



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2025-0035604
(43) 공개일자 2025년03월12일

- (51) 국제특허분류(Int. Cl.)
H04N 19/86 (2014.01) H04N 19/117 (2014.01)
H04N 19/13 (2014.01) H04N 19/139 (2014.01)
H04N 19/176 (2014.01) H04N 19/82 (2014.01)
H04N 19/96 (2014.01)
- (52) CPC특허분류
H04N 19/86 (2015.01)
H04N 19/117 (2015.01)
- (21) 출원번호 10-2025-7006333(분할)
(22) 출원일자(국제) 2019년03월29일
심사청구일자 없음
(62) 원출원 특허 10-2020-7031293
원출원일자(국제) 2019년03월29일
심사청구일자 2022년03월15일
(85) 번역문제출일자 2025년02월25일
(86) 국제출원번호 PCT/US2019/024936
(87) 국제공개번호 WO 2019/191646
국제공개일자 2019년10월03일
(30) 우선권주장
62/650,252 2018년03월29일 미국(US)
16/370,140 2019년03월29일 미국(US)
- (71) 출원인
앨리스 엔터프라이즈 엘엘씨
미국 텔라웨어 19810 월링턴 테트널 빌딩 #104 실
버사이드 로드 3411
(72) 발명자
베이론, 데이빗 엠.
미국 92130 캘리포니아주 샌디에고 미로 서클
11413
루쓰라, 에이제이
미국 92130 캘리포니아주 샌디에고 윈스턴리 웨이
13695
(74) 대리인
양영준, 김연송, 백만기

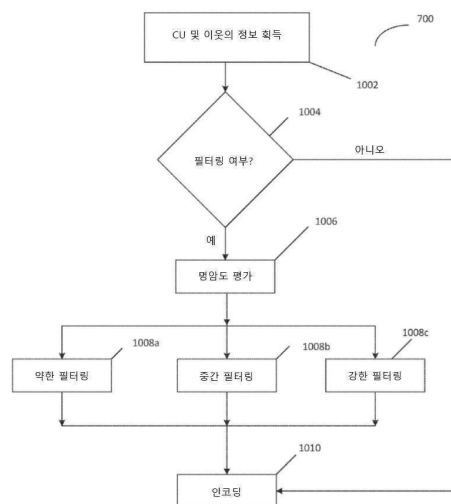
전체 청구항 수 : 총 3 항

(54) 발명의 명칭 HDR 콘텐츠를 디블록킹하기 위한 시스템 및 방법

(57) 요약

픽셀 명암도가 증가할 때 필터링 수준이 증가하도록, 코딩 유닛과 관련된 픽셀 명암도에 적어도 부분적으로 기초한, 변형된 필터링을 사용함으로써 디블록킹 아티팩트가 감소하는, 비디오를 코딩하기 위한 시스템 및 방법. 일부 구현예에서, 디블록킹 필터 파라미터와 관련된 파라미터에 대한 오프셋 값 또는 오프셋 값의 표시자는 명암도 값에 적어도 부분적으로 기초한 필터링 수준과 관련이 있을 수 있다.

대표도 - 도10



(52) CPC특허분류

H04N 19/13 (2015.01)

H04N 19/139 (2015.01)

H04N 19/176 (2015.01)

H04N 19/82 (2015.01)

H04N 19/96 (2015.01)

명세서

청구범위

청구항 1

비디오를 디코딩하는 방법으로서,

- (a) 분할 구조에 따라 코딩 유닛들로 분할된 코딩 트리 유닛들에 기초하여 코딩된, 인코딩된 비디오의 비트스트림을 수신하는 단계 - 상기 분할 구조는 쿼드트리 분할로 분할되는 정방형 부모 블록을 포함하고, 상기 쿼드트리 분할은 정방형 부모 노드를 수평 및 수직 방향 모두에서 반으로 분할하여 각각이 같은 크기를 갖는 정방형 형태의 리프 노드들로서 네 개의 블록들을 정의함 -;
- (b) 상기 인코딩된 비디오의 상기 비트스트림을 디코딩하는 단계;
- (c) 상기 인코딩된 비트스트림 내의 루마 성분들의 블록 및 크로마 성분들의 블록으로서 코딩되는, 양자화를 이용하여 코딩되는 직사각형 코딩 유닛을 결정하는 단계 - 상기 직사각형 코딩 유닛은 폭 및 높이를 갖고, 상기 폭은 상기 높이와는 상이하고, 직사각형 코딩 유닛들 각각은 인터-픽처 예측 또는 인트라-픽처 예측을 수행할지 여부에 대한 각각의 결정 지점임 -;
- (d) 상기 직사각형 코딩 유닛의 수직 경계 또는 수평 경계와 관련된 픽셀들의 명암도 정보(intensity information)를 결정하는 단계;
- (e) 상기 직사각형 코딩 유닛과 관련된 상기 명암도 정보에 적어도 부분적으로 기초하여 상기 직사각형 코딩 유닛에 디블록킹 필터링을 적용하는 단계를 포함하고, 상기 적용된 디블록킹 필터링은 양자화 파라미터들의 오프셋에 기초하여 선택적으로 수정되는 경계 필터링을 지정하는 필터링 파라미터들 β 및 t_c 에 기초하고, 상기 오프셋은 상기 경계와 관련된 픽셀들의 상기 결정된 명암도 정보에 적어도 부분적으로 기초하며;
- (f) 더 큰 상기 오프셋은 그러한 더 큰 오프셋이 없을 때 발생하는 것보다 더 강한 필터링을 야기하며;
- (g) 상기 오프셋은 필터링 파라미터들을 위한 상이한 오프셋들을 포함하고;
- (h) 상기 디블록킹 필터링은 클리핑에 더 기초하고; 및
- (i) 상기 클리핑은 상기 양자화 파라미터들에 기초하는, 방법.

청구항 2

인코더에 의해 비디오를 인코딩하는 방법으로서,

- (a) 분할 구조에 따라 코딩 유닛들로 분할된 코딩 트리 유닛들에 기초하여 코딩된, 인코딩된 비디오의 비트스트림을 제공하는 단계 - 상기 분할 구조는 쿼드트리 분할로 분할되는 정방형 부모 블록을 포함하고, 상기 쿼드트리 분할은 정방형 부모 노드를 수평 및 수직 방향 모두에서 반으로 분할하여 각각이 같은 크기를 갖는 정방형 형태의 리프 노드들로서 네 개의 블록들을 정의함 -;
- (b) 상기 비트스트림을 인코딩하는 단계를 포함하고,
- (c) 상기 인코딩하는 단계는 상기 인코딩된 비트스트림 내의 루마 성분들의 블록 및 크로마 성분들의 블록으로서 코딩되는, 양자화를 이용하여 코딩되는 직사각형 코딩 유닛을 결정하는 것에 기초하고, 상기 직사각형 코딩 유닛은 폭 및 높이를 갖고, 상기 폭은 상기 높이와는 상이하고, 직사각형 코딩 유닛들 각각은 인터-픽처 예측 또는 인트라-픽처 예측을 수행할지 여부에 대한 각각의 결정 지점이며;
- (d) 상기 인코딩하는 단계는 상기 직사각형 코딩 유닛의 수직 경계 또는 수평 경계와 관련된 픽셀들의 명암도 정보(intensity information)를 결정하는 것에 기초하고;
- (e) 상기 인코딩하는 단계는 상기 직사각형 코딩 유닛과 관련된 상기 명암도 정보에 적어도 부분적으로 기초하여 상기 직사각형 코딩 유닛에 디블록킹 필터링을 적용하는 것에 기초하고, 상기 적용된 디블록킹 필터링은 양자화 파라미터들의 오프셋에 기초하여 선택적으로 수정되는 경계 필터링을 지정하는 필터링 파라미터들 β 및 t_c

에 기초하고, 상기 오프셋은 상기 경계와 관련된 픽셀들의 상기 결정된 명암도 정보에 적어도 부분적으로 기초하며;

(f) 상기 인코딩하는 단계는 더 큰 상기 오프셋이 그러한 더 큰 오프셋이 없을 때 발생하는 것보다 더 강한 필터링을 야기하는 것에 기초하고;

(g) 상기 인코딩하는 단계는 상기 오프셋이 필터링 파라미터들에 대한 상이한 오프셋들을 포함하는 것에 기초하고;

(h) 상기 인코딩하는 단계는 상기 디블록킹 필터링이 클리핑에 더 기초하는 것에 기초하고; 및

(i) 상기 인코딩하는 단계는 상기 클리핑이 상기 양자화 파라미터들에 기초하는 것에 기초하는, 방법.

청구항 3

인코딩된 비디오의 비트스트림을 저장하는 컴퓨터 판독가능 저장 장치로서,

(a) 상기 비트스트림은 상기 인코딩된 비디오가 분할 구조에 따라 코딩 유닛들로 분할된 코딩 트리 유닛들에 기초하여 어떻게 코딩될지를 표시하는 데이터를 포함하고 - 상기 분할 구조는 쿼드트리 분할로 분할되는 정방형 부모 블록을 포함하고, 상기 쿼드트리 분할은 정방형 부모 노드를 수평 및 수직 방향 모두에서 반으로 분할하여 각각이 같은 크기를 갖는 정방형 형태의 리프 노드들로서 네 개의 블록들을 정의함 -;

(b) 상기 비트스트림은 상기 인코딩된 비디오의 상기 비트스트림을 어떻게 디코딩할지를 표시하는 데이터를 포함하고;

(c) 상기 비트스트림은 상기 인코딩된 비트스트림 내의 루마 성분들의 블록 및 크로마 성분들의 블록으로서 코딩되는, 양자화를 이용하여 코딩되는 직사각형 코딩 유닛을 어떻게 결정할지를 표시하는 데이터를 포함하고, 상기 직사각형 코딩 유닛은 폭 및 높이를 갖고, 상기 폭은 상기 높이와는 상이하고, 직사각형 코딩 유닛들 각각은 인터-픽처 예측 또는 인트라-픽처 예측을 수행할지 여부에 대한 각각의 결정 지점이며;

(d) 상기 비트스트림은 상기 직사각형 코딩 유닛의 수직 경계 또는 수평 경계와 관련된 픽셀들의 명암도 정보(intensity information)를 어떻게 결정할지를 표시하는 데이터를 포함하고;

(e) 상기 비트스트림은 상기 직사각형 코딩 유닛과 관련된 상기 명암도 정보에 적어도 부분적으로 기초하여 상기 직사각형 코딩 유닛에 디블록킹 필터링을 어떻게 적용할지를 표시하는 데이터를 포함하고, 상기 적용된 디블록킹 필터링은 양자화 파라미터들의 오프셋에 기초하여 선택적으로 수정되는 경계 필터링을 지정하는 필터링 파라미터들 β 및 t_c 에 기초하고, 상기 오프셋은 상기 경계와 관련된 픽셀들의 상기 결정된 명암도 정보에 적어도 부분적으로 기초하며;

(f) 더 큰 상기 오프셋은 그러한 더 큰 오프셋이 없을 때 발생하는 것보다 더 강한 필터링을 야기하며;

(g) 상기 오프셋은 필터링 파라미터들의 상이한 오프셋들을 포함하고;

(h) 상기 디블록킹 필터링은 클리핑에 더 기초하고; 및

(i) 상기 클리핑은 상기 양자화 파라미터들에 기초하는, 컴퓨터 판독가능 저장 장치.

발명의 설명

기술 분야

[0001]

우선권 주장

[0002]

본 출원은 35 U.S.C. § 119(e)에 의거하여 2018년 3월 29일에 선 출원된 미국 특허 가출원 일련번호 제 62/650,252호에 대한 우선권을 주장하여, 그 전체는 참조로서 본원에 통합된다.

[0003]

기술분야

[0004]

본 개시는 비디오 코딩 분야에 관한 것으로서, 특히 HDR 콘텐츠를 효율적으로 및 효과적으로 디블록킹하고 필터링하기 위한 시스템 및 방법에 관한 것이다.

배경 기술

[0005] 진화하는 비디오 코딩 표준의 기술적 개선은 더 높은 비트 전송률, 더 높은 해상도 및 더 양호한 비디오 품질을 가능하게 하기 위해 코딩 효율을 높이는 경향을 보여준다. Joint Video Exploration Team은 JVET로 지칭되는 새로운 비디오 코딩 방식을 개발하였으며, 다용도 비디오 코딩(VVC: Versatile Video Coding)으로 지칭되는 더 새로운 비디오 코딩 방식을 개발 중이다(Versatile Video Coding (Draft 2)이라는 제목으로 2018년 10월 1일에 공표된 JVET 표준의 2차 초안에 대한 VVC 7판의 전체 내용은 참조로서 본원에 통합됨). HVEC(High Efficiency Video Coding)과 같은 다른 비디오 코딩 방식과 마찬가지로, JVET와 VVC 둘 다는 블록 기반 하이브리드 공간 및 시간 예측 코딩 방식이다. 그러나, HEVC에 비해 JVET와 VVC는 비트스트림 구조, 선택스, 제약, 및 디코딩된 화상을 생성하기 위한 맵핑에 대한 많은 변형을 포함한다. JVET는 Joint Exploration Model(JEM) 인코더와 디코더에서 구현되었지만, VVC는 2020년 초까지 구현되기 어려울 것으로 예상된다.

[0006] 현재의 비디오 코딩 방식은 이미지 명암도(image intensity)를 고려하지 않고 디블록킹 및 필터링을 구현하여, 결과적으로는 모든 콘텐츠에 걸쳐 균일한 방식으로 콘텐츠 필터링을 구현한다. 그러나, 데이터에 의하면, 콘텐츠의 명암도가 디스플레이 문제를 감소시키기 위해 필요한 또는 바람직한 필터링의 정도 또는 수준에 영향을 줄 수 있는 것으로 밝혀졌다. 따라서, 코딩 유닛의 픽셀 명암도에 적어도 부분적으로 기초한 디블록킹 시스템 및 방법이 필요하다.

발명의 내용

[0007] 하나 이상의 컴퓨터로 이루어진 시스템은, 시스템 상에 설치되어 작동 시 시스템으로 하여금 동작을 수행하게 하는 소프트웨어, 펌웨어, 하드웨어 또는 이들의 조합을 가짐으로써 특정 작업 또는 동작을 수행하도록 구성될 수 있다. 하나 이상의 컴퓨터 프로그램은, 데이터 처리 장치에 의해 실행될 때 장치로 하여금 동작을 수행하게 하는 명령을 포함함으로써 특정 작업 또는 동작을 수행하도록 구성될 수 있다. 이와 같은 일반적인 양태는 코딩 유닛을 결정하는 단계, 상기 코딩 유닛의 경계와 관련이 있는 픽셀의 명암도 정보를 결정하는 단계, 인코딩에 앞서, 상기 코딩 유닛과 관련이 있는 명암도 정보에 적어도 부분적으로 기초한 디블록킹 필터링을 상기 코딩 유닛에 적용하는 단계, 및 전송을 위해 상기 코딩 유닛을 인코딩하는 단계를 포함한다. 본 양태의 다른 구현에는 상응하는 컴퓨터 시스템, 장치, 및 하나 이상의 컴퓨터 저장 장치 상에 기록되어 각각 상기 방법의 동작을 수행하도록 구성된 컴퓨터 프로그램을 포함한다.

[0008] 구현에는 다음 특징 중 하나 이상을 포함할 수도 있다: 비디오를 인코딩하는 방법으로서, 임계 값보다 더 큰 값을 갖는 명암도 정보와 관련된 코딩 유닛에 더 강한 디블록킹 필터링이 적용되는 방법. 비디오를 인코딩하는 방법으로서, 상기 임계 값은 소정의 값인 방법. 비디오를 인코딩하는 방법으로서, 상기 코딩 유닛에 인접한 이웃 코딩 유닛을 식별하는 단계, 상기 이웃 코딩 유닛의 경계와 관련된 픽셀의 명암도 정보를 결정하는 단계, 및 상기 코딩 유닛의 경계와 관련된 픽셀의 상기 명암도 정보와 상기 이웃 코딩 유닛과 관련된 픽셀의 상기 명암도 정보를 비교하는 단계를 추가로 포함하되, 상기 필터링은 상기 코딩 유닛의 경계와 관련된 픽셀의 상기 명암도 정보와 상기 이웃 코딩 유닛과 관련된 픽셀의 상기 명암도 정보를 비교하는 상기 단계에 적어도 부분적으로 기초하는, 방법. 비디오를 인코딩하는 방법으로서, 임계 값보다 더 큰 값을 갖는 명암도 정보와 관련된 코딩 유닛에 더 강한 디블록킹 필터링이 적용되는 방법. 비디오를 인코딩하는 방법으로서, 상기 임계 값은 소정의 값인 방법. 기술된 기술의 구현에는 하드웨어, 방법 또는 프로세스, 또는 컴퓨터-접근 가능 매체 상의 컴퓨터 소프트웨어가 포함될 수 있다.

