



공개특허 10-2022-0138017

(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)(11) 공개번호 10-2022-0138017  
(43) 공개일자 2022년10월12일

- (51) 국제특허분류(Int. Cl.)  
*H04N 19/11* (2014.01) *H04N 19/176* (2014.01)  
*H04N 19/184* (2014.01) *H04N 19/186* (2014.01)  
*H04N 19/44* (2014.01) *H04N 19/593* (2014.01)
- (52) CPC특허분류  
*H04N 19/11* (2015.01)  
*H04N 19/176* (2015.01)
- (21) 출원번호 10-2022-7034005(분할)
- (22) 출원일자(국제) 2021년01월13일  
심사청구일자 없음
- (62) 원출원 특허 10-2021-7032969  
원출원일자(국제) 2021년01월13일  
심사청구일자 2021년10월13일
- (85) 번역문제출일자 2022년09월29일
- (86) 국제출원번호 PCT/US2021/013264
- (87) 국제공개번호 WO 2021/150407  
국제공개일자 2021년07월29일
- (30) 우선권주장  
62/963,216 2020년01월20일 미국(US)  
17/097,415 2020년11월13일 미국(US)
- (71) 출원인  
텐센트 아메리카 엘엘씨  
미국 94306 캘리포니아주 팔로 알토 파크 블러바드 2747
- (72) 발명자  
수 샤오종  
미국 94306 캘리포니아주 팔로 알토 2747 파크 불러바드 텐센트 아메리카 엘엘씨 내  
리우 산  
미국 94306 캘리포니아주 팔로 알토 2747 파크 불러바드 텐센트 아메리카 엘엘씨 내
- (74) 대리인  
유미특허법인

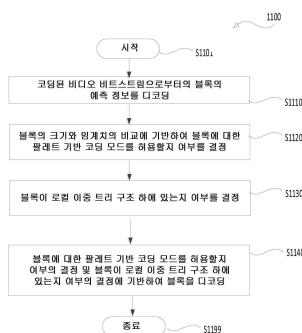
전체 청구항 수 : 총 1 항

(54) 발명의 명칭 로컬 이중 트리 구조 하에서 팔레트 기반 코딩 모드를 위한 방법 및 장치

### (57) 요 약

본 개시내용의 양태들은 비디오 인코딩/디코딩을 위한 방법들 및 장치들을 제공한다. 일부 예들에서, 비디오 디코딩을 위한 장치는 수신 회로 및 프로세싱 회로를 포함한다. 예를 들어, 프로세싱 회로는 코딩된 비디오 비트스트림으로부터 블록의 예측 정보를 디코딩하고, 블록의 크기와 임계치의 비교에 기반하여 블록에 대한 팔레트 기반 코딩 모드(palette based coding mode)를 허용할지 여부를 결정하고, 블록이 예측 정보에 기반하여 로컬 이중 트리 구조 하에 있는지 여부를 결정하고, 또한, 프로세싱 회로는 블록에 대한 팔레트 기반 코딩 모드를 허용할지 여부의 결정 및 블록이 로컬 이중 트리 구조 하에 있는지 여부의 결정에 기반하여 블록을 디코딩한다.

### 대 표 도



(52) CPC특허분류

*H04N 19/184* (2015.01)

*H04N 19/186* (2015.01)

*H04N 19/44* (2015.01)

*H04N 19/593* (2015.01)

---

## 명세서

### 청구범위

#### 청구항 1

디코더에 의해 수행되는 비디오 디코딩 방법으로서, 코딩된 비디오 비트스트림으로부터 블록의 예측 정보를 디코딩하는 단계; 상기 블록의 크기가 임계치 보다 작은지 여부에 기초하여 제1 결정을 실행하는 단계; 상기 예측 정보에 따라서, 상기 블록이 로컬 이중 트리 구조 하에 있는지 여부 및 상기 블록이 크로마 블록인지 여부에 기초하여 제2 결정을 실행하는 단계; 및 상기 제1 결정 및 상기 제2 결정에 기초하여, 상기 블록에 대한 팔레트 기반 코딩 모드(palette based coding mode)를 허용할지 여부를 결정하는 단계 를 포함하는 비디오 디코딩 방법.

### 발명의 설명

#### 기술 분야

[0001] 본 출원은 2020년 1월 20일에 출원되고 발명의 명칭이 "SIMPLIFIED PALETTE MODE CODING WITH LOCAL DUAL TREE CODING STRUCTURE"인 미국 가출원 제62/963,216호에 대한 우선권의 이익을 주장하는, 2020년 11월 13일에 출원되고 발명의 명칭이 "METHOD AND APPARATUS FOR PALETTE BASED CODING MODE UNDER LOCAL DUAL TREE STRUCTURE"인 미국 특허 출원 제17/097,415호에 대한 우선권의 이익을 주장한다. 종래 출원들의 전체 개시내용은 그 전체가 참조로 본원에 포함된다.

[0002] 본 개시내용은 일반적으로 비디오 코딩과 관련된 실시예들을 설명한다.

### 배경기술

[0003] 본원에 제공된 배경 설명은 본 개시내용의 맥락을 일반적으로 제시하기 위한 것이다. 출원 당시 종래 기술로 자격이 되지 않는 설명의 양태들뿐 아니라, 현재 명명된 발명가들의 작업은 본 배경 섹션에 설명된 범위 내에서, 명시적으로나 묵시적으로 본 개시내용에 대한 종래 기술로 인정되지 않는다.

[0004] 비디오 코딩 및 디코딩은 모션 보상을 갖는 인터-화상 예측을 사용하여 수행될 수 있다. 압축해제된 디지털 비디오는 일련의 화상들을 포함할 수 있고, 각각의 화상은 예를 들어  $1920 \times 1080$  휴도 샘플들 및 연관된 색차 샘플들의 공간 차원을 갖는다. 일련의 화상들은 예를 들어 초당 60개의 화상들 또는 60 Hz의 고정 또는 가변 화상 속도(또한 비공식적으로는 프레임 레이트로 알려짐)를 가질 수 있다. 압축해제된 비디오는 특정 비트레이트 요건들을 갖는다. 예를 들어 샘플당 8 비트의 1080p60 4:2:0 비디오(60 Hz 프레임 레이트에서  $1920 \times 1080$  휴도 샘플 해상도)는 1.5 Gbit/s에 가까운 대역폭을 요구한다. 이러한 비디오의 1 시간은 600 GBytes 초과의 저장 공간을 요구한다.

[0005] 비디오 코딩 및 디코딩의 하나의 목적은 압축을 통한 입력 비디오 신호의 중복 감소일 수 있다. 압축은 전술된 대역폭 및/또는 저장 공간 요건들을 감소시키는 데, 일부 경우들에서 2 이상의 승수만큼 감소시키는 데 도움을 줄 수 있다. 무손실 압축 및 손실 압축 둘 모두, 및 이들의 조합이 이용될 수 있다. 무손실 압축은 압축된 원본 신호에서 원본 신호의 정확한 복사본을 재구성할 수 있는 기법들을 지칭한다. 손실 압축을 사용할 때, 재구성된 신호는 원본 신호와 동일하지 않을 수 있지만, 원본 신호와 재구성된 신호 사이의 왜곡은 재구성된 신호가 의도된 애플리케이션에 유용하기에 충분히 작다. 비디오의 경우, 손실 압축이 널리 이용된다. 용인되는 왜곡의 양은 애플리케이션에 따르고; 예를 들어, 소정 소비자 스트리밍 애플리케이션들의 사용자는 텔레비전 배포 애플리케이션들의 사용자보다 더 높은 왜곡을 용인할 수 있다. 달성가능한 압축 비율은 허용/용인가능한 왜곡이 높을수록 압축 비율들이 높아질 수 있다는 것을 반영할 수 있다.

[0006] 비디오 인코더 및 디코더는 예를 들어 모션 보상, 변환, 양자화 및 엔트로피 코딩을 포함하는 여러 넓은 카테고

리들로부터의 기법을 활용할 수 있다.

[0007] 비디오 코덱 기술들은 인트라 코딩(intra coding)으로 알려진 기법들을 포함할 수 있다. 인트라 코딩에서, 샘플 값들은 이전에 재구성된 참조 화상들로부터의 샘플들 또는 다른 데이터를 참조하지 않고 표현된다. 일부 비디오 코덱들에서, 화상은 공간적으로 샘플들의 블록들로 세분화된다. 샘플의 모든 블록들이 인트라 모드에서 코딩될 때, 그 화상은 인트라 화상일 수 있다. 독립 디코더 리프레시 화상들과 같은 인트라 화상들 및 이들의 도출물들은 디코더 상태를 재설정하는 데 사용될 수 있으므로, 코딩된 비디오 비트스트림 및 비디오 세션의 제1 화상, 또는 정지 이미지로 사용될 수 있다. 인트라 블록의 샘플들은 변환에 노출될 수 있고, 변환 계수들은 엔트로피 코딩 전에 양자화될 수 있다. 인트라 예측은 변환-전 영역에서 샘플 값들을 최소화하는 기법일 수 있다. 일부 경우들에서, 변환 후 DC 값이 더 작고 AC 계수들 더 작을수록, 엔트로피 코딩 후 블록을 나타내기 위해 주어진 양자화 단계 크기에서 요구되는 비트들이 더 적다.

[0008] 예를 들어 MPEG-2 생성 코딩 기술들로부터 알려진 것과 같은 종래의 인트라 코딩은 인트라 예측을 사용하지 않는다. 그러나, 일부 더 새로운 비디오 압축 기술들은, 예를 들어, 주변의 샘플 데이터 및/또는 공간적으로 이웃한 데이터 블록들, 및 디코딩 순서에서 선행하는 데이터 블록들의 인코딩/디코딩 동안 획득되고 디코딩 순서에 선행하는 메타데이터를 시도하는 기법들을 포함한다. 이러한 기법들은 이후 "인트라 예측" 기법들이라고 호칭된다. 적어도 일부 경우들에서, 인트라 예측이 참조 화상들이 아닌 재구성 중인 현재 화상으로부터의 참조 데이터만 사용한다는 것을 주목하라.

[0009] 많은 상이한 형태들의 인트라 예측이 있을 수 있다. 하나 초과의 이러한 기법들이 주어진 비디오 코딩 기술에서 사용될 수 있는 경우, 사용 중인 기법은 인트라 예측 모드에서 코딩될 수 있다. 소정의 경우들에서, 모드들은 서브모드를 및/또는 파라미터들을 가질 수 있고, 이들은 개별적으로 코딩되거나 모드 코드워드에 포함될 수 있다. 주어진 모드/서브모드/파라미터 조합에 사용할 코드워드는 인트라 예측을 통한 코딩 효율 이득에 영향을 미칠 수 있고, 코드워드들을 비트스트림으로 변환하는 데 사용되는 엔트로피 코딩 기술도 마찬가지이다.

[0010] 소정 모드의 인트라 예측은 H.264와 함께 도입되었고, H.265에서 개선되었고, JEM(Joint Exploration Model), 다용도 비디오 코딩(VVC) 및 벤치마크 세트(BMS: benchmark set)와 같은 더 새로운 코딩 기술들에서 추가로 개선되었다. 이미 이용가능한 샘플들에 속하는 이웃 샘플 값들을 사용하여 예측기 블록이 형성될 수 있다. 이웃 샘플들의 샘플 값들은 방향에 따라 예측기 블록에 복사된다. 사용 방향에 대한 참조는 비트스트림에서 코딩되거나 자체적으로 예측될 수 있다.

[0011] 도 1을 참조하면, 우측 하부에는 H.265의 33개의 가능한 예측기 방향들(35개 인트라 모드들의 33개 각도 모드들에 대응)에서 알려진 9개의 예측기 방향들의 서브세트가 묘사된다. 화살표들 수렴하는 지점(101)은 예측 중인 샘플을 나타낸다. 화살표들은 샘플이 예측되는 방향을 나타낸다. 예를 들어, 화살표(102)는 샘플(101)이 수평으로부터 45도 각도로 우측 상부에 있는 샘플 또는 샘플들로부터 예측된다는 것을 나타낸다. 유사하게, 화살표(103)는 샘플(101)이 수평으로부터 22.5도 각도로 샘플(101)의 좌측 아래에 있는 샘플 또는 샘플들로부터 예측된다는 것을 나타낸다.

[0012] 여전히 도 1을 참조하면, 좌측 상단에는  $4 \times 4$  샘플들의 정사각형 블록(104)(굵은 파선에 의해 표시됨)이 묘사된다. 정사각형 블록(104)은 각각 "S", Y 차원에서의 위치(예를 들어, 행 인덱스) 및 X 차원에서의 위치(예를 들어, 열 인덱스)가 라벨링된 16개의 샘플들을 포함한다. 예를 들어, 샘플(S21)은 Y 차원에서 (상단으로부터) 제2 샘플과 X 차원에서 (좌측으로부터) 제1 샘플이다. 유사하게, 샘플(S44)은 Y 및 X 차원 둘 모두에서 블록(104)의 제4 샘플이다. 블록 크기가  $4 \times 4$  샘플이므로, S44는 우측 하단에 있다. 유사한 번호 매기기 체계를 따르는 참조 샘플들이 추가로 도시된다. 참조 샘플은 블록(104)에 관련하여 R, Y 위치(예를 들어, 행 인덱스) 및 X 위치(열 인덱스)가 라벨링된다. H.264 및 H.265 둘 모두에서, 예측 샘플들은 재구성 중인 블록에 이웃하고; 그러므로 음수 값들이 사용될 필요가 없다.

[0013] 인트라 화상 예측은 시그널링된 예측 방향에 따라 적절한 이웃 샘플들로부터 참조 샘플 값들을 복사함으로써 작동할 수 있다. 예를 들어, 코딩된 비디오 비트스트림이 이 블록에 대해 화살표(102)와 일치하는 예측 방향을 나타내는 시그널링을 포함한다고 가정하자 - 즉, 샘플들은 수평으로부터 45도 각도로 우측 상부에 있는 예측 샘플 또는 샘플들로부터 예측된다. 그 경우, 샘플들(S41, S32, S23, 및 S14)은 동일한 참조 샘플(R05)로부터 예측된다. 이어서, 샘플(S44)은 참조 샘플(R08)로부터 예측된다.

[0014] 소정 경우들에서, 다수의 참조 샘플들의 값들은, 특히 방향들이 45도로 균등하게 나누어지지 않을 때 참조 샘플을 계산하기 위해 예를 들어 보간을 통해 결합될 수 있다.

[0015] 비디오 코딩 기술이 발전함에 따라 가능한 방향들의 수는 증가했다. H.264(2003년)에서, 9개의 상이한 방향이 표현될 수 있다. H.265(2013년)에서 33개로 증가했고, 공개 당시 JEM/VVC/BMS는 최대 65개 방향들을 지원할 수 있다. 가장 가능성성이 높은 방향들을 식별하기 위한 실험이 수행되었고, 엔트로피 코딩의 소정 기법들은 가능성이 적은 방향들에 대해 소정 페널티를 허용하면서 적은 수의 비트들로 이러한 가능성 있는 방향을 나타내는 데 사용된다. 또한, 방향들 자체는 때때로 이미 디코딩된 이웃 블록들에서 사용되는 이웃 방향들로부터 예측될 수 있다.

[0016] 도 2는 시간 경과에 따른 증가하는 예측 방향들의 수를 예시하기 위해 JEM에 따른 65개의 인트라 예측 방향들을 묘사하는 개략도(201)를 도시한다.

[0017] 방향을 나타내는 코딩된 비디오 비트스트림의 인트라 예측 방향 비트들의 매핑은 비디오 코딩 기술에서 비디오 코딩 기술로 상이할 수 있고; 예를 들어, 예측 방향의 단순한 직접 매핑에서 인트라 예측 모드, 코드워드들, 가장 가능성 있는 모드들을 포함하는 복잡한 적응 방식들, 및 유사한 기법들에 이르기까지 다양할 수 있다. 그러나, 모든 경우들에서, 통계적으로 소정 다른 특정 방향들보다 비디오 콘텐츠에서 발생할 가능성이 적은 소정 방향들이 있을 수 있다. 비디오 압축의 목표는 중복의 감소이므로, 잘 작동하는 비디오 코딩 기술에서 가능성의 적은 방향들은 가능성이 더 높은 방향들보다 더 많은 비트 수로 표현된다.

### 발명의 내용

[0018] 본 개시내용의 양태들은 비디오 인코딩/디코딩을 위한 방법들 및 장치들을 제공한다. 일부 예들에서, 비디오 디코딩을 위한 장치는 수신 회로 및 프로세싱 회로를 포함한다. 예를 들어, 프로세싱 회로는 코딩된 비디오 비트 스트림으로부터 블록의 예측 정보를 디코딩하고, 블록의 크기와 임계치의 비교에 기반하여 블록에 대한 팔레트 기반 코딩 모드(palette based coding mode)를 허용할지 여부를 결정하고, 블록이 예측 정보에 기반하여 로컬 이중 트리 구조 하에 있는지 여부를 결정한다. 또한, 프로세싱 회로는 블록에 대한 팔레트 기반 코딩 모드를 허용할지 여부의 결정 및 블록이 로컬 이중 트리 구조 하에 있는지 여부의 결정에 기반하여 블록을 디코딩한다.

[0019] 일부 예들에서, 프로세싱 회로는 블록의 크기가 임계치보다 작은 것에 응답하여 블록에 대한 팔레트 기반 코딩 모드를 허용하지 않는다. 일부 예들에서, 임계치는 블록이 임계치보다 큰 것에 응답하여 로컬 이중 트리 구조를 허용하지 않도록 정의된다. 예에서, 프로세싱 회로는 블록이 크로마 블록이고 크로마 블록의 크기가 임계치보다 작은 것에 응답하여 블록에 대한 팔레트 기반 코딩 모드를 허용하지 않는다. 일부 예들에서, 프로세싱 회로는 블록이 로컬 이중 트리 구조 하에서 크로마 블록임을 나타내는 블록의 트리 유형과 모드 유형의 조합에 기반하여 블록에 대한 팔레트 기반 코딩 모드를 허용하지 않는다.

[0020] 일부 실시예들에서, 팔레트 기반 코딩 모드가 허용되는 것에 응답하여, 프로세싱 회로는 코딩된 비디오 비트스 템으로부터, 팔레트 기반 코딩 모드가 블록에 사용되는지 여부를 나타내는 플래그를 디코딩하고 플래그가 블록에 대한 팔레트 기반 코딩 모드의 사용을 나타내는 것에 응답하여 팔레트 기반 코딩 모드를 사용하여 블록을 디코딩한다. 또한, 일부 예들에서, 블록이 로컬 이중 트리 구조 하에 있는 것에 응답하여, 프로세싱 회로는 팔레트 예측기 리스트의 업데이트를 바이패싱한다. 예에서, 프로세싱 회로는 로컬 이중 트리 구조를 트리거하지 않는 것에 응답하여, 블록의 디코딩에 사용된 팔레트에 기반하여 팔레트 예측기 리스트를 업데이트한다.

[0021] 일부 다른 실시예들에서, 팔레트 기반 코딩 모드가 허용되는 것에 응답하여, 프로세싱 회로는 코딩된 비디오 비트스트림으로부터, 팔레트 기반 코딩 모드가 블록에 사용되는지 여부를 나타내는 플래그를 디코딩하고, 플래그가 블록에 대한 팔레트 기반 코딩 모드의 사용을 나타내고 블록이 로컬 이중 트리 구조 하에 있다는 것에 응답하여 감소된 최대 크기를 갖는 팔레트에 기반하여 블록을 디코딩한다. 또한, 프로세싱 회로는 블록의 디코딩에 사용된 팔레트에 기반하여 팔레트 예측기 리스트를 업데이트할 수 있고, 팔레트 예측기 리스트의 크기는 감소된 최대 크기에 의해 제한된다.

