



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2020-0057082  
(43) 공개일자 2020년05월25일

- (51) 국제특허분류(Int. Cl.)  
 HO4N 19/11 (2014.01) HO4N 19/105 (2014.01)  
 HO4N 19/119 (2014.01) HO4N 19/157 (2014.01)  
 HO4N 19/176 (2014.01) HO4N 19/182 (2014.01)  
 HO4N 19/593 (2014.01) HO4N 19/91 (2014.01)  
 HO4N 19/96 (2014.01)
- (52) CPC특허분류  
 HO4N 19/11 (2015.01)  
 HO4N 19/105 (2015.01)
- (21) 출원번호 10-2020-7012624
- (22) 출원일자(국제) 2018년10월09일  
 심사청구일자 없음
- (85) 번역문제출일자 2020년04월29일
- (86) 국제출원번호 PCT/US2018/055099
- (87) 국제공개번호 WO 2019/074985  
 국제공개일자 2019년04월18일
- (30) 우선권주장  
 62/569,868 2017년10월09일 미국(US)  
 16/155,858 2018년10월09일 미국(US)
- (71) 출원인  
 에리스 엔터프라이즈 엘엘씨  
 미국 조지아 30024, 스와니,레이크펠드 드라이브 3871
- (72) 발명자  
 파누소포네, 크리트  
 미국 92130 캘리포니아주 샌디에고 칼레 마 드 마리포사 10725  
 유, 유에  
 미국 92127 캘리포니아주 샌디에고 케이엔 크릭 로드 16150  
 (뒷면에 계속)
- (74) 대리인  
 양영준, 김연송, 백만기

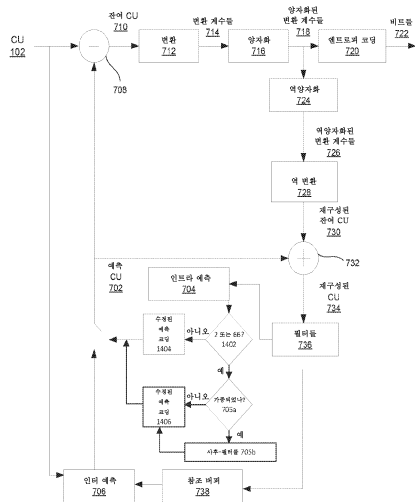
전체 청구항 수 : 총 16 항

(54) 발명의 명칭 적응적 동일하지 않은 가중 평면 예측

(57) 요약

JVET를 위한 비디오 코딩 블록을 파티셔닝하는 방법으로서, JVET 코딩 트리 유닛을, 루트 노드로부터 분기하는 쿼드트리 및 쿼드트리 리프 노드에 의해 표현되는 코딩 유닛을 동일하지 않은 크기의 2개의 자식 노드들로 분열시키기 위해 비대칭 바이너리 파티셔닝을 사용하여 쿼드트리의 리프 노드들 각각으로부터 분기하는 바이너리 트리들을 가질 수 있는 QTBT(quadtrees plus binary tree) 구조에서의 루트 노드로서 표현하는 단계, 이러한 2개의 자식 노드들을 쿼드트리 리프 노드로부터 분기하는 바이너리 트리에서의 리프 노드들로서 표현하는 단계 및 바이너리 트리의 리프 노드들에 의해 표현되는 자식 노드들을 JVET로 코딩하는 단계를 포함하고, 코딩 모드들 2 및 66의 유사성을 이용하는 것에 의해 코딩 효율이 개선된다.

대표도 - 도14



(52) CPC특허분류

*HO4N 19/119* (2015.01)

*HO4N 19/157* (2015.01)

*HO4N 19/176* (2015.01)

*HO4N 19/182* (2015.01)

*HO4N 19/593* (2015.01)

*HO4N 19/91* (2015.01)

*HO4N 19/96* (2015.01)

(72) 발명자

**홍, 승욱**

미국 92127 캘리포니아주 샌디에고 론 도브 스트리트 10165

**왕, 리민**

미국 92130 캘리포니아주 샌디에고 사케브러쉬 벤드 웨이 6338

## 명세서

### 청구범위

#### 청구항 1

JVET 비디오를 코딩하는 방법으로서,

CU x 및 CU y 좌표들을 갖는 CU(coding unit)를 비디오 프레임의 코딩 영역 내에서 정의하는 단계;

상기 메인 참조와 연관된 메인 x 및 메인 y 좌표들을 갖는 메인 참조 픽셀을 상기 코딩 영역 내에서 정의하는 단계;

상기 사이드 참조와 연관된 사이드 x 및 사이드 y 좌표들을 갖는 사이드 참조 픽셀을 상기 코딩 영역 내에서 정의하는 단계;

예측 모드들의 세트를 정의하는 단계;

상기 예측 모드들의 세트 내에서 2개의 이산 예측 모드들을 식별하는 단계;

상기 예측 모드들의 세트로부터 예측 모드를 선택하는 단계; 및

상기 메인 참조 픽셀과 상기 사이드 참조 픽셀의 조합에 적어도 부분적으로 기초하여 상기 코딩 유닛에 대한 예측 CU를 생성하는 단계를 포함하고;

상기 코딩 유닛에 대한 상기 예측 CU는 상기 2개의 이산 예측 모드들 각각에 대해 동일한 방식으로 코딩되고; 및

상기 2개의 이산 예측 모드들 각각은 예측 방향에 적어도 부분적으로 기초하여 구별되는 JVET 비디오를 코딩하는 방법.

#### 청구항 2

제1항에 있어서, 상기 예측 방향은 상기 코딩 유닛의 하나 이상의 특성에 기초하는 JVET 비디오를 코딩하는 방법.

#### 청구항 3

제2항에 있어서, 상기 예측 CU는 엔트로피 코딩되는 JVET 비디오를 코딩하는 방법.

#### 청구항 4

제2항에 있어서, 상기 예측 방향은 상기 코딩 유닛의 폭에 적어도 부분적으로 기초하는 JVET 비디오를 코딩하는 방법.

#### 청구항 5

제4항에 있어서, 상기 예측 방향은 상기 코딩 유닛의 높이에 적어도 부분적으로 기초하는 JVET 비디오를 코딩하는 방법.

#### 청구항 6

제2항에 있어서, 상기 예측 방향은 상기 코딩 유닛의 높이에 적어도 부분적으로 기초하는 JVET 비디오를 코딩하는 방법.

#### 청구항 7

제6항에 있어서, 상기 예측 방향은 상기 코딩 유닛의 폭에 적어도 부분적으로 기초하는 JVET 비디오를 코딩하는 방법.

**청구항 8**

제1항에 있어서, 상기 예측 모드들은 0과 66 사이의 정수 값들의 모드들을 포함하는 JVET 비디오를 코딩하는 방법.

**청구항 9**

제1항에 있어서, 상기 2개의 이산 예측 모드들은 모드 2 및 모드 66인 JVET 비디오를 코딩하는 방법.

**청구항 10**

제9항에 있어서, 예측 모드 2와 연관된 코딩은,

상기 메인 참조 픽셀과 연관된 메인 가중 값을 결정하는 단계;

상기 사이드 참조 픽셀과 연관된 사이드 가중 값을 결정하는 단계; 및

상기 메인 가중 값과 조합되는 상기 메인 참조 픽셀 및 상기 사이드 가중 값과 조합되는 상기 사이드 참조 픽셀의 조합에 적어도 부분적으로 기초하여 상기 코딩 유닛에 대한 예측 CU를 생성하는 단계를 포함하는 JVET 비디오를 코딩하는 방법.

**청구항 11**

제10항에 있어서, 상기 메인 가중 값은 상기 코딩 영역과 상기 메인 참조 픽셀 사이의 거리에 적어도 부분적으로 기초하고, 상기 사이드 가중 값은 상기 코딩 영역과 상기 사이드 참조 픽셀 사이의 거리에 적어도 부분적으로 기초하는 JVET 비디오를 코딩하는 방법.

**청구항 12**

제11항에 있어서, 상기 메인 참조 픽셀은 상기 코딩 영역 위에 위치되는 JVET 비디오를 코딩하는 방법.

**청구항 13**

제12항에 있어서, 상기 예측 CU는 엔트로피 코딩되는 JVET 비디오를 코딩하는 방법.

**청구항 14**

제9항에 있어서, 예측 모드 66과 연관된 코딩은,

상기 메인 참조 픽셀과 연관된 메인 가중 값을 결정하는 단계;

상기 사이드 참조 픽셀과 연관된 사이드 가중 값을 결정하는 단계; 및

상기 메인 가중 값과 조합되는 상기 메인 참조 픽셀 및 상기 사이드 가중 값과 조합되는 상기 사이드 참조 픽셀의 조합에 적어도 부분적으로 기초하여 상기 코딩 유닛에 대한 예측 CU를 생성하는 단계를 포함하는 JVET 비디오를 코딩하는 방법.

**청구항 15**

제14항에 있어서, 상기 메인 가중 값은 상기 코딩 영역과 상기 메인 참조 픽셀 사이의 거리에 적어도 부분적으로 기초하고, 상기 사이드 가중 값은 상기 코딩 영역과 상기 사이드 참조 픽셀 사이의 거리에 적어도 부분적으로 기초하는 JVET 비디오를 코딩하는 방법.

**청구항 16**

제15항에 있어서, 상기 메인 참조 픽셀은 상기 코딩 영역의 좌측에 위치되는 JVET 비디오를 코딩하는 방법.

**발명의 설명**

**기술 분야**

<우선권의 청구>

[0001]

[0002] 본 출원은 2017년 10월 9일 출원된 앞서 출원된 미국 임시 출원 제62/569,868호로부터 35 U.S.C § 119(e) 하의 우선권을 청구하며, 그 전체 내용은 본 명세서에 참조로 인용된다.

[0003] <기술 분야>

[0004] 본 개시내용은 비디오 코딩의 분야, 특히 인코딩을 위한 모드들의 수를 감소시키는 것에 의해 더 높은 비트-레이트들, 해상도들 및 더 양호한 품질 비디오를 가능하게 하는 증가된 코딩 효율에 관련된다.

**배경 기술**

[0005] 진화하는 비디오 코딩 표준들에서의 기술적 개선들은 더 높은 비트-레이트들, 더 높은 해상도들, 및 더 양호한 비디오 품질을 가능하게 하기 위해 코딩 효율을 증가시키는 추세를 예시한다. Joint Video Exploration Team은 JVET라고 지칭되는 새로운 비디오 코딩 스킴을 개발하고 있다. HEVC(High Efficiency Video Coding)와 같은 다른 비디오 코딩 스킴들과 유사하게, JVET는 블록-기반 하이브리드 공간 및 시간 예측 코딩 스킴이다. 그러나, HEVC에 비해, JVET는 디코딩된 화면들의 생성을 위한 비트스트림 구조, 구문, 제약들, 및 매핑에 대한 많은 수정들을 포함한다. JVET는 가중 각도 예측을 포함하는 다양한 코딩 기술들을 이용하는 JEM(Joint Exploration Model) 인코더들 및 디코더들에서 구현되었다.

[0006] 현재의 JVET 설계에서는, 예측 CU를 결정하는데 67개의 각도 코딩 모드들이 사용된다. 그러나, 이러한 코딩 모드들 중 2개(모드 2 및 모드 66)는 공통 각도를 공유한다. 따라서, 코딩 부담을 감소시키기 위해 모드들 2 및 66의 공통 각도를 활용하는 JVET를 코딩하는 시스템 및 방법이 필요하다.

**발명의 내용**

[0007] 동작시 시스템으로 하여금 액션들을 수행하게 하는 소프트웨어, 펌웨어, 하드웨어, 또는 이들의 조합을 시스템 상에 설치하는 것에 의해 특정 동작들 또는 액션들을 수행하도록 하나 이상의 컴퓨터의 시스템이 구성될 수 있다. 데이터 처리 장치에 의해 실행될 때, 장치로 하여금 액션들을 수행하게 하는 명령어들을 포함하는 것에 의해 특정 동작들 또는 액션들을 수행하도록 하나 이상의 컴퓨터 프로그램이 구성될 수 있다. 하나의 일반적인 양태는 CU x 및 CU y 좌표들을 갖는 CU(coding unit)를 비디오 프레임의 코딩 영역 내에서 정의하는 것을 포함하고, 이러한 정의하는 것은 상기 메인 참조와 연관된 메인 x 및 메인 y 좌표들을 갖는 메인 참조 픽셀을 상기 코딩 영역 내에서 정의하는 것을 또한 포함한다. 이러한 단계는 상기 사이드 참조와 연관된 사이드 x 및 사이드 y 좌표들을 갖는 사이드 참조 픽셀을 상기 코딩 영역 내에서 정의하는 것을 또한 포함할 수 있다. 이러한 시스템 및 방법은 예측 모드들의 세트를 정의하는 것 및/또는 상기 예측 모드들의 세트 내에서 2개의 이산 예측 모드들을 식별하는 것을 또한 포함할 수 있다. 추가로, 이러한 시스템 및 방법은 상기 예측 모드들의 세트로부터 예측 모드를 선택하는 것 및/또는 상기 메인 참조 픽셀과 상기 사이드 참조 픽셀의 조합에 적어도 부분적으로 기초하여 상기 코딩 유닛에 대한 예측 CU를 생성하는 것을 또한 포함할 수 있다. 추가적으로, 이러한 시스템 및 방법은 상기 코딩 유닛에 대한 상기 예측 CU가 상기 2개의 이산 예측 모드들 각각에 대해 동일한 방식으로 코딩되는 단계를 포함할 수 있으며, 상기 2개의 이산 예측 모드들 각각은 예측 방향에 적어도 부분적으로 기초하여 구별된다. 이러한 양태의 다른 실시예들은, 이러한 방법들의 액션들을 수행하도록 각각 구성되는, 대응하는 컴퓨터 시스템들, 장치들, 및 하나 이상의 컴퓨터 저장 디바이스 상에 기록되는 컴퓨터 프로그램들을 포함한다.

[0008] 구현들은 다음의 특징들 중 하나 이상을 포함할 수 있다: 예측 방향이 상기 코딩 유닛의 하나 이상의 특성에 기초하는 JVET 비디오를 코딩하는 방법; 상기 예측 CU가 엔트로피 코딩되는 JVET 비디오를 코딩하는 방법; 예측 방향이 상기 코딩 유닛의 폭에 적어도 부분적으로 기초하는 JVET 비디오를 코딩하는 방법; 및/또는 상기 예측 모드들이 0과 66 사이의 정수 값들의 모드들을 포함하는 JVET 비디오를 코딩하는 방법; 및/또는 상기 2개의 이산 예측 모드들이 모드 2 및 모드 66인 JVET 비디오를 코딩하는 방법. 그리고 일부 실시예들에서, JVET 비디오를 코딩하는 방법은, 예측 모드 2와 연관된 코딩이, 상기 메인 참조 픽셀과 연관된 메인 가중 값을 결정하는 것, 상기 사이드 참조 픽셀과 연관된 사이드 가중 값을 결정하는 것, 및 상기 메인 가중 값과 조합되는 상기 메인 참조 픽셀 및 상기 사이드 가중 값과 조합되는 상기 사이드 참조 픽셀의 조합에 적어도 부분적으로 기초하여 상기 코딩 유닛에 대한 예측 CU를 생성하는 것을 포함하는 단계에 기초할 수 있다. 설명되는 기술들의 구현들은 하드웨어, 방법 또는 프로세스, 또는 컴퓨터-액세스 가능 매체 상의 컴퓨터 소프트웨어를 포함할 수 있다.

**도면의 간단한 설명**

[0009] 본 발명의 추가의 상세사항들이 첨부된 도면들의 도움으로 설명된다.

도 1은 복수의 CTU들(Coding Tree Units)로의 프레임의 분할을 묘사한다.

도 2는 쿼드트리 파티셔닝 및 대칭 바이너리 파티셔닝을 사용하는 CU들(Coding Units)로의 CTU의 예시적인 파티셔닝을 묘사한다.

도 3은 도 2의 파티셔닝의 QTBT(quadtree plus binary tree) 표현을 묘사한다.

도 4는 2개의 더 작은 CU들로의 CU의 비대칭 바이너리 파티셔닝의 4개의 가능한 타입들을 묘사한다.

도 5는 쿼드트리 파티셔닝, 대칭 바이너리 파티셔닝, 및 비대칭 바이너리 파티셔닝을 사용하는 CU들로의 CTU의 예시적인 파티셔닝을 묘사한다.

도 6은 도 5의 파티셔닝의 QTBT 표현을 묘사한다.

도 7a 및 도 7b는 JVET 인코더에서의 CU 코딩을 위한 간략화된 블록도를 묘사한다.

도 8은 JVET에서의 루마 성분들에 대한 67개의 가능한 인트라 예측 모드들을 묘사한다.

도 9는 JVET 인코더에서의 CU 디코딩을 위한 간략화된 블록도를 묘사한다.

도 10은 JVET 인코더에서의 CU 코딩의 방법의 실시예를 묘사한다.

도 11은 JVET 인코더에서의 CU 코딩을 위한 간략화된 블록도를 묘사한다.

도 12는 JVET 디코더에서의 CU 디코딩을 위한 간략화된 블록도를 묘사한다.

도 13은 증가된 효율 코딩 시스템 및 방법의 간략화된 블록도를 묘사한다.

도 14는 JVET 인코더에서의 증가된 효율이 있는 CU 코딩을 위한 간략화된 블록도를 묘사한다.

도 15는 JVET 디코더에서의 증가된 효율이 있는 CU 디코딩을 위한 간략화된 블록도를 묘사한다.

도 16은 CU 코딩의 방법을 처리하도록 적응되는 및/또는 구성되는 컴퓨터 시스템의 실시예를 묘사한다.

도 17은 JVET 인코더/디코더에서의 CU 코딩/디코딩을 위한 코더/디코더 시스템의 실시예를 묘사한다.

