



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2019년10월16일
(11) 등록번호 10-2033206
(24) 등록일자 2019년10월10일

(51) 국제특허분류(Int. Cl.)
H04N 19/597 (2014.01) H04N 19/103 (2014.01)
H04N 19/105 (2014.01) H04N 19/51 (2014.01)
(52) CPC특허분류
H04N 19/597 (2015.01)
H04N 19/103 (2015.01)
(21) 출원번호 10-2015-7017669
(22) 출원일자(국제) 2013년12월06일
심사청구일자 2018년11월20일
(85) 번역문제출일자 2015년07월01일
(65) 공개번호 10-2015-0092249
(43) 공개일자 2015년08월12일
(86) 국제출원번호 PCT/US2013/073649
(87) 국제공개번호 WO 2014/089475
국제공개일자 2014년06월12일
(30) 우선권주장
61/734,929 2012년12월07일 미국(US)
(뒷면에 계속)

(73) 특허권자
켈컴 인코포레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
(72) 발명자
장 리
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
천 잉
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
카르체비츠 마르타
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
(74) 대리인
특허법인코리아나

(56) 선행기술조사문헌
L. Zhang, et al. 3D-CE5.h: Merge candidates derivation from disparity vector shifting. JCT-3V of ITU-T and ISO/IEC. JCT3V-C0045 Ver.1, Jan. 10, 2013. pp.1-4
L. Zhang, et al. 3D-CE5.h: Merge candidates derivation from disparity vector. JCT-3V of ITU-T and ISO/IEC. JCT3V-B0048 Ver.2, Oct. 15, 2012. pp.1-4
Y. Chang, et al. 3D-CE5.h related: Depth-oriented Neighboring Block Disparity Vector (DoNBVDV) with virtual depth. JCT-3V of ITU-T and ISO/IEC. JCT3V-B0090 Ver.2, Oct. 11, 2012. pp.1-5

전체 청구항 수 : 총 10 항

심사관 : 이상래

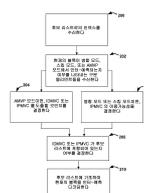
(54) 발명의 명칭 3D 비디오에 대한 진보된 병합/스킵 모드 및 진보된 모션 벡터 예측 (AMVP) 모드

(57) 요약

인터-뷰 예측된 모션 벡터 후보 (IPMVC) 및 인터-뷰 디스패리티 모션 벡터 후보 (IDMVC) 가 시프트된 디스패리티 벡터에 기초하면, 디스패리티 벡터가 IPMVC 및 IDMVC 에 대해 시프트되는 양이 상이한 기법들이 설명된다. 기법들은 또한 후보 리스트에 IDMVC 보다 IPMVC 의 포함을 우선적으로 처리하며, 후보 리스트에서 중복된 IPMVC

(뒷면에 계속)

대표도 - 도10



또는 ID MVC 가 존재하면 IP MVC 및 ID MVC 를 프루닝한다.

(52) CPC특허분류

H04N 19/105 (2015.01)

H04N 19/51 (2015.01)

(30) 우선권주장

61/747,350 2012년12월30일 미국(US)

14/098,151 2013년12월05일 미국(US)

명세서

청구범위

청구항 1

비디오 데이터를 코딩하는 방법으로서,

슬라이스, 화상 또는 시퀀스 내의 현재의 블록이 병합 모드에서 인터-예측되거나 스킵 모드에서 인터-예측되는 것을 결정하는 단계;

상기 현재의 블록이 병합 모드에서 인터-예측되거나 스킵 모드에서 인터-예측되는 것에 기초하여,

제 1 오프셋 값만큼 시프트된 디스패리티 벡터에 의해 참조되는 참조 뷰에서, 상기 현재의 블록에 대해, 참조 블록을 식별하는 단계;

IPMVC 또는 IDMVC 가 후보 리스트에 포함되는 것인지 여부를 결정하는 단계로서,

상기 IPMVC 는 상기 현재의 블록에 대해, 상기 제 1 오프셋 값만큼 시프트된 상기 디스패리티 벡터에 의해 참조되는 참조 뷰에서의 상기 참조 블록에 대한 모션 정보를 포함하고,

상기 IDMVC 는 상기 현재의 블록에 대해, 상기 참조 블록을 식별하기 위해 상기 디스패리티 벡터를 시프트하는데 이용되는 상기 제 1 오프셋 값과는 상이한 제 2 오프셋 값만큼 시프트된 상기 디스패리티 벡터를 포함하고, 상기 제 2 오프셋 값은, 상기 디스패리티 벡터를 수평으로 시프트하기 위한 고정된 오프셋 값을 포함하고, 상기 고정된 오프셋 값은 상기 슬라이스, 화상 또는 시퀀스 내의 모든 예측 유닛들에 대해 고정되는, 상기 IPMVC 또는 IDMVC 가 후보 리스트에 포함되는 것인지 여부를 결정하는 단계;

제 2 IPMVC 를 결정하는 단계로서, 상기 제 2 IPMVC 는, 상기 디스패리티 벡터를 시프트하지 않고서 상기 디스패리티 벡터에 의해 참조되는 상기 참조 뷰에서의 제 2 참조 블록에 대한 모션 정보를 포함하는, 상기 제 2 IPMVC 를 결정하는 단계;

제 1 IPMVC 의 모션 정보와 상기 제 2 IPMVC 의 모션 정보를 비교하는 단계로서, 상기 제 1 IPMVC 는 상기 현재의 블록에 대해, 상기 제 1 오프셋 값만큼 시프트된 상기 디스패리티 벡터에 의해 참조되는 참조 뷰에서의 제 1 참조 블록에 대한 모션 정보를 포함하는, 제 1 IPMVC 의 모션 정보와 상기 제 2 IPMVC 의 모션 정보를 비교하는 단계; 및

상기 후보 리스트에 기초하여, 상기 참조 뷰와는 상이한, 현재의 뷰에서의 현재의 화상에서의 상기 현재의 블록을 인터-예측 코딩하는 단계

를 포함하고,

상기 IPMVC 또는 IDMVC 가 후보 리스트에 포함되는 것인지 여부를 결정하는 단계는 상기 제 2 IPMVC 에 대한 모션 정보가 상기 제 1 IPMVC 에 대한 모션 정보와 동일하면 상기 제 1 IPMVC 가 상기 후보 리스트에 포함되지 않는다고 결정하는 것, 그리고 상기 제 2 IPMVC 에 대한 모션 정보가 상기 제 1 IPMVC 에 대한 모션 정보와는 상이하면 상기 제 1 IPMVC 가 상기 후보 리스트에 포함된다고 결정하는 것을 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 2

제 1 항에 있어서,

비교하는 단계는 상기 제 1 IPMVC 를 오직 상기 제 2 IPMVC 와 비교하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 3

제 1 항에 있어서,

상기 참조 블록의 상기 모션 정보는, 참조 화상 리스트들에 대한 하나 이상의 참조 인덱스들 및 모션 벡터 중

적어도 하나를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 4

제 1 항에 있어서,

상기 후보 리스트에 대한 인덱스를 수신하는 단계를 더 포함하고,

상기 현재의 블록이 병합 모드에서 인터-예측되거나 스킵 모드에서 인터-예측되는 것을 결정하는 단계는, 상기 현재의 블록이 병합 모드에서 인터-예측되거나 스킵 모드에서 인터-예측되는 것을 나타내는 구문 엘리먼트들을 디코딩하는 단계를 포함하며,

인터-예측 코딩하는 단계는 상기 후보 리스트에 대한 수신된 상기 인덱스에 의해 식별되는 정보에 기초하여 상기 현재의 블록을 인터-예측 디코딩하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 5

제 1 항에 있어서,

인터-예측 코딩하는 단계는 상기 후보 리스트에서의 정보에 기초하여 상기 현재의 블록을 인터-예측 인코딩하는 단계를 포함하고,

상기 방법은,

상기 현재의 블록이 병합 모드에서 인터-예측되거나 스킵 모드에서 인터-예측되는 것을 나타내는 구문 엘리먼트들을 시그널링하는 단계; 및

상기 후보 리스트에서의 상기 정보를 식별하는데 이용되는 상기 후보 리스트에 대한 인덱스를 시그널링하는 단계를 더 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 6

비디오 데이터를 코딩하는 디바이스로서,

슬라이스, 화상 또는 시퀀스 내의 현재의 블록이 병합 모드에서 인터-예측되거나 스킵 모드에서 인터-예측되는 것을 결정하는 수단;

상기 현재의 블록에 대해, 제 1 오프셋 값만큼 시프트된 디스패리티 벡터에 의해 참조되는 참조 뷰에서의 참조 블록을 식별하는 수단;

IPMVC 또는 IDMVC 가 후보 리스트에 포함되는 것인지 여부를 결정하는 수단으로서, 상기 IPMVC 는 상기 현재의 블록에 대해, 상기 제 1 오프셋 값만큼 시프트된 상기 디스패리티 벡터에 의해 참조되는 참조 뷰에서의 상기 참조 블록에 대한 모션 정보를 포함하고, 상기 IDMVC 는 상기 현재의 블록에 대해, 상기 참조 블록을 식별하기 위해 상기 디스패리티 벡터를 시프트하는데 이용되는 상기 제 1 오프셋 값과는 상이한 제 2 오프셋 값만큼 시프트된 상기 디스패리티 벡터를 포함하고, 상기 제 2 오프셋 값은, 상기 디스패리티 벡터를 수평으로 시프트하기 위한 고정된 오프셋 값을 포함하고, 상기 고정된 오프셋 값은 상기 슬라이스, 화상 또는 시퀀스 내의 모든 예측 유닛들에 대해 고정되는, 상기 IPMVC 또는 IDMVC 가 후보 리스트에 포함되는 것인지 여부를 결정하는 수단;

제 2 IPMVC 를 결정하는 수단으로서, 상기 제 2 IPMVC 는, 상기 디스패리티 벡터를 시프트하지 않고서 상기 디스패리티 벡터에 의해 참조되는 상기 참조 뷰에서의 제 2 참조 블록에 대한 모션 정보를 포함하는, 상기 제 2 IPMVC 를 결정하는 수단;

제 1 IPMVC 의 모션 정보와 상기 제 2 IPMVC 의 모션 정보를 비교하는 수단으로서, 상기 제 1 IPMVC 는 상기 현재의 블록에 대해, 상기 제 1 오프셋 값만큼 시프트된 상기 디스패리티 벡터에 의해 참조되는 참조 뷰에서의 제 1 참조 블록에 대한 모션 정보를 포함하는, 제 1 IPMVC 의 모션 정보와 상기 제 2 IPMVC 의 모션 정보를 비교하는 수단; 및

상기 후보 리스트에 기초하여, 상기 참조 뷰와는 상이한, 현재의 뷰에서의 현재의 화상에서의 상기 현재의 블록을 인터-예측 코딩하는 수단

을 포함하고,

상기 IPMVC 또는 ID MVC 가 후보 리스트에 포함되는 것인지 여부를 결정하는 것은 상기 제 2 IPMVC에 대한 모션 정보가 상기 제 1 IPMVC 에 대한 모션 정보와 동일하면 상기 제 1 IPMVC 가 상기 후보 리스트에 포함되지 않는다고 결정하는 것, 그리고 상기 제 2 IPMVC에 대한 모션 정보가 상기 제 1 IPMVC 에 대한 모션 정보와는 상이하면 상기 제 1 IPMVC 가 상기 후보 리스트에 포함된다고 결정하는 것을 포함하는, 비디오 데이터를 코딩하는 디바이스.

청구항 7

제 6 항에 있어서,

상기 현재의 블록은 제 1 블록을 포함하며, 상기 디스패리티 벡터는 제 1 디스패리티 벡터를 포함하고, 상기 참조 뷰는 제 1 참조 뷰를 포함하며,

상기 디바이스는,

병합 모드에서 인터-예측되거나 스킵 모드에서 인터-예측되는 제 2 블록에 대한 제 1 IPMVC 를 결정하는 수단으로서, 상기 제 1 IPMVC 는, 상기 제 2 블록에 대해, 제 3 오프셋 값만큼 시프트된 제 2 디스패리티 벡터에 의해 참조되는 제 2 참조 뷰에서의 제 2 참조 블록에 대한 모션 정보를 포함하는, 상기 제 2 블록에 대한 제 1 IPMVC 를 결정하는 수단;

상기 제 2 블록에 대한 제 2 IPMVC 를 결정하는 수단으로서, 상기 제 2 IPMVC 는, 상기 제 2 블록에 대해, 상기 제 2 디스패리티 벡터를 시프트하지 않고서, 상기 제 2 디스패리티 벡터에 의해 참조되는 상기 제 2 참조 뷰에서의 제 3 참조 블록에 대한 모션 정보를 포함하는, 상기 제 2 블록에 대한 제 2 IPMVC 를 결정하는 수단;

상기 제 1 IPMVC 를 오직 상기 제 2 IPMVC 와 비교하는 수단;

상기 제 2 IPMVC 가 상기 제 1 IPMVC 와는 상이하다고 나타내는 상기 비교에 기초하여 상기 제 2 블록에 대한 후보 리스트에 상기 제 1 IPMVC 를 포함시키는 수단; 및

상기 제 2 블록에 대한 상기 후보 리스트에 기초하여 상기 제 2 블록을 인터-예측 코딩하는 수단을 더 포함하는, 비디오 데이터를 코딩하는 디바이스.

청구항 8

비디오 데이터를 코딩하는 디바이스로서,

제 6 항에 기재된 디바이스를 포함하는 비디오 코더를 포함하는, 비디오 데이터를 코딩하는 디바이스.

청구항 9

제 8 항에 있어서,

상기 디바이스는,

집적 회로 (IC);

마이크로프로세서; 및

무선 통신 디바이스

중 하나를 포함하는, 비디오 데이터를 코딩하는 디바이스.

청구항 10

명령들이 저장된 컴퓨터 판독가능 저장 매체로서,

상기 명령들은 실행될 때 비디오 데이터를 코딩하는 디바이스의 하나 이상의 프로세서들로 하여금 제 1 항 내지 제 5 항 중 어느 한 항에 기재된 방법을 수행하게 하는, 명령들이 저장된 컴퓨터 판독가능 저장 매체.

청구항 11

삭제

청구항 12

삭제

청구항 13

삭제

청구항 14

삭제

청구항 15

삭제

청구항 16

삭제

청구항 17

삭제

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

청구항 21

삭제

청구항 22

삭제

청구항 23

삭제

청구항 24

삭제

청구항 25

삭제

청구항 26

삭제

청구항 27

삭제

청구항 28

삭제

청구항 29

삭제

청구항 30

삭제

발명의 설명

기술 분야

[0001] 본 출원은 2012년 12월 7일자에 출원된 미국 가출원 번호 제 61/734,929호, 및 2012년 12월 30일자에 출원된 미국 가출원 번호 제 61/747,350호의 이익을 주장하며, 각각의 전체 내용이 본원에서 참조로 포함된다.

[0002] 기술 분야

[0003] 본 개시물은 비디오 코딩에 관한 것이다.

배경 기술

[0004] 디지털 비디오 능력들은, 디지털 텔레비전, 디지털 직접 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인 휴대정보 단말기들 (PDAs), 랩탑 또는 데스크탑 컴퓨터들, 태블릿 컴퓨터들, e-북 리더들, 디지털 카메라들, 디지털 리코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 무선 전화기들, 소위 "스마트폰들", 원격 화상회의 디바이스들, 비디오 스트리밍 디바이스들 등을 포함한, 광범위한 디바이스들에 포함될 수 있다. 디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, 파트 10, AVC (Advanced Video Coding), 현재 개발중인 HEVC (High Efficiency Video Coding) 표준, 및 이런 표준들의 확장판들에 의해 정의된 표준들에서 설명되는 비디오 코딩 기법들과 같은, 비디오 압축 기법들을 구현한다. 비디오 디바이스들은 이런 비디오 압축 기법들을 구현함으로써, 디지털 비디오 정보를 좀더 효율적으로 송신, 수신, 인코딩, 디코딩, 및/또는 저장할 수도 있다.

[0005] 비디오 압축 기법들은 비디오 시퀀스들에 고유한 리던던시를 감소시키거나 또는 제거하기 위해 공간 (인트라-화상) 예측 및/또는 시간 (인터-화상) 예측을 수행한다. 블록-기반 비디오 코딩에 있어, 비디오 슬라이스 (즉, 비디오 프레임 또는 비디오 프레임의 부분) 는 비디오 블록들로 파티셔닝될 수도 있으며, 이 비디오 블록들은 또한 트리블록들, 코딩 유닛들 (CUs) 및/또는 코딩 노드들로서 지칭될 수도 있다. 화상의 인트라-코딩된 (I) 슬라이스에서 비디오 블록들은 동일한 화상에서 이웃하는 블록들에서의 참조 샘플들에 대한 공간 예측을 이용하여 인코딩된다. 화상의 인터-코딩된 (P 또는 B) 슬라이스에서 비디오 블록들은 동일한 화상에서 이웃하는 블록들에서의 참조 샘플들에 대한 공간 예측, 또는 다른 참조 화상들에서의 참조 샘플들에 대한 시간 예측을 이용할 수도 있다. 화상들은 프레임들로 지칭될 수 있으며, 참조 화상들은 참조 프레임들로서 지칭될 수도 있다.

[0006] 공간 또는 시간 예측은 코딩되는 블록에 대한 예측 블록을 초래한다. 잔여 데이터는 코딩되는 원래 블록과 예측 블록 사이의 픽셀 차이들을 나타낸다. 인터-코딩된 블록은 예측 블록을 형성하는 참조 샘플들의 블록을 가리키는 모션 벡터, 및 코딩된 블록과 예측 블록 사이의 차이를 나타내는 잔여 데이터에 따라서 인코딩된다. 인트라-코딩된 블록은 인트라-코딩 모드 및 잔여 데이터에 따라서 인코딩된다. 추가적인 압축을 위해, 잔여 데이터는 픽셀 도메인으로부터 변환 도메인으로 변환되어, 잔여 변환 계수들을 초래할 수도 있으며, 이 잔여 변환 계수는 그후 양자화될 수도 있다. 처음에 2차원 어레이로 배열된, 양자화된 변환 계수들은 변환 계수들의 1차원 벡터를 발생하기 위해 스캐닝될 수도 있으며, 엔트로피 코딩이 더욱 더 많은 압축을 달성하기 위해 적용될 수도 있다.

발명의 내용

해결하려는 과제

과제의 해결 수단

- [0007] 일반적으로, 본 개시물은 멀티뷰 및 3D 비디오 코딩에서 모션 예측의 코딩 효율을 향상시키기 위한 기법들을 기술한다. 예를 들어, 이 기법들은 병합/스킵 모드 및 진보된 모션 벡터 예측 (AMVP) 모드에 대한 후보 리스트에서의 후보들로서 인터-뷰 예측된 모션 벡터 후보들 및 인터-뷰 디스패리티 모션 벡터 후보들을 이용할 수도 있다. 이 기법들은 디스패리티 벡터를, 인터-예측되는 현재의 블록에 대해, 제 1 오프셋 만큼 시프트시킴으로써, 인터-뷰 예측 모션 벡터 후보를 도출하기 위한 참조 뷰에서의 참조 블록을 식별할 수도 있으며, 그리고 디스패리티 벡터를 상이한 제 2 오프셋 만큼 시프트시킴으로써 시프트된 디스패리티 벡터를 인터-뷰 디스패리티 모션 벡터 후보로 변환할 수도 있다.
- [0008] 일부 예들에서, 이 기법들은 인터-뷰 예측된 모션 벡터 후보가 이용가능하면, 후보 리스트에 인터-뷰 예측된 모션 벡터 후보만을 오직 포함시키고, 그리고 인터-뷰 예측된 모션 벡터 후보가 이용불가능하면, 후보 리스트에 인터-뷰 디스패리티 모션 벡터 후보만을 오직 포함시킬 수도 있다. 일부 예들에서, 이 기법들은 또 다른 인터-뷰 예측 모션 벡터 후보 (예컨대, 디스패리티 벡터를 시프트하지 않고 도출된 것)에 대한 모션 정보가 후보 리스트에 이미 포함되어 있으면, 인터-뷰 예측 모션 벡터 후보를, 설명 이용가능하더라도, 후보 리스트에 포함시키지 않을 수도 있다.
- [0009] 일 예에서, 본 개시물은 비디오 데이터를 코딩하는 방법을 기술하며, 본 방법은 현재의 블록이 병합 모드, 스킵 모드, 또는 진보된 모션 벡터 예측 (AMVP) 모드에서 인터-예측되는지 여부를 결정하는 단계; 현재의 블록이 AMVP 모드에서 인터-예측되면, 적어도 하나의 참조 화상 인덱스에 기초하여, 인터-뷰 디스패리티 모션 벡터 후보 (IDMVC) 또는 인터-뷰 예측된 모션 벡터 후보 (IPMVC) 를 도출할지 여부를 결정하는 단계; 및 현재의 블록이 병합 모드 또는 스킵 모드에서 인터-예측되면, IPMVC 의 이용가능성을 결정하는 단계를 포함한다. 이 예에서, IPMVC 는 현재의 블록에 대해, 제 1 오프셋 값만큼 시프트된 디스패리티 벡터에 의해 참조되는 참조 뷰에서의 참조 블록에 대한 모션 정보를 포함하며, IDMVC 는 현재의 블록에 대해, 상이한 제 2 오프셋 값만큼 시프트된 디스패리티 벡터를 포함한다. 본 방법은 또한 IPMVC 또는 IDMVC 가 후보 리스트에 포함되어 있는 것인지 여부를 결정하는 단계; 및 후보 리스트에 기초하여, 참조 뷰와는 상이한 현재의 뷰에서 현재의 화상에서의 현재의 블록을 인터-예측 코딩하는 단계를 포함한다.
- [0010] 일 예에서, 본 개시물 설명된 비디오 데이터를 코딩하는 디바이스로서, 상기 디바이스는 현재의 블록이 AMVP 모드에서 인터-예측되면, 현재의 블록이 병합 모드, 스킵 모드, 또는 진보된 모션 벡터 예측 (AMVP) 모드에서 인터-예측되는지 여부를 결정하고; 적어도 하나의 참조 화상 인덱스에 기초하여, 인터-뷰 디스패리티 모션 벡터 후보 (IDMVC) 또는 인터-뷰 예측된 모션 벡터 후보 (IPMVC) 를 도출할지 여부를 결정하고; 그리고 현재의 블록이 병합 모드 또는 스킵 모드에서 인터-예측되면, IPMVC 의 이용가능성을 결정하도록 구성된 비디오 코더를 포함한다. 이 예에서, IPMVC 는 현재의 블록에 대해, 제 1 오프셋 값만큼 시프트된 디스패리티 벡터에 의해 참조되는 참조 뷰에서의 참조 블록에 대한 모션 정보를 포함하며, IDMVC 는 현재의 블록에 대해, 상이한 제 2 오프셋 값만큼 시프트된 디스패리티 벡터를 포함한다. 비디오 코더는 또한 IPMVC 또는 IDMVC 가 후보 리스트에 포함되어 있는 것인지 여부를 결정하고; 그리고 후보 리스트에 기초하여, 참조 뷰와는 상이한 현재의 뷰에서 현재의 화상에서의 현재의 블록을 인터-예측하도록 구성된다.
- [0011] 일 예에서, 본 개시물은 실행될 때, 비디오 데이터를 코딩하는 디바이스의 하나 이상의 프로세서들로 하여금, 현재의 블록이 AMVP 모드에서 인터-예측되면, 현재의 블록이 병합 모드, 스킵 모드, 또는 진보된 모션 벡터 예측 (AMVP) 모드에서 인터-예측되는지 여부를 결정하고; 적어도 하나의 참조 화상 인덱스에 기초하여, 인터-뷰 디스패리티 모션 벡터 후보 (IDMVC) 또는 인터-뷰 예측된 모션 벡터 후보 (IPMVC) 를 도출할지 여부를 결정하고; 그리고 현재의 블록이 병합 모드 또는 스킵 모드에서 인터-예측되면, IPMVC 의 이용가능성을 결정하도록 하는 명령들을 안에 저장하고 있는 컴퓨터 판독가능 저장 매체를 기술한다. 이 예에서, IPMVC 는 현재의 블록에 대해, 제 1 오프셋 값만큼 시프트된 디스패리티 벡터에 의해 참조되는 참조 뷰에서의 참조 블록에 대한 모션 정보를 포함하며, IDMVC 는 현재의 블록에 대해, 상이한 제 2 오프셋 값만큼 시프트된 디스패리티 벡터를 포함한다. 명령들은 또한 하나 이상의 프로세서들로 하여금, IPMVC 또는 IDMVC 가 후보 리스트에 포함되어 있는 것인지 여부를 결정하고; 그리고 후보 리스트에 기초하여, 참조 뷰와는 상이한 현재의 뷰에서 현재의 화상에서의 현재의 블록을 인터-예측 코딩하도록 한다.

[0012] 일 예에서, 본 개시물은 현재의 블록이 병합 모드, 스킵 모드, 또는 진보된 모션 벡터 예측 (AMVP) 모드에서 인터-예측되는지 여부를 결정하는 수단; 현재의 블록이 AMVP 모드에서 인터-예측되면, 적어도 하나의 참조 화상 인덱스에 기초하여, 인터-뷰 디스패리티 모션 벡터 후보 (IDMVC) 또는 인터-뷰 예측된 모션 벡터 후보 (IPMVC) 를 도출할지 여부를 결정하는 수단; 및 현재의 블록이 병합 모드 또는 스킵 모드에서 인터-예측되면, IPMVC 의 이용가능성을 결정하는 수단을 포함하는, 비디오 데이터를 코딩하는 디바이스를 기술한다. 이 예에서, IPMVC 는 현재의 블록에 대해, 제 1 오프셋 값만큼 시프트된 디스패리티 벡터에 의해 참조되는 참조 뷰에서의 참조 블록에 대한 모션 정보를 포함하며, IDMVC 는 현재의 블록에 대해, 상이한 제 2 오프셋 값만큼 시프트된 디스패리티 벡터를 포함한다. 디바이스는 또한 IPMVC 또는 IDMVC 가 후보 리스트에 포함되어 있는 것인지 여부를 결정하는 수단; 및 후보 리스트에 기초하여, 참조 뷰와는 상이한 현재의 뷰에서 현재의 화상에서의 현재의 블록을 인터-예측 코딩하는 수단을 포함한다.

[0013] 하나 이상의 예들의 세부 사항들이 첨부도면 및 아래의 상세한 설명에서 개시된다. 다른 특성들, 목적들, 및 이점들은 설명 및 도면들로부터, 그리고 청구항들로부터 명백히 알 수 있을 것이다.

도면의 간단한 설명

[0014] 도 1 은 본 개시물의 인터-예측 기법들을 이용할 수도 있는 예시적인 비디오 인코딩 및 디코딩 시스템을 예시하는 블록도이다.

도 2 는 멀티-뷰 비디오에 대한 예시적인 디코딩 순서를 예시하는 개념도이다.

도 3 은 멀티-뷰 비디오에 대한 예시적인 예측 구조를 예시하는 개념도이다.

도 4 는 병합 모드 및 AMVP 모드 양쪽에서 사용될 수도 있는 예시적인 후보 블록들의 세트를 나타낸다.

도 5 는 현재의 블록에 대한 디스패리티 벡터를 결정하기 위해 후보 화상에서 체크되는 블록들의 일 예를 나타낸다.

도 6 은 인터-뷰 예측된 모션 벡터 후보의 예시적인 도출 프로세스를 예시하는 개념도이다.

도 7 은 본 개시물의 인터-예측 기법들을 구현할 수도 있는 비디오 인코더의 일 예를 예시하는 블록도이다.

도 8 은 본 개시물의 인터-예측 기법들을 구현할 수도 있는 비디오 디코더의 일 예를 예시하는 블록도이다.

도 9 는 본 개시물의 기법들에 따른 예시적인 인코딩 프로세스를 나타내는 플로우차트이다.

도 10 은 본 개시물의 기법들에 따른 예시적인 디코딩 프로세스를 나타내는 플로우차트이다.

발명을 실시하기 위한 구체적인 내용

[0015] 본 개시물은 3D HEVC 로서 일반적으로 지칭되는, 새롭게 대두하는 3차원 고효율 비디오 코딩 표준을 포함한, 멀티뷰 비디오 코딩에서 모션 파라미터들의 시그널링에 관련된 기법들을 기술한다. 비디오 압축 기법들은 이미 코딩된 비디오 데이터의 블록에 기초하여, 비디오 데이터의 현재의 블록을 예측하는 것을 포함할 수 있다.

본 개시물에서, 용어 현재의 블록은 일반적으로 이미 코딩되어 있는 비디오 블록들 또는 앞으로 코딩될 비디오 블록들과는 대조적으로, 현재 코딩중인 (예컨대, 인코딩중이거나 또는 디코딩중인) 비디오 데이터의 블록을 지칭한다. 인트라 예측 모드에서, 현재의 블록은 현재의 블록과 동일한 화상에서 하나 이상의 이전에 코딩된 블록들에 기초하여 예측되며, 반면 인터 예측 모드에서, 현재의 블록은 상이한 화상에서 이미 코딩된 블록에 기초하여 예측된다. 상이한 화상은 현재의 블록과 동일한 뷰로부터의 상이한 시간 인스턴스의 화상 (예컨대, 현재의 화상을 포함하는 액세스 유닛 이외의 또 다른 액세스 유닛에서의 화상) 일 수 있거나 또는 동일한 시간 인스턴스로부터 이지만 상이한 뷰로부터는 아닌 화상 (예컨대, 현재의 화상을 포함하는 액세스 유닛과 동일한 액세스 유닛에서의 화상) 일 수도 있다.