[0022] 본 개시내용의 양태들은 또한 비디오 디코딩을 위해 컴퓨터에 의해 실행될 때 컴퓨터로 하여금 비디오 디코딩을 위한 방법을 수행하게 하는 명령들을 저장한 비-일시적 컴퓨터-판독가능 매체를 제공한다.

### 도면의 간단한 설명

[0023] 개시된 주제의 추가 특징들, 성질, 및 다양한 장점들은 이하 상세한 설명 및 첨부 도면들로부터 더 명백하게 될 것이다.

도 1은 인트라 예측 모드들의 예시적인 서브세트의 개략도이다.

도 2는 예시적인 인트라 예측 방향들의 예시이다.

도 3은 일 실시예에 따른 통신 시스템(300)의 간략화된 블록도의 개략도이다.

도 4는 일 실시예에 따른 통신 시스템(400)의 간략화된 블록도의 개략도이다.

도 5는 일 실시예에 따른 디코더의 간략화된 블록도의 개략도이다.

도 6은 일 실시예에 따른 인코더의 간략화된 블록도의 개략도이다.

도 7은 다른 실시예에 따른 인코더의 블록도를 도시한다.

도 8은 다른 실시예에 따른 디코더의 블록도를 도시한다.

도 9는 일부 예들에서 팔레트 예측 리스트를 업데이팅하기 위한 선택스 예를 도시한다.

도 10은 일부 예들에서 팔레트 모드 플래그의 존재를 비트스트림에서 결정하기 위한 선택스 예를 도시한다.

도 11은 본 개시내용의 일부 실시예들에 따른 프로세스 따른 프로세스 예를 서술하는 흐름도를 도시한다.

도 12는 일 실시예에 따른 컴퓨터 시스템의 개략도이다.

### **발명을 실시하기 위한 구체적인 내용**

[0024]

도 3은 본 개시내용의 실시예에 따른 통신 시스템(300)의 간략화된 블록도 예시한다. 통신 시스템(300)은 예를 들어 네트워크(350)를 통해 서로 통신할 수 있는 복수의 단말 디바이스들을 포함한다. 예를 들어, 통신 시스템(300)은 네트워크(350)를 통해 상호연결된 단말 디바이스들(310 및 320)의 제1 쌍을 포함한다. 도 3 예에서, 단말 디바이스들(310 및 320)의 제1 쌍은 데이터의 단방향 송신을 수행한다. 예를 들어, 단말 디바이스(310)는 네트워크(350)를 통해 다른 단말 디바이스(320)로의 송신을 위해 비디오 데이터(예를 들어, 단말 디바이스(310)에 의해 캡처된 비디오 화상들의 스트림)를 코딩할 수 있다. 인코딩된 비디오 데이터는 하나 이상의 코딩된 비디오 비트스트림들의 형태로 송신될 수 있다. 단말 디바이스(320)는 네트워크(350)로부터 코딩된 비디오 데이터를 수신하고, 코딩된 비디오 데이터를 디코딩하여 비디오 화상을 복구하고, 복구된 비디오 데이터에 따라 비디오 화상을 디스플레이할 수 있다. 단방향 데이터 송신은 미디어 서빙 애플리케이션들 등에서 일반적일 수 있다.

[0025]

다른 예에서, 통신 시스템(300)은 예를 들어 화상회의 동안 발생할 수 있는 코딩된 비디오 데이터의 양방향 송신을 수행하는 단말 디바이스들(330 및 340)의 제2 쌍을 포함한다. 데이터의 양방향 송신을 위해, 예에서, 단말 디바이스들(330 및 340)의 각각의 단말 디바이스는 네트워크(350)를 통해 단말 디바이스들(330 및 340)의 다른 단말 디바이스로의 송신을 위해 비디오 데이터(예를 들어, 단말 디바이스에 의해 캡처된 비디오 화상들의 스트림)를 코딩할 수 있다. 단말 디바이스들(330 및 340)의 각각의 단말 디바이스는 또한 단말 디바이스들(330 및 340)의 다른 단말 디바이스에 의해 송신된 코딩된 비디오 데이터를 수신하고, 코딩된 비디오 데이터를 디코딩하여 비디오 화상을 복구할 수 있고 복구된 비디오 데이터에 따라 액세스 가능한 디스플레이 디바이스에 비디오 화상을 디스플레이할 수 있다.

[0026]

도 3 예에서, 단말 디바이스들(310, 320, 330, 및 340)은 서버들, 개인용 컴퓨터들 및 스마트 폰들로 예시될 수 있지만, 본 개시내용의 원리들은 이에 제한되지 않을 수 있다. 본 개시내용의 실시예들은 랩톱 컴퓨터들, 태블릿 컴퓨터들, 미디어 플레이어들 및/또는 전용 화상 회의 장비를 사용한 애플리케이션을 찾는다. 네트워크(350)는 예를 들어 와이어선(유선) 및/또는 무선 통신 네트워크들을 포함하여 단말 디바이스들(310, 320, 330 및 340) 간에 코딩된 비디오 데이터를 전달하는 임의의 수의 네트워크들을 나타낸다. 통신 네트워크(350)는 회선-교환 및/또는 패킷-교환 채널들에서 데이터를 교환할 수 있다. 대표적인 네트워크들은 원격통신 네트워크들, 근거리 통신망들, 광역 네트워크들 및/또는 인터넷을 포함한다. 본 논의의 목적들을 위해, 네트워크(350)의 아키텍처 및 토폴로지는 이하에서 본원에서 설명되지 않는 한 본 개시내용의 동작에 중요하지 않을 수 있다.

[0027]

도 4는 개시된 주제에 대한 애플리케이션에 대한 예로서, 스트리밍 환경에서 비디오 인코더 및 비디오 디코더의 배치를 예시한다. 개시된 주제는 예를 들어 화상 회의, 디지털 TV, CD, DVD, 메모리 스틱 등을 포함하는 디지털 미디어에 압축된 비디오의 저장 등을 포함하는 다른 비디오 가능 애플리케이션들에 같게 적용될 수 있다.

[0028]

스트리밍 시스템은 비디오 소스(401), 예를 들어 압축되지 않은 비디오 화상들(402)의 스트림을 생성하는 디지털 카메라를 포함할 수 있는 캡처 서브시스템(413)을 포함할 수 있다. 예에서, 비디오 화상들의 스트림(402)은 디지털 카메라에 의해 촬영된 샘플들을 포함한다. 인코딩된 비디오 데이터(404)(또는 코딩된 비디오 비트스트림들)와 비교할 때 높은 데이터 볼륨을 강조하기 위해 굵은 선으로 묘사된 비디오 화상들의 스트림(402)은 비디오

소스(401)에 결합된 비디오 인코더(403)를 포함하는 전자 디바이스(420)에 의해 프로세싱될 수 있다. 비디오 인코더(403)는 하드웨어, 소프트웨어, 또는 이들의 조합을 포함하여 아래에서 더 상세히 설명되는 바와 같이 개시된 주제의 양태들을 가능하게 하거나 구현할 수 있다. 비디오 화상들의 스트림(402)과 비교할 때 더 낮은 데이터 볼륨을 강조하기 위해 가는 선으로 묘사된 인코딩된 비디오 데이터(404)(또는 인코딩된 비디오 비트스트림(404))는 미래 사용을 위해 스트리밍 서버(405)에 저장될 수 있다. 도 4의 클라이언트 서브시스템들(406 및 408)과 같은 하나 이상의 스트리밍 클라이언트 서브시스템들은 인코딩된 비디오 데이터(404)의 복사본들(407 및 409)을 검색하기 위해 스트리밍 서버(405)에 액세스할 수 있다. 클라이언트 서브시스템(406)은 예를 들어 전자 디바이스(430)에 비디오 디코더(410)를 포함할 수 있다. 비디오 디코더(410)는 인코딩된 비디오 데이터의 인입 복사본(407)을 디코딩하고 디스플레이(412)(예를 들어, 디스플레이 스크린) 또는 다른 렌더링 디바이스(묘사되지 않음) 상에 렌더링될 수 있는 비디오 화상들(411)의 출력 스트림을 생성한다. 일부 스트리밍 시스템들에서, 인코딩된 비디오 데이터(404, 407 및 409)(예를 들어, 비디오 비트스트림들)는 소정 비디오 코딩/압축 표준들에 따라 인코딩될 수 있다. 이러한 표준들의 예들은 ITU-T 권장사항 H.265를 포함한다. 예에서, 개발 중인 비디오 코딩 표준은 비공식적으로 다음도 비디오 코딩(VVC)으로 알려져 있다. 개시된 주제는 VVC의 맥락에서 사용될 수 있다.

[0029]

전자 디바이스들(420, 430)이 다른 컴포넌트들(도시되지 않음)을 포함할 수 있음에 주목된다. 예를 들어, 전자 디바이스(420)는 비디오 디코더(도시되지 않음)를 포함할 수 있고, 전자 디바이스(430)는 비디오 인코더(도시되지 않음)를 포함할 수 있다.

[0030]

도 5는 본 개시내용의 일 실시예에 따른 비디오 디코더(510)의 블록도를 도시한다. 비디오 디코더(510)는 전자 디바이스(530)에 포함될 수 있다. 전자 디바이스(530)는 수신기(531)(예를 들어, 수신 회로)를 포함할 수 있다. 비디오 디코더(510)는 도 4 예에서 비디오 디코더(410) 대신에 사용될 수 있다.

[0031]

수신기(531)는 비디오 디코더(510)에 의해 디코딩될 하나 이상의 코딩된 비디오 시퀀스들을 수신할 수 있고; 동일 또는 다른 실시예에서, 한 번에 하나의 코딩된 비디오 시퀀스로, 각각의 코딩된 비디오 시퀀스의 디코딩은 다른 코딩된 비디오 시퀀스들과 독립적이다. 코딩된 비디오 시퀀스는 인코딩된 비디오 데이터를 저장하는 저장 디바이스에 대한 하드웨어/소프트웨어 링크일 수 있는 채널(501)로부터 수신될 수 있다. 수신기(531)는 인코딩된 비디오 데이터를 다른 데이터, 예를 들어 코딩된 오디오 데이터 및/또는 보조 데이터 스트림과 함께 수신할 수 있고, 이는 각각의 사용 엔티티들(묘사되지 않음)로 포워딩될 수 있다. 수신기(531)는 코딩된 비디오 시퀀스를 다른 데이터로부터 분리할 수 있다. 네트워크 지터(jitter)를 방지하기 위해, 버퍼 메모리(515)는 수신기(531)와 엔트로피 디코더/파서(parser)(520)(이하 "파서(520)") 사이에 결합될 수 있다. 소정 애플리케이션들에서, 버퍼 메모리(515)는 비디오 디코더(510)의 일부이다. 다른 애플리케이션들에서, 버퍼 메모리(515)는 비디오 디코더(510)(묘사되지 않음) 외측에 있을 수 있다. 또 다른 애플리케이션들에서, 예를 들어 네트워크 지터를 방지하기 위해 비디오 디코더(510) 외측에 버퍼 메모리(묘사되지 않음)가 있을 수 있고, 또한 재생 타이밍을 처리하기 위해 예를 들어 비디오 디코더(510) 내부에 다른 버퍼 메모리(515)가 있을 수 있다. 수신기(531)가 충분한 대역폭 및 제어성의 저장/포워드 디바이스, 또는 동시동기식 네트워크로부터 데이터를 수신할 때, 버퍼 메모리(515)는 필요하지 않거나, 작을 수 있다. 인터넷과 같은 최대 효과 패킷 네트워크들에서 사용하기 위해, 버퍼 메모리(515)가 요구될 수 있고, 비교적 클 수 있고 유리하게는 적응형 크기를 가질 수 있고, 비디오 디코더(510)의 외측의 운영 체계 또는 유사한 엘리먼트들(묘사되지 않음)로 적어도 부분적으로 구현될 수 있다.

[0032]

비디오 디코더(510)는 코딩된 비디오 시퀀스로부터 심볼들(521)을 재구성하기 위해 파서(520)를 포함할 수 있다. 이러한 심볼들의 카테고리들은 비디오 디코더(510)의 동작을 관리하는 데 사용되는 정보, 및 잠재적으로 전자 디바이스의 필수 부분이 아니지만 도 5에 도시된 바와 같이 전자 디바이스(530)에 결합될 수 있는 렌더링 디바이스(512)(예를 들어, 디스플레이 스크린)와 같은 렌더링 디바이스를 제어하기 위한 정보를 포함할 수 있다. 렌더링 디바이스(들)에 대한 제어 정보는 SEI(Supplemental Enhancement Information) 메시지들 또는 비디오 사용 정보(VUI: Video Usability Information) 파라미터 세트 프레그먼트들(묘사되지 않음)의 형태일 수 있다. 파서(520)는 수신된 코딩된 비디오 시퀀스를 파싱/엔트로피-디코딩할 수 있다. 코딩된 비디오 시퀀스의 코딩은 비디오 코딩 기술 또는 표준에 따를 수 있고, 가변 길이 코딩, 허프만 코딩, 문맥 감도가 있거나 없는 산술 코딩 등을 포함하는 다양한 원리들을 따를 수 있다. 파서(520)는 그룹에 대응하는 적어도 하나의 파라미터에 기반하여, 비디오 디코더의 핵심들의 서브그룹들 중 적어도 하나에 대한 서브그룹 파라미터들의 세트를 코딩된 비디오 시퀀스로부터 추출할 수 있다. 서브그룹들은 GOP(Groups of Picture)들, 화상들, 타일들, 슬라이스들, 매크로블록들, 코딩 유닛(CU)들, 블록들, 변환 유닛(TU: Transform Unit)들, 예측 유닛(PU: Prediction Unit)들 등을 포함할 수 있다. 파서(520)는 또한 변환 계수들, 양자화기 파라미터 값들, 모션 벡터

들 등과 같은 코딩된 비디오 시퀀스 정보로부터 추출할 수 있다.

[0033] 파서(520)는 버퍼 메모리(515)로부터 수신된 비디오 시퀀스에 대해 엔트로피 디코딩/파싱 동작을 수행하여 심볼들(521)을 생성할 수 있다.

[0034] 심볼들(521)의 재구성은 코딩된 비디오 화상의 유형 또는 그 일부(이를테면: 인터 및 인트라 화상, 인터 및 인트라 블록), 및 다른 요인들에 따라 다수의 상이한 유닛들을 포함할 수 있다. 어떤 유닛들이 관련되고, 그리고 어떻게 관련되는지는 파서(520)에 의해 코딩된 비디오 시퀀스로부터 파싱된 서브그룹 제어 정보에 의해 제어될 수 있다. 파서(520)와 아래의 다수의 유닛들 사이의 이러한 서브그룹 제어 정보의 흐름은 명확성을 위해 묘사되지 않는다.

[0035] 이미 언급된 기능 블록들을 넘어서, 비디오 디코더(510)는 개념적으로 아래에 설명된 바와 같이 다수의 기능 유닛들로 세분될 수 있다. 상업적 제약들 하에서 동작하는 실제 구현에서, 이러한 유닛들 중 많은 부분이 서로 밀접하게 상호 작용하고 적어도 부분적으로는 서로 통합될 수 있다. 그러나, 개시된 주제를 설명하기 위해, 아래의 기능 유닛들로의 개념적 세분화가 적절하다.

[0036] 제1 유닛은 스케일러/역변환 유닛(551)이다. 스케일러/역변환 유닛(551)은 파서(520)로부터 심볼(들)(521)로서 사용할 변환, 블록 크기, 양자화 인자, 양자화 스케일링 행렬 등을 포함하는 제어 정보뿐만 아니라 양자화된 변환 계수를 수신한다. 스케일러/역변환 유닛(551)은 어그리게이터(aggregator)(555)에 입력될 수 있는 샘플 값들을 포함하는 블록들을 출력할 수 있다.

[0037] 일부 경우들에서, 스케일러/역 변환(551)의 출력 샘플들은 인트라 코딩된 블록; 즉, 이전에 재구성된 화상들로부터 예측 정보를 사용하지 않지만 현재 화상의 이전에 재구성된 부분들로부터 예측 정보를 사용할 수 있는 블록에 속할 수 있다. 이러한 예측 정보는 인트라 화상 예측 유닛(552)에 의해 제공될 수 있다. 일부 경우들에서, 인트라 화상 예측 유닛(552)은 현재 화상 버퍼(558)로부터 폐치된 이미 재구성된 주변 정보를 사용하여 재구성 중인 블록과 동일한 크기 및 형상의 블록을 생성한다. 현재 화상 버퍼(558)는 예를 들어 부분적으로 재구성된 현재 화상 및/또는 완전히 재구성된 현재 화상을 버퍼링한다. 어그리게이터(555)는 일부 경우들에서, 샘플 단위로 인트라 예측 유닛(552)이 생성한 예측 정보를 스케일러/역변환 유닛(551)에 의해 제공되는 출력 샘플 정보에 추가한다.

[0038] 다른 경우들에서, 스케일러/역변환 유닛(551)의 출력 샘플들은 인터 코딩되고, 잠재적으로 모션 보상된 블록에 속할 수 있다. 그러한 경우에, 모션 보상 예측 유닛(553)은 예측에 사용되는 샘플들을 폐치하기 위해 참조 화상 메모리(557)에 액세스할 수 있다. 블록에 속하는 심볼들(521)에 따라 폐치된 샘플들을 모션 보상한 후, 이러한 샘플들은 출력 샘플 정보를 생성하기 위해 어그리게이터(555)에 의해 스케일러/역변환 유닛(551)의 출력에 추가된다(이 경우 잔차 샘플들 또는 잔차 신호로 호칭됨). 모션 보상 예측 유닛(553)이 예측 샘플들을 폐치하는 참조 화상 메모리(557) 내의 어드레스들은 모션 벡터들에 의해 제어될 수 있고, 이는 예를 들어 X, Y 및 참조 화상 성분들을 가질 수 있는 심볼들(521) 형태의 모션 보상 예측 유닛(553)에 이용가능하다. 모션 보상은 또한 정확한 서브-샘플 모션 벡터가 사용 중일 때 참조 화상 메모리(557)로부터 폐치된 샘플 값들의 보간, 모션 벡터 예측 메커니즘들 등을 포함할 수 있다.

[0039] 어그리게이터(555)의 출력 샘플들은 루프 필터 유닛(556)에서 다양한 루프 필터링 기법들의 대상이 될 수 있다. 비디오 압축 기술들은 코딩된 비디오 시퀀스(또한 코딩된 비디오 비트스트림이라고 지칭됨)에 포함된 파라미터들에 의해 제어되고 파서(520)로부터 심볼들(521)로서 루프 필터 유닛(556)에 이용가능하게 되지만, 코딩된 화상 또는 코딩된 비디오 시퀀스의 (디코딩 순서로) 이전 부분들의 디코딩 동안 획득된 메타-정보에 응답할 수 있을 뿐만 아니라, 이전에 재구성되고 루프-필터링된 샘플 값들에 응답할 수 있는 루프-내 필터 기술들을 포함할 수 있다.

[0040] 루프 필터 유닛(556)의 출력은 렌더 디바이스(512)로 출력될 수 있을 뿐만 아니라 미래의 인터-화상 예측에 사용하기 위해 참조 화상 메모리(557)에 저장될 수 있는 샘플 스트림일 수 있다.

[0041] 완전히 재구성되면, 소정 코딩된 화상은 미래 예측을 위한 참조 화상으로 사용될 수 있다. 예를 들어, 현재 화상에 대응하는 코딩된 화상이 완전히 재구성되고 코딩된 화상이 (예를 들어, 파서(520)에 의해) 참조 화상으로 식별되면, 현재 화상 버퍼(558)는 참조 화상 메모리(557)의 일부가 될 수 있고, 새로운 현재 화상 버퍼는 다음의 코딩된 화상의 재구성을 시작하기 전에 재배정될 수 있다.