**발명을 실시하기 위한 구체적인 내용**

[0010] 도 1은 복수의 CTU들(Coding Tree Units)(100)로의 프레임의 분할을 묘사한다. 프레임은 비디오 시퀀스에서의 이미지일 수 있다. 프레임은 이미지에서의 강도 척도들을 표현하는 픽셀 값들이 있는 매트릭스, 또는 매트릭스들의 세트를 포함할 수 있다. 따라서, 이러한 매트릭스들의 세트는 비디오 시퀀스를 생성할 수 있다. 픽셀 값들은 풀 컬러 비디오 코딩에서의 컬러 및 휘도를 표현하도록 정의될 수 있으며, 픽셀들은 3개의 채널들로 분할된다. 예를 들어, YCbCr 컬러 공간에서 픽셀들은, 이미지에서의 회색 레벨 강도를 표현하는 루마 값 Y, 및 컬러가 회색으로부터 청색 및 적색으로 상이한 정도를 표현하는 2개의 색차 값들 Cb 및 Cr을 가질 수 있다. 다른 실시예들에서, 픽셀 값들은 상이한 컬러 공간들 또는 모델들에서의 값들로 표현될 수 있다. 비디오의 해상도는 프레임에서의 픽셀들의 수를 결정할 수 있다. 더 높은 해상도는 이미지의 더 양호한 선명도 및 더 많은 픽셀들을 의미할 수 있지만, 더 높은 대역폭, 저장, 및 송신 요건들로 또한 이어질 수 있다.

[0011] 비디오 시퀀스의 프레임들은 JVET를 사용하여 인코딩 및 디코딩될 수 있다. JVET는 Joint Video Exploration Team에 의해 개발되고 있는 비디오 코딩 스킴이다. JVET의 버전들은 JEM(Joint Exploration Model) 인코더들 및 디코더들에서 구현되었다. HEVC(High Efficiency Video Coding)와 같은 다른 비디오 코딩 스킴들과 유사하게, JVET는 블록-기반 하이브리드 공간 및 시간 예측 코딩 스킴이다. JVET로 코딩하는 동안, 프레임은, 도 1에 도시되는 바와 같이, CTU들(100)이라고 불리는 정사각형 블록들로 먼저 분할된다. 예를 들어, CTU들(100)은 128x128 픽셀들의 블록일 수 있다.

[0012] 도 2는 CU들(102)로의 CTU(100)의 예시적인 파티셔닝을 묘사한다. 프레임에서의 각각의 CTU(100)는 하나 이상의 CU(Coding Units)(102)로 파티셔닝될 수 있다. CU들(102)은 아래에 설명되는 바와 같이 예측 및 변환을 위해 사용될 수 있다. HEVC와는 달리, JVET에서 CU들(102)은 직사각형 또는 정사각형일 수 있고, 예측 유닛들 또는 변환 유닛들로의 추가의 파티셔닝 없이 코딩될 수 있다. CU들(102)은 그들의 루트 CTU들(100)만큼 클 수 있거나, 또는 4x4 블록들만큼 작은 루트 CTU(100)의 더 작은 세분화들일 수 있다.

- [0013] JVET에서, CTU(100)는, CTU(100)가 쿼드트리에 따라 정사각형 블록들로 재귀적으로 분열될 수 있고, 다음으로 이러한 정사각형 블록들이 바이너리 트리들에 따라 수평으로 또는 수직으로 재귀적으로 분열될 수 있는 QTBT(quadtrees plus binary tree) 스킴에 따라 CU들(102)로 파티셔닝될 수 있다. CTU 크기, 쿼드트리 및 바이너리 트리 리프 노드들에 대한 최소 크기들, 바이너리 트리 루트 노드에 대한 최대 크기, 및 바이너리 트리들에 대한 최대 깊이와 같은, 파라미터들이 QTBT에 따라 분열을 제어하도록 설정될 수 있다.
- [0014] 일부 실시예들에서 JVET는 QTBT의 바이너리 트리 부분에서의 바이너리 파티셔닝을 대칭 파티셔닝으로 제한할 수 있고, 여기서 블록들은 중간선을 따라 수직으로 또는 수평으로 절반으로 분할될 수 있다.
- [0015] 비-제한적인 예로서, 도 2는 CU들(102)로 파티셔닝되는 CTU(100)를 도시하며, 실선들은 쿼드트리 분열을 표시하고 파선들은 대칭 바이너리 트리 분열을 표시한다. 예시되는 바와 같이, 바이너리 분열은 대칭 수평 분열 및 수직 분열을 허용하여 CTU의 구조 및 CU들로의 그것의 세분화를 정의한다.
- [0016] 도 3은 도 2의 파티셔닝의 QTBT 표현을 도시한다. 쿼드트리 루트 노드는 CTU(100)를 표현하고, 쿼드트리 부분에서의 각각의 자식 노드는 부모 정사각형 블록으로부터 분열되는 4개의 정사각형 블록들 중 하나를 표현한다. 쿼드트리 리프 노드들에 의해 표현되는 정사각형 블록들은 다음으로 바이너리 트리들을 사용하여 대칭적으로 0 회 이상 분할될 수 있고, 쿼드트리 리프 노드들은 바이너리 트리들의 루트 노드들이다. 바이너리 트리 부분의 각각의 레벨로, 블록이, 수직으로 또는 수평으로, 대칭적으로 분할될 수 있다. "0"으로 설정되는 플래그는 블록이 대칭적으로 수평으로 분열된다는 점을 표시하고, 한편 "1"로 설정되는 플래그는 블록이 대칭적으로 수직으로 분열된다는 점을 표시한다.
- [0017] 다른 실시예들에서, JVET는 QTBT의 바이너리 트리 부분에서의 대칭 바이너리 파티셔닝 또는 비대칭 바이너리 파티셔닝을 허용할 수 있다. PU들(prediction units)을 파티셔닝할 때 HEVC에서의 상이한 컨텍스트에서 AMP(asymmetrical motion partitioning)가 허용되었다. 그러나, QTBT 구조에 따라 JVET에서 CU들(102)을 파티셔닝하기 위해, CU(102)의 상관된 영역들이 CU(102)의 중심을 통해 이어지는 중간선의 어느 한 사이드 상에 배치되지 않을 때 비대칭 바이너리 파티셔닝은 대칭 바이너리 파티셔닝에 비해 개선된 파티셔닝으로 이어질 수 있다. 비-제한적인 예로서, CU(102)가 CU의 중심에 근접하는 하나의 객체 및 CU(102)의 사이드에 있는 다른 객체를 묘사할 때, CU(102)는 각각의 객체를 상이한 크기들의 별개의 더 작은 CU들(102)에 두도록 비대칭적으로 파티셔닝될 수 있다.
- [0018] 도 4는, CU(102)가 CU(102)의 길이 또는 높이에 걸쳐 이어지는 라인을 따라 2개의 더 작은 CU(102)로 분열되어, 더 작은 CU들(102) 중 하나가 부모 CU(102)의 크기의 25% 이고 다른 하나가 부모 CU(102)의 크기의 75%인 4개의 가능한 타입들의 비대칭 바이너리 파티셔닝을 묘사한다. 도 4에 도시되는 4개의 타입들의 비대칭 바이너리 파티셔닝은 CU(102)가 라인을 따라 CU(102)의 좌측 사이드로부터의 진로의 25%, CU(102)의 우측 사이드로부터의 진로의 25%, CU(102)의 상단으로부터의 진로의 25%, 또는 CU(102)의 하단으로부터의 진로의 25%로 분열되는 것을 허용한다. 대안적인 실시예들에서 CU(102)가 분열되는 비대칭 파티셔닝 라인은 CU(102)가 대칭적으로 절반으로 분할되지 않는 임의의 다른 위치에 배치될 수 있다.
- [0019] 도 5는 QTBT의 바이너리 트리 부분에서의 대칭 바이너리 파티셔닝 및 비대칭 바이너리 파티셔닝 양자 모두를 허용하는 스킴을 사용하여 CU들(102)로 파티셔닝되는 CTU(100)의 비-제한적인 예를 묘사한다. 도 5에서, 파선들은 도 4에 도시되는 파티셔닝 타입들 중 하나를 사용하여 부모 CU(102)가 분열된 비대칭 바이너리 파티셔닝 라인들을 묘사한다.
- [0020] 도 6은 도 5의 파티셔닝의 QTBT 표현을 도시한다. 도 6에서, 노드로부터 연장되는 2개의 실선은 QTBT의 바이너리 트리 부분에서의 대칭 파티셔닝을 표시하고, 한편 노드로부터 연장되는 2개의 파선은 바이너리 트리 부분에서의 비대칭 파티셔닝을 표시한다.
- [0021] CTU(100)가 CU들(102)로 어떻게 파티셔닝되었는지를 표시하는 비트스트림에서 구문이 코딩될 수 있다. 비-제한적인 예로서, 어느 노드들이 쿼드트리 파티셔닝으로 분열되었는지, 어느 것이 대칭 바이너리 파티셔닝으로 분열되었는지, 및 어느 것이 비대칭 바이너리 파티셔닝으로 분열되었는지를 표시하는 비트스트림에서 구문이 코딩될 수 있다. 유사하게, 도 4에 도시되는 4개의 타입 중 하나와 같은, 어느 타입의 비대칭 바이너리 파티셔닝이 사용되었는지를 표시하는 비대칭 바이너리 파티셔닝으로 분열되는 노드들에 대한 비트스트림에서 구문이 코딩될 수 있다.
- [0022] 일부 실시예들에서 비대칭 파티셔닝의 사용은 QTBT의 쿼드트리 부분의 리프 노드들에서 CU들(102)을 분열시키는 것으로 제한될 수 있다. 이러한 실시예들에서, 쿼드트리 부분에서의 쿼드트리 파티셔닝을 사용하여 부모 노드

로부터 분열된 자식 노드들에서의 CU들(102)이 최종 CU들(102)일 수 있거나, 또는 이들은 쿼드트리 파티셔닝, 대칭 바이너리 파티셔닝, 또는 비대칭 바이너리 파티셔닝을 사용하여 추가로 분열될 수 있다. 대칭 바이너리 파티셔닝을 사용하여 분열된 바이너리 트리 부분에서의 자식 노드들이 최종 CU들(102)일 수 있거나, 또는 이들은 대칭 바이너리 파티셔닝만을 사용하여 재귀적으로 1회 이상 추가로 분열될 수 있다. 비대칭 바이너리 파티셔닝을 사용하여 QT 리프 노드로부터 분열된 바이너리 트리 부분에서의 자식 노드들이 최종 CU들(102)일 수 있고, 추가의 분열이 허가되지 않는다.

[0023] 이러한 실시예들에서, 비대칭 파티셔닝의 사용을 쿼드트리 리프 노드들을 분열시키는 것으로 제한하는 것은 검색 복잡도를 감소시킬 수 및/또는 오버헤드 비트들을 제한할 수 있다. 쿼드트리 리프 노드들만이 비대칭 파티셔닝으로 분열될 수 있기 때문에, 비대칭 파티셔닝의 사용은 다른 구문 또는 추가의 시그널링 없이 QT 부분의 분기의 종료를 직접 표시할 수 있다. 유사하게, 비대칭적으로 파티셔닝되는 노드들은 추가로 분열될 수 없기 때문에, 노드 상의 비대칭 파티셔닝의 사용은 그 비대칭적으로 파티셔닝되는 자식 노드들이 다른 구문 또는 추가의 시그널링 없이 최종 CU들(102)이라는 점을 또한 직접 표시할 수 있다.

[0024] 대안적인 실시예들에서, 검색 복잡도를 제한하는 것 및/또는 오버헤드 비트들의 수를 제한하는 것이 관심이 적을 때와 같이, 쿼드트리 파티셔닝, 대칭 바이너리 파티셔닝, 및/또는 비대칭 바이너리 파티셔닝으로 생성되는 노드들을 분열시키는데 비대칭 파티셔닝이 사용될 수 있다.

[0025] 위에 설명된 QTBT 구조를 사용하는 쿼드트리 분열 및 바이너리 트리 분열 후에, QTBT의 리프 노드들에 의해 표현되는 블록들은, 인터 예측 또는 인트라 예측을 사용하는 코딩과 같이, 코딩될 최종 CU들(102)을 표현한다. 인터 예측으로 코딩되는 슬라이스들 또는 풀 프레임들에 대해서는, 루마 및 크로마 성분들에 대해 상이한 파티셔닝 구조들이 사용될 수 있다. 예를 들어, 인터 슬라이스에 대해 CU(102)는, 하나의 루마 CB 및 2개의 크로마 CB들과 같은, 상이한 컬러 성분들에 대한 CB들(Coding Blocks)을 가질 수 있다. 인트라 예측으로 코딩되는 슬라이스들 또는 풀 프레임들에 대해서는, 파티셔닝 구조가 루마 및 크로마 성분들에 대해 동일할 수 있다.

[0026] 대안적인 실시예들에서 JVET는 위에 설명된 QTBT 파티셔닝에 대한 대안으로서, 또는 그 연장으로서, 2-레벨 코딩 블록 구조를 사용할 수 있다. 2-레벨 코딩 블록 구조에서, CTU(100)는 BU들(base units)로 하이 레벨로 먼저 파티셔닝될 수 있다. 다음으로 BU들은 OU들(operating units)로 로우 레벨로 파티셔닝될 수 있다.

[0027] 2-레벨 코딩 블록 구조를 채택하는 실시예들에서, 하이 레벨로 CTU(100)는 위에 설명된 QTBT 구조들 중 하나에 따라, 또는 블록들이 4개의 동일한 크기의 서브-블록들로 분열될 수만 있는 HEVC에서 사용되는 것과 같은 QT(quadtree) 구조에 따라 BU들로 파티셔닝될 수 있다. 비-제한적인 예로서, CTU(102)는 도 5 내지 도 6에 관하여 위에 설명된 QTBT 구조에 따라 BU들로 파티셔닝될 수 있어, 쿼드트리 부분에서의 리프 노드들은 쿼드트리 파티셔닝, 대칭 바이너리 파티셔닝, 또는 비대칭 바이너리 파티셔닝을 사용하여 분열될 수 있다. 이러한 예에서는, QTBT의 최종 리프 노드들이 CU들 대신에 BU들일 수 있다.

[0028] 2-레벨 코딩 블록 구조에서의 하위 레벨로, CTU(100)로부터 파티셔닝되는 각각의 BU가 하나 이상의 OU로 추가로 파티셔닝될 수 있다. 일부 실시예들에서, BU가 정사각형일 때, 이것은, 대칭 또는 비대칭 바이너리 파티셔닝과 같은, 쿼드트리 파티셔닝 또는 바이너리 파티셔닝을 사용하여 OU들로 분열될 수 있다. 그러나, BU가 정사각형이 아닐 때, 이것은 바이너리 파티셔닝만을 사용하여 OU들로 분열될 수 있다. 비-정사각형 BU들에 대해 사용될 수 있는 파티셔닝의 타입을 제한하는 것은 BU들을 생성하기 위해 사용되는 파티셔닝의 타입을 시그널링하기 위해 사용되는 비트들의 수를 제한할 수 있다.

[0029] 아래의 논의가 CU들(102)을 코딩하는 것을 설명하더라도, 2-레벨 코딩 블록 구조를 사용하는 실시예들에서 CU들(102) 대신에 BU들 및 OU들이 코딩될 수 있다. 비-제한적인 예들로서, 인트라 예측 또는 인터 예측과 같은 상위 레벨 코딩 동작들에 대해 BU들이 사용될 수 있고, 한편 변환들 및 변환 계수들을 생성하는 것과 같은 하위 레벨 코딩 동작들에 대해 더 작은 OU들이 사용될 수 있다. 따라서, 그들이 인트라 예측 또는 인터 예측으로 코딩되는지를 표시하는 BU들에 대해 코딩되기 위한 구문, 또는 BU들 코딩하기 위해 사용되는 모션 벡터들 또는 특정 인트라 예측 모드들을 식별하는 정보. 유사하게, OU들에 대한 구문은 OU들을 코딩하기 위해 사용되는 양자화된 변환 계수들 또는 특정 변환 동작들을 식별할 수 있다.

[0030] 도 7a는 JVET 인코더에서의 CU 코딩을 위한 간략화된 블록도를 묘사한다. 비디오 코딩의 메인 스테이지들은 위에 설명된 바와 같이 CU들(102)을 식별하기 위한 파티셔닝을 포함하고, 704 또는 706에서의 예측, 708에서의 잔여 CU(710)의 생성, 712에서의 변환, 716에서의 양자화, 및 720에서의 엔트로피 코딩을 사용하여 CU들(102)을 인코딩하는 것이 뒤따른다. 도 7a에 예시되는 인코더 및 인코딩 프로세스는 아래에 더 상세히 설명되는 디코딩

프로세스를 또한 포함한다.

