(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2023년12월28일
(11) 등록번호 10-2619404
(24) 등록일자 2023년12월26일

(51) 국제특허분류(Int. Cl.)
H04N 19/117 (2014.01) *H04N 19/132* (2014.01)
H04N 19/186 (2014.01) *H04N 19/60* (2014.01)
H04N 19/82 (2014.01) *H04N 19/96* (2014.01)
(52) CPC특허분류
H04N 19/117 (2015.01)
H04N 19/132 (2015.01)
(21) 출원번호 10-2022-7018871
(22) 출원일자(국제) 2020년12월10일
심사청구일자 2023년04월19일
(85) 번역문제출일자 2022년06월03일
(65) 공개번호 10-2022-0107201
(43) 공개일자 2022년08월02일
(86) 국제출원번호 PCT/CN2020/135134
(87) 국제공개번호 WO 2021/115362
국제공개일자 2021년06월17일
(30) 우선권주장
PCT/CN2019/124481 2019년12월11일 중국(CN)
(56) 선행기술조사문헌
K. Misra, "CE5-related: On the design of CC-ALF", JVET-P1008, 2019.10.9
US20210152841 A1

(73) 특허권자
베이징 바이트댄스 네트워크 테크놀로지 컴퍼니, 리미티드
중국, 베이징 100041, 스징산 디스트릭트, 스싱 로드, 넘버 30, 넘버 3빌딩, 2층, 룸 비-0035
바이트댄스 아이엔씨
미국, 90066 캘리포니아 로스엔젤레스 스위트룸 137호 6층 제퍼슨 블러바드 웨스트 12655
(72) 발명자
왕 양
진르터우탸오 포스트 오피스 차이나 세틀라이트 커뮤니케이션 타워 63 지춘 로드 하이디안 디스트릭트 베이징 100080 중국
장 리
미국 90066 캘리포니아 로스엔젤레스 스위트룸 137호 6층 제퍼슨 블러바드 웨스트 12655
(뒷면에 계속)
(74) 대리인
성병기

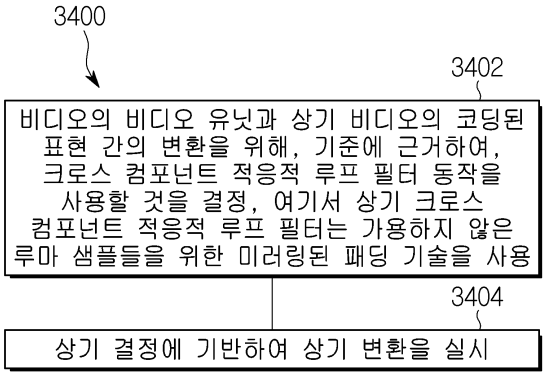
전체 청구항 수 : 총 15 항 심사관 : 김영태

(54) 발명의 명칭 **크로스 컴포넌트 적응적 루프 필터링을 위한 샘플 패딩**

(57) 요 약

비디오 처리를 위한 방법이 기술된다. 상기 방법은, 비디오의 비디오 유닛과 상기 비디오의 비트스트림 표현 간의 변환을 위해, 상기 비디오 유닛으로의 루프 필터링 툴 적용 동안 가용하지 않은 루마 샘플을 패딩하기 위한 미러링된 패딩 프로세스를 활성화할 것인지 여부를 결정하는 단계 그리고 상기 결정에 기반하여 상기 변환을 실시하는 단계를 포함한다.

대 표 도 - 도34



(52) CPC특허분류
H04N 19/186 (2015.01)
H04N 19/60 (2015.01)
H04N 19/82 (2015.01)
H04N 19/96 (2015.01)
(72) 발명자
리우 홍빈
진르터우탸오 포스트 오피스 차이나 세틀라이트 커뮤니케이션 타워 63 지춘 로드 하이디안 디스트릭트 베이징 100080 중국
장 카이
미국 90066 캘리포니아 로스엔젤레스 스위트룸 137호 6층 제퍼슨 블러바드 웨스트 12655
덩 즐핀
진르터우탸오 포스트 오피스 차이나 세틀라이트 커뮤니케이션 타워 63 지춘 로드 하이디안 디스트릭트 베이징 100080 중국
왕 유에
진르터우탸오 포스트 오피스 차이나 세틀라이트 커뮤니케이션 타워 63 지춘 로드 하이디안 디스트릭트 베이징 100080 중국

명 세 서

청구범위

청구항 1

비디오 데이터 처리 방법에 있어서,

비디오의 비디오 유닛과 상기 비디오의 비트스트림 간의 변환을 위해, 가상 경계(virtual boundary)가 상기 비디오 유닛에 적용되는 경우에 있어서 크로스 컴포넌트 적응적 루프 필터링(CC-ALF: cross-component adaptive loop filtering) 툴(tool)의 응용 동안, 상기 가상 경계로 인해 사용할 수 없는 가용하지 않은 루마 샘플(unavailable luma sample)을 패딩하기 위한 미러링된 패딩 프로세스 (mirrored padding process)가 상기 비디오 유닛에 사용되는 지를 결정하는 단계, 및

상기 결정에 기반하여 상기 변환을 실시하는 단계를 포함하되,

상기 미러링된 패딩 프로세스는 상기 가용하지 않은 루마 샘플의 해당 샘플(corresponding sample of the unavailable luma sample)을 패딩하고 상기 해당 샘플이 상기 가상 경계에 가용(available) 하더라도 상기 해당 샘플을 가용하지 않은 것으로 간주하는 것을 포함하고, 상기 해당 샘플은 상기 CC-ALF 툴의 필터 적용 영역(filter support region)내에 위치하며,

상기 가용하지 않은 루마 샘플은 상기 비디오 유닛의 M행에 위치하는 것으로 표시되고, 패딩될 상기 해당 샘플은 상기 비디오 유닛의 N행에 위치하는 것으로 표시되며,

d (C, M) = d (N, C) 이고, d (x, y)는 x행과 y행 간의 거리(distance)를 의미하고, C는 상기 비디오 유닛내의 대표 루마 샘플(representative luma sample)이 위치한 행을 나타내고, C, M, N은 정수이고, M은 N과 동일하지 않으며, 상기 대표 루마 샘플은 상기 비디오 유닛내 필터링될 크로마 샘플의 대응 루마 샘플(collocated luma sample)인, 비디오 데이터 처리 방법.

청구항 2

제1항에 있어서, 상기 크로마 샘플의 대응 루마 샘플의 위치는 상기 비디오의 칼라 포맷(color format)에 종속되는, 비디오 데이터 처리 방법.

청구항 3

제2항에 있어서, (x, y)에 위치한 상기 크로마 샘플의 대응 루마 샘플은, 칼라 포맷이 4:2:0 이면 (2x, 2y) 위치 또는 칼라 포맷이 4:2:2 이면 (2x, y) 위치 또는 칼라 포맷이 4:4:4 이면 (x, y) 위치 중 하나로 정의되는, 비디오 데이터 처리 방법.

청구항 4

제1항에 있어서, 상기 CC-ALF 툴은 K x L 필터 형태를 가지고, 여기서, K=4, L=3 이고, yM1, yL, yP1, yP2 는 상기 K x L 필터 형태에서 4개 샘플 행의 y축 (y-coordinates of four sample rows)이고, 여기서, yM1=-1, yL=0, yP1=1, yP2=2 이고, 상기 대표 루마 샘플은 yL=0 인 행이고, 상기 가상 경계는 상기 비디오 유닛내 CtbSizeY-4 행에 위치하며, CtbSizeY 는 코딩 트리 유닛 (CTU)인 상기 비디오 유닛의 크기를 나타내는, 비디오 데이터 처리 방법.

청구항 5

제4항에 있어서, 상기 가상 경계가 상기 대표 루마 샘플 위(above)에 있는 경우에, 상기 가용하지 않은 루마 샘

플은 상기 가상 경계 아래(below) 가장 근접한 사용 가능한 행을 이용하여 패딩되어 지고,

상기 해당 샘플(corresponding sample)은 상기 해당 샘플을 포함하는 행 위(above)의 가장 근접한 사용 가능한 행을 이용하여 패딩되어 지는, 비디오 데이터 처리 방법.

청구항 6

제5항에 있어서, yL 이 CtbSizeY-3 과 동일하고 yM1 위의 행이 가용하지 않은 경우, yP2 행내 해당 샘플 (x, yP2)은 yP1 행내의 샘플 (x, yP1) 을 이용하여 패딩되어 지는, 비디오 데이터 처리 방법.

청구항 7

제5항에 있어서, yL 이 CtbSizeY-4 와 동일하고 yM1 위(above)의 행 및 yM1 행이 가용하지 않은 경우, yM1 행내의 가용하지 않은 루마 샘플 (x, yM1)은 yL 행내의 샘플 (x, yL)을 이용하여 패딩되어 지고, yP1 행내의 해당 샘플 (x, yP1)은 yL 행내의 샘플 (x, yL)을 이용하여 패딩되어 지고, yP2 행내의 해당 샘플 (x, yP2)은 yL 행내의 샘플 (x, yL)을 이용하여 패딩되어 지는, 비디오 데이터 처리 방법.

청구항 8

제4항에 있어서, 상기 가상 경계가 상기 대표 루마 샘플 아래(below)에 있는 경우에, 상기 가용하지 않은 루마 샘플은 상기 가상 경계 위(above) 가장 근접한 사용 가능한 행을 이용하여 패딩되어 지고,

상기 해당 샘플(corresponding sample)은 상기 해당 샘플을 포함하는 행 아래(below)의 가장 근접한 사용 가능한 행을 이용하여 패딩되어 지는, 비디오 데이터 처리 방법.

청구항 9

제8항에 있어서, yL 이 CtbSizeY-6 과 동일하고 yP2 행이 가용하지 않은 경우, yP2 행내 상기 가용하지 않은 루마 샘플 (x, yP2)은 yP1 행내의 샘플 (x, yP1) 을 이용하여 패딩되어 지는, 비디오 데이터 처리 방법.

청구항 10

제8항에 있어서, yL 이 CtbSizeY-5 와 동일하고 yP2 및 yP1 행이 가용하지 않은 경우, yP2 행내의 가용하지 않은 루마 샘플 (x, yP2)은 yL 행내의 샘플 (x, yL)을 이용하여 패딩되어 지고, yP1 행내의 가용하지 않은 루마 샘플 (x, yP1)은 yL 행내의 샘플 (x, yL)을 이용하여 패딩되어 지고, yM1 행내의 해당 샘플 (x, yM1)은 yL 행내의 샘플 (x, yL)을 이용하여 패딩되어 지는, 비디오 데이터 처리 방법.

청구항 11

제1항에 있어서, 상기 변환은 상기 비디오를 상기 비트스트림으로 인코딩 하는 것을 포함하는, 비디오 데이터 처리 방법.

청구항 12

제1항에 있어서, 상기 변환은 상기 비트스트림을 상기 비디오로 디코딩 하는 것을 포함하는, 비디오 데이터 처리 방법.

청구항 13

프로세서와 명령어를 포함하는 비 일시적 메모리를 포함하는 비디오 데이터 처리 장치에 있어서, 상기 명령어는 상기 프로세서에 의해 실행될 때 상기 프로세서가,

비디오의 비디오 유닛과 상기 비디오의 비트스트림 간의 변환을 위해, 가상 경계(virtual boundary)가 상기 비디오 유닛에 적용되는 경우에 있어서 크로스 컴포넌트 적응적 루프 필터링(CC-ALF: cross-component adaptive loop filtering) 툴(tool)의 응용 동안, 상기 가상 경계로 인해 사용할 수 없는 가용하지 않은 루마 샘플(unavailable luma sample)을 패딩하기 위한 미러링된 패딩 프로세스 (mirrored padding process)가 상기 비디오 유닛에 사용되는 지를 결정하게 하고, 및

상기 결정에 기반하여 상기 변환을 실시하게 하되,

상기 미러링된 패딩 프로세스는 상기 가용하지 않은 루마 샘플의 해당 샘플(corresponding sample of the unavailable luma sample)을 패딩하고 상기 해당 샘플이 상기 가상 경계에 가용(available) 하더라도 상기 해당 샘플을 가용하지 않은 것으로 간주하는 것을 포함하고, 상기 해당 샘플은 상기 CC-ALF 툴의 필터 적용 영역(filter support region)내에 위치하며,

상기 가용하지 않은 루마 샘플은 상기 비디오 유닛의 M행에 위치하는 것으로 표시되고, 패딩될 상기 해당 샘플은 상기 비디오 유닛의 N행에 위치하는 것으로 표시되며,

d (C, M) = d (N, C) 이고, d (x, y)는 x행과 y행 간의 거리(distance)를 의미하고, C는 상기 비디오 유닛내의 대표 루마 샘플(representative luma sample)이 위치한 행을 나타내고, C, M, N은 정수이고, M은 N과 동일하지 않으며, 상기 대표 루마 샘플은 상기 비디오 유닛내 필터링될 크로마 샘플의 대응 루마 샘플(collocated luma sample)인, 비디오 데이터 처리 장치.

청구항 14

명령어가 저장된 비 일시적 컴퓨터 판독 가능한 저장 매체에 있어서, 상기 명령어는 프로세서가,

비디오의 비디오 유닛과 상기 비디오의 비트스트림 간의 변환을 위해, 가상 경계(virtual boundary)가 상기 비디오 유닛에 적용되는 경우에 있어서 크로스 컴포넌트 적응적 루프 필터링(CC-ALF: cross-component adaptive loop filtering) 툴(tool)의 응용 동안, 상기 가상 경계로 인해 사용할 수 없는 가용하지 않은 루마 샘플(unavailable luma sample)을 패딩하기 위한 미러링된 패딩 프로세스 (mirrored padding process)가 상기 비디오 유닛에 사용되는 지를 결정하게 하고, 및

상기 결정에 기반하여 상기 변환을 실시하게 하되,

상기 미러링된 패딩 프로세스는 상기 가용하지 않은 루마 샘플의 해당 샘플(corresponding sample of the unavailable luma sample)을 패딩하고 상기 해당 샘플이 상기 가상 경계에 가용(available) 하더라도 상기 해당 샘플을 가용하지 않은 것으로 간주하는 것을 포함하고, 상기 해당 샘플은 상기 CC-ALF 툴의 필터 적용 영역(filter support region)내에 위치하며,

상기 가용하지 않은 루마 샘플은 상기 비디오 유닛의 M행에 위치하는 것으로 표시되고, 패딩될 상기 해당 샘플은 상기 비디오 유닛의 N행에 위치하는 것으로 표시되며,

d (C, M) = d (N, C) 이고, d (x, y)는 x행과 y행 간의 거리(distance)를 의미하고, C는 상기 비디오 유닛내의 대표 루마 샘플(representative luma sample)이 위치한 행을 나타내고, C, M, N은 정수이고, M은 N과 동일하지 않으며, 상기 대표 루마 샘플은 상기 비디오 유닛내 필터링될 크로마 샘플의 대응 루마 샘플(collocated luma sample)인, 비 일시적 컴퓨터 판독 가능한 저장 매체.

청구항 15

비디오 처리 장치가 수행하는 방법에 의해 생성된 비디오의 비트스트림을 저장하는 비 일시적 컴퓨터 판독 가능한 기록 매체에 있어서, 상기 방법은,

가상 경계(virtual boundary)가 비디오 유닛에 적용되는 경우에 있어서 크로스 컴포넌트 적응적 루프 필터링(CC-ALF: cross-component adaptive loop filtering) 툴(tool)의 응용 동안, 상기 가상 경계로 인해 사용할 수 없는 가용하지 않은 루마 샘플(unavailable luma sample)을 패딩하기 위한 미러링된 패딩 프로세스 (mirrored padding process)가 상기 비디오 유닛에 사용되는 지를 결정하는 단계, 및

상기 결정에 기반하여 상기 비트스트림을 생성하는 단계를 포함하되,

상기 미러링된 패딩 프로세스는 상기 가용하지 않은 루마 샘플의 해당 샘플(corresponding sample of the unavailable luma sample)을 패딩하고 상기 해당 샘플이 상기 가상 경계에 가용(available) 하더라도 상기 해당 샘플을 가용하지 않은 것으로 간주하는 것을 포함하고, 상기 해당 샘플은 상기 CC-ALF 툴의 필터 적용 영역(filter support region)내에 위치하며,

상기 가용하지 않은 루마 샘플은 상기 비디오 유닛의 M행에 위치하는 것으로 표시되고, 패딩될 상기 해당 샘플은 상기 비디오 유닛의 N행에 위치하는 것으로 표시되며,

d (C, M) = d (N, C) 이고, d (x, y)는 x행과 y행 간의 거리(distance)를 의미하고, C는 상기 비디오 유닛내의 대표 루마 샘플(representative luma sample)이 위치한 행을 나타내고, C, M, N은 정수이고, M은 N과 동일하지 않으며, 상기 대표 루마 샘플은 상기 비디오 유닛내 필터링될 크로마 샘플의 대응 루마 샘플(collocated luma sample)인, 비 일시적 컴퓨터 판독 가능한 기록 매체.

청구항 16

삭제

청구항 17

삭제

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

청구항 21

삭제

청구항 22

삭제

청구항 23

삭제

청구항 24

삭제

청구항 25

삭제

청구항 26

삭제

청구항 27

삭제

청구항 28

삭제

청구항 29

삭제

청구항 30

삭제

청구항 31

삭제

청구항 32

삭제

청구항 33

삭제

청구항 34

삭제

청구항 35

삭제

청구항 36

삭제

청구항 37

삭제

청구항 38

삭제

청구항 39

삭제

청구항 40

삭제

청구항 41

삭제

청구항 42

삭제

청구항 43

삭제

청구항 44

삭제

청구항 45

삭제

청구항 46

삭제

청구항 47

삭제

청구항 48

삭제

청구항 49

삭제

청구항 50

삭제

청구항 51

삭제

청구항 52

삭제

청구항 53

삭제

청구항 54

삭제

청구항 55

삭제

청구항 56

삭제

청구항 57

삭제

청구항 58

삭제

청구항 59

삭제

청구항 60

삭제

청구항 61

삭제

청구항 62

삭제

청구항 63

삭제

청구항 64

삭제

청구항 65

삭제

청구항 66

삭제

청구항 67

삭제

청구항 68

삭제

발명의 설명

기 술 분 야

[0001] 본 출원은 2019년 12월 11일에 출원된 국제 특허 출원 번호 PCT/CN2019/124481 에 대한 우선권 및 혜택을 주장하는 2020년 12월 10일에 출원된 국제 특허 출원 번호 PCT/CN2020/135134 에 기반한다. 앞서 언급된 모든 특허 출원은 그 전체가 참조로서 본 명세서에서 통합된다.

