



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2024년10월11일
(11) 등록번호 10-2715060
(24) 등록일자 2024년10월02일

- (51) 국제특허분류(Int. Cl.)
H04N 19/117 (2014.01) H04N 19/124 (2014.01)
H04N 19/129 (2014.01) H04N 19/132 (2014.01)
H04N 19/176 (2014.01) H04N 19/186 (2014.01)
- (52) CPC특허분류
H04N 19/117 (2015.01)
H04N 19/124 (2015.01)
- (21) 출원번호 10-2024-7016411(분할)
- (22) 출원일자(국제) 2019년12월11일
심사청구일자 2024년05월17일
- (85) 번역문제출일자 2024년05월17일
- (65) 공개번호 10-2024-0090457
- (43) 공개일자 2024년06월21일
- (62) 원출원 특허 10-2023-7040842
원출원일자(국제) 2019년12월11일
심사청구일자 2023년12월27일
- (86) 국제출원번호 PCT/US2019/065627
- (87) 국제공개번호 WO 2020/131512
국제공개일자 2020년06월25일
- (30) 우선권주장
62/781,316 2018년12월18일 미국(US)
(뒷면에 계속)
- (56) 선행기술조사문헌
JP2015008341 A*
JP2018174567 A*
US20180063527 A1*
*는 심사관에 의하여 인용된 문헌

- (73) 특허권자
텐센트 아메리카 엘엘씨
미국 94306 캘리포니아주 팔로 알토 파크 블러바드 2747
- (72) 발명자
자오, 량
미국 94306 캘리포니아주 팔로 알토 파크 블러바드 2747 텐센트 아메리카 엘엘씨 내
자오, 신
미국 94306 캘리포니아주 팔로 알토 파크 블러바드 2747 텐센트 아메리카 엘엘씨 내
(뒷면에 계속)
- (74) 대리인
양영준, 김연송, 백만기

전체 청구항 수 : 총 15 항

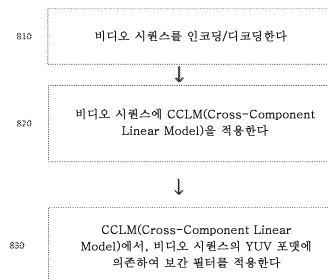
심사관 : 김건우

(54) 발명의 명칭 비디오 인코딩 또는 디코딩을 위한 방법 및 장치

(57) 요약

비디오 시퀀스를 인코딩 또는 디코딩하기 위한 방법 및 장치는 CCLM(Cross-Component Linear Model)을 비디오 시퀀스에 적용하는 것, 및 CCLM에서 보간 필터를 적용하는 것을 포함하고, 보간 필터는 비디오 시퀀스의 YUV 포맷에 의존한다.

대표도 - 도13



(52) CPC특허분류

HO4N 19/129 (2015.01)

HO4N 19/132 (2015.01)

HO4N 19/176 (2015.01)

HO4N 19/186 (2015.01)

(72) 발명자

리, 상

미국 94306 캘리포니아주 팔로 알토 파크 블러바드
2747 텐센트 아메리카 엘엘씨 내

류, 산

미국 94306 캘리포니아주 팔로 알토 파크 블러바드
2747 텐센트 아메리카 엘엘씨 내

(30) 우선권주장

62/785,678 2018년12월27일 미국(US)

62/788,729 2019년01월04일 미국(US)

62/789,992 2019년01월08일 미국(US)

16/523,258 2019년07월26일 미국(US)

명세서

청구범위

청구항 1

비디오 시퀀스를 디코딩하기 위한 디바이스로서,

프로그램 코드를 저장하도록 구성된 적어도 하나의 메모리;

상기 프로그램 코드를 판독하고 상기 프로그램 코드에 의해 지시된 대로 동작하도록 구성된 적어도 하나의 프로세서

를 포함하고, 상기 프로그램 코드는:

상기 적어도 하나의 프로세서로 하여금 코딩된 비디오 시퀀스를 수신하고, 보간 필터를 상기 코딩된 비디오 시퀀스에 적용하게 하도록 구성된 제1 디코딩 코드;

상기 적어도 하나의 프로세서로 하여금 루마 블록 중에서 상기 코딩된 비디오 시퀀스에 관련된 제1 블록의 복수의 이웃 샘플의 최대 값과 최소 값 사이의 차이를 획득하게 하도록 구성된 제1 획득 코드;

상기 적어도 하나의 프로세서로 하여금 상기 차이를 구간들로 분할하게 하도록 구성된 분할 코드;

상기 적어도 하나의 프로세서로 하여금 상기 차이의 양자화를 수행하게 하도록 구성된 제1 수행 코드;

상기 적어도 하나의 프로세서로 하여금 상기 양자화가 수행된 차이에 기초하여 플로어 값(floor value)을 획득하게 하도록 구성된 제2 획득 코드;

상기 적어도 하나의 프로세서로 하여금 적어도 상기 차이 및 상기 구간들을 사용하여 도출된 값을 획득하게 하도록 구성된 제3 획득 코드; 및

상기 적어도 하나의 프로세서로 하여금 상기 도출된 값에 기초하여, 상기 코딩된 비디오 시퀀스에 관련된 크로마 블록 중 상이한 블록의 샘플을 예측하게 하도록 구성된 예측 코드를 포함하는, 디바이스.

청구항 2

제1항에 있어서,

상기 제1 디코딩 코드는 상기 적어도 하나의 프로세서로 하여금 적용되는 상기 보간 필터의 탭들을 사용하게 하도록 구성된 코드를 추가로 포함하는, 디바이스.

청구항 3

제1항에 있어서,

상기 제1 디코딩 코드는 상기 적어도 하나의 프로세서로 하여금 상기 코딩된 비디오 시퀀스가 4:4:4: 또는 4:2:2의 YUV 포맷을 포함할 때 상기 보간 필터의 포맷을 상기 코딩된 비디오 시퀀스의 포맷에 대해 동일하게 설정하고, 상기 코딩된 비디오 시퀀스가 4:2:0의 YUV 포맷을 포함할 때 상기 보간 필터의 포맷을 상기 코딩된 비디오 시퀀스의 포맷과 상이하게 설정하게 하도록 구성된 코드를 추가로 포함하는, 디바이스.

청구항 4

제2항에 있어서,

상기 제1 디코딩 코드는 상기 보간 필터의 탭들이 상기 코딩된 비디오 시퀀스의 YUV 포맷과 동일한 형태로 되도록 구성되는, 디바이스.

청구항 5

제1항에 있어서,

상기 제1 디코딩 코드는 상기 적어도 하나의 프로세서로 하여금 상기 코딩된 비디오 시퀀스의 상이한 YUV 포맷

들에 대해 상이한 상기 보간 필터의 탭들을 사용하게 하도록 구성된 코드를 추가로 포함하는, 디바이스.

청구항 6

제1항에 있어서,

상기 제1 디코딩 코드는 상기 보간 필터를 상부 및 좌측 이웃 루마 재구성된 샘플들에 대해 상이하게 설정함으로써 상기 보간 필터를 적용하도록 구성되는, 디바이스.

청구항 7

제6항에 있어서,

상기 제1 디코딩 코드는 상기 보간 필터를 상기 상부 및 좌측 이웃 루마 재구성된 샘플들에 적용하여, 상기 보간 필터가 상기 코딩된 비디오 시퀀스의 YUV 포맷에 의존하도록 구성되는, 디바이스.

청구항 8

제1항에 있어서,

상기 제1 디코딩 코드는 상기 적어도 하나의 프로세서로 하여금 상부 이웃 루마 샘플들 내의 라인들의 수 및 좌측 이웃 루마 샘플들 내의 열들의 수를 상기 코딩된 비디오 시퀀스의 YUV 포맷에 의존하도록 설정하게 하도록 추가로 구성되는, 디바이스.

청구항 9

제8항에 있어서,

상기 제1 디코딩 코드는 상기 적어도 하나의 프로세서로 하여금 4:4:4 또는 4:2:2의 YUV 포맷 중 하나를 갖는 비디오 시퀀스들에 대해 상부 이웃 영역 내의 하나의 행 및 좌측 이웃 영역 내의 하나의 열 중 적어도 하나를 사용하게 하도록 추가로 구성되는, 디바이스.

청구항 10

제8항에 있어서,

상기 제1 디코딩 코드는 상기 적어도 하나의 프로세서로 하여금 4:2:2의 YUV 포맷을 갖는 비디오 시퀀스들에 대해 상부 이웃 영역 내의 하나의 행 및 좌측 이웃 영역 내의 적어도 2개의 열 중 적어도 하나를 사용하게 하도록 추가로 구성되는, 디바이스.

청구항 11

제1항에 있어서,

상기 제1 획득 코드는 상기 적어도 하나의 프로세서로 하여금, 상기 루마 블록 및 상기 크로마 블록의 N개의 이웃 샘플 쌍을 사용하여 상기 최대값 및 상기 최소 값을 획득하게 하도록 추가로 구성되고, N은 4, 8 및 16 중 하나의 양의 정수이고,

상기 N개의 이웃 샘플 쌍 각각은 상기 루마 블록 및 상기 크로마 블록 중 제1 블록에 이웃하는 제1 위치에 있는 제1 이웃 샘플, 및 상기 루마 블록 및 상기 크로마 블록 중 제2 블록에 이웃하고 상기 루마 블록 및 상기 크로마 블록 중 상기 제1 블록에 이웃하는 상기 제1 위치에 대응하는 제2 위치에 있는 제2 이웃 샘플을 포함하는, 디바이스.

청구항 12

제11항에 있어서,

상기 제1 획득 코드는 상기 적어도 하나의 프로세서로 하여금 상기 복수의 이웃 샘플을 하부에서 상부로의 순서 및 우측에서 좌측으로의 순서 중 적어도 하나에 의해 스캐닝함으로써 상기 제1 이웃 샘플을 선택하게 하도록 추가로 구성되는, 디바이스.

청구항 13

제1항에 있어서, 상기 프로그램 코드는,

상기 적어도 하나의 프로세서로 하여금, 현재 블록과 참조 블록의 N개의 이웃 샘플 쌍을 이용하여, 상기 현재 블록의 로컬 조명 보상(LIC)을 위한 선형 모델의 오프셋 및 스케일링 인자를 획득하게 하도록 구성된 제4 획득 코드 - N은 4, 8 및 16 중 하나의 양의 정수이고, 상기 N개의 이웃 샘플 쌍 각각은 상기 현재 블록에 이웃하는 제1 위치에서의 제1 이웃 샘플, 및 상기 참조 블록에 이웃하고 상기 현재 블록에 이웃하는 상기 제1 위치에 대응하는 제2 위치에서의 제2 이웃 샘플을 포함함 -; 및

상기 적어도 하나의 프로세서로 하여금 상기 획득된 스케일링 인자 및 상기 획득된 오프셋을 이용하여 상기 현재 블록의 로컬 조명 보상(LIC)을 수행하게 하도록 구성된 제2 수행 코드를 포함하는 디바이스.

청구항 14

비디오 시퀀스를 디코딩하는 방법으로서,

코딩된 비디오 시퀀스를 수신하고, 상기 코딩된 비디오 시퀀스에 보간 필터를 적용하는 단계;

루마 블록 중에서 상기 코딩된 비디오 시퀀스에 관련된 제1 블록의 복수의 이웃 샘플의 최대 값과 최소 값 사이의 차이를 획득하는 단계;

상기 차이를 구간들로 분할하는 단계;

상기 차이의 양자화를 수행하는 단계;

상기 양자화가 수행된 차이에 기초하여 플로어 값을 획득하는 단계;

적어도 상기 차이 및 상기 구간들을 사용하여 도출된 값을 획득하는 단계; 및

상기 도출된 값에 기초하여, 상기 코딩된 비디오 시퀀스에 관련된 크로마 블록 중 상이한 블록의 샘플을 예측하는 단계

를 포함하는, 방법.

청구항 15

제1항 내지 제13항 중 어느 한 항에 따른 프로그램 코드에 의해 디코딩된 비트스트림을 저장하는 비일시적 컴퓨터 판독가능 기록 매체.

발명의 설명

기술 분야

[0001] [관련 출원에 대한 상호 참조]

[0002] 본 출원은 미국특허상표청에서 2018년 12월 18일자로 출원된 미국 가출원 제62/781,316호, 2018년 12월 27일자로 출원된 미국 가출원 제62/785,678호, 2019년 1월 4일자로 출원된 미국 가출원 제62/788,729호, 2019년 1월 8일자로 출원된 미국 가출원 제62/789,992호, 및 2019년 7월 26일자로 출원된 미국 출원 제16/523,258호로부터 35 U.S.C. § 119 하에 우선권을 주장하며, 그 개시내용들은 그 전체가 본 명세서에 참고로 포함된다.

[0003] 실시예들에 따른 방법들 및 장치들은 비디오 처리에 관한 것으로, 특히, 크로스 컴포넌트 선형 모델(Cross Component Linear Model) 예측 모드들을 단순화하는 것에 초점을 맞추어 비디오 시퀀스를 인코딩 또는 디코딩하는 것에 관한 것이다.

배경 기술

[0004] 최근, ITU-T(International Telecommunication Standardization Sector)의 VCEG(Video Coding Experts Group), ITU(International Telecommunication Union)의 섹터, 및 ISO/IEC MPEG(JTC1/SC29/WG11), ISO(International Organization for Standardization) 및 IEC(International Electrotechnical Commission)의 ISO/IEC JTC(Joint Technical Committee)1의 표준화 서브커미티(standardization subcommittee)는 2013년에 H.265/HEVC(High Efficiency Video Coding) 표준(버전 1)을 발표하였다. 이 표준은 2014년에 버전 2로, 2015년에 버전 3으로, 그리고 2016년에 버전 4로 업데이트되었다.

[0005] 2017년 10월에, 이들은 HEVC를 뛰어넘는 능력을 갖는 비디오 압축에 대한 공동 제안 요청서(Call for Proposals, CfP)를 발행했다. 2018년 2월 15일에, 표준 동적 범위(standard dynamic range, SDR)에 대한 22개의 CfP 응답, 높은 동적 범위(high dynamic range, HDR)에 대한 12개의 CfP 응답, 및 360개의 비디오 카테고리 에 대한 12개의 CfP 응답이 제각기 제출되었다. 2018년 4월에, 모든 수신된 CfP 응답들은 122 MPEG/10번째 JVET 회의에서 평가되었다. 이 회의의 결과로서, JVET는 HEVC를 뛰어넘는 차세대 비디오 코딩의 표준화 프로세스를 공식적으로 개시했다. 새로운 표준은 다목적 비디오 코딩(Versatile Video Coding, VVC)으로 명명되었고, JVET는 공동 비디오 전문가 팀(Joint Video Expert Team)으로서 재명명되었다.

[0006] HEVC에서의 루마 성분들에 대한 인트라-예측 모드들이 이제 설명될 것이다. HEVC에서 사용되는 인트라-예측 모드들이 도 1에 예시되어 있다. HEVC에서, 총 35개의 인트라-예측 모드가 있을 수 있고, 그 중에서 모드 10은 수평 모드일 수 있고, 모드 26은 수직 모드일 수 있고, 모드 2, 18 및 34는 대각선 모드일 수 있다. 인트라-예측 모드들은 3개의 MPM(most probable mode) 및 32개의 나머지 모드에 의해 시그널링될 수 있다.

[0007] 이제, VVC에서의 루마 성분들에 대한 인트라-예측 모드들이 설명될 것이다. VVC의 현재 개발에서는, 도 2에 도시된 바와 같이 총 95개의 인트라-예측 모드가 있을 수 있으며, 여기서 모드 18은 수평 모드일 수 있고, 모드 50은 수직 모드일 수 있고, 모드들 2, 34 및 66은 대각선 모드일 수 있다. 모드들 -1 ~ -14 및 모드들 67 ~ 80은 WAIP(Wide-Angle Intra-Prediction) 모드들이라고 불릴 수 있다. 도 2에 도시된 바와 같이, HEVC에서는 35개의 인트라-예측 모드가 이용될 수 있다.

[0008] 이제, VVC에서의 루마 성분들에 대한 인트라-예측 모드들이 설명될 것이다. VVC의 현재 개발에서는, 도 2에 도시된 바와 같이 총 95개의 인트라-예측 모드가 있을 수 있으며, 여기서 모드 18은 수평 모드일 수 있고, 모드 50은 수직 모드일 수 있고, 모드들 2, 34 및 66은 대각선 모드일 수 있다. 모드들 -1 ~ -14 및 모드들 67 ~ 80은 WAIP(Wide-Angle Intra-Prediction) 모드들이라고 불릴 수 있다.

[0009] 이제, VVC에서의 크로마 성분들에 대한 인트라-모드 모드들이 설명될 것이다. VTM에서, 인트라-PU의 크로마 성분들에 대해, 인코더는 평면, DC, 수평, 수직, 루마 성분으로부터의 인트라-예측 모드(DM)의 직접 사본, LT_CCLM(Left and Top Cross-component Linear Mode), L_CCLM(Left Cross-component Linear Mode), 및 T_CCLM(Top Cross-component Linear Mode)를 포함하는 8개의 모드 중에서 최상의 크로마 예측 모드를 선택할 수 있다. LT_CCLM, L_CCLM 및 T_CCLM은 크로스 컴포넌트 선형 모드(Cross-component Linear Mode, CCLM)의 그룹으로 분류될 수 있다. 이러한 3개의 모드 사이의 차이는 이웃 샘플들의 상이한 영역들이 파라미터들 α 및 β 를 도출하기 위해 사용될 수 있다는 것이다. LT_CCLM에 대해, 왼쪽 및 상단 이웃 샘플들 둘 모두가 파라미터들 α 및 β 를 도출하기 위해 사용될 수 있다. L_CCLM에 대해, 일반적으로 왼쪽 이웃 샘플들만이 파라미터들 α 및 β 를 도출하기 위해 사용될 수 있다. T_CCLM에 대해, 일반적으로 상단 이웃 샘플들만이 파라미터들 α 및 β 를 도출하기 위해 사용될 수 있다.

[0010] 크로스 컴포넌트 선형 모델(Cross-Component Linear Model, CCLM) 예측 모드들은 크로스 컴포넌트 중복을 감소시키기 위하여 이용될 수도 있고, 여기서, 크로마 샘플들은 다음과 같이 선형 모델을 이용함으로써 동일한 CU의 재구성된 루마 샘플들에 기초하여 예측될 수 있다:

[0011]
$$pred_C(i, j) = \alpha \cdot rec_L'(i, j) + \beta$$

[0012] 여기서, $pred_C(i, j)$ 는 CU 내의 예측된 크로마 샘플들을 나타내고, $rec_L(i, j)$ 는 동일한 CU의 다운샘플링된 재구성된 루마 샘플들을 나타낸다. 파라미터들 α 및 β 는 최대-최소 방법(max-min method)이라고도 불리는 직선 방정식에 의해 도출될 수 있다. 이러한 계산 과정은 비단 인코더 검색 동작으로서뿐만 아니라 디코딩 프로세스의 일부로서 수행될 수 있으며, 따라서 α 및 β 값들을 전달하기 위해 어떤 신호가 반드시 사용되는 것은 아니다.

[0013] 도 3a 내지 도 3d에 도시된 상이한 YUV 포맷들이 있다. 4:2:0 포맷의 경우, LM 예측은 도 3a 내지 도 3d에 도시된 바와 같이 크로마 샘플에 대응하는 다운샘플링된 루마 샘플을 얻기 위해 6-탭 보간 필터를 적용할 수 있다. 공식적인 방식으로, 다운샘플링된 루마 샘플 $Rec'L[x, y]$ 는 재구성된 루마 샘플들로부터 다음과 같이 계산될 수 있다:

$$Rec'_L[x, y] = (2 \times Rec_L[2x, 2y] + 2 \times Rec_L[2x, 2y+1] + Rec_L[2x-1, 2y] + Rec_L[2x+1, 2y] + Rec_L[2x-1, 2y+1] + Rec_L[2x+1, 2y+1] + 4) \gg 3$$

[0014]

[0015] 다운샘플링된 루마 샘플들은 최대 및 최소 샘플 포인트들을 찾기 위해 사용될 수 있다. 2개의 포인트(루마 및 크로마의 커플)(A, B)는 도 4에 묘사된 바와 같은 이웃 루마 샘플들의 세트 내의 최소 및 최대 값들일 수 있다.

[0016] 여기서, 선형 모델 파라미터들 α 및 β 는 다음의 수학식에 따라 획득될 수 있다:

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

[0017]

$$\beta = y_A - \alpha x_A$$

[0018] 여기서, 나눗셈이 회피되고 곱셈 및 시프트로 대체될 수 있다. 하나의 룩업 테이블(LUT)이 미리 계산된 값들을 저장하기 위해 사용될 수 있고, 최대 및 최소 루마 샘플들 사이의 절대 차이 값들은 LUT의 엔트리 인덱스를 지정하기 위해 사용될 수 있고, LUT의 크기는 512일 수 있다.

