



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2024년07월04일
(11) 등록번호 10-2680903
(24) 등록일자 2024년06월28일

(51) 국제특허분류(Int. Cl.)
HO4N 19/52 (2014.01) *HO4N 19/105* (2014.01)
HO4N 19/176 (2014.01) *HO4N 19/61* (2014.01)
HO4N 19/70 (2014.01) *HO4N 19/96* (2014.01)

(52) CPC특허분류
HO4N 19/52 (2015.01)
HO4N 19/105 (2015.01)

(21) 출원번호 10-2022-7042182(분할)

(22) 출원일자(국제) 2019년07월01일
심사청구일자 2022년11월30일

(85) 번역문제출일자 2022년11월30일

(65) 공개번호 10-2023-0003229

(43) 공개일자 2023년01월05일

(62) 원출원 특허 10-2020-7038007
원출원일자(국제) 2019년07월01일
심사청구일자 2021년11월09일

(86) 국제출원번호 PCT/IB2019/055582

(87) 국제공개번호 WO 2020/003275
국제공개일자 2020년01월02일

(30) 우선권주장
PCT/CN2018/093663 2018년06월29일 중국(CN)
(뒷면에 계속)

(56) 선행기술조사문헌

(56) 선행기술조사문현

US20110194609 A1*

US20130336406 A1*

C. Rosewarne et al, High Efficiency Video Coding (HEVC) Test Model 16 (HM 16) Improved Encoder Description Update 9, JCT-VC of ITU-T and ISO/IEC, 28th meeting,
JCTVC-AB1002(2017.10.15.) 1부 *

WO2011095260 A1

*는 심사관에 의하여 이용될 문항

전체 청구항 수 : 총 16 항

심사관 : 황수진

(54) 발명의 명칭 HMVP 후보를 병합/AMVP에 추가할 때의 부분/풀 프루닝

(57) 요약

비디오 처리 방법이 제공되며, 이 방법은 테이블들의 세트를 유지하는 단계- 각각의 테이블은 모션 후보들을 포함하고, 각각의 모션 후보는 대응하는 모션 정보와 연관됨 -; 모션 후보들에 대한 프루닝 동작을 사용하여 하나 이상의 테이블 내의 모션 후보들에 기초하여 모션 후보 리스트를 업데이트하는 단계; 및 구성된 모션 후보 리스트 (뒷면에 계속)

대 표 도



트를 사용하여 제1 비디오 블록과 제1 비디오 블록을 포함하는 비디오의 비트스트림 표현 사이의 변환을 수행하는 단계를 포함한다.

(52) CPC특허분류

HO4N 19/176 (2015.01)

HO4N 19/61 (2015.01)

HO4N 19/70 (2015.01)

HO4N 19/96 (2015.01)

(72) 발명자

류, 홍빈

중국 100080 베이징 하이뎬 디스트릭트 즈춘 로드
넘버 63 차이나 세틀라이트 커뮤니케이션즈 타워
진르터우탸오 포스트 오피스

왕, 웨

중국 100080 베이징 하이뎬 디스트릭트 즈춘 로드
넘버 63 차이나 세틀라이트 커뮤니케이션즈 타워
진르터우탸오 포스트 오피스

(30) 우선권주장

PCT/CN2018/105193 2018년09월12일 중국(CN)

PCT/CN2018/117627 2018년11월27일 중국(CN)

PCT/CN2019/071214 2019년01월10일 중국(CN)

명세서

청구범위

청구항 1

비디오 처리 방법으로서,

테이블을 유지하는 단계- 상기 테이블은 현재 비디오 블록 이전에 코딩된 하나 이상의 비디오 블록으로부터 도출된 하나 이상의 모션 후보를 포함하고, 상기 테이블 내의 상기 모션 후보들의 배열은 상기 테이블로 상기 모션 후보들을 추가하는 시퀀스에 기초함 -;

후보 리스트를 구성하는 단계- 상기 테이블 내의 적어도 하나의 제1 모션 후보가 상기 구성 동안 검사되고, 상기 적어도 하나의 제1 모션 후보를 상기 후보 리스트에 추가할지 여부를 결정하기 위해 상기 적어도 하나의 제1 모션 후보에 대해 프루닝 동작이 사용됨 -;

상기 후보 리스트를 사용하여 모션 정보를 도출하는 단계;

상기 모션 정보에 기초하여 상기 현재 비디오 블록을 코딩하는 단계를 포함하고,

상기 적어도 하나의 제1 모션 후보는 상기 후보 리스트에 포함된 적어도 하나의 제2 모션 후보를 사용하여 프루닝되고, 상기 적어도 하나의 제2 모션 후보 중 각각의 모션 후보는 상기 각각의 모션 후보가 어디로부터 도출되는지에 기초하여 상기 후보 리스트로부터 선택되고, 상기 적어도 하나의 제2 모션 후보는 상기 현재 비디오 블록의 주어진 공간 이웃 블록으로부터 도출되는 공간 모션 후보이고,

상기 적어도 하나의 제2 모션 후보의 총 수는 K이고, K는 2 이하이고,

상기 적어도 하나의 제2 모션 후보는, 블록 A1로부터 도출된 공간 모션 후보 또는 블록 B1로부터 도출된 공간 모션 후보 중 적어도 하나이고; 블록 A1은 상기 현재 비디오 블록에 대해 좌측 이웃 블록이고 블록 B1은 상기 현재 비디오 블록에 대해 상위 이웃 블록이며,

상기 적어도 하나의 제2 모션 후보는 서브 블록 기반 모션 후보를 배제하는, 방법.

청구항 2

제1항에 있어서,

상기 적어도 하나의 제2 모션 후보는 상기 후보 리스트 내의 처음 한 개의 모션 후보 또는 처음 N개의 모션 후보를 포함하고, N은 1보다 큰 정수인, 방법.

청구항 3

삭제

청구항 4

제1항에 있어서, 상기 적어도 하나의 제2 모션 후보는 상기 후보 리스트에 있었던 모션 후보들의 일부인, 방법.

청구항 5

제4항에 있어서, 상기 후보 리스트 내의 모션 후보들의 상기 일부는 공간 모션 후보들의 일부인, 방법.

청구항 6

제1항에 있어서, 상기 적어도 하나의 제1 모션 후보의 모션 후보가 상기 적어도 하나의 제2 모션 후보 중 임의의 모션 후보와 상이한 경우, 상기 모션 후보는 상기 후보 리스트에 추가되는, 방법.

청구항 7

삭제

청구항 8

제1항에 있어서, 상기 테이블로부터의 모션 후보를 상기 후보 리스트에 추가할 때 상기 프루닝 동작이 호출되지 않는, 방법.

청구항 9

제1항에 있어서, 상기 테이블의 상기 모션 후보는 예측 방향, 참조 픽처 인덱스, 모션 벡터 값, 강도 보상 플래그, 아핀 플래그, 모션 벡터 차이 정밀도, 또는 모션 벡터 차이 값 중 적어도 하나를 포함하는 모션 정보와 연관되는, 방법.

청구항 10

제1항에 있어서,

상기 방법은:

상기 현재 비디오 블록의 상기 모션 정보를 사용하여 상기 테이블을 업데이트하는 단계를 더 포함하는, 방법.

청구항 11

제10항에 있어서, 상기 모션 정보에 대응하는 모션 후보가 상기 테이블에 추가되기 전에 상기 테이블이 가득 찬 경우 상기 테이블 내의 모션 후보가 삭제되는, 방법.

청구항 12

제1항에 있어서, 상기 후보 리스트는 병합 후보 리스트인, 방법.

청구항 13

제1항에 있어서, 상기 코딩 단계는 상기 현재 비디오 블록을 비트스트림으로 인코딩하는 단계를 포함하는, 방법.

청구항 14

제1항에 있어서, 상기 코딩 단계는 비트스트림으로부터 상기 현재 비디오 블록을 디코딩하는 단계를 포함하는, 방법.

청구항 15

프로세서 및 명령어들을 갖는 비일시적 메모리를 포함하는 비디오 데이터를 코딩하기 위한 장치로서, 상기 명령어들은 상기 프로세서에 의한 실행 시에, 상기 프로세서로 하여금,

테이블을 유지하고- 상기 테이블은 현재 비디오 블록 이전에 코딩된 하나 이상의 비디오 블록으로부터 도출된 하나 이상의 모션 후보를 포함하고, 상기 테이블 내의 상기 모션 후보들의 배열은 상기 테이블로 상기 모션 후보들을 추가하는 시퀀스에 기초함 -;

후보 리스트를 구성하고- 상기 테이블에서 적어도 하나의 제1 모션 후보가 상기 구성 동안 검사되고, 상기 적어도 하나의 제1 모션 후보를 상기 후보 리스트에 추가할지 여부를 결정하기 위해 상기 적어도 하나의 제1 모션 후보에 대해 프루닝 동작이 이용됨 -;

상기 후보 리스트를 사용하여 모션 정보를 도출하고;

상기 모션 정보에 기초하여 상기 현재 비디오 블록을 코딩하게 하고,

상기 적어도 하나의 제1 모션 후보는 상기 후보 리스트에 포함된 적어도 하나의 제2 모션 후보를 사용하여 프루닝되고, 상기 적어도 하나의 제2 모션 후보 중 각각의 모션 후보는 상기 각각의 모션 후보가 어디로부터 도출되는지에 기초하여 상기 후보 리스트로부터 선택되고, 상기 적어도 하나의 제2 모션 후보는 상기 현재 비디오 블록의 주어진 공간 이웃 블록으로부터 도출되는 공간 모션 후보이고,

상기 적어도 하나의 제2 모션 후보의 총 수는 K이고, K는 2 이하이고,

상기 적어도 하나의 제2 모션 후보는, 블록 A1로부터 도출된 공간 모션 후보 또는 블록 B1로부터 도출된 공간 모션 후보 중 적어도 하나이고; 블록 A1은 상기 현재 비디오 블록에 대해 좌측 이웃 블록이고 블록 B1은 상기 현재 비디오 블록에 대해 상위 이웃 블록이며,

상기 적어도 하나의 제2 모션 후보는 서브 블록 기반 모션 후보를 배제하는, 비디오 데이터를 코딩하기 위한 장치.

청구항 16

명령어들을 저장하는 비일시적 컴퓨터 판독가능 저장 매체로서, 상기 명령어들은 프로세서로 하여금:

테이블을 유지하고- 상기 테이블은 현재 비디오 블록 이전에 코딩된 하나 이상의 비디오 블록으로부터 도출된 하나 이상의 모션 후보를 포함하고, 상기 테이블 내의 상기 모션 후보들의 배열은 상기 테이블로 상기 모션 후보들을 추가하는 시퀀스에 기초함 -;

후보 리스트를 구성하고- 상기 테이블에서 적어도 하나의 제1 모션 후보가 상기 구성 동안 검사되고, 상기 적어도 하나의 제1 모션 후보를 상기 후보 리스트에 추가할지 여부를 결정하기 위해 상기 적어도 하나의 제1 모션 후보에 대해 프루닝 동작이 이용됨 -;

상기 후보 리스트를 사용하여 모션 정보를 도출하고;

상기 모션 정보에 기초하여 상기 현재 비디오 블록을 코딩하게 하고,

상기 적어도 하나의 제1 모션 후보는 상기 후보 리스트에 포함된 적어도 하나의 제2 모션 후보를 사용하여 프루닝되고, 상기 적어도 하나의 제2 모션 후보 중 각각의 모션 후보는 상기 각각의 모션 후보가 어디로부터 도출되는지에 기초하여 상기 후보 리스트로부터 선택되고, 상기 적어도 하나의 제2 모션 후보는 상기 현재 비디오 블록의 주어진 공간 이웃 블록으로부터 도출되는 공간 모션 후보이고,

상기 적어도 하나의 제2 모션 후보의 총 수는 K이고, K는 2 이하이고,

상기 적어도 하나의 제2 모션 후보는, 블록 A1로부터 도출된 공간 모션 후보 또는 블록 B1로부터 도출된 공간 모션 후보 중 적어도 하나이고; 블록 A1은 상기 현재 비디오 블록에 대해 좌측 이웃 블록이고 블록 B1은 상기 현재 비디오 블록에 대해 상위 이웃 블록이며,

상기 적어도 하나의 제2 모션 후보는 서브 블록 기반 모션 후보를 배제하는, 비일시적 컴퓨터 판독가능 저장 매체.

청구항 17

제1항에 있어서, 상기 후보 리스트 내의 모션 후보들의 개수가 상기 후보 리스트의 허용된 최대 후보 개수에서 임계값을 뺀 값에 도달할 때 검사가 종료되고, 상기 임계값은 양의 정수인, 방법.

청구항 18

비디오 처리 장치에 의해 수행되는 방법에 의해 생성되는 비트스트림을 저장하는 비일시적 컴퓨터 판독가능 기록 매체로서, 상기 방법은,

테이블을 유지하는 단계- 상기 테이블은 현재 비디오 블록 이전에 코딩된 하나 이상의 비디오 블록으로부터 도출된 하나 이상의 모션 후보를 포함하고, 상기 테이블 내의 상기 모션 후보들의 배열은 상기 테이블로 상기 모션 후보들을 추가하는 시퀀스에 기초함 -;

후보 리스트를 구성하는 단계- 상기 테이블 내의 적어도 하나의 제1 모션 후보가 상기 구성 동안 검사되고, 상기 적어도 하나의 제1 모션 후보를 상기 후보 리스트에 추가할지 여부를 결정하기 위해 상기 적어도 하나의 제1 모션 후보에 대해 프루닝 동작이 사용됨 -;

상기 후보 리스트를 사용하여 모션 정보를 도출하는 단계;

상기 모션 정보에 기초하여 현재 비디오 블록으로부터 상기 비트스트림을 생성하는 단계를 포함하고,

상기 적어도 하나의 제1 모션 후보는 상기 후보 리스트에 포함된 적어도 하나의 제2 모션 후보를 사용하여 프루닝되고, 상기 적어도 하나의 제2 모션 후보 중 각각의 모션 후보는 상기 각각의 모션 후보가 어디로부터 도출되

는지에 기초하여 상기 후보 리스트로부터 선택되고, 상기 적어도 하나의 제2 모션 후보는 상기 현재 비디오 블록의 주어진 공간 이웃 블록으로부터 도출되는 공간 모션 후보이고,

상기 적어도 하나의 제2 모션 후보의 총 수는 K이고, K는 2 이하이고,

상기 적어도 하나의 제2 모션 후보는, 블록 A1로부터 도출된 공간 모션 후보 또는 블록 B1로부터 도출된 공간 모션 후보 중 적어도 하나이고; 블록 A1은 상기 현재 비디오 블록에 대해 좌측 이웃 블록이고 블록 B1은 상기 현재 비디오 블록에 대해 상위 이웃 블록이며,

상기 적어도 하나의 제2 모션 후보는 서브 블록 기반 모션 후보를 배제하는, 비일시적 컴퓨터 판독가능 기록 매체.

발명의 설명

기술 분야

[0001]

관련 출원들에 대한 상호 참조

[0002]

본 출원은 2019년 7월 1일자로 출원된 국제 특허 출원 제PCT/IB2019/055582호의 국내 단계 출원인 2020년 12월 30일자로 출원된 한국 특허 출원번호 제10-2020-7038007호의 분할출원이고, 한국 특허출원번호 제10-2020-7038007호는 2018년 6월 29일자로 출원된 국제 특허 출원 제PCT/CN2018/093663호, 2018년 9월 12일자로 출원된 국제 특허 출원 제PCT/CN2018/105193호, 2018년 11월 27일자로 출원된 국제 특허 출원 제PCT/CN2018/117627호 및 2019년 1월 10일자로 출원된 국제 특허 출원 제PCT/CN2019/071214호에 대한 우선권 및 이익을 주장한다. 미국 법 하에서 모든 목적을 위해, 국제 특허 출원 번호 PCT/CN2018/093663, 국제 특허 출원 번호 PCT/CN2018/105193, 국제 특허 출원 번호 PCT/CN2018/117627 및 국제 특허 출원 번호 PCT/CN2019/071214의 전체 개시내용은 본 출원의 개시내용의 일부로서 참조로 포함된다.

[0003]

기술 분야

[0004]

본 특허 문서는 비디오 코딩 및 디코딩 기술, 디바이스 및 시스템에 관련한 것이다.

배경 기술

[0005]

비디오 압축의 발전에도 불구하고, 디지털 비디오는 인터넷 및 다른 디지털 통신 네트워크들에서 여전히 가장 큰 대역폭 사용을 차지한다. 비디오를 수신 및 디스플레이할 수 있는 연결된 사용자 디바이스들의 수가 증가함에 따라, 디지털 비디오 사용에 대한 대역폭 수요가 계속 증가할 것으로 예상된다.

발명의 내용

해결하려는 과제

[0006]

본 문서는 모션 벡터들의 병합 리스트를 사용하여 디지털 비디오를 인코딩 및 디코딩하기 위한 방법들, 시스템들 및 디바이스들을 개시한다.

과제의 해결 수단

[0007]

하나의 예시적인 양태에서, 비디오 처리 방법이 제공되며, 이 방법은: 테이블들의 세트를 유지하는 단계- 각각의 테이블은 모션 후보들을 포함하고, 각각의 모션 후보는 대응하는 모션 정보와 연관됨 -; 모션 후보들에 대한 프루닝 동작을 사용하여 하나 이상의 테이블 내의 모션 후보들에 기초하여 후보 리스트를 업데이트하는 단계; 및 구성된 후보 리스트를 사용하여 제1 비디오 블록과 제1 비디오 블록을 포함하는 비디오의 비트스트림 표현 사이의 변환을 수행하는 단계를 포함한다.

[0008]

다른 양태에서, 비디오 처리 방법이 제공되며, 이는 제1 비디오 블록을 포함하는 비디오의 비트스트림 표현을 수신하는 단계; 및 후보 리스트를 업데이트하기 위해 하나 이상의 테이블 내의 모션 후보들에 프루닝 동작을 적용하는 단계- 각각의 테이블은 모션 후보들을 포함하고, 각각의 모션 후보는 대응하는 모션 정보와 연관됨 -; 구성된 후보 리스트를 사용하여 비트스트림 표현과 제1 비디오 블록 사이의 변환을 수행하는 단계를 포함한다.

[0009]

또 다른 대표적인 양태에서, 본 명세서에 설명된 다양한 기술들이 비일시적 컴퓨터 판독가능 매체 상에 저장된 컴퓨터 프로그램 제품으로서 구현될 수 있다. 컴퓨터 프로그램 제품은 본 명세서에 설명된 방법들을 수행하기

위한 프로그램 코드를 포함한다.

[0010] 또 다른 대표적인 양태에서, 비디오 인코딩 또는 디코딩 장치는 본 명세서에 설명된 방법을 구현할 수 있다.

[0011] 하나 이상의 구현의 세부사항들이 이하의 첨부물들, 도면들 및 설명에서 기재되어 있다. 다른 특징들은 설명 및 도면들로부터 그리고 청구항들로부터 명백히 알 수 있을 것이다.

도면의 간단한 설명

[0012] 도 1은 비디오 인코더 구현의 예를 나타내는 블록도이다.

도 2는 H.264 비디오 코딩 표준의 매크로블록 파티셔닝을 도시한다.

도 3은 코딩 블록(CB)들을 예측 블록(PU)들로 분할하는 예를 도시한다.

도 4는 CTB를 CB들 및 변환 블록(TB)들로 세분화하기 위한 예시적인 구현을 도시한다. 실선들은 CB 경계들을 나타내고, 점선들은 그 파티셔닝을 갖는 예시적인 CTB, 및 대응하는 큐드트리를 포함하는 TB 경계들을 나타낸다.

도 5는 비디오 데이터를 파티셔닝하기 위한 QTBT(Quad Tree Binary Tree) 구조의 예를 도시한다.

도 6은 비디오 블록 파티셔닝의 예를 도시한다.

도 7은 큐드-트리 파티셔닝의 예를 도시한다.

도 8은 트리-타입 시그널링의 예를 도시한다.

도 9는 병합 후보 리스트 구성을 위한 도출 프로세스의 예를 도시한다.

도 10은 공간 병합 후보들의 예시적인 위치들을 도시한다.

도 11은 공간 병합 후보들의 중복 검사를 위해 고려되는 후보 쌍들의 예들을 도시한다.

도 12는 Nx2N 및 2NxN 파티션의 제2 PU에 대한 위치의 예를 도시한다.

도 13은 시간 병합 후보들에 대한 모션 벡터 스케일링을 예시한다.

도 14는 시간 병합 후보들 및 그들의 병치된 픽처에 대한 후보 위치들을 도시한다.

도 15는 조합된 양방향 예측 병합 후보의 예를 도시한다.

도 16은 모션 벡터 예측 후보에 대한 도출 프로세스의 예를 도시한다.

도 17은 공간 모션 벡터 후보에 대한 모션 벡터 스케일링의 예를 도시한다.

도 18은 CU의 모션 예측을 위한 예시적인 ATMVP(Alternative Temporal Motion Vector Prediction)를 도시한다.

도 19는 소스 블록 및 소스 픽처의 식별의 예를 도식적으로 도시한다.

도 20은 4개의 서브 블록 및 이웃 블록들을 갖는 하나의 CU의 예를 도시한다.

도 21은 양방향 매칭의 예를 도시한다.

도 22는 템플릿 매칭의 예를 도시한다.

도 23은 FRUC(Frame Rate Up Conversion)에서의 단방향 모션 추정(ME)의 예를 도시한다.

도 24는 양방향 템플릿 매칭에 기초한 DMVR의 예를 도시한다.

도 25는 공간 병합 후보들을 도출하는 데 사용되는 공간 이웃 블록들의 예를 도시하는 도면이다.

도 26은 루업 테이블 업데이트들을 위한 대표적인 위치의 선택 방법의 예를 도시한다.

도 27a 및 도 27b는 새로운 모션 정보 세트로 루업 테이블을 업데이트하는 예를 도시하는 도면이다.

도 28은 본 문서에 설명된 비주얼 미디어 디코딩 또는 비주얼 미디어 인코딩 기술을 구현하는 하드웨어 플랫폼의 예의 블록도이다.

도 29는 예시적인 비디오 처리 방법에 대한 흐름도이다.

도 30은 비디오 처리의 다른 예시적인 방법에 대한 흐름도이다.

도 31은 제안된 HMVP 방법을 갖는 디코딩 흐름도의 예를 도시한다.

도 32는 제안된 HMVP 방법을 사용하여 테이블들을 업데이트하는 예들을 도시한다.

도 33a 및 도 33b는 (하나의 중복 모션 후보가 제거된) 중복 제거 기반 LUT 업데이트 방법의 예들을 도시한다.

도 34a 및 도 34b는 (다수의 중복 모션 후보들이 제거된) 중복 제거 기반 LUT 업데이트 방법의 예들을 도시한다.

발명을 실시하기 위한 구체적인 내용

[0013] 비디오의 압축비를 향상시키기 위해, 연구자들은 비디오를 인코딩하는 새로운 기술들을 계속 찾고 있다.

1. 도입

[0015] 본 문서는 비디오 코딩 기술들에 관련한 것이다. 구체적으로, 이는 비디오 코딩에서의 모션 정보 코딩(예를 들어 병합 모드, AMVP 모드)에 관련된다. 이는 HEVC와 같은 기존의 비디오 코딩 표준, 또는 완성될 표준(다용도 비디오 코딩(Versatile Video Coding))에 적용될 수 있다. 이는 또한 장래의 비디오 코딩 표준들 또는 비디오 코덱에도 적용 가능할 수 있다.