[0009] 하나의 일반적인 양태는 비디오를 디코딩하는 방법을 포함할 수 있으며, 상기 방법은: 인코딩된 비디오의 비트스트림을 수신하는 단계; 상기 비트스트림을 디코딩하는 단계; 코딩 유닛을 결정하는 단계; 상기 코딩 유닛의 경계와 관련이 있는 픽셀의 명암도 정보를 결정하는 단계; 인코딩에 앞서, 상기 코딩 유닛과 관련이 있는 명암도 정보에 적어도 부분적으로 기초한 디블록킹 필터링을 상기 코딩 유닛에 적용하는 단계; 및 전송을 위해 상기 코딩 유닛을 인코딩하는 단계를 포함한다. 본 양태의 다른 구현에는 상응하는 컴퓨터 시스템, 장치, 및 하나 이상의 컴퓨터 저장 장치 상에 기록되어 각각 상기 방법의 동작을 수행하도록 구성된 컴퓨터 프로그램을 포함한다.

[0010] 구현에는 동일하거나 유사한 특징을 디코딩 프로세스에도 포함할 수 있다. 또한, 기술된 기술의 구현에는 하드웨어, 방법 또는 프로세스, 또는 컴퓨터-접근 가능 매체 상의 컴퓨터 소프트웨어가 포함될 수 있다.

도면의 간단한 설명

[0011] 본 발명의 추가적인 세부 내용은 첨부된 도면의 도움으로 설명되며, 여기서:

도 1은 프레임을 복수의 코딩 트리 유닛(CTU: Coding Tree Unit)으로 분할하는 것을 도시한다.

도 2a 내지 도 2c는 CTU를 코딩 유닛(CU)으로 예시적으로 분할하는 것을 도시한다.

도 3은 도 2의 CU 분할을 나타내는 쿼드트리+바이너리트리(QTBT)를 도시한다.

도 4는 JVET 또는 VVC 인코더에서의 CU 코딩에 대한 단순화된 블록도를 도시한다.

도 5는 VVC의 JVET에서의 루마 성분(luma component)에 대해 가능한 인트라 예측 모드를 도시한다.

도 6은 VVC 인코더의 JVET에서의 CU 코딩에 대한 단순화된 블록도를 도시한다.

도 7은 HDR 인코더/디코더 시스템의 블록도를 도시한다.

도 8은 정규화된 명암도에 대한 정규화된 PQ 곡선의 구현예를 도시한다.

도 9는 정규화된 명암도에 대한 JND 곡선의 구현예를 도시한다.

도 10은 명암도에 적어도 부분적으로 기초한 인코딩 시스템의 블록도의 구현예를 도시한다.

도 11은 명암도에 적어도 부분적으로 기초한 디코딩 시스템의 블록도의 구현예를 도시한다.

도 12a 내지 도 12c는 QP에 대한 일련의 예시적인 β 및 tc 곡선을 도시한 것으로서, 도 10 및 도 11에 기술되고 도시된 시스템을 그래픽으로 나타낸 것이다.

도 13은 템플릿 매칭을 위한 가변 템플릿 크기를 제공하도록 맞춰지고 구성된 컴퓨터 시스템의 구현예를 도시한다.

도 14는 템플릿 매칭을 위한 가변 템플릿 크기를 제공하도록 맞춰지고 구성된 비디오 인코더/디코더의 구현예를 도시한다.

발명을 실시하기 위한 구체적인 내용

[0012] 도 1은 프레임을 복수의 코딩 트리 유닛(CTU)(100)으로 분할하는 것을 도시한다. 프레임은 비디오 시퀀스 내의 이미지일 수 있다. 프레임은 이미지에서 강도 측정 값을 나타내는 픽셀 값을 갖는 매트릭스 또는 매트릭스 세트를 포함할 수 있다. 따라서, 이러한 매트릭스 세트는 비디오 시퀀스를 생성할 수 있다. 픽셀 값은 풀 컬러 비디오 코딩에서의 색상 및 명암을 나타내도록 정의될 수 있으며, 여기서 픽셀은 3개 채널로 분할된다. 예를 들어, YCbCr 색 공간에서, 픽셀은 그레이 레벨 강도를 나타내는 휘도 값 Y와, 색상이 그레이에서 블루와 레드로 달라지는 정도를 나타내는 2개의 색차(chrominance) 값 Cb와 Cr을 가질 수 있다. 다른 구현예에서, 픽셀 값은 상이한 색 공간 및 색 모델에서의 값으로 표현될 수 있다. 비디오의 해상도는 프레임 내 픽셀의 수를 결정할 수 있다. 해상도가 더 높다는 것은 픽셀이 더 많고 이미지의 선명도가 더 양호함을 의미할 수 있지만, 더 높은 대역폭, 저장 용량 및 전송 성능이 요구될 수도 있다.

[0013] 비디오 시퀀스의 프레임은 JVET를 사용해 인코딩 및 디코딩될 수 있다. JVET는 Joint Video Exploration Team에 의해 개발 중인 비디오 코딩 방식이다. JVET의 버전들은 JEM (Joint Exploration Model) 인코더와 디코더에 구현되었다. HVEC(High Efficiency Video Coding)과 같은 다른 비디오 코딩 방식과 마찬가지로, JVET는 블록 기반 하이브리드 공간 및 시간 예측 코딩 방식이다. JVET를 이용해 코딩하는 동안, 도 1에 도시된 바와 같이, 프레임은 먼저 CTU(100)로 불리는 정방형 블록으로 분할된다. 예를 들어, CTU(100)는 128x128 픽셀로 이루어진 블록일 수 있다.

[0014] 도 2는 CTU(100)를 CU(102)로 예시적으로 분할하는 것을 도시한다. 프레임 내의 각 CTU(100)는 하나 이상의 CU(102, 코딩 유닛)로 분할될 수 있다. CU(102)는 아래에 기술된 바와 같이 예측과 변환에 사용될 수 있다. HEVC와 달리, JVET에서, CU(102)는 직사각형 또는 정방형일 수 있고, 예측 유닛 또는 변환 유닛으로 추가로 분할되지 않고 코딩될 수 있다. CU(102)는 이들의 루트 CTU(100)만큼 크거나, 루트 CTU(100)의 더 작은 하위 분할 단위로서 작게는 4x4 블록일 수 있다.

[0015] JVET에서, CTU(100)는 쿼드트리+바이너리트리(QTBT) 방식에 따라 CU(102)로 분할될 수 있으며, 이 방식에서 CTU(100)는 쿼드트리에 따라 정방형 블록으로 재귀적으로 분할될 수 있고, 그런 다음, 이들 정방형 블록은 바이너리 트리에 따라 수평 또는 수직으로 재귀적으로 분할될 수 있다. CTU 크기, 쿼드트리 및 바이너리트리 리프 노드(binary tree leaf node), 바이너리트리 루트 노드의 최대 크기, 및 바이너리 트리의 최대 깊이와 같은 파라미터는 QTBT에 따라 분할을 조절하도록 설정될 수 있다. VVC에서도, CTU(100)는 터너리 분할(ternary splittin

g)을 활용해 CU로 분할될 수 있다.

- [0016] 비제한적 예로서, 도 2a는 CU(102)로 분할된 CTU(100)를 도시하며, 여기서 실선은 쿼드트리 분할을 나타내고 파선은 바이너리트리 분할을 나타낸다. 도시된 바와 같이, 바이너리 분할은 수평 분할 및 수직 분할을 통해 CTU의 구조를 정의할 수 있게 하고, 이의 CU로의 하위 분할을 가능하게 한다. 도 2b 및 2c는 CU의 터너리 분할의 대안적인 비제한적 예를 도시하며, 여기서 CU의 하위 분할은 동등하지 않다.
- [0017] 도 3은 도 2의 분할을 나타내는 QTBT를 도시한다. 쿼드트리 루트 노드는 CTU(100)을 나타내며, 여기서 쿼드트리 부분의 각 자식 노드(child node)는 부모 정방형 블록으로부터 분할된 4개의 정방형 블록 중 하나를 나타낸다. 그런 다음, 쿼드트리 리프 노드에 의해 나타나는 정방형 블록은 바이너리트리를 사용해 0회 또는 그 이상 분할될 수 있으며, 여기서 쿼드트리 리프 노드가 바이너리트리의 루트 노드이다. 바이너리 트리 부분의 각 레벨에서, 블록은 수직으로 또는 수평으로 분할될 수 있다. "0"으로 설정된 플래그는 블록이 수평으로 분할되었음을 나타내는 반면, "1"로 설정된 플래그는 블록이 수직으로 분할되었음을 나타낸다.
- [0018] 쿼드트리 분할 및 바이너리트리 분할 후, QTBT의 리프 노드에 의해 표시된 블록은, 예컨대 인터 예측 또는 인트라 예측을 사용하는 코딩 등에 의해 코딩될 최종 CU(102)를 나타낸다. 인터 예측으로 코딩된 슬라이스 또는 전체 프레임의 경우, 상이한 분할 구조가 루마 및 크로마 성분에 사용될 수 있다. 예를 들어, 인터 슬라이스의 경우, CU(102)는 상이한 색 성분을 위한 코딩 블록(CB), 예컨대 1개의 루마 CB 및 2개의 크로마 CB를 가질 수 있다. 인트라 예측으로 코딩된 슬라이스 또는 전체 프레임의 경우, 분할 구조는 루마 및 크로마 성분에 대해 동일할 수 있다.
- [0019] 도 4는 JVET 인코더에서의 CU 코딩에 대한 단순화된 블록도를 도시한다. 비디오 코딩의 주 단계는, 전술한 바와 같이 CU(102)를 식별하기 위해 분할하는 단계에 이어서, (404 또는 406)에서의 예측을 사용해 CU(102)를 인코딩하는 단계, 잔차 CU(410)을 (408에서) 생성하고, (412에서) 변환하고, (416에서) 양자화하고, (420에서) 엔트로피 코딩하는 단계를 포함한다. 도 4에 도시된 인코더 및 인코딩 과정은 아래에 보다 상세하게 기술되는 디코딩 과정도 포함한다.
- [0020] 현재 CU(102)가 주어지면, 인코더는 (404에서) 인트라 예측을 공간적으로 사용하거나 (406에서) 인터 예측을 시간적으로 사용해 예측 CU(402)를 획득할 수 있다. 예측 코딩의 기본 개념은 원래 신호와 원래 신호에 대한 예측 사이에서 차등 또는 잔차 신호를 전송하는 것이다. 수신기 측에서, 원래 신호는 후술하는 바와 같이 잔차와 예측을 추가함으로써 재구성될 수 있다. 차등 신호는 원래 신호보다 상관 관계가 더 낮기 때문에, 전송에 필요한 비트 수는 더 적다.
- [0021] 전체 화상이나 화상의 일부와 같은, 인트라 예측형 CU로 전체적으로 코딩된 슬라이스는 다른 슬라이스를 참조하지 않고 디코딩될 수 있는 I 슬라이스일 수 있으며, 따라서 디코딩이 시작될 수 있는 가능한 지점이 될 수 있다. 적어도 일부의 인터 예측형 CU로 코딩된 슬라이스는 하나 이상의 기준 화상을 기준으로 디코딩될 수 있는 예측형(P) 슬라이스 또는 이중예측형(B) 슬라이스일 수 있다. P 슬라이스는 이전에 코딩된 슬라이스와 함께 인트라 예측 및 인터 예측을 사용할 수 있다. 예를 들어, P 슬라이스는 인터 예측을 사용함으로써 I 슬라이스보다 더 압축될 수 있지만, 이를 코딩하기 위해서는 이전에 코딩된 슬라이스의 코딩이 필요하다. B 슬라이스는, 상이한 프레임으로부터의 보간 예측(interpolated prediction)을 사용하는 인터 예측 또는 인트라 예측을 사용해, 이의 코딩을 위한 이전 및/또는 후속 슬라이스의 데이터를 사용할 수 있으므로, 모션 추정 처리(motion estimation process)의 정확도를 증가시킬 수 있다. 일부 경우에, P 슬라이스와 B 슬라이스 또한 (또는 대안적으로) 인트라 블록 카피를 사용해 인코딩될 수 있으며, 여기서는 동일한 슬라이스의 다른 부분의 데이터가 사용된다.
- [0022] 후술하는 바와 같이, 인트라 예측 또는 인터 예측은 이전에 코딩된 CU(102), 예컨대 인접 CU(102) 또는 기준 화상의 CU(102)로부터 재구성된 CU(434)에 기초하여 수행될 수 있다.
- [0023] CU(102)가 (404에서) 인트라 예측을 이용해 공간적으로 코딩되는 경우, 화상에서 인접한 CU(102)의 샘플에 기초하여 CU(102)의 픽셀 값을 가장 잘 예측하는 인트라 예측 모드를 발견할 수 있다.
- [0024] CU의 루마 성분을 코딩할 때, 인코더는 후보 인트라 예측 모드들의 목록을 생성할 수 있다. HEVC는 루마 성분을 위한 35개의 가능한 인트라 예측 모드를 가진 반면, JVET에서는 루마 성분을 위한 67개의 가능한 인트라 예측 모드가 있으며 VVC에서는 85개의 예측 모드가 있다. 여기에는 인접 픽셀에서 생성된 값의 3차원 평면을 사용하는 플레인너 모드(planar mode), 인접 픽셀의 평균 값을 사용하는 DC 모드, 도 5에 도시되고, 실선 지시 방향을 따라 인접 픽셀에서 복사된 값을 사용하는 65개의 방향 모드, 및 비정방형 블록과 함께 사용될 수 있는 18개의

광각 예측 모드가 포함된다.

- [0025] CU의 루마 성분을 위한 후보 인트라 예측 모드의 목록을 생성할 때, 목록 상의 후보 모드의 수는 CU의 크기에 따라 달라질 수 있다. 후보 목록은: 절대 변환 차이 합(SATD: Sum of Absolute Transform Difference)의 비용이 가장 낮은 HEVC의 35개 모드의 하위 집합; HEVC 모드에서 발견된 후보와 인접하고, JVET를 위해 추가된 새로운 방향 모드; 기본 모드 목록뿐만 아니라 이전에 코딩된 인접 블록에 사용된 인트라 예측 모드에 기초하여 식별되는 CU(102)에 대한 6개의 가능성이 가장 높은 모드(MPM: most probable mode)로 이루어진 세트의 모드를 포함할 수 있다.
- [0026] CU의 크로마 성분을 코딩할 때, 후보 인트라 예측 모드들의 목록이 생성될 수도 있다. 후보 모드의 목록에는: 루마 샘플로부터 교차 성분 선형 모델 투영으로 생성된 모드; 크로마 블록 내의 특정 배열 위치에서 루마 CB에 대해 발견된 인트라 예측 모드; 및 인접 블록에 대해 이전에 발견된 크로마 예측 모드가 포함될 수 있다. 인코더는 목록 상에서 왜곡률 비용이 가장 낮은 후보 모드를 발견하여, CU의 루마 및 크로마 성분을 코딩할 때 이들 인트라 예측 모드를 사용할 수 있다. 신택스는 각각의 CU(102)를 코딩하는 데 사용되는 인트라 예측 모드를 나타내는 비트스트림으로 코딩될 수 있다.
- [0027] CU(102)에 대한 최선의 인트라 예측 모드를 선택한 후, 인코더는 이들 모드를 사용해 예측 CU(402)를 생성할 수 있다. 선택된 모드가 방향 모드인 경우, 4-탭 필터를 사용해 방향의 정확도를 개선할 수 있다. 예측 블록의 상단 또는 좌측에 있는 열 또는 행은 2-탭 또는 3-탭 필터와 같은 경계 예측 필터를 사용해 조정할 수 있다.
- [0028] 예측 CU(402)는 인접 블록의 필터링된 샘플을 기준으로 인접 블록의 필터링되지 않은 샘플을 사용해 생성한 예측 CU(402)를 조정하는 위치 의존적 인트라 예측 조합(PDPC) 프로세스, 또는 3-탭 또는 5-탭 저역 통과 필터를 사용해 참조 샘플을 가공하는 적응성 참조 샘플 평활화를 통해 더 평활화될 수 있다.
- [0029] CU(102)가 (406에서) 인터 예측을 사용해 시간적으로 코딩될 때, CU(102)의 픽셀 값을 가장 잘 예측하는 참조 화상에서 샘플들을 가리키는 한 세트의 모션 벡터(MV: motion vector)가 발견될 수 있다. 인터 예측은 슬라이스에서 픽셀 블록의 변위를 표시함으로써 슬라이스 간의 시간적 중복성(temporal redundancy)을 활용한다. 변위는 모션 보상(motion compensation)이라 불리는 프로세스를 통해 이전 또는 다음 슬라이스에서의 픽셀 값에 따라 결정된다. 특정 참조 화상에 대한 픽셀 변위를 나타내는 모션 벡터 및 연관된 참조 인덱스는, 원래 픽셀과 모션 보상된 픽셀 간의 잔차와 함께 비트스트림으로 디코더에 제공될 수 있다. 디코더는 잔차 및 전송된 모션 벡터 및 참조 인덱스를 사용해 재구성된 슬라이스에서 픽셀 블록을 재구성할 수 있다.
- [0030] JVET에서, 모션 벡터의 정확도는 1/16 픽셀로 저장될 수 있고, 모션 벡터와 CU의 예측 모션 벡터 간의 차이는 1/4 픽셀 해상도 또는 정수 픽셀 해상도로 코딩될 수 있다.
- [0031] JVET에서, 향상된 시간적 모션 벡터 예측(advanced temporal motion vector prediction, ATMVP), 공간-시간적 모션 벡터 예측(spatial-temporal motion vector prediction, STMVP), 아핀 모션 보상 예측, 패턴 정합 모션 벡터 유도(pattern matched motion vector derivation, PMMVD) 및/또는 양방향 광학 흐름(bi-directional optical flow, BIO)과 같은 기술을 사용해 CU(102) 내에서 다수의 하위 CU에 대한 모션 벡터를 찾을 수 있다.
- [0032] 인코더는 ATMVP를 사용하여 참조 화상에서 상응하는 블록을 가리키는 CU(102)에 대한 시간적 벡터를 찾을 수 있다. 시간적 벡터는 이전에 코딩된 인접 CU(102)에 대해 발견된 모션 벡터 및 참조 화상에 기초하여 발견될 수 있다. 전체 CU(102)에 대한 시간적 벡터가 가리키는 참조 블록을 사용하여 CU(102) 내의 각 하위 CU에 대한 모션 벡터를 찾을 수 있다.
- [0033] STMVP는 시간적 벡터와 함께 이전에 인터 예측으로 코딩된 인접 블록에 대해 발견된 모션 벡터를 스케일링하고 평균화함으로써 하위 CU에 대한 모션 벡터를 찾을 수 있다.
- [0034] 아핀 모션 보상 예측은, 블록의 상단 모서리에 대해 발견된 2개의 제어 모션 벡터에 기초하여 블록 내 각 하위 CU에 대한 모션 벡터 필드를 예측하는 데 사용할 수 있다. 예를 들어, 하위 CU에 대한 모션 벡터는 CU(102) 내의 각 4x4 블록에 대해 발견된 상단 모서리 모션 벡터에 기초하여 유도될 수 있다.
- [0035] PMMVD는 양방향 매칭 또는 템플릿 매칭을 사용해 현재 CU(102)에 대한 초기 모션 벡터를 찾을 수 있다. 양방향 매칭은 모션의 궤적을 따라 2개의 상이한 참조 화상에서 현재 CU(102)와 참조 블록을 볼 수 있는 반면, 템플릿 매칭은 현재 CU(102)에서 상응하는 블록과 템플릿에 의해 식별된 참조 화상을 볼 수 있다. 그런 다음, CU(102)에 대해 발견된 초기 모션 벡터는 각 하위 CU에 대해 개별적으로 조정(refined)될 수 있다.
- [0036] BIO는 인터 예측이 이전과 이후의 참조 화상에 기초한 이중 예측으로 수행될 때 사용될 수 있고, 2개의 참조 화

