



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2019-0033558
(43) 공개일자 2019년03월29일

- (51) 국제특허분류(Int. Cl.)
H04N 19/124 (2014.01) *H04N 19/176* (2014.01)
H04N 19/196 (2014.01) *H04N 19/51* (2014.01)
H04N 19/593 (2014.01) *H04N 19/61* (2014.01)
H04N 19/94 (2014.01)
- (52) CPC특허분류
H04N 19/124 (2015.01)
H04N 19/176 (2015.01)
- (21) 출원번호 10-2019-7003885
- (22) 출원일자(국제) 2017년08월11일
심사청구일자 없음
- (85) 번역문제출일자 2019년02월08일
- (86) 국제출원번호 PCT/US2017/046527
- (87) 국제공개번호 WO 2018/031899
국제공개일자 2018년02월15일
- (30) 우선권주장
62/373,884 2016년08월11일 미국(US)
15/674,035 2017년08월10일 미국(US)

- (71) 출원인
퀄컴 인코포레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
- (72) 발명자
루사노브스키 드미트로
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
부그다이지 산스리 도네
핀랜드 33720 땀뻬레 깐요닌레우나 10 에프 23
- (74) 대리인
특허법인코리아나

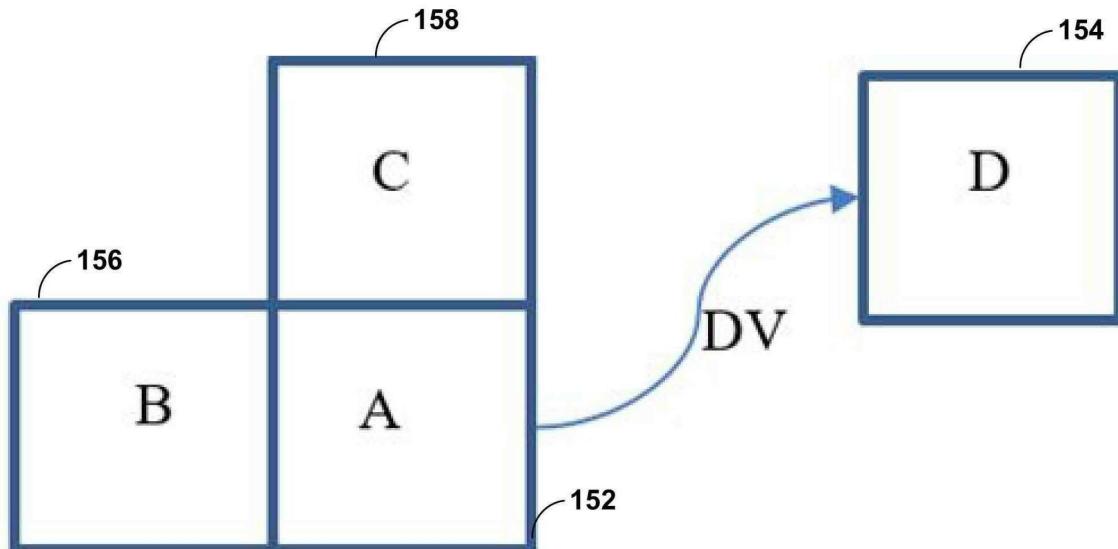
전체 청구항 수 : 총 22 항

(54) 발명의 명칭 **인-루프 샘플 프로세싱을 위한 비디오 코딩 툴들**

(57) 요 약

디바이스는 현재 블록을 포함하는 비디오 데이터를 저장하도록 구성된 메모리 디바이스, 및 메모리와 통신하는 프로세싱 회로를 포함한다. 프로세싱 회로는, 메모리 디바이스에 저장된 비디오 데이터의 하나 이상의 이웃 블록들과 연관된 하나 이상의 대응하는 파라미터 값들에 기초하는 파라미터 값을 획득하도록 구성되고, 그 하나 이상의 이웃 블록들은 현재 블록의 공간-시간적 이웃 내에 위치되고, 그 공간-시간적 이웃은 현재 블록과 연관된 디스패리티 벡터 (DV)에 의해 포인팅되는 시간적 이웃 블록 및 현재 블록에 인접하여 위치되는 하나 이상의 공간적 이웃 블록들을 포함한다. 프로세싱 회로는 또한, 메모리 디바이스에 저장된 비디오 데이터의 현재 블록을 코딩하도록 구성된다.

대 표 도



(52) CPC특허분류

H04N 19/196 (2015.01)

H04N 19/51 (2015.01)

H04N 19/593 (2015.01)

H04N 19/61 (2015.01)

H04N 19/94 (2015.01)

명세서

청구범위

청구항 1

비디오 데이터의 현재 블록을 코딩하는 방법으로서,

상기 현재 블록의 공간-시간적 이웃 내에 위치된 상기 비디오 데이터의 하나 이상의 이웃 블록들과 연관된 하나 이상의 대응하는 파라미터 값들에 기초하는 파라미터 값을 획득하는 단계로서, 상기 공간-시간적 이웃은 상기 현재 블록과 연관된 디스패리티 벡터 (DV) 에 의해 포인팅되는 시간적 이웃 블록 및 상기 현재 블록에 인접하여 위치되는 하나 이상의 공간적 이웃 블록들을 포함하고, 획득된 상기 파라미터 값은 코딩 프로세스에서 상기 현재 블록과 연관된 잔차 데이터를 수정하기 위해 사용되는, 상기 파라미터 값을 획득하는 단계; 및

상기 획득된 파라미터 값에 기초하여 상기 비디오 데이터의 상기 현재 블록을 코딩하는 단계를 포함하는, 비디오 데이터의 현재 블록을 코딩하는 방법.

청구항 2

제 1 항에 있어서,

상기 획득된 파라미터 값은 양자화 파라미터 (QP) 값을 포함하고,

상기 획득된 파라미터 값에 기초하여 상기 비디오 데이터의 상기 현재 블록을 코딩하는 단계는, 적어도 부분적으로, 상기 QP 값을 이용하여 상기 현재 블록의 샘플들을 역양자화하는 것에 의해, 상기 현재 블록을 디코딩하는 단계를 포함하는, 비디오 데이터의 현재 블록을 코딩하는 방법.

청구항 3

제 2 항에 있어서,

상기 QP 값을 획득하는 것은,

인코딩된 비디오 비트스트림에서, 엘타 양자화 파라미터 (QP) 값을 수신하는 것;

상기 공간-시간적 이웃의 샘플들에 기초하여 상기 현재 블록의 샘플들에 대한 레퍼런스 QP 값을 획득하는 것; 및

상기 현재 블록의 상기 샘플들을 역양자화하기 위한 상기 QP 값을 도출하기 위해 상기 레퍼런스 QP 값에 상기 엘타 QP 값을 가산하는 것을 포함하는, 비디오 데이터의 현재 블록을 코딩하는 방법.

청구항 4

제 1 항에 있어서,

상기 획득된 파라미터 값은 스케일링 파라미터 값을 포함하고,

상기 획득된 파라미터 값에 기초하여 상기 비디오 데이터의 상기 현재 블록을 코딩하는 단계는, 적어도 부분적으로, 상기 스케일링 파라미터 값을 이용하여 상기 현재 블록의 변환 계수들을 역 스케일링하는 것에 의해, 상기 현재 블록을 디코딩하는 단계를 포함하는, 비디오 데이터의 현재 블록을 코딩하는 방법.

청구항 5

제 4 항에 있어서,

상기 현재 블록의 상기 변환 계수들을 역 스케일링하는 것은,

복수의 역-스케일링된 DC 변환 계수들을 획득하기 위해 상기 현재 블록의 상기 변환 계수들의 복수의 DC 변환 계수들에 제 1 역 스케일링 도출 프로세스를 적용하는 것; 및

복수의 역-스케일링된 AC 변환 계수들을 획득하기 위해 상기 현재 블록의 상기 변환 계수들의 상기 복수의 역-

스케일링된 DC 변환 계수들에 제 2 역 스케일링 도출 프로세스를 적용하는 것을 포함하는, 비디오 데이터의 현재 블록을 코딩하는 방법.

청구항 6

제 1 항에 있어서,

상기 파라미터 값을 획득하는 단계는 양자화 파라미터 (QP) 값을 획득하는 단계를 포함하고,

상기 양자화 파라미터 (QP) 값을 획득하는 단계는,

상기 공간적 이웃 블록들 또는 상기 시간적 이웃 블록 중 2 개 이상의 블록들의 샘플들과 연관된 이웃 QP 값을 선택하는 단계;

선택된 상기 이웃 QP 값을 평균화하여 평균 QP 값을 획득하는 단계; 및

상기 평균 QP 값으로부터 상기 현재 블록에 대한 상기 QP 값을 도출하는 단계를 포함하고,

상기 획득된 파라미터 값에 기초하여 상기 비디오 데이터의 상기 현재 블록을 코딩하는 단계는, 적어도 부분적으로, 상기 QP 값을 이용하여 상기 현재 블록을 양자화하는 것에 의해, 상기 현재 블록을 인코딩하는 단계를 포함하는, 비디오 데이터의 현재 블록을 코딩하는 방법.

청구항 7

제 6 항에 있어서,

상기 공간-시간적 이웃의 샘플들에 기초하여 상기 현재 블록의 샘플들에 대한 레퍼런스 QP 값을 획득하는 단계;

상기 현재 블록의 샘플들에 대해 델타 양자화 파라미터 (QP) 값을 도출하기 위해 상기 QP 값을 감산하는 단계; 및

인코딩된 비디오 비트스트림에서 상기 델타 QP 값을 시그널링하는 단계를 더 포함하는, 비디오 데이터의 현재 블록을 코딩하는 방법.

청구항 8

제 1 항에 있어서,

상기 획득된 파라미터 값은 스케일링 파라미터 값을 포함하고,

상기 획득된 파라미터 값에 기초하여 상기 비디오 데이터의 상기 현재 블록을 코딩하는 단계는, 적어도 부분적으로, 상기 스케일링 파라미터 값을 이용하여 상기 현재 블록의 변환 계수들을 스케일링하는 것에 의해, 상기 현재 블록을 인코딩하는 단계를 포함하는, 비디오 데이터의 현재 블록을 코딩하는 방법.

청구항 9

제 8 항에 있어서,

상기 현재 블록의 변환 계수들을 스케일링하는 것은,

상기 현재 블록의 상기 변환 계수들의 복수의 DC 변환 계수들에 제 1 스케일링 도출 프로세스를 적용하는 것; 및

상기 현재 블록의 상기 변환 계수들의 복수의 DC 변환 계수들에 제 2 스케일링 도출 프로세스를 적용하는 것을 포함하는, 비디오 데이터의 현재 블록을 코딩하는 방법.

청구항 10

제 1 항에 있어서,

상기 획득된 파라미터 값은, 상기 현재 블록을 포함하는 슬라이스의 모든 블록들에 적용 가능한 글로벌 파라미터 값을 포함하는, 비디오 데이터의 현재 블록을 코딩하는 방법.

청구항 11

비디오 데이터를 코딩하기 위한 디바이스로서,
 상기 디바이스는,
 현재 블록을 포함하는 비디오 데이터를 저장하도록 구성된 메모리; 및
 상기 메모리와 통신하는 프로세싱 회로를 포함하고,
 상기 프로세싱 회로는,
 상기 메모리에 저장된 상기 비디오 데이터의 하나 이상의 이웃 블록들과 연관된 하나 이상의 대응하는 파라미터 값들에 기초하는 파라미터 값을 획득하는 것으로서, 상기 하나 이상의 이웃 블록들은 상기 현재 블록의 공간-시간적 이웃 내에 위치되고, 상기 공간-시간적 이웃은 상기 현재 블록과 연관된 디스패리티 벡터 (DV)에 의해 포인팅되는 시간적 이웃 블록 및 상기 현재 블록에 인접하여 위치되는 하나 이상의 공간적 이웃 블록들을 포함하고, 획득된 상기 파라미터 값은 코딩 프로세스에서 상기 현재 블록과 연관된 잔차 데이터를 수정하기 위해 사용되는, 상기 파라미터 값을 획득하는 것을 행하고; 그리고
 상기 메모리에 저장된 상기 비디오 데이터의 상기 현재 블록을 코딩하도록 구성되는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 12

제 11 항에 있어서,
 상기 획득된 파라미터 값은 양자화 파라미터 (QP) 값을 포함하고,
 상기 획득된 파라미터 값에 기초하여 상기 비디오 데이터의 상기 현재 블록을 코딩하기 위해, 상기 프로세싱 회로는, 적어도 부분적으로, 상기 QP 값을 이용하여 상기 현재 블록의 샘플들을 역양자화하는 것에 의해, 상기 현재 블록을 디코딩하도록 구성되는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 13

제 12 항에 있어서,
 상기 QP 값을 획득하기 위해, 상기 프로세싱 회로는,
 인코딩된 비디오 비트스트림에서, 엘타 양자화 파라미터 (QP) 값을 수신하고;
 상기 공간-시간적 이웃의 샘플들에 기초하여 상기 현재 블록의 샘플들에 대한 레퍼런스 QP 값을 획득하며; 그리고
 상기 현재 블록의 상기 샘플들을 역양자화하기 위한 상기 QP 값을 도출하기 위해 상기 레퍼런스 QP 값에 상기 엘타 QP 값을 가산하도록 구성되는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 14

제 11 항에 있어서,
 상기 획득된 파라미터 값은 스케일링 파라미터 값을 포함하고,
 상기 획득된 파라미터 값에 기초하여 상기 비디오 데이터의 상기 현재 블록을 코딩하기 위해, 상기 프로세싱 회로는, 적어도 부분적으로, 상기 스케일링 파라미터 값을 이용하여 상기 현재 블록의 변환 계수들을 역 스케일링하는 것에 의해, 상기 현재 블록을 디코딩하도록 구성되는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 15

제 14 항에 있어서,
 상기 현재 블록의 상기 변환 계수들을 역 스케일링하기 위해, 상기 프로세싱 회로는,
 복수의 역-스케일링된 DC 변환 계수들을 획득하기 위해 상기 현재 블록의 상기 변환 계수들의 복수의 DC 변환 계수들에 제 1 역 스케일링 도출 프로세스를 적용하고; 그리고
 복수의 역-스케일링된 AC 변환 계수들을 획득하기 위해 상기 현재 블록의 상기 변환 계수들의 상기 복수의 역-

스케일링된 DC 변환 계수들에 제 2 역 스케일링 도출 프로세스를 적용하도록 구성되는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 16

제 11 항에 있어서,

상기 파라미터 값은 양자화 파라미터 (QP) 값을 포함하고,

상기 QP 값을 획득하기 위해, 상기 프로세싱 회로는,

상기 공간적 이웃 블록들 또는 상기 시간적 이웃 블록 중 2 개 이상의 블록들의 샘플들과 연관된 이웃 QP 값을 선택하고;

선택된 상기 이웃 QP 값을 평균화하여 평균 QP 값을 획득하며; 그리고

상기 평균 QP 값으로부터 상기 현재 블록에 대한 상기 QP 값을 도출하도록 구성되고,

상기 획득된 파라미터 값에 기초하여 상기 현재 블록을 코딩하기 위해, 상기 프로세싱 회로는, 적어도 부분적으로, 상기 QP 값을 이용하여 상기 현재 블록을 양자화하는 것에 의해, 상기 현재 블록을 인코딩하도록 구성되는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 17

제 16 항에 있어서,

상기 프로세싱 회로는,

상기 공간-시간적 이웃의 샘플들에 기초하여 상기 현재 블록의 샘플들에 대한 레퍼런스 QP 값을 획득하고;

상기 현재 블록의 샘플들에 대해 엘타 양자화 파라미터 (QP) 값을 도출하기 위해 상기 QP 값으로부터 상기 레퍼런스 QP 값을 감산하며; 그리고

인코딩된 비디오 비트스트림에서 상기 엘타 QP 값을 시그널링하도록 구성되는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 18

제 11 항에 있어서,

상기 획득된 파라미터 값은 스케일링 파라미터 값을 포함하고,

상기 획득된 파라미터 값에 기초하여 상기 비디오 데이터의 상기 현재 블록을 코딩하기 위해, 상기 프로세싱 회로는, 적어도 부분적으로, 상기 스케일링 파라미터 값을 이용하여 상기 현재 블록의 변환 계수들을 스케일링하는 것에 의해, 상기 현재 블록을 인코딩하도록 구성되는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 19

제 18 항에 있어서,

상기 현재 블록의 변환 계수들을 스케일링하기 위해, 상기 프로세싱 회로는,

상기 현재 블록의 상기 변환 계수들의 복수의 DC 변환 계수들에 제 1 스케일링 도출 프로세스를 적용하고; 그리고

상기 현재 블록의 상기 변환 계수들의 복수의 DC 변환 계수들에 제 2 스케일링 도출 프로세스를 적용하도록 구성되는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 20

제 11 항에 있어서,

상기 획득된 파라미터 값은, 상기 현재 블록을 포함하는 슬라이스의 모든 블록들에 적용가능한 글로벌 파라미터 값을 포함하는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 21

비디오 데이터를 코딩하기 위한 장치로서,

상기 비디오 데이터의 현재 블록의 공간-시간적 이웃 내에 위치된 상기 비디오 데이터의 하나 이상의 이웃 블록들과 연관된 하나 이상의 대응하는 파라미터 값들에 기초하는 파라미터 값을 획득하는 수단으로서, 상기 공간-시간적 이웃은 상기 현재 블록과 연관된 디스패리티 벡터 (DV)에 의해 포인팅되는 시간적 이웃 블록 및 상기 현재 블록에 인접하여 위치되는 하나 이상의 공간적 이웃 블록들을 포함하고, 획득된 상기 파라미터 값은 코딩 프로세스에서 상기 현재 블록과 연관된 잔차 데이터를 수정하기 위해 사용되는, 상기 파라미터 값을 획득하는 수단; 및

상기 획득된 파라미터 값에 기초하여 상기 비디오 데이터의 상기 현재 블록을 코딩하는 수단을 포함하는, 비디오 데이터를 코딩하기 위한 장치.

청구항 22

명령들로 인코딩된 비-일시적 컴퓨터 판독가능 저장 매체로서,

상기 명령들은, 실행될 때, 비디오 코딩 디바이스의 프로세싱 회로로 하여금,

상기 비디오 데이터의 현재 블록의 공간-시간적 이웃 내에 위치된 상기 비디오 데이터의 하나 이상의 이웃 블록들과 연관된 하나 이상의 대응하는 파라미터 값들에 기초하는 파라미터 값을 획득하는 것으로서, 상기 공간-시간적 이웃은 상기 현재 블록과 연관된 디스패리티 벡터 (DV)에 의해 포인팅되는 시간적 이웃 블록 및 상기 현재 블록에 인접하여 위치되는 하나 이상의 공간적 이웃 블록들을 포함하고, 획득된 상기 파라미터 값은 코딩 프로세스에서 상기 현재 블록과 연관된 잔차 데이터를 수정하기 위해 사용되는, 상기 파라미터 값을 획득하는 것을 행하게 하고; 그리고

상기 획득된 파라미터 값에 기초하여 상기 비디오 데이터의 상기 현재 블록을 코딩하게 하는, 비-일시적 컴퓨터 판독가능 저장 매체.

발명의 설명

기술 분야

[0001]

본 출원은 2016년 8월 11일자로 출원된 미국 가출원 제 62/373,884 호의 이익을 주장하고, 이 가출원의 전체 내용들은 본 명세서에 참조에 의해 통합된다.

[0002]

본 개시물은 비디오 인코딩 및 비디오 디코딩에 관한 것이다.

배경 기술

[0003]

디지털 비디오 능력들은 디지털 텔레비전들, 디지털 직접 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인용 디지털 보조기들 (PDA들), 랩탑 또는 테스크탑 컴퓨터들, 태블릿 컴퓨터들, e-북 리더들, 디지털 카메라들, 디지털 레코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 무선 전화기들, 소위 "스마트 폰들", 비디오 텔레컨퍼런싱 디바이스들, 비디오 스트리밍 디바이스들 등을 포함한, 광범위한 디바이스들에 통합될 수 있다. 디지털 비디오 디바이스들은 ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 또는 ISO/IEC MPEG-2 Visual, MPEG-2, MPEG-4, MPEG-4 Visual, ITU-T H.263, ITU-T H.264/MPEG-4, Part 10, AVC (Advanced Video Coding), ISO/IEC MPEG-4 AVC ITU-T H.265, HEVC (High Efficiency Video Coding)에 의해 정의된 표준들, 및 SVC (Scalable Video Coding) 및/또는 MVC (Multi-View Video Coding) 확장들과 같은 이를 표준들의 어느 것의 확장들에서 설명된 것들과 같은 비디오 코딩 기법들을 구현한다. 비디오 디바이스들은 그러한 비디오 코딩 기법들을 구현함으로써 디지털 비디오 정보를 더 효율적으로 송신, 수신, 인코딩, 디코딩, 및/또는 저장할 수도 있다.

[0004]

비디오 코딩 기법들은 비디오 시퀀스들에 내재한 리던던시 (redundancy)를 감소 또는 제거하기 위해 공간 (인트라-픽처) 예측 및/또는 시간 (인터-픽처) 예측을 포함한다. 블록 기반 비디오 코딩에 대해, 비디오 슬라이스 (예를 들어, 비디오 프레임 또는 비디오 프레임의 부분)는 비디오 블록들로 패티셔닝될 수도 있으며, 이 비디오 블록들은 또한 트리블록들, 코딩 유닛들 (CU들) 및/또는 코딩 노드들로서 지칭될 수도 있다. 픽처의 인트라-코딩된 (I) 슬라이스에서의 비디오 블록들은 동일 픽처의 이웃 블록들에서의 레퍼런스 샘플들에 대한 공

간 예측을 이용하여 인코딩된다. 꾹처의 인터-코딩된 (P 또는 B) 슬라이스에서의 비디오 블록들은 동일 꾹처의 이웃 블록들에서의 레퍼런스 샘플들에 대한 공간 예측, 또는 다른 레퍼런스 꾹처들에서의 레퍼런스 샘플들에 대한 시간 예측을 이용할 수도 있다. 꾹처들은 프레임들로서 지칭될 수도 있으며, 레퍼런스 꾹처들은 레퍼런스 프레임들로서 지칭될 수도 있다.

[0005] 공간 또는 시간 예측은 코딩될 블록에 대한 예측성 블록을 발생시킨다. 잔차 데이터는 코딩될 오리지널 블록과 예측성 블록 간의 꾹셀 차이들을 나타낸다. 인터-코딩된 블록은 예측성 블록을 형성하는 레퍼런스 샘플들의 블록을 포인팅하는 모션 벡터, 및 코딩된 블록과 예측성 블록 간의 차이를 나타내는 잔차 데이터에 따라 인코딩된다. 인트라-코딩된 블록은 인트라-코딩 모드 및 잔차 데이터에 따라 인코딩된다. 추가적인 압축을 위해, 잔차 데이터는 꾹셀 도메인으로부터 변환 도메인으로 변환되어, 잔차 변환 계수들을 발생시킬 수도 있으며, 그 후, 이 잔차 변환 계수들은 양자화될 수도 있다. 2차원 어레이로 초기에 배열되는 양자화된 변환 계수들은 변환 계수들의 1차원 벡터를 생성하기 위해 스캐닝될 수도 있으며, 엔트로피 코딩이 훨씬 더 많은 압축을 달성하도록 적용될 수도 있다.

발명의 내용

[0006] 일반적으로, 이 개시물은 비디오 데이터의 코딩 (예컨대, 디코딩 또는 인코딩)에 관련된 기법들을 기술한다. 일부 예들에서, 이 개시의 기법들은 HDR (High Dynamic Range) 및 WCG (Wide Color Gamut) 표현들을 이용한 비디오 시그널들의 코딩에 관한 것이다. 설명된 기법들은 HEVC 또는 차세대 비디오 코딩 표준들의 확장들과 같은, 어드밴스드 (advanced) 비디오 코덱들의 맥락에서 사용될 수도 있다.

[0007] 하나의 예에서, 비디오 데이터를 코딩하기 위한 디바이스는, 메모리 및 그 메모리와 통신하는 프로세싱 회로를 포함한다. 메모리는 현재 블록을 포함하는 비디오 데이터를 저장하도록 구성다. 프로세싱 회로는, 메모리에 저장된 비디오 데이터의 하나 이상의 이웃 블록들 (neighbor blocks)과 연관된 하나 이상의 대응하는 파라미터 값들에 기초하는 파라미터 값을 획득하도록 구성된다. 하나 이상의 이웃 블록들은 현재 블록의 공간-시간적 이웃 (spatio-temporal neighborhood) 내에 위치된다. 공간-시간적 이웃은 현재 블록과 연관된 디스파리티 벡터 (disparity vector; DV)에 의해 포인팅되는 시간적 이웃 블록 및 현재 블록에 인접하여 위치되는 하나 이상의 공간적 이웃 블록들을 포함한다. 획득된 파라미터 값은 코딩 프로세스에서 현재 블록과 연관된 잔차 데이터를 수정하기 위해 사용된다. 프로세싱 회로는, 메모리에 저장된 비디오 데이터의 현재 블록을 코딩하도록 더 구성된다.

