

(52) CPC특허분류

H04N 19/463 (2015.01)

H04N 19/521 (2015.01)

H04N 19/593 (2015.01)

H04N 19/70 (2015.01)

H04N 19/90 (2015.01)

H04N 19/94 (2015.01)

명세서

청구범위

청구항 1

예측 백터를 도출하는 비디오 코딩 방법에 있어서,

비디오 블록의 예측을 위한 제 1 후보 블록 벡터 - 상기 제 1 후보 블록 벡터는 제 1 후보 블록을 나타냄 - 를 식별하는 단계;

상기 제 1 후보 블록과 연관된 적어도 제 1 예측 백터를 식별하는 단계;

상기 제 1 후보 블록 벡터 및 상기 제 1 예측 백터로부터 도출된 예측 백터를 생성하는 단계; 및

상기 도출된 예측 백터를 이용하여 상기 비디오 블록을 코딩하는 단계

를 포함하는 비디오 코딩 방법.

청구항 2

제 1 항에 있어서,

상기 도출된 예측 백터를 이용하여 상기 비디오 블록을 코딩하는 단계는,

상기 도출된 예측 백터가 나타내는 제 2 후보 블록을 식별하는 단계; 및

상기 제 2 후보 블록을 이용하여 상기 비디오 블록을 예측하는 단계

를 포함하는 비디오 코딩 방법.

청구항 3

제 1 항에 있어서,

상기 비디오 블록을 코딩하는 단계는, 비트 스트림에서 상기 제 1 후보 블록 벡터를 시그널링하는 단계를 포함하는 비디오 코딩 방법.

청구항 4

제 3 항에 있어서,

상기 비디오 블록을 코딩하는 단계는, 상기 비트 스트림에서 상기 제 1 예측 백터를 시그널링하는 단계를 더 포함하는 비디오 코딩 방법.

청구항 5

제 3 항에 있어서,

상기 비디오 블록을 코딩하는 단계는, 상기 비트 스트림에서 플래그를 시그널링하는 단계를 더 포함하고, 상기 플래그의 제 1 값은 상기 도출된 예측 백터가 상기 비디오 블록을 코딩하는데 사용되는 것을 나타내는 비디오 코딩 방법.

청구항 6

제 5 항에 있어서,

상기 플래그의 제 2 값은 상기 제 1 후보 블록 벡터가 상기 비디오 블록을 코딩하는데 사용되는 것을 나타내는 비디오 코딩 방법.

청구항 7

제 1 항에 있어서,

상기 도출된 예측 벡터는 상기 제 1 후보 블록 벡터 및 상기 제 1 예측 벡터를 부가함으로써 생성되는 비디오 코딩 방법.

청구항 8

제 7 항에 있어서,

상기 제 1 후보 블록 벡터 및 상기 제 1 예측 벡터는 상이한 정밀도를 가지며, 상기 제 1 블록 벡터 및 상기 제 1 예측 벡터를 부가하는 것은 상기 제 1 블록 벡터 및 상기 제 1 예측 벡터를 동일한 정밀도로 정렬한 후에 수행되는 비디오 코딩 방법.

청구항 9

제 1 항에 있어서,

상기 제 1 예측 벡터는 제 2 블록 벡터이고, 상기 도출된 예측 벡터는 도출된 블록 벡터인 비디오 코딩 방법.

청구항 10

제 9 항에 있어서,

상기 도출된 블록 벡터는 상기 제 1 후보 블록 벡터 및 상기 제 2 블록 벡터를 부가함으로써 생성되는 비디오 코딩 방법.

청구항 11

제 1 항에 있어서,

상기 제 1 예측 벡터는 제 1 움직임 벡터이고, 상기 도출된 예측 벡터는 도출된 움직임 벡터인 비디오 코딩 방법.

청구항 12

제 11 항에 있어서,

상기 도출된 움직임 벡터는 상기 제 1 후보 블록 벡터 및 상기 제 1 움직임 벡터를 부가함으로써 생성되는 비디오 코딩 방법.

청구항 13

제 12 항에 있어서,

상기 제 1 후보 블록 벡터 및 상기 제 1 움직임 벡터는 상이한 정밀도를 가지고, 상기 도출된 움직임 벡터는 상기 제 1 후보 블록 벡터 및 상기 제 1 움직임 벡터의 정밀도를 정렬하고, 상기 정밀도가 정렬된 제 1 후보 블록 벡터 및 상기 정밀도가 정렬된 제 1 움직임 벡터를 부가함으로써 생성되는 비디오 코딩 방법.

청구항 14

제 13 항에 있어서,

상기 정밀도가 정렬된 제 1 후보 블록 벡터 및 상기 정밀도가 정렬된 제 1 움직임 벡터는, 식 $MVd = BVO + ((MV1+2) \gg 2)$ 에 따라 부가되고, 여기서 BVO는 제 1 후보 블록 벡터이고, MV1은 제 1 움직임 벡터이며, MVd는 도출된 움직임 벡터인 비디오 코딩 방법.

청구항 15

제 1 항에 있어서,

상기 비디오 코딩 방법은, 상기 도출된 예측 벡터를 병합 후보 리스트에 삽입하는 단계를 더 포함하는 비디오 코딩 방법.

청구항 16

제 15 항에 있어서,

상기 도출된 예측 벡터가 유효한지를 결정하는 단계를 더 포함하며, 상기 도출된 예측 벡터는 상기 도출된 예측 벡터가 유효한 것으로 결정한 후에만 상기 병합 후보 리스트에 삽입되는 비디오 코딩 방법.

청구항 17

제 16 항에 있어서,

상기 도출된 예측 벡터가 유효한지를 결정하는 단계는,

상기 도출된 예측 벡터가 나타내는 제 2 후보 블록을 식별하는 단계;

상기 제 2 후보 블록 내의 모든 샘플이 이용 가능한지를 결정하는 단계; 및

상기 제 2 후보 블록 내의 모든 샘플이 이용 가능하다면 상기 도출된 예측 벡터가 유효한 것으로 결정하고, 상기 제 2 후보 블록 내의 적어도 하나의 샘플이 이용 가능하지 않으면 상기 도출된 예측 벡터가 유효하지 않은 것으로 결정하는 단계

를 포함하는 비디오 코딩 방법.

청구항 18

제 17 항에 있어서,

상기 제 2 후보 블록 내의 샘플은,

상기 샘플이 아직 코딩되지 않았거나,

상기 샘플이 상이한 슬라이스 또는 상이한 타일에 있거나,

상기 샘플이 비디오 화상 경계에서 벗어난 것

중 임의의 것이 참이면 이용 가능하지 않은 비디오 코딩 방법.

청구항 19

제 15 항에 있어서,

상기 도출된 예측 벡터를 이용하여 상기 비디오 블록을 코딩하는 단계는, 상기 병합 후보 리스트에서 상기 도출된 예측 벡터를 식별하는 인덱스를 제공하는 단계를 포함하는 비디오 코딩 방법.

청구항 20

비디오를 인코딩하는 비트 스트림으로부터 코딩된 비디오 블록을 디코딩하는 방법에 있어서,

상기 코딩된 비디오 블록의 예측을 위한 적어도 제 1 후보 블록 벡터 - 상기 제 1 후보 블록 벡터는 제 1 후보 블록을 나타냄 - 를 식별하는 단계;

상기 제 1 후보 블록과 연관된 적어도 제 1 예측 벡터를 식별하는 단계;

상기 제 1 후보 블록 벡터 및 상기 제 1 예측 벡터로부터 도출된 예측 벡터를 생성하는 단계; 및

상기 도출된 예측 벡터를 이용하여 상기 코딩된 비디오 블록을 디코딩하는 단계

를 포함하는 코딩된 비디오 블록의 디코딩 방법.

발명의 설명

기술 분야

[0001] 관련 출원에 대한 상호 참조

[0002] 본 출원은 2014년 6월 19일에 출원된 미국 가특허 출원 제62/014,664호의 정식 출원이고, 35 U.S.C. § 119(e)

하에 이러한 가특허 출원으로부터의 우선권을 주장한다. 이 출원의 내용은 그 전체가 본 명세서에 참고로 통합된다.

배경 기술

[0003] 지난 20년에 걸쳐, 효율적인 디지털 비디오 통신, 분배 및 소비를 가능하게 하기 위해 다양한 디지털 비디오 압축 기술이 개발되고 표준화되었다. 상업적으로 널리 배포된 표준의 대부분은 H.261, MPEG-1, MPEG-2 H.263, MPEG-4(part-2) 및 H.264/AVC(MPEG-4 part 10 Advance Video Coding)와 같이 ISO/IEC 및 ITU-T에 의해 개발되어 있다. 새로운 고급 비디오 압축 기술의 출현(emergence) 및 성숙(maturity)으로 인해, ITU-T VCEG(Video Coding Experts Group) 및 ISO/IEC MPEG에 의한 공동 개발 하에 새로운 비디오 코딩 표준인 HEVC(High Efficiency Video Coding)가 개발되었다. HEVC(ITU-T H.265/ISO/IEC 23008-2)는 2013년 초에 국제 표준으로 승인되었고, 현재 최첨단 H.264/AVC보다 실질적으로 더 높은 코딩 효율을 달성할 수 있다.

[0004] 스크린 콘텐츠 공유 애플리케이션은 원격 데스크톱, 비디오 회의 및 모바일 미디어 프레젠테이션 애플리케이션의 확산(proliferation)에 따라 최근에 점점 더 대중화되었다. 양방향 스크린 콘텐츠 공유 시스템은 캡처 장치, 인코더 및 송신기를 포함하는 호스트 서버 시스템과, 수신기, 디코더 및 디스플레이(렌더러(renderer))를 포함하는 클라이언트 서버 시스템을 포함할 수 있다. 스크린 콘텐츠 코딩(screen content coding; SCC)을 위한 산업에서 다양한 애플리케이션 요구 사항이 있다. 자연적(natural) 비디오 콘텐츠와 비교할 때, 스크린 콘텐츠는 종종 스크린 콘텐츠에 자주 나타나는 선명한 곡선과 텍스트 때문에 여러 주요 색상과 강한 에지(strong edge)가 있는 수많은 블록을 포함한다. 기존의 비디오 압축 방법이 스크린 콘텐츠를 인코딩하여 해당 콘텐츠를 수신기 측으로 전송하는데 이용될 수 있지만, 대부분의 기존의 방법은 스크린 콘텐츠의 특성을 수용하지 못하며, 따라서 압축 성능이 낮다. 종래의 비디오 코딩 기술을 이용한 스크린 콘텐츠의 재구성은 종종 심각한 품질 문제로 이어진다. 예를 들어, 곡선과 텍스트가 선명하지 않아 인식하기가 어려울 수 있다. 따라서, 스크린 콘텐츠를 효과적으로 재구성하기 위해 잘 설계된 스크린 콘텐츠 압축 방법이 바람직하다.

발명의 내용

과제의 해결 수단

[0005] 일부 예시적인 실시예에서, 입력 비디오 블록을 포함하는 비디오를 인코딩하는 비트 스트림을 생성하기 위한 방법이 제공된다. 인코더는 입력 비디오 블록의 예측을 위한 적어도 제 1 후보 블록 벡터(block vector; BV)를 식별하며, 여기서 제 1 후보 블록 벡터는 제 1 후보 블록을 나타낸다. 그 다음, 인코더는 제 1 후보 블록을 인코딩하는데 사용된 제 1 예측 벡터(예를 들어, 블록 벡터 또는 움직임 벡터(motion vector))를 식별한다. 제 1 후보 블록 벡터 및 제 1 예측 벡터로부터, 인코더는 제 1 후보 블록 벡터 및 제 1 예측 벡터로부터 도출된 예측 벡터(예를 들어, 도출된 블록 벡터 또는 도출된 움직임 벡터)를 생성한다. 그 다음, 인코더는 입력 비디오 블록의 예측을 위해 도출된 예측 벡터를 사용하여 비트 스트림에서 비디오 블록을 인코딩한다.

[0006] 일부 실시예에서, 인코더는 비트 스트림에서 도출된 예측 벡터를 시그널링한다. 일부 실시예에서, 인코더는 비트 스트림에서 제 1 예측 벡터를 시그널링하고, 또한 입력 비디오 블록이 제 1 예측 벡터로부터 도출되는 도출된 예측 벡터를 사용하여 인코딩됨을 나타내는 플래그를 비트 스트림에서 시그널링한다.

[0007] 일부 실시예에서, 인코더는 병합(merge) 후보 리스트에서 도출된 예측 벡터를 식별하는 인덱스를 비트 스트림에서 시그널링한다.

[0008] 도출된 예측 벡터는 제 1 후보 블록 벡터 및 제 1 예측 벡터를 부가함으로써 생성될 수 있다. 이러한 실시예에서, 제 1 예측 벡터가 제 2 블록 벡터인 경우, 도출된 예측 벡터는 제 1 후보 블록 벡터 및 제 2 블록 벡터(제 1 예측 벡터)를 부가함으로써 생성된 블록 벡터일 수 있다. 제 1 예측 벡터가 움직임 벡터인 경우, 도출된 예측 벡터는 다음 식에 따라 제 1 후보 블록 벡터 및 제 1 움직임 벡터를 부가함으로써 생성된 움직임 벡터일 수 있다:

[0009]
$$MVd = BV0 + ((MV1 + 2) \gg 2)$$

[0010] 여기서 BV0는 제 1 후보 블록 벡터이고, MV1은 제 1 움직임 벡터이며, MVd는 도출된 움직임 벡터이다.

[0011] 일부 예시적인 실시예에서, 도출된 예측 벡터(블록 벡터 또는 움직임 벡터)는 병합 후보로서 사용된다. 예시적인 방법에서, 인코더는 입력 비디오 블록의 인코딩을 위해 적어도 제 1 블록 벡터 병합 후보를 식별하고, 인코더는 제 1 후보 블록을 인코딩하는데 사용된 제 1 예측 벡터를 식별한다. 그 후, 인코더는 제 1 블록 벡터 병

합 후보 및 제 1 예측 벡터로부터 도출된 예측 벡터(도출된 블록 벡터 또는 도출된 움직임 벡터)를 생성한다. 도출된 예측 벡터는 병합 후보 리스트에 삽입된다. 병합 후보 리스트로부터, 인코더는 입력 비디오 블록의 예측을 위해 선택된 예측 벡터를 선택한다. 그 후, 인코더는 입력 비디오 블록의 예측을 위해 선택된 예측 벡터를 사용하여 비트 스트림에서 비디오 입력 블록을 인코딩한다. 선택된 예측 벡터는 도출된 예측 벡터일 수 있다.

[0012] 이러한 일부 실시예에서, 인코더는 도출된 예측 벡터를 생성하고 삽입하기 전에 병합 후보 리스트가 가득한(full)지를 결정한다. 병합 후보 리스트에서 도출된 예측 벡터를 생성하고 삽입하는 단계는 병합 후보 리스트가 가득하지 않은 것으로 결정이 이루어진 후에만 수행된다.

[0013] 이러한 일부 실시예에서, 인코더는 이전에 인코딩된 비디오 블록의 검색을 수행함으로써 제 1 후보 블록 벡터를 식별한다.

[0014] 비트 스트림으로부터 코딩된 비디오 블록을 디코딩하는 예시적인 방법에서, 디코더는 입력 비디오 블록의 예측을 위한 적어도 제 1 후보 블록 벡터를 식별하며, 여기서 제 1 후보 블록 벡터는 제 1 후보 블록을 나타낸다. 디코더는 제 1 후보 블록을 인코딩하는데 사용된 제 1 예측 벡터를 식별한다. 그 후, 디코더는 제 1 블록 벡터 및 제 1 예측 벡터로부터 도출된 예측 벡터를 생성하고, 코딩된 비디오 블록의 예측을 위해 도출된 예측 벡터를 사용하여 코딩된 비디오 블록을 디코딩한다.

[0015] 이러한 실시예에서, 제 1 후보 블록 벡터는 다양한 상이한 기술을 이용하여 식별될 수 있다. 이러한 하나의 방법에서, 제 1 후보 블록 벡터는 비트 스트림에서 시그널링되고, 제 1 후보 블록 벡터의 식별은 비트 스트림에서 시그널링된 제 1 후보 블록 벡터를 수신하는 것을 포함한다. 이러한 방법에서, 도출된 예측 벡터의 생성은 입력 비디오 블록이 도출된 예측 벡터로 인코딩된다는 것을 나타내는 플래그를 비트 스트림에서 수신하는 것에 응답하여 수행될 수 있다. 이러한 다른 방법에서, 제 1 후보 블록 벡터의 식별은 제 1 블록 벡터 병합 후보의 식별을 포함한다. 이러한 실시예에서, 도출된 예측 벡터는 또한 병합 후보일 수 있다. 디코더는 도출된 예측 벡터 병합 후보를 식별하는 비트 스트림에서 인덱스를 수신하는 것에 응답하여 코딩된 비디오 블록을 디코딩하기 위해 도출된 예측 벡터를 사용할 수 있다.

도면의 간단한 설명

[0016] 아래에서 먼저 간단히 설명되는 첨부된 다이어그램과 함께 예로서 제시된 다음의 설명으로부터 더욱 상세한 이해가 이루어질 수 있다.

도 1은 블록 기반 비디오 인코더의 일례를 예시하는 블록도이다.

도 2는 블록 기반 비디오 디코더의 일례를 예시하는 블록도이다.

도 3은 8개 방향성 예측 모드의 일례의 다이어그램이다.

도 4는 33개의 방향성 예측 모드와 2개의 무방향성 예측 모드의 일례를 예시하는 다이어그램이다.

도 5는 수평 예측의 일례의 다이어그램이다.

도 6은 평면 모드의 일례의 다이어그램이다.

도 7은 움직임 예측의 일례를 예시하는 다이어그램이다.

도 8은 화상 내의 블록 레벨 움직임의 일례를 예시하는 다이어그램이다.

도 9는 코딩된 비트 스트림 구조의 일례를 예시하는 다이어그램이다.

도 10은 예시적인 통신 시스템을 예시하는 다이어그램이다.

도 11은 예시적인 무선 송수신 유닛(wireless transmit/receive unit; WTRU)을 예시하는 다이어그램이다.

도 12는 예시적인 스크린 콘텐츠 공유 시스템을 예시하는 다이어그램이다.

도 13은 풀(full) 프레임 인트라 블록 복사 모드의 일례를 예시하는 다이어그램이다.

도 14는 로컬 영역 인트라 블록 복사 모드의 일례를 예시하는 다이어그램이다.

도 15는 인트라 블록 복사 병합을 위한 공간 후보의 2개의 예를 예시하는 다이어그램이다.

도 16은 예시적인 블록 벡터 도출을 예시하는 다이어그램이다.

도 17은 예시적인 움직임 벡터 도출을 예시하는 다이어그램이다.

도 18a 및 도 18b는 모두 예시적인 방법의 흐름도이다.

