



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2019년04월11일
(11) 등록번호 10-1968376
(24) 등록일자 2019년04월05일

(51) 국제특허분류(Int. Cl.)
H04N 21/434 (2011.01) H04N 21/81 (2011.01)
H04N 21/845 (2011.01)
(21) 출원번호 10-2014-7025853
(22) 출원일자(국제) 2013년02월27일
심사청구일자 2018년02월13일
(85) 번역문제출일자 2014년09월16일
(65) 공개번호 10-2014-0131360
(43) 공개일자 2014년11월12일
(86) 국제출원번호 PCT/US2013/028050
(87) 국제공개번호 WO 2013/130631
국제공개일자 2013년09월06일
(30) 우선권주장
13/777,785 2013년02월26일 미국(US)
61/605,136 2012년02월29일 미국(US)
(56) 선행기술조사문헌
KR1020110126103 A
(뒷면에 계속)

(73) 특허권자
켈컴 인코퍼레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
(72) 발명자
천 잉
미국 92121 캘리포니아주 샌디에고 모어하우스 드라이브 5775
왕 예-쿠이
미국 92121 캘리포니아주 샌디에고 모어하우스 드라이브 5775
카르체비츠 마르타
미국 92121 캘리포니아주 샌디에고 모어하우스 드라이브 5775
(74) 대리인
특허법인코리어나

전체 청구항 수 : 총 28 항

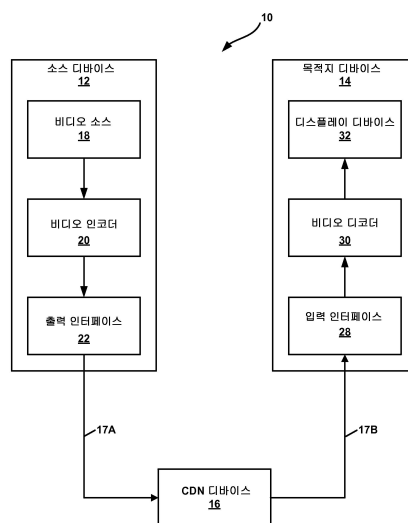
심사관 : 김웅권

(54) 발명의 명칭 3차원 비디오에서의 비트스트림 추출

(57) 요약

3차원 비디오(3DV) 비트스트림으로부터 서브-비트스트림을 추출하기 위해, 디바이스는 복수의 타겟 뷰들에서의 화상들을 디코딩하기 위해 요구되는 텍스처 뷰 성분들을 갖는 3DV 비트스트림에서의 뷰들을 나타내는 텍스처 타겟 뷰 리스트를 결정한다. 타겟 뷰들은, 비트스트림에서, 서브-비트스트림으로부터 디코딩가능하게 된 뷰들의 서브세트이다. 덧붙여서, 그 디바이스는 복수의 타겟 뷰들에서의 화상들을 디코딩하기 위해 요구되는 깊이 뷰 성분들을 갖는 3DV 비트스트림에서의 뷰들을 나타내는 깊이 타겟 뷰 리스트를 결정한다. 그 디바이스 텍스처 타겟 뷰 리스트 및 깊이 타겟 뷰 리스트에 적어도 부분적으로 기초하여 서브-비트스트림을 결정한다.

대표도 - 도1



(56) 선행기술조사문헌

KR1020120013966 A

US20110064146 A1

KR1020080063323 A

suzuki, T. et al, WD on MVC extensions for
inclusion of depth maps(2012.01.05. 공개)

명세서

청구범위

청구항 1

코딩된 텍스처 뷰 성분들 및 코딩된 깊이 뷰 성분들을 포함하는 3차원 비디오 (3DV) 비트스트림으로부터 서브-비트스트림을 추출하는 방법으로서,

복수의 타겟 뷰들에서의 화상들을 디코딩하기 위해 요구되는 텍스처 뷰 성분들을 갖는 상기 3DV 비트스트림에서의 뷰들을 나타내는 텍스처 타겟 뷰 리스트를 결정하는 단계;

상기 복수의 타겟 뷰들에서의 화상들을 디코딩하기 위해 요구되는 깊이 뷰 성분들을 갖는 상기 3DV 비트스트림에서의 뷰들을 나타내는 깊이 타겟 뷰 리스트를 결정하는 단계; 및

상기 텍스처 타겟 뷰 리스트 및 상기 깊이 타겟 뷰 리스트에 적어도 부분적으로 기초하여 상기 서브-비트스트림을 결정하는 단계를 포함하는, 3DV 비트스트림으로부터 서브-비트스트림을 추출하는 방법.

청구항 2

제 1 항에 있어서,

상기 텍스처 타겟 뷰 리스트는, 상기 복수의 타겟 뷰들에서의 화상들을 디코딩하기 위해 요구되는 텍스처 뷰 성분들을 갖는 상기 3DV 비트스트림에서의 상기 뷰들을 식별하는 뷰 식별자들을 포함하며,

상기 깊이 타겟 뷰 리스트는, 상기 복수의 타겟 뷰들에서의 화상들을 디코딩하기 위해 요구되는 깊이 뷰 성분들을 갖는 상기 3DV 비트스트림에서의 상기 뷰들을 식별하는 뷰 식별자들을 포함하고,

상기 텍스처 타겟 뷰 리스트는 상기 깊이 타겟 뷰 리스트에서의 뷰 식별자들과는 상이한 하나 이상의 뷰 식별자들을 포함하는, 3DV 비트스트림으로부터 서브-비트스트림을 추출하는 방법.

청구항 3

제 1 항에 있어서,

시퀀스 파라미터 세트 (SPS) 가 상기 SPS에 적용가능한 뷰들 및 뷰 성분들 사이의 잠재적 의존도들을 명시하며;

상기 텍스처 타겟 뷰 리스트를 결정하는 단계는, 상기 복수의 타겟 뷰들 및 상기 SPS에 의해 명시된 잠재적 의존도들에 적어도 부분적으로 기초하여, 상기 텍스처 타겟 뷰 리스트를 결정하는 단계를 포함하고;

상기 깊이 타겟 뷰 리스트를 결정하는 단계는, 상기 복수의 타겟 뷰들 및 상기 SPS에 의해 명시된 잠재적 의존도들에 적어도 부분적으로 기초하여, 상기 깊이 타겟 뷰 리스트를 결정하는 단계를 포함하는, 3DV 비트스트림으로부터 서브-비트스트림을 추출하는 방법.

청구항 4

제 1 항에 있어서,

상기 3DV 비트스트림은 일련의 네트워크 추상 계층 (NAL) 유닛들을 포함하고,

상기 서브-비트스트림을 결정하는 단계는, 상기 텍스처 타겟 뷰 리스트 또는 상기 깊이 타겟 뷰 리스트가 NAL 유닛의 뷰 식별자를 명시하는지 여부에 적어도 부분적으로 기초하여, 상기 3DV 비트스트림으로부터 상기 NAL 유닛을 제거할지 여부를 결정하는 단계를 포함하는, 3DV 비트스트림으로부터 서브-비트스트림을 추출하는 방법.

청구항 5

제 4 항에 있어서,

상기 텍스처 타겟 뷰 리스트 및 상기 깊이 타겟 뷰 리스트를 결정하는 단계는,

상기 복수의 타겟 뷰들에서의 앵커 화상들의 디코딩을 위해 요구되는 텍스처 뷰 성분들을 갖는 상기 3DV 비트스

트림에서의 뷰들을 나타내는 앵커 텍스처 타겟 뷰 리스트를 결정하는 단계;

상기 복수의 타겟 뷰들에서의 비 앵커 화상들의 디코딩을 위해 요구되는 텍스처 뷰 성분들을 갖는 상기 3DV 비트스트림에서의 뷰들을 나타내는 비 앵커 텍스처 타겟 뷰 리스트를 결정하는 단계;

상기 복수의 타겟 뷰들에서의 상기 앵커 화상들의 디코딩을 위해 요구되는 깊이 뷰 성분들을 갖는 상기 3DV 비트스트림에서의 뷰들을 나타내는 앵커 깊이 타겟 뷰 리스트를 결정하는 단계; 및

상기 복수의 타겟 뷰들에서의 상기 비 앵커 화상들의 디코딩을 위해 요구되는 깊이 뷰 성분들을 갖는 상기 3DV 비트스트림에서의 뷰들을 나타내는 비 앵커 깊이 타겟 뷰 리스트를 결정하는 단계를 포함하는, 3DV 비트스트림으로부터 서브-비트스트림을 추출하는 방법.

청구항 6

제 5 항에 있어서,

상기 3DV 비트스트림으로부터 상기 NAL 유닛을 제거할지 여부를 결정하는 단계는,

상기 NAL 유닛이 앵커 화상에 속하는지 여부, 상기 NAL 유닛이 깊이 뷰 성분 NAL 유닛 유형에 속하는지 여부, 및 상기 앵커 텍스처 타겟 뷰 리스트가 상기 NAL 유닛의 상기 뷰 식별자를 명시하는지 여부에 적어도 부분적으로 기초하여, 상기 3DV 비트스트림으로부터 상기 NAL 유닛을 제거할지 여부를 결정하는 단계; 및

상기 NAL 유닛이 앵커 화상에 속하는지 여부, 상기 NAL 유닛이 상기 깊이 뷰 성분 NAL 유닛 유형에 속하는지 여부, 및 상기 비 앵커 텍스처 타겟 뷰 리스트가 상기 NAL 유닛의 상기 뷰 식별자를 명시하는지 여부에 적어도 부분적으로 기초하여, 상기 3DV 비트스트림으로부터 상기 NAL 유닛을 제거할지 여부를 결정하는 단계를 포함하며,

상기 깊이 뷰 성분 NAL 유닛 유형에 속한 NAL 유닛들은 깊이 뷰 성분들에 대한 코딩된 슬라이스 확장들을 캡슐화하는, 3DV 비트스트림으로부터 서브-비트스트림을 추출하는 방법.

청구항 7

제 6 항에 있어서,

상기 3DV 비트스트림으로부터 상기 NAL 유닛을 제거할지 여부를 결정하는 단계는,

상기 NAL 유닛의 헤더에서의 앵커 화상 플래그 선택스 엘리먼트가 1과 동일하며, 상기 NAL 유닛의 상기 헤더에서의 NAL 유닛 유형 선택스 엘리먼트가 21과 동일하지 않고, 상기 NAL 유닛의 상기 헤더에서의 뷰 식별자 선택스 엘리먼트가 상기 앵커 텍스처 타겟 리스트에서의 뷰의 뷰 식별자와 동일한 경우, 상기 NAL 유닛은 상기 3DV 비트스트림으로부터 제거될 것이라고 결정하는 단계; 및

상기 NAL 유닛의 상기 헤더에서의 상기 앵커 화상 플래그 선택스 엘리먼트가 0과 동일하며, 상기 NAL 유닛의 상기 헤더에서의 상기 NAL 유닛 유형 선택스 엘리먼트가 21과 동일하지 않고, 상기 NAL 유닛의 상기 헤더에서의 상기 뷰 식별자 선택스 엘리먼트가 상기 비 앵커 텍스처 타겟 리스트에서의 뷰의 뷰 식별자와 동일한 경우, 상기 NAL 유닛은 상기 3DV 비트스트림으로부터 제거될 것이라고 결정하는 단계를 포함하는, 3DV 비트스트림으로부터 서브-비트스트림을 추출하는 방법.

청구항 8

제 5 항에 있어서,

상기 3DV 비트스트림으로부터 상기 NAL 유닛을 제거할지 여부를 결정하는 단계는,

상기 NAL 유닛이 앵커 화상에 속하는지 여부, 상기 NAL 유닛이 깊이 뷰 성분 NAL 유닛 유형에 속하는지 여부, 및 상기 앵커 깊이 타겟 뷰 리스트가 상기 NAL 유닛의 상기 뷰 식별자를 명시하는지 여부에 적어도 부분적으로 기초하여, 상기 3DV 비트스트림으로부터 상기 NAL 유닛을 제거할지 여부를 결정하는 단계; 및

상기 NAL 유닛이 앵커 화상에 속하는지 여부, 상기 NAL 유닛이 상기 깊이 뷰 성분 NAL 유닛 유형에 속하는지 여부, 및 상기 비 앵커 깊이 타겟 뷰 리스트가 상기 NAL 유닛의 상기 뷰 식별자를 명시하는지 여부에 적어도 부분적으로 기초하여, 상기 3DV 비트스트림으로부터 상기 NAL 유닛을 제거할지 여부를 결정하는 단계를 포함하며,

상기 깊이 뷰 성분 NAL 유닛 유형에 속한 NAL 유닛들은 깊이 뷰 성분들에 대한 코딩된 슬라이스 확장들을 캡슐

화하는, 3DV 비트스트림으로부터 서브-비트스트림을 추출하는 방법.

청구항 9

제 8 항에 있어서,

상기 3DV 비트스트림으로부터 상기 NAL 유닛을 제거할지 여부를 결정하는 단계는,

상기 NAL 유닛의 헤더에서의 앵커 화상 플래그 선택스 엘리먼트가 1과 동일하며, 상기 NAL 유닛의 상기 헤더에서의 NAL 유닛 유형 선택스 엘리먼트가 21과 동일하고, 상기 NAL 유닛의 상기 헤더에서의 뷰 식별자 선택스 엘리먼트가 상기 앵커 깊이 타겟 리스트에서의 뷰의 뷰 식별자와 동일한 경우, 상기 NAL 유닛은 상기 3DV 비트스트림으로부터 제거될 것이라고 결정하는 단계; 및

상기 NAL 유닛의 상기 헤더에서의 상기 앵커 화상 플래그 선택스 엘리먼트가 0과 동일하며, 상기 NAL 유닛의 상기 헤더에서의 상기 NAL 유닛 유형 선택스 엘리먼트가 21과 동일하고, 상기 NAL 유닛의 상기 헤더에서의 상기 뷰 식별자 선택스 엘리먼트가 상기 비 앵커 깊이 타겟 리스트에서의 뷰의 뷰 식별자와 동일한 경우, 상기 NAL 유닛은 상기 3DV 비트스트림으로부터 제거될 것이라고 결정하는 단계를 포함하는, 3DV 비트스트림으로부터 서브-비트스트림을 추출하는 방법.

청구항 10

제 1 항에 있어서,

상기 방법은 콘텐츠 전달 네트워크 (CDN) 디바이스에 의해 수행되는, 3DV 비트스트림으로부터 서브-비트스트림을 추출하는 방법.

청구항 11

하나 이상의 프로세서들을 포함하는 디바이스로서,

상기 하나 이상의 프로세서들은,

코딩된 텍스처 뷰 성분들 및 코딩된 깊이 뷰 성분들을 포함하는 3차원 비디오 (3DV) 비트스트림에서 복수의 타겟 뷰들에서의 화상들을 디코딩하기 위해 요구되는 텍스처 뷰 성분들을 갖는 뷰들을 나타내는 텍스처 타겟 뷰 리스트를 결정하며;

상기 복수의 타겟 뷰들에서의 화상들을 디코딩하기 위해 요구되는 깊이 뷰 성분들을 갖는 상기 3DV 비트스트림에서의 뷰들을 나타내는 깊이 타겟 뷰 리스트를 결정하고;

상기 텍스처 타겟 뷰 리스트 및 상기 깊이 타겟 뷰 리스트에 적어도 부분적으로 기초하여 서브-비트스트림을 결정하도록 구성되는, 하나 이상의 프로세서들을 포함하는 디바이스.

청구항 12

제 11 항에 있어서,

상기 텍스처 타겟 뷰 리스트는, 상기 복수의 타겟 뷰들에서의 화상들을 디코딩하기 위해 요구되는 텍스처 뷰 성분들을 갖는 상기 3DV 비트스트림에서의 상기 뷰들을 식별하는 뷰 식별자들을 포함하며,

상기 깊이 타겟 뷰 리스트는, 상기 복수의 타겟 뷰들에서의 화상들을 디코딩하기 위해 요구되는 깊이 뷰 성분들을 갖는 상기 3DV 비트스트림에서의 상기 뷰들을 식별하는 뷰 식별자들을 포함하고,

상기 텍스처 타겟 뷰 리스트는 상기 깊이 타겟 뷰 리스트에서의 상기 뷰 식별자들과는 상이한 하나 이상의 뷰 식별자들을 포함하는, 하나 이상의 프로세서들을 포함하는 디바이스.

청구항 13

제 11 항에 있어서,

시퀀스 파라미터 세트 (SPS) 가 상기 SPS에 적용가능한 뷰들 및 뷰 성분들 사이의 잠재적 의존도들을 명시하고;

상기 하나 이상의 프로세서들은,

상기 복수의 타겟 뷰들 및 상기 SPS에 의해 명시된 상기 잠재적 의존도들에 적어도 부분적으로 기초하여, 상기 텍스처 타겟 뷰 리스트를 결정하고;

상기 복수의 타겟 뷰들 및 상기 SPS에 의해 명시된 상기 잠재적 의존도들에 적어도 부분적으로 기초하여, 상기 깊이 타겟 뷰 리스트를 결정하도록 구성되는, 하나 이상의 프로세서들을 포함하는 디바이스.

청구항 14

제 11 항에 있어서,

상기 3DV 비트스트림은 일련의 네트워크 추상 계층 (NAL) 유닛들을 포함하고,

상기 하나 이상의 프로세서들은, 상기 텍스처 타겟 뷰 리스트 또는 상기 깊이 타겟 뷰 리스트가 NAL 유닛의 뷰 식별자를 명시하는지 여부에 적어도 부분적으로 기초하여, 상기 3DV 비트스트림으로부터 상기 NAL 유닛을 제거할지 여부를 결정하도록 구성되는, 하나 이상의 프로세서들을 포함하는 디바이스.

청구항 15

제 14 항에 있어서,

상기 하나 이상의 프로세서들은,

상기 복수의 타겟 뷰들에서의 앵커 화상들의 디코딩을 위해 요구되는 텍스처 뷰 성분들을 갖는 상기 3DV 비트스트림에서의 뷰들을 나타내는 앵커 텍스처 타겟 뷰 리스트를 결정하고;

상기 복수의 타겟 뷰들에서의 비 앵커 화상들의 디코딩을 위해 요구되는 텍스처 뷰 성분들을 갖는 상기 3DV 비트스트림에서의 뷰들을 나타내는 비 앵커 텍스처 타겟 뷰 리스트를 결정하며;

상기 복수의 타겟 뷰들에서의 상기 앵커 화상들의 디코딩을 위해 요구되는 깊이 뷰 성분들을 갖는 상기 3DV 비트스트림에서의 뷰들을 나타내는 앵커 깊이 타겟 뷰 리스트를 결정하고;

상기 복수의 타겟 뷰들에서의 상기 비 앵커 화상들의 디코딩을 위해 요구되는 깊이 뷰 성분들을 갖는 상기 3DV 비트스트림에서의 뷰들을 나타내는 비 앵커 깊이 타겟 뷰 리스트를 결정하도록 구성되는, 하나 이상의 프로세서들을 포함하는 디바이스.

청구항 16

제 15 항에 있어서,

상기 하나 이상의 프로세서들은,

상기 NAL 유닛이 앵커 화상에 속하는지 여부, 상기 NAL 유닛이 깊이 뷰 성분 NAL 유닛 유형에 속하는지 여부, 및 상기 앵커 텍스처 타겟 뷰 리스트가 상기 NAL 유닛의 상기 뷰 식별자를 명시하는지 여부에 적어도 부분적으로 기초하여, 상기 3DV 비트스트림으로부터 상기 NAL 유닛을 제거할지 여부를 결정하고;

상기 NAL 유닛이 앵커 화상에 속하는지 여부, 상기 NAL 유닛이 상기 깊이 뷰 성분 NAL 유닛 유형에 속하는지 여부, 및 상기 비 앵커 텍스처 타겟 뷰 리스트가 상기 NAL 유닛의 상기 뷰 식별자를 명시하는지 여부에 적어도 부분적으로 기초하여, 상기 3DV 비트스트림으로부터 상기 NAL 유닛을 제거할지 여부를 결정하도록 구성되며,

상기 깊이 뷰 성분 NAL 유닛 유형에 속한 NAL 유닛들은 깊이 뷰 성분들에 대한 코딩된 슬라이스 확장들을 캡슐화하는, 하나 이상의 프로세서들을 포함하는 디바이스.

청구항 17

제 16 항에 있어서,

상기 하나 이상의 프로세서들은,

상기 NAL 유닛의 헤더에서의 앵커 화상 플래그 선택스 엘리먼트가 1과 동일하며, 상기 NAL 유닛의 상기 헤더에서의 NAL 유닛 유형 선택스 엘리먼트가 21과 동일하지 않고, 상기 NAL 유닛의 상기 헤더에서의 뷰 식별자 선택스 엘리먼트가 상기 앵커 텍스처 타겟 리스트에서의 뷰의 뷰 식별자와 동일한 경우, 상기 NAL 유닛은 상기 3DV 비트스트림으로부터 제거될 것이라고 결정하고;

상기 NAL 유닛의 상기 헤더에서의 상기 앵커 화상 플래그 선택스 엘리먼트가 0과 동일하며, 상기 NAL 유닛의 상기 헤더에서의 상기 NAL 유닛 유형 선택스 엘리먼트가 21과 동일하지 않고, 상기 NAL 유닛의 상기 헤더에서의 상기 뷰 식별자 선택스 엘리먼트가 상기 비 앵커 텍스처 타겟 리스트에서의 뷰의 뷰 식별자와 동일한 경우, 상기 NAL 유닛은 상기 3DV 비트스트림으로부터 제거될 것이라고 결정하도록 구성되는, 하나 이상의 프로세서들을 포함하는 디바이스.

청구항 18

제 15 항에 있어서,

상기 하나 이상의 프로세서들은,

상기 NAL 유닛이 앵커 화상에 속하는지 여부, 상기 NAL 유닛이 깊이 뷰 성분 NAL 유닛 유형에 속하는지 여부, 및 상기 앵커 깊이 타겟 뷰 리스트가 상기 NAL 유닛의 상기 뷰 식별자를 명시하는지 여부에 적어도 부분적으로 기초하여, 상기 3DV 비트스트림으로부터 상기 NAL 유닛을 제거할지 여부를 결정하고;

상기 NAL 유닛이 앵커 화상에 속하는지 여부, 상기 NAL 유닛이 상기 깊이 뷰 성분 NAL 유닛 유형에 속하는지 여부, 및 상기 비 앵커 깊이 타겟 뷰 리스트가 상기 NAL 유닛의 상기 뷰 식별자를 명시하는지 여부에 적어도 부분적으로 기초하여, 상기 3DV 비트스트림으로부터 상기 NAL 유닛을 제거할지 여부를 결정하도록 구성되며,

상기 깊이 뷰 성분 NAL 유닛 유형에 속한 NAL 유닛들은 깊이 뷰 성분들에 대한 코딩된 슬라이스 확장들을 캡슐화하는, 하나 이상의 프로세서들을 포함하는 디바이스.

청구항 19

제 18 항에 있어서,

상기 하나 이상의 프로세서들은,

상기 NAL 유닛의 헤더에서의 앵커 화상 플래그 선택스 엘리먼트가 1과 동일하며, 상기 NAL 유닛의 상기 헤더에서의 NAL 유닛 유형 선택스 엘리먼트가 21과 동일하고, 상기 NAL 유닛의 상기 헤더에서의 뷰 식별자 선택스 엘리먼트가 상기 앵커 깊이 타겟 리스트에서의 뷰의 뷰 식별자와 동일한 경우, 상기 NAL 유닛은 상기 3DV 비트스트림으로부터 제거될 것이라고 결정하고;

상기 NAL 유닛의 상기 헤더에서의 상기 앵커 화상 플래그 선택스 엘리먼트가 0과 동일하며, 상기 NAL 유닛의 상기 헤더에서의 상기 NAL 유닛 유형 선택스 엘리먼트가 21과 동일하고, 상기 NAL 유닛의 상기 헤더에서의 상기 뷰 식별자 선택스 엘리먼트가 상기 비 앵커 깊이 타겟 리스트에서의 뷰의 뷰 식별자와 동일한 경우, 상기 NAL 유닛은 상기 3DV 비트스트림으로부터 제거될 것이라고 결정하도록 구성되는, 하나 이상의 프로세서들을 포함하는 디바이스.

청구항 20

제 11 항에 있어서,

상기 디바이스는 콘텐츠 전달 네트워크 (CDN) 디바이스인, 하나 이상의 프로세서들을 포함하는 디바이스.

청구항 21

복수의 타겟 뷰들에서의 화상들을 디코딩하기 위해 요구되는 텍스처 뷰 성분들을 갖는 3차원 비디오 (3DV) 비트스트림에서의 뷰들을 나타내는 텍스처 타겟 뷰 리스트를 결정하는 수단으로서, 상기 3DV 비트스트림은 코딩된 텍스처 뷰 성분들 및 코딩된 깊이 뷰 성분들을 포함하는, 상기 텍스처 타겟 뷰 리스트를 결정하는 수단;

상기 복수의 타겟 뷰들에서의 화상들을 디코딩하기 위해 요구되는 깊이 뷰 성분들을 갖는 상기 3DV 비트스트림에서의 뷰들을 나타내는 깊이 타겟 뷰 리스트를 결정하는 수단; 및

상기 텍스처 타겟 뷰 리스트 및 상기 깊이 타겟 뷰 리스트에 적어도 부분적으로 기초하여 서브-비트스트림을 결정하는 수단을 포함하는, 디바이스.

청구항 22

제 21 항에 있어서,

상기 3DV 비트스트림은 일련의 네트워크 추상 계층 (NAL) 유닛들을 포함하고,

상기 서브-비트스트림을 결정하는 수단은, 상기 텍스처 타겟 뷰 리스트 또는 상기 깊이 타겟 뷰 리스트가 NAL 유닛의 뷰 식별자를 명시하는지 여부에 적어도 부분적으로 기초하여, 상기 3DV 비트스트림으로부터 상기 NAL 유닛을 제거할지 여부를 결정하는 수단을 포함하는, 디바이스.

청구항 23

제 22 항에 있어서,

상기 텍스처 타겟 뷰 리스트를 결정하는 수단은,

상기 복수의 타겟 뷰들에서의 앵커 화상들의 디코딩을 위해 요구되는 텍스처 뷰 성분들을 갖는 상기 3DV 비트스트림에서의 뷰들을 나타내는 앵커 텍스처 타겟 뷰 리스트를 결정하는 수단;

상기 복수의 타겟 뷰들에서의 비 앵커 화상들의 디코딩을 위해 요구되는 텍스처 뷰 성분들을 갖는 상기 3DV 비트스트림에서의 뷰들을 나타내는 비 앵커 텍스처 타겟 뷰 리스트를 결정하는 수단;

상기 복수의 타겟 뷰들에서의 상기 앵커 화상들의 디코딩을 위해 요구되는 깊이 뷰 성분들을 갖는 상기 3DV 비트스트림에서의 뷰들을 나타내는 앵커 깊이 타겟 뷰 리스트를 결정하는 수단; 및

상기 복수의 타겟 뷰들에서의 상기 비 앵커 화상들의 디코딩을 위해 요구되는 깊이 뷰 성분들을 갖는 상기 3DV 비트스트림에서의 뷰들을 나타내는 비 앵커 깊이 타겟 뷰 리스트를 결정하는 수단을 포함하는, 디바이스.

청구항 24

제 23 항에 있어서,

상기 3DV 비트스트림으로부터 상기 NAL 유닛을 제거할지 여부를 결정하는 수단은,

상기 NAL 유닛이 앵커 화상에 속하는지 여부, 상기 NAL 유닛이 깊이 뷰 성분 NAL 유닛 유형에 속하는지 여부, 및 상기 앵커 텍스처 타겟 뷰 리스트가 상기 NAL 유닛의 상기 뷰 식별자를 명시하는지 여부에 적어도 부분적으로 기초하여, 상기 3DV 비트스트림으로부터 상기 NAL 유닛을 제거할지 여부를 결정하는 수단; 및

상기 NAL 유닛이 앵커 화상에 속하는지 여부, 상기 NAL 유닛이 상기 깊이 뷰 성분 NAL 유닛 유형에 속하는지 여부, 및 상기 비 앵커 텍스처 타겟 뷰 리스트가 상기 NAL 유닛의 상기 뷰 식별자를 명시하는지 여부에 적어도 부분적으로 기초하여, 상기 3DV 비트스트림으로부터 상기 NAL 유닛을 제거할지 여부를 결정하는 수단을 포함하고,

상기 깊이 뷰 성분 NAL 유닛 유형에 속한 NAL 유닛들은 깊이 뷰 성분들에 대한 코딩된 슬라이스 확장들을 캡슐화하는, 디바이스.

청구항 25

명령들을 저장하고 있는 컴퓨터 판독가능 저장 매체로서,

상기 명령들은, 디바이스의 하나 이상의 프로세서들에 의해 실행되는 경우,

코딩된 텍스처 뷰 성분들 및 코딩된 깊이 뷰 성분들을 포함하는 3DV 비트스트림에서 복수의 타겟 뷰들에서의 화상들을 디코딩하기 위해 요구되는 텍스처 뷰 성분들을 갖는 뷰들을 나타내는 텍스처 타겟 뷰 리스트를 결정하고;

상기 복수의 타겟 뷰들에서의 화상들을 디코딩하기 위해 요구되는 깊이 뷰 성분들을 갖는 상기 3DV 비트스트림에서의 뷰들을 나타내는 깊이 타겟 뷰 리스트를 결정하며;

상기 텍스처 타겟 뷰 리스트 및 상기 깊이 타겟 뷰 리스트에 적어도 부분적으로 기초하여 서브-비트스트림을 결정하도록

상기 디바이스를 구성하는, 컴퓨터 판독가능 저장 매체.

청구항 26

제 25 항에 있어서,

상기 3DV 비트스트림은 일련의 네트워크 추상 계층 (NAL) 유닛들을 포함하고,

상기 명령들은, 상기 텍스처 타겟 뷰 리스트 또는 상기 깊이 타겟 뷰 리스트가 NAL 유닛의 뷰 식별자를 명시하는지 여부에 적어도 부분적으로 기초하여, 상기 3DV 비트스트림으로부터 상기 NAL 유닛을 제거할지 여부를 결정하도록 상기 디바이스를 구성하는, 컴퓨터 판독가능 저장 매체.

청구항 27

제 26 항에 있어서,

상기 명령들은,

상기 복수의 타겟 뷰들에서의 앵커 화상들의 디코딩을 위해 요구되는 텍스처 뷰 성분들을 갖는 상기 3DV 비트스트림에서의 뷰들을 나타내는 앵커 텍스처 타겟 뷰 리스트를 결정하고;

상기 복수의 타겟 뷰들에서의 비 앵커 화상들의 디코딩을 위해 요구되는 텍스처 뷰 성분들을 갖는 상기 3DV 비트스트림에서의 뷰들을 나타내는 비 앵커 텍스처 타겟 뷰 리스트를 결정하며;

상기 복수의 타겟 뷰들에서의 상기 앵커 화상들의 디코딩을 위해 요구되는 깊이 뷰 성분들을 갖는 상기 3DV 비트스트림에서의 뷰들을 나타내는 앵커 깊이 타겟 뷰 리스트를 결정하고;

상기 복수의 타겟 뷰들에서의 상기 비 앵커 화상들의 디코딩을 위해 요구되는 깊이 뷰 성분들을 갖는 상기 3DV 비트스트림에서의 뷰들을 나타내는 비 앵커 깊이 타겟 뷰 리스트를 결정하도록

상기 디바이스를 구성하는, 컴퓨터 판독가능 저장 매체.

청구항 28

제 27 항에 있어서,

상기 명령들은,

상기 NAL 유닛이 앵커 화상에 속하는지 여부, 상기 NAL 유닛이 깊이 뷰 성분 NAL 유닛 유형에 속하는지 여부, 및 상기 앵커 텍스처 타겟 뷰 리스트가 상기 NAL 유닛의 상기 뷰 식별자를 명시하는지 여부에 적어도 부분적으로 기초하여, 상기 3DV 비트스트림으로부터 상기 NAL 유닛을 제거할지 여부를 결정하고;

상기 NAL 유닛이 앵커 화상에 속하는지 여부, 상기 NAL 유닛이 상기 깊이 뷰 성분 NAL 유닛 유형에 속하는지 여부, 및 상기 비 앵커 텍스처 타겟 뷰 리스트가 상기 NAL 유닛의 상기 뷰 식별자를 명시하는지 여부에 적어도 부분적으로 기초하여, 상기 3DV 비트스트림으로부터 상기 NAL 유닛을 제거할지 여부를 결정하도록

상기 디바이스를 구성하며,

상기 깊이 뷰 성분 NAL 유닛 유형에 속한 NAL 유닛들은 깊이 뷰 성분들에 대한 코딩된 슬라이스 확장들을 캡슐화하는, 컴퓨터 판독가능 저장 매체.

발명의 설명

기술 분야

[0001] 본 출원은 2012년 2월 29일자로 출원된 미국 특허 가출원 제61/605,136호를 우선권 주장하며, 그 전체 내용은 참조로 본원에 통합된다.

[0002] 기술 분야

[0003] 본 개시물은 비디오 코딩 (즉, 비디오 데이터의 인코딩 및/또는 디코딩) 에 관한 것이다.

배경 기술

[0004] 디지털 비디오 능력들은 디지털 텔레비전들, 디지털 직접 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인휴대 정보단말들 (PDA들), 랩톱 또는 데스크톱 컴퓨터들, 태블릿 컴퓨터들, e-북 리더들, 디지털 카메라들, 디지털 레코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 무선 전화기들, 이른바 "스마트 폰들", 비디오 원격회의 디바이스들, 비디오 스트리밍 디바이스들 등

을 포함하는 넓은 범위의 디바이스들에 통합될 수 있다. 디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263 또는 ITU-T H.264/MPEG-4, 파트 10, 고급 비디오 코딩 (AVC) 에 의해 규정된 표준들, 현재 개발중이고 효율 비디오 코딩 (High Efficiency Video Coding, HEVC) 표준, 및 이러한 표준들의 확장본들에 기재된 것들과 같은 비디오 압축 기법들을 구현한다. 비디오 디바이스들은 그런 비디오 압축 기법들을 구현하는 것에 의해 디지털 비디오 정보를 더 효율적으로 송신, 수신, 인코딩, 디코딩, 및/또는 저장할 수도 있다.

[0005] 비디오 압축 기법들은 공간적 (화상 내) 예측 및/또는 시간적 (화상 간) 예측을 수행하여 비디오 시퀀스들에 내재하는 리던던시를 감소시키거나 제거한다. 블록 기반 비디오 코딩의 경우, 비디오 슬라이스 (즉, 비디오 프레임 또는 비디오 프레임의 부분) 는 비디오 블록들로 구획될 수도 있다. 화상의 인트라 코딩식 (intra-coded; I) 슬라이스에서의 비디오 블록들은 동일한 화상의 이웃 블록들에서의 참조 샘플들에 관한 공간적 예측을 이용하여 인코딩된다. 화상의 인터 코딩식 (inter-coded; P 또는 B) 슬라이스에서의 비디오 블록들은 동일한 화상의 이웃 블록들에서의 참조 샘플들에 관한 공간적 예측 또는 다른 참조 화상들에서의 참조 샘플들에 관한 시간적 예측을 이용할 수도 있다. 화상들은 프레임들이라고 지칭될 수도 있고, 참조 화상들은 참조 프레임들이라고 말해질 수도 있다.

[0006] 공간적 또는 시간적 예측은 코딩될 블록에 대한 예측 블록으로 나타나게 된다. 잔차 데이터는 코딩될 원본 블록과 예측 블록 사이의 화소 차이들을 나타낸다. 인터 코딩식 블록은 예측 블록을 형성하는 참조 샘플들의 블록을 가리키는 모션 벡터에 따라 인코딩되고, 잔차 데이터는 코딩된 블록 및 예측 블록 사이의 차이를 나타낸다. 인트라 코딩식 블록은 인트라 코딩 모드 및 잔차 데이터에 따라 인코딩된다. 추가 압축을 위해, 잔차 데이터는 화소 도메인으로부터 변환 도메인으로 변환될 수도 있으며, 결과적으로 잔차 계수들이 생겨나며, 그 계수들은 그 다음에 양자화될 수도 있다. 처음에는 2차원 어레이로 배열된 양자화된 계수들은, 계수들의 1차원 벡터를 생성하기 위하여 스캐닝될 수도 있고, 엔트로피 코딩이 더 많은 압축을 달성하기 위해 스캔된 양자화된 계수들에 적용될 수도 있다.

[0007] 멀티-뷰 비트스트림이, 예컨대, 다수의 컬러 카메라들로부터의 뷰들을 인코딩함으로써 생성될 수도 있다. 멀티-뷰 비디오의 유연성을 추가로 확장하기 위해, 3차원 (3D) 비디오 표준들이 개발되어왔다. 3D 비디오 비트스트림은, 다수의 카메라들에 대응하는 뷰들, 즉 텍스처 뷰들뿐만 아니라, 적어도 하나 이상의 텍스처 뷰들에 연관된 깊이 뷰들도 포함할 수도 있다. 예를 들어, 각각의 뷰는 하나의 텍스처 뷰 및 하나의 깊이 뷰로 구성될 수도 있다.

발명의 내용

해결하려는 과제

과제의 해결 수단

[0008] 대체로, 본 개시물은 3차원 비디오 (3DV) 비트스트림으로부터 비디오 데이터 서브-비트스트림을 추출하는 기법들을 설명한다. 더 구체적으로는, 디바이스가, 복수의 타겟 뷰들에서의 화상들을 디코딩하기 위해 요구되는 텍스처 뷰 성분들을 갖는 3DV 비트스트림에서의 뷰들을 나타내는 텍스처 타겟 뷰 리스트를 결정한다. 타겟 뷰들은, 비트스트림에서, 서브-비트스트림으로부터 디코딩가능하게 된 뷰들의 서브세트이다. 덧붙여서, 그 디바이스는 복수의 타겟 뷰들에서의 화상들을 디코딩하기 위해 요구되는 깊이 뷰 성분들을 갖는 3DV 비트스트림에서의 뷰들을 나타내는 깊이 타겟 뷰 리스트를 결정한다. 그 디바이스 텍스처 타겟 뷰 리스트 및 깊이 타겟 뷰 리스트에 적어도 부분적으로 기초하여 서브-비트스트림을 결정한다.