[0008] 다른 예에서, 비디오 데이터의 현재 블록을 코딩하는 방법은, 현재 블록의 공간-시간적 이웃 내에 위치된 비디오 데이터의 하나 이상의 이웃 블록들과 연관된 하나 이상의 대응하는 파라미터 값들에 기초하는 파라미터 값을 획득하는 단계를 포함한다. 공간-시간적 이웃은 현재 블록과 연관된 디스파리티 벡터 (DV)에 의해 포인팅되는 시간적 이웃 블록 및 현재 블록에 인접하여 위치되는 하나 이상의 공간적 이웃 블록들을 포함한다. 획득된 파라미터 값은 코딩 프로세스에서 현재 블록과 연관된 잔차 데이터를 수정하기 위해 사용된다. 이 방법은, 획득된 파라미터 값에 기초하여 비디오 데이터의 현재 블록을 코딩하는 단계를 더 포함한다.

[0009] 또 다른 예에서, 비디오 데이터를 코딩하기 위한 장치는, 비디오 데이터의 현재 블록의 공간-시간적 이웃 내에 위치된 비디오 데이터의 하나 이상의 이웃 블록들과 연관된 하나 이상의 대응하는 파라미터 값들에 기초하는 파라미터 값을 획득하는 수단을 포함하고, 상기 공간-시간적 이웃은 현재 블록과 연관된 디스파리티 벡터 (DV)에 의해 포인팅되는 시간적 이웃 블록 및 현재 블록에 인접하여 위치되는 하나 이상의 공간적 이웃 블록들을 포함하고, 획득된 파라미터 값은 코딩 프로세스에서 현재 블록과 연관된 잔차 데이터를 수정하기 위해 사용된다. 이 장치는, 획득된 파라미터 값에 기초하여 비디오 데이터의 현재 블록을 코딩하는 수단을 더 포함한다.

[0010] 또 다른 예에서, 비-일시적 컴퓨터 판독가능 저장 매체는 명령들로 인코딩되고, 이 명령들은, 실행될 때, 비디오 코딩 디바이스의 프로세싱 회로로 하여금, 비디오 데이터의 현재 블록의 공간-시간적 이웃 내에 위치된 비디오 데이터의 하나 이상의 이웃 블록들과 연관된 하나 이상의 대응하는 파라미터 값들에 기초하는 파라미터 값을 획득하는 것으로서, 상기 공간-시간적 이웃은 현재 블록과 연관된 디스파리티 벡터 (DV)에 의해 포인팅되는 시간적 이웃 블록 및 현재 블록에 인접하여 위치되는 하나 이상의 공간적 이웃 블록들을 포함하고, 획득된 파라미터 값은 코딩 프로세스에서 현재 블록과 연관된 잔차 데이터를 수정하기 위해 사용되는, 상기 파라미터 값을 획득하는 것을 행하게 하고, 그리고, 획득된 파라미터 값에 기초하여 비디오 데이터의 현재 블록을 코딩하게 한다.

[0011] 하나 이상의 예들의 상세한 내용들은 첨부 도면들 및 이하의 설명에서 전개된다. 다른 특징들, 목적들, 및

이 점들은 상세한 설명 및 도면들로부터, 그리고 청구항들로부터 명백해질 것이다.

도면의 간단한 설명

[0012]

도 1 은 본 개시의 기법들을 구현하도록 구성된 예시적인 비디오 인코딩 및 디코딩 시스템을 나타내는 블록도이다.

도 2 는 고 다이나믹 레인지 (high dynamic range) 데이터의 개념들을 나타내는 개념도이다.

도 3 은 예시적인 컬러 영역들 (color gamuts) 을 나타내는 개념도이다.

도 4 는 고 다이나믹 레인지 (HDR)/광색역 (WCG) 표현 변환의 일 예를 나타내는 흐름도이다.

도 5 는 예시적인 HDR/WCG 역 변환을 나타내는 흐름도이다.

도 6 은 예시적인 전달 함수들을 나타내는 개념도이다.

도 7 은 비-상수 루미넌스에 대한 일 예를 나타내는 블록도이다.

도 8 은 현재 코딩되고 있는 블록의 공간-시간적 이웃으로부터의 양자화 파라미터를 또는 스케일링 파라미터들의 도출을 위한 본 개시의 기법들을 나타내는 블록도이다.

도 9 는 비디오 인코더의 일 예를 나타내는 블록도이다.

도 10 은 비디오 디코더의 일 예를 나타내는 블록도이다.

도 11 은 비디오 디코더가 본 개시의 기법들을 구현할 수도 있는 예시적인 프로세스를 나타내는 플로우차트이다.

도 12 는 비디오 디코더가 본 개시의 기법들을 구현할 수도 있는 예시적인 프로세스를 나타내는 플로우차트이다.

도 13 은 비디오 인코더가 본 개시의 기법들을 구현할 수도 있는 예시적인 프로세스를 나타내는 플로우차트이다.

도 14 는 비디오 인코더가 본 개시의 기법들을 구현할 수도 있는 예시적인 프로세스를 나타내는 플로우차트이다.

발명을 실시하기 위한 구체적인 내용

[0013]

이 개시물은 고 다이나믹 레인지 (High Dynamic Range; HDR) 및 광색역 (Wide Color Gamut; WCG) 표현들을 이용한 비디오 신호들의 코딩에 관련된다. 보다 구체적으로, 본 개시의 기법들은 HDR 및 WCG 비디오 데이터의 보다 효율적인 압축을 가능하게 하기 위해 특정 컬러 공간들에서 비디오 데이터에 적용되는 연산들 및 시그널링을 포함한다. 제안된 기법들은 HDR 및 WCG 비디오 데이터를 코딩하기 위해 사용되는 하이브리드 기반 비디오 코딩 시스템들 (예컨대, HEVC-기반 비디오 코더들) 의 압축 효율을 향상시킬 수도 있다. 본 개시의 하나 이상의 예들의 상세한 내용들은 첨부 도면들 및 이하의 상세한 설명에서 전개된다. 다른 특징들, 목적들, 및 이 점들은 상세한 설명, 도면들로부터, 그리고 청구항들로부터 명백해질 것이다.

[0014]

도 1 은 비디오 본 개시의 기법들을 이용할 수도 있는 예시적인 비디오 인코딩 및 디코딩 시스템 (10) 을 도시한 블록도이다. 도 1 에 도시된 바와 같이, 시스템 (10) 은, 목적지 디바이스 (14) 에 의해 더 나중 시간에 디코딩될 인코딩된 비디오 데이터를 제공하는 소스 디바이스 (12) 를 포함한다. 특히, 소스 디바이스 (12) 는 비디오 데이터를, 컴퓨터 판독가능 매체 (16) 를 통해 목적지 디바이스 (14) 에 제공한다. 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 데스크탑 컴퓨터들, 노트북 (즉, 랩톱) 컴퓨터들, 태블릿 컴퓨터들, 셋탑 박스들, 소위 "스마트" 폰들과 같은 전화기 핸드셋들, 소위 "스마트" 패드들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 콘솔들, 비디오 스트리밍 디바이스 등을 포함한 광범위한 디바이스들 중 임의의 디바이스를 포함할 수도 있다. 일부 경우들에 있어서, 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 무선 통신을 위해 장비될 수도 있다.

[0015]

도 1 의 예에서, 소스 디바이스 (12) 는 비디오 소스 (18), 비디오 프리프로세서 유닛 (19) 및 비디오 인코더 (20) 를 포함하는 비디오 인코딩 유닛 (21), 및 출력 인터페이스 (22) 를 포함한다. 목적지 디바이스 (14) 는 입력 인터페이스 (28), 비디오 디코더 (30) 및 비디오 포스트프로세서 유닛 (31) 을 포함하는 비디오 디코딩

유닛 (29), 및 디스플레이 디바이스 (32) 를 포함한다. 이 개시의 일부 예에 따라, 비디오 프리프로세서 유닛 (19) 및 비디오 포스트프로세서 유닛 (31) 은 이 개시물에서 기술된 특정 기법들의 전부 또는 일부들을 수행하도록 구성될 수도 있다. 예를 들어, 비디오 프리프로세서 유닛 (19) 및 비디오 포스트프로세서 유닛 (31) 은 정적 전달 함수를 적용하도록 구성된 정적 전달 함수 유닛을 포함할 수도 있지만, 프리- 및 포스트-프로세싱 유닛들은 신호 특성들을 적응시킬 수 있다.

[0016] 다른 예들에서, 소스 디바이스 및 목적지 디바이스는 다른 컴포넌트들 또는 배열들을 포함할 수도 있다. 예를 들어, 소스 디바이스 (12) 는 외부 카메라와 같은 외부 비디오 소스 (18) 로부터 비디오 데이터를 수신할 수도 있다. 마찬가지로, 목적지 디바이스 (14) 는 통합된 디스플레이 디바이스를 포함하기보다는 외부 디스플레이 디바이스와 인터페이싱할 수도 있다.

[0017] 도 1 의 예시된 시스템 (10) 은 단지 하나의 예일 뿐이다. 비디오 데이터를 프로세싱하기 위한 기법들은 임의의 디지털 비디오 인코딩 및/또는 디코딩 디바이스에 의해 수행될 수도 있다. 비록 일반적으로 이 개시의 기법들이 비디오 인코딩 디바이스에 의해 수행되지만, 본 기법들은 또한 "CODEC" 으로서 통상적으로 지칭되는 비디오 인코더/디코더에 의해 수행될 수도 있다. 설명의 용이성을 위해, 본 개시는 소스 디바이스 (12) 및 목적지 디바이스 (14) 의 각각의 것들에서 이 개시물에서 설명된 예시적인 기법들을 수행하는 비디오 프리프로세서 유닛 (19) 및 비디오 포스트프로세서 유닛 (31) 과 관련하여 기술된다. 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 단지 소스 디바이스 (12) 가 목적지 디바이스 (14) 에의 송신을 위해 코딩된 비디오 데이터를 생성하는 그러한 코딩 디바이스들의 예들일 뿐이다. 일부 예들에서, 디바이스들 (12, 14) 은, 그 디바이스들 (12, 14) 의 각각이 비디오 인코딩 및 디코딩 컴포넌트들을 포함하도록 실질적으로 대칭적인 방식으로 동작할 수도 있다. 그리하여, 시스템 (10) 은, 예컨대, 비디오 스트리밍, 비디오 재생, 비디오 브로드캐스팅, 또는 비디오 전화에 대해, 비디오 디바이스들 (12, 14) 사이의 일방향 또는 양방향 비디오 송신을 지원할 수도 있다.

[0018] 소스 디바이스 (12) 의 비디오 소스 (18) 는 비디오 카메라와 같은 비디오 캡처 디바이스, 이전에 캡처된 비디오를 포함하는 비디오 아카이브, 및/또는 비디오 콘텐츠 프로바이더로부터 비디오 데이터를 수신하기 위한 비디오 피드 인터페이스를 포함할 수도 있다. 추가적인 대안으로서, 비디오 소스 (18) 는 그래픽-기반 데이터를 소스 비디오로서, 또는 라이브 비디오, 아카이브된 비디오, 및 컴퓨터-생성된 비디오의 조합을 생성할 수도 있다. 일부 경우들에서, 비디오 소스 (18) 가 비디오 카메라인 경우에, 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 소위 카메라 폰들 또는 비디오 폰들을 형성할 수도 있다. 소스 디바이스 (12) 는 비디오 데이터를 저장하도록 구성된 하나 이상의 데이터 저장 매체들을 포함할 수도 있다. 하지만, 상기 언급된 바와 같이, 이 개시물에서 설명된 기법들은 일반적으로 비디오 코딩에 적용가능할 수도 있고, 무선 및/또는 유선 애플리케이션들에 적용될 수도 있다. 각각의 경우에, 캡처된, 사전-캡처된, 또는 컴퓨터-생성된 비디오는 비디오 인코딩 유닛 (21) 에 의해 인코딩될 수도 있다. 인코딩된 비디오 정보는 그 다음에, 출력 인터페이스 (22) 에 의해 컴퓨터-판독가능 매체 (16) 상으로 출력될 수도 있다.

[0019] 목적지 디바이스 (14) 는 디코딩될 인코딩된 비디오 데이터를 컴퓨터 판독가능 매체 (16) 를 통해 수신할 수도 있다. 컴퓨터 판독가능 매체 (16) 는 인코딩된 비디오 데이터를 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 이동 가능한 임의의 타입의 매체 또는 디바이스를 포함할 수도 있다. 일부 예에 있어서, 컴퓨터 판독가능 매체 (16) 는, 소스 디바이스 (12) 로 하여금 인코딩된 비디오 데이터를 직접 목적지 디바이스 (14) 로 실시간으로 송신할 수 있게 하는 통신 매체를 포함할 수도 있다. 인코딩된 비디오 데이터는 무선 통신 프로토콜과 같은 통신 표준에 따라 변조되고, 목적지 디바이스 (14) 로 송신될 수도 있다. 통신 매체는 무선 주파수 (RF) 스펙트럼 또는 하나 이상의 물리적인 송신 라인들과 같은 임의의 무선 또는 유선 통신 매체를 포함할 수도 있다. 통신 매체는 로컬 영역 네트워크, 광역 네트워크, 또는 인터넷과 같은 글로벌 네트워크와 같은 패킷 기반 네트워크의 부분을 형성할 수도 있다. 통신 매체는 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로의 통신을 용이하게 하는데 유용할 수도 있는 임의의 다른 장비를 포함할 수도 있다. 목적지 디바이스 (14) 는 인코딩된 비디오 데이터 및 디코딩된 비디오 데이터를 저장하도록 구성된 하나 이상의 데이터 저장 매체들을 포함할 수도 있다.

[0020] 일부 예들에 있어서, 인코딩된 데이터는 출력 인터페이스 (22) 로부터 저장 디바이스로 출력될 수도 있다. 유사하게, 인코딩된 데이터는 입력 인터페이스에 의해 저장 디바이스로부터 액세스될 수도 있다. 저장 디바이스는 하드 드라이브, 블루-레이 디스크들, DVD들, CD-ROM들, 플래시 메모리, 휴발성 또는 비휘발성 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 임의의 다른 적합한 디지털 저장 매체들과 같은 다양한 분산된 또는 국부적으로 액세스된 데이터 저장 매체들 중 임의의 데이터 저장 매체를 포함할 수도 있다. 추가의 예

에 있어서, 저장 디바이스는, 소스 디바이스 (12)에 의해 생성된 인코딩된 비디오를 저장할 수도 있는 파일 서버 또는 다른 중간 저장 디바이스에 대응할 수도 있다. 목적지 디바이스 (14)는 저장 디바이스로부터의 저장된 비디오 데이터에 스트리밍 또는 다운로드를 통해 액세스할 수도 있다. 파일 서버는, 인코딩된 비디오 데이터를 저장하고 그리고 그 인코딩된 비디오 데이터를 목적지 디바이스 (14)로 송신하는 것이 가능한 임의의 타입의 서버일 수도 있다. 예시적인 파일 서버들은 웹 서버 (예를 들어, 웹 사이트용), FTP 서버, 네트워크 접속형 저장 (NAS) 디바이스들, 또는 로컬 디스크 드라이브를 포함한다. 목적지 디바이스 (14)는 인코딩된 비디오 데이터에, 인터넷 커넥션을 포함한 임의의 표준 데이터 커넥션을 통해 액세스할 수도 있다. 이는 파일 서버 상에 저장되는 인코딩된 비디오 데이터에 액세스하기에 적합한 무선 채널 (예를 들어, Wi-Fi 커넥션), 유선 커넥션 (예를 들어, DSL, 케이블 모뎀 등), 또는 이들 양자의 조합을 포함할 수도 있다. 인코딩된 비디오 데이터의 저장 디바이스로부터의 송신은 스트리밍 송신, 다운로드 송신, 또는 이들의 조합일 수도 있다.

[0021]

본 개시의 기법들은 무선 어플리케이션들 또는 설정들로 반드시 한정되는 것은 아니다. 그 기법들은, 공중 경유 (over-the-air) 텔레비전 브로드캐스트들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, HTTP 상으로의 동적 적응 스트리밍 (DASH)과 같은 인터넷 스트리밍 비디오 송신들, 데이터 저장 매체 상으로 인코딩된 디지털 비디오, 데이터 저장 매체 상으로 저장된 디지털 비디오의 디코딩, 또는 다른 어플리케이션들과 같은 다양한 멀티미디어 어플리케이션들 중 임의의 멀티미디어 어플리케이션의 지원으로 비디오 코딩에 적용될 수도 있다. 일부 예들에 있어서, 시스템 (10)은 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅, 및/또는 비디오 전화와 같은 어플리케이션들을 지원하기 위해 일방향 또는 양방향 비디오 송신을 지원하도록 구성될 수도 있다.

[0022]

컴퓨터 판독가능 매체 (16)는 무선 브로드캐스트 또는 유선 네트워크 송신과 같은 일시적인 매체들, 또는 하드 디스크, 플래시 드라이브, 컴팩트 디스크, 디지털 비디오 디스크, 블루-레이 디스크, 또는 다른 컴퓨터 판독가능 매체들과 같은 저장 매체들 (즉, 비-일시적인 저장 매체들)을 포함할 수도 있다. 일부 예들에 있어서, 네트워크 서버 (도시 안됨)는 인코딩된 비디오 데이터를 소스 디바이스 (12)로부터 수신하고, 인코딩된 비디오 데이터를, 예를 들어, 네트워크 송신을 통해 목적지 디바이스 (14)에 제공할 수도 있다. 유사하게, 디스크 스탬핑 설비와 같은 매체 생성 설비의 컴퓨팅 디바이스는 인코딩된 비디오 데이터를 소스 디바이스 (12)로부터 수신하고, 인코딩된 비디오 데이터를 포함하는 디스크를 생성할 수도 있다. 따라서, 컴퓨터 판독가능 매체 (16)는, 다양한 예들에 있어서, 다양한 형태들의 하나 이상의 컴퓨터 판독가능 매체들을 포함하도록 이해될 수도 있다.

[0023]

목적지 디바이스 (14)의 입력 인터페이스 (28)는 컴퓨터 판독가능 매체 (16)로부터 정보를 수신한다. 컴퓨터 판독가능 매체 (16)의 정보는 비디오 인코딩 유닛 (21)의 비디오 인코더 (20)에 의해 정의되고 또한 비디오 디코딩 유닛 (29)의 비디오 디코더 (30)에 의해 이용되는 신팩스 정보를 포함할 수도 있으며, 이 신팩스 정보는 블록들 및 다른 코딩된 유닛들, 예를 들어, GOP들 (groups of pictures)의 특성 및/또는 프로세싱을 기술하는 신팩스 엘리먼트들을 포함한다. 디스플레이 디바이스 (32)는 디코딩된 비디오 데이터를 사용자에게 디스플레이하며, 음극선관 (CRT), 액정 디스플레이 (LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (OLED) 디스플레이, 또는 다른 타입의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들 중 임의의 디바이스를 포함할 수도 있다.

[0024]

예시된 바와 같이, 비디오 프리프로세서 유닛 (19)은 비디오 소스 (18)로부터 비디오 데이터를 수신한다. 비디오 프리프로세서 유닛 (19)은 비디오 데이터를 비디오 인코더 (20)로 인코딩하기에 적합한 형태로 변환하기 위해 비디오 데이터를 프로세싱하도록 구성될 수도 있다. 예를 들어, 비디오 프리프로세서 유닛 (19)은 (예컨대, 비-선형 전달 함수를 이용하는) 다이나믹 레인지 콤팩팅 (dynamic range compacting), 보다 콤팩트하거나 강건한 컬러 공간으로의 컬러 변환, 및/또는 부동소수점-대-정수 표현 변환을 수행할 수도 있다. 비디오 인코더 (20)는 비디오 프리프로세서 유닛 (19)에 의해 출력되는 비디오 데이터에 대해 비디오 인코딩을 수행할 수도 있다. 비디오 디코더 (30)는 비디오 데이터를 디코딩하기 위해 비디오 인코더 (20)의 역을 수행할 수도 있고, 비디오 포스트프로세서 유닛 (31)은 비디오 데이터를 디스플레이를 위해 적합한 형태로 변환하기 위해 비디오 프리프로세서 유닛 (19)에 의해 수행된 동작들의 역을 수행할 수도 있다. 실제로, 비디오 포스트프로세서 유닛 (31)은 디스플레이를 위해 적합한 비디오 데이터를 생성하기 위해, 정수-대-부동소수점 변환, 콤팩트하거나 강건한 컬러 공간으로부터의 컬러 변환, 및/또는 다이나믹 레인지 콤팩팅의 역을 수행할 수도 있다.

[0025]

비디오 인코딩 유닛 (21) 및 비디오 디코딩 유닛 (29) 각각은 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들 (DSP들), 주문형 집적회로들 (ASIC들), 필드 프로그래밍가능 게이트 어레이들 (FPGA들), 이산 로직,

소프트웨어, 하드웨어, 펌웨어 또는 이들의 임의의 조합들과 같은, 고정 기능 프로세싱 회로 및/또는 프로그래밍가능 프로세싱 회로를 포함하는, 다양한 적합한 프로세싱 회로의 임의의 것으로서 구현될 수도 있다. 기법들이 부분적으로 소프트웨어로 구현될 경우, 디바이스는 적합한 비-일시적인 컴퓨터 판독가능 매체에 소프트웨어에 대한 명령들을 저장하고, 본 개시의 기법들을 수행하기 위해 하나 이상의 프로세서들을 사용하는 하드웨어로 명령들을 실행할 수도 있다. 비디오 인코딩 유닛 (21) 및 비디오 디코딩 유닛 (29) 각각은 하나 이상의 인코더들 또는 디코더들에 포함될 수도 있으며, 이들 중 어느 하나는 개별 디바이스에 있어서 결합된 인코더/디코더 (CODEC) 의 부분으로서 통합될 수도 있다.

[0026] 비록 비디오 프리프로세서 유닛 (19) 및 비디오 인코더 (20) 는 비디오 인코딩 유닛 (21) 내의 별개의 유닛들인 것으로서 도시되고 비디오 포스트프로세서 유닛 (31) 및 비디오 디코더 (30) 는 비디오 디코딩 유닛 (29) 내의 별개의 유닛들인 것으로서 도시되지만, 본 개시에서 기술된 기법들은 그렇게 제한되지 아니한다. 비디오 프리프로세서 유닛 (19) 및 비디오 인코더 (20) 는 (예컨대, 집적 회로 또는 동일 칩 내에 하우징된) 공통 디바이스로서 형성될 수도 있다. 유사하게, 비디오 포스트프로세서 유닛 (31) 및 비디오 디코더 (30) 는 (예컨대, 집적 회로 또는 동일 칩 내에 하우징된) 공통 디바이스로서 형성될 수도 있다.

[0027] 비디오 인코더 (20) 및 비디오 디코더 (30) 는 ITU-T 비디오 코딩 전문가 그룹 (Video Coding Experts Group; VCEG) 및 ISO/IEC 동영상 전문가 그룹 (Motion Picture Experts Group; MPEG) 의 비디오 코딩에 관한 공동 협력팀 (Joint Collaboration Team on Video Coding; JCT-VC) 에 의해 개발된 고효율 비디오 코딩 (HEVC) 에 따라 동작할 수도 있다. "HEVC 초안 사양 (HEVC draft specification)" 으로서 지칭되는 HEVC 표준의 초안은 「Bross 등, "High Efficiency Video Coding (HEVC) Defect Report 3", ITU-T SG16 WP3 과 ISO/IEC JTC1/SC29/WG11 의 비디오 코딩에 관한 공동 협력팀 (JCT-VC), 16차 회의, 미국 산호세, 2014년 1월, 문서 번호 JCTVC-P1003_v1」에 기술되어 있다. HEVC 초안 사양은 http://phenix.itsudparis.eu/jct/doc_end_user/documents/16_San%20Jose/wg11/JCTVC-P1003-v1.zip 으로부터 이용가능하다. HEVC 사양은 또한 <http://www.itu.int/rec/T-REC-H.265-201504-I/en> 에서 액세스될 수 있다.

[0028] 또한, HEVC 에 대한 스케일러블 비디오 코딩 확장을 생성하기 위한 진행중인 노력들이 존재한다. HEVC 의 스케일러블 비디오 코딩 확장은 SHEVC 또는 SHVC 로서 지칭될 수도 있다. 추가적으로, VCEG 및 MPEG 의 3D 비디오 코딩 (JCT-3C) 에 관한 공동 협력팀은 HEVC 에 기초하여 3DV 표준을 개발하고 있다. HEVC 에 기초한 3DV 표준을 위한 표준화 노력들의 일부는 HEVC 에 기초한 멀티-뷰 비디오 코덱 (예컨대, MV-HEVC) 의 표준화를 포함한다.

[0029] HEVC 및 다른 비디오 코딩 사양들에 있어서, 비디오 시퀀스는 통상적으로 픽처들의 시리즈를 포함한다. 픽처들은 또한 "프레임들" 로서 지칭될 수도 있다. 픽처는 S_L , S_{Cb} , 및 S_{Cr} 로 표기되는 3개의 샘플 어레이들을 포함할 수도 있다. S_L 은 루마 (luma) 샘플들의 2차원 어레이 (즉, 블록) 이다. S_{Cb} 는 Cb 크로미넌스 샘플들의 2차원 어레이이다. S_{Cr} 는 Cr 크로미넌스 샘플들의 2차원 어레이이다. 크로미넌스 샘플들은 또한, 본 명세서에서 "크로마" 샘플들로서 지칭될 수도 있다. 다른 경우들에 있어서, 픽처는 단색일 수도 있으며 오직 루마 샘플들의 어레이만을 포함할 수도 있다.

