



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2022년11월07일  
(11) 등록번호 10-2464188  
(24) 등록일자 2022년11월02일

- (51) 국제특허분류(Int. Cl.)  
H04N 19/132 (2014.01) H04N 19/156 (2014.01)  
H04N 19/186 (2014.01) H04N 19/42 (2014.01)  
H04N 19/50 (2014.01)
- (52) CPC특허분류  
H04N 19/132 (2015.01)  
H04N 19/156 (2015.01)
- (21) 출원번호 10-2022-7020451(분할)
- (22) 출원일자(국제) 2019년02월21일  
심사청구일자 2022년06월16일
- (85) 번역문제출일자 2022년06월16일
- (65) 공개번호 10-2022-0085851
- (43) 공개일자 2022년06월22일
- (62) 원출원 특허 10-2022-7002463  
원출원일자(국제) 2019년02월21일  
심사청구일자 2022년01월24일
- (86) 국제출원번호 PCT/EP2019/054377
- (87) 국제공개번호 WO 2019/162414  
국제공개일자 2019년08월29일
- (30) 우선권주장  
1802972.8 2018년02월23일 영국(GB)  
1820023.8 2018년12월07일 영국(GB)
- (56) 선행기술조사문헌  
Kai Zhang, et al. Enhanced Cross-component Linear Model Intra-prediction, Joint Video exploration Team(JVET), JVET-D0110, 2016-10-16, pp. 1-6

- (73) 특허권자  
캐논 가부시끼가이샤  
일본 도쿄도 오오마루 시모마루쵸 3쵸메 30방 2고
- (72) 발명자  
은노 파트리스  
프랑스 35700 렌느 5 뒤편 메슬레  
지스케 크리스토프  
프랑스 35690 아씨뉴 뒤편 드 마르탱 페쉴르 43  
(뒷면에 계속)
- (74) 대리인  
장수길, 이중희

전체 청구항 수 : 총 8 항

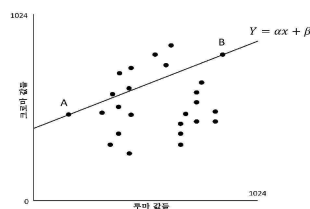
심사관 : 전용욱

(54) 발명의 명칭 인코딩 및 디코딩 방법, 인코딩 및 디코딩 디바이스, 및 컴퓨터 판독가능 매체

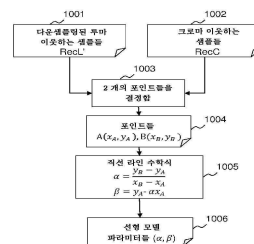
(57) 요약

개시내용은 동일한 프레임에서의 제2-컴포넌트 블록의 연관된 재구성된 제2-컴포넌트 샘플로부터 제1-컴포넌트 블록에 대한 제1-컴포넌트 샘플을 획득하기 위한 선형 모델의 유도를 위한 교차-컴포넌트 예측 및 방법들에 관한 것으로, 방법은 제1-컴포넌트 및 제2-컴포넌트 둘 모두의 재구성된 샘플들에 기초하여, 2개의 포인트들 - 각각의 (뒷면에 계속)

대표도



도 9



도 10

포인트는 2개의 변수들에 의해 정의되고, 제1 변수는 제2-컴포넌트 샘플 값에 대응하고, 제2 변수는 제1-컴포넌트 샘플 값에 대응함 - 을 통과하는 직선 라인을 표현하는 선형 수학적의 파라미터들을 결정하는 단계; 및 직선 라인 파라미터들에 의해 정의된 선형 모델을 유도하는 단계를 포함하고; 상기 파라미터들을 결정하는 단계는 정수 산술을 이용한다.

(52) CPC특허분류

*H04N 19/186* (2015.01)

*H04N 19/42* (2015.01)

*H04N 19/50* (2015.01)

(72) 발명자

**라로슈 기욤**

프랑스 35250 생 토뱅 도비네 라 바스 로므레 5

**타케 조나탕**

프랑스 35160 탈랑삭 라 프로아르디에르

## 명세서

### 청구범위

#### 청구항 1

타겟 에어리어의 크로마 샘플을 상기 타겟 에어리어의 연관된 루마 샘플로부터 획득하기 위한 선형 모델의 파라미터들을 유도함으로써 이미지들을 인코딩하는 방법으로서,

상기 선형 모델의 파라미터들을 결정하기 위한 값들의 2개의 쌍을 결정하는 단계로서, 상기 2개의 쌍의 각각은 2개의 변수에 의해 정의되고, 상기 2개의 변수 중 제1 변수는 루마 샘플에 대응하고, 상기 2개의 변수 중 제2 변수는 크로마 샘플에 대응하는, 단계;

상기 값들의 2개의 쌍을 이용함으로써 상기 선형 모델의 기울기에 대응하는 파라미터를 포함하는 상기 선형 모델의 파라미터들을 결정하는 단계로서, 상기 기울기에 대응하는 파라미터의 크기는, 상기 기울기를 정수 산술로 나타내는 비트의 수가 5비트를 초과하지 않도록 제한되는, 단계; 및

상기 선형 모델의 기울기에 대응하는 상기 결정된 파라미터 및 상기 2개의 쌍 중 하나를 사용함으로써 상기 선형 모델의 절편에 대응하는 파라미터를 결정하는 단계를 포함하는, 방법.

#### 청구항 2

제1항에 있어서, 상기 선형 모델의 기울기에 대응하는 파라미터의 결정은 제산 결정을 포함하고, 상기 결정은 상기 제산에 비트-시프트를 적용함으로써 상기 제산의 크기를 감소시키는 것을 포함하는, 방법.

#### 청구항 3

타겟 에어리어의 크로마 샘플을 상기 타겟 에어리어의 연관된 루마 샘플로부터 획득하기 위한 선형 모델의 파라미터들을 유도함으로써 이미지들을 디코딩하는 방법으로서,

상기 선형 모델의 파라미터들을 결정하기 위한 값들의 2개의 쌍을 결정하는 단계로서, 상기 2개의 쌍의 각각은 2개의 변수에 의해 정의되고, 상기 2개의 변수 중 제1 변수는 루마 샘플에 대응하고, 상기 2개의 변수 중 제2 변수는 크로마 샘플에 대응하는, 단계;

상기 값들의 2개의 쌍을 이용함으로써 상기 선형 모델의 기울기에 대응하는 파라미터를 포함하는 상기 선형 모델의 파라미터들을 결정하는 단계로서, 상기 기울기에 대응하는 파라미터의 크기는, 상기 기울기를 정수 산술로 나타내는 비트의 수가 5비트를 초과하지 않도록 제한되는, 단계; 및

상기 선형 모델의 기울기에 대응하는 상기 결정된 파라미터 및 상기 2개의 쌍 중 하나를 사용함으로써 상기 선형 모델의 절편에 대응하는 파라미터를 결정하는 단계를 포함하는, 방법.

#### 청구항 4

제3항에 있어서, 상기 선형 모델의 기울기에 대응하는 파라미터의 결정은 제산 결정을 포함하고, 상기 결정은 상기 제산에 비트-시프트를 적용함으로써 상기 제산의 크기를 감소시키는 것을 포함하는, 방법.

#### 청구항 5

타겟 에어리어의 크로마 샘플을 상기 타겟 에어리어의 연관된 루마 샘플로부터 획득하기 위한 선형 모델의 파라미터들을 유도함으로써 이미지들을 인코딩하기 위한 디바이스로서,

상기 선형 모델의 파라미터들을 유도하기 위한 유도 유닛;

상기 선형 모델의 파라미터들을 결정하기 위한 값들의 2개의 쌍을 결정하기 위한 제1 결정 유닛으로서, 상기 2개의 쌍의 각각은 2개의 변수에 의해 정의되고, 상기 2개의 변수 중 제1 변수는 루마 샘플에 대응하고, 상기 2개의 변수 중 제2 변수는 크로마 샘플에 대응하는, 제1 결정 유닛; 및

상기 값들의 2개의 쌍을 이용함으로써 상기 선형 모델의 기울기에 대응하는 파라미터를 포함하는 상기 선형 모델의 파라미터들을 결정하기 위한 제2 결정 유닛을 포함하고,

상기 기울기에 대응하는 파라미터의 크기는, 상기 기울기를 정수 산술로 나타내는 비트의 수가 5비트를 초과하지 않도록 제한되고,

상기 제2 결정 유닛은, 상기 선형 모델의 기울기에 대응하는 상기 결정된 파라미터 및 상기 2개의 쌍 중 하나를 사용함으로써 상기 선형 모델의 절편에 대응하는 파라미터를 결정하도록 구성된, 디바이스.

**청구항 6**

타겟 에어리어의 크로마 샘플을 상기 타겟 에어리어의 연관된 루마 샘플로부터 획득하기 위한 선형 모델의 파라미터들을 유도함으로써 이미지들을 디코딩하기 위한 디바이스로서,

상기 선형 모델의 파라미터들을 유도하기 위한 유도 유닛;

상기 선형 모델의 파라미터들을 결정하기 위한 값들의 2개의 쌍을 결정하기 위한 제1 결정 유닛으로서, 상기 2개의 쌍의 각각은 2개의 변수에 의해 정의되고, 상기 2개의 변수 중 제1 변수는 루마 샘플에 대응하고, 상기 2개의 변수 중 제2 변수는 크로마 샘플에 대응하는, 제1 결정 유닛; 및

상기 값들의 2개의 쌍을 이용함으로써 상기 선형 모델의 기울기에 대응하는 파라미터를 포함하는 상기 선형 모델의 파라미터들을 결정하기 위한 제2 결정 유닛을 포함하고,

상기 기울기에 대응하는 파라미터의 크기는, 상기 기울기를 정수 산술로 나타내는 비트의 수가 5비트를 초과하지 않도록 제한되고,

상기 제2 결정 유닛은, 상기 선형 모델의 기울기에 대응하는 상기 결정된 파라미터 및 상기 2개의 쌍 중 하나를 사용함으로써 상기 선형 모델의 절편에 대응하는 파라미터를 결정하도록 구성된, 디바이스.

**청구항 7**

프로그래밍가능 장치를 위한 실행가능 코드를 포함하는 비일시적 컴퓨터 판독가능 매체로서,

상기 실행가능 코드는, 타겟 에어리어의 크로마 샘플을 상기 타겟 에어리어의 연관된 루마 샘플로부터 획득하기 위한 선형 모델의 파라미터들을 유도함으로써 이미지들을 인코딩하는 방법을 구현하기 위한 명령어들의 시퀀스를 포함하고,

상기 방법은,

상기 선형 모델의 파라미터들을 결정하기 위한 값들의 2개의 쌍을 결정하는 단계로서, 상기 2개의 쌍의 각각은 2개의 변수에 의해 정의되고, 상기 2개의 변수 중 제1 변수는 루마 샘플에 대응하고, 상기 2개의 변수 중 제2 변수는 크로마 샘플에 대응하는, 단계;

상기 값들의 2개의 쌍을 이용함으로써 상기 선형 모델의 기울기에 대응하는 파라미터를 포함하는 상기 선형 모델의 파라미터들을 결정하는 단계로서, 상기 기울기에 대응하는 파라미터의 크기는, 상기 기울기를 정수 산술로 나타내는 비트의 수가 5비트를 초과하지 않도록 제한되는, 단계; 및

상기 선형 모델의 기울기에 대응하는 상기 결정된 파라미터 및 상기 2개의 쌍 중 하나를 사용함으로써 상기 선형 모델의 절편에 대응하는 파라미터를 결정하는 단계를 포함하는, 비일시적 컴퓨터 판독가능 매체.

**청구항 8**

프로그래밍가능 장치를 위한 실행가능 코드를 포함하는 비일시적 컴퓨터 판독가능 매체로서,

상기 실행가능 코드는, 타겟 에어리어의 크로마 샘플을 상기 타겟 에어리어의 연관된 루마 샘플로부터 획득하기 위한 선형 모델의 파라미터들을 유도함으로써 이미지들을 디코딩하는 방법을 구현하기 위한 명령어들의 시퀀스를 포함하고,

상기 방법은,

상기 선형 모델의 파라미터들을 결정하기 위한 값들의 2개의 쌍을 결정하는 단계로서, 상기 2개의 쌍의 각각은 2개의 변수에 의해 정의되고, 상기 2개의 변수 중 제1 변수는 루마 샘플에 대응하고, 상기 2개의 변수 중 제2 변수는 크로마 샘플에 대응하는, 단계;

상기 값들의 2개의 쌍을 이용함으로써 상기 선형 모델의 기울기에 대응하는 파라미터를 포함하는 상기 선형 모

델의 파라미터들을 결정하는 단계로서, 상기 기울기에 대응하는 파라미터의 크기는, 상기 기울기를 정수 산술로 나타내는 비트의 수가 5비트를 초과하지 않도록 제한되는, 단계; 및

상기 선형 모델의 기울기에 대응하는 상기 결정된 파라미터 및 상기 2개의 쌍 중 하나를 사용함으로써 상기 선형 모델의 절편에 대응하는 파라미터를 결정하는 단계를 포함하는, 비밀시적 컴퓨터 판독가능 매체.

**발명의 설명**

**기술 분야**

[0001] 본 발명은 주어진 비디오 컴포넌트(video component)의 블록들의 인코딩 또는 디코딩에 관한 것으로, 특히, 이러한 컴포넌트 블록들의 인트라 예측(intra prediction) 또는 이러한 블록들의 샘플들을 획득하는 것에 관한 것이다. 발명은 또 다른 컴포넌트의 샘플들, 전형적으로, 루마 샘플(luma sample)들로부터 비디오 데이터의 컴포넌트의 블록들, 전형적으로, 크로마 컴포넌트(chroma component)의 블록들을 획득할 시에 애플리케이션들을 발 견한다.

**배경 기술**

[0002] 비디오 데이터의 예측 인코딩은 픽셀(pixel)들의 블록들로의 프레임(frame)들의 분할에 기초한다. 픽셀들의 각 각의 블록에 대하여, 예측자 블록(predictor block)은 이용가능한 데이터에서 검색된다. 예측자 블록은 INTER 코딩 모드들에서 현재의 것과는 상이한 참조 프레임(reference frame)에서의 블록일 수 있거나, INTRA 코딩 모 드들에서 현재의 프레임에서의 이웃하는 픽셀들로부터 생성될 수 있다. 상이한 인코딩 모드들은 예측자 블록을 결정하는 상이한 방식들에 따라 정의된다. 인코딩의 결과는 예측자 블록, 및 인코딩되어야 할 블록과 예측자 블록 사이의 차인 잔차 블록(residual block)의 시그널링(signalling)이다.

[0003] INTRA 코딩 모드들에 관하여, 직류(Direct Current)(DC) 모드, 평면 모드, 및 각도 모드들과 같은 다양한 모 드들이 통상 제안된다. 이들의 각각은 공간적으로 이웃하는 블록들로부터의 이전에 디코딩된 경계 샘플(boundary sample)들을 이용하여 블록의 샘플들을 예측하는 것을 추구한다.

[0004] 인코딩은 비디오 데이터의 픽셀들을 형성하는 각각의 컴포넌트에 대하여 수행될 수 있다. (적색-녹색-청색에 대한) RGB 표현이 널리-알려져 있지만, YUV 표현은 바람직하게는 인터-채널 중복성(inter-channel redundanc y)을 감소시키기 위한 인코딩을 위하여 이용된다. 이 인코딩 모드들에 따르면, 픽셀들의 블록은 몇몇, 전형적 으로 3개의 컴포넌트 블록들로 구성된 것으로서 고려될 수 있다. RGB 픽셀 블록은 블록의 픽셀들의 R 컴포넌트 의 값들을 포함하는 R 컴포넌트 블록, 이 픽셀들의 G 컴포넌트의 값들을 포함하는 G 컴포넌트 블록, 이 픽셀들 의 B 컴포넌트의 값들을 포함하는 B 컴포넌트 블록으로 구성된다. 유사하게도, YUV 픽셀 블록은 Y 컴포넌트 블 록(루마), U 컴포넌트 블록(크로마), 및 V 컴포넌트 블록(또한 크로마)으로 구성된다. 또 다른 예는 YCbCr, 여 기서, Cb 및 Cr은 크로마 컴포넌트들로서 또한 알려져 있다. 그러나, 인터-컴포넌트(또한, 교차-컴포넌트 (cross-component)로서 알려짐) 상관이 여전히 국소적으로 관찰된다.

[0005] 압축 효율을 개선시키기 위하여, 교차-컴포넌트 예측(Cross-Component Prediction)(CCP)의 사용이 이 최신 기술 에서 연구되었다. CCP의 주요 애플리케이션은 루마-대-크로마(luma-to-chroma) 예측에 관한 것이다. 그것은 루마 샘플들이 이미 인코딩되었고 (디코더가 행하는 바와 같이) 인코딩된 데이터로부터 재구성되었다는 것과, 크로마는 루마로부터 예측된다는 것을 의미한다. 그러나, 변형들은 크로마-대-크로마(chroma-to-chroma) 예측 을 위하여, 또는 더 일반적으로, (RGB를 포함하는) 제1-컴포넌트 대 제2-컴포넌트 예측을 위하여 CCP를 이용할 다.

[0006] 교차-컴포넌트 예측은 크로마 픽셀들의 블록에 직접적으로 적용될 수 있거나, 잔차 크로마 블록(residual chroma block)(크로마 블록과 크로마 블록 예측자 사이의 차를 의미함)에 적용될 수 있다.

[0007] 선형 모델(Linear Model)(LM) 모드는 결정되어야 할 하나 또는 두 개의 파라미터들, 기울기( $\alpha$ ), 및 오프셋 ( $\beta$ )에 의존하면서, 크로마 인트라 예측 모드(chroma intra prediction mode)로서 루마로부터 크로마를 예측하 기 위하여 선형 모델을 이용한다. 크로마 인트라 예측자(chroma intra predictor)는 이에 따라, 파라미터들을 갖는 선형 모델을 이용하여 현재의 루마 블록의 재구성된 루마 샘플들로부터 유도된다.

[0008] 선형성, 즉, 파라미터들  $\alpha$  및  $\beta$ 는 재구성된 인과적 샘플(causal sample)들로부터, 특히, 예측하기 위한 현재

의 크로마 블록과 이웃하는 재구성된 크로마 샘플들을 포함하는 이웃하는 크로마 샘플 세트로부터, 그리고 현재의 루마 블록과 이웃하는 루마 샘플들을 포함하는 이웃하는 루마 샘플 세트로부터 유도된다.

- [0009] 구체적으로, NxN 크로마 블록에 대하여, 상부 행(row)의 N 개의 이웃들 및 좌측 열(column)의 N 개의 이웃들은 유도를 위한 이웃하는 크로마 샘플 세트를 형성하기 위하여 이용된다.
- [0010] 이웃하는 루마 샘플 세트는 또한, 대응하는 루마 블록 바로 상부의 N 개의 이웃하는 샘플들 및 루마 블록의 좌측 상의 N 개의 이웃하는 샘플들로 이루어진다.
- [0011] 크로마 컴포넌트들을 서브-샘플링(sub-sampling)함으로써, 시각적 렌더링(visual rendering)의 상당한 열화 없이 인코딩하기 위하여 비디오 데이터의 크기를 감소시키는 것이 알려져 있다. 알려진 서브샘플링 모드들은 4:1:1, 4:2:2, 4:2:0로 표기된다.
- [0012] 비디오 크로마 데이터가 서브샘플링되는 상황에서는, NxN 크로마 블록에 대응하는 루마 블록이 NxN보다 더 크다. 그 경우에, 이웃하는 루마 샘플 세트는 크로마 해상도(chroma resolution)와 정합하기 위하여 다운-샘플링(down-sample)된다. 현재의 NxN 크로마 블록에서의 크로마 샘플들을 예측하기 위한 크로마 인트라 예측자는 유도된 하나 이상의 파라미터들  $\alpha$  및  $\beta$  및 크로마 해상도와 정합하기 위하여 이전에 다운-샘플링되는 현재의 루마 블록의 재구성된 루마 샘플들을 갖는 선형 모델을 이용하여 생성되어야 한다. 크로마 해상도로의 재구성된 루마 샘플들의 다운-샘플링은 루마 샘플 세트 및 크로마 인트라 예측자의 양자를 형성하기 위하여 크로마 샘플들과 동일한 수의 샘플들을 추출(retrieve)하는 것을 가능하게 한다.
- [0013] 크로마 인트라 예측자는 이에 따라, 인코더에서 인코딩되는 잔차 크로마 블록을 획득하기 위하여 현재의 크로마 블록으로부터 감산(subtract)된다. 반대로, 디코더에서는, 디코딩된 블록의 재구성으로서 또한 알려진 크로마 블록을 추출하기 위하여, 크로마 인트라 예측자가 수신된 잔차 크로마 블록에 가산된다. 이것은 샘플 범위의 외부로 가는 가산의 결과들에 대한 클리핑(clipping)을 또한 수반할 수 있다.
- [0014] 때때로, 잔차 크로마 블록은 무시가능하고, 이에 따라, 인코딩 동안에 고려되지 않는다. 그 경우에, 전술한 크로마 인트라 예측자는 크로마 블록 자체로서 이용된다. 결과적으로, 위의 LM 모드는 하나 이상의 파라미터들을 갖는 선형 모델을 이용하여 동일한 프레임에서의 또 다른 컴포넌트의 블록의 연관된(즉, 공동위치된(collocated) 또는 대응하는) 재구성된 샘플로부터 주어진 컴포넌트의 현재의 블록에 대한 샘플을 획득하는 것을 가능하게 한다. 샘플은 유도된 하나 이상의 파라미터들 및 다른 컴포넌트의 블록에서의 연관된 재구성된 샘플들을 갖는 선형 모델을 이용하여 획득된다. 필요한 경우에, 다른 컴포넌트의 블록은 현재의 컴포넌트의 블록 해상도와 정합하기 위하여 다운-샘플링된 샘플들로 이루어진다. 현재의 컴포넌트의 블록은 전형적으로 크로마 블록이고 다른 컴포넌트의 블록은 루마 블록이지만, 이것은 그렇지 않을 수 있다. 명확함 및 단순함을 위하여, 여기에서 주어진 예들은 루마 블록으로부터의 크로마 블록의 예측에 초점을 맞추고, 설명된 메커니즘은 또 다른 컴포넌트로부터의 임의의 컴포넌트 예측에 적용될 수 있다는 것이 명확해야 한다.
- [0015] 공동 비디오 탐구 팀(Joint Video Exploration Team)(JVET)의 공동 탐구 모델(Joint Exploration Model)(JEM)은 6개의 교차-컴포넌트(루마-대-크로마) 선형 모델 모드들을 이미 알려진 기존의 인트라 예측 모드들에 추가한다. 모든 이러한 모드들은 크로마 블록들을 예측하거나 생성하기 위하여 서로 경합하고, 선택은 통상적으로 인코더 단에서의 레이트-왜곡 기준(rate-distortion criterion)에 기초하여 행해진다.
- [0016] 6개의 교차-컴포넌트(루마-대-크로마) 선형 모델 모드들은 재구성된 루마 샘플들을 다운-샘플링하기 위하여 이용된 상이한 다운-샘플링 방식들에 의해, 및/또는 파라미터들  $\alpha$  및  $\beta$  가 샘플들로부터 유도되는 이 샘플들의 상이한 샘플 세트들에 의해 서로 상이하다.
- [0017] 예를 들어, 샘플 세트들은 현재의 루마 또는 크로마 블록과 이웃하는 샘플들의 2개의 라인들(즉, 행들 및 열들)로 이루어질 수 있고, 이 라인들은 크로마 해상도에서의 현재의 루마 또는 크로마 블록의 상부 및/또는 좌측 경계들의 각각의 하나에 평행하고 바로 인접한다. 이러한 예시적인 샘플 세트는 공보 US 9,736,487에서 설명된다.
- [0018] 다른 예시적인 샘플 세트들은 공보들 US 9,288,500 및 US 9,462,273에서 또한 개시된다.
- [0019] JEM에서 이용된 다운-샘플링 방식들은 6개의 재구성된 루마 샘플들로부터 다운-샘플링된 재구성된 루마 샘플을 결정하는 6-탭(tap) 필터 뿐만 아니라, 6개의 재구성된 루마 샘플들 중으로부터의 상부 우측 및 하부 우측 샘플들, 또는 하부 및 하부 우측 샘플들, 또는 상부 및 상부 우측 샘플들 중의 어느 하나를 선택하는 3개의 2-탭 필터

터들, 및 6개의 재구성된 루마 샘플들의 상부, 상부 우측, 하부, 및 하부 우측 샘플들을 선택하는 4-탭 필터를 포함한다.

**발명의 내용**

- [0020] JEM은 프로세싱의 측면에서 복잡하다. 예를 들어, JEM은 크로마 예측자 블록 샘플들의 연산을 위한 선형 모델 파라미터들의 복잡한 유도를 요구한다.
- [0021] 본 발명은 상기한 관심사들 중의 하나 이상을 다루도록 고안되었다. 본 발명은 아마도 크로마 인트라 예측을 통해, 현재의 크로마 블록에 대한 크로마 샘플을 획득하기 위한 개선된 방법에 관한 것이다.
- [0022] 본 발명의 제1 양태에 따르면, 제1항에 따른 방법이 제공된다.
- [0023] 본 발명의 또 다른 양태에 따르면, 제29항에 따른 이미지들을 인코딩하기 위한 디바이스가 제공된다.
- [0024] 본 발명의 또 다른 양태에 따르면, 제30항에 따른 이미지들을 디코딩하기 위한 디바이스가 제공된다.
- [0025] 본 발명의 또 다른 양태에 따르면, 제31항 내지 제33항에 따른 컴퓨터 프로그램 제품, 컴퓨터-판독가능 매체, 또는 컴퓨터 프로그램이 제공된다.
- [0026] 발명의 추가의 양태들은 종속항들에서 제공된다.
- [0027] 추가의 양태에 따르면, 연관된 재구성된 제2-컴포넌트 샘플 값으로부터 제1-컴포넌트 샘플 값을 획득하기 위한 선형 모델을 유도하는 방법이 제공되고, 방법은: 2개 이상의 세트들로부터 2개의 세트들을 취하는 단계 - 각각의 세트는 제1-컴포넌트 및 제2-컴포넌트의 재구성된 샘플 값들로부터의 제1-컴포넌트 샘플 값 및 제2-컴포넌트 샘플 값을 포함함 -; 및 유도된 선형 모델을 이용하여, 2개의 세트들의 제1-컴포넌트 샘플 값들이 각각의 세트들의 제2-컴포넌트 샘플 값들로부터 획득가능하도록, 2개의 세트들 사이의 제1-컴포넌트 샘플 값들 및 제2-컴포넌트 샘플 값들에서의 변화들의 비율에 기초하여 선형 모델을 유도하는 단계를 포함한다.
- [0028] 제1-컴포넌트 샘플 값 및 연관된 재구성된 제2-컴포넌트 샘플 값은 사전설정된 관계를 통해 서로 연관된다는 것이 이해된다.
- [0029] 적당하게도, 사전설정된 관계는 이들이 서로 공동위치되거나 서로에 대응한다는 것이다. 이 공동위치된 또는 대응 관계는 각각의 샘플 값에 대하여 개별적으로, 또는 제1-컴포넌트 샘플 값들의 블록/그룹과 제2-컴포넌트 샘플 값들의 블록/그룹 사이에서 정의될 수 있다.
- [0030] 적당하게도, 사전설정된 관계는 이들이 프로세싱되어야 할 픽셀들의 현재의 블록의 적어도 하나의 픽셀과 연관되고, 예를 들어, 이들이 프로세싱되어야 하는 적어도 하나의 픽셀의 공동위치된 또는 대응하는 샘플 값들이라는 것이다. 이 공동위치된 또는 대응 관계는 각각의 샘플 값에 대하여 개별적으로, 또는 샘플 값들의 블록/그룹 내지 픽셀들의 블록/그룹 사이에서 정의될 수 있다.
- [0031] 블록들 사이, 또는 픽셀들의 현재의 블록의 적어도 하나의 픽셀과의 사전설정된 관계가 다운-샘플링/업-샘플링 후에 확립될 수 있도록, 다운-샘플링 또는 업-샘플링(up-sampling) 프로세스는 제1-컴포넌트 샘플 값들 또는 제2-컴포넌트 샘플 값들의 블록에 적용될 수 있다는 것이 또한 이해된다.
- [0032] 적당하게도, 제1-컴포넌트 샘플 값 및 연관된 제2-컴포넌트 샘플 값은 프로세싱되어야 하는 동일한 이미지 또는 프레임의 픽셀들의 블록들과 연관된다. 여기서, 제1-컴포넌트 샘플 값 및 제2-컴포넌트 샘플 값을 포함하는 세트는 제1-컴포넌트 샘플 값 및 제2-컴포넌트 샘플 값의 컴포넌트 샘플 값 세트인 것이 이해된다. 따라서, 세트는 그 엘리먼트(element)들로서 제1-컴포넌트 샘플 값 및 제2-컴포넌트 샘플 값을 갖는 n-튜플(tuple)이다. 적당하게도, 세트는 2-튜플이다. 대안적으로, 세트는 2 초과와 수의 엘리먼트들(n 엘리먼트들)을 가지는 n-튜플이다.
- [0033] 적당하게도, 제1-컴포넌트 및 제2-컴포넌트의 재구성된 샘플 값들은 프로세싱되어야 할 현재의 블록과 이웃하는 하나 이상의 블록들과 연관된다. 적당하게도, 현재의 블록과 이웃하는 하나 이상의 블록들은 현재의 블록의 상부 또는 좌측에 있다.
- [0034] 적당하게도, 취해진 2개의 세트들은 2개 이상의 세트들에서의 제2-컴포넌트 샘플 값들 중에서 가장 작은 제2-컴포넌트 샘플 값 및 가장 큰 제2-컴포넌트 샘플 값을 포함하는 세트들이다. 적당하게도, 취해진 2개의 세트들은 2개 이상의 세트들에서의 제1-컴포넌트 샘플 값들 중에서 가장 작은 제1-컴포넌트 샘플 값 및 가장 큰 제1-컴포

넌트 샘플 값을 포함하는 세트들이다.