간략한 설명

[0017] 비디오 코딩 표준들은 주로 알려진 ITU-T 및 ISO/IEC 표준들의 개발을 통해 발전했다. ITU-T는 H.261 및 H.263를 생성하고, ISO/IEC는 MPEG-1 및 MPEG-4 Visual을 생성하였으며, 두 기구가 공동으로 H.262/MPEG-2 비디오 및 H.264/MPEG-4 AVC(Advanced Video Coding) 및 H.265/HEVC 표준들을 생성하였다. H.262 이래로, 비디오 코딩 표준들은 시간 예측 + 변환 코딩이 이용되는 하이브리드 비디오 코딩 구조에 기초한다. 전형적인 HEVC 인코더 프레임워크의 예가 도 1에 도시된다.

2.1 파티션 구조

2.1.1 H.264/AVC에서의 파티션 트리 구조

[0020] 이전 표준들에서의 코딩 층의 코어는 루마 샘플들의 16X16 블록 및, 4:2:0 컬러 샘플링의 일반적인 경우에, 크로마 샘플들의 2개의 대응하는 8X8 블록들을 포함하는 매크로블록이었다.

[0021] 인트라-코딩된 블록은 픽셀들 간의 공간 상관을 활용하기 위해 공간 예측을 사용한다. 2개의 파티션, 즉 16x16 및 4x4가 정의된다.

[0022] 인터-코딩된 블록은 픽처들 간의 모션을 추정함으로써 공간 예측 대신에 시간 예측을 사용한다. 모션은 16x16 매크로블록 또는 그의 서브-매크로블록 파티션들- 16x8, 8x16, 8x8, 8x4, 4x8, 4x4(도 2 참조) - 중 임의의 것 중 어느 쪽이든 대해 독립적으로 추정될 수 있다: 서브-매크로블록 파티션당 단 하나의 모션 벡터(MV) 만이 허용된다.

2.1.2 HEVC에서의 파티션 트리 구조

[0024] HEVC에서, CTU는 다양한 로컬 특성들에 적응하기 위해 코딩 트리로서 표시된 큐드트리 구조를 사용함으로써 CU들로 분할된다. 인터-픽처(시간) 또는 인트라-픽처(공간) 예측 중 어느쪽을 사용하여 픽처 영역을 코딩할지의 결정은 CU 레벨에서 이루어진다. 각각의 CU는 PU 분할 타입에 따라 1개, 2개 또는 4개의 PU로 추가로 분할될 수 있다. 하나의 PU 내부에서, 동일한 예측 프로세스가 적용되고 관련 정보가 PU 기준으로 디코더에 송신된다. PU 분할 타입에 기초하여 예측 프로세스를 적용함으로써 잔차 블록을 획득한 이후에, CU는 CU에 대한 코딩 트리와 유사한 다른 큐드트리 구조에 따라 변환 유닛(TU)들로 파티셔닝될 수 있다. HEVC 구조의 주요 특징 중 하나는 그것이 CU, PU, 및 TU를 포함하는 다수의 파티션 개념들을 갖는다는 것이다.

[0025] 이하에서, HEVC를 사용하는 하이브리드 비디오 코딩에 수반되는 다양한 특징들이 다음과 같이 강조된다.

[0026] 1) 코딩 트리 유닛 및 코딩 트리 블록(CTB) 구조: HEVC에서의 유사한 구조는 인코더에 의해 선택된 크기를 가지며 전통적인 매크로블록보다 클 수 있는 코딩 트리 유닛(CTU)이다. CTU는 루마 CTB 및 대응하는 크로마 CTB들 및 신팩스 엘리먼트들로 구성된다. 루마 CTB의 크기 $L \times L$ 은 $L = 16, 32$ 또는 64 샘플로서 선택될 수 있고, 더 큰 크기들은 전형적으로 더 양호한 압축을 가능하게 한다. HEVC는 이어서 트리 구조 및 큐드트리형 시그널링을 사

용하여 CTB들의 더 작은 블록들로의 파티셔닝을 지원한다.

[0027] 2) 코딩 유닛(CU)들 및 코딩 블록(CB)들: CTU의 큐드트리 선택스는 그 루마 및 크로마 CB들의 크기 및 위치들을 지정한다. 큐드트리의 루트는 CTU와 연관된다. 따라서, 루마 CTU의 크기는 루마 CB에 대해 가장 큰 지원되는 크기이다. 루마 및 크로마 CB들로의 CTU의 분할은 공동으로 시그널링된다. 하나의 루마 CB 및 통상적으로 2개의 크로마 CB들은 연관된 선택스와 함께 코딩 유닛(CU)을 형성한다. CTU는 단 하나의 CU만을 포함할 수 있거나 다수의 CU들을 형성하도록 분할될 수 있고, 각각의 CU는 예측 유닛(PU)들 및 변환 유닛(TU)들의 트리로의 연관된 파티셔닝을 갖는다.

[0028] 3) 예측 유닛들 및 예측 블록(PB)들: 인터 픽처 또는 인트라 픽처 예측 중 어느 쪽을 사용하여 픽처 영역을 코딩할지의 결정은 CU 레벨에서 이루어진다. PU 파티셔닝 구조는 CU 레벨에서 그의 루트를 갖는다. 기본 예측 타입 결정에 따라, 루마 및 크로마 CB들은 이어서 크기가 추가로 분할되고 루마 및 크로마 예측 블록(PB)들로부터 예측될 수 있다. HEVC는 64X64에서 4X4 샘플까지의 가변 PB 크기들을 지원한다. 도 3은 MxM CU에 대한 허용된 PB들의 예들을 도시한다.

[0029] 4) TU들 및 변환 블록들: 예측 잔차는 블록 변환을 사용하여 코딩된다. TU 트리 구조는 CU 레벨에서 그의 루트를 갖는다. 루마 CB 잔차는 루마 변환 블록(TB)과 동일하거나 더 작은 루마 TB들로 추가로 분할될 수 있다. 이는 크로마 TB에도 적용된다. 이산 코사인 변환(DCT)의 것들과 유사한 정수 기저 함수들은 정사각형 TB 크기들 4X4, 8X8, 16X16 및 32X32에 대해 정의된다. 루마 인트라 픽처 예측 잔차들의 4X4 변환의 경우, 이산 사인 변환(DST)의 형태로부터 도출된 정수 변환이 대안적으로 지정된다.

[0030] 도 4는 CTU의 CB들[및 변환 블록(TB들)]로의 세분화의 예를 도시한다. 실선들은 CB 경계들을 나타내고 점선들은 TB 경계들을 나타낸다. (a) 자신의 파티셔닝을 갖는 CTU. (b) 대응하는 큐드트리

2.1.2.1 변환 블록들 및 유닛들로의 트리-구조 파티셔닝

[0032] 잔차 코딩의 경우, CB는 변환 블록(TB)들로 재귀적으로 파티셔닝될 수 있다. 파티셔닝은 잔차 큐드트리에 의해 시그널링된다. 정사각형 CB 및 TB 파티셔닝만이 지정되며, 여기서 블록은 도 4에 도시된 바와 같이 사분면들로 재귀적으로 분할될 수 있다. 크기 MxM의 주어진 루마 CB에 대해, 플래그는 그것이 크기 M/2xM/2의 4개의 블록들로 분할되는지 여부를 시그널링한다. SPS에 표시되는 잔차 큐드트리의 최대 깊이에 의해 시그널링되는 바와 같이, 추가의 분할이 가능하면, 각각의 사분면에는 이것이 4개의 사분면으로 분할되는지 여부를 표시하는 플래그가 할당된다. 잔차 큐드트리로부터 얻어지는 리프 노드 블록들은 변환 코딩에 의해 추가로 처리되는 변환 블록들이다. 인코더는 사용할 최대 및 최소 루마 TB 크기를 나타낸다. CB 크기가 최대 TB 크기보다 클 때 분할은 암시적이다. 분할이 표시된 최소값보다 작은 루마 TB 크기를 초래할 때는 분할은 암시적이지 않다. 크로마 TB 크기는, 루마 TB 크기가 4X4인 경우를 제외하고, 각 차원에서 루마 TB 크기의 절반이며, 이 경우 단일 4X4 크로마 TB가 4개의 4X4 루마 TB들에 의해 커버되는 영역에 사용된다. 인트라-픽처-예측된 CU들의 경우, (CB 내 또는 외부의) 가장 가까운-이웃하는 TB들의 디코딩된 샘플들이 인트라 픽처 예측을 위한 기준 데이터로서 사용된다.

[0033] 이전의 표준들과 대조적으로, HEVC 설계는 TB가 큐드트리 구조 TB 파티셔닝의 잠재적 코딩 효율 이점들을 최대화하기 위해 인터-픽처 예측 CU들에 대해 다수의 PB들에 걸쳐질 수 있게 한다.

2.1.2.2 부모 및 자식 노드들

[0035] CTU는 그 노드들이 코딩 유닛들인 큐드-트리 구조에 따라 분할된다. 큐드-트리 구조 내의 복수의 노드들은 리프 노드들 및 비-리프 노드들을 포함한다. 리프 노드들은 트리 구조에서 자식 노드들을 갖지 않는다(즉, 리프 노드들은 추가로 분할되지 않는다). 비-리프 노드들은 트리 구조의 루트 노드를 포함한다. 루트 노드는 비디오 데이터(예를 들어, CTU)의 초기 비디오 블록에 대응한다. 복수의 노드들의 각각의 비루트 노드 각각에 대해, 각각의 비루트 노드는 각각의 비루트 노드의 트리 구조 내의 부모 노드에 대응하는 비디오 블록의 서브 블록인 비디오 블록에 대응한다. 복수의 비-리프 노드 중 각각의 비-리프 노드 각각은 트리 구조에서 하나 이상의 자식 노드를 갖는다.

2.1.3 JEM에서 더 큰 CTU들을 갖는 큐드트리 + 이진 트리 블록 구조

[0037] HEVC를 넘어서는 장래의 비디오 코딩 기술들을 탐구하기 위해, JVET(Joint Video Exploration Team)가 2015년에 VCEG 및 MPEG에 의해 공동으로 설립되었다. 그 후, 많은 새로운 방법들이 JVET에 의해 채택되어 JEM(Joint Exploration Model)이라 명명된 참조 소프트웨어에 들어갔다.

2.1.3.1 QTBT 블록 파티셔닝 구조

HEVC와 달리, QTBT 구조는 다수의 파티션 타입들의 개념들을 제거하며, 즉, 이는 CU, PU 및 TU 개념들의 분리를 제거하고, CU 파티션 형상들에 대한 더 많은 유연성을 지원한다. QTBT 블록 구조에서, CU는 정사각형 또는 직사각형 형상 어느 쪽이든 가질 수 있다. 도 5에 도시된 바와 같이, 코딩 트리 유닛(CTU)이 쿼드트리 구조에 의해 먼저 파티셔닝된다. 쿼드트리 리프 노드들은 이진 트리 구조에 의해 추가로 파티셔닝된다. 이진 트리 분할에는 대칭 수평 분할 및 대칭 수직 분할의 2가지 분할 타입이 있다. 이진 트리 리프 노드들은 코딩 유닛(CU)들이라고 불리며, 이러한 세그먼트화는 어떠한 추가적인 파티셔닝도 없이 예측 및 변환 처리를 위해 사용된다. 이는 CU, PU 및 TU가 QTBT 코딩 블록 구조에서 동일한 블록 크기를 갖는다는 것을 의미한다. JEM에서, CU는 때때로 상이한 컬러 성분들의 코딩 블록(CB)들로 구성되고, 예를 들어, 하나의 CU는 4:2:0 크로마 포맷의 P 및 B 슬라이스들의 경우에 하나의 루마 CB 및 2개의 크로마 CB를 포함하고, 때때로 단일 성분의 CB로 구성되며, 예를 들어, 하나의 CU는 I 슬라이스들의 경우에 단 하나의 루마 CB 또는 단지 2개의 크로마 CB를 포함한다.

QTBT 파티셔닝 방식에 대해 다음 파라미터들이 정의된다:

- CTU 크기: 쿼드트리의 루트 노드 크기, HEVC에서와 동일한 개념

- *MinQTSsize*: 최소 허용 쿼드트리 리프 노드 크기

- *MaxBTSsize*: 최대 허용 이진 트리 루트 노드 크기

- *MaxBTDepth*: 최대 허용 이진 트리 깊이

- *MinBTSsize*: 최소 허용 이진 트리 리프 노드 크기

QTBT 파티셔닝 구조의 일 예에서, CTU 크기는 128X128 루마 샘플들과 2개의 대응하는 64X64 블록의 크로마 샘플들로서 설정되고, *MinQTSsize*은 16X16으로 설정되고, *MaxBTSsize*는 64X64로 설정되고, *MinBTSsize* (폭과 높이 둘 다에 대해)는 4X4로 설정되고, *MaxBTDepth*는 4로 설정된다. 쿼드트리 파티셔닝이 먼저 CTU에 적용되어 쿼드트리 리프 노드들을 생성한다. 쿼드트리 리프 노드들은 16X16(즉, *MinQTSsize*) 내지 128X128(즉, CTU 크기)의 크기를 가질 수 있다. 리프 쿼드트리 노드가 128X128이면, 크기는 *MaxBTSsize*(즉, 64X64)를 초과하기 때문에 이진 트리에 의해 추가로 분할되지 않을 것이다. 그렇지 않으면, 리프 쿼드트리 노드는 이진 트리에 의해 추가로 파티셔닝될 수 있다. 따라서, 쿼드트리 리프 노드는 또한 이진 트리에 대한 루트 노드이고 그것은 0으로서 이진 트리 깊이를 갖는다. 이진 트리 깊이가 *MaxBTDepth*(즉, 4)에 도달할 때, 더 이상의 분할이 고려되지 않는다. 이진 트리 노드가 *MinBTSsize*(즉, 4)와 동일한 폭을 가질 때, 더 이상 수평 분할이 고려되지 않는다. 유사하게, 이진 트리 노드가 *MinBTSsize*와 동일한 높이를 갖는 경우, 더 이상 수직 분할이 고려되지 않는다. 이진 트리의 리프 노드들은 임의의 추가적인 파티셔닝 없이 예측 및 변환 처리에 의해 추가로 처리된다. JEM에서, 최대 CTU 크기는 256X256 루마 샘플들이다.

도 5는 QTBT를 사용한 블록 파티셔닝의 예를 예시하고(좌측), 도 5는 대응하는 트리 표현을 예시한다(우측). 실선들은 쿼드트리 분할을 나타내고 점선들은 이진 트리 분할을 나타낸다. 이진 트리의 각각의 분할(즉, 비-리프) 노드에서, 어느 분할 타입(즉, 수평 또는 수직)이 사용되는지를 나타내기 위해 하나의 플래그가 시그널링되며, 여기서 0은 수평 분할을 나타내고 1은 수직 분할을 나타낸다. 쿼드트리 분할의 경우, 분할 타입을 나타낼 필요가 없는데, 그 이유는 쿼드트리 분할이 항상 블록을 수평 및 수직 양자 모두로 분할하여 동일한 크기를 갖는 4개의 서브 블록을 생성하기 때문이다.

또한, QTBT 방식은 루마 및 크로마가 별개의 QTBT 구조를 갖는 능력을 지원한다. 현재, P 및 B 슬라이스들에 대해, 하나의 CTU 내의 루마 및 크로마 CTB들은 동일한 QTBT 구조를 공유한다. 그러나, I 슬라이스들에 대해, 루마 CTB는 QTBT 구조에 의해 CU들로 파티셔닝되고, 크로마 CTB들은 다른 QTBT 구조에 의해 크로마 CU들로 파티셔닝된다. 이것은 I 슬라이스 내의 CU가 루마 성분의 코딩 블록 또는 2개의 크로마 성분의 코딩 블록들로 구성되고, P 또는 B 슬라이스 내의 CU가 3개의 컬러 성분 모두의 코딩 블록들로 구성된다는 것을 의미한다.

HEVC에서, 작은 블록들에 대한 인터 예측이 제한되어 모션 보상의 메모리 액세스를 감소시키며, 그래서, 양방향 예측은 4X8 및 8X4 블록들에 대해 지원되지 않고, 인터 예측은 4X4 블록들에 대해 지원되지 않는다. JEM의 QTBT에서, 이러한 제한들은 제거된다.

2.1.4 VVC를 위한 삼진-트리

일부 구현들에서, 쿼드-트리 및 이진-트리 이외의 트리 타입들이 지원된다. 이 구현에서, 도 6의 부분 (d) 및

(e)에 도시된 바와 같이, 2개의 더 많은 삼진 트리(TT) 파티션, 즉, 수평 및 수직 중심축 삼진-트리가 도입된다.

[0052] 도 6은 다음을 도시한다: (a) 쿼드-트리 파티셔닝 (b) 수직 이진-트리 파티셔닝 (c) 수평 이진-트리 파티셔닝 (d) 수직 중심축 삼진-트리 파티셔닝 (e) 수평 중심축 삼진-트리 파티셔닝.

[0053] 일부 구현들에서, 영역 트리(쿼드-트리) 및 예측 트리(이진-트리 또는 삼진-트리)의 2개의 레벨들의 트리들이 존재한다. CTU는 먼저 영역 트리(RT)에 의해 파티셔닝된다. RT 리프는 예측 트리(PT)로 추가로 분할될 수 있다. PT 리프는 또한 최대 PT 깊이에 도달할 때까지 PT로 추가로 분할될 수 있다. PT 리프는 기본 코딩 유닛이다. 이는 편의상 여전히 CU라고 불린다. CU는 추가로 분할될 수 없다. 예측 및 변환은 둘 다 JEM과 동일한 방식으로 CU에 적용된다. 전체 파티션 구조는 "다중 타입 트리"로 명명된다.

2.1.5 [8]에서의 파티셔닝 구조

[0055] MTT(Multi-Tree Type)라고 불리는, 이 응답에 사용되는 트리 구조는 QTBT의 일반화이다. QTBT에서, 도 5에 도시된 바와 같이, 코딩 트리 유닛(CTU)은 먼저 쿼드-트리 구조에 의해 파티셔닝된다. 쿼드-트리 리프 노드들은 이진-트리 구조에 의해 추가로 파티셔닝된다.

[0056] MTT의 기본 구조는 2가지 타입의 트리 노드로 구성된다: 도 7에 도시된 바와 같이, 9개의 타입의 파티션을 지원하는 영역 트리(RT) 및 예측 트리(PT).

[0057] 도 7은 다음을 도시한다: (a) 쿼드-트리 파티셔닝 (b) 수직 이진-트리 파티셔닝 (c) 수평 이진-트리 파티셔닝 (d) 수직 삼진-트리 파티셔닝 (e) 수평 삼진-트리 파티셔닝 (f) 수평-상측 비대칭 이진-트리 파티셔닝 (g) 수평-하측 비대칭 이진-트리 파티셔닝 (h) 수직-좌측 비대칭 이진-트리 파티셔닝 (i) 수직-우측 비대칭 이진-트리 파티셔닝.

[0058] 영역 트리는 CTU를 4x4 크기 영역 트리 리프 노드로 정사각형 블록들로 재귀적으로 분할할 수 있다. 영역 트리 내의 각각의 노드에서, 예측 트리는 3개의 트리 타입 중 하나로부터 형성될 수 있다: 이진 트리(BT), 삼진-트리(TT) 및 비대칭 이진 트리(ABT). PT 분할에서, 예측 트리의 분기들에 쿼드트리 파티션을 갖는 것이 금지된다. JEM에서와 같이, 루마 트리 및 크로마 트리는 I 슬라이스들에서 분리된다. RT 및 PT에 대한 시그널링 방법이 도 8에 예시된다.

2.2 HEVC/H.265에서의 인터 예측

[0060] 각각의 인터-예측된 PU는 하나 또는 2개의 참조 픽처 리스트에 대한 모션 파라미터를 갖는다. 모션 파라미터들은 모션 벡터 및 참조 픽처 인덱스를 포함한다. 2개의 참조 픽처 리스트 중 하나의 사용은 또한 *inter_pred_idc*를 사용하여 시그널링될 수 있다. 모션 벡터들은 예측자들에 대한 엘타들로서 명시적으로 코딩될 수 있고, 이러한 코딩 모드는 AMVP 모드로 불린다.

[0061] CU가 스킵 모드로 코딩될 때, 하나의 PU가 CU와 연관되고, 유의한 잔차 계수들이 없고, 코딩된 모션 벡터 엘타 또는 참조 픽처 인덱스가 없다. 병합 모드가 지정되고, 이에 의해, 공간 및 시간 후보들을 포함하는 이웃하는 PU들로부터 현재 PU에 대한 모션 파라미터들이 획득된다. 병합 모드는 스킵 모드에 대해서뿐만 아니라 임의의 인터-예측된 PU에 적용될 수 있다. 병합 모드에 대한 대안은 모션 파라미터들의 명시적 송신이며, 여기서 모션 벡터, 각각의 참조 픽처 리스트에 대한 대응하는 참조 픽처 인덱스 및 참조 픽처 리스트 사용은 각각의 PU마다 명시적으로 시그널링된다.

[0062] 시그널링이 2개의 참조 픽처 리스트 중 하나가 사용될 것임을 나타낼 때, PU는 샘플들의 하나의 블록으로부터 생성된다. 이것은 "단방향 예측"이라고 지칭된다. P-슬라이스 및 B-슬라이스 둘 다에 대해 단방향 예측이 사용 가능하다.

[0063] 시그널링이 양자 모두의 참조 픽처 리스트가 사용될 것을 나타내는 경우, PU는 샘플들의 2개의 블록으로부터 생성된다. 이것은 "양방향 예측"이라고 지칭된다. 양방향 예측은 B-슬라이스들에 대해서만 사용 가능하다.

[0064] 다음 텍스트는 HEVC에서 지정된 인터 예측 모드들에 대한 세부사항들을 제공한다. 설명은 병합 모드에서 시작할 것이다.

2.2.1 병합 모드

2.2.1.1 병합 모드에 대한 후보들의 도출

[0067] PU가 병합 모드를 사용하여 예측될 때, 병합 후보 리스트 내의 엔트리를 가리키는 인덱스가 비트스트림으로부터

파싱되고 모션 정보를 검색하는 데 사용된다. 이 리스트의 구성은 HEVC 표준에서 지정되고 다음 단계들의 시퀀스에 따라 요약될 수 있다:

[0068] · 단계 1: 초기 후보 도출

[0069] - 단계 1.1: 공간 후보 도출

[0070] - 단계 1.2: 공간 후보에 대한 중복 검사

[0071] - 단계 1.3: 시간 후보 도출

[0072] · 단계 2: 추가 후보 삽입

[0073] - 단계 2.1: 양방향 예측 후보들의 생성

[0074] - 단계 2.2: 제로 모션 후보들의 삽입

[0075] 이러한 단계들은 또한 도 9에 개략적으로 도시된다. 공간 병합 후보 도출의 경우, 5개의 상이한 위치에 위치하는 후보들 중에서 최대 4개의 병합 후보가 선택된다. 시간 병합 후보 도출의 경우, 2개의 후보 중에서 최대 하나의 병합 후보가 선택된다. 각각의 PU에 대한 일정한 수의 후보들이 디코더에서 가정되기 때문에, 후보들의 수가 슬라이스 헤더에서 시그널링되는 병합 후보의 최대 수(MaxNumMergeCand)에 도달하지 않을 때 추가 후보들이 생성된다. 후보들의 수가 일정하기 때문에, 최상의 병합 후보의 인덱스는 절단 단항 이진화(TU)를 사용하여 인코딩된다. CU의 크기가 8과 동일한 경우, 현재 CU의 모든 PU들은 2Nx2N 예측 유닛의 병합 후보 리스트와 동일한 단일 병합 후보 리스트를 공유한다.

[0076] 이하에서는, 전술한 단계들과 연관된 동작들이 상세히 설명된다.

