(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2025-0087731
(43) 공개일자 2025년06월16일

(51) 국제특허분류(Int. Cl.)
H04N 19/52 (2014.01) *H04N 19/105* (2014.01)
H04N 19/132 (2014.01) *H04N 19/137* (2014.01)
H04N 19/176 (2014.01) *H04N 19/436* (2014.01)
H04N 19/44 (2014.01) *H04N 19/577* (2014.01)
H04N 19/70 (2014.01)
(52) CPC특허분류
H04N 19/52 (2015.01)
H04N 19/105 (2015.01)
(21) 출원번호 10-2025-7018781(분할)
(22) 출원일자(국제) 2019년05월17일
심사청구일자 없음
(62) 원출원 특허 10-2020-7034025
원출원일자(국제) 2019년05월17일
심사청구일자 2022년05월17일
(85) 번역문제출일자 2025년06월05일
(86) 국제출원번호 PCT/US2019/033000
(87) 국제공개번호 WO 2019/231724
국제공개일자 2019년12월05일
(30) 우선권주장
18305646.4 2018년05월28일
유럽특허청(EPO)(EP)
18305913.8 2018년07월10일
유럽특허청(EPO)(EP)

(71) 출원인
인터디지털 브이씨 홀딩스 인코포레이티드
미국 19809 델라웨어주 윌밍턴 스위트 300 벨뷰 파크웨이 200
(72) 발명자
로베르, 앙투안
프랑스 35576 쎄쏭 쎄비네 자크 데 샹 블랑 아브뉘 데 샹 블랑 975 떼끄니꼴로르 에르 에 데 프랑스
르레아넥, 파브리쓰
프랑스 35576 쎄쏭 쎄비네 자크 데 샹 블랑 아브뉘 데 샹 블랑 975 떼끄니꼴로르 에르 에 데 프랑스
갈팡, 프랑크
프랑스 35576 쎄쏭 쎄비네 자크 데 샹 블랑 아브뉘 데 샹 블랑 975 떼끄니꼴로르 에르 에 데 프랑스
(74) 대리인
양영준, 김연송, 백만기

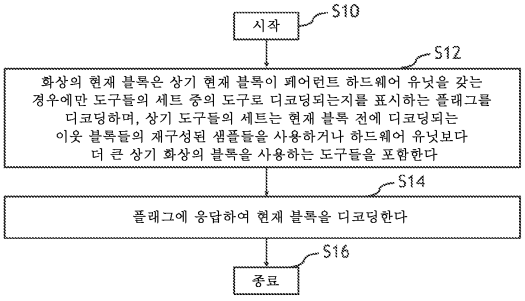
전체 청구항 수 : 총 12 항

(54) 발명의 명칭 **코딩/디코딩에서의 데이터 종속**

(57) 요 약

디코딩 방법이 개시된다. 플래그가 우선 디코딩된다. 플래그는 화상의 현재 블록이 도구들의 세트 중의 도구로 디코딩되는지를 표시한다. 도구들의 세트는 현재 블록 전에 디코딩되는 이웃 블록들의 재구성된 샘플들을 사용하거나 하드웨어 유닛보다 더 큰 화상의 블록을 사용하는 도구들을 포함한다. 플래그는 현재 블록이 페어런트 하드웨어 유닛을 갖는 경우에만 디코딩된다. 그 다음, 현재 블록은 플래그에 응답하여 디코딩된다.

대 표 도 - 도22



(52) CPC특허분류
H04N 19/132 (2015.01)
H04N 19/137 (2015.01)
H04N 19/176 (2015.01)
H04N 19/436 (2015.01)
H04N 19/44 (2015.01)
H04N 19/577 (2015.01)
H04N 19/70 (2015.01)

명 세 서

청구범위

청구항 1

디코딩 방법으로서,

- 비디오 데이터로부터 화상의 제1 블록에 대해 시그널링된 제1 움직임 벡터를 디코딩하는 단계;

- 제2 움직임 벡터를 생성하기 위해 상기 제1 움직임 벡터를 리파인하는 단계 - 상기 제1 블록은 상기 제2 움직임 벡터에 의해 지정된 제1 참조 화상의 제1 영역으로부터 시간적으로 예측됨 -; 및

- 상기 비디오 데이터로부터 제2 블록을 디코딩하는 단계 - 상기 제2 블록은 제3 움직임 벡터에 의해 지정된 제2 참조 화상의 제2 영역으로부터 시간적으로 예측되고, 상기 제3 움직임 벡터는 조건이 충족된 것에 응답하여 상기 제2 움직임 벡터로부터 예측되고, 상기 제3 움직임 벡터는 상기 조건이 충족되지 않은 것에 응답하여 상기 제1 움직임 벡터로부터 예측되고, 상기 조건은 상기 제1 블록 및 상기 제2 블록이 동일한 화상 데이터 유닛에 있는지 여부에 의존함 -를 포함하는, 디코딩 방법.

청구항 2

제1항에 있어서, 상기 제3 움직임 벡터는 상기 조건이 충족되지 않은 것에 응답하여 상기 제1 움직임 벡터의 디코딩 후에 계산되는 것인, 디코딩 방법.

청구항 3

적어도 하나의 프로세서를 포함하는 디코딩 장치로서, 상기 프로세서는,

- 비디오 데이터로부터 화상의 제1 블록에 대해 시그널링된 제1 움직임 벡터를 디코딩하고;

- 제2 움직임 벡터를 생성하기 위해 상기 제1 움직임 벡터를 리파인하고 - 상기 제1 블록은 상기 제2 움직임 벡터에 의해 지정된 제1 참조 화상의 제1 영역으로부터 시간적으로 예측됨 -; 그리고

- 상기 비디오 데이터로부터 제2 블록을 디코딩하는 것 - 상기 제2 블록은 제3 움직임 벡터에 의해 지정된 제2 참조 화상의 제2 영역으로부터 시간적으로 예측되고, 상기 제3 움직임 벡터는 조건이 충족된 것에 응답하여 상기 제2 움직임 벡터로부터 예측되고, 상기 제3 움직임 벡터는 상기 조건이 충족되지 않은 것에 응답하여 상기 제1 움직임 벡터로부터 예측되고, 상기 조건은 상기 제1 블록 및 상기 제2 블록이 동일한 화상 데이터 유닛에 있는지 여부에 의존함 - 을 수행하도록 구성되는, 디코딩 장치.

청구항 4

제3항에 있어서, 상기 제3 움직임 벡터는 상기 조건이 충족되지 않은 것에 응답하여 상기 제1 움직임 벡터의 디코딩 후에 계산되는 것인, 디코딩 장치.

청구항 5

인코딩 방법으로서,

- 비디오 데이터에서 화상의 제1 블록의 제1 움직임 벡터를 시그널링하는 단계;

- 제2 움직임 벡터를 생성하기 위해 상기 제1 움직임 벡터를 리파인하는 단계 - 상기 제1 블록은 상기 제2 움직임 벡터에 의해 지정된 제1 참조 화상의 제1 영역으로부터 시간적으로 예측됨 -; 및

- 상기 비디오 데이터에서 제2 블록을 시그널링하는 단계 - 상기 제2 블록은 제3 움직임 벡터에 의해 지정된 제2 참조 화상의 제2 영역으로부터 시간적으로 예측되고, 상기 제3 움직임 벡터는 조건이 충족된 것에 응답하여 상기 제2 움직임 벡터로부터 예측되고, 상기 제3 움직임 벡터는 상기 조건이 충족되지 않은 것에 응답하여 상기 제1 움직임 벡터로부터 예측되고, 상기 조건은 상기 제1 블록 및 상기 제2 블록이 동일한 화상 데이터 유닛에 있는지 여부에 의존함 -를 포함하는, 인코딩 방법.

청구항 6

제5항에 있어서, 상기 제3 움직임 벡터는 상기 조건이 충족되지 않은 것에 응답하여 상기 제1 움직임 벡터의 결정 후에 계산되는 것인 인코딩 방법.

청구항 7

적어도 하나의 프로세서를 포함하는 인코딩 장치로서, 상기 프로세서는,

- 비디오 데이터에서 화상의 제1 블록의 제1 움직임 벡터를 시그널링하고;

- 제2 움직임 벡터를 생성하기 위해 상기 제1 움직임 벡터를 리파인하고 - 상기 제1 블록은 상기 제2 움직임 벡터에 의해 지정된 제1 참조 화상의 제1 영역으로부터 시간적으로 예측됨 -; 그리고

- 상기 비디오 데이터에서 제2 블록을 시그널링하는 것 - 상기 제2 블록은 제3 움직임 벡터에 의해 지정된 제2 참조 화상의 제2 영역으로부터 시간적으로 예측되고, 상기 제3 움직임 벡터는 조건이 충족된 것에 응답하여 상기 제2 움직임 벡터로부터 예측되고, 상기 제3 움직임 벡터는 상기 조건이 충족되지 않은 것에 응답하여 상기 제1 움직임 벡터로부터 예측되고, 상기 조건은 상기 제1 블록 및 상기 제2 블록이 동일한 화상 데이터 유닛에 있는지 여부에 의존함 - 을 수행하도록 구성되는, 인코딩 장치.

청구항 8

제7항에 있어서, 상기 제3 움직임 벡터는 상기 조건이 충족되지 않은 것에 응답하여 상기 제1 움직임 벡터의 결정 후에 계산되는 것인, 인코딩 장치.

청구항 9

프로세서 판독 가능 매체로서, 제5항의 방법에 의해 발생되는 신호를 저장하는, 프로세서 판독 가능 매체.

청구항 10

프로세서 판독 가능 매체로서, 제7항의 장치에 의해 발생되는 신호를 저장하는, 프로세서 판독 가능 매체.

청구항 11

정보 저장 매체로서, 제1항에 따른 방법을 구현하기 위한 프로그램 코드 명령어들을 저장하는, 정보 저장 매체.

청구항 12

정보 저장 매체로서, 제5항에 따른 방법을 구현하기 위한 프로그램 코드 명령어들을 저장하는, 정보 저장 매체.

발명의 설명

기 술 분 야

[0001] 하나 이상의 구현의 기술분야는 일반적으로 비디오 압축에 관한 것이다. 본 실시예들은 일반적으로 화상 인코딩 및 디코딩을 위한 방법 및 디바이스에 관한 것이다.

배 경 기 술

[0002] 높은 압축 효율을 달성하기 위해, 이미지 및 비디오 코딩 방식들은 통상 비디오 콘텐츠에서 공간 및 시간 리던던시를 활용하는 예측 및 변환을 이용한다. 일반적으로, 인트라 또는 인터 예측은 인트라 또는 인터 프레임 상관을 활용하기 위해 사용되며, 그 다음 종종 예측 에러들 또는 예측 잔차들로 표시되는, 원래의 이미지 블록과 예측된 이미지 블록 사이의 차이들은 변환되고, 양자화되고, 엔트로피 코딩된다. 인코딩 동안, 원래의 이미지 블록은 통상 가능하게는 쿼드 트리 파티셔닝을 사용하여 서브 블록들로 파티션/분할된다. 비디오를 재구성하기 위해, 압축된 데이터는 예측, 변환, 양자화 및 엔크로피 코딩에 대응하는 역 프로세스들에 의해 디코딩된다.

발명의 내용

[0003] 디코딩 방법이 개시되며, 방법은,

[0004] - 화상의 현재 블록이 페어런트 하드웨어 유닛을 갖는 경우에만 현재 블록이 도구들의 세트 중의 도구로 디코딩되는지를 표시하는 플래그를 디코딩하는 단계 - 도구들의 세트는 현재 블록 전에 디코딩되는 이웃 블록들의 재구성된 샘플들을 사용하거나 하드웨어 유닛보다 더 큰 화상의 블록을 사용하는 도구들을 포함함 - ; 및

[0005] - 플래그에 응답하여 현재 블록을 디코딩하는 단계를 포함한다.

[0006] 적어도 하나의 프로세서를 포함하는 디코딩 장치가 개시되며, 프로세서는,

[0007] - 화상의 현재 블록이 페어런트 하드웨어 유닛을 갖는 경우에만 현재 블록이 도구들의 세트 중의 도구로 디코딩되는지를 표시하는 플래그를 디코딩하는 것 - 도구들의 세트는 현재 블록 전에 디코딩되는 이웃 블록들의 재구성된 샘플들을 사용하거나 하드웨어 유닛보다 더 큰 화상의 블록을 사용하는 도구들을 포함함 - ; 및

[0008] - 플래그에 응답하여 현재 블록을 디코딩하는 것을 수행하도록 구성된다.

[0009] 인코딩 방법이 개시되며, 방법은,

[0010] - 화상의 현재 블록이 페어런트 하드웨어 유닛을 갖는 경우에만 현재 블록이 도구들의 세트 중의 도구로 인코딩되는지를 표시하는 플래그를 인코딩하는 단계 - 도구들의 세트는 현재 블록 전에 인코딩되는 이웃 블록들의 재구성된 샘플들을 사용하거나 하드웨어 유닛보다 더 큰 화상의 블록을 사용하는 도구들을 포함함 - ; 및

[0011] - 플래그에 응답하여 현재 블록을 인코딩하는 단계를 포함한다.