[0042] 비디오 디코더(510)는 ITU-T Rec. H.265 같은 표준의 미리 결정된 비디오 압축 기술에 따른 디코딩 동작들을 수행할 수 있다. 코딩된 비디오 시퀀스가 비디오 압축 기술 또는 표준의 신택스와 비디오 압축 기술 또는 표준에

서 문서화된 프로파일 둘 모두를 고수한다는 점에서, 코딩된 비디오 시퀀스는 사용되는 비디오 압축 기술 또는 표준에 의해 지정된 선택지를 따를 수 있다. 특히, 프로파일은 비디오 압축 기술 또는 표준에서 이용가능한 모든 도구들에서 해당 프로파일에서 사용하기 위해 이용가능한 유일한 도구들로 소정 도구들을 선택할 수 있다. 또한 규정 준수를 위해 필요한 것은 코딩된 비디오 시퀀스의 복잡성이 비디오 압축 기술 또는 표준의 레벨에 의해 정의된 경계 내에 있어야 한다는 것일 수 있다. 일부 경우들에서, 레벨들은 최대 화상 크기, 최대 프레임 레이트, 최대 재구성 샘플 레이트(예를 들어, 초당 메가샘플들로 측정됨), 최대 참조 화상 크기 등을 제약한다. 레벨들에 의해 설정된 한계들은 일부 경우들에서, 코딩된 비디오 시퀀스에서 시그널링되는 가상 참조 디코더(HRD: Hypothetical Reference Decoder) 사양들 및 HRD 버퍼 관리를 위한 메타데이터를 통해 추가로 제약될 수 있다.

[0043] 실시예에서, 수신기(531)는 인코딩된 비디오와 함께 추가적인(중복) 데이터를 수신할 수 있다. 추가 데이터는 코딩된 비디오 시퀀스(들)의 일부로 포함될 수 있다. 추가 데이터는 데이터를 적절하게 디코딩하고/하거나 원본 비디오 데이터를 보다 정확하게 재구성하기 위해 비디오 디코더(510)에 의해 사용될 수 있다. 추가 데이터는 예를 들어 시간적, 공간적 또는 신호 잡음비(SNR) 향상 계층들, 중복 슬라이스들, 중복 화상들, 순방향 오류 정정 코드 등의 형태일 수 있다.

[0044] 도 6은 본 개시내용의 일 실시예에 따른 비디오 인코더(603)의 블록도를 도시한다. 비디오 인코더(603)는 전자 디바이스(620)에 포함된다. 전자 디바이스(620)는 송신기(640)(예를 들어, 송신 회로)를 포함한다. 비디오 인코더(603)는 도 4 예의 비디오 인코더(403) 대신에 사용될 수 있다.

[0045] 비디오 인코더(603)는 비디오 인코더(603)에 의해 코딩될 비디오 이미지(들)를 캡처할 수 있는 비디오 소스(601)(이는 도 6 예에서 전자 디바이스(620)의 일부가 아님)로부터 비디오 샘플들을 수신할 수 있다. 다른 예에서, 비디오 소스(601)는 전자 디바이스(620)의 일부이다.

[0046] 비디오 소스(601)는 임의의 적합한 비트 깊이(예를 들어, 8 비트, 10 비트, 12 비트, ...), 임의의 컬러 공간(예를 들어, BT.601 Y CrCb, RGB, ...), 및 임의의 적합한 샘플링 구조(예를 들어, Y CrCb 4:2:0, Y CrCb 4:4:4) 일 수 있는 디지털 비디오 샘플 스트림의 형태로 비디오 인코더(603)에 의해 코딩될 소스 비디오 시퀀스를 제공할 수 있다. 미디어 서빙 시스템에서, 비디오 소스(601)는 미리 준비된 비디오를 저장하는 저장 디바이스일 수 있다. 화상회의 시스템에서, 비디오 소스(601)는 로컬 이미지 정보를 비디오 시퀀스로 캡처하는 카메라일 수 있다. 비디오 데이터는 순서대로 볼 때 모션을 부여하는 복수의 개별 화상들로 제공될 수 있다. 화상 자체는 픽셀들의 공간 어레이로 구성될 수 있고, 여기서 각각의 픽셀은 사용 중인 샘플링 구조, 컬러 공간 등에 따라 하나 이상의 샘플들을 포함할 수 있다. 통상의 기술자는 픽셀들과 샘플들 간의 관계를 쉽게 이해할 수 있다. 아래 설명은 샘플들에 초점을 맞춘다.

[0047] 실시예에 따르면, 비디오 인코더(603)는 실시간으로 또는 애플리케이션에 의해 요구되는 임의의 다른 시간 제약들 하에서 소스 비디오 시퀀스의 화상들을 코딩된 비디오 시퀀스(643)로 코딩 및 압축할 수 있다. 적절한 코딩 속도를 시행하는 것은 제어기(650)의 하나의 기능이다. 일부 실시예들에서, 제어기(650)는 아래에 설명되는 바와 같이 다른 기능 유닛들을 제어하고 다른 기능 유닛들에 기능적으로 결합된다. 결합은 명확성을 위해 묘사되지 않는다. 제어기(650)에 의해 설정되는 파라미터들은 레이트 제어 관련 파라미터들(화상 스킁, 양자화기, 레이트-왜곡 최적화 기법들의 람다 값, ...), 화상 크기, 화상 그룹(GOP) 레이아웃, 최대 모션 벡터 탐색 범위 및 등을 포함할 수 있다. 제어기(650)는 소정 시스템 설계에 최적화된 비디오 인코더(603)에 속하는 다른 적합한 기능들을 갖도록 구성될 수 있다.

[0048] 일부 실시예들에서, 비디오 인코더(603)는 코딩 루프에서 동작하도록 구성된다. 과도하게 단순화된 설명으로서, 예에서, 코딩 루프는 소스 코더(630)(예를 들어, 코딩될 입력 화상에 기반하여 심볼 스트림과 같은 심볼들, 및 참조 화상(들)을 생성하는 역할을 함), 및 비디오 인코더(603)에 내장된 (로컬) 디코더(633)를 포함할 수 있다. 디코더(633)는 (개시된 주제에서 고려되는 비디오 압축 기술들에서 심볼들과 코딩된 비디오 비트스트림 간의 임의의 압축이 무손실이기 때문에) (원격) 디코더가 또한 생성하는 것과 유사한 방식으로 샘플 데이터를 생성하기 위해 심볼들을 재구성한다. 재구성된 샘플 스트림(샘플 데이터)은 참조 화상 메모리(634)에 입력된다. 심볼 스트림의 디코딩이 디코더 위치(로컬 또는 원격)에 관계없이 정확한 비트 결과들을 야기하기 때문에, 참조 화상 메모리(634)의 콘텐츠는 또한 로컬 인코더와 원격 인코더 사이에서 정확한 비트이다. 즉, 인코더의 예측 부분은 디코딩 동안 예측을 사용할 때 디코더가 "보는" 것과 정확히 동일한 샘플 값을 참조 화상 샘플들로 "본다". 참조 화상 동조화의 이런 기본 원리(및 예를 들어 채널 오류들로 인해 동조화가 유지될 수 없는 경우 결과적인 드리프트)는 일부 관련 기술에서 또한 사용된다.

- [0049] "로컬" 디코더(633)의 동작은 도 5와 관련하여 위에서 이미 상세히 설명된 비디오 디코더(510)와 같은 "원격" 디코더의 동작과 동일할 수 있다. 그러나, 또한 도 5를 간략히 참조하면, 심볼들이 이용가능하고 엔트로피 코더(645) 및 파서(520)에 의한 코딩된 비디오 시퀀스에 대한 심볼들의 인코딩/디코딩이 무손실일 수 있기 때문에, 베퍼 메모리(515), 및 파서(520)를 포함하는 비디오 디코더(510)의 엔트로피 디코딩 부분들은 로컬 디코더(633)에서 완전히 구현되지 않을 수 있다.
- [0050] 이 시점에서 이루어질 수 있는 관찰은 디코더에 존재하는 파싱/엔트로피 디코딩을 제외한 임의의 디코더 기술이 또한 반드시 상응하는 인코더에 실질적으로 동일한 기능 형태로 존재할 필요가 있다는 것이다. 이러한 이유로, 개시된 주제는 디코더 동작에 초점을 맞춘다. 인코더 기술들의 설명은 포괄적으로 설명된 디코더 기술들의 반대 이므로 축약될 수 있다. 소정 영역들에서만 더 자세한 설명이 요구되고 아래에 제공된다.
- [0051] 동작 동안, 일부 예들에서, 소스 코더(630)는 "참조 화상들"로서 지정된 비디오 시퀀스로부터의 하나 이상의 이전에 코딩된 화상을 참조하여 예측적으로 입력 화상을 코딩하는 모션 보상 예측 코딩을 수행할 수 있다. 이러한 방식으로, 코딩 엔진(632)은 입력 화상에 대한 예측 참조(들)로서 선택될 수 있는 참조 화상(들)의 픽셀 블록들과 입력 화상의 픽셀 블록들 간의 차이들을 코딩한다.
- [0052] 로컬 비디오 디코더(633)는 소스 코더(630)에 의해 생성된 심볼들에 기반하여 참조 화상들로 지정될 수 있는 화상들의 코딩된 비디오 데이터를 디코딩할 수 있다. 코딩 엔진(632)의 동작들은 유리하게는 순서 프로세스들일 수 있다. 코딩된 비디오 데이터가 비디오 디코더(도 6에 도시되지 않음)에서 디코딩될 수 있을 때, 재구성된 비디오 시퀀스는 통상적으로 약간의 오류들을 갖는 소스 비디오 시퀀스의 복제일 수 있다. 로컬 비디오 디코더(633)는 참조 화상들에 대해 비디오 디코더에 의해 수행될 수 있는 디코딩 프로세스들을 복제하고 재구성된 참조 화상들이 참조 화상 캐시(634)에 저장되게 할 수 있다. 이러한 방식으로, 비디오 인코더(603)는 파-엔드(far-end) 비디오 디코더에 의해 획득된 재구성된 참조 화상들로서 공통 콘텐츠를 갖는 재구성된 참조 화상들의 복사본들을 로컬적으로 저장할 수 있다(송신 오류들 없음).
- [0053] 예측기(635)는 코딩 엔진(632)에 대한 예측 탐색들을 수행할 수 있다. 즉, 코딩될 새로운 화상에 대해, 예측기(635)는 (후보 참조 픽셀 블록들로서) 샘플 데이터 또는 참조 화상 모션 벡터들, 블록 형상들 등과 같은 소정 메타데이터에 대해 참조 화상 메모리(634)를 탐색할 수 있고, 이는 새로운 화상들에 대한 적절한 예측 참조 역할을 할 수 있다. 예측기(635)는 적절한 예측 참조들을 찾기 위해 샘플 블록 단위로 동작할 수 있다. 일부 경우들에서, 예측기(635)에 의해 획득된 탐색 결과들에 의해 결정된 바와 같이, 입력 화상은 참조 화상 메모리(634)에 저장된 다수의 참조 화상들로부터 도출된 예측 참조들을 가질 수 있다.
- [0054] 제어기(650)는 예를 들어 비디오 데이터를 인코딩하는데 사용되는 파라미터들 및 서브그룹 파라미터들의 설정을 포함하는 소스 코더(630)의 코딩 동작들을 관리할 수 있다.
- [0055] 전술한 모든 기능 유닛들의 출력은 엔트로피 코더(645)에서 엔트로피 코딩될 수 있다. 엔트로피 코더(645)는 허프만 코딩, 가변 길이 코딩, 산술 코딩 등과 같은 기술들에 따라 심볼들을 무손실 압축함으로써 다양한 기능 유닛들에 의해 생성된 심볼들을 코딩된 비디오 시퀀스로 트랜스레이팅(translate)한다.
- [0056] 송신기(640)는 엔트로피 코더(645)에 의해 생성된 코딩된 비디오 시퀀스(들)를 베퍼링하여 인코딩된 비디오 데이터를 저장할 저장 디바이스에 대한 하드웨어/소프트웨어 링크일 수 있는 통신 채널(660)을 통한 송신을 준비할 수 있다. 송신기(640)는 비디오 코더(603)로부터의 코딩된 비디오 데이터를 송신될 다른 데이터, 예를 들어 코딩된 오디오 데이터 및/또는 보조 데이터 스트림들(소스들은 도시되지 않음)과 병합할 수 있다.
- [0057] 제어기(650)는 비디오 인코더(603)의 동작을 관리할 수 있다. 코딩 동안, 제어기(650)는 각 화상에 적용될 수 있는 코딩 기법들에 영향을 미칠 수 있는 소정 코딩된 화상 유형을 각각의 코딩된 화상에 할당할 수 있다. 예를 들어, 화상들은 종종 다음 화상 유형들 중 하나로 할당될 수 있다:
- [0058] 인트라 화상(I 화상)은 예측 소스로서 시퀀스의 임의의 다른 화상을 사용하지 않고 코딩 및 디코딩될 수 있는 것일 수 있다. 일부 비디오 코덱들은 예를 들어 독립 디코더 리프레시("IDR") 화상들을 비롯한 다양한 유형들의 인트라 화상들을 허용한다. 통상의 기술자는 I 화상들의 이러한 변형들 및 각 애플리케이션들 및 피처(feature)들을 알고 있다.
- [0059] 예측 화상(P 화상)은 각각의 블록의 샘플 값들을 예측하기 위해 최대 하나의 모션 벡터 및 참조 인덱스를 사용하는 인트라 예측 또는 인터 예측을 사용하여 코딩 및 디코딩될 수 있는 것일 수 있다.
- [0060] 양방향 예측 화상(B 화상)은 각각의 블록의 샘플 값들을 예측하기 위해 최대 2개의 모션 벡터들 및 참조 인덱스

들을 사용하는 인트라 예측 또는 인터 예측을 사용하여 코딩 및 디코딩될 수 있는 것일 수 있다. 유사하게, 다중-예측 화상들은 단일 블록의 재구성을 위해 2개 초과의 참조 화상들 및 연관된 메타데이터를 사용할 수 있다.

[0061] 소스 화상들은 일반적으로 복수의 샘플 블록들(예를 들어, 각각  $4\times 4$ ,  $8\times 8$ ,  $4\times 8$  또는  $16\times 16$  샘플들의 블록들)로 공간적으로 세분화되고 블록 단위로 코딩될 수 있다. 블록들은 블록의 각 화상들에 적용된 코딩 할당에 의해 결정된 바와 같이 다른 (이미 코딩된) 블록들을 참조하여 예측적으로 코딩될 수 있다. 예를 들어, I 화상들의 블록들은 비-예측적으로 코딩될 수 있거나 동일한 화상의 이미 코딩된 블록들을 참조하여 예측적으로 코딩될 수 있다(공간 예측 또는 인트라 예측). P 화상들의 픽셀 블록들은 하나의 이전에 코딩된 참조 화상을 참조하여 공간적 예측을 통해 또는 시간적 예측을 통해 예측적으로 코딩될 수 있다. B 화상들의 블록들은 1개 또는 2개의 이전에 코딩된 참조 화상들을 참조하여 공간적 예측을 통해 또는 시간적 예측을 통해 예측적으로 코딩될 수 있다.

[0062] 비디오 디코더(603)는 ITU-T Rec. H.265 같은 미리 결정된 비디오 코딩 기술 또는 표준에 따른 디코딩 동작들을 수행할 수 있다. 그 동작에서, 비디오 인코더(603)는 입력 비디오 시퀀스에서 시간적 및 공간적 중복을 이용하는 예측 코딩 동작들을 포함하는 다양한 압축 동작들을 수행할 수 있다. 그러므로, 코딩된 비디오 데이터는 사용되는 비디오 코딩 기술 또는 표준에 의해 지정된 선택스를 따를 수 있다.

[0063] 실시예에서, 송신기(640)는 인코딩된 비디오와 함께 추가 데이터를 송신할 수 있다. 소스 코더(630)는 코딩된 비디오 시퀀스의 일부로서 그러한 데이터를 포함할 수 있다. 추가 데이터는 시간/공간/SNR 향상 계층들, 중복 화상들 및 슬라이스들, SEI 메시지들, VUI 파라미터 세트 프레그먼트들 등과 같은 다른 형태들의 중복 데이터를 포함할 수 있다.

[0064] 비디오는 시간적 시퀀스로 복수의 소스 화상들(비디오 화상들)로 캡처될 수 있다. 인트라-화상 예측(종종 인트라 예측으로 축약됨)은 주어진 화상에서 공간적 상관을 이용하고, 인터 화상 예측은 화상들 간의 (시간적 또는 다른) 상관을 이용한다. 예에서, 현재 화상이라고 지칭되는 인코딩/디코딩 중인 특정 화상은 블록들로 파티셔닝된다. 현재 화상의 블록이 비디오의 이전에 코딩되고 여전히 버퍼링된 참조 화상의 참조 블록과 유사한 경우, 현재 화상의 블록은 모션 벡터라고 지칭되는 벡터에 의해 코딩될 수 있다. 모션 벡터는 참조 화상의 참조 블록을 가리키고, 다수의 참조 화상들이 사용 중인 경우 참조 화상을 식별하는 3 차원을 가질 수 있다.

[0065] 일부 실시예들에서, 양방향 예측 기법은 인터-화상 예측에서 사용될 수 있다. 양방향 예측 기법에 따르면, 비디오의 현재 화상에 대해 디코딩 순서가 둘 다 우선인(그러나 디스플레이 순서에서 각각 과거 및 미래일 수 있음) 제1 참조 화상 및 제2 참조 화상과 같은 2개의 참조 화상이 사용된다. 현재 화상의 블록은 제1 참조 화상의 제1 참조 블록을 가리키는 제1 모션 벡터, 및 제2 참조 화상의 제2 참조 블록을 가리키는 제2 모션 벡터에 의해 코딩될 수 있다. 블록은 제1 참조 블록과 제2 참조 블록의 조합에 의해 예측될 수 있다.

[0066] 또한, 코딩 효율을 향상시키기 위해 인터-화상 예측에 병합 모드 기법이 사용될 수 있다.

[0067] 본 개시내용의 일부 실시예들에 따르면, 인터-화상 예측들 및 인트라-화상 예측들과 같은 예측들은 블록들의 단위로 수행된다. 예를 들어, HEVC 표준에 따르면, 비디오 화상들의 시퀀스의 화상은 압축을 위해 코딩 트리 유닛(CTU(Coding Tree Units))들로 파티셔닝되고, 화상의 CTU들은  $64\times 64$  픽셀들,  $32\times 32$  픽셀들, 또는  $16\times 16$  픽셀들과 같은 동일한 크기를 갖는다. 일반적으로, CTU는 1개의 루마 CTB와 2개의 크로마 CTB들인 3개의 코딩 트리 블록(CTB)들을 포함한다. 각각의 CTU는 재귀적으로 하나 또는 다수 개의 코딩 유닛(CU)들로 쿼드트리(quadtree) 분할될 수 있다. 예를 들어,  $64\times 64$  픽셀들의 CTU는  $64\times 64$  픽셀들의 하나의 CU,  $32\times 32$  픽셀들의 4개의 CU들 또는  $16\times 16$  픽셀들의 16개의 CU들로 분할될 수 있다. 예에서, 각각의 CU는 인터 예측 유형 또는 인트라 예측 유형과 같은 CU에 대한 예측 유형을 결정하기 위해 분석된다. CU는 시간 및/또는 공간 예측성에 따라 하나 이상의 예측 유닛(PU)들로 분할된다. 일반적으로, 각각의 PU는 루마 예측 블록(PB)과, 2개의 크로마 PB들을 포함한다. 실시예에서, 코딩(인코딩/디코딩)에서의 예측 동작은 예측 블록 단위로 수행된다. 예측 블록의 예로서 루마 예측 블록을 사용하여, 예측 블록은  $8\times 8$  픽셀들,  $16\times 16$  픽셀들,  $8\times 16$  픽셀들,  $16\times 8$  픽셀들 등과 같은 픽셀들에 대한 값들(예를 들어, 루마 값)의 행렬을 포함한다.