[0009] 본 개시물의 기법들에 따라, 심지어 깊이 뷰에 연관된 텍스처 뷰가 텍스처 타겟 뷰 리스트에 있더라도 그 깊이 뷰는 깊이 타겟 뷰 리스트로부터 제외될 수도 있다. 마찬가지로, 본 개시물의 기법들은, 심지어 텍스처 뷰에 연관된 깊이 뷰가, 이용가능한 경우, 깊이 타겟 뷰 리스트에 있더라도, 텍스처 뷰가, 이용가능한 경우, 텍스처 타겟 뷰 리스트로부터 제외되는 것을 가능하게 할 수도 있다. 텍스처 뷰와 깊이 뷰는 그것들이 동일한 카메라 로케이션에 대응하면, 즉, MVC+D 또는 3D-AVC와 같은 3D 비디오 코덱들에서 뷰 식별자 (view_id) 의 동일한 값을 갖는다면, 연관된 것으로 간주될 수도 있다.

[0010] 하나의 예에서, 본 개시물은 코딩된 텍스처 뷰 성분들 및 코딩된 깊이 뷰 성분들을 포함하는 3DV 비트스트림으로부터 서브-비트스트림을 추출하는 방법을 설명한다. 그 방법은 복수의 타겟 뷰들에서의 화상들을 디코딩하기 위해 요구되는 텍스처 뷰 성분들을 갖는 3DV 비트스트림에서의 뷰들을 나타내는 텍스처 타겟 뷰 리스트를

결정하는 단계를 포함한다. 덧붙여서, 그 방법은 복수의 타겟 뷰들에서의 화상들을 디코딩하기 위해 요구되는 깊이 뷰 성분들을 갖는 3DV 비트스트림에서의 뷰들을 나타내는 깊이 타겟 뷰 리스트를 결정하는 단계를 포함한다. 더욱이, 그 방법은 텍스처 타겟 뷰 리스트 및 깊이 타겟 뷰 리스트에 적어도 부분적으로 기초하여 서브-비트스트림을 결정하는 단계를 포함한다.

[0011] 다른 예에서, 본 개시물은 복수의 타겟 뷰들에서의 화상들을 디코딩하기 위해 요구되는 텍스처 뷰 성분들을 갖는 3DV 비트스트림에서의 뷰들을 나타내는 텍스처 타겟 뷰 리스트를 결정하도록 구성된 하나 이상의 프로세서들을 포함하며, 3DV 비트스트림은 코딩된 텍스처 뷰 성분들 및 코딩된 깊이 뷰 성분들을 포함한다. 하나 이상의 프로세서들은 또한 복수의 타겟 뷰들에서의 화상들을 디코딩하기 위해 요구되는 깊이 뷰 성분들을 갖는 3DV 비트스트림에서의 뷰들을 나타내는 깊이 타겟 뷰 리스트를 결정하도록 구성된다. 더욱이, 하나 이상의 프로세서들은 텍스처 타겟 뷰 리스트 및 깊이 타겟 뷰 리스트에 적어도 부분적으로 기초하여 서브-비트스트림을 결정할 수도 있다.

[0012] 다른 예에서, 본 개시물은, 복수의 타겟 뷰들에서의 화상들을 디코딩하기 위해 요구되는 텍스처 뷰 성분들을 갖는 3DV 비트스트림에서의 뷰들을 나타내는 텍스처 타겟 뷰 리스트를 결정하는 수단을 포함하며 3DV 비트스트림은 코딩된 텍스처 뷰 성분들 및 코딩된 깊이 뷰 성분들을 포함하는 디바이스를 설명한다. 그 디바이스는 또한 복수의 타겟 뷰들에서의 화상들을 디코딩하기 위해 요구되는 깊이 뷰 성분들을 갖는 3DV 비트스트림에서의 뷰들을 나타내는 깊이 타겟 뷰 리스트를 결정하는 수단을 포함한다. 덧붙여서, 그 디바이스는 텍스처 타겟 뷰 리스트 및 깊이 타겟 뷰 리스트에 적어도 부분적으로 기초하여 서브-비트스트림을 결정하는 수단을 포함한다.

[0013] 다른 예에서, 본 개시물은, 디바이스의 하나 이상의 프로세서들에 의해 실행되는 경우, 복수의 타겟 뷰들에서의 화상들을 디코딩하기 위해 요구되는 텍스처 뷰 성분들을 갖는 3DV 비트스트림에서의 뷰들을 나타내는 텍스처 타겟 뷰 리스트를 결정하도록 상기 디바이스를 구성하는 명령들을 저장하며 3DV 비트스트림은 코딩된 텍스처 뷰 성분들 및 코딩된 깊이 뷰 성분들을 포함하는 컴퓨터 판독가능 저장 매체를 설명한다. 덧붙여서, 그 명령들은 복수의 타겟 뷰들에서의 화상들을 디코딩하기 위해 요구되는 깊이 뷰 성분들을 갖는 3DV 비트스트림에서의 뷰들을 나타내는 깊이 타겟 뷰 리스트를 결정하도록 디바이스를 구성한다. 더욱이, 그 명령들은 텍스처 타겟 뷰 리스트 및 깊이 타겟 뷰 리스트에 적어도 부분적으로 기초하여 서브-비트스트림을 결정하도록 디바이스를 구성한다.

[0014] 본 개시물의 하나 이상의 예들의 상세는 첨부 도면들 및 다음의 설명에서 언급된다. 다른 특징들, 목적들, 및 장점들은 상세한 설명 및 도면들로부터, 그리고 청구항들로부터 명확하게 될 것이다.

도면의 간단한 설명

[0015] 도 1은 본 개시물에서 설명되는 기법들을 활용할 수도 있는 일 예의 비디오 코딩 시스템을 도시하는 블록도이다.

도 2는 본 개시물에서 설명되는 기법들을 구현할 수도 있는 일 예의 비디오 인코더를 도시하는 블록도이다.

도 3은 본 개시물에서 설명되는 기법들을 구현할 수도 있는 일 예의 비디오 디코더를 도시하는 블록도이다.

도 4는 본 개시물의 하나 이상의 기법들에 따른, 일 예의 서브-비트스트림 추출 동작을 도시하는 흐름도이다.

도 5는 본 개시물의 하나 이상의 기법들에 따른, 멀티 뷰 코딩 (MVC) 호환 3차원 비디오 (3DV) 에서의 일 예의 서브-비트스트림 추출 프로세스를 도시하는 흐름도이다.

도 6은 도 5의 예의 서브-비트스트림 추출 프로세스의 연속을 도시하는 흐름도이다.

도 7은 본 개시물의 하나 이상의 기법들에 따른, 요청된 앵커 텍스처 뷰 성분들의 뷰 식별자들을 결정하는 일 예의 동작을 도시하는 흐름도이다.

도 8은 본 개시물의 하나 이상의 기법들에 따른, 요청된 앵커 깊이 뷰 성분들의 뷰 식별자들을 결정하는 일 예의 동작을 도시하는 흐름도이다.

도 9는 본 개시물의 하나 이상의 기법들에 따른, 요청된 비 앵커 (non-anchor) 텍스처 뷰 성분들의 뷰 식별자들을 결정하는 일 예의 동작을 도시하는 흐름도이다.

도 10은 본 개시물의 하나 이상의 기법들에 따른, 요청된 비 앵커 깊이 뷰 성분들의 뷰 식별자들을 결정하는 일

예의 동작을 도시하는 흐름도이다.

도 11은 본 개시물의 하나 이상의 기법들에 따른, 비디오 코딩 계층 (VCL) 네트워크 액세스 계층 (NAL) 유닛들 및 필러 (filler) 데이터 NAL 유닛들을 비트스트림으로부터 제거될 것으로서 마킹하는 제 1 예 동작을 도시하는 흐름도이다.

도 12는 본 개시물의 하나 이상의 기법들에 따른, VCL NAL 유닛들 및 필러 데이터 NAL 유닛들을 비트스트림으로부터 제거될 것으로서 마킹하는 제 2 예 동작을 도시하는 흐름도이다.

도 13은 일 예의 MVC 디코딩 순서를 도시하는 개념도이다.

도 14는 일 예의 MVC 시간적 및 뷰간 예측 구조를 도시하는 개념도이다.

발명을 실시하기 위한 구체적인 내용

[0016] 비트스트림이 인코딩된 멀티 뷰 비디오 코딩 (multi-view video coding; MVC) 비디오 데이터를 포함할 수도 있다. MVC 비디오 데이터는 장면의 다수의 뷰들을 정의하는 데이터를 포함할 수도 있다. 3차원 (3D) 비디오 데이터의 멀티뷰 플러스 깊이 코딩을 사용하여 3차원 비디오 (three-dimensional video; 3DV) 를 지원하기 위해, 뷰들 중 적어도 일부는 텍스처 컴포넌트 및 깊이 컴포넌트를 포함할 수도 있다. 멀티뷰 플러스 깊이 코딩 프로세스의 일 예가 ITU-T H.264 표준 하의 MVC의 확장본으로서 개발 중이다. 이 확장본은 H.264/AVC 의 3DV 확장본이라고 지칭될 수도 있다. 비디오 디코더가 3DV를 출력하기 위해 비트스트림을 디코딩할 수도 있다. 비트스트림은 각각이 인코딩된 MVC 비디오 데이터의 부분들을 포함하는 일련의 네트워크 추상 계층 (NAL) 유닛들을 포함할 수도 있다. 인코딩된 멀티-뷰 비디오 데이터 내의 추가적인 뷰들의 포함은 비트스트림의 비트 레이트를 상당히 증가시킬 수도 있다.

[0017] 비트스트림을 요구하는 일부 컴퓨팅 디바이스들은 멀티 뷰 비디오 데이터에 포함된 뷰들의 전부를 핸들링하도록 구성되지 않는다. 더구나, 비트스트림을 요구하는 컴퓨팅 디바이스에 전체 비트스트림을 송신하는 대역폭은 불충분할 수도 있다. 따라서, 컴퓨팅 디바이스가 비트스트림을 요청하는 경우, 중간 디바이스, 이를테면 콘텐츠 전달 네트워크 (content delivery network; CDN) 에서의 디바이스는, 비트스트림으로부터 서브-비트스트림을 추출할 수도 있다. 다르게 말하면, 중간 디바이스는 원래의 비트스트림으로부터 서브-비트스트림을 추출하기 위해 서브-비트스트림 추출 프로세스를 수행할 수도 있다. 일부 예들에서, 중간 디바이스는 원래의 비트스트림으로부터 NAL 유닛들을 선택적으로 제거함으로써 서브-비트스트림을 추출한다. 다르게 말하면, 추출은, NAL 유닛들이 제거되고 버려져서 남아있는 제거되지 않은 NAL 유닛들이 서브-비트스트림이 되는 프로세스이다. 그 결과, 서브-비트스트림은 원래의 비트스트림보다 더 적은 뷰들을 포함할 수도 있다. 서브-비트스트림에 포함될 뷰들은 "타겟 뷰들"이라고 지칭될 수도 있다. 서브-비트스트림을 추출하기 위해, 중간 디바이스는 원래의 비트스트림으로부터 일부 뷰들을 제거하며, 이에 의해 남아있는 뷰들을 갖는 서브-비트스트림을 생성할 수도 있다. 더구나, 뷰들이 비트스트림으로부터 제거된다는 것을 감안하면, 서브-비트스트림의 비트 레이트는 원래의 비트스트림의 비트 레이트보다 적으며, 이에 의해 CDN을 통해 비트스트림을 요청하는 디바이스로 전송되는 경우에 보다 적은 대역폭을 소비할 수도 있다.

[0018] 어떤 경우들에서는, 비디오 데이터의 각각의 액세스 유닛은 뷰들 중 각각의 뷰의 텍스처 뷰 성분 및 깊이 뷰 성분을 포함할 수도 있다. 본 개시물에서, 텍스처 뷰 성분 및 깊이 뷰 성분이 동일한 뷰에 있고 동일한 액세스 유닛에 있다면, 텍스처 뷰 성분은 깊이 뷰 성분에 대응할 수도 있거나, 또는 반대로도 대응할 수도 있다. 서브-비트스트림 추출 프로세스 동안 텍스처 뷰 성분이 타겟 뷰에서의 화상을 디코딩하는데 필요하다고 중간 디바이스가 결정하면, 중간 디바이스는 텍스처 뷰 성분 또는 대응하는 깊이 뷰 성분에 연관된 NAL 유닛들을 제거하지 않는다. 마찬가지로, 서브-비트스트림 추출 프로세스 동안 깊이 뷰 성분이 타겟 뷰에서의 화상을 디코딩하는데 필요하다고 중간 디바이스가 결정하면, 중간 디바이스는 깊이 뷰 성분 또는 대응하는 텍스처 뷰 성분에 연관된 NAL 유닛들을 제거하지 않는다.

[0019] 그러나, 텍스처 뷰 성분이 타겟 뷰의 화상을 디코딩하는데 필요하지만 대응하는 깊이 뷰 성분이 임의의 타겟 뷰의 임의의 화상을 디코딩하는데 필요하지 않은 경우들이 있다. 마찬가지로, 깊이 뷰 성분이 타겟 뷰의 화상을 디코딩하는데 필요하지만 대응하는 텍스처 뷰 성분이 임의의 타겟 뷰의 임의의 화상을 디코딩하는데 필요하지 않은 경우들이 있다. 이런 경우들에서, 텍스처 뷰 성분 및 대응하는 깊이 뷰 성분 양쪽 모두를 포함시키는 것은 대역폭이 비효율적일 수도 있다. 따라서, 텍스처 뷰 성분에 연관되지만 타겟 뷰의 화상을 디코딩하는데 필요하지 않은 깊이 뷰 성분을 원래의 비트스트림으로부터 추가로 제거하는 것 또는, 대안으로, 깊이 뷰 성분에 연관되지만 타겟 뷰의 화상을 디코딩하는데 필요하지 않은 텍스처 뷰 성분을 원래의 비트스트림으로부터

추가로 제거하는 것이 바람직할 수도 있다.

[0020] 본 개시물의 기법들은, H.264/AVC의 3DV 확장본이라고 지칭되는 H.264/AVC의 MVC 확장본에 기초한 MVC 및 3D 비디오 코딩에 관련된다. 더 구체적으로는, 본 개시물의 기법들은 3DV 비트스트림으로부터의 서브-비트스트림들의 추출에 관련한다. 3DV 비트스트림은 코딩된 텍스처 뷰 성분들 및 코딩된 깊이 뷰 성분들을 포함할 수도 있다. 본 개시물의 기법들은 MVC 및 3DV에서 발생하는 문제들을 해결할 수도 있는데, MVC 및 3DV 양쪽 모두는, 주어진 뷰에 대한 텍스처 뷰 성분 및 깊이 뷰 성분 양쪽 모두가 타겟 뷰의 화상을 디코딩하는데 실제로 필요한 안 하든, 그 주어진 뷰에 대한 텍스처 뷰 성분 및 대응하는 깊이 뷰 성분 양쪽 모두가 전송되는 것을 필요로 할 수도 있다.

[0021] 본 개시물의 기법들에 따라, 디바이스는, 서브-비트스트림 추출 프로세스 동안, 텍스처 뷰 성분들 및 깊이 뷰 성분들에 대해 별개의 타겟 뷰 리스트들을 유지할 수도 있다. 텍스처 뷰 성분들에 대한 타겟 뷰 리스트는 타겟 뷰에서의 화상을 디코딩하기 위해 필요한 텍스처 뷰 성분들을 갖는 뷰들을 식별할 수도 있다. 깊이 뷰 성분들에 대한 타겟 뷰 리스트는 타겟 뷰에서의 화상을 디코딩하기 위해 필요한 깊이 뷰 성분들을 갖는 뷰들을 식별할 수도 있다. 디바이스는 텍스처 및 깊이 뷰 성분들에 대한 타겟 뷰 리스트들에 적어도 부분적으로 기초하여 서브-비트스트림을 결정할 수도 있다. 예를 들어, 디바이스는, 텍스처 뷰 성분들에 대한 타겟 뷰 리스트에서 열거되지 않은 뷰들에 속하는 텍스처 뷰 성분들의 코딩된 슬라이스들을 포함하는 비디오 코딩 계층(VCL) NAL 유닛들을 비트스트림으로부터 제거할 수도 있다. 비슷하게, 이 예에서, 디바이스는, 깊이 뷰 성분들에 대한 타겟 뷰 리스트에서 열거되지 않은 뷰들에 속하는 깊이 뷰 성분들의 코딩된 슬라이스들을 포함하는 VCL NAL 유닛들을 비트스트림으로부터 제거할 수도 있다.

[0022] 첨부된 도면들이 예들을 도시한다. 첨부된 도면들에서 참조 번호들에 의해 나타내어진 엘리먼트들은 다음의 설명에서 유사한 참조 번호들에 의해 나타내어진 엘리먼트들에 대응한다. 본 개시물에서, 엘리먼트들이 서수 단어들(예컨대, "제 1", "제 2", "제 3" 등)로 시작하는 이름들을 갖는 것은 반드시 그 엘리먼트들이 특정 순서를 가진다는 것을 의미하지는 않는다. 오히려, 이러한 서수 단어들은 단지 동일한 또는 유사한 유형의 상이한 엘리먼트들을 언급하기 위해 사용될 수도 있다.

[0023] 도 1은 본 개시물의 기법들을 활용할 수도 있는 일 예의 비디오 코딩 시스템 (10)을 도시하는 블록도이다. 본원에서 사용되는 바와 같이, 용어 "비디오 코더"는 비디오 인코더들 및 비디오 디코더들 양쪽 모두를 일반적으로 지칭한다. 본 개시물에서, 용어들 "비디오 코딩" 또는 "코딩"은 비디오 인코딩 또는 비디오 디코딩을 일반적으로 지칭할 수도 있다.

[0024] 도 1에 도시된 바와 같이, 비디오 코딩 시스템 (10)은 소스 디바이스 (12), 목적지 디바이스 (14), 및 콘텐츠 전달 네트워크(CDN; 16)를 포함한다. 소스 디바이스 (12)는 인코딩된 비디오 데이터를 포함하는 비트스트림을 생성한다. 따라서, 소스 디바이스 (12)는 비디오 인코딩 디바이스 또는 비디오 인코딩 장치라고 지칭될 수도 있다. CDN 디바이스 (16)는 소스 디바이스 (12)에 의해 생성된 비트스트림으로부터 서브-비트스트림을 추출할 수도 있고 그 서브-비트스트림을 목적지 디바이스 (14)로 송신할 수도 있다. 목적지 디바이스 (14)는 서브-비트스트림에서의 인코딩된 비디오 데이터를 디코딩할 수도 있다. 따라서, 목적지 디바이스 (14)는 비디오 디코딩 디바이스 또는 비디오 디코딩 장치라고 지칭될 수도 있다. 소스 디바이스 (12)와 목적지 디바이스 (14)는 비디오 코딩 디바이스들 또는 비디오 코딩 장치들의 예들일 수도 있다.

[0025] 소스 디바이스 (12)와 목적지 디바이스 (14)는 데스크톱 컴퓨터들, 모바일 컴퓨팅 디바이스들, 노트북(예컨대, 랩톱) 컴퓨터들, 태블릿 컴퓨터들, 셋톱 박스들, 이른바 "스마트" 폰들과 같은 전화기 핸드셋들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 콘솔들, 차량내 컴퓨터 등을 포함한 다양한 범위의 디바이스들을 포함할 수도 있다. CDN 디바이스 (16)는 다양한 유형들의 컴퓨팅 디바이스들, 이를테면 서버 디바이스들, 개인용 컴퓨팅 디바이스들, 중간 네트워크 디바이스들(예컨대, 지능형 라우터들, 스위치들 등)들을 포함할 수도 있다. CDN 디바이스 (16)는 비디오 콘텐츠를 비디오 디코딩 디바이스들, 이를테면 목적지 디바이스 (14)로 전달하는 콘텐츠 전달 네트워크의 부분일 수도 있다. 비록 본 개시물이 본 개시물의 기법들을 구현하는 것으로서 CDN 디바이스를 설명하지만, 본 개시물의 기법들은 CDN의 사용 없이 구현될 수도 있다. 더욱이, 본 개시물의 기법들은 다른 유형들의 디바이스들에서 구현될 수도 있다. 예를 들어, 본 개시물의 기법들은 소스 디바이스 (12), 목적지 디바이스 (14), 또는 다른 컴퓨팅 디바이스에서 구현될 수도 있다.

[0026] CDN 디바이스 (16)는 소스 디바이스 (12)로부터 채널 (17A)을 통해 비트스트림을 수신할 수도 있다. 목적지 디바이스 (14)는 CDN 디바이스 (16)로부터 채널 (17B)을 통해 서브-비트스트림을 수신할 수도 있다.

본 개시물은 채널들 (17A 및 17B) 을 총칭하여 "채널들 (17)"이라고 지칭할 수도 있다. 채널들 (17) 의 각각은 하나의 컴퓨팅 디바이스로부터 다른 컴퓨팅 디바이스로 인코딩된 비디오 데이터를 이동시킬 수 있는 하나 이상의 매체들 또는 디바이스들을 포함할 수도 있다. 하나의 예에서, 채널들 (17) 중 어느 하나 또는 양쪽 모두는 디바이스가 인코딩된 비디오 데이터를 다른 디바이스로 실시간으로 송신하는 것을 가능하게 하는 하나 이상의 통신 매체들을 포함할 수도 있다. 이 예에서, 디바이스는 인코딩된 비디오 데이터를 통신 표준, 이를테면 유선 또는 무선 통신 프로토콜에 따라 변조할 수도 있고, 변조된 비디오 데이터를 다른 디바이스로 송신할 수도 있다. 하나 이상의 통신 매체들은 무선 및/또는 유선 통신 매체들, 이를테면 무선 주파수 (RF) 스펙트럼 또는 하나 이상의 물리적 송신 라인들을 포함할 수도 있다. 하나 이상의 통신 매체들은 패킷 기반 네트워크, 이를테면 로컬 영역 네트워크, 광역 네트워크, 또는 글로벌 네트워크 (예컨대, 인터넷) 의 부분을 형성할 수도 있다. 하나 이상의 통신 매체들은 라우터들, 스위치들, 기지국들, 또는 통신을 용이하게 하는 다른 장비를 포함할 수도 있다.

[0027] 다른 예에서, 채널들 (17) 중 어느 하나 또는 양쪽 모두는 인코딩된 비디오 데이터를 저장하는 저장 매체를 포함할 수도 있다. 이 예에서, CDN 디바이스 (16) 및/또는 목적지 디바이스 (14) 는 디스크 액세스 또는 카드 액세스를 통해 저장 매체에 액세스할 수도 있다. 저장 매체는 블루 레이 디스크들, DVD들, CD-ROM들, 플래시 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 다른 적합한 디지털 저장 매체들과 같은 다양한 로컬로 액세스되는 데이터 저장 매체들을 포함할 수도 있다.

[0028] 추가의 예에서, 채널들 (17) 중 어느 하나 또는 양쪽 모두는 인코딩된 비디오 데이터를 저장하는 파일 서버 또는 다른 중간 저장 디바이스를 포함할 수도 있다. 이 예에서, CDN 디바이스 (16) 및/또는 목적지 디바이스 (14) 는 파일 서버 또는 다른 중간 저장 디바이스에 저장된 인코딩된 비디오 데이터를 스트리밍 또는 다운로드를 통해 액세스할 수도 있다. 파일 서버는 인코딩된 비디오 데이터를 저장하고 그 인코딩된 비디오 데이터를 목적지 디바이스 (14) 에 송신할 수 있는 유형의 서버일 수도 있다. 예의 파일 서버들은 웹 서버들 (예컨대, 웹사이트용), 파일 전송 프로토콜 (FTP) 서버들, 네트워크 부속 스토리지 (network attached storage; NAS) 디바이스들, 및 로컬 디스크 드라이브들을 포함한다.

[0029] CDN 디바이스 (16) 및/또는 목적지 디바이스 (14) 는 표준 데이터 접속, 이를테면 인터넷 접속을 통해, 인코딩된 비디오 데이터에 액세스할 수도 있다. 예의 유형들의 데이터 접속들은 파일 서버 상에 저장된 인코딩된 비디오 데이터에 액세스하기에 적합한 무선 채널들 (예컨대, Wi-Fi 접속들), 유선 접속들 (예컨대, DSL, 케이블 모뎀 등), 또는 양쪽 모두의 조합들을 포함할 수도 있다. 파일 서버로부터의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 양쪽 모두의 조합일 수도 있다.

[0030] 본 개시물의 기법들은 무선 애플리케이션들 또는 설정 (setting) 들로 제한되지 않는다. 그 기법들은, 다양한 멀티미디어 애플리케이션들, 이를테면 OTA (over-the-air) 텔레비전 브로드캐스트들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, 예컨대, 인터넷을 통한 스트리밍 비디오 송신들 중 임의의 것의 지원 하의 비디오 코딩, 데이터 저장 매체 상의 저장을 위한 비디오 데이터의 인코딩, 데이터 저장 매체 상에 저장된 비디오 데이터의 디코딩, 또는 다른 애플리케이션들에 적용될 수도 있다. 일부 예들에서, 비디오 코딩 시스템 (10) 은 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅, 및/또는 화상 통화와 같은 애플리케이션들을 지원하기 위해 단방향 또는 양방향 비디오 송신을 지원하도록 구성될 수도 있다.

[0031] 도 1의 예에서, 소스 디바이스 (12) 는 비디오 소스 (18), 비디오 인코더 (20), 및 출력 인터페이스 (22) 를 구비한다. 일부 예들에서, 출력 인터페이스 (22) 는 변조기/복조기 (모뎀) 및/또는 송신기를 구비할 수도 있다. 비디오 소스 (18) 는 비디오 캡처 디바이스, 예컨대, 비디오 카메라, 이전에 캡처된 비디오 데이터를 포함한 비디오 아카이브, 비디오 콘텐츠 제공자로부터 비디오 데이터를 수신하는 비디오 피드 인터페이스, 및/또는 비디오 데이터를 생성하는 컴퓨터 그래픽 시스템, 또는 비디오 데이터의 이러한 소스들의 조합, 또는 비디오 데이터의 다른 소스 또는 소스들을 포함할 수도 있다.

[0032] 비디오 인코더 (20) 는 비디오 소스 (18) 로부터의 비디오 데이터를 인코딩할 수도 있다. 일부 예들에서, 소스 디바이스 (12) 는 출력 인터페이스 (22) 를 통해 인코딩된 비디오 데이터를 직접 송신한다. 다른 예들에서, 인코딩된 비디오 데이터는 또한 디코딩 및/또는 플레이백을 위한 목적지 디바이스 (14) 에 의한 나중의 액세스를 위해 저장 매체 또는 파일 서버 상에 저장될 수도 있다.

[0033] 도 1의 예에서, 목적지 디바이스 (14) 는 입력 인터페이스 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (32) 를 구비한다. 일부 예들에서, 입력 인터페이스 (28) 는 수신기 및/또는 모뎀을 구비한다. 입력 인터페이스 (28) 는 채널 (17B) 을 통해 인코딩된 비디오 데이터를 수신할 수도 있다. 디스플레이 디바이스

(32)는 목적지 디바이스(14)와 통합되거나 또는 그것 외부에 있을 수도 있다. 대체로, 디스플레이 디바이스(32)는 디코딩된 비디오 데이터를 디스플레이한다. 디스플레이 디바이스(32)는 액정 디스플레이(LCD), 플라즈마 디스플레이, 유기 발광 다이오드(OLED) 디스플레이, 또는 다른 유형의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들을 포함할 수도 있다.

[0034] 일부 예들에서, 비디오 인코더(20) 및 비디오 디코더(30)는, ISO/IEC MPEG-4 비주얼 그리고 SVC(Scalable Video Coding) 및 MVC(Multiview Video Coding) 확장본들을 포함한 ITU-T H.264(또한 ISO/IEC MPEG-4 AVC로 알려짐)와 같은 비디오 압축 표준에 따라 동작한다. H.264/AVC의 MVC 확장본의 최근의 초안은, 2013년 2월 26일 현재, http://wftp3.itu.int/av-arch/jvt-site/2009_01_Geneva/JVT-AD007.zip에서 다운로드 가능하며, 그 전체 내용은 본원에 참조로 통합된다. MVC의 다른 공동 초안(joint draft)은 『"Advanced Video Coding for Generic Audiovisual Services", ITU-T Recommendation H.264, 2010년 3월』에 기재되어 있으며, 그 전체 내용은 참조로 본원에 통합된다. 덧붙여서, MVC 표준의 확장본으로서, 『Suzuki 외, "WD of MVC extension for inclusion of depth maps", ISO/IEC/JTC1/SC29/WG11/N12351, 2011년 12월』(이후로는, "MPEG 문서 w12351")에 기재된 비디오 코딩 표준, 즉 "MVC 기반 3DV"이 있으며, 그 전체 내용은 참조로 본원에 통합된다.

[0035] 다른 예들에서, 비디오 인코더(20)와 비디오 디코더(30)는 H.265/고 효율 비디오 코딩(HEVC) 표준을 포함하는 다른 비디오 압축 표준들에 따라 동작할 수도 있다. 게다가, "HEVC 규격 초안 9"이라고 지칭되는 H.265/HEVC 표준의 초안은, 『Bross 외, "High Efficiency Video Coding (HEVC) text specification draft 9", Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 11th Meeting: Shanghai, China, 2012년 10월』에 기재되어 있으며, 이는 2013년 2월 26일 현재, http://phenix.int-evry.fr/jct/doc_end_user/documents/11_Shanghai/wg11/JCTVC-K1003-v8.zip에서 다운로드 가능하고, 그 전체가 참조로 본원에 통합된다. 다른 예의 비디오 코딩 표준들은 ITU-T H.261, ISO/IEC MPEG-1 비주얼, ITU-T H.262 또는 ISO/IEC MPEG-2 비주얼, ITU-T H.263, 및 ISO/IEC MPEG-4 비주얼을 포함한다. 비록 본 개시물의 기법들이 H.264/AVC에 관해 설명되고 있고, 예를 목적으로, MVC 플러스 깊이 기법들을 사용하는 H.264/AVC의 3DV 확장본에 관해 설명되고 있지만, 이러한 기법들은 또한 MVC 플러스 깊이 기법들을 사용하는 다른 코딩 표준들에 적용될 수도 있고, 임의의 특정 코딩 표준 또는 기법으로 제한되지 않는다.

[0036] 도 1은 단지 일 예이고 본 개시물의 기법들은 인코딩 및 디코딩 디바이스들 사이에 임의의 데이터 통신을 반드시 포함하지는 않는 비디오 코딩 설정들(예컨대, 비디오 인코딩 또는 비디오 디코딩)에 적용될 수도 있다. 다른 예들에서, 데이터는 로컬 메모리로부터 추출되며, 네트워크를 통해 스트리밍되는 등등이 된다. 비디오 인코딩 디바이스가 데이터를 인코딩하고 메모리에 저장할 수도 있으며, 그리고/또는 비디오 디코딩 디바이스는 메모리로부터 데이터를 추출하고 디코딩할 수도 있다. 많은 예들에서, 인코딩 및 디코딩은, 서로 통신하지 않지만 단순히 데이터를 메모리에 인코딩하고 및/또는 메모리로부터 데이터를 추출하고 디코딩하는 디바이스들에 의해 수행된다.

[0037] 비디오 인코더(20), 비디오 디코더(30), 및 CDN 디바이스(16) 각각은 다양한 적합한 회로, 이를테면 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들(DSP들), 주문형 집적회로들(ASIC들), 필드 프로그램가능 게이트 어레이들(FPGA들), 개별 로직, 하드웨어, 또는 그것들의 임의의 조합들 중 임의의 것으로서 구현될 수도 있다. 그 기법들이 소프트웨어에서 부분적으로 구현되면, 디바이스는 적합한 비일시적 컴퓨터 판독가능 저장 매체 내에 소프트웨어에 대한 명령을 저장할 수도 있고 하나 이상의 프로세서들을 사용하여 하드웨어에서 그 명령들을 실행하여 본 개시물의 기법들을 수행할 수도 있다. 전술한 바의 임의의 것(하드웨어, 소프트웨어, 하드웨어 및 소프트웨어의 조합 등을 포함)은 하나 이상의 프로세서들이라고 간주될 수도 있다. 비디오 인코더(20) 및 비디오 디코더(30)의 각각은 하나 이상의 인코더들 또는 디코더들 내에 구비될 수도 있고, 그것들 중 어느 하나는 결합형 인코더/디코더(CODEC)의 부분으로서 개별 디바이스 내에 통합될 수도 있다.

[0038] 본 개시물은 다른 디바이스, 이를테면 비디오 디코더(30)에 특정한 정보를 "시그널링하는" 비디오 인코더(20)를 일반적으로 지칭할 수도 있다. 용어 "시그널링"은 일반적으로 압축된 비디오 데이터를 디코딩하는데 사용되는 선택스 엘리먼트들 및/또는 다른 데이터의 통신을 말할 수도 있다. 이러한 통신은 실시간 또는 거의 실시간으로 일어날 수도 있다. 대안으로, 이러한 통신은, 인코딩 시에 선택스 엘리먼트들을 인코딩된 비트스트림으로 컴퓨터 판독가능 저장 매체에 저장하고 그 선택스 엘리먼트들이 이 매체에 저장된 후의 임의의 시간에 디코딩 디바이스에 의해 추출될 수도 있는 경우에 일어날 바와 같이 어떤 기간(a span of time)에 걸

쳐 일어날 수도 있다. 따라서, 시그널링은 인코딩된 비트스트림의 프로세싱 및/또는 디코딩에서 사용하기 위한 정보를 인코딩된 비트스트림으로 제공하는 것을 일반적으로 지칭할 수도 있다.

[0039] 비디오 시퀀스는 통상 일련의 비디오 프레임들을 포함한다. 화상들의 그룹 (GOP) 은 일반적으로 일련의 하나 이상의 비디오 프레임들을 포함한다. GOP는 선택스 데이터를 GOP의 헤더, GOP의 하나 이상의 프레임들의 헤더, 또는 GOP에 포함된 프레임들의 수를 기술하는 다른 곳에 포함할 수도 있다. 각각의 프레임은 개별 프레임에 대한 인코딩 모드를 기술하는 프레임 선택스 데이터를 포함할 수도 있다. 비디오 인코더 (20) 는 비디오 데이터를 인코딩하기 위하여 개개의 비디오 프레임들 내의 비디오 유닛들에 대해 통상 동작한다. H.264/AVC에서, 비디오 유닛은 매크로블록 또는 매크로블록의 구획에 대응할 수도 있다. MB는 3 개의 샘플 어레이들을 갖는 화상의 루마 샘플들의 16×16 블록 및 크로마 샘플들의 2 개의 대응하는 블록들, 또는 모노크롬 화상 또는 3 개의 별개의 컬러 평면들을 사용하여 코딩된 화상의 샘플들의 16×16 블록이다. MB 구획은, 3 개의 샘플 어레이들을 갖는 화상에 대한 인터 예측을 위한 매크로블록의 구획화의 결과로 생긴 루마 샘플들의 블록 및 크로마 샘플들의 2 개의 대응하는 블록들, 또는 모노크롬 화상 또는 3 개의 별개의 컬러 평면들을 사용하여 코딩된 화상의 인터 예측을 위한 매크로블록의 구획화의 결과로 생긴 루마 샘플들의 블록이다. H.265/HEVC에서, 비디오 유닛은 예측 유닛 (prediction unit; PU) 에 대응할 수도 있다. PU는 화상의 루마 샘플들의 예측 블록, 크로마 샘플들의 2 개의 대응하는 예측 블록들, 및 예측 블록 샘플들을 예측하는데 사용된 선택스 구조들일 수도 있다. 예측 블록은 동일한 예측이 적용되는 샘플들의 직사각형 (예컨대, $M \times N$) 블록일 수도 있다. 비디오 유닛들의 블록들은 고정된 또는 가변하는 사이즈들을 가질 수도 있고, 특정된 코딩 표준에 따라 사이즈를 달리할 수도 있다.

[0040] 비디오 인코더 (20) 가 현재 비디오 유닛을 인코딩하는 경우, 비디오 인코더 (20) 는 현재 비디오 유닛에 대응하는 예측 블록들을 생성할 수도 있다. 비디오 인코더 (20) 는 인트라 예측 또는 인터 예측을 수행하여 예측 블록들을 생성할 수도 있다. 비디오 인코더 (20) 가 현재 비디오 유닛에 대한 인트라 예측을 수행하는 경우, 비디오 인코더 (20) 는, 현재 비디오 유닛과 동일한 화상 내의 샘플들에 기초하여, 현재 비디오 유닛에 대한 예측 블록들을 생성할 수도 있다. 비디오 인코더 (20) 가 현재 비디오 유닛에 대한 예측 블록들을 생성하기 위해 인터 예측을 수행하는 경우, 비디오 인코더 (20) 는 하나 이상의 참조 화상들 내의 샘플들에 기초하여 예측 블록들을 생성할 수도 있다. 참조 화상들은 현재 비디오 유닛을 포함하는 화상과는 다른 화상들일 수도 있다.

[0041] 비디오 인코더 (20) 가 현재 비디오 유닛에 대한 예측 블록들을 생성한 후, 비디오 인코더 (20) 는 현재 비디오 유닛에 대한 잔차 블록들을 생성할 수도 있다. 잔차 블록에서의 각각의 샘플은 현재 비디오 유닛의 루마 또는 크로마 블록 및 현재 비디오 유닛에 대한 예측 블록에서의 대응하는 샘플들 사이의 차이에 기초할 수도 있다. 비디오 인코더 (20) 는 잔차 블록의 샘플들에 대해 변환을 적용하여 변환 계수 블록을 생성할 수도 있다. 비디오 인코더 (20) 는 다양한 변환들을 잔차 블록에 적용할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 이산 코사인 변환 (DCT), 정수 변환, 웨이브릿 변환과 같은 변환, 또는 개념적으로 유사한 변환을 잔차 블록에 적용할 수도 있다. 비디오 인코더 (20) 는 변환 계수 블록들을 양자화하여 현재 비디오 유닛을 표현하는데 사용된 비트들의 수를 더 감소시킬 수도 있다. 변환 계수 블록을 양자화한 후, 비디오 인코더 (20) 는 변환 계수 블록에서의 변환 계수들을 나타내는 선택스 엘리먼트들 및 현재 비디오 유닛에 연관된 다른 선택스 엘리먼트들을 엔트로피 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 선택스 엘리먼트들에 대해 컨텍스트 적응 이진 산술 코딩 (CABAC), 컨텍스트 적응 가변 길이 코딩 (CAVLC), 지수-골롬 코딩 (exponential-Golomb coding) 또는 다른 유형의 엔트로피 인코딩을 수행할 수도 있다. 비디오 인코더 (20) 는 현재 비디오 유닛에 연관된 엔트로피 인코딩된 선택스 엘리먼트들을 포함하는 비트스트림을 출력할 수도 있다.