발명을 실시하기 위한 구체적인 내용

- [0017] 예시적인 실시예의 상세한 설명이 이제 다양한 도면을 참조하여 제공될 것이다. 이러한 설명이 가능한 구현에 대한 상세한 예를 제공하지만, 제공된 상세 사항은 예로서 의도되고, 적용의 범위를 결코 제한하지 않는다는 것이 주목되어야 한다.
- [0018] 도 1은 블록 기반 비디오 인코더, 예를 들어, 하이브리드 비디오 인코딩 시스템의 일례를 예시하는 블록도이다. 비디오 인코더(100)는 입력 비디오 신호(102)를 수신할 수 있다. 입력 비디오 신호(102)는 블록 단위로 처리될 수 있다. 비디오 블록은 임의의 크기의 블록일 수 있다. 예를 들어, 비디오 블록 유닛은 16×16 픽셀을 포함할 수 있다. 16×16 픽셀의 비디오 블록 유닛은 매크로 블록(macroblock; MB)으로 지칭될 수 있다. 고효율 비디오 코딩(High Efficiency Video Coding; HEVC)에서, (예를 들어, 코딩 트리 유닛(coding tree unit; CTU) 또는 코딩 유닛(coding unit; CU)으로 지칭될 수 있고, 이러한 두 용어는 본 개시물을 위해 동등한) 확장된 블록 크기는 고해상도(예를 들어, 1080p 이상) 비디오 신호를 효율적으로 압축하는데 사용될 수 있다. HEVC에서, CU는 최대 64×64 픽셀일 수 있다. CU는 별개의 예측 방법이 적용될 수 있는 예측 유닛(prediction unit; PU)으로 분할될 수 있다.
- [0019] 입력 비디오 블록(예를 들어, MB 또는 CU)에 대해, 공간적 예측(160) 및/또는 시간적 예측(162)이 수행될 수 있다. 공간적 예측(예를 들어, "인트라 예측")은 현재 비디오 블록을 예측하기 위해 동일한 비디오 화상/슬라이스에서 이미 코딩된 이웃한 블록으로부터의 픽셀을 사용할 수 있다. 공간적 예측은 비디오 신호에 내재하는 공간적 중복성(spatial redundancy)을 감소시킬 수 있다. 시간적 예측(예를 들어, "인터 예측" 또는 "움직임 보상된 예측")은 현재 비디오 블록을 예측하기 위해 (예를 들어, "참조 화상"으로 지칭될 수 있는) 이미 코딩된 비디오 화상으로부터의 픽셀을 사용할 수 있다. 시간적 예측은 비디오 신호에 내재하는 시간적 중복성을 감소시킬 수 있다. 비디오 블록에 대한 시간적 예측 신호는 현재 블록과 참조 화상에서의 예측 블록 사이의 움직임의 양 및/또는 방향을 나타낼 수 있는 하나 이상의 움직임 벡터에 의해 시그널링될 수 있다. (예를 들어, H.264/AVC 및/또는 HEVC에 대한 경우와 같이) 다수의 참조 화상이 지원되면, 비디오 블록에 대해, 이의 참조 화상 인덱스가 전송될 수 있다. 참조 화상 인덱스는 참조 화상 저장부(164)에서의 어떤 참조 화상으로부터 시간적 예측 신호가 발생하는지를 식별하는데 사용될 수 있다.
- [0020] 인코더 내의 모드 판정 블록(180)은 예를 들어 공간적 및/또는 시간적 예측 후에 예측 모드를 선택할 수 있다. 예측 블록은 116에서 현재 비디오 블록으로부터 감산될 수 있다. 예측 잔차(residual)는 변환(104)되고/되거나 양자화(106)될 수 있다. 양자화된 잔차 계수는 재구성된 잔차를 형성하기 위해 역 양자화(110)되고/되거나 역 변환(112)될 수 있으며, 이는 재구성된 비디오 블록을 형성하기 위해 예측 블록(126)에 다시 부가될 수 있다.
- [0021] 인루프 필터링(in-loop filtering)(예를 들어, 디블로킹 필터, 샘플 적응 오프셋, 적응 루프 필터 및/또는 등)은 참조 화상 저장부(164)에 배치되고/되거나 차후 비디오 블록을 코딩하는데 사용되기 전에 재구성된 비디오 블록에 적용될 수 있다(166). 비디오 인코더(100)는 출력 비디오 스트림(120)을 출력할 수 있다. 출력 비디오 비트스트림(120)을 형성하기 위해, 코딩 모드(예를 들어, 인터 예측 모드 또는 인트라 예측 모드), 예측 모드 정보, 움직임 정보 및/또는 양자화된 잔차 계수는 비트 스트림을 형성하기 위해 압축 및/또는 패킹(packing)되도록 엔트로피 코딩 유닛(108)에 전송될 수 있다. 참조 화상 저장부(164)는 디코딩된 화상 버퍼(DPB)로서 지칭될 수 있다.
- [0022] 도 2는 블록 기반 비디오 디코더의 일례를 예시하는 블록도이다. 비디오 디코더(200)는 비디오 비트 스트림(202)을 수신할 수 있다. 비디오 비트 스트림(202)은 엔트로피 디코딩 유닛(208)에서 언패킹(unpacking) 및/또는 엔트로피 디코딩될 수 있다. 비디오 비트 스트림을 인코딩하는데 사용된 코딩 모드 및/또는 예측 정보는 예측 블록을 형성하기 위해 (예를 들어, 인트라 코딩된다면) 공간적 예측 유닛(260) 및/또는 (예를 들어, 인터 코딩된다면) 시간적 예측 유닛(262)에 전송될 수 있다. 인터 코딩된다면, 예측 정보는 예측 블록 크기, (예를 들어, 움직임의 방향 및 양을 나타낼 수 있는) 하나 이상의 움직임 벡터, 및/또는 (예를 들어, 예측 신호를 획득하기 위해 어떤 참조 화상으로부터 나타낼 수 있는) 하나 이상의 참조 인덱스를 포함할 수 있다. 움직임 보상된 예측은 시간적 예측 블록을 형성하기 위해 시간적 예측 유닛(262)에 의해 적용될 수 있다.

- [0023] 잔차 변환 계수는 잔차 블록을 재구성하기 위해 역 양자화 유닛(210) 및 역변환 유닛(212)에 전송될 수 있다. 예측 블록 및 잔차 블록은 226에서 함께 부가될 수 있다. 재구성된 블록은 참조 화상 저장부(264)에 저장되기 전에 인루프 필터링(266)을 통과할 수 있다. 참조 화상 저장부(264)에서의 재구성된 비디오는 디스플레이 디바이스를 구동하는데 사용되고/되거나 차후 비디오 블록을 예측하는데 사용될 수 있다. 비디오 디코더(200)는 재구성된 비디오 신호(220)를 출력할 수 있다. 참조 화상 저장부(264)는 또한 디코딩된 화상 버퍼(DPB)로서 지칭될 수 있다.
- [0024] 비디오 인코더 및/또는 디코더(예를 들어, 비디오 인코더(100) 또는 비디오 디코더(200))는 (예를 들어, 인트라 예측으로서 지칭될 수 있는) 공간적 예측을 수행할 수 있다. 공간적 예측은 (예를 들어, 방향성 인트라 예측으로서 지칭될 수 있는) 복수의 예측 방향 중 하나를 뒤따르는 이미 코딩된 이웃한 픽셀로부터 예측함으로써 수행될 수 있다.
- [0025] 도 3은 8개의 방향성 예측 모드의 일례의 다이어그램이다. 도 3의 8개의 방향성 예측 모드는 H.264/AVC에서 지원될 수 있다. 일반적으로 도 3의 300에 도시된 바와 같이, (DC 모드 2를 포함하는) 9개의 모드는 다음과 같다:
- [0026] • 모드 0 수직 예측
 - [0027] • 모드 1 수평 예측
 - [0028] • 모드 2 DC 예측
 - [0029] • 모드 3 대각선 왼쪽 아래 예측
 - [0030] • 모드 4 대각선 오른쪽 아래 예측
 - [0031] • 모드 5 수직 오른쪽 예측
 - [0032] • 모드 6 수평 아래 예측
 - [0033] • 모드 7 수직 왼쪽 예측
 - [0034] • 모드 8 수평 위 예측
- [0035] 공간적 예측은 다양한 크기 및/또는 형상의 비디오 블록 상에서 수행될 수 있다. 비디오 신호의 휘도 성분(luma component)의 공간적 예측은 예를 들어 (예를 들어, H.264/AVC에서) 4×4, 8×8 및 16×16 픽셀의 블록 크기에 대해 수행될 수 있다. 비디오 신호의 채도 성분(chroma component)의 공간적 예측은 예를 들어 (예를 들어, H.264/AVC에서) 8×8의 블록 크기에 대해 수행될 수 있다. 크기 4×4 또는 8×8의 휘도 블록에 대해, 총 9개의 예측 모드, 예를 들어, (예를 들어, H.264/AVC에서) 8개의 방향성 예측 모드 및 DC 모드가 지원될 수 있다. 4개의 예측 모드는 예를 들어, 16×16 크기의 휘도 블록에 대해 수평, 수직, DC 및 평면 예측 모드가 지원될 수 있다.
- [0036] 더욱이, 방향성 인트라 예측 모드 및 무방향성 예측 모드가 지원될 수 있다.
- [0037] 도 4는 33개의 방향성 예측 모드 및 2개의 무방향성 예측 모드의 일례를 예시하는 다이어그램이다. 일반적으로 도 4의 400에 도시된 33개의 방향성 예측 모드 및 2개의 무방향성 예측 모드가 HEVC에 의해 지원될 수 있다. 더욱 큰 블록 크기를 사용하는 공간적 예측이 지원될 수 있다. 예를 들어, 공간적 예측은 임의의 크기, 예를 들어 4×4, 8×8, 16×16, 32×32 또는 64×64의 정사각형 블록 크기의 블록 상에서 수행될 수 있다. (예를 들어, HEVC에서) 방향성 인트라 예측은 1/32 픽셀 정밀도로 수행될 수 있다.
- [0038] 예를 들어, 방향성 인트라 예측에 부가하여, (예를 들어, H.264/AVC, HEVC 등에서) 무방향성 인트라 예측 모드가 지원될 수 있다. 무방향성 인트라 예측 모드는 DC 모드 및/또는 평면 모드를 포함할 수 있다. DC 모드에 대해, 이용 가능한 이웃한 픽셀을 평균함으로써 예측 값이 획득될 수 있고, 예측 값은 전체 블록에 균일하게 적용될 수 있다. 평면 모드에 대해, 선형 보간은 느린 전환(slow transition)으로 평활한 영역(smooth regions)을 예측하는데 이용될 수 있다. H.264/AVC는 16×16 휘도 블록 및 채도 블록에 대한 평면 모드를 사용할 수 있다.

[0039] 인코더(예컨대, 인코더(100))는 비디오 블록에 대한 최상의 코딩 모드를 결정하기 위해 (예를 들어, 도 1의 블록(180)에서) 모드 관정을 수행할 수 있다. 인코더가 (예를 들어, 인터 예측 대신에) 인트라 예측을 적용하도록 결정할 때, 인코더는 이용 가능한 모드의 세트로부터 최적의 인트라 예측 모드를 결정할 수 있다. 선택된 방향성 인트라 예측 모드는 입력 비디오 블록 내의 임의의 텍스처, 에지 및/또는 구조의 방향에 대한 강한 암시(hint)를 제공할 수 있다.

[0040] 도 5는 일반적으로 도 5에서 500으로 도시된 바와 같은 (예를 들어, 4×4 블록에 대한) 수평 예측의 일례의 다이어그램이다. 이미 채구성된 픽셀 P0, P1, P2 및 P3(즉, 음영 박스)은 현재 4×4 비디오 블록 내의 픽셀을 예측하는데 사용될 수 있다. 수평 예측에서, 채구성된 픽셀, 예를 들어 픽셀 P0, P1, P2 및/또는 P3은 4×4 블록을 예측하기 위해 대응하는 행의 방향을 따라 수평으로 전파될 수 있다. 예를 들어, L(x,y)가 (x,y), x,y = 0...3에서 예측되는 픽셀일 수 있는 아래의 식 (1)에 따라 예측이 수행될 수 있다.

[0041] $L(x,0) = P0$

[0042] $L(x,1) = P1$ (1)

[0043] $L(x,2) = P2$

[0044] $L(x,3) = P3$

[0045] 도 6은 일반적으로 도 6의 600에 도시된 바와 같은 평면 모드의 일례의 다이어그램이다. 평면 모드는 이에 따라 수행될 수 있다: (T에 의해 표시된) 최상단 행의 가장 오른쪽 픽셀은 가장 오른쪽 열의 픽셀을 예측하기 위해 복제될 수 있다. (L에 의해 표시된) 왼쪽 열의 최하단 픽셀은 최하단 행의 픽셀을 예측하기 위해 복제될 수 있다. (왼쪽 블록에 도시된 바와 같은) 수평 방향의 이중 선형 보간은 중심 픽셀의 제 1 예측 H(x,y)를 생성하기 위해 수행될 수 있다. (예를 들어, 오른쪽 블록에 도시된 바와 같은) 수직 방향의 이중 선형 보간은 중심 픽셀의 제 2 예측 V(x,y)를 생성하기 위해 수행될 수 있다. 수평 예측과 수직 예측 사이의 평균은 $L(x,y) = ((H(x,y) + V(x,y)) >> 1)$ 를 이용하여 최종 예측 L(x,y)를 획득하기 위해 수행될 수 있다.

[0046] 도 7 및 도 8은 일반적으로 700 및 800에 도시된 바와 같이 (예를 들어, 도 1의 시간적 예측 유닛(162)을 사용하여) 비디오 블록의 움직임 예측의 일례를 예시하는 다이어그램이다. 화상 내의 블록 레벨 움직임의 일례를 예시하는 도 8은 예를 들어 참조 화상 "Ref pic 0", "Ref pic 1" 및 "Ref pic 2"를 포함하는 예시적인 디코딩된 화상 버퍼를 예시하는 다이어그램이다. 현재 화상의 블록 B0, B1 및 B2는 각각 참조 화상 "Ref pic 0", "Ref pic 1" 및 "Ref pic 2" 내의 블록으로부터 예측될 수 있다. 움직임 예측은 현재 비디오 블록을 예측하기 위해 이웃한 비디오 프레임으로부터의 비디오 블록을 사용할 수 있다. 움직임 예측은 시간적 상관을 활용하고/하거나, 비디오 신호에 내재하는 시간적 중복성을 제거할 수 있다. 예를 들어, H.264/AVC 및 HEVC에서, 시간적 예측은 다양한 크기의 비디오 블록 상에서 수행될 수 있다(예를 들어, 휘도 성분에 대해, 시간적 예측 블록 크기는 H.264/AVC에서는 16×16에서 4×4까지, HEVC에서는 64×64에서 4×4까지 다양할 수 있다). (mvx, mvy)의 움직임 벡터로, 시간적 예측은 다음의 식 (2)에 의해 제공되는 바와 같이 수행될 수 있다:

[0047] $P(x,y) = \text{ref}(x - \text{mvx}, y - \text{mvy})$ (2)

[0048] 여기서 ref(x,y)는 참조 화상의 위치(x,y)에서의 픽셀 값일 수 있고, P(x,y)는 예측된 블록일 수 있다. 비디오 코딩 시스템은 단편적인 픽셀 정밀도(fractional pixel precision)로 인터 예측을 지원할 수 있다. 움직임 벡터(mvx, mvy)가 단편적인 픽셀 값을 가질 때, 단편적인 픽셀 위치에서 픽셀 값을 획득하기 위해 하나 이상의 보간 필터가 적용될 수 있다. 블록 기반 비디오 코딩 시스템은 예를 들어 예측 신호가 상이한 참조 화상으로부터의 다수의 예측 신호를 조합함으로써 형성될 수 있는 시간적 예측을 개선하기 위해 다중 가설 예측을 사용할 수 있다. 예를 들어, H.264/AVC 및/또는 HEVC는 2개의 예측 신호를 조합할 수 있는 양방향 예측을 사용할 수 있다. 양방향 예측은 다음의 식 (3)과 같은 예측을 형성하기 위해 참조 화상으로부터의 각각의 2개의 예측 신호를 조합할 수 있다:

[0049] $P(x,y) = \frac{P_0(x,y) + P_1(x,y)}{2} = \frac{\text{ref}_0(x - \text{mvx}_0, y - \text{mvy}_0) + \text{ref}_1(x - \text{mvx}_1, y - \text{mvy}_1)}{2}$ (3)

[0050] 여기서, P₀(x,y) 및 P₁(x,y)는 각각 제 1 및 제 2 예측 블록일 수 있다. 식 (3)에 예시된 바와 같이, 각각 2개의 움직임 벡터(mvx₀, mvy₀) 및 (mvx₁, mvy₁)로 2개의 참조 화상(ref₀(x,y)) 및 (ref₁(x,y))로부터의 움직임 보상된 예측을 수행함으로써 2개의 예측 블록이 획득될 수 있다. 예측 블록 P(x,y)는 예측 잔차 블록을 형성하기

위해 (예를 들어, 116에서) 소스 비디오 블록으로부터 감소될 수 있다. 예측 잔차 블록은 (예를 들어, 변환 유닛(104)에서) 변환되고/되거나 (예를 들어, 양자화 유닛(106)에서) 양자화될 수 있다. 양자화된 잔차 변환 계수 블록은 비트 레이트를 감소시키기 위해 엔트로피 코딩되도록 엔트로피 코딩 유닛(예컨대, 엔트로피 코딩 유닛(108))에 전송될 수 있다. 엔트로피 코딩된 잔차 계수는 출력 비디오 비트 스트림(예를 들어, 비트 스트림(120))의 일부를 형성하도록 패키징될 수 있다.

[0051] 단층 비디오 인코더는 단일 비디오 시퀀스 입력을 취하고, 단층 디코더에 전송되는 단일 압축된 비트 스트림을 생성할 수 있다. 비디오 코덱은 (예를 들어, 위성, 케이블 및 지상 전송 채널을 통해 TV 신호를 전송하는 것과 같지만, 이에 제한되지 않는) 디지털 비디오 서비스를 위해 설계될 수 있다. 이기종(heterogeneous) 환경에서 배치된 비디오 중심 애플리케이션으로, 다양한 애플리케이션을 가능하게 하도록 비디오 코딩 표준의 확장으로서 다층 비디오 코딩 기술이 개발될 수 있다. 예를 들어, 스케일러블(scalable) 비디오 코딩 및/또는 멀티 뷰(multi-view) 비디오 코딩과 같은 다층 비디오 코딩 기술은 각각의 층이 특정 공간적 해상도, 시간적 해상도, 충실도(fidelity) 및/또는 뷰(view)의 비디오 신호를 재구성하기 위해 디코딩될 수 있는 하나 이상의 비디오 층을 처리하도록 설계될 수 있다. 단층 인코더 및 디코더가 도 1 및 도 2를 참조하여 설명되지만, 본 명세서에 설명된 개념은 예를 들어 멀티 뷰 및/또는 스케일러블 코딩 기술에 대해 다층 인코더 및/또는 디코더를 이용할 수 있다.

[0052] 스케일러블 비디오 코딩은 이기종 네트워크를 통해 상이한 능력을 가진 디바이스 상에서 실행하는 비디오 애플리케이션에 대한 경험의 품질을 향상시킬 수 있다. 스케일러블 비디오 코딩은 최고 표현(예를 들어, 시간적 해상도, 공간적 해상도, 품질 등)에서 신호를 한번 인코딩할 수 있지만, 클라이언트 디바이스 상에서 실행하는 어떤 애플리케이션이 필요로 하는 특정 레이트 및 표현에 따라 비디오 스트림의 서브셋으로부터의 디코딩을 가능하게 한다. 스케일러블 비디오 코딩은 언스케일러블(non-scalable) 솔루션에 비해 대역폭 및/또는 스토리지를 절약할 수 있다. 국제 비디오 표준, 예를 들어 MPEG-2 Video, H.263, MPEG4 Visual, H.264 등은 확장성(scalability)의 모드를 지원하는 도구 및/또는 프로파일들을 가질 수 있다.

[0053] 표 1은 이들을 지원할 수 있는 대응하는 표준과 함께 상이한 타입의 확장성의 일례를 제공한다. 비트 깊이 확장성 및/또는 채도 포맷 확장성은 예를 들어 프로페셔널(professional) 비디오 애플리케이션에 의해 주로 사용될 수 있는 비디오 포맷(예를 들어, 8비트 비디오보다 높고, YUV4:2:0보다 높은 채도 샘플링 포맷)에 연결될 수 있다. 중형비 확장성이 제공될 수 있다.