[0016] 비디오 디코더가 주어진 비디오 블록에 대해 인터 예측을 수행하려고 할 때, 모션 파라미터들은 인코딩된 비디오 비트스트림으로 시그널링된다. 모션 파라미터들은, 비디오 디코더가 어느 화상에서의 어느 블록을 현재의 블록에 대한 예측 블록으로서 사용할지를 결정할 수 있도록, 예를 들어, 모션 벡터, 참조 화상 인덱스, 및 예측 방향을 정의한다. 비디오 코더는 모션 파라미터들을 인코딩된 비트스트림으로 시그널링하기 위해 여러 모드들을 이용할 수도 있다. 인터-예측 코딩을 구현하는데 필요로 하는 비트수를 감소시키기 위해, 그들 모드들 중 일부는 모션 파라미터들의 명시적인 시그널링 대신, 모션 파라미터들의 예측을 이용할 수도 있다. HEVC (High Efficiency Video Coding) 표준에서, 모션 파라미터들의 예측을 위한 여러 모드들이 존재하며, 그

모션 파라미터들 중 하나는 병합 모드로서 지칭된다. 병합 모드에서, 비디오 코더는 공간 이웃하는 블록들 및 시간 이웃하는 블록들을 포함한, 그리고, 또한 3D-HEVC의 경우 인터-뷰 이웃하는 블록들을 포함한, 이웃하는 블록들로부터의 모션 파라미터들을 후보들로서 이용하여, 모션 파라미터들 (예컨대, 참조 화상들 및 모션 벡터들)의 후보 리스트를 구성한다. 선택된 모션 파라미터들은 후보 리스트로부터 그 선택된 후보의 인덱스를 송신함으로써 비디오 인코더로부터 비디오 디코더로 시그널링된다. 비디오 디코더에서, 인덱스가 디코딩되면, 선택된 후보의 대응하는 블록의 모든 모션 파라미터들이 상속된다. 비디오 인코더 및 비디오 디코더는 이미 코딩된 블록들에 기초하여 동일한 리스트들을 구성하도록 구성된다. 따라서, 인덱스에 기초하여, 비디오 디코더는 비디오 인코더에 의해 선택된 후보의 모션 파라미터들을 식별할 수 있다.

[0017] 모션 파라미터들을 예측하는 또 다른 모드는 스킵 모드 (skip mode)이다. 스킵 모드는 일반적으로 병합 모드에 대해 위에서 설명된 동일한 방법으로 동작하며, 그러나 스킵 모드에서는, 어떤 잔여 데이터도 예측된 블록에 추가되지 않으며, 반면 병합 모드에서는, 잔여 데이터가 예측된 블록에 추가된다. 병합 모드를 참조하여 위에서 설명된, 리스트의 구성 및 리스트에서 후보를 식별하는 인덱스의 송신은 일반적으로 또한 스킵 모드에서 수행된다. 일부 비디오 코덱들은 스킵 모드를 병합 모드와는 별개인 모드로서 취급할 수도 있으며, 한편 HEVC 및 그의 확장판들을 포함한, 다른 비디오 코딩 표준들은 스킵 모드를 병합 모드의 특수한 경우로서 취급할 수도 있다.

[0018] 모션 파라미터들을 예측하는 또 다른 모드는 진보된 모션 벡터 예측 (AMVP) 모드이다. AMVP 모드에서, 각각의 모션 가설에 대한 모션 벡터 예측자들의 후보 리스트는 코딩된 참조 인덱스에 기초하여 도출된다. 이 리스트는 시간 참조 화상에서 동일 장소에 배치된 블록의 이웃하는 블록의 모션 파라미터들에 기초하여 도출되는 시간 모션 벡터 예측자 뿐만 아니라, 동일한 참조 인덱스와 연관되는 이웃하는 블록들의 모션 벡터들을 포함한다. 선택된 모션 벡터들은 후보 리스트로의 인덱스를 송신함으로써 시그널링된다. 게다가, 참조 인덱스 값들 및 모션 벡터 차이들이 또한 시그널링된다. 본 개시물에서, 용어 모션 벡터 예측자는 일반적으로 하나 이상의 모션 벡터들이 이로부터 예측되는 임의의 모션 벡터를 지칭하기 위해 사용된다. 일부의 경우, 모션 벡터 예측자 및 예측된 모션 벡터는 동일할 수도 있으며, 반면, 다른 경우들에서 모션 벡터 예측자 및 예측된 모션 벡터는 상이할 수도 있다. AMVP 모드에서, 예를 들어, 예측된 모션 벡터는 모션 벡터 예측자 플러스 모션 벡터 차이 값들에 대응한다. 본 개시물은 또한 특정의 방향 (즉, 특정의 참조 화상 리스트)과 연관되는 모션 벡터 예측자를 일반적으로 지칭하는 용어 방향 모션 예측자를 참조한다. 양방향-예측의 경우, 모션 벡터 예측자는 2개의 방향 모션 예측자들을 포함할 수도 있다.

[0019] 디스패리티 벡터는 일반적으로 현재의 뷰의 현재의 블록에 대응하는 인터-뷰 참조 화상에서의 블록의 로케이션을 식별하는데 사용된다. 현재의 블록에 대한 디스패리티 벡터 (DV)를 도출하기 위해서, 이웃하는 블록들 기반의 디스패리티 벡터 (NBDV) 도출로 불리는 기법이 3D-HEVC 테스트 모델 (3D-HTM)의 일부 현재의 버전들에서 사용된다. NBDV는 현재의 블록의 공간 및 시간 이웃하는 블록들로부터의 디스패리티 모션 벡터들을 이용한다. NBDV에서, 공간 또는 시간 이웃하는 블록들의 디스패리티 모션 벡터들은 고정된 순서로 체크된다. 일단 디스패리티 모션 벡터 (즉, 인터-뷰 참조 화상을 가리키는 모션 벡터)가 식별되면, 체크 프로세스가 종료되며, 식별된 디스패리티 모션 벡터가 현재의 블록에 대한 DV로 변환된다. DV는 인터-뷰 모션 예측 및/또는 인터-뷰 잔여 예측에 사용될 수 있다. 어떤 디스패리티 모션 벡터도 모든 사전-정의된 이웃하는 블록들을 체크한 후에 발견되지 않으면, 제로 DV가 인터-뷰 모션 예측에 사용될 수도 있지만 인터-뷰 잔여 예측은 대응하는 예측 유닛 (PU)에 대해 사용불능될 수도 있다.

[0020] 일부 예들에서, 현재의 블록에 대한 디스패리티 벡터는 이웃하는 블록의 디스패리티 벡터로부터 도출될 수도 있다. 예를 들어, 이웃하는 블록 (예컨대, 공간 또는 시간 이웃하는 블록)은 디스패리티 모션 벡터로 인터-예측되지 않았을 수도 있다. 그러나, 이웃하는 블록의 인코딩 또는 디코딩 동안, 그 이웃하는 블록에 대한 디스패리티 벡터가 도출되었을 수도 있다. 이 이웃하는 블록의 도출된 디스패리티 벡터가 현재의 블록에 대한 디스패리티 벡터로서 사용될 수도 있다. 예를 들어, 어느 평가되는 이웃하는 블록들도 디스패리티 모션 벡터로 인터-예측되지 않지만 도출된 디스패리티 벡터가 이웃하는 블록에 존재하면, 비디오 인코더 및 비디오 디코더는 이웃하는 블록의 도출된 디스패리티 벡터를 현재의 블록에 대한 디스패리티 벡터로서 설정할 수도 있다.

[0021] DV에 기초하여, 인터-뷰 예측된 모션 벡터로서 지칭되는, 새로운 모션 벡터 후보가, 이용가능하면, AMVP 및 스킵/병합 모드들에 대한 후보 리스트들에 추가될 수도 있다. 인터-뷰 예측된 모션 벡터는, 이용가능하면, 시간 모션 벡터이다. 일반적으로, 본 개시물은 멀티뷰 및 3D 비디오 코딩에서 모션 예측의 코딩 효율을 향상

시키기 위한 기법들을 기술한다.

[0022] 도 1 은 본 개시물의 기법들을 이용할 수도 있는 예시적인 비디오 인코딩 및 디코딩 시스템 (10) 을 예시하는 블록도이다. 도 1 에 나타난 바와 같이, 시스템 (10) 은 목적지 디바이스 (14) 에 의해 추후에 디코딩되는 인코딩된 비디오 데이터를 제공하는 소스 디바이스 (12) 를 포함한다. 특히, 소스 디바이스 (12) 는 비디오 데이터를 목적지 디바이스 (14) 에 컴퓨터 판독가능 매체 (16) 를 통해서 제공한다. 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 데스크탑 컴퓨터들, 노트북 (즉, 랩탑) 컴퓨터들, 태블릿 컴퓨터들, 셋-탑 박스들, 소위 "스마트" 폰들과 같은 전화기 핸드셋들, 소위 "스마트" 패드들, 텔레비전, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 콘솔들, 비디오 스트리밍 디바이스 등을 포함한, 광범위한 디바이스들 중 임의의 디바이스를 포함할 수도 있다. 일부의 경우, 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 무선 통신용으로 탑재될 수도 있다.

[0023] 목적지 디바이스 (14) 는 디코딩되는 인코딩된 비디오 데이터를 컴퓨터 판독가능 매체 (16) 를 통해서 수신할 수도 있다. 컴퓨터 판독가능 매체 (16) 는 인코딩된 비디오 데이터를 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 이동시킬 수 있는 임의 종류의 매체 또는 디바이스를 포함할 수도 있다. 일 예에서, 컴퓨터 판독가능 매체 (16) 는 소스 디바이스 (12) 로 하여금 인코딩된 비디오 데이터를 직접 목적지 디바이스 (14) 로 실시간으로 송신할 수 있게 하는 통신 매체를 포함할 수도 있다. 인코딩된 비디오 데이터는 무선 통신 프로토콜과 같은 통신 표준에 따라서 변조되어 목적지 디바이스 (14) 로 송신될 수도 있다. 통신 매체는 무선 주파수 (RF) 스펙트럼 또는 하나 이상의 물리적인 송신 라인들과 같은, 임의의 무선 또는 유선 통신 매체를 포함할 수도 있다. 통신 매체는 근거리 네트워크, 광역 네트워크, 또는 글로벌 네트워크, 예컨대 인터넷과 같은 패킷-기반 네트워크의 일부를 형성할 수도 있다. 통신 매체는 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 통신을 용이하게 하는데 유용할 수도 있는 임의의 다른 장비를 포함할 수도 있다.

[0024] 일부 예들에서, 인코딩된 데이터는 출력 인터페이스 (22) 로부터 저장 디바이스로 출력될 수도 있다. 이와 유사하게, 인코딩된 데이터는 입력 인터페이스 (28) 에 의해 저장 디바이스로부터 액세스될 수도 있다. 저장 디바이스는 하드 드라이브, Blu-ray 디스크들, DVD들, CD-ROM들, 플래시 메모리, 휘발성 또는 비-휘발성 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 임의의 다른 적합한 디지털 저장 매체들과 같은 다양한 분산된 또는 로컬 액세스되는 데이터 저장 매체들 중 임의의 데이터 저장 매체를 포함할 수도 있다. 추가 예에서, 저장 디바이스는 소스 디바이스 (12) 에 의해 발생된 인코딩된 비디오를 저장할 수도 있는 파일 서버 또는 또 다른 중간 저장 디바이스에 대응할 수도 있다. 목적지 디바이스 (14) 는 저장된 비디오 데이터에 저장 디바이스로부터 스트리밍 또는 다운로드를 통해서 액세스할 수도 있다. 파일 서버는 인코딩된 비디오 데이터를 저장하고 그 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로 송신하는 것이 가능한 임의 종류의 서버일 수도 있다. 예시적인 파일 서버들은 웹 서버 (예컨대, 웹사이트용), FTP 서버, NAS (network attached storage) 디바이스들, 또는 로컬 디스크 드라이브를 포함한다. 목적지 디바이스 (14) 는 인터넷 접속을 포함한, 임의의 표준 데이터 접속을 통해서, 인코딩된 비디오 데이터에 액세스할 수도 있다. 이것은 파일 서버 상에 저장된 인코딩된 비디오 데이터에 액세스하는데 적합한, 무선 채널 (예컨대, Wi-Fi 접속), 유선 접속 (예컨대, DSL, 케이블 모뎀 등), 또는 양쪽의 조합을 포함할 수도 있다. 저장 디바이스로부터의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 이들의 조합일 수도 있다.

[0025] 본 개시물의 기법들은 무선 애플리케이션들 또는 설정들에 반드시 한정되지는 않는다. 이 기법들은 오버-디-에어 텔레비전 브로드캐스트들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, 인터넷 스트리밍 비디오 송신들, 예컨대 HTTP 를 통한 동적 적응 스트리밍 (DASH), 데이터 저장 매체 상에 인코딩된 디지털 비디오, 데이터 저장 매체 상에 저장된 디지털 비디오의 디코딩, 또는 다른 애플리케이션들과 같은, 다양한 멀티미디어 애플리케이션들 중 임의의 애플리케이션의 지원 하에 비디오 코딩에 적용될 수도 있다. 일부 예들에서, 시스템 (10) 은 비디오 스트리밍, 비디오 플레이어, 비디오 브로드캐스팅, 및/또는 비디오 전화 통신과 같은, 지원 애플리케이션들로의 1-방향 또는 2-방향 비디오 송신을 지원하도록 구성될 수도 있다.

[0026] 도 1 의 예에서, 소스 디바이스 (12) 는 비디오 소스 (18), 심도 추정 유닛 (19), 비디오 인코더 (20), 및 출력 인터페이스 (22) 를 포함한다. 목적지 디바이스 (14) 는 입력 인터페이스 (28), 비디오 디코더 (30), 심도 이미지 기반 렌더링 (DIBR) 유닛 (31), 및 디스플레이 디바이스 (32) 를 포함한다. 다른 예들에서, 소스 디바이스 및 목적지 디바이스는 다른 구성요소들 또는 배열들을 포함할 수도 있다. 예를 들어, 소스 디바이스 (12) 는 비디오 데이터를 외부 카메라와 같은 외부 비디오 소스 (18) 로부터 수신할 수도 있다. 이와 유사하게, 목적지 디바이스 (14) 는 통합된 디스플레이 디바이스를 포함하는 대신, 외부 디스플레이 디바이스와 인

터페이스할 수도 있다. 본 개시물에서 설명하는 기법들 중 많은 기법들은 비디오 인코더 (20) 및 비디오 디코더 (30) 양쪽에 의해 수행될 수 있다. 따라서, 설명의 용이성을 위해, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 비디오 코더 (20/30) 로서 공통적으로 종종 지칭될 수도 있다. 비디오 코더 (20/30) 와 관련하여 설명되는 기법들은 비디오 인코더 (20) 또는 비디오 디코더 (30) 에 의해 수행될 수도 있다.

[0027] 도 1 의 예시된 시스템 (10) 은 단지 일 예이다. 본 개시물의 기법들은 임의의 디지털 비디오 인코딩 및/또는 디코딩 디바이스에 의해 수행될 수도 있다. 일반적으로 본 개시물의 기법들은 비디오 인코딩 디바이스에 의해 수행되지만, 그 기법들은 또한 "코덱" 으로서 일반적으로 지칭되는, 비디오 인코더/디코더에 의해 수행될 수도 있다. 더욱이, 본 개시물의 기법들은 또한 비디오 프리프로세서에 의해 수행될 수도 있다. 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 단지 코딩 디바이스들의 예들이며, 여기서, 소스 디바이스 (12) 는 목적지 디바이스 (14) 로의 송신을 위해 코딩된 비디오 데이터를 발생한다. 일부 예들에서, 디바이스들 (12, 14) 은 디바이스들 (12, 14) 의 각각이 비디오 인코딩 및 디코딩 구성요소들을 포함하도록, 실질적으로 대칭적 방식으로 동작할 수도 있다. 그러므로, 시스템 (10) 은 예컨대, 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅, 또는 비디오 전화 통신을 위해, 비디오 디바이스들 (12, 14) 사이에 1-방향 또는 2-방향 비디오 송신을 지원할 수도 있다.

[0028] 소스 디바이스 (12) 의 비디오 소스 (18) 는 비디오 카메라와 같은 비디오 캡처 디바이스, 이전에 캡처된 비디오를 포함하는 비디오 아카이브, 및/또는 비디오 콘텐츠 제공자로부터 비디오를 수신하는 비디오 공급 인터페이스를 포함할 수도 있다. 추가 대안적인 예로서, 비디오 소스 (18) 는 컴퓨터 그래픽스-기반의 데이터를 소스 비디오, 또는 라이브 비디오, 아카이브된 비디오, 및 컴퓨터 발생된 비디오의 조합으로서 발생할 수도 있다. 어떤 경우, 비디오 소스 (18) 가 비디오 카메라이면, 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 소위 카메라 폰들 또는 비디오 폰들을 형성할 수도 있다. 그러나, 위에서 언급한 바와 같이, 본 개시물에서 설명하는 기법들은 비디오 코딩에 일반적으로 적용가능할 수도 있으며, 무선 및/또는 유선 애플리케이션들에 적용될 수도 있다. 각 경우, 캡처되거나, 사전-캡처되거나, 또는 컴퓨터-발생된 비디오는 비디오 인코더 (20) 에 의해 인코딩될 수도 있다. 인코딩된 비디오 정보는 그후 출력 인터페이스 (22) 에 의해 컴퓨터 판독가능 매체 (16) 상으로 출력될 수도 있다.

[0029] 비디오 소스 (18) 는 비디오 데이터의 다수의 뷰들을 비디오 인코더 (20) 에 제공할 수도 있다. 예를 들어, 비디오 소스 (18) 는 카메라들의 어레이에 대응할 수도 있으며, 각각의 카메라는 촬영되는 특정의 장면에 대해 고유한 수평 위치를 갖는다. 이의 대안으로, 비디오 소스 (18) 는 이종의 (disparate) 수평 카메라 원근법들로부터, 예컨대, 컴퓨터 그래픽스를 이용하여, 비디오 데이터를 발생할 수도 있다. 심도 추정 유닛 (19) 은 텍스처 이미지에서 픽셀들에 대응하는 심도 픽셀들에 대한 값들을 결정하도록 구성될 수도 있다. 예를 들어, 심도 추정 유닛 (19) 은 소나 (SONAR; Sound Navigation and Ranging) 유닛, 라이다 (LIDAR; Light Detection and Ranging) 유닛, 또는 심도 값들을 직접 결정하면서 실질적으로 동시에 장면의 비디오 데이터를 리코딩하는 것이 가능한 다른 유닛을 나타낼 수도 있다.

[0030] 이에 추가적으로 또는 대안적으로, 심도 추정 유닛 (19) 은 상이한 수평 카메라 원근법들로부터 실질적으로 동일한 시간에서 캡처된 2개 이상의 이미지들을 비교함으로써 심도 값들을 간접적으로 계산하도록 구성될 수도 있다. 이미지들에서 실질적으로 유사한 픽셀 값들 사이의 수평 디스패리티를 계산함으로써, 심도 추정 유닛 (19) 은 장면에서 여러 오브젝트들의 심도를 근사화할 수도 있다. 일부 예들에서, 심도 추정 유닛 (19) 은 비디오 소스 (18) 와 기능적으로 통합될 수도 있다. 예를 들어, 비디오 소스 (18) 가 컴퓨터 그래픽스 이미지들을 발생할 때, 심도 추정 유닛 (19) 은 예컨대, 텍스처 이미지들을 렌더링하는데 사용되는 픽셀들 및 오브젝트들의 z-좌표들을 이용하여, 그래픽 오브젝트들에 실제 심도 맵들을 제공할 수도 있다.

[0031] 컴퓨터 판독가능 매체 (16) 는 무선 브로드캐스트 또는 유선 네트워크 송신과 같은 일시성 매체, 또는 하드 디스크, 플래시 드라이브, 콤팩트 디스크, 디지털 비디오 디스크, Blu-ray 디스크, 또는 다른 컴퓨터 판독가능 매체들과 같은 저장 매체들 (즉, 비일시성 저장 매체들) 을 포함할 수도 있다. 일부 예들에서, 네트워크 서버 (미도시) 는 인코딩된 비디오 데이터를 소스 디바이스 (12) 로부터 수신하고 그 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로, 예컨대, 네트워크 송신을 통해서 제공할 수도 있다. 이와 유사하게, 디스크 스탬핑 설비와 같은 매체 생산 설비의 컴퓨팅 디바이스는 인코딩된 비디오 데이터를 소스 디바이스 (12) 로부터 수신하고 그 인코딩된 비디오 데이터를 포함하는 디스크를 제조할 수도 있다. 따라서, 컴퓨터 판독가능 매체 (16) 는 여러 예들에서, 여러 형태들의 하나 이상의 컴퓨터 판독가능 매체들을 포함하는 것으로 이해될 수도 있다.

[0032] 목적지 디바이스 (14) 의 입력 인터페이스 (28) 는 컴퓨터 판독가능 매체 (16) 로부터 정보를 수신한다. 컴

퓨터 관독가능 매체 (16) 의 정보는 블록들 및 다른 코딩된 유닛들, 예컨대, 화상들의 그룹 (GOPs) 의 특성들 및/또는 프로세싱을 기술하는 구문 엘리먼트들을 포함하는, 비디오 인코더 (20) 에 의해 정의되고 또한 비디오 디코더 (30) 에 의해 사용되는 구문 정보를 포함할 수도 있다. 디스플레이 디바이스 (32) 는 그 디코딩된 비디오 데이터를 사용자에게 디스플레이하며, 음극선관 (CRT), 액정 디스플레이 (LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (OLED) 디스플레이, 또는 또 다른 유형의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들 중 임의의 디바이스를 포함할 수도 있다. 일부 예들에서, 디스플레이 디바이스 (32) 는 예컨대, 뷰어에 대한 3D 시각 효과를 발생하기 위해 2개의 이상의 뷰들을 동시에 또는 실질적으로 동시에 디스플레이하는 것이 가능한 디바이스를 포함할 수도 있다.

[0033] 목적지 디바이스 (14) 의 DIBR 유닛 (31) 은 비디오 디코더 (30) 로부터 수신된 디코딩된 뷰들의 텍스처 및 심도 정보를 이용하여, 합성된 뷰들을 렌더링할 수도 있다. 예를 들어, DIBR 유닛 (31) 은 텍스처 이미지들의 픽셀 데이터에 대한 수평 디스패리티를 대응하는 심도 맵들에서 픽셀들의 값들의 함수로서 결정할 수도 있다.

DIBR 유닛 (31) 은 그 후 그 결정된 수평 디스패리티 만큼 텍스처 이미지에서의 픽셀들을 좌측 또는 우측으로 오프셋시킴으로써, 합성된 이미지를 발생할 수도 있다. 이러한 방법으로, 디스플레이 디바이스 (32) 는 디코딩된 뷰들 및/또는 합성된 뷰들에 대응할 수도 있는 하나 이상의 뷰들을, 임의의 조합으로 디스플레이할 수도 있다. 본 개시물의 기법들에 따르면, 비디오 디코더 (30) 는 심도 범위들 및 카메라 파라미터들에 대한 원래 및 업데이트된 정밀도 값들을 DIBR 유닛 (31) 에 제공할 수도 있으며, 그 DIBR 유닛은 심도 범위들 및 카메라 파라미터들을 이용하여 뷰들을 순서대로 합성할 수도 있다.

[0034] 도 1 에 나타내지는 않지만, 일부 양태들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 오디오 인코더 및 디코더와 각각 통합될 수도 있으며, 오디오 및 비디오 양쪽의 인코딩을 공통 데이터 스트림 또는 별개의 데이터 스트림들로 처리하기에 적합한 MUX-DEMUX 유닛들, 또는 다른 하드웨어 및 소프트웨어를 포함할 수도 있다. 적용가능한 경우, MUX-DEMUX 유닛들은 ITU H.223 멀티플렉서 프로토콜, 또는 다른 프로토콜들, 예컨대, 사용자 데이터그램 프로토콜 (UDP) 을 따를 수도 있다.

[0035] 비디오 인코더 (20) 및 비디오 디코더 (30) 는 HEVC (High Efficiency Video Coding) 표준과 같은 비디오 코딩 표준에 따라서 동작할 수도 있으며, HEVC 테스트 모델 (HM) 에 따를 수도 있다. 차기 HEVC 표준의 최신 안은 ITU-T SG16 WP3 와 ISO/IEC JTC1/SC29/WG11 의 JCT-VC (Joint Collaborative Team on Video Coding), 10차 회의: 2012년 7월 11일 내지 2012년 7월 12일, 스웨덴, 스톡홀름, 문서 HCTVC-J1003, Bross 등의, "High Efficiency Video Coding (HEVC) Text Specification Draft 8", 에 설명되어 있으며, 이것은 2013년 12월 4일 현재, http://phenix.int-evry.fr/jct/doc_end_user/documents/10_Stockholm/wg11/JCTVC-J1003-v8.zip 로부터 다운로드가능하다. "HEVC Working Draft 10" 또는 "WD10" 로서 지칭되는, HEVC 표준의 또 다른 최신 안은 ITU-T SG16 WP3 와 ISO/IEC JTC1/SC29/WG11 의 JCT-VC (Joint Collaborative Team on Video Coding), 12차 회의: 2013년 1월 14-23 일, 스위스, 제네바, 문서 JCTVC-L1003v34, Bross 등의, "High Efficiency Video Coding (HEVC) text specification draft 10 (for FDIS & Last Call)" 에 설명되어 있으며, 이것은 2013년 12월 4일 현재, http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v34.zip 로부터 다운로드가능하다. 여전히 HEVC 표준의 또 다른 초안은 본원에서, ITU-T SG16 WP3 와 ISO/IEC JTC1/SC29/WG11 의 JCT-VC (Joint Collaborative Team on Video Coding), 13차 회의, 2013년 4월, 한국, 인천, Bross 등의, "Editors' proposed corrections to HEVC version 1" 에 설명되어 있는 "WD10 revisions" 으로서 지칭되며, 이것은 2013년 12월 4일 현재, http://phenix.int-evry.fr/jct/doc_end_user/documents/13_Incheon/wg11/JCTVC-M0432-v3.zip 로부터 입수가가능하다.

[0036] 예시의 목적을 위해, 본 개시물의 기법들은 HEVC 의 3DV 확장판에 대해 주로 설명된다. 그러나, 이들 기법들은 3차원 효과를 발생하는데 사용되는 비디오 데이터를 코딩하는 다른 표준들에도 역시 적용될 수도 있는 것으로 이해되어야 한다.

[0037] VCEG 및 MPEG 의 3D 비디오 코딩 (JCT-3C) 에 관한 합동 연구팀은 HEVC 에 기초한 3DV 표준을 개발하고 있으며, 표준화 노력들의 부분은 HEVC (MV-HEVC) 에 기초한 멀티뷰 비디오 코덱의 표준화 및 HEVC (3D-HEVC) 에 기초한 3D 비디오 코딩을 위한 또 다른 부분을 포함한다. MV-HEVC 에 대해, HEVC 에서의 CU/PU 레벨에서 어떤 모듈도 재설계될 필요가 없고 MV-HEVC 에 충분히 재사용될 수 있도록, 단지 하이-레벨 구문 (HLS) 이 HEVC 에 대해 변경되는 것이 일반적으로 요망된다. 3D-HEVC 에 대해, 텍스처 및 심도 뷰들 양쪽에 대해, 유닛/예측 유닛 레벨에서 코딩할 때의 코딩 툴들을 포함한, 새로운 코딩 툴들이 포함되고 지원될 수도 있다. 3D-HEVC 에 대한 최신 소프트웨어 3D-HTM 는 다음 링크: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/trunk/ 또는 https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSsoftware/tags/HTM-5.0/ 로부터 2013년 12월 4일 현재, 다운로드

될 수 있다.

