로젝션할 수 있다. 이는 실린더형 프로젝션 스킴에서 윗면(top)과 바닥면(bottom)이 존재하지 않는 경우와 같을 수 있다.
- [0150] 한편, 본 발명의 실시예에 의하면, 스티칭없이 프로젝션이 수행될 수 있다. 도 9b의 (g)는 스티칭없이 프로젝션이 수행되는 경우를 나타낼 수 있다. 스티칭없이 프로젝션되는 경우, 전술한 프로젝션 처리부는, 도 9b의 (g)에 도시된 것과 같이, 360 비디오 데이터를 그대로 2D 이미지 상에 프로젝션할 수 있다. 이 경우 스티칭은 수행되지 않고, 카메라에서 획득된 각각의 이미지들이 그대로 2D 이미지 상에 프로젝션될 수 있다.
- [0151] 도 9b의 (g)를 참조하면 두 개의 이미지가 2D 이미지 상에 스티칭없이 프로젝션될 수 있다. 각 이미지는 구형 카메라(spherical camera) (또는 어안(fish-eye) 카메라)에서 각 센서를 통해 획득한 어안(fish-eye) 이미지일 수 있다. 전술한 바와 같이, 수신측에서 카메라 센서들로부터 획득하는 이미지 데이터를 스티칭할 수 있고, 스티칭된 이미지 데이터를 구형 면(spherical surface) 상에 맵핑하여 구형 비디오(spherical video), 즉, 360 비디오를 렌더링할 수 있다.
- [0152] 도 10a 및 10b는 본 발명의 일 실시예에 따른 타일(Tile)을 도시한 도면이다.
- [0153] 2D 이미지에 프로젝션된 360 비디오 데이터 또는 리전별 패키징까지 수행된 360 비디오 데이터는 하나 이상의 타일로 구분될 수 있다. 도시된 10a는 하나의 2D 이미지가 16 개의 타일로 나뉘어진 형태를 도시하고 있다. 여기서 2D 이미지란 전술한 프로젝티드 프레임 내지는 팩드 프레임일 수 있다. 본 발명에 따른 360 비디오 전송 장치의 또 다른 실시예에 의하면, 데이터 인코더는 각각의 타일을 독립적으로 인코딩할 수 있다.
- [0154] 전술한 리전별 패키징과 타일링(Tiling)은 구분될 수 있다. 전술한 리전별 패키징은 코딩 효율을 높이기 위해 또는 레졸루션을 조정하기 위하여 2D 이미지상에 프로젝션된 360 비디오 데이터를 리전으로 구분하여 처리하는 것을 의미할 수 있다. 타일링은 데이터 인코더가 프로젝티드 프레임 내지는 팩드 프레임을 타일이라는 구획별로 나누고, 해당 타일을 별로 독립적으로 인코딩을 수행하는 것을 의미할 수 있다. 360 비디오가 제공될 때, 사용자는 360 비디오의 모든 부분을 동시에 소비하지 않는다. 타일링은 제한된 밴드위스(bandwidth)상에서 사용자가 현재 보는 뷰포트 등 중요 부분 내지 일정 부분에 해당하는 타일만을 수신측으로 전송 혹은 소비하는 것을 가능케할 수 있다. 타일링을 통해 제한된 밴드위스가 더 효율적으로 활용될 수 있고, 수신측에서도 모든 360 비디오 데이터를 한번에 다 처리하는 것에 비하여 연산 부하를 줄일 수 있다.
- [0155] 리전과 타일은 구분되므로, 두 영역이 같을 필요는 없다. 그러나 실시예에 따라 리전과 타일은 같은 영역을 지칭할 수도 있다. 실시예에 따라 타일에 맞추어 리전별 패키징이 수행되어 리전과 타일이 같아질 수 있다. 또한 실시예에 따라, 프로젝션 스킴에 따른 각 면과 리전이 같은 경우, 프로젝션 스킴에 따른 각 면, 리전, 타일이 같은 영역을 지칭할 수도 있다. 문맥에 따라 리전은 VR 리전, 타일을 타일 리전으로 불릴 수도 있다.
- [0156] ROI (Region of Interest)는 360 콘텐츠 제공자가 제안하는, 사용자들의 관심 영역을 의미할 수 있다. 360 콘텐츠 제공자는 360 비디오를 제작할 때, 어느 특정 영역을 사용자들이 관심있어 할 것으로 보고, 이를 고려하여

360 비디오를 제작할 수 있다. 실시예에 따라 ROI 는 360 비디오의 콘텐츠 상, 중요한 내용이 재생되는 영역에 해당할 수 있다.

- [0157] 본 발명에 따른 360 비디오 전송/수신 장치의 또 다른 실시예에 의하면, 수신측 피드백 처리부는 뷰포트 정보를 추출, 수집하여 이를 송신측 피드백 처리부로 전달할 수 있다. 이 과정에서 뷰포트 정보는 양 측의 네트워크 인터페이스를 이용해 전달될 수 있다. 도시된 10a의 2D 이미지에서 뷰포트(1000) 가 표시되었다. 여기서 뷰포트는 2D 이미지 상의 9개의 타일에 걸쳐 있을 수 있다.
- [0158] 이 경우 360 비디오 전송 장치는 타일링 시스템을 더 포함할 수 있다. 실시예에 따라 타일링 시스템은 데이터 인코더 다음에 위치할 수도 있고(도시된 10b), 전송한 데이터 인코더 내지 전송 처리부 내에 포함될 수도 있고, 별개의 내/외부 엘리먼트로서 360 비디오 전송 장치에 포함될 수 있다.
- [0159] 타일링 시스템은 송신측 피드백 처리부로부터 뷰포트 정보를 전달받을 수 있다. 타일링 시스템은 뷰포트 영역이 포함되는 타일만을 선별하여 전송할 수 있다. 도시된 10a 의 2D 이미지에서 총 16개의 타일 중 뷰포트 영역(1000)을 포함하는 9개의 타일들만이 전송될 수 있다. 여기서 타일링 시스템은 브로드밴드를 통한 유니캐스트 방식으로 타일들을 전송할 수 있다. 사용자에 따라 뷰포트 영역이 다르기 때문이다.
- [0160] 또한 이 경우 송신측 피드백 처리부는 뷰포트 정보를 데이터 인코더로 전달할 수 있다. 데이터 인코더는 뷰포트 영역을 포함하는 타일들에 대해 다른 타일들보다 더 높은 퀄리티로 인코딩을 수행할 수 있다.
- [0161] 또한 이 경우 송신측 피드백 처리부는 뷰포트 정보를 메타데이터 처리부로 전달할 수 있다. 메타데이터 처리부는 뷰포트 영역과 관련된 메타데이터를 360 비디오 전송 장치의 각 내부 엘리먼트로 전달해주거나, 360 비디오 관련 메타데이터에 포함시킬 수 있다.
- [0162] 이러한 타일링 방식을 통하여, 전송 밴드위스(bandwidth)가 절약될 수 있으며, 타일 별로 차등화된 처리를 수행하여 효율적 데이터 처리/전송이 가능해질 수 있다.
- [0163] 전송한 뷰포트 영역과 관련된 실시예들은 뷰포트 영역이 아닌 다른 특정 영역들에 대해서도 유사한 방식으로 적용될 수 있다. 예를 들어, 전송한 게이즈 분석을 통해 사용자들이 주로 관심있어 하는 것으로 판단된 영역, ROI 영역, 사용자가 VR 디스플레이를 통해 360 비디오를 접할 때 처음으로 재생되는 영역(초기 시점, Initial Viewpoint) 등에 대해서도, 전송한 뷰포트 영역과 같은 방식의 처리들이 수행될 수 있다.
- [0164] 본 발명에 따른 360 비디오 전송 장치의 또 다른 실시예에 의하면, 전송 처리부는 각 타일 별로 다르게 전송을 위한 처리를 수행할 수 있다. 전송 처리부는 타일 별로 다른 전송 파라미터(모듈레이션 오더, 코드 레이트 등)를 적용하여, 각 타일 별로 전달되는 데이터의 강건성(robustness)을 다르게 할 수 있다.
- [0165] 이 때, 송신측 피드백 처리부는 360 비디오 수신 장치로부터 전달받은 피드백 정보를 전송 처리부로 전달하여, 전송 처리부가 타일별 차등화된 전송 처리를 수행하도록 할 수 있다. 예를 들어 송신측 피드백 처리부는 수신측 으로부터 전달받은 뷰포트 정보를 전송 처리부로 전달할 수 있다. 전송 처리부는 해당 뷰포트 영역을 포함하는 타일들에 대해 다른 타일들보다 더 높은 강건성을 가지도록 전송 처리를 수행할 수 있다.
- [0166] 도 11은 본 발명의 일 실시예에 따른 360도 비디오 관련 메타데이터의 일 예를 나타낸다. 상술한 내용과 같이 360도 비디오 관련 메타데이터는 360도 비디오에 대한 다양한 메타데이터를 포함할 수 있다. 문맥에 따라, 360도 비디오 관련 메타데이터는 360도 비디오 관련 시그널링 정보라고 불릴 수도 있다. 360도 비디오 관련 메타데이터는 별도의 시그널링 테이블에 포함되어 전송될 수도 있고, DASH MPD 내에 포함되어 전송될 수도 있고, ISOBMFF 등의 파일 포맷에 box 형태로 포함되어 전달될 수도 있다. 360도 비디오 관련 메타데이터가 box 형태로 포함되는 경우 파일, 프래그먼트, 트랙, 샘플 엔트리, 샘플 등등 다양한 레벨에 포함되어 해당되는 레벨의 데이터에 대한 메타데이터를 포함할 수 있다.
- [0167] 실시예에 따라, 후술하는 메타데이터의 일부는 시그널링 테이블로 구성되어 전달되고, 나머지 일부는 파일 포맷 내에 box 혹은 트랙 형태로 포함될 수도 있다.
- [0168] 본 발명에 따른 360도 비디오 관련 메타데이터의 일 실시예에 의하면, 360도 비디오 관련 메타데이터는 프로젝션 스킵 등에 관한 기본 메타데이터, 스테레오스코픽(stereoscopic) 관련 메타데이터, 초기 시점(Initial View/Initial Viewpoint) 관련 메타데이터, ROI 관련 메타데이터, FOV (Field of View) 관련 메타데이터 및/또는 크롭된 영역(cropped region) 관련 메타데이터를 포함할 수 있다. 실시예에 따라 360도 비디오 관련 메타데이터는 전송한 것 외에 추가적인 메타데이터를 더 포함할 수 있다.

- [0169] 본 발명에 따른 360도 비디오 관련 메타데이터의 실시예들은 전술한 기본 메타데이터, 스테레오스코픽 관련 메타데이터, 초기 시점 관련 메타데이터, ROI 관련 메타데이터, FOV 관련 메타데이터, 크롭된 영역 관련 메타데이터 및/또는 이후 추가될 수 있는 메타데이터들 중 적어도 하나 이상을 포함하는 형태일 수 있다. 본 발명에 따른 360도 비디오 관련 메타데이터의 실시예들은, 각각 포함하는 세부 메타데이터들의 경우의 수에 따라 다양하게 구성될 수 있다. 실시예에 따라 360도 비디오 관련 메타데이터는 전술한 것 외에 추가적인 정보들을 더 포함할 수도 있다.
- [0170] stereo_mode 필드는 해당 360도 비디오가 지원하는 3D 레이아웃을 지시할 수 있다. 본 필드만으로 해당 360도 비디오가 3D 를 지원하는지 여부를 지시할 수도 있는데, 이 경우 전술한 is_stereoscopic 필드는 생략될 수 있다. 본 필드 값이 0 인 경우, 해당 360도 비디오는 모노(mono) 모드일 수 있다. 즉 프로젝션된 2D 이미지는 하나의 모노 뷰(mono view) 만을 포함할 수 있다. 이 경우 해당 360도 비디오는 3D 를 지원하지 않을 수 있다.
- [0171] 본 필드 값이 1, 2 인 경우, 해당 360도 비디오는 각각 좌우(Left-Right) 레이아웃, 상하(Top-Bottom) 레이아웃에 따를 수 있다. 좌우 레이아웃, 상하 레이아웃은 각각 사이드-바이-사이드 포맷, 탑-바텀 포맷으로 불릴 수도 있다. 좌우 레이아웃의 경우, 좌영상/우영상이 프로젝션된 2D 이미지들은 이미지 프레임 상에서 각각 좌/우로 위치할 수 있다. 상하 레이아웃의 경우, 좌영상/우영상이 프로젝션된 2D 이미지들은 이미지 프레임 상에서 각각 위/아래로 위치할 수 있다. 해당 필드가 나머지 값을 가지는 경우는 향후 사용을 위해 남겨둘 수 있다 (Reserved for Future Use).
- [0172] 초기 시점 관련 메타데이터는 사용자가 360도 비디오를 처음 재생했을 때 보게되는 시점(초기 시점)에 대한 정보를 포함할 수 있다. 초기 시점 관련 메타데이터는 initial_view_yaw_degree 필드, initial_view_pitch_degree 필드 및/또는 initial_view_roll_degree 필드를 포함할 수 있다. 실시예에 따라 초기 시점 관련 메타데이터는 추가적인 정보들을 더 포함할 수도 있다.
- [0173] initial_view_yaw_degree 필드, initial_view_pitch_degree 필드, initial_view_roll_degree 필드는 해당 360도 비디오 재생 시의 초기 시점을 나타낼 수 있다. 즉, 재생시 처음 보여지는 뷰포트의 정중앙 지점이, 이 세 필드들에 의해 나타내어질 수 있다. 구체적으로, 상기 initial_view_yaw_degree 필드는 상기 초기 시점에 대한 yaw 값을 나타낼 수 있다. 즉, 상기 initial_view_yaw_degree 필드는 상기 정중앙 지점의 위치를 yaw 축을 기준으로 회전된 방향(부호) 및 그 정도(각도)로 나타낼 수 있다. 또한, 상기 initial_view_pitch_degree 필드는 상기 초기 시점에 대한 pitch 값을 나타낼 수 있다. 즉, 상기 initial_view_pitch_degree 필드는 상기 정중앙 지점의 위치를 pitch 축을 기준으로 회전된 방향(부호) 및 그 정도(각도)로 나타낼 수 있다. 또한, 상기 initial_view_roll_degree 필드는 상기 초기 시점에 대한 roll 값을 나타낼 수 있다. 즉, 상기 initial_view_roll_degree 필드는 상기 정중앙 지점의 위치를 roll 축을 기준으로 회전된 방향(부호) 및 그 정도(각도)로 나타낼 수 있다. 상기 initial_view_yaw_degree 필드, 상기 initial_view_pitch_degree 필드, 상기 initial_view_roll_degree 필드를 기반으로 해당 360도 비디오 재생 시의 초기 시점, 즉, 재생시 처음 보여지는 뷰포트의 정중앙 지점을 나타낼 수 있고, 이를 통하여 상기 360도 비디오의 특정 영역이 사용자에게 초기 시점에 디스플레이되어 제공될 수 있다. 또한, FOV(field of view)를 통하여, 지시된 초기 시점을 기준으로 한, 초기 뷰포트의 가로길이 및 세로길이(width, height)가 결정될 수 있다. 즉, 이 세 필드들 및 FOV 정보를 이용하여, 360도 비디오 수신 장치는 사용자에게 360도 비디오의 일정 영역을 초기 뷰포트로서 제공할 수 있다.
- [0174] 실시예에 따라, 초기 시점 관련 메타데이터가 지시하는 초기 시점은, 장면(scene) 별로 변경될 수 있다. 즉, 360 콘텐츠의 시간적 흐름에 따라 360도 비디오의 장면이 바뀌게 되는데, 해당 360도 비디오의 장면마다 사용자가 처음 보게되는 초기 시점 내지 초기 뷰포트가 변경될 수 있다. 이 경우, 초기 시점 관련 메타데이터는 각 장면별로의 초기 시점을 지시할 수 있다. 이를 위해 초기 시점 관련 메타데이터는, 해당 초기 시점이 적용되는 장면을 식별하는 장면(scene) 식별자를 더 포함할 수도 있다. 또한 360도 비디오의 장면별로 FOV(Field Of View)가 변할 수도 있으므로, 초기 시점 관련 메타데이터는 해당 장면에 해당하는 FOV를 나타내는 장면별 FOV 정보를 더 포함할 수도 있다.
- [0175] ROI 관련 메타데이터는 전술한 ROI에 관련된 정보들을 포함할 수 있다. ROI 관련 메타데이터는, 2d_roi_range_flag 필드 및/또는 3d_roi_range_flag 필드를 포함할 수 있다. 2d_roi_range_flag 필드는 ROI 관련 메타데이터가 2D 이미지를 기준으로 ROI를 표현하는 필드들을 포함하는지 여부를 지시할 수 있고, 3d_roi_range_flag 필드는 ROI 관련 메타데이터가 3D 공간을 기준으로 ROI를 표현하는 필드들을 포함하는지 여부를 지시할 수 있다. 실시예에 따라 ROI 관련 메타데이터는, ROI에 따른 차등 인코딩 정보, ROI에 따른 차등 전송처리 정보 등 추가적인 정보들을 더 포함할 수도 있다.

- [0176] ROI 관련 메타데이터가 2D 이미지를 기준으로 ROI를 표현하는 필드들을 포함하는 경우, ROI 관련 메타데이터는 min_top_left_x 필드, max_top_left_x 필드, min_top_left_y 필드, max_top_left_y 필드, min_width 필드, max_width 필드, min_height 필드, max_height 필드, min_x 필드, max_x 필드, min_y 필드 및/또는 max_y 필드를 포함할 수 있다.
- [0177] min_top_left_x 필드, max_top_left_x 필드, min_top_left_y 필드, max_top_left_y 필드는 ROI 의 좌측 상단 끝의 좌표의 최소/최대값을 나타낼 수 있다. 즉, 상기 필드들은 차례로 좌상단 끝의 최소 x 좌표, 최대 x 좌표, 최소 y 좌표, 최대 y 좌표를 나타낼 수 있다.
- [0178] min_width 필드, max_width 필드, min_height 필드, max_height 필드는 ROI 의 가로 크기(width), 세로 크기(height)의 최소/최대값을 나타낼 수 있다. 즉, 상기 필드들은 차례로 가로 크기의 최소값, 가로 크기의 최대값, 세로 크기의 최소값, 세로 크기의 최대값을 나타낼 수 있다.
- [0179] min_x 필드, max_x 필드, min_y 필드, max_y 필드는 ROI 내의 좌표들의 최소/최대값을 나타낼 수 있다. 즉, 상기 필드들은 차례로 ROI 내 좌표들의 최소 x 좌표, 최대 x 좌표, 최소 y 좌표, 최대 y 좌표를 나타낼 수 있다. 이 필드들은 생략될 수 있다.
- [0180] ROI 관련 메타데이터가 3D 렌더링 공간 상의 좌표 기준으로 ROI를 표현하는 필드들을 포함하는 경우, ROI 관련 메타데이터는 min_yaw 필드, max_yaw 필드, min_pitch 필드, max_pitch 필드, min_roll 필드, max_roll 필드, min_field_of_view 필드 및/또는 max_field_of_view 필드를 포함할 수 있다.
- [0181] min_yaw 필드, max_yaw 필드, min_pitch 필드, max_pitch 필드, min_roll 필드, max_roll 필드는 ROI가 3D 공간상에서 차지하는 영역을 yaw, pitch, roll 의 최소/최대값으로 나타낼 수 있다. 즉, 상기 필드들은 차례로 yaw 축 기준 회전량의 최소값, yaw 축 기준 회전량의 최대값, pitch 축 기준 회전량의 최소값, pitch 축 기준 회전량의 최대값, roll 축 기준 회전량의 최소값, roll 축 기준 회전량의 최대값을 나타낼 수 있다.
- [0182] min_field_of_view 필드, max_field_of_view 필드는 해당 360도 비디오 데이터의 FOV(Field Of View)의 최소/최대값을 나타낼 수 있다. FOV 는 360도 비디오의 재생시 한번에 디스플레이되는 시야범위를 의미할 수 있다. min_field_of_view 필드, max_field_of_view 필드는 각각 FOV의 최소값, 최대값을 나타낼 수 있다. 이 필드들은 생략될 수 있다. 이 필드들은 후술할 FOV 관련 메타데이터에 포함될 수도 있다.
- [0183] FOV 관련 메타데이터는 전술한 FOV 에 관련한 정보들을 포함할 수 있다. FOV 관련 메타데이터는 content_fov_flag 필드 및/또는 content_fov 필드를 포함할 수 있다. 실시예에 따라 FOV 관련 메타데이터는 전술한 FOV의 최소/최대값 관련 정보 등 추가적인 정보들을 더 포함할 수도 있다.
- [0184] content_fov_flag 필드는 해당 360도 비디오에 대하여 제작시 의도한 FOV에 대한 정보가 존재하는지 여부를 지시할 수 있다. 본 필드값이 1인 경우, content_fov 필드가 존재할 수 있다.
- [0185] content_fov 필드는 해당 360도 비디오에 대하여 제작시 의도한 FOV에 대한 정보를 나타낼 수 있다. 실시예에 따라 해당 360도 비디오 수신 장치의 수직(vertical) 혹은 수평(horizontal) FOV에 따라, 360 영상 중에서 사용자에게 한번에 디스플레이되는 영역이 결정될 수 있다. 혹은 실시예에 따라 본 필드의 FOV 정보를 반영하여 사용자에게 한번에 디스플레이되는 360도 비디오의 영역이 결정될 수도 있다.
- [0186] 크롭된 영역 관련 메타데이터는 이미지 프레임 상에서 실제 360도 비디오 데이터를 포함하는 영역에 대한 정보를 포함할 수 있다. 이미지 프레임은 실제 360도 비디오 데이터 프로젝션된 액티브 비디오 영역(Active Video Area)과 그렇지 않은 영역을 포함할 수 있다. 이 때 액티브 비디오 영역은 크롭된 영역 또는 디폴트 디스플레이 영역이라고 칭할 수 있다. 이 액티브 비디오 영역은 실제 VR 디스플레이 상에서 360도 비디오로서 보여지는 영역으로서, 360도 비디오 수신 장치 또는 VR 디스플레이는 액티브 비디오 영역만을 처리/디스플레이할 수 있다. 예를 들어 이미지 프레임의 종횡비(aspect ratio)가 4:3 인 경우 이미지 프레임의 윗 부분 일부와 아랫부분 일부를 제외한 영역만 360도 비디오 데이터를 포함할 수 있는데, 이 부분을 액티브 비디오 영역이라고 할 수 있다.
- [0187] 크롭된 영역 관련 메타데이터는 is_cropped_region 필드, cr_region_left_top_x 필드, cr_region_left_top_y 필드, cr_region_width 필드 및/또는 cr_region_height 필드를 포함할 수 있다. 실시예에 따라 크롭된 영역 관련 메타데이터는 추가적인 정보들을 더 포함할 수도 있다.
- [0188] is_cropped_region 필드는 이미지 프레임의 전체 영역이 360도 비디오 수신 장치 내지 VR 디스플레이에 의해 사용되는지 여부를 나타내는 플래그일 수 있다. 여기서, 360도 비디오 데이터가 매핑된 영역 혹은 VR 디스플레이

상에서 보여지는 영역은 액티브 비디오 영역(Active Video Area)라고 불릴 수 있다. 상기 is_cropped_region 필드는 이미지 프레임 전체가 액티브 비디오 영역인지 여부를 지시할 수 있다. 이미지 프레임의 일부만이 액티브 비디오 영역인 경우, 하기의 4 필드가 더 추가될 수 있다.

- [0189] cr_region_left_top_x 필드, cr_region_left_top_y 필드, cr_region_width 필드, cr_region_height 필드는 이미지 프레임 상에서 액티브 비디오 영역을 나타낼 수 있다. 이 필드들은 각각 액티브 비디오 영역의 좌상단의 x 좌표, 액티브 비디오 영역의 좌상단의 y 좌표, 액티브 비디오 영역의 가로 길이(width), 액티브 비디오 영역의 세로 길이(height)를 나타낼 수 있다. 가로 길이와 세로 길이는 픽셀을 단위로 나타내어질 수 있다.
- [0190] 360 비디오 기반 VR 시스템은 전술한 360 비디오 처리 과정을 기반으로 360 비디오에 대하여 사용자의 위치를 기준으로 서로 다른 뷰잉 오리엔테이션(viewing orientation)에 대한 시각적/청각적 경험을 제공할 수 있다. 360 비디오에 대하여 사용자의 고정 위치에서의 서로 다른 뷰잉 오리엔테이션에 대한 시각적/청각적 경험을 제공하는 VR 시스템은 3DoF(three degree of freedom) 기반 VR 시스템이라고 불릴 수 있다. 한편, 서로 다른 뷰포인트(viewpoint), 서로 다른 뷰잉 포지션(viewing position)에서의 서로 다른 뷰잉 오리엔테이션에 대한 확장된 시각적/청각적 경험을 제공할 수 있는 VR 시스템은 3DoF+ 또는 3DoF plus 기반 VR 시스템이라고 불릴 수 있다.
- [0191] 도 12는 뷰포인트, 뷰잉 포지션, 뷰잉 오리엔테이션의 개념을 개략적으로 나타낸다.
- [0192] 도 12를 참조하면, (a)와 같은 공간(ex. 공연장)을 가정했을 때, 표시된 각 서클은 서로 다른 뷰포인트를 나타낼 수 있다. 상기 같은 공간 내에 위치하는 각 뷰포인트에서 제공되는 영상/음성은 동일한 시간대에서 서로 연관될 수 있다. 이 경우, 특정 뷰포인트에서 사용자의 시선 방향 변화(ex. head motion)에 따라 서로 다른 시각적/청각적 경험을 사용자에게 제공할 수 있다. 즉, 특정 뷰포인트에 대해 (b)에 도시된 바와 같은 다양한 뷰잉 포지션의 스피어(sphere)를 가정할 수 있으며, 각 뷰잉 포지션의 상대적인 위치를 반영한 영상/음성/텍스트 정보를 제공할 수 있다.
- [0193] 한편, (c)에 도시된 바와 같이 특정 뷰포인트의 특정 뷰핑 포지션에서는 기존의 3DoF와 같이 다양한 방향의 시각적/청각적 정보를 전달할 수 있다. 이 때 메인 소스(ex. 영상/음성/텍스트)뿐만 아니라 추가적인 다양한 소스를 통합하여 제공할 수 있으며, 이 경우 사용자의 뷰잉 오리엔테이션과 연계되거나 독립적으로 정보가 전달될 수 있다.
- [0194] 도 13은 본 발명에 따른 3DoF+ 비디오 제공을 위한 아키텍처의 예를 개략적으로 도시한 도면이다. 도 13은 3DoF+의 영상획득, 전처리, 전송, (후)처리, 렌더링 및 피드백 과정을 포함한 3DoF+ 엔드 투 엔드 시스템 흐름도를 나타낼 수 있다.
- [0195] 도 13을 참조하면, 획득(acquisition) 과정은 360 비디오의 캡처, 합성 또는 생성 과정 등을 통한 360 비디오를 획득하는 과정을 의미할 수 있다. 이 과정을 통하여 다수의 위치에 대해 시선 방향 변화(ex. head motion)에 따른 다수의 영상/음성 정보를 획득할 수 있다. 이 때, 영상 정보는 시각적 정보(ex. texture)뿐 아니라 깊이 정보(depth)를 포함할 수 있다. 이 때 1310의 영상 정보 예시와 같이 서로 다른 뷰포인트(viewpoint)에 따른 서로 다른 뷰잉 포지션(viewing position)의 복수의 정보를 각각 획득할 수 있다.
- [0196] 합성(composition) 과정은 영상/음성 입력 장치를 통해 획득한 정보뿐 아니라 외부 미디어를 통한 영상(비디오/이미지 등), 음성(오디오/효과음향 등), 텍스트(자막 등)을 사용자 경험에 포함하기 위해 합성하기 위한 절차 및 방법을 포함할 수 있다.
- [0197] 전처리(pre-processing) 과정은 획득된 360 비디오의 전송/전달을 위한 준비(전처리) 과정으로서, 전술한 스티칭, 프로젝션, 리전별 패킹 과정 및/또는 인코딩 과정 등을 포함할 수 있다. 즉, 이 과정은 영상/음성/텍스트 정보를 제작자의 의도에 따라 데이터를 변경/보완 하기위한 전처리 과정 및 인코딩 과정이 포함될 수 있다. 예를 들어 영상의 전처리 과정에서는 획득된 시각 정보를 360 스피어(sphere) 상에 매핑하는 작업(stitching), 영역 경계를 없애거나 색상/밝기 차이를 줄이거나 영상의 시각적 효과를 주는 보정 작업(editing), 시점에 따른 영상을 분리하는 과정(view segmentation), 360 스피어(sphere) 상의 영상을 2D 영상으로 매핑하는 프로젝션 과정(projection), 영역에 따라 영상을 재배치 하는 과정(region-wise packing), 영상 정보를 압축하는 인코딩 과정이 포함될 수 있다. 1320의 비디오 측면의 예시와 같이 서로 다른 뷰포인트(viewpoint)에 따른 서로 다른 뷰잉 포지션(viewing position)의 복수의 프로젝션 영상이 생성될 수 있다.
- [0198] 전송 과정은 준비 과정(전처리 과정)을 거친 영상/음성 데이터 및 메타데이터들을 처리하여 전송하는 과정을 의미할 수 있다. 서로 다른 뷰포인트(viewpoint)에 따른 서로 다른 뷰잉 포지션(viewing position)의 복수의 영상/음성 데이터 및 관련 메타데이터를 전달하는 방법으로써 전술한 바와 같이 방송망, 통신망을 이용하거나, 단방

향 전달 등의 방법을 사용할 수 있다.

- [0199] 후처리 및 합성 과정은 수신된/저장된 비디오/오디오/텍스트 데이터를 디코딩하고 최종 재생을 위한 후처리 과정을 의미할 수 있다. 예를 들어 후처리 과정은 전송된 바와 같이 패킹 된 영상을 풀어주는 언패킹 및 2D 프로젝트된 영상을 3D 구형 영상으로 복원하는 리-프로젝션 과정 등이 포함될 수 있다.
- [0200] 렌더링 과정은 3D 공간상에 리-프로젝션된 이미지/비디오 데이터를 렌더링하고 디스플레이하는 과정을 의미할 수 있다. 이 과정에서 영상/음성 신호를 최종적으로 출력하기 위한 형태로 재구성할 수 있다. 사용자의 관심영역이 존재하는 방향(viewing orientation), 시점(viewing position/head position), 위치(viewpoint)를 추적할 수 있으며, 이 정보에 따라 필요한 영상/음성/텍스트 정보만을 선택적으로 사용할 수 있다. 이 때, 영상 신호의 경우 사용자의 관심영역에 따라 1330와 같이 서로 다른 시점이 선택될 수 있으며, 최종적으로 1340와 같이 특정 위치에서의 특정 시점의 특정 방향의 영상이 출력될 수 있다.
- [0201] 도 14a 및 14b는 3DoF+ 엔드 투 엔드 시스템 아키텍처의 예이다. 도 14a 및 14b의 아키텍처에 의하여 전송된 바와 같은 3DOF+ 360 콘텐츠가 제공될 수 있다.
- [0202] 도 14a를 참조하면, 360 비디오 전송 장치(송신단)은 크게 360 비디오(이미지)/오디오 데이터 획득이 이루어지는 부분(acquisition unit), 획득된 데이터를 처리하는 부분(video/audio pre-processor), 추가 정보를 합성하기 위한 부분(composition generation unit), 텍스트, 오디오 및 프로젝트된 360도 비디오를 인코딩하는 부분(encoding unit) 및 인코딩된 데이터를 인캡슐레이션하는 부분(encapsulation unit)으로 구성될 수 있다. 전송된 바와 같이 인코딩된 데이터는 비트스트림(bitstream) 형태로 출력될 수 있으며, 인코딩된 데이터는 ISO/BMFF, CFF 등의 파일 포맷으로 인캡슐레이션되거나, 기타 DASH 세그먼트 등의 형태로 처리할 수 있다. 인코딩된 데이터는 디지털 저장 매체를 통하여 360 비디오 수신 장치로 전달될 수 있으며, 또는 비록 명시적으로 도시되지는 않았으나, 전송된 바와 같이 전송 처리부를 통하여 전송을 위한 처리를 거치고, 이후 방송망 또는 브로드밴드 등을 통하여 전송될 수 있다.
- [0203] 데이터 획득 부분에서는 센서의 방향(sensor orientation, 영상의 경우 viewing orientation), 센서의 정보 획득 시점(sensor position, 영상의 경우 viewing position), 센서의 정보 획득 위치(영상의 경우 viewpoint)에 따라 서로 다른 정보를 동시에 혹은 연속적으로 획득할 수 있으며, 이 때 비디오, 이미지, 오디오, 위치 정보 등을 획득할 수 있다.
- [0204] 영상 데이터의 경우 텍스처(texture) 및 깊이 정보(depth)를 각각 획득할 수 있으며, 각 컴포넌트의 특성에 따라 서로 다른 전처리(video pre-processing)가 가능하다. 예를 들어 텍스처 정보의 경우 이미지 센서 위치 정보를 이용하여 동일 위치(viewpoint)에서 획득한 동일 시점(viewing position)의 서로 다른 방향 (viewing orientation)의 영상들을 이용하여 360 전방위 영상을 구성할 수 있으며, 이를 위해 영상 스티칭 (stitching) 과정을 수행할 수 있다. 또한 영상을 인코딩하기 위한 포맷으로 변경하기 위한 프로젝트(projection) 및/또는 리전별 패킹을 수행할 수 있다. 깊이 영상의 경우 일반적으로 탑스 카메라를 통해 영상을 획득할 수 있으며, 이 경우 텍스처와 같은 형태로 깊이 영상을 만들 수 있다. 혹은, 별도로 측정된 데이터를 바탕으로 깊이 데이터를 생성할 수도 있다. 컴포넌트 별 영상이 생성된 후 효율적인 압축을 위한 비디오 포맷으로의 추가 변환(packing)을 하거나 실제 필요한 부분으로 나누어 재 구성하는 과정 (sub-picture generation)이 수행될 수 있다. Video pre-processing 단에서 사용된 영상 구성에 대한 정보는 video metadata로 전달된다.
- [0205] 획득된 데이터(혹은 주요하게 서비스 하기 위한 데이터) 이외에 추가적으로 주어지는 영상/음성/텍스트 정보를 함께 서비스 하는 경우, 이들 정보를 최종 재생 시 합성하기 위한 정보를 제공할 필요가 있다. 컴포지션 생성부(Composition generation unit)에서는 제작자의 의도를 바탕으로 외부에서 생성된 미디어 데이터 (영상의 경우 비디오/이미지, 음성의 경우 오디오/효과 음향, 텍스트의 경우 자막 등)를 최종 재생 단에서 합성하기 위한 정보를 생성하며, 이 정보는 composition metadata로 전달된다.
- [0206] 각각의 처리를 거친 영상/음성/텍스트 정보는 각각의 인코더를 이용해 압축되고, 어플리케이션에 따라 파일 혹은 세그먼트 단위로 인캡슐레이션 된다. 이 때, 비디오, 파일 혹은 세그먼트 구성 방법에 따라 필요한 정보만을 추출(file extractor)이 가능하다.
- [0207] 또한 각 데이터를 수신기에서 재구성하기 위한 정보가 코덱 혹은 파일 포맷/시스템 레벨에서 전달되는데, 여기에서는 비디오/오디오 재구성을 위한 정보(video/audio metadata), 오버레이를 위한 합성 정보(composition metadata), 비디오/오디오 재생 가능 위치(viewpoint) 및 각 위치에 따른 시점(viewing position) 정보(viewing position and viewpoint metadata) 등이 포함된다. 이와 같은 정보의 처리는 별도의 메타데이터 처리

부를 통한 생성도 가능하다.

- [0208] 도 14b를 참조하면, 360 비디오 수신 장치(수신단)는 크게 수신된 파일 혹은 세그먼트를 디캡슐레이션하는 부분(file/segment decapsulation unit), 비트스트림으로부터 영상/음성/텍스트 정보를 생성하는 부분(decoding unit), 영상/음성/텍스트를 재생하기 위한 형태로 재구성하는 부분(post-processor), 사용자의 관심영역을 추적하는 부분(tracking unit) 및 재생 장치인 디스플레이로 구성될 수 있다.
- [0209] 디캡슐레이션을 통해 생성된 비트스트림은 데이터의 종류에 따라 영상/음성/텍스트 등으로 나뉘어 재생 가능한 형태로 개별적으로 디코딩될 수 있다.
- [0210] 트래킹 부분에서는 센서 및 사용자의 입력 정보 등을 바탕으로 사용자의 관심 영역(Region of interest)의 위치(viewpoint), 해당 위치에서의 시점(viewing position), 해당 시점에서의 방향(viewing orientation) 정보를 생성하게 되며, 이 정보는 360 비디오 수신 장치의 각 모듈에서 관심 영역 선택 혹은 추출 등에 사용되거나, 관심 영역의 정보를 강조하기 위한 후처리 과정 등에 사용될 수 있다. 또한 360 비디오 전송 장치에 전달되는 경우 효율적인 대역폭 사용을 위한 파일 선택(file extractor) 혹은 서브 픽처 선택, 관심영역에 기반한 다양한 영상 재구성 방법(viewport/viewing position / viewpoint dependent processing) 등에 사용될 수 있다.
- [0211] 디코딩 된 영상 신호는 영상 구성 방법에 따라 다양한 처리 방법에 따라 처리될 수 있다. 360 비디오 전송 장치에서 영상 패키징이 이루어진 경우 메타데이터를 통해 전달된 정보를 바탕으로 영상을 재구성 하는 과정이 필요하다. 이 경우 360 비디오 전송 장치에서 생성한 비디오 메타데이터를 이용할 수 있다. 또한 디코딩 된 영상 내에 복수의 시청 위치(viewpoint), 혹은 복수의 시점(viewing position), 혹은 다양한 방향(viewing orientation)의 영상이 포함된 경우 트래킹(tracking)을 통해 생성된 사용자의 관심 영역의 위치, 시점, 방향 정보와 매칭되는 정보를 선택하여 처리할 수 있다. 이 때, 송신단에서 생성한 viewing position 및 viewpoint 관련 메타데이터가 사용될 수 있다. 또한 특정 위치, 시점, 방향에 대해 복수의 컴포넌트가 전달되거나, 오버레이를 위한 비디오 정보가 별도로 전달되는 경우 각각에 따른 렌더링 과정이 포함될 수 있다. 별도의 렌더링 과정을 거친 비디오 데이터(텍스처, 맵스, 오버레이)는 합성 과정(composition)을 거치게 되며, 이 때, 송신단에서 생성한 합성 메타데이터(composition metadata)가 사용될 수 있다. 최종적으로 사용자의 관심 영역에 따라 뷰포트(viewport)에 재생하기 위한 정보를 생성할 수 있다.
- [0212] 디코딩된 음성 신호는 오디오 렌더러 그리고/혹은 후처리 과정을 통해 재생 가능한 음성 신호를 생성하게 되며, 이 때 사용자의 관심 영역에 대한 정보 및 360 비디오 수신 장치에 전달된 메타데이터를 바탕으로 사용자의 요구에 맞는 정보를 생성할 수 있다.
- [0213] 디코딩된 텍스트 신호는 오버레이 렌더러에 전달되어 서브타이틀 등의 텍스트 기반의 오버레이 정보로써 처리될 수 있다. 필요한 경우 별도의 텍스트 후처리 과정이 포함될 수 있다.
- [0214] 도 15는 FLUS(Framework for Live Uplink Streaming) 아키텍처의 예를 개략적으로 나타낸다.
- [0215] 도 14(도 14a 및 도 14b)에서 상술한 송신단 및 수신단의 세부 블록은 FLUS(Framework for Live Uplink Streaming)에서의 소스(source)와 싱크(sink)의 기능으로 각각 분류할 수 있다.
- [0216] 상술한 송신단 및 수신단의 세부 블록이 FLUS에서의 소스(source)와 싱크(sink)의 기능으로 분류되는 경우, 도 14와 같이 360 비디오 획득 장치에서 소스(source)의 기능을 구현하고, 네트워크 상에서 싱크(sink)의 기능을 구현하거나, 혹은 네트워크 노드 내에서 소스/싱크를 각각 구현할 수 있다.
- [0217] 상술한 아키텍처를 기반으로 한 송수신 처리 과정을 개략적으로 나타내면 예를 들어 다음 도 15 및 도 16과 같이 도시될 수 있다. 도 15 및 도 16의 송수신 처리 과정은 영상 신호 처리 과정을 기준으로 기술하며, 음성 혹은 텍스트와 같은 다른 신호를 처리하는 경우 일부 부분(ex. 스티처, 프로젝션 처리부, 패키징 처리부, 서브픽처 처리부, 언패킹/셀렉션, 렌더링, 컴포지션, 뷰포트 생성 등)은 생략될 수 있고, 또는 음성 혹은 텍스트 처리 과정에 맞도록 변경하여 처리될 수 있다.
- [0218] 도 16은 3DoF+ 송신단에서의 구성을 개략적으로 나타낸다.
- [0219] 도 16을 참조하면, 송신단(360 비디오 전송 장치)에서는 입력된 데이터가 카메라 출력 영상인 경우 스피어(sphere) 영상 구성을 위한 스티칭을 위치/시점/컴포넌트 별로 진행할 수 있다. 위치/시점/컴포넌트 별 스피어(sphere) 영상이 구성되면 코딩을 위해 2D 영상으로 프로젝션을 수행할 수 있다. 어플리케이션에 따라 복수의 영상을 통합 영상으로 만들기 위한 패키징 혹은 세부 영역의 영상으로 나누는 서브 픽처로 생성할 수 있다. 전송한 바와 같이 리전별 패키징 과정은 선택적(optional) 과정으로서 수행되지 않을 수 있으며, 이 경우 패키징 처리부

는 생략될 수 있다. 입력된 데이터가 영상/음성/텍스트 추가 정보인 경우 추가 정보를 중심 영상에 추가하여 디스플레이 하는 방법을 알려줄 수 있으며, 추가 데이터도 함께 전송할 수 있다. 생성된 영상 및 추가된 데이터를 압축하여 비트 스트림으로 생성하는 인코딩 과정을 거쳐 전송 혹은 저장을 위한 파일 포맷으로 변환하는 인캡슐레이션 과정을 거칠 수 있다. 이 때 어플리케이션 혹은 시스템의 요구에 따라 수신부에서 필요로하는 파일을 추출하는 과정이 처리될 수 있다. 생성된 비트스트림은 전송처리부를 통해 전송 포맷으로 변환된 후 전송될 수 있다. 이 때, 송신측 피드백 처리부에서는 수신단에서 전달된 정보를 바탕으로 위치/시점/방향 정보와 필요한 메타데이터를 처리하여 관련된 송신부에서 처리하도록 전달할 수 있다.

[0220] 도 17은 3DoF+ 수신단에서의 구성을 개략적으로 나타낸다.

[0221] 도 17을 참조하면, 수신단(360 비디오 수신 장치)에서는 송신단에서 전달한 비트스트림을 수신한 후 필요한 작업을 추출할 수 있다. 생성된 파일 포맷 내의 영상 스트림을 피드백 처리부에서 전달하는 위치/시점/방향 정보 및 비디오 메타데이터를 이용하여 선별하며, 선별된 비트스트림을 디코더를 통해 영상 정보로 재구성할 수 있다. 패키징된 영상의 경우 메타데이터를 통해 전달된 패키징 정보를 바탕으로 언패킹을 수행할 수 있다. 송신단에서 패키징 과정이 생략된 경우, 수신단의 언패킹 또한 생략될 수 있다. 또한 필요에 따라 피드백 처리부에서 전달된 위치(viewpoint)/시점(viewing position)/방향(viewing orientation)에 적합한 영상 및 필요한 컴포넌트를 선택하는 과정을 수행할 수 있다. 영상의 텍스처, 맵스, 오버레이 정보 등을 재생하기 적합한 포맷으로 재구성하는 렌더링 과정을 수행할 수 있다. 최종 영상을 생성하기에 앞서 서로 다른 레이어의 정보를 통합하는 컴포지션 과정을 거칠 수 있으며, 디스플레이 뷰포트(viewport)에 적합한 영상을 생성하여 재생할 수 있다.

[0222] 도 18은 360 비디오의 오버레이를 나타낸 모습의 예이다.

[0223] 본 발명의 일 실시예는 VR 미디어 서비스를 위한 오버레이 방법 및 이를 위한 시그널링 방법에 관한 것으로, 360 비디오를 저작하는 에디터(editor)는 오버레이들을 360 비디오 상에 배치할 수 있다.