[0068] 도 7은 본 개시내용의 다른 실시예에 따른 비디오 인코더(703)의 다이어그램을 도시한다. 비디오 인코더(703)는 비디오 화상들의 시퀀스에서 현재 비디오 화상 내의 샘플 값들의 프로세싱 블록(예를 들어, 예측 블록)을 수신하고, 그 프로세싱 블록을 코딩된 비디오 시퀀스의 일부인 코딩된 화상으로 인코딩하도록 구성된다. 예에서, 비디오 인코더(703)는 도 4 예의 비디오 인코더(403) 대신에 사용된다.

[0069] HEVC 예에서, 비디오 인코더(703)는  $8\times 8$  샘플들의 예측 블록 등과 같은 프로세싱 블록에 대한 샘플 값들의 행

렬을 수신한다. 비디오 인코더(703)는 프로세싱 블록이 예를 들어 레이트-왜곡 최적화를 사용하여 인트라 모드, 인터 모드, 또는 이중-예측 모드를 사용하여 가장 잘 코딩되는지를 결정한다. 프로세싱 블록이 인트라 모드에서 코딩될 때, 비디오 인코더(703)는 프로세싱 블록을 코딩된 화상으로 인코딩하기 위해 인트라 예측 기법을 사용할 수도 있고; 프로세싱 블록이 인터 모드 또는 이중-예측 예측 모드에서 코딩될 때, 비디오 인코더 (703)는 프로세싱 블록을 코딩된 화상으로 인코딩하기 위해 인터 예측 또는 이중-예측 기법을 각각 사용할 수 있다. 소정 비디오 코딩 기술들에서, 병합 모드는 모션 벡터가 예측기들 외부의 코딩된 모션 벡터 성분의 이점 없이 하나 이상의 모션 벡터 예측기들로부터 도출되는 인터 화상 예측 서브모드일 수 있다. 소정 다른 비디오 코딩 기술들에서, 대상 블록에 적용 가능한 모션 벡터 성분이 존재할 수 있다. 예에서, 비디오 인코더(703)는 프로세싱 블록의 모드를 결정하기 위한 모드 판정 모듈(도시되지 않음)과 같은 다른 컴포넌트들을 포함한다.

[0070] 도 7 예에서, 비디오 인코더(703)는 도 7에 도시된 바와 같이 함께 결합된 인터 인코더(730), 인트라 인코더(722), 잔차 계산기(residue calculator)(723), 스위치(726), 잔차 인코더(724), 일반 제어기(721), 엔트로피 인코더(725)를 포함한다.

[0071] 인터 인코더(730)는 현재 블록(예를 들어, 프로세싱 블록)의 샘플들을 수신하고, 블록을 참조 화상들의 하나 이상의 참조 블록들(예를 들어, 이전 화상들 및 이후 화상들의 블록들)과 비교하고, 인터 예측 정보(예를 들어, 인터 인코딩 기법에 따른 중복 정보의 기술, 모션 벡터들, 병합 모드 정보)를 생성하고, 및 임의의 적합한 기법을 사용하여 인터 예측 정보에 기반하여 인터 예측 결과들(예를 들어, 예측 블록)을 계산하도록 구성된다. 일부 예들에서, 참조 화상들은 인코딩된 비디오 정보에 기반하여 디코딩된 참조 화상들이다.

[0072] 인트라 인코더(722)는 현재 블록(예를 들어, 프로세싱 블록)의 샘플들을 수신하고, 일부 경우들에서 블록을 동일한 화상에서 이미 코딩된 블록들과 비교하고, 변환 후에 양자화된 계수들, 및 일부 경우들에서 또한 인트라 예측 정보(예를 들어, 하나 이상의 인트라 인코딩 기법들에 따른 인트라 예측 방향 정보)를 생성하도록 구성된다. 예에서, 인트라 인코더(722)는 또한 동일한 화상의 인트라 예측 정보 및 참조 블록들에 기반하여 인트라 예측 결과들(예를 들어, 예측된 블록)을 계산한다.

[0073] 일반 제어기(721)는 일반 제어 데이터를 결정하고 일반 제어 데이터에 기반하여 비디오 인코더(703)의 다른 컴포넌트들을 제어하도록 구성된다. 예에서, 일반 제어기(721)는 블록의 모드를 결정하고, 그 모드에 기반하여 제어 신호를 스위치(726)에 제공한다. 예를 들어, 모드가 인트라 모드인 경우, 일반 제어기(721)는 스위치(726)를 제어하여 인트라 모드 결과를 잔차 계산기(723)에서 사용하도록 선택하고, 엔트로피 인코더(725)를 제어하여 인트라 예측 정보를 선택하고 비트스트림에 인트라 예측 정보를 포함하고; 모드가 인터 모드인 경우, 일반 제어기 (721)는 스위치(726)를 제어하여 잔차 계산기(723)에서 사용할 인터 예측 결과를 선택하고, 엔트로피 인코더 (725)를 제어하여 인터 예측 정보를 선택하고 비트스트림에 인터 예측 정보를 포함한다.

[0074] 잔차 계산기(723)는 수신된 블록과 인트라 인코더(722) 또는 인터 인코더(730)로부터 선택된 예측 결과들 사이의 차이(잔차 데이터)를 계산하도록 구성된다. 잔차 인코더(724)는 잔차 데이터에 기반하여 동작하여 잔차 데이터를 인코딩하여 변환 계수들을 생성하도록 구성된다. 예에서, 잔차 인코더(724)는 잔차 데이터를 공간 도메인에서 주파수 도메인으로 변환하고, 변환 계수들을 생성하도록 구성된다. 이어서, 변환 계수들은 양자화 프로세싱을 거쳐 양자화된 변환 계수들을 획득한다. 다양한 실시예들에서, 비디오 인코더(703)는 또한 잔차 디코더(728)를 포함한다. 잔차 디코더(728)는 역변환을 수행하고, 디코딩된 잔차 데이터를 생성하도록 구성된다. 디코딩된 잔차 데이터는 인트라 인코더(722) 및 인터 인코더(730)에 의해 적합하게 사용될 수 있다. 예를 들어, 인터 인코더(730)는 디코딩된 잔차 데이터 및 인터 예측 정보에 기반하여 디코딩된 블록들을 생성할 수 있고, 인트라 인코더(722)는 디코딩된 잔차 데이터 및 인트라 예측 정보에 기반하여 디코딩된 블록들을 생성할 수 있다. 디코딩된 블록들은 디코딩된 화상들을 생성하도록 적합하게 프로세싱되고 디코딩된 화상들은 메모리 회로(도시되지 않음)에 버퍼링될 수 있고 일부 예들에서 참조 화상으로서 사용될 수 있다.

[0075] 엔트로피 인코더(725)는 인코딩된 블록을 포함하도록 비트스트림을 포맷하도록 구성된다. 엔트로피 인코더(725)는 HEVC 표준과 같은 적합한 표준에 따라 다양한 정보를 포함하도록 구성된다. 예에서, 엔트로피 인코더(725)는 일반 제어 데이터, 선택된 예측 정보(예를 들어, 인트라 예측 정보 또는 인터 예측 정보), 잔차 정보, 및 다른 적합한 정보를 비트스트림에 포함하도록 구성된다. 개시된 주제에 따르면, 인터 모드 또는 이중-예측 모드의 병합 서브모드에서 블록을 코딩할 때, 잔차 정보가 없음이 주목된다.

[0076] 도 8은 본 개시내용의 다른 실시예에 따른 비디오 디코더(810)의 다이어그램을 도시한다. 비디오 디코더(810)는 코딩된 비디오 시퀀스의 일부인 코딩된 화상들을 수신하고, 코딩된 화상들을 디코딩하여 재구성된 화상들을 생성하도록 구성된다. 예에서, 비디오 디코더(810)는 도 4 예의 비디오 디코더(410) 대신에 사용된다.

- [0077] 도 8 예에서, 비디오 디코더(810)는 도 8에 도시된 바와 같이 함께 결합된 엔트로피 디코더(871), 인터 디코더(880), 잔차 디코더(873), 재구성 모듈(874) 및 인트라 디코더(872)를 포함한다.
- [0078] 엔트로피 디코더(871)는 코딩된 화상을 이루는 선택스 엘리먼트들을 나타내는 소정 심볼들을 코딩된 화상으로부터 재구성하도록 구성될 수 있다. 그러한 심볼들은 예를 들어, 블록이 코딩되는 모드(이를테면, 예를 들어, 인트라 모드, 인터 모드, 이중-예측 모드, 병합 서브모드 또는 다른 서브모드에서 후자의 2개), 각각 인트라 디코더(872) 또는 인터 디코더(880)에 의해 예측에 사용되는 소정 샘플 또는 메타데이터를 식별할 수 있는 예측 정보(이를테면, 예를 들어, 인트라 예측 정보 또는 인터 예측 정보), 예를 들어, 양자화된 변환 계수들 형태의 잔차 정보 등을 포함할 수 있다. 예에서, 예측 모드가 인터 또는 이중-예측 모드인 경우, 인터 예측 정보는 인터 디코더(880)에 제공되고; 예측 유형이 인트라 예측 유형인 경우, 인트라 예측 정보는 인트라 디코더(872)에 제공된다. 잔차 정보는 역양자화될 수 있고 잔차 디코더(873)에 제공된다.
- [0079] 인터 디코더(880)는 인터 예측 정보를 수신하고, 인터 예측 정보에 기반하여 인터 예측 결과들을 생성하도록 구성된다.
- [0080] 인트라 디코더(872)는 인트라 예측 정보를 수신하고, 인트라 예측 정보에 기반하여 인트라 예측 결과들을 생성하도록 구성된다.
- [0081] 잔차 디코더(873)는 역양자화를 수행하여 역양자화된 변환 계수들을 추출하고, 역양자화된 변환 계수들을 프로세싱하여 주파수 도메인에서 공간 도메인으로 잔차를 변환하도록 구성된다. 잔차 디코더(873)는 또한 (양자화 파라미터(QP)를 포함하기 위해) 소정 제어 정보를 요구할 수 있고, 그 정보는 엔트로피 디코더(871)에 의해 제공될 수 있다(데이터 경로는 낮은 블롭 제어 정보일 수 있으므로 묘사되지 않음).
- [0082] 재구성 모듈(874)은 공간 도메인에서, 잔차 디코더(873)에 의해 출력된 잔차와 예측 결과들(경우에 따라 인터 또는 인트라 예측 모듈들에 의해 출력됨)을 결합하여 재구성된 화상의 일부일 수 있고, 차례로 재구성된 비디오의 일부일 수 있는 재구성된 블록을 형성하도록 구성된다. 디블록킹(deblocking) 동작과 같은 다른 적합한 동작들이 시각적 품질을 개선하기 위해 수행될 수 있다는 것이 주목된다.
- [0083] 비디오 인코더들(403, 603 및 703), 및 비디오 디코더들(410, 510 및 810)이 임의의 적합한 기법을 사용하여 구현될 수 있다는 것이 주목된다. 실시예에서, 비디오 인코더들(403, 603, 및 703), 및 비디오 디코더들(410, 510, 및 810)은 하나 이상의 집적 회로들을 사용하여 구현될 수 있다. 다른 실시예에서, 비디오 인코더들(403, 603, 603), 및 비디오 디코더들(410, 510, 810)은 소프트웨어 명령들을 실행하는 하나 이상의 프로세서들을 사용하여 구현될 수 있다.
- [0084] 본 개시내용의 일부 양태들에 따르면, 팔레트 기반 코딩 모드로 지칭되는 코딩 모드는 인터 예측 모드 및 인트라 예측 모드에 더하여 사용될 수 있다. 본 개시내용은 로컬 이중 트리 코딩 구조로 팔레트 기반 코딩 모드를 단순화하는 기법들을 제공한다.
- [0085] 일부 예들에서, 스크린 콘텐츠들은 특정 특성을 가지며 스크린 코딩을 위한 코딩 도구들이 개발될 수 있다. 스크린 코딩을 위한 코딩 도구들은 코딩 효율성에서 상당한 이득들을 달성할 수 있다. 팔레트 기반 코딩 모드는 블록의 주요 컬러들을 포함하는 팔레트에 대한 인덱스들을 사용하여 픽셀들의 블록을 나타낼 수 있다. 팔레트와 인덱스들은 공간 중복을 이용하여 인코딩될 수 있다.
- [0086] 특히, 일부 실시예들에서, 팔레트는 인덱스들을 픽셀 값들과 연관시킬 수 있는 엔트리들의 룩업 테이블로 정의된다. 일부 예들에서, 픽셀 값은 컬러 성분들에 대응하는 하나 이상의 값들을 포함한다. 예에서, 픽셀 값은 3개의 컬러 성분들을 각각 나타내는 3개의 값들을 포함할 수 있다. 다른 예에서, 모노크롬 팔레트의 경우, 픽셀 값은 단일 컬러 성분을 나타내는 값을 포함한다.
- [0087] 일부 실시예들에서, 픽셀들의 블록은 팔레트에 기반하여 인코딩될 수 있고, 팔레트를 사용하는 코딩 모드는 팔레트 기반 코딩 모드로 지칭된다. 팔레트 기반 코딩 모드는 VVC 등과 같은 일부 비디오 코딩 표준들에서 채택된다.
- [0088] 일부 실시예들에 따르면, 팔레트 기반 코딩 모드는 독립형 코딩 모드이다. 실시예에서, 팔레트 기반 코딩 모드는 인트라 인코더(722), 인터 인코더(730), 인트라 디코더(872), 인터 디코더(870)와 별개인 코딩 유닛(예를 들어, 팔레트 기반 인코더, 팔레트 기반 디코더 등)으로서 구현된다. 예에서, 팔레트 기반 코딩 모드는 인트라 인코더(722), 인터 인코더(730), 인트라 디코더(872), 인터 디코더(870)와 별개의 하드웨어 회로로서 구현된다. 다른 예에서, 팔레트 기반 코딩 모드는 인트라 인코더(722), 인터 인코더(730), 인트라 디코더(872), 인터 디코

더(870)에 대한 소프트웨어 모듈들과 별개의 소프트웨어 모듈로서 구현된다. 팔레트 기반 코딩 모드에 대한 소프트웨어 모듈 및 다른 소프트웨어 모듈들은 동일한 프로세서 또는 상이한 프로세서들에 의해 실행될 수 있다.

[0089] 본 개시내용의 일부 양태들에 따르면, 팔레트 기반 코딩 모드는 일반적으로 2개의 부분들: 팔레트에 대한 코딩 방법들의 제1 부분 및 팔레트를 사용하는 샘플들에 대한 코딩 방법들의 제2 부분을 포함한다. 제1 부분은 주요 컬러 선택 및 팔레트 코딩을 포함할 수 있다. 제2 부분은 팔레트 인덱스 코딩, 실행 길이 코딩 및 이스케이프 픽셀 코딩(escape pixel coding)을 포함할 수 있다.

[0090] 일부 예들에서, 팔레트 기반 인코더는 블록에 대한 팔레트를 결정하고(예를 들어, 팔레트를 명시적으로 코딩하고, 팔레트를 예측하고, 시그널링과 예측의 조합 등), 픽셀 값들 중 하나 이상을 나타내는 팔레트에 엔트리를 위치시키고, 블록의 픽셀 값을 나타내는 데 사용되는 팔레트의 엔트리들을 나타내는 인덱스 값들로 블록을 인코딩함으로써 비디오 데이터의 블록을 인코딩할 수 있다. 일부 예들에서, 팔레트 기반 인코더는 인코딩된 비트스트림에서 팔레트 및/또는 인덱스 값들을 시그널링할 수 있다. 차례로, 팔레트 기반 디코더는 인코딩된 비트스트림으로부터, 블록에 대한 팔레트뿐 아니라 블록의 개별 픽셀들에 대한 인덱스 값들을 획득할 수 있다. 일부 다른 예들에서, 팔레트는 예측될 수 있고, 팔레트 기반 인코더는 팔레트를 시그널링하지 않고 인코딩된 비트스트림의 인덱스 값들을 시그널링할 수 있고 팔레트 기반 인코더는 팔레트를 예측하고 인코딩된 비트스트림의 인덱스 값들을 수신할 수 있다. 이어서, 팔레트 기반 디코더는 픽셀들의 인덱스 값들을 픽셀 값을 제공하는 팔레트의 엔트리들에 관련시킬 수 있다. 팔레트 기반 디코더는 인덱스 값들과 연관된 픽셀 값들에 기반하여 블록의 다양한 픽셀들을 재구성할 수 있다.

[0091] 일부 실시예들에 따르면, 팔레트는 주어진 블록에서 가장 지배적인 픽셀 값을 포함할 수 있다. 예를 들어, 가장 지배적인 픽셀 값들은 주어진 블록 내에서 가장 자주 발생하는 하나 이상의 픽셀 값을 포함할 수 있다. 또한, 일부 예들에서, 임계값(예를 들어, 픽셀 값을 갖는 픽셀들의 수)은 픽셀 값이 블록에서 가장 지배적인 픽셀 값들 중 하나로서 포함되어야 하는지 여부를 결정하기 위해 사용될 수 있다.

[0092] 일부 실시예들에서, 히스토그램 기반 알고리즘은 팔레트를 형성하기 위해 블록 내 픽셀들의 픽셀 값을 분류하는 데 사용된다. 일부 예들에서, 히스토그램에서 최상위 L 피크 값들은 주요 컬러들로 선택될 수 있다(L은 양의 정수). 예에서, 주요 컬러에 가까운 픽셀 값들은 주요 컬러로 양자화될 수 있다. 일부 예들에서 주요 컬러들에 속하지 않는 픽셀들은 이스케이프 픽셀들이라고 지칭된다. 이스케이프 픽셀들은 코딩 전에 양자화될 수 있다. 일부 실시예들에서, 무손실 코딩의 경우, 양자화 프로세스들은 수행되지 않는다.

[0093] 또한, 일부 실시예들에서, 각각의 픽셀에 대해, 픽셀이 어느 컬러에 속하는지를 나타내기 위해 컬러 인덱스가 할당된다. 일부 예들에서, L 주요 컬러가 사용되는 경우, 0 내지 (L-1)은 주요 컬러들에 대한 컬러 인덱스들로 사용될 수 있고, 0 내지 (L-1)은 주요 컬러 세트로 지정될 수 있다. 주요 컬러 인덱스들이 존재하지 않는 픽셀들의 경우, 특정 인덱스(예를 들어, 0 내지 (L-1)까지가 아닌 인덱스 N)가 할당되고 이러한 픽셀들은 '이스케이프된 픽셀들'이라고 호칭된다.

[0094] 일반적으로, 팔레트는 컬러 룩업 테이블로 구현된다. 컬러 룩업 테이블은 복수의 엔트리들을 포함한다. 각각의 엔트리는 컬러 인덱스를 컬러(예를 들어, 주요 컬러)와 연관시킨다. 일부 예들에서, 단일 팔레트는 루마 및 크로마 성분들 둘 모두를 코딩하는데 사용된다. 각각의 엔트리는 3개의 컬러 성분을 사용하여 특정 RGB(YUV) 컬러를 나타낼 수 있다. 예를 들어, 팔레트는 순수한 흑색을 나타내는 (R, G, B) = (0, 0, 0)인 엔트리 1을 가질 수 있고, 푸르스름한 컬러를 나타내는 (R, G, B) = (2, 10, 200)인 엔트리 0을 가질 수 있다. 비디오 포맷이 420일 때, 크로마 평면은 팔레트에 대한 컬러 룩업 테이블을 생성하기 위해 업샘플링된다는 것이 주목된다.