2.2.1.2 공간 후보 도출

[0078] 공간 병합 후보들의 도출에서, 도 10에 도시된 위치들에 배치된 후보들 중에서 최대 4개의 병합 후보가 선택된다. 도출 순서는 A_1, B_1, B_0, A_0 및 B_2 이다. 위치 B_2 는 위치 A_1, B_1, B_0, A_0 의 임의의 PU가 사용가능하지 않을 때(예를 들어, 그것이 다른 슬라이스 또는 타일에 속하기 때문에) 또는 인트라 코딩될 때에만 고려된다. 위치 A_1 에서의 후보가 추가된 이후에, 나머지 후보들의 추가는 동일한 모션 정보를 갖는 후보들이 리스트로부터 배제되어 코딩 효율이 향상되도록 보장하는 중복 검사를 받는다. 계산 복잡성을 감소시키기 위해, 모든 가능한 후보 쌍들이 언급된 중복 검사에서 고려되지는 않는다. 대신에, 도 11에서 화살표로 연결된 쌍들만이 고려되고, 중복 검사를 위해 사용되는 대응하는 후보가 동일한 모션 정보를 갖지 않는 경우에만 후보가 리스트에 추가된다. 이중 모션 정보의 다른 소스는 2Nx2N과는 상이한 파티션들과 연관된 "제2 PU"이다. 예로서, 도 12는 각각 NX2N 및 2N \times N의 경우에 대한 제2 PU를 도시한다. 현재 PU가 NX2N으로 파티셔닝되면, 위치 A_1 에서의 후보는 리스트 구성을 위해 고려되지 않는다. 실제로, 이 후보를 추가하는 것은 동일한 모션 정보를 갖는 2개의 예측 유닛으로 이어질 것이고, 이는 코딩 유닛에서 단지 하나의 PU를 가지는 경우 중복된다. 유사하게, 현재 PU가 2N \times N으로 파티셔닝될 때, 위치 B_1 은 고려되지 않는다.

2.2.1.3 시간 후보 도출

[0080] 이 단계에서, 하나의 후보만이 리스트에 추가된다. 상세하게는, 이 시간 병합 후보의 도출에서, 주어진 참조 픽처 리스트 내의 현재 픽처와 가장 작은 POC 차이를 갖는 픽처에 속하는 병치된 PU에 기초하여 스케일링된 모션 벡터가 도출된다. 병치된 PU의 도출에 사용될 참조 픽처 리스트는 슬라이스 헤더에서 명시적으로 시그널링된다. 시간 병합 후보에 대한 스케일링된 모션 벡터는, POC 거리들 tb 및 td를 사용하여 병치된 PU의 모션 벡터로부터 스케일링되는, 도 13에서 점선으로 예시된 바와 같이 획득되고, 여기서 tb는 현재 픽처의 참조 픽처와 현재 픽처 사이의 POC 차이인 것으로 정의되고, td는 병치된 픽처의 참조 픽처와 병치된 픽처 사이의 POC 차이인 것으로 정의된다. 시간 병합 후보의 참조 픽처 인덱스는 제로와 동일하게 설정된다. 스케일링 프로세스의 실제적인 실현은 HEVC 사양[1]에서 설명된다. B-슬라이스의 경우, 2개의 모션 벡터- 하나는 참조 픽처 리스트 0에 대한 것이고 다른 하나는 참조 픽처 리스트 1에 대한 것임 -가 획득되고 조합되어 양방향 예측 병합 후보를 구성한다. 시간 병합 후보에 대한 모션 벡터 스케일링의 예시.

[0081] 참조 프레임에 속하는 병치된 PU(Y)에서, 시간 후보에 대한 위치는 도 14에 도시된 바와 같이 후보들 C_0 과 C_1 사이에서 선택된다. 위치 C_0 에서의 PU가 사용가능하지 않거나, 인트라 코딩되거나, 또는 현재 CTU의 외부에 있으

면, 위치 C_1 가 사용된다. 그렇지 않으면, 위치 C_0 가 시간 병합 후보의 도출에서 사용된다.

2.2.1.4 추가 후보 삽입

공간-시간 병합 후보를 이외에, 2개의 추가적인 타입의 병합 후보가 있다: 조합된 양방향 예측 병합 후보 및 제로 병합 후보. 조합된 양방향 예측 병합 후보들은 공간-시간 병합 후보들을 이용함으로써 생성된다. 조합된 양방향 예측 병합 후보는 B-슬라이스에 대해서만 사용된다. 조합된 양방향 예측 후보들은 초기 후보의 제1 참조 픽처 리스트 모션 파라미터들을 다른 후보의 제2 참조 픽처 리스트 모션 파라미터들과 조합함으로써 생성된다. 이들 2개의 튜플이 상이한 모션 가설을 제공하는 경우, 이들은 새로운 양방향 예측 후보를 형성할 것이다. 일 예로서, 도 15는 mvL0 및 refIdxL0 또는 mvL1 및 refIdxL1을 갖는, 원본 리스트(좌측) 내의 2개의 후보가 최종 리스트(우측)에 추가된 조합된 양방향 예측 병합 후보를 생성하는 데 사용되는 경우를 도시한다. 이러한 추가적인 병합 후보들을 생성하는 것으로 고려되는 조합들에 관련한 다수의 규칙들이 존재한다.

병합 후보들 리스트 내의 나머지 엔트리들을 채우기 위해 제로 모션 후보들이 삽입되며, 따라서 MaxNumMergeCand 용량에 도달한다. 이러한 후보들은 제로 공간 변위 및 제로로부터 시작하고 새로운 제로 모션 후보가 리스트에 추가될 때마다 증가하는 참조 픽처 인덱스를 갖는다. 이러한 후보들에 의해 사용되는 참조 프레임들의 수는 단방향 및 양방향 예측을 위해 각각 1개 및 2개이다. 마지막으로, 이러한 후보들에 대해 중복 검사가 수행되지 않는다.

2.2.1.5 병렬 처리를 위한 모션 추정 영역

인코딩 프로세스의 속도를 높이기 위해, 모션 추정이 병렬로 수행될 수 있고, 그에 의해 주어진 영역 내의 모든 예측 유닛들에 대한 모션 벡터들이 동시에 도출된다. 공간적 이웃으로부터의 병합 후보들의 도출은 병렬 처리를 방해할 수 있는데, 그 이유는 하나의 예측 유닛이 그의 연관된 모션 추정이 완료될 때까지 인접한 PU로부터 모션 파라미터들을 도출할 수 없기 때문이다. 코딩 효율과 처리 레이턴시 사이의 트레이드-오프를 완화하기 위해, HEVC는 "log2_parallel_merge_level_minus2" 선택적 엘리먼트를 사용하여 픽처 파라미터 세트에서 그 크기가 시그널링되는 모션 추정 영역(MER)을 정의한다. MER이 정의될 때, 동일한 영역에 속하는 병합 후보들은 사용 불가능한 것으로 표시되고, 따라서 리스트 구성에서 고려되지 않는다.

7.3.2.3 픽처 파라미터 세트 RBSP 선택

7.3.2.3.1 일반 픽처 파라미터 세트 RBSP 선택

pic_parameter_set_rbsp() {	디스크립터
pps_pic_parameter_set_id	ue(v)
pps_seq_parameter_set_id	ue(v)
dependent_slice_segments_enabled_flag	u(1)
...	
pps_scaling_list_data_present_flag	u(1)
if(pps_scaling_list_data_present_flag)	
scaling_list_data()	
lists_modification_present_flag	u(1)
log2_parallel_merge_level_minus2	ue(v)
slice_segment_header_extension_present_flag	u(1)
pps_extension_present_flag	u(1)
...	
rbsp_trailing_bits()	
}	

`log2_parallel_merge_level_minus2 + 2`는, 8.5.3.2.2 절에 명시된 병합 모드에 대한 루마 모션 벡터들에 대한 도출 프로세스 및 8.5.3.2.3 절에 명시된 공간 병합 후보들에 대한 도출 프로세스에서 사용되는, 변수 `Log2ParMrgLevel`의 값을 지정한다. `log2_parallel_merge_level_minus2`의 값은 0 내지 `CtbLog2SizeY - 2`(경계값 포함)의 범위에 있을 것이다.

변수 `Log2ParMrgLevel`은 다음과 같이 도출된다:

[0092] $\text{Log2ParMrgLevel} = \text{log2_parallel_merge_level_minus2} + 2$ (7-37)

[0093] 주 3 - Log2ParMrgLevel의 값은 병합 후보 리스트들의 병렬 도출의 내장 능력을 나타낸다는 점에 유의한다. 예를 들어, Log2ParMrgLevel이 6과 같을 때, 64x64 블록에 포함된 모든 예측 유닛(PU)들 및 코딩 유닛(CU)들에 대한 병합 후보 리스트들이 병렬로 도출될 수 있다.

[0094] 2.2.2 AMVP 모드에서의 모션 벡터 예측

[0095] 모션 벡터 예측은 모션 파라미터의 명시적 송신을 위해 사용되는 이웃 PU와의 모션 벡터의 공간-시간 상관을 이용한다. 이것은 먼저 좌측, 상위의 시간적으로 이웃하는 PU 위치들에서 가용성을 검사하고, 중복 후보들을 제거하고, 제로 벡터를 추가하여 후보 리스트가 일정한 길이가 되게 함으로써 모션 벡터 후보 리스트를 구성한다. 이어서, 인코더는 후보 리스트로부터 최상의 예측자를 선택하고 선택된 후보를 나타내는 대응하는 인덱스를 송신할 수 있다. 병합 인덱스 시그널링과 유사하게, 최상의 모션 벡터 후보의 인덱스는 절단 단항(truncated unary)을 사용하여 인코딩된다. 이 경우에 인코딩될 최대값은 2이다(예를 들어, 도 2 내지 도 8). 다음 섹션들에서, 모션 벡터 예측 후보의 도출 프로세스에 관련한 세부사항들이 제공된다.

[0096] 2.2.2.1 모션 벡터 예측 후보들의 도출

[0097] 도 16은 모션 벡터 예측 후보에 대한 도출 프로세스를 요약한다.

[0098] 모션 벡터 예측에서, 2가지 타입의 모션 벡터 후보가 고려된다: 공간 모션 벡터 후보 및 시간 모션 벡터 후보. 공간 모션 벡터 후보 도출의 경우, 2개의 모션 벡터 후보가 결국 도 11에 도시된 바와 같이 5개의 상이한 위치에 위치하는 각각의 PU의 모션 벡터에 기초하여 도출된다.

[0099] 시간 모션 벡터 후보 도출에 대해, 2개의 상이한 병치된 위치에 기초하여 도출되는 2개의 후보로부터 하나의 모션 벡터 후보가 선택된다. 공간-시간 후보들의 제1 리스트가 형성된 이후에, 리스트 내의 이중화된 모션 벡터 후보들이 제거된다. 잠재적 후보들의 수가 2보다 큰 경우, 연관된 참조 픽처 리스트 내의 참조 픽처 인덱스가 1보다 큰 모션 벡터 후보들이 리스트로부터 제거된다. 공간-시간 모션 벡터 후보들의 수가 2보다 작으면, 추가적인 제로 모션 벡터 후보들이 리스트에 추가된다.

[0100] 2.2.2.2 공간 모션 벡터 후보들

[0101] 공간 모션 벡터 후보들의 도출에서, 최대 2개의 후보들이 5개의 잠재적 후보들 중에서 고려되며, 이 후보들은 도 11에 도시된 바와 같은 위치들에 위치된 PU들로부터 도출되며, 그 위치들은 모션 병합의 위치들과 동일하다. 현재 PU의 좌측 측면에 대한 도출 순서는 A_0 , A_1 및 스케일링된 A_0 , 스케일링된 A_1 로서 정의된다. 현재 PU의 상위 측면에 대한 도출 순서는 B_0 , B_1 , B_2 , 스케일링된 B_0 , 스케일링된 B_1 , 스케일링된 B_2 로서 정의된다. 따라서, 각각의 측면에 대해, 모션 벡터 후보로서 사용될 수 있는 4개의 경우가 있으며, 2개의 경우는 공간 스케일링을 사용할 필요가 없고, 2개의 경우는 공간 스케일링이 사용된다. 4개의 상이한 경우들이 다음과 같이 요약된다.

[0102] · 공간 스케일링 없음

[0103] - (1) 동일한 참조 픽처 리스트, 및 동일한 참조 픽처 인덱스(동일한 POC)

[0104] - (2) 상이한 참조 픽처 리스트, 그러나 동일한 참조 픽처(동일한 POC)

[0105] · 공간 스케일링

[0106] - (3) 동일한 참조 픽처 리스트, 그러나 상이한 참조 픽처(상이한 POC)

[0107] - (4) 상이한 참조 픽처 리스트, 및 상이한 참조 픽처(상이한 POC)

[0108] 공간 스케일링이 없는 경우들이 먼저 검사되고 이어서 공간 스케일링이 후속된다. 참조 픽처 리스트에 관계없이 이웃 PU의 참조 픽처와 현재 PU의 참조 픽처 사이에 POC가 상이할 때 공간 스케일링이 고려된다. 좌측 후보들의 모든 PU들이 사용가능하지 않거나 인트라 코딩되는 경우, 좌측 및 상위 MV 후보들의 병렬 도출을 돋기 위해 상위 모션 벡터에 대한 스케일링이 허용된다. 그렇지 않으면, 상위 모션 벡터에 대해 공간 스케일링이 허용되지 않는다.

[0109] 공간 스케일링 프로세스에서, 이웃 PU의 모션 벡터는 도 17에 도시된 바와 같이 시간 스케일링에 대한 것과 유사한 방식으로 스케일링된다. 주된 차이는 현재 PU의 참조 픽처 리스트 및 인덱스가 입력으로서 주어진다는 것이다; 실제 스케일링 프로세스는 시간 스케일링의 것과 동일하다.

[0110] 2.2.2.3 시간 모션 벡터 후보들

참조 픽처 인덱스 도출 이외에, 시간 병합 후보들의 도출을 위한 모든 프로세스들은 공간 모션 벡터 후보들의 도출(예를 들어, 도 6 참조)과 동일하다. 참조 픽처 인덱스는 디코더에 시그널링된다.

[0112] 2.2.2.4 AMVP 정보의 시그널링

AMVP 모드의 경우, 4개의 부분, 즉 예측 방향, 참조 인덱스, MVD 및 mv 예측자 후보 인덱스가 비트스트림에서 시그널링될 수 있다.

[0114] 선택스 테이블

prediction_unit(x0, y0, nPbW, nPbH) {	디스크립터
if(cu_skip_flag[x0][y0]) {	
if(MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
} else { /* MODE_INTER */	
merge_flag[x0][y0]	ae(v)
if(merge_flag[x0][y0]) {	
if(MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
} else {	
if(slice_type == B)	
inter_pred_idc[x0][y0]	ae(v)
if(inter_pred_idc[x0][y0] != PRED_L1) {	
if(num_ref_idx_l0_active_minus1 > 0)	
ref_idx_l0[x0][y0]	ae(v)
mvd_coding(x0, y0, 0)	
mvp_l0_flag[x0][y0]	ae(v)
}	
if(inter_pred_idc[x0][y0] != PRED_L0) {	
if(num_ref_idx_l1_active_minus1 > 0)	
ref_idx_l1[x0][y0]	ae(v)
if(mvd_l1_zero_flag && inter_pred_idc[x0][y0]	
== PRED_BI) {	
MvdL1[x0][y0][0] = 0	
MvdL1[x0][y0][1] = 0	
} else	
mvd_coding(x0, y0, 1)	
mvp_l1_flag[x0][y0]	ae(v)
}	
}	
}	

[0115]

[0116]

7.3.8.9 모션 벡터 차이 선택스

코드	디스크립터
abs_mvd_greater0_flag[0]	ae(v)
abs_mvd_greater0_flag[1]	ae(v)
if(abs_mvd_greater0_flag[0])	
abs_mvd_greater1_flag[0]	ae(v)
if(abs_mvd_greater0_flag[1])	
abs_mvd_greater1_flag[1]	ae(v)
if(abs_mvd_greater0_flag[0]) {	
if(abs_mvd_greater1_flag[0])	
abs_mvd_minus2[0]	ae(v)
mvd_sign_flag[0]	ae(v)
}	
if(abs_mvd_greater0_flag[1]) {	
if(abs_mvd_greater1_flag[1])	
abs_mvd_minus2[1]	ae(v)
mvd_sign_flag[1]	ae(v)
}	
}	

[0117]

2.3 JEM(Joint Exploration Model)에서의 새로운 인터 예측 방법들

[0118]

2.3.1 서브-CU 기반 모션 벡터 예측

[0119]

QTBT를 갖는 JEM에서, 각각의 CU는 각각의 예측 방향에 대해 많아야 하나의 모션 파라미터 세트를 가질 수 있다. 큰 CU를 서브-CU들로 분할하고 큰 CU의 모든 서브-CU들에 대한 모션 정보를 도출함으로써 2개의 서브-CU 레벨 모션 벡터 예측 방법들이 인코더에서 고려된다. ATMVP(alternative temporal motion vector prediction) 방법은 각각의 CU가 병치된 참조 픽처에서 현재 CU보다 작은 다수의 블록들로부터 모션 정보의 다수의 세트들을 폐치할 수 있게 한다. STMVP(spatial-temporal motion vector prediction) 방법에서, 서브-CU들의 모션 벡터들은 시간 모션 벡터 예측자 및 공간 이웃 모션 벡터를 사용함으로써 재귀적으로 도출된다.

[0120]

서브-CU 모션 예측을 위한 더 정확한 모션 필드를 보존하기 위해, 참조 프레임들에 대한 모션 압축은 현재 디스에이블된다.

[0121]

2.3.1.1 대체 시간 모션 벡터 예측

[0122]

ATMVP(alternative temporal motion vector prediction) 방법에서, 모션 벡터들 TMVP(temporal motion vector prediction)은 현재 CU보다 작은 블록들로부터 (모션 벡터들 및 참조 인덱스들을 포함하는) 모션 정보의 다수의 세트를 폐치함으로써 수정된다. 도 18에 도시된 바와 같이, 서브-CU들은 정사각형 NXN 블록들(N은 디폴트로 4로 설정됨)이다.

[0123]

ATMVP는 2개의 단계들에서 CU 내의 서브-CU들의 모션 벡터들을 예측한다. 제1 단계는 소위 시간 벡터로 참조 픽처에서 대응하는 블록을 식별하는 것이다. 참조 픽처는 모션 소스 픽처라고 불린다. 제2 단계는 도 18에 도시된 바와 같이 현재 CU를 서브-CU들로 분할하고, 각각의 서브-CU에 대응하는 블록으로부터 각각의 서브-CU의 참조 인덱스들뿐만 아니라 모션 벡터들을 획득하는 것이다.

[0124]

제1 단계에서, 참조 픽처 및 대응하는 블록은 현재 CU의 공간 이웃 블록들의 모션 정보에 의해 결정된다. 이웃 블록들의 반복 스캐닝 프로세스를 피하기 위해, 현재 CU의 병합 후보 리스트 내의 제1 병합 후보가 사용된다. 제1 사용 가능한 모션 벡터 및 그의 관련 참조 인덱스는 시간 벡터 및 모션 소스 픽처에 대한 인덱스가 되도록 설정된다. 이러한 방식으로, ATMVP에서, 대응하는 블록은 TMVP와 비교하여 더 정확하게 식별될 수 있고, 대응하는 블록(때때로 병치된 블록이라고 불림)은 항상 현재 CU에 대해 하단-우측 또는 중심 위치에 있다. 일 예에서, 제1 병합 후보가 좌측 이웃 블록으로부터 온 것인 경우(즉, 도 19에서 A₁), 소스 블록 및 소스 픽처를 식별하기

위해 연관된 MV 및 참조 픽처가 이용된다.

[0126] 도 19는 소스 블록 및 소스 픽처의 식별의 예를 도시한다.

[0127] 제2 단계에서, 서브-CU의 대응하는 블록은 현재 CU의 좌표에 시간 벡터를 추가함으로써 모션 소스 픽처 내의 시간 벡터에 의해 식별된다. 각각의 서브-CU에 대하여, 그 대응하는 블록의 모션 정보(중심 샘플을 커버하는 최소 모션 그리드)는 서브-CU에 대한 모션 정보를 도출하기 위하여 사용된다. 대응하는 NXN 블록의 모션 정보가 식별된 이후에, 그것은 HEVC의 TMVP와 동일한 방식으로, 현재 서브-CU의 모션 벡터들 및 참조 인덱스들로 변환되고, 모션 스케일링 및 다른 절차들이 적용된다. 예를 들어, 디코더는 저지연 조건(즉, 현재 픽처의 모든 참조 픽처들의 POC들이 현재 픽처의 POC보다 작음)이 충족되는지를 검사하고, 가능하게는 모션 벡터 MV_x (참조 픽처 리스트 X에 대응하는 모션 벡터)를 사용하여 각각의 서브-CU에 대한 모션 벡터 MV_y (X는 0 또는 1과 같고 Y는 1-X와 같음)를 예측한다.

2.3.1.2 공간-시간 모션 벡터 예측

[0129] 이 방법에서, 서브-CU들의 모션 벡터들은, 래스터 스캔 순서에 따라, 재귀적으로 도출된다. 도 20은 이 개념을 도시한다. 4개의 4X4 서브-CU들 A, B, C, 및 D를 포함하는 8X8 CU를 고려한다. 현재 프레임 내의 이웃 4X4 블록들은 a, b, c 및 d로 표시된다.

[0130] 서브-CU A에 대한 모션 도출은 그의 2개의 공간 이웃들을 식별함으로써 시작된다. 제1 이웃은 서브-CU A 위의 N XN 블록이다(블록 c). 이 블록 c가 사용가능하지 않거나 인트라 코딩된 경우, 서브-CU A 위의 다른 NXN 블록들이 검사된다(블록 c에서 시작하여 좌측으로부터 우측으로). 제2 이웃은 서브-CU A의 좌측의 블록(블록 b)이다. 블록 b가 사용가능하지 않거나 인트라 코딩된 경우, 서브-CU A의 좌측에 있는 다른 블록들이 검사된다(상단에서 하단으로, 블록 b에서 시작). 각각의 리스트에 대해 이웃 블록들로부터 획득된 모션 정보는 주어진 리스트에 대해 제1 참조 프레임으로 스케일링된다. 다음으로, 서브 블록 A의 시간 모션 벡터 예측자(TMVP)는 HEVC에서 지정된 바와 같은 TMVP 도출의 동일한 절차를 따라 도출된다. 위치 D에서의 병치된 블록의 모션 정보가 그에 따라 폐치되고 스케일링된다. 마지막으로, 모션 정보를 검색하고 스케일링한 이후에, 모든 사용가능한 모션 벡터(최대 3)는 각각의 참조 리스트에 대해 개별적으로 평균화된다. 평균 모션 벡터는 현재 서브-CU의 모션 벡터로서 할당된다.

[0131] 도 20은 4개의 서브 블록(A-D) 및 그 이웃 블록(a-d)을 갖는 하나의 CU의 예를 도시한다.

2.3.1.3 서브-CU 모션 예측 모드 시그널링

[0132] 서브-CU 모드들은 추가적인 병합 후보들로서 인에이블되고 모드들을 시그널링하는데 요구되는 추가적인 선택스 엘리먼트가 없다. 각각의 CU의 병합 후보 리스트에 2개의 추가적인 병합 후보가 추가되어 ATMVP 모드 및 STMVP 모드를 표현한다. 시퀀스 파라미터 세트가 ATMVP 및 STMVP가 인에이블되는 것을 나타내는 경우, 최대 7개의 병합 후보가 사용된다. 추가적인 병합 후보들의 인코딩 로직은 HM 내의 병합 후보들에 대한 것과 동일하며, 이는 P 또는 B 슬라이스 내의 각각의 CU에 대해, 2개의 추가적인 병합 후보에 대해 2회의 더 많은 RD 검사가 필요함을 의미한다.