- [0031] 현재 CU(102)가 주어지면, 인코더는 704에서의 인트라 예측을 사용하여 공간적으로 또는 706에서의 인터 예측을 사용하여 시간적으로 예측 CU(702)를 획득할 수 있다. 예측 코딩의 기본 아이디어는 원래 신호와 원래 신호에 대한 예측 사이의 차동, 또는 잔여, 신호를 송신하는 것이다. 수신기 측에서, 아래에 설명되는 바와 같이, 잔여 및 예측을 추가하는 것에 의해 원래 신호가 재구성될 수 있다. 차동 신호는 원래 신호보다 더 낮은 상관을 갖기 때문에, 그 송신을 위해 더 적은 비트들이 필요하다.
- [0032] 인트라-예측 CU들(102)로 전체적으로 코딩되는, 전체 화면 또는 화면의 일부분과 같은, 슬라이스는 다른 슬라이스들을 참조하지 않고 디코딩될 수 있는 I 슬라이스일 수 있고, 이와 같이 디코딩이 시작될 수 있는 가능한 포인트일 수 있다. 적어도 일부 인터-예측된 CU들로 코딩되는 슬라이스는 하나 이상의 참조 화면에 기초하여 디코딩될 수 있는 예측(P) 또는 쌍-예측(B) 슬라이스일 수 있다. P 슬라이스들은 이전에 코딩된 슬라이스들과의 인터-예측 및 인트라-예측을 사용할 수 있다. 예를 들어, P 슬라이스들은 인터-예측의 사용에 의해 I-슬라이스들보다 더욱 압축될 수 있지만, 이들을 코딩하기 위해 이전에 코딩된 슬라이스의 코딩을 필요로 한다. B 슬라이스들은, 2개의 상이한 프레임들로부터의 보간된 예측을 사용하는 인트라-예측 또는 인터-예측을 사용하여, 그 코딩을 위해 이전 및/또는 후속 슬라이스들로부터의 데이터를 사용할 수 있고, 따라서 모션 추정 프로세스의 정확도를 증가시킨다. 일부 경우들에서, P 슬라이스들 및 B 슬라이스들은 인트라 블록 사본을 사용하여 또한 또는 대안적으로 인코딩될 수 있으며, 여기서 동일한 슬라이스의 다른 부분들로부터의 데이터가 사용된다.
- [0033] 아래에 논의되는 바와 같이, 인트라 예측 또는 인터 예측은, 이웃 CU들(102) 또는 참조 화면들에서의 CU들(102)과 같은, 이전에 코딩된 CU들(102)로부터의 재구성된 CU들(734)에 기초하여 수행될 수 있다.
- [0034] CU(102)가 704에서의 인트라 예측으로 공간적으로 코딩될 때, 화면에서의 이웃 CU들(102)로부터의 샘플들에 기초하여 CU(102)의 픽셀 값들을 최상으로 예측하는 인트라 예측 모드가 발견될 수 있다.
- [0035] CU의 루마 성분을 코딩할 때, 인코더는 후보 인트라 예측 모드들의 리스트를 생성할 수 있다. HEVC는 루마 성분들에 대한 35개의 가능한 인트라 예측 모드들을 갖는 한편, JVET에서는 루마 성분들에 대한 67개의 가능한 인트라 예측 모드들이 존재한다. 이들은 이웃 픽셀들로부터 생성되는 값들의 3차원 평면을 사용하는 평면 모드, 이웃 픽셀들로부터 평균화되는 값들을 사용하는 DC 모드, 및 표시된 방향들을 따라 이웃 픽셀들로부터 복사되는 값들을 사용하는 도 8에 도시되는 65개의 방향성 모드들을 포함한다.
- [0036] CU의 루마 성분에 대한 후보 인트라 예측 모드들의 리스트를 생성할 때, 이러한 리스트 상의 후보 모드들의 수는 CU의 크기에 의존할 수 있다. 후보 리스트는 SATD(Sum of Absolute Transform Difference) 비용들이 최저인 HEVC의 35개의 모드들의 서브세트; HEVC 모드들로부터 발견되는 후보들에 이웃하는 JVET에 대해 추가되는 새로운 방향성 모드들; 이전에 코딩된 이웃 블록들에 대해 사용되는 인트라 예측 모드들 뿐만 아니라 디폴트 모드들의 리스트에 기초하여 식별되는 CU(102)에 대한 6개의 MPM들(most probable modes)의 세트로부터의 모드들을 포함할 수 있다.
- [0037] CU의 크로마 성분들을 코딩할 때, 후보 인트라 예측 모드들의 리스트가 또한 생성될 수 있다. 후보 모드들의 리스트는 루마 샘플들로부터의 크로스-성분 선형 모델 투영으로 생성되는 모드들, 크로마 블록에서 특정 병치된 위치들에서의 루마 CB들에 대해 발견되는 인트라 예측 모드들, 및 이웃 블록들에 대해 이전에 발견된 크로마 예측 모드들을 포함할 수 있다. 인코더는 레이트 왜곡 비용들이 최저인 리스트들 상의 후보 모드들을 발견하고, CU의 루마 및 크로마 성분들을 코딩할 때 이러한 인트라 예측 모드들을 사용할 수 있다. 각각의 CU(102)를 코딩하기 위해 사용되는 인트라 예측 모드들을 표시하는 비트스트림에서 구분이 코딩될 수 있다.
- [0038] CU(102)에 대한 최상의 인트라 예측 모드가 선택된 후에, 인코더는 이러한 모드들을 사용하여 예측 CU(402)를 생성할 수 있다. 선택된 모드들이 방향성 모드들일 때, 방향성 정확도를 개선하는데 4-탭 필터가 사용될 수 있다. 예측 블록의 상단 또는 좌측에서의 열들 또는 행들은, 2-탭 또는 3-탭 필터들과 같은, 경계 예측 필터들로 조정될 수 있다.
- [0039] 예측 CU(702)는 이웃 블록들의 필터링되지 않은 샘플들을 사용하는 이웃 블록들의 필터링된 샘플들, 또는 3-탭 또는 5-탭 로우 패스 필터들을 사용하는 적응성 참조 샘플 평활화에 기초하여 생성되는 예측 CU(702)를 조정하여 단계 705b에서 참조 샘플들을 처리하는 PDPC(position dependent intra prediction combination) 프로세스를 추가로 평활화될 수 있다. 일부 실시예들에서, PDPC는 다음의 식 (1)에 따라 달성될 수 있다:

[0040] 
$$P'[x,y] = ((A*Recon[x,-1] - B*Recon[-1,-1] + C*Recon[-1,y] + D*P[x,y] + Round)/Denom$$

- [0041] 식 (1)
- [0042] 여기서  $A = (Cv1 \gg \text{int}(y/dy))$ ,  $B = ((Cv2 \gg \text{int}(y/dy)) + (Ch2 \gg \text{int}(x/dx)))$ ,  $C = (Ch1 \gg \text{int}(x/dx))$ , 및  $D = (1 \ll \text{Denom}) - A - C + B$ 이다. 이러한  $P[x,y]$ 는 현재 CU의 좌표(x,y)에서의 사후-필터링 동작 후의 필터링된 픽셀이다. Cv1, Cv2, Ch1, Ch2는 필터링 효과를 결정하는 PDPC 파라미터들이고, 'Round'는 라운딩 파라미터이고 'Denom'은 정규화 인자이다.
- [0043] 일부 실시예들에서, 상단 참조 행 및 좌측 참조 열 양자 모두 상의 투영 위치들에서의 픽셀들을 사용하여 각도 예측을 위한 예측기 픽셀들을 생성하는 가중 각도 예측이 채택될 수 있다. 가중 각도 예측을 채택하는 실시예들에서, 예측 생성은 3개의 단계들 - 메인 참조 투영 예측, 사이드 참조 투영 예측 및 이러한 투영 예측들의 조합으로 행해질 수 있다.
- [0044] 가중 각도 예측을 채택하는 일부 실시예들에서, 이러한 시스템 및 방법은 코딩 인트라 예측 모드의 각도 방향 선명도에 따라서 메인 참조를 따라 픽셀 위치를 투영하고, 2개의 이웃하는 재구성된 픽셀들 사이의 선형 보간을 사용하여 투영 위치의 픽셀 값을 결정할 수 있다. 이러한 시스템 및 방법은 또한 동일한 코딩 모드의 각도 선명도에 따라서 사이드 참조를 따라 픽셀 위치를 투영하고, 2개의 이웃하는 재구성된 픽셀들 사이의 선형 보간을 사용하여 투영 위치의 픽셀 값을 결정할 수 있다. 다음으로, 이러한 시스템 및 방법은 메인 참조의 투영 픽셀 값을 사이드 참조의 투영 픽셀 값과 조합할 수 있다. 비-제한적 예시적인 조합이 식 (2)에서 아래에 도시된다. 식 (2)에서 도시되는 예시적인 조합에서 값들은 메인 및 사이드 참조들 상의 예측기 픽셀들과 투영 픽셀 위치들 사이의 거리에 따라 가중된다. 그러나, 대안적인 실시예들에서는 메인 및 사이드 참조 픽셀들과 연관된 값들을 가중하는데 대안적인 값들이 사용될 수 있다.
- [0045] 
$$P[x,y] = (((w1 * \text{MainRecon}[x',y']) + (w2 * \text{SideRecon}[x'',y''])) + (w1+w2)/2) / (w1+w2)$$
- [0046] 식 (2)
- [0047] 위의 예시적인 식 (2)에서,  $\text{MainRecon}[x',y']$ 는 메인 참조를 따르는, 예측 픽셀(x,y)에 대응하는, 투영 위치(x', y')에서의 이웃의 픽셀 값이다.  $\text{SideRecon}[x'',y'']$ 는 사이드 참조를 따르는, 예측 픽셀(x,y)에 대응하는 투영 위치(x'', y'')에서의 이웃의 픽셀 값이다.
- [0048] 아래의 식(3)은 HEVC 모드 2 또는 모드 66을 사용하는 가중 각도 예측, 및 좌표(x,y)에서의 예측기 픽셀을 사용하는 비-제한적 예시적인 조합을 도시한다. 따라서,  $P[x,y]$ 는 식 (3)에 도시되고 설명되는 바와 같이 결정될 것이고, 여기서  $\text{Recon}[0,0]$ 은 현재 CU의 상단 좌측 좌표(0,0)에서의 재구성된 픽셀이다.
- [0049] 
$$P[x,y] = (((x+1) * \text{Recon}[x+y+2,-1]) + ((y+1) * (\text{Recon}[-1,x+y+2]))) + (y+x+2)/2) / (y+x+2)$$
- [0050] 식 (3)
- [0051] 가중 각도 예측이 채택되지 않을 수 있는 시스템 및 프로세스에 대한 예외는, 사이드 참조 상의 투영 참조 위치가 자립가능한 위치가 아닌 또는 이용가능하지 않은 재구성된 위치를 참조할 때 발생할 수 있다. 가중 각도 예측이 채택되지 않을 수 있는 이러한 경우들에서, 이러한 예외를 취급하기 위해 다수의 옵션들이 가능하다. 일부 실시예들에서, 이러한 예외는 마지막 이용가능 재구성된 픽셀의 값 또는 투영 위치에 대한 디폴트 값을 사용하여 취급될 수 있다. 다른 대안적인 실시예들에서, 이러한 예외는 가중 각도 예측을 디스에이블하는 것 및/또는 메인 참조의 투영 픽셀 위치만을 사용하는 것에 의해 취급될 수 있다. 따라서, 단계 705a에서, 가중 각도 예측이 단계 704에서 인트라 예측 모드로서 채택되었는지가 결정될 수 있다. 단계 705a에서, 인트라 예측 모드가 가중 각도 예측을 사용하는 것으로서 결정되면, 예측 코딩 유닛(702)은 엔트로피 코딩 부재 필터링을 위해 전달될 수 있다. 그러나, 단계 705a에서, 인트라 예측 모드가 가중 각도 예측 이외인 것으로 결정되면, 엔트로피 코딩을 위한 전달 이전에 예측 코딩 유닛에, PDPC 필터링과 같은, 사후 인트라 예측 필터링(705b)이 적용될 수 있다.
- [0052] 도 7b에 묘사되는 바와 같이, 일부 실시예들에서, 사후 인트라 예측 필터(705b)는 모든 인트라 예측들에 대해 단계 704 이후에 채택될 수 있다. 도 7b에 묘사되는 이러한 실시예들에서, 인트라 예측 모드가 가중 각도 예측 이외의 것에 기초하면, 적용되는 필터는 단계 705b에서 정상적으로 적용될 것으로서 적용될 수 있다. 그러나, 인트라 예측 모드가 가중 각도 예측에 기초하면, 단계 705b에서의 필터링이 바이패스될 수 있고 및/또는 일부 실시예들에서는, 적용되는 필터가 메인 참조, 사이드 참조 또는 메인 및 사이드 참조들을 향해 바이어스되지 않을 수 있다. 비-제한적인 예로서, Cv1 및 Ch1의 값들은 동일할 수 있고 및/또는 Cv2 및 Ch2의 값들은 동일할 수 있다.

- [0053] 706에서의 인터 예측으로 CU(102)가 시간적으로 코딩될 때, CU(102)의 픽셀 값을 최상으로 예측하는 참조 화면들에서의 샘플들을 포인팅하는 MV들(motion vectors)의 세트가 발견될 수 있다. 인터 예측은 슬라이스에서의 픽셀들의 블록의 변위를 표현하는 것에 의해 슬라이스들 사이의 시간 중복성을 활용한다. 이러한 변위는 모션 보상이라고 불리는 프로세스를 통해 이전 또는 다음 슬라이스들에서의 픽셀들의 값에 따라 결정된다. 특정 참조 화면에 상대적인 픽셀 변위를 표시하는 연관된 참조 인덱스들 및 모션 벡터들이, 원래의 픽셀들과 모션 보상된 픽셀들 사이의 잔여와 함께, 비트스트림에서 디코더에 제공될 수 있다. 디코더는 재구성된 슬라이스에서의 픽셀들의 블록을 재구성하는데 잔여 및 시그널링된 모션 벡터들 및 참조 인덱스들을 사용할 수 있다.
- [0054] JVET에서, 모션 벡터 정확도는 1/16 화소에서 저장될 수 있고, 모션 벡터와 CU의 예측 모션 벡터 사이의 차이는 1/4-화소 해상도 또는 정수-화소 해상도로 코딩될 수 있다.
- [0055] JVET에서 모션 벡터들은, ATMVP(advanced temporal motion vector prediction), STMVP(spatial-temporal motion vector prediction), 아핀 모션 보상 예측, PMMVD(pattern matched motion vector derivation), 및/또는 BIO(bi-directional optical flow)와 같은 기술들을 사용하여, CU(102) 내의 다수의 서브-CU들에 대해 발견될 수 있다.
- [0056] ATMVP를 사용하여, 인코더는 참조 화면에서의 대응하는 블록을 포인팅하는 CU(102)에 대한 시간 벡터를 발견할 수 있다. 시간 벡터는 이전에 코딩된 이웃 CU들(102)에 대해 발견되는 참조 화면들 및 모션 벡터들에 기초하여 발견될 수 있다. 전체 CU(102)에 대한 시간 벡터에 의해 포인팅되는 참조 블록을 사용하여, CU(102) 내의 각각의 서브-CU에 대해 모션 벡터가 발견될 수 있다.
- [0057] STMVP는, 시간 벡터와 함께, 인터 예측으로 이전에 코딩된 이웃 블록들에 대해 발견되는 모션 벡터들을 스케일링 및 평균화하는 것에 의해 서브-CU들에 대한 모션 벡터들을 발견할 수 있다.
- [0058] 블록의 상단 코너들에 대해 발견되는 2개의 제어 모션 벡터들에 기초하여, 블록에서의 각각의 서브-CU에 대한 모션 벡터들의 필드를 예측하는데 아핀 모션 보상 예측이 사용될 수 있다. 예를 들어, CU(102) 내의 각각의 4x4 블록에 대해 발견되는 상단 코너 모션 벡터들에 기초하여 서브-CU들에 대한 모션 벡터들이 도출될 수 있다.
- [0059] PMMVD는 쌍방 매칭 또는 템플릿 매칭을 사용하여 현재 CU(102)에 대한 초기 모션 벡터를 발견할 수 있다. 쌍방 매칭은 모션 궤적을 따라 2개의 상이한 참조 화면들에서의 참조 블록들 및 현재 CU(102)를 볼 수 있고, 한편 템플릿 매칭은 현재 CU(102)에서의 대응하는 블록들 및 템플릿에 의해 식별되는 참조 화면을 볼 수 있다. CU(102)에 대해 발견되는 초기 모션 벡터가 다음으로 각각의 서브-CU에 대해 개별적으로 정제될 수 있다.
- [0060] 인터 예측이 이전의 그리고 차후의 참조 화면들에 기초하여 쌍-예측으로 수행되고, 2개의 참조 화면들 사이의 차이의 변화도에 기초하여 서브-CU들에 대한 모션 벡터들이 발견되는 것을 허용할 때, BIO가 사용될 수 있다.
- [0061] 일부 상황들에서는, 현재 CU(102)에 이웃하는 샘플들 및 후보 모션 벡터에 의해 식별되는 참조 블록에 이웃하는 대응하는 샘플들에 기초하여, 스케일링 인자 파라미터 및 오프셋 파라미터에 대한 값들을 발견하는데 CU 레벨로 LIC(local illumination compensation)가 사용될 수 있다. JVET에서, 이러한 LIC 파라미터들은 CU 레벨로 변경되고 시그널링될 수 있다.
- [0062] 위 방법들 중 일부에 대해, CU의 서브-CU들 각각에 대해 발견되는 모션 벡터들이 CU 레벨로 디코더들에 시그널링될 수 있다. PMMVD 및 BIO와 같은, 다른 방법들에 대해, 모션 정보는 오버헤드를 절감하기 위해 비트스트림에서 시그널링되지 않고, 디코더들은 동일한 프로세스들을 통해 모션 벡터들을 도출할 수 있다.
- [0063] CU(102)에 대한 모션 벡터들이 발견된 후, 인코더는 이러한 모션 벡터들을 사용하여 예측 CU(702)를 생성할 수 있다. 일부 경우들에서, 모션 벡터들이 개별 서브-CU들에 대해 발견되었을 때, 이러한 모션 벡터들을 하나 이상의 이웃 서브-CU들에 대해 이전에 발견된 모션 벡터들과 조합하는 것에 의해 예측 CU(702)를 생성할 때 OBMC(Overlapped Block Motion Compensation)가 사용될 수 있다.
- [0064] 쌍-예측이 사용될 때, JVET는 모션 벡터들을 발견하는데 DMVR(decoder-side motion vector refinement)을 사용할 수 있다. DMVR은 쌍방 템플릿 매칭 프로세스를 사용하여 쌍-예측에 대해 발견되는 2개의 모션 벡터들에 기초하여 모션 벡터가 발견되는 것을 허용한다. DMVR에서, 2개의 모션 벡터들 각각으로 생성되는 예측 CU들(702)의 가중 조합이 발견될 수 있고, 2개의 모션 벡터들은 조합된 예측 CU(702)에 최상으로 포인팅하는 새로운 모션 벡터들로 이들을 대체하는 것에 의해 정제될 수 있다. 2개의 정제된 모션 벡터들은 최종 예측 CU(702)를 생성하기 위해 사용될 수 있다.