[0019] diff_Y로 표시되는, 특정된 이웃 샘플 영역들 내의 최대 및 최소 루마 샘플 값들 사이의 절대 차이는 불균일 양자화될 수 있고, 절대 차이의 양자화된 값은 LUT의 크기가 감소되도록 CCLM 룩업 테이블(LUT)의 엔트리 인덱스를 지정하기 위해 사용될 수 있다는 것이 또한 제안되었다. diff_Y의 범위는 다중 구간으로 분할되고, 상이한 양자화 스텝 크기들이 상이한 구간들에서 사용될 수 있다. 일 예에서, diff_Y의 범위는 2개의 구간으로 분할될 수 있고, diff_Y가 Thres_1로 명명된 임계값 이하이면, Step_A로 명명된 하나의 스텝 크기가 사용된다. 그렇지 않으면, Step_B로 명명된 또 스텝 크기가 사용될 수 있다. 결과적으로, CCLM에서의 파라미터 a는 다음과 같이 획득될 수 있다:

$$a = (\text{diff}_Y > \text{Thres}_1) ? \text{LUT}\left[\frac{\text{Thres}_1}{\text{Step}_A} + \frac{\text{diff}_Y - \text{Thres}_1}{\text{Step}_B} - 1\right] : \text{LUT}[\text{diff}_Y/\text{Step}_A - 1]$$

[0020]

[0021] 여기서, Thres_1, Step_A, 및 Step_B는 1, 2, 3, 4 등등과 같은 임의의 양의 정수일 수 있다. 또한, Step_A와 Step_B는 동일하지 않다.

[0022] 크로마 예측자를 도출하기 위해, 현재 VTM 구현에 대해 말하자면, 곱셈은 다음과 같은 정수 연산으로 대체되며, 여기서 maxY, minY, maxC 및 minC는 제각기 최대 루마 샘플 값, 최소 루마 샘플 값, 최대 크로마 샘플 값, 및 최소 크로마 샘플 값을 나타낸다. numSampL 및 numSampT는 제각기 이용가능한 왼쪽 및 상단 이웃 샘플들의 수를 나타낸다. 다음의 텍스트들은 VVC 초안 3 조항 8.2.4.2.8로부터의 것이다.

[0023] 변수들 a, b, 및 k는 다음과 같이 도출된다:

[0024] numSampL이 0과 동일하고, numSampT가 0과 동일하면, 이하가 적용된다:

$$k = 0$$

$$a = 0$$

$$b = 1 \ll (\text{BitDepthC} - 1)$$

[0025]

[0026] 그렇지 않으면, 다음이 적용된다:

$$\text{shift} = (\text{BitDepthC} > 8) ? \text{BitDepthC} - 9 : 0$$

[0027]

$add = shift ? 1 \ll (shift - 1) : 0$

$diff = (maxY - minY + add) \gg shift$

$k = 16$

[0028]

[0029]

diff가 0보다 큰 경우, 다음이 적용된다:

$div = ((maxC - minC) * (Floor(2^{32} / diff) - Floor(2^{16} / diff) * 2^{16}) + 2^{15}) \gg$

16

[0030]

$a = ((maxC - minC) * Floor(2^{16} / diff) + div + add) \gg shift$

[0031]

그렇지 않으면, 다음이 적용된다:

$a = 0$

[0032]

$b = minC - ((a * minY) \gg k)$

[0033]

diff가 0보다 큰 경우에 대한 수학적식은 또한 다음과 같이 단순화될 수 있다:

[0034]

$a = ((maxC - minC) * Floor(2^{16} / diff) + add) \gg shift$

[0035]

파라미터들 a 및 b를 도출한 후에, 크로마 예측자는 다음과 같이 계산된다:

[0036]

$pred_C(i, j) = (a \cdot rec_L'(i, j)) \gg S + b$

[0037]

변수 "diff"의 범위는 1 ~ 512이므로, $Floor(2^{16} / diff)$ 의 값은 미리 계산되어 512와 동일한 크기를 갖는 룩업 테이블(LUT)에 저장될 수 있다. 또한, diff의 값은 룩업 테이블의 엔트리 인덱스를 지정하기 위해 사용된다.

[0038]

이러한 계산 과정은 비단 인코더 검색 동작으로서뿐만 아니라 디코딩 프로세스의 일부로서 수행되고, 따라서 α 및 β 값들을 전달하기 위해 어떠한 신택스도 사용되지 않는다.

[0039]

T_CCLM 모드에서, (2*W 샘플들을 포함하는) 상부 이웃 샘플들만이 선형 모델 계수들을 계산하기 위해 사용된다. L_CCLM 모드에서, (2*H 샘플들을 포함하는) 왼쪽 이웃 샘플들만이 선형 모델 계수들을 계산하기 위해 사용된다. 이는 도 6a 내지 도 7b에 예시된다.

[0040]

CCLM 예측 모드는 또한 2개의 크로마 성분 사이의 예측을 포함하는데, 즉, Cr 성분은 Cb 성분으로부터 예측될 수 있다. 재구성된 샘플 신호를 사용하는 대신에, CCLM Cb-대-Cr 예측이 잔차 도메인에서 적용될 수 있다. 이는 최종 Cr 예측을 형성하기 위해 원래의 Cr 인트라-예측에 가중 재구성된 Cb 잔차를 더함으로써 구현될 수 있다.

[0041]

$pred_{Cr}^*(i, j) = pred_{Cr}(i, j) + \alpha \cdot resi_{Cb}'(i, j)$

[0042]

CCLM 루마-대-크로마 예측 모드는 하나의 추가적인 크로마 인트라-예측 모드로서 추가될 수 있다. 인코더 측에서, 크로마 인트라-예측 모드를 선택하기 위해 크로마 성분들에 대한 하나 더의 RD 비용 검사가 추가될 수 있다. CCLM 루마-대-크로마 예측 모드 이외의 인트라-예측 모드들이 CU의 크로마 성분들을 위하여 이용될 때, CCLM Cb-대-Cr 예측은 Cr 성분 예측을 위하여 이용될 수 있다.

[0043]

다중 모델 CCLM(Multiple Model CCLM, MMLM)은 CCLM의 또 다른 확장이다. 이름으로 표시된 바와 같이, MMLM에는 2개의 모델과 같은 하나보다 많은 모델이 있을 수 있다. MMLM에서, 현재 블록의 이웃 루마 샘플들 및 이웃 크로마 샘플들은 2개의 그룹으로 분류될 수 있고, 각각의 그룹은 선형 모델을 도출하기 위한 훈련 세트로서 사용될 수 있다(즉, 특정 그룹에 대해 특정 α 및 β 가 도출된다). 더욱이, 현재 루마 블록의 샘플들은 또한 이웃 루마 샘플들의 분류에 대한 것과 동일한 규칙에 기초하여 분류될 수 있다.

[0044] 도 8은 이웃 샘플들을 2개의 그룹으로 분류하는 예를 도시한다. 여기서, 임계값은 이웃하는 재구성된 루마 샘플들의 평균 값으로서 계산될 수 있다. $Rec'_L[x,y] \leq$ 임계값인 이웃 샘플은 그룹 1로 분류될 수 있다; 한편 $Rec'_L[x,y] >$ 임계값인 이웃 샘플은 그룹 2로 분류될 수 있다.

$$\begin{cases} Pred_C[x,y] = \alpha_1 \times Rec'_L[x,y] + \beta_1 \text{ 만일 } Rec'_L[x,y] \leq \text{임계값} \\ Pred_C[x,y] = \alpha_2 \times Rec'_L[x,y] + \beta_2 \text{ 만일 } Rec'_L[x,y] > \text{임계값} \end{cases}$$

[0045] 로컬 조명 보상(Local Illumination Compensation, LIC)은, 스케일링 인자 a 및 오프셋 b를 사용하여, 조명 변화에 대한 선형 모델에 기초한다. 이는 각각의 인터-모드 코딩된 코딩 유닛(CU)에 대해 적응적으로 인에이블되거나 디스에이블된다.

[0046] CU에 대해 LIC를 적용할 때, 현재 CU의 이웃 샘플들 및 그들의 대응하는 참조 샘플들을 이용하여 파라미터들 a 및 b를 도출하기 위해 최소 제곱 오류 방법(least square error method)이 이용된다. 보다 구체적으로, 도 14에 도시된 바와 같이, (도 14의 부분 (a)에 도시된 바와 같은) 현재 CU(1410)의 서브샘플링된 (2:1 서브샘플링) 이웃 샘플들 및 (도 14의 부분 (b)에 도시된 바와 같은) 참조 픽처 또는 블록(1420)에서의 (현재 CU 또는 서브-CU의 모션 정보에 의해 식별된) 대응하는 샘플들이 이용된다. IC 파라미터들은 각각의 예측 방향에 대해 개별적으로 도출되고 적용된다.

[0047] CU가 병합 모드로 코딩될 때, 병합 모드에서의 모션임 정보 복사와 유사한 방식으로, LIC 플래그가 이웃 블록들로부터 복사되고; 그렇지 않으면, LIC가 적용되는지 여부를 표시하기 위해 LIC 플래그가 CU에 대해 시그널링된다.

[0048] 전술한 진보들에도 불구하고, 최신 기술에 의한 문제들이 존재한다. 현재, VTM3.0에서, 4:2:0 YUV 포맷으로, 6-탭 보간 필터는 특정된 이웃 영역에서의 모든 루마 샘플들에 적용될 수 있지만, 궁극적으로, 최대 및 최소값들을 갖는 오직 2개의 루마 샘플이 CCLM 파라미터들을 도출하기 위하여 이용되고, 이는 코딩 효율에서의 명확한 이익 없이 디코더 복잡도를 증가시킨다. 또한, VTM3.0에서, CCLM에서의 루마 샘플들에 대한 보간 필터는 4:2:0 YUV 포맷만을 지원하지만, 4:4:4 및 4:2:2 YUV 포맷은 여전히 인기가 있고 지원될 필요가 있다.

[0049] 현재 VTM3.0에서, CCLM에 대한 룩업 테이블(look-up table, LUT)의 크기는 512인데, 즉, 룩업 테이블은 512개의 요소를 가지고, 각각의 요소는 16-비트 정수에 의해 표현되고, 이는 코딩 효율에서의 분명한 이익 없이 너무 많은 디코더 메모리 비용을 증가시킨다. 또한, CCLM 및 LIC의 개념들이 유사하지만, 이들은 선형 모델에 대한 파라미터들 a 및 b를 도출하기 위해 상이한 방법들을 사용하는데, 이는 바람직하지 않다.

[0050] 현재 VTM3.0에서, 32x32 크로마 블록들과 같은 큰 블록들에 대해, 지정된 이웃 영역에서의 64개의 샘플이 최대 및 최소값들을 계산하기 위해 사용되는데, 이는 코딩 효율에서의 분명한 이익 없이 디코더 복잡성을 증가시킨다.

[0051] 현재 VTM3.0에서, 64x64 블록들과 같은 큰 블록들에 대해, 지정된 이웃 영역에서의 64개의 샘플이 선형 모델 파라미터들 a 및 b를 계산하기 위해 사용되는데, 이는 코딩 효율에서의 분명한 이익 없이 디코더 복잡도를 증가시킨다.

발명의 내용

과제의 해결 수단

[0052] 본 개시내용의 양태에 따르면, 비디오 시퀀스를 인코딩 또는 디코딩하기 위한 방법은: CCLM(Cross-Component Linear Model)을 비디오 시퀀스에 적용하는 단계, 및 CCLM(Cross-Component Linear Model)에서 보간 필터를 적용하는 단계를 포함할 수 있고, 여기서 보간 필터는 비디오 시퀀스의 YUV 포맷에 의존할 수 있다.

[0053] 본 개시내용의 양태에 따르면, 앞서 기술된 방법에서, CCLM(Cross-Component Linear Model)에서 보간 필터를 적용할 시에, 방법은 비디오 시퀀스의 YUV 포맷에 의존하는 보간 필터의 탭들을 사용하는 단계를 추가로 포함한다.

[0054] 본 개시내용의 이런 양태에 따르면, 방법은 비디오 시퀀스의 YUV 포맷과 동일한 형태인, CCLM(Cross-Component Linear Model)에서 사용되는 보간 필터의 탭들을 사용하는 단계를 추가로 포함할 수 있다.

[0055] 본 개시내용의 양태에 따르면, 앞서 기술된 방법에서, CCLM(Cross-Component Linear Model)에서 보간 필터를

적용할 시에, 방법은 비디오 시퀀스가 4:4:4: 또는 4:2:2의 YUV 포맷을 포함할 때 보간 필터의 포맷을 비디오 시퀀스의 포맷에 대해 동일하게 설정하는 단계를 추가로 포함할 수 있고, 비디오 시퀀스가 4:2:0의 YUV 포맷을 포함할 때 보간 필터의 포맷을 비디오 시퀀스의 포맷과 상이하게 설정하는 단계를 포함할 수 있다.

- [0057] 본 개시내용의 양태에 따르면, 앞서 기술된 방법에서, CCLM(Cross-Component Linear Model)에서 보간 필터를 적용할 시에, 방법은 비디오 시퀀스의 상이한 YUV 포맷들에 대해 상이한 보간 필터의 탭들을 사용하는 단계를 추가로 포함할 수 있다.
- [0058] 본 개시내용의 양태에 따르면, 앞서 기술된 방법에서, CCLM(Cross-Component Linear Model)에서 보간 필터를 적용할 시에, 방법은 보간 필터를 상단 및 왼쪽 이웃 루마 재구성된 샘플들에 대해 상이하도록 설정하는 단계를 추가로 포함할 수 있다.
- [0059] 본 개시내용의 이런 양태에 따르면, 방법은 상단 및 왼쪽 이웃 루마 재구성된 샘플들에 적용되는 보간 필터를 비디오 시퀀스의 YUV 포맷에 의존하도록 설정하는 단계를 추가로 포함할 수 있다.
- [0060] 본 개시내용의 양태에 따르면, 방법은 CCLM(Cross-Component Linear Model)에서 사용되는 상단 이웃 루마 샘플 들에서의 라인들의 수 및 왼쪽 이웃 루마 샘플들에서의 열들의 수를 비디오 시퀀스의 YUV 포맷에 의존하도록 설정하는 단계를 추가로 포함할 수 있다.
- [0061] 본 개시내용의 이런 양태에 따르면, 방법은 4:4:4 또는 4:2:2의 YUV 포맷 중 하나를 갖는 비디오 시퀀스들에 대해 CCLM(Cross-Component Linear Model)에서 상단 이웃 영역에서의 하나의 행 또는/및 왼쪽 이웃 영역에서의 하나의 열을 사용하는 단계를 추가로 포함할 수 있다.
- [0062] 본 개시내용의 이런 양태에 따르면, 방법은 4:2:2의 YUV 포맷을 갖는 비디오 시퀀스들에 대해 CCLM(Cross-Component Linear Model)에서 상단 이웃 영역에서의 하나의 행 및/또는 왼쪽 이웃 영역에서의 적어도 2개의 열 을 사용하는 단계를 추가로 포함할 수 있다.
- [0063] 본 개시내용의 양태에 따르면, 비디오 시퀀스를 인코딩 또는 디코딩하기 위한 디바이스는: 프로그램 코드를 저장하도록 구성된 적어도 하나의 메모리; 프로그램 코드를 판독하고 프로그램 코드에 의해 지시된 대로 동작하도록 구성된 적어도 하나의 프로세서를 포함할 수 있고, 여기서 프로그램 코드는: 적어도 하나의 프로세서로 하여금 CCLM(Cross-Component Linear Model)을 비디오 시퀀스에 적용하고 CCLM(Cross-Component Linear Model)에서 보간 필터를 적용하게 야기하도록 구성된 제1 인코딩 또는 디코딩 코드를 포함할 수 있고, 보간 필터는 비디오 시퀀스의 YUV 포맷에 의존할 수 있다.
- [0064] 본 개시내용의 양태에 따르면, 제1 인코딩 또는 디코딩 코드는 적어도 하나의 프로세서로 하여금: 비디오 시퀀스의 YUV 포맷에 의존하는, CCLM(Cross-Component Linear Model)에서, 적용되는 보간 필터의 탭들을 사용하게 야기하도록 구성되는 코드를 추가로 포함할 수 있다.
- [0065] 본 개시내용의 이런 양태에 따르면, 제1 인코딩 또는 디코딩 코드는 적어도 하나의 프로세서로 하여금: 비디오 시퀀스의 YUV 포맷과 동일한 형태인, CCLM(Cross-Component Linear Model)에서, 적용되는 보간 필터의 탭들을 사용하게 야기하도록 구성되는 코드를 추가로 포함할 수 있다.
- [0066] 본 개시내용의 양태에 따르면, 제1 인코딩 또는 디코딩 코드는 적어도 하나의 프로세서로 하여금: CCLM(Cross-Component Linear Model)에서 적용되는 보간 필터의 포맷을, 비디오 시퀀스가 4:4:4: 또는 4:2:2의 YUV 포맷을 포함할 때 비디오 시퀀스의 포맷에 대해 동일하게 설정하게 야기하고, 보간 필터의 포맷을, 비디오 시퀀스가 4:2:0의 YUV 포맷을 포함할 때 비디오 시퀀스의 포맷과 상이하게 설정하게 야기하도록 구성되는 코드를 추가로 포함할 수 있다.
- [0067] 본 개시내용의 양태에 따르면, 제1 인코딩 또는 디코딩 코드는 적어도 하나의 프로세서로 하여금: 비디오 시퀀스의 상이한 YUV 포맷들에 대해 상이한, CCLM(Cross-Component Linear Model)에서 적용되는 보간 필터의 탭들 을 사용하게 야기하도록 구성되는 코드를 추가로 포함할 수 있다.
- [0068] 본 개시내용의 양태에 따르면, 제1 인코딩 또는 디코딩 코드는 적어도 하나의 프로세서로 하여금: CCLM(Cross-Component Linear Model)에서 적용되는 보간 필터를, 상단 및 왼쪽 이웃 루마 재구성된 샘플들에 대해 상이하도록 설정하게 야기하도록 구성되는 코드를 추가로 포함할 수 있다.
- [0069] 본 개시내용의 이런 양태에 따르면, 제1 인코딩 또는 디코딩 코드는 적어도 하나의 프로세서로 하여금: 상단 및 왼쪽 이웃 루마 재구성된 샘플들에 적용되는 보간 필터를 비디오 시퀀스의 YUV 포맷에 의존하도록 설정하게 야

기하도록 구성되는 코드를 추가로 포함할 수 있다.

[0070] 본 개시내용의 양태에 따르면, 제1 인코딩 또는 디코딩 코드는 적어도 하나의 프로세서로 하여금: CCLM(Cross-Component Linear Model)에서 사용되는 상단 이웃 루마 샘플들에서의 라인들의 수 및 왼쪽 이웃 루마 샘플들에서의 열들의 수를 비디오 시퀀스의 YUV 포맷에 의존하도록 설정하게 야기하도록 구성되는 코드를 추가로 포함할 수 있다.

[0071] 본 개시내용의 이런 양태에 따르면, 제1 인코딩 또는 디코딩 코드는 적어도 하나의 프로세서로 하여금: 4:4:4 또는 4:2:2의 YUV 포맷 중 하나를 갖는 비디오 시퀀스들에 대해 CCLM(Cross-Component Linear Model)에서 상단 이웃 영역에서의 하나의 행 또는/및 왼쪽 이웃 영역에서의 적어도 하나의 열을 사용하게 야기하도록 구성되는 코드를 추가로 포함할 수 있다.

[0072] 본 개시내용의 양태에 따르면, 프로그램 코드를 저장하는 비일시적 컴퓨터 판독가능 매체가 제공될 수 있고, 프로그램 코드는, 디바이스의 하나 이상의 프로세서에 의해 실행될 때, 하나 이상의 프로세서로 하여금: CCLM(Cross-Component Linear Model)을 비디오 시퀀스에 적용하고, CCLM에서 보간 필터를 적용하게 야기할 수 있는 하나 이상의 명령어를 포함하고, 여기서 보간 필터는 비디오 시퀀스의 YUV 포맷에 의존한다.

[0073] 앞서 기술된 방법들, 디바이스들, 및 비일시적 컴퓨터 판독가능 매체들이 개별적으로 기술되었지만, 이들 설명은 사용 또는 그 기능성의 범위에 대해 임의의 제한을 시사하도록 의도되지 않는다. 실제로, 이러한 방법들, 디바이스들, 및 비일시적 컴퓨터 판독가능 매체들은 본 개시내용의 다른 양태들에서 조합될 수 있다.

도면의 간단한 설명

[0074] 개시된 주제의 추가의 특징들, 본질, 및 다양한 이점들이 다음의 상세한 설명 및 첨부 도면들로부터 더 명백할 것이다.