상 간의 차이의 구배에 기초하여 하위 CU에 대한 모션 벡터를 찾을 수 있게 한다.

- [0037] 일부 상황에서는, 로컬 조명 보상(LIC)을 CU 레벨에서 사용하여, 현재 CU(102)에 인접한 샘플 및 후보 모션 벡터에 의해 식별된 참조 블록에 인접한 상응하는 샘플에 기초하여 스케일링 인자 파라미터 및 오프셋 파라미터에 대한 값을 찾을 수 있다. JVET에서, LIC 파라미터는 CU 레벨로 변경되어 전송될 수 있다.
- [0038] 상기 방법 중 일부의 경우, CU의 하위 CU 각각에 대해 발견된 모션 벡터는 CU 레벨로 디코더에 전송될 수 있다. PMMVD 및 BIO와 같은 다른 방법의 경우, 오버헤드를 줄이기 위해 모션 정보는 비트스트림으로 전송되지 않으며, 디코더는 동일한 프로세스를 통해 모션 벡터를 유도할 수 있다.
- [0039] CU(102)에 대한 모션 벡터가 발견된 후, 인코더는 이들 모션 벡터를 사용해 예측 CU(402)를 생성할 수 있다. 일부 경우에, 개별 하위 CU에 대한 모션 벡터가 발견된 경우, 이들 모션 벡터를 하나 이상의 인접 하위 CU에 대해 이전에 발견된 모션 벡터와 결합함으로써 예측 CU(402)를 생성할 때 중첩 블록 모션 보상(Overlapped Block Motion Compensation, OBMC)이 사용될 수 있다.
- [0040] 양방향 예측이 사용되는 경우, JVET는 디코더 측의 모션 벡터 조정(motion vector refinement, DMVR)을 사용하여 모션 벡터를 찾을 수 있다. DMVR은, 양방향 템플릿 매칭 프로세스를 사용해 양방향 예측을 위해 찾은 2개의 모션 벡터에 기초하여 모션 벡터를 찾을 수 있게 한다. DMVR에서는, 2개의 모션 벡터 각각을 이용해 생성된 예측 CU(402)의 가중 조합을 찾을 수 있고, 2개의 모션 벡터는 조합된 예측 CU(402)를 가장 잘 가리키는 새로운 모션 벡터와 대체됨으로써 조정될 수 있다. 2개의 조정된 모션 벡터는 최종 예측 CU(402)를 생성하는 데 사용될 수 있다.
- [0041] (408)에서, 예측 CU(402)가 전술한 바와 같이 (404)에서의 인트라 예측 또는 (406)에서의 인터 예측을 통해 발견되면, 인코더는 현재 CU(102)로부터 예측 CU(402)를 차감하여 잔차 CU(410)를 얻을 수 있다.
- [0042] 인코더는, 이산 코사인 블록 변환(DCT-변환)을 사용해 데이터를 변환 도메인으로 변환하는 것과 같이, (412)에서 하나 이상의 변환 연산을 사용해 잔차 CU(410)를, 변환 도메인에서 잔차 CU(410)를 표현하는 변환 계수(414)로 변환할 수 있다. JVET는 HEVC보다 더 많은 유형의 변환 연산을 가능하게 하며, 이에는 DCT-II, DST-VII, DST-VII, DCT-VIII, DST-I 및 DCT-V 연산이 포함된다. 허용된 변환 연산은 하위 집합으로 그룹화될 수 있으며, 이들 하위 집합에 어떤 하위 집합과 어떤 특정 연산이 사용되었는지에 대한 표시가 인코더에 의해 전송될 수 있다. 일부 경우에, 특정 크기보다 큰 CU(102)에서의 고주파 변환 계수를 "0"으로 출력하기 위해 큰 블록 변환(large block-size transform)이 사용될 수 있는데, 그 결과 이들 CU (102)에 대해서는 저주파 변환 계수만이 유지된다.
- [0043] 일부 경우에, 순방향 코어 변환 후에 모드 의존성 비분리형 2차 변환(MDNSST)이 저주파 변환 계수(414)에 적용될 수 있다. MDNSST 연산은 회전 데이터에 기초한 하이퍼큐브-기븐스 변환(Hypercube-Givens Transform, HyGT)을 사용할 수 있다. 사용 시, 특정 MDNSST 연산을 식별하는 인덱스 값이 인코더에 의해 전송될 수 있다.
- [0044] (416)에서, 인코더는 변환 계수(414)를 양자화된 변환 계수(416)로 양자화할 수 있다. 각 계수의 양자화는 계수 값을 양자화 파라미터(QP)로부터 도출된 양자화 단계로 나눔으로써 계산될 수 있다. 일부 구현예에서, Qstep은 $2^{(QP-4)/6}$ 으로 정의된다. 고정밀 변환 계수(414)는 가능한 값의 수가 한정된 양자화된 변환 계수(416)로 변환될 수 있기 때문에, 양자화는 데이터 압축을 도울 수 있다. 따라서, 변환 계수의 양자화는 변환 프로세스에 의해 생성되고 전송되는 비트의 양을 제한할 수 있다. 그러나, 양자화는 손실이 많은 연산이고 양자화에 의한 손실은 복구될 수 없는 반면, 양자화 프로세스는 재구성된 시퀀스의 품질과 시퀀스를 나타내는 데 필요한 정보의 양 간의 트레이드-오프(trade-off)를 제공한다. 예를 들어, 표현 및 전송에 더 많은 양의 데이터가 필요할 수 있지만, QP 값이 낮을수록 디코딩된 비디오의 품질은 더 나아질 수 있다. 대조적으로, QP 값이 높으면 재구성된 비디오 시퀀스의 품질이 낮아질 수 있지만, 필요한 데이터와 밴드폭은 더 낮다.
- [0045] JVET는 분산 기반 적응 양자화 기술을 활용할 수 있는데, 이는 모든 CU(102)가 이들의 코딩 프로세스에 대해 (프레임의 모든 CU(102)의 코딩에서 동일한 프레임 QP를 사용하는 대신) 상이한 양자화 파라미터를 사용할 수 있게 한다. 분산 기반 적응 양자화 기술은 특정 블록의 양자화 파라미터를 적응적으로 낮추면서 다른 블록에서는 증가시킨다. CU(102)에 대한 특정 QP를 선택하기 위해, CU의 분산이 계산된다. 간단히 말해, CU의 분산이 프레임의 평균 분산보다 높은 경우, 프레임의 QP보다 높은 QP가 CU(102)에 대해 설정될 수 있다. CU(102)가 프레임의 평균 분산보다 더 낮은 분산을 나타내는 경우, 더 낮은 QP가 할당될 수 있다.
- [0046] (420)에서, 인코더는 양자화된 변환 계수(418)를 엔트로피 코딩함으로써 최종 압축 비트(422)를 얻을 수 있다.

엔트로피 코딩은 전송할 정보의 통계적 중복을 제거하는 것을 목표로 한다. JVET에서, 컨텍스트 적응 바이너리 산술 코딩(Context Adaptive Binary Arithmetic Coding, CABAC)이 양자화된 변환 계수(418)를 코딩하는데 사용될 수 있는데, 이에겐 통계적 중복을 제거하기 위한 확률 측정치가 사용된다. "0"이 아닌 양자화된 변환 계수(418)를 갖는 CU(102)의 경우, 양자화된 변환 계수(418)를 이진수(binary)로 변환할 수 있다. 그런 다음, 이진수로 표현된 각 비트("bin")를 컨텍스트 모델(context model)을 사용해 인코딩할 수 있다. CU(102)는 3개의 영역으로 나눌 수 있으며, 각 영역은 해당 영역 내의 픽셀에 사용하기 위한 고유한 컨텍스트 모델 세트를 갖는다.

[0047] bin을 인코딩하기 위한 다수의 스캔 패스를 수행할 수 있다. 첫 3개의 bin(bin0, bin1 및 bin2)을 인코딩하기 위한 패스 도중에, bin에 사용할 컨텍스트 모델을 나타내는 인덱스 값은 템플릿에 의해 식별된 최대 5개의 이전 에 코딩된 인접 양자화된 변환 계수(418)에서 해당 bin 위치의 합을 구함으로써 얻을 수 있다.

[0048] 컨텍스트 모델은 bin의 값이 "0" 또는 "1"이 될 확률에 기초할 수 있다. 값이 코딩될 때, 컨텍스트 모델에서의 확률은 발견된 "0" 및 "1" 값의 실제 수에 기초하여 갱신될 수 있다. HEVC는 고정된 테이블을 사용하여 각각의 새로운 화상에 대한 컨텍스트 모델을 재-초기화한 반면, JVET에서는, 이전에 코딩된 인터 예측 화상에 대해 개발된 컨텍스트 모델에 기초하여 새로운 인터-예측 화상에 대한 컨텍스트 모델을 초기화할 수 있다.

[0049] 인코더는 잔차 CU(410)의 엔트로피 인코딩된 비트(422), 선택된 인트라 예측 모드 또는 모션 벡터와 같은 예측 정보, QTBT 구조에 따라 CU(102)가 CTU(100)로부터 어떻게 분할되었는지에 대한 표시자, 및/또는 인코딩된 비디오에 대한 다른 정보를 포함하는 비트스트림을 생성할 수 있다. 비트스트림은 후술하는 바와 같은 디코더에 의해 디코딩될 수 있다.

[0050] 최종 압축 비트(422)를 찾기 위해 양자화된 변환 계수(418)를 사용하는 것에 추가하여, 인코더는 디코더가 재구성된 CU(434)를 생성하기 위해 사용하게 될 동일한 디코딩 프로세스 이후에 재구성된 CU(434)를 생성하기 위해 양자화된 변환 계수(418)를 사용할 수도 있다. 따라서, 변환 계수가 인코더에 의해 계산되고 양자화되면, 양자화된 변환 계수(418)는 인코더 내의 디코딩 루프로 전송될 수 있다. CU의 변환 계수가 양자화된 후, 디코딩 루프는 디코딩 프로세스에서 디코더가 생성하는 것과 동일한 재구성된 CU(434)가 인코더에 의해 생성될 수 있게 한다. 따라서, 인코더는 새로운 CU(102)에 대한 인트라 예측 또는 인터 예측을 수행할 때 디코더가 인접 CU(102) 또는 참조 화상에 사용하게 될 동일한 재구성된 CU(434)를 사용할 수 있다. 재구성된 CU(102), 재구성된 슬라이스, 또는 재구성된 완전한 프레임은 추가의 예측 단계에 대한 참조의 역할을 할 수 있다.

[0051] 재구성된 이미지에 대한 픽셀 값을 얻기 위한 인코더의 디코딩 루프에서(디코더에서의 동일한 작업에 대해서는 아래 참조), 역양자화 프로세스가 수행될 수 있다. 프레임을 역양자화하기 위해, 예를 들어, 프레임의 각 픽셀의 양자화된 값에 양자화 단계(예를 들어, 전술한 Qstep)를 곱하여 재구성된 역양자화된 변환 계수(426)를 얻는다. 예를 들어, 도 4에 도시된 인코더 내의 디코딩 프로세스에서, 잔차 CU(410)의 양자화된 변환 계수(418)를 (424)에서 역양자화하여 역양자화된 변환 계수(426)를 얻을 수 있다. 인코딩 중에 MDNSST 연산이 수행되는 경우, 연산은 역양자화 후에 되돌릴 수 있다.

[0052] (428)에서, 역양자화된 변환 계수(426)는, 예컨대 DCT를 값에 적용하여 재구성된 이미지를 획득함으로써 역변환되어 재구성된 잔차 CU(430)를 얻을 수 있다. (432)에서, 재구성된 CU(434)를 찾기 위해, 재구성된 잔차 CU(430)가 (404)에서의 인트라 예측 또는 (406)에서의 인터 예측을 통해 발견된 상응하는 예측 CU(402)에 추가될 수 있다.

[0053] (436)에서, 하나 이상의 필터가 (인코더에서, 또는 후술하는 바와 같이 디코더에서의) 디코딩 프로세스 중에 재구성된 데이터에 화상 레벨 또는 CU 레벨로 적용될 수 있다. 예를 들어, 인코더는 디블록킹 필터, 샘플 적응 오프셋(SAO) 필터 및/또는 적응 루프 필터(ALF)를 적용할 수 있다. 인코더의 디코딩 프로세스는 재구성된 이미지에서 잠재적인 아티팩트(artifact)를 처리할 수 있는 최적의 필터 파라미터를 추정하고 이를 디코더에 전송하는 필터를 구현할 수 있다. 이러한 개선은 재구성된 비디오의 객관적 및 주관적 품질을 향상시킨다. 디블록킹 필터링에서는, 하위 CU 경계 근처에 있는 픽셀은 수정될 수 있는 반면, SAO에서는, CTU(100) 내의 픽셀이 에지 오프셋 또는 밴드 오프셋 분류를 사용해 수정될 수 있다. JVET의 ALF는 각 2x2 블록에 대해 원형으로 대칭인 필터를 사용할 수 있다. 각 2x2 블록에 사용되는 필터의 크기 및 식별자에 대한 표시가 전송될 수 있다.

[0054] 재구성된 화상이 참조 화상인 경우, 이들은 미래 CU(102)에 대한 (406)에서의 인터 예측을 위해 참조 버퍼(438)에 저장될 수 있다.

[0055] 상기 단계에서, JVET는 콘텐츠 적응 클리핑 연산을 사용하여 색상 값을 하향 및 상향 클리핑 경계 사이에 맞도록 조정할 수 있게 한다. 클리핑 경계는 각각의 슬라이스에 대해 변경될 수 있고, 경계를 식별하는 파라미터는

비트스트림으로 전송될 수 있다.