[0030] 픽처의 인코딩된 표현을 생성하기 위해, 비디오 인코더 (20) 는 코딩 트리 유닛 (CTU) 들의 세트를 생성할 수도 있다. CTU들 각각은 루마 샘플들의 코딩 트리 블록, 크로마 샘플들의 2개의 대응하는 코딩 트리 블록들, 및 코딩 트리 블록들의 샘플들을 코딩하는데 사용되는 선택스 구조들을 포함할 수도 있다. 단색 픽처들 또는 3개의 개별 컬러 평면들을 갖는 픽처들에서, CTU 는 단일 코딩 트리 블록 및 코딩 트리 블록의 샘플들을 코딩하는데 사용되는 선택스 구조들을 포함할 수도 있다. 코딩 트리 블록은 샘플들의 NxN 블록일 수도 있다. CTU 는 또한 "트리 블록" 또는 "최대 코딩 유닛 (largest coding unit)" (LCU) 이라고 지칭될 수도 있다. HEVC 의 CTU들은 H.264/AVC 와 같은 다른 표준들의 매크로블록들과 대략 유사할 수도 있다. 그러나, CTU 는 반드시 특정 사이즈로 제한되는 것은 아니고, 하나 이상의 코딩 유닛 (coding unit; CU) 들을 포함할 수도 있다. 슬라이스는 래스터 스캔 순서 (raster scan order) 로 연속적으로 순서화된 정수 (integer number) 의 CTU들을 포함할 수도 있다.

[0031] 이 개시물은 샘플들의 하나 이상의 블록들의 샘플들을 코딩하기 위해 사용되는 하나 이상의 샘플 블록들 및 선택스 구조들을 지칭하기 위해 용어 "비디오 유닛" 또는 "비디오 블록" 또는 "블록" 을 사용할 수도 있다. 비디오 유닛들의 일부 타입들은 CTU들, CU들, PU들, 변환 유닛들 (TU들), 매크로블록들, 매크로블록 파티션들, 및 기타 등을 포함할 수도 있다. 일부 상황들에서, PU 들의 논의는 매크로블록들 또는 매크로블록 파티션

들의 논의와 상호교환될 수도 있다.

[0032] 코딩된 CTU 를 생성하기 위해, 비디오 인코더 (20) 는 코딩 트리 블록들을 코딩 블록들, 이에 따라, 명칭 "코딩 트리 유닛들"로 분할하기 위해, CTU 의 코딩 트리 블록들에 대해 퀘드-트리 파티셔닝을 재귀적으로 수행할 수도 있다. 코딩 블록은 샘플들의 NxN 블록이다. CU 는, 루마 샘플 어레이, Cb 샘플 어레이, 및 Cr 샘플 어레이를 갖는 픽처의 크로마 샘플들의 2개의 대응하는 코딩 블록들 및 루마 샘플들의 코딩 블록, 및 코딩 블록들의 샘플들을 코딩하는데 사용되는 신택스 구조들을 포함할 수도 있다. 단색 픽처들 또는 3개의 개별 컬러 평면들을 갖는 픽처들에서, CU 는 단일 코딩 블록 및 코딩 블록의 샘플들을 코딩하는데 사용되는 신택스 구조들을 포함할 수도 있다.

[0033] 비디오 인코더 (20) 는 CU 의 코딩 블록을 하나 이상의 예측 블록들로 파티셔닝할 수도 있다. 예측 블록은 동일한 예측이 적용되는 샘플들의 직사각형 (즉, 정사각형 또는 비-정사각형) 블록이다. CU 의 예측 유닛 (prediction unit; PU) 은 루마 샘플들의 예측 블록, 크로마 샘플들의 2개의 대응하는 예측 블록들, 및 예측 블록들을 예측하는데 사용되는 신택스 구조들을 포함할 수도 있다. 단색 픽처들 또는 3개의 개별 컬러 평면들을 갖는 픽처들에서, PU 는 단일 예측 블록 및 예측 블록을 예측하는데 사용되는 신택스 구조들을 포함할 수도 있다. 비디오 인코더 (20) 는 CU 의 각 PU 의 예측 블록들 (예컨대, 루마, Cb, 및 Cr 예측 블록들) 에 대한 예측성 블록들 (예컨대, 루마, Cb, 및 Cr 예측성 블록들) 을 생성할 수도 있다.

[0034] 비디오 인코더 (20) 는 PU 에 대한 예측성 블록들을 생성하기 위해 인트라 예측 또는 인터 예측을 사용할 수도 있다. 비디오 인코더 (20) 가 PU 의 예측성 블록들을 생성하기 위해 인트라 예측을 사용하는 경우, 비디오 인코더 (20) 는 PU 를 포함하는 픽처의 디코딩된 샘플들에 기초하여 PU 의 예측성 블록들을 생성할 수도 있다.

[0035] 비디오 인코더 (20) 가 CU 의 하나 이상의 PU들에 대한 예측성 블록들 (예컨대, 루마, Cb, 및 Cr 예측성 블록들) 을 생성한 후에, 비디오 인코더 (20) 는 CU 에 대한 하나 이상의 잔차 블록을 생성할 수도 있다. 실제로, 비디오 인코더 (20) 는 CU 에 대한 루마 잔차 블록을 생성할 수도 있다. CU 의 루마 잔차 블록에서의 각각의 샘플은 CU 의 예측성 루마 블록들 중 하나에서의 루마 샘플과, CU 의 오리지널 루마 코딩 블록에서의 대응하는 샘플과의 사이의 차이를 나타낸다. 부가적으로, 비디오 인코더 (20) 는 CU 에 대한 Cb 잔차 블록을 생성할 수도 있다. CU 의 Cb 잔차 블록에서의 각각의 샘플은 CU 의 예측성 Cb 블록들 중 하나에서의 Cb 샘플과, CU 의 오리지널 Cb 코딩 블록에서의 대응하는 샘플과의 사이의 차이를 나타낼 수도 있다. 비디오 인코더 (20) 는 또한 CU 에 대한 Cr 잔차 블록을 생성할 수도 있다. CU 의 Cr 잔차 블록에서의 각각의 샘플은 CU 의 예측성 Cr 블록들 중 하나에서의 Cr 샘플과, CU 의 오리지널 Cr 코딩 블록에서의 대응하는 샘플과의 사이의 차이를 나타낼 수도 있다.

[0036] 또한, 비디오 인코더 (20) 는 CU 의 잔차 블록들 (예컨대, 루마, Cb, 및 Cr 잔차 블록들) 을 하나 이상의 변환 블록들 (예컨대, 루마, Cb, 및 Cr 변환 블록들) 로 분해하기 위해 퀘드-트리 파티셔닝을 사용할 수도 있다. 변환 블록은 동일한 변환이 적용되는 샘플들의 직사각형 (예를 들어, 정사각형 또는 비-정사각형) 블록이다. CU 의 변환 유닛 (TU) 은 루마 샘플들의 변환 블록, 크로마 샘플들의 2개의 대응하는 변환 블록들, 및 변환 블록 샘플들을 변환하는데 사용되는 신택스 구조들을 포함할 수도 있다. 따라서, CU 의 각각의 TU 는 루마 변환 블록, Cb 변환 블록, 및 Cr 변환 블록을 가질 수도 있다. TU 의 루마 변환 블록은 CU 의 루마 잔차 블록의 서브-블록일 수도 있다. Cb 변환 블록은 CU 의 Cb 잔차 블록의 서브-블록일 수도 있다. Cr 변환 블록은 CU 의 Cr 잔차 블록의 서브-블록일 수도 있다. 단색 픽처들 또는 3개의 개별 컬러 평면들을 갖는 픽처들에서, TU 는 단일 변환 블록 및 변환 블록의 샘플들을 변환하는데 사용되는 신택스 구조들을 포함할 수도 있다.

[0037] 비디오 인코더 (20) 는 TU 에 대한 계수 블록을 생성하기 위해 하나 이상의 변환들을 TU 의 변환 블록에 적용할 수도 있다. 실제로, 비디오 인코더 (20) 는 TU 에 대한 루마 계수 블록을 생성하기 위해 하나 이상의 변환들을 TU 의 루마 변환 블록에 적용할 수도 있다. 계수 블록은 변환 계수들의 2차원 어레이일 수도 있다. 변환 계수는 스칼라 양 (scalar quantity) 일 수도 있다. 비디오 인코더 (20) 는 TU 에 대한 Cb 계수 블록을 생성하기 위해 하나 이상의 변환들을 TU 의 Cb 변환 블록에 적용할 수도 있다. 비디오 인코더 (20) 는 TU 에 대한 Cr 계수 블록을 생성하기 위해 하나 이상의 변환들을 TU 의 Cr 변환 블록에 적용할 수도 있다.

[0038] 계수 블록 (예를 들어, 루마 계수 블록, Cb 계수 블록 또는 Cr 계수 블록) 을 생성한 후에, 비디오 인코더 (20) 는 계수 블록을 양자화할 수도 있다. 양자화는 일반적으로, 변환 계수들을 표현하는데 사용되는 데이터의 양을 가능한 대로 감소시키기 위해 변환 계수들이 양자화되어 추가의 압축을 제공하는 프로세스를 지칭한다. 비디오 인코더 (20) 가 계수 블록을 양자화한 후에, 비디오 인코더 (20) 는 양자화된 변환 계수들을 나타내는

신택스 엘리먼트들을 엔트로피 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 양자화된 변환 계수 들을 나타내는 선택스 엘리먼트들에 대해 콘텍스트-적응 이진 산술 코딩 (Context-Adaptive Binary Arithmetic Coding; CABAC) 을 수행할 수도 있다.

[0039] 비디오 인코더 (20) 는 코딩된 픽처들 및 연관된 데이터의 표현을 형성하는 비트들의 시퀀스를 포함하는 비트스 트림을 출력할 수도 있다. 따라서, 비트스트림은 비디오 데이터의 인코딩된 표현을 포함한다. 비트스트 림은 네트워크 추상화 레이어 (NAL) 유닛들의 시퀀스를 포함할 수도 있다. NAL 유닛은 NAL 유닛에서의 데이 터의 타입의 표시, 및 에뮬레이션 방지 비트들과 함께 필요에 따라 산재된 원시 바이트 시퀀스 페이로드 (raw byte sequence payload; RBSP) 의 형태로 그 데이터를 포함하는 바이트들을 포함하는 선택스 구조이다. NAL 유닛들 각각은 NAL 유닛 헤더를 포함할 수도 있고, RBSP 를 캡슐화한다. NAL 유닛 헤더는 NAL 유닛 타 입 코드를 나타내는 선택스 엘리먼트를 포함할 수도 있다. NAL 유닛의 NAL 유닛 헤더에 의해 특정된 NAL 유 닷 타입 코드는 NAL 유닛의 타입을 나타낸다. RBSP 는 NAL 유닛 내에 캡슐화되는 정수의 바이트들을 포함하 는 선택스 구조일 수도 있다. 일부의 경우들에서, RBSP 는 0 비트들을 포함한다.

[0040] 비디오 디코더 (30) 는 비디오 인코더 (20) 에 의해 생성된 비트스트림을 수신할 수도 있다. 부가적으로, 비디오 디코더 (30) 는 비트스트림으로부터 선택스 엘리먼트들을 획득하기 위해 비트스트림을 파싱 (parse) 할 수도 있다. 비디오 디코더 (30) 는 비트스트림으로부터 획득된 선택스 엘리먼트들에 적어도 부분적으로 기 초하여 비디오 데이터의 픽처들을 재구성할 수도 있다. 비디오 데이터를 재구성하기 위한 프로세스는 비디 오 인코더 (20) 에 의해 수행된 프로세스와 일반적으로 상반될 수도 있다. 예를 들어, 비디오 디코더 (30) 는 현재 CU 의 PU들에 대한 예측 블록들을 결정하기 위해 PU들의 모션 벡터들을 사용할 수도 있다. 부가적 으로, 비디오 디코더 (30) 는 현재 CU 의 TU들의 계수 블록들을 역양자화할 수도 있다. 비디오 디코더 (30) 는 현재 CU 의 TU들의 변환 블록들을 재구성하기 위해 계수 블록들에 대해 역변환들을 수행할 수도 있다. 비디오 디코더 (30) 는 현재 CU 의 PU들에 대한 예측 블록들의 샘플들을 현재 CU 의 TU들의 변환 블록들의 대응 하는 샘플들에 부가함으로써, 현재 CU 의 코딩 블록들을 재구성할 수도 있다. 픽처의 각각의 CU 에 대한 코 딩 블록들을 재구성함으로써, 비디오 디코더 (30) 는 픽처를 재구성할 수도 있다.

[0041] HDR/WCG 의 양태들이 이제 논의될 것이다. 차세대 비디오 애플리케이션들은 HDR 및 WCG 로 캡처된 장면을 표현하는 비디오 데이터로 동작할 것으로 기대된다. 이용되는 다이나믹 레인지 (dynamic range) 및 컬러 영 역 (color gamut) 의 파라미터들은 비디오 콘텐츠의 2 개의 독립적인 속성들이고, 디지털 텔레비전 및 멀티미디 어 서비스들의 목적들을 위한 그들의 사양은 몇가지 국제 표준들에 의해 정의된다. 예를 들어, 「Recommendation ITU-R BT. 709-5, "Parameter values for the HDTV standards for production and international programme exchange" (2002)」 (이하, "ITU-R BT. Rec. 709") 는 표준 다이나믹 레인지 (Standard Dynamic Range; SDR) 및 표준 색역 (standard color gamut) 과 같은, HDTV (high definition television) 를 위한 파라미터들을 정의한다. 다른 한편, ITU-R Rec. 2020 은 HDR 및 WCG 와 같은 UHDTV (ultra-high definition television) 파라미터들을 규정한다. 다른 시스템들에서 다이나믹 레인지 및 컬러 영역 속성들을 규정하는 다른 표준 개발 기구 (SDO) 문서들이 또한 존재한다. 예를 들어, P3 컬러 영역은 SMPTE-231-2 (Society of Motion Picture and Television Engineers) 에서 정의되고, HDR 의 일부 파라미터들은 SMPTE ST 2084 에서 정의된다. 비디오 데이터를 위한 다이나믹 레인지 및 컬러 영역의 간단한 설명이 이 하에서 제공된다.

[0042] 다이나믹 레인지의 양태들이 이제 논의될 것이다. 다이나믹 레인지는 통상적으로, 비디오 신호의 최소 및 최대 밝기 (brightness) 사이의 비율로서 정의된다. 다이나믹 레인지는 또한 'f-stop' 또는 "f-stops" 의 면에서 측정될 수도 있고, 여기서, 1 f-stop 은 신호 다이나믹 레인지의 배증 (doubling) 에 대응한다. MPEG 의 정의에서, HDR 콘텐츠는 16 f-stops 보다 더 많은 밝기 변화를 피처링하는 그러한 콘텐츠이다. 일부 용어들에서, 10 및 16 f-stops 사이의 레벨들은 중간 다이나믹 레인지로서 고려되지만, 그것은 다른 정의들에서 HDR 로 고려된다. 동시에, 인간 시각 시스템 (human visual system; HVS) 은 훨씬 더 큰 (예컨대, "더 광범위한 (broader)" 또는 "더 넓은 (wider)") 다이나믹 레인지를 감지하는 것이 가능하다. 하지만, HVS 는 소위 "동시적 범위 (simultaneous range)" 를 좁히기 위한 적응 메커니즘 (adaptation mechanism) 을 포함 한다.

[0043] 도 2 는 HDTV 의 SDR, 예상되는 UHDTV 의 HDR 및 HVS 다이나믹 레인지에 의해 제공되는 다이나믹 레인지의 시각화를 나타내는 개념도이다. 실제로, 도 2 는 현재의 비디오 애플리케이션들 및 서비스들이 ITU-R BT.709 에 의해 조정되고 SDR 을 제공하는 것을 나타낸다. 현재의 비디오 애플리케이션들 및 서비스들은 통상적으로 미터 제공 (m^2) 당 0.1 내지 100 칸델라 (cd/m^2 은 종종 "nits" 로서 지칭된다) 정도의 밝기 (또는 루미넌스

(luminance)) 의 범위를 지원하여, 10 f-stops 보다 더 적은 또는 미만으로 이끈다. 차세대 비디오 서비스들은 16 f-stops 의 다이나믹 레인지지를 제공할 것으로 예상되고, 비록 상세한 사양들은 현재 개발 중이지만, 몇몇 초기 파라미터들은 SMPTE ST 2084 및 ITU-R BT.2020 에서 명시되어 있다.

[0044] 컬러 영역이 이제 논의될 것이다. HDR 외에 보다 실제적인 비디오 경험을 위한 다른 양태는 종래에 컬러 영역에 의해 정의된 컬러 디멘션 (color dimension) 이다. 도 3 은 SDR 컬러 영역 (ITU-R BT.709 컬러 적, 녹 및 청 컬러 프라이머리들에 기초한 삼각형), 및 UHDTV 를 위한 더 넓은 컬러 영역 (ITU-R BT.2020 컬러 적, 녹 및 청 컬러 프라이머리들에 기초한 삼각형) 을 나타내는 개념도이다. 도 3 은 또한 자연적 컬러들의 한계들을 나타내는 (혀 형상의 영역에 의해 범위가 정해진) 소위 스펙트럼 궤적을 묘사한다. 도 3 에 의해 예시된 바와 같이, ITU-R BT.709 로부터 ITU-R BT.2020 컬러 프라이머리들로 이동하는 것은 약 70% 더 많은 컬러들 또는 더 큰 컬러들로 UHDTV 서비스들을 제공하기 위한 목적이다. D65 는 주어진 사양들에 대해 백색 컬러를 규정한다.

[0045] 컬러 영역 사양들의 몇가지 예들이 아래의 표 1 에 나타내어진다.

표 1

[0046] 표 1. 컬러 영역 파라미터들

컬러 공간	RGB 컬러 공간 파라미터들							
	백색 포인트		프라이머리 컬러들					
	xx _W	yy _W	xx _R	yy _R	xx _G	yy _G	xx _B	yy _B
DCI-P3	0.314	0.351	0.680	0.320	0.265	0.690	0.150	0.060
ITU-R BT.709	0.3127	0.3290	0.64	0.33	0.30	0.60	0.15	0.06
ITU-R BT.2020	0.3127	0.3290	0.708	0.292	0.170	0.797	0.131	0.046

[0047] HDR 비디오 데이터의 표현들의 양태들이 이제 논의될 것이다. HDR/WCG 는 통상적으로, 4:4:4 크로마 포맷 및 매우 넓은 컬러 공간 (예컨대, XYZ) 으로, 컴포넌트 당 매우 높은 정밀도로 (심지어 부동 소수점으로) 획득되고 저장된다. 국제 조명 위원회 (International Commission on Illumination) 에 의해 전개된 CIE 1931 은 XYZ 컬러 공간의 일 예이다. 이 표현은 고 정밀도를 목표로 하고, (거의) 수학적으로 무손실이다. 하지만, 이 포맷 피쳐는 많은 리던던시들을 포함할 수도 있고, 압축 목적들을 위해 최선의 것이 아니다. HVS-기반 가정을 갖는 보다 낮은 정밀도의 포맷이 종래 기술의 비디오 애플리케이션들에 대해 통상적으로 이용된다.

[0048] 압축의 목적들을 위한 비디오 데이터 포맷 변환 프로세스의 하나의 예는 도 4 의 변환 프로세스 (109) 에 의해 나타낸 바와 같이 3 개의 주요 프로세스들을 포함한다. 도 4 의 기법들은 소스 디바이스 (12) 에 의해 수행될 수도 있다. 선형 RGB 데이터 (110) 는 HDR/WCG 비디오 데이터일 수도 있고, 부동 소수점 표현으로 저장될 수도 있다. 선형 RGB 데이터 (110) 는 다이나믹 레인지 콤팩팅을 위해 비-선형 전달 함수 (transfer function; TF) 를 이용하여 콤팩트화될 수도 있다. 전달 함수 (112) 는 임의의 수의 비-선형 전달 함수들, 예컨대, SMPTE-2084 에서 정의된 바와 같은 PQ TF, 을 이용하여 선형 RGB 데이터 (110) 를 콤팩트화할 수도 있다. 일부 예들에서, 컬러 변환 프로세스 (114) 는 콤팩트화된 데이터를 하이브리드 비디오 인코더에 의한 압축을 위해 보다 적합한 보다 콤팩트하거나 강건한 컬러 공간 (예컨대, YUV 또는 YCrCb 컬러 공간) 으로 변환한다. 이 데이터는 그 다음에, 변환된 HDR' 데이터 (118) 를 생성하기 위해 부동소수점-대-정수 표현 양자화 유닛 (116) 을 이용하여 양자화된다. 이 예시적인 HDR' 데이터 (118) 는 정수 표현의 것이다. HDR' 데이터는 이제 하이브리드 비디오 인코더 (예컨대, HEVC 기법들을 적용하는 비디오 인코더 (20)) 에 의한 압축을 위해 보다 적합한 포맷의 것이다. 도 4 에서 나타낸 프로세스들의 순서는 일 예로서 주어진 것이고, 다른 적용들에서 변화할 수도 있다. 예를 들어, 컬러 변환은 TF 프로세스에 선행할 수도 있다. 또한, 컬러 컴포넌트들에 대해 추가적인 프로세싱, 예컨대, 공간적 서브샘플링, 이 적용될 수도 있다.

[0049] 디코더 측에서의 예시적인 역 변환은 프로세스 (129) 의 방식에 의해 도 5 에서 나타내어진다. 목적지 디바이스 (14) 의 비디오 포스트프로세서 유닛 (31) 은 도 5 의 기법들을 수행할 수도 있다. 변환된 HDR' 데이터 (120) 는 하이브리드 비디오 디코더 (예컨대, HEVC 기법들을 적용하는 비디오 디코더 (30)) 를 이용하여 비디오 데이터를 디코딩하는 것을 통해 목적지 디바이스 (14) 에서 획득될 수도 있다. HDR' 데이터 (120) 는 그 다음에, 역 양자화 유닛 (122) 에 의해 역 양자화될 수도 있다. 그 다음에, 역 컬러 변환 프로세스

(124) 가 역 양자화된 HDR' 데이터에 적용될 수도 있다. 역 컬러 변환 프로세스 (124)는 컬러 변환 프로세스 (114)의 역일 수도 있다. 예를 들어, 역 컬러 변환 프로세스 (124)는 HDR' 데이터를 YCrCb 포맷으로부터 다시 역으로 RGB 포맷으로 변환할 수도 있다. 다음으로, 역 전달 함수 (126)는 선형 RGB 데이터 (128)를 재형성하기 위해 전달 함수 (112)에 의해 콤팩트화되었던 다이나믹 레인지를 다시 역으로 부가하기 위해 데이터에 적용될 수도 있다. 선형 및 부동소수점 표현에서의 입력 RGB 데이터의 고 다이나믹 레인지는 이용된 비-선형 전달 함수 (TF)로 콤팩트화된다. 실제로, SMPTE ST 2084에서 정의된 바와 같은 지각적 양자화기 (perceptual quantizer; PQ) TF는 압축을 위해 보다 적합한 목표 컬러 공간, 예컨대, Y'CbCr, 으로 변환되는 것을 따르고, 그 다음에, 정수 표현을 달성하기 위해 양자화된다. 이들 엘리먼트들의 순서는 일 예로서 주어진 것이고, 실제 적용들에서는 변화할 수도 있다. 예컨대, 컬러 변환은 TF 모듈에 선행할 수도 있고, 뿐만 아니라, 추가적인 프로세싱, 예컨대, 공간적 서브샘플링이 컬러 컴포넌트들에 대해 적용될 수도 있다. 이들 3개의 컴포넌트들은 이하에서 보다 자세히 설명된다.

[0050] 전달 함수 (TF)와 같은, 도 4에 나타낸 특정 양태들이 이제 보다 자세히 논의될 것이다. 이미지 컨테이너 (image container)에서 나타나는 디지털 값들을 광학 에너지로 그리고 광학 에너지로부터 맵핑하는 것은 TF의 지식을 필요로 할 수도 있다. TF는 데이터의 다이나믹 레인지를 콤팩트화하기 위해 데이터에 적용되고, 그 데이터를 제한된 수의 비트들로 표현하는 것을 가능하게 한다. 이 함수는 통상적으로, ITU-R BT. 1886 및 Rec. 709에서 SDR에 대해 명시되는 바와 같이 최종 사용자 디스플레이의 전자-광학적 전달 함수 (EOTF)의 역을 반영하거나, 또는 아니면, HDR에 대해 SMPTE ST 2084에서 명시된 PQ TF에 대해서와 같이 밝기 변화들에 대해 HVS 감지를 근사화하는, 1-차원 (1D) 비-선형 함수이다. OETF의 역 프로세스는 코드 레벨들을 다시 역으로 루미넌스에 맵핑하는 EOTF (electro-optical transfer function)이다. 도 6은 TF들의 몇 가지 예들을 도시한다. 이를 맵핑들은 또한 각각의 R, G, 및 B 컴포넌트에 개별적으로 적용될 수도 있다. 이를 맵핑들을 R, G, 및 B 컴포넌트들에 적용하는 것은 그것들을 R', G', 및 B'으로 각각 변환할 수도 있다.