- [0038] 이의 대안으로, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 다른 독점 또는 산업 표준들, 예컨대, MPEG 4, 파트 10, AVC (Advanced Video Coding) 로서 대안적으로 지칭되는 ITU-T H.264 표준, 또는 이러한 표준들의 확장판들, 예컨대, ITU-T H.264/AVC 의 MVC 확장판에 따라서 동작할 수도 있다. 특히, 본 개시물의 기법들은 진보된 코덱들에 기초한 멀티뷰 및/또는 3D 비디오 코딩에 관련된다. 일반적으로, 본 개시물의 기법들은 다양한 상이한 비디오 코딩 표준들 중 임의의 표준에 적용될 수도 있다. 예를 들어, 이들 기법들은 ITU-T H.264/AVC (Advanced Video Coding) 의 멀티-뷰 비디오 코딩 (MVC) 확장판에, 차기 HEVC 표준 (예컨대, 3D-HEVC) 의 3D 비디오 (3DV) 확장판에, 또는 다른 코딩 표준에 적용될 수도 있다.
- [0039] ITU-T H.264/MPEG-4 (AVC) 표준은 ISO/IEC 동화상 전문가 그룹 (MPEG) 과 함께, ITU-T 비디오 코딩 전문가 그룹 (VCEG) 에 의해 조인트 비디오 팀 (JVT) 으로서 알려진 공동 파트너십의 성과로서, 정식화되었다. 일부 양태들에서, 본 개시물에서 설명하는 기법들은 H.264 표준에 일반적으로 따르는 디바이스들에 적용될 수도 있다. H.264 표준은 2005년 3월, ITU-T 스터디 그룹에 의한, ITU-T 권고안 H.264, Advanced Video Coding for generic audiovisual services 에 설명되어 있으며, 본원에서 H.264 표준 또는 H.264 사양, 또는 H.264/AVC 표준 또는 사양으로서 지칭될 수도 있다. 조인트 비디오 팀 (JVT) 은 H.264/MPEG-4 AVC 에 대한 확장판들에 대해 계속 노력을 기울이고 있다.
- [0040] 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들 (DSPs), 주문형 집적회로들 (ASICs), 필드 프로그래밍가능 게이트 어레이들 (FPGAs), 이산 로직, 소프트웨어, 하드웨어, 펌웨어 또는 임의의 이들의 조합들과 같은, 다양한 적합한 인코더 회로 중 임의의 회로로 구현될 수도 있다. 이 기법들이 소프트웨어로 부분적으로 구현되는 경우, 디바이스는 본 개시물의 기법들을 수행하기 위해 소프트웨어용 명령들을 적합한 비일시성 컴퓨터 판독가능 매체에 저장하고, 그 명령들을 하드웨어에서 하나 이상의 프로세서들을 이용하여 실행할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 인코더들 또는 디코더들에 포함될 수도 있으며, 이들 중 어느 쪽이든 각각의 디바이스에서 결합된 인코더/디코더 (CODEC) 의 일부로서 통합될 수도 있다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 를 포함하는 디바이스는 집적 회로, 마이크로프로세서, 및/또는 무선 통신 디바이스, 예컨대 셀룰러 전화기를 포함할 수도 있다.
- [0041] 먼저, 예시적인 HEVC 의 코딩 기법들이 설명될 것이다. JCT-VC 는 HEVC 표준을 개발하였다. HEVC 표준화 노력들은 HEVC 테스트 모델 (HM) 로서 지칭되는 비디오 코딩 디바이스의 진화 모델 (evolving model) 에 기초한다. HM 은 예컨대, ITU-T H.264/AVC 에 따른 기존 디바이스들에 관련된 비디오 코딩 디바이스들의 여러 추가적인 능력들을 가정한다. 예를 들어, H.264 는 9개의 인트라-예측 인코딩 모드들을 제공하는 반면, HM 은 33개 만큼이나 많은 각도 인트라-예측 인코딩 모드들에 더하여 DC 및 평면 모드들을 제공할 수도 있다.
- [0042] 일반적으로, HM 의 작업 모델은 비디오 프레임 또는 화상이 루마 샘플 및 크로마 샘플들 양쪽을 포함하는 트리블록들 또는 최대 코딩 유닛들 (LCU) 의 시퀀스로 분할될 수도 있다고 기술한다. 비트스트림 내 구문 데이터는 LCU 에 대한 사이즈를 정의할 수도 있으며, 이 최대 코딩 유닛은 픽셀들의 개수의 관점에서 최대 코딩 유닛이다. 슬라이스는 코딩 순서에서 다수의 연속되는 트리블록들을 포함한다. 비디오 프레임 또는 화상은 하나 이상의 슬라이스들로 파티셔닝될 수도 있다. 각각의 트리블록은 쿼드트리에 따라 코딩 유닛들 (CUs) 로 분할될 수도 있다. 일반적으로, 쿼드트리 데이터 구조는 CU 당 하나의 노드를 포함하며, 동시에, 루트 노드는 트리블록에 대응한다. CU 가 4개의 서브-CU들로 분할되면, CU 에 대응하는 노드는 4개의 리프 노드들을 포함하며, 그 리프 노드 각각은 서브-CU들 중 하나에 대응한다.
- [0043] 쿼드트리 데이터 구조의 각각의 노드는 대응하는 CU 에 대한 구문 데이터를 제공할 수도 있다. 예를 들어, 쿼드트리에서 노드는 그 노드에 대응하는 CU 가 서브-CU들로 분할되는지의 여부를 나타내는 분할 플래그를 포함할 수도 있다. CU 에 대한 구문 엘리먼트들은 회귀적으로 정의될 수도 있으며, CU 가 서브-CU들로 분할되는지의 여부에 의존할 수도 있다. CU 가 추가로 분할되지 않으면, 리프-CU 로서 지칭된다. 본 개시물에서, 리프-CU 의 4개의 서브-CU들은 또한 원래 리프-CU 의 명시적인 분할이 없더라도 리프-CU들로 지칭될 것이다. 예를 들어, 16x16 사이즈에서 CU 가 추가로 분할되지 않으면, 4개의 8x8 서브-CU들이 또한 16x16 CU 가 전혀 분할되지 않았더라도 리프-CU들로서 지칭될 것이다.
- [0044] CU 는 CU 가 사이즈 구별을 갖지 않는다는 점을 제외하고는, H.264 표준의 매크로블록과 유사한 목적을 갖는다. 예를 들어, 트리블록은 4개의 자식 노드들 (또한, 서브-CU들로서 지칭됨) 로 분할될 수도 있으며, 각각의 자식 노드는 결국 부모 노드일 수도 있으며 또 다른 4개의 자식 노드들로 분할될 수도 있다. 쿼드트리의 리프

노드로서 지칭되는, 최종, 미분할된 자식 노드는 리프-CU 로서 또한 지칭되는, 코딩 노드를 포함한다. 코딩된 비트스트림과 연관되는 구문 데이터는 최대 CU 깊이로서 지칭되는, 트리블록이 분할될 수도 있는 최대 횡수를 정의할 수도 있으며, 또한 코딩 노드들의 최소 사이즈를 정의할 수도 있다. 따라서, 비트스트림은 또한 최소 코딩 유닛 (SCU) 을 정의할 수도 있다. 본 개시물은 HEVC 의 상황에서, CU, PU, 또는 TU 중 임의의 것, 또는 다른 표준들의 상황에서 유사한 데이터 구조들 (예컨대, H.264/AVC 에서의 매크로블록들 및 그의 서브-블록들) 을 지칭하는데 용어 "블록" 을 사용한다.

[0045] CU 는 코딩 노드, 및 이 코딩 노드와 연관되는 변환 유닛들 (TUs) 및 예측 유닛들 (PUs) 을 포함한다. CU 의 사이즈는 코딩 노드의 사이즈에 대응하며 정사각형 형태이어야 한다. CU 의 사이즈는 8x8 픽셀들로부터 64x64 픽셀들 이상의 최대치를 갖는 트리블록의 사이즈까지 이를 수도 있다. 각각의 CU 는 하나 이상의 PUs 및 하나 이상의 TUs 을 포함할 수도 있다. CU 와 연관되는 구문 데이터는 예를 들어, 하나 이상의 PUs 로의 CU 의 파티셔닝을 기술할 수도 있다. 파티셔닝 모드들은 CU 가 스킵 또는 병합 모드 인코딩되는지, 인트라-예측 모드 인코딩되는지, 또는 인터-예측 모드 인코딩되는지 여부 사이에 상이할 수도 있다. PUs 은 비-정사각형의 형태로 파티셔닝될 수도 있다. CU 와 연관되는 구문 데이터는 또한 예를 들어, 쿼드트리에 따른 하나 이상의 TUs 로의 CU 의 파티셔닝을 기술할 수도 있다. TU 는 정사각형 또는 비-정사각형 (예컨대, 직사각형) 의 형태일 수 있다.

[0046] HEVC 표준은 TUs 에 따라서 변환들을 허용하며, 이 TUs 는 상이한 CUs 에 대해 상이할 수도 있다. TUs 는 일반적으로 파티셔닝된 LCU 에 대해 정의된 주어진 CU 내 PUs 의 사이즈에 기초하여 사이징되지만, 이것이 항상 그런 것은 아니다. TUs 는 일반적으로 PUs 과 동일한 사이즈이거나 또는 그보다 작다. 일부 예들에서, CU 에 대응하는 잔여 샘플들은 "잔여 쿼드 트리" (RQT) 로서 알려진 쿼드트리 구조를 이용하여 더 작은 유닛들로 세분될 수도 있다. RQT 의 리프 노드들은 변환 유닛들 (TUs) 로서 지칭될 수도 있다. TU들과 연관되는 픽셀 차이 값들은 변환 계수들을 발생하기 위해 변환될 수도 있으며, 그 변환 계수들은 양자화될 수도 있다.

[0047] 리프-CU 는 하나 이상의 예측 유닛들 (PUs) 을 포함할 수도 있다. 일반적으로, PU 는 대응하는 CU 의 모두 또는 부분에 대응하는 공간 영역을 나타내며, PU 에 대한 참조 샘플을 추출하기 위한 데이터를 포함할 수도 있다. 더욱이, PU 는 예측에 관련된 데이터를 포함한다. 예를 들어, PU 가 인트라-모드 인코딩될 때, PU 에 대한 데이터는 잔여 쿼드트리 (RQT) 에 포함될 수도 있으며, PU 에 대응하는 TU 에 대한 인트라-예측 모드를 기술하는 데이터를 포함할 수도 있다. 또 다른 예로서, PU 가 인터-모드 인코딩될 때, PU 는 PU 에 대한 하나 이상의 모션 벡터들을 정의하는 데이터를 포함할 수도 있다. PU 에 대한 모션 벡터를 정의하는 데이터는 예를 들어, 모션 벡터의 수평 성분, 모션 벡터의 수직 성분, 모션 벡터에 대한 해상도 (예컨대, 1/4 픽셀 정밀도 또는 1/8 픽셀 정밀도), 모션 벡터가 가리키는 참조 화상, 및/또는 모션 벡터에 대한 참조 화상 리스트 (예컨대, 리스트 0, 리스트 1, 또는 리스트 C) 를 기술할 수도 있다.

[0048] 하나 이상의 PUs 을 갖는 리프-CU 는 또한 하나 이상의 변환 유닛들 (TUs) 을 포함할 수도 있다. 변환 유닛들은 위에서 설명한 바와 같이, RQT (또한, TU 쿼드트리 구조로서 지칭됨) 를 이용하여 규정될 수도 있다. 예를 들어, 분할 플래그는 리프-CU 가 4개의 변환 유닛들로 분할되는지 여부를 나타낼 수도 있다. 그 후, 각각의 변환 유닛은 추가적인 서브-TU들로 추가로 분할될 수도 있다. TU 가 추가로 분할되지 않을 때, 리프-CU 로서 지칭될 수도 있다. 일반적으로, 인트라 코딩에 있어, 리프-CU 에 속하는 모든 리프-TU들은 동일한 인트라 예측 모드를 공유한다. 즉, 동일한 인트라-예측 모드가 일반적으로 리프-CU 의 모든 TU들에 대해 예측된 값들을 계산하기 위해 적용된다. 인트라 코딩에 있어, 비디오 인코더는 각각 리프-TU 에 대한 잔여 값을 인트라 예측 모드를 이용하여, TU 에 대응하는 CU 의 부분과 원래 블록 사이의 차이로서 계산할 수도 있다. TU 는 PU 의 사이즈에 반드시 제한되지는 않는다. 따라서, TU들은 PU 보다 더 크거나 또는 더 작을 수도 있다. 인트라 코딩에 있어, PU 는 동일한 CU 에 대한 대응하는 리프-TU 와 콜로케이션될 (collocate) 수도 있다. 일부 예들에서, 리프-TU 의 최대 사이즈는 대응하는 리프-CU 의 사이즈에 대응할 수도 있다.

[0049] 더욱이, 리프-CU들의 TU들은 또한 잔여 쿼드트리들 (RQTs) 로서 지칭되는, 각각의 쿼드트리 데이터 구조들과 연관될 수도 있다. 즉, 리프-CU 는 리프-CU 가 어떻게 TU들로 파티셔닝되는 지를 나타내는 쿼드트리를 포함할 수도 있다. TU 쿼드트리의 루트 노드는 일반적으로 리프-CU 에 대응하는 반면, CU 쿼드트리의 루트 노드는 일반적으로 트리블록 (또는, LCU) 에 대응한다. 분할되지 않은 RQT 의 TU들은 리프-TUs 로서 지칭된다. 일반적으로, 본 개시물은 달리 언급하지 않는 한, 리프-CU 및 리프-TU 를 지칭하기 위해, 각각 용어들 CU 및 TU 를 사용한다.

[0050] 비디오 시퀀스는 일반적으로 비디오 프레임들 또는 화상들의 시리즈를 포함한다. 화상들의 그룹 (GOP) 은

일반적으로 비디오 화상들의 하나 이상의 시리즈를 포함한다. GOP 는 GOP 의 헤더, 화상들의 하나 이상의 헤더, 또는 다른 곳에, GOP 에 포함된 다수의 화상들을 기술하는 구문 데이터를 포함할 수도 있다. 화상의 각각의 슬라이스는 각각의 슬라이스에 대한 인코딩 모드를 기술하는 슬라이스 구문 데이터를 포함할 수도 있다.

비디오 인코더 (20) 는 일반적으로 비디오 데이터를 인코딩하기 위해 개개의 비디오 슬라이스들 내 비디오 블록들에 대해 동작한다. 비디오 블록은 CU 내 코딩 노드에 대응할 수도 있다. 비디오 블록들은 고정 또는 가변 사이즈들을 가질 수도 있으며, 규정된 코딩 표준에 따라서 사이즈가 상이할 수도 있다.

[0051] 일 예로서, HM 은 여러 PU 사이즈들에서 예측을 지원한다. 특정의 CU 의 사이즈가 $2N \times 2N$ 이라고 가정하면, HM 은 $2N \times 2N$ 또는 $N \times N$ 의 PU 사이즈들에서는 인트라-예측을, 그리고 $2N \times 2N$, $2N \times N$, $N \times 2N$, 또는 $N \times N$ 의 대칭적인 PU 사이즈들에서는 인터-예측을 지원한다. HM 은 또한 $2N \times nU$, $2N \times nD$, $nL \times 2N$, 및 $nR \times 2N$ 의 PU 사이즈들에서 인터-예측에 대해 비대칭적인 파티셔닝을 지원한다. 비대칭적인 파티셔닝에서, CU 의 하나의 방향은 파티셔닝되지 않지만, 다른 방향은 25% 및 75% 로 파티셔닝된다. 25% 파티션에 대응하는 CU 의 부분은 "상부 (Up)", "하부 (Down)", "좌측 (Left)", 또는 "우측 (Right)" 의 표시가 뒤따르는 "n" 으로 표시된다. 따라서, 예를 들어, " $2N \times nU$ " 는 최상부에서 $2N \times 0.5N$ PU 로 그리고 최저부에서 $2N \times 1.5N$ PU 로 수평으로 파티셔닝된 $2N \times 2N$ CU 를 지칭한다.

[0052] 본 개시물에서, " $N \times N$ " 및 "N 곱하기 N" 은 수직 및 수평 치수들의 관점에서 비디오 블록의 픽셀 치수들, 예컨대, 16×16 픽셀들 또는 16 곱하기 16 픽셀들을 지칭하기 위해 상호교환가능하게 사용될 수도 있다. 일반적으로, 16×16 블록은 수직 방향으로 16개의 픽셀들 ($y = 16$) 및 수평 방향으로 16개의 픽셀들 ($x = 16$) 을 가질 것이다. 이와 유사하게, $N \times N$ 블록은 일반적으로 수직 방향으로 N 개의 픽셀들 및 수평 방향으로 N 개의 픽셀들을 가지며, 여기서 N 은 음이 아닌 정수 값을 나타낸다. 블록에서 픽셀들은 로우들 및 칼럼들로 배열될 수도 있다. 더욱이, 블록들은 수직 방향에서와 같이 수평 방향에서 동일한 픽셀들의 개수를 반드시 가질 필요는 없다. 예를 들어, 블록들은 $N \times M$ 픽셀들을 포함할 수도 있으며, 여기서 M 은 반드시 N 과 같은 필요는 없다.

[0053] CU 의 PUs 를 이용한 인트라-예측 또는 인터-예측 코딩 이후, 비디오 인코더 (20) 는 CU 의 TU들에 대한 잔여 데이터를 계산할 수도 있다. PUs 는 공간 도메인 (또한, 픽셀 도메인으로 지칭됨) 에서 예측 픽셀 데이터를 발생하는 방법 또는 모드를 기술하는 구문 데이터를 포함할 수도 있으며, TU들은 이산 코사인 변환 (DCT), 정수 변환, 웨이블릿 변환, 또는 개념적으로 유사한 변환과 같은 변환의 잔여 비디오 데이터에의 적용 이후 변환 도메인에서의 계수들을 포함할 수도 있다. 잔여 데이터는 미인코딩된 화상의 픽셀들과 PUs 에 대응하는 예측 값들 사이의 픽셀 차이들에 대응할 수도 있다. 비디오 인코더 (20) 는 CU 에 대한 잔여 데이터를 포함하는 TU들을 형성하고, 그후 그 TU들을 변환하여, 그 CU 에 대한 변환 계수들을 발생할 수도 있다.

[0054] 변환 계수들을 발생하는 임의의 변환들 이후, 비디오 인코더 (20) 는 변환 계수들의 양자화를 수행할 수도 있다. 양자화는 일반적으로 계수들을 나타내는데 사용되는 데이터의 양을 가능한 한 감축하기 위해 변환 계수들이 양자화되는 프로세스를 지칭하며, 추가적인 압축을 제공한다. 양자화 프로세스는 그 계수들의 일부 또는 모두와 연관되는 비트 심도를 감소시킬 수도 있다. 예를 들어, n-비트 값은 양자화 동안 m-비트 값까지 절사될 수도 있으며, 여기서, n 은 m 보다 더 크다.

[0055] 양자화 이후, 비디오 인코더는 변환 계수들을 스캐닝하여, 양자화된 변환 계수들을 포함하는 2차원 매트릭스로부터 1차원 벡터를 발생할 수도 있다. 스캐닝은 어레이의 전면에서 높은 에너지 (따라서, 낮은 주파수) 계수들을 배치하고, 그리고 어레이의 후면에서 낮은 에너지 (따라서, 높은 주파수) 계수들을 배치하도록 설계될 수도 있다. 일부 예들에서, 비디오 인코더 (20) 는 엔트로피 인코딩될 수 있는 직렬화된 벡터를 발생하기 위해, 미리 정의된 스캐닝 순서를 이용하여, 양자화된 변환 계수들을 스캐닝할 수도 있다. 다른 예들에서, 비디오 인코더 (20) 는 적응적 스캐닝을 수행할 수도 있다. 양자화된 변환 계수들을 스캐닝하여 1차원 벡터를 형성한 후, 비디오 인코더 (20) 는 예컨대, 컨텍스트-적응 가변 길이 코딩 (CAVLC), 컨텍스트-적응 2진 산술 코딩 (CABAC), 구문-기반의 컨텍스트-적응 2진 산술 코딩 (SBAC), 확률 간격 파티셔닝 엔트로피 (PIPE) 코딩 또는 또 다른 엔트로피 인코딩 방법론에 따라서, 1차원 벡터를 엔트로피 인코딩할 수도 있다. 비디오 인코더 (20) 는 또한 비디오 데이터를 디코딩할 때에 비디오 디코더 (30) 에 의해 사용하기 위한 인코딩된 비디오 데이터와 연관되는 구문 엘리먼트들을 엔트로피 인코딩할 수도 있다.

[0056] CABAC 를 수행하기 위해, 비디오 인코더 (20) 는 컨텍스트 모델 내 컨텍스트를 송신되는 심볼에 할당할 수도 있다. 컨텍스트는 예를 들어, 심볼의 이웃하는 값들이 비-제로인지 여부에 관련될 수도 있다. CAVLC 를 수행하기 위해, 비디오 인코더 (20) 는 송신되는 심볼에 대해 가변 길이 코드를 선택할 수도 있다. VLC 에

서의 코드워드들은 상대적으로 더 짧은 코드들이 더 가능성 있는 심볼들에 대응하지만, 더 긴 코드들이 덜 가능성 있는 심볼들에 대응하도록, 구성될 수도 있다. 이와 같이, VLC의 사용은 예를 들어, 송신되는 각각의 심볼에 대해 동일-길이 코드워드들을 사용하는 것을 넘어서 비트 절감을 달성할 수도 있다. 확률 결정은 그 심볼에 할당된 컨텍스트에 기초할 수도 있다.

[0057] 이 섹션에서, 멀티뷰 및 멀티뷰 플러스 심도 코딩 기법들이 설명될 것이다. 먼저, MVC 기법들이 설명될 것이다. 위에서 언급한 바와 같이, MVC는 ITU-T H.264/AVC의 확장판이다. MVC에서, 복수의 뷰들에 대한 데이터는 시간-우선 순서로 코딩되며, 따라서, 디코딩 순서 배열은 시간-우선 코딩으로서 지칭된다. 특히, 공통 시간 인스턴스에서 복수의 뷰들의 각각에 대한 뷰 성분들(즉, 화상들)이 코딩될 수도 있으며, 그 후 상이한 시간 인스턴스에 대한 또 다른 뷰 성분들의 세트가 코딩될 수도 있으며, 기타 등등으로 코딩될 수도 있다. 액세스 유닛은 하나의 출력 시간 인스턴스에 대한 모든 뷰들의 코딩된 화상들을 포함할 수도 있다. 액세스 유닛들의 디코딩 순서는 출력(또는, 디스플레이) 순서와 반드시 동일하지는 않는 것으로 이해되어야 한다.

[0058] 도 2는 전형적인 MVC 디코딩 순서(즉, 비트스트림 순서)를 나타낸다. 디코딩 순서 배열은 시간-우선 코딩(time-first coding)으로서 지칭된다. 액세스 유닛들의 디코딩 순서가 출력 또는 디스플레이 순서와 동일하지 않을 수도 있다는 점에 유의한다. 도 2에서, S0-S7는 각각 멀티뷰 비디오의 상이한 뷰들을 지칭한다. T0-T8는 각각 하나의 출력 시간 인스턴스를 나타낸다. 액세스 유닛은 하나의 출력 시간 인스턴스에 대한 모든 뷰들의 코딩된 화상들을 포함할 수도 있다. 예를 들어, 제 1 액세스 유닛은 시간 인스턴스 T0에 대한 뷰들 S0-S7의 모두를 포함할 수도 있으며, 제 2 액세스 유닛은 시간 인스턴스 T1에 대한 뷰들 S0-S7의 모두를 포함할 수도 있으며, 기타 등등으로 포함할 수도 있다.

[0059] 간결성의 목적들을 위해, 본 개시물은 다음의 정의들을 이용할 수도 있다:

[0060] 뷰 성분: 단일 액세스 유닛에서의 뷰의 코딩된 표현. 뷰가 코딩된 텍스처 및 심도 표현들 양쪽을 포함할 때, 뷰 성분은 텍스처 뷰 성분 및 심도 뷰 성분으로 구성된다.

[0061] 텍스처 뷰 성분: 단일 액세스 유닛에서의 뷰의 텍스처의 코딩된 표현.

[0062] 심도 뷰 성분: 단일 액세스 유닛에서의 뷰의 심도의 코딩된 표현.

[0063] 도 2에서, 뷰들 각각은 화상들의 세트들을 포함한다. 예를 들어, 뷰 S0는 화상들 0, 8, 16, 24, 32, 40, 48, 56, 및 64의 세트를 포함하며, 뷰 S1은 화상들 1, 9, 17, 25, 33, 41, 49, 57, 및 65 등의 세트를 포함한다. 각각의 세트는 2개의 화상들을 포함한다: 하나의 화상은 텍스처 뷰 성분으로서 지칭되며, 다른 화상은 심도 뷰 성분으로서 지칭된다. 뷰의 화상들의 세트 내 텍스처 뷰 성분 및 심도 뷰 성분은 서로에 대응하는 것으로 간주될 수도 있다. 예를 들어, 뷰의 화상들의 세트 내 텍스처 뷰 성분은 뷰의 화상들의 세트 내 심도 뷰 성분에 대응하는 것으로 간주되며, 반대로도 마찬가지이다(즉, 심도 뷰 성분은 그 세트에서의 그의 텍스처 뷰 성분에 대응하며, 반대로도 마찬가지이다). 본 개시물에서 사용될 때, 심도 뷰 성분에 대응하는 텍스처 뷰 성분은 단일 액세스 유닛의 동일한 뷰의 부분인, 텍스처 뷰 성분 및 심도 뷰 성분으로서 간주될 수도 있다.

[0064] 텍스처 뷰 성분은 디스플레이되는 실제 이미지 콘텐츠를 포함한다. 예를 들어, 텍스처 뷰 성분은 루마(Y) 및 크로마(Cb 및 Cr) 성분들을 포함할 수도 있다. 심도 뷰 성분은 그의 대응하는 텍스처 뷰 성분에서의 픽셀들의 상대적인 심도들을 나타낼 수도 있다. 일 예로서, 심도 뷰 성분은 오직 루마 값들을 포함하는 그레이 스케일 이미지이다. 즉, 심도 뷰 성분은 임의의 이미지 콘텐츠를 전달하기 보다는, 오히려 텍스처 뷰 성분에서의 픽셀들의 상대적인 심도들의 측정치를 제공할 수도 있다.

[0065] 예를 들어, 심도 뷰 성분에서의 순수 백색 픽셀은 대응하는 텍스처 뷰 성분에서의 그의 대응하는 픽셀 또는 픽셀들이 뷰어의 관점으로부터 더 가깝다는 것을 나타내며, 심도 뷰 성분에서의 순수 블랙 픽셀은 대응하는 텍스처 뷰 성분에서의 그의 대응하는 픽셀 또는 픽셀들이 뷰어의 관점으로부터 더 멀리 있다는 것을 나타낸다. 블랙과 백색 사이에서의 그레이의 여러 음영들은 상이한 심도 레벨들을 나타낸다. 예를 들어, 심도 뷰 성분에서의 순수(very) 그레이 픽셀은 텍스처 뷰 성분에서의 그의 대응하는 픽셀이 심도 뷰 성분에서의 약한(slightly) 그레이 픽셀보다 더 멀리 있다는 것을 나타낸다. 오직 그레이 스케일이 픽셀들의 심도를 식별하는데 요구되기 때문에, 심도 뷰 성분은, 심도 뷰 성분에 대한 칼라 값들이 임의의 목적에 적합하지 않을 수도 있어, 크로마 성분들을 포함할 필요가 없다.

[0066] 심도를 식별하는데 오직 루마 값들(예컨대, 강도 값들)을 이용하는 심도 뷰 성분은 예시 목적들을 위해 제공

되며, 한정하는 것으로 간주되어서는 안된다. 다른 예들에서, 임의의 기법이 텍스처 뷰 성분에서의 픽셀들의 상대적인 심도들을 나타내는데 이용될 수도 있다.

