



등록특허 10-2824171



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2025년06월23일

(11) 등록번호 10-2824171

(24) 등록일자 2025년06월19일

(51) 국제특허분류(Int. Cl.)

H04N 19/196 (2014.01) H04N 19/117 (2014.01)

H04N 19/132 (2014.01) H04N 19/154 (2014.01)

H04N 19/184 (2014.01) H04N 19/187 (2014.01)

H04N 19/36 (2014.01) H04N 19/44 (2014.01)

H04N 19/46 (2014.01) H04N 19/80 (2014.01)

H04N 19/85 (2014.01)

(52) CPC특허분류

H04N 19/196 (2015.01)

H04N 19/117 (2015.01)

(21) 출원번호 10-2018-7011007

(22) 출원일자(국제) 2016년09월20일

심사청구일자 2021년09월02일

(85) 번역문제출일자 2018년04월18일

(65) 공개번호 10-2018-0054782

(43) 공개일자 2018년05월24일

(86) 국제출원번호 PCT/US2016/052633

(87) 국제공개번호 WO 2017/053277

국제공개일자 2017년03월30일

(30) 우선권주장

62/221,586 2015년09월21일 미국(US)

(뒷면에 계속)

(56) 선행기술조사문헌

D. Bugdayci Sansli et al, Dynamic Range Adjustment SEI Message, JCT-VC of ITU-T and ISO/IEC, JCTVC-U0098 (2015-06-25)

A. Dsouza et al., Adaptive Gamut Expansion for Chroma Components, JCT-VC of ITU-T and ISO/IEC, JCTVC-W0040 (2016-02-10)

전체 청구항 수 : 총 50 항

심사관 : 황수진

(54) 발명의 명칭 비디오 코딩에서의 성분들의 범위 조정의 고정 소수점 구현

(57) 요 약

고정 소수점 구현을 이용하여 높은 동적 범위 또는 넓은 컬러 영역 비디오 데이터를 프로세싱하는 것이 개시된다. 비디오 데이터를 프로세싱하는 방법은 역 동적 범위 조정 프로세스를 수행하기 위한 파라미터들을 결정하는 방법을 규정하는 정보를 포함하는 하나 이상의 보충 강화 정보(SEI) 메시지들을 수신하는 단계, 디코

(뒷면에 계속)

대 표 도 - 도13

딩된 비디오 데이터를 수신하는 단계, 및 하나 이상의 SEI 메시지들에서의 정보에 따라 고정 소수점 컴퓨팅을 이용하여 디코딩된 비디오 데이터에 대한 역 동적 범위 조정 프로세스를 수행하는 단계를 포함한다.

(52) CPC특허분류

H04N 19/132 (2015.01)*H04N 19/154* (2015.01)*H04N 19/184* (2015.01)*H04N 19/187* (2015.01)*H04N 19/36* (2015.01)*H04N 19/44* (2015.01)*H04N 19/46* (2015.01)*H04N 19/80* (2015.01)*H04N 19/85* (2015.01)

(72) 발명자

솔레 로할스 호엘

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

이 성원

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

부그다이지 산스리 도네

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

카르체비츠 마르타

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

(30) 우선권주장

62/236,804 2015년10월02일 미국(US)

62/241,063 2015년10월13일 미국(US)

15/269,558 2016년09월19일 미국(US)

명세서

청구범위

청구항 1

비디오 데이터를 프로세싱하는 방법으로서,

역 동적 범위 조정 프로세스를 수행하기 위한 파라미터들을 결정하는 방법을 규정하는 정보를 포함하는 하나 이상의 선택스 엘리먼트들을 수신하는 단계로서, 상기 정보는 하나 이상의 컬러 성분에 대한 값들의 범위 파라미터를 결정하는데 이용되는 분수 비트들의 제 1 수, 스케일 파라미터를 결정하는데 이용되는 분수 비트들의 제 2 수, 및 오프셋 파라미터를 결정하는데 이용되는 분수 비트들의 제 3 수 중 하나 이상을 나타내는, 상기 하나 이상의 선택스 엘리먼트들을 수신하는 단계;

디코딩된 비디오 데이터를 수신하는 단계; 및

수신된 상기 정보에 따라 고정 소수점 컴퓨팅 (fixed-point computing) 을 이용하여 상기 디코딩된 비디오 데이터에 대한 역 동적 범위 조정 프로세스를 수행하는 단계를 포함하는, 비디오 데이터를 프로세싱하는 방법.

청구항 2

제 1 항에 있어서,

상기 하나 이상의 선택스 엘리먼트들을 수신하는 단계는 하나 이상의 보충 강화 정보 (supplemental enhancement information; SEI) 메시지들에서 하나 이상의 선택스 엘리먼트들을 수신하는 단계를 포함하고, 상기 파라미터들은 상기 범위 파라미터, 상기 스케일 파라미터, 또는 상기 오프셋 파라미터 중 하나 이상을 포함하는, 비디오 데이터를 프로세싱하는 방법.

청구항 3

삭제

청구항 4

제 1 항에 있어서,

상기 분수 비트들의 제 1 수, 상기 분수 비트들의 제 2 수 및 상기 분수 비트들의 제 3 수를 이용하여 상기 파라미터들을 결정하는 단계를 더 포함하는, 비디오 데이터를 프로세싱하는 방법.

청구항 5

제 4 항에 있어서,

상기 파라미터들을 결정하는 단계는:

상기 파라미터들을 결정하는데 이용되는 임의의 중간 계산 프로세스를 동안의 임의의 분수 비트들을 누적하는 것에 의해 상기 분수 비트들의 제 1 수, 상기 분수 비트들의 제 2 수, 또는 상기 분수 비트들의 제 3 수 중 적어도 하나가 서로 상이한 경우에 상기 파라미터들을 결정하는 단계; 및

미리 정해진 분수 정확도에 기초하여 상기 파라미터들을 결정하기 위한 최종 결과를 클립핑하는 단계를 더 포함하는, 비디오 데이터를 프로세싱하는 방법.

청구항 6

제 4 항에 있어서,

상기 파라미터들을 결정하는 단계는 상기 파라미터들을 결정하는데 이용되는 모든 중간 계산 프로세스를 동안 원하는 분수 정확도에 대해 임의의 분수 비트들을 절삭 (truncating) 하는 것에 의해 상기 파라미터들을 결정하는 단계를 더 포함하는, 비디오 데이터를 프로세싱하는 방법.

청구항 7

제 1 항에 있어서,

상기 정보는 상기 디코딩된 비디오 데이터의 하나 이상의 컬러 성분들에 대한 최소 값 및 최대 값을 포함하고,

상기 방법은:

수신된 상기 최소 값 및 수신된 상기 최대 값에 기초하여 상기 파라미터들을 결정하는 단계를 더 포함하는, 비디오 데이터를 프로세싱하는 방법.

청구항 8

제 1 항에 있어서,

상기 정보는 상기 디코딩된 비디오 데이터의 하나 이상의 컬러 성분들에 대한 미리 정해진 범위의 샘플 값들에 대한 인덱스를 포함하고,

상기 방법은:

수신된 상기 인덱스에 기초하여 상기 디코딩된 비디오 데이터의 하나 이상의 컬러 성분들에 대한 최소 값 및 최대 값을 결정하는 단계; 및

결정된 상기 최소 값 및 결정된 상기 최대 값에 기초하여 상기 파라미터들을 결정하는 단계를 더 포함하는, 비디오 데이터를 프로세싱하는 방법.

청구항 9

제 1 항에 있어서,

상기 파라미터들이 사인 (sign) 되는지 또는 비사인 (unsigned) 되는지를 나타내는 신택스 엘리먼트를 수신하는 단계; 및

SEI 메시지에서 상기 정보에 대한 과정 프로세스를 수행하는 단계를 더 포함하고, 상기 과정 프로세스는 상기 신택스 엘리먼트의 값과 무관하게 동일한, 비디오 데이터를 프로세싱하는 방법.

청구항 10

제 1 항에 있어서,

상기 디코딩된 비디오 데이터에 대한 역 동적 범위 조정 프로세스를 수행한 후 상기 디코딩된 비디오 데이터를 디스플레이하는 단계를 더 포함하는, 비디오 데이터를 프로세싱하는 방법.

청구항 11

비디오 데이터를 프로세싱하도록 구성되는 장치로서,

디코딩된 비디오 데이터를 저장하도록 구성된 메모리; 및

하나 이상의 프로세서들을 포함하고,

상기 하나 이상의 프로세서들은:

역 동적 범위 조정 프로세스를 수행하기 위한 파라미터들을 결정하는 방법을 규정하는 정보를 포함하는 하나 이상의 신택스 엘리먼트들을 수신하는 것으로서, 상기 정보는 하나 이상의 컬러 성분에 대한 값들의 범위 파라미터를 결정하는데 이용되는 분수 비트들의 제 1 수, 스케일 파라미터를 결정하는데 이용되는 분수 비트들의 제 2 수, 및 오프셋 파라미터를 결정하는데 이용되는 분수 비트들의 제 3 수 중 하나 이상을 나타내는, 상기 하나 이상의 신택스 엘리먼트들을 수신하고;

디코딩된 비디오 데이터를 수신하고; 그리고

수신된 상기 정보에 따라 고정 소수점 컴퓨팅을 이용하여 상기 디코딩된 비디오 데이터에 대한 역 동적 범위 조정 프로세스를 수행하도록 구성되는, 비디오 데이터를 프로세싱하도록 구성되는 장치.

청구항 12

제 11 항에 있어서,

상기 하나 이상의 선택스 엘리먼트들을 수신하기 위해 상기 하나 이상의 프로세서들은 또한, 하나 이상의 보충 강화 정보 (SEI) 메시지들에서 하나 이상의 선택스 엘리먼트들을 수신하도록 구성되고, 상기 파라미터들은 상기 범위 파라미터, 상기 스케일 파라미터, 또는 상기 오프셋 파라미터 중 하나 이상을 포함하는, 비디오 데이터를 프로세싱하도록 구성되는 장치.

청구항 13

삭제

청구항 14

제 11 항에 있어서,

상기 하나 이상의 프로세서들은 또한:

상기 분수 비트들의 제 1 수, 상기 분수 비트들의 제 2 수 및 상기 분수 비트들의 제 3 수를 이용하여 상기 파라미터들을 결정하도록 구성되는, 비디오 데이터를 프로세싱하도록 구성되는 장치.

청구항 15

제 14 항에 있어서,

상기 파라미터들을 결정하기 위해, 상기 하나 이상의 프로세서들은 또한:

상기 파라미터들을 결정하는데 이용되는 임의의 중간 계산 프로세스를 동안의 임의의 분수 비트들을 누적하는 것에 의해 상기 분수 비트들의 제 1 수, 상기 분수 비트들의 제 2 수, 또는 상기 분수 비트들의 제 3 수 중 적어도 하나가 서로 상이한 경우에 상기 파라미터들을 결정하고; 그리고

미리 정해진 분수 정확도에 기초하여 상기 파라미터들을 결정하기 위한 최종 결과를 클립핑하도록 구성되는, 비디오 데이터를 프로세싱하도록 구성되는 장치.

청구항 16

제 14 항에 있어서,

상기 파라미터들을 결정하기 위해, 상기 하나 이상의 프로세서들은 또한:

상기 파라미터들을 결정하는데 이용되는 모든 중간 계산 프로세스를 동안 원하는 분수 정확도에 대해 임의의 분수 비트들을 절삭하는 것에 의해 상기 파라미터들을 결정하도록 구성되는, 비디오 데이터를 프로세싱하도록 구성되는 장치.

청구항 17

제 11 항에 있어서,

상기 정보는 상기 디코딩된 비디오 데이터의 하나 이상의 컬러 성분들에 대한 최소 값 및 최대 값을 포함하고,

상기 하나 이상의 프로세서들은 또한:

수신된 상기 최소 값 및 수신된 상기 최대 값에 기초하여 상기 파라미터들을 결정하도록 구성되는, 비디오 데이터를 프로세싱하도록 구성되는 장치.

청구항 18

제 11 항에 있어서,

상기 정보는 상기 디코딩된 비디오 데이터의 하나 이상의 컬러 성분들의 미리 정해진 범위의 샘플 값들에 대한 인덱스를 포함하고,

상기 하나 이상의 프로세서들은 또한:

수신된 상기 인덱스에 기초하여 상기 디코딩된 비디오 데이터의 하나 이상의 컬러 성분들에 대한 최소 값 및 최대 값을 결정하고; 그리고

결정된 상기 최소 값 및 결정된 상기 최대 값에 기초하여 상기 파라미터들을 결정하도록 구성되는, 비디오 데이터를 프로세싱하도록 구성되는 장치.

청구항 19

제 11 항에 있어서,

상기 하나 이상의 프로세서들은 또한:

상기 파라미터들이 사인되는지 또는 비사인되는지를 나타내는 선택스 엘리먼트를 수신하고; 그리고

SEI 메시지에서 상기 정보에 대한 파싱 프로세스를 수행하도록 구성되고, 상기 파싱 프로세스는 상기 선택스 엘리먼트의 값과 무관하게 동일한, 비디오 데이터를 프로세싱하도록 구성되는 장치.

청구항 20

제 11 항에 있어서,

상기 장치는,

상기 하나 이상의 프로세서들이 상기 디코딩된 비디오 데이터에 대한 역 동적 범위 조정 프로세스를 수행한 후 상기 디코딩된 비디오 데이터를 디스플레이하도록 구성되는 디스플레이를 더 포함하는, 비디오 데이터를 프로세싱하도록 구성되는 장치.

청구항 21

제 11 항에 있어서,

상기 장치는 카메라, 컴퓨터, 모바일 디바이스, 브로드캐스트 수신기 디바이스 또는 셋톱 박스 중 하나 이상을 포함하는, 비디오 데이터를 프로세싱하도록 구성되는 장치.

청구항 22

비디오 데이터를 프로세싱하도록 구성되는 장치로서,

역 동적 범위 조정 프로세스를 수행하기 위한 파라미터들을 결정하는 방법을 규정하는 정보를 포함하는 하나 이상의 선택스 엘리먼트들을 수신하기 위한 수단으로서, 상기 정보는 하나 이상의 컬러 성분에 대한 값들의 범위 파라미터를 결정하는데 이용되는 분수 비트들의 제 1 수, 스케일 파라미터를 결정하는데 이용되는 분수 비트들의 제 2 수, 및 오프셋 파라미터를 결정하는데 이용되는 분수 비트들의 제 3 수 중 하나 이상을 나타내는, 상기 하나 이상의 선택스 엘리먼트들을 수신하기 위한 수단;

디코딩된 비디오 데이터를 수신하기 위한 수단; 및

수신된 상기 정보에 따라 고정 소수점 컴퓨팅을 이용하여 상기 디코딩된 비디오 데이터에 대한 역 동적 범위 조정 프로세스를 수행하기 위한 수단을 포함하는, 비디오 데이터를 프로세싱하도록 구성되는 장치.

청구항 23

제 22 항에 있어서,

상기 하나 이상의 선택스 엘리먼트들을 수신하기 위한 수단은 하나 이상의 보충 강화 정보 (SEI) 메시지들에서 하나 이상의 선택스 엘리먼트들을 수신하기 위한 수단을 포함하고, 상기 파라미터들은 상기 범위 파라미터, 상기 스케일 파라미터, 또는 상기 오프셋 파라미터 중 하나 이상을 포함하는, 비디오 데이터를 프로세싱하도록 구성되는 장치.

청구항 24

삭제

청구항 25

제 22 항에 있어서,

상기 분수 비트들의 제 1 수, 상기 분수 비트들의 제 2 수 및 상기 분수 비트들의 제 3 수를 이용하여 상기 파라미터들을 결정하기 위한 수단을 더 포함하는, 비디오 데이터를 프로세싱하도록 구성되는 장치.

청구항 26

제 25 항에 있어서,

상기 파라미터들을 결정하기 위한 수단은:

상기 파라미터들을 결정하는데 이용되는 임의의 중간 계산 프로세스들 동안의 임의의 분수 비트들을 누적하는 것에 의해 상기 분수 비트들의 제 1 수, 상기 분수 비트들의 제 2 수, 또는 상기 분수 비트들의 제 3 수 중 적어도 하나가 서로 상이한 경우에 상기 파라미터들을 결정하기 위한 수단; 및

미리 정해진 분수 정확도에 기초하여 상기 파라미터들을 결정하기 위한 최종 결과를 클립핑하기 위한 수단을 더 포함하는, 비디오 데이터를 프로세싱하도록 구성되는 장치.

청구항 27

제 25 항에 있어서,

상기 파라미터들을 결정하기 위한 수단은:

상기 파라미터들을 결정하는데 이용되는 모든 중간 계산 프로세스들 동안 원하는 분수 정확도에 대해 임의의 분수 비트들을 절삭하는 것에 의해 상기 파라미터들을 결정하기 위한 수단을 더 포함하는, 비디오 데이터를 프로세싱하도록 구성되는 장치.

청구항 28

제 22 항에 있어서,

상기 정보는 상기 디코딩된 비디오 데이터의 하나 이상의 컬러 성분들에 대한 최소 값 및 최대 값을 포함하고,

상기 장치는:

수신된 상기 최소 값 및 수신된 상기 최대 값에 기초하여 상기 파라미터들을 결정하기 위한 수단을 더 포함하는, 비디오 데이터를 프로세싱하도록 구성되는 장치.

청구항 29

제 22 항에 있어서,

상기 정보는 상기 디코딩된 비디오 데이터의 하나 이상의 컬러 성분들 또는 미리 정해진 범위의 샘플 값들에 대한 인덱스를 포함하고,

상기 장치는:

수신된 상기 인덱스에 기초하여 상기 디코딩된 비디오 데이터의 하나 이상의 컬러 성분들에 대한 최소 값 및 최대 값을 결정하기 위한 수단; 및

결정된 상기 최소 값 및 결정된 상기 최대 값에 기초하여 상기 파라미터들을 결정하기 위한 수단을 더 포함하는, 비디오 데이터를 프로세싱하도록 구성되는 장치.

청구항 30

제 22 항에 있어서,

상기 파라미터들이 사인되는지 또는 비사인되는지를 나타내는 선택스 엘리먼트를 수신하기 위한 수단; 및

SEI 메시지에서 상기 정보에 대한 파싱 프로세스를 수행하기 위한 수단을 더 포함하고, 상기 파싱 프로세스는 상기 선택스 엘리먼트의 값과 무관하게 동일한, 비디오 데이터를 프로세싱하도록 구성되는 장치.

청구항 31

제 22 항에 있어서,

상기 디코딩된 비디오 데이터에 대한 역 동적 범위 조정 프로세스를 수행한 후 상기 디코딩된 비디오 데이터를 디스플레이하기 위한 수단을 더 포함하는, 비디오 데이터를 프로세싱하도록 구성되는 장치.

청구항 32

명령들을 저장한 컴퓨터 판독가능 저장 매체로서,

상기 명령들은 실행시, 비디오 데이터를 프로세싱하도록 구성되는 디바이스의 하나 이상의 프로세서들로 하여금,

역 동적 범위 조정 프로세스를 수행하기 위한 파라미터들을 결정하는 방법을 규정하는 정보를 포함하는 하나 이상의 선택스 엘리먼트들을 수신하게 하는 것으로서, 상기 정보는 하나 이상의 컬러 성분에 대한 값들의 범위 파라미터를 결정하는데 이용되는 분수 비트들의 제 1 수, 스케일 파라미터를 결정하는데 이용되는 분수 비트들의 제 2 수, 및 오프셋 파라미터를 결정하는데 이용되는 분수 비트들의 제 3 수 중 하나 이상을 나타내는, 상기 하나 이상의 선택스 엘리먼트들을 수신하게 하고;

디코딩된 비디오 데이터를 수신하게 하고; 그리고

수신된 상기 정보에 따라 고정 소수점 컴퓨팅을 이용하여 상기 디코딩된 비디오 데이터에 대한 역 동적 범위 조정 프로세스를 수행하게 하는, 명령들을 저장한 컴퓨터 판독가능 저장 매체.

청구항 33

제 32 항에 있어서,

상기 명령들은 또한, 상기 하나 이상의 프로세서들로 하여금 하나 이상의 보충 강화 정보 (SEI) 메시지들에서 하나 이상의 선택스 엘리먼트들을 수신하게 하고, 상기 파라미터들은 상기 범위 파라미터, 상기 스케일 파라미터, 또는 상기 오프셋 파라미터 중 하나 이상을 포함하는, 명령들을 저장한 컴퓨터 판독가능 저장 매체.

청구항 34

삭제

청구항 35

제 32 항에 있어서,

상기 명령들은 또한 상기 하나 이상의 프로세서들로 하여금:

상기 분수 비트들의 제 1 수, 상기 분수 비트들의 제 2 수 및 상기 분수 비트들의 제 3 수를 이용하여 상기 파라미터들을 결정하게 하는, 명령들을 저장한 컴퓨터 판독가능 저장 매체.

청구항 36

제 35 항에 있어서,

상기 명령들은 또한 상기 하나 이상의 프로세서들로 하여금:

상기 파라미터들을 결정하는데 이용되는 임의의 중간 계산 프로세스를 동안의 임의의 분수 비트들을 누적하는 것에 의해 상기 분수 비트들의 제 1 수, 상기 분수 비트들의 제 2 수, 또는 상기 분수 비트들의 제 3 수 중 적어도 하나가 서로 상이한 경우에 상기 파라미터들을 결정하게 하고; 그리고

미리 정해진 분수 정확도에 기초하여 상기 파라미터들을 결정하기 위한 최종 결과를 클립핑하게 하는, 명령들을 저장한 컴퓨터 판독가능 저장 매체.

청구항 37

제 35 항에 있어서,

상기 명령들은 또한 상기 하나 이상의 프로세서들로 하여금:

상기 파라미터들을 결정하는데 이용되는 모든 중간 계산 프로세스를 동안 원하는 분수 정확도에 대해 임의의 분수 비트들을 절삭하는 것에 의해 상기 파라미터들을 결정하게 하는, 명령들을 저장한 컴퓨터 판독가능 저장 매체.

청구항 38

제 32 항에 있어서,

상기 정보는 상기 디코딩된 비디오 데이터의 하나 이상의 컬러 성분들의 최소 값 및 최대 값을 포함하고,

상기 명령들은 또한 하나 이상의 프로세서들로 하여금:

수신된 상기 최소 값 및 수신된 상기 최대 값에 기초하여 상기 파라미터들을 결정하게 하는, 명령들을 저장한 컴퓨터 판독가능 저장 매체.

청구항 39

제 32 항에 있어서,

상기 정보는 상기 디코딩된 비디오 데이터의 하나 이상의 컬러 성분들에 대한 미리 정해진 범위의 샘플 값들에 대한 인덱스를 포함하고,

상기 명령들은 또한 상기 하나 이상의 프로세서들로 하여금:

수신된 상기 인덱스에 기초하여 상기 디코딩된 비디오 데이터의 하나 이상의 컬러 성분들에 대한 최소 값 및 최대 값을 결정하게 하고; 그리고

결정된 상기 최소 값 및 결정된 상기 최대 값에 기초하여 상기 파라미터들을 결정하게 하는, 명령들을 저장한 컴퓨터 판독가능 저장 매체.

청구항 40

제 32 항에 있어서,

상기 명령들은 또한 상기 하나 이상의 프로세서들로 하여금:

상기 파라미터들이 사인되는지 또는 비사인되는지를 나타내는 신팩스 엘리먼트를 수신하게 하고; 그리고

SEI 메시지에서 상기 정보에 대한 파싱 프로세스를 수행하게 하고, 상기 파싱 프로세스는 상기 신팩스 엘리먼트의 값과 무관하게 동일한, 명령들을 저장한 컴퓨터 판독가능 저장 매체.

청구항 41

제 32 항에 있어서,

상기 명령들은 또한 상기 하나 이상의 프로세서들로 하여금:

상기 디코딩된 비디오 데이터에 대한 역 동적 범위 조정 프로세스를 수행한 후 상기 디코딩된 비디오 데이터를 디스플레이하게 하는, 명령들을 저장한 컴퓨터 판독가능 저장 매체.

청구항 42

비디오 데이터를 프로세싱하는 방법으로서,

고정 소수점 컴퓨팅을 이용하여 비디오 데이터에 대한 동적 범위 조정 프로세스를 수행하는 단계; 및

상기 고정 소수점 컴퓨팅을 이용하여 상기 동적 범위 조정 프로세스에 대해 역 동적 범위 조정 프로세스를 수행하기 위한 파라미터들을 결정하는 방법을 규정하는 정보를 포함하는 하나 이상의 신팩스 엘리먼트들을 생성하는 단계를 포함하고,

상기 정보는 하나 이상의 컬러 성분에 대한 값들의 범위 파라미터를 결정하는데 이용되는 분수 비트들의 제 1 수, 스케일 파라미터를 결정하는데 이용되는 분수 비트들의 제 2 수, 및 오프셋 파라미터를 결정하는데 이용되

는 분수 비트들의 제 3 수 중 하나 이상을 나타내는, 비디오 데이터를 프로세싱하는 방법.

청구항 43

제 42 항에 있어서,

상기 하나 이상의 선택스 엘리먼트들을 생성하는 단계는 하나 이상의 보충 강화 정보 (SEI) 메시지들에서 하나 이상의 선택스 엘리먼트들을 생성하는 단계를 포함하고, 상기 파라미터들은 상기 범위 파라미터, 상기 스케일 파라미터, 또는 상기 오프셋 파라미터 중 하나 이상을 포함하는, 비디오 데이터를 프로세싱하는 방법.

청구항 44

삭제

청구항 45

제 42 항에 있어서,

상기 정보는 상기 비디오 데이터의 하나 이상의 컬러 성분들에 대한 최소 값 및 최대 값을 포함하는, 비디오 데이터를 프로세싱하는 방법.

청구항 46

제 42 항에 있어서,

상기 정보는 디코딩된 상기 비디오 데이터의 미리 정의된 범위의 샘플 값들에 대한 인덱스를 포함하는, 비디오 데이터를 프로세싱하는 방법.

청구항 47

제 42 항에 있어서,

카메라로 상기 비디오 데이터를 캡처하는 단계를 더 포함하는, 비디오 데이터를 프로세싱하는 방법.

청구항 48

비디오 데이터를 프로세싱하도록 구성되는 장치로서,

비디오 데이터를 저장하도록 구성되는 메모리; 및

하나 이상의 프로세서들을 포함하고,

상기 하나 이상의 프로세서들은:

고정 소수점 컴퓨팅을 이용하여 비디오 데이터에 대한 동적 범위 조정 프로세스를 수행하고; 그리고

상기 고정 소수점 컴퓨팅을 이용하여 상기 동적 범위 조정 프로세스에 대해 역 동적 범위 조정 프로세스를 수행하기 위한 파라미터들을 결정하는 방법을 규정하는 정보를 포함하는 하나 이상의 선택스 엘리먼트들을 생성하도록 구성되고,

상기 정보는 하나 이상의 컬러 성분에 대한 값들의 범위 파라미터를 결정하는데 이용되는 분수 비트들의 제 1 수, 스케일 파라미터를 결정하는데 이용되는 분수 비트들의 제 2 수, 및 오프셋 파라미터를 결정하는데 이용되는 분수 비트들의 제 3 수 중 하나 이상을 나타내는, 비디오 데이터를 프로세싱하도록 구성되는 장치.

청구항 49

제 48 항에 있어서,

상기 하나 이상의 프로세서들은 또한, 하나 이상의 보충 강화 정보 (SEI) 메시지들에서 하나 이상의 선택스 엘리먼트들을 생성하도록 구성되고, 상기 파라미터들은 상기 범위 파라미터, 상기 스케일 파라미터, 또는 상기 오프셋 파라미터 중 하나 이상을 포함하는, 비디오 데이터를 프로세싱하도록 구성되는 장치.

청구항 50

삭제

청구항 51

제 48 항에 있어서,

상기 정보는 상기 비디오 데이터의 하나 이상의 컬러 성분들에 대한 최소 값 및 최대 값을 포함하는, 비디오 데이터를 프로세싱하도록 구성되는 장치.

청구항 52

제 48 항에 있어서,

상기 정보는 디코딩된 상기 비디오 데이터의 미리 정의된 범위의 샘플 값들에 대한 인덱스를 포함하는, 비디오 데이터를 프로세싱하도록 구성되는 장치.

청구항 53

제 48 항에 있어서,

상기 장치는,

상기 비디오 데이터를 캡처하도록 구성되는 카메라를 더 포함하는, 비디오 데이터를 프로세싱하도록 구성되는 장치.

청구항 54

제 48 항에 있어서,

상기 장치는 카메라, 컴퓨터, 모바일 디바이스, 브로드캐스트 수신기 디바이스 또는 셋톱 박스 중 하나 이상을 포함하는, 비디오 데이터를 프로세싱하도록 구성되는 장치.

청구항 55

비디오 데이터를 프로세싱하도록 구성되는 장치로서,

고정 소수점 컴퓨팅을 이용하여 비디오 데이터에 대한 동적 범위 조정 프로세스를 수행하기 위한 수단; 및

상기 고정 소수점 컴퓨팅을 이용하여 상기 동적 범위 조정 프로세스에 대해 역 동적 범위 조정 프로세스를 수행하기 위한 파라미터들을 결정하는 방법을 규정하는 정보를 포함하는 하나 이상의 신택스 엘리먼트들을 생성하기 위한 수단을 포함하고,

상기 정보는 하나 이상의 컬러 성분에 대한 값들의 범위 파라미터를 결정하는데 이용되는 분수 비트들의 제 1 수, 스케일 파라미터를 결정하는데 이용되는 분수 비트들의 제 2 수, 및 오프셋 파라미터를 결정하는데 이용되는 분수 비트들의 제 3 수 중 하나 이상을 나타내는, 비디오 데이터를 프로세싱하도록 구성되는 장치.

청구항 56

명령들을 저장한 컴퓨터 판독가능 저장 매체로서,

상기 명령들은 실행시, 비디오 데이터를 프로세싱하도록 구성되는 디바이스의 하나 이상의 프로세서들로 하여금,

고정 소수점 컴퓨팅을 이용하여 비디오 데이터에 대한 동적 범위 조정 프로세스를 수행하게 하고; 그리고

상기 고정 소수점 컴퓨팅을 이용하여 상기 동적 범위 조정 프로세스에 대해 역 동적 범위 조정 프로세스를 수행하기 위한 파라미터들을 결정하는 방법을 규정하는 정보를 포함하는 하나 이상의 신택스 엘리먼트들을 생성하게 하고,

상기 정보는 하나 이상의 컬러 성분에 대한 값들의 범위 파라미터를 결정하는데 이용되는 분수 비트들의 제 1 수, 스케일 파라미터를 결정하는데 이용되는 분수 비트들의 제 2 수, 및 오프셋 파라미터를 결정하는데 이용되는 분수 비트들의 제 3 수 중 하나 이상을 나타내는, 명령들을 저장한 컴퓨터 판독가능 저장 매체.

발명의 설명

기술 분야

[0001] 본 출원은 2015년 9월 21일자로 출원된 미국 임시 출원 제62/221,586호와, 2015년 10월 2일자로 출원된 미국 임시 출원 제62/236,804호 및 2015년 10월 13일자로 출원된 미국 임시 출원 제62/241,063호를 우선권 주장하며, 이들 모두의 전체 내용은 참조로 본 명세서에 포함된다.

[0002] 본 개시물은 비디오 프로세싱에 관한 것이다.

배경 기술

[0003] 디지털 비디오 능력들은 디지털 텔레비전들, 디지털 직접 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인 정보 단말기들 (PDA들), 랩톱 또는 데스크톱 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, 이른바 "스마트 폰들", 비디오 원격회의 디바이스들, 비디오 스트리밍 디바이스들 등을 포함한 넓은 범위의 디바이스들에 통합될 수 있다. 디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263 또는 ITU-T H.264/MPEG-4, 파트 10, 고급 비디오 코딩 (AVC), ITU-H.265, 고 효율 비디오 코딩 (HEVC), 및 이러한 표준들의 확장본들과 같은 비디오 코딩 기법들을 구현한다. 비디오 디바이스들은 이러한 비디오 코딩 기법들을 구현함으로써 디지털 비디오 정보를 더 효율적으로 송신, 수신, 인코딩, 디코딩, 및/또는 저장할 수도 있다.

[0004] 비디오 코딩 기법들은 비디오 시퀀스들에 내재하는 리던던시를 감소시키거나 제거하기 위해 공간적 (픽처 내) 예측 및/또는 시간적 (픽처 간) 예측을 포함한다. 블록 기반 비디오 코딩의 경우, 비디오 슬라이스 (예컨대, 비디오 프레임 또는 비디오 프레임의 부분) 가 비디오 블록들로 구획화될 수도 있으며, 그 비디오 블록들은 트리블록 (treeblock) 들, 코딩 유닛 (coding unit, CU) 들 및/또는 코딩 노드들이라고 또한 지칭될 수도 있다. 픽처의 인트라 코딩식 (intra-coded, I) 슬라이스에서의 비디오 블록들은 동일한 픽처의 이웃하는 블록들에서의 참조 샘플들에 관한 공간적 예측을 사용하여 인코딩된다. 픽처의 인터 코딩식 (inter-coded, P 또는 B) 슬라이스에서의 비디오 블록들은 동일한 픽처의 이웃하는 블록들에서의 참조 샘플들에 관한 공간적 예측 또는 다른 참조 픽처들에서의 참조 샘플들에 관한 시간적 예측을 사용할 수도 있다. 픽처들은 프레임들이라고 지칭될 수도 있고, 참조 픽처들은 참조 프레임들이라고 지칭될 수도 있다.

[0005] 공간적 또는 시간적 예측은 코딩될 블록에 대한 예측성 블록 (predictive block) 이 생기게 한다. 잔차 데이터는 코딩될 원래의 블록과 예측성 블록 사이의 화소 차이들을 나타낸다. 인터 코딩식 블록이 예측성 블록을 형성하는 참조 샘플들의 블록을 가리키는 모션 벡터와, 코딩된 블록 및 예측성 블록 간의 차이를 나타내는 잔차 데이터에 따라 인코딩된다. 인트라 코딩식 블록이 인트라 코딩 모드 및 잔차 데이터에 따라 인코딩된다. 추가 압축을 위해, 잔차 데이터는 화소 도메인으로부터 변환 도메인으로 변환될 수도 있으며, 결과적으로 잔차 변환 계수들이 생겨나며, 그 계수들은 그 다음에 양자화될 수도 있다. 처음에 2차원 어레이로 배열된 양자화된 변환 계수들은, 변환 계수들의 1차원 벡터를 생성하기 위하여 스캐닝될 수도 있고, 엔트로피 코딩이 더 많은 압축을 달성하기 위해 적용될 수도 있다.

[0006] 캡처, 코딩, 및 디스플레이될 수도 있는 컬러 값들의 총 수는 색 영역 (color gamut) 에 의해 정의될 수도 있다. 색 영역이 디바이스가 캡처 (예컨대, 카메라) 또는 재현 (예컨대, 디스플레이) 할 수 있는 컬러들의 범위를 지칭한다. 종종, 색 영역들은 디바이스마다 상이하다. 비디오 코딩의 경우, 비디오 코딩 프로세스에서의 각각의 디바이스가 동일한 색 영역에서 화소 값들을 프로세싱하게 구성될 수도 있도록 비디오 데이터에 대한 미리 정의된 색 영역이 사용될 수도 있다. 일부 색 영역들은 비디오 코딩을 위해 전통적으로 사용된 색 영역들보다 더 큰 범위의 컬러들로 정의된다. 더 큰 범위의 컬러들을 갖는 이러한 색 영역들은 와이드 색 영역 (wide color gamut, WCG) 이라고 지칭될 수도 있다.

[0007] 비디오 데이터의 다른 양태가 동적 범위 (dynamic range) 이다. 동적 범위는 비디오 신호의 최대와 최소 밝기 (예컨대, 휘도) 사이의 비율로서 통상적으로 정의된다. 과거에 사용된 공통 비디오 데이터의 동적 범위는 표준 동적 범위 (standard dynamic range, SDR) 로 간주된다. 비디오 데이터에 대한 다른 예시적인 사양들은 최대 및 최소 밝기 사이에 더 큰 비율을 갖는 컬러 데이터를 정의한다. 이러한 비디오 데이터는 높은 동적 범위 (high dynamic range, HDR) 를 갖는 것으로서 설명될 수도 있다.

발명의 내용

- [0008] 본 개시물은 고정 소수점 구현을 이용하여 비디오 데이터의 성분들의 동적 범위 조정을 구현하기 위한 예시적 기술들 및 디바이스들을 설명한다. 설명된 기술들은 높은 동적 범위 (High Dynamic Range; HDR) 컨텐츠를 인코딩 및 디코딩하도록 구성된 H.264/AVC, H.265/HEVC 및 다른 표준들로 한정되지 않는 비디오 코딩 표준들에 적용가능하다.
- [0009] 본 개시물의 일 예에서, 비디오 데이터를 프로세싱하는 방법은 역 동적 범위 조정 프로세스를 수행하기 위한 파라미터들을 결정하는 방법을 규정하는 정보를 포함하는 하나 이상의 신팩스 엘리먼트들을 수신하는 단계, 디코딩된 비디오 데이터를 수신하는 단계, 및 수신된 정보에 따라 고정 소수점 컴퓨팅을 이용하여 디코딩된 비디오 데이터에 대한 역 동적 범위 조정 프로세스를 수행하는 단계를 포함한다.
- [0010] 본 개시물의 다른 양태에서, 비디오 데이터를 프로세싱하도록 구성된 장치는 디코딩된 비디오 데이터를 저장하도록 구성된 메모리, 및 역 동적 범위 조정 프로세스를 수행하기 위한 파라미터들을 결정하는 방법을 규정하는 정보를 포함하는 하나 이상의 신팩스 엘리먼트들을 수신하고, 디코딩된 비디오 데이터를 수신하고, 그리고 수신된 정보에 따라 고정 소수점 컴퓨팅을 이용하여 디코딩된 비디오 데이터에 대한 역 동적 범위 조정 프로세스를 수행하도록 구성된 하나 이상의 프로세서들을 포함한다.
- [0011] 본 개시물의 다른 예에서, 비디오 데이터를 프로세싱하도록 구성된 장치는 역 동적 범위 조정 프로세스를 수행하기 위한 파라미터들을 결정하는 방법을 규정하는 정보를 포함하는 하나 이상의 신팩스 엘리먼트들을 수신하기 위한 수단, 디코딩된 비디오 데이터를 수신하기 위한 수단, 및 수신된 정보에 따라 고정 소수점 컴퓨팅을 이용하여 디코딩된 비디오 데이터에 대한 역 동적 범위 조정 프로세스를 수행하기 위한 수단을 포함한다.
- [0012] 다른 예에서, 본 개시물은 명령들을 저장한 컴퓨터 판독가능 저장 매체를 설명하며, 명령들은 실행될 때 비디오 데이터를 프로세싱하도록 구성된 디바이스의 하나 이상의 프로세서들로 하여금, 역 동적 범위 조정 프로세스를 수행하기 위한 파라미터들을 결정하는 방법을 규정하는 정보를 포함하는 하나 이상의 신팩스 엘리먼트들을 수신하게 하고, 디코딩된 비디오 데이터를 수신하게 하고, 그리고 수신된 정보에 따라 고정 소수점 컴퓨팅을 이용하여 디코딩된 비디오 데이터에 대한 역 동적 범위 조정 프로세스를 수행하게 한다.
- [0013] 본 개시물의 다른 양태에서, 비디오 데이터를 프로세싱하는 방법은 고정 소수점 컴퓨팅을 이용하여 비디오 데이터에 대한 동적 범위 조정 프로세스를 수행하는 단계, 및 고정 소수점 컴퓨팅을 이용하여 동적 범위 조정 프로세스에 대해 역 동적 범위 조정 프로세스를 수행하기 위한 파라미터들을 결정하는 방법을 규정하는 정보를 포함하는 하나 이상의 신팩스 엘리먼트들을 생성하는 단계를 포함한다.
- [0014] 본 개시물의 다른 양태에서, 비디오 데이터를 프로세싱하도록 구성된 장치는 비디오 데이터를 저장하도록 구성된 메모리, 및 고정 소수점 컴퓨팅을 이용하여 비디오 데이터에 대한 동적 범위 조정 프로세스를 수행하는 단계, 및 고정 소수점 컴퓨팅을 이용하여 동적 범위 조정 프로세스에 대해 역 동적 범위 조정 프로세스를 수행하기 위한 파라미터들을 결정하는 방법을 규정하는 정보를 포함하는 하나 이상의 신팩스 엘리먼트들을 생성하도록 구성된 하나 이상의 프로세서들을 포함한다.
- [0015] 본 개시물의 다른 양태에서, 비디오 데이터를 프로세싱하도록 구성된 장치는 고정 소수점 컴퓨팅을 이용하여 비디오 데이터에 대한 동적 범위 조정 프로세스를 수행하기 위한 수단, 및 고정 소수점 컴퓨팅을 이용하여 동적 범위 조정 프로세스에 대해 역 동적 범위 조정 프로세스를 수행하기 위한 파라미터들을 결정하는 방법을 규정하는 정보를 포함하는 하나 이상의 신팩스 엘리먼트들을 생성하기 위한 수단을 포함한다.
- [0016] 다른 예에서, 본 개시물은 명령들을 저장한 컴퓨터 판독가능 저장 매체를 설명하며, 명령들은 실행될 때 비디오 데이터를 프로세싱하도록 구성된 디바이스의 하나 이상의 프로세서들로 하여금, 고정 소수점 컴퓨팅을 이용하여 비디오 데이터에 대한 동적 범위 조정 프로세스를 수행하게 하고, 그리고 고정 소수점 컴퓨팅을 이용하여 동적 범위 조정 프로세스에 대해 역 동적 범위 조정 프로세스를 수행하기 위한 파라미터들을 결정하는 방법을 규정하는 정보를 포함하는 하나 이상의 신팩스 엘리먼트들을 생성하게 한다.
- [0017] 하나 이상의 예들의 세부사항들은 첨부 도면들 및 다음의 설명에서 언급된다. 다른 특징들, 목적들, 및 장점들은 상세한 설명, 도면들, 및 청구항들로부터 명확하게 될 것이다.

도면의 간단한 설명

- [0018] 도 1은 본 개시물의 기법들을 구현하도록 구성될 수도 있는 예시적인 비디오 인코딩 및 디코딩 시스템을 도시하

는 블록도이다.

도 2는 컬러 리매핑 정보 (color remapping information, CRI) 프로세스의 전형적인 구조를 도시하는 개념도이다.