[0051] ITU-R 권고안 BT.1886에서 명시된 레퍼런스 EOTF는 다음의 식에 의해 규정된다:

$$L = a(\max[(V + b), 0])^\gamma$$

[0052] 여기서,

[0054] L : cd/m^2에서의 스크린 루미넌스

[0055] L_W : 백색에 대한 스크린 루미넌스

[0056] L_B : 흑색에 대한 스크린 루미넌스

[0057] V : 입력 비디오 신호 레벨 (정규화됨, $V = 0$ 에서 흑색, 내지 $V = 1$ 에서 백색). 권고안 ITU-R BT.709에 따라 마스터된 콘텐츠에 대해, 10-비트 디지털 코드 값들 "D"는 다음 식에 따라 V 의 값들로 맵핑된다: $V = (D-64)/876$

[0058] γ : 멱 지수 함수, $\gamma = 2.40_4$

[0059] a : 사용자 이득에 대한 변수 (레거시 "콘트래스트" 제어)

$$a = (L_W^{1/\gamma} - L_B^{1/\gamma})^\gamma$$

[0060] b : 사용자 흑색 레벨 리프트에 대한 변수 (레거시 "밝기" 제어)

$$b = \frac{L_B^{1/\gamma}}{L_W^{1/\gamma} - L_B^{1/\gamma}}$$

[0062]

[0063] 상기 변수들 a 및 b 는 $V = 1$ 이 $L = L_W$ 를 제공하고 $V = 0$ 이 $L = L_B$ 를 제공하기 위해 다음 식들을 풀어쓰여 도출된다:

$$L_B = a \cdot b^\gamma$$

$$L_W = a \cdot (1+b)^\gamma$$

[0064]

더 높은 다이나믹 레인지 데이터를 보다 많은 효율성으로 지원하기 위해, SMPTE 는 최근에 SMPTE ST-2084 라고 불리는 새로운 전달 함수를 표준화하였다. ST2084 의 사양은 다음과 같이 기술되는 EOTF 애플리케이션을 정의하였다. TF 는 정규화된 선형 R, G, B 값들에 적용되고, 이는 R', G', B' 의 비선형적 표현을 초래한다. ST2084 는 10000 nits (cd/m^2) 의 피크 밝기와 연관되는 NORM=10000 에 의한 정규화를 정의한다.

[0066]

$$\text{o } R' = \text{PQ_TF}(\max(0, \min(R/\text{NORM}, 1)))$$

[0067]

$$\text{o } G' = \text{PQ_TF}(\max(0, \min(G/\text{NORM}, 1))) \quad (1)$$

[0068]

$$\text{o } B' = \text{PQ_TF}(\max(0, \min(B/\text{NORM}, 1)))$$

$$\text{여기서, } \text{PQ_TF}(L) = \left(\frac{c_1 + c_2 L^{m_1}}{1 + c_3 L^{m_1}} \right)^{m_2}$$

[0070]

$$m_1 = \frac{2610}{4096} \times \frac{1}{4} = 0.1593017578125$$

[0071]

$$m_2 = \frac{2523}{4096} \times 128 = 78.84375$$

[0072]

$$c_1 = c_3 - c_2 + 1 = \frac{3424}{4096} = 0.8359375$$

[0073]

$$c_2 = \frac{2413}{4096} \times 32 = 18.8515625$$

[0074]

$$c_3 = \frac{2392}{4096} \times 32 = 18.6875$$

통상적으로, EOTF 는 부동소수점 정밀도를 갖는 함수로서 정의된다. 는 경우에, 이 비-선형성을 갖는 신호에는 에러가 도입되지 않는다. (OETF) 는 다음과 같이 역 PQ (inverse PQ) 함수를 이용하여 정의된다:

[0076]

$$\text{o } R = 10000 * \text{inversePQ_TF}(R')$$

[0077]

$$\text{o } G = 10000 * \text{inversePQ_TF}(G') \quad (2)$$

[0078]

$$\text{o } B = 10000 * \text{inversePQ_TF}(B')$$

따라서, 역 TF (소위 OETF) 가 적용되는 ST2084 에서 명시된 바와 같은 역 TF

$$\text{inversePQ_TF}(N) = \left(\frac{\max[(N^{1/m_2} - c_1), 0]}{c_2 - c_3 N^{1/m_2}} \right)^{1/m_1}$$

$$m_1 = \frac{2610}{4096} \times \frac{1}{4} = 0.1593017578125$$

[0080]

$$m_2 = \frac{2523}{4096} \times 128 = 78.84375$$

[0081]

$$c_1 = c_3 - c_2 + 1 = \frac{3424}{4096} = 0.8359375$$

[0082]

$$c_2 = \frac{2413}{4096} \times 32 = 18.8515625$$

[0083]

$$c_3 = \frac{2392}{4096} \times 32 = 18.6875$$

[0084]

[0085] EOTF 및 OETF 는 활동적인 연구 및 표준화의 주제들이고, 일부 비디오 코딩 시스템들에서 이용되는 TF 는 ST2084 에서 명시된 TF 와는 상이할 수도 있다.

[0086]

이제 컬러 변환 (Color Transform) 이 논의될 것이다. RGB 데이터는 종종 이미지 캡처링 센서들에 의해 생성되기 때문에, RGB 데이터가 통상적으로 입력으로서 사용된다. 하지만, 이 컬러 공간은 그것의 컴포넌트들 사이에 높은 리던던시를 가지고, 콤팩트한 표현에 최적이지 않다. 보다 콤팩트하고 보다 강건한 표현을 달성하기 위해, RGB 컴포넌트들은 통상적으로 압축을 위해 보다 적합한 보다 비상관된 컬러 공간, 예컨대 YCbCr, 으로 변환된다 (즉, 컬러 변환이 수행된다). 이 컬러 공간은 상이한 비-상관된 컴포넌트들에서 루미넌스 및 컬러 정보에서 밝기를 구분한다.

[0087]

현대의 비디오 코딩 시스템들을 위해, 보통 사용되는 또는 통상적으로 사용되는 컬러 공간은 ITU-R BT.709 에서 명시된 바와 같은 YCbCr 이다. BT.709 표준에서의 YCbCr 컬러 공간은 R' G' B' 에서 Y'CbCr (비-상수 루미넌스 표현) 로의 다음과 같은 변환 프로세스를 명시한다:

[0088]

- $Y' = 0.2126 * R' + 0.7152 * G' + 0.0722 * B'$

[0089]

- $Cb = \frac{B' - Y'}{1.8556}$ (3)

[0090]

- $Cr = \frac{R' - Y'}{1.5748}$

[0091]

상기의 것은 Cb 및 Cr 컴포넌트들에 대한 분할을 회피하는 다음과 같은 근사 변환을 이용하여 또한 구현될 수 있다:

[0092]

- $Y' = 0.212600 * R' + 0.715200 * G' + 0.072200 * B'$

[0093]

- $Cb = -0.114572 * R' - 0.385428 * G' + 0.500000 * B'$ (4)

[0094]

- $Cr = 0.500000 * R' - 0.454153 * G' - 0.045847 * B'$

[0095]

ITU-R BT.2020 표준은 RGB 에서 Y'CbCr 로의 2 가지 상이한 변환 프로세스들을 명시한다: 상수-루미넌스 (Constant-luminance; CL) 및 비-상수 루미넌스 (Non-constant luminance; NCL), 권고안 ITU-R BT. 2020, "Parameter values for ultra-high definition television systems for production and international programme exchange" (2012). RGB 데이터는 선형 광에서 있을 수도 있고, Y'CbCr 데이터는 비-선형이다.

도 7 은 비-상수 루미넌스에 대한 일 예를 나타내는 블록도이다. 특히, 도 7 은 프로세스 131 에 의한 NCL 접근법의 일 예를 나타낸다. 도 7 의 NCL 접근법은 OETF (134) 후에 R' G' B' 에서 Y'CbCr 로의 변환 (136) 을 적용한다. ITU-R BT.2020 표준은 다음과 같은 R'G'B' 에서 Y'CbCr (비-상수 루미넌스 표현) 로의 변환 프로세스를 명시한다:

[0096]

- $Y' = 0.2627 * R' + 0.6780 * G' + 0.0593 * B'$

[0097]

- $Cb = \frac{B' - Y'}{1.8814}$ (5)

[0098]

- $Cr = \frac{R' - Y'}{1.4746}$

[0099]

상기한 것은 또한, 다음과 같은 식(들)에서 기술된 바와 같이, Cb 및 Cr 컴포넌트들에 대한 분할을 회피하는 다음과 같은 근사 변환을 이용하여 구현될 수 있다:

[0100]

- $Y' = 0.262700 * R' + 0.678000 * G' + 0.059300 * B'$

[0101] $\bullet \quad Cb = -0.139630 * R' - 0.360370 * G' + 0.500000 * B' \quad (6)$

[0102] $\bullet \quad Cr = 0.500000 * R' - 0.459786 * G' - 0.040214 * B'$

[0103] 양자화/고정소수점 변환이 이제 논의될 것이다. 컬러 변환에 이어서, 여전히 고 비트-심도 (예컨대, 부동 소수점 정밀도)로 표현된 목표 컬러 공간에서의 입력 데이터는 목표 비트-심도로 변환된다. 소정의 연구들은, PQ TF 와 결합한 십-대-십이 (10-12) 비트 정밀도는 변별역 (Just-Noticeable Difference; JND) 미만의 왜곡을 갖는 16 f-stops 의 HDR 데이터를 제공하기에 충분함을 보여준다. 10-비트 정밀도로 표현된 데이터는 종래 기술의 비디오 코딩 솔루션들의 대부분으로 추가로 코딩될 수 있다. 이 양자화 (138)는 손실있는 코딩의 엘리먼트이고, 변환된 데이터에 도입된 부정확성의 소스일 수도 있다.

[0104] 다양한 예들에서, 이러한 양자화는 목표 컬러 공간에서의 코드 워드들에 적용될 수도 있다. YCbCr 이 적용되는 일 예는 아래에 나타낸다. 부동소수점 정밀도로 표현된 입력 값들 YCbCr 는 고정-비트-심도의 신호 루마 (Y) 값에 대한 BitDepthY 및 크로마 값들 (Cb, Cr) 에 대한 BitDepthC 로 변환된다.

$$D_{Y'} = Clip1_Y \left(Round \left((1 \ll (BitDepth_Y - 8)) * (219 * Y' + 16) \right) \right)$$

o

$$D_{Cb} = Clip1_C \left(Round \left((1 \ll (BitDepth_C - 8)) * (224 * Cb + 128) \right) \right)$$

o

$$D_{Cr} = Clip1_C \left(Round \left((1 \ll (BitDepth_C - 8)) * (224 * Cr + 128) \right) \right)$$

o

[0108] 여기서, $Round(x) = Sign(x) * Floor(Abs(x) + 0.5)$

[0109] $Sign(x) = -1 \text{ if } x < 0, 0 \text{ if } x=0, 1 \text{ if } x > 0$

[0110] $Floor(x) x \text{ 이하의 최대 정수}$

[0111] $Abs(x) = x \text{ if } x \geq 0, -x \text{ if } x < 0$

[0112] $Clip1_Y(x) = Clip3(0, (1 \ll BitDepth_Y) - 1, x)$

[0113] $Clip1_C(x) = Clip3(0, (1 \ll BitDepth_C) - 1, x)$

[0114] $Clip3(x, y, z) = x \text{ if } z < x, y \text{ if } z > y, z \text{ 그 외의 경우.}$

[0115] 전달 함수들 및 컬러 변환들의 일부는 신호 표현의 다이나믹 레인지에 걸친 변별역 (JND) 임계 값의 중요한 변화를 피처링하는 비디오 데이터 표현을 초래할 수도 있다. 이러한 표현들에 대해, 루마 값들의 다이나믹 레인지에 걸쳐 균일한 양자화 방식은 (다이나믹 레인지의 파티션들을 나타내는) 신호 프래그먼트들에 걸친 지각의 상이한 메리트를 갖는 양자화 에러를 도입할 것이다. 신호들에 대한 이러한 영향은 프로세싱된 데이터 범위 내의 동등하지 않은 신호-대-잡음 비율을 초래하는 비-균일 양자화를 갖는 프로세싱 시스템으로서 해석될 수도 있다. 도 7 의 프로세스 (131)는 또한, 4:4:4에서 4:2:0로의 변환 (140) 및 HEVC 4:2:0 10b 인코딩 (142)을 포함한다.

[0116] 이러한 표현의 일 예는, 컬러 프라이머리들이 ITU-R Rec. BT. 2020에서 정의되고 ST 2084 전달 함수를 갖는 비상수 루미넌스 (NCL) YCbCr 컬러 공간에서 표현된 비디오 신호이다. 아래의 표 2에서 예시된 바와 같이, 이 표현 (예컨대, NCL YCbCr 컬러 공간에서 표현된 비디오 신호)은 신호의 저 강도 값들에 대해 현저하게 더 큰 양의 코드워드들을 할당한다. 실제로, 코드워드들의 30%는 10 nits 미만 (<10 nits)의 선형 광 샘플들을 나타낸다. 반면에, 고 강도 샘플들 (높은 밝기)은 상당히 더 작은 양의 코드워드들로 표현된다. 실제로, 코드워드들의 25%는 범위 1000-10,000 nits에서 선형 광에 대해 할당된다. 결과로서, 데이터의 모든 범위들에 대한 균일한 양자화를 피처링 (featuring) 하는, H.265/HEVC 비디오 코딩 시스템과 같은, 비디오 코딩 시스템은 고 강도 샘플들 (신호의 밝은 영역)에 훨씬 더 심각한 코딩 아티팩트들을 도입할 것인 반면에, 저 강도 샘플들 (동일한 신호의 어두운 영역)에 도입되는 왜곡은 두드러진 차이 훨씬 아래일 것이다.

[0117] 사실상, 상기 기술된 팩터들은, 비디오 코딩 시스템 설계, 또는 인코딩 알고리즘들이 매 선택되는 비디오 데이

터 표현에 대해, 즉, 매 선택되는 전달 함수 및 컬러 공간에 대해 조정될 필요가 있을 수도 있음을 의미할 수도 있다. 코드워드 차이들 때문에, SDR 코딩 디바이스들은 HDR 콘텐츠에 대해 최적화되지 않을 수도 있다. 또한, (Rec. 709 에 의해 제공되는) SCG 컬러들 및 SDR 다이나믹 레인지에서 상당한 양의 비디오 콘텐츠가 캡처되었다. HDR 및 WCG 에 비해, SDR-SCG 비디오 캡처는 좁은 범위를 제공한다. 이와 같이, SDR-SCG 캡처된 비디오 데이터는 HDR-WCG 비디오 데이터에 대해 비교적 작은 코드워드 방식의 풋프린트 (footprint) 를 점유할 수도 있다. 예시를 위해, Rec. 709 의 SCG 는 CIE 1931 컬러 공간의 35.9% 를 커버하는 한편, Rec. 2020 의 WCG 는 75.8% 를 커버한다.

표 2

선형 광 강도 (cd/m^2)	전체 범위	SDI 범위	좁은 범위
~ 0.01	21	25	83
~ 0.1	64	67	119
~ 1	153	156	195
~ 10	307	308	327
~ 100	520	520	509
~ 1,000	769	767	723
~ 4,000	923	920	855
~ 10,000	1023	1019	940

표 2. SMPTE ST 2084 에서의 코드 값 및 선형 광 강도 사이의 관계 (비트 심도 = 10)

상기 표 2 에서 나타낸 바와 같이, ("전체 범위 (full range)" 칼럼에서 나타낸) 코드워드들의 높은 집중은 낮은-밝기 범위에서 집중된다. 즉, (코드워드들의 대략적으로 30% 를 구성하는) 총 307 코드워드들은 선형 광 강도의 0-10 nits 범위 내에서 클러스터링된다. 낮은-밝기 시나리오들에서, 컬러 정보는 쉽게 감지가능하지 않을 수도 있고, 낮은 레벨들의 시각적 감도에서 가시적일 수도 있다. 낮은-밝기 범위에서 포지셔닝되는 코드워드들의 집중된 클러스터링 때문에, 비디오 인코딩 디바이스는 높은 품질 또는 매우 높은 품질로, 낮은-밝기 범위에서 상당한 양을 인코딩할 수도 있다. 더욱이 비트스트림은 인코딩된 잡음을 반송하기 위해 더 큰 양들의 대역폭을 소비할 수도 있다. 비디오 디코딩 디바이스는, 비트스트림을 재구성할 때, 비트스트림에 포함되는 인코딩된 잡음으로 인해, 더 큰 수의 아티팩트들을 생성할 수도 있다.

비-최적의 시각적 품질 코드워드 분배를 향상시키기 위한 기준의 제안들이 이하에서 논의된다. 하나의 이러한 제안은 「"Dynamic Range Adjustment SEI to enable High Dynamic Range video coding with Backward-Compatible Capability," by D. Rusanovskyy, A. K. Ramasubramonian, D. Bugdayci, S. Lee, J. Sole, M. Karczewicz, VCEG document COM16-C 1027-E, Sep.2015 (이하 "Rusanovskyy I")」 이다. Rusanovskyy I 은 비디오 코딩 이전에 비디오 데이터에 코드워드들 재분배를 적용하기 위한 제안을 포함하였다. 이 제안에 따르면, ST 2084/BT.2020 표현에서의 비디오 데이터는 비디오 암축 이전에 코드워드 재분배를 겪는다. 이 제안적 도입된 재분배는 동적 범위 조정 (Dynamical Range Adjustment) 을 통해 데이터의 동적 범위 내에서 감지된 왜곡 (신호 대 잡음 비) 의 선형화를 도입한다. 이 재분배는 비트레이트 제약 하에서 시각적 품질을 향상시키기 위해 발견되었다. 재분배를 보상하고 데이터를 원래의 ST 2084/BT.2020 표현으로 변환하기 위해, 비디오 디코딩 후에 데이터에 역 프로세스가 적용된다. Rusanovskyy I 에 의해 제안된 기법들은 추가적으로, (가 특허 출원 번호 제 62/149,446 호에 대해 우선권을 주장하는) 미국 특허 출원 제 15/099,256 호 및 (가 특허 출원 번호 제 62/184,216 호에 대해 우선권을 주장하는) 미국 특허 출원 제 15/176,034 호에서 추가로 기술되고, 이들의 각각의 전체 내용은 그 전체가 본원에 통합된다.

하지만, Rusanovskyy I 에서 기술된 기법들에 따르면, 프리- 및 포스트-프로세싱의 프로세스들은 일반적으로, 블록-기반 기초에서 중개 기술의 인코더들에 의해 채용되는 레이트 왜곡 최적화 프로세싱으로부터 디커플링된다. 따라서, 기술된 기법들은 비디오 코덱의 코딩 루프 밖 (또는 외부) 에 있는, 프리-프로세싱 및 포스트-프로세싱의 관점으로부터의 것이다.

또 다른 이러한 제안은 「"Performance investigation of high dynamic range and wide color gamut video coding techniques," by J. Zhao, S.-H. Kim, A. Segall, K. Misra, VCEG document COM16-C 1030-E, Sep.2015 (이하 "Zhao I")」 이다. Zhao 는 Y2020 (ST2084/BT2020) 및 Y709 (BT1886/BT 2020) 표현들에 대해 적용되는 비디오 코딩 사이에 비트레이트 할당 및 시각적으로-지각되는 왜곡을 정렬하기 위해 강도 종속적

공간적으로 변화하는 (블록 기반) 양자화 방식을 제안하였다. 위해, Y2020 및 Y709에서의 신호의 양자화는,

루마에서 동일한 레벨의 양자화를 유지하기

$$[0124] QP_Y2020 = QP_Y709 - f(Y2020)$$

[0125] 이도록, 루마에 의존하는 값에 의해 차별화될 수도 있음이 관찰되었다.

[0126] 함수 $f(Y2020)$ 은 Y2020에서 비디오의 강도 값들(밝기 레벨)에 대해 선형인 것으로 발견되었고, 그것은,

$$[0127] f(Y2020) = \max(0.03 * Y2020 - 3, 0)$$

[0128]으로서 극사화될 수도 있다.

[0129] Zhao I이 제안한 인코딩 스테이지에서 도입되는 공간적으로 변화하는 양자화 방식은 ST 2084/BT.2020 표현에서 코딩된 비디오 신호에 대한 시각적으로 지각되는 신호-대-양자화 잡음 비를 향상시킬 수 있는 것으로 발견되었다.

[0130] Zhao I에서 제안된 기법들의 잠재적인 결점은 QP 적응의 블록-기반 입도(granularity)이다. 통상적으로, 압축을 위해 인코더 측에서 선택된 이용된 블록 사이즈들은 레이트 왜곡 최적화 프로세스를 통해 도출되고, 비디오 신호의 동적 범위 특성들을 나타내지 않을 수도 있다. 따라서, 선택된 QP 설정들은 블록 내부의 신호에 대해 차선적일 수도 있다. 이러한 잠재적 문제점은 보다 큰 디멘션들의 예측 및 변환 블록 사이즈들을 채용하는 경향이 있는 차세대 비디오 코딩 시스템들에 대해 더욱 더 중요하게 될 수도 있다. 이 설계의 다른 양태는 QP 적응 파라미터들의 시그널링에 대한 필요성이다. QP 적응 파라미터들은 역 양자화해제를 위해 디코더에 시그널링된다. 추가적으로, 인코더 측에서의 양자화 파라미터들의 공간적 적응은 인코딩 최적화의 복잡도를 증가시킬 수도 있고, 레이트 제어 알고리즘들과 간섭할 수도 있다.

[0131] 또 다른 이러한 제안은 「"Intensity dependent spatial quantization with application in HEVC," by Matteo Naccari and Marta Mrak, In Proc. of IEEE ICME 2013, July 2013 (이하 "Naccari")」이다. Naccari는 강도 종속적 공간적 양자화(Intensity Dependent Spatial Quantization; IDSQ) 지각적 메커니즘을 제안하였고, 이는 인간 시각 시스템의 강도 마스킹을 이용하고, 블록 레벨에서 신호의 양자화를 지각적으로 조정한다. 이 논문은 인-루프 핵심 도메인 스케일링을 채용하는 것을 제안하였다. 이 제안에 따르면, 현재-프로세싱되는 블록에 대한 인-루프 스케일링의 파라미터들은 예측된 블록에서의 루마 컴포넌트의 평균 값들로부터 도출된다. 디코더 측에서, 역 스케일링이 수행되고, 디코더는 디코더 측에서 이용가능한 예측 블록으로부터 스케일링의 파라미터들을 도출한다.

[0132] 상기 논의된 Zhao I에서의 작업과 유사하게, 이 접근법의 블록-기반 입도는 프로세싱되는 블록의 모든 샘플들에 적용되는 스케일링 파라미터의 차선성으로 인해 이 방법의 퍼포먼스(performance)를 제한한다. 이 논문의 제안된 솔루션의 또 다른 양태는, 스케일 값이 예측된 블록으로부터 도출되고, 현재의 코덱 블록과 예측된 블록 사이에 일어날 수도 있는 신호 변동을 반영하지 않는다는 것이다.

[0133] 또 다른 이러한 제안은 「"De-quantization and scaling for next generation containers," by J. Zhao, A. Segal, S.-H. Kim, K. Misra, JVET document B0054, Jan 2016 (이하 Zhao II)」이다. ST 2084/BT2020 표현에서의 비-균일 지각된 왜곡을 향상시키기 위해, 이 논문은 인-루프 강도 종속적 블록 기반 변환 도메인 스케일링을 채용하는 것을 제안하였다. 이 제안에 따르면, 현재 프로세싱되는 블록의 선택된 변환 계수들(AC 계수들)에 대한 인-루프 스케일링의 파라미터들은 예측된 블록에서의 루마 컴포넌트 및 현재 블록에 대해 도출되는 DC 값의 평균 값들의 함수로서 도출된다. 디코더 측에서, 역 스케일링이 수행되고, 디코더는 디코더 측에서 이용가능한 예측된 블록으로부터 그리고 디코더에 시그널링되는 양자화된 DC 값으로부터 AC 계수 스케일링의 파라미터들을 도출한다.

[0134] 상기 논의된 Zhao I 및 Naccari에서의 작업들과 유사하게, 이 접근법의 블록-기반 입도는 프로세싱되는 블록의 모든 샘플들에 적용되는 스케일링 파라미터의 차선성으로 인해 이 방법의 퍼포먼스를 제한한다. 이 논문의 제안된 방식의 또 다른 양태는, 스케일 값이 오직 AC 변환 계수들에만 적용되고, 그를 위해, 신호-대-잡음 비 향상이 DC 값에 영향을 미치지 않는다는 것이고, 이는 그 방식의 퍼포먼스를 감소시킨다. 상기 논의된 양태들에 추가하여, 일부 비디오 코딩 시스템 설계들에서, 양자화된 DC 값은, 양자화 프로세스가 변환 연산들의 캐스케이드를 따르는 경우에서와 같이, AC 값들 스케일링 시에 이용가능하지 않을 수도 있다. 이 제안의 또 다른 제한은, 인코더가 현재 블록에 대해 변환 스kip(skip) 또는 변환/양자화 바이패스 모드들을 선택할 때, 스케일링이 적용되지 않고(그리하여, 디코더에서, 스케일링이 변환 스kip 및 변환/양자화 바이패스 모드들에 대해

정의되지 않고), 이는 이들 2 가지 모드들에 대한 잠재적 코딩 이득의 배제로 인해 차선적이다.

[0135] Dmytro Rusanovskyy 등에 의한 (가 특허 출원 제 62/337,303 호에 대해 우선권을 주장하는) 미국 특허 출원 제 15/595,793 호 (이하 "Rusanovskyy II") 에서, 불균일하게 분포된 변별역 (JND) 을 갖는 비디오 신호들에 대한 인-루프 샘플 프로세싱이 개시된다. Rusanovskyy II 의 기법들에 따르면, 몇가지 인-루프 코딩은 불균일하게 분포된 변별역 (JND) 을 갖는 신호들의 보다 효율적인 코딩에 대해 접근한다. Rusanovskyy II 는 광센, 잔차 또는 변환 도메인 중 어느 일방에서 표현된 신호 샘플들의 스케일 및 오프셋의 적용을 기술한다. 스케일 및 오프셋의 도출을 위한 몇가지 알고리즘들이 제안되었다. Rusanovskyy II 의 콘텐츠는 그 전체가 본원에 참조에 의해 통합된다.