[0002] 본 특허 문서는 비디오 및 영상 코딩 및 디코딩 기술들에 관한 것이다.

배 경 기 술

[0003] 디지털 비디오는 인터넷 및 기타 디지털 통신 네트워크들에서 가장 큰 대역폭을 사용한다. 비디오를 수신하고 디스플레이할 수 있는 접속된 사용자 디바이스의 수가 증가함에 따라, 디지털 비디오 사용을 위한 대역폭 수요가 계속 증가할 것으로 예상된다.

발명의 내용

해결하려는 과제

[0004] 본 문서는 비디오 인코딩 또는 디코딩 동안 크로스 컴포넌트 적응적 루프 필터링을 실시하기 위해 비디오 인코

더들 및 디코더들에 의해 사용될 수 있는 기술들을 개시한다.

과제의 해결 수단

[0005] 일 예시적 측면에서, 비디오 처리 방법이 개시된다. 상기 방법은, 비디오의 비디오 유닛과 상기 비디오의 비트스트림 표현 간의 변환을 위해, 상기 비디오 유닛으로의 루프 필터링 툴 적용 동안 가용하지 않은 루마 샘플을 패딩하기 위한 미러링된 패딩 프로세스를 활성화할 것인지 여부를 결정하는 단계 그리고 상기 결정에 기반하여 상기 변환을 실시하는 단계를 포함한다.

[0006] 다른 예시적 측면에서, 비디오 처리 방법이 개시된다. 상기 방법은, 비디오의 비디오 유닛과 상기 비디오의 비트스트림 표현 간의 변환을 위해, 상기 비디오 유닛의 코딩된 정보에 기반하여 가상 경계에 위치하는 샘플을 패딩하기 위해 반복적인 패딩 프로세스 및/또는 미러링된 패딩 프로세스를 적용할지 여부를 결정하는 단계 그리고 상기 결정에 기반하여 상기 변환을 실시하는 단계를 포함한다.

[0007] 또 다른 예시적 측면에서, 비디오 인코더 장치가 개시된다. 상기 비디오 인코더는 상기 기술된 방법들을 실시하도록 구성된 프로세서를 포함한다.

[0008] 또 다른 예시적 측면에서, 비디오 디코더 장치가 개시된다. 상기 비디오 디코더는 상기 기술된 방법들을 구현하도록 구성된 프로세서를 포함한다.

[0009] 또 다른 예시적 측면에서, 코드를 저장하는, 컴퓨터 판독가능한 매체가 개시된다. 상기 코드는 프로세서 실행가능한 코드의 형태로 여기에 기술된 상기 방법들 중 하나를 구현한다.

발명의 효과

[0010] 이러한 특징들 및 기타 특징들은 본 문서 전체에 걸쳐 기술된다.

도면의 간단한 설명

[0011] 도 1a는 픽처 내에서 4:2:2 루마 및 크로마 샘플들의 공칭 수직 및 수평 위치들을 도시한다.

도 2는 인코더 블록도의 일 예이다.

도 3은 12개의 타일과 3개의 래스터 스캔 슬라이스로 분할되는 18 x 12 루마 코딩 트리 유닛들(CTU들)을 갖는 픽처를 도시한다.

도 4는 24개의 타일과 9개의 직사각형 슬라이스로 분할되는 18 x 12 CTU들을 갖는 픽처를 도시한다.

도 5는 4개의 타일과 4개의 직사각형 슬라이스로 분할되는 픽처를 도시한다.

도 6은 15개의 타일과 24개의 직사각형 슬라이스 및 24개의 서브픽처로 분할되는 픽처를 도시한다.

도 7A 내지 도 7C에서, 도 7A는 하단 픽처 경계를 가로지르는 CTB들을 도시하고, 도 7B는 우측 픽처 경계를 가로지르는 CTB들을 도시하고, 도 7C는 우하단 픽처 경계를 가로 지르는 CTB들을 도시한다.

도 8은 픽처 샘플들, 8x8 그리드 상의 수평 및 수직 블록 경계들, 그리고 병렬 디블로킹될 수 있는 8x8 샘플들의 비중첩 블록들을 도시한다.

도 9는 필터 온/오프 결정 및 강/약 필터 선택에 관련된 픽셀들의 예를 도시한다.

도 10은 EO 샘플 분류를 위한 4 개의 1-D 방향성 패턴, 즉 수평(EO 클래스 = 0), 수직(EO 클래스 = 1), 135° 대각선(EO 클래스 = 2), 및 45° 대각선(EO 클래스 = 3)을 도시한다.

도 11은 ALF 필터 모양들(크로마: 5X5 다이아몬드, 루마: 7X7 다이아몬드)의 예들을 도시한다.

도 12A는 수직 그래디언트에 대해 서브샘플링된 위치들을 도시하고, 도 12B는 수평 그래디언트에 대해 서브샘플링된 위치들을 도시하고, 도 12C는 대각선 그래디언트에 대해 서브샘플링된 위치들을 도시하고, 도 12D는 대각선 그래디언트에 대해 서브샘플링된 위치들을 도시한다.

도 13은 루마 성분에 대한 VTM-4.0에서의 루프 필터 라인 버퍼 요건의 일 예를 도시한다.

도 14는 크로마 성분에 대한 VTM-4.0에서의 루프 필터 라인 버퍼 요건의 일 예를 도시한다.

도 15는 가상 경계들에서 수정된 블록 분류의 일 예를 도시한다.

도 16은 가상 경계들에서의 루마 성분에 대한 수정된 ALF 필터링의 일 예를 도시한다.

도 17A는 (각 변별로) 패딩되어야 할 VB 위/아래에 요구되는 하나의 라인을 도시하고, 도 17B는 (각 변별로) 패딩되어야 할 VB 위/아래에 요구되는 두 개의 라인을 도시하고, 도 17C는 (각 변별로) 패딩되어야 할 VB 위/아래에 요구되는 세 개의 라인을 도시한다.

도 18은 픽처/서브픽처/슬라이스/타일 경계에서 루마 ALF 필터링을 위한 반복 패딩의 예들을 도시한다.

도 19는 VVC에서 수평 랩 어라운드 모션 보상의 일 예를 도시한다.

도 20은 3x2 레이아웃에서 HEC의 이미지를 도시한다.

도 21A는 다른 루프 필터들에 대한 CC-ALF의 배치를 도시한다. 도 21B는 다이아몬드 모양 필터를 도시한다.

도 22는 JVET-P0080에서 CC-ALF를 위한 ALF 가상 경계에서의 반복 패딩의 일 예를 도시한다.

도 23은 8개의 고유한 계수들을 가지는 3x4 다이아몬드 모양 필터를 도시한다.

도 24는 JVET-P1008에서 CC-ALF를 위한 ALF 가상 경계에서의 반복 패딩의 일 예를 도시한다.

도 25는 JVET-P0106에서 8개의 계수들의 CC-ALF 필터 모양을 도시한다.

도 26은 JVET-P0173에서 6개의 계수들의 CC-ALF 필터 모양을 도시한다.

도 27은 JVET-P0251에서 6개의 계수들의 CC-ALF 필터 모양을 도시한다.

도 28은 JC-CCALF 워크플로우의 일 예를 도시한다.

도 29는 8 탭 4x3 필터 모양으로 CC-ALF 필터링 방법을 위해 패딩될 샘플들의 예시적 위치들을 도시한다.

도 30은 미러링된 패딩 방법 1의 예를 도시한다.

도 31은 미러링된 패딩 방법 2의 예를 도시한다.

도 32는 개시된 기술들이 실시되는 예시적인 비디오 처리 시스템의 블록도이다.

도 33은 비디오 처리를 위해 사용되는 예시적인 하드웨어 플랫폼의 블록도이다.

도 34는 비디오 처리의 예시적 방법을 위한 흐름도이다.

도 35는 본 개시의 일부 실시예들에 따른 비디오 코딩 시스템을 도시하는 블록도이다.

도 36은 본 개시의 일부 실시예들에 따른 인코더를 도시하는 블록도이다.

도 37은 본 개시의 일부 실시예들에 따른 디코더를 도시하는 블록도이다.

도 38A 및 도 38B는 개시된 기술의 일부 실시예들에 기반한 비디오 처리의 예시적 방법들을 위한 흐름도들이다.

발명을 실시하기 위한 구체적인 내용

[0012] 본 문서에서 이해의 편의를 위해 섹션 표제들이 사용되지만 이는 실시예들 및 기술들을 해당 섹션들로 한정하는 것은 아니다. 일부 설명에서 H.266 용어가 사용되지만 이는 오직 이해를 돕기 위한 것이지 개시된 기술들의 범위를 제한하기 위한 것이 아니다. 이와 같이, 여기에 설명된 기술은 다른 비디오 코덱 프로토콜 및 설계에도 적용할 수 있다.

[0013] ### 1. 간략한 요약 (Brief Summary)

[0014] 본 문서는 비디오 코딩 기술들에 관한 것이다. 구체적으로, 영상/비디오 코딩에서 특별히 크로스 컴포넌트 적응적 루프 필터(CC-ALF: cross component adaptive loop filter) 및 기타 코딩 툴들을 위한, 픽처/서브픽처/슬라이스/타일 경계, 360도 비디오 가상 경계 및 ALF 가상 경계 코딩에 관한 것이다. 본 문서는 HEVC와 같은 기존의 비디오 코딩 표준 또는 완결될 표준(Versatile Video coding)에 적용될 수 있다. 또한 미래의 비디오 코딩 표준들 또는 비디오 코덱에도 적용될 수 있다.

[0015] ### 2. 비디오 코딩 소개

[0016] 비디오 코딩 표준들은 잘 알려진 ITU-T 및 ISO/IEC 표준들의 개발을 통해 주로 발전해 왔다. ITU-T는 H.261및 H.263을 제작했으며, ISO/IEC는 MPEG-1 및 MPEG-4 Visual을 제작하였다. 두 기관은 H.262/MPEG-2 Video, H.264/MPEG-4 Advanced Video Coding (AVC) 그리고 H.265/HEVC 표준들을 공동으로 제작하였다. H.262 이후, 비디오 코딩 표준들은 시간 예측과 변환 코딩이 함께 활용되는 하이브리드 비디오 코딩 구조를 기반으로 한다. HEVC를 넘어서는 미래형 비디오 코딩 기술들을 탐구하기 위해, 2015년에 VCEG와 MPEG는 공동 비디오 탐사 팀 (Joint Video Exploration Team (JVET))를 공동 설립하였다. 그 이후로, 많은 새로운 기법들이 JVET에 의해 채택되어 공동 탐사 모델 (Joint Exploration Model (JEM))이라는 참조 소프트웨어에 포함되었다. 2018년 4월, HEVC 대비 비트 레이트 50% 감소를 목표로하는 VVC 표준 작업을 위해, VCEG(Q6/16)와 ISO/IEC JTC1 SC29/WG11(MPEG) 간의 공동 비디오 전문가 팀 (Joint Video Expert Team (JVET))이 결성되었다.

[0017] **2.1 컬러 공간 및 크로마 서브샘플링**

[0018] 컬러 모델(또는 컬러 시스템)이라고도 하는, 컬러 공간은 컬러들의 범위를 숫자들의 튜플들로서, 전형적으로 3개 또는 4 개의 값 또는 컬러 성분들(예를 들면, RGB)으로서 간단히 설명하는 추상 수학적 모델이다. 기본적으로 말하면, 컬러 공간은 좌표계와 서브 공간의 정교화이다.

[0019] 비디오 압축의 경우, 가장 자주 사용되는 컬러 공간들은 YCbCr 및 RGB이다

[0020] YCBCR 또는 Y'CBCR이라고도 기술되는, YCbCr, Y'CbCr 또는 Y Pb/Cb Pr/Cr은 비디오 및 디지털 사진 시스템들에서 컬러 이미지 파이프라인의 일부로서 사용되는 컬러 공간들의 계열이다. Y'은 루마 성분이고, CB와 CR은 청색 차이 및 적색 차이 크로마 성분들이다. Y'(프라임을 가짐)은 루미넌스인 Y와 구별되며, 감마 보정된 RGB 원색들에 기초하여 광 강도가 비선형적으로 인코딩된다는 것을 의미한다.

[0021] 크로마 서브샘플링은, 루미넌스에 대해서보다 컬러 차이들에 대한 인간 시각 시스템의 더 낮은 예민성을 이용하여, 루마 정보에 대해서보다 크로마 정보에 대해 더 낮은 분해능을 구현하는 것에 의해 이미지들을 인코딩하는 방식이다.

[0022] **2.1.1. 4:4:4**

[0023] 3개의 Y'CbCr 성분 각각은 동일한 샘플 레이트를 가지며, 따라서 크로마 서브샘플링이 없다. 이 방식은 때때로 고급 필름 스캐너들 및 영화 포스트 프로덕션에서 사용된다.

[0024] **2.1.2. 4:2:2**

[0025] 2개의 크로마 성분은 루마의 샘플 레이트의 절반으로 샘플링된다: 수평 크로마 분해능이 반으로 주는 반면에, 수직 크로마 분해능은 변하지 않는다. 이것은 시각적 차이가 거의 또는 전혀 없이 압축되지 않은 비디오 신호의 대역폭을 1/3만큼 감소시킨다. 4:2:2 컬러 포맷의 공칭 수직 및 수평 위치들의 일 예는 VVC 규격 초안에서도 1에 도시된다.

[0026] **2.1.3. 4:2:0**

[0027] 4:2:0에서는, 수평 샘플링이 4:1:1에 비해 2배이지만, Cb 및 Cr 채널들이 이 방식에서 각각의 교호 라인에서만 샘플링되므로, 수직 분해능이 반으로 준다. 따라서 데이터 레이트는 동일하다. Cb와 Cr은 각각 수평 및 수직 모두에서 2의 인자로 서브샘플링된다. 상이한 수평 및 수직 위치(siting)를 갖는, 4:2:0 방식들의 세 가지 변형이 있다.

[0028] · MPEG-2에서, Cb와 Cr은 수평으로 함께 배치된다(cosited). Cb 및 Cr은 수직 방향으로 픽셀들 사이에 배치된다(중간에 배치됨).

[0029] · JPEG/JFIF, H.261 및 MPEG-1에서, Cb와 Cr은 중간에, 즉 교호 루마 샘플들 사이의 중간에 배치된다.

[0030] · 4:2:0 DV에서, Cb와 Cr은 수평 방향으로 함께 배치된다. 수직 방향에서, 이들은 교호 라인들에 함께 배치된다.

[0031] 표 2-1. chroma_format_idc 및 separate_colour_plane_flag로부터 도출된 SubWidthC 및 SubHeightC 값들

chroma_format_idc	separate_colour_plane_flag	Chroma format	SubWidthC	SubHeightC
0	0	Monochrome	1	1
1	0	4:2:0	2	2
2	0	4:2:2	2	1
3	0	4:4:4	1	1
3	1	4:4:4	1	1

[0032]

[0033] **2.2. 전형적인 비디오 코덱의 코딩 흐름**

[0034] 도 2는 3 개의 인루프 필터링 블록, 즉 디블로킹 필터(DF: deblocking filter), 샘플 적응 오프셋(SAO: sample adaptive offset) 및 ALF를 포함하는 VVC의 인코더 블록 다이어그램의 일 예를 도시한다. 미리 정의된 필터들을 사용하는 DF와 달리, SAO 및 ALF는 오프셋을 가산하는 것 및 유한 임펄스 응답(FIR: finite impulse response) 필터를 적용하는 것에 의해 원래 샘플들과 재구성된 샘플들 간의 평균 제곱 오차들을 감소시키기 위해 현재 픽처의 원래 샘플들을 활용하며, 코딩된 부가 정보는 오프셋들 및 필터 계수들을 시그널링한다. ALF는 각 픽처의 마지막 프로세싱 단계에 위치하며, 이전 단계들에서 생성되는 아티팩트들을 잡아내어 수정하려고 하는 툴이라고 볼 수 있다.

[0035] **2.3. 비디오 유닛들의 예시적 정의들**

[0036] 픽처는 하나 또는 그 이상의 타일 행들 및 하나 또는 그 이상의 타일 열들로 분할된다. 타일은 픽처의 직사각형 영역을 덮는 CTU들의 시퀀스이다. 타일 내의 CTU들은 해당 타일 내에서 래스터 스캔 순서로 스캐닝된다.

[0037] 슬라이스는 정수개의 완전한 타일들 또는 픽처의 타일 내에서 정수개의 연속적인 완전한 CTU 행들로 구성된다.

[0038] 슬라이스들의 두 모드, 즉, 래스터 스캔 슬라이스 모드 및 직사각형 슬라이스 모드가 지원된다. 래스터 스캔 슬라이스 모드에서, 슬라이스는 픽처의 타일 래스터 스캔에서 완전한 타일들의 시퀀스를 포함한다. 직사각형 슬라이스 모드에서, 슬라이스는 상기 픽처의 직사각형 영역을 함께 형성하는 다수의 완전한 타일들을 포함하거나 또는 상기 픽처의 직사각형 영역을 함께 형성하는, 하나의 타일의 다수의 연속적인 완전한 CTU 행들을 포함한다. 직사각형 슬라이스 내의 타일들은 해당 슬라이스에 상응하는 상기 직사각형 영역 내에서 타일 래스터 스캔 순서로 스캐닝된다.

[0039] 서브픽처는 픽처의 직사각형 영역을 함께 덮는 하나 또는 그 이상의 슬라이스들을 포함한다.