[0224] 일 실시예에서는 배치된 오버레이들의 정보를 기반으로 메타데이터가 생성될 수 있으며, 이러한 내용은 3DoF+ 송신단의 데이터 입력부로 전달되고 메타데이터 처리부를 통해 데이터 인코더 또는 인캡슐레이션 처리부로 전달되어 3DoF+ 수신단으로 전송될 수 있다. 3DoF+ 수신부에서는 전달받은 비트스트림에서 필요한 파일을 추출하고 디캡슐레이션 처리부와 메타데이터 파서를 통해 오버레이와 관련된 메타데이터를 추출하여 렌더링으로 전달하고 렌더링에서 오버레이를 렌더링하고 컴포지션 과정을 거쳐서 화면에 출력될 수 있다.

[0225] 전체 아키텍처 상에서는 오버레이 되는 미디어(텍스트, 비주얼 및 오디오 등)와 함께 저작자의 입력이 입력부로 전달될 수 있고, 컴포지션 제너레이션(Composition Generation)을 통해 오버레이 위치/크기/렌더링 속성 관련 메타데이터가 생성될 수 있다. 미디어는 패키징되어 비디오/이미지 인코딩 과정을 거쳐서 파일/세그먼트 인캡슐레이션 처리가 되어 수신단으로 전송될 수 있고, 텍스트는 텍스트 인코딩 되고, 오디오는 오디오 인코딩 되어 비디오/이미지 인코딩 과정을 거쳐서 파일/세그먼트 인캡슐레이션 처리가 되어 수신단으로 전송될 수 있다. 수신부에서는 전달받은 비트스트림에서 필요한 파일을 추출하고 파일/세그먼트 디캡슐레이션 처리부와 메타데이터 파서를 통해 오버레이와 관련된 메타데이터를 추출하고, 비디오/이미지, 텍스트, 오디오 디코더를 통해 오버레이 될 미디어를 디코딩할 수 있다. 오버레이 관련 추출된 메타데이터와 미디어 데이터를 오버레이 렌더링으로 전달해서 오버레이를 렌더링하고 컴포지션 과정을 거쳐 사용자 뷰포트 렌더링을 수행하여 화면에 출력될 수 있다.

[0226] 일 실시예에 따르면, VR 미디어 서비스에서 오버레이를 제공하기 위해서는 기존의 일반적인 비디오 서비스와의 차이로 인해 다음과 같은 경우들을 고려하여 확장할 수 있다. 여기서, 오버레이는 그래픽(Graphic), 이미지(image), SVG(scalable vector graphic), 타임드 텍스트(timed text)(TTML(Tagged Text Markup Language), WebVTT(Web Video Text Tracks), IMCS1(Internet Media Subtitles and Captions 1.0.1) 및 EBU-TT-D(European Broadcasting Union Timed Text part D) 등) 및 비트맵 서브타이틀 데이터(bitmap subtitle data) 등 중 적어도 하나를 포함할 수 있으나, 이에 한정되는 것은 아니다.

[0227] 따라서, 일 실시예는 오버레이 미디어와 관련 데이터 정보가 어디에 어떻게 저장되는지에 대한 오버레이 미디어 트랙 구성, 오버레이 미디어가 어떻게 패키징 되는지에 대한 오버레이 미디어 패키징 정보, 오버레이 미디어에 프로젝션이 적용 되는지에 대한 오버레이 미디어 프로젝션 정보, 오버레이 미디어 프로젝션과 패키징 정보 시그널링, 오버레이 미디어 트랙과 VR 미디어 트랙의 링킹(Linking) 방법, VR 미디어가 플레이 될 때 오버레이를 언제, 어디에 위치 시키고, 어느 정도 크기로 보이게 할지에 대한 오버레이 렌더링 위치/크기 정보, 오버레이를 투명하게 보이게 할지 및 오버레이를 어떻게 블렌딩 할지에 대한 오버레이 렌더링 속성 정보, 어떤 오버레이 렌더링의 기타 기능을 제공할 수 있을지에 대한 오버레이 미셀레이니어스(Miscellaneous) 정보, 오버레이와 인터렉션이

가능한지 및 어느 범위에서 가능한지에 대한 오버레이 인터랙션 정보, 동적인 오버레이 메타데이터 시그널링, 오버레이 메타데이터 트랙과 오버레이 미디어 트랙과의 링킹(Linking) 방법 및 오버레이 미디어 트랙 상에 오버레이 메타데이터 시그널링 방법을 제안할 수 있다.

- [0228] 도 19는 오버레이 미디어 트랙 상에 오버레이 메타데이터 시그널링의 예를 나타낸다.
- [0229] VR 미디어 파일내에 오버레이 트랙을 구성하는 방안은 다음 두 가지를 모두 지원할 수 있다. 도 19를 참조하면, 파일#1과 같이 하나 이상의 오버레이 미디어 트랙, 그리고 오버레이 미디어와 관련된 메타데이터 등을 포함할 수 있다. 파일#2와 같이 오버레이 미디어는 VR 미디어 트랙 내에 포함되어서 하나의 트랙에 패키징 될 수 있다.
- [0230] 도 20은 VR 미디어 파일 내에 오버레이 트랙의 구성을 나타낸 예이다.
- [0231] 도 20를 참조하면, 파일 #1은 VR 미디어와 오버레이 미디어가 각각의 트랙으로 분리되어 있는 형태일 수 있다. 즉, 오버레이 미디어에 대응하는 이미지는 VR 미디어와 분리되어 있을 수 있다. 파일 #2는 VR 미디어 트랙 내에 VR 미디어와 오버레이 미디어들이 함께 패키징되어 있는 형태일 수 있다. 즉, 오버레이 미디어에 대응하는 이미지는 VR 미디어에 포함되어 있을 수 있다.
- [0232] 도 21은 오버레이 미디어 트랙 상에 오버레이 메타데이터 시그널링의 다른 예를 나타낸다.
- [0233] 도 21을 참조하면, 파일 #1의 경우, 오버레이 미디어 트랙은 오버레이 미디어의 프로젝션 정보와 패키징 정보를 포함할 수 있다. 파일 #2의 경우, 오버레이 미디어가 VR 미디어 트랙에 포함될 수 있다. 여기서, 오버레이 미디어가 어떻게 패키징이 되었는지에 대한 정보는 파일 #1과 같이 동일하게 필요할 수 있다. 다만, 오버레이 프로젝션 정보는 파일 #1과 다르게 다음의 두 가지를 지원할 수 있다.
- [0234] 첫 번째로, 오버레이 미디어는 VR 미디어 트랙의 프로젝션 정보를 공유할 수 있다. 즉, VR 미디어 트랙에 포함된 모든 오버레이 미디어는 VR 미디어 트랙에 적용된 프로젝션을 적용된 상태로 저장된다는 가정이 필요할 수 있다. 두 번째로, 파일 #1과 같이 각각의 패키징된 오버레이들에 대한 프로젝션 정보는 별도로 포함될 수 있다. 이 경우, VR 미디어 트랙에 포함된 오버레이들은 각각 다른 프로젝션 타입을 가질 수도 있으며, VR 미디어 트랙의 프로젝션과 일치할 필요가 없다.
- [0235] 도 22는 파일 #1의 경우 가능한 4가지의 오버레이 미디어 패키징 구성을 나타낸 예이다.
- [0236] 파일 #1의 경우, 하나의 오버레이 미디어 트랙에 다음과 같은 네 가지 경우로 오버레이 미디어를 패키징할 수 있다. 도 22에서 이미지는 오버레이 미디어를 지칭할 수 있다. 도 22를 참조하면, 제1 케이스(Case 1)는 오버레이 1개를 오버레이 미디어 1개로 패키징하는 경우일 수 있다. 즉, 하나의 이미지에 하나의 오버레이가 포함될 수 있다. 제2 케이스(Case 2)는 오버레이 N개를 오버레이 미디어 N개로 패키징하는 경우일 수 있다. 즉, 하나의 이미지에 하나의 오버레이가 포함될 수 있으며, 복수의 이미지가 이용되는 경우일 수 있다. 이러한 케이스를 서브 샘플 케이스라 지칭할 수 있다. 제3 케이스(Case 3)는 오버레이 N개를 오버레이 미디어 1개로 패키징하는 경우일 수 있다. 즉, 하나의 이미지에 복수의 오버레이가 포함될 수 있으며, 이러한 케이스를 통합 패키징 케이스라고 지칭할 수 있다. 제4 케이스(Case 4)는 오버레이 N개를 오버레이 미디어 M개로 패키징하는 경우일 수 있다. 즉, 하나의 이미지에 복수의 오버레이가 포함될 수 있고, 복수의 이미지가 이용되는 경우일 수 있다. 이러한 케이스를 통합 패키징 + 서브 샘플 케이스라고 지칭할 수 있다. 여기서, N 및 M은 1보다 큰 자연수일 수 있으며, 서로 다를 수 있다.
- [0237] 도 23은 파일 #1의 경우 트랙 내에 구조를 나타낸 예이다.
- [0238] 도 23을 참조하면, 일 실시예에서 트랙은 샘플을 포함할 수 있다. 미디어가 비디오인 경우 샘플은 특정 시간의 한 프레임에 대한 데이터일 수 있고, 이미지인 경우 샘플은 특정 시간의 이미지 데이터일 수 있다. 여기서, 샘플은 서브 샘플로 구성될 수 있다. 서브 샘플은 특정 시간을 위한 데이터가 여러 개 동시에 존재할 경우 구성될 수 있다.
- [0239] 여기서, 통합 패키징은 여러 개의 오버레이 미디어를 하나의 통합된 형태로 패키징하여 하나의 트랙을 하나의 샘플 또는 서브 샘플로 구성하는 방법을 의미할 수 있고, 상술한 제3 케이스를 의미할 수 있다.
- [0240] 일 실시예에서 하나의 오버레이 미디어 트랙에 여러 개의 오버레이 미디어들을 통합 패키징을 할 수 있는 방법은 다음의 두 가지 방법을 이용할 수 있다.
- [0241] 첫 번째는 텍스처 아틀라스(texture atlas) 방법으로 렌더링 되는 위치에 상관없이 프로젝션 여부와 관계없이 오버레이 미디어들을 하나의 텍스처로 패키징하는 방법일 수 있다. 두 번째는 리전별 패키징 방법으로 오버레이를

송신기에서 미리 특정 위치로 렌더링하고, 프로젝션 타입에 맞추어 프로젝션시킨 오버레이의 프로젝션된 픽처(Projected Picture)를 영역 기반으로 패킹하는 방법일 수 있다.

- [0242] 일 실시예에서 각 오버레이 미디어 트랙은 하나의 오버레이 미디어를 담고 있는 미디어 트랙일 수도 있고, 여러 개의 오버레이 미디어를 서브 샘플을 통해 가지고 있는 트랙일 수도 있고, 여러 개의 오버레이 미디어들이 하나의 샘플로 통합된 형태의 미디어 트랙일 수도 있다. 이러한 다양한 형태의 오버레이 미디어 트랙들이 하나의 파일 내에 공존할 수 있다.
- [0243] 일 실시예에서는 오버레이 미디어 패킹을 위해 텍스처 아틀라스 방법을 적용할 수 있다. 실시간 컴퓨터 그래픽스에서, 텍스처 아틀라스는 작은 텍스처들을 모아 함께 패킹하여 하나의 큰 텍스처로 만들어 사용하는 방법을 의미할 수 있으며, 하나로 합쳐진 큰 텍스처 자체를 텍스처 아틀라스라고 지칭할 수 있다. 텍스처 아틀라스는 같은 크기의 서브 텍스처들로 구성될 수 있고, 다양한 크기의 텍스처들로 구성될 수도 있다. 또는, 오버레이 미디어의 해상도를 유지하도록 구성할 수 있다. 각 서브 텍스처들은 패킹된 위치 정보 값을 가지고 콘텐츠를 추출할 수 있다.
- [0244] 도 24는 텍스처 아틀라스를 생성하는 방법의 순서도를 나타낸 예이다.
- [0245] 도 24를 참조하면, 텍스처 아틀라스를 생성하는 방법은 우선 패킹하여야 할 오버레이 미디어(이미지/비디오 프레임)가 존재하는 경우, 텍스처 아틀라스 내의 사용 가능한 공간을 검색할 수 있다. 여기서, 오버레이 미디어와 사용 가능한 공간을 기반으로 공간이 충분한지 판단할 수 있으며, 충분한 경우, 공간 내에 패킹할 수 있고, 충분하지 않은 경우 텍스처 아틀라스의 사이즈를 증가시킨 후 공간 내에 패킹할 수 있다.
- [0246] 도 25는 텍스처 아틀라스를 생성하는 모습을 나타낸 예이다.
- [0247] 상술한 과정을 도시화하면 도 25와 같이 나타낼 수 있다. 즉, 사용 가능한 공간에 오버레이 미디어를 포함시키며 하나의 이미지를 생성할 수 있으며, 반복 수행을 통해 하나의 이미지에 복수의 오버레이 미디어가 포함될 수 있다. 여기서, 사용 가능한 공간은 하나의 이미지 내에 오버레이 미디어가 포함되지 않은 공간을 지칭할 수 있다.
- [0248] 상술한 바와 같이 패킹을 수행하는 경우에는 수신기의 디코더의 개수를 감소시킬 수 있고, 렌더링 시에 메모리 참조의 근접성으로 성능의 이점이 있을 수 있다. 또한, 수신기의 성능에 따라 텍스처 아틀라스가 포함할 수 있는 서브 텍스처의 크기를 조정할 수 있도록 구성될 수도 있다. 또한, mip맵핑(Mipmapping) 및 텍스처 압축 과정에서 발생할 수 있는 부정적 요소들을 방지하기 위해 서브 텍스처들 사이에 가드 밴드(guard band)를 구성할 수도 있다. 여기서, 가드 밴드는 각 오버레이 미디어를 패킹할 때 주변 몇 개의 픽셀들을 비워 놓고 주변의 빈(empty) 픽셀 개수를 명시할 수 있다.
- [0249] 도 26은 VR 미디어의 리전별 패킹을 설명하기 위한 도면이다.
- [0250] 일 실시예에서는 오버레이 미디어 패킹을 위해 리전별(region-wise) 패킹 방법이 적용될 수 있다. 리전별 패킹 방법은 VR 미디어(또는 360 미디어)에 프로젝션이 적용된 상태인 프로젝션된 픽처(projected picture)에서 전체 영역을 구간으로 나누고, 구간을 중요도에 따라 다른 해상도로 패킹할 수 있다. 여기서, 중요도는 예를 들어, 사용자 뷰포트 구간에 따라 결정될 수 있다. 즉, 도 32를 참조하면, c의 프로젝션된 픽처 내의 구간들 중 1, 2 및 3 구간을 패킹하여 d의 팩드 픽처를 생성할 수 있다.
- [0251] 오버레이를 위한 리전별 패킹은 송신기에서 미리 렌더링된 또는 프로젝션된 결과에 맞추어 오버레이 미디어가 구성되는 방식일 수 있다. 다시 말해, 오버레이 미디어는 렌더링되는 위치, 크기 및 프로젝션이 적용된 형태로 재구성될 수 있다. 이러한 방식은 번인(Burn-In)이라 지칭할 수 있다. 이러한 번인 방법은 유연성(flexibility)이 떨어질 수 있는 단점이 존재하나, 수신기의 렌더러를 단순화할 수 있는 장점이 있다.
- [0252] 일 실시예에서는 전체 360도에 대하여 프로젝션된 360 미디어와 같은 형태의 360 오버레이 미디어가 생성될 수 있으며, 영역의 중요도나 미디어 존재 여부에 따라 오버레이 미디어 결과에 리전별 패킹이 수행될 수 있다.
- [0253] 여기서, 프로젝션된 오버레이 미디어의 형태는 항상 사각형의 형태가 되는 것은 아니며, 프로젝션된 오버레이의 형태를 고려하여 패킹된 위치 값을 명시할 수 있다. 이와 관련하여 일 실시예에서는 다음의 두 가지 방법을 지원할 수 있다.
- [0254] 첫 번째로, 프로젝션된 오버레이 미디어를 감싸는 가장 작은 2차원 사각형 바운딩 박스를 오버레이 미디어 영역으로 설정할 수 있고, Projected Picture(렌더링되는 위치 및 크기) 내의 위치를 재조정할 수도 있다. 즉, 렌더

링되는 위치 및 크기를 고려하여 프로젝션된 픽처 내의 위치를 재조정할 수 있다. 두 번째로, 다각형 형태를 표현할 수도 있다. 이 경우, 영역을 수평/수직으로 분할할 수 있고, 각각의 위치 점들의 정보를 명시할 수 있다.

- [0255] 도 27은 오버레이 미디어의 리전별 패킹 방법의 순서도를 나타낸 예이다.
- [0256] 도 27을 참조하면, 오버레이 미디어의 리전별 패킹 방법은 우선 오버레이 미디어들(이미지/비디오 프레임)이 존재하는 경우, 이에 렌더링될 위치/크기/프로젝션 타입을 적용할 수 있으며, 오버레이 360 프로젝션된 픽처(overlay 360 projected picture)로 구성할 수 있다. 이후, 영역의 중요도에 따라 오버레이 품질 랭킹(quality ranking)을 설정할 수 있고, 적용할 수 있으나, 이는 선택적으로 수행될 수 있다.
- [0257] 도 28은 오버레이 미디어의 리전별 패킹 모습을 나타낸 예이다.
- [0258] 상술한 과정을 도시화하면 도 28과 같이 나타날 수 있다. 즉, 오버레이 미디어에 렌더링될 위치/크기/프로젝션 타입을 적용하여 오버레이 360 프로젝션된 픽처인 렌더링된 오버레이 미디어 트랙을 생성할 수 있으며, 이를 VR 미디어 트랙과 함께 패킹할 수 있다.
- [0259] 도 29는 파일 #2인 경우 오버레이 미디어 패킹의 구성을 나타낸 예이다.
- [0260] 파일 #2의 경우, VR 미디어 트랙 내에 오버레이 미디어들을 도 29와 같이 세 가지의 경우로 패킹될 수 있다. 제 1 케이스(Case 1)는 VR 미디어가 프로젝션 스킵이 ERP이고, 리전별 패킹 과정을 통한 팩드 픽처(Region Wise Packed Picture)이고, 오버레이 미디어가 프로젝션 스킵이 ERP이고, 리전별 패킹된 경우일 수 있다. 제 2 케이스(Case 2)는 VR 미디어가 프로젝션 스킵이 ERP이고, 리전별 패킹 과정을 통한 팩드 픽처이고, 오버레이 미디어가 프로젝션되지 않고(None), 텍스처 아틀라스 패킹된 경우일 수 있다. 제 3 케이스(Case 3)는 VR 미디어가 프로젝션 스킵이 ERP이고, 프로젝션된 픽처(projected picture)이고, 오버레이 미디어가 프로젝션되지 않고, 텍스처 아틀라스 패킹된 경우일 수 있다.
- [0261] 즉, VR 미디어 트랙의 팩드 픽처(packed picture) 또는 프로젝션된 픽처(projected picture) 내에 VR 미디어와 오버레이 미디어들이 동시에 존재할 수 있다. 이러한 경우, 전체 픽처에서 오버레이 미디어들을 담고 있는 영역에 대한 정보를 명시할 수 있다. 영역에 대한 정보는 좌측(left) 포인트 위치 값, 상측(top) 포인트 위치 값, 너비(width) 값 및 높이(height) 값 중 적어도 하나를 포함할 수 있다.
- [0262] 도 30은 파일 #2인 경우 VR 미디어 트랙이 VR 미디어의 일부분과 오버레이 미디어로 패킹되는 모습을 나타낸 예이다.
- [0263] 일 실시예에서 VR 미디어는 분할되어 트랙에 저장될 수 있고, 각 VR 미디어 트랙들에 오버레이 미디어들이 저장되는 경우, 각 오버레이 미디어는 오버레이가 디스플레이되는 위치에 따라 각각의 VR 미디어 트랙에 함께 패킹될 수도 있다. 즉, 오버레이가 속한 디스플레이될 VR 미디어 트랙에 함께 패킹되는 경우에 해당할 수 있다. 또는 오버레이가 디스플레이될 VR 미디어 트랙에 함께 패킹되는 경우에 해당할 수 있다.
- [0264] 도 30을 참조하면, VR 미디어 트랙은 VR 미디어의 일부분 및 오버레이 미디어를 포함할 수 있다. 또는, 각 오버레이 미디어는 전체 VR 미디어 중 디스플레이될 일부분과 함께 패킹될 수 있다. 여기서, 각 트랙은 서로 다른 패킹 방법이 적용될 수 있다. 예를 들어, VR 미디어 트랙 #1의 경우 오버레이 1개를 1개의 이미지에 포함시켜 패킹을 수행할 수 있고, VR 미디어 트랙 #2 및 #3의 경우 적어도 하나의 오버레이를 텍스처 아틀라스 패킹 방법을 통해 패킹을 수행할 수 있다.
- [0265] 도 31은 파일 # 2인 경우 VR 미디어 트랙이 VR 미디어와 오버레이 미디어로 패킹되는 모습을 나타낸 예이다.
- [0266] 일 실시예에서는 VR 미디어 트랙에 오버레이가 함께 패킹되는 경우, 도 37과 같이 구성될 수 있다. 여기서, 오버레이 미디어가 저장되는 영역의 위치는 항상 VR 미디어의 오른쪽에만 존재하는 것은 아니며, 도 37과 같이 다양한 위치에 올 수 있다. 예를 들어, VR 미디어의 오른쪽, 왼쪽 및 하측에도 존재할 수 있다. 다만, 이러한 부분들은 명시될 수 있다.
- [0267] 도 32는 오버레이 프로젝션 지원 방법을 설명하는 순서도의 예이다.
- [0268] 일 실시예에서 오버레이 미디어 트랙은 각 오버레이에 적용된 프로젝션 정보를 포함할 수 있다. 또한, 오버레이 미디어 트랙에 여러 오버레이들이 패킹된 경우, 각 오버레이에 대한 프로젝션 정보가 명시될 수 있다. 여기서, 오버레이에 적용될 수 있는 프로젝션은 None, ERP(EquiRectangular Projection) 및 CMP(CubeMap Projection) 중 하나일 수 있다. 다만, CMP는 리전별 패킹이 적용되었을 경우에만 지원될 수 있으며, 그 외에 오버레이에 적용된 프로젝션 CMP는 None과 같을 수 있다.

- [0269] 또한, 오버레이 미디어 프로젝션 정보와 메타데이터에서 명시된 영역 정보가 매칭되지 않을 수도 있다. 예를 들어, 오버레이 미디어는 ERP로 프로젝션 되었으나, 해당 오버레이 미디어를 뷰포트 상에 렌더링하도록 렌더링 위치가 설정되었다면, 수신기는 ERP로 프로젝션된 오버레이 미디어를 un-projection 시켜서 렌더링할 수도 있다.
- [0270] 도 32를 참조하면, 일 실시예는 수신기가 오버레이 렌더링을 지원하는지 판단할 수 있으며, 오버레이 렌더링을 지원하지 않는 경우 메인 VR 미디어를 렌더링할 수 있고, 사용자 뷰포트를 렌더링할 수 있다.
- [0271] 다만, 수신기가 오버레이 렌더링을 지원하는 경우, 오버레이 미디어의 프로젝션, 패킹, 렌더링 관련 메타데이터를 파악할 수 있으며, 오버레이 미디어가 VR 미디어 트랙 내에 존재하는지 판단할 수 있다. 여기서, 오버레이 미디어가 VR 미디어 트랙 내에 존재하는 경우, 메인 VR 미디어 영역과 오버레이 미디어의 패킹 영역을 분리할 수 있고, 메인 VR 미디어를 렌더링할 수 있다. 다만, 오버레이 미디어가 VR 미디어 트랙 내에 존재하지 않는 경우, 분리 과정을 생략하고 메인 VR 미디어를 렌더링할 수 있다.
- [0272] 이후, 일 실시예는 텍스처 아틀라스 렌더링을 지원하는지 판단할 수 있다. 여기서, 텍스처 아틀라스 렌더링을 지원하는 경우, 텍스처 좌표값을 0 ~ 1.0 단위 값으로 변경할 수 있고, 텍스처 아틀라스 렌더링을 지원하지 않는 경우, 패킹 좌표를 기반으로 오버레이 미디어 콘텐츠를 언패킹할 수 있다.
- [0273] 일 실시예는 오버레이 미디어의 프로젝션과 렌더링 시 영역에서 기대되는 프로젝션이 일치하는지 판단할 수 있다. 여기서, 일치하는 경우, 오버레이 미디어를 렌더링할 수 있고, 일치하지 않는 경우, 프로젝션 재구성 및 적용을 수행한 후 오버레이 미디어를 렌더링할 수 있다. 여기서, 프로젝션 재구성 및 적용 시 프래그먼트 웨이더에 프로젝션 조정 기능 및 옵션 설정도 가능할 수 있다. 이후, 사용자 뷰포트를 렌더링할 수 있다.
- [0274] 일 실시예에서 오버레이 미디어 패킹 및 프로젝션 정보는 오버레이 미디어 패킹 및 프로젝션 관련 정보로 지칭할 수 있고, 메타데이터로써 시그널링될 수 있으므로, 메타데이터라 지칭할 수도 있다. 또는 메타데이터에서 OverlayMediaPackingStruct 내에 포함될 수 있다. 여기서, 오버레이 미디어 패킹 및 프로젝션 정보의 구조는 메타데이터 구조로 지칭할 수도 있다. OverlayMediaPackingStruct는 예를 들어, 표 1과 같이 다음을 포함할 수 있다.

표 1

```

aligned(8) class OverlayMediaPackingStruct {
    unsigned int(16) num_overlays;
    unsigned int(5) num_regions;
    for(i = 0; i < num_regions; i++) {
        unsigned int(8) overlay_region_id[i];
        unsigned int(16) overlay_region_width[i];
        unsigned int(16) overlay_region_height[i];
        unsigned int(16) overlay_region_top[i];
        unsigned int(16) overlay_region_left[i];
    }
    for (i = 0; i < num_overlays; i++) {
        unsigned int(16) overlay_source_id[i];
        unsigned int(5) projection_type[i];
        unsigned int(3) packing_type;
        unsigned int(8) overlay_region_id[i];
        if(packing_type != 0) {
            unsigned int(1) guard_band_flag[i];
            if (packing_type == 1)
                TextureAtlasPacking(i);
            else if (packing_type == 2)
                RectRegionPacking(i);
            else if (packing_type == 3)
                PolygonRegionPacking(i);
            if (guard_band_flag[i])
                GuardBand(i);
        }
    }
}
    
```

- [0275]
- [0276] 표 1에서, num_overlays 필드는 오버레이 미디어에 포함된 또는 패킹된 오버레이 개수를 지시할 수 있고,

packing_type 필드는 오버레이 미디어 패킹 타입을 지시할 수 있다. 여기서, packing_type 필드 값이 0인 경우 통합 패킹이 적용되지 않음(none)을 지시할 수 있고, 1인 경우 텍스처 아틀라스 패킹이 적용됨을 지시할 수 있고, 2인 경우 사각형(rectangle) 모양의 리전별 패킹이 적용됨을 지시할 수 있고, 3인 경우, 다각형(polygon) 모양의 리전별 패킹이 적용됨을 지시할 수 있다.

[0277] 또한, num_regions 필드는 오버레이가 패킹되어 있는 영역의 개수를 지시할 수 있고, overlay_region_id 필드는 패킹 영역의 식별자를 지시할 수 있다. 또한, overlay_region_width 필드, overlay_region_height 필드, overlay_region_left 필드 및 overlay_region_top 필드는 패킹 영역의 크기 및 위치 정보를 지시할 수 있다. 즉, 각각은 패킹 영역의 너비 값, 높이 값, 왼쪽 위치 값 및 상단 위치 값을 지시할 수 있다.

[0278] 또한, overlay_source_id 필드는 각 오버레이 미디어의 식별자를 지시할 수 있고, projection_type 필드는 각 오버레이 미디어에 적용된 프로젝션 타입을 지시할 수 있다. 여기서, projection_type 필드 값이 0인 경우 프로젝션이 적용되지 않음(none)을 지시할 수 있고, 1인 경우 ERP(Equirectangular projection) 이 적용됨을 지시할 수 있고, 2인 경우 CMP(Cubemap projection)이 적용됨을 지시할 수 있다.

[0279] 표 1에서 두 번째 for 문 내의 overlay_region_id 필드는 상술한 바와 동일하게 패킹 영역의 식별자를 지시할 수 있으나, 어떠한 오버레이 패킹 영역에 오버레이 미디어가 저장되어 있는지를 명시하기 위해 명시될 수 있다.

[0280] 또한, guard_band_flag 필드는 패킹이 적용되었을 경우, 서브 텍스처 가드 밴드의 존재 여부에 대한 플래그를 의미할 수 있다.

[0281] 표 1에서 TextureAtlasPacking은 텍스처 아틀라스 패킹에 관한 정보 또는 메타데이터를 포함할 수 있으며, packing_type 필드 값이 1인 경우(packing_type == 1)에 포함될 수 있다. TextureAtlasPacking은 표 2와 같이 다음을 포함할 수 있다.

표 2

```
aligned(8) class TextureAtlasPacking(i) {
    unsigned int(16) width[i];
    unsigned int(16) height[i];
    unsigned int(16) top[i];
    unsigned int(16) left[i];
    unsigned int(3) transform_type[i];
    bit(5) reserved = 0;
}
```

[0282]

[0283] 표 2에서, width 필드, height 필드, top 필드 및 left 필드는 텍스처 아틀라스 내에서의 위치 및 크기 정보를 지시할 수 있다. 또는 텍스처 아틀라스 내에서의 오버레이 미디어의 위치 및 크기 정보를 지시할 수 있다. 즉, 각각은 아틀라스 내에서 오버레이 미디어의 너비 값, 높이 값, 상측 포인트의 위치 값 및 좌측 포인트의 위치 값을 지시할 수 있다.

[0284] 또한, transform_type 필드는 텍스처 아틀라스 내에서의 회전 값을 지시할 수 있다. 또는 아틀라스 내에서의 오버레이 미디어의 회전 값을 지시할 수 있다. 여기서, transform_type 필드 값이 0인 경우 회전 없음을, 1인 경우 수평 미러링을, 2인 경우 180도 회전을, 3인 경우 180도 회전과 수평 미러링을, 4인 경우 90도 회전과 수평 미러링을, 5인 경우 90도 회전을, 6인 경우 270도 회전과 수평 미러링을, 7인 경우 270회전을 지시할 수 있다. 여기서, 회전은 시계 방향일 수도 있고, 반시계 방향일 수도 있다.

[0285] 상술한 표 1에서 RectRegionPacking은 사각형 모양의 리전별 패킹에 관한 정보 또는 메타데이터를 포함할 수 있으며, packing_type 필드 값이 2인 경우(packing_type == 2)에 포함될 수 있다. RectRegionPacking은 표 3와 같이 다음을 포함할 수 있다.

표 3

```
aligned(8) class RectRegionPacking(i) {
    unsigned int(32) proj_reg_width[i];
    unsigned int(32) proj_reg_height[i];
    unsigned int(32) proj_reg_top[i];
    unsigned int(32) proj_reg_left[i];
    unsigned int(3) transform_type[i];
    bit(5) reserved = 0;
    unsigned int(16) packed_reg_width[i];
    unsigned int(16) packed_reg_height[i];
    unsigned int(16) packed_reg_top[i];
    unsigned int(16) packed_reg_left[i];
}
```

[0286]

[0287]

표 3에서, proj_reg_width 필드, proj_reg_height 필드, proj_reg_top 필드 및 proj_reg_left 필드는 프로젝션된 픽처(projected picture)에서의 위치 및 크기 정보를 지시할 수 있다. 즉, 각각은 프로젝션된 픽처에서 오버레이 미디어의 너비 값, 높이 값, 상측 포인트의 위치 값 및 좌측 포인트의 위치 값을 지시할 수 있다. transform_type 필드는 프로젝션된 픽처에서의 회전 값을 지시할 수 있으며, transform_type 필드 값에 따라 지시하는 바는 표 2에서와 동일할 수도 있으나, 다를 수도 있다.

[0288]

또한, packed_reg_width 필드, packed_reg_height 필드, packed_reg_top 필드 및 packed_reg_left 필드는 팩드 픽처(packed picture)에서의 위치 및 크기 정보를 지시할 수 있다. 즉, 각각은 팩드 픽처에서 오버레이 미디어의 너비 값, 높이 값, 상측 포인트의 위치 값 및 좌측 포인트의 위치 값을 지시할 수 있다.

[0289]

상술한 표 1에서 PolygonRegionPacking은 다각형 모양의 리전별 패킹에 관한 정보 또는 메타데이터를 포함할 수 있으며, packing_type 필드 값이 3인 경우(packing_type == 3)에 포함될 수 있다. 일 실시예는 프로젝션된 오버레이 형태가 사각형이 아닌 경우 다각형으로 패킹 영역을 명시할 수 있다. PolygonRegionPacking은 표 4와 같이 다음을 포함할 수 있다.

표 4

```
aligned(8) class PolygonRegionPacking(i) {
    unsigned int(8) num_rings;
    unsigned int(8) num_sectors;
    for (i = 0; i < num_rings; i++) {
        for (j = 0; j < num_sectors; j++) {
            unsigned int(16) proj_points_x[i][j];
            unsigned int(16) proj_points_y[i][j];
        }
    }
    unsigned int(3) transform_type[i];
    bit(5) reserved = 0;
    for (i = 0; i < num_rings; i++) {
        for (j = 0; j < num_sectors; j++) {
            unsigned int(16) packed_points_x[i][j];
            unsigned int(16) packed_points_y[i][j];
        }
    }
}
```

[0290]

- [0291] 표 4에서, num_rings 필드는 프로젝션된 픽처에서 수평으로 영역을 분할한 개수를 지시할 수 있고, num_sectors 필드는 프로젝션된 픽처에서 수직으로 영역을 분할한 개수를 지시할 수 있다. proj_points_x 필드 및 proj_points_y 필드는 각 분할 점들의 프로젝션된 픽처에서의 위치 값을 지시할 수 있다. 즉, 각각 프로젝션된 픽처에서의 분할 점들의 x축 포인트의 위치 값(또는 x축 좌표 값) 및 y축 포인트의 위치 값(또는 y축 좌표 값)을 지시할 수 있다. 또한, transform_type 필드는 프로젝션된 픽처에서의 회전 값을 지시할 수 있으며, transform_type 필드 값에 따라 지시하는 바는 표 2에서와 동일할 수도 있으나, 다를 수도 있다.
- [0292] packed_points_x 필드 및 packed_points_y 필드는 각 분할 점들의 팩드 픽처에서의 위치 값을 지시할 수 있다. 즉, 각각 팩드 픽처에서의 분할 점들의 x축 포인트의 위치 값(또는 x축 좌표 값) 및 y축 포인트의 위치 값(또는 y축 좌표 값)을 지시할 수 있다.
- [0293] 일 실시예는 스피어(sphere) 상에 오버레이 평면을 생성할 수도 있으며, 이 경우 수평 영역 분할 개수 및 수직 영역 분할 개수를 참고하여 서페이스 메쉬(surface mesh)를 생성할 수도 있다.
- [0294] 상술한 표 1에서 GuardBand는 가드 밴드에 관한 정보 또는 메타데이터를 포함할 수 있으며, guard_band_flag 필드 값이 1인 경우(guard_band_flag == 1)에 포함될 수 있다. GuardBand는 표 5와 같이 다음을 포함할 수 있다.

표 5

```
aligned(8) class GuardBand(i) {
    unsigned int(8) left_gb_width[i];
    unsigned int(8) right_gb_width[i];
    unsigned int(8) top_gb_height[i];
    unsigned int(8) bottom_gb_height[i];
}
```

- [0295]
- [0296] 표 5에서, left_gb_width 필드, right_gb_width 필드, top_gb_height 필드 및 bottom_gb_height 필드는 하나의 오버레이 텍스처의 주변에 가드밴드 영역을 설정하기 위한 좌우상하 갭(gap)에 대한 정보를 지시할 수 있다. 즉, 각각은 오버레이 텍스처의 좌측 갭의 너비 값, 오버레이 텍스처의 우측 갭의 너비 값, 오버레이 텍스처의 상측 갭의 높이 값 및 오버레이 텍스처의 하측 갭의 높이 값을 지시할 수 있다.
- [0297] 도 33은 오버레이 미디어 패킹 및 프로젝션에 관한 메타데이터 시그널링 예를 나타낸다.
- [0298] 도 33은 오버레이 미디어 트랙이 이미지인 경우를 나타낼 수 있다. 도 33을 참조하면, 일 실시예에서 파일 #1과 같은 경우 및 오버레이 미디어 트랙이 이미지인 경우, moov 박스 내의 오버레이 미디어 트랙은 ItemPropertyContainerBox를 포함할 수 있고, ItemPropertyContainerBox는 OverlayConfigProperty를 포함할 수 있다. 여기서, OverlayConfigProperty는 오버레이 미디어의 프로젝션 및 패킹 정보를 포함할 수 있다. 또는 오버레이 미디어의 프로젝션 및 패킹 정보를 포함하는 OverlayMediaPackingStruct()를 포함할 수 있다. 여기서, OverlayMediaPackingStruct()는 표 3과 같을 수 있다.
- [0299] 파일 #2와 같은 경우 및 VR 미디어 트랙이 이미지인 경우, VR 미디어 트랙은 ItemPropertyContainerBox를 포함할 수 있고, ItemPropertyContainerBox는 OverlayConfigProperty를 포함할 수 있다. 여기서, ItemPropertyContainerBox는 ProjectionFormatBox 등도 포함할 수 있다. OverlayConfigProperty는 오버레이 미디어의 프로젝션 및 패킹 정보를 포함할 수 있다. 또는 오버레이 미디어의 프로젝션 및 패킹 정보를 포함하는 OverlayMediaPackingStruct()를 포함할 수 있다. 여기서, OverlayMediaPackingStruct()는 표 1과 같을 수 있다.
- [0300] 상술한 OverlayConfigProperty는 도 33에서 나타낸 속성을 가질 수 있으며, 예를 들어, 표 6과 같이 다음을 포함할 수 있다.

표 6

```
class OverlayConfigProperty (type) extends ItemFullProperty ( 'ovly' , 0, 0) {
    OverlayMediaPackingStruct();
}
```

[0301]

[0302]

즉, OverlayConfigProperty는 박스 타입(box type)이 ovly일 수 있고, 컨테이너(container)가 ItempropertycontainerBox일 수 있고, 필수 항목(mandatory)은 아닐 수 있고(No), 수량(quantity)은 0 또는 1일 수 있다. 또한, OverlayMediaPackingStruct()는 오버레이 미디어의 프로젝션 및 패킹 정보를 포함할 수 있고, 표 1과 같을 수 있다.

[0303]

도 34는 오버레이 미디어 패킹 및 프로젝션에 관한 메타데이터 시그널링 다른 예를 나타낸다.

[0304]

도 34는 오버레이 미디어 트랙이 비디오인 경우를 나타낼 수 있다. 도 34를 참조하면, 일 실시예에서 파일 #1과 같은 경우 및 오버레이 미디어 트랙이 비디오인 경우(프로젝션되지 않은 경우), 오버레이 미디어 트랙은 SchemeInformationBox를 포함할 수 있고, SchemeInformationBox는 OverlayConfigBox를 포함할 수 있다. 여기서, OverlayConfigBox는 오버레이 미디어의 프로젝션 및 패킹 정보를 포함할 수 있다. 또는 오버레이 미디어의 프로젝션 및 패킹 정보를 포함하는 OverlayMediaPackingStruct()를 포함할 수 있다.

[0305]

일 실시예는 프로젝션되지 않은 오버레이 비디오를 SchemeInformationBox에 포함시키기 위하여 다음 오버레이 비디오 스킴(overlay video scheme) 'resv'를 생성할 수 있다. 제한된 비디오 샘플 타입 'resv'에 대한 오버레이 비디오 스킴은 디코딩된 픽처가 오버레이 비디오 픽처임은 명시할 수 있다.

[0306]

일 실시예는 RestrictedSchemeInfoBox 내의 SchemeTypeBox의 scheme_type 필드 값을 'oldv'로 설정할 수 있다. RestrictedSchemeInfoBox 내의 SchemeTypeBox의 스킴 타입(Scheme type)이 'oldv'일 때, OverlayConfigBox를 호출할 수 있다. 여기서, 'oldv' 스킴 타입은 오버레이 비디오를 위한 확장 가능한 스킴(open-ended scheme) 타입으로 정의될 수 있다. 이 경우, 'oldv' 스킴 타입은 확장 가능한 스킴이므로, OverlayConfigBox를 위해 명시된 버전(version) 값을 사용할 수 있고, 다른 값이 추가될 수도 있다. OverlayCofigBox가 SchemeInformationBox에 존재하는 경우, StereoVideoBox는 SchemeInformationBox에 존재하지 않을 수 있고, SchemeInformationBox는 직접적으로 또는 간접적으로 다른 박스를 포함할 수 있다. 즉, 오버레이가 프로젝션되지 않은 비디오인 경우 (scheme_type 필드 값이 'oldv'인 경우), SchemeInformationBox가 OverlayConfigBox를 포함할 수 있다.

[0307]

파일 #2와 같은 경우 및 VR 미디어 트랙이 비디오인 경우(프로젝션된 경우), VR 미디어 트랙은 ProjectedOmniVideoBox를 포함할 수 있고, ProjectedOmniVideoBox는 OverlayConfigBox를 포함할 수 있다. 여기서, OverlayConfigBox는 오버레이 미디어의 프로젝션 및 패킹 정보를 포함할 수 있다. 또는 오버레이 미디어의 프로젝션 및 패킹 정보를 포함하는 OverlayMediaPackingStruct()를 포함할 수 있다. 즉, 오버레이가 프로젝션된 비디오인 경우(scheme_type 필드 값이 'podv'인 경우), ProjectedOmniVideoBox가 OverlayConfigBox를 포함할 수 있다.

[0308]

상술한 OverlayConfigbox는 도 40에서 나타낸 속성을 가질 수 있으며, 예를 들어, 표 7과 같이 다음을 포함할 수 있다.

표 7

```
class OverlayConfigBox (type) extends FullBox ( 'ovly' , 0, 0) {
    OverlayMediaPackingStruct();
}
```

[0309]

[0310]

즉, OverlayConfigBox는 박스 타입(box type)이 ovly일 수 있고, 컨테이너(container)가 SchemeInformationBox 또는 VR 미디어 트랙이 포함된 경우 ProjectedOmniVideoBox일 수 있고, 필수 항목(mandatory)은 아닐 수 있고(No), 수량(quantity)은 0 또는 1일 수 있다. 또한, OverlayMediaPackingStruct()는 오버레이 미디어의 프로젝

선 및 패키징 정보를 포함할 수 있고, 표 3과 같을 수 있다.

- [0311] 이하에서는, 본 발명의 일 실시예에서는 메인 VR 미디어 트랙 및 오버레이 미디어 트랙의 그룹핑(grouping) 및/또는 링킹(linking) 방법을 제안할 수 있다.
- [0312] 도 35a 및 도 35b는 VR 미디어 트랙 및 오버레이 미디어 트랙의 그룹핑 및 링킹 예를 나타낸다.
- [0313] 도 35a를 참조하면, 일 실시예에서 파일 #1과 같이, 메인 VR 미디어 및 오버레이 미디어가 별도의 트랙으로 파일 상에 포함되어 있는 경우, track_group_type 필드 값이 'ovgr'인 TrackGroupTypeBox는 메인 VR 미디어 및 오버레이 미디어를 포함하는 트랙 그룹을 지칭할 수 있다. 이는 360 씬(scene)에서 오버레이 등과 함께 렌더링될 수 있는 트랙 그룹을 지칭할 수 있다. 즉, 동일한 track_group_id 필드 값을 가지는 트랙들은 360 씬에서 오버레이 등 함께 렌더링될 수 있음을 나타낼 수 있다. 따라서, 이를 통해 플레이어는 메인 미디어 및 오버레이 미디어를 편리하게 찾을 수 있다(retrieve).
- [0314] 도 35a를 참조하면, VR 미디어 트랙 #1 및 오버레이 미디어 트랙 #1 ~ #N는 오버레이 트랙 그룹(overlay track group)이 될 수 있으며, 이들은 동일한 track_group_id 필드 값을 가질 수 있고, 함께 렌더링될 수 있다.
- [0315] 여기서, 상술한 track_group_type 필드 값이 'ovgr'인 TrackGroupTypeBox는 OverlayVideoGroupBox를 포함할 수 있고, OverlayVideoGroupBox는 예를 들어, 표 8과 같이 다음을 포함할 수 있다.