[0012] 적어도 하나의 프로세서를 포함하는 인코딩 장치로서, 프로세서는,

[0013] - 화상의 현재 블록이 페어런트 하드웨어 유닛을 갖는 경우에만 현재 블록이 도구들의 세트 중의 도구로 인코딩되는지를 표시하는 플래그를 인코딩하는 것 - 도구들의 세트는 현재 블록 전에 인코딩되는 이웃 블록들의 재구성된 샘플들을 사용하거나 하드웨어 유닛보다 더 큰 화상의 블록을 사용하는 도구들을 포함함 - ; 및

[0014] - 플래그에 응답하여 현재 블록을 인코딩하는 것을 수행하도록 구성된다.

[0015] 신호가 개시되며, 신호는,

[0016] - 화상의 현재 블록이 페어런트 하드웨어 유닛을 갖는 경우에만 현재 블록이 도구들의 세트 중의 도구로 인코딩되는지를 표시하는 플래그 - 도구들의 세트는 현재 블록 전에 인코딩되는 이웃 블록들의 재구성된 샘플들을 사용하거나 하드웨어 유닛보다 더 큰 화상의 블록을 사용하는 도구들을 포함함 - ; 및

[0017] - 플래그에 응답하여 코딩되는 현재 블록을 나타내는 데이터를 포함한다.

[0018] 디바이스가 개시되며, 디바이스는,

[0019] 개시되는 실시예들 중 어느 하나에 따른 디코딩 장치; 및

[0020] (i) 무선으로(over the air) 신호를 수신하도록 구성되는 안테나 - 신호는 이미지를 갖는 비디오를 포함함 - , (ii) 이미지를 갖는 비디오를 포함하는 수신된 신호를 주파수들의 대역으로 제한하도록 구성되는 대역 제한기, 또는 (iii) 출력을 디스플레이하도록 구성되는 디스플레이 중 적어도 하나를 포함한다.

[0021] 컴퓨터 프로그램 제품이 개시되며, 이 제품은 프로그램이 하나 이상의 프로세서에 의해 실행될 때, 하나 이상의 프로세서로 하여금 개시되는 실시예들 중 어느 하나의 디코딩 또는 인코딩 방법을 수행하게 하는 명령어들을 포함한다.

도면의 간단한 설명

[0022] 도 1은 HEVC에서의 코딩 유닛들(Coding Units)(CU)로 분할되는 코딩 트리 유닛들(Coding Tree Units)(CTU)을 표현한다.

도 2는 HEVC에서의 하나 이상의 예측 유닛(Prediction Units)(PU) 및 변환 유닛(Transform Units)(TU)으로 공간적으로 파티션되는 CTU를 표현한다.

도 3은 2개의 상이한 참조 화상 내의 현재 CU의 움직임 궤적을 따라 2개의 블록 사이의 매치를 발견하는 프로세스를 예시한다.

도 4는 현재 화상 내의 템플릿과 대응하는 템플릿 사이의 매치를 발견하는 프로세스를 예시한다.

도 5는 현재 화상 내에 위치되는 L-shape(L-형상)을 참조 화상 내의 대응하는 L-shape과 비교함으로써 조명 보

상 파라미터들을 유도하는 프로세스를 예시한다.

도 6은 적어도 하나의 실시예의 일반적 양태에 따른 2개의 코딩 유닛 및 공간적으로 대응하는 하드웨어 유닛들을 도시한다.

도 7은 적어도 하나의 실시예의 일반적 양태에 따른 전체 디코딩 프로세스의 각각의 스테이지에서 처리의 입도를 예시한다.

도 8은 적어도 하나의 실시예의 일반적 양태에 따른 인터 프레임을 디코딩하기 위한 처리 파이프라인의 일 예를 도시한다.

도 9는 비디오 인코더의 일 실시예의 블록도를 예시한다.

도 10은 비디오 디코더의 일 실시예의 블록도를 예시한다.

도 11은 분할 트리 및 연관된 분할 CTU의 일 예를 도시한다.

도 12는 적어도 하나의 실시예의 일반적 양태에 따른 페어런트 하드웨어 유닛(Hardware Unit)(HU)을 결정하기 위한 프로세스를 예시한다.

도 13은 적어도 하나의 실시예의 일반적 양태에 따른 Tdep 내의 도구를 활성하는지를 결정하기 위한 프로세스를 예시한다.

도 14는 적어도 하나의 실시예의 일반적 양태에 따른 Tdep 도구들이 사용될 수 있는 CU들을 그레이로 도시한다.

도 15는 적어도 하나의 실시예의 일반적 양태에 따른 플래그를 파싱하기 위한 프로세스를 예시한다.

도 16은 병합 후보들에 대한 움직임 벡터 예측기 유도를 예시한다.

도 17은 적어도 하나의 실시예의 일반적 양태에 따른 CU에 대한 움직임 벡터 업데이트(좌측에서) 및 움직임 예측기 유도(우측에서)의 프로세스를 예시한다.

도 18은 적어도 하나의 실시예의 일반적 양태에 따른 CU에 대한 움직임 벡터 업데이트(좌측) 및 HU에 의존하는 움직임 예측기 유도(우측에서)의 프로세스를 예시한다.

도 19는 적어도 하나의 실시예의 일반적 양태에 따른 CU를 2 HU로 분할하는 프로세스를 예시한다.

도 20은 적어도 하나의 실시예의 일반적 양태에 따른 CU를 분할하는 일 예를 도시한다.

도 21은 본 실시예들의 양태들이 구현될 수 있는 시스템의 블록도를 예시한다.

도 22는 일 실시예에 따른 디코딩 방법의 흐름도를 도시한다.

도 23은 일 실시예에 따른 인코딩 방법의 흐름도를 도시한다.

발명을 실시하기 위한 구체적인 내용

[0023] HEVC 비디오 압축 표준에서, 움직임 보상 시간 예측은 비디오의 연속적 화상들 사이에 존재하는 리던던시를 활용하기 위해 이용된다.

[0024] 그렇게 하기 위해, 움직임 벡터는 이제 도입하는 각각의 예측 유닛(PU)에 연관된다. 각각의 코딩 트리 유닛(CTU)는 보상된 도메인 내의 코딩 트리에 의해 표현된다. 이것은 도 1에 도시된 바와 같이, CTU의 쿼드 트리 분할이며, 각각의 리프는 코딩 유닛(CU)으로 칭해진다.

[0025] 그 다음, 각각의 CU는 일부 인트라 또는 인터 예측 파라미터들(예측 정보)을 제공받는다. 그렇게 하기 위해, 그것은 하나 이상의 예측 유닛(PUs)으로 공간적으로 파티션된다, 각각의 PU는 일부 예측 정보를 할당받는다. 인트라 또는 인터 코딩 모드는 도 2에 도시된 바와 같이, CU 레벨 상에 할당된다.

[0026] 움직임 벡터는 HEVC 내의 각각의 PU에 할당된다. 이러한 움직임 벡터는 고려된 PU의 움직임 보상 시간 예측을 위해 사용된다. 따라서, HEVC에서, 예측된 블록 및 그의 참조 블록을 링크하는 움직임 모델은 간단히 병진에 있다.

[0027] JVET(Joint Video Exploration Team) 그룹에 의해 개별되는 JEM(Joint Exploration Model)에서, 일부 움직임 모델들은 시간 예측을 개선하기 위해 지원된다. 그렇게 하기 위해, PU는 서브 PU로 공간적으로 분할될 수 있고

모델은 전용 움직임 벡터를 각각의 서브 PU에 할당하기 위해 사용될 수 있다.

[0028] JEM의 다른 버전들에서, CU는 더 이상 PU 또는 TU로 분할되지 않고, 일부 움직임 데이터는 각각의 CU에 직접 할당된다. 이러한 새로운 코덱 디자인에서, CU는 서브 CU로 분할될 수 있고 움직임 벡터는 각각의 서브 CU에 대해 계산될 수 있다.

[0029] 인터 프레임 움직임 보상을 위해, 디코더 측 파라미터 추정을 이용하는 한 세트의 새로운 도구들이 개발되었다: 프레임 레이트 업 변환(Frame Rate Up Conversion)(FRUC) 병합, FRUC 양측 및 조명 보상(Illumination Compensation)(IC).

[0030] FRUC 도구는 이하와 같이 설명된다. FRUC는 시그널링 없이 디코더 측에서 CU의 움직임 정보를 유도하는 것을 허용한다. 이러한 모드는 어느 매칭 비용 함수(양측 또는 템플릿)가 CU에 대한 움직임 정보를 유도하는 데 사용될지를 표시하기 위해 FRUC 플래그 및 부가적 FRUC 모드 플래그를 갖는 CU 레벨에서 시그널링된다.

[0031] 인코더 측에서, CU에 대한 FRUC 병합 모드을 사용하는지의 결정은 RD 비용 선택에 기초한다. 2개의 매칭 모드(양측 및 템플릿)는 CU에 대해 둘 다 체크된다. 최소 RD 비용을 초래하는 것은 다른 코딩 모드들과 추가로 비교된다. FRUC 모드가 가장 효율적인 것이면, FRUC 플래그가 CU에 적법하게 설정되고 관련 매칭 모드가 사용된다.

[0032] FRUC 병합 모드에서의 움직임 유도 프로세스는 2개의 단계를 갖는다. CU-레벨 움직임 검색은 우선 수행되며, 그 다음 서브 CU 레벨 움직임 리파인먼트가 이어진다. CU 레벨에서, 초기 움직임 벡터는 양측 또는 템플릿 매칭에 기초하여 전체 CU에 대한 MV 후보들의 리스트로부터 유도된다. 최소 매칭 비용을 초래하는 후보는 추가 CU 레벨 리파인먼트에 대한 시작 지점으로서 선택된다. 그 다음, 시작 지점 주위에서 양측 또는 템플릿 매칭에 기초한 국부 검색이 수행되고 최소 매칭 비용을 야기하는 MV는 전체 CU에 대한 MV로서 취해진다. 그 후에, 움직임 정보는 시작 지점으로서 유도된 CU 움직임 벡터들을 갖는 서브 CU 레벨에서 추가로 리파인된다.

[0033] 도 3에 도시된 바와 같이, 양측 매칭 비용 함수는 2개의 상이한 참조 화상 내의 현재 CU의 움직임 궤적을 따라 2개의 블록 사이의 최상의 매치를 발견함으로써 현재 CU의 움직임 정보를 유도하기 위해 사용된다. 연속적 움직임 궤적의 가정 하에, 2개의 참조 블록을 나타내는 움직임 벡터들(MV0 및 MV1)은 현재 화상과 2개의 참조 화상(TD0 및 TD1) 사이의 공간 거리들에 비례할 것이다.

[0034] 도 4에 도시된 바와 같이, 템플릿 매칭 비용 함수는 현재 화상 내의 템플릿(현재 CU의 상단 및/또는 좌측 이웃 블록들)과 참조 화상 내의 대응하는 템플릿(템플릿과 동일한 크기) 사이의 최상의 매치를 발견함으로써 현재 CU의 움직임 정보를 유도하기 위해 사용된다.

[0035] 템플릿 매칭 비용 함수를 사용하는 이러한 FRUC 모드가 또한 어드밴스트 움직임 벡터 예측(Advanced Motion Vector Prediction)(AMVP) 모드에 적용된다는 점을 주목한다. 이러한 경우에, AMVP는 2개의 후보를 갖는다. 새로운 후보는 템플릿 매칭을 갖는 FRUC 도구를 사용하여 유도된다. 이러한 FRUC 후보가 제1 기존 AMVP 후보들과 상이하면, 그것은 AMVP 후보 리스트의 초반부에 삽입되고 그 다음 리스트 크기는 2개로 설정된다(제2 기존 AMVP 후보를 제거하는 것을 의미함). AMVP 모드에 적용될 때, CU 레벨 검색만이 적용된다.

[0036] 인터 모드에서, 조명 보상(IC)은 공간 또는 시간 국부 조명 변화를 고려함으로써 움직임 보상(Motion Compensation)(MC)을 통해 획득되는 블록 예측 샘플들을 정정하는 것을 허용한다. IC 파라미터들은 도 5에 도시된 바와 같이 재구성된 이웃 샘플들(L-shape-cur)의 세트(S)를 reference-i 블록(i=0 또는 1)의 이웃 샘플들(L-shape-ref-i)과 비교함으로써 추정된다.

[0037] IC 파라미터들은 L-shape-cur 내의 샘플들과 IC 파라미터들로 정정되는 L-shape-ref-i의 샘플들 사이의 차이(최소 제곱법)를 최소화한다. 전형적으로, IC 모델은 선형이다: IC(x) = a.x+b이며, 여기서 x는 보상할 샘플의 값이다.

