



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2019년12월03일

(11) 등록번호 10-2051718

(24) 등록일자 2019년11월27일

(51) 국제특허분류(Int. Cl.)

H04N 19/436 (2014.01) H04N 19/11 (2014.01)

H04N 19/157 (2014.01) H04N 19/44 (2014.01)

H04N 19/50 (2014.01) H04N 19/55 (2014.01)

(52) CPC특허분류

H04N 19/436 (2015.01)

H04N 19/11 (2015.01)

(21) 출원번호 10-2017-7009608

(22) 출원일자(국제) 2015년10월09일

심사청구일자 2019년05월03일

(85) 번역문제출일자 2017년04월07일

(65) 공개번호 10-2017-0067766

(43) 공개일자 2017년06월16일

(86) 국제출원번호 PCT/US2015/054967

(87) 국제공개번호 WO 2016/057938

국제공개일자 2016년04월14일

(30) 우선권주장

62/062,122 2014년10월09일 미국(US)

14/878,825 2015년10월08일 미국(US)

(56) 선행기술조사문헌

Guillaume Laroche, et al., "AHG14: On IBC constraint for Wavefront Parallel Processing", (JCTVC-S0070), JCT-VC of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 19th Meeting: Strasbourg, FR, (2014.10).

(뒷면에 계속)

전체 청구항 수 : 총 28 항

심사관 : 조우연

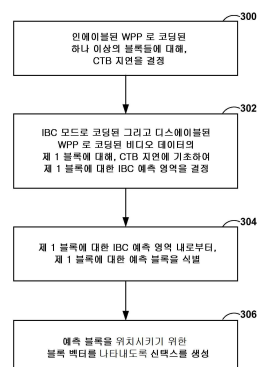
(54) 발명의 명칭 병렬 프로세싱을 위한 인트라 블록 복사 예측 제한들

(57) 요약

비디오 데이터는, 인에이블된 파면 병렬 프로세싱으로 코딩된 하나 이상의 블록들에 대해, 코딩 트리 블록 (CTB) 지연을 결정하는 것으로서, CTB 지연은 CTB들의 제 1 로우가 디코딩되기 시작하는 때와 CTB들의 제 1 로우 아래의 CTB들의 제 2 로우가 디코딩되기 시작하는 때 간의 지연을 식별하는, 상기 CTB 지연을 결정하고; 인트라-블록

(뒷면에 계속)

대표도 - 도17



복사 (IBC) 모드로 코딩되고 디스에이블된 화면 병렬 프로세싱으로 코딩된 비디오 데이터의 현재 블록에 대해, 인에이블된 화면 병렬 프로세싱으로 코딩된 하나 이상의 블록들에 대해 결정되었던 CTB 지연에 기초하여 현재 블록을 포함하는 픽처 내에서 현재 블록에 대한 IBC 예측 영역을 결정하고; 현재 블록에 대한 결정된 IBC 예측 영역 내로부터, 현재 블록에 대한 예측 블록을 식별하며; 예측 블록에 기초하여 현재 블록을 IBC 디코딩하도록 구성될 수도 있다.

(52) CPC특허분류

H04N 19/157 (2015.01)

H04N 19/44 (2015.01)

H04N 19/50 (2015.01)

H04N 19/55 (2015.01)

H04N 19/593 (2015.01)

(72) 발명자

세레긴 바딤

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

카르체비츠 마르타

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

(56) 선행기술조사문헌

Minhua Zhou, et al., “Non-SCCE1: Additional results on intra block copy (IBC)”, (JCTVC-R0208), JCT-VC of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 18th Meeting: Sapporo, JP, (2014.06.30.)

C. Gisquet, et al., “Non-SCCE1: IBC BV throughput issue”, (JCTVC-R0089), JCT-VC of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 18th Meeting: Sapporo, JP, (2014.06.30.)

Krishna Rapaka, et al., “On Parallel Processing capability of intra block copy”, (JCTVC-S0220), JCT-VC of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 19th Meeting: Strasbourg, FR, (2014.10.17.)

명세서

청구범위

청구항 1

비디오 데이터를 디코딩하는 방법으로서,

제 2 픽처에서 비디오 데이터의 제 2 블록이 인트라 블록 복사 (IBC) 모드로 코딩된다는 것을 결정하는 단계로서, 상기 비디오 데이터의 제 2 블록은 코딩 트리 블록들 (CTB들) 의 제 1 로우에 있는, 상기 인트라 블록 복사 (IBC) 모드로 코딩된다는 것을 결정하는 단계;

상기 비디오 데이터의 제 2 블록이 인에이블된 파면 병렬 프로세싱으로 코딩된다는 것을 결정하는 단계;

상기 비디오 데이터의 제 2 블록에 대한 CTB 지연을 사용하여 상기 제 2 블록에 대한 IBC 예측 영역의 형상을 결정하는 단계로서, 상기 CTB 지연은 상기 CTB들의 제 1 로우가 디코딩되기 시작하는 때와 상기 CTB들의 제 1 로우 위의 CTB들의 제 2 로우가 디코딩되기 시작하는 때 간의 지연을 특정하는, 상기 IBC 예측 영역의 형상을 결정하는 단계;

상기 제 2 블록에 대한 상기 IBC 예측 영역 내로부터, 상기 제 2 블록에 대한 예측 블록을 식별하는 단계;

상기 제 2 블록에 대한 상기 예측 블록에 기초하여 상기 제 2 블록을 IBC 디코딩하는 단계;

제 1 픽처에서 비디오 데이터의 제 1 블록이 상기 IBC 모드로 코딩된다는 것을 결정하는 단계;

상기 비디오 데이터의 제 1 블록이 디스에이블된 파면 병렬 프로세싱으로 코딩된다는 것을 결정하는 단계;

상기 제 2 블록에 대하여 사용되었던 상기 CTB 지연에 기초하여 상기 제 1 블록에 대한 IBC 예측 영역의 형상을 결정하는 단계;

상기 제 1 블록에 대한 결정된 상기 IBC 예측 영역 내로부터, 상기 제 1 블록에 대한 예측 블록을 식별하는 단계; 및

상기 예측 블록에 기초하여 상기 제 1 블록을 IBC 디코딩하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 2

제 1 항에 있어서,

신택스 엘리먼트를 수신하는 단계; 및

상기 신택스 엘리먼트의 값에 기초하여 파면 병렬 프로세싱이 상기 제 1 블록에 대해 디스에이블된다는 것을 결정하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 3

제 2 항에 있어서,

상기 신택스 엘리먼트는, 콘텍스트 변수들에 대한 특정 동기화 프로세스가 인보크될 것인지를 나타내는 동기화 프로세스 인에이블링 신택스 엘리먼트를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 4

제 1 항에 있어서,

상기 제 1 블록에 대한 상기 IBC 예측 영역은 이전에 디코딩된 필터링되지 않은 CTB들을 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 5

제 1 항에 있어서,

상기 IBC 예측 영역은 상기 제 1 블록의 우측 및 상기 제 1 블록 위의 적어도 2 이상의 로우들에 위치한, 대각선으로 위치한 CTB 를 포함하고, 상기 대각선으로 위치한 CTB 바로 아래의 CTB 를 제외시키는, 비디오 데이터를 디코딩하는 방법.

청구항 6

제 5 항에 있어서,

상기 대각선으로 위치한 CTB 바로 아래의 상기 CTB 를 상기 제 1 블록과 병렬로 디코딩하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 7

제 1 항에 있어서,

상기 CTB 지연은 하나의 CTB 의 지연을 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 8

제 1 항에 있어서,

비디오 데이터의 인코딩된 비트스트림에서, 상기 비디오 데이터의 제 1 블록에 대한 코딩 모드가 상기 IBC 모드라는 것을 나타내는 하나 이상의 선택스 엘리먼트들을 수신하는 단계;

상기 비디오 데이터의 인코딩된 비트스트림에서, 상기 비디오 데이터의 제 1 블록에 대한 블록 벡터를 식별하는 하나 이상의 선택스 엘리먼트들을 수신하는 단계를 더 포함하고,

상기 제 1 블록에 대한 상기 IBC 예측 영역 내로부터, 상기 제 1 블록에 대한 예측 블록을 식별하는 단계는 상기 블록 벡터를 이용하여 상기 예측 블록을 위치시키는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 9

제 1 항에 있어서,

상기 CTB 지연을 결정하는 단계는 상기 CTB 지연을 CTB들의 유닛들로 결정하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

청구항 10

비디오 데이터를 인코딩하는 방법으로서,

제 2 픽처에서 비디오 데이터의 제 2 블록이 인트라 블록 복사 (IBC) 모드로 코딩된다는 것을 결정하는 단계로서, 상기 비디오 데이터의 제 2 블록은 코딩 트리 블록들 (CTB들) 의 제 1 로우에 있는, 상기 인트라 블록 복사 (IBC) 모드로 코딩된다는 것을 결정하는 단계;

상기 비디오 데이터의 제 2 블록이 인에이블된 파면 병렬 프로세싱으로 코딩된다는 것을 결정하는 단계;

상기 비디오 데이터의 제 2 블록에 대한 CTB 지연을 사용하여 상기 제 2 블록에 대한 IBC 예측 영역의 형상을 결정하는 단계로서, 상기 CTB 지연은 상기 CTB들의 제 1 로우가 디코딩되기 시작하는 때와 상기 CTB들의 제 1 로우 위의 CTB들의 제 2 로우가 디코딩되기 시작하는 때 간의 지연을 특정하는, 상기 IBC 예측 영역의 형상을 결정하는 단계;

상기 제 2 블록에 대한 상기 IBC 예측 영역 내로부터, 상기 제 2 블록에 대한 제 2 예측 블록을 식별하는 단계;

상기 제 2 예측 블록을 로케이팅하기 위한 제 2 블록 벡터를 표시하기 위해 제 2 선택스를 생성하는 단계;

제 1 픽처에서 비디오 데이터의 제 1 블록이 상기 IBC 모드로 코딩된다는 것을 결정하는 단계;

상기 비디오 데이터의 제 1 블록이 디스에이블된 파면 병렬 프로세싱으로 코딩된다는 것을 결정하는 단계;

상기 제 2 블록에 대하여 사용되었던 상기 CTB 지연에 기초하여 상기 제 1 블록에 대한 IBC 예측 영역의 형상을

결정하는 단계;

상기 제 1 블록에 대한 결정된 상기 IBC 예측 영역 내로부터, 상기 제 1 블록에 대한 제 1 예측 블록을 식별하는 단계; 및

상기 제 1 예측 블록을 로케이팅하기 위한 제 1 블록 벡터를 표시하기 위한 제 1 신텍스를 생성하는 단계를 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 11

제 10 항에 있어서,

상기 IBC 예측 영역은 상기 제 1 블록의 우측 및 상기 제 1 블록 위의 적어도 2 이상의 로우들에 CTB 를 포함하고, 상기 제 1 블록의 우측 및 상기 제 1 블록 위의 적어도 2 이상의 로우들에 상기 CTB 바로 아래의 CTB 를 제외시키는, 비디오 데이터를 인코딩하는 방법.

청구항 12

제 10 항에 있어서,

상기 CTB 지연은 2 CTB 지연을 포함하는, 비디오 데이터를 인코딩하는 방법.

청구항 13

비디오 디코딩을 수행하기 위한 디바이스로서,

비디오 데이터를 저장하기 위한 메모리; 및

하나 이상의 프로세서들을 포함하며,

상기 하나 이상의 프로세서들은,

제 2 픽처에서 비디오 데이터의 제 2 블록이 인트라 블록 복사 (IBC) 모드로 코딩된다는 것을 결정하는 것으로서, 상기 비디오 데이터의 제 2 블록은 코딩 트리 블록들 (CTB들) 의 제 1 로우에 있는, 상기 인트라 블록 복사 (IBC) 모드로 코딩된다는 것을 결정하고;

상기 비디오 데이터의 제 2 블록이 인에이블된 파면 병렬 프로세싱으로 코딩된다는 것을 결정하고;

상기 비디오 데이터의 제 2 블록에 대한 CTB 지연을 사용하여 상기 제 2 블록에 대한 IBC 예측 영역의 형상을 결정하는 것으로서, 상기 CTB 지연은 상기 CTB들의 제 1 로우가 디코딩되기 시작하는 때와 상기 CTB들의 제 1 로우 위의 CTB들의 제 2 로우가 디코딩되기 시작하는 때 간의 지연을 특징하는, 상기 IBC 예측 영역의 형상을 결정하고;

상기 제 2 블록에 대한 상기 IBC 예측 영역 내로부터, 상기 제 2 블록에 대한 예측 블록을 식별하고;

상기 제 2 블록에 대한 상기 예측 블록에 기초하여 상기 제 2 블록을 IBC 디코딩하고;

제 1 픽처에서 비디오 데이터의 제 1 블록이 상기 IBC 모드로 코딩된다는 것을 결정하고;

상기 비디오 데이터의 제 1 블록이 디스에이블된 파면 병렬 프로세싱으로 코딩된다는 것을 결정하고;

상기 제 2 블록에 대하여 사용되었던 상기 CTB 지연에 기초하여 상기 제 1 블록에 대한 IBC 예측 영역의 형상을 결정하고;

상기 제 1 블록에 대한 결정된 상기 IBC 예측 영역 내로부터, 상기 제 1 블록에 대한 예측 블록을 식별하며; 그리고

상기 예측 블록에 기초하여 상기 제 1 블록을 IBC 디코딩하도록

구성되는, 비디오 디코딩을 수행하기 위한 디바이스.

청구항 14

제 13 항에 있어서,

상기 하나 이상의 프로세서들은 또한,

신택스 엘리먼트를 수신하고; 그리고

상기 신택스 엘리먼트의 값에 기초하여 화면 병렬 프로세싱이 상기 제 1 블록에 대하여 디스에이블된다는 것을 결정하도록

구성되는, 비디오 디코딩을 수행하기 위한 디바이스.

청구항 15

제 14 항에 있어서,

상기 신택스 엘리먼트는, 콘텍스트 변수들에 대한 특정 동기화 프로세스가 인보크될 것인지를 나타내는 동기화 프로세스 인에이블링 신택스 엘리먼트를 포함하는, 비디오 디코딩을 수행하기 위한 디바이스.

청구항 16

제 13 항에 있어서,

상기 제 1 블록에 대한 상기 IBC 예측 영역은 이전에 디코딩된 필터링되지 않은 CTB들을 포함하는, 비디오 디코딩을 수행하기 위한 디바이스.

청구항 17

제 13 항에 있어서,

상기 IBC 예측 영역은 상기 제 1 블록의 우측 및 상기 제 1 블록 위의 적어도 2 이상의 로우들에 위치한, 대각선으로 위치한 CTB 를 포함하고, 상기 대각선으로 위치한 CTB 바로 아래의 CTB 를 제외시키는, 비디오 디코딩을 수행하기 위한 디바이스.

청구항 18

제 17 항에 있어서,

상기 하나 이상의 프로세서들은 또한,

상기 대각선으로 위치한 CTB 바로 아래의 상기 CTB 를 상기 제 1 블록과 병렬로 디코딩하도록 구성되는, 비디오 디코딩을 수행하기 위한 디바이스.

청구항 19

제 13 항에 있어서,

상기 CTB 지연은 하나의 CTB 지연을 포함하는, 비디오 디코딩을 수행하기 위한 디바이스.

청구항 20

제 13 항에 있어서,

상기 하나 이상의 프로세서들은 또한,

비디오 데이터의 인코딩된 비트스트림에서, 상기 비디오 데이터의 제 1 블록에 대한 코딩 모드가 상기 IBC 모드라는 것을 나타내는 하나 이상의 신택스 엘리먼트들을 수신하고; 그리고

상기 비디오 데이터의 인코딩된 비트스트림에서, 상기 비디오 데이터의 제 1 블록에 대한 블록 벡터를 식별하는 하나 이상의 신택스 엘리먼트들을 수신하도록

구성되고,

상기 제 1 블록에 대한 상기 IBC 예측 영역 내로부터, 상기 제 1 블록에 대한 예측 블록을 식별하는 것은 상기 블록 벡터를 이용하여 상기 예측 블록을 위치시키는 것을 포함하는, 비디오 디코딩을 수행하기 위한 디바이스.

청구항 21

제 13 항에 있어서,

상기 CTB 지연을 결정하기 위해, 상기 하나 이상의 프로세서들은 또한, 상기 CTB 지연을 CTB들의 유닛들로 결정하도록 구성되는, 비디오 디코딩을 수행하기 위한 디바이스.

청구항 22

제 13 항에 있어서,

상기 비디오 디코딩을 수행하기 위한 디바이스는,

집적 회로;

마이크로프로세서; 또는

디스플레이를 포함하는 무선 통신 디바이스 중 적어도 하나를 포함하는, 비디오 디코딩을 수행하기 위한 디바이스.

청구항 23

비디오 인코딩을 수행하기 위한 디바이스로서,

비디오 데이터를 저장하기 위한 메모리; 및

하나 이상의 프로세서들을 포함하며,

상기 하나 이상의 프로세서들은,

제 2 픽처에서 비디오 데이터의 제 2 블록이 인트라 블록 복사 (IBC) 모드로 코딩된다는 것을 결정하는 것으로서, 상기 비디오 데이터의 제 2 블록은 코딩 트리 블록들 (CTB들) 의 제 1 로우에 있는, 상기 인트라 블록 복사 (IBC) 모드로 코딩된다는 것을 결정하고;

상기 비디오 데이터의 제 2 블록이 인에이블된 파면 병렬 프로세싱으로 코딩된다는 것을 결정하고;

상기 비디오 데이터의 제 2 블록에 대한 CTB 지연을 사용하여 상기 제 2 블록에 대한 IBC 예측 영역의 형상을 결정하는 것으로서, 상기 CTB 지연은 상기 CTB들의 제 1 로우가 디코딩되기 시작하는 때와 상기 CTB들의 제 1 로우 위의 CTB들의 제 2 로우가 디코딩되기 시작하는 때 간의 지연을 특정하는, 상기 IBC 예측 영역의 형상을 결정하고;

상기 제 2 블록에 대한 상기 IBC 예측 영역 내로부터, 상기 제 2 블록에 대한 제 2 예측 블록을 식별하고;

상기 제 2 예측 블록을 로케이팅하기 위한 제 2 블록 벡터를 표시하기 위해 제 2 선택스를 생성하고;

제 1 픽처에서 비디오 데이터의 제 1 블록이 상기 IBC 모드로 코딩된다는 것을 결정하고;

상기 비디오 데이터의 제 1 블록이 디스에이블된 파면 병렬 프로세싱으로 코딩된다는 것을 결정하고;

상기 제 2 블록에 대하여 사용되었던 상기 CTB 지연에 기초하여 상기 제 1 블록에 대한 IBC 예측 영역에 대한 형상을 결정하고;

상기 제 1 블록에 대한 결정된 상기 IBC 예측 영역 내로부터, 상기 제 1 블록에 대한 제 1 예측 블록을 식별하며; 그리고

상기 제 1 예측 블록을 로케이팅하기 위한 제 1 블록 벡터를 표시하기 위한 제 1 선택스를 생성하도록 구성되는, 비디오 인코딩을 수행하기 위한 디바이스.

청구항 24

제 23 항에 있어서,

상기 IBC 예측 영역은 상기 제 1 블록의 우측 및 상기 제 1 블록 위의 적어도 2 이상의 로우들에 CTB 를 포함하고, 상기 제 1 블록의 우측 및 상기 제 1 블록 위의 적어도 2 이상의 로우들에 상기 CTB 바로 아래의 CTB 를 제외시키는, 비디오 인코딩을 수행하기 위한 디바이스.

청구항 25

제 23 항에 있어서,

상기 CTB 지연은 하나의 CTB 지연을 포함하는, 비디오 인코딩을 수행하기 위한 디바이스.

청구항 26

제 23 항에 있어서,

상기 비디오 인코딩을 수행하기 위한 디바이스는,

집적 회로;

마이크로프로세서; 또는

카메라를 포함하는 무선 통신 디바이스 중 적어도 하나를 포함하는, 비디오 인코딩을 수행하기 위한 디바이스.

청구항 27

비디오 데이터를 디코딩하기 위한 장치로서,

제 2 픽처에서 비디오 데이터의 제 2 블록이 인트라 블록 복사 (IBC) 모드로 코딩된다는 것을 결정하는 수단으로서, 상기 비디오 데이터의 제 2 블록은 코딩 트리 블록들 (CTB들) 의 제 1 로우에 있는, 상기 인트라 블록 복사 (IBC) 모드로 코딩된다는 것을 결정하는 수단;

상기 비디오 데이터의 제 2 블록이 인에이블된 파면 병렬 프로세싱으로 코딩된다는 것을 결정하는 수단;

상기 비디오 데이터의 제 2 블록에 대한 CTB 지연을 사용하여 상기 제 2 블록에 대한 IBC 예측 영역의 형상을 결정하는 수단으로서, 상기 CTB 지연은 상기 CTB들의 제 1 로우가 디코딩되기 시작하는 때와 상기 CTB들의 제 1 로우 위의 CTB들의 제 2 로우가 디코딩되기 시작하는 때 간의 지연을 특정하는, 상기 IBC 예측 영역의 형상을 결정하는 수단;

상기 제 2 블록에 대한 상기 IBC 예측 영역 내로부터, 상기 제 2 블록에 대한 예측 블록을 식별하는 수단;

상기 제 2 블록에 대한 상기 예측 블록에 기초하여 상기 제 2 블록을 IBC 디코딩하는 수단;

제 1 픽처에서 비디오 데이터의 제 1 블록이 상기 IBC 모드로 코딩된다는 것을 결정하는 수단;

상기 비디오 데이터의 제 1 블록이 디스에이블된 파면 병렬 프로세싱으로 코딩된다는 것을 결정하는 수단;

상기 제 2 블록에 대하여 사용되었던 상기 CTB 지연에 기초하여 상기 제 1 블록에 대한 IBC 예측 영역의 형상을 결정하는 수단;

상기 제 1 블록에 대한 결정된 상기 IBC 예측 영역 내로부터, 상기 제 1 블록에 대한 예측 블록을 식별하는 수단; 및

상기 예측 블록에 기초하여 상기 제 1 블록을 IBC 디코딩하는 수단을 포함하는, 비디오 데이터를 디코딩하기 위한 장치.

청구항 28

명령들을 저장한 비일시적 컴퓨터 판독가능 저장 매체로서,

상기 명령들은, 하나 이상의 프로세서들에 의해 실행되는 경우, 상기 하나 이상의 프로세서들로 하여금,

제 2 픽처에서 비디오 데이터의 제 2 블록이 인트라 블록 복사 (IBC) 모드로 코딩된다는 것을 결정하게 하는 것으로서, 상기 비디오 데이터의 제 2 블록은 코딩 트리 블록들 (CTB들) 의 제 1 로우에 있는, 상기 인트라 블록 복사 (IBC) 모드로 코딩된다는 것을 결정하게 하고;

상기 비디오 데이터의 제 2 블록이 인에이블된 파면 병렬 프로세싱으로 코딩된다는 것을 결정하게 하고;

상기 비디오 데이터의 제 2 블록에 대한 CTB 지연을 사용하여 상기 제 2 블록에 대한 IBC 예측 영역의 형상을 결정하게 하는 것으로서, 상기 CTB 지연은 상기 CTB들의 제 1 로우가 디코딩되기 시작하는 때와 상기 CTB들의 제 1 로우 위의 CTB들의 제 2 로우가 디코딩되기 시작하는 때 간의 지연을 특정하는, 상기 IBC 예측 영역의 형

상을 결정하게 하고;

상기 제 2 블록에 대한 상기 IBC 예측 영역 내로부터, 상기 제 2 블록에 대한 예측 블록을 식별하게 하고;

상기 제 2 블록에 대한 상기 예측 블록에 기초하여 상기 제 2 블록을 IBC 디코딩하게 하고;

제 1 픽처에서 비디오 데이터의 제 1 블록이 상기 IBC 모드로 코딩된다는 것을 결정하게 하고;

상기 비디오 데이터의 제 1 블록이 디스에이블된 파면 병렬 프로세싱으로 코딩된다는 것을 결정하게 하고;

상기 제 2 블록에 대하여 사용되었던 상기 CTB 지연에 기초하여 상기 제 1 블록에 대한 IBC 예측 영역의 형상을 결정하게 하고;

상기 제 1 블록에 대한 결정된 상기 IBC 예측 영역 내로부터, 상기 제 1 블록에 대한 예측 블록을 식별하게 하며; 그리고

상기 예측 블록에 기초하여 상기 제 1 블록을 IBC 디코딩하게 하는, 비일시적 컴퓨터 판독가능 저장 매체.

청구항 29

삭제

청구항 30

삭제

청구항 31

삭제

청구항 32

삭제

청구항 33

삭제

청구항 34

삭제

청구항 35

삭제

청구항 36

삭제

발명의 설명

기술 분야

[0001] 본 출원은 2014 년 10 월 9 일에 출원된 미국 가특허출원 제 62/062,122 호의 우선권을 주장하며, 이것의 전체 내용이 본원에 참조로서 포함된다.

[0002] 기술 분야

[0003] 본 개시물은 비디오 코딩, 및 보다 구체적으로는 다른 비디오 블록들에 기초한 비디오 블록들의 예측에 관한 것이다.

배경 기술

[0004] 디지털 비디오 능력들은 디지털 텔레비전들, 디지털 직접 방송 시스템들, 무선 방송 시스템들, PDA (personal

digital assistant) 들, 랩톱이나 데스크톱 컴퓨터들, 태블릿 컴퓨터들, 전자책 리더들, 디지털 카메라들, 디지털 레코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게임 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 무선 전화들, 소위 "스마트 폰들", 비디오 원격화상회의 디바이스들, 비디오 스트리밍 디바이스들 등을 포함하는 광범위한 디바이스들에 포함될 수 있다. 디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, 파트 10, 어드밴스드 비디오 코딩 (AVC), 고 효율성 비디오 코딩 (HEVC) 표준 (H.265) 에 의해 정의된 표준들, 및 이러한 표준들의 확장들에서 설명된 바와 같은 비디오 압축 기법들을 구현한다. 비디오 디바이스들은 이러한 비디오 압축 기법들을 구현함으로써, 디지털 비디오 정보를 보다 효율적으로 송신, 수신, 인코딩, 디코딩, 및/또는 저장할 수도 있다.