표 1

확장성	예	표준
뷰 확장성	2D→3D (2 이상의 뷰)	MVC, MFC, 3DV
공간적 확장성	720p→1080p	SVC, 스케일러블 HEVC
품질(SNR) 확장성	35dB→38dB	SVC, 스케일러블 HEVC
시간적 확장성	30fps→60fps	H.264/AVC, SVC 스케일러블 HEVC
표준 확장성	H.264/AVC→HEVC	3DV, 스케일러블 HEVC
비트 깊이 확장성	8비트 비디오→10비트 비디오	스케일러블 HEVC
채도 포맷 확장성	YUV4:2:0→YUV4:2:2, YUV4:4:4	스케일러블 HEVC
색 재현율 확장성	BT.709→BT.2020	스케일러블 HEVC
중형비 확장성	4:3→16:9	스케일러블 HEVC

[0055]

[0056] 스케일러블 비디오 코딩은 기본 계층 비트 스트림을 사용하여 제 1 비디오 파라미터 세트와 연관된 제 1 비디오 품질 레벨을 제공할 수 있다. 스케일러블 비디오 코딩은 하나 이상의 향상 계층(enhancement layer) 비트 스트림을 사용하여 하나 이상의 향상된 파라미터 세트와 연관된 하나 이상의 고품질(higher quality) 레벨을 제공할 수 있다. 비디오 파라미터 세트는 공간적 해상도, 프레임 레이트, (예를 들어, SNR, PSNR, VQM, 시각적 품질 등의 형태의) 재구성된 비디오 품질, (예를 들어, 둘 이상의 뷰를 가진) 3D 능력, 휘도 및 채도 비트 깊이, 채도 포맷 및 기본 단층 코딩 표준 중 하나 이상을 포함할 수 있다. 상이한 사용 케이스는 예를 들어 표 1에 예시된 바와 같이 상이한 타입의 확장성을 사용할 수 있다. 스케일러블 코딩 아키텍처는 하나 이상의 확장성(예를 들어, 표 1에 나열된 확장성)을 지원하도록 구성될 수 있는 공통 구조를 제공할 수 있다. 스케일러블

코딩 아키텍처는 최소 구성 노력으로 상이한 확장성을 지원하기 위해 유연할 수 있다. 코딩 로직(예를 들어, 인코딩 및/또는 디코딩 로직)이 스케일러블 코딩 시스템 내에서 최대로 재사용될 수 있도록 스케일러블 코딩 아키텍처는 블록 레벨 동작으로의 변경을 필요로 하지 않을 수 있는 적어도 하나의 바람직한 동작 모드를 포함할 수 있다. 예를 들어, 화상 레벨 계층간 처리 및 관리 유닛에 기초한 스케일러블 코딩 아키텍처가 제공될 수 있고, 계층간 예측은 화상 레벨에서 수행될 수 있다.

[0057] 도 9는 코딩된 비트 스트림 구조의 일례를 예시하는 다이어그램이다. 코딩된 비트 스트림(1000)은 다수의 NAL(Network Abstraction layer) 유닛(1001)으로 구성된다. NAL 유닛은 코딩된 슬라이스(1006)와 같은 코딩된 샘플 데이터, 또는 파라미터 세트 데이터, 슬라이스 헤더 데이터(1005) 또는 (SEI 메시지로써 지칭될 수 있는) 보충 강화 정보(supplemental enhancement information) 데이터(1007)와 같은 고레벨의 선택스 메타데이터를 포함할 수 있다. 파라미터 세트는 필수 선택스 요소(syntax element)를 포함하는 고레벨의 선택스 구조이며, 이러한 선택스 구조는 다중 비트 스트림 계층(예를 들어 비디오 파라미터 세트(1002)(VPS))에 적용할 수 있거나, 하나의 계층 내의 코딩된 비디오 시퀀스(예를 들어 시퀀스 파라미터 세트(1003)(SPS))에 적용할 수 있으며, 또는 하나의 코딩된 비디오 시퀀스 내의 다수의 코딩된 화상(예를 들어 화상 파라미터 세트(1004)(PPS))에 적용할 수 있다. 파라미터 세트는 비디오 비트 스트림의 코딩된 화상과 함께 전송되거나, (신뢰 가능한 채널, 하드 코딩 등을 사용한 대역외 전송을 포함하는) 다른 수단을 통해 전송될 수 있다. 슬라이스 헤더(1005)는 또한 어떤 슬라이스 또는 화상 타입에 대해서만 상대적으로 작거나 관련이 있는 몇몇 화상 관련 정보를 포함할 수 있는 고레벨의 선택스 구조이다. SEI 메시지(1007)는 디코딩 프로세스에 의해 필요하지 않을 수 있지만, 화상 출력 타이밍 또는 디스플레이뿐만 아니라 손실 탐지 및 숨김과 같은 다양한 다른 목적을 위해 사용될 수 있는 정보를 반송한다.

[0058] 도 10은 통신 시스템의 일례를 예시하는 다이어그램이다. 통신 시스템(1300)은 인코더(1302), 통신 네트워크(1304) 및 디코더(1306)를 포함할 수 있다. 인코더(1302)는 유선 접속부 또는 무선 접속부일 수 있는 접속부(1308)를 통해 네트워크(1304)와 통신할 수 있다. 인코더(1302)는 도 1의 블록 기반 비디오 인코더와 유사할 수 있다. 인코더(1302)는 단층 코덱(예를 들어, 도 1) 또는 다층 코덱을 포함할 수 있다. 예를 들어, 인코더(1302)는 화상 레벨 ILP를 지원하는 다층(예를 들어, 2 계층) 스케일러블 코딩 시스템일 수 있다. 디코더(1306)는 유선 접속부 또는 무선 접속부일 수 있는 접속부(1310)를 통해 네트워크(1304)와 통신할 수 있다. 디코더(1306)는 도 2의 블록 기반 비디오 디코더와 유사할 수 있다. 디코더(1306)는 단층 코덱(예를 들어, 도 2) 또는 다층 코덱을 포함할 수 있다. 예를 들어, 디코더(1306)는 화상 레벨 ILP를 지원하는 다층(예를 들어, 2 계층) 스케일러블 디코딩 시스템일 수 있다.

[0059] 인코더(1302) 및/또는 디코더(1306)는 디지털 텔레비전, 무선 방송 시스템, 네트워크 요소/단말기, (예를 들어, HTTP(Hypertext Transfer Protocol) 서버와 같은) 콘텐츠 또는 웹 서버와 같은 서버, PDA(personal digital assistant), 랩톱 또는 데스크톱 컴퓨터, 태블릿 컴퓨터, 디지털 카메라, 디지털 기록 디바이스, 비디오 게임 디바이스, 비디오 게임 콘솔, 셀룰러 또는 위성 라디오 전화, 디지털 미디어 플레이어 등과 같지만, 이에 한정되지 않는 매우 다양한 유선 통신 디바이스 및/또는 무선 송수신 유닛(wireless transmit/receive unit; WTRU)에 통합될 수 있다.

[0060] 통신 네트워크(1304)는 적절한 타입의 통신 네트워크일 수 있다. 예를 들어, 통신 네트워크(1304)는 음성, 데이터, 비디오, 메시징, 방송 등과 같은 콘텐츠를 다수의 무선 사용자에게 제공하는 다중 액세스 시스템일 수 있다. 통신 네트워크(1304)는 다수의 무선 사용자가 무선 대역폭을 포함하는 시스템 자원의 공유를 통해 이러한 콘텐츠에 액세스하도록 할 수 있다. 예를 들어, 통신 네트워크(1304)는 CDMA(code division multiple access), TDMA(time division multiple access), FDMA(frequency division multiple access), OFDMA(orthogonal FDMA), SC-FDMA(single-carrier FDMA) 등과 같은 하나 이상의 채널 액세스 방법을 채용할 수 있다. 통신 네트워크(1304)는 다수의 접속된 통신 네트워크를 포함할 수 있다. 통신 네트워크(1304)는 인터넷, 및/또는 셀룰러 네트워크, WiFi 핫스팟, 인터넷 서비스 제공자(Internet Service Provider; ISP) 네트워크 등과 같은 하나 이상의 사설 상용 네트워크를 포함할 수 있다.

[0061] 도 11은 예시적인 WTRU의 시스템도이다. 도시된 바와 같이, 예시적인 WTRU(1202)는 프로세서(1218), 송수신기(1220), 송수신 요소(1222), 스피커/마이크로폰(1224), 키패드 또는 키보드(1226), 디스플레이/터치 패드(1228), 비이동식 메모리(1230), 이동식 메모리(1232), 전원(1234), GPS(global positioning system) 칩셋(1236) 및/또는 다른 주변 장치(1238)를 포함할 수 있다. WTRU(1202)는 실시예와 일치하면서 상술한 요소의 임의의 서브 조합을 포함할 수 있다는 것이 이해될 것이다. 더욱이, 인코더(예를 들어, 인코더(100)) 및/또는 디코더(예를 들어, 디코더(200))가 통합되는 단말기는 도 11의 WTRU(1202)를 참조하여 본 명세서에 도시되고 설명

된 요소의 일부 또는 전부를 포함할 수 있다.

- [0062] 프로세서(1218)는 범용 프로세서, 특수 목적 프로세서, 종래의 프로세서, 디지털 신호 프로세서(digital signal processor; DSP), 그래픽 처리 유닛(graphics processing unit; GPU), 복수의 마이크로 프로세서, DSP 코어와 관련된 하나 이상의 마이크로 프로세서, 제어기, 마이크로 제어기, ASIC(Application Specific Integrated Circuit), FPGA(Field Programmable Gate Array) 회로, 임의의 다른 타입의 집적 회로(integrated circuit; IC), 상태 머신 등일 수 있다. 프로세서(1218)는 신호 코딩, 데이터 처리, 전력 제어, 입출력 처리, 및/또는 WTRU(1500)가 유선 및/또는 무선 환경에서 동작하도록 할 수 있는 임의의 다른 기능을 수행할 수 있다. 프로세서(1218)는 송수신 요소(1222)에 결합될 수 있는 송수신기(1220)에 결합될 수 있다. 도 11은 프로세서(1218) 및 송수신기(1220)를 별개의 구성 요소로서 도시하지만, 프로세서(1218) 및 송수신기(1220)는 전자 패키지 및/또는 칩 내에 함께 통합될 수 있다는 것이 이해될 것이다.
- [0063] 송수신 요소(1222)는 무선 인터페이스(1215)를 통해 신호를 다른 단말기로 송신하고/하거나 다른 단말로부터 신호를 수신하도록 구성될 수 있다. 예를 들어, 하나 이상의 실시예에서, 송수신 요소(1222)는 RF 신호를 송신하고/하거나 수신하도록 구성된 안테나일 수 있다. 하나 이상의 실시예에서, 송수신 요소(1222)는 예를 들어 IR, UV 또는 가시광 신호를 송신하고/하거나 수신하도록 구성된 이미터/검출기일 수 있다. 하나 이상의 실시예에서, 송수신 요소(1222)는 RF 및 광 신호 둘 다를 송신하고/하거나 수신하도록 구성될 수 있다. 송수신 요소(1222)는 무선 신호의 임의의 조합을 송신하고/하거나 수신하도록 구성될 수 있다는 것이 이해될 것이다.
- [0064] 게다가, 송수신 요소(1222)가 도 11에서 단일 요소로서 도시되어 있지만, WTRU(1202)는 많은 송수신 요소(1222)를 포함할 수 있다. 특히, WTRU(1202)는 MIMO 기술을 채용할 수 있다. 따라서, 일 실시예에서, WTRU(1202)는 무선 인터페이스(1215)를 통해 무선 신호를 송수신하기 위한 2개 이상의 송수신 요소(1222)(예를 들어, 다중 안테나)를 포함할 수 있다.
- [0065] 송수신기(1220)는 송수신 요소(1222)에 의해 송신될 수 있는 신호를 변조하고/하거나, 송수신 요소(1222)에 의해 수신되는 신호를 복조하도록 구성될 수 있다. 상술한 바와 같이, WTRU(1202)는 다중 모드 능력을 가질 수 있다. 따라서, 송수신기(1220)는 예를 들어 UTRA 및 IEEE 802.11과 같은 다수의 RAT를 통해 WTRU(1500)가 전달할 수 있도록 하기 위한 다수의 송수신기를 포함할 수 있다.
- [0066] WTRU(1202)의 프로세서(1218)는 스피커/마이크로폰(1224), 키패드(1226), 및/또는 디스플레이/터치 패드(1228)(예를 들어, LCD(liquid crystal display) 디스플레이 유닛 또는 유기 발광 다이오드(organic light-emitting diode; OLED) 디스플레이 유닛)에 결합될 수 있고, 이로부터 사용자 입력 데이터를 수신할 수 있다. 프로세서(1218)는 또한 사용자 데이터를 스피커/마이크로폰(1224), 키패드(1226) 및/또는 디스플레이/터치 패드(1228)로 출력할 수 있다. 게다가, 프로세서(1218)는 비이동식 메모리(1230) 및/또는 이동식 메모리(1232)와 같은 임의의 타입의 적절한 메모리로부터의 정보에 액세스하고, 데이터를 이러한 메모리 내에 저장할 수 있다. 비이동식 메모리(1230)는 RAM(random-access memory), ROM(read-only memory), 하드 디스크 또는 임의의 다른 타입의 메모리 저장 디바이스를 포함할 수 있다. 이동식 메모리(1232)는 SIM(subscriber identity module) 카드, 메모리 스틱, 보안 디지털(secure digital; SD) 메모리 카드 등을 포함할 수 있다. 하나 이상의 실시예에서, 프로세서(1218)는 서버 또는 가정용 컴퓨터(도시되지 않음)와 같이 WTRU(1202) 상에 물리적으로 위치되지 않는 메모리로부터의 정보에 액세스하고, 데이터를 이러한 메모리에 저장할 수 있다.
- [0067] 프로세서(1218)는 전원(1234)으로부터 전력을 수신할 수 있고, 전력을 WTRU(1202) 내의 다른 구성 요소에 분배하고/하거나 제어하도록 구성될 수 있다. 전원(1234)은 WTRU(1202)에 전력을 공급하기 위한 임의의 적절한 디바이스일 수 있다. 예를 들어, 전원(1234)은 하나 이상의 건전지(예를 들어, 니켈 카드뮴(NiCd), 니켈 아연(NiZn), 니켈 금속 수소화물(NiMH), 리튬 이온(Li-ion) 등), 태양 전지, 연료 전지 등을 포함할 수 있다.
- [0068] 프로세서(1218)는 WTRU(1202)의 현재 위치에 관한 위치 정보(예를 들어, 경도 및 위도)를 제공하도록 구성될 수 있는 GPS 칩셋(1236)에 결합될 수 있다. GPS 칩셋(1236)으로부터의 정보에 부가하거나 대신에, WTRU(1202)는 단말기(예를 들어, 기지국)로부터 무선 인터페이스(1215)를 통해 위치 정보를 수신하고/하거나, 둘 이상의 인접 기지국으로부터 수신되는 신호의 타이밍에 기초하여 위치를 결정할 수 있다. WTRU(1202)는 실시예와 일치하면서 임의의 적절한 위치 결정 방법에 의해 위치 정보를 획득할 수 있다.
- [0069] 프로세서(1218)는 또한, 부가적인 특징, 기능성 및/또는 유선 또는 무선 연결성을 제공하는 하나 이상의 소프트웨어 및/또는 하드웨어 모듈을 포함할 수 있는 다른 주변 장치(1238)에 결합될 수 있다. 예를 들어, 주변 장치(1238)는 가속도계, 방향 센서, 움직임 센서, 근접 센서, 전자 나침반, 위성 송수신기, 디지털 카메라 및/또는

비디오 레코더(예를 들어, 사진 및/또는 비디오용), 범용 직렬 버스(USB) 포트, 진동 디바이스, 텔레비전 송수신기, 핸드프리 헤드셋, Bluetooth® 모듈, 주파수 변조된(FM) 라디오 유닛, 및 디지털 뮤직 플레이어, 미디어 플레이어, 비디오 게임 플레이어 모듈, 인터넷 브라우저 등과 같은 소프트웨어 모듈을 포함할 수 있다.

[0070] 예로서, WTRU(1202)는 무선 신호를 송신하고/하거나 수신하도록 구성될 수 있으며, 사용자 장비(UE), 이동국, 고정 또는 이동 가입자 유닛, 페이지, 셀룰러 전화, 개인 휴대 정보 단말기(PDA), 스마트 폰, 랩톱, 넷북, 태블릿 컴퓨터, 개인용 컴퓨터, 무선 센서, 가전 제품, 또는 압축된 비디오 통신을 수신하고 처리할 수 있는 임의의 다른 단말기를 포함할 수 있다.

[0071] WTRU(1202) 및/또는 통신 네트워크(예를 들어, 통신 네트워크(804))는 WCDMA(wideband CDMA)를 이용하여 무선 인터페이스(1215)를 확립할 수 있는 UMTS(Universal Mobile Telecommunications System) UTRA(Terrestrial Radio Access)와 같은 무선 기술을 구현할 수 있으며, WCDMA는 HSPA(High-Speed Packet Access) 및/또는 진화된 HSPA(HSPA+)와 같은 통신 프로토콜을 포함할 수 있다. HSPA는 HSDPA(High-Speed Downlink Packet Access) 및/또는 HSUPA(High-Speed Uplink Packet Access)를 포함할 수 있다. WTRU(1202) 및/또는 통신 네트워크(예를 들어, 통신 네트워크(804))는 LTE(Long Term Evolution) 및/또는 LTE-A(LTE-Advanced)를 이용하여 무선 인터페이스(1515)를 확립할 수 있는 E-UTRA(Evolved UMTS Terrestrial Radio Access)와 같은 무선 기술을 구현할 수 있다.

[0072] WTRU(1202) 및/또는 통신 네트워크(예를 들어, 통신 네트워크(804))는 IEEE 802.16(예를 들어, WiMAX(Worldwide Interoperability for Microwave Access)), CDMA2000, CDMA2000 1X, CDMA2000 EV-DO, IS-2000(Interim Standard 2000), IS-95(Interim Standard 95), IS-856(Interim Standard 856), GSM(Global System for Mobile communication), EDGE(Enhanced Data rates for GSM Evolution), GERAN(GSM EDGE) 등과 같은 무선 기술을 구현할 수 있다. WTRU(1500) 및/또는 통신 네트워크(예를 들어, 통신 네트워크(804))는 IEEE 802.11, IEEE 802.15 등과 같은 무선 기술을 구현할 수 있다.

[0073] 도 12는 예시적인 양방향 스크린 콘텐츠 공유 시스템(1600)을 도시하는 다이어그램이다. 다이어그램은 캡처(1602), 인코더(1604) 및 송신기(1606)를 포함하는 호스트 서버 시스템을 도시한다. 도 12는 또한, (수신된 입력 비트스트림(1610)을 출력하는) 수신기(1608), 디코더(1612) 및 디스플레이(렌더러(renderer))(1618)를 포함하는 클라이언트 서버 시스템을 도시한다. 디코더(1612)는 디스플레이 화상 버퍼(1614)에 출력하여, 디코딩된 화상(1616)을 디스플레이(1618)에 차례로 전송한다. 스크린 콘텐츠 코딩(SCC)을 위한 산업으로부터의 애플리케이션 요건이 있다. [R12], [R13]을 참조한다. 점점 더 많은 사람이 미디어 프레젠테이션 또는 원격 데스크톱 목적을 위해 디바이스 콘텐츠를 공유하기 때문에 스크린 콘텐츠 압축 방법은 일부 특정 애플리케이션에 중요해지고 있다. 모바일 디바이스의 스크린 디스플레이는 고화질 또는 초고화질 해상도를 지원하도록 크게 향상되었다. 종래의 비디오 코딩 방법은 스크린 공유 애플리케이션에서 스크린 콘텐츠를 전송하기 위한 대역폭 요건을 증가시킨다.