[0134] JEM에서, 병합 인덱스의 모든 빈들은 CABAC에 의해 컨텍스트 코딩된다. HEVC에서는 제1 빈만이 컨텍스트 코딩되고, 나머지 빈들은 컨텍스트 바이패스 코딩된다.

2.3.2 적응 모션 벡터 차이 분해능

[0136] HEVC에서, (PU의 모션 벡터와 예측된 모션 벡터 사이의) 모션 벡터 차이(MVD)들은 use_integer_mv_flag가 슬라이스 헤더에서 0과 동일할 때 쿼터 루마 샘플들의 유닛들로 시그널링된다. JEM에서, LAMVR(locally adaptive motion vector resolution)이 도입된다. JEM에서, MVD는 쿼터 루마 샘플, 정수 루마 샘플 또는 4 루마 샘플의 유닛으로 코딩될 수 있다. MVD 분해능은 코딩 유닛(CU) 레벨에서 제어되고, MVD 분해능 플래그들은 적어도 하나의 비-제로 MVD 성분을 갖는 각각의 CU에 대해 조건부로 시그널링된다.

[0137] 적어도 하나의 비-제로 MVD 성분을 갖는 CU에 대해, 제1 플래그는 CU에서 쿼터 루마 샘플 MV 정밀도가 사용되는지를 나타내기 위해 시그널링된다. 제1 플래그(1과 동일)가 쿼터 루마 샘플 MV 정밀도가 사용되지 않는다는 것을 나타낼 때, 정수 루마 샘플 MV 정밀도 또는 4 루마 샘플 MV 정밀도가 사용되는지를 나타내기 위해 다른 플래그가 시그널링된다.

[0138] CU의 제1 MVD 분해능 플래그가 0이거나 CU에 대해 코딩되지 않을 때(CU에서의 모든 MVD들이 0임을 의미함), 쿼

터 루마 샘플 MV 분해능이 CU에 대해 사용된다. CU가 정수 루마 샘플 MV 정밀도 또는 4 루마 샘플 MV 정밀도를 사용할 때, CU에 대한 AMVP 후보 리스트 내의 MVP들은 대응하는 정밀도로 라운딩된다.

[0139] 인코더에서, 어느 MVD 분해능이 CU에 대해 사용되어야 하는지를 결정하기 위해 CU-레벨 RD 검사들이 사용된다. 즉, CU-레벨 RD 검사는 각각의 MVD 분해능에 대해 3회 수행된다. 인코더 속도를 가속화하기 위해, 다음 인코딩 방식들이 JEM에서 적용된다:

[0140] 정상 쿼터 루마 샘플 MVD 분해능을 갖는 CU의 RD 검사 동안, 현재 CU의 모션 정보(정수 루마 샘플 정확도)가 저장된다. (라운딩 이후의) 저장된 모션 정보는 정수 루마 샘플 및 4 루마 샘플 MVD 분해능을 갖는 동일한 CU에 대한 RD 검사 동안 추가의 작은 범위 모션 벡터 정제를 위한 시작점으로서 사용되어, 시간 소모적인 모션 추정 프로세스가 3회 이중화되지 않도록 한다.

[0141] 4 루마 샘플 MVD 분해능을 갖는 CU에 대한 RD 검사가 조건부로 호출된다. CU의 경우, RD 비용 정수 루마 샘플 MVD 분해능이 쿼터 루마 샘플 MVD 분해능보다 훨씬 더 클 때, CU에 대한 4 루마 샘플 MVD 분해능의 RD 검사는 스킵된다.

2.3.3 패턴 매칭 모션 벡터 도출

[0143] PMMVD(pattern matched motion vector derivation) 모드는 FRUC(frame-rate up conversion) 기술들에 기초한 특별한 병합 모드이다. 이 모드에 의하면, 블록의 모션 정보가 시그널링되지 않고 디코더 측에서 도출된다.

[0144] FRUC 플래그는 그 병합 플래그가 참일 때 CU에 대해 시그널링된다. FRUC 플래그가 거짓일 때, 병합 인덱스가 시그널링되고 정규 병합 모드가 사용된다. FRUC 플래그가 참일 때, 블록에 대한 모션 정보를 도출하기 위해 어느 방법(양방향 매칭 또는 템플릿 매칭)이 사용되어야 하는지를 나타내기 위해 추가의 FRUC 모드 플래그가 시그널링된다.

[0145] 인코더 측에서, CU에 대해 FRUC 병합 모드를 사용하는지에 대한 결정은 정상 병합 후보에 대해 행해진 RD 비용 선택에 기초한다. 즉, 2개의 매칭 모드들(양방향 매칭 및 템플릿 매칭) 양자 모두가 RD 비용 선택을 사용하여 CU에 대해 검사된다. 최소 비용으로 이어지는 것이 다른 CU 모드들과 추가로 비교된다. FRUC 매칭 모드가 가장 효율적인 모드이면, CU에 대해 FRUC 플래그가 참으로 설정되고 관련된 매칭 모드가 사용된다.

[0146] FRUC 병합 모드에서의 모션 도출 프로세스는 2개의 단계들을 가진다. CU-레벨 모션 검색이 먼저 수행되고, 이어서 서브-CU 레벨 모션 정제가 뒤따른다. CU 레벨에서, 양방향 매칭 또는 템플릿 매칭에 기초하여 전체 CU에 대해 초기 모션 벡터가 도출된다. 먼저, MV 후보들의 리스트가 생성되고, 최소 매칭 비용을 초래하는 후보가 추가의 CU 레벨 정제를 위한 시작점으로서 선택된다. 그 후, 시작점 주위의 양방향 매칭 또는 템플릿 매칭에 기초한 로컬 검색이 수행되고, 최소 매칭 비용을 초래하는 MV가 전체 CU에 대한 MV로서 취해진다. 후속하여, 모션 정보는 도출된 CU 모션 벡터들을 시작점들로 하여 서브-CU 레벨에서 추가로 정제된다.

$$W \times H$$

[0147] 예를 들어, 다음의 도출 프로세스가 CU 모션 정보 도출에 대해 수행된다. 제1 스텝에서, 전체

$W \times H$ $M \times M$ M
CU에 대한 MV가 도출된다. 제2 스텝에서, CU는 서브-CU들로 추가로 분할된다. 의 값은 D

(16)에서와 같이 계산되고, 는 JEM에서 디폴트로 3으로 설정되는 미리 정의된 분할 깊이이다. 이어서, 각각의 서브-CU에 대한 MV가 도출된다.

$$M = \max\{4, \min\{\frac{M}{2^D}, \frac{N}{2^D}\}\} \quad (1)$$

[0149] 도 21에 도시된 바와 같이, 양방향 매칭이 사용되어 2개의 상이한 참조 픽처에서 현재 CU의 모션 채적을 따라 2개의 블록 사이의 가장 가까운 매칭을 발견함으로써 현재 CU의 모션 정보를 도출한다. 연속 모션 채적의 가정 하에, 2개의 참조 블록들을 가리키는 모션 벡터들 MV0 및 MV1은 현재 픽처와 2개의 참조 픽처들 사이의 시간 거리들, 즉, TD0 및 TD1에 비례하여야 한다. 특별한 경우로서, 현재 픽처가 시간적으로 2개의 참조 픽처 사이에 있고 현재 픽처로부터 2개의 참조 픽처까지의 시간 거리가 동일할 때, 양방향 매칭은 미리 기반 양방향 MV가 된다.

[0150] 도 22에 도시된 바와 같이, 템플릿 매칭은 현재 픽처 내의 템플릿(현재 CU의 상단 및/또는 좌측 이웃 블록들)과

참조 픽처 내의 블록(템플릿과 동일한 크기) 사이의 가장 가까운 매칭을 발견함으로써 현재 CU의 모션 정보를 도출하는데 사용될 수 있다. 전술한 FRUC 병합 모드를 제외하고, 템플릿 매칭은 AMVP 모드에도 적용된다. JEM에서, HEVC에서 행해진 바와 같이, AMVP는 2개의 후보를 갖는다. 템플릿 매칭 방법을 사용하여, 새로운 후보가 도출된다. 템플릿 매칭에 의해 새롭게 도출된 후보가 제1 기준의 AMVP 후보와 상이하면, 이는 AMVP 후보 리스트의 바로 시작점에 삽입되고, 그 다음으로, 리스트 크기는 2로 설정된다(제2 기준의 AMVP 후보를 제거함을 의미). AMVP 모드에 적용될 때, CU 레벨 검색만이 적용된다.

[0151] 2.3.3.1 CU 레벨 MV 후보 세트

[0152] CU 레벨에서의 MV 후보 세트는 다음으로 구성된다:

[0153] (i) 현재 CU가 AMVP 모드에 있는 경우의 원래 AMVP 후보들

[0154] (ii) 모든 병합 후보들,

[0155] (iii) 보간된 MV 필드 내의 몇몇 MV들

[0156] (iv) 상단 및 좌측 이웃 모션 벡터

[0157] 양방향 매칭을 사용할 때, 병합 후보의 각각의 유효 MV는 양방향 매칭을 가정하여 MV 쌍을 생성하기 위한 입력으로서 사용된다. 예를 들어, 병합 후보의 하나의 유효 MV는 참조 리스트 A에서 (MVa, refa)이다. 이어서, 그의 쌍을 이룬 양방향 MV의 참조 픽처 refb는, refa 및 refb가 현재 픽처의 시간적으로 상이한 측면들에 있도록, 다른 참조 리스트 B에서 발견된다. 이러한 refb가 참조 리스트 B에서 사용가능하지 않다면, refb는 refa과는 상이한 참조로서 결정되고, 현재 픽처까지의 그 시간 거리는 리스트 B에서의 최소 거리이다. refb가 결정된 이후, MVb는 현재 픽처와 refa, refb 사이의 시간 거리에 기초하여 MVa를 스케일링함으로써 도출된다.

[0158] 보간된 MV 필드로부터의 4개의 MV가 또한 CU 레벨 후보 리스트에 추가된다. 보다 구체적으로, 현재 CU의 위치 (0, 0), (W/2, 0), (0, H/2) 및 (W/2, H/2)에서의 보간된 MV들이 추가된다.

[0159] FRUC가 AMVP 모드에서 적용될 때, 원래의 AMVP 후보들이 또한 CU 레벨 MV 후보 세트에 추가된다.

[0160] CU 레벨에서, AMVP CU들에 대한 최대 15개의 MV들 및 병합 CU들에 대한 최대 13개의 MV들이 후보 리스트에 추가된다.

[0161] 2.3.3.2 서브-CU 레벨 MV 후보 세트

[0162] 서브-CU 레벨에서의 MV 후보 세트는 다음으로 구성된다:

[0163] (i) CU-레벨 검색으로부터 결정된 MV,

[0164] (ii) 상단, 좌측, 상단-좌측 및 상단-우측 이웃 MV들,

[0165] (iii) 참조 픽처들로부터의 병치된 MV들의 스케일링된 베전들,

[0166] (iv) 최대 4개의 ATMVP 후보,

[0167] (v) 최대 4개의 STMVP 후보

[0168] 참조 픽처들로부터의 스케일링된 MV들은 다음과 같이 도출된다. 양자 모두의 리스트 내의 모든 참조 픽처는 순회된다. 참조 픽처에서의 서브-CU의 병치된 위치에서의 MV들은 시작 CU-레벨 MV의 참조에 대해 스케일링된다.

[0169] ATMVP 및 STMVP 후보들은 4개의 첫 번째 후보들로 제한된다.

[0170] 서브-CU 레벨에서, 최대 17개의 MV들이 후보 리스트에 추가된다.

[0171] 2.3.3.3 보간된 MV 필드의 생성

[0172] 프레임을 코딩하기 전에, 단방향 ME에 기초하여 전체 픽처에 대해 보간된 모션 필드가 생성된다. 이어서, 모션 필드는 나중에 CU 레벨 또는 서브-CU 레벨 MV 후보로서 사용될 수 있다.

[0173] 먼저, 양자 모두의 참조 리스트 내의 각 참조 픽처의 모션 필드는 4X4 블록 레벨에서 순회된다. 각각의 4X4 블록에 대해, 모션이 현재 픽처 내의 4X4 블록을 통과하는 블록과 연관되고(도 23에 도시된 바와 같이), 블록이 어떠한 보간된 모션도 할당받지 않은 경우, 참조 블록의 모션은 시간 거리 TD0 및 TD1에 따라 현재 픽처에 대해 스케일링되고(HEVC에서 TMVP의 MV 스케일링의 방식과 동일한 방식), 스케일링된 모션은 현재 프레임 내의

블록에 할당된다. 스케일링된 MV가 4X4 블록에 할당되지 않으면, 블록의 모션은 보간된 모션 필드에서 사용 불 가능한 것으로 표시된다.

[0174] 2.3.3.4 보간 및 매칭 비용

모션 벡터가 분수 샘플 위치를 가리킬 때, 모션 보상된 보간이 필요하다. 복잡도를 감소시키기 위해, 정규 8-탭 HEVC 보간 대신에 이중 선형 보간이 양방향 매칭 및 템플릿 매칭 둘 다를 위해 사용된다.

매칭 비용의 계산은 상이한 단계들에서 다소 상이하다. CU 레벨에서 후보 세트로부터 후보를 선택할 때, 매칭 비용은 양방향 매칭 또는 템플릿 매칭의 SAD(absolute sum difference)이다. 시작 MV가 결정된 이후에, 서브-

C

CU 레벨 검색에서의 양방향 매칭의 매칭 비용 w 는 다음과 같이 계산된다:

$$C = SAD + w \cdot (|MV_x - MV_x^s| + |MV_y - MV_y^s|) \quad (2)$$

w MV MV^s

[0178] 여기서 w 은 경험적으로 4로 설정된 가중 인자이고, MV 및 MV^s 는 각각 현재 MV 및 시작 MV를 나타낸다. SAD는 여전히 서브-CU 레벨 검색에서의 템플릿 매칭의 매칭 비용으로서 사용된다.

[0179] FRUC 모드에서, MV는 루마 샘플들만을 사용하여 도출된다. 도출된 모션은 MC 인터 예측을 위해 루마와 크로마 양자 모두에 사용될 것이다. MV가 결정된 이후에, 최종 MC는 루마에 대한 8-탭 보간 필터 및 크로마에 대한 4-탭 보간 필터를 사용하여 수행된다.

[0180] 2.3.3.5 MV 정제

[0181] MV 정제는 양방향 매칭 비용 또는 템플릿 매칭 비용의 기준을 갖는 패턴 기반 MV 검색이다. JEM에서, 2개의 검색 패턴- UCBDS(unrestricted center-biased diamond search) 및 CU 레벨 및 서브-CU 레벨 각각에서의 MV 정제를 위한 적응적 교차 검색 -이 지원된다. CU 및 서브-CU 레벨 MV 정제 둘 다에 대해, MV는 쿼터 루마 샘플 MV 정확도에서 직접 검색되고, 이것에 이어서 1/8 루마 샘플 MV 정제가 뒤따른다. CU 및 서브-CU 단계에 대한 MV 정제의 검색 범위는 8 루마 샘플들과 동일하게 설정된다.

[0182] 2.3.3.6 템플릿 매칭 FRUC 병합 모드에서의 예측 방향의 선택

[0183] 양방향 매칭 병합 모드에서, CU의 모션 정보가 2개의 상이한 참조 픽처들에서의 현재 CU의 모션 궤적을 따라 2개의 블록들 사이의 가장 가까운 매칭에 기초하여 도출되기 때문에 양방향 예측이 항상 적용된다. 템플릿 매칭 병합 모드에 대한 이러한 제한은 없다. 템플릿 매칭 병합 모드에서, 인코더는 list0으로부터의 단방향 예측, list1로부터의 단방향 예측, 또는 CU에 대한 양방향 예측 중에서 선택할 수 있다. 선택은 다음과 같은 템플릿 매칭 비용에 기초한다:

만약, $costBi \leq factor * \min(cost0, cost1)$ 이면

양방향 예측 사용;

그렇지 않고, $cost0 \leq cost1$ 이면

list0으로부터의 단방향 예측 사용;

그 이외의 경우,

list1로부터의 단방향 예측 사용;

[0189] 여기서, cost0은 list0 템플릿 매칭의 SAD이고, cost1은 list1 템플릿 매칭의 SAD이고, costBi는 양방향 예측 템플릿 매칭의 SAD이다. 예를 들어, 인자의 값이 1.25와 같은 경우, 이는 선택 프로세스가 양방향 예측을 향해 바이어싱된다는 것을 의미한다.

[0191] 인터 예측 방향 선택은 CU-레벨 템플릿 매칭 프로세스에만 적용된다.

[0192] 2.3.4 디코더측 모션 벡터 정제

[0193] 양방향 예측 동작에서, 하나의 블록 영역의 예측에 대해, 각각 list0의 모션 벡터(MV) 및 list1의 MV를 사용하여 형성된 2개의 예측 블록이 조합되어 단일 예측 신호를 형성한다. DMVR(decoder-side motion vector

refinement) 방법에서, 양방향 예측의 2개의 모션 벡터는 양방향 템플릿 매칭 프로세스에 의해 추가로 정제된다. 양방향 템플릿 매칭은 추가적인 모션 정보의 송신 없이 정제된 MV를 획득하기 위해 참조 픽처들 내의 재구성 샘플들과 양방향 템플릿 사이의 왜곡 기반 검색을 수행하기 위해 디코더에 적용된다.

[0194] DMVR에서, 양방향 템플릿은 도 23에 도시된 바와 같이, list0의 초기 MV0 및 list1의 MV1로부터, 각각, 2개의 예측 블록의 가중된 조합(즉, 평균)으로서 생성된다. 템플릿 매칭 동작은 생성된 템플릿과 참조 픽처에서의 (초기 예측 블록 주위의) 샘플 영역 사이의 비용 측정치들을 계산하는 것으로 구성된다. 2개의 참조 픽처 각각에 대해, 최소 템플릿 비용을 산출하는 MV가 원래의 것을 대체하기 위해 그 리스트의 업데이트된 MV로서 고려된다. JEM에서, 9개의 MV 후보가 각각의 리스트에 대해 검색된다. 9개의 MV 후보는 원래의 MV 및 수평 또는 수직 방향 중 어느 하나 또는 양자 모두에서 원래의 MV에 대한 하나의 루마 샘플 오프셋을 갖는 8개의 주위의 MV를 포함한다. 마지막으로, 2개의 새로운 MV들, 즉, 도 24에 도시된 바와 같은 MV0' 및 MV1'이 최종 양방향 예측 결과들을 생성하기 위해 사용된다. SAD(sum of absolute differences)가 비용 측정치로서 사용된다.

[0195] DMVR은, 추가적인 선택스 엘리먼트들의 송신 없이, 과거의 참조 픽처로부터의 하나의 MV와 장래의 참조 픽처로부터의 다른 MV에 의한 양방향 예측의 병합 모드에 적용된다. JEM에서, LIC, 아핀 모션, FRUC, 또는 서브-CU 병합 후보가 CU에 대해 인에이블될 때, DMVR은 적용되지 않는다.

2.3.5 양방향 매칭 정제를 사용한 병합/스킵 모드

[0197] 사용가능한 후보들의 수가 19의 최대 후보 크기에 도달할 때까지 중복 검사와 함께 공간 이웃 및 시간 이웃 블록들의 모션 벡터들 및 참조 인덱스들을 후보 리스트에 삽입함으로써 병합 후보 리스트가 먼저 구성된다. 병합/스킵 모드에 대한 병합 후보 리스트는 미리 정의된 삽입 순서에 따라 HEVC(조합된 후보들 및 제로 후보들)에서 사용되는 바와 같이 공간 후보들, 시간 후보들, 아핀 후보들, ATMVP(advanced temporal MVP) 후보, STMVP(spatial temporal MVP) 후보 및 추가 후보들을 삽입함으로써 구성된다(도 11).

[0198] - 블록들 1-4에 대한 공간 후보들.

[0199] - 블록들 1-4에 대한 외삽된 아핀 후보들.

[0200] - ATMVP.

[0201] - STMVP.

[0202] - 가상 아핀 후보.

[0203] - 공간 후보(블록 5)(사용가능한 후보들의 수가 6보다 작을 때에만 사용됨).

[0204] - 외삽된 아핀 후보(블록 5)

[0205] - 시간 후보(HEVC에서와 같이 도출됨).

[0206] - 외삽된 아핀 후보가 뒤따르는 비인접 공간 후보들(도 25에 도시된 바와 같이, 블록들 6 내지 49).

[0207] - 조합된 후보들.

[0208] - 제로 후보들

[0209] IC 플래그들은 또한 STMVP 및 아핀을 제외한 병합 후보들로부터 상속된다는 점에 유의한다. 또한, 처음 4개의 공간 후보에 대해, 양방향 예측 후보들은 단방향 예측을 갖는 후보들 전에 삽입된다.

[0210] [8]에서, 현재 블록과 연결되지 않은 블록들이 액세스될 수 있다. 비인접 블록이 비-인트라 모드로 코딩되면, 연관된 모션 정보가 추가적인 병합 후보로서 추가될 수 있다.

3. 본 명세서에 개시된 실시예들에 의해 해결되는 문제점들의 예들

[0212] 현재 HEVC 설계는 모션 정보를 더 양호하게 코딩하기 위해 (현재 블록 옆의) 그의 이웃 블록들의 현재 블록의 상관을 취할 수 있다. 그러나, 이웃 블록들은 상이한 모션 케이스들을 갖는 상이한 객체들에 대응하는 것이 가능하다. 이 경우, 그의 이웃 블록들로부터의 예측은 효율적이지 않다.

[0213] 비인접 블록들의 모션 정보로부터의 예측은 모든 모션 정보(전형적으로 4x4 레벨)를 캐시에 저장하는 것을 막기로 추가적인 코딩 이득을 발생시킬 수 있고, 이러한 저장은 하드웨어 구현에 대한 복잡도를 상당히 증가시킨다.

4. 기술적 해결책들의 일부 예들

[0215] 현재 개시된 기술의 실시예들은 기존 구현들의 단점들을 극복하고, 그에 의해 보다 높은 코딩 효율들을 갖는 비디오 코딩을 제공한다. 본 문서의 설명에서, LUT는 테이블, 리스트, 어레이 또는 인덱싱된 엔트리들의 다른 배열일 수 있다.

[0216] 기존 구현들의 단점들을 극복하기 위해, 블록의 모션 정보를 예측하기 위해 저장된 적어도 하나의 모션 후보를 갖는 하나 이상의 테이블(예를 들어, 루업 테이블)을 사용하는 LUT-기반 모션 벡터 예측 기술들이 보다 높은 코딩 효율들을 갖는 비디오 코딩을 제공하기 위해 다양한 실시예들에서 구현될 수 있다. 루업 테이블은 블록의 모션 정보를 예측하기 위한 모션 후보들을 포함하도록 사용될 수 있는 테이블의 예이고, 다른 구현들이 또한 가능하다. 각각의 LUT는 각각이 대응하는 모션 정보와 연관되어 있는 하나 이상의 모션 후보를 포함할 수 있다. 모션 후보의 모션 정보는 예측 방향, 참조 인덱스들/픽처들, 모션 벡터들, LIC 플래그들, 아핀 플래그들, 모션 벡터 도출(MVD) 정밀도들, 및/또는 MVD 값들 중 일부 또는 전부를 포함할 수 있다. 모션 정보는 모션 정보가 어느 것으로부터 오는지를 나타내기 위해 블록 위치 정보를 추가로 포함할 수 있다.