[0042] 그 비트스트림은 코딩된 화상들의 표현 및 연관된 데이터를 형성하는 비트들의 시퀀스를 포함할 수도 있다. 그 비트스트림은 네트워크 추상 계층 (NAL) 유닛들의 시퀀스를 포함할 수도 있다. NAL 유닛들의 각각은 NAL 유닛 헤더를 포함하고 원시 바이트 시퀀스 페이로드 (raw byte sequence payload; RBSP) 를 캡슐화한다. NAL 유닛 헤더는 NAL 유닛 유형 코드를 나타내는 선택스 엘리먼트를 포함할 수도 있다. NAL 유닛의 NAL 유닛 헤더에 의해 특정된 NAL 유닛 유형 코드는 NAL 유닛의 유형을 나타낸다. RBSP는 NAL 유닛 내에 캡슐화되는 정수 수의 바이트들을 포함하는 선택스 구조일 수도 있다. 어떤 경우들에서, RBSP는 영 비트들을 포함한다.

[0043] 상이한 유형들의 NAL 유닛들이 상이한 유형들의 RBSP들을 캡슐화할 수도 있다. 예를 들어, 제 1 유형의 NAL 유닛은 화상 파라미터 세트 (picture parameter set; PPS) 에 대한 RBSP를 캡슐화할 수도 있으며, 제 2 유형의

NAL 유닛은 코딩된 슬라이스에 대한 RBSP를 캡슐화할 수도 있으며, 제 3 유형의 NAL 유닛은 추가 향상 정보 (supplemental enhancement information; SEI) 에 대한 RBSP를 캡슐화할 수도 있다는 등등이다. 비디오 코딩 데이터에 대한 RBSP들 (파라미터 세트들 및 SEI 메시지들에 대한 RBSP과는 대조적임) 을 캡슐화하는 NAL 유닛들은, 비디오 코딩 계층 (VCL) NAL 유닛들이라고 지칭될 수도 있다. 코딩된 슬라이스를 캡슐화하는 NAL 유닛은 코딩된 슬라이스 NAL 유닛이라고 지칭될 수도 있다.

[0044] 비디오 디코더 (30) 는 비디오 데이터의 인코딩된 표현을 포함하는 비트스트림을 수신할 수도 있다. 비디오 디코더 (30) 는 그 비트스트림을 파싱하여 비트스트림으로부터 신택스 엘리먼트들을 추출할 수도 있다. 비트스트림으로부터 신택스 엘리먼트들을 추출하는 부분으로서, 비디오 디코더 (30) 는 비트스트림의 부분들을 엔트로피 디코딩할 수도 있다. 비디오 디코더 (30) 는, 현재 비디오 유닛 (예컨대, MB 또는 MB 구획) 에 연관된 신택스 엘리먼트들에 적어도 부분적으로 기초하여, 인터 또는 인트라 예측을 수행하여 그 현재 비디오 유닛에 대한 예측 블록들을 생성할 수도 있다. 덧붙여서, 비디오 디코더 (30) 는 현재 비디오 유닛에 연관된 계수 변환 계수 블록들의 변환 계수들을 역 양자화할 수도 있고, 하나 이상의 역 변환들을 변환 계수 블록들에 적용하여 그 현재 비디오 유닛에 대한 잔차 블록들을 생성할 수도 있다. 비디오 디코더 (30) 는 그 다음에 잔차 블록들 및 예측 블록들에 적어도 부분적으로 기초하여 현재 비디오 유닛의 루마 및 크로마 블록들을 복원할 수도 있다. 이런 식으로, 화상의 각각의 비디오 유닛의 루마 및 크로마 블록들을 복원함으로써, 비디오 디코더 (30) 는 그 화상을 복원할 수도 있다.

[0045] 위에서 언급했듯이, 비디오 인코더 (20) 는 인터 예측을 수행하여 특정 비디오 유닛에 대한 예측 블록들을 생성할 수도 있다. 더 구체적으로는, 비디오 인코더 (20) 는 단방향 인터 예측 또는 양방향 인터 예측을 수행하여 예측 블록들을 생성할 수도 있다.

[0046] 비디오 인코더 (20) 가 현재 비디오 유닛 (예컨대, MB 또는 MB 구획) 에 대해 단방향 인터 예측을 수행하는 경우, 비디오 인코더 (20) 는 단일 참조 화상 리스트에서의 참조 화상들 내에서 참조 블록을 검색할 수도 있다. 참조 블록은 현재 비디오 유닛의 루마 및 크로마 블록들과 유사한 루마 샘플들의 블록 및 크로마 샘플들의 대응하는 블록들일 수도 있다. 더욱이, 비디오 인코더 (20) 가 단방향 인터 예측을 수행하는 경우, 비디오 인코더 (20) 는 특정 비디오 유닛에 대한 모션 정보를 생성할 수도 있다. 특정 비디오 유닛에 대한 모션 정보는 모션 벡터 및 참조 인덱스를 포함할 수도 있다. 모션 벡터는 현재 비디오 유닛 내의 블록들의 현재 화상 내의 포지션 및 참조 블록의 참조 화상 내 포지션 사이의 공간적 변위를 나타낼 수도 있다. 참조 인덱스는 참조 블록을 포함하는 참조 화상의 참조 화상 리스트 내의 포지션을 나타낸다. 현재 비디오 유닛에 대한 예측 블록들에서의 샘플들은 참조 블록에서의 대응하는 샘플들과 동일할 수도 있다.

[0047] 비디오 인코더 (20) 가 현재 비디오 유닛 (예컨대, MB 또는 MB 구획) 에 대해 양방향 인터 예측을 수행하는 경우, 비디오 인코더 (20) 는 제 1 참조 화상 리스트 ("리스트 0") 에서의 참조 화상들 내에서 제 1 참조 블록을 검색할 수도 있고, 제 2 참조 화상 리스트 ("리스트 1") 에서의 참조 화상들 내에서 제 2 참조 블록을 검색할 수도 있다. 비디오 인코더 (20) 는, 제 1 및 제 2 참조 블록들에 적어도 부분적으로 기초하여, 현재 비디오 유닛에 대한 예측 블록들을 생성할 수도 있다. 덧붙여서, 비디오 인코더 (20) 는 현재 비디오 유닛의 블록들 및 제 1 참조 블록 사이의 공간적 변위를 나타내는 제 1 모션 벡터를 생성할 수도 있다. 비디오 인코더 (20) 는 제 1 참조 블록을 포함하는 참조 화상의 제 1 참조 화상 리스트 내의 로케이션을 식별하는 제 1 참조 인덱스를 또한 생성할 수도 있다. 더욱이, 비디오 인코더 (20) 는 현재 비디오 유닛의 블록들 및 제 2 참조 블록 사이의 공간적 변위를 나타내는 제 2 모션 벡터를 생성할 수도 있다. 비디오 인코더 (20) 는 제 2 참조 블록을 포함하는 참조 화상의 제 2 참조 화상 리스트 내의 로케이션을 식별하는 제 2 참조 인덱스를 또한 생성할 수도 있다.

[0048] 비디오 인코더 (20) 가 현재 비디오 유닛 (예컨대, MB 또는 MB 구획) 에 대해 단방향 인터 예측을 수행하는 경우, 비디오 디코더 (30) 는 현재 비디오 유닛의 모션 정보를 사용하여 현재 비디오 유닛의 참조 블록을 식별할 수도 있다. 비디오 디코더 (30) 는 그 다음에 현재 비디오 유닛의 참조 블록에 기초하여 현재 비디오 유닛에 대한 예측 블록들을 생성할 수도 있다. 비디오 인코더 (20) 가 현재 비디오 유닛에 대해 양방향 인터 예측을 수행하는 경우, 비디오 디코더 (30) 는 현재 비디오 유닛에 대한 모션 정보를 사용하여 현재 비디오 유닛의 2 개의 참조 블록들을 식별할 수도 있다. 비디오 디코더 (30) 는 현재 비디오 유닛의 2 개의 참조 블록들에 기초하여 현재 비디오 유닛의 예측 블록들을 생성할 수도 있다.

[0049] 멀티뷰 비디오 코딩 (MVC) 은 H.264/AVC 표준의 확장본이다. 본 개시물은, H.264/AVC의 3DV 확장본에서처럼 3차원 (3D) 비디오 데이터의 MVC 플러스 깊이 코딩을 사용하여, 3차원 비디오 (3DV) 에 대한 기법들을

설명한다. MVC의 간략한 논의가 이제 제시될 것이다. H.264/AVC에 대한 MVC 확장본에서, 상이한 관점들로부터의 동일한 장면의 다수의 뷰들이 있을 수도 있다. 용어 "액세스 유닛"은 동일한 시간 인스턴스에 대응하는 뷰 성분들의 세트를 지칭하는데 사용된다. "뷰 성분"은 단일 액세스 유닛에서 뷰의 코딩된 표현일 수도 있다. 따라서, 비디오 데이터는 시간이 지남에 따라 발생하는 일련의 액세스 유닛들로서 개념화될 수도 있다. 본 개시물에서, "뷰"는 동일한 뷰 식별자에 연관된 뷰 성분들의 시퀀스를 지칭할 수도 있다. 동일한 뷰에 속하는 뷰 성분들의 코딩된 슬라이스들을 포함하는 VCL NAL 유닛들은 동일한 뷰 식별자를 특정한다. "뷰 순서 인덱스"는 액세스 유닛에서 뷰 성분들의 디코딩 순서를 나타내는 인덱스이다.

[0050] MVC는 뷰간 예측을 지원한다. 뷰간 예측은 H.264/AVC에서 사용된 인터 예측과 유사하고, 동일한 신택스 엘리먼트들을 사용할 수도 있다. 그러나, 비디오 코더가 현재 비디오 유닛에 대해 뷰간 예측을 수행하는 경우, 비디오 인코더 (20) 는, 참조 화상으로서, 현재 비디오 유닛과 동일한 액세스 유닛에 있지만 상이한 뷰에 있는 화상을 사용할 수도 있다. 그 반면, 기존의 인터 예측은 상이한 액세스 유닛들에서의 화상들만을 참조 화상들로서 사용한다. MVC에서, 비디오 디코더 (예컨대, 비디오 디코더 (30)) 가 뷰에서의 각각의 화상을 임의의 다른 뷰에서의 화상들을 참조하지 않고 디코딩할 수 있다면, 그 뷰는 "기본 뷰 (base view)"라고 지칭된다. 비 기본 뷰들 (non-base views) 중 하나의 비 기본 뷰에서의 화상을 코딩하는 경우, 화상이 상이한 뷰에 있지만 비디오 코더가 현재 코딩하고 있는 화상과는 동일한 시간 인스턴스 (즉, 액세스 유닛) 내에 있다면, 비디오 코더는 그 화상을 참조 화상 리스트에 추가할 수도 있다. 다른 인터 예측 참조 화상들처럼, 비디오 코더는 뷰간 예측 참조 화상을 참조 화상 리스트의 임의의 포지션에 삽입할 수도 있다. MVC에서, 뷰간 예측은 디스패리티 (disparity) 모션 보상에 의해 지원될 수도 있다. 디스패리티 모션 보상은 H.264/AVC 모션 보상의 신택스를 사용하지만, 상이한 뷰에서의 화상이 참조 화상으로서 사용되는 것을 허용할 수도 있다. 둘 이상의 뷰들의 코딩은 MVC에 의해 지원될 수도 있다. MVC의 장점들 중 하나는, MVC 인코더가 2 개를 초과하는 뷰들을 3D 비디오 입력으로서 사용할 수도 있고 MVC 디코더는 이러한 멀티뷰 표현을 디코딩할 수도 있다는 것일 수도 있다. 그 결과, MVC를 지원하는 비디오 디코더들은 2 개를 초과하는 뷰들로 3D 비디오 콘텐츠를 프로세싱할 수도 있다.

[0051] 더욱이, 위에서 언급했듯이, H.264/AVC에 대한 신폭 MVC-호환 3DV 확장본이 있다. MVC-호환 3DV는 MVC 호환성을 유지하면서 3D 강화를 가능하게 하도록 설계된다. MVC-호환 3DV는 깊이 맵들을 제공한다. 따라서, MVC-호환 3DV는 "MVC 플러스 깊이", "MVC+D"로서, 또는 "깊이를 포함하는 MVC-호환 확장본"으로서 지칭될 수도 있다. 그 전체 내용이 참조로 본원에 통합되는 『Suzuki 외, "WD on MVC extensions for inclusion of depth maps", ISO/IEC/JTC1/SC29/WG11/N12351, 2011년 12월』에서 제공되는 MVC-호환 3DV의 근간의 초안이, MVC-호환 3DV의 초안이다. 『Suzuki 외, "WD on MVC extensions for inclusion of depth maps", ISO/IEC/JTC1/SC29/WG11/N12544, 2012년 12월』 (이후로는 "문서 N12544"라 지칭됨), 그 전체 내용은 참조로 본원에 통합된다.

[0052] 깊이 뷰 성분은 깊이 맵을 포함한다. 깊이 맵들은, 화소 값들이 대응하는 "텍스처" 화상들에 보여진 오브젝트들의 3차원 깊이들을 나타내는 화상들이다. 일부 예들에서, 깊이 맵에서의 더 밝은 화소 값들일수록 카메라에 더 가까이 있는 오브젝트들에 대응할 수도 있고, 깊이 맵에서의 더 어두운 화소 값들일수록 카메라로부터 더 멀리 있는 오브젝트들에 대응할 수도 있다. "텍스처" 성분 화상들은 일반 H.264/AVC 화상들일 수도 있다.

[0053] 본 개시물에서, MVC-호환 3DV의 기술용어가 적용될 수도 있다. 예를 들면, 뷰의 텍스처 부분은 "텍스처 뷰"라고 지칭될 수도 있고 뷰의 깊이 부분은 "깊이 뷰"라고 지칭될 수도 있다. 하나의 액세스 유닛에서의 뷰의 텍스처 부분, 즉, 액세스 유닛에서의 텍스처 뷰는, "텍스처 뷰 성분"이라 지칭될 수도 있다. 하나의 액세스 유닛에서의 뷰의 깊이 부분, 즉, 액세스 유닛에서의 깊이 뷰는, "깊이 뷰 성분"이라고 지칭될 수도 있다. 용어 "뷰 성분"은 하나의 액세스 유닛에서의 뷰를 지칭하는데 그리고 동일한 액세스 유닛에서의 텍스처 뷰 성분 및 깊이 뷰 성분 양쪽 모두를 총칭하여 지칭하는데 사용될 수도 있다.

[0054] 비디오 인코더 (20) 는 깊이 이미지 기반 렌더링 (Depth Image Based Rendering; DIBR) 을 사용하여, 이용가능한 텍스처 및 깊이 뷰 성분들에 기초하여, 합성 텍스처 뷰 성분을 생성할 수도 있다. 합성 텍스처 뷰 성분은, 깊이 맵 및 하나 이상의 텍스처 뷰 성분들에 기초하여 합성된 텍스처 뷰 성분일 수도 있다. 예를 들어, 특정 텍스처 뷰 성분은 왼쪽 눈 텍스처 뷰 성분일 수도 있고 비디오 인코더 (20) 는 DIBR를 사용하여 3차원 비디오 플레이백을 위한 오른쪽 눈 텍스처 뷰 성분을 생성할 수도 있다. 어떤 경우들에서는, 합성 텍스처 뷰 성분이 액세스 유닛간 예측 또는 뷰간 예측을 위한 참조 화상으로서 사용될 수도 있다. 참조 화상들로서 사용되는 합성 텍스처 뷰 성분들은, 뷰 합성 참조 화상 (view synthesis reference picture; VSRP) 들이라고 지

칭될 수도 있다. 비디오 코더들은 VSRP들을 참조 화상 리스트들에 포함시킬 수도 있다.

[0055] 위에서 논의된 바와 같이, 뷰 성분은 참조를 위해 다른 뷰들에서의 뷰 성분들을 사용할 수 있으며; 이는 뷰간 예측이라 불린다. MVC (및 MVC-호환 3DV) 에서, 뷰간 예측은, 다른 뷰에서의 뷰 성분이 인터 예측 참조 화상이었던 것처럼 구현될 수도 있다. 잠재적 뷰간 참조 화상들은, MVC를 위한 SPS 확장 (즉, SPS MVC 확장) 에서 시그널링될 수도 있다. 아래의 표 1은, H.264를 위한 SPS MVC 확장본에 대한 일 예의 선택스를 도시한다.

[0056] 표 1

1	seq_parameter_set_mvc_extension() {	기술자
2	num_views_minus1	ue(v)
3	for(i = 0; i <= num_views_minus1; i++)	
4	view_id[i]	ue(v)
5	for(i = 1; i <= num_views_minus1; i++) {	
6	num_anchor_refs_l0[i]	ue(v)
7	for(j = 0; j < num_anchor_refs_l0[i]; j++)	
8	anchor_ref_l0[i][j]	ue(v)
9	num_anchor_refs_l1[i]	ue(v)
10	for(j = 0; j < num_anchor_refs_l1[i]; j++)	
11	anchor_ref_l1[i][j]	ue(v)
12	}	
13	for(i = 1; i <= num_views_minus1; i++) {	
14	num_non_anchor_refs_l0[i]	ue(v)
15	for(j = 0; j < num_non_anchor_refs_l0[i]; j++)	
16	non_anchor_ref_l0[i][j]	ue(v)
17	num_non_anchor_refs_l1[i]	ue(v)
18	for(j = 0; j < num_non_anchor_refs_l1[i]; j++)	
19	non_anchor_ref_l1[i][j]	ue(v)
20	}	
21	num_level_values_signalled_minus1	ue(v)
22	for(i = 0; i <= num_level_values_signalled_minus1; i++) {	
23	level_id[i]	u(8)
24	num_applicable_ops_minus1[i]	ue(v)
25	for(j = 0; j <= num_applicable_ops_minus1[i]; j++) {	
26	applicable_op_temporal_id[i][j]	u(3)
27	applicable_op_num_target_views_minus1[i][j]	ue(v)
28	for(k = 0; k <= applicable_op_num_target_views_minus1[i][j]; k++)	
29	applicable_op_target_view_id[i][j][k]	ue(v)
30	applicable_op_num_views_minus1[i][j]	ue(v)
31	}	
32	}	
33	}	

[0057]

[0058] 위의 표 1 및 본 개시물의 다른 선택스 표들에서, 유형 기술자 ue(v) 를 갖는 선택스 엘리먼트들은, 좌측 비트를 첫째로 하는 0차 지수 곱셈 (Exp-Golomb) 코딩을 사용하여 인코딩된 가변 길이 부호없는 정수들일 수도 있다. 표 1 및 다음의 표들의 예에서, n이 음이 아닌 정수인 형태 u(n) 의 기술자를 갖는 선택스 엘리먼트는, 길이 n의 부호없는 값들이다. 유형 기술자들 u(3) 및 u(8) 을 갖는 선택스 엘리먼트들은 3 개 및 8 개 비트들을 각각 갖는 부호없는 정수들일 수도 있다.

[0059]

표 1의 라인 3 및 라인 4에서 도시된 바와 같이, SPS MVC 확장은 적용가능한 뷰들의 뷰 식별자들 (예컨대, view_id[i]) 을 특정할 수도 있다. 더욱이, 비디오 인코더 (20) 는, 표 1의 SPS MVC 확장에서, 각각의 뷰에 대해, 참조 화상 리스트 0 및 참조 화상 리스트 1을 형성하는데 사용될 수 있는 뷰들의 수를 시그널링할 수도 있다. 예를 들면, 표 1의 라인 6 및 라인 14에서의 num_anchor_refs_l0[i] 및 num_non_anchor_refs_l0[i] 선택스 엘리먼트들은, i와 동일한 뷰 순서 인덱스를 각각 갖는 앵커 및 비 앵커 뷰 성분들의 디코딩에서 초기 참조 화상 리스트 0에서의 뷰간 예측을 위한 뷰 성분들의 수를 특정할 수도 있다. 표 1의 라인 9 및 라인 17에서의 num_anchor_refs_l1[i] 및 num_non_anchor_refs_l1[i] 선택스 엘리먼트들은, i와 동일한 뷰 순서 인덱스를 갖는 앵커 및 비 앵커 뷰 성분들의 디코딩에서 초기 참조 화상 리스트 1에서의 뷰간 예측을 위한 뷰 성분들의 수를 특정할 수도 있다.

[0060] SPS MVC 확장은 SPS에 적용가능한 뷰들 및 뷰 성분들 사이의 잠재적 의존도들을 특정할 수도 있다. 예를 들어, 표 1의 예에서, SPS MVC 확장은 anchor_ref_l0[i][j], anchor_ref_l1[i][j], non_anchor_ref_l0[i][j], 및 non_anchor_ref_l1[i][j]로 표시된 선택스 엘리먼트들을 포함할 수도 있다. anchor_ref_l0[i][j]는 i와 동일한 뷰 순서 인덱스 (VOIdx) 를 갖는 앵커 뷰 성분들의 디코딩에서 초기 참조 화상 리스트 RefPicList0에서의 뷰간 예측을 위한 j-번째 뷰 성분의 뷰 식별자 (view_id) 를 특정한다. anchor_ref_l1[i][j]는 i와 동일한 뷰 순서 인덱스 (VOIdx) 를 갖는 앵커 뷰 성분들의 디코딩에서 초기 참조 화상 리스트 RefPicList1에서의 뷰간 예측을 위한 j-번째 뷰 성분의 뷰 식별자 (view_id) 를 특정한다. non_anchor_ref_l0[i][j]는 i와 동일한 뷰 순서 인덱스 (VOIdx) 를 갖는 비 앵커 뷰 성분들의 디코딩에서 초기 참조 화상 리스트 RefPicList0에서의 뷰간 예측을 위한 j-번째 뷰 성분의 뷰 식별자 (view_id) 를 특정한다. anchor_ref_l1[i][j]는 i와 동일한 뷰 순서 인덱스 (VOIdx) 를 갖는 비 앵커 뷰 성분들의 디코딩에서 초기 참조 화상 리스트 RefPicList1에서의 뷰간 예측을 위한 j-번째 뷰 성분의 뷰 식별자 (view_id) 를 특정한다. 이런 식으로, SPS MVC 확장은 앵커 뷰 성분들 및 잠재적 뷰간 참조 화상들 사이의 가능한 예측 관계들 (즉, 의존도들) 을 특정할 수도 있다. 더구나, 이런 식으로, SPS MVC 확장본은 비 앵커 뷰 성분들 및 잠재적 뷰간 참조 화상들 사이의 가능한 예측 관계들 (즉, 의존도들) 을 특정할 수도 있다.

[0061] 앵커 화상에 대한 예측 관계는, SPS MVC 확장으로 시그널링된 바와 같이, 동일한 뷰의 (SPS MVC 확장으로 시그널링된) 비 앵커 화상에 대한 예측 관계와 상이할 수도 있다. 참조 화상 리스트들은 참조 화상 리스트들에서의 인터 예측 또는 뷰간 예측 참조 화상들의 유연한 순서화를 가능하게 하기 위하여 참조 화상 리스트구성 프로세스 동안 수정될 수도 있다.

[0062] MVC (및 MVC-호환 3DV) 에서, NAL 유닛이 접두 NAL 유닛 또는 MVC VCL NAL 유닛이면, 그 NAL 유닛은 1-바이트 NAL 유닛 헤더 및 3-바이트 MVC NAL 유닛 헤더 확장을 포함할 수도 있다. 1-바이트 NAL 유닛 헤더는 NAL 유닛 유형 및 nal_ref_idc 선택스 엘리먼트를 포함할 수도 있다. nal_ref_idc 선택스 엘리먼트는 NAL 유닛이 SPS, SPS 확장, 서브세트 SPS, PPS, 참조 화상의 슬라이스, 참조 화상의 슬라이스 데이터 구획, 또는 참조 화상의 슬라이스에 선행하는 접두 NAL 유닛을 포함하는지 여부를 특정한다. MVC (및 MVC-호환 3DV) 에서의 접두 NAL 유닛은 오직 NAL 유닛 헤더 및 MVC NAL 유닛 헤더 확장만을 포함할 수도 있다. 아래의 표 2는, MVC NAL 유닛 헤더 확장 (nal_unit_header_mvc_extension) 을 나타낸다.

[0063] 표 2

<code>nal_unit_header_mvc_extension()</code> {	C	기술자
non_idr_flag	All	u(1)
priority_id	All	u(6)
view_id	All	u(10)
temporal_id	All	u(3)
anchor_pic_flag	All	u(1)
inter_view_flag	All	u(1)
reserved_one_bit	All	u(1)
}		

[0064]

[0065] 표 2의 예의 선택스 구조에서, non_idr_flag는, NAL 유닛이, 폐쇄형-GOP 랜덤 액세스 포인트로서 사용될 수 있는 순시 디코딩 리프레시 (instantaneous decoding refresh; IDR) NAL 유닛에 속하는지 여부를 나타낸다. 랜덤 액세스 포인트는 I 슬라이스들만을 포함하는 화상이다. priority_id 선택스 엘리먼트는 일-경로 적응을 위해 사용될 수도 있는데, 여기서 적응은 단순히 priority_id를 체크함으로써 행해질 수 있다. view_id 선택스 엘리먼트는 현재 뷰의 뷰 식별자를 나타낼 수도 있다. NAL 유닛은 현재 뷰의 뷰 성분의 슬라이스의 코딩된 표현을 캡슐화할 수도 있다. temporal_id 선택스 엘리먼트는 NAL 유닛의 시간적 레벨을 나타낼 수도 있다. 시간적 레벨은 NAL 유닛에 연관된 프레임 레이트를 나타낼 수도 있다. anchor_pic_flag 선택스 엘리먼트는, NAL 유닛이, 개방형 GOP 랜덤 액세스 포인트로서 사용될 수 있는 앵커 화상에 속하는지 여부를 나타낼 수도 있다. 앵커 화상은, 모든 슬라이스들이 동일한 액세스 유닛에서의 슬라이스들만을 참조할 수도 있는 코딩된 화상이다. 다시 말하면, 뷰간 예측은 앵커 화상을 인코딩하는데 사용될 수도 있지만, 인터 예측은 앵커 화상을 인코딩하는데 사용되지 않을 수도 있다. inter_view_flag 선택스 엘리먼트는 현재 뷰 성분이 다른 뷰들에서의 NAL 유닛들에 대한 뷰간 예측을 위해 사용되는지 여부를 나타낸다. NAL 유닛은 현재 뷰 성분의 슬라이스의 코딩된 표현을 캡슐화할 수도 있다.

- [0066] 2012년 3월 7일자로 출원된 미국 특허 출원 제13/414,515호는, 현재 뷰 성분이 텍스처 뷰 성분이면, 그 현재 뷰 성분이 임의의 깊이 뷰 성분을 예측하는데 사용되지 않는지 여부를 나타내기 위해 depth_to_view_flag 선택스 엘리먼트를 NAL 유닛 헤더에 도입한다. 다르게 말하면, NAL 유닛이 텍스처 뷰 성분의 코딩된 슬라이스를 캡슐화하면, NAL 유닛의 NAL 유닛 헤더에서의 depth_to_view_flag 선택스 엘리먼트는 텍스처 뷰 성분이 깊이 뷰 성분을 예측하는데 사용되는지 여부를 나타낸다.
- [0067] 인터넷 또는 다른 데이터 네트워크들을 통한 멀티-뷰 3D 콘텐츠의 전달을 위한 콘텐츠 전달 네트워크들(CDN들)의 출현으로, MVC-호환 3DV는 다양한 뷰들의 전부를 포함하는 비디오 데이터 비트스트림들의 CDN들로의 전달을 용이하게 하도록 적응되었다. 예를 들어, 중앙 비디오 라이브러리 서버(또는 다른 디바이스들)는 멀티-뷰 비디오 데이터를 단일 비트스트림으로서 인코딩할 수도 있고 이 단일 비트스트림을 다양한 클라이언트 디바이스들에 서비스하는 CDN으로 전달할 수도 있다. CDN에서의 디바이스들, 이를테면 CDN 디바이스(16)는, 클라이언트 디바이스들, 이를테면 목적지 디바이스(14)로의 전달을 위한 비트스트림을 로컬로 저장할 수도 있다.
- [0068] CDN 디바이스(16)는 비트스트림(즉, 원래의 비트스트림)으로부터 서브-비트스트림들을 추출하기 위하여 서브-비트스트림 추출이라고 지칭된 프로세스를 수행할 수도 있다. 원래의 비트스트림은 복수의 뷰들의 코딩된 표현들을 포함할 수도 있다. 서브-비트스트림들은 원래의 비트스트림의 뷰들의 서브세트의 코딩된 표현들을 포함할 수도 있다. 일부 예들에서, CDN 디바이스(16)는 원래의 비트스트림으로부터 특정 NAL 유닛들을 선택적으로 추출함으로써 원래의 비트스트림으로부터 서브-비트스트림을 추출할 수도 있다. 추출된 NAL 유닛들은 서브-비트스트림을 형성한다.
- [0069] CDN 디바이스(16)는 상이한 클라이언트 디바이스들의 능력들 및/또는 상이한 클라이언트 디바이스들에 연관된 송신 대역폭들에 기초하여 상이한 서브-비트스트림들을 추출할 수도 있다. 예를 들어, 원래의 비트스트림에서의 다수의 뷰들 중 하나 이상의 뷰들은 더 작은 디스플레이들을 갖는 클라이언트 디바이스들, 이를테면 모바일 전화기들에 지정될 수도 있다(여기서 이 뷰는 위에서 설명된 중앙 뷰일 수도 있는데, 그것은 보통 단일 관람자에 의해 일반적으로 보여지는 더 작은 디스플레이 상의 3D 콘텐츠를 보는데 필요한 뷰일 뿐이다). 원래의 비트스트림을 요청하는 클라이언트 디바이스가 모바일 전화기라는 결정에 응답하여, CDN 디바이스(16)는 원래의 비트스트림으로부터 특정 서브-비트스트림을 추출하고 그 특정 서브-비트스트림을 모바일 전화기로 전달할 수도 있다. 이 예에서, 특정 서브-비트스트림은 원래의 비트스트림에서의 뷰들 중 하나의 뷰만을 위한 코딩된 표현을 포함할 수도 있다.
- [0070] MVC-호환 3DV에 대해, 비트스트림 추출 프로세스는 위에서 참조된 문서 N12544에 관해 다음과 같이 설명될 수도 있다. 아래의 하위절들에 대한 언급은 문서 N12544의 하위절들 또는 문서 N12544에 의해 참조될 수도 있는 H.264/AVC의 3DV 확장본에 관련한 다른 관련된 문서들을 참조한다. 문서 N12544의 다수의 하위절들은 아래에서 재현되거나 또는 참조되고, 비트스트림 추출 프로세스를 약술한다. 서브-비트스트림 추출 프로세스는 다음과 같이 설명된다.
- [0071] 비트스트림 서브 세트들의 사양: H.264의 부록 H의 하위절 H.8.5가 적용된다.
- [0072] 다음은 H.264의 다음 부록 I에 적용한다.
- [0073] 요청된 앵커 뷰 성분들에 대한 I.8.5.1 도출 프로세스: 하위절 H.8.5.1의 사양은 "뷰 성분"을 "깊이 뷰 성분" 또는 "텍스처 뷰 성분" 중 어느 하나로 그리고 "뷰"를 "깊이 뷰" 또는 "텍스처 뷰" 중 어느 하나로 치환하는데 준용한다.
- [0074] 요청된 비 앵커 뷰 성분들에 대한 I.8.5.2 도출 프로세스: 하위절 H.8.5.2의 사양은 "뷰 성분"을 "깊이 뷰 성분" 또는 "텍스처 뷰 성분" 중 어느 하나로 그리고 "뷰"를 "깊이 뷰" 또는 "텍스처 뷰" 중 어느 하나로 치환하는데 준용한다.
- [0075] I.8.5.3 서브-비트스트림 추출 프로세스: 0 내지 63의 범위의 임의의 값과 동일한 pIdTarget, 0 내지 7의 범위의 임의의 값과 동일한 tIdTarget, 비트스트림에서 뷰들을 식별하는 viewIdTarget의 임의의 하나 이상의 값들로 구성된 viewIdTargetList로 이 하위절에서 특정된 프로세스의 출력인 임의의 서브-비트스트림이, 이 권고(국제 표준에 부합할 것이라는 것이 비트스트림 적합성의 요건이다. 주 1 - 부합하는 비트스트림은 0과 동일한 priority_id 및 0과 동일한 temporal_id를 갖는 하나 이상의 코딩된 슬라이스 NAL 유닛들을 포함한다. 주 2 - 서브-비트스트림 추출 프로세스의 결과로 생긴 서브-비트스트림들의 모든 동작 포인트들이 적용가능한 level_idc 또는 level_idc[i]를 가지지는 않는다는 것이 또한 가능할 수도 있다. 이 경우, 서브-비트스트림에서의 각각의 코딩된 비디오 시퀀스는 ITU-T H.264에서 부록 A, 부록 H 및 부록 I로 특정된 프로파일들 중 하

나 이상에 여전히 부합할 수도 있지만, 하위절들인 A.3, H.10.2 및 I.10.2에서 각각 특정된 레벨 제약조건들을 충족하지 못할 수도 있다.