- [0065] 708에서, 앞서 설명된 바와 같이, 704에서의 인트라 예측 또는 706에서의 인터 예측으로 일단 예측 CU(702)가 발견되었으면, 인코더는 현재 CU(102)로부터 예측 CU(702)를 감산하여 잔여 CU(710)를 발견할 수 있다.
- [0066] 인코더는, 데이터를 변환 도메인으로 변환하는데 DCT-transform(discrete cosine block transform)을 사용하는 것과 같이, 잔여 CU(710)를 변환 도메인에서의 잔여 CU(710)를 표현하는 변환 계수들(714)로 변환하는데 712에서의 하나 이상의 변환 동작을 사용할 수 있다. JVET는, DCT-II, DST-VII, DST-VIII, DCT-VIII, DST-I, 및 DCT-V 동작들을 포함하는, HEVC보다 많은 타입들의 변환 동작들을 허용한다. 허용된 변환 동작들은 서브-세트들로 그룹화될 수 있고, 어느 서브-세트들 및 이러한 서브-세트들에서의 어느 구체적인 동작들이 사용되었는지의 표시가 인코더에 의해 시그널링될 수 있다. 일부 경우들에서는, 특정 크기보다 더 큰 CU들(102)에서의 고주파 변환 계수들을 제로화하는데 큰 블록 크기 변환들이 사용될 수 있어, 더 낮은-주파수 변환 계수들만이 이러한 CU들(102)에 대해 유지된다.
- [0067] 일부 경우들에서는 순방향 코어 변환 후에 저주파 변환 계수들(714)에 MDNSST(mode dependent non-separable secondary transform)가 적용될 수 있다. MDNSST 동작은 회전 데이터에 기초하여 HyGT(Hypercube-Givens Transform)를 사용할 수 있다. 사용될 때, 특정 MDNSST 동작을 식별하는 인덱스 값이 인코더에 의해 시그널링될 수 있다.
- [0068] 716에서, 인코더는 변환 계수들(714)을 양자화된 변환 계수들(716)로 양자화할 수 있다. 각각의 계수의 양자화는 양자화 단계에 의해 계수의 값을 분할하는 것에 의해 계산될 수 있고, 이는 QP(quantization parameter)로부터 도출된다. 일부 실시예들에서, Qstep은  $2^{(QP-4)/6}$  으로서 정의된다. 고정밀 변환 계수들(714)은 가능한 값들이 유한 수인 양자화된 변환 계수들(716)로 변환될 수 있기 때문에, 양자화는 데이터 압축을 보조할 수 있다. 따라서, 변환 계수들의 양자화는 변환 프로세스에 의해 생성되고 전송되는 비트들의 양을 제한할 수 있다. 그러나, 양자화는 손실성 동작이고, 양자화에 의한 손실은 복구될 수 없는 한편, 양자화 프로세스는 재구성된 시퀀스의 품질과 시퀀스를 표현하는데 필요한 정보의 양 사이의 트레이드-오프를 제시한다. 예를 들어, 표현 및 송신을 위해 더 높은 양의 데이터가 요구될 수 있더라도, 더 낮은 QP 값은 더 양호한 품질의 디코딩된 비디오를 초래할 수 있다. 반대로, 데이터 및 대역폭 수요들이 더 낮더라도 높은 QP 값은 더 낮은 품질의 재구성된 비디오 시퀀스들을 초래할 수 있다.
- [0069] JVET는 분산-기반 적응성 양자화 기술들을 이용할 수 있는데, 이는 모든 CU(102)가(프레임의 모든 CU(102)의 코딩에서 동일한 프레임 QP를 사용하는 대신에) 자신의 코딩 프로세스에 대해 상이한 양자화 파라미터를 사용하는 것을 허용한다. 분산-기반 적응성 양자화 기술들은 특정 블록들의 양자화 파라미터를 적응성으로 낮추는 한편 다른 것들에서는 이를 증가시킨다. CU(102)에 대한 구체적인 QP를 선택하기 위해, CU의 분산이 계산된다. 간단히, CU의 분산이 프레임의 평균 분산보다 더 높으면, 프레임의 QP보다 더 높은 QP가 CU(102)에 대해 설정될 수 있다. CU(102)가 프레임의 평균 분산보다 더 낮은 분산을 제시하면, 더 낮은 QP가 배정될 수 있다.
- [0070] 720에서, 인코더는 양자화된 변환 계수들(718)을 엔트로피 코딩하는 것에 의해 최종 압축 비트들(722)을 발견할 수 있다. 엔트로피 코딩은 송신될 정보의 통계적 중복성들을 제거하는 것을 목적으로 한다. JVET에서, CABAC(Context Adaptive Binary Arithmetic Coding)는 양자화된 변환 계수들(718)을 코딩하기 위해 사용될 수 있으며, 이는 통계적 중복성들을 제거하는데 확률 척도들을 사용한다. 양자화된 변환 계수들(718)이 0이 아닌 CU들(102)에 대해, 양자화된 변환 계수들(718)은 바이너리로 변환될 수 있다. 바이너리 표현의 각각의 비트("bin(bin)")는 다음으로 컨텍스트 모델을 사용하여 인코딩될 수 있다. CU(102)는 3개의 영역들로 나뉠 수 있고, 각각은 해당 영역 내의 픽셀들에 대해 사용할 자신의 컨텍스트 모델들의 세트가 있다.
- [0071] bin들을 인코딩하기 위해 다수의 스캔 패스들이 수행될 수 있다. 처음 3개의 bin들(bin0, bin1, 및 bin2)을 인코딩하기 위한 패스들 동안, bin에 대해 어느 컨텍스트 모델을 사용할지를 표시하는 인덱스 값은 템플릿에 의해 식별되는 5개까지의 이전에 코딩된 이웃 양자화된 변환 계수들(718)에서의 해당 bin 위치의 합을 발견하는 것에 의해 발견될 수 있다.
- [0072] 컨텍스트 모델은 '0' 또는 '1'인 bin의 값의 확률들에 기초할 수 있다. 값들이 코딩됨에 따라, 컨텍스트 모델에서의 확률들은 마주치는 '0' 및 '1' 값들의 실제 수에 기초하여 업데이트될 수 있다. HEVC는 각각의 새로운 화면에 대한 컨텍스트 모델들을 재-초기화하는데 고정 테이블들을 사용하는 한편, JVET에서 새로운 인터-예측된 화면들에 대한 컨텍스트 모델들의 확률들은 이전에 코딩된 인터-예측된 화면들에 대해 개발되는 컨텍스트 모델들에 기초하여 초기화될 수 있다.
- [0073] 인코더는 잔여 CU들(710)의 엔트로피 인코딩된 비트들(722), 선택된 인트라 예측 모드들 또는 모션 벡터들과 같

은 예측 정보, QTBT 구조에 따라 CU들(102)이 어떻게 CTU(100)로부터 파티셔닝되었는지의 표시자들, 및/또는 인코딩된 비디오에 관한 다른 정보를 포함하는 비트스트림을 생산할 수 있다. 이러한 비트스트림은 아래에 논의되는 바와 같이 디코더에 의해 디코딩될 수 있다.

[0074] 최종 압축 비트들(722)을 발견하는데 양자화된 변환 계수들(718)을 사용하는 것에 추가로, 인코더는 디코더가 재구성된 CU들(734)을 생성하기 위해 사용하는 것과 동일한 디코딩 프로세스를 따르는 것에 의해 재구성된 CU들(734)을 생성하는데 양자화된 변환 계수들(718)을 또한 사용할 수 있다. 따라서, 일단 변환 계수들이 인코더에 의해 계산되고 양자화되었으면, 양자화된 변환 계수들(718)은 인코더에서 디코딩 루프에 송신될 수 있다. CU의 변환 계수들의 양자화 후에, 디코딩 루프는 디코딩 프로세스에서 디코더가 생성하는 것과 동일한 재구성된 CU(734)를 인코더가 생성하는 것을 허용한다. 따라서, 인코더는 새로운 CU(102)에 대한 인트라 예측 또는 인터 예측을 수행할 때 이웃 CU들(102) 또는 참조 화면들에 대해 디코더가 사용하는 것과 동일한 재구성된 CU들(734)을 사용할 수 있다. 재구성된 CU들(102), 재구성된 슬라이스들, 또는 전체 재구성된 프레임들은 추가의 예측 스테이지들에 대한 참조들로서 역할을 할 수 있다.

[0075] 재구성된 이미지에 대한 픽셀 값들을 획득하기 위해 인코더의 디코딩 루프에서(디코더에서 동일한 동작들에 대해, 아래 참조), 역양자화 프로세스가 수행될 수 있다. 프레임을 역양자화하기 위해, 예를 들어, 프레임의 각각의 픽셀에 대한 양자화된 값은 위에 설명된 양자화 단계, 예를 들어, (Qstep)와 승산되어, 재구성된 역양자화된 변환 계수들(726)을 획득한다. 예를 들어, 인코더에서 도 7a에 도시되는 디코딩 프로세스에서, 잔여 CU(710)의 양자화된 변환 계수들(718)은 역양자화된 변환 계수들(726)을 발견하기 위해 724에서 역양자화될 수 있다. MDNSST 동작이 인코딩 동안 수행되었다면, 해당 동작은 역양자화 후에 반전될 수 있다.

[0076] 728에서, 역양자화된 변환 계수들(726)은, 재구성된 이미지를 획득하기 위해 값들에 DCT를 적용하는 것에 의해서와 같이, 재구성된 잔여 CU(730)를 발견하기 위해 역 변환될 수 있다. 732에서 재구성된 잔여 CU(730)는, 재구성된 CU(734)를 발견하기 위해, 704에서의 인트라 예측 또는 706에서의 인터 예측으로 발견되는 대응하는 예측 CU(702)에 추가될 수 있다.

[0077] 736에서, 화면 레벨 또는 CU 레벨로, 하나 이상의 필터가(인코더에서 또는, 아래에 설명되는 바와 같이, 디코더에서, ) 디코딩 프로세스 동안 재구성된 데이터에 적용될 수 있다. 예를 들어, 인코더는 디블록킹 필터, SAO(sample adaptive offset) 필터, 및/또는 ALF(adaptive loop filter)를 적용할 수 있다. 인코더의 디코딩 프로세스는 재구성된 이미지에서 잠재적인 아티팩트들을 다룰 수 있는 최적의 필터 파라미터들을 추정하고 이를 디코더에 송신하기 위해 필터들을 구현할 수 있다. 이러한 개선들은 재구성된 비디오의 객관적 및 주관적 품질을 증가시킨다. 디블록킹 필터링에서는, 서브-CU 경계 근처의 픽셀들이 수정될 수 있고, 반면 SAO에서는, CTU(100)에서의 픽셀들이 에지 오프셋 또는 대역 오프셋 분류를 사용하여 수정될 수 있다. JVET의 ALF는 각각의 2x2 블록에 대한 원형 대칭 형상들이 있는 필터들을 사용할 수 있다. 각각의 2x2 블록에 사용되는 필터의 크기 및 아이덴티티의 표시가 시그널링될 수 있다. 대안적으로, 가중 각도 예측이 예측 CU에 대해 구현되는 일부 실시예들에서는, 대안적인 필터들이 재구성된 CU에 적용될 수 있거나 또는 어떠한 필터들도 CU에 적용될 수 없다.

[0078] 재구성된 화면들이 참조 화면들이면, 706에서 미래 CU들(102)의 인터 예측을 위해 이들이 참조 버퍼(738)에 저장될 수 있다.

[0079] 위 단계들 동안, JVET는 콘텐츠 적응성 클리핑 동작들이 하위 및 상위 클리핑 경계들 사이에 어울리도록 컬러 값들을 조정하는데 사용되는 것을 허용한다. 이러한 클리핑 경계들이 각각의 슬라이스에 대해 변경될 수 있고, 이러한 경계들을 식별하는 파라미터들이 비트스트림에서 시그널링될 수 있다.

[0080] 도 9는 JVET 디코더에서의 CU 코딩을 위한 간략화된 블록도를 묘사한다. JVET 디코더는 인코딩된 CU들(102)에 관한 정보를 포함하는 비트스트림을 수신할 수 있다. 이러한 비트스트림은 화면의 CU들(102)이 QTBT 구조에 따라 CTU(100)로부터 어떻게 파티셔닝되었는지를 표시할 수 있다. 비-제한적인 예로서, 비트스트림은 CU(102)가 쿼드트리 파티셔닝, 대칭 바이너리 파티셔닝, 및/또는 비대칭 바이너리 파티셔닝을 사용하여 QTBT에서 각각의 CTU(100)로부터 어떻게 파티셔닝되었는지를 식별할 수 있다. 비트스트림은 인트라 예측 모드들 또는 모션 벡터들과 같은 CU들(102)에 대한 예측 정보, 및 엔트로피 인코딩된 잔여 CU들을 표현하는 비트들(902)을 또한 표시할 수 있다.

[0081] 904에서 디코더는 인코더에 의해 비트스트림에서 시그널링되는 CABAC 컨텍스트 모델들을 사용하여 엔트로피 인코딩된 비트들(902)을 디코딩할 수 있다. 디코더는 컨텍스트 모델들의 확률들을 인코딩 동안 업데이트된 것과

동일한 방식으로 업데이트하는데 인코더에 의해 시그널링되는 파라미터들을 사용할 수 있다.

- [0082] 양자화된 변환 계수들(906)을 발견하기 위해 904에서의 엔트로피 인코딩을 반전시킨 후에, 역양자화된 변환 계수들(910)을 발견하기 위해 디코더가 908에서 역양자화될 수 있다. MDNSST 동작이 인코딩 동안 수행되었다면, 해당 동작은 역양자화 후에 디코더에 의해 반전될 수 있다.
- [0083] 912에서, 재구성된 잔여 CU(914)를 발견하기 위해 역양자화된 변환 계수들(910)이 역 변환될 수 있다. 916에서, 재구성된 CU(918)를 발견하기 위해, 922에서의 인트라 예측 또는 924에서의 인터 예측으로 발견되는 대응하는 예측 CU(926)에 재구성된 잔여 CU(914)가 추가될 수 있다.
- [0084] 따라서, 단계 923a에서, 가중 각도 예측이 단계 922에서 인트라 예측 모드로서 채택되었는지가 결정될 수 있다. 단계 923a에서, 인트라 예측 모드가 가중 각도 예측을 사용하는 것으로 결정되면, 예측 코딩 유닛(926)은 엔트로피 코딩 부재 필터링을 위해 전달될 수 있다. 그러나, 단계 923a에서, 인트라 예측 모드가 가중 각도 예측 이외인 것으로 결정되면, 엔트로피 코딩을 위한 전달 이전에 예측 코딩 유닛에, PDPC 필터링과 같은, 사후 인트라 예측 필터링(923b)이 적용될 수 있다.
- [0085] 920에서, 화면 레벨 또는 CU 레벨로, 재구성된 데이터에 하나 이상의 필터가 적용될 수 있다. 예를 들어, 디코더는 더블록킹 필터, SAO(sample adaptive offset) 필터, 및/또는 ALF(adaptive loop filter)를 적용할 수 있다. 위에 설명된 바와 같이, 프레임의 객관적 및 주관적 품질을 증가시키기 위해 최적의 필터 파라미터들을 추정하는데 인코더의 디코딩 루프에 위치되는 인-루프 필터들이 사용될 수 있다. 인코더에서 필터링된 재구성된 프레임을 매칭시키기 위해 920에서 재구성된 프레임을 필터링하도록 이러한 파라미터들 디코더에 송신된다.
- [0086] 재구성된 CU들(918)을 발견하고 시그널링된 필터들을 적용하는 것에 의해 재구성된 화면들이 생성된 후에, 디코더는 재구성된 화면들을 출력 비디오(928)로서 출력할 수 있다. 재구성된 화면들이 참조 화면들로서 사용될 것 이면, 이들은 924에서의 미래 CU들(102)의 인터 예측을 위해 참조 버퍼(930)에 저장될 수 있다.
- [0087] 도 10은 JVET 디코더에서의 CU 코딩(1000)의 방법의 실시예를 묘사한다. 도 10에 묘사되는 실시예에서, 단계 1002에서는 인코딩된 비트스트림(902)이 수신될 수 있고 다음으로 단계 1004에서는 인코딩된 비트스트림(902)과 연관된 CABAC 컨텍스트 모델이 결정될 수 있고, 다음으로 단계 1006에서는 결정된 CABAC 컨텍스트 모델을 사용하여 인코딩된 비트스트림(902)이 디코딩될 수 있다.
- [0088] 단계 1008에서는, 인코딩된 비트스트림(902)과 연관된 양자화된 변환 계수들(906)이 결정될 수 있고 다음으로 단계 1010에서는 양자화된 변환 계수들(906)로부터 역양자화된 변환 계수들(910)이 결정될 수 있다.
- [0089] 단계 1012에서는, 인코딩 동안 MDNSST 동작이 수행되었는지 및/또는 MDNSST 동작이 비트스트림(902)에 적용되었다는 표시를 비트스트림(902)이 포함하는지가 결정될 수 있다. 인코딩 프로세스 동안 MDNSST 동작이 수행되었다고 또는 MDNSST 동작이 비트스트림(902)에 적용되었다는 표시를 비트스트림(902)이 포함한다고 결정되면, 단계 1016에서 비트스트림(902)에 대해 역 변환 동작(912)이 수행되기 전에 역 MDNSST 동작(1014)이 구현될 수 있다. 대안적으로, 단계 1014에서의 역 MDNSST 동작의 적용이 없으면 단계 1016에서 비트스트림(902)에 대해 역 변환 동작(912)이 수행될 수 있다. 단계 1016에서의 역변환 동작(912)은 재구성된 잔여 CU(914)를 결정 및/또는 구성할 수 있다.
- [0090] 단계 1018에서는, 단계 1016으로부터의 재구성된 잔여 CU(914)가 예측 CU(918)와 조합될 수 있다. 예측 CU(918)는 단계 1020에서 결정되는 인트라-예측 CU(922) 및 단계 1022에서 결정되는 인터 예측 유닛(924) 중 하나일 수 있다.
- [0091] 따라서, 단계 1023a에서, 가중 각도 예측이 단계 1020에서 인트라 예측 모드로서 채택되었는지가 결정될 수 있다. 단계 1023a에서, 인트라 예측 모드가 가중 각도 예측을 사용하는 것으로 결정되면, 예측 코딩 유닛(926)은 엔트로피 코딩 부재 필터링을 위해 전달될 수 있고 및/또는 단계 1024에서 수행되는 필터링은 수정될 수 있고 및/또는 부재일 수 있다. 그러나, 단계 1023a에서, 인트라 예측 모드가 가중 각도 예측 이외인 것으로 결정되면, 엔트로피 코딩을 위한 전달 이전에 예측 코딩 유닛에, PDPC 필터링과 같은, 사후 인트라 예측 필터링 (1023b) 및/또는 단계 1024에서의 것이 적용될 수 있다.
- [0092] 도 10에 묘사되는 바와 같이, 일부 실시예들에서는 단계 1023b가 부재일 수 있고, 모든 예측들에 대해 단계 1018 후에 사후 인트라 예측 필터(1024)가 채택될 수 있다. 도 10에 묘사되는 이러한 실시예들에서, 인트라 예측 모드가 가중 각도 예측 이외의 것에 기초하면, 적용되는 필터는 단계 1024에서 정상적으로 적용될 것으로서 적용될 수 있다. 그러나, 인트라 예측 모드가 가중 각도 예측에 기초하면, 단계 1024에서의 필터링이 바이패스

될 수 있고 및/또는 일부 실시예들에서는, 적용되는 필터가 단계 1026에서의 재구성된 CU의 출력 이전에 메인 참조, 사이드 참조 또는 메인 및 사이드 참조들을 향해 바이어스되지 않을 수 있다. 비-제한적인 예로서, Cv1 및 Ch1의 값들은 동일할 수 있고 및/또는 Cv2 및 Ch2의 값들은 동일할 수 있다.