[0217] 기존 및 장래의 비디오 코딩 표준 양자 모두를 향상시킬 수 있는, 개시된 기술에 기초한 LUT-기반 모션 벡터 예측이 다양한 구현들에 대해 설명된 이하의 예들에서 설명된다. LUT들은 인코딩/디코딩 프로세스가 이력 데이터(예를 들어, 처리된 블록들)에 기초하여 수행될 수 있게 하기 때문에, LUT-기반 모션 벡터 예측은 또한 HMVP(History-based Motion Vector Prediction) 방법이라고도 지칭될 수 있다. LUT-기반 모션 벡터 예측 방법에서, 이전에 코딩된 블록들로부터의 모션 정보를 갖는 하나 또는 다수의 테이블이 인코딩/디코딩 프로세스 동안 유지된다. LUT들에 저장된 이러한 모션 후보들은 HMVP 후보들로 명명된다. 하나의 블록의 인코딩/디코딩 동안, LUT들에서의 연관된 모션 정보가 모션 후보 리스트들(예를 들어, 병합/AMVP 후보 리스트들)에 추가될 수 있고, 하나의 블록을 인코딩/디코딩한 이후에, LUT들이 업데이트될 수 있다. 업데이트된 LUT는 이어서 후속 블록을 코딩하는 데 사용된다. 따라서, LUT들에서의 모션 후보들의 업데이트는 블록들의 인코딩/디코딩 순서에 기초한다. 아래의 예들은 일반적인 개념들을 설명하기 위한 예들로서 고려되어야 한다.

[0218] 아래의 예들은 일반적인 개념들을 설명하기 위한 예들로서 고려되어야 한다. 이러한 예들은 좁은 방식으로 해석되지 않아야 한다. 또한, 이러한 예들은 임의의 방식으로 조합될 수 있다.

[0219] 일부 실시예들은 블록의 모션 정보를 예측하기 위해 적어도 하나의 모션 후보가 저장된 하나 이상의 루업 테이블을 사용할 수 있다. 실시예들은 루업 테이블에 저장된 모션 정보 세트를 나타내기 위해 모션 후보를 사용할 수 있다. 종래의 AMVP 또는 병합 모드들에 대하여, 실시예들은 모션 정보를 저장하기 위해 AMVP 또는 병합 후보들을 사용할 수도 있다.

[0220] 아래의 예들은 일반적인 개념들을 설명한다.

[0221] **루업 테이블의 예들**

[0222] 각각의 루업 테이블은 하나 이상의 모션 후보를 포함할 수 있고 여기서 각각의 후보는 그 모션 정보와 연관된다.

[0223] **LUT들의 선택**

[0224] 예 B1: 블록을 코딩하기 위해, 하나의 루업 테이블로부터의 모션 후보들의 일부 또는 전부가 순서대로 검사될 수 있다. 블록을 코딩하는 동안 하나의 모션 후보가 검사될 때, 이는 모션 후보 리스트(예를 들어, AMVP, 병합 후보 리스트들)에 추가될 수 있다.

[0225] **루업 테이블의 사용**

[0226] 검사될 루업 테이블 내의 모션 후보들의 총 수는 미리 정의될 수 있다.

[0227] a. 이는 코딩된 정보, 블록 크기, 블록 형상 등에 추가로 의존할 수 있다.

[0228] 1. 예를 들어, AMVP 모드에 대해, m 모션 후보들만이 검사될 수 있는 반면, 병합 모드에 대해, n 모션 후보들이 검사될 수 있다(예를 들어, $m=2$, $n=44$).

[0229] b. 일 예에서, 검사될 모션 후보들의 총 수는 비디오 파라미터 세트(VPS), 시퀀스 파라미터 세트(SPS), 픽처 파라미터 세트(PPS), 슬라이스 헤더, 타일 헤더, 코딩 트리 유닛(CTU), 코딩 트리 블록(CTB), 코딩 유닛(CU) 또는 예측 유닛(PU), 다수의 CTU/CTB/CU/PU들을 커버하는 영역에서 시그널링된다.

[0230] 예 C1: 루업 테이블에 포함된 모션 후보(들)는 블록에 의해 직접 상속될 수 있다.

[0231] a. 이들은 병합 모드 코딩을 위해 사용될 수 있는데, 즉, 모션 후보들이 병합 후보 리스트 도출 프로세스에서 검사될 수 있다.

[0232] b. 이들은 아핀 병합 모드 코딩에 사용될 수 있다.

[0233] i. 루업 테이블 내의 모션 후보는 그 아핀 플래그가 1인 경우에 아핀 병합 후보로서 추가될 수 있다.

[0234] c. 루업 테이블 내의 모션 후보들의 검사는 다음과 같은 경우에 가능해질 수 있다:

[0235] i. TMVP 후보를 삽입한 이후에 병합 후보 리스트가 가득 차 있지 않은 경우;

[0236] ii. 공간 병합 후보 도출을 위해 특정 공간 이웃 블록을 검사한 이후에 병합 후보 리스트는 가득 차 있지 않은 경우;

[0237] iii. 모든 공간 병합 후보들 이후에 병합 후보 리스트가 가득 차 있지 않은 경우;

[0238] iv. 조합된 양방향 예측 병합 후보들 이후에 병합 후보 리스트가 가득 차 있지 않은 경우;

[0239] v. 다른 코딩 방법들(예를 들어, HEVC 설계, 또는 JEM 설계의 병합 도출 프로세스)로부터 병합 후보 리스트에 넣어진 공간 또는 시간(예를 들어, 인접 공간 및 비인접 공간, TMVP, STMVP, ATMVP 등을 포함함) 병합 후보들의 수가 최대 허용 병합 후보들에서 주어진 임계값을 뺀 값보다 더 작은 경우.

[0240] 1. 일 예에서, 임계값은 1 또는 0으로 설정된다.

[0241] 2. 대안적으로, 임계값은 SPS/PPS/시퀀스, 꾹처, 슬라이스 헤더/타일에서 시그널링되거나 미리 정의될 수 있다.

[0242] 3. 대안적으로, 임계값은 블록마다 적응적으로 변경될 수 있다. 예를 들어, 이는 블록 크기/블록 형상/슬라이스 타입과 같은 코딩된 블록 정보에 의존하거나 및/또는 사용가능한 공간 또는 시간 병합 후보들의 수에 의존할 수 있다.

[0243] 4. 다른 예에서, 병합 후보 리스트에 넣어진 특정 종류의 병합 후보들의 수가 최대 허용 병합 후보들에서 주어진 임계값을 뺀 값보다 작은 경우. "특정 종류의 병합 후보들"은 HEVC 또는 비인접 병합 후보들에서와 같이 공간 후보들일 수 있다.

[0244] vi. 병합 후보 리스트에 모션 후보를 추가하기 전에 프루닝이 적용될 수 있다. 본 특허 문서에 개시된 이 예 및 다른 예들의 다양한 구현들에서, 프루닝은 a) 모션 정보를 고유성을 위해 기존의 엔트리들과 비교하는 것, 또는 b) 고유하면, 모션 정보를 리스트에 추가하는 것, 또는 c) 고유하지 않으면, 그 후, c1) 모션 정보를 추가하지 않거나, 또는 c2) 모션 정보를 추가하고 매칭된 기존의 엔트리를 삭제하는 것을 포함할 수 있다. 일부 구현들에서, 테이블로부터의 모션 후보를 후보 리스트에 추가할 때 프루닝 동작이 호출되지 않는다.

[0245] 1. 일 예에서, 모션 후보는 병합 후보 리스트 내의 다른 코딩 방법들로부터의 사용가능한 공간 또는 시간(예를 들어, 인접 공간 및 비인접 공간, TMVP, STMVP, ATMVP 등을 포함함) 병합 후보들의 전부 또는 일부에 대해 프루닝될 수 있다.

[0246] 2. 모션 후보는 서브 블록 기반 모션 후보들, 예를 들어, ATMVP, STMVP에 대해 프루닝되지 않을 수 있다.

[0247] 3. 일 예에서, 현재 모션 후보가 병합 후보 리스트 내의 사용가능한 모션 후보들(현재 모션 후보 이전에 삽입됨)의 전부 또는 일부에 대해 프루닝될 수 있다.

[0248] 4. 모션 후보들에 관련된 프루닝 동작들의 수(예를 들어, 모션 후보들이 병합 리스트 내의 다른 후보들과 비교될 필요가 있는 횟수)는 사용가능한 공간 또는 시간 병합 후보들의 수에 의존할 수 있다. 예를 들어, 새로운 모션 후보를 검사할 때, 병합 리스트에 사용가능한 M개의 후보가 있는 경우, 새로운 모션 후보는 처음 K($K \leq M$) 후보들과만 비교될 수 있다. 프루닝 함수가 거짓(예를 들어, 처음 K개의 후보 중 어느 것과도 동일하지 않음)을 반환하는 경우, 새로운 모션 후보는 모든 M개의 후보와 상이한 것으로 고려되고 이는 병합 후보 리스트에 추가될 수 있다. 일 예에서, K는 $\min(K, 2)$ 로 설정된다.

[0249] 5. 일 예에서, 새롭게 첨부된 모션 후보는 병합 후보 리스트 내의 처음 N 후보와만 비교된다. 예를 들어, N= 3, 4 또는 5이다. N은 인코더로부터 디코더로 시그널링될 수 있다.

[0250] 6. 일 예에서, 검사될 새로운 모션 후보는 단지 병합 후보 리스트 내의 마지막 N 후보와 비교

된다. 예를 들어, N= 3, 4 또는 5이다. N은 인코더로부터 디코더로 시그널링될 수 있다.

[0251] 7. 일 예에서, 툭업 테이블로부터의 새로운 모션 후보와 비교될 리스트에 이전에 추가된 후보들을 선택하는 방법은 이전에 추가된 후보들이 어디로부터 도출되는지에 의존할 수 있다.

[0252] a. 일 예에서, 툭업 테이블 내의 모션 후보는 주어진 시간 및/또는 공간 이웃 블록으로부터 도출된 후보들과 비교될 수 있다.

[0253] b. 일 예에서, 툭업 테이블 내의 모션 후보들의 상이한 엔트리들은 상이한 이전에 추가된 후보들(즉, 상이한 위치들로부터 도출됨)과 비교될 수 있다.

[0254] 예 C2: 툭업 테이블에 포함되는 모션 후보(들)는 블록의 모션 정보를 코딩하기 위한 예측자로서 사용될 수 있다.

[0255] a. 이들은 AMVP 모드 코딩을 위해 사용될 수 있는데, 즉, 모션 후보들이 AMVP 후보 리스트 도출 프로세스에서 검사될 수 있다.

[0256] b. 툭업 테이블 내의 모션 후보들의 검사는 다음과 같은 경우에 가능해질 수 있다:

[0257] i. TMVP 후보를 삽입한 이후에 AMVP 후보 리스트가 가득 차 있지 않은 경우;

[0258] ii. 공간 이웃들로부터의 선택 및 프루닝 이후 TMVP 후보를 삽입하기 직전에 AMVP 후보 리스트가 가득 차 있지 않은 경우;

[0259] iii. 스케일링 없이 상위 이웃 블록들로부터 AMVP 후보가 없을 때 및/또는 스케일링 없이 좌측의 이웃 블록들로부터 AMVP 후보가 없을 때;

[0260] iv. 프루닝은 모션 후보를 AMVP 후보 리스트에 추가하기 전에 적용될 수 있다.

[0261] c. 현재 참조 픽처와 동일한 참조 픽처를 갖는 모션 후보들이 검사된다.

[0262] i. 대안적으로, 또한, 현재 참조 픽처로와 상이한 참조 픽처들을 갖는 모션 후보들이 또한 검사된다(MV 가 스케일링됨).

[0263] ii. 대안적으로, 현재 참조 픽처와 동일한 참조 픽처를 갖는 모든 모션 후보들이 먼저 검사되고, 그 후 현재 참조 픽처와 상이한 참조 픽처들을 갖는 모션 후보들이 검사된다.

[0264] iii. 대안적으로, 모션 후보들이 병합에서의 동일한 것에 후속하여 검사된다.

[0265] 예 C3: 이전에 코딩된 프레임들/슬라이스들/타일들에서의 툭업 테이블의 모션 후보들은 상이한 프레임/슬라이스/타일에서의 블록의 모션 정보를 예측하는 데 사용될 수 있다.

툭업 테이블의 업데이트

[0267] 예 D1: 모션 정보(즉, IntraBC 모드, 인터 코딩 모드)를 갖는 블록을 코딩한 이후에, 하나 또는 다수의 툭업 테이블이 업데이트될 수 있다.

[0268] 예 D2: 하나의 블록이 픽처/슬라이스/타일 경계에 위치하는 경우, 툭업 테이블들의 업데이트는 항상 허용되지 않을 수 있다.

[0269] 예 D3: 현재 LCU 행을 코딩하기 위해 LCU 행들의 모션 정보가 디스에이블될 수 있다.

[0270] 예 D4: 새로운 시간 레이어 인덱스로 슬라이스/타일을 코딩하는 시작부에서, 사용가능한 모션 후보들의 수는 0 으로 리셋될 수 있다.

[0271] 예 D5: 툭업 테이블은 동일한 시간 레이어 인덱스를 갖는 하나의 슬라이스/타일/LCU 행/슬라이스들로 연속적으로 업데이트될 수 있다.

[0272] 예 D6: 툭업 테이블 업데이트 프로세스는 상이한 절차들 내에서 호출될 수 있다.

추가의 예시적인 실시예들

[0274] HMVP 후보가 이전에 코딩된 블록의 모션 정보로서 정의되는 이력-기반 MVP(HMVP) 방법이 제안된다. 인코딩/디코딩 프로세스 동안 다수의 HMVP 후보들을 갖는 테이블이 유지된다. 새로운 슬라이스를 만날 때 테이블이 비워진다. 인터-코딩된 블록이 있을 때마다, 연관된 모션 정보가 새로운 HMVP 후보로서 테이블의 마지막 엔트리에 추

가된다. 전체 코딩 흐름이 도 31에 도시된다.

[0275] 일 예에서, 테이블 크기는 L(예를 들어, L= 16 또는 6, 또는 44)로 설정되고, 이는 최대 L개의 HMVP 후보가 테이블에 추가될 수 있다는 것을 나타낸다.

[0276] 일 실시예에서(예 11.g.i에 대응), 이전에 코딩된 블록들로부터 L개보다 많은 HMVP 후보들이 있는 경우, 테이블이 가장 최근의 이전에 코딩된 L 모션 후보들을 항상 포함하도록 FIFO(First-In-First-Out) 규칙이 적용된다. 도 32는 FIFO 규칙이 적용되어 HMVP 후보를 제거하고 제안된 방법에서 사용된 테이블에 새로운 것을 추가하는 예를 도시한다.

[0277] 또 다른 실시예에서(발명 11.g.iii에 대응), (현재 블록이 인터-코딩되고 비-아핀 모드인 것과 같은) 새로운 모션 후보를 추가할 때마다, LUT들에 동일하거나 유사한 모션 후보들이 있는지를 식별하기 위해 중복 검사 프로세스가 먼저 적용된다.

[0278] 일부 예들은 다음과 같이 도시된다:

[0279] 도 33a는 새로운 모션 후보를 추가하기 전에 LUT가 가득 찬 경우의 예를 도시한다.

[0280] 도 33b는 새로운 모션 후보를 추가하기 전에 LUT가 가득 차 있지 않을 때의 예를 도시한다.

[0281] 도 33a 및 도 33b는 함께 (하나의 중복 모션 후보가 제거된) 중복 제거 기반 LUT 업데이트 방법의 일 예를 도시한다.

[0282] 도 34a 및 도 34b는 중복 제거 기반 LUT 업데이트 방법의 2개의 경우에 대한 예시적인 구현을 도시한다(다수의 중복 모션 후보가 제거됨, 도면들에서 2개의 후보).

[0283] 도 34a는 새로운 모션 후보를 추가하기 전에 LUT가 가득 찼을 때의 예시적인 경우를 도시한다.

[0284] 도 34b는 새로운 모션 후보를 추가하기 전에 LUT가 가득 차지 않은 경우의 예시적인 경우를 도시한다.

[0285] 병합 후보 리스트 구성 프로세스에서 HMVP 후보들이 사용될 수 있다. 테이블에서의 마지막 엔트리로부터 첫 번째 엔트리(또는 마지막 K0 HMVP, 예를 들어, K0는 16 또는 6과 같음)까지의 모든 HMVP 후보들은 TMVP 후보 이후에 삽입된다. 프루닝은 HMVP 후보들에 적용된다. 사용 가능한 병합 후보들의 총 수가 시그널링된 최대 허용 병합 후보들에 도달하면, 병합 후보 리스트 구성 프로세스가 종료된다. 대안적으로, 추가된 모션 후보들의 총 수가 주어진 값에 도달하면, LUT들로부터의 모션 후보들의 폐치가 종료된다.

[0286] 유사하게, HMVP 후보들은 또한 AMVP 후보 리스트 구성 프로세스에서 사용될 수 있다. 테이블에서의 마지막 K1 HMVP 후보들의 모션 벡터들은 TMVP 후보 이후에 삽입된다. AMVP 타겟 참조 평균과 동일한 참조 평균을 갖는 HMVP 후보들만이 AMVP 후보 리스트를 구성하기 위해 사용된다. 프루닝은 HMVP 후보들에 적용된다. 일 예에서, K1는 4로 설정된다.

[0287] 도 28은 비디오 처리 장치(2800)의 블록도이다. 장치(2800)는 본 명세서에 설명된 방법들 중 하나 이상을 구현하는 데 사용될 수 있다. 장치(2800)는 스마트폰, 태블릿, 컴퓨터, 사물 인터넷(IoT) 수신기 등에서 구현될 수 있다. 장치(2800)는 하나 이상의 프로세서(2802), 하나 이상의 메모리(2804) 및 비디오 처리 하드웨어(2806)를 포함할 수 있다. 프로세서(들)(2802)는 본 문서에서 설명된 하나 이상의 방법을 구현하도록 구성될 수 있다. 메모리(메모리들)(2804)는 본 명세서에 설명된 방법들 및 기술들을 구현하는 데 사용되는 데이터 및 코드를 저장하는 데 사용될 수 있다. 비디오 처리 하드웨어(2806)는 본 문서에 설명된 일부 기술들을 하드웨어 회로에서 구현하는 데 사용될 수 있다.

[0288] 도 29는 비디오 처리 방법(2900)의 일 예에 대한 흐름도이다. 방법(2900)은, 단계 2902에서, 테이블들의 세트를 유지하는 단계를 포함하며, 여기서, 각각의 테이블은 모션 후보들을 포함하고, 각각의 모션 후보는 대응하는 모션 정보와 연관됨된다. 방법(2900)은, 단계 2904에서, 모션 후보들에 대한 프루닝 동작을 사용하여 하나 이상의 테이블 내의 모션 후보들에 기초하여 모션 후보 리스트를 업데이트하는 단계를 더 포함한다. 방법(2900)은, 단계 2906에서, 구성된 모션 후보 리스트를 사용하여 제1 비디오 블록과 제1 비디오 블록을 포함하는 비디오의 비트스트림 표현 사이의 변환을 수행하는 단계를 더 포함한다.

[0289] 도 30은 비디오 처리 방법(3000)의 일 예에 대한 흐름도이다. 방법(3000)은, 단계 3002에서, 제1 비디오 블록을 포함하는 비디오의 비트스트림 표현을 수신하는 단계를 포함한다. 방법(3000)은 단계 3004에서, 프루닝 동작을 하나 이상의 테이블 내의 모션 후보들에 적용하여 후보 리스트를 업데이트하는 단계를 더 포함하고, 여기서, 각

각의 테이블은 모션 후보들을 포함하고, 각각의 모션 후보는 대응하는 모션 정보와 연관됨된다. 방법(3000)은, 단계 3006에서, 구성된 후보 리스트를 사용하여 비트스트림 표현과 제1 비디오 블록 사이의 변환을 수행하는 단계를 더 포함한다.

[0290] 전술한 방법들/기술들의 추가적인 특징들 및 실시예들이 조항 기반 설명 포맷을 사용하여 후술된다.

[0291] 1. 비디오 처리 방법으로서, 테이블들의 세트를 유지하는 단계- 각각의 테이블은 모션 후보들을 포함하고, 각각의 모션 후보는 대응하는 모션 정보와 연관됨 -; 모션 후보들에 대한 프루닝 동작을 사용하여 하나 이상의 테이블 내의 모션 후보들에 기초하여 후보 리스트를 업데이트하는 단계; 및 구성된 후보 리스트를 사용하여 제1 비디오 블록과 제1 비디오 블록을 포함하는 비디오의 비트스트림 표현 사이의 변환을 수행하는 단계를 포함하는, 방법.

[0292] 2. 비디오 처리 방법으로서, 제1 비디오 블록을 포함하는 비디오의 비트스트림 표현을 수신하는 단계; 및 후보 리스트를 구성하기 위해 하나 이상의 테이블 내의 모션 후보들에 프루닝 동작을 적용하는 단계- 각각의 테이블은 모션 후보들을 포함하고, 각각의 모션 후보는 대응하는 모션 정보와 연관됨 -; 및 구성된 후보 리스트를 사용하여 비트스트림 표현과 제1 비디오 블록 사이의 변환을 수행하는 단계를 포함하는, 방법.

[0293] 3. 조항 1 또는 2에 있어서, 테이블들 중 적어도 하나는 제1 비디오 블록 이전에 디코딩된, 이전에 디코딩된 비디오 블록들로부터 도출된 모션 후보들을 포함하는, 방법.

[0294] 4. 조항 1 또는 2에 있어서, 프루닝 동작은 테이블로부터의 모션 후보를 후보 리스트에 추가하기 전에 수행되는, 방법.

[0295] 5. 조항 3에 있어서, 후보 리스트는 병합 후보 리스트에 대응하는, 방법.

[0296] 6. 조항 1 또는 2에 있어서, 프루닝 동작은 테이블로부터의 모션 후보를 사용가능한 공간 또는 시간 병합 후보들의 적어도 일부와 비교하는 것을 포함하는, 방법.

[0297] 7. 조항 1 또는 2에 있어서, 프루닝 동작은 테이블로부터의 모션 후보를 모든 사용가능한 공간 병합 후보들과 비교하는 것을 포함하는, 방법.

[0298] 8. 조항 1, 2, 또는 4에 있어서, 프루닝 동작은 테이블로부터의 모션 후보를 서브 블록 기반 모션 후보로 변경하지 않고 수행되는, 방법.

[0299] 9. 조항 1 또는 2에 있어서, 프루닝 동작은 모션 후보를 추가하기 전에 테이블로부터의 모션 후보를 병합 후보 리스트 내의 사용가능한 모션 후보들의 적어도 일부와 비교하는 것을 포함하는, 방법.

[0300] 10. 조항 1 또는 2에 있어서, 프루닝 동작은 다수의 동작을 포함하고, 그 수는 공간 또는 시간 병합 후보들의 수의 함수인, 방법.

[0301] 11. 조항 9에 있어서, 동작들의 수는 M개의 후보가 병합 후보 리스트에서 사용가능한 경우, 이 때, 프루닝이 K개의 병합 후보에만 적용되도록 하는 것이고, 여기서, $K \leq M$ 이고, K 및 M은 정수들인, 방법.

[0302] 12. 조항 10에 있어서, K는 $\min(K, 2)$ 로 설정되는, 방법.

[0303] 13. 조항 1에 있어서, 프루닝 동작은 후보 리스트에 모션 후보를 추가하기 전에, 테이블로부터의 모션 후보를 후보 리스트 내의 처음 N개의 후보와 비교하는 것을 포함하고, N은 정수인, 방법.

[0304] 14. 조항 1 또는 2에 있어서, 프루닝 동작은 후보 리스트에 모션 후보를 추가하기 전에, 검사될 테이블로부터의 모션 후보를 후보 리스트 내의 마지막 N개의 후보와 비교하는 것을 포함하고, N은 정수인, 방법.