도 3은 HDR 데이터의 개념들을 예시하는 개념도이다.

도 4는 예시적인 색 영역들을 도시하는 개념도이다.

도 5는 HDR/WCG 표현 변환의 일 예를 도시하는 개념도이다.

도 6은 HDR/WCG 역 변환의 일 예를 도시하는 개념도이다.

도 7은 인지적으로 균일한 코드 레벨들로부터 선형 휘도로의 비디오 데이터 변환 (SDR 및 HDR을 포함함) 을 위해 이용되는 전기 광학 전달 함수들 (Electro-optical transfer functions) (EOTF) 의 예를 도시하는 개념도이다.

도 8은 본 개시물의 기법들에 따라 동작하는 예시적 HDR/WCG 변환 장치를 예시하는 블록도이다.

도 9는 본 개시물의 기법들에 따른 예시적 HDR/WCG 역 변환 장치를 예시하는 블록도이다.

도 10은 본 개시물의 기법들을 구현할 수도 있는 비디오 인코더의 일 예를 예시하는 블록도이다.

도 11은 본 개시물의 기법들을 구현할 수도 있는 비디오 디코더의 일 예를 예시하는 블록도이다.

도 12는 본 개시물의 일 예시적 비디오 프로세싱 기법을 도시하는 흐름도이다.

도 13은 본 개시물의 다른 예시적 비디오 프로세싱 기법을 도시하는 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0019]

본 개시물은 높은 동적 범위 (HDR) 및 와이드 색 영역 (WCG) 표현들을 갖는 비디오 데이터의 프로세싱 및/또는 코딩에 관련된다. 더 구체적으로, 본 개시물의 기법들은 (예를 들어, 부동 소수점 프로세싱 동작들과 상반되어) 고정 소수점 프로세싱 동작들을 이용하여 비디오 데이터 성분들의 범위 조정을 수행하기 위한 기법들을 포함한다. 본 명세서에서 설명되는 기법들 및 디바이스들은 HDR 및 WCG 비디오 데이터를 포함하는 비디오 데이터를 코딩하기 위해 이용되는 하이브리드 기반 비디오 코딩 시스템들 (예컨대, H.265/HEVC, H.264/AVC 등)의 압축 효율을 개선시킬 수도 있다.

[0020]

하이브리드 기반 비디오 코딩 표준들을 포함하는 비디오 코딩 표준들은 ITU-T H.261, ISO/IEC MPEG-1 비주얼, ITU-T H.262 또는 ISO/IEC MPEG-2 비주얼, ITU-T H.263, ISO/IEC MPEG-4 비주얼 그리고 스케일러블 비디오 코딩 (SVC) 및 멀티-뷰 비디오 코딩 (MVC) 확장본들을 포함한 ITU-T H.264 (또한 ISO/IEC MPEG-4 AVC로 알려짐) 를 포함한다. 새로운 비디오 코딩 표준, 즉 고 효율 비디오 코딩 (HEVC, 또한 H.265라 지칭됨) 의 설계는 ITU-T 비디오 코딩 전문가 그룹 (Video Coding Experts Group, VCEG) 및 ISO/IEC 동 화상 전문가 그룹 (Motion Picture Experts Group, MPEG) 의 JCT-VC (Joint Collaboration Team on Video Coding) 에 의해 완결되었다.

HEVC 규격 초안 10 (WD10) 이라고 지칭되는 HEVC 초안 사양서인, Bross et al., "High efficiency video coding (HEVC) text specification draft 10 (for FDIS & Last Call)", Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 12th Meeting: Geneva, CH, 14-23 January 2013, JCTVC-L1003v34는 http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v34.zip에서 입수 가능하다. 완결된 HEVC 표준은 HEVC 버전 1이라고 지칭된다.

[0021]

결합 보고서인 Wang et al., "High efficiency video coding (HEVC) Defect Report," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 14th Meeting: Vienna, AT, 25 July-2 August 2013, JCTVC-N1003v1은, http://phenix.int-evry.fr/jct/doc_end_user/documents/14_Vienna/wg11/JCTVC-N1003-v1.zip에서 입수 가능하다. 완결된 HEVC 표준 문서는 ITU-T H.265, Series H: Audiovisual and Multimedia Systems, Infrastructure of audiovisual services - Coding of moving video, High efficiency video coding, Telecommunication Standardization Sector of International Telecommunication Union (ITU), April 2013로서 게시되었고, 완결된 HEVC 표준의 다른 버전이 2014년 10월에 게시되었다. H.265/HEVC 규격 텍스트의 사본이 <http://www.itu.int/rec/T-REC-H.265-201504-I/en>으로부터 다운로드될 수도 있다.

[0022] 도 1은 본 개시물의 기법들을 이용할 수도 있는 예시적인 비디오 인코딩 및 디코딩 시스템 (10)을 도시하는 블록도이다. 도 1에 도시된 바와 같이, 시스템 (10)은 목적지 디바이스 (14)에 의해 나중에 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)로 제공한다. 소스 디바이스 (12)와 목적지 디바이스 (14)는 데스크톱 컴퓨터들, 노트북 (즉, 램프) 컴퓨터들, 태블릿 컴퓨터들, 세트톱 박스들, 이른바 "스마트" 폰들과 같은 전화기 핸드셋들, 이른바 "스마트" 패드들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 콘솔들, 비디오 스트리밍 디바이스들, 브로드캐스트 수신기 디바이스 등을 포함하는 매우 다양한 디바이스들 중 임의의 것을 포함할 수도 있다. 일부 경우들에서, 소스 디바이스 (12)와 목적지 디바이스 (14)는 무선 통신을 위해 갖추어질 수도 있다.

[0023] 목적지 디바이스 (14)는 디코딩될 인코딩된 비디오 데이터를 컴퓨터 판독가능 매체 (16)를 통해 수신할 수도 있다. 컴퓨터 판독가능 매체 (16)는 소스 디바이스 (12)로부터의 인코딩된 비디오 데이터를 목적지 디바이스 (14)로 이동시킬 수 있는 임의 유형의 매체 또는 디바이스를 포함할 수도 있다. 하나의 예에서, 컴퓨터 판독가능 매체 (16)는 소스 디바이스 (12)가 인코딩된 비디오 데이터를 목적지 디바이스 (14)로 직접 실시간으로 송신하는 것을 가능하게 하는 통신 매체를 포함할 수도 있다. 인코딩된 비디오 데이터는 통신 표준, 이를테면 유선 또는 무선 통신 프로토콜에 따라 변조되고 목적지 디바이스 (14)로 송신될 수도 있다. 통신 매체는 임의의 무선 또는 유선 통신 매체, 이를테면 무선 주파수 (RF) 스펙트럼 또는 하나 이상의 물리적 송신 라인들을 포함할 수도 있다. 통신 매체는 패킷 기반 네트워크, 이를테면 국부 영역 네트워크, 광역 네트워크, 또는 인터넷과 같은 글로벌 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.

[0024] 다른 예들에서, 컴퓨터 판독가능 매체 (16)는 비-일시적 저장 매체, 이를테면 하드 디스크, 플래시 드라이브, 콤팩트 디스크, 디지털 비디오 디스크, 블루레이 디스크, 또는 다른 컴퓨터-판독가능 매체를 포함할 수도 있다. 일부 예들에서, 네트워크 서버 (미도시)가 소스 디바이스 (12)로부터 인코딩된 비디오 데이터를 수신하고 그 인코딩된 비디오 데이터를, 예컨대, 네트워크 송신을 통해 목적지 디바이스 (14)에 제공할 수도 있다. 마찬가지로, 매체 생산 설비, 이를테면 디스크 스탠드 평 설비의 컴퓨팅 디바이스가, 소스 디바이스 (12)로부터 인코딩된 비디오 데이터를 수신하고 그 인코딩된 비디오 데이터를 포함하는 디스크를 생산할 수도 있다. 그 러므로, 컴퓨터 판독가능 매체 (16)는 다양한 예들에서, 다양한 형태들의 하나 이상의 컴퓨터 판독가능 매체들을 포함한다고 이해될 수도 있다.

[0025] 일부 예들에서, 인코딩된 데이터는 출력 인터페이스 (22)로부터 저장 디바이스로 출력될 수도 있다. 마찬가지로, 인코딩된 데이터는 입력 인터페이스에 의해 저장 디바이스로부터 액세스될 수도 있다. 저장 디바이스는 하드 드라이브, 블루-레이 디스크들, DVD들, CD-ROM들, 플래시 메모리, 휘발성 또는 비휘발성 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 임의의 다른 적합한 디지털 저장 매체들과 같은 다양한 분산형 또는 국부적으로 액세스되는 데이터 저장 매체들 중 임의의 것을 포함할 수도 있다. 추가의 예에서, 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 인코딩된 비디오 데이터를 목적지 디바이스 (14)로 송신할 수 있는 임의 유형의 서버일 수도 있다. 예시적인 파일 서버들은 웹 서버 (예컨대, 웹사이트용), FTP 서버, 네트워크 부속 스토리지 (network attached storage; NAS) 디바이스들, 또는 로컬 디스크 드라이브를 포함한다. 목적지 디바이스 (14)는 인터넷 접속을 포함하는 임의의 표준 데이터 접속을 통해 인코딩된 비디오 데이터에 액세스할 수도 있다. 이는 무선 채널 (예컨대, Wi-Fi 접속), 유선 접속 (예컨대, DSL, 케이블 모뎀 등), 또는 파일 서버 상에 저장된 인코딩된 비디오 데이터에 액세스하기에 적합한 양쪽 모두의 조합을 포함할 수도 있다. 저장 디바이스로부터의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 그 조합일 수도 있다.

[0026] 본 개시물의 기법들은 무선 애플리케이션들 또는 설정 (setting) 들로 반드시 제한되지는 않는다. 그 기법들은 다양한 멀티미디어 애플리케이션들 중 임의의 것, 이를테면 OTA (over-the-air) 텔레비전 브로드캐스트, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, 인터넷 스트리밍 비디오 송신들, 이를테면 HTTP를 통한 동적 적응적 스트리밍 (DASH), 데이터 저장 매체 상에 인코딩된 디지털 비디오, 데이터 저장 매체 상에 저장된 디지털 비디오의 디코딩, 또는 다른 애플리케이션들의 지원으로 비디오 코딩에 적용될 수도 있다. 일부 예들에서, 시스템 (10)은 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅, 및/또는 비디오폰 통화 (video

telephony) 와 같은 애플리케이션들을 지원하기 위해 단방향 또는 양방향 비디오 송신을 지원하도록 구성될 수도 있다.

[0027] 도 1 의 예에서, 소스 디바이스 (12) 는 비디오 소스 (18), 비디오 프리프로세서 부 (19) 와 비디오 인코더 (20) 를 포함하는 비디오 인코딩 유닛 (21), 및 출력 인터페이스 (22) 를 포함한다. 목적지 디바이스 (14) 는 입력 인터페이스 (28), 비디오 포스트프로세서 부 (31) 와 비디오 디코더 (30) 를 포함하는 비디오 디코딩 유닛 (29), 및 디스플레이 디바이스 (32) 를 포함한다. 본 개시물에 따르면, 소스 디바이스 (12) 의 비디오 프리프로세서 부 (19) 및/또는 비디오 인코더 (20), 및 목적지 디바이스 (14) 의 포스트프로세서 부 (31) 및/또는 비디오 디코더 (30) 는 고정 소수점 구현으로 HDR 및 WCG 비디오 데이터의 보다 효율적인 구현을 가능하게 하도록 특정 컬러 공간들에서 비디오 데이터에 적용된 관련 동작들 및 시그널링을 포함하는 본 개시물의 기법들을 구현하도록 구성될 수도 있다. 일부 예들에서, 비디오 프리프로세서 부 (19) 는 비디오 인코더 (20) 와 별개일 수도 있다. 다른 예들에서, 비디오 프리프로세서 부 (19) 는 비디오 인코더 (20) 의 부분일 수도 있다. 마찬가지로, 일부 예들에서, 비디오 포스트프로세서 부 (31) 는 비디오 디코더 (30) 와 별개일 수도 있다. 다른 예들에서, 비디오 포스트프로세서 부 (19) 는 비디오 디코더 (30) 의 부분일 수도 있다. 다른 예들에서, 소스 디바이스와 목적지 디바이스가 다른 성분들 또는 배열체들을 포함할 수도 있다. 예를 들어, 소스 디바이스 (12) 는 외부 비디오 소스 (18), 이를테면 외부 카메라로부터 비디오 데이터를 수신할 수도 있다. 마찬가지로, 목적지 디바이스 (14) 는, 통합형 디스플레이 디바이스를 포함하기 보다는, 외부 디스플레이 디바이스와 인터페이싱할 수도 있다.

[0028] 도 1의 예시된 시스템 (10) 은 단지 하나의 예일 뿐이다. HDR 및 WCG 비디오 데이터를 프로세싱하기 위한 기법들은 임의의 디지털 비디오 인코딩 및/또는 비디오 디코딩 디바이스에 의해 수행될 수도 있다. 또한, 본 개시물의 기법들은 비디오 프리프로세서 및/또는 포스트프로세서 (예를 들어, 비디오 프리프로세서 부 (19) 및 비디오 포스트 프로세서 부 (31)) 에 의해 또한 수행될 수도 있다. 일반적으로, 비디오 프리프로세서는 인코딩 전 (예를 들어, HEVC 인코딩 전) 비디오 데이터를 프로세싱하도록 구성된 임의의 디바이스일 수도 있다.

일반적으로 비디오 포스트 프로세서는 디코딩 후 (예를 들어, HEVC 디코딩 후) 비디오 데이터를 프로세싱하도록 구성된 임의의 디바이스일 수도 있다. 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 소스 디바이스 (12) 가 목적지 디바이스 (14) 로의 송신을 위한 코딩된 비디오 데이터를 생성하는 이러한 코딩 디바이스들의 단지 예들에 불과하다. 일부 예들에서, 디바이스들 (12, 14) 은 디바이스들 (12, 14) 의 각각이 비디오 인코딩 및 디코딩 성분들 뿐만 아니라 비디오 프리프로세서 및 비디오 포스트 프로세서 (예를 들어, 비디오 프리프로세서 부 (19) 및 비디오 포스트 프로세서 부 (31)) 를 포함하도록 실질적으로 대칭적인 방식으로 동작할 수도 있다. 따라서, 시스템 (10) 은, 예컨대, 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅, 또는 비디오폰 통화를 위해 비디오 디바이스들 (12, 14) 간에 단방향 또는 양방향 비디오 송신을 지원할 수도 있다.

[0029] 소스 디바이스 (12) 의 비디오 소스 (18) 는 비디오 캡처 디바이스, 이를테면 비디오 카메라, 이전에 캡처된 비디오를 포함하는 비디오 아카이브, 및/또는 비디오 콘텐츠 제공자로부터의 비디오를 수신하는 비디오 페드 인터페이스를 포함할 수도 있다. 추가의 대체예로서, 비디오 소스 (18) 는 컴퓨터 그래픽 기반 데이터를, 소스 비디오, 또는 라이브 비디오, 아카이브된 비디오, 및 컴퓨터 생성 비디오의 조합으로서 생성할 수도 있다. 일부 경우들에서, 비디오 소스 (18) 가 비디오 카메라이면, 소스 디바이스 (12) 와 목적지 디바이스 (14) 는 이른바 카메라 폰들 또는 비디오 폰들을 형성할 수도 있다. 그러나, 위에서 언급된 바와 같이, 본 개시물에서 설명되는 기법들은 대체로 비디오 코딩 및 비디오 프로세싱에 적용 가능할 수도 있고, 무선 및/또는 유선 애플리케이션들에 적용될 수도 있다. 각각의 경우에서, 캡처된, 사전-캡처된 (pre-captured), 또는 컴퓨터 생성된 비디오는 비디오 인코딩부 (21) 에 의해 인코딩될 수도 있다. 인코딩된 비디오 정보는 그러면 출력 인터페이스 (22) 에 의해 컴퓨터 판독가능 매체 (16) 상으로 출력될 수도 있다.

[0030] 목적지 디바이스 (14) 의 입력 인터페이스 (28) 는 컴퓨터 판독가능 매체 (16) 로부터 정보를 수신한다. 컴퓨터 판독가능 매체 (16) 의 정보는, 블록들 및 다른 코딩된 유닛들, 예컨대, 그룹 (groups of pictures GOP) 들의 특성들 및/또는 프로세싱을 기술하는 신택스 엘리먼트들을 포함하는, 비디오 디코딩부 (29) 에 의해 또한 사용되는, 비디오 인코더 (20) 에 의해 정의된 신택스 정보를 포함할 수도 있다. 디스플레이 디바이스 (32) 는 디코딩된 비디오 데이터를 사용자에게 디스플레이하고, 음극선관 (CRT), 액정 디스플레이 (LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (OLED) 디스플레이, 또는 다른 유형의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들 중 임의의 것을 포함할 수도 있다.

[0031] 예시된 바와 같이, 비디오 프리프로세서 부 (19) 는 비디오 소스 (18) 로부터 비디오 데이터를 수신한다. 비디오 프리프로세서 부 (19) 는 비디오 인코더 (20) 로 인코딩하기에 적합한 형태로 이를 변환하기 위해 비디

오 데이터를 프로세싱하도록 구성될 수도 있다. 예를 들어, 비디오 프리프로세서 부 (19)는 (예를 들어, 비선형 전달 함수를 이용한) 동적 범위 콤팩트화, 보다 콤팩트하거나 보다 견고한 컬러 공간으로 컬러 변환, 및/또는 부동 소수점-투-정수 표현 변환을 수행할 수도 있다. 비디오 인코더 (20)는 비디오 프리프로세서 부 (19)에 의해 출력된 비디오 데이터에 대한 비디오 인코딩을 수행할 수도 있다. 비디오 디코더 (30)는 비디오 데이터를 디코딩하도록 비디오 인코더 (20)의 역 동작을 수행할 수도 있고, 비디오 포스트 프로세서 (31)는 비디오 데이터를 디스플레이에 적합한 형태로 변환하기 위해 비디오 프리프로세서 부 (19)의 역을 수행할 수도 있다. 예를 들어, 비디오 포스트 프로세서 부 (31)는 정수-투-부동 소수점 변환, 콤팩트 또는 견고한 컬러 공간으로부터의 컬러 변환, 및/또는 디스플레이에 적합한 비디오 데이터를 생성하기 위한 동적 범위 콤팩트화의 역을 수행할 수도 있다.

[0032] 비디오 인코더 (20)와 비디오 디코더 (30) 각각은 다양한 적합한 인코더 회로, 이를테면 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들 (DSP들), 주문형 집적회로들 (ASIC들), 현장 프로그램가능 게이트 어레이들 (FPGA들), 개별 로직, 소프트웨어, 하드웨어, 펌웨어 또는 그것들의 임의의 조합 중 임의의 것으로서 구현될 수도 있다. 그 기법들이 소프트웨어에서 부분적으로 구현되는 경우, 디바이스가 본 개시물의 기법들을 수행하기 위해, 적합한 비일시적 컴퓨터 판독가능 매체 내에 소프트웨어에 대한 명령들을 저장하고 하나 이상의 프로세서들을 사용하여 하드웨어에서 그 명령들을 실행할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30)의 각각은 하나 이상의 인코더들 또는 디코더들 내에 구비될 수도 있고, 그것들 중 어느 하나는 결합형 인코더/디코더 (CODEC)의 부분으로서 각각의 디바이스 내에 통합될 수도 있다.

[0033] 비디오 프리프로세서부 (19)와 비디오 포스트프로세서부 (31) 각각은, 하나 이상의 마이크로프로세서들, DSP들, ASIC들, FPGA들, 개별 로직, 소프트웨어, 하드웨어, 펌웨어 또는 그것들의 임의의 조합과 같은 다양한 적합한 인코더 중 임의의 것으로서 구현될 수도 있다. 그 기법들이 소프트웨어에서 부분적으로 구현되는 경우, 디바이스가 본 개시물의 기법들을 수행하기 위해, 적합한 비일시적 컴퓨터 판독가능 매체 내에 소프트웨어에 대한 명령들을 저장하고 하나 이상의 프로세서들을 사용하여 하드웨어에서 그 명령들을 실행할 수도 있다. 위에서 논의된 바와 같이 비디오 프리프로세서부 (19)와 비디오 포스트프로세서부 (31)는 각각 비디오 인코더 (20)와 비디오 디코더 (30)와는 별개의 디바이스들일 수도 있다. 다른 예들에서, 비디오 프리프로세서 (19)는 비디오 인코더 (20)와 단일 디바이스로 통합될 수도 있고 비디오 포스트프로세서 (31)는 비디오 디코더 (30)와 단일 디바이스로 통합될 수도 있다.

[0034] 일부 예들에서, 비디오 인코더 (20)와 비디오 디코더 (30)는, ISO/IEC MPEG-4 비주얼 그리고 SVC (Scalable Video Coding) 확장본, MVC (Multi-view Video Coding) 확장본, MVC 기반 3차원 비디오 (3DV) 확장본을 포함한 ITU-T H.264 (또한 ISO/IEC MPEG-4 AVC로 알려짐)와 같은 비디오 압축 표준에 따라 동작한다. 일부 경우들에서, MVC 기반 3DV에 부합하는 임의의 비트스트림은 MVC 프로파일, 예컨대, 스테레오 하이 프로파일을 준수하는 서브-비트스트림을 항상 포함한다. 더욱이, H.264/AVC에 대한 3DV 코딩 확장본, 즉 AVC 기반 3DV를 생성하려는 지속적인 노력이 있다. 비디오 코딩 표준들의 다른 예들은 ITU-T H.261, ISO/IEC MPEG-1 비주얼, ITU-T H.262 또는 ISO/IEC MPEG-2 비주얼, ITU-T H.263, ISO/IEC MPEG-4 비주얼 그리고 ITU-T H.264, ISO/IEC 비주얼을 포함한다. 다른 예들에서, 비디오 인코더 (20)와 비디오 디코더 (30)는 HEVC 표준에 따라 동작하도록 구성될 수도 있다.

[0035] HEVC 및 다른 비디오 코딩 표준들에서, 비디오 시퀀스가 일련의 픽처들을 통상 포함한다. 픽처들은 "프레임들"이라고 또한 지칭될 수도 있다. 픽처가 S_L , S_{Cb} 및 S_{Cr} 로 표시되는 세 개의 샘플 어레이들을 포함할 수도 있다. S_L 은 루마 샘플들의 2차원 어레이 (즉, 블록)이다. S_{Cb} 는 Cb 색차 샘플들의 2차원 어레이이다. S_{Cr} 은 Cr 색차 샘플들의 2차원 어레이이다. 색차 샘플들은 본 명세서에서 "크로마" 샘플들이라고 또한 지칭될 수도 있다. 다른 사례들에서, 픽처가 모노크롬일 수도 있고 루마 샘플들의 어레이만을 포함할 수도 있다.

[0036] 비디오 인코더 (20)는 코딩 트리 유닛들 (CTU들)의 세트를 생성할 수도 있다. CTU들의 각각은, 루마 샘플들의 코딩 트리 블록, 크로마 샘플들의 두 개의 대응 코딩 트리 블록들, 및 코딩 트리 블록들의 샘플들을 코딩하는데 사용된 신택스 구조들을 포함할 수도 있다. 모노크롬 픽처 또는 별개의 세 개의 컬러 평면들을 갖는 픽처에서, CTU가 단일 코딩 트리 블록과 그 코딩 트리 블록의 샘플들을 코딩하는데 사용된 신택스 구조들을 포함할 수도 있다. 코딩 트리 블록이 샘플들의 NxN 블록일 수도 있다. CTU가 "트리 블록" 또는 "최대 코딩 유닛 (LCU)"이라고 또한 지칭될 수도 있다. HEVC의 CTU들은 다른 비디오 코딩 표준들, 이를테면 H.264/AVC의 매크로블록들과 대체로 유사할 수도 있다. 그러나, CTU가 특정 사이즈로 반드시 제한되는 것은

아니고 하나 이상의 코딩 유닛들 (CU들) 을 포함할 수도 있다. 슬라이스가 래스터 스캔으로 연속하여 순서화된 정수 수의 CTU들을 포함할 수도 있다.

[0037] 본 개시물은 하나 이상의 샘플 블록들과 하나 이상의 샘플 블록들의 샘플들을 코딩하는데 사용된 선택스 구조들을 지칭하기 위해 "비디오 유닛" 또는 "비디오 블록"이란 용어를 사용할 수도 있다. 예시적인 유형들의 비디오 유닛들은 HEVC에서의 CTU들, CU들, PU들, 변환 유닛들 (TU들), 또는 다른 비디오 코딩 표준들에서의 매크로블록들, 매크로블록 파티션들 등을 포함할 수도 있다.

[0038] 코딩된 CTU를 생성하기 위해, 비디오 인코더 (20) 는 CTU의 코딩 트리 블록들에 대해 퀼트트리 파티셔닝을 재귀적으로 수행하여 코딩 트리 블록들을 코딩 블록들로 나눌 수도 있으며, 그래서 그 이름이 "코딩 트리 유닛들"이다. 코딩 블록이 샘플들의 NxN 블록이다. CU가, 루마 샘플 어레이, Cb 샘플 어레이 및 Cr 샘플 어레이를 갖는 픽처의 루마 샘플들의 코딩 블록 및 크로마 샘플들의 두 개의 대응하는 코딩 블록들과, 그 코딩 블록들의 샘플들을 코딩하는데 사용된 선택스 구조들을 포함할 수도 있다. 모노크롬 픽처 또는 별개의 세 개의 컬러 평면들을 갖는 픽처에서, CU가 단일 코딩 블록과 그 코딩 블록의 샘플들을 코딩하는데 사용된 선택스 구조들을 포함할 수도 있다.

[0039] 비디오 인코더 (20) 는 CU의 코딩 블록을 하나 이상의 예측 블록 (prediction block) 들로 파티셔닝할 수도 있다. 예측 블록이 동일한 예측이 적용되는 샘플들의 직사각형 (즉, 정사각형 또는 정사각형 아닌) 블록일 수도 있다. CU의 예측 유닛 (PU) 이 픽처의 루마 샘플들의 예측 블록, 크로마 샘플들의 두 개의 대응하는 예측 블록들, 및 예측 블록 샘플들을 예측하는데 사용된 선택스 구조들을 포함할 수도 있다. 모노크롬 픽처 또는 별개의 세 개의 컬러 평면들을 갖는 픽처에서, PU가 단일 예측 블록과 그 예측 블록 샘플들을 예측하는데 사용된 선택스 구조들을 포함할 수도 있다. 비디오 인코더 (20) 는 CU의 각각의 PU의 루마, Cb 및 Cr 예측 블록들에 대한 예측성 루마, Cb 및 Cr 블록들을 생성할 수도 있다.

[0040] 비디오 인코더 (20) 는 PU에 대한 예측성 블록들을 생성하기 위해 인트라 예측 또는 인터 예측을 사용할 수도 있다. 비디오 인코더 (20) 가 PU의 예측성 블록들을 생성하기 위해 인트라 예측을 사용하면, 비디오 인코더 (20) 는 그 PU에 연관된 픽처의 디코딩된 샘플들에 기초하여 그 PU의 예측성 블록들을 생성할 수도 있다.

[0041] 비디오 인코더 (20) 가 PU의 예측성 블록들을 생성하기 위해 인터 예측을 사용하면, 비디오 인코더 (20) 는 그 PU에 연관된 픽처 이외의 하나 이상의 픽처들의 디코딩된 샘플들에 기초하여 그 PU의 예측성 블록들을 생성할 수도 있다. 인터 예측은 단방향 인터 예측 (즉, 단예측 (uni-prediction)) 또는 양방향 인터 예측 (즉, 양예측 (bi-prediction)) 일 수도 있다. 단예측 또는 양예측을 수행하기 위해, 비디오 인코더 (20) 는 현재 슬라이스에 대한 제 1 참조 픽처 리스트 (RefPicList0) 및 제 2 참조 픽처 리스트 (RefPicList1) 를 생성할 수도 있다.

[0042] 참조 픽처 리스트들의 각각은 하나 이상의 참조 픽처들을 포함할 수도 있다. 단예측을 사용할 때, 비디오 인코더 (20) 는 참조 픽처 내의 참조 로케이션을 결정하기 위해 RefPicList0 및 RefPicList1 중 어느 하나 또는 양쪽 모두에서 참조 픽처들을 검색할 수도 있다. 더욱이, 단예측을 사용할 때, 비디오 인코더 (20) 는, 참조 로케이션에 대응하는 샘플들에 적어도 부분적으로 기초하여, PU에 대한 예측성 샘플 블록들을 생성할 수도 있다. 더구나, 단예측을 사용할 때, 비디오 인코더 (20) 는 PU의 예측 블록과 참조 로케이션 사이의 공간적 변위를 나타내는 단일 모션 벡터를 생성할 수도 있다. PU의 예측 블록과 참조 로케이션 사이의 공간적 변위를 나타내기 위해, 모션 벡터가 PU의 예측 블록과 참조 로케이션 사이의 수평 변위를 특정하는 수평 성분을 포함할 수도 있고, PU의 예측 블록과 참조 로케이션 사이의 수직 변위를 특정하는 수직 성분을 포함할 수도 있다.

[0043] 양예측을 사용하여 PU를 인코딩할 때, 비디오 인코더 (20) 는 RefPicList0에서의 참조 픽처의 제 1 참조 로케이션과 RefPicList1에서의 참조 픽처의 제 2 참조 로케이션을 결정할 수도 있다. 비디오 인코더 (20) 는 그 다음에, 제 1 및 제 2 참조 로케이션들에 대응하는 샘플들에 적어도 부분적으로 기초하여, PU에 대한 예측성 블록들을 생성할 수도 있다. 더구나, 양예측을 사용하여 PU를 인코딩할 때, 비디오 인코더 (20) 는, PU의 샘플 블록과 제 1 참조 로케이션 사이의 공간적 변위를 나타내는 제 1 모션과 PU의 예측 블록과 제 2 참조 로케이션 사이의 공간적 변위를 나타내는 제 2 모션을 생성할 수도 있다.

[0044] 비디오 인코더 (20) 가 CU의 하나 이상의 PU들에 대한 예측성 루마, Cb, 및 Cr 블록들을 생성한 후, 비디오 인코더 (20) 는 그 CU에 대한 루마 잔차 블록을 생성할 수도 있다. CU의 루마 잔차 블록에서의 각각의 샘플은 CU의 예측성 루마 블록들 중 하나의 예측성 루마 블록에서의 루마 샘플과 CU의 원래의 루마 코딩 블록에서의 대응하는 샘플 간의 차이를 나타낸다. 덧붙여서, 비디오 인코더 (20) 는 CU에 대한 Cb 잔차 블록을 생성할 수

도 있다. CU의 Cb 잔차 블록에서의 각각의 샘플은 CU의 예측성 Cb 블록들 중 하나의 예측성 Cb 블록에서의 Cb 샘플과 CU의 원래의 Cb 코딩 블록에서의 대응 샘플 사이의 차이를 나타낼 수도 있다. 비디오 인코더 (20)는 CU에 대한 Cr 잔차 블록을 또한 생성할 수도 있다. CU의 Cr 잔차 블록에서의 각각의 샘플은 CU의 예측성 Cr 블록들 중 하나의 예측성 Cr 블록에서의 Cr 샘플과 CU의 원래의 Cr 코딩 블록에서의 대응 샘플 사이의 차이를 나타낼 수도 있다.

[0045] 더욱이, 비디오 인코더 (20)는 퀘드트리 파티셔닝을 사용하여 CU의 루마, Cb, 및 Cr 잔차 블록들을 하나 이상의 루마, Cb, 및 Cr 변환 블록들로 분해할 수도 있다. 변환 블록이 동일한 변환이 적용되는 샘플들의 직사각형 블록일 수도 있다. CU의 변환 유닛 (TU)이 루마 샘플들의 변환 블록, 크로마 샘플들의 두 개의 대응하는 변환 블록들, 및 그 변환 블록 샘플들을 변환하는데 사용된 십екс 구조들을 포함할 수도 있다. 모노크롬 꾹처 또는 별개의 세 개의 컬러 평면들을 갖는 꾹처에서, TU가 단일 변환 블록과 그 변환 블록 샘플들을 변환하는데 사용된 십екс 구조들을 포함할 수도 있다. 따라서, CU의 각각의 TU는 루마 변환 블록, Cb 변환 블록, 및 Cr 변환 블록에 연관될 수도 있다. TU에 연관된 루마 변환 블록은 CU의 루마 잔차 블록의 서브-블록일 수도 있다. Cb 변환 블록은 CU의 Cb 잔차 블록의 서브-블록일 수도 있다. Cr 변환 블록은 CU의 Cr 잔차 블록의 서브-블록일 수도 있다.

[0046] 비디오 인코더 (20)는 하나 이상의 변환들을 TU의 루마 변환 블록에 적용하여 그 TU에 대한 루마 계수 블록을 생성할 수도 있다. 계수 블록이 변환 계수들의 2차원 어레이일 수도 있다. 변환 계수가 스칼라 양일 수도 있다. 비디오 인코더 (20)는 하나 이상의 변환들을 TU의 Cb 변환 블록에 적용하여 TU에 대한 Cb 계수 블록을 생성할 수도 있다. 비디오 인코더 (20)는 하나 이상의 변환들을 TU의 Cr 변환 블록에 적용하여 TU에 대한 Cr 계수 블록을 생성할 수도 있다.

[0047] 계수 블록 (예컨대, 루마 계수 블록, Cb 계수 블록 또는 Cr 계수 블록)을 생성한 후, 비디오 인코더 (20)는 그 계수 블록을 양자화할 수도 있다. 양자화는 변환 계수들이 그 변환 계수들을 표현하는데 사용된 데이터의 양을 가능한 한 줄이도록 양자화되어서, 추가의 압축을 제공하는 프로세스를 일반적으로 지칭한다. 더욱이, 비디오 인코더 (20)는 변환 계수들을 역 양자화하고 꾹처의 CU들의 TU들의 변환 블록들을 복원하기 위하여 변환 계수들에 역 변환을 적용할 수도 있다. 비디오 인코더 (20)는 CU의 TU들의 복원된 변환 블록들과 CU의 PU들의 예측성 블록들을 사용하여 CU의 코딩 블록들을 복원할 수도 있다. 꾹처의 각각의 CU의 코딩 블록들을 복원함으로써, 비디오 인코더 (20)는 그 꾹처를 복원할 수도 있다. 비디오 인코더 (20)는 복원된 꾹처들을 디코딩된 꾹처 버퍼 (DPB)에 저장할 수도 있다. 비디오 인코더 (20)는 인터 예측 및 인트라 예측을 위해 DPB에서의 복원된 꾹처들을 사용할 수도 있다.

[0048] 비디오 인코더 (20)가 계수 블록을 양자화한 후, 비디오 인코더 (20)는 양자화된 변환 계수들을 나타내는 십екс 엘리먼트들을 엔트로피 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20)는 양자화된 변환 계수들을 나타내는 십екс 엘리먼트들에 대해 콘텍스트 적응 이진 산술 코딩 (CABAC)을 수행할 수도 있다. 비디오 인코더 (20)는 엔트로피 인코딩된 십екс 엘리먼트들을 비트스트림으로 출력할 수도 있다.

[0049] 비디오 인코더 (20)는 코딩된 꾹처들의 표현 및 연관된 데이터를 형성하는 비트들의 시퀀스를 포함하는 비트스트림을 출력할 수도 있다. 그 비트스트림은 네트워크 추상화 계층 (NAL) 유닛들의 시퀀스를 포함할 수도 있다. NAL 유닛들의 각각은 NAL 유닛 헤더를 포함하고 원시 바이트 시퀀스 패이로드 (RBSP)를 캡슐화한다. NAL 유닛 헤더는 NAL 유닛 유형 코드를 나타내는 십екс 엘리먼트를 포함할 수도 있다. NAL 유닛의 NAL 유닛 헤더에 의해 특정된 NAL 유닛 유형 코드는 NAL 유닛의 유형을 나타낸다. RBSP가 NAL 유닛 내에 캡슐화되는 정수 수의 바이트들을 포함하는 십екс 구조일 수도 있다. 일부 사례들에서, RBSP가 영 비트들을 포함한다.

[0050] 상이한 유형들의 NAL 유닛들이 상이한 유형들의 RBSP들을 캡슐화할 수도 있다. 예를 들어, 제 1 유형의 NAL 유닛은 꾹처 파라미터 세트 (PPS)에 대한 RBSP를 캡슐화할 수도 있으며, 제 2 유형의 NAL 유닛은 코딩된 슬라이스에 대한 RBSP를 캡슐화할 수도 있으며, 제 3 유형의 NAL 유닛은 보충적 향상 정보 (SEI)에 대한 RBSP를 캡슐화할 수도 있다는 등등이다. PPS가, 0 이상의 전체 코딩된 꾹처들에 적용되는 십екс 엘리먼트들을 포함할 수도 있는 십екс 구조이다. 비디오 코딩 데이터에 대한 RBSP들 (파라미터 세트들 및 SEI 메시지들에 대한 RBSP과는 대조적임)을 캡슐화하는 NAL 유닛들은, 비디오 코딩 계층 (VCL) NAL 유닛들이라고 지칭될 수도 있다. 코딩된 슬라이스를 캡슐화하는 NAL 유닛이 코딩된 슬라이스 NAL 유닛이라고 본원에서 지칭될 수도 있다. 코딩된 슬라이스에 대한 RBSP가 슬라이스 헤더와 슬라이스 데이터를 포함할 수도 있다.

[0051] 비디오 디코더 (30)는 비트스트림을 수신할 수도 있다. 덧붙여서, 비디오 디코더 (30)는 그 비트스트림으

로부터 신택스 엘리먼트들을 디코딩하기 위해 그 비트스트림을 파싱할 수도 있다. 비디오 디코더 (30)는 비트스트림으로부터 디코딩된 신택스 엘리먼트들에 적어도 부분적으로 기초하여 비디오 데이터의 픽처들을 복원할 수도 있다. 비디오 데이터를 복원하는 프로세스는 비디오 인코더 (20)에 의해 수행된 프로세스에 일반적으로 역일 수도 있다. 예를 들면, 비디오 디코더 (30)는 현재 CU의 PU들에 대한 예측성 블록들을 결정하기 위해 그 PU들의 모션 벡터들을 사용할 수도 있다. 비디오 디코더 (30)는 PU들의 모션 벡터 또는 모션 벡터들을 사용하여 PU들에 대한 예측성 블록들을 생성할 수도 있다.

[0052] 덧붙여서, 비디오 디코더 (30)는 현재 CU의 TU들에 연관된 계수 블록들을 역 양자화할 수도 있다. 비디오 디코더 (30)는 현재 CU의 TU들에 연관된 변환 블록들을 복원하기 위해 계수 블록들에 대해 역 변환들을 수행할 수도 있다. 비디오 디코더 (30)는 현재 CU의 PU들에 대한 예측성 샘플 블록들의 샘플들을 현재 CU의 TU들의 변환 블록들의 대응하는 샘플들에 가산함으로써 현재 CU의 코딩 블록들을 복원할 수도 있다. 픽처의 각각의 CU에 대한 코딩 블록들을 복원함으로써, 비디오 디코더 (30)는 그 픽처를 복원할 수도 있다. 비디오 디코더 (30)는 디코딩된 픽처들을 디코딩된 픽처 베퍼에 출력을 위해 그리고/또는 다른 픽처들을 디코딩함에 있어서의 사용을 위해 저장할 수도 있다.

[0053] 보충적 향상 정보 (SEI) 메시지들은 다수의 목적들을 위해 사용되고 있고, 통상적으로 디코더 (예를 들어, 비디오 디코더 (3))에 의해 비트스트림을 디코딩하기 위하여 필수적이지 않은 정보를 운반하기 위하여 비디오 비트스트림들에 포함될 수도 있다. 이 정보는 디코딩된 출력의 디스플레이 또는 프로세싱을 개선함에 있어서 유용할 수도 있으며; 예컨대, 이러한 정보는 콘텐츠의 가시성을 개선하기 위해 디코더 즉 엔티티들에 의해 사용될 수 있다. 품질에서의 개선이 애플리케이션 표준을 준수하는 모든 디바이스들에 가져와질 수 있도록 하는 그런 SEI 메시지들의 비트스트림에서의 존재 (예컨대, ETSI - TS 101 547-2, Digital Video Broadcasting (DVB) Plano-stereoscopic 3DTV; Part 2: Frame compatible plano-stereoscopic 3DTV에서 설명된 바와 같은, SEI 메시지가 비디오의 모든 프레임에 대해 운반되는 프레임 호환 평면-스테레오스코픽 3DTV 비디오 포맷을 위한 프레임 패킹 (packing) SEI 메시지의 캐리지, 예컨대, 3GPP TS 26.114 v13.0.0, 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; IP Multimedia Subsystem (IMS); Multimedia Telephony; Media handling and interaction (Release 13)에서 설명되는 바와 같은 복원점 SEI 메시지의 핸들링, 또는, 예컨대, ETSI - TS 101 154, Digital Video Broadcasting (DVB); Specification for the use of Video and Audio Coding in Broadcasting Applications based on the MPEG-2 Transport Stream에서 설명된 바와 같은 DVB에서의 팬-스캔 스캔 직사각형 SEI 메시지의 사용)를 특정한 애플리케이션 표준들이 요구할 수 있다는 것이 또한 가능하다.

[0054] 톤-매핑 정보 SEI 메시지는 하나의 SEI 메시지는, 루마 샘플들, 또는 RGB 성분 샘플들의 각각을 매핑하는데 사용된다. tone_map_id의 상이한 값들이 상이한 목적들을 정의하는데 사용되고, 톤-맵 SEI 메시지의 신택스는 그에 따라 또한 수정된다. 예를 들어, tone_map_id에 대한 1의 값이 SEI 메시지에 작용하는 프로세서가 RGB 샘플들을 최소 및 최대 값으로 클리핑하는 것을 허용한다. tone_map_id에 대한 3의 값이 루업 테이블이 폐벗 포인트들의 형태로 시그널링될 것을 허용하거나 또는 나타낸다. 그러나, 적용될 때, 동일한 값들은 모든 RGB 성분들에 적용되거나, 또는 루마 성분에만 적용된다.

[0055] 다른 예가 무릎 함수 (knee function) SEI 메시지인데, 이는 정규화된 선형 도메인에서의 디코딩된 픽처들의 RGB 성분들의 매핑을 나타내는데 사용된다. 입력 및 출력 최대 휘도 값들은 또한 나타내어지고, 루업 테이블은 입력 휘도 값들을 출력 휘도 값들에 매핑한다. 동일한 루업 테이블은 모든 세 개의 컬러 성분들에 적용될 수도 있다.