- [0056] 도 6은 JVET에서의 CU 코딩에 대한 단순화된 블록도를 도시한다. JVET 디코더는 인코딩된 CU(102)에 관한 정보가 담긴 비트스트림을 수신할 수 있다. 비트스트림은 QTBT 구조에 따라 화상의 CU(102)가 어떻게 CTU(100)로부터 분할되었는지, 인트라 예측 모드 또는 모션 벡터와 같은 CU(102)에 대한 예측 정보, 및 엔트로피 인코딩된 잔차 CU를 나타내는 비트(602)를 나타낼 수 있다.
- [0057] (604)에서, 디코더는 인코더에 의해 비트스트림으로 전송된 CABAC 컨텍스트 모델을 사용해 엔트로피 인코딩된 비트(602)를 디코딩할 수 있다. 디코더는 인코더에 의해 전송된 파라미터를 사용해 컨텍스트 모델의 확률을 이들이 인코딩 중에 갱신된 것과 동일한 방식으로 갱신할 수 있다.
- [0058] 양자화된 변환 계수(606)를 구하기 위해 (604)에서의 엔트로피 인코딩을 되돌린 다음, 디코더는 (608)에서 이들을 역양자화하여 역양자화된 변환 계수(610)를 구할 수 있다. 인코딩 중에 MDNSST 연산이 수행되는 경우, 연산은 역양자화 후에 디코더에 의해 되돌릴 수 있다.
- [0059] (612)에서, 역양자화된 변환 계수(610)는 재구성된 잔차 CU(614)를 구하기 위해 역변환될 수 있다. (616)에서, 재구성된 CU(618)를 구하기 위해, 재구성된 잔차 CU(614)가 (622에서의) 인트라 예측 또는 (624에서의) 인터 예측을 통해 발견된 상응하는 예측 CU(626)에 추가될 수 있다.
- [0060] (620)에서, 하나 이상의 필터가 재구성된 데이터에 화상 레벨 또는 CU 레벨로 적용될 수 있다. 예를 들어, 디코더는 더블록킹 필터, 샘플 적응 오프셋(SAO) 필터 및/또는 적응 루프 필터(ALF)를 적용할 수 있다. 전술한 바와 같이, 인코더의 디코딩 루프 내에 위치한 인-루프 필터는 프레임의 객관적 및 주관적 품질을 향상시키기 위해 최적의 필터 파라미터를 추정하는 데 사용될 수 있다. 이들 파라미터는 디코더로 전송되어, 620에서 재구성된 프레임을 필터링하여 인코더에서 필터링된 재구성된 프레임과 일치시킨다.
- [0061] 재구성된 CU(618)를 구해 전송된 필터를 적용함으로써 재구성된 화상이 생성된 후, 디코더는 재구성된 화상을 출력 비디오(628)로서 출력할 수 있다. 재구성된 화상이 참조 화상으로 사용될 경우, 이들은 미래 CU(102)에 대한 (624)에서의 인터 예측을 위해 참조 버퍼(630)에 저장될 수 있다.
- [0062] 도 7은 HDR 인코딩(702) 및 디코딩(704)의 블록도(700)를 도시한다. 하나의 공통 HDR 비디오 포맷은 선형 광 RGB 영역을 사용하며, 여기서 각각의 채널은 고 비트 심도 포맷으로 (비제한적인 예로서, EXR 파일 포맷을 위한 반 부동 포맷(half-float format)으로 지정된다. 현재의 비디오 압축 알고리즘은 HDR 비디오 형식을 직접적으로 처리할 수 없기 때문에, HDR 비디오를 인코딩하는 하나의 접근법은 먼저 이를 비디오 인코더가 받아들일 수 있는 포맷으로 변환하는 것이다. 그런 다음 디코더 비디오가 HDR 포맷으로 다시 전환될 수 있다. 이러한 시스템의 예는 도 7에 도시되어 있으며, 여기서 인코딩 모듈(702) 및 디코딩 모듈(704)이 SDR 콘텐츠의 JVET 코딩에 대해 본원에 기술된 프로세스에 상응한다.
- [0063] 도 7의 상단에 도시된 시스템은 입력된 HDR 비디오 포맷을 JVET 인코더(또는 메인 10 HEVC 인코더 등)를 사용해 인코딩할 수 있는 10비트 4:2:0 비디오 포맷으로 전환하는 것의 예를 보여준다. 고 비트 심도 입력을 저 비트 심도로 전환하기 위해서는, 우선 입력 HDR 비디오의 각 RGB 채널을 코딩 전달 함수(TF)(706)를 통해 전달한다. 그런 다음, 출력 R'G'B'을 비디오 코딩에 더 적합한 색 공간 Y'CbCr(708)로 변환시킨다. 그런 다음, 단계(710)에서 지각 맵핑(perceptual mapping)을 수행하고, 이어서 단계(712)에서 각 채널을 10비트로 양자화한다. 단계(712)에서 각 채널을 10비트로 균일하게 양자화한 후, 단계(714)에서 크로마(Cb 및 Cr) 채널을 4:2:0 포맷으로 서브샘플링한다. 그런 다음, 단계(716)에서 인코더가 예를 들어 메인 10 HEVC 인코더를 사용해 10비트 4:2:0 비디오를 압축한다.
- [0064] 도 7의 하단에 도시된 시스템은 입력 비트스트림을 출력 HDR 비디오로 재구성하는 것을 보여준다. 일 예에서, 단계(817)에서 비트스트림을 디코딩하고, 단계(720)에서 JVET 디코더(또는 메인 10 HEVC 디코더 또는 알려진 다른 편리한 디코더 및/또는 바람직한 디코더)가 10비트 4:2:0 비디오를 재구성하고, 재구성된 비디오를 4:4:4 포맷으로 업샘플링(upsampling)한다. 단계(722)에서 10비트 데이터를 역 양자화 리매핑(remapping)한 후, 단계(724)에서 역 지각 맵핑을 적용해 Y'CbCr 값을 생성한다. 그런 다음, 단계(726)에서 Y'CbCr 데이터를 R'G'B' 색 공간으로 변환할 수 있고, 단계(728)에서 채널을 대상으로 역방향 코딩 TF 작업을 수행한 후 HDR 비디오 데이터가 출력될 수 있다.
- [0065] 블로킹 아티팩트는 대부분 블록 기반 비디오 코딩에서 이웃 유닛의 독립적 코딩에 의해 발생한다. 블로킹 아티팩트는 이웃 블록의 인트라/인터 코딩 유형이 상이한 경우 및 공간 활동이 낮은 영역에서 발생하고 낮은 비트 레이트에서 가시적으로 보이는 경향이 있다. 그 결과 인위적인 불연속성 또는 도입된 경계에 의해 시각적 아티

팩트가 발생한다.

[0066] HEVC [1] 및 현재 JVET에서의 필터와 같은 디블록킹 필터는 PU/TU 또는 CU 경계에 대한 평활화(smoothing) 또는 저역 통과 필터링(low pass filtering)을 통해 시각적 아티팩트를 줄이고자 한다. 일부 구현예에서, 수직 경계가 먼저 필터링되고, 이어서 수평 경계가 필터링된다. 경계의 어느 한쪽에 있는 4x4 영역에서 최대 4개의 재구성된 루마 픽셀 값을 사용해 경계의 어느 한쪽에 있는 최대 3개의 픽셀을 필터링할 수 있다. 정상 필터링이나 약한 필터링의 경우, 어느 한쪽에 있는 최대 2개의 픽셀을 필터링할 수 있고, 강한 필터링의 경우, 어느 한쪽에 있는 최대 3개의 픽셀을 필터링할 수 있다. 픽셀을 필터링할 것인지의 여부는 0, 1 또는 2의 경계 강도 값 Bs를 생성하기 위한 이웃 블록의 인트라/인터 모드 결정, 모션 정보, 및 잔차 정보에 따라 달라질 수 있다. Bs > 0인 경우 수직(또는 수평) 경계의 어느 한쪽에 있는 4x4 영역에서 첫 번째 및 마지막 행(또는 열)에 대한 평활도 조건을 확인할 수 있다. 이들 조건을 통해 주어진 경계에 걸쳐 램프로 인해 얼마나 많은 편차가 존재하는지를 결정할 수 있다. 일반적으로, 편차가 파라미터 β 에 의해 지정된 임계 값보다 작은 경우, 디블록킹 필터링을 전체 4x4 영역에 적용할 수 있지만, 큰 편차는 실제 경계 또는 의도된 경계를 나타내는 것일 수 있으므로, 디블록킹 필터링을 수행하지 않을 수 있다. 베타 파라미터는 블록 QP 값의 비감소 함수(non-decreasing function)이므로, QP 값이 커질수록 대응하는 임계 값이 커진다. 일부 구현예에서, Bs > 0이고 평활도 조건이 충족되는 경우, 강한 필터링과 약한 필터링 간의 결정은 QP의 비감소 함수이기도 한 또 다른 파라미터 tc와 추가의 평활도 조건에 기초하여 이루어질 수 있다. 일반적으로, 더 평활한 영역에 강한 필터링이 적용되는데, 이는 이러한 영역에서는 불연속성(discontinuities)이 시각적으로 더 분명해지기 때문이다.

[0067] 일부 구현예에서, 디블록킹 필터 연산은 효과적으로 4 또는 5-탭 필터링 연산이지만, 이 경우, 입력과 필터링된 출력 간의 차이를 먼저 클리핑(clipping)한 다음 입력에 다시 추가(또는 입력으로부터 차감)한다. 클리핑은 과도한 평활화(oversmoothing)를 제한하고자 하는 것이며, 클리핑 수준은 tc 및 QP에 의해 결정할 수 있다. 크로마 디블록킹의 경우, 블록 중 적어도 하나가 인트라 코딩될 때, 경계의 어느 한쪽에 있는 하나의 픽셀에 4-탭 필터를 적용할 수 있다.

[0068] 디블록킹 아티팩트는 블록 경계(예: CU 경계, 예측 경계, 변환 변화 및/또는 다른 분할 경계)에서의 불일치로 인해 발생할 수 있다. 이들 차이는 DC 레벨, 정렬, 위상 및/또는 다른 데이터에 존재할 수 있다. 이와 같이, 경계 차이는 신호에 부가되는 노이즈로 간주될 수 있다. 도 7에 도시된 바와 같이, 원래의 입력 HDR 신호가 코딩 TF 및 역 코딩 TF 둘 다를 거치지만, 디블록킹 노이즈는 역 코딩 TF만을 거친다. 전통적인 SDR 디블록킹 아티팩트는 이러한 추가의 TF를 고려하지 않았으며, 도 7에서의 디코더의 출력이 보여지는 곳에서 발생해 왔다. HDR의 경우, 디블록킹 노이즈가 역 코딩 TF를 거치므로 아티팩트의 가시성을 변경시킬 수 있다. 이와 같이, 동일한 불연속점 도약(discontinuity jump)이 밝은 영역과 어두운 영역 모두에 있는 경우, 역 코딩 TF 연산 후에 불연속점 도약은 더 커지거나 더 작아질 수 있다.

[0069] PQ, HLG 및 감마(Gamma)와 같은 전형적인 역 코딩 TF(종종 EOTF로 알려짐)는 명암도의 단조 증가 함수라는 특징으로 가지며, 도 8은 명암도(I)(802)에 대해 정규화된 PQ EOTF 곡선(800)을 도시한다. 예를 들어, 정규화된 PQ EOTF 곡선(800)이 도 8에 도시되어 있다. PQ EOTF 곡선(800)의 기울기가 증가하고 있기 때문에, 불연속점 도약은 더 어두운 영역 대비 더 밝은 영역에서 EOTF에 의해 확대되어, 디블록킹 아티팩트를 잠재적으로 보다 더 가시화시키게 된다. 웨버의 법칙(Weber's Law)에 따르면, JND(최소 식별차(just-noticeable difference))가 클수록 시청자는 더 밝은 영역에서 더 큰 차이를 용인할 수 있는 것으로 이해된다. 그러나, 명암도(802)에 대해 도표화된 JND(902)의 정규화 곡선(900)이 도시된 도 9는, 웨버의 법칙을 감안하더라도, PQ EOTF에 대한 명암도가 높을 때 JND가 감소한다는 것을 보여준다. 도 9는 $\alpha = 8\%$ 웨버의 법칙 JND 임계 값을 기준으로 계산된 것이며, 여기서는 피크 JND가 PQ에 대한 광범위한 임계 값에 매우 민감하지는 않은 것으로 보인다. 실제로, PQ에 대한 피크 JND는 도 8에서 PQ EOTF 상의 단위 기울기(unity slope) 근방에서 발생하는 것으로 보이며, 이는 대략 $I = 78\%$ (정규화된) 피크 명암도에서 발생한다. 대안적인 시험에서 나타난 바에 따르면, HLG EOTF의 경우, 피크 JND 명암도는 대략 $I = 50\%$ (정규화된)에서 발생하는 반면, 단위 기울기는 약 70% (정규화된)에서 발생하는 것으로 보인다.

[0070] 이러한 분석 및 관련 시각적 관찰에 기초하여, 명암도에 따라 달라지는 디블록킹 필터 연산을 통해 개선된 성능을 얻게 된다는 것이 분명해진다. 즉, 비제한적인 예로서, 디블록킹 필터 계수, 적용된 필터링 강도(정상 필터링 대 약한 필터링), 사용되었거나 영향을 받는 입력 및 출력 픽셀의 수, 필터링 크기/크기에 대한 결정, 및 기타 필터링 기준이 명암도에 영향을 받을 수 있으므로 명암도에 따라 달라질 수 있다. 명암도는 루마 및/또는 크로마에 대한 것일 수 있고, 이는 비선형 또는 선형 명암도에 따라 달라질 수 있다. 일부 구현예에서, 명암도는 국부적인 명암도에 기초하여, 예컨대 CU 명암도에 기초하여 연산되거나 블록 경계 주위에 있는 이웃 픽셀들에

기초하여 연산될 수 있다. 일부 구현예에서, 명암도는 이웃 픽셀 루마/크로마 값에 기초한 최대, 최소, 평균, 또는 일부 다른 통계치 또는 메트릭(metric)일 수 있다. 대안적인 구현예에서, 디블록킹 필터링은 장면, 시퀀스, 또는 다른 인터 또는 인트라 단위 값 별 프레임 또는 프레임 군의 명암도에 기초할 수 있다.

[0071] 일부 구현예에서, 디블록킹 연산은 인코더 및/또는 디코더에서 계산된 명암도 연산에 기초하여 결정될 수 있거나, 디블록킹 결정이나 필터링 연산을 수행하는 데 사용될 디코더에 파라미터(들)가 비트스트림을 통해 전송될 수 있다. 파라미터는 CU, 슬라이스, 사진, PPS, SPS 레벨로 전송될 수 있고/있거나 임의의 기타 알려진 통상적인 및/또는 바람직한 레벨로 전송될 수 있다.

[0072] 명암도-기반 디블록킹은 SDR 콘텐츠에도 적용될 수 있지만, 명암도-기반 디블록킹은 HDR에 적용된 역 코딩 TF로 인해 HDR 콘텐츠에 더 큰 영향을 미칠 것으로 예상된다. 일부 구현예에서, 디블록킹은 역 코딩 TF(또는 코딩 TF)에 기초할 수 있다. TF 정보는 비트스트림으로 신호화되어 디블록킹 연산에 사용될 수 있다. 비제한적인 예로서, (국부적인 또는 전체적인) 명암도가 일부 임계치보다 큰지 또는 작은지 여부에 기초하여 상이한 디블록킹 전략이 사용될 수 있고, 이러한 임계치는 TF에 기초할 수 있다. 또한, 일부 구현예에서, 둘 이상의 임계치가 식별되어 다중 레벨의 필터링 연산과 결합될 수 있다. 일부 구현예에서, 예시적인 디블록킹 전략은 필터링의 적용 여부, 강한 필터링 대 약한 필터링, 및/또는 다양한 명암도 레벨 유발 값(intensity level trigger values)에 기초한 다양한 레벨의 필터링을 포함할 수 있다. 일부 구현예에서, 디블록킹 필터링이 역 코딩 TF 이후에 필요하지 않음을 결정될 수 있는데, 이는 아티팩트가 덜 가시적일 수 있고 (또는 더 이상 가시적이지 않을 수 있고), 따라서 연산 필요성이 줄어들기 때문이다. I*에 대한 값(정규화된 명암도 값)은 TF에 기초하여 신호화되거나, 연산되거나 지정되어 필터링을 결정할 때 임계치로서 사용될 수 있다. 일부 구현예에서, 디블록킹 필터 연산을 변경하기 위해 둘 이상의 임계치를 사용할 수 있다.

[0073] HDR을 위한 명암도-기반 디블록킹을 통합하기 위해 HVEC 또는 JVET에서 기존 SDR 디블록킹에 대한 변경이 이루어질 수 있다. 비제한적인 예로서, HEVC의 경우, 명암도에 기초하여 디블록킹 파라미터 β (및 t_c)를 변경하여 강한/보통 필터링을 증가/감소시키거나 필터링을 켜고/끄 수 있으며, 명암도 값 또는 명암도 값의 범위에 기초하여 HDR을 위한 상이한 β (및 t_c) 파라미터 곡선을 정의할 수 있다. 대안적으로, 경계, CU, 영역 또는 프레임 군의 이웃에서의 명암도에 기초하여 시프트(shift) 또는 오프셋(offset)이 파라미터 및 곡선에 적용될 수 있다. 비제한적인 예로서, 더 밝은 영역에 더 강한 필터링이 적용되도록 시프트가 적용될 수 있다.