[0005] 비디오 압축 기법들은 비디오 시퀀스들에 내재하는 리던던시를 감소시키거나 제거하기 위한 공간 (인트라-픽처) 예측 및/또는 시간 (인터-픽처) 예측을 수행한다. 블록 기반 비디오 코딩에 있어서, 비디오 슬라이스 (예들 들어, 비디오 프레임, 또는 비디오 프레임의 일부분) 는 비디오 블록들로 파티셔닝될 수도 있으며, 이 비디오 블록들은 또한 트리블록들, 코딩 유닛 (CU) 들 및/또는 코딩 노드들로 지칭될 수도 있다. 픽처의 인트라-코딩된 (I) 슬라이스 내의 비디오 블록들은 동일한 픽처 내의 이웃하는 블록들 내의 레퍼런스 샘플들에 대한 공간 예측을 사용하여 인코딩된다. 픽처의 인터-코딩된 (P 또는 B) 슬라이스 내의 비디오 블록들은 동일한 픽처 내의 이웃하는 블록들 내의 레퍼런스 샘플들에 대한 공간 예측, 또는 다른 레퍼런스 픽처들 내의 레퍼런스 샘플들에 대한 시간 예측을 사용할 수도 있다. 픽처들은 프레임들로서 지칭될 수도 있고, 레퍼런스 픽처들은 레퍼런스 프레임들로 지칭될 수도 있다.

[0006] 공간 또는 시간 예측은 코딩될 블록에 대한 예측 블록을 초래한다. 잔여 데이터는 코딩될 원래의 블록과 예측 블록 간의 픽셀 차이들을 나타낸다. 인터-코딩된 블록은 예측 블록을 형성하는 레퍼런스 샘플들의 블록을 가리키는 모션 벡터, 및 코딩된 블록과 예측 블록 간의 차이를 나타내는 잔여 데이터에 따라 인코딩된다. 인트라-코딩된 블록은 인트라-코딩 모드와 잔여 데이터에 따라 인코딩된다. 추가의 압축을 위해, 잔여 데이터는 픽셀 도메인에서 변환 도메인으로 변환되어, 잔여 변환 계수들을 초래하고 이 변환 계수들은, 그 후 양자화될 수도 있다. 처음에 2 차원 어레이로 배열된 양자화된 변환 계수들은 스캐닝되어 변환 계수들의 1 차원 벡터를 생성할 수도 있고, 엔트로피 코딩이 적용되어 보다 많은 압축을 달성할 수도 있다.

발명의 내용

해결하려는 과제

과제의 해결 수단

[0007] 본 개시물은, IBC 모드가 인에이블되는 경우 병렬 프로세싱을 잠재적으로 강화시키기 위한 기법들을 도입한다.

[0008] 일 예에서, 비디오 데이터를 디코딩하는 방법은, 인에이블된 화면 병렬 프로세싱으로 코딩된 하나 이상의 블록들에 대해, 코딩 트리 블록 (CTB) 지연을 결정하는 단계로서, CTB 지연은 CTB들의 제 1 로우가 디코딩되기 시작하는 때와 CTB들의 제 1 로우 아래의 CTB들의 제 2 로우가 디코딩되기 시작하는 때 간의 지연을 식별하는, 상기 CTB 지연을 결정하는 단계; 인트라-블록 복사 (IBC) 모드로 코딩되고 디스에이블된 화면 병렬 프로세싱으로 코딩된 비디오 데이터의 현재 블록에 대해, 인에이블된 화면 병렬 프로세싱으로 코딩된 하나 이상의 블록들에 대해 결정되었던 CTB 지연에 기초하여 현재 블록을 포함하는 픽처 내에서 현재 블록에 대한 IBC 예측 영역을 결정하는 단계; 현재 블록에 대한 결정된 IBC 예측 영역 내로부터, 현재 블록에 대한 예측 블록을 식별하는 단계; 및 예측 블록에 기초하여 현재 블록을 IBC 디코딩하는 단계를 포함한다.

[0009] 다른 예에서, 비디오 데이터를 인코딩하는 방법은, 인에이블된 화면 병렬 프로세싱으로 코딩된 하나 이상의 블록들에 대해, 코딩 트리 블록 (CTB) 지연을 결정하는 단계로서, CTB 지연은 CTB들의 제 1 로우가 디코딩되기 시작하는 때와 CTB들의 제 1 로우 아래의 CTB들의 제 2 로우가 디코딩되기 시작하는 때 간의 지연을 식별하는, 상기 CTB 지연을 결정하는 단계; 인트라-블록 복사 (IBC) 모드로 코딩되고 디스에이블된 화면 병렬 프로세싱으로 코딩된 비디오 데이터의 제 1 블록에 대해, CTB 지연에 기초하여 제 1 블록에 대한 IBC 예측 영역을 결정하는 단계; 제 1 블록에 대한 IBC 예측 영역 내로부터, 제 1 블록에 대한 예측 블록을 식별하는 단계; 및 예측 블록을 위치시키기 위한 블록 벡터를 나타내도록 신텍스를 생성하는 단계를 포함한다.

[0010] 다른 예에서, 비디오 코딩을 수행하기 위한 디바이스는, 비디오 데이터를 저장하는 메모리, 및 인에이블된 화면 병렬 프로세싱으로 코딩된 하나 이상의 블록들에 대해, 코딩 트리 블록 (CTB) 지연을 결정하는 것으로서, CTB

지연은 CTB들의 제 1 로우가 디코딩되기 시작하는 때와 CTB들의 제 1 로우 아래의 CTB들의 제 2 로우가 디코딩되기 시작하는 때 간의 지연을 식별하는, 상기 CTB 지연을 결정하고; 인트라-블록 복사 (IBC) 모드로 코딩되고 디스에이블된 파면 병렬 프로세싱으로 코딩된 비디오 데이터의 현재 블록에 대해, 인에이블된 파면 병렬 프로세싱으로 코딩된 하나 이상의 블록들에 대해 결정되었던 CTB 지연에 기초하여 현재 블록을 포함하는 픽처 내에서 현재 블록에 대한 IBC 예측 영역을 결정하고; 현재 블록에 대한 결정된 IBC 예측 영역 내로부터, 현재 블록에 대한 예측 블록을 식별하며; 예측 블록에 기초하여 현재 블록을 IBC 디코딩하도록 구성된 하나 이상의 프로세서들을 포함한다.

[0011] 다른 예에서, 비디오 인코딩을 수행하기 위한 디바이스는, 비디오 데이터를 저장하기 위한 메모리, 및 인에이블된 파면 병렬 프로세싱으로 코딩된 하나 이상의 블록들에 대해 코딩 트리 블록 (CTB) 지연을 결정하는 것으로서, CTB 지연은 CTB들의 제 1 로우가 디코딩되기 시작하는 때와 CTB들의 제 1 로우 아래의 CTB들의 제 2 로우가 디코딩되기 시작하는 때 간의 지연을 식별하는, 상기 CTB 지연을 결정하고; 인트라-블록 복사 (IBC) 모드로 코딩되고 디스에이블된 파면 병렬 프로세싱으로 코딩된 비디오 데이터의 제 1 블록에 대해, CTB 지연에 기초하여 제 1 블록에 대한 IBC 예측 영역을 결정하고; 제 1 블록에 대한 IBC 예측 영역 내로부터, 제 1 블록에 대한 예측 블록을 식별하며; 예측 블록을 위치시키기 위한 블록 벡터를 나타내도록 신택스를 생성하도록 구성된 하나 이상의 프로세서들을 포함한다.

[0012] 다른 예에서, 비디오 데이터를 디코딩하기 위한 장치는, 인에이블된 파면 병렬 프로세싱으로 코딩된 하나 이상의 블록들에 대해, 코딩 트리 블록 (CTB) 지연을 결정하기 위한 수단으로서, CTB 지연은 CTB들의 제 1 로우가 디코딩되기 시작하는 때와 CTB들의 제 1 로우 아래의 CTB들의 제 2 로우가 디코딩되기 시작하는 때 간의 지연을 식별하는, 상기 CTB 지연을 결정하기 위한 수단; 인트라-블록 복사 (IBC) 모드로 코딩되고 디스에이블된 파면 병렬 프로세싱으로 코딩된 비디오 데이터의 현재 블록에 대해, 인에이블된 파면 병렬 프로세싱으로 코딩된 하나 이상의 블록들에 대해 결정되었던 CTB 지연에 기초하여 현재 블록을 포함하는 픽처 내에서 현재 블록에 대한 IBC 예측 영역을 결정하기 위한 수단; 현재 블록에 대한 결정된 IBC 예측 영역 내로부터, 현재 블록에 대한 예측 블록을 식별하기 위한 수단; 및 예측 블록에 기초하여 현재 블록을 IBC 디코딩하기 위한 수단을 포함한다.

[0013] 다른 예에서, 컴퓨터 판독가능 저장 매체는, 하나 이상의 프로세서들에 의해 실행되는 경우, 하나 이상의 프로세서들로 하여금, 인에이블된 파면 병렬 프로세싱으로 코딩된 하나 이상의 블록들에 대해, 코딩 트리 블록 (CTB) 지연을 결정하게 하는 것으로서, CTB 지연은 CTB들의 제 1 로우가 디코딩되기 시작하는 때와 CTB들의 제 1 로우 아래의 CTB들의 제 2 로우가 디코딩되기 시작하는 때 간의 지연을 식별하는, 상기 CTB 지연을 결정하게 하고; 인트라-블록 복사 (IBC) 모드로 코딩되고 디스에이블된 파면 병렬 프로세싱으로 코딩된 비디오 데이터의 현재 블록에 대해, 인에이블된 파면 병렬 프로세싱으로 코딩된 하나 이상의 블록들에 대해 결정되었던 CTB 지연에 기초하여 현재 블록을 포함하는 픽처 내에서 현재 블록에 대한 IBC 예측 영역을 결정하게 하고; 현재 블록에 대한 결정된 IBC 예측 영역 내로부터, 현재 블록에 대한 예측 블록을 식별하게 하며; 예측 블록에 기초하여 현재 블록을 IBC 디코딩하게 하는 명령들을 저장한다.

[0014] 다른 예에서, 비디오 데이터를 인코딩하기 위한 장치는, 인에이블된 파면 병렬 프로세싱으로 코딩된 하나 이상의 블록들에 대해 코딩 트리 블록 (CTB) 지연을 결정하기 위한 수단으로서, CTB 지연은 CTB들의 제 1 로우가 디코딩되기 시작하는 때와 CTB들의 제 1 로우 아래의 CTB들의 제 2 로우가 디코딩되기 시작하는 때 간의 지연을 식별하는, 상기 CTB 지연을 결정하기 위한 수단; 인트라-블록 복사 (IBC) 모드로 코딩되고 디스에이블된 파면 병렬 프로세싱으로 코딩된 비디오 데이터의 제 1 블록에 대해, CTB 지연에 기초하여 제 1 블록에 대한 IBC 예측 영역을 결정하기 위한 수단; 제 1 블록에 대한 IBC 예측 영역 내로부터, 제 1 블록에 대한 예측 블록을 식별하기 위한 수단; 및 예측 블록을 위치시키기 위한 블록 벡터를 나타내도록 신택스를 생성하기 위한 수단을 포함한다.

[0015] 다른 예에서, 컴퓨터 판독가능 저장 매체는, 하나 이상의 프로세서들에 의해 실행되는 경우, 하나 이상의 프로세서들로 하여금, 인에이블된 파면 병렬 프로세싱으로 코딩된 하나 이상의 블록들에 대해 코딩 트리 블록 (CTB) 지연을 결정하게 하는 것으로서, CTB 지연은 CTB들의 제 1 로우가 디코딩되기 시작하는 때와 CTB들의 제 1 로우 아래의 CTB들의 제 2 로우가 디코딩되기 시작하는 때 간의 지연을 식별하는, 상기 CTB 지연을 결정하게 하고; 인트라-블록 복사 (IBC) 모드로 코딩되고 디스에이블된 파면 병렬 프로세싱으로 코딩된 비디오 데이터의 제 1 블록에 대해, CTB 지연에 기초하여 제 1 블록에 대한 IBC 예측 영역을 결정하게 하고; 제 1 블록에 대한 IBC 예측 영역 내로부터, 제 1 블록에 대한 예측 블록을 식별하게 하며; 예측 블록을 위치시키기 위한 블록 벡터를 나타내도록 신택스를 생성하게 하는 명령들을 저장한다.

[0016] 본 기법들의 하나 이상의 양태들의 세부사항들은 첨부한 도면들 및 이하의 상세한 설명에서 기술된다. 본 기법들의 다른 특성들, 목적들 및 이점들은 상세한 설명 및 도면들로부터, 그리고 청구항들로부터 명백해질 것이다.

도면의 간단한 설명

[0017] 도 1 은 본 개시물에서 설명된 기법들을 이용할 수도 있는 예시의 비디오 인코딩 및 디코딩 시스템을 예시하는 블록도이다.

도 2a 내지 도 2c 는 비디오 데이터에 대한 상이한 샘플 포맷들을 예시하는 개념도들이다.

도 3 은 4:2:0 샘플 포맷에 따라 포맷된 16x16 코딩 유닛을 예시하는 개념도이다.

도 4 는 4:2:2 샘플 포맷에 따라 포맷된 16x16 코딩 유닛을 예시하는 개념도이다.

도 5 는 인트라 블록 복사 (IBC) 모드의 개념적인 예시를 나타낸다.

도 6 은 타일들이 사용되는 경우 픽처의 래스터 스캔의 예를 나타낸다.

도 7 은 병렬의 CTB들의 WPP 프로세스들 로우들의 예를 나타내고, 각각의 로우는 위의 로우의 제 2 CTB 를 프로세싱한 후에 이용 가능한 CABAC 확률들로 시작한다.

도 8 내지 도 12 는 본 개시물에 설명된 다양한 예들에 대한 유효 예측 영역들을 나타낸다.

도 13 은 intra_bc_flag 선택스 엘리먼트를 시그널링하는 예시의 방법을 나타낸다.

도 14 는 intra_bc_flag 선택스 엘리먼트를 시그널링하는 다른 예시의 방법을 나타낸다.

도 15 는 본 개시물에서 설명된 기법들을 구현할 수도 있는 예시의 비디오 인코더를 예시하는 블록도이다.

도 16 은 본 개시물에서 설명된 기법들을 구현할 수도 있는 예시의 비디오 디코더를 예시하는 블록도이다.

도 17 은 본 개시물에 설명된 기법들을 통합하는 예시의 비디오 인코딩 프로세스를 예시하는 블록도이다.

도 18 은 본 개시물에 설명된 기법들을 통합하는 예시의 비디오 디코딩 프로세스를 예시하는 블록도이다.

발명을 실시하기 위한 구체적인 내용

[0018] 최근 개발된 고 효율성 비디오 코딩 (HEVC) 표준을 포함하는, 다양한 비디오 코딩 표준들은 비디오 블록들에 대한 예측 코딩 모드들을 포함하고, 여기서 현재 코딩되고 있는 블록은 비디오 데이터의 이미 코딩된 블록에 기초하여 예측된다. 인트라 예측 모드에서, 현재 블록은 현재 블록과 동일한 픽처에서 하나 이상의 이전에 코딩된 이웃하는 블록들에 기초하여 예측되는 한편, 인터-예측 모드에서 현재 블록은 상이한 픽처에서 이미 코딩된 블록에 기초하여 예측된다. 인터-예측 모드에서, 예측 블록으로서 사용하기 위해 이전에 코딩된 프레임의 블록을 결정하는 프로세스는 때때로, 일반적으로 비디오 인코더에 의해 수행되는 모션 추정으로서 지칭되고, 예측 블록을 식별 및 취출하는 프로세스는 때때로, 비디오 인코더들 및 비디오 디코더들 양자 모두에 의해 수행되는 모션 보상으로 지칭된다.

[0019] 비디오 인코더는 통상적으로, 다수의 코딩 시나리오들을 사용하여 비디오를 코딩하고 바람직한 레이트-왜곡 트레이드오프를 생성하는 코딩 시나리오를 식별함으로써 비디오 데이터의 시퀀스를 코딩하는 방법을 결정한다.

특정 비디오 블록에 대한 인트라 예측 코딩 시나리오를 테스트하는 경우, 비디오 인코더는 통상적으로, 픽셀들의 이웃하는 로우 (즉, 코딩되고 있는 블록 바로 위의 픽셀들의 로우) 를 테스트하고, 픽셀들의 이웃하는 컬럼 (즉, 코딩되고 있는 블록 바로 좌측의 픽셀들의 컬럼) 을 테스트한다. 반대로, 인터 예측 시나리오들을 테스트하는 경우, 비디오 인코더는 통상적으로, 보다 큰 검색 영역에서 후보 예측 블록들을 식별하고, 여기서 검색 영역은 비디오 데이터의 이전에 코딩된 프레임들 내의 비디오 블록들에 대응한다.

[0020] 그러나, 텍스트, 심볼들, 또는 반복적인 패턴들을 포함하는 비디오 이미지들과 같은 비디오 이미지들의 소정 유형들에 대해, 때때로 인트라 모션 보상 (IMC) 모드로도 지칭되는, 인트라 블록 복사 (IBC) 모드를 사용함으로써 인트라 예측 및 인터 예측에 대해 코딩 이득들이 달성될 수 있다는 것이 발견되었다. 다양한 코딩 표준들의 개발에서, 용어 IMC 모드가 원래 사용되었지만, 이후에는 IBC 모드로 변경되었다. IBC 모드에서, 비디오 인코더는 인트라 예측 모드에서와 같이 코딩되고 있는 블록과 동일한 프레임 또는 픽처에서 예측 블록을 검색하지만, 비디오 인코더는 보통 픽셀들의 이웃하는 로우들 및 컬럼들 뿐만 아니라 더 넓은 검색 영역을 검색한다.

- [0021] IBC 모드에서, 비디오 인코더는 예측되고 있는 블록과 동일한 프레임 또는 픽처 내의 예측 블록을 식별하기 위해, 때때로 모션 벡터 또는 블록 벡터로도 지칭된, 오프셋 벡터를 결정할 수도 있다. 오프셋 벡터는, 예를 들어 x-컴포넌트 및 y-컴포넌트를 포함하고, 여기서 x-컴포넌트는 예측되고 있는 비디오 블록과 예측 블록 간의 수평 변위를 식별하며, y-컴포넌트는 예측되고 있는 비디오 블록과 예측 블록 간의 수직 변위를 식별한다. 비디오 인코더는, 인코딩된 비트스트림을 디코딩하는 경우, 비디오 디코더가 비디오 인코더에 의해 선택된 동일한 예측 블록을 식별할 수 있도록, 결정된 오프셋 벡터를 인코딩된 비트스트림에서 시그널링한다.
- [0022] HEVC 를 포함하는 다양한 비디오 코딩 표준들은 또한, 동일한 픽처 내의 상이한 블록들이 동시에 디코딩될 수도 있도록 타일들 및 파면 (wavefront) 병렬 프로세싱과 같은 병렬 프로세싱 메커니즘들을 지원한다. 타일들은, 비디오 디코더가 다수의 타일들을 병렬로 디코딩할 수 있도록, (파싱 및 복원을 포함하는) 다수의 독립적으로 디코딩 가능한 영역들로의 픽처의 (코딩된 트리 블록 (CTB) 입도를 갖는) 직사각형 파티셔닝을 제공한다. 타일들과 다르게, 파면들은 독립적으로 디코딩 가능하지 않지만, 비디오 디코더는 다양한 파면들의 디코딩이 시작되는 시간을 스테거링함으로써 다수의 파면들을 병렬로 디코딩할 수도 있다. 예를 들어, 비디오 디코더가 제 1 파면 아래의 제 2 파면을 디코딩하기 시작하기 전에 제 1 파면의 2 개의 블록들을 디코딩하면, 비디오 디코더는, 제 2 파면을 디코딩하기 위해 필요한 제 1 파면의 임의의 정보가 이미 디코딩되고, 따라서 제 2 파면을 디코딩하는데 사용하기 위해 이용 가능하다는 것을 보장할 수 있다.
- [0023] 본 개시물은, IBC 모드가 인에이블되는 경우 병렬 프로세싱을 잠재적으로 강화시키기 위한 기법들을 도입한다. 보다 구체적으로, 본 개시물은, 디코더가, 때때로 파면 병렬 프로세싱으로서 지칭되는 비-래스터 스캔 순서로 다수의 CTU들을 병렬로 프로세싱할 수 있도록 IBC 블록 벡터들 (BVs) 에 대한 제한들을 도입한다. 본 개시물의 기법들은, 가능하게는 고 비트 심도 (8 보다 큰 비트) 의, 4:4:4, 4:2:2, 4:2:0, 4:0:0 등과 같은 상이한 크로마 샘플링 포맷의 지원을 포함하는, 스크린 콘텐츠 코딩에 관한 것이지만, 이에 한정되지는 않는다.
- [0024] 도 1 은 IBC 모드에서 블록들을 코딩하기 위한 기법들 및 병렬 프로세싱을 위한 기법들을 포함하는, 본 개시물에 설명된 기법들을 이용할 수도 있는 예시의 비디오 인코딩 및 디코딩 시스템 (10) 을 예시하는 블록도이다. 도 1 에 도시된 바와 같이, 시스템 (10) 은 목적지 디바이스 (14) 에 의해 나중에 디코딩될 인코딩된 비디오 데이터를 제공하는 소스 디바이스 (12) 를 포함한다. 소스 디바이스 (12) 및 목적지 디바이스 (14) 는, 데스크톱 컴퓨터들, 노트북 (즉, 랩톱) 컴퓨터들, 태블릿 컴퓨터들, 셋톱 박스들, 소위 "스마트" 폰들과 같은 전화 핸드셋들, 소위 "스마트" 패드들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 콘솔들, 비디오 스트리밍 디바이스 등을 포함하는 다양한 범위의 디바이스들 중 임의의 것을 포함할 수도 있다. 일부 경우들에서, 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 무선 통신을 위해 구비될 수도 있다.
- [0025] 목적지 디바이스 (14) 는 디코딩될 인코딩된 비디오 데이터를 링크 (16) 를 통해 수신할 수도 있다. 링크 (16) 는 인코딩된 비디오 데이터를 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 이동시킬 수 있는 임의의 유형의 매체 또는 디바이스를 포함할 수도 있다. 일 예에서, 링크 (16) 는 소스 디바이스 (12) 로 하여금 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로 실시간으로 직접 송신하게 하는 통신 매체를 포함할 수도 있다. 인코딩된 비디오 데이터는 무선 통신 프로토콜과 같은 통신 표준에 따라 변조되고, 목적지 디바이스 (14) 로 송신될 수도 있다. 통신 매체는 임의의 무선 또는 유선 통신 매체, 예컨대 무선 주파수 (RF) 스펙트럼 또는 하나 이상의 물리적 송신 라인들을 포함할 수도 있다. 통신 매체는 패킷 기반의 네트워크, 예컨대 근거리 통신망, 광대역 통신망, 또는 인터넷과 같은 글로벌 네트워크의 일부를 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.
- [0026] 대안으로, 인코딩된 데이터는 출력 인터페이스 (22) 로부터 저장 디바이스 (17) 로 출력될 수도 있다. 유사하게, 인코딩된 데이터는 입력 인터페이스에 의해 저장 디바이스 (17) 로부터 액세스될 수도 있다. 저장 디바이스 (17) 는 하드 드라이브, 블루레이 디스크들, DVD들, CD-ROM들, 플래시 메모리, 휘발성 또는 비휘발성 메모리와 같은 임의의 다양한 분산된 또는 로컬하게 액세스된 데이터 저장 매체, 또는 인코딩된 비디오 데이터를 저장하기 위한 임의의 다른 적합한 디지털 저장 매체를 포함할 수도 있다. 추가의 예에서, 저장 디바이스 (17) 는 소스 디바이스 (12) 에 의해 생성된 인코딩된 비디오를 홀딩할 수도 있는 다른 중간 저장 디바이스 또는 파일 서버에 대응할 수도 있다. 목적지 디바이스 (14) 는 스트리밍 또는 다운로드를 통해 저장 디바이스 (17) 로부터 저장된 비디오 데이터에 액세스할 수도 있다. 파일 서버는 인코딩된 비디오 데이터를 저장하고 그 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로 송신할 수 있는 임의의 유형의 서버일 수도 있다. 예시의 파일 서버들은 (예를 들어, 웹사이트용의) 웹서버, FTP 서버, NAS (network attached storage) 디바이스

스들, 또는 로컬 디스크 드라이브를 포함한다. 목적지 디바이스 (14) 는 인터넷 접속을 포함하는 임의의 표준 데이터 접속을 통해 인코딩된 비디오 데이터에 액세스할 수도 있다. 이것은 파일 서버에 저장된 인코딩된 비디오 데이터를 액세스하는데 적합한 무선 채널 (예를 들어, Wi-Fi 접속), 유선 접속 (예를 들어, DSL, 케이블 모뎀 등), 또는 양자의 조합을 포함할 수도 있다. 저장 디바이스 (17) 로부터 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 양자의 조합일 수도 있다.