[0056] HEVC 표준에서 정의되는 컬러 리매핑 정보 (CRI) SEI 메시지는, 하나의 컬러 공간에서의 픽처들을 다른 색간으로 매핑하는데 사용되는 정보를 운반하는데 사용된다. 하나의 예에서, CRI SEI 메시지의 신택스는 세 개의 부분들, 즉, 제 1 루업 테이블 (프리-LUT) (1302), 뒤따르는 컬러 리매핑 계수들을 나타내는 3x3 매트릭스 (1304), 및 뒤따르는 제 2 루업 테이블 (포스트-LUT) (1306)을 포함한다. 각각의 컬러 성분, 예컨대, R,G,B 또는 Y,Cb,Cr에 대해, 독립적인 프리-LUT가 정의되고, 또한 독립적인 포스트-LUT가 정의된다. CRI SEI 메시지는 colour_remap_id라 불리는 신택스 엘리먼트를 또한 포함하는데, 그 신택스 엘리먼트의 상이한 값들은 SEI 메시지의 상이한 목적들을 나타내는데 사용될 수도 있다. 도 2는 CRI SEI 메시지에 의해 사용되는 컬러 리매핑 정보 프로세스의 전형적인 구조를 도시한다.

[0057] 동적 범위 조정 (DRA) SEI 메시지. 동적 범위 조정 SEI 메시지는, D. Bugdayci Sansli, A. K. Ramasubramonian, D. Rusanovskyy, S. Lee, J. Sole, M. Karczewicz, Dynamic range adjustment SEI message,

m36330, MPEG meeting, Warsaw, Poland, 22 - 26 June, 2015에서 설명되어 있다. 예시적인 DRA SEI 메시지가 입력 샘플들을 매핑하기 위한 스케일 및 오프셋 수들의 하나의 세트의 시그널링을 포함한다. SEI 메시지는 상이한 성분들에 대한 상이한 루업 테이블들의 시그널링을 또한 허용하고, 동일한 스케일 및 오프셋이 하나를 초과하는 성분을 위해 사용되는 것일 때 시그널링 최적화를 또한 허용한다. 스케일 및 오프셋 수들은 고정된 길이 정확도로 시그널링될 수도 있다.

[0058] 차세대 비디오 애플리케이션들은 HDR 및 WCG로 캡처된 경치를 표현하는 비디오 데이터로 동작할 것으로 예상된다. 이용되는 동적 범위 및 색 영역의 파라미터들은 비디오 콘텐츠의 두 개의 독립적인 속성들이고, 디지털 텔레비전 및 멀티미디어 서비스들의 목적들을 위한 그것들의 사양은 여러 국제 표준들에 의해 정의된다. 예를 들어, ITU-R Rec. BT.709, "Parameter values for the HDTV standards for production and international programme exchange", 및/또는 ITU-R Rec. BT.2020, "Parameter values for ultra-high definition television systems for production and international programme exchange"는, 표준 동적 범위 (SDR) 와 표준 색 영역을 넘어서 연장하는 원색들과 같은 HDTV (high definition television) 및 UHDTV (ultra-high definition television) 를 위한 파라미터들을 정의한다. Rec. BT.2100: Image parameter values for high dynamic range television for use in production and international programme exchange"는 와이드 색 영역 표현을 지원하는 원색들을 포함하는, HDR 텔레비전 사용을 위한 전달 함수들 및 표현들을 정의한다. 다른 시스템들에서의 동적 범위 및 색 영역 속성들을 특정하는 다른 표준 개발 조직 (SDO들) 문서들이 또한 있으며, 예컨대, DCI-P3 색 영역은 SMPTE-231-2 (Society of Motion Picture and Television Engineers) 에서 정의되고 HDR의 일부 파라미터들은 SMPTE-2084에서 정의된다. 비디오 데이터에 대한 동적 범위 및 색 영역의 간략한 설명이 아래에서 제공된다.

[0059] 동적 범위는 비디오 신호의 최대와 최소 밝기 (예컨대, 휙도) 사이의 비율로서 통상적으로 정의된다. 동적 범위는 'f-스톱'의 측면에서 또한 측정될 수도 있는데, 하나의 f-스톱은 신호의 동적 범위의 배가 (doubling)에 대응한다. MPEG의 정의에서, 16을 초과하는 f-스톱들로 밝기 변동을 특징화하는 콘텐츠가 HDR 콘텐츠라고 지칭된다. 일부 용어들에서, 10 f-스톱과 16 f-스톱 사이의 레벨들은 중간 동적 범위로서 간주되지만, 다른 정의들에서는 HDR로 간주된다. 본 개시물의 일부 예들에서, HDR 비디오 콘텐츠는 표준 동적 범위를 갖는 전통적으로 사용되는 비디오 콘텐츠 (예컨대, ITU-R Rec. BT.709에 의해 특정된 바와 같은 비디오 콘텐츠) 보다 더 높은 동적 범위를 갖는 임의의 비디오 콘텐츠일 수도 있다.

[0060] 인간 시각계 (HVS) 는 SDR 콘텐츠 및 HDR 콘텐츠보다 훨씬 더 큰 동적 범위들을 인지할 수도 있다. 그러나, HVS는 HVS의 동적 범위를 이른바 동시 범위 (simultaneous range) 로 좁히는 적응 메커니즘을 포함한다. 동시에 범위의 폭은 현재 조명 상태들 (예컨대, 현재 밝기)에 따라 달라질 수도 있다. HDTV의 SDR, UHDTV의 예상되는 HDR 및 HVS 동적 범위에 의해 제공되는 동적 범위의 시각화가 도 3에서 도시되지만, 정확한 범위는 각 개인과 디스플레이에 기초하여 가변할 수도 있다.

[0061] 현재 비디오 애플리케이션 및 서비스들은 ITU Rec.709에 의해 규제되고, 10 미만의 f-스톱으로 이어지는, 약 0.1 내지 100 칸텔라 (cd/m^2) (종종 "니트 (nit)"라고 지칭됨) 의 밝기 (예컨대, 휙도) 의 범위를 통상적으로 지원하는 SDR을 제공한다. 일부 예시적인 차세대 비디오 서비스들은 16 f-스톱까지의 동적 범위를 제공할 것이 예상된다. 비록 이러한 콘텐츠에 대한 상세한 사양들이 현재 개발 중이지만, 일부 초기 파라미터들은 SMPTE-2084 및 ITU-R Rec. 2020에서 특정되어 있다.

[0062] HDR 외에, 더욱 사실적인 비디오 경험을 위한 다른 양태가 컬러 차원이다. 컬러 차원은 색 영역에 의해 통상적으로 정의된다. 도 4는 SDR 색 영역 (BT.709 원색들에 기초한 삼각형 (100)) 과, UHDTV를 위한 더 넓은 색 영역 (BT.2020 원색들에 기초한 삼각형 (102)) 을 도시하는 개념도이다. 도 4는 자연색들의 한계들을 나타내는 이른바 스펙트럼 궤적 (locus) (햇바닥 형상 영역 (104) 으로 구분됨) 을 또한 묘사한다. 도 3에 의해 예시된 바와 같이, BT.709(삼각형 100) 으로부터 BT.2020(삼각형 102) 원색들로 이동하는 것은 약 70% 더 많은 컬러들에 대해 UHDTV 서비스들을 제공하는 것을 목표로 한다. D65는 BT.709 및/또는 BT.2020 사양들에 대한 예시적인 백색을 특정한다.

[0063]

DCI-P3, BT.709, 및 BT.2020 컬러 공간들에 대한 색 영역 사양들의 예들이 표 1에서 도시된다.

표 1 -색 영역 파라미터들

컬러 공간	RGB 컬러 공간 파라미터들							
	백색점		원색들					
	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

[0064]

표 1에서 알 수 있는 바와 같이, 색 영역이 백색점의 X 및 Y 값들에 의해, 그리고 원색들 (예컨대, 적색 (R), 녹색 (G), 및 청색 (B) 의 x 및 y 값들에 의해 정의될 수도 있다. x 및 y 값들은 CIE 1931 컬러 공간에 의해 정의된 바와 같이, 컬러들의 색도 (X 및 Z) 및 밝기 (Y)로부터 도출되는 정규화된 값들을 나타낸다. CIE 1931 컬러 공간은 순색들 (예컨대, 광장들의 측면에서) 사이의 링크들과 사람의 눈이 이러한 컬러들을 인지하는 방법을 정의한다.

[0066]

HDR/WCG 비디오 데이터는 4:4:4 크로마 포맷과 매우 넓은 컬러 공간 (예컨대, CIE XYZ)으로, 성분 당 매우 높은 정밀도 (심지어 부동소수점)로 보통 취득되고 저장된다. 이 표현은 고 정밀도를 목표로 하고 수학적으로 거의 무손실이다. 그러나, HDR/WCG 비디오 데이터를 저장하는 그런 포맷은 많은 리던던시들을 포함할 수도 있고 압축 목적들을 위한 최적이 아닐 수도 있다. HVS 기반 가정을 갖는 더 낮은 정밀도 포맷이 최신 비디오 애플리케이션들을 위해 통상적으로 이용된다.

[0067]

압축 목적을 위한 비디오 데이터 포맷 변환 프로세스의 일 예는 도 5에 도시된 바와 같이 3 개의 주요 프로세스들을 포함한다. 도 5의 기법들은 소스 디바이스 (12)에 의해 수행될 수도 있다. 선형 RGB 데이터 (110)는 HDR/WCG 비디오 데이터일 수도 있고, 부동 소수점 표현에 저장될 수도 있다. 선형 RGB 데이터 (110)는 동적 범위 콤팩트화를 위한 비선형 전달 함수 (transfer function; TF)(112)를 이용하여 콤팩트화될 수도 있다. 전달 함수 (112)는 SMPTE-2084에서 정의된 바와 같이 임의의 수의 비선형 전달 함수 예를 들어, PQ TF를 이용하여 선형 RGB 데이터 (110)를 콤팩트화할 수도 있다. 일부 예들에서, 컬러 변환 프로세스 (114)는 하이브리드 비디오 인코더에 의한 압축에 더욱 적합한 더 콤팩트하거나 또는 더 견고한 컬러 공간 (예를 들어, YUV 또는 YCrCb 컬러 공간)으로 콤팩트화된 데이터를 변환한다. 이 데이터는 그 후 변환된 HDR' 데이터 (118)를 생성하도록 부동 소수점-투-정수 표현 양자화 부 (116)를 이용하여 양자화된다. 이 예에서, HDR' 데이터 (118)는 정수 표현으로 된다. 이때, HDR' 데이터는 하이브리드 비디오 인코더 (예를 들어, HEVC 기법들을 적용하는 비디오 인코더 (20))에 의한 압축에 보다 적합한 포맷으로 된다. 도 5에 도시된 프로세스들의 출력은 주어진 예로서 주어지며, 다른 애플리케이션에서는 다를 수도 있다. 예를 들어, 컬러 변환은 TF 프로세스를 선택한다. 추가로, 추가적인 프로세싱, 예를 들어, 공간적 서브샘플링이 컬러 공간들에 적용될 수도 있다.

[0068]

디코더 측에서의 역 변환은 도 6에 도시된다. 도 6의 기법들은 목적지 디바이스 (14)에 의해 수행될 수도 있다. 변환된 HDR' 데이터 (120)는 하이브리드 비디오 디코더 (예를 들어, HEVC 기법들을 적용하는 비디오 디코더 (30))를 이용한 비디오 데이터의 디코딩을 통하여 목적지 디바이스 (14)에서 획득될 수도 있다. HDR' 데이터 (120)는 그 후, 역 양자화 부 (122)에 의해 역 양자화될 수도 있다. 그 후, 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 YCrCb 포맷으로부터 다시 RGB 포맷으로 HDR' 데이터를 변환할 수도 있다. 다음으로, 역 전달 함수 (126)가 선형 RGB 데이터 (128)를 재생성하도록 전달 함수 (112)에 의해 압축화되었던 동적 범위를 다시 추가하도록 데이터에 적용될 수도 있다.

[0069]

도 5에 묘사된 기법들이 이제 더 상세히 논의될 것이다. 대체로, 전달 함수는 양자화로 인한 에러들이 휘도

값들의 범위 전체에 걸쳐 (대략) 인지적으로 균일하도록 데이터의 동적 범위를 압축하기 위해 데이터 (예컨대, HDR/WCG 비디오 데이터)에 적용된다. 이러한 압축은 데이터가 더 적은 비트들로 표현되는 것을 허용한다.

하나의 예에서, 전달 함수는 1차원 (1D) 비선형 함수일 수도 있고, 예컨대, Rec. 709에서 SDR에 대해 특정된 바와 같이, 최종 사용자 디스플레이의 전기 광학 전달 함수 (EOTF)의 역을 반영할 수도 있다. 다른 예에서, 전달 함수는 HVS 인지를 밝기 변화들, 예컨대, HDR에 대해 SMPTE-2084에서 특정된 PQ 전달 함수로 근사화할 수도 있다. EOTF의 역 프로세스는 코드 레벨들을 휘도로 다시 매핑하는 EOTF (electro-optical transfer function)이다. 도 6은 EOTF들로서 사용되는 비선형 전달 함수의 여러 예들을 도시한다. 전달 함수들은 각각의 R, G 및 B 성분에 따로따로 또한 적용될 수도 있다.

[0070] 본 개시물의 맥락에서, "신호 값" 또는 "컬러 값"이라는 용어들은 이미지 엘리먼트에 대해 특정 컬러 성분 (이를테면 R, G, B, 또는 Y)의 값에 대응하는 휘도 레벨을 설명하는데 사용될 수도 있다. 신호 값은 선형 광레벨 (휘도 값)을 통상적으로 나타낸다. "코드 레벨" 또는 "디지털 코드 값"이란 용어들은 이미지 신호 값의 디지털 표현을 지칭할 수도 있다. 통상적으로, 그런 디지털 표현은 비선형 신호 값을 나타낸다. EOTF가 디스플레이 디바이스 (예컨대, 디스플레이 디바이스 (32))에 제공되는 비선형 신호 값들과 디스플레이 디바이스에 의해 생성되는 선형 컬러 값들 사이의 관계를 나타낸다.

[0071] RGB 데이터는 입력 컬러 공간으로서 통상적으로 이용되는데, RGB가 이미지 캡처링 센서들에 의해 통상적으로 생성되는 데이터의 유형이라서이다. 그러나, RGB 컬러 공간은 그것의 성분들 간에 높은 리던던시를 가지고 콤팩트 표현을 위한 최적은 아니다. 더욱 콤팩트하고 더욱 강건한 표현을 성취하기 위해, RGB 성분들은 압축에 더욱 적합한 더욱 비상관된 컬러 공간, 예컨대, YCbCr로 통상적으로 변환된다 (예컨대, 컬러 변환이 수행된다). YCbCr 컬러 공간이 밝기를 상이한 덜 상관된 성분들에서의 휘도 (Y) 및 컬러 정보 (CrCb)의 형태로 분리한다. 이 맥락에서, 강건한 표현이 제약된 비트레이트에서 압축될 때 더 높은 에러 내성 레벨들을 특징으로 하는 컬러 공간을 지칭할 수도 있다.

[0072] 컬러 변환을 뒤따라, 타겟 컬러 공간에서의 입력 데이터는 높은 비트-깊이 (예컨대, 부동소수점 정확도)로 여전히 표현될 수도 있다. 높은 비트-깊이 데이터는 타겟 비트-깊이로, 예를 들어, 양자화 프로세스를 사용하여 변환될 수도 있다. 특정한 연구들이 변환과 조합하는 10~12 비트 정확도가 16 f-스톱의 HDR 데이터를 JND (Just-Noticeable Difference) 미만의 왜곡으로 제공하기에 충분하다는 것을 보여준다. 대체로, JND이, 차이가 (예컨대, HVS에 의해) 현저해질 수 있도록 하기 위하여 무엇인가 (예컨대, 비디오 데이터)가 변화되어야만 하는 양이다. 10-비트 정확도로 표현되는 데이터는 대부분의 최신 비디오 코딩 해법들로 추가로 코딩될 수 있다. 이 양자화는 손실 코딩의 엘리먼트이고 변환된 데이터에 도입되는 부정확성의 소스이다.

[0073] 차세대 HDR/WCG 비디오 애플리케이션들이 HDR 및 CG의 상이한 파라미터들에서 캡처된 비디오 데이터로 동작할 것이라는 것이 예상된다. 상이한 구성들의 예들은 최대 1000 nit까지, 또는 최대 10,000 nit까지의 피크 밝기를 갖는 HDR 비디오 콘텐츠의 캡처일 수 있다. 상이한 색 영역의 예들은 BT.709, BT.2020 및 SMPTE 특정-P3, 또는 다른 것들을 포함할 수도 있다.

[0074] 모든 다른 현재 사용되는 색 영역을 포함 (하거나 또는 거의 포함) 하는 단일 컬러 공간, 예컨대, 타겟 컬러 컨테이너가 장래에 이용될 것이라는 것이 또한 예상된다. 그런 타겟 컬러 컨테이너의 하나의 예는 BT.2020이다. 단일 타겟 컬러 컨테이너의 지원이 HDR/WCG 시스템들의 표준화, 구현에 및 전개를 상당히 단순화시킬 것인데, 감소된 수의 동작 포인트들 (예컨대, 컬러 컨테이너들, 컬러 공간들, 컬러 변환 알고리즘들 등의 수) 및/또는 감소된 수의 요구된 알고리즘들이 디코더 (예컨대, 비디오 디코더 (30))에 의해 지원되어야 하여서이다.

[0075] 그런 시스템의 하나의 예에서, 타겟 컬러 컨테이너 (예컨대, BT.2020) 과는 상이한 네이티브 색 영역 (예컨대, P3 또는 BT.709)으로 캡처된 콘텐츠가 프로세싱에 앞서 (예컨대, 비디오 인코딩에 앞서) 타겟 컨테이너로 변환될 수도 있다. 이하는 이러한 변환의 여러 예들이다:

[0076] BT.709 컬러 컨테이너로부터 BT.2020로의 RGB 변환:

$$\text{o} \quad R_{2020} = 0.627404078626 * R_{709} + 0.329282097415 * G_{709} + 0.043313797587 * B_{709}$$

$$\text{o} \quad G_{2020} = 0.069097233123 * R_{709} + 0.919541035593 * G_{709} + 0.011361189924 * B_{709}$$

$$\text{o} \quad B_{2020} = 0.016391587664 * R_{709} + 0.088013255546 * G_{709} + 0.895595009604 * B_{709}$$

[0080] (1)

[0081] P3 으로부터 BT.2020 컬러 컨테이너로의 RGB 변환:

$$R_{2020} = 0.753832826496 * R_{P3} + 0.198597635641 * G_{P3} + 0.047569409186 * B_{P3}$$

$$G_{2020} = 0.045744636411 * R_{P3} + 0.941777687331 * G_{P3} + 0.012478735611 * B_{P3}$$

$$B_{2020} = -0.001210377285 * R_{P3} + 0.017601107390 * G_{P3} + 0.983608137835 * B_{P3}$$

[0085] (2)

[0086] 이 변환 동안, P3 또는 BT.709 색 영역에서 캡처된 신호의 각각의 성분 (예를 들어, RGB, YUV, YCrCb 등) 이 차지하는 값 범위는 BT.2020 표현에서 감소될 수도 있다. 데이터가 부동소수점 정확도로 표현되므로, 손실은 없지만, 컬러 변환 (예컨대, 아래의 식 3에 도시된 RGB로부터 YCrCB로의 변환) 과 양자화 (아래의 식 4에서의 예) 과 조합될 때, 값 범위의 축소는 입력 데이터에 대한 증가된 양자화 에러로 이어진다.

$$Cb = \frac{B' - Y'}{1.8814}; Cr = \frac{R' - Y'}{1.4746}$$

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

[0088] (3)

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

[0089] o

$$D_{Cb} = \left(\text{Round} \left((1 \ll (\text{BitDepth}_{Cr} - 8)) * (224 * Cb + 128) \right) \right)$$

[0090] o

$$D_{Cr} = \left(\text{Round} \left((1 \ll (\text{BitDepth}_{Cb} - 8)) * (224 * Cr + 128) \right) \right)$$

[0091] o

[0092] (4)

[0093] 식 (4)에서 $D_{Y'}$ 은 양자화된 Y' 성분이며, D_{Cb} 는 양자화된 Cb 이고 D_{Cr} 은 양자화된 Cr 성분이다. 표현 \ll 는 비트 단위 (bit-wise) 우측 시프트를 나타낸다. BitDepth_Y , BitDepth_{Cr} , 및 BitDepth_{Cb} 는 양자화된 성분들의 원하는 비트 깊이들이다.

[0094] 덧붙여서, 실-세계 코딩 시스템에서, 감소된 동적 범위로 신호를 코딩하는 것은 코딩된 크로마 성분들에 대한 상당한 정확도 손실로 이어질 수도 있고 코딩 아티팩트들, 예컨대, 컬러 불일치 및/또는 컬러 번짐 (bleeding)으로서 관찰자에 의해 관찰될 것이다.

[0095] 위에 설명된 문제들을 해결하기 위하여, 다음 기법들이 고려될 수도 있다. 일 예시적 기법은 네이티브 컬러 공간에서의 HDR 코딩을 포함한다. 이러한 기법에서, HDR 비디오 코딩 시스템은 현재 알려진 컬러 영역의 여러 유형들을 지원하고 장래 컬러 영역들을 지원하도록 비디오 코딩 표준의 확장들을 허용한다. 이는 상이한 컬러 컨버전 변환, 예를 들어, RGB 내지 YCbCr 로의 변환 및 이들의 역 변환들을 지원하도록 한정될 뿐만 아니라 컬러 영역들 각각으로 조정되는 변환 함수들을 규정한다. 이러한 여러 툴들의 지원은 복잡하고 고가이다.

[0096] 다른 예시적인 기법은 색 영역 인식 비디오 코덱을 포함한다. 그런 기법에서, 가상적인 비디오 인코더가 입력 신호의 네이티브 색 영역을 추정하고 감소된 동적 범위로부터 초래되는 임의의 왜곡을 감소시키기 위해 코딩 파라미터들 (예컨대, 코딩된 크로마 성분들에 대한 양자화 파라미터들) 을 조정하도록 구성된다. 그러나, 그런 기법은 정확도 손실을 복원할 수 없을 것이며, 이는 위의 식 (4)로 수행된 양자화로 인해 일어날 수도 있는데, 모든 입력 데이터가 전형적인 코덱에 정수 포인트 정확도로 제공되어서이다.

[0097] 본 개시물은 색 영역 변환에 의해 HDR 신호 표현들에 도입되는 동적 범위 변화들을 보상하기 위해 동적 범위 조정 (DRA) 을 수행하는 기법들, 방법들, 및 장치들을 설명한다. 동적 범위 조정은 컬러 불일치, 컬러 번짐 등을 포함하는, 색 영역 변환에 의해 초래되는 임의의 왜곡을 방지하는 그리고/또는 줄이는 것을 도울 수도 있

다. 본 개시물의 하나 이상의 예들에서, DRA는 인코더 측의 (예컨대, 소스 디바이스 (12)에 의한) 양자화에 앞서 그리고 디코더 측의 (예컨대, 목적지 디바이스 (14)에 의한) 역 양자화 후에 타겟 컬러 공간, 예컨대, YCbCr의 각각의 컬러 성분의 값들에 대해 수행된다.

[0098] 도 8은 본 개시물의 기법들에 따라 동작하는 예시적인 HDR/WCG 변환 장치를 도시하는 블록도이다. 도 8에서, 실선들은 데이터 흐름을 특정하고 파선들은 제어 신호들을 특정한다. 본 개시물의 기법들은 소스 디바이스 (12)의 비디오 프리프로세서부 (19)에 의해 수행될 수도 있다. 위에서 논의된 바와 같이, 비디오 프리프로세서부 (19)는 비디오 인코더 (20)와는 별개의 디바이스일 수도 있다. 다른 예들에서, 비디오 프리프로세서부 (19)는 비디오 인코더 (20)와는 동일한 디바이스에 통합될 수도 있다.

[0099] 도 8에 도시된 바와 같이, RGB 네이티브 CG 비디오 데이터 (200)가 비디오 프리프로세서부 (19)에 입력된다. 비디오 프리프로세서부 (19)에 의한 비디오 프리프로세싱의 맥락에서, RGB 네이티브 CG 비디오 데이터 (200)가 입력 컬러 컨테이너에 의해 정의된다. 입력 컬러 컨테이너는 비디오 데이터 (200)를 표현하는데 사용되는 원색들의 세트 (예컨대, BT. 709, BT. 2020, P3 등)를 특정한다. 본 개시물의 하나의 예에서, 비디오 프리프로세서부 (19)는 RGB 네이티브 CB 비디오 데이터 (200)의 컬러 컨테이너 및 컬러 공간 둘 다를 HDR' 데이터 (216)를 위한 타겟 컬러 컨테이너 및 타겟 컬러 공간으로 변환하도록 구성될 수도 있다. 입력 컬러 컨테이너처럼, 타겟 컬러 컨테이너는 HDR' 데이터 (216)를 표현하는데 사용되는 세트 또는 원색들을 특정할 수도 있다. 본 개시물의 하나의 예에서, RGB 네이티브 CB 비디오 데이터 (200)는 HDR/WCG 비디오일 수도 있고, BT.2020 또는 P3 컬러 컨테이너 (또는 임의의 WCG)를 가질 수도 있고, RGB 컬러 공간에 있을 수도 있다. 다른 예에서, RGB 네이티브 CB 비디오 데이터 (200)는 SDR 비디오일 수도 있고, BT.709 컬러 컨테이너를 가질 수도 있다. 하나의 예에서, HDR' 데이터 (216)를 위한 타겟 컬러 컨테이너는 HDR/WCG 비디오 (예컨대, BT.2020 컬러 컨테이너)를 위해 구성되었을 수도 있고 비디오 인코딩을 위한 더욱 최적인 컬러 공간 (예컨대, YCrCb)을 사용할 수도 있다.

[0100] 본 개시물의 하나의 예에서, CG 변환기 (202)는 RGB 네이티브 CG 비디오 데이터 (200)의 컬러 컨테이너를 입력 컬러 컨테이너 (예컨대, 제 1 컬러 컨테이너)로부터 타겟 컬러 컨테이너 (예컨대, 제 2 컬러 컨테이너)로 변환하도록 구성될 수도 있다. 하나의 예로서, CG 변환기 (202)는 RGB 네이티브 CG 비디오 데이터 (200)를 BT.709 컬러 표현으로부터 BT.2020 컬러 표현으로 변환할 수도 있으며, 그 예가 아래에서 도시된다.

[0101] RGB BT.709 샘플들 (R_{709} , G_{709} , B_{709}) 을 RGB BT.2020 샘플들 (R_{2020} , G_{2020} , B_{2020})로 변환하는 프로세스는 먼저 XYZ 표현으로의 변환과, 뒤따르는 적절한 변환 매트릭스들을 사용한 XYZ로부터 RGB BT.2020로의 변환을 수반하는 2-단계 변환으로 구현될 수 있다.

$$X = 0.412391 * R_{709} + 0.357584 * G_{709} + 0.180481 * B_{709}$$

$$Y = 0.212639 * R_{709} + 0.715169 * G_{709} + 0.072192 * B_{709} \quad (5)$$

$$Z = 0.019331 * R_{709} + 0.119195 * G_{709} + 0.950532 * B_{709}$$

[0105] XYZ로부터 $R_{2020}G_{2020}B_{2020}$ (BT.2020)로의 변환

$$R_{2020} = \text{clipRGB}(1.716651 * X - 0.355671 * Y - 0.253366 * Z)$$

$$G_{2020} = \text{clipRGB}(-0.666684 * X + 1.616481 * Y + 0.015768 * Z) \quad (6)$$

$$B_{2020} = \text{clipRGB}(0.017640 * X - 0.042771 * Y + 0.942103 * Z)$$

[0109] 마찬가지로, 단일 단계 및 권장된 방법은 다음과 같다:

$$R_{2020} = \text{clipRGB}(0.627404078626 * R_{709} + 0.329282097415 * G_{709} + 0.043313797587 * B_{709})$$

$$G_{2020} = \text{clipRGB}(0.069097233123 * R_{709} + 0.919541035593 * G_{709} + 0.011361189924 * B_{709}) \quad (7)$$

$$B_{2020} = \text{clipRGB}(0.016391587664 * R_{709} + 0.088013255546 * G_{709} + 0.895595009604 * B_{709})$$

[0113] CG 변환 후의 결과적인 비디오 데이터는 도 8에서 RGB 타겟 CG 비디오 데이터 (204)로서 도시된다. 본 개

시물의 다른 예들에서, 입력 데이터 및 출력 HDR' 데이터를 위한 컬러 컨테이너는 동일할 수도 있다. 이러한 일 예에서, CG 변환기 (202) 는 RGB 네이티브 CG 비디오 데이터 (200) 에 대해 임의의 변환을 수행할 필요가 없다.

[0114] 다음으로, 전달 함수 부 (206) 는 RGB 타겟 CG 비디오 데이터 (204) 의 동적 범위를 압축한다. 전달 함수 부 (206) 는 도 5를 참조하여 위에서 논의된 바와 동일한 방식으로 동적 범위를 압축하기 위해 전달 함수를 적용하도록 구성될 수도 있다. 컬러 변환 부 (208) 는 RGB 타겟 CG 컬러 데이터 (204) 를 입력 컬러 컨테이너의 컬러 공간 (예컨대, RGB) 으로부터 타겟 컬러 컨테이너의 컬러 공간 (예컨대, YCrCb) 으로 변환한다. 도 5를 참조하여 위에서 설명된 바와 같이, 컬러 변환 부 (208) 는 압축된 데이터를 하이브리드 비디오 인코더 (예컨대, 비디오 인코더 (20)) 에 의한 압축에 더욱 적합한 더욱 콤팩트하거나 또는 강건한 컬러 공간 (예컨대, YUV 또는 YCrCb 컬러 공간) 으로 변환한다.

[0115] 조정 부 (210) 는 DRA 파라미터들 추정 부 (212) 에 의해 도출된 DRA 파라미터들에 따라서 컬러 변환된 비디오 데이터의 동적 범위 조정 (DRA) 을 수행하도록 구성된다. 대체로, CG 변환기 (202) 에 의한 CG 변환 및 전달 함수 부 (206) 에 의한 동적 범위 압축 후, 결과적인 비디오 데이터의 실제 컬러 값들은 특정 타겟 컬러 컨테이너의 색 영역에 할당된 모든 이용가능 코드워드들 (예컨대, 각각의 컬러를 표현하는 고유 비트 시퀀스들) 을 사용하지 않을 수도 있다. 다시 말하면, 일부 환경들에서, 입력 컬러 컨테이너로부터 출력 컬러 컨테이너로의 RGB 네이티브 CG 비디오 데이터 (200) 의 변환은 결과적인 압축된 비디오 데이터가 모든 가능한 컬러 표현들의 효율적인 사용을 하지 못하게 하도록 비디오 데이터의 컬러 값들 (예컨대, Cr 및 Cb) 을 과도하게 압축 할 수도 있다. 위에서 설명된 바와 같이, 컬러들에 대한 값들의 감소된 범위를 이용하여 신호를 코딩하는 것은, 코딩된 크로마 성분들에 대한 상당한 정확도 손실로 이어질 수도 있고 코딩 아티팩트들, 예컨대, 컬러 불일치 및/또는 컬러 번짐으로서 관람자에 의해 관찰될 것이다.

[0116] 조정 부 (210) 는 특정 타겟 컬러 컨테이너에 대해 이용 가능한 코드워드들의 전체를 사용하기 위해 동적 범위 압축 및 컬러 변환 후 비디오 데이터, 예컨대, RGB 타겟 CG 비디오 데이터 (204) 의 컬러 성분들 (예컨대, YCrCb) 에 DRA 파라미터들을 적용하도록 구성될 수도 있다. 조정 부 (210) 는 DRA 파라미터를 화소 레벨에서 비디오 데이터에 적용할 수도 있다. 대체로, DRA 파라미터들은 실제 비디오 데이터를 표현하는데 사용되는 코드워드들을 가능한 타겟 컬러 컨테이너에 대해 이용 가능한 가능한 한 많은 코드워드들로 확장하는 함수를 정의한다.

[0117] 본 개시물의 하나의 예에서, DRA 파라미터들은 비디오 데이터의 성분들에 적용되는 스케일 및 오프셋 값을 포함한다. 대체로, 비디오 데이터의 컬러 성분들의 값 범위가 더 낮을수록, 더 큰 스케일링 계수가 사용될 수도 있다. 오프셋 파라미터는 타겟 컬러 컨테이너를 위한 이용가능 코드워드들의 중심에 컬러 성분들의 값들을 중심 맞추는데 사용될 수도 있다. 예를 들어, 타겟 컬러 컨테이너가 컬러 성분 당 1024 개 코드워드들을 포함한다면, 중심 코드워드가 코드워드 (512) (예컨대, 가장 중앙의 코드워드) 로 이동되도록 오프셋 값이 선택될 수도 있다. 다른 예들에서, 오프셋 파라미터는 타겟 컬러 컨테이너에서의 전체 표현이 코딩 아티팩트들을 방지함에 있어서 더욱 효율적이다. 더욱 코드워드들의 출력 코드워드들로의 더 나은 매핑을 제공하는데 사용될 수도 있다.

[0118] 하나의 예에서, 조정 부 (210) 는 DRA 파라미터들을 타겟 컬러 공간 (예컨대, YCrCb) 에서의 비디오 데이터에 다음과 같이 적용하며:

$$Y'' = scale1 * Y' + offset1$$

$$C_b'' = scale2 * C_b' + offset2 \quad (8)$$

$$Cr'' = scale3 * Cr' + offset3$$

[0122] 여기서 신호 성분들 (Y' , C_b' 및 Cr') 은 RGB로부터 YCbCr로의 변환 (식 3의 예) 으로부터 생성된 신호이다. Y' , Cr' 및 Cr' 은 또한 비디오 디코더 (30) 에 의해 디코딩된 비디오 신호일 수도 있다는 것에 주의한다. Y'' , C_b'' , 및 Cr'' 은 DRA 파라미터들이 각각의 컬러 성분에 적용된 후의 비디오 신호의 컬러 성분들이다. 위의 예에서 알 수 있는 바와 같이, 각각의 컬러 성분은 상이한 스케일 및 오프셋 파라미터들에 관련된다. 예를 들어, $scale1$ 및 $offset1$ 은 Y' 성분을 위해 사용되며, $scale2$ 및 $offset2$ 는 C_b' 성분을 위해 사용되고, $scale3$ 및 $offset3$ 은 Cr' 성분을 위해 사용된다. 이는 단지 일 예라는 것이 이해되어야 한다. 다른 예들에서, 동일한 스케일 및 오프셋 값들은 모든 컬러 성분에 대해 사용될 수도 있다.

[0123] 다른 예들에서, 각각의 컬러 성분은 다수의 스케일 및 오프셋 파라미터들과 연관될 수도 있다. 예를 들어,

Cr 또는 Cb 컬러 성분들에 대한 크로마 값들의 실제 분포는 상이한 파티션들 또는 코드워드들의 범위들에 대해 상이할 수도 있다. 일 예에서, 중앙 코드워드 아래에 존재하기 보다는 중앙 코드워드 (예를 들어, 코드워드 (512)) 위에 이용되는 보다 고유한 코드워드들이 존재할 수도 있다. 이러한 예에서, 조정 부 (210)는 (예를 들어, 중앙 코드워드보다 더 큰 값을 갖는) 중앙 코드워드 위의 크로마 값들에 대해 스케일 및 오프셋 파라미터들의 하나의 세트를 적용하고 (예를 들어, 중앙 코드워드보다 낮은 값을 갖는) 중앙 코드워드 아래의 크로마 값들에 대해 스케일 및 오프셋 파라미터들의 상이한 세트를 적용하도록 구성될 수도 있다.

[0124] 위의 예에서 알 수 있는 바와 같이, 조정 부 (210)는 스케일 및 오프셋 DRA 파라미터들을 선형 함수로서 적용할 수도 있다. 이와 같이, 컬러 변환 부 (208)에 의한 컬러 변환 후 조정 부 (210)가 타겟 컬러 공간에서 DRA 파라미터들을 적용하는 것이 필요하지 않다. 이는 컬러 변환이 자체가 선형 프로세스이기 때문이다.

이와 같이, 다른 예들에서, 조정 부 (210)는 임의의 컬러 변환 프로세스 전에 네이티브 컬러 공간 (예컨대, RGB)에서의 비디오 데이터에 DRA 파라미터들을 적용할 수도 있다. 이 예에서, 조정 부 (210)가 DRA 파라미터들을 적용한 후 컬러 변환 부 (208)는 컬러 변환을 적용할 것이다.

[0125] 본 개시물의 다른 예에서, 조정 부 (210)는 DRA 파라미터들을 타겟 컬러 공간 또는 네이티브 컬러 공간 중 어느 하나에서 다음과 같이 적용할 수도 있다:

[0126] - $Y'' = (\text{scale1} * (Y' - \text{offsetY}) + \text{offset1}) + \text{offsetY};$

[0127] - $Cb'' = \text{scale2} * Cb' + \text{offset2}$ (9)

[0128] - $Cr'' = \text{scale3} * Cr' + \text{offset3}$

[0129] 이 예에서, 파라미터 scale1 , scale2 , scale3 , offset1 , offset2 , 및 offset3 은 위에서 설명된 바와 동일한 의미를 가진다. 파라미터 offsetY 는 신호의 밝기를 반영하는 파라미터이고, Y' 의 평균 값과 동일할 수 있다.

다른 예들에서, 입력 및 출력 표현들에서 중심 값의 매핑을 더 잘 보존하기 위해 offsetY 와 유사한 오프셋 파라미터가 Cb' 및 Cr' 성분들에 적용될 수도 있다.

[0130] 본 개시물의 다른 예에서, 조정 부 (210)는 네이티브 컬러 공간 또는 타겟 컬러 공간이 아닌 컬러 공간에서 DRA 파라미터들을 적용하도록 구성될 수도 있다. 대체로, 조정 부 (210)는 DRA 파라미터들을 다음과 같이 적용하도록 구성될 수도 있으며:

[0131] - $A' = \text{scale1} * A + \text{offset1};$

[0132] - $B' = \text{scale2} * B + \text{offset2}$ (10)

[0133] - $C' = \text{scale3} * C + \text{offset3}$

[0134] 여기서 신호 성분들 (A , B , C)은 타겟 컬러 공간, 예컨대, RGB 또는 중간 컬러 공간과는 상이한 컬러 공간에서의 신호 성분들이다.

[0135] 본 개시물의 다른 예들에서, 조정 부 (210)는 DRA를 수행하기 위해 비디오에 선형 전달 함수를 적용하도록 구성될 수도 있다. 그런 전달 함수는 동적 범위를 압축하기 위해 전달 함수 부 (206)에 의해 사용되는 전달 함수와는 상이하다. 위에서 정의된 스케일 및 오프셋 항들과 유사하게, 조정 부 (210)에 의해 적용되는 전달 함수는 타겟 컬러 컨테이너의 이용가능 코드워드들로 컬러 값을 확장하고 중심 맞추는데 사용될 수도 있다.

DRA를 수행하기 위해 전달 함수를 적용하는 일 예가 아래에서 도시된다:

[0136] - $Y'' = \text{TF2}(Y')$

[0137] - $Cb'' = \text{TF2}(Cb')$

[0138] - $Cr'' = \text{TF2}(Cr')$

[0139] 항 TF2는 조정 부 (210)에 의해 적용되는 전달 함수를 특정한다. 일부 예들에서 조정 부 (210)는 상이한 전달 함수들을 성분들의 각각에 적용하도록 구성될 수도 있다.

[0140] 본 개시물 다른 예에서, 조정 부 (210)는 단일 프로세스에서 컬러 변환 부 (208)의 컬러 변환과 공동으로 DRA 파라미터들을 적용하도록 구성될 수도 있다. 다시 말하면, 조정 부 (210) 및 컬러 변환 부 (208)의 선형 함수들은 결합될 수도 있다. $f1$ 및 $f2$ 가 RGB 대 YCbCr 매트릭스와 DRA 스케일링 계수들의 조합인 조합된 애플리케이션의 일 예가 아래에서 도시된다:

$$C_b = \frac{B' - Y'}{f1}, C_r = \frac{R' - Y'}{f2}$$

[0142] 본 개시물의 다른 예에서, DRA 파라미터들을 적용한 후, 조정 부 (210)는 비디오 데이터가 특정한 타겟 컬러 컨테이너를 위해 특정된 코드워드 범위 밖의 값들을 갖는 것을 방지하는 클리핑 프로세스를 수행하도록 구성될 수도 있다. 일부 환경들에서, 조정 부 (210)에 의해 적용되는 스케일 및 오프셋 파라미터들은 일부 컬러 성분 값들이 허용가능 코드워드들의 범위를 초과하게 할 수도 있다. 이 경우, 조정 부 (210)는 범위를 초과하는 성분들의 값들을 그 범위에서의 최대 값으로 클리핑하도록 구성될 수도 있다.

[0143] 조정 부 (210)에 의해 적용되는 DRA 파라미터들은 DRA 파라미터들 추정 부 (212)에 의해 결정될 수도 있다. DRA 파라미터들 추정 부 (212)가 DRA 파라미터들을 업데이트하는 빈도 및 시간 인스턴스들은 유통성이 있다. 예를 들어, DRA 파라미터들 추정 부 (212)는 시간적 레벨에서 DRA 파라미터들을 업데이트할 수도 있다. 다시 말하면, 새로운 DRA 파라미터들은 픽처들의 그룹 (GOP), 또는 단일 픽처 (프레임)에 대해 결정될 수도 있다. 이 예에서, RGB 네이티브 CG 비디오 데이터 (200)는 GOP 또는 단일 픽처일 수도 있다. 다른 예들에서, DRA 파라미터들 추정 부 (212)는 공간적 레벨에서, 예컨대, 슬라이스 타일, 또는 블록 레벨에서 DRA 파라미터들을 업데이트할 수도 있다. 이 맥락에서, 비디오 데이터의 블록이 매크로블록, 코딩 트리 유닛 (CTU), 코딩 유닛, 또는 임의의 다른 사이즈 및 모양의 블록일 수도 있다. 블록이 정사각형, 직사각형, 또는 임의의 다른 형상일 수도 있다. 따라서, DRA 파라미터들은 더욱 효율적인 시간적 및 공간 예측과 코딩을 위해 사용될 수도 있다.