- 도 1은 일 실시예에 따른 예측 모델의 도면이다.
- 도 2는 일 실시예에 따른 예측 모델의 도면이다.
- 도 3a는 일 실시예에 따른 YUV 포맷의 도면이다.
- 도 3b는 일 실시예에 따른 YUV 포맷의 도면이다.
- 도 3c는 일 실시예에 따른 YUV 포맷의 도면이다.
- 도 3d는 일 실시예에 따른 YUV 포맷의 도면이다.
- 도 4는 일 실시예에 따른 상이한 루마 값들의 도면이다.
- 도 5a는 일 실시예에 따른 크로스 컴포넌트 선형 모델링에 이용되는 샘플들의 도면이다.
- 도 5b는 일 실시예에 따른 크로스 컴포넌트 선형 모델링에 이용되는 샘플들의 도면이다.
- 도 6a는 일 실시예에 따른 크로스 컴포넌트 선형 모델링에 이용되는 샘플들의 도면이다.
- 도 6b는 일 실시예에 따른 크로스 컴포넌트 선형 모델링에 이용되는 샘플들의 도면이다.
- 도 7a는 일 실시예에 따른 크로스 컴포넌트 선형 모델링에 이용되는 샘플들의 도면이다.
- 도 7b는 일 실시예에 따른 크로스 컴포넌트 선형 모델링에 이용되는 샘플들의 도면이다.
- 도 8은 일 실시예에 따른 다중 모델 CCLM을 이용한 분류의 예이다.
- 도 9는 일 실시예에 따른 통신 시스템의 단순화된 블록도이다.
- 도 10은 일 실시예에 따른 스트리밍 환경의 도면이다.
- 도 11은 일 실시예에 따른 비디오 디코더의 블록도이다.
- 도 12는 일 실시예에 따른 비디오 인코더의 블록도이다.
- 도 13은 일 실시예에 따른 비디오 시퀀스를 인코딩 또는 디코딩하기 위한 예시적인 프로세스의 흐름도이다.
- 도 14는 조명 보상(IC) 파라미터를 도출하는데 이용되는 이웃 샘플들의 도면이다.

도 15, 도 16, 도 17, 도 18, 도 19, 도 20, 도 21, 도 22, 도 23 및 도 24는 실시예들에 따른, 최대 및 최소 샘플 값들을 계산하기 위해 이용되는 현재 코딩 유닛(CU) 및 현재 CU의 이웃 재구성된 샘플들의 서브세트의 도면들이다.

도 25 및 도 26은 실시예들에 따른, 최대 및 최소 샘플 값들을 계산하기 위해 이용되는 크로마 블록 및 크로마 블록의 선택된 이웃 샘플들의 도면들이다.

도 27 및 도 28은 실시예들에 따른 루마 샘플들에 대한 필터들의 위치들의 도면들이다.

도 29 및 도 30은 실시예들에 따른, 선형 모델 예측 모드에서 파라미터들을 계산하기 위해 이용되는 현재 CU 및 현재 CU의 이웃 샘플 쌍의 도면들이다.

도 31, 도 32, 도 33, 도 34, 도 35 및 도 36은 실시예들에 따른, 선형 모델 파라미터들을 계산하기 위해 이용되는 블록 및 블록의 선택된 이웃 샘플들의 도면들이다.

도 37은 일 실시예에 따른 컴퓨터 시스템의 도면이다.

발명을 실시하기 위한 구체적인 내용

[0075] 도 9는 본 개시내용의 실시예에 따른 통신 시스템(400)의 단순화된 블록도를 도시한다. 통신 시스템(400)은 네트워크(450)를 통해 상호접속된 적어도 2개의 단말(410-420)을 포함할 수 있다. 데이터의 단방향 송신을 위해, 제1 단말(410)은 네트워크(450)를 통해 다른 단말(420)로의 송신을 위해 로컬 위치에서 비디오 데이터를 코딩할 수 있다. 제2 단말(420)은 네트워크(450)로부터 다른 단말의 코딩된 비디오 데이터를 수신하고, 코딩된 데이터를 디코딩하고 복구된 비디오 데이터를 디스플레이할 수 있다. 단방향 데이터 송신은 미디어 서빙 응용들(media serving applications) 등에서 흔한 것일 수 있다.

[0076] 도 9는, 예를 들어, 영상 회의 동안 발생할 수 있는 코딩된 비디오의 양방향 송신을 지원하기 위해 제공되는 제 2 쌍의 단말들(430, 440)을 도시한다. 데이터의 양방향 송신을 위해, 각각의 단말(430, 440)은 네트워크(450)를 통해 다른 단말로의 송신을 위해 로컬 위치에서 캡처된 비디오 데이터를 코딩할 수 있다. 각각의 단말(430, 440)은 또한 다른 단말에 의해 송신된 코딩된 비디오 데이터를 수신할 수 있고, 코딩된 데이터를 디코딩할 수 있고, 복구된 비디오 데이터를 로컬 디스플레이 디바이스에 디스플레이할 수 있다.

[0077] 도 9에서, 단말들(410-440)은 서버들, 개인용 컴퓨터들 및 스마트폰들로서 예시될 수 있지만, 본 개시내용의 원리들은 그렇게만 제한되지는 않는다. 본 개시내용의 실시예들은 랩톱 컴퓨터들, 태블릿 컴퓨터들, 미디어 플레이어 및/또는 전용 영상 회의 장비를 이용한 응용을 찾는다. 네트워크(450)는 예를 들어, 와이어라인 및/또는 무선 통신 네트워크들을 포함하여, 단말들(410-440) 사이에 코딩된 비디오 데이터를 전달하는 임의의 수의 네트워크를 나타낸다. 통신 네트워크(450)는 회선 교환 및/또는 패킷 교환 채널들에서 데이터를 교환할 수 있다. 대표적인 네트워크들은 원거리통신(telecommunications) 네트워크들, 로컬 영역 네트워크들, 광역 네트워크들 및/또는 인터넷을 포함한다. 본 논의의 목적을 위해, 네트워크(450)의 아키텍처 및 토폴로지는 아래에서 본 명세서에서 설명되지 않는 한 본 개시내용의 동작에 중요하지 않을 수 있다.

[0078] 도 10은, 개시된 주제를 위한 응용에 대한 예로서, 스트리밍 환경에서의 비디오 인코더 및 디코더의 배치를 예시한다. 개시된 주제는, 예를 들어, 영상 회의, 디지털 TV, CD, DVD, 메모리 스틱 등을 포함하는 디지털 미디어 상의 압축된 비디오의 저장 등을 포함하여, 다른 비디오 인에이블 응용들에 동등하게 적용가능할 수 있다.

[0079] 스트리밍 시스템은, 예를 들어, 압축되지 않은 비디오 샘플 스트림(502)을 생성하는 비디오 소스(501), 예를 들어, 디지털 카메라를 포함할 수 있는 캡처 서브시스템(513)을 포함할 수 있다. 인코딩된 비디오 비트스트림들과 비교할 때 많은 데이터 용량을 강조하기 위해 굵은 라인으로 묘사된 샘플 스트림(502)은 카메라(501)에 결합된 인코더(503)에 의해 처리될 수 있다. 비디오 인코더(503)는 아래에서 더 상세히 설명되는 바와 같이 개시된 주제의 양태들을 가능하게 하거나 구현하기 위해 하드웨어, 소프트웨어, 또는 이들의 조합을 포함할 수 있다. 샘플 스트림과 비교할 때 적은 데이터 용량을 강조하기 위해 얇은 라인으로서 묘사된 인코딩된 비디오 비트스트림(504)은 미래의 사용을 위해 스트리밍 서버(505)상에 저장될 수 있다. 하나 이상의 스트리밍 클라이언트(506, 508)는 스트리밍 서버(505)에 액세스하여 인코딩된 비디오 비트스트림(504)의 사본들(507, 509)을 검색할 수 있다. 클라이언트(506)는, 인코딩된 비디오 비트스트림(507)의 착신 사본을 디코딩하고 디스플레이(512) 또는 다른 렌더링 디바이스(묘사되지 않음)상에 렌더링될 수 있는 발신 비디오 샘플 스트림(511)을 생성하는 비디오 디코더(510)를 포함할 수 있다. 일부 스트리밍 시스템들에서, 비디오 비트스트림들(504, 507, 509)은 특정 비디오 코딩/압축 표준들에 따라 인코딩될 수 있다. 이러한 표준들의 예들은 H.265 HEVC를 포함한다.

VVC(Versatile Video Coding)로서 비공식적으로 알려진 비디오 코딩 표준이 개발 중이다. 개시된 주제는 VVC의 맥락에서 사용될 수 있다.

- [0080] 도 11은 본 발명의 일 실시예에 따른 비디오 디코더(510)의 기능 블록도일 수 있다.
- [0081] 수신기(610)는 디코더(610)에 의해 디코딩될 하나 이상의 코덱 비디오 시퀀스를 수신할 수 있다; 동일한 또는 또 다른 실시예에서, 한 번에 하나의 코딩된 비디오 시퀀스 - 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(612)로부터 수신될 수 있다. 수신기(610)는 인코딩된 비디오 데이터를 다른 데이터, 예를 들어, 코딩된 오디오 데이터 및/또는 보조 데이터 스트림들과 함께 수신할 수 있고, 이들은 그것들 각각의 사용 엔티티들(묘사되지 않음)에 포워딩될 수 있다. 수신기(610)는 코딩된 비디오 시퀀스를 다른 데이터로부터 분리할 수 있다. 네트워크 지터를 방지하기 위해, 수신기(610)와 엔트로피 디코더/과서(620)(이후 "과서") 사이 내에 버퍼 메모리(615)가 결합될 수 있다. 수신기(610)가 충분한 대역폭 및 제어가능성의 저장/포워드 디바이스로부터, 또는 동시 동기식 네트워크(isosynchronous network)로부터 데이터를 수신하고 있을 때, 버퍼(615)는 필요하지 않을 수 있거나, 작을 수 있다. 인터넷과 같은 베스트 에포트 패킷 네트워크들상에서의 사용을 위해, 버퍼(615)가 요구될 수 있고, 비교적 클 수 있고, 유리하게는 적응적 크기일 수 있다.
- [0082] 비디오 디코더(510)는 엔트로피 코딩된 비디오 시퀀스로부터 심벌들(621)을 재구성하기 위해 과서(620)를 포함할 수 있다. 해당 심벌들의 카테고리들은 디코더(510)의 동작을 관리하기 위해 사용되는 정보, 및 잠재적으로, 도 11에 도시된 바와 같이, 디코더의 일체 부분(integral part)은 아니지만 디코더에 결합될 수 있는 디스플레이(512)와 같은 렌더링 디바이스를 제어하기 위한 정보를 포함한다. 렌더링 디바이스(들)을 위한 제어 정보는 SEI 메시지(Supplementary Enhancement Information) 또는 VUI(Video Usability Information) 파라미터 세트 프래그먼트들(묘사되지 않음)의 형식일 수 있다. 과서(620)는 수신되는 코딩된 비디오 시퀀스를 과싱/엔트로피 디코딩할 수 있다. 코딩된 비디오 시퀀스의 코딩은 비디오 코딩 기술 또는 표준에 따를 수 있고, 가변 길이 코딩, 허프만 코딩(Huffman coding), 맥락 민감성(context sensitivity)을 갖거나 갖지 않는 산술 코딩 등을 포함하는, 본 기술분야의 통상의 기술자에게 잘 알려진 원리들을 따를 수 있다. 과서(620)는, 코딩된 비디오 시퀀스로부터, 그룹에 대응하는 적어도 하나의 파라미터에 기초하여, 비디오 디코더에서의 픽셀들의 서브그룹들 중 적어도 하나에 대한 서브그룹 파라미터들의 세트를 추출할 수 있다. 서브그룹들은 픽처 그룹들(Groups of Pictures, GOP들), 픽처들, 타일들, 슬라이스들, 매크로블록들, 코딩 유닛들(Coding Units, CU들), 블록들, 변환 유닛들(Transform Units, TU들), 예측 유닛들(Prediction Units, PU들) 등을 포함할 수 있다. 엔트로피 디코더/과서는 또한 코딩된 비디오 시퀀스로부터 변환 계수들, 양자화기 파라미터(QP) 값들, 모션 벡터들 등과 같은 정보를 추출할 수 있다.
- [0083] 과서(620)는 버퍼(615)로부터 수신된 비디오 시퀀스에 대해 엔트로피 디코딩/과싱 동작을 수행하여, 심벌들(621)을 생성할 수 있다. 과서(620)는 인코딩된 데이터를 수신하고, 특정 심벌들(621)을 선택적으로 디코딩할 수 있다. 또한, 과서(620)는 특정 심벌들(621)이 모션 보상 예측 유닛(653), 스케일러/역변환 유닛(651), 인트라 예측 유닛(652), 또는 루프 필터 유닛(658)에 제공될지를 결정할 수 있다.
- [0084] 심벌들(621)의 재구성은 코딩된 비디오 픽처 또는 그 부분들의 타입(예컨대: 인터 및 인트라 픽처, 인터 및 인트라 블록), 및 다른 인자들에 의존하여 다중의 상이한 유닛을 수반할 수 있다. 어느 유닛들이 수반되는지, 그리고 어떻게 되는지는 과서(620)에 의해 코딩된 비디오 시퀀스로부터 과싱된 서브그룹 제어 정보에 의해 제어될 수 있다. 과서(620)와 아래의 다중의 유닛 사이의 그러한 서브그룹 제어 정보의 흐름은 명확성을 위해 묘사되어 있지 않다.
- [0085] 이미 언급된 기능 블록들 이외에, 디코더(510)는 아래에 설명되는 바와 같이 개념적으로 다수의 기능 유닛으로 세분될 수 있다. 상업적 제약 하에서 동작하는 실제 구현에서, 이들 유닛 중 다수는 서로 밀접하게 상호작용하고, 적어도 부분적으로 서로 통합될 수 있다. 그러나, 개시된 주제를 설명하기 위한 목적으로, 아래의 기능 유닛들로의 개념적 세분이 적절하다.
- [0086] 제1 유닛은 스케일러/역변환 유닛(651)이다. 스케일러/역변환 유닛(651)은, 과서(620)로부터의 심벌(들)(621)로서, 어느 변환을 사용할지, 블록 크기, 양자화 인자, 양자화 스케일링 행렬들 등을 포함하여, 제어 정보뿐만 아니라 양자화된 변환 계수를 수신한다. 이것은 집계기(aggregator)(655)에 입력될 수 있는 샘플 값들을 포함하는 블록들을 출력할 수 있다.

- [0087] 일부 경우들에서, 스케일러/역변환(651)의 출력 샘플들은 인트라-코딩된 블록에 관련될 수 있다; 즉: 이전에 재구성된 픽처들로부터의 예측 정보를 이용하는 것이 아니라, 현재 픽처의 이전에 재구성된 부분들로부터의 예측 정보를 이용할 수 있는 블록. 이러한 예측 정보는 인트라-픽처 예측 유닛(652)에 의해 제공될 수 있다. 일부 경우들에서, 인트라-픽처 예측 유닛(652)은 현재 (부분적으로 재구성된) 픽처(656)로부터 폐치된 주위의 이미 재구성된 정보를 사용하여, 재구성 중인 블록의 동일한 크기 및 형상의 블록을 생성한다. 집계기(655)는, 일부 경우들에서, 샘플당 기준으로, 인트라-예측 유닛(652)이 생성한 예측 정보를 스케일러/역변환 유닛(651)에 의해 제공된 출력 샘플 정보에 추가한다.
- [0088] 다른 경우들에서, 스케일러/역변환 유닛(651)의 출력 샘플들은 인트라 코딩되고, 잠재적으로 모션 보상된 블록에 관련될 수 있다. 그러한 경우에, 모션 보상 예측 유닛(653)은 참조 픽처 메모리(657)에 액세스하여 예측을 위해 사용되는 샘플들을 폐치할 수 있다. 블록에 관련된 심벌들(621)에 따라 폐치된 샘플들을 모션 보상을 한 후에, 이들 샘플은 집계기(655)에 의해 스케일러/역변환 유닛의 출력(이 경우 잔차 샘플들 또는 잔차 신호라고 불림)에 더해져서 출력 샘플 정보를 생성할 수 있다. 모션 보상 유닛이 그로부터 예측 샘플들을 폐치하는 참조 픽처 메모리 내의 어드레스들은, 예를 들어, X, Y, 및 참조 픽처 컴포넌트들을 가질 수 있는 심벌들(621)의 형식으로 모션 보상 유닛에 이용가능한 모션 벡터들에 의해 제어될 수 있다. 모션 보상은 또한 서브샘플 정확한 모션 벡터들이 사용 중인 때 참조 픽처 메모리로부터 폐치된 샘플 값들의 보간, 모션 벡터 예측 메커니즘 등을 포함할 수 있다.
- [0089] 집계기(655)의 출력 샘플들에 대해 루프 필터 유닛(658)에서의 다양한 루프 필터링 기법들이 수행될 수 있다. 비디오 압축 기술들은, 파서(620)로부터의 심벌들(621)로서 루프 필터 유닛(658)에 이용가능하게 되고 코딩된 비디오 비트스트림에 포함된 파라미터들에 의해 제어되지만, 코딩된 픽처 또는 코딩된 비디오 시퀀스의 이전(디코딩 순서로) 부분들의 디코딩 동안 획득된 메타-정보에 응답할 뿐만 아니라, 이전에 재구성된 및 루프-필터링된 샘플 값들에 응답할 수도 있는 인-루프 필터(in-loop filter) 기술들을 포함할 수 있다.
- [0090] 루프 필터 유닛(658)의 출력은 렌더링 디바이스(512)에 출력될 뿐만 아니라 미래의 인트라-픽처 예측에서 사용하기 위해 참조 픽처 메모리(656)에 저장될 수도 있는 샘플 스트림일 수 있다.
- [0091] 특정 코딩된 픽처들은, 일단 완전히 재구성되면, 미래 예측을 위한 참조 픽처들로서 사용될 수 있다. 일단 코딩된 픽처가 완전히 재구성되고 코딩된 픽처가 참조 픽처로서 식별되면(예를 들어, 파서(620)에 의해), 현재 참조 픽처(656)는 참조 픽처 버퍼(657)의 일부가 될 수 있고, 다음의 코딩된 픽처의 재구성을 개시하기 전에 새로운 현재 픽처 메모리가 재할당될 수 있다.
- [0092] 비디오 디코더(510)는 H.265 HEVC와 같은 표준에 문서화될 수 있는 미리 결정된 비디오 압축 기술에 따라 디코딩 동작들을 수행할 수 있다. 코딩된 비디오 시퀀스는, 이것이 비디오 압축 기술 문서 또는 표준 및 구체적으로 그 가운데 프로파일 문서에 특정된 대로 비디오 압축 기술 또는 표준의 선택스(syntax)를 고수한다는 점에서, 사용되는 비디오 압축 기술 또는 표준에 의해 특정된 구문을 준수할 수 있다. 또한 준수를 위해 필요한 것은 코딩된 비디오 시퀀스의 복잡성이 비디오 압축 기술 또는 표준의 레벨에 의해 정의된 경계 내에 있는 것일 수 있다. 일부 경우들에서, 레벨들은 최대 픽처 크기, 최대 프레임 레이트, 최대 재구성 샘플 레이트(예를 들어, 초당 메가샘플수로 측정됨), 최대 참조 픽처 크기 등을 제한한다. 레벨들에 의해 설정된 한계들은, 일부 경우들에서, HRD(Hypothetical Reference Decoder) 사양들 및 코딩된 비디오 시퀀스에서 시그널링된 HRD 버퍼 관리를 위한 메타데이터를 통해 추가로 제한될 수 있다.
- [0093] 일 실시예에서, 수신기(610)는 인코딩된 비디오와 함께 추가적인 (중복) 데이터를 수신할 수 있다. 이 추가적인 데이터는 코딩된 비디오 시퀀스(들)의 일부로서 포함될 수 있다. 이 추가적인 데이터는 데이터를 적절히 디코딩하고 및/또는 원래의 비디오 데이터를 더 정확하게 재구성하기 위해 비디오 디코더(510)에 의해 사용될 수 있다. 추가적인 데이터는 예를 들어, 시간, 공간, 또는 신호 대 잡음 비(SNR) 향상 계층들, 중복 슬라이스들, 중복 픽처들, 순방향 오류 정정 코드들 등의 형식일 수 있다.
- [0094] 도 12는 본 개시내용의 일 실시예에 따른 비디오 인코더(503)의 기능 블록도일 수 있다.
- [0095] 인코더(503)는 인코더(503)에 의해 코딩될 비디오 이미지(들)를 캡처할 수 있는 비디오 소스(501)(인코더의 일부가 아님)로부터 비디오 샘플들을 수신할 수 있다.
- [0096] 비디오 소스(501)는, 임의의 적합한 비트 심도(예를 들어: 8 비트, 10 비트, 12 비트, ...), 임의의 컬러 공간(예를 들어, BT.601Y 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 형식으로 인코더(503)에 의해 코딩될 소스 비디오 시퀀스를 제

공할 수 있다. 미디어 서빙 시스템에서, 비디오 소스(501)는 이전에 준비된 비디오를 저장하는 저장 디바이스 일 수 있다. 영상 회의 시스템에서, 비디오 소스(503)는 비디오 시퀀스로서 로컬 이미지 정보를 캡처하는 카메라 일 수 있다. 비디오 데이터는 순차적으로 볼 때 모션을 부여하는 복수의 개별 픽처로서 제공될 수 있다. 픽처들 자체는 픽셀들의 공간 어레이로서 조직될 수 있고, 여기서 각각의 픽셀은 사용 중인 샘플링 구조, 컬러 공간 등에 의존하여 하나 이상의 샘플을 포함할 수 있다. 본 기술분야의 통상의 기술자는 픽셀들과 샘플들 사이의 관계를 용이하게 이해할 수 있다. 이하의 설명은 샘플들에 초점을 맞춘다.