[0074] 상술한 바와 같이, 도 2는 도 1의 인코더와 같은 인코더에 의해 생성된 비디오 비트 스트림을 수신하고, 디스플레이될 비디오 신호를 재구성하는 일반적인 블록 기반 단일 계층 디코더의 블록도이다. 또한 상술한 바와 같이, 비디오 디코더에서, 비트 스트림은 먼저 엔트로피 디코더에 의해 파싱(parsing)된다. 잔차 계수는 재구성된 잔차를 획득하기 위해 역 양자화 및 역 변환된다. 코딩 모드 및 예측 정보는 공간적 예측 또는 시간적 예측 중 어느 하나를 사용하여 예측 신호를 획득하는데 사용된다. 예측 신호 및 재구성된 잔차는 재구성된 비디오를 얻기 위해 함께 부가된다. 재구성된 비디오는 표시되고/되거나 향후 비디오 신호를 디코딩하는데 사용될 참조 화상 저장부에 저장되기 전에 루프 필터링을 부가적으로 거칠 수 있다. 도 1에 도시된 바와 같이, 효율적인 압축을 달성하기 위해, 단일 계층 인코더는 입력 비디오 신호를 예측하기 위해 공간적 예측(또한 인트라 예측으로서 지칭됨) 및 시간적 예측(또한 인터 예측 및/또는 움직임 보상된 예측으로서 지칭됨)과 같은 널리 공지된 기술을 채용한다. 인코더는 또한 일반적으로 레이트 및 왜곡 고려 사항의 조합과 같은 특정 기준에 기초하여 가장 적합한 형태의 예측을 선택하는 모드 관점 로직을 갖는다. [R11] 참조. 그 다음, 인코더는 예측 잔차(입력 신호와 예측 신호 사이의 차 신호)를 변환하고 양자화한다. 모드 정보(예를 들어, 인트라 또는 인터 예측) 및 예측 정보(움직임 벡터, 참조 화상 인덱스, 인트라 예측 모드 등)와 함께 양자화된 잔차는 엔트로피 코더에서 더 압축되고, 출력 비디오 비트 스트림으로 패키징된다. 도 1에 도시된 바와 같이, 인코더는 또한 재구성된 잔차를 획득하기 위해 역 양자화 및 역 변환을 양자화된 잔차에 적용하여, 그것을 예측 신호에 다시 부가함으로써 재구성된 비디오 신호를 생성한다. 재구성된 비디오 신호는 부가적으로 루프 필터 프로세스(예를 들어, 디블로킹 필터, 샘플 적응 오프셋 또는 적응 루프 필터)를 거쳐, 최종적으로 향후 비디오 신호를 예측하는데 사

용되도록 참조 화상 저장부에 저장될 수 있다.

[0075] 전송 대역폭과 저장 공간을 절약하기 위해, MPEG는 수년 동안 비디오 코딩 표준에 대해 작업해 왔다. 고효율 비디오 코딩(HEVC)([R13] 참조)은 떠오르는 비디오 압축 표준이다. HEVC는 현재 ITU-T VCEG(Video Coding Experts Group)과 ISO/IEC MPEG(Moving Picture Experts Group)에 의해 공동으로 개발되고 있다. 이것은 동일한 품질을 가진 H.264에 비해 50%의 대역폭을 절약할 수 있다. 인코더 및 디코더가 일반적으로 도 1 및 도 2와 관련하여 상술한 방식에 따라 동작한다는 점에서 HEVC는 여전히 블록 기반 하이브리드 비디오 코딩 표준이다. HEVC는 더 큰 비디오 블록의 사용을 허용하고, 쿼드트리 파티션(quadtrees partition)을 사용하여 블록 코딩 정보를 시그널링한다. 화상 또는 슬라이스는 먼저 동일한 크기(예를 들어, 64×64)를 갖는 코딩 트리 블록(coding tree block; CTB)으로 분할된다. 각각의 CTB는 쿼드트리를 가진 CU로 분할되고, 각각의 CU는 또한 쿼드트리를 사용하여 예측 유닛(prediction unit; PU) 및 변환 유닛(transform unit; TU)으로 더 분할된다. 각각의 인트라 코딩된 CU에 대해, PU는 도 3과 관련하여 도시되고 논의된 바와 같이 8개의 파티션 모드 중 하나일 수 있다. 또한 움직임 보상으로 불리는 시간적 예측은 모든 인트라 코딩된 PU를 재구성하는데 적용된다. (HEVC에서 최대 1/4 픽셀일 수 있는) 움직임 벡터의 정밀도에 따라, 선형 필터가 단편적인 위치에서 픽셀 값을 획득하기 위해 적용된다. HEVC에서, 보간 필터는 휘도에 대한 7개 또는 8개의 탭 및 채도에 대한 4개의 탭을 갖는다. HEVC의 더블로킹 필터는 콘텐츠 기반이고; 코딩 모드 차이, 움직임 차이, 참조 화상 차이, 픽셀 값 차이 등과 같은 다수의 인자에 따라, 상이한 더블로킹 필터 동작이 TU 및 PU 경계에 적용된다. 엔트로피 코딩의 경우, HEVC는 고레벨 파라미터를 제외한 대부분의 블록 레벨 신택스 요소에 대해 CABAC(context-based adaptive arithmetic binary coding)를 채택한다. CABAC 코딩에는 두 종류의 저장소(bin)가 있으며, 하나는 콘텍스트 기반 코딩된 일반 저장소이고, 다른 하나는 콘텍스트가 없는 바이패스(bypass) 코딩된 저장소이다.

[0076] 현재 HEVC 설계가 다양한 블록 코딩 모드를 포함하지만, 그것은 스크린 콘텐츠 코딩을 위한 공간적 중복성을 충분히 활용하지는 못한다. 이것은 HEVC가 4:2:0 포맷의 연속 톤 비디오 콘텐츠에 초점을 맞추고, 모드 판정 및 변환 코딩 도구가 종종 4:4:4 비디오의 포맷으로 캡처되는 이산 톤 스크린 콘텐츠에 대해 최적화되지 않기 때문이다. HEVC 표준이 2012년 말에 성숙하고 안정화하기 시작할 때, 표준화 기구 VCEG 및 MPEG는 스크린 콘텐츠 코딩을 위한 HEVC의 향후 확장에 작업하기 시작했다. 2014년 1월에, 스크린 콘텐츠 코딩의 CFP(Call for Proposal)가 ITU-T VCEG 및 ISO/IEC MPEG에 의해 공동으로 발급되었다. CFP는 많은 관심을 받았고, 다양한 효율적인 SCC 솔루션을 제안하는 다양한 상이한 회사로부터 7개의 응답 [R2]-[R8]이 있었다. 텍스트 및 그래픽과 같은 스크린 콘텐츠 자료가 자연적 비디오 콘텐츠에 비해 상이한 특성을 보여준다는 것을 고려하면, 스크린 콘텐츠 코딩의 코딩 효율을 향상시키는 일부 새로운 코딩 도구, 예를 들어, 1D 문자열 복사 [R9], 팔레트 코딩 [R10], [R11] 및 인트라 블록 복사(IntraBC) [R12], [R17]가 제안되었다. 핵심적인 실험 [R18]-[R22]에서 이러한 모든 스크린 콘텐츠 코딩 관련 도구가 조사되었다. 스크린 콘텐츠는 라인 세그먼트 또는 블록 및 많은 작은 균질 영역(예를 들어, 단색 영역)의 관점에서 매우 반복적인 패턴을 갖는다. 일반적으로 작은 블록 내에는 몇 가지 색상만이 존재한다. 반대로, 자연적 비디오를 위한 작은 블록에도 많은 색상이 있다. 각각의 위치에서의 색상 값은 일반적으로 위 또는 수평 인접 픽셀에서 반복된다. 1D 문자열 복사는 이전의 재구성된 픽셀 버퍼로부터의 가변 길이를 갖는 문자열을 예측하는 것을 포함한다. 위치 및 문자열 길이는 시그널링된다. 팔레트 코딩 모드에서, 픽셀 값을 직접 코딩하는 대신에, 팔레트 테이블은 사전(dictionary)으로서 중요한 색상을 기록하는데 사용된다. 그리고 대응하는 팔레트 인덱스 맵은 코딩 블록 내의 각각의 픽셀의 색상 값을 나타내는데 사용된다. 더욱이, "실행(run)" 값은 공간적 중복성을 줄이기 위해 동일한 중요한 색상(즉, 팔레트 인덱스)을 갖는 연속적 픽셀의 길이를 나타내는데 사용된다. 팔레트 코딩은 일반적으로 스파스(sparse) 색상을 포함하는 큰 블록에 적합하다. 인트라 블록 복사는 동일한 화상 내의 현재 코딩 블록을 예측하기 위해 재구성된 픽셀을 사용하는 것을 포함하고, 블록 벡터로서 지칭되는 변위 정보는 코딩된다.

[0077] 도 13은 풀 프레임 인트라 블록 복사 모드의 일례를 예시하는 다이어그램이다. 복잡성 및 대역폭 액세스를 고려하면, HEVC에서 스크린 콘텐츠 코딩 확장 참조 소프트웨어(SCM-1.0) [R31]는 인트라 블록 복사 모드에 대해 2개의 구성을 갖는다. 하나는 재구성된 모든 픽셀이 일반적으로 도 13의 1700에서 도시된 바와 같이 예측을 위해 사용될 수 있는 전체 프레임 인트라 블록 복사 모드이다. 블록 벡터 검색 복잡도를 줄이기 위해, 해시 기반 인트라 블록 복사 검색이 제안되었다 [R29], [R30]. 다른 하나는 다음에 논의되는 로컬 영역 인트라 블록 복사 모드이다.

[0078] 도 14는 일반적으로 1800에서 도시된 바와 같이 로컬 영역 인트라 블록 복사 모드의 일례를 예시하는 다이어그램이다. 로컬 영역 인트라 블록 복사 모드가 사용될 때, 왼쪽 및 현재 코딩 트리 유닛(coding tree unit; CTU)의 재구성된 픽셀만이 참조로서 사용되도록 허용된다.

- [0079] SCC와 자연적 비디오 코딩 사이에는 다른 차이가 있다. 자연적 비디오 코딩의 경우, 코딩 왜곡이 전체 화상에 분산한다. 그러나, 스크린 콘텐츠의 경우, PSNR(피크 신호 대 잡음비)이 전체 화상에 대해 상당히 높을 때에도 아티팩트를 더 잘 보이게 하는 강한 에지(strong edge) 주위에 에러가 일반적으로 국한된다. 따라서, 스크린 콘텐츠는 주관적인 품질 관점에서 인코딩하기가 더 어렵다.
- [0080] 인트라 블록 복사 모드를 사용하는 것은 블록 벡터의 시그널링을 필요로 한다. 전체 프레임 인트라 블록 복사 구성에서, 블록 벡터는 매우 커서, 인트라 블록 복사 모드에 대한 높은 오버헤드를 초래할 수 있다. 종종, 하나의 블록은 스크린 콘텐츠에 대해 매우 반복적인 패턴이 있기 때문에 다수의 유사한 매치(match)를 찾을 수 있다. 블록 벡터 코딩 효율을 향상시키기 위해, 다양한 예측 및 코딩 방법이 제안되었다 [R23]-[R28]. 현재 개시된 시스템 및 방법의 실시예는 인트라 블록 복사 코딩 효율을 향상시키기 위해 블록 벡터 도출을 이용한다. 이러한 개시물에서 논의되고 설명된 변형 중에는 (i) 인트라 블록 복사 병합 모드에서의 블록 벡터 도출 및 (ii) 명시적 블록 벡터 모드를 갖는 인트라 블록 복사에서의 블록 벡터 도출이 있다.
- [0081] 본 개시물에는 인트라 블록 복사 코딩을 위한 변위 정보 도출 방법에 대한 논의가 포함된다. 코딩 타입의 참조 블록에 따라, 도출된 블록 벡터 또는 움직임 벡터는 상이한 방식으로 사용될 수 있다. 하나의 방법은 도출된 BV를 IntraBC 병합 모드에서의 병합 후보로서 사용하는 것이다. 이러한 옵션은 제목이 "인트라 블록 복사 병합 모드(Intra Block Copy Merge Mode)"인 서브 섹션으로 아래에서 논의된다. 다른 방법은 일반적인 IntraBC 예측을 위해 도출된 BV/MV를 사용하는 것이다. 이러한 옵션은 제목이 "도출된 블록 벡터를 가진 인트라 블록 복사 모드(Intra Block Copy Mode with Derived Block Vector)"인 서브 섹션으로 아래에서 논의된다.
- [0082] 도 16은 예시적인 블록 벡터 도출을 예시하는 다이어그램이다. 블록 벡터가 주어진 경우, 주어진 BV에 의해 나타내어진 참조 블록이 IntraBC 코딩된 블록인 경우에 제 2 블록 벡터가 도출될 수 있다. 도출된 블록 벡터는 식 (4)에서 계산된다. 도 16은 일반적으로 2000에서 이러한 종류의 블록 벡터 도출을 도시한다.
- [0083]
$$BVd = BV0 + BV1 \quad (4)$$
- [0084] 도 17은 예시적인 움직임 벡터 도출을 예시하는 다이어그램이다. 주어진 BV에 의해 나타내어진 블록이 인트라 코딩된 블록이면, 움직임 벡터가 도출될 수 있다. 도 17은 일반적으로 2100에서의 MV 도출 경우를 도시한다. 도 17의 블록 B1이 단일 예측 모드인 경우, 블록 B0에 대한 정수 픽셀에서의 도출된 움직임 MVd는 다음과 같다:
- [0085]
$$MVd = BV0 + ((MV1+2)\gg 2) \quad (5)$$
- [0086] 일부 실시예에서, 1/4 픽셀 해상도의 도출된 값 MVd_q는 다음과 같이 계산된다:
- [0087]
$$MVd_q = (BV0\ll 2) + MV1$$
- [0088] 이러한 참조 화상은 B1의 참조 화상과 동일하다. HEVC에서, 보통의 움직임 벡터는 1/4 픽셀 정밀도(precision)를 갖고, 블록 벡터는 정수 정밀도(integer precision)를 갖는다. 도출된 움직임 벡터에 대한 정수 픽셀 움직임은 본 명세서에서 예로서 사용된다. 블록 B1이 양방향 예측 모드이면, 움직임 벡터 도출을 수행하는 두 방법이 있다. 하나는 두 방향에 대해 두 개의 움직임 벡터를 별개로 도출하고, 인덱스를 단방향 예측 모드와 동일하게 참조하는 것이다. 다른 하나는 더 작은 양자화 파라미터(더 높은 품질)로 참조 화상으로부터 움직임 벡터를 선택하는 것이다. 두 참조 화상이 동일한 양자화 파라미터를 갖는다면, POC(picture order of count) 거리(더 높은 상관)의 더 가까운 참조 화상으로부터 움직임 벡터를 선택할 수 있다. 다음의 논의는 복잡성을 줄이기 위해 양방향 예측을 단방향 예측으로 변환하는 제 2 방식을 이용하는 예를 사용한다.
- [0089] **인트라 블록 복사 병합 모드.**
- [0090] 도 15는 인트라 블록 복사 병합을 위한 공간적 후보의 두 개의 예를 예시하는 다이어그램이다. HEVC 메인 프로파일 [R13] 및 범위 확장 [R17]에서, 인트라 코딩 유닛 병합 모드는 움직임 정보를 직접 시그널링하지 않고, 인트라 병합 후보 리스트의 인덱스를 디코더에 시그널링할 것이다. 인트라 병합 후보 리스트는 결정적으로 인코더와 동일한 방식으로 구성될 것이다. 움직임 정보는 인덱스를 사용하여 후보 리스트로부터 도출된다. 도 15에서 1902로 번호가 매겨진 예에서, 5개의 공간적 인접한 블록과 하나의 시간적 배치된 블록이 있다. 인트라 코딩된 블록만이 인트라 병합 후보 리스트에 부가될 것이다. 후보 리스트가 공간적 및 시간적 인접한 블록으로 가득 차 있지 않으면, 2개의 리스트에서의 기존의 병합 후보와 제로 움직임을 조합함으로써 양방향 예측 움직임이 첨가될 것이다.
- [0091] 인트라 블록 복사 병합 모드의 경우, 병합 모드를 적용하는 유사한 방법이 수행된다. BV 정보는 명시적으로 코딩되지 않고, 병합 후보 인덱스는 코딩된다. HEVC SCC 확장(SCM-1.0) [R31]에서, BV는 마지막으로 코딩된 BV를

예측자(predictor)로서 사용하여 차등 코딩(differential coding)으로 코딩된다. BV 후보 리스트 구성에서, BV 예측자가 먼저 체크된다. BV 예측자가 현재 CU에 대해 유효하면, 제 1 병합 후보로서 추가된다. 그 다음, 도 15의 1902 및 1904로 번호가 매겨진 예에서, 5개의 공간적 블록이 체크되고; 유효한 BV는 (1) 공간적 인접한 블록이 IntraBC 코딩되어 BV를 갖는 경우, (2) BV가 현재 CU에 대해 유효한 경우(예를 들어, BV에 의해 나타내어진 참조 블록이 화상 경계를 벗어나지 않고 이미 코딩된 경우), 및 (3) BV가 현재 후보 리스트에 이미 나타나지 않은 경우의 순서대로 추가된다. 병합 후보 리스트가 가득 차 있지 않으면, BV는 이미 리스트 내에 있는 유효한 BV로 도출된다. 일 실시예에서, 식 (4)의 도출된 블록 벡터만이 고려되고, 식 (5)의 도출된 움직임 벡터는 고려되지 않으며; 이러한 예에서, 후보 리스트 내의 모든 병합 후보는 인트라 블록 복사 모드에 대응하는 블록 벡터이다.

[0092] 복잡한 설계에 대해, 식 (5)으로부터 도출된 움직임 벡터는 함께 혼합될 수 있고, 블록 벡터와 함께 병합 후보에 추가될 수 있다. 다른 실시예는 각각의 후보 블록 벡터(BV_o)에 대한 것이고, BV_o에 기초하여 도출된 BV_d 또는 MV_d가 유효하면, 후보 블록을 BV_o 및 BV_d/MV_d를 가진 양방향 예측 모드로서 간주할 것이며, 여기서 양방향 예측은 블록 벡터(BV_o)를 적용함으로써 획득된 제 1 예측을 도출된 블록 또는 움직임 벡터(BV_d/MV_d)를 적용함으로써 획득된 제 2 예측과 평균함으로써 획득된다.

[0093] 도 18a 및 도 18b는 모두 인트라 블록 복사 병합 후보 리스트 구성의 예시적인 방법의 흐름도이다. 예시적인 방법(2200)은 "IntraBC 병합 후보 도출"인 단계(2202)에서 시작한다. 그 다음, 방법은 "공간적 BV 후보 생성(Spatial BV Candidates Generation)"이라고 표시된 점선 박스로 진행하고, 특히 "현재 CU에 대해 유효하다면 후보 리스트를 병합하기 위해 BV 예측자를 추가함(Add the BV predictor to merge candidate list if it is valid for current CU)"인 단계(2204)로 진행한다. 그 다음, 처리는 "공간적 인접한 블록으로부터 BV를 체크하여, 그것이 유효하면 병합 후보 리스트에 추가함(Check the BV from spatial neighboring blocks, add it to merge candidate list if it is valid)"인 단계(2206)로 진행한다.

[0094] 그 다음, 방법은 다음의 조건: "((왼쪽, 최상부, 오른쪽 최상부, 왼쪽 최하부 인접한 블록이 체크됨(left, top, top right, bottom left Neighboring blocks are checked)) || (num_of_cand_list >= max_num_of_merge_cand))?"이 평가되는 판정 박스(2208)로 진행한다. 2208에서 조건이 거짓으로 결정되면, 처리는 단계(2206)로 복귀한다.