- [0035] 적당하게도, 2개의 세트들을 취하는 단계는: 2개 이상의 세트들에서의 제2-컴포넌트 샘플 값들 중에서 가장 작은 제2-컴포넌트 샘플 값 및 가장 큰 제2-컴포넌트 샘플 값을 포함하는 세트들의 제1 그룹을 결정하는 단계; 2개 이상의 세트들에서의 제1-컴포넌트 샘플 값들 중에서 가장 작은 제1-컴포넌트 샘플 값 및 가장 큰 제1-컴포넌트 샘플 값을 포함하는 세트들의 제2 그룹을 결정하는 단계; 및 제1 그룹 및 제2 그룹의 세트들로부터 2개의 세트들을 선택하는 단계를 포함한다.
- [0036] 적당하게도, 제1 그룹 및 제2 그룹의 세트들로부터 2개의 세트들을 선택하는 단계는: 가장 작은 제2-컴포넌트 샘플 값과 가장 큰 제2-컴포넌트 샘플 값 사이의 차가 가장 작은 제1-컴포넌트 샘플 값과 가장 큰 제1-컴포넌트 샘플 값 사이의 차보다 더 클 경우에 제1 그룹을 선택하는 단계; 및 그렇지 않을 경우에 제2 그룹을 선택하는 단계를 포함한다.
- [0037] 적당하게도, 제1 그룹 및 제2 그룹의 세트들로부터 2개의 세트들을 선택하는 단계는: 제1 그룹 및 제2 그룹의 세트들의 샘플 값들의 포지션(position)들을 결정하는 단계; 및 샘플 값들의 결정된 포지션들에 기초하여 2개의 세트들을 선택하는 단계를 포함한다. 적당하게도, 샘플 값들의 포지션들은 프로세싱되어야 할 제1-컴포넌트 샘플 값들의 블록과 연관되는 재구성된 제2-컴포넌트 샘플 값들의 블록과 관련하여, 재구성된 제2-컴포넌트 샘플 값들에 대하여 결정된다. 적당하게도, 샘플 값들의 포지션들은 프로세싱되어야 할 재구성된 제1-컴포넌트 샘플 값들의 블록과 관련하여, 재구성된 제1-컴포넌트 샘플 값들에 대하여 결정된다. 적당하게도, 샘플 값들의 포지션들은 프로세싱되어야 할 픽셀들의 블록과 관련하여 정의된 연관된/공동위치된/대응하는 포지션들에 기초하여 결정된다.
- [0038] 적당하게도, 샘플 값들의 결정된 포지션들에 기초하여 2개의 세트들을 선택하는 단계는 프로세싱되어야 할 블록과 이웃하는 미리 결정된 포지션에서의 샘플 값을 포함하는 세트를 선택하는 단계를 포함한다. 적당하게도, 샘플 값들의 결정된 포지션들에 기초하여 2개의 세트들을 선택하는 단계는: 제1 그룹 및 제2 그룹의 세트들 중의 임의의 것이 미리 결정된 포지션에서의 샘플 값을 포함하는지를 결정하는 단계; 및 미리 결정된 포지션에서의 샘플 값을 포함하는 세트를 2개의 세트들 중의 하나로서 선택하는 단계를 포함한다. 적당하게도, 샘플 값들의 미리 결정된 포지션들에 기초하여 2개의 세트들을 선택하는 단계는: 미리 결정된 포지션에서의 샘플 값을 포함하는 세트가 이용가능하지 않을 경우에, 제1 그룹 및 제2 그룹의 세트들 중의 임의의 것이 또 다른 미리 결정된 포지션에서의 샘플 값을 포함하는지를 결정하는 단계; 및 또 다른 미리 결정된 포지션에서의 샘플 값을 포함하는 세트를 2개의 세트들 중의 하나로서 선택하는 단계를 포함한다. 적당하게도, 미리 결정된 포지션 또는 또 다른 미리 결정된 포지션은 프로세싱되어야 할 현재의 블록과 이웃하는 포지션들 중에서 좌측 하부 또는 우측 상부 포지션이다.
- [0039] 적당하게도, 세트들의 제1 그룹 및 제2 그룹으로부터 2개의 세트들을 선택하는 단계는 세트들의 제1 그룹 및 제2 그룹으로부터의 2개의 세트들 사이의 거리들을 비교하는 단계를 포함하고, 여기서, 거리들은 제1 및 제2 컴포넌트 샘플 값들의 공간에서 정의되고, 이 공간은 2개 이상의 세트들의 각각의 세트가 상기 공간에서의 포지션에 대응하도록 세트들의 엘리먼트들에 의해 정의된다.
- [0040] 적당하게도, 2개의 세트들을 선택하는 단계는: 제1 그룹에서의 세트들 사이의 거리가 제2 그룹에서의 세트들 사이의 거리보다 더 크지를 결정하는 단계; 및 제1 그룹에서의 세트들 사이의 거리가 제2 그룹에서의 세트들 사이의 거리보다 더 클 경우에 제1 그룹을 선택하고, 그렇지 않을 경우에 제2 그룹을 선택하는 단계를 비교한다. 적당하게도, 2개의 세트들을 선택하는 단계는 세트들의 제1 그룹 및 제2 그룹으로부터 그 사이의 가장 큰 거리를 갖는 2개의 세트들을 선택하는 단계를 포함한다.
- [0041] 적당하게도, 2개의 세트들을 선택하는 단계는: 제1 그룹에서의 세트들의 대응하는 엘리먼트들이 동일한 또는 상이한 값들을 가지는지를 결정하는 단계; 및 대응하는 엘리먼트들이 동일한 값들을 가지지 않거나 상이한 값들을 가질 경우에 제1 그룹을 선택하고, 대응하는 엘리먼트들이 동일한 값들을 가지거나 상이한 값들을 가지지 않을 경우에 제2 그룹을 선택하는 단계를 포함한다. 적당하게도, 세트들의 대응하는 엘리먼트들은 제1-컴포넌트 샘플 값들 및 제2-컴포넌트 샘플 값들 중의 하나 또는 양자 중의 어느 하나이다.
- [0042] 적당하게도, 2개의 세트들을 선택하는 단계는: 제2 그룹에서의 세트들의 대응하는 엘리먼트들이 동일한 또는 상이한 값들을 가지는지를 결정하는 단계; 및 대응하는 엘리먼트들이 동일한 값들을 가지지 않거나 상이한 값들을 가질 경우에 제2 그룹을 선택하고, 대응하는 엘리먼트들이 동일한 값들을 가지거나 상이한 값들을 가지지 않을 경우에 제1 그룹을 선택하는 단계를 포함한다.

- [0043] 적당하게도, 2개의 세트들을 선택하는 단계는: 제1 그룹의 세트들 사이의 제1-컴포넌트 샘플 값들 및 제2-컴포넌트 샘플 값들에서의 변화들의 비율을 획득하는 단계; 획득된 비율이 사전설정된 값을 초과하거나, 이와 동일하거나, 그 미만인지를 결정하는 단계; 및 획득된 비율이 사전설정된 값을 초과하거나, 이와 동일하거나, 그 미만일 경우에 제1 그룹을 선택하고, 그렇지 않을 경우에 제2 그룹을 선택하는 단계를 포함한다. 적당하게도, 2개의 세트들을 선택하는 단계는: 제2 그룹의 세트들 사이의 제1-컴포넌트 샘플 값들 및 제2-컴포넌트 샘플 값들에서의 변화들의 비율을 획득하는 단계; 획득된 비율이 사전설정된 값을 초과하거나, 이와 동일하거나, 그 미만인지를 결정하는 단계; 및 획득된 비율이 사전설정된 값을 초과하거나, 이와 동일하거나, 그 미만일 경우에 제2 그룹을 선택하고, 그렇지 않을 경우에 제1 그룹을 선택하는 단계를 포함한다.
- [0044] 적당하게도, 취해진 2개의 세트들은 프로세싱되어야 할 현재의 블록과 연관되는 제2-컴포넌트 샘플 값들의 블록과 이웃하는 하나 이상의 블록들로부터의 제2-컴포넌트 샘플 값들을 포함하는 세트들이고, 2개의 세트들을 취하는 단계는 그 제2-컴포넌트 샘플 값들에 기초하여 2개의 세트들을 선택하는 단계를 포함한다. 적당하게도, 취해진 2개의 세트들은 대응하는 제2-컴포넌트 샘플 값들의 블록의 재구성된 샘플 값들 중에서 2개의 가장 빈번하게 발생하는 제2-컴포넌트 샘플 값들을 포함하는 세트들이다.
- [0045] 적당하게도, 제2-컴포넌트의 재구성된 샘플 값들은 적어도 2개의 그룹들로 분할되고, 각각의 그룹에 대하여, 2개의 세트들이 취해지고 선형 모델은 취해진 2개의 세트들에 기초하여 유도된다. 적당하게도, 그룹에 대하여 취해진 2개의 세트들이 사전설정된 값 이하인 세트들 사이의 제1-컴포넌트 샘플 값들 및 제2-컴포넌트 샘플 값들에서의 변화들의 비율을 가질 경우에, 그 그룹에 대한 선형 모델은 또 다른 그룹에 대해 취해진 2개의 세트들에 기초하여 유도된다. 적당하게도, 그룹에 대하여 취해진 2개의 세트들이 사전설정된 값 이하인 세트들 사이의 제1-컴포넌트 샘플 값들 및 제2-컴포넌트 샘플 값들에서의 변화들의 비율을 가질 경우에, 그 그룹에 대한 선형 모델은 제2-컴포넌트의 모든 재구성된 샘플 값들이 단일 그룹이었을 경우에 취해졌을 2개의 세트들에 기초하여 유도된다.
- [0046] 또 추가의 양태에 따르면, 하나 이상의 이미지들을 비트스트림으로 인코딩하거나 비트스트림으로부터 하나 이상의 이미지들을 디코딩하는 방법이 제공되고, 방법은 본 발명의 제1 양태의 방법에 따른 연관된 재구성된 제2-컴포넌트 샘플 값으로부터 제1-컴포넌트 샘플 값을 획득하기 위한 선형 모델을 유도하는 단계를 포함한다.
- [0047] 적당하게도, 방법은 프로세싱되어야 할 이미지의 현재의 블록에 대한 제1-컴포넌트 샘플 값을 획득하기 위한 복수의 선형 모델 유도 모드들 중의 하나를 선택하는 단계를 더 포함하고, 여기서, 복수의 선형 모델 유도 모드들은 단일 선형 모델을 이용하는 제1 모드, 및 하나 초과인 선형 모델들을 이용하는 제2 모드를 포함하고, 유도된 선형 모델은 선택된 선형 모델 유도 모드에서 이용가능하다. 적당하게도, 오직 제1 모드가 유도된 선형 모델을 이용한다. 대안적으로, 오직 제2 모드가 유도된 선형 모델을 이용한다.
- [0048] 또 추가의 양태에 따르면, 연관된 재구성된 제2-컴포넌트 샘플 값으로부터 제1-컴포넌트 샘플 값을 획득하기 위한 선형 모델을 유도하기 위한 디바이스가 제공되고, 디바이스는 본 발명의 제1 양태의 방법을 수행하도록 구성된다.
- [0049] 또 추가의 양태에 따르면, 하나 이상의 이미지들을 비트스트림으로 인코딩하거나 비트스트림으로부터 하나 이상의 이미지들을 디코딩하기 위한 디바이스가 제공되고, 디바이스는 본 발명의 제2 양태의 방법을 수행하도록 구성된다.
- [0050] 또 추가의 양태에 따르면, 연관된 재구성된 제2-컴포넌트 샘플 값으로부터 제1-컴포넌트 샘플 값을 획득하는 방법이 제공되고, 방법은: 제1-컴포넌트 샘플 값을 획득하기 위한 복수의 선형 모델 모드들로부터 하나의 선형 모델 모드를 선택하는 단계; 및 선택된 선형 모델 모드를 이용하여 제1-컴포넌트 샘플 값을 획득하는 단계 - 복수의 선형 모델 모드들 중의 적어도 하나는 본 발명의 제1 양태에 따른 유도의 방법을 이용하여 유도된 선형 모델을 이용함 - 를 포함한다. 적당하게도, 복수의 선형 모델 모드들은 단일 선형 모델을 이용하는 제1 모드, 및 하나 초과인 선형 모델들을 이용하는 제2 모드를 포함한다. 적당하게도, 오직 제1 모드가 본 발명의 제1 양태에 따른 유도의 방법을 이용한다. 대안적으로, 오직 제2 모드가 본 발명의 제1 양태에 따른 유도의 방법을 이용한다.
- [0051] 또 추가의 양태에 따르면, 연관된 재구성된 제2-컴포넌트 샘플 값으로부터 제1-컴포넌트 샘플 값을 획득하기 위한 디바이스가 제공되고, 디바이스는 본 발명의 제5 양태의 방법을 수행하도록 구성된다.
- [0052] 또 추가의 양태에 따르면, 하나 이상의 이미지들을 비트스트림으로 인코딩하는 방법이 제공되고, 여기서, 방법은 본 발명의 제5 양태에 따른 연관된 재구성된 제2-컴포넌트 샘플 값으로부터 제1-컴포넌트 샘플 값을 획득하

는 단계를 포함한다. 적당하게도, 방법은 비트스트림에서, 제1-컴포넌트 샘플을 획득하기 위하여 이용가능한 선형 모델 모드에 대한 선택을 표시하는 정보를 제공하는 단계를 더 포함한다.

- [0053] 또 추가의 양태에 따르면, 비트스트림으로부터 하나 이상의 이미지들을 디코딩하는 방법이 제공되고, 여기서, 방법은 본 발명의 제5 양태에 따른 연관된 재구성된 제2-컴포넌트 샘플 값으로부터 제1-컴포넌트 샘플 값을 획득하는 단계를 포함한다. 적당하게도, 방법은 비트스트림으로부터, 제1-컴포넌트 샘플을 획득하기 위하여 이용가능한 선형 모델 모드에 대한 선택을 표시하는 정보를 획득하는 단계를 더 포함하고, 복수의 선형 모델 모드들로부터 하나의 선형 모델 모드를 선택하는 단계는 획득된 정보에 기초하여 수행된다.
- [0054] 또 추가의 양태에 따르면, 하나 이상의 이미지들을 비트스트림으로 인코딩하기 위한 디바이스가 제공되고, 디바이스는 본원에서 설명된 방법을 수행하도록 구성된다.
- [0055] 또 추가의 양태에 따르면, 실행 시에, 본원에서 설명된 방법이 수행되게 하는 컴퓨터 프로그램, 및 본원에서 설명된 방법을 구현하기 위한 명령어들을 저장하는 (비-일시적) 컴퓨터-판독가능 매체가 제공된다.
- [0056] 본 발명에 따르면, 첨부된 청구항들에서 기재된 바와 같은 디바이스, 방법, 컴퓨터 프로그램(제품), 및 컴퓨터 판독가능 저장 매체가 제공된다. 발명의 실시예들의 다른 특징들은 첨부된 청구항들, 및 뒤따르는 설명에서 정의된다. 이 특징들의 일부는 방법을 참조하여 여기 이하에서 설명되는 한편, 이들은 디바이스에 전용된 시스템 특징들로 대치(transpose)될 수 있다.
- [0057] 발명에 따른 방법들의 적어도 일부들은 컴퓨터 구현될 수 있다. 따라서, 본 발명은 전적으로 하드웨어 실시예, (펌웨어, 상주 소프트웨어, 마이크로-코드 등을 포함하는) 전적으로 소프트웨어 실시예, 또는 "프로세서 및 메모리", "회로", "모듈", 또는 "시스템"으로서 본원에서 모두 일반적으로 지칭될 수 있는 소프트웨어 및 하드웨어 양태들을 조합하는 실시예의 형태를 취할 수 있다. 또한, 본 발명은 매체에서 구체화된 컴퓨터 이용가능 프로그램 코드를 가지는 표현의 임의의 유형의 매체에서 구체화된 컴퓨터 프로그램 제품의 형태를 취할 수 있다.
- [0058] 본 발명은 소프트웨어로 구현될 수 있으므로, 본 발명은 임의의 적당한 캐리어 매체(carrier medium) 상에서 프로그래밍가능한 장치로의 제공을 위한 컴퓨터 판독가능 코드로서 구체화될 수 있다. 유형의 캐리어 매체(tangible carrier medium)는 하드 디스크 드라이브, 자기 테이프 디바이스 또는 솔리드 스테이트(solid state) 메모리 디바이스 등과 같은 저장 매체를 포함할 수 있다. 순시적 캐리어 매체(transient carrier medium)는 전기적 신호, 전자 신호, 광학적 신호, 음향 신호, 자기적 신호, 또는 전자기적 신호, 예컨대, 마이크로파 또는 RF 신호와 같은 신호를 포함할 수 있다.

**도면의 간단한 설명**

- [0059] 발명의 실시예들은 오직 예로서, 그리고 다음의 도면들을 참조하여 이제 설명될 것이다:
  - 도 1은 비디오 인코더 논리적 아키텍처를 예시하고;
  - 도 2는 도 1에서 예시된 비디오 인코더 논리적 아키텍처에 대응하는 비디오 디코더 논리적 아키텍처를 예시하고;
  - 도 3은 4:2:0 샘플링을 위한 YUV 샘플링 방식의 예들을 개략적으로 예시하고;
  - 도 4는 플로우차트를 이용하여, 인코더 또는 디코더 중의 어느 하나에 의해 수행된 LM 모드를 이용하여 블록 예측자를 생성하기 위한 일반적인 단계들을 예시하고;
  - 도 5는 크로마 블록, 및 종래 기술에서 알려진 바와 같은, 루마 샘플들 및 이웃하는 크로마 및 루마 샘플들의 다운-샘플링을 갖는 연관된 또는 공동위치된 루마 블록을 개략적으로 예시하고;
  - 도 6은 종래 기술에서 알려진 바와 같은 LM 파라미터 유도를 위한 예시적인 샘플 세트들을 개략적으로 예시하고;
  - 도 7은 종래 기술에서 알려진 일부 다운-샘플링 필터들을 예시하고;
  - 도 8은 LM 모드들을 시그널링하기 위한 시그널링 플래그(signalling flag)들의 예시적인 코딩을 예시하고;
  - 도 9는 루마 및 크로마 이웃하는 샘플들의 포인트들, 및 발명의 하나의 실시예에서 획득된 선형 모델 파라미터들을 표현하는 직선 라인을 예시하고;
  - 도 10은 발명의 하나의 실시예에서의 단순화된 LM 유도의 프로세스의 주요 단계들을 예시하고;

도 11은 루마 및 크로마 이웃하는 샘플들의 몇몇 포인트들, 및 발명의 일부 실시예들에서 최상의 2개의 포인트들을 결정하기 위하여 이용된 세그먼트(segment)들을 예시하고;

도 12는 발명의 하나의 실시예에서의 MMLM 유도의 프로세스의 주요 단계들을 예시하고; 그리고

도 13은 발명의 하나 이상의 실시예들의 구현을 위한 컴퓨팅 디바이스의 개략적인 블록도이다.

**발명을 실시하기 위한 구체적인 내용**

- [0060] 도 1은 비디오 인코더 아키텍처를 예시한다. 비디오 인코더에서, 원래의 시퀀스(101)는 HEVC를 위한 코딩 블록들 또는 코딩 유닛들로 칭해진 픽셀들의 블록들로 분할된다(102). 코딩 모드는 그 다음으로, 각각의 블록에 영향이 미친다. 비디오 코딩에서 전형적으로 이용된 코딩 모드들의 2개의 계열들: 공간적 예측에 기초한 코딩 모드들 또는 "INTRA 모드들"(103), 및 시간적 예측에 기초한 코딩 모드들 또는 모션 추정(104) 및 모션 보상(105)에 기초한 "INTER 모드들"이 있다.
- [0061] INTRA 코딩 블록은 일반적으로, INTRA 예측으로 칭해진 프로세스에 의해 그 인과적 경계에서 인코딩된 픽셀들로부터 예측된다. INTRA 코딩 블록의 각각의 픽셀에 대한 예측자는 이에 따라, 예측자 블록을 형성한다. 어느 픽셀들이 INTRA 코딩 블록을 예측하기 위하여 이용되는지에 따라, 다양한 INTRA 모드들: 예를 들어, DC 모드, 평면 모드, 및 각도 모드가 제안된다.
- [0062] 도 1은 비디오 인코더 아키텍처의 일반적인 설명에 대한 것이지만, 픽셀은 여기에서, 몇몇 컴포넌트들, 예를 들어, 적색 컴포넌트, 녹색 컴포넌트, 및 청색 컴포넌트로 전형적으로 구성되는 이미지의 엘리먼트에 대응한다는 것이 주목되어야 한다. 이미지 샘플은 오직 하나의 컴포넌트를 포함하는 이미지의 엘리먼트이다.
- [0063] 시간적 예측은 먼저, 참조 프레임(116)으로 칭해진 이전의 또는 미래의 프레임에서, 모션 추정 단계(104)에서 코딩 블록에 가장 근접한 참조 에어리어(reference area)를 발견하는 것에 있다. 이 참조 에어리어는 예측자 블록을 구성한다. 다음으로, 이 코딩 블록은 모션 보상 단계(105)에서 잔차 또는 잔차 블록을 연산하기 위하여 예측자 블록을 이용하여 예측된다.
- [0064] 양자의 경우들, 공간적 및 시간적 예측에서는, 잔차 또는 잔차 블록은 코딩 블록으로부터 획득된 예측자 블록을 감산함으로써 연산된다.
- [0065] INTRA 예측에서는, 예측 모드가 인코딩된다.
- [0067] \*시간적 예측에서는, 이용된 참조 프레임을 표시하는 인덱스, 및 참조 프레임에서의 참조 에어리어를 표시하는 모션 벡터(motion vector)가 인코딩된다. 그러나, 모션 벡터 인코딩에 관련된 비트레이트 코스트(bitrate cost)를 추가로 감소시키기 위하여, 모션 벡터는 직접적으로 인코딩되지 않는다. 실제로, 모션이 균질(homogeneous)인 것으로 가정하면, 모션 벡터를 이 모션 벡터와, 그 주변들에서의 모션 벡터(또는 모션 벡터 예측자) 사이의 차로써 인코딩하는 것이 특히 유리하다. 예를 들어, H.264/AVC 코딩 표준에서, 모션 벡터들은 현재의 블록의 상부 및 좌측 상에 위치한 3개의 블록들과 연관된 모션 벡터들로부터 연산된 중위 벡터(median vector)에 대하여 인코딩된다. 중위 벡터와 현재의 블록 모션 벡터 사이에서 연산된, 잔차 모션 벡터(residual motion vector)로 또한 칭해진 차만이 비트스트림에서 인코딩된다. 이것은 모듈 "Mv 예측 및 코딩"(117)에서 프로세싱된다. 각각의 인코딩된 벡터의 값은 모션 벡터 필드(118)에서 저장된다. 예측을 위하여 이용된 이웃하는 모션 벡터들은 모션 벡터 필드(118)로부터 추출된다.
- [0068] HEVC 표준은 3개의 상이한 INTER 모드들: 비트스트림(110)에서의 모션 정보(즉, 모션 벡터, 및 그 소위 참조 프레임 인덱스를 통한 연관된 참조 프레임)의 시그널링에 의해 서로 주로 상이한 인터 모드(Inter mode), 병합 모드(Merge mode), 및 병합 스킵 모드(Merge Skip mode)를 이용한다. 단순함을 위하여, 모션 벡터 및 모션 정보는 이하에서 융합된다. 모션 벡터 예측에 관하여, HEVC는 각각 인터 또는 병합 모드에 대한 최상의 모션 벡터 예측자 또는 최상의 모션 정보를 발견하기 위하여 레이트-왜곡 경합 동안에 평가되는 모션 벡터 예측자의 몇몇 후보들을 제공한다. 모션 정보의 최상의 예측자들 또는 최상의 후보에 대응하는 인덱스는 비트스트림(110)에서 삽입된다. 이 시그널링 때문에, 디코더는 예측자들 또는 후보들의 동일한 세트를 유도할 수 있고, 디코딩된 인덱스에 따른 최상의 것을 이용한다.
- [0069] 모션 벡터 예측자들 및 후보들의 유도의 설계는 복잡도에 대한 큰 영향 없이 최상의 코딩 효율을 달성하는 것에 기여한다. 2개의 모션 벡터 유도들: (진보된 모션 벡터 예측(Advanced Motion Vector Prediction)(AMVP)으로서 알려진) 인터 모드에 대한 하나 및 (병합 유도 프로세스로서 알려진) 병합 모드들에 대한 하나가 HEVC에서

제안된다.