[0040] 도 3은 12개의 타일과 3개의 래스터 스캔 슬라이스로 분할되는 픽처의 래스터 스캔 슬라이스 분할의 일 예를 도시한다.

[0041] VVC 사양에서 도 4는 24개의 타일 (6개의 타일 열 및 4개의 타일 행)과 9개의 직사각형 슬라이스로 분할되는 픽처의 직사각 슬라이스 분할의 일 예를 도시한다.

[0042] 도 4는 24개의 타일과 9개의 직사각형 슬라이스로 분할되는 18 x 12 CTU들을 갖는 픽처를 도시한다 (정보적).

[0043] 도 5는 타일 및 직사각형 슬라이스로 분할되는 픽처의 일 예를 도시하며, 상기 픽처는 4개의 타일(2개의 타일 열 및 2개의 타일 행)과 4개의 직사각형 슬라이스로 분할된다.

[0044] 도 6은 픽처의 서브픽처 분할의 일 예를 도시하며, 픽처는 4x4 CTU들을 덮는 15개의 타일들, 24개의 슬라이스들 및 다양한 크기를 갖는 24개의 서브픽처들로 분할된다.

[0045] **2.3.1. CTU/CTB 크기들**

[0046] VVC에서, 신택스 요소 **log2_ctu_size_minus2**에 의해 SPS에서 시그널링되는 CTU 크기는 4x4만큼 작을 수 있다.

[0047] **7.3.2.3 시퀀스 파라미터 세트 RBSP 신택스**

표 1

[0048]

seq_parameter_set_rbsp( ) {	기술자
sps_decoding_parameter_set_id	u(4)
sps_video_parameter_set_id	u(4)
sps_max_sub_layers_minus1	u(3)
sps_reserved_zero_5bits	u(5)
profile_tier_level(sps_max_sub_layers_minus1)	
gra_enabled_flag	u(1)
sps_seq_parameter_set_id	ue(v)
chroma_format_idc	ue(v)
if(chroma_format_idc = = 3)	
separate_colour_plane_flag	u(1)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
conformance_window_flag	u(1)
if(conformance_window_flag) {	
conf_win_left_offset	ue(v)
conf_win_right_offset	ue(v)
conf_win_top_offset	ue(v)
conf_win_bottom_offset	ue(v)
}	
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
log2_max_pic_order_cnt_lsb_minus4	ue(v)
sps_sub_layer_ordering_info_present_flag	u(1)
for(i = (sps_sub_layer_ordering_info_present_flag ? 0 : sps_max_sub_layers_minus1); i <= sps_max_sub_layers_minus1; i++) {	
sps_max_dec_pic_buffering_minus1[i]	ue(v)
sps_max_num_reorder_pics[i]	ue(v)
sps_max_latency_increase_plus1[i]	ue(v)
}	
long_term_ref_pics_flag	u(1)
sps_idr_rpl_present_flag	u(1)
rpl1_same_as_rpl0_flag	u(1)
for(i = 0; i < !rpl1_same_as_rpl0_flag ? 2 : 1; i++) {	
num_ref_pic_lists_in_sps[i]	ue(v)
for(j = 0; j < num_ref_pic_lists_in_sps[i]; j++)	
ref_pic_list_struct(i, j)	
}	
qtbtt_dual_tree_intra_flag	u(1)
log2_ctu_size_minus2	ue(v)
log2_min_luma_coding_block_size_minus2	ue(v)
partition_constraints_override_enabled_flag	u(1)
sps_log2_diff_min_qt_min_cb_intra_slice_luma	ue(v)
sps_log2_diff_min_qt_min_cb_inter_slice	ue(v)
sps_max_mtt_hierarchy_depth_inter_slice	ue(v)
sps_max_mtt_hierarchy_depth_intra_slice_luma	ue(v)
if(sps_max_mtt_hierarchy_depth_intra_slice_luma != 0) {	
sps_log2_diff_max_bt_min_qt_intra_slice_luma	ue(v)
sps_log2_diff_max_tt_min_qt_intra_slice_luma	ue(v)
}	
if(sps_max_mtt_hierarchy_depth_inter_slices != 0) {	

sps_log2_diff_max_bt_min_qt_inter_slice	ue(v)
sps_log2_diff_max_tt_min_qt_inter_slice	ue(v)
}	
if(qtbtt_dual_tree_intra_flag) {	
sps_log2_diff_min_qt_min_cb_intra_slice_chroma	ue(v)
sps_max_mtt_hierarchy_depth_intra_slice_chroma	ue(v)
if (sps_max_mtt_hierarchy_depth_intra_slice_chroma != 0) {	
sps_log2_diff_max_bt_min_qt_intra_slice_chroma	ue(v)
sps_log2_diff_max_tt_min_qt_intra_slice_chroma	ue(v)
}	
}	
쪈	
rbsp_trailing_bits( )	
}	

[0049] log2_ctu_size_minus2 + 2는 각각의 CTU의 루마 코딩 트리 블록 크기를 명시한다.

[0050] log2_min_luma_coding_block_size_minus2 + 2는 최소 루마 코딩 블록 크기를 명시한다.

[0051] 변수들 CtbLog2SizeY, CtbSizeY, MinCbLog2SizeY, MinCbSizeY, MinTbLog2SizeY, MaxTbLog2SizeY, MinTbSizeY, MaxTbSizeY, PicWidthInCtbsY, PicHeightInCtbsY, PicSizeInCtbsY, PicWidthInMinCbsY, PicHeightInMinCbsY, PicSizeInMinCbsY, PicSizeInSamplesY, PicWidthInSamplesC and PicHeightInSamplesC은 다음과 같이 도출된다.

$$CtbLog2SizeY = log2_ctu_size_minus2 + 2 \quad (7\text{-}9)$$

$$CtbSizeY = 1 \ll CtbLog2SizeY \quad (7\text{-}10)$$

$$MinCbLog2SizeY = log2_min_luma_coding_block_size_minus2 + 2 \quad (7\text{-}11)$$

$$MinCbSizeY = 1 \ll MinCbLog2SizeY \quad (7\text{-}12)$$

$$MinTbLog2SizeY = 2 \quad (7\text{-}13)$$

$$MaxTbLog2SizeY = 6 \quad (7\text{-}14)$$

$$MinTbSizeY = 1 \ll MinTbLog2SizeY \quad (7\text{-}15)$$

$$MaxTbSizeY = 1 \ll MaxTbLog2SizeY \quad (7\text{-}16)$$

$$PicWidthInCtbsY = Ceil(pic_width_in_luma_samples \div CtbSizeY) \quad (7\text{-}17)$$

$$PicHeightInCtbsY = Ceil(pic_height_in_luma_samples \div CtbSizeY) \quad (7\text{-}18)$$

$$PicSizeInCtbsY = PicWidthInCtbsY * PicHeightInCtbsY \quad (7\text{-}19)$$

$$PicWidthInMinCbsY = pic_width_in_luma_samples / MinCbSizeY \quad (7\text{-}20)$$

$$PicHeightInMinCbsY = pic_height_in_luma_samples / MinCbSizeY \quad (7\text{-}21)$$

$$PicSizeInMinCbsY = PicWidthInMinCbsY * PicHeightInMinCbsY \quad (7\text{-}22)$$

$$PicSizeInSamplesY = pic_width_in_luma_samples * pic_height_in_luma_samples \quad (7\text{-}23)$$

$$PicWidthInSamplesC = pic_width_in_luma_samples / SubWidthC \quad (7\text{-}24)$$

$$PicHeightInSamplesC = pic_height_in_luma_samples / SubHeightC \quad (7\text{-}25)$$

[0052]

[0053] 2.3.2. 픽처에서의 CTU들

[0054] M x N으로 표시되는 CTB/LCU 크기(HEVC/VVC에 정의된 바와 같이, 전형적으로 M은 N과 동일함)를 가정하고, 픽처(또는 타일 또는 슬라이스 또는 다른 종류들의 유형들, 픽처 경계는 예로서 취해짐) 경계에 위치하는 CTB의 경우, K x L 개의 샘플들이 픽처 경계 내에 있으며 여기서 K < M이거나 L < N이다. 도 7A 내지 도 7C에 도시된 바와 같은 해당 CTB들에 대해, **CTB 크기는 여전히 MxN과 동일하지만, CTB의 하단 경계/우측 경계는 픽처 외부에 있다.**

[0055] 도 7은 픽처 경계들에 걸쳐 있는 CTB들의 예들을 도시하며, (a) K=M, L<N, (b) K<M, L=N 그리고 (c) K<M, L<N 이다. 또한, (a) 하단 픽처 경계에 걸쳐 있는 CTB들, (b) 우측 픽처 경계에 걸쳐 있는 CTB들, (c) 우측 하단 픽처 경계에 걸쳐 있는 CTB들.

[0056] **2.4. 디블로킹 필터 (DB: deblocking filter)**

[0057] **DB의 입력은 인루프 필터들 이전의 재구성된 샘플들이다.**

[0058] 픽처에서의 수직 에지들이 먼저 필터링된다. 이어서 픽처에서의 수평 에지들이 수직 에지 필터링 프로세스에 의해 수정된 샘플들을 입력으로 하여 필터링된다. 각 CTU의 CTB들에서의 수직 및 수평 에지들은 코딩 유닛 기반으로 별도로 프로세싱된다. 코딩 유닛에서의 코딩 블록들의 수직 에지들은 그것들의 기하학적 순서에서 코딩 블록들의 좌측에 있는 에지에서 시작하여 에지들을 통해 코딩 블록들의 우측을 향해 진행하면서 필터링된다. 코딩 유닛에서의 코딩 블록들의 수평 에지들은 그들의 기하학적 순서에서 코딩 블록들의 상단에 있는 에지에서 시작하여 에지들을 통해 코딩 블록들의 하단을 향해 진행하면서 필터링된다.

[0059] 도 8은 병렬로 디블로킹될 수 있는, 8X8 그리드 상의 픽처 샘플들과 수평 및 수직 블록 경계들 그리고 8X8 샘플들의 비중첩 블록들의 예시이다.

[0060] **2.4.1. 경계 결정**

[0061] 필터링이 8x8 블록 경계들에 적용된다. 추가적으로, 이는 (예를 들면, 아핀 모션 예측, ATMVP의 사용으로 인해) 변환 블록 경계 또는 코딩 서브블록 경계여야 한다. 그러한 경계들이 아닌 것들에 대해, 필터가 비활성화된다.

[0062] **2.4.2. 경계 강도 계산**

[0063] 변환 블록 경계/코딩 서브블록 경계에 대해, 8x8 그리드 내에 위치하는 경우, 그것은 필터링될 수 있으며, 이 에지에 대한 bS[xDi][yDj](여기서 [xDi][yDj]는 좌표를 나타냄)의 설정은, 표 2-2 및 표 2-3에 각각 정의되어 있다.

[0064] 표 2-2. 경계 강도 (SPS IBC가 비활성화되는 경우)

우선순위	조건들	Y	U	V
5	인접 블록들 중 적어도 하나는 인트라이다.	2	2	2
4	인접 블록들 중 적어도 하나 및 TU 경계는 0이 아닌 변환 계수들을 가진다.	1	1	1
3	인접 블록들의 MV 들의 수(단방향의 경우 1, 양방향의 경우 2) 또는 참조 픽처들이 상이하다.	1	N/A	N/A
2	인접 블록들에 속하는 동일 참조 픽처의 모션 벡터들 사이의 절대 차이가 하나의 정수 루마 샘플보다 크거나 같다.	1	N/A	N/A
1	그 외의 경우	0	0	0

[0065]

[0066] 표 2-3. 경계 강도 (SPS IBC가 활성화되는 경우)

우선순위	조건들	Y	U	V
8	인접 블록들 중 적어도 하나는 인트라이다.	2	2	2
7	인접 블록들 중 적어도 하나 및 TU 경계는 0이 아닌 변환 계수들을 가진다.	1	1	1
6	인접 블록들의 예측 모드는 상이하다(예를 들어, 하나는 IBC, 하나는 인터).	1		
5	인접 블록들에 속하는 모션 벡터들 간의 절대차 및 IBC 모두 하나의 정수 루마 샘플 보다 크거나 같다.	1	N/A	N/A
4	인접 블록들의 MV 들의 수(단방향의 경우 1, 양방향의 경우 2) 또는 참조 픽처들이 상이하다.	1	N/A	N/A
3	인접 블록들에 속하는 동일 참조 픽처의 모션 벡터들 사이의 절대 차이가 하나의 정수 루마 샘플 보다 크거나 같다.	1	N/A	N/A
1	그 외의 경우	0	0	0

[0067]

[0068] **2.4.3. 루마 성분에 대한 디블로킹 결정**

[0069] 디블로킹 결정 프로세스는 이 하위 섹션에 설명되어 있다.

[0070] Condition1, Condition2 및 Condition3 모두가 TRUE인 경우에만, 보다 넓고 보다 강한 루마 필터들이 사용되는 필터들이다.

[0071] 조건 1은 "큰 블록 조건"이다. 이 조건은 P 측과 Q 측에 있는 샘플들이, 제각기, 변수 bSidePisLargeBlk 및 bSideQisLargeBlk에 의해 표현되는, 큰 블록들에 속하는지 여부를 검출한다. bSidePisLargeBlk 및 bSideQisLargeBlk는 다음과 같이 정의된다.

bSidePisLargeBlk = ((edge type is vertical and p_0 belongs to CU with width >= 32) | | (edge type is horizontal and p_0 belongs to CU with height >= 32))? TRUE: FALSE

bSideQisLargeBlk = ((edge type is vertical and q_0 belongs to CU with width >= 32) | | (edge type is horizontal and q_0 belongs to CU with height >= 32))? TRUE: FALSE

[0072]

[0073] bSidePisLargeBlk 및 bSideQisLargeBlk에 기초하여, 조건 1은 아래와 같이 정의된다.

[0074] **Condition1** = (bSidePisLargeBlk || bSidePisLargeBlk) ? TRUE: FALSE

[0075] 다음으로, **Condition 1**이 참인 경우, 조건 2가 더 검사될 것이다. 우선, 아래 변수들이 도출된다.

[0076] - dp0, dp3, dq0, dq3가 HEVC에서와 같이 우선 도출된다.

- if (p side is greater than or equal to 32)

  $dp0 = (dp0 + Abs(p5_0 - 2 * p4_0 + p3_0) + 1) >> 1$

  $dp3 = (dp3 + Abs(p5_3 - 2 * p4_3 + p3_3) + 1) >> 1$

- if (q side is greater than or equal to 32)

  $dq0 = (dq0 + Abs(q5_0 - 2 * q4_0 + q3_0) + 1) >> 1$

  $dq3 = (dq3 + Abs(q5_3 - 2 * q4_3 + q3_3) + 1) >> 1$

[0077]

[0078] **Condition2** = (d < β) ? TRUE: FALSE

[0079] 여기서 d= dp0 + dq0 + dp3 + dq3이다.

[0080] Condition1 및 Condition2가 유효한 경우, 블록들 중 임의의 것이 서브블록들을 사용하는지 여부가 더 검사된다.

```
If (bSidePisLargeBlk)
{
   If (mode block P == SUBBLOCKMODE)
     Sp =5
    else
     Sp =7
}
else
 Sp = 3

If (bSideQisLargeBlk)
 {
   If (mode block Q == SUBBLOCKMODE)
      Sq =5
    else
     Sq =7
  }
else
  Sq = 3
```

[0081]

[0082] 마지막으로, **Condition 1** 및 **Condition 2** 둘 모두가 유효한 경우, 제안된 디블로킹 방법은 다음과 같이 정의되는 조건 3 (큰 블록 강한 필터 조건)을 검사할 것이다.

[0083] **Condition3** StrongFilterCondition에서, 아래 변수들이 도출된다.

[0084] dpq는 HEVC에서와 같이 도출된다.

sp_3 = Abs(p_3 − p_0), derived as in HEVC
if (p side is greater than or equal to 32)
  if(Sp==5)
    sp_3 = (sp_3 + Abs(p_5 − p_3) + 1) >> 1
  else
    sp_3 = (sp_3 + Abs(p_7 − p_3) + 1) >> 1
sq_3 = Abs(q_0 − q_3), derived as in HEVC
if (q side is greater than or equal to 32)
  If(Sq==5)
    sq_3 = (sq_3 + Abs(q_5 − q_3) + 1) >> 1
  else
    sq_3 = (sq_3 + Abs(q_7 − q_3) + 1) >> 1

[0085]

[0086] HEVC에서와 같이, StrongFilterCondition = (dpq가 (β >> 2)보다 작고, sp_3 + sq_3가 (3* β >> 5)보다 작고, Abs(p_0 - q_0)가 (5 * t_C + 1) >> 1 보다 작음) ? TRUE : FALSE.

[0087] **2.4.4. 루마에 대한 보다 강한 디블로킹 필터(보다 큰 블록들을 위해 설계됨)**

[0088] 경계의 어느 한쪽 측면에 있는 샘플들이 큰 블록에 속할 때 이중선형 필터가 사용된다. 큰 블록에 속하는 샘플은 수직 에지의 경우 폭 >= 32일 때와 같이 정의되고, 수평 에지의 경우 높이 >= 32일 때와 같이 정의된다.

[0089] 이중선형 필터는 아래에서 열거된다.

[0090] 블록 경계 샘플들 p_i (i는 0 내지 Sp-1) 및 q_i (j는 0 내지 Sq-1)(pi 및 qi는 위에서 설명된 HEVC 디블로킹에서 수직 에지를 필터링하기 위한 한 행 내에서의 i 번째 샘플 또는 수평 에지를 필터링하기 위한 한 열 내에서의 i 번째 샘플)는 이어서 다음과 같이 선형 보간으로 대체된다.