[0305] 15. 조항 12 또는 13에 있어서, N의 값을 시그널링하는 단계를 더 포함하는, 방법.

[0306] 16. 조항 1에 있어서, 프루닝 동작은 테이블로부터의 모션 후보를 후보가 도출되는 방법에 기초하여 후보 리스트 내의 특정 후보와 비교하는 것을 포함하는, 방법.

[0307] 17. 조항 15에 있어서, 프루닝 동작은 테이블로부터의 모션 후보를 주어진 시간 또는 공간 이웃 블록으로부터 도출된 후보들과 비교하는 것을 포함하는, 방법.

[0308] 18. 조항 15에 있어서, 테이블 내의 모션 후보들의 상이한 엔트리들이 상이한 위치들로부터 도출된 상이한 이전에 추가된 후보들과 비교되는, 방법.

[0309] 19. 조항 1에 있어서, 테이블로부터의 모션 후보를 후보 리스트에 추가할 때 프루닝 동작이 호출되지 않는 것을 더 포함하는, 방법.

[0310] 20. 조항 1 내지 19 중 어느 하나에 있어서, 변환을 수행하는 단계는 비디오 블록으로부터 비트스트림 표현을 생성하는 단계를 포함하는, 방법.

[0311] 21. 조항 1 내지 19 중 어느 하나에 있어서, 변환을 수행하는 단계는 비트스트림 표현으로부터 비디오 블록을 생성하는 단계를 포함하는, 방법.

[0312] 22. 조항 1 내지 21 중 어느 하나에 있어서, 모션 후보는 예측 방향, 참조 꾹처 인덱스, 모션 벡터 값, 강도 보상 플래그, 아핀 플래그, 모션 벡터 차이 정밀도 또는 모션 벡터 차이 값 중 적어도 하나를 포함하는 모션 정보와 연관되는, 방법.

[0313] 23. 조항 1 내지 22 중 어느 하나에 있어서, 모션 후보들은 인트라 모드 코딩을 위한 인트라 예측 모드들에 대한 모션 후보들에 대응하는, 방법.

[0314] 24. 조항 1 내지 22 중 어느 하나에 있어서, 모션 후보들은 IC 파라미터 코딩을 위한 조도 보상 파라미터들을 포함하는 모션 후보들에 대응하는, 방법.

[0315] 25. 조항 1 내지 24 중 어느 하나에 있어서, 변환에 기초하여, 하나 이상의 테이블을 업데이트하는 단계를 더 포함하는, 방법.

[0316] 26. 조항 25에 있어서, 하나 이상의 테이블을 업데이트하는 단계는 변환을 수행한 이후에 제1 비디오 블록의 모션 정보에 기초하여 하나 이상의 테이블을 업데이트하는 단계를 포함하는, 방법.

[0317] 27. 조항 26에 있어서, 업데이트된 테이블들에 기초하여 비디오의 후속 비디오 블록과 비디오의 비트스트림 표현 사이의 변환을 수행하는 단계를 더 포함하는, 방법.

[0318] 28. 프로세서 및 명령어들을 갖는 비일시적 메모리를 포함하는 장치로서, 명령어들은 프로세서에 의한 실행 시에, 프로세서로 하여금 조항 1 내지 27 중 어느 하나의 방법을 구현하게 하는, 장치.

[0319] 29. 비일시적 컴퓨터 판독가능 매체 상에 저장되는 컴퓨터 프로그램 제품으로서, 컴퓨터 프로그램 제품은 조항 1 내지 27 중 어느 하나의 방법을 수행하기 위한 프로그램 코드를 포함하는, 컴퓨터 프로그램 제품.

[0320] 전술한 설명으로부터, 현재 개시된 기술의 특정한 실시예들이 예시의 목적으로 본 명세서에 설명되었지만, 본 발명의 범위를 벗어나지 않고서 다양한 수정들이 이루어질 수 있다는 것을 알 것이다. 따라서, 현재 개시된 기술은 첨부된 청구항들에 의한 것을 제외하고는 제한되지 않는다.

[0321] 본 문서에 설명된 개시된 및 다른 실시예들, 모듈들 및 기능 동작들은 본 문서에 개시된 구조들 및 그들의 구조적 등가물들을 포함하는 디지털 전자 회로, 또는 컴퓨터 소프트웨어, 펌웨어, 또는 하드웨어로, 또는 이를 중 하나 이상의 조합들로 구현될 수 있다. 개시된 및 다른 실시예들은 하나 이상의 컴퓨터 프로그램 제품들, 즉 데이터 처리 장치에 의한 실행을 위해 또는 데이터 처리 장치의 동작을 제어하기 위해 컴퓨터 판독가능 매체 상에 인코딩된 컴퓨터 프로그램 명령어들의 하나 이상의 모듈들로서 구현될 수 있다. 컴퓨터 판독가능 매체는 머신 판독가능 저장 디바이스, 머신 판독가능 저장 기판, 메모리 디바이스, 머신 판독가능 전파 신호에 영향을 주는 물질의 조성, 또는 이를 중 하나 이상의 조합일 수 있다. 용어 "데이터 처리 장치"는 예로서 프로그램가능 프로세서, 컴퓨터, 또는 다수의 프로세서들 또는 컴퓨터들을 포함하는, 데이터를 처리하기 위한 모든 장치, 디바이스들, 및 머신들을 포함한다. 장치는 하드웨어에 더하여, 해당 컴퓨터 프로그램에 대한 실행 환경을 생성하는 코드, 예를 들어 프로세서 펌웨어, 프로토콜 스택, 데이터베이스 관리 시스템, 운영 체제, 또는 이를 중 하나 이상의 조합을 구성하는 코드를 포함할 수 있다. 전파 신호는 적합한 수신기 장치로의 송신을 위해 정보를 인코딩하도록 생성되는 인위적으로 생성된 신호, 예를 들어 머신 생성 전기, 광 또는 전자기 신호이다.

[0322] 컴퓨터 프로그램(또한, 프로그램, 소프트웨어, 소프트웨어 애플리케이션, 스크립트, 또는 코드라고도 알려져 있음)은 컴파일된 또는 해석된 언어들을 포함하는 임의의 형태의 프로그래밍 언어로 작성될 수 있고, 이는 독립형 프로그램(stand-alone program)으로서, 또는 모듈, 컴포넌트, 서브루틴, 또는 컴퓨팅 환경에서 사용하기에 적합한 다른 유닛으로서를 포함하여 임의의 형태로 배치될 수 있다. 컴퓨터 프로그램은 파일 시스템 내의 파일에 반드시 대응하지는 않는다. 프로그램은 다른 프로그램들 또는 데이터(예를 들어, 마크업 언어 문서에 저장된 하나 이상의 스크립트들)를 보유하는 파일의 일부 내에, 해당 프로그램에 전용된 단일 파일 내에, 또는 다수의 협력하는 파일들(예를 들어, 하나 이상의 모듈들, 서브 프로그램들, 또는 코드의 부분들을 저장하는 파일들) 내에

저장될 수 있다. 컴퓨터 프로그램은 하나의 컴퓨터 상에서, 또는 한 장소에 위치하거나 다수의 위치에 걸쳐 분산되어 통신 네트워크에 의해 상호접속되는 다수의 컴퓨터 상에서 실행되도록 배치될 수 있다.

[0323] 본 문서에 설명된 프로세스들 및 로직 회로들은 입력 데이터에 대해 동작하고 출력을 생성함으로써 기능들을 수행하기 위해 하나 이상의 컴퓨터 프로그램을 실행하는 하나 이상의 프로그램가능 프로세서에 의해 수행될 수 있다. 프로세스들 및 로직 회로들은 또한 특수 목적 로직 회로, 예를 들어, FPGA(field programmable gate array) 또는 ASIC(application specific integrated circuit)에 의해 수행될 수 있고, 장치는 또한 그들로서 구현될 수 있다.

[0324] 컴퓨터 프로그램의 실행에 적합한 프로세서들은, 예로서, 범용 및 특수 목적 마이크로프로세서 양자 모두, 및 임의의 종류의 디지털 컴퓨터의 임의의 하나 이상의 프로세서를 포함한다. 일반적으로, 프로세서는 판독 전용 메모리 또는 랜덤 액세스 메모리 또는 둘 다로부터 명령어들 및 데이터를 수신할 것이다. 컴퓨터의 필수 요소들은 명령어들을 수행하기 위한 프로세서 및 명령어들 및 데이터를 저장하기 위한 하나 이상의 메모리 디바이스이다. 일반적으로, 컴퓨터는 또한 데이터를 저장하기 위한 하나 이상의 대용량 저장 디바이스, 예를 들어, 자기, 광자기 디스크들, 또는 광 디스크들을 포함하거나, 또는 이들로부터 데이터를 수신하거나 데이터를 이에 전송하거나, 또는 양자 모두를 수행하도록 동작적으로 결합될 것이다. 그러나, 컴퓨터가 이러한 디바이스들을 가질 필요는 없다. 컴퓨터 프로그램 명령어들 및 데이터를 저장하기에 적합한 컴퓨터 판독가능 매체는 예로서 반도체 메모리 디바이스들, 예를 들어 EEPROM, EEPROM 및 플래시 메모리 디바이스들; 자기 디스크들, 예를 들어, 내부 하드 디스크들 또는 이동식 디스크들; 광자기 디스크들; 및 CD ROM 및 DVD-ROM 디스크들을 포함하는 모든 형태의 비휘발성 메모리, 매체 및 메모리 디바이스들을 포함한다. 프로세서 및 메모리는 특수 목적 로직 회로에 의해 보완되거나 그에 통합될 수 있다.

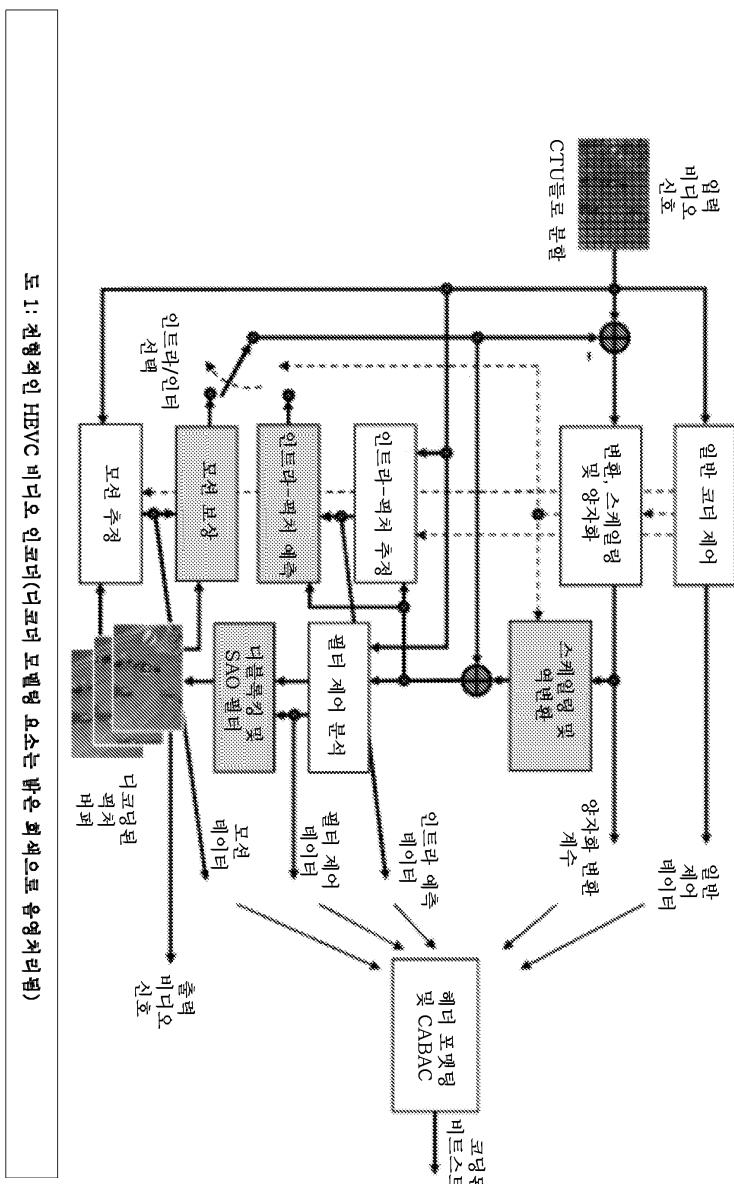
[0325] 본 특허 문서가 많은 세부사항을 포함하고 있지만, 이들은 임의의 발명 또는 청구될 수 있는 것의 범위에 대한 제한으로서 해석되어서는 안되며, 오히려 특정의 발명의 특정 실시예에 특정될 수 있는 특징들에 대한 설명으로서 해석되어야 한다. 개별 실시예들에 관련하여 본 특허 문서에 설명되는 특정 특징들은 또한 단일 실시예에서 조합하여 구현될 수 있다. 반대로, 단일 실시예에 관련하여 설명되는 다양한 특징들은 또한 다수의 실시예들에서 개별적으로 또는 임의의 적절한 하위조합으로 구현될 수 있다. 또한, 특징들이 특정 조합들로 작용하는 것으로 앞서 설명되고 심지어 처음에 이와 같이 청구될 수 있지만, 청구된 조합으로부터의 하나 이상의 특징은 일부 경우들에서 조합으로부터 삭제될 수 있고, 청구된 조합은 하위조합 또는 하위조합의 변형에 관련될 수 있다.

[0326] 유사하게, 동작들이 특정 순서로 도면들에 도시되지만, 이것은 바람직한 결과들을 달성하기 위해, 그러한 동작들이 도시된 특정 순서로 또는 순차적인 순서로 수행되거나, 모든 예시된 동작들이 수행되는 것을 요구하는 것으로 이해되어서는 안 된다. 또한, 본 특허 문서에 설명된 실시예들에서의 다양한 시스템 컴포넌트들의 분리는 모든 실시예들에서 이러한 분리가 요구되는 것으로 이해하여서는 안 된다.

[0327] 몇몇 구현들 및 예들만이 설명되고, 본 특허 문서에서 설명되고 예시된 것에 기초하여 다른 구현들, 향상들 및 변형들이 이루어질 수 있다.

도면

도면1

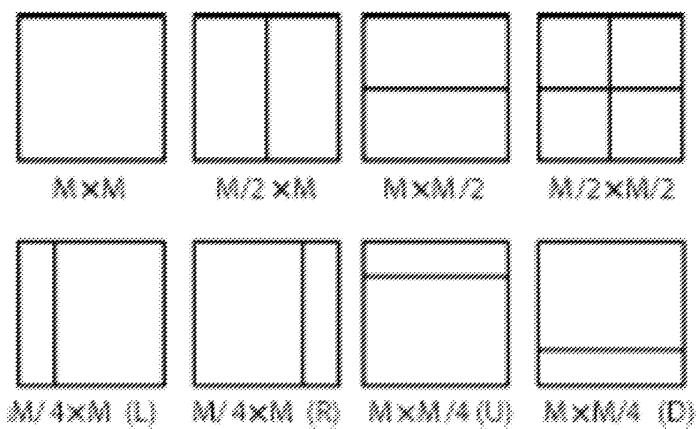


도 1: 천형적인 HEVC 비디오 인코더(디코더 모델링 요소는 빠운 회색으로 음영처리됨)

도면2

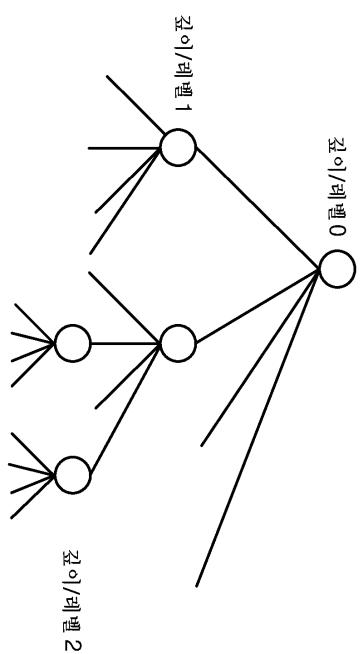
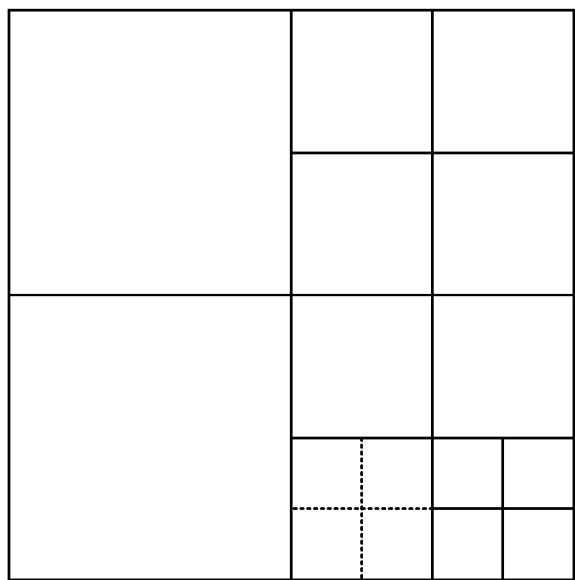
MB-모드들	16x16	16x8	8x16	8x8
	0	0	0 1	0 1
8x8		1		2 3
8x4			2	
4x8			3	
4x4				0 1
8x8-모드들	0	1	2	3

도면3

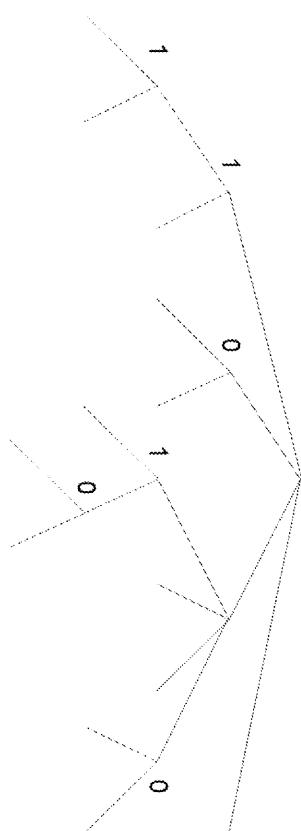
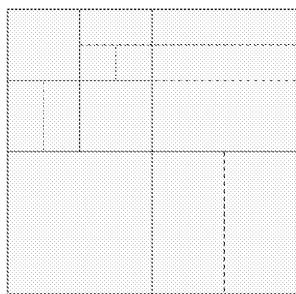


도 3: 특정 크기 제약에 따라 CB를 PB로 분할하는 모드 인트라-예측
CB들에 대해, $M \times M$ 및 $M/2 \times M/2$ 만이 지원됨.