[0144] 본 개시물의 일 예에서, DRA 파라미터들 추정 부 (212)는 RGB 네이티브 CG 비디오 데이터 (200)의 네이티브 컬러 영역 및 타겟 컬러 컨테이너의 컬러 영역의 대응성에 기초하여 DRA 파라미터들을 도출할 수도 있다. 예를 들어, DRA 파라미터들 추정 부 (212)는 특정 네이티브 컬러 영역 (예를 들어, BT.709) 및 타겟 컬러 컨테이너의 컬러 영역 (예를 들어, BT.2020)가 주어지면 스케일 및 오프셋 값들을 결정하도록 미리 정의된 룰들의 세트를 이용할 수도 있다.

[0145] 예를 들어, 네이티브 컬러 영역 및 타겟 컬러 컨테이너가 xy 공간 및 백색점 좌표에서 컬러 원색 좌표의 형태로 정의된다고 가정한다. BT.709 및 BT.2020에 대한 이러한 정보의 일 예가 아래의 표 2에 도시된다:

표 2- RGB 컬러 공간 파라미터들

컬러 공간	RGB 컬러 공간 파라미터들							
	백색점		원색들					
	XXW	YYW	XXR	YYR	XXG	YYG	XXB	YYB
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

[0146]

[0147] 일 예에서, BT.2020는 타겟 컬러 컨테이너의 컬러 영역이고 BT.709는 네이티브 컬러 컨테이너의 컬러 영역이다. 이 예에서, 조정 부 (210)는 DRA 파라미터들을 YCbCr 타겟 컬러 공간에 적용한다. DRA 파라미터들 추정 부 (212)는 다음과 같이 DRA 파라미터들을 추정하여 이 파라미터들을 조정 부 (210)에 포워딩하도록 구성된다:

[0148] scale1 = 1; offset1 = 0;

[0149] scale2 = 1.0698; offset2 = 0;

[0150] scale3 = 2.1735; offset3 = 0;

[0151] 다른 예에서, BT.2020가 타겟 컬러 공간이고, P3는 네이티브 컬러 공간이고 DRA는 YCbCr 타겟 컬러 공간에 적용되는 것에 의해 DRA 파라미터들 추정 부 (212)는 DRA 파라미터들을 다음과 같이 추정하도록 구성될 수도

있다:

```
[0152] scale1 = 1;      offset1 = 0;
[0153] scale2 = 1.0068;    offset2 = 0;
[0154] scale3 = 1.7913;    offset3 = 0;
```

[0155] 위의 예들에서, DRA 파라미터들 추정 부 (212) 는 특정 네이티브 컬러 영역 및 특정 타겟 컬러 영역이 주어지면, 이용할 DRA 파라미터들을 나타내는 루프 테이블을 컨설팅하는 것에 의해 위에 나열된 스케일 및 오프셋 값들을 결정한다. 다른 예에서, DRA 파라미터들 추정 부 (212) 는 예를 들어, 표 2 에 도시된 바와 같이 네이티브 컬러 영역 및 타겟 컬러 영역의 원색 및 백컬러 공간 값들로부터 DRA 파라미터들을 계산하도록 구성될 수도 있다.

[0156] 예를 들어, 타겟 (T) 컬러 컨테이너는 원색 좌표 (x_{Tt} , y_{Tt}) 로 규정되고 여기에서 X 는 R,G,B 컬러 성분들에 대해 나타내어지는 것으로 상정하며:

$$\text{primeT} = \begin{bmatrix} x_{Rt} & y_{Rt} \\ x_{Gt} & y_{Gt} \\ x_{Bt} & y_{Bt} \end{bmatrix}$$

[0157] 그리고, 네이티브 (N) 컬러 영역은 원색 좌표들 (x_{Nn} , y_{Nn}) 로 규정되고, 여기에서 X 는 R,G,B 컬러 성분들에 대해 나타내어지는 것으로 상정한다:

$$\text{primeN} = \begin{bmatrix} x_{Rn} & y_{Rn} \\ x_{Gn} & y_{Gn} \\ x_{Bn} & y_{Bn} \end{bmatrix}$$

[0159] [0160] 양쪽 영역들에 대한 백색점 좌표는 $\text{whiteP} = (x_W, y_W)$ 와 같다. DRA 파라미터들 추정 부 (212) 는 백색점으로의 원색 좌표들 사이의 거리들의 함수로서 DRA 에 대한 scale2 및 scale3 파라미터들을 도출할 수도 있다. 이러한 추정의 일 예가 아래에 주어진다:

$$\begin{aligned} \text{rdT} &= \sqrt{(\text{primeT}(1,1) - \text{whiteP}(1,1))^2 + (\text{primeT}(1,2) - \text{whiteP}(1,2))^2} \\ \text{gdT} &= \sqrt{(\text{primeT}(2,1) - \text{whiteP}(1,1))^2 + (\text{primeT}(2,2) - \text{whiteP}(1,2))^2} \\ \text{bdT} &= \sqrt{(\text{primeT}(3,1) - \text{whiteP}(1,1))^2 + (\text{primeT}(3,2) - \text{whiteP}(1,2))^2} \end{aligned}$$

$$\text{rdN} = \sqrt{(\text{primeN}(1,1) - \text{whiteP}(1,1))^2 + (\text{primeN}(1,2) - \text{whiteP}(1,2))^2}$$

$$\text{gdN} = \sqrt{(\text{primeN}(2,1) - \text{whiteP}(1,1))^2 + (\text{primeN}(2,2) - \text{whiteP}(1,2))^2}$$

$$\text{bdN} = \sqrt{(\text{primeN}(3,1) - \text{whiteP}(1,1))^2 + (\text{primeN}(3,2) - \text{whiteP}(1,2))^2}$$

$$\text{scale2} = \text{bdT}/\text{bdN}$$

$$\text{scale3} = \sqrt{(\text{rdT}/\text{rdN})^2 + (\text{gdT}/\text{gdN})^2}$$

[0161] [0162] 일부 예들에서, DRA 파라미터들 추정 부 (212) 는 네이티브 컬러 영역의 미리 정의된 원색 값들로부터가 아닌 RGB 네이티브 CG 비디오 데이터 (200) 에서의 컬러 값들의 실제 분포값으로부터 primeN 에서 원색 좌표들을 결정하는 것에 의해 DRA 파라미터들을 추정하도록 구성될 수도 있다. 즉, DRA 파라미터들 추정 부 (212) 는 RGB 네이티브 CG 비디오 데이터 (200) 에 존재하는 실제 컬러들을 분석하고 그리고 위에 설명된 기능으로 이러한 분석으로부터 결정된 원색 값들, 및 백색점을 이용하여 DRA 파라미터들을 계산하도록 구성될 수도 있다. 위에 정의된 일부 파라미터들의 근사치가 계산을 용이하게 하도록 DRA 로서 이용될 수도 있다. 예를 들어, $\text{scale3} = 2.1735$ 는 $\text{scale3} = 2$ 로 근사치될 수도 있고, 이는 일부 아키텍처에서 보다 쉬운 구현을 허용한다.

[0163] 본 개시물의 다른 예에서, DRA 파라미터들 추정 부 (212) 는 타겟 컬러 공간에서뿐만 아니라 타겟 컬러 컨테이너의 컬러 영역에 기초하여 DRA 파라미터들을 결정하도록 구성될 수도 있다. 성분 값들의 값들의 실제 분포들은 컬러 공간마다 상이할 수도 있다. 예를 들어, 크로마 값들 분포들은 일정하지 않는 휘도를 갖는 YCbCr

컬러 공간들에 비해, 일정한 휘도를 갖는 YCbCr 컬러 공간에 대해 상이할 수도 있다. DRA 파라미터들 추정 부 (212) 는 DRA 파라미터들을 결정하도록 상이한 컬러 공간들의 컬러 분포들을 이용할 수도 있다.

[0164] 본 개시물의 다른 예들에서, DRA 파라미터들 추정 부 (212) 는 비디오 데이터를 프리프로세싱하는 것 및/또는 인코딩하는 것에 연관된 특정한 비용 함수들을 최소화하기 위해서 DRA 파라미터들에 대한 값들을 도출하도록 구성될 수도 있다. 하나의 예로서, DRA 파라미터들 추정 부 (212) 는 위의 양자화 부 (214) (예컨대, 식 (4) 참조)에 의해 도입된 양자화 에러들을 최소화했던 DRA 파라미터들을 추정하도록 구성될 수도 있다. DRA 파라미터들 추정 부 (212) 는 적용되는 상이한 DRA 파라미터 세트들을 가졌던 비디오 데이터에 대해 양자화 에러 테스트들을 수행함으로써 이러한 에러를 최소화할 수도 있다. 다른 예에서, DRA 파라미터들 추정 부 (212) 는 양자화 부 (214)에 의해 도입된 양자화 에러들을 인지적 방식으로 최소화하는 DRA 파라미터들을 추정하도록 구성될 수도 있다. DRA 파라미터들 추정 부 (212) 는 적용되는 상이한 DRA 파라미터 세트들을 가졌던 비디오 데이터에 대한 인지 에러 테스트들에 기초하여 이러한 에러를 최소화할 수도 있다. DRA 파라미터들 추정 부 (212) 는 그 다음에 최저 양자화 에러를 생성했던 DRA 파라미터들을 선택할 수도 있다.

[0165] 다른 예에서, DRA 파라미터들 추정 부 (212) 는 조정 부 (210)에 의해 수행된 DRA 및 비디오 인코더 (20)에 의해 수행된 비디오 인코딩 둘 다에 연관된 비용 함수를 최소화하는 DRA 파라미터들을 선택할 수도 있다. 예를 들어, DRA 파라미터들 추정 부 (212) 는 DRA를 수행하고 다수의 상이한 DRA 파라미터 세트들과 함께 비디오 데이터를 인코딩할 수도 있다. DRA 파라미터들 추정 부 (212) 는 그 다음에 DRA 및 비디오 인코딩, 뿐만 아니라 이들 두 개의 손실 프로세스에 의해 도입된 왜곡으로부터 초래되는 비트레이트의 가중된 합을 형성함으로써 각각의 DRA 파라미터 세트에 대한 비용 함수를 계산할 수도 있다. DRA 파라미터들 추정 부 (212) 는 그 다음에 비용 함수를 최소화하는 DRA 파라미터들의 세트를 선택할 수도 있다.

[0166] DRA 파라미터들 추정을 위한 위의 기법들의 각각에서, DRA 파라미터들 추정 부 (212) 는 각각의 성분에 대해 그 성분에 관한 정보를 사용하여 따로따로 DRA 파라미터들을 결정할 수도 있다. 다른 예들에서, DRA 파라미터들 추정 부 (212) 는 교차-성분 정보를 사용하여 DRA 파라미터들을 결정할 수도 있다. 예를 들어, Cr 성분에 대해 도출된 DRA 파라미터들은 Cb 성분에 대한 DRA 파라미터들을 도출하는데 사용될 수도 있다.

[0167] DRA 파라미터들을 도출하는 것에 더하여, DRA 파라미터 추정 부 (212) 는 인코딩된 비트스트림에서 DRA 파라미터들을 시그널링하도록 구성될 수도 있다. DRA 파라미터들 추정 부 (212) 는 시그널링을 위하여 비디오 인코더 (20)에 하나 이상의 선택스 엘리먼트들을 제공하도록 구성될 수도 있거나 직접 DRA 파라미터들을 표시하는 하나 이상의 선택스 엘리먼트들을 시그널링할 수도 있다. 이러한 파라미터들의 선택스 엘리먼트들은 비디오 디코더 (30) 및/또는 비디오 포스트프로세서 부 (31)가 비디오 프리프로세서 부 (19)의 프로세스의 역을 수행하여 자신의 네이티브 컬러 컨테이너에서 비디오 데이터를 복원할 수도 있도록 하기 위해 비트스트림에서 시그널링될 수도 있다. DRA 파라미터들을 시그널링하는 예시적 기법들은 아래 설명된다.

[0168] 일 예에서, DRA 파라미터들 추정 부 (212) 는 메타데이터로서 인코딩된 비디오 비트스트림에서, 보충 강화 정보 (SEI) 메시지에서, 비디오 이용능력 정보 (VUI)에서, 비디오 파라미터 세트 (VPS)에서, 시퀀스 파라미터 세트 (SPS)에서, 픽쳐 파라미터 세트에서, 슬라이스 헤더에서, CTU 헤더에서, 또는 비디오 데이터 (예를 들어, GOP, 픽쳐들, 블록들, 매크로블록, CTU들 등))의 사이즈에 대해 DRA 파라미터들을 표시하는데 적합한 임의의 다른 선택스 구조에서 하나 이상의 선택스 엘리먼트들을 시그널링할 수도 있다.

[0169] 일부 예들에서, 하나 이상의 선택스 엘리먼트들은 DRA 파라미터들을 명시적으로 표시한다. 예를 들어, 하나 이상의 선택스 엘리먼트들은 DRA에 대한 여러 스케일 및 오프셋 값들일 수도 있다. 다른 예들에서, 하나 이상의 선택스 엘리먼트들은 DRA에 대한 스케일 및 오프셋 값을 포함하는 루업 테이블로의 하나 이상의 인덱스들일 수도 있다. 또 다른 예에서, 하나 이상의 선택스 엘리먼트들은 DRA에 대하여 이용할 선형 전달 함수를 규정하는 루업 테이블로의 인덱스들일 수도 있다.

[0170] 다른 예들에서, DRA 파라미터들은 명시적으로 시그널링되지 않고, 다만, 비디오 프리프로세서 부 (19) 및 비디오 포스트프로세서 부 (31) 양쪽이, 비트스트림의 식별 가능한 형태인 비디오 데이터의 동일한 정보 및/또는 특징들을 이용하여 동일한 미리 정의된 프로세스를 이용하여 DRA 파라미터들을 도출하도록 구성된다. 일 예로서, 비디오 포스트프로세서 부 (31)는 인코딩된 비트스트림에서 인코딩된 비디오 데이터의 타겟 컬러 컨테이너 뿐만 아니라 비디오 데이터의 네이티브 컬러 컨테이너를 표시하도록 구성될 수도 있다. 그 후, 비디오 포스트프로세서 부 (31)는 위에 정의된 바와 같은 동일한 프로세스를 이용하여 이러한 정보로부터 DRA 파라미터들을 도출하도록 구성될 수도 있다. 일부 예들에서, 네이티브 및 타겟 컬러 컨테이너들을 식별하는 하나 이상의 선택스 엘리먼트들은 선택스 엘리먼트들에 제공된다. 이러한 선택스 엘리먼트들은 컬러 컨테이너들을 명

시적으로 표시할 수도 있거나 또는 룩업 테이블로의 인덱스일 수도 있다. 다른 예에서, 비디오 프리프로세서 부 (19)는 특정 컬러 컨테이너에 대한 컬러 원색들 및 백색 점의 XY 값들을 표시하는 하나 이상의 선택스 엘리먼트들을 시그널링하도록 구성될 수도 있다. 다른 예에서, 비디오 프리프로세서 부 (19)는 DRA 파라미터들 추정 부 (212)에 의해 수행된 분석에 기초하여 비디오 데이터에서 실제 컬러 값들의 컬러 원색들 및 백색 점 (컨텐츠 원색 및 컨텐츠 백색점)의 XY 값들을 표시하는 하나 이상의 선택스 엘리먼트들을 시그널링하도록 구성될 수도 있다.

[0171] 일 예로서, 컨텐츠에서 컬러를 포함하는 최소 컬러 영역의 컬러 원색들이 시그널링될 수도 있고, 비디오 디코더 (30) 및/또는 비디오 포스트 프로세서 부 (31)에서, DRA 파라미터들이 컨텐츠 원색들 및 컨테이너 원색들 양쪽을 이용하여 도출된다. 일 예에서, 컨텐츠 원색들은 위에 설명된 바와 같이, R, G 및 B에 대한 x 및 y 성분들을 이용하여 시그널링될 수 있다. 다른 예에서, 컨텐츠 원색들은 알려진 2개의 원색 세트들 간의 비율로서 시그널링될 수 있다. 예를 들어, 컨텐츠 원색들은 BT.709 원색들과 BT.2020 원색들 사이의 선형 포지션: $X_{r_content} = alfa_r * X_{r_bt709} + (1-alfa_r) * X_{r_bt2020}$ (G 및 B 성분에 대해 $alfa_a$ 및 $alfa_b$ 와 유사한 식을 가짐)으로서 시그널링될 수 있고, 여기에서, 파라미터 $alfa_r$ 은 2개의 알려진 원색 세트들 사이의 비율을 규정한다. 일부 예들에서, 시그널링되고/되거나 도출된 DRA 파라미터들은 비디오 인코더 (20) 및/또는 비디오 디코더 (30)에 의해 이용되어, HDR/WCG 비디오 데이터를 코딩하기 위하여 이용된 기법들에 기초한 가중된 예측을 용이하게 한다.

[0172] 가중 예측을 이용하는 비디오 코딩 스킁들에서, 가중값 (W_{wp})이 취해진 참조 픽처의 (단일 방향 예측을 위한) 샘플 (S_r)과 예측된 샘플 (S_p)에서 초래되는 오프셋 (O_{wp})으로부터 현재 코딩되는 픽처의 샘플 (S_c)이 예측된다:

$$S_p = S_r * W_{wp} + O_{wp}.$$

[0174] DRA를 이용하는 일부 예들에서, 참조 및 현재 코딩되는 픽처의 샘플들은 상이한 파라미터들, 즉 현재 픽처에 대한 $\{scale1_{cur}, offset1_{cur}\}$ 및 참조 픽처에 대한 $\{scale1_{ref}, offset1_{ref}\}$ 을 사용하는 DRA로 프로세싱될 수 있다. 그런 실시형태들에서, 가중 예측의 파라미터들은 DRA로부터 도출될 수 있는데, 예를 들어:

$$W_{wp} = scale1_{cur} / scale1_{ref}$$

$$O_{wp} = offset1_{cur} - offset1_{ref}$$

[0177] 조정 부 (210)가 DRA 파라미터들에 적용한 후, 비디오 프리프로세서 부 (19)는 그러면 양자화 부 (214)를 사용하여 비디오 데이터를 양자화할 수도 있다. 양자화 부 (214)는 도 4를 참조하여 위에서 설명된 것과 동일한 방식으로 동작할 수도 있다. 양자화 후, 비디오 데이터는 이제 HDR' 데이터 (216)의 타겟 원색들의 색 영역들과 타겟 컬러 공간에서 조정된다. HDR' 데이터 (216)는 그 다음에 비디오 인코더 (20)에 압축을 위해 전송될 수도 있다.

[0178] 도 9는 본 개시물의 기법들에 따라 예시적인 HDR/WCG 역 변환 장치를 도시하는 블록도이다. 도 9에 도시된 바와 같이, 비디오 포스트프로세서 부 (31)는 도 8의 비디오 프리프로세서 부 (19)에 의해 수행되는 기법들의 역을 적용하도록 구성될 수도 있다. 다른 예들에서, 비디오 포스트프로세서 부 (31)의 기법들은 비디오 디코더 (30)에 통합되고 그 비디오 디코더에 의해 수행될 수도 있다.

[0179] 하나의 예에서, 비디오 디코더 (30)는 비디오 인코더 (20)에 의해 인코딩된 비디오 데이터를 디코딩하도록 구성될 수도 있다. 디코딩된 비디오 데이터 (타겟 컬러 컨테이너에서의 HDR' 데이터 (316))는 그러면 비디오 포스트프로세서 부 (31)에 포워딩된다. 역 양자화 부 (314)는 도 8의 양자화 부 (214)에 의해 수행되는 양자화 프로세스를 역전시키기 위해 HDR' 데이터 (316)에 대해 역 양자화 프로세스를 수행한다.

[0180] 비디오 디코더 (30)는 도 8의 DRA 파라미터들 추정 부 (212)에 의해 생성된 하나 이상의 선택스 엘리먼트들 중 임의의 선택스 엘리먼트를 디코딩하고 비디오 포스트프로세서 부 (31)의 DRA 파라미터들 도출 부 (312)에 전송하도록 또한 구성될 수도 있다. DRA 파라미터들 도출 부 (312)는 위에 설명된 하나 이상의 선택스 엘리먼트들에 기초하여 DRA 파라미터들을 결정하도록 구성될 수도 있다. 일부 예들에서, 하나 이상의 선택스 엘리먼트들은 DRA 파라미터들을 명시적으로 나타낼 수도 있다. 다른 예들에서, DRA 파라미터들 도출 부 (312)는 도 8의 DRA 파라미터들 추정 부 (212)에 의해 사용되는 동일한 기법들을 사용하여 DRA 파라미터들을

도출하도록 구성된다.

[0181] DRA 파라미터들 도출 부 (312)에 의해 도출되는 파라미터들은 역 조정 부 (310)로 전송될 수도 있다. 역 조정 부 (310)는 조정 부 (210)에 의해 수행되는 선형 DRA 조정의 역을 수행하기 위해 DRA 파라미터들을 사용한다. 역 조정 부 (310)는 조정 부 (210)에 대해 위에서 설명된 조정 기법들 중 임의의 조정 기법의 역을 적용할 수도 있다. 덧붙여서, 조정 부 (210)에서처럼, 역 조정 부 (310)는 임의의 역 컬러 변환 전 또는 후에 역 DRA를 적용할 수도 있다. 이와 같이, 역 조정 부 (310)는 타겟 컬러 컨테이너 또는 네이티브 컬러 컨테이너의 비디오 데이터에 DRA 파라미터를 적용할 수도 있다. 일부 예들에서, 역 조정 부는 역 양자화 부 (314) 전에 역 조정을 적용하도록 포지셔닝될 수도 있다.

[0182] 역 컬러 변환 부 (308)는 비디오 데이터를 타겟 컬러 공간 (예컨대, YCbCr)에서부터 네이티브 컬러 공간 (예컨대, RGB)으로 변환한다. 역 전달 함수 (306)는 그 다음에 비디오 데이터의 동적 범위를 비압축하기 위해 전달 함수 (206)에 의해 적용되는 전달 함수의 역을 적용한다. 일부 예들에서, 결과적인 비디오 데이터 (RGB 타겟 CG (304))는 타겟 색 영역에 여전히 있지만, 이제는 네이티브 동적 범위 및 네이티브 컬러 공간에 있다. 다음으로, 역 CG 변환기 (302)는 RGB 네이티브 CG (300)를 복원하기 위해 RGB 타겟 CG(304)를 네이티브 색 영역으로 변환한다.

[0183] 일부 예들에서, 추가적인 후처리 기법들은 비디오 포스트프로세서 부 (31)에 의해 채용될 수도 있다. DRA를 적용하면 비디오가 자신의 실제 네이티브 색 영역 밖에 놓일 수도 있다. 양자화 부 (214) 및 역 양자화 부 (314)에 의해 수행되는 양자화 단계들, 뿐만 아니라 조정 부 (210) 및 역 조정 부 (310)에 의해 수행되는 업 및 다운-샘플링 기법들은, 네이티브 색 영역 밖에 있는 네이티브 컬러 컨테이너에서의 결과적인 컬러 값들에 기여할 수도 있다. 네이티브 색 영역 (또는 위에서 설명된 바와 같이, 시그널링된다면, 실제 최소 콘텐츠 원색들)이 알려질 때, 추가적인 프로세스는 DRA에 대한 포스트프로세싱으로서 컬러 값들 (예컨대, RGB 또는 Cb 및 Cr)을 의도된 영역 (gamut)으로 다시 변환하기 위해 RGB 네이티브 CG 비디오 데이터 (304)에 적용될 수 있다. 다른 예들에서, 이러한 포스트프로세싱은 양자화 후 또는 DRA 적용 후 적용될 수도 있다.

[0184] 위에 언급된 바와 같이, 수개의 SEI 메시지들은 비디오 데이터의 여러 컬러 성분들에 대한 동적 범위 조정 정보에 관한 정보를 전달하는데 이용될 수도 있다. 성분 스케일링 SEI 메시지, 이를 테면, 위에 설명되고 아래 보다 자세히 설명된 메시지는 비디오 데이터의 여러 컬러 성분들에 대한 매핑 정보를 표시하는데 이용될 수 있는 스케일 팩터들, 오프셋들, 및 범위들 (예를 들어, 코드워드 값들의 파티션들)의 세트를 전달할 수도 있다. 매핑 정보는 샘플 값들의 상이한 범위를 확장 또는 수축하는 방법을 비디오 디코더 (30) 및/또는 비디오 포스트프로세서 부 (31)에 표시하는데 이용될 수도 있어, 이러한 방식으로, 백워드 호환 가능성이 요구되는 일부 경우들에서 복원된 HDR 비디오 데이터의 전체적인 품질 또는 또한 복원된 SDR 비디오 데이터의 품질이 개선되거나, 또는 복원된 출력이 디스플레이 용량들에 대하여 보다 적합하게 되도록 한다.

[0185] 아래의 표 3은 성분 스케일링 SEI 메시지의 선택스 구조의 일 변형 예를 제공한다. 아래의 선택스 엘리먼트들의 명칭들은 아래 예들에서 설명된 것과 상이한 프리픽스 "hdr_recon_"를 포함하고, 선택스 엘리먼트들의 명칭들이 component_scaling로서 프리픽스되어 있지만, 그렇지 않으면 선택스 엘리먼트는 동일함을 주지해야 한다.

[0186]

표 3 - 예시적 범위 조정 SEI 신택스

hdr_reconstruction_info(payloadSize) {	디스크립터
hdr_recon_id	ue(v)
hdr_recon_cancel_flag	u(1)
if(!hdr_recon_cancel_flag) {	
hdr_recon_persistence_flag	u(1)
hdr_recon_transfer_characteristics	u(8)
hdr_recon_default_flag	u(1)
if(!hdr_recon_default_flag) {	
hdr_recon_scale_bit_depth	u(4)
hdr_recon_offset_bit_depth	u(4)
hdr_recon_scale_frac_bit_depth	u(4)
hdr_recon_offset_frac_bit_depth	u(4)
hdr_recon_num_comps_minus1	ue(v)
}	
for(c = 0; c <= hdr_recon_num_comps_minus1; c++) {	
hdr_recon_num_ranges[c]	ue(v)
hdr_recon_equal_ranges_flag[c]	u(1)
hdr_recon_global_offset_val[c]	u(v)
for(i = 0; i <= hdr_recon_num_ranges[c]; i++)	
hdr_recon_scale_val[c][i]	u(v)
if(!hdr_recon_equal_ranges[c])	
for(i = 0; i <= hdr_recon_num_ranges[c]; i++)	
hdr_recon_range_val[c][i]	u(v)
}	
}	
}	

[0187]

[0188]

표 3 의 SEI 신택스의 시맨틱들은 아래 제시된다.

[0189]

매핑 프로세스는 다음과 같이 [0,1]에서의 값 x 를 값 $y = \text{map}[c](x)$ 에 매핑하는 $c = 0..hdr_{\text{recon}}_{\text{num}}_{\text{comps}}_{\text{minus1}}$ 에 대해 피스별 선형 함수들 $\text{map}[c]()$ 에 기초한다:

[0190]

- 0 내지 $\text{hdr}_{\text{recon}}_{\text{num}}_{\text{ranges}}[c] - 1$ 의 범위에서의 i 에 대해 다음이 적용된다:

[0191]

- 값 $\text{ScaleValue}[c][i]$ 은 신택스 엘리먼트 $\text{hdr}_{\text{recon}}_{\text{scale}}_{\text{val}}[c][i]$ 의 시맨틱스들에서 설명된 바와 같이 도출된다.

[0192]

- 값 $\text{RangeValue}[c][i]$ 은 신택스 엘리먼트 $\text{hdr}_{\text{recon}}_{\text{range}}_{\text{val}}[c][i]$ 의 시맨틱스들에서 설명된 바와 같이 도출된다.

[0193]

- 0 내지 $\text{hdr}_{\text{recon}}_{\text{num}}_{\text{ranges}}[c] - 1$ 의 범위에서의 i 에 대해 값 $\text{InputRanges}[c][i]$ 및 $\text{OutputRanges}[c][i]$ 는 다음과 같이 도출된다:

[0194]

- i 가 0 과 같으면, 다음이 적용된다:

$\text{OutputRanges}[c][i] = -\text{hdr_recon_global_offset_val}[c] * \text{ScaleValue}[c][i-1]$ (D-xx)

$\text{InputRanges}[c][i] = 0$ (D-xx)

[0195]

- 그렇지 않으면(i 가 0 과 같지 않으면), 다음이 적용된다:

$\text{InputRanges}[c][i] = \text{InputRanges}[c][i-1] + \text{RangeValue}[c][i-1]$ (D-xx)

$\text{OutputRanges}[c][i] = \text{OutputRanges}[c][i-1] +$

[0197]

$\text{RangeValue}[c][i-1] * \text{ScaleValue}[c][i-1]$ (D-xx)

[0198]

- 0 내지 $\text{hdr_recon_num_ranges}[c] - 1$ 의 범위에서의 i 에 대해 값들 $\text{OffsetValue}[c][i]$ 은 다음과 같이 도출된다:

$\text{OffsetValue}[c][i] = \text{InputRanges}[c][i+1] - \text{OutputRanges}[c][i+1]$

$\text{ScaleValue}[c][i-1]$ (D-xx)

[0199]

- 파라미터 $y = \text{map}[c](x)$ 는 다음과 같이 도출된다:

- x 가 $\text{OutputRanges}[c][0]$ 이하이면, 다음이 적용된다:

$y = \text{InputRanges}[c][0]$ (D-xx)

[0202]

- 그렇지 않고 x 가 $\text{OutputRanges}[c][\text{hdr_recon_num_ranges}[c]]$ 보다 크면, 다음이 적용된다:

$y = \text{InputRanges}[c][\text{hdr_recon_num_ranges}[c]]$ (D-xx)

[0203]

- 그렇지 않으면, 다음이 적용된다:

`for(i = 1; i <= hdr_recon_num_ranges[c]; i++)`

`if(OutputRanges[i-1] < x && x <= OutputRanges[i])`

$y = x / \text{ScaleValue}[c][i-1] + \text{OffsetValue}[c][i-1]$ (D-xx)

[0204]

성분 스케일링 정보 SEI 메시지들 및 성분들의 동적범위를 조정하는데 적용가능한 것에 여러 문제들이 식별되었다. 특히, 식별된 문제들은 샘플 값들 (RGB 값들, YCrCb 값들, YUV 값들, XYZ 값들 등)에 대해 코드워드들의 범위들 뿐만 아니라 스케일 및 오프셋 값들을 도출하기 위해 부동 소수점 숫자들의 이용에 관련된다. 예를 들어, 비트스트림에서 시그널링되는 스케일 값들은 역 동적 범위 조정 프로세스를 수행하도록 디코더 측에서, 예를 들어, 비디오 포스트프로세서 부 (31)에 의해 이용된다. 그러나, 샘플 값들의 범위들을 컴퓨팅하기 위하여 그리고 매핑 프로세스를 컴퓨팅하기 위하여 스케일 값들을 이용하기 위해, 상호 가역적 동작이 비디오 포스트 프로세서 (31)에서 수행된다. 성분 스케일링 SEI 메시지에 대한 이전 예시의 시맨틱들은 샘플 값들과 곱셈되도록 상호가역성 (예를 들어, 스케일 값의 상호가역성 또는 스케일 값 및 가산된 오프셋 값들의 상호가역성)을 규정한다. 이러한 상호 가역적 연산에 도입된 에러들은 상호가역성이 생성된 모든 샘플 값들에 적용될 때 순방향 연산에서의 잠재적 에러들보다 더 현저해진다.

[0205]

성분 스케일링 SEI 메시지의 시맨틱들은 샘플 값들의 범위들의 도출 프로세스 (예를 들어, 스케일 및 오프셋 값들의 적용) 및 부동 소수점 연산의 관점에서 컬러 성분들의 각각의 범위의 코드워드들 (예를 들어 샘플 값들)의 매핑 프로세스를 나타낸다. 이는 상이한 컴퓨팅 시스템들에서 여러 부동 소수점 산술 구현에 기초하는 복원된 HDR 출력에서의 차이를 가져온다.

[0206]

본 출원은 비디오 코딩 표준들, 이를 테면, H.265/HEVC, H.264/AVC, BDA, MPEG 또는 기타의 것에서 규정될 수도 있는 SEI 시그널링 및 프로세싱 또는 다른 유사한 시그널링 기법을 이용하여 성분 스케일링 정보의 통신을 개선하는 수개의 기술들을 설명한다. 다음 양태들 중 하나 이상이 독립적으로 적용될 수도 있거나 또는 임의의 특정 예에서 이를 양태들의 다른 것과 적절하게 조합하여 적용될 수도 있다.

- [0210] 일반적으로, 본 개시물은 비디오 인코더 (20) 및/또는 비디오 프리프로세서 부 (19) 가 하나 이상의 샘플 값 범위들의 샘플 값들 (예를 들어, 컬러 성분 값들)에 대한 스케일 값을 시그널링하도록 구성될 수도 있는 기법을 설명한다. 스메일 값은 비디오 디코더 (30) 및 비디오 포스트프로세서 부 (31) 가 매핑 프로세스를 수행하여, 입력 샘플 값을 포함하는 특정 샘플 값 범위에 대하여 규정된 스케일 값을 입력 샘플 값과 곱한 다음, 성분 스케일링 정보의 부분으로서 파라미터들에 기초하여 컴퓨팅된 오프셋을 가산하는 것에 의해 출력 샘플 값을 성분의 입력 샘플 값으로부터 획득하도록 구성될 수도 있도록 규정된다.
- [0211] 본 개시물의 다른 예에서, 컬러 성분의 코드워드들의 범위들의 사이즈 및 수를 컴퓨팅하도록 고정 소수점 구현을 이용하기 보다는, 비디오 인코더 (20) 및/또는 비디오 프리프로세서 부 (19) 는 고정 소수점 컴퓨팅 구현을 이용하여 컬러 성분의 코드워드들의 범위들의 수 및 사이즈를 도출하도록 구성될 수도 있다. 예를 들어, 비디오 인코더 (20) 및/또는 비디오 프리프로세서 부 (19) 는 동적 범위 조정 매핑 프로세스의 파라미터들을 결정하고 적용하기 위한 분수 비트들의 미리 정해진 수를 이용하도록 구성될 수도 있다. 분수 비트들의 수는 동적 범위 조정 프로세스의 각각의 파라미터들 (예를 들어, 각각의 컬러 성분 (코드워드)에 대한 값들의 범위, 스케일 값, 및 오프셋 값)마다 상이할 수도 있다.
- [0212] 예를 들어, 비디오 프리프로세서 부 (19) 는 컬러 성분에 대한 코드워드들의 범위들의 수 및 사이즈를 통신하는데 이용되는 임의의 파라미터들 또는 신팩스 엘리먼트 (예를 들어, `hdr_recon_num_ranges[c]`)에 대한 정수 연산들을 수행하도록 구성될 수도 있다. 비디오 프리프로세서 부 (19) 는 이용된 부동 소수점 구현에서의 범위들의 수 및 사이즈의 임의의 계산의 분수 부분에 의해 이용된 비트들의 수의 트랙을 유지하도록 구성될 수도 있다. 비디오 프리프로세서 부 (19) 및/또는 비디오 인코더 (20) 는 SEI 메시지 (예를 들어, `hdr_recon_offset_frac_bit_depth`, `hdr_recon_scale_frac_bit_depth`)에서의 분수 부분에 이용된 비트들의 수를 시그널링하도록 구성될 수도 있거나 또는 분수 부분에 이용된 비트들의 수는 미리 정해진 값일 수도 있다. 비디오 디코더 (30) 는 분수 부분에서 비트들의 수를 나타내는 SEI 메시지에서 신팩스 엘리먼트를 디코딩하도록 구성될 수도 있고, 비디오 포스트 프로세서 부 (31) 는 역 동적 범위 조정 프로세스의 파라미터들 중 하나 이상에 대한 분수 부분에서의 비트들의 동일 수를 이용하여 역 동적 범위 조정을 수행하도록 구성될 수도 있다.
- [0213] 본 개시물의 일 예에서, 매핑 프로세스에 대한 범위들 및/또는 다른 파라미터들을 결정할 때, 비디오 디코더 (30) 및/또는 비디오 포스트 프로세서 부 (31) 는, 상이한 파라미터들의 시그널링된 분수 비트 깊이들이 상이하도록 이러한 파라미터들을 결정하도록 구성될 수도 있고, 파라미터들에 대하여 수행된 계산들의 정확도는 가능한 멀리 리테인된다. 예를 들어, 비디오 디코더 (30) 및/또는 비디오 포스트프로세서 부 (31) 는 특정 파라미터를 결정하는데 이용되는 임의의 중간 계산 단계들에서 분수 비트들의 수를 누적하는 것에 의해 최소 값으로 라운딩하는 것에 기인하는 임의의 에러들을 리테인하도록 구성될 수도 있다. 그 후, 비디오 디코더 (30) 및/또는 비디오 포스트프로세서 부 (31) 는 특정 파라미터를 결정 및/또는 계산하는 최종 단계에서 원하는 분수 정확도로 특정 파라미터의 최종 값을 가져오도록 클립핑 프로세스를 수행할 수도 있다. 다른 예에서, 파라미터들의 시그널링 분수 비트 깊이들이 동일할 때, 비디오 디코더 (30) 및/또는 비디오 포스트 프로세서 부 (31) 는 중간 단계에서 분수 비트들의 수를 누적하고, 최종 단계(들)에서 원하는 정확도로 파라미터의 최종 값을 가져오도록 클립핑 동작을 수행하도록 구성될 수도 있다.
- [0214] 다른 예에서, 비디오 디코더 (30) 및/또는 비디오 포스트프로세서 부 (31) 는 계산 프로세스의 하나 이상의 중간 단계들에서 파라미터들의 값, 또는 파라미터에 대하여 획득된 값들의 분수 부분들이 미리 정해진 값으로 감소되도록 하는 파라미터의 값을 클립핑 및/또는 절삭하도록 구성될 수도 있다. 즉, 임의의 클립핑을 수행하도록 파라미터에 대한 최종 값을 결정할 때까지 기다리기 보다는, 비디오 디코더 (30) 및/또는 비디오 포스트프로세서 부 (31) 는 파라미터를 결정하도록 수행된 계산들의 중간 값들을 클립핑할 수도 있다. 이러한 클립핑 또는 절삭은 SEI 메시지에서 표시된 분수 비트들의 수에 기초할 수도 있다. 다른 예에서, 비디오 디코더 (30) 및/또는 비디오 포스트프로세서 부 (31) 는 연산/단계가 클립핑 없이 동작되면 분수 비트들의 누적된 수가 특정의 미리 정해진 값, 예를 들어, 중간 값들을 저장하는데 이용되는 레지스터들의 비트 깊이를 초과한다고 결정될 때 특정 연산/단계 전에 특정 파라미터를 계산할 때 이용되는 중간 값들을 클립핑 및/또는 절삭하도록 구성될 수도 있다.
- [0215] 본 개시물의 다른 예에서, 비디오 프리프로세서 부 (19) 및/또는 비디오 포스트프로세서 부 (31) 는 컬러 성분들의 고정 표현에 대하여 정의되는 정의된 최소 값 및 최대 값에 기초하여 미리 정해진 샘플 값 범위들에 따라 스케일, 오프셋 및 범위 값들을 도출하도록 구성될 수도 있다. 예를 들어, 컬러 성분들의 고정 표현은 예를 들어, 값들의 "표준" 범위, 값들의 "전체" 범위, 및 값들의 "제한된" 범위로서 정의된 값들의 복수의 범위들을 가질 수도 있다. 값들의 "전체" 범위는 값들의 "표준" 범위 (예를 들어, YCbCr 컬러 공간의 8-비트 표준 범

위 표현에 대해, Y 성분은 16 내지 235 의 범위에서의 값을 취하며, 그리고 Cb 및 Cr 성분은 16 내지 240 의 값들의 범위를 취한다)에 비해 특정 성분의 최소 값과 최대 값 사이의 더 큰 범위를 가질 수도 있다 (예를 들어, YCbCr 컬러 공간의 8-비트 전체 범위 표현에 대해, Y, Cb, 및 Cr 성분은 0 내지 255 의 값들의 범위를 취한다).

값들의 "제한된" 범위는 값들의 "표준" 범위에 비해, 특정 성분의 최소 값과 최대 값 사이의 보다 적은 범위를 가질 수도 있다 (예를 들어, YCbCr 컬러 공간의 10-비트 표준 범위 표현에 대해, Y, Cb, 및 Cr 성분들은 4 내지 1019 의 범위에서의 값을 취한다).

[0216] 일 예에서, 비디오 인코더 (20) 및/또는 비디오 프리프로세서 부 (19) 는 어떤 샘플 범위가 이용되는지 (예를 들어, 전체, 제한된, 표준, 또는 기타)에 기초하여 샘플들의 최소 및 최대 허용된 값들 (예를 들어, 컬러 성분 값들) 을 비디오 디코더 및/또는 비디오 포스트프로세서 부 (31)에 표시하도록 (예를 들어, SEI 메시지에서의) 신택스 엘리먼트를 시그널링하도록 구성될 수도 있다. 다른 예에서, 비디오 인코더 (20) 및/또는 비디오 프리프로세서 부 (19) 는 어떤 샘플 범위가 이용되는지 (예를 들어, 전체, 제한된, 표준)에 기초하여 비디오 디코더에 샘플들의 최소 및 최대 허용된 값들을 표시하도록 (예를 들어, SEI 메시지에서) 하나 이상의 신택스 값들을 시그널링하도록 구성될 수도 있다. 비디오 디코더 (30) 및/또는 비디오 포스트프로세서 부 (31)는 그 후, 수신된 최소 값 및 수신된 최대 값에 기초하여 역 동적 범위 조정 프로세스에 허용되는 성분 값들의 범위를 결정할 수도 있다.

[0217] 다른 예에서, 비디오 인코더 (20) 및/또는 비디오 프리프로세서 부 (19)는 스케일 값들이 사인되는지 또는 비사인되는지를 표시하기 위해 (예를 들어, SEI 메시지에서) 플래그를 시그널링하도록 구성될 수도 있다. 이 예에서, 임의의 SEI 메시지들의 파싱 프로세스는 플래그의 값과 무관하게 동일하다.