[0091]
- $p_i' = (f_i * Middle_{s,t} + (64 - f_i) * P_s + 32) \gg 6), clipped\ to\ p_i \pm tcPD_i$
- $q_j' = (g_j * Middle_{s,t} + (64 - g_j) * Q_s + 32) \gg 6), clipped\ to\ q_j \pm tcPD_j$

[0092] 여기서, $tcPD_i$ 및 $tcPD_j$ 항은 섹션 2.4.7에서 설명된 위치 의존적 클리핑이고, g_j, f_i, $Middle_{s,t}$, P_s 및 Q_s는 아래에 주어진다

[0093] **2.4.5. 크로마에 대한 디블로킹 제어**

[0094] 크로마 강한 필터들은 블록 경계의 양측에서 사용된다. 여기서, 크로마 필터는 크로마 에지의 양측이 8(크로마 위치) 이상이고, 세 가지 조건에 대한 다음 결정이 충족될 때 선택된다. 첫 번째 조건은 큰 블록 뿐 아니라 경계 강도의 결정을 위한 것이다. 크로마 샘플 도메인에서 블록 에지와 직교하는 블록 폭 또는 높이가 8 이상일 때 제안된 필터가 적용될 수 있다. 두 번째 조건 및 세 번째 조건은 HEVC 루마 디블로킹 결정에 대한 것과 기본적으로 동일하며, 각기 온/오프 결정 및 강한 필터 결정이다.

[0095] 첫 번째 결정에서, 경계 강도(bS)는 크로마 필터링을 위해 수정되고 조건들이 순차적으로 검사된다. 조건이 충족되는 경우, 보다 낮은 우선순위들을 갖는 나머지 조건들은 스킵된다.

[0096] bS가 2와 동일할 때, 또는 큰 블록 경계가 검출되는 경우 bS가 1과 동일할 때, 크로마 디블로킹이 수행된다.

[0097] 두 번째 조건 및 세 번째 조건은 다음과 같이 HEVC 루마 강한 필터 결정과 기본적으로 동일하다

[0098] 두 번째 조건에서,

[0099] d는 이어서 HEVC 루마 디블로킹에서와 같이 도출된다.

[0100] 두 번째 조건은 d가 β 보다 작을 때 TRUE일 것이다.

[0101] 세 번째 조건에서, StrongFilterCondition은 다음과 같이 도출된다.

[0102] dpq는 HEVC에서와 같이 도출된다.

[0103] $sp_3 = Abs(p_3 - p_0)$, HEVC에서와 같이 도출됨.

[0104] $sq_3 = Abs(q_0 - q_3)$, HEVC에서와 같이 도출됨.

[0105] HEVC 설계에서와 같이, StrongFilterCondition = (dpq가 (β >> 2) 보다 작고, $sp_3 + sq_3$가 (β >> 3) 보다 작고, 그리고 $Abs(p_0 - q_0)$가 ($5 * t_C + 1$) >> 1) 보다 작다)

[0106] **2.4.6. 크로마를 위한 강한 디블로킹 필터**

[0107] 크로마에 대한 아래의 강한 디블로킹 필터가 정의된다.

[0108]
$$p_2' = (3*p_3+2*p_2+p_1+p_0+q_0+4) >> 3$$
$$p_1' = (2*p_3+p_2+2*p_1+p_0+q_0+q_1+4) >> 3$$
$$p_0' = (p_3+p_2+p_1+2*p_0+q_0+q_1+q_2+4) >> 3$$

[0109] 제안된 크로마 필터는 4x4 크로마 샘플 그리드에 대해 디블로킹을 수행한다.

[0110] **2.4.7. 위치 의존적 클리핑**

[0111] 위치 의존적 클리핑 tcPD는 경계에서 7개, 5개 및 3개의 샘플을 수정하는 강하고 긴 필터들을 수반하는 루마 필터링 프로세스의 출력 샘플들에 적용된다. 양자화 오차 분포를 가정할 때, 보다 높은 양자화 잡음을 가질 것으로 예상되고, 따라서 재구성된 샘플 값과 실제 샘플 값 간의 보다 높은 편차를 가질 것으로 예상되는 샘플들에 대해 클리핑 값을 증가시키는 것이 제안된다.

[0112] 섹션 2.4.2에서의 의사결정 프로세스의 결과에 따라, 비대칭 필터로 필터링되는 각각의 P 또는 Q 경계에 대해, 위치 의존적 임계치 테이블이 디코더에 부가 정보로서 제공되는 2 개의 테이블(예를 들면, 아래에 표로 작성된 Tc7 및 Tc3)로부터 선택된다.

[0113]
$$Tc7 = \{ 6, 5, 4, 3, 2, 1, 1 \};\ Tc3 = \{ 6, 4, 2 \};$$
$$tcPD = (Sp == 3)\ ?\ Tc3 : Tc7;$$
$$tcQD = (Sq == 3)\ ?\ Tc3 : Tc7;$$

[0114] 짧은 대칭 필터로 필터링되는 P 또는 Q 경계들에 대해, 보다 낮은 크기의 위치 의존적 임계치가 적용된다.

[0115] $Tc3 = \{ 3, 2, 1 \};$

[0116] 임계치를 정의한 후에, 필터링된 p'_i 및 q'_i 샘플 값들이 tcP 및 tcQ 클리핑 값들에 따라 클리핑된다.

[0117]
$$p''_i = Clip3(p'_i + tcP_i, p'_i - tcP_i, p'_i);$$
$$q''_j = Clip3(q'_j + tcQ_j, q'_j - tcQ_j, q'_j);$$

[0118] 여기서 p'_i 및 q'_i 는 필터링된 샘플 값들이고, p''_i 및 q''_j 는 상기 클리핑 이후의 출력 샘플 값이며, tcP_i 및 tcP_i는 VVC tc 파라미터 및 $tcPD$ 및 $tcQD$로부터 도출되는 클리핑 임계치들이다. 함수 Clip3은 VVC에

명시된 바와 같이 클리핑 함수이다

[0119] **2.4.8. 서브블록 디블로킹 조정**

[0120] 긴 필터들 및 서브블록 디블로킹 둘 모두를 사용하여 병렬 친화적인 디블로킹을 가능하게 하기 위해, 긴 필터들에 대한 루마 제어에 나와 있는 바와 같이 서브블록 디블로킹(AFFINE 또는 ATMVP 또는 DMVR)을 사용하는 측면에서 최대 5 개의 샘플을 수정하도록 긴 필터들이 제한된다. 추가적으로, CU에 가까운 8x8 그리드 상의 서브블록 경계들 또는 암시적 TU 경계가 각각의 측면에서 최대 2 개의 샘플을 수정하게 제한되도록 서브블록 디블로킹이 조정된다.

[0121] 이하는 CU 경계와 정렬되지 않은 서브블록 경계들에 적용된다.

```
If (mode block Q == SUBBLOCKMODE && edge !=0) {
  if (!(implicitTU && (edge == (64 / 4))))
    if (edge == 2 || edge == (orthogonalLength - 2) || edge == (56 / 4) || edge == (72 / 4))
      Sp = Sq = 2;
    else
      Sp = Sq = 3;
  else
    Sp = Sq = bSideQisLargeBlk ? 5:3
}
```

[0122]

[0123] 0과 동일한 에지가 CU 경계에 대응하는 경우, 2와 동일하거나 orthogonalLength-2와 동일한 에지는 CU 경계 등으로부터 8개 샘플 떨어져 있는 서브블록 경계에 대응한다. 여기서 TU의 암시적 분할이 사용되는 경우 암시적 TU는 참이다.

[0124] **2.5. 샘플 적응적 오프셋 (SAO)**

[0125] **SAO의 입력은 DB 이후의 재구성된 샘플들이다.** SAO의 개념은 먼저 선택된 분류기로 영역 샘플들을 다수의 카테고리들로 분류하는 것, 각각의 카테고리에 대한 오프셋을 획득하는 것 및 그 다음으로 카테고리의 각각의 샘플에 오프셋을 가산하는 것에 의해 영역의 평균 샘플 왜곡을 감소시키는 것이며, 여기서 분류기 인덱스 및 영역의 오프셋들은 비트스트림에 코딩된다. HEVC 및 VVC에서, 영역(SAO 파라미터 시그널링을 위한 유닛)은 CTU이도록 정의된다.

[0126] 낮은 복잡도의 요구사항들을 충족시킬 수 있는 두 가지 SAO 유형이 HEVC에서 채택된다. 이러한 두 가지 유형은 에지 오프셋(EO) 및 밴드 오프셋(BO)이며, 이에 대해서는 아래에서 더 상세히 논의된다. SAO 유형의 인덱스가 코딩된다(이는 [0, 2]의 범위에 있음). EO의 경우, 샘플 분류는 1-D 방향 패턴들, 즉 수평, 수직, 135° 대각선 및 45° 대각선에 따라 현재 샘플들과 이웃 샘플들 간의 비교에 기초한다.

[0127] 도 10A 내지 도 10D는 EO 샘플 분류를 위한 4 개의 1-D 방향성 패턴, 즉 수평(EO 클래스 = 0), 수직(EO 클래스 = 1), 135° 대각선(EO 클래스 = 2), 및 45° 대각선(EO 클래스 = 3)을 도시한다.

[0128] 주어진 EO 클래스에 대해, CTB 내부의 각각의 샘플은 5가지 카테고리 중 하나로 분류된다. "c"로 라벨링된 현재 샘플 값은 선택된 1-D 패턴을 따라 ("a" 및 "b"로 라벨링된) 그것의 2개의 이웃들과 비교된다. 각각의 샘플에 대한 분류 규칙들은 표 2-3에 요약되어 있다. 카테고리 1 및 카테고리 4는 선택된 1-D 패턴을 따라 로컬 밸리 및 로컬 피크와 각각 연관된다. 카테고리 2와 카테고리 3은 선택된 1-D 패턴을 따라 오목 코너 및 볼록 코너와 각각 연관된다. 현재 샘플이 EO 카테고리 1 내지 EO 카테고리 4에 속하지 않는 경우, 이는 카테고리 0이며 SAO가 적용되지 않는다.

[0129] 표 2-4. 엣지 오프셋에 대한 샘플 분류 규칙들

Category	Condition
1	c<a and c<b
2	(c < a && c==b) ||(c == a && c < b)
3	(c > a && c==b) ||(c == a && c > b)
4	c > a && c > b
5	None of above

[0130]

[0131] **2.6. 적응적 루프 필터(ALF)**

[0132] VVC에서는 블록 기반 필터 적응을 가진 적응적 루프 필터(ALF: Adaptive Loop Filter)가 적용된다. 루마 성분에 대해, 로컬 그래디언트들의 방향 및 활성도에 기반하여, 25개 필터들 중 하나가 각 4x4 블록을 위해 선택된다.

[0133] 2.6.1. 필터 모양

[0134] (도 11에 도시되는 바와 같은) 2개의 다이아몬드 필터 모양들이 사용된다. 7Y7 다이아몬드 형상이 루마 성분에 대해 적용되고, 5Y5 다이아몬드 형상이 크로마 성분에 대해 적용된다.

[0135] 도 11은 ALF 필터 모양들을(크로마: 5X5 다이아몬드, 루마: 7X7 다이아몬드) 도시한다.

[0136] **2.6.2. 블록 분류**

[0137] 루마 성분에 대해, 각 4x4 블록은 25개의 클래스들 중 하나로 분류된다. 분류 인덱스 C는 그것의 방향성 D 및 활성도 $\hat{A}$의 양자화된 값에 기반하여 아래와 같이 도출된다.

[0138] $$C = 5D + \hat{A} \qquad (2\text{-}1)$$

[0139] D 및 $\hat{A}$를 계산하기 위해, 수평, 수직 그리고 두 대각선 방향의 그래디언트들이 1-D 라플라스(Laplacian)을 이용하여 먼저 계산된다.

$$g_v = \sum_{k=i-2}^{i+3}\sum_{l=j-2}^{j+3} V_{k,l},\ V_{k,l} = |2R(k,l) - R(k,l-1) - R(k,l+1)| \qquad (2\text{-}2)$$

$$g_h = \sum_{k=i-2}^{i+3}\sum_{l=j-2}^{j+3} H_{k,l},\ H_{k,l} = |2R(k,l) - R(k-1,l) - R(k+1,l)| \qquad (2\text{-}3)$$

$$g_{d1} = \sum_{k=i-2}^{i+3}\sum_{l=j-3}^{j+3} D1_{k,l},\ D1_{k,l} = |2R(k,l) - R(k-1,l-1) - R(k+1,l+1)| \qquad (2\text{-}4)$$

[0140] $$g_{d2} = \sum_{k=i-2}^{i+3}\sum_{j=j-2}^{j+3} D2_{k,l},\ D2_{k,l} = |2R(k,l) - R(k-1,l+1) - R(k+1,l-1)| \qquad (2\text{-}5)$$

[0141] 여기서 인덱스들 i 및 j는 4x4 블록 내 좌 상단 샘플의 좌표들을 가리키고, $R(i,j)$는 좌표 (i,j)에서의 재구성된 샘플을 지시한다.

[0142] 블록 분류의 복잡성을 줄이기 위해, 서브샘플링된 1-D 라플라스 계산이 적용된다. 도 12는 서브샘플링된 라플라스 계산을 도시한다. 도 12A는 수직 그래디언트에 대해 서브샘플링된 위치들을 도시하며, 도 12B는 수평 그래디언트에 대해 서브샘플링된 위치들을 도시하고, 도 12C는 대각선 그래디언트에 대해 서브샘플링된 위치들을 도시하고, 도 12D는 대각선 그래디언트에 대해 서브샘플링된 위치들을 도시한다.

[0143] 그 다음으로 수평 및 수직 방향들의 그래디언트들의 D 최대 및 최소 값들은 다음과 같이 설정된다.

[0144] $$g_{h,v}^{max} = max(g_h, g_v), \quad g_{h,v}^{min} = min(g_h, g_v) \tag{2-6}$$

[0145] 두 대각선 방향의 그래디언트의 최대 및 최소 값들은 아래와 같이 설정된다.

[0146] $$g_{d0,d1}^{max} = max(g_{d0}, g_{d1}), \quad g_{d0,d1}^{min} = min(g_{d0}, g_{d1}) \tag{2-7}$$

[0147] 방향성 D의 값들 도출하기 위해, 이 값들은 서로서로 그리고 2개의 임계치들 t_1 및 t_2와 비교된다.

[0148] **단계 1.** $g_{h,v}^{max} \le t_1 \cdot g_{h,v}^{min}$ 및 $g_{d0,d1}^{max} \le t_1 \cdot g_{d0,d1}^{min}$가 모두 참인 경우, D는 0으로 설정된다.

[0149] **단계 2.** $g_{h,v}^{max}/g_{h,v}^{min} > g_{d0,d1}^{max}/g_{d0,d1}^{min}$ 이면, 단계 3으로 진행하고, 그렇지 않은 경우 단계 4로 진행한다.

[0150] **단계 3.** $g_{h,v}^{max} > t_2 \cdot g_{h,v}^{min}$ 이면, D는 2로 설정되고, 그렇지 않은 경우 D는 1로 설정된다.

[0151] **단계 4.** $g_{d0,d1}^{max} > t_2 \cdot g_{d0,d1}^{min}$ 이면, D는 4로 설정되고, 그렇지 않은 경우 D는 3으로 설정된다

[0152] 활성도 값 A는 아래와 같이 계산된다.

[0153] $$A = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} (V_{k,l} + H_{k,l}) \tag{2-8}$$

[0154] A는 0부터 4까지를 포함하는 범위로 더 양자화되며, 양자화된 값은 $\hat{A}$으로 표시된다.

[0155] 픽처 내의 크로마 성분들에 대해서는 분류 방법이 적용되지 않는다. 즉, ALF 계수들의 단일한 세트가 각 크로마 성분에 대해 적용된다.

[0156] **2.6.3. Geometric transformations of filter coefficients and clipping values 필터 계수들 및 클리핑 값들의 기하학적 변환들**

[0157] 각 4Y4 루마 블록을 필터링하기 전에, 회전 또는 대각선 및 수직 플리핑(flipping) 등의 기하학적 변환들이 상기 블록에 대하여 산출된 그래디언트 값들에 따라 필터 계수들 $f(k,l)$ 및 해당 필터 클리핑 값들 $c(k,l)$에 적용된다. 이것은 필터 지원 영역에서 이들 변환들을 샘플들에 적용하는 것과 동등하다. 아이디어는 그것들의 방향성을 정렬함으로써 더 유사하게 ALF가 적용되는 상이한 블록들을 만드는 것이다.

[0158] 대각선(Diagonal), 수직 플립(Vertical flip) 및 회전(Rotation)을 포함하는 3개의 기하학적 변환들이 도입된다.

[0159]
$$\text{Diagonal: } f_D(k,l) = f(l,k), c_D(k,l) = c(l,k), \tag{2-9}$$
$$\text{Vertical flip: } f_V(k,l) = f(k, K-l-1), c_V(k,l) = c(k, K-l-1) \tag{2-10}$$
$$\text{Rotation: } f_R(k,l) = f(K-l-1, k), c_R(k,l) = c(K-l-1, k) \tag{2-11}$$

[0160] 여기서, K는 필터의 크기이고 $0 \le k,l \le K-1$는 계수 좌표들이어서, 위치(0,0)는 좌측 상단 코너에 있고, 위치$(K-1, K-1)$는 우측 하단 코너에 있다. 상기 변환들은 상기 블록에 대하여 산출된 그래디언트 값들에 따라 필터 계수들 f(k, l) 및 클리핑 값들 $c(k,l)$에 적용된다. 4개의 방향의 4개의 그래디언트 및 변환 간의 관계가 아래 표에 요약된다.

[0161] 표 2-5. 하나의 블록에 대해 산출된 그래디언트와 변환의 매핑

Gradient values	Transformation
$g_{d2} < g_{d1}$ and $g_h < g_v$	No transformation
$g_{d2} < g_{d1}$ and $g_v < g_h$	Diagonal
$g_{d1} < g_{d2}$ and $g_h < g_v$	Vertical flip
$g_{d1} < g_{d2}$ and $g_v < g_h$	Rotation

[0162]

[0163] **2.6.4. 필터 파라미터 시그널링**

[0164] ALF 필터 파라미터들은 적응 파라미터 세트(APS: Adaptation Parameter Set)에서 시그널링된다. 하나의 APS에서, 최대 25 세트의 루마 필터 계수들 및 클리핑 값 인덱스들 및 최대 8 세트의 크로마 필터 계수들 및 클리핑 값 인덱스들이 시그널링될 수 있다. 비트 오버헤드를 줄이기 위해, 루마 성분에 대한 상이한 분류의 필터 계수들이 병합될 수 있다. 슬라이스 헤더에서, 현재 슬라이스에 사용되는 APS들의 인덱스들이 시그널링된다.