- [0093] 단계 1024에서는, 임의의 하나 이상의 필터(920)가 재구성된 CU(914)에 적용되어 단계 1026에서 출력될 수 있다. 일부 실시예들에서는, 필터들(920)이 단계 1024에서 적용되지 않을 수 있다.
- [0094] 일부 실시예들에서는, 단계 1028에서, 재구성된 CU(918)가 참조 버퍼(930)에 저장될 수 있다.
- [0095] 도 11은 JVET 인코더에서의 CU 코딩을 위한 간략화된 블록도(1100)를 묘사한다. 단계 1102에서는 JVET 코딩 트리 유닛이 QTBT(quadtrees plus binary tree) 구조에서의 루트 노드로서 표현될 수 있다. 일부 실시예들에서, QTBT는 루트 노드로부터 분기하는 쿼드트리 및/또는 쿼드트리의 리프 노드들 중 하나 이상으로부터 분기하는 바이너리 트리들을 가질 수 있다. 단계 1102로부터의 표현은 단계 1104, 1106 또는 1108로 진행할 수 있다.
- [0096] 단계 1104에서는, 표현된 쿼드트리 노드를 동일하지 않은 크기의 2개의 블록으로 분열시키는데 비대칭 바이너리 파티셔닝이 채택될 수 있다. 일부 실시예들에서, 분열된 블록들은 최종 코딩 유닛들을 표현할 수 있는 리프 노드들로서 쿼드트리 노드로부터 분기하는 바이너리 트리에서 표현될 수 있다. 일부 실시예들에서, 리프 노드들로서 쿼드트리 노드로부터 분기하는 바이너리 트리는 추가의 분열이 허용되지 않는 최종 코딩 유닛들을 표현한다. 일부 실시예들에서, 비대칭 파티셔닝은 코딩 유닛을 동일하지 않은 크기의 블록들로 분열시킬 수 있고, 첫 번째는 쿼드트리 노드의 25%를 표현하고, 두 번째는 쿼드트리 노드의 75%를 표현한다.
- [0097] 단계 1106에서는, 표현된 쿼드트리 노드를 동일한 크기의 4개의 정사각형 블록들로 분열시키는데 쿼드트리 파티셔닝이 채택될 수 있다. 일부 실시예들에서 분열된 블록들은 최종 코딩 유닛들을 표현하는 쿼드트리 노드들로서 표현될 수 있거나 또는 쿼드트리 파티셔닝, 대칭 바이너리 파티셔닝 또는 비대칭 바이너리 파티셔닝으로 다시 분열될 수 있는 자식 노드들로서 표현될 수 있다.
- [0098] 단계 1108에서는 표현된 쿼드트리 노드를 동일 크기의 2개의 블록으로 분열시키는데 쿼드트리 파티셔닝이 채택될 수 있다. 일부 실시예들에서 분열된 블록들은 최종 코딩 유닛들을 표현하는 쿼드트리 노드들로서 표현될 수 있거나 또는 쿼드트리 파티셔닝, 대칭 바이너리 파티셔닝 또는 비대칭 바이너리 파티셔닝으로 다시 분열될 수 있는 자식 노드들로서 표현될 수 있다.
- [0099] 단계 1110에서는, 인코딩되도록 구성되는 자식 노드들로서 단계 1106 또는 단계 1108로부터의 자식 노드들이 표현될 수 있다. 일부 실시예들에서는 JVET로 바이너리 트리의 리프 노드들에 의해 자식 노드들이 표현될 수 있다.
- [0100] 단계 1112에서는, 단계 1104 또는 1110으로부터의 코딩 유닛들이 JVET를 사용하여 인코딩될 수 있다.
- [0101] 도 12는 JVET 디코더에서의 CU 디코딩을 위한 간략화된 블록도(1200)를 묘사한다. 도 12에 묘사되는 실시예에서, 단계 1202에서는 코딩 트리 유닛이 QTBT 구조에 따라 어떻게 코딩 유닛들로 파티셔닝되었는지를 표시하는 비트스트림이 수신될 수 있다. 이러한 비트스트림은 쿼드트리 노드들이 쿼드트리 파티셔닝, 대칭 바이너리 파티셔닝 또는 비대칭 바이너리 파티셔닝 중 적어도 하나로 어떻게 분열되는지를 표시할 수 있다.
- [0102] 단계 1204에서는, QTBT 구조의 리프 노드들에 의해 표현되는, 코딩 유닛들이 식별될 수 있다. 일부 실시예들에서, 이러한 코딩 유닛들은 노드가 비대칭 바이너리 파티셔닝을 사용하여 어떻게 쿼드트리 리프 노드로부터 분열되었는지를 표시할 수 있다. 일부 실시예들에서, 코딩 유닛은 디코딩될 최종 코딩 유닛을 노드가 표현한다는 점을 표시할 수 있다.
- [0103] 단계 1206에서는, 식별된 코딩 유닛(들)이 JVET를 사용하여 디코딩될 수 있다.
- [0104] 도 13은 증가된 효율 코딩 시스템 및 방법의 간략화된 블록도(1300)를 묘사한다. 코딩 및 디코딩 시스템들에서, 코딩 블록과 그 이웃들 사이의 상관을 활용하기 위해 인트라 코딩에서 예측기가 생성된다. JVET에서, 코딩 블록의 상단 경계에 인접하는 참조 행 및 좌측 경계에 인접하는 참조 열이 예측기 생성 프로세스에서 사용된다. 각각의 인트라 예측 모드에 대해, 결정된 인트라 모드와 연관된 각도 방향을 사용하여 PU 내의 각각의 픽셀에 대한 참조 라인을 따르는 투영 이웃 위치가 결정된다. 참조 열을 따르는 투영 이웃들은 수평 모드들(모드들 2-33)에 대한 메인 참조 라인으로서 역할을 하고, 참조 행을 따르는 투영 이웃들은 수직 모드들(모드들 35-66)에 대한 메인 참조 라인으로서 역할을 한다. 예측기 생성에서 부분적으로 사용되는 참조 열 또는 행은 사이드 참조 라인이라고 불리운다. 도 8에 도시되는 바와 같이, 인트라 예측 모드 2 및 66은 동일한 예측 각도를 공유한다. 그러나, 모드 2는 좌측 이웃을 참조로서 사용하고, 한편 모드 66은 상단 이웃을 참조로서 사

용한다. 따라서, 하나의 코드워드가 이러한 2개의 모드들을 시그널링하여 오버헤드 비트들의 감소를 초래하도록 이러한 2개의 모드들(2 및 66)을 함께 조합하는 것에 의해 개선된 코딩 효율이 달성될 수 있다.

- [0105] 단계 1302에서, 코딩 예측 모드가 결정되고, 다음으로 단계 1304에서 코딩 모드가 모드 2인지 또는 모드 66인지에 관한 결정이 이루어진다. 결정된 코딩 예측 모드가 모드 2 또는 모드 66 이외이면, 임의의 알려진, 편리한 및/또는 원하는 코딩 예측 기술이 채택될 수 있다. 그러나, 코딩 모드 예측 모드 2 또는 66이 결정되면, 수정된 그리고 더 효율적인 예측 코딩이 채택될 수 있다.
- [0106] 하나의 코딩 모드를 사용하여, 2개의 인트라 예측들; 모드 2 및 66을 조합하는 인트라 예측 모드가 개시된다. 방법(1300)은 2개의 인트라 예측 모드들, 2 및 66의 예측 정확도를 유지하고, 한편 인코더 및 디코더 양자 모두에서 예측 방향을 선택함에 있어서 부담을 상당히 증가시키지 않는다. 따라서, 새로운 모드는, 그 예측 방향이 더 정확한 예측기를 제공할 때, 다른 것 대신에, 하나의 모드의 예측기를 뒤따르도록 그 예측기를 적응적으로 설정할 수 있고, 그 반대도 가능하다. 일부 실시예들에서, 하나의 휴리스틱 접근법은 디코더 측에서 이용가능 코딩 정보를 사용하여 2개의 모드들(2 및 66) 사이에서 선택하는 것이다. 새로운 조합 모드에 대한 예측 방향을 결정하는데 다양한 정보가 사용될 수 있다. 일부 실시예들에서, 폭 또는 높이와 같은, 블록 치수가 선택 기준으로서 사용될 수 있다. 이러한 실시예들에서, 예측 방향은 더 긴 경계를 갖는 방향을 뒤따르도록 선택될 수 있다. 그러나, 대안적인 실시예들에서, 더 짧은 경계를 갖는 예측 방향이 선택될 수 있다.
- [0107] 비-제한적인 예로서, 선택 기준으로서의 블록 치수 및 예측 모드 2 및 66을 사용하여, 좌표(x,y)에서의 가중 각도 예측의 예측기 픽셀, P(x,y)는 다음과 같이 계산될 수 있다:
- [0108] 폭 > 높이일 때,  $P[x,y] = \text{Recon}[x+y+2,-1]$ ; 또는
- [0109] 대안적인 조건들에 대해,  $P[x,y] = \text{Recon}[-1,x+y+2]$
- [0110] 여기서 Recon[0,0]은 현재 CU의 상단 좌측 좌표(0,0)에서의 재구성된 픽셀이다.
- [0111] 대안적인, 비-제한적 예로서, 참조 행을 따르는 픽셀 차이(예를 들어, 분산) 및 참조 열을 따르는 픽셀 차이가 사용될 수 있다. 이러한 실시예들에서, 예측 방향은 더 작은(또는 더 큰) 픽셀 차이를 갖는 해당 방향을 뒤따르도록 이루어질 수 있다.
- [0112] 일부 실시예들에서, 가중 각도 예측은 상단 참조 행 및 좌측 참조 열 양자 모두 상의 투영 위치에서의 픽셀들을 사용하여 각도 예측을 위한 예측기 픽셀들을 생성할 수 있다. JVET 모드 2 또는 모드 66에 대해, 좌표(x,y)에서의 가중 각도 예측의 예측기 픽셀, P(x,y)는 다음과 같이 계산될 수 있다:
- [0113]  $P[x,y] = (((x+1)*\text{Recon}[x+y+2,-1]) + ((y+1)*(\text{Recon}[-1,x+y+2]))) + (y+x+2)/2 / (y+x+2)$
- [0114] 여기서 Recon[0,0]은 현재 CU의 상단 좌측 좌표(0,0)에서의 재구성된 픽셀이다.
- [0115] 이러한 시스템 및 방법은, 가중 각도 예측을 위해 사용되지 않은, 모드 2 또는 모드 66의 모드 인덱스를 배정하는 것에 의해 가중 각도 예측을 지원하도록 확장될 수 있다. 즉, 모드 2가 가중 각도 예측에 배정되면, 모드 66은 임의의 다른 알려진, 편리한 및/또는 원하는 예측 방법에 배정될 수 있다. 일부 실시예들에서는 그 반대가 참일 수 있으며, 모드 66은 가중 각도 예측에 배정되고, 모드 2는 임의의 다른 알려진, 편리한 및/또는 원하는 예측 방법에 배정될 수 있다.
- [0116] 도 14는 도 7a 및 도 7b에 묘사되고 설명되는 것과 실질적으로 유사한 JVET 인코더에서의 증가된 효율이 있는 CU 코딩을 위한 간략화된 블록도를 묘사한다. 도 14는 단계들 1402 1404 및 단계 1406을 추가로 포함하는 시스템 및 방법을 묘사하며, 단계 1402에서는 인트라 예측 모드들 2 또는 66이 채택되는지에 관한 결정이 이루어진다. 가중 또는 비-가중 각도 예측을 위해 도 13에 관련하여 위에 그리고 가중 또는 비-가중 각도 예측이 결정되는지에 관한 결정이 단계 705a에서 결정된 후 단계 1406에서 설명되는 바와 같이, 다음으로 단계 1404에서는 표준/알려진 및/또는 편리한 예측 코딩이 채택될 수 있고, 단계 1406에서는, 예측 모드들에 대해 선택된 수정된 예측 코딩이 구현될 수 있다. 즉, 새로운 모드는, 그 예측 방향이 더 정확한 예측기를 제공할 때, 다른 것 대신에, 하나의 모드의 예측기를 뒤따르도록 그 예측기를 적응적으로 설정할 수 있고, 그 반대도 가능하다. 일부 실시예들에서, 하나의 휴리스틱 접근법은 디코더 측에서 이용가능 코딩 정보를 사용하여 2개의 모드들(2 및 66) 사이에서 선택하는 것이다. 새로운 조합 모드에 대한 예측 방향을 결정하는데 다양한 정보가 사용될 수 있다. 일부 실시예들에서, 폭 또는 높이와 같은, 블록 치수가 선택 기준으로서 사용될 수 있다. 이러한 실시예들에서, 예측 방향은 더 긴 경계를 갖는 방향을 뒤따르도록 선택될 수 있다. 그러나, 대안적인 실시예들

에서, 더 짧은 경계를 갖는 예측 방향이 선택될 수 있다.

- [0117] 대안적인 실시예들에서, (도 7a 및 도 7b에 도시되는) 단계 705b의 사후 필터링이 도 7a 및 도 7b에 관련하여 묘사되고 설명되는 시스템 및 방법 내에서 동시에 구현될 수 있다는 점이 해당 분야에서의 통상의 기술자들에게 용이하게 명백할 것이다.
- [0118] 도 15는 JVET 디코더에서의 증가된 효율이 있는 CU 디코딩을 위한 간략화된 블록도를 묘사한다. 도 15는 단계들 1402 1404 및 단계 1406을 추가로 포함하는 시스템 및 방법을 묘사하며, 단계 1402에서는 인트라 예측 모드들 2 또는 66이 채택되는지에 관한 결정이 이루어진다. 가중 또는 비-가중 각도 예측을 위해 도 13에 관련하여 위에 그리고 가중 또는 비-가중 각도 예측이 결정되는지에 관한 결정이 단계 923a에서 결정된 후 단계 1406에서 설명되는 바와 같이, 다음으로 단계 1404에서는 표준/알려진 및/또는 편리한 예측 코딩이 채택될 수 있고, 단계 1406에서는, 예측 모드들에 대해 선택된 수정된 예측 코딩이 구현될 수 있다.
- [0119] 대안적인 실시예들에서, 단계 923b의 사후 필터링이 도 9에 관련하여 묘사되고 설명되는 시스템 및 방법 내에서 동시에 구현될 수 있다는 점이 해당 분야에서의 통상의 기술자들에게 용이하게 명백할 것이다.
- [0120] 실시예들을 실시하는데 요구되는 명령어들의 시퀀스들의 실행이 도 16에 도시되는 바와 같은 컴퓨터 시스템(1600)에 의해 수행될 수 있다. 실시예에서, 명령어들의 시퀀스들의 실행은 단일 컴퓨터 시스템(1600)에 의해 수행된다. 다른 실시예들에 따르면, 통신 링크(1615)에 의해 연결되는 2개 이상의 컴퓨터 시스템들(1600)이 서로 협력하여 명령어들의 시퀀스를 수행할 수 있다. 단 하나의 컴퓨터 시스템(1600)의 설명이 아래에 제시되더라도, 그러나, 임의의 수의 컴퓨터 시스템들(1600)이 실시예들을 실시하는데 채택될 수 있다는 점이 이해되어야 한다.
- [0121] 컴퓨터 시스템(1300)의 기능 컴포넌트들의 블록도인, 도 16을 참조하여 실시예에 따른 컴퓨터 시스템(1600)이 이제 설명될 것이다. 본 명세서에 사용되는 바와 같이, 컴퓨터 시스템(1600)이라는 용어는 하나 이상의 프로그램을 저장하고 독립적으로 실행할 수 있는 임의의 컴퓨팅 디바이스를 설명하는데 광범위하게 사용된다.
- [0122] 각각의 컴퓨터 시스템(1600)은 버스(1606)에 연결되는 통신 인터페이스(1614)를 포함할 수 있다. 통신 인터페이스(1614)는 컴퓨터 시스템들(1600) 사이의 양방향 통신을 제공한다. 각각의 컴퓨터 시스템(1600)의 통신 인터페이스(1614)는 다양한 타입들의 신호 정보, 예를 들어, 명령어들, 메시지들 및 데이터를 표현하는 데이터 스트림들을 포함하는 전기, 전자기 또는 광 신호들을 송신하고 수신한다. 통신 링크(1615)는 하나의 컴퓨터 시스템(1600)을 다른 컴퓨터 시스템(1600)과 링크한다. 예를 들어, 통신 링크(1615)는 LAN일 수 있으며, 이러한 경우 통신 인터페이스(1614)는 LAN 카드일 수 있거나, 또는 통신 링크(1615)는 PSTN일 수 있으며, 이러한 경우 통신 인터페이스(1614)는 ISDN(integrated services digital network) 카드 또는 모뎀일 수 있거나, 또는 통신 링크(1615)는 인터넷일 수 있으며, 이러한 경우 통신 인터페이스(1614)는 다이얼-업, 케이블 또는 무선 모뎀일 수 있다.
- [0123] 컴퓨터 시스템(1600)은, 프로그램, 즉, 애플리케이션, 코드를 포함하는, 메시지들, 데이터, 및 명령어들을, 자신의 각각의 통신 링크(1615) 및 통신 인터페이스(1614)를 통해 송신하고 수신할 수 있다. 수신된 프로그램 코드는 그것이 수신됨에 따라 각각의 프로세서(들)(1607)에 의해 실행되고, 및/또는, 차후 실행을 위해, 저장 디바이스(1610), 또는 다른 연관된 비-휘발성 매체에 저장될 수 있다.
- [0124] 실시예에서, 컴퓨터 시스템(1600)은 데이터 저장 시스템(1631), 예를 들어, 컴퓨터 시스템(1600)에 의해 용이하게 액세스가능한 데이터베이스(1632)를 포함하는 데이터 저장 시스템(1631)과 함께 동작한다. 컴퓨터 시스템(1600)은 데이터 인터페이스(1633)를 통해 데이터 저장 시스템(1631)과 통신한다. 버스(1606)에 연결되는, 데이터 인터페이스(1633)는, 다양한 타입들의 신호 정보, 예를 들어, 명령어들, 메시지들 및 데이터를 표현하는 데이터 스트림들을 포함하는, 전기, 전자기 또는 광 신호들을 송신하고 수신한다. 실시예들에서, 데이터 인터페이스(1633)의 기능들은 통신 인터페이스(1614)에 의해 수행될 수 있다.
- [0125] 컴퓨터 시스템(1600)은, 명령어들, 메시지들 및 데이터, 집합적으로는, 정보를 통신하기 위한 버스(1606) 또는 다른 통신 메커니즘, 및 정보를 처리하기 위해 버스(1606)와 연결되는 하나 이상의 프로세서(1607)를 포함한다. 컴퓨터 시스템(1600)은, 프로세서(들)(1607)에 의해 실행될 동적 데이터 및 명령어들을 저장하기 위해 버스(1606)에 연결되는, RAM(random access memory) 또는 다른 동적 저장 디바이스와 같은, 메인 메모리(1608)를 또한 포함한다. 메인 메모리(1608)는 프로세서(들)(1607)에 의한 명령어들의 실행 동안 임시 데이터, 즉, 변수들, 또는 다른 중간 정보를 저장하기 위해 또한 사용될 수 있다.
- [0126] 컴퓨터 시스템(1600)은 프로세서(들)(1607)에 대한 정적 데이터 및 명령어들을 저장하기 위해 버스(1606)에 연

결되는 ROM(read only memory)(1609) 또는 다른 정적 저장 디바이스를 추가로 포함할 수 있다. 프로세서(들)(1607)에 대한 데이터 및 명령어를 저장하기 위해 자기 디스크 또는 광 디스크와 같은, 저장 디바이스(1610)가 또한 제공되고 버스(1606)에 연결될 수 있다.