- [0070] 다음으로, 현재 고려된 코딩 블록에 대한 레이트-왜곡 기준을 최적화하는 코딩 모드는 모듈(106)에서 선택된다. 획득된 잔차 데이터 내의 중복성들을 추가로 감소시키기 위하여, 변환(transform), 전형적으로 DCT는 모듈(107)에서 잔차 블록에 적용되고, 양자화(quantization)는 모듈(108)에서 획득된 계수들에 적용된다. 계수들의 양자화된 블록은 그 다음으로, 모듈(109)에서 엔트로피 코딩(entropy code)되고, 결과는 비트스트림(110)으로 삽입된다.
- [0071] 인코더는 그 다음으로, 모듈들(111 내지 116)에서 미래의 모션 추정을 위한 프레임의 인코딩된 블록들의 각각의 디코딩을 수행한다. 이 단계들은 인코더 및 디코더가 동일한 참조 프레임들(116)을 가지는 것을 허용한다. 코딩된 프레임을 재구성하기 위하여, 양자화되고 변환된 잔차 블록들의 각각은 픽셀 도메인(pixel domain)에서 대응하는 "재구성된" 잔차 블록을 제공하기 위하여, 모듈(111)에서 역양자화(inverse quantize)되고, 모듈(112)에서 역변환(inverse transform)된다. 양자화의 손실로 인해, 이 "재구성된" 잔차 블록은 단계(106)에서 획득된 원래의 잔차 블록과 상이하다.
- [0072] 다음으로, 106에서 선택된 코딩 모드(INTER 또는 INTRA)에 따르면, 이 "재구성된" 잔차 블록은 "사전-재구성된(pre-reconstructed)" 블록(코딩 블록)을 획득하기 위하여, INTER 예측자 블록(114) 또는 INTRA 예측자 블록(113)에 가산된다.
- [0073] 다음으로, "사전-재구성된" 블록들은 "재구성된" 블록들(코딩 블록들)을 획득하기 위하여 하나 또는 몇몇 종류들의 사후 필터링(post filtering)에 의해 모듈(115)에서 필터링된다. 동일한 사후 필터들은 인코더 및 디코더 단계에서 정확하게 동일한 참조 프레임들을 획득하기 위하여 동일한 방식으로 이용되도록, (디코딩 루프에서의) 인코더 및 디코더에서 통합된다. 이 사후 필터링의 목적은 압축 아티팩트(compression artefact)들을 제거하기 위한 것이다.
- [0074] 도 2는 도 1에서 예시된 비디오 인코더 아키텍처에 대응하는 비디오 디코더 아키텍처를 예시한다.
- [0075] 비디오 스트림(201)은 먼저, 모듈(202)에서 엔트로피 디코딩(entropy decode)된다. 각각의 획득된 잔차 블록(코딩 블록)은 그 다음으로, "재구성된" 잔차 블록을 획득하기 위하여, 모듈(203)에서 역양자화되고, 모듈(204)에서 역변환된다. 이것은 인코더 단계에서의 디코딩 루프의 시작부와 유사하다.
- [0076] 다음으로, 비트스트림(201)에서 표시된 디코딩 모드(INTRA 유형 디코딩 또는 INTER 유형 디코딩 중의 어느 하나)에 따르면, 예측자 블록이 구축된다.
- [0077] INTRA 모드의 경우에, INTRA 예측자 블록은 비트스트림(201)에서 특정된 INTRA 예측 모드에 기초하여 결정된다(205).
- [0078] INTER 모드의 경우에, 모션 정보는 엔트로피 디코딩(202) 동안에 비트스트림으로부터 추출된다. 모션 정보는 예를 들어, HEVC 및 JVET에서, 참조 프레임 인덱스 및 모션 벡터 잔차로 구성된다.
- [0079] 모션 벡터 예측자는 모션 벡터 필드 데이터(211)에서 저장된 이미 연산된 모션 벡터들을 이용하여 (이웃하는 블록들로부터) 인코더에 의해 행해진 것과 동일한 방식으로 획득된다. 그것은 이에 따라, 모션 벡터를 획득하기 위하여 추출된 모션 벡터 잔차 블록에 가산된다(210). 이 모션 벡터는 다음 디코딩된 모션 벡터들의 예측을 위해 이용되도록 하기 위하여 모션 벡터 필드 데이터(211)에 가산된다.
- [0080] 모션 벡터는 또한, INTER 예측자 블록인 참조 프레임(206)에서 참조 에어리어를 위치시키기 위하여 이용된다.
- [0081] 다음으로, 204에서 획득된 "재구성된" 잔차 블록은 인코더의 디코딩 루프와 동일한 방식으로 "사전-재구성된" 블록(코딩 블록)을 획득하기 위하여, INTER 예측자 블록(206) 또는 INTRA 예측자 블록(205)에 가산된다.
- [0082] 다음으로, 이 "사전-재구성된" 블록은 인코더 단계에서 행해진 바와 같이 모듈(207)에서 사후 필터링된다(이용하기 위한 사후 필터링의 시그널링은 비트스트림(201)으로부터 추출될 수 있음).
- [0083] 디코더의 출력으로서 압축해제된(de-compressed) 비디오(209)를 형성하는 "재구성된" 블록(코딩 블록)이 이에 따라 획득된다.
- [0084] 전술한 인코딩/디코딩 프로세스는 단색 프레임(monochrome frame)들에 적용될 수 있다. 그러나, 가장 보편적인 프레임들은 컬러 샘플들의 3개의 어레이(array)들로 일반적으로 이루어진 컬러 프레임들이고, 각각의 어레이는 "컬러 컴포넌트", 예를 들어, R(적색), G(녹색), 및 B(청색)에 대응한다. 이미지의 픽셀은 각각의 컴포넌트에

대하여 하나씩인 3개의 공동위치된/대응하는 샘플들을 포함한다.

- [0085] R, G, B 컴포넌트들은 이들 사이의 통상적으로 높은 상관(correlation)을 가진다. 이에 따라, 컬러 컴포넌트들을 또 다른 컬러 공간에서 변환함으로써, 프레임들을 프로세싱하기 이전에 컬러 컴포넌트들을 역상관(decorrelate)시키는 것은 이미지 및 비디오 압축에서 매우 보편적이다. 가장 보편적인 포맷은 YUV(YCbCr)이고, 여기서, Y는 루마(또는 휘도(luminance)) 컴포넌트이고, U(Cb) 및 V(Cr)는 크로마(또는 색차(chrominance)) 컴포넌트들이다.
- [0086] 프로세싱하기 위한 데이터의 양을 감소시키기 위하여, 컬러 프레임들의 일부 컬러 컴포넌트들은 서브샘플링(subsample)될 수 있어서, 3개의 컬러 컴포넌트들에 대한 상이한 샘플링 비율들을 가지는 것으로 귀착될 수 있다. 서브샘플링 방식은 개념적인 2-픽셀-하이(2-pixel-high) 영역에서의 루마 및 크로마 샘플들의 수를 설명하는 3 부분 비율(three part ratio) J:a:b로서 보편적으로 표현된다. 'J'는 개념적인 영역의 수평 샘플링 참조(즉, 픽셀들에서의 폭), 통상적으로 4를 정의한다. 'a'는 J 픽셀들의 제1 행에서의 크로마 샘플들(Cr, Cb)의 수를 정의하는 반면, 'b'는 J 픽셀들의 제2 행에서의 (추가적인) 크로마 샘플들(Cr, Cb)의 수를 정의한다.
- [0087] 서브샘플링 방식들로, 크로마 샘플들의 수는 루마 샘플들의 수와 비교하여 감소된다.
- [0088] 4:4:4 YUV 또는 RGB 포맷은 서브샘플링을 제공하지 않고, 루마 및 크로마 프레임들이 동일한 크기 W x H를 가지는 비-서브샘플링된(non-sampled) 프레임에 대응한다.
- [0089] 4:0:0 YUV 또는 RGB 포맷은 오직 하나의 컬러 컴포넌트를 가지고, 이에 따라, 단색 프레임에 대응한다.
- [0090] 예시적인 샘플링 포맷들은 다음과 같다.
- [0091] 4:2:0 YUV 포맷은 제1 행에서 루마 샘플들의 절반만큼 많은 크로마 샘플들을 가지고, 제2 행에서 크로마 샘플들을 가지지 않는다. 2개의 크로마 프레임들은 이에 따라, W/2-픽셀 폭 및 H/2-픽셀 높이이고, 여기서, 루마 프레임은 W x H이다.
- [0092] 4:2:2 YUV 포맷은 제1 행에서 루마 샘플들의 절반만큼 많은 크로마 샘플들, 및 제2 행에서 루마 샘플들의 절반만큼 많은 크로마 샘플들을 가진다. 2개의 크로마 프레임들은 이에 따라, W/2-픽셀 폭 및 H-픽셀 높이이고, 여기서, 루마 프레임은 W x H이다.
- [0093] 4:1:1 YUV 포맷은 제1 행에서 루마 샘플들보다 75% 더 적은 크로마 샘플들, 및 제2 행에서 루마 샘플들보다 75% 더 적은 크로마 샘플들을 가진다. 2개의 크로마 프레임들은 이에 따라, W/4-픽셀 폭 및 H-픽셀 높이이고, 여기서, 루마 프레임은 W x H이다.
- [0094] 서브샘플링될 때, 프레임들에서의 크로마 샘플들의 포지션들은 루마 샘플 포지션들과 비교하여 시프팅(shift)된다.
- [0095] 도 3은 4:2:0 YUV 프레임에 대한 루마 샘플들(원(circle)들)에 대한 크로마 샘플들(삼각형(triangle)들)의 예시적인 위치결정을 예시한다.
- [0096] 도 1의 인코딩 프로세스는 입력 프레임의 각각의 컬러-컴포넌트 프레임에 적용될 수 있다.
- [0097] (RGB-대-YUV(RGB-to-YUV) 변환에도 불구하고 YUV 사이의 RGB 또는 나머지 상관들 사이의) 컬러 컴포넌트들 사이의 상관들로 인해, 교차-컴포넌트 예측(CCP) 방법들은 코딩 효율을 개선시키기 위하여 이 (나머지) 상관들을 활용하도록 개발되었다.
- [0098] CCP 방법들은 인코딩 또는 디코딩 프로세스의 상이한 스테이지(stage)들에서, 특히, (현재의 컬러 컴포넌트를 예측하기 위한) 제1 예측 스테이지 또는(컴포넌트의 현재의 잔차 블록을 예측하기 위한) 제2 예측 스테이지 중의 어느 하나에서 적용될 수 있다.
- [0099] 하나의 알려진 CCP 방법은 CCLM(Cross-Component Linear Model prediction)(교차-컴포넌트 선형 모델 예측)으로서 또한 지칭된 LM 모드이다. LM 모드는 루마 Y로부터, 더 구체적으로, (인코더 단 또는 디코더 단에서의) 재구성된 루마로부터 양자의 크로마 컴포넌트들 Cb 및 Cr(또는 U 및 V)을 예측하기 위하여 이용된다. 하나의 예측자가 각각의 컴포넌트에 대하여 생성된다. 방법은 (크로마 및 루마) 블록 레벨에서, 예를 들어, CTU(coding tree unit)(코딩 트리 유닛), CU(coding unit)(코딩 유닛) 레벨, PU(prediction unit)(예측 유닛) 레벨, 서브-PU 또는 TU(transform unit)(변환 유닛) 레벨에서 동작한다.
- [0100] 도 4는 플로우차트를 이용하여, (이하에서 참조로서 이용된) 인코더 또는 디코더 중의 어느 하나에 의해 수행된

LM 모드를 이용하여 블록 예측자를 생성하기 위한 일반적인 단계들을 예로서 예시한다.

- [0101] 이하의 설명에서, 예시적인 제1 컴포넌트는 크로마인 반면, 예시적인 제2 컴포넌트는 루마이다.
  - [0102] 인코딩하거나 디코딩하기 위한 현재의 크로마 블록(502)(도 5a) 및 동일한 프레임에서의 (즉, 예를 들어, 동일한 CU의) 그 연관된 또는 대응하는(즉, "공동위치된") 루마 블록(505)을 고려하면, 인코더(또는 디코더)는 단계(401)에서, 현재의 루마 블록과 이웃하는 루마 샘플들(503)을 포함하는 이웃하는 루마 샘플 세트 RecL을 수신하고, 402로 나타낸 현재의 크로마 블록과 이웃하는 크로마 샘플들(501)을 포함하는 이웃하는 크로마 샘플 세트 RecC를 수신한다. 일부 크로마 샘플링 포맷들 및 크로마 국면(chroma phase)에 대하여, 루마 샘플들(504 및 503)은 도 5a에서 도시된 바와 같이 루마 블록(505)에 직접적으로 인접하지 않는다는 것이 주목되어야 한다. 예를 들어, 도 5a에서는, 좌측 행 RecL'(503)을 획득하기 위하여, 직접적인 좌측 행이 아니라, 오직 제2 좌측 행이 필요하다. 동일한 방식으로, 상부 라인(504)에 대하여, 제2 상부 라인은 또한, 도 5a에서 도시된 바와 같은 루마 샘플의 다운-샘플링을 위하여 고려된다.
  - [0103] 크로마 샘플링 포맷이 이용될 때(예컨대, 4:2:0, 4:2:2 등), 이웃하는 루마 샘플 세트는 크로마 해상도(즉, 대응하는 크로마 프레임/블록의 샘플 해상도)와 정합하기 위하여 단계(403)에서 RecL'(404)으로 다운-샘플링된다. RecL'은 이에 따라, 다운-샘플링되는 현재의 루마 블록과 이웃하는 재구성된 루마 샘플들(504)을 포함한다. 다운-샘플링 때문에, RecL' 및 RecC는 동일한 수 2N의 샘플들을 포함한다(크로마 블록(502)은 N x N임). 하지만, 루마 경계의 특정한 다운-샘플링들은 RecL'을 획득하기 위하여 더 적은 샘플들이 필요한 종래 기술에서 존재한다. 추가적으로, RecL 및 RecC가 동일한 해상도를 가지더라도, RecL'은 저역-통과 콘볼루션 필터(low-pass convolution filter)의 이용을 통해, RecL의 잡음제거된 버전으로서 보여질 수 있다.
  - [0104] 도 5a의 예에서, 이웃하는 루마 및 크로마 샘플 세트들은 각각 다운-샘플링된 상부 및 좌측 이웃하는 루마 샘플들 및 상부 및 좌측 이웃하는 크로마 샘플들로 이루어진다. 더 정확하게, 2개의 샘플 세트들의 각각은 좌측 경계에 바로 인접한 제1 라인, 및 그 개개의 루마 또는 크로마 블록의 상부 경계에 바로 인접한 제1 라인으로 이루어진다. 다운-샘플링(도 5a에서의 4:2:0)으로 인해, 이웃하는 루마 샘플들 RecL'의 단일 라인은 비 다운-샘플링된 재구성된 루마 샘플들 RecL(좌측 또는 상부)의 2개의 라인들로부터 획득된다.
  - [0105] US 9,565,428은 (단계(408)를 참조하여 이하에서 설명된 바와 같이) 루마 블록 자체에 대한 것이 아니라, 오직(즉, 루마 블록의 상부 경계에 인접한) 상부 라인에 대하여 단일 샘플을 선택하는 서브-샘플링을 이용하는 것을 시사한다. 제안된 서브-샘플링은 도 6a에서 예시된다. 이 접근법을 위한 동기부여는 상부 라인의 라인 버퍼를 감소시키기 위한 것이다.
  - [0106] 하나 또는 두 개의 파라미터들(기울기  $\alpha$  및 오프셋  $\beta$ )에 의해 정의되는 선형 모델은 RecL'(존재할 경우, 그렇지 않으면 RecL) 및 RecC로부터 유도된다. 이것은 파라미터들(406)을 획득하기 위한 단계(405)이다.
  - [0107] LM 파라미터들  $\alpha$  및  $\beta$ 는 다음의 수학적식들을 이용하는 최소 제곱 평균(least mean square)-기반 방법을 이용하여 획득된다.
- $$\alpha = \frac{M \cdot \sum_{i=1}^M RecC_i \cdot RecL'_i - \sum_{i=1}^M RecC_i \cdot \sum_{i=1}^M RecL'_i}{M \cdot \sum_{i=1}^M RecL'^2_i - (\sum_{i=1}^M RecL'_i)^2} = \frac{A_1}{A_2}$$

$$\beta = \frac{\sum_{i=1}^M RecC_i - \alpha \cdot \sum_{i=1}^M RecL'_i}{M}$$
- [0108]
  - [0109] 여기서, M은 고려된 블록의 크기에 종속되는 값이다. 도 5a 및 도 5b에서 도시된 바와 같이 정사각형 블록들의 일반적인 경우들에는, M=2N이다. 그러나, LM-기반 CCP는 임의의 블록 형상에 적용될 수 있고, 여기서, M은 예를 들어, (직사각형 블록 형상에 대하여) 블록 높이 H 플러스(plus) 블록 폭 W의 합이다.
  - [0110] 이 수학적식에서 가중치(weight)로서 이용된 M의 값은 인코더 및 디코더에서의 연산 오버플로우(computational overflow)들을 회피하기 위하여 조절될 수 있다는 것이 주목되어야 한다. 정확하게 하기 위하여, 32-비트 또는 64-비트 부호형 아키텍처들을 갖는 산술(arithmetic)을 이용할 때, 연산들의 일부는 때때로 오버플로우할 수 있고, 이에 따라, (임의의 교차 플랫폼 표준에서 엄격하게 금지되는) 비특정된 거동을 야기시킬 수 있다. 이 상황에 직면하기 위하여, 최대 크기의 가능한 주어진 입력들 RecL' 및 RecC 값들이 평가될 수 있고, M(및 궁극적으로 위의 합들)은 오버플로우가 발생하지 않는다는 것을 보장하기 위하여 이에 따라 스케일링될 수 있다.

- [0111] 파라미터들의 유도는 통상적으로, 도 5a에서 도시된 샘플 세트들  $RecL'$  및  $RecC$ 로부터 이루어진다.
  - [0112] 샘플 세트들의 변형들이 제안되었다.
  - [0113] 예를 들어, US 9,288,500은, 상부 경계에 인접한 외부 라인 및 좌측 경계에 인접한 외부 라인으로 이루어진 제1 샘플 세트, 상부 경계에 인접한 외부 라인으로 오직 이루어진 제2 샘플 세트, 및 좌측 경계에 인접한 외부 라인으로 오직 이루어진 제3 샘플 세트를 포함하는 3개의 경합하는 샘플 세트들을 제안한다. 이 3개의 샘플 세트들은 오직 크로마 블록에 대하여 도 6b에서 도시된다(그리고 이에 따라, 루마 블록으로 대체될 수 있음).
  - [0114] US 9,462,273은 제2 및 제3 샘플 세트들을 외부 라인들을 연장시키는(통상적으로 그 길이를 2 배로 함) 추가적인 샘플들로 연장시킨다. 연장된 샘플 세트들은 오직 크로마 블록에 대하여 도 6c에서 도시된다. 이 문서는 또한, 비트스트림에서 이용된 LM 모드를 시그널링하기 위한 시그널링 코스트를 감소시키기 위하여 이용가능한 LM 모드들의 수에서의 감소를 제공한다. 감소는 예를 들어, 연관된 루마 블록에 대하여 선택된 인트라 모드에 기초하여 상황적일 수 있다.
  - [0115] US 9,736,487은, US 9,288,500의 것들과 유사하지만, 고려된 경계들에 평행하고 바로 인접한 외부의 이웃하는 샘플들의 2개의 라인들로 매번 이루어진 3개의 경합하는 샘플 세트들을 제안한다. 이 샘플 세트들은 오직 크로마 블록에 대하여 도 6d에서 도시된다.
  - [0116] 또한, US 9,153,040 및 동일한 특허 계의 문서들은, 이전의 세트들로부터 라인 당 더 적은 샘플들을 갖는, 경계 당 단일 라인으로 이루어진 추가적인 샘플 세트들을 제안한다.
  - [0117] 도 4의 프로세스로 돌아가서, 하나 이상의 유도된 파라미터들(406)을 갖는 선형 모델을 이용하면, 크로마 블록(502)에 대한 크로마 인트라 예측자(413)는 이에 따라, 505에서 표현된 현재의 루마 블록의 재구성된 루마 샘플들(407)로부터 획득될 수 있다. 다시, 크로마 샘플링 포맷이 이용될 경우에(예컨대, 4:2:0, 4:2:2 등), 재구성된 루마 샘플들은 크로마 해상도(즉, 대응하는 크로마 프레임/블록의 샘플 해상도)와 정합하기 위하여 단계(408)에서  $L'$ (409)으로 다운-샘플링된다.
  - [0118] 단계(403)에 대한 것과 동일한 다운-샘플링, 또는 라인 버퍼의 이유로 인한 또 다른 것이 이용될 수 있다. 예를 들어, 6-탭 필터는 다운-샘플링 포지션을 포위하는 상부 좌측, 상부, 상부 우측, 하부 좌측, 하부, 및 하부 우측 샘플들의 가중화된 합으로서 다운-샘플링된 값을 제공하기 위하여 이용될 수 있다. 일부 포위하는 샘플들이 결손될 때, 단지 2-탭 필터는 6-탭 필터 대신에 이용된다.
  - [0119] 재구성된 루마 샘플들  $L$ 에 적용되면, 예시적인 6-탭 필터의 출력  $L'$ 이 다음과 같이 획득된다:
- $$L'[i,j] = (2 \times L[2i,2j] + 2 \times L[2i,2j+1] + L[2i-1,2j] + L[2i+1,2j] + L[2i-1,2j+1] + L[2i+1,2j+1] + 4) \gg 3$$
- [0120] 여기서,  $(i,j)$ 는 다운-샘플링된 블록 내의 샘플의 좌표들이고,  $\gg$ 는 비트-우측-시프팅(bit-right-shifting) 동작이다.
  - [0122] 또한, 적응적 루마 다운-샘플링은 US 2017/0244975에서 설명된 바와 같이 이용될 수 있다. 오직 루마 블록의 내용은 루마 블록의 각각의 재구성된 루마 샘플에 대하여 어느 다운-샘플링 필터가 이용되는지를 결정하기 위하여 이용된다. 1-탭 필터가 이용가능하다. 이 접근법의 동기부여는 다운-샘플링된 루마 블록에서의 에지(edge)의 전파를 회피하기 위한 것이다.
  - [0123] 다운-샘플링 단계(408) 때문에,  $L'$  및  $C$  블록들(크로마 블록(502)에서의 크로마 샘플들의 세트)은 동일한 수  $N2$  개의 샘플들을 포함한다(크로마 블록(502)은  $N \times N$ 임).
  - [0124] 다음으로, 크로마 인트라 예측자  $PredC$ (413)의 각각의 샘플은 공식을 따르는 루프(410-411-412)를 이용하여 계산된다
- $$PredC[i,j] = \alpha \cdot L'[i,j] + \beta$$
- [0125] 여기서,  $(i,j)$ 는 크로마 및 루마 블록들 내의 모든 샘플들의 좌표들이다.
  - [0127] 계산(division)들 및 승산(multiplication)들을 회피하기 위하여, 연산들은 룩업 테이블(look-up table)들 및 시프트 동작들에 기초한 덜 복잡한 방법들을 이용하여 구현될 수 있다. 예를 들어, 실제적인 크로마 인트라 예측자 유도(411)는 다음과 같이 행해질 수 있다:

[0128]  $PredC[i, j] = (A \cdot L'[i, j]) \gg S + \beta$

[0129] 여기서, S는 정수이고, A는 이전에 언급된 룬업 테이블을 이용하여 ( $\alpha$  및  $\beta$ 를 연산할 때에 위에서 도입된) A1 및 A2로부터 유도된다. 그것은 실제로  $\alpha$ 의 리스케일링된 값(rescaled value)에 대응한다. 동작( $x \gg S$ )은  $2^S$ 에 의한 (절단(truncation)을 갖는) x의 정수 제산(integer division)과 동등한 비트-우측-시프팅 동작에 대응한다.

[0130] 다운-샘플링된 루마 블록의 모든 샘플들이 파싱(parse)되었을 때(412), 크로마 인트라 예측자(413)는 인코더 단에서의 (크로마 잔차 블록을 획득하기 위한) 크로마 블록(502)으로부터의 감산을 위하여, 또는 디코더 단에서의 (재구성된 크로마 블록을 획득하기 위한) 크로마 잔차 블록으로의 가산을 위하여 이용가능하다.

[0131] 크로마 잔차 블록은 사소할 수 있고 이에 따라 폐기될 수 있으며, 이 경우에, 획득된 크로마 인트라 예측자(413)는 (크로마 블록(502)을 형성하는) 예측된 크로마 샘플들에 직접적으로 대응한다는 것에 주목한다.

[0132] HEVC 표준을 정의하였던 양자의 표준화 그룹들 ITU-T VCEG(Q6/16) 및 ISO/IEC MPEG(JTC 1/SC 29/WG 11)는 공동 비디오 탐구 팀(JVET)으로서 알려진 공동 협력 노력으로 HEVC의 계승자를 위한 미래의 비디오 코딩 기술들을 연구하고 있다. 공동 탐구 모델(JEM)은 HEVC 툴(tool)들, 및 이 JVET 그룹에 의해 선택된 새로운 추가된 툴들을 포함한다. 특히, 이 참조 소프트웨어는 문서 JVET-G1001에서 설명된 바와 같이, 일부 CCP 툴들을 포함한다.

[0133] JEM에서는, 총 11개의 인트라 모드들이 크로마 코딩을 위하여 허용된다. 그 모드들은 (비트스트림(110, 201)에서 시그널링된 Y로부터 Cb를 예측하기 위한 5개의 전통적인 인트라 모드들 및 6개의 교차-컴포넌트 LM 모드들, 및 Cb로부터 Cr을 예측하기 위한 하나의 교차-컴포넌트 LM 모드를 포함한다.

[0134] 6개의 Y-대-Cb(Y-to-Cb) CC LM 모드들 중의 하나는 위에서 설명된 CCLM이고, 여기서, 이웃하는 루마 및 크로마 샘플 세트들 RecL' 및 RecC는 도 5a에서 도시된 바와 같이, 좌측 경계에 바로 인접한 제1 라인 및 그 개개의 루마 또는 크로마 블록의 상부 경계에 바로 인접한 제1 라인으로 각각 이루어진다.

[0135] 5개의 다른 Y-대-Cb CC LM 모드들은 다중 모델(Multiple Model)(MM)로서 알려진 특정한 유도에 기초한다. 이 모드들은 MMLM으로 표기된다.

[0136] CCLM과 비교하면, MMLM 모드들은 2개의 선형 모델들을 이용한다. RecL' 세트로부터의 이웃하는 재구성된 루마 샘플들 및 RecC 세트로부터의 이웃하는 크로마 샘플들은 2개의 그룹들로 분류되고, 각각의 그룹은 하나의 선형 모델의 파라미터들  $\alpha$  및  $\beta$ 를 유도하기 위하여 이용되어, 이에 따라, 선형 모델 파라미터들 ( $\alpha_1, \beta_1$ ) 및 ( $\alpha_2, \beta_2$ )의 2개의 세트들로 귀착된다.

[0137] 예를 들어, 임계치는 RecL'을 형성하는 이웃하는 재구성된 루마 샘플들의 평균 값으로서 계산될 수 있다. 다음으로,  $RecL'[i, j] \leq$  임계치인 이웃하는 루마 샘플은 그룹 1로 분류되는 반면;  $RecL'[i, j] >$  임계치인 이웃하는 루마 샘플은 그룹 2로 분류된다.

[0138] 다음으로, 크로마 인트라 예측자(또는 현재의 크로마 블록(602)에 대한 예측된 크로마 샘플들)는 다음의 공식들에 따라 획득된다:

[0139]  $PredC[i, j] = \alpha_1 \cdot L'[i, j] + \beta_1, \quad \text{if } L'[i, j] \leq \text{임계치}$

[0139]  $PredC[i, j] = \alpha_2 \cdot L'[i, j] + \beta_2, \quad \text{if } L'[i, j] > \text{임계치}$

[0140] 추가적으로, CCLM과 비교하면, MMLM 모드들은 고려된 블록의 좌측 및 상부 경계들에 평행하고 바로 인접한 외부의 이웃하는 샘플들의 2개의 라인들로 각각 이루어진 이웃하는 루마 및 크로마 샘플 세트들 RecL' 및 RecC를 이용한다. 일 예는 이웃하는 루마 샘플들의 2개의 라인들이 비 다운-샘플링된 재구성된 루마 샘플들의 4개의 라인들로부터 (다운-샘플링을 이용하여) 획득되는 4:2:0 샘플링 포맷을 예시하는 도 5b에서 도시된다.

[0141] 5개의 MMLM 모드들은 크로마 해상도와 정합하기 위하여(RecL' 및/또는 L'을 획득하기 위하여) 재구성된 루마 샘플들을 다운-샘플링하기 위한 5개의 상이한 다운-샘플링 필터들에 의해 서로 상이하다.