[0218] 다음 장은 이전 장에서 개시된 예시적 기법들을 이용하는 실시형태들의 수개의 예들을 포함한다. 이 실시형태에서, 성분 스케일링 함수는 루업 테이블로서 시그널링되고, 루업 테이블을 정의하는 포인트들을 시그널링하는데 이용된 비트들의 수가 시그널링된다. 일 예에서, 루업 테이블은 피스와이즈 선형 매핑 함수를 정의한다. 루업 테이블에 대한 포인트들은 피스와이즈 선형 매핑을 정의하는 (x,y) 좌표들에 대응한다. 명시적 점들이 시그널링되지 않는 샘플 값들에 대해, 값은 이웃하는 피봇 점에 기초하여 보간된다.

[0219] 범위들 및 출력 샘플 값들의 도출 프로세스는 아래와 같이 정의된다.

[0220] 성분 c로부터의 샘플 x의, 샘플 y = map[c](x)로의 매핑은 다음과 같이 규정된다:

[0221] - DefaultPrecShift의 값을 9와 동일하게 설정한다.

[0222] - 변수들 minSampleVal 및 maxSampleVal은 컨텐츠의 샘플 범위에 의해 정의된 바와 같이 최소 및 최대 샘플 값들을 나타내는 것으로 한다.

[0223] - 0 내지 hdr_recon_num_ranges[c] - 1의 범위에서의 i에 대해 변수ScaleValue[c][i]는 다음과 같이 도출된다:

```

SignValue[ c ][ i ] = 0 // 0 for positive, 1 for negative
hdrReconScaleBitDepth = hdr_recon_scale_bit_depth -
( hdr_negative_scales_present_flag ? 1 : 0 )
if( hdr_negative_scales_present_flag )
    ScaleValue[ c ][ i ] = hdr_recon_scale_val[ c ][ i ] & ((1 <<
hdrReconScaleBitDepth ) - 1 )      (D-xx)
    SignValue[ c ][ i ] = hdr_recon_scale_val[ c ][ i ] & ( 1 <<
hdrReconScaleBitDepth )
else
    ScaleValue[ c ][ i ] = hdr_recon_scale_val[ c ][ i ] (D-xx)
shiftInvScale = 1 << hdrReconScaleBitDepth
InvScaleValue[ c ][ i ] = ( 1 << (DefaultPrecShift + hdrReconScaleBitDepth)
+ shiftInvScale ) /
ScaleValue[ c ][ i ]

```

[0224] - 0 내지 hdr_recon_num_ranges[c]-1의 범위에서의 변수 RangeValue[c][i]는 다음과 같이

도출된다:

[0226] - `hdr_recon_equal_ranges_flag[c] <= 0` 과 같으면, 다음이 적용된다:

[0227] `RangeValue[c][i] = hdr_recon_range_val[c][i]` (D-xx)

[0228] - 그렇지 않으면 (`hdr_recon_equal_ranges_flag[c] > 1` 과 같으면), 다음이 적용된다:

```
RangeValue[c][i] = ( ( InputDynamicRangeValue <<
    hdr_recon_offset_frac_bit_depth ) +
    ( ( hdr_recon_num_ranges[c] + 1 ) >> 1 ) ) /
    hdr_recon_num_ranges[c]
```

(D-xx)

[0229] 여기에서, 샘플 범위가 0에서 1로 정규화될 때 `InputDynamicRangeValue`은 1과 같다.

[0231] - 0 내지 `hdr_recon_num_ranges[c]`의 범위에서의 `i`에 대해, 변수들 `InputRanges[c][i]` and `OutputRanges[c][i]`은 다음과 같이 도출된다:

[0232] - `i`가 0과 같을 때, 다음이 적용된다:

```
OutputRanges[c][i] = -hdr_recon_global_offset_val[c] *
    InvScaleValue[c][i-1]
```

(D-xx)

[0233] `InputRanges[c][i] = 0`

[0234] - 그렇지 않으면 (`i`가 0과 같지 않으면), 다음이 적용된다:

```
InputRanges[c][i] = InputRanges[c][i-1] + RangeValue[c][i-1]
```

(D-xx)

```
OutputRanges[c][i] = OutputRanges[c][i-1] +
    RangeValue[c][i-1] * InvScaleValue[c][i-1]
```

(D-xx)

[0235]

[0236] - 0 내지 `hdr_recon_num_ranges[c]-1`의 범위에서의 `i`에 대해 파라미터들 `OffsetValue[c][i]`은 다음과 같이 도출된다:

```
precOffsetDeltaBits = DefaultPrecShift + hdr_recon_scale_frac_bit_depth
OffsetValue[c][i] = InputRanges[c][i+1] * (1 << precOffsetDeltaBits)
```

- `OutputRanges[c][i+1] * ScaleValue[c][i-1]` (D-xx)

```
OffsetValue[c][i] = ( ( OffsetValue[c][i] + ( 1 << (BitDepth - 1) ) ) >>
    BitDepth ) *
```

`(maxSampleVal - minSampleVal)`

[0237] - 파라미터 `y = map[c](x)`는 다음과 같이 도출된다:

[0239] - 변수 `bitDepthDelta`는 `DefaultPrecShift +`

[0240] `hdr_recon_offset_frac_bit_depth - BitDepth`와 같게 설정된다:

[0241] - `(x << bitDepthDelta)`가 `OutputRanges[c][0]` 이하이면, 다음이 적용된다:

[0242] `y = InputRanges[c][0]` (D-xx)

[0243] `fracBitDepth = hdr_recon_offset_frac_bit_depth`

- 그렇지 않고 (`x << bitDepthDelta`) 가

[0244] OutputRanges[c][hdr_recon_num_ranges[c]] 보다 더 크면, 다음이 적용된다:

```
y = InputRanges[ c ][ hdr_recon_num_ranges[ c ] ](D-xx)
```

```
fracBitDepth = hdr_recon_offset_frac_bit_depth
```

[0245] - 그렇지 않으면 다음이 적용된다:

```
fracBitDepth = DefaultPrecShift + hdr_recon_scale_frac_bit_depth +
hdr_recon_offset_frac_bit_depth - BitDepth
for( i = 1; i <= hdr_recon_num_ranges[ c ]; i++ )
    if( OutputRanges[ i - 1 ] < ( x << bitDepthDelta ) &&
        ( x << bitDepthDelta ) <= OutputRanges[ i ] ) {
        rangeBitShift = DefaultPrecShift +
        hdr_recon_offset_frac_bit_depth - BitDepth
        y = ( x - minSampleVal ) * ScaleValue[ c ][ i - 1 ] * ( 1 <<
        rangeBitDepth ) +
        OffsetValue[ c ][ i - 1 ] +
        minSampleVal * ( 1 << fracBitDepth ) (D-xx)
    }
}
```

[0246] - fracShiftOffset = 1 << (fracBitDepth - 1)

```
y = ( y + fracShiftOffset ) >> fracBitDepth
```

[0247]

[0248] 대안으로서, minSampleVal 및 maxSampleVal에 기초한 샘플 범위의 조정은 다음과 같이 InputRanges 및 OutputRanges에 대해서기 보다는, OffsetValue에 대해 수행된다:

[0249] deltaSampleVal = maxSampleval - minSampleVal

```
deltaBitShift = DefaultPrecShift + hdr_recon_offset_frac_bit_depth
sampleShift = ( 1 << ( BitDepth - 1 ) ) + ( minSampleVal << deltaBitShift ) )
OutputRanges[ c ][ i ] = ( ( OutputRanges[ c ][ i ] * deltaSampleVal ) +
sampleShift ) >>
BitDepth
```

```
deltaBitShift = DefaultPrecShift + hdr_recon_offset_frac_bit_depth
sampleShift = ( 1 << ( BitDepth - 1 ) ) + ( minSampleVal << deltaBitShift ) )
InputRanges[ c ][ i ] = ( ( InputRanges[ c ][ i ] * deltaSampleVal ) + sampleShift )
>>
BitDepth
```

[0250]

[0251] 본 개시물은 비디오 코딩 표준들, 이를 테면, H.265/HEVC, H.264/AVC, BDA, MPEG 또는 기타의 것에서 규정될 수도 있는 SEI 시그널링 및 프로세싱 또는 다른 유사한 시그널링 기법을 이용하여 성분 스케일링 정보의 캐리지를 개선하는 수개의 기법들을 설명한다. 이들 기법들 중 하나 이상이 독립적으로 적용될 수도 있거나 또는 다른 것과 적절하게 조합하여 적용될 수도 있다. 추가로, 동적 범위 프로세스의 고정 소수점 구현을 수행하기 위하여 SEI 메시지들에서 정보를 시그널링 및/또는 이용하는 위에 설명된 기법들은 정보를 시그널링/수신하기 위하여 아래 설명된 선택스 구조들 중 하나 이상을 이용할 수도 있다.

[0252] 일부 예들에서, 각각의 성분에 대해, 비디오 인코더 (20)는 동적 범위 조정 후에 제 1 오프셋 값이 매핑되는 조정된 값을 특정하는 제 2 오프셋 값을 포함하는 하나 이상의 SEI 메시지들을 시그널링할 수도 있다. 비디오 디코더 (30)는 이러한 SEI 메시지들을 수신하며, 그 정보를 파싱 및/또는 디코딩하고, 그 정보를 비디오 포스트프로세서 (31)에 전달할 수도 있다.

[0253] 다른 예에서, 제 1 글로벌 오프셋 값 또는 제 2 글로벌 오프셋 값 중 어느 것도 SEI 메시지로 시그널링되지 않

는다. 대신, 디코더 (30) 는 제 1 글로벌 오프셋 및 제 2 글로벌 오프셋의 값들이 디코더 (30) 가 시퀀스마다 결정하거나 또는 외부 수단에 의해 수신하는 일정한, 미리 결정된, 또는 시그널링된 값이라고 가정한다. 다른 예에서, 비디오 인코더 (20) 는 SEI 메시지에서 제 1 글로벌 오프셋 값을 시그널링하지만, 제 2 글로벌 오프셋 값은 SEI 메시지에서 시그널링되지 않는다. 대신, 비디오 디코더 (30) 는 그것의 값이 디코더 (30) 가 시퀀스마다 결정하거나 또는 외부 수단에 의해 수신하는 일정하거나, 미리 결정되거나, 또는 시그널링되는 값이라고 유추한다. 더 추가의 예에서, 비디오 인코더 (20) 는 SEI 메시지에서 제 2 글로벌 오프셋 값을 시그널링하지만, 제 1 글로벌 오프셋 값은 SEI 메시지에서 시그널링되지 않는다. 대신, 비디오 디코더 (30) 는 제 1 글로벌 오프셋 값이 디코더 (30) 가 시퀀스마다 결정하거나 또는 외부 수단에 의해 수신하는 일정하거나, 미리 결정되거나, 또는 시그널링되는 값이라고 유추한다.

[0254] 일부 예들에서, 비디오 인코더 (20) 는 디코더 (30) 에 의해 수신되고, 디코더 (30) 에 의해 사용되는 오프셋 값들을, 글로벌 및 로컬 스케일 및 오프셋 값을 둘 다, 뿐만 아니라 비조정된 값들의 범위의 파티션들, 및 조정된 값들의 범위의 파티션들을 포함하는 다른 글로벌 또는 로컬 파라미터들을 도출하기 위해 시그널링할 수도 있다.

[0255] 일부 예들에서, 비디오 인코더 (20) 는 동적 범위 조정 동안 입력 표현 값들 (즉, 성분 값들) 의 범위가 나누어 졌던 파티션들의 수를 포함하는 하나 이상의 SEI 메시지들을 시그널링할 수도 있다. 하나의 예에서, 파티션들의 수는 2의 거듭제곱 (즉, 1, 2, 4, 8, 16 등) 으로 제한될 수도 있고 파티션들의 수는 로그로서 시그널링된다 (예컨대, $3 = \log_2 8$ 일 때 8 개 파티션들이 시그널링된다). 비디오 디코더 (30) 는 이러한 SEI 메시지를 수신하며, 그 정보를 파싱 및/또는 디코딩하고, 그 정보를 비디오 포스트프로세서 (31) 에 전달할 수도 있다.

[0256] 일부 예들에서, 크로마 성분들에 대한 파티션들의 수는 루마 성분에 대한 파티션들의 수와는 상이할 수도 있다. 파티션들의 수는 $2 + 1$ 의 거듭제곱으로 제한되고 로그로서 시그널링되고 마이너스 0으로 반올림될 수도 있다. 이런 식으로, 중간색 (neutral) 크로마를 갖는 화소들이 자신들 소유의 값들을 가질 수 있고 그 파티션의 사이즈는 다른 파티션들보다 더 작을 수 있다. 이러한 일 예에서, 중간색 크로마는 중간-값 (예컨대, 크로마 값들이 10-비트 표현에서 -0.5와 0.5 사이, 또는 -512와 511 사이의 범위일 때 0) 주위의 크로마의 값을 지칭할 수도 있다. 2의 거듭제곱으로서 파티션들의 수를 제한하는 것은 인코더 (20) 가 비트들을 절약하는 것을 가능하게 할 수도 있는데, 인코더 (20) 는 정수 값들에 대해 실제 값보다 더 적은 비트수로 값의 로그를 나타낼 수도 있기 때문이다. 파티션들의 수를 $2 + 1$ 의 거듭제곱으로 제한하는 것은 적어도 하나의 파티션이 중간색 크로마 값들에 전용될 수도 있는 것과, 일부 예들에서, 중간색 크로마 값들에 대응하는 파티션의 폭이 나머지보다 더 작을 수도 있는 것을 보장할 수도 있다. 다른 예들에서, 그런 파티션은 다른 파티션들 중 하나 이상의 파티션보다 더 클 수도 있다.

[0257] 일부 예들에서, 디코더 (30) 는, 글로벌 및 로컬 스케일 및 오프셋 값을 둘 다, 뿐만 아니라 비조정된 성분 값들의 범위의 파티션들의 실제 사이즈 및/또는 조정된 성분 값들의 범위의 파티션들의 사이즈를 포함하는 다른 글로벌 또는 로컬 파라미터들을 도출하기 위해 시그널링된 수의 파티션들을 사용할 수도 있다.

[0258] 일부 예들에서, 인코더 (20) 는, 각각의 파티션에 대해, 입력 성분 값들 및 대응하는 매핑 출력 성분 값들의 범위를 특정하는 로컬 스케일 및 로컬 오프셋 값을 포함할 수도 있는 하나 이상의 SEI 메시지들을 시그널링할 수도 있다. 일부 예들에서, 인코더 (20) 는 스케일 및 오프셋들을 시그널링하기 위해 신택스 엘리먼트들에 의해 사용되는 비트들의 수를 포함하는 SEI 메시지를 시그널링할 수도 있다. 다른 예들에서, 인코더 (20) 는 신택스 엘리먼트들에서 스케일 및 오프셋들의 소수 부분을 표현하는데 사용되는 비트들의 수를 나타내는 SEI 메시지를 시그널링할 수도 있다. 다른 예들에서, 인코더 (20) 는 스케일 파라미터들의 정수 부분이 부호 있는 표현으로 시그널링됨을 나타내는 하나 이상의 SEI 메시지를 또는 신택스 엘리먼트들을 시그널링할 수도 있다.

일부 예들에서, 부호 있는 표현은 2의 보수이다. 다른 예들에서, 부호 있는 표현은 부호 있는 크기 표현이다. 비디오 디코더 (30) 는 이러한 SEI 메시지를 및/또는 신택스 엘리먼트들을 수신하며, 그 정보를 파싱 및/또는 디코딩하고, 그 정보를 비디오 포스트프로세서 (31) 에 전달할 수도 있다.

[0259] 다른 예들에서, 인코더 (20) 는 각각의 오프셋 값을 연속적으로 사용하여, 조정된 성분 또는 표현 값들의 범위를 먼저 컴퓨팅한 다음, 스케일 값을 사용하여, 비조정된 표현에서의 대응하는 범위를 컴퓨팅할 수도 있다. 예를 들어, 하나의 오프셋 값은, 조정된 성분에 대해 도출되거나 또는 시그널링되는 글로벌 오프셋 값의 값을 사용하여, 뒤이어서 스케일 값 및 조정된 표현의 제 1 파티션의 범위를 사용하여 조정된 성분에서의 제 1 파티션의 범위를 컴퓨팅하여, 비조정된 표현의 대응하는 파티션에서의 범위를 도출하는데, 그리고 조정된 것의 제 1

파티션 및 비조정된 표현의 대응하는 파티션의 각각의 범위들로, 조정된 범위의 제 1 파티션 및 파티션들의 경계를 나타내는 비조정된 표현들의 대응하는 파티션에 대해 도출되는 각각의 값을 도출하는데 사용될 수도 있다.

이에 뒤이어, 다른 오프셋 값은, 이전의 단계에서 도출된 조정된 성분에서의 제 1 파티션의 경계 값을 사용하여, 뒤이어서 스케일 값 및 조정된 표현의 제 2 파티션의 범위를 사용하여, 조정된 성분에서의 제 2 파티션의 범위를 컴퓨팅하여, 비조정된 표현의 범위를 도출하는데, 그리고 조정된 표현의 제 2 파티션들 및 비조정된 표현들의 대응하는 파티션의 각각의 범위들로, 각각의 값을 각각의 표현들의 경계를 나타내는 조정된 및 비조정된 표현들에 대해 도출하는데 사용될 수도 있다. 이 방법은 모든 범위들 및 경계들이 조정된 및 비조정된 표현들의 모든 파티션들에 대해 도출되기까지 반복될 수도 있다. 다른 예에서, 인코더 (20) 는 각각의 오프셋 값을 연속적으로 사용하여 비조정된 성분 또는 표현 값들의 범위를 먼저 컴퓨팅한 다음, 스케일 값을 사용하여, 조정된 표현에서의 대응하는 범위를 컴퓨팅할 수도 있다. 다르게 말하면, 스케일 및 오프셋 값들이 적용되는 성분 또는 표현은 비조정된 표현과 조정된 표현 사이에서 스와핑될 수 있다.

[0260] 일부 예들에서, 스케일 및 오프셋 값들을 시그널링하기 위해 신택스 엘리먼트들에 의해 사용되는 비트들의 수는 성분에 따라 달라질 수도 있다. 다른 예들에서, 이들 수들이 명시적으로 시그널링되지 않을 때 디폴트 비트 수가 정의되고 사용된다.

[0261] 일부 예들에서, 인코더 (20) 는 출력 표현들 (즉, 출력 성분들) 의 파티션들의 길이들이 동일한지의 여부를 나타내는 신택스 엘리먼트를 시그널링할 수도 있다. 이러한 일 예에서, 인코더 (20) 는 하나 이상의 파티션들에 대한 오프셋 값을 시그널링하지 않을 수 있다. 디코더 (30) 는 일부 예들에서 오프셋 값들을 동일한 것으로 유추할 수도 있다. 다른 예에서, 디코더 (30) 는 파티션들이 동일한 길이로 된다고 가정할 수도 있고 그렇게 나타내는 신택스 엘리먼트를 수신하지 않을 수도 있다. 일부 예들에서, 디코더 (30) 는 시그널링된 신택스 엘리먼트들 및 표현의 미리 정의된 전체 동적 범위로부터 각각의 파티션의 사이즈를 도출할 수도 있다.

[0262] 다른 예들에서, 각각의 파티션에 대한 피벗 포인트들 뿐만 아니라 각각의 파티션에 대한 스케일 및 오프셋 값을 시그널링하는 것이 아니라, 비디오 인코더 (20) 는 하나 이상의 또는 모든 파티션들의 사이즈와 함께 각각의 파티션에 대한 스케일 값 또는 과생물 (derivative) 을 나타내는 하나 이상의 SEI 메시지들을 시그널링할 수도 있다. 이 접근법은 인코더 (20) 가 각각의 파티션에 대해 로컬 오프셋 값을 시그널링하는 것을 피하는 것을 허용할 수도 있다. 대신, 일부 예들에서, 인코더 (20) 는, 하나 이상의 SEI 메시지들에서, 하나 이상의 파티션들에 대한 파티션 사이즈 및 스케일 값 (또는 과생물) 을 시그널링할 수도 있다. 각각의 파티션 또는 파티셔닝에 대한 로컬 오프셋 값 (이는 더 높은 정확도를 요구할 수도 있음) 은 디코더 (30) 에 의해 결정 또는 도출될 수도 있다.

[0263] 일부 예들에서, 인코더 (20) 는 특정한 파티션들에 대한 오프셋 및 스케일 값을 위한 여러 디폴트 값을 특정하는 모드 값을 나타내는 하나 이상의 SEI 메시지들을 시그널링할 수도 있다. 비디오 디코더 (30) 는 이러한 SEI 메시지들을 수신하며, 그 정보를 파싱 및/또는 디코딩하고, 그 정보를 비디오 포스트프로세서 (31) 에 전달할 수도 있다.

[0264] 일부 예들에서, 인코더 (20) 는 성분들의 서브세트의 퍼시스턴스 (persistence) 가 정의될 수도 있고 성분들의 서브세트의 성분 스케일 값을 업데이트될 수도 있도록 SEI 메시지의 퍼시스턴스를 정의하는 값을 나타내는 하나 이상의 SEI 메시지들을 시그널링할 수도 있다. SEI 메시지의 퍼시스턴스는 SEI의 인스턴스에서 시그널링 되는 값들이 적용될 수도 있는 꽉쳐들을 나타낸다. 일부 예들에서, SEI 메시지의 퍼시스턴스는 SEI 메시지들의 하나의 인스턴스에서 시그널링되는 값들이 SEI 메시지가 적용되는 꽉쳐들의 모든 성분들에 대응하게 적용될 수도 있도록 정의된다. 다른 예들에서, SEI 메시지의 퍼시스턴스는 SEI의 하나의 인스턴스에서 시그널링 되는 값들이 성분들의 서브세트에 대응하게 적용됨이 나타내어질 수도 있도록 정의되며, SEI의 인스턴스에서의 값들이 적용되지 않는 성분들은 값을 이용 불가능하게 할 수도 있거나 또는 SEI 메시지의 다른 인스턴스에서 시그널링되는 값들을 이용 가능하게 할 수도 있다. 비디오 디코더 (30) 는 이러한 SEI 메시지들을 수신하며, 그 정보를 파싱 및/또는 디코딩하고, 그 정보를 비디오 포스트프로세서 (31) 에 전달할 수도 있다.

[0265] 일부 예들에서, 인코더 (20) 는 디코딩된 출력에 대해 수행될 포스트프로세싱 단계들을 나타내는 신택스 엘리먼트들을 포함하는 하나 이상의 SEI 메시지들을 시그널링할 수도 있다. 각각의 신택스 엘리먼트는 특정 프로세스 (예컨대, 스케일링 성분들, 컬러 변환들, 업-샘플링/다운-샘플링 필터들 등) 와 연관될 수도 있고 신택스 엘리먼트의 각각의 값을 그 프로세스에 연관된 파라미터들의 특정 세트가 사용되는 것을 특정할 수도 있다. 일부 예들에서, 그 프로세스에 연관된 파라미터들은 비디오 인코더 (20) 에 의해 비트스트림의 일부인 SEI 메시지들을 사용하여, 또는 다른 수단을 통해 송신될 수도 있는 메타데이터로서 시그널링된다. 비디오 디코더

(30) 는 이러한 SEI 메시지들을 수신하며, 그 정보를 파싱 및/또는 디코딩하고, 그 정보를 비디오 포스트프로세서 (31)에 전달할 수도 있다.

[0266] 일부 예들에서, 인코더 (20)는 입력 표현들 (즉, 입력 성분 값들)을 출력 표현들 (즉, 출력 성분 값들)에 매핑하기 위한 조각적 (piece-wise) 선형 모델 함수를 설명 및/또는 구축하기 위해 사용될 수도 있는 선택스 엘리먼트들 또는 하나 이상의 SEI 메시지들을 시그널링할 수도 있다. 비디오 디코더 (30)는 이러한 SEI 메시지를 수신하며, 그 정보를 파싱 및/또는 디코딩하고, 그 정보를 비디오 포스트프로세서 (31)에 전달할 수도 있다. 다른 예들에서, 미리 정의된 가정들이 입력 표현들을 출력 표현에 매핑하기 위한 조각적 선형 모델 함수를 설명 및/또는 구축하기 위해 사용될 수도 있다.

[0267] 일부 예들에서, 인코더 (20)는 SEI 메시지에서 시그널링되는 스케일 및 오프셋 파라미터들이 제 2 성분의 상이한 값들의 함수로서 제 1 성분에 적용될 스케일의 변동을 표현함을 나타내는 하나 이상의 선택스 엘리먼트들을 포함할 수도 있는 하나 이상의 SEI 메시지들을 시그널링할 수도 있다.

[0268] 일부 예들에서, 인코더 (20)는 제 2 성분의 상이한 값들의 함수로서 제 1 성분에 대한 스케일과 함께 적용되거나 또는 적용될 수도 있는 오프셋 파라미터들을 나타내는 하나 이상의 SEI 메시지들을 시그널링할 수도 있다.

일부 예들에서, 인코더 (20)는 제 2 성분의 상이한 값들의 함수로서 제 1 성분에 대한 스케일과 함께 적용되거나 또는 적용될 수도 있는 오프셋 파라미터들을 나타내는 하나 이상의 추가적인 선택스 엘리먼트들을 포함할 수도 있는 하나 이상의 SEI 메시지들을 시그널링할 수도 있다. 비디오 디코더 (30)는 이러한 SEI 메시지를 수신하며, 그 정보를 파싱 및/또는 디코딩하고, 그 정보를 비디오 포스트프로세서 (31)에 전달할 수도 있다.

[0269] 일부 예들에서, 인코더 (20)는 디코더 측에서 사용되는 전기 광학 전달 함수 특성들이 전기 광학 전달 함수 특성들의 제 1 세트와 유사할 때 SEI 메시지에서 시그널링되는 스케일, 오프셋 및 범위 파라미터들이 적용되도록 하는 전기 광학 전달 함수 특성들의 제 1 세트를 나타내는 제 1 선택스 엘리먼트를 포함하는 하나 이상의 SEI 메시지들을 시그널링할 수도 있다.

[0270] 다른 예에서, 인코더 (20)는 전기 광학 전달 함수 특성들의 제 1 세트, 또는 유사한 특성들을 갖는 것들이 디코더 (30)에 의해 사용될 때 SEI 메시지(들)에서 시그널링되는 오프셋, 스케일 및 범위 파라미터들이 HDR 출력의 최상의 복원을 위해 적용됨을 나타내는 하나 이상의 SEI 메시지들을 시그널링할 수도 있다. 비디오 디코더 (30)는 이러한 SEI 메시지들을 수신하며, 그 정보를 파싱 및/또는 디코딩하고, 그 정보를 비디오 포스트프로세서 (31)에 전달할 수도 있다.

[0271] 다른 예에서, 인코더 (20)는 광-전자 전달 함수 특성들의 제 1 세트와, 시그널링되는 스케일, 오프셋 및 범위 파라미터들이, 대응하는 역 전기 광학 전달 함수 특성들이 디코더 측에서 적용될 때, 디코더 (30)에 의해 적용됨을 나타내는 하나 이상의 SEI 메시지들을 시그널링할 수도 있다. 비디오 디코더 (30)는 이러한 SEI 메시지를 수신하며, 그 정보를 파싱 및/또는 디코딩하고, 그 정보를 비디오 포스트프로세서 (31)에 전달할 수도 있다.

[0272] 다른 예들에서, 인코더 (20)는 전기-광학/광-전자 특성들의 상이한 세트를 나타내고 현재 픽처에 적용 가능한 하나를 초과하는 SEI 메시지가 존재할 때, 단지 하나의 SEI 메시지가 적용되도록 하는 조건을 시그널링할 수도 있다. 인코더는 상이한 유형들의 디코더들, 또는 상이한 능력들을 갖는 디코더들을 충족시키기 위해 전기-광학/광전자 특성들의 상이한 세트를 시그널링할 수도 있다. 예를 들어, 디코더 측의 일부 디스플레이들은 적절한 도메인의 코딩된 성분 값들을 선형 광으로 변환하기 위해 PT EOTF를 적용할 수도 있는 반면, 다른 디스플레이들, 예컨대, 레거시 디스플레이들은 선형 광으로 변환하기 위해 감마 EOTF를 적용할 수도 있다. 인코더가 전송하는 특정 특성을 갖는 각각의 SEI는 디스플레이들의 특정한 유형들에 적합하거나 또는 유익할 수도 있고 디스플레이들의 다른 유형들에는 적합 또는 유익하지 않을 수도 있으며, 예컨대, PQ EOTF 특성들을 갖는 SEI 메시지는 코딩된 비디오를 선형 광으로 변환하기 위해 PQ EOTF를 적용하는 디스플레이들에 적합할 수도 있다. 디코더 (30)는 SEI 메시지가 적용되어야 하는지를 결정하고, 애플리케이션 표준에 기초하여, 최종 사용자 디바이스에 기초하여, 수신된 신호에 기초하여, 또는 외부 수단을 통해 수신된 다른 표시에 기초하여 그런 선택을 한다. 예를 들어, 디코더 (30)는, 현재 픽처에 적용되는 제 1 SEI 메시지의 제 1 선택스 엘리먼트는 그 SEI 메시지가 PQ OETF의 역으로 적용될 것임을 나타내고, 현재 픽처에 적용되는 제 2 SEI 메시지의 제 1 선택스 엘리먼트는 그 SEI 메시지가 다른 전달 함수 (이를테면 BBC, 또는 PH)로 적용될 것임을 나타낸다고 결정할 수도 있고, 디코더 (30) 또는 최종 사용자 디바이스는 그 디바이스가 PQ EOTF를 사용하기 때문에 제 1 SEI 메시지의 파라미터들을 적용할 것을 선택할 수도 있다. 일부 예들에서, 디코더가 준수하는 애플리케이션 표

준이 특성들의 특정 세트를 갖는 SEI 메시지가 사용될 것임을 특정할 수도 있다.

[0273] 다른 예들에서, 인코더 (20) 는 전송 특성들의 다수의 세트들에 대응하는 파라미터들을 운반하는 SEI 메시지를 시그널링 할 수도 있다. 다른 예들에서, 인코더 (20) 는 그 목적을 위해 상이한 SEI 메시지들을 시그널링 할 수도 있다. 비디오 디코더 (30) 는 이러한 SEI 메시지들을 수신하며, 그 정보를 파싱 및/또는 디코딩하고, 그 정보를 비디오 포스트프로세서 (31) 에 전달할 수도 있다.

[0274] 일부 예들에서, 인코더 (20) 는 SEI 메시지의 적용가능성을 나타내는 선택스 엘리먼트를 포함하는 하나 이상의 SEI 메시지들을 시그널링 할 수도 있다. SEI 메시지의 적용가능성은 (1) 스케일들 및 오프셋들이 적용되는 성분들, (2) 성분 스케일링이 적용되는 포지션, 및/또는 (3) 추가적인 스케일링 파라미터들이 시그널링되는지의 여부를 비제한적으로 포함할 수도 있다.

[0275] 설명되는 바와 같이, 인코더 (20) 는 스케일들 및 오프셋들이 적용되는 성분들을 나타내는 선택스 엘리먼트를 포함하는 하나 이상의 SEI 메시지들을 시그널링 할 수도 있다. 다음에서는 이러한 애플리케이션의 여러 예들을 열거한다. 예를 들어, 선택스 엘리먼트의 하나의 값은 제 1 성분 인덱스에 대한 시그널링된 파라미터들이 RGB 성분들에 적용될 것임을 나타낼 수 있다. 다른 값은 제 1 성분 인덱스에 대한 시그널링된 파라미터들이 루마 성분에 적용될 것임과, 제 2 및 제 3 인덱스들에 대한 것들이 Cb 및 Cr 성분들에 적용될 것임을 나타낼 수도 있다. 다른 값은 제 1 성분 인덱스에 대한 시그널링된 파라미터들이 R, G 및 B 성분들에 적용될 것임과, 제 2 및 제 3 인덱스들에 대한 것들이 Cb 및 Cr 성분들에 적용될 것임을 나타낼 수도 있다. 다른 값은 처음 세 개의 인덱스들에 대한 시그널링된 파라미터들이 루마, Cb 및 Cr 성분들에 적용될 것임과, 나머지 인덱스들에 대응하는 것들이 컬러 정정을 위해 적용될 것임을 나타낼 수도 있다. 비디오 디코더 (30) 는 이러한 SEI 메시지들을 수신하며, 그 정보를 파싱 및/또는 디코딩하고, 그 정보를 비디오 포스트프로세서 (31) 에 전달할 수도 있다.

[0276] 또한 설명된 바와 같이, 인코더 (20) 는 성분 스케일링이 적용되는 포지션을 나타내는 선택스 엘리먼트를 포함하는 하나 이상의 SEI 메시지들을 시그널링 할 수도 있다. 여러 프로세스들이 디코더 측에서, 비디오의 디코딩 후, 그리고 비디오 포스트프로세서 (31) 에서 일어난다. SEI에 연관된 프로세스가 적용될 포지션을 나타내는 선택스 엘리먼트의 시그널링, 다르게 말하면 SEI에서의 정보를 사용하는 것과 연관된 프로세스의 선행 또는 후속 동작들의 임의의 서브세트의 표시는 비디오 디코더 (30) 또는 비디오 포스트프로세서 (31) 가 비디오를 프로세싱하는 것에 도움이 될 것이다. 예를 들어, 그런 선택스 엘리먼트는 성분 스케일링이, 예를 들어 업샘플링 전 또는 후의 YCbCr 성분들에 적용되는 포지션을 나타낼 수 있다. 다른 예에서, 선택스 엘리먼트는 성분 스케일링이 양자화 전에 디코더 측에 적용됨을 나타낼 수 있다. 비디오 디코더 (30) 는 이러한 SEI 메시지들을 수신하며, 그 정보를 파싱 및/또는 디코딩하고, 그 정보를 비디오 포스트프로세서 (31) 에 전달할 수도 있다.

[0277] 또한 설명되는 바와 같이, 인코더 (20) 는, 예컨대 컬러 정정을 위한, 스케일링 및 파라미터들의 추가적인 세트가 시그널링되는지의 여부를 나타내는 선택스 엘리먼트를 포함하는 하나 이상의 SEI 메시지들을 시그널링 할 수도 있다. 파라미터들의 추가적인 세트는 특정 색 영역에 맞도록 컬러 성분들을 매핑하는 컬러 정정을 위해, 또는 VUI에서의 transfer_characteristics 선택스 엘리먼트에 의해 나타내어지는 것과는 상이한 전달 함수가 적용될 때의 성분 값들의 정정을 위해 사용될 수 있다.

[0278] 다른 예들에서, 인코더 (20) 는 위의 양태들을 나타내는 상이한 선택스 엘리먼트들; 예컨대, 어떤 성분(들)에 SEI가 적용되는 나타내는 하나의 선택스 엘리먼트, HDR-호환가능한 SDR-호환가능 콘텐츠에 적용되는지의 여부를 나타내는 하나의 선택스 엘리먼트, 및 성분 스케일링 SEI 메시지가 적용될 포지션(들)을 나타내는 하나의 선택스 엘리먼트를 시그널링 할 수도 있다.

[0279] 성분 스케일링 SEI 메시지 파라미터들이 적용되는 성분들의 수가 1을 초과할 때, 인코더 (20) 는 스케일 및 오프셋 파라미터들의 적용이 성분의 인덱스에 기초하여 순차적으로 행해질 수도 있음을 나타내는 선택스 엘리먼트를 포함하는 하나 이상의 SEI 메시지들을 시그널링 할 수도 있다. 예를 들어, 제 1 성분의 스케일 및 오프셋 파라미터들에 기초한 매핑은 적용될 수도 있고, 그 다음에 예를 들어 제 2 성분에 대해 시그널링되는 스케일 및 오프셋을 사용하는 제 2 성분의 매핑이 제 1 성분의 값들에 따라 달라질 수도 있다. 일부 예들에서, 이는, 예를 들어, 제 1 성분의 매핑된 값들이 사용되어야 함을 특정하는 선택스 엘리먼트에 의해 나타내어진다. 비디오 디코더 (30) 는 이러한 SEI 메시지들을 수신하며, 그 정보를 파싱 및/또는 디코딩하고, 그 정보를 비디오 포스트프로세서 (31) 에 전달할 수도 있다.

[0280] 다른 예에서, 비디오 인코더 (20)는 심지어 SEI 포스트프로세싱이 적용되지 않더라도 HDR10 수신기가 시청가능 HDR 비디오를 디코딩하고 보여줄 수 있는 그런 방식으로, 하나 이상의 SEI 메시지들에서, 또는 비트스트림에서 시그널링되는 값들을 제한할 수도 있다. SEI 메시지(들)은 이것이 그 경우 (예컨대, 비트스트림이 HDR10 하위 호환가능 비트스트림임)을 나타내는 신택스 엘리먼트를 포함할 수도 있다.

[0281] 이 섹션은 본 개시물의 하나 이상의 양태들에 따라서 개시되는 기법들을 사용하는 여러 예들을 포함한다.

[0282] 예 1

[0283] 이 예 1에서, 성분 스케일링 함수는 루업 테이블로서 시그널링되고 루업 테이블을 정의하는 포인트들을 시그널링하는데 사용되는 비트들의 수가 또한 시그널링된다. 시그널링되는 명시적인 포인트들을 가지지 않는 샘플 값들의 경우, 그 값은 이웃하는 괴벽 포인트들에 기초하여 보간된다.

[0284] 성분 스케일링 SEI 메시지의 신택스

component_scale_info(payloadSize) {	디스크립터
comp_scale_id	ue(v)
comp_scale_cancel_flag	u(1)
if(!comp_scale_cancel_flag) {	
comp_scale_persistence_flag	u(1)
comp_scale_num_comps_minus1	ue(v)
comp_scale_input_bit_depth	ue(v)
comp_scale_output_bit_depth	ue(v)
for(c = 0; c <= comp_scale_num_comps_minus1; c++) {	
comp_scale_num_points_minus1[c]	ue(v)
for(i = 0; i <= comp_scale_num_points_minus1[c]; i++) {	
comp_scale_input_point[c][i]	u(v)
comp_scale_output_point[c][i]	u(v)
}	
}	
}	

[0285]

[0286] 성분 스케일링 SEI 메시지의 시맨틱스

[0287] 성분 스케일링 SEI 메시지는 디코딩된 픽처들의 다양한 성분들에 대해 스케일링 동작들을 수행하기 위한 정보를 제공한다. 스케일링 동작들이 수행되어야 하는 컬러 공간 및 성분들은 SEI 메시지에서 시그널링되는 신택스 엘리먼트들의 값에 의해 결정된다.

[0288] **comp_scale_id**는 성분 스케일링 SEI 메시지의 목적을 식별하는데 사용될 수도 있는 식별 번호를 포함한다. **comp_scale_id**의 값은 0 내지 $2^{32} - 2$ 의 범위에 있을 것이다. **comp_scale_id**의 값은 성분 스케일링 SEI 메시지가 있는 컬러 공간, 또는 성분 스케일링 SEI 메시지가 선형 또는 비선형 도메인에서 적용되는지의 여부를 특정하는데 사용될 수도 있다.

[0289] 0 내지 255의 그리고 512 내지 $2^{31} - 1$ 의 **comp_scale_id**의 값들은 애플리케이션에 의해 결정된 대로 사용될 수도 있다. 256 내지 511의 그리고 2^{31} 내지 $2^{32} - 2$ 의 **comp_scale_id**의 값들은 ITU-T | ISO/IEC에 의한 장래의 사용을 위해 유보된다. 디코더들은 256 내지 511의 범위에서의, 또는 2^{31} 내지 $2^{32} - 2$ 의 범위에서의

`comp_scale_id`의 값을 포함하는 모든 성분 스케일 정보 SEI 메시지들을 무시할 것이고, 비트스트림들은 이러한 값을 포함하지 않을 것이다.

[0290] 주 1 - `comp_scale_id`는 상이한 디스플레이 시나리오들에 적합한 성분 스케일링 프로세스들을 지원하기 위해 사용될 수 있다. 예를 들어, `comp_scale_id`의 상이한 값들은 스케일링이 적용되는 상이한 디스플레이 비트 깊이들 또는 상이한 컬러 공간들에 대응할 수도 있다.

[0291] 대안적으로, `comp_scale_id`는 특정한 유형들의 디스플레이들 또는 디코더, 예컨대, HDR, SDR에 대한 호환성을 위해 스케일링이 수행되는지의 여부를 식별하는데 또한 사용될 수도 있다.

[0292] 1과 동일한 `comp_scale_cancel_flag`는 성분 스케일링 정보 SEI 메시지가 현재 계층에 적용되는 출력 순서에서 임의의 이전의 성분 정보 SEI 메시지들의 퍼시스턴스를 취소함을 나타낸다. 0과 동일한 `comp_scale_cancel_flag`는 성분 스케일링 정보가 뒤따를 것을 나타낸다.

[0293] `comp_scale_persistence_flag`는 현재 계층에 대한 성분 스케일링 정보 SEI 메시지의 퍼시스턴스를 특정한다.

[0294] 0과 동일한 `comp_scale_persistence_flag`는 성분 스케일링 정보가 현재 디코딩되는 픽처에만 적용됨을 특정한다.

[0295] `picA`를 현재 픽처이라고 하자. 1과 동일한 `comp_scale_persistence_flag`는 다음의 조건들 중 임의의 조건이 참이 되기까지 성분 스케일링 정보가 출력 순서에서의 현재 계층에 대해 지속함을 특정한다:

- 현재 계층의 새로운 CLVS가 시작한다.

- 비트스트림이 종료한다.

- `PicOrderCnt(picB)` 가 `PicOrderCnt(picA)` 보다 큰 경우, 동일한 값의 `comp_scale_id`를 갖고 현재 계층에 적용 가능한 성분 스케일링 정보 SEI 메시지를 포함하는 액세스 유닛의 현재 계층에서의 픽처 `picB`가 출력되며, 여기서 `PicOrderCnt(picB)` 와 `PicOrderCnt(picA)` 는 `picB`에 대한 픽처 순서 카운트를 위한 디코딩 프로세스의 호출 직후 각각 `picB` 및 `picA`의 `PicOrderCntVal` 값들이다.

[0299] `comp_scale_num_comps_minus1` 더하기 1은 성분 스케일링 함수가 특정되는 성분들의 수를 특정한다. `comp_scale_num_comps_minus1`은 0 내지 2의 범위에 있을 것이다.

[0300] `comp_scale_num_comps_minus1`이 2 미만이고 c -번째 성분의 성분 스케일링 파라미터들이 시그널링되지 않을 때, $(c - 1)$ -번째 성분의 그것들과 동일한 것으로 유추된다.