- [0067] 도 3 은 멀티-뷰 비디오 코딩을 위한 (각각의 뷰 내에서의 인터-화상 예측 및 인터-뷰 예측 양쪽을 포함한) 전형적인 MVC 예측 구조를 나타낸다. 도 3 에서 예측 방향들은 화살표들로 표시되며, 여기서, 지시도달 오브젝트는 지시출발 오브젝트를 예측 참조로서 이용한다. MVC 에서, 인터-뷰 예측은 H.264/AVC 모션 보상의 구문을 이용하지만, 상이한 뷰에서의 화상이 참조 화상으로서 사용될 수 있게 하는 디스패리티 모션 보상에 의해 지원된다.
- [0068] 도 3 의 예에서, (뷰 IDs "S0" 내지 "S5" 를 갖는) 6개의 뷰들이 예시되며, 각각의 뷰에 대해 12개의 시간 로케이션들 ("T0" 내지 "T11") 이 예시된다. 즉, 도 3 에서 각각의 로우는 뷰에 대응하지만, 각각의 칼럼은 시간 로케이션을 나타낸다.
- [0069] MVC 가 H.264/AVC 디코더들에 의해 디코딩가능한 소위 베이스 뷰를 갖더라도, 스테레오 뷰 쌍들은 또한 MVC 에 의해 지원될 수 있으며, MVC 의 이점은 2개보다 많은 뷰들을 3D 비디오 입력으로서 이용하고 다수의 뷰들에 의해 표현되는 이 3D 비디오를 디코딩하는 예를 지원할 수 있다는 점이다. MVC 디코더를 갖는 클라이언트의 렌더러는 다수의 뷰들을 가진 3D 비디오 콘텐츠를 기대할 수도 있다.
- [0070] 도 3 에서의 화상들은 각각의 로우와 각각의 칼럼의 교차점에 표시된다. H.264/AVC 표준은 비디오의 부분을 나타내기 위해 용어 프레임을 이용할 수도 있다. 본 개시물은 용어 화상과 프레임을 상호교환가능하게 사용할 수도 있다.
- [0071] 도 3 에서의 화상들은 문자를 포함하는 블록을 이용하여 예시되며, 이 문자는 대응하는 화상이 인트라-코딩되거나 (즉, I-화상), 또는 하나의 방향으로 (즉, P-화상으로서) 또는 다수의 방향으로 (즉, B-화상으로서) 인터-코딩되는지 여부를 지시한다. 일반적으로, 예측들은 화살표들로 표시되며, 여기서, 지시도달 화상들은 예측 참조를 위해 지시출발 화상을 이용한다. 예를 들어, 시간 로케이션 T0 에서의 뷰 S2 의 P-화상은 시간 로케이션 T0 에서 뷰 S0 의 I-화상으로부터 예측된다.
- [0072] 단일 뷰 비디오 인코딩에서와 같이, 멀티뷰 비디오 코딩 비디오 시퀀스의 화상들은 상이한 시간 로케이션들에서의 화상들에 대해 예측 인코딩될 수도 있다. 예를 들어, 시간 로케이션 T1 에서의 뷰 S0 의 b-화상은 b-화상이 I-화상으로부터 예측된다는 것을 나타내는, 시간 로케이션 T0 에서 뷰 S0 의 I-화상으로부터 그를 가리키는 화살표를 갖는다. 게다가, 그러나, 멀티뷰 비디오 인코딩의 상황에서, 화상들은 인터-뷰 예측될 수도 있다. 즉, 뷰 성분은 참조를 위해 다른 뷰들에서의 뷰 성분들을 이용할 수도 있다. MVC 에서, 예를 들어, 인터-뷰 예측은 마치 또 다른 뷰에서의 뷰 성분이 인터-예측 참조인 것처럼 실현된다. 잠재적인 인터-뷰 참조들은 시퀀스 파라미터 세트 (SPS) MVC 확장판으로 시그널링되며, 인터-예측 또는 인터-뷰 예측 참조들의 유연한 순서정렬을 가능하게 하는 참조 화상 리스트 구성 프로세스에 의해 수정될 수 있다. 인터-뷰 예측은 또한 3D-HEVC (멀티뷰 플러스 심도) 를 포함한, HEVC 의 제안된 멀티뷰 확장판의 특성이다.
- [0073] 도 3 은 인터-뷰 예측의 여러 예들을 제공한다. 뷰 S1 의 화상들은, 도 3 의 예에서, 뷰 S1 의 상이한 시간 로케이션들에서 화상들로부터 예측될 뿐만 아니라, 동일한 시간 로케이션들에서 뷰들 S0 및 S2 의 화상들로부터 인터-뷰 예측되는 것으로 예시된다. 예를 들어, 시간 로케이션 T1 에서의 뷰 S1 의 b-화상은 시간 로케이션들 T0 및 T2 에서 뷰 S1 의 B-화상들의 각각 뿐만 아니라, 시간 로케이션 T1 에서 뷰들 S0 및 S2 의 b-화상들로부터 예측된다.
- [0074] 일부 예들에서, 도 3 은 텍스처 뷰 성분들을 예시하는 것으로 보여질 수도 있다. 예를 들어, 도 2 에 예시된 I-, P-, B-, 및 b-화상들은 뷰들의 각각에 대한 텍스처 뷰 성분들로서 간주될 수도 있다. 본 개시물에서 설명하는 기법들에 따르면, 도 3 에 예시된 텍스처 뷰 성분들의 각각에 대해 대응하는 심도 뷰 성분이 존재한다. 일부 예들에서, 심도 뷰 성분들은 대응하는 텍스처 뷰 성분들에 도 3 에 예시된 방법과 유사한 방법으로 예측될 수도 있다.
- [0075] 2개의 뷰들의 코딩은 또한 MVC 에 의해 지원될 수 있다. MVC 의 이점들 중 하나는 MVC 인코더가 2개보다 많은 뷰들을 3D 비디오 입력으로서 취할 수 있고 MVC 디코더가 이러한 멀티뷰 표현을 디코딩할 수 있다는 점이다. 이와 같이, MVC 디코더를 가진 임의의 렌더러는 2개보다 많은 뷰들을 가진 3D 비디오 콘텐츠를 기대할 수도 있다.
- [0076] MVC 에서, 인터-뷰 예측은 동일한 액세스 유닛에서 (즉, 동일한 시간 인스턴스를 가진) 화상들 사이에 허용된다. 비-베이스 뷰들 중 하나에서의 화상을 코딩할 때, 상이한 뷰에 있지만 동일한 시간 인스턴스 내

에 있으면, 화상이 참조 화상 리스트에 추가될 수도 있다. 인터-뷰 참조 화상은 마치 임의의 인터 예측 참조 화상처럼, 참조 화상 리스트의 임의의 위치에 삽입될 수 있다. 도 3에 나타낸 바와 같이, 뷰 성분은 참조를 위해 다른 뷰들에서의 뷰 성분들을 이용할 수 있다. MVC에서, 인터-뷰 예측은, 또 다른 뷰에서의 뷰 성분이 인터-예측 참조였던 것처럼, 실현된다.

[0077] 다음은 멀티뷰 코딩 및/또는 심도에 의한 멀티뷰 코딩 (MV-HEVC) (3D-HEVC) 과 함께 사용될 수도 있는 인터-예측에 관련된 일부 적절한 HEVC 기법들을 기술한다. 논의를 위한 제 1 기법은 인터-예측을 위한 참조 화상 리스트 구성이다.

[0078] 인터-예측을 이용하여 PU 를 코딩하는 것은 현재의 블록 (예컨대, PU) 과 참조 프레임에서의 블록 사이에 모션 벡터를 계산하는 것을 수반한다. 모션 벡터들은 모션 추정 (또는, 모션 탐색) 으로 불리는 프로세스를 통해서 계산된다. 모션 벡터는, 예를 들어, 참조 프레임의 참조 샘플에 대한, 현재의 프레임에서의 예측 유닛의 변위를 나타낼 수도 있다. 참조 샘플은 코딩중인 PU 를 포함하는 CU 의 부분에 픽셀 차이의 관점에서 가깝게 매칭하는 것으로 발견되는 블록일 수도 있으며, 이것은 SAD (sum of absolute difference), SSD (sum of squared difference), 또는 다른 차이 메트릭들에 의해 결정될 수도 있다. 참조 샘플은 참조 프레임 또는 참조 슬라이스 내 어딘가에서 발생할 수도 있다. 일부 예들에서, 참조 샘플은 분수 픽셀 위치에서 발생할 수도 있다. 현재의 부분에 최상으로 매칭하는 참조 프레임의 부분을 발견하자마자, 인코더는 현재의 블록에 대한 현재의 모션 벡터를, 현재의 블록으로부터 참조 프레임에서의 매칭 부분까지 (예컨대, 현재의 블록의 중심으로부터 매칭 부분의 중심까지) 의 로케이션에서의 차이로서 결정한다.

[0079] 일부 예들에서, 인코더는 각각의 블록에 대한 모션 벡터를 인코딩된 비디오 비트스트림으로 시그널링할 수도 있다. 시그널링된 모션 벡터는 비디오 데이터를 디코딩하기 위해 모션 보상을 수행하는 디코더에 의해 사용된다. 그러나, 원래 모션 벡터를 직접 시그널링하는 것은 정보를 운반하는데 일반적으로 다수의 비트들이 요구되기 때문에, 덜 효율적인 코딩을 초래할 수도 있다.

[0080] 일부의 경우, 원래 모션 벡터를 직접 시그널링하기 보다는, 인코더는 각각의 파티션에 대해, 즉, 각각의 PU 에 대해 모션 벡터를 예측할 수도 있다. 이 모션 벡터 예측을 수행할 때에, 인코더는 현재의 블록과 동일한 프레임에서 공간적으로 이웃하는 블록들로부터 결정된 모션 벡터 후보들의 세트 또는 참조 프레임 (즉, 현재의 프레임 이외의 프레임) 에서 동일 장소에 배치된 블록으로부터 결정된 시간 모션 벡터 후보를 선택할 수도 있다. 비디오 인코더 (20) 는 모션 벡터 예측을 수행할 수도 있으며, 필요하면, 시그널링에서의 비트 레이트를 감소시키기 위해 원래 모션 벡터를 시그널링하기 보다는, 모션 벡터를 예측하기 위해 참조 화상으로서의 인덱스를 시그널링할 수도 있다. 공간적으로 이웃하는 블록들로부터의 모션 벡터 후보들은 공간 MVP 후보들로서 지칭될 수도 있으며, 반면 또 다른 참조 프레임에서 동일 장소에 배치된 블록들로부터의 모션 벡터 후보들은 시간 MVP 후보들로서 지칭될 수도 있다.

[0081] 모션 벡터 예측의 2개의 상이한 모드들 또는 유형들이 HEVC 표준에 제안되어 있다. 하나의 모드는 "병합" 모드로서 지칭된다. 다른 모드는 적응적 모션 벡터 예측 (AMVP) 로서 지칭된다.

[0082] 병합 모드에서, 비디오 인코더 (20) 는 비디오 디코더 (30) 에게, 예측 구문의 비트스트림 시그널링을 통해서, 모션 벡터, (모션 벡터가 가리키는 참조 프레임을, 주어진 참조 화상 리스트에서, 식별하는) 참조 인덱스 및 (참조 화상 리스트 (리스트 0 또는 리스트 1) 를, 즉, 참조 프레임이 현재 프레임에 시간적으로 선행하는지 또는 뒤따르는지 여부의 관점에서 식별하는) 모션 예측 방향을 프레임의 현재의 블록에 대한 선택된 모션 벡터 후보로부터 복사하도록 명령한다. 이것은 선택된 모션 벡터 후보 (즉, 특정의 공간 MVP 후보 또는 시간 MVP 후보) 를 식별하는 모션 벡터 후보 리스트로의 인덱스를 비트스트림으로 시그널링함으로써 달성된다.

[0083] 따라서, 병합 모드에 있어, 예측 구문은 모드 (이 경우, "병합" 모드) 를 식별하는 플래그, 및 선택된 모션 벡터 후보를 식별하는 인덱스를 포함할 수도 있다. 일부의 경우, 모션 벡터 후보는 현재의 블록과 관련하여 인과적 블록에 있을 것이다. 즉, 모션 벡터 후보는 비디오 디코더 (30) 에 의해 이미 디코딩되어 있을 것이다. 이와 같이, 비디오 디코더 (30) 는 인과적 블록에 대한 모션 벡터, 참조 인덱스, 및 모션 예측 방향을 이미 수신하였거나 및/또는 결정하였다. 따라서, 비디오 디코더 (30) 는 간단히 메모리로부터 인과적 블록과 연관된 모션 벡터, 참조 인덱스, 및 모션 예측 방향을 추출하여 이 값들을 현재의 블록에 대한 모션 정보로서 복사할 수도 있다. 병합 모드에서 블록을 재구성하기 위해, 비디오 디코더 (30) 는 도출된 현재의 블록에 대한 모션 정보를 이용하여 예측 블록을 획득하고, 그 예측 블록에 잔여 데이터를 가산하여, 코딩된 블록을 재구성한다. 스킵 모드에 있어서는, 동일한 병합 후보 리스트가 발생되지만 어떤 잔여도 시그널링되지 않는다. 간결성을 위해, 스킵 모드가 병합 모드와 동일한 모션 벡터 도출 프로세스를 가지기 때문에, 이 문서에

서 설명되는 모든 기법들은 병합 및 스킵 모드들 양쪽에 적용된다.

[0084] AMVP 에서, 비디오 인코더 (20) 는 비디오 디코더 (30) 에게, 비트스트림 시그널링을 통해서, 단지 후보 블록으로부터 모션 벡터를 복사하고, 그 복사된 벡터를 현재의 블록의 모션 벡터에 대한 예측자로서 이용하여, 모션 벡터 차이 (MVD) 를 시그널링하도록 명령한다. 현재의 블록의 모션 벡터와 연관되는 예측 방향 및 참조 프레임은 별개로 시그널링된다. MVD 는 현재의 블록에 대한 현재의 모션 벡터와 후보 블록으로부터 도출된 모션 벡터 예측자 사이의 차이이다. 이 경우, 비디오 인코더 (20) 는, 모션 추정을 이용하여, 코딩되는 블록에 대한 실제 모션 벡터를 결정하고, 그 후 실제 모션 벡터와 모션 벡터 예측자 사이의 차이를 MVD 값으로 결정한다. 이러한 방법으로, 비디오 디코더 (30) 는 병합 모드에서 처럼, 정확한 모션 벡터 후보의 복사본을 현재의 모션 벡터로서 이용하지 않으며, 그러나, 대신, 모션 추정으로부터 결정된 현재의 모션 벡터에 값에 있어서 "가까울" 수도 있는 모션 벡터 후보를 이용하고 MVD 를 가산하여 현재의 모션 벡터를 복원할 수도 있다. AMVP 모드에서 블록을 재구성하기 위해, 디코더는 대응하는 잔여 데이터를 가산하여 그 코딩된 블록을 재구성한다.

[0085] 대부분의 상황들에서, MVD 는 전체 현재의 모션 벡터보다 더 적은 시그널링할 비트들을 필요로 한다. 이와 같이, AMVP 는 현재의 모션 벡터의 더 정확한 시그널링을 가능하게 하면서도 전체 모션 벡터를 전송하는 것을 넘어서는 코딩 효율을 유지한다. 이에 반해, 병합 모드는 MVD 의 사양을 고려하지 않으며, 이와 같이, 병합 모드는 증가된 시그널링 효율 (즉, 더 적은 비트들) 을 위해 모션 벡터 시그널링의 정확도를 희생한다. AMVP 에 대한 예측 구문은 그 모드에 대한 플래그 (이 경우, AMVP 플래그), 후보 블록에 대한 인덱스, 현재의 모션 벡터와 후보 블록으로부터의 예측 모션 벡터 사이의 MVD, 참조 인덱스, 및 모션 예측 방향을 포함할 수도 있다.

[0086] 도 4 는 병합 모드 및 AMVP 모드 양쪽에서 사용될 수도 있는 예시적인 후보 블록들의 세트 (120) 를 나타낸다. 이 예에서, 후보 블록들은 좌하측 (A0) (121), 좌측 (A1) (122), 좌상측 (B2) (125), 상측 (B1) (124), 및 우상측 (B0) (123) 공간 위치들에, 그리고 시간 (T) (126) 위치(들) 에 있다. 이 예에서, 좌측 후보 블록 (122) 은 현재의 블록 (127) 의 좌측 에지에 인접한다. 좌측 블록 (122) 의 하부 에지는 현재의 블록 (127) 의 하부 에지와 정렬된다. 상측 블록 (124) 은 현재의 블록 (127) 의 상부 에지와 인접한다. 상측 블록 (124) 의 우측 에지는 현재의 블록 (127) 의 우측 에지와 정렬된다.

[0087] 블록들 A_0 , A_1 , B_0 , B_1 및 B_2 는 공간적으로 이웃하는 블록들 또는 공간적으로 이웃하는 PUs 로서 일반적으로 지칭된다. 비디오 코더 (20/30) 는 현재의 화상의 좌상단 샘플에 대해 현재의 PU (예컨대, 도 4 에서의 현재의 블록 (127)) 의 좌상단 루마 샘플의 로케이션을 먼저 결정함으로써, 공간적으로 이웃하는 블록들의 로케이션들을 결정할 수도 있다. 이 루마 로케이션은 로케이션 (xP, yP) 로서 지칭될 것이다. 비디오 코더 (20/30) 는 루마 PU (nPSW 로 축약됨) 의 폭 및 루마 PU (nPSH 로 축약됨) 의 높이를 추가로 결정할 수도 있다. 현재의 PU 의 좌상단 루마 샘플의 로케이션 및 현재의 PU 의 높이 및 폭에 기초하여, 공간 이웃하는 블록들의 루마 샘플의 로케이션은 다음과 같이 결정될 수 있다:

[0088] 블록 A0 는 로케이션 (xP - 1, yP + nPSH) 에서의 루마 샘플을 포함하며;

[0089] 블록 A1 은 로케이션(xP - 1, yP + nPSH - 1) 에서의 루마 샘플을 포함하며;

[0090] 블록 B0 은 로케이션 (xP + nPSW, yP - 1) 에서의 루마 샘플을 포함하며;

[0091] 블록 B1 은 로케이션 (xP + nPSW - 1, yP - 1) 에서의 루마 샘플을 포함하며;

[0092] 블록 B2 는 로케이션 (xP - 1, yP - 1) 에서의 루마 샘플을 포함한다.

[0093] 공간 이웃하는 블록들은 동일한 사이즈일 수도 있거나 또는 현재의 PU 와 상이한 사이즈들일 수도 있다.

[0094] 병합 모드에서의 사용을 위한 후보 리스트는 위에서 설명된 공간 이웃하는 블록들을 이용하여 도출될 수 있다. 모션 벡터들 및 양쪽의 참조 리스트들에 대한 참조 인덱스들은 공간 병합 후보들에 대한 도출 프로세스의 일 예인 다음 순서로 된 단계들에 의해 규정되는 바와 같이, 도출될 수도 있다. 도출 프로세스의 일부로서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 공간 이웃하는 PUs 의 모션 정보를 다음 순서로 체크한다: A_1 , B_1 , B_0 , A_0 , 또는 B_2 . 비디오 인코더 (20) 및 비디오 디코더 (30) 는 복제된 후보들이 후보 리스트에 추가되는 것을 감소시키거나 또는 제거하기 위해, 속박된 프루닝을 수행할 수도 있다. 비디오 코더 (20/30) 는 병합 모드 후보 리스트에 포함될 공간 후보들을 다음과 같이 도출할 수도 있다:

- [0095] - A_1 이 이용가능하면 비디오 코더 (20/30) 는 A_1 을 후보 리스트에 삽입한다.
- [0096] - B_1 및 A_1 이 동일한 모션 벡터들 및 동일한 참조 인덱스들을 가지면, 비디오 코더 (20/30) 는 B_1 을 후보 리스트에 삽입하지 않는다. B_1 및 A_1 이 동일한 모션 벡터들 및 동일한 참조 인덱스들을 갖지 않으면, 비디오 코더 (20/30) 는 B_1 이 이용가능하면 B_1 을 후보 리스트에 삽입한다.
- [0097] - B_0 및 B_1 이 동일한 모션 벡터들 및 동일한 참조 인덱스들을 가지면, 비디오 코더 (20/30) 는 B_0 를 후보 리스트에 삽입하지 않는다. B_0 및 B_1 이 동일한 모션 벡터들 및 동일한 참조 인덱스들을 갖지 않으면, 비디오 코더 (20/30) 는 B_0 가 이용가능하면, B_0 를 후보 리스트에 삽입한다.
- [0098] - A_0 및 A_1 이 동일한 모션 벡터들 및 동일한 참조 인덱스들을 가지면, 비디오 코더 (20/30) 는 A_0 를 후보 리스트에 삽입하지 않는다. A_0 및 A_1 이 동일한 모션 벡터들 및 동일한 참조 인덱스들을 갖지 않으면, 비디오 코더 (20/30) 는 A_0 가 이용가능하면 A_0 를 후보 리스트에 삽입한다.
- [0099] - 비디오 코더 (20/30) 는 다음 조건들 양쪽이 만족되지 않을 때 B_2 를 후보 리스트에 삽입한다:
- [0100] - B_2 및 B_1 , 또는 B_2 및 A_1 은 동일한 모션 벡터들 및 동일한 참조 인덱스들을 갖는다.
- [0101] - A_1 , B_1 , B_0 , A_0 로부터 도출된 4개의 공간 병합 후보들 모두가 후보 리스트에 포함된다.
- [0102] 비디오 코더 (20/30) 는 병합 모드 후보 리스트에 포함될 시간 후보를 다음과 같이 도출할 수도 있다:
- [0103] - 시간 병합 후보를 얻기 위해, 비디오 코더 (20/30) 는 동일 장소에 배치된 화상을 먼저 식별한다. 현재의 화상이 B 슬라이스이면, 동일 장소에 배치된 화상이 RefPicList0 또는 RefPicList1 로부터 유래하는지 여부를 표시하기 위해, 구문 엘리먼트 "collocated_from_l0_flag" 가 슬라이스 헤더로 시그널링된다.
- [0104] - 비디오 코더 (20/30) 가 참조 화상 리스트를 식별한 후, 비디오 코더 (20/30) 는 슬라이스 헤더로 시그널링된, 구문 엘리먼트 collocated_ref_idx 를 이용하여, 그 리스트에서의 화상에서 화상을 식별한다.
- [0105] - 비디오 코더 (20/30) 는 그후 동일 장소에 배치된 화상을 체크함으로써, 동일 장소에 배치된 PU 를 식별한다. 비디오 코더 (20/30) 는 이 PU 를 포함하는 CU 의 우하단 PU 의 모션, 또는 이 PU 를 포함하는 CU 의 중심 PUs 내 우하단 PU 의 모션을 시간 후보로서 이용한다.
- [0106] - 상기 프로세스에 의해 식별되는 모션 벡터들이 병합 모드에 대한 모션 후보를 발생하는데 사용될 때, 비디오 코더 (20/30) 는 (화상 순서 카운트, 즉, POC 에 의해 반영된) 시간 로케이션에 기초하여, 모션 벡터들을 스케일링할 필요가 있을 수도 있다.
- [0107] - HEVC 에서, PPS 는 플래그 enable_temporal_mvp_flag 를 포함한다. 0 과 동일한 temporal_id 를 가진 특정의 화상이 0 과 동일한 enable_temporal_mvp_flag 를 갖는 PPS 를 참조할 때, 비디오 코더 (20/30) 는 DPB 에서의 모든 참조 화상들을 "시간 모션 벡터 예측용으로 미사용됨" 으로 표시할 수도 있으며, 디코딩 순서에서 그 특정의 화상 이전의 화상들로부터의 어떤 모션 벡터도 디코딩 순서에서 특정의 화상 이후 특정의 화상 또는 화상의 디코딩에서 시간 모션 벡터 예측자로서 사용되지 않을 것이다.
- [0108] 현재의 슬라이스가 B 슬라이스이고 위에서 설명된 공간 및 시간 후보 도출 프로세스들로부터 도출된 후보들의 총 개수가 후보들의 최대 개수 미만이고 1 보다 크면, 비디오 코더 (20/30) 는 양방향-예측 후보들을 발생하는 프로세스를 호출할 수도 있다. 비디오 코더 (20/30) 는 병합 모드 후보 리스트에 포함될 양방향-예측 후보들을 다음과 같이 도출할 수도 있다:
- [0109] - 아래 테이블 1 에 기초하여, 인덱스 combIdx 를 가진 결합된 양방향-예측 후보를 발생하기 위해, 이용가능하면, 10CandIdx 과 동일한 엔트리를 가진 후보 리스트의 RefList0 모션 정보 (MotList0), 및 이용가능하면, 11CandIdx 과 동일하고 MotList0 과 동일하지 않은 엔트리를 가진 후보 리스트의 RefList1 모션 정보 (MotList1) 가 비디오 코더 (20/30) 에 의해, 결합된 양방향-예측 후보의 RefList0 및 RefList1 모션 정보로서 재사용될 수도 있다.

[0110] HEVC 에서 10CandIdx 및 11CandIdx 의 사양

combIdx	0	1	2	3	4	5	6	7	8	9	10	11
10CandIdx	0	1	0	2	1	2	0	3	1	3	2	3
11CandIdx	1	0	2	0	2	1	3	0	3	1	3	2

[0111] 비디오 코더 (20/30) 는 공간, 시간, 및 양방향-예측 후보들을 결정하는 위에서 설명된 상기 3개의 프로세스들로부터 도출된 후보들의 총 개수가 후보들의 최대 개수 미만으로 발생하면, 병합 모드 후보 리스트에 포함될 제로 모션 벡터 후보들을 도출할 수도 있다. 이러한 경우에, 비디오 코더 (20/30) 는 각각의 참조 화상에 대한 제로 모션 벡터들을 후보 리스트에 삽입할 수도 있다. HEVC WD 10 에서, 병합 (MRG) 리스트에서의 후보들의 총 개수는 최고 5 이며, five_minus_max_num_merge_cand 가 5 에서 감소된 MRG 후보들의 최대 개수를 규정하기 위해 슬라이스 헤더로 시그널링된다.

[0112] 3D 비디오를 코딩하는 코딩 효율을 더욱 향상시키기 위해, 2개의 새로운 기술들, 즉 "인터-뷰 모션 예측" 및 "인터-뷰 잔여 예측" 이 3D-HEVC 에 도입되었다. 이들 2개의 코딩 툴들을 동작시키기 위해서, 비디오 코더 (20/30) 는 현재의 블록에 대한 디스패리티 벡터들을 도출하도록 구성된다.

[0113] 비디오 코더 (20/30) 는 공간 및 시간 이웃하는 블록들로부터의 디스패리티 모션 벡터들 또는 이웃하는 블록들로부터의 도출된 디스패리티 벡터를 이용하여 현재의 블록에 대한 디스패리티 벡터를 결정하는, 이웃하는 블록들 기반의 디스패리티 벡터 (NBDV) 로서 지칭되는 기법을 이용하여, 디스패리티 벡터를 도출하도록 구성될 수 있다. NBDV 에서, 비디오 코더 (20/30) 는 공간 또는 시간 이웃하는 블록들의 모션 벡터들을 고정된 체킹 순서로 체킹하고, 일단 디스패리티 모션 벡터가 식별되면, 즉, 모션 벡터가 인터-뷰 참조 화상을 가리키면, 비디오 코더 (20/30) 는 체킹 프로세스를 종료한다. 비디오 코더 (20/30) 는 식별된 디스패리티 모션 벡터를 반환하고 그것을 인터-뷰 모션 예측 및 인터-뷰 잔여 예측에 사용될 수 있는 현재의 블록에 대한 디스패리티 벡터로 변환한다. 어떤 디스패리티 모션 벡터도 모든 사전-정의된 이웃하는 블록들을 체킹한 후에 발견되지 않으면, 비디오 코더 (20/30) 는 인터-뷰 모션 예측에 제로 디스패리티 벡터를 이용할 수도 있지만 인터-뷰 잔여 예측은 대응하는 예측 유닛 (PU) 에 대해 사용불능될 수 있다.

[0114] NBDV 에 대해 비디오 코더 (20/30) 에 의해 사용되는 블록들은 공간 이웃하는 블록들 및 시간 이웃하는 블록들을 포함한다. 비디오 코더 (20/30) 는 디스패리티 벡터를 도출하는 것의 일부로서 최고 5개의 공간 이웃하는 블록들을 분석한다. 5개의 공간 이웃하는 블록들은 A₀, A₁, B₀, B₁ 또는 B₂ 로 표시된, 현재의 PU 의 좌하부 블록, 좌측 블록, 우상부 블록, 상부 블록, 및 좌상부 블록을 포함한다. 이 점에 있어서, NBDV 에 사용되는 공간 이웃하는 블록들 및 AMVP 및 병합 모드들에 대한 후보 리스트 구성이 반드시 동일할 필요가 없지만, NBDV 의 부분으로서 분석된 공간 이웃하는 블록들은, 도 4 에서 병합 모드에 관해 위에서 설명된 동일한 공간 이웃하는 블록들에 대응할 수도 있다.

[0115] 비디오 코더 (20/30) 는 또한 시간 이웃하는 블록들을 결정한다. 비디오 코더 (20/30) 는 현재의 뷰로부터의 모든 참조 화상들을 후보 화상들로서 취급할 수도 있거나, 또는 일부 구현예들에서, 후보 화상들의 개수는 예를 들어, 4개의 후보 화상들까지 추가로 제한될 수 있다. 비디오 코더 (20/30) 는 동일 장소에 배치된 참조 화상을 먼저 체킹하고, 후보 화상들의 나머지를, 예를 들어, 참조 인덱스 (refIdx) 의 오름 차순으로 체킹한다. RefPicList0[refIdx] 및 RefPicList1[refIdx] 양쪽이 이용가능할 때, RefPicListX[refIdx] 는 다른 화상보다 선행하며, 여기서, X 는 collocated_from_10_flag 와 동일하다.

[0116] 각각의 후보 화상에 대해, 비디오 코더 (20/30) 는 시간 이웃하는 블록들을 도출하기 위해 3개의 후보 영역들을 결정한다. 영역이 하나 보다 많은 16x16 블록을 커버할 때, 비디오 코더 (20/30) 는 모든 16x16 블록들을 이러한 영역에서 예를 들어, 래스터 스캐닝 순서로 체킹할 수 있다. 3개의 후보 영역들은 다음과 같이 정의된다: (1) CPU - 현재의 PU 또는 현재의 CU 의 동일 장소에 배치된 영역, (2) CLCU - 현재의 PU 의 동일 장소에 배치된 영역을 커버하는 최대 코딩 유닛 (LCU), 및 (3) BR - CPU 의 우하측 4x4 블록.

[0117] 참조 화상들의 시간 이웃하는 블록들을 체킹하는, 위에서 설명된 기법들은 일 예인 것으로 이해되어야 한다. 체킹되는 참조 화상들의 다른 예들이 존재할 수도 있다.

[0118] 도 5 는 현재의 블록에 대한 디스패리티 벡터를 결정하기 위해 후보 화상에서 체킹되는 블록들의 일 예를 나타낸다. 예를 들어, 비디오 코더 (20/30) 는 현재의 뷰로부터 최고 2개의 후보 화상들을 고려할 수도 있다. 현재의 뷰로부터의 제 1 후보 화상은 후보 리스트의 구성의 부분 (예컨대, 시간 후보) 으로서 식별된 동일

장소에 배치된 화상일 수도 있다. 제 2 후보 화상은 가장 작은 POC 차이 및 가장 작은 temporalId 을 가진 참조 화상 또는 랜덤-액세스 화상일 수도 있다. 일부 예들에서, 비디오 코더 (20/30) 는 랜덤-액세스 화상을 먼저 체크할 수도 있으며, 어떤 디스패리티 모션 벡터도 발견되지 않으면, 동일 장소에 배치된 화상을 체크하지 않는다.

[0119] 이들 후보 화상들의 각각에 대해, 비디오 코더 (20/30) 는 2개의 블록들을 체크할 수도 있다. 제 1 블록은 도 5 에 Pos. A 로 예시된 바와 같이, 현재의 PU 의 동일 장소에 배치된 영역의 중심 4x4 블록일 수도 있는, 중심 블록 (CR) 일 수도 있다. 제 2 블록은 도 5 에서 Pos. B 로 예시된 바와 같이, 현재의 PU 의 동일 장소에 배치된 영역의 우하측 4x4 블록일 수도 있는, 우하측 블록 (BR) 일 수도 있다.