[0142] 제1 MMLM 모드는 CCLM에서 이용된 것과 동일한 6-탭 필터에 의존한다(도 7의 참조(701)에서의 6개의 흑색 포인 트들을 참조). 제2 내지 제4 MMLM 모드들은 다음의 가중화된 합으로서 다운-샘플링된 값을 각각 제공하는 2-탭

필터들에 의존한다:

[0143] - 다운-샘플링 포지션을 포위하는 (6-탭 필터에 의해 이용된) 6개의 샘플들의 상부 우측 및 하부 우측 샘플들 (도 7의 필터 1(702)을 참조):

[0144] 
$$L'[i, j] = (L[2i + 1, 2j] + L[2i + 1, 2j + 1] + 1) \gg 1$$
 (동일사항은 RecL로 적용됨),

[0145] - 다운-샘플링 포지션을 포위하는 (6-탭 필터에 의해 이용된) 6개의 샘플들의 하부 및 하부 우측 샘플들(도 7의 필터 2(703)를 참조):

[0146] 
$$L'[i, j] = (L[2i, 2j + 1] + L[2i + 1, 2j + 1] + 1) \gg 1$$
, 및

[0147] - 다운-샘플링 포지션을 포위하는 (6-탭 필터에 의해 이용된) 6개의 샘플들의 상부 및 상부 우측 샘플들(도 7의 필터 4(705)를 참조):

[0148] 
$$L'[i, j] = (L[2i, 2j] + L[2i + 1, 2j] + 1) \gg 1.$$

[0149] 제5 MMLM 모드는 다운-샘플링 포지션을 포위하는 (6-탭 필터에 의해 이용된) 6개의 샘플들의 상부, 상부 우측, 하부, 및 하부 우측 샘플들의 가중화된 합으로서 다운-샘플링된 값을 제공하는 4-탭 필터에 의존한다(도 7의 필터 3(704)을 참조):

$$L'[i, j] = (L[2i, 2j] + L[2i, 2j + 1] + L[2i + 1, 2j] + L[2i + 1, 2j + 1] + 2) \gg$$

[0150] 
$$2.$$

[0151] 위에서 표시된 바와 같이, CCLM 또는 MMLM 모드는 비트스트림(110 또는 201)에서 시그널링되어야 한다. 도 8은 JEM의 예시적인 LM 모드 시그널링을 예시한다. 제1 이진 플래그(binary flag)는 현재의 블록이 LM 모드, 또는 소위 DM 모드들을 포함하는 다른 인트라 모드들을 이용하여 예측되는지 여부를 표시한다. LM 모드의 경우에, 6개의 가능한 LM 모드들이 시그널링될 필요가 있다. (6-탭 필터를 이용하는) 제1 MMLM 모드는 1로 설정된 하나의 제2 이진 플래그로 시그널링된다. 이 제2 이진 플래그는 나머지 모드들에 대하여 0으로 설정되고, 이 경우에, 제3 이진 플래그는 CCLM 모드를 시그널링하기 위하여 1로 설정되고, 나머지 MMLM 모드들에 대하여 0으로 설정된다. 2개의 추가적인 이진 플래그들은 그 다음으로, 4개의 나머지 MMLM 모드들 중의 하나를 시그널링하기 위하여 이용된다.

[0152] 하나의 모드는 각각의 크로마 컴포넌트에 대하여 시그널링된다.

[0153] 위에서 도입된 Cb-대-Cr(Cb-to-Cr) CCLM 모드는 DM 모드들에서 이용되고, 잔차 레벨에서 적용된다. 실제로, DM 모드는 크로마에 대하여, 미리 결정된 위치에서 루마에 의해 이용되었던 인트라 모드를 이용한다. 전통적으로, HEVC와 같은 코딩 모드는 CU의 상부-좌측 코너와 공동-위치된 하나의 단일 DM 모드를 이용한다. 너무 많은 세부사항들로 들어가지 않으면서, 그리고 명확함을 위하여, JVET는 몇몇 이러한 위치들을 제공한다. 이 모드는 그 다음으로, 예측 방법을 결정하기 위하여 이용되어, 참조/원래 데이터로부터 감소될 때, 진술한 잔차 데이터를 산출하는 크로마 컴포넌트에 대한 통상적인 인트라 예측을 생성한다. Cr 잔차에 대한 예측은 다음의 공식에 의해 Cb 잔차(이하의 ResidualCb)로부터 획득된다:

[0154] 
$$PredCr[i, j] = \alpha \cdot ResidualCb[i, j]$$

[0155] 여기서,  $\alpha$ 는 CCLM 루마-대-크로마 예측에서와 유사한 방식으로 유도된다. 유일한 차는 유도된 스케일링 인자 (scaling factor)가 다음과 같이 -0.5의 디폴트 값을 향해 바이어싱되도록, 오차 함수(error function)에서의 디폴트  $\alpha$  값에 대한 회귀 코스트(regression cost)의 추가이다:

[0156] 
$$\alpha = \frac{M \cdot \sum_{i=1}^M RecCb_i \cdot RecCr_i - \sum_{i=1}^M RecCb_i \cdot \sum_{i=1}^M RecL'_i + \lambda (-0.5)}{M \cdot \sum_{i=1}^M RecCb_i^2 - (\sum_{i=1}^M RecCb_i)^2 + \lambda}$$

[0157] 여기서,  $RecCb_i$ 은 이웃하는 재구성된 Cb 샘플들의 값들을 표현하고,  $RecCr_i$ 은 이웃하는 재구성된 Cr 샘플들을 표현하고, 그리고

[0158]  $\lambda = \sum_{i=1}^M RecCb_i^2 \gg 9.$

[0159] 알려진 LM 모드들은 특히, 최소 제곱 기반 방법들을 이용하여 선형 모델 파라미터를 유도할 때에 큰 연산 복잡도를 나타낸다.

[0160] 본 발명은 코딩 효율 및/또는 연산 복잡도의 측면에서 상황을 개선시키는 것을 추구한다.

[0161] 발명은 직선 라인의 수학적식에 기초한 선형 모델의 파라미터들의 결정에 의해 루마 블록 샘플들로부터 크로마 예측자 블록 샘플들을 연산하기 위하여 이용된 선형 모델의 유도의 대체(replacement)에 기초한다. 직선 라인은 블록의 이웃에서의 재구성된 샘플 쌍들에 기초하여 정의된 2개의 샘플 쌍들에 의해 정의된다. 먼저, 이용되어야 할 2개의 샘플 쌍들이 결정된다. 그 다음으로, 선형 모델의 파라미터들은 이 2개의 샘플 쌍들로부터 결정된다. 선형 모델의 결정에서 이용된 샘플 쌍들의 수를 2로 제한함으로써, 최소 제곱 평균 방법의 이용이 회피될 수 있다. 제안된 방법은 그러므로, 최소 제곱 평균 방법을 이용하는 알려진 방법보다 덜 연산 집약적이다.

[0162] 도 9는 현재의 블록의 이웃에서의 샘플 쌍들의 세트에서의 루마 샘플 값들의 최소 및 최대를 여기에서 고려함으로써 이 방법의 원리를 예시한다. 모든 샘플 쌍들은 그 크로마 값 및 그 루마 값에 따라 도면 상에서 그려진다. 2개의 상이한 포인트들, 즉, 포인트 A 및 포인트 B는 도면 상에서 식별되고, 각각의 포인트는 샘플 쌍에 대응한다. 포인트 A는  $RecL'$ 으로부터의 최저 루마 값  $x_A$  및  $RecC$ 로부터의 그 공동위치된 크로마 값  $y_A$  을 갖는 샘플 쌍에 대응한다. 포인트 B는 최고 루마 값  $x_B$  및 그 공동위치된 크로마 값  $y_B$  을 갖는 샘플 쌍에 대응한다.

[0163] 도 10은 선형 모델 파라미터들을 유도하기 위한 제안된 방법의 플로우차트를 제공한다. 이 플로우차트는 도 4의 단순화된 버전이다. 방법은 단계(1001)에서 획득된 이웃하는 루마 샘플들  $RecL'$  및 단계(1002)에서 획득된 크로마 샘플들  $RecC$ 에 기초한다.

[0164] 단계(1003)에서, 2개의 샘플 쌍들에 대응하는 2개의 포인트들 A 및 B(1004)가 결정된다. 제1 실시예에서, 이 2개의 포인트들 A 및 B는 최저 및 최고 루마 샘플 값들  $x_A$  및  $x_B$  및 그 대응하는 크로마 샘플 값들  $y_A$  및  $y_B$  을 각각 갖는 샘플 쌍들에 대응한다.

[0165] 그 다음으로, 포인트들 A 및 B와 교차하는 직선 라인 수학적식은 다음의 수학적식에 따라 단계(1005)에서 연산된다:

[0166] 
$$\alpha = \frac{y_B - y_A}{x_B - x_A}$$

[0167] 
$$\beta = y_A - \alpha x_A$$

[0168] 획득된  $\alpha$ ,  $\beta$  는 크로마 예측자를 생성하기 위하여 이용된 선형 모델 파라미터들(1006)이다.

[0169] 종래 기술에서 이용된 LMS 알고리즘에 기초한 선형 모델 유도는 어떤 복잡도를 가진다. 이 알려진 방법에서, 모델의  $\alpha$  파라미터의 연산은 다음의 수학적식에 의해 획득된다:

[0170] 
$$\alpha = \frac{M \sum_{i=1}^M RecC_i RecL'_i - \sum_{i=1}^M RecC_i \sum_{i=1}^M RecL'_i}{M \sum_{i=1}^M RecL_i'^2 - \left( \sum_{i=1}^M RecL'_i \right)^2} = \frac{B_1 - B_2}{B_3 - B_4} = \frac{A_1}{A_2}$$

[0171] 연산 복잡도에 관한 이 수학적식의 분석은 다음의 결과들을 제공한다.  $B_1$ 의 연산은 M+1 승산들 및 M 합산들을 요구하고, M은 샘플 쌍들의 수이다.  $B_2$ 의 연산은 1 승산 및 2M 합산들을 요구한다.  $B_3$ 의 연산은 M+1 승산 및

M 합산들을 요구하고,  $B_4$ 의 연산은 하나의 승산 및 2M 합산들을 요구한다.  $\frac{B_1 - B_2}{B_3 - B_4}$ 에 대응하는  $\alpha$ 의 연산은 2개의 추가적인 합산들 및 하나의 제산을 요구한다.

[0172]  $\beta$ 를 연산하기 위하여, 하나의 승산 및  $2M+1$  승산들 및 하나의 제산. 이전에 설명된 바와 같이,  $M$ 은 샘플들  $RecC_i$  및  $Recl'_i$ 의 쌍들의 수이다.

[0173]  $\alpha$  및  $\beta$ 의 LMS 유도 복잡도는 그러므로,  $(2M+2+2)$  승산들,  $(7M+3)$  가산들, 및 2개의 제산들이다.

[0174] 비교 시에, 오직 2개의 포인트들을 이용한 직선 라인의 수학적 연산에 기초한 제안된 방법의 분석은 다음의 결과들을 제공한다. 보고된 바와 같이, 유도 단계(1005)는 오직 하나의 승산, 3개의 합산들, 및 하나의 제산을 요구한다. 선형 모델 파라미터들을 생성할 시의 이 큰 복잡도 감소는 제안된 방법의 주요한 장점이다.

[0175] 최소 및 최대 값들에 대한 검색은 전형적으로 소팅 알고리즘(sorting algorithm)에 관련된 그 자신의 복잡도를 가진다는 것이 주목되어야 한다. 동작은 완전히 직렬이 아니다:  $N$  개의 포인트들은  $N$  개의 다른 포인트들과 비교될 수 있어서,  $N$  최소/최대를 생성할 수 있다. 그 다음으로,  $N/2$  최소 및  $N/2$  최대 포인트들은 최소 및 최대 포인트들의 오직 희망된 수들이 남을 때까지,  $N/2$  그 외의 것들, 그 다음으로, 다시  $N/4$  등과 비교될 수 있다. 전형적으로, 최소 및 최대에 대한 검색은 이에 따라, 대략적으로  $2*N-2$  비교들(각각에 대하여  $N-1$ )로 귀착된다.

[0176] 이미 설명된 바와 같이, 크로마 예측자는 부동 소수점(floating-point) 승산 대신에 정수 승산 및 시프트, 그리고 기울기를 연산할 때의 제산으로 계산될 수 있다. 이 단순화는 다음을:

[0177] 
$$pred_c(i, j) = \alpha \cdot rec_L^{(i, j)} + \beta$$

[0178] 이하에 의해 대체하는 것에 있다:

[0179] 
$$pred_c(i, j) = (L \cdot rec_L^{(i, j)}) \gg S + \beta$$

[0180] 오직 정수 승산 및 시프트를 이용하기 위하여, 하나의 실시예에서, 직선 라인 수학적식은 다음과 같이 획득된다:

[0181] 
$$\begin{aligned} S &= 10 \\ L &= \frac{(y_B - y_A) \ll S}{x_B - x_A} \\ \beta &= y_A - L(x_A \gg S) \end{aligned}$$

[0182]  $\beta$ 는  $\alpha$ 가  $L$  및  $S$ 에 의해 대체될 경우에 다음에서의 이 수학적식을 지칭하고, 그렇지 않으면, 그것은 전통적인 수학적식  $\beta = y_A - \alpha x_A$ 을 지칭한다는 것에 주목하십시오.

[0183] 이 유도의 또 다른 장점은 시프트 값  $S$ 가 항상 동일한 값을 가진다는 것이다. 이것은 특히, 이 성질을 활용할 시에 더 간단하게 만들어질 수 있는 하드웨어 구현을 위하여 흥미롭다.

[0184] 또 다른 실시예에서,  $S$ 의 값은  $L$ 이 클 수 있으므로 낮도록 강제되고, 더 큰 승수(multiplier) 동작들을 요구한다. 실제로, 8 비트들 값들 대 8-비트들 값의 승산은 예컨대,  $8*16$  승수보다 구현하기가 훨씬 더 용이하다.  $L$ 에 대한 전형적인 실제적 값들은 8 비트들 미만인 승수와 종종 동등하다.

[0185] 그러나, 바람직한 실시예는 고정 소수점(fixed point)으로서 알려진 구현이고:  $D=(x_B - x_A)$ 의 매 값에 대하여, 아마도 양자화되고(예컨대,  $2D+0$  및  $2D+1$ 에 대한 결과들은 단일의 하나로서 저장됨),  $(1 \ll S)/D$ 의 값은 표(table)에서 저장된다. 바람직하게는, 부호(sign)가 용이하게 추출될 수 있으므로, 이것들은 오직 양수 값들을 위한 것이다. 어레이 TAB를 이용하면,  $L$ 의 연산은 이에 따라 다음으로 된다:

[0186] 
$$L = \begin{cases} (y_B - y_A) * TAB[abs(x_B - x_A)/Q] & \text{if } x_B - x_A \geq 0 \\ -1 * (y_B - y_A) * TAB[abs(x_B - x_A)/Q] & \text{그 외} \end{cases}$$

[0187]  $Q$ 는 양자화 및 이에 따라, 표에서의 엘리먼트들의 수를 제어한다.  $Q=1$ 을 이용하는 것은 이에 따라, 양자화 없음을 의미한다. 또한, 특업된 인덱스는 그 대신에  $(abs(x_B - x_A) + R)/Q$ 일 수 있고, 전형적으로 여기서,  $R=Q/2$ , 또는 제산 버림(division rounding)의 그 변형인 것에 주목한다. 결과적으로,  $Q=2^P$ 에 의한 제산이  $P$ 에

의한 우측-시프트와 동등하도록, Q는 이상적으로는 2의 제곱이다.

[0188] 최종적으로, 그 표에서의 값들의 일부는 0과 동일하지 않을 수 있고:  $abs(x_B - x_A)$  또는  $abs(y_B - y_A)$ 의 낮은 값들은 종종 L의 매우 열악한 추정들로 귀착된다. 미리 결정된 또는 (슬라이스 헤더(slice header), 또는 PPS 또는 SPS와 같은 파라미터 세트에서와 같은) 명시적인 값들이 그 다음으로 이용될 수 있다. 예를 들어, 4 미만인 D의 모든 값들에 대하여, 어레이 TAB는 디폴트 값, 예컨대,  $-(1 \ll S)/8$ 을 포함할 수 있다.

[0189] 10 비트들 내용 및 Q=1에 대하여, 어레이에서의 최대 2048 엔트리들이 필요하다. 위에서 도시된 바와 같은 부호를 갖는 대칭성을 활용함으로써, 이것은 1024로 감소될 수 있다. 추가의 Q를 증가시키는 것은 TAB의 크기를 유사하게 감소시킬 것이다.

[0190] 샘플들의 일부(RecL 또는 RecC 중의 어느 하나, 또는 양자 모두가 (즉, 아마도 양자화된 2개의 블록들 사이의 차로부터 기인하는) 잔차 샘플들일 경우에, 그것은 Cb 대 Cr 예측을 갖는 JVET에서 그러하므로, 표 크기(및 내용)는 이에 따라 적응될 수 있다.

[0191] 또 다른 바람직한 실시예에서, 다음의 공식  $pred_c(i, j) = \alpha \cdot rec_L^{(i, j)} + \beta$ 에서의 직선 라인의 2개의 파라미터들  $\alpha$  및  $\beta$ 의 결정은 정수 산술이 하드웨어로 용이하게 구현되도록 하기 위하여 이용될 수 있도록 연산된다.

[0192] 더 정확하게, 파라미터들  $\alpha$  및  $\beta$ 의 결정은 정수 승산들 및 정수들에 대한 비트 시프트 동작들을 통해 오직 수행될 수 있다. 이러한 계산들은 부동 소수점 산술과 같은 다른 유형들의 계산들보다 더 적은 하드웨어 자원(예컨대, 메모리 및 시간)을 사용한다.

[0193] 이 정수 산술을 수행하기 위하여, 다음의 단계들이 수행된다.

[0194] 제1 중간 값 "pshift"는 취급되어야 할 루마 및 크로마 샘플들의 비트 심도(bit depth)를 참작함으로써 결정된다. 이 비트 시프트 값은  $\alpha$ 의 ('diff'로서 지칭된) 분모의 특정된 최대 값을 보장한다. 본 실시예에서, 'diff'의 최대 값은 512이고, 이와 같이, 512 엔트리들을 가지는 표에서 표현될 수 있다. 'diff'의 특정 최대 값을 강제함으로써, 보편적인 표(또는 표들의 세트)은 총 메모리 요건들을 감소시키는 다양한 상이한 비트심도(bitdepth)들을 위하여 이용될 수 있다.

[0195] pshift의 값은 이에 따라, 샘플들의 비트심도에 종속되고, 이것은 예를 들어, 샘플들이 10 비트들을 이용하여 인코딩될 경우에, 최대 및 최소 사이의 최대 차가 1024이기 때문이다. 512 엔트리들의 표에서 이것을 표현하기 위하여, 최대 차는 2에 의해 계산되어야 하거나 1 비트만큼 비트시프팅되어야 하므로, pshift = 1이다. pshift와 비트심도 사이의 관계는 예를 들어, 다음의 표 1에 대하여 추출될 수 있거나, 다음의 표현식에 의해 주어질 수 있다:

[0196]  $pShift = (BitDepth > 9) ? BitDepth - 9 : 0$

[0197] 이것은 대안적으로, 다음의 표현식에 의해 표현될 수 있다:

[0198]  $pShift = \begin{cases} BitDepth - 9, & \text{if } BitDepth > 9 \\ 0, & \text{그 외} \end{cases}$

[0199] 임의적인 버림 값(rounding value) "add"는 또한, 'diff'를 비트 시프팅을 따르는 정수로 만들기 위하여 연산될 수 있다. 'add' 값은 다음의 표현식에 따른 pshift에 관련된다:

[0200]  $add = pshift ? 1 \ll (pshift - 1) : 0 .$

[0201] 이것은 대안적으로, 다음의 표현식에 의해 표현될 수 있다:

[0202]  $add = \begin{cases} 2^{(pshift-1)}, & \text{if } pshift > 0 \\ 0, & \text{그 외} \end{cases}$

[0203] 대안적으로, 'BitDepth'와 'add' 사이의 직접적인 관계는 다음의 표현식들에 의해 제공될 수 있다:

[0204]  $add = (BitDepth > 9) ? 1 \ll (BitDepth - 10) : 0$

[0205] 또는:

[0206]  $add = \begin{cases} 2^{(Bitdepth-10)} & \text{if } Bitdepth > 9 \\ 0 & \text{그 외} \end{cases}$

[0207] 이하의 표 1은 8로부터 16 비트들로 변동되는 취급되어야 할 루마 및 크로마 샘플들의 비트-심도에 대응하는 "pshift" 및 "add"의 일 예의 값들을 제공한다.

비트-심도	"pShift" 의 값	"add" 의 값
8 또는 9 비트들	0	0
10 비트들	1	1
11 비트들	2	2
12 비트들	3	4
13 비트들	4	8
14 비트들	5	16
15 비트들	6	32
16 비트들	7	64

표 1: "pshift" 및 "add" 값의 예

[0208]

[0209] 이 표는 프로세싱 동작들의 수를 감소시킬 수 있는 'pshift' 및 'add'를 재계산하기 위한 필요성을 회피하기 위하여 메모리에서 저장될 수 있다. 그러나, 어떤 구현들은 프로세싱 동작들의 수에 비해 메모리 이용의 감소를 우선순위화할 수 있고, 이와 같이, 'pshift' 및 'add'는 매번 계산될 수 있다.

[0210] 그 다음으로, "diff" 값은 정수 산술을 이용하는 프로세싱을 위하여 적당한 방식으로 루마 샘플들의 최소 및 최대 값들 사이의 범위 값을 표현한다. 'diff'의 값은 'pshift'의 이용에 의해 어떤 범위 내에서 제약되는 정수이다. 이 값 "diff"는 다음의 공식으로부터 연산된다:

[0211]  $diff = (x_B - x_A + add) \gg pshift$

[0212] 그 다음으로,  $\alpha$  및  $\beta$  파라미터들이 연산되고 -  $\alpha$  및  $\beta$  는 선형 모델 절편 포인트들 A 및 B의 기울기(slope) 및 절편(intercept)을 정의한다는 것이 환기되어야 한다:

[0213]  $\alpha = \frac{y_B - y_A}{x_B - x_A}$

$\beta = y_A - \alpha x_A$

[0214] 표현되는 값 "diff"가 제로(zero)와 동일할 경우에,  $\alpha$  및  $\beta$  파라미터는 다음과 같이 배정된다:

[0215]  $\alpha = 0$

[0215]  $\beta = y_A$  (또는  $\beta = y_B$ )

[0216] 포인트 A 또는 B를 이용하는 것의 선택은 프로세싱 동작들의 수를 감소시키기 위하여 어느 포인트가 현재 메모리에서 저장되는지에 의해 결정될 수 있다.

[0217] 이와 다르게, 값 "diff"가 엄격하게 양수일 경우에,  $\alpha$ 의 값은 다음의 공식에 의해 결정된다

[0218]  $\alpha = (((y_B - y_A) * \text{Floor}(2^k / \text{diff}) + \text{div} + \text{add}) \gg \text{pshift})$  (1)

[0219] 여기서, 함수 Floor(x)는 x 이하인 가장 큰 정수 값을 제공하고,

[0220] 그리고 여기서, 중간 파라미터 "div"는 다음과 같이 연산된다:

[0221]  $\text{div} = ((y_B - y_A) * (\text{Floor}((2^k * 2^k) / \text{diff}) - \text{Floor}(2^k / \text{diff}) * 2^k) + 2^{(k-1)}) \gg k$  (2)

[0222] 이 계산의 정확도는 변수 "k"에 의해 표현된다. k=16의 값은 최상의 코딩 효율을 제공하기 위하여 구해졌고, 정수 산술을 이용하여  $\alpha$  및  $\beta$ 의 정확한 표현을 허용한다. 이것은 그 다음으로, 대응하는 루마 샘플을 이용할 때에 크로마 샘플의 정밀한 예측을 가능하게 할 것이다.

[0223] 이하에서 더 상세하게 설명되는 바와 같이, k의 값은 또한, 각각의 엔트리가 얼마나 많은 메모리를 요구하는지를 정의한다. k=16의 값은 표에서의 엔트리들의 각각을 어드레싱할 때에 이용되도록 하기 위하여 2 바이트들로 표현될 수 있는 16 비트 메모리 레지스터(memory register)를 허용한다.

[0224] 파라미터  $\beta$ 는 이하의 포인트 A

[0225]  $\beta = y_A - ((\alpha * x_A) \gg k),$

[0226] 또는 이하의 포인트 B의 어느 하나일 수 있는 직선 라인의 단일 포인트 상에서 직선-라인 수학적식을 적용함으로써 결정된다

[0227]  $\beta = y_B - ((\alpha * x_B) \gg k).$

[0228] 포인트 A 또는 B를 이용하는 것의 선택은 프로세싱 동작들의 수를 감소시키기 위하여 어느 포인트가 현재 메모리에서 저장되는지에 의해 결정될 수 있다. 대안적으로, 그것은 예를 들어, 표준에서 정의된 고정된 선택일 수 있다.

[0229] 하드웨어로의 구현의 관점으로부터, 공식들 (1) 및 (2)에서의 항들의 일부는 사전연산된 값들이 저장되는 표들로 대체될 수 있다. 이러한 표들의 주요한 장점은  $\alpha$  및  $\beta$  파라미터들의 유도가 수행될 때마다 중간 "Floor" 함수를 연산하는 것을 회피하는 것이다. 이러한 방식으로, 다수의 프로세싱 동작들은 룩업 동작에 의해 대체될 수 있다.

[0230] 예를 들어, 수학적식 (1)은 표 TAB1[diff]을 이용함으로써 수학적식 (3)을 제공하기 위하여 다음과 같이 단순화될 수 있다:

[0231]  $\alpha = (((y_B - y_A) * \text{TAB1}[\text{diff}] + \text{div} + \text{add}) \gg \text{pshift})$  (3)

[0232] 여기서,  $\text{TAB1}[\text{diff}] = \text{Floor}(2^k / \text{diff})$ .

[0233] 유사하게, 수학적식 (2)는 동일한 동작들을 반복시키는 것을 회피하기 위하여 미리 결정된 표들 TAB1[diff] 및 TAB2[diff]를 이용함으로써 단순화될 수 있다.

[0234]  $div = ((y_B - y_A) * (TAB2[diff] - TAB1[diff] * 2^k) + 2^{(k-1)}) >> k$  (4)

[0235] 여기서,  $TAB2[diff] = \text{Floor}((2^k * 2^k) / diff)$ .

[0236] 이 수학적식 (4)는 다음의 수학적식에 의해 추가로 단순화될 수 있다:

[0238] \*  $div = ((y_B - y_A) * (TAB3[diff]) + 2^{(k-1)}) >> k$  (5)

[0239] 여기서,

[0240]  $TAB3[diff] = TAB2[diff] - TAB1[diff]*2^k = \text{Floor}((2^k * 2^k) / diff) - 2^k * \text{Floor}(2^k / diff)$ .

[0241] TAB1 및 TAB3(그리고 또한, TAB2)은 N 엔트리들  $N=2^{(\text{BitDepth} - \text{pshift})}$  을 각각 가지는 표들이고, 각각의 엔트리는 k 비트들에 의해 표현된다.

[0242] 위의 정수 변수 "diff"의 정의에 따르면, 그리고 예를 들어, 10 비트들 상에서 표현된 루마 또는 크로마 샘플 값을 취하면, 최대 "diff" 값은 (위의 표 1a를 이용하여) 512이다. 이것은 표 TAB1 및 TAB3(그리고 또한, TAB2)이 이들의 각각에 대하여, 512 엔트리들을 갖는 어레이에 의해 표현될 수 있고 각각의 엔트리가 "k=16" 비트들 상에서 코딩된다는 것을 의미한다. 표 1에서 주어진 "pshift" 변수는 프로세싱되어야 할 샘플들의 비트 심도에 따라 동일한 수의 엔트리들(여기서, 512)을 획득하는 것을 가능하게 한다.

[0243] 이전에 언급된 바와 같이, 이 어레이들(TAB1 내지 TAB3)은 수학적식 (1) 및 (2)에서의 파라미터  $\alpha$  및  $\beta$ 의 유도를 위하여 수행되어야 할 동작들의 수를 감소시키기 위하여 메모리 내에 저장될 수 있다.

[0244] VVC 표준화 작업의 범위에서, 이러한 방법은 루마 샘플로부터 크로마 샘플을 예측하기 위하여 선형 모델의  $\alpha$  및  $\beta$  파라미터들을 추출하기 위한 계산을 구현하기 위하여 이용될 수 있다. 그러나, 놀랍게도, 표의 크기 및 각각의 엔트리의 표현은 코딩 효율에 불리하게 영향을 주지 않으면서 감소될 수 있다는 것이 발견되었다.

[0245] 위에서 논의된 바와 같이, 각각의 표를 저장하기 위하여 요구된 총 메모리는 "pshift" 값의 값에 종속되고, 각각의 엔트리를 표현하기 위한 비트들의 수는 "k" 값을 이용함으로써 코딩될 수 있다. 위에서 논의된 실시예에서는, 2개의 표들(TAB1 및 TAB3)이 이용되고, 각각의 표는 512 엔트리들을 가지고, k=16이다. 이 2개의 표들 TAB1 및 TAB3을 표현하기 위하여 필요한 메모리는: 2048 바이트들 상에서 저장될 수 있는  $2 * 512 * 16 = 16384$  비트들이다.