[0097] 일 실시예에 따르면, 인코더(503)는 소스 비디오 시퀀스의 픽처들을 실시간으로 또는 응용에 의해 요구되는 임의의 다른 시간 제약들 하에서 코딩된 비디오 시퀀스(743)로 코딩 및 압축할 수 있다. 적절한 코딩 속도를 시행하는 것이 컨트롤러(750)의 하나의 기능이다. 컨트롤러(750)는 후술하는 바와 같이 다른 기능 유닛들을 제어하고 이들 유닛들에 기능적으로 결합된다. 결합은 명료성을 위해 묘사되지 않는다. 컨트롤러에 의해 설정된 파라미터들은 레이트 제어 관련 파라미터들(픽처 스킵, 양자화기, 레이트-왜곡 최적화 기법들의 램다 값들 등), 픽처 크기, 픽처 그룹(GOP) 레이아웃, 최대 모션 벡터 검색 범위 등을 포함할 수 있다. 본 기술분야의 통상의 기술자는 컨트롤러(750)의 다른 기능들을 쉽게 식별할 수 있는데, 그 이유는 그것들이 특정 시스템 설계에 대해 최적화된 비디오 인코더(503)에 관한 것일 수 있기 때문이다.

[0098] 일부 비디오 인코더들은 본 기술분야의 통상의 기술자가 "코딩 루프"로서 쉽게 인식하는 것에서 동작한다. 과도하게 단순화된 설명으로서, 코딩 루프는 인코더(730)(이후 "소스 코더")의 인코딩 부분(코딩될 입력 픽처, 및 참조 픽처(들)에 기초하여 심벌들을 생성하는 것을 담당함), 및 (심벌들과 코딩된 비디오 비트 스트림 사이의 임의의 압축이 개시된 주제에서 고려되는 비디오 압축 기술들에서 무손실이기 때문에) (원격) 디코더가 또한 생성할 샘플 데이터를 생성하기 위해 심벌들을 재구성하는 인코더(503)에 임베드된 (로컬) 디코더(733)로 구성될 수 있다. 재구성된 샘플 스트림은 참조 픽처 메모리(734)에 입력된다. 심벌 스트림의 디코딩이 디코더 위치 (로컬 또는 원격)와는 독립적으로 비트-정확한 결과들(bit-exact results)을 야기하기 때문에, 참조 픽처 버퍼 콘텐츠는 또한 로컬 인코더와 원격 인코더 사이에서 비트 정확(bit exact)하다. 다시 말해서, 인코더의 예측 부분은 디코딩 동안 예측을 사용할 때 디코더가 "보는(see)" 것과 정확히 동일한 샘플 값들을 참조 픽처 샘플들로서 "본다(sees)". 참조 픽처 동기성(reference picture synchronicity)의 이러한 기본적인 원리(그리고, 예를 들어, 채널 오류들 때문에, 동기성이 유지될 수 없는 경우, 결과적인 드리프트)는 본 기술분야의 통상의 기술자에게 잘 알려져 있다.

[0099] "로컬" 디코더(733)의 동작은 도 10과 관련하여 위에서 이미 상세히 설명된 "원격" 디코더(510)의 것과 동일할 수 있다. 그러나, 또한 도 10을 잠시 참조하면, 심벌들이 이용가능하고 또한 엔트로피 코더(745) 및 파서(620)에 의한 코딩된 비디오 시퀀스로의 심벌들의 인코딩/디코딩이 무손실일 수 있기 때문에, 채널(612), 수신기(610), 버퍼(615), 및 파서(620)를 포함하는, 디코더(510)의 엔트로피 디코딩 부분들은 로컬 디코더(733)에서 완전히 구현되지 않을 수 있다.

[0100] 이 시점에서 이루어질 수 있는 관찰은, 디코더에 존재하는 파싱/엔트로피 디코딩을 제외한 임의의 디코더 기술이 또한 필연적으로, 대응하는 인코더에서, 실질적으로 동일한 기능 형식으로 존재할 필요가 있다는 점이다. 인코더 기술들은 포괄적으로 설명된 디코더 기술들의 역(inverse)이기 때문에 그것들에 대한 설명은 축약될 수 있다. 특정 영역들에서만 더 상세한 설명이 요구되고 아래에 제공된다.

[0101] 그 동작의 일부 동안, 소스 코더(730)는, "참조 프레임들"로 지정된 비디오 시퀀스로부터의 하나 이상의 이전에 코딩된 프레임을 참조하여 예측적으로 입력 프레임을 코딩하는, 모션 보상된 예측 코딩을 수행할 수 있다. 이러한 방식으로, 코딩 엔진(732)은 입력 프레임의 픽셀 블록들과 입력 프레임에 대한 예측 참조(들)로서 선택될 수 있는 참조 프레임(들)의 픽셀 블록들 사이의 차이들을 코딩한다.

[0102] 로컬 비디오 디코더(733)는, 소스 코더(730)에 의해 생성된 심벌들에 기초하여, 참조 프레임들로서 지정될 수 있는 프레임들의 코딩된 비디오 데이터를 디코딩할 수 있다. 코딩 엔진(732)의 동작들은 유리하게는 손실 프로세스들일 수 있다. 코딩된 비디오 데이터가 비디오 디코더(도 11에 도시되지 않음)에서 디코딩될 수 있는 경우, 재구성된 비디오 시퀀스는 전형적으로 일부 오류들을 갖는 소스 비디오 시퀀스의 복제본일 수 있다. 로컬 비디오 디코더(733)는 참조 프레임들에 대해 비디오 디코더에 의해 수행될 수 있는 디코딩 프로세스들을 복제하고, 재구성된 참조 프레임들이 참조 픽처 캐시(734)에 저장되게 야기할 수 있다. 이러한 방식으로, 인코더(503)는 (송신 오류들 없이) 원단(far-end) 비디오 디코더에 의해 획득될 재구성된 참조 프레임으로서 공통 콘텐츠를 갖는 재구성된 참조 프레임들의 사본들을 저장할 수 있다.

[0103] 예측기(735)는 코딩 엔진(732)을 위한 예측 검색들을 수행할 수 있다. 즉, 코딩될 새로운 프레임에 대해, 예측

기(735)는 새로운 픽처들에 대한 적절한 예측 참조로서 역할할 수 있는 참조 픽처 모션 벡터들, 블록 형상들 등과 같은 특정 메타데이터 또는 샘플 데이터(후보 참조 픽셀 블록들로서)에 대해 참조 픽처 메모리(734)를 검색할 수 있다. 예측기(735)는 적절한 예측 참조들을 찾기 위해 샘플 블록 바이 픽셀 블록(sample block-by-pixel block) 기준으로 동작할 수 있다. 일부 경우들에서, 예측기(735)에 의해 획득된 검색 결과들에 의해 결정된 바와 같이, 입력 픽처는 참조 픽처 메모리(734)에 저장된 다중의 참조 픽처로부터 인출된 예측 참조들을 가질 수 있다.

- [0104] 컨트롤러(750)는, 예를 들어, 비디오 데이터를 인코딩하기 위해 사용되는 파라미터들 및 서브그룹 파라미터들의 설정을 포함하여, 비디오 코더(730)의 코딩 동작들을 관리할 수 있다.
- [0105] 전술한 모든 기능 유닛들의 출력은 엔트로피 코더(745)에서 엔트로피 코딩을 겪을 수 있다. 엔트로피 코더는, 예를 들어, 허프만 코딩, 가변 길이 코딩, 산술 코딩 등으로서 본 기술분야의 통상의 기술자에게 공지된 기술들에 따라 심벌들을 무손실 압축함으로써 다양한 기능 유닛들에 의해 생성된 심벌들을 코딩된 비디오 시퀀스로 변환한다.
- [0106] 송신기(740)는, 인코딩된 비디오 데이터를 저장할 저장 디바이스에 대한 하드웨어/소프트웨어 링크일 수 있는, 통신 채널(760)을 통한 송신을 위해 코딩된 비디오 시퀀스(들)를 준비하도록 엔트로피 코더(745)에 의해 생성된 코딩된 비디오 시퀀스(들)를 버퍼링할 수 있다. 송신기(740)는 비디오 코더(730)로부터의 코딩된 비디오 데이터를 송신될 다른 데이터, 예를 들어, 코딩된 오디오 데이터 및/또는 보조 데이터 스트림들(소스들이 도식되지 않음)과 병합할 수 있다.
- [0107] 컨트롤러(750)는 비디오 인코더(503)의 동작을 관리할 수 있다. 코딩 동안, 컨트롤러(750)는, 각자의 픽처에 적용될 수 있는 코딩 기법들에 영향을 미칠 수 있는, 특정 코딩된 픽처 타입을 각각의 코딩된 픽처에 할당할 수 있다. 예를 들어, 픽처들은 종종 이하의 프레임 타입들 중 하나로서 할당될 수 있다:
- [0108] 인트라-픽처(I 픽처)는 예측의 소스로서 시퀀스 내의 임의의 다른 프레임 사용하지 않고 코딩되고 디코딩될 수 있는 것일 수 있다. 일부 비디오 코덱들은, 예를 들어, 독립적인 디코더 리프레시 픽처들을 포함하는, 상이한 타입들의 인트라-픽처들을 허용한다. 본 기술분야의 통상의 기술자는 I 픽처들의 해당 변형들 및 그것들 각각의 응용들 및 특징들을 인식한다.
- [0109] 예측 픽처(P 픽처)는 각각의 블록의 샘플 값들을 예측하기 위해 많아야 하나의 모션 벡터 및 참조 인덱스를 사용하여 인트라-예측 또는 인터 예측을 사용하여 코딩되고 디코딩될 수 있는 것일 수 있다.
- [0110] 양방향 예측 픽처(Bi-directionally Predictive Picture, B 픽처)는 각각의 블록의 샘플 값들을 예측하기 위해 많아야 2개의 모션 벡터 및 참조 인덱스를 사용하여 인트라-예측 또는 인터 예측을 사용하여 코딩되고 디코딩될 수 있는 것일 수 있다. 유사하게, 다중-예측 픽처들은 단일 블록의 재구성을 위해 2개보다 많은 참조 픽처 및 연관된 메타데이터를 사용할 수 있다.
- [0111] 소스 픽처들은 흔히 복수의 샘플 블록(예를 들어, 각각 4x4, 8x8, 4x8, 또는 16x16 샘플들의 블록들)으로 공간적으로 세분되고 블록 바이 블록(block-by-block) 기준으로 코딩될 수 있다. 블록들은 블록들의 각자의 픽처들에 적용되는 코딩 할당에 의해 결정된 다른 (이미 코딩된) 블록들을 참조하여 예측적으로 코딩될 수 있다. 예를 들어, I 픽처들의 블록들은 비-예측적으로 코딩될 수 있거나 또는 이들은 동일한 픽처의 이미 코딩된 블록들을 참조하여 예측적으로 코딩될 수 있다(공간 예측 또는 인트라-예측). P 픽처들의 픽셀 블록들은, 하나의 이전에 코딩된 참조 픽처를 참조하여, 공간 예측을 통해 또는 시간 예측을 통해 비-예측적으로 코딩될 수 있다. B 픽처들의 블록들은, 하나 또는 2개의 이전에 코딩된 참조 픽처를 참조하여 공간 예측을 통해 또는 시간 예측을 통해, 비-예측적으로 코딩될 수 있다.
- [0112] 비디오 코더(503)는 H.265 HEVC와 같은 표준 또는 미리 결정된 비디오 코딩 기술에 따라 코딩 동작들을 수행할 수 있다. 그것의 동작 중에, 비디오 코더(503)는, 입력 비디오 시퀀스에서의 시간 및 공간 중복성을 활용하는 예측 코딩 동작들을 포함하여, 다양한 압축 동작들을 수행할 수 있다. 따라서, 코딩된 비디오 데이터는 사용 중인 비디오 코딩 기술 또는 표준에 의해 특정된 신택스(syntax)를 준수할 수 있다.
- [0113] 일 실시예에서, 송신기(740)는 인코딩된 비디오와 함께 추가적인 데이터를 송신할 수 있다. 비디오 코더(730)는 코딩된 비디오 시퀀스의 일부로서 그러한 데이터를 포함할 수 있다. 추가적인 데이터는 시간/공간/SNR 향상 계층들, 중복 픽처들 및 슬라이스들과 같은 다른 형태들의 중복 데이터, SEI(Supplementary Enhancement Information) 메시지들, VUI(Visual Usability Information) 파라미터 세트 프래그먼트들 등을 포함할 수

있다.

- [0114] 본 개시내용은 크로스 컴포넌트 선형 모델 예측 모드를 위한 다양한 방법들에 관한 것이다.
- [0115] 위에서 설명된 바와 같이, 본 출원에서, CCLM은 LT_CCLM, T_CCLM, L_CCLM, MMLM, LT_MMLM, L_MMLM, 및 T_MMLM 과 같은 크로스 컴포넌트 선형 모델의 임의의 변형들을 지칭할 수 있다. 또한, 본 출원에서, 평활화 필터가 홀수 개의 탭을 갖는 선형 필터로서 정의될 수 있고, 그 필터 계수들은 대칭일 수 있다. 예를 들어, 3-탭 필터 [1, 2, 1] 및 5-탭 필터 [1, 3, 8, 3, 1]은 둘 다 평활화 필터들이다. 보간 필터는 정수 위치 샘플들을 사용하여 분수 위치 샘플을 생성하는 선형 필터로서 정의될 수 있다.
- [0116] CCLM에 대해 이용된 보간 필터는 특정된 영역에서의 이웃 루마 재구성된 샘플들의 서브세트에 적용될 수 있다는 것이 제안된다.
- [0117] 일 양태에서, 지정된 이웃 샘플 영역은 L_CCLM 및 L_MMLM에 대해 사용되는 왼쪽 이웃 샘플 영역, 또는 T_CCLM 및 T_MMLM에 대해 사용되는 상단 이웃 샘플 영역, 또는 LT_CCLM 및 LT_MMLM에 대해 사용되는 상단 및 왼쪽 이웃 샘플 영역일 수 있다.
- [0118] 또 다른 양태에서, 특정된 영역에서의 재구성된 루마 샘플들은 최대 및 최소값들을 발견하기 위하여 직접적으로 이용될 수 있다. 최대 및 최소 루마 샘플 값들을 획득한 후에, 보간 (또는 평활화) 필터가 (예를 들어, 단지) 이들 2개의 샘플에 적용될 수 있다.
- [0119] 또 다른 양태에서, 최대 및 최소 루마 샘플 값들을 획득한 후에, 크로마 이웃 샘플 영역에서의 최소 및 최대 루마 샘플들의 공동 위치된 위치들은 제각기 (x_min, y_min) 및 (x_max, y_max)로서 기록될 수 있으며, 위치 (x_min, y_min) 및 (x_max, y_max)를 갖는 크로마 샘플들에 대한 공동 위치된 루마 샘플들에 다운-샘플 필터가 적용될 수 있다.
- [0120] 또 다른 양태에서, 4:2:0 YUV 포맷에 대해, 다운-샘플 필터는 다음과 같이 적용될 수 있다: 다음의 수학적식에서 의 (x, y)는 (x_min, y_min) 또는 (x_max, y_max)로 대체될 수 있고, $rec'_L(x,y)$ 는 다운-샘플링된 루마 샘플 값들을 나타낼 수 있고, $rec_L(kx,ky)$ 는 특정된 이웃 위치 (k·x, k·y)에서 구성된 루마 샘플들을 나타낼 수 있고, k는 1, 2, 3, 또는 4와 같은 양의 정수일 수 있다.

$$rec'_L(x,y) = (2 \times rec_L(kx,ky) + 2 \times rec_L(kx,ky+1) + rec_L(kx-1,ky) + rec_L(kx-1,ky+1) + rec_L(kx-1,ky) + rec_L(kx-1,ky+1)) \gg 3$$
- [0121]
- [0122] 또 다른 양태에서, 4:2:0 YUV 포맷에 대해, 위치 (x, y)를 갖는 하나의 크로마 샘플에 대한 루마 샘플 영역에서의 공동 위치된 위치들은 다음 6개의 위치 (k*x, k*y), (k*x - 1, k*y), (k*x + 1, k*y), (k*x - 1, k*y + 1), (k*x, k*y + 1), 및 (k*x + 1, k*y + 1) 중 어느 하나일 수 있다. 여기서 k는 양의 정수, 예컨대 1, 2, 3, 또는 4이다.
- [0123] 또 다른 양태에서, 4:4:4 및 4:2:2 YUV 포맷들에 대해, 전술한 보간 필터가 여기서 사용될 수 있다.
- [0124] 또 다른 양태에서, 특정된 영역에서의 재구성된 루마 샘플들은 최대 및 최소값들을 발견하기 위하여 직접적으로 이용될 수 있다.
- [0125] 또 다른 양태에서, LT_CCLM 모드에 대해, (도 6a-도 6b에 도시된 바와 같은) 제1 상단 참조 행 및 (도 7a-도 7b에 도시된 바와 같은) 제2 왼쪽 참조 열에서의 이웃 루마 샘플들이 최대 및 최소값들을 발견하기 위하여 직접적으로 이용될 수 있다.
- [0126] 또 다른 양태에서, LT_CCLM 모드에 대해, (도 6a-도 6b에 도시된 바와 같은) 제1 및 제2 상단 참조 행에서의 이웃 루마 샘플들 및 (도 7a-도 7b에 도시된 바와 같은) 제2 왼쪽 참조 열에서의 샘플들의 절반이 최대 및 최소값들을 발견하기 위해 직접적으로 이용될 수 있다.
- [0127] 또 다른 양태에서, T_CCLM 모드에 대해, 제1 상단 참조 행에서의 이웃 루마 샘플들은 최대 및 최소값들을 발견하기 위해 직접적으로 이용될 수 있다.
- [0128] 또 다른 양태에서, T_CCLM 모드에 대해, 제1 및 제2 상단 참조 행에서의 이웃 루마 샘플들의 절반이 최대 및 최

소값들을 발견하기 위해 직접적으로 이용될 수 있다.

- [0129] 또 다른 양태에서, L_CCLM 모드에 대해, 제2 왼쪽 참조 열에서의 이웃 루마 샘플들이 최대 및 최소값들을 발견하기 위해 직접적으로 이용될 수 있다.
- [0130] 또 다른 양태에서, 보간 필터가 이웃 루마 샘플들의 일부에 대해 사용될 때(예를 들어, 단지 사용될 때), 현재 CCLM에서 사용되는 것과 상이한 다운샘플링 필터, 즉, 6-탭 [1,2,1;1,2,1]/8 필터가 적용될 수 있다. 예를 들어, 10-탭 필터는 [1,4,6,4,1; 1,4,6,4,1]/16일 수 있다.
- [0131] 또 다른 양태에서, 다운샘플링 필터는 8-탭 필터일 수 있으며, 여기서 (도 6a-도 6b에 도시된 바와 같은) (현재 블록에 가장 가까운) 제1 참조 행에서의 필터 탭들의 수는 (도 7a-도 7b에 도시된 바와 같은) (현재 블록에 더 먼) 다른 참조 열과 상이할 수 있다. 예를 들어, 8-탭 필터는 [1,2,1; 1,2,6,2,1]/16일 수 있고, 제1 참조 행 (또는 열)에 대한 필터 탭들의 수는 5일 수 있는 반면, 제2 참조 행(또는 열)에 대한 탭들의 수는 3일 수 있다.
- [0132] 또 다른 양태에서, 특정된 영역에서의 재구성된 크로마 샘플들은 최대 및 최소값들을 발견하기 위하여 이용될 수 있다는 것이 제안된다.
- [0133] 일 양태에서, 최대 및 최소 크로마 샘플 값들을 획득한 후에, 보간 (또는 평활화) 필터가 이들 2개의 크로마 샘플의 공동 위치된 루마 샘플들에 적용(예를 들어, 단지 적용)될 수 있다. 예를 들어, 4:2:0 YUV 포맷에 대해, 위에 설명된 보간 (또는 평활화) 필터가 이용될 수 있다. 또 다른 예로서, 4:2:0 YUV 포맷에 대해, 위에 설명된 위치 (x,y)를 갖는 하나의 크로마 샘플에 대한 공동 위치된 위치들이 여기서 이용될 수 있다. 또 다른 예에서, 4:4:4 및 4:2:2 YUV 포맷들에 대해, 이하의 설명에서의 보간 필터가 여기서 사용될 수 있다.
- [0134] 일 양태에 따르면, CCLM에서 사용되는 보간 필터(또는 평활화 필터)가, 4:4:4, 4:2:2, 또는 4:2:0 YUV 포맷과 같은 YUV 포맷에 의존할 수 있는 것이 제안된다.
- [0135] 도 13은 비디오 시퀀스를 인코딩 또는 디코딩하기 위한 예시적인 프로세스(800)의 흐름도이다. 일부 구현들에서, 도 13의 하나 이상의 프로세스 블록은 디코더(510)에 의해 수행될 수 있다. 일부 구현들에서, 도 13의 하나 이상의 프로세스 블록은, 인코더(503)와 같이, 디코더(510)와 별개이거나 이를 포함하는 또 다른 디바이스 또는 디바이스들의 그룹에 의해 수행될 수 있다.
- [0136] 도 13에 도시된 바와 같이, 프로세스(800)는 비디오 시퀀스를 인코딩 또는 디코딩하는 것을 포함할 수 있다 (810).
- [0137] 도 13에 더 도시된 바와 같이, 프로세스(800)는 CCLM(Cross-Component Linear Model)을 비디오 시퀀스에 적용하는 것(820)을 추가로 포함할 수 있다.
- [0138] 도 13에 더 도시된 바와 같이, 프로세스(800)는 CCLM(Cross-Component Linear Model)에서 보간 필터를 적용하는 것을 추가로 포함할 수 있으며, 여기서 보간 필터는 비디오 시퀀스의 YUV 포맷에 의존한다(830).
- [0139] 도 13은 프로세스(800)의 예시적인 블록들을 도시하지만, 일부 구현들에서, 프로세스(800)는 추가적인 블록들, 더 적은 블록들, 상이한 블록들, 또는 도 13에 묘사된 것들과는 상이하게 배열된 블록들을 포함할 수 있다. 추가적으로 또는 대안적으로, 프로세스(800)의 블록들 중 2개 이상이 병렬로 수행될 수 있다.
- [0140] 또 다른 양태에 따르면, CCLM에서 사용되는 보간(또는 평활화) 필터의 탭들은 YUV 포맷들에 의존할 수 있다.
- [0141] 또 다른 양태에서, CCLM에서 사용되는 보간(또는 평활화) 필터의 탭들은 동일한 YUV 포맷에 대해 동일할 수 있다. 예를 들어, CCLM에서 사용되는 보간 필터의 탭들은 4:2:0 YUV 포맷에 대해 동일할 수 있다.
- [0142] 또 다른 양태에서, 4:4:4 및 4:2:2 YUV 포맷들에 대한 보간 또는 평활화 필터는 동일하지만, 4:2:0 YUV 포맷에 대한 필터와는 상이하다. 예를 들어, 3-탭 (1,2,1) 필터는 4:4:4 및 4:2:2 YUV 포맷들에 대해 사용될 수 있는 반면, 6-탭 (1,2,1;1,2,1) 필터는 4:2:0 YUV 포맷에 대해 사용될 수 있다. 또 다른 예로서, 4:4:4 및 4:2:2 YUV 포맷들에 대해 어떤 다운-샘플(또는 평활화) 필터도 사용되지 않을 수 있는 반면, 4:2:0 YUV 포맷에 대해 6-탭 (1,2,1,1,2,1) 또는 3-탭 (1,2,1) 필터가 사용될 수 있다.
- [0143] 또 다른 양태에 따르면, CCLM에서 사용되는 보간(또는 평활화) 필터의 탭들은 상이한 YUV 포맷들에 대해 상이할 수 있다. 예를 들어, 5-탭 (1,1,4,1,1) 필터는 4:4:4 YUV 포맷을 위해 사용될 수 있고, (1,2,1) 필터는 4:2:2 YUV 포맷을 위해 사용될 수 있고, 6 탭 (1,2,1,1,2,1) 필터는 4:2:0 YUV 포맷을 위해 사용될 수 있다.
- [0144] 또 다른 양태에서, 보간(또는 평활화) 필터는 재구성된 샘플들에서의 상단 및 왼쪽 이웃 루마에 대해 상이할 수

있다.