표 8

```
aligned(8) class OverlayVideoGroupBox extends TrackGroupTypeBox( 'ovgr' ) {
    unsigned int(5) media_type;
    unsigned int(1) main_media_flag;
    unsigned int(1) overlay_media_flag;
    if ((media_type == 1) || (overlay_media_flag)){
        unsigned int(1) overlay_essential_flag;
    }
    bit(1) reserved = 0;
}
```

- [0316]
- [0317] 표 8에서, media_type 필드는 트랙 그룹 내에서 미디어의 타입을 지시할 수 있다. 예를 들어, media_type 필드 값이 0인 경우 메인 미디어임을 지시할 수 있고, 1인 경우 오버레이 미디어임을 지시할 수 있다. 또한, main_media_flag 필드는 메인 미디어 인지 여부를 나타내는 플래그를 의미할 수 있고, overlay_media_flag 필드는 오버레이 미디어 인지 여부를 나타내는 플래그를 의미할 수 있다. overlay_essential_flag 필드는 오버레이 미디어가 필수적으로 오버레이되어야 하는지 여부를 나타내는 플래그를 의미할 수 있다. 여기서, 필수적으로 오버레이되어야 하는 오버레이 미디어인 경우, 오버레이를 지원하지 않는 플레이어는 동일 그룹 내에 메인 미디어를 재생하지 않을 수 있다.
- [0318] 도 35b를 참조하면, 일 실시예에서 메인 VR 미디어 및 오버레이 미디어가 별도의 트랙으로 파일 상에 포함되어 있는 경우, 오버레이 미디어 트랙의 TrackReferenceTypeBox를 이용하여 오버레이 미디어가 오버레이될 메인 VR 미디어를 지칭할 수 있다. 이를 위하여 새로운 레퍼런스 타입(reference type)을 추가하여, 즉 reference_type 필드 값이 'ovmv'이고, track_IDs 필드에 하나 이상의 메인 VR 미디어 트랙 식별자 또는 트랙 그룹 식별자(메인 VR 미디어가 하나 이상의 트랙을 통하여 전달되는 경우)를 지칭함으로써, 오버레이 미디어가 오버레이되는 대상인 메인 미디어를 지칭할 수 있다. 다시 말해, 'ovmv' 및 track_IDs 필드를 통해 지칭되는 트랙들은 현재 오버레이 미디어가 오버레이되는 메인 미디어의 트랙들일 수 있다.
- [0319] 도 35b를 참조하면, 오버레이 미디어 트랙 #1 ~ #N은 'ovmv' track reference를 기반으로 오버레이될 VR 미디어 트랙 #1을 지시할 수 있다.
- [0320] TrackReferenceBox 및 TrackReferenceTypeBox는 예를 들어, 표 9와 같이 다음을 포함할 수 있다.

표 9

```
aligned(8) class TrackReferenceBox extends Box( 'tref' ) {
}
aligned(8) class TrackReferenceTypeBox (unsigned int(32) reference_type)
extends Box {
    unsigned int(32) track_IDs[];
}
```

[0321]

[0322]

표 9에서, track_ID 필드는 프레젠테이션(presentation) 내에서 컨테이너 트랙(containing track)으로부터 다른 트랙(another track)으로의 레퍼런스를 제공하는 정수일 수 있으며, track_IDs 필드는 재사용될 수 없으며, 0과 같은 값을 가질 수 없다. 또한, reference_type 필드는 상술한 바와 같이 지칭 또는 지시할 수 있고, 다음의 값들(following values) 중 하나로 설정될 수 있다.

[0323]

일 실시예에서 파일 #2와 같이, 메인 VR 미디어 및 오버레이 미디어가 동일한 트랙 내에 포함되어 있는 경우, 해당 트랙은 grouping_type 필드 값이 'ovmv'인 SampleToGroupBox를 포함할 수 있다. SampleToGroupBox는 해당 트랙에 포함된 샘플들 중 함께 렌더링(오버레이 포함)되어야 하는 샘플들을 지칭할 수 있다. 해당 트랙에서 grouping_type 필드 값이 'ovmv'인 SampleToGroupBox가 존재하는 경우, grouping_type 필드 값이 'ovmv'인 SampleGroupDescriptionBox가 존재할 수 있다. 이에는 함께 렌더링(오버레이)되는 샘플들에 공통적으로 적용되는 정보들이 포함될 수 있다. 또는 OverlayEntry가 포함될 수 있다. OverlayEntry는 grouping_type 필드 값이 'ovmv'인 샘플 그룹 엔트리(sample group entry)를 의미할 수 있으며, OverlayEntry는 예를 들어, 표 10과 같이 다음을 포함할 수 있다.

표 10

```
class OverlayEntry() extends SampleGroupDescriptionEntry(' ovmv' ) {
    unsigned int(32) overlay_essential_flag;
}
```

[0324]

[0325]

표 10에서, overlay_essential_flag 필드는 오버레이 미디어가 필수적으로 오버레이되어야 하는지 여부를 나타내는 플래그를 의미할 수 있다. 여기서, 필수적으로 오버레이되어야 하는 오버레이 미디어인 경우, 오버레이를 지원하지 않는 플레이어는 동일 그룹 내에 메인 미디어를 재생하지 않을 수 있다.

[0326]

일 실시예에서는 하나의 샘플에 VR 미디어 및 오버레이 미디어가 포함될 수도 있다. 이러한 경우, 하나의 샘플 내에서 서브 샘플(sub-sample)들로 나누어질 수 있고, 각 서브 샘플이 VR 미디어 또는 오버레이 미디어를 포함할 수 있다. 또한, 서브 샘플 관련 정보를 포함하는 박스(box)에 서브 샘플이 오버레이 미디어를 포함하고 있는지 메인 VR 미디어를 포함하고 있는지에 대한 지시자(indicator) 및 오버레이 미디어가 필수적으로 오버레이 되어야 하는지 여부에 대한 플래그 등이 포함될 수 있다.

[0327]

일 실시예에서는 메인 VR 미디어 간에, 오버레이 미디어 간에 스위칭을 위한 대체 미디어 그룹핑 방법을 제안할 수 있다. EntityToGroupBox 내의 grouping_type 필드(필드 값이'altr'인 grouping_type 필드)를 활용하여 대체 가능한 메인 VR 미디어를 명시할 수 있고, 대체 가능한 오버레이 미디어들을 명시할 수 있다. 이는 씬 그래프(scene graph)의 스위치 노드(switch node)와 유사한 개념일 수 있다. 즉, 노드 상에 여러 개의 노드들을 가지고 있고, 그 노드들 중 한 개의 노드만 액티브(active)/비저블(visible) 상태가 될 수 있다. 스위치 노드는 현재 액티브한 노드의 인덱스를 가지고 있고, 인덱스를 바꿔서 다른 노드를 액티브하게 변경할 수 있다. 일 실시예에서 필드 값이 'altr'인 grouping_type 필드를 통해 그룹핑된 미디어는 오버레이와 인터랙션 시, 메인 VR 미디어를 대체 VR 미디어 또는 대체 메인 미디어로 스위칭 한다거나, 오버레이 미디어를 대체 오버레이 미디어로 스위칭할 때 이용될 수 있다.

[0328]

즉, 일 실시예는 대체 가능한 미디어들의 그룹핑을 통해 메인 VR 미디어 간에 스위칭할 수 있고, 오버레이 미디어 간에 스위칭할 수 있으며, 이는 관련 오버레이와 인터랙션을 통해 수행될 수 있다. 또한, 그룹핑된 미디어들

은 EntityToGroupBox 내의 grouping_type 필드를 통해 명시될 수 있다.

[0329] 도 36은 파일 #1인 경우 오버레이 메타데이터 트랙을 나타낸 예이다.

[0330] 일 실시예에서 오버레이 메타데이터 트랙은 오버레이 렌더링을 위한 오버레이 위치, 사이즈 및 속성에 관한 정보(opacity 및 interaction 등)를 포함할 수 있다. 오버레이의 렌더링 메타데이터는 시간에 따라 변할 수 있다. 따라서, 타임드(timed) 메타데이터로 저장될 수 있다. 즉, 오버레이가 시간에 따라 사이즈 또는 위치 등이 변할 수 있으며, 이와 같은 시간에 따라 변할 수 있는 메타데이터는 오버레이의 렌더링 메타데이터로 지칭할 수 있고, 타임드 메타데이터로 저장될 수 있다. 즉, 시간에 따라 변하는 메타데이터는 샘플에 저장할 수 있으나, 시간에 따라 변하지 않는 스태틱(static)한 메타데이터는 샘플 엔트리에 저장될 수 있다.

[0331] 도 37a 내지 도 37c는 오버레이를 배치할 위치를 나타낸 예이다.

[0332] 일 실시예에서 오버레이 렌더링 위치는 오버레이를 배치할 위치에 따라 3가지의 케이스로 구분될 수 있다.

[0333] 도 37a를 참조하면, 제1 케이스(Case 1)는 오버레이가 사용자의 현재 뷰포트에 위치하는 경우일 수 있다. 이 경우, 뷰포트 상에 그려질 위치 및 크기 정보를 디스플레이 크기 대비 퍼센트로 위치가 명시될 수 있다. 또한, 오버레이들이 겹쳐지는 경우를 고려하기 위해 그려지는 순서가 명시될 수 있다. 여기서, 위치 및 크기 정보는 x 축 포인트 위치 정보(또는 좌측 포인트의 위치 정보), y 축 포인트 위치 정보(또는 상측 포인트의 위치 정보), 너비(width) 정보 및 높이(height) 정보를 포함할 수 있다.

[0334] 도 37b를 참조하면, 제2 케이스(Case 2)는 오버레이가 스피어(sphere) 상에 위치하는 경우일 수 있다. 이 경우, 고도각(elevation) 정보로 중심 위치가 명시될 수 있고, 방위각(azimuth) 및 고도각 범위를 지정하여 오버레이의 크기가 명시될 수 있다. 다만, 오버레이의 중심점으로부터 스피어의 원점으로의 벡터를 축으로 한 회전만 지원될 수 있다. 또는 프로젝션을 고려한 리전별 패키지의 프로젝션 내의 위치 정보 또는 위치 표현으로 정의될 수 있다. 여기서, 오버레이는 스피어 상에 위치하지만, 플레이어 측에서는 오버레이를 곡면으로 처리할 수 있고, 사격형 평면으로 처리할 수도 있다.

[0335] 도 37c를 참조하면, 제3 케이스(Case 3)는 오버레이가 스피어 내부에 존재하는 경우일 수 있다. 이 경우, 인접 평면(near plane) 및 스피어 내부에 존재할 수 있으며, 평면은 사각형으로 가정할 수 있고, 그 크기는 y 축 및 z 축 기반의 평면을 포인트로 너비 정보 및 높이 정보를 통해 명시될 수 있다. 또한, 평면의 크기가 결정된 후, 스피어 좌표계 상에서 x 축 기준 위치 정보, y 축 기준 위치 정보 및 z 축 기준 위치 정보를 기반으로 이동될 수 있다. 또는 스피어 좌표계 상에서의 (x, y, z) 좌표로 이동될 수 있다. 여기서, 오버레이의 중심점을 중심으로 하고, 스피어의 각 축과 평행인 오버레이 좌표계를 포인트로 각 축에 대한 회전이 지원될 수 있다.

[0336] 일 실시예에서, 오버레이 미디어가 오버레이되는 위치 관련 정보는 오버레이 관련 메타데이터에 포함될 수 있으며, OverlayPosStruct()에 포함될 수 있다. OverlayPosStruct()는 예를 들어, 표 11과 같이 다음을 포함할 수 있다.

표 11

```
aligned(8) class OverlayPosStruct() {
    unsigned int(8) region_type;
    if(region_type == 0) {
        ViewportOverlayRegion();
    } else if (region_type == 1) {
        SphereOverlayRegion();
    } else if (region_type == 2) {
        3DOverlayRegion();
    }
}
```

[0337]

[0338] 표 11에서, region_type 필드는 오버레이가 배치되는 위치에 관한 정보를 지시할 수 있다. 여기서, region_type 필드 값이 0인 경우, 오버레이가 사용자 뷰포트에 위치함을 지시할 수 있다. 이는 상술한 제1 케이스와 같은 경우를 의미할 수 있으며, ViewportOverlayRegion()이 호출될 수 있다. region_type 필드 값이 1인 경우, 오버레이

이가 스피어 상에 위치함을 지시할 수 있다. 이는 상술한 제2 케이스와 같은 경우를 의미할 수 있으며, SphereOverlayRegion()이 호출될 수 있다. region_type 필드 값이 2인 경우 오버레이가 3차원 공간 상에 위치함을 지시할 수 있다. 이는 상술한 제3 케이스와 같은 경우를 의미할 수 있으며, 3DOverlayRegion()이 호출될 수 있다.

[0339] 도 38은 오버레이가 뷰포트 상에 배치되는 경우에 대한 예이다.

[0340] 도 38을 참조하면, 오버레이는 사용자의 뷰포트 상에 위치할 수 있다. 이를 위해 사용자 뷰포트에 배치되는 오버레이의 위치 관련 정보가 시그널링될 수 있으며, 이는 상술한 ViewportOverlayRegion()에 포함될 수 있다. ViewportOverlayRegion()는 예를 들어, 표 12와 같이 다음을 포함할 수 있다.

표 12

```
aligned(8) class ViewportOverlayRegion() {
    unsigned int(16) rect_left_percent;
    unsigned int(16) rect_top_percent;
    unsigned int(16) rect_width_percent;
    unsigned int(16) rect_height_percent;
    unsigned int(16) order;

    unsigned int(1) stereoscopic_flag;
    bit(7) reserved=0;
    if (stereoscopic_flag == 1) {
        unsigned int(1) relative_disparity_flag;
        if (relative_disparity_flag == 1) {
            signed int(16) disparity_in_percent;
        } else {
            signed int(16) disparity_in_pixels;
        }
    }
}
```

[0341]

[0342] 표 12에서, rect_left_percent 필드, rect_top_percent 필드, rect_width_percent 필드 및 rect_height_percent 필드는 사각 평면인 오버레이의 위치 및 크기 정보를 지시할 수 있다. 즉, 각각은 오버레이의 좌측 포인트의 위치 정보, 상측 포인트의 위치 정보, 너비 정보 및 높이 정보를 지시할 수 있으며, 디스플레이 크기에 따라 달라질 수 있으므로 퍼센트로 지시될 수 있다.

[0343] 또한, order 필드는 다른 오버레이들과 겹쳐지는 경우 그려질 순서를 지시할 수 있다. 또는 오버레이 순서를 지시할 수 있다. 이를 통해 수신기에서는 렌더링시 순서를 조정하거나 배치 값을 조정할 수 있다.

[0344] 또한, stereoscopic_flag 필드는 오버레이가 스테레오를 지원하는지 여부에 대한 플래그를 의미할 수 있으며, relative_disparity_flag 필드는 스테레오 시 상대적 디스패리티(disparity) 값을 가지는지 여부에 대한 플래그를 의미할 수 있고, disparity_in_percent 필드 및 disparity_in_pixels 필드는 각각 상대적 디스패리티 값 및 픽셀 단위의 디스패리티 값을 지시할 수 있다.

[0345] 도 39는 오버레이가 스피어 상에 배치되는 경우에 대한 예이다.

[0346] 도 39를 참조하면, 오버레이는 스피어 상에 위치할 수 있다. 이를 위해 스피어 상에 배치되는 오버레이의 위치 관련 정보가 시그널링될 수 있으며, 이는 상술한 SphereOverlayRegion()에 포함될 수 있다. SphereOverlayRegion()는 예를 들어, 표 13과 같이 다음을 포함할 수 있다.

표 13

```
aligned(8) class SphereOverlayRegion() {
    unsigned int(1) proj_shape=0;
    if (proj_shape == 1) {
        unsigned int(32) proj_reg_width_percent;
        unsigned int(32) proj_reg_height_percent;
        unsigned int(32) proj_reg_top_percent;
        unsigned int(32) proj_reg_left_percent;

    } else if (proj_shape == 2) {
        unsigned int(8) num_rings;
        unsigned int(8) num_sectors;
        for (i = 0; i < num_rings; i++) {
            for (j =0; j < num_sectors; j++) {
                unsigned int(16) proj_points_x[i][j];
                unsigned int(16) proj_points_y[i][j];
            }
        }
        unsigned int(3) transform_type[i];
        bit(5) reserved = 0;
        for (i = 0; i < num_rings; i++) {
            for (j =0; j < num_sectors; j++) {
                unsigned int(16) proj_points_x[i][j];
                unsigned int(16) proj_points_y[i][j];
            }
        }
    } else if (proj_shape ==0) {
        unsigned int(8) shape_type;
        signed int(32) centre_azimuth;
        signed int(32) centre_elevation;
        unsigned int(32) azimuth_range;
        unsigned int(32) elevation_range;
        signed int(32) centre_tilt;
    }
    unsigned int(1) interpolate;
    unsigned int(16) depth;
}
```

[0347]

[0348]

표 13에서, proj_shape 필드는 프로젝션된 형태를 지시할 수 있으며, proj_shape 필드 값이 0인 경우 프로젝션되지 않음(none)을 지시할 수 있고, 1인 경우 사각형(rectangle) 형태로 프로젝션됨을 지시할 수 있고, 2인 경우 다각형(polygon) 형태로 프로젝션됨을 지시할 수 있다.

[0349]

프로젝션된 형태가 사각형인 경우(proj_shape == 1), proj_reg_top_percent 필드, proj_reg_left_percent 필드, proj_reg_width_percent 필드 및 proj_reg_height_percent 필드는 프로젝션된 픽처에서의 오버레이의 위치 정보를 지시할 수 있다. 즉, 각각은 프로젝션된 픽처에서 오버레이의 상측 포인트 위치 정보, 좌측 포인트 위치 정보, 너비 정보 및 높이 정보를 퍼센트로 지시할 수 있다.

[0350]

프로젝션된 형태가 다각형인 경우(proj_shape == 2), num_rings 필드 및 num_sectors 필드는 프로젝션된 픽처에서의 오버레이의 위치 정보를 지시할 수 있다. 즉, 각각은 프로젝션된 픽처에서 수평으로 영역을 분할하는 개수 및 수직으로 영역을 분할하는 개수를 지시할 수 있다. 또한, proj_points_x 필드 및 proj_points_y 필드는

각 분할 점들의 프로젝션된 픽처에서의 위치 정보를 지시할 수 있다. 즉, 각각은 프로젝션된 픽처에서의 x축 기준의 위치 값 및 y축 기준의 위치 값을 지시할 수 있다. 또한, packed_points_x 필드 및 packed_points_y 필드는 각 분할 점들의 팩드 픽처에서의 위치 정보를 지시할 수 있다. 즉, 각각은 팩드 픽처에서의 x축 기준의 위치 값 및 y축 기준의 위치 값을 지시할 수 있다.

[0351] 프로젝션되지 않은 경우(proj_shape == 0), shape_type 필드는 스피어 상의 위치 표현 타입을 지시할 수 있다. 여기서, shape_type 필드 값이 0인 경우 4개의 대원(great circle)로 구성될 수 있고, 1인 경우 2개의 방위 원(azimuth circle) 및 2개의 고도 원(elevation circle)로 구성될 수 있다. 여기서, centre_azimuth 필드 및 centre_elevation 필드는 오버레이 중심 위치의 위치 정보를 지시할 수 있다. 즉, 각각은 오버레이 중심 위치의 방위 값 및 고도 값을 지시할 수 있다. 또한, azimuth_range 필드 및 elevation_range 필드는 오버레이의 크기 정보를 지시할 수 있다. 즉, 각각은 오버레이의 방위 범위 및 고도 범위를 지시할 수 있다. 또한, centre_tilt 필드는 오버레이의 중심점으로부터 스피어의 원점으로부터의 벡터를 축으로 한 회전 값을 지시할 수 있다.

[0352] 또한, interpolate 필드는 변경된 값들 사이에 값을 채워 부드럽게 변경하기 위한 플래그를 의미할 수 있고, depth 필드는 오버레이들이 겹쳐질 경우, 우선 적으로 나타낼 오버레이의 순서를 위해 원점으로부터 오버레이 중심까지의 거리 값을 지시할 수 있다.

[0353] 도 40은 오버레이가 스피어 내부의 3차원 공간 상에 배치되는 경우에 대한 예이다.

[0354] 도 40을 참조하면, 오버레이는 스피어 내부의 3차원 공간 상에 위치할 수 있다. 이를 위해 스피어 내부의 3차원 공간 상에 배치되는 오버레이의 위치 관련 정보가 시그널링될 수 있으며, 이는 상술한 3DOverlayRegion()에 포함될 수 있다. 3DOverlayRegion()는 예를 들어, 표 14와 같이 다음을 포함할 수 있다.

표 14

```
aligned(8) class Overlay3DPositionStruct() {
    signed int(32) overlay_pos_x;
    signed int(32) overlay_pos_y;
    signed int(32) overlay_pos_z;
}

aligned(8) class OverlayRotationStruct() {
    signed int(32) overlay_rot_yaw;
    signed int(32) overlay_rot_pitch;
    signed int(32) overlay_rot_roll;
}

aligned(8) class 3DOverlayRegion() {
    unsigned int(32) width;
    unsigned int(32) height;
    bit(7) reserved=0;
    unsigned int(1) interpolate;
    Overlay3DPositionStruct();
    OverlayRotationStruct();
}
```

[0355]

[0356] 표 14에서, width 필드 및 height 필드는 오버레이 미디어가 사각형인 것으로 가정하고, y축 및 z축 기반의 평면을 기준으로 너비 정보 및 높이 정보를 지시할 수 있다. 여기서, 사각형이 오버레이 미디어 또는 오버레이 평면은 크기가 지시 또는 결정될 수 있다. 또한, interpolate 필드는 변경된 값들 사이에 값을 채워 부드럽게 변경하기 위한 플래그를 의미할 수 있으며, 3DOverlayRegion()는 Overlay3DPositionStruct() 및 OverlayRotationStruct()를 포함할 수 있다.

- [0357] Overlay3DPositionStruct()는 스피어 좌표계 상에서 오버레이 미디어의 위치 정보를 포함할 수 있다. 여기서, overlay_pos_x 필드, overlay_pos_y 필드 및 overlay_pos_z 필드는 각각 스피어 좌표계 상에서 오버레이 미디어의 x축 기준의 위치 값, y축 기준의 위치 값 및 z축 기준의 위치 값을 지시할 수 있으며, 오버레이 미디어는 스피어 좌표계 상에서 x축 기준 위치 값, y축 기준 위치 값 및 z축 기준 위치 값으로 이동될 수 있다. 또는 스피어 좌표계 상에서의 (x, y, z) 좌표로 이동될 수 있다.
- [0358] OverlayRotationStruct()는 오버레이 중심점을 중심으로 하고, 스피어의 각 축과 평행인 오버레이 좌표계를 기준으로 각 축에 대한 회전 정보를 지시할 수 있다. 여기서, overlay_rot_yaw 필드, overlay_rot_pitch 필드 및 overlay_rot_roll 필드는 각각 요(yaw) 축에 대한 회전 정보, 피치(pitch) 축에 대한 회전 정보 및 롤(roll) 축에 대한 회전 정보를 지시할 수 있다. 즉, 일 실시예는 오버레이 중심점을 중심으로 하여 스피어의 각 축과 평행인 오버레이 좌표계를 기준으로 각 축에 대한 회전을 지원할 수 있다.
- [0359] 도 41은 오버레이가 스피어 내부의 3차원 공간 상에 존재하는 경우, 오버레이의 위치/크기/회전을 나타낸다.
- [0360] 도 41을 참조하면, 좌측 스피어에서 너비(width), 높이(height) 및 (x, y, z) 좌표에 대한 정보는 표 14의 width 필드, height 필드, overlay_pos_x 필드, overlay_pos_y 필드 및 overlay_pos_z 필드에 의해 지시될 수 있다.
- [0361] 또한, 우측 스피어에서 요(yaw) 축 회전, 피치(pitch) 축 회전 및 롤(roll) 축 회전에 대한 정보는 표 14의 overlay_rot_yaw 필드, overlay_rot_pitch 필드 및 overlay_rot_roll 필드에 의해 지시될 수 있다.
- [0362] 도 42는 오버레이 렌더링 속성의 예를 나타낸다.
- [0363] 일 실시예에서 오버레이 메타데이터는 오버레이 렌더링 속성 정보를 포함할 수 있다. 이는 오버레이를 렌더링할 때 적용되는 오버레이 평면(surface)의 투명도, VR 미디어 상에 오버레이를 블렌딩할 때 수행하는 렌더링 옵션 및 포커스(focus) 효과 등에 대한 정보를 포함할 수 있으며, 메타데이터에 포함되어 시그널링될 수 있다. 여기서, 메타데이터는 오버레이 메타데이터, 오버레이 관련 메타데이터 또는 오버레이 렌더링 관련 메타데이터라고 지칭할 수도 있다. 오버레이 렌더링 속성 정보는 오버레이 미디어가 디스플레이/렌더링 시 적용될 수 있는 렌더링 속성 정보라고 지칭할 수 있으며, OverlayRenderStruct()에 포함될 수 있고, OverlayRenderStruct()는 예를 들어, 표 15와 같이 다음을 포함할 수 있다.

표 15

```

aligned(8) class OverlayRenderStruct() {
    unsigned int(1) opacity_info_flag;
    unsigned int(1) alpha_composition_flag;
    unsigned int(1) blending_flag;
    unsigned int(1) focus_flag;
    unsigned int(4) reserve =0;
    if(opacity_info_flag == 1) {
        unsigned int(8) opacity;
    }
    if(alpha_compositino_flag == 1) {
        unsigned int(8) composition_type=1;
    }
    if (blending_flag == 1) {
        unsigned int(8) blending_mode;
    }
    if (focus_flag == 1) {
        unsigned int(8) focus;
    }
}
    
```

[0364]

[0365] 표 15에서, opacity_info_flag 필드는 오버레이 평면의 전체 투명도 명시 여부를 나타내는 플래그를 의미할 수 있고, opacity 필드는 투명 정도에 대한 정보 또는 투명 정도 값을 지시할 수 있다.

[0366] 또한, alpha_composition_flag 필드는 오버레이 합성 시 오버레이 미디어가 알파 채널을 가지고 있고, 그 알파 값을 합성할 때에 알파 컴포지션 적용 여부를 나타내는 플래그를 의미할 수 있고, composition_type 필드는 알파 컴포지션 타입을 지시할 수 있다. 여기서, composition_type 필드 값이 1인 경우, source_over를, 2인 경우 source_atop을, 3인 경우 source_in을, 4인 경우 source_out을, 5인 경우 dest_atop을, 6인 경우 dest_over를, 7인 경우 dest_in을, 8인 경우 dest_out을, 9인 경우 clear를, 10인 경우 xor을 지시할 수 있고, 디폴트는 composition_type 필드 값이 1인 source_over일 수 있으며, 각 타입 별로 적용되는 공식(formula)는 예를 들어, 표 16과 같을 수 있다.

표 16

Composition Type	Formula
source_over	$a_s \cdot (1 - a_d) \cdot s + a_d \cdot (1 - a_s) \cdot d + a_s \cdot a_d \cdot s$
source_atop	$a_d \cdot (1 - a_s) \cdot d + a_s \cdot a_d \cdot s$
source_in	$a_s \cdot a_d \cdot s$
source_out	$a_s \cdot (1 - a_d) \cdot s$
dest_atop	$a_s \cdot (1 - a_d) \cdot s + a_s \cdot a_d \cdot d$
dest_over	$a_s \cdot (1 - a_d) \cdot s + a_d \cdot (1 - a_s) \cdot d + a_s \cdot a_d \cdot d$
dest_in	$a_s \cdot a_d \cdot d$
dest_out	$a_d \cdot (1 - a_s) \cdot d$
clear	0
xor	$a_s \cdot (1 - a_d) \cdot s + a_d \cdot (1 - a_s) \cdot d$

[0367]

[0368] 표 16에서, a_s 는 소스(source) 픽셀의 알파 값을 의미할 수 있고, a_d 는 대상(destination) 픽셀의 알파 값을 의미할 수 있다. s 는 소스 픽셀의 색상(RGBA) 값을 의미할 수 있고, d 는 대상 픽셀의 색상(RGBA) 값을 의미할 수 있다.

[0369] 또한, blending_flag 필드는 오버레이 합성 시 적용할 블렌딩 명시 여부를 나타내는 플래그를 의미할 수 있고, 여기서, blending_mode 필드는 블렌딩 모드를 지시할 수 있다. 블렌딩은 알파 컴포지션보다 복잡한 오퍼레이션(operation)으로 픽셀의 색상을 블렌딩하는 것까지 포함될 수 있다.

[0370] 여기서, blending_mode 필드 값이 1인 경우 normal을, 2인 경우 multiply을, 3인 경우 screen을, 4인 경우 overlay을, 5인 경우 darken을, 6인 경우 lighten를, 7인 경우 color dodge을, 8인 경우 color-burn을, 9인 경우 hard-light를, 10인 경우 soft-light를, 11인 경우 difference를, 12인 경우 exclusion을 지시할 수 있고, 각 모드 별로 적용되는 공식(formula)는 예를 들어, 표 17과 같을 수 있다.

표 17

Blending Mode	Formula
normal	$B(s,d)=s$
multiply	$B(s,d)=s \cdot d$
screen	$B(s,d)=d+s-(d \cdot s)$
over lay	$B(s,d)=\text{hard-light}(d,s)$
darken	$B(s,d)=\min(s,d)$
lighten	$B(s,d)=\max(s,d)$
color dodge	$B(s,d)=0$ (if $d=0$), 1 (if $s=1$), $\min(1, d/(1-s))$ (otherwise)
color-burn	$B(s,d)=1$ (if $d=1$), 0 (if $s=0$), $1-\min(1, (1-d)/s)$ (otherwise)
hard-light	$B(s,d)=\text{multiply}(2 \cdot s, d)$ (if $s \leq 0.5$), $\text{screen}(2 \cdot s-1, d)$ (otherwise)
soft-light	$B(s,d)=d-(1-2 \cdot s) \cdot d \cdot (1-d)$ (if $s \leq 0.5$), $((16 \cdot d-12) \cdot d+4) \cdot d$ (otherwise)
difference	$B(s,d)= d-s $
exclusion	$B(s,d)=d+s-2 \cdot d \cdot s$

[0371]

[0372]

[0373]

[0374]

[0375]

[0376]

표 17에서, s는 소스 픽셀의 RGBA 값을 의미할 수 있고, d는 대상 픽셀의 RGBA 값을 의미할 수 있다.

focus_flag 필드는 오버레이 포커스 여부를 나타내는 플래그를 의미할 수 있고, focus 필드는 포커스 정도에 대한 정보 또는 포커스 정도 값을 지시할 수 있다. 여기서, 포커스 정도 값은 0~1.0 범위를 가질 수 있다. 오버레이에 포커스가 명시 또는 지시되면, 수신기에서 렌더링되고 있는 다른 오버레이와 VR 미디어에 블러(blur)가 적용될 수 있다.

도 43는 오버레이 미셀레이니어스(miscellaneous)의 예를 나타낸다.

일 실시예에서 오버레이 메타데이터는 오버레이 미셀레이니어스 정보를 포함할 수 있다. 여기서, 오버레이 미셀레이니어스 정보는 오버레이 렌더링 기타 정보라 지칭할 수도 있다. 이는 오버레이 테두리 지원에 대한 정보, 다양한 오버레이 모양(shape) 지원에 대한 정보, 빌보드(billboard) 지원 여부에 대한 정보 및 타겟과 오버레이의 위치가 다름에 따라 오버레이의 위치가 가리키는 특정 포인트를 지시하는 정보를 포함할 수 있다. 여기서, 빌보드는 사용자의 뷰잉 오리엔테이션에 맞추어 오버레이의 회전 값이 변경되는 방법을 의미할 수 있다.

상술한 오버레이 메타데이터는 시그널링될 수 있으며, 오버레이 메타데이터는 메타데이터, 오버레이 관련 메타데이터, 오버레이 렌더링 기타 메타데이터, 오버레이 렌더링 관련 메타데이터 또는 오버레이 미셀레이니어스 관련 메타데이터라고 지칭할 수도 있다. 오버레이 리셀레이니어스 정보는 오버레이 관련 추가로 설정 가능한 기타 렌더링 정보라고 지칭할 수 있으며, OverlayMiscStruct()에 포함될 수 있고, OverlayMiscStruct()는 예를 들어, 표 18과 같이 다음을 포함할 수 있다.

표 18

```

aligned(8) class OverlayMiscStruct() {
    unsigned int(1) frame_flag;
    unsigned int(1) shape_flag;
    unsigned int(1) billboard_flag;
    unsigned int(1) target_flag;
    if (frame_flag == 1) {
        unsigned int(8) frame_border_width;
        unsigned int(32) frame_color[4];
    }
    if (shape_flag == 1) {
        unsigned int(8) shape_type;
        if (shape_type == 1) { // curve
            unsigned int(32) h_curvature;
            unsigned int(32) v_curvature;
        } else if (shape_type == 3) {
            unsigned int(8) num_vertices;
            unsigned int(32) scale;
            for (int i=0; i< num_vertices; i++) {
                unsigned int(32) xyz[3];
                unsigned int(32) st[2];
            }
        }
    }
    if (target_flag == 1) {
        signed int(32) target_azimuth;
        signed int(32) target_elevation;
    }
}

```

[0377]

[0378]

표 18에서, frame_flag 필드는 오버레이 평면의 테두리를 그리는지 여부에 대한 플래그를 의미할 수 있고, frame_border_width 필드는 테두리를 그리는 경우 테두리 두께 사이즈를 지시할 수 있으며, frame_color 필드는 테두리에 대한 투명도를 포함하는 RGBA 색상 값을 지시할 수 있다. shape_flag 필드는 오버레이 평면의 모양을 사각형 이외의 다른 모양으로 지정하려는지 여부에 대한 플래그를 의미할 수 있다. 여기서, shape_flag 필드 값이 1인 경우 커브(curve) 타입을, 2인 경우 원형(circle) 타입을, 3인 경우 사용자 정의에 따른 타입을 지시할 수 있으며, 다른 값은 예비적으로 남겨둘 수 있고, 다른 설정에 따라 정의될 수 있다.

[0379]

여기서, shape_flag 필드 값이 1인 경우(shape_flag == 1), h_curvature 필드 및 v_curvature 필드는 커브 정도를 지시할 수 있다. 즉, 각각은 수평 커버처(curvature) 값 및 수직 커버처 값을 지시할 수 있다.

[0380]

또한, shape_flag 필드 값이 3인 경우(shape_flag ==3), num_vertices 필드, scale 필드, xyz 필드 및 st 필드는 각각 정점의 개수, 스케일 정보, 각 정점의 (x, y, z) 좌표 정보 또는 위치 정보 및 텍스처 좌표 정보를 지시할 수 있다.

[0381]

billboard_flag 필드는 오버레이 평면에 빌보드 적용 여부에 대한 플래그를 의미할 수 있고, target_flag 필드는 오버레이 타겟 유무 여부에 대한 플래그를 의미할 수 있다. 여기서, target_flag 필드에 의해 타겟이 존재하는 것을 지시되는 경우, target_azimuth 필드 및 target_elevation 필드는 타겟의 위치 정보를 지시할 수 있다. 즉, 각각은 타겟의 고도 정보(또는 고도 값) 및 방위 정보(또는 방위 값)를 지시할 수 있다.

[0382]

도 44는 뷰포트 내에 이동 가능한 공간을 나타낸 예이다.

- [0383] 일 실시예에서 VR 미디어는 몰입감을 위해 인터랙션(interaction)을 제공할 수 있다. 또는 VR 미디어의 오버레이 인터랙션을 제공할 수 있다. 기본적인 인터랙션은 HMD(head mounted display)를 착용하고, 사용자의 위치 및 바라보는 방향이 변경될 때 그에 따라서 변화가 적용되어 화면을 구성할 수 있다. 인터랙션을 더 추가하기 위해서는 VR 미디어 상에서 오버레이와 인터랙션을 할 수도 있다. 이러한 경우 인터랙션이 가능한 오버레이인지 여부 및 인터랙션이 가능한 범위가 지시될 수 있다.
- [0384] 여기서, 인터랙션이 가능한 범위는 뷰포트(viewport) 영역 내의 움직임이 가능한 공간 및 각 오버레이의 움직임이 가능한 공간으로 나누어질 수 있고, 두 가지 공간이 모두 정의될 수 있다.
- [0385] 또한, 인터랙션이 가능한 오버레이에 대한 각 오버레이의 위치/깊이/회전/스케일 정보를 추가로 제어할 수 있다. 오버레이는 항상 뷰포트 영역 내에 존재할 필요는 없다. 다만, 사용자는 뷰포트 상에 존재하는 오버레이와 인터랙션을 수행할 수 있다. 따라서, 오버레이 미디어 인터랙션을 위한 전체 공간은 사용자 뷰포트 영역일 수 있다. 사용자가 현재 뷰포트에 보이는 오버레이 중 인터랙션이 가능한 오버레이를 선택한 경우, 사용자는 해당 오버레이의 위치, 오리엔테이션 및 스케일을 변경할 수 있다. 오버레이를 감싸는 바운드 박스(Bound Box)는 변경에 맞추어 업데이트될 수 있으며, 업데이트된 바운드 박스는 사용자 뷰포트 영역 안에 존재할 수 있다.
- [0386] 도 44를 참조하면, 뷰포트 영역 내에 움직임이 가능한 공간을 지시하기 위해서는 수평 FOV (field of View), 고도(azimuth) 정보, 수직 FOV, 방위(elevation) 정보 및 니어 플레인(near plane)의 위치 값이 이용될 수 있다. 여기서, 수평 FOV, 고도 정보, 수직 FOV 및 방위 정보는 HMD에 따라 적용될 수 있으며, 플레이어에서 지정 가능할 수 있다. 또한, 니어 플레인의 위치 값도 플레이어에서 지정 가능할 수 있다.
- [0387] 일 실시예는 수평 FOV, 수직 FOV, 니어 플레인의 위치 값, 파 플레인(far plane)의 위치 값으로 뷰잉 프러스텀(viewing frustum)을 생성할 수 있다. 여기서, 파 플레인의 위치 값은 스피어가 단위 스피어(unit sphere)이므로, 1의 값을 가질 수 있다.
- [0388] 도 45는 VFC 알고리즘을 설명하기 위한 예이다.
- [0389] 뷰잉 프러스텀 내에 존재하는지 여부를 체크하는 다양한 뷰잉 프러스텀 컬링(VFC: Viewing Frustum Culling) 알고리즘이 존재할 수 있고, VFC 알고리즘을 사용하여 오버레이의 바운드 박스가 컬링되는 영역이 존재하는지 판단할 수 있다.
- [0390] 여기서, 컬링되는 영역이 있는 경우, 해당 방향으로 움직이지 못하게 제어할 수 있고, 또는 추가로 보상되는 다른 영역이 존재 가능한 경우 해당 위치로 업데이트할 수 있다. 상술한 동작은 수신기에서 처리할 수 있다.
- [0391] 기본적인 VFC는 AABBvsFrustum을 이용할 수 있다. 다만, VFC는 다양한 다른 방법을 이용할 수 있으므로, 이에 한정되는 것은 아니다. 도 51을 참조하면, AABBvsFrustum은 바운드 박스와 인터랙션되는 면이 존재하는지 확인하여 외부(outside) 또는 인터섹트(intersect)가 아닌 경우, 뷰포트 영역 내에 안전하게 존재하는 것으로 결정할 수 있다. AABBvsFrustum은 예를 들어, 표 19와 같이 다음을 포함할 수 있다.

표 19

```

int AABBvsFrustum(AABB *b, FRUSTUM *f){
    float m, n;
    int i, result = INSIDE;
    for (i = 0; i < 6; i++){
        PLANE *p = f->plane + i;
        m = (b->mx * p->a) + (b->my * p->b) + (b->mz * p->c) + p->d;
        n = (b->dx * fabs(p->a)) + (b->dy * fabs(p->b)) + (b->dz * fabs(p->c));
        if (m + n < 0) return OUTSIDE;
        if (m - n < 0) result = INTERSECT;
    }
    return result;
}
    
```

- [0392]
- [0393] 일 실시예에서는 인터랙션을 통해 특정 오버레이들이 현재 뷰포트 영역 내에서 자유롭게 움직일 수 있으나, 각

오버레이 별로 움직일 수 있는 영역을 추가로 지정할 수 있다. 예를 들어, 특정 오버레이는 위치를 고정하고, 특정 방향으로만 회전할 수 있도록 움직임을 한정시킬 수 있다.

[0394] 일 실시예에서는 각 오버레이의 움직임이 가능한 공간을 표현하기 위하여 고도(azimuth) 범위, 방위(elevation) 범위 및 깊이(depth) 범위에 대한 정보가 이용될 수 있다. 이 경우, 오버레이가 뷰포트 내에서 움직이는 경우뿐만 아니라 그 외의 공간까지도 정의할 수 있으며, 수신기의 영역 내 여부 처리 방법은 뷰포트 영역 내 여부 처리 방법과 같을 수 있다.

[0395] 또한, 일 실시예는 추가적으로 각 오버레이의 움직임에 대한 제한 여부를 정할 수 있다. 또는 각 오버레이의 움직임을 한정시킬 수 있다. 이를 위해 각 축에 대한 회전 범위 및 스케일 범위에 대한 정보가 이용될 수 있다.

[0396] 상술한 정보들은 오버레이 인터랙션 관련 정보 또는 오버레이 인터랙션 정보일 수 있으며, 오버레이 인터랙션 메타데이터에 포함될 수 있고, 오버레이 인터랙션 메타데이터는 시그널링될 수 있다. 또는 오버레이 인터랙션 관련 정보는 OverlayInteractionStruct()에 포함될 수 있으며, OverlayInteractionStruct()는 오버레이 인터랙션 메타데이터에 포함될 수 있다. OverlayInteractionStruct()는 예를 들어, 표 20과 같이 다음을 포함할 수 있다.

표 20

```

aligned(8) class OverlayInteractionStruct() {
    unsigned int(1) switch_on_off_flag;
    unsigned int(1) change_opacity_flag;
    unsigned int(1) position_flag;
    unsigned int(1) depth_flag;
    unsigned int(1) rotation_flag;
    unsigned int(1) resize_flag;
    unsigned int(1) limit_in_viewport_flag;
    unsigned int(1) limit_transform_flag;

    if (switch_on_off_flag == 1) {
        unsigned int(32) available_levels;
        for(i=0; i< available_levels; i++) {
            unsigned int(32) ref_overlay_IDs[i];
            unsigned int(1) altr_track_flag;
            bit(7) reserved=0;
        }
    }
    if (change_opacity_flag == 1) {
        unsigned int(16) opacity_min;
        unsigned int(16) opacity_max;
    }
    if (limit_transform_flag) {
        if (position_flag == 1) {
            unsigned int(32) azimuth_min;
            unsigned int(32) azimuth_max;
            unsigned int(32) elevation_min;
            unsigned int(32) elevation_max;
        }
        if (depth_flag == 1) {
            unsigned int(32) depth_min;
            unsigned int(32) depth_max;
        }
        if (rotation_flag == 1) {
            unsigned int(1) rotation_x_axis_flag;
            unsigned int(1) rotation_y_axis_flag;
            unsigned int(1) rotation_z_axis_flag;
            bit(5) reserved=0;
            if (rotation_x_axis_flag) {
                unsigned int(32) x_rotation_min;
                unsigned int(32) x_rotation_max;
            }
            if (rotation_y_axis_flag) {
                unsigned int(32) y_rotation_min;
                unsigned int(32) y_rotation_max;
            }
            if (rotation_z_axis_flag) {
                unsigned int(32) z_rotation_min;
                unsigned int(32) z_rotation_max;
            }
        }
    }
    if (resize_flag == 1) {
        unsigned int(32) resize_min;
        unsigned int(32) resize_max;
    }
}
}
    
```

[0397]

[0398]

표 20에서, switch_on_off_flag 필드는 오버레이를 보여주거나 감출 수 있는 인터랙션을 허용하는 플래그를 의미할 수 있고, change_opacity_flag 필드는 오버레이 평면의 글로벌 불투명도(opacity)를 조절할 수 있도록 허용하는 플래그를 의미할 수 있다. position_flag 필드, depth_flag 필드, rotation_flag 필드 및 resize_flag 필드는 각각 위치, 깊이, 회전, 스케일을 변경할 수 있도록 허용하는 플래그를 의미할 수 있고, limit_in_viewport_flag 필드는 움직임을 뷰포트 영역으로 한정하는 플래그를 의미할 수 있다. 또한, limit_transform_flag 필드는 각 오버레이가 움직이는 범위가 한정되어 있는지 표시하는 플래그를 의미할 수 있다.

[0399]

여기서, switch_on_off_flag 필드 값이 1인 경우, available_levels 필드는 변경 가능한 레벨의 개수를 지시할 수 있다. available_levels 필드 값이 0인 경우, 오버레이의 visibility가 on/off될 수 있음을 지시할 수

있다. 또한, available_levels 필드 값이 0보다 큰 경우, ref_overlay_IDs 필드를 통해 참조 오버레이 ID를 명시할 수 있다. 즉, 변경 가능한 레벨의 개수가 적어도 하나가 존재하는 경우, 이를 위해 참조할 오버레이를 지시할 수 있다. 또한, altr_track_flag 필드는 오버레이 미디어가 다른 트랙이나 다른 이미지 아이템(item)에 포함되는지의 관련 정보를 지시할 수 있다. 여기서, altr_track_flag 필드가 1인 경우, 오버레이 미디어가 다른 트랙이나 다른 이미지 아이템에 포함될 수 있고, 'altr'로 EntityGroupToBox를 통해 그룹핑된 엔티티(entity)의 소스로 변경할 수 있다. 즉, grouping_type 필드 값이 altr인 EntityGroupToBox를 통해 그룹핑된 엔티티의 소스로 변경할 수 있다.