[0074] 도 10은 필터링을 결정할 목적으로 명암도가 고려된 인코딩 시스템(1000)의 블록도를 도시한다. 단계(1002)에서, 코딩 유닛 및 인접/이웃 코딩 유닛에 관한 정보를 얻을 수 있다. 그런 다음, 단계(1004)에서 필터링의 적용 여부에 관한 결정이 이루어질 수 있다. 단계(1004)에서, 필터링을 적용하기로 결정하는 경우, 단계(1006)에서 코딩 유닛 및/또는 인접한/이웃하는 코딩 유닛(들)과 관련된 명암도 값을 평가할 수 있다. 단계(1006)에서의 명암도 값의 평가에 기초하여, 단계(1008a 내지 1008c) 중 하나에서 원하는 레벨의 필터링을 코딩 유닛에 적용할 수 있다. 일부 구현예에서, 필터링의 레벨을 선택하는 것은 코딩 유닛의 명암도 값 및/또는 코딩 유닛과 관련된 명암도 값과 하나 이상의 코딩 유닛과 관련된 명암도 값을 비교하는 것에 기초할 수 있다. 일부 구현예에서, 이는 하나 이상의 확립된 임계 명암도 값에 기초할 수 있다. 단계(1008a 내지 1008c) 중 하나에서 필터링을 적용한 후, 단계(1010)에서 전송을 위해 코딩 유닛을 인코딩할 수 있다. 그러나, 필터링이 적용되어서는 안된다고 단계(1004)에서 결정하는 경우, 필터링되지 않은 코딩 유닛이 단계(1010)에서의 인코딩으로 직접 진행될 수 있도록 단계(1006 내지 1008c)를 우회할 수 있다.

[0075] 대안적인 구현예에서, 단계(1006)는 단계(1004)에 선행할 수 있고, 명암도의 평가가 단계(1006)에서 필터링을 결정하는 데 사용될 수 있으며, 단계(1004) 다음에 단계(1010)에서의 인코딩이 이어지거나 (필터링이 바람직하지 않은 경우), 단계(1008a 내지 1008c) 중 하나가 이어질 수 있다(필터링이 바람직한 경우).

[0076] 도 11은 명암도가 디스플레이를 위한 필터링에서 고려되는 인자인 디코딩 시스템의 블록도를 도시한다. 도 11에 도시된 구현예에서, 단계(1102)에서 비트스트림을 수신하여 디코딩할 수 있다. 일부 구현예에서, 단계(1104)에서 적절한 및/또는 바람직한 디블록킹 레벨을 결정할 수 있다. 그러나, 일부 대안적인 구현예에서는, 단계(1104)에서 인코딩 단계 동안에 인코딩을 적용하였는지 여부를 결정할 수 있다. 단계(1104)에서, 필터링이 바람직한 것으로 결정하는 경우(또는 일부 구현예에서 암호화 단계 동안에 인코딩을 적용한 경우), 단계(1106)에서 필터링 레벨을 결정한다. 일부 구현예에서, 이는 필터링과 관련된 하나 이상의 인자를 확립하는데 사용하기 위한 오프셋 값일 수 있고/있거나, 인코딩 단계 동안 적용된 필터링 레벨의 표시자일 수 있다. 단계(1106)에서의 결정에 적어도 부분적으로 기초하여, 단계(1110)에서의 디스플레이를 위한 이미지의 렌더링을 위한 필터링 레벨이 단계(1108a 내지 1108c)에서 적용될 것이다. 인코딩 단계 동안 필터링이 적용되지 않은 경우, 단계(1110)에

서의 디스플레이를 위한 이미지가 단계(1104)에서 렌더링될 수 있다.

- [0077] 도 12a 내지 도 12c는 QP에 대한 일련의 예시적인 β 및 tc 곡선을 도시한 것(1200)으로서, 도 10 및 도 11에 기술되고 도시된 시스템을 그래픽으로 나타낸 것이다. 도 12a에 도시된 구현예에서, 명암도가 바람직한 x의 임계값 미만인 경우(1206)에 사용할 수 있는, QP에 대한 예시적인 β 및 tc 곡선의 쌍(1202 및 1204)이 제시되어 있다. 따라서, 명암도 값이 바람직한 x 값 미만인 경우(1202), β 및 tc의 정상 값 또는 표준 값을 사용해 적용할 디블록킹 레벨을 결정할 수 있다. 도 12b 및 12c는,
- [0078] 명암도가 바람직한 x 값 이상인 것(1208)으로 결정된 경우에 사용할 수 있는, QP에 대한 대안적인 β 및 tc 곡선(1212, 1214, 1222, 1224)을 도시한다. 도 12b는 도 12a에 도시되었지만 좌측으로 시프트된 동일한 세트의 곡선(1212 및 1214)을 나타내고, 도 12c는 도 12a에 도시되었지만 상방으로 시프트된 동일한 세트의 곡선(1222 및 1224)을 나타낸다. 따라서, 명암도 값이 바람직한 x 값 이상인 (또는 초과인) 경우, β 및 tc의 오프셋 값, 비-표준 값 또는 변경된 값을 사용해 적용할 디블록킹 레벨을 결정할 수 있다. 따라서, 명암도 값이 증가할 때, β 및 tc의 증가된 값이 선택될 것이고 적용된 필터링 레벨은 증가하게 된다. 도 12b 내지 도 12c는 명암도(I)가 x의 단일 값을 초과하거나 이상인 변형예를 도시하고 있지만, 시스템은 다양한 경계와 각각 관련된 QP에 대한 β 및 tc 곡선의 다수의 세트가 존재하는 시스템을 포함하도록 확장될 수 있음을 이해해야 한다. 즉, $I < x$, $x \leq I \leq y$, 및 $I > y$ 과 같은 조건이 존재하는 조건들 및/또는 다수의 경계 또는 영역을 사용하는 시스템이 고려된다. 또한, $<$, $>$, \leq 및 \geq 의 사용은 임의적인 것이며, 임의의 논리적 경계 조건이 사용될 수 있음에 특히 유의해야 한다. 마지막으로, 도 12a 내지 도 12c에 제시된 곡선은 본질적으로 예시적인 것이며, 동일하거나 유사한 기술, 방법 및 로직(logic)이 임의의 공지된 편의적인 및/또는 바람직한 곡선의 세트에 적용될 수 있다는 점을 잘 이해해야 한다.
- [0079] 구현예를 실시하는 데 필요한 명령 시퀀스의 실행은 도 13에 도시된 바와 같은 컴퓨터 시스템(1300)에 의해 수행될 수 있다. 일 구현예에서, 명령 시퀀스의 실행은 하나의 컴퓨터 시스템(1300)에 의해 수행된다. 다른 구현예에 따르면, 통신 링크(1315)에 의해 결합된 둘 이상의 컴퓨터 시스템(1300)이 서로 협력하여 명령 시퀀스를 수행할 수 있다. 단 하나의 컴퓨터 시스템(1300)에 대한 설명이 아래에 제시되겠지만, 임의의 수의 컴퓨터 시스템(1300)이 구현예를 실시하는 데 사용될 수 있는 것으로 이해해야 한다.
- [0080] 이제부터, 컴퓨터 시스템(1300)의 기능적 구성 요소들의 블록도인 도 13을 참조하여, 일 구현예에 따른 컴퓨터 시스템(1300)이 기술될 것이다. 본원에서 사용되는 바와 같이, 컴퓨터 시스템(1300)이란 용어는 하나 이상의 프로그램을 저장하고 독립적으로 실행할 수 있는 임의의 컴퓨팅 장치를 기술하도록 광범위하게 사용된다.
- [0081] 각각의 컴퓨터 시스템(1300)은 버스(1306)에 결합된 통신 인터페이스(1314)를 포함할 수 있다. 통신 인터페이스(1314)는 컴퓨터 시스템들(1300) 간의 양방향 통신을 제공한다. 각 컴퓨터 시스템(1300)의 통신 인터페이스(1314)는 전기적, 전자기적 또는 광학적 신호를 송신하고 수신하며, 이에는 다양한 유형의 신호 정보, 예를 들어, 명령, 메시지 및 데이터를 나타내는 데이터 스트림이 포함된다. 통신 링크(1315)는 하나의 컴퓨터 시스템(1300)을 또 다른 컴퓨터 시스템(1300)과 연결시킨다. 예를 들어, 통신 링크(1315)는 LAN이거나(이 경우의 통신 인터페이스(1314)는 LAN 카드일 수 있음), 통신 링크(1315)는 PSTN이거나(이 경우의 통신 인터페이스(1314)는 종합 정보 통신망(integrated services digital network, ISDN) 카드 또는 모뎀일 수 있음), 통신 링크(1315)는 인터넷일 수 있다(이 경우의 통신 인터페이스(1314)는 다이얼-업 모뎀, 케이블 모뎀 또는 무선 모뎀일 수 있음).
- [0082] 컴퓨터 시스템(1300)은 프로그램(애플리케이션, 코드)을 포함하여 메시지, 데이터 및 명령을 각각의 통신 링크(1315) 및 통신 인터페이스(1314)를 통해 송신하고 수신할 수 있다. 수신된 프로그램 코드는 수신되는 즉시 각각의 프로세서(들)(1307)에 의해 실행되고/되거나 나중에 실행하기 위해 저장 장치(1310) 또는 다른 연관 비휘발성 매체에 저장될 수 있다.
- [0083] 구현예에서, 컴퓨터 시스템(1300)은 데이터 저장 시스템(1331), 예를 들어, 컴퓨터 시스템(1300)에 의해 쉽게 접근할 수 있는 데이터베이스(1332)를 포함하는 데이터 저장 시스템(1331)과 함께 작동한다. 컴퓨터 시스템(1300)은 데이터 인터페이스(1333)를 통해 데이터 저장 시스템(1331)과 통신한다. 버스(1306)에 결합된 데이터 인터페이스(1333)는 전기적, 전자기적 또는 광학적 신호를 송신하고 수신하며, 이에에는 다양한 유형의 신호 정보, 예를 들어, 명령, 메시지 및 데이터를 나타내는 데이터 스트림이 포함된다. 구현예에서, 데이터 인터페이스(1333)의 기능은 통신 인터페이스(1314)에 의해 수행될 수 있다.
- [0084] 컴퓨터 시스템(1300)은 정보로 통칭되는 명령, 메시지 및 데이터 통신을 위한 버스(1306) 또는 다른 통신 매커

니즘; 및 버스(1306)에 결합되고 정보를 처리하기 위한 하나 이상의 프로세서(1307)를 포함한다. 컴퓨터 시스템(1300)은 또한, 버스(1306)에 결합되고 동적 데이터 및 프로세서(들)(1307)에 의해 실행될 명령을 저장하기 위한 주 메모리(1308), 예컨대 랜덤 액세스 메모리(RAM) 또는 기타 동적 저장 장치를 포함한다. 주 메모리(1308)는 프로세서(들)(1307)가 명령을 수행하는 동안 임시 데이터, 즉 변수 또는 다른 중간 정보를 저장하는 데 사용될 수도 있다.

[0085] 컴퓨터 시스템(1300)은, 버스(1306)에 결합되어 정적 데이터 및 프로세서(들)(1307)를 위한 명령을 저장하기 위한 읽기 전용 메모리(ROM)(1309) 또는 기타 정적 저장 장치를 추가로 포함할 수 있다. 또한, 자기 디스크 또는 광학 디스크와 같은 저장 장치(1310)가 제공되어, 데이터 및 프로세서(들)(1307)를 위한 명령을 저장하기 위해 버스(1306)에 결합될 수 있다.

[0086] 컴퓨터 시스템(1300)은 사용자에게 정보를 디스플레이하기 위해 버스(1306)를 통해 음극선관(CRT) 또는 액정 디스플레이(LCD) 모니터와 같은 그러나 이들로 한정되지 않는 디스플레이 장치(1311)에 결합될 수 있다. 영숫자 및 기타 키와 같은 입력 장치(1312)가 정보 및 명령 선택을 프로세서(들)(1307)에 전달하기 위해 버스(1306)에 결합된다.

[0087] 일 구현예에 따르면, 개별 컴퓨터 시스템(1300)은 주 메모리(1308)에 포함된 하나 이상의 명령의 하나 이상의 시퀀스를 실행하는 이들 각각의 프로세서(들)(1307)에 의해 특정 동작을 수행한다. 이러한 명령은 ROM(1309) 또는 저장 장치(1310)와 같은 다른 컴퓨터-가용 매체로부터 주 메모리(1308)로 판독될 수 있다. 주 메모리(1308)에 포함된 명령 시퀀스의 실행은 프로세서(들)(1307)로 하여금 본원에 기술된 프로세스를 수행하게 한다. 대안적인 구현예에서, 하드-와이어드 회로(hard-wired circuitry)가 소프트웨어 명령 대신에 또는 이와 조합하여 사용될 수 있다. 따라서, 구현예는 하드웨어 회로 및/또는 소프트웨어의 임의의 특정 조합으로 한정되지는 않는다.

[0088] 본원에서 사용된 바와 같이, 용어 "컴퓨터 가용 매체"는 정보를 제공하거나 프로세서(들)(1307)에 의해 사용 가능한 임의의 매체를 지칭한다. 이러한 매체는 많은 형태를 취할 수 있으며, 이에 비휘발성, 휘발성 및 전송 매체를 포함하지만 이에 한정되지는 않는다. 비휘발성 매체, 즉 전원이 없을 때 정보를 유지할 수 있는 매체는 ROM(1309), CD ROM, 자기 테이프 및 자기 디스크를 포함한다. 휘발성 매체, 즉 전원이 없을 때 정보를 유지할 수 없는 매체는 주 메모리(1308)를 포함한다. 전송 매체는 버스(1306)를 구성하는 와이어를 포함하여 동축 케이블, 구리 와이어 및 광섬유를 포함한다. 전송 매체는 정보 신호를 전송하기 위해 주파수, 진폭 또는 위상에 있어서의 변조가 가능한 반송파, 즉 전자기파의 형태를 취할 수도 있다. 또한, 전송 매체는 전파 통신 및 적외선 데이터 통신 중에 생성되는 것들과 같은 음향파 또는 광파의 형태를 취할 수 있다.

[0089] 명세서의 기술한 내용 중에서, 구현예는 이의 특정 요소를 참조하여 기술되었다. 그러나, 구현예의 더 넓은 사상 및 범주를 벗어나지 않고도 구현예에 대한 다양한 수정 및 변경이 이루어질 수 있음이 명백할 것이다. 예를 들어, 독자는 본원에 기술된 프로세스 흐름도에 나타난 프로세스 동작의 특정 순서 및 조합이 단지 예시적이며, 상이한 또는 추가의 프로세스 동작을 사용하거나 프로세스 동작의 상이한 조합 또는 순서를 사용해 구현예를 제정할 수 있다는 것을 이해할 것이다. 따라서, 명세서 및 도면은 제한적인 의미가 아니라 예시적인 의미로 간주되어야 할 것이다.

[0090] 본 발명은 다양한 컴퓨터 시스템으로 구현될 수 있다는 점에도 유의해야 한다. 본원에 기술된 다양한 기술은 하드웨어나 소프트웨어, 또는 이 둘의 조합으로 구현될 수 있다. 바람직하게는, 기술은 각각 프로세서, 프로세서에 의해 판독 가능한 저장 매체(휘발성 및 비휘발성 메모리 및/또는 저장 요소 포함), 적어도 하나의 입력 장치, 및 적어도 하나의 출력 장치를 포함하는 프로그램 가능한 컴퓨터에서 실행되는 컴퓨터 프로그램으로 구현된다. 프로그램 코드는, 기술한 기능을 수행하고 출력 정보를 생성하기 위해 입력 장치를 사용하여 입력한 데이터에 적용된다. 출력 정보는 하나 이상의 출력 장치에 적용된다. 각각의 프로그램은 컴퓨터 시스템과 통신하기 위해 고급 절차 언어 또는 객체 지향 프로그래밍 언어로 구현되는 것이 바람직하다. 그러나, 프로그램은 필요에 따라 어셈블리어 또는 기계어로 구현될 수 있다. 임의의 경우에, 언어는 컴파일러형 언어 또는 해석형 언어일 수 있다. 이러한 각각의 컴퓨터 프로그램은 바람직하게는 기술한 절차를 수행하기 위해 컴퓨터가 저장 매체 또는 장치를 판독할 때, 컴퓨터를 구성하고 작동시키기 위해 범용 또는 특수 목적의 프로그램 가능한 컴퓨터에 의해 판독 가능한 저장 매체 또는 장치(예: ROM 또는 자기 디스크)에 저장된다. 상기 시스템은 컴퓨터 프로그램으로 구성된 컴퓨터-판독 가능 저장 매체로서 구현되는 것으로 간주될 수도 있으며, 이렇게 구성된 저장 매체는 컴퓨터로 하여금 특정한 사전 정의된 방식으로 작동하게 한다. 또한, 예시적인 컴퓨팅 애플리케이션의 저장 요소는 다양한 조합 및 구성으로 데이터를 저장할 수 있는 관계형 또는 순차형(플랫 파일) 컴퓨팅 데이터베이스일

수 있다.