[0136] 이 개시물은 비디오 코딩 시스템의 루프에서 적용될 수 있는 프로세싱을 위한 몇가지 디바이스들, 컴포넌트들, 장치들, 및 방법들을 논의한다. 이 개시물의 기법들은 프로세싱된 데이터에 대한 신호-대-양자화잡음 비율을 향상시키기 위해 광센 도메인에서 또는 변환 도메인에서 비디오 신호의 양자화 및/또는 스케일링의 프로세스들을 포함할 수도 있다. 실제로, 본 개시의 시스템들 및 기법들은 HDR-WCG 포맷으로 변환될 때 SDR-SCG 포맷에서 캡처된 비디오 데이터의 변환에 의해 야기되는 아티팩트들을 감소시킬 수도 있다. 본원에서 설명된 기법들은 루미넌스 및/또는 크로미넌스 데이터의 일방 또는 양방을 이용하여 정밀도를 해결할 수도 있다. 개시된 시스템들 및 기법들은 또한, 신호의 공간-시간적 이웃으로부터 양자화 또는 스케일링 파라미터들의 도출을 위한 몇가지 알고리즘들을 통합 또는 포함한다. 즉, 본 개시의 예시적인 시스템들 및 기법들은 코딩 프로세스에서 현재 블록과 연관된 잔차 데이터를 수정하기 위해 사용되는 하나 이상의 파라미터 값들을 획득하는 것에 지향된다. 본 명세서에서 사용된 바와 같이, 잔차 데이터를 수정하기 위해 사용되는 파라미터 값은 (인코딩 프로세스 또는 디코딩 프로세스에서 각각 잔차 데이터를 양자화 또는 역양자화함으로써 잔차 데이터를 수정하기 위해 사용되는) 양자화 파라미터, 또는 (인코딩 프로세스 또는 디코딩 프로세스에서 각각 잔차 데이터를 스케일링 또는 역-스케일링함으로써 잔차 데이터를 수정하기 위해 사용되는) 스케일링 파라미터를 포함할 수도 있다.

[0137] 도 8 은 현재-코딩되는 블록 (152) 의 공간-시간적 이웃의 양태들을 나타내는 개념도이다. 본 개시의 하나 이상의 기법들에 따르면, 비디오 인코더 (20) 는 현재-코딩되는 블록 (152) 의 공간-시간적 이웃으로부터의 정보를 이용하여 (현재-코딩되는 블록 (152) 의 샘플들의 양자화에서 사용될) 양자화 파라미터를 도출할 수도 있다. 실제로, 비디오 인코더 (20) 는 이웃 블록들 (154, 156, 및 158) 의 하나 이상에 대해 사용되는 QP 값들을 이용하여 현재-코딩되는 블록 (152) 에 있어서의 사용을 위해 레퍼런스 QP 또는 디폴트 QP 를 도출할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 현재-코딩되는 블록 (152) 에 대해 엘타 QP 도출 프로세스에서의 기준들 또는 피연산자들로서 이웃 블록들 (154-158) 의 하나 이상에 대해 QP 값들을 이용할 수도 있다. 이러한 방식으로, 비디오 인코더 (20) 는, 디스패리티 벡터 "DV" 에 의해 포인팅되는, 좌측 이웃 블록 (156) 의 샘플들, 상부 이웃 블록 (158) 의 샘플들, 및 시간 이웃 블록 (154) 의 샘플들을 고려하기 위해 본 개시의 하나 이상의 기법들을 구현할 수도 있다.

[0138] 이와 같이, 비디오 인코더 (20) 는, 비디오 인코더 (20) 가 공간-시간적 이웃 블록들의 샘플들이 현재-코딩되는 블록 (152) 의 샘플들에 대해 잘 매칭된다고 결정하는 경우에, 공간-시간적 이웃의 다양한 이웃하는 블록들에 적어도 부분적으로 기초하여 엘타 QP 도출 프로세스를 행하기 위해 현재-코딩되는 블록 (152) 에 대한 엘타 QP 도출 프로세스를 확장하기 위해 본 개시의 기법들을 구현할 수도 있다. 레퍼런스 샘플들의 블록이 블록 패티셔닝의 다수의 CU 들과 중첩되고 따라서 상이한 QP 를 가질 수도 있는 경우들에서, 비디오 인코더 (20) 는 다수의 이용가능한 QP 들로부터 QP 를 도출할 수도 있다. 실제로, 비디오 인코더 (20) 는 현재-코딩되는 블록 (152) 의 샘플들에 대해 QP 값을 도출하기 위해, 다수의 QP 값들에 대해 평균화의 프로세스를 구현할 수도 있다. 다양한 예들에서, 비디오 인코더 (20) 는 QP 값 및/또는 엘타 QP 파라미터들의 일방 또는 양방을 도출하기 위해 상술된 도출 기법들을 구현할 수도 있다.

[0139] 다양한 사용-경우 시나리오들에서, 비디오 인코더 (20) 는 또한, 현재-코딩되는 블록 (152) 의 공간-시간적 이웃으로부터의 정보를 이용하여 현재-코딩되는 블록 (152) 의 샘플들에 대한 스케일링 파라미터들을 도출할 수도 있다. 예를 들어, 스케일링 연산이 균일한 양자화를 대체하는 설계들에 따르면, 비디오 인코더 (20) 는 현재-코딩되는 블록 (152) 에 대한 레퍼런스 스케일링 파라미터들 또는 디폴트 스케일링 파라미터들을 도출하기 위해 상술된 공간-시간적 이웃 기반 도출 프로세스를 적용할 수도 있다.

[0140] 일부 기준의 HEVC/JEM 기법들에 따르면, 비디오 코딩 디바이스는 현재-프로세싱되는 블록의 모든 변환 계수들에 스케일링 연산들을 적용할 수도 있다. 실제로, 일부 HEVC/JEM 설계들에서, 비디오 코딩 디바이스는 변환 계

수들의 서브-셋트에 하나 이상의 스케일링 파라미터들을 적용할 수도 있는 한편, 스케일링 파라미터(들)의 도출을 위해 나머지 변환 계수들을 이용할 수도 있다. 실제로, JVET B0054 에 따르면, 비디오 코딩 디바이스는, 예측된 블록에서의 루마 컴포넌트의 평균 값들의 함수로서 현재-프로세싱되는 블록의 선택된 변환 계수들 (즉, AC 계수들)에 대한 인-루프 스케일링 파라미터들을 도출할 수도 있고, 현재 블록에 대한 DC 값을 도출할 수도 있다.

[0141] 본 개시의 하나 이상의 기법들에 따르면, 비디오 인코더 (20) 는 현재-코딩되는 블록 (152)에 대한 스케일링 프로세스에서 하나 이상의 DC 변환 계수들을 포함할 수도 있다. 일부 예들에서, 비디오 인코더 (20) 는 예측된 샘플들로부터 도출되는 파라미터들 및 DC 값의 함수로서 현재-코딩되는 블록 (152)에 대한 스케일링 파라미터들을 도출할 수도 있다. 비디오 인코더 (20) 는 AC 스케일링을 위한 룩-업 테이블 (look-up table; LUT), 및 DC 값(들)에 대한 독립적 LUT 를 포함하는 스케일링 파라미터 도출 프로세스를 구현할 수도 있다. DC 및 AC 변환 계수들의 포워드 스케일링은 DC' 및 AC' 로서 표기된 스케일링된 값들을 초래한다. 비디오 인코더 (20) 는 스케일링된 값들 DC' 및 AC' 을 획득하기 위해 아래에 기술된 바와 같은 스케일링 연산들을 구현할 수도 있다:

[0142] $AC' = \text{scale}(\text{fun1}(DC, \text{avgPred})) * AC;$ 및

[0143] $DC' = \text{scale}(\text{fun2}(DC, \text{avgPred})) * DC$

[0144] 본 개시의 스케일링 파라미터 기반 기법들에 따르면, 비디오 디코더 (30) 는 비디오 인코더 (20)에 대해 상기 논의된 것들에 대해 일반적으로 상호역의 연산들을 구현할 수도 있다. 실제로, 비디오 디코더 (30) 는 스케일링된 값들 DC' 및 AC' 을 피연산자들로서 이용하는 역 스케일링 프로세스를 구현할 수도 있다. 역 스케일링 프로세스의 결과들은 아래의 식들에서 DC'' 및 AC'' 으로서 표시된다. 비디오 디코더 (30) 는 다음 식들에서 나타낸 바와 같이 역 스케일링 연산들을 구현할 수도 있다:

[0145] $DC'' = DC' / \text{scale}(\text{fun1}(DC, \text{avgPred}));$ 및

[0146] $AC'' = AC' / \text{scale}(\text{fun2}(DC'', \text{avgPred}))$

[0147] 스케일링 및 역 스케일링 연산들 양자에 대해, 용어들 'fun1' 및 'fun2' 은 레퍼런스 샘플들 및 DC-기반 값들의 평균을 인수들로서 이용하는 스케일 도출 함수들/프로세스들을 정의한다. 비디오 인코더 (20) 및 비디오 디코더 (30)에 의해 구현되는 스케일링 및 역 스케일링 기법들 양자에 대해 예시된 바와 같이, 본 개시의 기법들은 스케일링된 및 역-스케일링된 DC 및 AC 변환 계수 값들 양자의 도출에서 DC 변환 계수 값들의 사용을 가능하게 한다. 이러한 방식으로, 본 개시의 기법들은 비디오 인코더 (20) 및 비디오 디코더 (30) 가, 스케일링/역-스케일링 연산들이 변환 계수들의 양자화 및 역양자화를 대신하여 수행되는 경우에, 스케일링 및 역-스케일링 연산들에서 DC 변환 계수 값들을 래버리징하는 것을 가능하게 한다.

[0148] 본 개시는 또한, 비디오 인코더 (20) 가 임의의 비-제로 변환 계수들을 시그널링하지 않는 경우들에서 양자화 파라미터들 또는 스케일링 파라미터들의 도출을 위한 기법들을 제공한다. HEVC 의 현재 사양, JVET 개발의 초기 테스트 모델, 및 JVET B0054 에서 기술된 설계는 존재하는 인코딩된 비-제로 변환 계수들의 함수로서 QP 값들 (또는 경우일 수도 있는 스케일링 파라미터들)의 도출을 명시한다. 모든 변환 계수들이 제로로 양자화되는 경우에, 비 QP 조정 또는 로컬로 적용되는 스케일은 HEVC 의 현재 사양, JVET 의 초기 테스트된 모델, 및 JVET B0054 의 설계에 따라서 시그널링된다. 대신에, 디코딩 디바이스는, 변환 계수들에, 글로벌 (예컨대, 슬라이스 레벨) QP/스케일링 파라미터, 또는 공간적 이웃하는 CU 들로부터 도출되는 QP 중 어느 일방을 변환 계수들에 적용한다.

[0149] 본 개시의 기법들은 비-제로 변환 계수들의 부존재를 초래하는 예측 (인트라이든지 또는 인터이든지 간에)의 상대적 정확도를 래버리징한다. 실제로, 비디오 디코더 (30) 는 QP 값들 또는 스케일링 파라미터들을 도출하기 위해 예측된 샘플들로부터의 파라미터들을 이용하기 위해 본 개시의 기법들을 구현할 수도 있다. 다음으로, 비디오 디코더 (30) 는 현재 블록의 샘플들을 역양자화하기 위해 또는 현재 블록의 변환 계수들을 역-스케일링하기 위해, 도출된 QP 값들 또는 스케일링 파라미터들을 이용할 수도 있다. 이러한 방식으로, 비디오 디코더 (30) 는, 비디오 디코더 (30) 가 블록에 대해 아무런 비-제로 변환 계수들도 수신하지 않고, 이에 의해, HEVC/JEM 프랙티스들의 하나 이상의 디폴트-기반 역양자화 및 역-스케일링 양태들을 대체하는 시나리오들에서 예측 정확도를 래버리징하기 위해 본 개시의 기법들을 구현할 수도 있다.

[0150] 개시된 기법들의 다양한 예시적인 구현들이 이하에서 설명된다. 이하에서 설명되는 구현들은 비제한적인 예

들이고, 개시된 기법들의 다른 구현들이 또한 이 개시의 양태들에 따라서 가능하다는 것이 이해될 것이다.

[0151] 일부 구현들에 따르면, 비디오 인코더 (20) 는 부착된 (상부 및 좌측) 블록들 (CU들)로부터 레퍼런스 QP 값을 도출할 수도 있다. 도 8 과 관련하여 설명하면, 비디오 인코더 (20) 는 상부 이웃 블록 (158) 및 좌측 이웃 블록 (156) 과 연관된 데이터로부터 현재-코딩되는 블록 (152) 을 위한 레퍼런스 QP 를 도출할 수도 있다. 이 예시적인 구현의 일 예는 아래의 의사코드에 의해 기술된다:

```
[0152] Char TComDataCU::getRefQP( UInt uiCurrAbsIdxInCtu )
[0153] {
[0154]     TComDataCU* cULeft = getQpMinCuLeft( lPartIdx, m_absZIdxInCtu + uiCurrAbsIdxInCtu );
[0155]     TComDataCU* cUAbove = getQpMinCuAbove( aPartIdx, m_absZIdxInCtu + uiCurrAbsIdxInCtu );
[0156]     return ((cULeft? cULeft->getQP( lPartIdx ): m_QuLastCodedQP) + (cUAbove? cUAbove->getQP( aPartIdx ): m_QuLastCodedQP) + 1) >> 1);
[0157] }
```

[0158] 상기 의사코드에서, 부착된 블록들은 심볼들 "cUAbove" 및 "cULeft" 에 의해 표현된다.

[0159] 본 개시의 기법들의 일부 구현들에 따르면, 비디오 인코더 (20) 는 레퍼런스 샘플(들)의 하나 이상의 QP 값을 QP 도출 프로세스에서의 고려사항으로 취할 수도 있다. 이러한 구현의 일 예는 아래의 의사코드에 의해 기술된다:

```
[0160] Char TComDataCU::getRefQP2( UInt uiCurrAbsIdxInCtu )
[0161] {
[0162]     TComDataCU* cULeft = getQpMinCuLeft( lPartIdx, m_absZIdxInCtu + uiCurrAbsIdxInCtu );
[0163]     TComDataCU* cUAbove = getQpMinCuAbove( aPartIdx, m_absZIdxInCtu + uiCurrAbsIdxInCtu );
[0164]     TComDataCU* cURefer = getQpMinCuReference( aPartIdx, m_absZIdxInCtu + uiCurrAbsIdxInCtu );
[0165]     return value = function (cULeft->getLastQP(), cUAbove->getLastQP(), cURefer ->getLastQP());
[0166] }
```

[0167] 상기 의사코드에서, 심볼 "cURefer" 은 레퍼런스 샘플들을 포함하는 블록을 나타낸다.

[0168] 설명된 기법들의 일부 구현들에 따르면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 레퍼런스 픽처들로서 이용되는 모든 픽처들에 대해 레퍼런스 블록(들)의 샘플들에 대해 적용되는 QP 들 및/또는 글로벌 QP 들 (예컨대, 슬라이스-레벨 QP 들) 을 저장할 수도 있다. 일부 구현들에 따르면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 레퍼런스 픽처들로서 이용되는 모든 픽처들에 대해 레퍼런스 블록(들)의 샘플들에 대해 적용되는 스케일링 파라미터들 및/또는 글로벌 스케일링 (예컨대, 슬라이스-레벨 스케일링) 파라미터들을 저장할 수도 있다. 레퍼런스 샘플들의 블록이 파티셔닝된 블록의 다수의 CU 들과 중첩하는 (그리고 파티션들에 걸쳐 상이한 QP 들의 가능성을 도입하는) 경우에, 비디오 인코더 (20) 는 다수의 이용가능한 QP 들로부터 QP 를 도출할 수도 있다. 일 예로서, 비디오 인코더 (20) 는 다수의 CU 들로부터 다수의 QP 들에 대한 평균화 프로세스를 구현할 수도 있다. 이러한 구현의 일 예는 아래 의사코드에 의해 기술된다:

```
[0169] Int sum= 0;
[0170] for (Int i=0; i < numMinPart; i++)
[0171] {
[0172]     sum += m_phInferQP[COMPONENT_Y][uiAbsPartIdxInCTU + i];
[0173] }
[0174] avgQP = (sum)/numMinPart;
```

[0175] 상기 의사코드에 따르면, 비디오 인코더 (20) 는 블록 파티션들에 걸쳐 QP 들의 평균 값을 계산함으로써 평균화

프로세싱을 수행한다. 평균 QP 계산은 상기 의사코드에서의 마지막 연산에서 나타난다. 즉, 비디오 인코더 (20)는 (피연산자 "numMinPart"에 의해 표현되는) 파티션들의 카운트에 의해 나누어진 (정수 "sum"의 최종 값에 의해 표현되는) 집합을 분할한다.

[0176] 본원에 기술된 기법들의 또 다른 구현에서, 비디오 인코더 (20)는 루마 컴포넌트들의 평균 밝기의 함수로서 QP를 도출할 수도 있다. 실제로, 비디오 인코더 (20)는 룩업 테이블 (LUT)로부터 루마 컴포넌트들의 평균 밝기를 획득할 수도 있다. 이 구현은 다음과 같은 의사코드에 의해 기술되고, 여기서, 심볼 "avgPred"는 레퍼런스 샘플들의 평균 밝기 값을 나타낸다:

[0177] $QP = PQ_LUT[\text{avgPred}];$

[0178] 일부 구현들에서, 비디오 인코더 (20)는 하나 이상의 글로벌 QP 값들로부터 현재 블록에 대한 레퍼런스 QP 값을 도출할 수도 있다. 비디오 인코더 (20)가 이용하는 글로벌 QP 값의 일 예는 슬라이스 레벨에서 명시된 QP이다. 즉, 비디오 인코더 (20)는 현재 블록을 포함하는 슬라이스의 전체에 대해 명시된 QP 값을 이용하여 현재 블록에 대한 QP 값을 도출할 수도 있다. 이 구현은 다음과 같은 의사코드에 의해 기술된다:

[0179] $qp = (((\text{Int}) \text{ pcCU->getSlice()}->\text{getSliceQp}() + iDQp + 52 + 2*\text{qpBdOffsetY}) \% (52 + \text{qpBdOffsetY})) - \text{qpBdOffsetY};$

[0180] 상기 의사코드에서, 비디오 인코더 (20)는 ("qp"에 의해 표시된) 현재 블록에 대한 QP를 획득하기 위해 연산에서의 피연산자로서 `getSliceQp()` 함수에 의해 리턴되는 값을 사용한다.

[0181] 본원에서 기술된 기법들의 일부 구현들에서, 비디오 인코더 (20)는 QP들을 도출함에 있어서 하나 이상의 레퍼런스 샘플 값을 이용할 수도 있다. 이러한 구현은 다음과 같은 의사코드에 의해 기술된다:

[0182] $QP = PQ_LUT[\text{avgPred}];$

[0183] 상기 의사코드에서, "PQ_LUT"는 비디오 인코더 (20)가 ("avgPred"에 의해 표현되는) 예측된 블록의 평균 밝기 값을 연관된 지각적 양자화기 (PQ) 값에 맵핑하기 위해 이용할 수도 있다. 비디오 인코더 (20)는 레퍼런스 샘플들의 평균 값들과 같이, 레퍼런스 샘플들의 함수로서 avgPred의 값을 계산할 수도 있다. 본 개시의 계산들에 따라 사용될 수 있는 평균 값들의 예들은 평균, 중간, 및 모드 값을 중 하나 이상을 포함한다.

[0184] 일부 구현들에서, 비디오 인코더 (20)는 QP를 대신에 현재 블록에 대한 스케일링 파라미터들을 도출할 수도 있다. 비디오 인코더 (20)는 도출된 QP(들)로부터 스케일 파라미터(들)로, 또는 그 역으로 변환 프로세스를 수행할 수도 있다. 일부 구현들에서, 비디오 인코더 (20)는 레퍼런스 샘플들로부터 QP를 도출하기 위해 분석적 표현을 이용할 수도 있다. 비디오 인코더 (20)가 QP 도출을 위해 이용할 수도 있는 분석적 표현의 하나의 예는 파라미터적 도출 모델이다.

[0185] 비디오 인코더 (20)가 현재 블록에 대해 QP를 도출하는 것이 상술된 기법들 중 무엇이든지간에 상관 없이, 비디오 인코더 (20)는 도출된 QP에 기초한 데이터를 비디오 디코더 (30)에 시그널링할 수도 있다. 실제로, 비디오 인코더 (20)는, 비디오 인코더 (20)가 샘플들 현재 블록을 양자화하기 위해 사용한 QP 값으로부터 도출되는 멜타 QP 값을 시그널링할 수도 있다. 다음으로, 비디오 디코더 (30)는 블록에 대해 QP 값을 획득하기 위해 인코딩된 비디오 비트스트림에서 수신된 멜타 QP 값을 사용할 수도 있고, 그 QP 값을 이용하여 블록의 샘플들을 역양자화할 수도 있다.

[0186] 비디오 인코더 (20)가 현재 블록에 대한 QP 값에 추가하여 또는 그 대신에 스케일링 파라미터들을 획득하는 예들에서, 비디오 인코더 (20)는 그 스케일링 파라미터들 (또는 거기에서 도출되는 데이터)을 비디오 디코더 (30)에 시그널링할 수도 있다. 다음으로, 비디오 디코더 (30)는 인코딩된 비디오 비트스트림으로부터, 시그널링된 데이터로부터 파라미터들을 도출함으로써 또는 아니면 직접적으로, 스케일링 파라미터들을 재구성할 수도 있다. 비디오 디코더 (30)는 스케일링된 변환 계수들의 역 스케일링을 수행할 수도 있다. 실제로, 비디오 디코더 (30)는, 본 개시의 양태들에 따라서, DC 및 AC 양방의 변환 계수들의 스케일링된 변환들의 역 스케일링을 수행할 수도 있다.

[0187] 다양한 예들 (예컨대, 구현들)이 상기 설명되었다. 이 개시의 예들은 별개로 또는 다른 예들 중 하나 이상과의 다양한 결합들로 사용될 수도 있다.

[0188] 도 9는 본 개시의 기법들을 구현할 수도 있는 비디오 인코더 (20)의 일 예를 나타내는 블록도이다. 비디오 인코더 (20)는 비디오 슬라이스들 내에서 비디오 블록들의 인트라-코딩 및 인터-코딩을 수행할 수도 있다.

인트라-코딩은 소정의 비디오 프레임 또는 픽처 내 비디오에 있어서 공간 리던던시를 감소 또는 제거하기 위해 공간 예측에 의존한다. 인터-코딩은 비디오 시퀀스의 인접한 프레임들 또는 픽처들 내 비디오에 있어서 시간 리던던시를 감소 또는 제거하기 위해 시간 예측에 의존한다. 인트라-모드 (I 모드) 는 수개의 공간 기반 코딩 모드들 중 임의의 모드를 지칭할 수도 있다. 단방향 예측 (P 모드) 또는 양방향 예측 (B 모드) 과 같은 인터-모드들은 수개의 시간 기반 코딩 모드들 중 임의의 모드를 지칭할 수도 있다.

[0189] 도 9 에 도시된 바와 같이, 비디오 인코더 (20) 는 인코딩될 비디오 프레임 내의 현재 비디오 블록을 수신한다.

도 9 의 예에 있어서, 비디오 인코더 (20) 는 모드 선택 유닛 (40), 비디오 데이터 메모리 (41), 디코딩된 픽처 버퍼 (64), 합산기 (50), 변환 프로세싱 유닛 (52), 양자화 유닛 (54), 및 엔트로피 인코딩 유닛 (56) 을 포함한다. 모드 선택 유닛 (40) 은 차례로, 모션 보상 유닛 (44), 모션 추정 유닛 (42), 인트라-예측 프로세싱 유닛 (46), 및 파티션 유닛 (48) 을 포함한다. 비디오 블록 재구성을 위해, 비디오 인코더 (20) 는 또한 역양자화 유닛 (58), 역변환 프로세싱 유닛 (60), 및 합산기 (62) 를 포함한다. 디블로킹 (deblocking) 필터 (도 9 에 도시 안됨) 가 또한, 블록 경계들을 필터링하여 재구성된 비디오로부터 블록 키니스 아티팩트들을 제거하기 위해 포함될 수도 있다. 요구된다면, 디블로킹 필터는 통상적으로 합산기 (62) 의 출력을 필터링 할 것이다. 부가적인 필터 (예컨대, 인 루프 (in loop) 또는 포스트 루프 (post loop)) 가 또한 디블로킹 필터에 부가하여 이용될 수도 있다. 그러한 필터들은 간략화를 위해 도시되지 않지만, 요구된다면, (인-루프 필터로서) 합산기 (50) 의 출력을 필터링할 수도 있다.