[0095] 일부 실시예들에서, 팔레트 기반 코딩은 CU 기반으로 수행된다. 예에서, 현재 CU에 대해, 현재 CU에서 가장 지배적인 픽셀 값을 포함하는 컬러 룩업 테이블이 도출된다. 일부 예들에서, 컬러 룩업 테이블의 크기 및 엘리먼트들이 송신될 수 있다. 일부 다른 예들에서, 컬러 룩업 테이블의 크기 및 엘리먼트들은 다른 CU들의 컬러 룩업 테이블의 크기 및/또는 엘리먼트들을 사용하여 예측적으로 코딩될 수 있다.

[0096] 본 개시내용의 일부 양태들에 따르면, 팔레트 예측은 팔레트 예측기 리스트에 기반하여 수행될 수 있다. 예에서, 팔레트 예측기 리스트는 다른 CU들에 사용되는 다중 팔레트들을 포함할 수 있다. 팔레트 예측기 리스트를 참조로 사용하여 현재 팔레트를 코딩하기 위해, 이진 벡터는, 팔레트 예측기 리스트의 각각의 엔트리가 현재 팔레트에서 재사용되는지 여부를 나타내는 데 사용된다.

[0097] 일부 예들(예를 들어, HEVC SCC)에서, 팔레트 예측기 리스트는 현재 팔레트를 예측하기 위한 참조들로서 이전에 코딩된 팔레트 엔트리들을 저장한다. 팔레트 예측기 리스트는 각각의 팔레트 모드 CU 이후에 업데이트된다. 예

에서, 팔레트 예측기 리스트는 가장 최근에 사용된 캐시와 유사한 방식으로 동작한다. 최신 팔레트는 팔레트 예측기 리스트의 시작 부분에 삽입할 수 있고 리스트 크기가 임계치를 초과하면 스캔 순서에서 가장 먼 CU들로부터의 엔트리들이 버려진다. 일부 예들에서, 최대 허용 팔레트 예측기 크기 및 팔레트 크기는 고레벨 선택으로 시그널링되거나, 인코더 및 디코더들 둘 모두에 의해 합의된 정의된 값들로서 시그널링된다. 실시예에서, 팔레트 예측기 리스트의 크기의 상한은 SPS에서 (직접적으로 또는 간접적으로) 시그널링된다. 예를 들어, 팔레트 예측기 리스트는 팔레트에 대한 크기 제한의 대략 2배 크기를 가질 수 있다. 일부 예들에서, 팔레트 예측기 리스트는 또한 히스토리 팔레트 예측기 리스트(HPPL)로 지칭된다. 예에서, SPS에서, 최대 예측기 팔레트 크기는 63으로 시그널링되고, 최대 팔레트 크기는 31로 시그널링된다. 다른 예에서, 이 2개의 값들은 시그널링 없이 인코더 및 디코더에 의해 가정된다.

[0098] 일부 실시예들에서, 팔레트 예측 리스트의 엔트리들은 각각 재사용 플래그를 가질 수 있다. 엔트리에 대한 재사용 플래그가 참(예를 들어, "1")인 경우, 그 엔트리는 재사용 엔트리로 지칭된다. 팔레트 예측 리스트에서 재사용된 엔트리들의 컬러 성분들은 현재 팔레트의 컬러 루프 테이블에 복사될 수 있다. 재사용된 엔트리들은 팔레트 예측기 리스트에서 그 순서를 유지하는 방식으로 현재 팔레트의 시작 부분에 배치된다. 현재 팔레트에서 재사용된 엔트리를 다음에 팔레트 예측기 리스트에 없는 새로운 팔레트 엔트리들이 올 수 있다. 새로운 팔레트 엔트리들은 시그널링될 수 있다. 각각의 새로운 팔레트 엔트리는 예를 들어 3개의 컬러 성분을 포함할 수 있다.

[0099] 일부 실시예들에서, 미리 정의된 엔트리들을 갖는 팔레트 초기화기는 팔레트 예측기 리스트를 초기화하기 위해 사용될 수 있다. 팔레트 초기화기의 사용은 일부 시나리오들에서 코딩 효율성을 향상시킬 수 있다. 팔레트 초기화기는 예를 들어 화상 파라미터 세트(PPS), 시퀀스 파라미터 세트(SPS) 등에서 시그널링될 수 있다.

[0100] 컬러 인덱스들은 아래와 같이 인코딩될 수 있다. 분류 후, 블록의 픽셀들은 선택된 주요 컬러 세트에 따라 컬러 인덱스들로 변환될 수 있다. 일부 예들에서, 예측 코딩 방법은 컬러 인덱스들에 적용될 수 있고, 여기서 픽셀 라인은 수평 모드(예를 들어, 복사 인덱스 모드), 수직 모드(예를 들어, 복사 어보브 모드(copy above mode)), 및 일반 모드(예를 들어, 이스케이프 모드)를 포함하는 다수의 상이한 모드들(예를 들어, 3개의 상이한 모드들)에 의해 예측될 수 있다. 일부 예들에서, 컬러 인덱스들을 코딩할 때 2개의 인덱스 스캔 순서들(예를 들어, 수평 횡단 스캔 및 수직 횡단 스캔)이 사용된다. 인덱스 회전 플래그는 2개의 인덱스 스캔 순서들 중 어느 것이 사용되는지를 나타내기 위해 시그널링될 수 있다.

[0101] 복사 인덱스 모드에서, 제1 픽셀부터 시작하여, 제1 픽셀에서 하나 이상의 연속 인덱스들이 복사될 수 있다. 제1 픽셀의 컬러 인덱스가 시그널링될 수 있다.

[0102] 복사 어보브 모드에서, 하나 이상의 연속 컬러 인덱스들은 어보브 픽셀 라인, 예를 들어 현재 픽셀 라인 위에 있는 픽셀 라인에서 복사될 수 있다.

[0103] 이스케이프 모드에서, 예를 들어 이스케이프 픽셀이 조우할 때, 예를 들어 주요 색상 세트에서 가장 큰 인덱스(예를 들어, N)로 시그널링될 때, 대응하는 픽셀 값은 가장 큰 인덱스(예를 들어, N) 이후에 코딩될 수 있다. CU들에는 컬러 값들이 다른 다수의 이스케이프 픽셀들이 있을 수 있다. 상이한 이스케이프 픽셀 위치들의 경우, 이스케이프 픽셀들의 픽셀 값들이 상이할 수 있다.

[0104] 각각의 복사 인덱스 모드에 대해, 인덱스 값이 시그널링될 수 있다. 인덱스 시그널링은 예를 들어 컨텍스트-적응 이진 산술 코딩(CABAC) 스루풋을 개선하기 위해 앞(또는 시작)에서 그룹화될 수 있다. 유사하게, 이스케이프 픽셀들의 픽셀 값들은 예를 들어 CABAC 스루풋을 개선하기 위해 후면에서 시그널링될 수 있다. 복사 인덱스 모드 및 복사 어보브 모드는 인덱스 코딩과 이스케이프 코딩 사이에서 시그널링될 수 있다.

[0105] 실시예에서, 코딩 트리 방식은 루마 성분 및 대응하는 크로마 성분(들)이 별개의 블록 트리 구조들을 가질 수 있는 능력을 지원한다. 예에서, P 및 B 슬라이스들의 경우, CTU의 루마 및 크로마 CTB들은 동일한 코딩 트리 구조(예를 들어, 단일 트리)를 공유한다. I 슬라이스들의 경우, CTU의 루마 및 크로마 CTB들은 별개의 블록 트리 구조들(예를 들어, 이중 트리)을 가질 수 있고, 별개의 블록 트리 구조들을 사용하는 CTU의 파티션 경우는 이중 트리 파티션으로 지칭된다. 예에서, 이중 트리 파티션이 적용되는 경우, 루마 CTB는 루마 코딩 트리 구조에 의해 루마 CU들로 파티셔닝될 수 있고, 크로마 CTB들은 크로마 코딩 트리 구조에 의해 크로마 CU들로 파티셔닝될 수 있다.

[0106] JVET에서와 같은 일부 예들에서, 크로마 성분의 샘플들은 루마 성분과 비교하여 독립적인 또는 별개의 분할 트리 구조(또는 코딩 트리 구조)를 가질 수 있다. 별개의 코딩 트리 구조는 CTU 레벨에서 시작될 수 있고, CTU 레벨로부터의 이중 트리 구조는 일부 예들에서 글로벌 이중 트리 구조로 지칭된다. 예에서, 크로마 CU(예를 들어,

2개의 크로마 성분들만을 포함하는 CU)는 대응하는 샘플 위치에서 크로마 CU의 루마 대응물보다 크다.

[0107] CTU 레벨 이중-트리(글로벌 이중 트리 구조)의 일부 예들에서, 최대 허용 예측기 팔레트 크기 및 팔레트 크기는 예를 들어 (루마 또는 크로마의) 각각의 채널에 대해 복잡성이 감소되도록 절반으로 감소된다. 예를 들어, 단일 트리 코딩의 경우, 루마 및 크로마 팔레트 코딩된 블록들은 공동으로 코딩되고, 최대 허용 예측기 팔레트 크기는 63개 엔트리들이고 최대 허용 팔레트 크기는 31개 엔트리들이다. 이중-트리 코딩의 경우, 루마 팔레트 코딩된 블록들의 경우, 최대 허용 예측기 팔레트 크기는 31개 엔트리들이고 최대 허용 팔레트 크기는 15개 엔트리들이고; 크로마 팔레트 코딩된 블록들의 경우, 최대 허용 예측기 팔레트 크기는 31개 엔트리들이고 최대 허용 팔레트 크기는 15개 엔트리이다.

[0108] 본 개시내용의 일부 양태들에 따르면, ( $4 \times 4$  크로마 샘플들보다 작은 CU와 같은) 작은 크로마 블록들의 사용을 피하기 위해 로컬 이중 트리로 지정되는 기법이 사용된다. 예에서, 일부 조건들이 충족될 때(이를테면 부모 CU 크기는  $4 \times 4$  샘플들보다 작은 크로마 블록을 유발할 수 있는 일부 임계치 이하임), 로컬 이중 트리 기법이 트리거될 수 있다. 그 다음부터, CTU 레벨에서 이중-트리와 유사한 방식으로 CU에 대해 루마 및 크로마 코딩이 분리될 수 있다.

[0109] 본 개시내용의 양태에 따르면, 분할 트리 구조들이 사용되는 경우, 루마 성분들은 인트라 예측 모드, IBC 모드 및 팔레트 기반 코딩 모드 중 하나로 코딩될 수 있고, 크로마 성분들은 인트라 예측 모드 및 팔레트 기반 코딩 모드 중 하나로 코딩될 수 있다. 그러나, 로컬 이중-트리의 사용은 팔레트 기반 코딩 모드에서의 동작들을 복잡하게 한다. 예를 들어, 로컬 이중 트리 구조의 블록은 단일 트리 구조에서 이웃 블록들을 가질 수 있다. 동일한 팔레트 예측기 리스트는 로컬 이중 트리 구조의 블록과 단일 트리 구조의 이웃 블록들에 의해 사용될 수 있다. 팔레트 예측기 리스트의 업데이트는 복잡할 수 있다.

[0110] 본 개시내용의 양태들은 로컬 이중 트리 코딩 구조로 팔레트 기반 코딩을 단순화하는 기법들을 제공한다.

[0111] 본 개시내용의 양태에 따르면, 일부 조건들이 참일 때, 팔레트 예측기 리스트 업데이트 프로세스는 팔레트 모드 코딩을 위해 비활성화되거나 바이패스된다. 일부 예들에서, 로컬 이중-트리가 트리거될 수 있고, 로컬 이중 트리 구조의 블록들이 팔레트 기반 코딩 모드에서 코딩될 때, 팔레트 예측기 리스트의 업데이트 프로세스는 비활성화될 수 있다. 따라서, 로컬 이중 트리 구조 하에서, 팔레트 기반 코딩 모드에서 코딩된 루마 블록들 및/또는 팔레트 기반 코딩 모드에서 코딩된 크로마 블록들의 경우, 이를 블록들의 팔레트 엔트리들은 팔레트 예측기 리스트에서 팔레트 엔트리들을 업데이트하는 데 사용되지 않는다.

[0112] 실시예에서, 로컬 이중 트리의 트리거가 검출될 수 있다. 예에서, 로컬 이중 트리가 트리거될 때, 팔레트 예측기 리스트는 로컬 이중 트리 구조 하에서 팔레트 기반 코딩 모드에서 코딩되는 크로마 블록들에 기반하여 업데이트되지 않고 로컬 이중 트리 구조 하에서 팔레트 기반 코딩 모드에서 코딩되는 루마 블록들에 기반하여 업데이트되지 않는다. 다른 예에서, 로컬 이중 트리가 트리거될 때, 팔레트 예측기 리스트는 로컬 이중 트리 구조 하에서 팔레트 기반 코딩된 루마 블록들에 기반하여 업데이트되지 않고, 로컬 이중 트리 구조에서 팔레트 기반 코딩 모드에서 코딩된 크로마 블록들에 기반하여 업데이트될 수 있다. 다른 예에서, 로컬 이중 트리가 트리거될 때, 팔레트 예측기 리스트는 로컬 이중 트리 구조 하에서 팔레트 기반 코딩 모드에서 코딩되는 크로마 블록들에 기반하여 업데이트되지 않고 로컬 이중 트리 구조 하에서 팔레트 기반 코딩 모드에서 코딩되는 루마 블록들에 기반하여 업데이트될 수 있다.

[0113] 특히, 예에서 localDualTree로 표시된 변수는 로컬 이중 트리가 트리거될 수 있는지 여부를 검출하는 데 사용된다. 예에서 변수 localDualTree는 방정식 1에 따라 도출된다:

[0114] `localDualTree = treeType != SINGLE_TREE && ( slice_type != I || ( slice_type == I && qtbt_dual_tree_intra_flag == 0 ) ) ? 1 : 0` (방정식 1)

[0115] 여기서 treeType은 분할 트리 구조의 유형(예를 들어, SINGLE\_TREE, DUAL\_TREE\_LUMA, DUAL\_TREE\_CHROMA 등)을 나타내고, slice\_type은 현재 슬라이스의 유형(예를 들어, I, P 또는 B)을 나타내고 qtbt\_dual\_tree\_intra\_flag는 글로벌 이중 트리에 대한 플래그를 나타낸다. 변수 localDualTree가 1과 같으면, 로컬 이중 트리가 트리거되고 검출될 수 있고; 변수 localDualTree가 0과 같으면, 로컬 이중 트리가 트리거되지 않는다.

[0116] 도 9는 일부 예들에서 팔레트 예측 리스트를 업데이팅하기 위한 선택스 예(900)를 도시한다. 일부 예들에서, localDualtree가 0과 같을 때, 팔레트 예측 리스트는 선택스 예(900)에 따라 업데이트될 수 있고;

localDualtree가 1과 같을 때, 신팩스 예(900)에 따른 업데이트 프로세스는 스kip된다.

[0117] 예에서, 단일 트리 구조 하에서 팔레트 코딩된 블록에 대해, 팔레트 예측 리스트는 블록에 기반하여 업데이트될 수 있다. 예를 들어, treeType은 SINGLE\_TREE와 같으므로, localDualtree는 0과 같다. 또한, 변수 startComp(예를 들어, 시작 컬러 성분)은 0으로 설정되고, 변수 numComps(예를 들어, 컬러 성분들의 수)는 루마 성분만 코딩된 경우 1로 설정되고, 루마 성분과 크로마 성분들 둘 모두가 코딩된 경우 3으로 설정된다. 이어서, 팔레트 예측기 리스트의 크기(예를 들어, PredictorPaletteSize로 표시됨) 및 팔레트 예측기 리스트의 엔트리(예를 들어, predictorPaletteEntries로 표시됨)은 신팩스(900)에 따라 업데이트될 수 있다.

[0118] 다른 예에서, 글로벌 이중 트리 하에서 팔레트 코딩된 루마 블록에 대해, 팔레트 예측 리스트는 블록에 기반하여 업데이트될 수 있다. 예를 들어, CTU 레벨 이중 트리(글로벌 이중 트리)의 경우들에서, treeType은 DUAL\_TREE\_LUMA와 같고, slice\_type은 I와 같고, 글로벌 이중 트리 플래그 qtbt\_dual\_tree\_intra\_flag는 1과 같으므로, localDualtree는 0과 같다. 또한, 변수 startComp는 0으로 설정되고, numComps는 1로 설정된다. 이어서, 팔레트 예측기 리스트의 크기(예를 들어, PredictorPaletteSize로 표시됨) 및 팔레트 예측기 리스트의 엔트리(예를 들어, predictorPaletteEntries로 표시됨)는 신팩스(900)에 따라 업데이트될 수 있다.

[0119] 다른 예에서, 글로벌 이중 트리 하에서 팔레트 코딩된 크로마 블록에 대해, 팔레트 예측 리스트는 블록에 기반하여 업데이트될 수 있다. 예를 들어, CTU 레벨 이중 트리(글로벌 이중 트리)의 경우들에서, treeType은 DUAL\_TREE\_CHROMA와 같고, slice\_type은 I와 같고, 글로벌 이중 트리 플래그 qtbt\_dual\_tree\_intra\_flag는 1과 같으므로, localDualtree는 0과 같다. 또한, 변수 startComp는 1로 설정되고, numComps는 2로 설정된다. 이어서, 팔레트 예측기 리스트의 크기(예를 들어, PredictorPaletteSize로 표시됨) 및 팔레트 예측기 리스트의 엔트리(예를 들어, predictorPaletteEntries로 표시됨)은 신팹스(900)에 따라 업데이트될 수 있다.

[0120] 다른 예에서, 로컬 이중 트리 하에서 팔레트 코딩된 크로마 블록에 대해, 팔레트 예측 리스트의 업데이트는 스kip될 수 있다. 예를 들어 treeType은 DUAL\_TREE\_CHROMA이고, slice\_type은 I와 같고, 글로벌 이중 트리 플래그 qtbt\_dual\_tree\_intra\_flag는 0과 같으므로, localDualtree는 1이다. 이어서, 신팹스 예(900)에 따른 업데이트 프로세스가 스kip된다.

[0121] 본 개시내용의 다른 양태에 따르면, 로컬 이중-트리의 사용을 트리거할 수 있는 코딩 구성들이 사용 중일 때, 팔레트 기반 코딩 모드는 작은 블록들에 대해 허용되지 않는다. 따라서, 로컬 이중 트리의 경우 팔레트 예측기 리스트 업데이트의 복잡성은 회피될 수 있다.

[0122] 예에서, slice\_type이 I와 같지 않거나, slice\_type이 I와 같지만 (글로벌) 이중 트리(예를 들어, qtbt\_dual\_tree\_intra\_flag)를 활성화하기 위한 슬라이스 레벨 플래그가 0과 같으면, 로컬 이중 트리가 트리거될 수 있다(예를 들어, localDualTree는 1과 같음). 로컬 이중 트리가 트리거될 때, 폭×높이 < 임계치를 갖는 코딩 유닛 등과 같은 작은 블록들은 팔레트 기반 코딩 모드에 의해 코딩되는 것이 허용되지 않는다. 실시예에서, 임계치는 현재 코딩 트리의 블록 크기가 이 임계치를 초과할 때 로컬 이중-트리가 사용되지 않도록 정의된다. 일부 예들에서, 팔레트 기반 코딩 모드는 로컬 이중 트리 구조 하에서 작은 루마 블록들 및 작은 크로마 블록들에 대해 허용되지 않는다. 일부 예들에서, 팔레트 기반 코딩 모드는 로컬 이중 트리 구조 하에서 작은 크로마 블록들에만 허용되지 않는다.