[0027] 본 개시물의 기법들은 무선 애플리케이션들 또는 설정들에 반드시 한정되는 것은 아니다. 본 기법은 임의의 다양한 멀티미디어 어플리케이션들, 예컨대 지상파 (over-the-air) 텔레비전 방송들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, 예를 들어 인터넷을 통한 스트리밍 비디오 송신들, 데이터 저장 매체 상의 저장을 위한 디지털 비디오의 인코딩, 데이터 저장 매체 상에 저장된 디지털 비디오의 디코딩, 또는 다른 애플리케이션들을 지원하여 비디오 코딩에 적용될 수도 있다. 일부 예들에서, 시스템 (10) 은, 비디오 스트리밍, 비디오 재생, 비디오 방송, 및/또는 비디오 텔레포니와 같은 애플리케이션들을 지원하기 위해 일방향 또는 양방향 비디오 송신을 지원하도록 구성될 수도 있다.

[0028] 도 1 의 예에서, 소스 디바이스 (12) 는 비디오 소스 (18), 비디오 인코더 (20) 및 출력 인터페이스 (22) 를 포함한다. 일부 경우들에서, 출력 인터페이스 (22) 는 변조기/복조기 (modem) 및/또는 송신기를 포함할 수도 있다. 소스 디바이스 (12) 에서, 비디오 소스 (18) 는 비디오 캡처 디바이스, 예를 들어 비디오 카메라와 같은 소스, 이전에 캡처된 비디오를 포함하는 비디오 아카이브, 비디오 콘텐츠 공급자로부터 비디오를 수신하는 비디오 공급 인터페이스, 및/또는 소스 비디오와 같은 컴퓨터 그래픽 데이터를 생성하는 컴퓨터 그래픽 시스템, 또는 이러한 소스들의 조합을 포함할 수도 있다. 일 예로서, 비디오 소스 (18) 가 비디오 카메라이면, 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 소위 카메라 폰들 또는 비디오 폰들을 형성할 수도 있다. 그러나, 본 개시물에서 설명된 기법들은 일반적으로 비디오 코딩에 적용될 수 있으며, 무선 및/또는 유선 애플리케이션들에 적용될 수도 있다.

[0029] 캡처된, 미리-캡처된, 또는 컴퓨터-생성된 비디오는 비디오 인코더 (20) 에 의해 인코딩될 수도 있다. 인코딩된 비디오 데이터는 소스 디바이스 (12) 의 출력 인터페이스 (22) 를 통해 목적지 디바이스 (14) 로 직접적으로 송신될 수도 있다. 인코딩된 비디오 데이터는, 디코딩 및/또는 재생을 위해 목적지 디바이스 (20) 또는 다른 디바이스들에 의한 나중의 액세스를 위해 저장 디바이스 (17) 에 또한 (또는 대안으로) 저장될 수도 있다.

[0030] 목적지 디바이스 (14) 는 입력 인터페이스 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (32) 를 포함한다. 일부 경우들에서, 입력 인터페이스 (28) 는 수신기 및/또는 모뎀을 포함할 수도 있다. 목적지 디바이스 (14) 의 입력 인터페이스 (28) 는 링크 (16) 를 통해 인코딩된 비디오 데이터를 수신한다. 링크 (16) 를 통해 통신된, 또는 저장 디바이스 (17) 상에 제공된 인코딩된 비디오 데이터는, 비디오 데이터의 디코딩에서 비디오 디코더, 예컨대 비디오 디코더 (30) 에 의한 사용을 위해 비디오 인코더 (20) 에 의해 생성된 다양한 선택스 엘리먼트들을 포함할 수도 있다. 이러한 선택스 엘리먼트들은 통신 매체 상에서 송신된 인코딩된 비디오 데이터에 포함되거나, 저장 매체에 저장되거나, 파일 서버에 저장될 수도 있다.

[0031] 디스플레이 디바이스 (32) 는 목적지 디바이스 (14) 와 통합될 수도 있거나 또는 목적지 디바이스 (14) 외부에 있을 수도 있다. 일부 예들에서, 목적지 디바이스 (14) 는 통합된 디스플레이 디바이스를 포함할 수도 있고, 또한 외부 디스플레이 디바이스와 인터페이스하도록 구성될 수도 있다. 다른 예들에서, 목적지 디바이스 (14) 는 디스플레이 디바이스일 수도 있다. 일반적으로, 디스플레이 디바이스 (32) 는 디코딩된 비디오 데이터를 사용자에게 디스플레이하고, 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.

[0032] 비디오 인코더 (20) 및 비디오 디코더 (30) 는 비디오 압축 표준, 예컨대 HEVC 에 따라 동작할 수도 있고, HEVC 테스트 모델 (HM) 을 준수할 수도 있다. "HEVC 작업 초안 10" 또는 "HEVC WD10" 으로서 지칭된, HEVC 표준의 작업 초안은, ITU-T SG16 WP3 및 ISO/IEC JTC1/SC29/WG11 의 비디오 코딩에 관한 합동 연구팀 (JCT-VC), 제 13 차 회의, 한국 인천, 2013 년 4월, Bross 등의, "Editors' proposed corrections to HEVC version 1" 에서 설명된다. 다른 HEVC 초안 사양은 http://phenix.int-evry.fr/jct/doc_end_user/documents/15_Geneva/wg11/JCT_VC-01003-v2.zip 로부터 입수 가능하다. 본 개시물에 설명된 기법들은 또한, 현재 개발 중에 있는 HEVC 표준의 확장들에 따라 동작할 수도 있다.

[0033] 대안으로 또는 부가적으로, 비디오 인코더 (20) 및 비디오 디코더 (30) 는, MPEG-4, 파트 10, 어드밴스드 비디오 코딩 (AVC) 으로서 다르게 지칭된 ITU-T H.264 표준과 같은 다른 사실 표준 또는 산업 표준, 또는 이러한 표

준들의 확장안들에 따라 동작할 수도 있다. 그러나, 본 개시물의 기술들은 임의의 특정 코딩 표준에 한정되지 않는다. 비디오 압축 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 및 그 자신의 SVC (Scalable Video Coding) 및 MVC (Multiview Video Coding) 확장안들을 포함하는 (또한 ISO/IEC MPEG-4 AVC 로도 알려진) ITU-T H.264 을 포함한다.

[0034] HEVC 의 설계는 ITU-T VCEG (Video Coding Experts Group) 및 ISO/IEC MPEG (Motion Picture Experts Group) 의 JCT-VC 에 의해 최근 완결되어 있다. HEVC RExt 로서 지칭된, HEVC 에 대한 범위 확장들은 또한, JCT-VC 에 의해 개발되고 있다. 이하에서 RExt WD7 로서 지칭된, 범위 확장들의 최근의 작업 초안은 http://phenix.int-evry.fr/jct/doc_end_user/documents/17_Valencia/wg11/JCTVC-Q_1005-v4.zip 로부터 입수 가능하다.

[0035] 본 개시물은 일반적으로, HEVC 버전 1 또는 기본 HEVC 로서 최근 완결된 HEVC 사양 텍스트를 지칭할 것이다. 범위 확장 사양은 HEVC 의 버전 2 가 될 수도 있다. 모션 벡터 예측과 같은 많은 코딩 툴들에 대하여, HEVC 버전 1 및 범위 확장 사양은 기술적으로 유사하다. 따라서, 본 개시물이 HEVC 버전 1 과 관련된 변화들을 설명할 때마다, 동일한 변화들이 또한, 기본 HEVC 사양, 및 일부 추가적인 코딩 툴들을 포함하는 범위 확장 사양에 적용할 수도 있다. 또한, 일반적으로 HEVC 버전 1 모듈들이 또한, HEVC 범위 확장을 구현하는 디코더에 통합 될 수도 있다는 것이 가정될 수 있다.

[0036] 모션이 있는 그래픽들 및 텍스트와 같은 스크린-콘텐츠 자료에 대한 새로운 코딩 툴들은 현재 개발 중에 있고, HEVC 의 향후 버전을 포함하여 향후 비디오 코딩 표준들에 포함되도록 고려되고 있다. 이들 새로운 코딩 툴들은 잠재적으로, 스크린 콘텐츠에 대한 코딩 효율성을 개선시킨다. 신규한 전용 코딩 툴들을 이용하여 스크린 콘텐츠의 특징을 활용함으로써 코딩 효율성에서 상당한 개선들이 획득될 수도 있다는 증거가 존재하기 때문에, CfP (Call for Proposals) 가 (SCC 용 특정 툴들을 포함하는) HEVC 표준의 향후 확장들을 가능하게는 개발하는 목표를 갖고 이슈화되고 있다. 이 콜에 응답하여, 회사들 및 조직들은 제안서를 제출하도록 요청받아 왔다. 이 CfP 의 사용 케이스들 및 요건들은 MPEG 문헌 N14174 에 설명된다. 제 17 차 JCT-VC 미팅 동안, SCC 테스트 모델 (SCM) 이 확립된다. SCC 의 최근 작업 초안 (WD) 은 http://phenix.int-evry.fr/jct/doc_end_user/documents/18_Sapporo/wg11/JCTVC-R1005-v3.zip 에서 입수 가능하다.

[0037] 일반적으로, 소스 디바이스 (12) 의 비디오 인코더 (20) 는 이들 현재 또는 향후의 표준들에 따라 비디오 데이터를 인코딩하도록 구성될 수도 있는 것으로 고려된다. 유사하게, 목적지 디바이스 (14) 의 비디오 디코더 (30) 는 이들 현재 또는 향후의 표준들에 따라 비디오 데이터를 디코딩하도록 구성될 수도 있는 것으로 고려된다.

[0038] 도 1 에 도시되지 않았으나, 일부 양태들에서 비디오 인코더 (20) 및 비디오 디코더 (30) 는 각각 오디오 인코더 및 디코더와 통합될 수도 있고, 공통의 데이터 스트림 또는 별개의 데이터 스트림들에서 오디오 및 비디오 양자의 인코딩을 핸들링하기 위해 적절한 MUX-DEMUX 유닛들, 또는 다른 하드웨어 및 소프트웨어를 포함할 수도 있다. 적용 가능하다면, 일부 예들에서, MUX-DEMUX 유닛들은 ITU H.223 멀티플렉서 프로토콜, 또는 사용자 데이터그램 프로토콜 (UDP) 과 같은 다른 프로토콜을 준수할 수도 있다.

[0039] 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 임의의 다양한 적합한 인코더 회로부, 예컨대 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서 (DSP) 들, 주문형 집적 회로들 (ASIC) 들, 필드 프로그래머블 게이트 어레이 (FPGA) 들, 이산 로직, 소프트웨어, 하드웨어, 펌웨어, 또는 이들의 임의의 조합들로서 구현될 수도 있다. 이 기법들이 부분적으로 소프트웨어로 구현되는 경우, 디바이스는 그 소프트웨어에 대한 명령들을 적합한, 비일시적 컴퓨터 판독가능 매체에 저장할 수도 있고, 본 개시물의 기법들을 수행하기 위해 하나 이상의 프로세서들을 사용하는 하드웨어에서 그 명령들을 실행할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 인코더들 또는 디코더들에 포함될 수도 있고, 이들 중 어느 것이 결합된 인코더/디코더 (CODEC) 의 일부로서 각각의 디바이스에 통합될 수도 있다.

[0040] 위에서 소개된 바와 같이, JCT-VC 는 최근에, HEVC 표준의 개발을 완결하였다. HEVC 표준화 노력들은 HEVC 테스트 모델 (HM) 로 지칭되는 비디오 코딩 디바이스의 진화 모델에 기초하였다. HM 은, 예를 들어 ITU-T H.264/AVC 에 따른 기존 디바이스들에 비해 비디오 코딩 디바이스들의 여러 추가적인 성능들을 가정한다. 예를 들어, H.264 가 9 개의 인트라 예측 인코딩 모드들을 제공하는 반면에, HM 은 35 개만큼 많은 인트라 예측 인코딩 모드들을 제공할 수도 있다.

- [0041] HEVC 및 다른 비디오 코딩 사양들에서, 비디오 시퀀스는 통상적으로 일련의 픽처들을 포함한다. 픽처들은 또한, "프레임들"로서 지칭될 수도 있다. 픽처는 S_L , S_{Cb} 및 S_{Cr} 로 표기된 3개의 샘플 어레이들을 포함할 수도 있고, S_L 은 루마 샘플들의 2-차원 어레이(즉, 블록)이다. S_{Cb} 는 Cb 크로미넌스 샘플들의 2-차원 어레이이다. S_{Cr} 은 Cr 크로미넌스 샘플들의 2-차원 어레이이다. 크로미넌스 샘플들은 또한, 본원에서 "크로마" 샘플들로서 지칭될 수도 있다. 다른 경우들에서, 픽처는 모노크롬일 수도 있고, 단지 루마 샘플들의 어레이를 포함할 수도 있다.
- [0042] 픽처의 인코딩된 표현을 생성하기 위해, 비디오 인코더(20)는 코딩 트리 유닛(CTU)들의 세트를 생성할 수도 있다. CTU들 각각은 루마 샘플들의 코딩 트리 블록(CTB), 크로마 샘플들의 2개의 대응하는 코딩 트리 블록들, 및 코딩 트리 블록들의 샘플들을 코딩하는데 사용된 신택스 구조들을 포함할 수도 있다. 모노크롬 픽처들 또는 3개의 별개의 컬러 평면들을 갖는 픽처들에서, CTU는 단일의 코딩 트리 블록 및 코딩 트리 블록의 샘플들을 코딩하는데 사용된 신택스 구조들을 포함할 수도 있다. 코딩 트리 블록은 샘플들의 $N \times N$ 블록일 수도 있다. CTU는 또한, "트리 블록" 또는 "최대 코딩 유닛"(LCU)으로서 지칭될 수도 있다. HEVC의 CTU들은 H.264/AVC와 같은 다른 표준들의 매크로블록들과 광범위하게 유사할 수도 있다. 그러나, CTU는 반드시 특정 사이즈에 한정되지 않고, 하나 이상의 코딩 유닛들(CUs)을 포함할 수도 있다. 슬라이스는 래스터 스캔 순서로 연속적으로 오더링된 정수의 CTU들을 포함할 수도 있다.
- [0043] 코딩된 CTU를 생성하기 위해, 비디오 인코더(20)는 CTU의 코딩 트리 블록들(CTBs)상에서 쿼드-트리 파티셔닝을 재귀적으로 수행하여 코딩 트리 블록들을 코딩 블록들, 따라서 명칭 "코딩 트리 유닛들"로 분할할 수도 있다. 코딩 블록은 샘플들의 $N \times N$ 블록일 수도 있다. CU는, 루마 샘플 어레이, Cb 샘플 어레이, 및 Cr 샘플 어레이를 갖는 픽처의 루마 샘플들의 코딩 블록 및 크로마 샘플들의 2개의 대응하는 코딩 블록들, 및 코딩 블록들의 샘플들을 코딩하는데 사용된 신택스 구조들을 포함할 수도 있다. 모노크롬 픽처들 또는 3개의 별개의 컬러 평면들을 갖는 픽처들에서, CU는 단일의 코딩 블록 및 코딩 블록의 샘플들을 코딩하는데 사용된 신택스 구조들을 포함할 수도 있다.
- [0044] 비디오 인코더(20)는 CU의 코딩 블록을 하나 이상의 예측 블록들로 파티셔닝할 수도 있다. 예측 블록은, 동일한 예측이 적용되는 샘플들의 직사각형(즉, 정사각형 또는 비-정사각형) 블록이다. CU의 예측 유닛(PU)은 루마 샘플들의 예측 블록, 크로마 샘플들의 2개의 대응하는 예측 블록들, 및 예측 블록들을 예측하는데 사용된 신택스 구조들을 포함할 수도 있다. 모노크롬 픽처들 또는 3개의 별개의 컬러 평면들을 갖는 픽처들에서, PU는 단일의 예측 블록 및 예측 블록을 예측하는데 사용된 신택스 구조들을 포함할 수도 있다. 비디오 인코더(20)는 CU의 각각의 PU의 루마, Cb 및 Cr 예측 블록들에 대한 예측 루마, Cb, 및 Cr 블록들을 생성할 수도 있다.
- [0045] 비디오 인코더(20)는 인트라 예측 또는 인터 예측을 사용하여 PU에 대한 예측 블록들을 생성할 수도 있다. 비디오 인코더(20)가 인트라 예측을 사용하여 PU의 예측 블록을 생성하면, 비디오 인코더(20)는 PU와 연관된 픽처의 디코딩된 샘플들에 기초하여 PU의 예측 블록들을 생성할 수도 있다. 비디오 인코더(20)가 인터 예측을 사용하여 PU의 예측 블록을 생성하면, 비디오 인코더(20)는 PU와 연관된 픽처 외에 하나 이상의 픽처들의 디코딩된 샘플들에 기초하여 PU의 예측 블록들을 생성할 수도 있다.
- [0046] 비디오 인코더(20)가 CU의 하나 이상의 PU들에 대한 예측 루마, Cb, 및 Cr 블록들을 생성한 후에, 비디오 인코더(20)는 CU에 대한 루마 잔여 블록을 생성할 수도 있다. CU의 루마 잔여 블록에서 각각의 샘플은, CU의 예측 루마 블록들 중 하나에서의 루마 샘플과 CU의 원래의 루마 코딩 블록에서의 대응하는 샘플 간의 차이를 나타낼 수도 있다. 또한, 비디오 인코더(20)는 CU에 대한 Cb 잔여 블록을 생성할 수도 있다. CU의 Cb 잔여 블록에서 각각의 샘플은, CU의 예측 Cb 블록들 중 하나에서의 Cb 샘플과 CU의 원래의 Cb 코딩 블록에서의 대응하는 샘플 간의 차이를 나타낼 수도 있다. 비디오 인코더(20)는 또한, CU에 대한 Cr 잔여 블록을 생성할 수도 있다. CU의 Cr 잔여 블록에서 각각의 샘플은, CU의 예측 Cr 블록들 중 하나에서의 Cr 샘플과 CU의 원래의 Cr 코딩 블록에서의 대응하는 샘플 간의 차이를 나타낸다.
- [0047] 또한, 비디오 인코더(20)는 쿼드-트리 파티셔닝을 사용하여, CU의 루마, Cb, 및 Cr 잔여 블록들을 하나 이상의 루마, Cb, 및 Cr 변환 블록들로 분할할 수도 있다. 변환 블록은, 동일한 변환이 적용되는 샘플들의 직사각형(예를 들어, 정사각형 또는 비-정사각형) 블록이다. CU의 변환 유닛(TU)은 루마 샘플들의 변환 블록, 크로마 샘플들의 2개의 대응하는 변환 블록들, 및 변환 블록 샘플들을 변환하는데 사용된 신택스 구조들을 포함할 수도 있다. 따라서, CU의 각각의 TU는 루마 변환 블록, Cb 변환 블록, 및 Cr 변환 블록과 연관될

수도 있다. TU 와 연관된 루마 변환 블록은 CU 의 루마 잔여 블록의 서브-블록일 수도 있다. Cb 변환 블록은 CU 의 Cb 잔여 블록의 서브-블록일 수도 있다. Cr 변환 블록은 CU 의 Cr 잔여 블록의 서브-블록일 수도 있다. 모노크롬 픽처들 또는 3 개의 별개의 컬러 평면들을 갖는 픽처들에서, TU 는 단일의 변환 블록 및 변환 블록의 샘플들을 변환하는데 사용된 선택스 구조들을 포함할 수도 있다.

[0048] 비디오 인코더 (20) 는 TU 의 루마 변환 블록에 하나 이상의 변환들을 적용하여 TU 에 대한 루마 계수 블록을 생성할 수도 있다. 계수 블록은 변환 계수들의 2-차원 어레이일 수도 있다. 변환 계수는 스칼라 양일 수도 있다. 비디오 인코더 (20) 는 TU 의 Cb 변환 블록에 하나 이상의 변환들을 적용하여 TU 에 대한 Cb 계수 블록을 생성할 수도 있다. 비디오 인코더 (20) 는 TU 의 Cr 변환 블록에 하나 이상의 변환들을 적용하여 TU 에 대한 Cr 계수 블록을 생성할 수도 있다.

[0049] 계수 블록 (예를 들어, 루마 계수 블록, Cb 계수 블록 또는 Cr 계수 블록) 을 생성한 후에, 비디오 인코더 (20) 는 계수 블록을 양자화할 수도 있다. 양자화는 일반적으로, 변환 계수들을 표현하기 위해 사용된 데이터의 양을 가능하게는 감소시키기 위해 변환 계수들이 양자화되어, 추가의 압축을 제공하는 프로세스를 지칭한다. 비디오 인코더 (20) 가 계수 블록을 양자화한 후에, 비디오 인코더 (20) 는 양자화된 변환 계수들을 나타내는 선택스 엘리먼트들을 엔트로피 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 양자화된 변환 계수들을 나타내는 선택스 엘리먼트들 상에 컨텍스트-적응 바이너리 산술 코딩 (CABAC) 을 수행할 수도 있다.

[0050] 비디오 인코더 (20) 는 코딩된 픽처들 및 연관된 데이터의 표현을 형성하는 비트들의 시퀀스를 포함하는 비트스트림을 출력할 수도 있다. 비트스트림은 NAL 유닛들의 시퀀스를 포함할 수도 있다. NAL 유닛은, NAL 유닛으로 데이터 유형의 표시를 포함하는 선택스 구조 및 에플레이션 방지 비트들 사이에 필요에 따라 배치된 RBSP 의 형태로 그 데이터를 포함하는 바이트들이다. NAL 유닛들 각각은 NAL 유닛 헤더를 포함하고 RBSP 를 캡슐화한다. NAL 유닛 헤더는 NAL 유닛 유형 코드를 나타내는 선택스 엘리먼트를 포함할 수도 있다. NAL 유닛의 NAL 유닛 헤더에 의해 지정된 NAL 유닛 유형 코드는 NAL 유닛의 유형을 나타낸다. RBSP 는 NAL 유닛 내에 캡슐화되는 정수의 바이트들을 포함하는 선택스 구조일 수도 있다. 일부 경우들에서, RBSP 는 제로 비트들을 포함한다.

[0051] NAL 유닛들의 상이한 유형들은 RBSP들의 상이한 유형들을 캡슐화할 수도 있다. 예를 들어, NAL 유닛의 제 1 유형은 PPS 에 대한 RBSP 를 캡슐화할 수도 있고, NAL 유닛의 제 2 유형은 코딩된 슬라이스에 대한 RBSP 를 캡슐화할 수도 있고, NAL 유닛의 제 3 유형은 SEI 메시지들에 대한 RBSP 를 캡슐화할 수도 있는 등이다. (SEI 메시지들 및 파라미터 세트들에 대한 RBSP들과 대조적으로) 비디오 코딩 데이터에 대한 RBSP들을 캡슐화하는 NAL 유닛들은 VCL NAL 유닛들로서 지칭될 수도 있다.

[0052] 비디오 디코더 (30) 는 비디오 인코더 (20) 에 의해 생성된 비트스트림을 수신할 수도 있다. 또한, 비디오 디코더 (30) 는 비트스트림을 파싱 (parse) 하여 비트스트림으로부터 선택스 엘리먼트들을 획득할 수도 있다. 비디오 디코더 (30) 는 비트스트림으로부터 획득된 선택스 엘리먼트들에 적어도 부분적으로 기초하여 비디오 데이터의 픽처들을 복원할 수도 있다. 비디오 데이터를 복원하기 위한 프로세스는 일반적으로, 비디오 인코더 (20) 에 의해 수행된 프로세스와 상반될 수도 있다. 또한, 비디오 디코더 (30) 는 현재 CU 의 TU들과 연관된 계수 블록들을 역양자화할 수도 있다. 비디오 디코더 (30) 는 계수 블록들 상에서 역변환을 수행하여, 현재 CU 의 TU들과 연관된 변환 블록들을 복원할 수도 있다. 비디오 디코더 (30) 는 현재 CU 의 PU들에 대한 예측 블록들의 샘플들을 현재 CU 의 TU들의 변환 블록들의 대응하는 샘플들에 추가함으로써 현재 CU 의 코딩 블록들을 복원할 수도 있다. 픽처의 각각의 CU 에 대한 코딩 블록들을 복원함으로써, 비디오 디코더 (30) 는 픽처를 복원할 수도 있다.

[0053] 크로마 포맷으로도 지칭될 수도 있는 비디오 샘플링 포맷은 CU 에 포함된 루마 샘플들의 수에 대하여 CU 에 포함된 크로마 샘플들의 수를 정의할 수도 있다. 크로마 컴포넌트들에 대한 비디오 샘플링 포맷에 따라, U 및 V 컴포넌트들의, 샘플들의 수의 관점들에서, 사이즈는 Y 컴포넌트의 사이즈와 동일하거나 이와 상이할 수도 있다. HEVC 표준에서, chroma_format_idc 로 지칭된 값은, 루마 컴포넌트와 관련된, 크로마 컴포넌트들의 상이한 샘플링 포맷들을 나타내도록 정의된다. HEVC 에서, chroma_format_idc 는 SPS 에서 시그널링된다. 표 1 은 chroma_format_idc 의 값들과, 연관된 크로마 포맷들 간의 관계를 예시한다.

표 1

chroma_format_idc	크로마 포맷	서브폭C	서브높이C
0	모노크롬	-	-
1	4:2:0	2	2
2	4:2:2	2	1
3	4:4:4	1	1

[0054]

[0055]

표 1: HEVC 에서 정의된 상이한 크로마 포맷들

[0056]

표 1 에서, 변수들 서브폭C (SubWidthC) 및 서브높이C (SubHeightC) 는 각각의 크로마 컴포넌트에 대한 샘플들의 수와 루마 컴포넌트에 대한 샘플들의 수 간의 수평 및 수직 샘플링 레이트 비를 나타내는데 사용될 수 있다.