[0095] 대신에 2208에서의 조건이 참으로 결정되면, 처리는 다음의 조건: "(num_of_cand_list <max_num_of_merge_cand -1)?"이 평가되는 판정 박스(2210)로 진행한다. 2210에서의 조건이 참으로 결정되면, 처리는 "왼쪽 최상부 인접한 블록의 BV를 체크하여, 그것이 유효하면 병합 후보 리스트에 추가함(Check the BV of top left neighboring block, add it to merge candidate list if it is valid.)"인 단계(2212)로 진행한다. 2210에서의 조건이 거짓으로 결정되면, 단계(2212)는 바이패스된다. 어느 쪽이든, 그 다음 처리는 도 18b로 진행하고, 제목이 "BVd 후보 생성(BVd Candidates Generation)"인 점선 박스, 구체적으로는 다음의 조건: "((체크된 리스트 내의 모든 공간적 BV 후보(all spatial BV candidates in the list checked)) || (num_of_cand_list >= max_num_of_merge_cand)?"이 평가되는 판정 박스(2216)로 진행한다.

[0096] 2216에서의 조건이 참으로 결정되면, 처리는 2224에서 종료한다. 2216에서의 조건이 거짓으로 결정되면, 처리는 "후보 리스트로부터 하나의 공간적 BV를 취하고, BVd를 도출함(Take one spatial BV from candidate list, and derive BVd)."인 단계(2218)로 진행한다.

[0097] 다음으로, 처리는 다음의 조건: "BVd가 유효한가(BVd is valid)?"이 평가되는 판정 박스(2220)로 진행한다. 2220에서의 조건이 참으로 결정되면, 처리는 "BVd를 병합 후보 리스트에 추가함(add BVd to merge candidate list)"하는 단계(2222)로 진행한다. 대신에 2220에서의 조건이 거짓으로 결정되면, 처리는 판정 박스(2216)로 복귀한다.

[0098] **도출된 블록 벡터를 가진 인트라 블록 복사 모드.**

[0099] 정상적인 인트라 블록 복사 모드에서, 블록 벡터는 코딩 유닛 내의 각각의 예측 유닛에 대해 명시적으로 시그널링될 것이다. 일부 실시예에서, 이러한 모드는 시그널링된 블록 벡터 또는 도출된 블록 벡터가 IntraBC 예측에 사용되는지를 나타내는 플래그를 추가함으로써 확장된다. 플래그가 0이면, 시그널링된 블록 벡터는 IntraBC 예측을 위해 사용되고, BV 도출을 적용할 필요가 없다. 플래그가 1이면, BV 또는 MV는 시그널링된 블록 벡터에 기초하여 식 (4) 또는 식 (5)을 이용하여 도출되고, 도출된 BV 또는 MV는 인트라 블록 복사 예측 또는 움직임

보상된 예측을 위해 사용될 것이다.

- [0100] 다른 실시예는 2개의 플래그를 정상적인 IntraBC 모드에 추가하는 것이다. 제 1 플래그는 BV 도출 프로세스가 적용되는지 여부를 나타내는데 사용된다. 제 1 플래그가 1이면, 제 2 플래그는 단방향 예측이 사용되는지 양방향 예측이 사용되는지를 나타내기 위해 코딩된다. 제 2 플래그가 0이면, 도출된 BV 또는 MV만이 인트라 블록 복사 예측 또는 움직임 보상된 예측을 위해 사용된다. 그렇지 않으면, 제 2 플래그가 1이면, 시그널링된 BV는 제 1 예측을 생성하는데 사용되고, 도출된 BV 또는 MV는 제 2 예측을 생성하는데 사용되며; 최종 예측은 양방향 예측 모드와 유사하게 이러한 2개의 예측을 평균함으로써 생성된다.
- [0101] **블록 벡터 도출에 대한 메모리 액세스 대역폭 감소.**
- [0102] 블록 벡터 도출은 도 16 및 도 17의 블록 B1의 정보와 같은 블록 코딩 모드 및 블록 벡터/움직임 벡터에 관한 정보를 사용하여 동작한다. 디코더 칩 설계에 대해, 코딩된 모든 블록의 모드/BV/움직임 정보를 저장하는 두 가지 방식이 있다. 하나는 정보를 외부 메모리에 저장하는 것이다. 이러한 기술은 외부 메모리에 대한 액세스를 필요로 하며, 따라서 메모리 액세스 대역폭을 증가시킨다. 다른 기술은 온칩 메모리에 정보를 캐시하는 것이며, 이는 캐시 크기를 증가시킨다.
- [0103] 저장될 필요가 있는 정보의 양을 감소시키기 위한 2개의 예시적인 방법이 본 명세서에서 설명된다. 하나는 해당 정보를 거친 입도(coarse granularity)로 저장하는 것이다. HEVC에서, 원래의 BV/MV 정보는 4×4 블록 크기에 기초하여 저장된다. 메모리 크기는 더 큰 블록 크기, 예를 들어, 16×16 블록 크기에 기초하여 압축된 형태로 원래의 BV/MV 정보를 저장함으로써 크게 감소될 수 있다. 16×16 블록 크기가 사용되면, 필요한 BV 저장 공간은 HEVC에서의 압축된 움직임과 동일한 입도를 갖는다. 이러한 방식으로, 합리적인 크기로 이러한 데이터를 캐시할 수 있다. 제 2 솔루션은 코딩된 블록의 정보를 이미 코딩된 모든 블록 대신에 제한된 범위에서 캐시하는 것이다. 예를 들어, 디코더는 현재 CTU 행에 관한 정보와 현재 CTU 행 위에 있는 제한된 수의 코딩된 인접한 CTU 행만을 캐시할 수 있다. 도 16 및 도 17에서 제 1 BV에 의해 나타낸 블록(B1)이 디코더가 캐시하는 범위 밖에 있으면, 이러한 BV는 유효하지 않은 것으로 간주될 것이고, BV/MV 도출은 적용되지 않을 것이다.
- [0104] **코딩 신택스 및 세맨틱스(semantics).**
- [0105] HEVC 범위 확장 드래프트(draft) [R17]의 현재 신택스 설계에 기초하여 도출된 블록 벡터를 가진 인트라 블록 복사 병합 및 인트라 블록 복사로 코딩되는 CU를 시그널링하기 위해 새로운 신택스 요소가 제안되었다. 이러한 섹션에서 논의된 제안된 팔레트 코딩 방법은 추가적인 신택스를 도입함으로써 비트 스트림으로 시그널링될 수 있다. 다음의 표(표 2)는 제안된 신택스 요소를 나타내고, HEVC 범위 확장 드래프트 [R17]에 대한 변경 사항은 [10], [12], [13], [14], [17], [27], [28], [29], [40], [41] 및 [42]으로 번호가 매겨진 라인에 포함된다.
- [0106] **신택스**

표 2

[01]	coding_unit(x0, y0, log2CbSize) {	Descriptor
[02]	if(transquant_bypass_enabled_flag)	
[03]	cu_transquant_bypass_flag	ae(v)
[04]	if(slice_type != I)	
[05]	cu_skip_flag[x0][y0]	ae(v)
[06]	nCbS = (1 << log2CbSize)	
[07]	if(cu_skip_flag[x0][y0])	
[08]	prediction_unit(x0, y0, nCbS, nCbS)	
[09]	else {	
[10]	if(intra_block_copy_enabled_flag) {	
[11]	intra_bc_flag[x0][y0]	ae(v)
[12]	if(intra_bc_flag[x0][y0])	
[13]	intra_bc_merge_flag[x0][y0]	ae(v)
[14]	}	
[15]	if(slice_type != I && !intra_bc_flag[x0][y0])	
[16]	pred_mode_flag	ae(v)
[17]	if (CuPredMode[x0][y0] != MODE_INTRA (intra_bc_flag[x0][y0] && ! intra_bc_merge_flag[x0][y0]) log2CbSize == MinCbLog2SizeY)	
[18]	part_mode	ae(v)
[19]	if(CuPredMode[x0][y0] == MODE_INTRA) {	
[20]	if(PartMode == PART_2Nx2N && pcm_enabled_flag && !intra_bc_flag[x0][y0] && log2CbSize >= Log2MinIpcmCbSizeY && log2CbSize <= Log2MaxIpcmCbSizeY)	
[21]	pcm_flag[x0][y0]	ae(v)

[0107]

[22]	if(pcm_flag[x0][y0]) {	
[23]	while(!byte_aligned())	
[24]	pcm_alignment_zero_bit	f(1)
[25]	pcm_sample(x0, y0, log2CbSize)	
[26]	} else if(intra_bc_flag[x0][y0]) {	
[27]	if(intra_bc_merge_flag[x0][y0])	
[28]	intra_bc_merge_index[x0][y0]	ae(v)
[29]	if(!intra_bc_merge_flag[x0][y0]) {	
[30]	mvd_coding(x0, y0, 2)	
[31]	if(PartMode == PART_2NxN)	
[32]	mvd_coding(x0, y0 + (nCbS / 2), 2)	
[33]	else if(PartMode == PART_Nx2N)	
[34]	mvd_coding(x0 + (nCbS / 2), y0, 2)	
[35]	else if(PartMode == PART_NxN) {	
[36]	mvd_coding(x0 + (nCbS / 2), y0, 2)	
[37]	mvd_coding(x0, y0 + (nCbS / 2), 2)	
[38]	mvd_coding(x0 + (nCbS / 2), y0 + (nCbS / 2), 2)	
[39]	}	
[40]	if(PartMode == PART_2Nx2N)	
[41]	intra_bc_bv_derivation_flag[x0][y0]	ae(v)
[42]	}	
[43]	} else {	
[44]	pbOffset = (PartMode == PART_NxN) ? (nCbS / 2) : nCbS	
[45]	for(j = 0; j < nCbS; j = j + pbOffset)	
[46]	for(i = 0; i < nCbS; i = i + pbOffset)	
[47]	prev_intra_luma_pred_flag[x0 + i][y0 + j]	ae(v)
[48]	for(j = 0; j < nCbS; j = j + pbOffset)	
[49]	for(i = 0; i < nCbS; i = i + pbOffset)	
[50]	if(prev_intra_luma_pred_flag[x0 + i][y0 + j])	
[51]	mpm_idx[x0 + i][y0 + j]	ae(v)
[52]	else	
[53]	rem_intra_luma_pred_mode[x0 + i][y0 + j]	ae(v)
[54]	if(ChromaArrayType == 3)	
[55]	for(j = 0; j < nCbS; j = j + pbOffset)	
[56]	for(i = 0; i < nCbS; i = i + pbOffset)	
[57]	intra_chroma_pred_mode[x0 + i][y0 + j]	ae(v)
[58]	else if(ChromaArrayType != 0)	
[59]	intra_chroma_pred_mode[x0][y0]	ae(v)

[0108]

[60]	}	
[61]	} else {	
[62]	if(PartMode == PART_2Nx2N)	
[63]	prediction_unit(x0, y0, nCbS, nCbS)	
[64]	else if(PartMode == PART_2NxN) {	
[65]	prediction_unit(x0, y0, nCbS, nCbS / 2)	
[66]	prediction_unit(x0, y0 + (nCbS / 2), nCbS, nCbS / 2)	
[67]	} else if(PartMode == PART_Nx2N) {	
[68]	prediction_unit(x0, y0, nCbS / 2, nCbS)	
[69]	prediction_unit(x0 + (nCbS / 2), y0, nCbS / 2, nCbS)	
[70]	} else if(PartMode == PART_2NxnU) {	
[71]	prediction_unit(x0, y0, nCbS, nCbS / 4)	
[72]	prediction_unit(x0, y0 + (nCbS / 4), nCbS, nCbS * 3 / 4)	
[73]	} else if(PartMode == PART_2NxnD) {	
[74]	prediction_unit(x0, y0, nCbS, nCbS * 3 / 4)	
[75]	prediction_unit(x0, y0 + (nCbS * 3 / 4), nCbS, nCbS / 4)	
[76]	} else if(PartMode == PART_nLx2N) {	
[77]	prediction_unit(x0, y0, nCbS / 4, nCbS)	
[78]	prediction_unit(x0 + (nCbS / 4), y0, nCbS * 3 / 4, nCbS)	
[79]	} else if(PartMode == PART_nRx2N) {	
[80]	prediction_unit(x0, y0, nCbS * 3 / 4, nCbS)	
[81]	prediction_unit(x0 + (nCbS * 3 / 4), y0, nCbS / 4, nCbS)	
[82]	} else { /* PART_NxN */	
[83]	prediction_unit(x0, y0, nCbS / 2, nCbS / 2)	
[84]	prediction_unit(x0 + (nCbS / 2), y0, nCbS / 2, nCbS / 2)	
[85]	prediction_unit(x0, y0 + (nCbS / 2), nCbS / 2, nCbS / 2)	
[86]	prediction_unit(x0 + (nCbS / 2), y0 + (nCbS / 2), nCbS / 2, nCbS / 2)	
[87]	}	
[88]	}	
[89]	if(!pcm_flag[x0][y0]) {	
[90]	if(CuPredMode[x0][y0] != MODE_INTRA && !(PartMode == PART_2Nx2N && merge_flag[x0][y0]) (CuPredMode[x0][y0] == MODE_INTRA && intra_bc_flag[x0][y0]))	
[91]	rqt_root_cbf	ac(v)
[92]	if(rqt_root_cbf) {	
[93]	MaxTrafoDepth = (CuPredMode[x0][y0] == MODE_INTRA ? (max_transform_hierarchy_depth_intra + IntraSplitFlag) : max_transform_hierarchy_depth_inter)	
[94]	transform_tree(x0, y0, x0, y0, log2CbSize, 0, 0)	

[95]	}	
[96]	}	
[97]	}	
[98]	}	

[0109]

[0110]

[0111]

[0112]

[0113]

표 2와 관련하여, 다음의 것이 언급된다:

1과 동일한 **intra_bc_merge_flag[x0][y0]**는 현재 코딩 유닛이 병합 모드로 코딩되고, 블록 벡터가 병합 후보로부터 선택된다는 것을 지정한다. 0과 동일한 **intra_bc_merge_flag[x0][y0]**는 코딩 유닛이 병합 모드로 코딩되지 않고, 현재 코딩 유닛의 블록 벡터가 명시적으로 코딩된다는 것을 지정한다. 존재하지 않으면, **intra_bc_merge_flag**의 값은 0과 동일한 것으로 추론된다. 어레이 인덱스 **x0, y0**는 화상의 왼쪽 최상부 휘도 샘플에 대한 고려된 코딩 블록의 왼쪽 최상부 휘도 샘플의 위치(**x0, y0**)를 지정한다.

intra_bc_merge_index[x0][y0]는 현재 코딩 유닛의 블록 벡터가 동일한 병합 후보 중에서 인덱스를 지정한다.

intra_bc_merge_index[x0][y0]는 0 내지 인트라 블록 복사 병합 후보의 최대 수 마이너스 1의 범위(range of 0 to maximal number of intra block copying merge candidates minus 1) 내에 있어야 한다. 어레이 인덱스 x0, y0는 화상의 왼쪽 최상부 휘도 샘플에 대한 고려된 코딩 블록의 왼쪽 최상부 휘도 샘플의 위치(x0, y0)를 지정한다.

- [0114] 1과 동일한 **intra_bc_bv_derivation_flag[x0][y0]**는 도출된 BV 또는 MV가 현재 PU 예측을 위해 사용된다는 것을 지정한다.
- [0115] **추가적 예시적인 실시예.**
- [0116] 예시적인 실시예에서, 도출된 블록 벡터가 생성되고, 도출된 블록 벡터는 인트라 블록 복사 병합 모드에서 병합 후보로서 사용된다. 이러한 일부 방법에서, 도출된 블록 벡터 BVd는 블록 벡터 BV0가 주어지면 $BVd = BV0 + BV1$ 에 따라 결정된다.
- [0117] 다른 예시적인 실시예에서, 도출된 움직임 벡터가 생성되고, 도출된 움직임 벡터는 인트라 블록 복사 병합 모드에서 병합 후보로서 사용된다. 이러한 일부 방법에서, 정수 픽셀 유닛의 도출된 움직임 벡터 MVd는 블록 벡터 BV0가 주어지면 $MVd = BV0 + ((MV1 + 2) \gg 2)$ 에 따라 결정된다.
- [0118] 예시적인 실시예에서, 도출된 블록 벡터가 생성되고, 도출된 블록 벡터는 인트라 블록 복사 방법을 사용하여 정상적인 IntraBC 예측을 위해 사용된다.
- [0119] 다른 예시적인 실시예에서, 도출된 움직임 벡터가 생성되고, 도출된 움직임 벡터는 움직임 보상 예측 방법을 사용하여 정상적인 IntraBC 예측을 위해 사용된다.
- [0120] 예시적인 실시예에서, 블록 벡터(BV) 후보 리스트가 형성된다. 도출된 BV(BVd)가 생성되고, BVd가 후보 리스트에 추가된다.
- [0121] 이러한 일부 실시예에서, BV 후보 리스트의 형성은 BV 예측자가 현재의 코딩 유닛(CU)에 대해 유효하면 BV 예측자를 추가하고, 5개의 공간 블록을 체크하며, 유효한 각각의 공간 블록 BV를 추가하는 것을 포함한다. 이러한 일부 실시예에서, 공간 블록 BV는 (i) 공간적 인접한 블록이 IntraBC 코딩되고, (ii) BV가 현재 CU에 대해 유효하며, (iii) BV가 현재 후보 리스트에 이미 나타나지 않은 경우에만 추가된다. 이러한 일부 실시예에서, BVd는 병합 후보 리스트가 가득 차 있지 않은 경우에만 생성된다. BVd는 후보 리스트에 추가되기 전에 유효성에 대해 체크될 수 있다. 일부 실시예에서, 생성되는 BVd는 $BVd = BV0 + BV1$ 에 따라서만 도출된다. 이러한 일부 방법에서, 도출된 움직임 벡터(MVd)가 생성되고, MVd가 후보 리스트에 추가된다. 일부 실시예에서, BVd는 (i) $BVd = BV0 + BV1$ 및 (ii) $MVd = BV0 + ((MV1 + 2) \gg 2)$ 둘 다에 따라 도출된다. 일부 실시예에서, 후보 블록은 BV0 및 BVd/MVd를 가진 양방향 예측 모드로서 처리된다. 양방향 예측은 먼저 블록 벡터(BV0)를 적용함으로써 획득된 제 1 예측을 블록 또는 움직임 벡터(BVd/MVd)를 적용함으로써 획득된 제 2 예측과 평균함으로써 획득될 수 있다.
- [0122] 예시적인 실시예에서, 비디오 코딩된 비트 스트림이 수신되고, 시그널링된 블록 벡터 또는 도출된 블록 벡터가 IntraBC 예측에 사용되는지를 나타내는 플래그가 비트 스트림에서 식별된다. 플래그가 제 1 값이면, 시그널링된 블록 벡터는 IntraBC 예측에 사용된다. 플래그가 제 2 값이면, BV 또는 MV는 시그널링된 블록 벡터에 기초하여 도출된다. 이러한 일부 방법에서, 도출된 BV 또는 MV는 인트라 블록 복사 예측 또는 움직임 보상 예측에 사용된다.
- [0123] 다른 예시적인 실시예에서, 코딩된 비디오 비트 스트림이 수신된다. 비트 스트림에서의 제 1 플래그는 시그널링된 블록 벡터 또는 도출된 블록 벡터가 IntraBC 예측에 사용되는지를 나타내는 것으로 식별된다. 비트 스트림에서의 제 2 플래그는 단방향 예측 또는 양방향 예측이 사용되는지를 나타내는 것으로 식별된다.
- [0124] 이러한 일부 실시예에서, 제 1 플래그가 1이고, 제 2 플래그가 0이면, 도출된 BV/MV만이 인트라 블록 복사 예측 또는 움직임 보상된 예측을 위해 사용된다. 제 1 플래그가 1이고, 제 2 플래그가 1이면, 시그널링된 BV는 제 1 예측을 생성하는데 사용되고, 도출된 BV/MV는 제 2 예측을 생성하는데 사용되며, 최종 예측은 이러한 2개의 예측을 평균함으로써 생성된다.
- [0125] 예시적인 실시예에서, 원래의 BV/MV 정보는 더 큰 블록 크기에 기초하여 BV/MV 정보를 저장함으로써 압축된다. 더 큰 블록 크기는 예를 들어 16x16 블록 크기일 수 있다.
- [0126] 다른 예시적인 실시예에서, 디코더는 이미 코딩된 모든 블록보다 적은 제한된 범위 내의 코딩된 블록의 정보를

캐시하는데 사용된다. 이러한 일부 방법에서, 디코더는 현재 CTU 행의 정보 및 현재 CTU 행보다 많은 미리 결정된 수의 코딩된 인접한 CTU 행의 정보만을 캐시한다.