[0127] 사용자에게 정보를 디스플레이하기 위해, 이에 제한되는 것은 아니지만, CRT(cathode ray tube) 또는 LCD(liquid-crystal display) 모니터와 같은, 디스플레이 디바이스(1611)에 버스(1606)를 통해 컴퓨터 시스템(1600)이 연결될 수 있다. 프로세서(들)(1607)에 정보 및 커맨드 선택들을 통신하기 위해 버스(1606)에 입력 디바이스(1612), 예를 들어, 영숫자 및 다른 키들이 연결된다.

[0128] 하나의 실시예에 따르면, 개별 컴퓨터 시스템(1600)은 메인 메모리(1608)에 포함되는 하나 이상의 명령어들의 하나 이상의 시퀀스를 실행하는 그들 각각의 프로세서(들)(1607)에 의해 구체적인 동작들을 수행한다. 이러한 명령어들은, ROM(1609) 또는 저장 디바이스(1610)와 같은, 다른 컴퓨터-사용가능 매체로부터 메인 메모리(1608) 내로 판독될 수 있다. 메인 메모리(1608)에 포함되는 명령어들의 시퀀스들의 실행은 프로세서(들)(1607)로 하여금 본 명세서에 설명되는 프로세스들을 수행하게 한다. 대안적인 실시예들에서는, 소프트웨어 명령어들 대신에 또는 이들과 조합하여 하드-와이어드 회로가 사용될 수 있다. 따라서, 실시예들이 하드웨어 회로 및/또는 소프트웨어의 임의의 구체적인 조합에 제한되는 것은 아니다.

[0129] "컴퓨터-사용가능 매체(computer-usable medium)"라는 용어는, 본 명세서에서 사용되는 바와 같이, 정보를 제공하거나 또는 프로세서(들)(1607)에 의해 사용가능한 임의의 매체를 지칭한다. 이러한 매체는, 이에 제한되는 것은 아니지만, 비-휘발성, 휘발성 및 송신 매체를 포함하는, 많은 형태들을 취할 수 있다. 비-휘발성 매체, 즉, 전력의 부재 시에 정보를 보유할 수 있는 매체는, ROM(1309), CD ROM, 자기 테이프, 및 자기 디스크들을 포함한다. 휘발성 매체, 즉, 전력의 부재 시에 정보를 보유할 수 없는 매체는 메인 메모리(1608)를 포함한다. 송신 매체는, 버스(1606)를 구성하는 와이어들을 포함하는, 동축 케이블들, 구리 와이어 및 광 섬유들을 포함한다. 송신 매체는 반송파들의 형태를 또한 취할 수 있다; 즉, 정보 신호들을 송신하도록, 주파수, 진폭 또는 위상에서와 같이, 변조될 수 있는 전자기파들. 추가적으로, 송신 매체는, 전파 및 적외선 데이터 통신들 동안 생성되는 것들과 같은, 음향 또는 광 파들의 형태를 취할 수 있다.

[0130] 진술한 명세서에서는, 실시예들이 그 구체적인 엘리먼트들을 참조하여 설명되었다. 그러나, 실시예들의 더 넓은 사상 및 범위로부터 벗어나지 않고 다양한 수정들 및 변경들이 이루어질 수 있다는 점이 명백할 것이다. 예를 들어, 독자는 본 명세서에 설명되는 프로세스 흐름도들에 도시되는 프로세스 액션들의 구체적인 순서화 및 조합이 단지 예시적이라는 점, 및 상이한 또는 추가적인 프로세스 액션들, 또는 프로세스 액션들의 상이한 조합 또는 순서화를 사용하는 것이 실시예들을 행하기 위해 사용될 수 있다는 점을 이해해야 한다. 따라서, 본 명세서 및 도면들은 한정적인 것 보다는 오히려 예시적인 의미로 고려되어야 한다.

[0131] 본 발명이 다양한 컴퓨터 시스템들에서 구현될 수 있다는 점이 또한 주목되어야 한다. 본 명세서에 설명되는 다양한 기술들은 하드웨어 또는 소프트웨어, 또는 양자 모두의 조합으로 구현될 수 있다. 바람직하게는, 이러한 기술들이, 프로세서, 프로세서에 의해 판독가능한 저장 매체(휘발성 및 비-휘발성 메모리 및/또는 저장 엘리먼트들을 포함함), 적어도 하나의 입력 디바이스, 및 적어도 하나의 출력 디바이스를 각각 포함하는 프로그램가능 컴퓨터들 상에서 실행되는 컴퓨터 프로그램들에서 구현된다. 위에 설명된 기능들을 수행하고 출력 정보를 생성하기 위해 입력 디바이스를 사용하여 들어오는 데이터에 프로그램 코드가 적용된다. 이러한 출력 정보는 하나 이상의 출력 디바이스에 적용된다. 각각의 프로그램이 바람직하게는 컴퓨터 시스템과 통신하도록 하이 레벨 프로시저형 또는 객체 지향 프로그래밍 언어로 구현된다. 그러나, 이러한 프로그램들은, 원하면, 어셈블리 또는 기계 언어로 구현될 수 있다. 임의의 경우에, 이러한 언어는 컴파일형 또는 해석형 언어일 수 있다. 각각의 이러한 컴퓨터 프로그램이 바람직하게는 위에 설명된 프로시저들을 수행하기 위해 저장 매체 또는 디바이스가 컴퓨터에 의해 판독될 때 컴퓨터를 구성하고 동작시키기 위해 범용 또는 특수 목적 프로그램가능 컴퓨터에 의해 판독가능한 저장 매체 또는 디바이스(예를 들어, ROM 또는 자기 디스크) 상에 저장된다. 이러한 시스템은, 컴퓨터 프로그램으로 구성되는, 컴퓨터-판독가능 저장 매체로서 구현되는 것으로 또한 고려될 수 있으며, 그렇게 구성되는 저장 매체는 컴퓨터로 하여금 구체적인 그리고 미리 정의된 방식으로 동작하게 한다. 추가로, 예시적인 컴퓨팅 애플리케이션들의 저장 엘리먼트들은 다양한 조합들 및 구성들로 데이터를 저장할 수 있는 관계형 또는 시퀀스형(플랫 파일) 타입 컴퓨팅 데이터베이스들일 수 있다.

[0132] 도 17은 본 명세서에 설명되는 시스템들 및 디바이스들의 특징들을 포함할 수 있는 소스 디바이스(1712) 및 목적지 디바이스(1710)의 하이 레벨 뷰이다. 도 17에 도시되는 바와 같이, 예시적인 비디오 코딩 시스템(1710)은 소스 디바이스(1712) 및 목적지 디바이스(1714)를 포함하며, 이러한 예에서, 소스 디바이스(1712)는 인코딩된

비디오 데이터를 생성한다. 따라서, 소스 디바이스(1712)는 비디오 인코딩 디바이스라고 지칭될 수 있다. 목적지 디바이스(1714)는 소스 디바이스(1712)에 의해 생성되는 인코딩된 비디오 데이터를 디코딩할 수 있다. 따라서, 목적지 디바이스(1714)는 비디오 디코딩 디바이스라고 지칭될 수 있다. 소스 디바이스(1712) 및 목적지 디바이스(1714)는 비디오 코딩 디바이스들의 예들일 수 있다.

[0133] 목적지 디바이스(1714)는 채널(1716)을 통해 소스 디바이스(1712)로부터 인코딩된 비디오 데이터를 수신할 수 있다. 채널(1716)은 인코딩된 비디오 데이터를 소스 디바이스(1712)로부터 목적지 디바이스(1714)로 이동시킬 수 있는 매체 또는 디바이스의 타입을 포함할 수 있다. 하나의 예에서, 채널(1716)은 소스 디바이스(1712)로 하여금 인코딩된 비디오 데이터를 실시간으로 목적지 디바이스(1714)에 직접 송신할 수 있게 하는 통신 매체를 포함할 수 있다.

[0134] 이러한 예에서, 소스 디바이스(1712)는 인코딩된 비디오 데이터를, 무선 통신 프로토콜과 같은, 통신 표준에 따라 변조할 수 있고, 변조된 비디오 데이터를 목적지 디바이스(1714)에 송신할 수 있다. 통신 매체는, RF(radio frequency) 스펙트럼 또는 하나 이상의 물리적 송신 라인과 같은, 무선 또는 유선 통신 매체를 포함할 수 있다. 통신 매체는, 로컬 영역 네트워크, 광역 네트워크, 또는 인터넷과 같은 글로벌 네트워크와 같이, 패킷-기반 네트워크의 부분을 형성할 수 있다. 통신 매체는 라우터들, 스위치들, 기지국들, 또는 소스 디바이스(1712)로부터 목적지 디바이스(1714)로의 통신을 용이하게 하는 다른 장비를 포함할 수 있다. 다른 예에서, 채널(1716)은 소스 디바이스(1712)에 의해 생성되는 인코딩된 비디오 데이터를 저장하는 저장 매체에 대응할 수 있다.

[0135] 도 17의 예에서, 소스 디바이스(1712)는 비디오 소스(1718), 비디오 인코더(1720) 및 출력 인터페이스(1722)를 포함한다. 일부 경우들에서, 출력 인터페이스(1728)는 변조기/복조기(모뎀) 및/또는 송신기를 포함할 수 있다. 소스 디바이스(1712)에서, 비디오 소스(1718)는 비디오 캡처 디바이스, 예를 들어, 비디오 카메라, 이전에 캡처된 비디오 데이터를 포함하는 비디오 아카이브, 비디오 콘텐츠 제공자로부터 비디오 데이터를 수신하기 위한 비디오 피드 인터페이스, 및/또는 비디오 데이터를 생성하기 위한 컴퓨터 그래픽 시스템, 또는 이러한 소스들의 조합과 같은 소스를 포함할 수 있다.

[0136] 비디오 인코더(1720)는 캡처된, 사전-캡처된 또는 컴퓨터-생성된 비디오 데이터를 인코딩할 수 있다. 입력 이미지는 비디오 인코더(1720)에 의해 수신되어 입력 프레임 메모리(1721)에 저장될 수 있다. 범용 프로세서(1723)는 여기서부터 정보를 로딩하고 인코딩을 수행할 수 있다. 범용 프로세서를 구동하기 위한 프로그램이, 도 17에 묘사되는 예시적인 메모리 모듈들과 같은, 저장 디바이스로부터 로딩될 수 있다. 범용 프로세서는 인코딩을 수행하는데 처리 메모리(1722)를 사용할 수 있고, 일반 프로세서에 의한 인코딩 정보의 출력은, 출력 버퍼(1726)와 같은, 버퍼에 저장될 수 있다.

[0137] 비디오 인코더(1720)는 적어도 하나의 베이스 레이어 및 적어도 하나의 강화 레이어를 정의하는 스케일가능 비디오 코딩 스킴에서 비디오 데이터를 코딩(예를 들어, 인코딩)하도록 구성될 수 있는 리샘플링 모듈(1725)을 포함할 수 있다. 리샘플링 모듈(1725)은 인코딩 프로세스의 부분으로서 적어도 일부 비디오 데이터를 리샘플링할 수 있으며, 리샘플링은 리샘플링 필터들을 사용하여 적응성 방식으로 수행될 수 있다.

[0138] 인코딩된 비디오 데이터, 예를 들어, 코딩된 비트 스트림이, 소스 디바이스(1712)의 출력 인터페이스(1728)를 통해 목적지 디바이스(1714)에 직접 송신될 수 있다. 도 17의 예에서, 목적지 디바이스(1714)는 입력 인터페이스(1738), 비디오 디코더(1730), 및 디스플레이 디바이스(1732)를 포함한다. 일부 경우들에서, 입력 인터페이스(1728)는 수신기 및/또는 모뎀을 포함할 수 있다. 목적지 디바이스(1714)의 입력 인터페이스(1738)는 인코딩된 비디오 데이터를 채널(1716)을 통해 수신한다. 인코딩된 비디오 데이터는 비디오 데이터를 표현하는 비디오 인코더(1720)에 의해 생성되는 다양한 구문 엘리먼트들을 포함할 수 있다. 이러한 구문 엘리먼트들은 통신 매체 상에 송신되는 인코딩된 비디오 데이터와 함께 포함될 수 있거나, 저장 매체 상에 저장될 수 있거나, 또는 파일 서버에 저장될 수 있다.

[0139] 인코딩된 비디오 데이터는 디코딩 및/또는 재생을 위한 목적지 디바이스(1714)에 의한 차후 액세스를 위해 저장 매체 또는 파일 서버 상에 또한 저장될 수 있다. 예를 들어, 코딩된 비트스트림은 입력 버퍼(1731)에 임시로 저장되고, 다음으로 범용 프로세서(1733)에 로딩될 수 있다. 범용 프로세서를 구동하기 위한 프로그램이 저장 디바이스 또는 메모리로부터 로딩될 수 있다. 범용 프로세서는 디코딩을 수행하는데 프로세스 메모리(1732)를 사용할 수 있다. 비디오 디코더(1730)는 비디오 인코더(1720)에서 채택되는 리샘플링 모듈(1725)과 유사한 리샘플링 모듈(1735)을 또한 포함할 수 있다.

[0140] 도 17은 범용 프로세서(1733)와 별개로 리샘플링 모듈(1735)을 묘사하지만, 리샘플링 기능이 범용 프로세서에

의해 실행되는 프로그램에 의해 수행될 수 있고, 비디오 인코더에서의 처리가 하나 이상의 프로세서를 사용하여 달성될 수 있다는 점이 해당 분야에서의 기술자에 의해 인식될 것이다. 디코딩된 이미지(들)는 출력 프레임 버퍼(1736)에 저장되고 다음으로 입력 인터페이스(1738)에 송출될 수 있다.

[0141] 디스플레이 디바이스(1738)는 목적지 디바이스(1714)와 통합될 수 있거나 또는 그 외부에 있을 수 있다. 일부 예들에서, 목적지 디바이스(1714)는 통합 디스플레이 디바이스를 포함할 수 있고 외부 디스플레이 디바이스와 인터페이스하도록 또한 구성될 수 있다. 다른 예들에서, 목적지 디바이스(1714)는 디스플레이 디바이스일 수 있다. 일반적으로, 디스플레이 디바이스(1738)는 디코딩된 비디오 데이터를 사용자에게 디스플레이한다.

[0142] 비디오 인코더(1720) 및 비디오 디코더(1730)는 비디오 압축 표준에 따라 동작할 수 있다. ITU-T VCEG(Q6/16) 및 ISO/IEC MPEG(JTC1/SC29/WG11)은 현재 HEVC(High Efficiency Video Coding) 표준의 것을 상당히 초과하는 압축 능력이 있는 미래 비디오 코딩 기술의 표준화(스크린 콘텐츠 코딩 및 높은-동적-범위 코딩을 위한 자신의 현재 확장들 및 근방 확장들을 포함함)에 대한 잠재적 필요를 연구하고 있다. 이러한 그룹들은 이러한 영역에서 그들의 전문가들에 의해 제안되는 압축 기술 설계들을 평가하기 위해 JVET(Joint Video Exploration Team)로서 알려진 공동 협업 노력에서 이러한 탐사 활동에 대해 함께 작업한다. JVET 개발의 최근 캡처는, J. Chen, E. Alshina, G. Sullivan, J. Ohm, J. Boyce에 의해 저술된, "Algorithm Description of Joint Exploration Test Model 5(JEM 5)", JVET-E1001-V2에서 설명된다.