[0038] 파라미터들(a 및 b)은 인코더에서(및 디코더에서) L-shape들 상에 최소 제곱 최소화를 해결함으로써 유도된다:

[0039] $$(a_i, b_i) = \underset{(a,b)}{\operatorname{argmin}} \left(\sum_{\substack{x \in L-shape-cur, \\ y \in L-shape-ref-i}} (x - a.y - b)^2 \right) \qquad (2)$$

[0040] 최종적으로, a_i 는 정수 가중치(a_i) 및 시프트(sh_i)로 변환되고 MC 블록은 IC에 의해 정정된다:

[0041] $$Pred_i = (a_i * x_i >> sh_i) + b_i \qquad (3)$$

[0042] 적어도 하나의 실시예에 의해 해결되는 문제는 주어진 블록 크기 외부에서 데이터 종속들을 제한하는 방법이다.

[0043] 도 6은 하드웨어 디코더 상의 가능한 처리의 일 예를 도시한다. 이러한 예에서, 디코더가 취급할 수 있는 하드웨어 유닛(HU)의 최대 크기(전형적으로 코덱의 최대 변환에 설정됨)가 64 픽셀인 것을 가정한다. 병렬성을 증가시키기 위해, 디코더는 그들을 계산 모듈(전형적으로 움직임 보상 및 잔차 계산)에 송신하기 전에 64 픽셀보다 더 큰 CU를 64의 최대 크기의 블록으로 분할할 것이다.

[0044] 도 7은 전체 디코딩 프로세스의 각각의 스테이지에서 입도(가능한 병렬성을 암시함)의 일 예를 도시한다. 처음에, 비트스트림은 파싱되고 디코딩할 한 세트의 CU를 발생시킨다. 각각의 CU가 디코딩되어, 재구성된 샘플들을 계산하기 위해 사용되는 최종 값들을 제공한다. 일부 인과 종속들이 CU 디코딩 사이에 존재할 수 있다는 점을 주목한다(전형적으로 움직임 벡터의 값은 이미 디코딩된(그러나 재구성되지 않은) CU의 값으로부터 예측될 수 있음). 그 다음, CU는 필요하면(크기에 따라) 수개의 HU로 분할된다. 이러한 지점에서, HU 내부의 모든 계산은 다른 HU와 독립적으로 행해져야 한다. 인트라 CU에 대해, 인트라 예측 프로세스가 예측 스테이지 동안 이웃 샘플들에 액세스할 필요가 있기 때문에 재구성된 CU 사이에 인과 종속이 여전히 존재한다는 점을 주목한다.

[0045] 적어도 하나의 실시예에 의해 해결되는 문제는 FRUC와 같은 도구들에 의해 생성되는 데이터 종속을 완화하는 방법이다. 도 8은 인터 프레임을 디코딩하기 위한 처리 파이프라인의 일 예를 도시한다. 이러한 종류의 파이프라인을 고려할 때 FRUC 또는 국부 조명 보상(Local Illumination Compensation)(LIC)과 같은 도구들로 발생하는 하나의 쟁점은 이전 CU의 재구성된 샘플들 상에 데이터 종속을 도입하므로, HU 처리의 가능한 병렬성을 중단하라는 것이다.

[0046] 여기에 설명되는 실시예들 중 적어도 하나는 이러한 종속을 회피하고 디코더에서 높은 병렬 파이프라인을 허용하는 방법들이다.

[0047] FRUC, 디코더 움직임 벡터 리파인먼트(Decoder Motion Vector Refinement)(DMVR) 및 LIC는 JEM에서 새로운 모드들이므로 파이프라인 스톨링은 비교적 새로운 문제이다.

[0048] 적어도 하나의 실시예의 기본적 생각은 HU 레벨에서 종속을 중단시키는 것이다. 제안된 실시예들 중 적어도 하나는 코덱의 규범 수정들을 수반한다: 인코딩 및 디코딩 프로세스들은 완전히 대칭이다. 하나 이상의 실시예의 임팩티드(impacted) 코덱 모듈들은 모듈(170) 및 가능하게는 도 9의 160 및 도 10의 275 가능하게는 260이다.

[0049] 전형적으로, 이전 CU, 즉 현재 CU 전에 코딩되는 CU로부터 재구성된 샘플들을 사용하거나, 디코더 측에서 HU보다 더 큰 CU를 사용하는 도구들의 세트를 Tdep로 칭한다:

[0050] - 템플릿 모드에서의 FRUC는 현재 CU 외부의 재구성된 샘플들을 사용할 수 있고;

[0051] - LIC는 현재 CU 외부의 재구성된 샘플들을 사용할 수 있고;

[0052] - DMVR은 HU 크기보다 더 큰 CU를 사용할 수 있고;

[0053] - BIO(BI-directional Optical flow)(양방향 광학적 흐름)는 HU 크기보다 더 큰 CU를 사용할 수 있다.

[0054] 도 22는 일 실시예의 디코딩 방법의 흐름도를 도시한다. 방법은 S10에서 시작된다. 플래그는 디코딩된다(S12). 플래그는 화상의 현재 블록이 도구들의 세트(Tdep) 중의 도구로 디코딩되는지를 표시한다. 플래그는 현재 블록이 페어런트 하드웨어 유닛을 갖는 경우에만 디코딩된다. 도구들의 세트는 현재 블록 전에 디코딩되는 이웃 블록들의 재구성된 샘플들을 사용하거나 하드웨어 유닛보다 더 큰 상기 화상의 블록을 사용하는 도구들을 포함한다. 현재 블록은 플래그에 응답하여 디코딩된다(S14). 방법은 S16에서 종료된다.

[0055] 도 23은 일 실시예에 따른 인코딩 방법의 흐름도를 도시한다. 방법은 S20에서 시작된다. 플래그는 인코딩된다(S22). 플래그는 화상의 현재 블록이 도구들의 세트(Tdep) 중의 도구로 인코딩되는지를 표시한다. 플래그는 현재 블록이 페어런트 하드웨어 유닛을 갖는 경우에만 인코딩된다. 도구들의 세트는 현재 블록 전에 인코딩되는 이웃 블록들의 재구성된 샘플들을 사용하거나 하드웨어 유닛보다 더 큰 상기 화상의 블록을 사용하는 도구들을 포함한다. 현재 블록은 플래그에 응답하여 인코딩된다(S24). 방법은 S26에서 종료된다.

[0056] **HU 제한**

[0057] 독립 HU 기반 처리를 사용할 수 있기 위해, 새로운 제약들은 Tdep 도구들 내의 도구의 사용을 위해 추가된다:

[0058] 1. 현재 CU는 페어런트 HU를 갖는다.

[0059] 2. 좌측 및 상단 이웃 CU들은은 현재 CU의 페어런트 HU 내부에 있다.

[0060] 도 11은 분할 트리 및 연관된 분할 CTU의 일 예를 도시한다. CU의 페어런트 HU는 S 이하인 크기(폭 및 높이)를 갖는 분할 트리 내의 제1 CU로서 정의된다. 일 예로서, S=64이다. 예를 들어, 도 11에서, C 및 D는 페어런트 HU이다.

[0061] 도 12는 블록, 예를 들어 CU의 페어런트 HU를 결정하기 위한 프로세스를 도시한다. 프로세스는 단계(S120)에서 시작된다. 단계(S121)에서, h 또는 w가 S보다 더 큰지가 체크되며, 여기서 h 및 w는 현재 블록의 높이 및 폭 각각이다. h>S 또는 w>S인 경우에, 이때 프로세스는 단계(S122)에서 계속되며, 그렇지 않으면 프로세스는 단계(S125)에서 계속된다.

[0062] S122에서, 현재 블록은 어떠한 페어런트 HU도 갖지 않는다.

[0063] 단계(S123)에서, 현재 블록이 차일드 블록들로 추가로 분할되는지가 체크된다. 그렇지 않으면, 이때 프로세스는 종료된다(S129). 예이면, 이때 현재 블록은 차일드 블록들로 추가로 분할되고(S124) 프로세스는 S121에서 차일드 블록들에 대해 계속된다.

[0064] 예이면, 이때 프로세스는 S124에서 계속되며, 그렇지 않으면 프로세스는 종료된다(S129).

[0065] S125에서, HU는 현재 블록(예를 들어 블록 그의 차일드 블록들에 대한 블록(C)과 같음)과 동일하게 설정된다.

[0066] S126에서, 현재 블록의 페어런트 HU는 HU이다.

[0067] S127에서, 현재 블록이 추가로 분할되는지가 체크된다. 그렇지 않으면, 이때 프로세스는 종료된다(S129). 예이면, 이때 현재 블록은 추가로 분할되고(S128) 프로세스는 S126.마진에서 차일드 블록들에 대해 계속된다.

[0068] 제2 기준들은 현재 CU의 이웃 CU가 현재 CU와 동일한 페어런트 HU를 공유하는지를 체크한다. 도 13은 Tdep 도구들 내의 도구를 사용하거나 사용하지 않는 것을 결정하기 위한 프로세스의 흐름도를 도시한다.

[0069] 프로세스는 S130에서 시작된다.

[0070] 단계(S132)에서, 현재 CU가 도 12에 도시된 바와 같은 페어런트 HU를 갖는지가 체크된다. 아니오이면, 이때 프로세스는 S137에서 종료된다(Tdep 도구는 사용되지 않음). 그렇지 않으면, 프로세스는 S133에서 계속되며, 이웃 CU들(예를 들어, 상단 및 좌측)이 현재 CU와 동일한 페어런트 HU를 공유하는지가 체크된다. 그렇지 않으면, 이때 프로세스는 종료된다. 그렇지 않으면, 프로세스는 S134에서 계속된다.

[0071] S134에서, Tdep 도구가 사용되는지의 여부를 표시하는 플래그는 디코딩된다. 플래그가 참인 경우에(S135), 이때 대응하는 도구는 현재 CU를 디코딩하기 위해 사용된다(S136). 그렇지 않으면, 이때 프로세스는 종료되며(S137), 즉 대응하는 Tdep 도구는 사용되지 않는다.

[0072] 도 14는 Tdep 도구들이 사용될 수 있는 CU들의 일부 예들을 (그레이로) 도시한다. HU 크기는 64로 설정된다. 일 실시예에서, Tdep 도구 내의 도구의 플래그의 파싱은 도 13에 설명되는 프로세스에 의해 제약된다. 일 실시예에서, Tdep 도구 내의 도구의 플래그의 디코딩은 도 15에 도시된 바와 같이 페어런트 HU의 존재에 의해서만 제약된다.

[0073] 프로세스는 시작된다(S130). 현재 CU가 페어런트 HU를 갖는지가 체크된다(S132). 플래그가 참인 경우에(S135), 이때 대응하는 도구는 현재 CU를 디코딩하기 위해 사용된다(S136). 그렇지 않으면, 이때 프로세스는 종료되며(S137), 즉 대응하는 Tdep 도구는 사용되지 않는다.

[0074] **HU의 움직임 벡터 전파**

[0075] HU 내부의 CU들의 재구성이 독립적이므로, HU 사이의 값들의 예측은 또한 제한될 수 있다.

[0076] 여기에 Tdep 세트 내의 도구들에 의해 발생되는 값들의 예들이 있다:

[0077] FRUC: CU의 움직임 벡터의 최종 값은 최종 움직임 리파인먼트 후에 주어진다.

[0078] DMVR: CU의 움직임 벡터의 최종 값은 최종 움직임 리파인먼트 후에 주어진다.

[0079] 그 다음, 발생되는 값들은 추가 CU들에 대한 예측으로서 사용될 수 있다(도 16은 예측된 값들의 일 예에 대한 것임). S160에서, 주어진 움직임 예측기가 움직임 벡터 후보들의 리스트에 있는지가 체크된다. 그렇지 않으면, 그것은 리스트에 추가된다(S162).

[0080] CU에 대한 새로운 움직임 벡터를 발생시키는 도구들에 대해, 움직임 벡터 예측은 또한 HU 내부에 제약될 수 있다. 이러한 실시예에서, CU에 의해 사용되는 예측기는 항상 움직임 리파인먼트 전의 값이며, 즉 최종 움직임 벡터는 디코더 측 움직임 벡터 유도 도구들(FRUC, DMVR 등)을 갖는 블록들에 대해 업데이트되지 않는다. S170-1에서 S178-1까지, CU는 FRUC 플래그에 따라, 그 자체의 움직임 보상을 위한 움직임 벡터 후보를 리파인한다. 그러나, 그것은 예측기로서 이웃 CU의 논-리파인된 움직임 벡터를 사용한다(S170-2 내지 S178-2). 도 17은 현재 CU에 대한 움직임 벡터 업데이트(좌측) 및 움직임 예측기 유도(우측)의 프로세스를 도시한다.

[0081] 움직임 벡터 업데이트 프로세스는 S170-1에서 시작된다. S172에서, MV 후보가 디코딩된다.

[0082] S174에서, MV1은 디코딩된 MV 후보와 동일하게 설정된다.