- [0127] 예시적인 실시예에서, 예측 벡터를 도출하는 비디오 코딩 방법이 제공된다. 방법은 비디오 블록의 예측을 위한 제 1 후보 블록 벡터를 식별하는 단계를 포함하며, 제 1 후보 블록 벡터는 제 1 후보 블록을 나타낸다. 제 1 후보 블록과 관련된 적어도 제 1 예측 벡터가 식별된다. 도출된 예측 벡터는 제 1 후보 블록 벡터 및 제 1 예측 벡터로부터 생성되고, 비디오 블록은 도출된 예측 벡터를 사용하여 코딩된다. 일부 실시예에서, 도출된 예측 벡터를 사용하는 비디오 블록의 코딩은 도출된 예측 벡터가 나타내는 제 2 후보 블록을 식별하고, 제 2 후보 블록을 사용하여 비디오 블록을 예측하는 것을 포함한다.
- [0128] 일부 실시예에서, 비디오 블록을 코딩하는 단계는 비트 스트림에서 제 1 후보 블록 벡터를 시그널링하는 단계를 포함한다. 일부 실시예에서, 비디오 블록을 코딩하는 단계는 비트 스트림에서 제 1 예측 벡터를 시그널링하는 단계를 더 포함한다. 일부 실시예에서, 비디오 블록을 코딩하는 단계는 비트 스트림에서 플래그를 시그널링하는 단계를 더 포함하며, 플래그의 제 1 값은 도출된 예측 벡터가 비디오 블록을 코딩하는데 사용되는 것을 나타내고, 플래그의 제 2 값은 제 1 후보 블록 벡터가 비디오 블록을 코딩하는데 사용되는 것을 나타낸다. 일부 실시예에서, 비트 스트림에서 입력 비디오 블록을 인코딩하는 단계는 병합 후보 리스트에서 도출된 예측 벡터를 식별하는 인덱스를 비트 스트림에서 인코딩하는 단계를 포함한다.
- [0129] 일부 실시예에서, 도출된 예측 벡터는 제 1 후보 블록 벡터 및 제 1 예측 벡터를 부가함으로써 생성된다. 제 1 후보 블록 벡터 및 제 1 예측 벡터가 상이한 정밀도를 갖는 경우, 제 1 블록 벡터 및 제 1 예측 벡터의 부가는 이러한 두 벡터를 동일한 정밀도로 정렬한 후에 수행될 수 있다.
- [0130] 일부 실시예에서, 비디오 코딩 방법은 병합 후보 리스트를 생성하는 단계와, 도출된 예측 벡터를 병합 후보 리스트에 삽입하는 단계를 더 포함한다. 이러한 일부 실시예에서, 도출된 예측 벡터가 유효한지에 대한 결정이 이루어지고, 도출된 예측 벡터가 유효한 것으로 결정한 후에만 도출된 예측 벡터는 병합 후보 리스트에 삽입된다. 도출된 예측 벡터를 사용하여 비디오 블록을 코딩하는 단계는 병합 후보 리스트에서 도출된 예측 벡터를 식별하는 인덱스를 제공하는 단계를 포함한다. 일부 실시예에서, 도출된 예측 벡터가 유효한지의 결정은 도출된 예측 벡터가 나타내는 제 2 후보 블록을 식별하고, 제 2 후보 블록의 모든 샘플이 이용 가능한지를 결정하는 것을 포함한다. 도출된 예측 벡터는 제 2 후보 블록의 모든 샘플이 이용 가능하다면 유효한 것으로 결정된다. 도출된 예측 벡터는 제 2 후보 블록의 적어도 하나의 샘플이 이용 가능하지 않다면 유효하지 않은 것으로 결정된다. 일부 실시예에서, 제 2 후보 블록의 샘플은 다음 것: 샘플이 아직 코딩되지 않았거나, 샘플이 상이한 슬라이스 또는 상이한 타일(tile)에 있거나, 샘플이 비디오 화상 경계에서 벗어난 것 중 하나가 참이라면 이용 가능하지 않다.
- [0131] 예시적인 실시예에서, 비디오를 인코딩하는 비트 스트림으로부터 코딩된 비디오 블록을 디코딩하는 방법이 제공된다. 적어도 제 1 후보 블록 벡터는 입력 비디오 블록의 예측을 위해 식별되고, 제 1 후보 블록 벡터는 제 1 후보 블록을 나타낸다. 제 1 후보 블록과 관련된 적어도 제 1 예측 벡터가 식별된다. 도출된 예측 벡터는 제 1 블록 벡터 및 제 1 예측 벡터로부터 생성되고, 코딩된 비디오 블록은 도출된 예측 벡터를 사용하여 디코딩된다. 이러한 일부 실시예에서, 제 1 후보 블록 벡터의 식별은 비트 스트림에서 시그널링된 제 1 후보 블록 벡터를 수신하는 것을 포함한다.
- [0132] 이러한 일부 실시예에서, 도출된 예측 벡터를 사용하는 코딩된 비디오 블록의 디코딩은 입력 비디오 블록이 도출된 예측 벡터로 인코딩되는 것을 나타내는 플래그를 비트 스트림에서 수신하는 것에 응답하여 수행된다.
- [0133] 일부 실시예에서, 제 1 후보 블록 벡터의 식별은 제 1 블록 벡터 병합 후보의 식별을 포함한다. 일부 실시예에서, 도출된 예측 벡터는 도출된 예측 벡터 병합 후보이고, 도출된 예측 벡터 병합 후보를 사용하여 코딩된 비디오 블록의 디코딩은 도출된 예측 벡터 병합 후보를 식별하는 인덱스를 비트 스트림에서 수신하는 것에 응답하여 수행된다.
- [0134] 예시적인 실시예에서, 입력 비디오 블록을 포함하는 비디오를 인코딩하는 비트 스트림을 생성하기 위한 비디오 인코딩 방법이 제공된다. 입력 비디오 블록의 인접한 블록이 식별된다. 인접한 블록은 예를 들어 입력 비디오 블록의 왼쪽, 최상부 또는 왼쪽 최상부 인접 블록일 수 있다. 인접한 블록과 연관된 제 1 블록 벡터가 식별되며, 여기서 제 1 블록 벡터는 제 1 후보 블록을 나타낸다. 제 1 후보 블록과 연관된 제 2 블록 벡터가 식별된다. 도출된 블록 벡터는 제 1 블록 벡터 및 제 2 블록 벡터를 부가함으로써 생성되고, 제 1 예측 블록은 도출된 블록 벡터를 사용하여 입력 비디오 블록의 예측을 위해 생성된다.

- [0135] 이러한 일부 실시예에서, 입력 비디오 블록의 인코딩은 제 3 블록 벡터를 사용하여 입력 비디오 블록의 예측을 위한 적어도 제 2 예측 블록을 생성하는 것을 더 포함한다. 제 1 예측 블록 및 제 2 예측 블록은 비교되고, 예측 블록 및 연관된 블록 벡터는 인코딩 메트릭에 기초하여 선택된다. 인코딩 메트릭은 예를 들어 라그랑지안 레이트 왜곡 비용(Lagrangian rate-distortion cost)일 수 있다.
- [0136] 예시적인 실시예에서, 입력 비디오 블록을 포함하는 비디오를 인코딩하도록 비트 스트림을 생성하기 위한 비디오 인코더가 제공된다. 인코더는 프로세서와, 프로세서상에서 실행될 때, 다음의 기능을 수행하도록 동작 가능한 명령어를 저장한 비일시적 저장 매체를 포함하고, 이러한 기능은, 입력 비디오 블록의 예측을 위한 적어도 제 1 후보 블록 벡터 - 상기 제 1 후보 블록 벡터는 제 1 후보 블록을 나타냄 - 를 식별하고; 제 1 후보 블록을 인코딩하는데 사용된 제 1 예측 벡터를 식별하고; 제 1 후보 블록 벡터 및 제 1 예측 벡터로부터 도출된 예측 벡터를 생성하며; 입력 비디오 블록의 예측을 위해 도출된 예측 벡터를 사용하여 비트 스트림에서 입력 비디오 블록을 인코딩하는 것을 포함한다.
- [0137] 예시적인 실시예에서, 입력 비디오 블록을 포함하는 비디오를 인코딩하는 비트 스트림을 생성하기 위한 비디오 인코더가 제공된다. 인코더는 프로세서와, 프로세서상에서 실행될 때, 다음의 기능을 수행하도록 동작 가능한 명령어를 저장한 비일시적 저장 매체를 포함하고, 이러한 기능은, 입력 비디오 블록의 인코딩을 위한 적어도 제 1 블록 벡터 병합 후보를 식별하고; 제 1 후보 블록을 인코딩하는데 사용된 제 1 예측 벡터를 식별하고; 제 1 블록 벡터 병합 후보 및 제 1 예측 벡터로부터 도출된 예측 벡터를 생성하고; 도출된 예측 벡터를 병합 후보 리스트에 삽입하고; 병합 후보 리스트로부터 입력 비디오 블록의 예측을 위해 선택된 예측 벡터를 선택하며; 입력 비디오 블록의 예측을 위해 선택된 예측 벡터를 사용하여 비트 스트림에서 입력 비디오 블록을 인코딩하는 것을 포함한다.
- [0138] 예시적인 실시예에서, 비디오를 인코딩하는 비트 스트림으로부터 코딩된 비디오 블록을 디코딩하기 위한 비디오 디코더가 제공된다. 디코더는 프로세서와, 프로세서상에서 실행될 때, 다음의 기능을 수행하도록 동작 가능한 명령어를 저장한 비일시적 저장 매체를 포함하고, 이러한 기능은, 코딩된 비디오 블록의 디코딩을 위해 적어도 제 1 블록 벡터를 식별하고; 제 1 블록 벡터를 인코딩하는데 사용된 제 1 예측 벡터를 식별하고; 제 1 블록 벡터 및 제 1 예측 벡터로부터 도출된 예측 벡터를 생성하며; 코딩된 비디오 블록의 예측을 위해 도출된 예측 벡터를 사용하여 코딩된 비디오 블록을 디코딩하는 것을 포함한다.
- [0139] 특징 및 요소가 특정 조합으로 상술되었지만, 당업자는 각각의 특징 또는 요소가 단독으로 또는 다른 특징 및 요소와 임의의 조합으로 사용될 수 있다는 것을 이해할 것이다. 게다가, 본 명세서에 설명된 방법은 컴퓨터 또는 프로세서에 의한 실행을 위해 컴퓨터 판독 가능 매체에 통합된 컴퓨터 프로그램, 소프트웨어 또는 펌웨어로 구현될 수 있다. 컴퓨터 판독 가능 매체의 예는 (유선 또는 무선 연결을 통해 전송되는) 전자 신호 및 컴퓨터 판독 가능 저장 매체를 포함한다. 컴퓨터 판독 가능 저장 매체의 예는 판독 전용 메모리(ROM), 랜덤 액세스 메모리(RAM), 레지스터, 캐시 메모리, 반도체 메모리 디바이스, 내부 하드 디스크 및 이동식 디스크와 같은 자기 매체, 광 자기 매체 및 CD-ROM 디스크 및 DVD(digital versatile disk)와 같은 광학 매체를 포함하지만, 이에 한정되지 않는다. 소프트웨어와 연관된 프로세서는 WTRU, UE, 단말기, 기지국, RNC 또는 임의의 호스트 컴퓨터에 사용하기 위한 무선 주파수 송수신기를 구현하는데 사용될 수 있다.

[0140] 참고 문헌.

- [R1] ITU-T Q6/16 and ISO/IEC JCT1/SC29/WG11, “Joint Call for Proposals for Coding of Screen Content”, MPEG2014/N14175, Jan. 2014, San Jose, USA.
- [R2] J. Chen, Y. Chen, T. Hsieh, R. Joshi, M. Karczewicz, W.-S. Kim, X. Li, C. Pang, W. Pu, K. Rapaka, J. Sole, L. Zhang, F. Zou, “Description of screen content coding technology proposal by Qualcomm”, JCTVC-Q0031, Mar. 2014, Valencia, ES.
- [R3] C.-C. Chen, T.-S. Chang, R.-L. Liao, C.-W. Kuo, W.-H. Peng, H.-M. Hang, Y.-J. Chang, C.-H. Hung, C.-C. Lin, J.-S. Tu, E.-C. Ke, J.-Y. Kao, C.-L. Lin, F.-D. Jou, F.-C. Chen, “Description of screen content coding technology proposal by NCTU and ITRI International”, JCTVC-Q0032, Mar. 2014, Valencia, ES.
- [R4] P. Lai, T.-D. Chuang, Y.-C. Sun, X. Xu, J. Ye, S.-T. Hsiang, Y.-W. Chen, K. Zhang, X. Zhang, S. Liu, Y.-W. Huang, S. Lei, “Description of screen content coding technology proposal by MediaTek”, JCTVC-Q0033, Mar. 2014, Valencia, ES.

[0141]

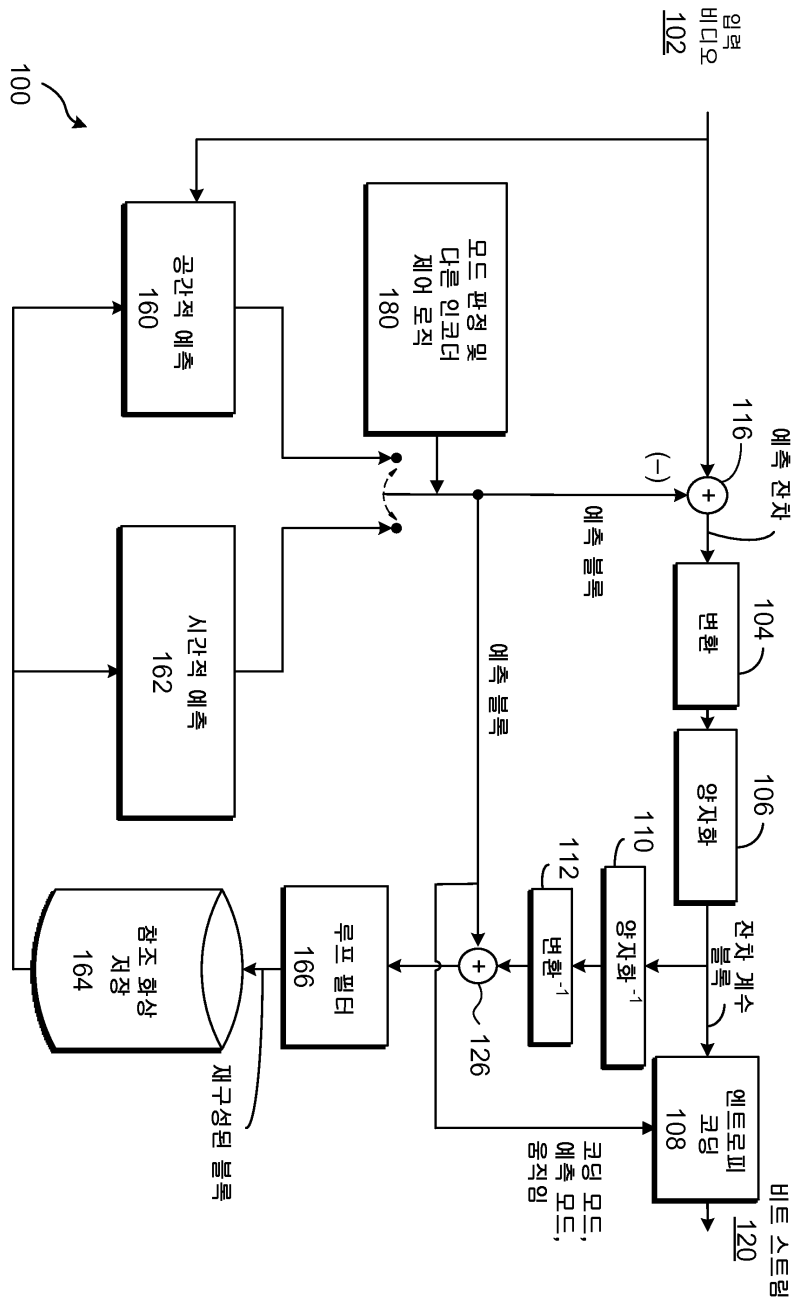
- [R5] Z. Ma, W. Wang, M. Xu, X. Wang, H. Yu, "Description of screen content coding technology proposal by Huawei Technologies", JCTVC-Q0034, Mar. 2014, Valencia, ES.
- [R6] B. Li, J. Xu, F. Wu, X. Guo, G. J. Sullivan, "Description of screen content coding technology proposal by Microsoft", JCTVC-Q0035, Mar. 2014, Valencia, ES.
- [R7] R. Cohen, A. Minezawa, X. Zhang, K. Miyazawa, A. Vetro, S. Sekiguchi, K. Sugimoto, T. Murakami, "Description of screen content coding technology proposal by Mitsubishi Electric Corporation", JCTVC-Q0036, Mar. 2014, Valencia, ES.
- [R8] X. Xiu, C.-M. Tsai, Y. He, Y. Ye, "Description of screen content coding technology proposal by InterDigital", JCTVC-Q0037, Mar. 2014, Valencia, ES.
- [R9] T. Lin, S. Wang, P. Zhang, and K. Zhou, "AHG8: P2M based dual-coder extension of HEVC", Document no JCTVC-L0303, Jan. 2013.
- [R10] X. Guo, B. Li, J.-Z. Xu, Y. Lu, S. Li, and F. Wu, "AHG8: Major-color-based screen content coding", Document no JCTVC-O0182, Oct. 2013.
- [R11] L. Guo, M. Karczewicz, J. Sole, and R. Joshi, "Evaluation of Palette Mode Coding on HM-12.0+RExt-4.1", JCTVC-O0218, Oct. 2013.
- [R12] C. Pang, J. Sole, L. Guo, M. Karczewicz, and R. Joshi, "Non-RCE3: Intra Motion Compensation with 2-D MVs", JCTVC-N0256, July 2013.
- [R13] B. Bross, W.-J. Han, G. J. Sullivan, J.-R. Ohm, T. Wiegand, "High Efficiency Video Coding (HEVC) Text Specification Draft 10", JCTVC-L1003. Jan 2013.
- [R14] G.J. Sullivan and T. Wiegand, Rate-distortion optimization for video compression. IEEE Signal Processing Magazine, vol. 15, issue 6, November 1998.
- [R15] T. Vermeir, "Use cases and requirements for lossless and screen content coding", JCTVC-M0172, Apr. 2013, Incheon, KR.
- [R16] J. Sole, R. Joshi, M. Karczewicz, "AhG8: Requirements for wireless display applications", JCTVC-M0315, Apr. 2013, Incheon, KR.
- [R17] D. Flynn, M. Naccari, K. Sharman, C. Rosewarne, J. Sole, G. J. Sullivan, T. Suzuki, "HEVC Range Extension Draft 6", JCTVC-P1005, Jan. 2014, San Jose.
- [R18] J. Sole, S. Liu, "HEVC Screen Content Coding Core Experiment 1 (SCCE1): Intra Block Copying Extensions", JCTVC-Q1121, Mar. 2014, Valencia.