- [0400] change_opacity_flag 필드 값이 1인 경우, opacity_min 필드 및 opacity_max 필드는 불투명도의 최소 값 및 최대 값을 지시할 수 있다. position_flag 필드 값이 1인 경우, 위치 정보를 지시하는 azimuth_min 필드, azimuth_max 필드, elevation_min 필드 및 elevation_max 필드를 변경할 수 있다. 여기서, azimuth_min 필드, azimuth_max 필드, elevation_min 필드 및 elevation_max 필드는 각각 최소 고도 값, 최대 고도 값, 최소 방위 값 및 최대 방위 값을 지시할 수 있다. 또한, limit_transform_flag 필드 값이 1이므로, 오버레이의 움직임 범위를 지정할 수 있다.
- [0401] depth_flag 필드 값이 1인 경우, 각각 최소 깊이 값 및 최대 깊이 값을 지시하는 depth_min 필드 및 depth_max 필드를 조절할 수 있고, 이에 따라 깊이값 변경의 범위를 지정할 수 있다. 이 경우 깊이는 오버레이의 크기를 유지한 상태에서 깊이 값이 변경될 수 있다.
- [0402] 또한, rotation_x_axis_flag 필드, rotation_y_axis_flag 필드 및 rotation_z_axis_flag 필드는 각각 x축, y축, z축에 대한 회전 가능 여부를 지정하는 플래그를 의미할 수 있다. 여기서, 각 축에 대한 회전 여부가 1일 경우, 각 축에 대한 회전 각도의 범위를 지정할 수 있다. 즉, rotation_x_axis_flag 필드 값이 1인 경우 각각 최소 및 최대 x축에 대한 회전 값을 지시하는 x_rotation_min 필드 및 x_rotation_max 필드를 조절할 수 있고, rotation_y_axis_flag 필드 값이 1인 경우 각각 최소 및 최대 y축에 대한 회전 값을 지시하는 y_rotation_min 필드 및 y_rotation_max 필드를 조절할 수 있고, rotation_z_axis_flag 필드 값이 1인 경우 각각 최소 및 최대 z축에 대한 회전 값을 지시하는 z_rotation_min 필드 및 z_rotation_max 필드를 조절할 수 있다.
- [0403] resize_flag 필드 값이 1인 경우, 각각 최소 오버레이 크기 및 최대 오버레이 크기를 지시하는 resize_min 필드 및 resize_max 필드를 변경할 수 있으며, 이를 조절하여 스케일의 범위를 지정할 수 있다. 여기서, 오버레이의 종횡비(aspect ratio)를 고려하여 스케일이 같은 비율로 적용될 수 있다.
- [0404] 도 46은 오버레이 인터랙션 제공 방법을 설명하는 순서도의 예이다.
- [0405] 도 46을 참조하면, 일 실시예는 오버레이를 선택하고 움직임 등과 같은 사용자 입력을 획득하는 경우, 해당 오버레이가 인터랙션이 가능한 오버레이인지 판단할 수 있으며, 인터랙션이 불가능한 오버레이인 경우, 관련 과정을 종료할 수 있다. 또한, 인터랙션이 가능한 오버레이인 경우, 오버레이의 위치/크기/회전/변경이 가능한지를 판단할 수 있으며, 불가능한 경우, 관련 과정을 종료할 수 있다. 다만, 가능한 경우, 일 실시예는 변경 위치를 계산할 수 있으며, 오버레이의 움직임 범위가 정해져 있는지 판단할 수 있다.
- [0406] 일 실시예는 오버레이의 움직임 범위가 정해져 있는 경우, 움직임이 범위 내에 있는지 판단할 수 있으며, 범위 내에 있는 경우, 뷰포트 영역에서의 움직임 제한이 있는지 판단할 수 있다. 다만, 범위 내에 있지 않는 경우, 이전 위치/크기/회전으로 설정 또는 보상 계산을 통해 값을 설정할 수 있고, 이후, 뷰포트 영역에서의 움직임 제한이 있는지 판단할 수 있다. 또한, 오버레이의 움직임 범위가 정해져 있지 않는 경우에도 뷰포트 영역에서의 움직임 제한이 있는지 판단할 수 있다.
- [0407] 일 실시예는 움직임 제한이 있는 경우, VFC(viewing frustum culling) 체크를 수행할 수 있고, 뷰잉 프러스텀(viewing frustum) 내에 있는지 판단할 수 있다. 여기서, 뷰잉 프러스텀 내에 있는 경우, 메인 VR 미디어와 오버레이 미디어를 합성하고 렌더링할 수 있다. 다만, 뷰잉 프러스텀 내에 있지 않는 경우, 이전 위치/크기/회전으로 설정 또는 보상 계산을 통해 값을 설정할 수 있고, 이후, 메인 VR 미디어와 오버레이 미디어를 합성하고 렌더링할 수 있다. 또한, 움직임 제한이 없는 경우에도 메인 VR 미디어와 오버레이 미디어를 합성하고 렌더링할 수 있다.
- [0408] 일 실시예는 메인 VR 미디어 및 오버레이 미디어의 합성 및 렌더링을 수행하고, 사용자 뷰포트를 렌더링할 수 있다.
- [0409] 일 실시예에서, 오버레이 메타데이터는 상술한 바에 따라 오버레이의 위치 정보, 크기 정보, 렌더링 속성 정보 및 인터랙션 정보 중 적어도 하나를 포함할 수 있다. 또는 오버레이 메타데이터는 상술한 바에 따라 오버레이

위치 관련 정보(위치 및 크기), 오버레이 렌더링 속성 정보, 오버레이 렌더링 기타 정보 및 오버레이 인터랙션 정보를 포함할 수 있다. 오버레이 메타데이터는 OverlayInfoStruct()를 포함할 수 있으며, OverlayInfoStruct()는 오버레이 위치 관련 정보(위치 및 크기), 오버레이 렌더링 속성 정보, 오버레이 렌더링 기타 정보 및 오버레이 인터랙션 정보를 포함할 수 있다. OverlayInfoStruct()는 예를 들어, 표 21과 같이 다음을 포함할 수 있다.

표 21

```
aligned(8) class OverlayInfoStruct() {
    unsigned int(32) overlay_id;
    unsigned int(16) overlay_source_id;
    unsigned int(1) overlay_essential_flag;
    unsigned int(1) overlay_priority;

    OverlayPosStruct();
    OverlayRenderStruct();
    OverlayMiscStruct();
    OverlayInteractionStruct();
}
```

[0410]

[0411] 표 21에서, overlay_id 필드는 오버레이 메타데이터 식별자를 지시할 수 있고, overlay_source_id 필드는 오버레이 미디어 소스데이터의 식별자를 지시할 수 있다. overlay_essential_flag 필드는 오버레이가 필수적으로 오버레이되어야 하는 오버레이인지 여부를 지시할 수 있고, overlay_priority 필드는 오버레이 미디어의 오버레이 시 우선 순위를 지시할 수 있다. 여기서, 우선 순위는 디코딩에 영향을 줄 수 있다.

[0412] 또한, OverlayPosStruct()는 오버레이 위치 관련 정보를 포함할 수 있고, 예를 들어, 표 11과 같을 수 있다. OverlayRenderStruct()는 오버레이 렌더링 속성 정보 또는 오버레이 렌더링 속성 관련 정보를 포함할 수 있고, 예를 들어, 표 15과 같을 수 있다. OverlayMiscStruct()는 오버레이 기타 정보를 포함할 수 있고, 예를 들어, 표 18과 같을 수 있다. OverlayInteractionStruct()는 오버레이 인터랙션 정보를 포함할 수 있고, 예를 들어, 표 20과 같을 수 있다.

[0413] 도 47은 동적 오버레이 메타데이터의 구성 예를 나타낸다.

[0414] 예를 들어, 동적 오버레이 메타데이터가 타임드(timed)-메타데이터로 구성되는 경우, 도 47에 도시된 바와 같이 OverlaySampleEntry가 정의 되고, OverlaySampleEntry는 MetadataSampleEntry를 상속받고, OverlayConfigBox를 호출할 수 있다. OverlayConfigBox내에는 스테틱(static)한 오버레이 렌더링 메타데이터를 정의할 수 있다. 실제 동적인 오버레이 메타데이터는 샘플에 저장될 수 있다. OverlaySample에는 오버레이 개수의 OverlayInfoStruct로 구성될 수 있다. 여기서, OverlaySampleEntry, OverlayConfigBox 및 OverlaySample은 도 47에 도시된 바와 같을 수 있고, OverlayInfoStruct는 표 21와 같을 수 있다.

[0415] 오버레이 미디어가 시간에 따라 오버레이 위치 혹은 렌더링 속성 등이 변화하는 경우를 지원하기 위하여 오버레이 메타데이터를 별도의 메타데이터 트랙으로 저장하여 전달할 수 있다. 해당 오버레이 미디어 메타데이터 트랙은 하나 이상의 샘플을 포함할 수 있으며, 각 샘플은 하나 이상의 오버레이 메타데이터를 포함할 수 있다. 각 샘플은 하나 이상의 OverlayInfoStruct를 포함할 수 있다.

[0416] 도 48은 동적 오버레이 메타데이터 트랙과 오버레이 미디어 트랙 링크 시그널링 예를 나타낸다.

[0417] 오버레이 메타데이터 트랙의 TrackReferenceTypeBox 을 이용하여 오버레이 미디어 트랙을 지칭할 수 있다. 즉, reference type 값을 'cdsc'로 할당하고 track_IDs 에 하나 이상의 오버레이 미디어 트랙 식별자 혹은 트랙 그룹 식별자 (오버레이 미디어가 하나 이상의 트랙을 통하여 전달되는 경우)를 지칭함으로써, 오버레이 메타데이터와 연관된 오버레이 미디어 트랙을 지칭 할 수 있다.

[0418] 도 49는 오버레이 메타데이터와 관련 오버레이 미디어의 링킹 예를 나타낸다.

[0419] 일 실시예에서는 오버레이 메타데이터 트랙이 참조하고 있는 오버레이 미디어 트랙을 'cdsc'를 통해 명시할 수

있다. 오버레이 메타데이터는 하나 이상의 오버레이 미디어 트랙을 참조할 수 있다. 여기서, 오버레이 미디어 트랙과의 링킹(linking)은 'cdsc'를 사용할 수 있으나, 오버레이 미디어가 메타데이터 트랙내에 저장되어 있는 경우는 'cdsc'를 사용할 수 없다.

- [0420] 다만, 메타데이터 트랙이 오버레이 미디어 콘텐츠를 가지고 있는 경우도 존재할 수 있다. 이러한 경우, 오버레이 미디어 트랙이 메타데이터 트랙으로 구성되어, 메타데이터 트랙인 오버레이 미디어 트랙을 오버레이 렌더링 메타데이터 트랙이 참조할 경우에 대한 방법이 요구될 수 있다. 이러한 경우, 'cdsc'를 통해 참조 트랙을 연결할 수 없으며, 예를 들어, 추천 뷰포트(Recommended Viewport)가 이러한 경우일 수 있다.
- [0421] 도 50은 추천 뷰포트 오버레이 예를 나타낸다.
- [0422] 추천 뷰포트는 타임드-메타데이터에 시간별 추천하는 뷰포트의 위치들이 저장될 수 있다. 이러한 추천 뷰포트는 사용자의 뷰포트를 자동으로 변경할 수도 있으나, VR 미디어가 렌더링될 때 특정 위치에 오버레이로 보여질 수도 있다.
- [0423] 도 50에서 왼쪽 및 오른쪽에 보이는 윈도우가 추천 뷰포트의 오버레이에 해당할 수 있다. 이러한 경우, 오버레이 미디어인 메타데이터 트랙과 오버레이 렌더링 메타데이터 트랙을 링킹(linking)하는 방법이 요구될 수 있다.
- [0424] 도 51은 'ovrc' 트랙 참조의 예를 나타낸다.
- [0425] 일 실시예에서는 ROI(region of interest) 영역 등과 같은 VR 미디어의 특정 영역을 VR 미디어 상에 오버레이할 수 있다. 이를 지원하기 위하여 별도의 오버레이 메타데이터 트랙과 VR 미디어의 추천 뷰포트 등을 포함하는 메타데이터 트랙이 존재하는 경우, 오버레이 메타데이터 트랙과 VR 미디어의 메타데이터 트랙 간의 관계성을 시그널링할 수 있다.
- [0426] 일 실시예에서는 오버레이 메타데이터 트랙의 TrackReferenceTypeBox을 이용하여 오버레이 메타데이터가 적용될 메타데이터 트랙(추천 뷰포트 메타데이터 트랙 등)을 지칭할 수 있다. 이를 위하여 새로운 참조 타입(reference type)을 추가하여, 즉 reference_type 필드 값이 'ovrc'이며 track_IDs 에 하나 이상의 메타데이터 트랙(추천 뷰포트 메타데이터 트랙 등) 또는 오버레이 미디어 아이템(item) 식별자를 지칭함으로써, 오버레이 메타데이터가 적용되는 메타데이터 트랙 및 이미지 아이템을 지칭할 수 있다. 'ovrc' 및 track_IDs 필드를 통해 지칭되는 트랙(들)은 현재 오버레이 메타데이터가 적용되는 메타데이터 트랙(들) 또는 이미지 아이템일 수 있다. TrackReferenceBox 및 TrackReferenceTypeBox는 예를 들어, 표 22와 같이 다음을 포함할 수 있다.

표 22

```
aligned(8) class TrackReferenceBox extends Box( 'tref' ) {
}
aligned(8) class TrackReferenceTypeBox (unsigned int(32) reference_type) extends Box {
unsigned int(32) track_IDs[];
}
```

- [0427] 표 22에서, track_ID 필드는 프레젠테이션(presentation) 내에서 컨테이너 트랙(containing track)으로부터 다른 트랙(another track) 또는 이미지 아이템(image item ID)으로의 레퍼런스를 제공하는 정수일 수 있으며, track_IDs 필드는 재사용될 수 없으며, 0과 같은 값을 가질 수 없다. 또한, reference_type 필드는 상술한 바와 같이 지칭 또는 지시할 수 있고, 다음의 값들(following values) 중 하나로 설정될 수 있다.
- [0429] 도 52는 메타데이터 트랙 그룹핑 예를 나타낸다.
- [0430] 일 실시예에서는 ROI(region of interest) 영역 등과 같은 VR 미디어의 특정 영역을 VR 미디어 상에 오버레이할 수 있다. 이를 지원하기 위하여 별도의 오버레이 메타데이터 트랙과 VR 미디어의 추천 뷰포트 등을 포함하는 메타데이터 트랙이 존재하는 경우, 오버레이 메타데이터 트랙과 VR 미디어의 메타데이터 트랙 간의 관계성을 시그널링 해 줄 수 있어야 한다.
- [0431] 일 실시예에서는 Track_group_type 필드 값이 'mtgr'인 TrackGroupTypeBox는 360 씬(scene)에서 오버레이 등의 미디어에 함께 적용되는 메타데이터 트랙 그룹을 지칭할 수 있다. 동일한 track_group_id 필드 값을 가지는 트랙들은 360 씬에서 오버레이 등과 함께 적용되어 프로세싱될 수 있다. TrackGroupTypeBox는 MetadataGroupBox

를 포함할 수 있으며, MetadataGroupBox는 표 23과 같이 다음을 포함할 수 있다.

표 23

```
aligned(8) class MetadataGroupBox extends TrackGroupTypeBox( 'mtgr' ) {
    unsigned int(7) metadata_type;
    unsigned int(1) metadata_essential_flag;
}
```

[0432]

[0433]

표 23에서, metadata_type 필드는 메타데이터의 타입을 지시할 수 있다. 예를 들어, metadata_type 필드 값이 0인 경우 추천 뷰포트 메타데이터를 지시할 수 있고, 1인 경우 오버레이 메타데이터를 지시할 수 있다. 또한, metadata_essential_flag 필드는 메타데이터가 필수적으로 처리 및 미디어에 적용되어야 하는지 여부를 나타내는 플래그를 의미할 수 있다. 필수적으로 해당 메타데이터가 처리 및 미디어에 적용되어야 하는 경우 해당 메타데이터 처리를 지원하지 않는 플레이어는 연관된 미디어를 재생하지 않을 수 있다.

[0434]

일 실시예에서 샘플 엔트리 타입(sample entry type)이 'rcvp'인 타임드 메타데이터 트랙은 0개 또는 1개의 SampleToGroupBox를 포함할 수 있다. 여기서, SampleToGroupBox의 grouping_type 필드는 'ovmt'일 수 있다. SampleToGroupBox는 타임드 메타데이터(및 결과적으로 미디어 트랙들 내의 해당 샘플들) 내의 샘플들을 특정 오버레이 메타데이터에 할당하는 정보를 나타낼 수 있다.

[0435]

group_type 필드 값이 'ovmt'인 SampleToGroupBox가 존재하는 경우, 이에 따라 동일한 그룹 타입(group type)을 가지는 SampleGroupDescriptionBox도 존재할 수 있고, SampleGroupDescriptionBox는 샘플 그룹이 속하는 특정 오버레이 메타데이터의 ID를 포함할 수 있다. group_type 필드 값이 'ovmt'인 샘플 그룹 엔트리(sample group entry), 즉 OverlayMetaRefEntry는 예를 들어, 표 24와 같이 다음을 포함할 수 있다.

표 24

```
class OverlayMetaRefEntry() extends SampleGroupDescriptionEntry( ' ovmt' ) {
    OverlayInfoStrut();
}
```

[0436]

[0437]

표 24에서, OverlayInfoStrut()는 그룹에 포함된 메타데이터 샘플들에 적용될 오버레이 메타데이터를 포함할 수 있으며, 표 21와 같을 수 있다.

[0438]

일 실시예에서는 트랙을 통합하여 오버레이 미디어 메타데이터 트랙을 확장할 수도 있다. 따라서, 메타데이터 트랙이 오버레이 미디어 콘텐츠 데이터와 오버레이 렌더링 메타데이터를 함께 담고 있도록 확장하여 링킹(linking)을 하지 않아도 될 수 있도록 구성할 수 있다. 추천 뷰포트를 예로 들 수 있으며, 이를 지원하기 위해서는 OverlayRcvpSampleEntry가 이용될 수 있다. OverlayRcvpSampleEntry는 표 25과 같이 다음을 포함할 수 있다.

표 25

```
class OverlayRcvpSampleEntry() extends RcvpSampleEntry( ' ovmv' ) {
    // overlay rendering info(position, size, opacity ... etc)
    ...
}
```

[0439]

[0440]

도 53은 VR 미디어 상에 배치된 오버레이를 지원하는 송신기의 아키텍처 예를 나타낸다.

[0441]

일 실시예에 따른 송신기는 오버레이 미디어를 획득하여 저작자가 오버레이의 위치/크기/렌더링 옵션 등을 조절하여 생성한 메타데이터 및 오버레이 미디어 데이터를 그대로 파일/세그먼트 인캡슐레이터의 처리를 통해 수신

기로 전송 또는 전달할 수 있다. 또는, 패킹과 프로젝션 방법에 따라 디코딩 후 오버레이에 특정 프로젝션을 적용하거나 적용하지 않을 수 있으며, 이후, 텍스처 아틀라스 패킹 또는 리전별 패킹을 수행하여 분리된 오버레이 미디어 트랙 또는 오버레이 미디어 트랙이 함께 패킹된 VR 미디어 트랙을 인코딩하고, 파일/세그먼트 인캡슐레이터의 처리를 수행하여 수신기로 전달할 수 있다.

[0442] 도 54는 VR 미디어 상에 배치된 오버레이를 지원하는 송신기의 아키텍처 예를 나타낸다.

[0443] 일 실시예에 따른 수신기는 전달 받은 데이터를 디캡슐레이션하고, 오버레이 메타데이터를 렌더링하는 렌더러에게 전달할 수 있다. 오버레이될 미디어 데이터는 디코딩될 수 있으며, 디코딩 후, 리전별 패킹 또는 텍스처 아틀라스로 패킹되어 있는 경우, 각각의 오버레이들을 언패킹하여 렌더러에게 전달할 수도 있다. 또는, 전체 데이터를 렌더러로 전달하며, 렌더러가 패킹 정보를 통해 렌더링시에 조절할 수도 있다. 일 실시예에 따른 수신기는 상술한 두 가지 중 하나를 지원할 수 있고, 두 가지 모두를 지원할 수도 있으며, 수신기의 하드웨어 사양에 따라 적용 방법을 수신기에서 조정할 수도 있다.

[0444] 도 55는 오버레이 미디어 트랙 상에 오버레이 메타데이터 시그널링의 또 다른 예를 나타낸다.

[0445] 도 55를 참조하면, 일 실시예에서 오버레이 메타데이터는 오버레이 미디어 트랙 상에도 다음과 같은 방법으로 시그널링 될 수 있다.

[0446] 오버레이 미디어 트랙의 샘플 엔트리는 OverlayConfigBox를 포함할 수 있다. 이를 통하여 해당 미디어 트랙이 오버레이 미디어를 포함하고 있으며, 트랙 내에 포함된 오버레이 미디어 관련 메타데이터를 시그널링 할 수 있다. OverlayConfigBox는 오버레이 메타데이터에 포함될 수 있으며, 예를 들어, 표 26과 같이 다음을 포함할 수 있다.

표 26

```
class OverlayConfigBox extends FullBox('ovcf', 0, 0) {
    unsigned int(8) num_overlays;
    OverlayMediaPackingStruct(num_overlays)
    for (i=0; i< num_overlays; i++) {
        OverlayInfoStruct();
    }
}
```

[0447]

[0448] 표 26에서, num_overlay 필드는 오버레이 미디어 트랙의 각 샘플에 포함된 오버레이 미디어의 개수 또는 샘플에 포함된 최대 오버레이 미디어의 개수를 지시할 수 있다. OverlayMediaPackingStruct()는 오버레이 미디어의 프로젝션 및 패킹 정보를 포함할 수 있고, 표 1과 같을 수 있다. 또한, OverlayInfoStruct()는 오버레이 메타데이터를 포함할 수 있으며, 이는 트랙의 샘플에 포함된 오버레이 미디어에 적용될 수 있고, 표 21과 같을 수 있다.

[0449] 일 실시예에서 오버레이 미디어 트랙은 grouping_type 필드 값이 'ovgr' 인 SampleToGroupBox를 포함할 수 있다. SampleToGroupBox는 해당 트랙에 포함된 샘플들 중 동일한 오버레이 메타데이터가 적용될 샘플들을 지칭할 수 있다.

[0450] 해당 트랙에 grouping_type 필드 값이 'ovgr' 인 SampleToGroupBox가 존재하는 경우, grouping_type 필드 값이 'ovgr' 인 SampleGroupDescriptionBox가 존재할 수 있고, 해당 샘플들에 공통적으로 적용되는 다음과 같은 정보가 포함될 수 있다. grouping_type 필드 값이 'ovgr' 인 샘플 그룹 엔트리(sample group entry)는 OverlayGroupEntry로 지칭할 수 있으며, 예를 들어, 표 27과 같이 다음을 포함할 수 있다.

표 27

```
class OverlayGroupEntry() extends SampleGroupDescriptionEntry(' ovmm' ) {
    OverlayInfoStruct();
}
```

[0451]

[0452] 표 27에서, OverlayinfoStruct()는 그룹에 포함된 샘플들에 적용되는 오버레이 메타데이터를 포함할 수 있으며, 표 21와 같을 수 있다. 또한, ovmm은 ovgr로 대체될 수도 있다.

[0453] 도 56은 오버레이 미디어 패킹, 프로젝션과 디폴트 렌더링 시그널링 예들을 나타낸다.

[0454] 도 57은 오버레이 미디어 패킹, 프로젝션과 디폴트 렌더링 시그널링 다른 예들을 나타낸다.

[0455] 도 56은 오버레이 미디어 트랙이 이미지인 경우를 나타낼 수 있으며, 도 57은 오버레이 미디어 트랙이 비디오인 경우를 나타낼 수 있다.

[0456] 일 실시예에서 오버레이 미디어 트랙은 샘플 엔트리(sample entry)에 상속한 OverlayConfigBox를 포함할 수 있으며, 동시에 grouping_type 필드 값이 'ovgr' 인 SampleToGroupBox 및 OverlayGroupEntry()를 포함할 수 있다. 이러한 경우 OverlayGroupEntry()에 연관된 오버레이 미디어 샘플들에 포함된 오버레이 메타데이터가 적용될 수 있다.

[0457] 또는, 오버레이 디폴트 렌더링 정보를 오버레이 미디어 트랙 내에 프로젝션 및 패킹 정보와 함께 명시하기 위하여 트랙 내에 존재하는 오버레이 개수인 num_overlay 필드를 도 56의 OverlayConfigProperty 또는 도 57의 OverlayConfigBox 내부에서 정의하고, 파라미터로 전달하는 방식으로 변경하고, OverlayInfoStruct()를 추가할 수 있다. 이러한 경우, 오버레이 메타데이터에 포함되는 OverlayMediaPackingStruct는 표 28과 같이 다음을 포함할 수 있다.

표 28

```
aligned(8) class OverlayMediaPackingStruct(num_overlays) {
    unsigned int(5) num_regions;
    for(i = 0; i < num_regions; i++) {
        unsigned int(8) overlay_region_id[i];
        unsigned int(16) overlay_region_width[i];
        unsigned int(16) overlay_region_height[i];
        unsigned int(16) overlay_region_top[i];
        unsigned int(16) overlay_region_left[i];
    }
    for (i = 0; i < num_overlays; i++) {
        unsigned int(16) overlay_source_id[i];
        unsigned int(5) projection_type[i];
        unsigned int(3) packing_type;
        unsigned int(8) overlay_region_id[i];
        if(packing_type != 0) {
            unsigned int(1) guard_band_flag[i];
            if (packing_type == 1)
                TextureAtlasPacking(i);
            else if (packing_type == 2)
                RectRegionPacking(i);
            else if (packing_type == 3)
                PolygonRegionPacking(i);
            if (guard_band_flag[i])
                GaurdBand(i);
        }
    }
}
```

[0458]

[0459] 표 28에서, 각 필드들은 표 1의 각 필드들에 대응할 수 있으며, 동일하게 정보들을 지시할 수 있으나, 이에 한정되는 것은 아니다.

[0460] 일 실시예에 따른 오버레이는 VR 미디어 또는 360도 미디어 내에 보충 정보(supplement information), 광고 및 로고 등을 추가하기 위해 사용될 수 있다. 또한, 오버레이는 VR 미디어뿐만 아니라 AR(Augmented Reality/MR(Mixed Reality)에서 360도 비디오/이미지 대신에 시각을 통해(see-through)로 보이는 360도 실제 환경에 오버레이를 추가할 수 있으므로, AR/MR 오버레이 시그널링으로 확장이 가능할 수 있다.

[0461] 일 실시예는 VR 미디어 또는 360도 미디어 내에 오버레이 미디어와 렌더링 관련 메타데이터들을 명시하는 방법과 시그널 방법을 제공할 수 있으며, 오버레이 미디어 트랙에는 프로젝션과 패킹 정보를, 메타데이터 트랙에는 시간에 따른 렌더링 정보(위치, 크기, 속성 및 인터랙션 정보 등)를 시그널 하는 방식으로 구성될 수 있다. 또한, 일 실시예에서 오버레이 미디어 트랙에는 프로젝션, 패킹 및 디폴트 렌더링 정보를 포함할 수 있고, 메타데이터 트랙은 상술한 바와 같이 시간에 따른 렌더링 정보를 포함할 수 있다.

[0462] 도 58은 VR 미디어 트랙, 오버레이 미디어 트랙 및 오버레이 미디어 아이템의 그룹핑 예를 나타낸다.

[0463] 일 실시예에서, 파일 #1의 경우, 메인 VR 미디어와 오버레이 미디어가 별도의 트랙으로 파일 상에 포함되어 있는 경우, grouping_type 필드 값이 'ovgr' 을 가지는 EntityToGroupBox는 메인 VR 미디어와 오버레이 미디어를 포함하는 트랙 및/또는 아이템(item)의 그룹을 지칭할 수 있다. 오버레이 미디어는 비디오 및 이미지 등을 포함하기 때문에 트랙뿐만 아니라 이미지 아이템을 포함할 수 있다. 즉, 이를 통해 360 씬(scene)에서 오버레이 등과 함께 렌더링 될 수 있는 트랙 그룹을 지칭 할 수 있다. 동일한 group_id 필드 값을 가지는 트랙/아이템들은 360 씬(scene)에서 오버레이 등과 함께 렌더링 될 수 있음을 나타낼 수 있다. 따라서, 이를 통해 플레이어는 메인 미디어 및 오버레이 미디어를 편리하게 찾을 수 있다(retrieve).

[0464] 도 58을 참조하면, VR 미디어 트랙 #1은 오버레이 미디어 아이템 #1 및 오버레이 미디어 트랙 #1~N과 그룹핑될 수 있고, 오버레이 미디어 아이템 #1 및 오버레이 미디어 트랙 #1~N 중 일부와 그룹핑될 수도 있다. 이는 오버레이 엔티티 그룹(overaly entity group)이라 지칭될 수 있다. 오버레이 엔티티 그룹 내의 트랙 및/또는 아이템은 동일한 group_id 필드 값을 포함할 수 있다. 또는 group_id 필드 값이 동일한 트랙 및/또는 아이템은 동일한 그룹 내에 포함될 수 있으며, 함께 렌더링될 수 있다. 여기서, VR 미디어 트랙은 메인 미디어 트랙 또는 메인 VR 미디어 트랙을 지칭할 수 있다. 또한, 상술한 정보/필드들은 오버레이 관련 메타데이터에 포함될 수 있다. 또한, 이러한 경우 트랙 및/또는 아이템은 OverlayVideoGroupBox를 포함할 수 있으며, OverlayVideoGroupBox는 EntityToGroupBox에 포함될 수 있고, 예를 들어, 표 29과 같이 다음을 포함할 수 있다.

표 29

```
aligned(8) class OverlayVideoGroupBox extends EntityToGroupBox( 'ovgr ' ) {
    unsigned int(5) media_type;
    unsigned int(1) main_media_flag;
    unsigned int(1) overlay_media_flag;
    if ((media_type == 1) || (overlay_media_flag)){
        unsigned int(1) overlay_essential_flag;
    }
    bit(1) reserved = 0;
}
```

[0465]

[0466] 표 29에서, media_type 필드는 트랙 그룹 내에서 미디어의 타입을 지시할 수 있다. 예를 들어, media_type 필드 값이 0인 경우 메인 미디어임을 지시할 수 있고, 1인 경우 오버레이 미디어임을 지시할 수 있다. 또한, main_media_flag 필드는 메인 미디어인지 여부를 나타내는 플래그를 의미할 수 있고, overlay_media_flag 필드는 오버레이 미디어인지 여부를 나타내는 플래그를 의미할 수 있다. overlay_essential_flag 필드는 오버레이 미디어가 필수적으로 오버레이되어야 하는지 여부를 나타내는 플래그를 의미할 수 있다. 여기서, 필수적으로 오버레이되어야 하는 오버레이 미디어인 경우, 오버레이를 지원하지 않는 플레이어는 동일 그룹 내에 메인 미디어를 재생하지 않을 수 있다.

[0467] 도 59는 본 발명에 따른 360 비디오 전송 장치에 의한 360 비디오 데이터 처리 방법을 개략적으로 나타낸다. 도 59에서 개시된 방법은 도 5 또는 도 16에서 개시된 360 비디오 전송 장치에 의하여 수행될 수 있다.

[0468] 도 59를 참조하면, 360 비디오 전송 장치는 360 비디오를 획득한다(S5900). 여기서 360 비디오는 적어도 하나의 카메라에 의해 캡처된 비디오/영상일 수 있다. 또는 360 비디오의 일부 또는 전부는 컴퓨터 프로그램 등에 의하여 생성된 가상의 영상일 수도 있다. 360 영상은 독자적인 정지 영상일 수 있고, 또는 360 비디오의 일부일 수 있다.

[0469] 360 비디오 전송 장치는 상기 360 비디오/영상을 처리하여 픽처를 도출한다(S5910). 360 비디오 전송 장치는 상술한 여러 프로젝트 포맷, 리전별 패킹 절차 등을 기반으로 2D 기반의 상기 픽처를 도출할 수 있다. 상기 도출된 픽처는 프로젝트드 픽처에 대응할 수 있고, 또는 팩드 픽처(리전별 패킹 과정이 적용된 경우)에 대응할 수도 있다.

[0470] 360 비디오 전송 장치는 상기 360 비디오/영상에 관한 메타데이터를 생성한다(S5920). 여기서, 상기 메타데이터는 본 명세서에서 상술한 필드들을 포함할 수 있다. 상기 필드들은 다양한 레벨의 박스(box)에 포함되거나 파일 내에서 별도의 트랙내의 데이터로 포함될 수 있다. 예를 들어, 상기 메타데이터는 상기 표 1 내지 29에서 상술한 필드/정보의 일부 또는 전부를 포함할 수 있다. 예를 들어, 상기 메타데이터는 상술한 오버레이 관련 메타데이터(정보/필드 포함)를 포함할 수 있다.

[0471] 예를 들어, 오버레이 관련 메타데이터는 오버레이의 그룹 정보를 포함할 수 있다.

[0472] 여기서, 오버레이의 그룹 정보는 서로 스위칭이 가능한 오버레이들에 대한 정보를 포함할 수 있다. 또한, 서로 스위칭이 가능한 오버레이들에 대한 정보는 ref_overlay_IDs 필드에 의해 지시되는 상기 서로 스위칭이 가능한 오버레이들에 대한 식별 정보를 포함할 수 있다. 여기서, ref_overlay_IDs 필드는 ref_overlay_id 필드라고 지칭할 수도 있으며, 스위칭 가능한 참조 오버레이 ID를 포함할 수 있다.

- [0473] 또한, 서로 스위칭 가능한 오버레이들에 대한 정보는 OverlayInteractionStruct에 포함될 수 있으며, 오버레이를 보여주거나 감출 수 있는 인터랙션이 허용되는 경우 포함될 수 있다. 즉, OverlayInteractionStruct 내에 switch_on_off_flag 필드에 의해 인터랙션이 허용되는 경우 포함될 수 있다. 또한, 인터랙션이 허용되는 경우, 스위칭 또는 변경 가능한 레벨의 개수 또는 오버레이의 개수가 지시될 수도 있다. 이는 available_levels 필드에 의해 지시될 수 있으며, 1개 이상인 경우, 상술한 서로 스위칭 가능한 오버레이들에 대한 정보를 포함할 수 있다. 이에 대한 상세한 설명은 표 20과 함께 상술하였다.
- [0474] 여기서, 오버레이의 그룹 정보는 오버레이와 함께 렌더링될 메인 미디어를 지시하는 정보를 포함할 수 있으며, 상술한 디코딩된 픽처는 메인 미디어를 포함할 수 있다. 또는 메인 미디어는 디코딩된 픽처일 수 있고, 디코딩된 픽처의 일부일 수 있다. 또한, 오버레이와 함께 렌더링될 메인 미디어를 지시하는 정보는 grouping_type 필드를 가지는 EntityToGroupBox에 의해 지시될 수 있다. 또는 EntityToGroupBox 내의 grouping_type 필드에 의해 지시될 수 있다. 즉, grouping_type 필드 값이 ovgr를 가지는 EntityToGroupBox는 메인 VR 미디어 및 오버레이 미디어를 포함하는 트랙 및/또는 아이템 그룹을 지칭할 수 있으며, 그룹 내의 메인 VR 미디어 및 오버레이 미디어는 함께 렌더링될 수 있다. 이는 메인 VR 미디어와 오버레이 미디어가 별도의 트랙에 포함되어 있는 경우에 이용될 수 있다. 즉, 함께 렌더링될 메인 VR 미디어 트랙과 오버레이 미디어 트랙이 링킹(linking) 또는 그룹핑(grouping)될 수 있다. 여기서, 메인 VR 미디어는 메인 미디어 또는 VR 미디어 또는 백그라운드(background) 미디어 또는 디코딩된 픽처 또는 디코딩된 픽처의 일부라 지칭할 수도 있다. 이에 대한 상세한 설명은 도 58과 함께 상술하였다.
- [0475] 예를 들어, 오버레이 관련 메타데이터는 현재 액티브 상태의 오버레이에 대한 식별 정보를 포함할 수 있다. 여기서, 액티브 상태는 비저블(visible) 상태로 지칭할 수도 있으며, 현재 액티브 상태의 오버레이는 스위칭 가능한 오버레이 중 현재 액티브되는 오버레이를 지칭할 수 있다. 또한, 식별 정보는 인덱스 또는 ID와 같은 고유 정보를 포함할 수 있다. 일 실시예는 식별 정보를 변경함으로써 스위칭 가능한 다른 오버레이를 액티브 상태로 변경할 수 있으며, 스위칭 가능한 오버레이들은 그룹핑되어 있을 수 있다. 또는 'altr'를 통해 그룹핑되어 있는 미디어일 수 있다. 또는 EntityToGroupBox의 grouping_type 필드 값이 altr인 경우 스위칭 가능한 오버레이들의 그룹을 지시할 수 있으며, 현재 액티브 상태의 오버레이는 상기 오버레이들의 그룹에 포함되는 오버레이일 수 있고, 식별 정보 변경에 따라 변경되는 오버레이도 상기 오버레이들의 그룹에 포함될 수 있다.
- [0476] 예를 들어, 오버레이 관련 메타데이터는 오버레이에 적용할 알파 컴포지션 타입(alpha composition type)에 대한 정보를 포함할 수 있다. 여기서, 알파 컴포지션은 오버레이 합성 시 오버레이 미디어가 알파 채널을 가지고 있으며 그 알파 값의 합성을 의미할 수 있다. 이는 composition_type 필드에 의해 지시될 수 있다. 또한, 오버레이 관련 메타데이터는 알파 컴포지션 적용 여부에 대한 정보도 포함할 수 있고, alpha_composition_flag 필드에 의해 지시될 수 있다. 여기서, 알파 컴포지션 타입은 알파 컴포지션 모드 또는 컴포지션 타입 또는 컴포지션 모드라 지칭할 수 있으며, 소스 오버(source over)를 포함할 수 있다. 소스 오버 또는 소스 오버 기반의 컬러 값의 결과는 표 16과 같이 계산될 수 있다. 알파 컴포지션 타입이 소스 오버를 지시하는 경우, composition_type 필드 값이 1일 수 있다. 또한, 알파 컴포지션 타입은 source_atop, source_in, source_out, dest_atop, dest_over, dest_in, dest_out, clear 및 xor 중 하나를 지시할 수도 있으며, 각각은 표 16과 같이 계산될 수 있다.
- [0477] 예를 들어, 오버레이 관련 메타데이터는 오버레이에 적용할 블렌딩 모드에 대한 정보를 포함할 수 있다. 또한, 블렌딩 적용 여부에 대한 정보도 포함할 수 있다. 여기서, 블렌딩은 알파 컴포지션보다 복잡한 오퍼레이션을 지칭할 수 있고, 픽셀의 색상을 블렌딩하는 것도 포함할 수 있다. 블렌딩 적용 여부에 대한 정보는 blending_flag 필드에 의해 지시될 수 있으며, 블렌딩 모드에 대한 정보는 blending_mode 필드에 의해 지시될 수 있다.
- [0478] 상술한 알파 컴포지션 적용 여부에 대한 정보, 알파 컴포지션 타입에 대한 정보, 블렌딩 적용 여부에 대한 정보 및 블렌딩 모드에 대한 정보 중 적어도 하나는 오버레이 렌더링 관련 정보 또는 오버레이 렌더링 관련 메타데이터에 포함될 수도 있다. 또는 OverlayRenderStruct에 포함될 수 있다. 이에 대한 상세한 설명은 표 15와 함께 상술하였다.
- [0479] 예를 들어, 오버레이 관련 메타데이터 중 스태틱(static) 메타데이터는 OverlayConfigBox에 저장될 수 있다. 또는 오버레이 관련 메타데이터 중 타임드(timed) 메타데이터는 샘플에 저장될 수 있다. 여기서, 스태틱 메타데이터는 시간에 따라 변하지 않는 메타데이터를 지칭할 수 있고, 타임드 메타데이터는 시간에 따라 변하는 메타데이터를 지칭할 수 있다.
- [0480] 여기서, SchemeTypeBox 내의 scheme_type 필드 값이 podv인 경우, OverlayConfigBox는 ProjectedOmniVideoBox

에 포함될 수 있다. 여기서, SchemeTypeBox는 RestrictedSchemeInfoBox에 포함될 수 있다. scheme_type 필드 값이 oldv인 경우 또는 오버레이가 프로젝션되지 않은 비디오인 경우 SchemeInformationBox가 OverlayConfigBox를 포함할 수 있으며, scheme_type 필드 값이 podv인 경우 또는 오버레이가 프로젝션된 비디오인 경우, VR ProjectedOmniVideoBox가 OverlayConfigBox를 포함할 수 있다. 이 경우, VR 미디어 트랙은 ProjectedOmniVideoBox를 포함할 수 있고, ProjectedOmniVideoBox는 OverlayConfigBox를 포함할 수 있다. 여기서, OverlayConfigBox는 오버레이 미디어의 프로젝션 및 패킹 정보를 포함할 수 있다. 또는 오버레이 미디어의 프로젝션 및 패킹 정보를 포함하는 OverlayMediaPackingStruct()를 포함할 수 있다. 이에 대한 상세한 설명은 도 34와 함께 상술하였다.

- [0481] 예를 들어, 메타데이터는 서로 스위칭이 가능한 픽처들에 대한 정보를 포함할 수 있다. 이는 grouping_type 필드 값이 altr인 EntityTogroupBox에 의해 지시될 수 있다. 즉, 대체 가능한 메인 VR 미디어들을 명시할 수 있다. 또는 서로 스위칭 가능한 픽처들은 그룹핑될 수 있고, 해당 그룹에 대한 정보를 명시할 수도 있다. 여기서, 픽처는 메인 VR 미디어 또는 메인 미디어 또는 백그라운드 미디어 또는 디코딩된 픽처 또는 디코딩된 픽처의 일부라 지칭할 수도 있다.
- [0482] 즉, 일 실시예에 따른 메타데이터는 서로 스위칭 가능한 오버레이들에 대한 정보를 포함할 수 있고, 서로 스위칭 가능한 픽처들(오버레이가 나타날 수 있는 백그라운드 미디어들)에 대한 정보를 포함할 수도 있다.
- [0483] 예를 들어, 360 영상 데이터는 복수의 트랙을 포함하고, 메타데이터는 각 트랙이 상기 디코딩된 픽처에 관한 트랙인지를 나타내는 플래그 및 상기 오버레이에 관한 트랙인지를 나타내는 플래그를 포함할 수 있다. 또는 메타데이터는 그룹 정보에 의해 지시되는 엔티티 그룹 내의 제1 트랙이 메인 미디어를 포함하는지를 나타내는 제1 플래그 및 상기 엔티티 그룹 내의 제2 트랙이 오버레이를 포함하는지를 나타내는 제2 플래그를 포함할 수 있다. 여기서, 제1 트랙 및 제2 트랙은 엔티티 그룹 내의 트랙을 지칭할 수 있으며, 서로 동일할 수 있고, 다를 수도 있다. 또한, 360 영상 데이터는 엔티티 그룹 또는 엔티티 그룹에 대한 정보를 포함할 수 있다. 즉, 일 실시예는 메인 미디어 트랙 및 오버레이 미디어 트랙을 그룹핑할 수 있으며, 이는 오버레이 엔티티 그룹 또는 엔티티 그룹이라 지칭할 수 있고, 동일한 group_id 필드 값을 포함할 수 있다. 여기서, 메타데이터는 트랙이 디코딩된 픽처에 대한 트랙인지 또는 메인 미디어 트랙인지에 대한 정보를 포함할 수 있고, 이는 main_media_flag 필드에 의해 지시될 수 있다. 또한, 메타데이터는 트랙이 오버레이에 대한 트랙인지에 대한 정보를 포함할 수 있고, 이는 overlay_media_flag 필드에 의해 지시될 수 있다. 이에 대한 상세한 설명은 표 29와 함께 상술하였다.
- [0484] 360 비디오 전송 장치는 상기 도출된 픽처를 인코딩한다(S5930). 360 비디오 전송 장치는 상기 2차원 픽처를 인코딩하고 비트스트림 형태로 출력할 수 있다.
- [0485] 360 비디오 전송 장치는 오버레이될 텍스처(미디어)의 종류에 따라 상기 오버레이 텍스처(미디어)를 인코딩하여 출력할 수도 있다. 이 경우 상기 인코딩된 오버레이 텍스처(미디어)는 후술하는 360 영상/비디오 데이터에 포함될 수 있다.
- [0486] 또는 상기 오버레이될 텍스처(미디어)는 360 비디오 수신 장치에 기 저장될 수도 있고, 네트워크를 통하여 별도로 전송될 수도 있다.
- [0487] 360 비디오 전송 장치는 상기 인코딩된 픽처 및 상기 메타데이터에 대하여 저장 또는 전송을 위한 처리를 수행한다(S5940). 360 비디오 전송 장치는 상기 인코딩된 픽처에 관한 데이터 및/또는 상기 메타데이터를 기반으로 360 영상/비디오 데이터를 생성할 수 있다. 360 비디오를 구성하는 일련의 영상에 대한 일련의 픽처들을 인코딩한 경우, 상기 인코딩된 픽처들을 포함하는 상기 360 비디오 데이터를 생성할 수 있다. 상기 픽처는 상술한 바와 같이 메인 미디어(백그라운드 미디어)를 포함할 수 있다.
- [0488] 360 비디오 전송 장치는 오버레이 미디어의 종류에 따라 상기 오버레이 미디어를 인코딩하여 출력할 수도 있다. 이 경우 상기 인코딩된 오버레이 미디어는 후술하는 360 영상/비디오 데이터에 포함될 수 있다. 예를 들어, 상기 360 영상/비디오 데이터는 트랙 단위로 상기 메인 미디어 및/또는 상기 오버레이 미디어를 포함할 수 있다.
- [0489] 또는 상기 오버레이 미디어는 360 비디오 수신 장치에 미리 저장될 수도 있고, 360 영상/비디오 데이터와 별도로 네트워크를 통하여 360 비디오 수신 장치로 시그널링될 수도 있다. 또는 상기 오버레이 미디어는 네트워크를 통하여 별도의 엔티티로부터 360 비디오 수신 장치로 시그널링될 수도 있다.
- [0490] 360 비디오 전송 장치는 상기 인코딩된 픽처(들)에 관한 데이터 및/또는 상기 메타데이터를 파일 등의 형태로 인캡슐레이션(encapsulation)할 수 있고, 360 비디오 전송 장치는 인코딩된 360 비디오 데이터 및/또는 상기 메타데이터를 저장 또는 전송하기 위하여 ISOBMFF, CFF 등의 파일 포맷으로 인캡슐레이션하거나, 기타 DASH 세그

먼트 등의 형태로 처리할 수 있다. 360 비디오 전송 장치는 상기 메타데이터를 파일 포맷 상에 포함시킬 수 있다. 예를 들어, 상기 메타데이터는 ISOBMFF 파일 포맷 상의 다양한 레벨의 박스(box)에 포함되거나 파일 내에서 별도의 트랙내의 데이터로 포함될 수 있다.

