



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2018년03월06일
 (11) 등록번호 10-1835240
 (24) 등록일자 2018년02월27일

- | | |
|--|---|
| (51) 국제특허분류(Int. Cl.)
H04N 19/513 (2014.01) H04N 19/573 (2014.01)
H04N 19/593 (2014.01) H04N 19/597 (2014.01)
(52) CPC특허분류
H04N 19/513 (2015.01)
H04N 19/573 (2015.01)
(21) 출원번호 10-2016-7007542
(22) 출원일자(국제) 2014년09월19일
심사청구일자 2017년03월31일
(85) 번역문제출일자 2016년03월22일
(65) 공개번호 10-2016-0058809
(43) 공개일자 2016년05월25일
(86) 국제출원번호 PCT/US2014/056557
(87) 국제공개번호 WO 2015/042399
국제공개일자 2015년03월26일
(30) 우선권주장
61/880,737 2013년09월20일 미국(US)
14/489,679 2014년09월18일 미국(US)
(56) 선행기술조사문헌
JCT3V-E1001 | (73) 특허권자
쉐컴 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
(74) 대리인
특허법인코리아나 |
|--|---|

전체 청구항 수 : 총 24 항

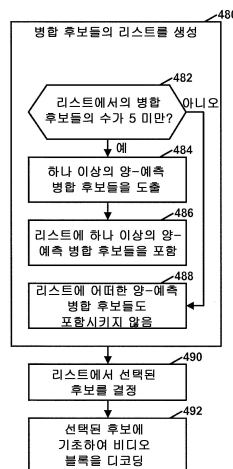
심사관 : 박상철

(54) 발명의 명칭 3D 비디오 코딩을 위한 조합된 양방향-예측 병합 후보들

(57) 요약

비디오 코더가 3D 비디오의 비디오 블록을 코딩하기 위한 병합 후보들의 리스트를 생성한다. 병합 후보들의 리스트에서의 병합 후보들의 최대 수는 6과 동일할 수도 있다. 병합 후보들의 리스트를 생성하는 부분으로서, 비디오 코더는 상기 병합 후보들의 리스트에서의 병합 후보들의 수가 5 미만인지의 여부를 결정한다. 만약 그렇다면, 비디오 코더는 하나 이상의 조합된 양방향-예측 병합 후보들을 도출한다. 비디오 코더는 병합 후보들의 리스트에 하나 이상의 조합된 양방향-예측 병합 후보들을 포함시킨다.

대표도 - 도14b



(52) CPC특허분류

H04N 19/593 (2015.01)

H04N 19/597 (2015.01)

명세서

청구범위

청구항 1

3차원 (3D) 비디오 데이터를 코딩하는 방법으로서,

상기 3D 비디오 데이터의 비디오 블록을 코딩하기 위한 병합 후보들의 현재 리스트를 생성하는 단계로서, 상기 병합 후보들의 현재 리스트에서의 병합 후보들의 최대 수는 6 과 동일하고, 5 개의 양방향-예측 병합 후보들을 갖는 병합 후보들의 리스트들에서 상이한 양방향-예측 병합 후보들의 리스트 0 및 리스트 1 모션 벡터들의 20 개의 가능한 조합들이 존재하는, 상기 병합 후보들의 현재 리스트를 생성하는 단계를 포함하며,

상기 병합 후보들의 현재 리스트를 생성하는 단계는,

상기 병합 후보들의 현재 리스트에서의 초기의 병합 후보들의 수가 5 미만인 것을 결정하는 단계로서, 0 내지 11 의 조합 인덱스의 각각의 개별 값은 0 내지 3 의 값들의 개별 사전-정의된 조합에 대응하는, 상기 초기의 병합 후보들의 수가 5 미만인 것을 결정하는 단계; 및

상기 병합 후보들의 현재 리스트에서의 상기 병합 후보들의 수가 5 미만이라고 결정하는 것에 응답하여:

상기 조합 인덱스의 상기 개별 값이 상기 병합 후보들의 현재 리스트에서의 상기 초기의 병합 후보들의 수를 상기 병합 후보들의 현재 리스트에서의 상기 초기의 병합 후보들의 수보다 1 적은 수에 의해 곱한 것과 동일하다는 조건, 및 상기 병합 후보들의 현재 리스트가 6 개의 병합 후보들을 갖는다는 조건 중 적어도 하나의 조건이 참일 때까지, 상기 0 내지 11 의 조합 인덱스의 각각의 개별 값에 대해:

상기 병합 후보들의 현재 리스트에서의 제 1 병합 후보가 리스트 0 모션 벡터를 갖는지 여부 및 상기 병합 후보들의 현재 리스트에서의 제 2 병합 후보가 리스트 1 모션 벡터를 갖는지 여부를 결정하는 것으로서, 상기 제 1 병합 후보 및 상기 제 2 병합 후보는 상기 조합 인덱스의 상기 개별 값에 대응하는 값들의 상기 사전-정의된 조합에 의해 표시되는 상기 병합 후보들의 현재 리스트에서의 포지션들에 있는, 상기 병합 후보들의 현재 리스트에서의 제 1 병합 후보가 리스트 0 모션 벡터를 갖는지 여부 및 상기 병합 후보들의 현재 리스트에서의 제 2 병합 후보가 리스트 1 모션 벡터를 갖는지 여부를 결정하는 것;

상기 제 1 병합 후보가 리스트 0 모션 벡터를 가지고 상기 제 2 병합 후보가 리스트 1 모션 벡터를 갖는다는 결정에 응답하여, 개별의 조합된 양방향-예측 병합 후보를 도출하는 것으로서, 상기 개별의 조합된 양방향-예측 병합 후보는 상기 제 1 병합 후보의 상기 리스트 0 모션 벡터와 상기 제 2 병합 후보의 상기 리스트 1 모션 벡터의 조합이고, 상기 제 1 병합 후보의 상기 모션 벡터와 상기 제 2 병합 후보의 상기 모션 벡터는 상이한 참조 화상 리스트들에서의 화상들을 가리키는, 상기 개별의 조합된 양방향-예측 병합 후보를 도출하는 것; 및

상기 병합 후보들의 현재 리스트에 상기 개별의 조합된 양방향-예측 병합 후보를 포함시키는 것

을 수행하는 단계를 포함하는, 3D 비디오 데이터 코딩 방법.

청구항 2

제 1 항에 있어서,

상기 병합 후보들의 현재 리스트를 생성하는 단계는,

상기 하나 이상의 양방향-예측 병합 후보들 중 임의의 양방향-예측 병합 후보를 상기 병합 후보들의 현재 리스트에 추가하기 전에 상기 병합 후보들의 현재 리스트에 5 개의 병합 후보들이 있다는 결정에 응답하여, 상기 병합 후보들의 현재 리스트에 제로 후보를 포함시키는 단계로서, 상기 제로 후보의 모션 벡터 성분들이 0 과 동일하고 상기 제로 후보의 참조 인덱스가 0 과 동일하며, 상기 참조 인덱스는 참조 화상 리스트에서의 참조 화상의 로케이션을 표시하는, 상기 제로 후보를 포함시키는 단계를 더 포함하는, 3D 비디오 데이터 코딩 방법.

청구항 3

제 1 항에 있어서,

상기 병합 후보들의 현재 리스트를 생성하는 단계는, 임의의 백워드 워핑 뷰 합성 (BVSP) 플래그들을 체크하는 일 없이 발생하는, 3D 비디오 데이터 코딩 방법.

청구항 4

제 1 항에 있어서,

상기 3D 비디오 데이터 코딩 방법은 상기 3D 비디오 데이터를 디코딩하는 방법을 포함하고 상기 비디오 블록은 예측 유닛 (PU) 이며,

상기 방법은,

비트스트림으로부터, 상기 병합 후보들의 현재 리스트에서 선택된 병합 후보를 표시하는 신택스 엘리먼트를 획득하는 단계; 및

상기 PU의 예측 샘플들을 생성하기 위해 상기 선택된 병합 후보의 모션 정보를 사용하는 단계를 더 포함하는, 3D 비디오 데이터 코딩 방법.

청구항 5

제 1 항에 있어서,

상기 3D 비디오 데이터 코딩 방법은 상기 3D 비디오 데이터를 인코딩하는 방법을 포함하며,

상기 방법은,

상기 병합 후보들의 현재 리스트에서 병합 후보를 선택하는 단계; 및

선택된 상기 병합 후보의 상기 병합 후보들의 현재 리스트에서의 위치를 시그널링하는 단계를 포함하는, 3D 비디오 데이터 코딩 방법.

청구항 6

제 1 항에 있어서,

상기 병합 후보들의 현재 리스트를 생성하는 단계는, 공간적 병합 후보들에 대한 도출 프로세스를 수행한 후에 그리고 시간적 병합 후보에 대한 도출 프로세스를 수행한 후에, 뷰 간 예측 모션 벡터 후보 (IPMVC) 를, 이용 가능하다면, 상기 병합 후보들의 현재 리스트에 삽입한 후 상기 하나 이상의 조합된 양방향-예측 병합 후보들을 도출하는 단계를 포함하며,

상기 공간적 병합 후보들에 대한 도출 프로세스는 상기 병합 후보들의 현재 리스트에 4 개까지의 공간적 모션 벡터 후보들을 도출 및 삽입하고,

상기 시간적 병합 후보에 대한 도출 프로세스는 시간적 모션 벡터 예측자 (TMVP) 후보를, 이용 가능하다면, 상기 병합 후보들의 현재 리스트에 추가하는, 3D 비디오 데이터 코딩 방법.

청구항 7

비디오 코딩 디바이스로서,

3차원 (3D) 비디오 데이터를 저장하도록 구성된 데이터 저장 매체; 및

하나 이상의 프로세서들을 포함하며,

상기 하나 이상의 프로세서들은,

상기 3D 비디오 데이터의 비디오 블록을 코딩하기 위한 병합 후보들의 현재 리스트를 생성하도록 구성되고, 상기 병합 후보들의 현재 리스트에서의 병합 후보들의 최대 수는 6 과 동일하고, 5 개의 양방향-예측 병합 후보들을 갖는 병합 후보들의 리스트들에서 상이한 양방향-예측 병합 후보들의 리스트 0 및 리스트 1 모션

벡터들의 20 개의 가능한 조합들이 존재하며,

상기 병합 후보들의 현재 리스트를 생성하는 것의 부분으로서, 상기 하나 이상의 프로세서들은,

상기 병합 후보들의 현재 리스트에서의 초기의 병합 후보들의 수가 5 미만인 것을 결정하는 것으로서, 0 내지 11 의 조합 인덱스의 각각의 개별 값은 0 내지 3 의 값들의 개별 사전-정의된 조합에 대응하는, 상기 초기의 병합 후보들의 수가 5 미만인 것을 결정하는 것을 행하고; 그리고

상기 병합 후보들의 현재 리스트에서의 상기 병합 후보들의 수가 5 미만이라고 결정하는 것에 응답하여:

상기 조합 인덱스의 상기 개별 값이 상기 병합 후보들의 현재 리스트에서의 상기 초기의 병합 후보들의 수를 상기 병합 후보들의 현재 리스트에서의 상기 초기의 병합 후보들의 수보다 1 적은 수에 의해 곱한 것과 동일하다는 조건, 및 상기 병합 후보들의 현재 리스트가 6 개의 병합 후보들을 갖는다는 조건 중 적어도 하나의 조건이 참일 때까지, 상기 0 내지 11 의 조합 인덱스의 각각의 개별 값에 대해:

상기 병합 후보들의 현재 리스트에서의 제 1 병합 후보가 리스트 0 모션 벡터를 갖는지 여부 및 상기 병합 후보들의 현재 리스트에서의 제 2 병합 후보가 리스트 1 모션 벡터를 갖는지 여부를 결정하는 것으로서, 상기 제 1 병합 후보 및 상기 제 2 병합 후보는 상기 조합 인덱스의 상기 개별 값에 대응하는 값들의 상기 사전-정의된 조합에 의해 표시되는 상기 병합 후보들의 현재 리스트에서의 포지션들에 있는, 상기 병합 후보들의 현재 리스트에서의 제 1 병합 후보가 리스트 0 모션 벡터를 갖는지 여부 및 상기 병합 후보들의 현재 리스트에서의 제 2 병합 후보가 리스트 1 모션 벡터를 갖는지 여부를 결정하는 것;

상기 제 1 병합 후보가 리스트 0 모션 벡터를 가지고 상기 제 2 병합 후보가 리스트 1 모션 벡터를 갖는다는 결정에 응답하여, 개별의 조합된 양방향-예측 병합 후보를 도출하는 것으로서, 상기 개별의 조합된 양방향-예측 병합 후보는 상기 제 1 병합 후보의 상기 리스트 0 모션 벡터와 상기 제 2 병합 후보의 상기 리스트 1 모션 벡터의 조합이고, 상기 제 1 병합 후보의 상기 모션 벡터와 상기 제 2 병합 후보의 상기 모션 벡터는 상이한 참조 화상 리스트들에서의 화상들을 가리키는, 상기 개별의 조합된 양방향-예측 병합 후보를 도출하는 것; 및

상기 병합 후보들의 현재 리스트에 상기 개별의 조합된 양방향-예측 병합 후보를 포함시키는 것

을 수행하는, 비디오 코딩 디바이스.

청구항 8

제 7 항에 있어서,

상기 병합 후보들의 현재 리스트를 생성하는 것의 부분으로서, 상기 하나 이상의 프로세서들은,

상기 하나 이상의 양방향-예측 병합 후보들 중 임의의 양방향-예측 병합 후보를 상기 병합 후보들의 현재 리스트에 추가하기 전에 상기 병합 후보들의 현재 리스트에 5 개의 병합 후보들이 있다는 결정에 응답하여, 상기 병합 후보들의 현재 리스트에 제로 후보를 포함시키며, 상기 제로 후보의 모션 벡터 성분들이 0 과 동일하고 상기 제로 후보의 참조 인덱스가 0 과 동일하며, 상기 참조 인덱스는 참조 화상 리스트에서의 참조 화상의 로케이션을 표시하는, 비디오 코딩 디바이스.

청구항 9

제 7 항에 있어서,

상기 하나 이상의 프로세서들은, 임의의 백워드 워핑 뷰 합성 (BVSP) 플래그들을 체크하는 일 없이 상기 병합 후보들의 현재 리스트를 생성하는, 비디오 코딩 디바이스.

청구항 10

제 7 항에 있어서,

상기 하나 이상의 프로세서들은 상기 3D 비디오 데이터를 디코딩하도록 구성되고 상기 비디오 블록은 예측 유닛 (PU) 이며,

상기 하나 이상의 프로세서들은,

비트스트림으로부터, 상기 병합 후보들의 현재 리스트에서 선택된 병합 후보를 표시하는 신택스 엘리먼트를 획득하며; 그리고

상기 PU의 예측 샘플들을 생성하기 위해 상기 선택된 병합 후보의 모션 정보를 사용하도록 구성되는, 비디오 코딩 디바이스.

청구항 11

제 7 항에 있어서,

상기 하나 이상의 프로세서들은 상기 3D 비디오 데이터를 인코딩하도록 구성되며,

상기 하나 이상의 프로세서들은,

상기 병합 후보들의 현재 리스트에서 병합 후보를 선택하며; 그리고

상기 선택된 병합 후보의 상기 병합 후보들의 현재 리스트에서의 포지션을 시그널링하도록 구성되는, 비디오 코딩 디바이스.

청구항 12

제 7 항에 있어서,

상기 하나 이상의 프로세서들은, 공간적 병합 후보들에 대한 도출 프로세스를 수행한 후에 그리고 시간적 병합 후보에 대한 도출 프로세스를 수행한 후에, 뷰 간 예측 모션 벡터 후보 (IPMVC) 를, 이용 가능하다면, 상기 병합 후보들의 현재 리스트에 삽입한 후 상기 하나 이상의 조합된 양방향-예측 병합 후보들을 도출하도록 구성되며,

상기 공간적 병합 후보들에 대한 도출 프로세스는 상기 병합 후보들의 현재 리스트에 4 개까지의 공간적 모션 벡터 후보들을 도출 및 삽입하고,

상기 시간적 병합 후보에 대한 도출 프로세스는 시간적 모션 벡터 예측자 (TMVP) 후보를, 이용 가능하다면, 상기 병합 후보들의 현재 리스트에 추가하는, 비디오 코딩 디바이스.

청구항 13

비디오 코딩 디바이스로서,

3차원 (3D) 비디오 데이터를 저장하는 수단; 및

상기 3D 비디오 데이터의 비디오 블록을 코딩하기 위한 병합 후보들의 현재 리스트를 생성하는 수단으로서, 상기 병합 후보들의 현재 리스트에서의 병합 후보들의 최대 수는 6 과 동일하고, 5 개의 양방향-예측 병합 후보들을 갖는 병합 후보들의 리스트들에서 상이한 양방향-예측 병합 후보들의 리스트 0 및 리스트 1 모션 벡터들의 20 개의 가능한 조합들이 존재하는, 상기 병합 후보들의 현재 리스트를 생성하는 수단을 포함하며,

상기 병합 후보들의 현재 리스트를 생성하는 수단은,

상기 병합 후보들의 현재 리스트에서의 초기의 병합 후보들의 수가 5 미만인 것을 결정하는 수단으로서, 0 내지 11 의 조합 인덱스의 각각의 개별 값은 0 내지 3 의 값들의 개별 사전-정의된 조합에 대응하는, 상기 초기의 병합 후보들의 수가 5 미만인 것을 결정하는 수단; 및

상기 병합 후보들의 현재 리스트에서의 상기 병합 후보들의 수가 5 미만이라고 결정하는 것에 응답하여:

상기 조합 인덱스의 상기 개별 값이 상기 병합 후보들의 현재 리스트에서의 상기 초기의 병합 후보들의 수를 상기 병합 후보들의 현재 리스트에서의 상기 초기의 병합 후보들의 수보다 1 적은 수에 의해 곱한 것과 동일하다는 조건, 및 상기 병합 후보들의 현재 리스트가 6 개의 병합 후보들을 갖는다는 조건 중 적어도 하나의 조건이 참일 때까지, 상기 0 내지 11 의 조합 인덱스의 각각의 개별 값에 대해:

상기 병합 후보들의 현재 리스트에서의 제 1 병합 후보가 리스트 0 모션 벡터를 갖는지 여부 및 상기 병합 후보들의 현재 리스트에서의 제 2 병합 후보가 리스트 1 모션 벡터를 갖는지 여부를 결정하는 것

으로서, 상기 제 1 병합 후보 및 상기 제 2 병합 후보는 상기 조합 인덱스의 상기 개별 값에 대응하는 값들의 상기 사전-정의된 조합에 의해 표시되는 상기 병합 후보들의 현재 리스트에서의 포지션들에 있는, 상기 병합 후보들의 현재 리스트에서의 제 1 병합 후보가 리스트 0 모션 벡터를 갖는지 여부 및 상기 병합 후보들의 현재 리스트에서의 제 2 병합 후보가 리스트 1 모션 벡터를 갖는지 여부를 결정하는 것;

상기 제 1 병합 후보가 리스트 0 모션 벡터를 가지고 상기 제 2 병합 후보가 리스트 1 모션 벡터를 갖는다는 결정에 응답하여, 개별의 조합된 양방향-예측 병합 후보를 도출하는 것으로서, 상기 개별의 조합된 양방향-예측 병합 후보는 상기 제 1 병합 후보의 상기 리스트 0 모션 벡터와 상기 제 2 병합 후보의 상기 리스트 1 모션 벡터의 조합이고, 상기 제 1 병합 후보의 상기 모션 벡터와 상기 제 2 병합 후보의 상기 모션 벡터는 상이한 참조 화상 리스트들에서의 화상들을 가리키는, 상기 개별의 조합된 양방향-예측 병합 후보를 도출하는 것; 및

상기 병합 후보들의 현재 리스트에 상기 개별의 조합된 양방향-예측 병합 후보를 포함시키는 것

을 수행하는 수단을 포함하는, 비디오 코딩 디바이스.

청구항 14

제 13 항에 있어서,

상기 병합 후보들의 현재 리스트를 생성하는 수단은,

상기 하나 이상의 양방향-예측 병합 후보들 중 임의의 양방향-예측 병합 후보를 상기 병합 후보들의 현재 리스트에 추가하기 전에 상기 병합 후보들의 현재 리스트에 5 개의 병합 후보들이 있다는 결정에 응답하여, 상기 병합 후보들의 현재 리스트에 제로 후보를 포함시키는 수단으로서, 상기 제로 후보의 모션 벡터 성분들이 0 과 동일하고 상기 제로 후보의 참조 인덱스가 0 과 동일하며, 상기 참조 인덱스는 참조 화상 리스트에서의 참조 화상의 로케이션을 표시하는, 상기 제로 후보를 포함시키는 수단을 더 포함하는, 비디오 코딩 디바이스.

청구항 15

제 13 항에 있어서,

상기 병합 후보들의 현재 리스트를 생성하는 것은, 임의의 백워드 워핑 뷰 합성 (BVSP) 플래그들을 체크하는 일 없이 발생하는, 비디오 코딩 디바이스.

청구항 16

제 13 항에 있어서,

상기 비디오 코딩 디바이스는 상기 3D 비디오 데이터를 디코딩하고 상기 비디오 블록은 예측 유닛 (PU) 이며,

상기 비디오 코딩 디바이스는,

비트스트림으로부터, 상기 병합 후보들의 현재 리스트에서 선택된 병합 후보를 표시하는 신덱스 엘리먼트를 획득하는 수단; 및

상기 PU의 예측 샘플들을 생성하기 위해 상기 선택된 병합 후보의 모션 정보를 사용하는 수단을 더 포함하는, 비디오 코딩 디바이스.

청구항 17

제 13 항에 있어서,

상기 비디오 코딩 디바이스는 상기 3D 비디오 데이터를 인코딩하고,

상기 비디오 코딩 디바이스는,

상기 병합 후보들의 현재 리스트에서 병합 후보를 선택하는 수단; 및

선택된 상기 병합 후보의 상기 병합 후보들의 현재 리스트에서의 포지션을 시그널링하는 수단을 포함하는, 비디오 코딩 디바이스.

청구항 18

제 13 항에 있어서,

상기 병합 후보들의 현재 리스트를 생성하는 것은, 공간적 병합 후보들에 대한 도출 프로세스를 수행한 후에 그리고 시간적 병합 후보에 대한 도출 프로세스를 수행한 후에, 뷰 간 예측 모션 벡터 후보 (IPMVC) 를, 이용 가능하다면, 상기 병합 후보들의 현재 리스트에 삽입한 후 상기 하나 이상의 조합된 양방향-예측 병합 후보들을 도출하는 것을 포함하며,

상기 공간적 병합 후보들에 대한 도출 프로세스는 상기 병합 후보들의 현재 리스트에 4 개까지의 공간적 모션 벡터 후보들을 도출 및 삽입하고,

상기 시간적 병합 후보에 대한 도출 프로세스는 시간적 모션 벡터 예측자 (TMVP) 후보를, 이용 가능하다면, 상기 병합 후보들의 현재 리스트에 추가하는, 비디오 코딩 디바이스.

청구항 19

실행되는 경우 비디오 코딩 디바이스가 3차원 (3D) 비디오 데이터를 코딩하게 하는 명령들을 저장하고 있는 비밀시적 컴퓨터-판독가능 데이터 저장 매체로서,

상기 명령들은 상기 비디오 코딩 디바이스로 하여금,

상기 3D 비디오 데이터의 비디오 블록을 코딩하기 위한 병합 후보들의 현재 리스트를 생성하게 하고, 상기 병합 후보들의 현재 리스트에서의 병합 후보들의 최대 수는 6 과 동일하고, 5 개의 양방향-예측 병합 후보들을 갖는 병합 후보들의 리스트들에서 상이한 양방향-예측 병합 후보들의 리스트 0 및 리스트 1 모션 벡터들의 20 개의 가능한 조합들이 존재하며,

상기 병합 후보들의 현재 리스트를 생성하는 것의 부분으로서, 하나 이상의 프로세서들은,

상기 병합 후보들의 현재 리스트에서의 초기의 병합 후보들의 수가 5 미만인 것을 결정하는 것으로서, 0 내지 11 의 조합 인덱스의 각각의 개별 값은 0 내지 3 의 값들의 개별 사전-정의된 조합에 대응하는, 상기 초기의 병합 후보들의 수가 5 미만인 것을 결정하는 것을 행하고; 그리고

상기 병합 후보들의 현재 리스트에서의 상기 병합 후보들의 수가 5 미만이라고 결정하는 것에 응답하여:

상기 조합 인덱스의 상기 개별 값이 상기 병합 후보들의 현재 리스트에서의 상기 초기의 병합 후보들의 수를 상기 병합 후보들의 현재 리스트에서의 상기 초기의 병합 후보들의 수보다 1 적은 수에 의해 곱한 것과 동일하다는 조건, 및 상기 병합 후보들의 현재 리스트가 6 개의 병합 후보들을 갖는다는 조건 중 적어도 하나의 조건이 참일 때까지, 상기 0 내지 11 의 조합 인덱스의 각각의 개별 값에 대해:

상기 병합 후보들의 현재 리스트에서의 제 1 병합 후보가 리스트 0 모션 벡터를 갖는지 여부 및 상기 병합 후보들의 현재 리스트에서의 제 2 병합 후보가 리스트 1 모션 벡터를 갖는지 여부를 결정하는 것으로서, 상기 제 1 병합 후보 및 상기 제 2 병합 후보는 상기 조합 인덱스의 상기 개별 값에 대응하는 값들의 상기 사전-정의된 조합에 의해 표시되는 상기 병합 후보들의 현재 리스트에서의 포지션들에 있는, 상기 병합 후보들의 현재 리스트에서의 제 1 병합 후보가 리스트 0 모션 벡터를 갖는지 여부 및 상기 병합 후보들의 현재 리스트에서의 제 2 병합 후보가 리스트 1 모션 벡터를 갖는지 여부를 결정하는 것;

상기 제 1 병합 후보가 리스트 0 모션 벡터를 가지고 상기 제 2 병합 후보가 리스트 1 모션 벡터를 갖는다는 결정에 응답하여, 개별의 조합된 양방향-예측 병합 후보를 도출하는 것으로서, 상기 개별의 조합된 양방향-예측 병합 후보는 상기 제 1 병합 후보의 상기 리스트 0 모션 벡터와 상기 제 2 병합 후보의 상기 리스트 1 모션 벡터의 조합이고, 상기 제 1 병합 후보의 상기 모션 벡터와 상기 제 2 병합 후보의 상기 모션 벡터는 상이한 참조 화상 리스트들에서의 화상들을 가리키는, 상기 개별의 조합된 양방향-예측 병합 후보를 도출하는 것; 및

상기 병합 후보들의 현재 리스트에 상기 개별의 조합된 양방향-예측 병합 후보를 포함시키는 것

을 수행하는, 비밀시적 컴퓨터-판독가능 데이터 저장 매체.

청구항 20

제 19 항에 있어서,

상기 비디오 코딩 디바이스로 하여금 상기 병합 후보들의 현재 리스트를 생성하게 하는 것의 부분으로서, 상기 명령들은 상기 비디오 코딩 디바이스로 하여금,

상기 하나 이상의 양방향-예측 병합 후보들 중 임의의 양방향-예측 병합 후보를 상기 병합 후보들의 현재 리스트에 추가하기 전에 상기 병합 후보들의 현재 리스트에 5 개의 병합 후보들이 있다는 결정에 응답하여, 상기 병합 후보들의 현재 리스트에 제로 후보를 포함시키게 하고, 상기 제로 후보의 모션 벡터 성분들이 0 과 동일하고 상기 제로 후보의 참조 인덱스가 0 과 동일하며, 상기 참조 인덱스는 참조 화상 리스트에서의 참조 화상의 로케이션을 표시하는, 비밀시적 컴퓨터-판독가능 데이터 저장 매체.

청구항 21

제 19 항에 있어서,

상기 명령들은 상기 비디오 코딩 디바이스로 하여금, 임의의 백워드 워핑 뷰 합성 (BVSP) 플래그들을 체크하는 일 없이 상기 병합 후보들의 현재 리스트를 생성하게 하는, 비밀시적 컴퓨터-판독가능 데이터 저장 매체.

청구항 22

제 19 항에 있어서,

상기 비디오 블록은 예측 유닛 (PU) 이며,

상기 명령들은 또한 상기 비디오 코딩 디바이스로 하여금,

비트스트림으로부터, 상기 병합 후보들의 현재 리스트에서 선택된 병합 후보를 표시하는 신택스 엘리먼트를 획득하게 하며; 그리고

상기 PU의 예측 샘플들을 생성하기 위해 상기 선택된 병합 후보의 모션 정보를 사용하게 하는, 비밀시적 컴퓨터-판독가능 데이터 저장 매체.

청구항 23

제 19 항에 있어서,

상기 명령들은 또한, 상기 비디오 코딩 디바이스로 하여금,

상기 병합 후보들의 현재 리스트에서 병합 후보를 선택하게 하며; 그리고

상기 선택된 병합 후보의 상기 병합 후보들의 현재 리스트에서의 위치를 시그널링하게 하는, 비밀시적 컴퓨터-판독가능 데이터 저장 매체.

청구항 24

제 19 항에 있어서,

상기 명령들은, 상기 비디오 코딩 디바이스로 하여금, 공간적 병합 후보들에 대한 도출 프로세스를 수행한 후에 그리고 시간적 병합 후보에 대한 도출 프로세스를 수행한 후에, 뷰 간 예측 모션 벡터 후보 (IPMVC) 를, 이용 가능하다면, 상기 병합 후보들의 현재 리스트에 삽입한 후 상기 하나 이상의 조합된 양방향-예측 병합 후보들을 도출하게 하며,

상기 공간적 병합 후보들에 대한 도출 프로세스는 상기 병합 후보들의 현재 리스트에 4 개까지의 공간적 모션 벡터 후보들을 도출 및 삽입하고,

상기 시간적 병합 후보에 대한 도출 프로세스는 시간적 모션 벡터 예측자 (TMVP) 후보를, 이용 가능하다면, 상기 병합 후보들의 현재 리스트에 추가하는, 비밀시적 컴퓨터-판독가능 데이터 저장 매체.

발명의 설명

기술분야

- [0001] 본 출원은 2013년 9월 20일자로 출원된 미국 가 특허 출원 제61/880,737호를 우선권 주장하며, 그 전체 내용은 참조로 본원에 통합된다.
- [0002] 본 개시물은 비디오 코딩 및 압축에 관한 것이고, 더 상세하게는, 3차원 (3D) 비디오를 코딩함에 있어서 사용될 수도 있는 코딩 기법들에 관한 것이다.

배경기술

- [0003] 디지털 비디오 능력들은 디지털 텔레비전들, 디지털 직접 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인 정보 단말기들 (PDA들), 랩톱 또는 데스크톱 컴퓨터들, 디지털 카메라들, 디지털 레코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 무선 전화기들, 비디오 원격회의 디바이스들 등을 포함한 넓은 범위의 디바이스들에 통합될 수 있다. 디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263 또는 ITU-T H.264/MPEG-4, 파트 10, 고급 비디오 코딩 (Advanced Video Coding, AVC) 에 의해 규정된 표준들, 고 효율 비디오 코딩 (High Efficiency Video Coding, HEVC) 표준, 및 이러한 표준들의 확장본들에 기재된 것들과 같은 비디오 압축 기법들을 구현하여, 디지털 비디오 정보를 더 효율적으로 송신하며, 수신하고 저장한다.
- [0004] 비디오 압축 기법들은 공간적 (화상 내) 예측 및/또는 시간적 (화상 간) 예측을 수행하여 비디오 시퀀스들에 내재하는 리던던시를 감소시키거나 또는 제거한다. 블록 기반 비디오 코딩을 위해, 비디오 슬라이스가 비디오 블록들로 구획화될 수도 있으며, 그 비디오 블록들은 또한 트리블록들 (treeblocks), 코딩 단위 (coding unit, CU) 들 및/또는 코딩 노드들이라고 지칭될 수도 있다. 화상의 인트라 코딩식 (intra-coded; I) 슬라이스에서의 비디오 블록들은 동일한 화상의 이웃 블록들에서의 참조 샘플들에 관한 공간적 예측을 이용하여 인코딩된다. 화상의 인터 코딩식 (inter-coded; P 또는 B) 슬라이스에서의 비디오 블록들은 동일한 화상의 이웃 블록들에서의 참조 샘플들에 관한 공간적 예측 또는 다른 참조 화상들에서의 참조 샘플들에 관한 시간적 예측을 이용할 수도 있다. 화상들은 프레임들이라고 지칭될 수도 있고, 참조 화상들은 참조 프레임들이라고 지칭될 수도 있다.
- [0005] 멀티뷰 코딩 비트스트림이, 예컨대, 다수의 관점들에서 뷰들을 인코딩함으로써 생성될 수도 있다. 멀티-뷰 코딩은 디코더가 상이한 뷰들 간을 선택하는 것, 또는 아마도 다수의 뷰들을 랜더링하는 것을 허용할 수도 있다. 더구나, 개발되어 있는 또는 개발 중에 있는 몇몇 3차원 (3D) 비디오 기법들과 표준들은, 멀티-뷰 코딩 양태들을 이용한다. 3차원 비디오는 또한 "3DV"라고 지칭된다.
- [0006] 예를 들어, 상이한 뷰들이 3D 비디오를 지원하기 위해 좌안 및 우안 뷰들을 송신할 수도 있다. 대안으로, 몇몇 3D 비디오 코딩 프로세스들이 이른바 멀티뷰 플러스 깊이 (multi-view plus depth) 코딩을 적용할 수도 있다. 멀티뷰 플러스 깊이 코딩에서, 3D 비디오 비트스트림이 텍스처 뷰 성분들뿐만 아니라, 깊이 뷰 성분들도 포함할 수도 있다. 예를 들어, 각각의 뷰는 하나의 텍스처 뷰 성분과 하나의 깊이 뷰 성분을 포함할 수도 있다.
- [0007] 현재, VCEG 및 MPEG의 JCT-3C (Joint Collaboration Team on 3D video Coding) 가 "고 효율 비디오 코딩 (HEVC)"이라고 지칭되는 신흥 표준에 기초하여 3D 비디오 표준을 개발하고 있는데, 그 표준화 노력의 일부는 HEVC에 기초한 멀티뷰 비디오 코덱 (MV-HEVC) 의 표준화를 포함하고 다른 일부는 HEVC에 기초한 3D 비디오 코딩 (3D-HEVC) 를 위한 것이다. 3D-HEVC는, 코딩 단위 및/또는 예측 유닛 레벨에서의 코딩 도구들을 포함하는 새로운 코딩 도구들을, 텍스처 뷰 및 깊이 뷰 양쪽 모두를 위해, 포함하고 지원할 수도 있다.

발명의 내용

과제의 해결 수단

- [0008] 대체로, 본 개시물은 고 효율 비디오 코딩 (HEVC) 코덱을 이용한 둘 이상의 뷰들의 코딩을 포함하는, 고급 코덱들에 기초한 3차원 (3D) 비디오 코딩에 관한 것이다. 예를 들면, 본 개시물의 몇몇 예들은 조합된 양방향-예측 (combined bi-predictive) 병합 후보들에 관련된 기법들을 설명한다. 몇몇 이러한 예들에서, 병합 후보들의 리스트를 생성하는 부분으로서, 비디오 코더가 그 리스트에서의 병합 후보들의 수가 5 미만인지의 여부를 결정한다. 만약 그렇다면, 비디오 코더는 하나 이상의 조합된 양방향-예측 병합 후보들을 도출한다. 비디오

코더는 병합 후보들의 리스트에 하나 이상의 조합된 양방향-예측 병합 후보들을 포함시킨다.

[0009] 하나의 양태에서, 본 개시물은 3D 비디오 데이터를 코딩하기 위한 방법을 설명한다. 그 방법은 3D 비디오 데이터의 비디오 블록을 코딩하기 위한 병합 후보들의 리스트를 생성하는 단계를 포함한다. 병합 후보들의 리스트에서의 병합 후보들의 최대 수가 6과 동일하고, 병합 후보들의 리스트를 생성하는 단계는, 병합 후보들의 리스트에서의 병합 후보들의 수가 5 미만인지의 여부를 결정하는 단계; 및 병합 후보들의 리스트에서의 병합 후보들의 수가 5 미만이라는 결정에 응답하여, 하나 이상의 조합된 양방향-예측 병합 후보들을 도출하는 단계로서, 하나 이상의 조합된 양방향-예측 병합 후보들 중 각각의 개별 (each respective) 조합된 양방향-예측 병합 후보는 병합 후보들의 리스트에서의 병합 후보들의 개별 쌍에 이미 대응하며, 각각의 조합된 양방향-예측 병합 후보는 개별 쌍의 제 1 병합 후보의 모션 벡터 및 개별 쌍의 제 2 병합 후보의 모션 벡터의 조합이며, 제 1 병합 후보의 모션 벡터 및 제 2 병합 후보의 모션 벡터는 상이한 참조 화상 리스트들에서의 화상들을 가리키는, 상기 하나 이상의 조합된 양방향-예측 병합 후보들을 도출하는 단계를 포함한다. 그 방법은 병합 후보들의 리스트에 하나 이상의 조합된 양방향-예측 병합 후보들을 포함시키는 단계를 또한 포함한다.