표 1 에 설명된 크로마 포맷들에서, 2 개의 크로마 컴포넌트들은 동일한 샘플링 레이트를 갖는다. 따라서, 4:2:0 샘플링에서, 2 개의 크로마 어레이들 각각은 루마 어레이의 높이의 절반 및 폭의 절반을 갖는 한편, 4:2:2 샘플링에서, 2 개의 크로마 어레이들 각각은 루마 어레이의 동일한 높이 및 폭의 절반을 갖는다. 4:4:4 샘플링에서, 2 개의 크로마 어레이들 각각은 루마 어레이와 동일한 높이 및 폭을 가질 수도 있고, 또는 일부 경우들에서 3 개의 컬러 평면들은 모두 모노크롬 샘플링된 픽처들로서 별개로 프로세싱될 수도 있다.

[0057]

표 1 의 예에서, 4:2:0 포맷에 대해, 루마 컴포넌트에 대한 샘플링 레이트는 수평 및 수직 방향을 양자 모두에 대해 크로마 컴포넌트들의 샘플링 레이트의 2 배이다. 결과적으로, 4:2:0 포맷에 따라 포맷된 코딩 유닛에 대해, 루마 컴포넌트에 대한 샘플들의 어레이의 폭 및 높이는 크로마 컴포넌트들에 대한 샘플들의 각각의 어레이의 폭 및 높이의 2 배이다.

유사하게, 4:2:2 포맷에 따라 포맷된 코딩 유닛에 대해, 루마 컴포넌트에 대한 샘플들의 어레이의 폭은 각각의 크로마 컴포넌트에 대한 샘플들의 어레이의 폭의 2 배의 폭이지만, 루마 컴포넌트에 대한 샘플들의 어레이의 높이는 각각의 크로마 컴포넌트에 대한 샘플들의 어레이의 높이와 동일하다.

4:4:4 포맷에 따라 포맷된 코딩 유닛에 대해, 루마 컴포넌트에 대한 샘플들의 어레이는 각각의 크로마 컴포넌트에 대한 샘플들의 어레이와 동일한 폭 및 높이를 갖는다. YUV 컬러 공간에 추가하여, 비디오 데이터는 RGB 공간 컬러에 따라 정의될 수 있다는 것이 주목되어야 한다. 이 방식에서, 본원에 설명된 크로마 포맷들은 YUV 또는 RGB 컬러 공간 중 어느 하나에 적용할 수도 있다. RGB 크로마 포맷들은 통상적으로, 레드 샘플들의 수, 그린 샘플들의 수 및 블루 샘플들의 수가 동일하도록 샘플링된다. 따라서, 본원에 사용된 바와 같이 용어 "4:4:4 크로마 포맷" 은 YUV 컬러 공간 또는 RGB 컬러 공간 중 어느 하나를 지칭할 수도 있고, 여기서 샘플들의 수는 모든 컬러 컴포넌트들에 대해 동일하다.

[0058]

도 2a 내지 도 2c 는 비디오 데이터에 대한 상이한 샘플 포맷들을 예시하는 개념도들이다. 도 2a 는 4:2:0 샘플 포맷을 예시하는 개념도이다. 도 2a 에 예시된 바와 같이, 4:2:0 샘플 포맷에 대해서는, 크로마 컴포넌트들은 루마 컴포넌트의 사이즈의 4 분의 1 이다. 따라서, 4:2:0 샘플 포맷에 따라 포맷된 CU 에 대해, 크로마 컴포넌트의 모든 샘플에 대해 4 개의 루마 샘플들이 존재한다. 도 2b 는 4:2:2 샘플 포맷을 예시하는 개념도이다. 도 2b 에 예시된 바와 같이, 4:2:2 샘플 포맷에 대해서는, 크로마 컴포넌트들은 루마 컴포넌트의 사이즈의 2 분의 1 이다. 따라서, 4:2:2 샘플 포맷에 따라 포맷된 CU 에 대해, 크로마 컴포넌트의 모든 샘플에 대해 2 개의 루마 샘플들이 존재한다. 도 2c 는 4:4:4 샘플 포맷을 예시하는 개념도이다. 도 2c 에 예시된 바와 같이, 4:4:4 샘플 포맷에 대해서는, 크로마 컴포넌트들은 루마 컴포넌트와 동일한 사이즈이다. 따라서, 4:4:4 샘플 포맷에 따라 포맷된 CU 에 대해, 크로마 컴포넌트의 모든 샘플에 대해 1 개의 루마 샘플들이 존재한다.

[0059]

도 3 은 4:2:0 샘플 포맷에 따라 포맷된 16x16 코딩 유닛을 예시하는 개념도이다. 도 3 은 CU 내의 루마 샘플들에 대하여 크로마 샘플들의 상대적인 위치를 예시한다. 전술된 바와 같이, CU 는 통상적으로, 수평 및 수직 루마 샘플들의 수에 따라 정의된다. 따라서, 도 3 에 예시된 바와 같이, 4:2:0 샘플 포맷에 따라 포맷된 16x16 CU 는 루마 컴포넌트들의 16x16 샘플들 및 각각의 크로마 컴포넌트에 대한 8x8 샘플들을 포함한다. 또한, 전술된 바와 같이, CU 는 더 작은 CU들로 파티셔닝될 수도 있다. 예를 들어, 도 3 에 예시된 CU 는 4 개의 8x8 CU들로 파티셔닝될 수도 있고, 여기서 각각의 8x8 CU 는 루마 컴포넌트에 대한 8x8 샘플들 및 각각의 크로마 컴포넌트에 대한 4x4 샘플들을 포함한다.

- [0060] 도 4 는 4:2:2 샘플 포맷에 따라 포맷된 16x16 코딩 유닛의 예를 제시하는 개념도이다. 도 4 는 CU 내의 루마 샘플들에 대하여 크로마 샘플들의 상대적인 포지션을 예시한다. 전술된 바와 같이, CU 는 통상적으로, 수평 및 수직 루마 샘플들의 수에 따라 정의된다. 따라서, 도 4 에 예시된 바와 같이, 4:2:2 샘플 포맷에 따라 포맷된 16x16 CU 는 루마 컴포넌트들의 16x16 샘플들 및 각각의 크로마 컴포넌트에 대한 8x16 샘플들을 포함한다. 또한, 전술된 바와 같이, CU 는 더 작은 CU들로 파티셔닝될 수도 있다. 예를 들어, 도 4 에 예시된 CU 는 4 개의 8x8 CU들로 파티셔닝될 수도 있고, 여기서 각각의 CU 는 루마 컴포넌트에 대한 8x8 샘플들 및 각각의 크로마 컴포넌트에 대한 4x8 샘플들을 포함한다.
- [0061] 도 5 는 IBC 모드의 개념적 예시를 나타낸다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는, 예를 들어 IBC 모드를 사용하여 비디오 데이터의 블록들을 인코딩 및 디코딩하도록 구성될 수도 있다. 많은 애플리케이션들, 예컨대 원격 데스크톱, 원격 게이밍, 무선 디스플레이들, 자동 인포테인먼트, 클라우드 컴퓨팅 등은 사람의 일상 생활들에서 일상화되고 있고, 이러한 콘텐츠를 코딩할 때 코딩 효율성은 IBC 모드의 사용에 의해 개선될 수도 있다. 도 1 의 시스템 (10) 은 이들 애플리케이션들 중 임의의 것을 실행하도록 구성된 디바이스들을 나타낼 수도 있다. 이들 애플리케이션들에서 비디오 콘텐츠는 종종, 자연스러운 콘텐츠, 텍스트, 인공 그래픽들 등의 조합들이다. 비디오 프레임들의 텍스트 및 인공 그래픽 영역들에는, 반복된 패턴들 (예컨대, 문자들, 아이콘들, 심볼들 등) 이 종종 존재한다. 위에서 소개된 바와 같이, IBC 는, 이 종류의 리던던시를 제거하고 JCT-VC M0350 에 보고된 바와 같이 인트라-프레임 코딩 효율성을 잠재적으로 증가시키는 것을 가능하게 하는 전용 기법이다. 도 5 에 예시된 바와 같이, IBC 를 사용하는 CU들에 대해, 동일한 프레임 (예를 들어, 픽처) 에서 이미 복원된 영역으로부터 예측 신호들이 획득된다. 결국, 현재 CU 로부터 변위된 예측 신호의 포지션을 나타내는 오프셋 또는 블록 벡터는, 잔여 신호와 함께 인코딩된다.
- [0062] 예를 들어, 도 5 는 본 개시물의 기법들에 따른 IBC 모드에 따라 현재 픽처 (103) 내의 비디오 데이터의 현재 블록 (102) 을 예측하는 예시의 기법을 예시한다. 도 5 는 현재 픽처 (103) 내의 예측 비디오 블록 (104) 을 예시한다. 비디오 코더, 예를 들어 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 본 개시물의 기법들에 따른 IBC 모드에 따라 현재 비디오 블록 (102) 을 예측하기 위해 예측 비디오 블록 (104) 을 사용할 수도 있다.
- [0063] 비디오 인코더 (20) 는 비디오 데이터의 이전에 복원된 블록들의 세트로부터 현재 비디오 블록 (102) 을 예측하기 위해 예측 비디오 블록 (104) 을 선택한다. 비디오 인코더 (20) 는 인코딩된 비디오 비트스트림에 또한 포함되는 비디오 데이터를 역양자화 및 역변환하고, 비디오 데이터의 복원된 블록들을 예측하는데 사용된 예측 블록들과 결과의 잔여 블록들을 합산함으로써 비디오 데이터의 블록들을 복원한다. 도 5 의 예에서, "의도된 영역" 또는 "래스터 영역" 으로는 지칭될 수도 있는, 픽처 (103) 내의 의도된 영역 (108) 은 이전에 복원된 비디오 블록들의 세트를 포함한다. 비디오 인코더 (20) 는 이하에서 더 상세히 설명되는 바와 같이, 다양한 방식으로 픽처 (103) 내의 의도된 영역 (108) 을 정의할 수도 있다. 비디오 인코더 (20) 는, 의도된 영역 (108) 내의 다양한 비디오 블록들에 기초하여 현재 비디오 블록 (102) 을 예측 및 코딩하는 상대적인 효율성 및 정확도의 분석에 기초하여, 의도된 영역 (108) 내의 비디오 블록들 중에서부터 현재 비디오 블록 (102) 을 예측하도록 예측 비디오 블록 (104) 을 선택할 수도 있다.
- [0064] 의도된 영역 (108) 은 또한, IBC 예측 영역으로서 본 개시물에 지칭될 수도 있다. 본 개시물은 어느 블록들이 의도된 영역 (108) 에 포함되는지를 수정할 수도 있는 다양한 기법들을 설명한다. 따라서, 본 개시물의 기법들을 구현하는 경우, 의도된 영역 (108) 의 사이즈 및 형상은 도 5 의 예에 도시된 것과 상이할 수도 있다.
- [0065] 비디오 인코더 (20) 는 현재 비디오 블록 (102) 에 대한 예측 비디오 블록 (104) 의 로케이션 또는 변위를 나타내는 2-차원 벡터 (106) 를 결정한다. 오프셋 벡터의 예인 2-차원 벡터 (106) 는, 현재 비디오 블록 (102) 에 대한 예측 비디오 블록 (104) 의 수평 및 수직 변위를 각각 나타내는 수평 변위 컴포넌트 (112) 및 수직 변위 컴포넌트 (110) 를 포함한다. 비디오 인코더 (20) 는, 인코딩된 비디오 비트스트림에서, 2-차원 벡터 (106) 를 식별 또는 정의하는, 예를 들어 수평 변위 컴포넌트 (112) 및 수직 변위 컴포넌트 (110) 를 정의하는 하나 이상의 신택스 엘리먼트들을 포함할 수도 있다. 비디오 디코더 (30) 는 하나 이상의 신택스 엘리먼트들을 디코딩하여 2-차원 벡터 (106) 를 결정하고, 결정된 벡터를 사용하여 현재 비디오 블록 (102) 에 대한 예측 비디오 블록 (104) 을 식별할 수도 있다.
- [0066] 일부 예들에서, 2-차원 벡터 (106) 의 레졸루션은 정수 픽셀일 수 있다, 예를 들어 정수 픽셀 레졸루션을 갖도록 제한될 수 있다. 이러한 예들에서, 수평 변위 컴포넌트 (112) 및 수직 변위 컴포넌트 (110) 의 레졸루션은 정수 픽셀일 것이다. 이러한 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 현재 비디오 블록

(102)에 대한 예측자를 결정하기 위해 예측 비디오 블록 (104)의 픽셀 값들을 보간할 필요가 없다.

[0067] 다른 예들에서, 수평 변위 컴포넌트 (112) 및 수직 변위 컴포넌트 (110) 중 하나 또는 양자 모두의 레졸루션은 서브-픽셀일 수 있다. 예를 들어, 컴포넌트들 (112 및 110) 중 하나는 정수 픽셀 레졸루션을 가질 수도 있는 한편, 다른 컴포넌트는 서브-픽셀 레졸루션을 갖는다. 일부 예들에서, 수평 변위 컴포넌트 (112) 및 수직 변위 컴포넌트 (110)의 양자 모두의 레졸루션은 서브-픽셀일 수 있지만, 수평 변위 컴포넌트 (112) 및 수직 변위 컴포넌트 (110)는 상이한 레졸루션들을 가질 수도 있다.

[0068] 일부 예들에서, 비디오 코더, 예를 들어 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 특정 레벨, 예를 들어 블록-레벨, 슬라이스-레벨, 또는 픽처-레벨 적용에 기초하여 수평 변위 컴포넌트 (112) 및 수직 변위 컴포넌트 (110)의 레졸루션을 적응시킨다. 예를 들어, 비디오 인코더 (20)는, 수평 변위 컴포넌트 (112) 및 수직 변위 컴포넌트 (110)의 레졸루션이 정수 픽셀 레졸루션인지 또는 정수 픽셀 레졸루션이 아닌지 여부를 나타내는 플래그를 슬라이스 레벨에서, 예를 들어 슬라이스 헤더에서 시그널링할 수도 있다. 플래그가 수평 변위 컴포넌트 (112) 및 수직 변위 컴포넌트 (110)의 레졸루션이 정수 픽셀 레졸루션이 아니라는 것을 나타내면, 비디오 디코더 (30)는 그 레졸루션이 서브-픽셀 레졸루션이라고 추정할 수도 있다. 일부 예들에서, 반드시 플래그가 아닌 하나 이상의 신택스 엘리먼트들이 수평 변위 컴포넌트 (112) 및/또는 수직 변위 컴포넌트 (110)의 집합적 또는 개별적 레졸루션들을 나타내도록 비디오 데이터의 각각의 슬라이스 또는 다른 유닛에 대해 송신될 수도 있다.

[0069] 또 다른 예들에서, 플래그 또는 신택스 엘리먼트 대신에, 비디오 인코더 (20)는 레졸루션 콘텍스트 정보로부터의 수평 변위 컴포넌트 (112) 및/또는 수직 변위 컴포넌트 (110)의 레졸루션에 기초하여 설정할 수도 있고, 비디오 디코더 (30)는 레졸루션 콘텍스트 정보로부터 수평 변위 컴포넌트 (112) 및/또는 수직 변위 컴포넌트 (110)의 레졸루션을 추론할 수도 있다. 레졸루션 콘텍스트 정보는, 예를 들어, 컬러 공간 (예를 들어, YUV, RGB, 등), 특정 컬러 포맷 (예를 들어, 4:4:4, 4:2:2, 4:2:0, 등), 프레임 사이즈, 프레임 레이트, 또는 현재 비디오 블록 (102)을 포함하는 픽처들의 시퀀스 또는 픽처에 대한 양자화 파라미터 (QP)를 포함할 수도 있다. 적어도 일부 예들에서, 비디오 코더는 이전에 코딩된 프레임들 또는 픽처들에 관련된 정보에 기초하여 수평 변위 컴포넌트 (112) 및/또는 수직 변위 컴포넌트 (110)의 레졸루션을 결정할 수도 있다. 이 방식에서, 수평 변위 컴포넌트 (112)의 레졸루션 및 수직 변위 컴포넌트 (110)의 레졸루션은 미리 정의되고 시그널링될 수도 있고, 다른 사이트 정보 (예를 들어, 레졸루션 콘텍스트 정보)로부터 추론될 수도 있거나, 또는 이미 코딩된 프레임들에 기초할 수도 있다.

[0070] 현재 비디오 블록 (102)은 CU, 또는 CU의 PU일 수도 있다. 일부 예들에서, 비디오 코더, 예를 들어 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 IBC에 따라 예측되는 CU를 다수의 PU들로 스플릿할 수도 있다. 이러한 예들에서, 비디오 코더는 CU의 PU들 각각에 대해 각각의 (예를 들어, 상이한) 2-차원 벡터 (106)를 결정할 수도 있다. 예를 들어, 비디오 코더는 2Nx2N CU를 2개의 2NxN PU들, 2개의 Nx2N PU들, 또는 4개의 NxN PU들로 스플릿할 수도 있다. 다른 예들로서, 비디오 코더는 2Nx2N CU를 ((N/2)xN + (3N/2)xN) PU들, ((3N/2)xN + (N/2)xN) PU들, (Nx(N/2) + Nx(3N/2)) PU들, (Nx(3N/2) + Nx(N/2)) PU들, 4개의 (N/2)x2N PU들, 또는 4개의 2Nx(N/2) PU들로 스플릿할 수도 있다. 일부 예들에서, 비디오 코더는 2Nx2N PU를 사용하여 2Nx2N CU를 예측할 수도 있다.

[0071] 현재 비디오 블록 (102)은 루마 비디오 블록 (예를 들어, 루마 컴포넌트) 및 루마 비디오 블록에 대응하는 크로마 비디오 블록 (예를 들어, 크로마 컴포넌트)을 포함한다. 일부 예들에서, 비디오 인코더 (20)는 단지, 루마 비디오 블록들에 대한 2-차원 벡터들 (106)을 정의하는 하나 이상의 신택스 엘리먼트들을 인코딩된 비디오 비트스트림으로 인코딩할 수도 있다. 이러한 예들에서, 비디오 디코더 (30)는 루마 블록에 대해 시그널링된 2-차원 벡터에 기초하여 루마 블록에 대응하는 하나 이상의 크로마 블록들 각각에 대해 2-차원 벡터들 (106)을 도출할 수도 있다. 본 개시물에 설명된 기법들에서, 하나 이상의 크로마 블록들에 대한 2-차원 벡터들의 도출에서, 비디오 디코더 (30)는, 루마 블록에 대한 2-차원 벡터가 크로마 샘플 내의 서브-픽셀 포지션을 가리키면 루마 블록에 대한 2-차원 벡터를 수정할 수도 있다.

[0072] 컬러 포맷, 예를 들어 컬러 샘플링 포맷 또는 크로마 샘플링 포맷에 따라, 비디오 코더는 루마 비디오 블록에 대하여 대응하는 크로마 비디오 블록들을 다운샘플링할 수도 있다. 컬러 포맷 4:4:4는 다운샘플링을 포함하지 않는데, 이는 크로마 블록들이 루마 블록과 수평 및 수직 방향들에서 동일한 수의 샘플들을 포함한다는 것을 의미한다. 컬러 포맷 4:2:2는 수평 방향에서 다운샘플링되는데, 이는 루마 블록에 대하여 크로마 블록들에서 수평 방향으로 절반의 샘플들이 존재한다는 것을 의미한다. 컬러 포맷 4:2:0은 수평 및 수직 방향

들에서 다운샘플링되는데, 이는 루마 블록에 대하여 크로마 블록들에서 수평 및 수직 방향으로 절반의 샘플들이 존재한다는 것을 의미한다.

[0073] 비디오 코더들이 대응하는 루마 블록들에 대한 벡터들 (106) 에 기초하여 크로마 비디오 블록들에 대한 벡터들 (106) 을 결정하는 예들에서, 비디오 코더들은 루마 벡터를 수정할 필요가 있을 수도 있다. 예를 들어, 루마 벡터 (106) 가, 수평 변위 컴포넌트 (112) 및/또는 수직 변위 컴포넌트 (110) 가 홀수의 픽셀들인 정수 레졸루션을 갖고, 컬러 포맷이 4:2:2 또는 4:2:0 이면, 변환된 루마 벡터는 대응하는 크로마 블록에서의 정수 픽셀 로케이션을 가리키지 않을 수도 있다. 이러한 예들에서, 비디오 코더들은 크로마 벡터로서 사용하기 위해 루마 벡터를 스케일링하여 대응하는 크로마 블록을 예측할 수도 있다.

[0074] 설명된 바와 같이, 도 5 는 IBC 모드로 코딩되고 있는 현재의 CU 를 나타낸다. 현재 CU 에 대한 예측 블록은 검색 영역으로부터 획득될 수도 있다. 검색 영역은 현재 CU 와 동일한 프레임으로부터 이미 코딩된 블록들을 포함한다. 예를 들어, 프레임이 래스터 스캔 순서로 (즉, 좌측에서 우측으로 및 상부에서 하부로) 코딩되고 있다고 가정하면, 프레임의 이미 코딩된 블록들은 도 5 에 도시된 바와 같이 현재 CU 의 좌측 및 위에 있는 블록들에 대응한다. 일부 예들에서, 검색 영역은 프레임에 이미 코딩된 블록들 모두를 포함할 수도 있는 한편, 다른 예들에서 검색 영역은 이미 코딩된 블록들의 모두보다 적은 블록들을 포함할 수도 있다. 때때로 모션 벡터 또는 예측 벡터로서 지칭된 도 5 의 오프셋 벡터는, 현재 CU 의 상부-좌측 픽셀과 (도 5 에서 예측 신호로 라벨링된) 예측 블록의 상부-좌측 픽셀 간의 차이들을 식별한다. 따라서, 인코딩된 비디오 비트스트림에서 오프셋 벡터를 시그널링함으로써, 비디오 디코더는, 현재 CU 가 IBC 모드로 코딩되는 경우 현재 CU 에 대한 예측 블록을 식별할 수도 있다.

[0075] IBC 는 HEVC 에 대한 SCC 확장을 포함하는, SCC 의 다양한 구현들에 포함되어 있다. IBC 의 예는, 현재 CU/PU 가 현재 픽처/슬라이스의 이미 디코딩된 블록으로부터 예측되는 도 5 에 관하여 위에서 설명된다. IBC 에서, 예측 블록 (예를 들어, 도 5 의 예측 비디오 블록 (104)) 은 루프 필터링되지 않은, 예를 들어 디블록 필터링 또는 SAO 필터링되지 않은 복원된 블록일 수도 있다.

[0076] IBC 로 코딩되는 루마 컴포넌트 또는 크로마 컴포넌트들에 대해, 정수 블록 보상과 함께 블록 보상이 행해지고, 따라서 보간이 필요하지 않다. 따라서, 블록 벡터는 정수 레벨 정확도로 예측 및 시그널링된다.

[0077] SCC 의 현재 구현들에서, 블록 벡터 예측자는 각각의 CTB 의 시작에서 $(-w, 0)$ 로 설정되고, 여기서 w 는 CU 의 폭에 대응한다. 이러한 블록 벡터 예측자는, CU/PU 가 IBC 모드로 코딩되면, 최근에 코딩된 CU/PU 중 하나 이도록 업데이트된다. CU/PU 가 IBC 로 코딩되지 않으면, 블록 벡터 예측자는 변경되지 않은 채로 있다. 블록 벡터 예측 후에, HEVC 와 같은 MV 차이 (MVD) 코딩 방법을 사용하여 블록 벡터 차이가 인코딩된다.

[0078] IBC 의 현재 구현들은 CU 및 PU 레벨들 양자 모두에서 IBC 코딩을 가능하게 한다. PU 레벨 IBC 에 대해, $2N \times N$ 및 $N \times 2N$ PU 파티션들이 모든 CU 사이즈들에 대해 지원된다. 또한, CU 가 최소 CU 인 경우, $N \times N$ PU 파티션이 지원된다.

[0079] 위에서 소개된 바와 같이, HEVC 는 타일들 및 파면 병렬 프로세싱 (WPP) 을 포함하여, 코덱을 더욱 병렬-우호적으로 만들기 위해 여러 제안들을 포함한다. HEVC 는 타일의 코딩 트리 블록 래스터 스캔으로 연속적으로 오더링된, 하나의 컬럼 및 하나의 로우에서 함께-발생하는 코딩 트리 블록들의 정수로서 타일들을 정의한다. 각각의 픽처의 타일들로의 분할은 파티셔닝이다.

[0080] 도 6 은 타일들이 사용되는 경우 픽처의 래스터 스캔을 나타낸다. 픽처 내의 타일들은 도 6 에 도시된 바와 같이 픽처의 타일 래스터 스캔에서 연속적으로 오더링된다. 타일들의 수 및 그 경계들의 로케이션은 전체 시퀀스에 대해 정의되거나 픽처에서 픽처로 변경될 수 있다. 슬라이스 경계들과 유사한 타일 경계들은, 타일이 독립적으로 프로세싱될 수 있도록 파싱 및 예측 의존성들을 브레이크한다. 그러나, 인-루프 필터들 (디-블록킹 및 SAO) 은 여전히 타일 경계들을 교차할 수도 있다. HEVC 는 또한, 슬라이스들과 타일들 간의 관계에 대한 일부 제한들을 지정한다. 도 6 의 예에서, 라인들 (114A 및 114B) 은 수직 타일 경계들을 나타내고, 라인들 (114C 및 114D) 은 수평 타일 경계들을 나타낸다. 각각의 타일 내의 수들은 타일 내의 CTB들에 대한 래스터 스캔 순서를 나타낸다. 예를 들어, 상위 좌측의 가장 큰 타일에 대해, 0 으로 라벨링된 블록이 먼저 디코딩되고 그 후, 1 로 라벨링된 블록이, 그 후 2 로 라벨링된 블록이 디코딩되는 등등이다.