[0491] 또한, 360 비디오 전송 장치는 상기 메타데이터 자체를 파일로 인캡슐레이션할 수 있다. 360 비디오 전송 장치는 파일 포맷에 따라 인캡슐레이션된 상기 360 비디오 데이터에 전송을 위한 처리를 가할 수 있다. 360 비디오 전송 장치는 임의의 전송 프로토콜에 따라 상기 360 비디오 데이터를 처리할 수 있다. 전송을 위한 처리에는 방송망을 통한 전달을 위한 처리, 또는 브로드밴드 등의 통신 네트워크를 통한 전달을 위한 처리를 포함할 수 있다. 또한, 360 비디오 전송 장치는 상기 메타데이터에 전송을 위한 처리를 가할 수도 있다. 360 비디오 전송 장치는 전송 처리된 상기 360 영상/비디오 데이터(상기 메타데이터 포함)을 방송망 및/또는 브로드밴드를 통해 전송할 수 있다.

[0492] 도 60은 본 발명에 따른 360 비디오 수신 장치에 의한 360 비디오 데이터 처리 방법을 개략적으로 나타낸다. 도 60에서 개시된 방법은 도 6 또는 도 17에서 개시된 360 비디오 수신 장치에 의하여 수행될 수 있다.

[0493] 도 60을 참조하면, 360 비디오 수신 장치는 360 영상/비디오 데이터 (신호)를 수신한다(S6000). 360 비디오 수신 장치는 방송망을 통하여 360 비디오 전송 장치로부터 시그널링된 상기 360 영상/비디오 데이터를 수신할 수 있다. 상기 360 영상/비디오 데이터는 360 영상/비디오의 인코딩된 픽처(들)에 대한 정보 및 상기 메타데이터를 포함할 수 있다. 또한, 360 비디오 수신 장치는 브로드밴드 등의 통신 네트워크, 또는 저장매체를 통하여 360 영상/비디오 데이터를 수신할 수도 있다.

[0494] 360 비디오 수신 장치는 상기 인코딩된 픽처에 대한 정보 및 상기 메타데이터를 획득한다(S6010). 상기 360 영상/비디오 데이터로부터 파일/세그먼트 디캡슐레이션 등의 절차를 통하여 상기 인코딩된 픽처에 대한 정보 및 상기 메타데이터를 획득할 수 있다.

[0495] 상기 메타데이터는 본 명세서에서 기술한 필드들을 포함할 수 있다. 상기 필드들은 다양한 레벨의 박스(box)에 포함되거나 파일 내에서 별도의 트랙내의 데이터로 포함될 수 있다. 예를 들어, 상기 메타데이터는 상기 표 1 내지 29에서 상술한 필드/정보의 일부 또는 전부를 포함할 수 있다. 예를 들어, 상기 메타데이터는 상술한 오버레이 관련 메타데이터(정보/필드 포함)를 포함할 수 있다.

[0496] 예를 들어, 오버레이 관련 메타데이터는 오버레이의 그룹 정보를 포함할 수 있다.

[0497] 여기서, 오버레이의 그룹 정보는 서로 스위칭이 가능한 오버레이들에 대한 정보를 포함할 수 있다. 또한, 서로 스위칭이 가능한 오버레이들에 대한 정보는 ref_overlay_IDs 필드에 의해 지시되는 상기 서로 스위칭이 가능한 오버레이들에 대한 식별 정보를 포함할 수 있다. 여기서, ref_overlay_IDs 필드는 ref_overlay_id 필드라고 지칭할 수도 있으며, 스위칭 가능한 참조 오버레이 ID를 포함할 수 있다.

[0498] 또한, 서로 스위칭 가능한 오버레이들에 대한 정보는 OverlayinteractionStruct에 포함될 수 있으며, 오버레이를 보여주거나 감출 수 있는 인터랙션이 허용되는 경우 포함될 수 있다. 즉, OverlayinteractionStruct 내에 switch_on_off_flag 필드에 의해 인터랙션이 허용되는 경우 포함될 수 있다. 또한, 인터랙션이 허용되는 경우, 스위칭 또는 변경 가능한 레벨의 개수 또는 오버레이의 개수가 지시될 수도 있다. 이는 available_levels 필드에 의해 지시될 수 있으며, 1개 이상인 경우, 상술한 서로 스위칭 가능한 오버레이들에 대한 정보를 포함할 수 있다. 이에 대한 상세한 설명은 표 20과 함께 상술하였다.

[0499] 여기서, 오버레이의 그룹 정보는 오버레이와 함께 렌더링될 픽처를 지시하는 정보를 포함할 수 있다. 또한, 오버레이와 함께 렌더링될 픽처를 지시하는 정보는 EntityToGroupBox 내의 grouping_type 필드에 의해 지시될 수 있다. 즉, grouping_type 필드 값이 ovgr을 가지는 EntityToGroupBox는 메인 VR 미디어 및 오버레이 미디어를 포함하는 트랙 및/또는 아이템 그룹을 지칭할 수 있으며, 그룹 내의 메인 VR 미디어 및 오버레이 미디어는 함께 렌더링될 수 있다. 이는 메인 VR 미디어와 오버레이 미디어가 별도의 트랙에 포함되어 있는 경우에 이용될 수 있다. 즉, 함께 렌더링될 메인 VR 미디어 트랙과 오버레이 미디어 트랙이 링킹(linking) 또는 그룹핑(grouping)될 수 있다. 여기서, 메인 VR 미디어는 또는 VR 미디어 또는 백그라운드(background) 미디어 또는 디코딩된 픽처 또는 디코딩된 픽처의 일부라 지칭할 수도 있다. 이에 대한 상세한 설명은 도 58과 함께 상술하였다.

[0500] 예를 들어, 오버레이 관련 메타데이터는 현재 액티브 상태의 오버레이에 대한 식별 정보를 포함할 수 있다. 여기서, 액티브 상태는 비저블(visible) 상태로 지칭할 수도 있으며, 현재 액티브 상태의 오버레이는 스위칭 가능한 오버레이 중 현재 액티브되는 오버레이를 지칭할 수 있다. 또한, 식별 정보는 인덱스 또는 ID와 같은 고유 정보를 포함할 수 있다. 일 실시예는 식별 정보를 변경함으로써 스위칭 가능한 다른 오버레이를 액티브 상태로

변경할 수 있으며, 스위칭 가능한 오버레이들은 그룹핑되어 있을 수 있다. 또는 'altr'를 통해 그룹핑되어 있는 미디어일 수 있다. 또는 EntityToGroupBox의 grouping_type 필드 값이 altr인 경우 스위칭 가능한 오버레이들의 그룹을 지시할 수 있으며, 현재 액티브 상태의 오버레이는 상기 오버레이들의 그룹에 포함되는 오버레이일 수 있고, 식별 정보 변경에 따라 변경되는 오버레이도 상기 오버레이들의 그룹에 포함될 수 있다.

[0501] 예를 들어, 오버레이 관련 메타데이터는 오버레이에 적용할 알파 컴포지션 타입(alpha composition type)에 대한 정보를 포함할 수 있다. 여기서, 알파 컴포지션은 오버레이 합성 시 오버레이 미디어가 알파 채널을 가지고 있으며 그 알파 값의 합성을 의미할 수 있다. 이는 composition_type 필드에 의해 지시될 수 있다. 또한, 오버레이 관련 메타데이터는 알파 컴포지션 적용 여부에 대한 정보도 포함할 수 있고, alpha_composition_flag 필드에 의해 지시될 수 있다. 여기서, 알파 컴포지션 타입은 알파 컴포지션 모드 또는 컴포지션 타입 또는 컴포지션 모드라 지칭할 수 있으며, 소스 오버(source over)를 포함할 수 있다. 소스 오버는 표 16과 같이 계산될 수 있다. 알파 컴포지션 타입이 소스 오버를 지시하는 경우, composition_type 필드 값이 1일 수 있다. 또한, 알파 컴포지션 타입은 source_atop, source_in, source_out, dest_atop, dest_over, dest_in, dest_out, clear 및 xor 중 하나를 지시할 수도 있으며, 각각은 표 16과 같이 계산될 수 있다.

[0502] 예를 들어, 오버레이 관련 메타데이터는 오버레이에 적용할 블렌딩 모드에 대한 정보를 포함할 수 있다. 또한, 블렌딩 적용 여부에 대한 정보도 포함할 수 있다. 여기서, 블렌딩은 알파 컴포지션보다 복잡한 오퍼레이션을 지칭할 수 있고, 픽셀의 색상을 블렌딩하는 것도 포함할 수 있다. 블렌딩 적용 여부에 대한 정보는 blending_flag 필드에 의해 지시될 수 있으며, 블렌딩 모드에 대한 정보는 blending_mode 필드에 의해 지시될 수 있다.

[0503] 상술한 알파 컴포지션 적용 여부에 대한 정보, 알파 컴포지션 타입에 대한 정보, 블렌딩 적용 여부에 대한 정보 및 블렌딩 모드에 대한 정보 중 적어도 하나는 오버레이 렌더링 관련 정보 또는 오버레이 렌더링 관련 메타데이터에 포함될 수도 있다. 또는 OverlayRenderStruct에 포함될 수 있다. 이에 대한 상세한 설명은 표 15와 함께 상술하였다.

[0504] 예를 들어, 오버레이 관련 메타데이터 중 스태틱(static) 메타데이터는 OverlayConfigBox에 저장될 수 있다. 또는 오버레이 관련 메타데이터 중 타임드(timed) 메타데이터는 샘플에 저장될 수 있다. 여기서, 스태틱 메타데이터는 시간에 따라 변하지 않는 메타데이터를 지칭할 수 있고, 타임드 메타데이터는 시간에 따라 변하는 메타데이터를 지칭할 수 있다.

[0505] 여기서, SchemeTypeBox 내의 scheme_type 필드 값이 podv인 경우, OverlayConfigBox는 ProjectedOmniVideoBox에 포함될 수 있다. 여기서, SchemeTypeBox는 RestrictedSchemeInfoBox에 포함될 수 있다. scheme_type 필드 값이 oldv인 경우 또는 오버레이가 프로젝션되지 않은 비디오인 경우 SchemeInformationBox가 OverlayConfigBox를 포함할 수 있으며, scheme_type 필드 값이 podv인 경우 또는 오버레이가 프로젝션된 비디오인 경우, VR ProjectedOmniVideoBox가 OverlayConfigBox를 포함할 수 있다. 이 경우, VR 미디어 트랙은 ProjectedOmniVideoBox를 포함할 수 있고, ProjectedOmniVideoBox는 OverlayConfigBox를 포함할 수 있다. 여기서, OverlayConfigBox는 오버레이 미디어의 프로젝션 및 패킹 정보를 포함할 수 있다. 또는 오버레이 미디어의 프로젝션 및 패킹 정보를 포함하는 OverlayMediaPackingStruct()를 포함할 수 있다. 이에 대한 상세한 설명은 도 34와 함께 상술하였다.

[0506] 예를 들어, 메타데이터는 서로 스위칭이 가능한 픽처들에 대한 정보를 포함할 수 있다. 이는 grouping_type 필드 값이 altr인 EntityToGroupBox에 의해 지시될 수 있다. 즉, 대체 가능한 메인 VR 미디어들을 명시할 수 있다. 또는 서로 스위칭 가능한 픽처들은 그룹핑될 수 있고, 해당 그룹에 대한 정보를 명시할 수도 있다. 여기서, 픽처는 메인 VR 미디어 또는 메인 미디어 또는 백그라운드 미디어 또는 디코딩된 픽처 또는 디코딩된 픽처의 일부라 지칭할 수도 있다.

[0507] 즉, 일 실시예에 따른 메타데이터는 서로 스위칭 가능한 오버레이들에 대한 정보를 포함할 수 있고, 서로 스위칭 가능한 픽처들(오버레이가 나타날 수 있는 백그라운드 미디어들)에 대한 정보를 포함할 수도 있다.

[0508] 예를 들어, 360 영상 데이터는 복수의 트랙을 포함하고, 메타데이터는 각 트랙이 상기 디코딩된 픽처에 관한 트랙인지를 나타내는 플래그 및 상기 오버레이에 관한 트랙인지를 나타내는 플래그를 포함할 수 있다. 즉, 일 실시예는 메인 미디어 트랙 및 오버레이 미디어 트랙을 그룹핑할 수 있으며, 이는 오버레이 엔티티 그룹이라 지칭할 수 있고, 동일한 group_id 필드 값을 포함할 수 있다. 여기서, 메타데이터는 트랙이 디코딩된 픽처에 대한 트랙인지 또는 메인 미디어 트랙인지에 대한 정보를 포함할 수 있고, 이는 main_media_flag 필드에 의해 지시될 수 있다. 또한, 메타데이터는 트랙이 오버레이에 대한 트랙인지에 대한 정보를 포함할 수 있고, 이는

overlay_media_flag 필드에 의해 지시될 수 있다. 이에 대한 상세한 설명은 표 29와 함께 상술하였다.

- [0509] 360 비디오 수신 장치는 상기 인코딩된 픽처에 대한 정보를 기반으로 픽처(들)를 디코딩한다(S6020). 상기 디코딩된 픽처는 프로젝티브 픽처에 대응할 수 있고, 또는 팩트 픽처(리전별 패킹 과정이 적용된 경우)에 대응할 수도 있다. 상기 디코딩된 픽처는 메인 미디어(백그라운드 미디어)를 포함할 수 있다. 또는 상기 디코딩된 픽처는 오버레이 미디어를 포함할 수도 있다.
- [0510] 360 비디오 수신 장치는 오버레이될 텍스처(미디어)의 종류에 따라 상기 오버레이 텍스처(미디어)를 디코딩할 수도 있다. 이 경우 인코딩된 오버레이 텍스처(미디어)는 상기하는 360 영상/비디오 데이터에 포함될 수 있다.
- [0511] 또는 상기 오버레이 미디어는 360 비디오 수신 장치에 미리 저장될 수도 있고, 360 영상/비디오 데이터와 별도로 네트워크를 통하여 360 비디오 수신 장치로 시그널링될 수도 있다. 또는 상기 오버레이 미디어는 네트워크를 통하여 별도의 엔티티로부터 360 비디오 수신 장치로 시그널링될 수도 있다.
- [0512] 360 비디오 수신 장치는 경우에 따라 상기 메타데이터를 기반으로 상기 픽처를 디코딩할 수도 있다. 이는 예를 들어, 픽처 중 뷰포트가 위치하는 일부 영역에 대한 디코딩을 수행하거나, 뷰포인트 변경이나 오버레이에 링크된 위치의 다른 특정 픽처의 디코딩이 요구되는 경우 등을 포함할 수 있다.
- [0513] 360 비디오 수신 장치는 상기 메타데이터를 기반으로 상기 디코딩된 픽처 및 오버레이를 렌더링한다(S6030). 360 비디오 수신 장치는 상기 메타데이터를 기반으로 상기 디코딩된 픽처 및 오버레이를 처리하여 렌더링할 수 있다. 이 경우 상기 디코딩된 픽처는 상술한 바와 같이 리프로젝션 등의 절차를 거쳐서 3D 서페이스에 렌더링될 수 있다. 상기 오버레이의 경우 상기 메타데이터를 기반으로 상술한 오버레이 타입에 따라 뷰포트 상, 3D 서페이스, 3D 공간 등 위치에 렌더링될 수 있다.
- [0514] 전술한 단계들은 실시예에 따라 생략되거나, 유사/동일한 동작을 수행하는 다른 단계에 의해 대체될 수 있다.
- [0515] 전술한 장치의 내부 컴포넌트들은 메모리에 저장된 연속된 수행과정들을 실행하는 프로세서들이거나, 그 외의 하드웨어로 구성된 하드웨어 컴포넌트들일 수 있다. 이 들은 장치 내/외부에 위치할 수 있다.
- [0516] 전술한 모듈들은 실시예에 따라 생략되거나, 유사/동일한 동작을 수행하는 다른 모듈에 의해 대체될 수 있다.
- [0517] 전술한 각각의 파트, 모듈 또는 유닛은 메모리(또는 저장 유닛)에 저장된 연속된 수행과정들을 실행하는 프로세서이거나 하드웨어 파트일 수 있다. 전술한 실시예에 기술된 각 단계들은 프로세서 또는 하드웨어 파트들에 의해 수행될 수 있다. 전술한 실시예에 기술된 각 모듈/블락/유닛들은 하드웨어/프로세서로서 동작할 수 있다. 또한, 본 발명이 제시하는 방법들은 코드로서 실행될 수 있다. 이 코드는 프로세서가 읽을 수 있는 저장매체에 쓰여질 수 있고, 따라서 장치(apparatus)가 제공하는 프로세서에 의해 읽혀질 수 있다.
- [0518] 상술한 실시예에서, 방법들은 일련의 단계 또는 블록으로써 순서도를 기초로 설명되고 있지만, 본 발명은 단계들의 순서에 한정되는 것은 아니며, 어떤 단계는 상술한 바와 다른 단계와 다른 순서로 또는 동시에 발생할 수 있다. 또한, 당업자라면 순서도에 나타내어진 단계들이 배타적이지 않고, 다른 단계가 포함되거나 순서도의 하나 또는 그 이상의 단계가 본 발명의 범위에 영향을 미치지 않고 삭제될 수 있음을 이해할 수 있을 것이다.
- [0519] 본 발명에서 실시예들이 소프트웨어로 구현될 때, 상술한 방법은 상술한 기능을 수행하는 모듈(과정, 기능 등)로 구현될 수 있다. 모듈은 메모리에 저장되고, 프로세서에 의해 실행될 수 있다. 메모리는 프로세서 내부 또는 외부에 있을 수 있고, 잘 알려진 다양한 수단으로 프로세서와 연결될 수 있다. 프로세서는 ASIC(application-specific integrated circuit), 다른 칩셋, 논리 회로 및/또는 데이터 처리 장치를 포함할 수 있다. 메모리는 ROM(read-only memory), RAM(random access memory), 플래쉬 메모리, 메모리 카드, 저장 매체 및/또는 다른 저장 장치를 포함할 수 있다.
- [0521] 상술한 본 발명의 실시예들은 VR 및 AR에 적용될 수 있다. 상술한 본 발명의 실시예들은 다음과 같은 칩셋 기반으로 구현될 수 있다.
- [0522] 도 61은 본 발명의 실시예들을 지원할 수 있는 장치를 예시적으로 나타낸다. 예를 들어, 상기 제1 장치(first device)는 전송 장치(ex. 360 비디오 전송 장치)를 포함할 수 있고, 상기 제2 장치는 수신 장치(ex. 360 비디오 수신 장치)를 포함할 수 있다. 상술한 전송 장치 및 수신 장치에 대한 본 명세서에서의 기술적 특징이 이 실시예에 적용될 수 있다.
- [0523] 예를 들어, 제1 장치는 프로세서, 메모리, 비디오/이미지 획득 장치 및 송수신부를 포함할 수 있다. 프로세서는 본 명세서에서 설명된 제안된 기능, 절차 및/또는 방법을 수행하도록 구성될 수 있다. 예를 들어, 상기 프로세

서는 상술한 스티칭, 프로젝션, (리전와이즈) 패키징, 컴포지션, (비디오/이미지) 인코딩, 메타데이터 생성 및 처리 등의 절차를 제어 및/또는 수행하도록 구성될 수 있다. 상기 프로세서는 360 비디오/이미지 획득 절차 및 VR/AR 정보(ex. 360 비디오/이미지 데이터 등)의 인캡슐레이션 및 송신 처리를 위한 절차를 제어 및/또는 수행하도록 구성될 수도 있다. 상기 프로세서는 본 발명의 실시예들에서 개시한 메타데이터를 구성 및 전송을 제어할 수 있다. 메모리는 프로세서와 동작 가능하게 결합되고, 프로세서를 동작시키기 위한 다양한 정보를 저장한다. 송수신부는 프로세서와 동작 가능하게 결합되고, 유/무선 신호를 송신 및/또는 수신한다.

[0524] 또한, 예를 들어, 제2 장치는 프로세서, 메모리, 송수신부, 렌더러를 포함할 수 있다. 렌더러는 생략되고 외부 장치로 구현될 수 있다. 프로세서는 본 명세서에서 설명된 제안된 기능, 절차 및/또는 방법을 수행하도록 구성될 수 있다. 예를 들어, 상기 프로세서는 상술한 메타데이터 획득 및 처리, (비디오/이미지) 디코딩, (리전와이즈) 언패킹, 셀렉션, 컴포지션, 리프로젝션, 렌더링 등의 절차를 제어 및/또는 수행하도록 구성될 수 있다. 상기 프로세서는 VR/AR 정보(ex. 360 비디오/이미지 데이터 등)의 디캡슐레이션 및 수신 처리를 위한 절차를 제어 및/또는 수행하도록 구성될 수도 있다. 상기 프로세서는 본 발명의 실시예들에서 개시한 메타데이터를 구성 및 전송을 제어할 수 있다. 메모리는 프로세서와 동작 가능하게 결합되고, 프로세서를 동작시키기 위한 다양한 정보를 저장한다. 송수신부는 프로세서와 동작 가능하게 결합되고, 유/무선 신호를 송신 및/또는 수신한다.

[0525] 본 명세서에서 프로세서는 ASIC(application-specific integrated circuit), 다른 칩셋, 논리 회로 및/또는 데이터 처리 장치를 포함할 수 있다. 메모리는 ROM(read-only memory), RAM(random access memory), 플래시 메모리, 메모리 카드, 저장 매체 및/또는 다른 저장 장치를 포함할 수 있다. 송수신부는 무선 주파수 신호를 처리하기 위한 기저 대역 회로를 포함할 수 있다. 실시예가 소프트웨어로 구현되는 경우, 본 명세서에서 설명된 기술들은 본 명세서에서 설명된 기능을 수행하는 모듈(예컨대, 절차, 기능 등)로 구현될 수 있다. 모듈은 메모리에 저장될 수 있고 프로세서에 의해 실행될 수 있다. 메모리는 프로세서 내부에 구현될 수 있다. 또는, 메모리는 프로세서 외부에 구현될 수 있으며, 기술 분야에서 공지된 다양한 수단을 통해 프로세서에 통신 가능하게 연결될 수 있다.

[0526] 상기 제 1 장치는 기지국, 네트워크 노드, 전송 단말, 수신 단말, 무선 장치, 무선 통신 장치, 차량, 자율주행 기능을 탑재한 차량, 커넥티드카(Connected Car), 드론(Unmanned Aerial Vehicle, UAV), AI(Artificial Intelligence) 모듈, 로봇, AR(Augmented Reality) 장치, VR(Virtual Reality) 장치, MR(Mixed Reality) 장치, 홀로그램 장치, 공공 안전 장치, MTC 장치, IoT 장치, 의료 장치, 핀테크 장치(또는 금융 장치), 보안 장치, 기후/환경 장치, 5G 서비스와 관련된 장치 또는 그 이외 4차 산업 혁명 분야와 관련된 장치일 수 있다.

[0527] 상기 제 2 장치는 기지국, 네트워크 노드, 전송 단말, 수신 단말, 무선 장치, 무선 통신 장치, 차량, 자율주행 기능을 탑재한 차량, 커넥티드카(Connected Car), 드론(Unmanned Aerial Vehicle, UAV), AI(Artificial Intelligence) 모듈, 로봇, AR(Augmented Reality) 장치, VR(Virtual Reality) 장치, MR(Mixed Reality) 장치, 홀로그램 장치, 공공 안전 장치, MTC 장치, IoT 장치, 의료 장치, 핀테크 장치(또는 금융 장치), 보안 장치, 기후/환경 장치, 5G 서비스와 관련된 장치 또는 그 이외 4차 산업 혁명 분야와 관련된 장치일 수 있다.

[0528] 예를 들어, 단말은 휴대폰, 스마트 폰(smart phone), 노트북 컴퓨터(laptop computer), 디지털 방송용 단말기, PDA(personal digital assistants), PMP(portable multimedia player), 네비게이션, 슬레이트 PC(slate PC), 태블릿 PC(tablet PC), 울트라북(ultrabook), 웨어러블 디바이스(wearable device, 예를 들어, 위치형 단말기 (smartwatch), 글래스형 단말기 (smart glass), HMD(head mounted display)) 등을 포함할 수 있다. 예를 들어, HMD는 머리에 착용하는 형태의 디스플레이 장치일 수 있다. 예를 들어, HMD는 VR, AR 또는 MR을 구현하기 위해 사용될 수 있다.

[0529] 예를 들어, 드론은 사람이 타지 않고 무선 컨트롤 신호에 의해 비행하는 비행체일 수 있다. 예를 들어, VR 장치는 가상 세계의 객체 또는 배경 등을 구현하는 장치를 포함할 수 있다. 예를 들어, AR 장치는 현실 세계의 객체 또는 배경 등에 가상 세계의 객체 또는 배경을 연결하여 구현하는 장치를 포함할 수 있다. 예를 들어, MR 장치는 현실 세계의 객체 또는 배경 등에 가상 세계의 객체 또는 배경을 융합하여 구현하는 장치를 포함할 수 있다. 예를 들어, 홀로그램 장치는 홀로그래피라는 두 개의 레이저 광이 만나서 발생하는 빛의 간섭현상을 활용하여, 입체 정보를 기록 및 재생하여 360도 입체 영상을 구현하는 장치를 포함할 수 있다. 예를 들어, 공공 안전 장치는 영상 중계 장치 또는 사용자의 인체에 착용 가능한 영상 장치 등을 포함할 수 있다. 예를 들어, MTC 장치 및 IoT 장치는 사람의 직접적인 개입이나 또는 조작이 필요하지 않는 장치일 수 있다. 예를 들어, MTC 장치 및 IoT 장치는 스마트 미터, 벤딩 머신, 온도계, 스마트 전구, 도어락 또는 각종 센서 등을 포함할 수 있다. 예를 들어, 의료 장치는 질병을 진단, 치료, 경감, 처치 또는 예방할 목적으로 사용되는 장치일 수 있다. 예를 들어,

의료 장치는 상해 또는 장애를 진단, 치료, 경감 또는 보정할 목적으로 사용되는 장치일 수 있다. 예를 들어, 의료 장치는 구조 또는 기능을 검사, 대체 또는 변형할 목적으로 사용되는 장치일 수 있다. 예를 들어, 의료 장치는 임신을 조절할 목적으로 사용되는 장치일 수 있다. 예를 들어, 의료 장치는 진료용 장치, 수술용 장치, (체외) 진단용 장치, 보청기 또는 시술용 장치 등을 포함할 수 있다. 예를 들어, 보안 장치는 발생할 우려가 있는 위험을 방지하고, 안전을 유지하기 위하여 설치한 장치일 수 있다. 예를 들어, 보안 장치는 카메라, CCTV, 녹화기(recorder) 또는 블랙박스 등일 수 있다. 예를 들어, 핀테크 장치는 모바일 결제 등 금융 서비스를 제공할 수 있는 장치일 수 있다. 예를 들어, 핀테크 장치는 결제 장치 또는 POS(Point of Sales) 등을 포함할 수 있다. 예를 들어, 기후/환경 장치는 기후/환경을 모니터링 또는 예측하는 장치를 포함할 수 있다.

[0530] 상기 제 1 장치 및/또는 상기 제 2 장치는 하나 이상의 안테나를 가질 수 있다. 예를 들어, 안테나는 무선 신호를 송수신하도록 구성될 수 있다.

[0531] 상술한 본 발명에 따른 기술적 특징은 VR/AR 등 다양한 서비스에 적용될 수 있다. 또한, 상술한 본 발명에 따른 기술적 특징은 5G(fifth generation) 또는 차세대 통신 등을 통하여 수행될 수 있다. 예를 들어, 송신장치(ex. 360 비디오 전송 장치)에서 출력된 데이터(ex. 비디오/영상 비트스트림, 메타데이터 등 포함)는 상기 5G 통신을 통하여 수신장치(ex. 360 비디오 수신 장치)로 전송될 수 있다. 또한, (VR/AR) 영상/비디오 획득 장치가 외부에 별도로 구비되고, 상기 송신장치로 5G 통신을 통하여 획득된 영상/비디오를 전달할 수 있다. 또한, 본 발명에 따른 송신장치 및/또는 수신장치는 5G 통신을 통하여 다양한 서비스 시나리오를 지원할 수 있다.

[0532] 도 62는 본 발명의 기술적 특징이 적용될 수 있는 5G 사용 시나리오의 예를 나타낸다. 여기서 도시된 5G 사용 시나리오는 단지 예시적인 것이며, 본 발명의 기술적 특징은 도시되지 않은 다른 5G 사용 시나리오에도 적용될 수 있다.

[0533] 도 62를 참조하면, 5G의 세 가지 주요 요구 사항 영역은 (1) 개선된 모바일 광대역(eMBB; enhanced mobile broadband) 영역, (2) 다량의 머신 타입 통신(mMTC; massive machine type communication) 영역 및 (3) 초-신뢰 및 저 지연 통신(URLLC; ultra-reliable and low latency communications) 영역을 포함한다. 일부 사용 예는 최적화를 위해 다수의 영역을 요구할 수 있고, 다른 사용 예는 단지 하나의 핵심 성능 지표(KPI; key performance indicator)에만 포커싱 할 수 있다. 5G는 이러한 다양한 사용 예들을 유연하고 신뢰할 수 있는 방법으로 지원하는 것이다.

[0534] eMBB는 데이터 속도, 지연, 사용자 밀도, 모바일 광대역 접속의 용량 및 커버리지의 전반적인 향상에 중점을 둔다. eMBB는 10Gbps 정도의 처리량을 목표로 한다. eMBB는 기본적인 모바일 인터넷 접속을 훨씬 능가하게 하며, 풍부한 양방향 작업, 클라우드 또는 증강 현실에서 미디어 및 엔터테인먼트 애플리케이션을 커버한다. 데이터는 5G의 핵심 동력 중 하나이며, 5G 시대에 처음으로 전용 음성 서비스를 볼 수 없을 수 있다. 5G에서, 음성은 단순히 통신 시스템에 의해 제공되는 데이터 연결을 사용하여 응용 프로그램으로서 처리될 것으로 기대된다. 증가된 트래픽 양의 주요 원인은 콘텐츠 크기의 증가 및 높은 데이터 전송률을 요구하는 애플리케이션 수의 증가이다. 스트리밍 서비스(오디오 및 비디오), 대화형 비디오 및 모바일 인터넷 연결은 더 많은 장치가 인터넷에 연결될수록 더 널리 사용될 것이다. 이러한 많은 애플리케이션은 사용자에게 실시간 정보 및 알림을 푸시하기 위해 항상 켜져 있는 연결성을 필요로 한다. 클라우드 스토리지 및 애플리케이션은 모바일 통신 플랫폼에서 급속히 증가하고 있으며, 이것은 업무 및 엔터테인먼트 모두에 적용될 수 있다. 클라우드 스토리지는 상향링크 데이터 전송률의 성장을 견인하는 특별한 사용 예이다. 5G는 또한 클라우드 상의 원격 업무에도 사용되며, 촉각 인터페이스가 사용될 때 우수한 사용자 경험을 유지하도록 훨씬 더 낮은 단-대-단(end-to-end) 지연을 요구한다. 엔터테인먼트에서 예를 들면, 클라우드 게임 및 비디오 스트리밍은 모바일 광대역 능력에 대한 요구를 증가시키는 또 다른 핵심 요소이다. 엔터테인먼트는 기차, 차 및 비행기와 같은 높은 이동성 환경을 포함하여 어떤 곳에서든지 스마트폰 및 태블릿에서 필수적이다. 또 다른 사용 예는 엔터테인먼트를 위한 증강 현실 및 정보 검색이다. 여기서, 증강 현실은 매우 낮은 지연과 순간적인 데이터 양을 필요로 한다.

[0535] mMTC는 배터리에 의해 구동되는 다량의 저비용 장치 간의 통신을 가능하게 하기 위하여 설계되며, 스마트 계량, 물류, 현장 및 신체 센서와 같은 애플리케이션을 지원하기 위한 것이다. mMTC는 10년 정도의 배터리 및/또는 1km² 당 백만 개 정도의 장치를 목표로 한다. mMTC는 모든 분야에서 임베디드 센서를 원활하게 연결할 수 있게 하며, 가장 많이 예상되는 5G 사용 예 중 하나이다. 잠재적으로 2020년까지 IoT 장치들은 204억 개에 이를 것으로 예측된다. 산업 IoT는 5G가 스마트 도시, 자산 추적(asset tracking), 스마트 유틸리티, 농업 및 보안 인프라를 가능하게 하는 주요 역할을 수행하는 영역 중 하나이다.

[0536] URLLC는 장치 및 기계가 매우 신뢰성 있고 매우 낮은 지연 및 높은 가용성으로 통신할 수 있도록 함으로써 차량

통신, 산업 제어, 공장 자동화, 원격 수술, 스마트 그리드 및 공공 안전 애플리케이션에 이상적이다. URLLC는 1ms의 정도의 지연을 목표로 한다. URLLC는 주요 인프라의 원격 제어 및 자율 주행 차량과 같은 초 신뢰/지연이 적은 링크를 통해 산업을 변화시킬 새로운 서비스를 포함한다. 신뢰성과 지연의 수준은 스마트 그리드 제어, 산업 자동화, 로봇 공학, 드론 제어 및 조정에 필수적이다.

- [0537] 다음으로, 도 62의 삼각형 안에 포함된 다수의 사용 예에 대해 보다 구체적으로 살펴본다.
- [0538] 5G는 초당 수백 메가 비트에서 초당 기가 비트로 평가되는 스트림을 제공하는 수단으로 FTTH(fiber-to-the-home) 및 케이블 기반 광대역(또는 DOCSIS)을 보완할 수 있다. 이러한 빠른 속도는 가상 현실(VR; virtual reality)과 증강 현실(AR; augmented reality) 뿐 아니라 4K 이상(6K, 8K 및 그 이상)의 해상도로 TV를 전달하는 데에 요구될 수 있다. VR 및 AR 애플리케이션은 거의 몰입형(immersive) 스포츠 경기를 포함한다. 특정 애플리케이션은 특별한 네트워크 설정이 요구될 수 있다. 예를 들어, VR 게임의 경우, 게임 회사가 지연을 최소화하기 위해 코어 서버를 네트워크 오퍼레이터의 에지 네트워크 서버와 통합해야 할 수 있다.
- [0539] 자동차(Automotive)는 차량에 대한 이동 통신을 위한 많은 사용 예와 함께 5G에 있어 중요한 새로운 동력이 될 것으로 예상된다. 예를 들어, 승객을 위한 엔터테인먼트는 높은 용량과 높은 모바일 광대역을 동시에 요구한다. 그 이유는 미래의 사용자는 그들의 위치 및 속도와 관계 없이 고품질의 연결을 계속해서 기대하기 때문이다. 자동차 분야의 다른 사용 예는 증강 현실 대시보드이다. 운전자는 증강 현실 대시보드를 통해 앞면 창을 통해 보고 있는 것 위에 어둠 속에서 물체를 식별할 수 있다. 증강 현실 대시보드는 물체의 거리와 움직임에 대해 운전자에게 알려줄 정보를 겹쳐서 디스플레이 한다. 미래에, 무선 모듈은 차량 간의 통신, 차량과 지원하는 인프라 구조 사이에서 정보 교환 및 자동차와 다른 연결된 장치(예를 들어, 보행자에 의해 수반되는 장치) 사이에서 정보 교환을 가능하게 한다. 안전 시스템은 운전자가 보다 안전한 운전을 할 수 있도록 행동의 대체 코스를 안내하여 사고의 위험을 낮출 수 있게 한다. 다음 단계는 원격 조종 차량 또는 자율 주행 차량이 될 것이다. 이는 서로 다른 자율 주행 차량 사이 및/또는 자동차와 인프라 사이에서 매우 신뢰성이 있고 매우 빠른 통신을 요구한다. 미래에, 자율 주행 차량이 모든 운전 활동을 수행하고, 운전자는 차량 자체가 식별할 수 없는 교통 이상에만 집중하도록 할 것이다. 자율 주행 차량의 기술적 요구 사항은 트래픽 안전을 사람이 달성할 수 없을 정도의 수준까지 증가하도록 초 저 지연과 초고속 신뢰성을 요구한다.
- [0540] 스마트 사회로서 언급되는 스마트 도시와 스마트 홈은 고밀도 무선 센서 네트워크로 임베디드 될 것이다. 지능형 센서의 분산 네트워크는 도시 또는 집의 비용 및 에너지 효율적인 유지에 대한 조건을 식별할 것이다. 유사한 설정이 각 가정을 위해 수행될 수 있다. 온도 센서, 창 및 난방 컨트롤러, 도난 경보기 및 가전 제품은 모두 무선으로 연결된다. 이러한 센서 중 많은 것들이 전형적으로 낮은 데이터 전송 속도, 저전력 및 저비용을 요구한다. 하지만, 예를 들어, 실시간 HD 비디오는 감시를 위해 특정 타입의 장치에서 요구될 수 있다.
- [0541] 열 또는 가스를 포함한 에너지의 소비 및 분배는 고도로 분산화되고 있어, 분산 센서 네트워크의 자동화된 제어가 요구된다. 스마트 그리드는 정보를 수집하고 이에 따라 행동하도록 디지털 정보 및 통신 기술을 사용하여 이런 센서를 상호 연결한다. 이 정보는 공급 업체와 소비자의 행동을 포함할 수 있으므로, 스마트 그리드가 효율성, 신뢰성, 경제성, 생산의 지속 가능성 및 자동화된 방식으로 전기와 같은 연료의 분배를 개선하도록 할 수 있다. 스마트 그리드는 지연이 적은 다른 센서 네트워크로 볼 수도 있다.
- [0542] 건강 부문은 이동 통신의 혜택을 누릴 수 있는 많은 애플리케이션을 보유하고 있다. 통신 시스템은 멀리 떨어진 곳에서 임상 진료를 제공하는 원격 진료를 지원할 수 있다. 이는 거리에 대한 장벽을 줄이는 데에 도움을 주고, 거리가 먼 농촌에서 지속적으로 이용하지 못하는 의료 서비스로의 접근을 개선시킬 수 있다. 이는 또한 중요한 진료 및 응급 상황에서 생명을 구하기 위해 사용된다. 이동 통신 기반의 무선 센서 네트워크는 심박수 및 혈압과 같은 파라미터에 대한 원격 모니터링 및 센서를 제공할 수 있다.
- [0543] 무선 및 모바일 통신은 산업 응용 분야에서 점차 중요해지고 있다. 배선은 설치 및 유지 비용이 높다. 따라서, 케이블을 재구성할 수 있는 무선 링크로의 교체 가능성은 많은 산업 분야에서 매력적인 기회이다. 그러나, 이를 달성하는 것은 무선 연결이 케이블과 비슷한 지연, 신뢰성 및 용량으로 동작하는 것과, 그 관리가 단순화될 것을 요구한다. 낮은 지연과 매우 낮은 오류 확률은 5G로 연결될 필요가 있는 새로운 요구 사항이다.
- [0544] 물류 및 화물 추적은 위치 기반 정보 시스템을 사용하여 어디에서든지 인벤토리(inventory) 및 패키지의 추적을 가능하게 하는 이동 통신에 대한 중요한 사용 예이다. 물류 및 화물 추적의 사용 예는 전형적으로 낮은 데이터 속도를 요구하지만 넓은 범위와 신뢰성 있는 위치 정보가 필요하다.
- [0545] 또한, 본 발명에 따른 실시예들은 확장 현실(XR: eXtended Reality)을 지원하기 위하여 수행될 수 있다. 확장

현실은 가상 현실(VR: Virtual Reality), 증강 현실(AR: Augmented Reality), 혼합 현실(MR: Mixed Reality)을 총칭한다. VR 기술은 현실 세계의 객체나 배경 등을 CG 영상으로만 제공하고, AR 기술은 실제 사물 영상 위에 가상으로 만들어진 CG 영상을 함께 제공하며, MR 기술은 현실 세계에 가상 객체들을 섞고 결합시켜서 제공하는 컴퓨터 그래픽 기술이다.