[0246] 메모리 요건들을 (독립적으로 그리고 공동으로 양자 모두) 결정하는 파라미터들을 수정하는 것이 지금부터 논의 될 것이다.

[0247] 각각의 표(어레이)에서의 엔트리들의 수

[0248] 512 엔트리들의 크기를 가지는 표를 이용함으로써 양호한 압축에 도달하는 것에도 불구하고, 이 표들은 상당히 큰 것으로서 고려될 수 있고, 그 크기가 감소되는 것이 바람직하다.

[0249] 표들 3 및 4는 엔트리의 수를 512로부터 256 및 128 엔트리들로 각각 감소시킴으로써 Bjontegard 메트릭들(예를 들어, rd-곡선들 사이의 평균 PSNR 차들의 Bjontegard. 계산 참조. 이 메트릭들이 어떻게 결정되는지의 설명을 위한 Doc. VCEG-M33 ITU-T Q6/16, April 2001. 79 Z. Xiong, A. Liveris, 및 S. Cheng)에 따른 코딩 효율 영향을 도시한다. 표의 크기에서의 감소는 'pshift'의 값을 1만큼 증분시킴으로써 수행된다( $N=2^{(\text{BitDepth} - \text{pshift})}$ ). 'add'의 값은 이에 따라 조절될 수 있다. N을 256 또는 128로 교대시키는 것이 이하의 표 2에서 도시된다:

비트-심도	256 엔트리들		128 엔트리들	
	“pShift”의 값	“add”의 값	“pShift”의 값	“add”의 값
7 이하	0	0	0	0
8 비트들	0	0	1	1
9 비트들	1	1	2	2
10 비트들	2	2	3	4
11 비트들	3	4	4	8
12 비트들	4	8	5	16
13 비트들	5	16	6	32
14 비트들	6	32	7	64
15 비트들	7	64	8	128
16 비트들	8	128	9	256

표 2: 256 또는 128 엔트리들의 표에 대한 "pshift" 및 "add" 값의

예

[0250]

이 표는 다음의 표현식들에 의해 표현될 수 있다:

[0251]

[0252]

256 엔트리들

[0253]

$$pShift = (BitDepth > 8) ? BitDepth - 8 : 0$$

[0254]

이것은 대안적으로, 다음의 표현식에 의해 표현될 수 있다:

[0255]

$$pShift = \begin{cases} BitDepth - 8, & \text{if } BitDepth > 8 \\ 0, & \text{그 외} \end{cases}$$

[0256]

임의적인 버림 값 "add"는 또한, 'diff'를 비트 시프팅을 따르는 정수로 만들기 위하여 연산될 수 있다. 'add' 값은 다음의 표현식에 따른 pshift에 관련된다:

[0257]

$$add = pshift ? 1 << (pshift - 1) : 0$$

[0258]

이것은 대안적으로, 다음의 표현식에 의해 표현될 수 있다:

[0259]

$$add = \begin{cases} 2^{(pshift-1)}, & \text{if } pshift > 0 \\ 0, & \text{그 외} \end{cases}$$

[0260]

대안적으로, 'BitDepth'와 'add' 사이의 직접적인 관계는 다음의 표현식들에 의해 제공될 수 있다:

[0261]

$$add = (BitDepth > 8) ? 1 << (BitDepth - 9) : 0$$

[0262]

또는:

[0263]

$$add = \begin{cases} 2^{(Bitdepth-9)} & \text{if } Bitdepth > 8 \\ 0 & \text{그 외} \end{cases}$$

[0264] 128 엔트리들

[0265]  $pShift = (BitDepth > 7) ? BitDepth - 7 : 0$

[0266] 이것은 대안적으로, 다음의 표현식에 의해 표현될 수 있다:

[0267] 
$$pShift = \begin{cases} BitDepth - 7, & \text{if } BitDepth > 7 \\ 0, & \text{그 외} \end{cases}$$

[0268] 임의적인 버림 값 "add"는 또한, 'diff'를 비트 시프팅을 따르는 정수로 만들기 위하여 연산될 수 있다. 'add' 값은 다음의 표현식에 따른 pshift에 관련된다:

[0269]  $add = pshift ? 1 << (pshift - 1) : 0 .$

[0270] 이것은 대안적으로, 다음의 표현식에 의해 표현될 수 있다:

[0271] 
$$add = \begin{cases} 2^{(pshift-1)}, & \text{if } pshift > 0 \\ 0, & \text{그 외} \end{cases}$$

[0272] 대안적으로, 'BitDepth'와 'add' 사이의 직접적인 관계는 다음의 표현식들에 의해 제공될 수 있다:

[0273]  $add = (BitDepth > 7) ? 1 << (BitDepth - 8) : 0$

[0274] 또는:

[0275] 
$$add = \begin{cases} 2^{(Bitdepth-8)} & \text{if } Bitdepth > 7 \\ 0 & \text{그 외} \end{cases}$$

[0276] 표의 크기의 감소는 샘플 값들의 최대값과 최소값 사이의 차의 더 대략적 표현으로 귀착된다.

[0277] 코딩 효율 평가 테스트는 문서 JVET-L1010에서 정의된 JVET 표준화 위원회에 의해 이용된 비디오 시퀀스들의 세트에 대해 수행되었다. 이하의 표에서, 음수 값들은 더 양호한 코딩 효율을 도시하는 반면, 양수 값들은 코딩 효율에서의 감소에 대응한다.

	모든 인트라 주요부 10		
	Y	U	V
클래스 A1	-0.03%	0.11%	0.10%
클래스 A2	0.00%	-0.02%	-0.02%
클래스 B	0.02%	0.01%	0.01%
클래스 C	0.03%	0.00%	0.00%
클래스 D	0.08%	-0.46%	-0.46%
클래스 E	0.01%	0.16%	0.16%
전체	0.01%	0.04%	0.13%

표 3: 크기 256의 표들을 이용할 때의 압축 성능

[0278]

[0279] 표 3에서 도시되는 바와 같이, 표 TAB1 및 TAB3에서의 엔트리의 수가 2 배만큼 감소되었음에도, 코딩 효율은 놀랍게도 필수적으로 영향받지 않는다. 우리는 CCLM 모드에 영향을 주는 수정에 의해 도입된 손실들이 매우 제한되고, 손실들이 크로마 채널들 (U) 및 (V)에서 0.2% 미만이고, 이것은 필수적으로 무시가능하다는 것(그리고 잡음을 표현할 가능성이 가장 많음)을 관찰할 수 있다.

[0280] (위의 표 2에서 도시된 바와 같이) 추가적인 1만큼 pshift의 값을 증분시킴으로써 생성된 128의 표 크기에 대한 유사한 실험이 행해졌다.

	모든 인트라 주요부 10		
	K16S128		
	Y	U	V
클래스 A1	0.03%	0.15%	0.08%
클래스 A2	0.02%	0.04%	0.04%
클래스 B	0.02%	-0.13%	-0.13%
클래스 C	0.03%	0.06%	0.06%
클래스 D	-0.01%	-0.19%	-0.19%
클래스 E	-0.01%	-0.06%	-0.06%
전체	0.02%	0.00%	0.03%

표 4: 크기 128의 표들을 이용할 때의 압축 성능

[0281]

[0282]

표 4에서 도시되는 바와 같이, 훨씬 더 놀랍게도, 표 TAB1 및 TAB3에서의 엔트리의 수가 4 배만큼 감소되었음에도, 코딩 효율은 여전히 필수적으로 영향받지 않는다. 우리는 CCLM 모드에 영향을 주는 수정에 의해 도입된 손실들이 매우 제한되고, 손실들이 크로마 채널들 (U) 및 (V)에서 0.05% 미만이고, 이것은 필수적으로 무시가능하다는 것(그리고 잡음을 표현할 가능성이 가장 많음)을 관찰할 수 있다.

[0283]

그러나, (pshift를 추가적인 1만큼 증분시킴으로써) 64 엔트리들로의 표의 크기의 추가의 감소는 이하의 표 5에서 제시된 바와 같은 압축 성능의 더 큰 손실로 귀착된다:

	모든 인트라 주요부 10		
	K16S64		
	Y	U	V
클래스 A1	0.14%	0.39%	0.36%
클래스 B	0.00%	0.10%	0.10%
클래스 C	-0.02%	0.07%	0.07%
클래스 D	0.04%	-0.16%	-0.16%
클래스 E	0.04%	0.07%	0.07%

[0284]

[0285]

이 결과들은 이들이 클래스 A2를 포함하지 않고, 이와 같이, "전체" 수치를 포함하지 않으므로 부분적이다.

[0286]

위에서 제시된 결과들은 CCLM 모드의 효율에 불리하게 영향을 미치지 않으면서, 표들의 크기가 2 또는 심지어 4 배만큼 감소될 수 있고, 여기서, 파라미터들  $\alpha$  및  $\beta$  는 2개의 포인트들 A 및 B를 이용하여 유도된다는 것을 보여준다.

[0287]

또 다른 실시예에서, 표에서의 엔트리들의 수(즉, pshift의 값)는 비트 심도에 따라 변동될 수 있다(예컨대, 최대 10 비트들에 대하여 128(또는 256) 및 10 초과 비트들에 대하여 256(또는 512)). 이것은 (예를 들어) 16 비트들에 의해 표현된 샘플들을 인코딩하기 위하여 더 강력한 인코더가 요구될 것이고, 이와 같이, 더 큰 표를 이용하는 계산의 복잡도는 그렇게 큰 쟁점을 제시하지 않을 것이라는 사실에 기인할 수 있다. 이러한 경우에, 더 큰 수(예컨대, 512 이상)의 엔트리들을 이용함으로써 인코딩 성능에서의 근소한 증가가 우선순위화될 수 있다.

[0288]

표들(어레이들)에서의 각각의 엔트리를 표현하는 비트들의 수

[0289]

표의 크기를 추가로 감소시키기 위하여, 표 내의 각각의 엔트리는 엔트리 당 2 바이트들을 이용하는 초기 k=16 비트들 미만에 대해 또한 표현될 수 있다. k의 값을 감소시키는 것은 그것이 정수들로  $\alpha$ 를 표현하기 위한 크기를 감소시키는 것에 필수적으로 대응하므로, 계산의 정확도를 감소시키는 것을 표현한다. 이하의 표 6은 (k=16과 비교하여) 각각의 엔트리를 표현하기 위한 비트들의 수를 감소시킬 때의 코딩 효율의 영향을 도시한다.

모든 인트라 주요부 10						
k=14			k=12			
Y	U	V	Y	U	V	
클래스 A1	-0.02%	0.15%	0.08%	-0.02%	0.15%	0.08%
클래스 A2	0.00%	0.02%	0.02%	0.00%	0.02%	0.02%
클래스 B	0.01%	-0.18%	-0.18%	0.01%	-0.18%	-0.18%
클래스 C	-0.02%	0.01%	0.01%	-0.02%	0.01%	0.01%
클래스 D	0.02%	-0.24%	-0.24%	0.02%	-0.24%	-0.24%
클래스 E	0.04%	-0.14%	-0.14%	0.04%	-0.14%	-0.14%
<b>전체</b>	<b>0.00%</b>	<b>-0.04%</b>	<b>0.05%</b>	<b>0.00%</b>	<b>-0.04%</b>	<b>0.05%</b>
k=10			k=8			
Y	U	V	Y	U	V	
클래스 A1	-0.02%	0.11%	0.08%	0.02%	0.25%	0.05%
클래스 A2	0.01%	-0.10%	-0.10%	0.02%	-0.02%	-0.02%
클래스 B	0.00%	-0.03%	-0.03%	0.03%	0.04%	0.04%
클래스 C	-0.03%	0.15%	0.15%	0.00%	-0.16%	-0.16%
클래스 D	0.04%	-0.54%	-0.54%	0.07%	0.08%	0.08%
클래스 E	0.02%	0.06%	0.06%	0.01%	-0.20%	-0.20%
<b>전체</b>	<b>0.00%</b>	<b>0.04%</b>	<b>0.16%</b>	<b>0.01%</b>	<b>-0.02%</b>	<b>0.07%</b>
k=7			k=6			
Y	U	V	Y	U	V	
클래스 A1	0.02%	0.33%	0.05%	0.05%	0.80%	0.31%
클래스 A2	0.02%	0.07%	0.07%	0.03%	0.21%	0.21%
클래스 B	0.03%	-0.09%	-0.09%	0.05%	0.03%	0.03%
클래스 C	-0.03%	0.09%	0.09%	0.00%	0.22%	0.22%
클래스 D	0.01%	0.05%	0.05%	-0.01%	-0.04%	-0.04%
클래스 E	0.01%	0.09%	0.09%	0.02%	0.31%	0.31%
<b>전체</b>	<b>0.01%</b>	<b>0.08%</b>	<b>0.07%</b>	<b>0.03%</b>	<b>0.28%</b>	<b>0.28%</b>
k=5			k=4			
Y	U	V	Y	U	V	
클래스 A1	0.30%	2.13%	1.04%	0.89%	6.18%	3.31%
클래스 A2	0.09%	0.83%	0.83%	0.30%	2.73%	2.73%
클래스 B	0.11%	0.31%	0.31%	0.33%	1.60%	1.60%
클래스 C	0.08%	0.75%	0.75%	0.40%	2.78%	2.78%
클래스 D	0.07%	-0.10%	-0.10%	0.26%	1.49%	1.49%
클래스 E	0.07%	0.50%	0.50%	0.15%	1.83%	1.83%
<b>전체</b>	<b>0.13%</b>	<b>0.83%</b>	<b>0.70%</b>	<b>0.40%</b>	<b>2.85%</b>	<b>2.47%</b>

[0290]

k=3			
Y	U	V	
클래스 A1	2.46%	17.76%	10.99%
클래스 A2	1.06%	8.33%	8.33%
클래스 B	0.78%	6.86%	6.86%
클래스 C	1.27%	9.95%	9.95%
클래스 D	0.72%	5.90%	5.90%
클래스 E	0.33%	3.66%	3.66%
<b>전체</b>	<b>1.14%</b>	<b>9.07%</b>	<b>8.35%</b>

[0291]

표 6: k 값을 감소시킬 때의 코딩 성능

[0292]

이 위의 표들 6은 엔트리들이 8 비트들 이하 상에서 표현될 때, 코딩 효율이 k=16 비트들과 비교하여 필수적으

로 동일하다는 놀라운 결과를 도시한다.

- [0293] 위의 표들은 수용가능한 코딩 효율 결과들이 6 내지 8의 간격에서 포함되는 k에 대하여 획득된다는 것을 도시한다. 유사한 결과는 간격 9 내지 15인 k에 대하여 달성되지만, 이러한 표현은 2 바이트들을 여전히 요구할 것이므로, 표를 저장하기 위하여 요구된 메모리를 감소시킬 시에 큰 장점을 제공하지 않을 것이다.
- [0294] 표들 6은 5 비트들 이하와 동일한 k에 대하여, 알파(alpha)를 계산할 시의 계산이 부정확하게 되는 것에 기인하여 더 큰 열화(degradation)가 관찰된다는 것을 도시한다.
- [0295] 이에 따라, 성능과 저장 사이의 최상의 절충은 k=8 비트들일 경우라는 것이 놀랍게도 발견되었다.
- [0296] k=16 비트들과 비교하면, 각각의 엔트리가 2 바이트들 상에서 표현될 경우에, 이 현재의 실시예는 표 TAB1 또는 TAB3의 엔트리를 표현하기 위하여 단일 바이트를 오직 이용할 수 있다. 이것은 k를 수반하는 모든 연산들의 복잡도를 감소시키고, 이와 같이, 프로세싱 수요를 감소시킨다.
- [0297] k=6, 7, 또는 8의 값은 놀랍게도, k=16과 유사한 코딩 성능을 제공하고, k의 더 높은 값들은 근소하게 더 양호한 코딩 성능을 제공한다.
- [0298] 특히 놀랍게도, 코딩 성능에서의 임의의 인식가능한 감소를 보이지 않으면서, 전체 바이트(8 비트들)만큼 k를 감소시키는 것이 가능하다.
- [0299] 또한, k가 k=6만큼 낮게 감소될 수 있고, 이것을 넘는 것만이 검출된 코딩 성능에서의 인식가능한 감소라는 것을 발견하는 것이 놀랍다.
- [0300] k를 16으로부터 8로 감소시킴으로써, 각각의 표를 위하여 이용된 총 메모리는 2 배만큼 감소된다.
- [0301] 이것은 전형적으로, 비트-시프트들을 수반하는 동작들의 정확도가 비트-시프트 파라미터의 값에 대한 작은 변경들에 의해 강력하게 영향받기 때문에 놀랍고, 코딩 성능에서의 큰 열화는 심지어 작은 양만큼 k의 값을 감소시킬 때에 예상될 것이다. 그러나, 위의 결과들은 (예컨대, 16으로부터 6 아래로의) k의 값에서의 큰 변경이 무시가능한 양(<0.14%)만큼 코딩 성능을 오직 감소시킨다는 것을 반직관적으로 보여준다.
- [0302] k의 선택은 비트 심도에 따라 변동될 수 있다(예컨대, 최대 10 비트들에 대하여 8 및 10 초과 비트들에 대하여 16). 이것은 (예를 들어) 16 비트들에 의해 표현된 샘플들을 인코딩하기 위하여 더 강력한 인코더가 요구될 것이고, 이와 같이, 더 큰 표를 이용하는 계산의 복잡도는 그렇게 큰 장점을 제시하지 않을 것이라는 사실에 기인할 수 있다. 이러한 경우에, 더 큰 수(예컨대, 8 초과)의 비트들을 이용함으로써 인코딩 성능에서의 근소한 증가가 우선순위화될 수 있다.
- [0303] 엔트리들의 수 및 각각의 엔트리를 표현하는 비트들의 수의 조합
- [0304] 이하의 표 7은 k=16 및 각각의 표에서의 512 엔트리들의 기준선과 비교하여, 각각의 표에서 256 엔트리들이 있을 때(즉, 위의 표들 3 및 6에서 제시된 결과들의 2개의 세트들의 조합) k를 8로부터 5로 변동시키기 위한 코딩 성능에 대한 결과들을 도시한다.

모든 인트라 주요부 10						
k=16			k=14			
Y	U	V	Y	U	V	
클래스 A1	-0.03%	0.11%	0.10%	0.01%	0.23%	0.16%
클래스 A2	0.00%	-0.02%	-0.02%	-0.02%	0.08%	0.08%
클래스 B	0.02%	0.01%	0.01%	0.01%	0.04%	0.04%
클래스 C	0.03%	0.00%	0.00%	-0.02%	0.18%	0.18%
클래스 D	0.08%	-0.46%	-0.46%	0.00%	-0.23%	-0.23%
클래스 E	0.01%	0.16%	0.16%	0.00%	0.04%	0.04%
<b>전체</b>	<b>0.01%</b>	<b>0.04%</b>	<b>0.13%</b>	<b>0.00%</b>	<b>0.11%</b>	<b>0.11%</b>
k=12			k=10			
Y	U	V	Y	U	V	
클래스 A1	-0.01%	0.13%	-0.02%	-0.04%	-0.02%	0.08%
클래스 A2	0.01%	-0.13%	-0.13%	0.03%	-0.12%	-0.12%
클래스 B	-0.01%	0.08%	0.08%	-0.01%	0.01%	0.01%
클래스 C	-0.03%	0.15%	0.15%	-0.02%	-0.01%	-0.01%
클래스 D	0.04%	-0.13%	-0.13%	0.03%	0.01%	0.01%
클래스 E	-0.04%	0.11%	0.11%	0.00%	-0.24%	-0.24%
<b>전체</b>	<b>-0.01%</b>	<b>0.07%</b>	<b>0.13%</b>	<b>-0.01%</b>	<b>-0.06%</b>	<b>0.20%</b>
k=8			k=7			
Y	U	V	Y	U	V	
클래스 A1	0.00%	0.02%	-0.06%	0.01%	0.11%	0.01%
클래스 A2	0.02%	-0.05%	-0.05%	0.02%	-0.01%	-0.01%
클래스 B	0.03%	-0.04%	-0.04%	0.01%	-0.03%	-0.03%
클래스 C	0.00%	0.13%	0.13%	0.01%	-0.12%	-0.12%
클래스 D	0.01%	-0.12%	-0.12%	0.02%	0.04%	0.04%
클래스 E	0.00%	0.06%	0.06%	-0.03%	0.37%	0.37%
<b>전체</b>	<b>0.01%</b>	<b>0.02%</b>	<b>0.10%</b>	<b>0.01%</b>	<b>0.04%</b>	<b>0.03%</b>
k=6			k=5			
Y	U	V	Y	U	V	
클래스 A1	0.03%	0.23%	0.08%	0.12%	0.80%	0.43%
클래스 A2	0.02%	0.02%	0.02%	0.06%	0.24%	0.24%
클래스 B	0.03%	-0.06%	-0.06%	0.06%	0.00%	0.00%
클래스 C	0.01%	0.17%	0.17%	0.02%	0.10%	0.10%
클래스 D	0.06%	-0.23%	-0.23%	0.06%	0.06%	0.06%
클래스 E	-0.01%	0.17%	0.17%	0.01%	0.20%	0.20%
<b>전체</b>	<b>0.02%</b>	<b>0.09%</b>	<b>0.14%</b>	<b>0.05%</b>	<b>0.23%</b>	<b>0.26%</b>

[0305]

k=4			k=3			
Y	U	V	Y	U	V	
클래스 A1	0.46%	2.62%	1.67%	1.36%	8.47%	5.60%
클래스 A2	0.16%	1.33%	1.33%	0.52%	4.45%	4.45%
클래스 B	0.17%	0.74%	0.74%	0.54%	3.53%	3.53%
클래스 C	0.18%	1.05%	1.05%	0.76%	5.21%	5.21%
클래스 D	0.16%	0.96%	0.96%	0.49%	3.29%	3.29%
클래스 E	0.13%	1.13%	1.13%	0.24%	2.85%	2.85%
<b>전체</b>	<b>0.21%</b>	<b>1.29%</b>	<b>1.21%</b>	<b>0.67%</b>	<b>4.77%</b>	<b>4.60%</b>

표 7: k 값 및 엔트리들의 수를 512 로부터 256 으로 감소시킬 때의 코딩 성능

[0306]

[0307] 위의 코딩 효율 결과들은 단일 바이트 각각(즉,  $k \leq 8$ )을 이용하여 인코딩된 256 엔트리들을 가지는 2개의 표들(TAB1 및 TAB3)을 이용함으로써, 2 바이트들 상에서 표현된 512 엔트리들을 갖는 2개의 표들의 경우와 비교하여 유사한 결과들이 획득될 수 있다는 놀라운 결과를 보여준다. 이 실시예의 특정한 장점은 코딩 효율 결과에 영향을 주지 않으면서 이 표들 TAB1 및 TAB3을 저장하기 위하여 필요한 메모리의 4 배만큼의 감소이다. 이 특정한 실시예에서, 2개의 표들(TAB1 및 TAB3)을 표현하기 위하여 필요한 메모리는: 512 바이트들 상에서 저장될 수 있는  $2 * 256 * 8 = 4,092$  비트들이다.

[0308] 이하의 표 8은  $k=16$  및 각각의 표에서의 512 엔트리들의 기준선과 비교하여, 각각의 표에서 128 엔트리들이 있을 때(즉, 위의 표들 4 및 6에서 제시된 결과들의 2개의 세트들의 조합)  $k$ 를 8로부터 5로 변동시키기 위한 코딩 성능에 대한 결과들을 도시한다.

모든 인트라 주요부 10						
	k=16			k=14		
	Y	U	V	Y	U	V
클래스 A1	0.03%	0.15%	0.08%	0.01%	0.25%	0.15%
클래스 A2	0.02%	0.04%	0.04%	0.02%	-0.04%	-0.04%
클래스 B	0.02%	-0.13%	-0.13%	0.01%	-0.12%	-0.12%
클래스 C	0.03%	0.06%	0.06%	-0.03%	0.22%	0.22%
클래스 D	-0.01%	-0.19%	-0.19%	0.05%	0.02%	0.02%
클래스 E	-0.01%	-0.06%	-0.06%	-0.01%	0.04%	0.04%
<b>전체</b>	<b>0.02%</b>	<b>0.00%</b>	<b>0.03%</b>	<b>0.00%</b>	<b>0.06%</b>	<b>0.16%</b>
	k=12			k=10		
	Y	U	V	Y	U	V
클래스 A1	0.05%	-0.04%	-0.06%	0.00%	0.05%	0.20%
클래스 A2	0.01%	0.01%	0.01%	0.02%	-0.17%	-0.17%
클래스 B	0.00%	0.15%	0.15%	0.02%	-0.06%	-0.06%
클래스 C	-0.02%	0.11%	0.11%	0.00%	0.02%	0.02%
클래스 D	0.10%	-0.58%	-0.58%	0.05%	-0.29%	-0.29%
클래스 E	-0.04%	0.07%	0.07%	0.01%	-0.05%	-0.05%
<b>전체</b>	<b>0.00%</b>	<b>0.07%</b>	<b>0.09%</b>	<b>0.01%</b>	<b>-0.04%</b>	<b>0.13%</b>
	k=8			k=7		
	Y	U	V	Y	U	V
클래스 A1	0.03%	0.02%	0.04%	0.02%	0.07%	0.01%
클래스 A2	0.01%	-0.13%	-0.13%	0.01%	-0.01%	-0.01%
클래스 B	0.01%	-0.09%	-0.09%	0.01%	0.02%	0.02%
클래스 C	0.01%	-0.17%	-0.17%	-0.01%	0.11%	0.11%
클래스 D	0.01%	-0.25%	-0.25%	0.01%	0.08%	0.08%
클래스 E	0.00%	-0.20%	-0.20%	0.00%	-0.09%	-0.09%
<b>전체</b>	<b>0.01%</b>	<b>-0.11%</b>	<b>0.09%</b>	<b>0.01%</b>	<b>0.03%</b>	<b>0.06%</b>
	k=6			k=5		
	Y	U	V	Y	U	V
클래스 A1	0.04%	0.27%	0.17%	0.10%	0.35%	0.34%
클래스 A2	0.04%	-0.07%	-0.07%	0.03%	0.12%	0.12%
클래스 B	0.02%	-0.02%	-0.02%	0.04%	-0.03%	-0.03%
클래스 C	0.01%	0.19%	0.19%	0.03%	0.20%	0.20%
클래스 D	0.07%	0.22%	0.22%	0.03%	-0.04%	-0.04%
클래스 E	0.03%	-0.04%	-0.04%	-0.01%	0.14%	0.14%
<b>전체</b>	<b>0.03%</b>	<b>0.06%</b>	<b>0.17%</b>	<b>0.04%</b>	<b>0.14%</b>	<b>0.21%</b>

[0309]

	k=4			k=3		
	Y	U	V	Y	U	V
클래스 A1	0.27%	1.41%	0.98%	0.82%	4.99%	3.28%
클래스 A2	0.09%	0.84%	0.84%	0.30%	2.84%	2.84%
클래스 B	0.11%	0.18%	0.18%	0.31%	1.89%	1.89%
클래스 C	0.08%	0.60%	0.60%	0.38%	2.42%	2.42%
클래스 D	0.13%	0.33%	0.33%	0.26%	1.51%	1.51%
클래스 E	0.12%	0.47%	0.47%	0.20%	2.07%	2.07%
전체	<b>0.13%</b>	<b>0.64%</b>	<b>0.59%</b>	<b>0.39%</b>	<b>2.71%</b>	<b>2.78%</b>

표 8: k 값 및 엔트리들의 수를 512로부터 128로 감소시킬 때의 코딩

성능

[0310]

[0311]

위의 코딩 효율 결과들은 단일 바이트 각각(즉,  $k \leq 8$ )을 이용하여 인코딩된 128 엔트리들을 가지는 2개의 표들(TAB1 및 TAB3)을 이용함으로써, 2 바이트들 상에서 표현된 512 엔트리들을 갖는 2개의 표들의 경우와 비교하여 유사한 결과들이 획득될 수 있다는 놀라운 결과를 보여준다. 다수의 예들에서, 128 엔트리들을 이용하는 것은 256 엔트리들을 이용하는 것과 비교하여 코딩 성능을 실제적으로 개선시킨다는 것이 특히 놀랍다. 예를 들어,  $k=8$ (엔트리 당 1 바이트)에 대하여, 결과들은 128의 표 크기가 256 엔트리들의 표와 비교하여 개선된 코딩 성능으로 귀착된다는 것을 보여준다.