- [0091] 도 14는 본원에서 기술된 시스템 및 장치의 특징을 통합할 수 있는 소스 장치(1412) 및 목적지 장치(1410)의 개략도(high level view)이다. 도 14에 도시된 바와 같이, 예시적인 비디오 코딩 시스템(1410)은 소스 장치(1412) 및 목적지 장치(1414)를 포함하며, 본 예에서는 소스 장치(1412)가 인코딩된 비디오 데이터를 생성한다. 따라서, 소스 장치(1412)는 비디오 인코딩 장치로서 지칭될 수 있다. 목적지 장치(1414)는 소스 장치(1412)에 의해 생성된 인코딩된 비디오 데이터를 디코딩할 수 있다. 따라서, 목적지 장치(1414)는 비디오 디코딩 장치로서 지칭될 수 있다. 소스 장치(1412) 및 목적지 장치(1414)는 비디오 코딩 장치의 예일 수 있다.
- [0092] 목적지 장치(1414)는 채널(1416)을 통해 소스 장치(1412)로부터 인코딩된 비디오 데이터를 수신할 수 있다. 채널(1416)은 인코딩된 비디오 데이터를 소스 장치(1412)로부터 목적지 장치(1414)로 이동시킬 수 있는 유형의 매체 또는 장치를 포함할 수 있다. 일 예에서, 채널(1416)은 소스 장치(1412)가 인코딩된 비디오 데이터를 목적지 장치(1414)에 실시간으로 직접 전송할 수 있게 하는 통신 매체를 포함할 수 있다.
- [0093] 본 예에서, 소스 장치(1412)는 인코딩된 비디오 데이터를 통신 표준, 예컨대 무선 통신 프로토콜에 따라 변조할 수 있고, 변조된 비디오 데이터를 목적지 장치(1414)에 전송할 수 있다. 통신 매체는 무선 또는 유선 통신 매체, 예컨대 전파(RF) 스펙트럼 또는 하나 이상의 물리적 전송 라인을 포함할 수 있다. 통신 매체는 근거리 통신망(local area network), 광역망(wide-area network), 또는 인터넷과 같은 글로벌 네트워크(global network) 등과 같은 패킷-기반 네트워크의 일부를 형성할 수 있다. 통신 매체는 소스 장치(1412)로부터 목적지 장치(1414)로의 통신을 용이하게 하는 라우터, 스위치, 기지국 또는 다른 장비를 포함할 수 있다. 또 다른 예에서, 채널(1416)은 소스 장치(1412)에 의해 생성된 인코딩된 비디오 데이터를 저장하는 저장 매체에 상응할 수 있다.
- [0094] 도 14의 예에서, 소스 장치(1412)는 비디오 소스(1418), 비디오 인코더(1420) 및 출력 인터페이스(1422)를 포함한다. 일부 경우에, 출력 인터페이스(1428)는 변조기/복조기(모뎀) 및/또는 송신기를 포함할 수 있다. 소스 장치(1412)에서, 비디오 소스(1418)는 비디오 캡처 장치, 예를 들어, 비디오 카메라와 같은 소스, 이전에 캡처된 비디오 데이터를 포함하는 비디오 아카이브, 비디오 콘텐츠 제공자로부터 비디오 데이터를 수신하기 위한 비디오 피드 인터페이스 및/또는 비디오 데이터를 생성하기 위한 컴퓨터 그래픽 시스템, 또는 이러한 소스의 조합을 포함할 수 있다.
- [0095] 비디오 인코더(1420)는 캡처된, 미리 캡처된, 또는 컴퓨터로 생성한 비디오 데이터를 인코딩할 수 있다. 입력 이미지는 비디오 인코더(1420)에 의해 수신되어 입력 프레임 메모리(1421)에 저장될 수 있다. 범용 프로세서(1423)는 여기서 정보를 로딩하여 인코딩을 수행할 수 있다. 범용 프로세서를 구동시키기 위한 프로그램은 저장 장치, 예컨대 도 14에 도시된 예시적인 메모리 모듈로부터 로딩될 수 있다. 범용 프로세서는 인코딩을 수행하기 위해 처리 메모리(1422)를 사용할 수 있고, 범용 프로세서에 의한 인코딩 정보의 출력은 출력 버퍼(1426)와 같은 버퍼에 저장될 수 있다.
- [0096] 비디오 인코더(1420)는 적어도 하나의 기본 계층(base layer) 및 적어도 하나의 향상 계층(enhancement layer)을 정의하는 스케일러블 비디오 코딩 방식(scalable video coding scheme)으로 비디오 데이터를 코딩(예를 들어, 인코딩)하도록 구성될 수 있는 리샘플링 모듈(1425)을 포함할 수 있다. 리샘플링 모듈(1425)은 인코딩 프로세스의 일부로서 적어도 일부 비디오 데이터를 리샘플링할 수 있으며, 여기서 리샘플링은 리샘플링 필터를 사용해 적응 방식으로 수행될 수 있다.
- [0097] 인코딩된 비디오 데이터, 예를 들어, 코딩된 비트스트림은 소스 장치(1412)의 출력 인터페이스(1428)를 통해 목적지 장치(1414)에 직접 전송될 수 있다. 도 14의 예에서, 목적지 장치(1414)는 입력 인터페이스(1438), 비디오 디코더(1430) 및 디스플레이 장치(1432)를 포함한다. 일부 경우에, 입력 인터페이스(1428)는 수신기 및/또는 모뎀을 포함할 수 있다. 목적지 장치(1414)의 입력 인터페이스(1438)는 인코딩된 비디오 데이터를 채널(1416)을 통해 수신한다. 인코딩된 비디오 데이터는 비디오 인코더(1420)에 의해 생성되어 비디오 데이터를 나타내는 다양한 신택스 요소를 포함할 수 있다. 이러한 신택스 요소는 통신 매체를 이용해 전송되거나, 저장 매체에 저장되거나, 파일 서버에 저장되는 인코딩된 비디오 데이터에 함께 포함될 수 있다.
- [0098] 인코딩된 비디오 데이터도 목적지 장치(1414)가 디코딩 및/또는 재생을 위해 추후에 접근할 수 있도록 저장 매체 또는 파일 서버 상에 저장될 수 있다. 예를 들어, 코딩된 비트스트림은 입력 버퍼(1431)에 임시로 저장된 다음, 범용 프로세서(1433)에 로딩될 수 있다. 범용 프로세서를 구동시키기 위한 프로그램은 저장 장치 또는 메모리로부터 로딩될 수 있다. 범용 프로세서는 디코딩을 수행하기 위한 처리 메모리(1432)를 사용할 수 있다. 비디오 디코더(1430)는 비디오 인코더(1420)에 사용된 리샘플링 모듈(1425)과 유사한 리샘플링 모듈(1435)을 포함할

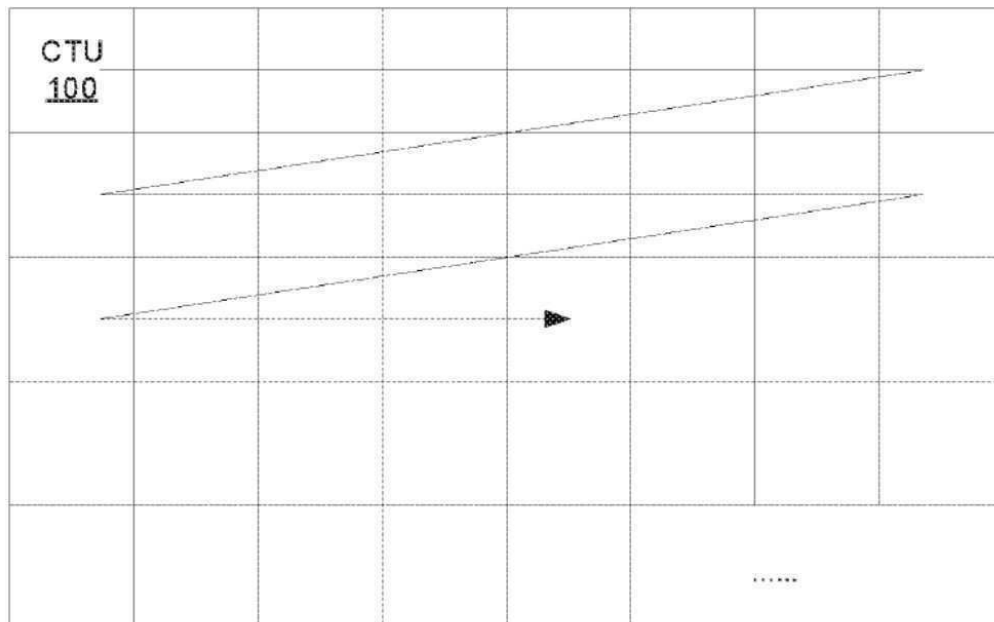
수도 있다.

- [0099] 도 14는 범용 프로세서(1433)와 별개로 리샘플링 모듈(1435)을 도시하고 있지만, 당업자는 리샘플링 기능이 범용 프로세서에 의해 실행되는 프로그램에 의해 수행될 수 있고, 비디오 인코더에서의 처리가 하나 이상의 프로세서를 사용해 달성될 수 있음을 이해할 것이다. 디코딩된 이미지(들)는 출력 프레임 버퍼(1436)에 저장된 다음 입력 인터페이스(1438)로 보내질 수 있다.
- [0100] 디스플레이 장치(1438)는 목적지 장치(1414)에 통합되거나 목적지 장치의 외부에 있을 수 있다. 일부 예에서, 목적지 장치(1414)는 통합형 디스플레이 장치를 포함할 수 있고, 외부 디스플레이 장치와 접속되도록 구성될 수도 있다. 다른 예에서, 목적지 장치(1414)는 디스플레이 장치일 수 있다. 일반적으로, 디스플레이 장치(1438)는 디코딩된 비디오 데이터를 사용자에게 디스플레이한다.
- [0101] 비디오 인코더(1420) 및 비디오 디코더(1430)는 비디오 압축 표준에 따라 동작할 수 있다. ITU-T VCEG (Q6/16) 및 ISO/IEC MPEG (JTC 1/SC 29/WG 11)는 현재의 고효율 비디오 코딩(High Efficiency Video Coding, HEVC) 표준(화면 콘텐츠 코딩 및 높은 동적 범위 코딩을 위한 현재 확장 및 단기 확장을 포함함)을 훨씬 능가하는 압축 성능을 갖는 미래 비디오 코딩 기술의 표준화에 대한 잠재적인 필요성을 연구하고 있다. 이들 그룹은 이 분야의 전문가들에 의해 제안된 압축 기술 설계를 평가하기 위해 JVET(Joint Video Exploration Team)로서 알려진 공동 협업체에서 본 탐사 활동에 협력하고 있다. 최근에 수집된 JVET 개발에 대한 내용은 J. Chen, E. Alshina, G. Sullivan, J. Ohm, J. Boyce에 의해 집필된 "Algorithm Description of Joint Exploration Test Model 5 (JEM 5)", JVET-E1001-V2에 기술되어 있다.
- [0102] 추가적으로 또는 대안적으로, 비디오 인코더(1420) 및 비디오 디코더(1430)는 개시된 JVET 특징과 함께 기능하는 다른 배타적 또는 산업 표준에 따라 동작할 수 있다. 따라서, MPEG-4, Part 10, Advanced Video Coding (AVC)으로 대안적으로 지칭되는 ITU-T H.264 표준과 같은 다른 표준 또는 이러한 표준들의 확장. 따라서, JVET를 위해 새롭게 개발되었지만, 본 개시의 기술은 임의의 특정 코딩 표준 또는 기술로 한정되지 않는다. 비디오 압축 표준 및 기술의 다른 예는 MPEG-2, ITU-T H.263 및 배타적 또는 오픈 소스 압축 포맷 및 관련 포맷을 포함한다.
- [0103] 비디오 인코더(1420) 및 비디오 디코더(1430)는 하드웨어, 소프트웨어, 펌웨어 또는 이들의 임의의 조합으로 구현될 수 있다. 예를 들어, 비디오 인코더(1420) 및 디코더(1430)는 하나 이상의 프로세서, 디지털 신호 처리기(DSP), 주문형 집적 회로(ASIC), 필드 프로그래머블 게이트 어레이(FPGA), 이산 논리회로, 또는 이들의 임의의 조합을 포함할 수 있다. 비디오 인코더(1420) 및 디코더(1430)가 부분적으로 소프트웨어로 구현되는 경우, 장치는 소프트웨어용 명령을 적합한 비일시적 컴퓨터-판독 가능한 저장 매체에 저장할 수 있고, 본 개시의 기술을 수행하기 위해 하나 이상의 프로세서를 사용해 하드웨어에서 이 명령을 실행할 수 있다. 비디오 인코더(1420) 및 비디오 디코더(1430)의 각각은 하나 이상의 인코더 또는 디코더에 포함될 수 있으며, 이들 중 어느 하나는 결합된 인코더/디코더(CODEC)의 일부로서 각 장치에 통합될 수 있다.
- [0104] 본원에서 기술된 주제의 양태는 컴퓨터-실행형 명령, 예컨대 전술한 범용 프로세서(1423 및 1433)와 같은 컴퓨터에 의해 실행되는 프로그램 모듈의 일반적인 맥락에서 기술될 수 있다. 일반적으로, 프로그램 모듈은 특정 작업을 수행하거나 특정 추상 데이터 유형을 구현하는 루틴, 프로그램, 객체, 구성 요소, 데이터 구조 등을 포함한다. 본원에서 기술된 주제의 양태는, 통신망을 통해 연결되는 원격 처리 장치들에 의해 작업이 수행되는 분산 컴퓨팅 환경에서 실시될 수도 있다. 원격 컴퓨팅 환경에서, 프로그램 모듈은 메모리 저장 장치를 포함하는 로컬 또는 원격 컴퓨터 저장 매체 모두에 위치할 수 있다.
- [0105] 메모리의 예는 랜덤 액세스 메모리(RAM), 읽기 전용 메모리(ROM) 또는 둘 다를 포함한다. 메모리는 전술한 기술을 수행할 수 있는 명령, 예컨대 소스 코드 또는 바이너리 코드를 저장할 수 있다. 메모리는 프로세서(1423 및 1433)와 같은 프로세서에 의해 실행될 명령을 실행하는 도중의 변수 또는 다른 중간 정보를 저장하는 데 사용될 수도 있다.
- [0106] 저장 장치가 전술한 기술을 수행할 수 있는 명령, 예컨대 소스 코드 또는 바이너리 코드를 저장할 수도 있다. 저장 장치는 컴퓨터 프로세서에 의해 사용되고 조작되는 데이터를 추가로 저장할 수 있다. 예를 들어, 비디오 인코더(1420) 또는 비디오 디코더(1430) 내의 저장 장치는 컴퓨터 시스템(1423 또는 1433)에 의해 접근되는 데이터베이스일 수 있다. 저장 장치의 다른 예를 랜덤 액세스 메모리(RAM), 읽기 전용 메모리(ROM), 하드 드라이브, 자기 디스크, 광학 디스크, CD-ROM, DVD, 플래시 메모리, USB 메모리 카드, 또는 컴퓨터가 판독할 수 있는 임의의 다른 매체를 포함한다.