[0083] S175에서, FRUC가 사용되는지(플래그가 참인지)가 체크된다. 그 다음, 예이면, 움직임 벡터 MV1은 MV2를 획득하기 위해 FRUC로 리파인된다(S176). 현재 CU는 MV2로 움직임 보상된다(S177).

[0084] 프로세스는 종료된다(S178-1).

[0085] 움직임 예측기 유도 프로세스는 S170-2에서 시작된다. CUn으로 표시되는 이웃 CU는 예측을 위해 고려된다(S171). S172에서 사용되는 움직임 벡터 예측기는 MV1과 동일하게 설정되고 이웃 CU에 리파인된 벡터와 동일하게 설정되지 않는다.

[0086] **동일한 HU 내부의 CU에 대해서만 전파**

[0087] 도 18은 적어도 하나의 실시예의 일반적 양태에 따른 CU에 대한 움직임 벡터 업데이트(좌측) 및 HU에 의존하는 움직임 예측기 유도(우측에서)의 프로세스를 예시한다. S180-1에서 S190-1까지, CU는 FRUC 플래그에 따라, 움직임 벡터 후보를 리파인한다.

[0088] 그러나, 그것은 이웃 CU가 현재 CU의 HU(S183) 외부에 있는 경우 예측기로서 이웃 CU의 논-리파인된 움직임 벡터(S185)를 사용하고, 그들이 동일한 HU 내에 있으면 리파인된 움직임 벡터(S187)를 사용한다.

[0089] 움직임 벡터 업데이트의 프로세스는 S180-1에서 시작된다. S182에서, MV 후보는 디코딩된다. S184에서, MV1 및 MV2는 디코딩된 움직임 벡터 후보에 설정된다.

[0090] S186에서, FRUC가 사용되는지가 체크된다(플래그는 참임). 그 다음, 예이면, 움직임 벡터(MV1)는 MV2를 획득하기 위해 FRUC로 리파인된다(S188). 현재 CU는 MV2로 움직임 보상된다(S189).

[0091] 프로세스는 종료된다(S190-1).

[0092] 움직임 예측기 유도 프로세스는 S180-2에서 시작된다. CUn으로 표시되는 이웃 CU는 예측을 위해 고려된다(S181). CUn이 현재 CU와 동일한 HU에 있는지가 체크된다. 예이면, 이때 리파인된 움직임 벡터는 예측을 위해 사용된다(S187). 그렇지 않으면, 리파인되지 않은 움직임 벡터는 사용된다(S185). 방법은 S190-2에서 종료된다.

[0093] 이러한 실시예에서, 사용되는 움직임 벡터 예측기는 CU의 페어런트 HU에 의존한다.

[0094] HU의 크기에 관한 및/또는 하나 이상의 플래그, 리스트들로부터의 선택들, 다른 표시기들과 같은, 값 전파에 대한 제약들에 관한, 예를 들어, FRUC 또는 IC의 제한에 관한 연관된 신택스는 예를 들어, 슬라이스, 화상 파라미터 세트(Picture Parameter Set)(PPS), 또는 시퀀스 파라미터 세트(Sequence Parameter Set)(SPS) 레벨들 중 하나 이상에서 시그널링될 수 있다. 다른 레벨들, 고레벨 신택스 또는 그 외의 것들이 다른 실시예들에서 사용될 수 있다. 이러한 시그널링을 위해 사용되는 연관된 신택스는 예를 들어, 하나 이상의 플래그, 리스트들로부터의 선택들, 다른 표시기들을 포함한다.

[0095] **CU 당 멀티 HU 독립 계산**

[0096] 도 6에 도시된 바와 같이, CU는 계산을 위해 수개의 HU로 분할될 수 있다. 이러한 경우에, 단일 신택스 요소는 CU 레벨에서 시그널링되지만 처리는 HU 당 행해진다. 그 다음, 계산이 CU 내부에 상이한 결과들을 초래할 수 있다는 점을 주목한다.

[0097] 이러한 처리에 영향을 받는 도구들의 예는 이하이다:

[0098] - DMVR;

[0099] - 양방향 모드에서의 LIC.

[0100] 이러한 경우에, 처리(예를 들어 DMVR의 양측 움직임 벡터 리파인먼트, 또는 조명 보상 파라미터 양방향성)는 각각의 HU에 대해 독립적으로 행해질 것이다.

[0101] 도 19는 CU가 2 HU로 분할되는 일 예를 도시한다: 고유 플래그는 CU를 위해 파싱되고 디코딩되며, 그 다음, CU 크기가 HU 크기보다 더 크면, 각각의 서브 HU는 독립 DMVR 처리를 실행한다.

[0102] S191에서 S201까지, DMVR 모드를 사용하는 CU는 CU 크기가 HU 크기보다 더 크면 수개의 HU로 분할된다. 그 다음, 발생된 HU는 독립 DMVR 프로세스를 사용한다(S199 및 S200).

[0103] 프로세스는 S191에서 시작된다. DMVR가 적용되는지를 표시하는 플래그가 디코딩된다(S192). DMVR이 적용되면, 이때 그것은 CU 크기가 HU 크기보다 더 큰 경우 체크된다(S194). 그렇지 않으면, 이때 DMVR은 CU(S198) 상에 적용된다. 예이면, CU는 더 작은 하드웨어 유닛들(또한 서브 HU로 칭함)(HU0 및 HU1)로 분할된다(S196). CU는 도 20에 도시된 바와 같이 2개보다 많은 서브 HU로 분할될 수 있다. 각각의 서브 HU는 독립 DMVR 처리를 실행한다(S199 및 S200). 프로세스는 S201에서 종료된다.

[0104] **HU 분할**

[0105] 서브 HU들로 CU의 분할은 수개의 방식으로 행해질 수 있다:

[0106] - 최대 HU 크기에 의해(둘다 수평으로 그리고 수직으로)에 의해 분할되고 마지막 HU들(수평으로 그리고/또는 수직으로)이 CU 외부에 있으면, 그들의 크기를 적응시킨다(도 20).

[0107] - 각각의 HU이 최대 크기보다 크기에 있어서 더 크지 않아야 하고 또한 4(최소 CU 크기)의 배수이어야 한다는 제약들을 고려하면, 동일한 수의 HU를 수평으로 그리고 수직으로 갖기 위해 CU를 분할한다.

[0108] 다양한 실시예들은 이하 중 하나 이상을 포함한다:

[0109] - HU 레벨에서 Tdep 내의 도구들의 사용의 제한 + HU의 인과 경계 상의 부가적 제약들.

[0110] - 신택스를 이러한 제약들에 적응시킨다.

[0111] - 프레임 상의 HU 토폴로지의 수개의 변형.

[0112] - 일부 도구들(DMVR, LIC bidir)의 시그널링 및 HU 처리를 적응시킨다.

[0113] 이러한 문서는 도구들, 특징들, 실시예들, 모델들, 접근법들 등을 포함하는, 다양한 양태들을 설명한다. 이러한 양태들의 다수는 구체적으로 설명되고, 개별 특성들을 적어도 나타내기 위해, 종종 제한할 수 있는 방식으로 설명된다. 그러나, 이것은 설명의 명료성의 목적들을 위한 것이고, 그러한 양태들의 적용 또는 범위를 제한하지 않는다. 실제로, 상이한 양태들의 전부는 추가 양태들을 제공하기 위해 조합되고 교환될 수 있다. 더욱이, 양태들은 또한 앞선 출원들에 설명되는 양태들과 조합되고 교환될 수 있다.

[0114] 이러한 문서에 설명되고 생각된 양태들은 많은 상이한 형태들로 구현될 수 있다. 도 10, 도 11 및 도 21은 일부 실시예들을 아래에 제공하지만, 다른 실시예들이 생각되고 도 10, 도 11 및 도 21의 논의는 구현들의 범위를 제한하지 않는다. 양태들 중 적어도 하나는 일반적으로 비디오 인코딩 및 디코딩에 관한 것이고, 적어도 하나의 다른 양태는 일반적으로 발생되거나 인코딩되는 비트스트림을 송신하는 것에 관한 것이다. 이러한 및 다른 양태들은 방법, 장치, 설명되는 방법들 중 어느 하나에 따라 비디오 데이터를 인코딩하거나 디코딩하기 위한 명령어들을 저장한 컴퓨터 판독 가능 저장 매체, 및/또는 설명되는 방법들 중 어느 하나에 따라 발생되는 비트스트림을 저장한 컴퓨터 판독 가능 저장 매체로서 구현될 수 있다.

[0115] 본 출원에서, 용어들 "재구성된" 및 "디코딩된"은 상호교환 가능하게 사용될 수 있고, 용어들 "픽셀" 및 "샘플"은 상호교환 가능하게 사용될 수 있고, 용어들 "이미지", "화상" 및 "프레임"은 상호교환 가능하게 사용될 수 있다. 반드시 필요한 것은 아닌, 통상적으로, 용어 "재구성된"은 인코더 측에 사용되는 한편 "디코딩된"은 디코더 측에 사용된다.

[0116] 다양한 방법들은 위에 설명되고, 방법들 각각은 설명된 방법을 달성하기 위한 하나 이상의 단계 또는 액션을 포함한다. 단계들 또는 액션들의 특정 순서가 방법의 적절한 동작을 위해 요구되지 않는 한, 특정 단계들 및/또

는 액션들의 순서 및/또는 사용은 수정되거나 조합될 수 있다.

[0117] 이러한 문서에 설명되는 다양한 방법들 및 다른 양태들은 예를 들어 도 10 및 도 11에 도시된 바와 같이 JVET 또는 HEVC 인코더(100) 및 디코더(200)의 움직임 보상 및 가능하게는 인트라 예측 모듈들(170, 275, 160, 260)과 같은 모듈들을 수정하기 위해 사용될 수 있다. 더욱이, 본 양태들은 JVET 또는 HEVC에 제한되지 않고, 예를 들어, 다른 표준들 및 권고들, 이전부터 존재하는지 또는 장래에 개발될지, 및 임의의 그러한 표준들 및 권고들(JVET 및 HEVC를 포함함)의 확장들에 적용될 수 있다. 달리 지시하지 않거나, 기술적으로 배제되지 않는 한, 이러한 문서에 설명되는 양태들은 개별적으로 또는 조합하여 사용될 수 있다.

[0118] 다양한 수치 값들은 본 문서에 제시될 수 있다. 특정 값들은 예시적 목적들을 위한 것이고 설명되는 양태들은 이러한 특정 값들에 제한되지 않는다.

[0119] 도 10은 예시적 인코더(100)를 예시한다. 이러한 인코더(100)의 변형들이 생각되지만, 인코더(100)는 모든 예상된 변형들을 설명하는 것 없이 명료성의 목적들을 위해 아래에 설명된다.

[0120] 인코딩되기 전에, 비디오 시퀀스는 프리 인코딩 처리(101), 예를 들어, 컬러 변환을 입력 컬러 화상에 적용하는 것(예를 들어, RGB 4:4:4로부터 YCbCr 4:2:0으로의 변환), 또는 압축에 더 회복력 있는 신호 분배를 얻기 위해 입력 화상 성분들의 재매핑을 수행하는 것(예를 들어 컬러 성분들 중 하나의 히스토그램 등화를 사용하는 것)을 거칠 수 있다. 메타데이터는 사전 처리와 연관되고, 비트스트림에 부착될 수 있다.

[0121] 예시적 인코더(100)에서, 화상은 아래에 설명되는 바와 같이 인코더 요소들에 의해 인코딩된다. 인코딩될 화상은 파티션(102)되고 예를 들어, CU들의 유닛들에서 처리된다. 각각의 유닛은 예를 들어, 인트라 또는 인터 모드를 사용하여 인코딩된다. 유닛이 인트라 모드에서 인코딩될 때, 그것은 인트라 예측(160)을 수행한다. 인터 모드에서, 움직임 추정(175) 및 보상(170)이 수행된다. 인코더는 유닛을 인코딩하기 위해 사용할 인트라 모드 또는 인터 모드 중 하나를 결정(105)하고, 예를 들어, 예측 모드 플래그에 의해 인트라/인터 결정을 표시한다. 예측 잔차들은 예를 들어, 원래의 이미지 블록으로부터 예측된 블록을 감산(110)함으로써 계산된다.

[0122] 그 다음, 예측 잔차들이 변환(125)되고 양자화(130)된다. 양자화된 변환 계수들뿐만 아니라, 움직임 벡터들 및 다른 신택스 요소들은 비트스트림을 출력하기 위해 엔트로 코딩(145)된다. 인코더는 변환을 스킵하고 비변환된 잔차 신호에 직접 양자화를 적용할 수 있다. 인코더는 변환 및 양자화 둘 다를 우회할 수 있으며, 즉 잔차는 변환 또는 양자화 프로세스들의 적용 없이 직접 코딩된다.