[0081] 타일들을 사용하는 잠재적인 이점은, 타일들이 엔트로피 디코딩 및 모션 보상 복원을 위해 비디오 디코더 (30) 와 같은 비디오 디코더의 프로세서들 간의 통신을 항상 요구하지 않는다는 것이다. 그러나, 이러한 통신은, 신텍스 엘리먼트 `loop_filter_across_tiles_enabled_flag` 가 1 로 설정되는 경우 필요할 수도 있다. 슬라

이스들과 비교하여, 타일들은 슬라이스들보다 잠재적인 더 높은 상관을 갖는 샘플들을 포함하는 픽처 파티션 형상들을 허용하기 때문에, 그리고 또한 타일들은 잠재적으로 슬라이스 헤더 오버헤드를 감소시키기 때문에 타일들은 잠재적으로 더 좋은 코딩 효율성을 갖는다.

[0082] HEVC 에서의 타일 설계는 여러 이점들을 제공할 수도 있다. 일 예로서, 타일들은 비디오 디코더 (30) 에 의한 병렬 프로세싱을 가능하게 할 수도 있다. 다른 예로서, 타일들은 슬라이스들의 사용과 비교하여 CTU들의 변경된 디코딩 순서를 허용함으로써 코딩 효율성을 개선시킬 수도 있지만, 주요 이점은 첫 번째 것이다. 타일이 단일-계층 코딩에서 사용되는 경우, 디코더가 병렬 디코딩 정보를 최대한으로 이용한다는 가정을 하고, 하나의 프로세싱 스레드에 의해 프로세싱될 루마 샘플들의 최대 수를 계산하도록 선택스 엘리먼트 min_spatial_segmentation_idc 가 디코더에 의해 사용될 수도 있다. HEVC 에는, 예를 들어 타일 또는 슬라이스 경계들 전체에 걸친 엔트로피 코딩 동기화 또는 디블록킹 필터링으로 인해, 상이한 스레드들 간의 동일한 픽처 인터-의존성들이 존재할 수도 있다. HEVC 는, 인코더들이 min_spatial_segmentation_idc 의 값을 가능한 최고 값으로 설정하도록 권장하는 메모를 포함한다.

[0083] 위에서 소개된 바와 같이, HEVC 는 또한, WPP 를 지원한다. WPP 가 인에이블되는 경우, 픽처의 각각의 CTU 로우는 분리된 파티션이다. 그러나, 슬라이스들 및 타일들과 비교하여, WPP 에서는 CTU 로우 경계들에서 어느 코딩 의존성들도 브레이크되지 않는다. 부가적으로, CABAC 확률들은 코딩 손실을 더욱 감소시키기 위해 이전 로우의 제 2 CTU 로부터 전파된다. 또한, WPP 는 규칙적인 래스터 스캔 순서를 변경하지 않는다. 의존성들이 깨지지 않기 때문에, WPP 비트스트림의 레이트-왜곡 손실은 비 병렬 비트스트림의 레이트-왜곡 손실에 비해 적을 수도 있다.

[0084] WPP 가 인에이블되는 경우, CTU 로우들의 수만큼의 프로세서들의 수가 CTU 로우들 (또는 라인들) 을 프로세싱하도록 병렬로 작동할 수 있다. 그러나, 파면 의존성들은 모든 CTU 로우들이 픽처의 시작에서 디코딩을 시작하는 것을 허용하지 않는다. 결과적으로, CTU 로우들은 또한, 픽처의 끝에서 동시에 디코딩을 완료할 수 없다. 이것은, 많은 수의 프로세서가 사용되는 경우, 더 분명해지는 병렬화 비효율성들을 도입한다.

[0085] 도 7 은 CTB들의 로우들을 병렬로 WPP 프로세스하는 예를 나타내고, 각각의 로우는 위의 로우의 제 2 CTB 를 프로세싱한 후에 이용 가능한 CABAC 확률들로 시작한다. 로우들 (116A-116G) 각각은 병렬로 디코딩될 수도 있지만, 각각의 로우가 위의 로우의 정보에 잠재적으로 의존하기 때문에, 모든 로우들의 디코딩은 동시에 시작할 수 없을 수도 있다. 예를 들어, 비디오 디코더 (30) 는, 로우 (116C) 의 블록들의 소정 수가 디코딩될 때까지 로우 (116D) 를 디코딩하기 시작할 수 없다. 유사하게, 비디오 디코더 (30) 는, 로우 (116D) 의 블록들의 소정 수가 이미 디코딩될 때까지 로우 (116E) 를 디코딩하기 시작할 수 없다. 이하에서 더 상세히 설명되는 바와 같이, 위의 로우를 디코딩하기 시작한 후 로우를 디코딩하기 전에, 비디오 디코더 (30) 가 대기하는 시간 양은 지연으로서 지칭될 수 있다. 도 7 의 예에서, 그레이 블록들은 이미 디코딩된 블록들 나타내는 한편, 화이트 블록들은 이제 디코딩될 블록들을 나타낸다. 도 7 에서 알 수 있는 바와 같이, 로우는 통상적으로 바로 아래의 로우보다 더 많은 이미 디코딩된 블록들을 갖는다.

[0086] 타일들 및 WPP 와 같은 병렬 프로세싱 기법들과 함께 IBC 모드로 비디오 데이터를 코딩하는 것은 잠재적인 어려움들을 제기할 수도 있다. IBC 모드는 예측을 위한 동일한 픽처 내에서 이전에 디코딩된 필터링되지 않은 샘플들을 사용한다. 현재 테스트 모델에서, IBC 모드에 대해 검색 범위는 제한되지 않고, 현재 픽처의 임의의 필터링되지 않은 디코딩된 샘플들 (풀 검색 IBC) 을 사용할 수 있다. 실-시간 애플리케이션들에서, 비-래스터 순서 (예를 들어, WPP) 로 프로세싱하여 다수의 CTU 를 동시에 프로세싱할 수 있도록 하는 것이 통상적이다. HEVC 는, WPP 또는 entropy_coding_sync_enabled_flag 가 인에이블되는 경우, 엔트로피 디코딩 순서를 정의한다.

[0087] WPP 및 타일들을 갖는 SCC 의 소정 구현들은 비-래스터 스캔이 인에이블되는 경우 일부 문제들을 갖는다. 제 1 예로서, 예측 샘플들의 이용 가능성은 항상, 래스터 순서 순차적 프로세싱에 기초하여 고려된다. 이것은, 시스템의 병렬 프로세싱 성능들에 잠재적으로 상당한 영향을 준다. 제 2 예로서, 최근 JCT-VC 회의에서, WPP 가 인에이블되는 경우, 엔트로피 파싱과 유사한 라인들에서 IBC 예측 영역을 제한하기 위한 일부 제안들이 있었다. 그러나, 이것은, IBC 모드에 대해 이용 가능한 제한된 검색 범위로 인해 코딩 효율성에 상당한 영향을 잠재적으로 갖는다.

[0088] WPP 형 병렬 프로세싱 스킴들을 허용하고 코딩 효율성 손실을 감소시키기 위해, 본 개시물은 IBC 검색 범위 및/또는 IBC 블록 벡터들에 일부 플렉서블한 제한들을 부가하기 위한 기법들을 설명한다.

- [0089] 다음의 기법들 각각은 별개로 또는 합동으로 적용될 수도 있다. 본 개시물에 설명된 기법들은 IBC 예측을 위한 예측 영역의 이용 가능성을 상세히 설명한다. 또한, 이 영역은 WPP 가 인에이블되는지 아닌지에 의존할 수도 있다.
- [0090] 제 1 기법에 따르면, 샘플들을 복원하는 고정된 프로세싱 순서 (예를 들어, WPP 가 인에이블되는 경우, 엔트로피 파싱 순서 또는 이하의 프로세싱 순서 중 임의의 것) 가 고려될 수 있다. 이미 디코딩/복원되는 샘플들은 단지 예측을 위해 사용될 수도 있다.
- [0091] 제 2 기법에 따르면, 샘플들을 복원하는 고정된 프로세싱 순서 (예를 들어, WPP 가 인에이블되는 경우, 엔트로피 파싱 순서 또는 이하의 프로세싱 순서 중 임의의 것) 가 고려될 수 있다. 이미 디코딩/복원되는 샘플들은 단지 예측을 위해 사용될 수도 있다. 또한, 현재 CTB 아래의 임의의 영역은 IBC 예측에 이용 가능하지 않은 것으로 고려된다.
- [0092] 제 3 기법에 따르면, 샘플들을 복원하는 고정된 프로세싱 순서 (예를 들어, WPP 가 인에이블되는 경우 엔트로피 파싱 순서 또는 아래의 프로세싱 순서 중 임의의 것) 가 고려될 수도 있고, 현재 CTB 아래의 임의의 영역은 IBC 예측에 이용 가능하지 않은 것으로 고려되며, 부분 영역들은 SPS 헤더에 지정된 최대 TU 사이즈에 기초하여 현재 CTB 위에 대해 이용 가능하지 않은 것으로 고려된다.
- [0093] 제 4 기법에 따르면, 샘플들을 복원하는 고정된 프로세싱 순서 (예를 들어, WPP 가 인에이블되는 경우 엔트로피 파싱 순서 또는 아래의 프로세싱 순서 중 임의의 것) 가 고려될 수도 있고, 부분 영역들은 SPS 헤더에 지정된 최대 TU 사이즈에 기초하여 현재 CTB 위에 대해 이용 가능하지 않은 것으로 고려된다.
- [0094] 제 5 기법에 따르면, 샘플들을 복원하는 플렉서블한 프로세싱 순서는 IBC 예측을 위해 유효한 것으로 고려될 수도 있고, 이 영역들은 비트스트림에서 시그널링된다.
- [0095] 제 6 기법에 따르면, 샘플들을 복원하는 플렉서블한 프로세싱 순서는 IBC 예측을 위해 유효한 것으로 고려될 수도 있고, 이 영역들은 비트스트림에서 시그널링되며, 현재 CTB 아래의 임의의 영역은 IBC 예측을 위해 이용 가능하지 않은 것으로 고려된다. 위에서 소개된 다양한 기법들의 예들이 이제, 더 상세히 설명될 것이다.
- [0096] 도 8 내지 도 12 는 WPP 를 사용하여 디코딩되도록 구성된 블록들의 예들을 나타낸다. 도 8 내지 도 12 에 도시된 영역들 각각은 CTB 에 대응한다. 도 8 내지 도 12 의 예들에서, X 로 라벨링된 CTB 는 디코딩되고 있는 현재 블록을 나타낸다. IBC 예측에 있어서, 1 로 라벨링된 블록들은 IBC 에 대해 사용될 수도 있는 한편, 0 으로 라벨링된 블록들은 IBC 에 대해 사용되지 않을 수도 있다.
- [0097] 도 8 은 위의 CTB 로우에 관하여 1 CTB 지연이 있는 IBC 프로세싱 순서가 이제 설명되는 제 1 예를 나타낸다. 도 8 에 도시된 바와 같은 다음의 제한은, IBC 블록 벡터들에 적용되어, IBC 블록이 0 으로 표시된 영역에서 예측하지 않도록 한다. 도 8 에서, "1들" 로 표시된 영역은 이미 복원되었던 유효한 예측 영역이다. 이들 제한들은 현재 블록 x 와 병렬로 임의의 "0" 표시된 영역의 프로세싱을 허용할 것이다.
- [0098] 8.4.4 인트라 블록 복사 예측 모드에서 블록 벡터 컴포넌트들에 대한 도출 프로세스
- [0099] - 비트스트림 적합성의 요건은,
- $$(xPb + bvIntra[xPb][yPb][0] + nPbSw - 1)/CtbSizeY - xCurr/CtbSizeY \leq yCurr/CtbSizeY - (yPb + bvIntra[xPb][yPb][1] + nPbSh - 1)/CtbSizeY$$
- [0100]
- [0101] 및
- $$yCurr/CtbSizeY - (yPb + bvIntra[xPb][yPb][1] + nPbSh - 1)/CtbSizeY \geq 0$$
- [0102] 이다.
- [0103] 본 개시물의 하나의 기법에 따르면, 인에이블된 파면 병렬 프로세싱으로 코딩된 하나 이상의 블록들에 대해, 비디오 디코더 (30) 는 CTB 들의 제 1 로우가 디코딩되기 시작하는 때와 CTB들의 제 1 로우 아래의 CTB들의 제 2 로우가 디코딩되기 시작하는 때 간의 지연을 식별하는 CTB 지연을 결정할 수도 있다. IBC 모드로 코딩된 그리고 디스에이블된 파면 병렬 프로세싱으로 코딩된 비디오 데이터의 현재 블록에 대해, 비디오 디코더 (30) 는 인에이블된 WPP 로 코딩된 하나 이상의 블록들에 대해 결정되었던 CTB 지연에 기초하여 현재 블록을 포함하는

픽처 내에서 현재 블록에 대한 IBC 예측 영역을 결정할 수도 있다. 다시 말해, 비디오 디코더 (30) 는, WPP 가 인에이블 또는 디스에이블되는지 또는 아닌지 여부에 관계없는 것을 의미하는, entropy_coding_sync_enabled_flag 의 값에 관계없이 CTB 지연에 기초하여 현재 블록에 대한 IBC 예측 영역을 결정할 수도 있다. CTB 지연에 기초하여 IBC 예측 영역을 항상 결정함으로써, 전체 디코딩 복잡성은 감소되지만, 비디오 디코더 (30) 가 인에이블된 WPP 로 IBC 모드를 여전히 지원할 수 있도록 하는 방식으로 감소된다.

[0104] 일부 대안의 구현들에서, IBC 블록이 비-디코딩된 영역으로부터 예측하지 않도록 entropy_coding_sync_enabled_flag 가 1 과 동일한 경우, 상기 제한은 IBC 블록 벡터들에만 적용될 수도 있다.

[0105] 8.4.4 인트라 블록 복사 예측 모드에서 블록 벡터 컴포넌트들에 대한 도출 프로세스

[0106] - 비트스트림 적합성의 요건은,

[0107] entropy_coding_sync_enabled_flag 가 1 과 동일한 경우,

[0108] $y_{Curr/CtbSizeY} - (y_{Pb} + bv_{Intra}[x_{Pb}][y_{Pb}][1] + n_{PbSh} - 1)/CtbSizeY \geq 0$ 이고,

[0109] entropy_coding_sync_enabled_flag 가 1 과 동일한 경우,

[0110] $(x_{Pb} + bv_{Intra}[x_{Pb}][y_{Pb}][0] + n_{PbSw} - 1)/CtbSizeY - x_{Curr/CtbSizeY} \leq y_{Curr/CtbSizeY} - (y_{Pb} + bv_{Intra}[x_{Pb}][y_{Pb}][1] + n_{PbSh} - 1)/CtbSizeY$ 이다.

[0111] 위의 CTB 로우에 관하여 2 CTB 지연이 있는 IBC 프로세싱 순서를 갖는 제 2 예가 이제 설명될 것이다. 도 9 에 도시된 바와 같은 다음의 제한은, IBC 블록 벡터들에 적용되어 IBC 블록이 "0" 영역에서 예측하지 않도록 한다. 도 9 에서, 1 들로 표시된 영역은 이미 복원되었던 유효한 예측 영역이다. 이들 제한들은 현재 블록 x 와 병렬로 임의의 "0" 표시된 영역의 프로세싱을 허용할 것이다. 여기서, 각각의 영역은 CTB 에 대응한다.

[0112] 8.4.4 인트라 블록 복사 예측 모드에서 블록 벡터 컴포넌트들에 대한 도출 프로세스

[0113] - 비트스트림 적합성의 요건은,

[0114] $(x_{Pb} + bv_{Intra}[x_{Pb}][y_{Pb}][0] + n_{PbSw} - 1)/CtbSizeY - x_{Curr/CtbSizeY} \leq 2 * (y_{Curr/CtbSizeY} - (y_{Pb} + bv_{Intra}[x_{Pb}][y_{Pb}][1] + n_{PbSh} - 1)/CtbSizeY)$

[0115] 및

[0116] $y_{Curr/CtbSizeY} - (y_{Pb} + bv_{Intra}[x_{Pb}][y_{Pb}][1] + n_{PbSh} - 1)/CtbSizeY \geq 0$ 이다.

[0117] 대안으로, IBC 블록이 비-디코딩된 영역으로부터 예측하지 않도록, entropy_coding_sync_enabled_flag 가 1 과 동일한 경우에만, 상기 제한 조건들은 IBC 블록 벡터들에 적용된다.

[0118] 8.4.4 인트라 블록 복사 예측 모드에서 블록 벡터 컴포넌트들에 대한 도출 프로세스

[0119] - 비트스트림 적합성의 요건은,

[0120] entropy_coding_sync_enabled_flag 가 1 과 동일한 경우,

[0121] $y_{Curr/CtbSizeY} - (y_{Pb} + bv_{Intra}[x_{Pb}][y_{Pb}][1] + n_{PbSh} - 1)/CtbSizeY \geq 0$ 이고,

[0122] entropy_coding_sync_enabled_flag 가 1 과 동일한 경우,

[0123] $(x_{Pb} + bv_{Intra}[x_{Pb}][y_{Pb}][0] + n_{PbSw} - 1)/CtbSizeY - x_{Curr/CtbSizeY} \leq 2 * (y_{Curr/CtbSizeY} - (y_{Pb} + bv_{Intra}[x_{Pb}][y_{Pb}][1] + n_{PbSh} - 1)/CtbSizeY)$ 이다.

[0124] 타일 형상의 영역들을 갖는 IBC 프로세싱 순서를 갖는 제 3 예가 이제 설명될 것이다. 도 10 에 도시된 바와 같은 다음의 제한은, IBC 블록 벡터들에 적용되어 IBC 블록이 "0" 영역에서 예측하지 않도록 한다. 도 10 에서, 1 들로 표시된 영역은 이미 복원되었던 유효한 예측 영역이다. 이들 제한들은 현재 블록 x 와 병렬로 임의의 "0" 표시된 영역을 프로세싱하는 것을 허용할 것이다. 여기서, 각각의 영역은 CTB 에 대응한다.

[0125] 일 예에서, 상기 제한은, 아래와 같이 entropy_coding_sync_enabled_flag 가 1 과 동일한 경우에만 적용된다.

[0126] 8.4.4 인트라 블록 복사 예측 모드에서 블록 벡터 컴포넌트들에 대한 도출 프로세스

[0127] ----

[0128] - 비트스트림 적합성의 요건은,

$$x_{Curr/CtbSizeY} - (x_{Pb} + bv_{Intra}[x_{Pb}][y_{Pb}][0] + n_{PbSw} - 1)/CtbSizeY \geq 0$$

$$y_{Curr/CtbSizeY} - (y_{Pb} + bv_{Intra}[x_{Pb}][y_{Pb}][1] + n_{PbSh} - 1)/CtbSizeY \geq 0$$

[0129] 이다.

[0130] 대안으로, IBC 블록이 비-디코딩된 영역으로부터 예측하지 않도록 entropy_coding_sync_enabled_flag 가 1 과 동일한 경우에만, 상기 제한 조건들은 IBC 블록 벡터들에 적용된다.

[0131] - entropy_coding_sync_enabled_flag 가 1 과 동일한 경우 비트스트림 적합성의 요건은,

$$x_{Curr/CtbSizeY} - (x_{Pb} + bv_{Intra}[x_{Pb}][y_{Pb}][0] + n_{PbSw} - 1)/CtbSizeY \geq 0$$

$$y_{Curr/CtbSizeY} - (y_{Pb} + bv_{Intra}[x_{Pb}][y_{Pb}][1] + n_{PbSh} - 1)/CtbSizeY \geq 0$$

[0132] 이다.

[0133] 위의 CTB 로우에 관하여 1 미만의 CTB 지연이 있는 IBC 프로세싱 순서를 갖는 제 4 예가 이제 설명될 것이다.

이 예에서는, 위에서 논의된 "위의 CTB 로우에 관하여 1 CTB 지연이 있는 IBC 프로세싱 순서" 와 유사하지만 대신에, 1 미만의 CTB 지연인, 최대 TU 블록의 지연을 갖는 IBC 에 대한 예측 샘플들을 제한하는 것이 제안된다. 최대 TU 사이즈 maxTUSizeY 는 $maxTUSizeY = 1 \ll MaxTbLog2SizeY$ 로서 도출된다. (SPS 에서 시그널링됨).

[0134] 8.4.4 인트라 블록 복사 예측 모드에서 블록 벡터 컴포넌트들에 대한 도출 프로세스

[0135] - 비트스트림 적합성의 요건은,

$$(x_{Pb} + bv_{Intra}[x_{Pb}][y_{Pb}][0] + n_{PbSw} - 1)/maxTUSizeY - x_{Curr}/maxTUSizeY \leq$$

$$y_{Curr}/maxTUSizeY -$$

$$(y_{Pb} + bv_{Intra}[x_{Pb}][y_{Pb}][1] + n_{PbSh} - 1)/maxTUSizeY$$

[0136] 및

$$y_{Curr}/maxTUSizeY -$$

$$(y_{Pb} + bv_{Intra}[x_{Pb}][y_{Pb}][1] + n_{PbSh} - 1)/maxTUSizeY \geq 0 \quad \text{이다.}$$

[0139] 대안으로, IBC 블록이 비-디코딩된 영역으로부터 예측하지 않도록 entropy_coding_sync_enabled_flag 가 1 과 동일한 경우에만, 상기 제한 조건들이 IBC 블록 벡터들에 적용된다.

[0140] 8.4.4 인트라 블록 복사 예측 모드에서 블록 벡터 컴포넌트들에 대한 도출 프로세스

[0141] - 비트스트림 적합성의 요건은,

[0142] entropy_coding_sync_enabled_flag 가 1 과 동일한 경우,

$$yCurr / \max TUSizeY - (yPb + bvIntra[xPb][yPb][1] + nPbSh - 1) / \max TUSizeY \geq 0$$

이고,

[0144] entropy_coding_sync_enabled_flag 가 1 과 동일한 경우,

$$(xPb + bvIntra[xPb][yPb][0] + nPbSw - 1) / \max TUSizeY - xCurr / \max TUSizeY \leq yCurr / \max TUSizeY - (yPb + bvIntra[xPb][yPb][1] + nPbSh - 1) / \max TUSizeY$$
 이다.

[0146] 화면 병렬 프로세싱은 픽처에서의 각각의 CTB 로우를 병렬 프로세싱하게 할 수 있다. 예를 들어, 1080p 픽처에서, 시스템이 17 개의 병렬 프로세싱 코어들을 갖는다면, 최대 17 CTB 로우들이 병렬로 프로세싱될 수 있다. 그러나, 대부분의 멀티-코어 시스템들에서는, 단지 제한된 수 (예를 들어, 4) 의 병렬 프로세싱 코어들이 사용되는 것이 통상적이다. 이 시나리오에서, 단지 4 개의 CTB 로우들이 병렬로 프로세싱되고, 5 번째 CTB 로우는, 위의 4 개의 CTB 로우들 중 하나의 완료 후에 프로세싱된다. 이러한 시나리오에서, 5 번째 CTB 로우는 이전의 4 개의 CTB 로우들로부터 이미 디코딩되는 영역들로부터 예측하는 것이 가능하다. 이 예에서, 각각의 CTB 로우에 대해, 모든 그 이전의 디코딩된 CTB 로우들에 대한 유효 디코딩 영역 (CTB's) 을 시그널링하는 것이 제안된다. 다른 실시형태에서, 각각의 CTB 로우에 대해, 모든 그 CTB 로우들에 대한 유효 디코딩된 영역 (CTB's) 를 시그널링하는 것이 제안된다. 이 정보는 SPS, VPS, PPS, 슬라이스 헤더 또는 그 각각의 확장에서 시그널링될 수 있다. 대안으로, 이 정보는 SEI 메시지로 시그널링될 수 있다.

[0147] 일 예에서, 아래의 정보는, entropy_coding_sync_enabled_flag 가 1 과 동일한 때에 기초하여 조건부로 시그널링된다. 대안으로, 아래의 정보는, entropy_coding_sync_enabled_flag 가 인에이블되는지 아닌지 여부와는 관계없이 시그널링된다.

[0148] 다른 예에서, 아래의 정보는, 적어도 하나의 파라미터 세트 (PPS, SPS, VPS) 또는 그 확장에서 시그널링될 수 있는 IBC 툴 인에이블링 플래그에 대해 조건부로 시그널링된다.

[0149] 제 1 예에서, 각각의 CTB 로우에 대해, 그 위의 CTB 의 로우들 모두에 대한 IBC 예측을 위한 이용 가능성 영역들이 시그널링된다. 예를 들어, 도 11 을 참조한다. 현재 CTB 아래의 영역들은 IBC 예측을 위해 이용 가능하지 않은 것으로 고려된다. 이하는, 슬라이스 헤더에서 제안된 방법의 예시의 구현이다.

pps_scc_extension () {	디스크립터
...	
if(intra_block_copy_enabled_flag){	
pps_ibc_ref_avail_signal_present_flag	
}	
...	
}	

[0150]

slice_segment_header() {	디스크립터
first_slice_segment_in_pic_flag	u(1)
if(tiles_enabled_flag entropy_coding_sync_enabled_flag) {	
num_entry_point_offsets	ue(v)
if(num_entry_point_offsets > 0) {	
offset_len_minus1	ue(v)
for(i = 0; i < num_entry_point_offsets; i++)	
entry_point_offset_minus1[i]	u(v)
}	
}	
if(pps_ibc_ref_avail_restriction_present_flag) {	
num_ctbY_in_slice_minus1	ue(v)
for(i=0;i< num_ctbY_in_slice;i++)	
for(j=0;j< i;i++)	
max_delay_IBCPred_in_CTBs[i][j]	uc(v)
}	
byte_alignment()	
}	

[0151]

[0152]

1 과 동일한 **pps_ibc_ref_avail_restriction_present_flag** 는, 인트라 블록 복사 레퍼런스 사용 제한들이 존재하고 슬라이스의 모든 코딩된 트리 블록들에 대해 슬라이스 세그먼트 헤더에서 시그널링된다는 것을 명시한다.