[0165] APS로부터 디코딩된 클리핑 값 인덱스들은 루마 및 크로마 성분 모두에 대해 클리핑 값들의 표를 사용하여 클리핑 값을 결정하도록 허용한다. 이들 클리핑 값들은 내부 비트뎁스(bitdepth)에 의존한다. 더 상세히 말하면, 상기 클리핑 값들은 아래 식에 의해 획득된다.

[0166] $$\text{AlfClip}= \{\text{round}(2^{B-\alpha*n}) \text{ for } n \in [0..N-1]\} \quad (2\text{-}12)$$

[0167] B는 내부 비트뎁스와 동일하고, a는 미리 정의된 상수로 2.35와 같으며, N은 VVC에서 허용된 클리핑 값들의 수인 4와 동일하다.

[0168] 슬라이스 헤더에서, 최대 7개의 APS 인덱스들이 현재 슬라이스를 위해 사용되는 루마 필터 세트들을 명시하기 위해 시그널링될 수 있다. 필터링 프로세스는 CTB 레벨에서 더 제어될 수 있다. ALF가 루마 CTB에 적용되는지 여부를 지시하기 위해 플래그가 항상 시그널링된다. 루마 CTB는 16개의 고정된 필터 세트들 및 APS들로부터의 필터 세트들 중 하나의 필터 세트를 선택할 수 있다. 필터 세트 인덱스는 어느 필터 세트가 적용되는지 지시하도록 루마 CTB를 위해 시그널링된다. 16개의 고정된 필터 세트들은 미리 정의되고 인코더와 디코더 양쪽에 모두 하드코딩된다(hard-coded).

[0169] 크로마 성분을 위해, APS 인덱스는 현재 슬라이스를 위해 사용되는 크로마 필터 세트들을 지시하기위해 슬라이스 헤더에서 시그널링된다. CTB 레벨에서, APS에 두개 이상의 크로마 필터 세트가 있는 경우에 필터 인덱스는 각 크로마 CTB를 위해 시그널링된다.

[0170] 필터 계수들은 128을 기저(norm)로 양자화된다. 곱셈 복잡도를 제한하기 위해, 비트스트림 적합도(bitstream conformance)가 적용될 수 있고 따라서 중심 위치에 있지 않은(non-central position) 계수 값은 -2^7부터 2^7-1까지의 범위 내에 있어야 한다. 중심 위치 계수는 비트스트림에서 시그널링되지 않으며 128과 같은 것으로 간주된다.

[0171] **2.6.5. 필터링 프로세스**

[0172] 디코더 측에서, ALF가 CTB에 대해 활성화되는 경우, CU 내의 각 샘플 (i,j) 가 필터링되어, 아래와 같은 샘플 값 $R'(i,j)$이 산출된다.

[0173] $$R'(i,j) = R(i,j) + \left(\left(\sum_{k\neq 0}\sum_{l\neq 0} f(k,l) \times K\left(R(i+k,j+l) - R(i,j), c(k,l)\right) + 64\right) \gg 7\right) \quad (2\text{-}13)$$

[0174] 여기서, $f(k,l)$는 디코딩된 필터 계수들을 나타내고, $K(x,y)$ 는 클리핑 함수이며, $c(k,l)$는 디코딩된 파라미터들을 나타낸다. 변수 k 및 l는 $-\frac{L}{2}$ 및 $\frac{L}{2}$ 사이에서 변화하며, 이때 L은 필터 길이를 나타낸다. 클리핑 함수 $K(x,y) = \min(y, \max(-y, x))$는 함수 $Clip3(-y, y, x)$에 대응한다.

[0175] **2.6.6. 라인 버퍼 감소를 위한 가상 경계 필터링 프로세스**

[0176] 하드웨어 및 임베디드 소프트웨어에서, 픽처 기반 프로세싱은 그것의 높은 픽처 버퍼 요구사항으로 인해 실제로 용인 가능하지 않다. 온칩 픽처 버퍼들을 사용하는 것은 매우 비싸고 오프칩 픽처 버퍼들을 사용하는 것은 외부 메모리 액세스, 전력 소비 및 데이터 액세스 지연시간을 상당히 증가시킨다. 따라서, 실제 제품들에서는 DF, SAO 및 ALF가 픽처 기반 디코딩으로부터 LCU 기반 디코딩으로 변경될 것이다. LCU 기반 프로세싱이 DF, SAO 및 ALF에 사용될 때, 전체 디코딩 프로세스는 다수의 LCU들의 병렬 프로세싱을 위한 LCU 파이프라이닝 방식으로 래스터 스캔에서 LCU별로 행해질 수 있다. 이 경우에, 하나의 LCU 행을 프로세싱하는 것이 상단 LCU 행으로부터의 픽셀들을 필요로 하기 때문에 DF, SAO 및 ALF에 대해 라인 버퍼들이 필요하다. 오프칩 라인 버퍼들(예를 들면, DRAM)이 사용되는 경우, 외부 메모리 대역폭과 전력 소비가 증가될 것이며, 온칩 라인 버퍼들(예를 들면, SRAM)이 사용되는 경우, 칩 면적이 증가될 것이다. 따라서, 라인 버퍼들이 이미 픽처 버퍼들보다 훨씬 더 작더라도, 라인 버퍼들을 감소시키는 것이 여전히 바람직하다.

[0177] VTM-4.0에서, 도 13에 도시된 바와 같이, 필요한 라인 버퍼들의 총수는 루마 성분의 경우 11.25 라인이다. 라인 버퍼 요구사항에 대한 설명은 다음과 같다: 결정 및 필터링이 첫 번째 CTU로부터의 라인들 K, L, M, M 및 하단 CTU로부터의 라인들 O, P를 필요로 하기 때문에 CTU 에지와 중첩하는 수평 에지의 디블로킹이 수행될 수 없다. 따라서, CTU 경계와 중첩하는 수평 에지들의 디블로킹은 하위 CTU가 올 때까지 연기된다. 따라서, 라인들 K, L, M, N에 대해, 재구성된 루마 샘플들은 라인 버퍼(4개의 라인)에 저장되어야 한다. 이어서 라인들 A 내지 J에 대해 SAO 필터링이 수행될 수 있다. 디블로킹이 라인 K에 있는 샘플들을 변경하지 않기 때문에 라인 J는 SAO 필터링될 수 있다. 라인 K의 SAO 필터링의 경우, 에지 오프셋 분류 결정은 (0.25 루마 라인인) 라인 버퍼에만 저장된다. ALF 필터링은 라인들 A 내지 F에 대해서만 수행할 수 있다. 도 13에 도시된 바와 같이, 각각의 4x4 블록에 대해 ALF 분류가 수행된다. 각각의 4x4 블록 분류는 8x8 크기의 활성도 창을 필요로 하며, 이는 차례로 그래디언트를 결정하기 위해 1d 라플라시안을 계산하기 위해 9x9 창을 필요로 한다.

[0178] 따라서, 라인들 G, H, I, J와 중첩하는 4x4 블록의 블록 분류를 위해, 가상 경계 아래의 SAO 필터링된 샘플들이 필요하다. 추가적으로, ALF 분류를 위해 라인들 D, E, F의 SAO 필터링된 샘플들이 필요하다. 더욱이, 라인 G의 ALF 필터링은 위쪽 라인들로부터의 3 개의 SAO 필터링된 라인 D, E, F를 필요로 한다. 따라서, 총 라인 버퍼 요구사항은 다음과 같다.

[0179] - 라인들 K 내지 N(수평 DF 픽셀들): 4 개의 라인

[0180] - 라인들 D 내지 J(SAO 필터링된 픽셀들): 7 개의 라인

[0181] - 라인 J와 라인 K 사이의 SAO 에지 오프셋 분류기 값들: 0.25 라인

[0182] 따라서, 필요한 루마 라인들의 총수는 7 + 4 + 0.25 = 11.25이다.

[0183] 유사하게, 크로마 성분의 라인 버퍼 요구사항은 도 14에 예시되어 있다. 크로마 성분에 대한 라인 버퍼 요구사항은 6.25개의 라인인 것으로 평가된다.

[0184] 도 13은 루마 성분에 대한 VTM-4.0에서의 루프 필터 라인 버퍼 요구사항을 보여준다.

[0185] 도 14은 크로마 성분에 대한 VTM-4.0에서의 루프 필터 라인 버퍼 요구사항을 보여준다.

[0186] SAO 및 ALF의 라인 버퍼 요구사항들을 제거하기 위해, 최신 VVC에서 가상 경계(VB)의 개념이 도입된다. 수정된 블록 분류 및 필터링이 수평 CTU 경계들 근처의 샘플들에 대해 채용된다. 도 13에 도시된 바와 같이, VB들은 N개의 픽셀만큼 상향 시프트된 수평 LCU 경계들이다. 각각의 LCU에 대해, SAO 및 ALF는 하위 LCU가 오기 전에 VB 위쪽의 픽셀들을 프로세싱할 수 있지만, 하위 LCU가 올 때까지 VB 아래의 픽셀들을 프로세싱할 수 없으며, 이는 DF에 의해 야기된다. 하드웨어 구현 비용을 고려하여, 제안된 VB와 수평 LCU 경계 사이의 공간은 루마 성분의 경우 4개의 픽셀로서 설정되고(예를 들면, 도 13 또는 도 15에서 N=4) 크로마 성분의 경우 2개의 픽셀로서 설정된다(예를 들면, N=2).

[0187] 도 15는 가상 경계들에서 수정된 블록 분류를 보여준다.

[0188] 수정된 블록 분류는 도 16에 도시된 것과 같이 루마 성분에 대해 적용된다. 가상 경계 상단의 4x4 블록의 1D 라플라스 그래디언트 계산을 위해, 오직 가상 경계 상단의 샘플들만이 사용된다. 유사하게, 가상 경계 하단의 4x4 블록의 1D 라플라스 그래디언트 계산을 위해, 오직 가상 경계 하단의 샘플들만이 사용된다. 활성화 값 A의 양자화는 1D 라플라스 그래디언트 계산에 사용되는 샘플들의 감소된 수를 고려하여 그에 맞게 스케일링 된다.

[0189] 필터링 처리를 위해, 가상 경계들에서의 미러링된 (대칭) 패딩 동작이 루마 및 크로마 성분들 모두에 대해 사용된다. 도 16에 도시된 바와 같이, 필터링되는 샘플이 가상 경계 아래에 위치하는 경우, 가상 경계 위에 위치하는 이웃 샘플들이 패딩된다. 한편, 다른 변들의 상응하는 샘플들 또한 대칭적으로 패딩된다.

[0190] 도 16은 가상 경계들에서의 루마 성분을 위한 수정된 ALF 필터링을 도시한다.

[0191] 다른 예로서, (i, j)에 위치한 하나의 샘플이 패딩되는 경우(예를 들어, 도 17B에서 점선으로 표시된 P0A), (m, n)에 위치하는 대응 샘플(예를 들어, 도 17B에서 동일한 필터 계수를 공유하는 점선으로 표시된 P3B) 또한 패딩되며, 도 17A 내지 도17C에 도시되는 것과 같이 해당 샘플도 이용가능하다.

[0192] 도 17A는 (각 변별로) 패딩되어야 할 VB 위/아래에 요구되는 하나의 라인을 도시한다.

[0193] 도 17B는 (각 변별로) 패딩되어야 할 VB 위/아래에 요구되는 두 개의 라인을 도시한다.

[0194] 도 17C는 (각 변별로) 패딩되어야 할 VB 위/아래에 요구되는 세 개의 라인을 도시한다.

[0195] 도 27은 가상 경계에서의 수정된 루마 ALF 필터링의 예들을 도시한다.

[0196] 수평 CTU 경계들에 사용되는 미러링된 (대칭) 패딩 방법과는 달리, 경계들을 가로지르는 필터가 비활성화되는 경우, 반복적인 (단변) 패딩 프로세스가 슬라이스, 타일 및 서브픽처 경계들에 대해 적용된다. 상기 반복적 (단변) 패딩 프로세스는 픽처 경계에서도 적용된다. 패딩된 샘플들은 분류 및 필터링 프로세스에 모두 사용된다. 도 18은 픽처/서브픽처/슬라이스/타일 경계에서 루마 ALF 필터링에 대한 반복 패딩 방법의 일 예를 도시한다.

[0197] 도 18은 픽처/서브픽처/슬라이스/타일 경계에서 루마 ALF 필터링을 위한 반복 패딩의 예들을 도시한다.

[0198] ### 2.7. 360-degree video coding

[0199] VTM5에서의 수평 랩 어라운드 모션 보상은 ERP(equi-rectangular) 투영 포맷의 재구성된 360도 비디오의 시각적 품질을 개선시키도록 설계된 360 전용 코딩 툴이다. 종래의 모션 보상에서는, 모션 벡터가 참조 픽처의 픽처 경계들 너머의 샘플들을 참조할 때, 대응하는 픽처 경계 상의 해당 최근접 이웃들로부터 복사하는 것에 의해 범위 밖의(out-of-bounds) 샘플들의 값들을 도출하기 위해 반복적인 패딩이 적용된다. 360도 비디오의 경우, 이 반복적인 패딩 방법은 적합하지 않으며, 재구성된 뷰포트 비디오에서 "이음매 아티팩트(seam artefact)"라는 시각적 아티팩트를 야기할 수 있다. 360도 비디오는 구면 상에서 캡처되고 본질적으로 "경계"를 갖지 않기 때문에, 투영된 도메인에서 참조 픽처의 경계들을 벗어난 참조 샘플들은 항상 구면 도메인에서 이웃 샘플들로부터 획득될 수 있다. 일반적인 투영 포맷의 경우, 2D에서 3D로 및 3D에서 2D로의 좌표 변환은 물론 분수 샘플 위치들에 대한 샘플 보간을 수반하기 때문에, 구면 도메인에서 대응하는 이웃 샘플들을 도출하기 어려울 수 있다. 이 문제는 ERP 투영 포맷의 좌측 및 우측 경계들에 대해 훨씬 더 간단한데, 그 이유는 좌측 픽처 경계 외부의 구면 이웃들이 우측 픽처 경계 내부의 샘플들로부터 획득될 수 있고 그 반대도 마찬가지이기 때문이다.

[0200] 도 19는 VVC에서의 수평 랩 어라운드 모션 보상의 일 예를 도시한다.

[0201] **수평 랩 어라운드 모션 보상 프로세스는 도 19에 묘사된 바와 같다.** 참조 블록의 일부가 투영된 도메인에서 참조 픽처의 좌측(또는 우측) 경계 외부에 있을 때, 반복적인 패딩 대신에, "경계 외부" 부분이 투영된 도메인에서 우측(또는 좌측) 경계 쪽에 있는 참조 픽처의 대응하는 구면 이웃들로부터 취해진다. **반복적인 패딩은 상단 및 하단 픽처 경계들에 대해서만 사용된다.** 도 19에 도시된 바와 같이, 수평 랩 어라운드 모션 보상은 360도 비디오 코딩에서 종종 사용되는 비규범적(non-normative) 패딩 방법과 결합될 수 있다. VVC에서, 이것은 패딩 이전에 ERP 픽처 폭으로 설정되어야 하는 랩 어라운드 오프셋을 나타내기 위해 상위 레벨 신택스 요소를 시그널링하는 것에 의해 달성되며; 이 신택스는 그에 따라 수평 랩 어라운드의 위치를 조정하는 데 사용된다. 이 신택스는 좌측 및 우측 픽처 경계들에서의 특정 패딩 양에 의해 영향을 받지 않으며, 따라서, 예를 들면, 좌측 및 우측 패딩이 상이할 때, ERP 픽처의 비대칭 패딩을 자연스럽게 지원한다. 수평 랩 어라운드 모션 보상은 참조 샘플들이 참조 픽처의 좌측 및 우측 경계들 외부에 있을 때 모션 보상에 대한 보다 의미 있는 정보를 제공한다.

[0202] 복수의 면들로 구성된 투영 포맷들의 경우, 어떤 종류의 콤팩트한 프레임 패킹 배열이 사용되더라도, 프레임 패킹된 픽처에서 2개 이상의 인접한 면 사이에 불연속부들이 나타난다. 예를 들어, 도 20에 도시된 3x2 프레임 패킹 구성을 고려할 때, 상반부에 있는 3개의 면은 3D 지오메트리에서 연속적이고, 하반부에 있는 3개의 면은 3D 지오메트리에서 연속적이지만, 프레임 패킹된 픽처의 상반부 및 하반부는 3D 지오메트리에서 불연속적이다. 인루프 필터링 동작들이 이러한 불연속부에 걸쳐 수행되는 경우, 재구성된 비디오에서 면 이음매 아티팩트들이

보일 수 있다.

[0203] 면 이음매 아티팩트들 (face seam artifacts)을 완화시키기 위해, 프레임 패킹된 픽처에서의 불연속부들에 걸쳐 인루프 필터링 동작들이 비활성화될 수 있다. 인루프 필터링 동작들이 비활성화되는 **수직 및/또는 수평 가상 경계들**을 시그널링하기 위해 신택스가 제안되었다. 연속적인 면들의 각각의 세트에 대해 하나씩, 2개의 타일을 사용하는 것과 비교하여 그리고 타일들에 걸친 인루프 필터링 동작들을 비활성화시키기 위해, 제안된 시그널링 방법은, 면 크기가 CTU 크기의 배수일 것을 요구하지 않기 때문에, 더 유연하다.

[0204] 도 20은 3x2 레이아웃에서 HEC의 이미지를 도시한다.