[0143] 추가적으로 또는 대안적으로, 비디오 인코더(1720) 및 비디오 디코더(1730)는 개시되는 JVET 특징들과 함께 기능하는 다른 독점적 또는 산업 표준들에 따라 동작할 수 있다. 따라서, ITU-T H.264 표준과 같은 다른 표준들이, MPEG-4, Part 10, AVC(Advanced Video Coding), 또는 이러한 표준들의 확장들이라고 대안적으로 지칭된다. 따라서, JVET를 위해 새롭게 개발되는 동안, 본 개시내용의 기술들이 임의의 특정 코딩 표준 또는 기술에 제한되는 것은 아니다. 비디오 압축 표준들 및 기술들의 다른 예들은 MPEG-2, ITU-T H.263 및 독점적 또는 오픈 소스 압축 포맷들 및 관련 포맷들을 포함한다.

[0144] 비디오 인코더(1720) 및 비디오 디코더(1730)는 하드웨어, 소프트웨어, 펌웨어 또는 이들의 임의의 조합으로 구현될 수 있다. 예를 들어, 비디오 인코더(1720) 및 디코더(1730)는 하나 이상의 프로세서, DSP(digital signal processors), ASIC(application specific integrated circuits), FPGA(field programmable gate arrays), 이산 로직, 또는 이들의 임의의 조합을 채택할 수 있다. 비디오 인코더(1720) 및 디코더(1730)가 부분적으로 소프트웨어로 구현될 때, 디바이스는 이러한 소프트웨어에 대한 명령어들을 적합한, 비-일시적 컴퓨터-판독가능 저장 매체에 저장할 수 있고, 본 개시내용의 기술들을 수행하는데 하나 이상의 프로세서를 사용하는 하드웨어로 이러한 명령어들을 실행할 수 있다. 비디오 인코더(1720) 및 비디오 디코더(1730) 각각은 하나 이상의 인코더 또는 디코더에 포함될 수 있고, 이들 중 어느 하나는 각각의 디바이스에서 조합된 CODEC(encoder/decoder)의 부분으로서 통합될 수 있다.

[0145] 본 명세서에 설명되는 주제의 양태들은, 위에 설명된 범용 프로세서들(1723 및 1733)과 같은, 컴퓨터에 의해 실행되는, 프로그램 모듈들과 같은, 컴퓨터-실행가능 명령어들의 일반적인 컨텍스트에서 설명될 수 있다. 일반적으로, 프로그램 모듈들은, 특정 태스크들을 수행하거나 또는 특정 추상 데이터 타입들을 구현하는, 루틴들, 프로그램들, 객체들, 컴포넌트들, 및 데이터 구조들 등을 포함한다. 본 명세서에 설명되는 주제의 양태들은 통신 네트워크를 통해 링크되는 원격 처리 디바이스들에 의해 태스크들이 수행되는 분산형 컴퓨팅 환경들에서 또한 실시될 수 있다. 분산형 컴퓨팅 환경에서, 프로그램 모듈들은 메모리 저장 디바이스들을 포함하는 로컬 및 원격 컴퓨터 저장 매체 양자 모두에 위치될 수 있다.

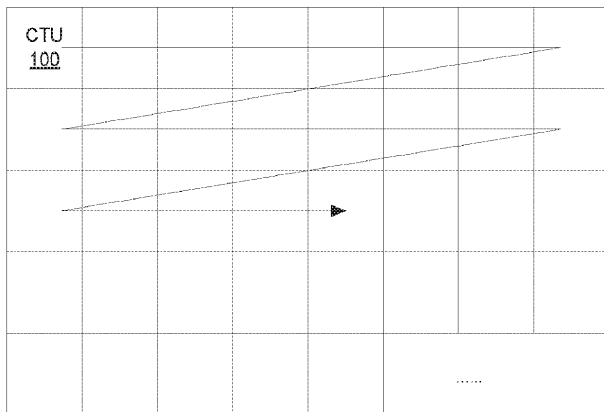
[0146] 메모리의 예들은 RAM(random access memory), ROM(read only memory), 또는 양자 모두를 포함한다. 메모리는, 위에 설명된 기술들을 수행하기 위해, 소스 코드 또는 바이너리 코드와 같은, 명령어들을 저장할 수 있다. 메모리는, 프로세서(1723 및 1733)와 같은, 프로세서에 의해 실행될 명령어들의 실행 동안 변수들 또는 다른 중간 정보를 저장하기 위해 또한 사용될 수 있다.

[0147] 저장 디바이스는, 위에 설명된 기술들을 수행하기 위해, 소스 코드 또는 바이너리 코드와 같은, 명령어들을 또한 저장할 수 있다. 저장 디바이스는 컴퓨터 프로세서에 의해 사용되고 조작되는 데이터를 추가적으로 저장할 수 있다. 예를 들어, 비디오 인코더(1720) 또는 비디오 디코더(1730)에서의 저장 디바이스는 컴퓨터 시스템(1723 또는 1733)에 의해 액세스되는 데이터베이스일 수 있다. 저장 디바이스의 다른 예들은 RAM(random access memory), ROM(read only memory), 하드 드라이브, 자기 디스크, 광 디스크, CD-ROM, DVD, 플래시 메모리, USB 메모리 카드, 또는 컴퓨터가 판독할 수 있는 임의의 다른 매체를 포함한다.

- [0148] 메모리 또는 저장 디바이스는 비디오 인코더 및/또는 디코더에 의해 또는 이와 관련하여 사용하기 위한 비-일시적 컴퓨터-관독가능 저장 매체의 예일 수 있다. 이러한 비-일시적 컴퓨터-관독가능 저장 매체는 특정 실시예들에 의해 설명되는 기능들을 수행하도록 구성되게 컴퓨터 시스템을 제어하기 위한 명령어들을 포함한다. 이러한 명령어들은, 하나 이상의 컴퓨터 프로세서에 의해 실행될 때, 특정 실시예들에서 설명되는 것을 수행하도록 구성될 수 있다.
- [0149] 또한, 일부 실시예들은 흐름도 또는 블록도로서 묘사될 수 있는 프로세스로서 설명되었다는 점이 주목된다. 각각이 이러한 동작들을 시퀀스형 프로세스로서 설명할 수 있더라도, 이러한 동작들 중 많은 것은 병렬로 또는 동시에 수행될 수 있다. 또한, 이러한 동작들의 순서가 재배열될 수 있다. 프로세스는 도면들에 포함되지 않은 추가적인 단계들을 가질 수 있다.
- [0150] 특정 실시예들은, 명령어 실행 시스템, 장치, 시스템, 또는 머신에 의해 또는 이와 관련하여 사용하기 위해 비-일시적 컴퓨터-관독가능 저장 매체에 구현될 수 있다. 이러한 컴퓨터-관독가능 저장 매체는, 특정 실시예들에 의해 설명되는 방법을 수행하도록 컴퓨터 시스템을 제어하기 위한 명령어들을 포함한다. 이러한 컴퓨터 시스템은 하나 이상의 컴퓨팅 디바이스를 포함할 수 있다. 이러한 명령어들은, 하나 이상의 컴퓨터 프로세서에 의해 실행될 때, 특정 실시예들에서 설명되는 것을 수행하도록 구성될 수 있다.
- [0151] 본 명세서에서의 설명에서 그리고 다음의 청구항들 전반적으로 사용되는 바와 같이, "a", "an", 및 "the"는 문맥이 명확하게 달리 구술하지 않는 한 복수의 참조를 포함한다. 또한, 본 명세서에서의 설명에서 그리고 다음의 청구항 전반적으로 사용되는 바와 같이, "in"의 의미는 문맥이 명확하게 달리 구술하지 않는 한 "in" 및 "on"을 포함한다.
- [0152] 본 발명의 예시적인 실시예들이 상세히 그리고 위 구조적 특징들 및/또는 방법론적 행동들에 구체적인 언어로 설명되었더라도, 해당 분야에서의 기술자는 본 발명의 신규한 교시들 및 이점들로부터 실질적으로 벗어나지 않고 예시적인 실시예들에서 많은 추가적인 수정들이 가능하다는 점을 용이하게 인식할 것이라는 점이 이해되어야 한다. 또한, 첨부된 청구항들에서 정의되는 주제가 반드시 위에 설명된 구체적인 특징들 또는 행동들로 제한되는 것은 아니라는 점이 이해되어야 한다. 따라서, 이들 및 모든 이러한 수정들은 첨부된 청구항들에 따르는 폭 및 범위에서 해석되는 본 발명의 범위 내에서 포함되도록 의도된다.