[0010] 다른 양태에서, 본 개시물은, 3차원 비디오 데이터를 저장하도록 구성된 데이터 저장 매체; 및 3D 비디오 데이터의 비디오 블록을 코딩하기 위한 병합 후보들의 리스트를 생성하도록 구성된 하나 이상의 프로세서들을 포함하는 비디오 코딩 디바이스를 설명하는데, 병합 후보들의 리스트에서의 병합 후보들의 최대 수는 6과 동일하고, 병합 후보들의 리스트를 생성하는 부분으로서 하나 이상의 프로세서들은, 병합 후보들의 리스트에서의 병합 후보들의 수가 5 미만인지의 여부를 결정하고; 병합 후보들의 리스트에서의 병합 후보들의 수가 5 미만이라는 결정에 응답하여, 하나 이상의 조합된 양방향-예측 병합 후보들을 도출하도록 구성되며, 하나 이상의 조합된 양방향-예측 병합 후보들 중 각각의 개별 조합된 양방향-예측 병합 후보는 병합 후보들의 리스트에서의 병합 후보들의 개별 쌍에 이미 대응하며, 각각의 조합된 양방향-예측 병합 후보는 개별 쌍의 제 1 병합 후보의 모션 벡터 및 개별 쌍의 제 2 병합 후보의 모션 벡터의 조합이며, 제 1 병합 후보의 모션 벡터 및 제 2 병합 후보의 모션 벡터는 상이한 참조 화상 리스트들에서의 화상들을 가리킨다. 하나 이상의 프로세서들은 병합 후보들의 리스트에 하나 이상의 조합된 양방향-예측 병합 후보들을 포함시키도록 구성된다.

[0011] 다른 양태에서, 본 개시물은 3D 비디오 데이터의 비디오 블록을 코딩하기 위한 병합 후보들의 리스트를 생성하는 수단을 포함하는 비디오 코딩 디바이스를 설명한다. 병합 후보들의 리스트에서의 병합 후보들의 최대 수가 6과 동일하고, 병합 후보들의 리스트를 생성하는 수단은, 병합 후보들의 리스트에서의 병합 후보들의 수가 5 미만인지의 여부를 결정하는 수단; 및 병합 후보들의 리스트에서의 병합 후보들의 수가 5 미만이라는 결정에 응답하여, 하나 이상의 조합된 양방향-예측 병합 후보들을 도출하는 수단으로서, 하나 이상의 조합된 양방향-예측 병합 후보들 중 각각의 개별 조합된 양방향-예측 병합 후보는 병합 후보들의 리스트에서의 병합 후보들의 개별 쌍에 이미 대응하며, 각각의 조합된 양방향-예측 병합 후보는 개별 쌍의 제 1 병합 후보의 모션 벡터 및 개별 쌍의 제 2 병합 후보의 모션 벡터의 조합이며, 제 1 병합 후보의 모션 벡터 및 제 2 병합 후보의 모션 벡터는 상이한 참조 화상 리스트들에서의 화상들을 가리키는, 상기 하나 이상의 조합된 양방향-예측 병합 후보들을 도출하는 수단을 포함한다. 비디오 코딩 디바이스는 병합 후보들의 리스트에 하나 이상의 조합된 양방향-예측 병합 후보들을 포함시키는 수단을 또한 포함한다.

[0012] 다른 양태에서, 본 개시물은 실행되는 경우 비디오 코딩 디바이스가 3D 비디오 데이터를 코딩하게 하는 명령들을 저장하고 있는 컴퓨터-판독가능 데이터 저장 매체를 설명하는데, 그 명령들은 비디오 코딩 디바이스로 하여금, 3D 비디오 데이터의 비디오 블록을 코딩하기 위한 병합 후보들의 리스트를 생성하게 한다. 병합 후보들의 리스트에서의 병합 후보들의 최대 수는 6과 동일하다. 병합 후보들의 리스트를 생성하게 하는 것은, 병합 후보들의 리스트에서의 병합 후보들의 수가 5 미만인지의 여부를 결정하게 하는 것; 및 병합 후보들의 리스트에서의 병합 후보들의 수가 5 미만이라는 결정에 응답하여: 하나 이상의 조합된 양방향-예측 병합 후보들을 도출하게 하는 것으로서, 하나 이상의 조합된 양방향-예측 병합 후보들 중 각각의 개별 조합된 양방향-예측 병합 후보는 병합 후보들의 리스트에서의 병합 후보들의 개별 쌍에 이미 대응하며, 각각의 조합된 양방향-예측 병합 후보는 개별 쌍의 제 1 병합 후보의 모션 벡터 및 개별 쌍의 제 2 병합 후보의 모션 벡터의 조합이며, 제 1 병합 후보의 모션 벡터 및 제 2 병합 후보의 모션 벡터는 상이한 참조 화상 리스트들에서의 화상들을 가리키는, 상기 하나 이상의 조합된 양방향-예측 병합 후보들을 도출하게 하는 것; 및 상기 병합 후보들의 리스트에 상기 하나 이상의 조합된 양방향-예측 병합 후보들을 포함시키게 하는 것을 포함한다.

[0013] 하나 이상의 예들의 세부사항들은 첨부 도면들 및 다음의 설명에서 언급된다. 다른 특징들, 목적들, 및 장점들은 상세한 설명, 도면들, 및 청구항들로부터 명확하게 될 것이다.

도면의 간단한 설명

- [0014] 도 1은 본 개시물의 기법들을 이용할 수도 있는 일 예의 비디오 코딩 시스템을 도시하는 블록도이다.
- 도 2는 병합 리스트에 대한 잠재적 후보들인 공간적 이웃들을 보여주는 개념도이다.
- 도 3은 현재 코딩 단위에 관하여 공간적 및 시간적 이웃 블록들을 도시하는 개념도이다.
- 도 4는 뷰 간 예측된 모션 벡터 후보 (inter-view predicted motion vector candidate) 의 도출 프로세스의 일 예를 보여준다.
- 도 5는 백워드 워핑 뷰 합성 예측 (backward-warping view synthesis prediction, BVSP) 을 수행하기 위한 참조 뷰로부터의 깊이 블록 도출을 예시하는 개념도이다.
- 도 6은 하나의 8x8 깊이 블록의 네 개의 코너 화소들을 예시하는 개념도이다.
- 도 7은 3D-HEVC에서의 10CandIdx 및 11CandIdx의 일 예의 사양을 제공하는 표이다.
- 도 8은 본 개시물의 기법들을 구현할 수도 있는 일 예의 비디오 인코더를 도시하는 블록도이다.
- 도 9은 본 개시물의 기법들을 구현할 수도 있는 일 예의 비디오 디코더를 도시하는 블록도이다.
- 도 10a는 본 개시물의 하나 이상의 기법들에 따른, 3D 비디오에 연관된 데이터를 인코딩하는 비디오 인코더의 일 예의 동작을 도시하는 흐름도이다.
- 도 10b는 본 개시물의 하나 이상의 기법들에 따른, 3D 비디오에 연관된 데이터를 디코딩하는 비디오 디코더의 일 예의 동작을 도시하는 흐름도이다.
- 도 11은 본 개시물의 하나 이상의 기법들에 따른, 현재 블록에 대한 병합 후보 리스트를 구축하는 일 예의 동작의 제 1 부분을 도시하는 흐름도이다.
- 도 12는 본 개시물의 하나 이상의 기법들에 따른, 현재 블록에 대한 병합 후보 리스트를 구축하는 도 11의 예의 동작의 제 2 부분을 도시하는 흐름도이다.
- 도 13은 본 개시물의 하나 이상의 기법들에 따른, 조합된 양방향-예측 병합 후보들에 대한 일 예의 도출 프로세스를 도시하는 흐름도이다.
- 도 14a는 본 개시물의 하나 이상의 기법들에 따른, 비디오 블록을 인코딩하는 비디오 인코더의 일 예의 동작을 도시하는 흐름도이다.
- 도 14b는 본 개시물의 하나 이상의 기법들에 따른, 비디오 블록 디코딩하는 비디오 디코더의 일 예의 동작을 도시하는 흐름도이다.

발명을 실시하기 위한 구체적인 내용

- [0015] 비디오 인코딩이 비디오 데이터를 인코딩된 비디오 데이터로 변환하는 프로세스이다. 대체로, 비디오 디코딩은 그 변환을 뒤집음으로써, 비디오 데이터를 복원한다. 비디오 인코딩과 비디오 디코딩은 양쪽 모두가 비디오 코딩으로서 지칭될 수도 있다. 블록 기반 비디오 코딩이, 적어도 부분적으로, 화상들 내의 비디오 데이터의 블록들에 대해 동작하는 유형의 비디오 코딩이다.
- [0016] 인터 예측이 비디오 인코더가, 참조 화상의 샘플들에 기초하여, 현재 블록 (즉, 비디오 인코더가 현재 코딩하고 있는 블록) 에 대한 예측 블록을 결정하는 비디오 코딩 기법이다. 참조 화상은 비디오 인코더가 현재 코딩하고 있는 화상과는 다른 화상이다. 비디오 인코더는, 비트스트림 내에, 그 블록에 대한 잔차 데이터를 표현하는 데이터를 포함시킬 수도 있다. 블록에 대한 잔차 데이터는 현재 블록 및 예측 블록 간의 차이들을 나타낸다. 블록에 대한 모션 벡터가 현재 블록 및 예측 블록 간의 공간적 변위를 나타낼 수도 있다. 참조 인덱스가 현재 화상을 코딩함에 있어서 사용할 수 있는 참조 화상들의 리스트 내의 참조 화상의 로케이션을 나타낼 수도 있다. 참조 인덱스들은 "참조 화상 인덱스들"을 또한 지칭할 수도 있다. 비디오 디코더가 현재 블록에 대한 예측 블록을 결정하기 위해 현재 블록의 모션 벡터를 사용할 수도 있다. 더욱이, 비디오 디코더는 예측 블록과 현재 블록에 대한 잔차 데이터를 결합하여 현재 블록을 복원할 수도 있다.
- [0017] 양방향 인터 예측에서, 비디오 인코더는 현재 블록에 대한 두 개의 예측 블록들을 결정한다. 따라서, 비디

오 인코더는 현재 블록에 대한 두 개의 모션 벡터들을 또한 결정한다. 현재 블록에 대한 두 개의 예측 블록들은 상이한 참조 화상들에 있을 수도 있다. 그러므로, 양방향 인터 예측에서, 비디오 인코더는 현재 블록에 대해 두 개의 참조 인덱스들 (즉, 제 1 참조 인덱스 및 제 2 참조 인덱스) 을 결정할 수도 있다. 제 1 및 제 2 참조 인덱스들은 제 1 및 제 2 참조 화상 리스트 내의 참조 화상들의 로케이션들을 각각 표시한다. 현재 블록에 대한 잔차 데이터는 현재 블록에 대한 두 개의 예측 블록들에 기초하는 현재 블록 및 합성된 예측 블록 간의 차이들을 나타낼 수도 있다.

[0018] 현재 블록의 모션 벡터들은 현재 블록에 공간적으로 또는 시간적으로 이웃하는 블록들 (즉, 이웃 블록들) 의 모션 벡터들과 유사할 수도 있다. 그러므로, 비디오 인코더가 현재 블록의 모션 벡터들 및 참조 인덱스들을 명시적으로 시그널링하는 것이 필요할 수도 있다. 오히려, 비디오 인코더는 현재 블록에 대한 병합 후보들의 리스트 (즉, "병합 후보 리스트" 또는 "머지 후보 리스트") 를 결정할 수도 있다. 병합 후보들의 각각은 모션 정보의 세트 (예컨대, 하나 이상의 모션 벡터들, 하나 이상의 참조 인덱스들 등) 를 특정한다. 병합 후보들의 리스트는 이웃 블록들 중 상이한 이웃 블록들의 모션 정보를 각각 특정하는 하나 이상의 병합 후보들을 포함할 수도 있다. 이웃 블록들은 공간적 이웃 블록들 및/또는 시간적 이웃 블록을 포함할 수도 있다. 본 개시물은 공간적 이웃 블록들에 기초한 병합 후보들을 공간적 병합 후보들이라고 지칭할 수도 있다. 본 개시물은 시간적 이웃 블록들에 기초한 병합 후보들을 시간적 병합 후보들이라고 지칭할 수도 있다. 몇몇 예들에서, 병합 후보들의 리스트에서의 두 개의 병합 후보들은 동일한 모션 정보를 가질 수도 있다. 비디오 인코더는 병합 후보들 중 하나를 선택할 수도 있고 선택된 병합 후보의 병합 후보 리스트 내의 포지션을 표시하는 신택스 엘리먼트를 시그널링할 수도 있다.

[0019] 비디오 디코더는 동일한 병합 후보 리스트 (즉, 비디오 인코더에 의해 결정된 병합 후보 리스트에 중복하는 병합 후보 리스트) 를 생성할 수도 있고, 시그널링된 신택스 엘리먼트의 수신에 기초하여, 선택된 병합 후보를 결정할 수도 있다. 비디오 디코더 (30) 는 그 다음에 선택된 병합 후보의 모션 정보를 현재 블록의 모션 정보로서 사용할 수도 있다. 이런 식으로, 현재 블록은 이웃 블록들 중 하나의 이웃 블록의 모션 정보를 상속받을 수도 있다.

[0020] 몇몇 환경들에서, 이웃 블록의 모션 정보는 이용 불가능할 수도 있다. 예를 들어, 이웃 블록은 인트라 예측을 사용하여 코딩될 수도 있거나, 이웃 블록은 상이한 슬라이스에 있을 수도 있거나, 또는 이웃 블록은 단순히 존재하지 않을 수도 있다. 그러므로, 요구된 수 (예컨대, 슬라이스 헤더에서 표시될 수도 있는 병합 후보들의 최대 수) 보다 더 적은 수의 병합 후보들이 현재 블록에 대한 병합 후보 리스트 내에 있을 수도 있다. 따라서, 비디오 코더 (예컨대, 비디오 인코더 또는 비디오 디코더) 가 현재 블록에 대한 병합 후보 리스트를 생성하는 경우, 비디오 코더는 현재 블록에 대한 병합 후보 리스트에 하나 이상의 인공 병합 후보들을 포함시킴으로써 현재 블록에 대한 병합 후보 리스트가 소망의 수의 병합 후보들을 포함하는 것을 보장할 수도 있다. 인공 병합 후보들은 임의의 공간적 또는 시간적 이웃 블록의 모션 정보를 반드시 특정하지는 않는 병합 후보들이다.

[0021] 인공 병합 후보들은 하나 이상의 조합된 양방향-예측 병합 후보들을 포함할 수도 있다. 위에서 나타낸 바와 같이, 병합 후보가 두 개의 모션 벡터들 및 두 개의 참조 인덱스들을 특정할 수도 있다. 조합된 양방향-예측 병합 후보가 현재 블록에 대한 병합 후보들의 리스트에서의 병합 후보들의 개별 쌍에 이미 대응한다. 구체적으로는, 조합된 양방향-예측 병합 후보는, 이용 가능하다면 개별 쌍의 제 1 병합 후보의 모션 벡터 및 참조 인덱스와, 이용 가능하다면 개별 쌍의 제 2 병합 후보의 모션 벡터 및 참조 인덱스의 조합이다. 제 1 병합 후보의 모션 벡터 및 제 2 병합 후보의 모션 벡터는 상이한 참조 화상 리스트들에서의 화상들을 가리킨다. 따라서, 조합된 양방향-예측 병합 후보들은 상이한 기존 병합 후보들 (예컨대, 조합된 양방향-예측 병합 후보들과는 다른 병합 후보들, 이를테면 공간적 또는 시간적 병합 후보들) 과는 상이한 모션 벡터들/참조 인덱스들의 조합들에 대응할 수도 있다. 예를 들어, 제 1 병합 후보의 RefPicList0 모션 정보와 제 2 병합 후보의 RefPicList1 모션 정보가 양쪽 모두 이용 가능하고 동일하지 않은 (즉, 참조 화상들이 상이하거나 또는 모션 벡터들이 상이한) 경우, 하나의 조합된 양방향-예측 병합 후보가 구축된다. 그렇지 않으면, 다음의 개별 쌍이 체크된다.

[0022] HEVC 규격의 몇몇 버전들에서, 병합 후보들의 리스트에서의 병합 후보들의 요구된 수의 최대 값은 5이다. 더욱이, 몇몇 경우들에서, 병합 후보들의 리스트에서의 병합 후보들의 소망의 수는 5이다. 그러므로, 조합된 양방향-예측 병합 후보들을 병합 후보 리스트에 포함시키기 전에 병합 후보 리스트에 5 개 미만의 병합 후보들이 있다면, 조합된 양방향-예측 병합 후보들에서 사용 가능한 모션 벡터들의 열두 (즉, 4*3) 개까지의 가능한 조합들이 있다. 개별 쌍의 선택 (즉, 어떤 후보가 제 1 후보이고 어떤 후보가 제 2 후보인지) 은 다음의 표

에 도시된 바와 같이 HEVC에서 미리 정의된다:

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

[0023]

[0024]

[0025]

[0026]

[0027]

[0028]

[0029]

[0030]

위의 표에서, 10CandIdx는 선택된 제 1 기존 병합 후보의 인덱스를 나타내며, 11CandIdx는 선택된 제 2 기존 병합 후보의 인덱스를 나타내고, combIdx는 구축되는 조합된 양방향-예측 후보 인덱스를 나타낸다.

다층 비디오 코딩이 다수의 계층들에 걸친 비디오 코딩을 허용한다. 다층 비디오 코딩은 스케일러블 비디오 코딩, 멀티-뷰 비디오 코딩, 및 3차원 (3D) 비디오 코딩을 구현하는데 사용될 수도 있다. 멀티-뷰 비디오 코딩 및 3D 비디오 코딩에서, 계층들의 각각은 상이한 관점에 대응할 수도 있다. 몇몇 비디오 코딩 표준들에서, 병합 후보 리스트에서의 병합 후보들의 요구된 수는 단층 비디오 코딩을 사용하는 경우보다 다층 비디오 코딩을 사용하는 경우 더 많다. 병합 후보들의 더 많은 수가 상이한 뷰들에서의 블록들의 모션 정보를 특정하는 병합 후보들을 수용하기 위하여 허용될 수도 있다.

단층 비디오 코딩의 경우에서처럼, 비디오 코더가 다층 코딩을 사용하고 병합 후보 리스트에서의 병합 후보들의 수가 병합 후보들의 소망의 수 미만인 경우, 비디오 코더는 하나 이상의 조합된 양방향-예측 병합 후보들을 생성할 수도 있다. 그러나, 다층 코딩을 사용하는 경우의 병합 후보들의 더 많은 수로 인해, 조합된 양방향-예측 병합 후보들에서 이용 가능한 모션 벡터들의 조합들의 수는 더 많다. 예를 들어, 병합 후보들의 요구된 수가 6이면, 조합된 양방향-예측 병합 후보들에서 사용 가능한 모션 벡터들의 이십 (5*4) 개까지의 가능한 조합들이 있다.

비디오 코더가 조합된 양방향-예측 병합 후보를 병합 후보들의 특정 쌍들로부터 생성하지 못할 수도 있다. 예를 들어, 비디오 코더는 병합 후보들 중 하나의 병합 후보만이 단일 모션 벡터 및 단일 참조 인덱스를 갖는다면 조합된 양방향-예측 병합 후보를 생성하지 못할 수도 있다. 조합된 양방향-예측 병합 후보가 병합 후보들의 특정 쌍의 모션 정보로부터 생성될 수 있는지의 여부를 결정하기 위하여, 비디오 코더는 메모리로부터 병합 후보들의 쌍에 관한 정보를 추출하는 것이 필요할 수도 있다.

메모리로부터 정보를 추출하는 것은 다른 코딩 프로세스들에 비하여 비교적 느린 프로세스일 수도 있다. 더구나, 메모리에 대한 액세스는 전력을 요구한다. 그러므로, 메모리에 대한 액세스들의 수를 제한하는 것은 바람직할 수도 있다. 조합된 양방향-예측 병합 후보들에서 사용 가능한 모션 벡터들의 조합들의 수가 증가함에 따라, 메모리로부터 추출될 필요가 있는 정보의 양은 증가한다. 따라서, 멀티-뷰 비디오 코딩에 연관된 병합 후보들의 요구된 수에서의 증가는 비디오 코딩 프로세스를 상당히 느리게 할 수도 있고 그렇지 않으면 사용될 것보다 더 많은 전력을 사용할 수도 있다.

그러므로, 본 개시물의 일 예에 따라, 비디오 코더가 3D 비디오의 비디오 블록을 코딩하기 위한 병합 후보들의 리스트를 메모리에 대한 액세스를 제한할 수 있는 방식으로 생성할 수도 있다. 더욱이, 이 예에서, 병합 후보들의 리스트를 생성하는 부분으로서, 비디오 코더는 리스트에서의 병합 후보들의 수가 5 미만인지의 여부를 결정할 수도 있다. 리스트에서의 병합 후보들의 수가 5 미만이라는 결정에 응답하여, 비디오 코더는 하나 이상의 조합된 양방향-예측 병합 후보들을 도출할 수도 있다. 이 예에서, 하나 이상의 조합된 양방향-예측 병합 후보들 중 각각의 개별 조합된 양방향-예측 병합 후보는 그 리스트에서의 병합 후보들의 개별 쌍에 이미 대응한다. 더욱이, 이 예에서, 각각의 조합된 양방향-예측 병합 후보는 개별 쌍의 제 1 병합 후보의 모션 벡터 및 개별 쌍의 제 2 병합 후보의 모션 벡터의 조합이다. 이 예에서, 제 1 병합 후보의 모션 벡터 및 제 2 병합 후보의 모션 벡터는 상이한 참조 화상 리스트들에서의 화상들을 가리킨다. 비디오 코더는 그 리스트에 하나 이상의 조합된 양방향-예측 병합 후보들을 포함시킬 수도 있다. 몇몇 예들에서, 리스트에서의 병합 후보들의 최대 수가 5보다 더 크다 (예컨대, 6과 동일하다). 이 예의 프로세스의 효과는, 심지어 리스트에서의 병합 후보들의 최대 수가 6 이상이라도, 조합들의 수가 여전히 12로 제한된다는 것이다. 이는 메모리로부터 추출되는 정보의 양을 감소시킴으로써 코딩 프로세스의 가속화를 도울 수도 있고 전력을 또한 절약할 수도 있다.

도 1은 본 개시물의 기법들을 이용할 수도 있는 일 예의 비디오 코딩 시스템 (10) 을 도시하는 블록도이다. 본원에서 설명되는 바와 같이, "비디오 코더"라는 용어는 비디오 인코더들 및 비디오 디코더들 양쪽 모두를 일반적으로 지칭한다. 본 개시물에서, "비디오 코딩" 또는 "코딩"이란 용어들은 비디오 인코딩 또는 비디오

디코딩을 일반적으로 지칭할 수도 있다.

- [0031] 도 1에 도시된 바와 같이, 비디오 코딩 시스템 (10) 은 소스 디바이스 (12) 와 목적지 디바이스 (14) 를 구비한다. 소스 디바이스 (12) 는 인코딩된 비디오 데이터를 생성한다. 따라서, 소스 디바이스 (12) 는 비디오 인코딩 디바이스 또는 비디오 인코딩 장치라고 지칭될 수도 있다. 목적지 디바이스 (14) 는 소스 디바이스 (12) 에 의해 생성된 인코딩된 비디오 데이터를 디코딩할 수도 있다. 따라서, 목적지 디바이스 (14) 는 비디오 디코딩 디바이스 또는 비디오 디코딩 장치라고 지칭될 수도 있다. 소스 디바이스 (12) 와 목적지 디바이스 (14) 는 비디오 코딩 디바이스들 또는 비디오 코딩 장치들의 예들일 수도 있다.
- [0032] 소스 디바이스 (12) 와 목적지 디바이스 (14) 는 데스크톱 컴퓨터들, 모바일 컴퓨팅 디바이스들, 노트북 (예컨대, 랩톱) 컴퓨터들, 태블릿 컴퓨터들, 셋톱 박스들, 이른바 "스마트" 폰들과 같은 전화기 핸드셋들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 콘솔들, 차량내 컴퓨터들 등을 포함한 다양한 범위의 디바이스들을 포함할 수도 있다.
- [0033] 목적지 디바이스 (14) 는 소스 디바이스 (12) 로부터의 인코딩된 비디오 데이터를 채널 (16) 을 통해 수신할 수도 있다. 채널 (16) 은 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 인코딩된 비디오 데이터를 이동시킬 수 있는 하나 이상의 매체들 또는 디바이스들을 포함할 수도 있다. 하나의 예에서, 채널 (16) 은 소스 디바이스 (12) 가 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로 직접 실시간으로 송신하는 것을 가능하게 하는 하나 이상의 통신 매체들을 포함할 수도 있다. 이 예에서, 소스 디바이스 (12) 는 인코딩된 비디오 데이터를 통신 표준, 이를테면 무선 통신 프로토콜에 따라 변조할 수도 있고, 변조된 비디오 데이터를 목적지 디바이스 (14) 로 송신할 수도 있다. 하나 이상의 통신 매체들은 무선 및/또는 유선 통신 매체들, 이를테면 무선 주파수 (RF) 스펙트럼 또는 하나 이상의 물리적 송신 라인들을 포함할 수도 있다. 하나 이상의 통신 매체들은 패킷 기반 네트워크, 이를테면 로컬 영역 네트워크, 광역 네트워크, 또는 글로벌 네트워크 (예컨대, 인터넷) 의 일부를 형성할 수도 있다. 채널 (16) 은 다양한 유형들의 디바이스들, 이를테면 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로의 통신을 용이하게 하는 다른 장비를 포함할 수도 있다.
- [0034] 다른 예에서, 채널 (16) 은 소스 디바이스 (12) 에 의해 생성된 인코딩된 비디오 데이터를 저장하는 저장 매체를 포함할 수도 있다. 이 예에서, 목적지 디바이스 (14) 는 디스크 액세스 또는 카드 액세스를 통해 저장 매체에 액세스할 수도 있다. 저장 매체는 블루 레이 디스크들, DVD들, CD-ROM들, 플래시 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 다른 적합한 디지털 저장 매체들과 같은 다양한 국부적으로 액세스되는 데이터 저장 매체들을 포함할 수도 있다.
- [0035] 추가의 예에서, 채널 (16) 은 소스 디바이스 (12) 에 의해 생성된 인코딩된 비디오 데이터를 저장하는 파일 서버 또는 다른 중간 저장 디바이스들을 포함할 수도 있다. 이 예에서, 목적지 디바이스 (14) 는 파일 서버 또는 다른 중간 저장 디바이스에 저장된 인코딩된 비디오 데이터를 스트리밍 또는 다운로드를 통해 액세스할 수도 있다. 파일 서버는 인코딩된 비디오 데이터를 저장하고 그 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로 송신할 수 있는 유형의 서버일 수도 있다. 예의 파일 서버들은 웹 서버들 (예컨대, 웹사이트용), 파일 전송 프로토콜 (FTP) 서버들, 네트워크 부속 스토리지 (network attached storage; NAS) 디바이스들, 및 로컬 디스크 드라이브들을 포함한다.
- [0036] 목적지 디바이스 (14) 는 표준 데이터 접속, 이를테면 인터넷 접속을 통해, 인코딩된 비디오 데이터에 액세스할 수도 있다. 예의 유형들의 데이터 접속들은 파일 서버 상에 저장된 인코딩된 비디오 데이터에 액세스하기에 적합한 무선 채널들 (예컨대, Wi-Fi 접속들), 유선 접속들 (예컨대, DSL, 케이블 모뎀 등), 또는 양쪽 모두의 조합들을 포함할 수도 있다. 파일 서버로부터의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 양쪽 모두의 조합일 수도 있다.
- [0037] 본 개시물의 기법들은 무선 애플리케이션들 또는 설정 (setting) 들로 제한되지 않는다. 그 기법들은, 다양한 멀티미디어 애플리케이션들, 이를테면 OTA (over-the-air) 텔레비전 브로드캐스트들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, 예컨대, 인터넷을 통한 스트리밍 비디오 송신들 중 임의의 것의 지원 하의 비디오 코딩, 데이터 저장 매체 상의 저장을 위한 비디오 데이터의 인코딩, 데이터 저장 매체 상에 저장된 비디오 데이터의 디코딩, 또는 다른 애플리케이션들에 적용될 수도 있다. 몇몇 예들에서, 비디오 코딩 시스템 (10) 은 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅, 및/또는 화상 통화와 같은 애플리케이션들을 지원하기 위해 단방향 또는 양방향 비디오 송신을 지원하도록 구성될 수도 있다.

- [0038] 도 1의 예에서, 소스 디바이스 (12) 는 비디오 소스 (18), 비디오 인코더 (20), 및 출력 인터페이스 (22) 를 구비한다. 몇몇 예들에서, 출력 인터페이스 (22) 는 변조기/복조기 (모뎀) 및/또는 송신기를 구비할 수도 있다. 비디오 소스 (18) 는 비디오 캡처 디바이스, 예컨대, 비디오 카메라, 이전에 캡처된 비디오 데이터를 포함한 비디오 아카이브, 비디오 콘텐츠 제공자로부터 비디오 데이터를 수신하는 비디오 피드 인터페이스, 및/또는 비디오 데이터를 생성하는 컴퓨터 그래픽 시스템, 또는 비디오 데이터의 이러한 소스들의 조합을 포함할 수도 있다.
- [0039] 비디오 인코더 (20) 는 비디오 소스 (18) 로부터의 비디오 데이터를 인코딩할 수도 있다. 몇몇 예들에서, 소스 디바이스 (12) 는 출력 인터페이스 (22) 를 통해 목적지 디바이스 (14) 로 인코딩된 비디오 데이터를 직접 송신할 수도 있다. 다른 예들에서, 인코딩된 비디오 데이터는 디코딩 및/또는 플레이백을 위한 목적지 디바이스 (14) 에 의한 나중의 액세스를 위해 저장 매체 또는 파일 서버 상에 또한 저장될 수도 있다.
- [0040] 도 1의 예에서, 목적지 디바이스 (14) 는 입력 인터페이스 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (32) 를 구비한다. 몇몇 예들에서, 입력 인터페이스 (28) 는 수신기 및/또는 모뎀을 구비한다. 입력 인터페이스 (28) 는 채널 (16) 을 통해 인코딩된 비디오 데이터를 수신할 수도 있다. 디스플레이 디바이스 (32) 는 목적지 디바이스 (14) 와 통합되거나 또는 그것 외부에 있을 수도 있다. 대체로, 디스플레이 디바이스 (32) 는 디코딩된 비디오 데이터를 디스플레이한다. 디스플레이 디바이스 (32) 는 액정 디스플레이 (LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (OLED) 디스플레이, 또는 다른 유형의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들을 포함할 수도 있다. 본 개시물에 따라, 비디오 인코더 (20) 와 비디오 디코더 (30) 는 본원에서 설명되는 하나 이상의 기법들을 비디오 코딩 프로세스의 부분 (예컨대, 비디오 인코딩 또는 비디오 디코딩) 으로서 수행할 수도 있다.
- [0041] 도 1은 단지 일 예이고 본 개시물의 기법들은 비디오 인코딩 디바이스 및 비디오 디코딩 디바이스 사이에 임의의 데이터 통신을 반드시 포함하지는 않는 비디오 코딩 설정들 (예컨대, 비디오 인코딩 또는 비디오 디코딩) 에 적용될 수도 있다. 다른 예들에서, 데이터는 로컬 메모리로부터 추출되며, 네트워크를 통해 스트리밍되는 등등이 된다. 비디오 인코딩 디바이스가 데이터를 인코딩하고 메모리에 저장할 수도 있으며, 그리고/또는 비디오 디코딩 디바이스가 메모리로부터 데이터를 추출하고 디코딩할 수도 있다. 많은 예들에서, 비디오 인코딩 및 디코딩은, 서로 통신하지 않지만 단순히 데이터를 메모리에 인코딩하고 및/또는 메모리로부터 데이터를 추출하고 디코딩하는 디바이스들에 의해 수행된다.
- [0042] 비디오 인코더 (20) 와 비디오 디코더 (30) 각각은 다양한 적합한 회로, 이를테면 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들 (DSP들), 주문형 집적회로들 (ASIC들), 필드 프로그램가능 게이트 어레이들 (FPGA들), 개별 로직, 하드웨어, 또는 그것들의 임의의 조합 중 임의의 것으로서 구현될 수도 있다. 그 기법들이 부분적으로 소프트웨어로 구현되면, 디바이스가 적합한 비일시적 컴퓨터 판독가능 저장 매체 내에 소프트웨어에 대한 명령을 저장할 수도 있고 하나 이상의 프로세서들을 사용하여 하드웨어에서 그 명령들을 실행하여 본 개시물의 기법들을 수행할 수도 있다. 전술한 바 (하드웨어, 소프트웨어, 하드웨어 및 소프트웨어의 조합 등을 포함) 중 임의의 것은 하나 이상의 프로세서들이라고 간주될 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 의 각각은 하나 이상의 인코더들 또는 디코더들 내에 구비될 수도 있고, 그것들 중 어느 하나는 결합형 인코더/디코더 (CODEC) 의 일부로서 개별 디바이스 내에 통합될 수도 있다.
- [0043] 본 개시물은 특정한 정보를 "시그널링하는" 비디오 인코더 (20) 에 일반적으로 관련이 있을 수도 있다. "시그널링"이란 용어는 일반적으로는 압축된 비디오 데이터를 디코딩하는데 사용되는 신택스 엘리먼트들 및/또는 다른 데이터의 통신을 말할 수도 있다. 이러한 통신은 실시간 또는 거의 실시간으로 일어날 수도 있다. 대안으로, 이러한 통신은 기간 (span of time) 에 걸쳐 일어날 수도 있어, 이를테면 인코딩 시에 신택스 엘리먼트들을 컴퓨터 판독가능 저장 매체에 인코딩된 비트스트림으로 저장하는 경우에 일어날 것이며, 그 신택스 엘리먼트들을 비디오 디코딩 디바이스가 그 다음에 이 매체에 저장된 후의 임의의 시간에 추출할 수도 있다. 몇몇 예들에서, 인코더 관점에서, 시그널링은 인코딩된 비트스트림을 생성하는 것을 포함할 수도 있고, 디코더 관점에서, 시그널링은 코딩된 비트스트림을 수신 및 파싱하는 것을 포함할 수도 있다.
- [0044] 몇몇 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는, ISO/IEC MPEG-4 비주얼 그리고 SVC (Scalable Video Coding) 및 MVC (Multiview Video Coding) 확장본들을 포함한 ITU-T H.264 (또한 ISO/IEC MPEG-4 AVC로 알려짐) 와 같은 비디오 압축 표준에 따라 동작한다. MVC의 최신 공동 초안이 『"Advanced video coding for generic audiovisual services," ITU-T Recommendation H.264, Mar 2010』 에 기재되어 있다. 다른 예들에서, 비디오 인코더 (20) 와 비디오 디코더 (30) 는 ITU-T H.261, ISO/IEC MPEG-1 비주얼, ITU-T H.262 또

는 ISO/IEC MPEG-2 비주얼, ITU-T H.263 등을 포함하는 다른 비디오 코딩 표준들에 따라 동작할 수도 있다. 본 개시물의 기법들은, 그러나, 임의의 특정 코딩 표준 또는 기법으로 제한되지 않는다.

[0045] 다른 예들에서, 비디오 인코더 (20) 와 비디오 인코더 (30) 는, ITU-T 비디오 코딩 전문가 그룹 (VCEG) 및 ISO/IEC 동 화상 전문가 그룹 (motion Picture Experts Group, MPEG) 의 JCT-VC (Joint Collaboration Team on Video Coding) 에 의해 개발된 고 효율 비디오 코딩 (HEVC) 표준을 포함하는, 다른 비디오 압축 표준들에 따라 동작할 수도 있다. "HEVC 규격 초안 9"라고 지칭되는 HEVC 표준의 다른 초안이, http://phenix.int-evry.fr/jct/doc_end_user/documents/11_Shanghai/wg11/JCTVC-K1003-v8.zip로부터 다운로드 가능한, 『Bross et al., "High Efficiency Video Coding (HEVC) text specification draft 9," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 11th Meeting, Shanghai, China, October 2012』 에 기재되어 있다. "HEVC 규격 초안 10" 또는 "WD10"이라고 지칭되는 HEVC 표준의 다른 최근 초안은, 문서 JCTVC-L1003v34 『Bross et al., "High efficiency video coding (HEVC) text specification draft 10 (for FDIS & Last Call)", Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 12th Meeting: Geneva, CH, 14-23 January, 2013』 에 기재되어 있으며, 이는 http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v34.zip으로부터 다운로드 가능하다. 본원에서 "WD 10 개정본"이라고 지칭되는 HEVC 표준의 또 다른 초안이, 『Bross et al., "Editors' proposed corrections to HEVC version 1", Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 13th Meeting, Incheon, KR, April 2013』 에 기재되어 있으며, 이는 http://phenix.int-evry.fr/jct/doc_end_user/documents/13_Incheon/wg11/JCTVC-M0432-v3.zip로부터 입수가능하다.