[0123] 도 10은 비트스트림에서, 팔레트 기반 코딩 모드가 현재 블록에 사용되는지 여부를 나타내는 플래그 (pred\_mode\_plt\_flag)의 존재를 결정하기 위한 신팹스 예를 도시한다. 도 10 예에서, 부분(1001)은 크기 제약을 적용하는 데 사용되고, 부분(1002)은 크로마 블록 및 로컬 이중 트리 제약을 적용하는 데 사용된다. 예를 들어, 블록이 작을 때(예를 들어, 폭×높이가 루마 블록의 경우 16보다 작거나, 폭×높이가 크로마 블록의 경우 16×SubWidthC×SubHeightC보다 작은 경우), 부분(1001)은 0일 수 있으므로, 블록에 대한 pre\_mode\_plt\_flag가 비트스트림에 존재하지 않고, 팔레트 기반 코딩 모드는 그 블록에 대해 허용되지 않는다.

[0124] 예에서, 로컬 이중 트리 구조 하에서 크로마 블록의 경우, modeType은 MODE\_TYPE\_INTRA와 같고, treeType은 DUAL\_TREE\_CHROMA와 같으므로, 부분(1002)은 0이 될 수 있다. 따라서, 크로마 블록에 대한 pred\_mode\_plt\_flag는 비트스트림에 존재하지 않고, 따라서 팔레트 기반 코딩 모드는 크로마 블록에 대해 허용되지 않는다. 다른 예에서, 글로벌 이중 트리의 크로마 블록에 대해, modeType은 MODE\_TYPE\_ALL과 같고, treeType은 DUAL\_TREE\_CHROMA와 같으므로, 부분(1002)은 결과 1이 될 수 있다. 따라서, 글로벌 이중 트리 구조 하에서 크로마 블록은 팔레트 기반 코딩 모드에서 코딩될 수 있다. 다른 예에서, 로컬 이중 트리 구조 하의 루마 블록의 경우, modeType은 MODE\_TYPE\_INTRA와 같고 treeType은 DUAL\_TREE\_LUMA와 같으므로, 부분(1002)은 결과 1이 될 수

있다. 따라서, 로컬 이중 트리 구조 하에서 루마 블록은 팔레트 기반 코딩 모드에서 코딩될 수 있다.

[0125] 본 개시내용의 다른 양태에 따르면, 로컬 이중-트리가 사용될 때, 최대 허용 예측기 팔레트 크기 및 팔레트 크기는 예를 들어 절반으로 감소된다. 예측기 팔레트 크기와 팔레트 크기의 감소는 업데이트 복잡성을 감소시킬 수 있다. 일부 예들에서, 루마 및 크로마 블록들이 별개로 코딩되도록 하는(예를 들어, 로컬 이중 트리) 각각의 로컬 영역(CTU보다 작음)에 대해, 최대 허용 예측기 팔레트 크기 및 팔레트 크기는 공동 팔레트 케이스(루마 및 크로마 성분들 둘 모두를 갖는 팔레트를 사용)에 대한 것의 절반으로 설정된다.

[0126] 실시예에서, 최대 허용 예측기 팔레트 크기 및 팔레트 크기는 공동 팔레트 경우에 대해 각각 63 및 31이다. 이어서, 로컬 이중-트리 구조 하에서, 루마 블록들의 팔레트 기반 코딩 모드에 대해 최대 허용 예측기 팔레트 크기 및 팔레트 크기는 각각 31 및 15로 설정되고; 크로마 블록들의 팔레트 기반 코딩 모드에 대한 최대 허용 예측기 팔레트 크기 및 팔레트 크기는 각각 31 및 15로 설정된다.

[0127] 도 11은 본 개시내용의 일 실시예에 따른 프로세스(1100)를 서술하는 흐름도를 도시한다. 프로세스(1100)는 블록의 재구성에 사용될 수 있다. 다양한 실시예들에서, 프로세스(1100)는 단말 디바이스들(310, 320, 330 및 340)의 프로세싱 회로와 같은 프로세싱 회로에 의해 실행되고, 프로세싱 회로는 비디오 인코더(403)의 기능을 수행하고, 프로세싱 회로는 비디오 디코더(410)의 기능을 수행하고, 프로세싱 회로는 비디오 디코더(510)의 기능을 수행하고, 프로세싱 회로는 비디오 인코더(603)의 기능을 수행하는 등등이 있다> 일부 실시예들에서, 프로세스(1100)는 소프트웨어 명령들로 구현되고, 따라서 프로세싱 회로가 소프트웨어 명령들을 실행할 때, 프로세싱 회로는 프로세스(1100)를 수행한다. 프로세스는 S1101에서 시작하여 S1110로 진행한다.

[0128] S1110에서, 블록의 예측 정보는 코딩된 비디오 비트스트림으로부터 디코딩된다.

[0129] S1120에서, 블록에 대한 팔레트 기반 코딩 모드를 허용할지 여부는 블록의 크기와 임계치의 비교에 기반하여 결정된다.

[0130] S1130에서, 블록이 로컬 이중 트리 구조 하에 있는지 여부가 결정된다.

[0131] S1140에서, 블록은 블록에 대한 팔레트 기반 코딩 모드 허용 여부의 결정 및 블록이 로컬 이중 트리 구조 하에 있는지 여부의 결정에 기반하여 디코딩된다.

[0132] 일부 실시예에서, 블록에 대한 팔레트 기반 코딩 모드는 블록의 크기가 임계치보다 작은 것에 응답하여 허용되지 않는다. 예를 들어, 블록에 대한 조건(1001)이 충족되지 않으면, 팔레트 기반 코딩 모드는 허용되지 않는다. 일부 예들에서, 블록이 임계치보다 클 때 로컬 이중 트리 구조가 사용되지 않도록 임계치가 정의된다.

[0133] 실시예에서, 블록에 대한 팔레트 기반 코딩 모드는 블록이 크로마 블록이고 크로마 블록의 크기가 임계치보다 작은 것에 응답하여 허용되지 않는다. 예에서, 팔레트 기반 코딩 모드는 작은 루마 블록들에서 사용될 수 있지만 작은 크로마 블록들에서는 사용될 수 없다.

[0134] 다른 실시예에서, 팔레트 기반 코딩 모드는 블록이 로컬 이중 트리 구조 하에서 크로마 블록임을 나타내는 블록의 트리 유형과 모드 유형의 조합에 기반하여 블록에 대해 허용되지 않는다. 예에서, 블록을 디코딩하기 위해, 모드 유형이 MODE\_TYPE\_INTRA와 같고 트리 유형이 DUAL\_TREE\_CHROMA와 같을 때, 조건(1002)의 값이 0과 같도록, 블록에 대한 팔레트 기반 코딩 모드가 허용되지 않는다.

[0135] 일부 실시예들에서, 팔레트 기반 코딩 모드가 허용될 때, 팔레트 기반 코딩 모드가 블록에 사용되는지 여부를 나타내는 플래그(예를 들어, pred\_mode\_plt\_flag)는 디코딩된다. 플래그가 블록에 대한 팔레트 기반 코딩 모드의 사용을 나타내는 것에 응답하여, 블록은 팔레트 기반 코딩 모드를 사용하여 디코딩된다. 일부 실시예들에서, 블록이 로컬 이중 트리 구조 하에 있는 것에 응답하여, 팔레트 예측기 리스트의 업데이트는 스kip되거나 바이패스될 수 있다.

[0136] 일부 예들에서, 팔레트 예측기 리스트는 로컬 이중 트리 구조의 트리거가 없는 것에 응답하여 블록 업데이팅의 디코딩에 사용된 팔레트에 기반하여 업데이트된다. 예에서, 변수 localDualTree는 로컬 이중 트리가 트리거되는지 여부를 검출하는 데 사용된다. 예를 들어, 변수 localDualTree가 0인 경우, 블록의 현재 팔레트에 기반한 팔레트 예측기 리스트의 업데이트가 수행될 수 있다. 예에서, 변수 localDualTree가 1일 때, 블록의 현재 팔레트에 기반한 팔레트 예측기 리스트의 업데이트는 스kip된다. 다른 예에서, 변수 localDualTree가 1일 때, 팔레트 예측기 리스트는 블록이 루마 블록일 때 업데이트될 수 있고, 블록이 크로마 블록일 때 스kip된다.

[0137] 다른 실시예에서, 로컬 이중 트리 구조 하에서의 블록들, 팔레트의 최대 크기 및 팔레트 예측기 리스트의 최대

크기는 예를 들어 절반으로 감소된다. 일부 예들에서, 팔레트 기반 코딩 모드가 블록에 대해 허용될 때, 팔레트 기반 코딩 모드가 블록에 사용되는지 여부를 나타내는 플래그가 디코딩된다. 이어서, 플래그가 블록에 대한 팔레트 기반 코딩 모드의 사용을 나타내고 블록이 로컬 이중 트리 구조 하에 있는 경우, 블록은 팔레트에 대해 감소된 최대 크기를 갖는 팔레트에 기반하여 디코딩된다. 예를 들어 팔레트의 일반 최대 크기는 31이고, 팔레트의 감소된 최대 크기는 15이다. 또한, 팔레트 예측기 리스트는 블록의 디코딩에 사용된 팔레트에 기반하여 업데이트되고, 팔레트 예측기 리스트의 크기는 팔레트 예측기 리스트에 대해 감소된 최대 크기에 의해 제한된다. 예를 들어, 팔레트 예측기 리스트의 일반 최대 크기는 63이고, 팔레트 예측기 리스트의 감소된 최대 크기는 31이다.

[0138] 이어서, 프로세스는 S1199로 진행하여 종료한다.

[0139] 위에서 설명된 기법들은 컴퓨터-관독가능 명령을 사용하여 컴퓨터 소프트웨어로서 구현될 수 있고 하나 이상의 컴퓨터-관독가능 매체에 물리적으로 저장될 수 있다. 예를 들어, 도 12는 개시된 주제의 소정 실시예를 구현하기에 적합한 컴퓨터 시스템(1200)을 도시한다.

[0140] 컴퓨터 소프트웨어는 하나 이상의 컴퓨터 중앙 처리 장치(CPU), 그래픽 처리 장치(GPU) 등에 의해 어셈블리, 컴파일, 링킹될 수 있는 임의의 적합한 기계 코드 또는 컴퓨터 언어, 또는 마이크로-코드 실행이 직접 실행되거나 해석을 통해 실행될 수 있는 명령들을 포함하는 코드를 생성하는 유사한 메커니즘 등을 사용하여 코딩될 수 있다.

[0141] 명령들은 예를 들어 개인용 컴퓨터, 태블릿 컴퓨터, 서버들, 스마트폰들, 게이밍 디바이스들, 사물인터넷 디바이스들 등을 포함하는 다양한 유형들의 컴퓨터들 또는 그 컴퓨트트들에서 실행될 수 있다.

[0142] 컴퓨터 시스템(1200)에 대한 도 12에 도시된 컴퓨트트들은 본질적으로 예시적인 것이며 본 개시내용의 실시예들을 구현하는 컴퓨터 소프트웨어의 사용 또는 기능의 범위에 대한 어떠한 제한도 제시하도록 의도되지 않는다. 컴퓨트트들의 구성이 컴퓨터 시스템(1200)의 예시적인 실시예에 예시된 컴퓨트트들 중 임의의 하나 또는 조합과 관련된 임의의 종속성 또는 요건을 갖는 것으로 해석되어서도 안 된다.

[0143] 컴퓨터 시스템(1200)은 특정 인간 인터페이스 입력 디바이스들을 포함할 수 있다. 이러한 인간 인터페이스 입력 디바이스는 예를 들어 촉각 입력(이를테면: 키스트로크들, 스와이프들, 데이터 장갑 움직임들), 오디오 입력(이를테면: 음성, 박수), 시각적 입력(이를테면: 제스처들), 후각적 입력(묘사되지 않음)을 통해 한 명 이상의 인간 사용자에 의한 입력에 응답할 수 있다. 인간 인터페이스 디바이스들은 또한 오디오(이를테면: 음성, 음악, 주변 사운드), 이미지들(이를테면: 스캔 이미지들, 스틸 이미지 카메라로부터 획득된 사진 이미지들), 비디오(이를테면: 2-차원 비디오, 입체 비디오를 포함하는 3-차원 비디오) 같은 반드시 인간에 의한 의식적인 입력에 직접 관련되지 않는 소정의 매체에 사용될 수 있다.

[0144] 입력 인간 인터페이스 디바이스들은 키보드(1201), 마우스(1202), 트랙패드(1203), 터치 스크린(1210), 데이터-글러브(도시되지 않음), 조이스틱(1205), 마이크(1206), 스캐너(1207), 카메라(1208) 중 하나 이상(각각 하나만 도시됨)을 포함할 수 있다.

[0145] 컴퓨터 시스템(1200)은 또한 소정 인간 인터페이스 출력 디바이스들을 포함할 수 있다. 이러한 인간 인터페이스 출력 디바이스들은 예를 들어 촉각 출력, 사운드, 광, 및 냄새/미각을 통해 하나 이상의 인간 사용자들의 감각들을 자극할 수 있다. 그러한 인간 인터페이스 출력 디바이스들은 촉각 출력 디바이스들(예를 들어 터치 스크린(1210), 데이터-글러브(도시되지 않음), 또는 조이스틱(1205)에 의한 촉각 피드백, 그러나 또한 입력 디바이스들로 역할을 하지 않는 촉각 피드백 디바이스들이 있을 수 있음), 오디오 출력 디바이스들(이를테면: 스피커(1209), 헤드폰들(도시되지 않음)), 시각 출력 디바이스들(이를테면 각각 터치-스크린 입력 능력을 갖거나 가지지 않고, 각각 촉각 피드백 능력을 갖거나 가지지 않는 CRT 스크린들, LCD 스크린들, 플라즈마 스크린들, OLED 스크린들을 포함하는 스크린들(1210)-이들 중 일부는 입체 출력 같은 수단을 통해 2 차원 시각 출력 또는 3 차원 초과의 출력을 출력할 수 있음; 가상-현실 안경(묘사되지 않음), 홀로그래픽 디스플레이들 및 스모크 텡크들(묘사되지 않음)), 및 프린터들(묘사되지 않음)을 포함할 수 있다.

[0146] 컴퓨터 시스템(1200)은 또한 인간 액세스가능 저장 디바이스들 및 이의 연관된 매체, 이를테면 CD/DVD를 갖는 CD/DVD ROM/RW(1220) 또는 이와 유사한 매체(1221)를 포함하는 광학 매체, 썬-드라이브(1222), 이동식 하드 드라이브 또는 솔리드 스테이트 드라이브(1223), 테이프 및 플로피 디스크(묘사되지 않음)와 같은 레거시 자기 매체, 보안 동글들(묘사되지 않음)과 같은 특수 ROM/ASIC/PLD 기반 디바이스들 등을 포함할 수 있다.

[0147] 통상의 기술자들은 또한 현재 개시된 주제와 관련하여 사용되는 "컴퓨터 관독가능 매체"라는 용어가 송신 매체,

반송파들, 또는 다른 일시적 신호들을 포함하지 않는다는 것을 이해해야 한다.

[0148] 컴퓨터 시스템(1200)은 또한 하나 이상의 통신 네트워크들에 대한 인터페이스를 포함할 수 있다. 네트워크들은 예를 들어 무선, 유선, 광일 수 있다. 네트워크들은 또한 로컬, 광역, 대도시, 차량 및 산업, 실시간, 지연-허용 등이 될 수 있다. 네트워크들의 예들은 이더넷과 같은 근거리 통신망, 무선 LAN들, GSM, 3G, 4G, 5G, LTE 등을 포함하는 셀룰러 네트워크들, 케이블 TV, 위성 TV 및 지상파 방송 TV를 포함하는 TV 유선 또는 무선 광역 디지털 네트워크들, CANBus 등을 포함하는 차량 및 산업용을 포함한다. 소정 네트워크들은 일반적으로 소정 범용 데이터 포트들 또는 주변 버스들(1249)(이를테면, 예를 들어 컴퓨터 시스템(1200)의 USB 포트들)에 부착된 외부 네트워크 인터페이스 어댑터들을 요구하고; 다른 것들은 일반적으로 아래에 설명된 바와 같이 시스템 버스에 부착함으로써 컴퓨터 시스템(1200)의 코어에 통합된다(예를 들어, PC 컴퓨터 시스템에 대한 이더넷 인터페이스 또는 스마트폰 컴퓨터 시스템에 대한 셀룰러 네트워크 인터페이스). 이러한 네트워크들 중 임의의 것을 사용하여, 컴퓨터 시스템(1200)은 다른 엔티티들과 통신할 수 있다. 그러한 통신은 단방향, 수신 전용(예를 들어, 방송 TV), 단방향 송신 전용(예를 들어 CANbus에서 소정 CANbus 디바이스들로), 또는 양방향(예를 들어 로컬 또는 광역 디지털 네트워크들을 사용하여 다른 컴퓨터 시스템들로)일 수 있다. 소정 프로토콜들 및 프로토콜 스택들은 위에서 설명된 바와 같이 각각의 이들 네트워크들 및 네트워크 인터페이스들에서 사용될 수 있다.

[0149] 컴퓨터 시스템(1200)의 코어(1240)에는 전술한 인간 인터페이스 디바이스들, 인간-액세스가능 저장 디바이스들, 및 네트워크 인터페이스들이 부착될 수 있다.

[0150] 코어(1240)는 하나 이상의 중앙 처리 장치(CPU)(1241), 그래픽 처리 장치(GPU)(1242), FPGA(Field Programmable Gate Areas)(FPGA) 형태의 특수 프로그램가능 프로세싱 유닛들(1243), 소정 작업들을 위한 하드웨어 가속기들(1244) 등을 포함할 수 있다. 이러한 디바이스들은 판독-전용 메모리(ROM)(1245), 랜덤-액세스 메모리(1246), 사용자가 액세스할 수 없는 내부 하드 드라이브들, SSD들 등과 같은 내부 대용량 저장장치(1247)와 함께, 시스템 버스(1248)를 통해 연결될 수 있다. 일부 컴퓨터 시스템들에서, 시스템 버스(1248)는 추가 CPU들, GPU 등에 의한 확장들을 가능하게 하는 하나 이상의 물리적 플러그들의 형태로 액세스할 수 있다. 주변 디바이스들은 코어의 시스템 버스(1248)에 직접 부착되거나 주변 버스(1249)를 통해 부착될 수 있다. 주변 버스를 위한 아키텍처들은 PCI, USB 등을 포함한다.

[0151] CPU들(1241), GPU들(1242), FPGA들(1243) 및 가속기들(1244)은 조합하여 전술한 컴퓨터 코드를 구성할 수 있는 소정 명령들을 실행할 수 있다. 해당 컴퓨터 코드는 ROM(1245) 또는 RAM(1246)에 저장될 수 있다. 과도 데이터는 또한 RAM(1246)에 저장될 수 있는 반면, 영구 데이터는 예를 들어 내부 대용량 저장장치(1247)에 저장될 수 있다. 하나 이상의 CPU(1241), GPU(1242), 대용량 저장장치(1247), ROM(1245), RAM(1246) 등과 밀접하게 연관된 캐시 메모리의 사용을 통해 메모리 디바이스들 중 임의의 것에 대한 고속 저장 및 검색이 가능해질 수 있다.