[0205] **2.8. JVET-P0080: CE5-2.1, CE5-2.2: 크로스 컴포넌트 적응적 루프 필터**

[0206] 도 21A는 다른 루프 필터들에 대한 CC-ALF [1]의 배치를 도시한다. CC-ALF는 도 21B의 선형 다이아몬드 모양 필터를 각 크로마 성분에 대한 루마 채널에 적용하여 작동하며, 이는 다음과 같이 표현된다.

[0207] $$\Delta I_i(x,y) = \sum_{(x_0,y_0)\in S_i} I_0(x_C + x_0, y_C + y_0) c_i(x_0,y_0),$$

[0208] 여기서

[0209] (x,y)는 개선되는 크로마 성분 i 위치이다.

[0210] (x_C,y_C)는 (x,y)에 기반한 루마 위치이다.

[0211] S_i는 크로마 성분 i에 대한 루마 내 필터 서포트이다.

[0212] $c_i(x_0,y_0)$는 필터 계수들을 나타낸다. (2-14)

[0213] 도 21A는 다른 루프 필터들에 대한 CC-ALF의 예시적 배치를 도시한다. 도 21B는 다이아몬드 모양 필터를 도시한다.

[0214] 서포트 영역이 집중되는 루마 위치 (x_C,y_C)는 루마 및 크로마 평면들 사이이의 공간적 스케일링 팩터에 기반하여 계산된다. 모든 필터 계수들은 APS에서 전송되고 8 비트 동적 범위를 가진다. APS는 슬라이스 헤더에서 참조된다. 슬라이스의 각 크로마 성분에 대해 사용되는 CC-ALF 계수들 또한 시간적 서브 레이어에 해당하는 버퍼에 저장된다. 시간적 서브레이어 필터 계수들의 이러한 세트들의 재사용은 슬라이스 레벨 플래그들을 사용하여 촉진된다. CC-ALF 필터들의 적용은 가변 블록 크기 (즉, 16x16, 32x32, 64x64, 128x128)에 기반하여 제어되고 샘플들의 각 블록에 대해 수신된 문맥 코딩된 플래그로 시그널링된다. CC-ALF 활성화 플래그와 함께 블록 크기가 각 크로마 성분에 대한 슬라이스 레벨에서 수신된다. 수평 가상 경계들에 대한 경계 패딩은 반복을 이용한다. 나머지 경계들에 대해, 동일한 유형의 패딩이 정규 ALF에 대해서와 같이 사용된다.

[0215] **2.8.1. JVET-P0080의 CC-ALF 관련 사양**

[0216] **x.x.x.x 크로마 샘플들의 블록에 대한 크로스 컴포넌트 필터링 프로세스**

[0217] 이 프로세스의 입력들은 아래와 같다.

[0218] - 루마 적응적 루프 필터링 프로세스 이전의 재구성된 루마 픽처 샘플 어레이 $recPicture_L$,

[0219] - 필터링된 재구성된 크로마 픽처 샘플 어레이 $alfPicture_C$,

[0220] - 현재 픽처의 좌측 상단 샘플에 상대적인 크로마 샘플들의 현재 블록의 좌측 상단 샘플을 명시하는 크로마 위치 (xC, yC),

[0221] - 크로마 샘플들의 블록의 폭 ccAlfWidth,

[0222] - 크로마 샘플들의 블록의 높이 ccAlfHeight,

[0223] - 크로스 컴포넌트 필터 계수들 CcAlfCoeff[j] (j = 0..13)

[0224] 이 프로세스의 출력은 수정된 필터링된 재구성된 크로마 픽처 샘플 어레이ccAlfPicture이다.

[0225] 코딩 트리 블록 루마 위치(xCtb, yCtb)는 아래와 같이 도출된다.

[0226] $$xCtb = (((xC * SubWidthC) >> CtbLog2SizeY) << CtbLog2SizeY \quad (8\text{-}1229)$$
$$yCtb = (((yC * SubHeightC) >> CtbLog2SizeY) << CtbLog2SizeY \quad (8\text{-}1229)$$

[0227] 필터링된 재구성된 크로마 샘플들ccAlfPicture[xC + x][yC + y]의 도출을 위해, 샘플들의 현재 크로마 블록 내의 각 재구성된 크로마 샘플 $alfPicture_C$[xC + x][yC + y] (x = 0..ccAlfWidth - 1, y = 0..ccAlfHeight - 1)은 아래와 같이 필터링된다.

[0228] - 크로마 위치(xC + x, yC + y)의 현재 크로마 샘플에 상응하는 루마 위치(xL, yL)는 ((xC + x) * SubWidthC, (yC + y) * SubHeightC)와 같게 설정된다.

[0229] - 어레이 $recPicture_L$ 내부의 루마 위치들(h_{xL+i}, v_{yL+j}) (i = -2..2, j = -2..3)은 아래와 같이 도출된다.

[0230] - 만약 pps_loop_filter_across_virtual_boundaries_disabled_flag 가 1과 같고, 임의의 n = 0..pps_num_ver_virtual_boundaries - 1 에 대해, PpsVirtualBoundariesPosX[n] % CtbSizeY 가 0과 같지 않고, xL - PpsVirtualBoundariesPosX[n]가 0 이상 3 미만인 경우, 다음이 적용된다.

[0231] $$h_{xL+i} = Clip3(PpsVirtualBoundariesPosX[n], pic_width_in_luma_samples - 1, xL + i) \quad (8\text{-}1229)$$

[0232] - 그렇지 않고, 만약

[0233] pps_loop_filter_across_virtual_boundaries_disabled_flag가 1과 같고, 임의의 n = 0..pps_num_ver_virtual_boundaries - 1 에 대해, PpsVirtualBoundariesPosX[n] % CtbSizeY가 0 과 같지 않고, PpsVirtualBoundariesPosX[n]- xL가 0 보다 크고 4 미만이면, 다음이 적용된다

[0234] $$h_{x+i} = Clip3(0, PpsVirtualBoundariesPosX[n] - 1, xL + i) \quad (8\text{-}1230)$$

[0235] - 그렇지 않은 경우, 다음이 적용된다.

[0236] $$h_{x+i} = Clip3(0, pic_width_in_luma_samples - 1, xL + i) \quad (8\text{-}1231)$$

[0237] - 만약

[0238] pps_loop_filter_across_virtual_boundaries_disabled_flag가 1과 같고, 임의의 n = 0..pps_num_hor_virtual_boundaries - 1에 대해, PpsVirtualBoundariesPosY[n] % CtbSizeY가 0과 같지 않고 yL - PpsVirtualBoundariesPosY[n]가 0 이상 3 미만인 경우, 다음이 적용된다.

[0239] $$v_{y+j} = Clip3(PpsVirtualBoundariesPosY[n], pic_height_in_luma_samples - 1, yL + j) \quad (8\text{-}1232)$$

[0240] - 그렇지 않고, 만약

[0241] pps_loop_filter_across_virtual_boundaries_disabled_flag가 1과 같고, 임의의 n = 0..pps_num_hor_virtual_boundaries - 1에 대해, PpsVirtualBoundariesPosY[n] % CtbSizeY가 0과 같지 않고 PpsVirtualBoundariesPosY[n] - yL이 0 보다 크고 4 보다 작으면, 다음이 적용된다.

[0242] $$v_{y+j} = Clip3(0, PpsVirtualBoundariesPosY[n] - 1, yL + j) \quad (8\text{-}1233)$$

[0243] - 그렇지 않은 경우, 다음이 적용된다.

[0244] $$v_{y+j} = Clip3(0, pic_height_in_luma_samples - 1, yL + j) \quad (8\text{-}1234)$$

[0245] - 변수 clipLeftPos, clipRightPos, clipTopPos 및 clipBottomPos는 (xCtb, yCtb) 및 (xL - xCtb, yL - yCt

b)를 입력들로 하여 8.8.5.5 절에 명시된 바와 같이 ALF 경계 위치 도출 프로세스를 호출하는 것에 의해 도출된다.

[0246] - 수직 샘플 위치 오프셋 yM2, yM1, yP1, yP2 및 yP3는 수직 루마 샘플 위치 yL, clipLeftPos 및 clipRightPos에 따라 표 2-6에 명시된다.

[0247] - 수평 샘플 위치 오프셋 xM1, xM2, xP1 및 xP2는 수평 루마 샘플 위치 xL, clipLeftPos 및 clipRightPos에 따라 표 2-7에 명시된다.

[0248] - 변수 curr는 다음과 같이 도출된다.

[0249] $$curr = alfPicture_C[xC + x, yC + y] \quad (8\text{-}1286)$$

[0250] - 크로스 컴포넌트 필터 계수들의 어레이 f[j]는 j = 0..13로 아래와 같이 도출된다.

[0251] $$f[j] = CcAlfCoeff[j] \quad (8\text{-}1287)$$

[0252] - 변수 sum은 아래와 같이 도출된다.

$$\begin{aligned} sum = {} & f[0] * recPicture_L[h_x, v_{y+yM2}] + \\ & f[1] * recPicture_L[h_{x+xM1}, v_{y+yM1}] + \\ & f[2] * recPicture_L[h_x, v_{y+yM1}] + \\ & f[3] * recPicture_L[h_{x+xP1}, v_{y+yM1}] + \\ & f[4] * recPicture_L[h_{x+xM2}, v_y] + \\ & f[5] * recPicture_L[h_{x+xM1}, v_y] + \\ & f[6] * recPicture_L[h_x, v_y] + \\ & f[7] * recPicture_L[h_{x+xP1}, v_y] + \\ & f[4] * recPicture_L[h_{x+xP2}, v_y] + \\ & f[4] * recPicture_L[h_{x+xM2}, v_{y+yP1}] + \\ & f[8] * recPicture_L[h_{x+xM1}, v_{y+yP1}] + \\ & f[9] * recPicture_L[h_x, v_{y+yP1}] + \\ & f[10] * recPicture_L[h_{x+xP1}, v_{y+yP1}] + \\ & f[4] * recPicture_L[h_{x+xP2}, v_{y+yP1}] + \\ & f[11] * recPicture_L[h_{x+xM1}, v_{y+yP2}] + \\ & f[12] * recPicture_L[h_x, v_{y+yP2}] + \\ & f[13] * recPicture_L[h_{x+xP1}, v_{y+yP2}] + \\ & f[0] * recPicture_L[h_x, v_{y+yP3}] \end{aligned} \quad (8\text{-}1289)$$

[0253] $$sum = curr + (sum + 64) >> 7) \quad (8\text{-}1290)$$

[0254] - 수정된 필터링된 재구성된 크로마 픽처 샘플 어레이 ccAlfPicture[xC + x][yC + y]는 다음과 같이 도출된다.

[0255] $$ccAlfPicture[xC + x][yC + y] = Clip3(0, (1 << BitDepth_C) - 1, sum) \quad (8\text{-}1291)$$

[0256] 표 2-6. 수직 루마 샘플 위치 yL, clipTopPos 및 clipBottomPos에 따른 yM1, yM2, yP1, yP2 및 yP3의 사양

Condition	yM2	yM1	yP1	yP2	yP3
yL = = clipTopPos + 1	-1	-1	1	2	3
yL = = clipTopPos	0	0	1	2	3
yL = = clipBottomPos - 1	-2	-1	0	0	0
yL = = clipBottomPos - 2	-2	-1	1	1	1
yL = = clipBottomPos - 3	-2	-1	1	2	2
Otherwise	-2	-1	1	2	3

[0257]

[0258] 표 2-7. 수평 루마 샘플 위치 xL, clipLeftPos 및 clipRightPos에 따른 xM1, xM2, xP1 및 xP2의 사양

Condition	xM2	xM1	xP1	xP2
xL = = clipLeftPos + 1	-1	-1	1	2
xL = = clipLeftPos	0	0	1	2
xL = = clipRightPos - 1	-2	-1	0	0
xL = = clipRightPos - 2	-2	-1	1	1
Otherwise	-2	-1	1	2

[0259]

[0260] **2.8.2. JVET-P0080에서 가상 경계에서의 패딩 방법**

[0261] 루마 ALF/크로마 ALF와 유사하게, JVET-P0080에서 CC-ALF에 대한 ALF 가상 경계에서 반복 패딩이 활용된다. 도 22에 도시된 것과 같이, 만약 ALF 가상 경계 위 또는 아래의 루마 샘플들이 가용하지 않는 경우, 최근접 샘플 라인은 패딩에 대해 활용된다. 상세한 패딩 방법은 표 2-6에서도 제시된다.

[0262] **2.9. JVET-P1008: CE5 관련: CC-ALF의 설계에 관하여**

[0263] JVET-O0636 [1] 및 CE5-2.1 [2]에서, 크로스 컴포넌트 적응적 루프 필터(CC-ALF: Cross Component Adaptive Loop Filter)가 도입되어 연구되었다. 상기 필터는 선형 필터를 사용하여 루마 샘플 값들을 필터링하고 동위치의(co-located) 필터링된 출력으로부터 크로마 채널들에 대한 잔여 정정을 생성한다. 상기 필터는 기존 루마 ALF와 나란히 작동하도록 설계되었다.

[0264] CC-ALF 설계는 단순화됨과 동시에 기존 ALF와 더 잘 정렬된다. 상기 설계는 8개의 고유한 계수들을 가지는 3x4 다이아몬드 모양을 사용한다. 이는 CE5-2.1에서 연구된 5x6 설계와 비교하여 배수들의 수를 43% 감소시킨다. CTU의 크로마 성분에 대해 크로마 ALF 또는 CC-ALF를 활성화시키는 제약이 부가되는 경우, 우리는 픽셀 당 배수의 수를 16(현재 ALF는 15)으로 제한한다. 필터 계수 동적 범위는 서명된 6 비트로 제한된다. 제안되고 CE5-2.1 해결책에 대한 필터들의 예시는 도 23에 제시된다.

[0265] 기존 ALF 설계와 더 잘 정렬되도록, 필터 계수들은 APS에서 시그널링된다. 최대 4개의 필터들의 지원되며, 필터 선택은 CTU 레벨에서 지시된다. 대칭 라인 선택은 ALF와 더 조화롭도록 가상 경제에서 사용된다. 마지막으로, 정정 출력에 의해 필요한 스토리지의 양을 제한하기 위해, CC-ALF 잔여 출력은 $2^{BitDepthC-1}$ 내지 $2^{BitDepthC-1}$-1으로 클리핑된다.

[0266] JVET-P1008에서 CC-ALF에 관한 사양

[0267] **x.x.x.x 크로마 샘플들의 블록에 대한 크로스 컴포넌트 필터링 프로세스**

[0268] 이 프로세스의 입력들은 아래와 같다.

[0269] - 루마 적응적 루프 필터링 프로세스 이전의 재구성된 루마 픽처 샘플 어레이 $recPicture_L$,

[0270] - 필터링된 재구성된 크로마 픽처 샘플 어레이 $alfPicture_C$,

[0271] - 현재 픽처의 좌측 상단 샘플에 상대적인 현재 크로마 코딩 트리 블록의 좌측 상단 샘플을 명시하는 크

로마 위치 (xCtbC, yCtbC)

[0272] - 크로마 샘플들의 블록의 폭 ccAlfWidth

[0273] - 크로마 샘플들의 블록의 높이 ccAlfHeight

[0274] - 크로스 컴포넌트 필터 계수들 CcAlfCoeff[j] (j = 0..7)

[0275] 이 프로세스의 출력은 수정된 필터링된 재구성된 크로마 픽처 샘플 어레이 ccAlfPicture이다.

[0276] 코딩 트리 블록 루마 위치(xCtb, yCtb)는 아래와 같이 도출된다

[0277]
$$xCtb = ((xCtbC * SubWidthC) >> CtbLog2SizeY) << CtbLog2SizeY \quad (8\text{-}1229)$$
$$yCtb = ((yCtbC * SubHeightC) >> CtbLog2SizeY) << CtbLog2SizeY \quad (8\text{-}1229)$$

[0278] 필터링된 재구성된 크로마 샘플들ccAlfPicture[xCtbC + x][yCtbC + y] 의 도출을 위해, 샘플들의 현재 크로마 블록 내의 각 재구성된 크로마 샘플 $alfPicture_C$[xCtbC + x][yCtbC + y] (x = 0..ccAlfWidth - 1, y = 0..ccAlfHeight - 1)은 아래와 같이 필터링된다.

[0279] - 크로마 위치 (xCtbC + x, yCtbC + y)의 현재 크로마 샘플에 상응하는 루마 위치(xL, yL)는 ((xCtbC + x) * SubWidthC, (yCtbC + y) * SubHeightC)와 같게 설정된다.

[0280] - 어레이 $recPicture_L$ 내부의 루마 위치들(h_{xL+i}, v_{yL+j}) (i = -1..1, j = -1..2)은 아래와 같이 도출된다.

[0281] - 만약

[0282] pps_loop_filter_across_virtual_boundaries_disabled_flag가 1과 같고, 임의의 n = 0..pps_num_ver_virtual_boundaries - 1 에 대해, PpsVirtualBoundariesPosX[n] % CtbSizeY 가 0과 같지 않고 xL - PpsVirtualBoundariesPosX[n]가 0 이상 3 미만이라면, 다음이 적용된다.

[0283]
$$h_{xL+i} = Clip3(PpsVirtualBoundariesPosX[n], pic_width_in_luma_samples - 1, xL + i) \quad (8\text{-}1229)$$

[0284] - 그렇지 않고, 만약

[0285] pps_loop_filter_across_virtual_boundaries_disabled_flag가 1과 같고, 임의의 n = 0..pps_num_ver_virtual_boundaries - 1에 대해, PpsVirtualBoundariesPosX[n] % CtbSizeY가 0과 같지 않고 PpsVirtualBoundariesPosX[n]- xL가 0 보다 크고 4 보다 작으면, 다음이 적용된다.

[0286]
$$h_{x+i} = Clip3(0, PpsVirtualBoundariesPosX[n] - 1, xL + i) \quad (8\text{-}1230)$$

[0287] - 그렇지 않은 경우, 다음이 적용된다.