[0046] 현재, VCEG 및 MPEG의 JCT-3C (Joint Collaboration Team on 3D video Coding) 가 HEVC에 기초하여 3DV 표준을 개발하고 있는데, 표준화 노력의 일부는 HEVC에 기초한 멀티뷰 비디오 코덱 (MV-HEVC) 의 표준화를 포함하고 다른 일부는 HEVC에 기초한 3D 비디오 코딩 (3D-HEVC) 를 위한 것이다. 3D-HEVC의 경우, 코딩 단위 및/또는 예측 유닛 레벨에서의 코딩 도구들을 포함하는 새로운 코딩 도구들이, 텍스처 뷰 및 깊이 뷰 양쪽 모두에 대해, 포함되고 지원될 수도 있다. 3D-HEVC에 대한 소프트웨어 (즉, 3D-HTM) 가 다음의 링크로부터 다운로드될 수 있다:

[0047] [3D-HTM version 8.0]: https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-8.0/

[0048] 3D-HEVC의 규격 초안 (즉, 『Tech et al., "3D-HEVC Draft Text 1," Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 5th Meeting, Vienna, AT, 27 July - 2 August 2013, document number: JCT3V-E1001-v2』 (이하, "JCT3V-E1001" 또는 "3D-HEVC 초안 텍스트 1")) 이 다음으로부터 입수 가능하다:

[0049] http://phenix.it-sudparis.eu/jct2/doc_end_user/documents/5_Vienna/wg11/JCT3V-E1001-v3.zip. 3D-HEVC의 소프트웨어 설명 (『Zhang et al., "3D-HEVC Test Model 3," Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 3rd Meeting, Geneva, CH, 17-23 Jan. 2013, document number: JCT3V-C1005_d0』 (이하, "JCT3V-C1005" 또는 "3D-HEVC 테스트 모델 3")) 이 다음으로부터 입수 가능하다: http://phenix.int-evry.fr/jct3v/doc_end_user/documents/3_Geneva/wg11/JCT3V-C1005-v2.zip. 3D-HEVC의 다른 소프트웨어 설명 (『Zhang et al., "3D-HEVC Test Model 5," Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 5th Meeting, Vienna, AT, 27 July - 2 August 2013, document number: JCT3V-E1005』 (이하, "JCT3V-E1005") 이 다음으로부터 입수 가능하다: http://phenix.it-sudparis.eu/jct2/doc_end_user/current_document.php?id=1360.

[0050] 위에서 간략히 언급된 바와 같이, 비디오 인코더 (20) 는 비디오 데이터를 인코딩한다. 비디오 데이터는 하나 이상의 화상들을 포함할 수도 있다. 화상들의 각각은 비디오의 일부를 형성하는 스틸 이미지이다. 비디오 인코더 (20) 가 비디오 데이터를 인코딩하는 경우, 비디오 인코더 (20) 는 비트스트림을 생성할 수도 있다. 비트스트림은 비디오 데이터의 코딩된 표현을 형성하는 비트들의 시퀀스를 포함할 수도 있다. 비트스트림은 코딩된 화상들 및 연관된 데이터를 포함할 수도 있다. 코딩된 화상이 화상의 인코딩된 표현이다. 연관된 데이터는 시퀀스 파라미터 세트 (sequence parameter set, SPS) 들, 화상 파라미터 세트 (picture parameter set, PPS) 들, 비디오 파라미터 세트 (video parameter set, VPS) 들, 적응 파라미터 세트 (adaptive parameter set, APS) 들, 슬라이스 헤더들, 블록 헤더들, 및 다른 신택스 구조들을 포함할 수도

있다.

[0051] 화상이 S_L , S_{Cb} 및 S_{Cr} 로 표시되는 세 개의 샘플 어레이들을 포함할 수도 있다. S_L 은 루마 샘플들의 2차원 어레이 (즉, 블록) 이다. 루마 샘플들은 또한 본원에서 "Y" 샘플들이라고 지칭될 수도 있다. S_{Cb} 는 Cb 색차 샘플들의 2차원 어레이이다. S_{Cr} 은 Cr 색차 샘플들의 2차원 어레이이다. 색차 샘플들은 본원에서 "크로마" 샘플들이라고 또한 지칭될 수도 있다. Cb 색차 샘플들은 본원에서 "U 샘플들"이라고 지칭될 수도 있다. Cr 색차 샘플들은 본원에서 "V 샘플들"이라고 지칭될 수도 있다.

[0052] 몇몇 예들에서, 비디오 인코더 (20) 는 화상의 크로마 어레이들 (즉, S_{Cb} 및 S_{Cr}) 을 다운 샘플링할 수도 있다.

예를 들어, 비디오 인코더 (20) 는 YUV 4:2:0 비디오 포맷, YUV 4:2:2 비디오 포맷, 또는 4:4:4 비디오 포맷을 사용할 수도 있다. YUV 4:2:0 비디오 포맷에서, 비디오 인코더 (20) 는 크로마 어레이들이 루마 어레이의 1/2의 높이 및 1/2의 폭이 되도록 크로마 어레이들을 다운 샘플링할 수도 있다. YUV 4:2:2 비디오 포맷에서, 비디오 인코더 (20) 는 크로마 어레이들이 루마 어레이의 1/2의 폭 및 동일한 높이를 가지도록 크로마 어레이들을 다운 샘플링할 수도 있다. YUV 4:4:4 비디오 포맷에서, 비디오 인코더 (20) 는 크로마 어레이들을 다운 샘플링하지 않는다.

[0053] 화상의 인코딩된 표현을 생성하기 위해, 비디오 인코더 (20) 는 코딩 트리 단위 (coding tree unit, CTU) 들의 세트를 생성할 수도 있다. CTU들의 각각은, 루마 샘플들의 코딩 트리 블록, 크로마 샘플들의 2 개의 대응 코딩 트리 블록들, 및 코딩 트리 블록들의 샘플들을 코딩하는데 사용된 신택스 구조들일 수도 있다. 모노크롬 화상 또는 별개의 세 개의 컬러 평면들을 갖는 화상에서, CTU가 단일 코딩 트리 블록과 그 코딩 트리 블록의 샘플들을 코딩하는데 사용된 신택스 구조들을 포함할 수도 있다. 코딩 트리 블록 (CTB) 이 샘플들의 NxN 블록일 수도 있다. CTU가 "트리 블록" 또는 "최대 코딩 단위 (largest coding unit; LCU)"라고 또한 지칭될 수도 있다. HEVC의 CTU들은 다른 표준들, 이를테면 H.264/AVC의 매크로블록들과 대체로 유사할 수도 있다. 그러나, CTU가 특정 사이즈로 반드시 제한되는 것은 아니고 하나 이상의 코딩 단위들 (CU들) 을 포함할 수도 있다.

[0054] 화상을 인코딩하는 부분으로서, 비디오 인코더 (20) 는 화상의 각각의 슬라이스의 인코딩된 표현들 (즉, 코딩된 슬라이스들) 을 생성할 수도 있다. 코딩된 슬라이스를 생성하기 위해, 비디오 인코더 (20) 는 일련의 CTU들을 인코딩할 수도 있다. 본 개시물은 CTU의 인코딩된 표현을 코딩된 CTU라고 지칭할 수도 있다. 몇몇 예들에서, 슬라이스들의 각각은 정수 수의 코딩된 CTU들을 포함한다.

[0055] 코딩된 CTU를 생성하기 위해, 비디오 인코더 (20) 는 CTU의 코딩 트리 블록들에 대해 쿼드트리 구획화를 재귀적으로 수행하여 코딩 트리 블록들을 코딩 블록들로 나눌 수도 있으며, 그래서 그 이름이 "코딩 트리 단위들"이다. 코딩 블록이 샘플들의 NxN 블록이다. CU가, 루마 샘플 어레이, Cb 샘플 어레이 및 Cr 샘플 어레이를 갖는 화상의 루마 샘플들의 코딩 블록 및 크로마 샘플들의 대응하는 두 개의 코딩 블록들과, 그 코딩 블록들의 샘플들을 코딩하는데 사용된 신택스 구조들일 수도 있다. 모노크롬 화상 또는 별개의 세 개의 컬러 평면들을 갖는 화상에서, CU가 단일 코딩 블록과 그 코딩 블록의 샘플들을 코딩하는데 사용된 신택스 구조들을 포함할 수도 있다.

[0056] 비디오 인코더 (20) 는 CU의 코딩 블록을 하나 이상의 예측 블록들로 구획화할 수도 있다. 예측 블록이 동일한 예측이 적용되는 샘플들의 직사각형 (즉, 정사각형 또는 정사각형 아닌) 블록일 수도 있다. CU의 예측 유닛 (PU) 이 화상의 루마 샘플들의 예측 블록, 크로마 샘플들의 2 개의 대응하는 예측 블록들, 및 예측 블록 샘플들을 예측하는데 사용된 신택스 구조들일 수도 있다. 모노크롬 화상 또는 별개의 세 개의 컬러 평면들을 갖는 화상에서, PU가 단일 예측 블록과 그 예측 블록 샘플들을 예측하는데 사용된 신택스 구조들을 포함할 수도 있다. 비디오 인코더 (20) 는 PU의 각각의 예측 블록에 대한 예측 블록을 생성할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 CU의 각각의 PU의 루마, Cb 및 Cr 예측 블록들에 대한 예측 루마, Cb 및 Cr 블록들을 생성할 수도 있다. 예측 블록들은 예측 샘플 블록들이라고 또한 지칭될 수도 있다.

[0057] 비디오 인코더 (20) 는 PU에 대한 예측 블록들을 생성하기 위해 인트라 예측 또는 인터 예측을 사용할 수도 있다. 비디오 인코더 (20) 가 PU의 예측 블록들을 생성하기 위해 인트라 예측을 사용하면, 비디오 인코더 (20) 는 그 PU에 연관된 화상의 디코딩된 샘플들에 기초하여 그 PU의 예측 블록들을 생성할 수도 있다.

[0058] 비디오 인코더 (20) 가 PU의 예측 블록들을 생성하기 위해 인터 예측을 사용하면, 비디오 인코더 (20) 는 그 PU에 연관된 화상 이외의 하나 이상의 화상들의 디코딩된 샘플들에 기초하여 그 PU의 예측 블록들을 생성할 수도

있다. 비디오 인코더 (20) 는 PU의 예측 블록들을 생성하기 위해 단-예측 (uni-prediction) 또는 양방향-예측을 사용할 수도 있다. 비디오 인코더 (20) 가 단-예측을 사용하여 PU에 대한 예측 블록들을 생성하는 경우, 그 PU는 단일 모션 벡터를 가질 수도 있다. 비디오 인코더 (20) 가 단-예측을 사용하여 PU에 대한 예측 블록들을 생성하는 경우, 그 PU는 두 개의 모션 벡터들을 가질 수도 있다.

[0059] 비디오 인코더 (20) 가 CU의 하나 이상의 PU들에 대해 예측 블록들 (예컨대, 예측 루마, Cb 및 Cr 블록들) 을 생성한 후, 비디오 인코더 (20) 는 그 CU에 대해 하나 이상의 잔차 블록들을 생성할 수도 있다. CU에 대한 잔차 블록에서의 각각의 샘플은 CU의 PU의 예측 블록에서의 샘플 및 CU의 코딩 블록에서의 대응 샘플 간의 차이를 나타낼 수도 있다. 예를 들어, 비디오 인코더 (20) 는 CU에 대한 루마 잔차 블록을 생성할 수도 있다.

CU의 루마 잔차 블록에서의 각각의 샘플은 CU의 PU의 예측 루마 블록에서의 루마 샘플 및 CU의 원래의 루마 코딩 블록에서의 대응 샘플 간의 차이를 나타낼 수도 있다. 덧붙여서, 비디오 인코더 (20) 는 CU에 대한 Cb 잔차 블록을 생성할 수도 있다. CU의 Cb 잔차 블록에서의 각각의 샘플은 CU의 PU의 하나의 예측 Cb 블록에서의 Cb 샘플 및 그 CU의 원래의 Cb 코딩 블록에서의 대응 샘플 간의 차이를 나타낼 수도 있다. 비디오 인코더 (20) 는 CU에 대한 Cr 잔차 블록을 또한 생성할 수도 있다. CU의 Cr 잔차 블록에서의 각각의 샘플은 CU의 PU의 예측 Cr 블록에서의 Cr 샘플 및 그 CU의 원래의 Cr 코딩 블록에서의 대응 샘플 간의 차이를 나타낼 수도 있다.

[0060] 더욱이, 비디오 인코더 (20) 는 쿼드트리 구획화를 사용하여 CU의 잔차 블록 (예컨대, 루마, Cb 및 Cr 잔차 블록들) 을 하나 이상의 변환 블록들 (예컨대, 루마, Cb 및 Cr 변환 블록들) 로 분해할 수도 있다. 변환 블록이 동일한 변환이 적용되는 샘플들의 직사각형 블록일 수도 있다. CU의 변환 단위 (PU) 가 루마 샘플들의 변환 블록, 크로마 샘플들의 대응하는 2 개의 변환 블록들, 및 변환 블록 샘플들을 변환하는데 사용된 선택스 구조들일 수도 있다. 따라서, CU의 각각의 TU는 루마 변환 블록, Cb 변환 블록, 및 Cr 변환 블록에 연관될 수도 있다. 모노크롬 화상 또는 별개의 세 개의 컬러 평면들을 갖는 화상에서, TU가 단일 변환 블록과 그 변환 블록 샘플들을 변환하는데 사용된 선택스 구조들을 포함할 수도 있다. CU의 TU의 (즉, 그것에 연관된) 루마 변환 블록은 CU의 루마 잔차 블록의 서브블록일 수도 있다. CU의 TU의 Cb 변환 블록은 CU의 Cb 잔차 블록의 서브블록일 수도 있다. CU의 TU의 Cr 변환 블록은 CU의 Cr 잔차 블록의 서브블록일 수도 있다.

[0061] 3D 코딩의 경우, 깊이 블록들에서의 깊이 값들은 각각이 주어진 화소 로케이션에 연관된 깊이의 레벨을 나타내는 샘플 값들 (예컨대, 루마 값들) 로서 또한 표현될 수도 있다. 본 개시물의 기법들 중 하나 이상은, 깊이 블록을 코딩함에 있어서 후보들의 리스트가 선택된 후보의 모션 정보를 상속 또는 사용하기 위해 생성되는 특히스킵 모드 또는 병합 모드와 같은 모드들에서, 깊이 블록들의 코딩에 적용 가능하다.

[0062] 비디오 인코더 (20) 는 하나 이상의 변환들을 TU의 변환 블록에 적용하여 TU에 대한 계수 블록을 생성할 수도 있다. 계수 블록이 변환 계수들의 2차원 어레이일 수도 있다. 변환 계수가 스칼라 양일 수도 있다. 예를 들어, 비디오 인코더 (20) 는 하나 이상의 변환들을 TU의 루마 변환 블록에 적용하여 그 TU에 대한 루마 계수 블록을 생성할 수도 있다. 비디오 인코더 (20) 는 하나 이상의 변환들을 TU의 Cb 변환 블록에 적용하여 TU에 대한 Cb 계수 블록을 생성할 수도 있다. 비디오 인코더 (20) 는 하나 이상의 변환들을 TU의 Cr 변환 블록에 적용하여 TU에 대한 Cr 계수 블록을 생성할 수도 있다.

[0063] 계수 블록 (예컨대, 루마 계수 블록, Cb 계수 블록 또는 Cr 계수 블록) 을 생성한 후, 비디오 인코더 (20) 는 그 계수 블록을 양자화할 수도 있다. 양자화는 변환 계수들이 그 변환 계수들을 표현하는데 사용된 데이터의 양을 가능한 한 줄이도록 양자화되어서, 추가의 압축을 제공하는 프로세스를 일반적으로 지칭한다. 비디오 인코더 (20) 가 계수 블록을 양자화한 후, 비디오 인코더 (20) 는 양자화된 변환 계수들을 나타내는 선택스 엘리먼트들을 엔트로피 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 양자화된 변환 계수들을 나타내는 선택스 엘리먼트들에 대해 컨텍스트 적응 이진 산술 코딩 (Context-Adaptive Binary Arithmetic Coding; CABAC) 을 수행할 수도 있다. 비디오 인코더 (20) 는 엔트로피 인코딩된 선택스 엘리먼트들을 비트스트림으로 출력할 수도 있다. 비트스트림은 엔트로피 인코딩되지 않는 선택스 엘리먼트들을 또한 포함할 수도 있다.

[0064] 비디오 디코더 (30) 는 비디오 인코더 (20) 에 의해 생성된 비트스트림을 수신할 수도 있다. 덧붙여서, 비디오 디코더 (30) 는 비트스트림으로부터 선택스 엘리먼트들을 획득 (예컨대, 디코딩) 하기 위해 그 비트스트림을 파싱할 수도 있다. 비디오 디코더 (30) 는 비트스트림으로부터 디코딩된 (또는 다르게는 획득된) 선택스 엘리먼트들에 적어도 부분적으로 기초하여 비디오 데이터의 화상들을 복원할 수도 있다. 비디오 데이터를 복원하는 프로세스는 비디오 인코더 (20) 에 의해 수행된 프로세스에 일반적으로 역일 수도 있다. 예를 들

면, 비디오 디코더 (30) 는 현재 CU의 PU들에 대한 예측 블록들을 결정하기 위해 그 PU들의 모션 벡터들을 사용할 수도 있다. 덧붙여서, 비디오 디코더 (30) 는 현재 CU의 TU들에 연관된 변환 계수 블록들을 역 양자화할 수도 있다. 비디오 디코더 (30) 는 현재 CU의 TU들에 연관된 변환 블록들을 복원하기 위해 변환 계수 블록들에 대해 역 변환들을 수행할 수도 있다. 몇몇 예들에서, 비디오 디코더 (30) 는 현재 CU의 PU들에 대한 예측 블록들의 샘플들을 현재 CU의 TU들의 변환 블록들의 대응 샘플들에 가산함으로써 현재 CU의 코딩 블록들을 복원할 수도 있다. 화상의 각각의 CU에 대한 코딩 블록들을 복원함으로써, 비디오 디코더 (30) 는 그 화상을 복원할 수도 있다.

[0065] 몇몇 경우들에서, 비디오 인코더 (20) 는 병합 모드 또는 스킵 모드 또는, 아마도 고급 모션 벡터 예측 (advanced motion vector prediction, AMVP) 모드를 사용하여 PU의 모션 정보를 시그널링할 수도 있다. 다르게 말하면, HEVC 표준에서, PU에 대한 두 개의 인터 예측 모드들, 즉 각각 병합 (스킵이 병합의 특수한 경우로서 간주됨) 모드 및 AMVP 모드가 있다. 병합 모드 또는 AMVP 모드 중 어느 하나에서, 비디오 인코더가 다수의 모션 벡터 예측자 (predictor) 들에 대한 모션 벡터 후보 리스트를 유지한다. 설명의 편의를 위해, 본 개시물은 병합 모드에 대한 모션 벡터 후보 리스트를 "머지 후보 리스트" 또는 "병합 후보 리스트"로서 지칭할 수도 있다. 마찬가지로, 본 개시물은 AMVP 모드에 대한 모션 벡터 후보 리스트를 AMVP 후보 리스트로서 지칭할 수도 있다. PU의 모션 정보는 PU의 모션 벡터(들) 및 PU의 참조 인덱스(들)를 포함할 수도 있다.

[0066] 비디오 인코더 (20) 가 병합 모드를 사용하여 현재 PU의 모션 정보를 시그널링하는 경우, 비디오 인코더 (20) 는 병합 후보 리스트를 생성한다. 병합 후보 리스트는 후보들의 세트를 포함한다. 병합 후보 리스트에서의 후보들은 "머지 후보들" 또는 "병합 후보들"이라고 지칭될 수도 있다. 그 후보들은 현재 PU에 공간적으로 또는 시간적으로 이웃하는 PU들의 모션 정보를 나타낼 수도 있다. 현재 PU에 공간적으로 이웃하는 PU들은 현재 PU와 동일한 화상에서의 현재 PU의 예측 블록에 인접한 예측 블록들을 가질 수도 있다. 현재 PU에 시간적으로 인접하는 PU들은 현재 PU와는 상이한 화상에 있을 수도 있다. 비디오 인코더 (20) 는 그 다음에 후보 리스트로부터 후보를 선택할 수도 있고 선택된 후보에 의해 나타내어진 모션 정보를 현재 PU의 모션 정보로서 사용할 수도 있다. 더욱이, 병합 모드에서, 비디오 인코더 (20) 는 선택된 후보의 후보 리스트에서의 포지션을 시그널링할 수도 있다. 예를 들면, 비디오 인코더 (20) 는 선택된 병합 후보의 병합 후보 리스트에서의 포지션을 나타내는 인덱스 (즉, merge_idx) 를 시그널링할 수도 있다. 비디오 디코더 (30) 는 동일한 후보 리스트를 생성할 수도 있고 선택된 후보의 포지션 (병합 인덱스에 의해 나타내어진 포지션) 의 표시에 기초하여, 선택된 후보를 결정할 수도 있다. 비디오 디코더 (30) 는 그 다음에 선택된 후보의 모션 정보를 사용하여 현재 PU에 대한 하나 이상의 예측 블록들 (예컨대, 예측 샘플들) 을 생성할 수도 있다. 비디오 디코더 (30) 는 현재 PU에 대한 예측 블록들 (예컨대, 예측 샘플들) 및 잔차 신호에 기초하여 샘플들을 복원할 수도 있다. 이런 식으로, 비디오 인코더가 현재 PU의 모션 벡터(들)뿐만 아니라 병합 모드에서의 참조 인덱스들을, 모션 벡터 후보 리스트로부터 하나의 후보를 취함으로써 생성할 수도 있다.

[0067] 스킵 모드는 비디오 인코더 (20) 가 후보 리스트를 생성하고 후보들의 리스트로부터 후보를 선택한다는 점에서 병합 모드와 유사하다. 그러나, 비디오 인코더 (20) 가 스킵 모드를 사용하여 현재 PU (예컨대 깊이 블록) 의 모션 정보를 시그널링하는 경우, 비디오 인코더 (20) 는 임의의 잔차 신호의 생성을 피할 수도 있다. 스킵 모드가 병합 모드와 동일한 모션 벡터 도출 프로세스를 가지기 때문에, 이 문서에서 설명되는 기법들은 병합 모드 및 스킵 모드 양쪽 모두에 적용할 수도 있다. 본 개시물의 하나 이상의 양태들이 후보 리스트들을 이용하는 AMVP 모드 또는 다른 모드들을 위해 또한 사용될 수도 있다.

[0068] AMVP 모드는 비디오 인코더 (20) 가 후보 리스트를 생성하고 후보들의 리스트로부터 후보를 선택한다는 점에서 병합 모드와 유사하다. 그러나, 비디오 인코더 (20) 가 AMVP 모드를 사용하여 현재 PU (예컨대, 깊이 블록) 의 모션 정보를 시그널링하는 경우, 비디오 인코더 (20) 는 후보 리스트에서의 선택된 후보의 포지션을 시그널링하는 것에 더하여 현재 PU에 대한 모션 벡터 차이 (MVD) 와 참조 인덱스를 시그널링할 수도 있다. 현재 PU에 대한 MVD가 현재 PU의 모션 벡터와 선택된 모션 벡터 후보의 모션 벡터 간의 차이를 나타낼 수도 있다. 단-예측에서, 비디오 인코더 (20) 는 현재 PU에 대한 하나의 MVD 및 하나의 참조 인덱스들을 시그널링할 수도 있다. 양방향-예측에서, 비디오 인코더 (20) 는 현재 PU에 대한 두 개의 MVD들 및 두 개의 참조 인덱스들을 시그널링할 수도 있다. 몇몇 예들에서, 비디오 인코더 (20) 는 현재 PU에 대해 하나의 MVD 및 하나의 참조 인덱스들을 통상 시그널링할 수도 있지만, 깊이 블록 예측은 두 개의 MVD들 및 두 개의 참조 인덱스들이 시그널링되는 양방향-예측과 유사한 기법들을 또한 사용할 수 있다.

[0069] 더욱이, 현재 PU의 모션 정보가 AMVP 모드를 사용하여 시그널링되는 경우, 비디오 디코더 (30) 는 동일한 후보 리스트를 생성할 수도 있고, 선택된 후보의 포지션의 표시에 기초하여, 선택된 후보를 결정할 수도 있다.

비디오 디코더 (30) 는 MVD를 선택된 후보의 모션 벡터에 가산함으로써 현재 PU의 모션 벡터를 복원할 수도 있다. 비디오 디코더 (30) 는 그 다음에 현재 PU의 복원된 모션 벡터 또는 모션 벡터들을 사용하여 현재 PU에 대한 예측 블록들을 생성할 수도 있다.

[0070] 몇몇 예들에서, 모션 벡터 후보 리스트는 병합 모드에 대한 다섯 개까지의 후보들과 AMVP 모드에 대한 두 개의 후보들만을 포함한다. 다르게 말하면, 병합 후보 리스트가 다섯 개까지의 후보들을 포함할 수도 있는 반면 AMVP 후보 리스트는 두 개의 후보들만을 포함할 수도 있다. 병합 후보 (즉, 병합 모드에 대한 모션 벡터 후보 리스트에서의 후보) 가 참조 화상 리스트들 (리스트 0 및 리스트 1) 및 참조 인덱스들 양쪽 모두에 대응하는 모션 벡터들을 포함할 수도 있다. 병합 후보가 병합 인덱스에 의해 식별된다면, 현재 블록들의 예측을 위해 사용되는 참조 화상들뿐만 아니라 연관된 모션 벡터들이 결정된다. 그러나, AMVP 모드 하에서 리스트 0 또는 리스트 1 중 어느 하나로부터의 각각의 잠재적 예측 방향에 대해, 참조 인덱스가 모션 벡터 후보 리스트에 대한 모션 벡터 예측자 인덱스와 함께 명시적으로 시그널링되는데, AMVP 후보가 모션 벡터만을 포함해서이다. AMVP 모드에서, 예측된 모션 벡터들은 추가로 리파인될 수 있다.

[0071] 위에서 나타난 바와 같이, 비디오 코더가 공간적 및 시간적 이웃 블록들로부터 병합 모드에 대한 후보들을 도출할 수도 있다. 비디오 코더는 슬라이스에 대한 슬라이스 헤더에 포함되는 코딩된 신택스 엘리먼트 **five_minus_max_num_merge_cand**로부터 후보들의 최대 수를 도출할 수도 있다. 신택스 엘리먼트 **five_minus_max_num_merge_cand**는 5에서 뺀 슬라이스에서 지원되는 병합 후보들의 최대 수를 특정한다. 비디오 코더는 병합 후보들의 최대 수, MaxNumMergeCand를 다음과 같이 도출할 수도 있다:

[0072]
$$\text{MaxNumMergeCand} = 5 - \text{five_minus_max_num_merge_cand} \quad (7-39)$$

[0073] MaxNumMergeCand의 값은 1 내지 5의 범위에 있다.

[0074] 비디오 코더가 다음의 단계들로 병합 후보 리스트를 구축할 수도 있다. 첫째, 비디오 코더는 도 1에 도시된 다섯 개의 공간적 이웃 블록들로부터 네 개까지의 공간적 모션 벡터 후보들을 도출할 수도 있다. 도 2는 병합 리스트에 대한 잠재적 후보들인 공간적 이웃들을 보여주는 개념도이다. 화살표들은 어떤 공간적 후보(들)가 비교되는 것인지를 표시한다. 비디오 코더는 공간적 모션 벡터 후보들을, 도 2에 도시된 바와 같이, 좌측 (A1), 상측 (B1), 우상측 (B0), 좌하측 (A0), 및 좌상측 (B2) 의 순서로 도출할 수도 있다. 더욱이, 비디오 코더는 동일한 공간적 모션 벡터 후보들을 제거하기 위해 전지 (pruning) 프로세스를 적용할 수도 있다. 예를 들어, 비디오 코더는 B1과 A1를 비교하며, B0와 B1을 비교하며, A0와 A1을 비교하고 B2를 B1 및 A1 양쪽 모두와 비교할 수도 있다. 전지 프로세스 후에 이용 가능한 이미 네 개의 병합 후보들이 있다면, 비디오 코더는 B2를 병합 후보 리스트에 삽입하지 않는다.

[0075] 둘째, 비디오 코더는 시간적 병합 후보들을 결정할 수도 있다. 예를 들면, 비디오 코더는 (가능화되고 이용 가능하다면) 병치된 (co-located) 참조 화상으로부터의 시간적 모션 벡터 예측자 (temporal motion vector predictor, TMVP) 후보를 병합 후보 리스트 (즉, 모션 벡터 후보 리스트) 속 공간적 모션 벡터 후보들 뒤에 추가할 수도 있다.

[0076] 셋째, 병합 후보 리스트 (즉, 모션 벡터 후보 리스트) 가 완전하지 않다면, 비디오 코더는 병합 후보 리스트가 모든 후보들 (즉, MaxNumMergeCand에 의해 표시된 모든 후보들) 을 갖기까지 인공 모션 벡터 후보들을 병합 후보 리스트의 끝에 생성 및 삽입할 수도 있다. 다르게 말하면, 비디오 코더는 병합 후보 리스트에서의 병합 후보의 수가 MaxNumMergeCand 미만이라면 인공 모션 벡터 후보를 병합 후보 리스트 속에 삽입할 수도 있다. 두 가지 유형들의 인공 모션 벡터 후보들, 즉, 조합된 양방향-예측 병합 후보들 (이것들은 B-슬라이스들에 대해서만 도출됨) 및 제로 모션 벡터 병합 후보들이 있다. 병합 후보 리스트는 제 1 유형 (즉, 조합된 양방향-예측 병합 후보들) 이 충분한 인공 후보들을 제공하지 않는다면 하나 이상의 제로 모션 벡터 병합 후보들을 포함할 수도 있다.

[0077] 현재 슬라이스 (즉, 비디오 코더가 현재 코딩하고 있는 슬라이스) 가 B 슬라이스인 경우, 비디오 코더는 조합된 양방향-예측 병합 후보들에 대한 도출 프로세스를 호출할 수도 있다. 적어도 몇몇 예들에서, B 슬라이스가 인트라 예측, 단방향 인터 예측, 및 양방향 인터 예측이 허용되는 슬라이스이다. 도출 프로세스가 호출되는 경우, 비디오 코더는, 병합 후보 리스트에 이미 있고 필요한 모션 정보를 갖는 병합 후보들의 각각의 쌍에 대해, (이용 가능하면) 리스트 0에서의 화상을 가리키는, 그 쌍의 제 1 병합 후보 (10CandIdx와 동일한 병합 후보 인덱스를 가짐) 의 모션 벡터 (및, 어떤 경우들에서는, 참조 인덱스) 와 (이용 가능하면 그리고 참조 화상 또는 모션 벡터 중 어느 하나가 제 1 후보와는 상이한) 리스트 1에서의 화상을 가리키는, 그 쌍의 제 2 병합 후

보 (11CandIdx와 동일한 병합 후보 인덱스를 가짐) 의 모션 벡터 (및, 어떤 경우들에서는, 참조 인덱스) 의 조합에 의해 조합된 양방향-예측 모션 벡터 후보들 (combIdx에 의해 표시된 인덱스를 가짐) 을 도출할 수도 있다.

병합 후보 쌍은 동일한 두 개의 병합 후보들의 상이한 순서들이 상이한 쌍들로서 간주된다는 의미에서 순서화된 쌍일 수도 있다. combIdx에 대응하는 10CandIdx 및 11CandIdx의 정의들은 아래의 표 1에 도시되어 있다.

표 1: 10CandIdx 및 11CandIdx의 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

표 1에서, 10CandIdx에 대한 행은 RefPicList0 모션 정보 (예컨대, 모션 벡터들, 참조 인덱스들) 를 뽑아내는 병합 후보들의 인덱스들을 표시한다. 마찬가지로, 표 1에서, 11CandIdx에 대한 행은 RefPicList1 모션 정보를 뽑아내는 병합 후보들의 인덱스들을 표시한다. 따라서, 조합 0에 대한 열 (즉, combIdx = 0) 은 조합된 양방향-예측 모션 벡터 후보가 병합 후보 0의 RefPicList0 모션 정보를 특정하고 병합 후보 1의 RefPicList1 모션 정보를 특정함을 표시한다. 모든 병합 후보들이 조합에 대한 적용 가능 모션 정보를 반드시 가질 필요는 없거나 (예컨대, 병합 후보 1이 RefPicList1 모션 정보를 갖지 않을 수도 있거나) 또는 병합 후보 0에 연관된 RefPicList0 및 병합 후보 1에 연관된 RefPicList1의 모션 정보가 동일하지 않기 때문에, 비디오 코더는 이용 가능한 남아있는 조합들이 없거나 또는 비디오 코더가 충분한 수의 조합된 양방향-예측 모션 벡터 후보들을 생성하기까지 표 1의 조합들을 combIdx의 순서로 프로세싱할 수도 있다.

0... 11인 combIdx에 대해, 조합된 양방향-예측 모션 벡터 후보들의 생성 프로세스는 다음의 조건들 중 하나가 참인 경우 종료된다:

- numOrigMergeCand가 이 프로세스를 호출하기 전의 병합 리스트에서의 후보들의 수를 나타내는, combIdx는 (numOrigMergeCand * (numOrigMergeCand - 1)) 과 동일하다.
- 병합 리스트에서의 전체 후보들 (새로 생성된 조합된 양방향-예측 병합 후보들을 포함함) 의 수는 MaxNumMergeCand와 동일하다.

위에서 나타난 바와 같이, 비디오 인코더가 병합 후보 리스트에 하나 이상의 제로 모션 벡터 병합 후보들을 포함시킬 수도 있다. 각각의 개별 제로 모션 벡터 병합 후보에 대해, 각각의 제로 모션 벡터 병합 후보의 모션 벡터가 0으로 설정되고 각각의 제로 모션 벡터 병합 후보에 대한 참조 인덱스가 0부터 가용 참조 인덱스들의 수 빼기 1까지 설정된다. 병합 후보 리스트에서의 병합 후보들의 수가 여전히 MaxNumMergeCand 미만이면, 비디오 코더는 병합 후보 리스트에서의 병합 후보들의 총 수가 MaxNumMergeCand와 동일하기까지 하나 이상의 제로 모션 벡터 후보들 (예컨대, 제로 참조 인덱스들 및 모션 벡터들) 을 삽입할 수도 있다.

본 개시물의 다음의 서브-섹션들은 본 개시물에 관련된 AVC-기반 및 HEVC-기반 3D 비디오 코딩 기법들을 재검토한다. 멀티-뷰 코딩 (예컨대, 3D 비디오 코딩) 에서, 동일한 장면의 상이한 관점들에서부터의 다수의 뷰들이 있을 수도 있다. "액세스 단위"라는 용어는 동일한 시간 인스턴스에 대응하는 화상들의 세트를 지칭하는데 사용될 수도 있다. 다르게 말하면, 액세스 단위가 하나의 출력 시간 인스턴스에 대한 모두의 뷰들의 코딩된 화상들을 포함할 수도 있다. "뷰 성분"이 단일 액세스 단위에서의 뷰의 코딩된 표현일 수도 있다. 몇몇 예들에서, 뷰 성분이 텍스처 뷰 성분과 깊이 뷰 성분을 포함할 수도 있다. 본 개시물에서, "뷰"가 동일한 뷰 식별자에 연관된 뷰 성분들의 시퀀스를 지칭할 수도 있다. 따라서, 뷰가 코딩된 텍스처 및 깊이 표현들 양쪽 모두를 포함하는 경우, 뷰 성분이 텍스처 뷰 성분과 깊이 뷰 성분을 포함할 (그런 성분들로 이루어질) 수도 있다. 몇몇 예들에서, 텍스처 뷰 성분이 단일 액세스 단위에서의 뷰의 텍스처의 코딩된 표현이다. 더욱이, 몇몇 예들에서, 깊이 뷰 성분이 단일 액세스 단위에서의 뷰의 깊이의 코딩된 표현이다. 깊이 뷰 성분이 깊이 화상이라고 또한 지칭될 수도 있다.

각각의 텍스처 뷰 성분은 디스플레이될 실제 이미지 콘텐츠를 포함한다. 예를 들어, 텍스처 뷰 성분이 루마 (Y) 및 크로마 (Cb 및 Cr) 성분들을 포함할 수도 있다. 각각의 깊이 뷰 성분은 그것의 대응하는 텍스처 뷰 성분에서의 화상들의 상대 깊이들을 나타낼 수도 있다. 몇몇 예들에서, 깊이 뷰 성분들이 루마 값들만을 포함하는 그레이 스케일 이미지들이다. 다르게 말하면, 깊이 뷰 성분들은 임의의 이미지 콘텐츠를 전달하지

않는 대신, 대응하는 텍스처 뷰 성분들에서의 화소들의 상대 깊이들의 측정치들을 제공할 수도 있다.

[0087] 예를 들어, 깊이 뷰 성분에서의 순수 백색 화소가 그 화소의 대응 텍스처 뷰 성분에서의 대응 화소 또는 화소들이 관람자의 관점에서 더 가깝다는 것을 나타낼 수도 있다. 이 예에서, 깊이 뷰 성분에서의 순수 흑색 화소가 그 화소의 대응 텍스처 뷰 성분에서의 대응 화소 또는 화소들이 관람자의 관점에서 더 멀리 있다는 것을 나타낸다. 흑색 및 백색 사이에서의 그레이의 다양한 음영들이 상이한 깊이 레벨들을 나타낸다. 예를 들면, 깊이 뷰 성분에서의 짙은 회색 화소가 그 화소의 텍스처 뷰 성분에서의 대응 화소가 깊이 뷰 성분에서의 옅은 회색 화소보다 더 멀리 있다는 것을 나타낸다. 이 예에서, 그레이 스케일만이 화소들의 깊이를 식별하기 위해 필요하기 때문에, 깊이 뷰 성분은 크로마 성분들을 포함할 필요가 없는데, 깊이 뷰 성분들에 대한 크로마 성분들이 어떠한 목적으로도 역할을 하지 않을 수도 있어서이다. 본 개시물은 예시 목적으로 깊이를 식별하기 위해 루마 값들 (예컨대, 세기 값들) 만을 사용하는 깊이 뷰 성분들의 예들을 제공하고 제한하는 것으로 간주되지 않아야 한다. 다른 예들에서, 다른 기법들이 텍스처 뷰 성분들에서 화소들의 상대 깊이들을 나타내기 위해 이용될 수도 있다.