[0123] 인코더는 추가 예측들을 위한 참조를 제공하기 위해 인코딩된 블록을 디코딩한다. 양자화된 변환 계수들은 예측 잔차들을 디코딩하기 위해 역양자화(140)되고 역변환(150)된다. 디코딩된 예측 잔차들 및 예측된 블록을 조합(155)하면, 이미지 블록이 재구성된다. 루프 내 필터들(165)은 예를 들어, 인코딩 아티팩트들을 감소시키기 위한 디블로킹/샘플 적응 오프셋(Sample Adaptive Offset)(SAO) 필터링을 수행하기 위해 재구성된 화상에 적용된다. 필터링된 이미지는 참조 화상 버퍼(180)에 저장된다.

[0124] 도 11은 예시적 비디오 디코더(200)의 블록도를 예시한다. 예시적 디코더(200)에서, 비트스트림은 아래에 설명되는 바와 같이 디코더 요소들에 의해 디코딩된다. 비디오 디코더(200)는 일반적으로 도 10에 설명된 바와 같이 인코딩 패스에 역방향인 디코딩 패스를 수행한다. 인코더(100)는 또한 일반적으로 비디오 데이터 인코딩의 일부로서 비디오 디코딩을 수행한다.

[0125] 특히, 디코더의 입력은 비디오 비트스트림을 포함하며, 이 비트스트림은 비디오 인코더(100)에 의해 발생될 수 있다. 비트스트림은 변환 계수들, 움직임 벡터들, 및 다른 코딩된 정보를 획득하기 위해 첫째로 엔트로피 디코딩(230)된다. 화상 파티션 정보는 화상이 어떻게 파티션되는지를 표시한다. 따라서, 디코더는 디코딩된 화상 파티셔닝 정보에 따라 화상을 분할(235)할 수 있다. 변환 계수들은 예측 잔차들을 디코딩하기 위해 역양자화되고(240) 역변환(250)된다. 디코딩된 예측 잔차들 및 예측된 블록을 조합(255)하면, 이미지 블록이 재구성된다. 예측된 블록은 인트라 예측(260) 또는 움직임 보상된 예측(즉, 인터 예측)(275)으로부터 획득(270)될 수 있다. 루프 내 필터들(265)은 재구성된 이미지에 적용된다. 필터링된 이미지는 참조 화상 버퍼(280)에 저장된다.

[0126] 디코딩된 화상은 포스트 디코딩 처리(285), 예를 들어, 역 컬러 변환(예를 들어, YCbCr 4:2:0으로부터 RGB 4:4:4로의 변환) 또는 프리 인코딩 처리(101)에서 수행되는 재매핑 프로세스의 역을 수행하는 역 재매핑을 추가로 거칠 수 있다. 포스트 디코딩 처리는 프리 인코딩 처리에서 유도되고 비트스트림에서 시그널링되는 메타데이터를 사용할 수 있다.

[0127] 도 21은 다양한 양태들 및 예시적 실시예들이 구현되는 예시적 시스템의 블록도를 예시한다. 시스템(1000)은

아래에 설명되는 다양한 구성요소들을 포함하는 디바이스로서 구현될 수 있고 이러한 문서에 설명되는 양태들 중 하나 이상을 수행하도록 구성된다. 그러한 디바이스들의 예들은 개인용 컴퓨터들, 랩톱 컴퓨터들, 스마트폰들, 태블릿 컴퓨터들, 디지털 멀티미디어 셋톱 박스들, 디지털 텔레비전 수신기들, 개인용 비디오 레코딩 시스템들, 연결된 가정용 기기들, 및 서버들을 포함하지만, 이에 제한되지 않는다. 시스템(1000)은 이러한 문서에 설명되는 다양한 양태들을 구현하기 위해 도 21에 도시된 바와 같이 그리고 본 기술분야의 통상의 기술자들에 의해 공지된 바와 같이 통신 채널을 통해 다른 유사한 시스템들, 및 디스플레이에 통신 결합될 수 있다.

[0128] 시스템(1000)은 이러한 문서에 설명되는 다양한 양태들을 구현하기 위해 그 안에 로딩된 명령어들을 실행하도록 구성되는 적어도 하나의 프로세서(1010)를 포함할 수 있다. 프로세서(1010)는 본 기술분야에 공지된 바와 같이 내장 메모리, 입출력 인터페이스, 및 다양한 다른 회로들을 포함할 수 있다. 시스템(1000)은 적어도 하나의 메모리(1020)(예를 들어, 휘발성 메모리 디바이스, 비휘발성 메모리 디바이스)를 포함할 수 있다. 시스템(1000)은 EEPROM, ROM, PROM, RAM, DRAM, SRAM, 플래시, 자기 디스크 드라이브, 및/또는 광 디스크 드라이브를 포함하지만, 이에 제한되지 않는, 비휘발성 메모리를 포함할 수 있는 저장 디바이스(1040)를 포함할 수 있다. 저장 디바이스(1040)는 비제한적 예들로서, 내부 저장 디바이스, 부착 저장 디바이스, 및/또는 네트워크 액세스가능 저장 디바이스를 포함할 수 있다. 시스템(1000)은 인코딩된 비디오 또는 디코딩된 비디오를 제공하기 위해 데이터를 처리하도록 구성되는 인코더/디코더 모듈(1030)을 포함할 수 있고, 인코더/디코더 모듈(1030)은 그 자체의 프로세서 및 메모리를 포함할 수 있다.

[0129] 인코더/디코더 모듈(1030)은 인코딩 및/또는 디코딩 기능들을 수행하기 위해 디바이스에 포함될 수 있는 모듈(들)을 표현한다. 공지된 바와 같이, 디바이스는 인코딩 및 디코딩 모듈들 중 하나 또는 둘 다를 포함할 수 있다. 추가적으로, 인코더/디코더 모듈(1030)은 시스템(1000)의 개별적 요소로서 구현될 수 있거나 본 기술분야의 통상의 기술자들에게 공지된 바와 같이 하드웨어 및 소프트웨어의 조합으로서 프로세서들(1010) 내에 통합될 수 있다.

[0130] 이러한 문서에 설명되는 다양한 양태들을 수행하기 위해 프로세서들(1010) 상에 로딩될 프로그램 코드는 저장 디바이스(1040)에 저장되고 프로세서들(1010)에 의한 실행을 위해 메모리(1020) 상에 나중에 로딩될 수 있다. 예시적 실시예들에 따르면, 프로세서(들)(1010), 메모리(1020), 저장 디바이스(1040), 및 인코더/디코더 모듈(1030) 중 하나 이상은 이러한 문서에 설명되는 프로세스들의 수행 동안, 입력 비디오, 디코딩된 비디오, 비트스트림, 수학식들, 공식들, 매트릭스들, 변수들, 연산들, 및 연산 로직을 포함하지만, 이에 제한되지 않는, 다양한 아이템들 중 하나 이상을 저장할 수 있다.

[0131] 시스템(1000)은 통신 채널(1060)을 통해 다른 디바이스들과의 통신을 가능하게 하는 통신 인터페이스(1050)를 포함할 수 있다. 통신 인터페이스(1050)는 통신 채널(1060)로부터 데이터를 송신하고 수신하도록 구성되는 송수신기를 포함할 수 있지만, 이에 제한되지 않는다. 통신 인터페이스(1050)는 모뎀 또는 네트워크 카드를 포함할 수 있지만, 이에 제한되지 않고 통신 채널은 유선 및/또는 무선 매체 내에 구현될 수 있다. 시스템(1000)의 다양한 구성요소들은 내부 버스들, 와이어들, 및 인쇄 회로 보드들을 포함하지만, 이에 제한되지 않는 다양한 적절한 연결들을 사용하여 전부 연결되거나 함께 통신 결합될 수 있다.

[0132] 예시적 실시예들은 프로세서(1010)에 의해 또는 하드웨어에 의해 구현되는 컴퓨터 소프트웨어에 의해, 또는 하드웨어 및 소프트웨어의 조합에 의해 수행될 수 있다. 비제한적 예로서, 예시적 실시예들은 하나 이상의 집적 회로에 의해 구현될 수 있다. 메모리(1020)는 기술적 환경에 적절한 임의의 타입일 수 있고, 비제한적 예들로서, 광학 메모리 디바이스들, 자기 메모리 디바이스들, 반도체 기반 메모리 디바이스들, 고정 메모리, 및 제거식 메모리와 같은 임의의 적절한 데이터 저장 기술을 사용하여 구현될 수 있다. 프로세서(1010)는 기술적 환경에 적절한 임의의 타입일 수 있고, 비제한적 예들로서, 마이크로프로세서들, 일반 목적 컴퓨터들, 특수 목적 컴퓨터들, 및 멀티 코어 아키텍처 기반 프로세서들 중 하나 이상을 포함할 수 있다.

[0133] 여기서 설명되는 구현들 및 양태들은 예를 들어, 방법 또는 프로세스, 장치, 소프트웨어 프로그램, 데이터 스트림, 또는 신호로 구현될 수 있다. 단일 형태의 구현의 맥락에서만 논의되더라도(예를 들어, 방법으로서만 논의됨), 논의되는 특징들의 구현은 또한 다른 형태들(예를 들어, 장치 또는 프로그램)로 구현될 수 있다. 장치는 예를 들어, 적절한 하드웨어, 소프트웨어, 및 펌웨어로 구현될 수 있다. 방법들은 예를 들어, 컴퓨터, 마이크로프로세서, 집적 회로, 또는 프로그래밍가능 로직 디바이스를 포함하는 처리 디바이스들을 일반적으로 지칭하는, 예를 들어, 프로세서와 같은 예를 들어, 장치로 구현될 수 있다. 프로세서들은 또한 예를 들어, 컴퓨터들, 휴대 전화들, 휴대용/개인 정보 단말기들(portable/personal digital assistants)("PDAs"), 및 최종 사용자들 사이의 정보의 통신을 용이하게 하는 다른 디바이스들과 같은, 통신 디바이스들을 포함한다.

[0134] "하나의 실시예" 또는 "일 실시예" 또는 "하나의 구현" 또는 "일 구현"뿐만 아니라, 그의 다른 변형들에 대한 참조는 실시예와 관련하여 설명되는 특정 특징, 구조, 특성 등이 적어도 하나의 실시예에 포함되는 것을 의미한다. 따라서, 이러한 문서 전반의 다양한 장소들에 나타나는 구 "하나의 실시예에서" 또는 "일 실시예에서" 또는 "하나의 구현에서" 또는 "일 구현에서"뿐만 아니라, 임의의 다른 변형들의 출현들은 반드시 동일한 실시예를 모두 지칭하는 것은 아니다.

[0135] 추가적으로, 이러한 문서는 다양한 정보를 "결정"하는 것을 언급할 수 있다. 정보를 결정하는 것은 예를 들어, 정보를 추정하는 것, 정보를 계산하는 것, 정보를 예측하는 것, 또는 메모리로부터 정보를 검색하는 것 중 하나 이상을 포함할 수 있다.

[0136] 게다가, 이러한 문서는 다양한 정보에 "액세스"하는 것을 언급할 수 있다. 정보에 액세스하는 것은 예를 들어, 정보를 수신하는 것, (예를 들어, 메모리로부터) 정보를 검색하는 것, 정보를 저장하는 것, 정보를 처리하는 것, 정보를 송신하는 것, 정보를 이동시키는 것, 정보를 복사하는 것, 정보를 소거하는 것, 정보를 계산하는 것, 정보를 결정하는 것, 정보를 예측하는 것, 또는 정보를 추정하는 것 중 하나 이상을 포함할 수 있다.

[0137] 추가적으로, 이러한 문서는 다양한 정보를 "수신"하는 것을 언급할 수 있다. 수신은 "액세스"에서와 같이, 넓은 용어이도록 의도된다. 정보를 수신하는 것은 예를 들어, 정보에 액세스하는 것, 또는 정보(예를 들어, 메모리로부터) 정보를 검색하는 것 중 하나 이상을 포함할 수 있다. 게다가, "수신"은 전형적으로 예를 들어, 정보를 저장하는 것, 정보를 처리하는 것, 정보를 송신하는 것, 정보를 이동시키는 것, 정보를 복사하는 것, 정보를 소거하는 것, 정보를 계산하는 것, 정보를 결정하는 것, 정보를 예측하는 것, 또는 정보를 추정하는 것과 같은 동작들 동안, 하나의 방식으로 또는 다른 방식으로 수반된다.

[0138] 본 기술분야의 통상의 기술자에게 명백한 바와 같이, 구현들은 예를 들어, 저장되거나 송신될 수 있는 정보를 반송하도록 포맷되는 다양한 신호들을 생성할 수 있다. 정보는 예를 들어, 방법을 수행하기 위한 명령어들, 또는 설명된 구현들 중 하나에 의해 생성되는 데이터를 포함할 수 있다. 예를 들어, 신호는 설명된 실시예의 비트스트림을 반송하도록 포맷될 수 있다. 그러한 신호는 예를 들어, (예를 들어, 스펙트럼의 라디오 주파수 부분을 사용하여) 전자기 파로서 또는 기저대역 신호로서 포맷될 수 있다. 포맷은 예를 들어, 데이터 스트림을 인코딩하고 인코딩된 데이터 스트림으로 반송파를 변조하는 것을 포함할 수 있다. 신호가 반송하는 정보는 예를 들어, 아날로그 또는 디지털 정보일 수 있다. 신호는 공지된 바와 같이, 여러 상이한 유선 또는 무선 링크들을 통해 송신될 수 있다. 신호는 프로세서 판독 가능 매체 상에 저장될 수 있다.