[0301] 대안적으로, `comp_scale_num_comps_minus1`이 2 미만이고, c -번째 성분의 성분 스케일링 파라미터들이 시그널링되지 않을 때, c -번째 성분의 성분 스케일링 파라미터들은 사실상 그 성분의 스케일링이 없도록 디폴트 값들과 동일한 것으로 유추된다.

[0302] 대안적으로, 성분 스케일링 파라미터들의 추론은 SEI 메시지가 적용되는 컬러 공간에 기초하여 특정될 수도 있다.

- 컬러 공간이 YCbCr이고, `comp_scale_num_comps_minus1`이 1과 동일할 때, 성분 스케일링 파라미터들은 Cb 및 Cr 성분들 둘 다에 적용된다.

- 컬러 공간이 YCbCr이고, `comp_scale_num_comps_minus1`이 2와 동일할 때, 제 1 및 제 2 성분 스케일링 파라미터들은 Cb 및 Cr 성분들에 적용된다.

[0305] 하나의 대체예에서, 상이한 추론은 `comp_scale_id`의 값에 기초하여 또는 명시적 신팩스 엘리먼트를 기초로 특정된다.

[0306] 대안적으로, 제약조건이 다음과 같이 추가된다:

[0307] `comp_scale_num_comps_minus1`의 값이 `comp_scale_id`의 주어진 값이 CLVS 내에 있는 모든 성분 스케일링 SEI 메시지에 대해 동일해야 한다는 것이 비트스트림 적합성에 대한 제약조건이다.

[0308] `comp_scale_input_bit_depth_minus8` 더하기 8은 신팩스 엘리먼트 `comp_scale_input_point[c][i]`를 시그널링하는데 사용되는 비트들의 수를 특정한다. `comp_scale_input_bit_depth_minus8`의 값은 0 내지 8의 범위에 있을 것이다.

- [0309] 성분 스케일링 SEI 메시지가 범위 0.0 내지 1.0의 정규화된 부동소수점 표현에 있는 입력에 적용될 때, SEI 메시지는 입력 비디오를 colour_remap_input_bit_depth_minus8 + 8과 동일한 비트 깊이를 갖는 변환된 비디오 표현으로 변환하기 위해 수행되는 양자화 동작의 가상적인 결과와 관련된다.
- [0310] 성분 스케일링 SEI 메시지가 comp_scale_input_bit_depth_minus8 + 8과 동일하지 않은 비트 깊이를 갖는 입력에 적용될 때, SEI 메시지는 입력 비디오 표현을 colour_remap_input_bit_depth_minus8 + 8과 동일한 비트 깊이를 갖는 변환된 비디오 표현으로 변환하기 위해 수행되는 트랜스코딩 동작의 가상적인 결과와 관련된다.
- [0311] comp_scale_output_bit_depth_minus8 더하기 8은 선택스 엘리먼트 comp_scale_output_point[c][i]를 시그널링하기 위해 사용되는 비트들의 수를 특정한다. comp_scale_output_bit_depth_minus8의 값은 0 내지 8의 범위에 있을 것이다.
- [0312] 성분 스케일링 SEI 메시지가 부동소수점 표현의 입력에 적용될 때, SEI 메시지는 성분 스케일링 SEI 메시지의 프로세싱 후에 획득되는 colour_remap_output_bit_depth_minus8 + 8과 동일한 비트 깊이를 갖는 비디오 표현을 범위 0.0 내지 1.0의 부동소수점 표현으로 변환하기 위해 수행되는 역 양자화 동작의 가상적인 결과와 관련된다.
- [0313] 대안적으로, comp_scale_input_point[c][i] 및 comp_scale_output_point[c][i]를 시그널링하는데 사용되는 비트들의 수들은 각각 comp_scale_input_bit_depth 및 comp_scale_output_bit_depth로서, 또는 다르게 말하면 8을 감산하는 일 없이 시그널링된다.
- [0314] comp_scale_num_points_minus1[c] 더하기 1은 성분 스케일링 함수를 정의하는데 사용되는 피벗 포인트들의 수를 특정한다. comp_scale_num_points_minus1[c]은 0 내지 $(1 \ll \text{Min}(\text{comp_scale_input_bit_depth_minus8} + 8, \text{comp_scale_output_bit_depth_minus8} + 8)) - 1$ 의 범위에 있을 것이다.
- [0315] comp_scale_input_point[c][i]은 입력 꽈치의 c-번째 성분의 i-번째 피벗 포인트를 특정한다. comp_scale_input_point[c][i]의 값은 0 내지 $(1 \ll \text{comp_scale_input_bit_depth_minus8}[c] + 8) - 1$ 의 범위에 있을 것이다. comp_scale_input_point[c][i]의 값은 1 내지 comp_scale_points_minus1[c]의 범위에서의 i에 대해, comp_scale_input_point[c][i - 1]의 값 이상에 있을 것이다.
- [0316] comp_scale_output_point[c][i]는 출력 꽈치의 c-번째 성분의 i-번째 피벗 포인트를 특정한다. comp_scale_output_point[c][i]의 값은 1 내지 $(1 \ll \text{comp_scale_output_bit_depth_minus8}[c] + 8) - 1$ 의 범위에 있을 것이다. 1 내지 comp_scale_points_minus1[c]의 범위에서의 i에 대해, comp_scale_output_point[c][i]의 값은 comp_scale_output_point[c][i - 1]의 값 이상일 것이다.
- [0317] 입력 신호 표현 x와 출력 신호 표현 y를 매핑하는 프로세스는, 입력 및 출력 둘 다에 대한 샘플 값들이 각각 0 내지 $(1 \ll \text{comp_scale_input_bit_depth_minus8}[c] + 8) - 1$ 의 범위와, 0 내지 $(1 \ll \text{comp_scale_output_bit_depth_minus8}[c] + 8) - 1$ 의 범위에 있을 때, 다음과 같이 특정된다:

```

if( x <= comp_scale_input_point[ c ][ 0 ] )
    y = comp_scale_output_point[ c ][ 0 ]
else if( x > comp_scale_input_point[ c ][ comp_scale_input_point_minus1[ c ] ] )
    y = comp_scale_output_point[ c ][ comp_scale_output_point_minus1[ c ] ]
else
    for( i = 1; i <= comp_scale_output_point_minus1[ c ]; i++ )
        if( comp_scale_input_point[ i - 1 ] < x && x <=
comp_scale_input_point[ i ] )

        y = ( ( comp_scale_output_point[ c ][ i ] - comp_scale_output_point[ c ][ i - 1 ] )
) +

```

(comp_scale_input_point[c][i] - comp_scale_input_point[c][i - 1]) *
(x - comp_scale_input_point[c][i - 1]) +
(comp_scale_output_point[c][i - 1])

[0318]

[0319] 하나의 대체예에서, 입력 및 출력 피벗 포인트들 comp_scale_input_point[c][i] 및 comp_scale_output_point[c][i]는 인접한 값들의 차이, 예컨대, delta_comp_scale_input_point[][] 및 delta_comp_scale_output_point[][]로서 코딩되고, 신택스 엘리먼트들은 지수 골룸 코드를 사용하여 코딩된다.

[0320]

다른 대체예에서, 입력 및 출력 표현 값을 매핑하는 프로세스는 스프라인들 및 큐빅 보간을 비제한적으로 포함하는 다른 보간 방법들에 의해 특정된다.

[0321]

예 2

[0322]

이 예 2는 예 1에서 설명된 SEI 신택스 구조와 비교되는 상이한 신택스 구조를 보여준다. 이 신택스 구조에서, 매핑 함수는 피벗 포인트들 대신 스케일들 및 오프셋들의 측면에서 설명된다.

[0323]

성분 스케일링 SEI 메시지의 속성

component_scale_info(payloadSize) {	디스크립터
comp_scale_id	ue(v)
comp_scale_cancel_flag	u(1)
if(!comp_scale_cancel_flag) {	
comp_scale_persistence_flag	u(1)
comp_scale_num_comps	ue(v)
comp_scale_input_bit_depth	ue(v)
comp_scale_output_bit_depth	ue(v)
comp_scale_bit_depth_scale_val	ue(v)
comp_scale_log2_denom_scale_val	ue(v)
for(c = 0; c < comp_scale_num_comps; c++) {	
comp_scale_num_points_minus1[c]	ue(v)
comp_scale_global_offset_input_val[c]	u(v)
comp_scale_global_offset_output_val[c]	u(v)
for(i = 0; i < comp_scale_num_points_minus1[c]; i++) {	
comp_scale_offset_val[c][i]	u(v)
comp_scale_val[c][i]	u(v)
}	
}	
}	
}	

[0324]

[0325]

comp_scale_bit_depth_scale_val은 속성 엘리먼트 **comp_scale_val[c][i]**를 시그널링하는데 사용되는 비트들의 수를 특정한다. **comp_scale_bit_depth_scale_val**의 값은 0 내지 24의 범위에 있을 것이다.

[0326]

comp_scale_log2_denom_scale_val은 스케일 값의 베이스 2 분모를 특정한다. **comp_scale_log2_denom_scale_val**의 값은 0 내지 16의 범위에 있을 것이다.

[0327]

comp_scale_global_offset_input_val[c] 더하기 1은 모든 입력 표현 값들이 CompScaleOffsetOutputVal[c][0]으로 클리핑되게 하는 입력 샘플 값을 특정한다. 성분 스케일링 함수를 정의하는데 사용됨. **comp_scale_num_points_minus1[c]**는 0 내지 $(1 \ll \text{comp_scale_input_bit_depth}) - 1$ 의 범위에 있을 것이다. **comp_scale_global_offset_input_val[c]**를 표현하는데 사용되는 비트들의 수는 **comp_scale_input_bit_depth**이다.

[0328]

comp_scale_global_offset_output_val[c] 더하기 1은 **comp_scale_global_offset_input_val[c]** 미만의 모든 입력 표현 값들이 클리핑될 출력 샘플 값을 특정한다. **comp_scale_num_points_minus1[c]**는 0 내지 $(1 \ll \text{comp_scale_output_bit_depth}) - 1$ 의 범위에 있을 것이다. **comp_scale_global_offset_output_val[c]**를 표현하는데 사용되는 비트들의 수는 **comp_scale_output_bit_depth**이다.

[0329]

comp_scale_num_points_minus1[c] 더하기 1은 성분 스케일링 함수를 정의하는데 사용되는 피벗 포인트들의 수를 특정한다. **comp_scale_num_points_minus1[c]**는 0 내지 $(1 \ll \text{Min}(\text{comp_scale_input_bit_depth}, \text{comp_scale_output_bit_depth})) - 1$ 의 범위에 있을 것이다.

[0330]

입력 신호 표현 x와 출력 신호 표현 y를 매핑하는 프로세스는, 양 입력 표현에 대한 샘플 값들이 0 내지 $(1 \ll \text{comp_scale_input_bit_depth}) - 1$ 의 범위에, 그리고 출력 표현이 0 내지 $(1 \ll \text{comp_scale_output_bit_depth})$

- 1의 범위에 있을 때, 다음과 같이 특정된다:

```

if( x <= CompScaleOffsetInputVal[ c ][ 0 ] )
    y = CompScaleOffsetOutputVal[ c ][ 0 ]
else if( x > CompScaleOffsetInputVal[ c ][ comp_scale_output_point_minus1 ] )
    y = CompScaleOffsetOutputVal[ c ][ comp_scale_output_point_minus1 ]
else
    for( i = 1; i <= comp_scale_output_point_minus1; i++ )
        if( CompScaleOffsetInputVal[ i - 1 ] < x && x <=
CompScaleOffsetInputVal[ i ] )
            y = ( x -
CompScaleOffsetInputVal[ i - 1 ] * ( comp_scale_val[ c ][ i ] +
CompScaleOffsetOutputVal[ c ][ i ] )

```

[0331]

[0332] **comp_scale_offset_val[c][i]**는 c-번째 성분의 i-번째 샘플 값 지역의 오프셋 값을 특정한다. **comp_scale_offset_val[c]**를 표현하는데 사용되는 비트들의 수는 **comp_scale_input_bit_depth**와 동일하다.

[0333]

[0333] **comp_scale_val[c][i]**는 c-번째 성분의 i-번째 샘플 값 지역 포인트의 스케일 값을 특정한다. **comp_scale_val[c]**를 표현하는데 사용되는 비트들의 수는 **comp_scale_bit_depth_scale_val**과 동일하다.

[0334]

0 내지 **comp_scale_num_points_minus1[c]**의 범위에서의 i에 대한 변수 **CompScaleOffsetOutputVal[c][i]** 및 **CompScaleOffsetInputVal[c][i]**는 다음과 같이 도출된다:

```

roundingOffset = (comp_scale_log2_denom_scale_val == 0 ) ? 0 : (1 <<
                comp_scale_log2_denom_scale_val - 1)
for( i = 0; i <= comp_scale_num_points_minus1[ c ]; i++ )
    if( i == 0 )
        CompScaleOffsetOutputVal[ c ][ i ] =
comp_scale_global_offset_output_val[ c ]
        CompScaleOffsetInputVal[ c ][ i ] =
comp_scale_global_offset_input_val[ c ]
    else
        CompScaleOffsetOutputVal[ c ][ i ] = CompScaleOffsetOutputVal[ c ][ i -
1 ] +
                    (comp_scale_offset_val[ c ][ i - 1 ] - comp_scale_val[ c ][ i -
1 ] +
                     roundingOffset ) >>
comp_scale_log2_denom_scale_val
        CompScaleOffsetInputVal[ c ][ i ] = CompScaleOffsetInputVal[ c ][ i - 1 ] +
                    comp_scale_offset_val[ c ][ i - 1 ]

```

[0335]

[0336] 하나의 대체예에서, **comp_scale_offset_val[c][i]**는 0 내지 **comp_scale_num_points_minus1[c]**의 범위에서의 i에 대해 **CompScaleOffsetOutputVal[]][i]**를 직접적으로 계산하고 **CompScaleOffsetInputVal[]][i]**를 간접적으로 계산하는데 다음과 같이 사용된다:

```

for( i = 0; i < comp_scale_num_points_minus1[ c ]; i++ )
    if( i == 0 )
        CompScaleOffsetOutputVal[ c ][ i ] =
comp_scale_global_offset_output_val[ c ]
        CompScaleOffsetInputVal[ c ][ i ] =
comp_scale_global_offset_input_val[ c ]
    else
        CompScaleOffsetInputVal[ c ][ i ] = CompScaleOffsetInputVal[ c ][ i - 1 ]

```

[0337]

```

+
(comp_scale_offset_val[ c ][ i - 1 ] *
comp_scale_val[ c ][ i - 1 ]
+ roundingOffset ) >>
comp_scale_log2_denom_scale_val )
CompScaleOffsetOutputVal[ c ][ i ] = CompScaleOffsetOutputVal[ c ][ i -
1 ] +
comp_scale_offset_val[ c ][ i - 1 ]

```

[0338]

[0339] 하나의 대체예에서, 0 내지 comp_scale_num_points_minus1[c]의 범위에서의 i에 대한 comp_scale_offset_val[c][i]는 시그널링되지 않고, comp_scale_offset_val[c][i]의 값들은 스케일이 특정되는 comp_scale_num_points_minus1[c] 동일하게 이격된 간격들에 기초하여 도출된다. 0 내지 comp_scale_num_points_minus1[c] - 1의 범위에서의 i에 대한 comp_scale_offset_val[c][i]의 값은 다음과 같이 도출된다:

[0340]

```
comp_scale_offset_val[ c ][ i ] = ((1 << comp_scale_output_bit_depth)
```

[0341]

```
- comp_scale_global_offset_output_val[ c ])
```

[0342]

```
÷ (comp_scale_num_points_minus1[ c ])
```

[0343]

[0343] 다른 대체예에서, 0 내지 comp_scale_num_points_minus1[c]의 범위에서의 i에 대한 comp_scale_offset_val[c][i]는 다음과 같이 계산된다:

[0344]

```
comp_scale_offset_val[ c ][ i ] = (1 << comp_scale_output_bit_depth ) ÷ (comp_scale_num_points_minus1[ c ])
```

[0345]

[0345] 하나의 대체예에서, comp_scale_num_points_minus1[c]를 시그널링하는 대신, 피벗 포인트들의 수는 log2_comp_scale_num_points[c]를 사용하여 시그널링되는데, 여기서 (1 << log2_comp_scale_num_points[c]) 는 c-번째 성분에 대한 피벗 포인트들의 수를 특정한다.

[0346]

[0346] 대안적으로, comp_scale_offset_val[c][] 및 comp_scale_val[c][]의 각각은 부동소수점 수들로서, 또는 지수 및 가수를 갖는 두 개의 십진수 엘리먼트들로서 시그널링된다.

[0347]

[0347] 다른 대체예에서, comp_scale_val[c][i]의 시그널링은 comp_scale_output_point[c][i]에 의해 대체된다.

[0348]

[0348] 십진수 엘리먼트들의 나머지의 시멘틱스는 예 1에서 설명된 것들과 유사하다.

예 3

[0350]

[0350] 예 3에서 설명되는 이 방법은, 성분 스케일링 함수들이 독립적으로 업데이트되는 것이 허용된다는 것을 제외하면, 예 2에서 설명되는 대체예들 중 하나의 대체예와 유사하다.

[0351]

성분 스케일링 SEI 메시지의 신택스

component_scale_info(payloadSize) {	디스크립터
comp_scale_id	ue(v)
comp_scale_cancel_flag	u(1)
if(!comp_scale_cancel_flag) {	
comp_scale_persistence_flag	u(1)
comp_scale_num_comps	ue(v)
comp_scale_input_bit_depth	ue(v)
comp_scale_output_bit_depth	ue(v)
for(c = 0; c < comp_scale_num_comps; c++) {	
comp_scale_persist_component_flag[c]	u(1)
if(!comp_scale_persist_component_flag[c])	
comp_scale_num_scale_regions[c]	ue(v)
comp_scale_global_offset_input_val[c]	u(v)
comp_scale_global_offset_output_val[c]	u(v)
for(i = 0; i < comp_scale_num_scale_regions[c]; i++) {	
comp_scale_offset_val[c][i]	u(v)
comp_scale_val[c][i]	u(v)
}	
}	
}	
}	
}	

[0352]

성분 스케일링 SEI 메시지의 시맨틱스

[0353] 시맨틱스는, 다음의 신택스 엘리먼트들을 제외하면, 예 2와 유사하다.

[0354] **comp_scale_num_scale_regions[c]**는 신택스 엘리먼트 **comp_scale_val[c][i]**가 c-번째 성분에 대해 시그널링되는 지역들의 수를 특정한다. **comp_scale_num_scale_regions[c]**는 0 내지 (**1 << comp_scale_input_bit_depth**) - 1의 범위에 있을 것이다.

[0355] 0과 동일한 **comp_scale_persist_component_flag[c]**는 c-번째 성분에 대한 성분 스케일링 파라미터들이 SEI 메시지에서 명시적으로 시그널링됨을 특정한다. 1과 동일한 **comp_scale_persist_component_flag[c]**는 c-번째 성분에 대한 성분 스케일링 파라미터들이 SEI 메시지에서 명시적으로 시그널링되지 않음을 특정하고, 그것은 출력 순서에서, 이전의 픽처에 적용되는 성분 스케일링 SEI 메시지의 c-번째 성분의 성분 스케일링 파라미터들에서부터 지속된다.

[0356] 성분 스케일링 SEI 메시지가 IRAP 액세스 유닛에 존재할 때, **comp_scale_persist_component_flag[c]**의 값은, 존재하는 경우, 0과 동일할 것이라는 것이 비트스트림 적합성의 요건이다.

[0357] 다르게는, 다음의 조건이 추가된다:

[0358] 성분 스케일링 SEI 메시지가 IRAP 액세스 유닛이 아닌 액세스 유닛에 존재하고 **comp_scale_persist_component_flag[c]**가 1과 동일할 때,

[0359] 출력 순서에서 현재 픽처에 선행하고, 출력 순서에서, 디코딩 순서에서의 이전의 IRAP 픽처에 후행하는 적어도 하나의 픽처가 있어서, 1과 동일한 **comp_scale_persistence_flag**를 갖는 성분 스케일링 SEI 메시지에 하나의 픽

처가 연관된다는 것이 비트스트림 적합성의 요건이다.

[0361] **comp_scale_persistence_flag**는 현재 계층에 대한 성분 스케일링 정보 SEI 메시지의 퍼시스턴스를 특정한다.

[0362] 0과 동일한 **comp_scale_persistence_flag**는 성분 스케일링 정보가 현재 디코딩되는 블록에만 적용됨을 특정한다.

[0363] **picA**가 현재 블록이라고 하자. 1과 동일한 **comp_scale_persistence_flag**는 다음의 조건들 중 임의의 조건이 참이 되기까지 c-번째 성분의 성분 스케일링 정보가 출력 순서에서의 현재 계층에 대해 지속함을 특정한다:

- 현재 계층의 새로운 CLVS가 시작한다.

- 비트스트림이 종료한다.

[0366] - **PicOrderCnt(picB)** 가 **PicOrderCnt(picA)** 보다 큰 경우, 동일한 값의 **comp_scale_id** 및 0과 동일한 **comp_scale_persist_component_flag[c]**를 갖고 현재 계층에 적용 가능한 성분 스케일링 정보 SEI 메시지를 포함하는 액세스 유닛의 현재 계층에서의 블록 **picB**가 출력되며, 여기서 **PicOrderCnt(picB)** 와 **PicOrderCnt(picA)** 는 **picB**에 대한 블록 순서 카운트를 위한 디코딩 프로세스의 호출 직후 각각 **picB** 및 **picA**의 **PicOrderCntVal** 값들이다.

예 4

[0368] 이 예 4에서, 스케일 지역들을 시그널링하는 상이한 방법이 개시된다.

성분 스케일링 SEI 메시지 선택스에 대한 변경들

component_scale_info(payloadSize) {	디스크립터
comp_scale_id	ue(v)
comp_scale_cancel_flag	u(1)
if(!comp_scale_cancel_flag) {	
comp_scale_persistence_flag	u(1)
comp_scale_num_comps	ue(v)
comp_scale_input_bit_depth	ue(v)
comp_scale_output_bit_depth	ue(v)
for(c = 0; c < comp_scale_num_comps; c++) {	
comp_scale_persist_component_flag[c]	u(1)
if(!comp_scale_persist_component_flag[c])	
comp_scale_global_offset_input_val[c]	u(v)
comp_scale_global_offset_output_val[c]	u(v)
comp_scale_num_scale_regions[c]	ue(v)
for(i = 0; i < comp_scale_num_scale_regions[c]; i++) {	
comp_scale_offset_begin_val[c][i]	u(v)
comp_scale_offset_end_val[c][i]	u(v)
comp_scale_val[c][i]	u(v)
}	
}	
}	
}	

성분 스케일링 SEI 메시지 시맨틱스에 대한 변경들

[0372] 선택스 엘리먼트들의 시맨틱스는, 다음을 제외하면, 이전의 예들에서의 것들과 유사하다:

- [0373] **comp_scale_offset_begin_val[c][i]**는 스케일 값 **comp_scale_val[c][i]**가 적용 가능한 샘플 값 범위의 시작부분을 특정한다. **comp_scale_offset_begin_val[c]**를 표현하는데 사용되는 비트들의 수는 **comp_scale_input_bit_depth**와 동일하다.
- [0374] **comp_scale_offset_end_val[c][i]**는 스케일 값 **comp_scale_val[c][i]**가 적용 가능한 샘플 값 범위의 말단을 특정한다. **comp_scale_offset_end_val[c]**를 표현하는데 사용되는 비트들의 수는 **comp_scale_input_bit_depth**와 동일하다.
- [0375] **comp_scale_offset_begin_val** 및 **comp_scale_offset_end_val**에 의해 명시적으로 특정되지 않는 지역들의 경우, 그들 지역들에 대한 **comp_scale_value[c][i]**는 0과 동일한 것으로 유추된다.
- [0376] 대안적으로, **comp_scale_offset_end_val[c][i]**는 시그널링되지 않고 대신 **comp_scale_offset_end_val[c][i]**와 **comp_scale_offset_begin_val[c][i]** 사이의 차이는 시그널링되고, **comp_scale_offset_end_val[c][i]**의 값은 디코더 측에서 도출된다.
- [0377] 다른 대체예에서, 출력 샘플 범위가 분할되는 지역들의 총 수는 특정되고, 스케일 지역들이 명시적으로 시그널링되는 지역들의 수는 시그널링된다.

...	u(v)
comp_scale_global_offset_output_val[c]	u(v)
comp_scale_tot_scale_regions[c]	ue(v)
comp_scale_num_scale_regions[c]	ue(v)
for(i = 0; i < comp_scale_num_scale_regions[c] ; i++) {	
comp_scale_region_idx[c][i]	u(v)
comp_scale_val[c][i]	u(v)
}	
...	

- [0378] **comp_scale_tot_scale_regions[c]**는 샘플 값들이 분할되는 동일한 길이 샘플 값 범위들의 총 수를 특정한다. **comp_scale_tot_scale_regions[c]**를 표현하는데 사용되는 비트들의 수는 **comp_scale_input_bit_depth**와 동일하다.
- [0380] 하나의 대체예에서, **comp_scale_tot_scale_regions[c]** 샘플 값 범위들은 길이가 정확히 동일하지 않을 수도 있지만 지역 길이들의 정수 정확도를 고려하면 정말 거의 동일하다.
- [0381] **comp_scale_region_idx[c][i]**는 스케일 값 **comp_scale_val[c][i]**가 적용되는 샘플 값 범위의 인덱스를 특정한다. 선택스 엘리먼트 **comp_scale_region_idx[c]**의 길이는 Ceil(Log2(**comp_scale_tot_scale_regions[c]**)) 비트이다.
- [0382] **대체예들**
- [0383] 대안적으로, 크로마 중간색 (10-비트 데이터에 대해 511) 주위의 지역은 더 작은 사이즈, 즉, 다른 지역들의 절반 사이즈를 가진다.
- [0384] 예 5

[0385]

성분 스케일 SEI 메시지의 십택스

component_scale_info(payloadSize) {	디스크립터
comp_scale_id	ue(v)
comp_scale_cancel_flag	u(1)
if(!comp_scale_cancel_flag) {	
comp_scale_persistence_flag	u(1)
comp_scale_scale_bit_depth	u(4)
comp_scale_offset_bit_depth	u(4)
comp_scale_scale_frac_bit_depth	u(4)
comp_scale_offset_frac_bit_depth	u(4)
comp_scale_num_comps_minus1	ue(v)
for(c = 0; c <= comp_scale_num_comps_minus1; c++) {	
comp_scale_num_ranges[c]	ue(v)
comp_scale_equal_ranges_flag[c]	u(1)
comp_scale_global_offset_val[c]	u(v)
for(i = 0; i <= comp_scale_num_ranges[c]; i++)	
comp_scale_scale_val[c][i]	u(v)
if(!comp_scale_equal_ranges[c])	u(v)
for(i = 0; i <= comp_scale_num_ranges[c]; i++)	
comp_scale_offset_val[c][i]	u(v)
}	
}	

[0386]

성분 스케일 SEI 메시지의 시맨틱스

[0388]

성분 스케일링 SEI 메시지는 디코딩된 픽처들의 다양한 성분들에 대해 스케일링 동작들을 수행하기 위한 정보를 제공한다. 스케일링 동작들이 수행되어야 하는 컬러 공간 및 성분들은 SEI 메시지에서 시그널링되는 십택스 엘리먼트들의 값에 의해 결정된다.

[0389]

comp_scale_id는 성분 스케일링 SEI 메시지의 목적을 식별하는데 사용될 수도 있는 식별 번호를 포함한다. **comp_scale_id**의 값은 0 내지 $2^{32} - 2$ 의 범위에 있을 것이다. **comp_scale_id**의 값은 성분 스케일링 SEI 메시지가 있는 컬러 공간, 또는 성분 스케일링 SEI 메시지가 선형 또는 비선형 도메인에서 적용되는지의 여부를 특정하는데 사용될 수도 있다.

[0390]

일부 예들에서, **comp_scale_id**는 HDR 복원 프로세스의 구성을 특정할 수 있다. 일부 예들에서, **comp_scale_id**의 특정 값은 3 개의 성분들에 대한 스케일링 파라미터들의 시그널링과 연관될 수도 있다. R', G', B' 컬러 공간의 샘플들에 적용될 제 1 성분들의 스케일링과, 뒤따르는 2 개의 성분들의 파라미터들은 Cr 및 Cb의 스케일링에 적용된다.

[0391]

또 다른 **comp_scale_id** 값의 경우, hdr 복원 프로세스가 3 개의 성분들에 대한 파라미터들을 이용할 수 있고, 스케일링은 루마, Cr 및 Cb 컬러 성분들의 샘플들에 적용된다.

[0392]

또 다른 **comp_scale_id** 값에서, hdr 복원 프로세스는 4 개의 성분들에 대한 시그널링을 이용할 수 있으며, 그것들 중 3은 루마, Cr 및 Cb 스케일링에 적용되고, 4번째 성분은 컬러 정정의 파라미터들을 제공한다.

[0393]

일부 예들에서, **comp_scale_id** 값들의 특정한 범위는 SDR 하위 호환가능 구성으로 수행되는 HDR 복원과 연관될 수도 있는 한편, **comp_scale_id** 값들의 다른 범위는 비-하위 호환가능 구성으로 수행되는 HDR 복원과 연관될 수도 있다.

- [0394] 0 내지 255의 그리고 512 내지 $2^{31} - 1$ 의 comp_scale_id의 값들은 애플리케이션에 의해 결정된 대로 사용될 수도 있다. 256 내지 511의 그리고 2^{31} 내지 $2^{32} - 2$ 의 comp_scale_id의 값들은 ITU-T | ISO/IEC에 의한 장래의 사용을 위해 유보된다. 디코더들은 256 내지 511의 범위에서의, 또는 2^{31} 내지 $2^{32} - 2$ 의 범위에서의 comp_scale_id의 값을 포함하는 모든 성분 스케일 정보 SEI 메시지를 무시할 것이고, 비트스트림들은 이러한 값을 포함하지 않을 것이다.
- [0395] 주 1 - comp_scale_id는 상이한 디스플레이 시나리오들에 적합한 성분 스케일링 프로세스들을 지원하기 위해 사용될 수 있다. 예를 들어, comp_scale_id의 상이한 값들은 스케일링이 적용되는 상이한 디스플레이 비트 깊이를 또는 상이한 컬러 공간들에 대응할 수도 있다.
- [0396] 대안적으로, comp_scale_id는 특정한 유형들의 디스플레이들 또는 디코더, 예컨대, HDR, SDR에 대한 호환성을 위해 스케일링이 수행되는지의 여부를 식별하는데 또한 사용될 수도 있다.
- [0397] 1과 동일한 comp_scale_cancel_flag는 성분 스케일링 정보 SEI 메시지가 현재 계층에 적용되는 출력 순서에서 임의의 이전의 성분 정보 SEI 메시지들의 퍼시스턴스를 취소함을 나타낸다. 0과 동일한 comp_scale_cancel_flag는 성분 스케일링 정보가 뒤따름을 나타낸다.
- [0398] comp_scale_persistence_flag는 현재 계층에 대한 성분 스케일링 정보 SEI 메시지의 퍼시스턴스를 특정한다.
- [0399] 0과 동일한 comp_scale_persistence_flag는 성분 스케일링 정보가 현재 디코딩되는 픽처에만 적용됨을 특정한다.
- [0400] picA를 현재 픽쳐이라고 하자. 1과 동일한 comp_scale_persistence_flag는 다음의 조건들 중 임의의 조건이 참이 되기까지 성분 스케일링 정보가 출력 순서에서의 현재 계층에 대해 지속함을 특정한다:
- 현재 계층의 새로운 CLVS가 시작한다.
 - 비트스트림이 종료한다.
 - PicOrderCnt(picB) 가 PicOrderCnt(picA) 보다 큰 경우, 동일한 값의 comp_scale_id를 갖고 현재 계층에 적용 가능한 성분 스케일링 정보 SEI 메시지를 포함하는 액세스 유닛의 현재 계층에서의 픽처 picB가 출력되며, 여기서 PicOrderCnt(picB) 와 PicOrderCnt(picA) 는 picB에 대한 픽처 순서 카운트를 위한 디코딩 프로세스의 호출 직후 각각 picB 및 picA의 PicOrderCntVal 값들이다.
- [0401] comp_scale_scale_bit_depth는 선택스 엘리먼트 comp_scale_scale_val[c][i]를 시그널링하는데 사용되는 비트들의 수를 특정한다. comp_scale_scale_bit_depth의 값은 0 내지 15의 범위에 있을 것이다.
- [0402] comp_scale_offset_bit_depth는 선택스 엘리먼트 comp_scale_global_offset_val[c] 및 comp_scale_offset_val[c][i]를 시그널링하는데 사용되는 비트들의 수를 특정한다. comp_scale_offset_bit_depth의 값은 0 내지 15의 범위에 있을 것이다.
- [0403] comp_scale_scale_frac_bit_depth는 c-번째 성분의 i-번째 파티션의 스케일 파라미터의 소수 부분을 나타내는데 사용되는 LSB들의 수를 특정한다. comp_scale_scale_frac_bit_depth의 값은 0 내지 15의 범위에 있을 것이다. comp_scale_offset_bit_depth의 값은 comp_scale_scale_bit_depth의 값 이하일 것이다.
- [0404] comp_scale_offset_frac_bit_depth는 c-번째 성분의 i-번째 파티션의 오프셋 파라미터의 소수 부분과 c-번째 성분의 클로벌 오프셋을 나타내는데 사용되는 LSB들의 수를 특정한다. comp_scale_offset_frac_bit_depth의 값은 0 내지 15의 범위에 있을 것이다. comp_scale_offset_bit_depth의 값은 comp_scale_offset_bit_depth의 값 이하일 것이다.
- [0405] comp_scale_num_comps_minus1 더하기 1은 성분 스케일링 함수가 특정되는 성분들의 수를 특정한다. comp_scale_num_comps_minus1은 0 내지 2의 범위에 있을 것이다.
- [0406] comp_scale_num_ranges[c]는 출력 샘플 범위가 파티셔닝되는 범위들의 수를 특정한다. comp_scale_num_ranges[c]의 값은 0 내지 63의 범위에 있을 것이다.
- [0407] 1과 동일한 comp_scale_equal_ranges_flag[c]는 그 출력 샘플 범위가 comp_scale_num_ranges[c] 거의 동일한 파티션들로 파티셔닝되고, 파티션 폭들이 명시적으로 시그널링되지 않음을 나타낸다. 0과 동일한 comp_scale_equal_ranges_flag[c]는 그 출력 샘플 범위가 모두가 동일한 사이즈로 되는 것은 아닌

`comp_scale_num_ranges[c]` 파티션들로 파티셔닝될 수도 있고, 파티션들 폭들이 명시적으로 시그널링됨을 나타낸다.

[0411] `comp_scale_global_offset_val[c]`는 c -번째 성분에 대해 유효한 입력 데이터 범위의 최소 값을 매핑하는데 사용되는 오프셋 값을 도출하는데 사용된다. `comp_scale_global_offset_val[c]`의 길이는 `comp_scale_offset_bit_depth` 비트이다.

[0412] `comp_scale_scale_val[c][i]`는 c -번째 성분의 i -번째 파티션의 폭을 도출하는데 사용되는 오프셋 값을 도출하는데 사용된다. `comp_scale_global_offset_val[c]`의 길이는 `comp_scale_offset_bit_depth` 비트이다.

[0413] 변수 `CompScaleScaleVal[c][i]`는 다음과 같이 도출된다:

```
CompScaleScaleVal[ c ][ i ] = ( comp_scale_scale_val[ c ][ i ] >>
                                comp_scale_scale_frac_bit_depth ) +
                                ( comp_scale_scale_val[ c ][ i ] &
                                  ( 1 << comp_scale_scale_frac_bit_depth ) -
                                  1 )
                                ( 1 << comp_scale_scale_frac_bit_depth )
```

[0414] `comp_scale_offset_val[c][i]`는 c -번째 성분의 i -번째 파티션의 폭을 도출하는데 사용되는 오프셋 값을 도출하는데 사용된다. `comp_scale_global_offset_val[c]`의 길이는 `comp_scale_offset_bit_depth` 비트이다.

[0415] `comp_scale_offset_val[c][i]`가 시그널링될 때, `CompScaleOffsetVal[c][i]`의 값은 다음과 같이 도출된다:

```
CompScaleOffsetVal[ c ][ i ] = ( comp_scale_offset_val[ c ][ i ] >>
                                 comp_scale_offset_frac_bit_depth ) +
                                 ( comp_scale_offset_val[ c ][ i ] &
                                   ( 1 << comp_scale_offset_frac_bit_depth ) -
                                   1 )
```

[0416] 대안적으로, 변수 `CompScaleScaleVal[c][i]` 및 `CompScaleOffsetVal[c][i]`는 다음과 같이 도출된다:

```
CompScaleScaleVal[ c ][ i ] = comp_scale_scale_val[ c ][ i ] &
                               ( 1 << comp_scale_scale_frac_bit_depth )
CompScaleOffsetVal[ c ][ i ] = comp_scale_offset_val[ c ][ i ] ÷
                               ( 1 << comp_scale_offset_frac_bit_depth )
```

[0417] `comp_scale_equal_ranges_flag[c]`가 1과 동일할 때, `comp_scale_offset_val[c][i]`는 시그널링되지 않고, `CompScaleOffsetVal[c][i]`의 값은 다음과 같이 도출된다:

[0418] `CompScaleOffsetVal[c][i] = 1 ÷ comp_scale_num_ranges[c]`

[0419] 0 내지 `comp_scale_num_ranges[c]`의 범위에서의 i 에 대한 변수 `CompScaleOutputRanges[c][i]` 및 `CompScaleOutputRanges[c][i]`는 다음과 같이 도출된다:

```

for( i = 0; i <= comp_scale_num_ranges[ c ]; i++ )
    if( i == 0 )
        CompScaleOutputRanges[ c ][ i ] = comp_scale_global_offset_val[ c ]÷
            (1 << comp_scale_offset_frac_bit_depth )
        CompScaleInputRanges[ c ][ i ] = 0
    else
        CompScaleInputRanges[ c ][ i ] = CompScaleOffsetInputRanges[ c ][ i -
1 ] +
            (CompScaleOffsetVal[ c ][ i - 1 ] *
CompScaleScaleVal[ c ][ i - 1 ]
        CompScaleOutputRanges[ c ][ i ] = CompScaleOutputRanges[ c ][ i - 1 ] +
            CompScaleOffsetVal[ c ][ i - 1 ]

```

[0424]

[0425] 하나의 대체예에서, CompScaleOutputRanges[][] 및 CompScaleOutputRanges[][]의 값들은 다음과 같아도 출된다:

```

for( i = 0; i <= comp_scale_num_ranges[ c ]; i++ )
    if( i == 0 )
        CompScaleInputRanges[ c ][ i ] = comp_scale_global_offset_val[ c ]÷
            (1 << comp_scale_offset_frac_bit_depth )
        CompScaleOutputRanges[ c ][ i ] = 0
    else
        CompScaleInputRanges[ c ][ i ] = CompScaleOffsetInputRanges[ c ][ i -
1 ] +
            (CompScaleOffsetVal[ c ][ i - 1 ] *
CompScaleScaleVal[ c ][ i - 1 ]
        CompScaleOutputRanges[ c ][ i ] = CompScaleOutputRanges[ c ][ i - 1 ] +
            CompScaleOffsetVal[ c ][ i - 1 ]

```

[0426]

[0427]

[0428] 입력 신호 표현 (이)는 정수 뿐만 아니라 부동소수점 둘 다를 커버하는데 사용될 수도 있음) x와 출력 신호 표현 y를 매핑하는 프로세스는, 양 입력 표현에 대한 샘플 값들이 0 내지 1의 범위로 정규화되고 출력 표현이 0 내지 1의 범위에 있는 경우, 다음과 같이 특정된다:

```

if( x <= CompScaleInputRanges[ c ][ 0 ] )
    y = CompScaleOutputRanges[ c ][ 0 ]
else if( x > CompScaleInputRanges[ c ][ comp_scale_num_ranges[ c ] ] )
    y = CompScaleOutputRanges[ c ][ comp_scale_num_ranges[ c ]; ]
else
    for( i = 1; i <= comp_scale_num_ranges[ c ]; i++ )
        if( CompScaleInputRanges[ i - 1 ] < x && x <=
CompScaleInputRanges[ i ] )
            y = ( x -
CompScaleInputRanges[ i - 1 ] ) / comp_scale_val[ c ][ i ] +
            CompScaleOutputRanges[ c ][ i - 1 ]

```

[0429]

[0430] 하나의 대체예에서, CompScaleOutputRanges[c][0]의 값은 허용된 샘플 값 범위에 기초하여 설정된다.

[0431] 대안적으로, 입력 값 valIn을 출력 값 valOut으로 매핑하는 프로세스는 다음과 같이 정의된다:

```
m_pAtfRangeIn[ 0 ] = 0;
m_pAtfRangeOut[ 0 ] = -m_offset2 *m_pAtfScale2[c][0];
for (int j = 1; j < m_atfNumberRanges + 1; j++)
{
    m_pAtfRangeIn[ j ] = m_pAtfRangeIn[ j - 1 ] + m_pAtfDelta[ j - 1 ];
    m_pAtfRangeOut[ j ] = m_pAtfRangeOut[ j - 1 ] + m_pAtfScale2[ c ][ j -
1 ] *
    m_pAtfDelta[ j - 1 ];
}
```

[0432]

```
for (int j = 0; j < numRanges && skip == 0; j++)
{
    if (valIn <= pAtfRangeIn[ j + 1 ])
    {
        valOut = (valIn - pOffset[component][ j ]) *
pScale[ component ][ j ];
        skip = 1;
    }
}
```

[0433]

[0434] 하나의 대체예에서, c-번째 성분에 대해 m_offset2는 comp_scale_global_offset_val[c] ($1 \ll comp_scale_offset_frac_bit_depth$) 와 동일하며, m_pAtfScale[c][i]는 CompScaleScaleVal[c][i]와 동일하고 m_pAtfDelta[i]는 CompScaleOffsetVal[c][i]와 동일하고, pScale 및 pOffset는 m_pAtfScale 및 m_pAtfDelta로부터 도출되는 스케일 및 오프셋 파라미터이다.

[0435] 역 동작이 그에 따라 정의될 것이다.

예 6

[0437] 일부 예들에서, 위에서, 예컨대, 예 5에서 설명된 시그널링 방법들 중 일부는, 다음의 의사 코드에서 보인 바와 같이 이용될 수 있다.