[0088] 멀티-뷰 코딩에서, 비트스트림이 복수의 계층들을 가질 수도 있다. 그 계층들의 각각은 상이한 뷰에 대응할 수도 있다. 멀티-뷰 코딩에서, 비디오 디코더 (예컨대, 비디오 디코더 (30)) 가 뷰에서의 화상들을 임의의 다른 뷰에서의 화상들에 대한 참조 없이 디코딩할 수 있다면, 그 뷰는 "기본 뷰"라고 지칭될 수도 있다. 뷰의 디코딩이 하나 이상의 다른 뷰들에서의 화상들의 디코딩에 의존한다면 그 뷰는 비-기본 뷰 (non-base view) 라고 지칭될 수도 있다. 비-기본 뷰들 중 하나의 비-기본 뷰에서의 화상을 코딩하는 경우, 그 화상이 비디오 코더가 현재 코딩하고 있는 화상과는 상이한 뷰에 있지만 동일한 시간 인스턴스 (즉, 액세스 단위) 내에 있다면, 비디오 코더 (이들테면 비디오 인코더 (20) 또는 비디오 디코더 (30)) 는 그 화상을 참조 화상 리스트에 추가할 수도 있다. 다른 인터 예측 참조 화상들처럼, 비디오 코더는 뷰 간 예측 참조 화상을 참조 화상 리스트의 임의의 포지션에 삽입할 수도 있다.

[0089] 3D-HEVC에서, 디스패리티 벡터 (disparity vector, DV) 가 두 개의 뷰들 간의 변위의 추정자 (estimator) 로서 사용될 수도 있다. 이웃 블록들이 비디오 코딩에서 거의 동일한 모션/디스패리티 정보를 공유하기 때문에, 현재 블록은 양호한 예측자로서 이웃 블록들에서의 모션 벡터 정보를 사용할 수 있다. 이 아이디어를 추종하여, 이웃 블록 기반 디스패리티 벡터 도출 (neighboring block based disparity vector derivation, NBDV) 프로세스는 상이한 뷰들에서의 디스패리티 벡터를 추정하기 위해 이웃 모션 벡터 정보를 사용한다. 3D-HEVC는 다음의 문서, 즉 『Zhang et al., "3D-CE5.h: Disparity vector generation results," Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 1st Meeting: Stockholm, SE, 16-20 July 2012, document JCT3V-A0097』 (이하, "JCT3V-A0097") 에서 제안된 이웃 블록 (기반) 디스패리티 벡터 (NBDV) 방법을 먼저 채택했다.

[0090] 여러 공간적 및 시간적 이웃 블록들이 NBDV 프로세스에서 정의된다. NBDV 프로세스를 수행하는 비디오 코더가 현재 블록 및 후보 블록 (즉, 공간적 및 시간적 이웃 블록) 간의 상관의 우선순위에 의해 결정된 미리 정의된 순서로 공간적 및 시간적 이웃 블록들의 각각을 체크할 수도 있다. 따라서, NBDV 프로세스에서, 비디오 코더는 이웃 블록들의 두 개의 세트들을 이용한다. 이웃 블록들의 하나의 세트는 공간적 이웃 블록들로부터이고 다른 세트는 시간적 이웃 블록들로부터이다. 비디오 코더가 이웃 블록을 체크하는 경우, 비디오 코더는 이웃 블록이 디스패리티 모션 벡터를 갖는지 (즉, 모션 벡터가 뷰 간 참조 화상을 가리키는지) 의 여부를 결정할 수도 있다. 일단 비디오 코더가 디스패리티 모션 벡터를 발견하면, 비디오 코더는 디스패리티 모션 벡터를 디스패리티 벡터로 변환할 수도 있다. 예를 들어, 디스패리티 모션 벡터를 디스패리티 벡터로 변환하기 위해, 비디오 코더는 디스패리티 모션 벡터와 동일한 디스패리티 벡터를 설정할 수도 있다. 한편, 연관된 참조 뷰 순서 인덱스는 또한 반환된다. 다르게 말하면, NBDV 프로세스를 수행하는 부분으로서, 비디오 코더는 참조 뷰 순서 인덱스를 또한 결정할 수도 있다.

[0091] 3D-HEVC의 몇몇 버전들에서, 비디오 코더는 디스패리티 벡터 도출을 위해 NBDV 프로세스에서 두 개의 공간적 이웃 블록들을 사용한다. 두 개의 공간적 이웃 블록들은 도 3에 도시된 바와 같은 A1, B1에 의해 표시된 바와 같이, 현재 CU의 좌측 및 상측이다. 도 3은 현재 코딩 단위에 관하여 공간적 및 시간적 이웃 블록들을 도시하는 개념도이다. NBDV 프로세스에서 사용되는 공간적 이웃 블록들은 HEVC의 병합 모드들에서 사용되는 것들과 동일하다는 것에 주의해야 한다. 그러므로, 적어도 몇몇 예들에서, 추가적인 메모리 액세스는 NBDV 프로세스에서 공간적 이웃 블록들을 프로세싱하는 경우 요구되지 않는다.

[0092] 몇몇 예들에서, NBDV 프로세스에서 시간적 이웃 블록들을 체크하기 위해, 비디오 코더는 후보 화상 리스트를 생

성하기 위한 구축 프로세스를 먼저 수행할 수도 있다. 현재 뷰 (즉, 현재 코딩되고 있는 화상을 포함하는 뷰) 로부터의 두 개까지의 참조 화상들이 후보 화상들로서 취급될 수도 있다. 병치된 참조 화상 (즉, 병치된 화상) 이 후보 화상 리스트에 먼저 삽입되며, 뒤이어 나머지 후보 화상들 (즉, RefPicList0 및 RefPicList1에서의 참조 화상들의 모두) 이 참조 인덱스의 오름 차순으로 삽입된다.

[0093] 현재 화상의 현재 슬라이스가 B 슬라이스 (즉, 양 방향으로 인터 예측된 PU들을 포함하는 것이 허용된 슬라이스) 이면, 비디오 인코더 (20) 는, 슬라이스 헤더에서, 병치된 화상이 RefPicList0로부터 유래하는지 또는 RefPicList1로부터 유래하는지를 나타내는 선택스 엘리먼트 (예컨대, collocated_from_l0_flag) 를 시그널링할 수도 있다. 다르게 말하면, TMVP들의 사용이 현재 슬라이스에 대해 가능하게 되고 현재 슬라이스가 B 슬라이스 (예컨대, 양 방향으로 인터 예측된 PU들을 포함하는 것이 허용된 슬라이스) 인 경우, 비디오 인코더 (20) 는 병치된 화상이 RefPicList0에 있는지 또는 RefPicList1에 있는지를 나타내는 선택스 엘리먼트 (예컨대, collocated_from_l0_flag) 를 슬라이스 헤더에서 시그널링할 수도 있다. 현재 슬라이스가 B 슬라이스가 아니면, 비디오 인코더 (20) 가 병치된 화상이 RefPicList0에 있는지 또는 RefPicList1에 있는지를 나타내는 선택스 엘리먼트를 시그널링하는 것이 불필요할 수도 있는데, 현재 슬라이스가 I 슬라이스이면, 인터 예측이 허용되지 않고, 현재 슬라이스가 P 슬라이스이면, 그 슬라이스에 대해 하나의 참조 화상 리스트만이 있기 때문이다. 비디오 디코더 (30) 가 병치된 화상을 포함하는 참조 화상 리스트를 식별한 후, 비디오 디코더 (30) 는 슬라이스 헤더에서 시그널링될 수도 있는 다른 선택스 엘리먼트 (예컨대, collocated_ref_idx) 를 사용하여, 식별된 참조 화상 리스트에서 화상 (즉, 병치된 화상) 을 식별할 수도 있다. 다시 말하면, 참조 화상 리스트가 식별된 후, 슬라이스 헤더에서 시그널링된 collocated_ref_idx는 참조 화상 리스트에서 화상을 식별하는데 사용될 수도 있다.

[0094] 참조 화상 리스트들 양쪽 모두에서 동일한 참조 인덱스를 갖는 두 개의 참조 화상들이 이용 가능한 경우, 병치된 화상의 동일한 참조 화상 리스트에서의 참조 화상은 다른 참조 화상에 선행한다. 후보 화상 리스트에서의 각각의 후보 화상에 대해, 비디오 코더는 중앙 포지션을 커버하는 병치된 지역의 블록을 시간적 이웃 블록으로서 결정할 수도 있다.

[0095] 블록이 뷰 간 모션 예측으로 코딩되는 경우, 비디오 코더는 상이한 뷰에서 대응 블록을 선택하기 위해 디스패리티 벡터를 도출할 필요가 있을 수도 있다. 암시적 디스패리티 벡터 (IDV 또는 일명 도출된 디스패리티 벡터) 가 뷰 간 모션 예측에서 도출되는 디스패리티 벡터라고 지칭될 수도 있다. 심지어 블록이 모션 예측으로 코딩되더라도, 도출된 디스패리티 벡터는 다음의 블록을 코딩할 목적으로 버려지지 않는다.

[0096] 3D-HTM의 적어도 몇몇 설계들에서, NBDV 프로세스는 시간적 이웃 블록들에서의 디스패리티 모션 벡터들, 공간적 이웃 블록들에서의 디스패리티 모션 벡터들, 및 그 다음에 IDV들을 순서대로 체크한다. 일단 비디오 코더가 디스패리티 모션 벡터 또는 IDV를 발견하는 경우, 비디오 코더는 NBDV 프로세스를 종료한다.

[0097] 몇몇 예들에서, 비디오 코더가 NBDV 프로세스로부터 디스패리티 벡터를 도출하는 경우, 비디오 코더는 참조 뷰의 깊이 지도로부터 깊이 데이터 (즉, 깊이 뷰 성분) 를 추출함으로써 디스패리티 벡터를 추가로 리파인한다. 리파인먼트 프로세스는 깊이-지향 NBDV (DoNBDV) 라고 명명되고 다음 두 개의 단계들을 포함할 수도 있다. 첫째, 이전에 코딩된 참조 깊이 뷰, 이를테면 기본 뷰에서의 도출된 디스패리티 벡터에 의해 대응하는 깊이 블록의 위치를 찾음; 대응하는 깊이 블록의 사이즈는 현재 PU의 사이즈와 동일하다. 둘째, (『Chang et al., "3D-CE2.h related: Simplified DV derivation for DoNBDV and BVSP," Joint Collaborative Team on 3D Video Coding Extensions of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 4th Meeting, Incheon, KR 20-26 Apr. 2013, document no. JCT3V-D0138』 (이하, "JCT3V-D0138") 의 채택으로 인한) 대응 깊이 블록의 네 개의 코너 화소들로부터 하나의 깊이 값을 선택하고 선택된 깊이 값을 라파인된 디스패리티 벡터의 수평 성분으로 변환. 디스패리티 벡터의 수직 성분은 변경되지 않는다. JCT3V-D0138는 http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=823에서 입수 가능하다.

[0098] 3D-HEVC에서, 병합 후보 리스트들에 대한 구축 프로세스는 HEVC에서 사용되는 병합 후보 리스트들에 대한 구축 프로세스와 상이하다. 예를 들면, NBDV 프로세스 또는 DoNBDV로부터 도출된 디스패리티 벡터에 기초하여, 비디오 코더는 새로운 모션 벡터 후보 (즉, 뷰 간 예측된 모션 벡터 후보 (IPMVC)) 를, 이용 가능하다면, AMVP 및 스킵/병합 모드들에 추가할 수도 있다. 다르게 말하면, 비디오 코더는 병합 후보 리스트 또는 AMVP 후보 리스트에 IPMVC를 포함시킬 수도 있다. IPMVC는 참조 뷰에서의 참조 블록의 모션 정보를 특정할 수도 있다. 예를 들면, IPMVC가 하나 이상의 시간적 모션 벡터들, 뿐만 아니라 예측 방향 표시자들 및 참조 인덱스들을 특정할 수도 있다.

- [0099] 병합/스킵 모드의 경우, 비디오 코더는 뷰 간 예측된 모션 벡터를 다음의 단계들에 의해 도출할 수도 있다. 첫째, 비디오 코더는 동일한 액세스 단위의 참조 뷰에서의 현재 PU/CU의 대응 블록을 디스패리티 벡터에 의해 위치 찾을 수도 있다. 둘째, 대응 블록이 인트라 코딩되지 않고 뷰 간 예측되지 않고 그것의 참조 화상이 현재 PU/CU의 동일한 참조 화상 리스트에서의 하나의 엔트리의 화상 순서 카운트 (picture order count, POC) 값과 동일한 화상 순서 카운트 값을 갖는다면, 비디오 인코더 (20) 는, POC에 기초하여 참조 인덱스를 변환한 후, 뷰 간 예측된 모션 벡터가 될, 대응 블록의 모션 정보 (예측 방향, 참조 화상들, 및 모션 벡터들) 를 도출할 수도 있다.
- [0100] 도 4는 뷰 간 예측된 모션 벡터 후보의 도출 프로세스의 일 예를 보여준다. 특히, 도 4는 병합/스킵 모드에 대한 뷰 간 예측된 모션 벡터 후보의 도출을 도시하는 개념도이다. 도 4의 예에서, 현재 PU (40) 가 시간 인스턴스 T1에 뷰 V1에서 발생한다. 현재 PU (40) 에 대한 참조 PU (42) 가 현재 PU (40) 와는 상이한 뷰 (즉, 뷰 V0) 에서 그리고 현재 PU (40) 와는 동일한 시간 인스턴스 (즉, 시간 인스턴스 T1) 에 발생한다. 도 4의 예에서, 참조 PU (42) 는 양방향으로 인터 예측된다. 그러므로, 참조 PU (42) 는 제 1 모션 벡터 (44) 와 제 2 모션 벡터 (46) 를 갖는다. 모션 벡터 (44) 는 참조 화상 (48) 에서의 포지션을 나타낸다. 참조 화상 (48) 은 뷰 V0에서 그리고 시간 인스턴스 T0에 발생한다. 모션 벡터 (46) 는 참조 화상 (50) 에서의 포지션을 나타낸다. 참조 화상 (50) 은 뷰 V0에서 그리고 시간 인스턴스 T3에 발생한다.
- [0101] 비디오 코더는, 참조 PU (42) 의 모션 정보에 기초하여, IPMVC를 현재 PU (40) 의 병합 후보 리스트 내의 포함을 위해 생성할 수도 있다. IPMVC는 제 1 모션 벡터 (52) 와 제 2 모션 벡터 (54) 를 가질 수도 있다. 모션 벡터 (52) 는 모션 벡터 (44) 와 일치하고 모션 벡터 (54) 는 모션 벡터 (46) 와 일치한다. 비디오 코더는 IPMVC의 제 1 참조 인덱스가 참조 화상 (48) 과 동일한 타임 인스턴스 (즉, 시간 인스턴스 T0) 에서 발생하는 참조 화상 (즉, 참조 화상 (56)) 의 현재 PU (40) 에 대한 RefPicList0에서의 포지션을 나타내도록 IPMVC를 생성한다. 도 4의 예에서, 참조 화상 (56) 은 현재 PU (40) 에 대한 RefPicList0에서의 제 1 포지션 (즉, Ref0) 에서 발생한다. 더욱이, 비디오 코더는 IPMVC의 제 2 참조 인덱스가 참조 화상 (50) 과 동일한 타임 인스턴스에서 발생하는 참조 화상 (즉, 참조 화상 (58)) 의 현재 PU (40) 에 대한 RefPicList1에서의 포지션을 나타내도록 IPMVC를 생성한다. 따라서, 도 4의 예에서, IPMVC의 RefPicList0 참조 인덱스는 0과 동일할 수도 있다. 도 4의 예에서, 참조 화상 (59) 이 현재 PU (40) 에 대한 RefPicList1에서의 제 1 포지션 (즉, Ref0) 에서 발생하고 참조 화상 (58) 은 현재 PU (40) 에 대한 RefPicList1에서의 제 2 포지션 (즉, Ref1) 에서 발생한다. 따라서, IPMVC의 RefPicList1 참조 인덱스는 1과 동일할 수도 있다.
- [0102] 따라서, 도 4의 예에서, 디스패리티 벡터가 현재 코딩된 뷰 (뷰 1 또는 V1) 에서의 현재 PU (40) 에 대해 상이한 뷰 (예컨대, 뷰 0 또는 V0) 에서의 대응 블록 (42) 을 발견함으로써 계산된다. 대응 블록 (42) 이 인트라 코딩되지 않고 뷰 간 예측되지 않고 그것의 참조 화상이 현재 PU (40) 의 참조 화상 리스트 (예컨대, 도 4에 도시된 바와 같이, Ref0, List 0; Ref0, List1; Ref1, List 1) 에서의 POC 값을 갖는다면, 대응 블록 (42) 에 대한 모션 정보는 뷰 간 예측된 모션 벡터로서 사용된다. 비디오 코더는 POC에 기초하여 참조 인덱스를 스케일링할 수도 있다.
- [0103] 더욱이, 블록 (예컨대, PU) 에 대해 병합 후보 리스트 (또는 몇몇 예들에서, AMVP 후보 리스트) 를 생성하는 경우, 비디오 코더는 블록의 디스패리티 벡터를 뷰 간 디스패리티 모션 벡터 후보 (inter-view disparity motion vector candidate, IDMVC) 로 변환할 수도 있다. IDMVC는 블록의 디스패리티 벡터를 특정할 수도 있다. 비디오 코더는 IDMVC를 병합 후보 리스트 (또는 몇몇 예들에서, AMVP 후보 리스트) 속의 IPMVC와는 상이한 포지션에 추가할 수도 있다. 대안으로, 몇몇 예들에서, IDMVC가 이용 가능한 경우, 비디오 코더는 IDMVC를 병합 후보 리스트 (또는 일부 예들에서, AMVP 후보 리스트) 속의 IPMVC와는 동일한 포지션에 추가할 수도 있다. 이 맥락에서, IPMVC 또는 IDMVC 중 어느 하나가 "뷰 간 후보"로 지칭될 수도 있다. 몇몇 예들에서는, 병합/스킵 모드에서, 비디오 코더는 항상, IPMVC를, 이용 가능하다면, 병합 후보 리스트에서 모든 공간적 및 시간적 병합 후보들 앞에 삽입한다. 몇몇 이러한 예들에서, 비디오 코더는 A₀로부터 도출된 공간적 병합 후보 앞에 IDMVC를 삽입할 수도 있다.
- [0104] 『Thirumalai et al., "Merge candidates derivation from vector shifting," Joint Collaborative Team on 3D Video Coding Extensions of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 5th Meeting, Vienna, AU, July 27 - Aug. 2, 2013, document no. JCT3V-E0126』 (이하, "JCT3V-E0126") 는 벡터 시프팅으로부터의 병합 후보 도출을 설명한다. JCT3V-E0126는 http://phenix.it-sudparis.eu/jct3v/doc_end_user/current_document.php?id=1140에서 입수 가능하다. JCT3V-E0126의 채택으

로 인해, "시프트된 후보" 또는 "시프트된 IvMVC"라 명명된 하나 이상의 후보가 시프트된 디스패리티 벡터로 도출될 수도 있다. 이러한 후보는 시프트된 디스패리티 벡터들을 이용하여 참조 뷰에서의 참조 블록으로부터 도출된 또는 디스패리티 모션 벡터 또는 IDMVC를 포함하는 제 1 가용 공간적 병합 후보로부터 도출된 IPMVC일 수 있다. 추가적인 후보를 생성하기 위한 세부 단계들 및 병합 후보 리스트에 대한 삽입이 다음과 같이 설명된다.

[0105] 첫째, 비디오 코더가 디스패리티 벡터 (DV) 를 $((PuWidth/2 * 4 + 4, (PuHeight/2 * 4 + 4))$ 만큼 시프트시킨다. 비디오 코더는 그 DV를 사용하여 참조 뷰로부터 시프트된 IvMC 후보를 도출한다. 여기서, 현재 PU의 사이즈는 $PuWidth \times PuHeight$ 이다. 시프트된 IvMVC가 이용 가능하면, 비디오 코더는 단계 2 (즉, 아래에서 설명되는 제 2 단계) 를 스킵할 수도 있고, 이 시프트된 IvMVC가 디스패리티 벡터 시프팅 없는 IvMC와 동일하지 않으면, 비디오 코더는 시프트된 IvMC를 시간적 병합 후보 바로 직전의 병합 후보 리스트에 삽입한다.

[0106] 둘째, 비디오 코더는 디스패리티 시프트된 모션 벡터 (Disparity Shifted Motion Vector, DSMV) 로서 표시된 후보를 도출할 수도 있다. 비디오 코더는 DSMV를 추가적인 후보인 것으로 설정할 수도 있다. DSMV가 이용 가능하면, 비디오 코더는 DSMV를 병합 후보 리스트 속의 시프트된 IvMC와는 동일한 포지션에 직접 삽입할 수도 있다. 비디오 코더는 DSMV를 다음과 같이 도출할 수도 있다. 첫째, 비디오 코더는 공간적 이웃 블록들로부터 RefPicList0에 대응하는 제 1 가용 디스패리티 모션 벡터 (DMV) 를 식별한다. 둘째, DMV가 이용 가능하다면, 비디오 코더는 List 0에서의 모션 벡터의 수평 성분을 4만큼 시프트된 DMV로 설정하고, 비디오 코더는, BVSP가 가능해지는지의 여부에 의존하여, 변경되지 않은 모션 벡터의 수직 성분을 유지하거나 또는 모션 벡터의 수직 성분을 0으로 설정한다. List 1에서의 참조 인덱스들 및 모션 벡터들은 직접 상속된다. 그렇지 않으면 (즉, DMV가 이용 가능하지 않다면), 비디오 코더는 List 0 및 List 1에서의 모션 벡터의 수평 성분을 4만큼 시프트된 DV로 설정하고, 비디오 코더는 List 0 및 List 1에서의 모션 벡터들의 수직 성분들 양쪽 모두를 0으로 설정한다.

[0107] 『Tian et al., "CE1.h: Backward View Synthesis Prediction using Neighbouring Blocks," Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JCT 1/SC 29/WG 11, 3rd Meeting, Geneva, CH, 17-23 Jan. 2013, document no. JCT3V-C0152』 (이하, "JCT3V-C0152") 는 이웃 블록들을 이용한 백워드 뷰 합성 예측을 설명한다. JCT3V-C0152는 http://phenix.it-sudparis.eu/jct2/doc_end_user/current_document.php?id=594에서 입수 가능하다. JCT3V-C0152에서 제안된 바와 같은 백워드 워핑 VSP 접근법은 3차 JCT-3V 회의에서 채택되었다. JCT3V-C0152에서 제안된 바와 같은 이 백워드 워핑 VSP의 기본 아이디어는 3D-AVC에서의 블록 기반 VSP와 동일하다. 이들 두 기법들의 양쪽 모두는 모션 벡터 차이들의 송신을 피하고 더욱 정확한 모션 벡터들을 사용하기 위해 백워드 워핑 및 블록 기반 VSP를 사용한다. 구현 세부사항들은 상이한 플랫폼들로 인해 상이하다. 다음의 단락들은 "BVSP"라는 용어를 3D-HEVC에서의 백워드 워핑 VSP 접근법을 나타내기 위해 사용한다.

[0108] 3D-HTM의 몇몇 설계들에서, BVSP 모드는 스킵 모드 또는 병합 모드 중 어느 하나에서 인터 코드 블록에 대해서만 지원된다. BVSP 모드는 AMVP 모드에서 코딩된 블록에 대해 허용되지 않는다. BVSP 모드의 사용을 나타내기 위해 플래그를 송신하는 대신, 하나의 추가적인 병합 후보 (즉, BVSP 병합 후보) 가 도입되고 각각의 후보는 하나의 BVSP 플래그와 연관된다. 위에서 나타낸 바와 같이, 비디오 인코더 (20) 는 병합 인덱스 (예컨대, merge_idx) 를 비트스트림에서 시그널링할 수도 있고 비디오 디코더 (30) 는 그 비트스트림으로부터 병합 인덱스를 획득할 수도 있다. 디코딩된 병합 인덱스가 BVSP 병합 후보에 대응하는 경우, 현재 PU는 BVSP 모드를 사용한다. 더욱이, 디코딩된 병합 인덱스가 BVSP 병합 후보에 대응하는 경우, 현재 PU 내의 각각의 서브블록에 대해, 비디오 코더는 깊이 참조 뷰에서의 깊이 값을 변환함으로써 서브블록에 대한 디스패리티 모션 벡터를 도출할 수도 있다.

[0109] BVSP 플래그들의 설정은 다음과 같이 정의될 수도 있다. 공간적 병합을 도출하기 위해 사용되는 공간적 이웃 블록 후보가 BVSP 모드로 코딩되는 경우, 연관된 모션 정보는 기존의 병합 모드에서처럼 현재 블록에 의해 상속된다. 덧붙여서, 이 공간적 병합 후보에는 1과 동일한 BVSP 플래그가 태깅된다. 새로 도입된 BVSP 병합 후보에 대해, BVSP 플래그는 1로 설정된다. 모든 다른 병합 후보들에 대해, 연관된 BVSP 플래그들은 0으로 설정된다.

[0110] 위에서 나타낸 바와 같이, 3D-HEVC에서, 비디오 코더가 새로운 후보 (즉, BVSP 병합 후보) 를 도출할 수도 있고 BVSP 병합 후보를 병합 후보 리스트에 삽입할 수도 있다. 비디오 코더는 BVSP 병합 후보에 대한 대응하는 참조 인덱스들 및 모션 벡터들을 다음의 방법에 의해 설정할 수도 있다. 첫째, 비디오 코더는 NBDV로부터의

도출된 디스패리티 벡터의 뷰 인덱스 (refVIdxLX에 의해 표시됨) 를 획득할 수도 있다. 둘째, 비디오 코더는 refVIdxLX와 동일한 뷰 순서 색인을 갖는 참조 화상에 연관되는 참조 화상 리스트 RefPicListX (RefPicList0 또는 RefPicList1 중 어느 하나) 를 획득할 수도 있다. 비디오 코더는 NBDV 프로세스로부터의 대응하는 참조 인덱스 및 디스패리티 벡터를 RefPicListX에서의 BVSP 병합 후보의 모션 정보로서 사용할 수도 있다.

[0111] 셋째, 현재 슬라이스가 B 슬라이스이면, 비디오 코더는 RefPicListX와는 상이한 참조 화상 리스트 (즉, Y가 1-X 인 RefPicListY) 에서의 refVIdxLX와는 동일하지 않은 뷰 순서 색인 (refVIdxLY에 의해 표시됨) 을 갖는 뷰 간 참조 화상의 가용성을 체크할 수도 있다. 이러한 상이한 뷰 간 참조 화상이 발견되면, 비디오 코더는 양방향-예측 VSP를 적용한다. 한편, 비디오 코더는 NBDV 프로세스로부터의 스케일링된 디스패리티 벡터 및 상이한 뷰 간 참조 화상의 대응하는 참조 인덱스를 RefPicListY에서의 BVSP 병합 후보의 모션 정보로서 사용한다. 비디오 코더는 뷰 순서 인덱스가 refVIdxLX와 동일한 뷰로부터의 깊이 블록을 현재 블록의 깊이 정보 (텍스처-우선 코딩 순서의 경우) 로서 사용할 수도 있고, 비디오 코더는 두 개의 상이한 뷰 간 참조 화상들 (각각 하나의 참조 화상 리스트부터임) 을 백워드 워핑 프로세스를 통해 액세스하고 추가로 가중하여 최종 백워드 VSP 예측자를 획득할 수도 있다. 그렇지 않으면, 비디오 코더는 예측을 위한 참조 화상 리스트로서 RefPicListX 를 이용하여 단-예측 VSP를 적용한다.

[0112] 3D-HTM에서, 텍스처 우선 코딩이 공통 테스트 조건들에서 적용된다. 그러므로, 대응하는 비-기본 깊이 뷰는 하나의 비-기본 텍스처 뷰를 디코딩하는 경우 이용 불가능하다. 그러므로, 깊이 정보는 BVSP를 수행하기 위해 추정되고 사용된다. 블록에 대한 깊이 정보를 추정하기 위하여, 비디오 코더는 이웃 블록들로부터 디스패리티 벡터를 먼저 도출한 다음, 도출된 디스패리티 벡터를 사용하여 참조 뷰로부터 깊이 블록을 획득할 수도 있다. 3D-HTM 8.0 테스트 모델에서, NBDV (Neighboring Block Disparity Vector) 로서 알려진 디스패리티 벡터 예측자를 도출하는 프로세스가 존재한다. (dv_x, dv_y) 가 NBDV 함수로부터 식별된 디스패리티 벡터를 나타내고 현재 블록 위치는 $(block_x, block_y)$ 라고 하자.

[0113] 단-예측 BVSP의 몇몇 예들에서, 비디오 코더가 참조 뷰의 깊이 이미지에서 좌상측 위치 $(block_x+dv_x, block_y+dv_y)$ 를 갖는 깊이 블록을 폐지한다. 현재 블록은 여러 서브-블록들로 먼저 분할되는데, 각각의 서브-블록은 동일한 사이즈의 $W*H$ 를 갖는다. 사이즈가 $W*H$ 와 동일한 각각의 서브블록에 대해, 비디오 코더는 폐지된 깊이 블록 내의 대응 깊이 서브블록을 사용하고, 깊이 서브블록의 네 개의 코너 화상들로부터의 최대 깊이 값을 디스패리티 모션 벡터로 변환한다. 비디오 코더는 그 다음에 각각의 서브블록에 대한 도출된 디스패리티 모션 벡터를 모션 보상을 위해 사용한다. 도 5는 참조 뷰로부터 깊이 블록의 위치가 찾아진 다음 BVSP (또한 "BVSP 예측"이라 지칭됨) 를 위해 사용되는 방법의 세 개의 단계들을 예시한다.

[0114] 특히, 도 5는 BVSP 예측을 수행하기 위한 참조 뷰로부터의 깊이 블록 도출을 예시하는 개념도이다. 양방향-예측 BVSP의 몇몇 예들에서, RefPicList0 및 RefPicList1의 상이한 뷰들로부터 다수의 뷰 간 참조 화상들이 있는 경우, 비디오 코더는 양방향-예측 VSP를 적용한다. 다시 말하면, 비디오 코더는, 위에서 설명된 바와 같이, 두 개의 VSP 예측자들을 각각의 참조 리스트로부터 생성할 수도 있다. 비디오 코더는 그 다음에 두 개의 VSP 예측자들을 평균하여 최종 VSP 예측자를 획득할 수도 있다.

[0115] 도 5의 예에서, 비디오 코더가 현재 텍스처 화상 (60) 을 코딩하고 있다. 현재 텍스처 화상 (60) 은 "의존성 텍스처 화상"으로 라벨 표시되는데 현재 텍스처 화상 (60) 이 합성된 참조 텍스처 화상 (62) 에 의존하기 때문이다. 다르게 말하면, 비디오 코더는 현재 텍스처 화상 (60) 을 디코딩하기 위하여 참조 텍스처 화상 (62) (또는 그것의 부분들) 을 합성하는 것이 필요할 수도 있다. 참조 텍스처 화상 (62) 과 현재 텍스처 화상 (60) 은 동일한 액세스 단위에 있지만 상이한 뷰들에 있다.

[0116] 참조 텍스처 화상 (62) (또는 그것의 부분들) 을 합성하기 위하여, 비디오 코더는 현재 텍스처 화상 (60) 의 블록들 (즉, 비디오 단위들) 을 프로세싱할 수도 있다. 도 5의 예에서, 비디오 코더는 현재 블록 (64) 을 프로세싱하고 있다. 비디오 코더가 현재 블록 (64) 을 프로세싱하는 경우, 비디오 코더는 현재 블록 (64) 에 대한 디스패리티 벡터를 도출하기 위해 NBDV 도출 프로세스를 수행할 수도 있다. 예를 들면, 도 5의 예에서, 비디오 코더는 현재 블록 (64) 에 이웃하는 블록 (68) 의 디스패리티 벡터 (66) 를 식별한다. 디스패리티 벡터 (66) 의 식별은 도 5의 단계 1로서 예시된다. 더욱이, 도 5의 예에서, 비디오 코더는, 디스패리티 벡터 (66) 에 기초하여, 현재 블록 (64) 의 디스패리티 벡터 (69) 를 결정한다. 예를 들면, 디스패리티 벡터 (69) 는 디스패리티 벡터 (66) 의 사본일 수도 있다. 디스패리티 벡터 (66) 를 복사하는 것이 도 5

의 단계 2로서 도시되어 있다.

[0117] 비디오 코더는, 현재 블록 (64) 의 디스패리티 벡터 (69) 에 기초하여, 참조 깊이 화상 (72) 에서의 참조 블록 (70) 을 식별할 수도 있다. 참조 깊이 화상 (72), 현재 텍스처 화상 (60), 및 참조 텍스처 화상 (62) 은 각각이 동일한 액세스 단위에 있을 수도 있다. 참조 깊이 화상 (72) 과 참조 텍스처 화상 (62) 은 동일한 뷰에 있을 수도 있다. 비디오 코더는, 현재 블록 (64) 의 텍스처 샘플 값들 및 참조 블록 (70) 의 깊이 샘플 값들에 기초하여, 참조 텍스처 화상 (62) 의 텍스처 샘플 값들을 결정할 수도 있다. 텍스처 샘플 값들을 결정하는 프로세스는 백워드 워핑이라고 지칭될 수도 있다. 3D-HEVC 테스트 모델 3의 섹션 H.8.5.2.2.7은 백워드 워핑의 프로세스를 설명한다. 백워드 워핑은 도 5의 단계 3으로서 도시되어 있다. 이런 식으로, 도 5는 참조 뷰로부터의 깊이 블록의 위치가 찾아진 다음 BVSP 예측을 위해 사용되는 방법의 세 개의 단계들을 예시한다.

[0118] BVSP에서 사용된 모션 보상 사이즈 (즉, 위에서 설명된 바와 같은 W*H) 는 8x4 또는 4x8 중 어느 하나일 수 있다. 모션 보상 사이즈를 결정하기 위해, 다음의 규칙이 적용된다. 각각의 8x8 블록에 대해, 비디오 코더는 대응하는 깊이 8x8 블록의 네 개의 코너들을 체크한다:

[0119] $if(vdepth[TL]<vdepth[BR]?0:1) != (vdepth[TR]<vdepth[BL]?0:1)$

[0120] $use\ 4x8\ partion\ (W=4,\ H=8)$

[0121] $else$

[0122] $use\ 8x4\ partion\ (W=8,\ H=4)$

[0123] 도 6은 하나의 8x8 깊이 블록의 네 개의 코너 화소들을 예시하는 개념도이다.

[0124] 3D-HEVC에 대한 병합 후보들의 최대 수 및 병합 리스트 구축 프로세스는 다음의 단락들에서 설명된다. 3D-HEVC의 몇몇 버전들에서, 병합 리스트에서의 후보들의 총 수는 여섯 개까지이고 five_minus_max_num_merge_cand는 5에서 뺀 병합 후보들의 최대 수를 특정하기 위해 슬라이스 헤더에서 시그널링된다. five_minus_max_num_merge_cand는 0 내지 5의 범위에 있다. five_minus_max_num_merge_cand는 5에서 뺀 슬라이스에서 지원되는 병합 모션 벡터 예측자 (MVP) 후보들 (즉, 병합 후보들) 의 최대 수를 특정한다. 비디오 코더가 병합 MVP 후보들의 최대 수 (즉, MaxNumMergeCand) 를 다음과 같이 컴퓨팅할 수도 있다:

[0125] $MaxNumMergeCand = 5 - five_minus_max_num_merge_cand + iv_mv_pred_flag[nuh_layer_id]$ (H-1)

[0126] 3D-HEVC의 이러한 버전들에서, five_minus_max_num_merge_cand의 값은 MaxNumMergeCand가 0 내지 (5 + iv_mv_pred_flag[nuh_layer_id]) 의 범위에 있도록 제한될 것이다.

[0127] 더욱이, 3D-HEVC의 이러한 버전들에서, iv_mv_pred_flag[layerId] 선택스 엘리먼트가, 뷰 간 모션 파라미터 예측이 layerId와 동일한 nuh_layer_id를 갖는 계층의 디코딩 프로세스에서 사용되는지의 여부를 나타낸다. 0과 동일한 iv_mv_pred_flag[layerId]는 뷰 간 모션 파라미터 예측이 layerId와 동일한 nuh_layer_id를 갖는 계층에 대해 사용되지 않는다는 것을 특정한다. 1과 동일한 iv_mv_pred_flag[layerId]는 뷰 간 모션 파라미터 예측이 layerId와 동일한 nuh_layer_id를 갖는 계층에 대해 사용될 수도 있다는 것을 특정한다. 존재하지 않는 경우, iv_mv_pred_flag[layerId]의 값은 0과 동일한 것으로 유추될 것이다.

[0128] 3D-HEVC에서의 병합 후보 리스트 구축 프로세스는 다음과 같이 정의될 수 있다:

[0129] 1. **IPMVC 삽입:** 뷰 간 모션 예측이 적용되는 경우, 비디오 코더는 위에서 설명된 절차에 의해 IPMVC를 도출한다. IPMVC가 이용 가능하다면, 비디오 코더는 IPMVC를 병합 리스트 (즉, 병합 후보 리스트) 에 삽입한다.