[0139] 다수의 실시예를 설명했다. 이러한 실시예들은 여러 상이한 청구항 카테고리들 및 타입들에 걸쳐, 모든 조합들을 포함하는, 이하의 일반화된 발명들 및 청구항들을 적어도 제공한다:

[0140] ● 코딩 및/또는 디코딩 도구들에 의해 생성되는 데이터 종속을 완화하거나, 감소시키거나, 그렇지 않으면 수정한다:

[0141] o 도구들은 FRUC, DMVR, LIC, BIO를 포함한다.

[0142] o 데이터 종속은 디코딩되는 블록(현재 블록)과 이웃 블록, 예를 들어 CU 사이의 종속이다.

[0143] ● HU 레벨에서 일부 도구들의 사용을 제한하며, 도구들은 FRUC, DMVR, LIC, BIO를 포함한다.

[0144] ● 이웃 블록에 기초하여 HU 레벨에서 일부 도구들의 사용을 제한한다.

[0145] ● 적어도 하나의 이웃 블록(예를 들어, 상단 및/또는 좌측 이웃 블록)이 현재 블록과 동일한 페어런트 HU를 공유하는 경우에 도구들의 일부를 사용한다.

[0146] ● 이웃 블록은 현재 블록의 상단 및 좌측 블록이다.

[0147] ● 일부 도구들, 예를 들어 DMVR 및 LIC bidir의 HU 처리를 적응시킨다.

[0148] ● HU들 사이의 예측 값들을 제한한다.

[0149] ● 설명된 신택스 요소들, 또는 그의 변형들 중 하나 이상을 포함하는 비트스트림 또는 신호.

[0150] ● 디코더가 인코더에 의해 사용되는 것에 대응하는 방식으로 비트스트림을 처리할 수 있게 하는 시그널링 신택스 요소들에 삽입한다.

[0151] ● 설명된 신택스 요소들, 또는 그의 변형들 중 하나 이상을 포함하는 비트스트림 또는 신호를 생성 및/또는 송

신 및/또는 수신 및/또는 디코딩한다.

[0152] ● 설명되는 실시예들 중 어느 하나를 수행하는 TV, 셋톱 박스, 휴대 전화, 태블릿, 또는 다른 전자 디바이스.

[0153] ● 설명되는 실시예들 중 어느 하나를 수행하고, (예를 들어, 모니터, 스크린, 또는 다른 타입의 디스플레이를 사용하여) 결과적인 이미지를 디스플레이하는 TV, 셋톱 박스, 휴대 전화, 태블릿, 또는 다른 전자 디바이스.

[0154] ● 인코딩된 이미지를 포함하는 신호를 수신하기 위해 (예를 들어, 튜너를 사용하여) 채널을 동조시키고, 설명되는 실시예들 중 어느 하나를 수행하는 TV, 셋톱 박스, 휴대 전화, 태블릿, 또는 다른 전자 디바이스.

[0155] ● 인코딩된 이미지를 포함하는 신호를 무선으로 (예를 들어, 안테나를 사용하여) 수신하고, 설명되는 실시예들 중 어느 하나를 수행하는 TV, 셋톱 박스, 휴대 전화, 태블릿, 또는 다른 전자 디바이스.

[0156] 다양한 다른 일반화될 뿐만 아니라, 특정화된 발명들 및 청구항들은 또한 본 개시 전반에서 지지되고 생각된다.

[0157] 디코딩 방법이 개시되며, 방법은,

[0158] - 화상의 현재 블록이 페어런트 하드웨어 유닛을 갖는 경우에만 현재 블록이 도구들의 세트 중의 도구로 디코딩되는지를 표시하는 플래그를 디코딩하는 단계 - 도구들의 세트는 현재 블록 전에 디코딩되는 이웃 블록들의 재구성된 샘플들을 사용하거나 하드웨어 유닛보다 더 큰 화상의 블록을 사용하는 도구들을 포함함 - ; 및

[0159] - 플래그에 응답하여 현재 블록을 디코딩하는 단계를 포함한다.

[0160] 적어도 하나의 프로세서를 포함하는 디코딩 장치가 개시되며, 프로세서는,

[0161] - 화상의 현재 블록이 페어런트 하드웨어 유닛을 갖는 경우에만 현재 블록이 도구들의 세트 중의 도구로 디코딩되는지를 표시하는 플래그를 디코딩하는 것 - 도구들의 세트는 현재 블록 전에 디코딩되는 이웃 블록들의 재구성된 샘플들을 사용하거나 하드웨어 유닛보다 더 큰 화상의 블록을 사용하는 도구들을 포함함 - ; 및

[0162] - 플래그에 응답하여 현재 블록을 디코딩하는 것을 수행하도록 구성된다.

[0163] 인코딩 방법이 개시되며, 방법은,

[0164] - 화상의 현재 블록이 페어런트 하드웨어 유닛을 갖는 경우에만 현재 블록이 도구들의 세트 중의 도구로 인코딩되는지를 표시하는 플래그를 인코딩하는 단계 - 도구들의 세트는 현재 블록 전에 인코딩되는 이웃 블록들의 재구성된 샘플들을 사용하거나 하드웨어 유닛보다 더 큰 화상의 블록을 사용하는 도구들을 포함함 - ; 및

[0165] - 플래그에 응답하여 현재 블록을 인코딩하는 단계를 포함한다.

[0166] 적어도 하나의 프로세서를 포함하는 인코딩 장치로서, 프로세서는,

[0167] - 화상의 현재 블록이 페어런트 하드웨어 유닛을 갖는 경우에만 현재 블록이 도구들의 세트 중의 도구로 인코딩되는지를 표시하는 플래그를 인코딩하는 것 - 도구들의 세트는 현재 블록 전에 인코딩되는 이웃 블록들의 재구성된 샘플들을 사용하거나 하드웨어 유닛보다 더 큰 화상의 블록을 사용하는 도구들을 포함함 - ; 및

[0168] - 플래그에 응답하여 현재 블록을 인코딩하는 것을 수행하도록 구성된다.

[0169] 신호가 개시되며, 신호는,

[0170] - 화상의 현재 블록이 페어런트 하드웨어 유닛을 갖는 경우에만 현재 블록이 도구들의 세트 중의 도구로 인코딩되는지를 표시하는 플래그 - 도구들의 세트는 현재 블록 전에 인코딩되는 이웃 블록들의 재구성된 샘플들을 사용하거나 하드웨어 유닛보다 더 큰 화상의 블록을 사용하는 도구들을 포함함 - ; 및

[0171] - 플래그에 응답하여 코딩되는 현재 블록을 나타내는 데이터를 포함한다.

[0172] 디바이스가 개시되며, 디바이스는,

[0173] 개시되는 실시예들 중 어느 하나에 따른 디코딩 장치; 및

[0174] (i) 무선으로 신호를 수신하도록 구성되는 안테나 - 신호는 이미지를 갖는 비디오를 포함함 - , (ii) 이미지를 갖는 비디오를 포함하는 수신된 신호를 주파수들의 대역으로 제한하도록 구성되는 대역 제한기, 또는 (iii) 출력을 디스플레이하도록 구성되는 디스플레이 중 적어도 하나를 포함한다.

[0175] 컴퓨터 프로그램 제품이 개시되며, 이 제품은 프로그램이 하나 이상의 프로세서에 의해 실행될 때, 하나 이상의 프로세서로 하여금 개시되는 실시예들 중 어느 하나의 디코딩 또는 인코딩 방법을 수행하게 하는 명령어들을 포

함한다.

[0176] 일 실시예에서, 플래그는 현재 블록이 페어런트 하드웨어 유닛을 갖고 적어도 하나의 이웃 블록이 현재 블록과 동일한 페어런트 하드웨어 유닛 내부에 위치되는 경우에 디코딩(각각 인코딩)된다.

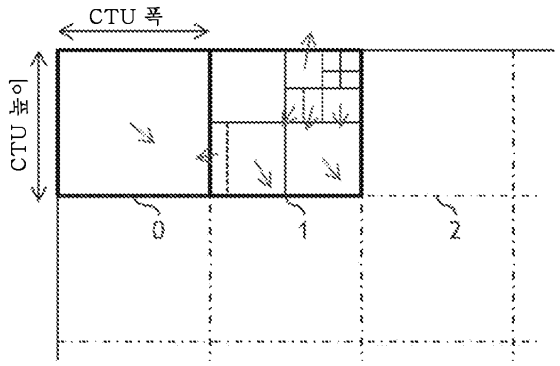
[0177] 일 실시예에서, 도구들의 세트는 이하의 도구들을 포함한다: 프레임 레이트 업 변환, 국부 조명 보상, 디코더 움직임 벡터 리파인먼트 및 양방향 광학적 흐름.

[0178] 일 실시예에서, 도구는 템플릿을 사용하며, 현재 블록과 동일한 페어런트 하드웨어 유닛 내부에 위치되는 템플릿의 일부만이 사용된다.

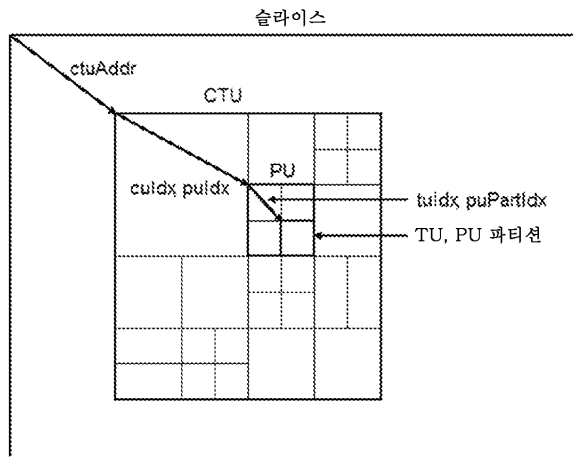
[0179] 일 실시예에서, 도구는 리파인먼트를 갖는 움직임 벡터 예측기를 사용하며, 움직임 벡터 리파인먼트는 현재 블록 및 이웃 블록이 동일한 하드웨어 유닛 내에 위치되는 경우에 움직임 벡터 예측기에 추가되고 움직임 벡터 리파인먼트는 현재 블록 및 이웃 블록이 상이한 하드웨어 유닛들 내에 위치되는 경우에 움직임 벡터 예측기에 추가되지 않는다.