**도면**

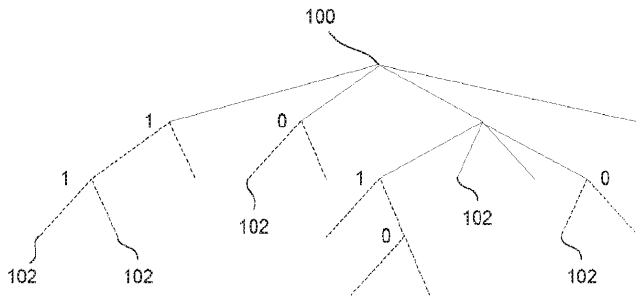
**도면1**



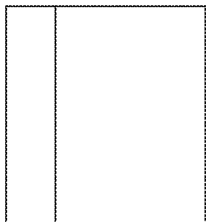
도면2

			CU 102
CU 102	CU 102	CU 102	CU 102
CU 102	CU 102	CU 102	CU 102
CU 102	CU 102	CU 102	CU 102

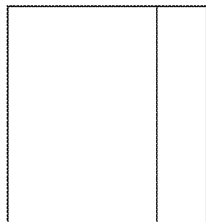
도면3



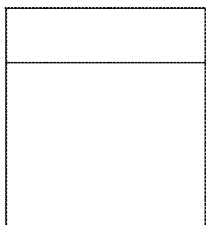
도면4



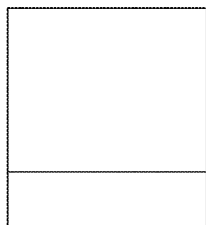
비대칭 파티셔닝  
타입 1



비대칭 파티셔닝  
타입 2

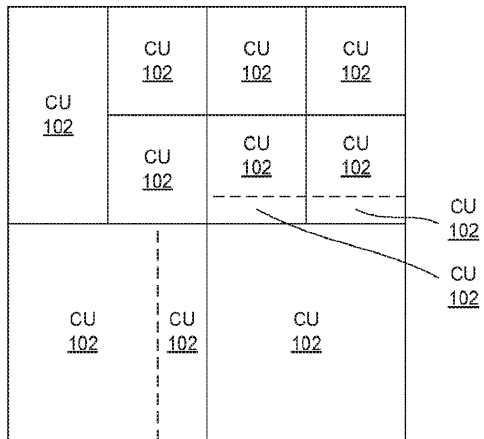


비대칭 파티셔닝  
타입 3

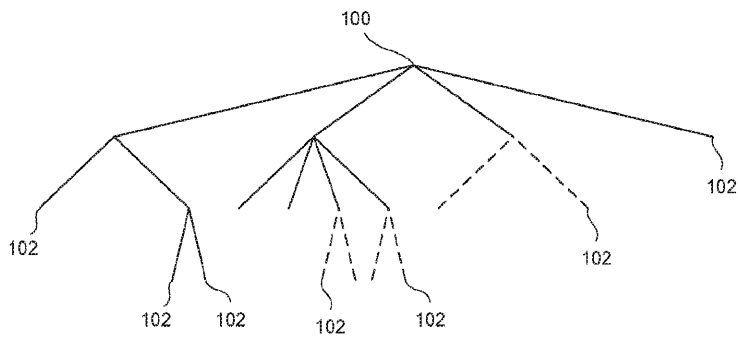


비대칭 파티셔닝  
타입 4

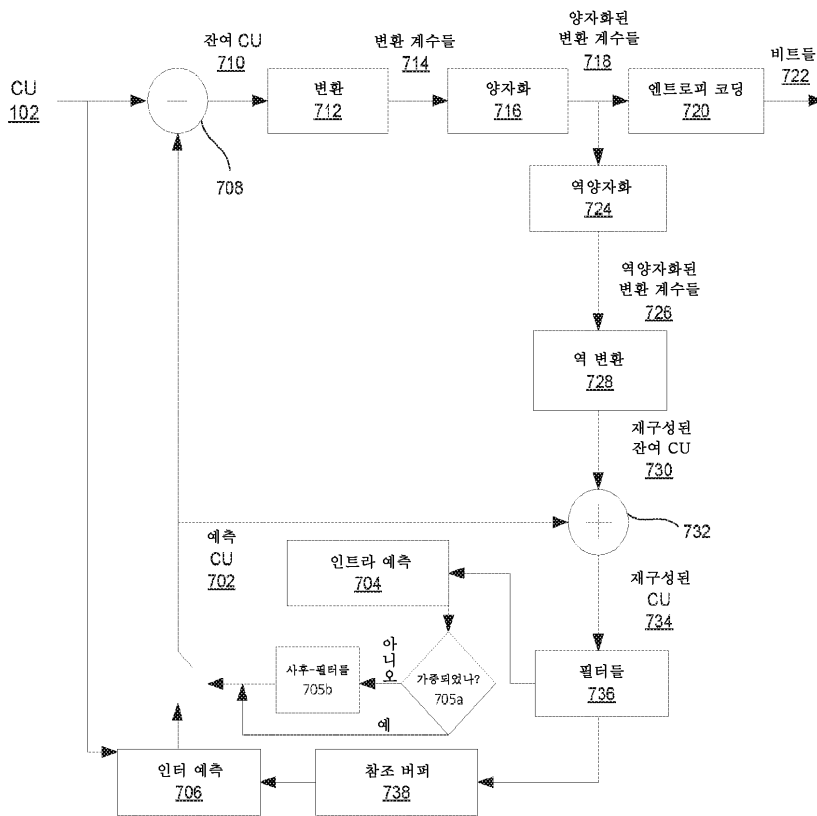
도면5



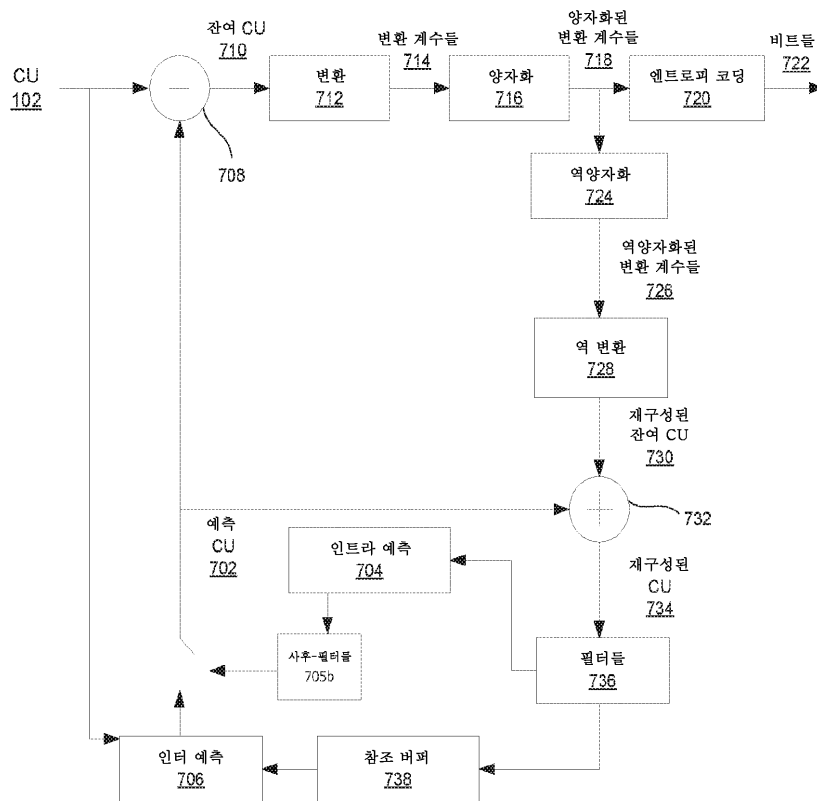
도면6



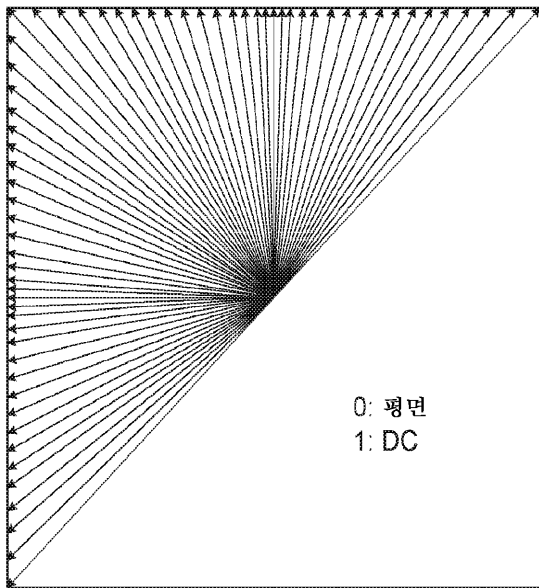
도면7a



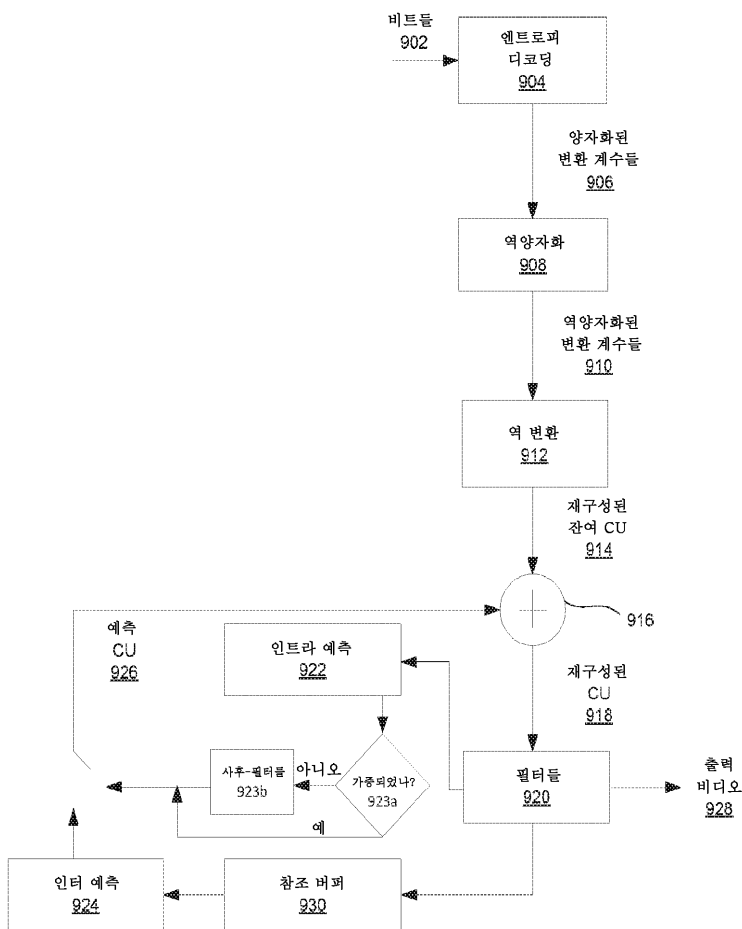
도면7b



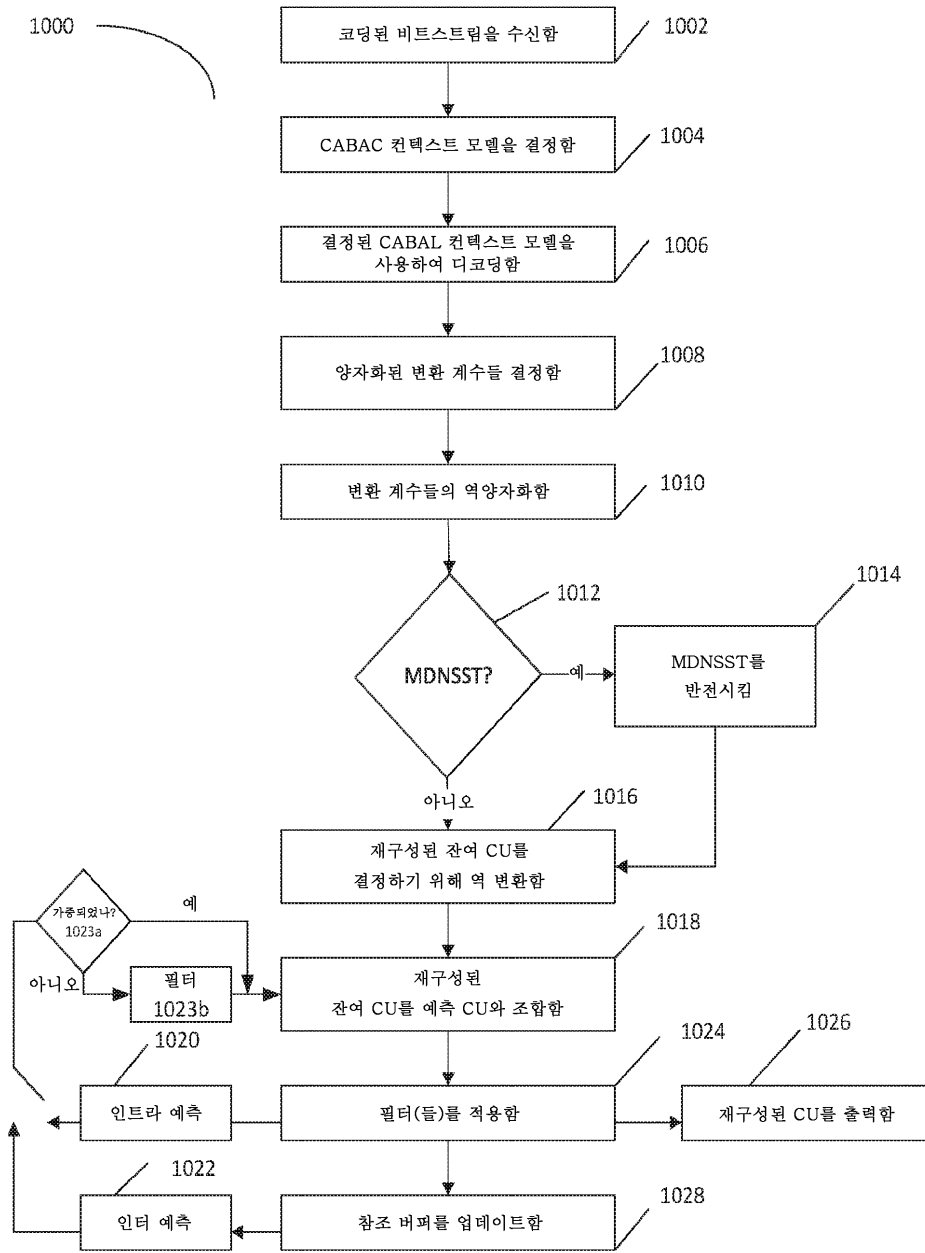
도면8



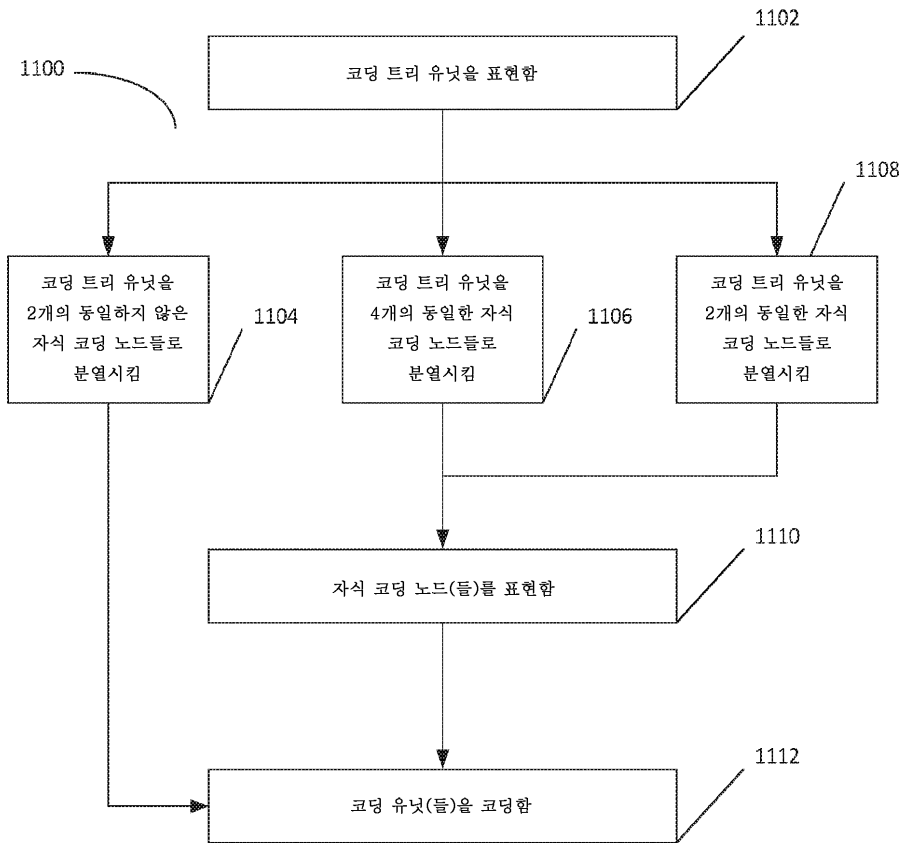
도면9



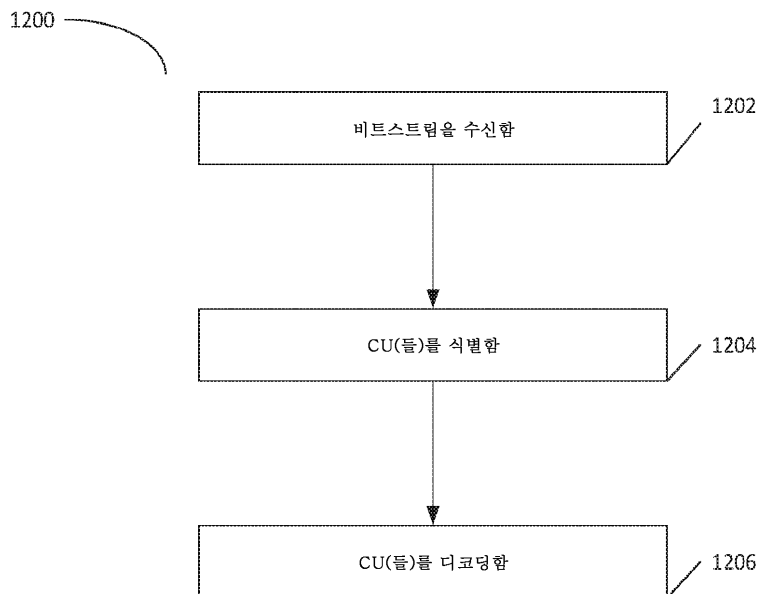
도면10



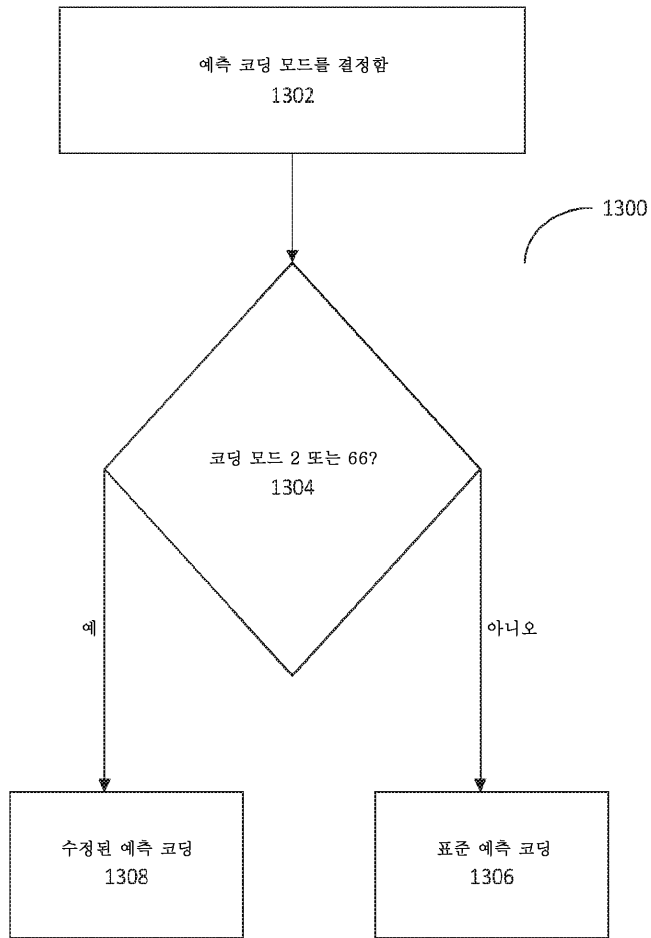
도면11



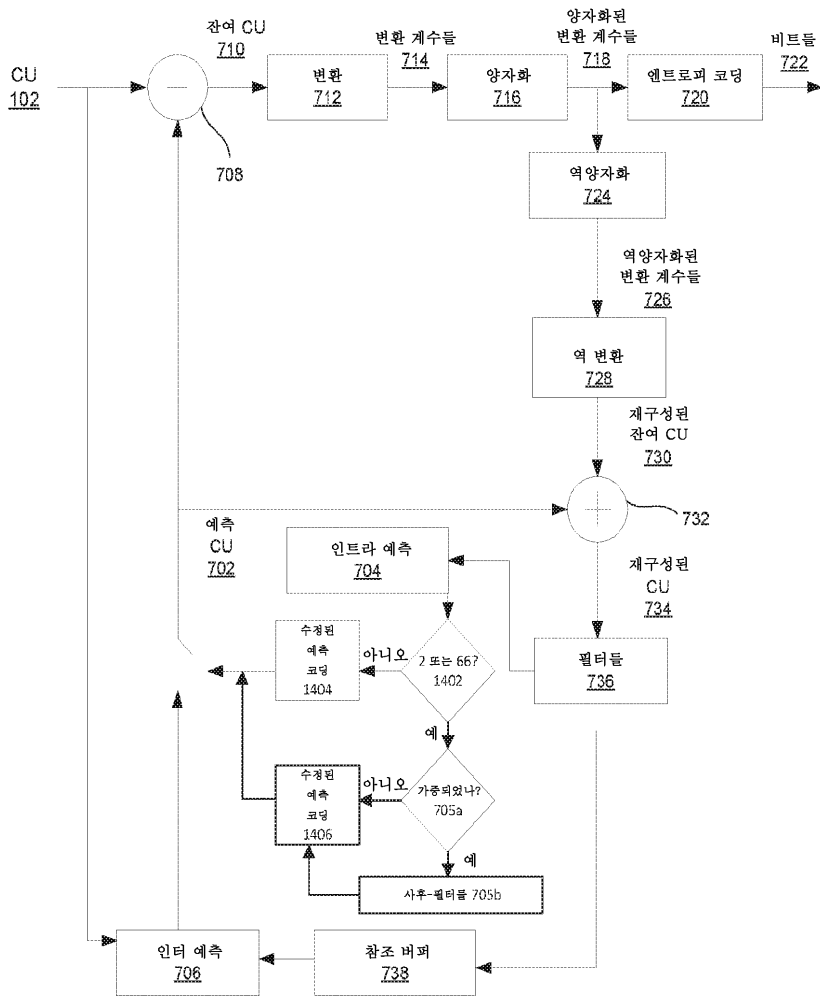
도면12



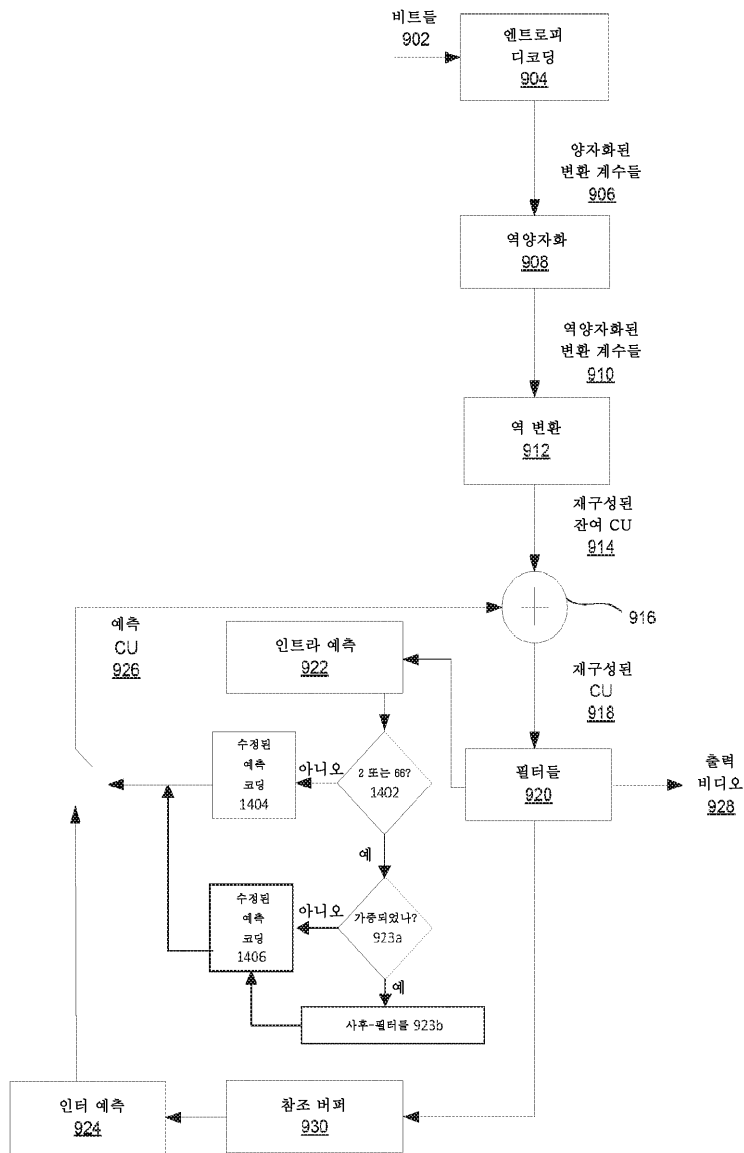
도면13



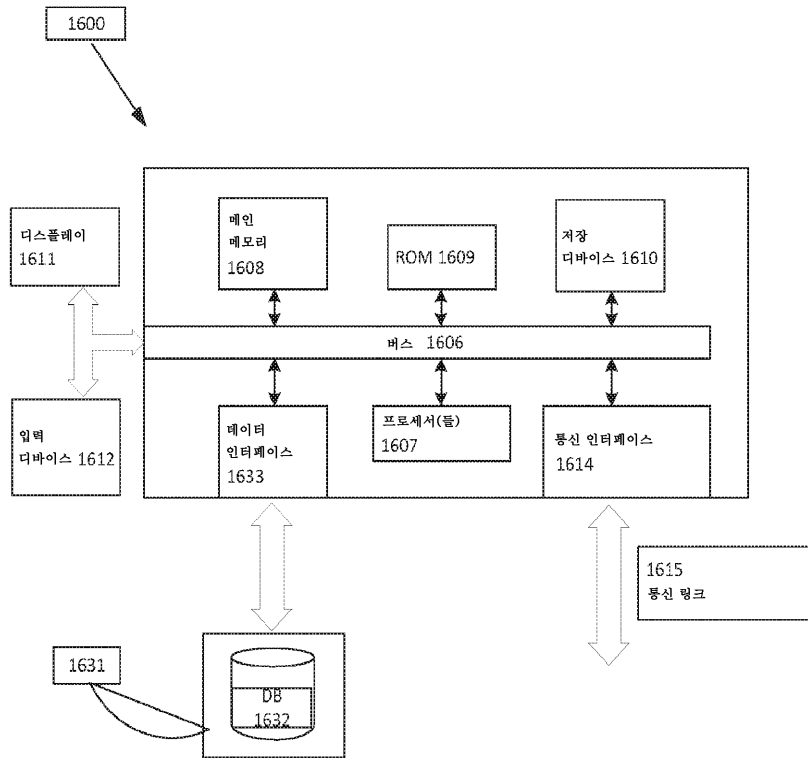
도면14



도면15



도면16



도면17