[0130] 2. **3D-HEVC에서의 공간적 병합 후보들에 대한 도출 프로세스 및 IDMVC 삽입**

[0131] 비디오 코더는 공간적 이웃 PU들의 모션 정보를 다음의 순서로 체크한다: A₁, B₁, B₀, A₀, 또는 B₂. 더욱이, 비디오 코더는 다음의 절차들에 의해 제약된 전지를 수행할 수도 있다.

[0132] - A₁ (즉, 공간적 이웃 PU A₁로부터 도출된 병합 후보) 과 IPMVC가 동일한 모션 벡터들 및 동일한 참조 인덱스들을 갖는다면, 비디오 코더는 A₁을 후보 리스트 (즉, 병합 후보 리스트) 에 삽입하지 않는다. 그렇지 않으면, 비디오 코더는 A₁을 리스트 (즉, 병합 후보 리스트) 에 삽입한다.

[0133] - B_1 및 A_1 /IPMVC가 동일한 모션 벡터들 및 동일한 참조 인덱스들을 갖는다면, 비디오 코더는 B_1 (즉, 공간적 이웃 PU B_1 로부터 도출된 병합 후보) 을 후보 리스트 (즉, 병합 후보 리스트) 에 삽입하지 않는다. 그렇지 않으면, 비디오 코더는 B_1 을 리스트 (즉, 병합 후보 리스트) 에 삽입한다.

[0134] - B_0 (즉, 공간적 이웃 PU B_0 로부터 도출된 병합 후보) 가 이용 가능하면, 비디오 코더는 B_0 를 후보 리스트 (즉, 병합 후보 리스트) 에 추가한다.

[0135] - 뷰 간 모션 예측이 적용되는 경우, 비디오 코더는 IDMVC를 위에서 설명된 절차에 의해 도출한다. IDMVC가 이용 가능하고 IDMVC가 A_1 및 B_1 로부터 도출된 후보들과는 상이하다면, 비디오 코더는 IDMVC를 후보 리스트 (즉, 병합 후보 리스트) 에 삽입한다.

[0136] - BVSP가 전체 화상에 대해 또는 현재 슬라이스에 대해 가능하게 되면, 비디오 코더는 BVSP 병합 후보를 병합 후보 리스트에 삽입한다.

[0137] - A_0 (즉, 공간적 이웃 PU A_0 로부터 도출된 병합 후보) 가 이용 가능하다면, 비디오 코더는 A_0 를 후보 리스트 (즉, 병합 후보 리스트) 에 추가한다.

[0138] - B_2 (즉, 공간적 이웃 PU B_2 로부터 도출된 병합 후보) 가 이용 가능하다면, 비디오 코더는 B_2 를 후보 리스트 (즉, 병합 후보 리스트) 에 추가한다.

[0139] - 뷰 간 모션 예측이 적용되는 경우, 비디오 코더는 시프트된 후보 (즉, DSMV) 를, 이용 가능하다면, 위에서 설명된 바와 같이 삽입한다.

[0140] **3. 시간적 병합 후보에 대한 도출 프로세스**

[0141] 시간적 병합 후보들에 대한 도출 프로세스는 병치된 PU의 모션 정보가 이용되는 HEVC에서의 시간적 병합 후보 도출 프로세스와 유사하다. 그러나, 시간적 병합 후보의 타겟 참조 인덱스가 0으로 고정되는 대신 변경될 수도 있다. 시간적 병합 후보의 타겟 참조 인덱스는 비디오 코더가 시간적 병합 후보에 기초하는 참조 화상의 참조 인덱스이다. 0과 동일한 타겟 참조 인덱스가 시간적 참조 화상 (즉, 현재 PU와 동일한 뷰에서의 참조 화상) 에 대응하는 한편 병치된 PU의 모션 벡터가 뷰 간 참조 화상을 가리키는 경우, 비디오 코더는 타겟 참조 인덱스를 참조 화상 리스트에서의 뷰 간 참조 화상의 제 1 엔트리에 대응하는 인덱스로 변경한다. 다르게 말하면, 비디오 코더는 타겟 참조 인덱스가 참조 화상 리스트에서의 제 1 뷰 간 참조 화상을 나타내도록 타겟 참조 인덱스를 변경한다. 그러나, 0과 동일한 타겟 참조 인덱스가 뷰 간 참조 화상에 대응하는 한편 병치된 PU의 모션 벡터가 시간적 참조 화상을 가리키는 경우, 비디오 코더는 타겟 참조 인덱스를 참조 화상 리스트에서의 시간적 참조 화상의 제 1 엔트리에 대응하는 다른 인덱스로 변경한다. 다르게 말하면, 비디오 코더는 타겟 참조 인덱스가 참조 화상 리스트에서의 제 1 뷰 간 참조 화상을 나타내도록 타겟 참조 인덱스를 변경한다.

[0142] **4. 3D-HEVC에서의 조합된 양방향-예측 병합 후보들에 대한 도출 프로세스**

[0143] 위의 세 개의 단계들로부터 도출된 후보들의 총 수가 후보들의 최대 수 미만이면, 비디오 코더는 HEVC에서 정의된 바와 동일한 프로세스를 다음 두 개의 변경들을 가지고서 수행한다:

[0144] o 첫째, 조합된 양방향-예측 병합 후보를 획득하는 조건들이 제 1/제 2 후보에 연관된 BVSP 플래그들의 체크를 추가함으로써 변경된다.

[0145] o 둘째, 10CandIdx 및 11CandIdx의 사양이 수정된다. combIdx, 10CandIdx 및 11CandIdx 사이의 관계는 3D-HEVC에서 10CandIdx 및 11CandIdx의 사양을 제공하는 표인 도 7에서 정의된다.

[0146] **5. 제로 모션 벡터 병합 후보들에 대한 도출 프로세스**

[0147] - 비디오 코더는 HEVC에서 정의된 (및 위에서 설명된) 바와 동일한 절차를 수행하여 제로 모션 벡터 병합 후보들을 도출한다.

[0148] 3D-HEVC에서의 조합된 양방향-예측 병합 후보들의 도출 프로세스의 설계는 하나 이상의 잠재적 문제들을 가질 수도 있다. 예를 들어, 3D-HEVC에서의 조합된 양방향-예측 병합 후보들의 도출 프로세스의 현재 설계는 조합된 양방향-예측 병합 후보를 구축하는데 사용되는 제 1 및 제 2 기존 병합 후보들의 BVSP 플래그들을 체크하

기 위해 추가될 부가적인 로직 유닛들을 요구할 수도 있다. 그러나, BVSP 플래그들의 추가적인 체크는 코딩 효율의 측면에서 도움이 되지 않는다. 따라서, BVSP 플래그들의 추가적인 체크는 복잡도를 증가시킨다.

[0149] 3D-HEVC에서의 조합된 양방향-예측 병합 후보들의 도출 프로세스에 연관된 잠재적 문제들의 다른 예에서, 조합된 양방향-예측 병합 후보들의 HEVC 도출 프로세스를 직접적으로 재사용하는 것은 예측 불가능한 디코딩 프로세스를 초래할 수도 있다. 조합된 양방향-예측 병합 후보들의 HEVC 도출 프로세스는 새로운 후보들을 생성하기 위해 네 개까지의 병합 후보들만을 취할 수 있다. 그러나, 이 프로세스가 3D-HEVC에서 직접적으로 사용된다면, 다섯 개의 병합 후보들이 이 프로세스에 대한 입력으로서 사용되는 경우가 있을 수 있다. 네 개까지의 병합 후보들이 있는 경우, 열두 개의 가능한 조합들만이 이용 가능하며, 따라서 그 조합들은 이 프로세스에서 표로 정의된다. 그러나, 다섯 개의 병합 후보들이 이용 가능한 경우, 이십 개의 가능한 조합들이 있을 수 있는 반면, 현재의 표 (즉, 위의 표 1) 는 그 많은 조합들을 지원하지 않는다.

[0150] 본 개시물의 기법들 중 하나 이상이 3D-HEVC에서의 조합된 양방향-예측 병합 후보들의 도출 프로세스에 관련한다. 본 개시물의 일 예의 기법에 따라, 3D-HEVC에서의 조합된 양방향-예측 병합 후보들의 도출 프로세스의 설계는 HEVC에서 사용되는 것에 의해 대체된다. 그러므로, 조합된 양방향-예측 병합 후보 도출 프로세스에서 BVSP 플래그들을 체크할 필요가 없다. 다르게 말하면, 병합 후보들의 리스트를 생성하는 프로세스는 임의의 BVSP 플래그들을 체크하는 일 없이 발생한다. 조합된 양방향-예측 병합 후보 도출 프로세스에서 BVSP 플래그들을 체크하지 않는 것은 코딩 효율에 대한 부정적인 영향 없이 인코딩/디코딩 프로세스의 복잡도를 줄일 수도 있다.

[0151] 이런 식으로, 본 개시물은 3D 비디오에 연관된 데이터를 코딩하는 방법을 제공할 수도 있다. 이 방법은 병합 리스트 도출 프로세스에 따라 3D 비디오에 연관된 비디오 블록을 코딩하기 위한 병합 후보들의 리스트를 생성하는 단계를 포함할 수도 있다. 그 리스트는 하나 이상의 양방향-예측 병합 후보들을 포함한다. 3D 비디오에 대한 병합 리스트 도출 프로세스는 비-3D 비디오에 연관되는 동일한 병합 리스트 도출 프로세스에 대응한다.

[0152] 더욱이, 본 개시물의 하나 이상의 기법들에 따라, HEVC에서의 조합된 양방향-예측 병합 후보들의 도출 프로세스를 호출하는 경우, 슬라이스 유형이 B 슬라이스와 동일하다는 것을 단지 체크하는 대신, 다른 조건이 또한 만족될 수 있으며, 다시 말하면, 병합 후보 리스트에 삽입되는 가용 병합 후보들의 수는 다섯 미만이 되어야 한다.

[0153] 따라서, 몇몇 예들에서, 비디오 코더가 3D 비디오에 연관된 데이터를 코딩할 수도 있다. 데이터를 코딩하는 부분으로서, 비디오 코더는 3D 비디오의 비디오 블록 (예컨대 PU) 을 코딩하기 위한 병합 후보들의 리스트를 생성할 수도 있다. 병합 후보들의 리스트를 생성하는 부분으로서, 비디오 코더는 리스트에서의 병합 후보들의 수가 5 미만인지의 여부를 결정할 수도 있다. 리스트에서의 병합 후보들의 수가 5 미만이라는 결정에 응답하여, 비디오 코더는 하나 이상의 조합된 양방향-예측 병합 후보들을 도출할 수도 있다. 이 예에서, 하나 이상의 조합된 양방향-예측 병합 후보들 중 각각의 개별 조합된 양방향-예측 병합 후보는 그 리스트에서의 병합 후보들의 개별 쌍 (예컨대, 순서화된 쌍) 에 이미 대응한다. 각각의 조합된 양방향-예측 병합 후보는 개별 쌍의 제 1 병합 후보의 모션 벡터 및 개별 쌍의 제 2 병합 후보의 모션 벡터의 조합이다. 제 1 병합 후보의 모션 벡터 및 제 2 병합 후보의 모션 벡터는 상이한 참조 화상 리스트들에서의 화상들을 가리킨다. 비디오 코더는 병합 후보들의 리스트에 하나 이상의 조합된 양방향-예측 병합 후보들을 포함시킬 수도 있다.

[0154] 대안으로, 몇몇 예들에서, 조합된 양방향-예측 병합 후보들의 도출 프로세스가 호출되기 전에, 병합 MVP 후보들의 최대 수, 즉 $MaxNumMergeCand$ 가 다음과 같이 리셋된다: $MaxNumMergeCand = 5 - five_minus_max_num_merge_cand$. 조합된 양방향-예측 병합 후보들의 도출 프로세스가 호출된 후, $MaxNumMergeCand$ 는 3D-HEVC에서와 같이 다음의 값으로 다시 설정된다: $MaxNumMergeCand = 5 - five_minus_max_num_merge_cand + iv_mv_pred_flag[nuh_layer_id]$. nuh_layer_id 는 계층 식별자를 특정하는 선택스 엘리먼트이다. 따라서, 이러한 몇몇 예들에서, 하나 이상의 조합된 양방향-예측 병합 후보들을 도출하기 전에, 비디오 코더가 병합 후보들의 최대 수를 5 빼기 제 1 선택스 엘리먼트의 값과 동일하게 리셋할 수도 있다. 제 1 선택스 엘리먼트는 5에서 뺀 슬라이스에서 지원되는 병합 후보들의 최대 수를 특정한다. 하나 이상의 조합된 양방향-예측 병합 후보들을 도출한 후, 비디오 코더는 병합 후보들의 최대 수를 5 빼기 제 1 선택스 엘리먼트의 값 더하기 제 2 선택스 엘리먼트의 값으로 설정할 수도 있는데, 제 2 선택스 엘리먼트는 뷰 간 모션 파라미터 예측이 계층의 디코딩 프로세스에서 사용되는지의 여부를 나타낸다.

[0155] $MaxNumMergeCand$ 가 6과 동일하고 HEVC에서의 조합된 양방향-예측 병합 후보들의 도출 프로세스가 호출되기 전에 다섯 개의 후보들이 있는 경우, HEVC 규격 초안 10의 하위절 8.5.3.2.4에서 특정된 바와 같이, 제로 후보 (참조

인덱스 및 모션 벡터 성분들 모두가 0임) 가 병합 후보 리스트 속으로 항상 생성 및 삽입된다.

[0156] 대안으로, 비디오 코더는 양방향-예측 병합 후보들을 결정하는 프로세스의 호출 전에 MaxNumMergeCand를 5로 설정하고 비디오 코더는 처음 네 개의 후보들만을 이 프로세스의 입력으로서 고려한다. 비디오 코더가 양방향-예측 병합 후보들을 결정하는 프로세스를 호출한 후, 비디오 코더는 새로 생성된 양방향-예측 병합 후보를, 이용 가능하다면 병합 후보 리스트의 말단에 놓는다. 따라서, 새로 생성된 양방향-예측 병합 후보는 병합 후보 리스트에서의 4번째 후보를 뒤따르며, 새로 생성된 양방향-예측 병합 후보를 비디오 코더는 양방향-예측 병합 후보들을 결정하기 위한 프로세스의 입력의 부분으로서 간주하지 않는다. 그 후, 이 예에서, MaxNumMergeCand는 6으로 다시 설정된다. 양방향-예측 병합 후보들을 결정하는 프로세스가 새로운 양방향-예측 병합 후보를 제공하지 않는 경우, HEVC 규격 초안 10의 하위절 8.5.3.2.4에서 특정된 바와 같이, 비디오 코더는 제로 후보를 생성하고 그 제로 후보를 병합 후보 리스트 속에 삽입한다. HEVC 규격 초안 10의 하위절 8.5.3.2.4가 아래에 재현된다.

[0157] **8.5.3.2.4 제로 모션 벡터 병합 후보들에 대한 도출 프로세스**

[0158] 이 프로세스에 대한 입력들은 다음이 된다:

- [0159] - 병합 후보 리스트 (mergeCandList),
- [0160] - mergeCandList에서의 모든 후보 N의 참조 인덱스들 (refIdxL0N 및 refIdxL1N),
- [0161] - mergeCandList에서의 모든 후보 N의 예측 리스트 이용 플래그들 (predFlagL0N 및 predFlagL1N),
- [0162] - mergeCandList에서의 모든 후보 N의 모션 벡터들 (mvL0N 및 mvL1N),
- [0163] - mergeCandList 내의 엘리먼트들의 수 (numCurrMergeCand).

[0164] 이 프로세스의 출력들은 다음이 된다:

- [0165] - 병합 후보 리스트 (mergeCandList),
- [0166] - mergeCandList 내의 엘리먼트들의 수 (numCurrMergeCand),
- [0167] - 이 프로세스의 호출 동안 mergeCandList에 추가된 모든 새로운 후보 (zeroCand_m)의 참조 인덱스들 (refIdxL0zeroCand_m 및 refIdxL1zeroCand_m),
- [0168] - 이 프로세스의 호출 동안 mergeCandList에 추가된 모든 새로운 후보 (zeroCand_m)의 예측 리스트 이용 플래그들 (predFlagL0zeroCand_m 및 predFlagL1zeroCand_m),
- [0169] - 이 프로세스의 호출 동안 mergeCandList에 추가된 모든 새로운 후보 (zeroCand_m)의 모션 벡터들 (mvL0zeroCand_m 및 mvL1zeroCand_m).

[0170] 변수 numRefIdx는 다음과 같이 도출된다:

- [0171] - slice_type이 P와 동일하면, numRefIdx는 num_ref_idx_l0_active_minus1 + 1과 동일하게 설정된다.
- [0172] - 그렇지 않으면 (slice_type이 B와 동일하면), numRefIdx는 Min (num_ref_idx_l0_active_minus1 + 1, num_ref_idx_l1_active_minus1 + 1) 과 동일하게 설정된다.

[0173] numCurrMergeCand가 MaxNumMergeCand 미만인 경우, 변수 numInputMergeCand는 numCurrMergeCand와 동일하게 설정되며, 변수 zeroIdx는 0과 동일하게 설정되고, 다음의 단계들은 numCurrMergeCand가 MaxNumMergeCand와 동일하기까지 반복된다:

[0174] 1. 제로 모션 벡터 병합 후보의 참조 인덱스들, 예측 리스트 이용 플래그들 및 모션 벡터들의 도출을 위해, 다음이 적용된다:

- [0175] - slice_type이 P와 동일하면, m이 (numCurrMergeCand - numInputMergeCand) 와 동일한 후보 zeroCand_m은 mergeCandList의 말단에 추가되며, 즉, mergeCandList[numCurrMergeCand]는 zeroCand_m과 동일하게 설정되고 zeroCand_m의 참조 인덱스들, 예측 리스트 이용 플래그들, 및 모션 벡터들은 다음과 같이 도출되고 numCurrMergeCand는 1만큼 증가된다:

[0176] $refIdxL0zeroCand_m = (zeroIdx < numRefIdx) ? zeroIdx : 0$ (8-122)

[0177] $refIdxL1zeroCand_m = -1$ (8-123)

[0178] $predFlagL0zeroCand_m = 1$ (8-124)

[0179] $predFlagL1zeroCand_m = 0$ (8-125)

[0180] $mvL0zeroCand_m[0] = 0$ (8-126)

[0181] $mvL0zeroCand_m[1] = 0$ (8-127)

[0182] $mvL1zeroCand_m[0] = 0$ (8-128)

[0183] $mvL1zeroCand_m[1] = 0$ (8-129)

[0184] $numCurrMergeCand = numCurrMergeCand + 1$ (8-130)

[0185] - 그렇지 않으면 (slice_type이 B와 동일하면), m이 (numCurrMergeCand - numInputMergeCand) 와 동일한 후보 zeroCand_m은 mergeCandList의 말단에 추가되며, 즉, mergeCandList[numCurrMergeCand]는 zeroCand_m과 동일하게 설정되고, zeroCand_m의 참조 인덱스들, 예측 리스트 이용 플래그들, 및 모션 벡터들은 다음과 같이 도출되고 numCurrMergeCand는 1만큼 증가된다:

[0186] $refIdxL0zeroCand_m = (zeroIdx < numRefIdx) ? zeroIdx : 0$ (8-131)

[0187] $refIdxL1zeroCand_m = (zeroIdx < numRefIdx) ? zeroIdx : 0$ (8-132)

[0188] $predFlagL0zeroCand_m = 1$ (8-133)

[0189] $predFlagL1zeroCand_m = 1$ (8-134)

[0190] $mvL0zeroCand_m[0] = 0$ (8-135)

[0191] $mvL0zeroCand_m[1] = 0$ (8-136)

[0192] $mvL1zeroCand_m[0] = 0$ (8-137)

[0193] $mvL1zeroCand_m[1] = 0$ (8-138)

[0194] $numCurrMergeCand = numCurrMergeCand + 1$ (8-139)

[0195] 2. 변수 zeroIdx는 1만큼 증가된다.

[0196] 따라서, 병합 후보들의 최대 수 (예컨대, MaxNumMergeCand) 가 6과 동일한 몇몇 예들에서, 비디오 코더가, 하나 이상의 양방향-예측 병합 후보들 중 임의의 것을 리스트에 추가하기 전에 병합 후보들의 리스트에 5 개의 병합 후보들이 있다는 결정에 응답하여, 제로 후보를 리스트에 포함시킬 수도 있다. 제로 후보의 모션 벡터 성분들은 0과 동일하고 제로 후보의 참조 인덱스는 0과 동일하다.

[0197] 본 개시물의 다음의 섹션은 HEVC의 맥락에서 본 개시물의 기법들과 일치하는 몇몇 예시적 구현 세부사항들을 설명한다. 3D-HEVC 초안 텍스트 1의 섹션들에 대한 변경들이 아래에서 도시된다. **굵게 및 밑줄** 또는 굵지 않은 **밑줄 및 이탤릭체**로 도시된 다양한 부분들은 HEVC 섹션들에 대한 추가들에 대응할 수도 있고, 이중 꺾쇠 괄호들에 의해 둘러싸인 이탤릭체로 도시된 부분들 (예컨대, *[[text]]*) 은 삭제들에 대응할 수도 있다. 본 개시물의 기법들은, 몇몇 예들에서, **굵게 및 밑줄**로 도시된 추가들과 이중 꺾쇠 괄호에 의해 둘러싸인 이탤릭체로 도시된 삭제들에 대응할 수도 있다.

[0198] **H.8.5.3.2.1 병합 모드를 위한 루마 모션 벡터들에 대한 도출 프로세스**

[0199] 이 프로세스는 merge_flag[xPb][yPb]가 1과 동일한 경우에만 호출되는데, (xPb, yPb) 는 현재 화상의 좌측상단 루마 샘플을 기준으로 현재 루마 예측 블록의 좌측상단 샘플을 특정한다.

- [0200] 이 프로세스에 대한 입력들은 다음이 된다:
- [0201] - 현재 화상의 좌측상단 루마 샘플을 기준으로 한 현재 루마 코딩 블록의 좌측상단 샘플의 루마 로케이션 (x_C, y_C),
- [0202] - 현재 화상의 좌측상단 루마 샘플을 기준으로 한 현재 루마 예측 블록의 좌측상단 샘플의 루마 로케이션 (x_{Pb}, y_{Pb}),
- [0203] - 현재 루마 코딩 블록의 크기를 특징하는 변수 (n_{CbS}),
- [0204] - 루마 예측 블록의 폭 및 높이를 특징하는 두 개의 변수들 (n_{PbW} 및 n_{PbH}),
- [0205] - 현재 코딩 단위 내의 현재 예측 유닛의 인덱스를 특징하는 변수 ($partIdx$).
- [0206] 이 프로세스의 출력들은 다음이 된다:
- [0207] - 루마 모션 벡터들 ($mvL0$ 및 $mvL1$),
- [0208] - 참조 인덱스들 ($refIdxL0$ 및 $refIdxL1$),
- [0209] - 예측 리스트 이용 플래그들 ($predFlagL0$ 및 $predFlagL1$),
- [0210] - 디스패리티 벡터 가용성 플래그들 ($ivpMvFlagL0$ 및 $ivpMvFlagL1$),
- [0211] - 현재 PU가 뷰 합성 예측을 사용하여 코딩되는지의 여부를 특징하는 플래그 ($vspModeFlag$).
- [0212] 로케이션 (x_{OrigP}, y_{OrigP}) 및 변수들 ($n_{OrigPbW}$ 및 $n_{OrigPbH}$) 은 (x_{Pb}, y_{Pb}), n_{PbW} , 및 n_{PbH} 의 값들을 저장하기 위해 다음과 같이 도출된다:
- [0213] (x_{OrigP}, y_{OrigP}) 는 (x_{Pb}, y_{Pb}) 과 동일하게 설정된다 (H-81)
- [0214] $n_{OrigPbW} = n_{PbW}$ (H-82)
- [0215] $n_{OrigPbH} = n_{PbH}$ (H-83)
- [0216] $\log_2 ParMrgLevel$ 이 2보다 더 크고 n_{CbS} 가 8과 동일한 경우, (x_{Pb}, y_{Pb}), n_{PbW} , n_{PbH} , 및 $partIdx$ 는 다음과 같이 수정된다:
- [0217] (x_{Pb}, y_{Pb}) = (x_{Cb}, y_{Cb}) (H-84)
- [0218] $n_{PbW} = n_{CbS}$ (H-85)
- [0219] $n_{PbH} = n_{CbS}$ (H-86)
- [0220] $partIdx = 0$ (H-87)
- [0221] 주 - $\log_2 ParMrgLevel$ 이 2보다 더 크고 n_{CbS} 가 8과 동일한 경우, 현재 코딩 단위의 모든 예측 유닛들은 $2N \times 2N$ 예측 유닛의 병합 후보 리스트와 동일한 단일 병합 후보 리스트를 공유한다.
- [0222] 모션 벡터들 ($mvL0$ 및 $mvL1$), 참조 인덱스들 ($refIdxL0$ 및 $refIdxL1$), 및 예측 이용 플래그들 ($predFlagL0$ 및 $predFlagL1$) 이 다음의 순서화된 단계들에 의해 도출된다:
- [0223] 1. 하위절 8.5.3.2.2에서의 이웃하는 예측 유닛 구획들로부터의 병합 후보들에 대한 도출 프로세스는 루마 코딩 블록 로케이션 (x_{Cb}, y_{Cb}), 코딩 블록 크기 (n_{CbS}), 루마 예측 블록 로케이션 (x_{Pb}, y_{Pb}), 루마 예측 블록 폭 (n_{PbW}), 루마 예측 블록 높이 (n_{PbH}), 및 구획화 인덱스 ($partIdx$) 를 입력들로 하여 호출되고, 출력은 가용성 플래그들 ($availableFlagA_0, availableFlagA_1, availableFlagB_0, availableFlagB_1, availableFlagB_2$), 참조 인덱스들 ($refIdxLXA_0, refIdxLXA_1, refIdxLXB_0, refIdxLXB_1, refIdxLXB_2$), 예측 리스트 이용 플래그들 ($predFlagLXA_0, predFlagLXA_1, predFlagLXB_0, predFlagLXB_1, predFlagLXB_2$), 및 모션 벡터들 ($mvLXA_0, mvLXA_1, mvLXB_0, mvLXB_1, mvLXB_2$) 이 되고, X는 0 또는 1이다.
- [0224] 2. X가 0 또는 1인 시간적 병합 후보에 대한 참조 인덱스들 ($refIdxLXC_0$) 이 0과 동일하게 설정된다.
- [0225] 3. 하위절 H.8.5.3.2.7에서의 시간적 루마 모션 벡터 예측에 대한 도출 프로세스는 루마 로케이션 (x_{Pb}, y_{Pb}), 루마 예측 블록 폭 (n_{PbW}), 루마 예측 블록 높이 (n_{PbH}), 및 변수 ($refIdxL0C_0$) 를 입력들로 하여 호출

되고, 출력은 가용성 플래그 (availableFlagL0Col) 및 시간적 모션 벡터 (mvL0Col) 이다. 변수들 (availableFlagCol, predFlagL0Col 및 predFlagL1Col) 은 다음과 같이 도출된다:

[0226] availableFlagCol = availableFlagL0Col (H-88)

[0227] predFlagL0Col = availableFlagL0Col (H-89)

[0228] predFlagL1Col = 0 (H-90)

[0229] 4. slice_type이 B와 동일한 경우, 하위절 H.8.5.3.2.7에서의 시간적 루마 모션 벡터 예측에 대한 도출 프로세스는 루마 로케이션 (xPb, yPb), 루마 예측 블록 폭 (nPbW), 루마 예측 블록 높이 (nPbH), 및 변수 (refIdxL1Col) 를 입력들로 하여 호출되고, 출력은 가용성 플래그 (availableFlagL1Col) 및 시간적 모션 벡터 (mvL1Col) 이다. 변수들 (availableFlagCol 및 predFlagL1Col) 은 다음과 같이 도출된다:

[0230] availableFlagCol = availableFlagL0Col || availableFlagL1Col (H-91)

[0231] predFlagL1Col = availableFlagL1Col (H-92)

[0232] 5. iv_mv_pred_flag[nuh_layer_id]에 의존하여, 다음이 적용된다.

[0233] - iv_mv_pred_flag[nuh_layer_id]가 0과 동일하면, 플래그들 (availableFlagIvMC, availableIvMCShift 및 availableFlagIvDC) 은 0과 동일하게 설정된다.

[0234] - 그렇지 않으면 (iv_mv_pred_flag[nuh_layer_id]가 1과 동일하면), 하위절 H.8.5.3.2.10에서 특정된 바와 같은 뷰 간 병합 후보들에 대한 도출 프로세스는 루마 로케이션 (xPb, yPb), 변수들 (nPbW 및 nPbH) 을 입력들로 하여 호출되고, 출력은 가용성 플래그들 (availableFlagIvMC, availableIvMCShift 및 availableFlagIvDC), 참조 인덱스들 (refIdxLXIvMC, refIdxLXIvMCShift 및 refIdxLXIvDC), 예측 리스트 이용 플래그들 (predFlagLXIvMC, predFlagLXIvMCShift 및 predFlagLXIvDC), 및 모션 벡터들 (mvLXIvMC, mvLXIvMCShift 및 mvLXIvDC) 에 배정된다 (X는 각각 0 또는 1임).

[0235] 6. view_synthesis_pred_flag[nuh_layer_id]에 의존하여, 다음이 적용된다.

[0236] - view_synthesis_pred_flag[nuh_layer_id]가 0과 동일하면, 플래그 (availableFlagVSP) 는 0과 동일하게 설정된다.

[0237] - 그렇지 않으면 (view_synthesis_pred_flag[nuh_layer_id]가 1과 동일하면), 하위절 H.8.5.3.2.13에서 특정된 바와 같은 뷰 합성 예측 병합 후보에 대한 도출 프로세스는 루마 로케이션들 (xCb, yCb) 를 입력으로 하여 호출되고, 출력들은 가용성 플래그 (availableFlagVSP), 참조 인덱스들 (refIdxLOVSP 및 refIdxLIVSP), 예측 리스트 이용 플래그들 (predFlagLOVSP 및 predFlagLIVSP), 및 모션 벡터들 (mvLOVSP 및 mvLIVSP) 이다.

[0238] 7. DepthFlag에 의존하여, 다음이 적용된다.

[0239] - DepthFlag가 0과 동일하면, 변수 availableFlagT는 0과 동일하게 설정된다.

[0240] - 그렇지 않으면 (DepthFlag가 1과 동일하면), 하위절 H.8.5.3.2.14에서 특정된 바와 같은 텍스처 병합 후보에 대한 도출 프로세스는 루마 로케이션 (xPb, yPb), 변수들 (nPbW 및 nPbH) 을 입력들로 하여 호출되고 출력들은 플래그 (availableFlagT), 예측 이용 플래그들 (predFlagLOT 및 predFlagL1T), 참조 인덱스들 (refIdxLOT 및 refIdxL1T), 및 모션 벡터들 (mvLOT 및 mvL1T) 이다.

[0241] 8. 병합 후보 리스트들 (mergeCandList 및 mergeCandIsVspFlag) 은 다음의 순서화된 단계들에 의해 특정된 바와 같이 구축된다:

[0242] a. 변수 numMergeCand는 0과 동일하게 설정된다.

[0243] b. availableFlagT가 1과 동일한 경우, 엔트리 mergeCandList[numMergeCand]는 T와 동일하게 설정되며, 엔트리 mergeCandIsVspFlag[numMergeCand]는 0과 동일하게 설정되고 변수 numMergeCand는 1만큼 증가된다.

[0244] c. availableFlagIvMC가 1과 동일한 경우, 엔트리 mergeCandList[numMergeCand]는 IvMC와 동일하게 설정되며, 엔트리 mergeCandIsVspFlag[numMergeCand]는 0과 동일하게 설정되고 변수 numMergeCand는 1만큼 증가된다.

[0245] d. availableFlagA₁이 1과 동일한 경우, 다음이 적용된다:

- [0246] - 다음의 조건이 참인 경우,
- [0247] - availableFlagT == 0 && availableFlagIvMC == 0,
- [0248] 또는 N은 T 및 IvMC에 의해 대체되는 다음의 조건들:
- [0249] - availableFlagN == 1 && predFlagLXN != predFlagLXA₁, (X가 0 및 1에 의해 대체
[0250] 됨),
- [0250] - availableFlagN == 1 && mvLXN != mvLXA₁ (X가 0 및 1에 의해 대체됨),
- [0251] - availableFlagN == 1 && refIdxLXN != refIdxLXA₁ (X가 0 및 1에 의해 대체됨),
- [0252] 중 하나 이상이 참이면,
- [0253] 엔트리 mergeCandList[numMergeCand]는 A₁과 동일하게 설정되며, 엔트리
mergeCandIsVspFlag[numMergeCand]는 VspModeFlag[xPb - 1][yPb + nPbH - 1]과 동일하게 설정되고 변수
numMergeCand는 1만큼 증가된다.
- [0254] e. availableFlagB₁이 1과 동일한 경우, 다음이 적용된다:
- [0255] - 다음의 조건이 참인 경우,
- [0256] - availableFlagT == 0 && availableFlagIvMC == 0,
- [0257] 또는 N은 T 및 IvMC에 의해 대체되는 다음의 조건들:
- [0258] - availableFlagN == 1 && predFlagLXN != predFlagLXB₁, (X가 0 및 1에 의해 대체됨),
- [0259] - availableFlagN == 1 && mvLXN != mvLXB₁ (X가 0 및 1에 의해 대체됨),
- [0260] - availableFlagN == 1 && refIdxLXN != refIdxLXB₁ (X가 0 및 1에 의해 대체됨)
- [0261] 중 하나 이상이 참이면,
- [0262] 엔트리 mergeCandList[numMergeCand]는 B₁과 동일하게 설정되며, 엔트리
mergeCandIsVspFlag[numMergeCand]는 VspModeFlag[xPb + nPbW - 1][yPb - 1]과 동일하게 설정되고 변수
numMergeCand는 1만큼 증가된다.
- [0263] f. availableFlagB₀가 1과 동일한 경우, 엔트리 mergeCandList[numMergeCand]는 B₀와 동일하게 설정되며,
엔트리 mergeCandIsVspFlag[numMergeCand]는 VspModeFlag[xPb + nPbW][yPb - 1]과 동일하게 설정되고 변수
numMergeCand는 1만큼 증가된다.
- [0264] g. availableFlagIvDC가 1과 동일하고, 다음의 조건들 중 하나 이상이 참이고,
- [0265] - availableFlagA₁ == 0,
- [0266] - predFlagLXA₁ != predFlagLXIvDC, (X가 0 및 1에 의해 대체됨),
- [0267] - mvLXA₁ != mvLXIvDC (X가 0 및 1에 의해 대체됨),
- [0268] - refIdxLXA₁ != refIdxLXIvDC (X가 0 및 1에 의해 대체됨),
- [0269] 다음의 조건들:
- [0270] - availableFlagB₁ == 0,
- [0271] - predFlagLXB₁ != predFlagLXIvDC, (X가 0 및 1에 의해 대체됨),
- [0272] - mvLXB₁ != mvLXIvDC (X가 0 및 1에 의해 대체됨),

- [0273] - refIdxLXB₁ != refIdxLXIvDC (X가 0 및 1에 의해 대체됨)
- [0274] 중 하나 이상이 참이면,
- [0275] 엔트리 mergeCandList[numMergeCand]는 IvDC와 동일하게 설정되며, 엔트리 mergeCandIsVspFlag[numMergeCand]는 0과 동일하게 설정되고 numMergeCand는 1만큼 증가된다.
- [0276] h. availableFlagVSP가 1과 동일한 경우, 엔트리 mergeCandList[numMergeCand]는 VSP와 동일하게 설정되며, 엔트리 mergeCandIsVspFlag[numMergeCand]는 1과 동일하게 설정되고 변수 numMergeCand는 1만큼 증가된다.
- [0277] i. availableFlagA₀가 1과 동일한 경우, 엔트리 mergeCandList[numMergeCand]는 A₀와 동일하게 설정되며, 엔트리 mergeCandIsVspFlag[numMergeCand]는 VspModeFlag[xPb - 1][yPb + nPbH]와 동일하게 설정되고 변수 numMergeCand는 1만큼 증가된다.
- [0278] j. availableFlagB₂가 1과 동일하고 numMergeCand가 4 + iv_mv_pred_flag[nuh_layer_id] + DepthFlag 미만인 경우, 엔트리 mergeCandList[numMergeCand]는 B₂와 동일하게 설정되며, 엔트리 mergeCandIsVspFlag[numMergeCand]는 VspModeFlag[xPb - 1][yPb - 1]과 동일하게 설정되고 변수 numMergeCand는 1만큼 증가된다.
- [0279] k. availableFlagIvMCShift가 1과 동일하고 numMergeCand가 6 미만이고, 다음의 조건들:
 - [0280] - availableFlagIvMC == 0,
 - [0281] - predFlagLXMC != predFlagLXMCShift (X가 0 및 1에 의해 대체됨),
 - [0282] - mvLXMC != mvLXIvMCShift (X가 0 및 1에 의해 대체됨),
 - [0283] - refIdxLXMC != refIdxLXMCShift (X가 0 및 1에 의해 대체됨)
- [0284] 중 하나 이상이 참이면,
- [0285] 엔트리 mergeCandList[numMergeCand]는 IvMCShift와 동일하게 설정되며, 엔트리 mergeCandIsVspFlag[numMergeCand]는 0과 동일하게 설정되고 변수 numMergeCand는 1만큼 증가된다.
- [0286] l. 변수 availableFlagIvDCShift는 0으로 설정되고 다음의 조건들:
 - [0287] - DepthFlag는 0과 동일하며, [Ed (CY): 이 조건은 제거될 수 있는 것으로 이미 가정된다.]
 - [0288] - availableFlagIvMCShift는 0과 동일,
 - [0289] - numMergeCand는 6 미만
- [0290] 의 모두가 참인 경우,
- [0291] 하위절 H.8.5.3.2.15에서 특정된 바와 같은 시프트된 디스패리티 병합 후보에 대한 도출 프로세스는 mergeCandList에 있는 모든 후보 N의 가용성 플래그들 (availableFlagN), 참조 인덱스들 (refIdxLON 및 refIdxLIN), 예측 리스트 이용 플래그들 (predFlagLON 및 predFlagLIN), 모션 벡터들 (mvLON 및 mvLIN), mergeCandList, mergeCandIsVspFlag, 및 numMergeCand를 입력들로 하여 호출되고 출력들은 플래그들 (availableFlagIvDCShift), 예측 이용 플래그들 (predFlagL0IvDCShift 및 predFlagL1IvDCShift), 참조 인덱스들 (refIdxL0IvDCShift 및 refIdxL1IvDCShift), 및 모션 벡터들 (mvL0IvDCShift 및 mvL1IvDCShift) 이다. availableFlagIvDCShift가 1과 동일한 경우, 엔트리 mergeCandList[numMergeCand]은 IvDCShift와 동일하게 설정되며, 엔트리 mergeCandIsVspFlag[numMergeCand]는 0과 동일하게 설정되고 변수 numMergeCand는 1만큼 증가된다.
- [0292] m. availableFlagCol가 1과 동일하고 numMergeCand가 5 + iv_mv_pred_flag[nuh_layer_id] + DepthFlag 미만인 경우, 엔트리 mergeCandList[numMergeCand]는 Col과 동일하게 설정되며, 엔트리 mergeCandIsVspFlag[numMergeCand]는 0과 동일하게 설정되고 변수 numMergeCand는 1만큼 증가된다.
- [0293] 9. 변수 numOrigMergeCand는 numMergeCand와 동일하게 설정된다.
- [0294] 10. slice_type이 B와 동일하고 **numMergeCand가 5 미만**인 경우, 하위절 **8.5.3.2.3** *[[H.8.5.3.2.3]]* 에서 특정되는 조합된 양방향-예측 병합 후보들에 대한 도출 프로세스는 mergeCandList, *[[mergeCandIsVspFlag]]*

mergeCandList에서의 모든 후보 N 의 참조 인덱스들 (refIdxLON 및 refIdxLIN), 예측 리스트 이용 플래그들 (predFlagLON 및 predFlagLIN), 모션 벡터들 (mvLON 및 mvLIN), numCurrMergeCand, 및 numOrigMergeCand를 입력들로 하여 호출되고, 출력은 mergeCandList에 추가된 모든 새로운 후보 ($combCand_k$)의 mergeCandList, numCurrMergeCand, 참조 인덱스들 (refIdxL0combCand $_k$ 및 refIdxL1combCand $_k$), 예측 리스트 이용 플래그들 (predFlagL0combCand $_k$ 및 predFlagL1combCand $_k$), 및 모션 벡터들 (mvL0combCand $_k$ 및 mvL1combCand $_k$)에 배정된다. 추가된 후보들의 수 (numCombMergeCand)는 (numCurrMergeCand - numOrigMergeCand)와 동일하게 설정된다. numCombMergeCand가 0보다 더 큰 경우, k 는 0 내지 numCombMergeCand - 1의 범위에 있고, mergeCandIsVspFlag[numOrigMergeCand + k]는 0과 동일하게 설정된다다.