도면

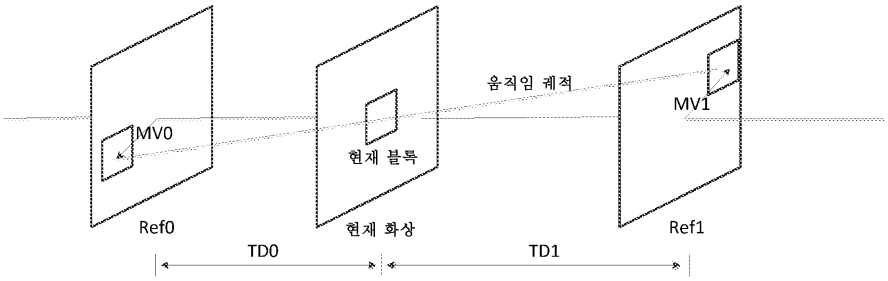
도면1



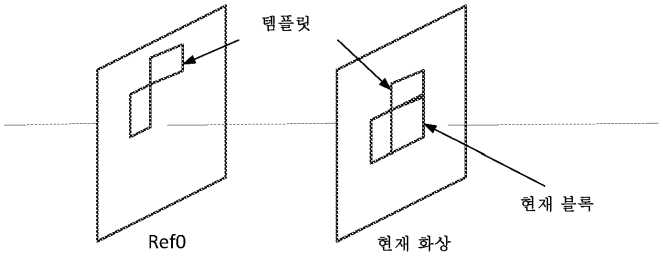
도면2



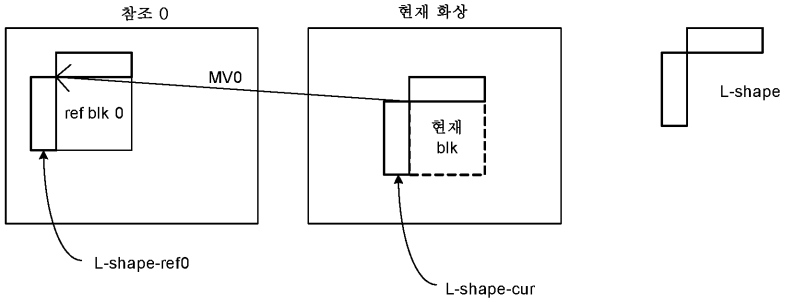
도면3



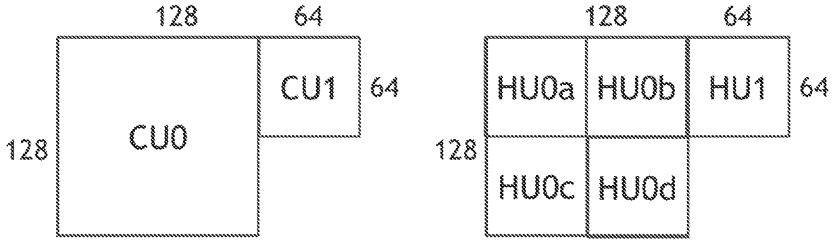
도면4



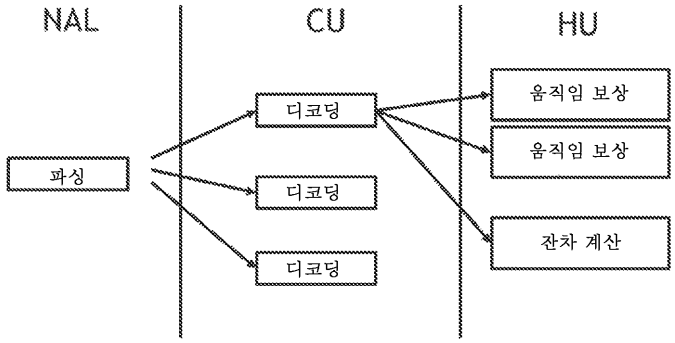
도면5



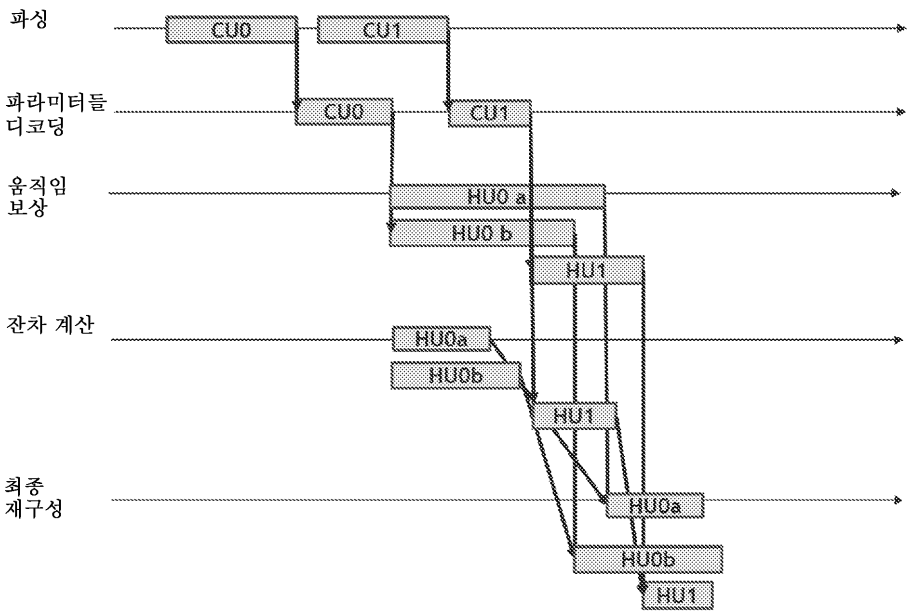
도면6



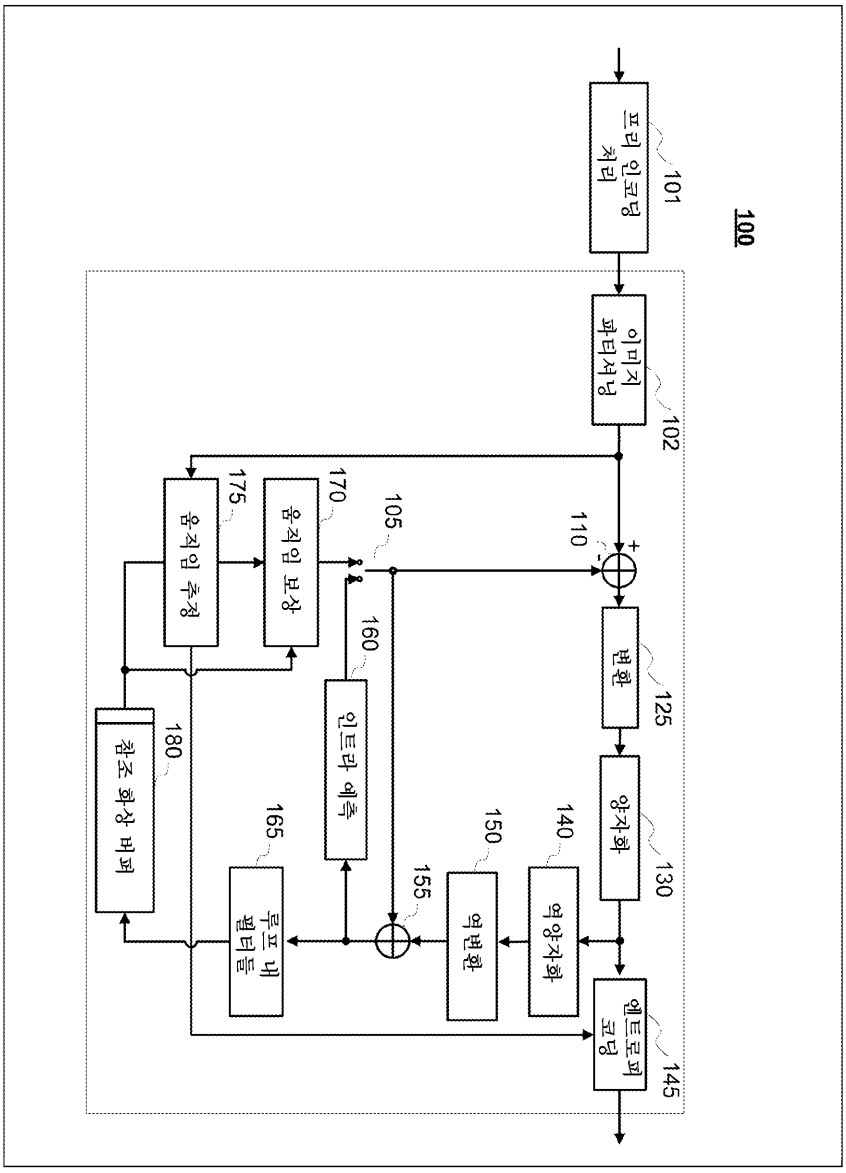
도면7


NAL
CU
HU
파싱
디코딩
디코딩
디코딩
움직임 보상
움직임 보상
잔차 계산

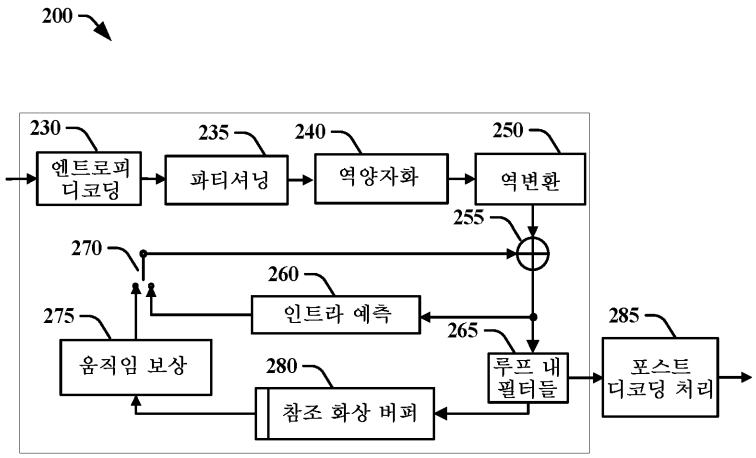
도면8


파싱
CU0
CU1
파라미터들 디코딩
CU0
CU1
움직임 보상
HU0 a
HU0 b
HU1
잔차 계산
HU0a
HU0b
HU1
최종 재구성
HU0a
HU0b
HU1

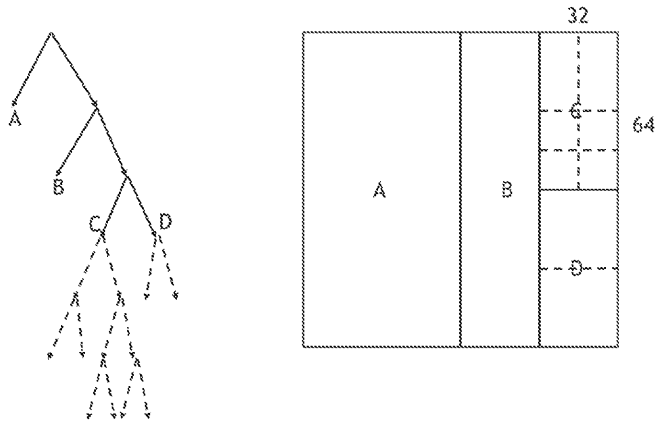
도면9


100
프리 인코딩 처리
101
이미지 파티셔닝
102
움직임 추정
175
움직임 보상
170
105
110
+
-
변환
125
양자화
130
엔트로피 코딩
145
역양자화
140
역변환
150
155
인트라 예측
160
루프 내 필터들
165
참조 화상 버퍼
180

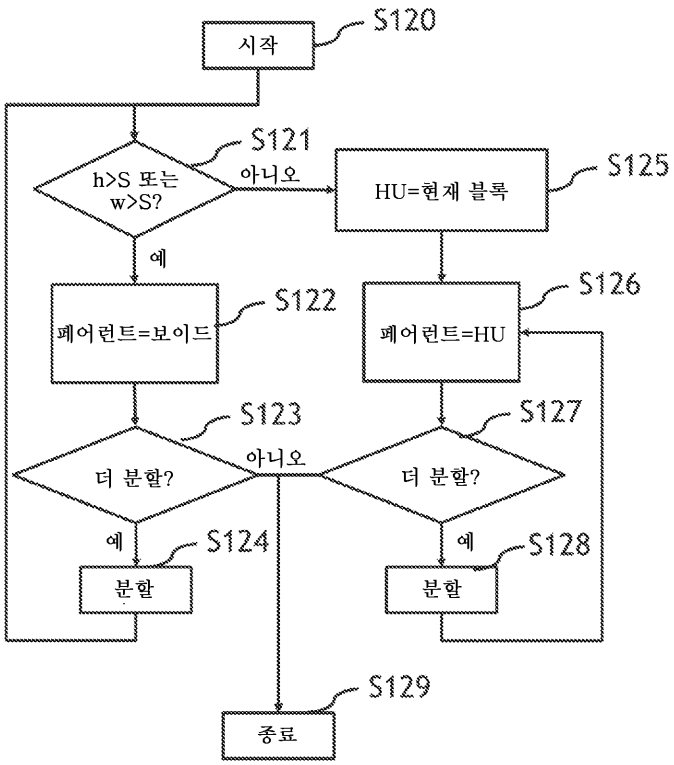
도면10


200
230
엔트로피 디코딩
235
파티셔닝
240
역양자화
250
역변환
255
270
260
인트라 예측
275
움직임 보상
265
280
참조 화상 버퍼
루프 내 필터들
285
포스트 디코딩 처리

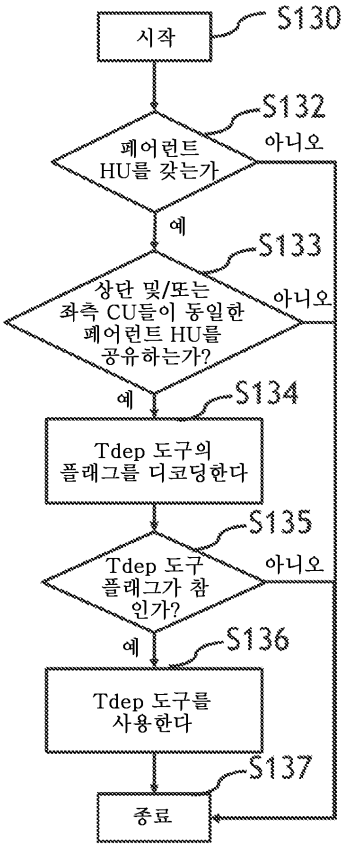
도면11


A
B
C
D
32
64
A
B
C
D

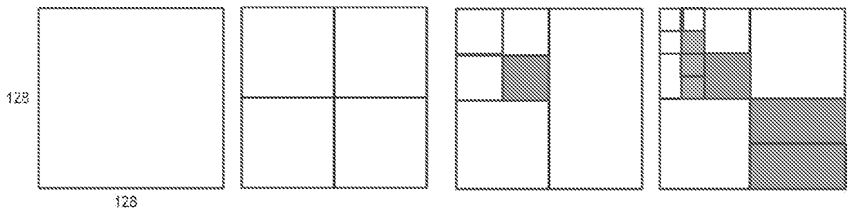
도면12


시작
S120
h>S 또는 w>S?
S121
아니오
HU=현재 블록
S125
예
페어런트=보이드
S122
페어런트=HU
S126
더 분할?
S123
아니오
더 분할?
S127
예
분할
S124
예
분할
S128
종료
S129