[0312]

이 실시예의 특정한 장점은 코딩 효율 결과에 영향을 주지 않으면서 이 표들 TAB1 및 TAB3을 저장하기 위하여 필요한 메모리의 8 배만큼의 감소이다. 이 특정한 실시예에서, 2개의 표들(TAB1 및 TAB3)을 표현하기 위하여 필요한 메모리는: 256 바이트들 상에서 저장될 수 있는  $2 * 128 * 8 = 2046$  비트들이다.

[0313]

CCLM 모드는 이에 따라, 효율적인 하드웨어 구현을 위한 정수 산술을 통해 구현될 수 있는  $\alpha$  및  $\beta$  파라미터들을 추출하기 위한 이 계산 방법을 이용할 수 있다.

[0314]

특히, 표에서의 엔트리들의 수를 감소시키는 것 뿐만 아니라, 각각의 엔트리의 크기를 감소시키는 것의 조합은(예상될 수 있는 바와 같이) 성능에서의 감소의 혼합이 아니라, 오히려, 어느 하나를 독립적으로 행하는 것과 비교하여, 표에서의 엔트리들의 수에서의 감소 및 각각의 엔트리의 크기에서의 감소를 조합할 때에 필수적으로 동일한 성능이 달성된다는 것이 밝혀졌다.

[0315]

완전함을 위하여, 표 9는  $k=16$  및  $N=512$ 의 기준선과 비교하여,  $N=64$  엔트리들의 표가 이용될 때의 부분적인 결과들을 도시한다. 성능의 손실은 U 및 V 컴포넌트들에서의 (VVC에 대한 주요 목적인) 클래스 A1에 대하여 상당하다는 것이 주목되어야 한다:

모든 인트라 주요부 10						
k=16			k=14			
	Y	U	V	Y	U	V
클래스 A1	0.14%	0.39%	0.36%	0.11%	0.52%	0.49%
클래스 B	0.00%	0.10%	0.10%	0.02%	0.11%	0.11%
클래스 C	-0.02%	0.07%	0.07%	0.03%	0.08%	0.08%
클래스 D	0.04%	-0.16%	-0.16%	0.05%	-0.57%	-0.57%
클래스 E	0.04%	0.07%	0.07%	0.04%	-0.19%	-0.19%
k=12			k=10			
	Y	U	V	Y	U	V
클래스 A1	0.13%	0.49%	0.56%	0.11%	0.39%	0.33%
클래스 B	0.04%	0.09%	0.09%	0.02%	-0.06%	-0.06%
클래스 C	0.00%	-0.04%	-0.04%	-0.04%	0.19%	0.19%
클래스 D	-0.01%	0.07%	0.07%	0.03%	-0.32%	-0.32%
클래스 E	0.01%	-0.14%	-0.14%	0.04%	-0.08%	-0.08%
k=8			k=7			
	Y	U	V	Y	U	V
클래스 A1	0.10%	0.43%	0.34%	0.14%	0.57%	0.42%
클래스 B	0.03%	0.04%	0.04%	0.00%	-0.02%	-0.02%
클래스 C	-0.05%	0.27%	0.27%	-0.04%	0.19%	0.19%
클래스 D	0.07%	-0.64%	-0.64%	-0.04%	0.10%	0.10%
클래스 E	0.06%	-0.37%	-0.37%	0.03%	0.22%	0.22%
k=6			k=5			
	Y	U	V	Y	U	V
클래스 A1	0.13%	0.39%	0.56%	0.14%	0.74%	0.35%
클래스 B	0.04%	0.10%	0.10%	0.03%	-0.04%	-0.04%
클래스 C	0.00%	0.05%	0.05%	0.00%	0.08%	0.08%
클래스 D	0.01%	-0.14%	-0.14%	0.03%	-0.01%	-0.01%
클래스 E	0.05%	-0.05%	-0.05%	0.07%	0.15%	0.15%
k=4			k=3			
	Y	U	V	Y	U	V
클래스 A1	0.29%	1.20%	0.87%	0.63%	4.01%	2.39%
클래스 B	0.03%	0.22%	0.22%	0.23%	1.43%	1.43%
클래스 C	0.04%	0.19%	0.19%	0.25%	1.39%	1.39%
클래스 D	0.18%	-0.40%	-0.40%	0.18%	1.50%	1.50%
클래스 E	0.06%	0.70%	0.70%	0.16%	1.66%	1.66%

표 9: k 값 및 엔트리들의 수를 512 로부터 64 로 감소시킬 때의 부분적인 코딩 성능 결과들

[0316]

[0317]  $\alpha$ 의 표현

[0318] 또 다른 실시예에서,  $\alpha$  파라미터의 값은 그것이 "L" 비트들 상에서 표현될 수 있도록 수정된다. 위에서 설명된  $\alpha$  및  $\beta$ 의 유도 프로세스로,  $\alpha$ 의 값은 k가 8과 동일할 대에 최대 17 비트들에 도달할 수 있다(그리고 k=16 일 경우에 최대 25 비트들).  $\alpha$ 의 값을 수정하기 위한 일차적인 이유는 이하의 예측 공식에서의 승산의 비트 폭을 제한하기 위한 것이다:

[0319] 
$$C_c = ((\alpha * L_c) \gg k) + \beta$$

[0320] 여기서,  $C_c$ 는 루마  $L_c$  값에 대응하는 크로마 예측된 값이고,  $\alpha$  및  $\beta$ 는 (위에서 설명된 바와 같이 유도될 수 있는) 기울기의 파라미터들이다.

[0321] 루마 샘플들이 10 비트들 상에서 코딩될 경우에, 그것은 예측의 핵심 루프가 연산적으로 복잡한(그리고 메모리의 많은 양들을 사용할 수 있는) 최대 10 비트들 x 17 비트들의 승산을 취급할 수 있도록 요구된다는 것을 의미

한다. 이 실시예에서,  $\alpha$ 의 값은 승산이 16 비트들을 초과하지 않도록 수정된다. 이러한 계산은 하드웨어 구현에 양호하게 적합하고, 예를 들어, 16-비트 프로세서는 단일 메모리 레지스터를 이용하여 계산을 착수할 수 있다.

[0322] 이것을 달성하기 위하여,  $\alpha$ 는 6비트 이하로 표현될 필요가 있다. 이 6-비트 표현에 도달하기 위하여,  $\alpha$ 의 범위는 더 큰 값들이 특정된 범위로 강제되도록 '클리핑'될 수 있다.

[0323] 또 다른 실시예에서,  $\alpha$ 의 크기는 그것을 적절한 양에 의해 제한함으로써, 그리고 이에 따라 시프트 파라미터 k를 조절함으로써 크기가 감소되도록 결정된다. 시프트 값(k)에 대한 조절의 양을 결정하는 것은 "최상위 비트(most significant bit)" 포지션 P를 발견하는 것(예컨대, 선두 0들을 카운팅하거나 베이스-2 로그를 취함으로써 수행된 고전적인 동작)을 포함한다. 이 경우에, 그것이 한계 L(바람직한 실시예에서 5 또는 6)을 초과할 경우에, 다음의 동작들이 수행된다:

$$\alpha = \alpha \gg (P-L)$$

[0324]  $k = k + L - P$

[0325] 즉,  $\alpha$ 는 인자  $2^{(P-L)}$ 에 의해 제한되고, k의 값은 반대의 양(L-P)에 의해 보상된다.

[0326] L의 값은 비트심도에 종속될 수 있지만, 구현들을 단순화하기 위하여 몇몇 비트심도들에 대하여 동일할 수 있다. 추가적으로, L은 부호 비트를 참작할 수 있고, 즉, L=5이다.

[0327] 대부분의 구현의 경우들에는,  $\alpha$ 의 부동 소수점 값이 CCLM 모드에서 범위 [-2.0; 2.0] 내에 있다는 것이 또한 관찰되었다. 오직 6 비트들의 사용은 예를 들어, 간격 [-2.0; 2.0]에서 64개의 값들을 표현할 수 있고, 여기서, 이 범위 외부에 속하는 임의의 계산된 값들은 범위의 중점에 의해 대체될 수 있다.

[0328] 어떤 경우에도, 임의의 범위의  $\alpha$ 의 클리핑은 그 크기를 감소시키기 이전에 수행될 수 있다. 이것은 외부에 놓이는 값들이 크기 감소 프로세싱을 착수하기 이전에 제거된다는 것을 보장한다.

[0329] 도 11은 발명의 실시예들에서 2개의 포인트들(A 및 B)을 선택하는 상이한 방식들을 예시한다.

[0330] 유도의 제안된 단순화는 코딩 효율에 대한 영향을 가진다. 코딩 효율 손실을 감소시키기 위하여, 2개의 포인트들의 사려깊은 선택은 매우 중요한 단계이다.

[0331] 이전에 설명된 바와 같은 제1 실시예에서, 이웃하는 루마 샘플 값들의 최소 및 최대는 도 11의 포인트들 A 및 B에 대응하여 선택된다.

[0332] 대안적인 실시예에서, 2개의 선택된 포인트들은 이웃하는 크로마 샘플 값들의 최소 및 최대에 대응하는 루마 및 크로마 샘플들의 쌍에 대응하는 도 11의 포인트들 C 및 D이다. 이 대안적인 실시예는 코딩 효율의 관점에서 때때로 흥미롭다.

[0333] 대안적인 실시예에서, 세그먼트들 [AB] 및 [CD] 사이의 가장 긴 세그먼트가 결정되고, [AB] 세그먼트가 [CD] 세그먼트보다 더 길 경우에, 포인트들 A 및 B가 선택되고, 그렇지 않을 경우에는 포인트들 C 및 D가 선택된다. 각각의 세그먼트의 길이는 유클리드 거리(Euclidian distance)로 연산될 수 있다. 하지만 또 다른 거리 척도가 이용될 수 있다. 이 실시예는 2개의 첫 번째 것들과 비교하여 코딩 효율을 개선시킨다. 실제로, 2개의 선택된 포인트들이 떨어져 있을 때, 일반적으로, 생성된 선형 모델이 관련된다. 결과적으로, 생성된 크로마 블록 예측자는 현재의 블록을 예측하기 위하여 관련된다.

[0334] 대안적인 실시예에서, A, B, C, D 사이에서 생성될 수 있는 모든 가능한 세그먼트들 중에서 가장 긴 세그먼트는 2개의 선택된 포인트들을 제공한다. 이것은 도 11에서 도시된 바와 같은 세그먼트들 [AB], [CD], [AC], [AD], [CB], 및 [DB]에 대응한다. 이 실시예는 더 높은 복잡도를 희생하면서, 이전의 것들과 비교하여 코딩 효율을 개선시킨다.

[0335] 바람직한 실시예에서, RecL' 루마 샘플 값들의 최소 및 최대를 표현하는 포인트들은 A 및 B 포인트들을 생성하기 위하여 설정되고, 포인트 A의 하나의 컴포넌트가 B로부터의 그 대응하는 컴포넌트와 동일할 경우에 ( $x_B = x_A$  또는  $y_B = y_A$ ), 크로마 샘플 값들 C 및 D의 최소 및 최대를 표현하는 포인트들이 선택된다. 이 실시예는

$x_B = x_A$  또는  $y_B = y_A$  일 경우에,  $\alpha$  (또는 L)가 각각 무한대이거나 0과 동일하고, 결과적으로, 크로마 예측자 블록은 각각 이용불가능하거나 DC 예측과 동등하기 때문에, 최상의 코딩 효율을 획득한다. 이것은  $\alpha$  (또는 L)를 나타내는 분수의 분자 또는 분모의 어느 하나가 너무 낮을 시에 즉시 그러하고(예를 들어, 다음의 조건이 검증될 수 있음:  $|\alpha| < 0.1$ ): 심지어 작은 양만큼 (양자화로 인한 것과 같은) 그것에 대한 임의의 오차는  $\alpha$  (또는 L)의 매우 상이한 값들을 초래한다. 문서의 나머지에서, 기본적으로 거의 수평 또는 수직 기울기들인 이러한 경우들은 비정상적인 기울기가  $\alpha$  또는 L이든지 간에, 비정상적인 기울기로서 지칭되는 것을 초래한다.

[0336] 추가적인 실시예에서, 도 11에서 모두 도시된 바와 같은 포인트들의 몇몇 쌍들은  $\alpha$ 가 "비정상적"이 아닐 때까지 테스트된다. 이 실시예는 이전의 것의 코딩 효율을 개선시키지만, 그것은 연산 복잡도를 증가시킨다.

[0337] 하나의 대안적인 실시예에서는, 2개의 컴포넌트들(크로마 및 루마)의 최대 및 최소 사이의 차들이 연산된다. 추가적으로, 최대 차를 갖는 컴포넌트는 모델 파라미터들의 연산을 위한 라인을 정의하는 2개의 포인트들을 결정하기 위하여 선택된다. 이 실시예는 2개의 컴포넌트들이 2개의 크로마 컴포넌트들 또는 2개의 RGB 컴포넌트들일 때에 효율적이다.

[0338] 2개의 포인트들 A 및 B의 선택은 현재의 블록의 샘플들의 값들에 대해 행해질 수 있다. 하나의 실시예에서, 단순화된 선형 모델 유도를 위한 2개의 포인트들은 현재의 다운샘플링된 루마 블록(도 5에서의 505)의 샘플 값들에 기초하여 설정된다. 블록의 이웃에서의 샘플 쌍들의 루마 샘플들 값들은 루마 블록의 루마 샘플 값들과 비교된다. 최대 빈도를 갖는 값은  $x_A$ 를 생성하기 위하여 선택되고, 최대 빈도를 갖는 제2 값은  $x_B$ 를 생성하기 위하여 선택된다. 대응하는 크로마 값들  $y_A$  및  $y_B$ 는 블록의 이웃에서의 샘플 쌍들에서의 공동위치된 크로마 샘플들의 평균 값들이다.  $\alpha$  (또는 L)가 "비정상적"(0과 동일하거나 0에 근접함( $|\alpha| < 0.1$ ))일 때,  $x_B$ 는 제2 가장 많이 선택된 값 대신에, 더 적은 선택을 갖는 루마 값들 중의 하나이다. 동일한 방식으로,  $y_B$ 는 공동위치된 크로마 샘플들의 평균 값들이다. 이 실시예는 최고 복잡도를 희생하면서, 이전의 실시예와 비교하여 코딩 효율을 개선시킨다.

[0339] 2개의 포인트들 A 및 B의 선택은 샘플 쌍들의 공간적 포지션들에 대해 행해질 수 있다.

[0340] 이전의 실시예들에서는, 루마 및 크로마 이웃하는 샘플들의 M 개의 쌍들 중에서 루마(A, B) 또는/및 크로마(C, D)에 대한 최소 및 최대 값들을 결정하는 것이 필요하다. 이것은 추가적인 복잡도로서 고려될 수 있다. 그러므로, 일부 구현들을 위하여, 최소 복잡도를 갖는 이 2개의 포인트들을 획득하는 것이 바람직하다.

[0341] 하나의 실시예에서, 하나의 선형 모델은 크로마 샘플들 RecC(도 5 상의 501)로, 그리고 경계 RecL'(503)의 다운샘플링된 루마 샘플들로 생성된다. 선택된 제1 포인트는 루마의 5004로 참조된, 좌측 행의 하부 샘플, 및 공동위치된 크로마 샘플(5001)이다. 선택된 제2 포인트는 상부 우측 루마 샘플(5003) 및 공동위치된 크로마 샘플(5002)이다. 2개의 포인트들의 이 선택은 매우 간단하지만, 그것은 또한, 값들에 기초한 이전의 실시예들보다 덜 효율적이다.

[0342] 추가적으로, 상부 또는 좌측 에지 중의 하나가 예를 들어, 이미지 또는 슬라이스의 경계 상의 블록에 대하여 존재하지 않거나, 예를 들어, 복잡도 또는 오차 탄력성(error resilience)의 이유로 이용불가능할 경우에, 결손된 것 대신에, 2개의 샘플들(예컨대, 그 루마가 이용가능한 에지 상에서 504 또는 5003인 것들)이 선택된다. 이에 따라, 몇몇 조건들이 샘플들을 선택하기 위하여 존재한다는 것을 알 수 있다.

[0343] 기울기를 연산하기 위하여 충분한 포인트들이 선택될 수 없거나, 이들이 "비정상적"인  $\alpha$  (또는 L)로 귀착될 경우에, 디폴트 포인트가 그 대신에 선택될 수 있는 추가의 실시예가 설명된다. 이 실시예는 또한, 적응(adaptation)을 갖는 MMLM 모드에 대하여 적용될 수 있다. 제1 그룹에 대한 선형 모델 파라미터들을 생성하기 위하여, 제1 포인트는 루마의 제2 좌측 행(5009)의 하부 샘플 및 공동위치된 크로마 샘플(5005)이다. 또한, 제2 포인트는 제1 좌측 행(5010)의 상부 루마 샘플 및 공동위치된 크로마 샘플(5006)이다.

[0344] 제2 그룹에 대한 선형 모델 파라미터들을 생성하기 위하여, 제1 포인트는 루마의 제1 상부 라인(5011)의 좌측 샘플 및 공동위치된 크로마 샘플(5007)이다. 그리고, 제2 포인트는 제2 상부 라인(5012)의 우측 루마 샘플 및 공동위치된 크로마 샘플(5008)이다.

[0345] 이 실시예는 제1 및 제2 그룹에 대한 4개의 포인트들의 선택을 단순화한다.

[0346] 또 다른 실시예에서, MMLM 모드의 임계치는 포인트들(5010 또는 5011)의 루마 값, 즉, 좌측 이웃의 상부 우측 포인트 및 예에서의 상부 이웃의 하부 좌측 포인트, 또는 이 포인트들 사이의 평균이다. 이 추가적인 실시예는 임계치의 연산을 단순화한다.

[0347] 2개의 포인트들의 선택에 관련된 이 모든 실시예들에 대한 또 추가의 실시예에서, 루마의 다운샘플링 프로세스가 디스에이블되고, 그것은 10진법(decimation)에 의해 대체되고, 즉, 2개의 루마 샘플들에서의 하나는 ReCL' 샘플들을 위하여 이용된다. 이 경우에, 이하에서 상세하게 설명되는 도 10에서의 단계(1001) 및 도 12에서의 단계(1201)는 스킵(skip)될 것이다. 이 실시예는 코딩 효율에 대한 미세한 영향을 갖는 복잡도를 감소시킨다.

[0348] 포인트들 A, B, C, 및 D는 샘플들의 디코딩된 버전들에 기초하여 결정되고, 이에 따라, 원래의 샘플들 값들과 정합하지 않을 수 있다. 이것은 "비정상적"인 기울기인 것을 정의할 때에 이미 설명된 바와 같이, 비정상적으로 짧은 세그먼트들 또는 단지 잡음성 추정을 야기시킬 수 있다. A 및 C가 2개의 최저 포인트들이고, B 및 D가 2개의 최고이면, 이들 중의 임의의 2개를 이용하는 대신에, 몇몇 간단한 보충적인 동작들을 희생하면서, A 및 C 사이의 평균으로서 정의된 포인트 E, 및 B 및 D 사이의 평균으로서 정의된 포인트 F가 이용될 수 있다:

$$x_E = (x_A + x_C + 1) \gg 1 \text{ and } y_E = (y_A + y_C + 1) \gg 1$$

$$x_F = (x_B + x_D + 1) \gg 1 \text{ and } y_F = (y_B + y_D + 1) \gg 1$$

$$A = \frac{(y_E - y_F) \ll S}{x_E - x_F}$$

$$\beta = y_E - A \cdot (x_E \gg S)$$

[0349]

[0350] 분명히,  $y_E - y_F$  또는  $x_E - x_F$ 가 0과 동일하거나 너무 낮을(즉, 유도된 기울기가 "비정상적"임) 경우에, 포인트들 A, B, C, 및 D는 더 양호한 파라미터들을 획득하기 위하여 통상적인 것에 따라 고려된다.

[0351] 이것으로부터, 모델에서의 기울기의 연산에서 이용된 2개의 포인트들이 ReCL' 또는 RecC의 샘플들 값들로부터 이루어진 2개의 실제적인 포인트들이 아닐 수 있다는 것이 이해되어야 한다. 이것은 "선택" 대신에, 단계(1003)에서의 "결정"이라는 자구(wording)의 이용을 설명한다.

[0352] 또 추가의 실시예에서, MMLM 모드에 대하여, 하나의 그룹의 기울기를 정의하는  $\alpha$ (또는 L) 파라미터가 "비정상적"일 경우에, 대응하는 LM 파라미터들은 다른 그룹, 또는 LM 파라미터들의 2개 초과 그룹들일 경우에 또 다른 그룹의 LM 파라미터들과 동일하게 설정된다. 도 12는 이 실시예를 예시한다. 단계(1203)에서의 2개의 그룹들에 대한 2개의 모델들을 정의하는  $(\alpha_1, \beta_1)$  및  $(\alpha_2, \beta_2)$ 의 결정 후에,  $\alpha_1$  및  $\alpha_2$ 는 이들이 단계들(1204 및 1205)에서 0과 동일한지를 체크하기 위하여 테스트된다. 그러할 경우에, "비정상적"인 기울기 파라미터  $\alpha$ (또는 L)는 다른 기울기 파라미터  $\alpha$ 와 동일하게 설정되고, 유사하게, 다른 그룹의 대응하는  $\beta$  파라미터 값은 또한, 단계들(1206 및 1207)에서 이용된다. 따라서, 그 경우에, 현재의 블록의 다운샘플링된 루마 샘플 값들의 값이 무엇이든지 간에, 파라미터들의 오직 하나의 세트가 이용되고, 임계치와의 비교가 없고, CCLM 모드와 동일한 복잡도가 획득된다. 이 실시예의 장점은 추가적인 선형 모델 파라미터들이 유도될 필요가 없으므로, 작은 복잡도를 갖는 코딩 효율 개선이다.

[0353] 대안적인 실시예에서, 하나의 기울기 파라미터  $\alpha$ (또는 L)가 "비정상적"일 때, 선형 파라미터들의 하나의 세트는 MMLM의 모든 초기 샘플들을 고려함으로써 재유도된다(그것은 하나의 상부 라인 및 하나의 이웃하는 행 대신에, 2개의 상부 라인들 및 2개의 이웃하는 행들에 의한 CCLM 유도에 대응함). 이 실시예는 이전의 것보다 더 양호한 코딩 효율을 제공하지만, 선형 모델 파라미터들의 세트를 재유도하는 것이 필요하므로, 이 실시예는 더 복잡하다.

[0354] 이 문서에서 설명된 바와 같은 2개의 포인트들에 의한 단순화된 LM 유도는 그것이 몇몇 LM 모드들이 경합하고 있을 때에 모든 LMS 유도를 대체하지 않는 경우를 제외하고는, 고전적인 LMS 유도보다 일반적으로 덜 효율적이다.

[0355] 하나의 실시예에서, 2개의 포인트들에 의한 LM 유도는 크로마 블록 예측자를 유도하기 위한 오직 CCLM 모드를

위하여 이용된다. 이 실시예는 코딩 효율 개선들을 제공한다.

- [0356] 하나의 실시예에서, 2개의 포인트들에 의한 유도는 가장 복잡한 예측 방법이므로, 2개의 포인트들에 의한 유도는 오직 MMLM 모드를 위하여 이용된다.
- [0357] 하나의 실시예에서, 2개의 포인트들에 의한 LM 유도는 크로마 블록 예측자를 유도하기 위한 CCLM 모드 및 MMLM 모드를 위하여 이용된다. 이 실시예는 JEM과 유사한 코딩 효율을 가지지만, 그것은 크로마 블록 예측자들의 생성을 위한 단순화된 LM 유도를 이용함으로써 최악의 경우의 복잡도를 감소시킨다. 실제로, 루마에 기초한 크로마 예측은 예측 선형 모델 모드들 중에서 최악의 경우의 복잡도를 제시하는 모드이다. 그것은 잔차 크로마 예측보다 더 복잡하다.
- [0358] 하나의 실시예에서, 2개의 포인트들에 한 LM 유도는 모든 LMS 유도들(크로마 블록 예측자 생성 및 잔차 예측)을 대체한다. 이 실시예는 JEM과 비교하여 코딩 효율을 감소시키지만, 그것은 복잡도를 상당히 감소시킨다. 이 2개의 실시예들은 파라미터들에 대하여 단계(1203)에서 이용된 유도 방법이 무엇이든지 간에 코딩 효율 개선을 제공한다는 것에 주목하십시오.
- [0359] 또 다른 실시예에서, 기울기 파라미터들  $\alpha$  (또는 L) 중의 하나 또는 양자가 "비정상적"일 경우에,  $(-1 < S) / 8$ 과 같은) 디폴트 값이 단계들(1206 및/또는 1207)에서 그 대신에 이용되고, 대응하는 값  $\beta$  가 연산된다.
- [0360] 또 추가의 실시예에서, 몇몇 LM 모드들은 인코더 측에서 경합하고 있고, 선택스 엘리먼트(syntax element)들은 디코더 측에서의 비트스트림에서 선택된 LM 모드를 시그널링할 수 있다. 이 시그널링은 어느 세트들이 이용되어야 하는지를 표시하거나, 블록-레벨 선택을 위한 후보들을 적어도 제공하기 위하여, 슬라이스-레벨(또는 PPS 또는 SPS)에서 있을 수 있다. 이 경합하는 LM 모드들 사이의 차들 중의 적어도 하나는 LM 파라미터들을 유도하기 위하여 이용된 2개의 포인트들의 세트이다. 2개의 포인트들의 세트 및 이 2개의 포인트들을 생성하기 위한 방법은 경합 중인 상이한 LM 모드들을 정의한다. 예를 들어, 하나의 LM 모드에 대하여, 2개의 포인트들은 최소 및 최대 루마 값들에 기초하여 결정되고, 하나의 또 다른 LM 모드에 대하여, 2개의 포인트들은 최대 및 최소 크로마 값들에 기초하여 선택된다.
- [0361] 또 다른 실시예는 도 5 상에서 예시된 바와 같은 가능한 위치들로부터 다수의 세트들을 정의하는 것에 있다. 4개의 이러한 상이한 포인트들은 최대 12개의 상이한 쌍들을 초래할 수 있지만, 기울기 파라미터  $\alpha$  (또는 L)의 계산을 위한 수학적에서의 분자 및 분모에 대한 가장 큰 값들로 귀착되는 것들이 선호될 수 있다. 인코더는 쌍들의 리스트를 구축하고, (루마 컴포넌트에서의 거리, 또는 양자의 루마 및 크로마 컴포넌트들을 이용하는 데카르트 거리(Cartesian distance)와 같은) 일부 기준들에 따라 이들을 소팅하여, 아마도, (즉, 그 기울기가 또 다른 것에 너무 근접할 경우에) 이들의 일부를 제거하고, 이에 따라, 선택될 수 있고 시그널링될 수 있는 파라미터들의 리스트를 구축한다.
- [0362] 이 실시예들의 장점은 코딩 효율 개선이다.
- [0363] 이 실시예들의 설명은 루마 및 크로마 컴포넌트를 언급하지만, 양자의 크로마 컴포넌트들 또는 RGB 컴포넌트들과 같은 다른 컴포넌트들에 용이하게 적용될 수 있다. 실시예에 따르면, 본 발명은 제2 크로마 컴포넌트로부터 제1 크로마 컴포넌트 샘플 값을 예측할 때에 이용된다. 또 다른 실시예에서, 본 발명은 하나 초과와 컴포넌트의 하나 초과와 샘플 값들로부터 하나의 컴포넌트의 샘플 값을 예측할 때에 이용된다. 이러한 경우에, 선형 모델은 2개의 포인트들/세트들에 기초하여 유도되고, 각각의 포인트/세트는 하나의 컴포넌트의 샘플 값, 및 하나 초과와 컴포넌트의 하나 초과와 샘플 값들을 포함한다는 것이 이해된다. 예를 들어, 2개의 컴포넌트들의 샘플 값들이 하나의 컴포넌트의 샘플 값을 예측하기 위하여 이용될 경우에, 각각의 포인트/세트는 3 차원 공간에서의 포지션으로서 표현될 수 있고, 선형 모델은 재구성된 샘플 값들의 2개의 포인트들/세트들에 대응하는 3 차원 공간에서의 2개의 포지션들을 통과하는 직선 라인에 기초한다.
- [0364] 도 13은 발명의 하나 이상의 실시예들의 구현을 위한 컴퓨팅 디바이스(1300)의 개략적인 블록도이다. 컴퓨팅 디바이스(1300)는 마이크로-컴퓨터, 워크스테이션, 또는 광 휴대용 디바이스와 같은 디바이스일 수 있다. 컴퓨팅 디바이스(1300)는 이하에 접속된 통신 버스를 포함한다:
- [0365] - CPU로 나타낸 마이크로프로세서와 같은 중앙 프로세싱 유닛(1301);
- [0366] - 발명의 실시예들의 방법의 실행가능 코드를 저장하기 위한 RAM으로 나타낸 랜덤 액세스 메모리(random access memory)(1302) 뿐만 아니라, 발명의 실시예들에 따라 이미지의 적어도 일부를 인코딩하거나 디코딩하기 위한 방