[0152] 컴퓨터 판독가능 매체는 다양한 컴퓨터-구현 동작들을 수행하기 위한 컴퓨터 코드를 가질 수 있다. 매체 및 컴퓨터 코드는 본 개시내용의 목적들을 위해 특별히 설계되고 구성된 것일 수 있거나, 컴퓨터 소프트웨어 기술분야들의 통상의 기술자들에게 잘 알려져 있고 이용가능한 종류일 수 있다.

[0153] 제한이 아닌 예로서, 아키텍처(1200), 특히 코어(1240)를 갖는 컴퓨터 시스템은 하나 이상의 유형의 컴퓨터-판독가능 매체에 구현된 소프트웨어를 실행하는 프로세서(들)(CPU들, GPU들, FPGA, 가속기들 등을 포함함)의 결과로서 기능을 제공할 수 있다. 이러한 컴퓨터-판독가능 매체는 위에서 소개된 사용자-액세스가능 대용량 저장장치와 연관된 매체일 수 있을 뿐만 아니라, 코어-내부 대용량 저장장치(1247) 또는 ROM(1245)과 같은 비-일시적 특성을 갖는 코어(1240)의 소정 저장장치일 수 있다. 본 개시내용의 다양한 실시예들을 구현하는 소프트웨어는 이러한 디바이스들에 저장되어 코어(1240)에 의해 실행될 수 있다. 컴퓨터-판독가능 매체는 특정 요구에 따라 하나 이상의 메모리 디바이스들 또는 칩들을 포함할 수 있다. 소프트웨어는 코어(1240) 및 특히 내부의 프로세서들(CPU, GPU, FPGA 등 포함)이 RAM(1246)에 저장된 데이터 구조들의 정의 및 소프트웨어에 의해 정의된 프로세스에 따라 이러한 데이터 구조들을 수정하는 것을 포함하여, 본원에 설명된 특정 프로세스들 또는 특정 프로세스들의 특정 부분들을 실행하게 할 수 있다. 추가로 또는 대안으로, 컴퓨터 시스템은 본원에 설명된 특정 프로세스들 또는 특정 프로세스들의 특정 부분들을 실행하기 위해 소프트웨어 대신에 또는 함께 동작할 수 있는 회로(예를 들어: 가속기(1244))에 하드웨어 되거나 달리 내장된 로직의 결과로서 기능을 제공할 수 있다. 소프트웨어에 대한 참조는 로직을 포함할 수 있고, 해당되는 경우 그 반대의 경우도 마찬가지이다. 컴퓨터-판독가능 매체에 대한 언급들은 적절한 경우 실행을 위한 소프트웨어를 저장하는 회로(이를테면 집적 회로(IC)), 실행을 위한 로직을 구현하는 회로, 또는 둘 모두를 포함할 수 있다. 본 개시내용은 하드웨어 및 소프트웨어의 임의의 적합한 조합을 포함한다.

[0154]	부록 A: 약어들
[0155]	JEM: JEM(joint exploration model)
[0156]	VVC: 다용도 비디오 코딩
[0157]	BMS: 벤치마크 세트
[0158]	MV: 모션 벡터
[0159]	HEVC: HEVC(High Efficiency Video Coding)
[0160]	SEI: SEI(Supplementary Enhancement Information)
[0161]	VUI: 비디오 사용 정보
[0162]	GOPs: GOP(Groups of Picture)들
[0163]	TUs: 변환 유닛들
[0164]	PUs: 예측 유닛들
[0165]	CTUs: 코딩 트리 유닛들
[0166]	CTBs: 코딩 트리 블록들
[0167]	PBs: 예측 블록들
[0168]	HRD: HRD(Hypothetical Reference Decoder)
[0169]	SNR: 신호 잡음비
[0170]	CPUs: 중앙 처리 장치들
[0171]	GPUs: 그래픽 처리 장치들
[0172]	CRT: 음극선관
[0173]	LCD: 액정 디스플레이
[0174]	OLED: 유기 발광 다이오드
[0175]	CD: 컴팩트 디스크
[0176]	DVD: 디지털 비디오 디스크
[0177]	ROM: 판독-전용 메모리
[0178]	RAM: 랜덤 액세스 메모리
[0179]	ASIC: 주문형 집적 회로
[0180]	PLD: 프로그램가능 로직 디바이스
[0181]	LAN: 근거리 통신망
[0182]	GSM: GSM(Global System for Mobile communication)들
[0183]	LTE: LTE(Long-Term Evolution)
[0184]	CANBus: CANBus(Controller Area Network Bus)
[0185]	USB: USB(Universal Serial Bus)
[0186]	PCI: PCI(Peripheral Component Interconnect)
[0187]	FPGA: FPGA(Field Programmable Gate Area)들
[0188]	SSD: 솔리드-스테이트 드라이브
[0189]	IC: 집적 회로

[0190]

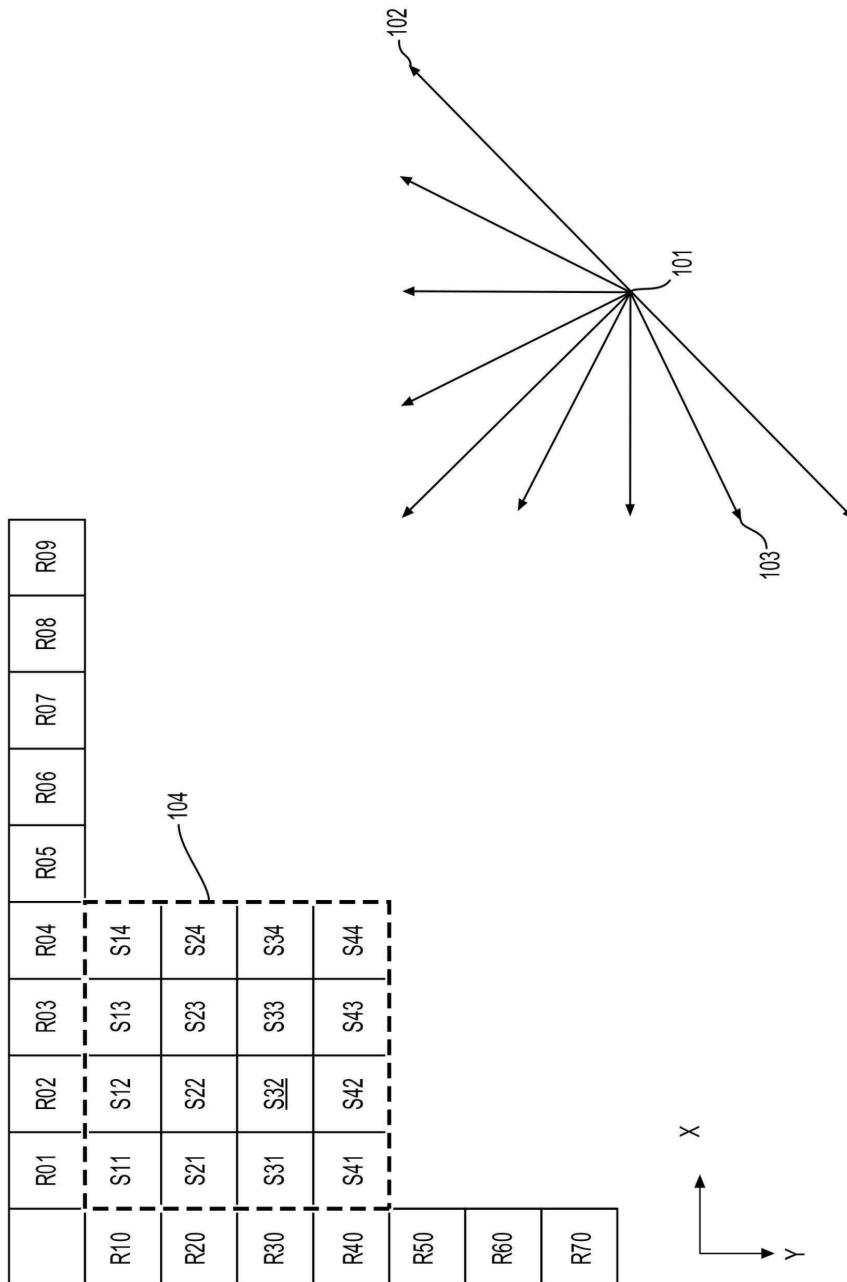
CU: 코딩 유닛

[0191]

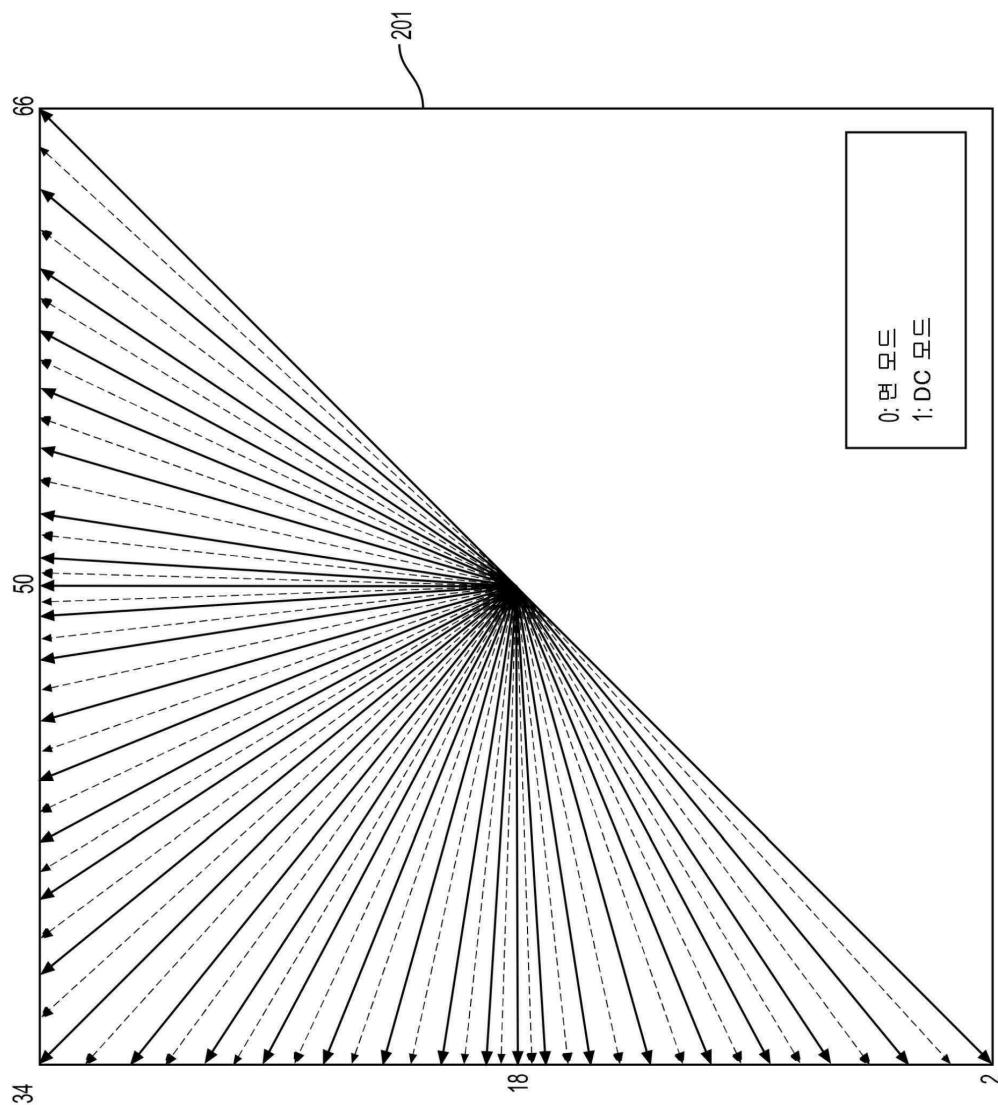
본 개시내용이 몇몇 예시적인 실시예들을 설명하였지만, 본 개시내용의 범위 내에 속하는 변경들, 순열들, 및 다양한 대체 등가물들이 있다. 따라서, 통상의 기술자들이 본원에 명시적으로 도시되거나 설명되지는 않았지만, 본 개시내용의 원리들을 구현하고 따라서 본 개시내용의 사상 및 범위 내에 있는 다수의 시스템들 및 방법들을 고안할 수 있음이 인식될 것이다.