[0190] 비디오 데이터 메모리 (41) 는, 비디오 인코더 (20) 의 컴포넌트들에 의해 인코딩될 비디오 데이터를 저장할 수도 있다. 비디오 데이터 메모리 (41) 에 저장된 비디오 데이터는, 예를 들어, 비디오 소스 (18) 로부터 획득될 수도 있다. 디코딩된 픽처 버퍼 (64) 는, 예를 들어, 인트라- 또는 인터-코딩 모드들에서 비디오 인코더 (20) 에 의해 비디오 데이터를 인코딩함에 있어서 사용하기 위한 레퍼런스 비디오 데이터를 저장하는 레퍼런스 픽처 메모리일 수도 있다. 비디오 데이터 메모리 (41) 및 디코딩된 픽처 버퍼 (64) 는 동기식 DRAM (SDRAM) 을 포함한 동적 랜덤 액세스 메모리 (DRAM), 자기저항성 RAM (MRAM), 저항성 RAM (RRAM), 또는 다른 타입들의 메모리 디바이스들과 같은 다양한 메모리 디바이스들 중 임의의 메모리 디바이스에 의해 형성될 수도 있다. 비디오 데이터 메모리 (41) 및 디코딩된 픽처 버퍼 (64) 는 동일한 메모리 디바이스 또는 별도의 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에 있어서, 비디오 데이터 메모리 (41) 는 비디오 인코더 (20) 의 다른 컴포넌트들과 온-칩형이거나 그 컴포넌트들에 대하여 오프-칩형일 수도 있다.

[0191] 인코딩 프로세스 동안, 비디오 인코더 (20) 는 코딩될 비디오 프레임 또는 슬라이스를 수신한다. 그 프레임 또는 슬라이스는 다중의 비디오 블록들로 분할될 수도 있다. 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 은 시간 예측을 제공하기 위해 하나 이상의 레퍼런스 프레임들에 있어서의 하나 이상의 블록들에 대해 수신된 비디오 블록의 인터-예측 코딩을 수행한다. 대안적으로, 인트라-예측 프로세싱 유닛 (46) 은 공간 예측을 제공하기 위해 코딩될 블록과 동일한 프레임 또는 슬라이스에 있어서의 하나 이상의 이웃한 블록들에 대해 수신된 비디오 블록의 인트라-예측 코딩을 수행할 수도 있다. 비디오 인코더 (20) 는, 예를 들어, 비디오 데이터의 각각의 블록에 대한 적당한 코딩 모드를 선택하기 위해 다중의 코딩 패스들을 수행할 수도 있다.

[0192] 더욱이, 파티션 유닛 (48) 은 이전 코딩 패스들에 있어서의 이전 파티셔닝 방식들의 평가에 기초하여 비디오 데이터의 블록들을 서브-블록들로 파티셔닝할 수도 있다. 예를 들어, 파티션 유닛 (48) 은, 레이트-왜곡 분석 (예를 들어, 레이트-왜곡 최적화) 에 기초하여, 처음에 프레임 또는 슬라이스를 LCU들로 파티셔닝하고, LCU들 각각을 서브-CU들로 파티셔닝할 수도 있다. 모드 선택 유닛 (40) 은 추가로, LCU 의 서브-CU들로의 파티셔닝을 표시하는 큐드트리 데이터 구조를 생성할 수도 있다. 큐드트리의 리프 노드 CU들은 하나 이상의 PU들 및 하나 이상의 TU들을 포함할 수도 있다.

[0193] 모드 선택 유닛 (40) 은, 예를 들어, 여러 결과들에 기초하여 코딩 모드들 중 하나, 즉, 인트라 모드 또는 인터 모드를 선택할 수도 있고, 결과적인 인트라- 또는 인터-코딩된 블록을 합산기 (50) 에 제공하여 잔차 블록 데이터를 생성하고, 합산기 (62) 에 제공하여 레퍼런스 프레임으로서의 사용을 위한 인코딩된 블록을 재구성한다.

모드 선택 유닛 (40) 은 또한, 모션 벡터들, 인트라 모드 표시자들, 파티션 정보, 및 다른 그러한 선택스 정보와 같은 선택스 엘리먼트들을 엔트로피 인코딩 유닛 (56) 에 제공한다.

[0194] 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 은 고도로 통합될 수도 있지만, 개념적인 목적을 위해 별도로 도시된다. 모션 추정 유닛 (42) 에 의해 수행된 모션 추정은, 비디오 블록들에 대한 모션을 추정하는 모션 벡터들을 생성하는 프로세스이다. 모션 벡터는, 예를 들어, 현재 픽처 (또는 다른 코딩된 유닛) 내에서 코딩 되는 현재 블록에 대한 레퍼런스 픽처 (또는 다른 코딩된 유닛) 내 예측성 블록에 대한 현재 비디오 프레임 또

는 꽉쳐 내 비디오 블록의 PU 의 변위를 표시할 수도 있다. 예측성 블록은 픽셀 차이의 관점에서 코딩될 블록과 밀접하게 매칭되도록 발견되는 블록이며, 이 픽셀 차이는 절대 차이의 합 (SAD), 제곱 차이의 합 (SSD), 또는 다른 상이한 메트릭들에 의해 결정될 수도 있다. 일부 예들에 있어서, 비디오 인코더 (20) 는 디코딩된 꽉쳐 베파 (64) 에 저장된 레퍼런스 꽉쳐들의 서브-정수 픽셀 포지션들에 대한 값들을 계산할 수도 있다.

예를 들어, 비디오 인코더 (20) 는 레퍼런스 꽉쳐의 1/4 픽셀 포지션들, 1/8 픽셀 포지션들, 또는 다른 분수 픽셀 포지션들의 값들을 보간할 수도 있다. 따라서, 모션 추정 유닛 (42) 은 풀 픽셀 포지션들 및 분수 픽셀 포지션들에 대한 모션 탐색을 수행하고, 분수 픽셀 정밀도로 모션 벡터를 출력할 수도 있다.

[0195]

모션 추정 유닛 (42) 은 인터-코딩된 슬라이스에 있어서의 비디오 블록의 PU 에 대한 모션 벡터를, 그 PU 의 포지션을 레퍼런스 꽉쳐의 예측 블록의 포지션과 비교함으로써 계산한다. 레퍼런스 꽉쳐는 제 1 레퍼런스 꽉쳐 리스트 (리스트 0) 또는 제 2 레퍼런스 꽉쳐 리스트 (리스트 1) 로부터 선택될 수도 있으며, 이 리스트들 각각은 디코딩된 꽉쳐 베파 (64) 에 저장된 하나 이상의 레퍼런스 꽉쳐들을 식별한다. 모션 추정 유닛 (42) 은 계산된 모션 벡터를 엔트로피 인코딩 유닛 (56) 및 모션 보상 유닛 (44) 으로 전송한다.

[0196]

모션 보상 유닛 (44) 에 의해 수행된 모션 보상은 모션 추정 유닛 (42) 에 의해 결정된 모션 벡터에 기초하여 예측 블록을 폐치 또는 생성하는 것을 수반할 수도 있다. 다시, 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 은 일부 예들에 있어서 기능적으로 통합될 수도 있다. 현재 비디오 블록의 PU 에 대한 모션 벡터를 수신할 시, 모션 보상 유닛 (44) 은, 모션 벡터가 레퍼런스 꽉쳐 리스트들 중 하나에 포인팅하는 예측 블록을 로케이팅할 수도 있다. 합산기 (50) 는, 하기에서 논의되는 바와 같이, 코딩되는 현재 비디오 블록의 픽셀 값들로부터 예측 블록의 픽셀 값들을 감산하여 픽셀 차이 값들을 형성함으로써 잔차 비디오 블록을 형성한다. 일반적으로, 모션 추정 유닛 (42) 은 루마 컴포넌트들에 대한 모션 추정을 수행하고, 모션 보상 유닛 (44) 은 크로마 컴포넌트들 및 루마 컴포넌트들 양자에 대한 루마 컴포넌트들에 기초하여 계산된 모션 벡터들을 이용한다. 모드 선택 유닛 (40) 은 또한, 비디오 슬라이스의 비디오 블록들을 디코딩함에 있어서 비디오 디코더 (30) 에 의한 사용을 위해 비디오 블록들 및 비디오 슬라이스와 연관된 신택스 엘리먼트들을 생성할 수도 있다.

[0197]

인트라-예측 프로세싱 유닛 (46) 은 현재 블록을, 상기 설명된 바와 같은 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 에 의해 수행된 인터-예측에 대한 대안으로서 인트라-예측할 수도 있다. 특히, 인트라-예측 프로세싱 유닛 (46) 은 현재 블록을 인코딩하는데 이용하기 위한 인트라-예측 모드를 결정할 수도 있다. 일부 예들에 있어서, 인트라-예측 프로세싱 유닛 (46) 은 예를 들어 별도의 인코딩 패스를 동안에 다양한 인트라-예측 모드들을 이용하여 현재 블록을 인코딩할 수도 있으며, 인트라-예측 프로세싱 유닛 (46) (또는 일부 예들에서는 모드 선택 유닛 (40)) 은 테스팅된 모드들로부터의 이용을 위해 적당한 인트라-예측 모드를 선택할 수도 있다.

[0198]

예를 들어, 인트라-예측 프로세싱 유닛 (46) 은 다양한 테스팅된 인트라-예측 모드들에 대한 레이트-왜곡 분석을 이용하여 레이트-왜곡 값들을 계산하고, 테스팅된 모드들 중 최상의 레이트-왜곡 특성들을 갖는 인트라-예측 모드를 선택할 수도 있다. 레이트-왜곡 분석은 일반적으로, 인코딩된 블록과 그 인코딩된 블록을 생성하도록 인코딩되었던 오리지널의 인코딩되지 않은 블록 간의 왜곡 (또는 에러) 의 양뿐 아니라 인코딩된 블록을 생성하는데 사용된 비트 레이트 (즉, 비트들의 수) 를 결정한다. 인트라-예측 프로세싱 유닛 (46) 은 다양한 인코딩된 블록들에 대한 왜곡들 및 레이트들로부터의 비율들을 계산하여, 어느 인트라-예측 모드가 그 블록에 대한 최상의 레이트-왜곡 값을 나타내는지를 결정할 수도 있다.

[0199]

블록에 대한 인트라-예측 모드를 선택한 이후, 인트라-예측 프로세싱 유닛 (46) 은 블록에 대한 선택된 인트라-예측 모드를 표시하는 정보를 엔트로피 인코딩 유닛 (56) 에 제공할 수도 있다. 엔트로피 인코딩 유닛 (56) 은 선택된 인트라-예측 모드를 표시한 정보를 인코딩할 수도 있다. 비디오 인코더 (20) 는 송신된 비트스트림에, 복수의 인트라-예측 모드 인덱스 테이블들 및 복수의 수정된 인트라-예측 모드 인덱스 테이블들 (코드워드 매핑 테이블들로서도 또한 지칭됨) 을 포함할 수도 있는 구성 데이터, 다양한 블록들에 대한 인코딩 컨텍스트들의 정의들, 및 컨텍스트들 각각에 대한 사용을 위한 가장 가능성있는 인트라-예측 모드, 인트라-예측 모드 인덱스 테이블, 및 수정된 인트라-예측 모드 인덱스 테이블의 표시들을 포함할 수도 있다.

[0200]

비디오 인코더 (20) 는 모드 선택 유닛 (40) 으로부터의 예측 데이터를, 코딩되는 오리지널 비디오 블록으로부터 감산함으로써 잔차 비디오 블록을 형성한다. 합산기 (50) 는 이러한 감산 연산을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다. 변환 프로세싱 유닛 (52) 은 이산 코사인 변환 (DCT) 또는 개념적으로 유사한 변환과 같은 변환을 잔차 블록에 적용하여, 잔차 변환 계수 값들을 포함하는 비디오 블록을 생성한다. 변환 프로세싱 유닛 (52) 은, DCT 와 개념적으로 유사한 다른 변환들을 수행할 수도 있다. 웨이블릿 변환들, 정수 변환들, 서브-대역 변환들, 또는 다른 타입들의 변환들이 또한 이용될 수 있다. 어떠한 경우라도, 변환

프로세싱 유닛 (52) 은 변환을 잔차 블록에 적용하여, 잔차 변환 계수들의 블록을 생성한다. 그 변환은 잔차 정보를 픽셀 값 도메인으로부터 주파수 도메인과 같은 변환 도메인으로 컨버팅할 수도 있다. 변환 프로세싱 유닛 (52) 은 결과적인 변환 계수들을 양자화 유닛 (54) 으로 전송할 수도 있다.

[0201] 양자화 유닛 (54) 은 비트 레이트를 추가로 감소시키기 위해 변환 계수들을 양자화한다. 양자화 프로세스는 그 계수들의 일부 또는 그 모두와 연관되는 비트 심도를 감소시킬 수도 있다. 양자화의 정도는 양자화 파라미터를 조정함으로써 수정될 수도 있다. 그 후, 일부 예들에 있어서, 양자화 유닛 (54) 은, 양자화된 변환 계수들을 포함하는 매트릭스의 스캔을 수행할 수도 있다. 대안적으로, 엔트로피 인코딩 유닛 (56) 은 스캔을 수행할 수도 있다.

[0202] 양자화에 이어서, 엔트로피 인코딩 유닛 (56) 은 양자화된 변환 계수들을 엔트로피 코딩한다. 예를 들어, 엔트로피 인코딩 유닛 (56) 은 컨텍스트 적응 가변 길이 코딩 (CAVLC), 컨텍스트 적응 바이너리 산술 코딩 (CABAC), 십액스 기반 컨텍스트 적응 바이너리 산술 코딩 (SBAC), 확률 간격 파티셔닝 엔트로피 (PIPE) 코딩, 또는 다른 엔트로피 코딩 기법을 수행할 수도 있다. 컨텍스트 기반 엔트로피 코딩의 경우, 컨텍스트는 이웃한 블록들에 기초할 수도 있다. 엔트로피 인코딩 유닛 (56) 에 의한 엔트로피 코딩 이후, 인코딩된 비트스트림은 다른 디바이스 (예를 들어, 비디오 디코더 (30)) 로 송신되거나 또는 더 나중의 송신 또는 취출을 위해 아카이브될 수도 있다.

[0203] 역양자화 유닛 (58) 및 역변환 프로세싱 유닛 (60) 은, 예를 들어, 레퍼런스 블록으로서의 더 나중 사용을 위해 픽셀 도메인에서 잔차 블록을 재구성하도록, 각각, 역양자화 및 역변환을 적용한다. 모션 보상 유닛 (44) 은 디코딩된 픽처 베퍼 (64) 의 프레임들 중 하나의 예측 블록에 잔차 블록을 부가함으로써 레퍼런스 블록을 계산할 수도 있다. 모션 보상 유닛 (44) 은 또한, 하나 이상의 보간 필터들을 재구성된 잔차 블록에 적용하여, 모션 추정에서 사용하기 위한 서브-정수 픽셀 값들을 계산할 수도 있다. 합산기 (62) 는 재구성된 잔차 블록을, 모션 보상 유닛 (44) 에 의해 생성된 모션 보상된 예측 블록에 부가하여, 디코딩된 픽처 베퍼 (64) 로의 저장을 위한 재구성된 비디오 블록을 생성한다. 재구성된 비디오 블록은, 후속 비디오 프레임에서의 블록을 인터-코딩하기 위해 레퍼런스 블록으로서 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 에 의해 사용될 수도 있다.

[0204] 비디오 인코더 (20) 는 블록의 공간-시간적 이웃 블록들로부터 현재-인코딩되는 블록에 대한 양자화 파라미터 (quantization parameter; QP) 값들을 도출하기 위해서, 및/또는 현재-인코딩되는 블록의 모든 (예컨대, DC 및 AC) 변환 계수들에 스케일링 연산들을 적용하기 위해서 본 개시의 다양한 기법들을 구현할 수도 있다.

[0205] 이하의 설명에서 도 8 에 대한 참조가 또한 이루어진다. 일부 구현들에서, 비디오 인코더 (20) 는 공간-시간적 이웃의 부착된 블록들 (CU들) 로부터 현재-코딩되는 블록 (152) 에 대한 레퍼런스 QP 값을 도출할 수도 있다. 즉, 비디오 인코더 (20) 는 상부 이웃 블록 (158) 및 좌측 이웃 블록 (156) 을 이용하여 현재-코딩되는 블록 (152) 에 대한 QP 값을 도출할 수도 있다. 비디오 인코더 (20) 가 상부 이웃 블록 (158) 및 좌측 이웃 블록 (156) 을 이용하여 현재-코딩되는 블록 (152) 에 대한 QP 값을 도출하는 이러한 구현의 일 예는 아래의 의사코드에 의해 기술된다:

```
[0206] Char TComDataCU::getRefQP( UInt uiCurrAbsIdxInCtu )
[0207] {
[0208]     TComDataCU* cULeft = getQpMinCuLeft( lPartIdx, m_absZIdxInCtu + uiCurrAbsIdxInCtu );
[0209]     TComDataCU* cUAbove = getQpMinCuAbove( aPartIdx, m_absZIdxInCtu + uiCurrAbsIdxInCtu );
[0210]     return ((cULeft? cULeft->getQP( lPartIdx ): m_QuLastCodedQP) + (cUAbove? cUAbove->getQP( aPartIdx ): m_QuLastCodedQP) + 1) >> 1);
[0211] }
```

[0212] 일부 구현들에서, 비디오 인코더 (20) 는 레퍼런스 샘플들의 하나 이상의 QP 값들을 고려함으로써 현재-코딩되는 블록 (152) 에 대한 QP 값을 도출할 수도 있다. 비디오 인코더 (20) 가 현재-코딩되는 블록 (152) 에 대한 QP 값을 도출하기 위해 레퍼런스 샘플들의 QP 값(들)을 이용하는 이러한 구현의 일 예는 아래 의사코드에 의해 기술된다:

```
[0213] Char TComDataCU::getRefQP2( UInt uiCurrAbsIdxInCtu )
```

```
[0214] {
[0215] TComDataCU* cULeft = getQpMinCuLeft( lPartIdx, m_absZIdxInCtu + uiCurrAbsIdxInCtu );
[0216] TComDataCU* cUAbove = getQpMinCuAbove( aPartIdx, m_absZIdxInCtu + uiCurrAbsIdxInCtu );
[0217] TComDataCU* cURefer = getQpMinCuReference( aPartIdx, m_absZIdxInCtu + uiCurrAbsIdxInCtu );
[0218] return value = function (cULeft->getLastQP(), cUAbove->getLastQP(), cURefer ->getLastQP());
[0219] }
[0220] 본원에 기술되는 기법들의 일부 구현들에 따르면, 비디오 인코더 (20) 는 레퍼런스 픽처들로서 이용되는 모든 픽처들에 대해 레퍼런스 블록(들)의 샘플들에 대해 적용되는 QP 들 및/또는 글로벌 QP 들 (예컨대, 슬라이스-레벨 QP 들) 을 저장할 수도 있다. 본원에 기술되는 기법들의 일부 구현에 따르면, 비디오 인코더 (20) 는 레퍼런스 픽처들로서 이용되는 모든 픽처들에 대해 레퍼런스 블록(들)의 샘플들에 대해 적용되는 스케일링 파라미터들 및/또는 글로벌 스케일링 (예컨대, 슬라이스-레벨 스케일링) 파라미터들을 저장할 수도 있다. 레퍼런스 샘플들의 블록이 블록 파티셔닝의 다수의 CU 들과 중첩하는 (따라서 가능하게는 파티션들에 걸쳐 상이한 QP 들을 갖는) 경우에, 비디오 인코더 (20) 는 다수의 이용가능한 QP 들로부터 QP 를 도출할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 다수의 이용가능한 QP 들에 대해 평균화하는 프로세스를 구현함으로써 현재-코딩되는 블록 (152) 에 대한 QP 를 도출할 수도 있다. 비디오 인코더 (20) 가 레퍼런스 샘플들로부터 다수의 이용가능한 QP 들을 평균화함으로써 현재-코딩되는 블록 (152) 에 대한 QP 를 도출할 수도 있는 구현의 일 예는 아래의 의사코드에 의해 기술된다:
[0221] Int sum= 0;
[0222] for (Int i=0; i < numMinPart; i++)
[0223] {
[0224] sum += m_phInferQP[COMPONENT_Y][uiAbsPartIdxInCTU + i];
[0225] }
[0226] avgQP = (sum)/numMinPart;
[0227] 본원에서 기술되는 QP-도출 기법들의 또 다른 구현에서, 비디오 인코더 (20) 는 툭업 테이블 (LUT) 등으로부터, 루마 컴포넌트의 평균 밝기의 함수로서 QP 를 도출할 수도 있다. 이러한 구현은 다음과 같은 의사코드에 의해 기술되고, 여기서, 'avgPred' 는 레퍼런스 샘플들의 평균 밝기이다:
[0228] QP = PQ_LUT[avgPred];
[0229] 본원에 기술되는 QP-도출 기법들의 일부 구현들에 따르면, 비디오 인코더 (20) 는 하나 이상의 글로벌 QP 값들로부터 레퍼런스 QP 값을 도출할 수도 있다. 글로벌 QP 값의 일 예는 슬라이스 레벨에서 명시되는 QP 값이다. 이러한 구현은 다음과 같은 의사코드에 의해 기술된다:
[0230] qp = (((Int) pcCU->getSlice()->getSliceQP() + iDQP + 52 + 2*qpBdOffsetY )%(52+ qpBdOffsetY)) - qpBdOffsetY;
[0231] 본원에 기술되는 QP-도출 기법들의 일부 구현들에 따르면, 비디오 인코더 (20) 는 하나 이상의 레퍼런스 샘플 값을 이용함으로써 QP 값을 도출할 수도 있다. 이러한 구현은 다음과 같은 의사코드에 의해 기술된다:
[0232] QP = PQ_LUT[avgPred];
[0233] 상기 의사코드에서, "PQ_LUT" 는, 비디오 인코더 (20) 가 예측된 블록의 평균 밝기 ("avgPred") 값을 연관된 QP 값에 맵핑하기 위해 이용할 수도 있는 툭업 테이블을 나타낸다. 비디오 인코더 (20) 는 레퍼런스 샘플들의 평균 값을 계산하는 등에 의해, 레퍼런스 샘플들의 함수로서 avgPred 의 값을 계산할 수도 있다. 비디오 인코더 (20) 가 본 개시의 계산들에 따라 이용할 수도 있는 평균 값들의 예들은 평균, 중간, 및 모드 값을 중 하나 이상을 포함한다.
[0234] 일부 구현들에서, 비디오 인코더 (20) 는 QP 값을 대신에 스케일링 파라미터들을 도출할 수도 있다. 다른 구현들에서, 비디오 인코더 (20) 는 도출된 QP 값(들)을 스케일 파라미터(들)로, 또는 그 역으로 변환하는 변환
```

프로세스를 이용할 수도 있다. 일부 구현들에서, 비디오 인코더 (20)는 하나 이상의 레퍼런스 샘플들로부터 QP 값을 도출하기 위해 분석적 표현을 이용할 수도 있다. 실제로, 분석적 표현을 이용하기 위해, 비디오 인코더 (20)는 파라미터적 도출 모델을 이용할 수도 있다.

[0235] 도 10 은 본 개시의 기법들을 구현할 수도 있는 비디오 디코더 (30)의 일 예를 나타내는 블록도이다. 도 10 의 예에 있어서, 비디오 디코더 (30)는 엔트로피 디코딩 유닛 (70), 비디오 데이터 메모리 (71), 모션 보상 유닛 (72), 인트라 예측 프로세싱 유닛 (74), 역양자화 유닛 (76), 역변환 프로세싱 유닛 (78), 디코딩된 픽처 버퍼 (82) 및 합산기 (80)를 포함한다. 일부 예들에 있어서, 비디오 디코더 (30)는 비디오 인코더 (20) (도 9)에 대하여 설명된 인코딩 패스에 일반적으로 역인 디코딩 패스를 수행할 수도 있다. 모션 보상 유닛 (72)은 엔트로피 디코딩 유닛 (70)으로부터 수신된 모션 벡터들에 기초하여 예측 데이터를 생성할 수도 있지만, 인트라-예측 프로세싱 유닛 (74)은 엔트로피 디코딩 유닛 (70)으로부터 수신된 인트라-예측 모드 표시자들에 기초하여 예측 데이터를 생성할 수도 있다.

[0236] 비디오 데이터 메모리 (71)는, 비디오 디코더 (30)의 컴포넌트들에 의해 디코딩될 인코딩된 비디오 비트스트림과 같은 비디오 데이터를 저장할 수도 있다. 비디오 데이터 메모리 (71)에 저장된 비디오 데이터는, 예를 들어, 컴퓨터 판독가능 매체 (16)로부터, 예컨대, 카메라와 같은 로컬 비디오 소스로부터, 비디오 데이터의 유선 또는 무선 네트워크 통신을 통해, 또는 물리적 데이터 저장 매체들에 액세스함으로써 획득될 수도 있다.

비디오 데이터 메모리 (71)는, 인코딩된 비디오 비트스트림으로부터 인코딩된 비디오 데이터를 저장하는 코딩된 픽처 버퍼 (CPB)를 형성할 수도 있다. 디코딩된 픽처 버퍼 (82)는, 예를 들어, 인트라- 또는 인터-코딩 모드들에서 비디오 디코더 (30)에 의해 비디오 데이터를 디코딩함에 있어서 사용하기 위한 레퍼런스 비디오 데이터를 저장하는 레퍼런스 픽처 메모리일 수도 있다. 비디오 데이터 메모리 (71) 및 디코딩된 픽처 버퍼 (82)는 동기식 DRAM (SDRAM)을 포함한 동적 랜덤 액세스 메모리 (DRAM), 자기저항성 RAM (MRAM), 저항성 RAM (RRAM), 또는 다른 타입들의 메모리 디바이스들과 같은 다양한 메모리 디바이스들 중 임의의 메모리 디바이스에 의해 형성될 수도 있다. 비디오 데이터 메모리 (71) 및 디코딩된 픽처 버퍼 (82)는 동일한 메모리 디바이스 또는 별도의 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에 있어서, 비디오 데이터 메모리 (71)는 비디오 디코더 (30)의 다른 컴포넌트들과 온-칩형이거나 그 컴포넌트들에 대하여 오프-칩형일 수도 있다.