[0120] 비디오 코더 (20/30) 는 블록들을 특정의 순서, 즉 체크 순서로 체크하도록 구성될 수도 있다. 비디오 코더 (20/30) 는 공간 이웃하는 블록들을 먼저 체크하고, 뒤이어서 시간 이웃하는 블록들을 체크할 수도 있다. 5 개의 공간 이웃하는 블록들의 체크 순서는 예를 들어, A₁, B₁, B₀, A₀ 및 B₂, 또는 임의의 다른 이러한 순서로서 정의될 수도 있다. 각각의 후보 화상에 대해, 비디오 코더 (20/30) 는 후보 화상에서 3개의 후보 영역들을 체크 순서로 체크할 수 있다. 3개의 영역들의 체크 순서는 예를 들어, 제 1 비-베이스 뷰에 대해서는 CPU, CLCU, 및 BR, 제 2 비-베이스 뷰에 대해서는 BR, CPU, 및 CLU, 또는 임의의 다른 이러한 순서로서 정의될 수도 있다. 또 다른 예에서, 각각의 후보 화상에 대해 (예컨대, 최고 2개의 후보 화상들이 존재하는 경우), 2개의 블록들은 제 1 비-베이스 뷰에 대해서는 CR 및 BR, 또는 제 2 비-베이스 뷰에 대해서는 BR, CR 순서로 체크된다.

[0121] 위에서 설명된 바와 같이, 비디오 코더 (20/30) 는 이웃하는 블록들에 대한 디스패리티 모션 벡터들을 체크하고 이웃하는 블록에 대한 디스패리티 모션 벡터가 발견되면, 현재의 블록의 디스패리티 벡터로서 그 디스패리티 모션 벡터를 변환함으로써, 현재의 블록에 대한 디스패리티 벡터를 결정할 수도 있다. 게다가, 일부 예들에서, 이웃하는 블록에 대한 도출된 디스패리티 벡터는 현재의 블록에 대한 디스패리티 벡터를 결정하는데 사용될 수도 있다. 예를 들어, 이웃하는 블록은 디스패리티 모션 벡터로 인터-예측되지 않을 수도 있으며, 그러나 이웃하는 블록에 대한 디스패리티 벡터는 이웃하는 블록의 예측 동안 도출되었을 수도 있다. 이 도출된 디스패리티 벡터는 암시적인 디스패리티 벡터 (IDV; implicit disparity vector) 로서 지칭될 수도 있으며, 현재의 블록에 대한 디스패리티 벡터를, 예컨대, 체크되는 어떤 이웃하는 블록들도 디스패리티 모션 벡터로 인터-예측되지 않을 때, 결정하는데 사용될 수도 있다.

[0122] 예를 들어, 암시적인 디스패리티 벡터 (IDV) 는, PU 가 인터-뷰 모션 벡터 예측, 즉, AMVP 에 대한 후보를 채용하거나 또는 병합 모드들이 디스패리티 벡터의 도움으로 다른 뷰에서의 대응하는 블록으로부터 도출될 때, 발생된다. 이러한 디스패리티 벡터는 IDV 로 지칭된다. IDV 는 디스패리티 벡터 도출의 목적을 위해 PU 에 저장된다.

[0123] 좀더 자세하게 설명되는 바와 같이, 비디오 코더 (20/30) 는 인터-뷰 후보 도출을 수행하도록 구성될 수도 있다. 디스패리티 벡터 (DV) 에 기초하여, 비디오 코더 (20/30) 는 인터-뷰 후보로서 지칭되는 새로운 모션 벡터 후보를 결정하고, AMVP 및 스킵/병합 모드들에서 사용을 위해 새로운 모션 벡터 후보를 후보 리스트에 추가할 수 있다.

[0124] 예를 들어, 인터-뷰 후보 도출의 일 예는 인터-뷰 예측된 모션 벡터 후보 (IPMVC) 이며, 또 다른 예는 인터-뷰 디스패리티 모션 벡터 후보 (IDMVC) 이다. IPMVC 는 현재의 블록의 디스패리티 벡터에 의해 참조되는 참조 뷰에서의 참조 블록에 대한 모션 정보 (예컨대, 참조 화상 리스트들에 대한 하나 이상의 참조 인덱스들 및 모션 벡터) 를 포함하며, 여기서, 현재의 블록에 대한 디스패리티 벡터는 상기 예시적인 기법에 기초하여 결정된다. IDMVC 는 디스패리티 모션 벡터로 변환되는 디스패리티 벡터이다.

[0125] 일부 예들에서, 비디오 코더 (20/30) 는 (예컨대, 디스패리티 벡터로부터 오프셋 값을 감산하거나 또는 가산함으로써) 디스패리티 벡터를 시프트시킬 수도 있다. 비디오 코더 (20/30) 는 디스패리티 벡터를 시프트하지 않고 그들과 유사한 기법을 이용하여, 그 시프트된 디스패리티 벡터에 기초하여, 추가적인 IPMVCs 및 IDMVCs 를 도출할 수도 있다. 예를 들어, 비디오 코더 (20/30) 는 (예컨대, 디스패리티 벡터의 x-성분으로부터 고정된 오프셋 값을 감산하거나 또는 가산함으로써) 디스패리티 벡터를 수평으로 한 방향으로 시프트시킬 수도 있다.

[0126] 비디오 코더 (20/30) 는 오프셋 값만큼 시프트된 디스패리티 벡터에 의해 식별되는 참조 뷰에서 참조 블록을 식별할 수도 있다. 오프셋 값만큼 시프트된 디스패리티 벡터에 의해 식별되는 참조 블록에 대한 모션 정보는 (즉, 디스패리티 벡터를 시프트하지 않고 도출되는 IPMVC 에 더해) 추가적인 IPMVC 의 일 예일 수도 있다.

또 다른 예로서, 비디오 코더 (20/30) 는 디스패리티 벡터를 시프트시킬 수도 있으며, 시프트된 디스패리티 벡터를 (즉, 디스패리티 벡터를 시프트하지 않고 도출되는 IDMVC 에 더해) 추가적인 IDMVC 로서 변환할 수 있다.

[0127] 일부 예들에서, 비디오 코더 (20/30) 가 IPMVC 를 도출하기 위해 디스패리티 벡터를 시프트시킬 수도 있는 양은, 비디오 코더 (20/30) 가 IDMVC 를 도출하기 위해 디스패리티 벡터를 시프트시킬 수도 있는 양과 상이할 수도 있다. 다시 말해서, 비디오 코더 (20/30) 는 추가적인 IPMVC 를 도출하기 위해 디스패리티 벡터를, 현재의 블록에 대해, 제 1 오프셋 값만큼 시프트시키고, 추가적인 IDMVC 를 도출하기 위해 디스패리티 벡터를, 현재의 블록에 대해, 상이한 제 2 오프셋 값만큼 시프트시킬 수도 있다.

[0128] 제 1 오프셋 값 또는 제 2 오프셋 값은 수평 시프트된 값 및 수직 시프트된 값을 포함하는 벡터로서 간주될 수도 있다. 예를 들어, IPMVC 를 도출하기 위해 제 1 오프셋 값만큼 디스패리티 벡터를 시프트시키기 위해, 비디오 코더 (20/30) 는 현재의 블록에 대해, 수평 시프트된 값 및 수직 시프트된 값을 포함하는 제 1 벡터로 디스패리티 벡터를 가산하거나 또는 감산할 수도 있다. 이와 유사하게, IDMVC 를 도출하기 위해 제 2 오프셋 값만큼 디스패리티 벡터를 시프트시키기 위해, 비디오 코더 (20/30) 는 현재의 블록에 대해, 수평 시프트된 값 및 수직 시프트된 값을 포함하는 제 2 벡터로, 디스패리티 벡터를 가산하거나 또는 감산할 수도 있다. 본 개시물에서 설명하는 기법들에서, 디스패리티 벡터를 제 1 오프셋 값만큼 시프트시키는데 사용되는 제 1 벡터와, 디스패리티 벡터를 제 2 오프셋 값만큼 시프트시키는데 사용되는 제 2 벡터는 상이한 벡터들일 수도 있다. 일부 예들에서, 제 1 또는 제 2 오프셋 값들에 대한 수직 시프트된 값은 IPMVC 및 IDMVC 를 도출하기 위한 디스패리티 벡터에서 단지 수평 시프트를 획득하기 위해 0 과 동일할 수도 있다.

[0129] 도 6 은 인터-뷰 예측된 모션 벡터 후보 (IPMVC) 의 도출 프로세스의 일 예를 나타낸다. (예컨대, 위에서 설명된 기법들을 이용하여 결정된) 디스패리티 벡터에 의해, 비디오 코더 (20/30) 는 상이한 뷰 (예컨대, 또한 참조 뷰인, 뷰 0 또는 V0) 에서의 대응하는 블록 (142) 을 현재 코딩된 뷰 (뷰 1 또는 V1) 에서의 현재의 PU (140) 에 로케이트시킬 수도 있다. 참조 블록으로서 또한 지칭되는 대응하는 블록 (142) 이 인트라-코딩되지 않고 인터-뷰 예측되지 않고, 그리고 그의 참조 화상이 현재의 PU (140) 의 참조 화상 리스트 (예컨대, 도 6 에 도시된 바와 같은, Ref0, List 0; Ref0, List1; Ref1, List 1) 내에 있는 POC 값을 가지면, 대응하는 블록 (142) 에 대한 모션 정보 (예컨대, 모션 벡터 및 참조 화상 인덱스들 중 적어도 하나) 는 인터-뷰 예측된 모션 벡터로서 사용된다.

[0130] 대응하는 블록 (142) (예컨대, 참조 블록) 에 대한 모션 정보가 IPMVC 인 것으로 간주될 수 있는지 여부에 관해 어떤 요구사항들이 있을 수도 있다. 다시 말해서, IPMVC 가 이용가능한지 여부를 결정하는데 어떤 요구사항들이 있을 수도 있다. 하나의 요구사항은 대응하는 블록 (142) 에 대한 모션 벡터가 시간 모션 벡터이어야 한다는 것일 수도 있다. 시간 모션 벡터는 동일한 뷰에서의 화상을 참조하는 모션 벡터이다. 예를 들어, 도 6 에서, 대응하는 블록 (142) (예컨대, 참조 블록 (142)) 은 뷰 0 내에 있으며, 그리고 대응하는 블록 (142) 이 시간 모션 벡터를 갖기 위해서는, 대응하는 블록 (142) 에 대한 모션 벡터는 대응하는 블록 (142) 과 동일한 뷰 (예컨대, 뷰 0) 에서의 화상들을 참조해야 할 것이다. 도 6 에 예시된 바와 같이, 대응하는 블록 (142) 에 대한 모션 벡터들은, 그들이 대응하는 블록 (142) 과 동일한 참조 뷰 (예컨대, 뷰 0) 에서의 화상들을 참조하기 때문에, 시간 모션 벡터들이다.

[0131] 또 다른 요구사항은, 참조 블록 (예컨대, 대응하는 블록 (142)) 의 시간 모션 벡터가 현재의 블록 (예컨대, 현재의 PU (140)) 에 대한 참조 화상 리스트 (예컨대, RefPicList0 또는 RefPicList1) 에서 화상의 POC 값과 동일한 POC (picture order count) 값을 가진 참조 뷰에서의 화상을 참조해야 한다는 것일 수도 있다. 이해를 돕기 위한 하나의 실례로서, 대응하는 블록 (142) 에 대한 참조 화상 리스트들은 대응하는 블록 (142) 을 인터-예측하는데 사용되는 2개의 화상들을, RefPicList0 에서 제 1 참조 인덱스에 그리고 RefPicList1 에서 제 2 참조 인덱스에 로케이트되는 것으로, 식별할 수도 있다. 대응하는 블록 (142) 을 인터-예측하는데 사용되는 2개의 화상들에 대한 POC 값들은 POC X 및 POC Y 일 수도 있다. 이용가능한 것으로 간주되는 IPMVC 에 대해, 적어도 하나의 참조 화상 리스트 (즉, RefPicList0 또는 RefPicList1) 는 POC X 및 POC Y 와 동일한 POC 값을 가진 화상들을 포함해야 한다. 다시 말해서, 이용가능한 것으로 간주되는 IPMVC 에 대해, 참조 블록 (예컨대, 대응하는 블록 (142)) 에 대한 시간 모션 벡터는 현재의 블록 (예컨대, 현재의 PU (140)) 에 대한 참조 화상 리스트에서 화상의 POC 값과 동일한 POC 값을 가진 참조 뷰 (예컨대, 뷰 0) 에서의 화상을 참조할 필요가 있다.

[0132] 현재의 PU (140) 에 대해, POC 값들 X 및 Y 를 가진 화상들은 RefPicList0 에서 제 1 참조 인덱스 및 제 2 RefPicList1 에서 참조 인덱스에 로케이트되지 않을 수도 있으며, 그러나 대신, RefPicList0 에서 제 3 참조 인

텍스 및 RefPicList1 에서 제 4 참조 인덱스에, 각각 로케이트될 수도 있다. 이 예에서, 대응하는 블록 (142) 의 모션 정보가 인터-뷰 예측된 모션 벡터 후보로서 추가될 때, 비디오 코더 (20/30) 는 RefPicList0 에서 제 3 참조 인덱스 및 RefPicList1 에서 제 4 참조 인덱스를 참조 인덱스들로서 추가할 수도 있다.

[0133] 게다가, 비디오 코더 (20/30) 는 디스패리티 벡터를 인터-뷰 디스패리티 모션 벡터로 변환할 수도 있으며, 이 인터-뷰 디스패리티 모션 벡터는 이용가능할 때 병합 후보 리스트에 IPMVC 과는 상이한 위치에 추가되거나 또는 AMVP 후보 리스트에 IPMVC 와 동일한 위치에 추가될 수도 있다. IPMVC 및 인터-뷰 디스패리티 모션 벡터 후보 (IDMVC) 양쪽은 이 상황에서 '인터-뷰 후보' 로서 지칭될 수도 있다.

[0134] AMVP 모드에서, 참조 화상 인덱스가 시간 참조 화상을 참조하면 (예컨대, 목표 참조 인덱스가 시간 모션 벡터에 대응하면), 비디오 코더 (20/30) 는 IPMVC 를 도출할 수도 있으며 IDMVC 를 도출하지 않을 수도 있다. 예를 들어, 비디오 코더 (20/30) 는 디스패리티 벡터에 의해 로케이트된 현재의 PU (즉, 블록 (140)) 의 대응하는 블록 (즉, 블록 (142)) 에서 모션 벡터를 체크함으로써 IPMVC 를 찾아낼 수도 있다. 이용불가능하면 (즉, 어떤 모션 벡터도 없으면), 비디오 코더 (20/30) 는 제로 모션 벡터 후보를 후보 리스트에 추가할 수도 있다. AMVP 모드에서, 참조 화상 인덱스가 인터-뷰 참조 화상을 참조하면 (예컨대, 목표 참조 인덱스가 디스패리티 모션 벡터에 대응하면), 비디오 코더 (20/30) 는 IPMVC 를 도출하지 않을 수도 있으며, 그러나 대신, 디스패리티 벡터를 인터-뷰 디스패리티 모션 벡터로 변환할 수도 있으며 그 변환된 디스패리티 벡터를 후보 리스트에 추가할 (즉, 인터-뷰 디스패리티 모션 벡터를 후보 리스트에 IDMVC 로서 추가할) 수도 있다.

[0135] 다시 말해서, 현재의 블록이 AMVP 모드에서 인터-예측되면, 비디오 코더 (20/30) 는 적어도 하나의 참조 화상 인덱스가 시간 참조 화상 또는 인터-뷰 참조 화상을 참조하는지 여부에 기초하여, IPMVC 또는 IDMVC 를 도출할지 여부를 결정할 수도 있다. 예를 들어, AMVP 모드에서, 비디오 인코더 (20) 는 비디오 디코더 (30) 가 인터-예측에 사용될 참조 화상을 식별하는데 사용하는 참조 화상 인덱스를 시그널링한다. 이러한 방법으로, 비디오 코더 (20/30) 는 참조 화상 인덱스가 시간 참조 화상 또는 인터-뷰 참조 화상을 참조할지 여부를 결정할 수도 있다. 시간 참조 화상은 동일한 뷰에서의 참조 화상이며, 인터-뷰 참조 화상은 또 다른, 상이한 뷰에서의 참조 화상이다. 일부 예에서, 참조 화상 인덱스가 시간 참조 화상을 참조한다고 비디오 코더 (20/30) 가 결정하면, 비디오 코더 (20/30) 는 IPMVC 를 도출할 수도 있으며, 참조 화상 인덱스가 인터-뷰 참조 화상을 참조한다고 비디오 코더 (20/30) 가 결정하면, 비디오 코더 (20/30) 는 IDMVC 를 도출할 수도 있다.

[0136] 병합/스킵 모드에서, 비디오 코더 (20/30) 는 IPMVC 를, 이용가능하면, 병합 후보 리스트에서 모든 공간 및 시간 병합 후보들 앞에 항상 삽입할 수도 있다. 비디오 코더 (20/30) 는 예를 들어, 위에서 도 4 에서 나타난 바와 같이, IPMVC 를 A₀ 로부터 도출된 공간 병합 후보 앞에 삽입할 수도 있다. 3D-HTM 소프트웨어의 일부 구현예에서, 인터-뷰 예측된 모션 벡터 또는 인터-뷰 디스패리티 모션 벡터는 AMVP 후보 리스트에서 모든 공간 및 시간 후보들 앞에 항상 삽입된다.

[0137] 위에서 소개한 바와 같이, 비디오 코더 (20/30) 는 3D-HEVC 에서 병합 후보 리스트를 구성할 수도 있다. 비디오 코더 (20/30) 는 예를 들어, 위에서 설명된 NBDV 기법들을 이용하여 디스패리티 벡터를 먼저 도출한다. 이러한 디스패리티 벡터가 발견되지 않으면, 비디오 코더 (20/30) 는 제로 디스패리티 벡터를 이용할 수도 있다. 이 디스패리티 벡터에 의해, 비디오 코더 (20/30) 는 병합 후보 리스트를 다음과 같이 구성한다. 비디오 코더 (20/30) 는 위에서 설명한 바와 같이 IPMVC 를 도출하며, 이용가능하면, IPMVC 를 병합 리스트에 삽입한다.

[0138] 비디오 코더 (20/30) 는 공간 병합 후보들에 대한 도출 프로세스 및 IDMVC 삽입을 구현하도록 구성될 수도 있다. 비디오 코더 (20/30) 는 예를 들어, 공간 이웃하는 PUs 의 모션 정보를 다음의 순서로 체크할 수도 있다: A₁, B₁, B₀, A₀, 또는 B₂. 비디오 코더 (20/30) 는 다음 프로시저들에 의해 속박된 프루닝을 수행할 수도 있다:

[0139] - 이용가능하고 IPMVC 와 상이하면 A₁ 을 후보 리스트에 삽입한다.

[0140] - B₁ 및 A₁/IPMVC 가 동일한 모션 벡터들 및 동일한 참조 인덱스들을 가지면, B₁ 은 후보 리스트에 삽입되지 않는다. 그렇지 않고, 이용가능하면 B₁ 을 후보 리스트에 삽입한다.

[0141] - B₀ 및 B₁ 이 동일한 모션 벡터들 및 동일한 참조 인덱스들을 가지면, B₀ 는 후보 리스트에 삽입되지 않는다. 그렇지 않고, 이용가능하면 B₀ 를 후보 리스트에 삽입한다.

- [0142] - IDMVC 는 위에서 설명된 프로시저에 의해 도출된다. 이용가능하고 A_1 및 B_1 로부터 도출된 후보들과 상이하면, 후보 리스트에 삽입된다.
- [0143] - A_0 및 A_1 이 동일한 모션 벡터들 및 동일한 참조 인덱스들을 가지면, A_0 는 후보 리스트에 삽입되지 않는다. 그렇지 않고, 이용가능하면 A_0 를 후보 리스트에 삽입한다.
- [0144] - 다음 조건들 양쪽이 만족되지 않을 때 B_2 가 후보 리스트에 삽입된다:
- [0145] - B_2 및 B_1 또는 B_2 및 A_1 은 동일한 모션 벡터들 및 동일한 참조 인덱스들을 갖는다.
- [0146] - A_1 , B_1 , B_0 , A_0 및 IPMVC 로부터 도출된 4개의 공간 병합 후보들 모두가 후보 리스트에 포함된다.
- [0147] 비디오 코더 (20/30) 는 또한 시간 병합 후보에 대한 도출 프로세스를 구현하도록 구성될 수도 있다. 이 도출 프로세스는 예를 들어, 위에서 간단히 설명된 HEVC 에 정의된 프로시저와 동일한 프로시저일 수도 있다.
- [0148] 비디오 코더 (20/30) 는 또한 결합된 양방향-예측 병합 후보들에 대한 도출 프로세스를 구현하도록 구성될 수도 있다. 예를 들어, 상기 2개의 단계들로부터 도출된 후보들의 총 개수가 후보들의 최대 개수 미만이면, 비디오 코더 (20/30) 는 10CandIdx 및 11CandIdx 의 사양을 제외하고, HEVC 에서 정의된 프로세스와 동일한 프로세스를 수행할 수도 있다. combIdx, 10CandIdx 및 11CandIdx 사이의 관계는 다음의 테이블에 정의된다:

표 2

3D-HEVC 에서 10CandIdx 및 11CandIdx 의 사양

com bId x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
10C and Idx	0	1	0	2	1	2	0	3	1	3	2	3	0	4	1	4	2	4	3	4
11C and Idx	1	0	2	0	2	1	3	0	3	1	3	2	4	0	4	1	4	2	4	3

- [0150] 비디오 코더 (20/30) 는 또한 제로 모션 벡터 병합 후보들에 대한 도출 프로세스를 구현하도록 구성될 수도 있다. 비디오 코더 (20/30) 는 예를 들어, HEVC 에 정의된 프로시저와 동일한 프로시저를 구현할 수도 있다.
- [0151] 최신 소프트웨어에서, 병합 (MRG) 리스트에서의 후보들의 총 개수는 최고 6 개이며, 6 으로부터 감소한 MRG 후보들의 최대 개수를 규정하기 위해 구문 엘리먼트 "five_minus_max_num_merge_cand" 가 슬라이스 헤더로 시그널링된다. five_minus_max_num_merge_cand 는 0 내지 5 의 범위 내인 점에 유의해야 한다. 위에서 언급한 바와 같이, WD10 에서, MRG 리스트에서 후보의 총 개수는 최고 5개이다.
- [0152] 위에서 설명된 방법들에 추가하여, 디스패리티 벡터로부터의 병합 후보들 도출에 관한 다른 제안들에 따르면, 다음의 기법이 또한 2013년 12월 4일 현재, http://phenix.it-sudparis.eu/jct2/doc_end_user/current_document.php?id=237 로부터 입수가 가능한, 2012년 10월 15일, Zhang 등의, "3D-CE5.h: Merge candidates derivation from disparity vector" 란 제목으로 된 JCT3V-B0048 에서 제안되었다. 참조 뷰에서 디스패리티 벡터에 의해 식별되는 대응하는 블록 (즉, 참조 블록) 이 현재의 방법처럼 IPMVC 를 도출하는데 사용된다. 대응하는 블록을 포함하는 PU 의 그의 좌측 및 우측 PUs 가 로케이트되며, 적용가능하면, 2개의 시프트된 디스패리티 벡터들이 도출된다. 좌측 시프트된 디스패리티 벡터 (LDV): 수평 성분에 대해 좌측 PU 의 폭 만큼 디스패리티 벡터를 감소한다. 우측 시프트된 디스패리티 벡터 (RDV): 수평 성분에서 대응하는 블록을 포함하는 PU 의 폭 만큼 디스패리티 벡터를 가산한다. IPMVCs 를 도출하는데 LDV/RDV 의 용도는 현재의 3D-HEVC 에서 디스패리티 벡터와 동일하다. 도출된 IPMVC 가 이용가능하면, 병합 후보 리스트에 추가되며, 그렇지 않으면, 시프트된 디스패리티 벡터는 병합 후보 리스트에 추가될 디스패리티 모션 벡터로 변환된다.
- [0153] HEVC 기반의 멀티뷰/3DV 코딩에서 모션 관련 기술들의 현재의 설계는 도출된 디스패리티 벡터가 충분히 정확하

지 않을 지도 모른다는 사실로 인해, 여러 잠재적인 문제들을 갖는다. 일 예로서, 부정확한 디스패리티 벡터는 부정확한 인터-뷰 예측된 모션 벡터 후보들 및 인터-뷰 디스패리티 모션 벡터 후보들을 초래할 수도 있다.

또 다른 예로서, IPMVCs 를 도출하기 위해 참조 뷰에서의 대응하는 블록의 좌측/우측 PUs 의 이용은, 좌측/우측 PUs 에의 액세스가 대응하는 블록으로부터 변수인 그의 폭 만큼 시프트시킬 필요가 있기 때문에, 하드웨어 구현에 친화적이지 않을 수도 있다. 또 다른 예로서, 재복제된 후보들을 프루닝하기 위해, 너무 많은 복잡성을 도입하는, 모든 참조 인덱스들 및 모션 벡터들이 체크되는 것이 요구된다.

[0154] 본 개시물은 인터-뷰 예측된 모션 벡터들 및 인터-뷰 디스패리티 모션 벡터들의 정확도 뿐만 아니라 디스패리티 벡터 정확도를 향상시킬 수도 있는 여러 기법들을 소개한다. 본 개시물의 일 기법에 따르면, 비디오 코더 (20/30) 는 하나의 슬라이스/화상/시퀀스 내 모든 PUs 에 대해 고정된 오프셋 값으로 시프트된 디스패리티 벡터들로부터의 참조 뷰로부터 더 많은 IPMVCs 를 도출할 수도 있다. 일 예에서, 비디오 코더 (20/30) 는 디스패리티 벡터를 $-4 * M$ (M 은 1, 2, 4, 8 와 같은, 양의 정수임) 및/또는 (M 개의 픽셀들에 대응하는) $4 * M$ 만큼 수평으로 시프트시킬 수도 있다. 시프트된 디스패리티 벡터에 의해, 비디오 코더 (20/30) 는 대응하는 블록을 포함하는 PU 의 좌측 및/또는 우측면에 로케이트된 PUs 를 이용하여, 대응하는 블록으로부터의 IPMVC 발생 프로세스와 유사하게 IPMVCs 를 발생할 수도 있다.

[0155] 본 개시물의 또 다른 기법에 따르면, 비디오 코더 (20/30) 는 시프트된 디스패리티 벡터들에 의해 더 많은 IDMVCs 를 변환할 수 있다. 일 예에서, 시프트되는 값 (즉, 오프셋 값) 은 수평으로 $-4 * N$ (N 은 1, 2, 4, 8 과 같은, 양의 정수임) 및/또는 (N 개의 픽셀들에 대응하는) $4 * N$ 이다. 일 예에서는, 더욱이, IPMVC 및 IDMVC 를 도출하기 위한 시프트되는 값 (즉, 오프셋 값) 은 상이할 수 있으며, 즉, N 은 M 과 상이하다.

[0156] 본 개시물의 또 다른 기법에 따르면, 디스패리티 벡터에 기초한 각각의 변화 (mutation) (즉, 디스패리티 벡터의 각각의 시프팅) 을 위해, 비디오 코더 (20/30) 는 오직 하나의 추가적인 후보를 발생할 수도 있으며, 후보는 제 1 시프트된 디스패리티 벡터에 기초한 IPMVC 또는 제 2 시프트된 디스패리티 벡터에 기초한 IDMVC 로부터 유래한다. IPMVC 가 이용가능하면, 비디오 코더 (20/30) 는 추가적인 후보로서 그것을 이용할 수도 있다. IPMVC 가 병합 후보 리스트에 추가되면, 비디오 코더 (20/30) 는 IDMVC 를 변환하지 않을 수도 있다. 일 예에서, IDMVC 가 이용가능하면, 비디오 코더 (20/30) 는 IPMVC 를 도출하지 않을 수도 있다. 또 다른 예에서, 그들 중 2개는, 이용가능하면, 추가적인 후보들로서 도출될 수 있다.

[0157] 다시 말해서, 현재의 블록이 병합 모드 또는 스킵 모드에서 인터-예측되면, 비디오 코더 (20/30) 는 IPMVC 의 이용가능성을 결정할 수도 있으며, 여기서, IPMVC 는 현재의 블록에 대해 오프셋 값만큼 시프트된 디스패리티 벡터에 의해 참조되는 참조 뷰에서의 참조 블록에 대한 모션 정보 (예컨대, 모션 벡터 및 하나 이상의 참조 인덱스들) 이다. 예를 들어, 비디오 코더 (20/30) 는 참조 블록에 대한 모션 벡터가 현재의 블록에 대한 참조 화상 리스트에서의 화상의 POC 값과 동일한 POC (picture order count) 값을 가진 참조 뷰에서의 화상을 참조하는 시간 모션 벡터이면, IPMVC 가 이용가능하다고 결정할 수도 있다.

[0158] 이러한 예에서, 비디오 코더 (20/30) 는 (아래에서 설명되는 프루닝을 조건으로 하여) 가능한 후보 리스트에의 포함을 위해 IPMVC 를 도출할 수도 있으며, IDMVC 의 도출을 회피할 수도 있다 (즉, IDMVC 의 도출을 회피할 수도 있다). 예를 들어, 이 예에서, 비디오 코더 (20/30) 는 현재의 블록이 병합 또는 스킵 모드에서 인터-예측되면, IDMVC 보다 IPMVC 을 우선적으로 처리할 수도 있으며, 여기서, IPMVC 가 이용가능하면, 비디오 코더 (20/30) 는 IPMVC 를 도출할 수도 있으며 IDMVC 를 도출하지 않을 수도 있다 (즉, IDMVC 의 도출을 회피할 수도 있다).