- [0076] 이 서브-비트스트림 추출 프로세스에 대한 입력들은 다음을 포함한다: 1) 변수 depthPresentFlagTarget (존재하는 경우), 2) 변수 pIdTarget (존재하는 경우), 3) 변수 tIdTarget (존재하는 경우), 및 4) viewIdTarget (존재하는 경우) 의 하나 이상의 값들로 구성된 리스트 viewIdTargetList. 이 프로세스의 출력들은 서브-비트스트림과 VOIdx 값들의 리스트 VOIdxList이다. 덧붙여, depthPresentFlagTarget이 이 하위절에 대한 입력으로서 존재하지 않는 경우, depthPresentFlagTarget은 0과 동일하다고 유추된다. pIdTarget이 이 하위절에 대한 입력으로서 존재하지 않는 경우, pIdTarget은 63과 동일하다고 유추된다. tIdTarget이 이 하위절에 대한 입력으로서 존재하지 않는 경우, tIdTarget은 7과 동일하다고 유추된다. viewIdTargetList가 이 하위절에 대한 입력으로서 존재하지 않는 경우, viewIdTargetList에서 유추된 viewIdTarget의 하나의 값이 있을 것이고 viewIdTarget의 값은 기본 뷰의 view_id와 동일하다고 유추된다.
- [0077] 서브-비트스트림은 다음의 동작들을 순차적 순서로 적용함으로써 도출된다:
- [0078] 1. VOIdxList를 비우고 minVOIdx를 기본 뷰의 VOIdx 값이라고 둔다.
- [0079] 2. viewIdTargetList에 포함된 viewIdTarget의 각각의 값에 대해, viewIdTarget을 입력으로 하여 텍스처 뷰들에 대해 하위절 H.8.5.1에서 특정된 프로세스를 호출한다.
- [0080] 3. depthPresentFlagTarget이 1과 동일하면, viewIdTargetList에 포함된 viewIdTarget의 각각의 값에 대해, viewIdTarget을 입력으로 하여 깊이 뷰들에 대해 하위절 H.8.5.1에서 특정된 프로세스를 호출한다.
- [0081] 4. viewIdTargetList에 포함된 viewIdTarget의 각각의 값에 대해, viewIdTarget의 값을 입력으로 하여 텍스처 뷰들에 대해 하위절 H.8.5.2에서 특정된 프로세스를 호출한다.
- [0082] 5. depthPresentFlagTarget이 1과 동일하면, viewIdTargetList에 포함된 viewIdTarget의 각각의 값에 대해, viewIdTarget을 입력으로 하여 깊이 뷰들에 대해 하위절 H.8.5.2에서 특정된 프로세스를 호출한다.
- [0083] 6. 다음의 조건들 중 임의의 것들이 참인 모든 VCL NAL 유닛들 및 필러 데이터 NAL 유닛들을 "비트스트림으로부터 제거될 것"으로서 마킹한다:
- [0084] a) priority_id가 pIdTarget보다 크며,
- [0085] b) temporal_id가 tIdTarget보다 크며,
- [0086] c) anchor_pic_flag가 1과 동일하고 view_id는 "앵커에 필요함"으로 마킹되지 않으며,
- [0087] d) anchor_pic_flag가 0과 동일하고 view_id는 "비 앵커를 위해 요청됨"으로서 마킹되지 않으며,
- [0088] e) nal_ref_idc가 0과 동일하고 inter_view_flag는 0과 동일하고 view_id는 리스트 viewIdTargetList에서의 임의의 값과 동일하지 않으며,
- [0089] f) nal_unit_type이 21과 동일하고 depthPresentFlagTarget이 0과 동일하다.
- [0090] 7. 모든 VCL NAL 유닛들이 "비트스트림으로부터 제거될 것"으로서 마킹된 모든 액세스 유닛들을 제거한다.
- [0091] 8. "비트스트림으로부터 제거될 것"으로서 마킹된 모든 VCL NAL 유닛들 및 필러 데이터 NAL 유닛들을 제거한다.
- [0092] 9. VOIdxList가 minVOIdx와 동일한 VOIdx의 하나의 값만을 포함하는 경우, 다음의 NAL 유닛들을 제거한다:
- [0093] a) 14 또는 15와 동일한 nal_unit_type을 갖는 모든 NAL 유닛들,
- [0094] b) 제 1 SEI 메시지가 36 내지 44의 범위의 payloadType을 가지는 6과 동일한 nal_unit_type을 갖는 모든 NAL 유닛들.
- [0095] VOIdxList가 minVOIdx와 동일한 VOIdx의 하나의 값만을 포함하는 경우, 서브-비트스트림은 기본 뷰만을 또는 기본 뷰의 시간적 서브세트만을 포함한다.
- [0096] 10. depthPresentFlagTarget이 0과 동일한 경우, 제 1 SEI 메시지가 45 내지 47의 범위의 payloadType을 가지는 6과 동일한 nal_unit_type을 갖는 모든 NAL 유닛들을 제거한다.
- [0097] 11. maxTId를 모든 남아있는 VCL NAL 유닛들의 최대 temporal_id라고 둔다. 다음의 속성들 중 임의의 것을 갖는 MVC 스케일러블 네스팅 SEI 메시지의 부분인 SEI 메시지들만을 포함하는 6과 동일한 nal_unit_type을 갖는

모든 NAL 유닛들을 제거한다:

- [0098] a) operation_point_flag는 0과 동일하고 all_view_components_in_au_flag는 0과 동일하고 0 내지 num_view_components_minus1의 범위의 모든 i에 대해, sei_view_id[i] 중 어느 것도 VOIdxList에 포함된 VOIdx 값에 대응하지 않으며,
- [0099] b) operation_point_flag는 1과 동일하고, sei_op_temporal_id가 maxTid보다 크거나 또는 0 내지 num_view_components_op_minus1의 범위의 모든 i에 대한 sei_op_view_id[i]의 리스트가 viewIdTargetList의 서브셋이 아니다 (즉, 0 내지 num_view_components_op_minus1의 범위의 임의의 i에 대한 sei_op_view_id[i]가 viewIdTargetList에서의 값과 동일하다는 것이 참이 아니다).
- [0100] 12. 각각의 뷰 확장성 정보 SEI 메시지와 각각의 동작 포인트 비 존재 SEI 메시지를, 존재하는 경우, 제거한다.
- [0101] 13. VOIdxList가 minVOIdx와 동일한 VOIdx의 값을 포함하지 않는 경우, VOIdxList에 포함된 최소 VOIdx 값과 동일한 VOIdx를 갖는 뷰는 추출된 서브-비트스트림의 기본 뷰로 변환된다. 기본 뷰를 생성하는 키 프로세싱 단계들을 약술하는 유용한 정보를 주는 프로시저가 하위절 I.8.5.6에서 설명된다. VOIdxList가 minVOIdx와 동일한 VOIdx의 값을 포함하지 않는 경우, 위의 동작 단계 1 내지 단계 9에 따른 결과적인 서브-비트스트림은, 부록 A에 특정된 하나 이상의 프로파일들에 부합하는 기본 뷰를 포함하지 않는다. 이 경우, 이 동작 단계에 의해, 새로운 최소 VOIdx 값을 갖는 남아있는 뷰는 부록 A 및 부록 H에 특정된 하나 이상의 프로파일들에 부합하는 새로운 기본 뷰가 되도록 변환된다.
- [0102] 이 서브-비트스트림 추출 프로세스를 수행함에 있어서, CDN 디바이스 (16) 는 CDN으로부터 클라이언트 디바이스, 이를테면 목적지 디바이스 (14) 로의 대역폭 요건 다운스트림을 감소시킬 수도 있다. 이 대역폭 감소를 용이하게 하기 위해, MVC-호환 3DV는 서브-비트스트림 추출 프로세스를 용이하게 하는 다수의 비디오 코딩 기법들을 특징으로 한다. 예를 들어, MVC-호환 3DV는 뷰 내에서뿐만 아니라 뷰들에 걸쳐서도 인터 예측을 제공한다. 뷰간 예측은 동일한 액세스 유닛의 화상들 사이에서 일반적으로 허용되지만 상이한 액세스 유닛들에서의 화상들 사이에서 MVC-호환 3DV에 허용된다. 이런 점에서, 주어진 액세스 유닛에서의 특정 뷰 성분의 디코딩은, 주어진 액세스 유닛 또는 다른 액세스 유닛들에서의 하나 이상의 다른 뷰 성분들의 디코딩을 필요로 할 수도 있다. 다르게 말하면, 특정 뷰 성분은 주어진 액세스 유닛 또는 다른 액세스 유닛들에서의 하나 이상의 다른 뷰 성분들에 의존할 수도 있다. MVC-호환 3DV는, CDN 디바이스들, 이를테면 CDN 디바이스 (16) 가 임의의 뷰 성분들을 디코딩하지 않고 뷰 성분들 사이의 의존도들을 결정할 수 있도록 액세스 유닛들 및 파라미터 세트들 (예컨대, 시퀀스 파라미터 세트들, 화상 파라미터 세트들 등) 을 구조화한다. 오히려, CDN 디바이스들은 시퀀스 파라미터 세트들로 시그널링된 타겟 뷰 리스트들에 기초하여 뷰 성분들 사이의 의존도들을 결정할 수도 있다.
- [0103] 뷰 성분들 사이의 의존도들 (또한 "예측 관계들"이라고 지칭될 수도 있음) 은 앵커 및 비 앵커 화상들에 대해 상이할 수도 있다. 앵커 화상은, 코딩된 화상의 모든 슬라이스들이 동일한 액세스 유닛 내의 슬라이스들만을 참조하는 코딩된 화상이다. 결과적으로, 뷰간 예측은 앵커 화상에서 사용될 수도 있지만, 인터 예측 (즉, 인터 액세스 유닛 예측) 은 앵커 화상에서 사용되지 않는다. 출력 순서에서 앵커 화상을 뒤따르는 모든 코딩된 화상들이 디코딩 순서에서 코딩된 화상에 앞서는 임의의 화상으로부터의 인터 예측을 사용하는 것은 아니다. 비 앵커 화상은 앵커 화상 이외의 임의의 화상을 말한다.
- [0104] 보통, 앵커 화상들은 콘텐츠를 적시에 디코딩하는 것을 가능하게 하기 위해 비트스트림 내에서 주기적인 방식으로 활용된다. 예를 들어, 비디오 인코더 (20) 는 앵커 화상을 비트스트림에 주기적으로 삽입할 수도 있어서 비디오 디코더, 이를테면 비디오 디코더 (30) 는, 상당한 수의 부가적인 화상들을 버퍼링하지 않고 하나 이상의 뷰 성분들을 디코딩할 수도 있다. 이런 이유로, 앵커 화상들은 이러한 콘텐츠의 소비자에 의해 허용되는 알려진 최대 시간 한도로 디코딩 시간들을 감소시킴으로써 채널 변경들을 용이하게 할 수도 있다.
- [0105] 서브-비트스트림 추출 프로세스를 지원하기 위해, MVC-호환 3DV는 앵커 또는 비 앵커 화상들 중 어느 하나의 상이한 뷰 성분들 간을 구별하는 일 없이 앵커 및 비 앵커 화상들에 관해 서브-비트스트림들을 형성하는 상이한 규칙들을 제공한다. 다시 말하면, 코딩된 뷰는 3D 비디오가 실현될 수도 있도록 텍스처 뷰 성분뿐만 아니라 깊이 뷰 성분 또한 일반적으로 포함한다. 어떤 경우들에서, 깊이 뷰 성분은 텍스처 뷰 성분이 적절히 디코딩되는 것을 필요로 하는 반면, 다른 경우들에서, 깊이 뷰 성분은 텍스처 뷰 성분이 적절히 디코딩되는 것을 필요로 하지 않는다. 따라서, 현재 제안된 바대로의 MVC-호환 3DV는, 심지어 특정한 텍스처 뷰 성분들이 깊이 뷰 성분을 디코딩하는데 필요하지 않을 수도 있더라도, 서브-비트스트림을 형성하는 경우 비트스트림으로부터

이러한 텍스처 뷰 성분들을 추출할 수도 있다. 다르게 말하면, 출력을 목표로 한 view_id 값들의 각각의 세트에 대해, 일단 깊이 뷰 또는 텍스처 뷰가 뷰간 예측을 위해 앵커 화상 또는 비 앵커 화상에 의해 사용되는 것으로 식별되면, 그 뷰의 모든 VCL NAL 유닛들은, 그것들이 텍스처 뷰 성분 또는 깊이 뷰 성분에 속하는지 여부에 관계없이, 추출된 서브-비트스트림에 포함된다. 필요하지 않은 텍스처 뷰 성분들을 비트스트림으로부터 추출하지 않는 것은 (즉, 필요하지 않은 텍스처 뷰 성분들을 서브-비트스트림에 포함시키는 것은) 대역폭 비효율성들을 초래할 수도 있다.

[0106] 본 개시물의 기법들에 따라, CDN 디바이스 (16) 는 코딩된 화상에 대해 단일 리스트를 유지하기 보다는 그것의 깊이 뷰 성분 및 텍스처 뷰 성분들에 상관없이 텍스처 뷰 성분 및 깊이 뷰 성분에 대해 별개의 타겟 뷰 리스트들을 유지할 수도 있다. 이들 리스트들을 별개로 유지함으로써, CDN 디바이스 (16) 는 텍스처 뷰 성분 및 깊이 뷰 성분을 추출할 경우를 식별할 수도 있고, 단지 하나 또는 다른 하나가 요청되는 경우 항상 양쪽 모두를 전송하지 않는다. 타겟 뷰 리스트들이 유지되는 방도에서의 이 변화는 앵커 타겟 뷰 리스트들 및 비 앵커 타겟 뷰 리스트들 양쪽 모두에 적용될 수도 있다.

[0107] 3DV 비트스트림으로부터 서브-비트스트림을 추출하기 위해, CDN 디바이스 (16) 는 3DV 비트스트림에서의 뷰의 텍스처 뷰 성분에 대해 텍스처 타겟 뷰 리스트를 결정할 수도 있다. 텍스처 타겟 뷰 리스트는 텍스처 뷰 성분을 예측하는데 사용된 3DV 비트스트림의 하나 이상의 부분들을 나타낼 수도 있다. 덧붙여서, CDN 디바이스 (16) 는 3DV 비트스트림에서의 뷰의 NAL 유닛의 깊이 뷰 성분에 대해 깊이 타겟 뷰 리스트를 결정할 수도 있다. 이 깊이 타겟 뷰 리스트는 깊이 뷰 성분을 예측하는데 사용된 3DV 비트스트림의 하나 이상의 부분들을 나타낼 수도 있다. CDN 디바이스 (16) 는 텍스처 타겟 뷰 리스트 및 깊이 타겟 뷰 리스트에 적어도 부분적으로 기초하여 서브-비트스트림을 결정할 수도 있다. 텍스처 타겟 뷰 리스트가 깊이 타겟 뷰 리스트와는 상이한 뷰 식별자들을 포함할 수도 있기 때문에, CDN 디바이스 (16) 는 텍스처 뷰 성분을 연관된 깊이 뷰 성분을 추출하는 일 없이 추출할 수도 있다. 비슷하게, CDN 디바이스 (16) 는 깊이 뷰 성분을 연관된 텍스처 뷰 성분을 추출하는 일 없이 추출할 수도 있다. 텍스처 뷰 성분과 깊이 뷰 성분은 그것들이 동일한 카메라 로케이션 (즉, 동일한 뷰) 에 대응하면 연관된 것으로 간주될 수도 있다.

[0108] 위에서 나타낸 바와 같이, SPS는 SPS에 적용가능한 뷰들 및 뷰 성분들 사이의 잠재적 의존도들을 특정할 수도 있다. CDN 디바이스 (16) 는, 복수의 타겟 뷰들과 SPS에 의해 명시된 잠재적 의존도들에 적어도 부분적으로 기초하여, 텍스처 타겟 뷰 리스트를 결정할 수도 있다. 비슷하게, CDN 디바이스 (16) 는, 복수의 타겟 뷰들과 SPS에 의해 명시된 잠재적 의존도들에 적어도 부분적으로 기초하여, 깊이 타겟 뷰 리스트를 결정할 수도 있다.

[0109] 어떤 경우들에서, 비트스트림에서의 NAL 유닛들의 헤더들은, 위에서 설명된 바와 같이, depth_to_view 신택스 엘리먼트들을 포함할 수도 있다. 본 개시물의 기법들에 따라, depth_to_view_flag 신택스 엘리먼트의 의미 체계는 깊이 뷰 성분이 연관된 텍스처 뷰 성분을 디코딩하는 일 없이 성공적으로 디코딩될 수 있는지 여부를 나타내도록 확장될 수도 있다. 다시 말하면, NAL 유닛의 depth_to_view_flag 신택스 엘리먼트는 NAL 유닛에 의해 캡슐화된 깊이 뷰 성분의 코딩된 슬라이스가 텍스처 뷰 성분을 참조하는지 여부를 나타낼 수도 있다. 이런 식으로, 심지어 텍스처 뷰 성분이 텍스처에 대해 추출될 뷰들의 타겟 리스트에 존재하지 않는 경우에도, NAL 유닛의 depth_to_view_flag 신택스 엘리먼트가 1로 설정되고 대응하는 깊이 뷰 성분이 깊이를 위해 추출될 뷰들의 타겟 리스트에 있으면, 텍스처 뷰 성분의 VCL NAL 유닛들은 추출될 수도 있다.

[0110] 대안으로, 사용 텍스처 플래그 (use_texture_flag) 신택스 엘리먼트가 NAL 유닛 헤더들 속으로 도입될 수도 있다. 사용 텍스처 플래그 신택스 엘리먼트는 깊이 뷰 성분이 연관된 텍스처 뷰 성분을 디코딩하는 일 없이 성공적으로 디코딩될 수 있는지 여부를 나타낼 수도 있다. 다르게 말하면, NAL 유닛의 사용 텍스처 플래그 신택스 엘리먼트는 NAL 유닛에 의해 캡슐화된 깊이 뷰 성분이 깊이 뷰 성분에 대응하는 텍스처 뷰 성분을 디코딩하는 일 없이 디코딩가능한지 여부를 나타낼 수도 있다. 사용 텍스처 플래그 신택스 엘리먼트는, 깊이 뷰 성분이 올바른 디코딩을 위해 대응하는 텍스처 뷰 성분을 필요로 하는 경우, 텍스처 뷰 성분들에 대해 0으로 설정될 수도 있고 깊이 뷰 성분들에 대해 1로 설정될 수도 있다. 사용 텍스처 플래그 신택스 엘리먼트는, 깊이 뷰 성분이 올바른 디코딩을 위해 대응하는 텍스처 뷰 성분에 필요하지 않은 경우, 깊이 뷰 성분에 대해 0으로 설정될 수도 있다.

[0111] 이들 플래그들을 사용하는 것은, CDN 디바이스 (16) 에서, 텍스처 타겟 뷰 리스트 및 깊이 타겟 뷰 리스트들 양쪽 모두가 활용되고 있는 경우, 서브-비트스트림 추출 프로세스의 더 효율적인 수행을 용이하게 할 수도 있다.

[0112] 도 2는 본 개시물에서 설명되는 기법들을 구현할 수도 있는 일 예의 비디오 인코더 (20) 를 도시하는 블록도이

다. 도 2는 설명의 목적으로 제공되고 본 개시물에서 폭넓게 예시되고 설명된 바와 같은 기법들의 제한으로서 고려되지 않아야 한다. 설명의 목적으로, 본 개시물은 H.264/AVC 코딩의 맥락에서 비디오 인코더 (20)를 설명한다. 그러나, 본 개시물의 기법들은 다른 코딩 표준들 또는 방법들에 적용가능할 수도 있다.

[0113] 도 2의 예에서, 비디오 인코더 (20)는 예측 프로세싱 유닛 (100), 잔차 생성 유닛 (102), 변환 프로세싱 유닛 (104), 양자화 유닛 (106), 역 양자화 유닛 (108), 역 변환 프로세싱 유닛 (110), 복원 유닛 (112), 필터 유닛 (113), 디코딩된 화상 버퍼 (114), 및 엔트로피 인코딩 유닛 (116)을 포함한다. 예측 프로세싱 유닛 (100)은 인터 예측 프로세싱 유닛 (121)과 인트라 예측 프로세싱 유닛 (126)을 구비한다. 인터 예측 프로세싱 유닛 (121)은 모션 추정 유닛 (122)과 모션 보상 유닛 (124)을 구비한다. 다른 예들에서, 비디오 인코더 (20)는 더 많거나, 더 적거나, 또는 상이한 기능성 컴포넌트들을 포함할 수도 있다.

[0114] 비디오 인코더 (20)는 비디오 데이터를 수신한다. 비디오 데이터를 인코딩하기 위해, 비디오 인코더 (20)는 비디오 데이터의 각각의 화상의 각각의 슬라이스를 인코딩할 수도 있다. 슬라이스를 인코딩하는 부분으로서, 비디오 인코더 (20)는 그 슬라이스의 비디오 유닛들을 인코딩할 수도 있다.

[0115] 인터 예측 프로세싱 유닛 (121)은 인터 예측을 수행함으로써 현재 비디오 유닛에 대한 예측 데이터를 생성할 수도 있다. 현재 비디오 유닛에 대한 예측 데이터는 현재 비디오 유닛에 대한 예측 블록들 및 모션 정보를 포함할 수도 있다.

[0116] 슬라이스들은 I 슬라이스들, P 슬라이스들, SP 슬라이스들, 또는 B 슬라이스들일 수도 있다. 모션 추정 유닛 (122)과 모션 보상 유닛 (124)은 비디오 유닛이 I 슬라이스에 있는지, P 슬라이스에 있는지, SP 슬라이스에 있는지, 또는 B 슬라이스에 있는지에 의존하여 비디오 유닛에 대해 상이한 동작들을 수행할 수도 있다. I 슬라이스에서, 모든 비디오 유닛들이 인트라 예측된다. 그런고로, 비디오 유닛이 I 슬라이스에 있으면, 모션 추정 유닛 (122)과 모션 보상 유닛 (124)은 비디오 유닛에 대해 인터 예측을 수행하지 않는다. SP 슬라이스는, 각각의 블록의 샘플 값들을 예측하기 위해 기껏해야 하나의 모션 벡터 및 참조 인덱스를 사용하여 예측 샘플들의 양자화와 함께 인트라 예측 또는 인터 예측을 사용하여 코딩될 수도 있는 슬라이스이다. SP 슬라이스는 그것의 디코딩된 샘플들이 다른 SP 슬라이스 또는 SI 슬라이스와 동일하게 구성될 수 있도록 코딩될 수 있다. SI 슬라이스는 인트라 예측만을 사용하여 그리고 예측 샘플들의 양자화를 사용하여 코딩된 슬라이스이다. SI 슬라이스는 그것의 디코딩된 샘플들이 SP 슬라이스와 동일하게 구성될 수 있도록 코딩될 수 있다.

[0117] 인터 예측 프로세싱 유닛 (121)은 각각의 P, SP, 또는 B 슬라이스를 코딩하는 초반에 참조 화상 리스트 구성 프로세스를 수행할 수도 있다. 인터 예측 프로세싱 유닛 (121)이 P 또는 SP 슬라이스를 코딩하고 있으면, 인터 예측 프로세싱 유닛 (121)은 제 1 참조 화상 리스트 (예컨대, 리스트 0)를 생성할 수도 있다. 인터 예측 프로세싱 유닛 (121)이 B 슬라이스를 코딩하고 있으면, 인터 예측 프로세싱 유닛 (121)은 제 1 참조 화상 리스트 (예컨대, 리스트 0)를 생성하고 또한 제 2 참조 화상 리스트 (예컨대, 리스트 1)를 생성할 수도 있다.

[0118] 위에서 언급했듯이, MVC NAL 유닛 헤더 확장은 non_idr_flag 선택스 엘리먼트를 포함한다. NAL 유닛이 IDR 화상의 코딩된 슬라이스를 캡슐화한다는 것을 non_idr_flag 선택스 엘리먼트가 나타내면 (예컨대, non_idr_flag 선택스 엘리먼트가 0과 동일하면), 인터 예측 프로세싱 유닛 (121)은 모든 엔트리들이 참조 화상을 나타내지 않도록 리스트 0을 (그리고 코딩된 슬라이스가 코딩된 B 슬라이스이면 리스트 1)을 생성할 수도 있다. 그렇지 않고, NAL 유닛이 비-IDR 화상의 코딩된 슬라이스를 캡슐화한다는 것을 non_idr_flag 선택스 엘리먼트가 나타내면, 인터 예측 프로세싱 유닛 (121)은 리스트 0의 초기 버전을 생성하고, B 슬라이스에 대해, H.264/AVC 표준의 절 8.2.4.1에서 설명된 바와 같은 리스트 1의 초기 버전을 생성할 수도 있다.

[0119] 더욱이, 리스트 0의 초기 버전을 그리고, B 슬라이스에 대해, 리스트 1의 초기 버전을 생성한 후, 인터 예측 프로세싱 유닛 (121)은 뷰간 참조 성분들 또는 오직 뷰간의 (inter-view only) 참조 성분들을 리스트 0의 초기 버전에 그리고, B 슬라이스에 대해, 리스트 1의 초기 버전에 첨부할 수도 있다. 뷰간 참조 성분들 또는 오직 뷰간의 참조 성분들을 첨부한 후, 인터 예측 프로세싱 유닛 (121)은 참조 화상 리스트 수정 프로세스를 수행하여 리스트 0의 최종 버전과, B 슬라이스에 대해, 리스트 1의 최종 버전을 생성할 수도 있다.

[0120] 뷰간 참조 성분들을 리스트 0 및 리스트 1에 첨부하기 위해, 인터 예측 프로세싱 유닛 (121)은 NAL 유닛에 의해 캡슐화된 코딩된 슬라이스 (즉, 현재 슬라이스)가 앵커 화상의 부분인지 여부를 먼저 결정할 수도 있다. 위에서 설명된 바와 같이, MVC NAL 유닛 헤더 확장은 anchor_pic_flag 선택스 엘리먼트를 포함할 수도 있다.

NAL 유닛의 anchor_pic_flag 선택스 엘리먼트는 현재 슬라이스가 앵커 화상의 부분인지 여부를 나타낼 수도 있다. 현재 슬라이스가 앵커 화상의 부분이면, 인터 예측 프로세싱 유닛 (121) 은, 리스트 X (여기서 X는 0 또는 1) 에, 현재 슬라이스와 동일한 액세스 유닛에 속하고 적용가능한 SPS에서의 앵커 화상들에 특정된 각각의 뷰간 참조 화상을 첨부할 수도 있다. 따라서, 리스트 X에서의 뷰간 참조 화상들만이 현재 슬라이스와 동일한 액세스 유닛에 속한다.

[0121] 예를 들어, 위에서 나타낸 바와 같이, MVC SPS 확장은 num_anchor_refs_l0[i] 선택스 엘리먼트들, num_anchor_refs_l1 선택스 엘리먼트들, anchor_ref_l0[i][j] 선택스 엘리먼트들, 및 anchor_ref_l1[i][j] 선택스 엘리먼트들을 포함할 수도 있다. 이 예에서, 현재 슬라이스가 1과 동일한 anchor_pic_flag를 가지면, 인터 예측 프로세싱 유닛 (121) 은, 0 내지 num_anchor_refs_l0[i] - 1의 참조 뷰 인덱스 j의 각각의 값에 대해, j의 오름 차순으로, 리스트 0에, 현재 슬라이스와 동일한 액세스 유닛로부터 anchor_ref_l0[i][j]와 동일한 view_id를 갖는 뷰간 예측 참조를 첨부할 수도 있다. 마찬가지로, 현재 슬라이스가 1과 동일한 anchor_pic_flag를 가지면, 인터 예측 프로세싱 유닛 (121) 은, 0 내지 num_anchor_refs_l1[i] - 1의 참조 뷰 인덱스 j의 각각의 값에 대해, j의 오름 차순으로, 리스트 1에, 현재 슬라이스와 동일한 액세스 유닛로부터 anchor_ref_l1[i][j]와 동일한 view_id를 갖는 뷰간 예측 참조를 첨부할 수도 있다.

[0122] 그렇지 않고, 현재 슬라이스가 앵커 화상의 부분이 아니면, 인터 예측 프로세싱 유닛 (121) 은, 리스트 X (여기서 X는 0 또는 1) 에, 현재 슬라이스와 동일한 액세스 유닛에 속하고 적용가능한 SPS에서의 비 앵커 화상들에 특정된 각각의 뷰간 참조 화상을 첨부할 수도 있다. 따라서, 리스트 X에서의 뷰간 참조 화상들만이 현재 슬라이스와 동일한 액세스 유닛에 속한다.

[0123] 예를 들어, 위에서 나타낸 바와 같이, MVC SPS 확장은 num_non_anchor_refs_l0[i] 선택스 엘리먼트들, num_non_anchor_refs_l1 선택스 엘리먼트들, non_anchor_ref_l0[i][j] 선택스 엘리먼트들, 및 non_anchor_ref_l1[i][j] 선택스 엘리먼트들을 포함할 수도 있다. 이 예에서, 현재 슬라이스가 0과 동일한 anchor_pic_flag를 가지면, 인터 예측 프로세싱 유닛 (121) 은, 0 내지 num_non_anchor_refs_l0[i] - 1의 참조 뷰 인덱스 j의 각각의 값에 대해, j의 오름 차순으로, 리스트 0에, 현재 슬라이스와 동일한 액세스 유닛로부터 non_anchor_ref_l0[i][j]와 동일한 view_id를 갖는 뷰간 예측 참조를 첨부할 수도 있다. 마찬가지로, 현재 슬라이스가 0과 동일한 anchor_pic_flag를 가지면, 인터 예측 프로세싱 유닛 (121) 은, 0 내지 num_non_anchor_refs_l1[i] - 1의 참조 뷰 인덱스 j의 각각의 값에 대해, j의 오름 차순으로, 리스트 1에, 현재 슬라이스와 동일한 액세스 유닛로부터 non_anchor_ref_l1[i][j]와 동일한 view_id를 갖는 뷰간 예측 참조를 첨부할 수도 있다.

[0124] 현재 비디오 유닛이 P 슬라이스에 있으면, 모션 추정 유닛 (122) 은 현재 비디오 유닛에 대한 참조 블록을 참조 화상 리스트 (예컨대, 리스트 0) 에서의 참조 화상들에서 검색할 수도 있다. 비디오 유닛의 참조 블록은 현재 비디오 유닛의 루마 및 크로마 블록들에 가장 밀접하게 대응하는 루마 블록 및 대응하는 크로마 블록들을 포함할 수도 있다. 모션 추정 유닛 (122) 은 참조 화상에서의 참조 블록들이 현재 비디오 유닛의 루마 및 크로마 블록들에 얼마나 밀접하게 대응하는지를 결정하기 위해 다양한 메트릭들을 사용할 수도 있다. 예를 들어, 모션 추정 유닛 (122) 은 절대 차이 합 (sum of absolute differences; SAD), 차이 절대값 합 (sum of square differences; SSD), 또는 다른 차이 메트릭들에 의해 참조 화상에서의 참조 블록이 현재 비디오 유닛의 루마 및 크로마 블록들에 얼마나 밀접하게 대응하는지를 결정할 수도 있다.

[0125] 모션 추정 유닛 (122) 은 P 슬라이스에서의 현재 비디오 유닛 (예컨대, MB 또는 MB 구획) 의 참조 블록을 포함하는 리스트 0에서의 참조 화상을 나타내는 참조 인덱스와 현재 비디오 유닛의 블록들 및 참조 블록 사이의 공간적 변위를 나타내는 모션 벡터를 생성할 수도 있다. 모션 추정 유닛 (122) 은 참조 인덱스 및 모션 벡터를 비디오 유닛의 모션 정보로서 출력할 수도 있다. 모션 보상 유닛 (124) 은 현재 비디오 유닛의 모션 정보에 의해 나타내어진 참조 블록의 루마 및 크로마 블록들에 기초하여 현재 비디오 유닛에 대한 예측 루마 및 크로마 블록들을 생성할 수도 있다.

[0126] 현재 비디오 유닛이 B 슬라이스에 있으면, 모션 추정 유닛 (122) 은 현재 비디오 유닛에 대해 단방향 인터 예측 또는 양방향 인터 예측을 수행할 수도 있다. 현재 비디오 유닛에 대해 단방향 인터 예측을 수행하기 위해, 모션 추정 유닛 (122) 은 그 비디오 유닛에 대한 참조 블록을 리스트 0 또는 제 2 참조 화상 리스트 (예컨대, 리스트 1) 의 참조 화상들에서 검색할 수도 있다. 비디오 인코더 (20) 가 MVC 또는 3DV를 사용하는 예들에서, 리스트 0 및/또는 리스트 1은 뷰간 참조 화상들을 포함할 수도 있다. 모션 추정 유닛 (122) 은 참조 블록을 포함하는 참조 화상의 리스트 0 및 리스트 1에서의 포지션을 나타내는 참조 인덱스를 생성할 수도 있다.

더욱이, 모션 추정 유닛 (122) 은 현재 비디오 유닛의 블록들 및 참조 블록 사이의 공간적 변위를 나타내는 모션 벡터를 결정할 수도 있다. 모션 추정 유닛 (122) 은 참조 화상이 리스트 0에 있는지 또는 리스트 1에 있는지를 나타내는 예측 방향 표시자를 또한 생성할 수도 있다.

[0127] 현재 비디오 유닛 (예컨대, MB 또는 MB 구획) 에 대한 양방향 인터 예측을 수행하기 위해, 모션 추정 유닛 (122) 은 참조 블록을 리스트 0에서의 참조 화상들에서 검색할 수도 있고 또한 다른 참조 블록을 리스트 1에서의 참조 화상들에서 검색할 수도 있다. 모션 추정 유닛 (122) 은 참조 블록들을 포함하는 참조 화상들의 리스트 0 및 리스트 1에서의 포지션들을 나타내는 참조 인덱스들을 생성할 수도 있다. 덧붙여서, 모션 추정 유닛 (122) 은 참조 블록들 및 현재 비디오 유닛의 블록들 사이의 공간적 변위들을 나타내는 모션 벡터들을 생성할 수도 있다. 현재 비디오 유닛에 대한 모션 정보는 참조 인덱스들 및 현재 비디오 유닛의 모션 벡터들을 포함할 수도 있다. 모션 보상 유닛 (124) 은 현재 비디오 유닛의 모션 정보에 의해 나타내어진 참조 블록들에 기초하여 현재 비디오 유닛에 대한 예측 루마 및 크로마 블록들을 생성할 수도 있다.

[0128] 인트라 예측 프로세싱 유닛 (126) 은 인트라 예측을 수행함으로써 현재 비디오 유닛에 대한 예측 데이터를 생성할 수도 있다. 현재 비디오 유닛에 대한 예측 데이터는 현재 비디오 유닛에 대한 예측 블록들과 다양한 신택스 엘리먼트들을 포함할 수도 있다. 인트라 예측 프로세싱 유닛 (126) 은 I 슬라이스들, P 슬라이스들, SP 슬라이스들 및 B 슬라이스들에서의 비디오 유닛들에 대해 인트라 예측을 수행할 수도 있다.

[0129] 현재 비디오 유닛에 대해 인트라 예측을 수행하기 위해, 인트라 예측 프로세싱 유닛 (126) 은 현재 비디오 유닛에 대한 예측 데이터의 다수의 세트들을 생성하기 위해 다수의 인트라 예측 모드들을 사용할 수도 있다. 인트라 예측 모드를 사용하여 현재 비디오 유닛에 대한 예측 데이터의 세트를 생성하기 위해, 인트라 예측 프로세싱 유닛 (126) 은 인트라 예측 모드에 연관된 방향에서 현재 비디오 유닛의 블록들 전체에 걸쳐 이웃하는 블록들로부터 샘플들을 확장할 수도 있다. 비디오 유닛들에 대한 좌측에서 우측으로, 상단에서 하단으로의 인코딩 순서를 가정하면, 이웃하는 블록들은 현재 비디오 유닛의 블록들의 상측, 우상측, 좌상측, 또는 좌측에 있을 수도 있다. 일부 예들에서, 인트라 예측 모드들의 수는 현재 비디오 유닛의 블록들의 사이즈에 의존할 수도 있다. 인트라 예측 프로세싱 유닛 (126) 은 현재 비디오 유닛에 대한 예측 블록들을 선택하기 위해 인트라 예측 모드들 중 하나를 선택한다.

[0130] 예측 프로세싱 유닛 (100) 은 현재 비디오 유닛에 대해 인터 예측 프로세싱 유닛 (121) 에 의해 생성된 예측 데이터 또는 그 현재 비디오 유닛에 대해 인트라 예측 프로세싱 유닛 (126) 에 의해 생성된 예측 데이터 중에서 현재 비디오 유닛에 대한 예측 데이터를 선택할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (100) 은 예측 데이터의 세트들의 레이트/왜곡 메트릭들에 기초하여 현재 비디오 유닛에 대한 예측 데이터를 선택한다.

[0131] 잔차 생성 유닛 (102) 은 예측 루마 및 크로마 블록들에서의 샘플들을 현재 비디오 유닛의 루마 및 크로마 블록들의 대응하는 샘플들로부터 감산함으로써 잔차 루마 및 크로마 블록들을 생성할 수도 있다. 변환 프로세싱 유닛 (104) 은 하나 이상의 변환들을 잔차 블록에 적용함으로써 각각의 잔차 블록에 대한 변환 계수 블록들을 생성할 수도 있다. 변환 프로세싱 유닛 (104) 은 다양한 변환들을 잔차 블록에 적용할 수도 있다. 예를 들어, 변환 프로세싱 유닛 (104) 은 이산 코사인 변환 (DCT), 방향성 변환, 또는 개념적으로 유사한 변환을 잔차 블록에 적용할 수도 있다.

[0132] 양자화 유닛 (106) 은 변환 계수 블록에서의 변환 계수들을 양자화할 수도 있다. 양자화 프로세스는 변환 계수들의 일부 또는 전부에 연관된 비트 깊이를 감소시킬 수도 있다. 예를 들어, n-비트 변환 계수는 양자화 동안에 m-비트 변환 계수로 버림될 (rounded down) 수도 있으며, 여기서 n은 m보다 크다. 양자화 유닛 (106) 은 양자화 파라미터 (QP) 값에 기초하여 변환 계수 블록을 양자화할 수도 있다. 비디오 인코더 (20) 는 QP 값을 조정함으로써 변환 계수 블록들에 적용되는 양자화 정도를 조정할 수도 있다.

[0133] 역 양자화 유닛 (108) 과 역 변환 프로세싱 유닛 (110) 은 역 양자화 및 역 변환들을 변환 계수 블록에 각각 적용하여, 변환 계수 블록들로부터 잔차 블록을 복원할 수도 있다. 복원 유닛 (112) 은 복원된 잔차 블록들에서의 샘플들을 예측 프로세싱 유닛 (100) 에 의해 생성된 하나 이상의 예측 블록들로부터의 대응하는 샘플들에 가산하여 복원된 블록들을 생성할 수도 있다. 필터 유닛 (113) 은 블록화제거 동작을 수행하여 복원된 블록들에서의 블록화 아티팩트들을 감소시킬 수도 있다. 디코딩된 화상 버퍼 (114) 는, 필터 유닛 (113) 이 복원된 블록들에 대해 하나 이상의 블록화제거 동작들을 수행한 후에 복원된 블록들을 저장할 수도 있다. 모션 추정 유닛 (122) 과 모션 보상 유닛 (124) 은 후속 화상들의 비디오 유닛들에 대해 인터 예측을 수행하기 위해 복원된 블록들을 포함하는 참조 화상을 사용할 수도 있다. 덧붙여서, 인트라 예측 프로세싱 유닛 (126) 은 디코딩된 화상 버퍼 (114) 에서의 복원된 블록들을 사용하여 인트라 예측을 수행할 수도 있다.

- [0134] 엔트로피 인코딩 유닛 (116)은 비디오 인코더 (20)의 다른 기능성 컴포넌트들로부터 데이터를 수신할 수도 있다. 예를 들어, 엔트로피 인코딩 유닛 (116)은 양자화 유닛 (106)으로부터 변환 계수 블록들을 수신할 수도 있고 예측 프로세싱 유닛 (100)으로부터 선택스 엘리먼트들을 수신할 수도 있다. 엔트로피 인코딩 유닛 (116)은 데이터에 대해 하나 이상의 엔트로피 인코딩 동작들을 수행하여 엔트로피 인코딩된 데이터를 생성할 수도 있다. 예를 들어, 비디오 인코더 (20)는 그 데이터에 대해 CAVLC 동작, CABAC 동작, 가변 대 가변 (variable-to-variable; V2V) 길이 코딩 동작, 선택스 기반 컨텍스트 적응 이진 산술 코딩 (SBAC) 동작, 확률 간격 구획화 엔트로피 (PIPE) 코딩 동작, 지수-골롬 코딩 동작, 또는 다른 유형의 엔트로피 인코딩 동작을 수행할 수도 있다.
- [0135] 도 3은 본 개시물에서 설명되는 기법들을 구현할 수도 있는 일 예의 비디오 디코더 (30)를 도시하는 블록도이다. 도 3은 설명의 목적으로 제공되고 본 개시물에서 폭넓게 예시되고 설명된 바와 같은 기법들로 제한하고 있지는 않다. 설명의 목적으로, 본 개시물은 H.264/AVC 코딩의 맥락에서 비디오 디코더 (30)를 설명한다. 그러나, 본 개시물의 기법들은 다른 코딩 표준들 또는 방법들에 적용가능할 수도 있다.
- [0136] 도 3의 예에서, 비디오 디코더 (30)는 엔트로피 디코딩 유닛 (150), 예측 프로세싱 유닛 (152), 역 양자화 유닛 (154), 역 변환 프로세싱 유닛 (156), 복원 유닛 (158), 필터 유닛 (159), 및 디코딩된 화상 버퍼 (160)를 구비한다. 예측 프로세싱 유닛 (152)은 모션 보상 유닛 (162)과 인트라 예측 프로세싱 유닛 (164)을 구비한다. 다른 예들에서, 비디오 디코더 (30)는 더 많거나, 더 적거나, 또는 상이한 기능성 컴포넌트들을 포함할 수도 있다.
- [0137] 비디오 디코더 (30)는 비트스트림을 수신한다. 엔트로피 디코딩 유닛 (150)은 비트스트림을 파싱하여 비트스트림으로부터 선택스 엘리먼트들을 추출할 수도 있다. 비트스트림을 파싱하는 부분으로서, 엔트로피 디코딩 유닛 (150)은 비트스트림에서의 엔트로피 인코딩된 선택스 엘리먼트들을 엔트로피 디코딩할 수도 있다. 예측 프로세싱 유닛 (152), 역 양자화 유닛 (154), 역 변환 프로세싱 유닛 (156), 복원 유닛 (158), 및 필터 유닛 (159)은 비트스트림으로부터 추출된 선택스 엘리먼트들에 기초하여 디코딩된 비디오 데이터를 생성 (즉, 그 비디오 데이터를 복원)할 수도 있다. 비트스트림으로부터 추출된 선택스 엘리먼트들은 변환 계수 블록들을 나타내는 선택스 엘리먼트들을 포함할 수도 있다.
- [0138] 역 양자화 유닛 (154)은 변환 계수 블록들을 역 양자화, 즉 양자화해제 (de-quantization)할 수도 있다. 역 양자화 유닛 (154)은 QP 값을 사용하여 양자화 정도를 결정하고, 비슷하게, 역 양자화 유닛 (154)에 대해 적용할 역 양자화 정도를 결정할 수도 있다. 역 양자화 유닛 (154)이 변환 계수 블록을 역 양자화한 후, 역 변환 프로세싱 유닛 (156)은 잔차 블록을 생성하기 위하여 하나 이상의 역 변환들을 변환 계수 블록에 적용할 수도 있다. 예를 들어, 역 변환 프로세싱 유닛 (156)은 역 DCT, 역 정수 변환, 역 카루넨-로베 변환 (Karhunen-Loeve transform; KLT), 역 회전 변환, 역 방향성 변환, 또는 다른 역 변환을 변환 계수 블록에 적용할 수도 있다.
- [0139] 현재 비디오 유닛 (예컨대, MB 또는 MB 구획)가 인트라 예측을 사용하여 인코딩되면, 인트라 예측 프로세싱 유닛 (164)은 인트라 예측을 수행하여 현재 비디오 유닛에 대해 예측 루마 및 크로마 블록들을 생성할 수도 있다. 예를 들어, 인트라 예측 프로세싱 유닛 (164)은 비트스트림에서의 선택스 엘리먼트들에 적어도 부분적으로 기초하여 현재 비디오 유닛에 대한 인트라 예측 모드를 결정할 수도 있다. 인트라 예측 프로세싱 유닛 (164)은 인트라 예측 모드를 사용하여 공간적으로 이웃하는 블록들에 기초하여 현재 비디오 유닛에 대한 예측 블록들을 생성할 수도 있다.
- [0140] 예측 프로세싱 유닛 (152)은 비트스트림으로부터 추출된 선택스 엘리먼트들에 기초하여 제 1 참조 화상 리스트 (리스트 0) 및 제 2 참조 화상 리스트 (리스트 1)를 구성할 수도 있다. 비트스트림이 MVC 또는 MVC-호환 3DV를 사용하여 인코딩되는 예들에서, 리스트 0 및/또는 리스트 1은 뷰간 참조 화상들을 포함할 수도 있다. 예측 프로세싱 유닛 (152)은 도 2의 인트라 예측 프로세싱 유닛 (121)에 관해 위에서 설명된 바와 동일한 방식으로 참조 화상 리스트들을 구성할 수도 있다.
- [0141] 더욱이, 현재 비디오 유닛 (예컨대, MB 또는 MB 구획)가 인트라 예측을 사용하여 인코딩되면, 엔트로피 디코딩 유닛 (150)은, 비트스트림으로부터, 현재 비디오 유닛에 대한 모션 정보를 추출할 수도 있다. 모션 보상 유닛 (162)은, 현재 비디오 유닛의 모션 정보에 적어도 부분적으로 기초하여, 현재 비디오 유닛에 대한 하나 이상의 참조 블록들을 결정할 수도 있다. 모션 보상 유닛 (162)은, 현재 비디오 유닛에 대한 하나 이상의 참조 블록들에 적어도 부분적으로 기초하여, 현재 비디오 유닛에 대한 예측 블록들을 생성할 수도 있다.