[0145] 또 다른 양태에 따르면, 상단 및 왼쪽 이웃 루마 재구성된 샘플들에 대한 보간(또는 평활화) 필터는 YUV 포맷들에 의존할 수 있다.

[0146] 또 다른 양태에 따르면, 4:2:2 YUV 포맷의 경우, 보간(또는 평활화) 필터는 상단 및 왼쪽 이웃 샘플들에 대해 상이할 수 있다. 예를 들어, 4:2:2 YUV 포맷의 경우, (1,2,1) 필터가 상단 이웃 루마 샘플들에 적용될 수 있고, 어떤 보간(또는 평활화) 필터도 왼쪽 이웃 루마 샘플들에 적용되지 않는다.

[0147] 또 다른 양태에 따르면, CCLM에서 사용되는 상단 이웃 루마 샘플들에서의 라인들의 수 및 왼쪽 이웃 루마 샘플들에서의 열들의 수는 YUV 포맷에 의존할 수 있다. 예를 들어, 상단 이웃 영역에서의 하나의 행 및/또는 왼쪽 이웃 영역에서의 하나의 열만이 4:4:4 YUV 또는 4:2:2 YUV 포맷에 대해 사용될 수 있다. 또 다른 예로서, 상단 이웃 영역에서의 하나의 행 및/또는 왼쪽 이웃 영역에서의 3(또는 2)개의 열이 4:2:2 YUV 포맷에 대해 사용될 수 있다.

[0148] 또 다른 양태에서, diff_Y로 표시되는, 특정된 이웃 샘플 영역들 내의 최대 및 최소 루마 샘플 값들 사이의 절대 차이는 불균일 양자화되고, 절대 차이의 양자화된 값은 룩업 테이블(LUT) 크기가 감소되도록 CCLM LUT의 엔트리 인덱스를 특정하기 위해 사용될 수 있는 것이 제안된다. diff_Y의 범위는 2개의 구간으로 분할될 수 있고, diff_Y가 Thres_1로 명명되는 임계값 이하이면, Step_A로 명명되는 하나의 스텝 크기가 사용될 수 있다. 그렇지 않으면, Step_B로 명명된 또 다른 스텝 크기가 사용된다. 결과적으로, CCLM에서의 파라미터 a는 다음과 같이 획득될 수 있다:

$$a = (\text{diff}_Y > \text{Thres}_1) ? \text{LUT}\left[\frac{\text{Thres}_1}{\text{Step}_A} + \frac{\text{diff}_Y - \text{Thres}_1}{\text{Step}_B} - 1\right] : \text{LUT}[\text{diff}_Y/\text{Step}_A - 1]$$

[0150] 여기서 이 양태에서, Thres_1은 64와 동일하게 설정될 수 있고, Step_A는 1과 동일하게 설정될 수 있고, Step_B는 8과 동일하게 설정될 수 있다.

[0151] 이제, VVC 초안 3 조항 8.2.4.2.8에 더하여 수정된 CCLM 파라미터 도출 프로세스가 아래에 설명될 것이다.

[0152] 또 다른 양태에 따르면, 바로 위의 수학적식에서, 변수들 a, b, 및 k는 다음과 같이 도출될 수 있다:

[0153] - numSampL이 0과 동일하고, numSampT가 0과 동일하면, 이하가 적용된다:

$$k = 0$$

$$a = 0$$

$$b = 1 \ll (\text{BitDepth}_C - 1)$$

[0155] - 그렇지 않으면, 다음이 적용된다:

$$\text{shift} = (\text{BitDepth}_C > 8) ? \text{BitDepth}_C - 9 : 0$$

$$\text{add} = \text{shift} ? 1 \ll (\text{shift} - 1) : 0$$

$$\text{diff} = (\text{maxY} - \text{minY} + \text{add}) \gg \text{shift}$$

$$k = 16$$

[0157] - diff가 0보다 큰 경우, 다음이 적용된다:

$$\text{diff} = (\text{diff} > 64) ? (56 + (\text{diff} \gg 3)) : \text{diff}$$

$$a = ((\text{maxC} - \text{minC}) * \text{g_aiLMDivTableHigh}[\text{diff} - 1] + \text{add}) \gg \text{shift}$$

[0159] - 그렇지 않으면, 다음이 적용된다:

$$a = 0$$

$$b = \text{minC} - ((a * \text{minY}) \gg k)$$

[0160]

[0161] 여기서, $x = 0..nTbW - 1$, $y = 0..nTbH - 1$ 인 예측 샘플들 $predSamples[x][y]$ 는 다음과 같이 도출될 수 있다:

[0162] $predSamples[x][y] = Clip1C(((pDsY[x][y] * a) \gg k) + b)$

[0163] 또한, 단순화된 CCLM 룩업 테이블이 아래에 도시된다:

```
[0164] int g_aiLMDivTableHigh[] = {
    65536, 32768, 21845, 16384, 13107, 10922, 9362, 8192, 7281, 6553, 5957, 5461,
    5041, 4681, 4369, 4096,
    3855, 3640, 3449, 3276, 3120, 2978, 2849, 2730, 2621, 2520, 2427, 2340, 2259,
    2184, 2114, 2048,
    1985, 1927, 1872, 1820, 1771, 1724, 1680, 1638, 1598, 1560, 1524, 1489, 1456,
    1424, 1394, 1365,
    1337, 1310, 1285, 1260, 1236, 1213, 1191, 1170, 1149, 1129, 1110, 1092, 1074,
    1057, 1040, 1024,
    910, 819, 744, 682, 630, 585, 546, 512, 481, 455, 431, 409, 390, 372, 356,
    341,
    327, 315, 303, 292, 282, 273, 264, 256, 248, 240, 234, 227, 221, 215, 210,
    204,
    199, 195, 190, 186, 182, 178, 174, 170, 167, 163, 160, 157, 154, 151, 148,
    146,
    143, 141, 138, 136, 134, 132, 130, 128,};
```

[0165] 실시예들에서, $diff_Y$ 로 표시되는, 특정된 이웃 샘플 영역들 내의 최대 및 최소 루마 샘플 값들 사이의 절대 차이는 불균일 양자화되고, 절대 차이의 양자화된 값은 룩업 테이블(LUT)의 크기가 감소되도록 CCLM LUT의 엔트리 인덱스를 특정하기 위해 사용된다.

[0166] 예를 들어, 특정된 이웃 샘플 영역은 L_CCLM 및 L_MMLM 에 대해 사용되는 왼쪽 이웃 샘플 영역, 또는 T_CCLM 및 T_MMLM 에 대해 사용되는 상단 이웃 샘플 영역, 또는 TL_CCLM 및 TL_MMLM 에 대해 사용되는 상단 및 왼쪽 이웃 샘플 영역들일 수 있다.

[0167] 또 다른 예에서, $diff_Y$ 의 범위는 다중의 구간으로 분할되고, 상이한 양자화 스텝 크기들이 상이한 구간들에서 사용된다. 제1 실시예들에서, $diff_Y$ 의 범위는 2개의 구간으로 분할된다; $diff_Y$ 가 $Thres_1$ 로 명명된 임계값 이하이면, $Step_A$ 로 명명된 하나의 스텝 크기가 사용된다. 그렇지 않으면, $Step_B$ 로 명명된 또 다른 스텝 크기가 사용된다. 결과적으로, CCLM에서의 파라미터 a 는 다음과 같이 획득될 수 있다:

$$a = (diff_Y > Thres_1) ? LUT[diff_Y/Step_A - 1] : LUT[\frac{Thres_1}{Step_A} + \frac{diff_Y - Thres_1}{Step_B} - 1]$$

[0168]

[0169] $Thres_1$, $Step_A$, 및 $Step_B$ 각각은 1, 2, 3, 4 등과 같은 임의의 양의 정수일 수 있다. $Step_A$ 및 $Step_B$ 는 동일하지 않다. 일 예에서, $Thres_1$ 은 32, 또는 48, 또는 64와 동일하게 설정된다. 또 다른 예에서, $Thres_1$ 은 64와 동일하게 설정되고, $Step_A$ 는 1과 동일하게 설정되고, $Step_B$ 는 8과 동일하게 설정된다. 또 다른 예에서, $Thres_1$ 은 64와 동일하게 설정되고, $Step_A$ 는 2(또는 1)와 동일하게 설정되고, $Step_B$ 는 8(또는 4)과 동일하게 설정된다. 또 다른 예에서, $Step_A$ 의 값은 $Step_B$ 의 값보다 작다.

[0170] 제2 실시예들에서, $diff_Y$ 의 범위는 2개의 임계값, 즉 $Thres_1$ 및 $Thres_2$ 에 의해 3개의 구간으로 분할된다. $diff_Y$ 가 $Thres_1$ 이하이면, 하나의 스텝 크기, 즉 $Step_A$ 가 사용된다; 그렇지 않고, $diff_Y$ 가 $Thres_2$ 이하이면, 또 다른 스텝 크기, 즉 $Step_B$ 가 사용된다; 그렇지 않으면, 제3 스텝 크기, 즉 $Step_C$ 가 사용된다. $Thres_1$, $Thres_2$, $Step_A$, $Step_B$, 및 $Step_C$ 각각은, 1, 2, 3, 4 등과 같은 임의의 양의 정수일 수 있다. $Thres_1$ 은 $Thres_2$ 보다 작다. $Step_A$, $Step_B$, 및 $Step_C$ 는 동일하지 않다. 일 예에서, $Thres_1$ 및 $Thres_2$ 는 제각기 64 및 256과 동일하게 설정된다. 또 다른 예에서, $Step_A$, $Step_B$, 및 $Step_C$ 는 제각기 2(또는 1), 8(또는 4), 및 32(또는 16)와 동일하게 설정된다.

[0171] 제3 실시예들에서, ($Thres_1$ 및 $Thres_2$ 와 같은) 구간들을 특정하기 위해 사용되는 임계값들은 2의 거듭제곱들,

예를 들어, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024이다.

- [0172] 추가 실시예들에서, LIC는 CCLM에서 최대-최소 방법을 재사용하여 파라미터들 a 및 b를 도출한다.
- [0173] 예를 들어, LIC는 나눗셈 연산을 피하기 위해 CCLM에서 룩업 테이블을 재사용한다.
- [0174] 또 다른 예에서, LIC는 파라미터들 a 및 b를 도출하기 위해 TL_CCLM에서의 루마 성분에 대해 사용되는 것과 동일한 이웃 재구성된 샘플 영역을 사용한다.
- [0175] 또 다른 예에서, LIC는 CCLM에서 동일한 다운-샘플 방법을 재사용하여 이웃 재구성된 샘플들을 필터링한다.
- [0176] 추가의 실시예들에서, 특정된 영역에서의 이웃 재구성된 샘플들의 서브세트만이 CCLM 예측 모드에서의 최대-최소 방법에 대한 최대 및 최소 샘플 값들을 계산하기 위하여 이용된다.
- [0177] 예를 들어, 이웃 재구성된 샘플 영역에서의 4개의 샘플만이 현재 블록의 블록 크기에 관계없이 최대 및 최소 샘플 값들을 계산하기 위해 사용된다. 이 예에서, 선택된 4개의 샘플의 위치는 고정되고 도 15에 도시된다. 구체적으로, 현재 CU(1510)와 관련하여, 위치 A, B, C 및 D에서의 이웃 재구성된 샘플들의 샘플 값들만이 최대 및 최소 샘플 값들을 계산하기 위해 사용된다.
- [0178] 또 다른 예에서, TL_CCLM 모드에 대해, 왼쪽(또는 상부) 이웃 샘플들만이 최소 샘플 값을 계산하기 위해 사용되고, 상부(또는 좌측) 이웃 샘플들만이 최대 샘플 값을 계산하기 위해 사용된다.
- [0179] 또 다른 예에서, CCLM 예측 모드에서 최대-최소 방법에 대한 최대 및 최소 샘플 값들을 발견하는 대신에, 상단 루마 및 크로마 참조 샘플들의 평균 값, 즉 top_mean_luma, top_mean_chroma, 및 왼쪽 루마 및 크로마 참조 샘플들의 평균값들, 즉, left_mean_luma, left_mean_chroma가 CCLM 파라미터들을 도출하기 위하여 이용될 수 있다.
- [0180] 다음의 실시예들에서, 도 16을 참조하면, 현재 CU(1610)의 상단-왼쪽 샘플 및 하단-왼쪽 샘플이 사용될 때, 이들 블록은 현재 CU(1610)의 상부 및 왼쪽 샘플의 범위 밖에 위치하는 그들의 직접 이웃 샘플들로 대체될 수 있다. 예를 들어, 도 16에서, 샘플들 A 및 B가 사용될 수 있다.
- [0181] 또 다른 예에서, 도 17에서, 샘플들 A, B 및 C가 현재 CU(1710)에서 사용될 수 있다.
- [0182] 이하의 실시예들에서, (U 또는 V 성분에서의) 크로마 샘플은 그의 동일-위치된 루마 샘플과 함께 샘플 쌍으로 명명된다. 대안적으로, LIC에 대해, 현재 블록의 이웃 샘플은 참조 블록의 동일-위치된 위치에서의 이웃 샘플과 함께 샘플 쌍으로 명명되고, 따라서 샘플 쌍에 포함된 샘플들이 동일한 컬러 성분(루마, U 또는 V)으로부터 온다.
- [0183] 다음의 실시예들에서, CCLM 또는 CCLM 모드의 임의의 변형/확장에 대하여, 블록 크기는 크로마 블록의 것일 수 있다. LIC 모드 또는 LIC의 임의의 변형에 대해, 블록 크기는 현재 컬러 성분(루마, U 또는 V)의 현재 블록의 것일 수 있다.
- [0184] 실시예들에서, 특정된 영역에서의 최대 N개의 이웃 재구성된 샘플 쌍이 CCLM 예측 모드에서 최대-최소 방법에 대한 최대 및 최소 샘플 값들을 계산하기 위해 사용되고, N은 4, 8, 또는 16과 같은 양의 정수이다.
- [0185] 예를 들어, Nx2 또는 2xN 크로마 블록에 대해, 왼쪽으로부터의 하나의 이웃 샘플 쌍 및 상부로부터의 하나의 이웃 샘플 쌍만이 최대 및 최소 샘플 값들을 계산하기 위해 사용된다. 사용된 이웃 샘플 쌍의 위치는 고정된다. 일 예에서, 샘플 쌍에서의 크로마 샘플의 위치가 제각기 현재 CU들(1810, 1910)에 대해 도 18 및 도 19에 도시된다. 또 다른 예에서, 샘플 쌍에서의 크로마 샘플의 위치가 현재 CU(2010)에 대해 도 20에 도시된다.
- [0186] 또 다른 예에서, 샘플 쌍에서의 크로마 샘플의 위치가 현재 CU(2110)에 대해 도 21에 도시되며, 여기서 샘플 B는 현재 CU(2110)의 더 긴 변의 임의의 위치에 있을 수 있다. 도 21에 도시된 바와 같이, 샘플 B는 현재 CU(2110)의 더 긴 변의 중간에 있다. 샘플 B의 x-좌표는 (폭>>1)-1 또는 (폭>>1) 또는 (폭>>1)+1일 수 있다.
- [0187] 또 다른 예에서, 샘플 쌍에서의 크로마 샘플의 위치가 현재 CU(2210)에 대해 도 22에 도시되며, 여기서 샘플 B는 현재 CU(2210)의 더 긴 변의 임의의 위치에 있을 수 있다. 도 22에 도시된 바와 같이, 샘플 B는 현재 CU(2210)의 더 긴 변의 중간에 있다. 샘플 B의 x-좌표는 (폭>>1)-1 또는 (폭>>1) 또는 (폭>>1)+1일 수 있다.
- [0188] 또 다른 예에서, 2xN 또는 Nx2 크로마 블록들에 대해, N은 2, 4, 8, 16, 32 또는 64와 같은 임의의 양의 정수일 수 있다. N이 2와 동일할 때, 왼쪽으로부터의 하나의 이웃 샘플 쌍 및 상부로부터의 하나의 이웃 샘플 쌍만이 최대 및 최소 샘플 값들을 계산하기 위해 사용된다. 그렇지 않으면, N이 2보다 클 때, 현재 CU의 짧은 변으로

부터의 하나의 이웃 샘플 쌍 및 더 긴 변으로부터의 2개 이상의 이웃 샘플 쌍만이 최대 및 최소 샘플 값들을 계산하기 위해 사용된다. 사용된 이웃 샘플 쌍의 위치는 고정된다. 일 예에서, 샘플 쌍에서의 크로마 샘플의 위치들은 현재 CU(2310)에 대해 도 23에 도시되고, 여기서 샘플 B는 현재 CU(2310)의 더 긴 변의 중간에 있고 샘플 C는 더 긴 변의 끝에 있다.

[0189] 또 다른 예에서, 크로마 블록에 대해, 현재 블록의 폭 및 높이 둘 다가 2보다 클 때, 현재 블록의 좌측으로부터의 2개 이상의 샘플 쌍 및 상부측으로부터의 2개 이상의 샘플 쌍이 최대 및 최소 샘플 값들을 계산하기 위해 사용된다. 사용된 이웃 샘플 쌍의 위치는 고정된다. 일 예에서, 도 24를 참조하면, 현재 CU(2410)의 각각의 변에 대해, 하나의 샘플은 (샘플들 B 및 C와 같이) 변의 중간에 있고 또 다른 샘플은 (샘플들 A 및 D와 같이) 변의 끝에 있다.

[0190] 또 다른 예에서, CCLM 모드에 대해, 특정된 이웃 샘플들에서의 이웃 샘플들의 서브세트만이 최대 및 최소 샘플 값들을 계산하기 위해 사용된다. 선택된 샘플들의 수는 블록 크기에 의존한다. 일 예에서, 선택된 이웃 샘플들의 수는 좌측 및 상부측에 대해 동일하다. 크로마 블록에 대해, 사용된 샘플들은 특정된 스캐닝 순서에 의해 선택된다. 도 25를 참조하면, 크로마 블록(2510)의 좌측에 대해, 스캐닝 순서는 상단에서 하단으로이고, 상부측에 대해, 스캐닝 순서는 왼쪽에서 오른쪽으로이다. 선택된 이웃 샘플들은 어두운 원들로 강조된다. 도 26을 참조하면, 크로마 블록(2610)에 대해, 사용된 샘플들은 반전된 스캐닝 순서에 의해 선택된다. 즉, 크로마 블록(2610)의 좌측에 대해, 스캐닝 순서는 하단에서 상단으로이고, 상부측에 대해, 스캐닝 순서는 오른쪽에서 왼쪽으로이다. 다수의 선택된 이웃 샘플들은 좌측 및 상부측에 대해 동일하고, 선택된 이웃 샘플들은 어두운 원들로 강조된다.