## 도면

### 도면1

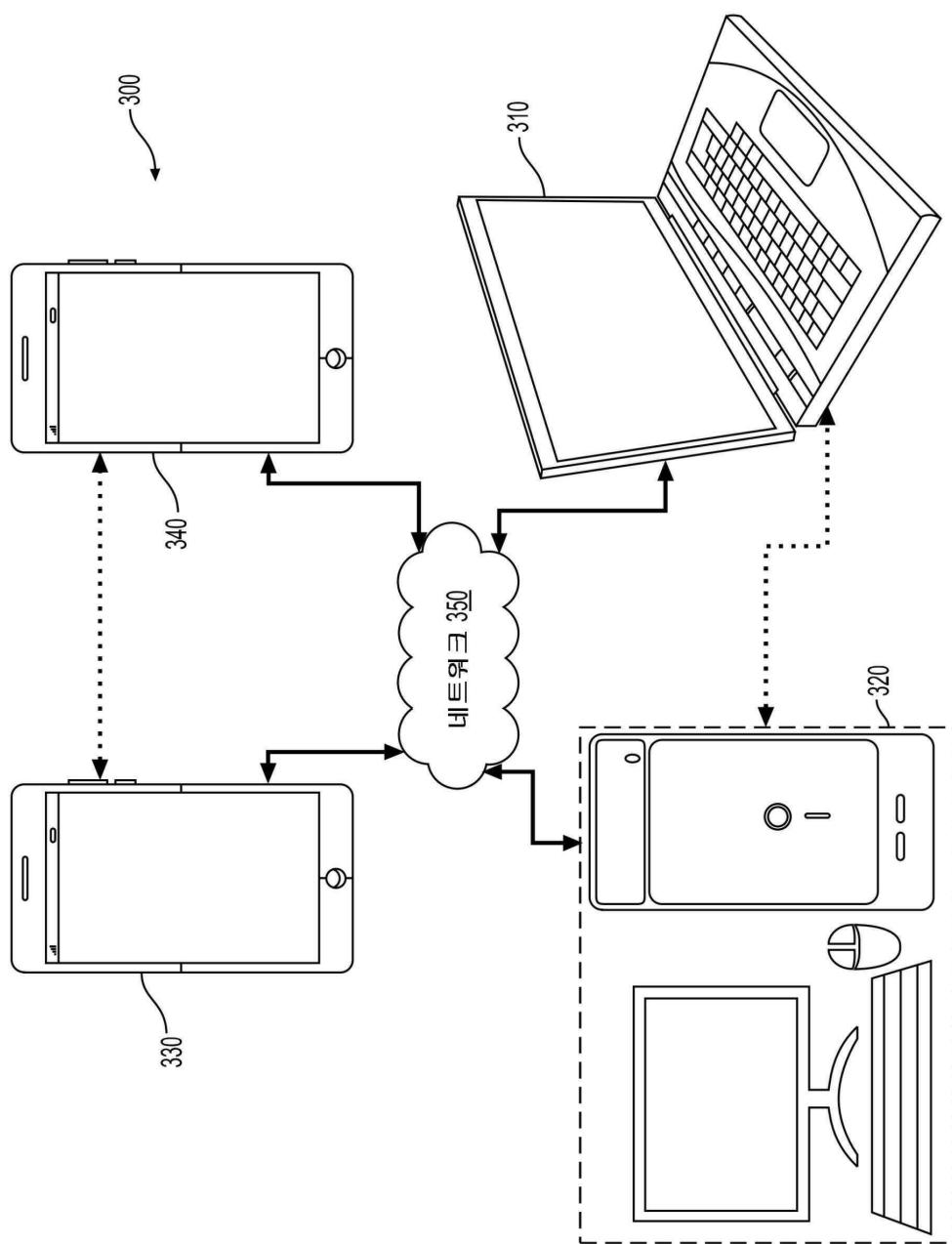


도면2

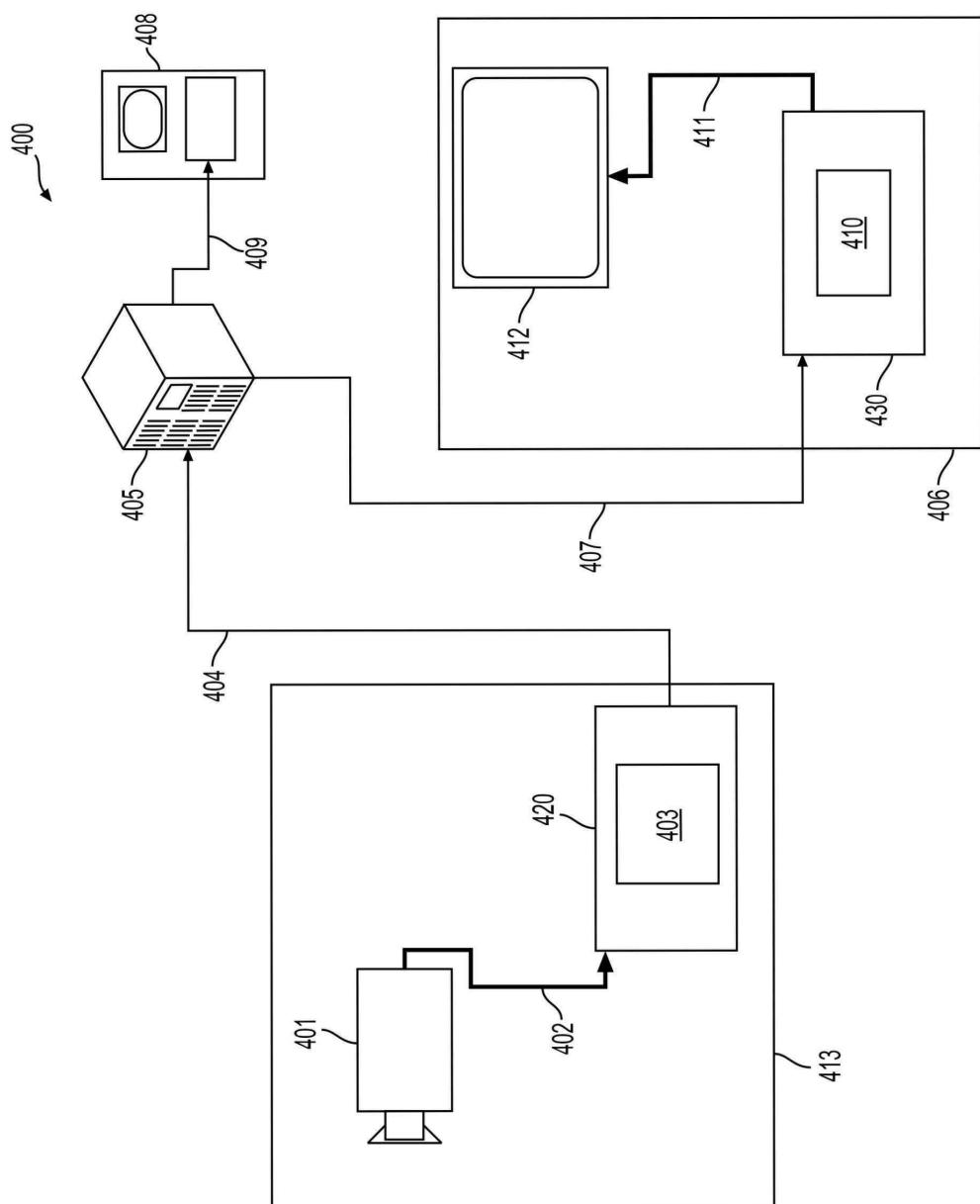


34

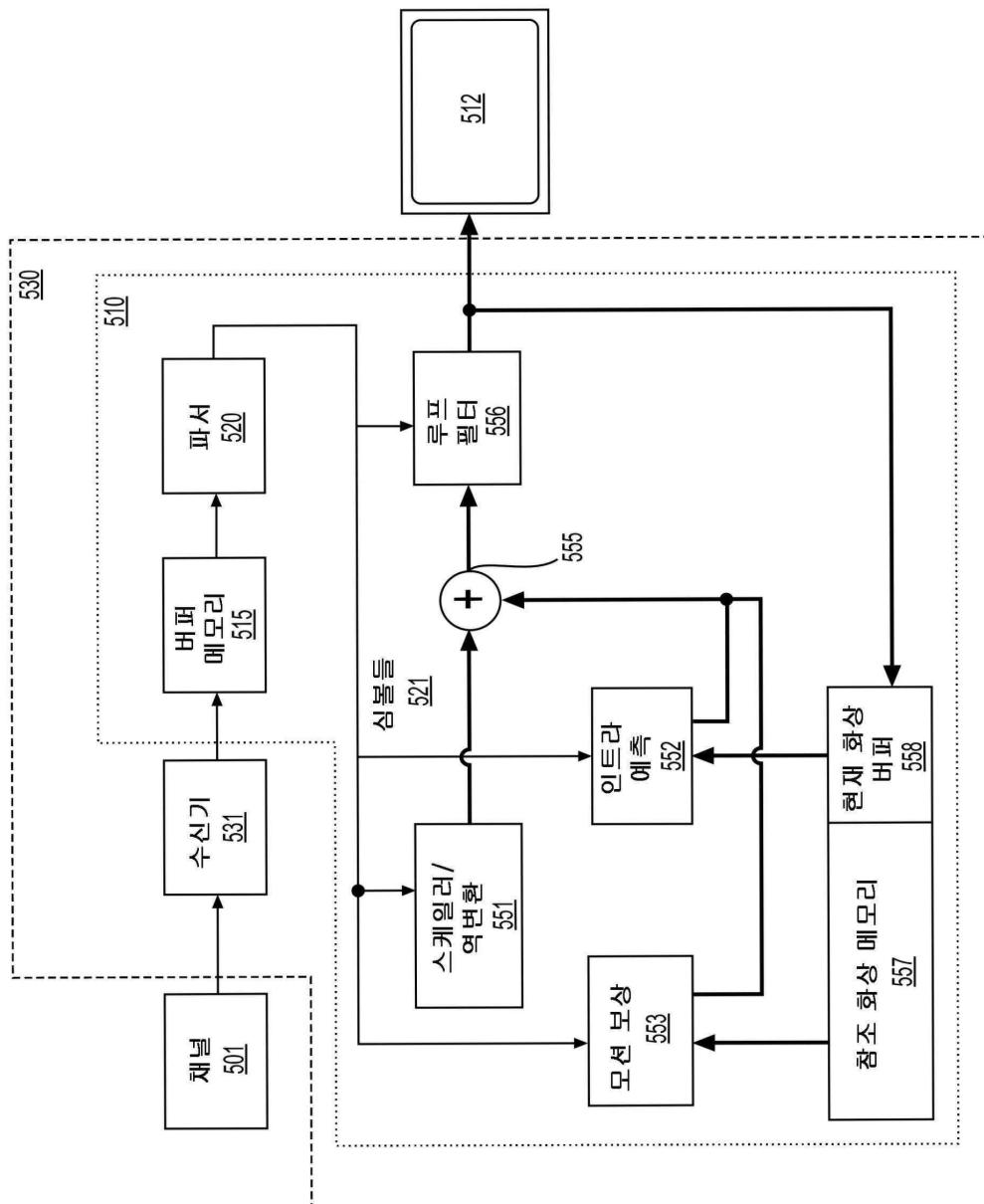
도면3



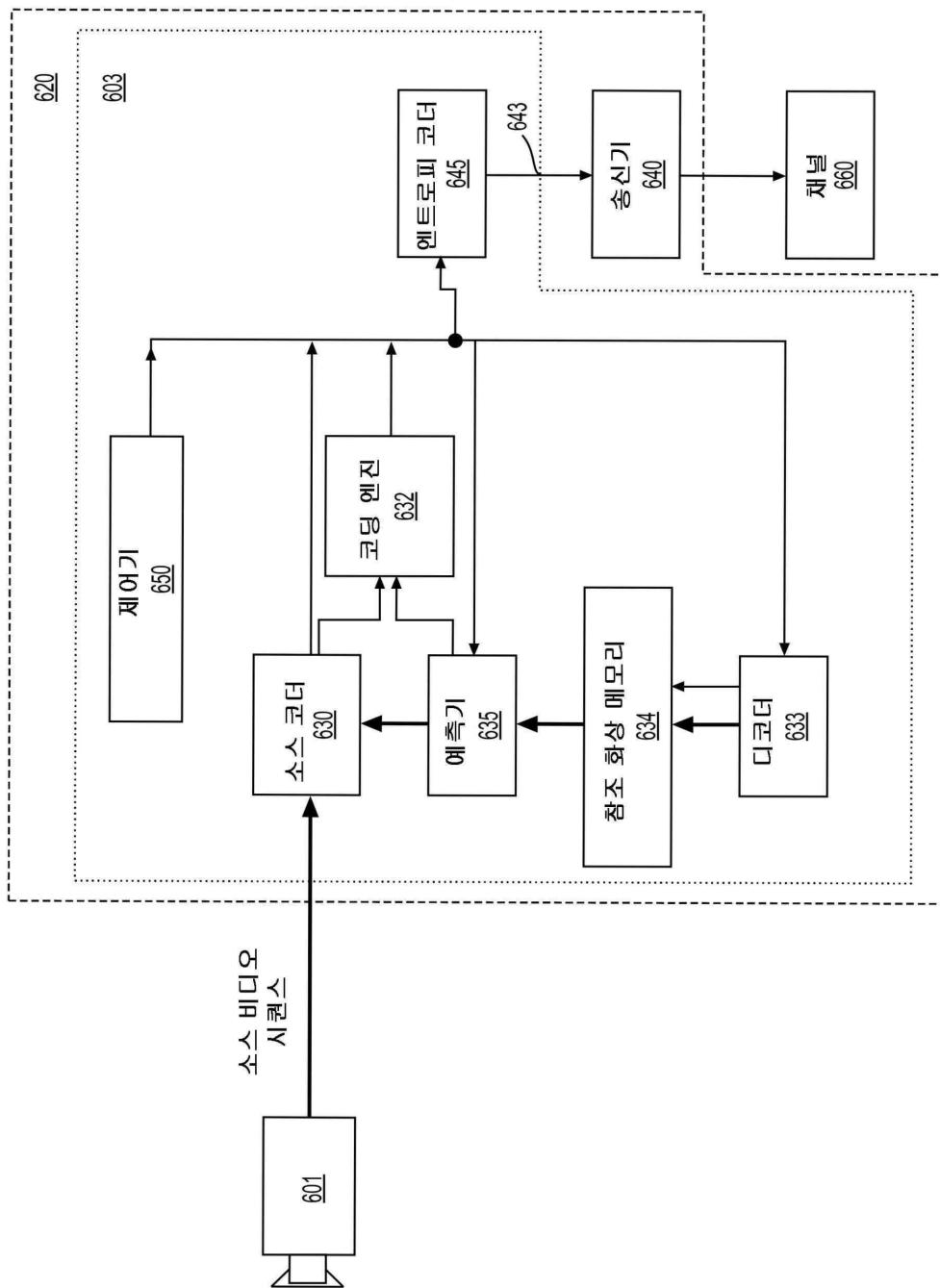
도면4



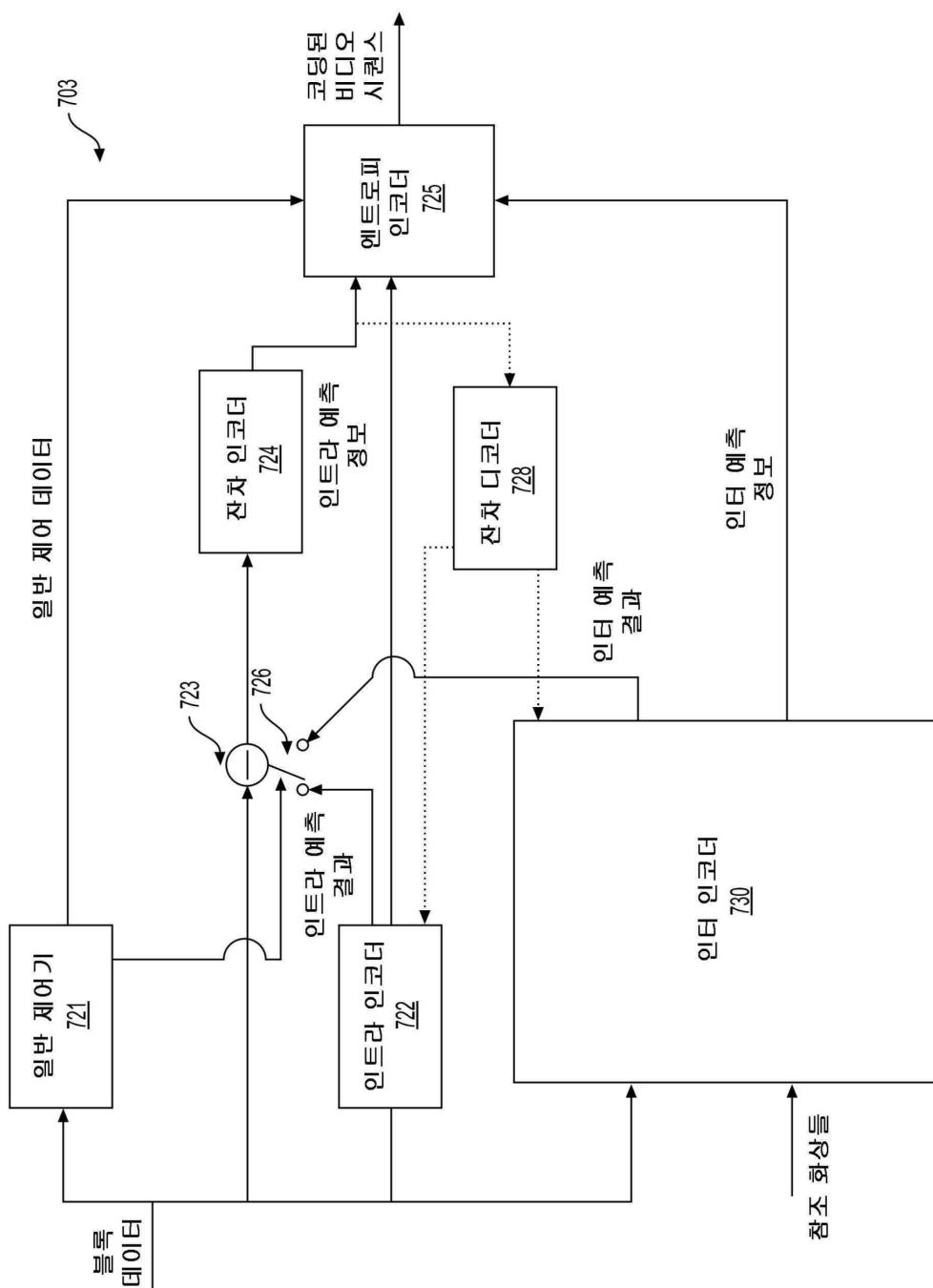
도면5



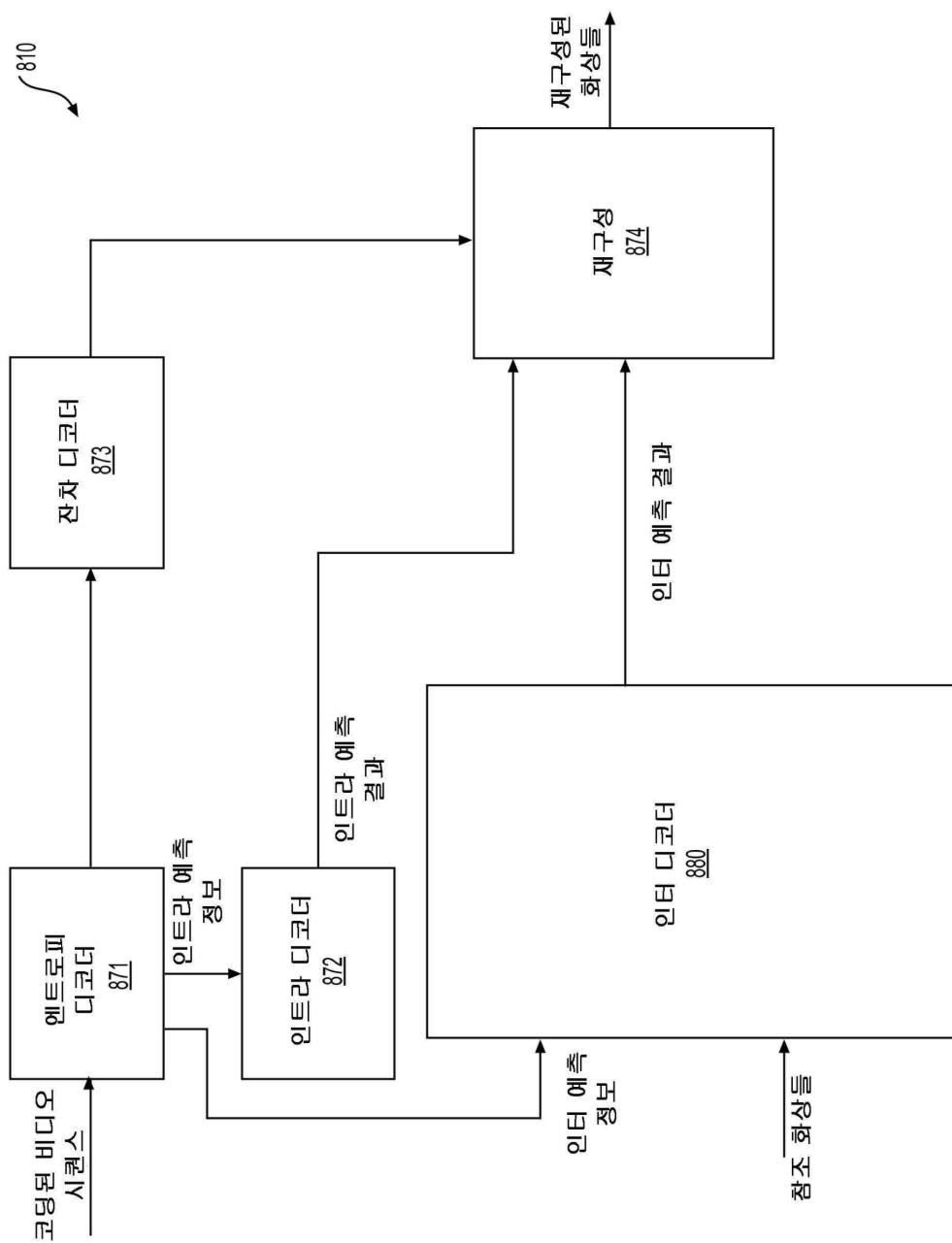
도면6



도면7



## 도면8



## 도면9

↙ 900 ↘

```

for( i = 0; i < CurrentPaletteSize[ startComp ]; i++ )

    for( compIdx = startComp; compIdx < (startComp + numComps); compIdx++ )

        newPredictorPaletteEntries[ compIdx ][ i ] = CurrentPaletteEntries[ compIdx ][ i ]

        newPredictorPaletteSize = CurrentPaletteSize[ startComp ]

        for( i = 0; i < PredictorPaletteSize[ startComp ] && newPredictorPaletteSize < 63; i++ )

            if( !PalettePredictorEntryReuseFlags[ i ] ) {

                for( compIdx = startComp; compIdx < (startComp + numComps); compIdx++ )

                    newPredictorPaletteEntries[ compIdx ][ newPredictorPaletteSize ] = PredictorPaletteEntries[ compIdx ][ i ]

                    newPredictorPaletteEntries[ compIdx ][ i ] = PredictorPaletteEntries[ compIdx ][ i ]

                    newPredictorPaletteSize++

            }

        }

        for( compIdx = startComp; compIdx < ( startComp + numComps ); compIdx++ )

            for( i = 0; i < newPredictorPaletteSize; i++ )

                PredictorPaletteEntries[ compIdx ][ i ] = newPredictorPaletteEntries[ compIdx ][ i ]

            PredictorPaletteSize[ startComp ] = newPredictorPaletteSize

```

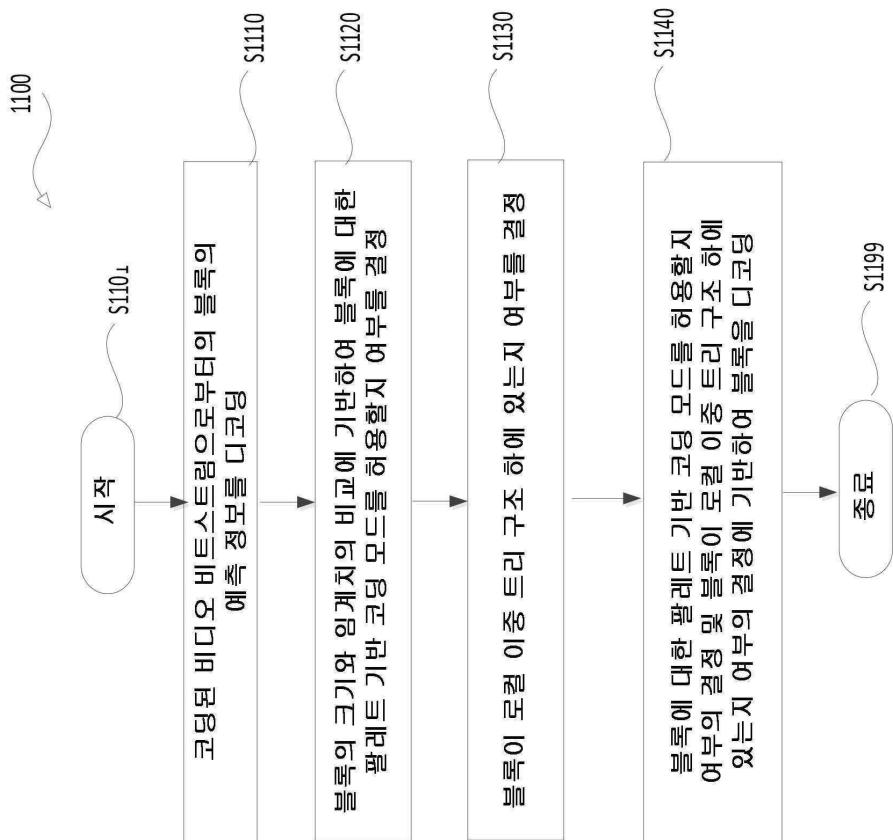
## 도면10

```

...
if( CuPredModel[ chType ][ x0 ][ y0 ] == MODE_INTRA && sps_palette_enabled_flag && cbWidth <=
64 && cbHeight <= 64 && cu_skip_flag[ x0 ][ y0 ] == 0 && modeType != MODE_TYPE_INTER &&
[((cbWidth * cbHeight) > (treeType != DUAL_TREE_CHROMA ? 16 : 16 * SubWidthC * SubHeightC)) &&
((modeType != MODE_TYPE_INTRABLOCK || treeType != DUAL_TREE_CHROMA) ||
pred_mode_plt_flag
...
...

```

## 도면11



## 도면12