- [0142] 복원 유닛 (158) 은, 현재 비디오 유닛에 대한 잔차 루마 및 크로마 블록들과 현재 비디오 유닛의 예측 루마 및 크로마 블록들에 적어도 부분적으로 기초하여, 현재 비디오 유닛에 대한 루마 및 크로마 블록들을 복원할 수도 있다. 특히, 복원 유닛 (158) 은 잔차 블록들의 샘플들 (예컨대, 루마 또는 크로마 성분들) 을 예측 블록들의 대응하는 샘플들에 가산하여 현재 비디오 유닛의 루마 및 크로마 성분들을 복원할 수도 있다. 필터 유닛 (159) 은 블록화제거 동작을 수행하여 현재 비디오 유닛의 복원된 블록들에 연관된 블록화 아티팩트들을 감소시킬 수도 있다. 비디오 디코더 (30) 는 복원된 블록들을 디코딩된 화상 버퍼 (160) 에 저장할 수도 있다. 디코딩된 화상 버퍼 (160) 는 후속하는 모션 보상, 인트라 예측, 및 디스플레이 디바이스, 이를테면 도 1의 디스플레이 디바이스 (32) 상의 프레젠테이션을 위해 참조 화상들을 제공할 수도 있다. 예를 들면, 비디오 디코더 (30) 는, 디코딩된 화상 버퍼 (160) 에서의 블록들에 기초하여, 다른 CU들의 PU들에 대해 인트라 예측 또는 인터 예측 동작들을 수행할 수도 있다.
- [0143] 도 4는 본 개시물의 하나 이상의 기법들에 따른, 일 예의 서브-비트스트림 추출 동작 (200) 을 도시하는 흐름도이다. 도 4의 흐름도 및 다음의 도면들의 흐름도들은 예들로서 제공된다. 다른 예들에서, 본 개시물의 기법들은 도 4의 예 및 다음의 도면들에서 도시된 것들에 비해 많거나, 적거나, 또는 상이한 단계들을 사용하여 구현될 수도 있다.
- [0144] 도 4의 예에서 도시된 바와 같이, CDN 디바이스 (16) 는 복수의 타겟 뷰들에서의 화상들을 디코딩하기 위해 요구되는 텍스처 뷰 성분들을 갖는 3DV 비트스트림에서의 뷰들을 나타내는 텍스처 타겟 뷰 리스트를 결정할 수도 있다 (202). 타겟 뷰들은 서브-비트스트림으로부터 디코딩가능한 3DV 비트스트림에서의 뷰들의 서브세트일 수도 있다. 덧붙여서, CDN 디바이스 (16) 는 복수의 타겟 뷰들에서의 화상들을 디코딩하기 위해 요구되는 깊이 뷰 성분들을 갖는 3DV 비트스트림에서의 뷰들을 나타내는 깊이 타겟 뷰 리스트를 결정할 수도 있다 (204). CDN 디바이스 (16) 는 텍스처 타겟 뷰 리스트 및 깊이 타겟 뷰 리스트에 적어도 부분적으로 기초하여 서브-비트스트림을 결정할 수도 있다 (206).
- [0145] 일부 예들에서, 텍스처 타겟 뷰 리스트는, 복수의 타겟 뷰들에서의 화상들을 디코딩하기 위해 요구되는 텍스처 뷰 성분들을 갖는 뷰들을 3DV 비트스트림에서 식별하는 뷰 식별자들을 포함한다. 더욱이, 깊이 타겟 뷰 리스트는, 복수의 타겟 뷰들에서의 화상들을 디코딩하기 위해 요구되는 깊이 뷰 성분들을 갖는 뷰들을 3DV 비트스트림에서 식별하는 뷰 식별자들을 포함할 수도 있다. 텍스처 타겟 뷰 리스트는 깊이 타겟 뷰 리스트에서의 뷰 식별자들과는 상이한 하나 이상의 뷰 식별자들을 포함할 수도 있다.
- [0146] 위에서 논의된 바와 같이, 3DV 비트스트림은 일련의 NAL 유닛들을 포함할 수도 있다. 일부 예들에서, CDN 디바이스 (16) 는, 텍스처 타겟 뷰 리스트 또는 깊이 타겟 뷰 리스트가 NAL 유닛의 뷰 식별자를 명시하는지 여부에 적어도 부분적으로 기초하여, 3DV 비트스트림으로부터 NAL 유닛을 제거할지 여부를 결정할 수도 있다.
- [0147] 도 5는 본 개시물의 하나 이상의 기법들에 따른, MVC-호환 3DV에서의 일 예의 서브-비트스트림 추출 프로세스 (298) 를 도시하는 흐름도이다. 서브-비트스트림 추출 프로세스 (298) 는 도 4의 서브-비트스트림 추출 프로세스 (200) 의 더 구체적인 예일 수도 있다. 도 5의 예에서, 텍스처 뷰 성분들에 대한 타겟 뷰 리스트 (예컨대, 텍스처 타겟 뷰 리스트) 는 2 개의 부분들, 즉 앵커 텍스처에 필요한 뷰들의 리스트 및 비 앵커 텍스처에 필요한 뷰들의 리스트로 나누어진다. 마찬가지로, 도 5의 예에서, 깊이 뷰 성분들에 대한 타겟 뷰 리스트 (예컨대, 깊이 타겟 뷰 리스트) 는 2 개의 부분들, 즉 앵커 깊이에 필요한 뷰들의 리스트 및 비 앵커 깊이에 필요한 뷰들의 리스트로 나누어진다.
- [0148] 서브-비트스트림 추출 프로세스 (298) 는 여러 변수들을 입력들로서 취할 수도 있다. 이들 입력들은 깊이 존재 플래그 타겟 (depthPresentFlagTarget), 변수 pIdTarget, 변수 tIdTarget, 및 뷰 식별자 타겟 리스트 (viewIdTargetList) 를 포함할 수도 있다. 뷰 식별자 타겟 리스트는 viewIdTarget의 하나 이상의 값들로 구성될 수도 있다. viewIdTarget의 값들은 타겟 서브-비트스트림 (즉, 서브-비트스트림 추출 프로세스 (298) 에 의해 추출된 서브-비트스트림) 에 포함될 뷰들의 뷰 식별자들을 포함할 수도 있다. depthPresentFlagTarget은 깊이 뷰들이 타겟 서브-비트스트림에 포함되는지 여부를 나타낼 수도 있다. depthPresentFlagTarget이 입력으로서 존재하지 않는 경우, CDN 디바이스 (16) 는 depthPresentFlagTarget이 0과 동일하다고 유추 (즉, 자동으로 결정) 할 수도 있다. pIdTarget은 타겟 우선순위 레벨을 식별할 수도 있다. pIdTarget이 입력으로서 존재하지 않는 경우, CDN 디바이스 (16) 는 pIdTarget이 63과 동일하다고 유추할 수도 있다. tIdTarget은 타겟 시간적 레벨을 식별할 수도 있다. tIdTarget이 입력으로서 존재하지 않는 경우, CDN 디바이스 (16) 는 tIdTarget이 7과 동일하다고 유추할 수도 있다. viewIdTargetList가 입력으로서 존재하지 않는 경우, CDN 디바이스 (16) 는 viewIdTarget의 하나의 값이 viewIdTargetList에 있다는

것과 viewIdTarget의 값이 기본 뷰의 뷰 식별자 (view_id) 와 동일하다고 유추할 수도 있다.

[0149] 서브-비트스트림 추출 프로세스 (298) 의 출력들은 서브-비트스트림과 뷰 순서 인덱스 값들의 리스트 (VOIdxList) 를 포함할 수도 있다. 일부 예들에서, pIdTarget이 0 내지 63의 범위의 임의의 값과 동일하며 tIdTarget이 0 내지 7의 범위의 임의의 값과 동일하며 viewIdTargetList는 비트스트림에서의 뷰를 식별하는 viewIdTarget의 임의의 하나 이상의 값들로 구성되는, 서브-비트스트림 추출 프로세스 (298) 의 출력인 임의의 서브-비트스트림이 H.264/AVC 표준에 부합한다는 비트스트림 적합성의 조건이 있을 수도 있다. 부합하는 비트스트림은 0과 동일한 priority_id 및 0과 동일한 temporal_id를 갖는 하나 이상의 코딩된 슬라이스 NAL 유닛들을 포함한다. 서브-비트스트림 추출 프로세스 (298) 의 결과로 생긴 서브-비트스트림의 모든 동작 포인트들이 적용가능한 level_idc 또는 level_idc[i]를 가지지는 않는다는 것이 가능할 수도 있다. 이 경우, 서브-비트스트림에서의 각각의 코딩된 비디오 시퀀스는 H.264/AVC 표준의 부록 A, 부록 H 및 부록 I에서 특정된 프로파일들 중 하나 이상에 여전히 부합할 수도 있지만 (또는 부합해야 하지만), H.264/AVC 표준의 하위절들 A.3, H.1.0.2 및 I.1.0.2에서 각각 특정된 레벨 제약조건들을 충족시키지 못할 수도 있다.

[0150] 도 5의 예에서, CDN 디바이스 (16) 는 비어있는 뷰 순서 인덱스 리스트 (VOIdxList) 를 생성할 수도 있고 기본 뷰의 뷰 순서 인덱스가 되도록 최소 뷰 순서 인덱스 (minVOIdx) 를 초기화할 수도 있다 (300). 다시 말하면, CDN 디바이스 (16) 는 VOIdxList를 빈 채로 둘 수도 있고 minVOIdx를 기본 뷰의 VOIdx 값으로 둘 수도 있다.

[0151] 덧붙여서, CDN 디바이스 (16) 는 앵커 텍스처 타겟 뷰 리스트를 생성할 수도 있다 (302). 앵커 텍스처 타겟 뷰 리스트는 타겟 뷰들의 앵커 화상들을 디코딩하기 위해 요구되는 텍스처 뷰 성분들을 갖는 뷰들의 리스트일 수도 있다. 본 개시물은 앵커 텍스처 타겟 뷰 리스트에서의 뷰들을 "앵커 텍스처를 위해 요청됨"으로서 마킹된 뷰들로서 지칭할 수도 있다. CDN 디바이스 (16) 가 앵커 텍스처 타겟 뷰 리스트를 생성하는 경우, CDN 디바이스 (16) 는 "앵커 텍스처를 위해 요청됨"으로서 마킹된 뷰들의 뷰 순서 인덱스들을 뷰 순서 인덱스 리스트 (VOIdxList) 에 포함시킬 수도 있다. 본 개시물에서의 다른 곳에서 설명되는 바와 같이, CDN 디바이스 (16) 는, 앵커 텍스처 타겟 뷰 리스트에 적어도 부분적으로 기초하여, 비트스트림으로부터 제거할 VCL NAL 유닛들 및 필터 데이터 NAL 유닛들을 결정할 수도 있다. 다양한 예들에서, CDN 디바이스 (16) 는 앵커 텍스처 타겟 뷰 리스트를 다양한 방도들로 생성할 수도 있다. 예를 들어, CDN 디바이스 (16) 는, viewIdTargetList에 포함된 viewIdTarget의 각각의 값에 대해, viewIdTarget을 입력으로 하여 앵커 텍스처 타겟 뷰 리스트 도출 프로세스를 호출할 수도 있다. 본 개시물에서 나중에 상세히 설명되는 도 7은, 일 예의 앵커 텍스처 타겟 뷰 리스트 도출 프로세스의 흐름도이다.

[0152] 앵커 텍스처 타겟 뷰 리스트를 생성한 후, CDN 디바이스 (16) 는 깊이 뷰 성분들이 서브-비트스트림에 포함될 것인지 여부를 결정할 수도 있다 (304). 도 5의 예에서, CDN 디바이스 (16) 는, depthPresentFlagTarget에 기초하여, 깊이 뷰 성분들이 서브-비트스트림에 포함될 것인지 여부를 결정할 수도 있다. 깊이 뷰 성분들이 서브-비트스트림에 포함될 것이라는 결정에 응답하여 (304의 "예"), CDN 디바이스 (16) 는 앵커 깊이 타겟 뷰 리스트를 생성할 수도 있다 (306). 앵커 깊이 타겟 뷰 리스트는 타겟 뷰들의 앵커 화상들을 디코딩하기 위해 요구되는 깊이 뷰 성분들을 갖는 뷰들의 리스트일 수도 있다. 본 개시물은 앵커 깊이 타겟 뷰 리스트에서의 뷰들을 "앵커 깊이를 위해 요청됨"으로서 마킹된 뷰들로서 지칭할 수도 있다. CDN 디바이스 (16) 가 앵커 깊이 타겟 뷰 리스트를 생성하는 경우, CDN 디바이스 (16) 는 "앵커 깊이를 위해 요청됨"으로서 마킹된 뷰들의 뷰 순서 인덱스들을 뷰 순서 인덱스 리스트 (VOIdxList) 에 포함할 수도 있다. 본 개시물에서의 다른 곳에서 설명되는 바와 같이, CDN 디바이스 (16) 는, 앵커 깊이 타겟 뷰 리스트에 적어도 부분적으로 기초하여, 비트스트림으로부터 제거할 VCL NAL 유닛들 및 필터 데이터 NAL 유닛들을 결정할 수도 있다. 다양한 예들에서, CDN 디바이스 (16) 는 앵커 깊이 타겟 뷰 리스트를 다양한 방도들로 생성할 수도 있다. 예를 들어, depthPresentFlagTarget이 1과 동일하면, CDN 디바이스 (16) 는, viewIdTargetList에서의 각각의 viewIdTarget에 대해, viewIdTarget을 입력으로 하여 앵커 깊이 타겟 뷰 리스트 도출 프로세스를 수행할 수도 있다. 본 개시물에서 나중에 상세히 설명되는 도 8은, 일 예의 앵커 깊이 타겟 뷰 리스트 도출 프로세스의 흐름도이다.

[0153] 앵커 깊이 타겟 뷰 리스트를 생성한 후 또는 깊이 뷰 성분들이 서브-비트스트림에 포함될 것이 아니라는 결정에 응답하여 (304의 "아니오"), CDN 디바이스 (16) 는 비 앵커 텍스처 타겟 뷰 리스트를 생성할 수도 있다 (308). 비 앵커 텍스처 타겟 뷰 리스트는 타겟 뷰들의 비 앵커 화상들을 디코딩하기 위해 요구되는 텍스처 뷰 성분들을 갖는 뷰들의 리스트일 수도 있다. 본 개시물은 비 앵커 텍스처 타겟 뷰 리스트에서의 뷰들을 "비 앵커 텍스처를 위해 요청됨"으로서 마킹된 뷰들로서 지칭할 수도 있다. CDN 디바이스 (16) 가 비 앵커 텍스처 타겟 뷰 리스트를 생성하는 경우, CDN 디바이스 (16) 는 "비 앵커 텍스처를 위해 요청됨"으로서 마킹된 뷰들의 뷰

순서 인덱스들을 뷰 순서 인덱스 리스트 (VOIdxList)에 포함할 수도 있다. 본 개시물에서의 다른 곳에서 설명되는 바와 같이, CDN 디바이스 (16)는, 비 앵커 텍스트 타겟 뷰 리스트에 적어도 부분적으로 기초하여, 비트스트림으로부터 제거할 VCL NAL 유닛들 및 필터 데이터 NAL 유닛들을 결정할 수도 있다. 다양한 예들에서, CDN 디바이스 (16)는 비 앵커 텍스처 타겟 뷰 리스트를 다양한 방도들로 생성할 수도 있다. 예를 들어, CDN 디바이스 (16)는, viewIdTargetList에 포함된 viewIdTarget의 각각의 값에 대해, viewIdTarget을 입력으로 하여 비 앵커 텍스처 타겟 뷰 리스트 도출 프로세스를 호출할 수도 있다. 본 개시물에서 나중에 상세히 설명되는 도 9는, 일 예의 비 앵커 텍스처 타겟 뷰 리스트 도출 프로세스의 흐름도이다.

[0154] 덧붙여서, CDN 디바이스 (16)는 깊이 뷰 성분들이 서브-비트스트림에 포함될 것인지 여부를 결정할 수도 있다 (310). 깊이 뷰 성분들이 서브-비트스트림에 포함될 것이라는 결정에 응답하여 (310의 "예"), CDN 디바이스 (16)는 비 앵커 깊이 타겟 뷰 리스트를 생성할 수도 있다 (312). 비 앵커 깊이 타겟 뷰 리스트는 타겟 뷰들의 비 앵커 화상들을 디코딩하기 위해 요구되는 깊이 뷰 성분들을 갖는 뷰들의 리스트일 수도 있다. 본 개시물은 비 앵커 깊이 타겟 뷰 리스트에서의 뷰들을 "비 앵커 깊이를 위해 요청됨"으로서 마킹된 뷰들로서 지칭할 수도 있다. CDN 디바이스 (16)가 비 앵커 깊이 타겟 뷰 리스트를 생성하는 경우, CDN 디바이스 (16)는 "비 앵커 깊이를 위해 요청됨"으로서 마킹된 뷰들의 뷰 순서 인덱스들을 뷰 순서 인덱스 리스트 (VOIdxList)에 포함할 수도 있다. 본 개시물에서의 다른 곳에서 설명되는 바와 같이, CDN 디바이스 (16)는, 비 앵커 깊이 타겟 뷰 리스트에 적어도 부분적으로 기초하여, 비트스트림으로부터 제거할 VCL NAL 유닛들 및 필터 데이터 NAL 유닛들을 결정할 수도 있다. 다양한 예들에서, CDN 디바이스 (16)는 비 앵커 깊이 타겟 뷰 리스트를 다양한 방도들로 생성할 수도 있다. 예를 들어, depthPresentFlagTarget이 1과 동일하면, CDN 디바이스 (16)는, viewIdTargetList에 포함된 viewIdTarget의 각각의 값에 대해, viewIdTarget을 입력으로 하여 비 앵커 깊이 타겟 뷰 리스트 도출 프로세스를 호출할 수도 있다. 본 개시물에서 나중에 상세히 설명되는 도 10은, 일 예의 비 앵커 텍스처 타겟 뷰 리스트 도출 프로세스의 흐름도이다.

[0155] 비 앵커 깊이 타겟 뷰 리스트를 생성한 후 또는 깊이 뷰 성분들이 서브-비트스트림에 포함될 것이라는 결정에 응답하여 (310의 "아니오"), CDN 디바이스 (16)는 적용가능한 VCL NAL 유닛들 및 필터 데이터 NAL 유닛들을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다 (314). CDN 디바이스 (16)는 적용가능한 VAL 및 필터 데이터 NAL 유닛들을 다양한 방도들로 결정할 수도 있다. 예를 들어, CDN 디바이스 (16)는 도 11의 예의 동작 또는 도 12의 예의 동작을 수행함으로써 적용가능한 VAL 및 필터 데이터 NAL 유닛들을 결정할 수도 있다. 도 11 및 도 12는 본 개시물에서 나중에 상세히 설명된다.

[0156] 다음으로, CDN 디바이스 (16)는, 액세스 유닛의 모든 VCL NAL 유닛들이 "비트스트림으로부터 제거될 것"으로서 마킹된 각각의 액세스 유닛을 비트스트림으로부터 제거할 수도 있다 (316). 덧붙여서, CDN 디바이스 (16)는, "비트스트림으로부터 제거될 것"으로서 마킹된 모든 VCL NAL 유닛들 및 모든 필터 데이터 NAL 유닛들을 비트스트림으로부터 제거할 수도 있다 (318).

[0157] CDN 디바이스 (16)는 그 다음에 뷰 순서 인덱스 리스트 (VOIdxList)가 최소 뷰 순서 인덱스 (minVOIdx)와 동일한 하나의 뷰 순서 인덱스 (VOIdx)만을 포함하는지 여부를 결정할 수도 있다 (320). 위에서 나타난 바와 같이, 뷰 순서 인덱스 리스트는 "앵커 텍스처를 위해 요청됨", "앵커 깊이를 위해 요청됨", "비 앵커 텍스처를 위해 요청됨", 및 "비 앵커 깊이를 위해 요청됨"으로서 마킹된 뷰들의 뷰 순서 인덱스들을 포함할 수도 있다. VOIdxList가 minVOIdx와 동일한 VOIdx의 하나의 값만을 포함하는 경우, 서브-비트스트림은 기본 뷰만을 또는 기본 뷰의 시간적 서브세트만을 포함한다.

[0158] 뷰 순서 인덱스 리스트 (VOIdxList)가 최소 뷰 순서 인덱스 (minVOIdx)와 동일한 하나의 뷰 순서 인덱스 (VOIdx)만을 포함한다는 결정에 응답하여 (320의 "예"), CDN 디바이스 (16)는, 비트스트림으로부터, 모든 접두 NAL 유닛들 및 모든 서브 세트 시퀀스 파라미터 세트 NAL 유닛들 (즉, 각각 14 또는 15와 동일한 nal_unit_type을 갖는 NAL 유닛들)을 제거할 수도 있다 (322).

[0159] 덧붙여서, CDN 디바이스 (16)는, 비트스트림으로부터, SEI NAL 유닛의 제 1 SEI 메시지가 36 내지 44의 범위의 페이로드 유형을 가지는 각각의 SEI NAL 유닛 (즉, 6과 동일한 nal_unit_type을 갖는 NAL 유닛들)를 제거할 수도 있다 (324). 이런 식으로, VOIdxList가 minVOIdx와 동일한 VOIdx의 하나의 값만을 포함하는 경우, CDN 디바이스 (16)는 제 1 SEI 메시지가 36 내지 44의 범위의 payloadType을 가지는 다음의 NAL 유닛들, 즉, 14 또는 15와 동일한 nal_unit_type을 갖는 모든 NAL 유닛들 및 6과 동일한 nal_unit_type을 갖는 모든 NAL 유닛들을 제거할 수도 있다.

[0160] 페이로드 유형 36을 갖는 SEI 메시지들은 병렬 디코딩 정보를 포함한다. 페이로드 유형 37을 갖는 SEI 메시

지들은 MVC 스케일러블 네스팅 정보를 포함한다. 패이로드 유형 38을 갖는 SEI 메시지들은 뷰 확장성 정보를 포함한다. 패이로드 유형 39를 갖는 SEI 메시지들은 멀티뷰 장면 정보를 포함한다. 패이로드 유형 40을 갖는 SEI 메시지들은 멀티뷰 획득 정보를 포함한다. 패이로드 유형 41을 갖는 SEI 메시지들은 요청되지 않은 뷰 성분 정보를 포함한다. 패이로드 유형 42를 갖는 SEI 메시지들은 뷰 의존도 변화 정보를 포함한다. 패이로드 유형 43을 갖는 SEI 메시지들은 정보를 제시하는 않는 동작 포인트들을 포함한다. 패이로드 유형 44를 갖는 SEI 메시지들은 기본 뷰 임시 가상 참조 디코더 (hypothetical reference decoder; HRD) 정보를 포함한다.

[0161] SEI NAL 유닛의 제 1 SEI 메시지가 36 내지 44의 범위의 패이로드 유형을 가지는 각각의 SEI NAL 유닛을 제거한 후 또는 뷰 순서 인덱스 리스트 (VOIdxList) 가 최소 뷰 순서 인덱스 (minVOIdx) 와 동일한 하나의 뷰 순서 인덱스 (VOIdx) 만을 포함하지 않는다는 결정에 응답하여 (320의 "아니오"), CDN 디바이스 (16) 는 깊이 뷰 성분들이 서브-비트스트림에 포함될 것인지 여부를 결정할 수도 있다 (326). 깊이 뷰 성분들이 서브-비트스트림에 포함될 것이 아니라는 결정에 응답하여 (326의 "아니오"), CDN 디바이스 (16) 는, 비트스트림으로부터, 제 1 SEI 메시지가 45 내지 47의 범위의 패이로드 유형을 가지는 각각의 SEI NAL 유닛 (즉, 6과 동일한 nal_unit_type을 가지는 각각의 NAL 유닛) 를 제거할 수도 있다 (328). 예를 들면, depthPresentFlagTarget이 0과 동일한 경우, CDN 디바이스 (16) 는, 제 1 SEI 메시지가 45 내지 47의 범위의 payloadType을 가지는 6과 동일한 nal_unit_type을 갖는 모든 NAL 유닛들을 제거할 수도 있다. 패이로드 유형 45를 갖는 SEI 메시지들은 3DV 스케일러블 네스팅 정보를 포함한다. 패이로드 유형 46을 갖는 SEI 메시지들은 3D 뷰 확장성 정보를 포함한다. 패이로드 유형 47을 갖는 SEI 메시지들은 3DV 획득 정보를 포함한다.

[0162] 45 내지 47의 범위의 패이로드 유형을 갖는 제 1 SEI 메시지를 가지는 각각의 SEI NAL 유닛을 제거한 후 (328) 또는 깊이 뷰 성분들이 서브-비트스트림에 포함될 것이라는 결정에 응답하여 (326의 "예"), CDN 디바이스 (16) 는 도 6에 도시된 서브-비트스트림 추출 프로세스 (298) 의 연속 (continuation) 을 수행할 수도 있다.

[0163] 도 6은 도 5의 서브-비트스트림 추출 프로세스 (298) 의 연속을 도시하는 흐름도이다. 도 6의 예에서 도시된 바와 같이, CDN 디바이스 (16) 는, 비트스트림으로부터, 적용가능한 MVC 스케일러블 네스팅 메시지의 부분인 SEI 메시지들만을 포함하는 모든 SEI NAL 유닛들 (즉, 6과 동일한 nal_unit_type을 갖는 NAL 유닛들) 을 제거할 수도 있다 (330). 만약 MVC 스케일러블 네스팅 메시지의 operation_point_flag 선택스 엘리먼트가 0과 동일하며, MVC 스케일러블 네스팅 메시지의 all_view_components_in_au_flag 선택스 엘리먼트가 0과 동일하고, MVC 스케일러블 네스팅 메시지의 sei_view_id[i] 선택스 엘리먼트들 중 어느 것도, 0 내지 num_view_components_minus1의 범위의 모든 i에 대해, 뷰 순서 인덱스 리스트 (VOIdxList) 에 포함된 뷰 순서 인덱스 (VOIdx) 에 대응하지 않으면, MVC 스케일러블 네스팅 메시지는 적용가능한 MVC 스케일러블 네스팅 메시지일 수도 있다. 더욱이, 만약 MVC 스케일러블 네스팅 메시지의 operation_point_flag 선택스 엘리먼트가 1과 동일하고, MVC 스케일러블 네스팅 메시지의 sei_op_temporal_id 선택스 엘리먼트가 maxTid보다 크거나 또는 0 내지 num_view_components_op_minus1의 범위의 모든 i에 대한 MVC 스케일러블 네스팅 메시지에서 특정된 sei_op_view_id[i]의 리스트가 viewIdTargetList의 서브세트가 아니면, MVC 스케일러블 네스팅 메시지는 적용가능한 MVC 스케일러블 네스팅 메시지일 수도 있다. MaxTid는 모든 남아있는 VCL NAL 유닛들의 최대 temporal_id일 수도 있다. 다르게 말하면, 만약 MVC 스케일러블 네스팅 메시지의 operation_point_flag가 1과 동일하고 0 내지 num_view_components_minus1의 범위에서의 임의의 i에 대한 sei_op_view_id[i]가 viewIdTargetList에서의 값과 동일하다는 것이 참이 아니면, MVC 스케일러블 네스팅 메시지는 적용가능한 MVC 스케일러블 네스팅 메시지일 수도 있다.

[0164] 이런 식으로, CDN 디바이스 (16) 는 maxTid를 모든 남아있는 VCL NAL 유닛들의 최대 temporal_id로 둘 수도 있다. 더욱이, CDN 디바이스 (16) 는 다음의 속성들 중 임의의 것을 갖는 MVC 스케일러블 네스팅 SEI 메시지의 부분인 SEI 메시지들만을 포함하는 6과 동일한 nal_unit_type을 갖는 모든 NAL 유닛들을 제거할 수도 있다:

[0165] - operation_point_flag는 0과 동일하고 all_view_components_in_au_flag는 0과 동일하고 0 내지 num_view_components_minus1의 범위의 모든 i에 대해, sei_view_id[i] 중 어느 것도 VOIdxList에 포함된 VOIdx 값에 대응하지 않으며,

[0166] - operation_point_flag는 1과 동일하고, sei_op_temporal_id가 maxTid보다 크거나 또는 0 내지 num_view_components_op_minus1의 범위의 모든 i에 대한 sei_op_view_id[i]의 리스트가 viewIdTargetList의 서브세트가 아니다 (즉, 0 내지 num_view_components_op_minus1의 범위의 임의의 i에 대한 sei_op_view_id[i]가

viewIdTargetList에서의 값과 동일하다는 것이 참이 아니다).

- [0167] 다음으로, CDN 디바이스 (16) 는, 비트스트림에서의 각각의 SEI NAL 유닛로부터, 각각의 뷰 확장성 정보 SEI 메시지 및 각각의 동작 포인트 비 존재 SEI 메시지를, 존재하는 경우, 제거할 수도 있다 (332). 더욱이, CDN 디바이스 (16) 는 뷰 순서 인덱스 리스트 (VOIdxList) 가 최소 뷰 순서 인덱스 (minVOIdx) 와 동일한 뷰 순서 인덱스를 포함하는지 여부를 결정할 수도 있다 (334). 뷰 순서 인덱스 리스트가 최소 뷰 순서 인덱스와 동일한 뷰 순서 인덱스를 포함한다는 결정에 응답하여 (334의 "예"), CDN 디바이스 (16) 는 서브-비트스트림 추출 프로세스 (298) 를 종료할 수도 있다. 한편, 뷰 순서 인덱스 리스트가 최소 뷰 순서 인덱스와 동일한 뷰 순서 인덱스를 포함하지 않는다는 결정에 응답하여 (334의 "아니오"), CDN 디바이스 (16) 는 최소 뷰 순서 인덱스와 동일한 뷰 순서 인덱스를 갖는 뷰를 추출된 서브-비트스트림의 기본 뷰로 변환할 수도 있다 (336). 뷰를 변환한 후, CDN 디바이스 (16) 는 서브-비트스트림 추출 프로세스 (298) 를 종료할 수도 있다. 비트스트림에 남아있는 데이터는 CDN 디바이스 (16) 가 다른 디바이스, 이를테면 목적지 디바이스 (14) 로 포워딩할 수도 있는 서브-비트스트림이다. 따라서, VOIdxList가 minVOIdx와 동일한 VOIdx의 값을 포함하지 않는 경우, VOIdxList에 포함된 최소 VOIdx 값과 동일한 VOIdx를 갖는 뷰는 추출된 서브-비트스트림의 기본 뷰로 변환된다. 기본 뷰를 생성하는 키 프로세싱 단계들을 약속하는 유익한 프로시저는 MVC-호환 3DV의 하위절 I.8.5.6에서 설명된다.
- [0168] 뷰 순서 인덱스 리스트가 최소 뷰 순서 인덱스와 동일한 뷰 순서 인덱스를 포함하지 않는 경우, 서브-비트스트림 추출 프로세스 (298) 에 의해 생성된 결과적인 서브-비트스트림은 H.264/AVC 표준의 부록 A에서 특정된 하나 이상의 프로파일들에 부합하는 기본 뷰를 포함하지 않을 수도 있다. 따라서, 뷰 순서 인덱스 리스트가 최소 뷰 순서 인덱스와 동일한 뷰 순서 인덱스를 포함하지 않는 경우, 새로운 최소 뷰 순서 인덱스 값을 갖는 남아있는 뷰는 H.264/AVC 표준의 부록 A 및 부록 H에서 특정된 하나 이상의 프로파일들에 부합하는 새로운 기본 뷰가 되도록 액션 336에서 변환될 수도 있다.
- [0169] 도 7은 본 개시물의 하나 이상의 기법들에 따른, 요청된 앵커 텍스처 뷰 성분들의 뷰 식별자들을 결정하는 일 예의 동작 (350) 을 도시하는 흐름도이다. 도 5의 액션 302에 관해 위에서 설명된 바와 같이, CDN 디바이스 (16) 는 "앵커 텍스처를 위해 요청됨"으로서 마킹된 뷰 식별자들을 결정할 수도 있다. CDN 디바이스 (16) 는 현재 뷰 식별자를 특정하는 입력 파라미터로 동작 350을 호출함으로써 "앵커 텍스처를 위해 요청됨"으로서 마킹된 뷰 식별자들의 리스트를 결정할 수도 있다. 도 7의 맥락에서, 본 개시물은 현재 뷰 식별자에 대응하는 뷰 순서 인덱스를 "현재 vOIdx" 또는 간단히 "vOIdx"로서 지칭할 수도 있다. 동작 350은 용어 "뷰 성분"을 "텍스처 뷰 성분"으로 치환하고 "앵커를 위해 요청됨"을 "앵커 텍스처를 위해 요청됨"으로 치환하면, H.264/AVC 표준의 절 H.8.5.1에서 설명된 프로세스와 유사할 수도 있다.
- [0170] 도 7의 예에서 도시된 바와 같이, CDN 디바이스 (16) 는 현재 뷰 식별자가 앵커 텍스처 뷰 성분을 디코딩하기 위해 요구된다고 결정할 수도 있다 (352). 예를 들면, CDN 디바이스 (16) 는 현재 뷰 식별자를 "앵커 텍스처를 위해 요청됨"으로서 마킹할 수도 있다.
- [0171] 덧붙여서, CDN 디바이스 (16) 는 리스트 0에서 현재 뷰 순서 인덱스에 연관된 앵커 뷰간 텍스처 뷰 성분들의 수가 0과 동일할 뿐만 아니라 리스트 1에서 현재 뷰 순서 인덱스에 연관된 앵커 뷰간 텍스처 뷰 성분들의 수도 0과 동일한지 여부를 결정할 수도 있다 (354). 리스트 0에서 현재 뷰 순서 인덱스에 연관된 앵커 뷰간 텍스처 뷰 성분들의 수가 0과 동일하고 리스트 1에서 현재 뷰 순서 인덱스에 연관된 앵커 뷰간 텍스처 뷰 성분들의 수가 0과 동일하다는 결정에 응답하여 (354의 "예"), CDN 디바이스 (16) 는 동작 350을 종료할 수도 있다.
- [0172] 한편, 리스트 0에서 현재 뷰 순서 인덱스에 연관된 앵커 뷰간 텍스처 뷰 성분들의 수가 0과 동일하지 않다는 또는 리스트 1에서 현재 뷰 순서 인덱스에 연관된 앵커 뷰간 텍스처 뷰 성분들의 수가 0과 동일하지 않다는 결정에 응답하여 (354의 "아니오"), CDN 디바이스 (16) 는 리스트 0에서 현재 뷰 순서 인덱스에 연관된 앵커 뷰간 텍스처 뷰 성분들의 수가 0보다 큰지 여부를 결정할 수도 있다 (356).
- [0173] 리스트 0에서 현재 뷰 순서 인덱스에 연관된 앵커 뷰간 텍스처 뷰 성분들의 수가 0보다 크다는 결정에 응답하여 (356의 "예"), CDN 디바이스 (16) 는 현재 뷰 순서 인덱스에 연관되는 리스트 0에서의 각각의 앵커 텍스처 뷰 성분의 디코딩을 위해 요구되는 앵커 텍스처 뷰 성분들의 뷰 식별자들을 결정할 수도 있다 (358). 일부 예들에서, CDN 디바이스 (16) 는 현재 뷰 순서 인덱스에 연관되는 리스트 0에서의 각각의 앵커 텍스처 뷰 성분에 대해 동작 350을 재귀적으로 수행함으로써 그렇게 할 수도 있다. 그런 예들에서, CDN 디바이스 (16) 는 앵커 텍스처 뷰 성분에 대해 동작 350을 재귀적으로 수행하는 경우 앵커 텍스처 뷰 성분의 뷰 식별자를 동작 350에 대한 입력으로서 제공할 수도 있다.