[0191] 추가의 실시예들에서, CCLM 모드에 대하여, 상이한 다운-샘플링(또는 보간) 필터 타입들이 왼쪽 이웃 루마 샘플들 및 상부 이웃 루마 샘플들에 대하여 이용된다.

[0192] 예를 들어, 다수의 필터 타입이 좌측 및 위의 상부 샘플들에 대해 동일하지만, 좌측 및 상부 이웃 샘플들에 대해 사용되는 필터들의 위치는 상이하다. 도 27을 참조하면, 좌측에 루마 샘플들에 대한 필터들의 위치들이 도시되어 있으며, 여기서 W 및 X는 필터들의 계수들을 나타내고, 블록은 루마 샘플들의 위치들을 나타낸다. 도 28을 참조하면, 상부측에 루마 샘플들에 대한 필터들의 위치가 도시되어 있으며, 여기서 W 및 X는 필터들의 계수들을 나타내고, 블록은 루마 샘플들의 위치들을 나타낸다.

[0193] 추가의 실시예들에서, 특정된 영역에서의 최대 N개의 이웃 재구성된 샘플 쌍들은 선형 모델 예측 모드에서 파라미터들 a 및 b를 계산하기 위하여 이용되고, N은 4, 8, 또는 16과 같은 양의 정수이다. N의 값은 현재 블록의 블록 크기에 의존할 수 있다.

[0194] 예를 들어, LIC 및 CCLM(또는 L_CCLM, T_CCLM, 또는 LT_CCLM과 같은 CCLM의 임의의 변형)은 선형 모델 파라미터들 a 및 b를 계산하기 위해 동일한 이웃 샘플 위치들을 공유한다.

[0195] 또 다른 예에서, LIC의 파라미터들 a 및 b를 계산하기 위해 사용되는 샘플 쌍들은 상이한 컬러 성분들에 대해 상이한 위치들에 위치될 수 있다. 상이한 컬러 성분들에 대한 샘플 쌍들은 독립적으로 결정될 수 있지만, 알고리즘은, 예를 들어, 모델 파라미터들을 도출하기 위한 최소 및 최대 샘플 값들을 사용하여 동일할 수 있다.

[0196] 또 다른 예에서, 하나의 컬러 성분(예를 들어, 루마)에 대한 LIC의 파라미터들 a 및 b를 계산하기 위해 사용되는 제1 샘플 쌍이 먼저 위치되고, 이어서 또 다른 성분(예를 들어, U 또는 V)에 대한 LIC의 파라미터들 a 및 b를 계산하기 위해 사용되는 샘플 쌍들이 제1 샘플 쌍과 동일-위치된 샘플들로서 결정된다.

[0197] 또 다른 예에서, LIC 및 CCLM(또는 L_CCLM, T_CCLM, 또는 LT_CCLM과 같은 CCLM의 임의의 변형)은 선형 모델 파라미터들 a 및 b를 계산하기 위하여, 최소 제곱 오류 방법(least square error method) 또는 최대-최소 방법과 같은 동일한 방법을 공유한다. LIC 및 CCLM 둘 다는 선형 모델 파라미터들 a 및 b를 계산하기 위해 최대-최소 방법을 사용할 수 있다. LIC에 대해, 각각의 샘플 쌍에서의 2개의 샘플의 절대 값들의 합이 선형 모델의 최대 및 최소값들을 계산하기 위해 사용될 수 있다.

[0198] 또 다른 예에서, 3가지 종류의 LIC 모드, LT_LIC, L_LIC, 및 T_LIC가 있다. LT_LIC에 대해, 왼쪽 및 상부 이웃 샘플들 둘 모두가 선형 모델 파라미터들 a 및 b를 도출하기 위해 사용된다. L_LIC에 대해, 왼쪽 이웃 샘플들만이 선형 모델 파라미터들 a 및 b를 도출하기 위해 사용된다. T_LIC의 경우, 상단 이웃 샘플들만이 선형 모델 파라미터들 a 및 b를 도출하기 위해 사용된다. 현재 모드가 LIC 모드인 경우, LT_LIC 모드가 선택되는지를 표시하기 위해 또 다른 플래그가 시그널링될 수 있다. 그렇지 않다면, L_LIC 모드 또는 T_LIC 모드가 시그널링

되는지를 표시하기 위해 하나의 추가적인 플래그가 시그널링될 수 있다.

- [0199] 또 다른 예에서, $N \times 2$ 또는 $2 \times N$ 블록에 대해, 2개의 이웃 샘플 쌍만이 선형 모델 예측 모드에서 파라미터들 a 및 b 를 계산하기 위해 (또는 최대 및 최소 샘플 값들을 계산하기 위해) 사용된다. 사용된 이웃 샘플 쌍들의 위치들은 미리 정의되고 고정된다.
- [0200] 이 예의 실시예들에서, 파라미터들 a 및 b 를 계산하기 위해 양쪽 측을 사용하는 선형 모델 예측 모드에 대해, 왼쪽 및 상부 이웃 샘플들 둘 다가 이용가능한 경우, 왼쪽으로부터의 하나의 이웃 샘플 쌍 및 상부로부터의 하나의 이웃 샘플 쌍만이 선형 모델 예측 모드에서 파라미터들 a 및 b 를 계산하기 위해 사용된다.
- [0201] 이 예의 실시예들에서, 파라미터들 a 및 b 를 계산하기 위해 양쪽 측을 사용하는 선형 모델 예측 모드에 대해, 왼쪽 또는 상부 이웃 샘플들이 이용가능하지 않은 경우, 이용가능한 측으로부터의 2개의 이웃 샘플 쌍이 선형 모델 예측 모드에서 파라미터들 a 및 b 를 계산하기 위해 사용된다. 현재 CU들(2910 및 3010)에 대한 미리 정의된 위치들의 두 가지 예가 제각기 도 29 및 도 30에 도시되어 있다.
- [0202] 이 예의 실시예들에서, 파라미터들 a 및 b 를 계산하기 위해 양쪽 측을 사용하는 선형 모델 예측 모드에 대해, 왼쪽 또는 상부 이웃 샘플들이 이용가능하지 않은 경우, 파라미터들 a 및 b 는 디폴트 값들에 설정되는데, 예컨대 a 는 1에 설정되고 b 는 512에 설정된다.
- [0203] 이 예의 실시예들에서, 샘플 쌍의 위치들이 도 18 및 도 19에 도시된다.
- [0204] 이 예의 실시예들에서, 샘플 쌍의 위치들이 도 20에 도시된다.
- [0205] 이 예의 실시예들에서, 샘플 쌍의 위치들이 도 21에 도시되어 있고, 여기서 샘플 B는 현재 CU(2110)의 더 긴 변상의 임의의 위치에 있을 수 있다. 예를 들어, 샘플 B는 더 긴 변의 중간에 있을 수 있다. 또 다른 예에서, 샘플 B의 x 좌표는 $(\text{폭} \gg 1) - 1$ 또는 $(\text{폭} \gg 1)$ 또는 $(\text{폭} \gg 1) + 1$ 일 수 있다.
- [0206] 이 예의 실시예들에서, 샘플 쌍의 위치들이 도 22에 도시되어 있고, 여기서 샘플 B는 현재 CU(2210)의 더 긴 변상의 임의의 위치에 있을 수 있다. 예를 들어, 샘플 B는 더 긴 변의 중간에 있을 수 있다. 또 다른 예에서, 샘플 B의 x 좌표는 $(\text{폭} \gg 1) - 1$ 또는 $(\text{폭} \gg 1)$ 또는 $(\text{폭} \gg 1) + 1$ 일 수 있다.
- [0207] 또 다른 예에서, $2 \times N$ 또는 $N \times 2$ 블록들에 대해, N 은 2, 4, 8, 16, 32 또는 64와 같은 임의의 양의 정수일 수 있다. N 이 2와 동일할 때, 좌측으로부터의 하나의 이웃 샘플 쌍 및 상부측으로부터의 하나의 이웃 샘플 쌍만이 선형 모델 파라미터들 a 및 b 를 계산하기 위해 사용된다. 그렇지 않고, N 이 2보다 클 때, 더 짧은 변으로부터의 하나의 이웃 샘플 쌍 및 더 긴 변으로부터의 2개 이상의 이웃 샘플 쌍만이 선형 모델 파라미터들 a 및 b 를 계산하기 위해 사용된다. 사용된 이웃 샘플 쌍의 위치들은 미리 정의되고 고정된다. 도 23을 참조하면, 샘플 쌍의 위치들이 도시되며, 여기서 샘플 B는 현재 CU(2310)의 더 긴 변상의 임의의 위치에 있을 수 있다. 샘플 B는 긴 변의 중간에 있고, 샘플 C는 긴 변의 끝에 있다.
- [0208] 또 다른 예에서, 좌측으로부터의 K_1 개 이상의 샘플 쌍 및 상부측으로부터의 K_2 개 이상의 샘플 쌍이 선형 모델 파라미터들 a 및 b 를 계산하기 위해 사용된다. 사용된 이웃 샘플 쌍들의 위치들은 미리 정의되고 고정된다. 또한, K 와 같은 선택된 샘플들의 수는 블록 크기에 의존한다. K_1 및 K_2 각각은 1, 2, 4, 8과 같은 양의 정수이다. K_1 및 K_2 는 동일할 수 있다.
- [0209] 이 예의 실시예들에서, 각각의 측 상의 선택된 샘플 쌍들은 고르게 분포된다.
- [0210] 이 예의 실시예들에서, K_1 및 K_2 의 값들은 현재 블록의 폭 및/또는 높이와 동일하거나 작아야 한다.
- [0211] 이 예의 실시예들에서, 이웃 샘플들은 특정된 스캐닝 순서에 의해 선택된다. 선택된 샘플들의 수가 지정된 수를 충족하면 선택 프로세스가 종료된다. 좌측 및 상부측 상의 선택된 샘플들의 수는 동일할 수 있다. 좌측의 경우, 스캐닝 순서는 상단에서 하단으로일 수 있으며, 상부측의 경우, 스캐닝 순서는 왼쪽에서 오른쪽으로일 수 있다. 도 25 및 도 31을 참조하면, 선택된 이웃 샘플들은 어두운 원들로 강조된다. 도 25에서, 이웃 샘플들은 다운-샘플링되지 않으며, 도 31에서, 이웃 샘플들은 다운-샘플링된다. 대안적으로, 이웃 샘플들은 반전된 스캐닝 순서에 의해 선택될 수 있다. 즉, 좌측의 경우, 스캐닝 순서는 하단에서 상단으로일 수 있으며, 상부측의 경우, 스캐닝 순서는 오른쪽에서 왼쪽으로일 수 있다. 도 26, 도 32 및 도 33을 참조하면, 선택된 샘플들의 위치들이 도시된다. 도 26에서, 이웃 샘플들은 다운-샘플링되지 않으며, 도 32 및 도 33에서, 이웃 샘플들은 다운-샘플링된다.
- [0212] 이 예의 실시예들에서, 현재 블록의 폭 및 높이 둘 다가 T_h 보다 클 때, 좌측으로부터의 K 개 이상의 샘플 쌍 및

상부측으로부터의 2개 이상의 샘플 쌍이 선형 모델 파라미터들 a 및 b를 계산하기 위해 사용된다. 사용된 이웃 샘플 쌍들의 위치들은 미리 정의되고 고정된다. T_h 는 2, 4, 또는 8과 같은 양의 정수이다. K 는 2, 4, 8과 같은 양의 정수이다.

- [0213] 이 예의 실시예들에서, 왼쪽 또는 상부의 이웃 샘플들이 이용가능하지 않을 때, 이용가능한 측으로부터의 K 개의 샘플이 파라미터들 a 및 b를 도출하기 위해 사용된다.
- [0214] 또 다른 예에서, T_{CCLM} 또는 L_{CCLM} 과 같이 좌측 또는 상부측 이웃 샘플들만을 이용하는 선형 모델의 경우, T_{CCLM} 에 대한 상부측 또는 L_{CCLM} 에 대한 좌측과 같은 사용된 측의 크기가 2와 동일할 때, M 개의 샘플만이 이용된다. M 은 2 또는 4와 같은 양의 정수이고, 위치들은 미리 정의된다. 그렇지 않으면, 최대 N 개의 샘플이 사용된다. N 은 4 또는 8과 같은 양의 정수이다. M 은 N 과 동일하지 않을 수 있다.
- [0215] 이 예의 실시예들에서, 도 34를 참조하면, 사용된 측의 크기가 2보다 클 때, 미리 정의된 위치들이 도시되고, 여기서 상부측이 사용된다.
- [0216] 이 예의 실시예들에서, 도 35 및 도 36을 참조하면, 사용된 측의 크기가 2와 동일할 때, 미리 정의된 위치들이 도시되며, 여기서 도 35에서 상부측이 사용되고 도 36에서 좌측이 사용된다.
- [0217] 또한, 제안된 방법들은 처리 회로(예를 들어, 하나 이상의 프로세서 또는 하나 이상의 집적 회로)에 의해 구현될 수 있다. 일 예에서, 하나 이상의 프로세서는 제안된 방법들 중 하나 이상을 수행하기 위해 비일시적 컴퓨터 판독가능 매체에 저장되는 프로그램을 실행한다.
- [0218] 위에서 설명된 기법들은 컴퓨터 판독가능 명령어들을 이용하여 컴퓨터 소프트웨어로서 구현되고 하나 이상의 컴퓨터 판독가능 매체에 물리적으로 저장될 수 있다. 예를 들어, 도 37은 개시된 주제의 특정 실시예들을 구현하기에 적합한 컴퓨터 시스템(3700)을 도시한다.
- [0219] 컴퓨터 소프트웨어는, 컴퓨터 중앙 처리 유닛(CPU), 그래픽 처리 유닛(GPU) 등에 의해, 직접적으로, 또는 해석, 마이크로코드 실행 등을 통해 실행될 수 있는 명령어들을 포함하는 코드를 생성하기 위해, 어셈블리, 컴파일러, 링킹, 또는 유사한 메커니즘을 겪을 수 있는 임의의 적절한 머신 코드 또는 컴퓨터 언어를 사용하여 코딩될 수 있다.
- [0220] 명령어들은, 예를 들어, 개인용 컴퓨터, 태블릿 컴퓨터, 서버, 스마트폰, 게이밍 디바이스, 사물 인터넷 디바이스 등을 포함하여, 다양한 타입의 컴퓨터들 또는 그것의 컴포넌트들 상에서 실행될 수 있다.
- [0221] 컴퓨터 시스템(3700)에 대한 도 37에 도시된 컴포넌트들은 사실상 예시적인 것이고, 본 개시내용의 실시예들을 구현하는 컴퓨터 소프트웨어의 사용 또는 기능성의 범위에 대한 임의의 제한을 시사하도록 의도되지 않는다. 컴포넌트들의 구성이 컴퓨터 시스템(3700)의 예시적인 실시예에서 예시된 컴포넌트들 중 임의의 하나 또는 이들의 조합과 관련한 임의의 종속성 또는 요건을 갖는 것으로 해석해서도 안 된다.
- [0222] 컴퓨터 시스템(3700)은 특정 인간 인터페이스 입력 디바이스들을 포함할 수 있다. 그러한 인간 인터페이스 입력 디바이스는, 예를 들어, 촉각 입력(예컨대: 키스트로크들, 스위치들, 데이터 글러브 모션들), 오디오 입력(예컨대: 음성, 박수), 시각적 입력(예컨대 제스처들), 후각적 입력(묘사되지 않음)을 통한 하나 이상의 인간 사용자에게 의한 입력에 응답할 수 있다. 인간 인터페이스 디바이스들은 또한 오디오(예컨대: 음성, 음악, 주변 사운드), 이미지들(예컨대: 스캐닝된 이미지들, 스틸 이미지 카메라로부터 획득된 사진 이미지들), 비디오(예컨대 2차원 비디오, 입체적 비디오를 포함하는 3차원 비디오)와 같은, 인간에 의한 의식적인 입력과 반드시 직접적으로 관련되는 것은 아닌 특정 미디어를 캡처하기 위해 사용될 수 있다.
- [0223] 입력 인간 인터페이스 디바이스들은: 키보드(3701), 마우스(3702), 트랙패드(3703), 터치 스크린(3710), 데이터-글러브(3704), 조이스틱(3705), 마이크로폰(3706), 스캐너(3707), 카메라(3708) 중 하나 이상(각각의 하나만이 묘사됨)을 포함할 수 있다.
- [0224] 컴퓨터 시스템(3700)은 또한 특정 인간 인터페이스 출력 디바이스들을 포함할 수 있다. 그러한 인간 인터페이스 출력 디바이스들은, 예를 들어, 촉각 출력, 사운드, 광, 및 냄새/맛을 통해 하나 이상의 인간 사용자의 감각들을 자극하고 있을 수 있다. 그러한 인간 인터페이스 출력 디바이스들은 촉각 출력 디바이스들(예를 들어, 터치-스크린(3710), 데이터-글러브(3704), 또는 조이스틱(3705)에 의한 촉각 피드백이지만, 입력 디바이스들로서 역할하지 않는 촉각 피드백 디바이스들도 있을 수 있음), 오디오 출력 디바이스들(예컨대: 스피커들(3709), 헤드폰들(묘사되지 않음)), 시각적 출력 디바이스들(예컨대 음극선관(CRT) 스크린들, 액정 디스플레이(LCD) 스크린들, 플라즈마 스크린들, 유기 발광 다이오드(OLED) 스크린들을 포함하는 스크린들(3710), 각각은 터치-스크린

입력 능력이 있거나 없고, 각각은 촉각 피드백 능력이 있거나 없고 - 이들 중 일부는 스테레오그래픽 출력과 같은 수단을 통해 2차원 시각적 출력 또는 3개보다 많은 차원의 출력을 출력할 수 있음 -; 가상 현실 안경(묘사되지 않음), 홀로그래픽 디스플레이 및 연기 탱크(묘사되지 않음), 및 프린터(묘사되지 않음)를 포함할 수 있다.

[0225] 컴퓨터 시스템(3700)은 인간 액세스 가능한 저장 디바이스들 및 그것들과 연관된 매체들, 예컨대 CD/DVD 등의 매체(3721)를 갖는 CD/DVD ROM/RW(3720)를 포함하는 광학 매체, 썸-드라이브(3722), 이동식 하드 드라이브 또는 솔리드 스테이트 드라이브(3723), 테이프 및 플로피 디스크(묘사되지 않음)와 같은 레거시 자기 매체, 보안 동글(묘사되지 않음)과 같은 특수화된 ROM/ASIC/PLD 기반 디바이스들 등을 또한 포함할 수 있다.

[0226] 본 기술분야의 통상의 기술자들은 현재 개시된 주제와 관련하여 사용되는 용어 "컴퓨터 판독가능 매체(computer readable media)"가 송신 매체, 반송파들, 또는 다른 일시적 신호들을 포함하지 않는다는 점을 또한 이해해야 한다.

[0227] 컴퓨터 시스템(3700)은 또한 하나 이상의 통신 네트워크에 대한 인터페이스(들)를 포함할 수 있다. 네트워크들은 예를 들어, 무선, 와이어라인, 광학적일 수 있다. 네트워크들은 추가로 로컬, 광역, 대도시, 차량 및 산업, 실시간, 지연-허용(delay-tolerant) 등일 수 있다. 네트워크들의 예들은 로컬 영역 네트워크들, 예컨대 이더넷, 무선 LAN들, GSM(global systems for mobile communications), 제3 세대(3G), 제4 세대(4G), 제5 세대(5G), 롱-텀 에볼루션(LTE) 등을 포함하는 셀룰러 네트워크들, 케이블 TV, 위성 TV 및 지상파 브로드캐스트 TV를 포함하는 TV 와이어라인 또는 무선 광역 디지털 네트워크들, CANBus를 포함하는 차량 및 산업 등을 포함한다. 특정 네트워크들은 흔히 외부 네트워크(3755)와 통신하기 위해 특정 범용 데이터 포트들 또는 주변 버스들(3749)(예컨대, 예를 들어, 컴퓨터 시스템(3700)의 USB(universal serial bus) 포트들)에 부착된 외부 네트워크 인터페이스 어댑터들(3754)을 요구한다; 다른 것들은 흔히 아래에 설명되는 바와 같은 시스템 버스의 부착에 의해 컴퓨터 시스템(3700)의 코어에 통합된다(예를 들어, PC 컴퓨터 시스템으로의 이더넷 인터페이스 또는 스마트폰 컴퓨터 시스템으로의 셀룰러 네트워크 인터페이스). 이러한 네트워크들 중 임의의 것을 사용하여, 컴퓨터 시스템(3700)은 다른 엔티티들과 통신할 수 있다. 그러한 통신은 단방향성 수신 전용(예를 들어, 브로드캐스트 TV), 단방향성 송신 전용(예를 들어, CANbus 대 특정 CANbus 디바이스들), 또는 예를 들어, 로컬 또는 광역 디지털 네트워크들을 사용하는 다른 컴퓨터 시스템들과의 양방향성일 수 있다. 위에서 설명된 바와 같은 네트워크들 및 네트워크 인터페이스들 각각에 대해 특정 프로토콜들 및 프로토콜 스택들이 사용될 수 있다.