[0142]

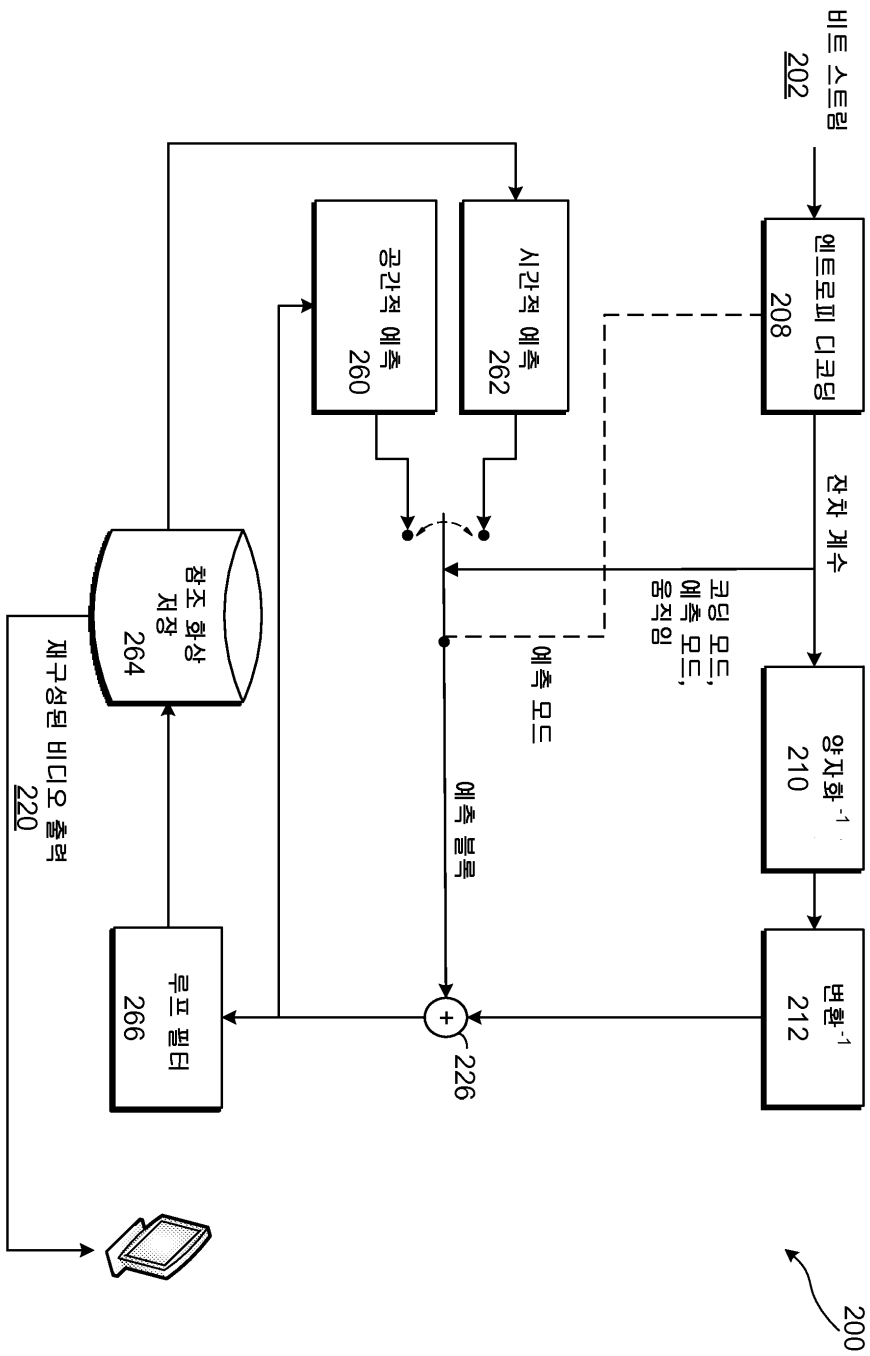
- [R19] C.-C. Chen, X. Xu, L. Zhang, “HEVC Screen Content Coding Core Experiment 2 (SCCE2): Line-based Intra Copy”, JCTVC-Q1122, Mar. 2014, Valencia.
- [R20] Y.-W. Huang, P. Onno, R. Joshi, R. Cohen, X. Xiu, Z. Ma, “HEVC Screen Content Coding Core Experiment 3 (SCCE3): Palette mode”, JCTVC-Q1123, Mar. 2014, Valencia.
- [R21] Y. Chen, J. Xu, “HEVC Screen Content Coding Core Experiment 4 (SCCE4): String matching for sample coding”, JCTVC-Q1124, Mar. 2014, Valencia.
- [R22] X. Xiu, J. Chen, “HEVC Screen Content Coding Core Experiment 5 (SCCE5): Inter-component prediction and adaptive color transforms”, JCTVC-Q1125, Mar. 2014, Valencia.
- [R23] P. Onno, G. Laroche, T. Poirier, C. Gisquet, “AhG5: On the displacement vector prediction scheme for Intra Block Copy”, JCTVC-Q0062, Mar. 2014, Valencia.
- [R24] X. Zhang, K. Zhang, J. An, H. Huang, S. Lei, “Block vector prediction for intra block copy”, JCTVC-Q0080, Mar. 2014, Valencia.
- [R25] K. Zhang, J. An, X. Zhang, H. Huang, S. Lei, “Symmetric intra block copy”, JCTVC-Q0082, Mar. 2014, Valencia.
- [R26] S.-T. Hsiang, T.-D. Chuang, S. Lei, “AHG8: Coding the prediction differences of the intra BC vectors”, JCTVC-Q0095, Mar. 2014, Valencia.
- [R27] C. Pang, J. Sole, R. Joshi, M. Karczewicz, “Block vector prediction method for intra block copy”, JCTVC-Q0114, Mar. 2014, Valencia.
- [R28] L. Zhu, J. Xu, G. J. Sullivan, Y. Wu, S. Sankuratri, B.A.Kumar, “Ping-pong block vector predictor for intra block copy”, JCTVC-Q0134, Mar. 2014, Valencia.
- [R29] B. Li, J. Xu, “Hash-based intraBC search”, JCTVC-Q0252, Mar. 2014, Valencia.
- [R30] C. Pang, J. Sole, T. Hsieh, M. Karczewicz, “Intra block copy with larger search region”, JCTVC-Q0139, Mar. 2014, Valencia.
- [R31] R. Joshi, J. Xu, R. Cohen, S. Liu, Z. Ma, Y. Ye, “Screen content coding test model 1 (SCM 1)”, JCTVC-Q1014, Mar. 2014, Valencia.

[0143]