- [0174] 예를 들면, num_anchor_refs_l0[v0Idx]가 0과 동일하지 않은 경우, CDN 디바이스 (16) 는 0 내지 num_anchor_refs_l0[v0Idx] - 1의 범위의 모든 i에 대해, i의 오름 차순으로, anchor_ref_l0[v0Idx][i]에서의 각각의 텍스처 뷰 성분에 대해 동작 350을 호출할 수도 있다. num_anchor_refs_l0[i] 및 anchor_ref_l0[i][j]는 SPS MVC 확장에서의 신택스 엘리먼트들의 세트들이다. num_anchor_refs_l0[i]는 i와 동일한 뷰 순서 인덱스들을 갖는 앵커 뷰 성분들의 디코딩에서 초기 참조 화상 리스트 RefPicList0 (즉, 리스트 0) 에서의 뷰간 예측을 위한 뷰 성분들의 수를 특정한다. 다르게 말하면, num_anchor_refs_l0[i]는 i와 동일한 뷰 순서 인덱스들을 갖는 앵커 뷰 성분들의 뷰간 예측을 위해 처음에 리스트 0에 있는 뷰 성분들이 얼마나 많이 사용되는지를 나타낸다. num_anchor_refs_l0[i]는 i와 동일한 뷰 순서 인덱스를 갖는 앵커 뷰 성분들의 디코딩에서 초기 참조 화상 리스트 RefPicList0 (즉, 리스트 0) 에서의 뷰간 예측을 위해 얼마나 많은 anchor_ref_l0[i][j]가 j-번째 뷰 성분의 뷰 식별자 (view_id) 를 명시하는지를 특정한다.
- [0175] 그렇지 않으면, 리스트 0에서 현재 뷰 순서 인덱스에 연관된 앵커 뷰간 텍스처 뷰 성분들의 수가 0보다 크지 않다는 결정에 응답하여 (356의 "아니오") 또는 액션 358을 수행한 후, CDN 디바이스 (16) 는 리스트 1에서 현재 뷰 순서 인덱스에 연관된 앵커 뷰간 텍스처 뷰 성분들의 수가 0보다 큰지 여부를 결정할 수도 있다 (360). 리스트 1에서 현재 뷰 순서 인덱스에 연관된 앵커 뷰간 텍스처 뷰 성분들의 수가 0보다 크지 않다는 결정에 응답하여 (360의 "아니오"), CDN 디바이스 (16) 는 동작 350을 종료할 수도 있다.
- [0176] 그러나, 리스트 1에서 현재 뷰 순서 인덱스에 연관된 앵커 뷰간 텍스처 뷰 성분들의 수가 0보다 크다는 결정에 응답하여 (360의 "예"), CDN 디바이스 (16) 는 현재 뷰 순서 인덱스에 연관되는 리스트 1에서의 각각의 앵커 텍스처 뷰 성분의 디코딩을 위해 요구되는 앵커 텍스처 뷰 성분들의 뷰 식별자들을 결정할 수도 있다 (362). 일부 예들에서, CDN 디바이스 (16) 는 현재 뷰 순서 인덱스에 연관되는 리스트 1에서의 각각의 앵커 텍스처 뷰 성분에 대해 동작 350을 재귀적으로 수행함으로써 그렇게 할 수도 있다. 그런 예들에서, CDN 디바이스 (16) 는 앵커 텍스처 뷰 성분에 대해 동작 350을 재귀적으로 수행하는 경우 앵커 텍스처 뷰 성분의 뷰 식별자를 동작 350에 대한 입력으로서 제공할 수도 있다.
- [0177] 예를 들면, num_anchor_refs_l1[v0Idx]가 0과 동일하지 않은 경우, CDN 디바이스 (16) 는 0 내지 num_anchor_refs_l1[v0Idx] - 1의 범위의 모든 i에 대해, i의 오름 차순으로, anchor_ref_l1[v0Idx][i]에서의 각각의 텍스처 뷰 성분에 대해 동작 448을 호출할 수도 있다. num_anchor_refs_l1[i] 및 anchor_ref_l1[i][j]는 SPS MVC 확장에서의 신택스 엘리먼트들의 세트들이다. num_anchor_refs_l1[i]는 i와 동일한 뷰 순서 인덱스들을 갖는 앵커 뷰 성분들의 디코딩에서 초기 참조 화상 리스트 RefPicList1 (즉, 리스트 1) 에서의 뷰간 예측을 위한 뷰 성분들의 수를 특정한다. anchor_ref_l1[i][j]는 i와 동일한 뷰 순서 인덱스를 갖는 앵커 뷰 성분들의 디코딩에서 초기 참조 화상 리스트 RefPicList1 (즉, 리스트 1) 에서의 뷰간 예측을 위한 j-번째 뷰 성분의 뷰 식별자 (view_id) 를 특정한다. 액션 362를 수행한 후, CDN 디바이스 (16) 는 동작 350을 종료할 수도 있다.
- [0178] 도 8은 본 개시물의 하나 이상의 기법들에 따른, 요청된 앵커 깊이 뷰 성분들의 뷰 식별자들을 결정하는 일 예의 동작 (400) 을 도시하는 흐름도이다. 도 5의 액션 306에 관해 위에서 설명된 바와 같이, CDN 디바이스 (16) 는 "앵커 깊이를 위해 요청됨"으로서 마킹된 뷰 식별자들을 결정할 수도 있다. CDN 디바이스 (16) 는 현재 뷰 식별자를 특정하는 입력 파라미터로 동작 400을 호출함으로써 "앵커 깊이를 위해 요청됨"으로서 마킹된 뷰 식별자들의 리스트를 결정할 수도 있다. 도 8의 맥락에서, 본 개시물은 현재 뷰 식별자에 대응하는 뷰 순서 인덱스를 "현재 v0Idx" 또는 간단히 "v0Idx"로서 지칭할 수도 있다. 동작 400은 용어 "뷰 성분"을 "깊이 뷰 성분"으로 치환하고 "앵커를 위해 요청됨"을 "앵커 깊이를 위해 요청됨"으로 치환하면, H.264/AVC 표준의 절 H.8.5.1에서 설명된 프로세스와 유사할 수도 있다.
- [0179] 도 8의 예에서 도시된 바와 같이, CDN 디바이스 (16) 는 현재 뷰 식별자가 앵커 깊이 뷰 성분을 디코딩하기 위해 요구된다고 결정할 수도 있다 (402). 예를 들면, CDN 디바이스 (16) 는 현재 뷰 식별자를 "앵커 깊이를 위해 요청됨"으로서 마킹할 수도 있다.
- [0180] 덧붙여서, CDN 디바이스 (16) 는 리스트 0에서 현재 뷰 순서 인덱스에 연관된 앵커 뷰간 깊이 뷰 성분들의 수가 0과 동일할 뿐만 아니라 리스트 1에서 현재 뷰 순서 인덱스에 연관된 앵커 뷰간 깊이 뷰 성분들의 수도 0과 동일한지 여부를 결정할 수도 있다 (404). 리스트 0에서 현재 뷰 순서 인덱스에 연관된 앵커 뷰간 깊이 뷰 성분들의 수가 0과 동일하고 리스트 1에서 현재 뷰 순서 인덱스에 연관된 앵커 뷰간 깊이 뷰 성분들의 수가 0과 동일하다는 결정에 응답하여 (404의 "예"), CDN 디바이스 (16) 는 동작 400을 종료할 수도 있다.
- [0181] 한편, 리스트 0에서 현재 뷰 순서 인덱스에 연관된 앵커 뷰간 깊이 뷰 성분들의 수가 0과 동일하지 않다는 또는

리스트 1에서 현재 뷰 순서 인덱스에 연관된 앵커 뷰간 깊이 뷰 성분들의 수가 0과 동일하지 않다는 결정에 응답하여 (404의 "아니오"), CDN 디바이스 (16) 는 리스트 0에서 현재 뷰 순서 인덱스에 연관된 앵커 뷰간 깊이 뷰 성분들의 수가 0보다 큰지 여부를 결정할 수도 있다 (406).

[0182] 리스트 0에서 현재 뷰 순서 인덱스에 연관된 앵커 뷰간 깊이 뷰 성분들의 수가 0보다 크다는 결정에 응답하여 (406의 "예"), CDN 디바이스 (16) 는 현재 뷰 순서 인덱스에 연관되는 리스트 0에서의 각각의 앵커 깊이 뷰 성분의 디코딩을 위해 요구되는 앵커 깊이 뷰 성분들의 뷰 식별자들을 결정할 수도 있다 (408). 일부 예들에서, CDN 디바이스 (16) 는 현재 뷰 순서 인덱스에 연관되는 리스트 0에서의 각각의 앵커 깊이 뷰 성분에 대해 동작 400을 재귀적으로 수행함으로써 그렇게 할 수도 있다. 그런 예들에서, CDN 디바이스 (16) 는 앵커 깊이 뷰 성분에 대해 동작 400을 재귀적으로 수행하는 경우 앵커 깊이 뷰 성분의 뷰 식별자를 동작 400에 대한 입력으로서 제공할 수도 있다.

[0183] 예를 들면, $\text{num_anchor_refs_l0}[\text{vIdx}]$ 가 0과 동일하지 않은 경우, CDN 디바이스 (16) 는 0 내지 $\text{num_anchor_l0}[\text{vIdx}] - 1$ 의 범위의 모든 i 에 대해, i 의 오름 차순으로, $\text{anchor_ref_l0}[\text{vIdx}][i]$ 에서의 각각의 깊이 뷰 성분에 대해 동작 400을 호출할 수도 있다. $\text{num_anchor_refs_l0}[i]$ 및 $\text{anchor_ref_l0}[i][j]$ 는 SPS MVC 확장에서의 선택스 엘리먼트들의 세트들이다. $\text{num_anchor_refs_l0}[i]$ 는 i 와 동일한 뷰 순서 인덱스들을 갖는 앵커 뷰 성분들의 디코딩에서 초기 참조 화상 리스트 RefPicList0 (즉, 리스트 0) 에서의 뷰간 예측을 위한 뷰 성분들의 수를 특징한다. $\text{anchor_ref_l0}[i][j]$ 는 i 와 동일한 뷰 순서 인덱스를 갖는 앵커 뷰 성분들의 디코딩에서 초기 참조 화상 리스트 RefPicList0 (즉, 리스트 0) 에서의 뷰간 예측을 위한 j -번째 뷰 성분의 뷰 식별자 (view_id) 를 특징한다.

[0184] 그렇지 않으면, 리스트 0에서 현재 뷰 순서 인덱스에 연관된 앵커 뷰간 깊이 뷰 성분들의 수가 0보다 크지 않다는 결정에 응답하여 (406의 "아니오") 또는 액션 408을 수행한 후, CDN 디바이스 (16) 는 리스트 1에서 현재 뷰 순서 인덱스에 연관된 앵커 뷰간 깊이 뷰 성분들의 수가 0보다 큰지 여부를 결정할 수도 있다 (410). 리스트 1에서 현재 뷰 순서 인덱스에 연관된 앵커 뷰간 깊이 뷰 성분들의 수가 0보다 크지 않다는 결정에 응답하여 (410의 "아니오"), CDN 디바이스 (16) 는 동작 400을 종료할 수도 있다.

[0185] 그러나, 리스트 1에서 현재 뷰 순서 인덱스에 연관된 앵커 뷰간 깊이 뷰 성분들의 수가 0보다 크다는 결정에 응답하여 (410의 "예"), CDN 디바이스 (16) 는 현재 뷰 순서 인덱스에 연관되는 리스트 1에서의 각각의 앵커 깊이 뷰 성분의 디코딩을 위해 요구되는 앵커 텍스처 뷰 성분들의 뷰 식별자들을 결정할 수도 있다 (412). 일부 예들에서, CDN 디바이스 (16) 는 현재 뷰 순서 인덱스에 연관되는 리스트 1에서의 각각의 앵커 깊이 뷰 성분에 대해 동작 400을 재귀적으로 수행함으로써 그렇게 할 수도 있다. 그런 예들에서, CDN 디바이스 (16) 는 앵커 깊이 뷰 성분에 대해 동작 400을 재귀적으로 수행하는 경우 앵커 깊이 뷰 성분의 뷰 식별자를 동작 400에 대한 입력으로서 제공할 수도 있다.

[0186] 예를 들면, $\text{num_anchor_refs_l1}[\text{vIdx}]$ 가 0과 동일하지 않은 경우, CDN 디바이스 (16) 는 0 내지 $\text{num_anchor_l1}[\text{vIdx}] - 1$ 의 범위의 모든 i 에 대해, i 의 오름 차순으로, $\text{anchor_ref_l1}[\text{vIdx}][i]$ 에서의 각각의 깊이 뷰 성분에 대해 동작 400을 호출할 수도 있다. $\text{num_anchor_refs_l1}[i]$ 및 $\text{anchor_ref_l1}[i][j]$ 는 SPS MVC 확장에서의 선택스 엘리먼트들의 세트들이다. $\text{num_anchor_refs_l1}[i]$ 는 i 와 동일한 뷰 순서 인덱스들을 갖는 앵커 뷰 성분들의 디코딩에서 초기 참조 화상 리스트 RefPicList1 (즉, 리스트 1) 에서의 뷰간 예측을 위한 뷰 성분들의 수를 특징한다. $\text{anchor_ref_l1}[i][j]$ 는 i 와 동일한 뷰 순서 인덱스를 갖는 앵커 뷰 성분들의 디코딩에서 초기 참조 화상 리스트 RefPicList1 (즉, 리스트 1) 에서의 뷰간 예측을 위한 j -번째 뷰 성분의 뷰 식별자 (view_id) 를 특징한다. 액션 412를 수행한 후, CDN 디바이스 (16) 는 동작 400을 종료할 수도 있다.

[0187] 도 9는 본 개시물의 하나 이상의 기법들에 따른, 요청된 비 앵커 텍스처 뷰 성분들의 뷰 식별자들을 결정하는 일 예의 동작 (450) 을 도시하는 흐름도이다. 도 5의 액션 308에 관해 위에서 설명된 바와 같이, CDN 디바이스 (16) 는 "비 앵커 텍스처를 위해 요청됨"으로서 마킹된 뷰 식별자들을 결정할 수도 있다. CDN 디바이스 (16) 는 현재 뷰 식별자를 특징하는 입력 파라미터로 동작 450을 호출함으로써 "비 앵커 텍스처를 위해 요청됨"으로서 마킹된 뷰 식별자들의 리스트를 결정할 수도 있다. 도 9의 맥락에서, 본 개시물은 현재 뷰 식별자에 대응하는 뷰 순서 인덱스를 "현재 vIdx" 또는 간단히 "vIdx"로서 지칭할 수도 있다. 동작 450은 용어 "뷰 성분"을 "텍스처 뷰 성분"으로 치환하고 "비 앵커를 위해 요청됨"을 "비 앵커 텍스처를 위해 요청됨"으로 치환하면, H.264/AVC 표준의 절 H.8.5.2에서 설명된 프로세스와 유사할 수도 있다.

[0188] 도 9의 예에서 도시된 바와 같이, CDN 디바이스 (16) 는 현재 뷰 식별자가 비 앵커 텍스처 뷰 성분을 디코딩하

기 위해 요구된다고 결정할 수도 있다 (452). 예를 들면, CDN 디바이스 (16) 는 현재 뷰 식별자를 "비 앵커 텍스처를 위해 요청됨"으로서 마킹할 수도 있다.

[0189] 덧붙여서, CDN 디바이스 (16) 는 리스트 0에서 현재 뷰 순서 인덱스에 연관된 비 앵커 뷰간 텍스처 뷰 성분들의 수가 0과 동일할 뿐만 아니라 리스트 1에서 현재 뷰 순서 인덱스에 연관된 비 앵커 뷰간 텍스처 뷰 성분들의 수도 0과 동일하지 여부를 결정할 수도 있다 (454). 리스트 0에서 현재 뷰 순서 인덱스에 연관된 비 앵커 뷰간 텍스처 뷰 성분들의 수가 0과 동일하고 리스트 1에서 현재 뷰 순서 인덱스에 연관된 비 앵커 뷰간 텍스처 뷰 성분들의 수가 0과 동일하다는 결정에 응답하여 (454의 "예"), CDN 디바이스 (16) 은 동작 450을 종료할 수도 있다.

[0190] 한편, 리스트 0에서 현재 뷰 순서 인덱스에 연관된 비 앵커 뷰간 텍스처 뷰 성분들의 수가 0과 동일하지 않다는 또는 리스트 1에서 현재 뷰 순서 인덱스에 연관된 비 앵커 뷰간 텍스처 뷰 성분들의 수가 0과 동일하지 않다는 결정에 응답하여 (454의 "아니오"), CDN 디바이스 (16) 는 리스트 0에서 현재 뷰 순서 인덱스에 연관된 비 앵커 뷰간 텍스처 뷰 성분들의 수가 0보다 큰지 여부를 결정할 수도 있다 (456).

[0191] 리스트 0에서 현재 뷰 순서 인덱스에 연관된 비 앵커 뷰간 텍스처 뷰 성분들의 수가 0보다 크다는 결정에 응답하여 (456의 "예"), CDN 디바이스 (16) 는 현재 뷰 순서 인덱스에 연관되는 리스트 0에서의 각각의 비 앵커 텍스처 뷰 성분의 디코딩을 위해 요구되는 비 앵커 텍스처 뷰 성분들의 뷰 식별자들을 결정할 수도 있다 (458). 일부 예들에서, CDN 디바이스 (16) 는 현재 뷰 순서 인덱스에 연관되는 리스트 0에서의 각각의 비 앵커 텍스처 뷰 성분에 대해 동작 450을 재귀적으로 수행함으로써 그렇게 할 수도 있다. 그런 예들에서, CDN 디바이스 (16) 는 비 앵커 텍스처 뷰 성분에 대해 동작 450을 재귀적으로 수행하는 경우 비 앵커 텍스처 뷰 성분의 뷰 식별자를 동작 450에 대한 입력으로서 제공할 수도 있다.

[0192] 예를 들면, num_non_anchor_refs_l0[v0Idx]가 0과 동일하지 않은 경우, CDN 디바이스 (16) 는 0 내지 num_non_anchor_refs_l0[v0Idx] - 1의 범위의 모든 i에 대해, i의 오름 차순으로, non_anchor_ref_l0[v0Idx][i]에서의 각각의 텍스처 뷰 성분에 대해 동작 450을 호출할 수도 있다. num_non_anchor_refs_l0[i] 및 non_anchor_ref_l0[i][j]는 SPS MVC 확장에서의 선택스 엘리먼트들의 세트들이다. num_non_anchor_refs_l0[i]는 i와 동일한 뷰 순서 인덱스들을 갖는 비 앵커 뷰 성분들의 디코딩에서 초기 참조 화상 리스트 RefPicList0 (즉, 리스트 0) 에서의 뷰간 예측을 위한 뷰 성분들의 수를 특정한다. non_anchor_ref_l0[i][j]는 i와 동일한 뷰 순서 인덱스를 갖는 비 앵커 뷰 성분들의 디코딩에서 초기 참조 화상 리스트 RefPicList0 (즉, 리스트 0) 에서의 뷰간 예측을 위한 j-번째 뷰 성분의 뷰 식별자 (view_id) 를 특정한다.

[0193] 그렇지 않으면, 리스트 0에서 현재 뷰 순서 인덱스에 연관된 비 앵커 뷰간 텍스처 뷰 성분들의 수가 0보다 크지 않다는 결정에 응답하여 (456의 "아니오") 또는 액션 458을 수행한 후, CDN 디바이스 (16) 는 리스트 1에서 현재 뷰 순서 인덱스에 연관된 비 앵커 뷰간 텍스처 뷰 성분들의 수가 0보다 큰지 여부를 결정할 수도 있다 (460). 리스트 1에서 현재 뷰 순서 인덱스에 연관된 비 앵커 뷰간 텍스처 뷰 성분들의 수가 0보다 크지 않다는 결정에 응답하여 (460의 "아니오"), CDN 디바이스 (16) 는 동작 450을 종료할 수도 있다.

[0194] 그러나, 리스트 1에서 현재 뷰 순서 인덱스에 연관된 비 앵커 뷰간 텍스처 뷰 성분들의 수가 0보다 크다는 결정에 응답하여 (460의 "예"), CDN 디바이스 (16) 는 현재 뷰 순서 인덱스에 연관되는 리스트 1에서의 각각의 비 앵커 텍스처 뷰 성분의 디코딩을 위해 요구되는 비 앵커 텍스처 뷰 성분들의 뷰 식별자들을 결정할 수도 있다 (462). 일부 예들에서, CDN 디바이스 (16) 는 현재 뷰 순서 인덱스에 연관되는 리스트 1에서의 각각의 비 앵커 텍스처 뷰 성분에 대해 동작 450을 재귀적으로 수행함으로써 그렇게 할 수도 있다. 그런 예들에서, CDN 디바이스 (16) 는 비 앵커 텍스처 뷰 성분에 대해 동작 450을 재귀적으로 수행하는 경우 비 앵커 텍스처 뷰 성분의 뷰 식별자를 동작 450에 대한 입력으로서 제공할 수도 있다.

[0195] 예를 들면, num_non_anchor_refs_l1[v0Idx]가 0과 동일하지 않은 경우, CDN 디바이스 (16) 는 0 내지 num_non_anchor_refs_l1[v0Idx] - 1의 범위의 모든 i에 대해, i의 오름 차순으로, non_anchor_ref_l1[v0Idx][i]에서의 각각의 텍스처 뷰 성분에 대해 동작 450을 호출할 수도 있다. num_non_anchor_refs_l1[i] 및 non_anchor_ref_l1[i][j]는 SPS MVC 확장에서의 선택스 엘리먼트들의 세트들이다. num_non_anchor_refs_l1[i]는 i와 동일한 뷰 순서 인덱스들을 갖는 비 앵커 뷰 성분들의 디코딩에서 초기 참조 화상 리스트 RefPicList1 (즉, 리스트 1) 에서의 뷰간 예측을 위한 뷰 성분들의 수를 특정한다. non_anchor_ref_l1[i][j]는 i와 동일한 뷰 순서 인덱스를 갖는 비 앵커 뷰 성분들의 디코딩에서 초기 참조 화상 리스트 RefPicList1 (즉, 리스트 1) 에서의 뷰간 예측을 위한 j-번째 뷰 성분의 뷰 식별자 (view_id) 를 특정한다.

다. 액션 462를 수행한 후, CDN 디바이스 (16) 는 동작 450을 종료할 수도 있다.

[0196] 도 10은 본 개시물의 하나 이상의 기법들에 따른, 요청된 비 앵커 깊이 뷰 성분들의 뷰 식별자들을 결정하는 일예의 동작 (500) 을 도시하는 흐름도이다. 도 5의 액션 312에 관해 위에서 설명된 바와 같이, CDN 디바이스 (16) 는 "비 앵커 깊이를 위해 요청됨"으로서 마킹된 뷰 식별자들을 결정할 수도 있다. CDN 디바이스 (16) 는 현재 뷰 식별자를 특징하는 입력 파라미터로 동작 500을 호출함으로써 "비 앵커 깊이를 위해 요청됨"으로서 마킹된 뷰 식별자들의 리스트를 결정할 수도 있다. 도 10의 맥락에서, 본 개시물은 현재 뷰 식별자에 대응하는 뷰 순서 인덱스를 "현재 vOIdx" 또는 간단히 "vOIdx"로서 지칭할 수도 있다. 동작 500은 용어 "뷰 성분"을 "깊이 뷰 성분"으로 치환하고 "비 앵커를 위해 요청됨"을 "비 앵커 깊이를 위해 요청됨"으로 치환하면, H.264/AVC 표준의 절 H.8.5.2에서 설명된 프로세스와 유사할 수도 있다.

[0197] 도 10의 예에서 도시된 바와 같이, CDN 디바이스 (16) 는 현재 뷰 식별자가 비 앵커 깊이 뷰 성분을 디코딩하기 위해 요구된다고 결정할 수도 있다 (502). 예를 들면, CDN 디바이스 (16) 는 현재 뷰 식별자를 "비 앵커 깊이를 위해 요청됨"으로서 마킹할 수도 있다.

[0198] 덧붙여서, CDN 디바이스 (16) 는 리스트 0에서 현재 뷰 순서 인덱스에 연관된 비 앵커 뷰간 깊이 뷰 성분들의 수가 0과 동일할 뿐만 아니라 리스트 1에서 현재 뷰 순서 인덱스에 연관된 비 앵커 뷰간 깊이 뷰 성분들의 수도 0과 동일한지 여부를 결정할 수도 있다 (504). 리스트 0에서 현재 뷰 순서 인덱스에 연관된 비 앵커 뷰간 깊이 뷰 성분들의 수가 0과 동일하고 리스트 1에서 현재 뷰 순서 인덱스에 연관된 비 앵커 뷰간 깊이 뷰 성분들의 수가 0과 동일하다는 결정에 응답하여 (504의 "예"), CDN 디바이스 (16) 은 동작 500을 종료할 수도 있다.

[0199] 한편, 리스트 0에서 현재 뷰 순서 인덱스에 연관된 비 앵커 뷰간 깊이 뷰 성분들의 수가 0과 동일하지 않다는 또는 리스트 1에서 현재 뷰 순서 인덱스에 연관된 비 앵커 뷰간 깊이 뷰 성분들의 수가 0과 동일하지 않다는 결정에 응답하여 (504의 "아니오"), CDN 디바이스 (16) 는 리스트 0에서 현재 뷰 순서 인덱스에 연관된 비 앵커 뷰간 깊이 뷰 성분들의 수가 0보다 큰지 여부를 결정할 수도 있다 (506).

[0200] 리스트 0에서 현재 뷰 순서 인덱스에 연관된 비 앵커 뷰간 깊이 뷰 성분들의 수가 0보다 크다는 결정에 응답하여 (506의 "예"), CDN 디바이스 (16) 는 현재 뷰 순서 인덱스에 연관되는 리스트 0에서의 각각의 비 앵커 깊이 뷰 성분의 디코딩을 위해 요구되는 비 앵커 깊이 뷰 성분들의 뷰 식별자들을 결정할 수도 있다 (508). 일부 예들에서, CDN 디바이스 (16) 는 현재 뷰 순서 인덱스에 연관되는 리스트 0에서의 각각의 비 앵커 깊이 뷰 성분 에 대해 동작 500을 재귀적으로 수행함으로써 그렇게 할 수도 있다. 그런 예들에서, CDN 디바이스 (16) 는 비 앵커 깊이 뷰 성분에 대해 동작 500을 재귀적으로 수행하는 경우 비 앵커 깊이 뷰 성분의 뷰 식별자를 동작 500에 대한 입력으로서 제공할 수도 있다.

[0201] 예를 들면, num_non_anchor_refs_10[vOIdx]가 0과 동일하지 않은 경우, CDN 디바이스 (16) 는 0 내지 num_non_anchor_10[vOIdx] - 1의 범위의 모든 i에 대해, i의 오름 차순으로, non_anchor_ref_10[vOIdx][i]에서의 각각의 깊이 뷰 성분에 대해 동작 500을 호출할 수도 있다. num_non_anchor_refs_10[i] 및 non_anchor_ref_10[i][j]는 SPS MVC 확장에서의 선택스 엘리먼트들의 세트들이다. num_non_anchor_refs_10[i]는 i와 동일한 뷰 순서 인덱스들을 갖는 비 앵커 뷰 성분들의 디코딩에서 초기 참조 화상 리스트 RefPicList0 (즉, 리스트 0) 에서의 뷰간 예측을 위한 뷰 성분들의 수를 특징한다. non_anchor_ref_10[i][j]는 i와 동일한 뷰 순서 인덱스를 갖는 비 앵커 뷰 성분들의 디코딩에서 초기 참조 화상 리스트 RefPicList0 (즉, 리스트 0) 에서의 뷰간 예측을 위한 j-번째 뷰 성분의 뷰 식별자 (view_id) 를 특징한다.

[0202] 그렇지 않으면, 리스트 0에서 현재 뷰 순서 인덱스에 연관된 비 앵커 뷰간 깊이 뷰 성분들의 수가 0보다 크지 않다는 결정에 응답하여 (506의 "아니오") 또는 액션 508을 수행한 후, CDN 디바이스 (16) 는 리스트 1에서 현재 뷰 순서 인덱스에 연관된 비 앵커 뷰간 깊이 뷰 성분들의 수가 0보다 큰지 여부를 결정할 수도 있다 (510). 리스트 1에서 현재 뷰 순서 인덱스에 연관된 비 앵커 뷰간 깊이 뷰 성분들의 수가 0보다 크지 않다는 결정에 응답하여 (510의 "아니오"), CDN 디바이스 (16) 는 동작 500을 종료할 수도 있다.

[0203] 그러나, 리스트 1에서 현재 뷰 순서 인덱스에 연관된 비 앵커 뷰간 깊이 뷰 성분들의 수가 0보다 크다는 결정에 응답하여 (510의 "예"), CDN 디바이스 (16) 는 현재 뷰 순서 인덱스에 연관되는 리스트 1에서의 각각의 비 앵커 깊이 뷰 성분의 디코딩을 위해 요구되는 비 앵커 텍스처 뷰 성분들의 뷰 식별자들을 결정할 수도 있다 (512). 일부 예들에서, CDN 디바이스 (16) 는 현재 뷰 순서 인덱스에 연관되는 리스트 1에서의 각각의 비 앵커 깊이 뷰 성분에 대해 동작 500을 재귀적으로 수행함으로써 그렇게 할 수도 있다. 그런 예들에서, CDN 디바이스

(16) 는 비 앵커 깊이 뷰 성분에 대해 동작 500을 재귀적으로 수행하는 경우 비 앵커 깊이 뷰 성분의 뷰 식별자를 동작 500에 대한 입력으로서 제공할 수도 있다.

[0204] 예를 들면, num_non_anchor_refs_l1[v0Idx]가 0과 동일하지 않은 경우, CDN 디바이스 (16) 는 0 내지 num_non_anchor_l1[v0Idx] - 1의 범위의 모든 i에 대해, i의 오름 차순으로, non_anchor_ref_l1[v0Idx][i]에서의 각각의 깊이 뷰 성분에 대해 동작 500을 호출할 수도 있다. num_non_anchor_refs_l1[i] 및 non_anchor_ref_l1[i][j]는 SPS MVC 확장에서의 선택스 엘리먼트들의 세트들이다. num_non_anchor_refs_l1[i]는 i와 동일한 뷰 순서 인덱스들을 갖는 비 앵커 뷰 성분들의 디코딩에서 초기 참조 화상 리스트 RefPicList1 (즉, 리스트 1) 에서의 뷰간 예측을 위한 뷰 성분들의 수를 특정한다. non_anchor_ref_l1[i][j]는 i와 동일한 뷰 순서 인덱스를 갖는 비 앵커 뷰 성분들의 디코딩에서 초기 참조 화상 리스트 RefPicList1 (즉, 리스트 1) 에서의 뷰간 예측을 위한 j-번째 뷰 성분의 뷰 식별자 (view_id) 를 특정한다. 액션 512를 수행한 후, CDN 디바이스 (16) 는 동작 500을 종료할 수도 있다.

[0205] 도 11은 본 개시물의 하나 이상의 기법들에 따른, VCL NAL 유닛들 및 필터 데이터 NAL 유닛들을 비트스트림으로부터 제거될 것으로서 마킹하는 제 1 예 동작 (550) 을 도시하는 흐름도이다. CDN 디바이스 (16) 는 도 5의 예에서 적어도 액션 316을 수행하는 부분으로서 도 11의 동작을 수행할 수도 있다. 도 11의 다음의 설명은 VCL NAL 유닛들 및 필터 데이터 NAL 유닛들에 적용가능할 수도 있다.

[0206] 도 11의 예에서 도시된 바와 같이, CDN 디바이스 (16) 는 NAL 유닛의 우선순위 식별자 (priority_id) 가 pIdTarget보다 큰지 여부를 결정할 수도 있다 (552). 위에서 논의된 바와 같이, NAL 유닛의 MVC NAL 유닛 헤더 확장은 priority_id 선택스 엘리먼트를 포함할 수도 있다. pIdTarget은 서브-비트스트림 추출 프로세스, 이를테면 도 5의 서브-비트스트림 추출 프로세스 (298) 에 제공되는 파라미터이다. pIdTarget은 타겟 우선순위 레벨을 식별할 수도 있다. NAL 유닛의 우선순위 식별자가 pIdTarget보다 크다는 결정에 응답하여 (552의 "예"), CDN 디바이스 (16) 는 그 NAL 유닛을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다 (554). 이런 식으로, CDN 디바이스 (16) 는 NAL 유닛들의 priority_id 선택스 엘리먼트들이 pIdTarget보다 큰 경우 모든 VCL NAL 유닛들 및 필터 데이터 NAL 유닛들을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다.

[0207] 한편, NAL 유닛의 우선순위 식별자가 pIdTarget보다 크지 않다는 결정에 응답하여 (552의 "아니오"), CDN 디바이스 (16) 는 NAL 유닛의 시간적 식별자 (temporal_id) 가 tIdTarget보다 큰지 여부를 결정할 수도 있다 (556). 위에서 논의된 바와 같이, NAL 유닛의 MVC NAL 유닛 헤더 확장은 temporal_id 선택스 엘리먼트를 포함할 수도 있고, tIdTarget은 서브-비트스트림 추출 프로세스, 이를테면 서브-비트스트림 추출 프로세스 (298) 에 제공되는 파라미터이다. tIdTarget은 타겟 시간적 레벨을 식별할 수도 있다. NAL 유닛의 시간적 식별자가 tIdTarget보다 크다는 결정에 응답하여 (556의 "예"), CDN 디바이스 (16) 는 그 NAL 유닛을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다 (554). 이런 식으로, CDN 디바이스 (16) 는 NAL 유닛들의 temporal_id 선택스 엘리먼트들이 tIdTarget보다 큰 경우 모든 VCL NAL 유닛들 및 필터 데이터 NAL 유닛들을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다.

[0208] NAL 유닛의 시간적 식별자가 tIdTarget보다 크지 않다는 결정에 응답하여 (556의 "아니오"), CDN 디바이스 (16) 는, 본 개시물의 기법들에 따라, NAL 유닛이 앵커 화상에 속하며, NAL 유닛이 깊이 뷰 성분 NAL 유닛 유형 이외의 NAL 유닛 유형에 속하고, NAL 유닛의 뷰 식별자 (view_id) 가 "앵커 텍스처를 위해 요청됨"으로서 마킹되지 않는지 여부를 결정할 수도 있다 (558). CDN 디바이스 (16) 는, NAL 유닛의 앵커 화상 플래그 (anchor_pic_flag) 에 기초하여, NAL 유닛이 앵커 화상에 속하는지 여부를 결정할 수도 있다. 깊이 뷰 성분 NAL 유닛 유형에 속하는 NAL 유닛들은 깊이 뷰 성분들에 대한 코딩된 슬라이스 확장들을 포함할 수도 있다. 일부 예들에서, 21과 동일한 nal_unit_type을 갖는 NAL 유닛들은 깊이 뷰 성분 NAL 유닛 유형에 속한다.

[0209] NAL 유닛이 앵커 화상에 속하며, NAL 유닛이 깊이 뷰 성분 NAL 유닛 유형 이외의 NAL 유닛 유형에 속하고, NAL 유닛의 뷰 식별자가 "앵커 텍스처를 위해 요청됨"으로서 마킹되지 않는다는 결정에 응답하여 (558의 "예"), CDN 디바이스 (16) 는 그 NAL 유닛을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다 (554). 이런 식으로, CDN 디바이스 (16) 는, 만약 NAL 유닛의 anchor_pic_flag가 1과 동일하며, NAL 유닛의 nal_unit_type이 21이 아니고, NAL 유닛의 view_id가 "앵커 텍스처를 위해 요청됨"으로서 마킹되지 않으면, VCL NAL 유닛 또는 필터 데이터 NAL 유닛을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다.

[0210] "앵커 텍스처를 위해 요청됨"으로서 마킹된 뷰 식별자들은 앵커 텍스처 타겟 뷰 리스트 내에 있다. 따라서, CDN 디바이스 (16) 는, NAL 유닛이 앵커 화상에 속하는지 여부, NAL 유닛이 깊이 뷰 성분 NAL 유닛 유형에 속하

는지 여부, 및 앵커 텍스처 타겟 뷰 리스트가 NAL 유닛의 뷰 식별자를 명시하는지 여부에 적어도 부분적으로 기초하여, 3DV 비트스트림으로부터 NAL 유닛을 제거할지 여부를 결정할 수도 있고, 여기서 깊이 뷰 성분 NAL 유닛 유형에 속한 NAL 유닛들은 깊이 뷰 성분들에 대한 코딩된 슬라이스 확장들을 캡슐화한다. 예를 들어, CDN 디바이스 (16) 는, NAL 유닛의 헤더에서의 앵커 화상 플래그 신텍스 엘리먼트가 1과 동일하며, NAL 유닛의 헤더에서의 NAL 유닛 유형 신텍스 엘리먼트가 21과 동일하지 않고, NAL 유닛의 헤더에서의 뷰 식별자 신텍스 엘리먼트가 앵커 텍스처 타겟 리스트에서의 뷰의 뷰 식별자와 동일한 경우, 그 NAL 유닛이 3DV 비트스트림으로부터 제거될 것이라고 결정할 수도 있다.

[0211] 그러나, NAL 유닛이 앵커 화상에 속하거나, NAL 유닛이 깊이 뷰 성분 NAL 유닛 유형에 속하거나, 또는 NAL 유닛의 뷰 식별자가 "앵커 텍스처를 위해 요청됨"으로서 마킹된다는 결정에 응답하여 (558의 "아니오"), CDN 디바이스 (16) 는, 본 개시물의 기법들에 따라, NAL 유닛이 비 앵커 화상에 속하며, NAL 유닛 유형이 깊이 뷰 성분 NAL 유닛 유형 이외의 NAL 유닛 유형에 속하고, NAL 유닛의 뷰 식별자 (view_id) 가 "비 앵커 텍스처를 위해 요청됨"으로서 마킹되지 않는지 여부를 결정할 수도 있다 (560).

[0212] NAL 유닛이 비 앵커 화상에 속하며, NAL 유닛이 깊이 뷰 성분 NAL 유닛 유형 이외의 NAL 유닛 유형에 속하고, NAL 유닛의 뷰 식별자가 "비 앵커 텍스처를 위해 요청됨"으로서 마킹되지 않는다는 결정에 응답하여 (560의 "예"), CDN 디바이스 (16) 는 그 NAL 유닛을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다 (554).
이런 식으로, CDN 디바이스 (16) 는, 만약 NAL 유닛의 anchor_pic_flag가 0과 동일하며, NAL 유닛의 nal_unit_type이 21이 아니고, NAL 유닛의 view_id가 "비 앵커 텍스처를 위해 요청됨"으로서 마킹되지 않으면, VCL NAL 유닛 또는 필러 데이터 NAL 유닛을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다.