[0295] 11. 하위절 8.5.3.2.4에서 특정된 제로 모션 벡터 병합 후보들에 대한 도출 프로세스는 mergeCandList, mergeCandList에서의 모든 후보 N 의 참조 인덱스들 (refIdxLON 및 refIdxLIN), 예측 리스트 이용 플래그들 (predFlagLON 및 predFlagLIN), 모션 벡터들 (mvLON 및 mvLIN), 및 numCurrMergeCand를 입력들로 하여 호출되고, 출력은 mergeCandList, numCurrMergeCand, mergeCandList에 추가된 모든 새로운 후보 ($zeroCand_m$)의 참조 인덱스들 (refIdxL0zeroCand $_m$ 및 refIdxL1zeroCand $_m$), 예측 리스트 이용 플래그들 (predFlagL0zeroCand $_m$ 및 predFlagL1zeroCand $_m$), 및 모션 벡터들 (mvL0zeroCand $_m$ 및 mvL1zeroCand $_m$)에 배정된다. 추가된 후보들의 수 (numZeroMergeCand)는 (numCurrMergeCand - numOrigMergeCand - numCombMergeCand)와 동일하게 설정된다. numZeroMergeCand가 0보다 더 큰 경우, m 는 0 내지 numZeroMergeCand - 1의 범위에 있고, mergeCandIsVspFlag[numOrigMergeCand + numCombMergeCand + m]은 0과 동일하게 설정된다.

[0296] ...

[0297] **[[H.8.5.3.2.3 조합된 양방향-예측 병합 후보들에 대한 도출 프로세스**

[0298] 이 프로세스에 대한 입력들은 다음이 된다:

- [0299] - 병합 후보 리스트 (mergeCandList),
- [0300] - 리스트 (mergeCandIsVspFlag).
- [0301] - mergeCandList에서의 모든 후보 N 의 참조 인덱스들 (refIdxLON 및 refIdxLIN),
- [0302] - mergeCandList에서의 모든 후보 N 의 예측 리스트 이용 플래그들 (predFlagLON 및 predFlagLIN),
- [0303] - mergeCandList에서의 모든 후보 N 의 모션 벡터들 (mvLON 및 mvLIN),
- [0304] - mergeCandList 내의 엘리먼트들의 수 (numCurrMergeCand),
- [0305] - 공간적 및 시간적 병합 후보 도출 프로세스 후의 mergeCandList 내의 엘리먼트들의 수 (numOrigMergeCand).

[0306] 이 프로세스의 출력들은 다음이 된다:

- [0307] - 병합 후보 리스트 (mergeCandList),
- [0308] - mergeCandList 내의 엘리먼트들의 수 (numCurrMergeCand),
- [0309] - 이 프로세스의 호출 동안 mergeCandList에 추가된 모든 새로운 후보 ($combCand_k$)의 참조 인덱스들 (refIdxL0combCand $_k$ 및 refIdxL1combCand $_k$),
- [0310] - 이 프로세스의 호출 동안 mergeCandList에 추가된 모든 새로운 후보 ($combCand_k$)의 예측 리스트 이용 플래그들 (predFlagL0combCand $_k$ 및 predFlagL1combCand $_k$),
- [0311] - 이 프로세스의 호출 동안 mergeCandList에 추가된 모든 새로운 후보 ($combCand_k$)의 모션 벡터들 (mvL0combCand $_k$ 및 mvL1combCand $_k$).

[0312] numOrigMergeCand가 1보다 더 크고 MaxNumMergeCand 미만인 경우, 변수 numInputMergeCand는 numCurrMergeCand와 동일하게 설정되며, 변수 combIdx는 0과 동일하게 설정되며, 변수 combStop은 FALSE와 동일하게 설정되고, 다음의 단계들은 combStop이 TRUE와 동일하기까지 반복된다:

- [0313] 1. 변수들 (l0CandIdx 및 l1CandIdx)은 표 8-6에서 특정된 바와 같은 combIdx를 사용하여 도출된다.

[0314] 2. 병합 후보 리스트 (*mergeCandList*) 에서 10Cand가 포지션 10CandIdx에서의 후보가 되고 11Cand가 포지션 11CandIdx에서의 후보가 되는 다음의 배정들이 이루어진다:

[0315] - $10Cand = mergeCandList[10CandIdx]$

[0316] - $11Cand = mergeCandList[11CandIdx]$

[0317] 3. 다음의 조건들의 모두가 참인 경우:

[0318] - $mergeCandIsVspFlag[10CandIdx] == 0.$

[0319] - $mergeCandIsVspFlag[11CandIdx] == 0.$

[0320] - $predFlagL010Cand = 1$

[0321] - $predFlagL111Cand = 1$

[0322] - $(DiffPicOrderCnt(RefPicList0[refIdxL010Cand], RefPicList1[refIdxL111Cand]) != 0) \parallel$

[0323] - $(mvL010Cand != mvL111Cand)$

[0324] k 가 ($numCurrMergeCand - numInputMergeCand$) 와 동일한 후보 $combCand_k$ 가 *mergeCandList*의 말단에 추가되며, 즉, $mergeCandList[numCurrMergeCand]$ 는 $combCand_k$ 와 동일하게 설정되고, $combCand_k$ 의 참조 인덱스들, 예측 리스트 이용 플래그들, 및 모션 벡터들은 다음과 같이 도출되고 $numCurrMergeCand$ 는 1만큼 증가된다:

[0325] $refIdxL0combCand_k = refIdxL010Cand$ (H-101)

[0326] $refIdxL1combCand_k = refIdxL111Cand$ (H-102)

[0327] $predFlagL0combCand_k = 1$ (H-103)

[0328] $predFlagL1combCand_k = 1$ (H-104)

[0329] $mvL0combCand_k[0] = mvL010Cand[0]$ (H-105)

[0330] $mvL0combCand_k[1] = mvL010Cand[1]$ (H-106)

[0331] $mvL1combCand_k[0] = mvL111Cand[0]$ (H-107)

[0332] $mvL1combCand_k[1] = mvL111Cand[1]$ (H-108)

[0333] $numCurrMergeCand = numCurrMergeCand + 1$ (H-109)

[0334] 4. 변수 $combIdx$ 는 1만큼 증가된다.

[0335] 5. $combIdx$ 가 ($numOrigMergeCand * (numOrigMergeCand - 1)$) 과 동일하거나 또는 $numCurrMergeCand$ 가 $MaxNumMergeCand$ 과 동일한 경우, $combStop$ 은 TRUE와 동일하게 설정된다.]]

[0336] 위에서 도시된 바와 같이, "mergeCandIsVspFlag"는 3D-HEVC 초안 텍스트 1의 섹션 H.8.5.3.2.1에 정의된 BVSP 플래그들의 임의의 어레이이다. "mergeCandIsVspFlag" 어레이에서의 각각의 값은 리스트에서의 병합 후보에 대응하고 대응하는 병합 후보가 BVSP에 기초하는지의 여부를 나타낸다. 섹션 H.8.5.3.2.1의 단계 10에서 "mergeCandIsVspFlag"는 삭제되어서, "mergeCandIsVspFlag"는 조합된 양방향-예측 병합 후보들에 대한 도출 프로세스에의 입력으로서 제공되지 않는다. 더욱이, 본 개시물의 하나 이상의 기법들에 따라, 섹션 H.8.5.3.2.3은 3D-HEVC 초안 텍스트 1로부터 삭제되는데, 조합된 양방향-예측 병합 후보들에 대한 도출 프로세스가 3D-HEVC에서는 HEVC에서 정의된 것 (즉, HEVC 규격 초안 10의 섹션 8.5.3.2.3) 과는 동일하기 때문이다. 덧붙여, 본 개시물의 하나 이상의 기법들에 따라, 위의 텍스트에서 도시된 바와 같이, $mergeCandIsVspFlag[numOrigMergeCand + k]$ 는 0과 동일하게 설정되지 않는데, 그렇게 할 필요가 더 이상 없기 때문이다.

[0337] 도 8은 본 개시물의 기법들을 구현할 수도 있는 일 예의 비디오 인코더 (20) 를 도시하는 블록도이다. 도 8은 설명의 목적으로 제공되고 본 개시물에서 폭넓게 예시되고 설명된 바와 같은 기법들의 제한으로서 고려되지

않아야 한다. 설명의 목적으로, 본 개시물은 HEVC 코딩의 맥락에서 비디오 인코더 (20) 를 설명한다. 그러나, 본 개시물의 기법들은 다른 코딩 표준들 또는 방법들에 적용가능할 수도 있다.

[0338] 도 8의 예에서, 비디오 인코더 (20) 는 예측 프로세싱 유닛 (100), 비디오 데이터 메모리 (101), 잔차 생성 유닛 (102), 변환 프로세싱 유닛 (104), 양자화 유닛 (106), 역 양자화 유닛 (108), 역 변환 프로세싱 유닛 (110), 복원 유닛 (112), 필터 유닛 (114), 디코딩된 화상 버퍼 (116), 및 엔트로피 인코딩 유닛 (118) 을 포함한다. 예측 프로세싱 유닛 (100) 은 인터 예측 프로세싱 유닛 (120) 과 인트라 예측 프로세싱 유닛 (126) 을 구비한다. 인터 예측 프로세싱 유닛 (120) 은 모션 추정 유닛 (122) 과 모션 보상 유닛 (124) 을 구비한다. 다른 예들에서, 비디오 인코더 (20) 는 더 많거나, 더 적거나, 또는 상이한 기능성 컴포넌트들을 포함할 수도 있다.

[0339] 비디오 인코더 (20) 는 비디오 데이터를 수신할 수도 있다. 비디오 데이터 메모리 (101) 는 비디오 인코더 (20) 의 컴포넌트들에 의해 인코딩될 비디오 데이터를 저장할 수도 있다. 비디오 데이터 메모리 (101) 에 저장된 비디오 데이터는, 예를 들어, 비디오 소스 (18) 로부터 획득될 수도 있다. 디코딩된 화상 버퍼 (116) 가, 예컨대, 인트라 코딩 또는 인터 코딩 모드들에서 비디오 인코더 (20) 에 의해 비디오 데이터를 인코딩함에 있어서의 사용을 위해 참조 비디오 데이터를 저장하는 참조 화상 메모리일 수도 있다. 비디오 데이터 메모리 (101) 와 디코딩된 화상 버퍼 (116) 는 동기식 DRAM (SDRAM) 을 포함한 다이내믹 랜덤 액세스 메모리 (DRAM), 자기저항 RAM (MRAM), 저항 RAM (RRAM), 또는 다른 유형의 메모리 디바이스들과 같은 다양한 메모리 디바이스들 중 임의의 것에 의해 형성될 수도 있다. 비디오 데이터 메모리 (101) 와 디코딩된 화상 버퍼 (116) 는 동일한 메모리 디바이스 또는 별개의 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에서, 비디오 데이터 메모리 (101) 는 비디오 인코더 (20) 의 다른 컴포넌트들과 온-칩, 또는 그들 컴포넌트들에 대하여 오프-칩일 수도 있다.

[0340] 비디오 인코더 (20) 는 비디오 데이터의 화상의 슬라이스에서의 각각의 CTU를 인코딩할 수도 있다. CTU들의 각각은 화상의 동일 사이즈로 된 루마 코딩 트리 블록 (CTB) 들 및 대응하는 CTB들에 연관될 수도 있다. CTU를 인코딩하는 부분으로서, 예측 프로세싱 유닛 (100) 은 쿼드트리 구획화를 수행하여 CTU의 CTB들을 점차적으로 더 작은 블록들로 분할할 수도 있다. 더 작은 블록은 CU들의 코딩 블록들일 수도 있다. 예를 들어, 예측 프로세싱 유닛 (100) 은 CTU에 연관된 CTB를 네 개의 동일 사이즈로 된 서브-블록들로 구획화하며, 그 서브-블록들 중 하나 이상을 네 개의 동일 사이즈로 된 서브 서브-블록들로 구획하는 등등을 수행할 수도 있다.

[0341] 비디오 인코더 (20) 는 CTU의 CU들을 인코딩하여 CU들의 인코딩된 표현들 (즉, 코딩된 CU들) 을 생성할 수도 있다. CU를 인코딩하는 부분으로서, 예측 프로세싱 유닛 (100) 은 CU의 하나 이상의 PU들 중에서 CU에 연관된 코딩 블록들을 구획화할 수도 있다. 따라서, 몇몇 예들에서, 각각의 PU는 루마 예측 블록 및 대응하는 크로마 예측 블록들에 연관될 수도 있다. 비디오 인코더 (20) 와 비디오 디코더 (30) 는 다양한 사이즈들을 갖는 PU들을 지원할 수도 있다. 위에서 나타낸 바와 같이, CU의 사이즈는 CU의 루마 코딩 블록의 사이즈를 지칭할 수도 있고 PU의 사이즈는 PU의 루마 예측 블록의 사이즈를 지칭할 수도 있다. 특정 CU의 사이즈가 $2N \times 2N$ 이라고 가정하면, 비디오 인코더 (20) 와 비디오 디코더 (30) 는 인트라 예측을 위한 $2N \times 2N$ 또는 $N \times N$ 의 PU 사이즈들과, 인터 예측을 위한 $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$, 또는 유사한 것의 대칭적 PU 사이즈들을 지원할 수도 있다. 비디오 인코더 (20) 와 비디오 디코더 (30) 는 인터 예측을 위해 $2N \times nU$, $2N \times nD$, $nL \times 2N$, 및 $nR \times 2N$ 의 PU 사이즈들에 대한 비대칭 구획화를 또한 지원할 수도 있다.

[0342] 인터 예측 프로세싱 유닛 (120) 은 CU의 각각의 PU에 대해 인터 예측을 수행함으로써 PU에 대한 예측 데이터를 생성할 수도 있다. PU에 대한 예측 데이터는 PU의 예측 블록들 및 그 PU에 대한 모션 정보를 포함할 수도 있다. 인터 예측 프로세싱 유닛 (120) 은 PU가 I 슬라이스, P 슬라이스, 또는 B 슬라이스 중 어느 것에 있는지에 의존하여 CU의 PU에 대해 상이한 동작들을 수행할 수도 있다. I 슬라이스에서, 모든 PU들이 인트라 예측될 수도 있다. 그러므로, PU가 I 슬라이스에 있다면, 인터 예측 프로세싱 유닛 (120) 은 PU에 대해 인터 예측을 수행하지 않는다. 따라서, I-모드에서 인코딩된 블록들에 대해, 예측된 블록은 동일한 프레임 내의 이전에 인코딩된 이웃 블록들로부터 공간적 예측을 사용하여 형성된다.

[0343] PU가 P 슬라이스에 있다면, 모션 추정 유닛 (122) 은 PU에 대한 참조 지역을 참조 화상들의 리스트 (예컨대, "RefPicList0") 에서의 참조 화상들에서 검색할 수도 있다. PU에 대한 참조 지역은, 참조 화상 내의, PU의 샘플 블록들에 가장 밀접하게 대응하는 샘플 블록들을 포함하는 지역일 수도 있다. 모션 추정 유닛 (122) 은 PU에 대한 참조 지역을 포함하는 참조 화상의 RefPicList0에서의 포지션을 나타내는 참조 인덱스를 생성할

수도 있다. 덧붙여서, 모션 추정 유닛 (122)은 PU의 코딩 블록 및 참조 지역에 연관된 참조 로케이션 사이의 공간적 변위를 나타내는 모션 벡터를 생성할 수도 있다. 예를 들면, 모션 벡터는 현재 화상에서의 좌표들로부터 참조 화상에서의 좌표들로의 오프셋을 제공하는 2차원 벡터일 수도 있다. 모션 추정 유닛 (122)은 참조 인덱스 및 모션 벡터를 PU의 모션 정보로서 출력할 수도 있다. 모션 보상 유닛 (124)은 PU의 모션 벡터에 의해 나타내어진 참조 로케이션에 연관된 실제 또는 보간된 샘플들에 기초하여 PU의 예측 블록들 (즉, predictive blocks)을 생성할 수도 있다.

- [0344] PU가 B 슬라이스에 있다면, 모션 추정 유닛 (122)은 PU에 대해 단-예측 또는 양방향-예측을 수행할 수도 있다. PU에 대한 단-예측을 수행하기 위해, 모션 추정 유닛 (122)은 PU에 대한 참조 지역을 RefPicList0 또는 제 2 참조 화상 리스트 (예컨대, "RefPicList1")의 참조 화상들에서 검색할 수도 있다. 모션 추정 유닛 (122)은, PU의 모션 정보로서, 참조 지역을 포함하는 참조 화상의 RefPicList0 또는 RefPicList1에서의 포지션을 나타내는 참조 인덱스, PU의 샘플 블록 및 참조 지역에 연관된 참조 로케이션 사이의 공간적 변위를 나타내는 모션 벡터, 및 참조 화상이 RefPicList0에 있는지 또는 RefPicList1에 있는지를 나타내는 하나 이상의 예측 방향 표시자들을 출력할 수도 있다. 모션 보상 유닛 (124)은 PU의 모션 벡터에 의해 나타내어진 참조 로케이션에 연관된 실제 또는 보간된 샘플들에 적어도 부분적으로 기초하여 PU의 예측 블록들을 생성할 수도 있다.
- [0345] PU에 대한 양방향 인터 예측을 수행하기 위해, 모션 추정 유닛 (122)은 그 PU에 대한 참조 지역을 RefPicList0에서의 참조 화상들에서 검색할 수도 있고, 또한 그 PU에 대한 다른 참조 지역을 RefPicList1에서의 참조 화상들에서 검색할 수도 있다. 모션 추정 유닛 (122)은 참조 지역들을 포함하는 참조 화상들의 RefPicList0 및 RefPicList1에서의 포지션들을 나타내는 참조 인덱스들을 생성할 수도 있다. 덧붙여서, 모션 추정 유닛 (122)은 그 참조 지역들에 연관된 참조 로케이션들 및 PU의 예측 블록 (예컨대, 샘플 블록) 사이의 공간적 변위들을 나타내는 모션 벡터들을 생성할 수도 있다. PU의 모션 정보는 PU의 참조 인덱스들 및 모션 벡터들을 포함할 수도 있다. 모션 보상 유닛 (124)은 PU의 모션 벡터들에 의해 나타내어진 참조 지역에 연관된 실제 또는 보간된 샘플들에 적어도 부분적으로 기초하여 PU의 예측 블록들을 생성할 수도 있다.
- [0346] 본 개시물의 하나 이상의 기법들에 따라, 모션 추정 유닛 (122)은 3D 비디오의 비디오 블록을 코딩하기 위해 병합 후보들의 리스트를 생성할 수도 있다. 병합 후보들의 리스트를 생성하는 부분으로서, 모션 추정 유닛 (122)은 병합 후보들의 리스트에서의 병합 후보들의 수가 5 미만인지의 여부를 결정할 수도 있다. 병합 후보들의 리스트에서의 병합 후보들의 수가 5 미만이라는 결정에 응답하여, 모션 추정 유닛 (122)은 하나 이상의 조합된 양방향-예측 병합 후보들을 도출할 수도 있다. 모션 추정 유닛 (122)은 병합 후보들의 리스트에 하나 이상의 조합된 양방향-예측 병합 후보들을 포함시킬 수도 있다. 더욱이, 몇몇 예들에서, 모션 추정 유닛 (122)은 병합 후보들의 리스트에서 병합 후보를 선택할 수도 있다. 비디오 인코더 (20)는 선택된 병합 후보의 병합 후보들의 리스트에서의 포지션을 시그널링할 수도 있다. 몇몇 예들에서, 병합 후보들의 리스트에서의 병합 후보들의 최대 수는 5보다 더 크다 (예컨대, 6과 동일하다).
- [0347] 이제 도 8의 예가 계속 참조된다. 인트라 예측 프로세싱 유닛 (126)은 PU에 대해 인트라 예측을 수행함으로써 그 PU에 대한 예측 데이터를 생성할 수도 있다. PU에 대한 예측 데이터는 PU에 대한 예측 블록들과 다양한 신덱스 엘리먼트들을 포함할 수도 있다. 인트라 예측 프로세싱 유닛 (126)은 I 슬라이스들, P 슬라이스들, 및 B 슬라이스들에서의 PU들에 대해 인트라 예측을 수행할 수도 있다.
- [0348] PU에 대해 인트라 예측을 수행하기 위해, 인트라 예측 프로세싱 유닛 (126)은 PU에 대한 예측 데이터의 다수의 세트들을 생성하기 위해 다수의 인트라 예측 모드들을 사용할 수도 있다. 몇몇 인트라 예측 모드들을 사용하여 PU에 대한 예측 데이터의 세트를 생성하기 위해, 인트라 예측 프로세싱 유닛 (126)은 이웃 블록들로부터 인트라 예측 모드에 연관된 방향에서 PU의 예측 블록을 가로질러 샘플들을 확장할 수도 있다. PU들, CU들, 및 CTU들에 대한 좌측에서 우측으로, 상단에서 하단으로의 인코딩 순서를 가정하면, 이웃하는 PU들은 PU의 상측, 우상측, 좌상측, 또는 좌측에 있을 수도 있다. 인트라 예측 프로세싱 유닛 (126)은 다양한 수들의 인트라 예측 모드들, 예컨대, 33 개의 방향성 인트라 예측 모드들을 사용할 수도 있다. 몇몇 예들에서, 인트라 예측 모드들의 수는 PU에 연관된 지역의 사이즈에 의존할 수도 있다.
- [0349] 예측 프로세싱 유닛 (100)은 CU의 PU들에 대한 예측 데이터를, 그 PU들에 대해 인터 예측 프로세싱 유닛 (120)에 의해 생성된 예측 데이터 또는 그 PU들에 대해 인트라 예측 프로세싱 유닛 (126)에 의해 생성된 예측 데이터 중에서 선택할 수도 있다. 몇몇 예들에서, 예측 프로세싱 유닛 (100)은 예측 데이터의 세트들의 레이트/왜곡 메트릭들에 기초하여 CU의 PU들에 대한 예측 데이터를 선택한다. 선택된 예측 데이터의 예측 블록들은 본원에서는 선택된 예측 블록들이라고 지칭될 수도 있다.

- [0350] 잔차 생성 유닛 (102) 은, CU의 코딩 블록 (예컨대, 루마, Cb 및 Cr 코딩 블록들) 및 그 CU의 PU들의 선택된 예측 블록들 (예컨대, 예측 루마, Cb 및 Cr 블록들) 에 기초하여, CU의 잔차 블록들 (예컨대, 잔차 루마, Cb 및 Cr 잔차 블록들) 을 생성할 수도 있다. 다르게 말하면, 잔차 생성 유닛 (102) 은 CU에 대한 잔차 신호를 생성할 수도 있다. 예를 들면, 잔차 생성 유닛 (102) 은 CU의 잔차 블록들에서의 각각의 샘플이 그 CU의 코딩 블록에서의 샘플 및 그 CU의 PU의 대응하는 선택된 예측 블록에서의 대응하는 샘플 사이의 차이와 동일한 값을 가지도록 CU의 잔차 블록들을 생성할 수도 있다.
- [0351] 변환 프로세싱 유닛 (104) 은 쿼드트리 구획화를 수행하여 CU에 연관된 잔차 블록들을 CU의 TU들에 대응하는 (즉, 그 TU들에 연관된) 변환 블록들로 구획화할 수도 있다. 따라서, TU가 루마 변환 블록 및 2 개의 크로마 변환 블록들에 연관될 수도 있다. CU의 TU들의 변환 블록들 (예컨대, 루마 및 크로마 변환 블록들) 의 사이즈들 및 포지션들은 그 CU의 PU들의 예측 블록들의 사이즈들 및 포지션들에 기초할 수도 있거나 또는 기초하지 않을 수도 있다. "잔차 쿼드 트리" (residual quad-tree; RQT) 로서 알려진 쿼드트리 구조가 TU들의 각각에 연관된 노드들을 포함할 수도 있다. CU의 TU들은 RQT의 리프 노드들에 대응할 수도 있다.
- [0352] 변환 프로세싱 유닛 (104) 은 CU의 각각의 TU에 대한 변환 계수 블록들을, 하나 이상의 변환들을 그 TU의 변환 블록들에 적용함으로써 생성할 수도 있다. 변환 프로세싱 유닛 (104) 은 다양한 변환들을 TU에 연관된 변환 블록에 적용할 수도 있다. 예를 들어, 변환 프로세싱 유닛 (104) 은 이산 코사인 변환 (DCT), 방향성 변환, 또는 개념적으로 유사한 변환을 변환 블록에 적용할 수도 있다. 몇몇 예들에서, 변환 프로세싱 유닛 (104) 은 변환들을 변환 블록에 적용하지 않는다. 그런 예들에서, 변환 블록은 변환 계수 블록으로서 다루어질 수도 있다.
- [0353] 양자화 유닛 (106) 은 변환 계수 블록에서의 변환 계수들을 양자화할 수도 있다. 양자화 프로세스는 변환 계수 블록의 변환 계수들의 일부 또는 전부에 연관된 비트 깊이를 감소시킬 수도 있다. 예를 들어, n -비트 변환 계수가 양자화 동안에 m -비트 변환 계수로 버림될 (rounded down) 수도 있으며, 여기서 n 은 m 보다 크다. 양자화 유닛 (106) 은 CU에 연관된 양자화 파라미터 (QP) 값에 기초하여 그 CU의 TU에 연관된 변환 계수 블록을 양자화할 수도 있다. 비디오 인코더 (20) 는 CU에 연관된 QP 값을 조정하는 것에 의해 그 CU에 연관된 변환 계수 블록들에 적용되는 양자화 정도를 조정할 수도 있다. 양자화는 정보의 손실을 도입할 수도 있고, 그러므로, 양자화된 변환 계수들은 원래의 것들보다 낮은 정밀도를 가질 수도 있다.
- [0354] 역 양자화 유닛 (108) 과 역 변환 프로세싱 유닛 (110) 은 역 양자화 및 역 변환들을 변환 계수 블록에 각각 적용하여, 변환 계수 블록으로부터 잔차 블록 (즉, 변환 블록) 을 복원할 수도 있다. 복원 유닛 (112) 은 코딩 블록의 각각의 샘플이 CU의 PU의 예측 블록의 샘플 및 CU의 TU의 변환 블록의 대응 샘플의 합과 동일하도록 CU의 코딩 블록을 복원할 수도 있다. 예를 들어, 복원 유닛 (112) 은 CU의 TU들의 복원된 잔차 블록들을 예측 프로세싱 유닛 (100) 에 의해 생성된 CU의 PU들의 하나 이상의 예측 블록들로부터의 대응 샘플들에 가산하여 CU의 복원된 코딩 블록들을 생성할 수도 있다. 따라서, CU의 각각의 TU에 대한 변환 블록들을 복원함으로써, 비디오 인코더 (20) 는 CU의 코딩 블록들을 복원할 수도 있다.
- [0355] 필터 유닛 (114) 은 하나 이상의 블록화제거 동작들을 수행하여 CU에 연관된 코딩 블록들에서의 블록화 아티팩트들을 감소시킬 수도 있다. 디코딩된 화상 버퍼 (116) 는, 필터 유닛 (114) 이 복원된 코딩 블록들에 대해 하나 이상의 블록화제거 동작들을 수행한 후에 복원된 코딩 블록들을 저장할 수도 있다. 따라서, 디코딩된 화상 버퍼 (116) 는 비디오 데이터를 저장하도록 구성된 메모리일 수도 있다. 인터 예측 프로세싱 유닛 (120) 은 다른 화상들의 PU들에 대해 인터 예측을 수행하기 위해 복원된 코딩 블록들을 포함하는 참조 화상을 사용할 수도 있다. 덧붙여서, 인트라 예측 프로세싱 유닛 (126) 은 CU와 동일한 화상에서의 다른 PU들에 대해 인트라 예측을 수행하기 위해 디코딩된 화상 버퍼 (116) 에서의 복원된 코딩 블록들을 사용할 수도 있다.
- [0356] 엔트로피 인코딩 유닛 (118) 은 비디오 인코더 (20) 의 다른 기능성 컴포넌트들로부터 데이터를 수신할 수도 있다. 예를 들어, 엔트로피 인코딩 유닛 (118) 은 양자화 유닛 (106) 으로부터 계수 블록들을 수신할 수도 있고 예측 프로세싱 유닛 (100) 으로부터 선택스 엘리먼트들을 수신할 수도 있다. 엔트로피 인코딩 유닛 (118) 은 데이터에 대해 하나 이상의 엔트로피 인코딩 동작들을 수행하여 엔트로피 인코딩된 데이터를 생성할 수도 있다. 예를 들어, 엔트로피 인코딩 유닛 (118) 은 그 데이터에 대해 CABAC 동작, 컨텍스트 적응 가변 길이 코딩 (context-adaptive variable length coding, CAVLC) 동작, 가변 대 가변 (variable-to-variable, V2V) 길이 코딩 동작, 선택스 기반 컨텍스트 적응 이진 산술 코딩 (syntax-based context-adaptive binary arithmetic coding, SBAC) 동작, 확률 간격 구획화 엔트로피 (Probability Interval Partitioning Entropy, PIPE) 코딩 동작, 지수-골롬 (Exponential-Golomb) 인코딩 동작, 또는 다른 유형의 엔트로피 인코딩 동작을 수

행할 수도 있다. 비디오 인코더 (20) 는 엔트로피 인코딩 유닛 (118) 에 의해 생성된 엔트로피 인코딩된 데이터를 포함하는 비트스트림을 출력할 수도 있다. 예를 들면, 그 비트스트림은 CU에 대한 RQT를 표현하는 데이터를 포함할 수도 있다. 비트스트림은 엔트로피 인코딩되지 않은 데이터를 포함할 수도 있다.

[0357] 도 9는 본 개시물의 기법들을 구현하도록 구성되는 일 예의 비디오 디코더 (30) 를 도시하는 블록도이다. 도 9는 설명의 목적으로 제공되고 본 개시물에서 폭넓게 예시되고 설명된 바와 같은 기법들로 제한하고 있지는 않다. 설명의 목적으로, 본 개시물은 HEVC 코딩의 맥락에서 비디오 디코더 (30) 를 설명한다. 그러나, 본 개시물의 기법들은 다른 코딩 표준들 또는 방법들에 적용 가능할 수도 있다.

[0358] 도 9의 예에서, 비디오 디코더 (30) 는 엔트로피 디코딩 유닛 (150), 비디오 데이터 메모리 (151), 예측 프로세싱 유닛 (152), 역 양자화 유닛 (154), 역 변환 프로세싱 유닛 (156), 복원 유닛 (158), 필터 유닛 (160), 및 디코딩된 화상 버퍼 (162) 를 구비한다. 예측 프로세싱 유닛 (152) 은 모션 보상 유닛 (164) 과 인트라 예측 프로세싱 유닛 (166) 을 구비한다. 다른 예들에서, 비디오 디코더 (30) 는 더 많거나, 더 적거나, 또는 상이한 기능성 컴포넌트들을 포함할 수도 있다.

[0359] 비디오 디코더 (30) 는 비트스트림을 수신할 수도 있다. 비디오 데이터 메모리 (151) 는, 비디오 디코더 (30) 의 컴포넌트들에 의해 디코딩될 인코딩된 비디오 비트스트림과 같은 비디오 데이터를 저장할 수도 있다. 비디오 데이터 메모리 (151) 에 저장된 비디오 데이터는, 예를 들어, 채널 (16) 로부터, 예컨대, 카메라와 같은 로컬 비디오 소스로부터, 비디오 데이터의 유선 또는 무선 네트워크 통신을 통해, 또는 물리적 데이터 저장 매체들에 액세스함으로써 획득될 수도 있다. 비디오 데이터 메모리 (151) 는 인코딩된 비디오 비트스트림으로부터의 인코딩된 비디오 데이터를 저장하는 코딩된 화상 버퍼 (CPB) 를 형성할 수도 있다. 디코딩된 화상 버퍼 (162) 가, 예컨대, 인트라 코딩 또는 인터 코딩 모드들에서 비디오 디코더 (30) 에 의해 비디오 데이터를 디코딩함에 있어서의 사용을 위해 참조 비디오 데이터를 저장하는 참조 화상 메모리일 수도 있다. 비디오 데이터 메모리 (151) 와 디코딩된 화상 버퍼 (162) 는 동기식 DRAM (SDRAM) 을 포함한 다이내믹 랜덤 액세스 메모리 (DRAM), 자기저항 RAM (MRAM), 저항 RAM (RRAM), 또는 다른 유형들의 메모리 디바이스들과 같은 다양한 메모리 디바이스들 중 임의의 것에 의해 형성될 수도 있다. 비디오 데이터 메모리 (151) 와 디코딩된 화상 버퍼 (162) 는 동일한 메모리 디바이스 또는 별개의 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에서, 비디오 데이터 메모리 (151) 는 비디오 디코더 (30) 의 다른 컴포넌트들과 온-칩, 또는 그들 컴포넌트들에 대하여 오프-칩일 수도 있다.

[0360] 엔트로피 디코딩 유닛 (150) 은 비트스트림을 파싱하여 비트스트림으로부터 신택스 엘리먼트들을 디코딩할 수도 있다. 엔트로피 디코딩 유닛 (150) 은 비트스트림에서의 엔트로피 인코딩된 신택스 엘리먼트들을 엔트로피 디코딩할 수도 있다. 예측 프로세싱 유닛 (152), 역 양자화 유닛 (154), 역 변환 프로세싱 유닛 (156), 복원 유닛 (158), 및 필터 유닛 (160) 은 비트스트림으로부터 획득된 (예컨대, 추출된) 신택스 엘리먼트들에 기초하여 디코딩된 비디오 데이터를 생성할 수도 있다.