- [0546] MR 기술은 현실 객체와 가상 객체를 함께 보여준다는 점에서 AR 기술과 유사하다. 그러나, AR 기술에서는 가상 객체가 현실 객체를 보완하는 형태로 사용되는 반면, MR 기술에서는 가상 객체와 현실 객체가 동등한 성격으로 사용된다는 점에서 차이점이 있다.
- [0547] XR 기술은 HMD(Head-Mount Display), HUD(Head-Up Display), 휴대폰, 태블릿 PC, 랩탑, 데스크탑, TV, 디지털 사이니지 등에 적용될 수 있고, XR 기술이 적용된 장치를 XR 장치(XR Device)라 칭할 수 있다. XR 장치는 상술한 제1 장치 및/또는 제2 장치를 포함할 수 있다.
- [0548] XR 장치는 5G 통신 등에 기반한 통신 네트워크를 통하여 다양한 서비스에 연결될 수 있다.
- [0549] 도 63은 본 발명의 일 실시 예에 따른 서비스 시스템을 나타낸다.
- [0550] 도 63을 참조하면, XR 장치(100c)는 네트워크(10)를 통하여 AI 서버(200a), 로봇(100a), 자율 주행 차량(100b), 스마트폰(100d) 또는 가전(100e) 중에서 적어도 하나 이상과 연결될 수 있다. 여기서, AI 기술이 적용된 로봇(100a), 자율 주행 차량(100b), XR 장치(100c), 스마트폰(100d) 또는 가전(100e) 등을 AI 장치라 칭할 수 있다.
- [0551] 네트워크(10)은 유/무선 통신 네트워크를 포함할 수 있다. 네트워크(10)는 클라우드 네트워크를 포함할 수 있다. 클라우드 네트워크는 클라우드 컴퓨팅 인프라의 일부를 구성하거나 클라우드 컴퓨팅 인프라 안에 존재하는 네트워크를 의미할 수 있다. 여기서, 클라우드 네트워크는 3G 네트워크, 4G 또는 LTE(Long Term Evolution) 네트워크 또는 5G 네트워크 등을 이용하여 구성될 수 있다.
- [0552] 상기 시스템(1)을 구성하는 각 장치들(100a 내지 100e, 200a)은 클라우드 네트워크(10)를 통해 서로 연결될 수 있다. 특히, 각 장치들(100a 내지 100e, 200a)은 기지국을 통해서 서로 통신할 수도 있지만, 기지국을 통하지 않고 직접 서로 통신할 수도 있다.
- [0553] AI 서버(200a)는 AI 프로세싱을 수행하는 서버와 빅 데이터에 대한 연산을 수행하는 서버를 포함할 수 있다.
- [0554] AI 서버(200a)는 로봇(100a), 자율 주행 차량(100b), XR 장치(100c), 스마트폰(100d) 또는 가전(100e) 중에서 적어도 하나 이상과 네트워크(10)를 통하여 연결되고, 연결된 AI 장치들(100a 내지 100e)의 AI 프로세싱을 적어도 일부를 도울 수 있다.
- [0555] 이때, AI 서버(200a)는 AI 장치(100a 내지 100e)를 대신하여 머신 러닝 알고리즘에 따라 인공 신경망을 학습시킬 수 있고, 학습 모델을 직접 저장하거나 AI 장치(100a 내지 100e)에 전송할 수 있다.
- [0556] 이때, AI 서버(200a)는 AI 장치(100a 내지 100e)로부터 입력 데이터를 수신하고, 학습 모델을 이용하여 수신한 입력 데이터에 대하여 결과 값을 추론하고, 추론한 결과 값에 기초한 응답이나 제어 명령을 생성하여 AI 장치(100a 내지 100e)로 전송할 수 있다.
- [0557] 또는, AI 장치(100a 내지 100e)는 직접 학습 모델을 이용하여 입력 데이터에 대하여 결과 값을 추론하고, 추론한 결과 값에 기초한 응답이나 제어 명령을 생성할 수도 있다.
- [0558] XR 장치(100c)는 HMD(Head-Mount Display), 차량에 구비된 HUD(Head-Up Display), 텔레비전, 휴대폰, 스마트폰, 컴퓨터, 웨어러블 디바이스, 가전 기기, 디지털 사이니지, 차량, 고정형 로봇이나 이동형 로봇 등으로 구현될 수 있다.
- [0559] XR 장치(100c)는 다양한 센서들을 통해 또는 외부 장치로부터 획득한 3차원 포인트 클라우드 데이터 또는 이미지 데이터를 분석하여 3차원 포인트들에 대한 위치 데이터 및 속성 데이터를 생성함으로써 주변 공간 또는 현실 객체에 대한 정보를 획득하고, 출력할 XR 객체를 렌더링하여 출력할 수 있다. 예컨대, XR 장치는 인식된 물체에 대한 추가 정보를 포함하는 XR 객체를 해당 인식된 물체에 대응시켜 출력할 수 있다.
- [0560] XR 장치(100c)는 적어도 하나 이상의 인공 신경망으로 구성된 학습 모델을 이용하여 상기한 동작들을 수행할 수 있다. 예컨대, XR 장치(100c)는 학습 모델을 이용하여 3차원 포인트 클라우드 데이터 또는 이미지 데이터에서 현실 객체를 인식할 수 있고, 인식한 현실 객체에 상응하는 정보를 제공할 수 있다. 여기서, 학습 모델은 XR 장

치(100c)에서 직접 학습되거나, AI 서버(200a) 등의 외부 장치에서 학습된 것일 수 있다.

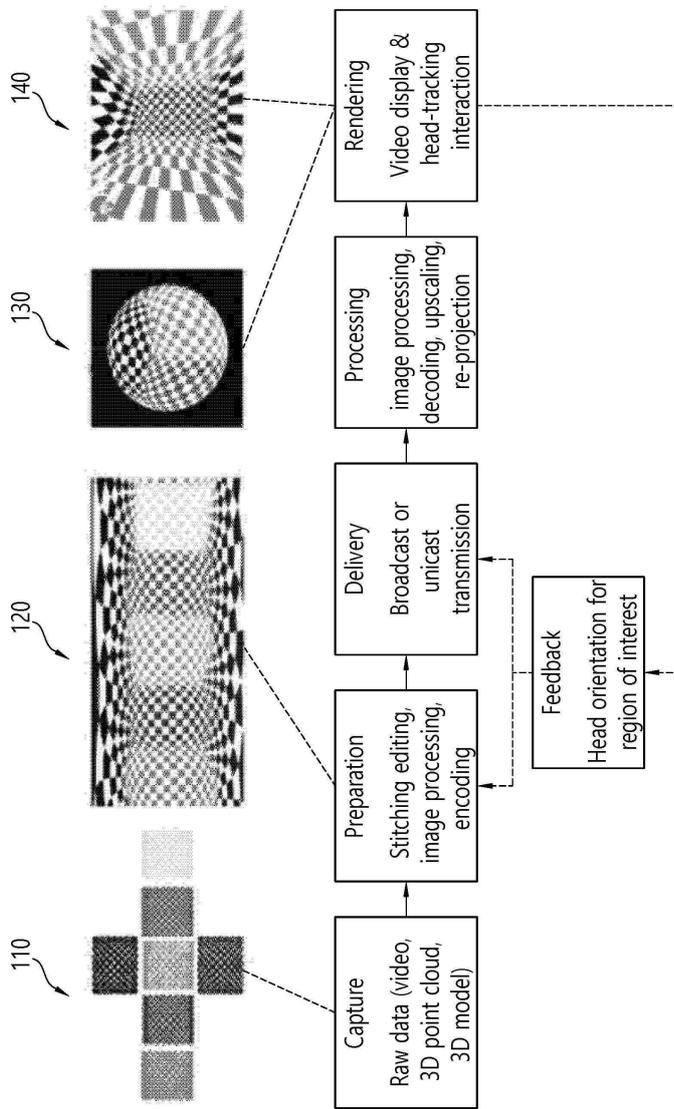
- [0561] 이때, XR 장치(100c)는 직접 학습 모델을 이용하여 결과를 생성하여 동작을 수행할 수도 있지만, AI 서버(200a) 등의 외부 장치에 센서 정보를 전송하고 그에 따라 생성된 결과를 수신하여 동작을 수행할 수도 있다.
- [0562] 로봇(100a)은 안내 로봇, 운반 로봇, 청소 로봇, 웨어러블 로봇, 엔터테인먼트 로봇, 펫 로봇, 무인 비행 로봇, 드론 등을 포함할 수 있다.
- [0563] 로봇(100a)은 동작을 제어하기 위한 로봇 제어 모듈을 포함할 수 있고, 로봇 제어 모듈은 소프트웨어 모듈 또는 이를 하드웨어로 구현한 칩을 의미할 수 있다.
- [0564] 로봇(100a)은 다양한 종류의 센서들로부터 획득한 센서 정보를 이용하여 로봇(100a)의 상태 정보를 획득하거나, 주변 환경 및 객체를 검출(인식)하거나, 맵 데이터를 생성하거나, 이동 경로 및 주행 계획을 결정하거나, 사용자 상호작용에 대한 응답을 결정하거나, 동작을 결정할 수 있다.
- [0565] 여기서, 로봇(100a)은 이동 경로 및 주행 계획을 결정하기 위하여, 라이더, 레이더, 카메라 중에서 적어도 하나 이상의 센서에서 획득한 센서 정보를 이용할 수 있다.
- [0566] XR 장치(100c)는 네트워크(10)를 통하여 로봇(100a)을 원격접속 및/또는 원격조정할 수도 있다. 이 경우, 로봇(100a)은 XR 장치(100c)를 사용하는 사용자와 시야 또는 화면을 공유하고, 상기 사용자의 제어/상호작용에 기초하여 구동부를 제어함으로써, 동작을 수행하거나 주행할 수 있다. 이때, 로봇(100a)은 사용자의 동작이나 음성 발화에 따른 상호작용의 의도 정보를 획득하고, 획득한 의도 정보에 기초하여 응답을 결정하여 동작을 수행할 수 있다.
- [0567] XR 기술이 적용된 로봇(100a)은 XR 영상 내에서의 제어/상호작용의 대상이 되는 로봇을 의미할 수 있다. 이 경우, 로봇(100a)은 XR 장치(100c)와 구분되며 서로 연동될 수 있다. 이 경우, 로봇(100a)은 XR 장치(100c)와 구분되며 서로 연동될 수 있다. XR 영상 내에서의 제어/상호작용의 대상이 되는 로봇(100a)은 카메라를 포함하는 센서들로부터 센서 정보를 획득하면, 로봇(100a) 또는 XR 장치(100c)는 센서 정보에 기초한 XR 영상을 생성하고, XR 장치(100c)는 생성된 XR 영상을 출력할 수 있다. 그리고, 이러한 로봇(100a)은 XR 장치(100c)를 통해 입력되는 제어 신호 또는 사용자의 상호작용에 기초하여 동작할 수 있다.
- [0568] 예컨대, 사용자는 XR 장치(100c) 등의 외부 장치를 통해 원격으로 연동된 로봇(100a)의 시점에 상응하는 XR 영상을 확인할 수 있고, 상호작용을 통하여 로봇(100a)의 자율 주행 경로를 조정하거나, 동작 또는 주행을 제어하거나, 주변 객체의 정보를 확인할 수 있다.
- [0569] 자율 주행 차량(100b)은 이동형 로봇, 차량, 기차, 유/무인 비행체, 선박 등을 포함할 수 있다.
- [0570] 자율 주행 차량(100b)은 자율 주행 기능을 제어하기 위한 자율 주행 제어 모듈을 포함할 수 있고, 자율 주행 제어 모듈은 소프트웨어 모듈 또는 이를 하드웨어로 구현한 칩을 의미할 수 있다. 자율 주행 제어 모듈은 자율 주행 차량(100b)의 구성으로써 내부에 포함될 수도 있지만, 자율 주행 차량(100b)의 외부에 별도의 하드웨어로 구성되어 연결될 수도 있다.
- [0571] 자율 주행 차량(100b)은 다양한 종류의 센서들로부터 획득한 센서 정보를 이용하여 자율 주행 차량(100b)의 상태 정보를 획득하거나, 주변 환경 및 객체를 검출(인식)하거나, 맵 데이터를 생성하거나, 이동 경로 및 주행 계획을 결정하거나, 동작을 결정할 수 있다.
- [0572] 여기서, 자율 주행 차량(100b)은 이동 경로 및 주행 계획을 결정하기 위하여, 로봇(100a)과 마찬가지로, 라이더, 레이더, 카메라 중에서 적어도 하나 이상의 센서에서 획득한 센서 정보를 이용할 수 있다.
- [0573] 특히, 자율 주행 차량(100b)은 시야가 가려지는 영역이나 일정 거리 이상의 영역에 대한 환경이나 객체는 외부 장치들로부터 센서 정보를 수신하여 인식하거나, 외부 장치들로부터 직접 인식된 정보를 수신할 수 있다.
- [0574] XR 장치(100c)는 네트워크(10)를 통하여 자율 주행 차량(100b)을 원격접속 및/또는 원격조정할 수도 있다. 이 경우, 자율 주행 차량(100b)은 XR 장치(100c)를 사용하는 사용자와 시야 또는 화면을 공유하고, 상기 사용자의 제어/상호작용에 기초하여 구동부를 제어함으로써, 동작을 수행하거나 주행할 수 있다. 이때, 로봇(100a)은 사용자의 동작이나 음성 발화에 따른 상호작용의 의도 정보를 획득하고, 획득한 의도 정보에 기초하여 응답을 결정하여 동작을 수행할 수 있다.
- [0575] XR 기술이 적용된 자율 주행 차량(100b)은 XR 영상을 제공하는 수단을 구비한 자율 주행 차량이나, XR 영상 내

에서의 제어/상호작용의 대상이 되는 자율 주행 차량 등을 의미할 수 있다. 특히, XR 영상 내에서의 제어/상호작용의 대상이 되는 자율 주행 차량(100b)은 XR 장치(100c)와 구분되며 서로 연동될 수 있다.

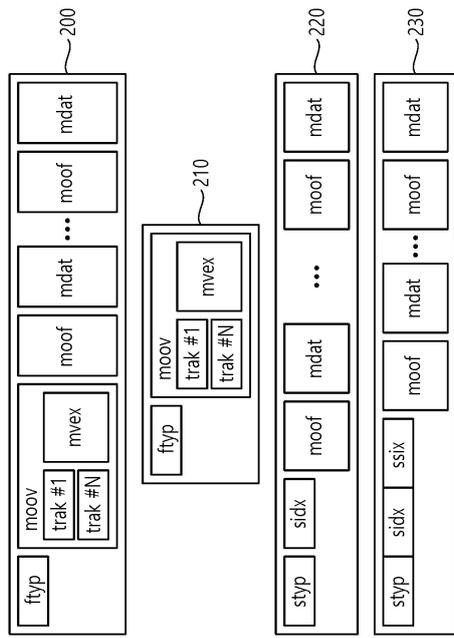
- [0576] XR 영상을 제공하는 수단을 구비한 자율 주행 차량(100b)은 카메라를 포함하는 센서들로부터 센서 정보를 획득하고, 획득한 센서 정보에 기초하여 생성된 XR 영상을 출력할 수 있다. 예컨대, 자율 주행 차량(100b)은 HUD를 구비하여 XR 영상을 출력함으로써, 탑승자에게 현실 객체 또는 화면 속의 객체에 대응되는 XR 객체를 제공할 수 있다.
- [0577] 이때, XR 객체가 HUD에 출력되는 경우에는 XR 객체의 적어도 일부가 탑승자의 시선이 향하는 실제 객체에 오버랩되도록 출력될 수 있다. 반면, XR 객체가 자율 주행 차량(100b)의 내부에 구비되는 디스플레이에 출력되는 경우에는 XR 객체의 적어도 일부가 화면 속의 객체에 오버랩되도록 출력될 수 있다. 예컨대, 자율 주행 차량(100b)은 차로, 타 차량, 신호등, 교통 표지판, 이륜차, 보행자, 건물 등과 같은 객체와 대응되는 XR 객체들을 출력할 수 있다.
- [0578] XR 영상 내에서의 제어/상호작용의 대상이 되는 자율 주행 차량(100b)은 카메라를 포함하는 센서들로부터 센서 정보를 획득하면, 자율 주행 차량(100b) 또는 XR 장치(100c)는 센서 정보에 기초한 XR 영상을 생성하고, XR 장치(100c)는 생성된 XR 영상을 출력할 수 있다. 그리고, 이러한 자율 주행 차량(100b)은 XR 장치(100c) 등의 외부 장치를 통해 입력되는 제어 신호 또는 사용자의 상호작용에 기초하여 동작할 수 있다.
- [0579] XR 장치(100c)는 로봇(100a) 및/또는 자율 주행 차량(100b) 내부에 구비되어 사용자에게 별도의 XR 콘텐츠를 제공할 수도 있고, 또는 로봇(100a) 및/또는 자율 주행 차량(100b) 내/외부의 영상을 사용자에게 제공할 수도 있다.
- [0580] XR 장치(100c)는 그 외에도 엔터테인먼트, 운동, 교육, 교통, 의료, 전자상거래, 제조, 국방 등 다양한 서비스에 사용될 수 있다. 예를 들어, 영화, 테마파크, 스포츠 등을 XR 장치(100c)를 통하여 체험 및/또는 관람할 수 있고, 의료용 실습, 화재 현장 등 위험한 환경에서의 훈련 등을 지원할 수 있다. 또한, XR 장치(100c)를 통하여 위치인식 및 지도 생성(SLAM) 기술을 활용한 AR 웨이즈(AR Ways) 등 길찾기 서비스를 제공할 수 있고, 또한, 가상의 쇼핑_물에 접속하여 물건을 쇼핑하고 구매할 수도 있다.

도면

도면1



도면2



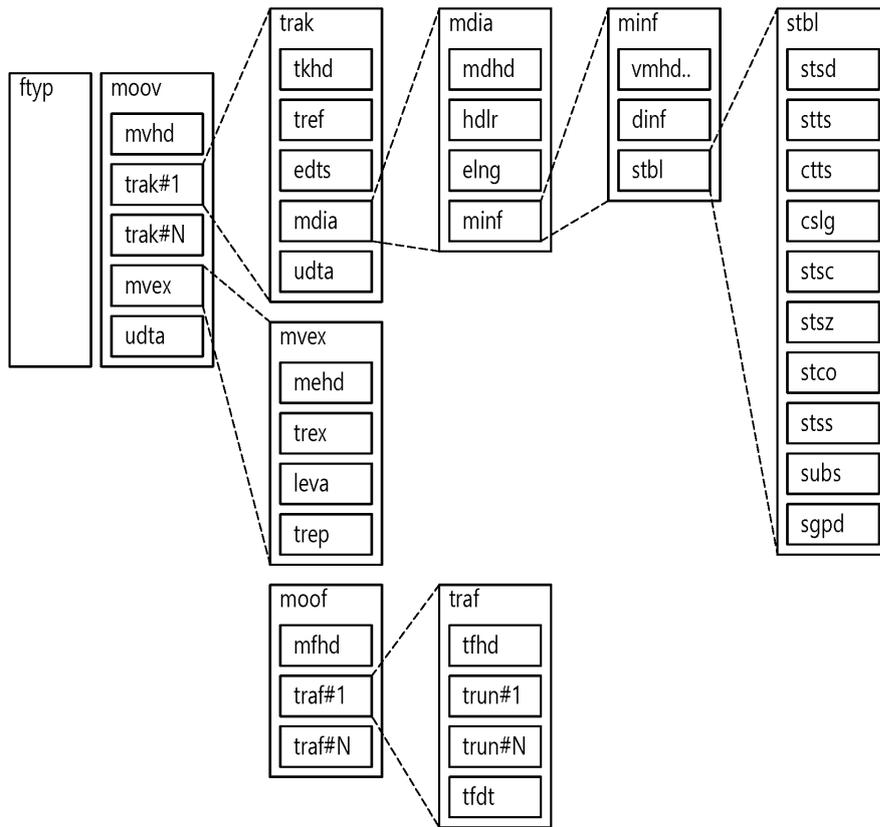
250

```

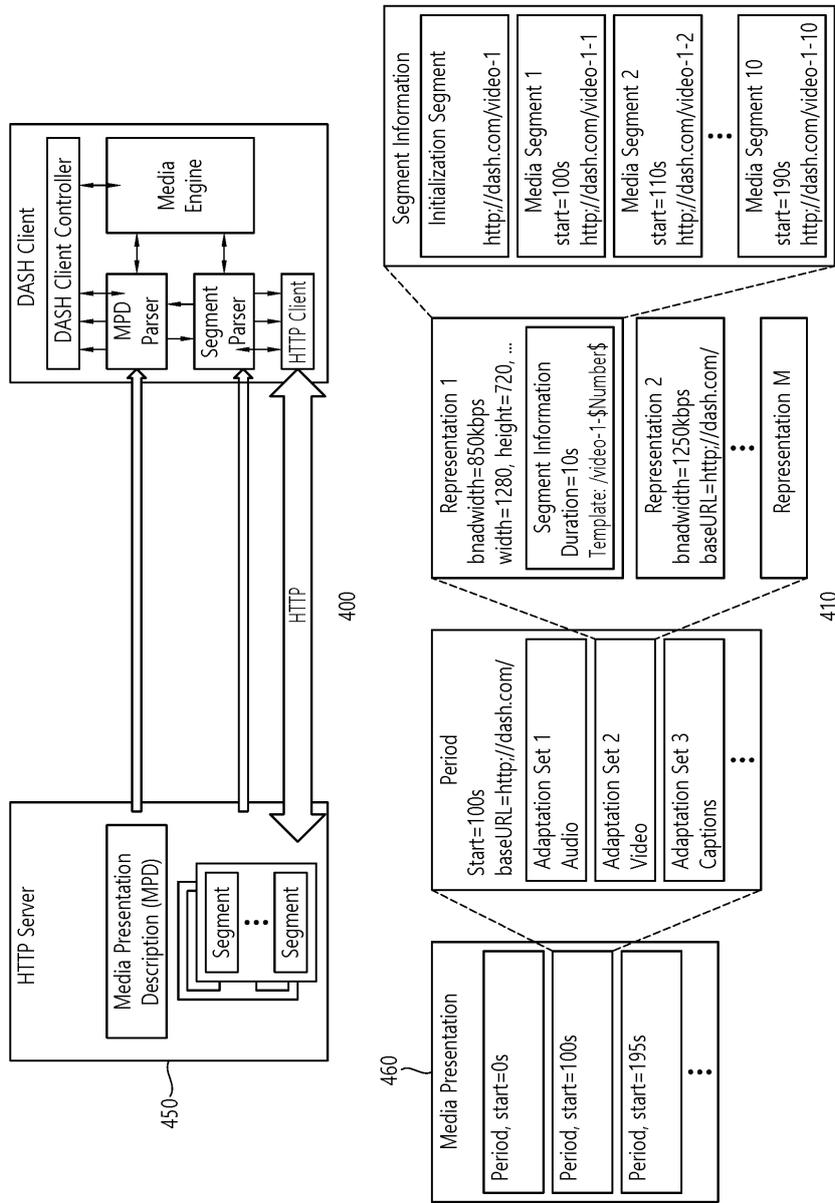
aligned(8), class Box (unsigned int(32) boxtype, optional
unsigned int(8)[16] extended_type)
{
    unsigned int(32) size;
    unsigned int(32) type = boxtype;
    if (size==1) {
        unsigned int(64) largesize;
    } else if (size==0) {
        // box extends to end of file
    }
    if (boxtype=='uuid') {
        unsigned int(8)[16] usertype = extended_type;
    }
}

aligned(8), class FullBox(unsigned int(32) boxtype, unsigned int(8)
v, bit(24) f) extends Box(boxtype) {
    unsigned int(8) version = v;
    bit(24) flags = f;
}
    
```

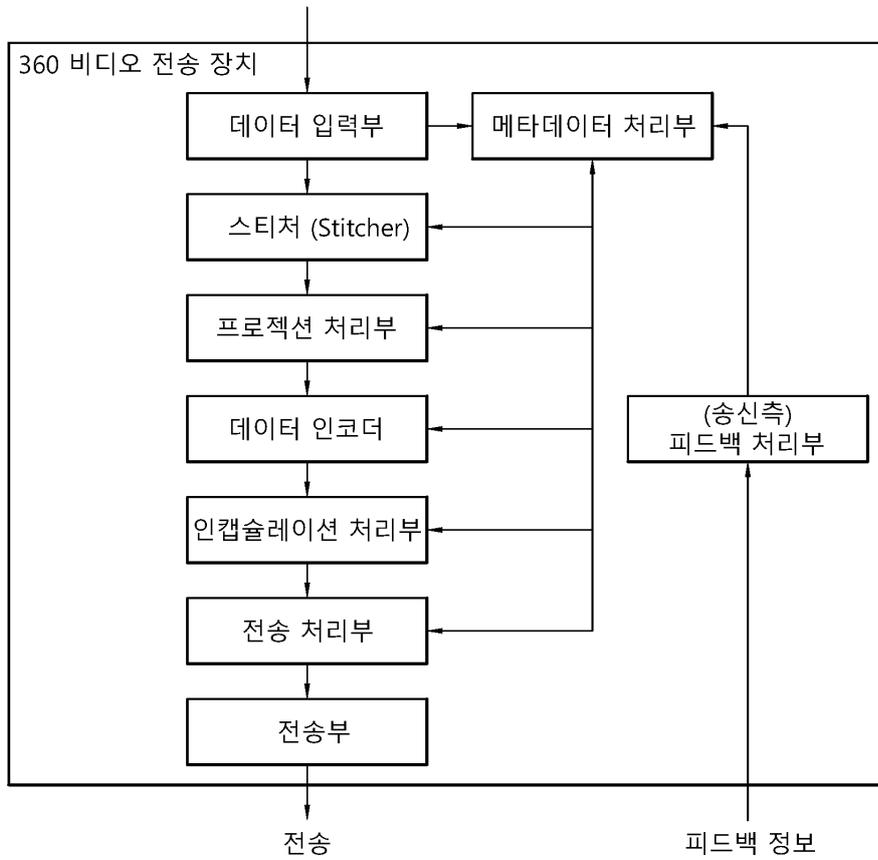
도면3



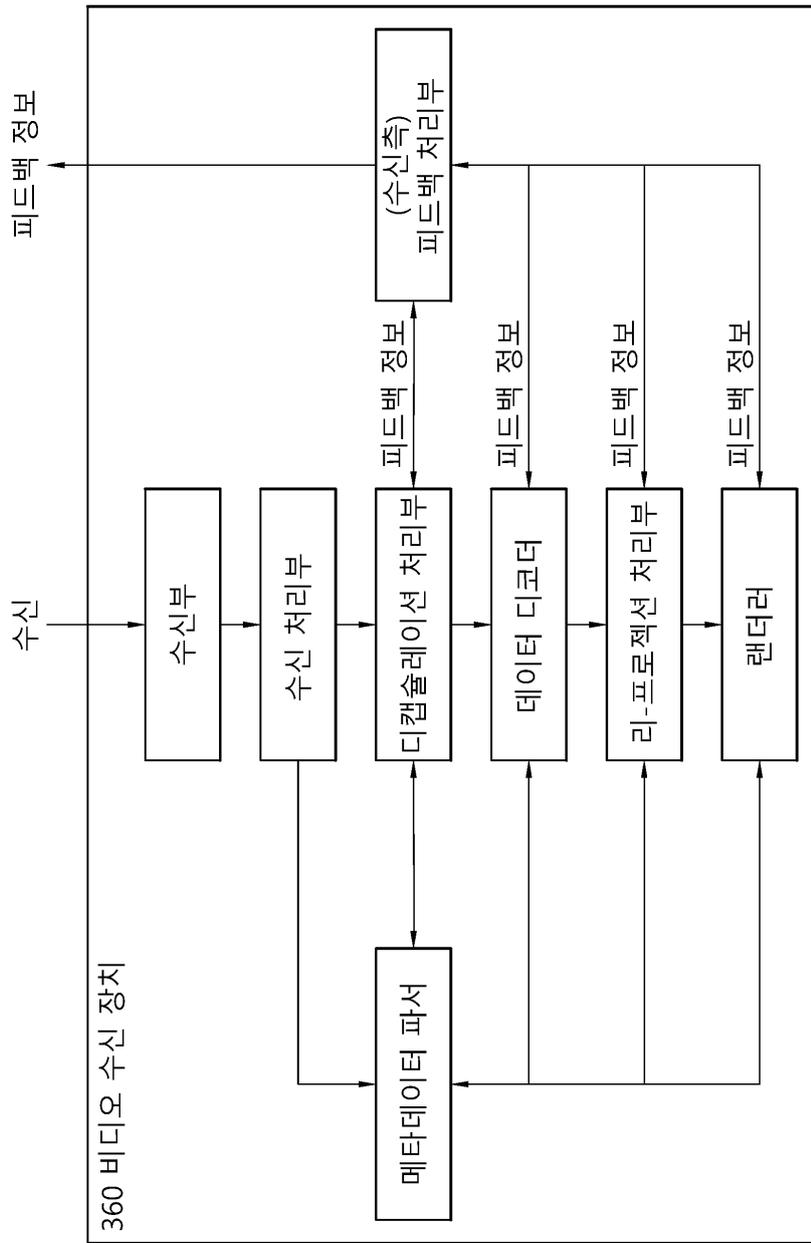
도면4



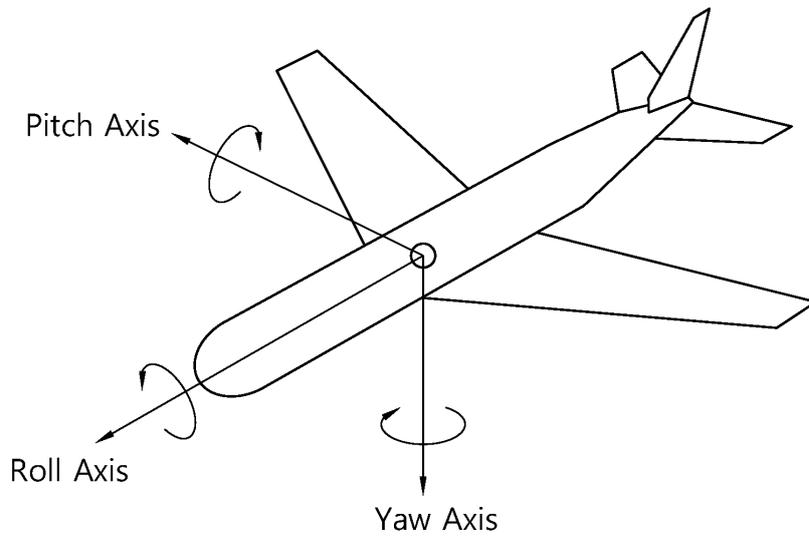
도면5



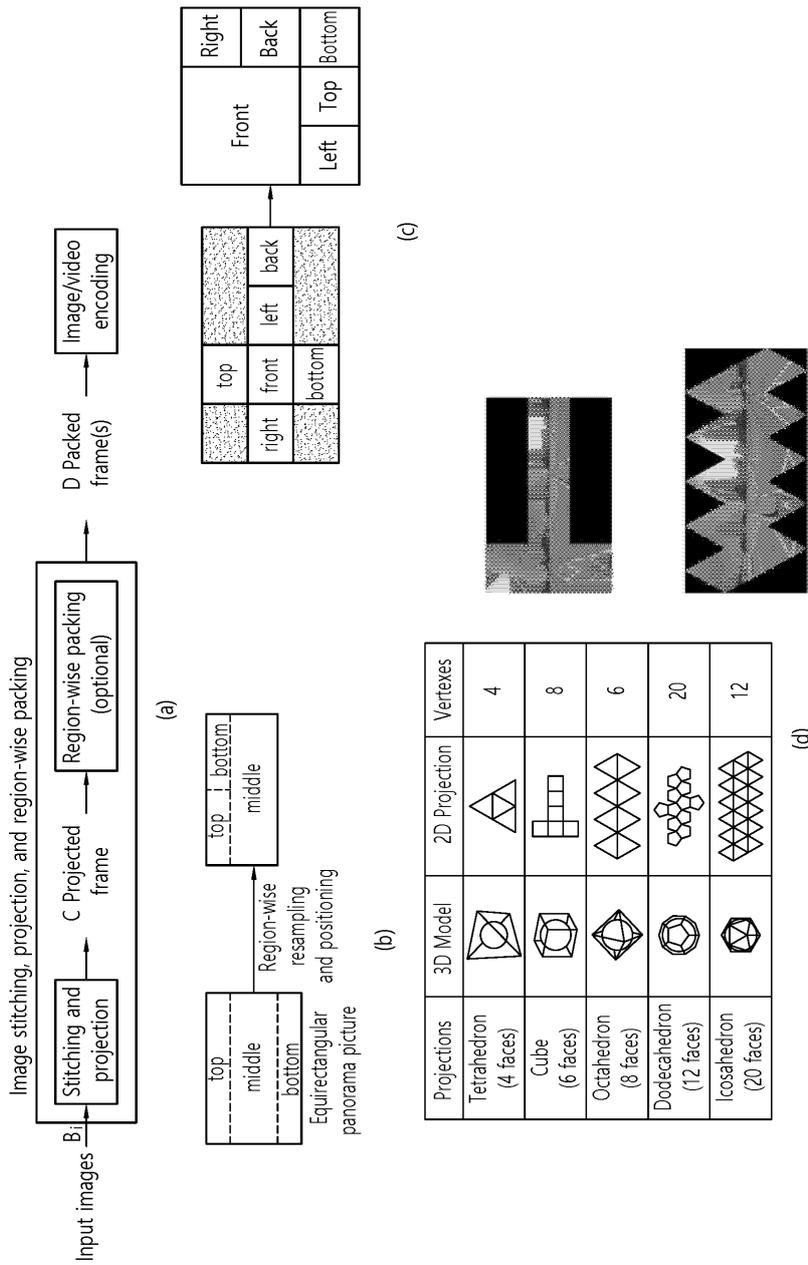
도면6



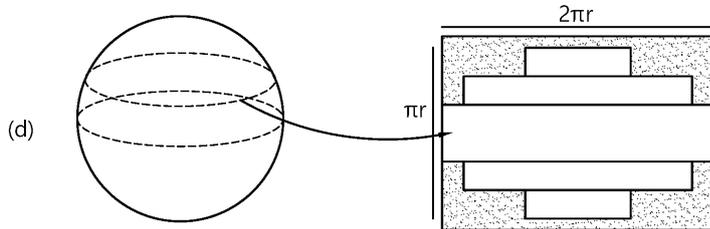
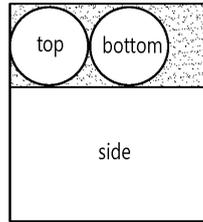
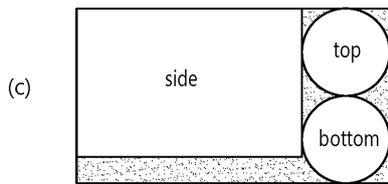
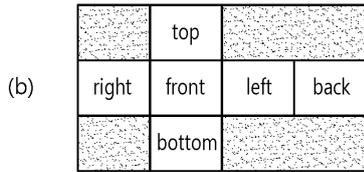
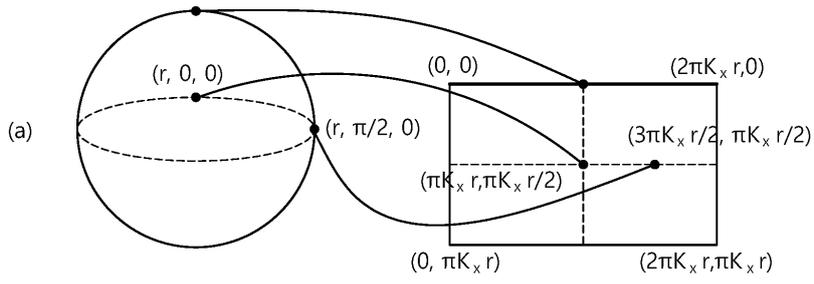
도면7



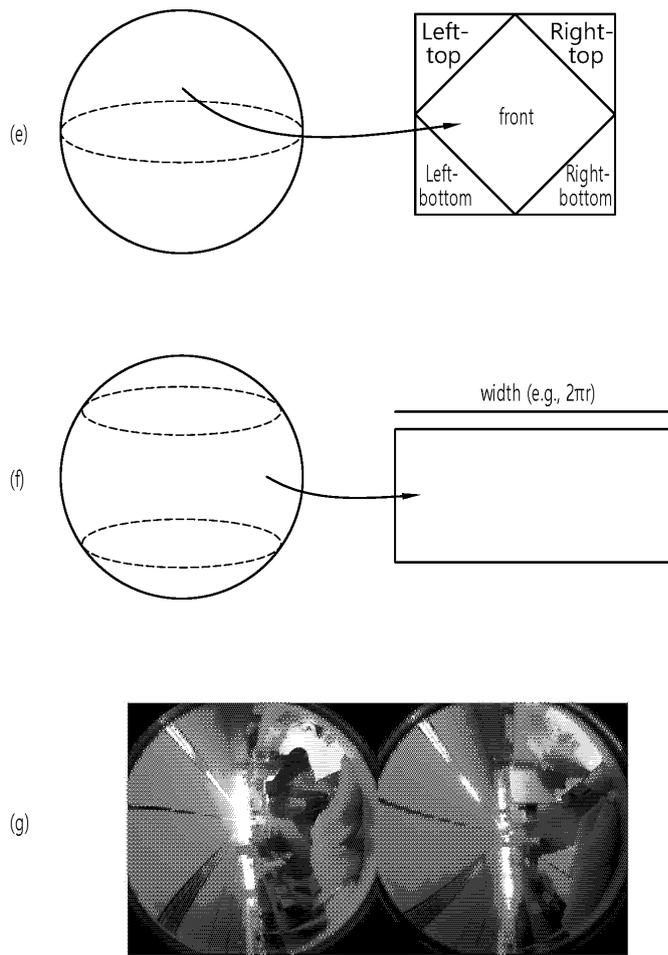
도면8



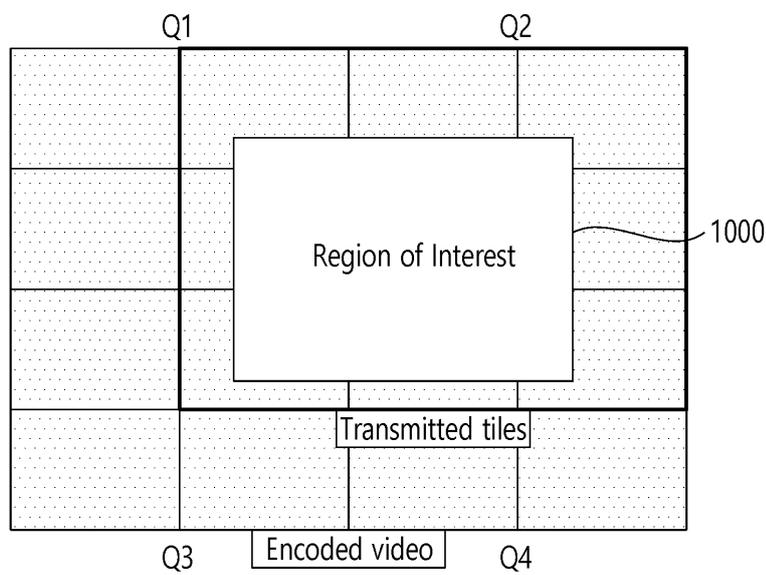
도면9a



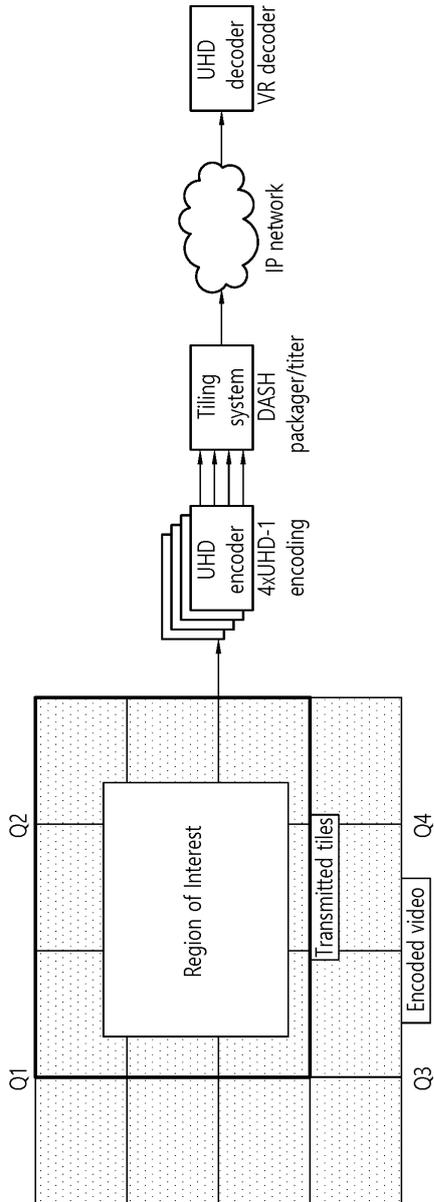
도면9b



도면10a



도면10b



도면11

```

...
    <기본 메타데이터>
    unsigned int(8)    vr_geometry;
    unsigned int(8)    projection_schme;

    <Stereoscopic 관련 메타데이터>
    unsigned int(1)    is_stereoscopic;
    unsigned int(3)    stereo_mode;

    <Initial View 관련 메타데이터>
    signed int(8)      initial_view_yaw_degree;
    signed int(8)      initial_view_pitch_degree;
    signed int(8)      initial_view_roll_degree;

    <ROI 관련 메타데이터>
    unsigned int(1)    2d_roi_range_flag;
    unsigned int(1)    3d_roi_range_flag;
    if (2d_roi_region_flag==1) {
        unsigned int(16) min_top_left_x;
        unsigned int(16) max_top_left_x;
        unsigned int(16) min_top_left_y;
        unsigned int(16) max_top_left_y;
        unsigned int(16) min_width;
        unsigned int(16) max_width;
        unsigned int(16) min_height;
        unsigned int(16) max_height;
        unsigned int(16) min_x;
        unsigned int(16) max_x;
        unsigned int(16) min_y;
        unsigned int(16) max_y;
    }
    if (3d_roi_region_flag==1){
        unsigned int(16) min_yaw;
        unsigned int(16) max_yaw;
        unsigned int(16) min_pitch;
        unsigned int(16) max_pitch;
        unsigned int(16) min_roll;
        unsigned int(16) max_roll;
        unsigned int(16) min_field_of_view;
        unsigned int(16) max_field_of_view;
    }

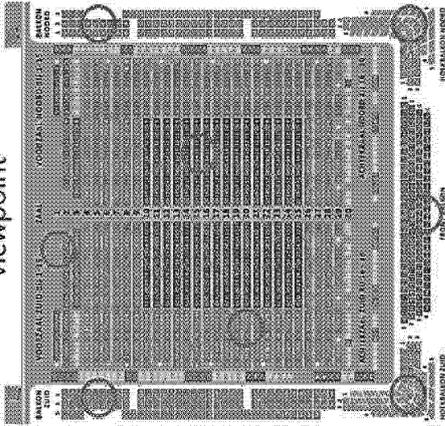
    <Field Of View 관련 메타데이터>
    unsigned int(1)    content_fov_flag;
    if (content_fov_flag==1) {
        unsigned int(16)    content_fov;
    }

    <Cropped Region 관련 메타데이터>
    unsigned int(1)    is_copped_region;
    if (content_fov_flag==1) {
        unsigned int(16)    cr_region_left_top_x;
        unsigned int(16)    cr_region_left_top_y;
        unsigned int(16)    cr_region_width;
        unsigned int(16)    cr_region_height;
    }
}
...