[0159] 본 개시물의 또 다른 기법에 따르면, 디스패리티 벡터에 기초한 각각의 변화에 대해, IPMVC 의 도출에 사용되는 제 1 시프트된 디스패리티 벡터 및 IPMVC 의 변환에 사용되는 제 2 시프트된 디스패리티 벡터는 상이한 오프셋 값들을 가질 수 있으며, 여기서, 오프셋 값들은 비디오 코더 (20/30) 가 디스패리티 벡터를 시프트시키는 양을 나타낸다. 예를 들어, 제 1 시프트된 디스패리티 벡터에 대한 오프셋 값은 16 이며, 제 2 시프트된 디스패리티 벡터에 대한 오프셋은 4 이다. 일부 예들에서, 시프트된 디스패리티 벡터들 양쪽은 수평 방향 및 동일한 방향이어야 하며, 오프셋들의 부호 값들이 동일하다는 것을 의미한다. 일 예에서, 단지 2개의 변화들, 즉 하나는 좌측으로의 변화 그리고 하나는 우측으로의 변화가 존재한다. 이의 대안으로, 오직 하나의 좌측으로의 변화가 존재한다. 이의 대안으로, 오직 하나의 우측으로의 변화가 존재한다. 본 개시물의 또 다른 기법에 따르면, 후보들의 각각이 후보 리스트의 임의의 위치에 삽입될 수도 있다.

[0160] 일 예로서, 비디오 코더 (20/30) 가 IPMVC 를 도출하면, IPMVC 에 대한 모션 정보는 현재의 블록에 대해 제 1 오프셋 값만큼 시프트된 디스패리티 벡터에 의해 참조되는 참조 뷰에서의 참조 블록에 대한 모션 벡터 (예컨대,

시간 모션 벡터) 를 포함할 수도 있다. 또한, IPMVC 의 모션 벡터는 그의 POC 값이 현재의 블록의 참조 화상 리스트에서 식별된 화상의 POC 값과 동일한 동일한 뷰 (예컨대, 시간 참조 화상) 에서의 화상을 참조할 수도 있다. 비디오 코더 (20/30) 가 IDMVC 를 도출하면, IDMVC 는 현재의 블록에 대해, 상이한 제 2 오프셋 값만큼 시프트된 디스패리티 벡터일 수도 있다. 일부 예들에서, 비디오 코더 (20/30) 는 제 1 오프셋 값을 디스패리티 벡터에 가산하거나 또는 디스패리티 벡터로부터 제 1 오프셋 값을 감산하여, 제 1 오프셋 값만큼 시프트된 디스패리티 벡터를 결정할 수도 있다. 일부 예에서, 비디오 코더 (20/30) 는 제 2 오프셋 값을 디스패리티 벡터에 가산하거나 또는 디스패리티 벡터로부터 제 2 오프셋 값을 감산하여, 제 2 오프셋 값만큼 시프트된 디스패리티 벡터를 결정할 수도 있다.

[0161] 본 개시물의 또 다른 기법에 따르면, 유형-기반의 프루닝 (pruning) 이 제 1 디스패리티 벡터로부터 도출된 것을 심지어 포함한, 새롭게 추가된 모션 벡터 후보의 각각에 적용될 수도 있다. 비디오 코더 (20/30) 는 단지 참조 화상 유형들 (인터-뷰 참조 화상들 또는 시간 참조 화상들) 이 동일할 때에만, 2개의 후보들에 대해 프루닝 프로세스를 적용할 수도 있다. 이 경우, RefPicList0 또는 RefPicList1 에 대응하는 참조 화상의 유형이 비교될 후보의 유형과 상이할 때, 참조 인덱스들과 모션 벡터들의 어떤 비교도 요구되지 않는다 (즉, 비디오 코더 (20/30) 는 제 1 후보에 의해 참조되는 참조 화상의 유형이 제 2 후보와 상이하면, 참조 인덱스들과 모션 벡터들의 비교를 수행하지 않을 수도 있다). 비디오 코더 (20/30) 는 단지 그들이 IDMVCs 또는 IPMVCs 양쪽 일 때에만 2개의 인터-뷰 후보들에 대해 프루닝 프로세스를 적용할 수도 있다.

[0162] 예를 들어, 시프트된 디스패리티 벡터들에 기초하여 IPMVC 및 IDMVC 를 도출하는 것에 더해서, 비디오 코더 (20/30) 는 디스패리티 벡터를 시프트하지 않고 현재의 블록에 대한 디스패리티 벡터에 기초하여 IPMVC 및 IDMVC 를 도출할 수도 있다. 일 예로서, 제 1 IPMVC 가 현재의 블록에 대해 오프셋 값만큼 시프트된 디스패리티 벡터에 의해 참조되는 제 1 참조 블록에 대한 모션 벡터라고 가정한다. 일부 예들에서, 비디오 코더 (20/30) 는 제 2 IPMVC 를 후보 리스트에 포함시킬 수도 있으며, 여기서 제 2 IPMVC 는 디스패리티 벡터를 시프트하지 않고 제 2 디스패리티 벡터에 의해 참조되는 참조 뷰에서의 참조 블록에 대한 모션 벡터이다.

[0163] 비디오 코더 (20/30) 는 모션 벡터, 참조 화상 인덱스, 및 참조 화상 리스트와 같은, 제 1 IPMVC 의 모션 정보와 제 2 IPMVC 의 모션 정보를 비교할 수도 있다. 일부 예들에서, 비디오 코더 (20/30) 는 제 1 IPMVC 의 모션 정보를, 단지 제 2 IPMVC 의 모션 정보와 비교하고, 그리고 후보 리스트에서의 어떤 다른 후보들과는 비교하지 않을 수도 있다. 제 1 IPMVC 의 모션 정보 및 제 2 IPMVC 의 모션 정보가 동일하다고 (예컨대, 모션 벡터들이 동일하거나, 또는 모션 벡터들 및 참조 화상 인덱스들 양쪽이 동일하다고) 비디오 코더 (20/30) 가 결정하면, 비디오 코더 (20/30) 는 제 1 IPMVC 가 후보 리스트에 포함되어 있는 것이 아니라고 결정할 수도 있다. 제 1 IPMVC 의 모션 정보 및 제 2 IPMVC 의 모션 정보가 상이하다고 (예컨대, 모션 벡터들이 상이하거나, 모션 벡터들이 동일하지만 참조 화상 인덱스들이 상이하거나, 모션 벡터들이 상이하지만 참조 화상 인덱스들이 상이하거나, 또는 모션 벡터들 및 참조 화상 인덱스들 양쪽이 상이하다고) 비디오 코더 (20/30) 가 결정하면, 비디오 코더 (20/30) 는 제 1 IPMVC 가 후보 리스트에 포함되어 있는 것이라고 결정할 수도 있다.

[0164] 일부 예들에서, 상기 예의 제 2 IPMVC 는 후보 리스트에 항상 포함될 수도 있다. 이러한 예들에서, (예컨대, 제 1 IPMVC 와 제 2 IPMVC 를 그리고 일부 예들에서는, 단지 제 2 IPMVC 와 비교하는) 프루닝 기법은 제 1 IPMVC 를 후보 리스트에 포함시킬지 여부를 결정할 수도 있다.

[0165] 본 개시물의 또 다른 기법에 따르면, 상기 기법들은 디스패리티 벡터가 NBDV 이후에 이용가능할 때에 단지 적용할 수도 있다. 어떤 디스패리티 벡터도 NBDV 이후에 발견되지 않으면, 비디오 코더 (20/30) 는 추가적인 IPMVCs 및 IDMVCs 를 도출하기 위해 디스패리티 벡터를 시프트시키는 상기 기법들을 적용하지 않을 수도 있다 (상기가 적용되지 않을 것이다). 일 예에서, 상기 방법들로 도출된 IPMVCs 가 항상 적용된다. 그러나, 새롭게 도출된 IDMVCs 는 단지 디스패리티 벡터가 NBDV 와 함께 이용가능할 때에만 병합 후보 리스트에 삽입된다.

[0166] 더욱이, 병합 및 스킵 모드에서, 비디오 코더 (20/30) 는 IPMVC 가 이용가능하면, 현재의 블록에 대해 제 1 오프셋 값만큼 시프트된 디스패리티 벡터로부터 IPMVC 를 도출할 수도 있으며, IPMVC 가 이용불가능한 것으로 결정되면, 현재의 블록에 대해, 상이한 제 2 오프셋 값만큼 시프트된 디스패리티 벡터로부터 IDMVC 를 도출할 수도 있다. 또한, AMVP 모드에서, 비디오 코더 (20/30) 는 참조 화상 인덱스에 기초하여, IPMVC 또는 IDMVC 를 도출할 수도 있다. 따라서, 일부 예들에서, 단지 IPMVC 또는 IDMVC 중 하나만이 후보 리스트에 포함된다.

[0167] 위에서 소개된 여러 기법들은 비디오 코더 (20/30) 에 의해 구현될 수도 있다. 위에서 소개된 기법들의 여

러 예시적인 구현예들이 이하 설명될 것이다. 디스패리티 벡터는 HEVC 방법들에 따라서 도출될 수도 있다. 그러나, 더 많은 후보들이 스킵/병합 모드들 또는 AMVP 모드에 대한 디스패리티 벡터에 기초하여 도출될 수도 있다.

- [0168] 본 개시물의 일 양태에 따르면, 비디오 코더 (20/30) 는 IPMVCs 및 IDMVCs 에 대한 디스패리티 벡터의 변화를 수행하도록 구성될 수도 있다. 디스패리티 벡터에 기초하여, 디스패리티 벡터들의 2개의 변화들이 IPMVCs 및 IDMVCs 를 도출하는데 사용된다. IPMVCs 의 도출을 위한 좌측 및/또는 우측 시프트된 디스패리티 벡터 (LDV 및/또는 RDV) 의 변화는 디스패리티 벡터를 수평 성분에서 -16 및/또는 16 만큼 가산함으로써 달성된다. IDMVCs 의 도출을 위한 좌측 및/또는 우측 시프트된 디스패리티 벡터 (LDV 및/또는 RDV) 의 변화는 디스패리티 벡터를 수평 성분에서 -4 및/또는 4 만큼 가산함으로써 달성된다.
- [0169] 비디오 코더 (20/30) 는 변화된 디스패리티 벡터들로부터 IPMVCs 를 도출할 수도 있다. 시프트된 디스패리티 벡터들에 의해, 2개의 추가적인 IPMVCs 가 현재의 방법에서 IPMVC 와 동일한 방법으로 도출될 수도 있다. LDV 및 RDV 에 의한 2개의 새롭게 도출된 IPMVCs 는 좌측 IPMVC 및 우측 IPMVC 로 각각 표시된다. 간결성을 위해, LDV 및 RDV 로부터 도출된 IPMVCs (시간 모션 후보들) 는 LTMV 및 RTMV 로 각각 명명된다.
- [0170] 비디오 코더 (20/30) 는 변화된 디스패리티 벡터들로부터 IDMVCs 를 도출할 수도 있다. 시프트된 디스패리티 벡터들에 의해, 2개의 추가적인 IDMVCs 가 현재의 방법에서 IDMVC 와 동일한 방법으로 도출될 수도 있다. LDV 및 RDV 에 의한 2개의 새롭게 도출된 IDMVCs 는 좌측 IDMVC 및 우측 IDMVC 로 각각 표시된다. 간결성을 위해, LDV 및 RDV 로부터 도출된 IDMVCs (디스패리티 모션 벡터 후보들) 는 LDMV 및 RDMV 로 각각 명명된다. 변화된 디스패리티 벡터, 즉, LDV 또는 RDV 에 기초하여, 하나 및 오직 하나의 후보, 즉 LTMV (RTMV) 또는 LDMV (RDMV) 가 도출된다. 이러한 후보는 LDV 로부터 도출될 때는 LMV, 또는 RDV 로부터 도출될 때는 RMV 로서 표시된다. LMV 는 다음과 같이 도출될 수도 있다: LTMV 가 이용가능하면: LMV 는 LTMV 로 설정되며; 그렇지 않으면, LMV 는 LDMV 로 설정된다. RMV 는 다음과 같이 도출될 수도 있다: RTMV 가 이용가능하면: RMV 는 RTMV 로 설정되며; 그렇지 않으면, RMV 는 RDMV 로 설정된다.
- [0171] 비디오 코더 (20/30) 는 IPMVCs 및 IDMVCs 가 디스패리티 벡터들을 시프트시킴으로써 발생된 이후에 다음 체크 및 프루닝 프로세스를 적용할 수도 있다. 새롭게 도출된 좌측 IPMVC 는 이용가능하면, 변화 없이 디스패리티 벡터로 도출된 IPMVC, A₁ 또는 B₁ 로부터의 공간 병합 후보와 순서대로 비교된다. 이들 3개의 후보들 중 임의의 후보와 동일하면, 병합 후보 리스트로부터 제거된다. 그렇지 않으면, 병합 후보 리스트에 모든 공간 병합 후보들 이후, 따라서, 시간 병합 후보 이전에 추가된다.
- [0172] 좌측 IPMVC 가 병합 후보 리스트에 추가되지 않으면, 새롭게 도출된 좌측 IDMVC 는 이용가능하면, A₁ 또는 B₁ 로부터의 공간 병합 후보와 순서대로 비교된다. 이들 2개의 후보들 중 어느 하나와 동일하면, 병합 후보 리스트로부터 제거된다. 그렇지 않으면, 이용가능하면, 병합 후보 리스트에 좌측 IPMVC 와 동일한 위치에, 즉, 모든 공간 병합 후보들 이후에, 따라서, 시간 병합 후보 이전에 추가된다.
- [0173] 새롭게 도출된 우측 IPMVC 는 이용가능하면, 변화 없이 디스패리티 벡터로 도출된 IPMVC, A₁ 또는 B₁ 로부터의 공간 병합 후보, 좌측 IPMVC 와 순서대로 비교된다. 이것이 이들 4개의 후보들 중 임의의 후보와 동일하면, 병합 후보 리스트로부터 제거된다. 그렇지 않으면, 병합 후보 리스트에 모든 공간 병합 후보들 및 좌측 IPMVC 또는 좌측 IDMVC 이후에 추가된다.
- [0174] 우측 IPMVC 가 병합 후보 리스트에 추가되지 않으면, 새롭게 도출된 우측 IDMVC 는 이용가능하면, A₁ 또는 B₁ 으로부터의 공간 병합 후보와 순서대로 비교된다. 이것이 이들 2개의 후보들 중 어느 하나와 동일하면, 병합 후보 리스트로부터 제거된다. 그렇지 않으면, 이것은 이용가능하면, 병합 후보 리스트에 우측 IPMVC 와 동일한 위치에, 즉, 모든 공간 병합 후보들 및 좌측 IPMVC 또는 좌측 IDMVC 이후에 추가된다.
- [0175] 하나의 대안예에서, 단지 2개의 후보들이 동일한 유형을 갖는, 즉, 그들이 디스패리티 모션 벡터들이거나 또는 그들이 시간 모션 벡터들일 때에만, 비교된다. 예를 들어, A₁ 으로부터 도출된 후보가 시간 참조 화상을 가지면, 그것과 좌측/우측 IDMVC 사이의 비교가 필요되지 않는다.
- [0176] 비디오 코더 (20/30) 는 위에서 설명된 여러 양태들의 대안들을 구현하도록 구성될 수도 있다. 예를 들어, 비디오 코더 (20/30) 는 새롭게 도출된 후보들에 대한 삽입 위치들을 결정하도록 구성될 수도 있다. 새롭게 도출된 후보들이 모든 공간 병합 후보들 이후에 삽입되고 LDV 로부터 도출된 후보 (IPMVC 또는 IDMVC) 가 RDV 로부터 도출된 것보다 선행하는, 위에서 설명된 예와는 대조적으로, 다음 위치들 및 순서들이 다음과 같이 적용

될 수도 있다:

- [0177] ° 공간 병합 후보들의 서브세트 이후.
- [0178] ° 이의 대안으로, 모든 공간, 시간 후보들 및 디스패리티 벡터에 의해 도출된 후보들 이후, 따라서 결합된 후보들 이후.
- [0179] ° 이의 대안으로, $-4*M$ 와 동일한 시프트된 값으로 도출된 IPMVC 가 이용가능하면, $-4*N$ 와 동일한 시프트된 값으로 도출된 IDMVC 이전에 삽입된다. $4*M$ 와 동일한 시프트된 값으로 도출된 IPMVC 가, 이용가능하면, $4*N$ 와 동일한 시프트된 값으로 도출된 IDMVC 이전에 삽입된다.
- [0180] ° 이의 대안으로, 시프트된 값들로 도출된 모든 IPMVCs 가 리스트에, 디스패리티 벡터로부터 도출된 IPMVC 의 다음에, 따라서 모든 공간 병합 후보들 이전에 삽입된다.
- [0181] ° 이의 대안으로, 시프트된 값들로 도출된 모든 IDMVCs 가 리스트에, 디스패리티 벡터로부터 도출된 IDMVC 의 다음에, 따라서 A_0 및 B_2 로부터 도출된 공간 병합 후보들 이전에 삽입된다.
- [0182] ° 하나의 대안예에서, 새롭게 추가된 IPMVCs 및/또는 IDMVCs 는 후보 리스트에 임의의 위치에 그리고 임의의 순서로 삽입될 수 있다.
- [0183] ° 더욱이, RDVs 로부터 도출된 후보는 LDVs 로부터 도출된 것보다 선행한다.
- [0184] ° 이의 대안으로, LDVs 및 RDVs 로부터 도출된 후보가 연속적으로 삽입되지 않을 수도 있다. 대신, LDV 로부터 도출된 후보는 최종 공간 병합 후보 직후에 그리고 시간 병합 후보 직전에 삽입되지만, RDV 로부터 도출된 후보는 시간 병합 후보 직후에 삽입될 수도 있으며, 반대의 경우도 마찬가지이다.
- [0185] 비디오 코더 (20/30) 는 위에서 설명된 프루닝 프로세스와는 다른 프루닝 프로세스를 수행하도록 구성될 수도 있다. 예를 들어, 비디오 코더 (20/30) 는 다음 프루닝 프로세스를 적용하도록 구성될 수도 있다.
- [0186] ° 단지 하나 이상의 선택적 공간 후보들 (예컨대, A_1) 이 새롭게 도출된 후보들과 비교된다.
- [0187] a. 예를 들어, 새롭게 도출된 후보들 (LMV 및/또는 RMV) 은 단지 A_1 로부터의 공간 병합 후보와 비교된다.
- [0188] b. 예를 들어, 새롭게 도출된 후보들 (LMV 및/또는 RMV) 은 단지 B_1 으로부터의 공간 병합 후보와 비교된다.
- [0189] ° 이의 대안으로, 새롭게 도출된 후보들은 임의의 공간 또는 시간 후보와 비교되지 않는다.
- [0190] ° 게다가, 디스패리티 벡터로부터 도출된 것들을 제외한, 새롭게 추가된 모션 벡터 후보들은 중복들을 피하기 위해 서로 비교되어야 한다.
- [0191] ° 이의 대안으로, 디스패리티 벡터로부터 도출된 것들을 포함한, 새롭게 추가된 모션 벡터 후보들은 중복들을 피하기 위해 서로 비교되어야 한다.
- [0192] ° 이의 대안으로, 새롭게 추가된 모션 벡터 후보들은 단지 중복들을 피하기 위해 디스패리티 벡터로부터 도출된 것들과 비교되며 그들은 서로 비교되지 않는다.
- [0193] ° 하나의 대안예에서, 새롭게 추가된 IDMVCs 및/또는 IPMVCs 는 가용 병합 후보들의 모두 또는 서브세트와 비교되어야 한다.
- [0194] ° 이의 대안으로, 게다가, RDV 또는 LDV (LMV 및/또는 RMV) 로부터의 도출된 후보가 디스패리티 모션 벡터이면, IDMVC 와 전혀 비교되지 않는다.
- [0195] ° 하나의 대안예에서, 시프트된 디스패리티 벡터들 ($-4*M/4*M$) 로 로케이트된 참조 블록들 및 디스패리티 벡터로 로케이트된 참조 블록이 동일한 PU 에 속할 때, 새롭게 추가된 IPMVCs 와 디스패리티 벡터로부터 도출된 것 사이의 프루닝 프로세스가 스킵될 수도 있으며 새롭게 추가된 IPMVCs 가 병합 후보 리스트로부터 제거된다.
- [0196] a. 하나의 대안예에서, 더욱이, 시프트된 디스패리티 벡터들 ($-4*N/4*N$) 로부터 도출된 IDMVC 가 병합 후보 리스트에 추가되지 않는다.
- [0197] b. 또 다른 대안적인 예에서, 더욱이, 시프트된 디스패리티 벡터들 ($-4*N/4*N$) 로부터 도출된

IDMVC 가 병합 후보 리스트에 추가되지 않는다.

- [0198] 프루닝 프로세스의 일부로서, 비디오 코더 (20/30) 는 위에서 소개한 바와 같이 하나 이상의 비교들을 수행할 수도 있다. 일 예에서, 다음의 (최고 5개의) 모션 벡터 후보 쌍들 (그들 양쪽이 이용가능하면) 이 비교된다:
- [0199] LTMV 가 이용가능할 때, (LMV, 디스패리티 벡터로부터 도출된 IPMVC);
- [0200] (LMV, B_1 로부터의 공간 병합 후보);
- [0201] RTMV 가 이용가능할 때, (RMV, 디스패리티 벡터로부터 도출된 IPMVC);
- [0202] (RMV, B_1 로부터의 공간 병합 후보);
- [0203] LTMV 및 LTMV 양쪽이 이용가능할 때, (RMV, LMV)
- [0204] 주어진 후보 쌍으로 예컨대, "LTMV 가 이용가능할 때" 조건이 주어지면, 이것은 그 조건이 참이 아니면 대응하는 쌍이 비교되지 않는다는 것을 의미한다.
- [0205] 각각의 후보 쌍 (C_0 , C_1) 에 대해, 그들이 상이하면, C_0 가 병합 후보 리스트에 삽입되며, 그렇지 않으면, C_0 가 병합 후보 리스트에 추가로 삽입되지 않는다는 점에 유의한다. 게다가, C_1 이 병합 후보 리스트에 있지 않으면, 비교는 회피된다. C_0 및 C_1 의 참조 인덱스들 및 모션 벡터들이 각각 RefIdxLXC0, mvLXC0 그리고 RefIdxLXC1, mvLXC1 이라고 가정하며, 여기서, X 는 0 및 1 과 동일하다. C_0 및 C_1 는 다음 조건들 중 하나 이상이 참이면 상이한 것으로 간주된다:
- [0206] 0 및 1 인 예측 리스트 표시 X 에 대해,
- [0207] - RefIdxLXC0 는 RefIdxLXC1 과 동일하지 않다;
- [0208] - RefIdxLXC0 가 -1 과 동일하지 않을 때 mvLXC0[0] 는 mvLXC1[0] 과 동일하지 않다;
- [0209] - RefIdxLXC0 가 -1 과 동일하지 않을 때 mvLXC0[1] 는 mvLXC1[1] 과 동일하지 않다;
- [0210] 모션 벡터가 주어진 예측 리스트 표시 X 에 대해 이용불가능하면, 대응하는 참조 인덱스는 -1 과 동일한 것을 간주될 수도 있다는 점에 유의한다.
- [0211] 제 2 예에서, 다음 (최고 2) 모션 벡터 후보 쌍들이 (그들 둘 모두가 이용가능하면) 비교된다:
- [0212] LTMV 가 이용가능할 때, (LMV, 디스패리티 벡터로부터 도출된 IPMVC)
- [0213] RTMV 가 이용가능할 때, (RMV, 디스패리티 벡터로부터 도출된 IPMVC).
- [0214] 제 3 예에서, 다음 모션 벡터 후보 쌍들이 (그들 둘 모두가 이용가능하면) 비교된다:
- [0215] LTMV 가 이용가능할 때, (LMV, 디스패리티 벡터로부터 도출된 IPMVC);
- [0216] (LMV, B_1 로부터의 공간 병합 후보);
- [0217] RTMV 가 이용가능할 때, (RMV, 디스패리티 벡터로부터 도출된 IPMVC)
- [0218] (RMV, B_1 로부터의 공간 병합 후보).
- [0219] 제 4 예에서, 다음 모션 벡터 후보 쌍들이 (그들 둘 모두가 이용가능하면) 비교된다:
- [0220] LTMV 가 이용가능할 때, (LMV, 디스패리티 벡터로부터 도출된 IPMVC);
- [0221] (LMV, A_1 으로부터의 공간 병합 후보);
- [0222] RTMV 가 이용가능할 때, (RMV, 디스패리티 벡터로부터 도출된 IPMVC)
- [0223] (RMV, A_1 으로부터의 공간 병합 후보)
- [0224] 제 5 예에서, 다음 (최고 5개의) 모션 벡터 후보 쌍들이 (그들 둘 모두가 이용가능하면) 비교된다:
- [0225] LTMV 가 이용가능할 때, (LMV, 디스패리티 벡터로부터 도출된 IPMVC);

- [0226] (LMV, A₁ 으로부터의 공간 병합 후보);
- [0227] RTMV 가 이용가능할 때, (RMV, 디스패리티 벡터로부터 도출된 IPMVC);
- [0228] (RMV, A₁ 으로부터의 공간 병합 후보);
- [0229] LTMV 및 LTMV 양쪽이 이용가능할 때, (RMV, LMV)
- [0230] 도 7 은 위에서 설명한 인터-뷰 예측 기술들을 포함한, 본 개시물에서 설명하는 기법들을 구현할 수도 있는 예시적인 비디오 인코더 (20) 를 예시하는 블록도이다. 비디오 인코더 (20) 는 비디오 슬라이스들 내 비디오 블록들의 인트라-코딩 및 인터-코딩을 수행할 수도 있다. 비디오 인코더 (20) 에 의해 수행될 때, 인트라-코딩은 인트라-예측 인코딩으로서 지칭될 수도 있으며, 인터-코딩은 인터-예측 인코딩으로서 지칭될 수도 있다.

인트라 코딩은 주어진 비디오 프레임 또는 화상 내 비디오에서 공간 리던던시를 감소시키거나 또는 제거하기 위해, 공간 예측에 의존한다. 인터-코딩은 비디오 시퀀스의 인접 프레임들 또는 화상들 내 비디오에서 시간 리던던시를 감소시키거나 또는 제거하기 위해, 시간 예측에 의존한다. 인터-코딩은 또한 또 다른 뷰에서의 화상이 인코딩 현재의 뷰에서 현재의 화상에서의 현재의 블록을 인코딩하기 위한 참조 화상으로서 사용되는 인터-뷰 코딩을 지칭할 수도 있다. 인트라-모드 (I 모드) 는 여러 공간 기반의 압축 모드들 중 임의의 압축 모드를 지칭할 수도 있다. 단방향 예측 (P 모드) 또는 양방향-예측 (B 모드) 과 같은 인터-모드들은 여러 시간-기반의 압축 모드들 또는 인터-뷰 압축 모드들 중 임의의 모드를 지칭할 수도 있다.
- [0231] 도 7 의 예에서, 비디오 인코더 (20) 는 파티셔닝 유닛 (35), 예측 프로세싱 유닛 (41), 필터 유닛 (63), 참조 화상 메모리 (64), 합산기 (50), 변환 유닛 (52), 양자화 유닛 (54), 및 엔트로피 인코딩 유닛 (56) 을 포함한다. 예측 프로세싱 유닛 (41) 은 모션 및 디스패리티 추정 유닛 (42), 모션 및 디스패리티 보상 유닛 (44), 및 인트라 예측 프로세싱 유닛 (46) 을 포함한다. 비디오 블록 재구성을 위해, 비디오 인코더 (20) 는 또한 역양자화 유닛 (58), 역변환 프로세싱 유닛 (60), 및 합산기 (62) 를 포함한다. 필터 유닛 (63) 은 디블록킹 필터, 적응 루프 필터 (ALF), 및 샘플 적응 오프셋 (SAO) 필터와 같은, 하나 이상의 루프 필터들을 나타내도록 의도된다. 필터 유닛 (63) 이 도 7 에 루프 필터인 것으로 도시되지만, 다른 구성들에서는, 필터 유닛 (63) 은 사후 루프 필터로서 구현될 수도 있다.
- [0232] 도 7 에 나타낸 바와 같이, 비디오 인코더 (20) 는 비디오 데이터를 수신하고, 파티셔닝 유닛 (35) 은 데이터를 비디오 블록들로 파티셔닝한다. 이 파티셔닝은 또한 슬라이스들, 타일들, 또는 다른 더 큰 유닛들 뿐만 아니라, 예컨대, LCU들 및 CU들의 쿼드트리 구조에 따라서 파티셔닝하는 비디오 블록으로 파티셔닝하는 것을 포함할 수도 있다. 비디오 인코더 (20) 는 일반적으로 인코딩되는 비디오 슬라이스 내 비디오 블록들을 인코딩하는 구성요소들을 예시한다. 슬라이스는 다수의 비디오 블록들로 (그리고, 아마도, 타일들로서 지칭되는 비디오 블록들의 세트들로) 분할될 수도 있다. 예측 프로세싱 유닛 (41) 은 여러 결과들 (예컨대, 코딩 레이트 및 왜곡의 레벨) 에 기초하여, 현재의 비디오 블록에 대해, 복수의 인트라 코딩 모드들 중 하나 또는 복수의 인터 코딩 모드들 중 하나와 같은, 복수의 가능한 코딩 모드들 중 하나를 선택할 수도 있다. 예측 프로세싱 유닛 (41) 은 최종 인트라- 또는 인터-코딩된 블록을, 합산기 (50) 에 제공하여 잔여 블록 데이터를 발생시키고, 그리고 합산기 (62) 에 제공하여, 참조 화상으로서 사용하기 위한 인코딩된 블록을 재구성할 수도 있다.
- [0233] 예측 프로세싱 유닛 (41) 내의 인트라-예측 프로세싱 유닛 (46) 은 코딩되는 현재의 블록과 동일한 프레임 또는 슬라이스에서의 하나 이상의 이웃하는 블록들에 대해, 현재의 비디오 블록의 인트라-예측 코딩을 수행하여, 공간 압축을 제공할 수도 있다. 예측 프로세싱 유닛 (41) 내의 모션 및 디스패리티 추정 유닛 (42) 및 모션 및 디스패리티 보상 유닛 (44) 은 시간 또는 뷰 압축을 제공하기 위해, 하나 이상의 참조 화상들 및/또는 참조 뷰들에서 하나 이상의 예측 블록들에 대한, 현재의 비디오 블록의 인터-예측 코딩 및/또는 인터-뷰 코딩을 수행한다.
- [0234] 모션 및 디스패리티 추정 유닛 (42) 은 비디오 슬라이스에 대한 인터-예측 모드 및/또는 인터-뷰 예측 모드를 비디오 시퀀스에 대한 미리 결정된 패턴에 따라서 결정하도록 구성될 수도 있다. 미리 결정된 패턴은 그 시퀀스에서의 비디오 슬라이스들을 P 슬라이스들, 또는 B 슬라이스들로서 지시할 수도 있다. 모션 및 디스패리티 추정 유닛 (42) 및 모션 및 디스패리티 보상 유닛 (44) 은 고도로 통합될 수도 있지만, 개념적인 목적들을 위해 별개로 예시된다. 모션 및 디스패리티 추정 유닛 (42) 에 의해 수행되는 모션 및 디스패리티 추정은 비디오 블록들에 대한 모션을 추정하는 모션 벡터들을 발생하는 프로세스이다. 모션 벡터는, 예를 들어, 참조 화상 내 예측 블록에 대한, 현재의 비디오 프레임 또는 화상 내 비디오 블록의 PU 의 변위를 나타낼 수도 있다. 모션 및 디스패리티 추정 유닛 (42) 에 의해 수행되는 디스패리티 추정은 상이한 뷰에서의 블록으로부터

터 현재 코딩된 블록을 예측하는데 사용될 수도 있는, 디스패리티 모션 벡터들을 발생하는 프로세스이다.