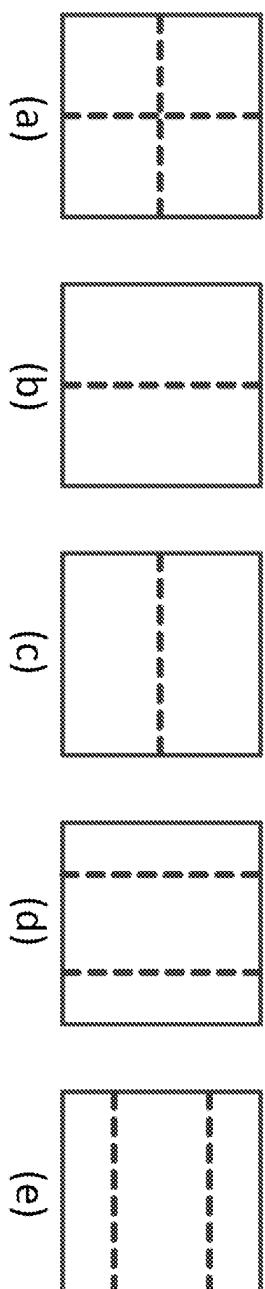
도면4



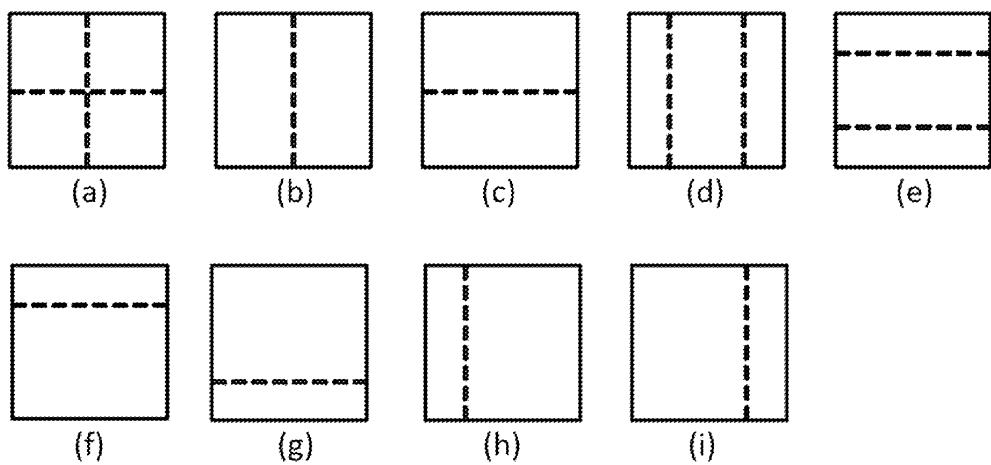
도면5



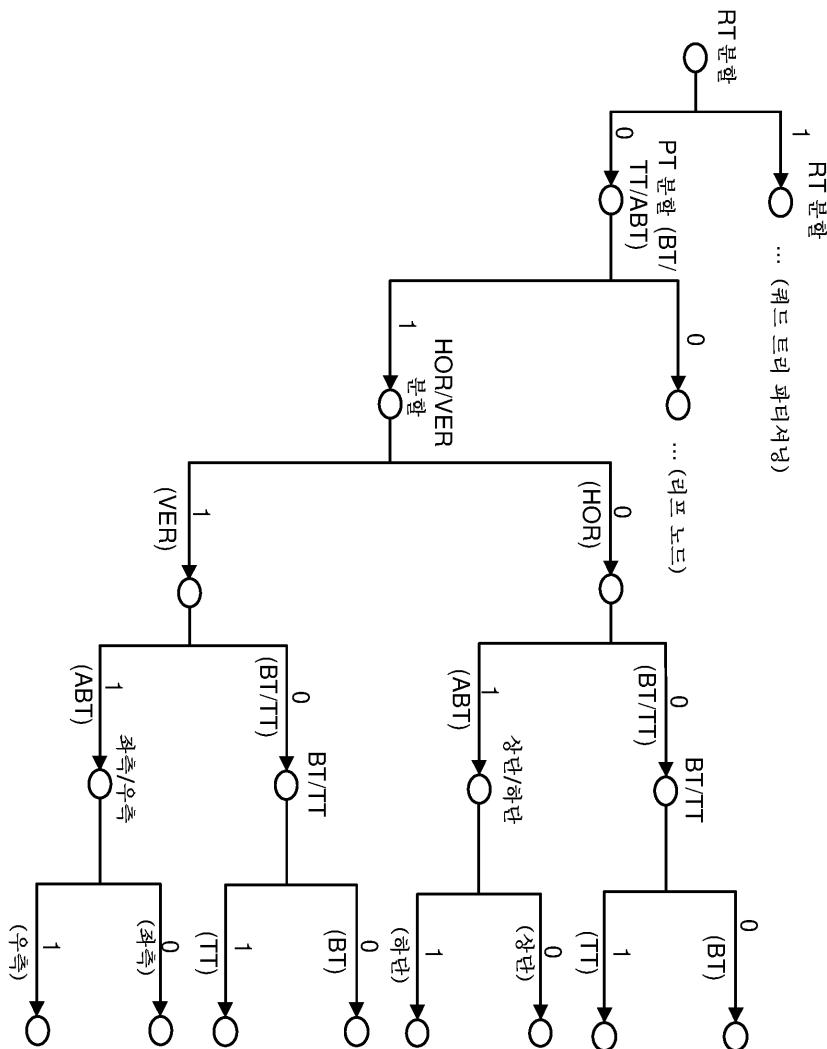
도면6



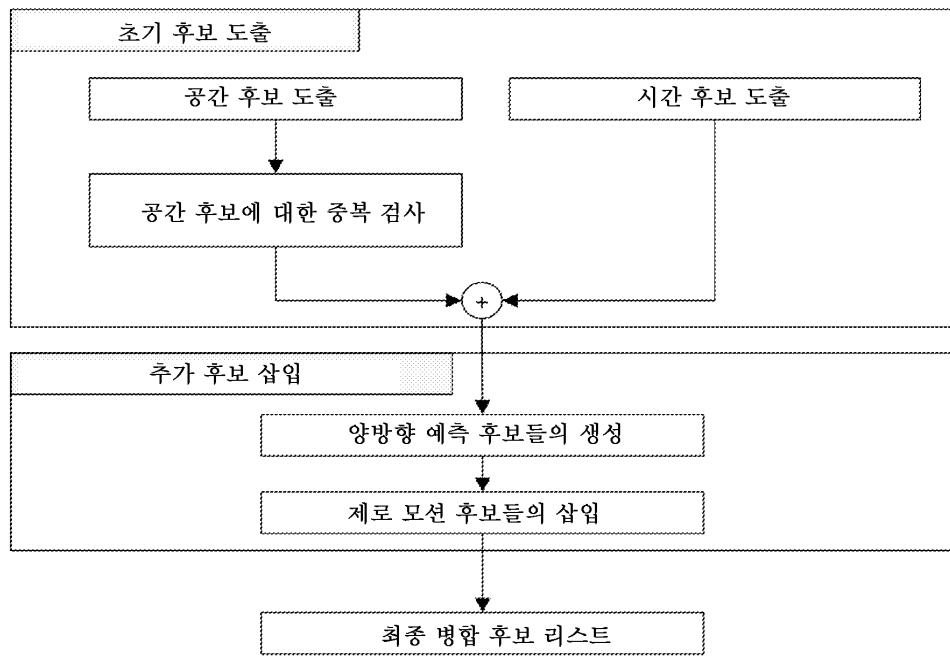
도면7



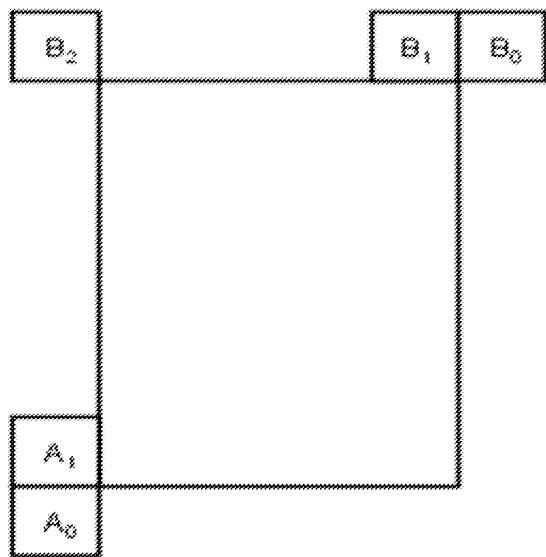
도면8



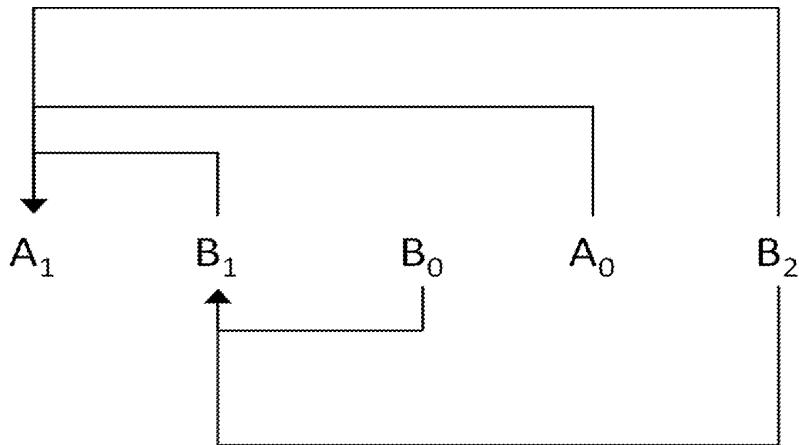
도면9



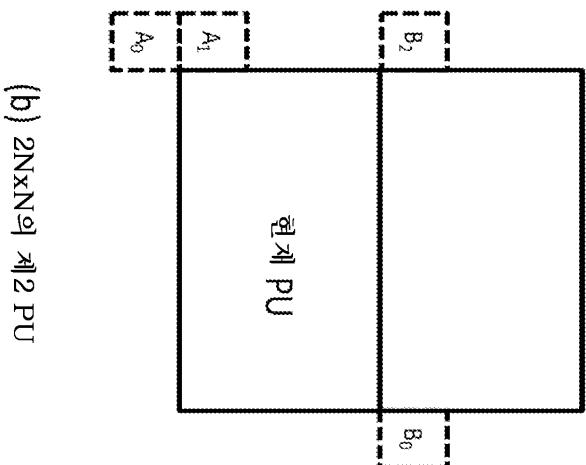
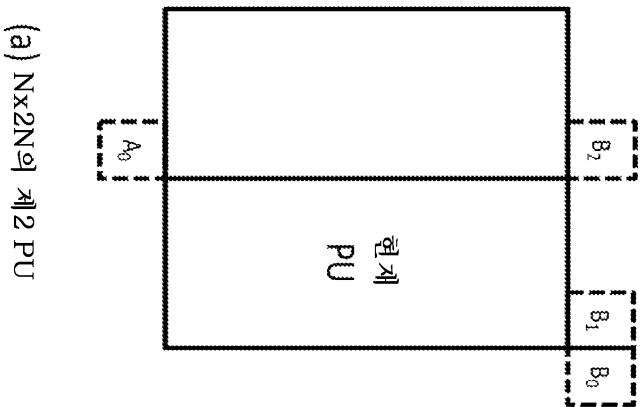
도면10



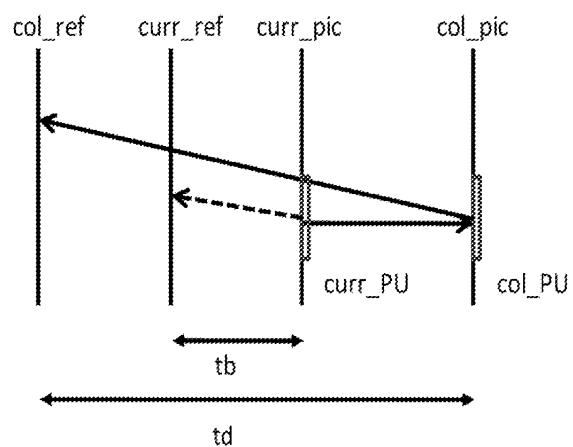
도면11



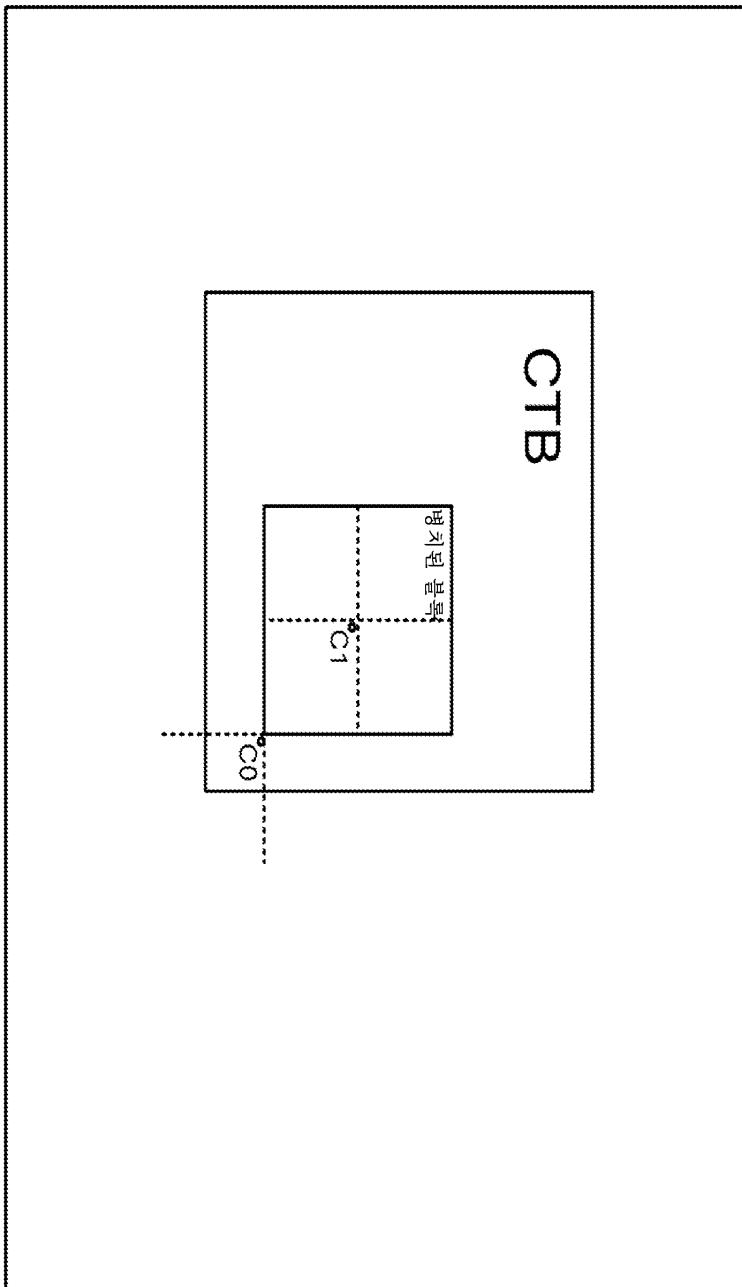
도면12



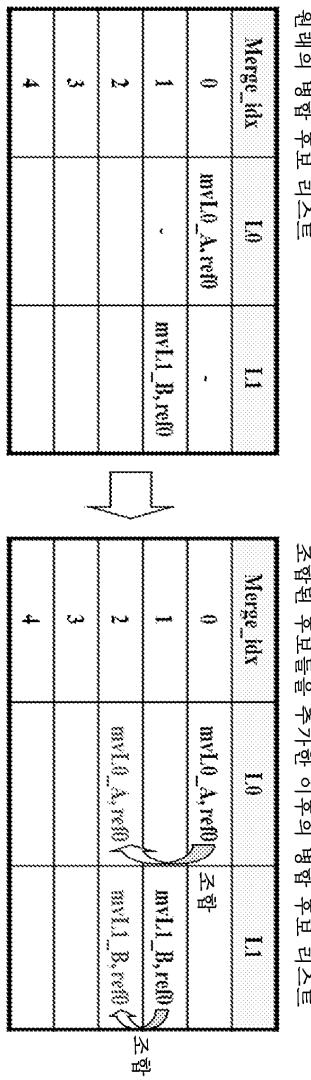
도면13



도면14

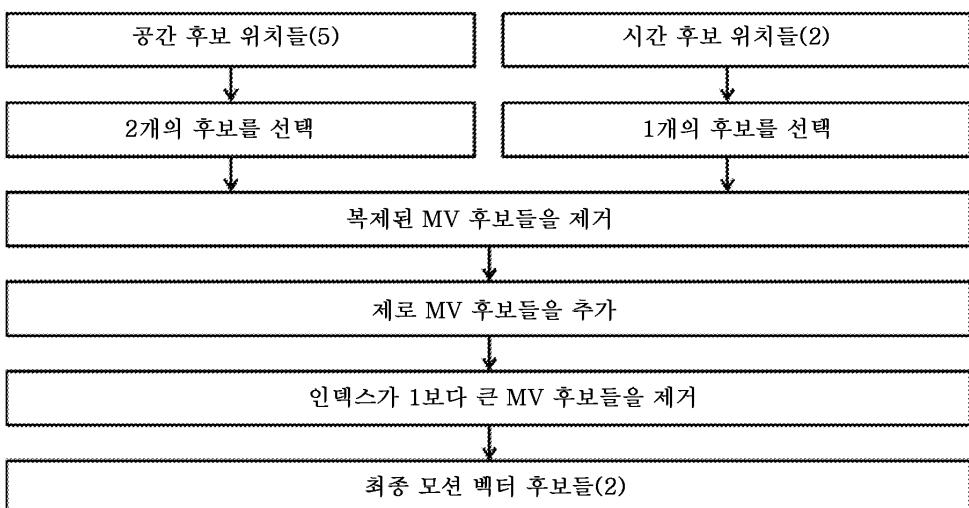


도면15

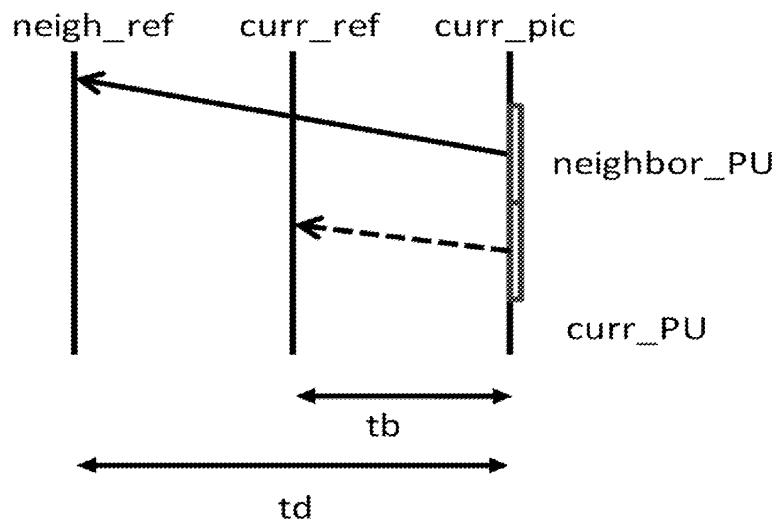


도면16

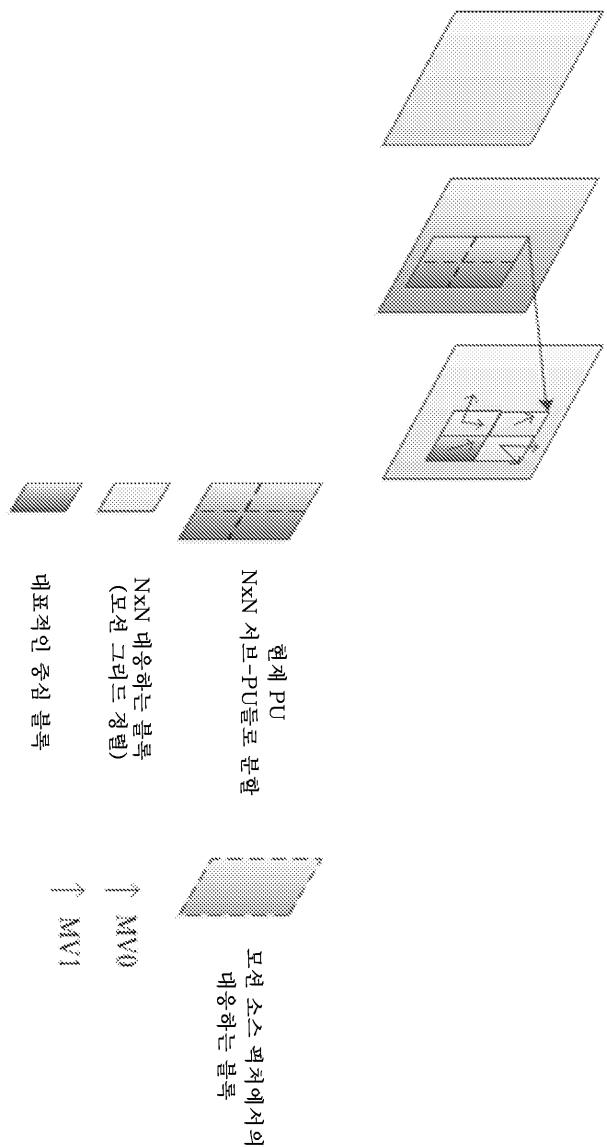
입력으로서 refidx를 갖는 각각의 참조 꽈치 리스트에 대해



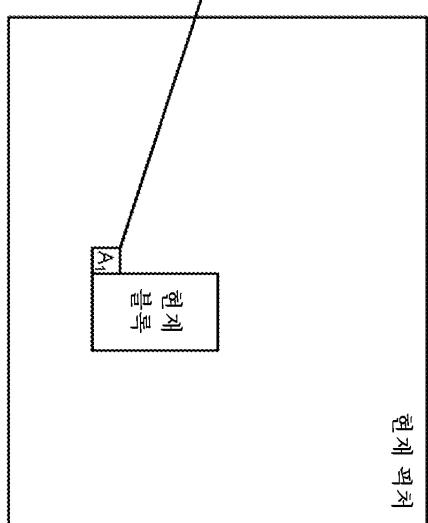
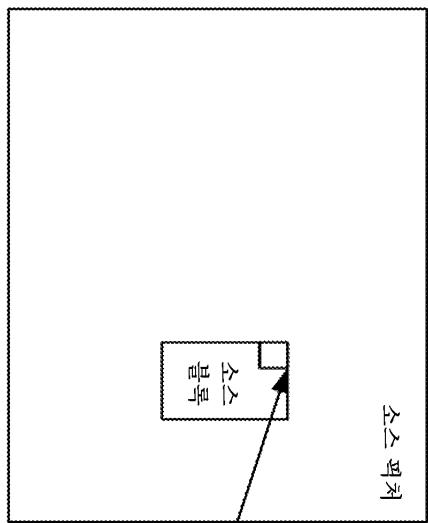
도면17