```

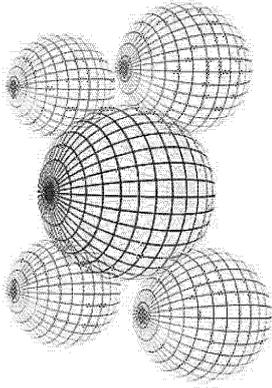
도면12

viewpoint



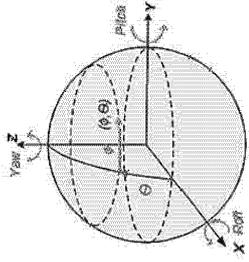
(a)

viewing position



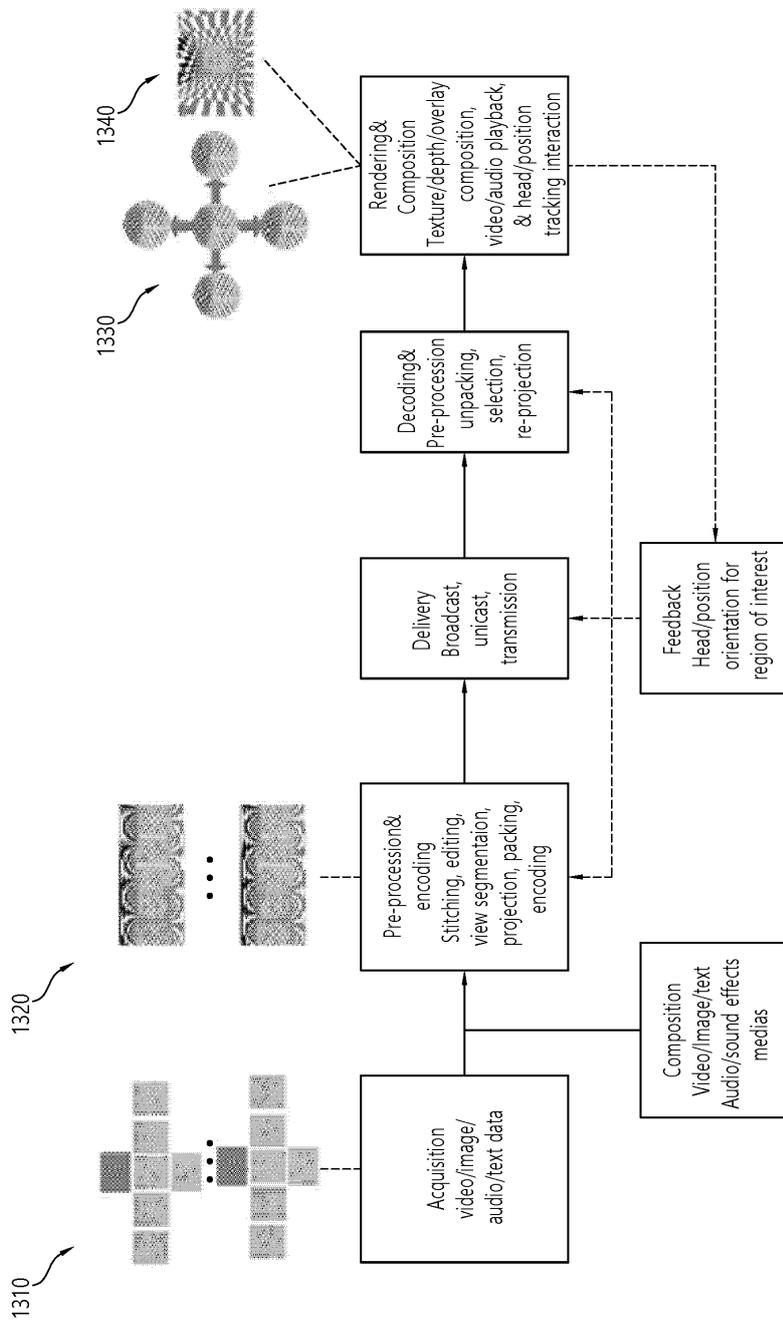
(b)

viewing orientation

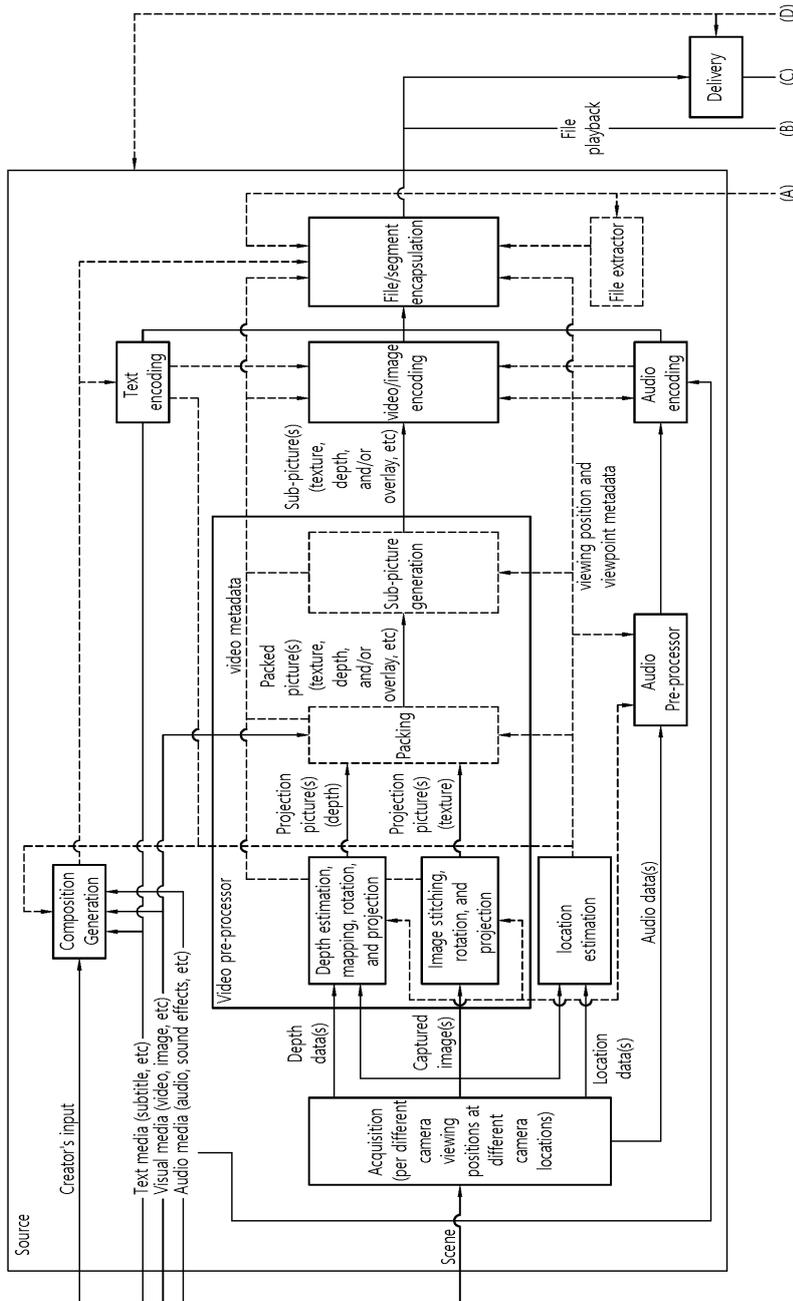


(c)

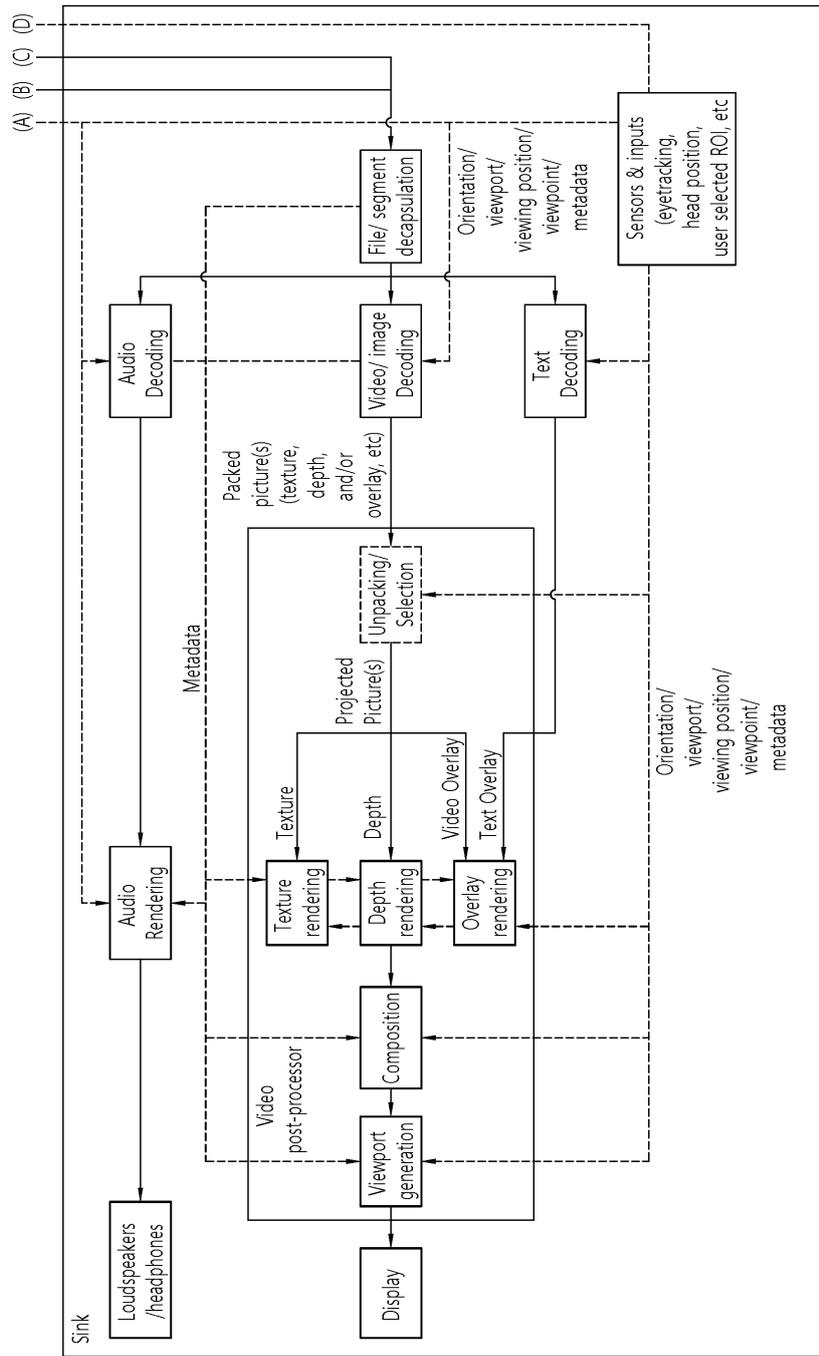
도면13



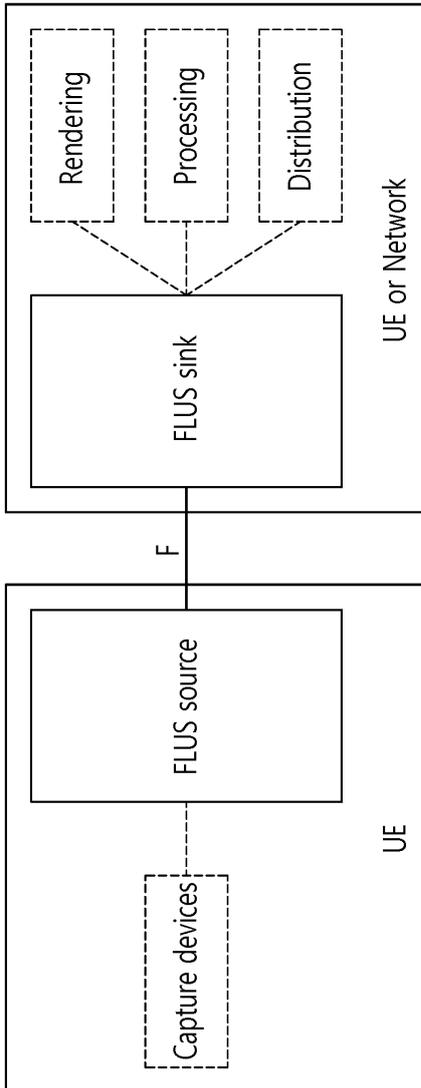
도면14a



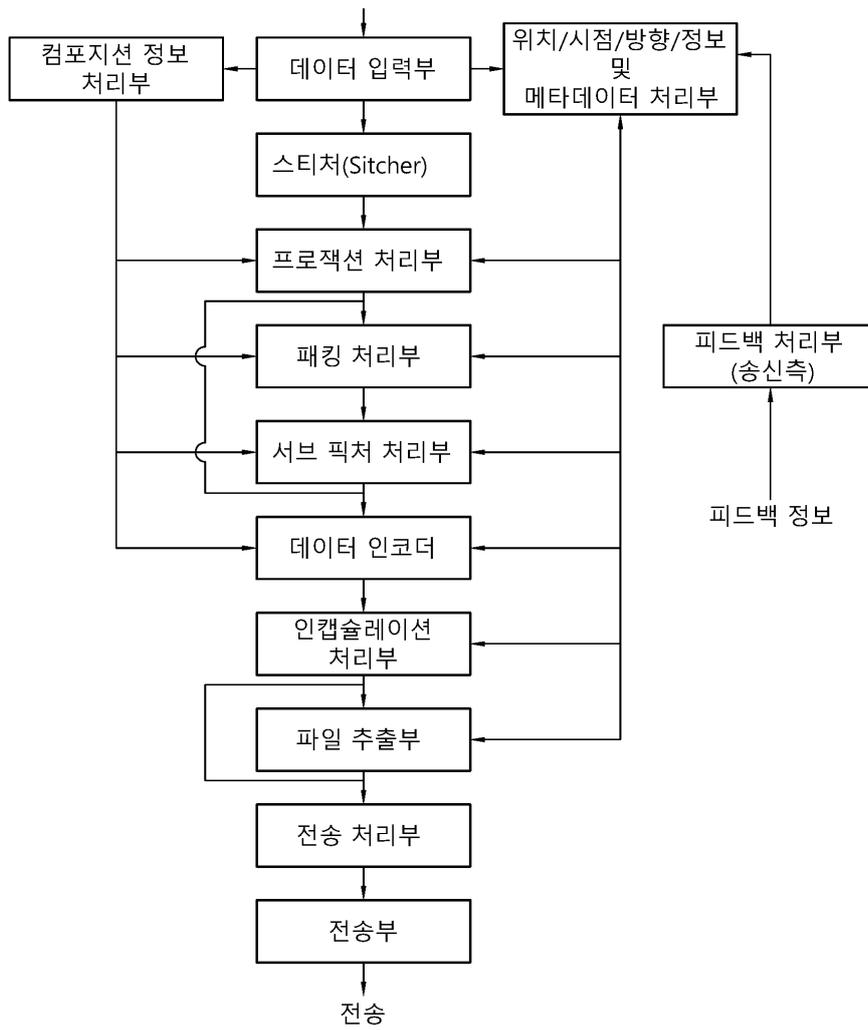
도면14b



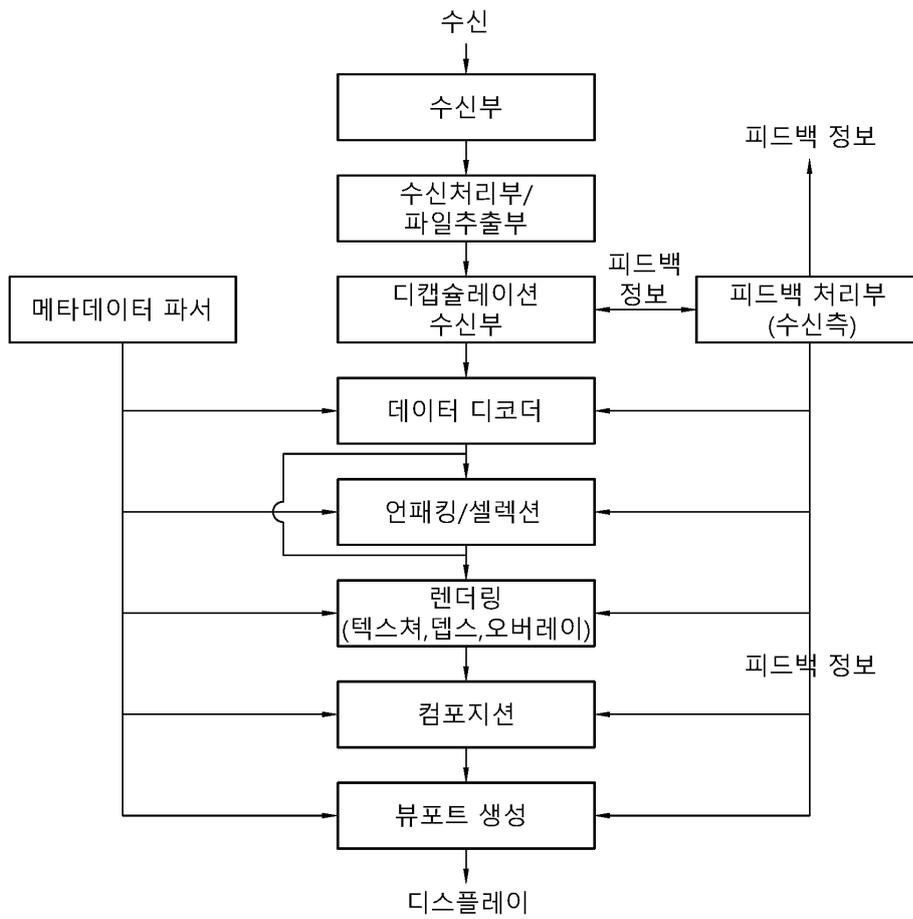
도면15



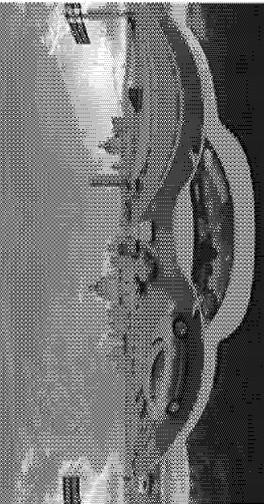
도면16



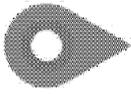
도면17



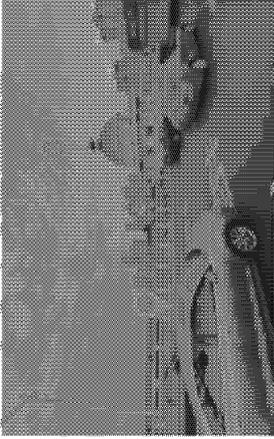
도면18



360 비디오

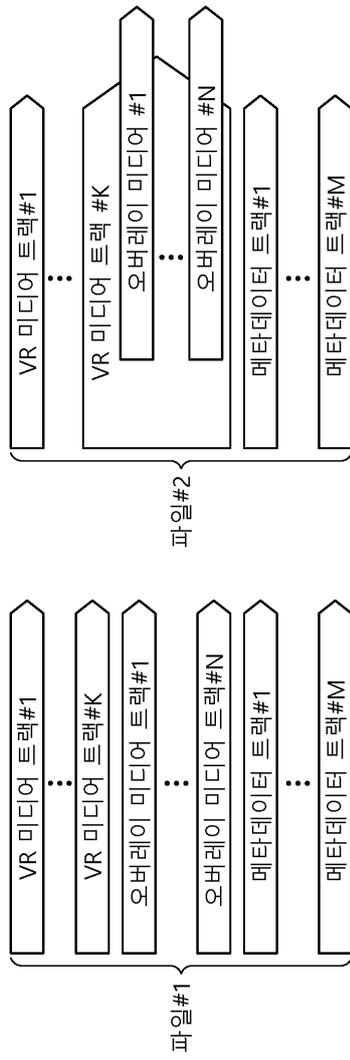


오버레이되는
이미지

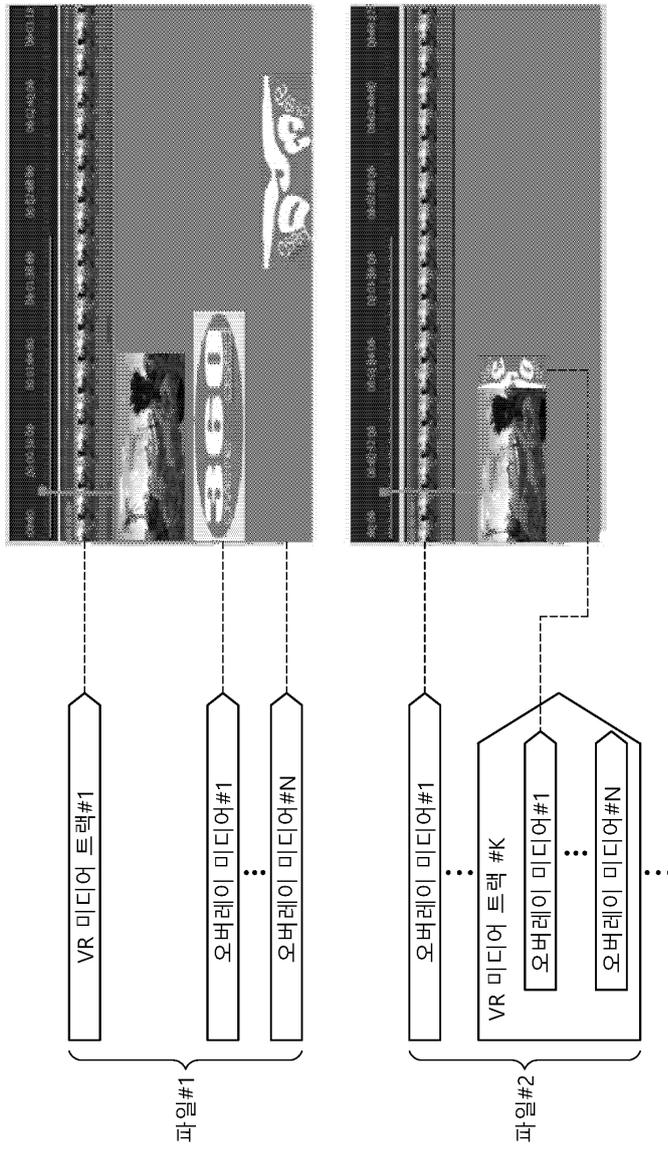


360 비디오 상에 렌더링된 오버레이

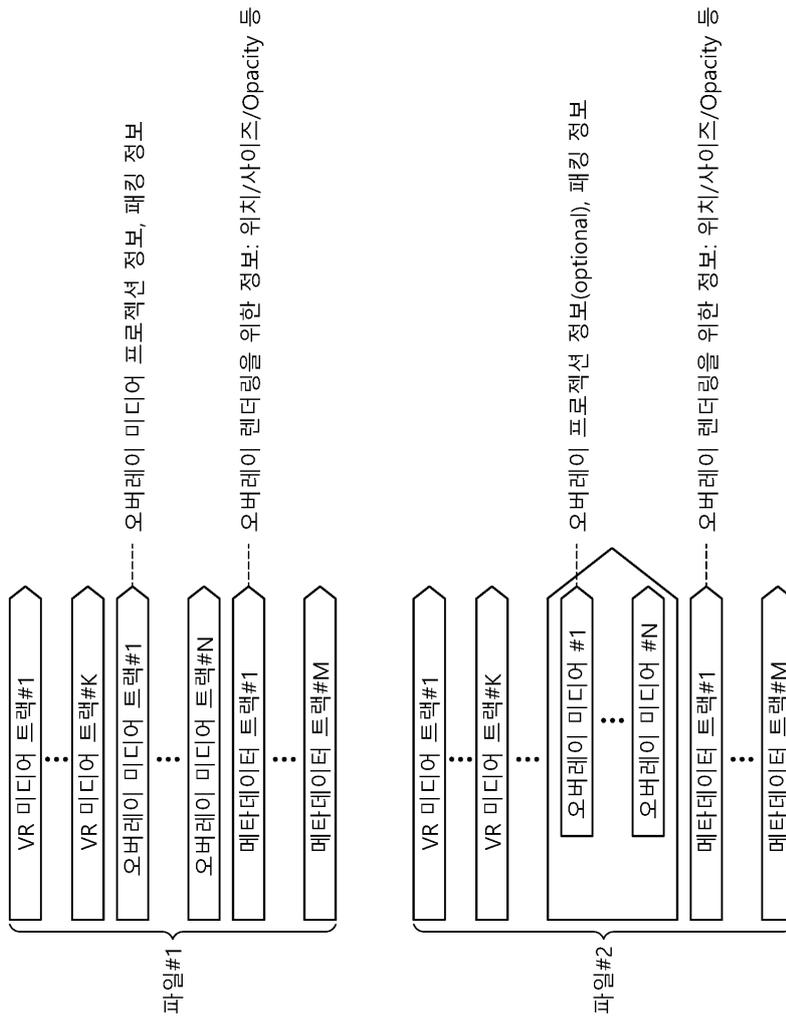
도면19



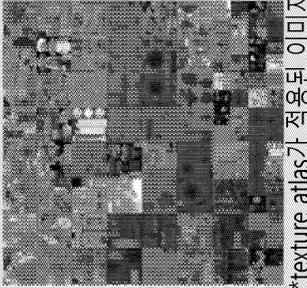
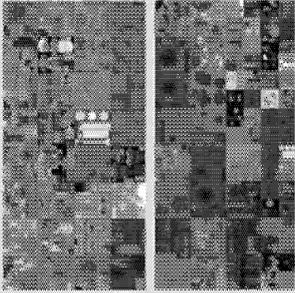
도면20



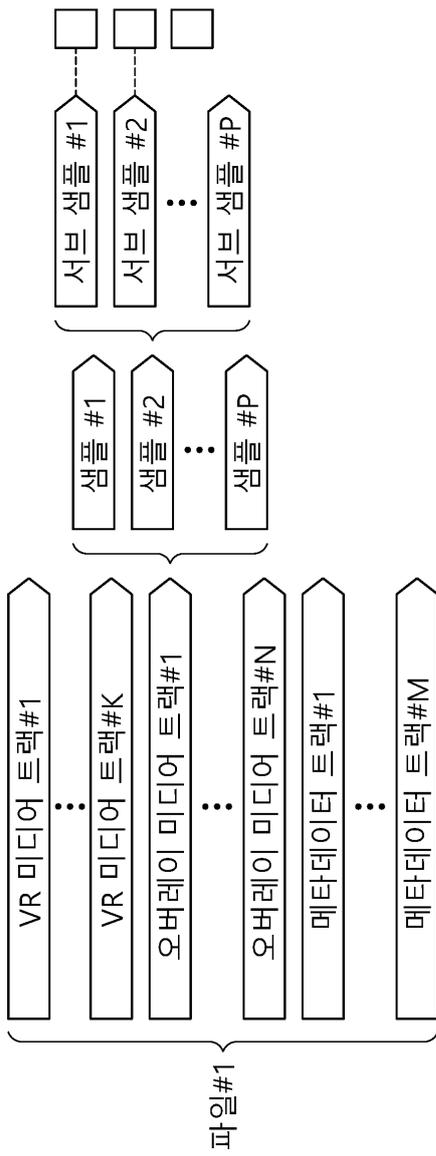
도면21



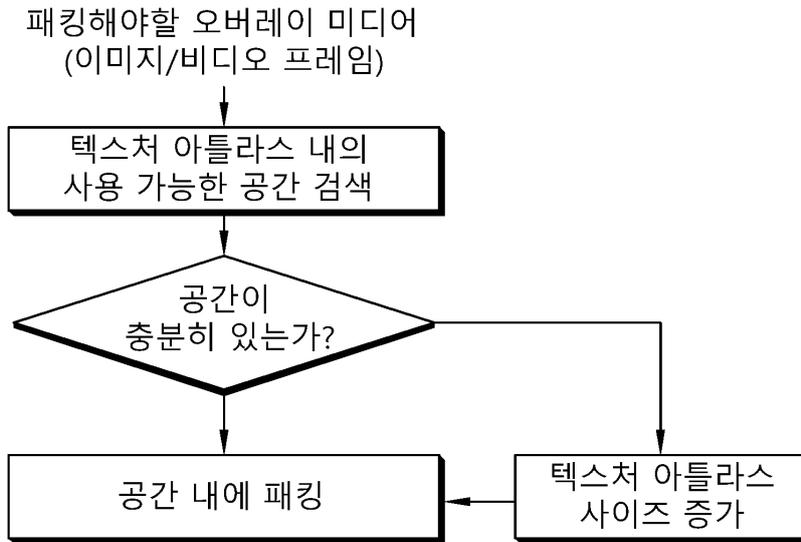
도면22

	Case1	Case2	Case3	Case4
# of 이미지	1	N (서브샘플스)	1	M (서브샘플스)
# of 오버레이	1	N	N	M
Example			 *texture atlas가 적용된 이미지	