0 과 동일한 **pps_ibc_ref_avail_restriction_present_flag** 는, 인트라 블록 복사 레퍼런스 사용 정보가 슬라이스 세그먼트 헤더에 존재하지 않는다는 것을 명시한다. **pps_ibc_ref_avail_restriction_present_flag** 가 존재하지 않는 경우, 이것은 0 인 것으로 추론된다.

[0153]

num_ctbY_in_slice_minus1 플러스 1 은, 슬라이스에서의 CTB 로우들의 수를 명시한다.

[0154]

max_delay_IBCPred_in_CTBs[i][j] 는, IBC 예측을 위해 이용 가능한 각각의 이전의 CTB 로우 (j) 에 대한 CTB의 관점들에서 최대 지연을 현재의 CTB 로우 (i) 에 대해 명시한다. 존재하지 않는 경우, 그것은 로우에서 CTB 의 수와 동일한 것으로 추론된다.

[0155]

다른 예시의 구현에서, 각각의 CTB 로우에 대해, 모든 CTB들의 로우들에 대한 IBC 예측을 위한 이용 가능성 영역들이 시그널링된다. 예를 들어, 도 12 를 참조한다. 이하는, 슬라이스 헤더에서 제안된 방법의 예시의 구현이다.

slice_segment_header() {	디스크립터
first_slice_segment_in_pic_flag	u(1)
if(tiles_enabled_flag entropy_coding_sync_enabled_flag) {	
num_entry_point_offsets	ue(v)
if(num_entry_point_offsets > 0) {	
offset_len_minus1	uc(v)
for(i = 0; i < num_entry_point_offsets; i++)	
entry_point_offset_minus1[i]	u(v)
}	
}	
if(pps_ibc_ref_avail_restriction_present_flag) {	
num_ctbY_in_slice_minus1	ue(v)
for(i=0;i< num_ctbY_in_slice;i++)	
for(j=0;j< num_ctbY_in_slice;i++)	
max_delay_IBCPred_in_CTBs[i][j]	uc(v)
}	
byte_alignment()	
}	

[0156]

[0157]

1 과 동일한 **pps_ibc_ref_avail_restriction_present_flag** 는, 인트라 블록 복사 레퍼런스 사용 제한들이 존재하고 슬라이스의 모든 코딩된 트리 블록들에 대해 슬라이스 세그먼트 헤더에서 시그널링된다는 것을 명시한다.

0 과 동일한 **pps_ibc_ref_avail_restriction_present_flag** 는, 인트라 블록 복사 레퍼런스 사용 정보가 슬라이스 세그먼트 헤더에 존재하지 않는다는 것을 명시한다. **pps_ibc_ref_avail_restriction_present_flag** 가 존재하지 않는 경우, 이것은 0 인 것으로 추론된다.

[0158]

num_ctbY_in_slice_minus1 플러스 1 은, 슬라이스에서의 CTB 로우들의 수를 명시한다.

[0159]

max_delay_IBCPred_in_CTBs[i][j] 는, IBC 예측을 위해 이용 가능한 이전의 CTB 로우 (j) 각각에 대한 CTB 의 관점들에서 최대 지연을 현재 CTB 로우 (i) 에 대해 명시한다. 존재하지 않는 경우, 그것은 로우에서 CTB 의 수와 동일한 것으로 추론된다.

[0160]

대안으로, 이것은 SPS VUI 또는 SEI 메시지에서 시그널링되고 2014년 10월 7일자로 출원된 미국 가특허출원 제 62/061,063 호 및 JCTVC-S0145 에서 제안된 양태들과 결합될 수 있다.

[0161]

본 개시물의 다른 예시의 기법은 머지를 위한 IBC 모드의 시그널링에 관한 것이다. 본 개시물은 머지 모드에 대한 IBC 모드의 사용을 시그널링하기 위한 기법들을 도입한다. 제안된 방법들은 주로, 가능하게는 고 비트 심도 (8 보다 큰 비트) 의, 4:4:4, 4:2:2, 4:2:0, 4:0:0 등과 같은 상이한 크로마 샘플링 포맷의 지원을 포함하는, 스크린 콘텐츠 코딩에 관련된다.

[0162]

최근의 JCT-VC 회의에서, IBC 모드가 사용되는 경우 머지 후보 리스트 생성을 수정하기 위한 제안들이 있었다.

IBC 의 예측 특징이 인터와 상이한 것으로 관측됨에 따라, 머지 후보 리스트 생성 프로세스가 인터와 상이하게 수정되는 경우 코딩 효율성을 제공하는 것이 나타났다.

[0163]

도 13 은 IBC 머지 프로세스가 인터 머지 프로세서로부터 별개로 시그널링되는 예시의 시그널링 기법들을 나타낸다. 도 13 의 예에서, 비디오 디코더 (30) 는, PU 에 대해, PU 의 CU 가 인터-모드로 코딩되는지 (210) 를 나타내는 선택스 엘리먼트를 수신한다. CU 에 대한 **cu_skip_flag** 가 참이면 (212, 예), CU 는 스킵 모드로 코딩되고, 비디오 디코더 (30) 는 CU 에 대한 머지 인덱스 (214) 를 수신 또는 추론하고, 결정된 머지 인덱스와 연관된 모션 정보에 따라 CU 를 코딩한다. CU 에 대한 **cu_skip_flag** 가 거짓이면 (212, 아니오), 비디오 디코더 (30) 는 선택스 엘리먼트 **intra_bc_flag** (216) 및 **merge_flag** (218) 를 수신한다. **merge_flag** 가 참이면 (220, 예), 비디오 디코더 (30) 는 **merge_index** (214) 를 수신하고, **merge_index** 와 연관된 모션 정보에 따라 CU 를 디코딩한다. **merge_flag** 가 거짓이면 (220, 아니오), 비디오 디코더 (30) 는 다른 **intra_bc_flag** (224) 를 수신한다. **intra_bc_flag** 가 참이면 (224, 예), 비디오 디코더 (30) 는 블록 벡터

정보 (226) 를 수신한다. 수신된 블록 벡터 정보에 기초하여, 비디오 디코더 (30) 는 CU 를 디코딩한다. intra_bc_flag 가 거짓이면 (224, 아니오), 비디오 디코더 (30) 는 모션 벡터 정보 (228) 를 수신한다. 모션 벡터 정보에 기초하여, 비디오 디코더 (30) 는 CU 를 디코딩한다.

[0164] 도 13 에 설명된 시그널링의 스킴은 잠재적인 문제들을 갖는다. 일 예로서, 모든 PU 에 대한 intra_bc_flag 의 별개의 인터 및 IBC 모드로의 시그널링은 IBC 및 인터-모드들의 단일화에 대해 효율적이지 않을 수도 있다. 이 문제와 관련된 양태들/솔루션들은 2014년 10월 7일자로 출원된 미국 가특허출원 제 62/061,121 호, 2015년 10월 6일자로 출원된 미국 특허출원 제 14/876,699 호, 및 JCTVC-S0113 에서 커버된다. 인터 예측에 대하여 MVD 및 MVP 인덱스의 코딩과 BVD 및 BVP_idx 의 코딩을 정렬하는 것이 제안되어 있다.

[0165] 본 개시물의 하나의 기법에 따르면, 도 14 에 도시된 바와 같이 현재 PU 가 머지되는 (즉, merge_flag 가 1 인) 경우에만 intra_bc_flag 를 시그널링하는 것이 제안된다. 부가적으로, intra_bc_flag 는 적어도 하나의 파라미터 세트 (PPS, SPS, VPS) 또는 그 밖에 세트에서 시그널링될 수 있는 IBC 툴 인에이블 플래그에 대해 조건부로 시그널링될 수도 있다. 본 개시물은 머지 PU 에 대해서만 시그널링되는 intra_bc_flag 에 기초하여 인터 머지 프로세스로부터 IBC 머지 프로세스를 분리시키기 위한 기법들을 도입한다. 이러한 경우에서, 종래의 인터 예측 및 IBC 모드들에 대한 머지 후보 리스트들은 상이할 수도 있다. 신택스 엘리먼트 intra_bc_flag 는 다음의 경우들에서 1 과 같이 시그널링 및 추론되지 않을 수도 있다: (1) 현재 슬라이스가 I-슬라이스이고; (2) 현재 CU 사이즈가 8x8 이고 그 파티션 사이즈는 NxN 임.

[0166] 도 14 는 도 13 의 별개로 시그널링하는 것과 비교하여 IBC 머지 프로세스가 단지 한 번 시그널링되는 예시의 시그널링 기법들을 나타낸다. 도 14 의 예에서, 비디오 디코더 (30) 는 PU 에 대해, PU 의 CU 가 인터-모드로 코딩되는지 (230) 를 나타내는 신택스 엘리먼트들을 수신한다. CU 에 대한 cu_skip_flag 가 참이면 (232, 예), CU 는 스킵 모드로 코딩되고, 비디오 디코더 (30) 는 intra_bc_flag (234) 및 CU 에 대한 merge_index (214) 를 수신하고 결정된 merge_index 와 연관된 모션 정보에 따라 CU 를 코딩한다. CU 에 대한 cu_skip_flag 가 거짓이면 (232, 아니오), 비디오 디코더 (30) 는 merge_flag (240) 를 수신한다. merge_flag 가 참이면 (240, 예), 비디오 디코더 (30) 는 intra_bc_flag (234) 및 merge_index (236) 를 수신하고, merge_index 와 연관된 모션 정보에 따라 CU 를 디코딩한다. merge_flag 가 거짓이면 (240, 아니오), 비디오 디코더 (30) 는 가능하게는 IBC 블록 벡터 정보를 포함하는 모션 정보 (242) 를 수신하고, 모션 정보에 따라 CU 를 디코딩한다.

[0167] 도 15 는 본 개시물에서 설명된 기법들을 구현할 수도 있는 예시적인 비디오 인코더 (20) 를 예시하는 블록도이다. 비디오 인코더 (20) 는 비디오를 포스트-프로세싱 엔티티 (27) 로 출력하도록 구성될 수도 있다. 포스트-프로세싱 엔티티 (27) 는 비디오 엔티티의 예, 예컨대 비디오 인코더 (20) 로부터의 인코딩된 비디오 데이터를 프로세싱할 수도 있는 MANE 또는 스플라이싱/편집 디바이스를 표현하도록 의도된다. 일부 경우들에서, 포스트-프로세싱 엔티티는 네트워크 엔티티의 일 예일 수도 있다. 일부 비디오 인코딩 시스템들에서, 포스트-프로세싱 엔티티 (27) 및 비디오 인코더 (20) 는 별개의 디바이스들의 부분들일 수도 있지만, 다른 경우들에서 포스트-프로세싱 엔티티 (27) 에 대하여 설명된 기능성은 비디오 인코더 (20) 를 포함하는 동일한 디바이스에 의해 수행될 수도 있다. 일부 예들에서, 포스트-프로세싱 엔티티 (27) 는 도 1 의 저장 디바이스 (17) 의 예이다.

[0168] 비디오 인코더 (20) 는 비디오 슬라이스들 내에서 비디오 블록들의 인트라-, 인터-, 및 IBC 코딩을 수행할 수도 있다. 인트라-코딩은 주어진 비디오 프레임 또는 픽처 내의 비디오에서 공간적 리던던시를 감소시키거나 제거하기 위해 공간 예측에 의존한다. 인터-코딩은 비디오 시퀀스의 인접한 프레임들 또는 픽처들 내의 비디오에서의 시간적 리던던시를 감소시키거나 제거하기 위해 시간 예측에 의존한다. 인트라-모드 (I 모드) 는 여러 공간 기반의 압축 모드들 중 임의의 것을 지칭할 수도 있다. 단방향-예측 (P 모드) 또는 양-방향 예측 (B 모드) 과 같은 인터-모드들은 여러 시간 기반의 압축 모드들 중 임의의 것을 지칭할 수도 있다. 전술한 바와 같은 IBC 코딩 모드들은 비디오 데이터의 프레임으로부터 공간적 리던던시를 제거할 수 있지만, 전통적 인트라-모드들과 달리, IBC 코딩 코드들은 프레임 내의 더 큰 검색 영역에 예측 블록을 위치시키고, 인트라-예측 코딩 모드들에 의존하기보다는 오프셋 벡터를 사용하여 예측 블록들을 지칭하는데 사용될 수도 있다.

[0169] 도 15 의 예에서, 비디오 인코더 (20) 는 비디오 데이터 메모리 (33), 파티셔닝 유닛 (35), 예측 프로세싱 유닛 (41), 필터 유닛 (63), 디코딩된 픽처 버퍼 (64), 합산기 (50), 변환 프로세싱 유닛 (52), 양자화 유닛 (54), 및 엔트로피 인코딩 유닛 (56) 을 포함한다. 예측 프로세싱 유닛 (41) 은 모션 추정 유닛 (42), 모션 보상 유닛 (44), 인트라-예측 프로세싱 유닛 (46), 및 IBC 유닛 (48) 을 포함한다. 비디오 블록 복원을 위해, 비

디오 인코더 (20) 는 또한, 역양자화 유닛 (58), 역변환 프로세싱 유닛 (60), 및 합산기 (62) 를 포함한다. 필터 유닛 (63) 은 하나 이상의 루프 필터들, 예컨대 디블록킹 필터, 적응 루프 필터 (ALF), 및 샘플 적응 오프셋 (SAO) 필터를 나타내도록 의도된다. 필터 유닛 (63) 은 도 15 에서 루프 필터에 있는 것으로서 예시되었으나, 다른 구성들에서 필터 유닛 (63) 은 포스트 루프 필터로서 구현될 수도 있다.

[0170] 다양한 예들에서, 비디오 인코더 (20) 의 고정된 또는 프로그래머블 하드웨어 유닛은 본 개시물의 기법들을 수행하도록 태스크될 수도 있다. 또한, 일부 예들에서, 본 개시물의 기법들은 도 15 에 도시된 비디오 인코더 (20) 의 예시된 고정된 또는 프로그래머블 하드웨어 유닛들 중 하나 이상 사이에서 분배될 수도 있지만, 다른 디바이스들이 또한 본 개시물의 기법들을 수행할 수도 있다. 예를 들어, 도 15 의 예와 일치되어, 비디오 인코더 (20) 의 IBC 유닛 (48) 은 본 개시물의 기법들을 단독으로, 또는 비디오 인코더 (20) 의 다른 유닛들, 예컨대 모션 추정 유닛 (42), 모션 보상 유닛 (44), 인트라-예측 프로세싱 유닛 (46), 및 엔트로피 인코딩 유닛 (56) 과 결합하여 수행할 수도 있다. 일부 예들에서, 비디오 인코더 (20) 는 IBC 유닛 (48) 을 포함하지 않을 수도 있고, IBC 유닛 (48) 의 기능성은 모션 추정 유닛 (42) 및/또는 모션 보상 유닛 (44) 과 같은 예측 프로세싱 유닛 (41) 의 다른 컴포넌트들에 의해 수행될 수도 있다.

[0171] 비디오 데이터 메모리 (33) 는 비디오 인코더 (20) 의 컴포넌트들에 의해 인코딩될 비디오 데이터를 저장할 수도 있다. 비디오 데이터 메모리 (33) 에 저장된 비디오 데이터는, 예를 들어 비디오 소스 (18) 로부터 획득될 수도 있다. 디코딩된 픽처 버퍼 (64) 는, 예를 들어 인트라-, 인터-, 또는 IBC 코딩 모드들로, 비디오 인코더 (20) 에 의해 비디오 데이터를 인코딩하는데 있어서 사용하기 위해 레퍼런스 비디오 데이터를 저장하는 레퍼런스 픽처 메모리일 수도 있다. 비디오 데이터 메모리 (33) 및 디코딩된 픽처 버퍼 (64) 는 다양한 메모리 디바이스들 중 임의의 것, 예컨대 동기식 DRAM (SDRAM) 을 포함하는 동적 랜덤 액세스 메모리 (DRAM), 자기저항 RAM (MRAM), 저항성 RAM (RRAM), 또는 다른 유형들의 메모리 디바이스들에 의해 수행될 수도 있다. 비디오 데이터 메모리 (33) 및 디코딩된 픽처 버퍼 (64) 는 동일한 메모리 디바이스 또는 별개의 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에서, 비디오 데이터 메모리 (33) 는 비디오 인코더 (20) 의 다른 컴포넌트들과 함께 온-칩, 또는 이들 컴포넌트들에 대한 오프-칩일 수도 있다.

[0172] 도 15 에 도시된 바와 같이, 비디오 인코더 (20) 는 비디오 데이터를 수신하고, 이 비디오 데이터를 비디오 데이터 메모리 (33) 에 저장한다. 파티셔닝 유닛 (35) 은 데이터를 비디오 블록들로 파티셔닝한다. 이 파티셔닝은 또한, 슬라이스들, 타일들, 또는 다른 더 큰 유닛들로의 파티셔닝 뿐만 아니라, 예를 들어 LCU 들 및 CU 들의 쿼드트리 구조에 따른 비디오 블록 파티셔닝을 포함할 수도 있다. 비디오 인코더 (20) 는 일반적으로, 인코딩될 비디오 슬라이스 내의 비디오 블록들을 인코딩하는 컴포넌트들을 예시한다. 이 슬라이스는 다수의 비디오 블록들 (및 가능하게는 타일들로서 지칭된 비디오 블록들의 세트들) 로 분할될 수도 있다. 예측 프로세싱 유닛 (41) 은, 여러 결과들 (예를 들어, 코딩 레이트 및 왜곡 레벨) 에 기초하여 현재 비디오 블록에 대해, 복수의 가능한 코딩 모드들 중 하나, 예컨대 복수의 인트라-코딩 모드들 중 하나, 복수의 인터-코딩 모드들 중 하나, 또는 복수의 IBC 코딩 모드들 중 하나를 선택할 수도 있다. 예측 프로세싱 유닛 (41) 은 결과의 인트라-, 인터-, 또는 IBC 코딩된 블록을 합산기 (50) 에 제공하여 잔여 블록 데이터를 생성하고, 합산기 (62) 에 제공하여 레퍼런스 픽처로서 사용을 위해 인코딩된 블록을 복원할 수도 있다.

[0173] 예측 프로세싱 유닛 (41) 내의 인트라-예측 프로세싱 유닛 (46) 은 코딩된 현재 블록과 동일한 프레임 또는 슬라이스 내의 하나 이상의 이웃하는 블록들에 대한 현재 비디오 블록의 인트라 예측 코딩을 수행하여 공간적 압축을 제공할 수도 있다. 예측 프로세싱 유닛 (41) 내의 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 은 하나 이상의 레퍼런스 픽처들에서의 하나 이상의 예측 블록들에 대한 현재 비디오 블록의 인터-예측 코딩을 수행하여 시간적 압축을 제공할 수도 있다. 예측 프로세싱 유닛 (41) 내의 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 은 또한, 동일한 픽처에서 하나 이상의 예측 블록들에 대한 현재 비디오 블록의 IBC 코딩을 수행하여 공간적 보상을 제공할 수도 있다.

[0174] 모션 추정 유닛 (42) 은 비디오 시퀀스에 대해 미리결정된 패턴에 따라 비디오 슬라이스에 대한 인터-예측 모드 또는 IBC 모드를 결정하도록 구성될 수도 있다. 미리결정된 패턴은 시퀀스에서의 비디오 슬라이스들을 P 슬라이스들, B 슬라이스들 또는 GPB 슬라이스들로서 지정할 수도 있다. 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 은 고도로 통합될 수도 있지만, 개념적 목적들을 위해 별개로 예시되어 있다. 모션 추정 유닛 (42) 에 의해 수행된 모션 추정은 모션 벡터들을 생성하는 프로세스이며, 이 모션 벡터들은 비디오 블록들에 대한 모션을 추정한다. 모션 벡터는, 예를 들어 레퍼런스 픽처 내에서의 예측 블록에 대한 현재 비디오 프레임 또는 픽처 내에서의 비디오 블록의 PU 의 변위를 나타낼 수도 있다. IBC 코딩의 경우에서, IBC 에서 오프셋 벡터로도 지칭될 수도 있는 모션 벡터는 현재 비디오 프레임 내의 예측 블록에 대한 현재 비디오 프레임 또는

픽처 내의 비디오 블록의 PU 의 변위를 나타낼 수도 있다. IBC 유닛 (48) 은, 인터 예측을 위해 모션 추정 유닛 (42) 에 의한 모션 벡터들의 결정과 유사한 방식으로 IBC 코딩을 위해 벡터들, 예를 들어 블록 벡터들을 결정할 수도 있거나, 또는 모션 추정 유닛 (42) 를 이용하여 블록 벡터를 결정할 수도 있다.

[0175]

예측 블록은 픽셀 차이의 관점에서 코딩된 비디오 블록의 PU 와 밀접하게 일치하는 것으로 발견되는 블록이고, 픽셀 차이는 절대 차의 합 (SAD), 제곱 차의 합 (SSD), 또는 다른 차이 메트릭들에 의해 결정될 수도 있다. 일부 예들에서, 비디오 인코더 (20) 는 디코딩된 픽처 버퍼 (64) 에 저장된 레퍼런스 픽처들의 서브-정수 픽셀 포지션들에 대한 값들을 계산할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 레퍼런스 픽처의 1/4 픽셀 포지션들, 1/8 픽셀 포지션들, 또는 다른 분수의 픽셀 포지션들의 값들을 보간할 수도 있다. 따라서, 모션 추정 유닛 (42) 은 전픽셀 (full pixel) 포지션들 및 분수적 픽셀 포지션들에 대한 모션 검색을 수행하고 분수적 픽셀 정밀도를 갖는 모션 벡터를 출력한다.

[0176]

모션 추정 유닛 (42) 은 PU 의 포지션을 레퍼런스 픽처의 예측 블록의 포지션에 비교함으로써 인터-코딩된 슬라이스에서의 비디오 블록의 PU 에 대한 모션 벡터를 계산한다. 레퍼런스 픽처는 제 1 레퍼런스 픽처 리스트 (List 0) 또는 제 2 레퍼런스 픽처 리스트 (List 1) 로부터 선택될 수도 있는데, 이들 각각은 디코딩된 픽처 버퍼 (64) 에 저장된 하나 이상의 레퍼런스 픽처들을 식별한다. 모션 추정 유닛 (42) 은 계산된 모션 벡터를 엔트로피 인코딩 유닛 (56) 및 모션 보상 유닛 (44) 으로 전송한다.

[0177]

일부 예들에서, IBC 유닛 (48) 은 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 에 대하여 전송된 것과 유사한 방식으로 벡터들을 생성하고 예측 블록들을 폐지할 수도 있지만, 예측 블록들은 현재 블록과 동일한 픽처 또는 프레임에 있고 벡터들은 모션 벡터들과 대조적으로 블록 벡터들로서 지칭된다. 다른 예들에서, IBC 유닛 (48) 은 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 을 전체적으로 또는 부분적으로 사용하여, 본원에 설명된 기법들에 따라 IBC 예측을 위한 이러한 기능들을 수행할 수도 있다. 어느 경우에서나, IBC 에 대해, 예측 블록은, 픽셀 차이의 관점에서, 절대 차의 합 (SAD), 제곱 차의 합 (SSD), 또는 다른 차이 메트릭들에 의해 결정될 수도 있는, 코딩된 블록에 밀접하게 일치하도록 발견되는 블록일 수도 있고, 이 블록의 식별은 서브-정수 픽셀 포지션들에 대한 값들의 계산을 포함할 수도 있다.

[0178]

모션 보상 유닛 (44) 에 의해 수행된 모션 보상은 가능하게는 서브 픽셀 정확도로 보간을 수행함으로써, 모션 추정에 의해 결정된 모션 벡터에 기초한 예측 블록의 페치 (fetch) 또는 생성을 수반할 수도 있다. 보간 필터링은 알려진 픽셀 샘플들로부터 추가적인 픽셀 샘플들을 생성할 수도 있고, 따라서 비디오 블록을 코딩하는데 사용될 수도 있는 후보 예측 블록들의 수를 잠재적으로 증가시킨다. 현재 비디오 블록의 PU 에 대한 모션 벡터의 수신 시에, 모션 보상 유닛 (44) 은, 모션 벡터가 가리키는 예측 블록을 레퍼런스 픽처 리스트들 중 하나에, 또는 IBC 코딩의 경우에서 코딩되고 있는 픽처 내에 위치시킬 수도 있다. 비디오 인코더 (20) 는 코딩되고 있는 현재 비디오 블록의 픽셀 값들로부터 예측 블록의 픽셀 값을 감산함으로써 픽셀 차이 값들을 형성하는 잔여 비디오 블록을 형성한다. 픽셀 차이 값들은 블록에 대한 잔여 데이터를 형성하며, 루마 및 크로마 차이 컴포넌트들 양자를 포함할 수도 있다. 합산기 (50) 는 이 감산 연산을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다. 모션 보상 유닛 (44) 은 또한, 비디오 슬라이스의 비디오 블록들을 디코딩하는데 있어서 비디오 디코더 (30) 에 의한 사용을 위해 비디오 슬라이스 및 비디오 블록들과 연관된 신덱스 엘리먼트들을 생성할 수도 있다.