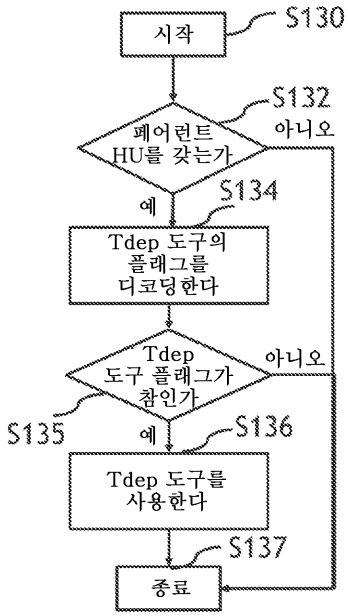
도면13


시작
S130
S132
페어런트 HU를 갖는가
아니오
예
S133
상단 및/또는 좌측 CU들이 동일한 페어런트 HU를 공유하는가?
아니오
예
S134
Tdep 도구의 플래그를 디코딩한다
S135
Tdep 도구 플래그가 참인가?
아니오
예
S136
Tdep 도구를 사용한다
S137
종료

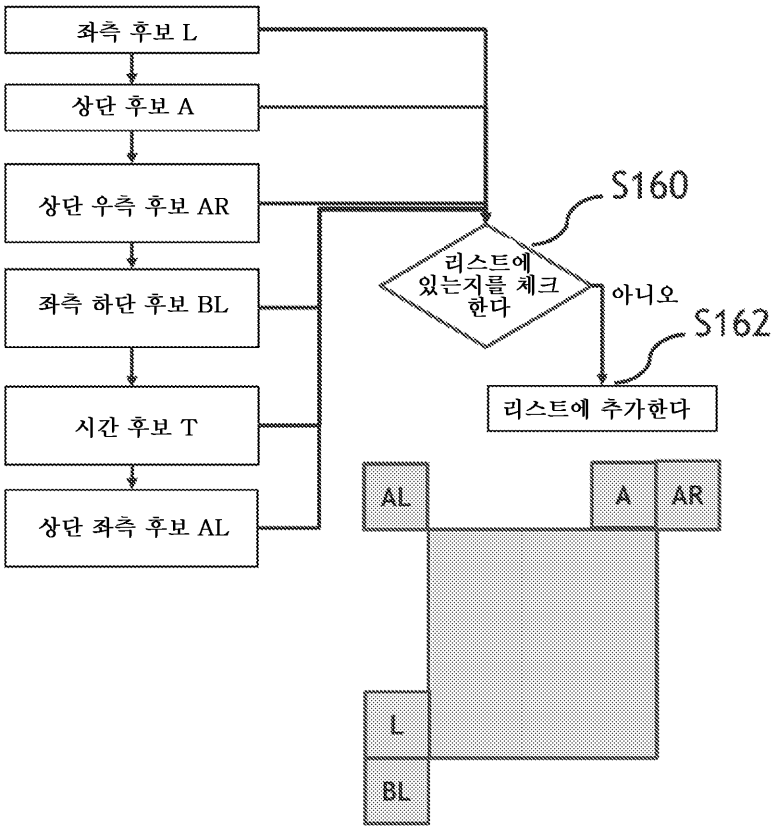
도면14


128
128

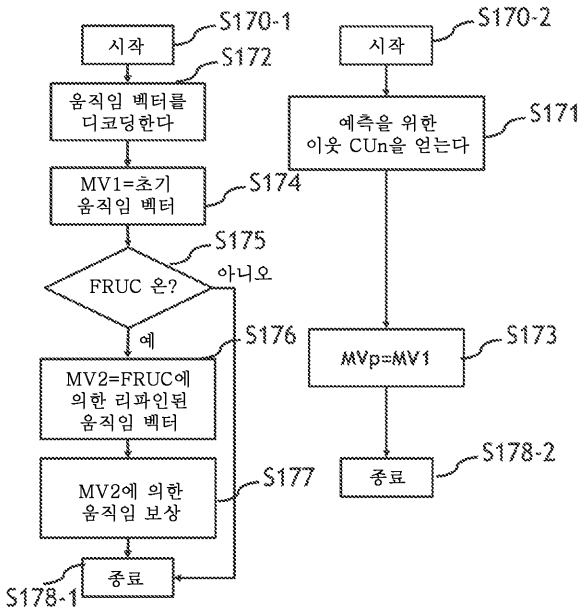
도면15


시작
S130
S132
페어런트 HU를 갖는가
아니오
예
S134
Tdep 도구의 플래그를 디코딩한다
Tdep 도구 플래그가 참인가
아니오
S135
예
S136
Tdep 도구를 사용한다
S137
종료

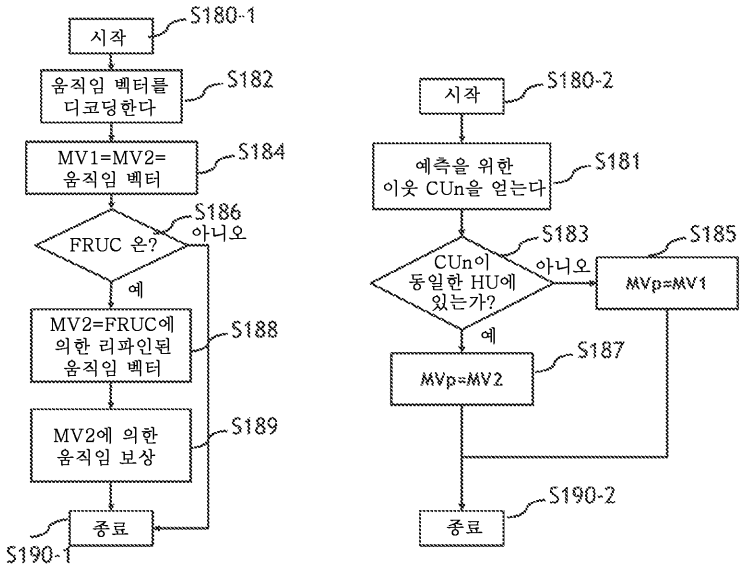
도면16


좌측 후보 L
상단 후보 A
상단 우측 후보 AR
좌측 하단 후보 BL
시간 후보 T
상단 좌측 후보 AL
S160
리스트에 있는지를 체크한다
아니오
S162
리스트에 추가한다
AL
A
AR
L
BL

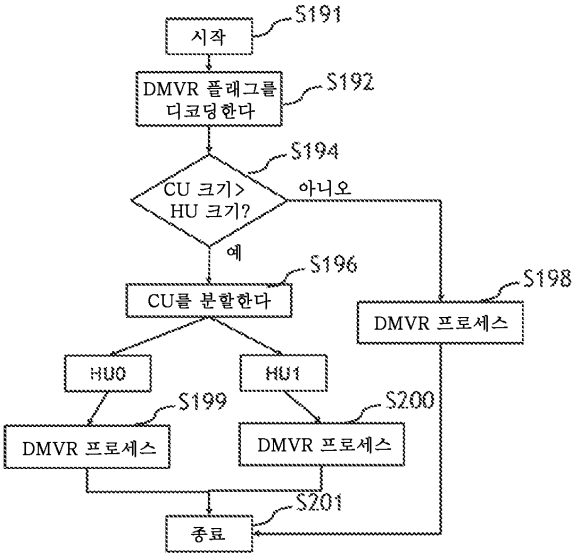
도면17


시작
S170-1
움직임 벡터를 디코딩한다
S172
MV1=초기 움직임 벡터
S174
FRUC 온?
S175
아니오
예
MV2=FRUC에 의한 리파인된 움직임 벡터
S176
MV2에 의한 움직임 보상
S177
종료
S178-1
시작
S170-2
예측을 위한 이웃 CUn을 얻는다
S171
MVp=MV1
S173
종료
S178-2

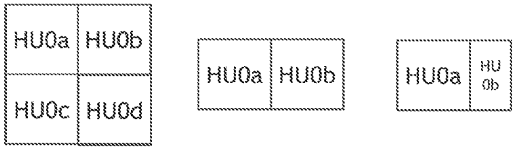
도면18


시작
S180-1
움직임 벡터를 디코딩한다
S182
MV1=MV2= 움직임 벡터
S184
FRUC 온?
S186
아니오
예
MV2=FRUC에 의한 리파인된 움직임 벡터
S188
MV2에 의한 움직임 보상
S189
종료
S190-1
시작
S180-2
예측을 위한 이웃 CUn을 얻는다
S181
CUn이 동일한 HU에 있는가?
S183
아니오
S185
MVp=MV1
예
MVp=MV2
S187
종료
S190-2

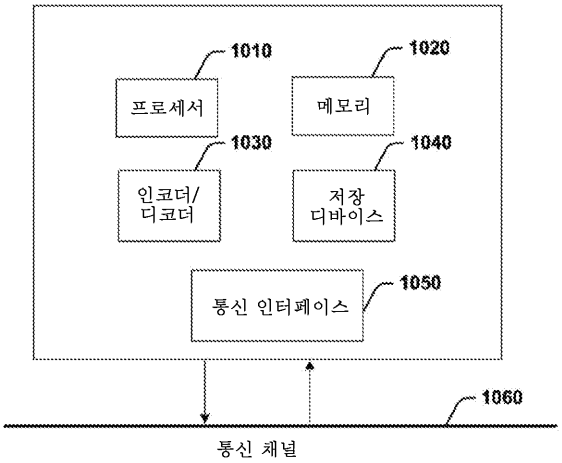
도면19



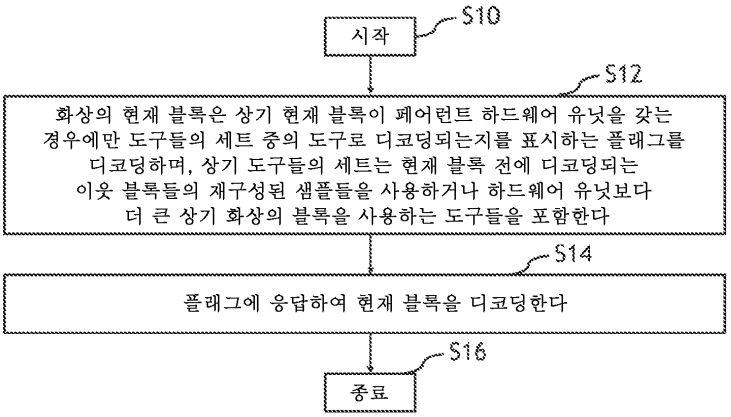
도면20



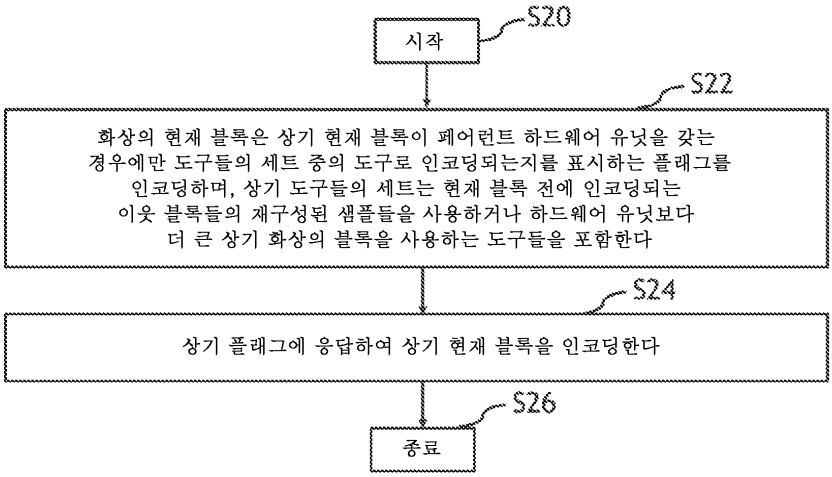
도면21



도면22


시작
S10
S12
화상의 현재 블록은 상기 현재 블록이 페어런트 하드웨어 유닛을 갖는
경우에만 도구들의 세트 중의 도구로 디코딩되는지를 표시하는 플래그를
디코딩하며, 상기 도구들의 세트는 현재 블록 전에 디코딩되는
이웃 블록들의 재구성된 샘플들을 사용하거나 하드웨어 유닛보다
더 큰 상기 화상의 블록을 사용하는 도구들을 포함한다
S14
플래그에 응답하여 현재 블록을 디코딩한다
S16
종료

도면23


시작
S20
S22
화상의 현재 블록은 상기 현재 블록이 페어런트 하드웨어 유닛을 갖는
경우에만 도구들의 세트 중의 도구로 인코딩되는지를 표시하는 플래그를
인코딩하며, 상기 도구들의 세트는 현재 블록 전에 인코딩되는
이웃 블록들의 재구성된 샘플들을 사용하거나 하드웨어 유닛보다
더 큰 상기 화상의 블록을 사용하는 도구들을 포함한다
S24
상기 플래그에 응답하여 상기 현재 블록을 인코딩한다
S26
종료