[0288]
$$h_{x+i} = Clip3(0, pic_width_in_luma_samples - 1, xL + i) \quad (8\text{-}1231)$$

[0289] - 만약

[0290] pps_loop_filter_across_virtual_boundaries_disabled_flag가 1과 같고, 임의의 n = 0..pps_num_hor_virtual_boundaries - 1에 대해, PpsVirtualBoundariesPosY[n] % CtbSizeY가 1과 같지 않고 yL - PpsVirtualBoundariesPosY[n]가 0 이상 3 미만인 경우, 다음이 적용된다.

[0291]
$$v_{y+j} = Clip3(PpsVirtualBoundariesPosY[n], pic_height_in_luma_samples - 1, yL + j) \quad (8\text{-}1232)$$

[0292] - 그렇지 않고, 만약

[0293] pps_loop_filter_across_virtual_boundaries_disabled_flag가 1과 같고, 임의의 n = 0..pps_num_hor_virtual_boundaries - 1에 대해, PpsVirtualBoundariesPosY[n] % CtbSizeY가 0과 같지

않고 PpsVirtualBoundariesPosY[n] - yL가 0 보다 크고 4 보다 작은 경우, 다음이 적용된다.

[0294] $$v_{y+j} = \text{Clip3}(0, \text{PpsVirtualBoundariesPosY}[n] - 1, yL + j) \quad (8\text{-}1233)$$

[0295] - 그렇지 않은 경우, 다음이 적용된다.

[0296] $$v_{y+j} = \text{Clip3}(0, \text{pic_height_in_luma_samples} - 1, yL + j) \quad (8\text{-}1234)$$

[0297] - 변수 clipLeftPos, clipRightPos, clipTopPos 및 clipBottomPos는 (xCtb, yCtb) 및 (xL - xCtb, yL - yCtb)를 입력들로 하여 8.8.5.5 절에 명시된 바와 같이 ALF 경계 위치 도출 프로세스를 호출하는 것에 의해 도출된다.

[0298] - 수직 샘플 위치 오프셋 yM1, yP1 및 yP2는 수직 루마 샘플 위치 yL, clipLeftPos 및 clipRightPos에 따라 표 2-8에 명시된다.

[0299] - 수평 샘플 위치 오프셋 xM1및 xP1은 수평 루마 샘플 위치 xL, clipLeftPos 및 clipRightPos에 따라 표 2-9에 명시된다.

[0300] - 변수 curr은 다음과 같이 도출된다

[0301] $$\text{curr} = \text{alfPicture}_C[\text{xCtbC} + x, \text{yCtbC} + y] \quad (8\text{-}1286)$$

[0302] - 크로스 컴포넌트 필터 계수들의 어레이 f[j]는 j = 0..7로 아래와 같이 도출된다.

[0303] $$f[j] = \text{CcAlfCoeff}[j] \quad (8\text{-}1287)$$

[0304] - 변수 sum은 다음과 같이 도출된다.

$$\begin{aligned}\text{sum} = {} & f[0] * \text{recPicture}_L[h_x, v_{y+yM1}] + \\ & f[1] * \text{recPicture}_L[h_{x+xM1}, v_y] + \\ & f[2] * \text{recPicture}_L[h_x, v_y] + \\ & f[3] * \text{recPicture}_L[h_{x+xP1}, v_y] + \\ & f[4] * \text{recPicture}_L[h_{x+xM1}, v_{y+yP1}] + \\ & f[5] * \text{recPicture}_L[h_x, v_{y+yP1}] + \\ & f[6] * \text{recPicture}_L[h_{x+xP1}, v_{y+yP1}] + \\ & f[7] * \text{recPicture}_L[h_x, v_{y+yP2}]\end{aligned} \quad (8\text{-}1289)$$

$$\text{sum} = \text{Clip3}(-(1 << (\text{BitDepth}_C - 1)), (1 << (\text{BitDepth}_C - 1)) - 1, \text{sum}) \quad (8\text{-}1290)$$

[0305] $$\text{sum} = \text{curr} + (\text{sum} + 64) >> (7 + (\text{BitDepth}_Y - \text{BitDepth}_C)) \quad (8\text{-}1290)$$

[0306] - 수정된 필터링된 재구성된 크로마 픽처 샘플 어레이 ccAlfPicture[xCtbC + x][yCtbC + y]는 다음과 같이 도출된다.

[0307] $$\text{ccAlfPicture}[\text{xCtbC} + x][\text{yCtbC} + y] = \text{Clip3}(0, (1 << \text{BitDepth}_C) - 1, \text{sum}) \quad (8\text{-}1291)$$

[0308] 표 2-8. 수직 루마 샘플 위치 yL, clipTopPos 및 clipBottomPos에 따른 yM1, yP1 및 yP2의 사양

Condition	yM1	yP1	yP2
yL = = clipTopPos + 1	−1	1	1
yL = = clipTopPos	0	0	1
yL = = clipBottomPos − 1	0	0	1
yL = = clipBottomPos − 2	−1	1	1
Otherwise	−1	1	2

[0309]

[0310] 표 2-9. 수평 루마 샘플 위치 xL, clipLeftPos 및 clipRightPos에 따른 xM1 및 xP1의 사양

Condition	xM1	xP1
xL = = clipLeftPos	0	0
xL = = clipRightPos − 1	0	0
xL = = clipRightPos – 2	–1	1
Otherwise	–1	1

[0311]

[0312] **2.9.1. JVET-P1008에서 가상 경계에서의 패딩 방법**

[0313] 미러링된 (대칭) 패딩은 JVET-P1008에서 CC-ALF에 대해 ALF 가상 경계에서 활용된다. 도 24에서와 같이, 만약 상기 ALF 가상 경계 위 또는 아래의 루마 샘플들이 가용하지 않는 경우, 최근접 샘플 라인은 패딩에 대해 활용되고, 해당 샘플들 또한 패딩될 필요가 있다. 상세한 패딩 방법은 표 2-9에서도 제시된다.

[0314] **2.10. JVET-P2025에서 CC-ALF의 단순화된 방법들**

[0315] **2.10.1. 대안적인 필터 모양들**

[0316] CC-ALF 필터 모양은 아래 수치와 같이 8개 또는 6개의 계수들을 가지도록 수정된다.

[0317] 도 25는 JVET-P0106에서 8개의 계수들의 CC-ALF 필터 모양을 도시한다.

[0318] 도 26은 JVET-P0173에서 6개의 계수들의 CC-ALF 필터 모양을 도시한다.

[0319] 도 27은 JVET-P0251에서 6개의 계수들의 CC-ALF 필터 모양을 도시한다.

[0320] **2.10.2. 결합 크로마 크로스 컴포넌트 적응적 필터링**

[0321] 결합 크로마 크로스 컴포넌트 적응적 필터링(JC-CCALF: Joint Chroma Cross-Component Adaptive Loop Filter)은 인코더에서 훈련된 CCALF 필터 계수들의 오직 하나의 세트만을 사용하여 하나의 필터링된 출력을 개선 신호(refinement signal)로 생성하며, 이것은 Cb 성분으로 직접적으로 가산되고 적절하게 가중치를 받은 후에 Cr 성분으로 가산될 것이다. 필터들은 CTU 레벨에서 지시되거나 블록 크기와 함께 지시되며, 이것은 슬라이스마다 시그널링된다.

[0322] 지지되는 이와 같은 크로마 블록 크기들의 범위는 최소 크로마 CTU 크기부터 현재 크로마 CTU 크기까지 이다. 최소 크로마 CTU 크기는 크로마 CTU의 가능한 최소한의 폭 및 높이 가운데 최소값, 즉, Min(32 / SubWidthC, 32 / SubHeightC)인 반면에, 현재 크로마 CTU 크기는 현재 크로마 CTU의 폭 및 높이 가운데 최소값, 즉, Min(CtbWidthC, CtbHeightC)이다. 예를 들어, 만약 CTU 크기가 최대 128x128에 설정된다면, 슬라이스의 JC-CCALF 크로마 블록 크기는 4:4:4 비디오의 경우 32x32, 64x64 그리고 128x128 중 하나일 것이며, 4:2:0 및 4:2:2 비디오의 경우 16x16, 32x32 그리고 64x64 중 하나일 것이다.

[0323] 도 28은 JC-CCALF 워크플로우의 일 예를 도시한다

[0324] **3. 본 문서에 기술된 기술적 해결수단들에 의해 해결되는 기술적 문제들**

[0325] CC-ALF를 위한 경계 패딩의 현재 설계는 다음과 같은 문제들을 가진다.

[0326] 1. CC-ALF에서 ALF 가상 경계에서의 패딩 방법은 차선의 방법인데, 이는 덜 효율적일 수 있는 패딩된 샘플들이

활용되기 때문이다.

[0327] 2. ALF 가상 경계 및 비디오 유닛 경계 (예를 들어, 픽처/서브픽처/슬라이스/타일 경계) 그리고 360도 가상 경계를 다루는 상이한 방식들, 즉, 상이한 패딩 방법들이 존재한다.

[0328] 3. ALF에서, 미러 패딩이 적용되는데, 이때 현재 샘플까지의 거리가 계산되어 어떤 대응하는 샘플이 패딩될 필요가 있는지를 판단한다. 그러나, CC-ALF에서, 특히 4:2:0의 경우, 하나의 크로마 샘플을 필터링하기 위해, 복수의 루마 샘플들이 관련된다. 어떤 대응하는 샘플이 패딩되어야 하는지를 어떻게 판단할지는 알려지지 않았다.

[0329] **4. 기술 및 실시예들의 예시적 리스트**

[0330] 아래 리스트는 일반적인 개념을 설명하기 위한 예로서 고려되어야 한다. 이러한 항목들은 좁은 의미로 해석해서는 안 된다. 또한 이러한 항목들은 임의의 방식으로 결합될 수 있다.

[0331] 본 개시에 기술된 일부 실시예들에서, 용어 "CC-ALF"는 제1 컬러 성분 (예를 들어, Cb)에서 샘플들을 개선하기 위해 제2 컬러 성분 (예를 들어, Y) or 복수의 컬러 성분들 (예를 들어, Y 및 Cr 모두)에서의 샘플 값들을 활용하는 코딩 툴을 나타낸다. 이것은 [1] 내지 [4]에서 설명된 CC-ALF 기술들에 제한되지 않는다. "해당 필터링 샘플 세트"는 필터 서포트에 포함되는 샘플들을 나타내는 데에, 즉 CC-ALF를 위해 사용될 수 있으며, 상기 "해당 필터링 샘플 세트"는 크로마 샘플의 개선/오프셋을 도출하는 데에 활용되는, 크로마 샘플의 동위치 루마 샘플 및 동위치 루마 샘플의 이웃 루마 샘플들을 나타내는 데에 사용될 수 있다.

[0332] ALF 가상 경계들을 위해 사용되는 패딩 방법은 '미러링된 패딩'으로 표시될 수 있으며, 이때 (i, j)에 위치한 제1 비가용 샘플이 패딩되고, 필터 서포트에 있는 '상기 제1 샘플의 해당 샘플'로 정의되는, ALF내의 제2 샘플 (예를 들어, 현재 루마 샘플로부터 동일한 거리를 공유하는 (m, n)에 위치한 해당 샘플) 또한, 비록 상기 제2 샘플이 가용함에도 불구하고 패딩된다.

[0333] 일 예에서, 수평 패딩이 활용되어, (x, y1)에 위치하여 패딩될 샘플은 (x, y2)에 위치한 샘플과 같은 것으로 설정되며, 이때 x1은 상기 샘플 또는 상응하는 샘플의 x 좌표를 나타내고, x2는 패딩을 위해 활용되는 샘플의 x 좌표를 나타낸다.

[0334] 일 예에서, 수평 패딩이 활용되어, (x1, y)에 위치하여 패딩될 샘플은 (x2, y)에 위한 샘플과 같은 것으로 설정되며, 이때 y1은 상기 샘플 또는 상응하는 샘플의 y 좌표를 나타내고, y2는 패딩을 위해 활용되는 샘플의 y 좌표를 나타낸다.

[0335] 픽처/서브픽처/슬라이스/타일 경계들/360도 비디오 가상 경계들, 정상 경계들(예를 들어, 상단 및 하단 경계들)에 대해 사용되는 패딩 방법은 '반복 패딩'으로 지칭될 수 있으며, 이때 사용될 하나의 샘플이 상기 경계들 밖에 있는 경우, 그것은 상기 경계 내에 있는 가용한 것으로부터 복사된다.

[0336] 본 개시에서, 이웃(인접하거나 인접하지 않는) 샘플이 상이한 비디오 처리 유닛에 (예를 들어, 현재 픽처, 현재 서브픽처, 현재 타일, 현재 슬라이스, 현재 브릭, 현재 CTU, 현재 프로세싱 유닛 (ALF 프로세싱 유닛 또는 좁은 ALF 프로세싱 유닛과 같은) 또는 기타 현재 비디오 유닛의 외부에) 위치하거나 재구성되지 않거나, 또는 크로스 필터링 비디오 처리가 허용되지 않는 경우, 상기 이웃 샘플은 "가용하지 않다"

[0337] <u>CC-ALF를 위한 ALF 가상 경계 처리</u>

[0338] 1. 가용하지 않은 루마 샘플이 ALF 가상 경계에서 패딩되도록, 미러링된 패딩이 활용되어 상기 가용하지 않은 루마 샘플 및 상기 가용하지 않은 루마 샘플에 상응하는 하나 또는 복수의 루마 샘플들을, CC-ALF에서 필터링을 위해, 도출할 수 있다. 다시 말해, 상기 가용하지 않은 루마 샘플에 상응하는 적어도 하나의 루마 샘플은 그것이 가용함에도 여전히 패딩될 필요가 있다.

[0339] a. 일 예에서, 가용하지 않은 루마 샘플에 상응하는 샘플로 판단되는 루마 샘플이 미러링된 패딩 방법을 사용하여 패딩될 수 있다.

[0340] b. 일 예에서, (해당 필터링 샘플 세트 내의) 루마 샘플이 가용하지 않은 샘플에 상응하는 샘플로 판단되는지 여부는 대표 루마 샘플에 대한 상기 샘플의 거리 또는/및 상기 대표 루마 샘플에 대한 상기 가용하지 않은 샘플의 거리에 의존적일 수 있다. 상기 대표 루마 샘플이 위치하는 중심을 C로 표시한다. 샘플들의 K개 행들 및 샘플들의 L개 열들을 활용하는 $K \times L$ 필터 모양이 CC-ALF에서 사용된다고 가정한다.

[0341] i. 일 예에서, 상기 대표 루마 샘플은 필터링되어야 하는 현재 크로마 샘플의 대응(collocated) 루마 샘플로 정의된다.

[0342] 1) 일 예에서, 현재 크로마 샘플의 대응 루마 샘플의 위치는 컬러 포맷에 의존할 수 있다.

[0343] a) 일 예에서, (x, y)에 위치한 크로마 샘플의 대응 루마 샘플은 4:2:0 크로마 포맷에서 (2x, 2y)에 위치하는 것으로 정의된다.

[0344] b) 일 예에서, (x, y)에 위치한 크로마 샘플의 대응 루마 샘플은 4:2:2 크로마 포맷에서 (2x, y)에 위치하는 것으로 정의된다.

[0345] c) 일 예에서, (x, y)에 위치한 크로마 샘플의 대응 루마 샘플은 4:4:4 크로마 포맷에서 (x, y)에 위치하는 것으로 정의된다.

[0346] ii. 일 예에서, 거리는 루마 샘플을 포함하는 행과 상기 대표 루마 샘플을 포함하는 행 사이의 수직 거리를 가리킨다. 예를 들어, 상기 거리는 루마 샘플과 상기 대표 루마 샘플 간의 절대 y 좌표 차이로 계산될 수 있다.

[0347] 1) 도 29에서와 같이, 대표 루마 샘플이 위치하는 가운데 행, 가용하지 않은 샘플의 행, 그리고 상응하는 샘플들의 행을 각각 *C*, *M* 및 *N*으로 표시하면, *M*은 *N*과 같지 않다. d(x, y)를 x와 y 사이의 절대 y 좌표 차이로 표시하고, 이는 행 x와 행 y 간의 거리를 의미한다.

[0348] iii. 일 예에서, 미러링된 패딩에서 패딩되어야 하는 해당 샘플들의 판단은 샘플들의 얼마나 많은 행들이 필터 모양에 의해 활용될 것인지에 의존할 할 수 있다.

[0349] iv. 일 예에서, 만약 가용하지 않은 샘플이 행 M에 위치하면(예를 들어, $M < C < N$ 또는 $M > C > N$), 행 N에 위치하는 샘플들은 패딩되어야 하는 해당 샘플들로 판단되며, 이때 $d(C, M) = d(N, C)$ 이다.

[0350] 1) 일 예에서, 만약 값 *K* (예를 들어, *KxL* CC-ALF 필터 모양)가 홀수이면, ALF (예를 들어, 도 16)에 대한 미러링된 패딩 방법은 CC-ALF에 대해 활용될 수 있으며, 이때 중앙 루마 샘플이 대표 루마 샘플로 선택된다.

[0351] a) 일 예에서, *K* = 5라고 가정하고, yM2 = -2, yM1 = -1, yL = 0, yP1 = 1, yP2 = 2를 각각 표 4-5에 제시되는 5개의 샘플 행들의 y 좌표를 나타낸다. ALF 가상 경계는 CtbSizeY - 4와 같다.

[0352] i. 일 예에서, ALF 가상 경계가 대표 루마 샘플 위에 있는 경우, 가용하지 않은 샘플들이 ALF 가상 경계 아래의 최근접 행을 사용하여 패딩될 수 있다. 한편으로, 해당 샘플들은 그것들이 위치하는 행 위의 최근접 행을 사용하여 패딩될 수 있다.

[0353] 1. 일 예에서, yL이 CtbSizeY - 3과 같고 행 yM2가 가용하지 않는 경우, 행 yM2에서의 샘플들 (x, yM2)은 행 yM1에서의 샘플들 (x, yM1)을 사용하여 패딩될 수 있다. 한편으로, 대응하는(corresponding) 행 yP2에서의 샘플들(x, yP2)은 행 yP1의 샘플들(x, yP1)을 사용하여 패딩될 수 있다.