[0179]

인트라-예측 프로세싱 유닛 (46) 은, 전송된 바와 같이 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 에 의해 수행된 인터 예측 또는 IBC 유닛 (48) 에 의해 수행된 IBC 에 대한 대안으로서, 현재 블록을 인트라 예측할 수도 있다. 특히, 인트라-예측 프로세싱 유닛 (46) 은 현재 블록을 인코딩하는데 사용할 인트라 예측 모드를 결정할 수도 있다. 일부 예들에서, 인트라-예측 프로세싱 유닛 (46) 은, 예를 들어 별개의 인코딩 과정들 동안 다양한 인트라 예측 모드들을 사용하여 현재 블록을 인코딩할 수도 있고, 인트라-예측 프로세싱 유닛 (46) (또는, 일부 예들에서 모드 선택 유닛 (40)) 은 테스트된 모드들로부터 사용할 적합한 인트라 예측 모드를 선택할 수도 있다. 예를 들어, 인트라-예측 프로세싱 유닛 (46) 은 다양한 테스트된 인트라 예측 모드들에 대한 레이트 왜곡 분석을 사용하여 레이트 왜곡 값들을 계산하고, 테스트된 모드들 중에서 최선의 레이트 왜곡 특징들을 갖는 인트라 예측 모드를 선택할 수도 있다. 레이트 왜곡 분석은 일반적으로 인코딩된 블록과 원래의, 인코딩된 블록을 생성하기 위해 인코딩된 인코딩되지 않은 블록 간의 왜곡 (또는 오류) 의 양 뿐만 아니라 인코딩된 블록을 생성하는데 사용된 비트 레이트 (즉, 비트들의 수) 를 결정한다. 인트라-예측 프로세싱 유닛 (46) 은 왜곡들로부터의 비율들 및 다양한 인코딩된 블록들에 대한 레이트들을 계산하여 어느 인트라 예측 모드가 블록에 대한 최선의 레이트 왜곡 값을 보이는지를 결정할 수도 있다.

- [0180] 임의의 경우, 블록에 대한 인트라 예측 모드를 선택한 후에, 인트라-예측 프로세싱 유닛 (46) 은 블록에 대해 선택된 인트라 예측 모드를 나타내는 정보를 엔트로피 인코딩 유닛 (56) 에 제공할 수도 있다. 엔트로피 인코딩 유닛 (56) 은 본 개시물의 기법들에 따른 선택된 인트라 예측 모드를 나타내는 정보를 인코딩할 수도 있다. 비디오 인코더 (20) 는 송신된 비트스트림에서 구성 데이터를 포함할 수도 있는데, 이 구성 데이터는 복수의 인트라-예측 모드 인덱스 테이블들 및 복수의 수정된 인트라-예측 모드 인덱스 테이블들 (또한, 코드워드 맵핑 테이블들로서 지칭됨), 다양한 블록들에 대한 인코딩 컨텍스트들의 정의들, 및 가장 가능성 있는 인트라-예측 모드의 표시들, 인트라-예측 모드 인덱스 테이블, 및 컨텍스트들 각각에 대해 사용하기 위한 수정된 인트라-예측 모드 인덱스 테이블을 포함할 수도 있다.
- [0181] 예측 프로세싱 유닛 (41) 이 현재 비디오 블록에 대한 예측 블록을 인터 예측, 인트라 예측, 또는 IBC 를 통해 생성한 후에, 비디오 인코더 (20) 는 현재 비디오 블록에서 예측 블록을 감산함으로써 잔여 비디오 블록을 형성한다. 잔여 블록에서의 잔여 비디오 데이터는 하나 이상의 TU들에 포함되고 변환 프로세싱 유닛 (52) 에 적용될 수도 있다. 변환 프로세싱 유닛 (52) 은 변환, 예컨대 이산 코사인 변환 (DCT) 또는 개념적으로 유사한 변환을 사용하여 잔여 비디오 데이터를 잔여 변환 계수들로 변환한다. 변환 프로세싱 유닛 (52) 은 잔여 비디오 데이터를 픽셀 도메인에서 주파수 도메인과 같은 변환 도메인으로 변환할 수도 있다.
- [0182] 변환 프로세싱 유닛 (52) 은 결과의 변환 계수들을 양자화 유닛 (54) 으로 전송할 수도 있다. 양자화 유닛 (54) 은 변환 계수들을 양자화하여 비트 레이트를 더 감소시킬 수도 있다. 양자화 프로세스는 계수들의 일부 또는 전부와 연관된 비트 심도를 감소시킬 수도 있다. 양자화의 정도는 양자화 파라미터를 조정함으로써 수정될 수도 있다. 일부 예들에서, 양자화 유닛 (54) 은 그 후, 양자화된 변환 계수들을 포함하는 매트릭스의 스캔을 수행할 수도 있다. 대안으로, 엔트로피 인코딩 유닛 (56) 이 이 스캔을 수행할 수도 있다.
- [0183] 양자화 다음에, 엔트로피 인코딩 유닛 (56) 은 양자화된 변환 계수들을 엔트로피 인코딩한다. 예를 들어, 엔트로피 인코딩 유닛 (56) 은 컨텍스트 적응 이진 산술 코딩 (CABAC) 또는 다른 엔트로피 인코딩 방법론 또는 기법을 수행할 수도 있다. 엔트로피 인코딩 유닛 (56) 에 의한 엔트로피 인코딩 다음에, 인코딩된 비트스트림은 비디오 디코더 (30) 로 송신되거나, 또는 비디오 디코더 (30) 에 의한 추출 또는 나중의 송신을 위해 아카이브될 수도 있다. 엔트로피 인코딩 유닛 (56) 은 또한, 코딩되고 있는 현재 비디오 슬라이스에 대한 다른 신덱스 엘리먼트들 및 모션 벡터들을 엔트로피 인코딩할 수도 있다.
- [0184] 역양자화 유닛 (58) 및 역변환 프로세싱 유닛 (60) 은, 각각 역양자화 및 역변환을 적용하여, 레퍼런스 픽처의 레퍼런스 블록으로서 추후 사용을 위해 픽셀 도메인에서 잔여 블록을 복원한다. 모션 보상 유닛 (44) 은 레퍼런스 픽처 리스트들 중 하나 내의 레퍼런스 픽처들의 하나의 예측 블록에 잔여 블록을 가산함으로써 레퍼런스 블록을 계산할 수도 있다. 모션 보상 유닛 (44) 은 또한, 모션 추정에서 사용하기 위한 서브-정수 픽셀 값들을 계산하기 위해 복원된 잔여 블록에 하나 이상의 보간 필터들을 적용할 수도 있다. 보간 필터링은 알려진 픽셀 샘플들로부터 추가적인 픽셀 샘플들을 생성할 수도 있고, 따라서 비디오 블록을 코딩하는데 사용될 수도 있는 후보 예측 블록들의 수를 잠재적으로 증가시킨다. 합산기 (62) 는 복원된 잔여 블록을 모션 보상 유닛 (44) 에 의해 생성된 모션 보상된 예측 블록에 가산하여 디코딩된 픽처 버퍼 (64) 에 저장하기 위한 레퍼런스 픽처를 생성한다. 레퍼런스 블록은 후속하는 비디오 프레임 또는 픽처에서의 블록을 인터 예측하기 위한 레퍼런스 블록으로서 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 에 의해 사용될 수도 있다.
- [0185] 비디오 인코더 (20) 는 본 개시물의 기법들에 따라 비디오 데이터를 인코딩하도록 구성된 비디오 인코더의 예를 나타낸다. 예를 들어, 비디오 인코더 (20) 는 인에이블된 WPP 로 코딩된 하나 이상의 블록들에 대한 CTB 지연을 결정할 수도 있다. CTB 는, 예를 들어 CTB들의 제 1 로우가 디코딩되기 시작하는 때와 CTB들의 제 1 로우 아래의 CTB들의 제 2 로우가 디코딩되기 시작하는 때 간의 지연을 식별할 수도 있다. 인트라-블록 복사 (IBC) 모드로 코딩되고 디스에이블된 WPP 로 코딩된 비디오 데이터의 제 1 블록에 대해, 비디오 인코더 (20) 는 CTB 지연에 기초하여 제 1 블록에 대한 IBC 예측 영역을 결정할 수도 있다. 비디오 인코더 (20) 는 제 1 블록에 대한 IBC 예측 영역 내로부터, 제 1 블록에 대한 예측 블록을 식별하고, 예측 블록을 위시키기 위한 블록 벡터를 나타내도록 신덱스를 생성할 수도 있다.
- [0186] 도 16 은 본 개시물에서 설명된 기법들을 구현할 수도 있는 예시의 비디오 디코더 (30) 를 예시하는 블록도이다. 도 16 의 예에서, 비디오 디코더 (30) 는 비디오 데이터 메모리 (78), 엔트로피 디코딩 유닛 (80), 예측 프로세싱 유닛 (81), 역양자화 유닛 (86), 역변환 프로세싱 유닛 (88), 합산기 (90), 필터 유닛 (91), 및 디코딩된 픽처 버퍼 (DPB)(92) 를 포함한다. 예측 프로세싱 유닛 (81) 은 모션 보상 유닛 (82), 인트라-예측 프로세싱 유닛 (84), 및 IBC 유닛 (85) 을 포함한다. 비디오 디코더 (30) 는, 일부 예들에서

도 15 의 비디오 인코더 (20) 에 대해 설명된 인코딩 과정과 일반적으로 상반되는 디코딩 과정을 수행할 수도 있다.

[0187] 다양한 예들에서, 비디오 디코더 (30) 의 유닛은 본 개시물의 기법들을 수행하도록 태스크될 수도 있다. 또한, 일부 예들에서, 본 개시물의 기법들은 비디오 디코더 (30) 의 유닛들 중 하나 이상 사이에 분배될 수도 있다. 예를 들어, IBC 유닛 (85) 은 본 개시물의 기법들을 단독으로, 또는 비디오 디코더 (30) 의 다른 유닛들, 예컨대 모션 보상 유닛 (82), 인트라-예측 프로세싱 유닛 (84), 및 엔트로피 디코딩 유닛 (80) 과 결합하여 수행할 수도 있다. 일부 예들에서, 비디오 디코더 (30) 는 IBC 유닛 (85) 을 포함하지 않을 수도 있고, IBC 유닛 (85) 의 기능성은 모션 보상 유닛 (82) 과 같은 예측 프로세싱 유닛 (81) 의 다른 컴포넌트들에 의해 수행될 수도 있다.

[0188] 디코딩 프로세스 동안, 비디오 디코더 (30) 는 비디오 인코더 (20) 로부터, 비디오 데이터, 예를 들어 인코딩된 비디오 슬라이스의 비디오 블록들 및 연관된 신텍스 엘리먼트들을 표현하는 인코딩된 비디오 비트스트림을 수신한다. 비디오 디코더 (30) 는 네트워크 엔티티 (29) 로부터 비디오 데이터를 수신하고, 이 비디오 데이터를 비디오 데이터 메모리 (78) 에 저장할 수도 있다. 비디오 데이터 메모리 (78) 는 비디오 디코더 (30) 의 컴포넌트들에 의해 디코딩될 비디오 데이터, 예컨대 인코딩된 비디오 비트스트림을 저장할 수도 있다. 비디오 데이터 메모리 (78) 에 저장된 비디오 데이터는, 예를 들어 저장 디바이스 (17) 로부터, 예를 들어 카메라와 같은 로컬 비디오 소스로부터, 비디오 데이터의 유선 또는 무선 네트워크 통신을 통해, 또는 물리적 데이터 저장 매체에 액세스함으로써 획득될 수도 있다. 비디오 데이터 메모리 (78) 는 인코딩된 비디오 비트스트림으로부터 인코딩된 비디오 데이터를 저장하는 코딩된 픽처 버퍼를 형성할 수도 있다. 따라서, 도 16 에는 별개로 도시되었지만, 비디오 데이터 메모리 (78) 및 DPB (92) 는 동일한 메모리 디바이스 또는 별개의 메모리 디바이스들에 의해 제공될 수도 있다. 비디오 데이터 메모리 (78) 및 DPB (92) 는 다양한 메모리 디바이스들 중 임의의 것, 예컨대 동기식 DRAM (SDRAM) 을 포함하는 동적 랜덤 액세스 메모리 (DRAM), 자기저항 RAM (MRAM), 저항성 RAM (RRAM), 또는 다른 유형들의 메모리 디바이스들에 의해 수행될 수도 있다. 다양한 예들에서, 비디오 데이터 메모리 (78) 는 비디오 디코더 (30) 의 다른 컴포넌트들과 함께 온-칩, 또는 이들 컴포넌트들에 대한 오프-칩일 수도 있다.

[0189] 네트워크 엔티티 (29) 는, 예를 들어 서버, MANE, 비디오 에디터/스플라이서, 또는 전송된 기법들 중 하나 이상을 구현하도록 구성된 다른 이러한 디바이스일 수도 있다. 네트워크 엔티티 (29) 는 비디오 인코더 (20) 와 같은 비디오 인코더를 포함하거나 포함하지 않을 수도 있다. 본 개시물에 설명된 기법들 중 일부는, 네트워크 (29) 가 인코딩된 비디오 비트스트림을 비디오 디코더 (30) 로 송신하기 전에 네트워크 엔티티 (29) 에 의해 구현될 수도 있다. 일부 비디오 디코딩 시스템들에서, 네트워크 엔티티 (29) 및 비디오 디코더 (30) 는 별개의 디바이스들의 부분일 수도 있지만, 다른 경우들에서 네트워크 엔티티 (29) 에 대하여 설명된 기능성은 비디오 디코더 (30) 를 포함하는 동일한 디바이스에 의해 수행될 수도 있다. 네트워크 엔티티 (29) 는 일부 경우들에서 도 1 의 저장 디바이스 (17) 의 예일 수도 있다.

[0190] 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (80) 은 비트스트림을 엔트로피 디코딩하여 양자화된 계수들, 모션 벡터들, 및 다른 신텍스 엘리먼트들을 생성한다. 엔트로피 디코딩 유닛 (80) 은 모션 벡터들 및 다른 신텍스 엘리먼트들을 예측 프로세싱 유닛 (81) 으로 포워딩한다. 비디오 디코더 (30) 는 비디오 슬라이스 레벨 및/또는 비디오 블록 레벨에서 신텍스 엘리먼트들을 수신할 수도 있다.

[0191] 비디오 슬라이스가 인트라-코딩된 (I) 슬라이스로서 코딩되는 경우, 예측 프로세싱 유닛 (81) 의 인트라-예측 프로세싱 유닛 (84) 은 현재 프레임 또는 픽처의 이전에 디코딩된 블록들로부터의 데이터 및 시그널링된 인트라 예측 모드에 기초하여 현재 비디오 슬라이스의 비디오 블록에 대한 예측 데이터를 생성할 수도 있다. 비디오 프레임이 인터-코딩된 (즉, B 또는 P) 슬라이스로서 코딩되는 경우 또는 블록이 IBC 코딩되는 경우, 예측 프로세싱 유닛 (81) 의 모션 보상 유닛 (82) 은 엔트로피 디코딩 유닛 (80) 으로부터 수신된 다른 신텍스 엘리먼트들 및 모션 벡터들에 기초하여 현재 비디오 슬라이스의 비디오 블록에 대한 예측 블록들을 생성한다. 인터 예측을 위해, 예측 블록들은 레퍼런스 픽처 리스트들 중 하나 내의 레퍼런스 픽처들 중 하나로부터 생성될 수도 있다. 비디오 디코더 (30) 는, DPB (92) 에 저장된 레퍼런스 픽처들에 기초하여 디폴트 구성 기법들을 사용하여 레퍼런스 프레임 리스트들, 리스트 0 및 리스트 1 을 구성할 수도 있다. IBC 코딩을 위해, 예측 블록들은 예측되고 있는 블록과 동일한 픽처로부터 생성될 수도 있다.

[0192] 다른 예들에서, 비디오 블록이 본원에 설명된 IBC 모드에 따라 코딩되는 경우, 예측 프로세싱 유닛 (81) 의 IBC 유닛 (85) 은 엔트로피 디코딩 유닛 (80) 으로부터 수신된 다른 신텍스 엘리먼트들 및 블록 벡터들에 기초하여

현재 비디오 블록에 대한 예측 블록들을 생성한다. 예측 블록들은 비디오 인코더 (20) 에 의해 정의되고, DPB (92) 로부터 취출된 현재 비디오 블록과 동일한 픽처 내의 복원된 영역 내에 있을 수도 있다.

[0193] 모션 보상 유닛 (82) 및/또는 IBC 유닛 (85) 은 모션 벡터들 및 다른 신택스 엘리먼트들을 파싱함으로써 현재 비디오 슬라이스의 비디오 블록에 대한 예측 정보를 결정하고, 이 예측 정보를 사용하여 디코딩되고 있는 현재 비디오 블록에 대한 예측 블록들을 생성한다. 예를 들어, 모션 보상 유닛 (82) 은 수신된 신택스 엘리먼트들의 일부를 사용하여, 비디오 슬라이스의 비디오 블록들을 코딩하는데 사용된 예측 모드 (예를 들어, 인트라 예측 또는 인터 예측), 인터 예측 슬라이스 유형 (예를 들어, B 슬라이스 또는 P 슬라이스), 슬라이스에 대한 레퍼런스 픽처 리스트들 중 하나 이상에 대한 구성 정보, 슬라이스의 각각의 인터 인코딩된 비디오 블록에 대한 모션 벡터들, 슬라이스의 각각의 인터-코딩된 비디오 블록에 대한 인터 예측 스테이터스들, 및 현재 비디오 슬라이스에서 비디오 블록들을 디코딩하기 위한 다른 정보를 결정한다.

[0194] 유사하게, IBC 유닛 (85) 은 현재 비디오 블록이 IBC 모드를 사용하여 예측되었다고 결정하기 위해, 수신된 신택스 엘리먼트들의 일부, 예를 들어 플래그, 픽처의 비디오 블록이 복원된 영역 내에 있고 DPB (92) 에 저장되어야 한다는 것을 나타내는 구성 정보, 슬라이스의 각각의 IBC 예측된 비디오 블록에 대한 블록 벡터들, 슬라이스의 각각의 IBC 예측된 비디오 블록에 대한 IBC 예측 스테이터스 및 현재 비디오 슬라이스에서 비디오 블록들을 디코딩하기 위한 다른 정보를 사용할 수도 있다.

[0195] 모션 보상 유닛 (82) 은 또한, 보간 필터들에 기초하여 보간을 수행할 수도 있다. 모션 보상 유닛 (82) 은 비디오 블록들의 인코딩 동안 비디오 인코더 (20) 에 의해 사용되는 것과 같이 보간 필터들을 사용하여 레퍼런스 블록들의 서브-정수 픽셀들에 대한 보간된 값들을 계산할 수도 있다. 이 경우, 모션 보상 유닛 (82) 은 수신된 신택스 엘리먼트들로부터 비디오 인코더 (20) 에 의해 사용된 보간 필터들을 결정하고, 이 보간 필터들을 사용하여 예측 블록들을 생성할 수도 있다.

[0196] 역양자화 유닛 (86) 은 비트스트림에서 제공되고 엔트로피 디코딩 유닛 (80) 에 의해 디코딩된 양자화된 변환 계수들을 역양자화, 즉 양자화해제한다 (dequantizes). 역양자화 프로세스는 양자화의 정도, 및 마찬가지로 적용되어야 하는 역양자화의 정도를 결정하기 위해, 비디오 슬라이스의 각각의 비디오 블록에 대해 비디오 인코더 (20) 에 의해 계산된 양자화 파라미터의 사용을 포함할 수도 있다. 역변환 프로세싱 유닛 (88) 은, 픽셀 도메인에서 잔여 블록들을 생성하기 위해 변환 계수들에 대해 역변환, 예를 들어, 역 DCT, 역 정수 변환, 또는 개념적으로 유사한 역변환 프로세스를 적용한다.

[0197] 모션 보상 유닛 (82) 또는 IBC 유닛 (85) 이 모션 벡터들 및 다른 신택스 엘리먼트들에 기초하여 현재 비디오 블록에 대한 예측 블록을 생성한 후, 비디오 디코더 (30) 는 역변환 프로세싱 유닛 (88) 으로부터의 잔여 블록들을 모션 보상 유닛 (82) 에 의해 생성된 대응하는 예측 블록들과 합산함으로써 디코딩된 비디오 블록을 형성한다. 합산기 (90) 는 이 합산 연산을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다. 원한다면, (코딩 루프에서 또는 코딩 루프 후에) 루프 필터들이 또한 픽셀 천이들을 평활화하거나, 다르게는 비디오 품질을 개선시키는데 사용될 수도 있다. 필터 유닛 (91) 은 하나 이상의 루프 필터들, 예컨대 디블록킹 필터, 적응 루프 필터 (ALF), 및 샘플 적응 오프셋 (SAO) 필터를 나타내도록 의도된다. 필터 유닛 (91) 은 도 16 에서 루프 필터에 있는 것으로서 도시되었으나, 다른 구성들에서 필터 유닛 (91) 은 포스트 루프 필터로서 구현될 수도 있다. 그 후, 소정의 프레임 또는 픽처에서의 디코딩된 비디오 블록들은 DPB (92) 에 저장되고, 이 DPB (92) 는 후속의 모션 보상을 위해 사용된 레퍼런스 픽처들을 저장한다. DPB (92) 는 디스플레이 디바이스, 예컨대 도 1 의 디스플레이 디바이스 (32) 상의 추후의 프리젠테이션을 위해 디코딩된 비디오를 또한 저장하는 메모리의 부분일 수도 있거나, 또는 이러한 메모리로부터 분리될 수도 있다.

[0198] 비디오 디코더 (30) 는 본 개시물의 기법들에 따라 비디오 데이터를 디코딩하도록 구성된 비디오 디코더의 예를 나타낸다. 인에이블된 WPP 로 코딩된 하나 이상의 블록들에 대해, 비디오 디코더 (30) 는, CTB들의 제 1 로우가 디코딩되기 시작하는 때와 CTB들의 제 1 로우 아래의 CTB들의 제 2 로우가 디코딩되기 시작하는 때 간의 지연을 식별하는 CTB 지연을 결정한다. CTB 지연은, 예를 들어 CTB 들의 유닛들로 있을 수도 있다. IBC 모드로 코딩된 그리고 디스에이블된 파면 병렬 프로세싱으로 코딩된 비디오 데이터의 현재 블록에 대해, 비디오 디코더 (30) 는 인에이블된 WPP 로 코딩된 하나 이상의 블록들에 대해 결정되었던 CTB 지연에 기초하여 현재 블록을 포함하는 픽처 내에서 현재 블록에 대한 IBC 예측 영역을 결정할 수도 있다. 비디오 디코더 (30) 는, 현재 블록에 대한 결정된 IBC 예측 영역 내로부터, 현재 블록에 대한 예측 블록을 식별하고, 이 예측 블록에 기초하여 현재 블록을 IBC 디코딩한다. IBC 모드로 코딩되고 인에이블된 WPP 로 코딩된 비디오 데이터의 제 2 블록에 대해, 비디오 디코더 (30) 는 또한, CTB 지연에 기초하여 제 2 블록에 대한 IBC 예측 영역을 결정하고,

제 2 블록에 대한 IBC 예측 영역 내로부터, 제 2 블록에 대한 예측 블록을 식별할 수도 있으며; 예측 블록에 기초하여 현재 블록을 IBC 디코딩할 수도 있다.

[0199] 비디오 디코더 (30) 는, 예를 들어 선택스 엘리먼트의 값 (예를 들어, 전송된 `entropy_coding_sync_enabled_flag`) 에 기초하여 제 1 블록에 대해 WPP 가 디스에이블된다는 것을 결정할 수도 있다. 선택스 엘리먼트는, 예를 들어 콘텍스트 변수들에 대한 특성의 동기화 프로세스가 인코딩되는지를 나타내는 동기화 프로세스 인에이블링 선택스 엘리먼트일 수도 있다.

[0200] 제 1 블록에 대한 IBC 예측 영역은 이전에 디코딩된 필터링되지 않은 CTB들을 포함할 수도 있다. 부가적으로 또는 대안으로, IBC 예측 영역은 제 1 블록의 우측 및 제 1 블록 위의 적어도 2 이상의 로우들에 위치한 대각선으로 위치한 CTB 를 포함하고, 대각선으로 위치한 CTB 바로 아래의 CTB 를 제외시킬 수도 있다. 대각선으로 위치한 CTB 에 대해, 비디오 디코더 (30) 는 대각선으로 위치한 CTB 바로 아래의 CTB 를 제 1 블록과 병렬로 디코딩할 수도 있다.

[0201] 비디오 디코더 (30) 는, 비디오 데이터의 인코딩된 비트스트림으로, 비디오 데이터의 제 1 블록에 대한 코딩 모드가 IBC 모드라는 것을 나타내는 하나 이상의 선택스 엘리먼트들을 수신하고, 비디오 데이터의 인코딩된 비트스트림으로, 비디오 데이터의 제 1 블록에 대한 블록 벡터를 식별하는 하나 이상의 선택스 엘리먼트들을 수신할 수도 있다. 제 1 블록에 대한 IBC 예측 영역 내로부터, 제 1 블록에 대한 예측 블록을 식별하기 위해, 비디오 디코더 (30) 는 블록 벡터를 이용하여 예측 블록을 위치시킬 수도 있다.

[0202] 도 17 은 본 개시물의 기법들에 따른 비디오 데이터를 인코딩하기 위한 기법들을 예시하는 흐름도이다. 도 17 의 기법들은 일반적인 비디오 인코더를 참조하여 설명될 것이다. 일반적인 비디오 인코더는 비디오 인코더 (20) 의 특성들을 통합할 수도 있거나, 비디오 인코더의 상이한 구성일 수도 있다. 인에이블된 WPP 로 코딩된 하나 이상의 블록들에 대해, 비디오 인코더는 CTB 지연을 결정한다 (300). CTB 지연은, CTB들의 유닛들로, CTB들의 제 1 로우가 디코딩되기 시작하는 때와 CTB들의 제 1 로우 아래의 CTB들의 제 2 로우가 디코딩되기 시작하는 때 간의 지연을 식별한다. CTB 지연은, 예를 들어 하나의 CTB, 2 개의 CTB들, 또는 일부 다른 이러한 지연일 수도 있다. IBC 모드로 코딩되고 디스에이블된 WPP 로 코딩된 비디오 데이터의 제 1 블록에 대해, 비디오 인코더는 CTB 지연에 기초하여 제 1 블록에 대한 IBC 예측 영역을 결정한다 (302). 비디오 인코더는, 제 1 블록에 대한 IBC 예측 영역 내로부터, 제 1 블록에 대한 예측 블록을 식별하고 (304), 예측 블록을 위치시키기 위한 블록 벡터를 나타내도록 선택스를 생성한다 (306).