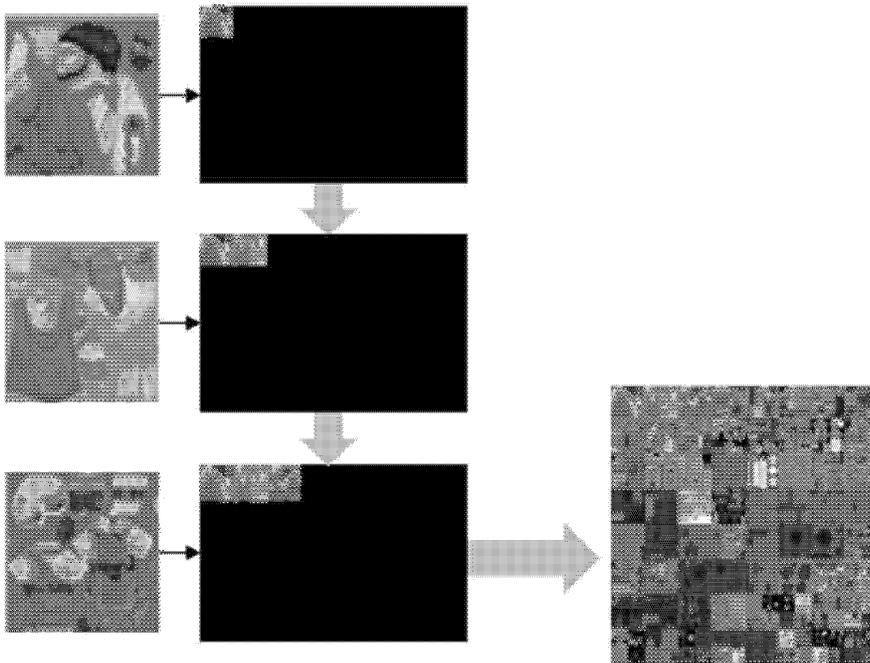
도면23



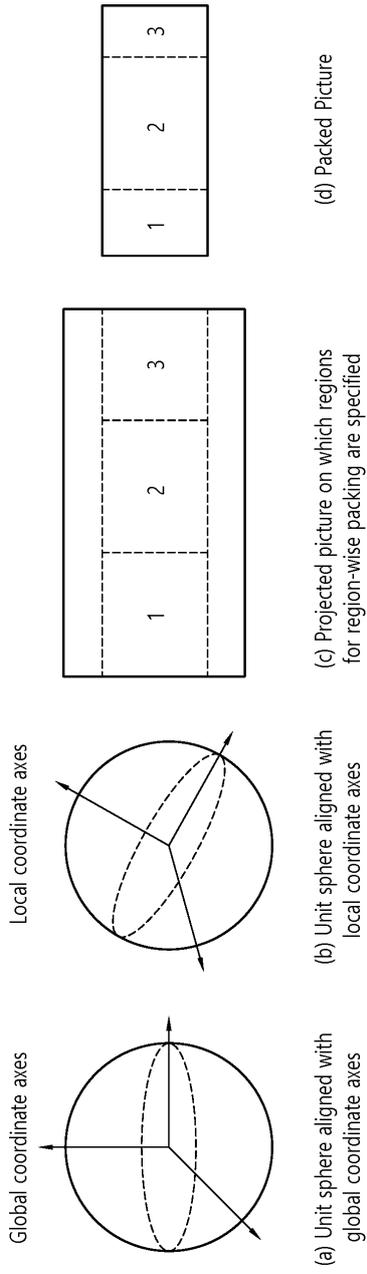
도면24



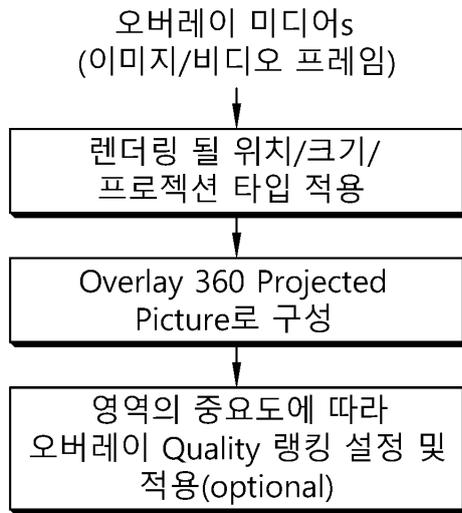
도면25



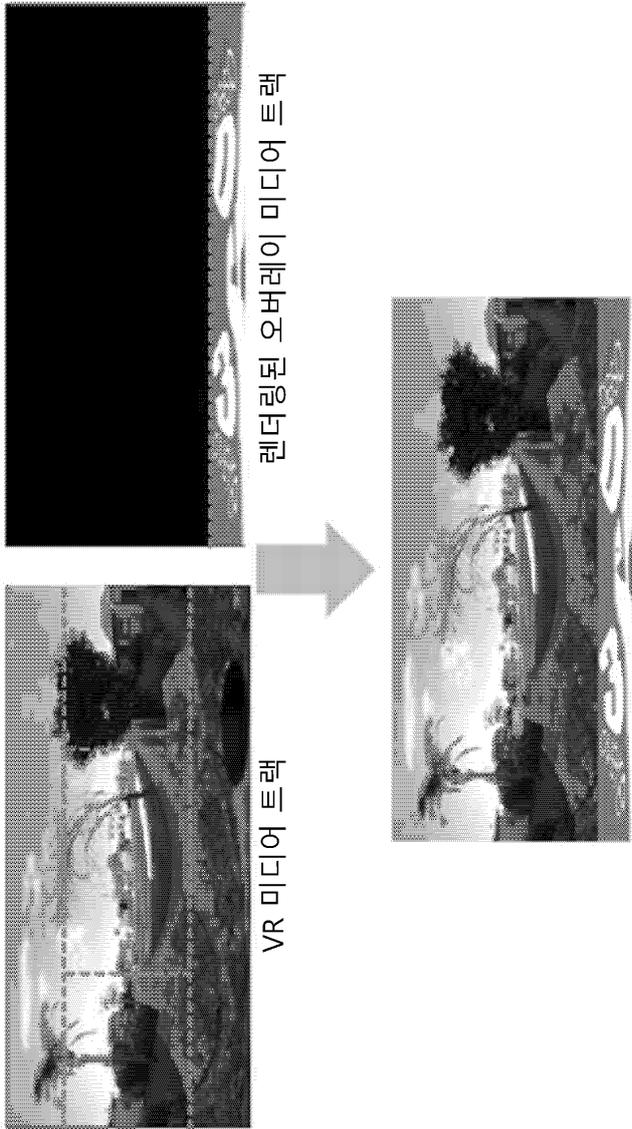
도면26



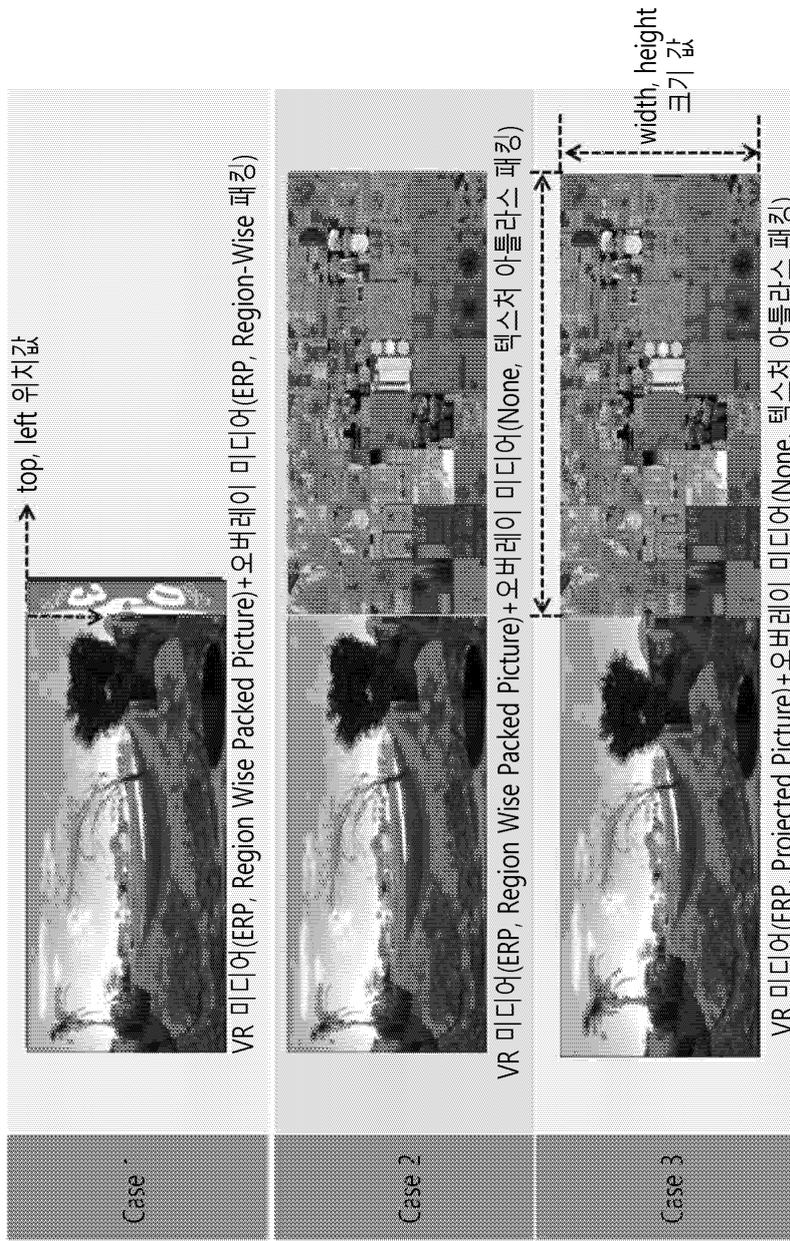
도면27



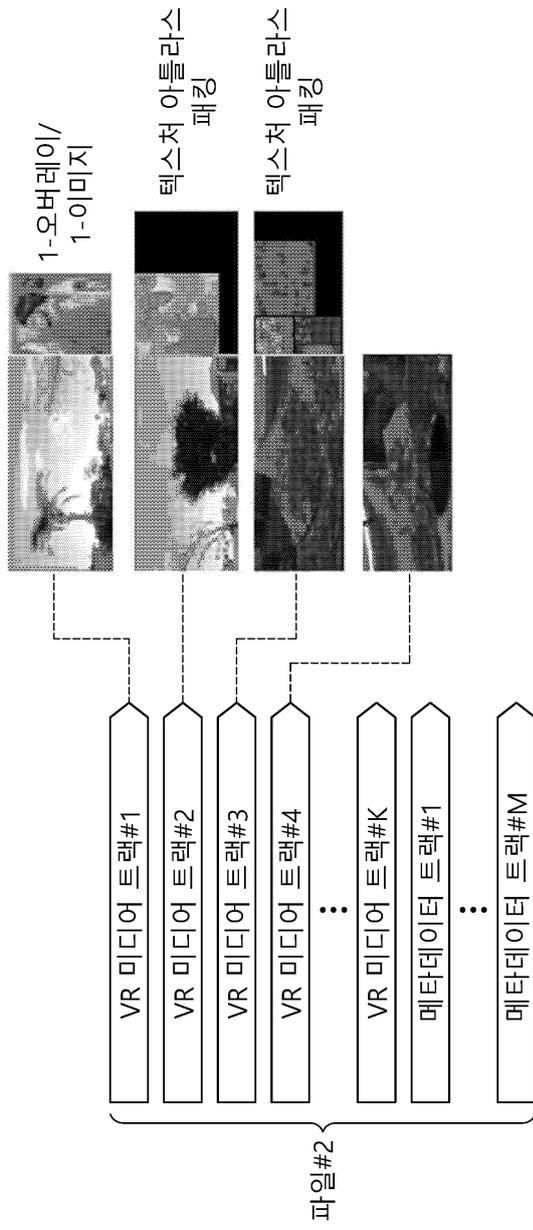
도면28



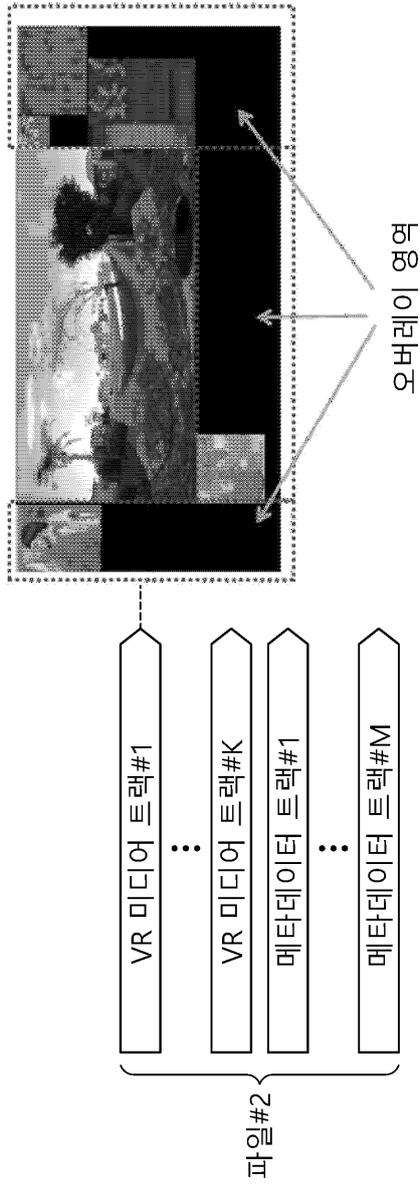
도면29



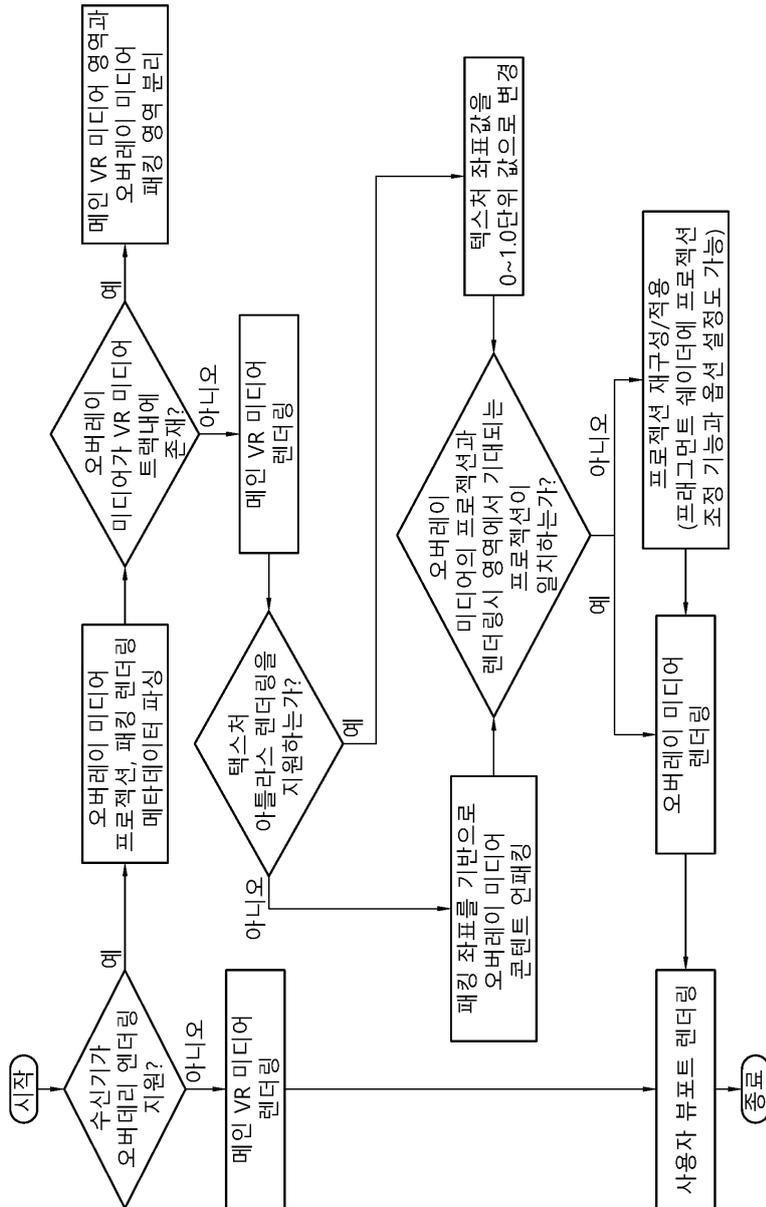
도면30



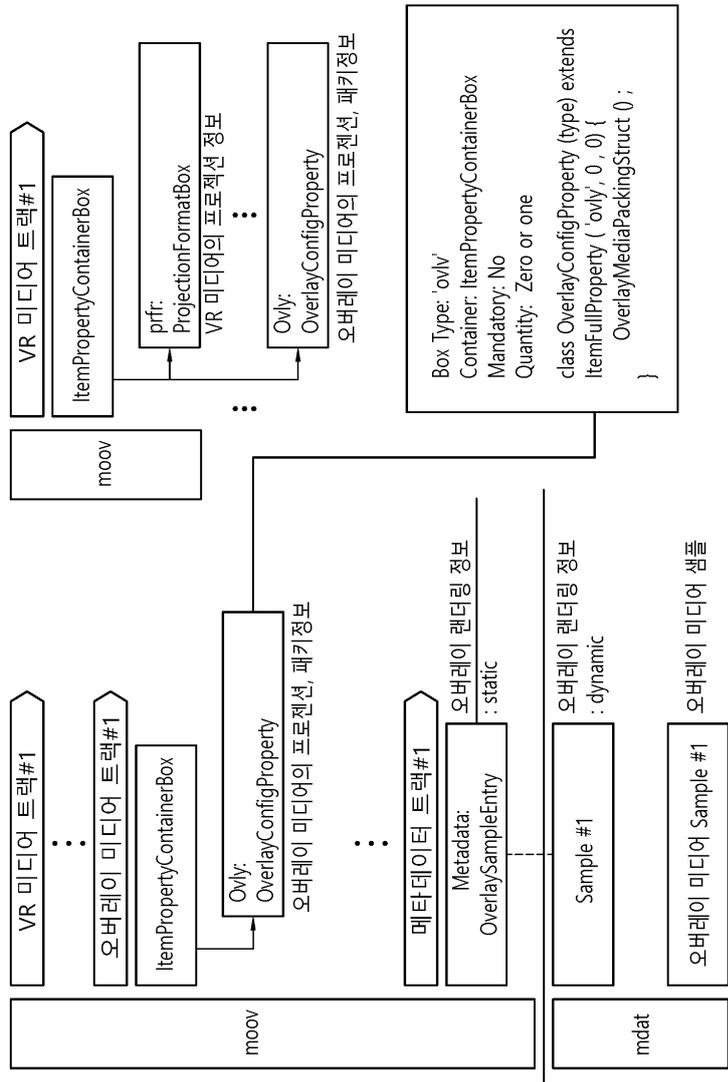
도면31



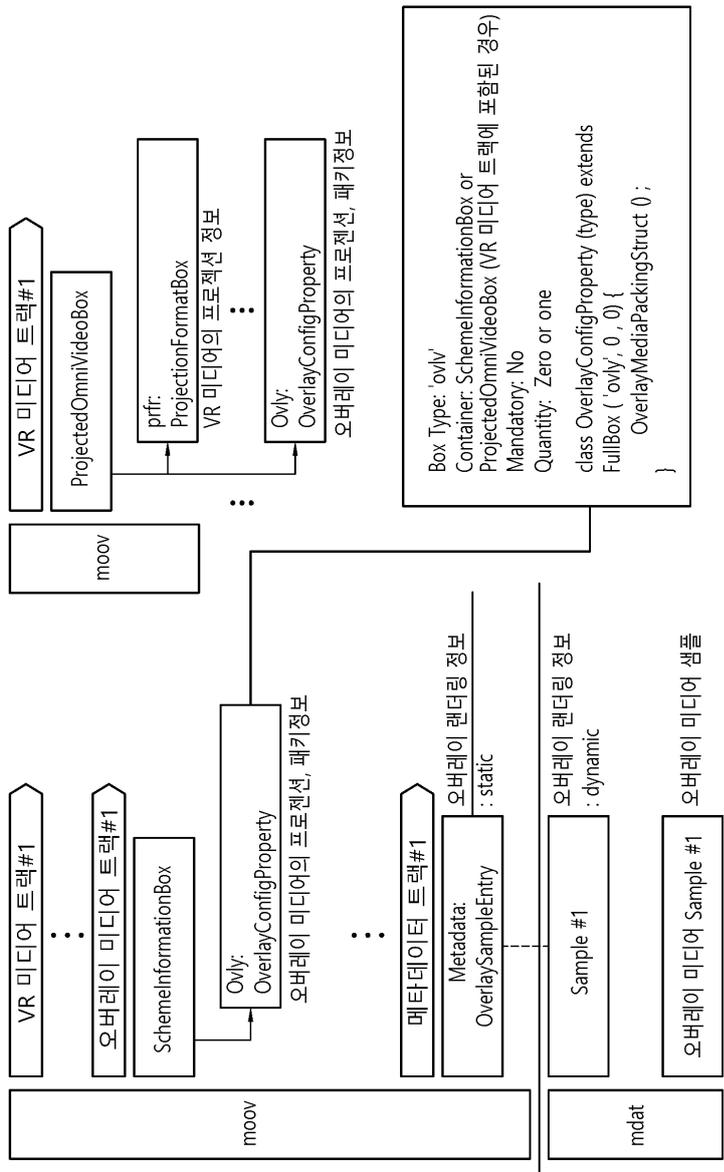
도면32



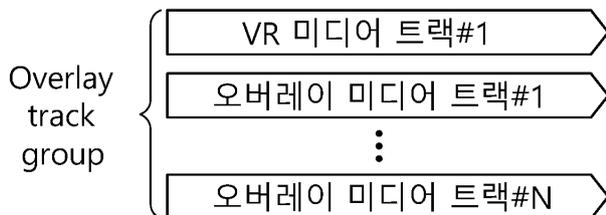
도면33



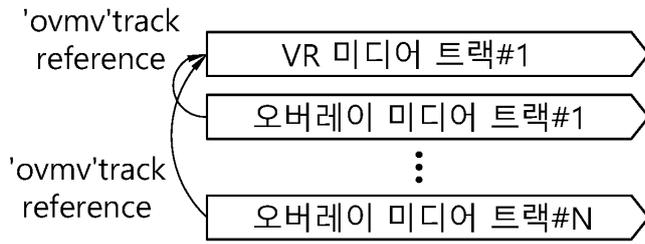
도면34



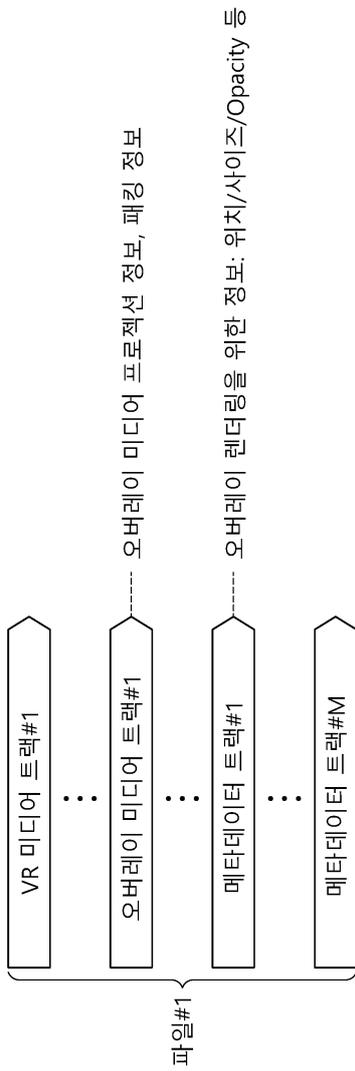
도면35a



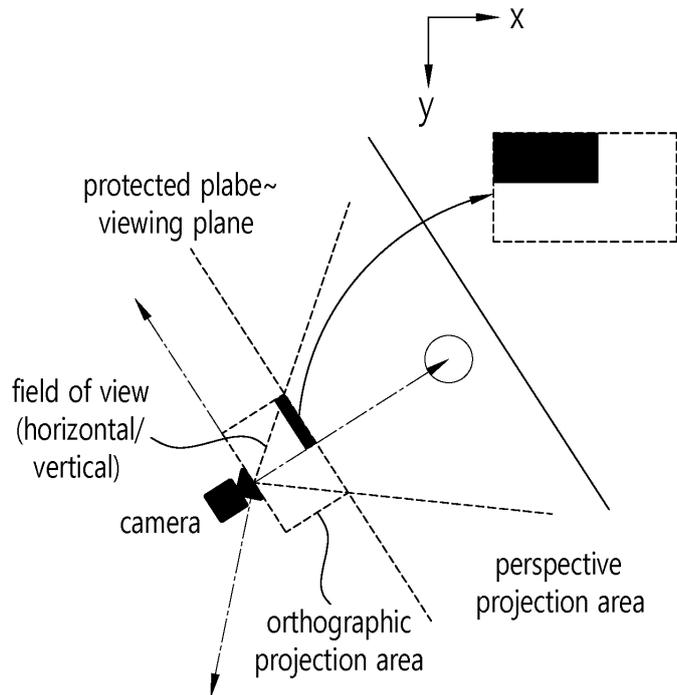
도면35b



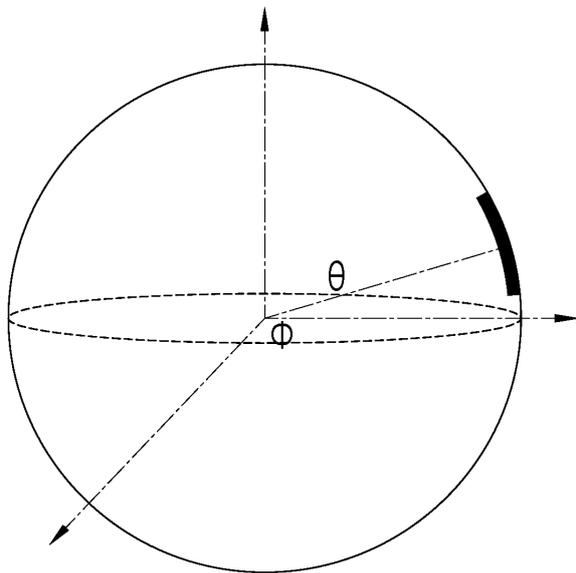
도면36



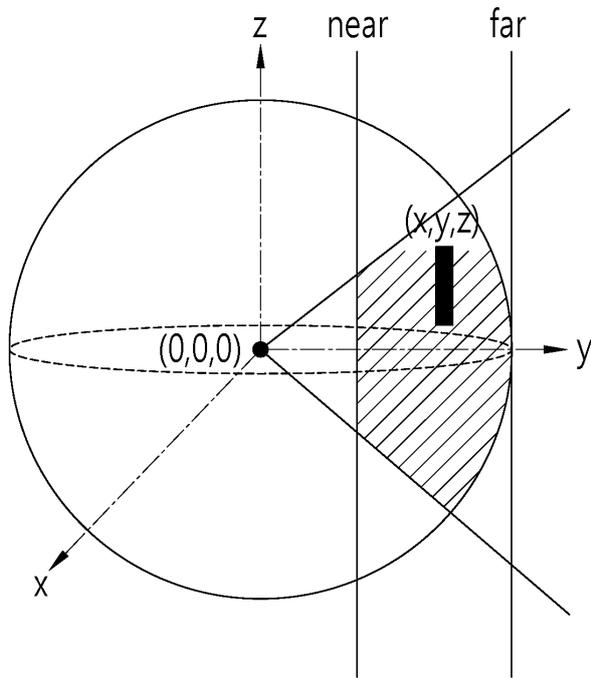
도면37a



도면37b



도면37c



도면38



뷰포트 상에
배치하는 경우

도면39

Sphere 상에
배치하는 경우

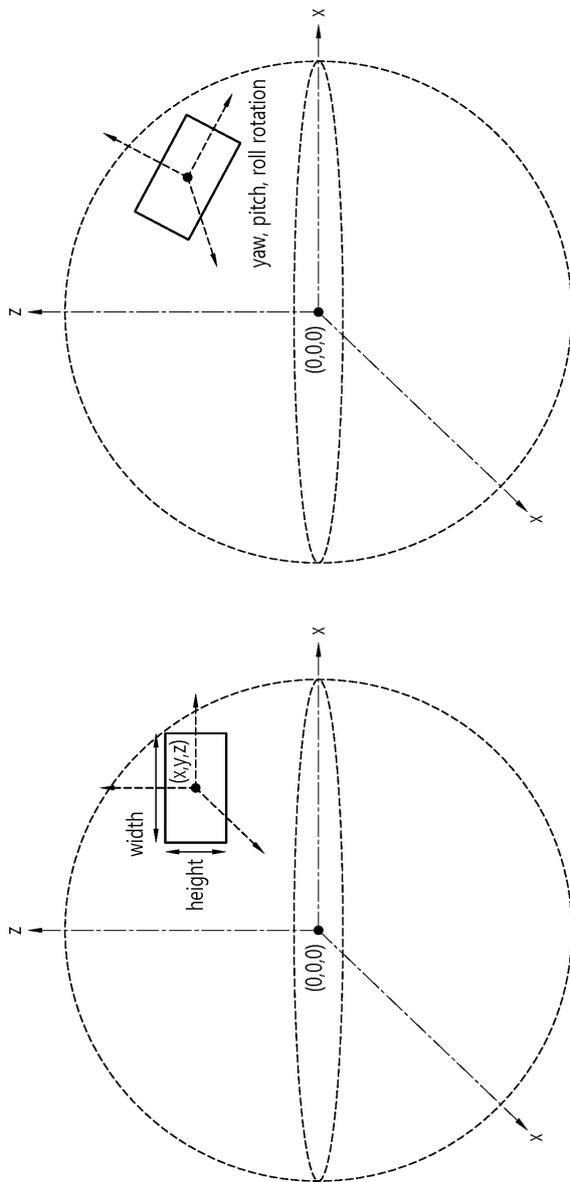


도면40

3차원 공간 상에
배치하는 경우



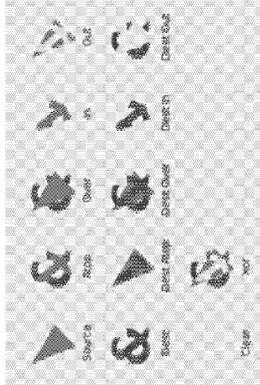
도면41



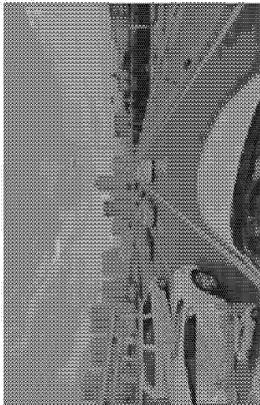
도면42



Focus 효과-
다른 부분들을 Blur 처리함



블렌딩 옵션-



오버레이 평면의 투명도 예-
VR 미디어 영상의 이미지가 보임

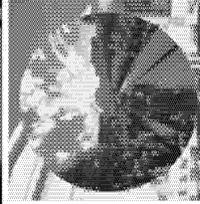
도면43



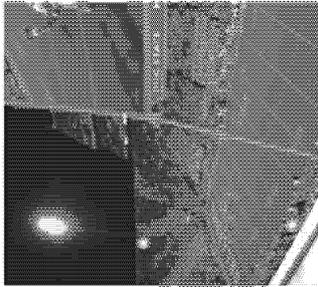
타겟 표현



뷰포인트 적용

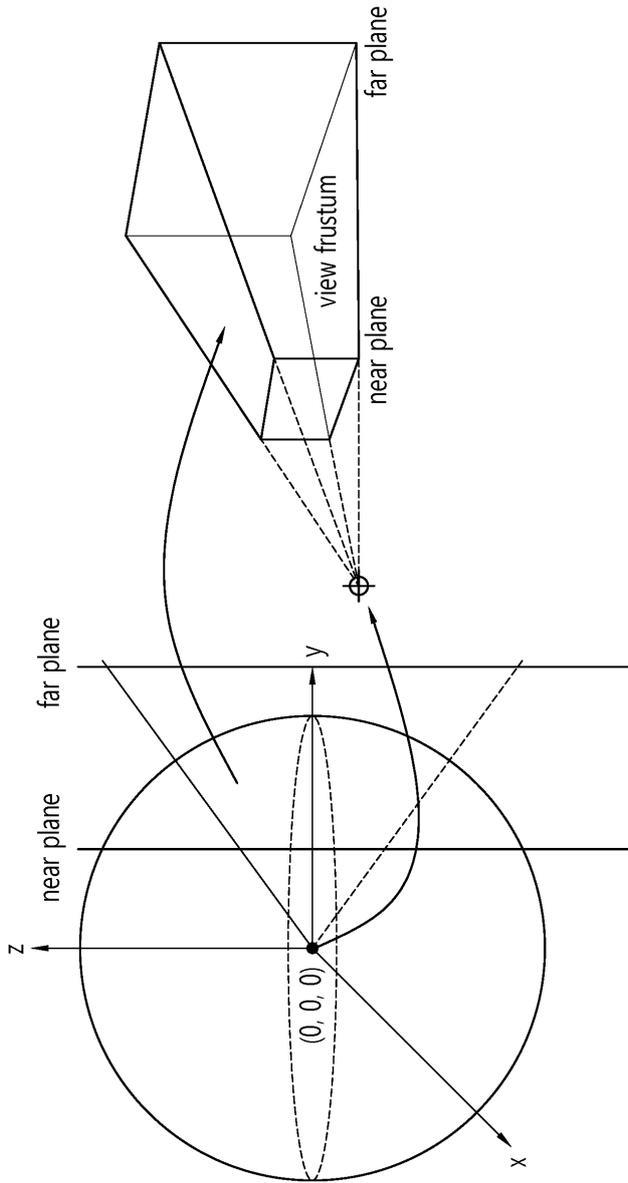


다양한 오버레이
Shape 지원

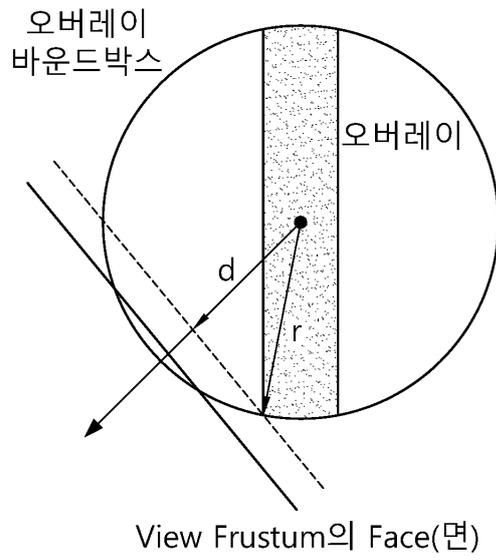


오버레이 테두리 지원

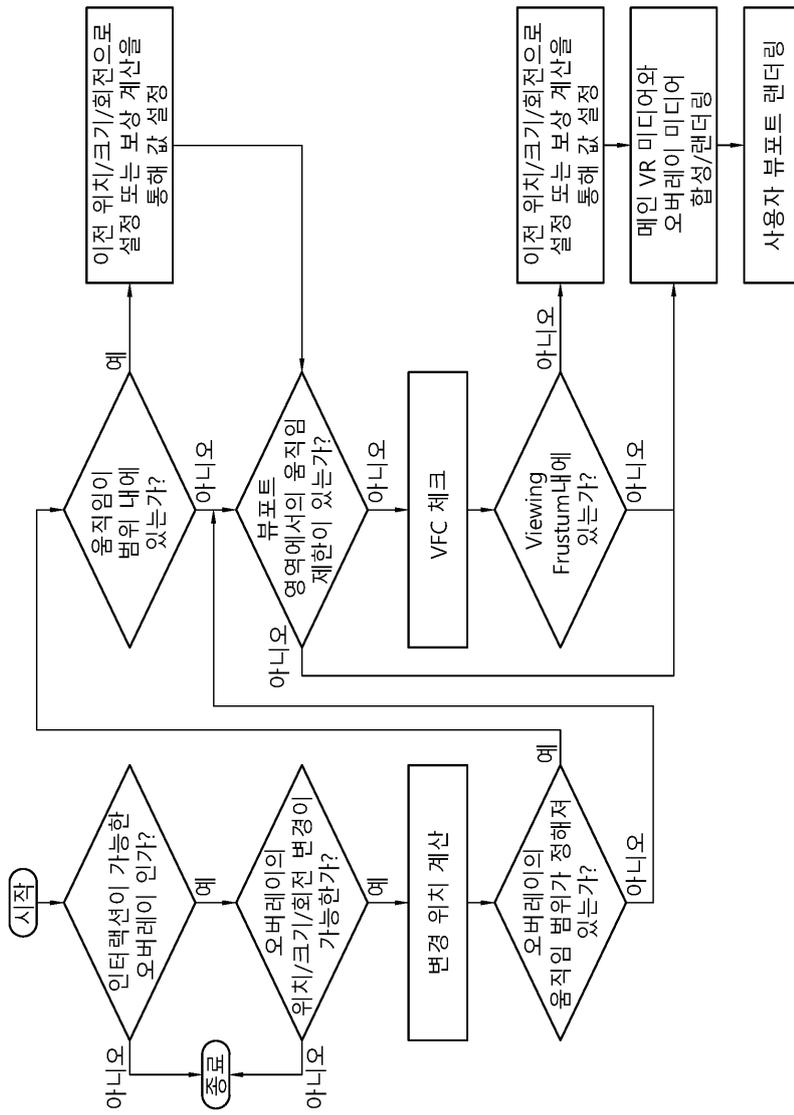
도면44



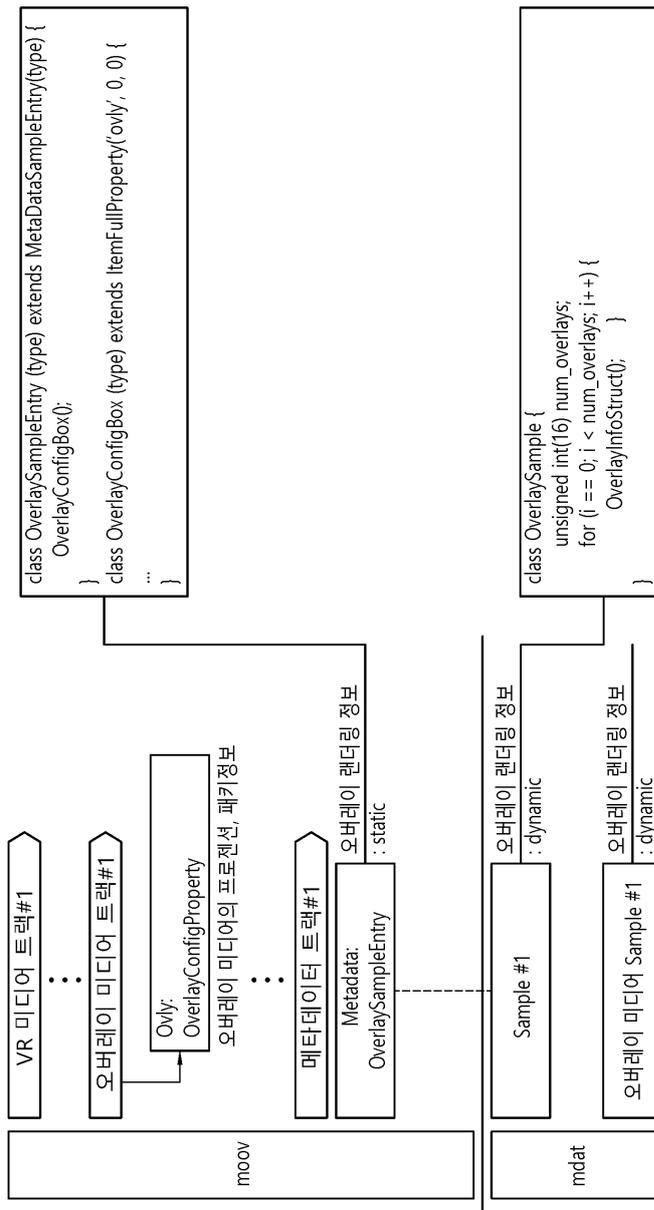
도면45



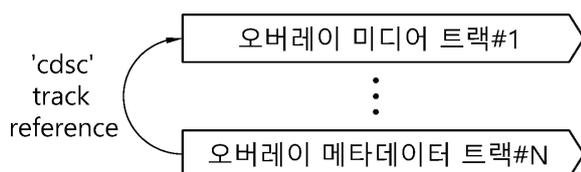
도면46



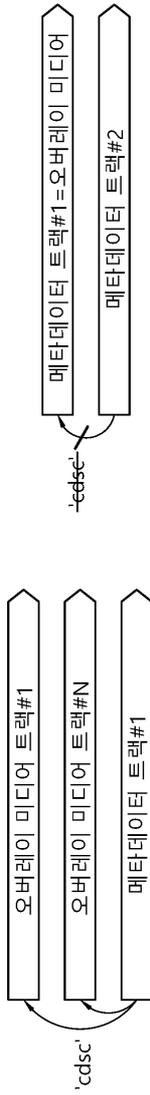
도면47



도면48



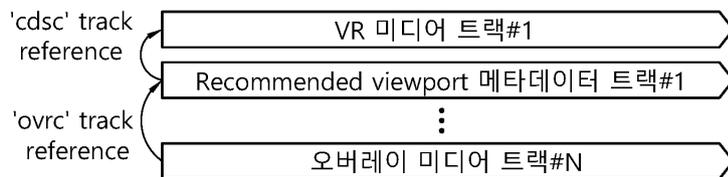
도면49



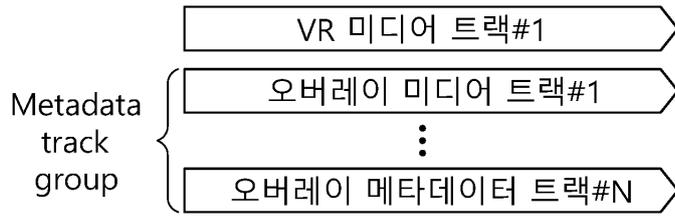
도면50



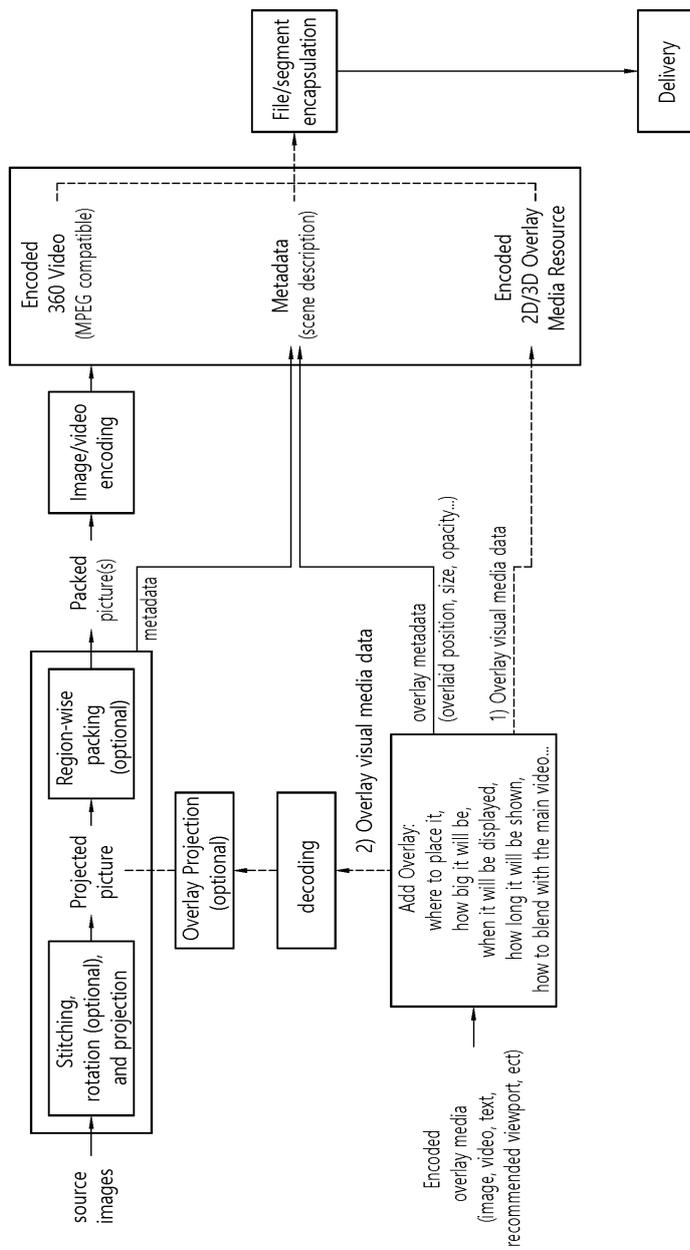
도면51



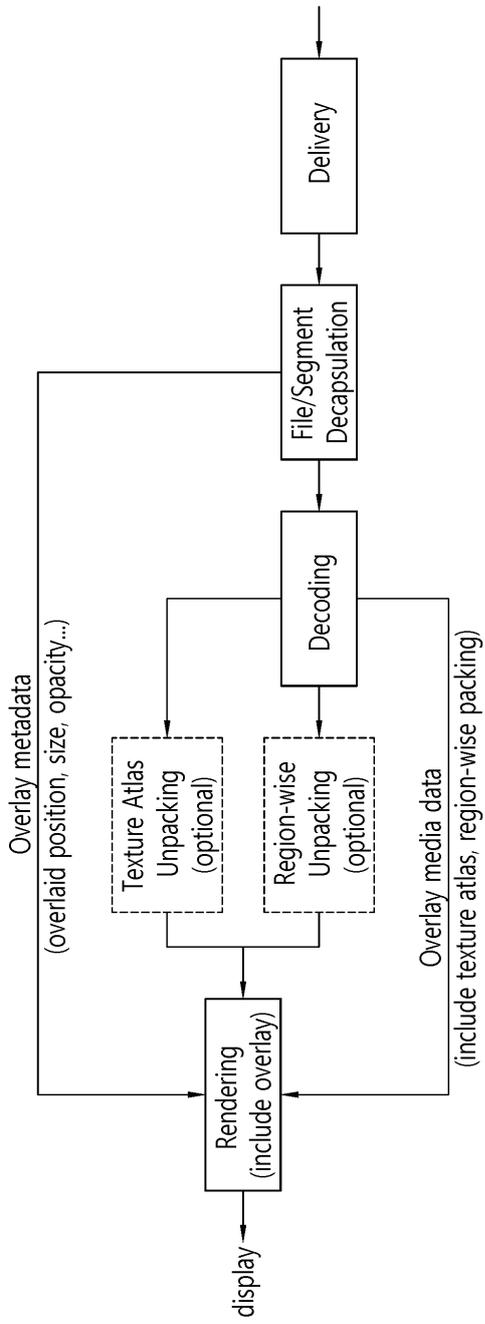
도면52



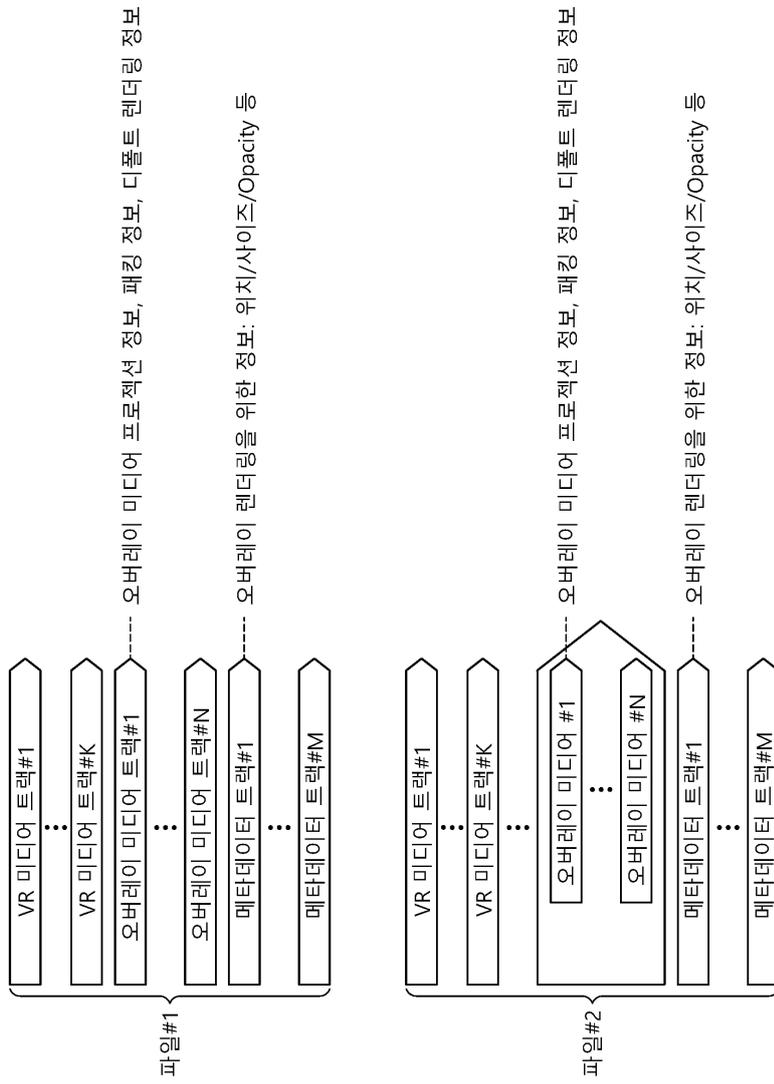
도면53



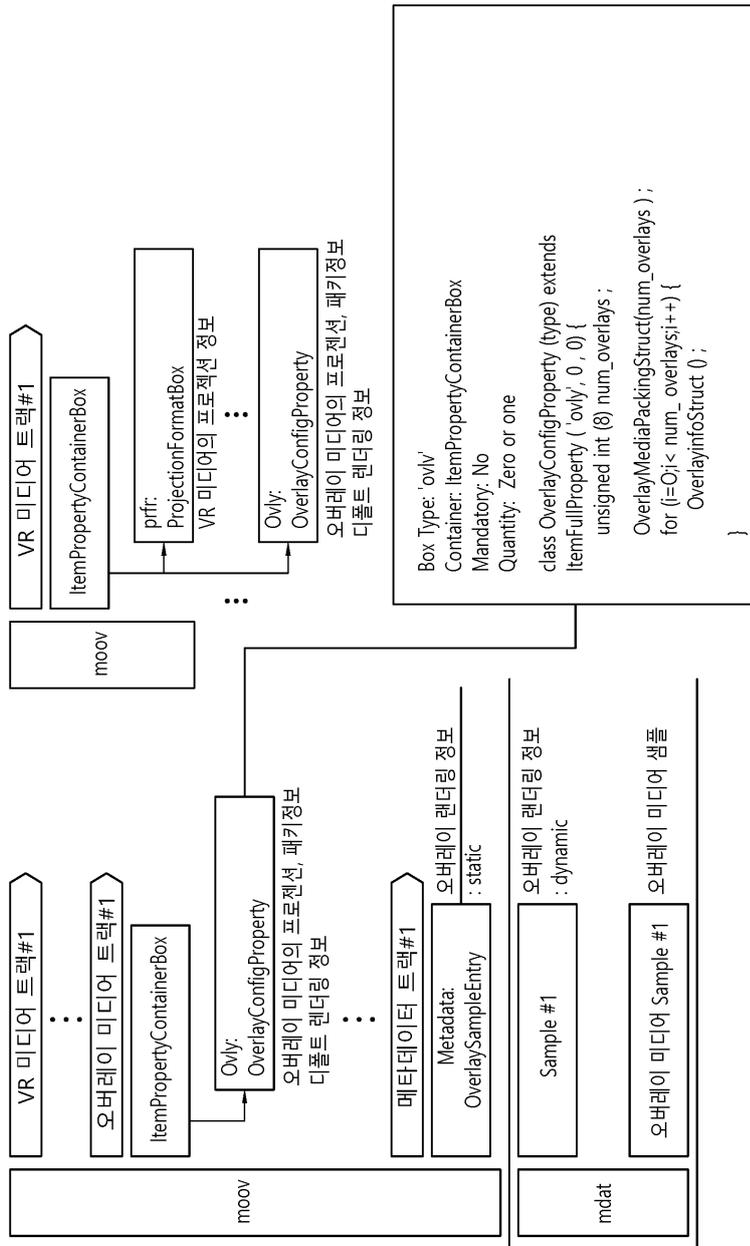
도면54



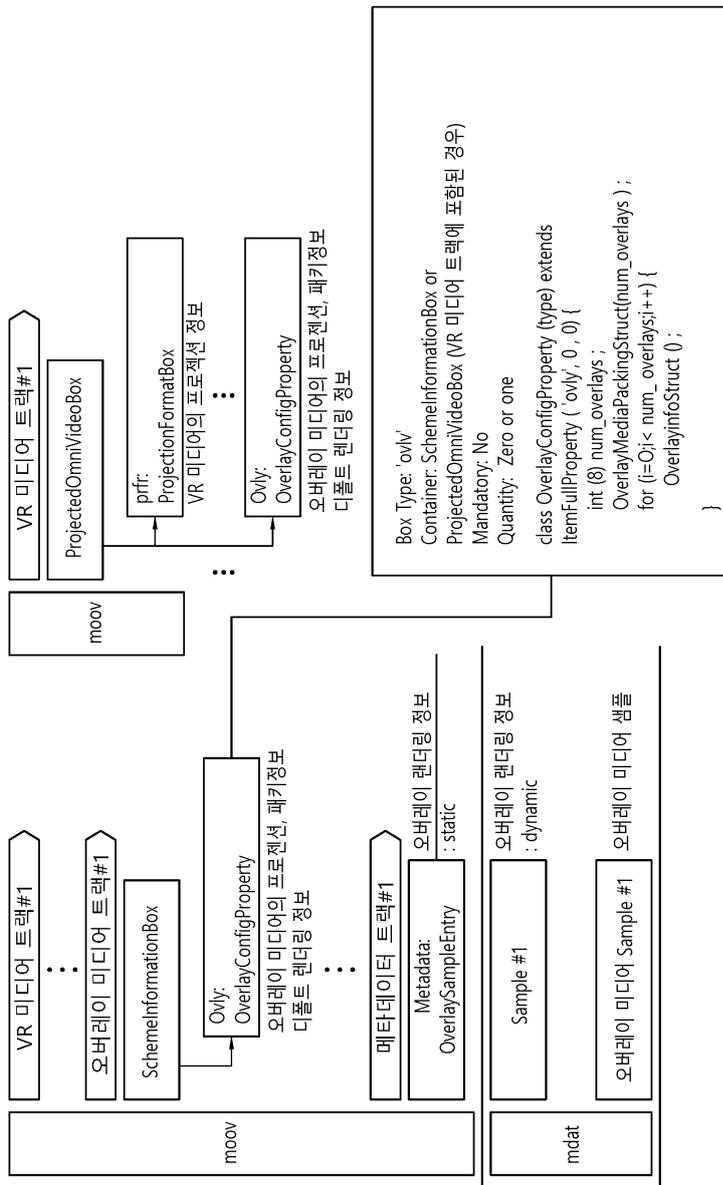
도면55



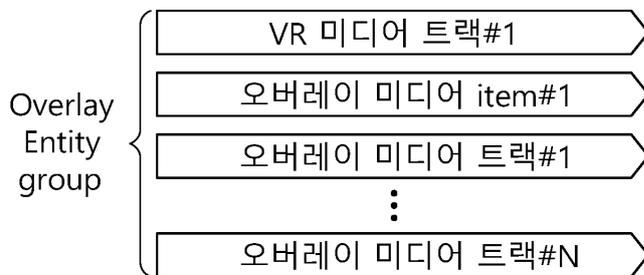
도면56



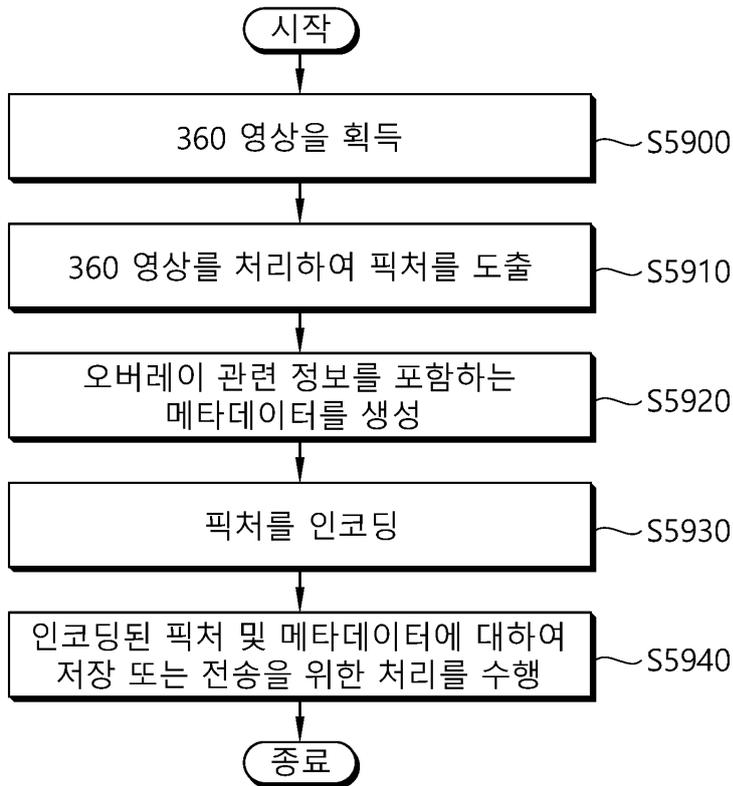
도면57



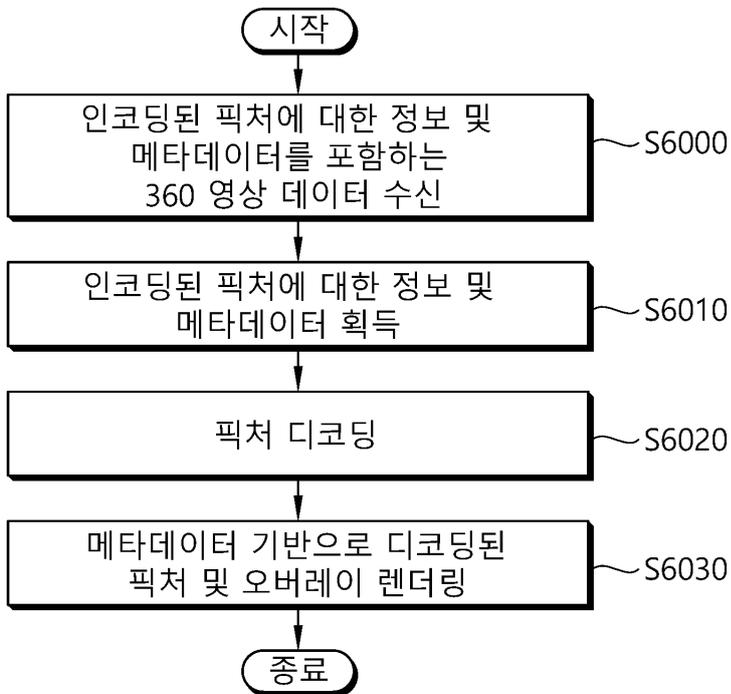
도면58



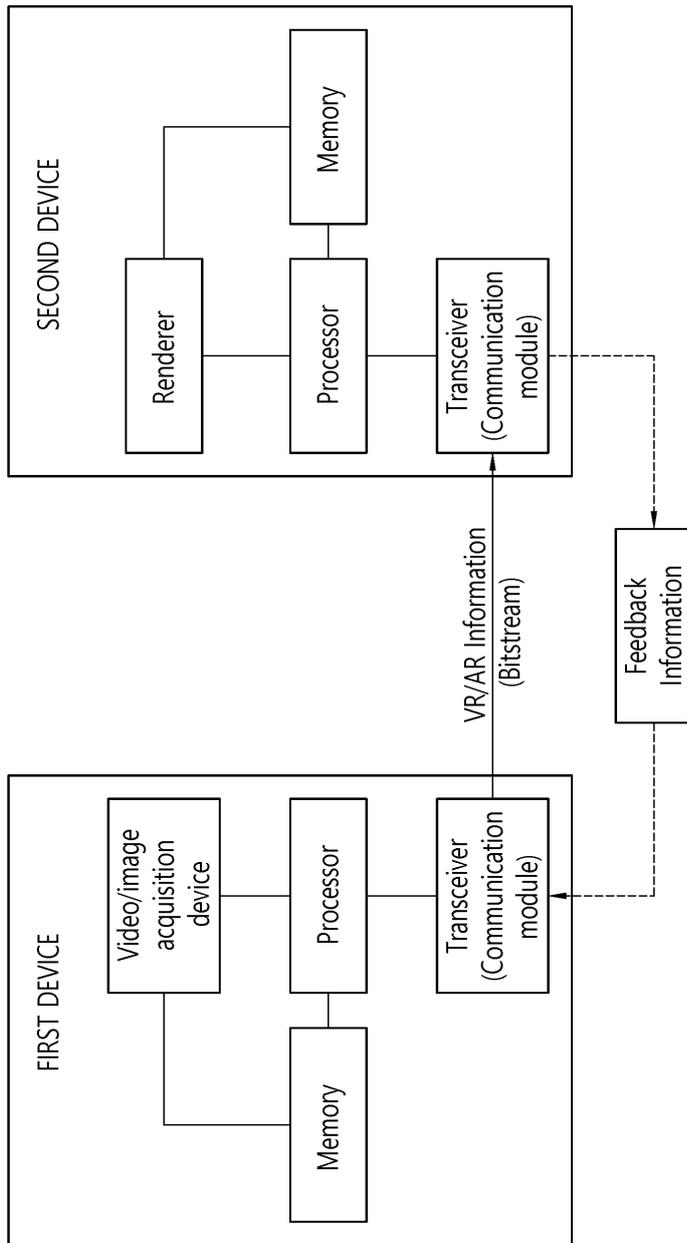
도면59



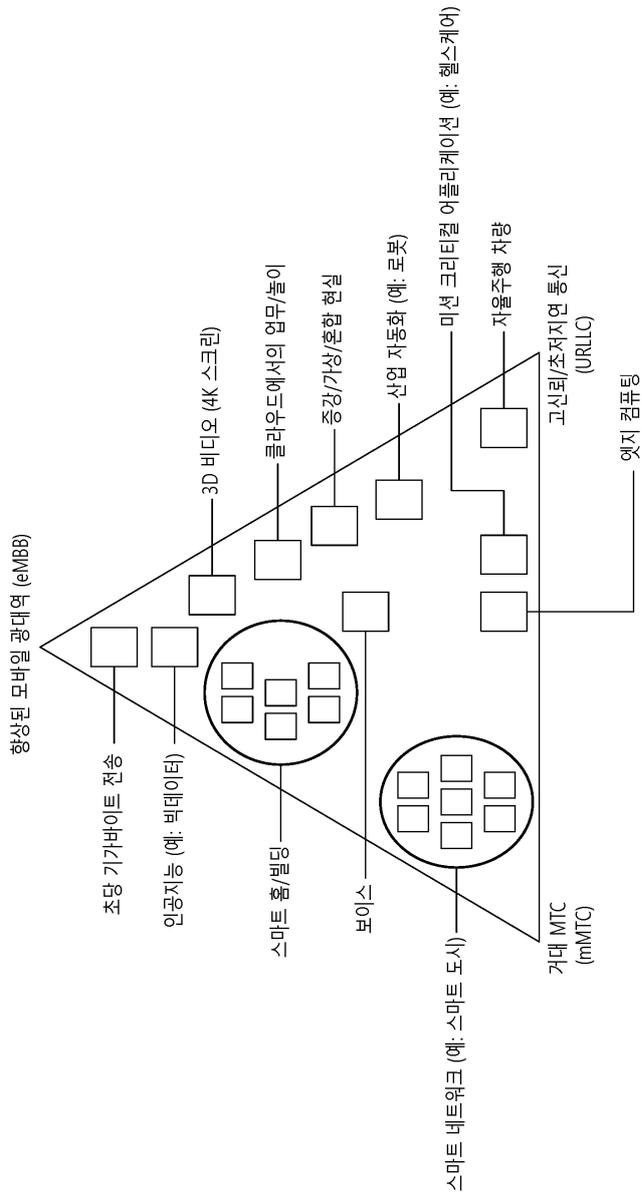
도면60



도면61



도면62



도면63