[0237] 디코딩 프로세스 동안, 비디오 디코더 (30)는, 인코딩된 비디오 슬라이스의 비디오 블록들 및 관련 신팩스 엘리먼트들을 나타내는 인코딩된 비디오 비트스트림을 비디오 인코더 (20)로부터 수신한다. 비디오 디코더 (30)의 엔트로피 디코딩 유닛 (70)은 비트스트림을 엔트로피 디코딩하여, 양자화된 계수들, 모션 벡터들 또는 인트라-예측 모드 표시자들, 및 다른 신팩스 엘리먼트들을 생성한다. 엔트로피 디코딩 유닛 (70)은 모션 벡터들 및 다른 신팩스 엘리먼트들을 모션 보상 유닛 (72)으로 포워딩한다. 비디오 디코더 (30)는 신팩스 엘리먼트들을 비디오 슬라이스 레벨 및/또는 비디오 블록 레벨로 수신할 수도 있다.

[0238] 비디오 슬라이스가 인트라-코딩된 (I) 슬라이스로서 코딩된 경우, 인트라 예측 프로세싱 유닛 (74)은 현재 프레임 또는 픽처의 이전에 디코딩된 블록들로부터의 데이터 및 시그널링된 인트라 예측 모드에 기초하여 현재 비디오 슬라이스의 비디오 블록에 대한 예측 데이터를 생성할 수도 있다. 비디오 프레임이 인터-코딩된 (즉, B 또는 P) 슬라이스로서 코딩된 경우, 모션 보상 유닛 (72)은 엔트로피 디코딩 유닛 (70)으로부터 수신된 모션 벡터들 및 다른 신팩스 엘리먼트들에 기초하여 현재 비디오 슬라이스의 비디오 블록에 대한 예측 블록들을 생성한다. 예측 블록들은 레퍼런스 픽처 리스트들 중 하나 내에서의 레퍼런스 픽처들 중 하나로부터 생성될 수도 있다. 비디오 디코더 (30)는 디코딩된 픽처 버퍼 (82)에 저장된 레퍼런스 픽처들에 기초한 디폴트 구성 기법들을 이용하여 레퍼런스 픽처 리스트들, 즉, 리스트 0 및 리스트 1을 구성할 수도 있다. 모션 보상 유닛 (72)은 모션 벡터들 및 다른 신팩스 엘리먼트들을 포함함으로써 현재 비디오 슬라이스의 비디오 블록에 대한 예측 블록들을 생성한다. 예를 들어, 모션 보상 유닛 (72)은 수신된 신팩스 엘리먼트들 중 일부를 사용하여, 비디오 슬라이스의 비디오 블록들을 코딩하는데 사용되는 예측 모드 (예를 들어, 인트라-예측 또는 인터-예측), 인터-예측 슬라이스 타입 (예를 들어, B 슬라이스 또는 P 슬라이스), 슬라이스에 대한 레퍼런스 픽처 리스트들 중 하나 이상에 대한 구성 정보, 슬라이스의 각각의 인터-인코딩된 비디오 블록에 대한 모션 벡터들, 슬라이스의 각각의 인터-코딩된 비디오 블록에 대한 인터-예측 상태, 및 현재 비디오 슬라이스에 있어서의 비디오 블록들을 디코딩하기 위한 다른 정보를 결정한다.

[0239] 모션 보상 유닛 (72)은 또한, 보간 필터들에 기초하여 보간을 수행할 수도 있다. 모션 보상 유닛 (72)은 비디오 블록들의 인코딩 동안 비디오 인코더 (20)에 의해 사용된 바와 같은 보간 필터들을 이용하여, 레퍼런스

블록들의 서브-정수 픽셀들에 대한 보간된 값들을 계산할 수도 있다. 이 경우, 모션 보상 유닛 (72)은 수신된 신택스 엘리먼트들로부터 비디오 인코더 (20)에 의해 사용된 보간 필터들을 결정하고, 보간 필터들을 이용하여 예측 블록들을 생성할 수도 있다.

[0240] 역양자화 유닛 (76)은 비트스트림에서 제공되고 엔트로피 디코딩 유닛 (70)에 의해 디코딩되는 양자화된 변환 계수들을 역양자화, 즉, 양자화해제한다. 역양자화 프로세스는 비디오 슬라이스에 있어서의 각각의 비디오 블록에 대해 비디오 디코더 (30)에 의해 계산된 양자화 파라미터 (QP_Y)의 이용을 포함하여, 적용되어야 하는 양자화의 정도 및 유사하게 역양자화의 정도를 결정할 수도 있다.

[0241] 역변환 프로세싱 유닛 (78)은 픽셀 도메인에서 잔차 블록들을 생성시키기 위해, 역변환, 예를 들어, 역 DCT, 정수 역변환, 또는 개념적으로 유사한 역변환 프로세스를 변환 계수들에 적용한다.

[0242] 모션 보상 유닛 (72)이 모션 벡터들 및 다른 신택스 엘리먼트들에 기초하여 현재 비디오 블록에 대한 예측 블록을 생성한 이후, 비디오 디코더 (30)는 역변환 프로세싱 유닛 (78)으로부터의 잔차 블록들을 모션 보상 유닛 (72)에 의해 생성된 대응하는 예측 블록들과 합산함으로써 디코딩된 비디오 블록을 형성한다. 합산기 (80)는 이러한 합산 연산을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다. 요구된다면, 디블로킹 필터가 또한, 블록기니스 아티팩트들을 제거하기 위해, 디코딩된 블록들을 필터링하는데 적용될 수도 있다. (코딩 루프에 있어서 또는 코딩 루프 이후에) 다른 루프 필터들이 또한 픽셀 천이들을 평활하게 하거나 그렇지 않으면 비디오 품질을 개선하기 위해 이용될 수도 있다. 그 후, 소정의 프레임 또는 꾹처에 있어서의 디코딩된 비디오 블록들이 디코딩된 꾹처 버퍼 (82)에 저장되고, 이 디코딩된 꾹처 버퍼 (82)는 후속적인 모션 보상을 위해 사용되는 레퍼런스 꾹처들을 저장한다. 디코딩된 꾹처 버퍼 (82)는 또한, 도 1의 디스플레이 디바이스 (32)와 같은 디스플레이 디바이스 상으로의 더 나중의 프리젠테이션을 위해 디코딩된 비디오를 저장한다.

[0243] 비디오 디코더 (30)는 상술된 기법들 중 하나 이상에 따라 비디오 인코더 (20)에 의해 획득된 QP 값으로부터 도출되는 엘타 QP 값을 인코딩된 비디오 비트스트림에서 수신할 수도 있다. 엘타 QP 값을 이용하여, 비디오 디코더 (30)는, 도 8에서 예시된 현재-코딩되는 블록 (152)과 같이, 현재 코딩되고 있는 블록에 속하는 QP 값을 획득할 수도 있다. 다음으로, 비디오 디코더 (30)는 그 QP 값을 이용하여 현재-코딩되는 블록 (152)을 역양자화할 수도 있다.

[0244] 비디오 디코더 (30)가 현재-코딩되는 블록 (152)에 대한 스케일링 파라미터들을 수신하는 경우들에서, 비디오 디코더 (30)는 스케일링된 값을 DC' 및 AC' 을 피연산자들로서 사용하는 다양한 것들에 대해 일반적으로 상호 역인 역 스케일링 프로세스를 구현하기 위해 스케일링 파라미터들을 이용할 수도 있다. 즉, 비디오 디코더 (30)는 아래 식들에 의해 표현되는 바와 같이 역-스케일링된 DC 계수들 DC'' 및 역-스케일링된 AC 변환 계수들 AC'' 을 획득하기 위해 스케일링된 DC 변환 계수들 DC' 및 스케일링된 AC 변환 계수들 AC' 을 역-스케일링하기 위해 스케일링 파라미터들을 적용할 수도 있다. 비디오 디코더 (30)는 다음과 같은 식들에서 예시된 바와 같이 역 스케일링 연산들을 구현할 수도 있다:

[0245] $DC'' = DC' / \text{scale}(\text{fun1}(DC', \text{avgPred}))$; 및

[0246] $AC'' = AC' / \text{scale}(\text{fun2}(DC'', \text{avgPred}))$

[0247] 용어들 'fun1' 및 'fun2'는 DC-기반 값들 및 레퍼런스 샘플들의 평균을 인수들로서 사용하는 스케일 도출 함수들/프로세스들을 정의한다. 비디오 디코더 (30)에 의해 구현되는 역-스케일링 기법들에 대해 예시된 바와 같이, 본 개시의 기법들은 DC 및 AC 양 변환 계수 값들의 도출에서 DC 변환 계수 값들의 사용을 가능하게 한다. 이러한 방식으로, 본 개시의 기법들은, 비디오 디코더 (30)가, 역-스케일링 연산들이 변환 계수들의 양자화 및 역양자화 대신에 또는 조합하여 수행되든지 여부에 관계 없이, 역-스케일링 연산들에서 DC 변환 계수 값을 레버리징하는 것을 가능하게 한다.

[0248] 도 11은 본 개시의 다양한 양태들에 따라, 비디오 디코더 (30)가 수행할 수도 있는 예시적인 프로세스 (170)를 나타내는 플로우차트이다. 프로세스 (170)는 비디오 디코더 (30)가 현재 블록 (152)의 인코딩된 표현을 포함하는 인코딩된 비디오 비트스트림을 수신할 때 시작할 수도 있다 (172). 비디오 디코더 (30)는 현재 블록 (152)에 대해 공간-시간적 이웃하는 QP 정보에 기초하는 QP 값을 재구성할 수도 있다 (174). 실제로, 비디오 디코더 (30)는 인코딩된 비디오 비트스트림에서 시그널링된 엘타 QP 값으로부터 QP 를 재구성할 수도 있다. 재구성된 QP 값은 도 8에서 예시된 블록들 (154-158) 중 하나 이상으로부터의 QP 정보에 기초할 수도 있다. 상기 논의된 바와 같이, QP 값을 재구성하기 위해, 비디오 디코더 (30)는 레퍼런스 QP 값을 생성하기 위해 공간-시간적 이웃 블록들 (154-158) 중 2개 이상의 이웃 블록들의 QP 값을 평균화하고, 그 다음

예, 레퍼런스 QP 값에 델타 QP 값을 가산하여 종국적으로 현재 블록에 대한 재구성된 QP 값을 생성할 수도 있다. 다음으로, 비디오 디코더 (30) (및 보다 상세하게는, 역 양자화 유닛 (76))는 공간-시간적 이웃 QP 정보에 기초하는 재구성된 QP 값을 이용하여 현재 블록 (152)의 CABAC-디코딩된 변환 계수들을 양자화해제 (즉, 역-양자화) 할 수도 있다 (176). 일부 예들에서, 비디오 디코더 (30)는 공간-시간적 이웃의 샘플들에 기초하여 현재 블록 (152)의 샘플들에 대한 레퍼런스 QP 값을 획득할 수도 있고, 현재 블록 (152)의 샘플들을 역양자화하기 위한 QP 값을 도출하기 위해 레퍼런스 QP 값에 델타 QP 값을 가산할 수도 있다.

[0249] 도 12는 비디오 디코더 (30)가 본 개시의 다양한 양태들에 따라 수행할 수도 있는 예시적인 프로세스 (190)를 나타내는 플로우차트이다. 프로세스 (190)는 비디오 디코더 (30)가 현재 블록 (152)의 인코딩된 표현을 포함하는 인코딩된 비디오 비트스트림을 수신할 때 시작할 수도 있다 (192). 비디오 디코더 (30)는 현재 블록 (152)에 대해 공간-시간적 이웃 스케일링 정보에 기초하는 스케일링 파라미터를 재구성할 수도 있다 (194). 실제로, 재구성된 스케일링 파라미터는 도 8에서 예시된 블록들 (154-158)의 하나 이상으로부터의 스케일링 정보에 기초할 수도 있다. 다음으로, 비디오 디코더 (30)는 공간-시간적 이웃 QP 정보에 기초하는 재구성된 스케일링 파라미터를 이용하여 현재 블록 (152)을 역 스케일링할 수도 있다 (196). 일부 예들에서, 비디오 디코더 (30)는, 복수의 역-스케일링된 DC 변환 계수들을 획득하기 위해 현재 블록 (152)의 변환 계수들의 복수의 DC 변환 계수들에 제 1 역 스케일링 도출 프로세스를 적용할 수도 있고, 복수의 역-스케일링된 AC 변환 계수들을 획득하기 위해 현재 블록 (152)의 변환 계수들의 복수의 역-스케일링된 DC 변환 계수들에 제 2 역 스케일링 도출 프로세스를 적용할 수도 있다.

[0250] 도 13은 비디오 인코더 (20)가 본 개시의 다양한 양태들에 따라 수행할 수도 있는 예시적인 프로세스 (210)를 나타내는 플로우차트이다. 프로세스 (210)는 비디오 인코더 (20)가 현재 블록 (152)의 공간-시간적 이웃하는 QP 정보로부터 현재 블록 (152)에 대한 QP 값을 도출할 때 시작할 수도 있다 (212). 비디오 인코더 (20)는 공간-시간적 이웃 QP 정보로부터 도출된 QP 값을 이용하여 현재 블록 (152)을 양자화할 수도 있다 (214). 다음으로, 비디오 인코더 (20)는 인코딩된 비디오 비트스트림에서 공간-시간적 이웃 QP 정보에 기초하는 QP로부터 도출된 델타 QP 값을 시그널링할 수도 있다 (216). 일부 예들에서, 비디오 인코더 (20)는 공간적 이웃 블록들 (154 및/또는 156) 및/또는 시간적 이웃 블록 (158) 중 2 개 이상의 블록의 샘플들과 연관된 이웃 QP 값을 선택할 수도 있다. 일부 예들에서, 비디오 인코더 (20)는 평균 QP 값을 획득하기 위해 선택된 이웃 QP 값을 평균화할 수도 있고, 평균 QP 값으로부터 현재 블록에 대한 QP 값을 도출할 수도 있다. 일부 예들에서, 비디오 인코더 (20)는 공간-시간적 이웃의 샘플들에 기초하여 현재 블록 (152)의 샘플들에 대한 레퍼런스 QP 값을 획득할 수도 있다. 이들 예들에서, 비디오 인코더 (20)는 현재 블록 (152)의 샘플들에 대한 델타 양자화 파라미터 (QP) 값을 도출하기 위해 QP 값으로부터 레퍼런스 QP 값을 감산할 수도 있고, 그 델타 QP 값을 인코딩된 비디오 비트스트림에서 시그널링할 수도 있다.

[0251] 도 14는 비디오 인코더 (20)가 본 개시의 다양한 양태들에 따라 수행할 수도 있는 예시적인 프로세스 (240)를 나타내는 플로우차트이다. 프로세스 (240)는 비디오 인코더 (20)가 현재 블록 (152)의 공간-시간적 이웃 스케일링 정보로부터 현재 블록 (152)에 대한 스케일링 파라미터를 도출할 때 시작할 수도 있다 (242). 비디오 인코더 (20)는 공간-시간적 이웃 스케일링 정보로부터 도출된 스케일링 파라미터를 이용하여 현재 블록 (152)을 스케일링할 수도 있다 (244). 다음으로, 비디오 인코더 (20)는 인코딩된 비디오 비트스트림에서 공간-시간적 이웃 스케일링 정보에 기초하는 스케일링 파라미터를 시그널링할 수도 있다 (246).

[0252] 상술된 바와 같이, 개시된 시스템들 및 기법들은 또한, 신호의 공간-시간적 이웃으로부터 양자화 또는 스케일링 파라미터들의 도출을 위한 수개의 알고리즘들을 통합 또는 포함한다. 즉, 본 개시의 예시적인 시스템들 및 기법들은 코딩 프로세스에서 현재 블록과 연관된 잔차 데이터를 수정하기 위해 사용되는 하나 이상의 파라미터 값을 획득하는 것에 지향된다. 본 명세서에서 사용된 바와 같이, 잔차 데이터를 수정하기 위해 사용되는 파라미터 값은 (인코딩 프로세스 또는 디코딩 프로세스에서 각각 잔차 데이터를 양자화 또는 역양자화함으로써 잔차 데이터를 수정하기 위해 사용되는) 양자화 파라미터, 또는 (인코딩 프로세스 또는 디코딩 프로세스에서 각각 잔차 데이터를 스케일링 또는 역-스케일링함으로써 잔차 데이터를 수정하기 위해 사용되는) 스케일링 파라미터를 포함할 수도 있다.

[0253] 본 개시의 특정 양태들은 예시의 목적들로 HEVC 표준의 화장들에 관하여 설명되었다. 하지만, 본 개시에서 설명된 기법들은, 다른 표준 또는 아직 개발되지 않은 전매특허의 비디오 코딩 프로세스들을 포함한 다른 비디오 코딩 프로세스들에 대해 유용할 수도 있다.

[0254] 본 개시에서 설명된 바와 같은 비디오 코더는 비디오 인코더 또는 비디오 디코더를 지칭할 수도 있다. 유사

하게, 비디오 코딩 유닛은 비디오 인코더 또는 비디오 디코더를 지칭할 수도 있다. 유사하게, 비디오 코딩은, 적용가능할 때, 비디오 인코딩 또는 비디오 디코딩을 지칭할 수도 있다.

[0255] 예에 의존하여, 본 명세서에서 설명된 기법들의 임의의 특정 작동들 또는 이벤트들은 상이한 시퀀스로 수행될 수 있고, 전체적으로 부가되거나 병합되거나 또는 제거될 수도 있음 (예를 들어, 설명된 모든 작동들 또는 이벤트들이 그 기법들의 실시를 위해 필수적인 것은 아님) 이 인식되어야 한다. 더욱이, 특정 예들에 있어서, 작동들 또는 이벤트들은 순차적인 것보다는, 예를 들어, 다중-스레딩된 프로세싱, 인터럽트 프로세싱, 또는 다중의 프로세서들을 통해 동시에 수행될 수도 있다.

[0256] 하나 이상의 예들에 있어서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현된다면, 그 기능들은 하나 이상의 명령들 또는 코드로서 컴퓨터 관독가능 매체 상으로 저장 또는 전송되고 하드웨어 기반 프로세싱 유닛에 의해 실행될 수도 있다. 컴퓨터 관독가능 매체는, 예를 들어, 통신 프로토콜에 따라 일 장소로부터 다른 장소로의 컴퓨터 프로그램의 전송을 용이하게 하는 임의의 매체를 포함하는 통신 매체, 또는 데이터 저장 매체와 같은 유형의 매체에 대응하는 컴퓨터 관독가능 저장 매체를 포함할 수도 있다. 이러한 방식으로, 컴퓨터 관독가능 매체는 일반적으로 (1) 비-일시적인 유형의 컴퓨터 관독가능 저장 매체 또는 (2) 신호 또는 캐리어파와 같은 통신 매체에 대응할 수도 있다. 데이터 저장 매체는 본 개시에서 설명된 기법들의 구현을 위한 명령들, 코드 및/또는 데이터 구조들을 취출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 가용 매체일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터 관독가능 매체를 포함할 수도 있다.

[0257] 한정이 아닌 예로서, 그러한 컴퓨터 관독가능 저장 매체는 RAM, ROM, EEPROM, CD-ROM 또는 다른 광학 디스크 저장부, 자기 디스크 저장부 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 원하는 프로그램 코드를 명령들 또는 데이터 구조들의 형태로 저장하는데 이용될 수 있고 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 커넥션이 컴퓨터 관독가능 매체로 적절히 명명된다. 예를 들어, 동축 케이블, 광섬유 케이블, 꼬임쌍선, 디지털가입자 라인 (DSL), 또는 적외선, 무선, 및 마이크로파와 같은 무선 기술들을 이용하여 웹사이트, 서버, 또는 다른 원격 소스로부터 명령들이 송신된다면, 동축 케이블, 광섬유 케이블, 꼬임쌍선, DSL, 또는 적외선, 무선, 및 마이크로파와 같은 무선 기술들은 매체의 정의에 포함된다. 하지만, 컴퓨터 관독가능 저장 매체 및 데이터 저장 매체는 커넥션들, 캐리어파들, 신호들, 또는 다른 일시적 매체들을 포함하지 않지만 대신 비-일시적인 유형의 저장 매체들로 지향됨을 이해해야 한다. 본 명세서에서 사용된 바와 같은 디스크 (disk) 및 디스크 (disc) 는 컴팩트 디스크 (CD), 레이저 디스크, 광학 디스크, 디지털 다기능 디스크 (DVD), 플로피 디스크 및 블루레이 디스크를 포함하며, 여기서, 디스크 (disk) 는 통상적으로 데이터를 자기적으로 재생하지만 디스크 (disc) 는 레이저들을 이용하여 데이터를 광학적으로 재생한다. 상기의 조합들이 또한, 컴퓨터 관독가능 매체의 범위 내에 포함되어야 한다.

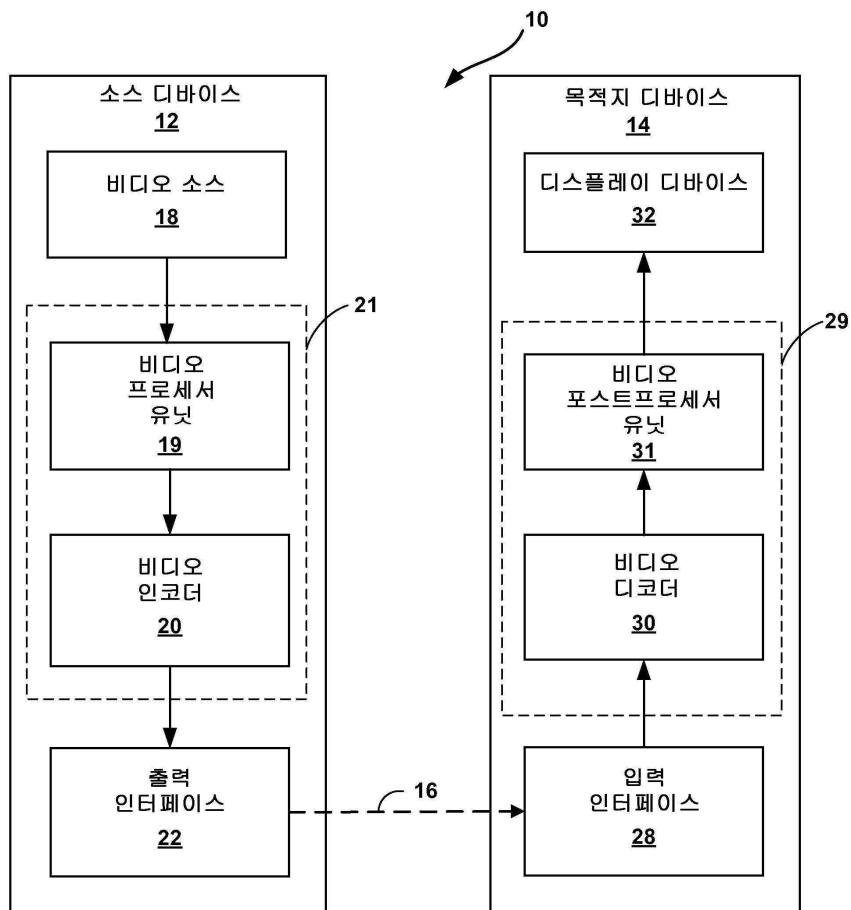
[0258] 명령들은 하나 이상의 디지털 신호 프로세서들 (DSP들), 범용 마이크로프로세서들, 주문형 집적회로들 (ASIC들), 필드 프로그래밍가능 로직 어레이들 (FPGA들), 또는 다른 등가의 집적된 또는 별도의 로직 회로와 같은 하나 이상의 프로세서들에 의해 실행될 수도 있다. 이에 따라, 본 명세서에서 사용된 바와 같은 용어 "프로세서" 는 본 명세서에서 설명된 기법들의 구현에 적합한 전술한 구조 또는 임의의 다른 구조 중 임의의 구조를 지칭할 수도 있다. 추가적으로, 일부 양태들에 있어서, 본 명세서에서 설명된 기능은 인코딩 및 디코딩을 위해 구성되고 결합된 코덱에서 통합된 전용 하드웨어 및/또는 소프트웨어 모듈들 내에 제공될 수도 있다. 또한, 그 기법들은 하나 이상의 회로들 또는 로직 엘리먼트들에서 완전히 구현될 수 있다.

[0259] 본 개시의 기법들은 무선 핸드셋, 집적 회로 (IC) 또는 IC들의 세트 (예를 들어, 칩 세트) 를 포함하여 매우 다양한 디바이스들 또는 장치들에서 구현될 수도 있다. 다양한 컴포넌트들, 모듈들 또는 유닛들이 개시된 기법들을 수행하도록 구성된 디바이스들의 기능적 양태들을 강조하기 위해 본 개시에서 설명되지만, 반드시 상이한 하드웨어 유닛들에 의한 실현을 요구하지는 않는다. 오히려, 상기 설명된 바와 같이, 다양한 유닛들은 적합한 소프트웨어 및/또는 펌웨어와 함께 상기 설명된 바와 같은 하나 이상의 프로세서들을 포함하여 코덱 하드웨어 유닛으로 결합되거나 또는 상호운용식 하드웨어 유닛들의 집합에 의해 제공될 수도 있다.