[0228] 전술한 인간 인터페이스 디바이스들, 인간 액세스 가능한 저장 디바이스들, 및 네트워크 인터페이스들은 컴퓨터 시스템(3700)의 코어(3740)에 부착될 수 있다.

[0229] 코어(3740)는 하나 이상의 중앙 처리 유닛(CPU)(3741), 그래픽 처리 유닛(GPU)(3742), 필드 프로그래머블 게이트 영역(FPGA)(3743)의 형식으로 특수화된 프로그래머블 처리 유닛들, 특정 태스크들에 대한 하드웨어 가속기들(3744) 등을 포함할 수 있다. 이들 디바이스는, 판독 전용 메모리(ROM)(3745), 랜덤 액세스 메모리(RAM)(3746), 내부 비-사용자 액세스 가능 하드 드라이브들, 솔리드-스테이트 드라이브들(SSD들) 등과 같은 내부 대용량 저장소(3747)와 함께, 시스템 버스(3748)를 통해 접속될 수 있다. 일부 컴퓨터 시스템들에서, 시스템 버스(3748)는 추가적인 CPU들, GPU들 등에 의한 확장을 가능하게 하기 위해 하나 이상의 물리적 플러그의 형태로 액세스 가능할 수 있다. 주변 디바이스들은 코어의 시스템 버스(3748)에 직접적으로, 또는 주변 버스(3749)를 통해 부착될 수 있다. 주변 버스를 위한 아키텍처들은 PCI(peripheral component interconnect), USB 등을 포함한다.

[0230] CPU들(3741), GPU들(3742), FPGA들(3743), 및 가속기들(3744)은, 조합하여, 전술한 컴퓨터 코드를 구성할 수 있는 특정 명령어들을 실행할 수 있다. 그 컴퓨터 코드는 ROM(3745) 또는 RAM(3746)에 저장될 수 있다. 과도적인 데이터가 또한 RAM(3746)에 저장될 수 있는 반면, 영구 데이터는, 예를 들어, 내부 대용량 저장소(3747)에 저장될 수 있다. 메모리 디바이스들 중 임의의 것에 대한 고속 저장 및 검색은 하나 이상의 CPU(3741), GPU(3742), 대용량 저장소(3747), ROM(3745), RAM(3746) 등과 밀접하게 연관될 수 있는 캐시 메모리의 사용을 통해 가능하게 될 수 있다.

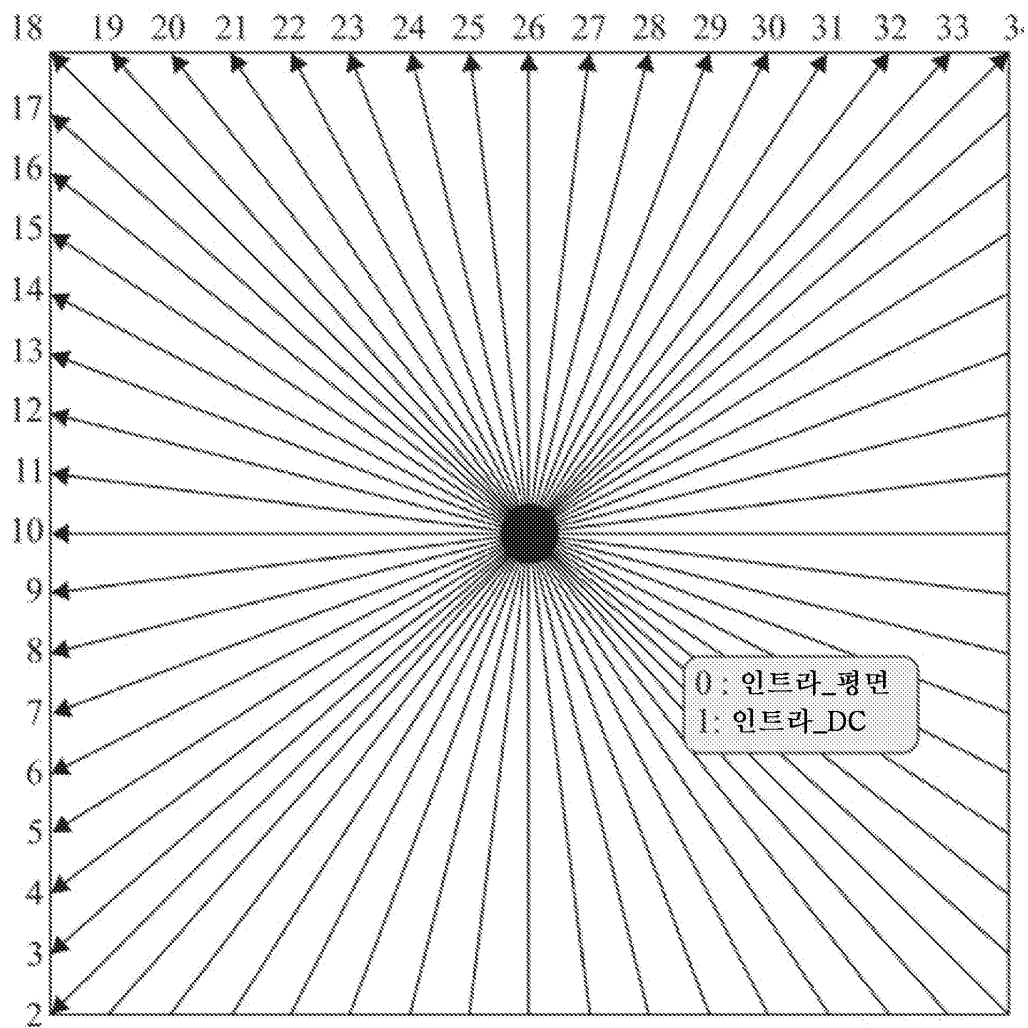
[0231] 컴퓨터 판독가능 매체는 다양한 컴퓨터 구현 동작들을 수행하기 위한 컴퓨터 코드를 그 상에 가질 수 있다. 매체 및 컴퓨터 코드는 본 개시내용의 목적을 위해 특별히 설계되고 구성된 것들일 수 있거나, 또는 그것들은 컴퓨터 소프트웨어 기술분야의 기술자들에게 잘 알려져 있고 이용가능한 종류의 것일 수 있다.

[0232] 제한이 아니라 예로서, 아키텍처를 갖는 컴퓨터 시스템(3700), 및 구체적으로 코어(3740)는 프로세서(들)(CPU들, GPU들, FPGA, 가속기들 등을 포함함)가 하나 이상의 유형의(tangible) 컴퓨터 판독가능 매체에 구현된 소프트웨어를 실행하는 결과로서 기능성을 제공할 수 있다. 그러한 컴퓨터 판독가능 매체는 위에 소개된 바와 같은 사용자-액세스 가능 대용량 저장소뿐만 아니라, 코어-내부 대용량 저장소(3747) 또는 ROM(3745)과 같은 비밀리적인 본질의 것인 코어(3740)의 특정 저장소와 연관된 매체일 수 있다. 본 개시내용의 다양한 실시예들을 구현하는 소프트웨어가 이러한 디바이스들에 저장되고 코어(3740)에 의해 실행될 수 있다. 컴퓨터 판독가능 매체는 특정 필요에 따라 하나 이상의 메모리 디바이스 또는 칩을 포함할 수 있다. 소프트웨어는 코어(3740) 및 구체적으로 그 안의 프로세서들(CPU, GPU, FPGA 등을 포함함)로 하여금, RAM(3746)에 저장된 데이터 구조들을 정의하는 것 및 소프트웨어에 의해 정의된 프로세스들에 따라 그러한 데이터 구조들을 수정하는 것을 포함하여, 본 명세서에 설명된 특정 프로세스들 또는 특정 프로세스들의 특정 부분들을 실행하게 야기할 수 있다. 추가로 또는 대안으로서, 컴퓨터 시스템은, 본 명세서에 설명된 특정 프로세스들 또는 특정 프로세스들의 특정 부분들을 실행하기 위해 소프트웨어 대신에 또는 그와 함께 동작할 수 있는, 회로(예를 들어: 가속기(3744))에 하드와이어링되거나 다른 방식으로 구현된 로직의 결과로서 기능성을 제공할 수 있다. 소프트웨어에 대한 참조는 로직을 포함할 수 있고, 적절한 경우, 그 반대로도 가능하다. 컴퓨터 판독가능 매체에 대한 참조는, 적절한 경우, 실행을 위한 소프트웨어를 저장하는 회로(예컨대 집적 회로(IC)), 또는 실행을 위한 로직을 구현하는 회로, 또는 둘 다를 포함할 수 있다. 본 발명은 하드웨어 및 소프트웨어의 임의의 적합한 조합을 포함한다.

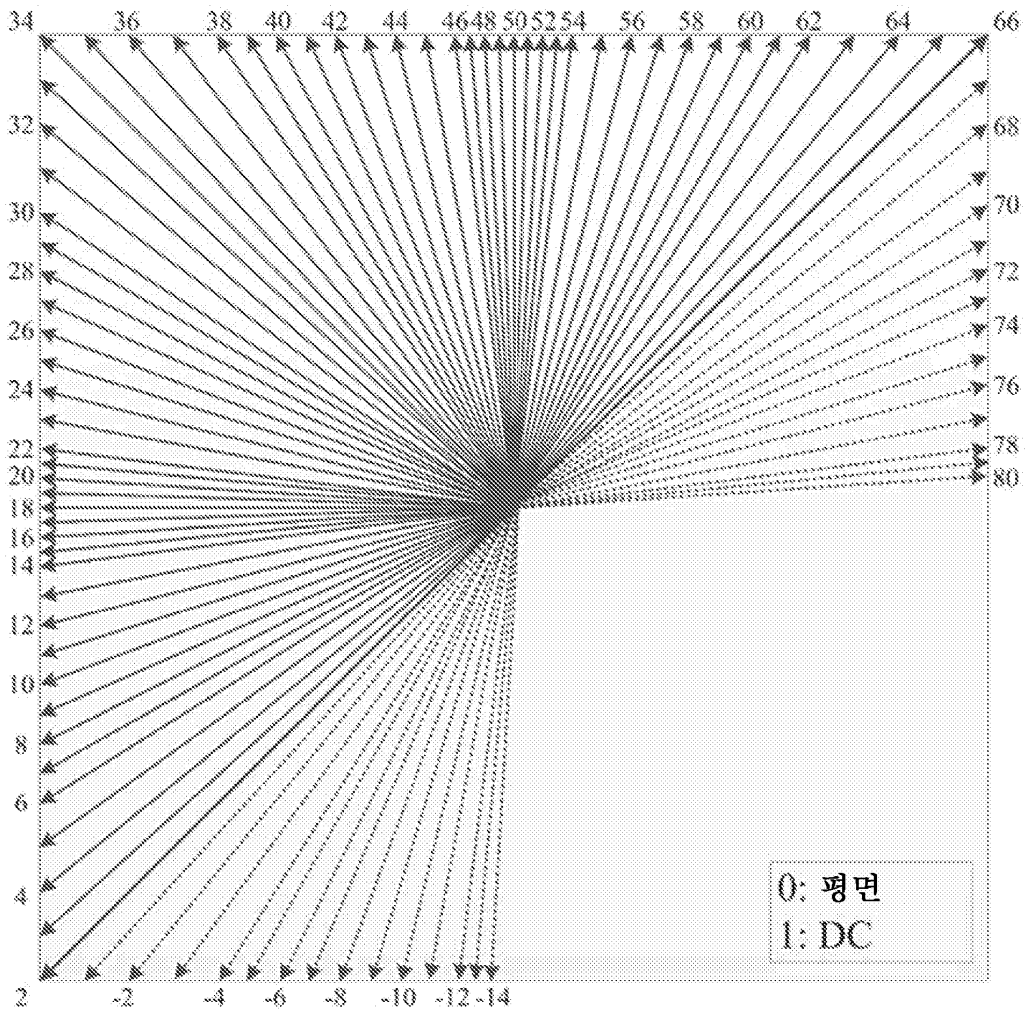
[0233] 본 개시내용이 여러 예시적인 실시예들을 설명하였지만, 본 개시내용의 범위 내에 속하는 변경들, 치환들, 및 다양한 대체 균등물들이 존재한다. 따라서, 본 기술분야의 통상의 기술자들은, 비록 본 명세서에 명시적으로 도시되거나 설명되지는 않았지만, 본 개시내용의 원리들을 구현하고 따라서 그것의 사상 및 범위 내에 있는, 다수의 시스템 및 방법들을 고안할 수 있을 것이라는 점이 인정될 것이다.