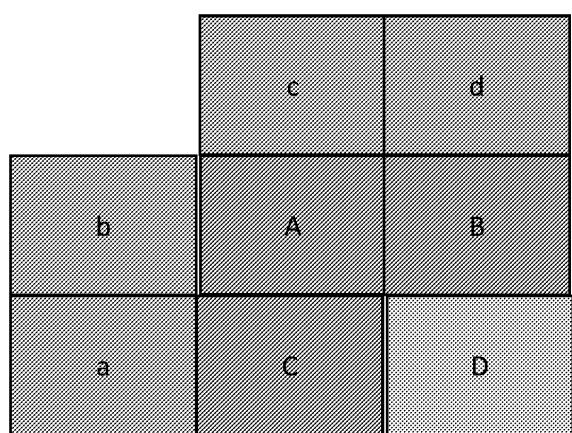
도면 18



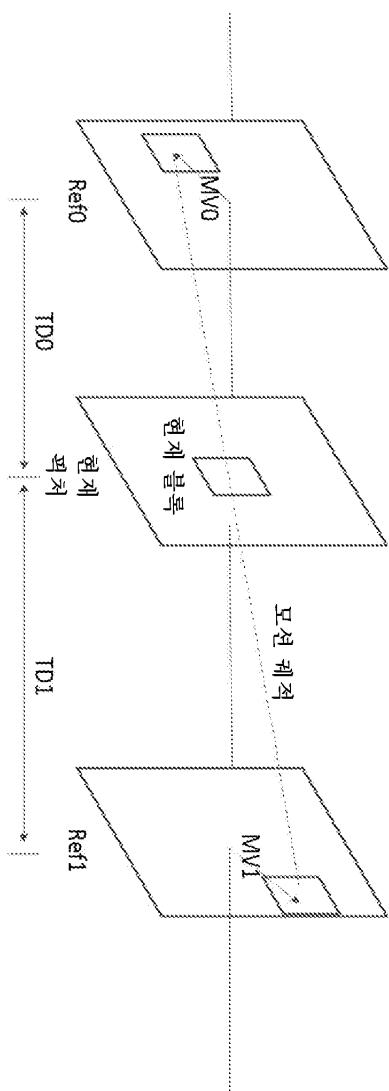
도면19



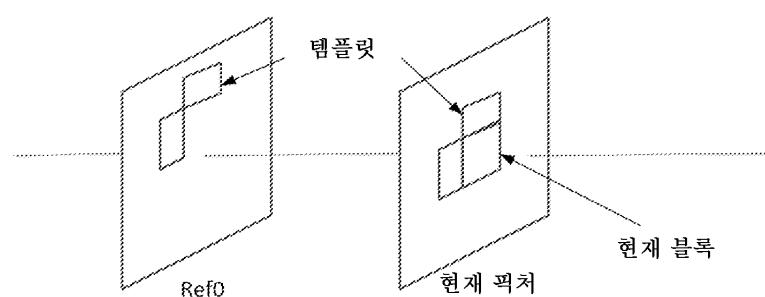
도면20



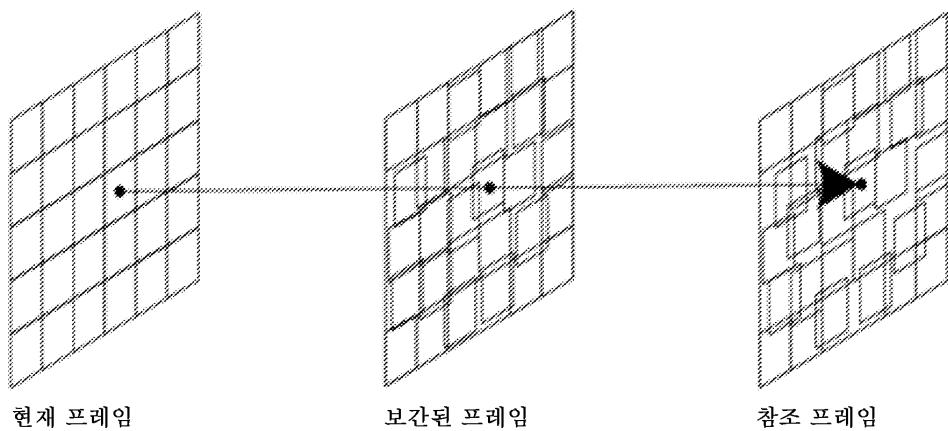
도면21



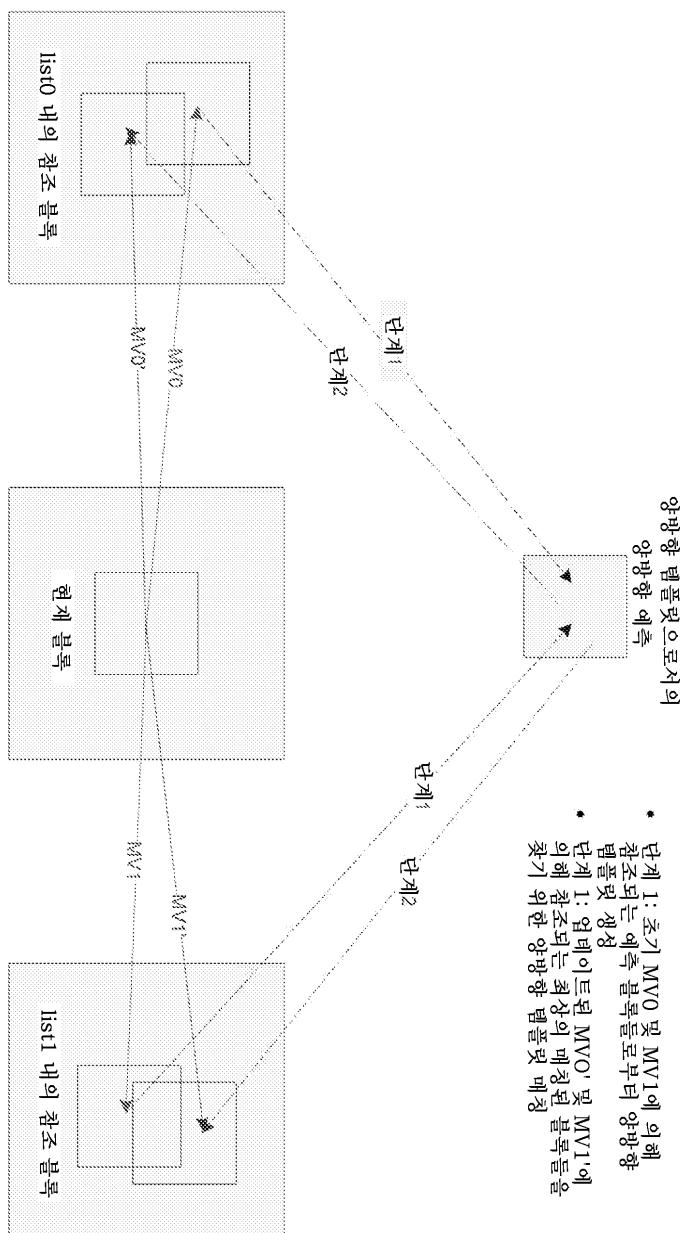
도면22



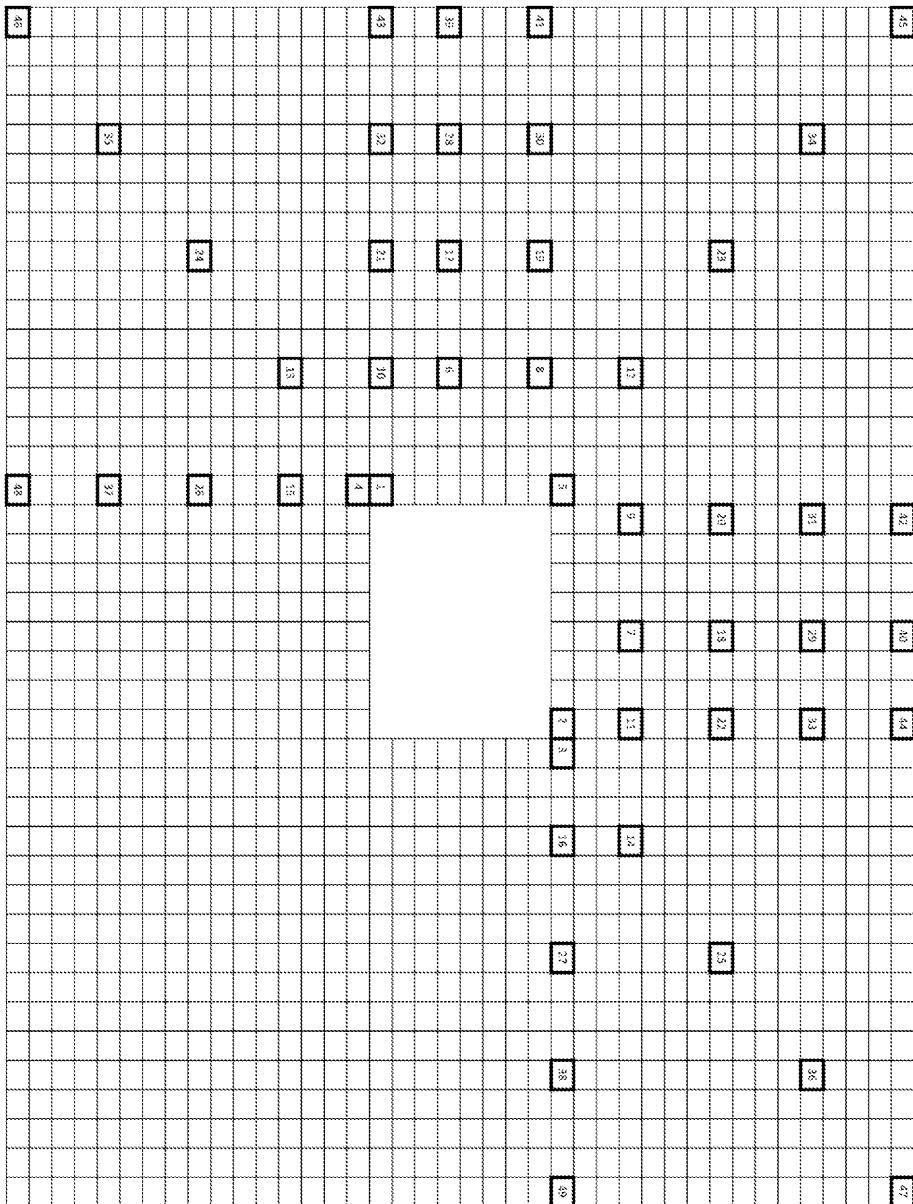
도면23



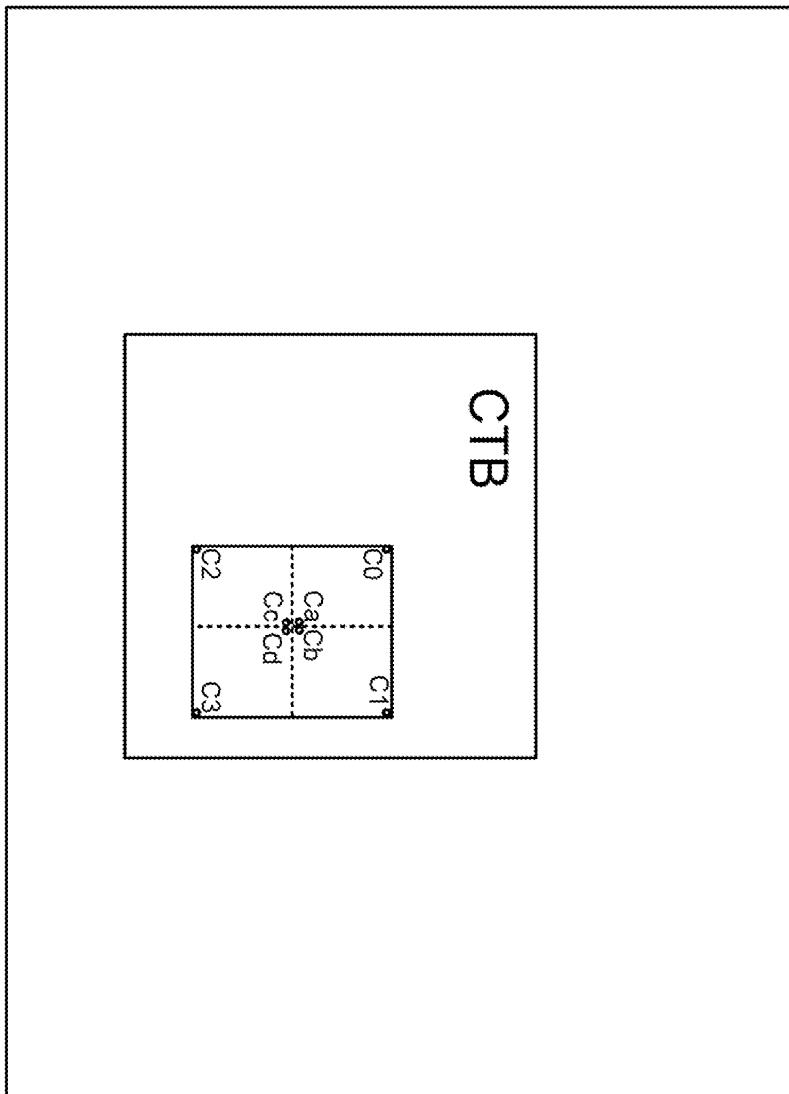
도면24



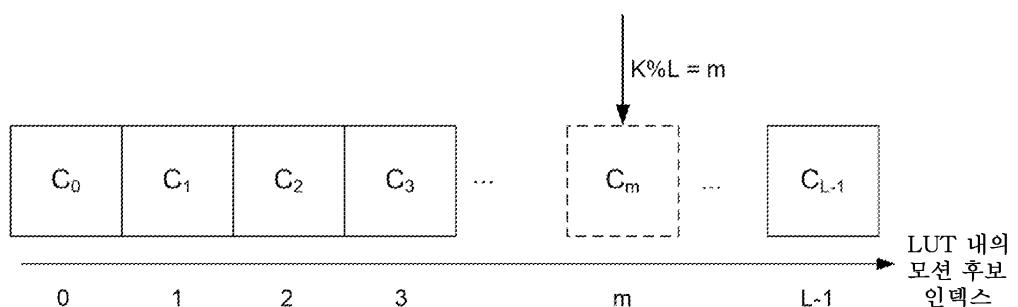
도면25



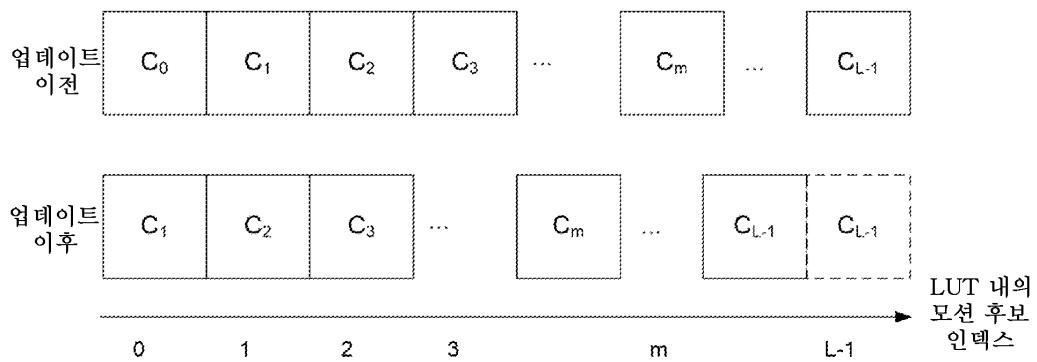
도면26



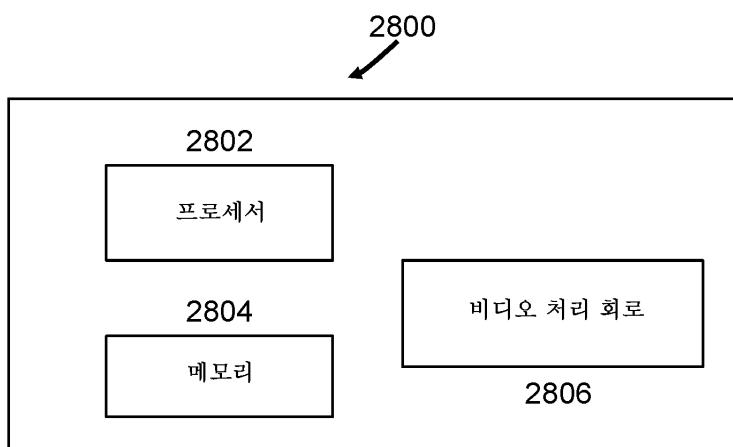
도면27a



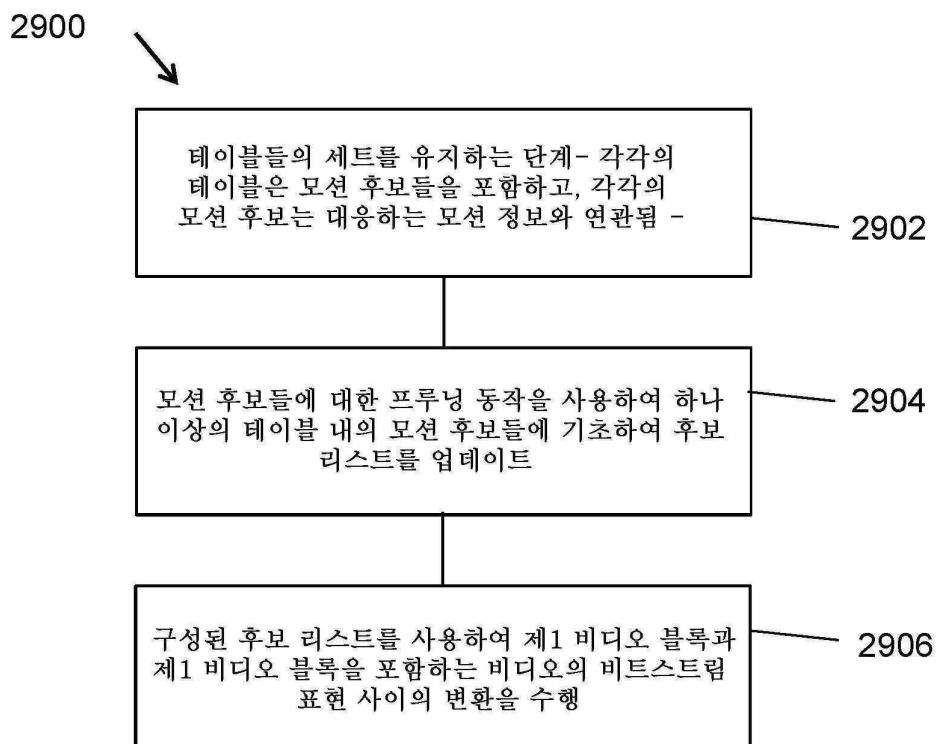
도면27b



도면28



도면29



도면30

3000



제1 비디오 블록을 포함하는 비디오의
비트스트림 표현을 수신

3002

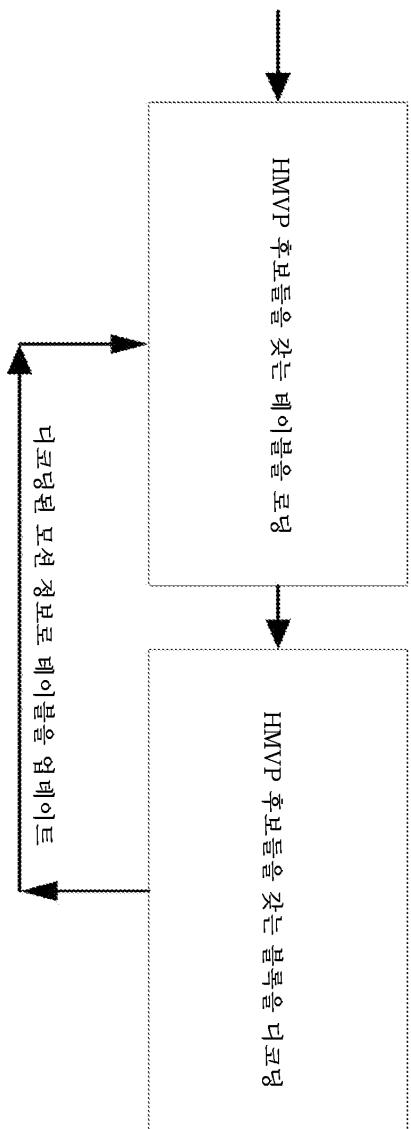
후보 리스트를 구성하기 위해 하나 이상의 테이블
내의 모션 후보들에 프루닝 동작을 적용- 각각의
테이블은 모션 후보들을 포함하고, 각각의 모션
후보는 대응하는 모션 정보와 연관됨 -

3004

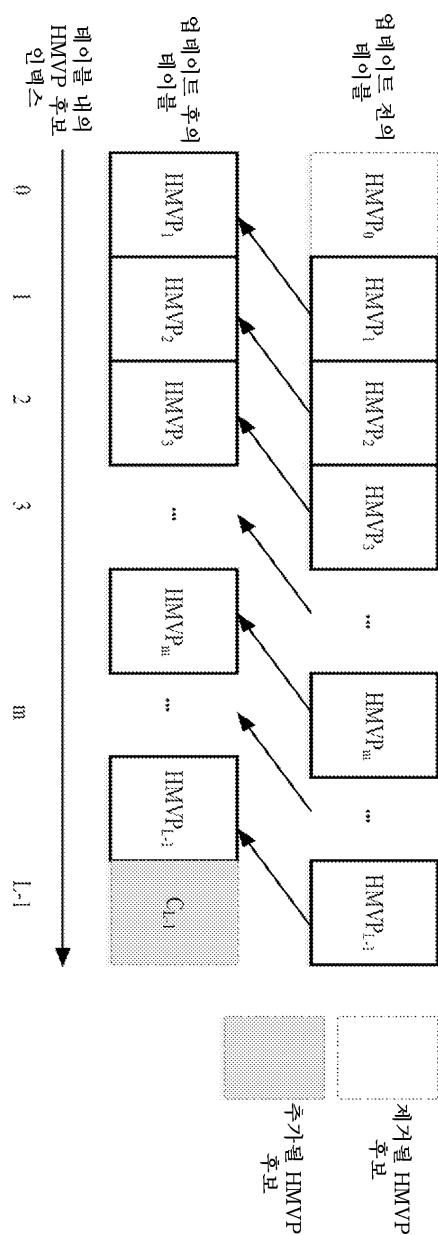
구성된 후보 리스트를 사용하여 비트스트림
표현과 제1 비디오 블록 사이의 변환을 수행하는
단계를 포함한다.

3006

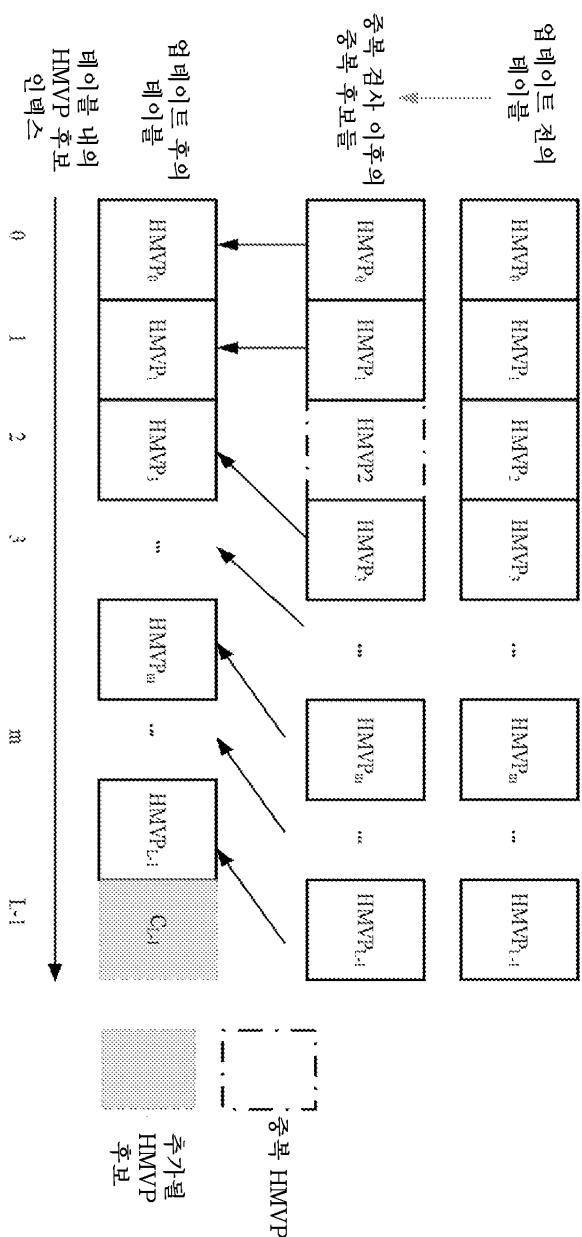
도면31



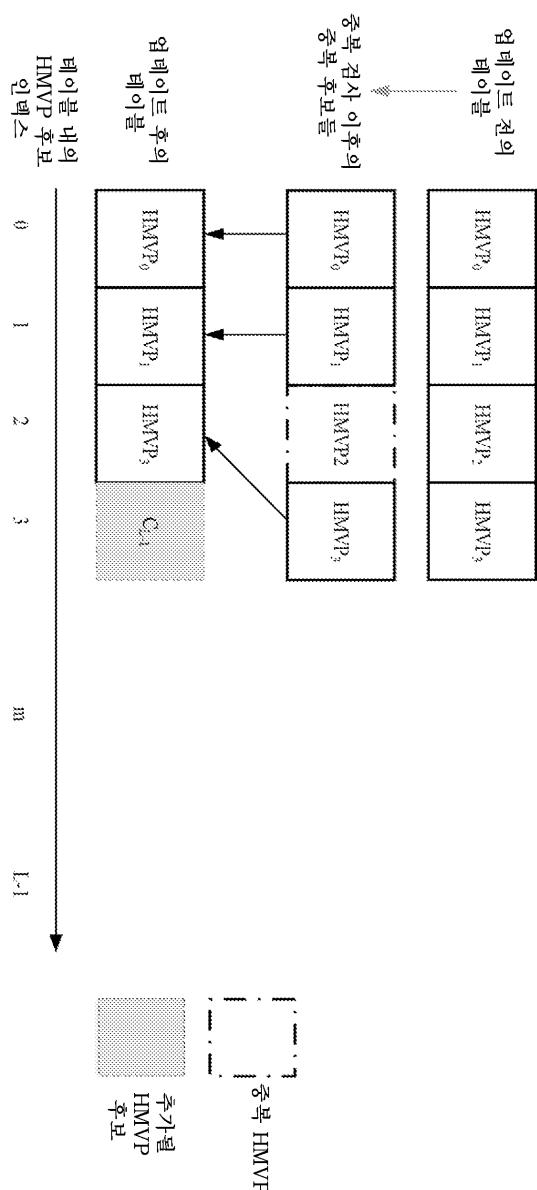
도면32



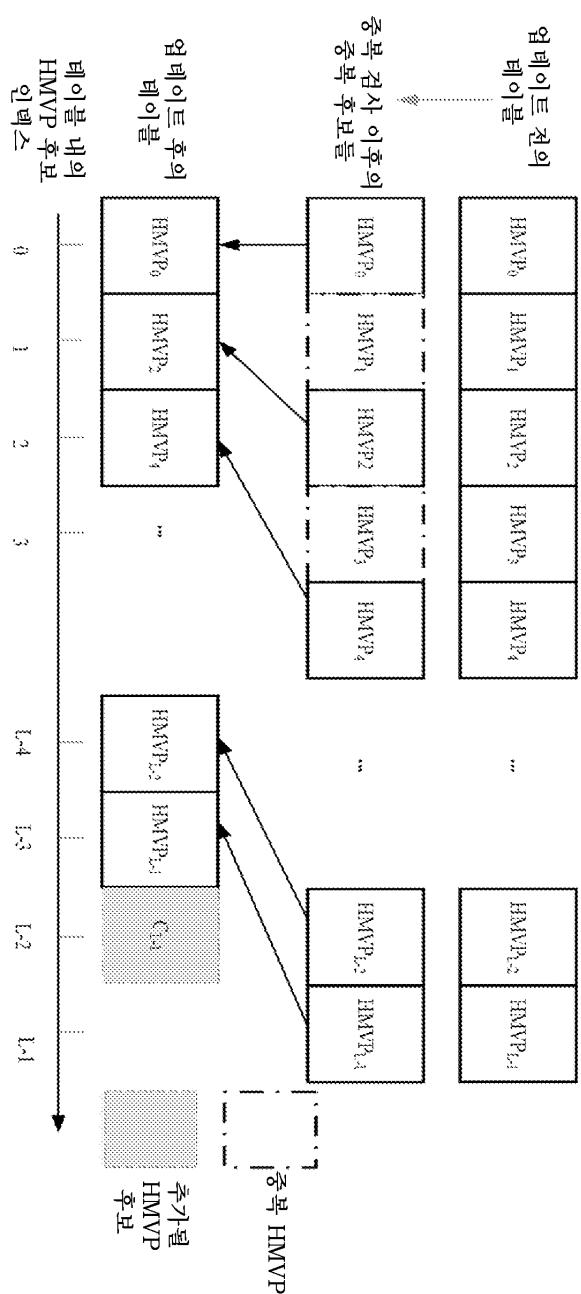
도면33a



도면33b



도면34a



도면34b