[0361] 그 비트스트림은 일련의 NAL 단위들을 포함할 수도 있다. 비트스트림의 NAL 단위들은 코딩된 슬라이스 NAL 단위들을 포함할 수도 있다. 비트스트림을 디코딩하는 부분으로서, 엔트로피 디코딩 유닛 (150) 은 코딩된 슬라이스 NAL 단위들로부터 신택스 엘리먼트들을 획득 (예컨대, 추출) 및 엔트로피 디코딩할 수도 있다. 코딩된 슬라이스들의 각각은 슬라이스 헤더 및 슬라이스 데이터를 포함할 수도 있다. 슬라이스 헤더는 슬라이스에 관계된 신택스 엘리먼트들을 포함할 수도 있다. 슬라이스 헤더에서의 신택스 엘리먼트들은 슬라이스를 포함하는 화상에 연관된 PPS를 식별하는 신택스 엘리먼트를 포함할 수도 있다.

[0362] 비트스트림으로부터 신택스 엘리먼트들을 획득 (예컨대, 디코딩) 하는 것 외에도, 비디오 디코더 (30) 는 CU들에 대해 복원 동작을 수행할 수도 있다. CU (예컨대, 비구획화된 CU) 에 대해 복원 동작을 수행하기 위해, 비디오 디코더 (30) 는 CU의 각각의 TU에 대해 복원 동작을 수행할 수도 있다. CU의 각각의 TU에 대한 복원 동작을 수행함으로써, 비디오 디코더 (30) 는 CU의 TU들의 잔차 블록들 (예컨대, 변환 블록들) 을 복원할 수도 있다.

[0363] CU의 TU에 대해 복원 동작을 수행하는 부분으로서, 역 양자화 유닛 (154) 은 TU의 (즉, TU에 연관된) 계수 블록들을 역 양자화, 즉, 탈양자화 (de-quantization) 할 수도 있다. 역 양자화 유닛 (154) 은 TU의 CU에 연관된 QP 값을 사용하여 양자화 정도를 결정하고, 비슷하게, 역 양자화 유닛 (154) 에 대해 적용할 역 양자화 정도를 결정할 수도 있다. 다시 말하면, 압축 비율, 즉, 원래의 시퀀스 및 압축된 시퀀스를 표현하는데 사용된 비트들의 수의 비율은, 변환 계수들을 양자화하는 경우에 사용된 QP의 값을 조정함으로써 제어될 수도 있다. 압축 비율은 채용된 엔트로피 코딩하는 방법에 또한 의존할 수도 있다.

- [0364] 역 양자화 유닛 (154) 이 계수 블록을 역 양자화한 후, 역 변환 프로세싱 유닛 (156) 은 TU에 연관된 잔차 블록을 생성하기 위하여 하나 이상의 역 변환들을 계수 블록에 적용할 수도 있다. 예를 들어, 역 변환 프로세싱 유닛 (156) 은 역 DCT, 역 정수 변환, 역 카루넨-뢰베 변환 (Karhunen-Loeve transform; KLT), 역 회전 변환, 역 방향성 변환, 또는 다른 역 변환을 계수 블록에 적용할 수도 있다.
- [0365] PU가 인트라 예측을 사용하여 인코딩되면, 인트라 예측 프로세싱 유닛 (166) 은 PU에 대한 예측 블록들을 생성하기 위해 인트라 예측을 수행할 수도 있다. 예를 들면, 인트라 예측 프로세싱 유닛 (166) 은 인트라 예측 모드를 사용하여 공간적으로 이웃하는 PU들의 예측 블록들에 기초하여 PU에 대한 예측 루마, Cb 및 Cr 블록들을 생성할 수도 있다. 인트라 예측 프로세싱 유닛 (166) 은 비트스트림으로부터 디코딩된 하나 이상의 신택스 엘리먼트들에 기초하여 PU에 대한 인트라 예측 모드를 결정할 수도 있다.
- [0366] 예측 프로세싱 유닛 (152) 은 비트스트림으로부터 획득된 신택스 엘리먼트들에 기초하여 제 1 참조 화상 리스트 (RefPicList0) 및 제 2 참조 화상 리스트 (RefPicList1) 를 구축할 수도 있다. 더욱이, PU가 인트라 예측을 사용하여 인코딩되면, 엔트로피 디코딩 유닛 (150) 은 그 PU에 대한 모션 정보를 결정 (예컨대, 추출) 할 수도 있다. 모션 보상 유닛 (164) 은, PU의 모션 정보에 기초하여, PU에 대한 하나 이상의 참조 블록들을 결정할 수도 있다. 모션 보상 유닛 (164) 은, PU에 대한 하나 이상의 참조 블록들에서의 샘플 블록들에 기초하여, PU에 대한 예측 블록들 (예컨대, 예측 루마, Cb 및 Cr 블록들) 을 생성할 수도 있다.
- [0367] 위에서 나타난 바와 같이, 비디오 인코더 (20) 는 병합 모드, 스킵 모드 또는 AMVP 모드를 사용하여 PU의 모션 정보를 시그널링할 수도 있다. 비디오 인코더 (20) 가 AMVP 모드를 사용하여 현재 PU의 모션 정보를 시그널링하는 경우, 엔트로피 디코딩 유닛 (150) 은, 비트스트림으로부터, 참조 인덱스, 현재 PU에 대한 MVD, 및 후보 인덱스를 디코딩할 수도 있다. 더욱이, 모션 보상 유닛 (164) 은 현재 PU에 대한 AMVP 후보 리스트를 생성할 수도 있다. AMVP 후보 리스트는 하나 이상의 모션 벡터 예측자 후보들을 포함한다. 모션 벡터 예측자 후보들의 각각은 현재 PU에 공간적으로 또는 시간적으로 이웃하는 PU의 모션 벡터를 특정한다. 모션 보상 유닛 (164) 은, 후보 인덱스에 적어도 부분적으로 기초하여, AMVP 후보 리스트에서 선택된 모션 벡터 예측자 후보를 결정할 수도 있다. 모션 보상 유닛 (164) 은 그 다음에 MVD를 선택된 모션 벡터 예측변수 후보에 의해 특정된 모션 벡터에 가산함으로써 현재 PU의 모션 벡터를 결정할 수도 있다. 다르게 말하면, AMVP에 대해, 모션 벡터는 모션 벡터 (MV) = MVP + MVD로서 계산되며, 여기서 모션 벡터 예측자 (MVP) 의 인덱스는 시그널링되고 MVP는 AMVP 리스트로부터의 (공간적 또는 시간적) 모션 벡터 후보들 중 하나이고, MVD는 디코더 측으로 시그널링된다.
- [0368] 현재 PU가 양방향-예측되고 PU의 모션 정보가 AMVP 모드에서 시그널링되면, 엔트로피 디코딩 유닛 (150) 은 비트스트림으로부터 부가적인 참조 인덱스, MVD, 및 후보 인덱스를 디코딩할 수도 있다. 모션 보상 유닛 (162) 은 부가적인 참조 인덱스, MVD, 및 후보 인덱스를 사용하여 위에서 설명된 프로세스를 반복하여 현재 PU에 대한 제 2 모션 벡터를 도출할 수도 있다. 이런 식으로, 모션 보상 유닛 (162) 은 RefPicList0에 대한 모션 벡터 (즉, RefPicList0 모션 벡터) 및 RefPicList1에 대한 모션 벡터 (즉, RefPicList1 모션 벡터) 를 도출할 수도 있다.
- [0369] 본 개시물의 하나 이상의 기법들에 따라, 모션 보상 유닛 (164) 은 3D 비디오의 비디오 블록을 코딩하기 위한 병합 후보들의 리스트를 생성할 수도 있다. 병합 후보들의 리스트를 생성하는 부분으로서, 모션 보상 유닛 (164) 은 병합 후보들의 리스트에서의 병합 후보들의 수가 5 미만인지의 여부를 결정할 수도 있다. 병합 후보들의 리스트에서의 병합 후보들의 수가 5 미만이라는 결정에 응답하여, 모션 보상 유닛 (164) 은 하나 이상의 조합된 양방향-예측 병합 후보들을 도출할 수도 있다. 모션 보상 유닛 (164) 은 병합 후보들의 리스트에 하나 이상의 조합된 양방향-예측 병합 후보들을 포함시킬 수도 있다. 더욱이, 몇몇 예들에서, 비디오 디코더 (30) 는, 비트스트림으로부터, 병합 후보들의 리스트에서의 선택된 병합 후보를 나타내는 신택스 엘리먼트를 획득할 수도 있다. 모션 보상 유닛 (30) 은 현재 PU의 예측 샘플들을 생성하기 위해 선택된 후보의 모션 정보를 사용할 수도 있다. 몇몇 예들에서, 병합 후보들의 리스트에서의 병합 후보들의 최대 수는 5보다 더 크다 (예컨대, 6과 동일하다).
- [0370] 이제 도 9에 대해 계속 참조한다. 복원 유닛 (158) 은, 적용가능한 것으로서, CU의 TU들의 변환 블록들 (예컨대, 루마, Cb 및 Cr 변환 블록들) 과 그 CU의 PU들의 예측 블록들 (예컨대, 예측 루마, Cb 및 Cr 블록들), 즉 인트라 예측 데이터 또는 인트라 예측 데이터 중 어느 하나를 사용하여 그 CU의 코딩 블록들 (예컨대, 루마, Cb 및 Cr 코딩 블록들) 을 복원할 수도 있다. 예를 들어, 복원 유닛 (158) 은 변환 블록들 (예컨대, 루마, Cb 및 Cr 변환 블록들) 의 샘플들을 예측 블록들 (예컨대, 예측 루마, Cb 및 Cr 블록들) 의 대응하는 샘플들에 가

산하여 CU의 코딩 블록들 (예컨대, 루마, Cb 및 Cr 코딩 블록들) 을 복원할 수도 있다.

[0371] 필터 유닛 (160) 은 블록화제거 동작을 수행하여 CU의 코딩 블록들 (예컨대, 루마, Cb 및 Cr 코딩 블록들) 에 연관된 블록화 아티팩트들을 감소시킬 수도 있다. 비디오 디코더 (30) 는 CU의 코딩 블록들 (예컨대, 루마, Cb 및 Cr 코딩 블록들) 을 디코딩된 화상 버퍼 (162) 에 저장할 수도 있다. 디코딩된 화상 버퍼 (162) 는 후속하는 모션 보상, 인트라 예측, 및 디스플레이 디바이스, 이를테면 도 1의 디스플레이 디바이스 (32) 상의 프레젠테이션을 위해 참조 화상들을 제공할 수도 있다. 예를 들면, 비디오 디코더 (30) 는, 디코딩된 화상 버퍼 (162) 에서의 블록들 (예컨대, 루마, Cb 및 Cr 블록들) 에 기초하여, 다른 CU들의 PU들에 대해 인트라 예측 또는 인트라 예측 동작들을 수행할 수도 있다. 이런 식으로, 비디오 디코더 (30) 는, 비트스트림으로부터, 중대한 루마 계수 블록의 변환 계수 레벨들을 획득하며, 그 변환 계수 레벨들을 역 양자화하며, 그 변환 계수 레벨들에 변환을 적용하여 변환 블록을 생성하며, 그 변환 블록에 적어도 부분적으로 기초하여, 코딩 블록을 생성하고, 그 코딩 블록을 디스플레이를 위해 출력할 수도 있다.

[0372] 도 10a는 본 개시물의 하나 이상의 기법들에 따른, 3D 비디오에 연관된 데이터를 인코딩하는 비디오 인코더 (20) 의 일 예의 동작을 도시하는 흐름도이다. 도 10a의 동작은, 본 개시물의 다른 흐름도들에 예시된 동작들과 함께, 예들이다. 본 개시물의 기법들에 따른 다른 예의 동작들은 더 많거나, 더 적거나, 또는 상이한 액션들을 포함할 수도 있다.

[0373] 도 10a의 예에서, 비디오 인코더 (20) 는 병합 후보들의 리스트를 생성할 수도 있다 (200). 다르게 말하면, 비디오 인코더 (20) 는 병합 후보 리스트를 생성할 수도 있다. 본 개시물에서의 다른 곳에서 설명되는 도 11와 도 12는, 병합 후보들의 리스트를 생성하는 일 예의 동작을 도시한다. 몇몇 예들에서, 비디오 인코더 (20) 는 병합 후보 리스트를 비디오 디코더 (30) 와 동일한 방식으로 생성할 수도 있다. 본 개시물의 하나 이상의 기법들에 따라, 비디오 인코더 (20) 가 병합 후보 리스트를 생성하는 경우, 비디오 인코더 (20) 는 병합 후보 리스트에서의 병합 후보들의 수가 5 미만인지의 여부를 결정할 수도 있다. 병합 후보 리스트에서의 병합 후보들의 수가 5 미만이라는 결정에 응답하여, 비디오 인코더 (20) 는 하나 이상의 양방향-예측 병합 후보들을 도출할 수도 있다. 비디오 인코더 (20) 는 병합 후보 리스트에 하나 이상의 양방향-예측 병합 후보들을 포함시킬 수도 있다. 몇몇 예들에서, 병합 후보 리스트에서의 병합 후보들의 최대 수는 6과 동일하다.

[0374] 더욱이, 도 10a의 예에서, 비디오 인코더 (20) 는 병합 후보들의 리스트에서 후보를 선택할 수도 있다 (202). 몇몇 예들에서, 비디오 인코더 (20) 는 선택된 후보를 비트스트림에서 시그널링할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 비트스트림에 병합 인덱스 신텍스 엘리먼트를 포함시킬 수도 있다. 비디오 인코더 (20) 는 선택된 후보에 기초하여 비디오 블록을 인코딩할 수도 있다 (204). 예를 들어, 비디오 블록은 CU일 수도 있다. 이 예에서, 비디오 인코더 (20) 는 CU의 PU에 대한 예측 블록을 결정하기 위해 선택된 후보의 모션 정보 (예컨대, 모션 벡터들, 참조 인덱스들 등) 를 사용할 수도 있다. 더욱이, 이 예에서, 비디오 인코더 (20) 는 CU의 예측 블록의 샘플들 및 코딩 블록의 대응 샘플들에 기초하여 변환 블록 (예컨대, 잔차 블록) 의 적어도 일부 샘플들의 값들을 결정할 수도 있다. 예를 들면, 비디오 인코더 (20) 는 변환 블록의 적어도 일부의 샘플들의 값들을 그 샘플들이 CU의 예측 블록의 샘플들 및 코딩 블록의 대응 샘플들 간의 차이들과 동일하도록 결정할 수도 있다.

[0375] 도 10b는 본 개시물의 하나 이상의 기법들에 따른, 3D 비디오에 연관된 데이터를 디코딩하는 비디오 디코더 (30) 의 일 예의 동작을 도시하는 흐름도이다. 도 10b의 예에서, 비디오 디코더 (30) 는 병합 후보들의 리스트를 생성할 수도 있다 (220). 다르게 말하면, 비디오 디코더 (30) 는 병합 후보 리스트를 생성할 수도 있다. 본 개시물에서의 다른 곳에서 설명되는 도 11와 도 12는, 병합 후보들의 리스트를 생성하는 일 예의 동작을 도시한다. 몇몇 예들에서, 비디오 디코더 (30) 는 병합 후보 리스트를 비디오 인코더 (20) 와 동일한 방식으로 생성할 수도 있다. 본 개시물의 하나 이상의 기법들에 따라, 비디오 디코더 (30) 가 병합 후보 리스트를 생성하는 경우, 비디오 디코더 (30) 는 병합 후보 리스트에서의 병합 후보들의 수가 5 미만인지의 여부를 결정할 수도 있다. 병합 후보 리스트에서의 병합 후보들의 수가 5 미만이라는 결정에 응답하여, 비디오 디코더 (30) 는 하나 이상의 양방향-예측 병합 후보들을 도출할 수도 있다. 비디오 디코더 (30) 는 병합 후보 리스트에 하나 이상의 양방향-예측 병합 후보들을 포함시킬 수도 있다. 몇몇 예들에서, 병합 후보 리스트에서의 병합 후보들의 최대 수는 6과 동일하다.

[0376] 더욱이, 도 10b의 예에서, 비디오 디코더 (30) 는 병합 후보들의 리스트에서 선택된 후보를 결정할 수도 있다 (222). 몇몇 예들에서, 비디오 디코더 (30) 는 비트스트림에서 시그널링된 신텍스 엘리먼트에 의해 표시된 값에 기초하여 선택된 후보를 결정할 수도 있다. 비디오 디코더 (30) 는 선택된 후보에 기초하여 비디오 블

록을 디코딩할 수도 있다 (224). 예를 들어, 비디오 블록은 CU일 수도 있다. 이 예에서, 비디오 디코더 (30) 는 CU의 PU에 대한 예측 블록을 결정하기 위해 선택된 후보의 모션 정보 (예컨대, 모션 벡터들, 참조 인덱스들 등) 를 사용할 수도 있다. 더욱이, 이 예에서, 비디오 디코더 (30) 는 CU의 코딩 블록의 적어도 일부의 샘플들의 값들을 예측 블록에 기초하여 결정할 수도 있다. 예를 들면, 비디오 디코더 (30) 는 코딩 블록의 적어도 일부의 샘플들의 값들을 그 샘플들이 CU의 TU의 예측 블록의 샘플들 및 변환 블록의 대응 샘플들의 합들과 동일하도록 결정할 수도 있다.

[0377] 도 11은 본 개시물의 하나 이상의 기법들에 따른, 현재 블록에 대한 병합 후보 리스트를 구축하는 일 예의 동작 (300) 의 제 1 부분을 도시하는 흐름도이다. 도 11의 예의 동작에서, 하나 이상의 액션들이 재배열 또는 생략될 수도 있다. 다른 예들에서, 유사한 동작들이 추가적인 액션들을 포함할 수도 있다.

[0378] 도 11의 예에서, 비디오 코더 (예컨대, 비디오 인코더 (20) 또는 비디오 디코더 (30)) 가 IPMVC를 결정할 수도 있다 (302). 몇몇 예들에서, 비디오 코더는 현재 블록에 대한 디스패리티 벡터를 사용하여 IPMVC를 결정함으로써 뷰 간 참조 화상에서의 대응 블록을 식별할 수도 있다. 그런 예들에서, 대응 블록이 인트라 예측되지 않고 뷰 간 예측되지 않고 시간적 모션 벡터 (즉, 대응 블록과는 상이한 시간 인스턴스에 연관된 참조 화상에서의 로케이션을 표시하는 모션 벡터) 를 갖는다면, IPMVC는 대응 블록의 모션 벡터들, 대응 블록의 예측 방향 표시자들, 및 대응 블록의 변환된 참조 인덱스들을 특정할 수도 있다. 그 뒤에, 비디오 코더는 IPMVC가 이용 가능한지의 여부를 결정할 수도 있다 (304). 몇몇 예들에서, 뷰 간 참조 화상에서의 대응 블록이 인트라 예측되거나 또는 뷰 간 참조 화상의 경계들 외부에 있다면 IPMVC는 이용 불가능하다. IPMVC가 이용 가능하다는 결정에 응답하여 (304의 "예"), 비디오 코더는 병합 후보 리스트에 IPMVC를 삽입할 수도 있다 (306).

[0379] 병합 후보 리스트에 IPMV를 삽입한 후 또는 IPMVC가 이용 가능하지 않다는 결정에 응답하여 (304의 "아니오"), 비디오 코더는 공간적 이웃 PU들이 가용 모션 벡터들을 갖는지의 여부를 결정하기 위해 공간적 이웃 PU들을 체크할 수도 있다 (308). 몇몇 예들에서, 공간적 이웃 PU들은 도 2에서 A₀, A₁, B₀, B₁, 및 B₂로서 나타내어진 로케이션들을 커버한다. 설명의 편의를 위해, 본 개시물은 로케이션들 (A₀, A₁, B₀, B₁, 및 B₂) 을 커버하는 PU들의 모션 정보를 각각 A₀, A₁, B₀, B₁, 및 B₂로서 지칭할 수도 있다.

[0380] 도 11의 예에서, 비디오 코더는 A₁이 IPMVC와 일치하는지의 여부를 결정할 수도 있다 (310). A₁ 이 IPMVC와 일치하지 않는다는 결정에 응답하여 (310의 "아니오"), 비디오 코더는 A₁을 병합 후보 리스트에 삽입할 수도 있다 (312). 그렇지 않으면, A₁이 IPMVC와 일치한다는 결정에 응답하여 (310의 "예") 또는 A₁을 병합 후보 리스트에 삽입한 후, 비디오 코더는 B₁이 A₁ 또는 IPMVC와 일치하는지의 여부를 결정할 수도 있다 (314). B₁이 A₁ 또는 IPMVC와 일치하지 않는다는 결정에 응답하여 (314의 "아니오"), 비디오 코더는 B₁을 병합 후보 리스트에 삽입할 수도 있다 (316). 한편, B₁이 A₁ 또는 IPMVC와 일치한다는 결정에 응답하여 (314의 "예") 또는 B₁을 병합 후보 리스트에 삽입한 후, 비디오 코더는 B₀가 이용 가능한지의 여부를 결정할 수도 있다 (318). B₀가 이용 가능하다는 결정에 응답하여 (318의 "예"), 비디오 코더는 B₀를 병합 후보 리스트에 삽입할 수도 있다 (320). B₀가 이용 가능하지 않다면 (318의 "아니오") 또는 B₀를 병합 후보 리스트에 삽입한 후, 비디오 코더는 IDMVC가 이용 가능하고 A₁ 또는 B₁과 일치하지 않는지를 결정할 수도 있다 (332). IDMVC는 현재 PU에 대한 디스패리티 벡터를 특정할 수도 있다. IDMVC가 뷰 간 참조 화상의 경계들 외부에 있는 로케이션을 나타낸다면 IDMVC는 이용 불가능할 수도 있다. IDMVC가 이용 가능하고 A₁ 또는 B₁과 일치하지 않는다는 결정에 응답하여 (332의 "예"), 비디오 코더는 IDMVC를 병합 후보 리스트에 삽입할 수도 있다 (334). IDMVC가 이용 가능하지 않거나 또는 IDMVC가 A₁ 또는 B₁과 일치한다면 (332의 "아니오") 또는 IDMVC를 병합 후보 리스트에 삽입한 후, 비디오 코더는 도 12에 도시된 동작 (300) 의 부분 ("A"로 표시됨) 을 수행할 수도 있다.

[0381] 도 12는 본 개시물의 하나 이상의 기법들에 따른, 현재 블록에 대한 병합 후보 리스트를 구축하는 도 11의 예의 동작 (300) 의 제 2 부분을 도시하는 흐름도이다. 위에서 나타난 바와 같이, 비디오 코더는 IDMVC가 이용 가능하지 않거나 또는 IDMVC가 A₁ 또는 B₁과 일치한다면 (332의 "아니오") 또는 IDMVC를 병합 후보 리스트에 삽입한 후 도 12에 도시된 동작 (300) 의 부분을 수행할 수도 있다. 그런고로, IDMVC가 이용 가능하지 않거나 또는 IDMVC가 A₁ 또는 B₁과 일치한다면 (332의 "아니오") 또는 IDMVC를 병합 후보 리스트에 삽입한 후, 비디오 코더는 BVSP가 가능해지는지의 여부를 결정할 수도 있다 (336). BVSP가 가능화된다면 (336의 "예"), 비디오

코더는 BVSP 후보를 병합 후보 리스트에 삽입할 수도 있다 (338). BVSP가 가능화되지 않는다면 (336의 "아니오") 또는 BVSP 후보를 병합 후보 리스트에 삽입한 후, 비디오 코더는 A_0 가 이용 가능한지의 여부를 결정할 수도 있다 (340). A_0 가 이용 가능하다면 (340의 "예"), 비디오 코더는 A_0 를 병합 후보 리스트에 삽입할 수도 있다 (342). 그렇지 않고, A_0 가 이용 가능하지 않다면 (340의 "아니오") 또는 A_0 를 병합 후보 리스트에 삽입한 후, 비디오 코더는 B_2 가 이용 가능한지의 여부를 결정할 수도 있다 (344). B_2 가 이용 가능하다면 (344의 "예"), 비디오 코더는 B_2 를 병합 후보 리스트에 삽입할 수도 있다 (346).

[0382] B_2 가 이용 가능하지 않다면 (344의 "아니오") 또는 B_2 를 병합 후보 리스트에 삽입한 후, 비디오 코더는 뷰 간 모션 예측이 적용되는지의 여부를 결정할 수도 있다 (348). 다르게 말하면, 비디오 코더는 현재 블록이 뷰 간 모션 예측을 사용하여 코딩될 수도 있는지의 여부를 결정할 수도 있다. 뷰 간 모션 예측이 적용된다는 결정에 응답하여 (348의 "예"), 비디오 코더는 시프트된 후보를 결정할 수도 있다 (350). 다르게 말하면, 비디오 코더는, 본 개시물의 다른 곳에서 설명되는 바와 같이, DSMV 후보를 결정할 수도 있다. 시프트된 후보를 결정한 후, 비디오 코더는 시프트된 후보가 이용 가능한지의 여부를 결정할 수도 있다 (352). 시프트된 후보가 이용 가능하다면 (352의 "예"), 비디오 코더는 병합 후보 리스트에 시프트된 후보를 포함시킬 수도 있다 (354). 뷰 간 모션 예측이 적용되지 않는다면 (348의 "아니오"), 시프트된 후보가 이용 가능하지 않다면 (352의 "아니오"), 또는 병합 후보 리스트에 시프트된 후보를 삽입한 후, 비디오 코더는 시간적 병합 후보를 병합 후보 리스트에 포함시킬 수도 있다 (356).

[0383] 더욱이, 비디오 코더는 조합된 양방향-예측 병합 후보들에 대한 도출 프로세스를 수행할 수도 있다 (358). 본 개시물의 하나 이상의 기법들에 따른 조합된 양방향-예측 병합 후보들에 대한 일 예의 도출 프로세스가 도 13에 관해 아래에서 설명된다. 덧붙여서, 비디오 코더는 제로 모션 벡터 후보들에 대한 도출 프로세스를 수행할 수도 있다 (360). 제로 모션 벡터 후보들에 대한 일 예의 도출 프로세스가 HEVC WD 10의 섹션 8.5.3.2.4에서 설명된다.

[0384] 도 13은 본 개시물의 하나 이상의 기법들에 따른, 조합된 양방향-예측 병합 후보들에 대한 일 예의 도출 프로세스를 도시하는 흐름도이다. 도 13의 도출 프로세스는 임의의 BVSP 플래그들을 체크하는 일 없이 수행될 수도 있다. 예를 들면, 도 13의 도출 프로세스는 3D-HEVC 초안 텍스트 1의 섹션 H.8.5.3.2.1에서 행해지는 바와 같이, mergeCandIsVspFlag를 조합된 양방향-예측 병합 후보들에 대한 도출 프로세스에의 입력으로서 제공하는 일 없이 수행될 수도 있다. 더욱이, 도 13의 도출 프로세스는, 3D-HEVC 초안 텍스트 1의 섹션 H.8.5.3.2.3에서 행해지는 바와 같이, 조합된 양방향-예측 병합 후보들에 대한 도출 프로세스에서 mergeCandIsVspFlag를 사용하는 일 없이 수행될 수도 있다.

[0385] 도 13의 예에서, 비디오 코더 (예컨대, 비디오 인코더 (20) 또는 비디오 디코더 (30))가 현재 슬라이스 (즉, 비디오 코더가 현재 코딩하고 있는 슬라이스)가 B 슬라이스인지의 여부를 결정할 수도 있다 (400). 현재 슬라이스가 B 슬라이스가 아니면 (400의 "아니오"), 비디오 코더는 조합된 양방향-예측 병합 후보들에 대한 도출 프로세스를 종료할 수도 있다. 그러나, 현재 슬라이스가 B 슬라이스라는 결정에 응답하여 (400의 "예"), 비디오 코더는 병합 후보들의 리스트 (즉, 병합 후보 리스트)에서의 병합 후보들의 수가 5 미만인지의 여부를 결정할 수도 있다 (402). 병합 후보들의 리스트에서의 병합 후보들의 수가 5 미만이면, 비디오 코더는 조합된 양방향-예측 병합 후보들에 대한 도출 프로세스를 종료할 수도 있다.

[0386] 한편, 병합 후보들의 리스트에서의 병합 후보들의 수가 5 미만이라는 결정에 응답하여 (402의 "예"), 비디오 코더는 조합 인덱스 (예컨대, combIdx)의 값을 0으로 설정할 수도 있다 (404). 비디오 코더는 그 다음에 조합 인덱스의 현재 값에 대응하는 모션 벡터들이 이용 가능한지의 여부를 결정할 수도 있다 (406).

[0387] 조합 인덱스의 현재 값에 대응하는 모션 벡터들이 이용 가능하다는 결정에 응답하여 (406의 "예"), 비디오 코더는 병합 후보들의 리스트에 조합 인덱스의 현재 값에 연관된 조합된 양방향-예측 병합 후보를 포함시킬 수도 있다 (408). 조합 인덱스의 현재 값에 연관된 조합된 양방향-예측 병합 후보는 표 1에 따라 RefPicList0 모션 정보 및 RefPicList1 모션 정보를 특정할 수도 있다.

[0388] 더욱이, 비디오 코더는 조합 인덱스의 현재 값이 $(\text{numOrigMergeCand} * (\text{numOrigMergeCand} - 1))$ 과 동일할지의 여부를 결정할 수도 있는데, 여기서 numOrigMergeCand는 도 13의 도출 프로세스를 호출하기 전에 병합 후보들의 리스트에서의 병합 후보들의 수를 나타낸다 (410). 조합 인덱스의 현재 값이 $(\text{numOrigMergeCand} * (\text{numOrigMergeCand} - 1))$ 과 동일하면 (410의 "예"), 비디오 코더는 조합된 양방향-예측 병합 후보들에 대한 도

출 프로세스를 종료할 수도 있다. 한편, 조합 인덱스의 현재 값이 ($\text{numOrigMergeCand} * (\text{numOrigMergeCand} - 1)$) 과 동일하지 않다면 (410의 "아니오"), 비디오 코더는 병합 후보들의 리스트에서의 병합 후보들의 총 수가 MaxNumMergeCand 와 동일한지의 여부를 결정할 수도 있다 (412). 본 개시물의 다른 곳에서 나타낸 바와 같이, MaxNumMergeCand 는 병합 후보들의 리스트에서의 병합 후보들의 최대 수를 나타낸다. 병합 후보들의 리스트에서의 병합 후보들의 총 수가 MaxNumMergeCand 와 동일하면 (412의 "예"), 비디오 코더는 조합된 양방향-예측 병합 후보들에 대한 도출 프로세스를 종료할 수도 있다.

[0389] 그러나, 병합 후보들의 리스트에서의 병합 후보들의 총 수가 MaxNumMergeCand 와 동일하지 않다는 결정에 응답하여 (412의 "아니오") 또는 조합 인덱스의 현재 값에 대응하는 모션 벡터들이 이용 가능하지 않다는 결정에 응답하여 (406의 "아니오"), 비디오 코더는 조합 인덱스의 현재 값을 증가시킬 수도 있다 (414). 비디오 코더는 그 다음에 조합 인덱스의 증가된 값에 관해 액션들 (406-414) 을 수행할 수도 있다. 이런 식으로, 비디오 코더는 조합 인덱스의 현재 값이 ($\text{numOrigMergeCand} * (\text{numOrigMergeCand} - 1)$) 과 동일하거나 또는 병합 리스트에서의 총 후보들의 수 (새로 생성된 조합된 양방향-예측 병합 후보들을 포함함) 가 MaxNumMergeCand 와 동일하기까지 조합된 양방향-예측 병합 후보들을 도출하는 것을 계속할 수도 있다.

[0390] 도 14a는 본 개시물의 하나 이상의 기법들에 따른, 비디오 블록을 인코딩하는 비디오 인코더 (20) 의 일 예의 동작을 도시하는 흐름도이다. 도 14a의 예에서, 비디오 인코더 (20) 는 병합 후보들의 리스트를 생성할 수도 있다 (450). 다르게 말하면, 비디오 인코더 (20) 는 병합 후보 리스트를 생성할 수도 있다. 도 14a의 예에서, 비디오 인코더 (20) 는 리스트에서의 병합 후보들의 수가 5 미만인지의 여부를 결정할 수도 있다 (452). 몇몇 예들에서, 비디오 인코더 (20) 는, 이 단계에서, 리스트에서의 병합 후보들의 수가 5 미만이고 리스트에서의 병합 후보들의 최대 수가 5 보다 큰지 (예컨대, 6과 동일한지) 를 결정할 수도 있다. 리스트에서의 병합 후보들의 수가 5 미만이라는 결정에 응답하여 (452의 "예"), 비디오 인코더 (20) 는 하나 이상의 조합된 양방향-예측 병합 후보들을 도출할 수도 있고 (454) 병합 후보들의 리스트에 하나 이상의 조합된 양방향-예측 병합 후보들을 포함시킬 수도 있다 (456). 하나 이상의 조합된 양방향-예측 병합 후보들의 각각의 개별 조합된 양방향-예측 병합 후보는 리스트에서의 병합 후보들의 개별 쌍에 이미 대응할 수도 있다. 각각의 조합된 양방향-예측 병합 후보는 개별 쌍의 제 1 병합 후보의 모션 벡터 및 개별 쌍의 제 2 병합 후보의 모션 벡터의 조합일 수도 있다. 제 1 병합 후보의 모션 벡터 및 제 2 병합 후보의 모션 벡터는 상이한 참조 화상 리스트들 (예컨대, 리스트 1 및 리스트 2) 에서의 화상들을 가리킨다. 한편, 몇몇 예들에서, 리스트에서의 병합 후보들의 수가 5 미만이면 (452의 "아니오"), 비디오 인코더 (20) 는 리스트에 임의의 조합된 양방향-예측 병합 후보들을 포함시키지 않는다 (458).

[0391] 몇몇 예들에서, 비디오 인코더 (20) 는 IPMVC를, 이용 가능하다면, 병합 후보들의 리스트에 삽입한 후, 공간적 병합 후보들에 대한 도출 프로세스를 수행한 후, 및 시간적 병합 후보에 대한 도출 프로세스를 수행한 후, 하나 이상의 조합된 양방향-예측 병합 후보들을 도출할 수도 있다. 공간적 병합 후보들에 대한 도출 프로세스는 네 개까지의 공간적 모션 벡터 후보들을 병합 후보들의 리스트에 도출 및 삽입할 수도 있다. 시간적 병합 후보에 대한 도출 프로세스는 시간적 모션 벡터 예측자 (TMVP) 후보를 이용 가능하다면, 병합 후보들의 리스트에 추가할 수도 있다.

[0392] 더욱이, 도 14a의 예에서, 비디오 인코더 (20) 는 병합 후보들의 리스트에서 후보를 선택할 수도 있다 (460). 몇몇 예들에서, 비디오 인코더 (20) 는 비트스트림에서 시그널링된 선택스 엘리먼트에 의해 표시된 값에 기초하여 선택된 후보를 결정할 수도 있다. 덧붙여서, 비디오 인코더 (20) 는 선택된 병합 후보의 병합 후보들의 리스트에서의 위치를 시그널링할 수도 있다 (462). 비디오 인코더 (20) 는 선택된 후보에 기초하여 비디오 블록을 인코딩할 수도 있다 (464). 비디오 인코더 (20) 는 본 개시물의 다른 곳에서 제공된 예들 중 하나 이상에 따라 비디오 블록을 인코딩할 수도 있다.

[0393] 도 14b는 본 개시물의 하나 이상의 기법들에 따른, 비디오 블록을 디코딩하는 비디오 디코더 (30) 의 일 예의 동작을 도시하는 흐름도이다. 도 14b의 예에서, 비디오 디코더 (30) 는 병합 후보들의 리스트를 생성할 수도 있다 (480). 도 14b의 예에서, 비디오 디코더 (30) 는 리스트에서의 병합 후보들의 수가 5 미만인지의 여부를 결정할 수도 있다 (482). 몇몇 예들에서, 비디오 디코더 (30) 는, 이 단계에서, 리스트에서의 병합 후보들의 수가 5 미만이고 리스트에서의 병합 후보들의 최대 수가 5 보다 큰지 (예컨대, 6과 동일한지) 를 결정할 수도 있다. 리스트에서의 병합 후보들의 수가 5 미만이라는 결정에 응답하여 (452의 "예"), 비디오 디코더 (30) 는 하나 이상의 조합된 양방향-예측 병합 후보들을 도출할 수도 있다 (484). 하나 이상의 조합된 양방향-예측 병합 후보들의 각각의 개별 조합된 양방향-예측 병합 후보는 리스트에서의 병합 후보들의 개별 쌍에 이미 대응할 수도 있다. 각각의 조합된 양방향-예측 병합 후보는 개별 쌍의 제 1 병합 후보의 모션 벡터

및 개별 쌍의 제 2 병합 후보의 모션 벡터의 조합일 수도 있다. 제 1 병합 후보의 모션 벡터 및 제 2 병합 후보의 모션 벡터는 상이한 참조 화상 리스트들 (예컨대, 리스트 1 및 리스트 2) 에서의 화상들을 가리킨다.

비디오 디코더 (30) 는 그 리스트에 하나 이상의 조합된 양방향-예측 병합 후보들을 포함시킬 수도 있다 (486). 한편, 몇몇 예들에서, 리스트에서의 병합 후보들의 수가 5 미만이 아니면 (482의 "아니오"), 비디오 디코더 (30) 는 리스트에 임의의 조합된 양방향-예측 병합 후보들을 포함시키지 않는다 (488).