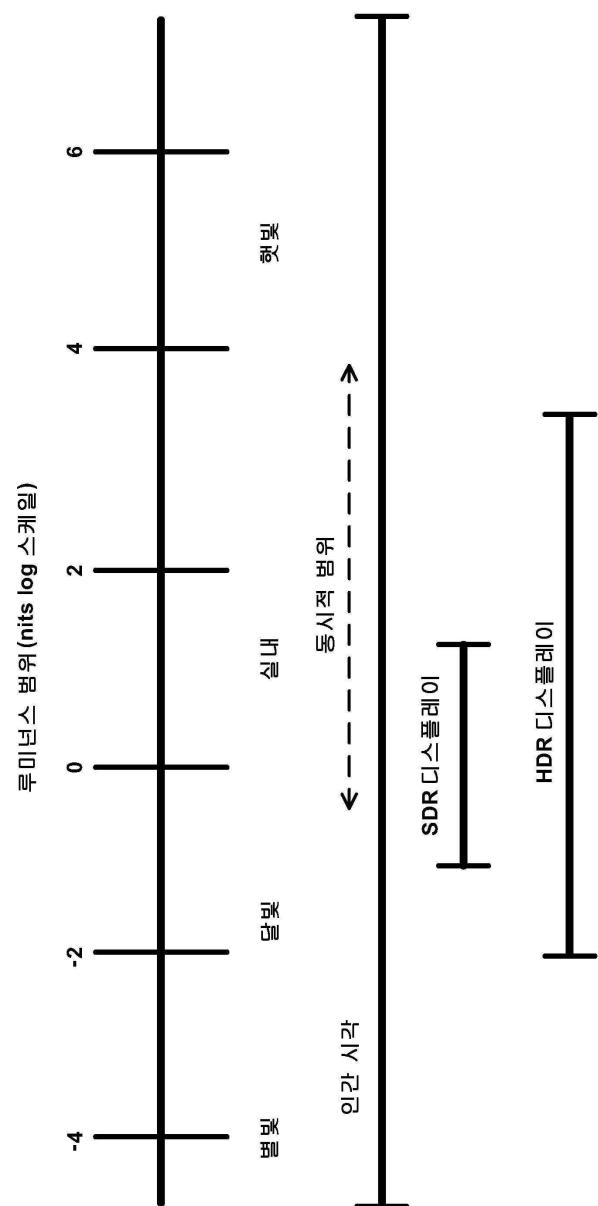
[0260] 다양한 예들이 설명되었다. 이들 및 다른 예들은 다음의 청구항들의 범위 내에 있다.

도면

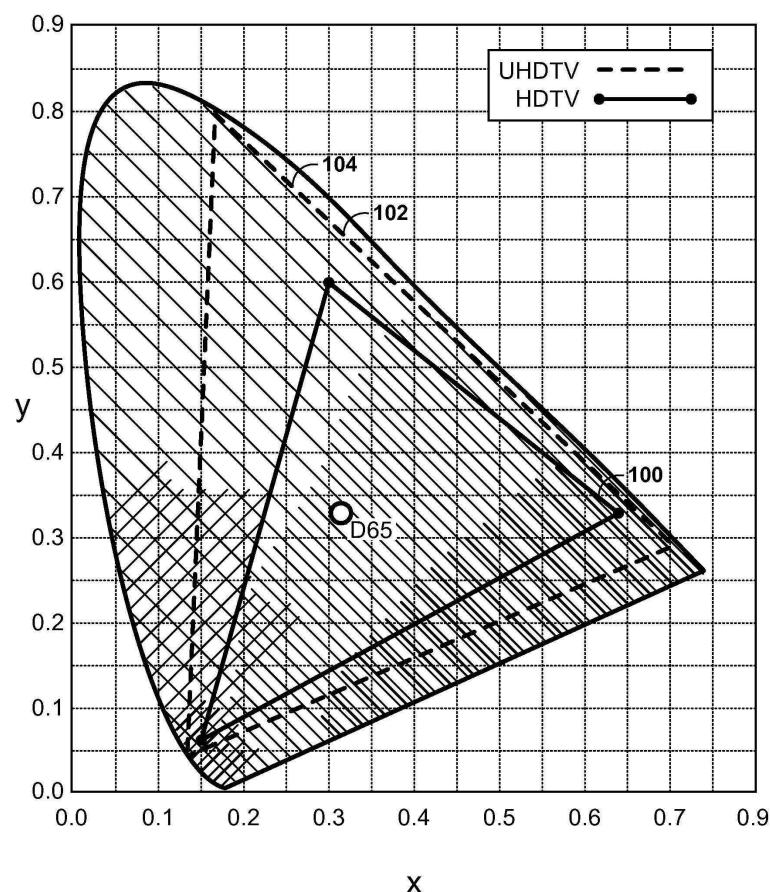
도면1



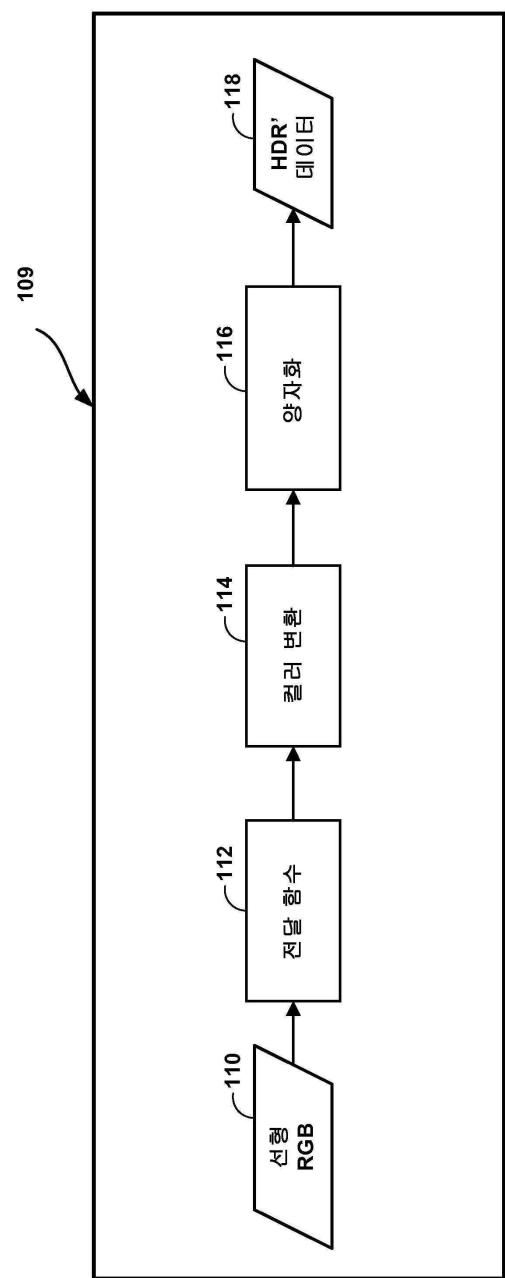
도면2



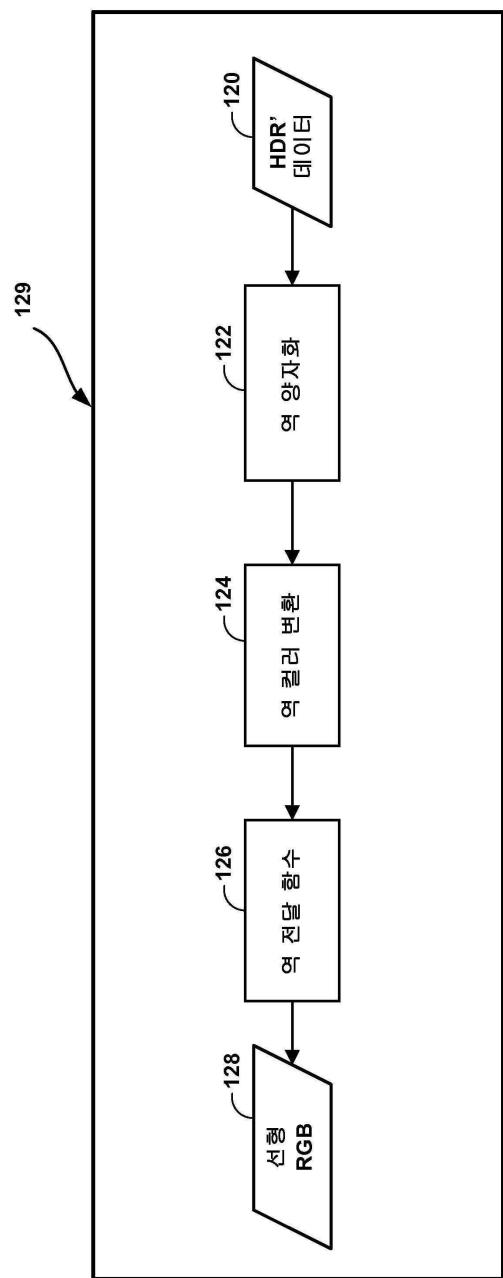
도면3



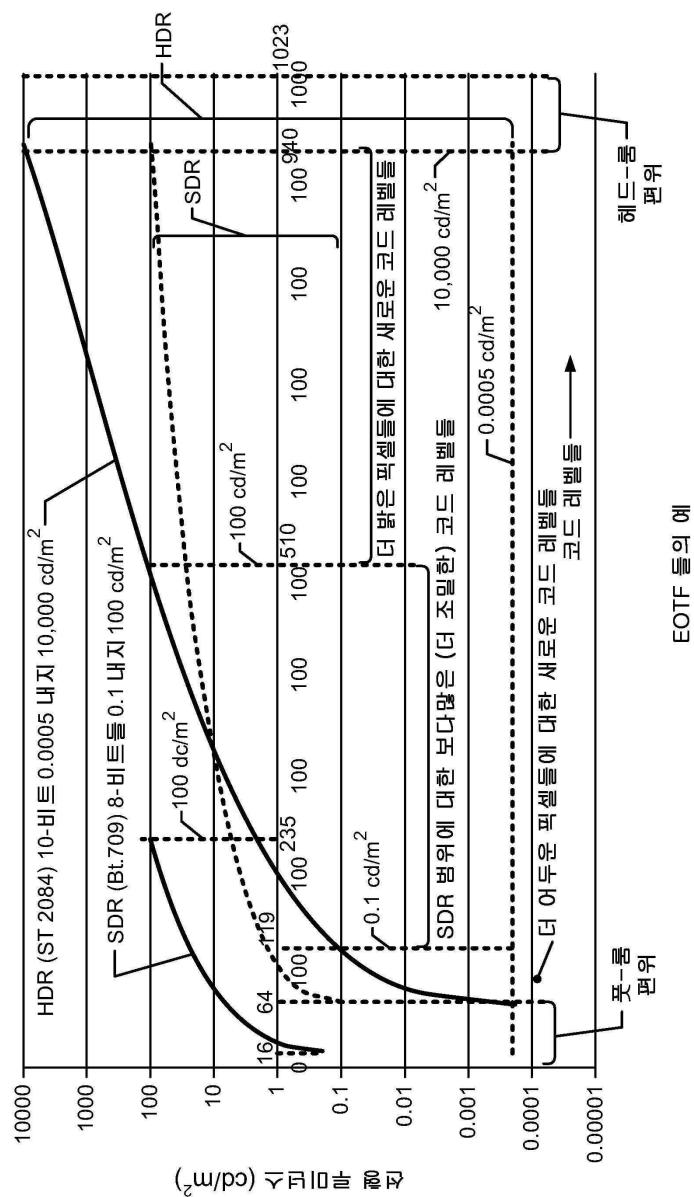
도면4



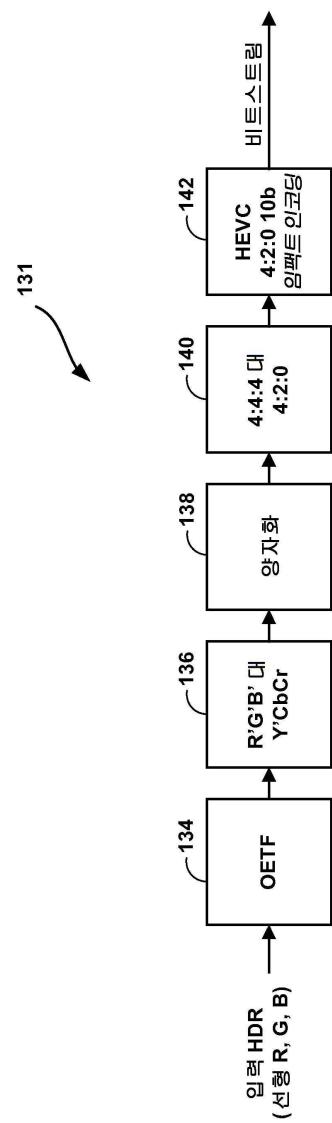
도면5



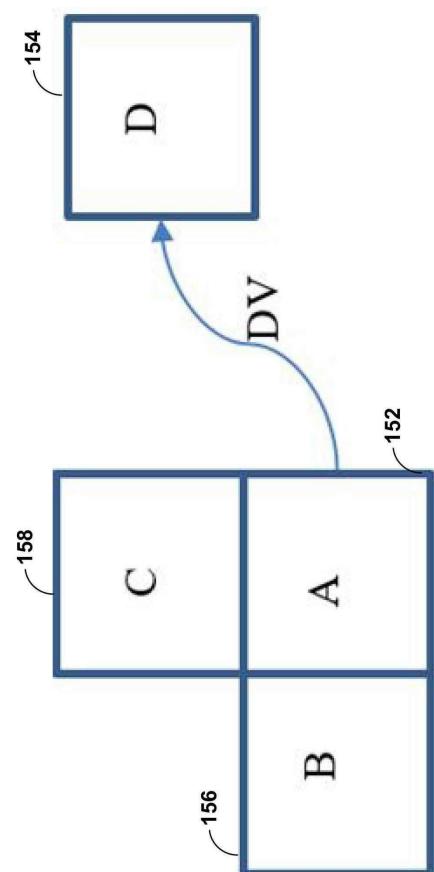
도면6



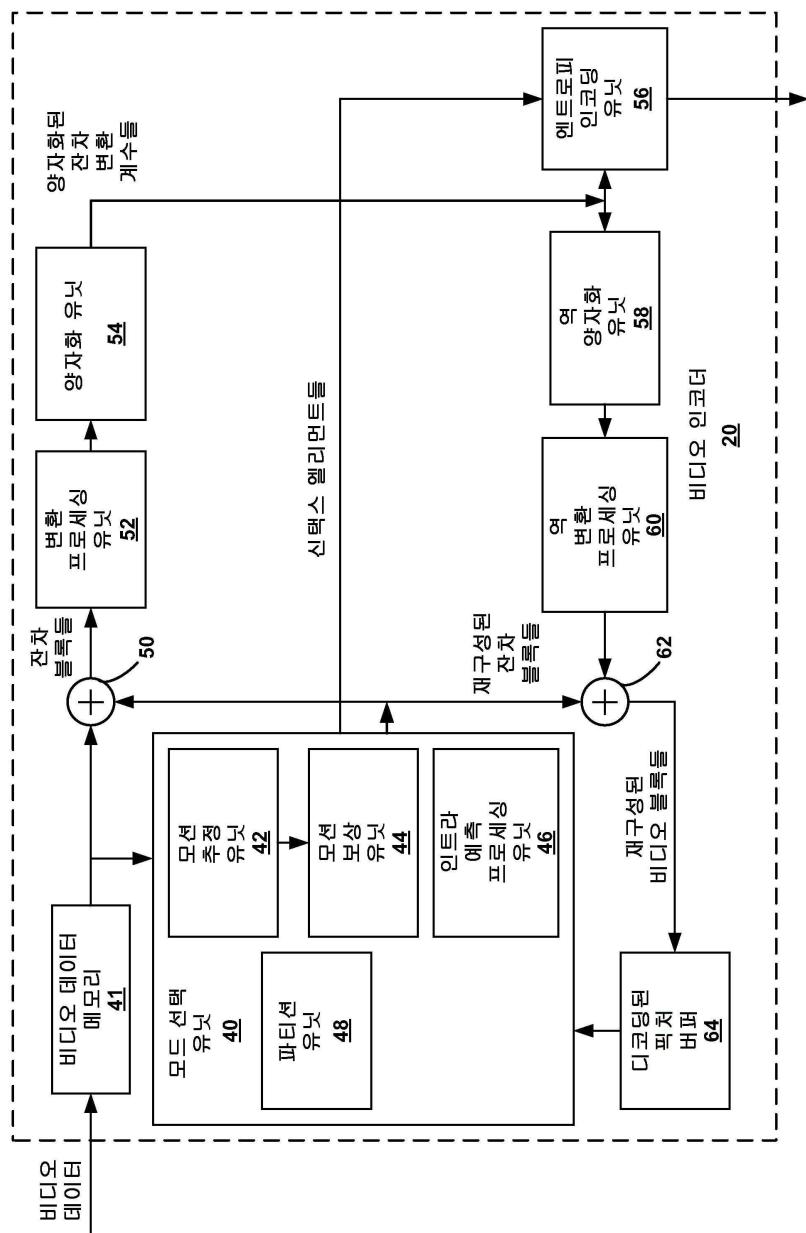
도면7



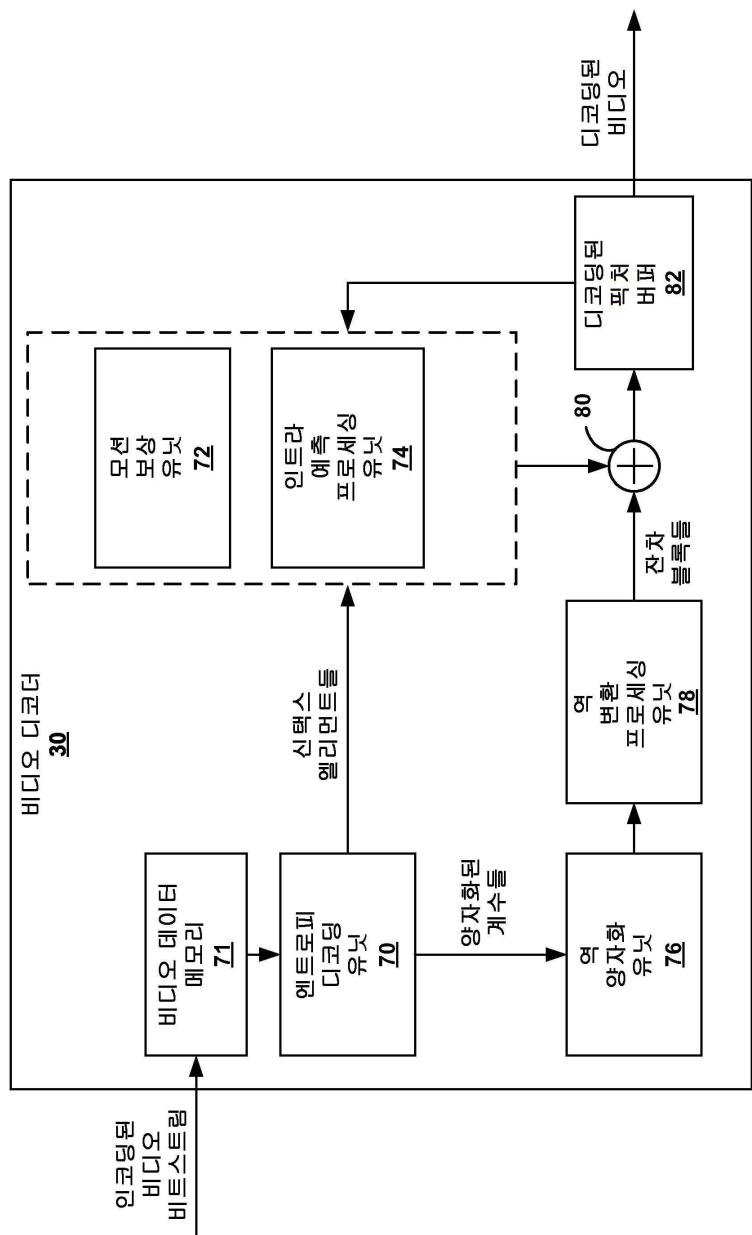
도면8



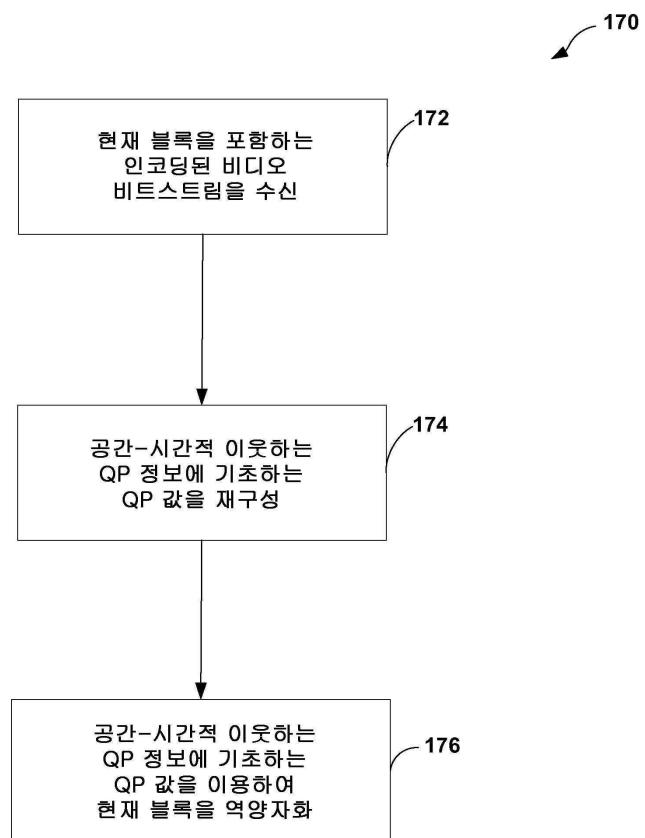
도면9



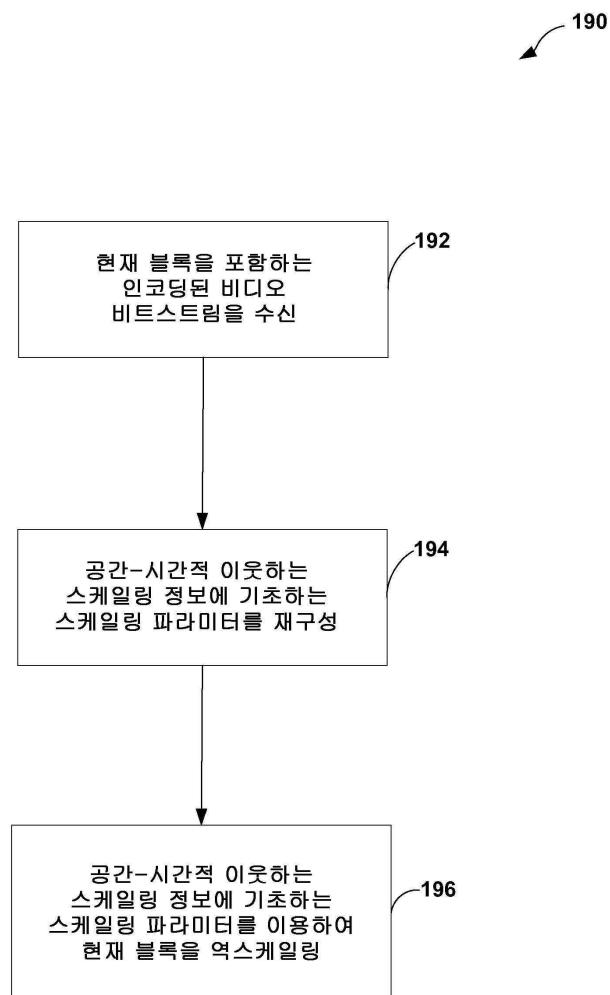
도면10



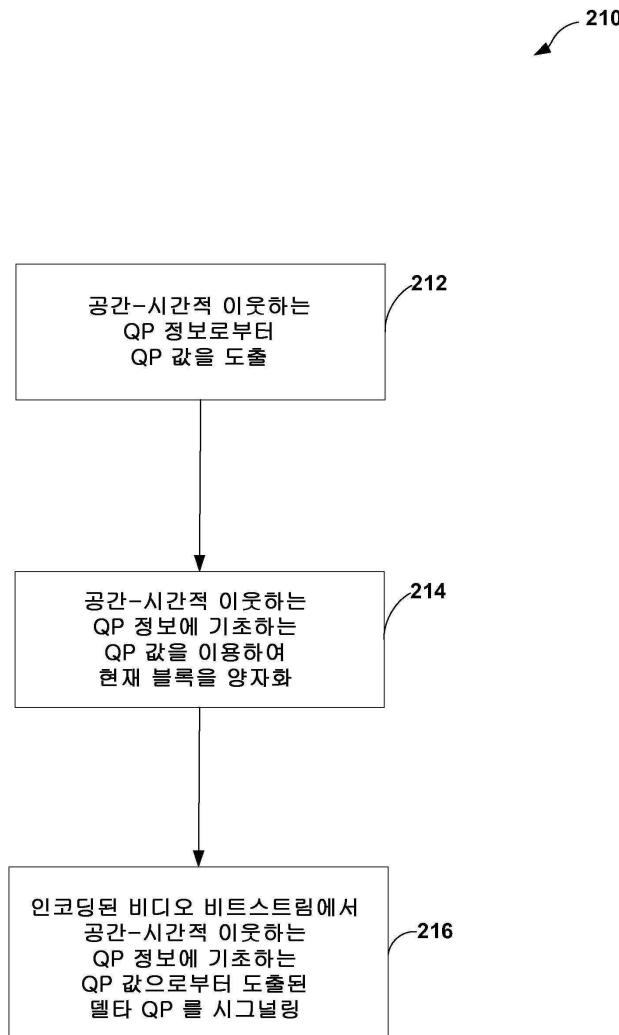
도면11



도면12



도면13



도면14