법을 구현하기 위하여 필요한 변수들 및 파라미터들을 레코딩하도록 적응된 레지스터들 - 그 메모리 용량은 예를 들어, 확장 포트에 접속된 임의적인 RAM에 의해 확대될 수 있음 -;

- [0367] - 발명의 실시예들을 구현하기 위한 컴퓨터 프로그램들을 저장하기 위한, ROM으로 나타낸 판독 전용 메모리(read only memory)(1303);
- [0368] - 네트워크 인터페이스(1304)는 프로세싱되어야 할 디지털 데이터가 그 상에서 송신되거나 수신되는 통신 네트워크에 전형적으로 접속된다. 네트워크 인터페이스(1304)는 단일 네트워크 인터페이스일 수 있거나, 상이한 네트워크 인터페이스들(예를 들어, 유선 및 무선 인터페이스들, 또는 상이한 종류들의 유선 또는 무선 인터페이스들)의 세트로 구성될 수 있다. 데이터 패킷들은 CPU(1301)에서 작동되는 소프트웨어 애플리케이션의 제어 하에서, 송신을 위하여 네트워크 인터페이스로 기록되거나, 수신을 위하여 네트워크 인터페이스로부터 판독된다;
- [0369] - 사용자 인터페이스(1305)는 사용자로부터 입력들을 수신하거나 정보를 사용자에게 디스플레이하기 위하여 이용될 수 있다;
- [0370] - HD로 나타낸 하드 디스크(1306)는 대용량 저장 디바이스로서 제공될 수 있다;
- [0371] - I/O 모듈(1307)은 비디오 소스 또는 디스플레이와 같은 외부 디바이스들로부터/외부 디바이스들로 데이터를 수신/전송하기 위하여 이용될 수 있다.
- [0372] 실행가능 코드는 판독 전용 메모리(1303) 내에, 하드 디스크(1306) 상에서, 또는 예를 들어, 디스크와 같은 분리가능한 디지털 매체 상에서 중의 어느 하나로 저장될 수 있다. 변형에 따르면, 프로그램들의 실행가능 코드는 실행되기 전에, 하드 디스크(1306)와 같은, 통신 디바이스(1300)의 저장 수단들 중의 하나에서 저장되도록 하기 위하여, 네트워크 인터페이스(1304)를 통해, 통신 네트워크의 수단에 의해 수신될 수 있다.
- [0373] 중앙 프로세싱 유닛(1301)은 발명의 실시예들에 따라 프로그램 또는 프로그램들의 소프트웨어 코드의 명령어들 또는 부분들의 실행을 제어하고 지시하도록 적응되고, 명령어들은 전송한 저장 수단들 중의 하나에서 저장된다. 파워 온(power on) 후, CPU(1301)는 그 명령어들이 예를 들어, 프로그램 ROM(1303) 또는 하드-디스크(hard-disc)(HD)(1306)로부터 로딩된 후에, 소프트웨어 애플리케이션에 관련되는 주요 RAM 메모리(1302)로부터의 명령어들을 실행할 수 있다. 이러한 소프트웨어 애플리케이션은, CPU(1301)에 의해 실행될 때, 발명에 따른 방법의 단계들이 수행되게 한다.
- [0374] 발명에 따른 방법들의 임의의 단계는 PC("Personal Computer(개인용 컴퓨터)"), DSP("Digital Signal Processor(디지털 신호 프로세서)"), 또는 마이크로제어기와 같은 프로그래밍가능 컴퓨팅 머신에 의해 명령어들의 세트 또는 프로그램의 실행에 의해 소프트웨어로 구현될 수 있거나; 또는 그 밖에, 특히 최소들 및 최대들 선택을 위한 FPGA("Field-Programmable Gate Array(필드-프로그래밍가능 게이트 어레이)"), 또는 ASIC("Application-Specific Integrated Circuit(애플리케이션-특정 집적 회로)")과 같은 머신 또는 전용 컴포넌트에 의해 하드웨어로 구현될 수 있다.
- [0375] 일부 예들은 예시를 위하여 HEVC에 기초하고 있지만, 발명은 HEVC로 제한되지는 않는다는 것이 또한 주목되어야 한다. 예를 들어, 본 발명은 또한, 2개 이상의 컴포넌트들의 샘플 값들 사이의 관계가 모델로 추정/예측될 수 있는 임의의 다른 추정/예측 프로세스에서 이용될 수 있고, 여기서, 모델은 관련된/연관된 컴포넌트 샘플 값들의 모든 이용가능한 세트들로부터 선택된 관련된/연관된 컴포넌트 샘플 값들의 적어도 2개의 세트들에 기초하여 결정된 근사 모델이다.
- [0376] 샘플 쌍(즉, 상이한 컴포넌트들에 대한 연관된 샘플 값들의 세트)에 대응하는 각각의 포인트는 어레이의 측면에서 저장될 수 있고 및/또는 프로세싱될 수 있다는 것이 이해된다. 예를 들어, 각각의 컴포넌트의 샘플 값들은 그 컴포넌트의 각각의 샘플 값이 예를 들어, 그 샘플 값에 대한 인덱스를 이용하여 그 어레이의 엘리먼트를 참조함으로써 참조가능/액세스가능/획득가능하도록 어레이에서 저장될 수 있다. 대안적으로, 어레이는 샘플 쌍들의 각각의 샘플 값이 어레이의 엘리먼트로서 액세스가능/획득가능한 각각의 샘플 쌍들을 저장하고 프로세싱하기 위하여 이용될 수 있다.
- [0377] 위에서 설명된 비교, 결정, 평가, 선택, 또는 고려, 예를 들어, 인코딩 프로세스 동안에 행해진 선택의 임의의 결과는 비트스트림에서의 데이터, 예를 들어, 결과를 표시하는 플래그 또는 데이터에서 표시될 수 있거나 이로부터 결정가능할 수 있어서, 표시된 또는 결정된 결과는 예를 들어, 디코딩 프로세스 동안에, 비교, 결정, 평가, 선택, 또는 고려를 실제적으로 수행하는 대신에, 프로세싱에서 이용될 수 있다는 것이 또한 이해된다.
- [0378] 본 발명은 특정 실시예들을 참조하여 상기에서 설명되었지만, 본 발명은 특정 실시예들로 제한되지는 않고, 수

정들은 본 기술분야의 통상의 기술자에게 명백할 것이고, 이것은 본 발명의 범위 내에 있다.

- [0379] 많은 추가의 수정들 및 변형들은, 오직 예로서 제공되고, 첨부된 청구항들에 의해 전적으로 결정되는 발명의 범위를 제한하도록 의도되지 않은 상이한 예시적인 실시예들을 참조할 시에 본 기술분야에서 정통한 기술자들에게 시사할 것이다. 특히, 상이한 실시예들로부터의 상이한 특징들은 적절한 경우에 교환될 수 있다.
- [0380] 위에서 설명된 발명의 실시예들의 각각은 단독으로, 또는 복수의 실시예들의 조합으로서 구현될 수 있다. 또한, 상이한 실시예들로부터의 특징들은 필요한 경우에, 또는 단일 실시예에서의 개별적인 실시예들로부터의 엘리먼트들 또는 특징들의 조합이 유익할 경우에 조합될 수 있다.
- [0381] (임의의 동반 청구항들, 요약서, 및 도면들을 포함하는) 이 명세서에서 개시된 각각의 특징은 이와 다르게 명백히 기재되지 않으면, 동일한, 동등한, 또는 유사한 목적을 제공하는 대안적인 특징들에 의해 대체될 수 있다. 이에 따라, 이와 다르게 명백히 기재되지 않으면, 개시된 각각의 특징은 전반적인 일련의 동등한 또는 유사한 특징들의 오직 하나의 예이다.
- [0382] 청구항들에서, 단어 "포함하는"은 다른 엘리먼트들 또는 단계들을 배제하지 않고, 부정 관사 "a" 또는 "an"은 배제하지 않는다. 상이한 특징들이 상호 상이한 종속항들에서 인용된다는 단순한 사실은 이 특징들의 조합이 유리하게 이용될 수 없다는 것을 표시하지는 않는다.
- [0383] 다음의 번호부여된 조항들은 어떤 실시예들을 또한 정의한다:
- [0384] 1. 동일한 프레임에서의 제2-컴포넌트 블록의 연관된 재구성된 제2-컴포넌트 샘플로부터 제1-컴포넌트 블록에 대한 제1-컴포넌트 샘플을 획득하기 위한 선형 모델을 유도하는 방법으로서,
- [0385] 제1-컴포넌트 및 제2-컴포넌트 둘 모두의 재구성된 샘플들에 기초하여 2개의 포인트들 - 각각의 포인트는 2개의 변수들에 의해 정의되고, 제1 변수는 제2-컴포넌트 샘플 값에 대응하고, 제2 변수는 제1-컴포넌트 샘플 값에 대응함 - 을 결정하는 단계;
- [0386] 2개의 포인트들에 의한 직선 라인 통과를 표현하는 선형 수학식의 파라미터들을 결정하는 단계; 및
- [0387] 직선 라인 파라미터들에 의해 정의된 선형 모델을 유도하는 단계를 포함하는, 방법.
- [0388] 2. 조항 1의 방법에 있어서, 2개의 포인트들은 제2-컴포넌트 블록의 이웃에서의 샘플 쌍들에 기초하여 결정되는, 방법.
- [0389] 3. 조항 2의 방법에 있어서, 2개의 포인트들은 제2-컴포넌트 블록의 이웃에서의 샘플 쌍들의 샘플 값들에 기초하여 결정되는, 방법.
- [0390] 4. 조항 3의 방법에 있어서, 2개의 포인트들은 최저 제2-컴포넌트 샘플 값 및 최고 제2-컴포넌트 샘플 값을 갖는 샘플 쌍들에 각각 대응하는, 방법.
- [0391] 5. 조항 3의 방법에 있어서, 2개의 포인트들은 최저 제1-컴포넌트 샘플 값 및 최고 제1-컴포넌트 샘플 값을 갖는 샘플 쌍들에 각각 대응하는, 방법.
- [0392] 6. 조항 3의 방법에 있어서, 상기 방법은,
- [0393] - 최저 제2-컴포넌트 샘플 값 및 최고 제2-컴포넌트 샘플 값을 갖는 샘플 쌍들에 각각 대응하는 2개의 제1 포인트들을 결정하는 단계;
- [0394] - 최저 제1-컴포넌트 샘플 값 및 최고 제1-컴포넌트 샘플 값을 갖는 샘플 쌍들에 각각 대응하는 2개의 제2 포인트들을 결정하는 단계; 및
- [0395] - 2개의 포인트들이 더 긴 세그먼트를 형성할 경우에, 2개의 포인트들을 2개의 제1 포인트들로서 결정하고, 그렇지 않을 경우에, 2개의 포인트들을 2개의 제2 포인트들로서 결정하는 단계를 포함하는, 방법.
- [0396] 7. 조항 3의 방법에 있어서, 상기 방법은,
- [0397] - 최저 제2-컴포넌트 샘플 값 및 최고 제2-컴포넌트 샘플 값을 갖는 샘플 쌍들에 각각 대응하는 2개의 제1 포인트들을 결정하는 단계;
- [0398] - 최저 제1-컴포넌트 샘플 값 및 최고 제1-컴포넌트 샘플 값을 갖는 샘플 쌍들에 각각 대응하는 2개의 제2 포인트들을 결정하는 단계; 및

- [0399] - 2개의 제1 포인트들 및 2개의 제2 포인트들 중의 2개의 포인트들을, 가장 긴 세그먼트를 형성하는 2개의 포인트들로서 결정하는 단계를 포함하는, 방법.
- [0400] 8. 조항 3의 방법에 있어서, 상기 방법은,
- [0401] - 최저 제2-컴포넌트 샘플 값 및 최고 제2-컴포넌트 샘플 값을 갖는 샘플 쌍들에 각각 대응하는 2개의 제1 포인트들을 결정하는 단계;
- [0402] - 최저 제1-컴포넌트 샘플 값 및 최고 제1-컴포넌트 샘플 값을 갖는 샘플 쌍들에 각각 대응하는 2개의 제2 포인트들을 결정하는 단계; 및
- [0403] - 모든 그 변수들이 상이할 경우에, 2개의 포인트들을 2개의 제1 포인트들로서 결정하고, 그렇지 않을 경우에, 2개의 포인트들을 2개의 제2 포인트들로서 결정하는 단계를 포함하는, 방법.
- [0404] 9. 조항 3의 방법에 있어서, 상기 방법은,
- [0405] - 최저 제2-컴포넌트 샘플 값 및 최고 제2-컴포넌트 샘플 값을 갖는 샘플 쌍들에 각각 대응하는 2개의 제1 포인트들을 결정하는 단계;
- [0406] - 최저 제1-컴포넌트 샘플 값 및 최고 제1-컴포넌트 샘플 값을 갖는 샘플 쌍들에 각각 대응하는 2개의 제2 포인트들을 결정하는 단계; 및
- [0407] - 이 2개의 포인트들에 의해 정의된 직선 라인의 기울기 파라미터가 주어진 임계치보다 더 클 경우에, 2개의 포인트들을 2개의 제1 포인트들로서 결정하고, 그렇지 않을 경우에, 2개의 포인트들을 2개의 제2 포인트들로서 결정하는 단계를 포함하는, 방법.
- [0408] 10. 조항 3의 방법에 있어서, 상기 방법은,
- [0409] - 최저 제2-컴포넌트 샘플 값 및 최고 제2-컴포넌트 샘플 값을 갖는 샘플 쌍들에 각각 대응하는 2개의 제1 포인트들을 결정하는 단계;
- [0410] - 최저 제1-컴포넌트 샘플 값 및 최고 제1-컴포넌트 샘플 값을 갖는 샘플 쌍들에 각각 대응하는 2개의 제2 포인트들을 결정하는 단계; 및
- [0411] - 최저 제2-컴포넌트 샘플 값과 최고 제2-컴포넌트 샘플 값 사이의 차가 최저 제1-컴포넌트 샘플 값과 최고 제1-컴포넌트 샘플 값 사이의 차보다 더 클 경우에, 2개의 포인트들을 2개의 제1 포인트들로서 결정하고, 그렇지 않을 경우에, 2개의 포인트들을 2개의 제2 포인트들로서 결정하는 단계를 포함하는, 방법.
- [0412] 11. 조항 2의 방법에 있어서, 2개의 포인트들은 제2-컴포넌트 블록의 이웃에서의 샘플 쌍들의 제2-컴포넌트 샘플 값의 포지션에 기초하여 결정되는, 방법.
- [0413] 12. 조항 11의 방법에 있어서, 2개의 포인트들은 제2-컴포넌트 블록의 이웃에서의 미리 결정된 포지션에서의 샘플 쌍들에 대응하는 것으로서 결정되는, 방법.
- [0414] 13. 조항 12의 방법에 있어서, 미리 결정된 포지션에서의 샘플 쌍이 이용가능하지 않을 때, 2개의 포인트들 중의 적어도 하나를 제2 미리 결정된 포지션에서의 샘플 쌍에 대응하는 것으로서 결정하는 단계를 더 포함하는, 방법.
- [0415] 14. 조항 1의 방법에 있어서, 2개의 포인트들은 제2-컴포넌트 블록의 이웃에서의 샘플 쌍들 및 제2-컴포넌트 블록 샘플 값들에 기초하여 결정되는, 방법.
- [0416] 15. 조항 14의 방법에 있어서,
- [0417] - 2개의 포인트들의 제1 변수들은 제2-컴포넌트 블록에서의 최대 빈도 및 제2-컴포넌트 블록에서의 제2 최대 빈도로, 제2-컴포넌트 블록의 이웃에서의 샘플 쌍들 중에서 샘플 값으로서 결정되고;
- [0418] - 2개의 포인트들의 제2 변수들은 제2-컴포넌트 블록의 이웃에서의 샘플 쌍들에 기초하여 대응하는 제1-컴포넌트 샘플 값으로서 결정되는, 방법.
- [0419] 16. 조항 1 내지 15 중 어느 한 조항의 방법에 있어서,
- [0420] - 제2-컴포넌트 블록의 샘플들은 적어도 2개의 그룹들로 편성되고; 그리고

- [0421] - 2개의 포인트들은 제2-컴포넌트 블록의 샘플들의 각각의 그룹에 대한 선형 모델의 정의를 위하여 결정되는, 방법.
- [0422] 17. 조항 16의 방법에 있어서, 그룹에 대하여 결정된 2개의 포인트들이 미리 결정된 임계치보다 더 낮은 기울기 파라미터에 대응할 경우에, 2개의 포인트들은 또 다른 그룹에 대하여 결정된 2개의 포인트들에 의해 대체되는, 방법.
- [0423] 18. 조항 16의 방법에 있어서, 그룹에 대하여 결정된 2개의 포인트들이 미리 결정된 임계치보다 더 낮은 기울기 파라미터에 대응할 경우에, 2개의 새로운 포인트들은 단일 그룹으로서 고려된 모든 그룹들의 샘플들에 기초하여 결정되는, 방법.
- [0424] 19. 동일한 프레임에서의 제2-컴포넌트 블록의 연관된 재구성된 제2-컴포넌트 샘플로부터 제1-컴포넌트 블록에 대한 제1-컴포넌트 샘플을 획득하는 방법으로서,
- [0425] - 단일 선형 모델을 이용하는 CCLM 모드들 및 몇몇 선형 모델들을 이용하는 MMLM 모드들을 포함하는 복수의 선형 모델 유도 모드들을 정의하는 단계; 및
- [0426] - 제1-컴포넌트 블록에 대한 상기 제1-컴포넌트 샘플들을 획득하기 위한 상기 선형 모델 유도 모드들 중의 하나를 선택하는 단계를 포함하고,
- [0427] - 선형 모델 유도 모드들 중의 적어도 하나는 조항 1 내지 18 중 어느 한 조항에 따른 유도의 방법을 이용하는, 방법.
- [0428] 20. 조항 19의 방법에 있어서, 오직 CCLM 모드들은 조항 1 내지 18 중 어느 한 조항에 따른 유도의 방법을 이용하는, 방법.
- [0429] 21. 조항 19의 방법에 있어서, 오직 MMLM 모드들은 조항 1 내지 18 중 어느 한 조항에 따른 유도의 방법을 이용하는, 방법.
- [0430] 22. 하나 이상의 이미지들을 비트스트림으로 인코딩하는 방법으로서, 방법은 조항 1 내지 18 중 어느 한 조항에 따른 선형 모델을 유도하는 단계를 포함하는, 방법.
- [0431] 23. 하나 이상의 이미지들을 비트스트림으로 인코딩하는 방법으로서, 방법은 조항들 19 내지 21 중 어느 한 조항에 따른 연관된 재구성된 제2-컴포넌트 샘플 블록으로부터 하나 이상의 이미지들의 제1-컴포넌트 블록에 대한 제1-컴포넌트 샘플을 획득하는 단계를 포함하는, 방법.
- [0432] 24. 비트스트림으로부터 하나 이상의 이미지들을 디코딩하는 방법으로서, 방법은 조항들 1 내지 18 중 어느 한 조항에 따른 선형 모델을 유도하는 단계를 포함하는, 방법.
- [0433] 25. 비트스트림으로부터 하나 이상의 이미지들을 디코딩하는 방법으로서, 방법은 조항들 19 내지 21 중 어느 한 조항에 따른 연관된 재구성된 제2-컴포넌트 샘플 블록으로부터 하나 이상의 이미지들의 제1-컴포넌트 블록에 대한 제1-컴포넌트 샘플을 획득하는 단계를 포함하는, 방법.
- [0434] 26. 동일한 프레임에서의 제2-컴포넌트 블록의 연관된 재구성된 제2-컴포넌트 샘플로부터 제1-컴포넌트 블록에 대한 제1-컴포넌트 샘플을 획득하기 위한 선형 모델을 유도하기 위한 디바이스로서,
- [0435] - 제1-컴포넌트 및 제2-컴포넌트 둘 모두의 재구성된 샘플들에 기초하여 2개의 포인트들 - 각각의 포인트는 2개의 변수들에 의해 정의되고, 제1 변수는 제2-컴포넌트 샘플 값에 대응하고, 제2 변수는 제1-컴포넌트 샘플 값에 대응함 - 을 결정하기 위한 수단;
- [0436] - 2개의 포인트들에 의한 직선 라인 통과를 표현하는 선형 수학식의 파라미터들을 결정하기 위한 수단; 및
- [0437] - 직선 라인 파라미터들에 의해 정의된 선형 모델을 유도하기 위한 수단을 포함하는, 디바이스.
- [0438] 27. 조항 26의 디바이스에 있어서, 2개의 포인트들은 제2-컴포넌트 블록의 이웃에서의 샘플 쌍들에 기초하여 결정되는, 디바이스.
- [0439] 28. 조항 27의 디바이스에 있어서, 2개의 포인트들은 제2-컴포넌트 블록의 이웃에서의 샘플 쌍들의 샘플 값들에 기초하여 결정되는, 디바이스.
- [0440] 29. 조항 26의 디바이스에 있어서, 2개의 포인트들은 최저 제2-컴포넌트 샘플 값 및 최고 제2-컴포넌트 샘플 값

을 갖는 샘플 쌍들에 각각 대응하는, 디바이스.

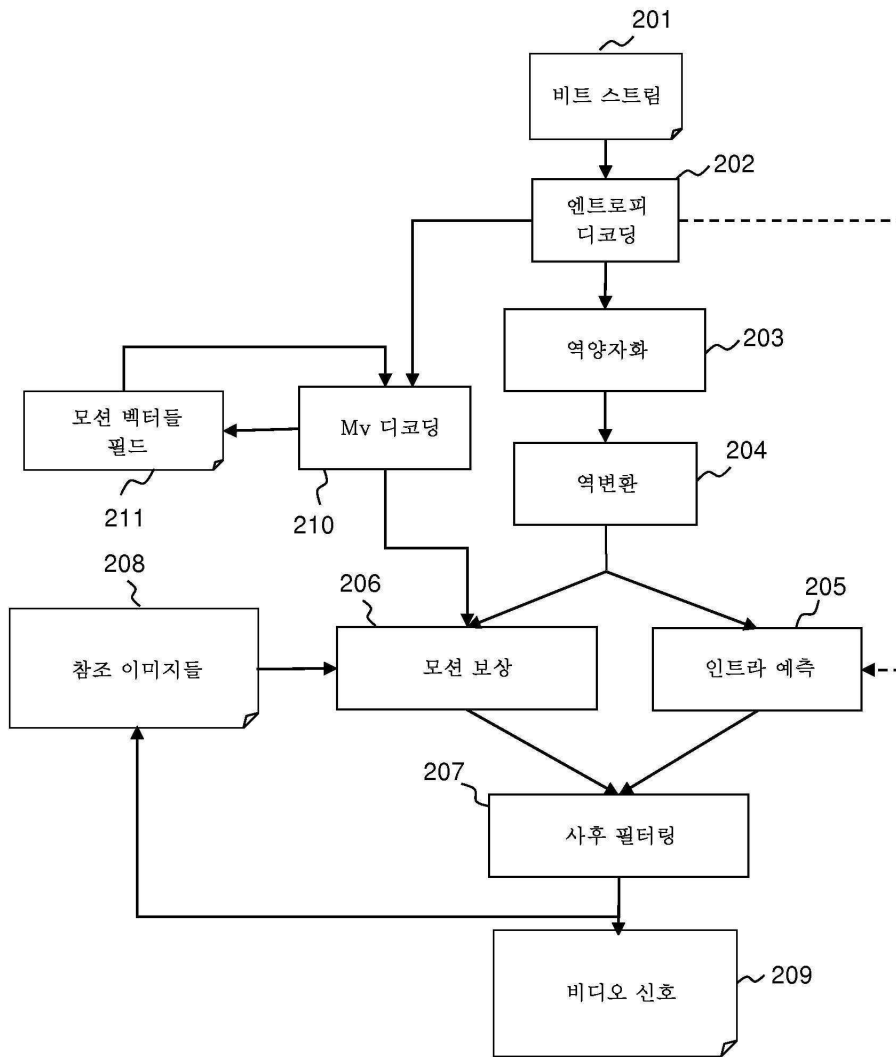
- [0441] 30. 조항 26의 디바이스에 있어서, 2개의 포인트들은 최저 제1-컴포넌트 샘플 값 및 최고 제1-컴포넌트 샘플 값을 갖는 샘플 쌍들에 각각 대응하는, 디바이스.
- [0442] 31. 조항 26의 디바이스에 있어서, 수단은,
- [0443] - 최저 제2-컴포넌트 샘플 값 및 최고 제2-컴포넌트 샘플 값을 갖는 샘플 쌍들에 각각 대응하는 2개의 제1 포인트들을 결정하는 것;
- [0444] - 최저 제1-컴포넌트 샘플 값 및 최고 제1-컴포넌트 샘플 값을 갖는 샘플 쌍들에 각각 대응하는 2개의 제2 포인트들을 결정하는 것; 및
- [0445] - 2개의 포인트들이 더 긴 세그먼트를 형성할 경우에, 2개의 포인트들을 2개의 제1 포인트들로서 결정하고, 그렇지 않을 경우에, 2개의 포인트들을 2개의 제2 포인트들로서 결정하는 것을 수행하도록 구성되는, 디바이스.
- [0446] 32. 조항 26의 디바이스에 있어서, 수단은,
- [0447] - 최저 제2-컴포넌트 샘플 값 및 최고 제2-컴포넌트 샘플 값을 갖는 샘플 쌍들에 각각 대응하는 2개의 제1 포인트들을 결정하는 것;
- [0448] - 최저 제1-컴포넌트 샘플 값 및 최고 제1-컴포넌트 샘플 값을 갖는 샘플 쌍들에 각각 대응하는 2개의 제2 포인트들을 결정하는 것; 및
- [0449] - 2개의 제1 포인트들 및 2개의 제2 포인트들 중의 2개의 포인트들을, 가장 긴 세그먼트를 형성하는 2개의 포인트들로서 결정하는 것을 수행하도록 구성되는, 디바이스.
- [0450] 33. 조항 26의 디바이스에 있어서, 수단은,
- [0451] - 최저 제2-컴포넌트 샘플 값 및 최고 제2-컴포넌트 샘플 값을 갖는 샘플 쌍들에 각각 대응하는 2개의 제1 포인트들을 결정하는 것;
- [0452] - 최저 제1-컴포넌트 샘플 값 및 최고 제1-컴포넌트 샘플 값을 갖는 샘플 쌍들에 각각 대응하는 2개의 제2 포인트들을 결정하는 것; 및
- [0453] - 모든 그 변수들이 상이할 경우에, 2개의 포인트들을 2개의 제1 포인트들로서 결정하고, 그렇지 않을 경우에, 2개의 포인트들을 2개의 제2 포인트들로서 결정하는 것을 수행하도록 구성되는, 디바이스.
- [0454] 34. 조항 26의 디바이스에 있어서, 수단은,
- [0455] - 최저 제2-컴포넌트 샘플 값 및 최고 제2-컴포넌트 샘플 값을 갖는 샘플 쌍들에 각각 대응하는 2개의 제1 포인트들을 결정하는 것;
- [0456] - 최저 제1-컴포넌트 샘플 값 및 최고 제1-컴포넌트 샘플 값을 갖는 샘플 쌍들에 각각 대응하는 2개의 제2 포인트들을 결정하는 것; 및
- [0457] - 이 2개의 포인트들에 의해 정의된 직선 라인의 기울기 파라미터가 주어진 임계치보다 더 클 경우에, 2개의 포인트들을 2개의 제1 포인트들로서 결정하고, 그렇지 않을 경우에, 2개의 포인트들을 2개의 제2 포인트들로서 결정하는 것을 수행하도록 구성되는, 디바이스.
- [0458] 35. 조항 26의 디바이스에 있어서, 수단은,
- [0459] - 최저 제2-컴포넌트 샘플 값 및 최고 제2-컴포넌트 샘플 값을 갖는 샘플 쌍들에 각각 대응하는 2개의 제1 포인트들을 결정하는 것;
- [0460] - 최저 제1-컴포넌트 샘플 값 및 최고 제1-컴포넌트 샘플 값을 갖는 샘플 쌍들에 각각 대응하는 2개의 제2 포인트들을 결정하는 것; 및
- [0461] - 최저 제2-컴포넌트 샘플 값과 최고 제2-컴포넌트 샘플 값 사이의 차가 최저 제1-컴포넌트 샘플 값과 최고 제1-컴포넌트 샘플 값 사이의 차보다 더 클 경우에, 2개의 포인트들을 2개의 제1 포인트들로서 결정하고, 그렇지 않을 경우에, 2개의 포인트들을 2개의 제2 포인트들로서 결정하는 것을 수행하도록 구성되는, 디바이스.
- [0462] 36. 조항 27의 디바이스에 있어서, 2개의 포인트들은 제2-컴포넌트 블록의 이웃에서의 샘플 쌍들의 포지션에 기

초하여 결정되는, 디바이스.