[0394] 몇몇 예들에서, 비디오 디코더 (30) 는 IPMVC를, 이용 가능하다면, 병합 후보들의 리스트에 삽입한 후, 공간적 병합 후보들에 대한 도출 프로세스를 수행한 후, 및 시간적 병합 후보에 대한 도출 프로세스를 수행한 후, 하나 이상의 조합된 양방향-예측 병합 후보들을 도출할 수도 있다. 공간적 병합 후보들에 대한 도출 프로세스는 네 개까지의 공간적 모션 벡터 후보들을 병합 후보들의 리스트에 도출 및 삽입할 수도 있다. 시간적 병합 후보에 대한 도출 프로세스는 시간적 모션 벡터 예측자 (TMVP) 후보를 이용 가능하다면, 병합 후보들의 리스트에 추가할 수도 있다.

[0395] 더욱이, 도 14b의 예에서, 비디오 디코더 (30) 는 병합 후보들의 리스트에서 선택된 후보를 결정할 수도 있다 (490). 몇몇 예들에서, 비디오 디코더 (30) 는 비트스트림에서 시그널링된 선택스 엘리먼트에 의해 표시된 값에 기초하여 선택된 후보를 결정할 수도 있다. 예를 들면, 비디오 디코더 (30) 는, 비트스트림으로부터, 선택스 병합 후보들의 리스트에서의 선택된 병합 후보를 나타내는 엘리먼트를 획득할 수도 있다. 비디오 디코더 (30) 는 선택된 후보에 기초하여 비디오 블록을 디코딩할 수도 있다 (492). 예를 들면, 비디오 디코더 (30) 는 현재 CU의 예측 샘플들을 생성하기 위해 선택된 후보의 모션 정보를 사용할 수도 있다. 비디오 디코더 (30) 는 본 개시물의 다른 곳에서 제공되는 예들 중 하나 이상에 따라 비디오 블록 (예컨대, CU, PU 등) 을 디코딩할 수도 있다.

[0396] 다음의 단락들은 본 개시물의 추가적인 예들을 제공한다.

[0397] 실시예 1. 비디오 데이터를 코딩하는 방법에 있어서, 3차원 비디오 데이터에 연관되지 않은 비디오 블록을 코딩하기 위한 제 1 프로세스에 따라 병합 후보들의 제 1 리스트를 생성하는 단계로서, 제 1 리스트는 하나 이상의 양방향-예측 병합 후보들을 포함하는, 상기 병합 후보들의 제 1 리스트를 생성하는 단계; 및 3차원 비디오 데이터에 연관되는 비디오 블록을 코딩하기 위한 제 2 프로세스에 따라 병합 후보들의 제 2 리스트를 생성하는 단계로서, 제 2 리스트는 하나 이상의 양방향-예측 병합 후보들을 포함하며, 제 1 프로세스 및 제 2 프로세스는 동일한, 상기 병합 후보들의 제 2 리스트를 생성하는 단계를 포함하는, 비디오 데이터 코딩 방법.

[0398] 실시예 2. 실시예 1에 있어서, 제 1 리스트를 생성하는 단계와 제 2 리스트를 생성하는 단계는, 다음의 조건, 즉, 가용 병합 후보들의 수가 5 미만이란 조건이 충족되는 경우에만 발생하는, 비디오 데이터 코딩 방법.

[0399] 실시예 3. 실시예 1 또는 2에 있어서, 임의의 병합 리스트를 생성하기 위한 도출 프로세스를 호출하기 전에 병합 MVP 후보들의 최대 수를 정의하는 단계를 더 포함하는, 비디오 데이터 코딩 방법.

[0400] 실시예 4. 실시예 4에 있어서, 병합 MVP 후보들의 최대 수는 실질적으로, $MaxNumMergeCand = 5 - five_minus_max_num_merge_cand$ 와 같이 정의되며, 이 프로세스가 호출된 후, $MaxNumMergeCand$ 는 $MaxNumMergeCand = 5 - five_minus_max_num_merge_cand + iv_mv_pred_flag[nuh_layer_id]$ 로 다시 설정되는, 비디오 데이터 코딩 방법.

[0401] 실시예 5. 3차원 (3D) 비디오에 연관된 데이터를 코딩하는 방법에 있어서, 3D 비디오에 연관된 비디오 블록을 코딩하기 위한 병합 후보들의 리스트를 생성하는 단계를 포함하며, 그 리스트는 하나 이상의 양방향-예측 병합 후보들을 포함하고 병합 후보들의 최대 수가 6과 동일하고 조합된 양방향-예측 병합 후보들의 도출 프로세스가 호출되기 전에 정의된 5 개의 후보들이 있는 경우, 제로 후보가 그 리스트에 생성 및 포함되고, 제로 후보는 참조 인덱스 및 모션 벡터 성분들을 0으로서 정의하는, 3D 비디오에 연관된 데이터 코딩 방법.

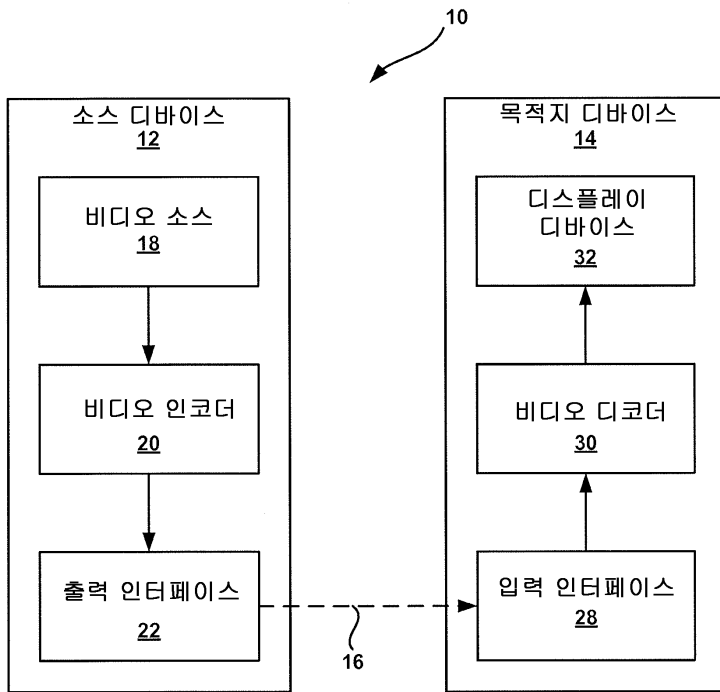
[0402] 실시예 6. 3차원 (3D) 비디오에 연관된 데이터를 코딩하는 방법에 있어서, 3D 비디오에 연관된 비디오 블록을 코딩하기 위한 병합 후보들의 리스트를 생성하는 단계를 포함하며, 그 리스트는 하나 이상의 양방향-예측 병합 후보들을 포함하고, 그 리스트를 생성하기 전에, 병합 후보들의 최대 수는 5로 설정되며, 네 개의 후보들은 병합 리스트 도출 프로세스에 입력되고, 하나의 후보는 병합 리스트 도출 프로세스 동안 새로 생성되는, 3D 비디오에 연관된 데이터 코딩 방법.

[0403] 실시예 7. 실시예 6에 있어서, 새로 생성된 후보는 리스트에서 다섯 번째로서 순서 매겨지는, 3D 비디오에 연관된 데이터 코딩 방법.

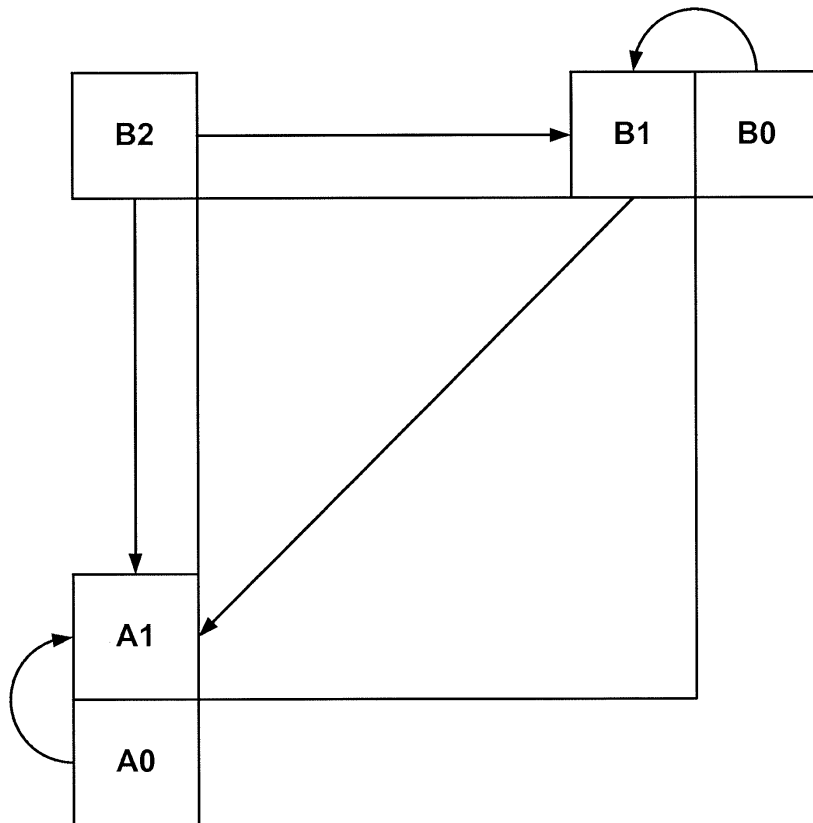
- [0404] 실시예 8. 실시예 6에 있어서, 병합 리스트 도출 프로세스가 비-제로의 새로 생성된 후보를 생성할 수 없다면, 병합 리스트 도출 프로세스는 제로 값 후보를 새로 생성된 후보로서 생성하는, 3D 비디오에 연관된 데이터 코딩 방법.
- [0405] 하나 이상의 예들에서, 본원에서 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 그것들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현된다면, 그 기능들은 하나 이상의 명령들 또는 코드로서 컴퓨터 판독가능 매체 상에 저장되거나 또는 그것을 통해 송신될 수도 있고 하드웨어 기반 프로세싱 유닛에 의해 실행될 수도 있다. 컴퓨터 판독가능 매체들은, 데이터 저장 매체들과 같은 유형의 (tangible) 매체에 대응하는 컴퓨터 판독가능 저장 매체들, 또는 예컨대 통신 프로토콜에 따라 한 장소에서 다른 장소로 컴퓨터 프로그램의 전달을 용이하게 하는 임의의 매체를 포함하는 통신 매체들을 포함할 수도 있다. 이런 방식으로, 컴퓨터 판독가능 매체들은 일반적으로 (1) 비일시적 (non-transitory) 인 유형의 컴퓨터 판독가능 저장 매체들 또는 (2) 신호 또는 반송파와 같은 통신 매체에 해당할 수도 있다. 데이터 저장 매체들은 본 개시물에서 설명된 기법들의 구현을 위한 명령들, 코드 및/또는 데이터 구조들을 추출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 이용가능 매체들일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터 판독가능 매체를 포함할 수도 있다.
- [0406] 비제한적인 예로, 이러한 컴퓨터 판독가능 저장 매체들은 RAM, ROM, EEPROM, CD-ROM 또는 다른 광 디스크 스토리지, 자기 디스크 스토리지, 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 소망의 프로그램 코드를 컴퓨터에 의해 액세스될 수 있는 명령들 또는 데이터 구조들의 형태로 저장하는데 사용될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 접속이 컴퓨터 판독가능 매체로 적절히 칭해진다. 예를 들어, 명령들이 웹사이트, 서버, 또는 다른 원격 자원으로부터 동축 케이블, 광섬유 케이블, 연선 (twisted pair), 디지털 가입자 회선 (DSL), 또는 무선 기술들 이를테면 적외선, 라디오, 및/또는 마이크로파를 이용하여 송신된다면, 동축 케이블, 광섬유 케이블, 연선, DSL, 또는 적외선, 라디오, 및 마이크로파와 같은 무선 기술은 매체의 정의에 포함된다. 그러나, 컴퓨터 판독가능 저장 매체들 및 데이터 저장 매체들은 커넥션들, 반송파들, 신호들, 또는 다른 일시적인 매체들을 포함하지 않지만, 대신 비일시적, 유형의 저장 매체들을 지향하고 있음이 이해되어야 한다. 디스크 (disk 및 disc) 는 본원에서 사용되는 바와 같이, 콤팩트 디스크 (compact disc, CD), 레이저 디스크, 광 디스크, 디지털 다용도 디스크 (DVD), 플로피 디스크 (floppy disk) 및 블루레이 디스크를 포함하는데, disk들은 보통 데이터를 자기적으로 재생하지만, disc들은 레이저들로서 광적으로 데이터를 재생한다. 상기한 것들의 조합들은 또한 컴퓨터 판독가능 매체들의 범위 내에 포함되어야 한다.
- [0407] 명령들은 하나 이상의 프로세서들, 이를테면 하나 이상의 디지털 신호 프로세서들 (DSP들), 범용 마이크로프로세서들, 주문형 집적회로들 (ASIC들), 필드 프로그램가능 게이트 어레이들 (FPGA들), 또는 다른 동등한 집적 또는 개별 로직 회로에 의해 실행될 수도 있다. 따라서, 본원에서 사용되는 바와 같은 "프로세서"라는 용어는 앞서의 구조 또는 본원에서 설명된 기법들의 구현에 적합한 임의의 다른 구조 중 임의의 것을 나타낼 수도 있다. 덧붙여서, 일부 양태들에서, 본원에서 설명된 기능성은 인코딩 및 디코딩을 위해 구성되는, 또는 결합형 코덱 (codec) 으로 통합되는 전용 하드웨어 및/또는 소프트웨어 모듈들 내에 제공될 수도 있다. 또한, 본 기법들은 하나 이상의 회로들 또는 로직 엘리먼트들 내에 완전히 구현될 수 있다.
- [0408] 본 개시물의 기법들은 무선 핸드셋, 집적회로 (IC) 또는 IC들의 세트 (예컨대, 칩 셋) 를 포함하는 매우 다양한 디바이스들 또는 장치들로 구현될 수도 있다. 다양한 컴포넌트들, 모듈들, 또는 유닛들은 개시된 기법들을 수행하도록 구성된 디바이스들의 기능적 양태들을 강조하기 위해 본 개시물에서 설명되지만, 상이한 하드웨어 유닛들에 의한 실현을 반드시 요구하지는 않는다. 대신에, 위에서 설명된 바와 같이, 다양한 유닛들은 코덱 하드웨어 유닛에 결합되거나 또는 적합한 소프트웨어 및/또는 펌웨어와 함께, 위에서 설명된 바와 같은 하나 이상의 프로세서들을 포함하는, 상호운용적 하드웨어 유닛들의 컬렉션에 의해 제공될 수도 있다.
- [0409] 다양한 예들이 설명되어 있다. 이들 및 다른 예들은 다음의 청구항들의 범위 내에 있다.

도면

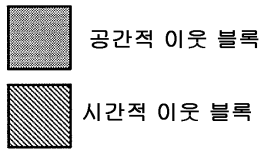
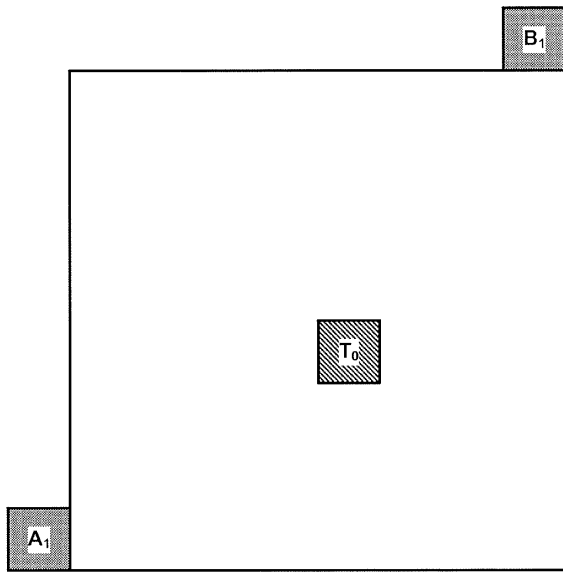
도면1



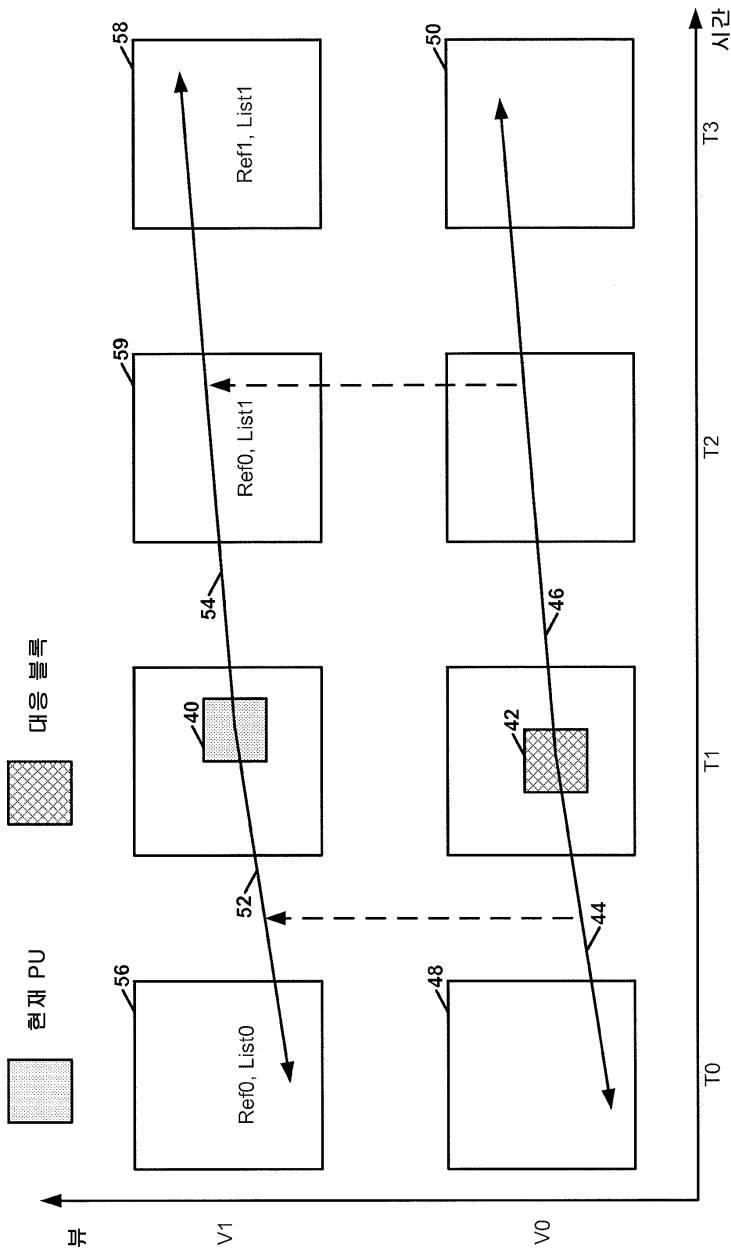
도면2



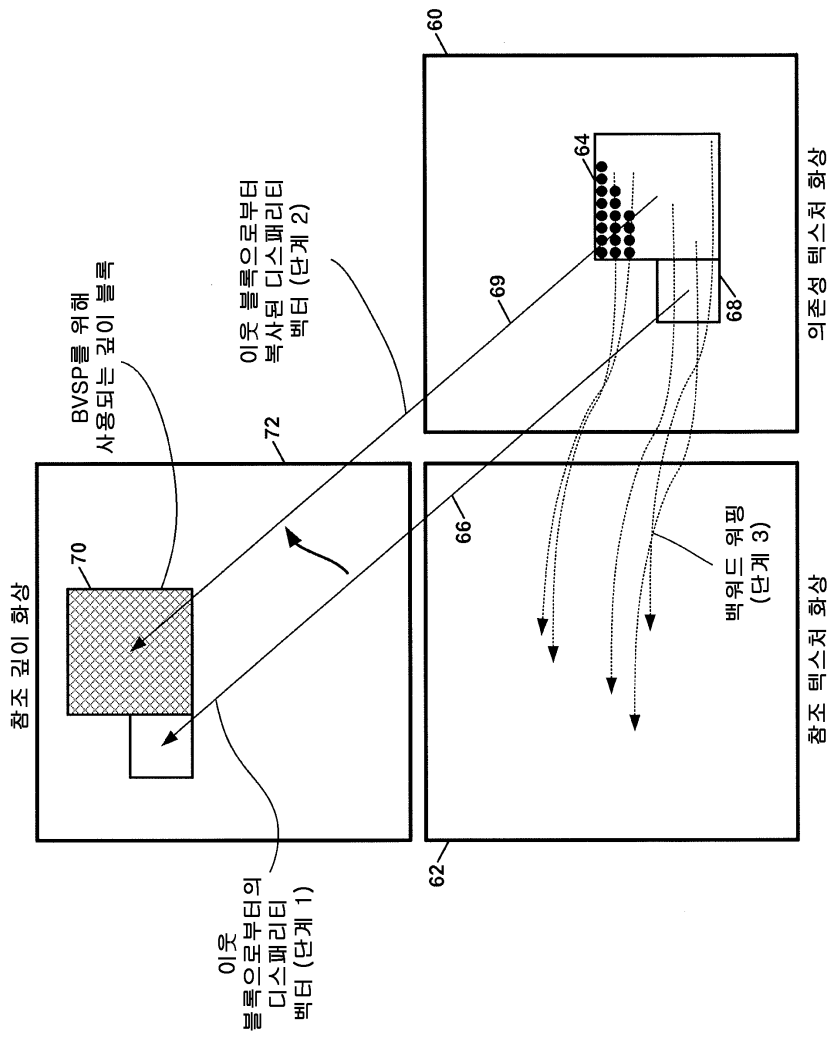
도면3



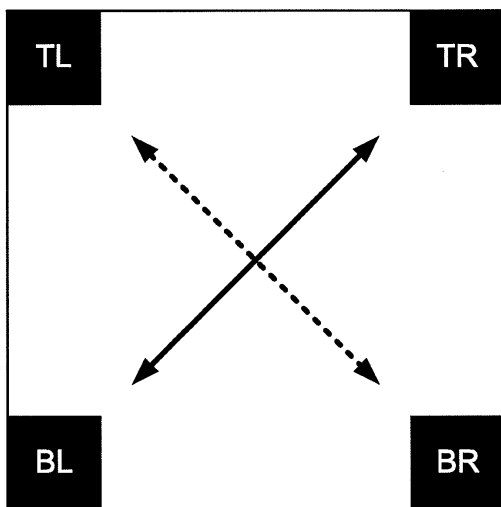
도면4



도면5



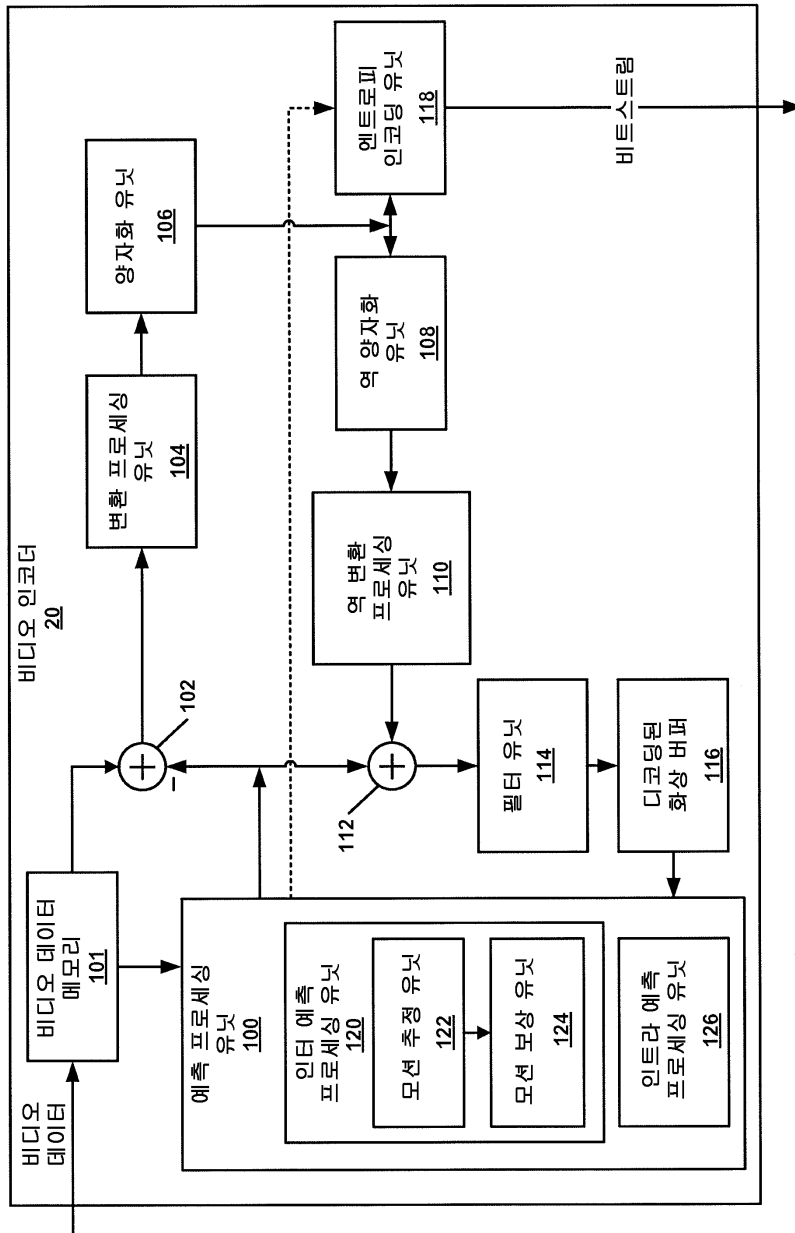
도면6



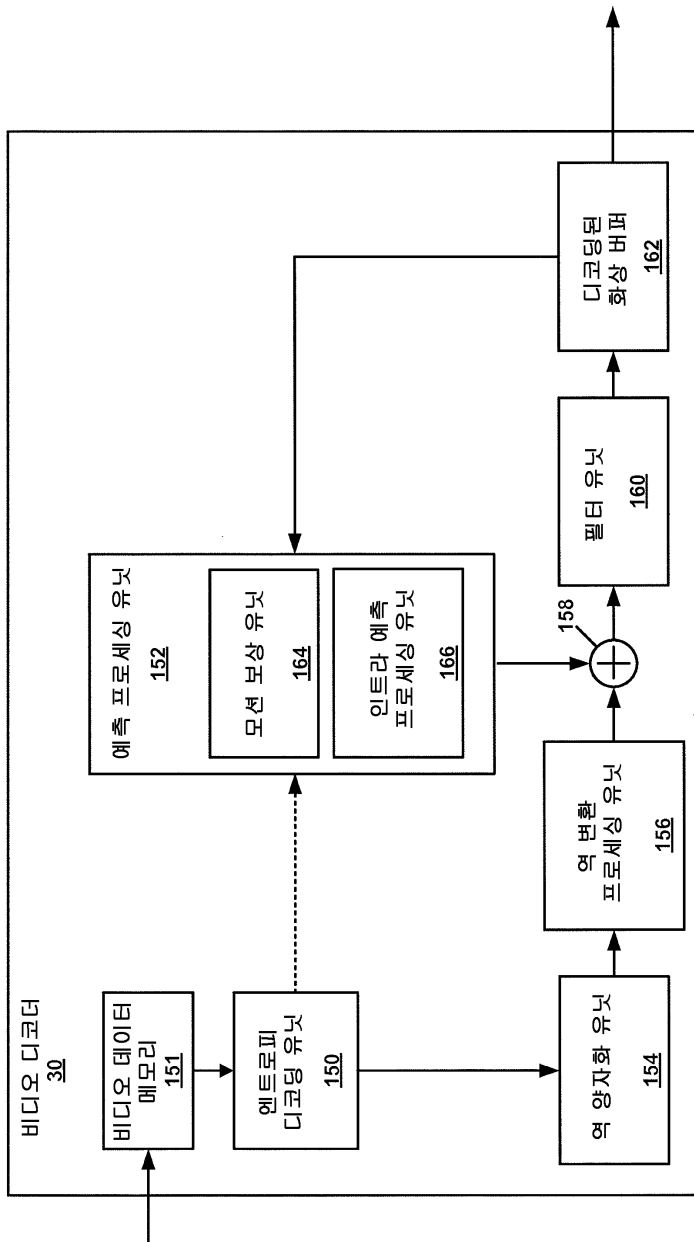
도면7

comIdx	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
I0CandIdx	0	1	0	2	1	2	0	3	1	3	2	3	0	4	1	4	2	4	3	4
I1CandIdx	1	0	2	0	2	1	3	0	3	1	3	2	4	0	4	1	4	2	4	3

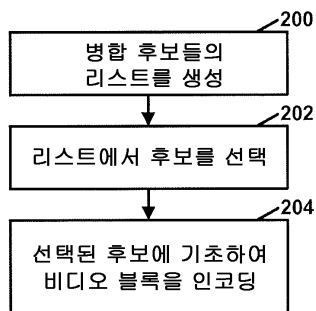
도면8



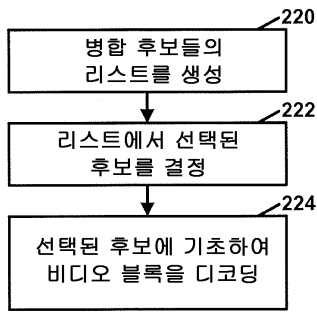
도면9



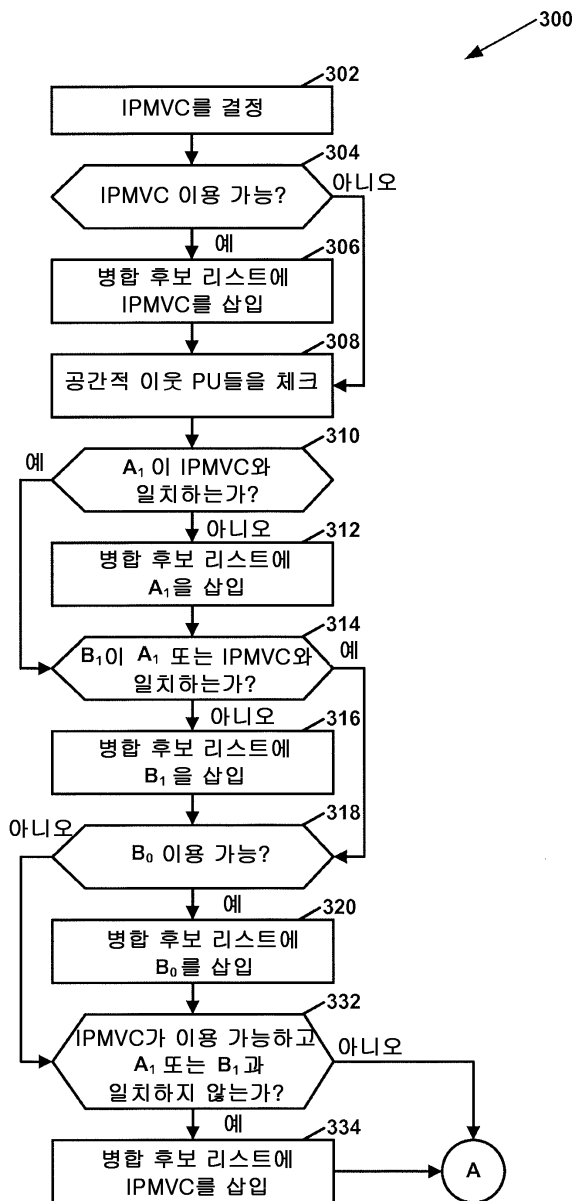
도면10a



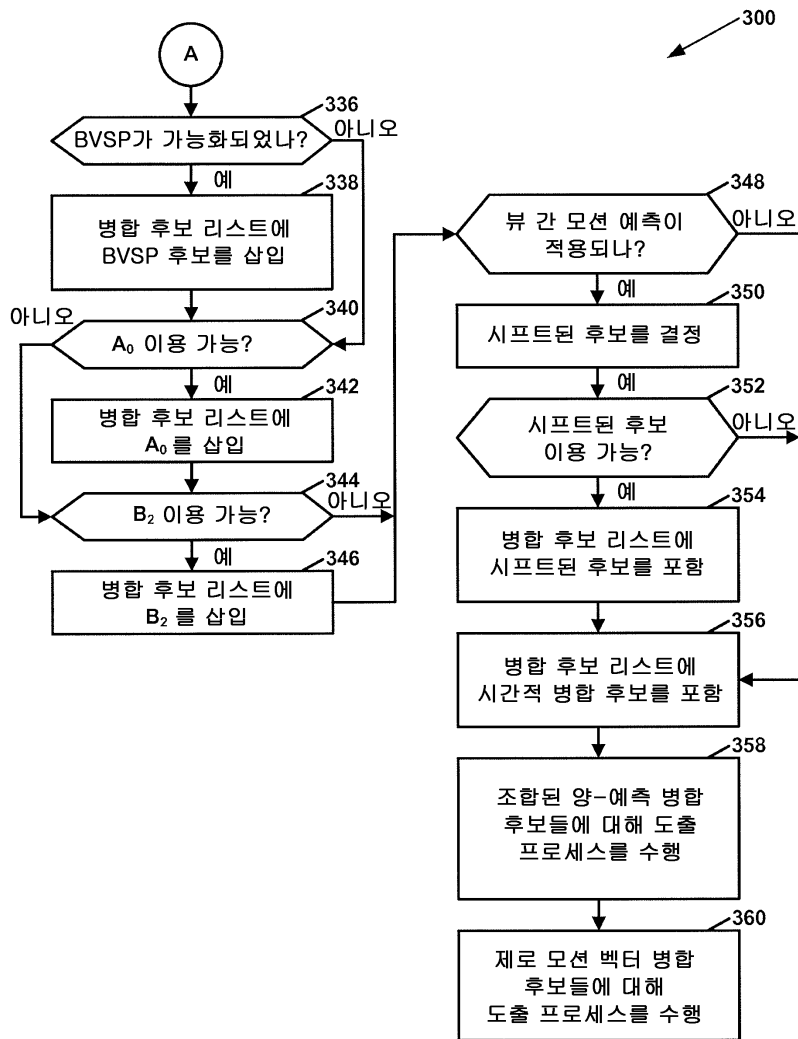
도면10b



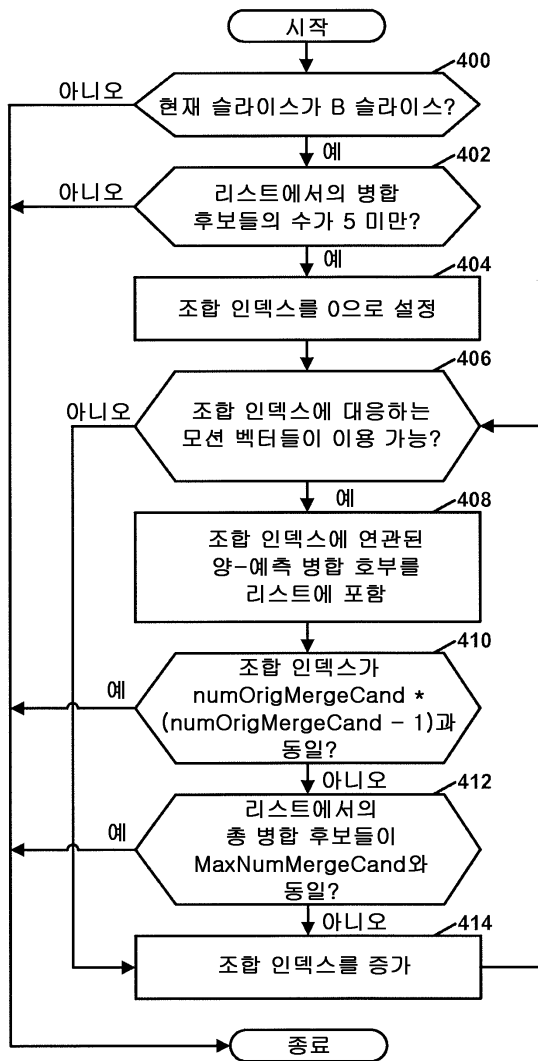
도면11



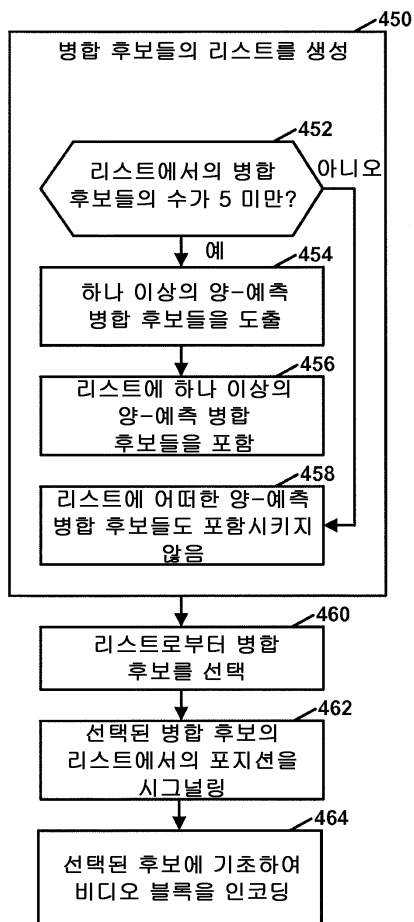
도면12



도면13



도면14a



도면14b