[0438] m_atfNumberRanges는 매핑된 데이터에 대한 동적 범위 파티셔닝의 수를 특정하는 주어진 c에 대한 선택스 엘리먼트들 comp_scale_num_ranges[c]를 위한 항이다.

[0439] m_pAtfRangeIn은 CompScaleInputRanges를 위한 항이며, 두 개의 연접된 파티션들, 예컨대, i 및 i+1 사이의 경계를 특정하는 입력 샘플 값을 포함하는 m_atfNumberRanges+1의 어레이 사이즈이다.

[0440] m_pAtfRangeOut은 CompScaleOutputRanges를 위한 항이며, 두 개의 연접된 파티션들, 예컨대, i 및 i+1 사이의 경계를 특정하는 출력 샘플 값을 포함하는 m_atfNumberRanges+1의 어레이 사이즈이다.

[0441] m_pAtfScale2는 변수 CompScaleScaleVal[c]를 위한 항이며 각각의 파티션들에 대한 스케일 값을 포함하는 m_atfNumberRanges의 어레이 사이즈이다.

[0442] m_pAtfOffset2는 각각의 파티션에 대한 오프셋 값을 포함하는 m_atfNumberRanges의 어레이 사이즈이다.

[0443] m_offset2는 comp_scale_global_offset_val을 위한 항이다.

[0444] 이 예에서, 조각적 선형 모델의 파라미터들은 알고리즘 1에서처럼 선택스 엘리먼트들로부터 결정될 수 있다:

[0445] 알고리즘 1:

```
m_pAtfRangeIn[0] = 0;
m_pAtfRangeOut[0] = -m_offset2 * m_pAtfScale2[c][0];
for (int j = 1; j < m_atfNumberRanges + 1; j++)
{
    m_pAtfRangeIn[j] = m_pAtfRangeIn[j - 1] + m_pAtfDelta[j - 1];
    m_pAtfRangeOut[j] = m_pAtfRangeOut[j - 1] + m_pAtfScale2[c][j - 1]
    * m_pAtfDelta[j - 1];
}

for (int j = 0; j < m_atfNumberRanges; j++)
{
    temp = m_pAtfRangeIn[j + 1] - m_pAtfRangeOut[j + 1] /
    m_pAtfScale2[c][j];
    m_pAtfOffset2[c][j] = temp;
}
```

[0446]

[0447] 일단 결정되면, 조각적 선형 모델이 알고리즘 2에서처럼 출력 샘플 값 outValue을 결정하기 위해 입력 샘플 값 inValue에 적용될 수 있다:

[0448] 알고리즘 2:

```
for (int j = 0; j < m_atfNumberRanges && skip == 0; j++)
{
    if (inValue <= m_pAtfRangeIn[j + 1])
    {
        outValue = (inValue - m_pAtfOffset2[j]) *
        m_pAtfScale2[j];
        skip = 1;
    }
}
```

[0449]

[0450] 알고리즘 3에서 역 프로세스가 수행됨:

[0451] 알고리즘 3:

```
for (int j = 0; j < m_atfNumberRanges && skip == 0; j++)
{
    if (inValue <= m_pAtfRangeOut[j + 1])
    {
        outValue = inValue / m_pAtfScale2[j] + m_pAtfOffset2
        [j];
        skip = 1;
    }
}
```

[0452]

[0453] 일부 예들에서, 두 개의 연접된 파티션들 i 및 i+1 사이의 경계 샘플 값 (m_pAtfRangeIn 또는 m_pAtfRangeOut

의 엔트리) 은 알고리즘 2 및 3에서 도시된 바와 같이 i 파티션에 속하는 대신, i+1에 속하는 것으로서 상이하게 해석될 수 있다.

[0455] 일부 예들에서, 알고리즘 3에서 보인 역 프로세스는, `m_pAtfScale2[j]`에 의한 나눗셈 대신, `m_pAtfInverseScale2` 값에 의한 곱셈으로 구현될 수 있다. 이러한 예들에서, `m_pAtfScale2[j]`의 값은 `m_pAtfScale2[j]`로부터 미리 결정된다.

[0456] 일부 예들에서, `m_pAtfInverseScale2[j]`는 디코더 측에서 $1/m_pAtfScale2[j]$ 으로서 결정된다.

[0457] 일부 예들에서, `m_pAtfInverseScale2[j]`는 인코더 측에서 컴퓨팅되고, 비트스트림을 통해 시그널링될 수 있다. 이러한 예들에서, 알고리즘 1, 2 및 3에서 주어진 동작은 그에 따라 조정될 것이다.

다양한 예들

[0459] 일부 예들에서, 제안된 시그널링 메커니즘이 입력 신호의 샘플들에 대한 동적 범위 조정을 가능하게 하는데, 예컨대, 비디오 코딩 시스템들의 압축 효율을 개선하는데 이용될 수 있는 조작적 함수를 모델링하기 위해 사용될 수 있다.

[0460] 일부 예들에서, 제안된 시그널링 메커니즘은 OETF에 의해, 예컨대, ST.2084의 PQ TF, 또는 다른 것들에 의해 생성되는 코드워드들 (R,G,B 샘플들의 비선형 표현)에 적용될 수 있는 조작적 함수를 모델링하는데 사용될 수 있다.

[0461] 일부 예들에서, 제안된 시그널링 메커니즘은 YCbCr 컬러들의 샘플들에 적용될 수 있는 조작적 함수를 모델링하는데 사용될 수 있다.

[0462] 일부 예들에서, 제안된 시그널링 메커니즘은 SDR 호환성을 갖는 HDR/WCG 해법들에 이용될 수 있는 조작적 함수를 모델링하는데 사용될 수 있다.

[0463] 일부 예들에서, 제안된 시그널링 메커니즘은 샘플들에 적용될 수 있는 조작적 함수를 부동소수점 표현으로 모델링하는데 사용될 수 있다. 또 다른 예에서, 제안된 시그널링 메커니즘 및 결과적인 함수는 정수 표현, 예컨대, 10 비트로 샘플들에 적용될 수 있다.

[0464] 일부 예들에서, 제안된 시그널링 메커니즘은 샘플들에 적용될 수 있는 조작적 함수를 루프 테이블들의 형태로 모델링하는데 사용될 수 있다. 또 다른 예들에서, 제안된 시그널링은 샘플에 적용될 수 있는 함수를 곱셈기의 형태로 모델링하는데 사용될 수 있다.

조합들 및 확장들

[0466] 위의 예들에서, 선형 모델이 각각의 지역 (즉, 스케일 더하기 오프셋)에 대해 가정되며; 본 개시물의 기법들은 고차 다항식 모델들에, 예를 들어, 두 개 대신 세 개의 파라미터들을 요구하는 2 차 다항식으로 또한 적용 가능할 수도 있다. 시그널링 및 신택스는 이 시나리오에 대해 적절히 확장될 것이다.

[0467] 위에서 설명된 양태들의 조합들이 가능하고 본 개시물의 기법들의 일부가 된다.

[0468] 도구상자 조합: 본 개시물에서 설명되는 SEI들의 목표들과 다소 유사한 목표들을 타겟으로 할 수 있는 여러 HDR 방법들이 있다. 그것들 중 하나를 초과하는 것을 수용하지만, 동시에, 프레임 당 가용 SEI 프로세싱의 수를 제한하기 위하여, 이를 방법들 (중 하나 이상의 방법)을 단일 SEI로 조합하는 것이 제안된다. 제안된 신택스 엘리먼트는 각각의 사례에 적용하는 특정 방법을 나타낼 것이다. 예를 들어, 두 개의 가능한 방법들이 SEI에 있다면, 신택스 엘리먼트는 사용될 하나를 나타내는 플래그일 것이다.

예 7

[0470] 이 예에서, 스케일 파라미터들의 시그널링은 음의 스케일들이 송신될 수 있도록 수정되고, 시그널링된 스케일 파라미터들은 다양한 성분들의 상이한 범위들에 적용될 스케일의 변동을 나타낸다. 예 5에 관한 변경들은 아래와 같다.

[0471]

SEI 메시지의 선택스에 대한 변경들

component_scale_info(payloadSize) {	디스크립터
comp_scale_id	ue(v)
comp_scale_cancel_flag	u(1)
if(!comp_scale_cancel_flag) {	
comp_scale_persistence_flag	u(1)
comp_scale_scale_bit_depth	u(4)
comp_scale_offset_bit_depth	u(4)
comp_scale_scale_frac_bit_depth	u(4)
comp_scale_offset_frac_bit_depth	u(4)
comp_scale_negative_scales_present_flag	u(1)
comp_scale_dep_component_id	ue(v)
comp_scale_num_comps_minus1	ue(v)
for(c = 0; c <= comp_scale_num_comps_minus1; c++) {	
comp_scale_num_ranges[c]	ue(v)
comp_scale_equal_ranges_flag[c]	u(1)
comp_scale_global_offset_val[c]	u(v)
for(i = 0; i <= comp_scale_num_ranges[c]; i++)	
comp_scale_scale_val[c][i]	u(v)
if(!comp_scale_equal_ranges[c])	u(v)
for(i = 0; i <= comp_scale_num_ranges[c]; i++)	
comp_scale_offset_val[c][i]	u(v)
}	
}	

[0472]

SEI 메시지의 시맨틱스에 대한 변경들

[0473]

1과 동일한 **comp_scale_negative_scales_present_flag**는 **comp_scale_scale_val[c][i]**로부터 도출되는 스케일 파라미터들의 정수 부분이 부호 있는 정수로서 표현됨을 특정한다. 0과 동일한 **comp_scale_negative_scales_present_flag**는 **comp_scale_scale_val[c][i]**로부터 도출되는 정수 부분 스케일 파라미터들이 부호 없는 정수로서 표현됨을 특정한다.

[0475]

하나의 대체예에서, 오프셋 파라미터들의 다른 세트는 제 2 성분의 값의 함수로서 제 1 성분에 대한 스케일과 함께 적용되는 오프셋을 정의하는데 사용되는 **comp_scale_scale_val**와 함께 시그널링된다.

[0476]

부호 있는 정수 표현은 2의 보수 표현법과 부호 있는 크기 표현 (한 비트는 부호를 위한 것이고 나머지 비트들은 정수 부분임) 을 비제한적으로 포함한다. 아래의 도출은 부호 있는 크기 표현에 대해 주어진다. 도출은 부호 있는 표현들의 다른 형태들에 대해 유사하게 정의될 수 있다.

[0477]

변수 CompScaleScaleVal[c][i]는 다음과 같이 도출된다:

```

compScaleScaleFracPart = ( comp_scale_scale_val[ c ][ i ] &
                           ( (1 << comp_scale_scale_frac_bit_depth ) -
                             1 ) )
                           (1 << comp_scale_scale_frac_bit_depth )

if( comp_scale_negative_scales_present_flag ) {
    compScaleSignPart = comp_scale_scale_val[ c ][ i ] >>
    (comp_scale_scale_bit_depth - 1)

    compScaleIntegerPart = comp_scale_scale_val[ c ][ i ] - ( compScaleSignPart
                                                               << (comp_scale_scale_bit_depth - 1) )

    compScaleIntegerVal = ( ( compScaleSignPart == 1 ) : -1 : 1 ) *
    compScaleIntegerPart
} else
    compScaleIntegerVal = comp_scale_scale_val[ c ][ i ] >>
    comp_scale_scale_frac_bit_depth

```

[0478]

```
CompScaleScaleVal[ c ][ i ] = compScaleIntegerVal + compScaleScaleFracPart
```

[0479]

comp_scale_negative_scale_present_flag가 1과 동일할 때, comp_scale_scale_bit_depth의 값은 comp_scale_scale_frac_bit_depth 이상일 것이라는 것이 비트스트림 적합성의 요건이다.

[0480]

comp_scale_dependent_component_id는 비디오의 다양한 성분들에 대한 스케일 및 오프셋 파라미터들의 적용을 특정한다. comp_scale_dependent_component_id가 0과 동일할 때, 신택스 엘리먼트 comp_scale_global_offset_val[c], comp_scale_scale_val[c][i] 및 comp_scale_offset_val[c][i]는 c-번째 성분의 입력 및 출력 값들의 매핑을 식별하는데 사용된다. comp_scale_dependent_component_id가 0보다 더 클 때, 신택스 엘리먼트 comp_scale_global_offset_val[c], comp_scale_scale_val[c][i] 및 comp_scale_offset_val[c][i]가 샘플의 (comp_scale_dependent_component_id - 1)-번째 성분의 값의 함수로서 그 샘플의 c-번째 성분에 적용될 스케일 파라미터의 매핑을 특정하도록 comp_scale_dependent_component_id - 1은 성분의 인덱스를 특정한다.

[0481]

시맨틱스의 나머지는 예 5에서 설명된 것들과 유사하다.

[0482]

예 8

[0483]

이 예에서, ATF 파라미터들의 비트 깊이는 성분에 따라 다르다. 각각의 성분에 대해, 신택스 엘리먼트들의 비트 깊이는 명시적으로 시그널링된다. 덧붙여서, 그들 신택스 엘리먼트들에 대한 디폴트 비트-깊이가 있다. 디폴트 값은 비트 깊이가 명시적으로 시그널링되지 않을 때 지정된다. 디폴트 값들이 적용되는지 또는 그것들이 명시적으로 시그널링되는지를 플래그가 나타낼 수 있다.

[0484]

아래의 표는 이를 개념들의 일 예를 도시한다. ATF 파라미터들의 신택스 엘리먼트들은 스케일 hdr_recon_scale_val[][][] 및 범위 hdr_recon_range_val[][][]이다. 대응하는 비트 깊이를 나타내는 신택스 엘리먼트들 (정수 및 소수 부분) 은 다음의 것들이며:

[0485]

- hdr_recon_scale_bit_depth[c],

[0486]

- hdr_recon_offset_bit_depth[c],

[0487]

- hdr_recon_scale_frac_bit_depth[c],

[0488]

- hdr_recon_offset_frac_bit_depth[c],

[0489]

여기서 c는 성분 인덱스이다. 스케일 및 오프셋 (범위) 에 대한 디폴트 비트-깊이들은 다음으로 설정될 수 있다:

- [0490] • **hdr_recon_scale_bit_depth[c] = 8,**
- [0491] • **hdr_recon_offset_bit_depth[c] = 8,**
- [0492] • **hdr_recon_scale_frac_bit_depth[c] = 6,**
- [0493] • **hdr_recon_offset_frac_bit_depth[c] = 8.**

[0494] 파라미터들의 정확도는 ATF 파라미터들 및 컬러 조정 파라미터들에 대해 또한 상이할 수 있다. 또한, 디폴트는 성분마다 그리고 컬러 조정 파라미터들에 대해 상이할 수 있다. 이 예에서, 디폴트들은 동일한 것으로 가정된다.

<code>hdr_reconstruction_info(payloadSize)</code>	디스크립터
hdr_recon_id	ue(v)
hdr_recon_cancel_flag	u(1)
<code>if(!hdr_recon_cancel_flag) {</code>	
hdr_recon_persistence_flag	u(1)
if (hdr_recon_id == 1) {	
hdr_output_full_range_flag	
hdr_output_colour_primaries	
hdr_output_transfer_characteristics	
hdr_output_matrix_coeffs	
}	
SYNTAX FOR THE MAPPING LUTs	
hdr_recon_num_comps_minus1	ue(v)
<code>for(c = 0; c <= hdr_recon_num_comps_minus1;</code>	
<code>c++) {</code>	
hdr_recon_default_bit_depth[c]	u(1)
if (hdr_recon_default_bit_depth[c] == 0) {	
hdr_recon_scale_bit_depth[c]	u(4)
hdr_recon_offset_bit_depth[c]	u(4)
hdr_recon_scale_frac_bit_depth[c]	u(4)
hdr_recon_offset_frac_bit_depth[c]	u(4)
}	
hdr_recon_num_ranges[c]	ue(v)
hdr_recon_equal_ranges_flag[c]	u(1)
hdr_recon_global_offset_val[c]	u(v)
<code>for(i = 0; i <= hdr_recon_num_ranges[c];</code>	
<code>i++)</code>	
hdr_recon_scale_val[c][i]	u(v)
if(!hdr_recon_equal_ranges[c])	u(v)
for(i = 0; i <= hdr_recon_num_ranges[c];	
i++)	
hdr_recon_range_val[c][i]	u(v)

- [0495]

}	u(v)
SYNTAX FOR THE COLOR CORRECTION PART	
if (hdr_recon_id == 1) {	Params related to Colour correction
hdr_color_correction_type	0: on U,V - 1: on R,G,B
hdr_color_accuracy_flag	Syntax for coding the colour
if(!hdr_recon_color_accuracy_flag) {	correction LUT
hdr_color_scale_bit_depth	u(4)
hdr_color_offset_bit_depth	u(4)
hdr_color_scale_frac_bit_depth	u(4)
hdr_color_offset_frac_bit_depth	u(4)
}	
color_correction_num_ranges	
color_correction_equal_len_ranges_flag	
color_correction_zero_offset_val	
for(i = 0; i < color_correction_num_ranges;	
i++)	
color_correction_scale_val[i]	
if(!color_correction_equal_len_ranges_flag)	
for(i = 0; i < color_correction_num_ranges;	
i++)	
color_correction_range_val[i]	
}	
}	
}	
}	

[0496]

예 9

[0498]

새로운 HDR 해법의 바람직한 특성은 그것이 HDR10같은 이전의 HDR 해법들과 하위 호환 가능하다는 것이다. 선택스 엘리먼트가 이것이 그 경우임을 나타낼 수도 있다. 이는 비트스트림의 특성을 나타내고, ATF 버전이 이미 가시적이지 않다면 HDR 디코더가 일부 환경들 하에서 역 ATF 프로세싱에 대해 계산 리소스들을 소비하지 않기로 결정할 수 있다.

[0499]

하나의 예에서, `hdr_recon_id` 선택스 엘리먼트의 일부 값들은 HDR10 하위 호환성, 또는 어느 정도 하위 호환성이 있는지를 나타내기 위해 예약된다.

[0500]

다른 예에서, 플래그 (`hdr_recon_hdr10_bc`) 가 이 상황을 나타낸다.

[0501]

하나의 예에서, 시그널링된 HDR10 하위 호환성은 비트스트림이 가시적임을 나타낸다. 대안적으로, 그것은 시그널링된 값들의 일부 구체적인 특성들: 예를 들어, 이 특성을 보장하는 값들의 범위를 나타낼 수도 있다. 예를 들면, 스케일이 0.9와 1.1 사이라는 제약조건이 있을 수 있다.

[0502]

도 10은 본 개시물의 기법들을 구현할 수도 있는 비디오 인코더 (20) 의 일 예를 도시하는 블록도이다. 비디오 인코더 (20) 는 비디오 프리프로세서 (19) 에 의해 프로세싱된 타겟 컬러 컨테이너에서의 비디오 슬라이스들 내의 비디오 블록들의 인트라 코딩 및 인터 코딩을 수행할 수도 있다. 인트라 코딩은 공간적 예측에 의존하여, 주어진 비디오 프레임 또는 픽처 내의 비디오에서 공간적 리던던시를 감소시키거나 또는 제거한다. 인터 코딩은 시간적 예측에 의존하여, 비디오 시퀀스의 인접한 프레임들 또는 픽처들 내의 비디오에서 시간적 리던던시를 감소시키거나 또는 제거한다. 인트라 모드 (I 모드) 는 여러 공간 기반 코딩 모드들 중 임의의

것을 지칭할 수도 있다. 단방향 예측 (P 모드) 또는 양-예측 (B 모드) 과 같은 인터 모드들은 여러 시간 기반 코딩 모드들 중 임의의 것을 지칭할 수도 있다.

[0503] 도 10에 도시된 바와 같이, 비디오 인코더 (20)는 인코딩될 비디오 프레임 내의 현재 비디오 블록을 수신한다.

도 10의 예에서, 비디오 인코더 (20)는 모드 선택 부 (40), 비디오 데이터 메모리 (40), 디코딩된 픽처 버퍼 (64), 합산기 (50), 변환 프로세싱 부 (52), 양자화 부 (54), 및 엔트로피 인코딩 부 (56)를 포함한다.

모드 선택 부 (40)는, 차례로, 모션 보상 부 (44), 모션 추정 부 (42), 인트라 예측 프로세싱 부 (46), 및 파티션 부 (48)를 포함한다. 비디오 블록 복원을 위해, 비디오 인코더 (20)는 역 양자화 부 (58), 역 변환 부 (60), 및 합산기 (62)를 또한 포함한다. 블록화제거 필터 (도 10에서 도시되지 않음)가 블록 경계들을 필터링하여 복원된 비디오로부터 블록현상 (blockiness) 아티팩트들을 제거하기 위해 또한 포함될 수도 있다.

원한다면, 블록화제거 필터는 합산기 (62)의 출력을 통상 필터링할 것이다. 추가적인 필터들 (인 루프 또는 포스트 루프)이 블록화제거 필터에 부가하여 또한 사용될 수도 있다. 그런 필터들은 간결함을 위해 도시되지 않았지만, 원한다면, (인 루프 필터로서) 합산기 (50)의 출력을 필터링할 수도 있다.

[0504] 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)가, 예컨대, 인트라 코딩 또는 인터 코딩 모드들에서 비디오 인코더 (20)에 의해 비디오 데이터를 인코딩함에 있어서의 사용을 위한 참조 비디오 데이터를 저장하는 참조 픽처 메모리일 수도 있다. 비디오 데이터 메모리 (41)와 디코딩된 픽처 버퍼 (64)는 동기식 DRAM (SDRAM)을 포함한 다이나믹 랜덤 액세스 메모리 (DRAM), 자기저항 RAM (MRAM), 저항 RAM (RRAM), 또는 다른 유형들의 메모리 디바이스들과 같은 다양한 메모리 디바이스들 중 임의의 것에 의해 형성될 수도 있다. 비디오 데이터 메모리 (41)와 디코딩된 픽처 버퍼 (64)는 동일한 메모리 디바이스 또는 별개의 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에서, 비디오 데이터 메모리 (41)는 비디오 인코더 (20)의 다른 컴포넌트들과 온-칩, 또는 그들 컴포넌트들에 대하여 오프-칩일 수도 있다.

[0505] 인코딩 프로세스 동안, 비디오 인코더 (20)는 코딩될 비디오 프레임 또는 슬라이스를 수신한다. 프레임 또는 슬라이스는 다수의 비디오 블록들로 세분될 수도 있다. 모션 추정 부 (42)와 모션 보상 부 (44)는 시간적 예측을 제공하기 위해 하나 이상의 참조 프레임들에서 하나 이상의 블록들에 관하여 수신된 비디오 블록의 인터 예측성 코딩을 수행한다. 인트라 예측 프로세싱 부 (46)는 공간적 예측을 제공하기 위해 코딩될 블록과 동일한 프레임 또는 슬라이스에서의 하나 이상의 이웃 블록들에 대해, 수신된 비디오 블록의 인트라 예측성 코딩을 대신 수행할 수도 있다. 비디오 인코더 (20)는 다수의 코딩 과정들을, 예컨대, 비디오 데이터의 각각의 블록에 대한 적절한 코딩 모드를 선택하기 위해 수행할 수도 있다.

[0506] 더구나, 파티션 부 (48)는 이전의 코딩 과정들에서의 이전의 파티셔닝 스킴들의 평가에 기초하여, 비디오 데이터의 블록들을 서브 블록들로 파티셔닝할 수도 있다. 예를 들어, 파티션 부 (48)는 초기에는 프레임 또는 슬라이스를 LCU들로 파티셔닝하고, 레이트 왜곡 분석 (예컨대, 레이트 왜곡 최적화)에 기초하여 그 LCU들의 각각을 서브 CU들로 파티셔닝할 수도 있다. 모드 선택 부 (40)는 LCU의 서브 CU들로의 파티셔닝을 나타내는 큐드트리 데이터 구조를 추가로 생성할 수도 있다. 큐드트리의 리프 노드 CU들은 하나 이상의 PU들과 하나 이상의 TU들을 포함할 수도 있다.

[0507] 모드 선택 부 (40)는 코딩 모드들인 인트라 또는 인터 중의 하나를, 예컨대 여러 결과들에 기초하여 선택할 수도 있고, 결과적인 인트라 코딩된 또는 인터 코딩된 블록을, 합산기 (50)에 제공하여 잔차 블록 데이터를 생성하고, 합산기 (62)에 제공하여 참조 프레임으로서 사용하기 위한 인코딩된 블록을 복원한다. 모드 선택 부 (40)는, 선택스 엘리먼트들, 이를테면 모션 벡터들, 인트라 모드 표시자들, 파티션 정보, 및 다른 이러한 선택스 정보를 엔트로피 인코딩 부 (56)에 또한 제공한다.

[0508] 모션 추정 부 (42)와 모션 보상 부 (44)는 고도로 통합될 수도 있지만 개념적 목적들을 위해 별개로 예시된다. 모션 추정 부 (42)에 의해 수행되는 모션 추정은 비디오 블록들에 대한 모션을 추정하는 모션 벡터들을 생성하는 프로세스이다. 모션 벡터가, 예를 들어, 현재 픽처 내의 코딩되고 있는 현재 블록 (또는 다른 코딩되는 유닛)을 기준으로 참조 픽처 내의 예측성 블록 (또는 다른 코딩된 유닛)에 대한 현재 비디오 프레임 또는 픽처 내의 비디오 블록의 PU의 변위를 나타낼 수도 있다. 예측성 블록이 차의 절대값 합 (SAD), 차의 제곱 합 (SSD), 또는 다른 차이 메트릭들에 의해 결정될 수도 있는, 화소 차이의 관점에서 코딩될 블록에 밀접하게 매칭된다고 생각되는 블록이다. 일부 예들에서, 비디오 인코더 (20)는 디코딩된 픽처 버퍼 (64)에 저장된 참조 픽처들의 부 정수 (sub-integer) 화소 위치들에 대한 값들을 계산할 수도 있다. 예

를 들어, 비디오 인코더 (20) 는 참조 픽처의 1/4 화소 위치들, 1/8 화소 위치들, 또는 다른 분수 화소 위치들의 값들을 보간할 수도 있다. 그러므로, 모션 추정 부 (42) 는 풀 (full) 화소 위치들 및 분수 화소 위치들에 대한 모션 검색을 수행하여 분수 화소 정밀도를 갖는 모션 벡터를 출력할 수도 있다.

[0509] 모션 추정 부 (42) 는 PU의 위치와 참조 픽처의 예측성 블록의 위치를 비교함으로써 인터 코딩된 슬라이스에서의 비디오 블록의 PU에 대한 모션 벡터를 계산한다. 참조 픽처는 디코딩된 픽처 버퍼 (64) 에 저장된 하나 이상의 참조 픽처들을 각각 식별하는 제 1 참조 픽처 리스트 (List 0) 또는 제 2 참조 픽처 리스트 (List 1) 로부터 선택될 수도 있다. 모션 추정 부 (42) 는 계산된 모션 벡터를 엔트로피 인코딩 부 (56) 와 모션 보상 부 (44) 로 전송한다.

[0510] 모션 보상 부 (44) 에 의해 수행되는 모션 보상은 모션 추정 부 (42) 에 의해 결정된 모션 벡터에 기초하여 예측성 블록을 폐치하는 것 또는 생성하는 것을 수반할 수도 있다. 다시, 모션 추정 부 (42) 와 모션 보상 부 (44) 는 몇몇 예들에서 기능적으로 통합될 수도 있다. 현재 비디오 블록의 PU에 대한 모션 벡터를 수신 시, 모션 보상 부 (44) 는 참조 픽처 리스트들 중 하나의 참조 픽처 리스트에서 모션 벡터가 가리키는 예측성 블록을 찾을 수도 있다. 합산기 (50) 는, 아래에서 논의되는 바와 같이, 코딩되고 있는 현재 비디오 블록의 화소 값들로부터 예측성 블록의 화소 값들을 감산하여 화소 차이 값들을 형성함으로써 잔차 비디오 블록을 형성한다. 대체로, 모션 추정 부 (42) 는 루마 성분들에 관하여 모션 추정을 수행하고, 모션 보상 부 (44) 는 크로마 성분들 및 루마 성분들 양쪽 모두에 대해 루마 성분들에 기초하여 계산된 모션 벡터들을 사용한다. 모드 선택 부 (40) 는 비디오 슬라이스의 비디오 블록들을 디코딩함에 있어서 비디오 디코더 (30) 에 의한 사용을 위해 비디오 블록들 및 비디오 슬라이스에 연관된 신택스 엘리먼트들을 또한 생성할 수도 있다.

[0511] 인트라 예측 프로세싱 부 (46) 는, 위에서 설명된 바와 같이, 모션 추정 부 (42) 및 모션 보상 부 (44) 에 의해 수행된 인터 예측에 대한 대안으로서 현재 블록을 인트라 예측할 수도 있다. 특히, 인트라 예측 프로세싱 부 (46) 는 현재 블록을 인코딩하는데 사용하기 위한 인트라 예측 모드를 결정할 수도 있다. 일부 예들에서, 인트라 예측 프로세싱 부 (46) 는 예컨대, 개별 인코딩 과정들 동안에 다양한 인트라 예측 모드들을 사용하여 현재 블록을 인코딩할 수도 있고, 인트라 예측 프로세싱 부 (46) (또는 일부 예들에서, 모드 선택 부 (40)) 는 테스트된 모드들로부터 사용할 적절한 인트라 예측 모드를 선택할 수도 있다.

[0512] 예를 들어, 인트라 예측 프로세싱 부 (46) 는 다양한 테스트된 인트라 예측 모드들에 대한 레이트 왜곡 분석을 사용하여 레이트 왜곡 값을 계산하고, 테스트된 모드들 중에서 최상의 레이트 왜곡 특성을 갖는 인트라 예측 모드를 선택할 수도 있다. 레이트 왜곡 분석은 일반적으로, 인코딩된 블록과 인코딩된 블록을 생성하기 위해 인코딩되었던 원래의 인코딩되지 않은 블록 사이의 왜곡 (또는 에러) 의 양, 뿐만 아니라 인코딩된 블록을 생성하는데 사용된 비트 레이트 (다시 말하면, 비트들의 수) 를 결정한다. 인트라 예측 프로세싱 부 (46) 는 어떤 인트라 예측 모드가 그 블록에 대한 최상의 레이트 왜곡 값을 나타내는지를 결정하기 위해 다양한 인코딩된 블록들에 대한 왜곡들 및 레이트들로부터 비율들을 계산할 수도 있다.

[0513] 블록에 대해 인트라 예측 모드를 선택한 후, 인트라 예측 프로세싱 부 (46) 는 그 블록에 대한 선택된 인트라 예측 모드를 나타내는 정보를 엔트로피 인코딩 부 (56) 에 제공할 수도 있다. 엔트로피 인코딩 부 (56) 는 선택된 인트라 예측 모드를 나타내는 정보를 인코딩할 수도 있다. 비디오 인코더 (20) 는 송신되는 비트스트림에 구성 데이터를 포함시킬 수도 있는데, 이 구성 데이터는 복수의 인트라 예측 모드 인덱스 테이블들 및 복수의 수정된 인트라 예측 모드 인덱스 테이블들 (코드워드 매핑 테이블들이라고 또한 지칭됨), 다양한 블록들에 대한 콘텍스트들을 인코딩하는 정의들, 그리고 그 콘텍스트들의 각각에 대한 사용을 위한 가장 가능성 있는 인트라 예측 모드, 인트라 예측 모드 인덱스 테이블, 및 수정된 인트라 예측 모드 인덱스 테이블의 표시들을 포함할 수도 있다.

[0514] 비디오 인코더 (20) 는 코딩되고 있는 원래의 비디오 블록으로부터 모드 선택 부 (40) 로부터의 예측 데이터를 감산함으로써 잔차 비디오 블록을 형성한다. 합산기 (50) 는 이 감산 동작을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다. 변환 프로세싱 부 (52) 는 변환, 이를테면 이산 코사인 변환 (DCT) 또는 개념적으로 유사한 변환을 잔차 블록에 적용하여, 잔차 변환 계수 값을 포함하는 비디오 블록을 생성한다. 변환 프로세싱 부 (52) 는 DCT에 개념적으로 유사한 다른 변환들을 수행할 수도 있다. 웨이브릿 변환들, 정수 변환들, 서브 벤드 변환들 또는 다른 유형들의 변환들이 또한 이용될 수 있다. 어느 경우에나, 변환 프로세싱 부 (52) 는 잔차 블록에 변환을 적용하여, 잔차 변환 계수들의 블록을 생성한다. 그 변환은 잔차 정보를 화소 값 도메인으로부터 변환 도메인, 이를테면 주파수 도메인으로 변환할 수도 있다. 변환 프로세싱 부 (52) 는 결과적인 변환 계수들을 양자화 부 (54) 에 전송할 수도 있다.

- [0515] 양자화 부 (54) 는 변환 계수들을 양자화하여 비트 레이트를 더욱 감소시킨다. 양자화 프로세스는 그 계수들의 일부 또는 전부에 연관된 비트 깊이를 감소시킬 수도 있다. 양자화 정도는 양자화 파라미터를 조정함으로써 수정될 수도 있다. 일부 예들에서, 양자화 부 (54) 는 그 후, 양자화된 변환 계수들을 포함하는 매트릭스의 스캔을 수행할 수도 있다. 대안적으로, 엔트로피 인코딩 부 (56) 가 그 스캔을 수행할 수도 있다.
- [0516] 양자화를 뒤따라, 엔트로피 인코딩 부 (56) 는 양자화된 변환 계수들을 엔트로피 코딩한다. 예를 들어, 엔트로피 인코딩 부 (56) 는 콘텍스트 적응 가변 길이 코딩 (CAVLC), 콘텍스트 적응 이진 산술 코딩 (CABAC), 신택스 기반 콘텍스트 적응 이진 산술 코딩 (SBAC), 확률 간격 구획화 엔트로피 (PIPE) 코딩 또는 다른 엔트로피 코딩 기법을 수행할 수도 있다. 콘텍스트 기반 엔트로피 코딩의 경우, 콘텍스트는 이웃 블록들에 기초할 수도 있다. 엔트로피 코딩 부 (56) 에 의한 엔트로피 코딩에 뒤이어, 인코딩된 비트스트림은 다른 디바이스 (예컨대, 비디오 디코더 (30)) 로 송신되거나 또는 나중의 송신 또는 취출을 위해 보관될 수도 있다.
- [0517] 역 양자화 부 (58) 및 역 변환 프로세싱 부 (60) 는 역 양자화 및 역 변환을 각각 적용하여, 화소 도메인에서 잔차 블록을, 예컨대, 나중에 참조 블록으로서 사용하기 위해 복원한다. 모션 보상 부 (44) 는 잔차 블록을 디코딩된 픽처 버퍼 (64) 의 프레임들 중 하나의 프레임의 예측성 블록에 가산함으로써 참조 블록을 계산할 수도 있다. 모션 보상 부 (44) 는 또한 하나 이상의 보간 필터들을 복원된 잔차 블록에 적용하여 모션 추정에서의 사용을 위한 부 정수 화소 값들을 계산할 수도 있다. 합산기 (62) 는 복원된 잔차 블록을 모션 보상 부 (44) 에 의해 생성된 모션 보상 예측 블록에 가산하여, 디코딩된 픽처 버퍼 (64) 에 저장하기 위한 복원된 비디오 블록을 생성한다. 복원된 비디오 블록은 모션 추정 부 (42) 및 모션 보상 부 (44) 에 의해 후속 비디오 프레임에서의 블록을 인터 코딩하기 위한 참조 블록으로서 사용될 수도 있다.
- [0518] 도 11은 본 개시물의 기법들을 구현할 수도 있는 비디오 디코더 (30) 의 일 예를 도시하는 블록도이다. 특히, 비디오 디코더 (30) 는 비디오 데이터를, 위에서 설명된 바와 같이, 비디오 포스트프로세서 (31) 에 의해 그 다음에 프로세싱될 수도 있는 타겟 컬러 표현으로 디코딩할 수도 있다. 도 11의 예에서, 비디오 디코더 (30) 는 엔트로피 디코딩 부 (70), 비디오 데이터 메모리 (71), 모션 보상 부 (72), 인트라 예측 프로세싱 부 (74), 역 양자화 부 (76), 역 변환 프로세싱 부 (78), 디코딩된 픽처 버퍼 (82) 및 합산기 (80) 를 포함한다. 비디오 디코더 (30) 는, 일부 예들에서, 비디오 인코더 (20) (도 10) 에 관해 설명된 인코딩 과정에 일반적으로 역인 디코딩 과정을 수행할 수도 있다. 모션 보상 부 (72) 는 엔트로피 디코딩 부 (70) 로부터 수신된 모션 벡터들에 기초하여 예측 데이터를 생성할 수도 있는 한편, 인트라 예측 프로세싱 부 (74) 는 엔트로피 디코딩 부 (70) 로부터 수신된 인트라 예측 모드 표시자들에 기초하여 예측 데이터를 생성할 수도 있다.
- [0519] 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 국부 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 비트스트림으로부터의 인코딩된 비디오 데이터를 저장하는 코딩된 픽처 버퍼 (coded picture buffer, CPB) 를 형성할 수도 있다. 디코딩된 픽처 버퍼 (82) 가, 예컨대 인트라 코딩 또는 인터 코딩 모드들에서, 비디오 디코더 (30) 에 의해 비디오 데이터를 디코딩함에 있어서의 사용을 위한 참조 비디오 데이터를 저장하는 참조 픽처 메모리일 수도 있다. 비디오 데이터 메모리 (71) 와 디코딩된 픽처 버퍼 (82) 는 동기식 DRAM (SDRAM) 을 포함한 다이나믹 랜덤 액세스 메모리 (DRAM), 자기저항 RAM (MRAM), 저항 RAM (RRAM), 또는 다른 유형들의 메모리 디바이스들과 같은 다양한 메모리 디바이스들 중 임의의 것에 의해 형성될 수도 있다. 비디오 데이터 메모리 (71) 와 디코딩된 픽처 버퍼 (82) 는 동일한 메모리 디바이스 또는 별개의 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에서, 비디오 데이터 메모리 (71) 는 비디오 디코더 (30) 의 다른 컴포넌트들과 온-칩, 또는 그들 컴포넌트들에 대하여 오프-칩일 수도 있다.
- [0520] 디코딩 프로세스 동안, 비디오 디코더 (30) 는 인코딩된 비디오 슬라이스의 비디오 블록들 및 연관된 신택스 엘리먼트들을 나타내는 인코딩된 비디오 비트스트림을 비디오 인코더 (20) 로부터 수신한다. 비디오 디코더 (30) 의 엔트로피 디코딩 부 (70) 는 그 비트스트림을 엔트로피 디코딩하여 양자화된 계수들, 모션 벡터들 또는 인트라 예측 모드 표시자들, 및 다른 신택스 엘리먼트들을 생성한다. 엔트로피 디코딩 부 (70) 는 모션 벡터들을 그리고 다른 신택스 엘리먼트들을 모션 보상 부 (72) 로 포워딩한다. 비디오 디코더 (30) 는 비디오 슬라이스 레벨 및/또는 비디오 블록 레벨에서 신택스 엘리먼트들을 수신할 수도 있다.
- [0521] 비디오 슬라이스가 인트라 코딩된 (I) 슬라이스로서 코딩되는 경우, 인트라 예측 프로세싱 부 (74) 는 현재 비

디오 슬라이스의 비디오 블록에 대한 예측 데이터를 현재 프레임 또는 꾹처의 이전에 디코딩된 블록들로부터의 시그널링된 인트라 예측 모드 및 데이터에 기초하여 생성할 수도 있다. 비디오 프레임이 인터 코딩된 (즉, B 또는 P) 슬라이스로서 코딩되는 경우, 모션 보상 부 (72)는 엔트로피 디코딩 부 (70)로부터 수신된 모션 벡터들 및 다른 신택스 엘리먼트들에 기초하여, 현재 비디오 슬라이스의 비디오 블록에 대한 예측성 블록들을 생성한다. 그 예측성 블록들은 참조 꾹처 리스트들 중 하나의 참조 꾹처 리스트 내의 참조 꾹처들 중 하나의 참조 꾹처로부터 생성될 수도 있다. 비디오 디코더 (30)는 디코딩된 꾹처 베퍼 (82)에 저장된 참조 꾹처들에 기초하여 디폴트 구축 기법들을 사용하여, 참조 꾹처 리스트들 (List 0 및 List 1)을 구축할 수도 있다.

모션 보상 부 (72)는 모션 벡터들 및 다른 신택스 엘리먼트들을 파싱함으로써 현재 비디오 슬라이스의 비디오 블록에 대한 예측 정보를 결정하고, 그 예측 정보를 사용하여 디코딩되고 있는 현재 비디오 블록에 대한 예측성 블록들을 생성한다. 예를 들어, 모션 보상 부 (72)는 수신된 신택스 엘리먼트들의 일부를 이용하여, 비디오 슬라이스의 비디오 블록들을 코딩하는데 사용된 예측 모드 (예컨대, 인트라 또는 인터 예측), 인터 예측 슬라이스 유형 (예컨대, B 슬라이스 또는 P 슬라이스), 슬라이스에 대한 참조 꾹처 리스트들 중 하나 이상에 대한 구축 정보, 슬라이스의 각각의 인터 인코딩된 비디오 블록에 대한 모션 벡터들, 슬라이스의 각각의 인터 코딩된 비디오 블록에 대한 인터 예측 스테이터스, 및 현재 비디오 슬라이스 내의 비디오 블록들을 디코딩하기 위한 다른 정보를 결정한다.

[0522] 모션 보상 부 (72)는 보간 필터들에 기초하여 보간을 또한 수행할 수도 있다. 모션 보상 부 (72)는 비디오 블록들의 인코딩 동안에 비디오 인코더 (20)에 의해 사용된 것과 같은 보간 필터들을 사용하여 참조 블록들의 부 정수 화소들에 대한 보간된 값들을 계산할 수도 있다. 이 경우, 모션 보상 부 (72)는 수신된 신택스 엘리먼트들로부터 비디오 인코더 (20)에 의해 사용된 보간 필터들을 결정하고 그 보간 필터들을 사용하여 예측성 블록들을 생성할 수도 있다.

[0523] 역 양자화 부 (76)는 비트스트림으로 제공되고 엔트로피 디코딩 부 (70)에 의해 디코딩된 양자화된 변환 계수들을 역 양자화, 즉, 탈 양자화한다. 역 양자화 프로세스는 양자화 정도와, 마찬가지로 적용되어야 할 역 양자화의 정도를 결정하기 위해, 비디오 슬라이스에서의 각각의 비디오 블록에 대해 비디오 디코더 (30)에 의해 계산된 양자화 파라미터 (QP_Y)의 사용을 포함할 수도 있다. 역 변환 프로세싱 부 (78)는 화소 도메인에서 잔차 블록들을 생성하기 위하여 역 변환, 예컨대, 역 DCT, 역 정수 변환, 또는 개념적으로 유사한 역 변환 프로세스를 변환 계수들에 적용한다.