[0213] "비 앵커 텍스처를 위해 요청됨"으로서 마킹된 뷰 식별자들은 비 앵커 텍스처 뷰 타겟 리스트 내에 있다. 따라서, CDN 디바이스 (16) 는 NAL 유닛이 앵커 화상에 속하는지 여부, NAL 유닛이 상기 깊이 뷰 성분 NAL 유닛 유형에 속하는지 여부, 및 비 앵커 텍스처 타겟 뷰 리스트가 NAL 유닛의 뷰 식별자를 명시하는지 여부에 적어도 부분적으로 기초하여, 3DV 비트스트림으로부터 NAL 유닛을 제거할지 여부를 결정할 수도 있다. 예를 들어, CDN 디바이스 (16) 는, NAL 유닛의 헤더에서의 앵커 화상 플래그 신텍스 엘리먼트가 0과 동일하며, NAL 유닛의 헤더에서의 NAL 유닛 유형 신텍스 엘리먼트가 21과 동일하지 않고, NAL 유닛의 헤더에서의 뷰 식별자 신텍스 엘리먼트가 비 앵커 텍스처 타겟 리스트에서의 뷰의 뷰 식별자와 동일한 경우, 그 NAL 유닛이 3DV 비트스트림으로부터 제거될 것이라고 결정할 수도 있다.

[0214] NAL 유닛이 앵커 화상에 속하지 않거나, NAL 유닛이 깊이 뷰 성분 NAL 유닛 유형에 속하거나, 또는 NAL 유닛의 뷰 식별자가 "비 앵커 텍스처를 위해 요청됨"으로서 마킹된다는 결정에 응답하여 (560의 "아니오"), CDN 디바이스 (16) 는 NAL 유닛이 앵커 화상에 속하며, NAL 유닛이 깊이 뷰 성분 NAL 유닛 유형에 속하고, NAL 유닛의 뷰 식별자 (view_id) 가 "앵커 깊이를 위해 요청됨"으로서 마킹되지 않는지 여부를 결정할 수도 있다 (562).

[0215] NAL 유닛이 앵커 화상에 속하며, NAL 유닛이 깊이 뷰 성분 NAL 유닛 유형에 속하고, NAL 유닛의 뷰 식별자가 "앵커 깊이를 위해 요청됨"으로서 마킹되지 않는다는 결정에 응답하여 (562의 "예"), CDN 디바이스 (16) 는 NAL 유닛을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다 (554).
이런 식으로, CDN 디바이스 (16) 는, 만약 NAL 유닛의 anchor_pic_flag가 1과 동일하며, NAL 유닛의 nal_unit_type이 21이고, NAL 유닛의 view_id가 "앵커 깊이를 위해 요청됨"으로서 마킹되지 않으면, VCL NAL 유닛 또는 필러 데이터 NAL 유닛을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다.

[0216] "앵커 깊이를 위해 요청됨"으로서 마킹된 뷰 식별자들은 앵커 깊이 타겟 뷰 리스트 내에 있다. 따라서, CDN 디바이스 (16) 는, NAL 유닛이 앵커 화상에 속하는지 여부, NAL 유닛이 깊이 뷰 성분 NAL 유닛 유형에 속하는지 여부, 및 앵커 깊이가 타겟 뷰 리스트가 NAL 유닛의 뷰 식별자를 명시하는지 여부에 적어도 부분적으로 기초하여, 3DV 비트스트림으로부터 NAL 유닛을 제거할지 여부를 결정할 수도 있고, 여기서 깊이 뷰 성분 NAL 유닛 유형에 속한 NAL 유닛들은 깊이 뷰 성분들에 대한 코딩된 슬라이스 확장들을 캡슐화한다. 예를 들면, CDN 디바이스 (16) 는, NAL 유닛의 헤더에서의 앵커 화상 플래그 신텍스 엘리먼트가 1과 동일하며, NAL 유닛의 헤더에서의 NAL 유닛 유형 신텍스 엘리먼트가 21과 동일하고, NAL 유닛의 헤더에서의 뷰 식별자 신텍스 엘리먼트가 앵커 깊이 타겟 리스트에서의 뷰의 뷰 식별자와 동일한 경우, 그 NAL 유닛이 3DV 비트스트림으로부터 제거될 것이라고 결정할 수도 있다.

[0217] 더욱이, NAL 유닛이 앵커 화상에 속하지 않거나, NAL 유닛이 깊이 뷰 성분 NAL 유닛 유형에 속하지 않거나, 또는 NAL 유닛의 뷰 식별자가 "앵커 깊이를 위해 요청됨"으로서 마킹된다는 결정에 응답하여 (562의 "아니오"), CDN 디바이스 (16) 는 NAL 유닛이 비 앵커 화상에 속하며, NAL 유닛이 깊이 뷰 성분 NAL 유닛 유형에 속하고,

NAL 유닛의 뷰 식별자 (view_id) 가 "비 앵커 깊이를 위해 요청됨"으로서 마킹되지 않는지 여부를 결정할 수도 있다 (564).

[0218] NAL 유닛이 비 앵커 화상에 속하며, NAL 유닛이 깊이 뷰 성분 NAL 유닛 유형에 속하고, NAL 유닛의 뷰 식별자가 "비 앵커 깊이를 위해 요청됨"으로서 마킹되지 않는다는 결정에 응답하여 (564의 "예"), CDN 디바이스 (16) 는 NAL 유닛을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다 (554). 이런 식으로, CDN 디바이스 (16) 는, 만약 NAL 유닛의 anchor_pic_flag가 0과 동일하며, NAL 유닛의 nal_unit_type이 21이고, NAL 유닛의 view_id가 "비 앵커 깊이를 위해 요청됨"으로서 마킹되지 않으면, VCL NAL 유닛 또는 필터 데이터 NAL 유닛을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다.

[0219] "비 앵커 깊이를 위해 요청됨"으로서 마킹된 뷰 식별자들은 비 앵커 깊이 타겟 뷰 리스트에서의 뷰 식별자들이다. 따라서, CDN 디바이스 (16) 는 NAL 유닛이 앵커 화상에 속하는지 여부, NAL 유닛이 상기 깊이 뷰 성분 NAL 유닛 유형에 속하는지 여부, 및 비 앵커 깊이 타겟 뷰 리스트가 NAL 유닛의 뷰 식별자를 명시하는지 여부에 적어도 부분적으로 기초하여, 3DV 비트스트림으로부터 NAL 유닛을 제거할지 여부를 결정할 수도 있다. 예를 들면, CDN 디바이스 (16) 는, NAL 유닛의 헤더에서의 앵커 화상 플래그 선택스 엘리먼트가 0과 동일하며, NAL 유닛의 헤더에서의 NAL 유닛 유형 선택스 엘리먼트가 21과 동일하고, NAL 유닛의 헤더에서의 뷰 식별자 선택스 엘리먼트가 비 앵커 깊이 타겟 리스트에서의 뷰의 뷰 식별자와 동일한 경우, 그 NAL 유닛이 3DV 비트스트림으로부터 제거될 것이라고 결정할 수도 있다.

[0220] NAL 유닛이 비 앵커 화상에 속하지 않거나, NAL 유닛이 깊이 뷰 성분 NAL 유닛 유형에 속하지 않거나, 또는 NAL 유닛의 뷰 식별자가 "비 앵커 깊이를 위해 요청됨"으로서 마킹된다는 결정에 응답하여 (564의 "아니오"), CDN 디바이스 (16) 는 NAL 유닛의 NAL 참조 표시자 (nal_ref_idc) 가 0과 동일하며, NAL 유닛의 뷰간 플래그 (inter_view_flag) 가 0과 동일하고, NAL 유닛의 뷰 식별자가 뷰 식별자 타겟 리스트 (viewIdTargetList) 에서의 임의의 값과 동일한지 여부를 결정할 수도 있다 (566).

[0221] NAL 유닛의 NAL 참조 표시자가 0과 동일하며, NAL 유닛의 뷰간 플래그가 0과 동일하고, NAL 유닛의 뷰 식별자가 뷰 식별자 타겟 리스트에서의 임의의 값과 동일하다는 결정에 응답하여 (566의 "예"), CDN 디바이스 (16) 는 그 NAL 유닛을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다 (554). 이런 식으로, CDN 디바이스 (16) 는, VCL NAL 유닛 또는 필터 데이터 NAL 유닛의 nal_ref_idc가 0과 동일하고 inter_view_flag가 0과 동일하고 그 NAL 유닛의 view_id가 리스트 viewIdTargetList에서의 임의의 값과 일치하지 않으면, 그 NAL 유닛을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다.

[0222] 따라서, CDN 디바이스 (16) 는, NAL 유닛의 헤더에서의 앵커 화상 플래그 선택스 엘리먼트가 1과 동일하며, NAL 유닛의 헤더에서의 NAL 유닛 유형 선택스 엘리먼트가 21과 동일하고, NAL 유닛의 헤더에서의 뷰 식별자 선택스 엘리먼트가 앵커 깊이 타겟 리스트에서의 뷰의 뷰 식별자와 동일한 경우, 그 NAL 유닛이 3DV 비트스트림으로부터 제거될 것이라고 결정할 수도 있다. 더욱이, CDN 디바이스 (16) 는, NAL 유닛의 헤더에서의 앵커 화상 플래그 선택스 엘리먼트가 0과 동일하며, NAL 유닛의 헤더에서의 NAL 유닛 유형 선택스 엘리먼트가 21과 동일하고, NAL 유닛의 헤더에서의 뷰 식별자 선택스 엘리먼트가 비 앵커 깊이 타겟 리스트에서의 뷰의 뷰 식별자와 동일한 경우, 그 NAL 유닛이 3DV 비트스트림으로부터 제거될 것이라고 결정할 수도 있다.

[0223] 한편, NAL 유닛의 NAL 참조 표시자가 0과 동일하지 않거나, NAL 유닛의 뷰간 플래그가 0과 동일하지 않거나, 또는 NAL 유닛의 뷰 식별자가 뷰 식별자 타겟 리스트에서의 임의의 값과 동일하지 않다는 결정에 응답하여 (566의 "아니오"), CDN 디바이스 (16) 는 NAL 유닛의 NAL 유닛 유형이 21과 동일하고 깊이 존재 플래그 타겟 (depthPresentFlagTarget) 이 0과 동일한지 여부를 결정할 수도 있다 (568). NAL 유닛의 NAL 유닛 유형이 21과 동일하고 깊이 존재 플래그 타겟이 0과 동일하다는 결정에 응답하여 (568의 "예"), CDN 디바이스 (16) 는 NAL 유닛을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다 (554). 이런 식으로, CDN 디바이스 (16) 는, 만약 VCL NAL 유닛 또는 필터 데이터 NAL 유닛의 nal_unit_type이 21과 동일하고 depthPresentFlagTarget이 0과 동일하면, 그 NAL 유닛을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다.

[0224] NAL 유닛의 NAL 유닛 유형이 21과 동일하지 않거나 또는 깊이 존재 플래그 타겟이 0과 동일하지 않다는 결정에 응답하여 (568의 "아니오"), CDN 디바이스 (16) 는 깊이 존재 플래그 타겟이 1과 동일한지 여부를 결정할 수도 있다 (570). 깊이 존재 플래그 타겟이 1과 동일하다는 결정에 응답하여 (570의 "예"), CDN 디바이스 (16) 는 그 NAL 유닛을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다 (554). 그렇지 않으면, 깊이 존재 플래그 타겟이 1과 동일하지 않다는 결정에 응답하여 (570의 "아니오"), CDN 디바이스 (16) 는 그 NAL 유닛

을 "비트스트림으로부터 제거될 것"으로서 마킹하지 않을 수도 있다 (572). 이런 식으로, CDN 디바이스 (16) 는 만일 depthPresentFlagTarget이 1과 동일하면 VCL NAL 유닛 또는 필터 데이터 NAL 유닛을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다.

[0225] 도 12는 본 개시물의 하나 이상의 기법들에 따른, VCL NAL 유닛들 및 필터 데이터 NAL 유닛들을 비트스트림으로부터 제거될 것으로서 마킹하는 제 2 예 동작 (600) 을 도시하는 흐름도이다. CDN 디바이스 (16) 는 도 11의 예의 동작을 수행하는 대신 도 12의 예의 동작을 수행할 수도 있다. 동작 600은 사용 선택스 플래그 (use_texture_flag) 선택스 엘리먼트를 고려할 수도 있다. 다시 말하면, CDN 디바이스 (16) 가 동작 600을 사용하여 VCL NAL 유닛들 및 필터 데이터 NAL 유닛들을 비트스트림으로부터 제거할 것으로 결정하는 서브-비트스트림 추출 동작을 수행하는 경우, CDN 디바이스는, NAL 유닛의 헤더에서의 사용 텍스처 플래그 선택스 엘리먼트에 적어도 부분적으로 기초하여, 3DV 비트스트림으로부터 NAL 유닛을 제거할 것인지 여부를 결정할 수도 있다. 이전에 나타난 바와 같이, 사용 텍스처 플래그 선택스 엘리먼트는 NAL 유닛에 의해 캡슐화된 깊이 뷰 성분이 그 깊이 뷰 성분에 대응하는 텍스처 뷰 성분을 디코딩하는 일 없이 디코딩가능한지 여부를 나타낼 수도 있다. CDN 디바이스 (16) 는 도 5의 예에서 적어도 액션 316을 수행하는 부분으로서 도 12의 동작을 수행할 수도 있다. 도 12의 다음의 설명은 VCL NAL 유닛들 및 필터 데이터 NAL 유닛들에 적용가능할 수도 있다.

[0226] 도 12의 예에서 도시된 바와 같이, CDN 디바이스 (16) 는 NAL 유닛의 우선순위 식별자 (priority_id) 가 pIdTarget보다 큰지 여부를 결정할 수도 있다 (602). 위에서 논의된 바와 같이, NAL 유닛의 MVC NAL 유닛 헤더 확장은 priority_id 선택스 엘리먼트를 포함한다. pIdTarget은 서브-비트스트림 추출 프로세스, 이를테면 도 5의 서브-비트스트림 추출 프로세스 (298) 에 제공되는 파라미터이다. pIdTarget은 타겟 우선순위 레벨을 식별할 수도 있다. NAL 유닛의 우선순위 식별자가 pIdTarget보다 크다는 결정에 응답하여 (602의 "예"), CDN 디바이스 (16) 는 그 NAL 유닛을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다 (604). 이런 식으로, CDN 디바이스 (16) 는 VCL NAL 유닛 또는 필터 데이터 NAL 유닛의 priority_id가 pIdTarget보다 크면, 그 NAL 유닛을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다.

[0227] 한편, NAL 유닛의 우선순위 식별자가 pIdTarget보다 크지 않다는 결정에 응답하여 (602의 "아니오"), CDN 디바이스 (16) 는 NAL 유닛의 시간적 식별자 (temporal_id) 가 tIdTarget보다 크다고 결정할 수도 있다 (606). 위에서 논의된 바와 같이, NAL 유닛의 MVC NAL 유닛 헤더 확장은 temporal_id 선택스 엘리먼트를 포함하고, tIdTarget는 서브-비트스트림 추출 프로세스, 이를테면 서브-비트스트림 추출 프로세스 (298) 에 제공되는 파라미터이다. tIdTarget은 타겟 시간적 레벨을 식별할 수도 있다. NAL 유닛의 시간적 식별자가 tIdTarget보다 크다는 결정에 응답하여 (606의 "예"), CDN 디바이스 (16) 는 그 NAL 유닛을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다 (604). 이런 식으로, CDN 디바이스 (16) 는 VCL NAL 유닛 또는 필터 데이터 NAL 유닛의 temporal_id가 tIdTarget보다 크면, 그 NAL 유닛을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다.

[0228] NAL 유닛의 시간적 식별자가 tIdTarget보다 크지 않다는 결정에 응답하여 (606의 "아니오"), CDN 디바이스 (16) 는, 본 개시물의 기법들에 따라, NAL 유닛이 앵커 화상에 속하며, NAL 유닛이 깊이 뷰 성분 NAL 유닛 유형 이외의 NAL 유닛 유형에 속하며, NAL 유닛의 뷰 식별자 (view_id) 가 "앵커 텍스처를 위해 요청됨"으로서 마킹되지 않고 사용 텍스처 플래그 (use_texture_flag) 가 0이거나 또는 뷰 식별자가 "앵커 깊이를 위해 요청됨"으로서 마킹되지 않는지 여부를 결정할 수도 있다 (608). CDN 디바이스 (16) 는, NAL 유닛의 앵커 화상 플래그 (anchor_pic_flag) 에 기초하여, NAL 유닛이 앵커 화상에 속하는지 여부를 결정할 수도 있다. 깊이 뷰 성분 NAL 유닛 유형에 속하는 NAL 유닛들은 깊이 뷰 성분들에 대한 코딩된 슬라이스 확장들을 포함할 수도 있다. 일부 예들에서, 21과 동일한 nal_unit_type을 갖는 NAL 유닛들은 깊이 뷰 성분 NAL 유닛 유형에 속한다.

[0229] NAL 유닛이 앵커 화상에 속하며, NAL 유닛이 깊이 뷰 성분 NAL 유닛 유형 이외의 NAL 유닛 유형에 속하며, NAL 유닛의 뷰 식별자가 "앵커 텍스처를 위해 요청됨"으로서 마킹되지 않고, 사용 텍스처 플래그 (use_texture_flag) 가 0이거나 또는 뷰 식별자가 "앵커 깊이를 위해 요청됨"으로서 마킹되지 않는다는 결정에 응답하여 (608의 "예"), CDN 디바이스 (16) 는 그 NAL 유닛을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다 (604). 이런 식으로, CDN 디바이스 (16) 는, 만일 VCL NAL 유닛 또는 필터 데이터 NAL 유닛의 anchor_pic_flag가 1과 동일하며, 그 NAL 유닛의 nal_unit_type이 21과 동일하지 않고 다음의 두 조건들, 즉 NAL 유닛의 view_id가 "앵커 텍스처를 위해 요청됨"으로서 마킹되지 않는다는 및 use_texture_flag가 0이거나 또는 NAL 유닛의 view_id가 "앵커 깊이를 위해 요청됨"으로서 마킹되지 않는다는 양쪽 모두가 충족되면, 그 NAL 유닛을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다.

- [0230] 그러나, NAL 유닛이 앵커 화상에 속하지 않으며, NAL 유닛이 깊이 뷰 성분 NAL 유닛 유형에 속하며, NAL 유닛의 뷰 식별자가 "앵커 텍스처를 위해 요청됨"으로서 마킹되거나, 또는 사용 텍스처 플래그가 0이 아니고 뷰 식별자가 "앵커 깊이를 위해 요청됨"으로서 마킹된다는 결정에 응답하여 (608의 "아니오"), CDN 디바이스 (16) 는, 본 개시물의 기법들에 따라, NAL 유닛이 비 앵커 화상에 속하며, NAL 유닛이 깊이 뷰 성분 NAL 유닛 유형 이외의 NAL 유닛 유형에 속하며, NAL 유닛의 뷰 식별자 (view_id) 가 "비 앵커 텍스처를 위해 요청됨"으로서 마킹되지 않고, 사용 텍스처 플래그가 0이거나 또는 뷰 식별자가 "비 앵커 깊이를 위해 요청됨"으로서 마킹되지 않는지 여부를 결정할 수도 있다 (610).
- [0231] NAL 유닛이 앵커 화상에 속하며, NAL 유닛이 깊이 뷰 성분 NAL 유닛 유형 이외의 NAL 유닛 유형에 속하며, NAL 유닛의 뷰 식별자가 "비 앵커 텍스처를 위해 요청됨"으로서 마킹되지 않고, 사용 텍스처 플래그가 0이거나 또는 뷰 식별자가 "비 앵커 깊이를 위해 요청됨"으로서 마킹되지 않는다는 결정에 응답하여 (610의 "예"), CDN 디바이스 (16) 는 그 NAL 유닛을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다 (604). 이런 식으로, CDN 디바이스 (16) 는, 만일 VCL NAL 유닛 또는 필터 데이터 NAL 유닛의 anchor_pic_flag가 0과 동일하며, 그 NAL 유닛의 nal_unit_type이 21과 동일하지 않고 다음의 두 조건들, 즉 NAL 유닛의 view_id가 "비 앵커 텍스처를 위해 요청됨"으로서 마킹되지 않는다 및 use_texture_flag가 0이거나 또는 NAL 유닛의 view_id가 "비 앵커 깊이를 위해 요청됨"으로서 마킹되지 않는다는 양쪽 모두가 충족되면, 그 NAL 유닛을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다.
- [0232] NAL 유닛이 앵커 화상에 속하지 않거나, NAL 유닛이 깊이 뷰 성분 NAL 유닛 유형에 속하거나, NAL 유닛의 뷰 식별자가 "비 앵커 텍스처를 위해 요청됨"으로서 마킹되거나, 또는 사용 텍스처 플래그가 1이고 뷰 식별자가 "비 앵커 깊이를 위해 요청됨"으로서 마킹된다는 결정에 응답하여 (610의 "아니오"), CDN 디바이스 (16) 는 NAL 유닛이 앵커 화상에 속하며, NAL 유닛이 깊이 뷰 성분 NAL 유닛 유형에 속하고, NAL 유닛의 뷰 식별자 (view_id) 가 "앵커 깊이를 위해 요청됨"으로서 마킹되지 않는지 여부를 결정할 수도 있다 (612).
- [0233] NAL 유닛이 앵커 화상에 속하며, NAL 유닛이 깊이 뷰 성분 NAL 유닛 유형에 속하고, NAL 유닛의 뷰 식별자가 "앵커 깊이를 위해 요청됨"으로서 마킹되지 않는다는 결정에 응답하여 (612의 "예"), CDN 디바이스 (16) 는 그 NAL 유닛을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다 (604). 이런 식으로, CDN 디바이스 (16) 는, 만약 VCL NAL 유닛 또는 필터 데이터 NAL 유닛의 anchor_pic_flag가 1과 동일하며, 그 NAL 유닛의 nal_unit_type이 21이고, 그 NAL 유닛의 view_id가 "앵커 깊이를 위해 요청됨"으로서 마킹되지 않으면, 그 NAL 유닛을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다.
- [0234] 따라서, CDN 디바이스 (16) 는, NAL 유닛의 헤더에서의 앵커 화상 플래그 선택스 엘리먼트가 1과 동일하며, NAL 유닛의 헤더에서의 NAL 유닛 유형 선택스 엘리먼트가 21과 동일하며, NAL 유닛의 헤더에서의 뷰 식별자 선택스 엘리먼트가 앵커 텍스처 타겟 리스트에서의 임의의 뷰의 뷰 식별자와 동일하지 않고, NAL 유닛의 헤더에서의 사용 텍스처 플래그 선택스 엘리먼트가 0과 동일하거나 또는 NAL 유닛의 헤더에서의 뷰 식별자 선택스 엘리먼트가 앵커 깊이 타겟 리스트에서의 임의의 뷰의 뷰 식별자와 동일하지 않은 경우, 그 NAL 유닛이 3DV 비트스트림으로부터 제거될 것으로 결정할 수도 있다.
- [0235] 더욱이, NAL 유닛이 앵커 화상에 속하지 않거나, NAL 유닛이 깊이 뷰 성분 NAL 유닛 유형에 속하지 않거나, 또는 NAL 유닛의 뷰 식별자가 "앵커 깊이를 위해 요청됨"으로서 마킹된다는 결정에 응답하여 (612의 "아니오"), CDN 디바이스 (16) 는 NAL 유닛이 비 앵커 화상에 속하며, NAL 유닛이 깊이 뷰 성분 NAL 유닛 유형에 속하고, NAL 유닛의 뷰 식별자 (view_id) 가 "비 앵커 깊이를 위해 요청됨"으로서 마킹되지 않는지 여부를 결정할 수도 있다 (614).
- [0236] NAL 유닛이 비 앵커 화상에 속하며, NAL 유닛이 깊이 뷰 성분 NAL 유닛 유형에 속하고, NAL 유닛의 뷰 식별자가 "비 앵커 깊이를 위해 요청됨"으로서 마킹되지 않는다는 결정에 응답하여 (614의 "예"), CDN 디바이스 (16) 는 NAL 유닛을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다 (604). 이런 식으로, CDN 디바이스 (16) 는 VCL NAL 유닛 또는 필터 데이터 NAL 유닛의 앵커 화상 플래그가 0과 동일하며, 그 NAL 유닛의 nal_unit_type이 21과 동일하고, 그 NAL 유닛의 뷰 식별자가 "비 앵커 깊이를 위해 요청됨"으로서 마킹되지 않은 모든 VCL NAL 유닛들 및 필터 데이터 NAL 유닛들을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다.
- [0237] 따라서, CDN 디바이스 (16) 는, NAL 유닛의 헤더에서의 앵커 화상 플래그 선택스 엘리먼트가 0과 동일하며, NAL 유닛의 헤더에서의 NAL 유닛 유형 선택스 엘리먼트가 21과 동일하며, NAL 유닛의 헤더에서의 뷰 식별자 선택스 엘리먼트가 앵커 텍스처 타겟 리스트에서의 임의의 뷰의 뷰 식별자와 동일하지 않고, NAL 유닛의 헤더에서의 사

용 텍스처 플래그 선택스 엘리먼트가 0과 동일하거나 또는 NAL 유닛의 헤더에서의 뷰 식별자 선택스 엘리먼트가 비 앵커 깊이 타겟 리스트에서의 임의의 뷰의 뷰 식별자와 동일하지 않은 경우, 그 NAL 유닛이 3DV 비트스트림으로부터 제거될 것으로 결정할 수도 있다.

[0238] NAL 유닛이 비 앵커 화상에 속하지 않거나, NAL 유닛이 깊이 뷰 성분 NAL 유닛 유형에 속하지 않거나, 또는 NAL 유닛의 뷰 식별자가 "비 앵커 깊이를 위해 요청됨"으로서 마킹된다는 결정에 응답하여 (614의 "아니오"), CDN 디바이스 (16) 는 NAL 유닛의 NAL 참조 표시자 (nal_ref_idc) 가 0과 동일하며, NAL 유닛의 뷰간 플래그 (inter_view_flag) 가 0과 동일하고, NAL 유닛의 뷰 식별자가 뷰 식별자 타겟 리스트 (viewIdTargetList) 에서의 임의의 값과 동일한지 여부를 결정할 수도 있다 (616). NAL 유닛의 NAL 참조 표시자가 0과 동일하며, NAL 유닛의 뷰간 플래그가 0과 동일하고, NAL 유닛의 뷰 식별자가 뷰 식별자 타겟 리스트에서의 임의의 값과 동일하다는 결정에 응답하여 (616의 "예"), CDN 디바이스 (16) 는 그 NAL 유닛을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다 (604). 이런 식으로, CDN 디바이스 (16) 는 VCL NAL 유닛 또는 필터 데이터 NAL 유닛의 NAL 참조 표시자가 0과 동일하며, NAL 유닛의 뷰간 플래그가 0과 동일하고, NAL 유닛의 뷰 식별자가 뷰 식별자 타겟 리스트에서의 임의의 값과 동일한 모든 VCL NAL 유닛들 및 필터 데이터 NAL 유닛들을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다.

[0239] 한편, NAL 유닛의 NAL 참조 표시자가 0과 동일하지 않거나, NAL 유닛의 뷰간 플래그가 0과 동일하지 않거나, 또는 NAL 유닛의 뷰 식별자가 뷰 식별자 타겟 리스트에서의 임의의 값과 동일하지 않다는 결정에 응답하여 (616의 "아니오"), CDN 디바이스 (16) 는 NAL 유닛의 NAL 유닛 유형이 21과 동일하고 깊이 존재 플래그 타겟 (depthPresentFlagTarget) 이 0과 동일한지 여부를 결정할 수도 있다 (618). NAL 유닛의 NAL 유닛 유형이 21과 동일하고 깊이 존재 플래그 타겟이 0과 동일하다는 결정에 응답하여 (618의 "예"), CDN 디바이스 (16) 는 NAL 유닛을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다 (604). 이런 식으로, CDN 디바이스 (16) 는 NAL 유닛의 NAL 유닛 유형이 21과 동일하고 깊이 존재 플래그 타겟이 0과 동일한 모든 VCL NAL 유닛들 및 필터 데이터 NAL 유닛들을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다.

[0240] NAL 유닛의 NAL 유닛 유형이 21과 동일하지 않거나 또는 깊이 존재 플래그 타겟이 0과 동일하지 않다는 결정에 응답하여 (618의 "아니오"), CDN 디바이스 (16) 는 깊이 존재 플래그 타겟이 1과 동일한지 여부를 결정할 수도 있다 (620). 깊이 존재 플래그 타겟이 1과 동일하다는 결정에 응답하여 (620의 "예"), CDN 디바이스 (16) 는 그 NAL 유닛을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다 (604). 이런 식으로, CDN 디바이스 (16) 는 depthPresentFlagTarget이 1과 동일한 모든 VCL NAL 유닛들 및 필터 데이터 NAL 유닛들을 "비트스트림으로부터 제거될 것"으로서 마킹할 수도 있다. 그렇지 않으면, 깊이 존재 플래그가 1과 동일하지 않다는 결정에 응답하여 (620의 "아니오"), CDN 디바이스 (16) 는 NAL 유닛을 "비트스트림으로부터 제거될 것"으로서 마킹하지 않을 수도 있다 (622).

[0241] 다른 예들에서, NAL 유닛 헤더들은 use_texture_flag 선택스 엘리먼트들을 포함하지 않는다. 그런 예들에서, CDN 디바이스 (16) 는 텍스처 뷰 성분들의 depth_to_view_flag 선택스 엘리먼트들을 사용하여 use_texture_flag의 값을 도출할 수도 있다. 다르게 말하면, use_texture_flag는 NAL 유닛 헤더에서 시그널링되지 않고 텍스처 뷰 성분에서의 depth_to_view_flag가 use_texture_flag의 값을 도출하는데 사용된다. 일부 예들에서, use_texture_flag는 depth_to_view_flag와 동일하게 되도록 도출된다.

[0242] 도 13은 일 예의 MVC 디코딩 순서를 도시하는 개념도이다. 도 13의 예에서, 각각의 정사각형은 뷰 성분에 대응한다. 뷰 성분들의 각각은 텍스처 뷰 성분 및 깊이 뷰 성분을 포함할 수도 있다. 정사각형들의 열들은 액세스 유닛들에 대응한다. 각각의 액세스 유닛은 시간 인스턴스의 모든 뷰들의 코딩된 화상들을 포함하도록 정의될 수도 있다. 정사각형들의 행들은 뷰들에 대응한다. 도 13의 예에서, 액세스 유닛들은 T0, ..., T7로 라벨 표시되고 뷰들은 S0, ..., S7로 라벨 표시된다. 액세스 유닛의 각각의 뷰 성분이 다음 액세스 유닛의 임의의 뷰 성분 전에 디코딩되기 때문에, 도 13의 디코딩 순서는 시간 우선 코딩 (time-first coding) 이라고 지칭될 수도 있다. 도 13의 예에 도시된 바와 같이, 액세스 유닛들의 디코딩 순서는 뷰들의 출력 또는 디스플레이 순서와 동일하지 않을 수도 있다.

[0243] 위에서 언급했듯이, 뷰 순서 인덱스는 액세스 유닛에서 뷰 성분들의 디코딩 순서를 나타내는 인덱스이다. 따라서, 도 13의 예에서, 뷰 S0에서의 뷰 성분들의 뷰 순서 인덱스는 0일 수도 있으며, 뷰 S1에서의 뷰 성분들의 뷰 순서 인덱스는 1일 수도 있고, 뷰 S2에서의 뷰 성분들의 뷰 순서 인덱스는 2일 수도 있다는 등등이다.

[0244] 도 14는 일 예의 MVC 시간적 및 뷰간 예측 구조를 도시하는 개념도이다. 다시 말하면, 멀티-뷰 비디오 코딩을 위한 전형적인 MVC 예측 (각각의 뷰 내의 화상간 예측 및 뷰간 예측 양쪽 모두를 포함함) 구조가 도 14에 도

시되어 있다. 도 14의 예에서, 각각의 정사각형은 뷰 성분에 대응한다. 뷰 성분들의 각각은 텍스처 뷰 성분 및 깊이 뷰 성분을 포함할 수도 있다. "I"로 라벨 표시된 정사각형들은 인트라 예측된 뷰 성분들이다. "P"로 라벨 표시된 정사각형들은 단방향으로 인트라 예측된 뷰 성분들이다. "B" 및 "b"로 라벨 표시된 정사각형들은 양방향으로 인트라 예측된 뷰 성분들이다. "b"로 라벨 표시된 정사각형들은 "B" 라벨 표시된 정사각형들을 참조 화상들로서 사용할 수도 있다. 예측들은 화살표들에 의해 나타내어져, 가리켜진 대상 (pointed-to object) 이 예측 참조를 위해 가리키는 대상 (pointed-from object) 을 사용한다. 예를 들면, 제 1 정사각형에서부터 제 2 정사각형을 가리키는 화살표는, 제 1 정사각형이 인트라 예측에서 제 2정사각형에 대한 참조 화상으로서 이용가능하다는 것을 나타낸다. 도 14에서 수직 화살표들에 의해 나타낸 바와 같이, 동일한 액세스 유닛의 상이한 뷰들에서의 뷰 성분들은 참조 화상들로서 이용가능할 수도 있다. 하나의 액세스 유닛의 하나의 뷰 성분의 동일한 액세스 유닛의 다른 뷰 성분에 대한 참조 화상으로서의 사용은 뷰 간 예측으로서 지칭될 수도 있다.

[0245] 비디오 코더는 시간적 및 뷰 예측 참조들을 유연하게 배열하기 위해 참조 화상 리스트 구성 프로세스를 수행할 수도 있다. 참조 화상 리스트 구성 프로세스를 수행하는 것은 잠재적인 코딩 효율 이득들뿐 아니라 여러 내성 또한 제공할 수도 있는데, 참조 화상 섹션들 및 용장성 화상 메커니즘들이 뷰 차원으로 확장될 수 있기 때문이다.

[0246] 참조 화상 리스트 구성은 다음의 단계들을 포함할 수도 있다. 첫째, 비디오 코더는 H.264/AVC 표준에서 특정된 바와 같은 시간적 (인트라 뷰) 참조 화상들에 대한 참조 화상 리스트 초기화 프로세스를, 다른 뷰들로부터의 참조 화상들의 사용 없이, 적용할 수도 있다. 둘째, 비디오 코더는 인트라 뷰 참조 화상들이 SPS MVC 확장에서 발생하는 순서로 리스트의 말단에 뷰 간 참조 화상들을 첨부할 수도 있다. 셋째, 비디오 코더는 뷰 내 및 뷰 간 참조 화상들 양쪽 모두에 대해 참조 화상 리스트 수정 (reference picture list modification; RPLM) 프로세스를 적용한다. 비디오 코더는 뷰 간 참조 화상들을 MVC SPS 확장에서 특정되는 바와 같은 그것들의 인덱스 값들에 의해 RPLM 커맨드들로 식별할 수도 있다.

[0247] 하나 이상의 예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 그것들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현된다면, 그 기능들은 하나 이상의 명령들 또는 코드로서 컴퓨터 판독가능 매체 상에 저장되거나 또는 그것을 통해 송신될 수도 있고 하드웨어 기반 프로세싱 유닛에 의해 실행될 수도 있다. 컴퓨터 판독가능 매체들은, 데이터 저장 매체들과 같은 유형의 (tangible) 매체에 대응하는 컴퓨터 판독가능 저장 매체들, 또는 에컨대 통신 프로토콜에 따라 한 장소에서 다른 장소로 컴퓨터 프로그램의 전달을 용이하게 하는 임의의 매체를 포함하는 통신 매체들을 포함할 수도 있다. 이런 방식으로, 컴퓨터 판독가능 매체들은 일반적으로 (1) 비일시적 (non-transitory) 인 유형의 컴퓨터 판독가능 저장 매체들 또는 (2) 신호 또는 반송파와 같은 통신 매체에 해당할 수도 있다. 데이터 저장 매체들은 본 개시물에서 설명된 기법들의 구현을 위한 명령들, 코드 및/또는 데이터 구조들을 추출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 이용가능 매체들일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터 판독가능 매체를 포함할 수도 있다.

[0248] 비제한적인 예로, 이러한 컴퓨터 판독가능 저장 매체는 RAM, ROM, EEPROM, CD-ROM 또는 다른 광 디스크 스토리지, 자기 디스크 스토리지, 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 소망의 프로그램 코드를 컴퓨터에 의해 액세스될 수 있는 명령들 또는 데이터 구조들의 형태로 저장하는데 사용될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 접속이 컴퓨터 판독가능 매체로 적절히 칭해진다. 예를 들어, 명령들이 웹사이트, 서버, 또는 다른 원격 자원으로부터 동축 케이블, 광섬유 케이블, 연선 (twisted pair), 디지털 가입자 회선 (DSL), 또는 무선 기술들 이를테면 적외선, 라디오, 및/또는 마이크로파를 이용하여 송신된다면, 동축 케이블, 광섬유 케이블, 연선, DSL, 또는 적외선, 라디오, 및 마이크로파와 같은 무선 기술은 매체의 정의에 포함된다. 그러나, 컴퓨터 판독가능 저장 매체들 및 데이터 저장 매체들은 커넥션들, 반송파들, 신호들, 또는 다른 일시적인 매체들을 포함하지 않지만, 대신 비일시적 (non-transient), 유형의 저장 매체들을 지향하고 있음이 이해되어야 한다. 디스크 (disk 및 disc) 는 본원에서 사용되는 바와 같이, 콤팩트 디스크 (compact disc, CD), 레이저 디스크, 광 디스크, 디지털 다용도 디스크 (DVD), 플로피 디스크 (floppy disk) 및 블루레이 디스크를 포함하는데, disk들은 보통 데이터를 자기적으로 재생하지만, disc들은 레이저들으로써 광적으로 데이터를 재생한다. 상기한 것들의 조합들은 또한 컴퓨터 판독가능 매체들의 범위 내에 포함되어야 한다.