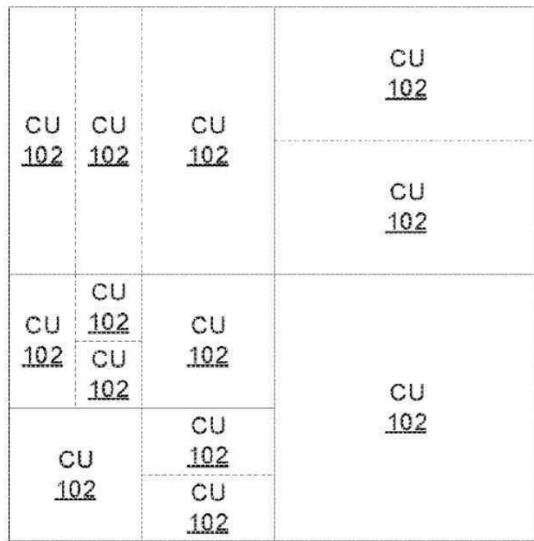
- [0107] 메모리 또는 저장 장치는 비디오 인코더 및/또는 비디오 디코더에 의해 사용되거나, 이와 결합하여 사용하기 위한 비밀시적 컴퓨터-판독 가능한 저장 매체의 일 예일 수 있다. 비밀시적 컴퓨터-판독 가능한 저장 매체는 특정 구현예에서 기술된 기능을 수행하도록 구성될 컴퓨터 시스템을 제어하기 위한 명령을 포함한다. 하나 이상의 컴퓨터 프로세서에 의해 실행될 때의 명령은 특정 구현예에서 기술된 것을 수행하도록 구성될 수 있다.
- [0108] 또한, 일부 구현예는 흐름도 또는 블록도로서 도시될 수 있는 프로세스로서 기술되었음에 유의한다. 각각은 작업을 순차적 프로세스로서 기술할 수 있지만, 많은 작업이 병렬로 또는 동시에 수행될 수 있다. 또한, 작업의 순서는 재배열될 수 있다. 프로세스는 도면에 포함되지 않은 추가의 단계를 가질 수 있다.
- [0109] 특정 구현예는 명령 실행 시스템, 장치, 시스템 또는 기계에 의해 또는 이와 결합하여 사용하기 위한 비밀시적 컴퓨터-판독 가능한 저장 매체에 구현될 수 있다. 컴퓨터-판독 가능한 저장 매체는 특정 구현예에서 기술된 방법을 수행하기 위한 컴퓨터 시스템을 제어하기 위한 명령을 포함한다. 컴퓨터 시스템은 하나 이상의 컴퓨팅 장치를 포함할 수 있다. 하나 이상의 컴퓨터 프로세서에 의해 실행될 때의 명령은 특정 구현예에서 기술된 것을 수행하도록 구성될 수 있다.
- [0110] 본원의 상세한 내용 및 후속 청구범위 전반에 걸쳐 사용된 바와 같이 단수형 표현("a", "an" 및 "the")은 문맥에 의해 명확히 달리 지시되지 않는 한 복수의 지시 대상을 포함한다. 또한, 본원의 상세한 내용 및 후속 청구범위 전반에 걸쳐 사용된 바와 같이 내("in")의 의미는 문맥에 의해 명확히 달리 지시되지 않는 한 내("in") 및 위("on")를 포함한다.
- [0111] 본 발명의 예시적인 구현예가 상기 구조적 특징 및/또는 방법론적 행위에 특이적인 언어로 상세하게 설명되었지만, 당업자는 본 발명의 신규한 교시 및 장점으로 부터 실질적으로 벗어나지 않고도 예시적인 구현예에서 많은 추가적인 수정이 가능하다는 것을 쉽게 인식할 수 있다는 점을 이해해야 한다. 또한, 첨부된 청구범위에서 정의된 주제는 본질적으로 전술한 특정 특징 또는 행위로 제한되는 것이 아님을 이해해야 한다. 따라서, 이들 및 모든 이러한 수정은 첨부된 청구범위에 따른 범위와 범주로 해석되는 본 발명의 범위에 포함되도록 의도된다.