- [0235] 예측 블록은 픽셀 차이의 관점에서 코딩되는 비디오 블록의 PU 에 가깝게 매칭하는 것으로 발견되는 블록이며, 절대 차이의 합 (SAD), 제곱 차이의 합 (SSD), 또는 다른 차이 메트릭들에 의해 결정될 수도 있다. 일부 예들에서, 비디오 인코더 (20) 는 참조 화상 메모리 (64) 에 저장된 참조 화상들의 서브-정수 픽셀 위치들에 대한 값들을 계산할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 참조 화상의 1/4 픽셀 위치들, 1/8 픽셀 위치들, 또는 다른 분수 픽셀 위치들의 값들을 내삽할 수도 있다. 따라서, 모션 및 디스패리티 추정 유닛 (42) 은 풀 픽셀 위치들 및 분수 픽셀 위치들에 대해, 모션 탐색을 수행하고, 분수 픽셀 정밀도를 가진 모션 벡터를 출력할 수도 있다.
- [0236] 모션 및 디스패리티 추정 유닛 (42) 은 PU 의 위치를 참조 화상의 예측 블록의 위치와 비교함으로써, 인터-코딩된 또는 인터뷰 예측된 슬라이스에서의 비디오 블록의 PU 에 대해 (모션 보상되는 예측을 위한) 모션 벡터 및/또는 (디스패리티 보상되는 예측을 위한) 디스패리티 모션 벡터를 계산한다. 참조 화상은 제 1 참조 화상 리스트 (RefPicList0) 또는 제 2 참조 화상 리스트 (RefPicList1) 로부터 선택될 수도 있으며, 이 리스트 각각은 하나 이상의 참조 화상 메모리 (64) 에 저장된 참조 화상들을 식별한다. 모션 및 디스패리티 추정 유닛 (42) 은 그 계산된 모션 벡터 및/또는 디스패리티 벡터를 엔트로피 인코딩 유닛 (56) 및 모션 보상 유닛 (44) 으로 전송한다.
- [0237] 모션 및 디스패리티 보상 유닛 (44) 에 의해 수행되는 모션 보상 및/또는 디스패리티 보상은 모션 추정 및/또는 디스패리티 추정에 의해 결정된 모션 벡터에 기초하여 예측 블록을 폐치하거나 또는 발생하여, 가능한 한, 내삽들을 서브-픽셀 정밀도까지 수행하는 것을 수반할 수도 있다. 현재의 비디오 블록의 PU 에 대한 모션 벡터 및/또는 디스패리티를 수신하자 마자, 모션 및 디스패리티 보상 유닛 (44) 은 모션 벡터 및/또는 디스패리티 벡터가 참조 화상 리스트들 중 하나에서 가리키는 예측 블록을 로케이트할 수도 있다. 비디오 인코더 (20) 는 코딩중인 현재의 비디오 블록의 픽셀 값들로부터 예측 블록의 픽셀 값들을 감산하여 픽셀 차이 값들을 형성함으로써 잔여 비디오 블록을 형성한다. 픽셀 차이 값들은 블록에 대한 잔여 데이터를 형성하며, 루마 및 크로마 차이 성분들 양쪽을 포함할 수도 있다. 합산기 (50) 는 이 감산 동작을 수행하는 구성요소 또는 구성요소들을 나타낸다. 모션 및 디스패리티 보상 유닛 (44) 은 또한 비디오 슬라이스의 비디오 블록들을 디코딩할 때에 비디오 디코더 (30) 에 의한 사용을 위해 비디오 블록들 및 비디오 슬라이스와 연관되는 구문 엘리먼트들을 발생할 수도 있다.
- [0238] 인트라-예측 프로세싱 유닛 (46) 은 위에서 설명한 바와 같이, 모션 및 디스패리티 추정 유닛 (42) 및 모션 및 디스패리티 보상 유닛 (44) 에 의해 수행되는 인트라-예측에 대한 대안으로서, 현재의 블록을 인트라-예측할 수도 있다. 특히, 인트라-예측 프로세싱 유닛 (46) 은 현재의 블록을 인코딩하는데 사용할 인트라-예측 모드를 결정할 수도 있다. 일부 예들에서, 인트라-예측 프로세싱 유닛 (46) 은 예컨대, 별개의 인코딩 과정들 동안 여러 인트라-예측 모드들을 이용하여 현재의 블록을 인코딩할 수도 있으며, 인트라-예측 프로세싱 유닛 (46) 은 테스트된 모드들로부터 사용할 적합한 인트라-예측 모드를 선택할 수도 있다. 예를 들어, 인트라-예측 프로세싱 유닛 (46) 은 여러 테스트된 인트라-예측 모드들에 대한 레이트-왜곡 분석을 이용하여 레이트-왜곡 값들을 계산하고, 그 테스트된 모드들 중에서 최상의 레이트-왜곡 특성들을 갖는 인트라-예측 모드를 선택할 수도 있다. 레이트-왜곡 분석은 일반적으로 인코딩된 블록과 그 인코딩된 블록을 발생하기 위해 인코딩되었던 원래의 미인코딩된 블록 사이의 왜곡의 양 (또는, 에러) 뿐만 아니라, 그 인코딩된 블록을 발생하는데 사용되는 비트 레이트 (즉, 비트들의 수) 를 결정한다. 인트라-예측 프로세싱 유닛 (46) 은 여러 인코딩된 블록들에 대한 왜곡들 및 레이트들로부터 비율들 (비율) 을 계산하여, 어느 인트라-예측 모드가 그 블록에 대해 최상의 레이트-왜곡 값을 나타내는 지를 결정할 수도 있다.
- [0239] 어쨌든, 블록에 대한 인트라-예측 모드를 선택한 후, 인트라-예측 프로세싱 유닛 (46) 은 그 블록에 대한 선택된 인트라-예측 모드를 나타내는 정보를 엔트로피 코딩 유닛 (56) 에 제공할 수도 있다. 엔트로피 코딩 유닛 (56) 은 본 개시물의 기법에 따라서 그 선택된 인트라-예측 모드를 나타내는 정보를 인코딩할 수도 있다. 비디오 인코더 (20) 는 복수의 인트라-예측 모드 인덱스 테이블들 및 복수의 수정된 인트라-예측 모드 인덱스 테이블들 (또한, 코드워드 맵핑 테이블들로서 지칭됨) 을 포함할 수도 있는 그 송신되는 비트스트림 구성 데이터에, 여러 블록들에 대한 인코딩 컨텍스트들의 정의들, 및 가장 가능성있는 인트라-예측 모드, 인트라-예측 모드 인덱스 테이블 및 컨텍스트들의 각각에 사용할 수정된 인트라-예측 모드 인덱스 테이블의 표시들을 포함할 수도 있다.
- [0240] 예측 프로세싱 유닛 (41) 이 현재의 비디오 블록에 대한 예측 블록을 인트라-예측 또는 인트라-예측을 통해서 받

생한 후, 비디오 인코더 (20) 는 현재의 비디오 블록으로부터 예측 블록을 감산함으로써 잔여 비디오 블록을 형성한다. 잔여 블록에서의 잔여 비디오 데이터는 하나 이상의 TU들에 포함되어, 변환 유닛 (52) 에 적용될 수도 있다. 변환 유닛 (52) 은 이산 코사인 변환 (DCT) 또는 개념적으로 유사한 변환과 같은 변환을 이용하여 잔여 비디오 데이터를 잔여 변환 계수들로 변환한다. 변환 유닛 (52) 은 잔여 비디오 데이터를 픽셀 도메인으로부터 주파수 도메인과 같은 변환 도메인으로 변환할 수도 있다.

[0241] 변환 유닛 (52) 은 최종 변환 계수들을 양자화 유닛 (54) 으로 전송할 수도 있다. 양자화 유닛 (54) 은 비트 레이트를 추가로 감소시키기 위해 변환 계수들을 양자화한다. 양자화 프로세스는 그 계수들의 일부 또는 모두와 연관되는 비트 심도를 감소시킬 수도 있다. 양자화의 정도는 양자화 파라미터를 조정함으로써 변경될 수도 있다. 일부 예들에서, 양자화 유닛 (54) 은 그후 양자화된 변환 계수들을 포함하는 매트릭스의 스캐닝을 수행할 수도 있다. 이의 대안으로, 엔트로피 인코딩 유닛 (56) 은 그 스캐닝을 수행할 수도 있다.

[0242] 양자화 이후, 엔트로피 인코딩 유닛 (56) 은 양자화된 변환 계수들을 엔트로피 인코딩한다. 예를 들어, 엔트로피 인코딩 유닛 (56) 은 컨텍스트 적응 가변 길이 코딩 (CAVLC), 컨텍스트 적응 2진 산술 코딩 (CABAC), 구문-기반 컨텍스트-적용 2진 산술 코딩 (SBAC), 확률 간격 파티셔닝 엔트로피 (PIPE) 코딩 또는 또 다른 엔트로피 인코딩 방법론 또는 기법을 수행할 수도 있다. 엔트로피 인코딩 유닛 (56) 에 의한 엔트로피 인코딩에 이어서, 인코딩된 비트스트림은 비디오 디코더 (30) 로 송신되거나, 또는 비디오 디코더 (30) 에 의한 추후 송신 또는 취출을 위해 아카이브될 수도 있다. 엔트로피 인코딩 유닛 (56) 은 또한 코딩중인 현재의 비디오 슬라이스에 대한 모션 벡터들 및 다른 구문 엘리먼트들을 엔트로피 인코딩할 수도 있다.

[0243] 역양자화 유닛 (58) 및 역변환 프로세싱 유닛 (60) 은 역양자화 및 역변환을 각각 적용하여, 참조 화상의 참조 블록으로 추후 사용을 위해 픽셀 도메인에서 잔여 블록을 재구성한다. 모션 및 디스패리티 보상 유닛 (44) 은 잔여 블록을 참조 화상 리스트들 중 하나 내에서 참조 화상들 중 하나의 예측 블록에 가산함으로써 참조 블록을 계산할 수도 있다. 모션 및 디스패리티 보상 유닛 (44) 은 또한 하나 이상의 내삽 필터들을 그 재구성된 잔여 블록에 적용하여, 모션 추정에 사용하기 위한 서브-정수 픽셀 값들을 계산할 수도 있다. 합산기 (62) 는 재구성된 잔여 블록을 모션 및 디스패리티 보상 유닛 (44) 에 의해 발생하는 모션 보상된 예측 블록에 가산하여, 참조 화상 메모리 (64) 에의 저장을 위한 참조 블록을 발생한다. 참조 블록은 후속 비디오 프레임 또는 화상에서의 블록을 인터-예측하기 위해, 모션 및 디스패리티 추정 유닛 (42) 및 모션 및 디스패리티 보상 유닛 (44) 에 의해 참조 블록으로서 사용될 수도 있다.

[0244] 도 8 은 위에서 설명한 인터-뷰 예측 기술들을 포함한, 본 개시물에서 설명하는 기법들을 구현할 수도 있는 예시적인 비디오 디코더 (30) 를 예시하는 블록도이다. 도 8 의 예에서, 비디오 디코더 (30) 는 엔트로피 디코딩 유닛 (80), 예측 프로세싱 유닛 (81), 역양자화 유닛 (86), 역변환 프로세싱 유닛 (88), 합산기 (90), 필터 유닛 (91), 및 참조 화상 메모리 (92) 를 포함한다. 예측 프로세싱 유닛 (81) 은 모션 및 디스패리티 보상 유닛 (82) 및 인트라 예측 프로세싱 유닛 (84) 을 포함한다. 비디오 디코더 (30) 는 일부 예들에서, 도 7 로부터의 비디오 인코더 (20) 에 대해 설명된 인코딩 과정과는 일반적으로 반대인 디코딩 과정을 수행할 수도 있다.

[0245] 디코딩 프로세스 동안, 비디오 디코더 (30) 는 인코딩된 비디오 슬라이스의 비디오 블록들 및 연관되는 구문 엘리먼트들을 나타내는 인코딩된 비디오 비트스트림을 비디오 인코더 (20) 로부터 수신한다. 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (80) 은 비트스트림을 엔트로피 디코딩하여, 양자화된 계수들, 모션 벡터들, 및 다른 구문 엘리먼트들을 발생한다. 엔트로피 디코딩 유닛 (80) 은 모션 벡터들 및 다른 구문 엘리먼트들을 예측 프로세싱 유닛 (81) 으로 포워딩한다. 비디오 디코더 (30) 는 구문 엘리먼트들을 비디오 슬라이스 레벨 및/또는 비디오 블록 레벨에서 수신할 수도 있다.

[0246] 비디오 슬라이스가 인트라-코딩된 (I) 슬라이스로서 코딩될 때, 예측 프로세싱 유닛 (81) 의 인트라 예측 프로세싱 유닛 (84) 은 시그널링된 인트라 예측 모드 및 현재의 프레임 또는 화상의 이전에 디코딩된 블록들로부터의 데이터에 기초하여, 현재의 비디오 슬라이스의 비디오 블록에 대한 예측 데이터를 발생할 수도 있다. 비디오 프레임이 인터-코딩된 (즉, B, P 또는 GPB) 슬라이스로서 코딩될 때, 예측 프로세싱 유닛 (81) 의 모션 및 디스패리티 보상 유닛 (82) 은 엔트로피 디코딩 유닛 (80) 으로부터 수신된 모션 벡터들 및 다른 구문 엘리먼트들에 기초하여 현재의 비디오 슬라이스의 비디오 블록에 대한 예측 블록들을 발생한다. 예측 블록들은 참조 화상 리스트들 중 하나 내 참조 화상들 중 하나로부터 발생될 수도 있다. 비디오 디코더 (30) 는 디폴트 구성 기법들을 이용하여, 참조 화상 메모리 (92) 에 저장된 참조 화상들에 기초하여, 참조 프레임 리스트들, 즉, 리스트 0 및 리스트 1 를 구성할 수도 있다.

- [0247] 모션 및 디스패리티 보상 유닛 (82) 은 모션 벡터들 및 다른 구문 엘리먼트들을 파싱하여 현재의 비디오 슬라이스의 비디오 블록에 대한 예측 정보를 결정하고, 그리고, 그 예측 정보를 이용하여, 디코딩중인 현재의 비디오 블록에 대한 예측 블록들을 발생한다. 예를 들어, 모션 및 디스패리티 보상 유닛 (82) 은, 현재의 비디오 슬라이스에서의 비디오 블록들을 디코딩하기 위해, 그 수신된 구문 엘리먼트들 중 일부를 이용하여, 비디오 슬라이스의 비디오 블록들을 코딩하는 사용되는 예측 모드 (예컨대, 인트라- 또는 인터-예측), 인터-예측 또는 인터뷰 예측 슬라이스 유형 (예컨대, B 슬라이스 또는 P 슬라이스), 슬라이스에 대한 참조 화상 리스트들 중 하나 이상에 대한 구성 정보, 슬라이스의 각각의 인터-인코딩된 비디오 블록에 대한 모션 벡터들 및/또는 디스패리티 벡터들, 슬라이스의 각각의 인터-코딩된 비디오 블록에 대한 인터-예측 상태, 및 다른 정보를 결정한다.
- [0248] 일부 예들에서, 모션 및 디스패리티 보상 유닛 (82) 은 모션 벡터 예측 프로세스를 이용하여, 모션 벡터들을 나타내는 시그널링된 구문 엘리먼트들을 결정할 수도 있다. 모션 벡터 예측 프로세스들은 AMVP 모드 및 병합 모드를 포함할 수도 있다. 모션 및 디스패리티 보상 유닛 (82) 은 또한 내삽 필터들에 기초하여 내삽을 수행할 수도 있다. 모션 및 디스패리티 보상 유닛 (82) 은 비디오 블록들의 인코딩 동안 비디오 인코더 (20) 에 의해 사용되는 것과 같은 내삽 필터들을 이용하여, 참조 블록들의 서브-정수 픽셀들에 대해 내삽된 값들을 계산할 수도 있다. 이 경우, 모션 및 디스패리티 보상 유닛 (82) 은 수신된 구문 엘리먼트들로부터 비디오 인코더 (20) 에 의해 사용되는 내삽 필터들을 결정하고 그 내삽 필터들을 이용하여 예측 블록들을 발생할 수도 있다.
- [0249] 역양자화 모듈 (86) 은 비트스트림으로 제공되어 엔트로피 디코딩 유닛 (80) 에 의해 디코딩되는 양자화된 변환 계수들을 역양자화한다, 즉 양자화 해제한다. 역양자화 프로세스는 양자화의 정도와, 마찬가지로, 적용되어야 하는 역양자화의 정도를 결정하기 위해, 비디오 슬라이스에서의 각각의 비디오 블록에 대해 비디오 인코더 (20) 에 의해 계산된 양자화 파라미터의 사용을 포함할 수도 있다. 역변환 프로세싱 유닛 (88) 은 역 DCT, 역 정수 변환, 또는 개념적으로 유사한 역변환 프로세스와 같은 역변환을 변환 계수들에 적용하여, 픽셀 도메인에서 잔여 블록들을 발생한다.
- [0250] 모션 및 디스패리티 보상 유닛 (82) 이 모션 벡터들 및 다른 구문 엘리먼트들에 기초하여 현재의 비디오 블록에 대한 예측 블록을 발생한 후, 비디오 디코더 (30) 는 역변환 프로세싱 유닛 (88) 으로부터의 잔여 블록들을 모션 및 디스패리티 보상 유닛 (82) 에 의해 발생된 대응하는 예측 블록들과 합산함으로써, 디코딩된 비디오 블록을 형성한다. 합산기 (90) 는 이 합산 동작을 수행하는 구성요소 또는 구성요소들을 나타낸다. 원할 경우, (코딩 루프 중에 또는 코딩 루프 이후에) 루프 필터들이 또한 픽셀 전환들 (pixel transitions) 을 평활화하거나 또는 아니면 비디오 품질을 향상시키기 위해 사용될 수도 있다. 필터 유닛 (91) 은 디블록킹 필터, 적응 루프 필터 (ALF), 및 샘플 적응 오프셋 (SAO) 필터와 같은, 하나 이상의 루프 필터들을 나타내도록 의도된다. 필터 유닛 (91) 이 도 8 에 루프 필터인 것으로 도시되지만, 다른 구성들에서는, 필터 유닛 (91) 은 사후 루프 필터로서 구현될 수도 있다. 주어진 프레임 또는 화상에서 디코딩된 비디오 블록들은 그후 참조 화상 메모리 (92) 에 저장되며, 이 메모리는 후속 모션 보상을 위해 사용되는 참조 화상들을 저장한다. 참조 화상 메모리 (92) 는 또한 도 1 의 디스플레이 디바이스 (32) 와 같은 디스플레이 디바이스 상의 추후 프리젠테이션을 위해, 디코딩된 비디오를 저장한다.
- [0251] 도 9 는 본 개시물의 기법들에 따른 예시적인 인코딩 프로세스를 나타내는 플로우차트이다. 도 9 의 기법들은 비디오 인코더 (20) 에 대해 설명되지만 다른 유형들의 비디오 인코더들에 의해 수행될 수도 있다. 비디오 인코더 (20) 는 현재의 블록이 병합 모드, 스킵 모드, 또는 AMVP 모드에서 인터-예측되는지 여부를 결정한다 (180).
- [0252] 현재의 블록이 AMVP 모드에서 인터-예측되면, 비디오 인코더 (20) 는 참조 화상 인덱스에 기초하여 IDMVC 또는 IPMVC 를 도출할지 여부를 결정할 수도 있다 (182). 예를 들어, 참조 화상 인덱스가 시간 참조 화상을 참조하면, 비디오 인코더 (20) 는 IPMVC 를 결정할 수도 있으며, 참조 화상 인덱스가 인터-뷰 참조 화상을 참조하면, 비디오 인코더 (20) 는 IDMVC 를 결정할 수도 있다.
- [0253] 현재의 블록이 병합 모드 또는 스킵 모드에서 인터-예측되면, 비디오 인코더 (20) 는 IPMVC 의 이용가능성을 결정할 수도 있다 (184). 비디오 인코더 (20) 는 오프셋 값만큼 시프트된 디스패리티 벡터에 의해 참조되는, 현재의 블록의 참조 블록에 대한 모션 정보가 현재의 블록의 참조 화상 리스트에서 식별된 화상의 POC 값과 동일한 POC 값을 가진 참조 뷰에서의 화상을 참조하는 시간 모션 벡터인 모션 벡터이면, IPMVC 가 이용가능하다고 결정할 수도 있다. 위에서 설명된 바와 같이, IPMVC 는 현재의 블록에 대해, 제 1 오프셋 값만큼 시프트된 디스패리티 벡터에 의해 참조되는 참조 뷰에서 참조 블록의 모션 정보일 수도 있으며, IDMVC 는 현재의 블록에

대해, 상이한 제 2 오프셋 값만큼 시프트된 디스패리티 벡터일 수도 있다.

- [0254] 일부 예들에서, IPMVC 가 이용가능할 때, 비디오 인코더 (20) 는 IPMVC 를 도출할 수도 있으며, 병합 또는 스킵 모드에서 IDMVC 의 도출을 피할 수도 있다. IPMVC 가 이용불가능하고 현재의 블록이 병합 모드 또는 스킵 모드에서 인터-예측되면, 비디오 인코더 (20) 는 IDMVC 를 도출할 수도 있다.
- [0255] 비디오 인코더 (20) 는 IPMVC 또는 IDMVC 가 후보 리스트에 포함되어 있는 것인지 여부를 결정할 수도 있다 (186). 일부 예들에서, 비디오 인코더 (20) 는 단지 IPMVC 및 IDMVC 중 하나를 포함하고, 양쪽 모두를 포함하지 않을 수도 있다 (즉, 단지 IPMVC 및 IDMVC 중 하나가 후보 리스트에 포함된다). 예를 들어, 병합 모드에서, IPMVC 가 이용가능하기 때문에 비디오 인코더 (20) 가 IPMVC 를 도출하면, 비디오 인코더 (20) 는 IDMVC 의 도출을 피할 수도 있다. AMVP 모드에서, 비디오 인코더 (20) 는 참조 화상 인덱스가 시간 참조 화상 또는 인터-뷰 참조 화상을 참조하는지 여부에 기초하여, 후보 리스트에 IPMVC 또는 IDMVC 중 하나를 포함시킬 수도 있다.
- [0256] 또한, 일부 예에서, 비디오 인코더 (20) 는 디스패리티 벡터를 시프트하지 않고 디스패리티 벡터에 기초하여 또 다른 IPMVC 를 도출할 수도 있다. 이들 예들에서, 비디오 인코더 (20) 는 2개의 IPMVCs (즉, 하나는 시프트된 디스패리티 벡터에 기초하고 하나는 시프트하지 않고 디스패리티 벡터에 기초한다) 에 대한 모션 정보를 비교할 수도 있다. 일부 예들에서, 비디오 인코더 (20) 는 디스패리티 벡터를 시프트시킨 것으로부터 도출된 IPMVC 에 대한 모션 정보를 단지 디스패리티 벡터를 시프트하지 않고 도출된 IPMVC 에 대한 모션 정보와 비교할 수도 있다. 2개의 IPMVCs 의 모션 정보가 동일하면, 비디오 인코더 (20) 는 시프트된 디스패리티 벡터에 기초한 IPMVC 를 후보 리스트에 포함시키지 않을 수도 있으며, 2개의 IPMVCs 의 모션 정보가 상이하면, 비디오 인코더 (20) 는 시프트된 디스패리티 벡터에 기초한 IPMVC 를 후보 리스트에 포함시키지 않을 수도 있다. 비디오 인코더 (20) 는 시프트시키지 않은 디스패리티 벡터에 기초한 IPMVC 를 항상 포함시킬 수도 있다.
- [0257] 비디오 인코더 (20) 는 후보 리스트에 기초하여 현재의 블록을 인터-예측 인코딩할 수도 있다 (188). 비디오 인코더 (20) 는 또한 현재의 블록이 병합 모드, 스킵 모드, 또는 AMVP 모드에서 인터-예측되는지 여부를 나타내는 구문 엘리먼트들 (예컨대, 시그널링 구문 엘리먼트들), 및 비디오 디코더 (30) 가 현재의 블록을 인터-예측 인코딩하기 위한 현재의 블록에 대한 모션 벡터를 결정하는데 사용하는 후보 리스트로의 인덱스를 시그널링할 수도 있다 (190).
- [0258] 도 10 은 본 개시물의 기법들에 따른 예시적인 디코딩 프로세스를 나타내는 플로우차트이다. 도 10 의 기법들은 비디오 디코더 (30) 에 대해 설명되지만 다른 유형들의 비디오 디코더에 의해 수행될 수도 있다. 비디오 디코더 (30) 는 후보 리스트로의 인덱스를 수신한다 (200). 비디오 디코더 (30) 는 현재의 블록이 병합 모드, 스킵 모드, 또는 AMVP 모드에서 인터-예측되는지 여부를 결정하기 위해, 현재의 블록이 병합 모드, 스킵 모드, 또는 AMVP 모드에서 인터-예측되는지 여부를 나타내는 구문 엘리먼트들을 수신할 수도 있다 (예컨대, 구문 엘리먼트들을 디코딩하거나 또는 구문 엘리먼트들을 디코딩하고 있을 수도 있다) (202).
- [0259] 현재의 블록이 AMVP 모드에서 인터-예측된다고 비디오 디코더 (30) 가 결정하면, 비디오 디코더 (30) 는 참조 화상 인덱스에 기초하여 IDMVC 또는 IPMVC 를 도출할지 여부를 결정할 수도 있다 (204). 예를 들어, 참조 화상 인덱스가 시간 참조 화상을 참조하면, 비디오 디코더 (30) 는 IPMVC 를 결정할 수도 있으며, 참조 화상 인덱스가 인터-뷰 참조 화상을 참조하면, 비디오 디코더 (30) 는 IDMVC 를 결정할 수도 있다.
- [0260] 현재의 블록이 병합 모드 또는 스킵 모드에서 인터-예측되면, 비디오 디코더 (30) 는 IPMVC 의 이용가능성을 결정할 수도 있다 (206). 비디오 디코더 (30) 는 오프셋 값만큼 시프트된 디스패리티 벡터에 의해 참조되는, 현재의 블록의 참조 블록에 대한 모션 정보가 현재의 블록의 참조 화상 리스트에서 식별된 화상의 POC 값과 동일한 POC 값을 가진 참조 뷰에서의 화상을 참조하는 시간 모션 벡터인 모션 벡터를 포함하면, IPMVC 가 이용가능하다고 결정할 수도 있다. 또, 위에서 설명한 바와 같이, IPMVC 는 현재의 블록에 대해, 제 1 오프셋 값만큼 시프트된 디스패리티 벡터에 의해 참조되는 참조 뷰에서 참조 블록에 대한 모션 벡터일 수도 있으며, IDMVC 는 현재의 블록에 대해, 상이한 제 2 오프셋 값만큼 시프트된 디스패리티 벡터일 수도 있다.
- [0261] 일부 예들에서, IPMVC 가 이용가능할 때, 비디오 디코더 (30) 는 IPMVC 를 도출할 수도 있으며, 병합 또는 스킵 모드에서 IDMVC 의 도출을 피할 수도 있다. IPMVC 가 이용불가능하고 현재의 블록이 병합 모드 또는 스킵 모드에서 인터-예측되면, 비디오 디코더 (30) 는 IDMVC 를 도출할 수도 있다.
- [0262] 비디오 디코더 (30) 는 IPMVC 또는 IDMVC 가 후보 리스트에 포함되어 있는 것인지 여부를 결정할 수도 있다 (208). 일부 예들에서, 비디오 디코더 (30) 는 단지 IPMVC 및 IDMVC 중 하나를 포함하고, 양쪽 모두를 포함

하지 않을 수도 있다 (즉, 단지 IPMVC 및 IDMVC 중 하나가 후보 리스트에 포함된다). 예를 들어, 병합 모드에서, IPMVC 가 이용가능하기 때문에 비디오 디코더 (30) 가 IPMVC 를 도출하면, 비디오 디코더 (30) 는 IDMVC 의 도출을 피할 수도 있다. AMVP 모드에서, 비디오 디코더 (30) 는 참조 화상 인덱스가 시간 참조 화상 또는 인터-뷰 참조 화상을 참조하는지 여부에 기초하여, 후보 리스트에 IPMVC 또는 IDMVC 중 하나를 포함시킬 수도 있다. 비디오 디코더 (30) 는 후보 리스트에 기초하여 현재의 블록을 인터-예측 디코딩할 수도 있다 (210).