도면

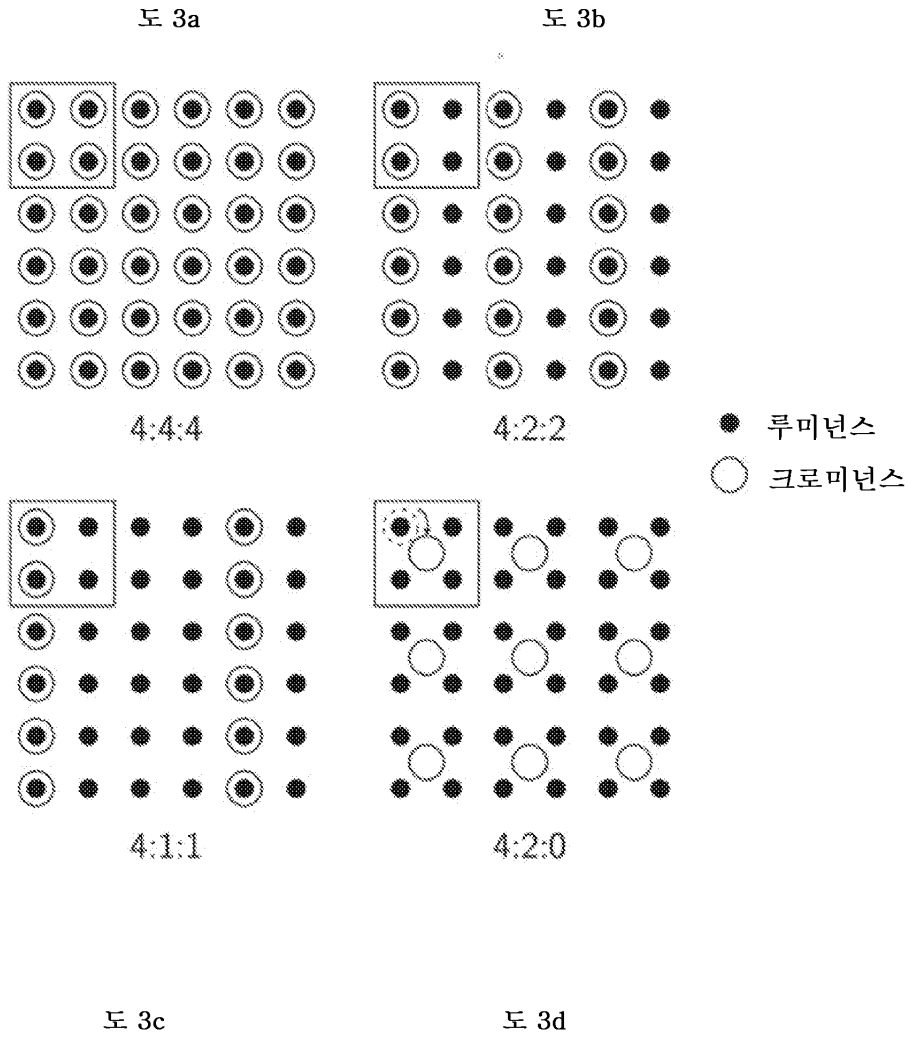
도면1



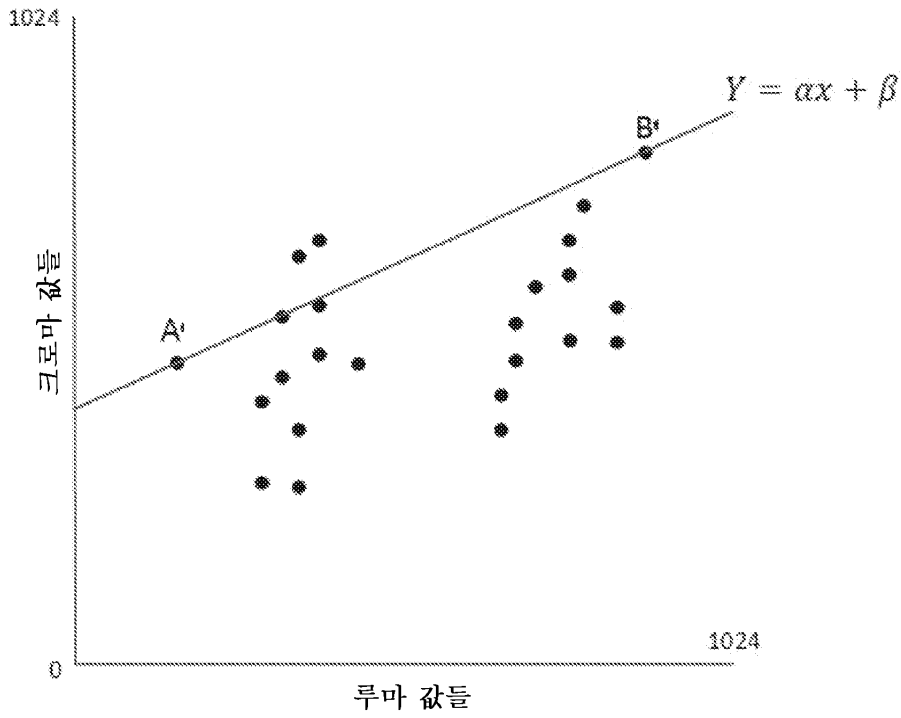
도면2



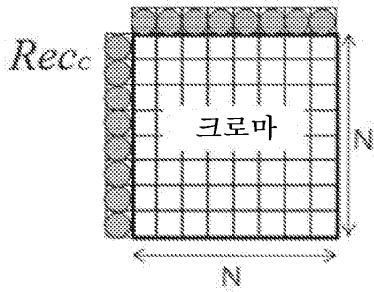
도면3



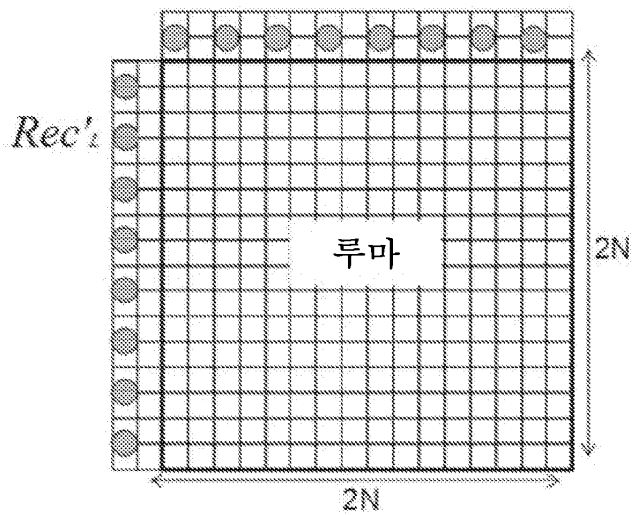
도면4



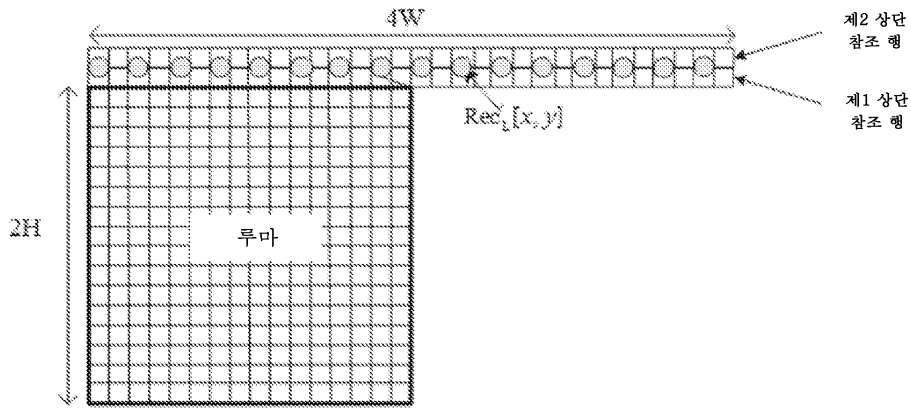
도면5a



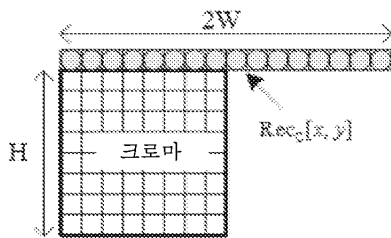
도면5b



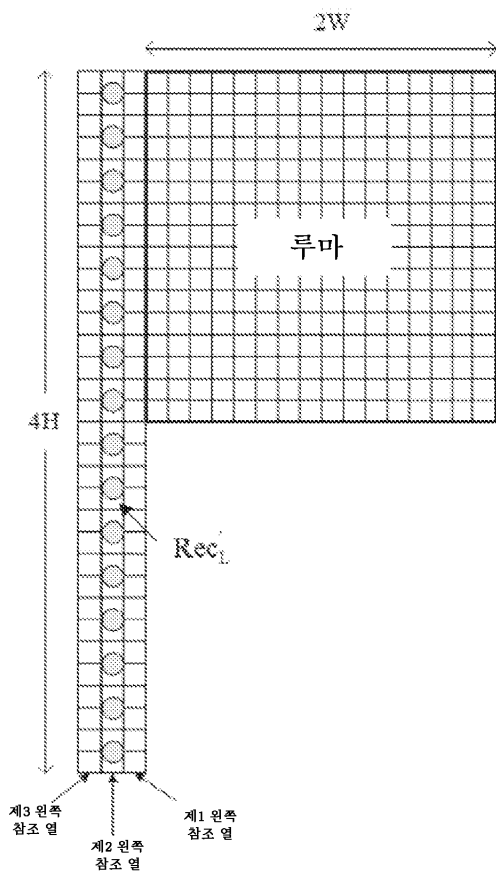
도면6a



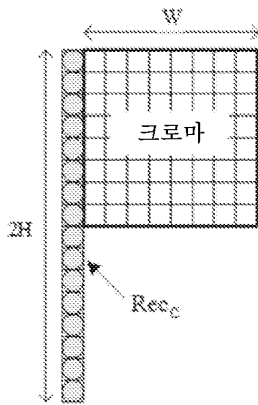
도면6b



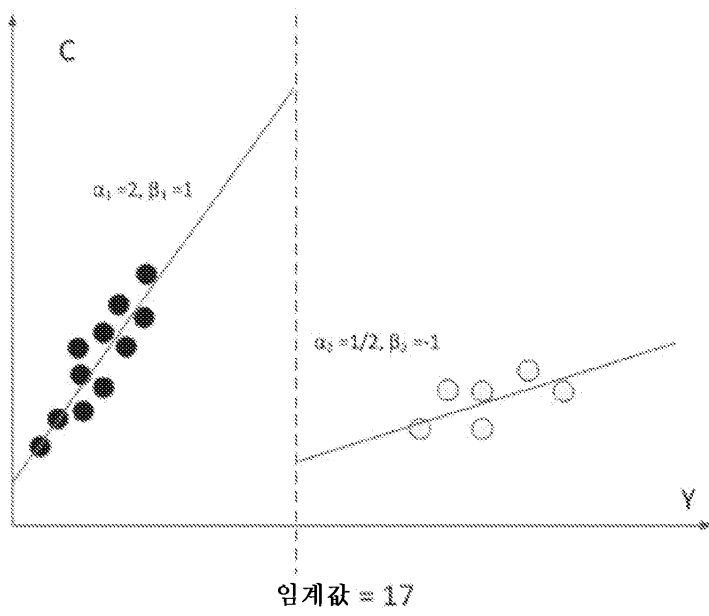
도면7a



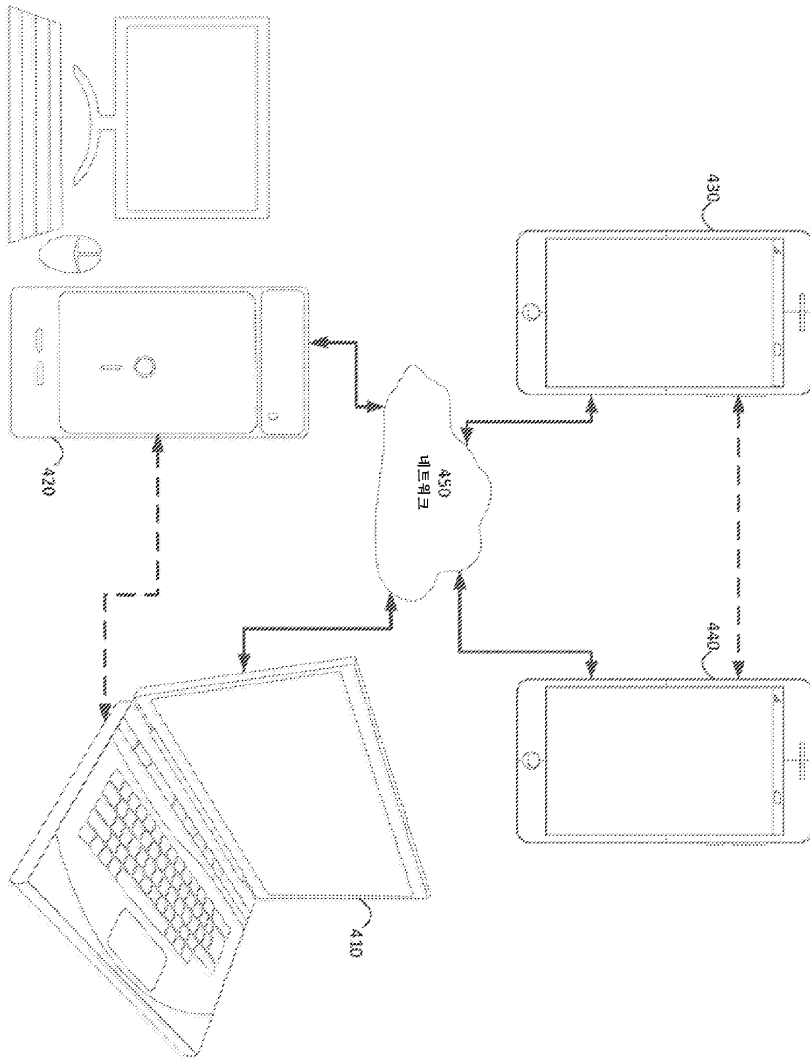
도면7b



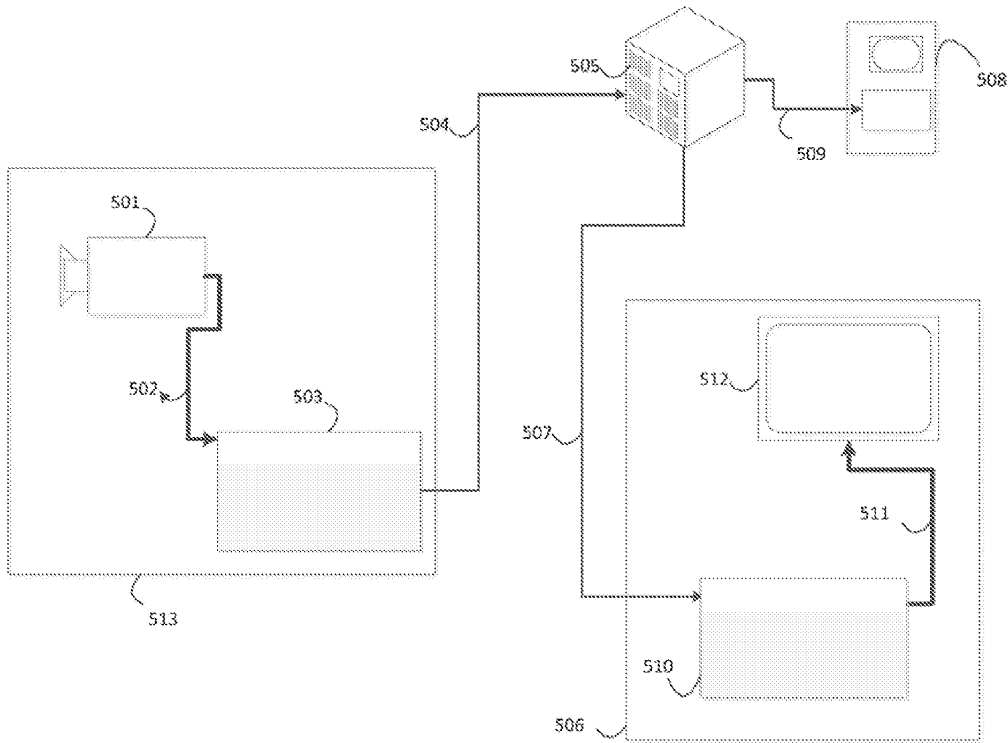
도면8



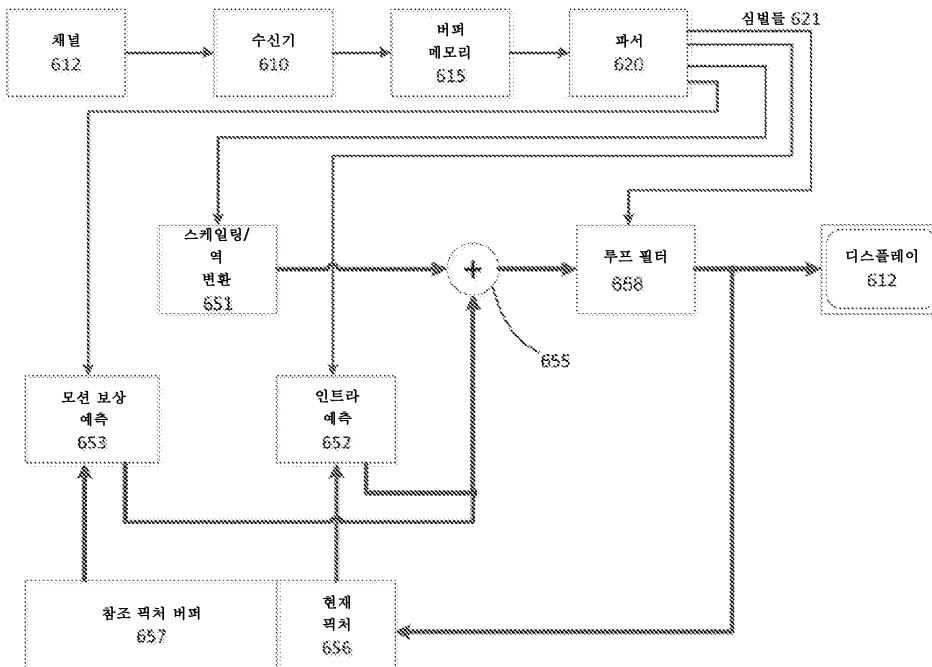
도면9



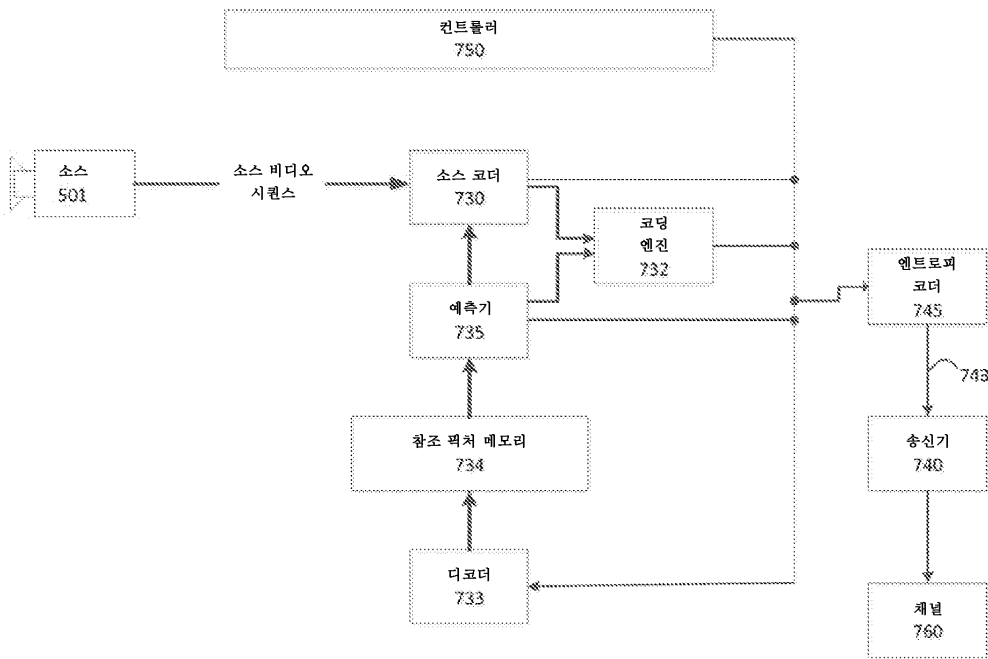
도면10



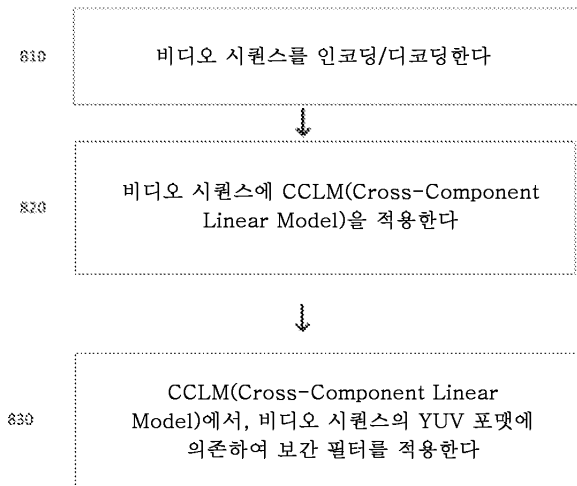
도면11



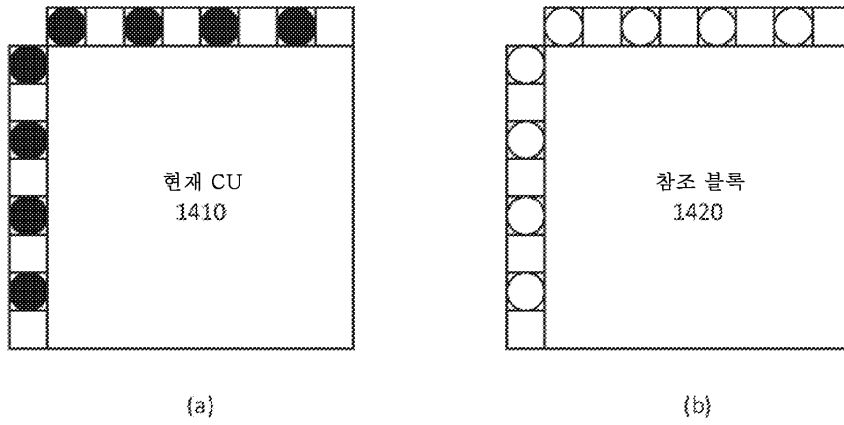
도면12



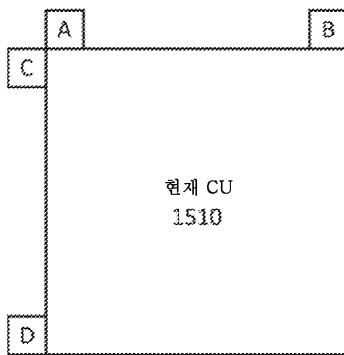
도면13



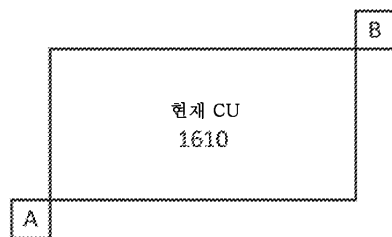
도면14



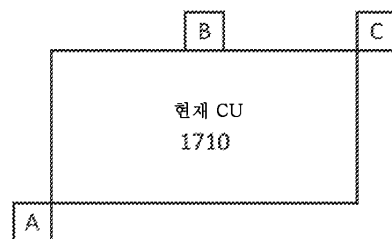
도면15



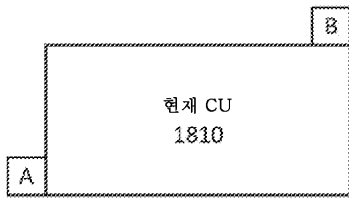
도면16



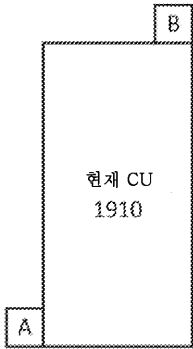
도면17



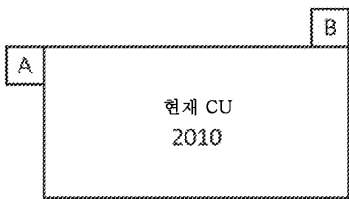
도면18



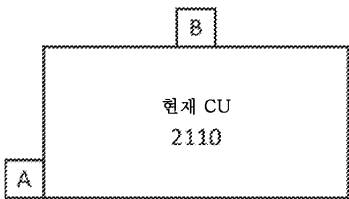
도면19



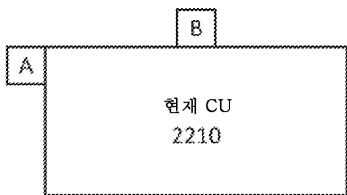
도면20



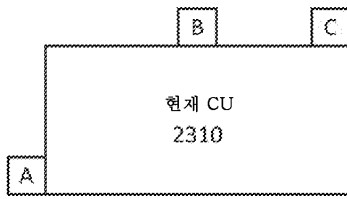
도면21



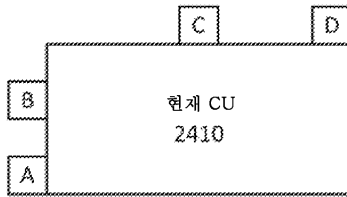
도면22



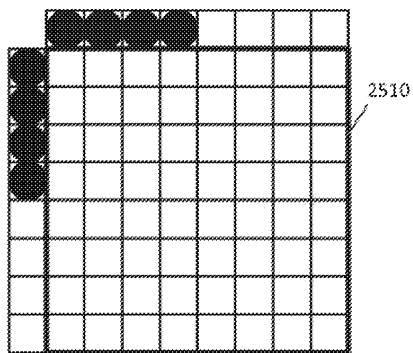
도면23



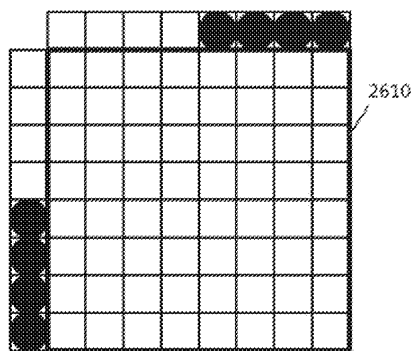
도면24



도면25



도면26



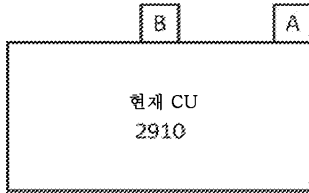
도면27



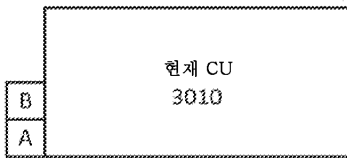
도면28

X		W		X
X		W		X

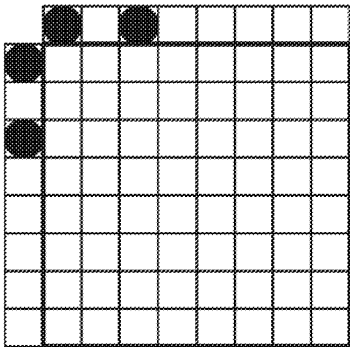
도면29



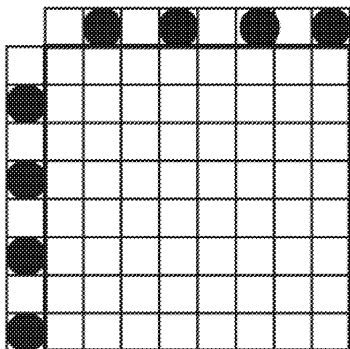
도면30



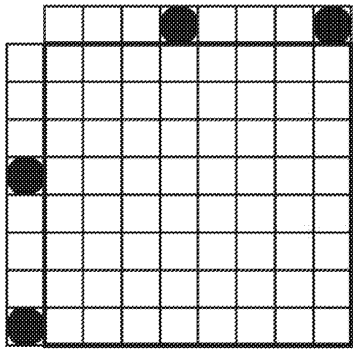
도면31



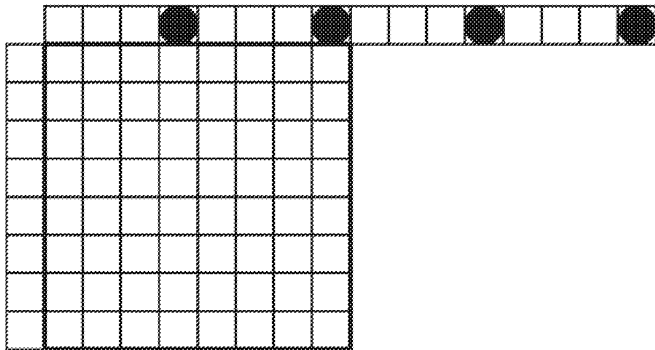
도면32



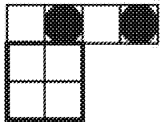
도면33



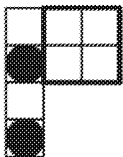
도면34



도면35



도면36



도면37