[0524] 모션 보상 부 (72)가 현재 비디오 블록에 대한 예측성 블록을 모션 벡터들 및 다른 신택스 엘리먼트들에 기초하여 생성한 후, 비디오 디코더 (30)는 역 변환 프로세싱 부 (78)로부터의 잔차 블록들을 모션 보상 부 (72)에 의해 생성된 대응하는 예측성 블록들과 합산함으로써 디코딩된 비디오 블록을 형성한다. 합산기 (80)는 이 합산 동작을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다. 원한다면, 블록화제거 필터가 블록현상 아티팩트들을 제거하기 위하여 디코딩된 블록들을 필터링하는데 또한 적용될 수도 있다. 다른 루프 필터들이 (코딩 루프 내 또는 코딩 루프 후 중 어느 하나에서) 화소 전환 (transition)들을 부드럽게 하기 위해, 또는 그렇지 않으면 비디오 품질을 개선하기 위해 또한 사용될 수도 있다. 주어진 프레임 또는 꾹처에서의 디코딩된 비디오 블록들은 그러면 디코딩된 꾹처 베퍼 (82)에 저장되며, 그 디코딩된 꾹처 베퍼는 후속 모션 보상을 위해 사용되는 참조 꾹처들을 저장한다. 디코딩된 꾹처 베퍼 (82)는 디스플레이 디바이스, 이를테면 도 1의 디스플레이 디바이스 (32) 상의 나중의 프레젠테이션을 위해 디코딩된 비디오를 또한 저장한다.

[0525] 도 12는 본 개시물의 일 예시적 비디오 프로세싱 기법을 도시하는 흐름도이다. 도 12의 기법들은 비디오 인코더 (20) 및/또는 비디오 프리프로세서 부 (19)에 의해 수행될 수도 있다. 도 12의 예에서, 소스 디바이스 (12)는 카메라를 이용하여 비디오 데이터를 캡처하도록 구성될 수도 있다 (1200). 비디오 인코더 (20) 및/또는 비디오 프리프로세서 부 (19)는 고정 소수점 컴퓨팅을 이용하여 비디오 데이터에 대해 동적 범위 조정 프로세스를 수행하도록 구성될 수도 있다 (1210). 비디오 인코더 (20) 및/또는 비디오 프리프로세서 부 (19)는 또한, 고정 소수점 컴퓨팅을 이용하여 동적 범위 조정 프로세스에 대한, 역 동적 범위 조정 프로세스를 수행하기 위한 파라미터들을 결정하는 방법을 규정하는 정보를 포함하는 하나 이상의 신택스 엘리먼트들을 생성하도록 구성될 수도 있다 (1220).

[0526] 일 예에서, 비디오 인코더 (20) 및/또는 비디오 프리프로세서 부 (19)는 하나 이상의 보충 강화 정보 (SEI) 메시지들에서 하나 이상의 신택스 엘리먼트들을 생성하는 것에 의해 하나 이상의 신택스 엘리먼트들을 생성하도록 구성될 수도 있다. 일 예에서, 범위 파라미터, 스케일 파라미터, 또는 오프셋 파라미터 중 하나 이상을 포함한다. 다른 예에서, 정보는 범위 파라미터를 결정하는데 이용되는 분수 비트들의 제 1 수, 스케일 파라미

터를 결정하는데 이용되는 분수 비트들의 제 2 수, 및 오프셋 파라미터를 결정하는데 이용되는 분수 비트들의 제 3 수 중 하나 이상을 나타낸다. 다른 예에서, 정보는 비디오 데이터의 하나 이상의 컬러 컴포넌트들에 대한 최소 값 및 최대 값을 포함한다. 다른 예에서, 정보는 디코딩된 비디오 데이터의 미리 정의된 범위의 샘플 값들에 대한 인덱스를 포함한다.

[0527] 도 13 은 본 개시물의 다른 예시적 비디오 프로세싱 기법을 도시하는 흐름도이다. 도 13 의 기법들은 비디오 디코더 (30) 및/또는 비디오 포스트프로세서 부 (31) 에 의해 수행될 수도 있다. 본 개시물의 일 예에서, 비디오 디코더 (30) 및/또는 비디오 포스트프로세서 부 (31) 는 역 동적 범위 조정 프로세스를 수행하기 위한 파라미터들을 결정하기 위한 방법을 규정하는 정보를 포함하는 하나 이상의 선택스 엘리먼트들을 수신하고 (1300), 그리고 디코딩된 비디오 데이터를 수신하도록 (1310) 구성될 수도 있다.

[0528] 비디오 디코더 (30) 및/또는 비디오 포스트프로세서 부 (31) 는 또한 수신된 정보로부터 역 동적 범위 조정 프로세스에 대한 파라미터들을 결정하고 (1320), 그리고 수신된 정보 및 결정된 파라미터들에 따라 고정 소수점 컴퓨팅을 이용하여 디코딩된 비디오 데이터에 대한 역 동적 범위 조정 프로세스를 수행하도록 (1330) 구성될 수도 있다. 목적지 디바이스 (14) 는 또한, 디코딩된 비디오 데이터에 대한 역 동적 범위 조정 프로세스를 수행한 후 디코딩된 비디오 데이터를 디스플레이하도록 (1340) 구성될 수도 있다.

[0529] 본 개시물의 일 예에서, 비디오 디코더 (30) 및/또는 비디오 포스트프로세서 부 (31) 는 하나 이상의 보충 강화 정보 (SEI) 메시지들에서 하나 이상의 선택스 엘리먼트들을 수신하도록 구성될 수도 있다. 일 예에서, 범위 파라미터, 스케일 파라미터, 또는 오프셋 파라미터 중 하나 이상을 포함한다. 다른 예에서, 정보는 범위 파라미터를 결정하는데 이용되는 분수 비트들의 제 1 수, 스케일 파라미터를 결정하는데 이용되는 분수 비트들의 제 2 수, 및 오프셋 파라미터를 결정하는데 이용되는 분수 비트들의 제 3 수 중 하나 이상을 나타낸다.

[0530] 본 개시물의 다른 예에서, 비디오 디코더 (30) 및/또는 비디오 포스트프로세서 부 (31) 는 파라미터들을 결정하는데 이용되는 임의의 중간 계산 프로세스들 동안의 임의의 분수 비트들을 누적하는 것에 의해 분수 비트들의 제 1 수, 분수 비트들의 제 2 수, 또는 분수 비트들의 제 3 수 중 적어도 하나가 서로 상이한 경우에 파라미터들을 결정하고, 그리고 미리 정해진 분수 정확도에 기초하여 파라미터들을 결정하기 위한 최종 결과를 클립핑하도록 구성될 수도 있다.

[0531] 본 개시물의 다른 예에서, 비디오 디코더 (30) 및/또는 비디오 포스트프로세서 부 (31) 는 파라미터들을 결정하는데 이용되는 모든 중간 계산 프로세스들 동안 원하는 분수 정확도에 대해 임의의 분수 비트들을 절삭하는 것에 의해 파라미터들을 결정하도록 구성될 수도 있다.

[0532] 다른 예에서, 디코딩된 비디오 데이터의 하나 이상의 컬러 성분들에 대한 최소 값 및 최대 값을 포함하고, 비디오 디코더 (30) 및/또는 비디오 포스트프로세서 부 (31) 는 수신된 최소 값 및 수신된 최대 값에 기초하여 파라미터들을 결정하도록 구성될 수도 있다.

[0533] 다른 예에서, 정보는 디코딩된 비디오 데이터의 하나 이상의 컬러 성분들에 대한 샘플 값들의 미리 정해진 범위들에 대한 인덱스를 포함하고, 비디오 디코더 (30) 및/또는 비디오 포스트프로세서 부 (31) 는 수신된 인덱스에 기초하여 디코딩된 비디오 데이터의 하나 이상의 컬러 성분들에 대한 최소 값 및 최대 값을 결정하고, 그리고 결정된 최소 값 및 결정된 최대 값에 기초하여 파라미터들을 결정하도록 구성될 수도 있다.

[0534] 본 개시물의 다른 예에서, 비디오 디코더 (30) 및/또는 비디오 포스트프로세서 부 (31) 는 파라미터들이 사인되거나 비사인되는지를 나타내는 선택스 엘리먼트를 수신하고, 그리고 SEI 메시지에서의 정보에 대한 파싱 프로세스를 수행하도록 구성될 수도 있고, 파싱 프로세스는 선택스 엘리먼트의 값과 무관하게 동일하다.

[0535] 본 개시물의 특정한 양태들은 예시의 목적으로 HEVC 표준의 확장본들에 대해 설명되었다. 그러나, 본 개시물에서 설명된 기법들은, 아직 개발되지 않은 다른 표준 또는 자유의 비디오 코딩 프로세스들을 포함하는 다른 비디오 코딩 프로세스들에 대해 유용할 수도 있다.

[0536] 비디오 코더는, 본 개시물에서 설명된 바와 같이, 비디오 인코더 또는 비디오 디코더를 지칭할 수도 있다. 마찬가지로, 비디오 코딩 유닛은 비디오 인코더 또는 비디오 디코더를 지칭할 수도 있다. 비슷하게, 비디오 코딩은 해당되는 경우 비디오 인코딩 또는 비디오 디코딩을 지칭할 수도 있다.

[0537] 예에 의존하여, 상이한 시퀀스로 수행될 수도 있는 본 명세서에서 설명된 기법들 중 임의의 기법의 특정 액트들 또는 이벤트들이 추가되거나, 병합되거나, 또는 다 함께 제외될 수도 있다 (예컨대, 모든 설명된 액트들 또는 이벤트들이 그 기법들의 실용화에 필요한 것은 아니다) 는 것이 이해되어야 한다. 더구나, 특정 예들에서,

액트들 또는 이벤트들은 순차적으로라기 보다는, 예컨대, 다중 스레드식 프로세싱, 인터럽트 프로세싱, 또는 다수의 프로세서들을 통하여 동시에 수행될 수도 있다.

[0538] 하나 이상의 예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 그것들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현된다면, 그 기능들은 하나 이상의 명령들 또는 코드로서 컴퓨터 관독가능 매체 상에 저장되거나 또는 그것을 통해 송신될 수도 있고 하드웨어 기반 프로세싱 유닛에 의해 실행될 수도 있다. 컴퓨터 관독가능 매체들은, 데이터 저장 매체들과 같은 유형의 (tangible) 매체에 대응하는 컴퓨터 관독가능 저장 매체들, 또는 예컨대 통신 프로토콜에 따라 한 장소에서 다른 장소로 컴퓨터 프로그램의 전달을 용이하게 하는 임의의 매체를 포함하는 통신 매체들을 포함할 수도 있다. 이런 방식으로, 컴퓨터 관독가능 매체들은 일반적으로 (1) 비일시적인 유형의 컴퓨터 관독가능 저장 매체들 또는 (2) 신호 또는 반송파와 같은 통신 매체에 해당할 수도 있다. 데이터 저장 매체들은 본 개시물에서 설명된 기법들의 구현을 위한 명령들, 코드 및/또는 데이터 구조들을 취출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 이용가능 매체들일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터 관독가능 매체를 포함할 수도 있다.

[0539] 비제한적인 예로, 이러한 컴퓨터 관독가능 저장 매체들은 RAM, ROM, EEPROM, CD-ROM 또는 다른 광 디스크 스토리지, 자기 디스크 스토리지, 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 소망의 프로그램 코드를 컴퓨터에 의해 액세스될 수 있는 명령들 또는 데이터 구조들의 형태로 저장하는데 사용될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 접속이 컴퓨터 관독가능 매체로 적절히 칭해진다. 예를 들어, 명령들이 웹사이트, 서버, 또는 다른 원격 리소스로부터 동축 케이블, 광섬유 케이블, 연선 (twisted pair), 디지털 가입자 회선 (DSL), 또는 무선 기술들 이를테면 적외선, 라디오, 및/또는 마이크로파를 이용하여 송신된다면, 동축 케이블, 광섬유 케이블, 연선, DSL, 또는 적외선, 라디오, 및 마이크로파와 같은 무선 기술은 매체의 정의에 포함된다. 그러나, 컴퓨터 관독가능 저장 매체들 및 데이터 저장 매체들은 커넥션들, 반송파들, 신호들, 또는 다른 일시적 매체들을 포함하지 않지만, 대신 비일시적, 유형의 저장 매체들을 지향하고 있음이 이해되어야 한다. 디스크 (disk 및 disc) 는 본원에서 사용되는 바와 같이, 콤팩트 디스크 (compact disc, CD), 레이저 디스크, 광 디스크, 디지털 다용도 디스크 (DVD), 플로피 디스크 (floppy disk) 및 블루레이 디스크를 포함하는데, disk들은 보통 데이터를 자기적으로 재생하지만, disc들은 레이저들로써 광적으로 데이터를 재생한다. 상기한 것들의 조합들은 컴퓨터 관독가능 매체들의 범위 내에 또한 포함되어야 한다.

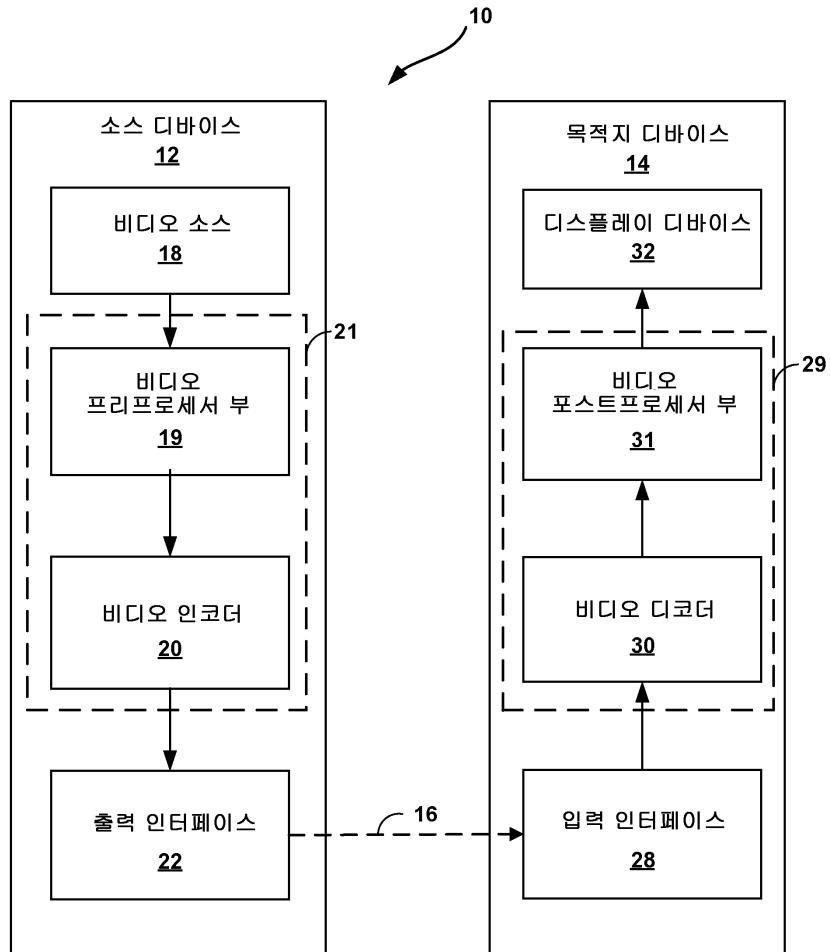
[0540] 명령들은 하나 이상의 프로세서들, 이를테면 하나 이상의 디지털 신호 프로세서들 (DSP들), 범용 마이크로프로세서들, 주문형 집적회로들 (ASIC들), 필드 프로그램가능 로직 어레이들 (FPGA들), 또는 다른 동등한 집적 또는 개별 로직 회로에 의해 실행될 수도 있다. 따라서, 본 명세서에서 사용되는 바와 같은 "프로세서"라는 용어는 앞서의 구조 또는 본 명세서에서 설명된 기법들의 구현에 적합한 임의의 다른 구조 중 임의의 것을 나타낼 수도 있다. 덧붙여서, 일부 양태들에서, 본 명세서에서 설명된 기능성은 인코딩 및 디코딩을 위해 구성되는, 또는 결합형 코덱으로 통합되는 전용 하드웨어 및/또는 소프트웨어 모듈들 내에 제공될 수도 있다. 또한, 본 기법들은 하나 이상의 회로들 또는 로직 엘리먼트들 내에 완전히 구현될 수 있다.

[0541] 본 개시물의 기법들은 무선 핸드셋, 집적회로 (IC) 또는 IC들의 세트 (예컨대, 칩 셋) 를 포함하는 매우 다양한 디바이스들 또는 장치들로 구현될 수도 있다. 다양한 컴포넌트들, 모듈들, 또는 유닛들이 개시된 기법들을 수행하도록 구성되는 디바이스들의 기능적 양태들을 강조하기 위해 본 개시물에서 설명되지만, 상이한 하드웨어 유닛들에 의한 실현을 반드시 요구하지는 않는다. 대신에, 위에서 설명된 바와 같이, 다양한 유닛들은 코덱 하드웨어 유닛에 결합되거나 또는 적합한 소프트웨어 및/또는 펌웨어에 연계하여, 위에서 설명된 바와 같은 하나 이상의 프로세서들을 포함한 상호운용적 하드웨어 유닛들의 컬렉션에 의해 제공될 수도 있다.

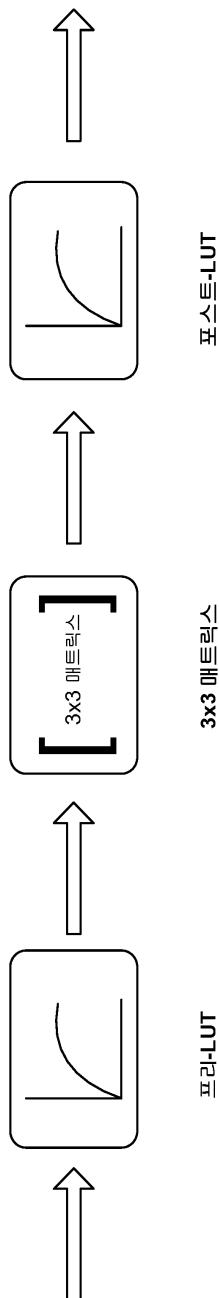
[0542] 다양한 예들이 설명되어 있다. 이들 및 다른 예들은 다음의 청구항들의 범위 내에 있을 수도 있다.

도면

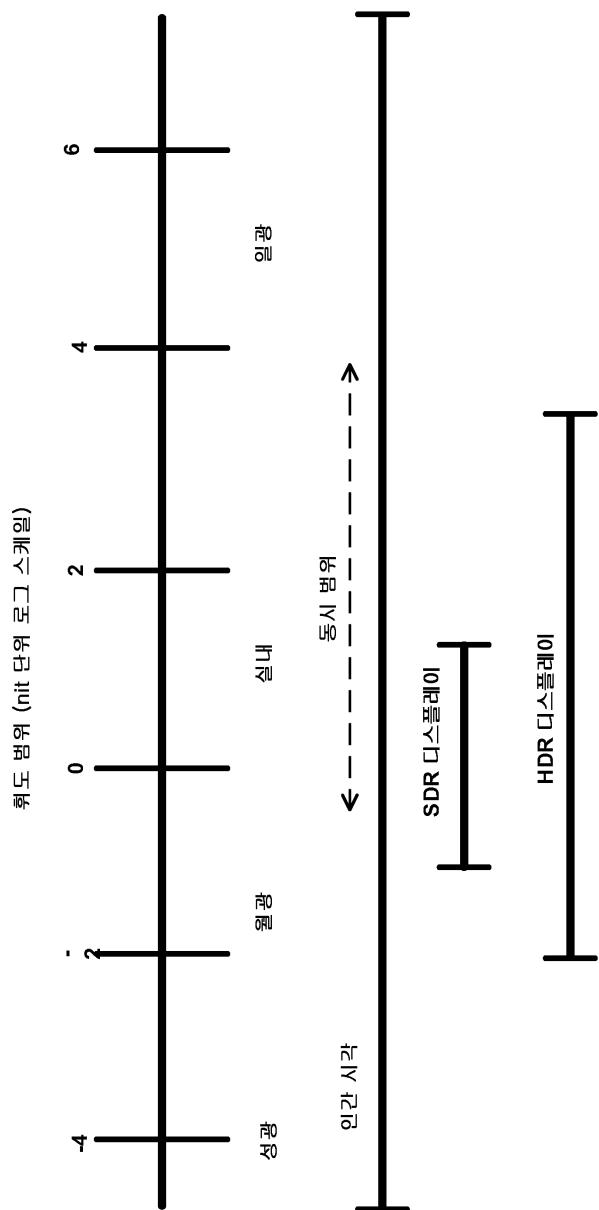
도면1



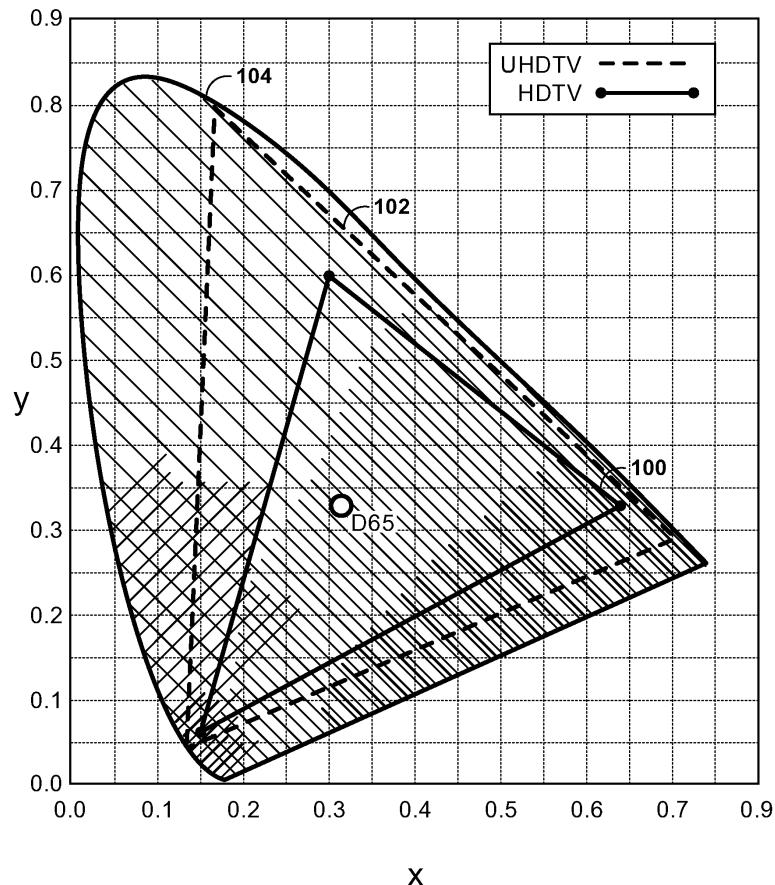
도면2



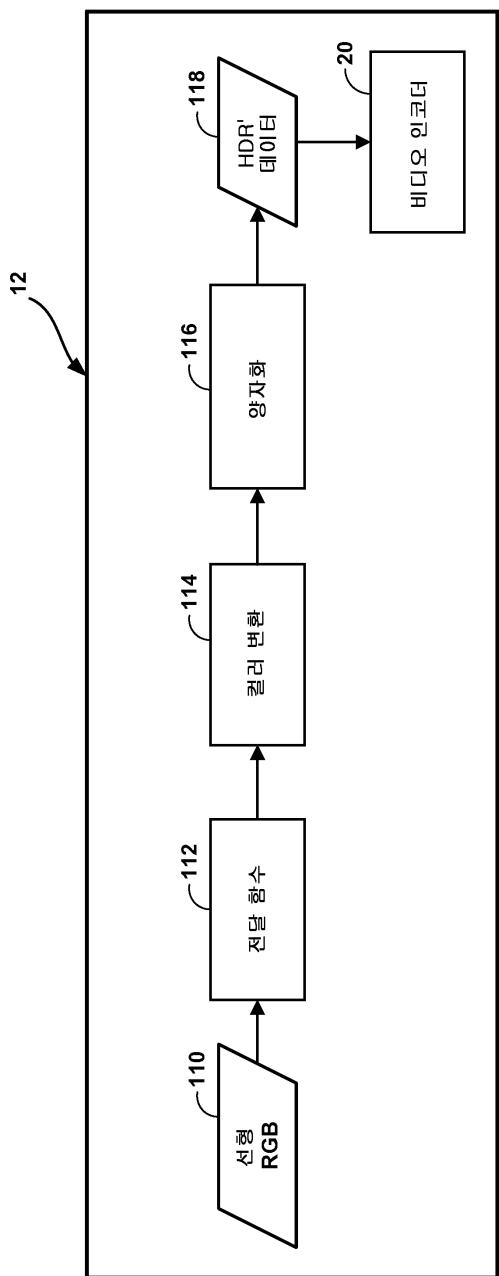
도면3



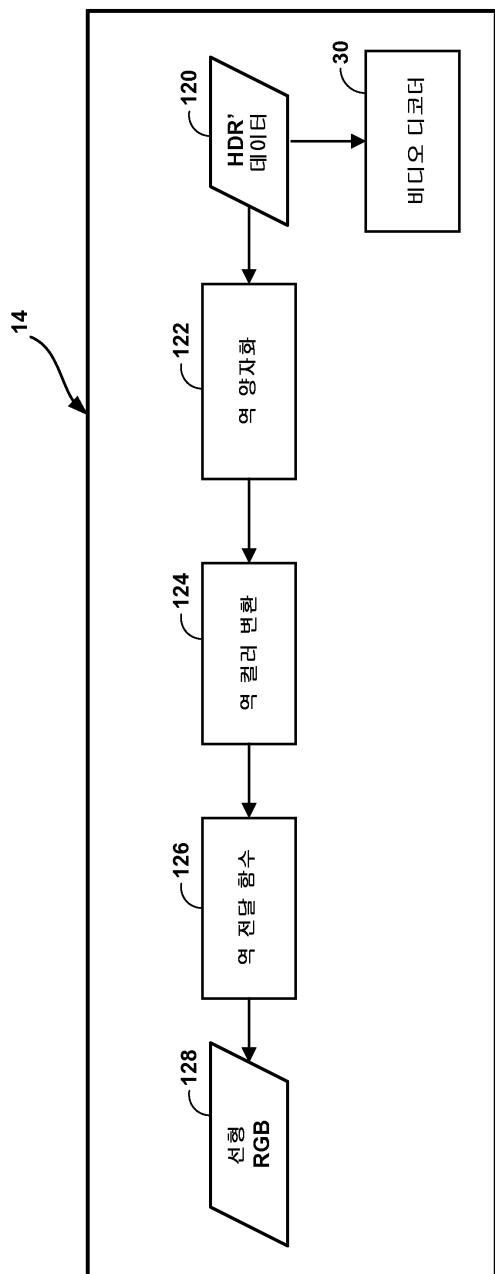
도면4



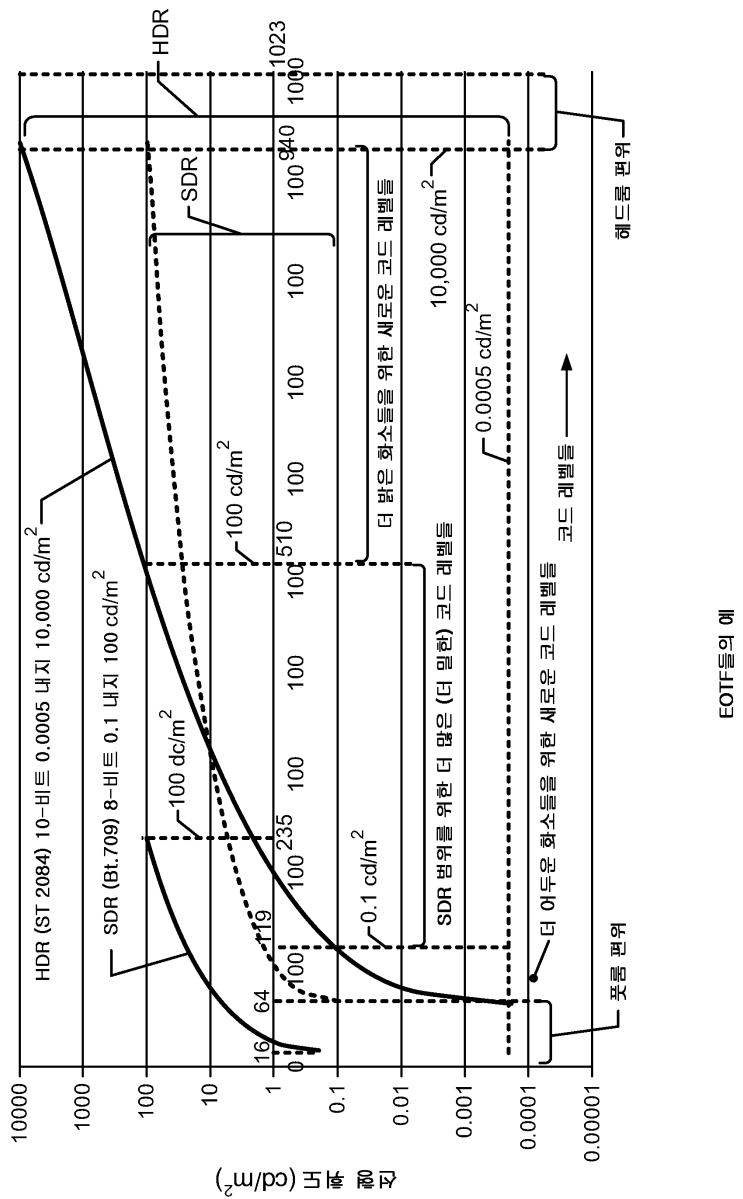
도면5



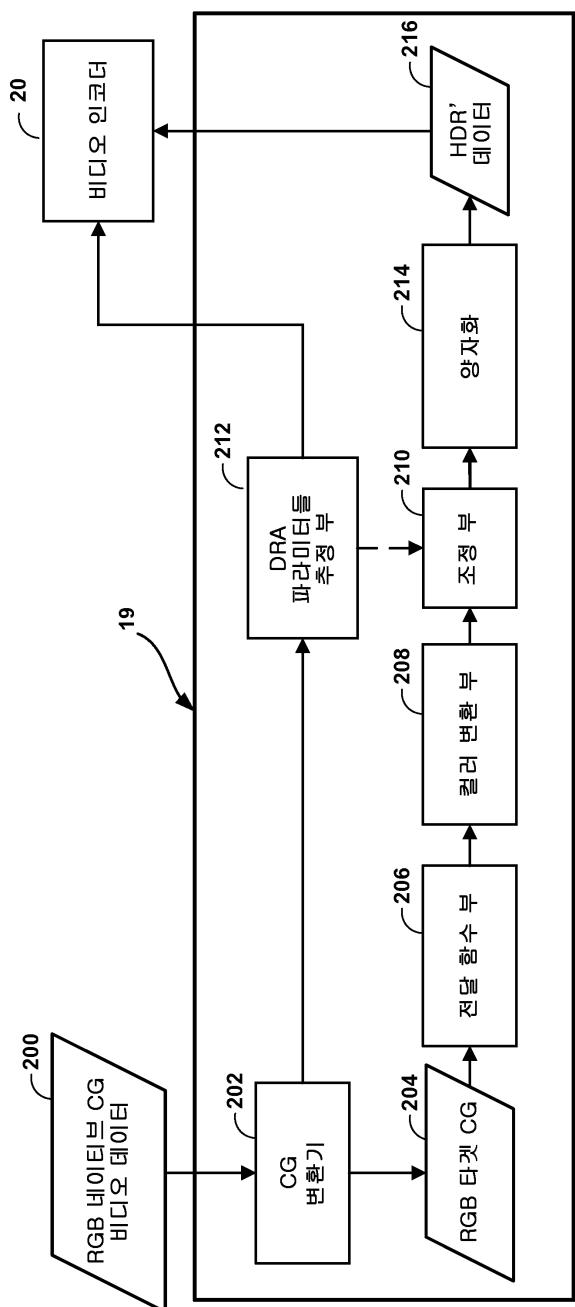
도면6



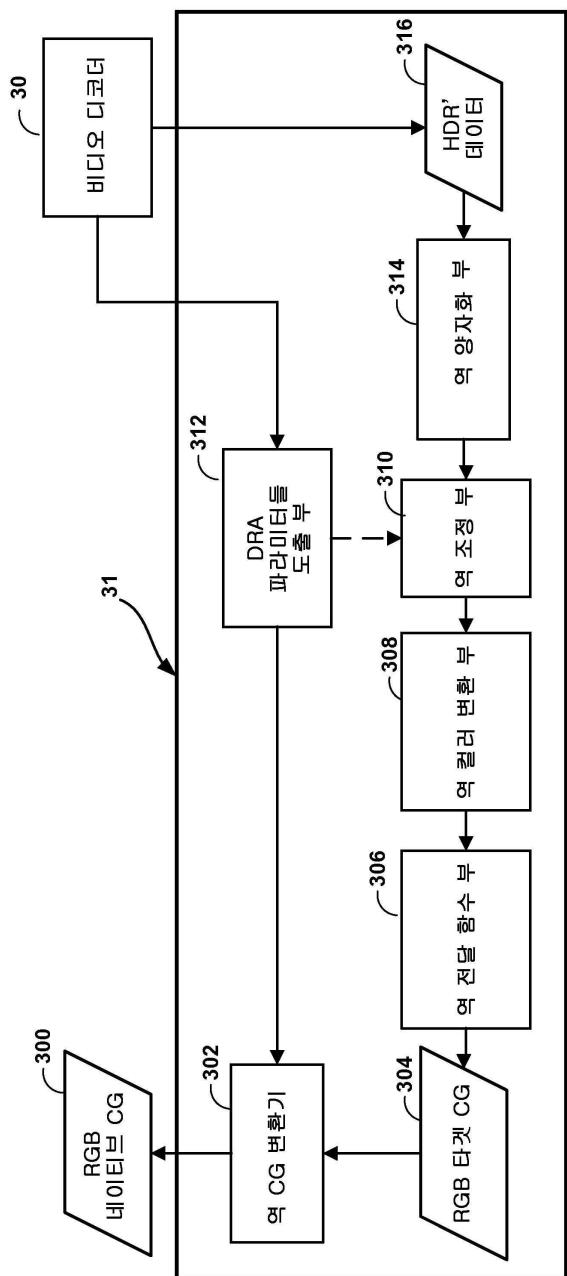
도면7



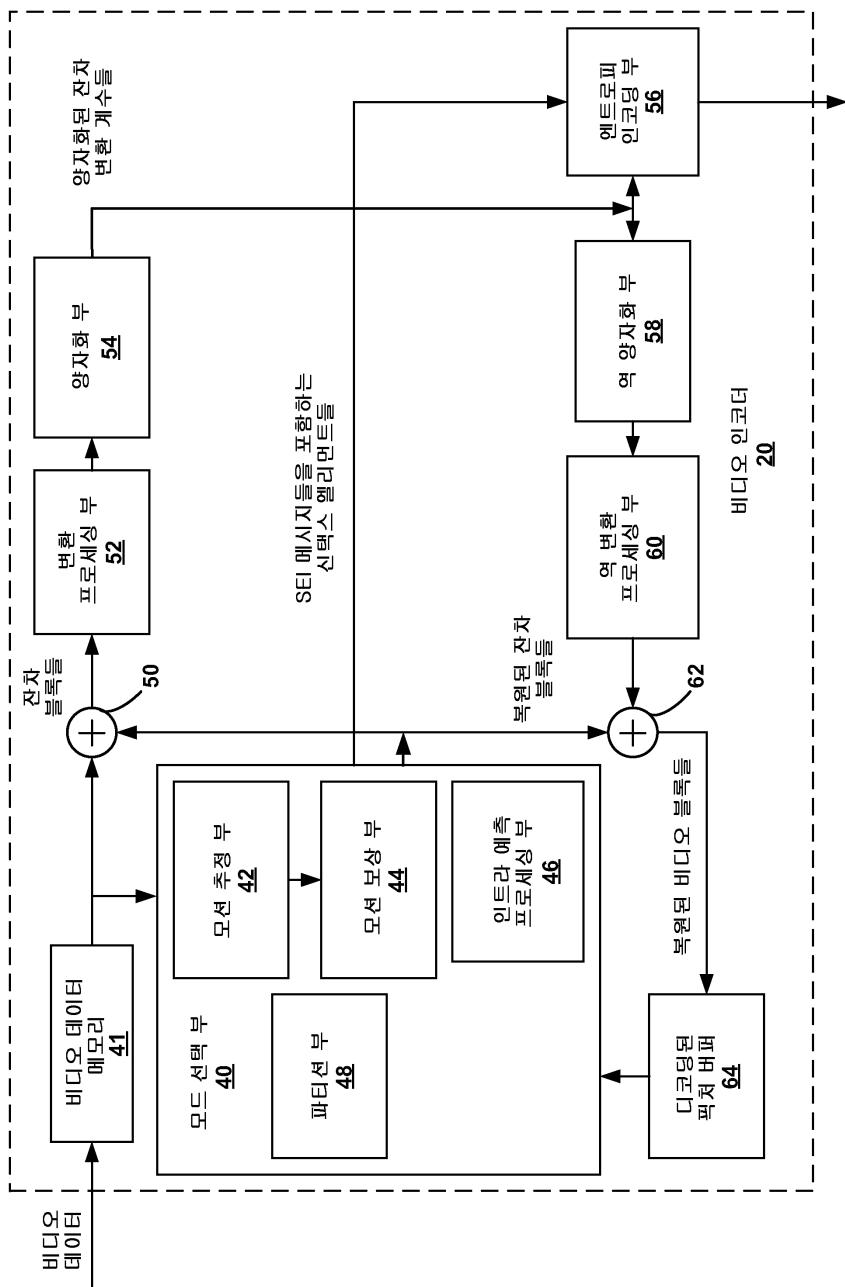
도면8



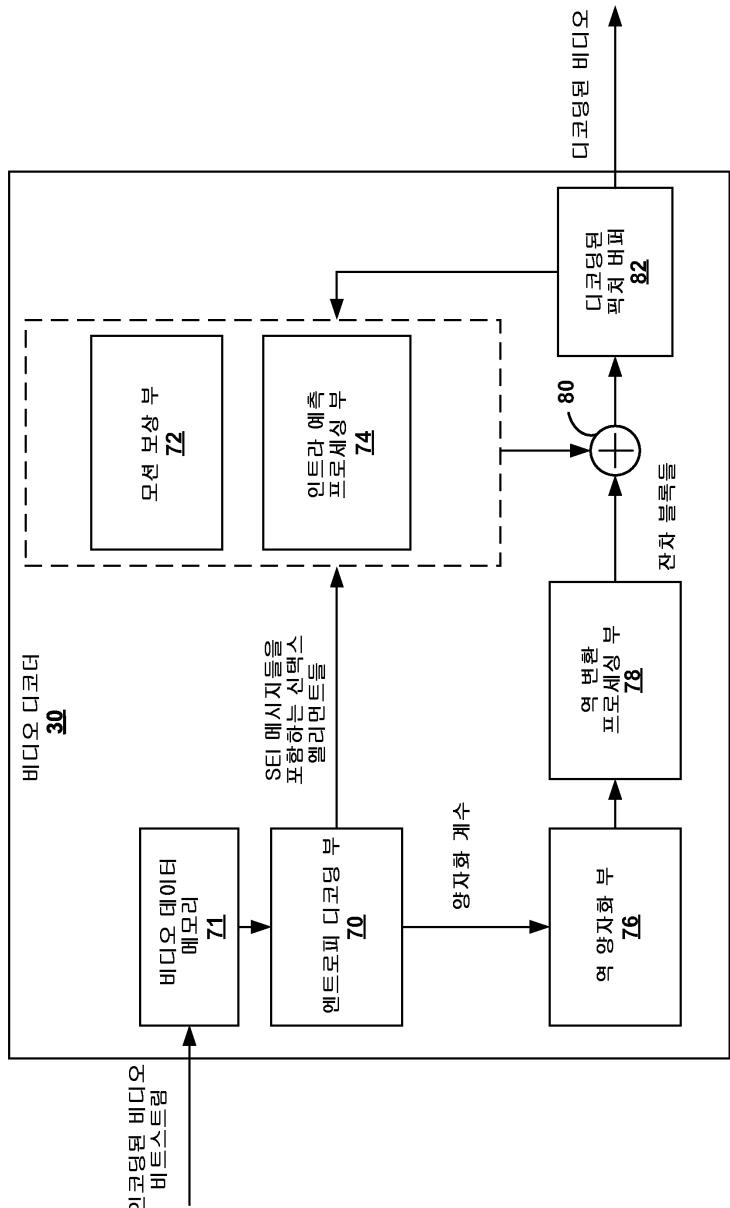
도면9



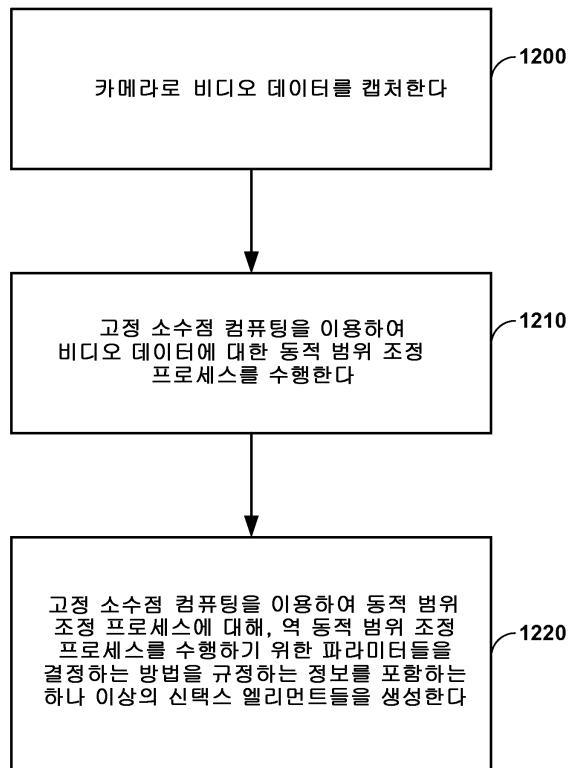
도면10



도면11



도면12



도면13