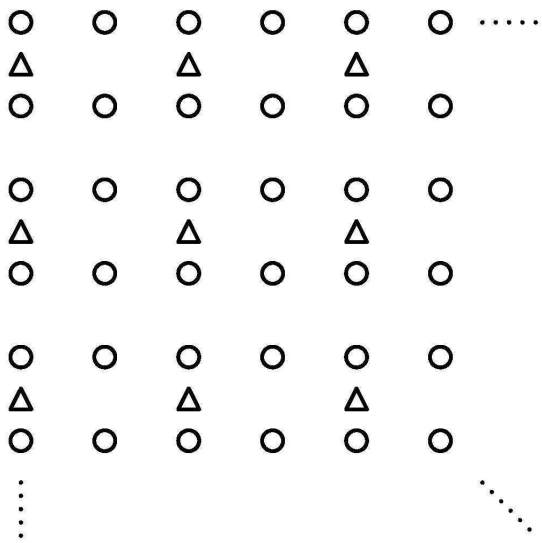
- [0463] 37. 조항 36의 디바이스에 있어서, 2개의 포인트들은 제2-컴포넌트 블록의 이웃에서의 미리 결정된 포지션에서의 샘플 쌍들에 대응하는 것으로서 결정되는, 디바이스.
- [0464] 38. 조항 37의 디바이스에 있어서, 미리 결정된 포지션에서의 샘플 쌍이 이용가능하지 않을 때, 2개의 포인트들 중의 적어도 하나를 제2 미리 결정된 포지션에서의 샘플 쌍에 대응하는 것으로서 결정하는 것을 더 포함하는, 디바이스.
- [0465] 39. 조항 26의 디바이스에 있어서, 2개의 포인트들은 제2-컴포넌트 블록의 이웃에서의 샘플 쌍들 및 제2-컴포넌트 블록 샘플 값들에 기초하여 결정되는, 디바이스.
- [0466] 40. 조항 39의 디바이스에 있어서,
- [0467] - 2개의 포인트들의 제1 변수들은 제2-컴포넌트 블록에서의 최대 빈도 및 제2-컴포넌트 블록에서의 제2 최대 빈도로 샘플 값으로서 결정되고;
- [0468] - 2개의 포인트들의 제2 변수들은 제2-컴포넌트 블록의 이웃에서의 샘플 쌍들에 기초하여 대응하는 제1-컴포넌트 샘플 값으로서 결정되는, 디바이스.
- [0469] 41. 조항 26 내지 40 중 어느 한 조항의 디바이스에 있어서,
- [0470] - 제2-컴포넌트 블록의 샘플들은 적어도 2개의 그룹들로 편성되고; 그리고
- [0471] - 2개의 포인트들은 제2-컴포넌트 블록의 샘플들의 각각의 그룹에 대한 선형 모델의 정의를 위하여 결정되는, 디바이스.
- [0472] 42. 조항 41의 디바이스에 있어서, 그룹에 대하여 결정된 2개의 포인트들이 미리 결정된 임계치보다 더 낮은 기울기 파라미터에 대응할 경우에, 2개의 포인트들은 또 다른 그룹에 대하여 결정된 2개의 포인트들에 의해 대체되는, 디바이스.
- [0473] 43. 조항 41의 디바이스에 있어서, 그룹에 대하여 결정된 2개의 포인트들이 미리 결정된 임계치보다 더 낮은 기울기 파라미터에 대응할 경우에, 2개의 새로운 포인트들은 단일 그룹으로서 고려된 모든 그룹들의 샘플들에 기초하여 결정되는, 디바이스.
- [0474] 44. 동일한 프레임에서의 제2-컴포넌트 블록의 연관된 재구성된 제2-컴포넌트 샘플로부터 제1-컴포넌트 블록에 대한 제1-컴포넌트 샘플을 획득하기 위한 디바이스로서,
- [0475] - 단일 선형 모델을 이용하는 CCLM 모드들 및 몇몇 선형 모델들을 이용하는 MMLM 모드들을 포함하는 복수의 선형 모델 유도 모드들을 정의하기 위한 수단;
- [0476] - 제1-컴포넌트 블록에 대한 상기 제1-컴포넌트 샘플들을 획득하기 위한 상기 선형 모델 유도 모드들 중의 하나를 선택하기 위한 수단을 포함하고,
- [0477] - 상기 선형 모델 유도 모드들 중의 적어도 일부는 조항 1 내지 18 중 어느 한 조항에 따른 유도의 방법을 이용하는, 디바이스.
- [0478] 45. 조항 44의 디바이스에 있어서, 오직 CCLM 모드들은 조항 1 내지 18 중 어느 한 조항에 따른 유도의 방법을 이용하는, 디바이스.
- [0479] 46. 조항 44의 디바이스에 있어서, 오직 MMLM 모드들은 조항 1 내지 18 중 어느 한 조항에 따른 유도의 방법을 이용하는, 디바이스.
- [0480] 47. 이미지들을 인코딩하기 위한 디바이스로서, 디바이스는 조항 1 내지 18 중 어느 한 조항에 따른 선형 모델을 유도하기 위한 수단을 포함하는, 디바이스.
- [0481] 48. 이미지들을 디코딩하기 위한 디바이스로서, 디바이스는 조항 1 내지 18 중 어느 한 조항에 따른 선형 모델을 유도하기 위한 수단을 포함하는, 디바이스.
- [0482] 49. 프로그래밍가능 장치를 위한 컴퓨터 프로그램 제품으로서, 컴퓨터 프로그램 제품은, 프로그래밍가능 장치에 의해 로딩되고 실행될 때, 조항들 1 내지 25 중 어느 한 조항에 따른 방법을 구현하기 위한 명령어들의 시퀀스를 포함하는, 컴퓨터 프로그램 제품.



도면2

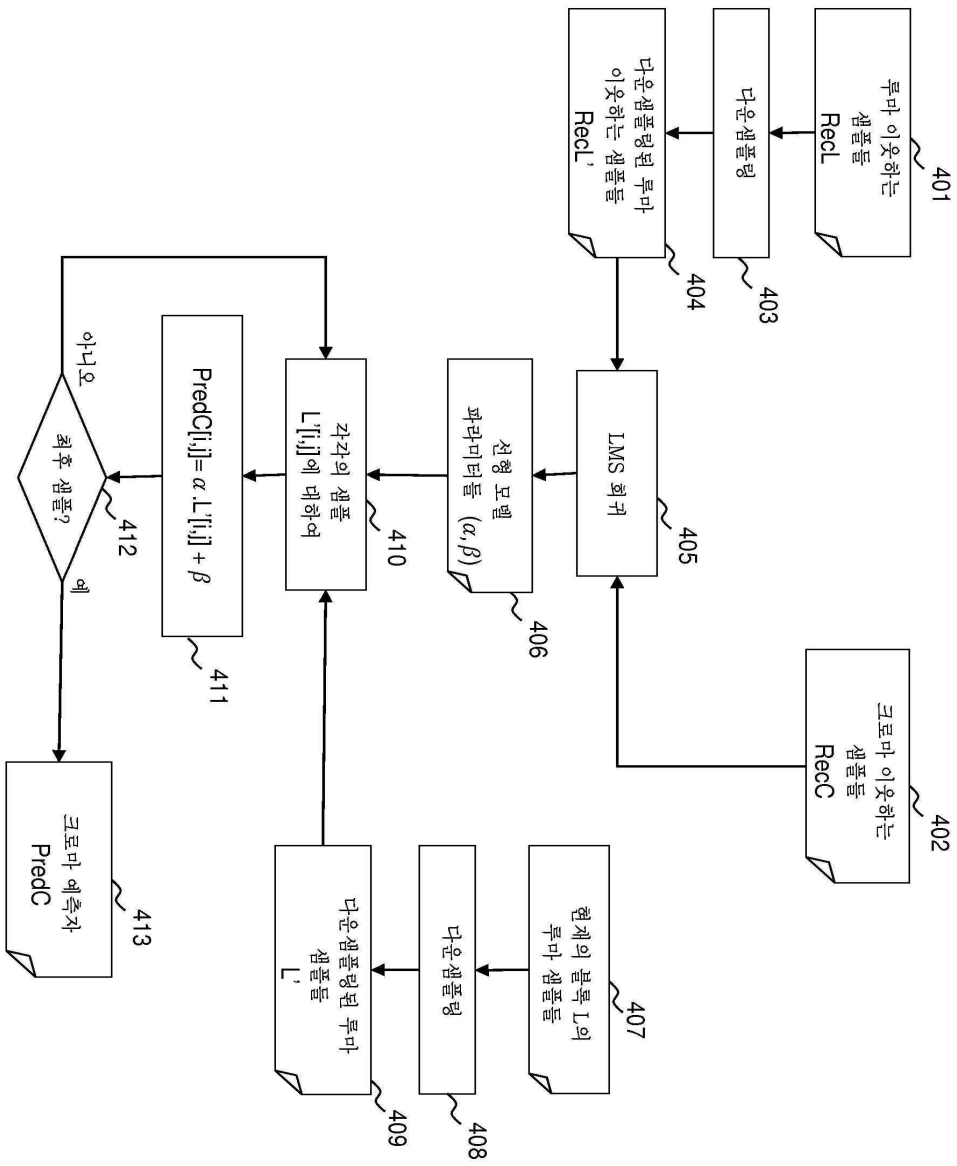


도면3

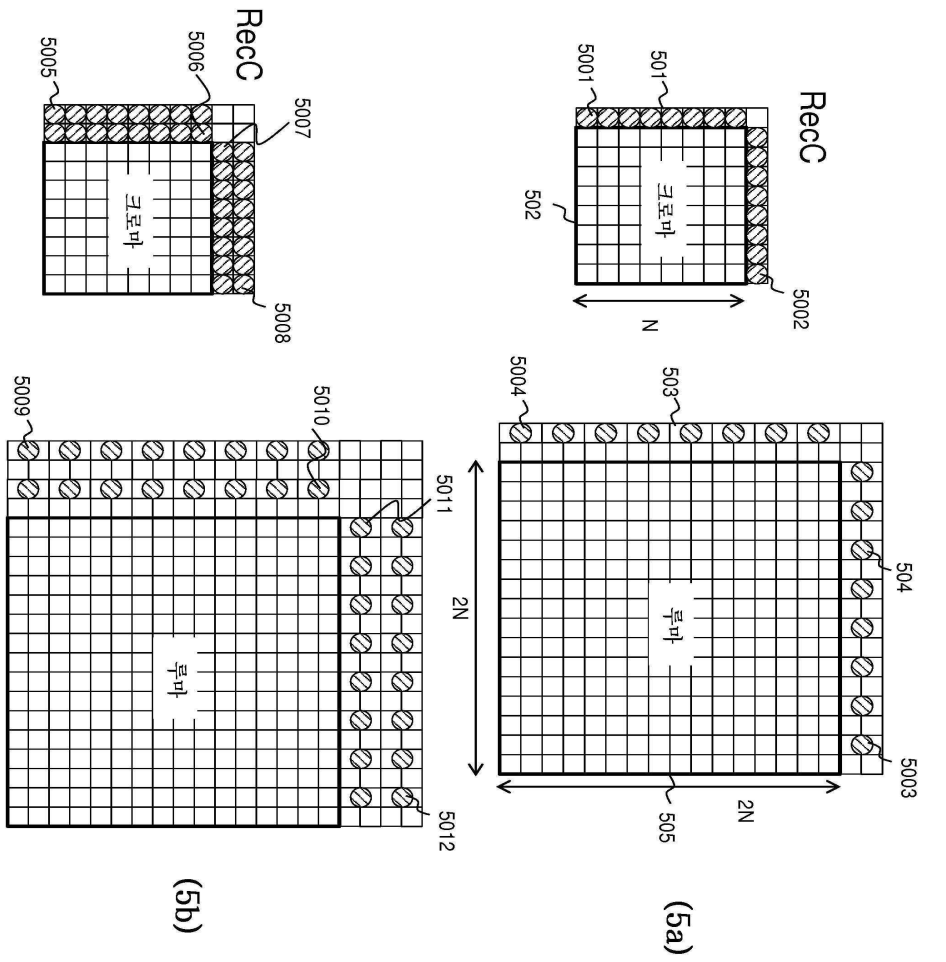


○ 루마 샘플들  
△ 크로마 샘플들

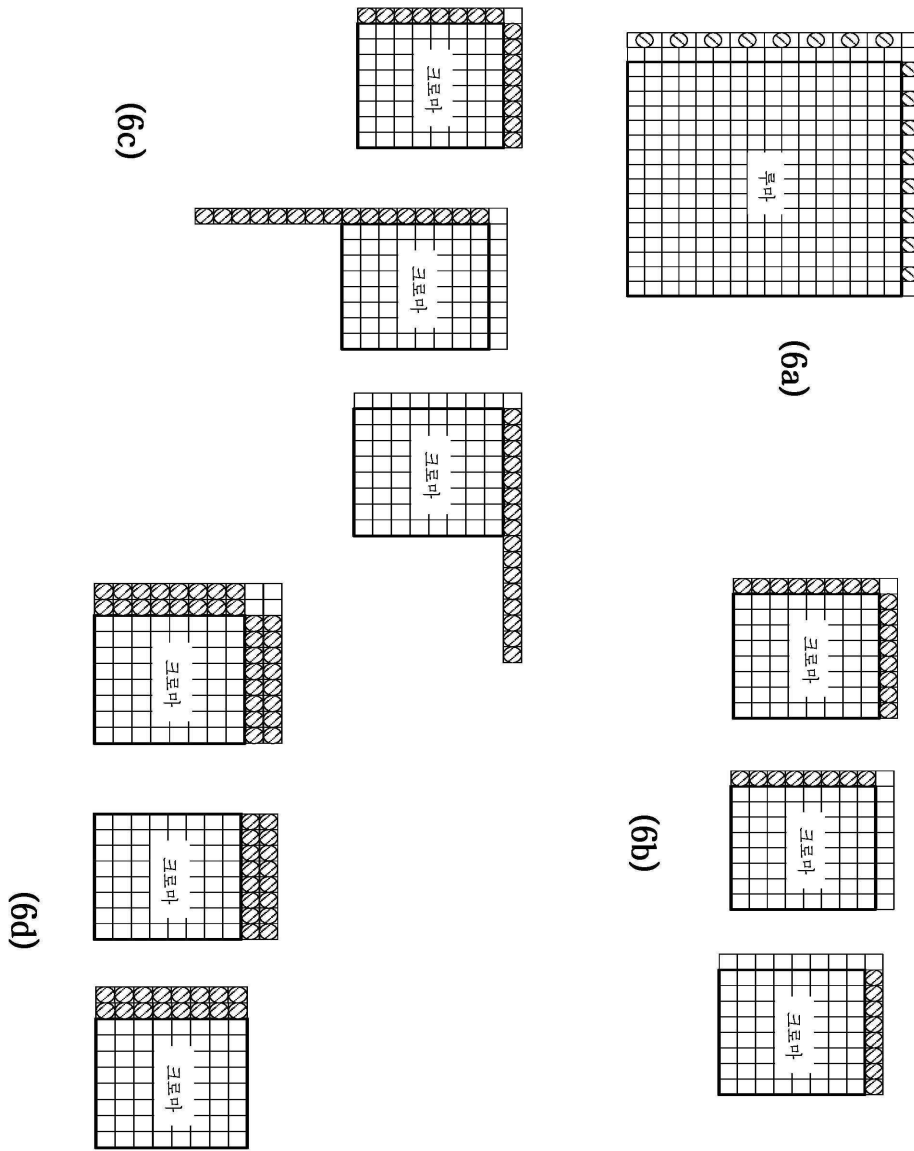
도면4



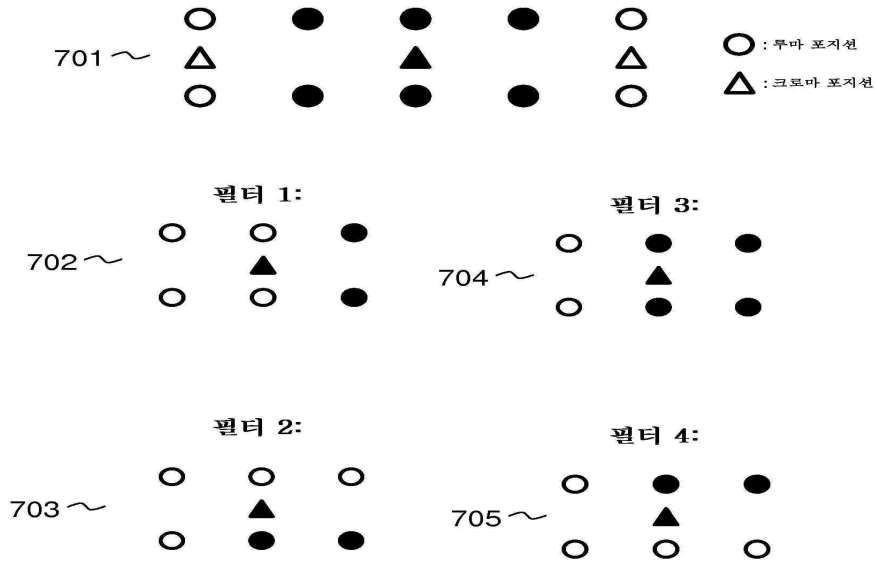
도면5



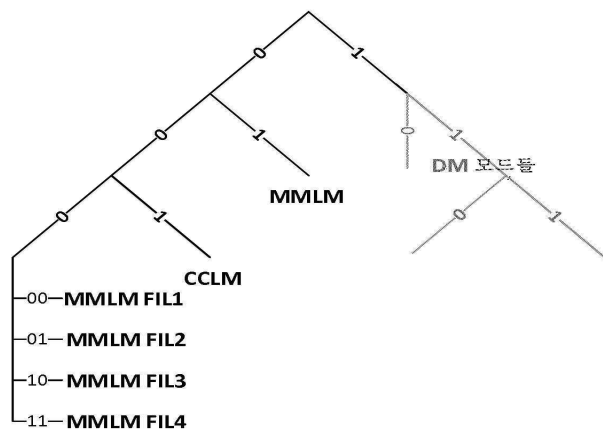
도면6



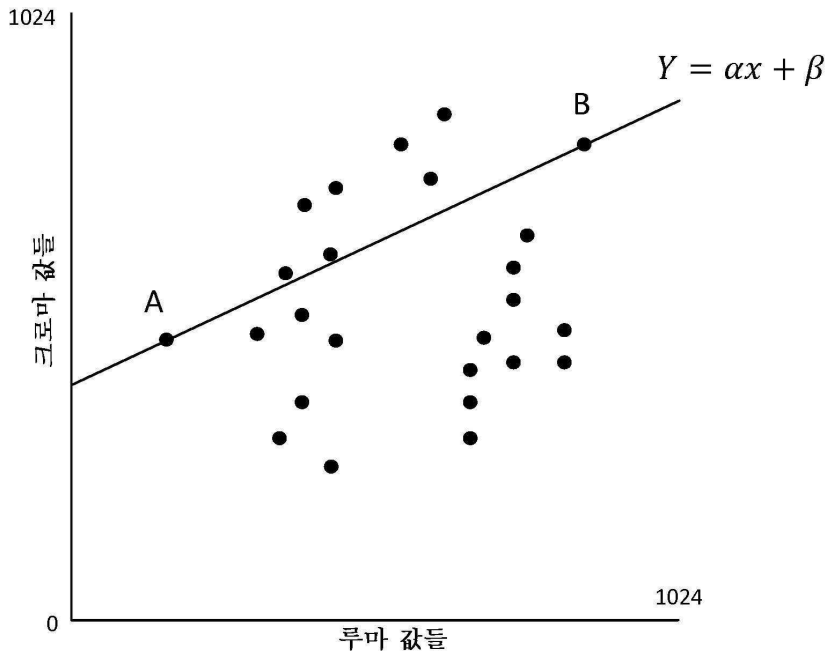
도면7



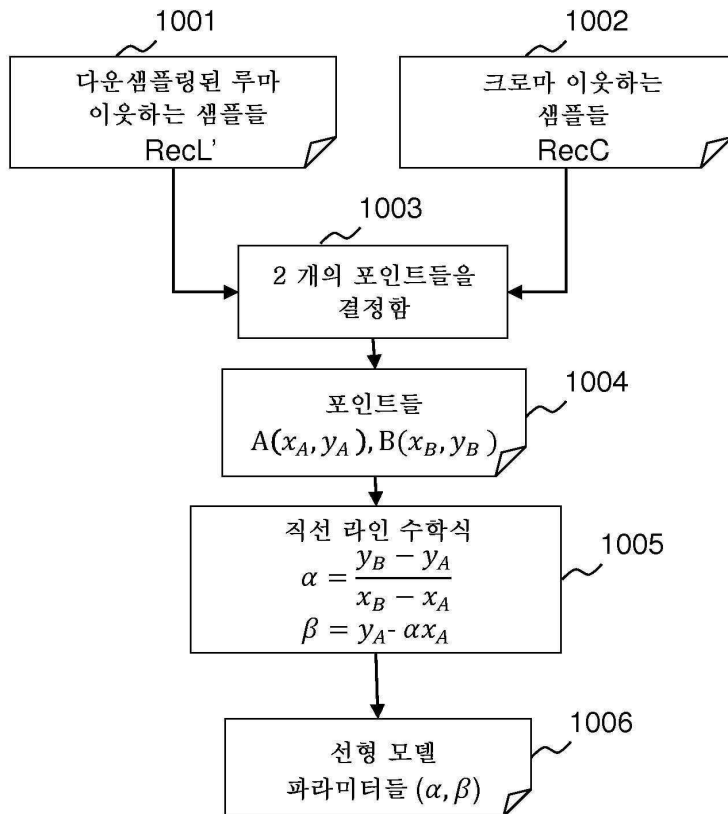
도면8



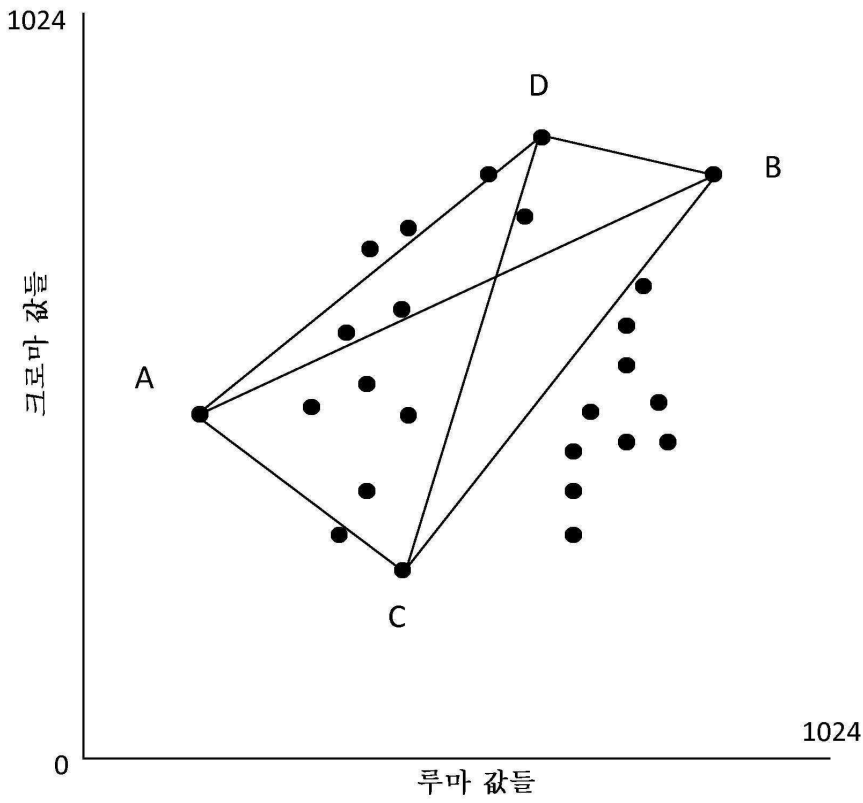
도면9



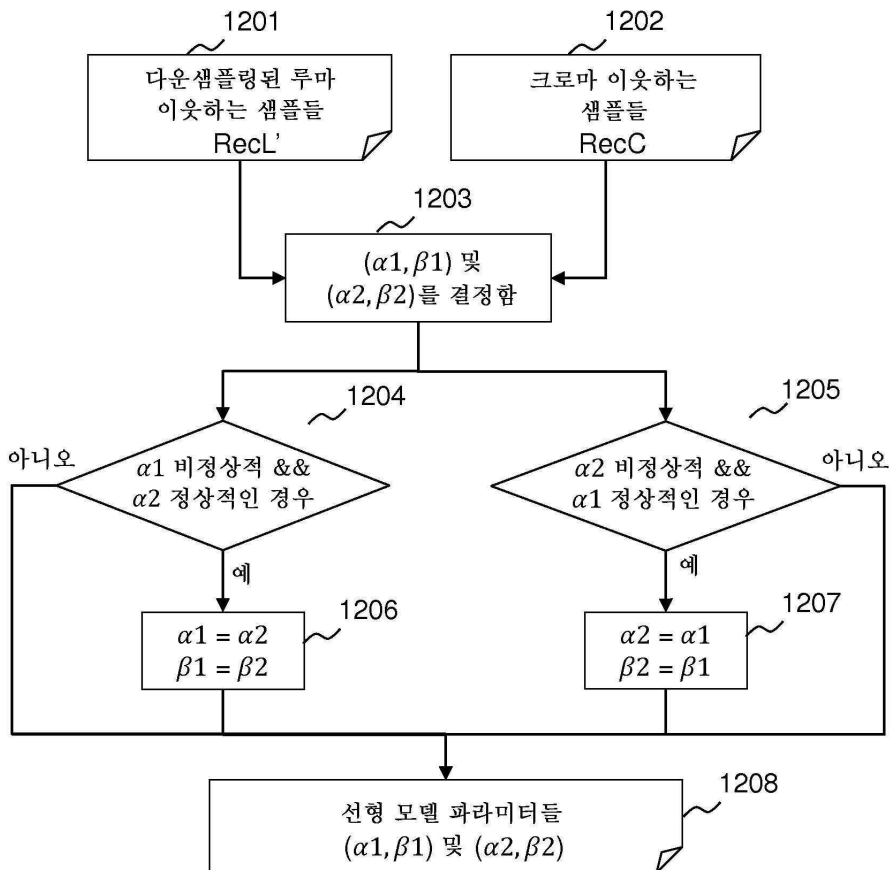
도면10



도면11



도면12



도면13