도면

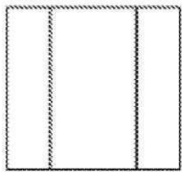
도면1



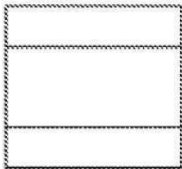
도면2a



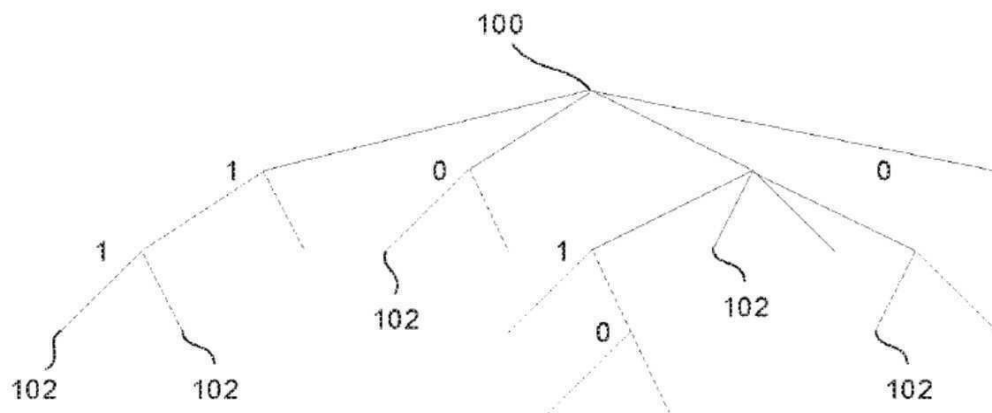
도면2b



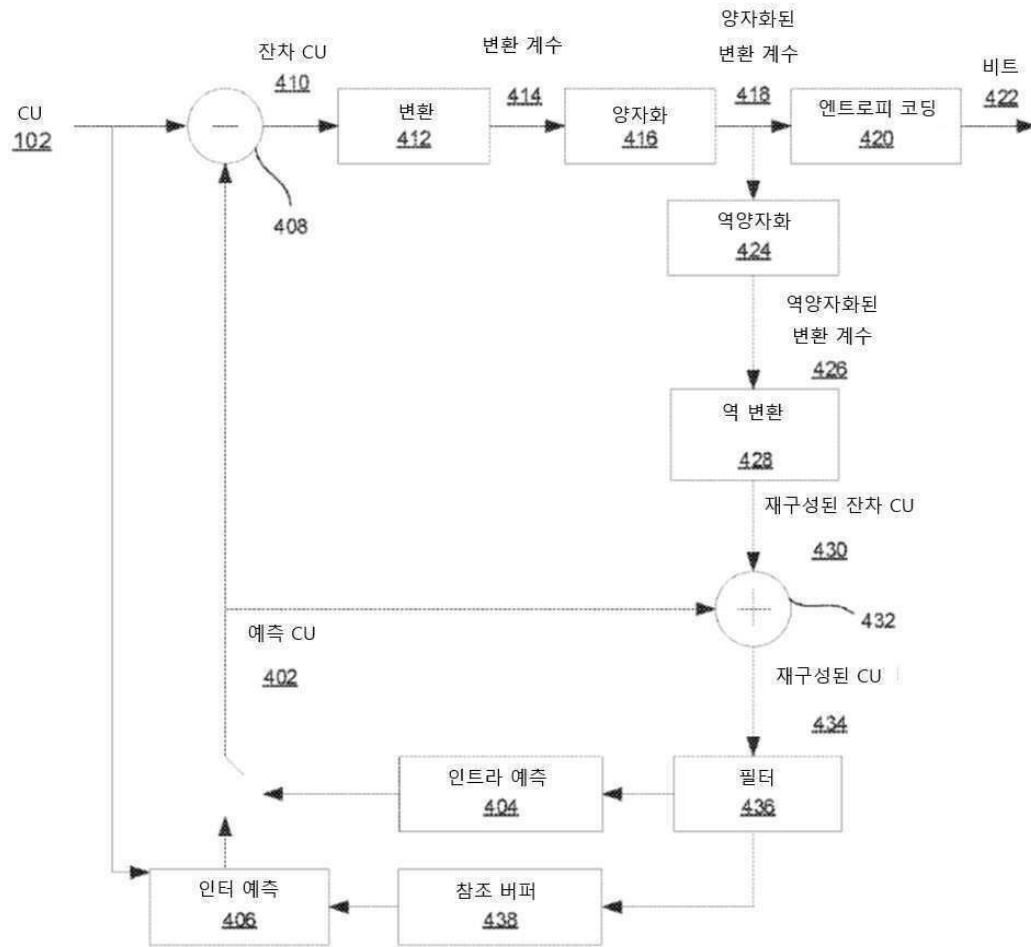
도면2c



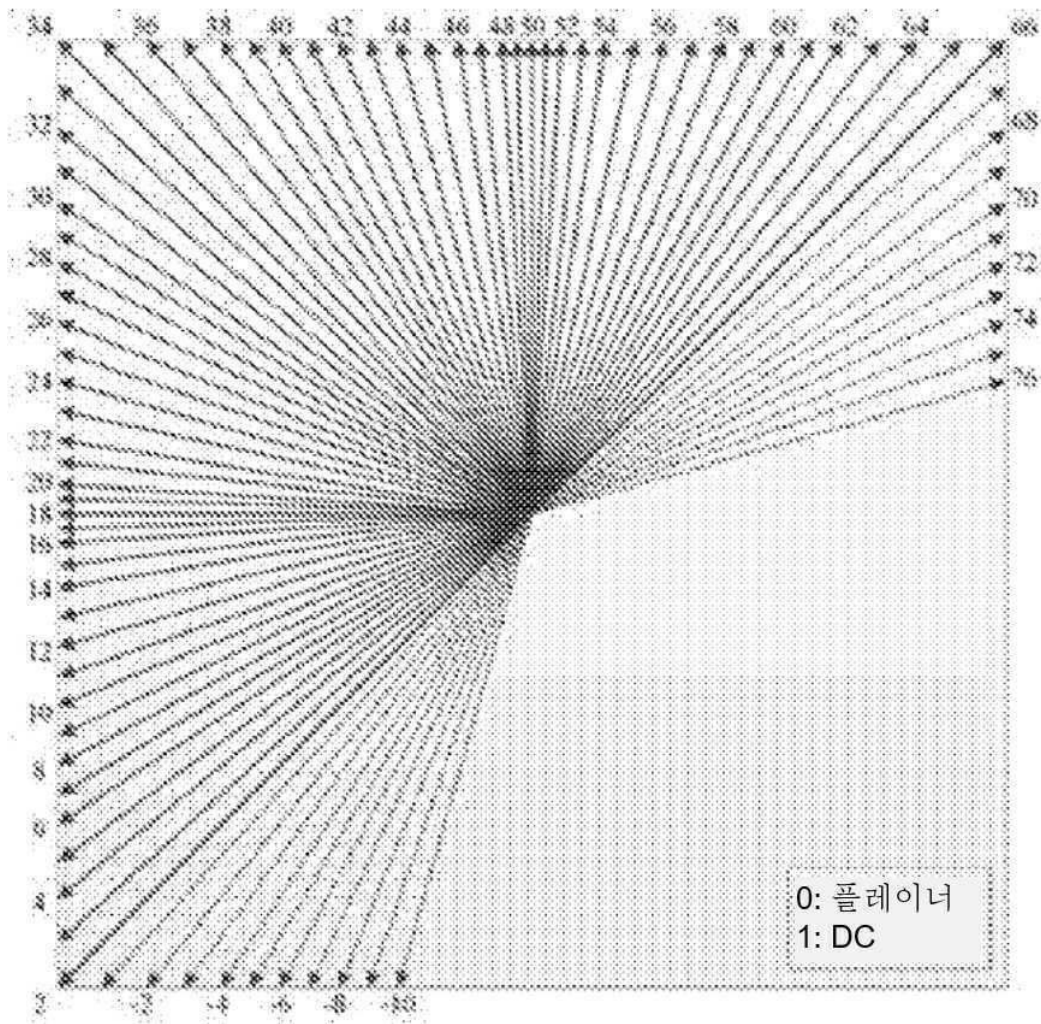
도면3



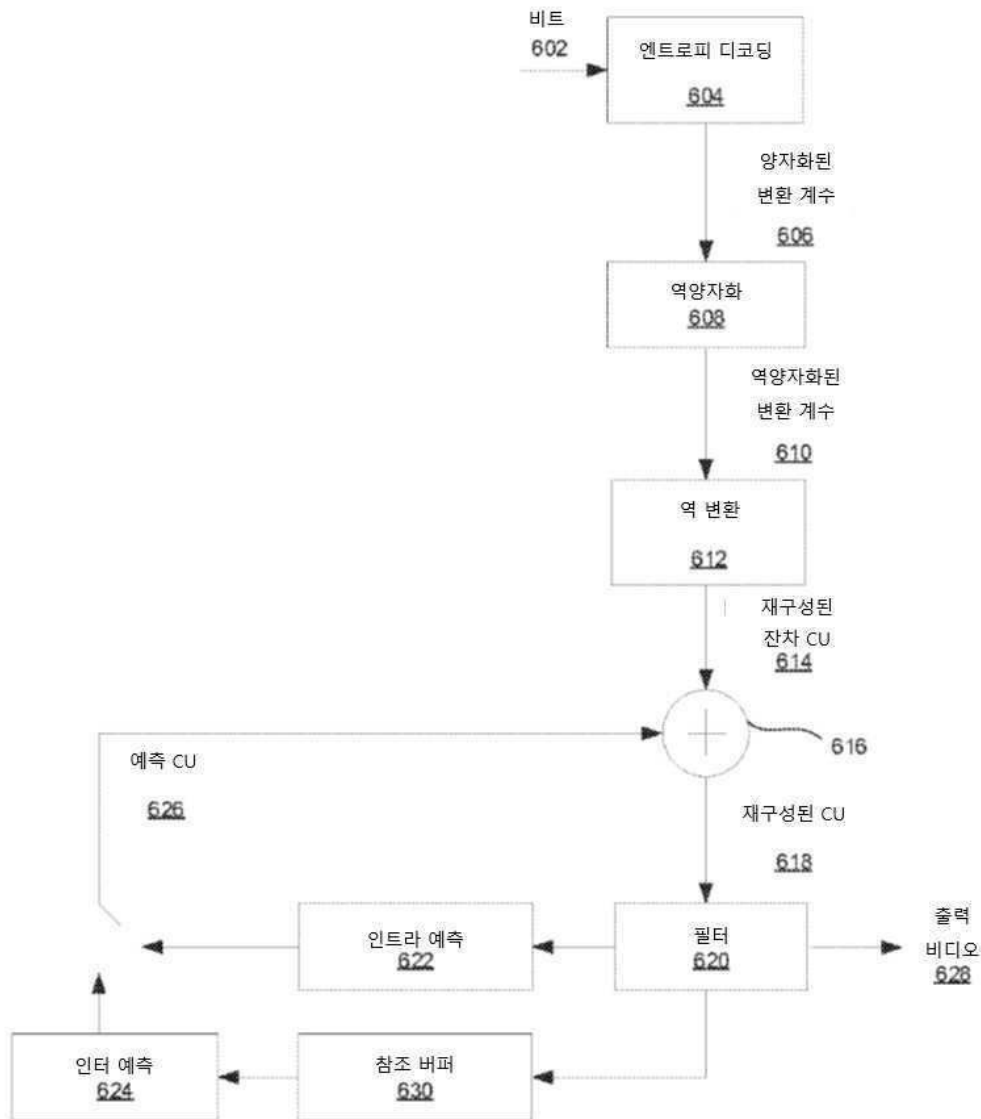
도면4



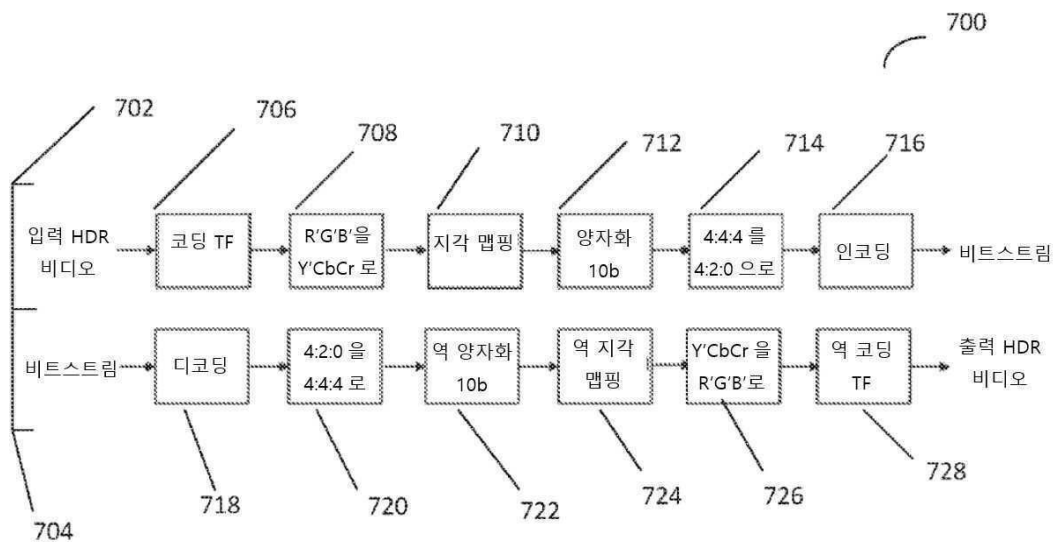
도면5



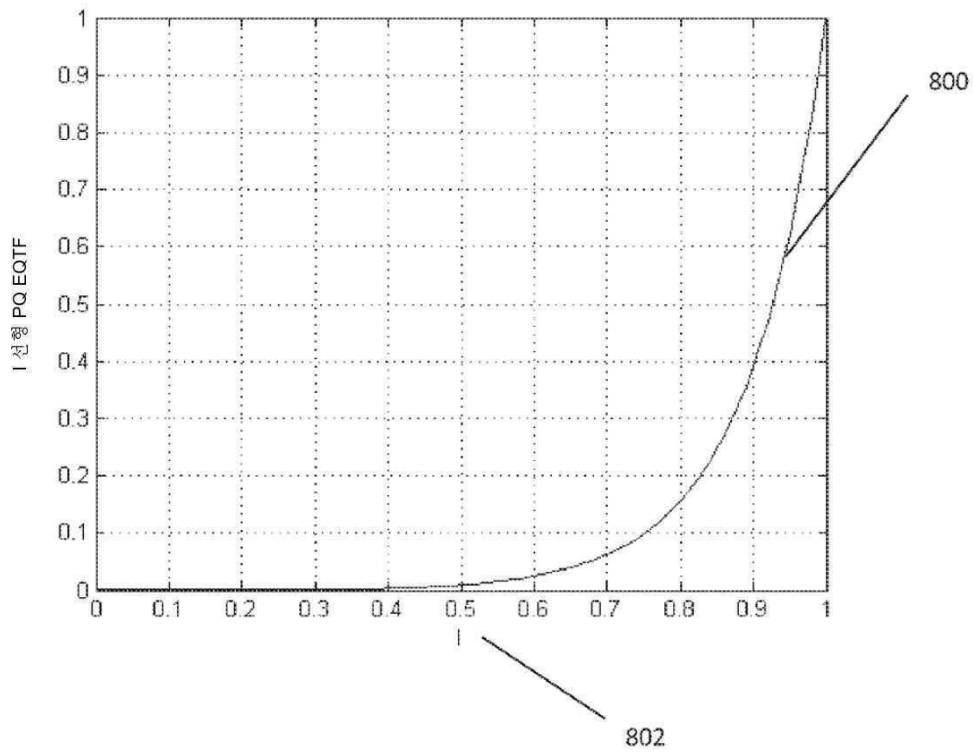
도면6



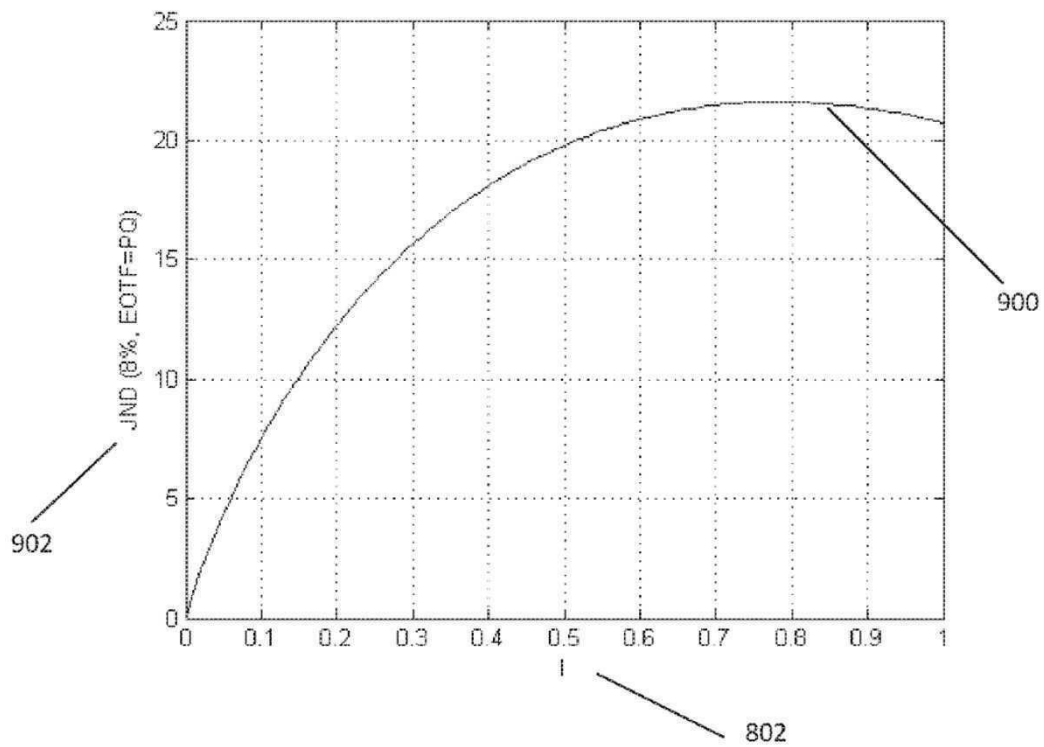
도면7



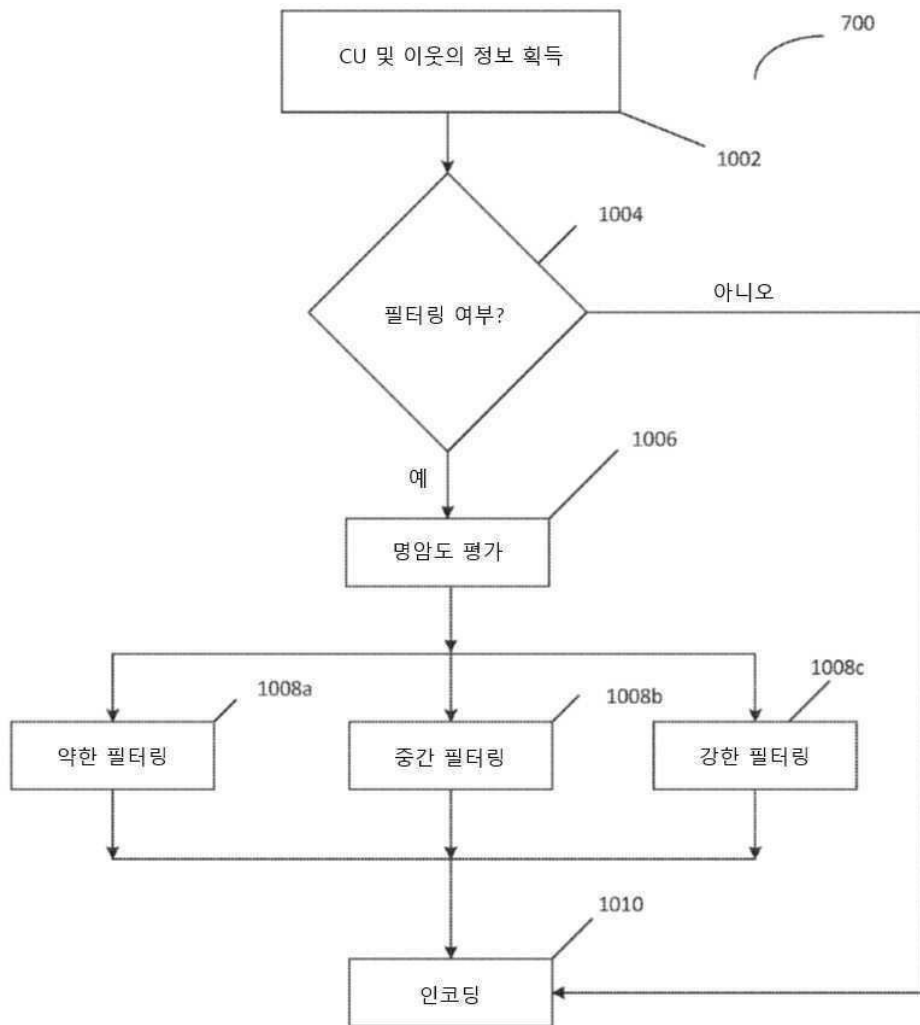
도면8



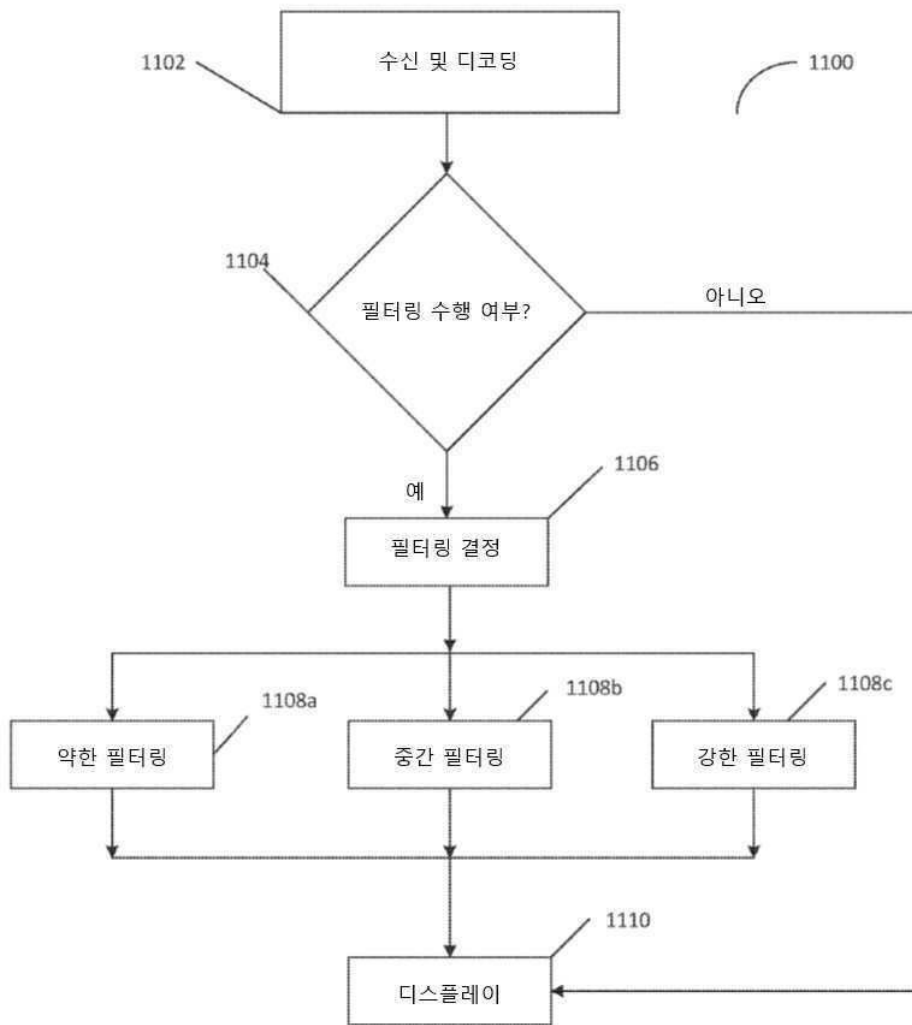
도면9



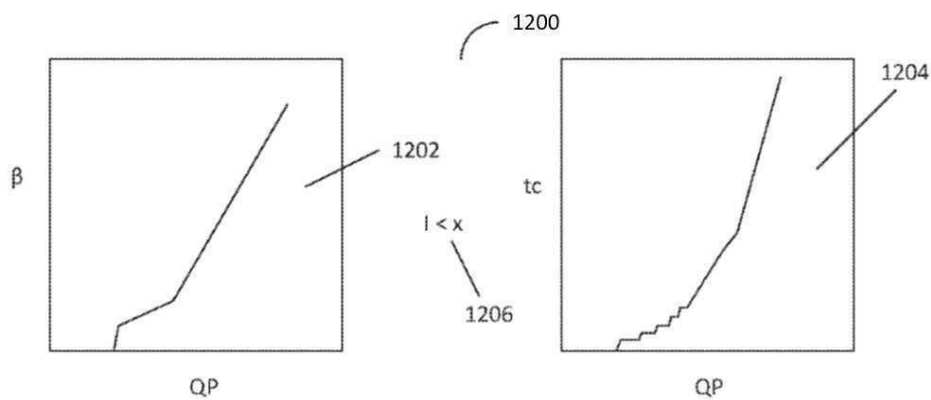
도면10



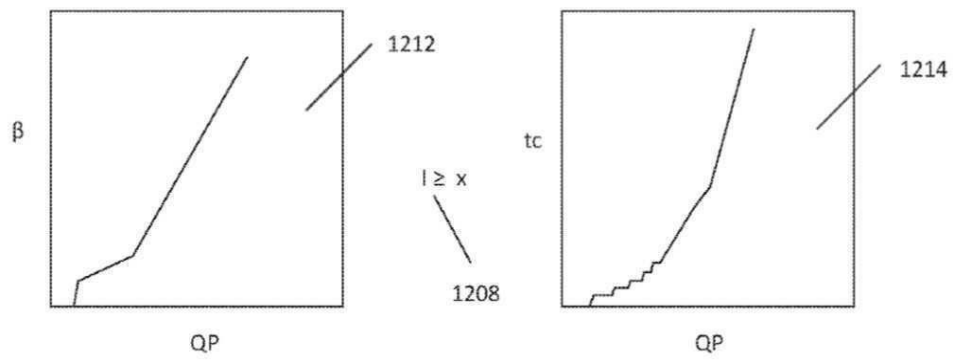
도면11



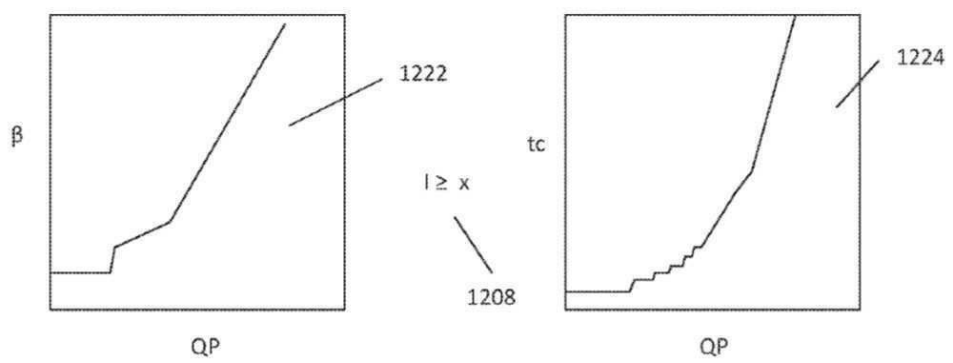
도면12a



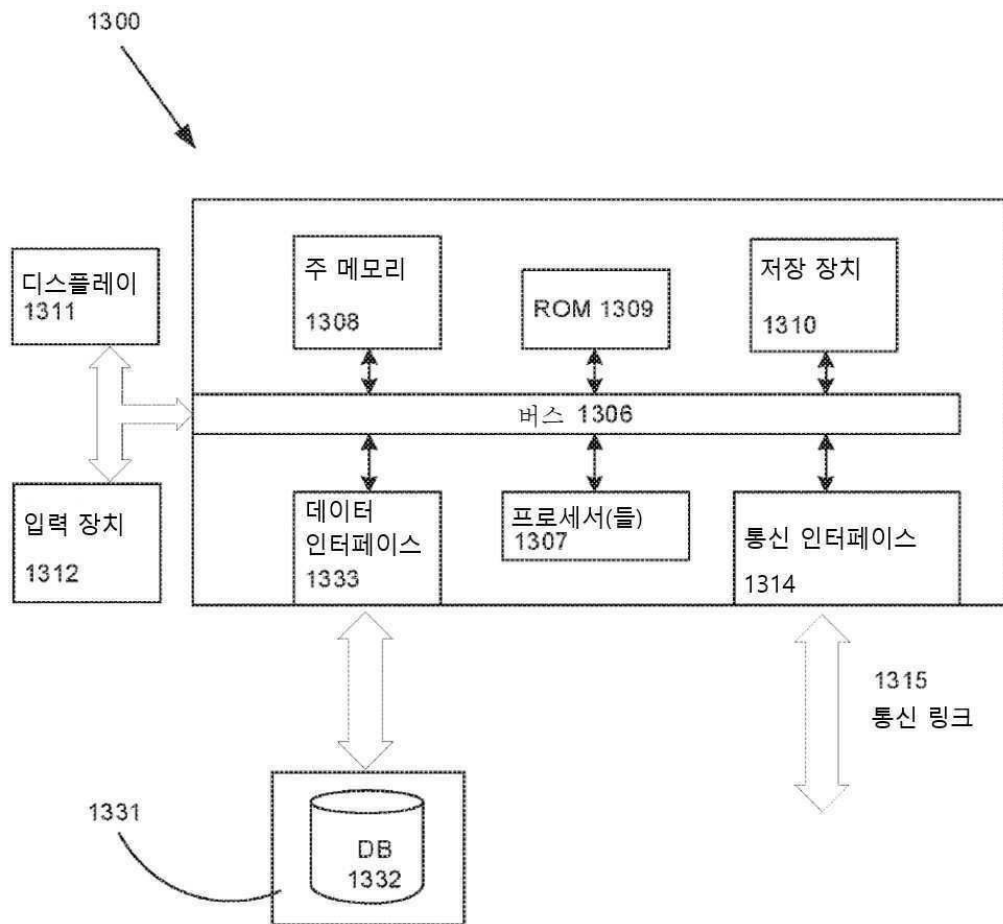
도면12b



도면12c



도면13



도면14