[0203] IBC 모드로 코딩되고 인에이블된 WPP 로 코딩된 비디오 데이터의 제 2 블록에 대해, 비디오 인코더는 CTB 지연에 기초하여 제 2 블록에 대한 IBC 예측 영역을 결정하고, 제 2 블록에 대한 IBC 예측 영역 내로부터, 제 2 블록에 대한 예측 블록을 식별할 수도 있다. IBC 예측 영역은 제 1 블록의 우측 및 제 1 블록 위의 적어도 2 이상의 로우들의 CTB 를 포함하고, 제 1 블록의 우측 및 제 1 블록 위의 적어도 2 이상의 로우들의 CTB 바로 아래의 CTB 를 제외시킬 수도 있다.

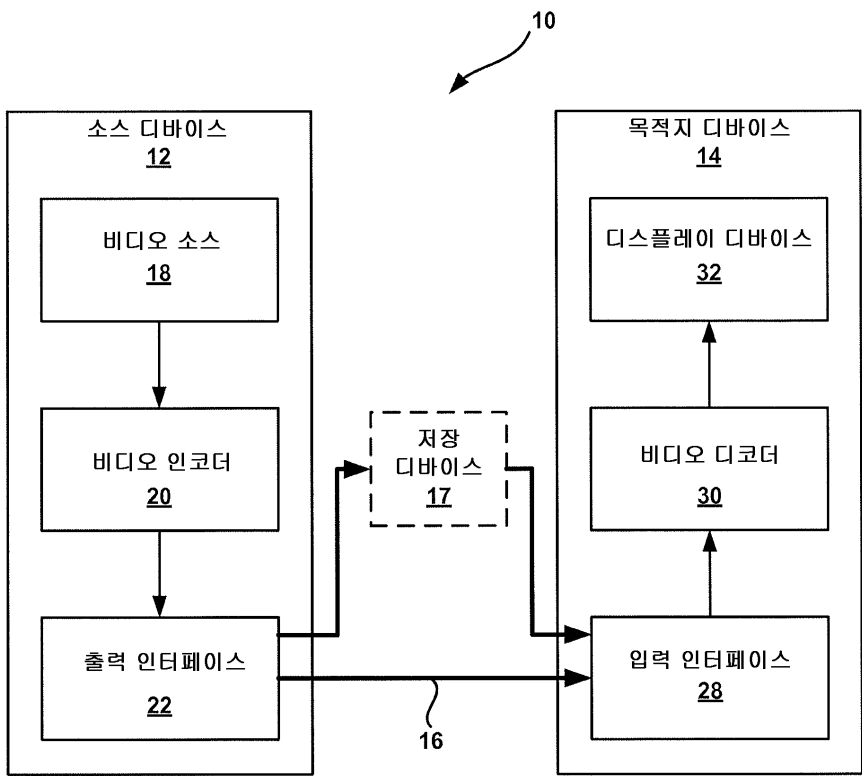
[0204] 도 18 은 본 개시물의 기법들에 따른 비디오 데이터를 디코딩하기 위한 기법들을 예시하는 흐름도이다. 도 18 의 기법들은 일반적인 비디오 디코더를 참조하여 설명될 것이다. 일반적인 비디오 디코더는 비디오 디코더 (30) 의 특성들을 통합할 수도 있거나, 상이한 구성의 비디오 디코더일 수도 있다. 인에이블된 WPP 로 코딩된 하나 이상의 블록들에 대해, 비디오 디코더는 CTB 지연을 결정한다 (310). CTB 지연은, CTB들의 유닛들로, CTB들의 제 1 로우가 디코딩되기 시작하는 때와 CTB들의 제 1 로우 아래의 CTB들의 제 2 로우가 디코딩되기 시작하는 때 간의 지연을 식별한다. CTB 지연은, 예를 들어, 하나의 CTB, 2 개의 CTB들, 또는 일부 다른 이러한 지연일 수도 있다. IBC 모드로 코딩되고 디스에이블된 WPP 로 코딩된 비디오 데이터의 제 1 블록에 대해, 비디오 디코더는 CTB 지연에 기초하여 제 1 블록에 대한 IBC 예측 영역을 결정한다 (312). 비디오 디코더는, 제 1 블록에 대한 IBC 예측 영역 내로부터, 제 1 블록에 대한 예측 블록을 식별한다 (314). 비디오 디코더 (30) 는 예측 블록에 기초하여 현재 블록을 IBC 디코딩한다 (316).

[0205] IBC 모드로 코딩되고 인에이블된 WPP 로 코딩된 비디오 데이터의 제 2 블록에 대해, 비디오 디코더는 CTB 지연에 기초하여 제 2 블록에 대한 IBC 예측 영역을 결정하고, 제 2 블록에 대한 IBC 예측 영역 내로부터, 제 2 블록에 대한 예측 블록을 식별한다. 비디오 디코더는 부가적으로, 선택스 엘리먼트를 수신하고 선택스 엘리먼트의 값에 기초하여 WPP 가 제 1 블록에 대해 디스에이블된다는 것을 결정할 수도 있다. 선택스 엘리먼트는, 콘텍스트 변수들에 대한 특정 동기화 프로세스가 인코딩되는지를 나타내는 예를 들어 동기화 프로세스 인에이블링 선택스 엘리먼트 (예를 들어, 전송된 `entropy_coding_sync_enabled_flag`) 일 수도 있다.

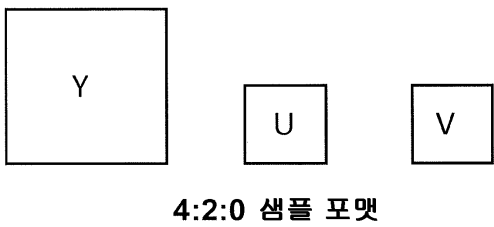
- [0206] 제 1 블록에 대한 IBC 예측 영역은, 예를 들어 이전에 디코딩된 필터링되지 않은 CTB들을 포함할 수도 있다. IBC 예측 영역은, 예를 들어 제 1 블록의 우측 및 제 1 블록 위의 적어도 2 이상의 로우들의 CTB를 포함하고, 제 1 블록의 우측 및 제 1 블록 위의 적어도 2 이상의 로우들의 CTB 바로 아래의 CTB를 제외시킬 수도 있다. 비디오 디코더는 제 1 블록의 우측 및 제 1 블록 위의 적어도 2 이상의 로우들의 CTB 바로 아래의 CTB를 제 1 블록과 병렬로 디코딩할 수도 있다.
- [0207] 비디오 디코더는 부가적으로, 비디오 데이터의 인코딩된 비트스트림으로, 비디오 데이터의 제 1 블록에 대한 코딩 모드가 IBC 모드라는 것을 나타내는 하나 이상의 신택스 엘리먼트들을 수신하고, 비디오 데이터의 제 1 블록에 대한 블록 벡터를 식별하는 하나 이상의 신택스 엘리먼트들을 비디오 데이터의 인코딩된 비트스트림으로 수신할 수도 있다. 제 1 블록에 대한 IBC 예측 영역 내로부터, 제 1 블록에 대한 예측 블록을 식별하기 위해, 비디오 디코더는 블록 벡터를 이용하여 예측 블록을 위치시킬 수도 있다.
- [0208] 하나 이상의 예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현되는 경우, 이 기능들은 하나 이상의 명령들 또는 코드로서 컴퓨터 판독 가능 매체 상에 저장되거나 송신될 수도 있고, 하드웨어 기반의 프로세싱 유닛에 의해 실행될 수도 있다.
- [0209] 컴퓨터 판독가능 매체는, 데이터 저장 매체와 같은 유형의 매체에 대응하는 컴퓨터 판독가능 저장 매체, 또는 예를 들어 통신 프로토콜에 따라, 한 곳에서 다른 곳으로 컴퓨터 프로그램의 전송을 용이하게 하는 임의의 매체를 포함하는 통신 매체를 포함할 수도 있다. 이 방식으로, 컴퓨터 판독가능 매체는 일반적으로 (1) 비일시적인 유형의 컴퓨터 판독가능 저장 매체 또는 (2) 신호 또는 반송파와 같은 통신 매체에 대응할 수도 있다. 데이터 저장 매체는 본 개시물에 설명된 기법들의 구현을 위한 명령들, 코드, 및/또는 데이터 구조들을 취출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 이용가능한 매체일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터 판독가능 매체를 포함할 수도 있다.
- [0210] 비제한적인 예로서, 이러한 컴퓨터 판독가능 저장 매체는 RAM, ROM, EEPROM, CD-ROM 또는 다른 광학 디스크 저장 디바이스, 자기 디스크 저장 디바이스 또는 다른 자기 저장 디바이스, 플래시 메모리, 또는 원하는 프로그램 코드를 명령들 또는 데이터 구조들의 형태로 저장하는데 사용될 수 있으며 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 접속이 컴퓨터 판독가능 매체라고 적절히 칭해진다. 예를 들어, 동축 케이블, 광섬유 케이블, 연선, 디지털 가입자 회선 (DSL), 또는 적외선, 무선, 및 마이크로파와 같은 무선 기술들을 사용하여 웹사이트, 서버, 또는 다른 원격 소스로부터 명령들이 송신되면, 동축 케이블, 광섬유 케이블, 연선, DSL, 또는 적외선, 무선, 및 마이크로파와 같은 무선 기술들은 매체의 정의 내에 포함된다. 그러나, 컴퓨터 판독가능 저장 매체 및 데이터 저장 매체는 접속들, 반송파들, 신호들, 또는 다른 일시적 매체들을 포함하지 않고, 대신에 비일시적인, 유형의 저장 매체들에 관련된다. 본원에서 사용된 디스크 (disk) 및 디스크 (disc)는, 콤팩트 디스크(CD), 레이저 디스크, 광학 디스크, 디지털 다기능 디스크 (DVD), 플로피 디스크, 및 블루레이 디스크를 포함하며, 여기서 디스크 (disk)들은 통상 자기적으로 데이터를 재생하는 반면, 디스크 (disc)들은 레이저들을 이용하여 광학적으로 데이터를 재생한다. 상기의 조합들이 또한, 컴퓨터 판독가능 매체의 범위 내에 포함되어야 한다.
- [0211] 명령들은, 하나 이상의 디지털 신호 프로세서 (DSP) 들, 범용 마이크로프로세서들, 주문형 집적 회로 (ASIC) 들, 필드 프로그래머블 로직 어레이 (FPGA) 들, 또는 다른 등가의 집적 또는 이산 로직 회로부와 같은, 하나 이상의 프로세서들에 의해 실행될 수도 있다. 따라서, 본원에서 사용되는 바와 같은 용어 "프로세서"는 상기의 구조 또는 본원에 설명된 기법들의 구현에 적합한 임의의 다른 구조 중 임의의 것을 지칭할 수도 있다. 또한, 일부 양태들에서, 본원에 설명된 기능성은 인코딩 및 디코딩을 위해 구성된 전용 하드웨어 및/또는 소프트웨어 모듈들 내에 제공될 수도 있고, 또는 결합형 코덱에 통합될 수도 있다. 또한, 본 기법들은 하나 이상의 회로들 또는 로직 엘리먼트들에서 완전히 구현될 수 있다.
- [0212] 본 개시물의 기법들은 무선 핸드셋, 집적 회로 (IC), 또는 IC 들의 세트 (예를 들어, 칩 세트)를 포함하는 광범위한 디바이스들 또는 장치들로 구현될 수도 있다. 개시된 기법들을 수행하도록 구성된 디바이스들의 기능적 양태를 강조하기 위해 다양한 컴포넌트들, 모듈들, 또는 유닛들이 본 개시물에서 설명되었지만, 반드시 상이한 하드웨어 유닛들에 의해 실행될 필요는 없다. 차라리, 전술한 바와 같이 다양한 유닛들은 적합한 소프트웨어 및/또는 펌웨어와 관련되어, 전술된 하나 이상의 프로세서들을 포함하여 상호 동작적인 하드웨어 유닛들의 집합에 의해 제공되거나 또는 코덱 하드웨어 유닛에 결합될 수도 있다.
- [0213] 다양한 예들이 설명되었다. 이들 및 다른 예들은 다음의 청구범위 내에 있다.

도면

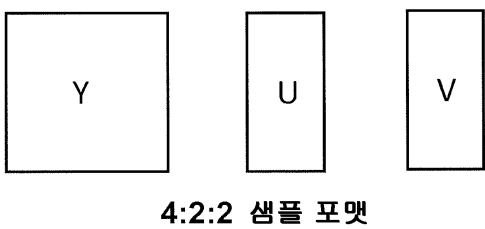
도면1



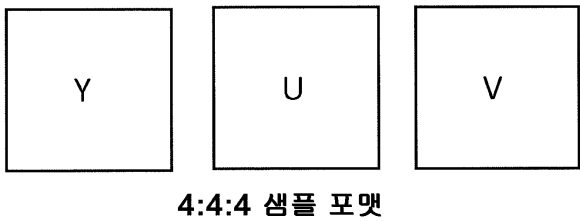
도면2a



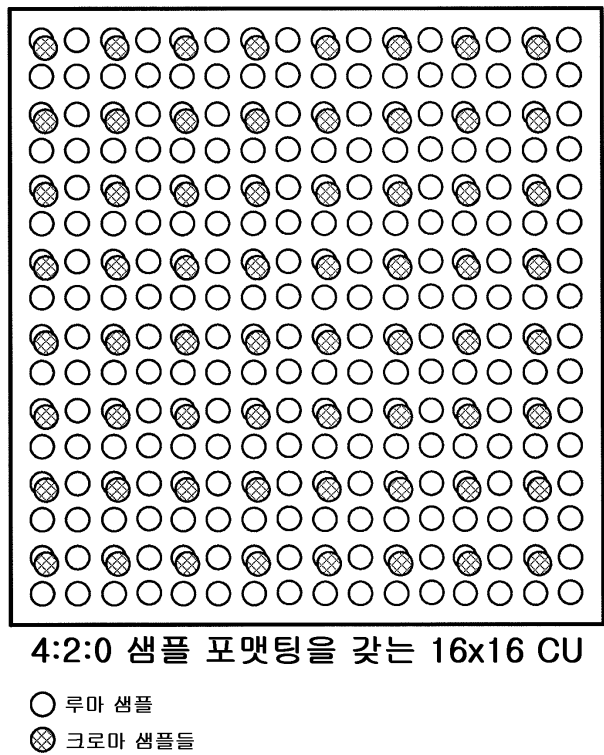
도면2b



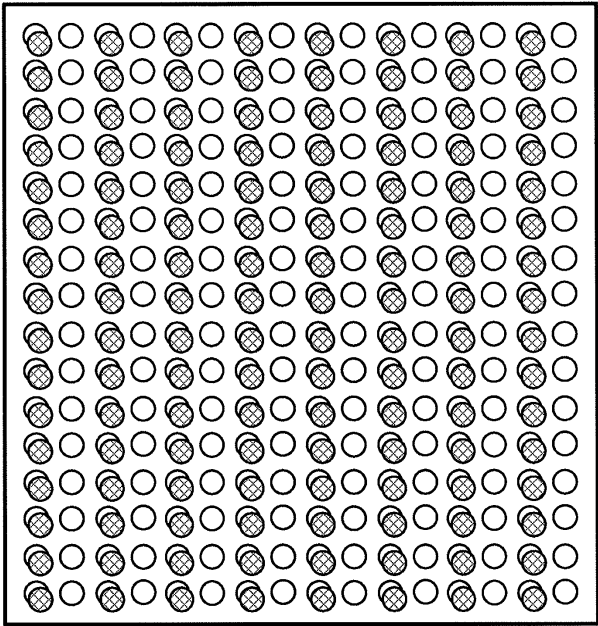
도면2c



도면3



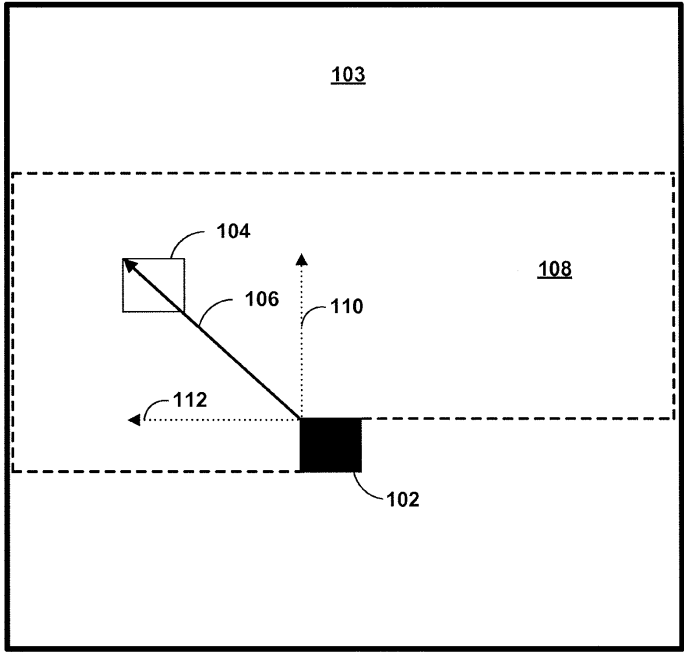
도면4



4:2:2 샘플 포매팅을 갖는 16x16 CU

- 루마 샘플
- ⊗ 크로마 샘플들

도면5



도면6

				114A			114B					
0	1	2	3	12	13	14	21	22	23	24		
4	5	6	7	15	16	17	25	26	27	28		
8	9	10	11	18	19	20	29	30	31	32		
33	34	35	36	41	42	43	47	48	49	50	114C	
37	38	39	40	44	45	46	51	52	53	54		
55	56	57	58	63	64	65	69	70	71	72	114D	
59	60	61	62	66	67	68	73	74	75	76		

도면7

116A											
116B											
116C											
116D											
116E											
116F											
116G											

도면8

1	1	1	1	1	1	0
1	1	1	1	1	0	0
1	1	1	X	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

도면9

1	1	1	1	1	1	1
1	1	1	1	1	1	0
1	1	1	X	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

도면10

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
1	1	X	0	0
1	1	1	0	0
1	1	1	0	0
1	1	1	0	0

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
1	1	1	X	0
1	1	1	1	0

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
1	1	1	X	0

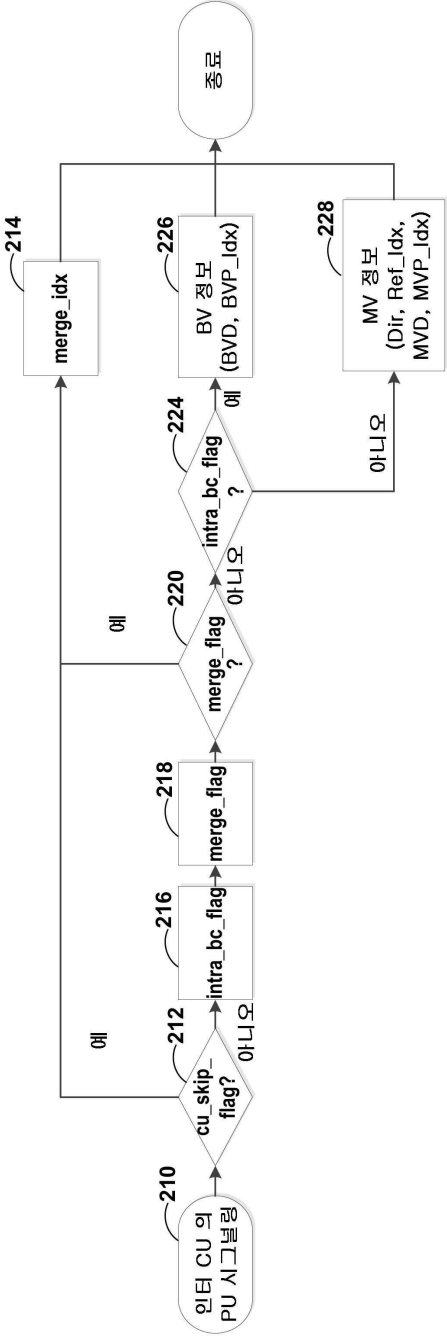
도면11

1	1	1	1	1	1	1
1	1	1	1	1	1	0
1	X	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

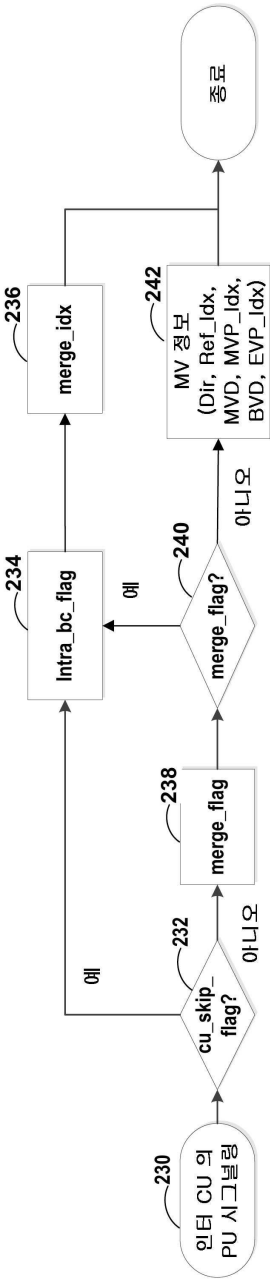
도면12

1	1	1	1	1	1	1
1	1	1	1	1	1	0
1	1	X	0	0	0	0
1	0	0	0	0	0	0
1	0	0	0	0	0	0

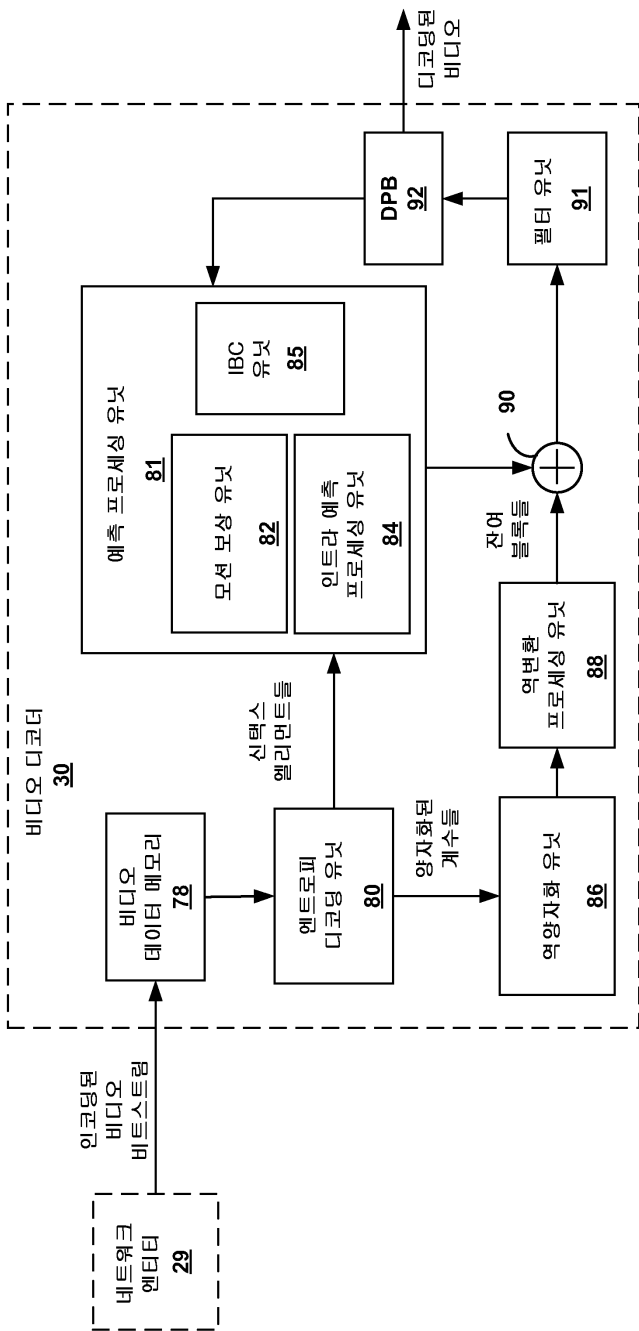
도면13



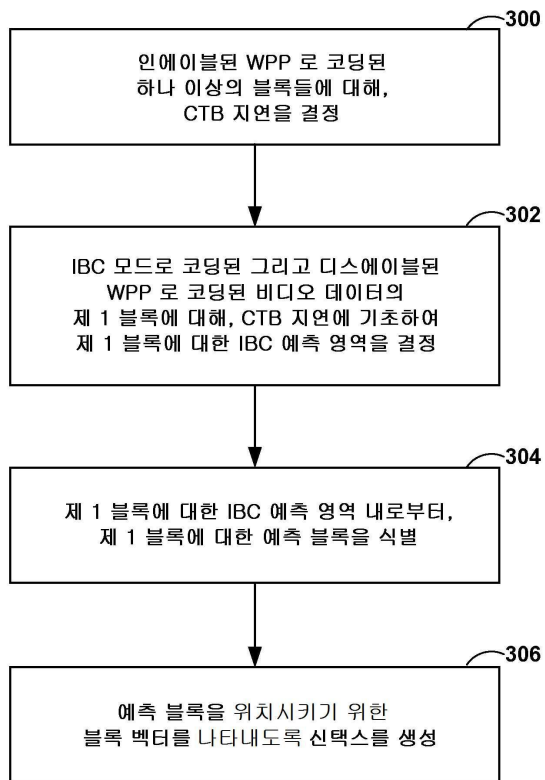
도면14



도면16



도면17



도면18