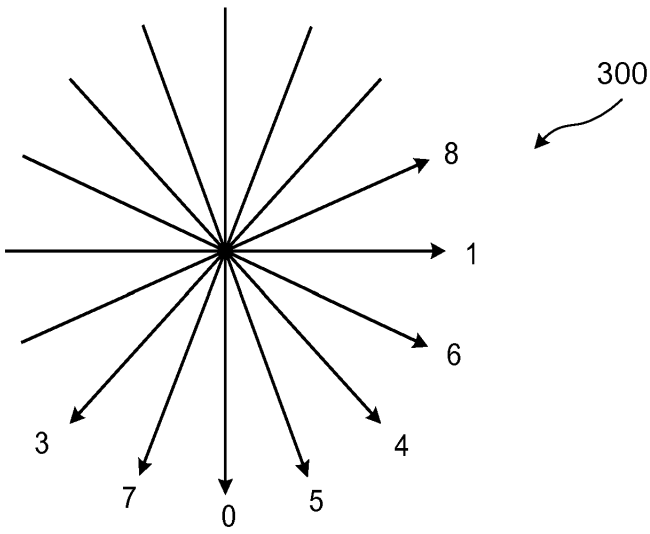
도면
도면1



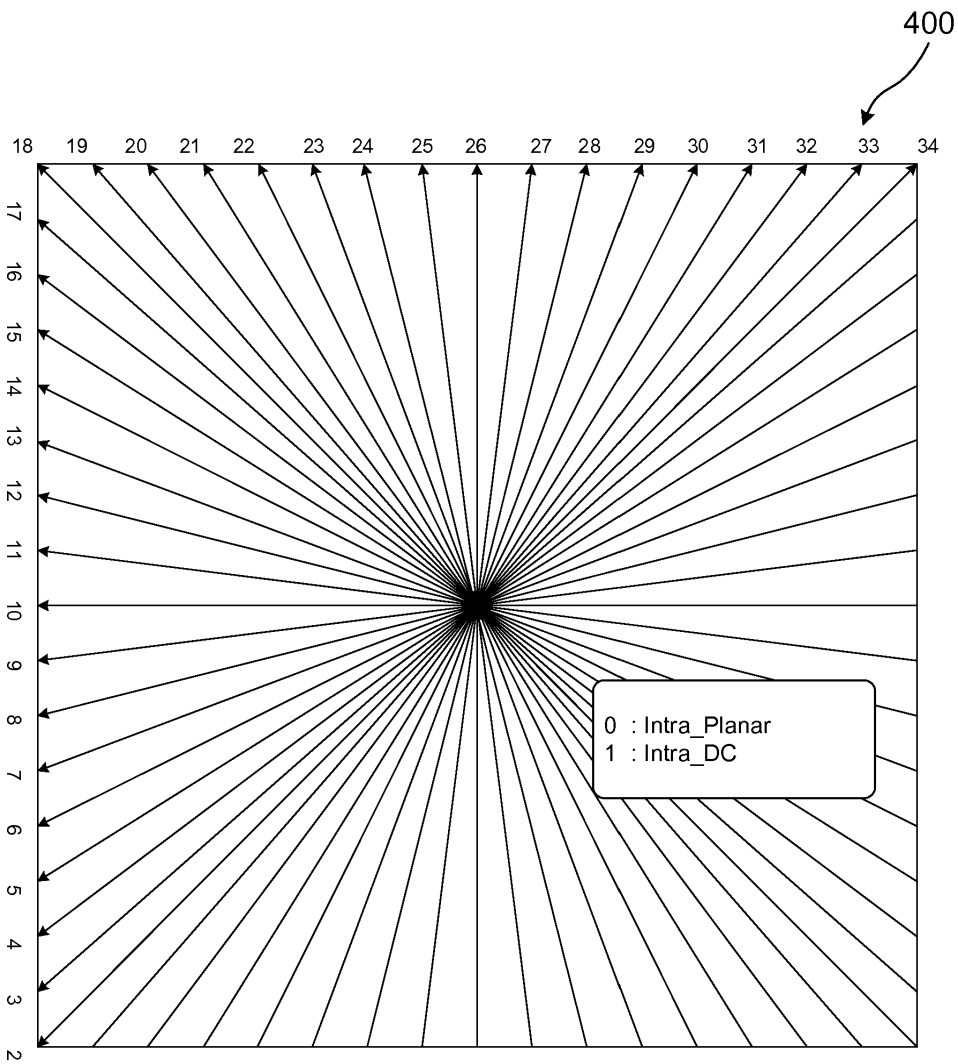
도면2



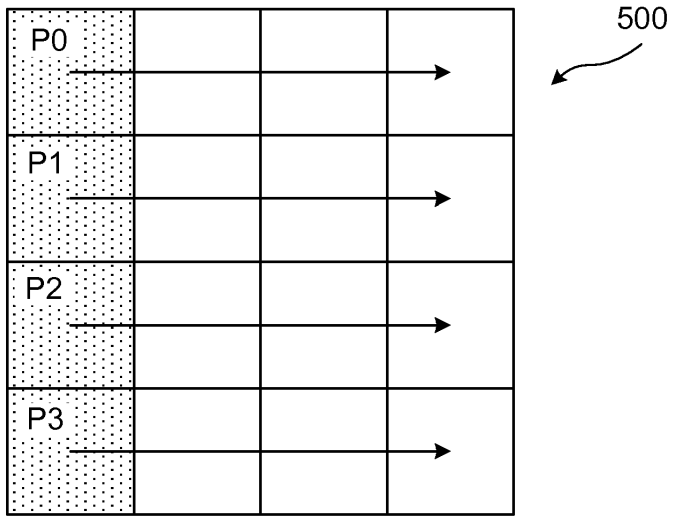
도면3



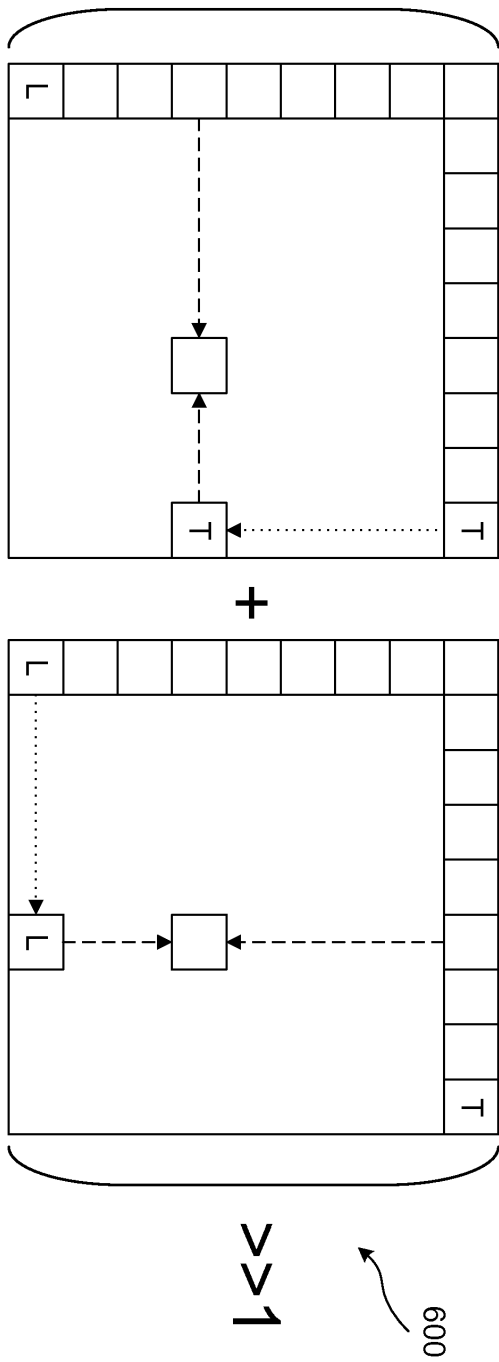
도면4



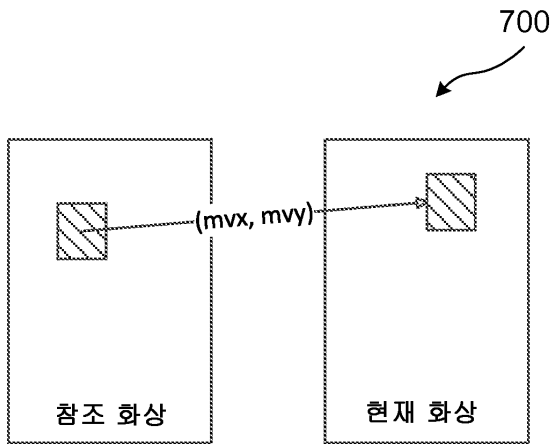
도면5



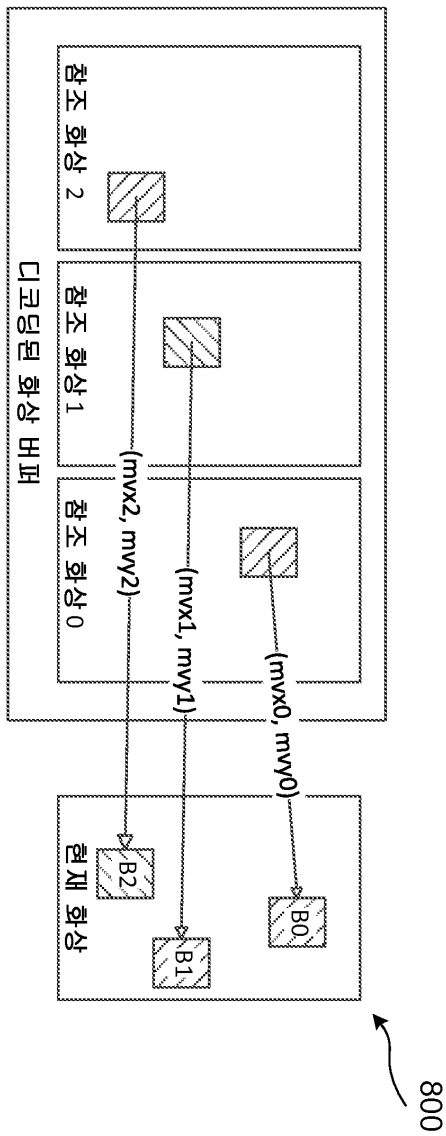
도면6



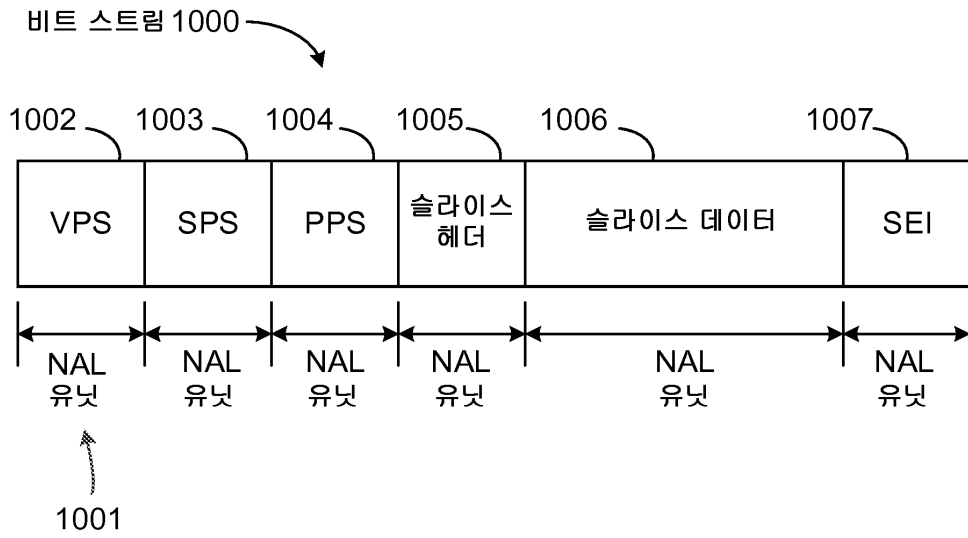
도면7



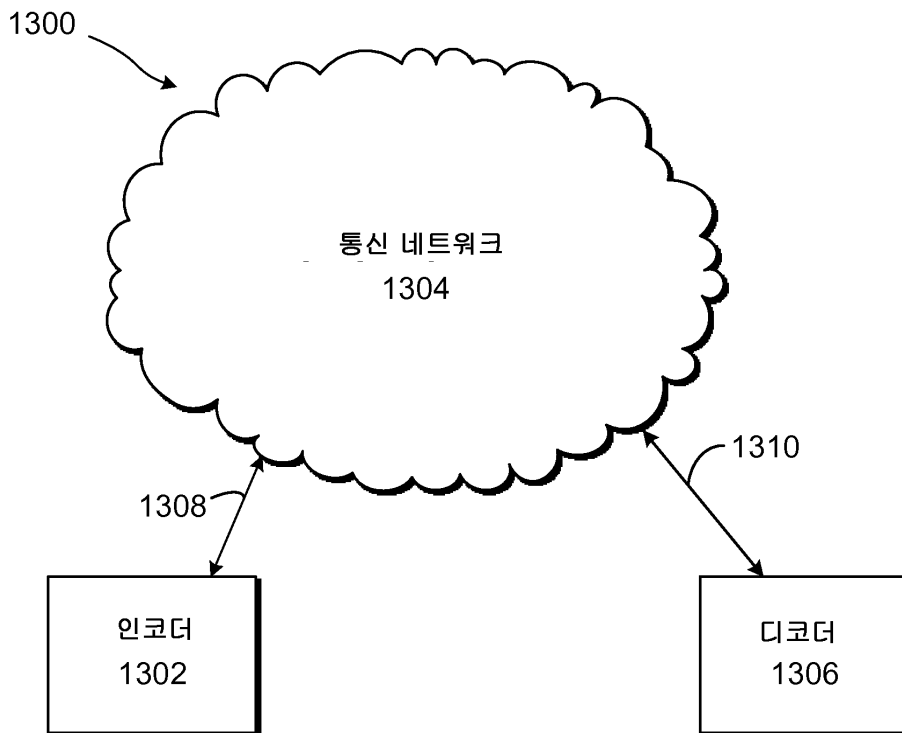
도면8



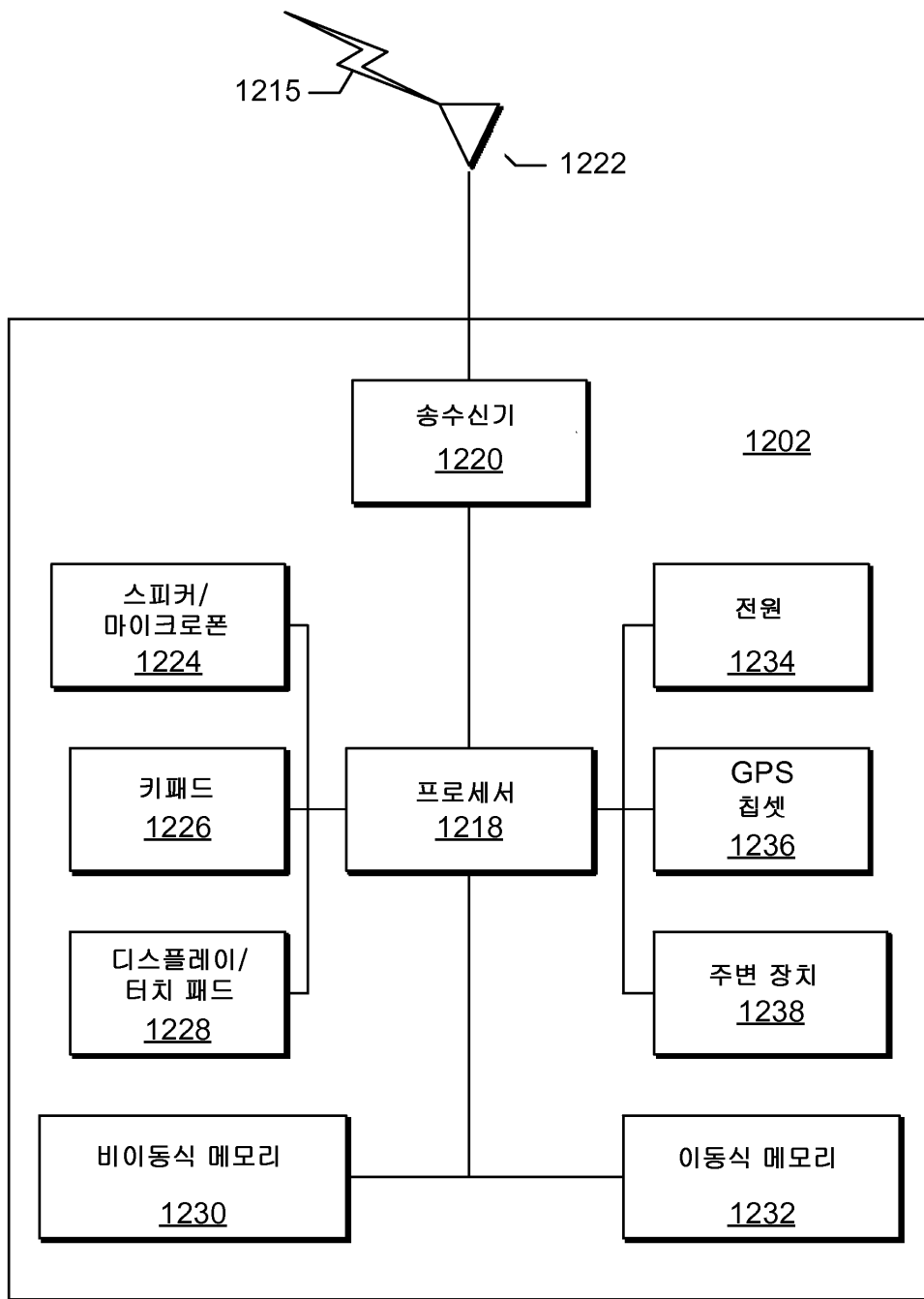
도면9



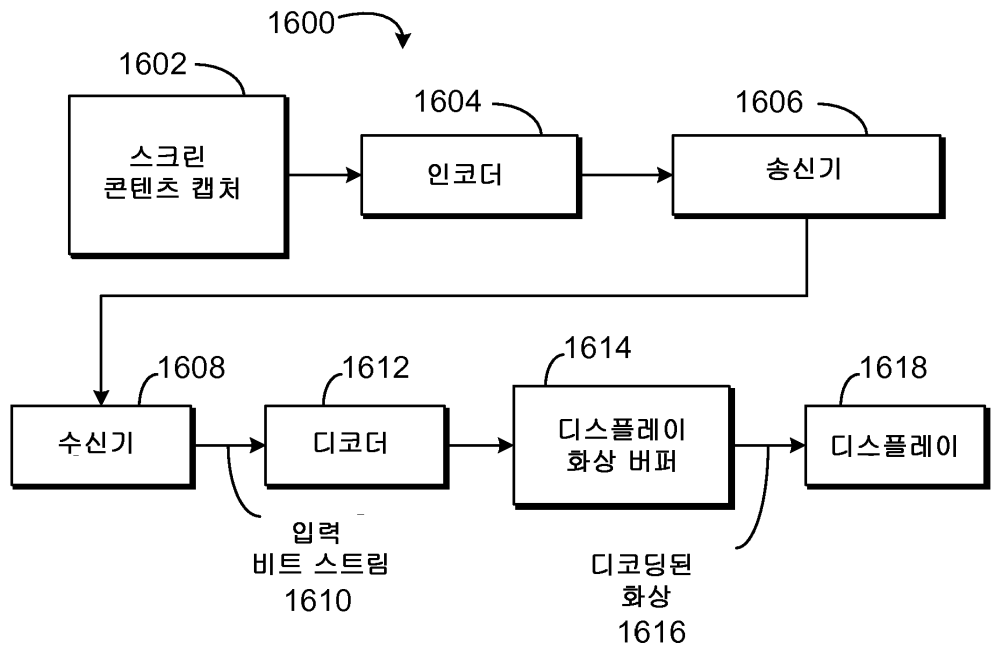
도면10



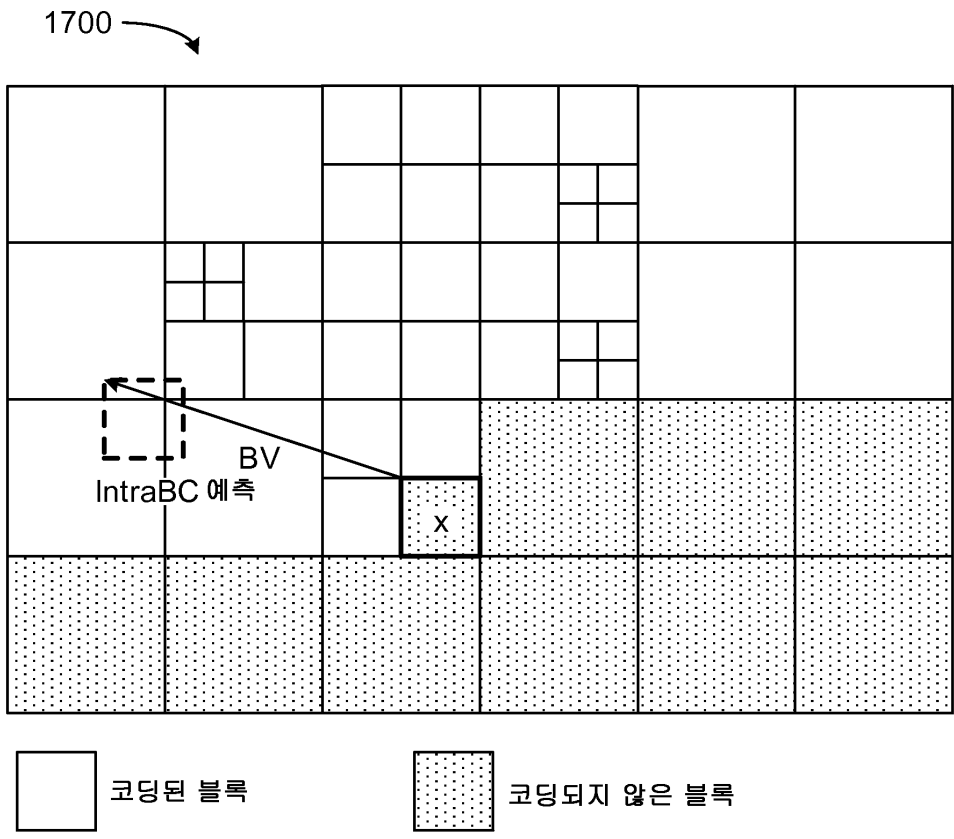
도면11



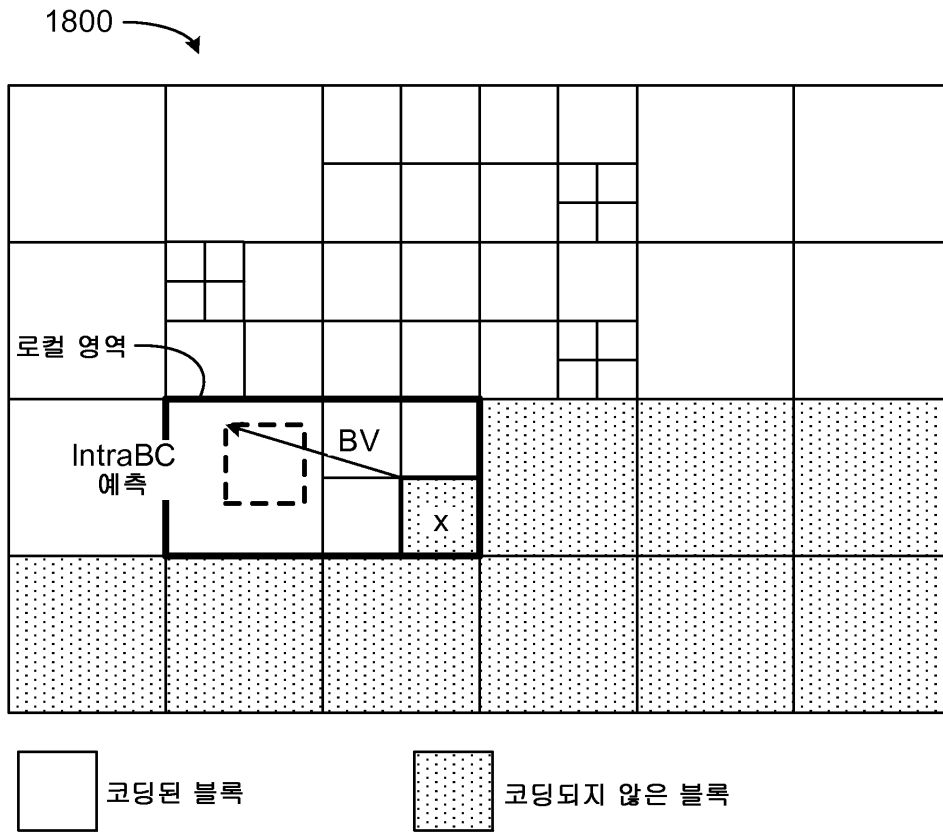
도면12



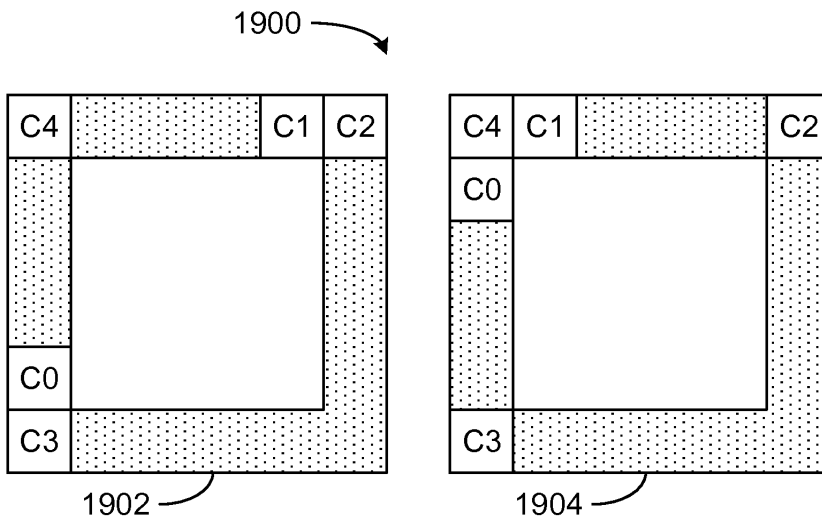
도면13



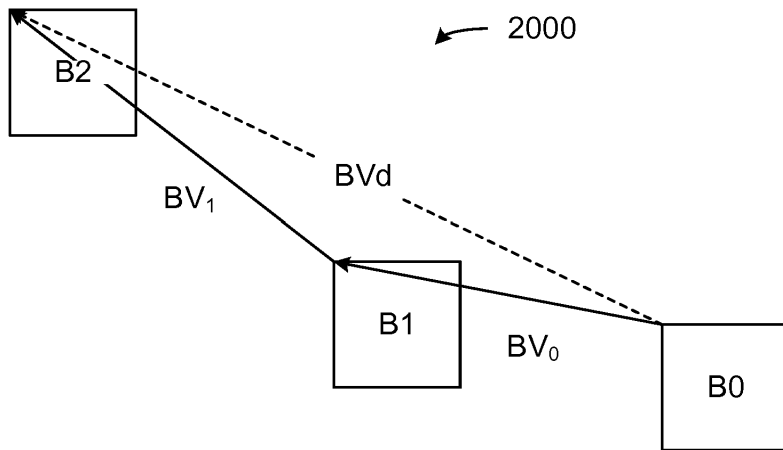
도면14



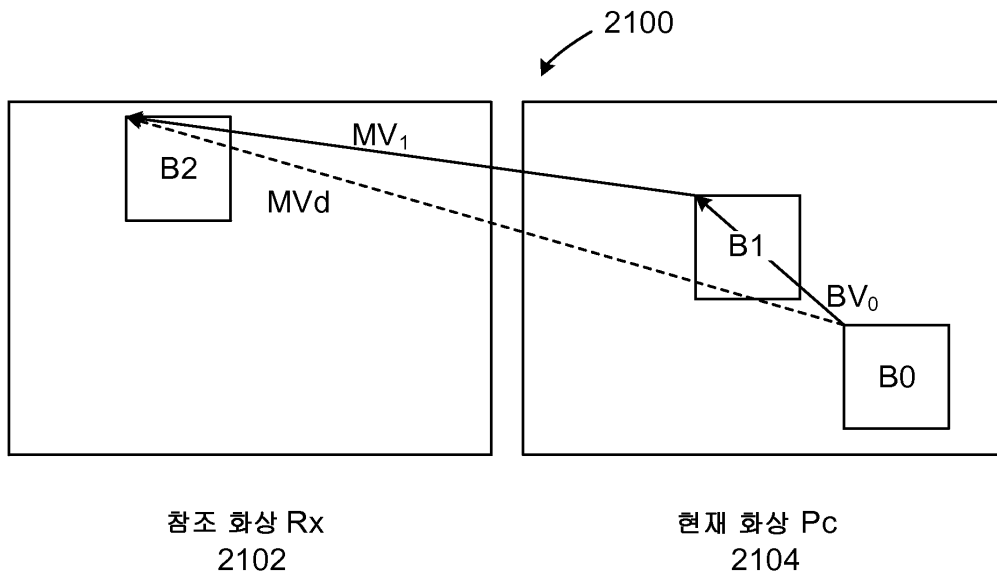
도면15



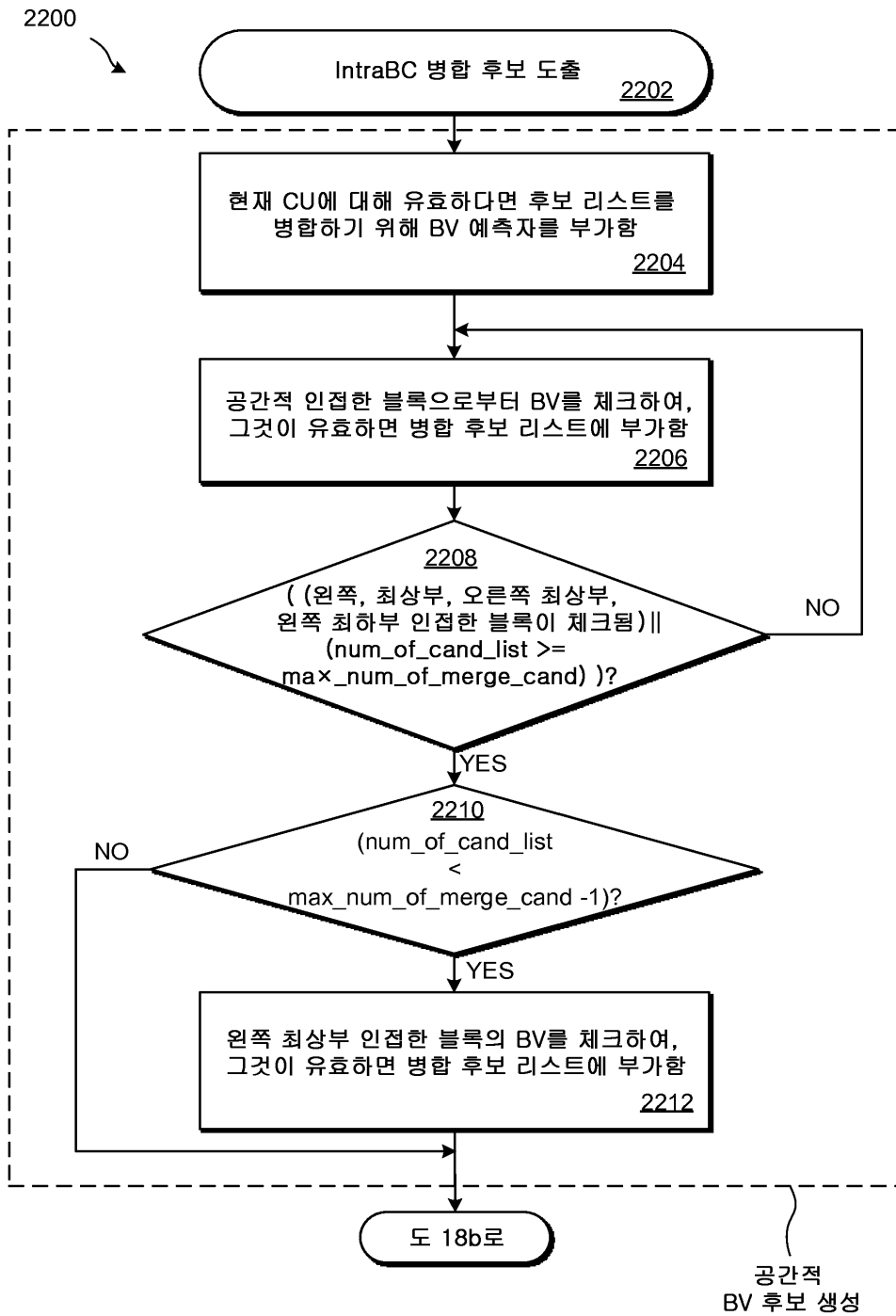
도면16



도면17



도면18a



도면18b