[0249] 명령들은 하나 이상의 프로세서들, 이를테면 하나 이상의 디지털 신호 프로세서들 (DSP들), 범용 마이크로프로세서들, 주문형 집적회로들 (ASIC들), 필드 프로그램가능 로직 어레이들 (FPGA들), 또는 다른 동등한 집적 또는

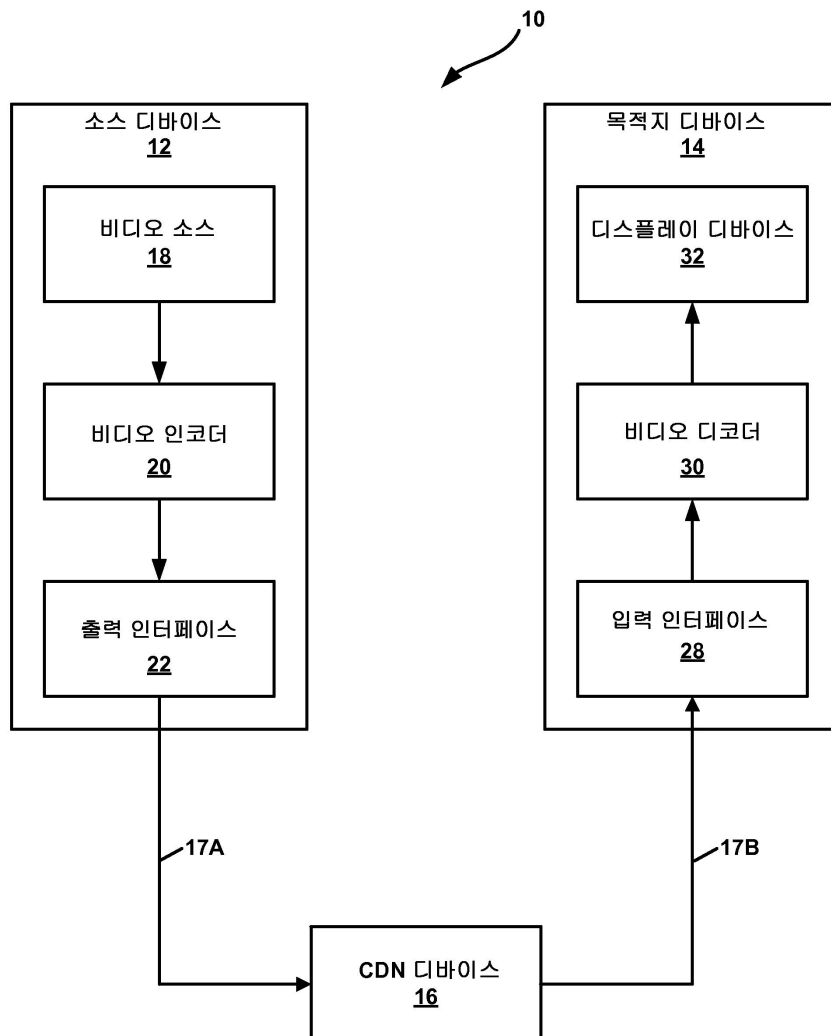
개별 로직 회로에 의해 실행될 수도 있다. 따라서, 본원에서 사용되는 바와 같은 용어 "프로세서"는 앞서의 구조 또는 본원에서 설명된 기법들의 구현에 적합한 임의의 다른 구조 중 임의의 것을 말할 수도 있다. 덧붙여서, 일부 양태들에서, 본원에서 설명된 기능성은 인코딩 및 디코딩을 위해 구성되는, 또는 결합형 코덱(codec)으로 통합되는 전용 하드웨어 및/또는 소프트웨어 모듈들 내에 제공될 수도 있다. 또한, 본 기법들은 하나 이상의 회로들 또는 로직 엘리먼트들 내에 완전히 구현될 수 있다.

[0250] 본 개시내용의 기법들은 무선 핸드셋, 집적회로(IC) 또는 IC들의 세트(예컨대, 칩셋)을 포함하는, 매우 다양한 디바이스들 또는 장치들로 구현될 수도 있다. 다양한 컴포넌트들, 모듈들, 또는 유닛들은 개시된 기법들을 수행하도록 구성된 디바이스들의 기능적 양태들을 강조하기 위해 본 개시물에서 설명되지만, 상이한 하드웨어 유닛들에 의한 실현을 반드시 요구하지는 않는다. 대신에, 위에서 설명된 바와 같이, 다양한 유닛들은 코덱 하드웨어 유닛에 결합되거나 또는 적합한 소프트웨어 및/또는 펌웨어와 함께, 위에서 설명된 바와 같은 하나 이상의 프로세서들을 포함하는, 상호운용적 하드웨어 유닛들의 컬렉션에 의해 제공될 수도 있다.

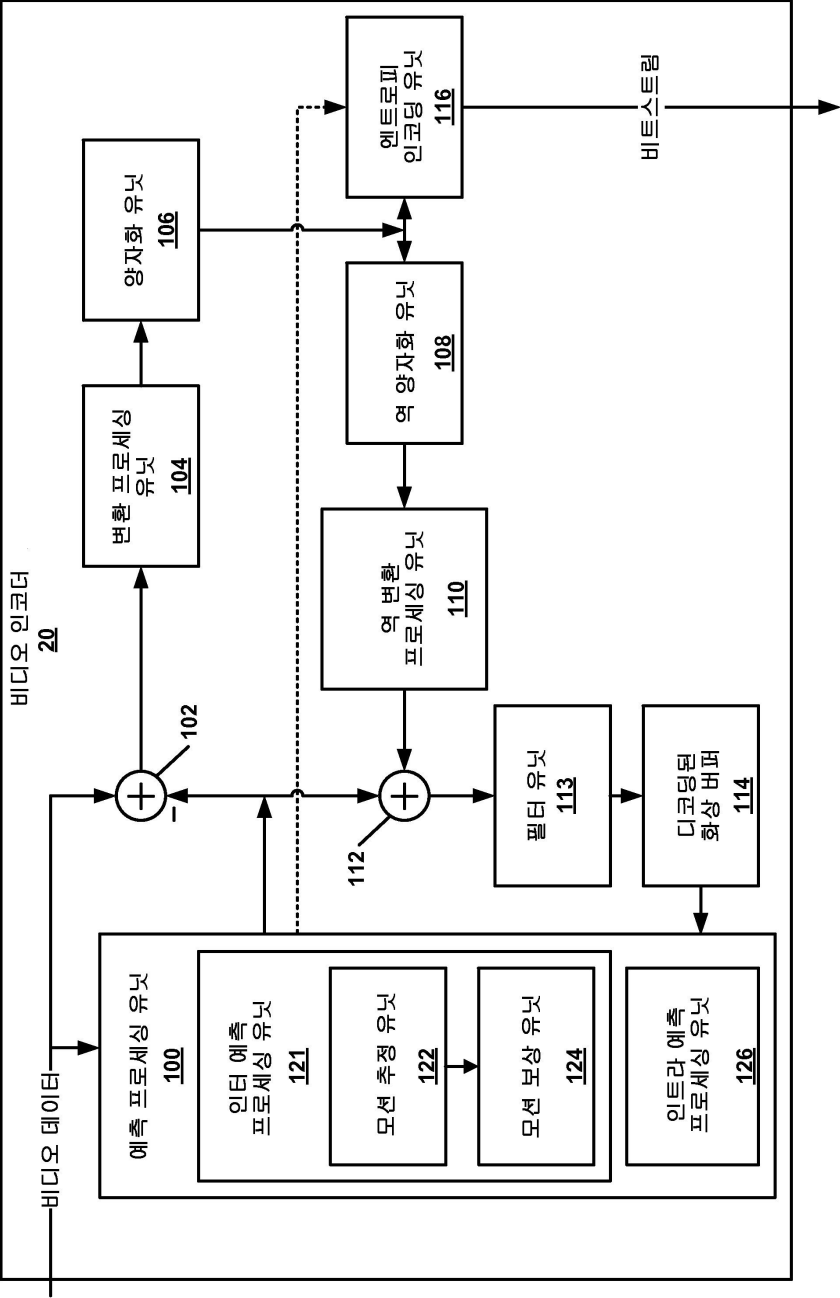
[0251] 다양한 예들이 설명되어 있다. 이들 및 다른 예들은 다음의 청구항들의 범위 내에 있다.

도면

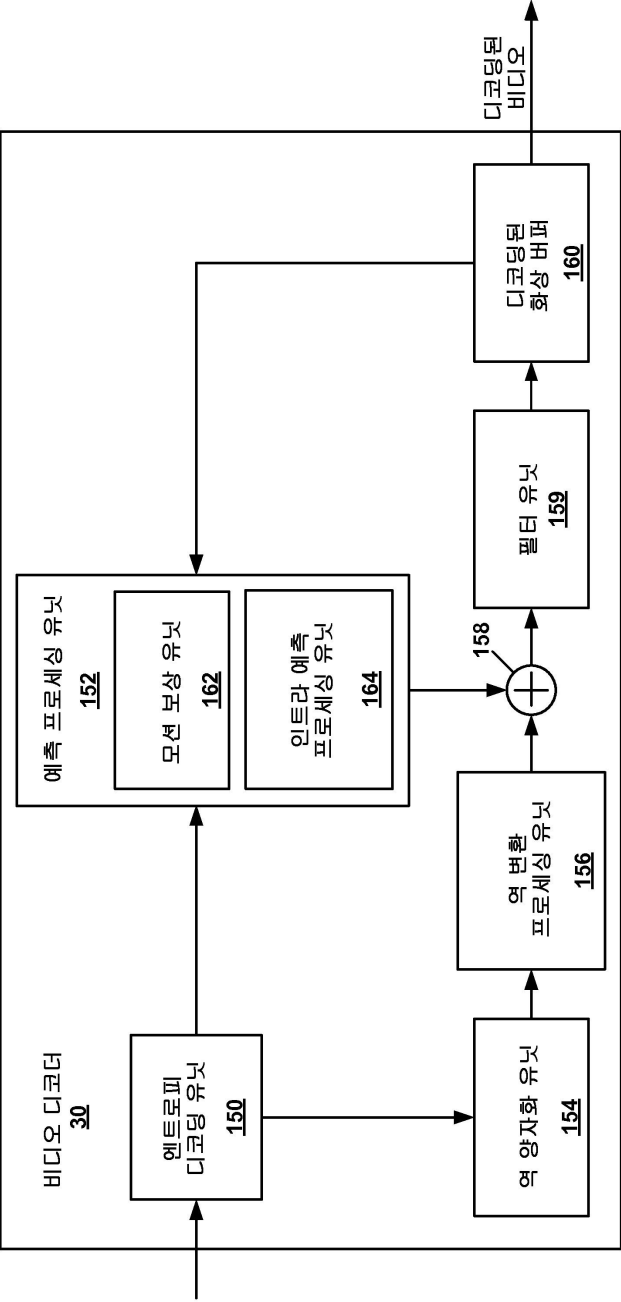
도면1



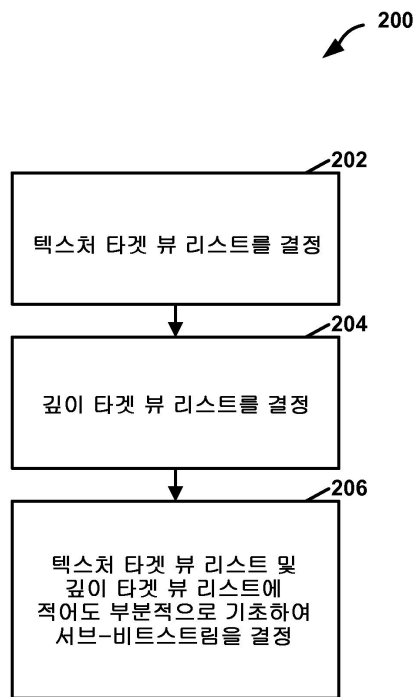
도면2



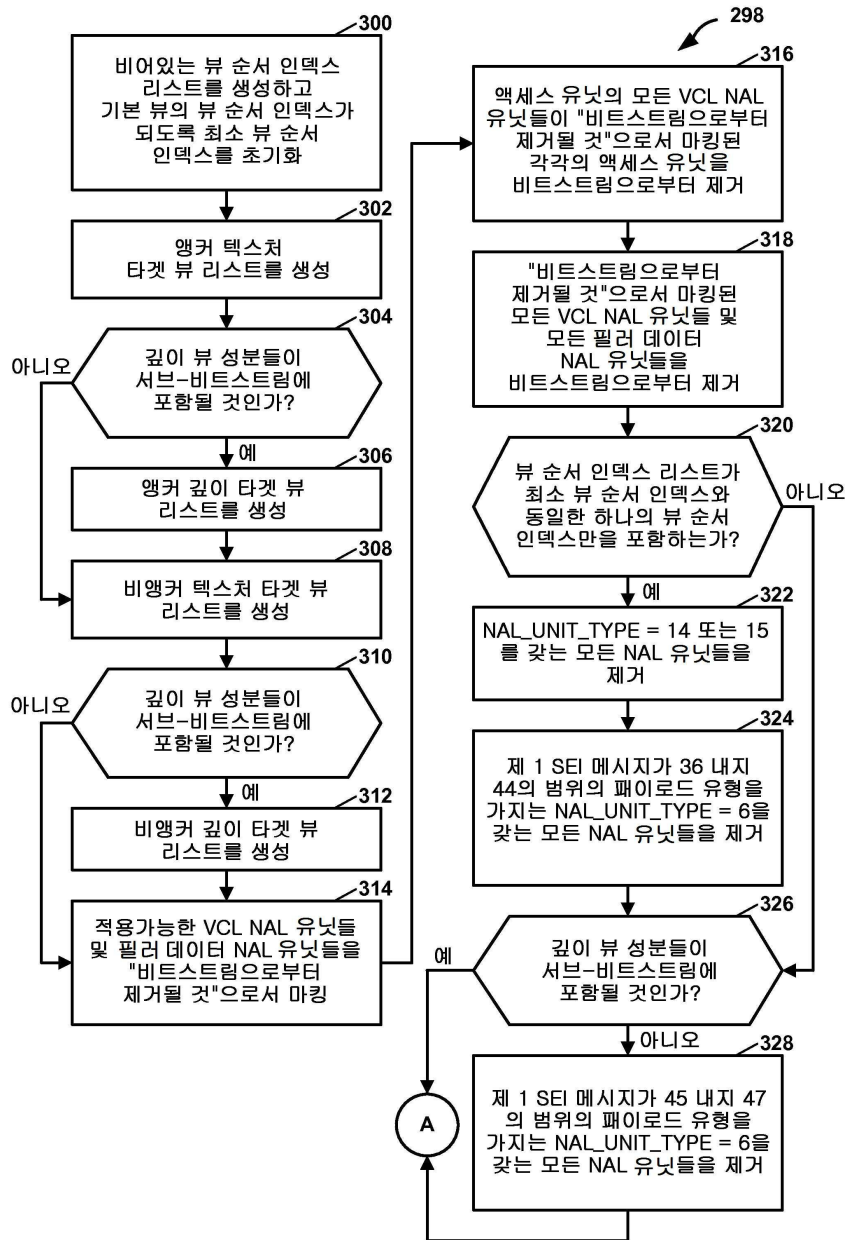
도면3



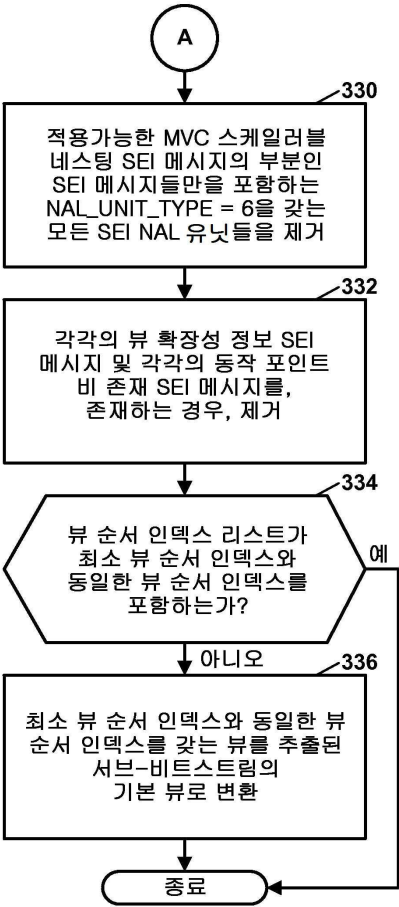
도면4



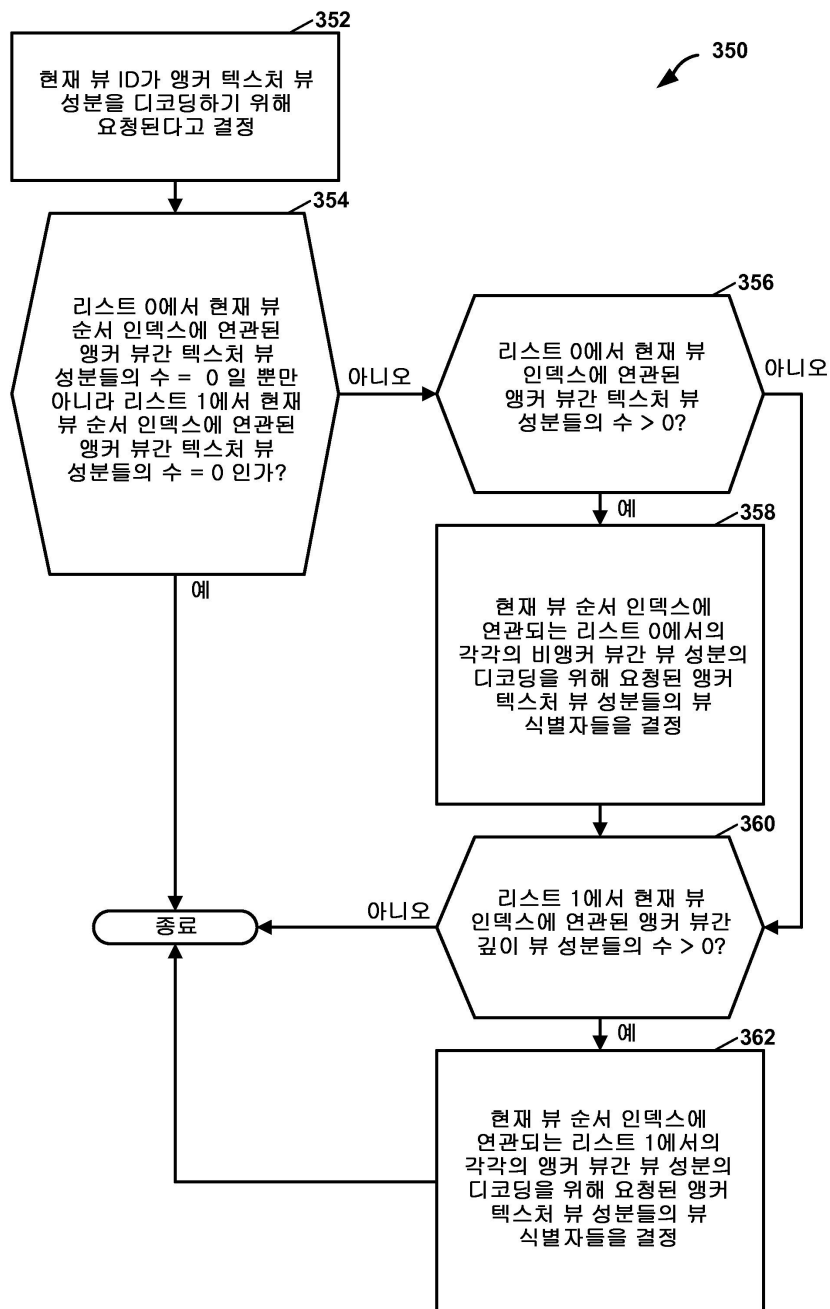
도면5



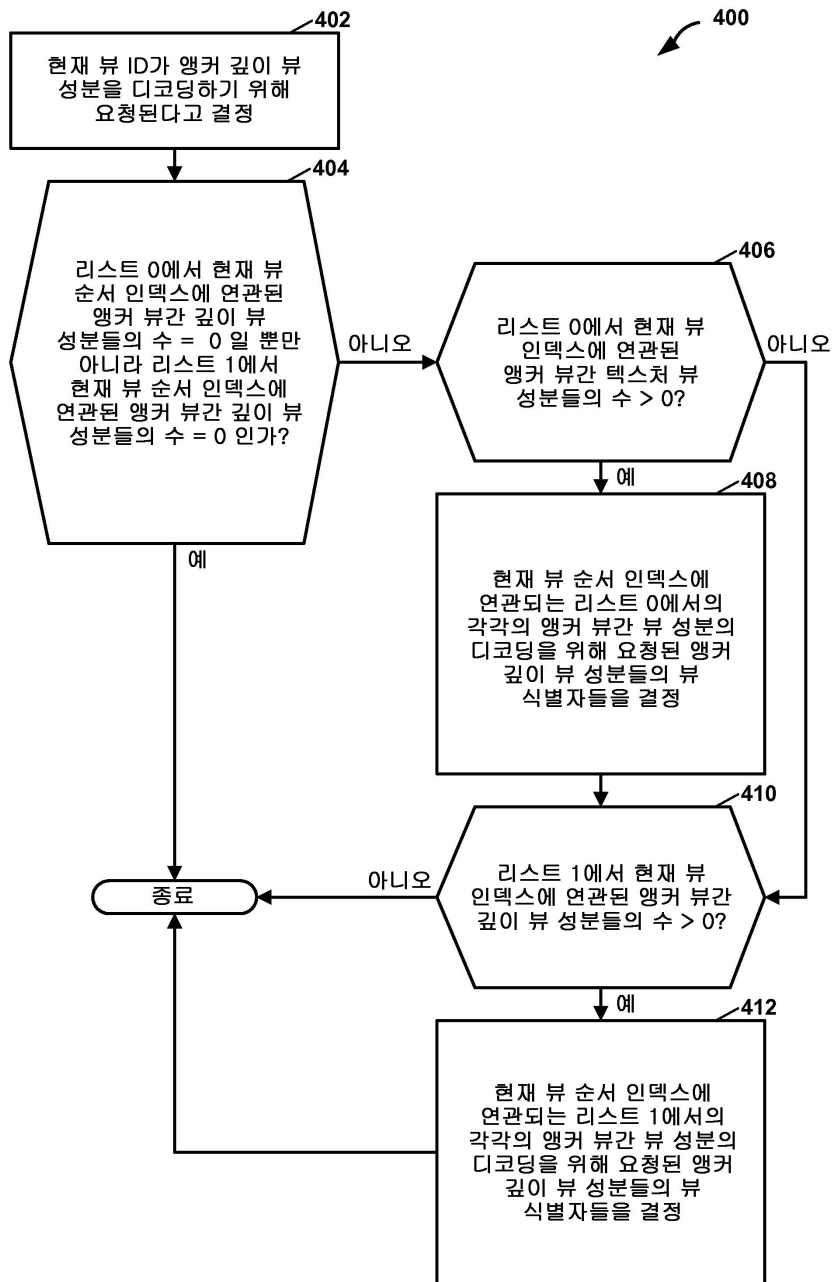
도면6



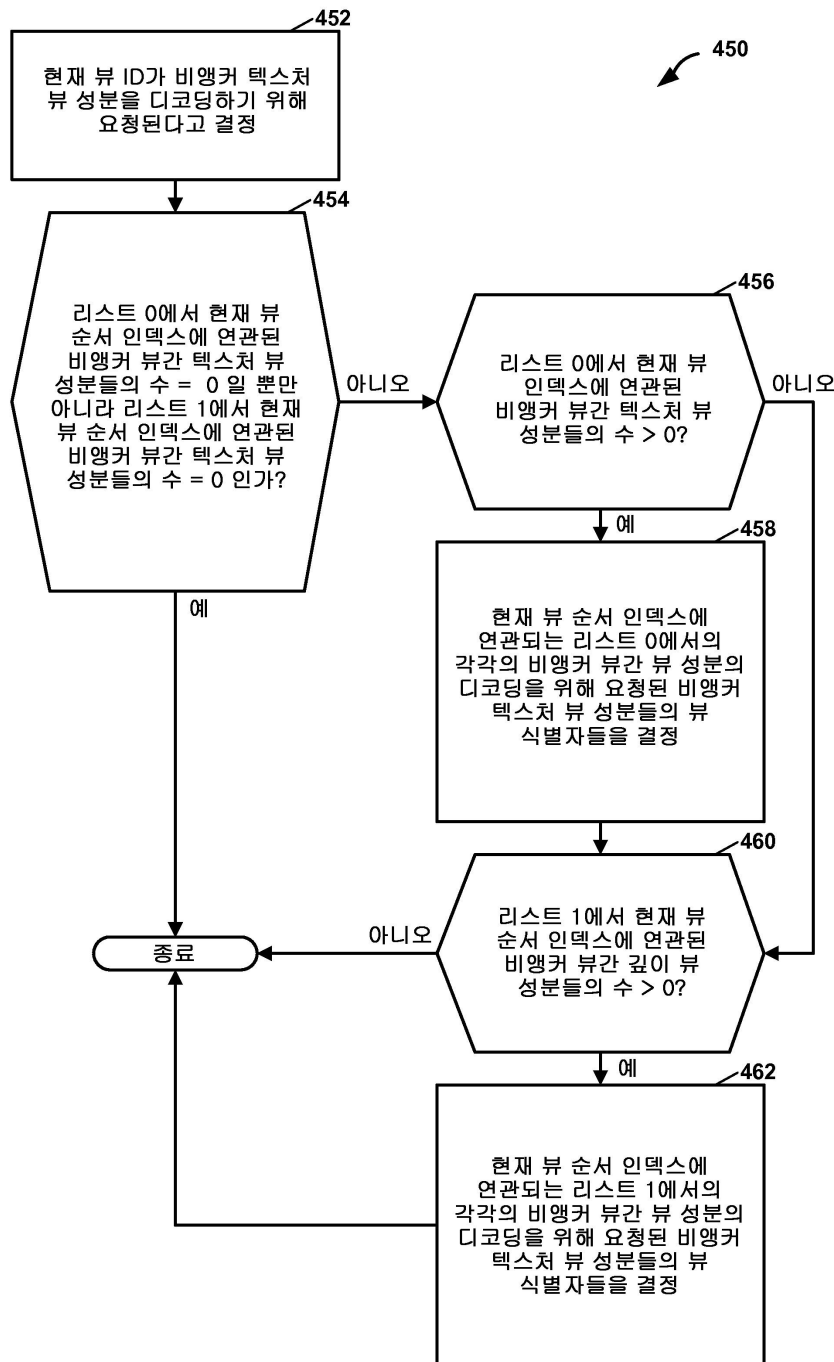
도면7



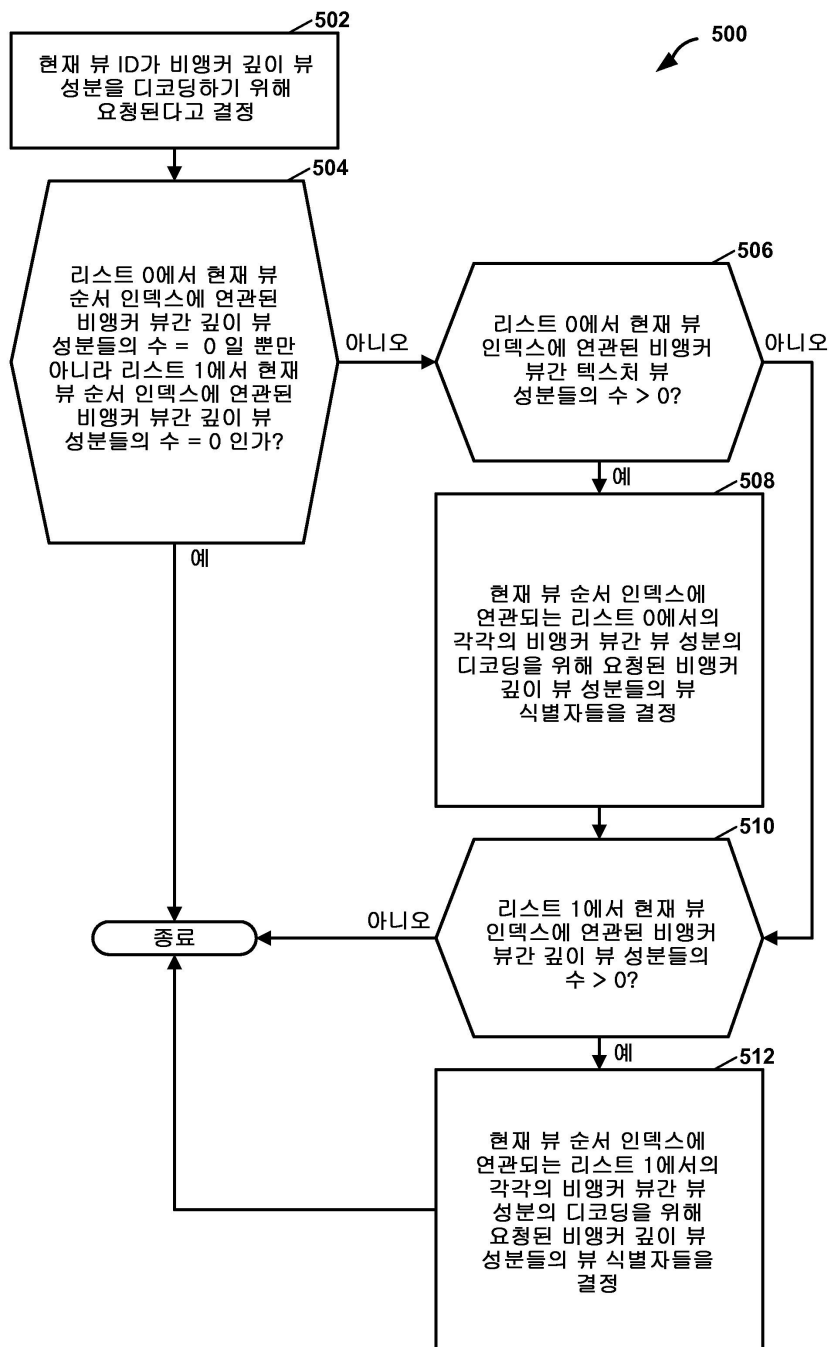
도면8



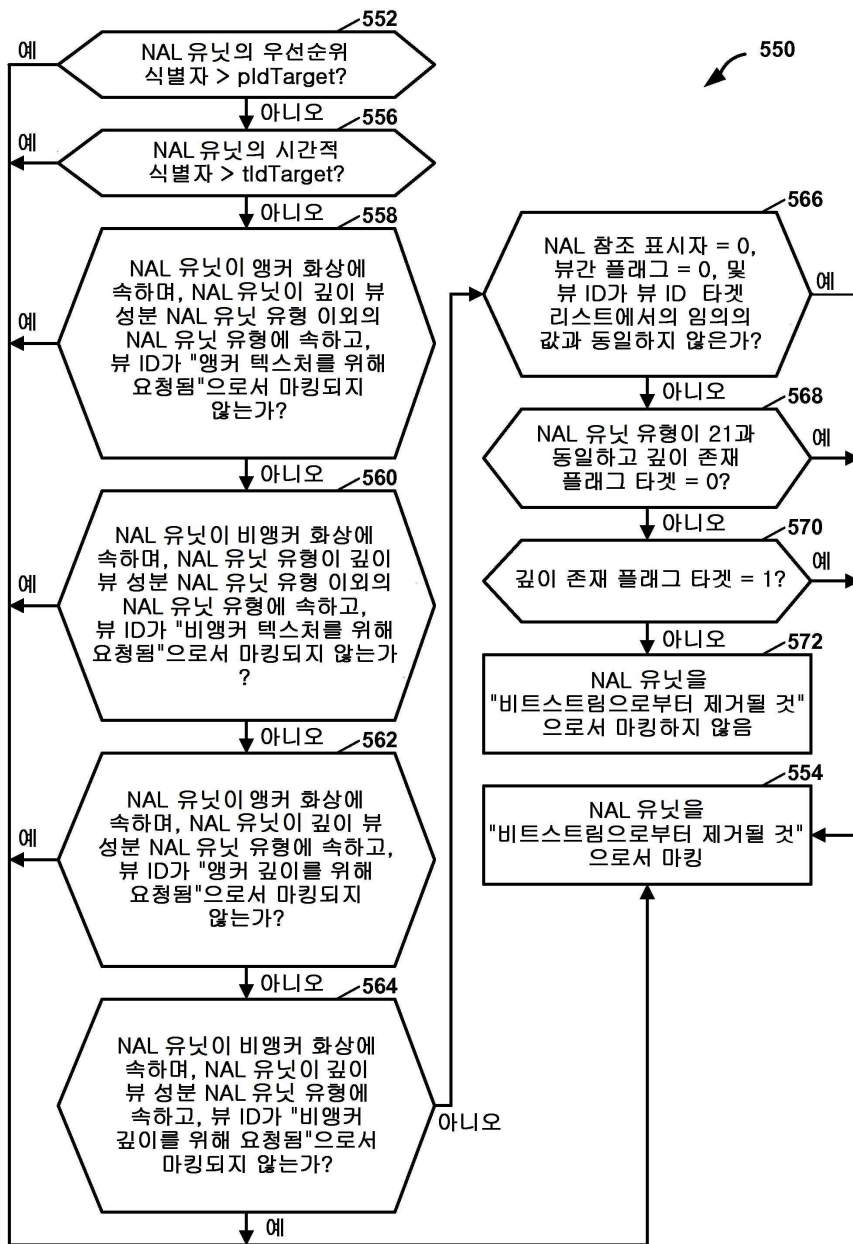
도면9



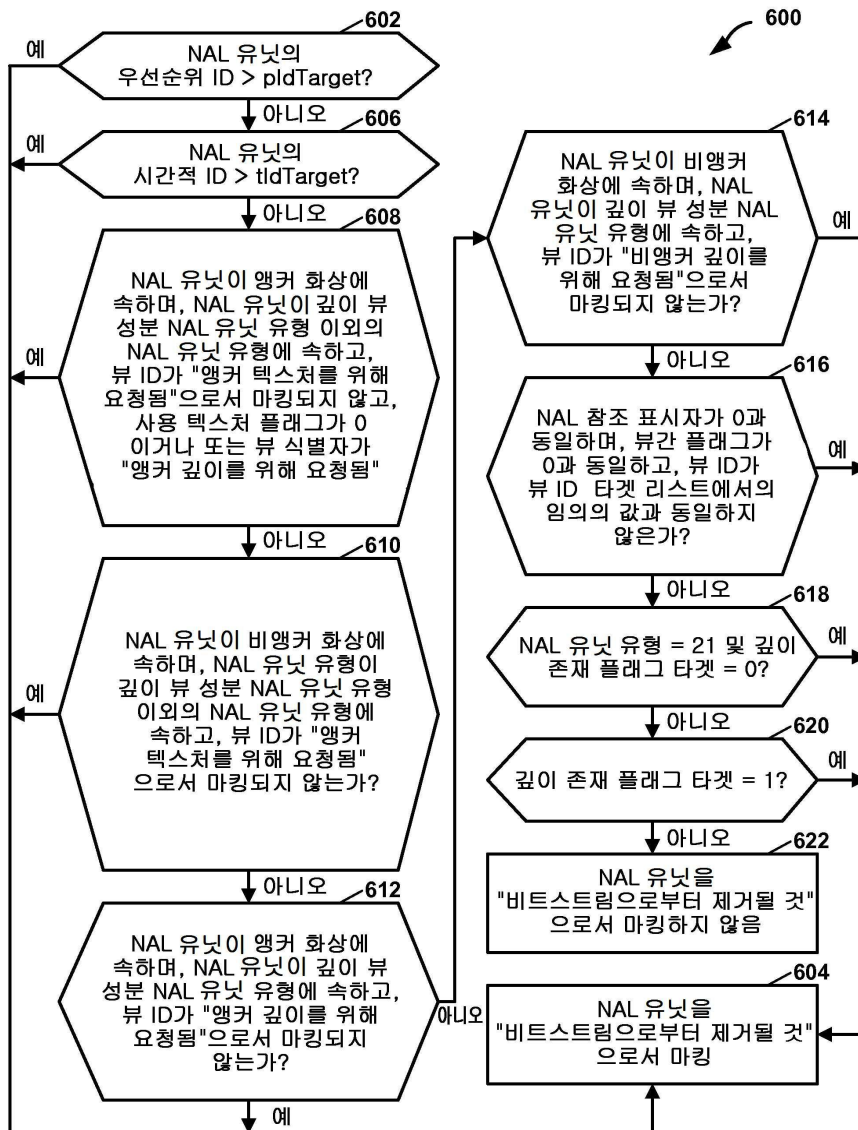
도면10



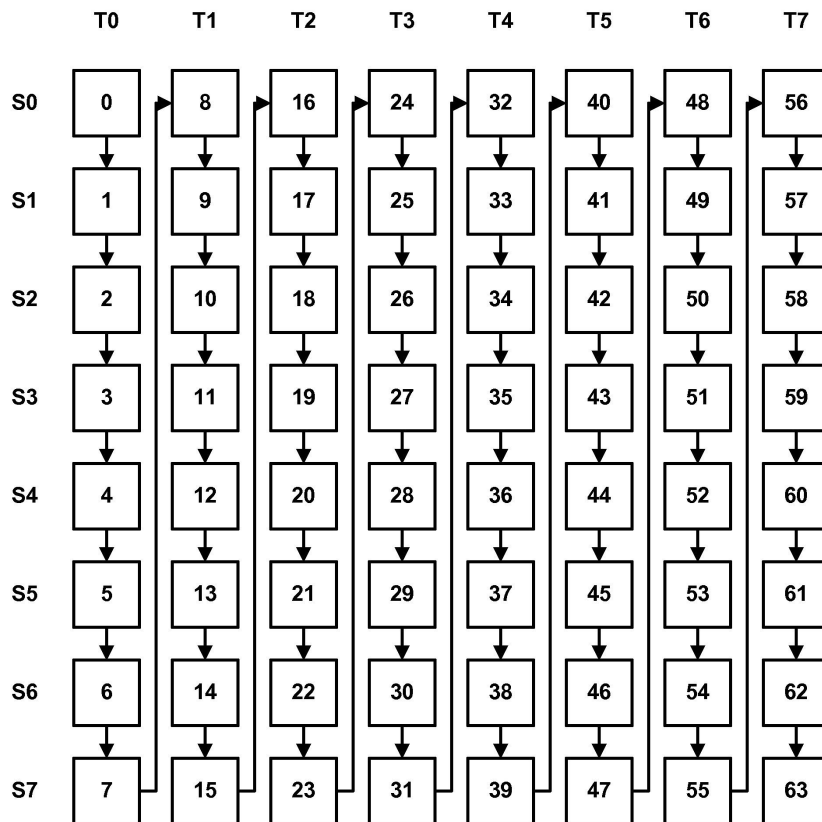
도면11



도면12



도면13



도면14