[0263] 또한, 일부 예에서, 비디오 디코더 (30) 는 디스패리티 벡터를 시프트하지 않고 디스패리티 벡터에 기초하여 또 다른 IPMVC 를 도출할 수도 있다. 이들 예들에서, 비디오 디코더 (30) 는 2개의 IPMVCs (즉, 하나는 시프트된 디스패리티 벡터에 기초하고 하나는 시프트하지 않고 디스패리티 벡터에 기초한다) 에 대한 모션 정보를 비교할 수도 있다. 일부 예들에서, 비디오 디코더 (30) 는 디스패리티 벡터를 시프트시킨 것으로부터 도출된 IPMVC 에 대한 모션 정보를, 단지 디스패리티 벡터를 시프트하지 않고 도출된 IPMVC 에 대한 모션 정보와 비교할 수도 있다. 2개의 IPMVCs 의 모션 정보가 동일하면, 비디오 디코더 (30) 는 시프트된 디스패리티 벡터에 기초한 IPMVC 를 후보 리스트에 포함시키지 않을 수도 있으며, 2개의 IPMVCs 의 모션 정보가 상이하면, 비디오 디코더 (30) 는 시프트된 디스패리티 벡터에 기초한 IPMVC 를 후보 리스트에 포함시키지 않을 수도 있다. 비디오 디코더 (30) 는 시프트시키지 않은 디스패리티 벡터에 기초한 IPMVC 를 항상 포함시킬 수도 있다.

[0264] 더욱이, 상기 예시적인 기법들이 멀티뷰 비디오 코딩에 대해 설명되지만, 본 개시물에서 설명하는 기법들은 이에 한정되지 않는다. 예를 들어, 본 개시물에서 설명하는 기법들은 스케일러블 비디오 코딩을 위한 HEVC 표준 (SHEVC) 에 대한 확장관들을 포함한, 스케일러블 비디오 코딩 (SVC) 으로 확장될 수도 있다.

[0265] 스케일러블 비디오 코딩 (SVC) 에서는, 다수의 계층들이 있을 수 있다. 최하위 레벨에서의 계층은 마치 기초 계층 (BL) 처럼 기능하고, 최상위 레벨에서의 계층은 마치 항상 계층 (EL) 처럼 기능할 수도 있다. 중간에서의 모든 계층들은 EL 및 BL 양쪽으로서 기능할 수도 있다. 예를 들어, 중간에서의 계층은 그 아래의 계층들에 대해 EL 일 수도 있으며, 동시에 그 위에 있는 계층들에 대해 BL 로서 일 수도 있다. 설명의 단순성을 위해, 본 개시물은 현재의 기법들을 예시할 때에 2개의 계층들, 즉, BL 및 EL 이 존재한다고 가정한다. 본 개시물에서 설명하는 기법들은 다수의 계층들을 가진 경우들에도 또한 적용가능하다는 점에 유의해야 한다.

[0266] SCV 에서 디스패리티 벡터의 컨셉은 참조 계층에서의 참조 블록을 참조하는 벡터를 지칭할 수도 있으며, 여기서, 참조 계층은 인터-예측중인 블록을 포함하는 계층과는 또 다른 계층이다. 이러한 인터-계층 벡터 (inter-layer vector) 의 일 예는 제로 벡터 (예컨대, 성분 (0, 0) 을 가진 벡터) 일 수도 있다. SCV 에 있어, 인터-계층 벡터는 3D 비디오를 참조하여 디스패리티 벡터에 대해 위에서 설명한 방법과 유사한 방법으로 이용될 수도 있다.

[0267] 위에서 설명된 바와 같이, 3D 비디오에 대해, 비디오 코더 (20/30) 는 디스패리티 벡터를 시프트하지 않고 IPMVC 및 IDMVC 를 도출할 수도 있으며, 디스패리티 벡터를 그리고, 일부 예들에서는, 인터-뷰 후보들로서 일반적으로 지칭되는, 추가적인 IDMVCs 에 상대적인 추가적인 IPMVCs 에 대해 상이한 양들 (즉, 상이한 오프셋 값들) 만큼, 시프트시킴으로써 추가적인 IPMVCs 및 IDMVCs 를 도출할 수도 있다. SVC 에 대해, 비디오 코더 (20/30) 는 유사하게 인터-계층 후보들을 도출할 수도 있다. 일 예로서, 비디오 코더 (20/30) 는 인터-계층 벡터가 참조하는 참조 계층에서의 참조 블록으로부터 상속된 모션 정보에 기초하여, 및/또는 오프셋 값만큼 시프트된 인터-계층 벡터가 참조하는 참조 계층에서의 참조 블록으로부터 상속된 모션 정보에 기초하여, 인터-계층 후보를 도출할 수도 있다. 또 다른 예로서, 비디오 코더 (20/30) 는 인터-계층 벡터를 인터-계층 후보로 변환함으로써 및/또는 오프셋 값만큼 시프트된 인터-계층 벡터를 인터-계층 후보로 변환함으로써, 인터-계층 후보를 도출할 수도 있다.

[0268] 따라서, 일부 예들에서, IPMVC 및 IDMVC 를 도출하기 위한 본 개시물에서 설명하는 기법들, 및 IPMVC 및 IDMVC 가 (예컨대, 병합 모드, 스킵 모드, 또는 AMVP 모드에서) 도출될 때의 기법들이 SVC 로 확장될 수도 있다. SVC 에 대해, 비디오 코더 (20/30) 는 IPMVC 를 도출하는 방법과 유사한 방법으로 (예컨대, 현재의 블록에 대한 인터-계층 벡터 또는 시프트된 인터-계층 벡터가 참조하는 참조 계층에서의 참조 블록에 대한 모션 정보에 기초하여) 제 1 인터-계층 후보를 도출할 수도 있으며, IDMVC 를 도출하는 (예컨대, 현재의 블록에 대한 인터-계층 벡터 또는 시프트된 인터-계층 벡터를 인터-계층 후보로 변환하는) 방법과 유사한 방법으로 제 2 인터-계층 후보를 도출할 수도 있다. SVC 에 대해, 비디오 코더 (20/30) 는 제 1 인터-계층 후보 및 제 2 인터-계층 후보에 대한 3D 비디오에 대해 위에서 설명한 기법들과 유사한 기법들을 구현할 수도 있다.

[0269] 하나 이상의 예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합으로 구현될

수도 있다. 소프트웨어로 구현되는 경우, 그 기능들은 하나 이상의 명령들 또는 코드로서, 컴퓨터 판독가능 매체 상에 저장되거나 또는 컴퓨터 판독가능 매체를 통해서 송신될 수도 있으며, 하드웨어-기반의 프로세싱 유닛에 의해 실행될 수도 있다. 컴퓨터 판독가능 매체는 컴퓨터 판독가능 저장 매체들을 포함할 수도 있으며, 이 컴퓨터 판독가능 저장 매체들은 데이터 저장 매체와 같은 유형의 매체, 또는 예컨대, 통신 프로토콜에 따라서 한 장소로부터 다른 장소로의 컴퓨터 프로그램의 전송을 용이하게 하는 임의의 매체를 포함한 통신 매체들에 대응한다. 이런 방법으로, 컴퓨터 판독가능 매체들은 일반적으로 (1) 비밀시성 유형의 컴퓨터 판독가능 저장 매체, 또는 (2) 신호 또는 캐리어 파와 같은 통신 매체에 대응할 수도 있다. 데이터 저장 매체는 본 개시물에서 설명하는 기법들의 구현을 위한 명령들, 코드 및/또는 데이터 구조들을 추출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 가용 매체들일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터 판독가능 매체를 포함할 수도 있다.

[0270] 일 예로서, 이에 한정하지 않고, 이런 컴퓨터 판독가능 저장 매체는 RAM, ROM, EEPROM, CD-ROM 또는 다른 광디스크 스토리지, 자기디스크 스토리지, 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 원하는 프로그램 코드를 명령들 또는 데이터 구조들의 형태로 저장하는데 사용될 수 있고 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 접속이 컴퓨터 판독가능 매체로 적절히 지칭된다. 예를 들어, 동축 케이블, 광섬유 케이블, 연선, 디지털 가입자 회선 (DSL), 또는 무선 기술들 예컨대 적외선, 라디오, 및 마이크로파를 이용하여 명령들이 웹사이트, 서버, 또는 다른 원격 소스로부터 송신되는 경우, 동축 케이블, 광섬유 케이블, 연선, DSL, 또는 무선 기술들, 예컨대 적외선, 라디오, 및 마이크로파가 그 매체의 정의에 포함된다. 그러나, 컴퓨터 판독가능 저장 매체 및 데이터 저장 매체는 접속부들, 반송파들, 신호들, 또는 다른 일시성 매체를 포함하지 않고, 그 대신, 비-일시성 유형의 저장 매체로 송신되는 것으로 해석되어야 한다. 디스크 (disk) 및 디스크 (disc) 는, 본원에서 사용할 때, 콤팩트 디스크 (CD), 레이저 디스크, 광디스크, 디지털 다기능 디스크 (DVD), 블루레이 디스크 및 Blu-ray 디스크를 포함하며, 디스크들 (disks) 은 데이터를 자기적으로 보통 재생하지만, 디스크들 (discs) 은 레이저로 데이터를 광학적으로 재생한다. 앞에서 언급한 것들의 결합들이 또한 컴퓨터 판독가능 매체들의 범위 내에 포함되어야 한다.

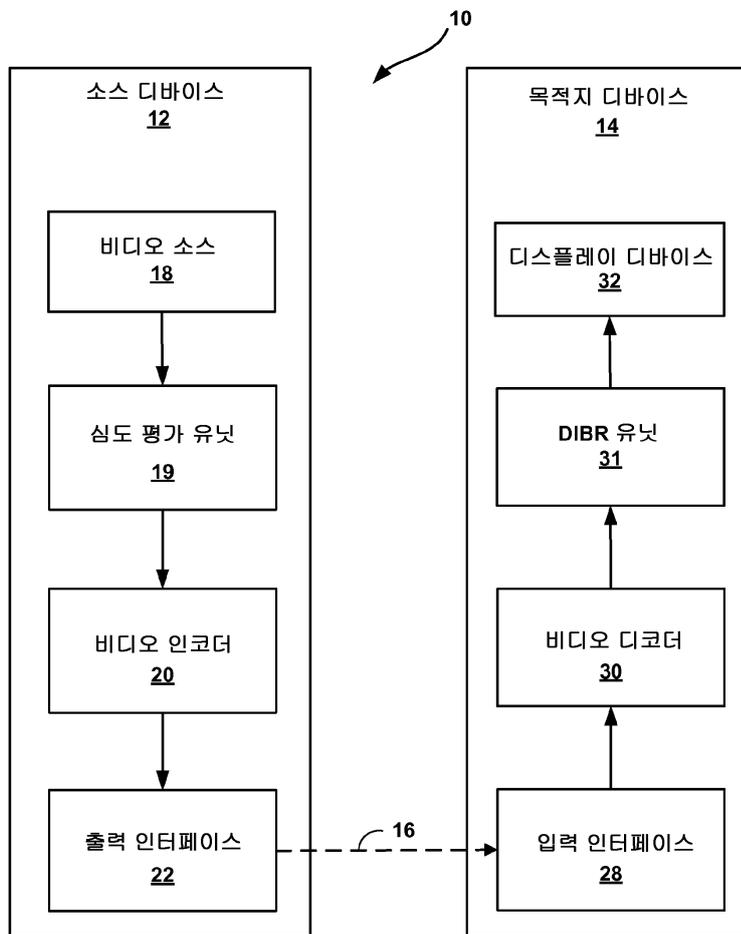
[0271] 명령들은 하나 이상의 디지털 신호 프로세서들 (DSPs), 범용 마이크로프로세서들, 주문형 집적회로들 (ASICs), 필드 프로그래밍가능 로직 어레이들 (FPGAs), 또는 다른 등가의 집적 또는 이산 로직 회로와 같은, 하나 이상의 프로세서들에 의해 실행될 수도 있다. 따라서, 용어 "프로세서" 는, 본원에서 사용될 때 전술한 구조 중 임의의 구조 또는 본원에서 설명하는 기법들의 구현에 적합한 임의의 다른 구조를 지칭할 수도 있다. 게다가, 일부 양태들에서, 본원에서 설명하는 기능은 전용 하드웨어 및/또는 인코딩 및 디코딩을 위해 구성되는 소프트웨어 모듈들 내에 제공되거나, 또는 결합된 코덱에 포함될 수도 있다. 또한, 이 기법들은 하나 이상의 회로들 또는 로직 엘리먼트들로 전적으로 구현될 수 있다.

[0272] 본 개시물의 기법들은 무선 핸드셋, 집적 회로 (IC) 또는 IC들의 세트 (예컨대, 칩 세트) 를 포함한, 매우 다양한 디바이스들 또는 장치들로 구현될 수도 있다. 개시한 기법들을 수행하도록 구성되는 디바이스들의 기능적 양태들을 강조하기 위해서 여러 구성요소들, 모듈들, 또는 유닛들이 본 개시물에서 설명되지만, 상이한 하드웨어 유닛들에 의한 실현을 반드시 필요로 하지는 않는다. 대신, 위에서 설명한 바와 같이, 여러 유닛들이 코덱 하드웨어 유닛에 결합되거나 또는 적합한 소프트웨어 및/또는 펌웨어와 함께, 위에서 설명한 바와 같은 하나 이상의 프로세서들을 포함한, 상호작용하는 하드웨어 유닛들의 컬렉션으로 제공될 수도 있다.

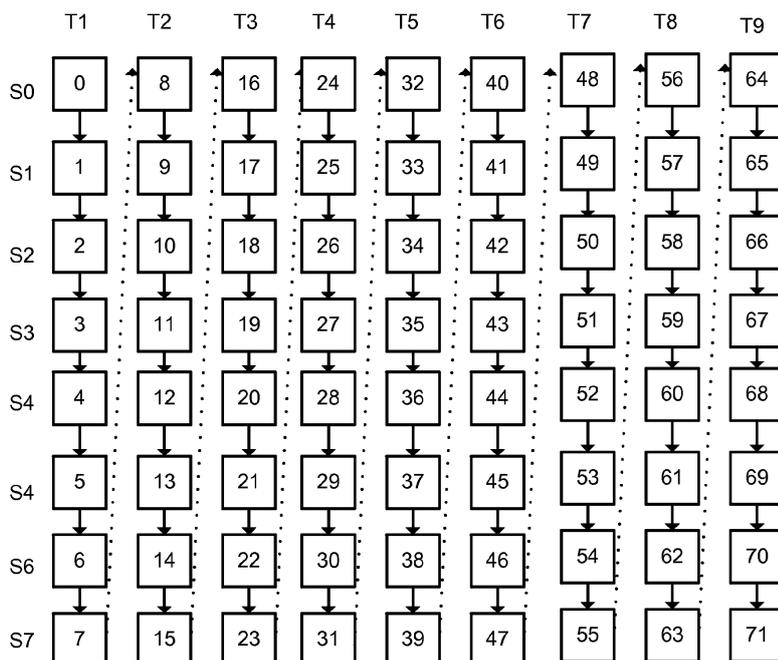
[0273] 여러 예들이 설명되었다. 이들 및 다른 예들은 다음 청구항들의 범위 이내이다.

도면

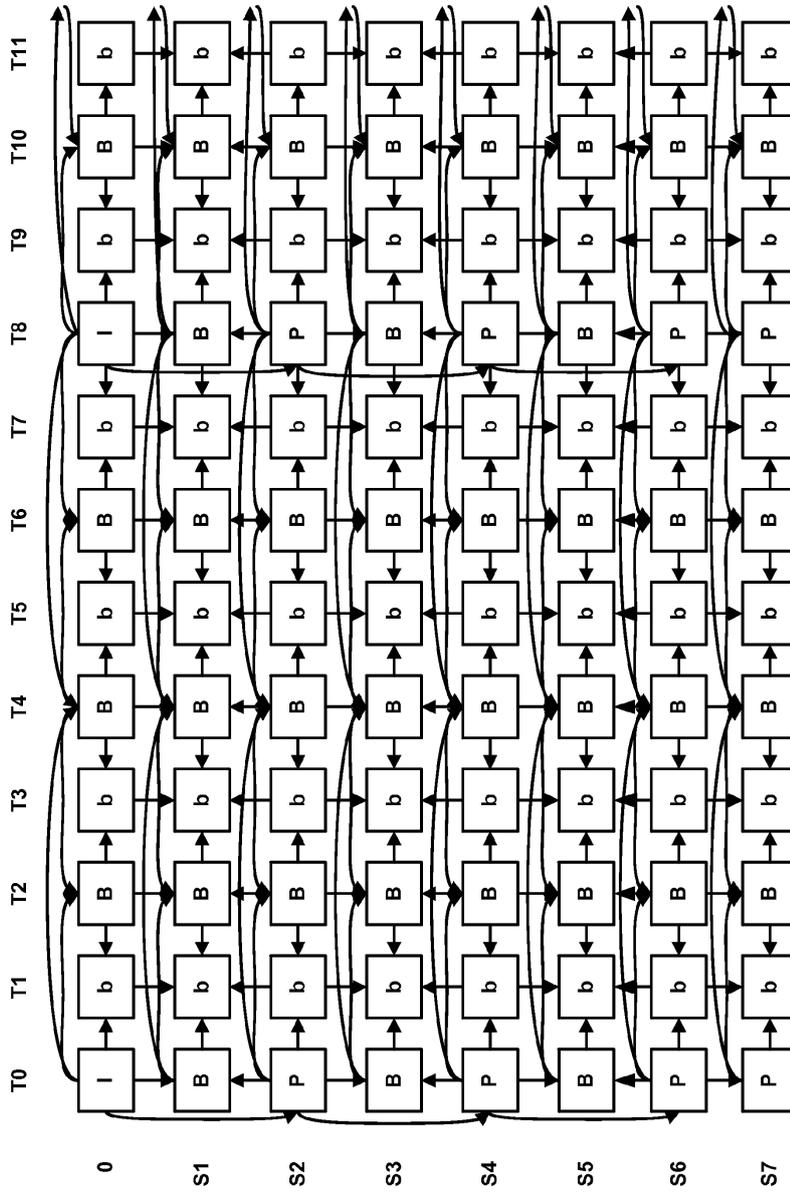
도면1



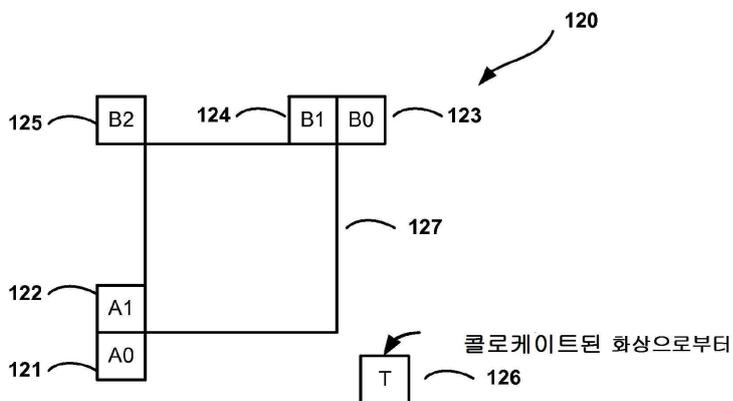
도면2



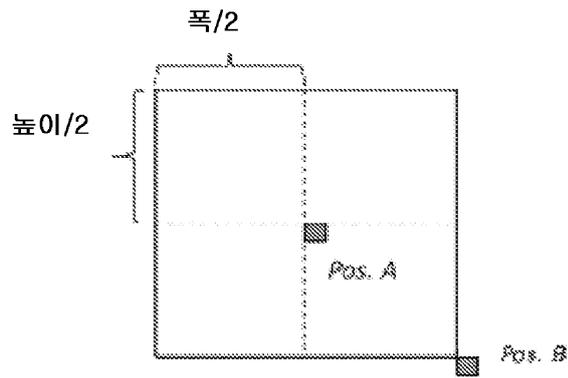
도면3



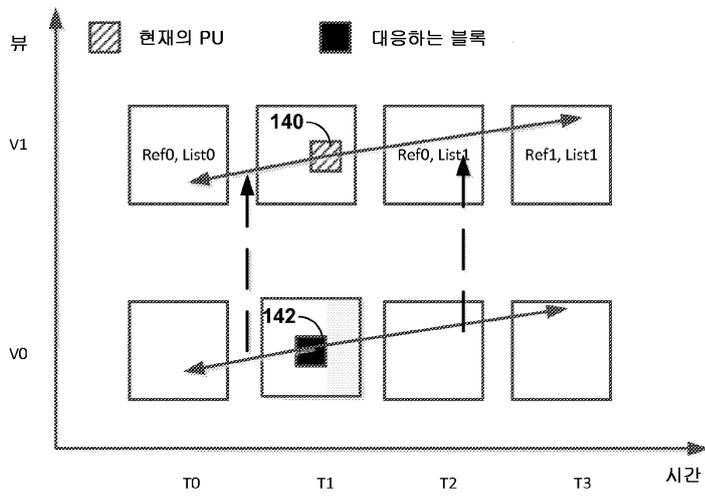
도면4



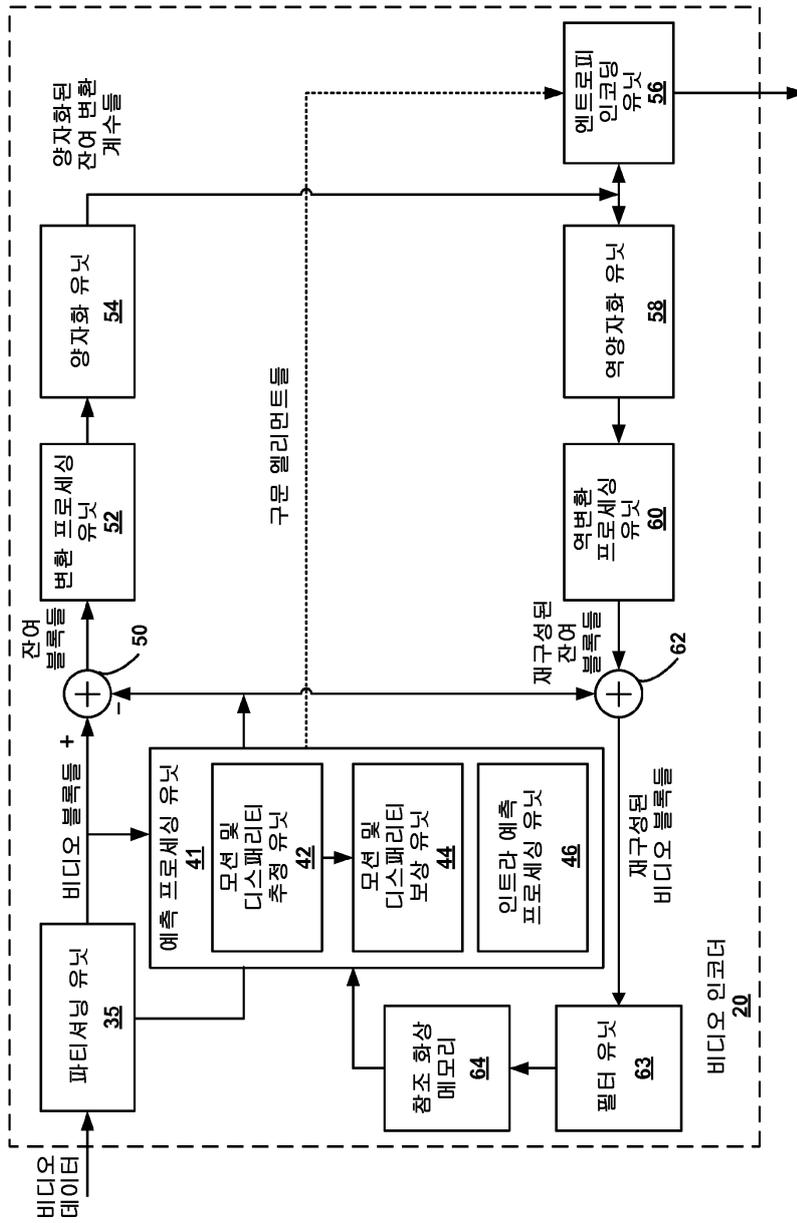
도면5



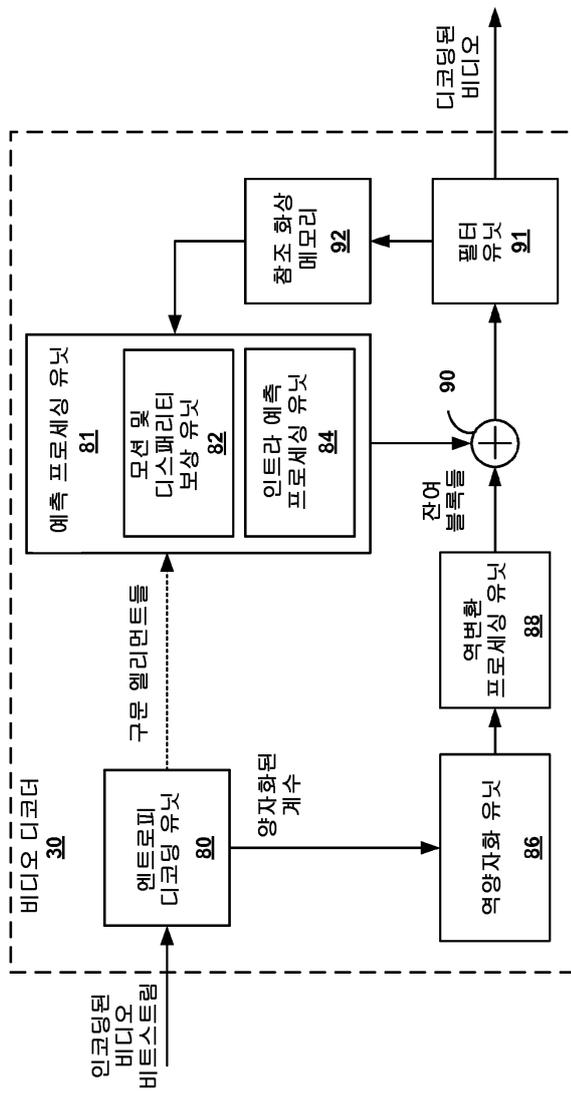
도면6



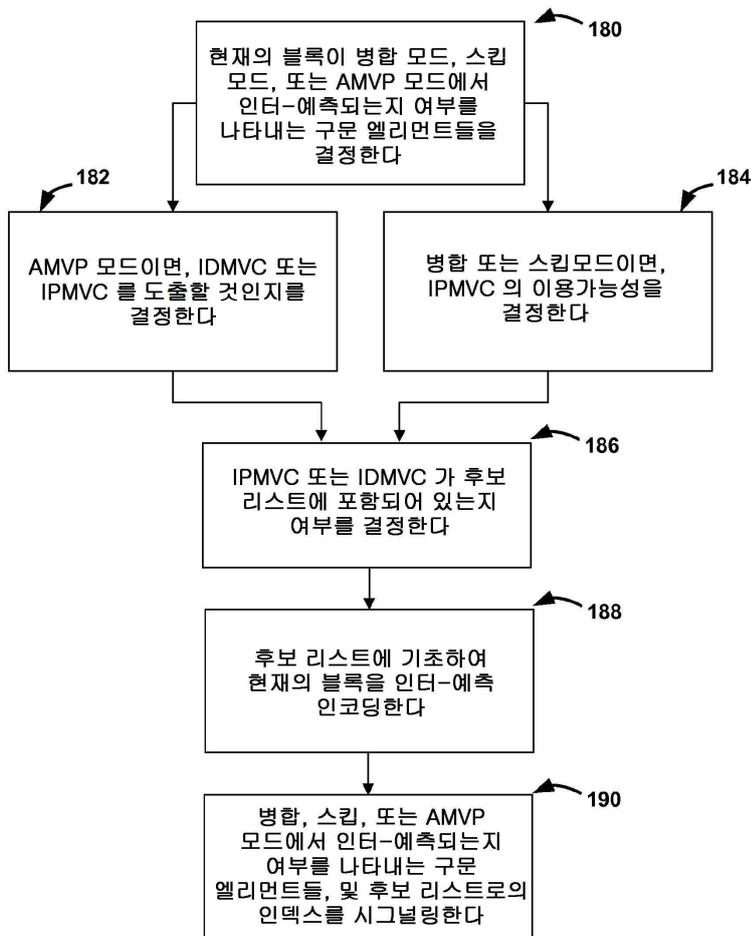
도면7



도면8



도면9



도면10