[0354] 2. 일 예에서, yL이 CtbSizeY - 4와 같고 행 yM2 및 행 yM1이 가용하지 않는 경우, 행 yM2 및 yM1에서의 샘플들 (x, yM2) 및 (x, yM1)은 행 yL에서의 샘플들 (x, yL)을 사용하여 패딩될 수 있다. 한편으로, 대응하는(corresponding) 행 yP2 및 yP1에서의 샘플들(x, yP2) 및 (x, yP1)은 행 yL에서의 샘플들(x, yL)을 사용하여 패딩될 수 있다.

[0355] ii. 일 예에서, ALF 가상 경계가 대표 루마 샘플 아래에 있는 경우, 가용하지 않은 샘플들이 ALF 가상 경계 위의 최근접 행을 사용하여 패딩될 수 있다. 한편으로, 해당 샘플들은 그것들이 위치하는 행 아래의 최근접 행을 사용하여 패딩될 수 있다.

[0356] 1. 일 예에서, yL이 CtbSizeY - 6과 같고 행 yP2가 가용하지 않는 경우, 행 yP2에서의 샘플들 (x, yP2)은 행 yP1에서의 샘플들 (x, yP1)을 사용하여 패딩될 수 있다. 한편으로, 대응하는(corresponding) 행 yM2에서의 샘플들(x, yM2)은 행 yM1에서의 샘플들(x, yM1)을 사용하여 패딩될 수 있다.

[0357] 2. 일 예에서, yL이 CtbSizeY - 5와 같고 행 yP2 및 행 yP1이 가용하지 않는 경우, 행 yP2 및 yP1에서의 샘플들 (x, yP2) 및 (x, yP1)은 행 yL에서의 샘플들 (x, yL)을 사용하여 패딩될 수 있다. 한편으로, 대응하는(corresponding) 행 yM2 및 yM1에서의 샘플들(x, yM2) 및 (x, yM1)은 행 yL에서의 샘플들(x, yL)을 사용하여 패딩될 수 있다.

[0358] 2) 일 예에서, 만약 값 *K* (예를 들어, *KxL* CC-ALF 필터 모양)가 짝수이면, 도 30에 정의된 미러링된 패딩 방법이 활용될 수 있다. 가용하지 않은 샘플들이 ALF 가상 경계 위(아래)의 행 M(N)에 위치하는 경우, 그것들은 ALF 가상 경계 위(아래)의 최근접 샘플로부터 패딩된다. AFL 가상 경계 아래(위)의 행 N(M)에 위치한 해당 샘플들은 행 N(M) 위(아래)의 최근접 샘플로부터 패딩될 수 있음이 제안된다.

[0359] a) 일 예에서, *K* = 2라고 가정하고, yL = 0 및 yP1 = 1을 각각 표 4-1에 제시되는 2개의 샘플 행들의 y 좌표로 나타낸다. ALF 가상 경계는 CtbSizeY - 4와 같다.

[0360] i. 일 예에서, ALF 가상 경계가 대표 루마 샘플 위에 있는 경우, 가용하지 않은 샘플들이 ALF 가상 경계 아래의 최근접 행을 사용하여 패딩될 수 있다. 한편으로, 해당 샘플들은 그것들이 위치하는 행 위의 최근접 행을 사용하여 패딩될 수 있다.

[0361] 1. 일 예에서, yL이 CtbSizeY - 4와 같고 yL 위의 행이 가용하지 않는 경우, 행 yP1에서의 샘플들 (x, yP1)은 행 yL에서의 샘플들 (x, yL)을 사용하여 패딩될 수 있다.

[0362] ii. 일 예에서, ALF 가상 경계가 대표 루마 샘플 아래에 있는 경우, 가용하지 않은 샘플들이 ALF 가상 경계 위의 최근접 행을 사용하여 패딩될 수 있다. 한편으로, 해당 샘플들은 그것들이 위치하는 행 아래의 최근접 행을 사용하여 패딩될 수 있다.

[0363] 1. 일 예에서, yL이 CtbSizeY - 5와 같고 행 yP1이 가용하지 않는 경우, 행 yP1에서의 샘플들 (x, yP1)은 행 yL에서의 샘플들 (x, yL)을 사용하여 패딩될 수 있다.

[0364] b) 일 예에서, *K* = 4라고 가정하고, yM1 = -1, yL = 0, yP1 = 1, yP2 = 2을 각각 표 4-3에 제시되는 4개의 샘플 행들의 y 좌표로 나타낸다. ALF 가상 경계는 CtbSizeY - 4와 같다.

[0365] i. 일 예에서, ALF 가상 경계가 대표 루마 샘플 위에 있는 경우, 가용하지 않은 샘플들이 ALF 가상 경계 아래의 최근접 행을 사용하여 패딩될 수 있다. 한편으로, 해당 샘플들은 그것들이 위치하는 행 위의 최근접 행을 사용하여 패딩될 수 있다.

[0366] 1. 일 예에서, yL이 CtbSizeY - 3과 같고 yM1 위의 행이 가용하지 않는 경우, 해당 행 yP2에서의 샘플들 (x, yP2)는 행 yP1에서의 샘플들 (x, yP1)을 사용하여 패딩될 수 있다.

[0367] 2. 일 예에서, yL이 CtbSizeY - 4와 같고 yM1위의 행 및 행 yM1이 가용하지 않는 경우, 행 yM1에서의 샘플들 (x, yM1)은 행 yL에서의 샘플들 (x, yL)을 사용하여 패딩될 수 있다. 한편으로, 대응하는(corresponding) 행 yP2 및 yP1에서의 샘플들(x, yP2) 및 (x, yP1)은 행 yL에서의 샘플들(x, yL)을 사용하여 패딩될 수 있다.

[0368] ii. 일 예에서, ALF 가상 경계가 대표 루마 샘플 아래에 있는 경우, 가용하지 않은 샘플들이 ALF 가상 경계 위의 최근접 행을 사용하여 패딩될 수 있다. 한편으로, 해당 샘플들은 그것들이 위치하는 행 아래의 최근접 행을 사용하여 패딩될 수 있다.

[0369] 1. 일 예에서, yL이 CtbSizeY - 6와 같고 행 yP2이 가용하지 않는 경우, 행 yP2에서의 샘플들 (x, yP2)은 행 yP1에서의 샘플들 (x, yP1)을 사용하여 패딩될 수 있다.

[0370] 2. 일 예에서, yL이 CtbSizeY - 5와 같고 행 yP2 및 행 yP1이 가용하지 않는 경우, 행 yP2 및 yP1에서의 샘플들 (x, yP2) 및 (x, yP1)은 행 yL에서의 샘플들 (x, yL)을 사용하여 패딩될 수 있다. 한편으로, 대응하는(corresponding) 행 yM1에서의 샘플들(x, yM1)은 행 yL에서의 샘플들 (x, yL)을 사용하여 패딩될 수 있다.

[0371] c) 일 예에서, *K* = 6이라 가정하고, yM2 = -2, yM1 = -1, yL = 0, yP1 = 1, yP2 = 2, yP3 =3을 각각 표 4-6에 제시되는 6개의 샘플 행들의 y 좌표로 나타낸다. ALF 가상 경계는 CtbSizeY - 4와 같다.

[0372] i. 일 예에서, ALF 가상 경계가 대표 루마 샘플 위에 있는 경우, 가용하지 않은 샘플들이 ALF 가상 경계 아래의 최근접 행을 사용하여 패딩될 수 있다. 한편으로, 해당 샘플들은 그것들이 위치하는 행 위의 최근접 행을 사용하여 패딩될 수 있다.

[0373] 1. 일 예에서, yL이 CtbSizeY - 2와 같고 행 yM2가 가용하지 않는 경우, 행 yP3에서의 샘플들 (x, yP3)은 행 yP2에서의 샘플들 (x, yP2)를 사용하여 패딩될 수 있다.

[0374] 2. 일 예에서, yL이 CtbSizeY - 3과 같고, yM2 위의 행 및 행 yM2가 가용하지 않는 경우, 행 yM2에서의 샘플들 (x, yM2)는 행 yM1에서의 샘플들 (x, yM1)을 사용하여 패딩될 수 있다. 한편으로, 대응하는(corresponding) 행 yP3 및 yP2에서의 샘플들 (x, yP3) 및 (x, yP2)는 행 yP1의 샘플들(x, yP1)을 사용하여 패딩될 수 있다.

[0375] 3. 일 예에서, yL이 CtbSizeY - 4와 같고 yM2위의 행, yM2 및 yM1이 가용하지 않는 경우, 행 yM2 및 yM1에서의 샘플들 (x, yM2) 및 (x, yM1)은 행 yL에서의 샘플들 (x, yL)을 사용하여 패딩될 수 있다. 한편으로, 대응하는(corresponding) 행 yP3, yP2 및 yP1에서의 샘플들 (x, yP2), (x, yP2) 및 (x, yP1)은 행 yL에서의 샘플들(x, yL)을 사용하여 패딩될 수 있다.

[0376] ii. 일 예에서, ALF 가상 경계가 대표 루마 샘플 아래에 있는 경우, 가용하지 않은 샘플들이 ALF 가상 경계 위의 최근접 행을 사용하여 패딩될 수 있다. 한편으로, 해당 샘플들은 그것들이 위치하는 행 아래의 최근접 행을 사용하여 패딩될 수 있다.

[0377] 1. 일 예에서, yL이 CtbSizeY - 7와 같고 행 yP3이 가용하지 않는 경우, 행 yP3에서의 샘플들 (x, yP3)은 행 yP2에서의 샘플들 (x, yP2)을 사용하여 패딩될 수 있다.

[0378] 2. 일 예에서, yL이 CtbSizeY - 6과 같고, 행 yP3 및 yP2가 가용하지 않는 경우, 행 yP3 및 yP2에서의 샘플들 (x, yP3) 및 (x, yP2)는 상기 행에서의 샘플들 (x, yP1)을 사용하여 패딩될 수 있다. 한편으로, 대응하는(corresponding) 행 yM2에서의 샘플들 (x, yM2)는 행 yM1에서의 샘플들(x, yM1)을 사용하여 패딩될 수 있다.

[0379] 3. 일 예에서, yL이 CtbSizeY - 5와 같고 행 yP3, yP2 및 yP1이 가용하지 않는 경우, 행 yP3, yP2 및 yP1에서의 샘플들 (x, yP3), (x, yP2) 및 (x, yP1)은 행 yL에서의 샘플들 (x, yL)을 사용하여 패딩될 수 있다. 한편으로, 대응하는(corresponding) 행 yM2 및 yM1에서의 샘플들 (x, yM2) 및 (x, yM1)은 행 yL에서의 샘플들(x, yL)을 사용하여 패딩될 수 있다.

[0380] v. 일 예에서, 만약 가용하지 않은 샘플이 행 M에 위치하면(예를 들어, $M < C$), $d(C, M) = d(N, C) -$ 오프셋(이때, 오프셋은 정수 값, 예를 들어 1과 같다) 또는 $d(C, M) < d(N, C)$ 일 때, 행 N에 위치하는 샘플들은 패딩되어야 하는 해당 샘플들로 판단된다.

[0381] 1) 일 예에서, 만약 가용하지 않은 샘플이 행 M (예를 들어, $M > C$)에 위치한다면, $d(M, C) = d(C, N) -$ 오프셋 (이때, 오프셋은 정수 값, 예를 들어 1과 같다) 또는 $d(C, M) < d(N, C)$일 때, 행 N에 위치한 샘플들이 패딩되어야 하는 해당 샘플들로 취급된다.

[0382] 2) 일 예에서, 도 31에 정의된 미러링된 패딩 방법이 활용될 수 있다. 가용하지 않은 샘플들이 ALF 가상 경계 위(아래)의 행 M(N)에 위치하는 경우, 그것들은 ALF 가상 경계 위(아래)의 최근접 샘플로부터 패딩된다. AFL 가상 경계 아래(위)의 행 N(M)에 위치한 해당 샘플들은 행 N(M) 위(아래)의 최근접 샘플로부터 패딩될 수 있음이 제안된다.

[0383] a) 일 예에서, $K = 2$라고 가정하고, yL = 0 및 yP1 = 1을 각각 표 4-2에 제시되는 2개의 샘플 행들의 y 좌표로 나타낸다. ALF 가상 경계는 CtbSizeY - 4와 같다.

[0384] i. 일 예에서, ALF 가상 경계가 대표 루마 샘플 아래에 있는 경우, 가용하지 않은 샘플들이 ALF 가상 경계 위의 최근접 행을 사용하여 패딩될 수 있다. 한편으로, 해당 샘플들은 그것들이 위치하는 행 아래의 최근접 행을 사용하여 패딩될 수 있다.

[0385] 1. 일 예에서, yL이 CtbSizeY - 5와 같고 행 yP1이 가용하지 않는 경우, 행 yP1에서의 샘플들 (x, yP1)은 행 yL에서의 샘플들 (x, yL)을 사용하여 패딩될 수 있다.

[0386] b) 일 예에서, $K = 4$라고 가정하고, yM1 = -1, yL = 0, yP1 = 1, yP2 = 2을 각

각 표 4-4에 제시되는 4개의 샘플 행들의 y 좌표로 나타낸다. ALF 가상 경계는 CtbSizeY - 4와 같다.

[0387] i. 일 예에서, ALF 가상 경계가 대표 루마 샘플 위에 있는 경우, 가용하지 않은 샘플들이 ALF 가상 경계 아래의 최근접 행을 사용하여 패딩될 수 있다. 한편으로, 해당 샘플들은 그것들이 위치하는 행 위의 최근접 행을 사용하여 패딩될 수 있다.

[0388] 1. 일 예에서, yL이 CtbSizeY - 4와 같고 yM1 위의 행 및 yM1이 가용하지 않는 경우, 행 yM1에서의 샘플들 (x, yM1)은 행 yL에서의 샘플들 (x, yL)을 사용하여 패딩될 수 있다. 한편으로, 행 yP2에서의 샘플들 (x, yP2)은 행 yP1에서의 샘플들 (x, yP1)을 사용하여 패딩될 수 있다.

[0389] ii. 일 예에서, ALF 가상 경계가 대표 루마 샘플 아래에 있는 경우, 가용하지 않은 샘플들이 ALF 가상 경계 위의 최근접 행을 사용하여 패딩될 수 있다. 한편으로, 해당 샘플들은 그것들이 위치하는 행 아래의 최근접 행을 사용하여 패딩될 수 있다.

[0390] 1. 일 예에서, yL이 CtbSizeY - 6와 같고 행 yP2이 가용하지 않는 경우, 행 yP2에서의 샘플들 (x, yP2)은 행 yP1에서의 샘플들 (x, yP1)을 사용하여 패딩될 수 있다.

[0391] 2. 일 예에서, yL이 CtbSizeY - 5와 같고 행 yP2 및 행 yP1이 가용하지 않는 경우, 행 yP2 및 yP1에서의 샘플들 (x, yP2) 및 (x, yP1)은 행 yL에서의 샘플들 (x, yL)을 사용하여 패딩될 수 있다. 한편으로, 대응하는(corresponding) 행 yM1에서의 샘플들(x, yM1)은 행 yL에서의 샘플들 (x, yL)을 사용하여 패딩될 수 있다.

[0392] c) 일 예에서, K = 6이라 가정하고, yM2 = -2, yM1 = -1, yL = 0, yP1 = 1, yP2 = 2, yP3 =3을 각각 표 4-7에 제시되는 6개의 샘플 행들의 y 좌표로 나타낸다. ALF 가상 경계는 CtbSizeY - 4와 같다.

[0393] i. 일 예에서, ALF 가상 경계가 대표 루마 샘플 위에 있는 경우, 가용하지 않은 샘플들이 ALF 가상 경계 아래의 최근접 행을 사용하여 패딩될 수 있다. 한편으로, 해당 샘플들은 그것들이 위치하는 행 위의 최근접 행을 사용하여 패딩될 수 있다.

[0394] 1. 일 예에서, yL이 CtbSizeY - 3와 같고 yM2 위의 행 및 yM2가 가용하지 않는 경우, 행 yM2에서의 샘플들 (x, yM2)는 행 yM1에서의 샘플들 (x, yM1)을 사용하여 패딩될 수 있다. 한편으로, 대응하는(corresponding)행 yP3에서의 샘플들 (x, yP3)는 행 yP2에서의 샘플들 (x, yP2)을 사용하여 패딩될 수 있다.

[0395] 2. 일 예에서, yL이 CtbSizeY - 4와 같고 yM2 위의 행, yM2 및 yM1이 가용하지 않는 경우, 행 yM2 및 yM1에서의 샘플들 (x, yM2) 및 (x, yM1)은 행 yL에서의 샘플들 (x, yL)을 사용하여 패딩될 수 있다. 한편, 대응하는 행 yP3 및 yP2에서의 샘플들 (x, yP3) 및 (x, yP2)는 행 yP1에서의 샘플들 (x, yP1)을 사용하여 패딩될 수 있다.

[0396] ii. 일 예에서, ALF 가상 경계가 대표 루마 샘플 아래에 있는 경우, 가용하지 않은 샘플들이 ALF 가상 경계 위의 최근접 행을 사용하여 패딩될 수 있다. 한편, 해당 샘플들은 그것들이 위치하는 행 아래의 최근접 행을 사용하여 패딩될 수 있다.

[0397] 1. 일 예에서, yL이 CtbSizeY - 7과 같고 행 yP3가 가용하지 않는 경우, 행 yP3에서의 샘플들 (x, yP3)는 행 yP2에서의 샘플들 (x, yP2)를 사용하여 패딩될 수 있다. 한편으로, 대응하는(corresponding)행 yM2에서의 샘플들 (x, yM2)는 행 yM1에서의 샘플들 (x, yM1)을 사용하여 패딩될 수 있다.

[0398] 2. 일 예에서, yL이 CtbSizeY - 6과 같고 행 yP3 및 yP2가 가용하지 않는 경우, 행 yP3 및 yP2에서의 샘플들 (x, yP3) 및 (x, yP2)는 행 yP1에서의 샘플들 (x, yP1)을 사용하여 패딩될 수 있다. 한편, 대응하는 행 yM2 및 yM1에서의 샘플들 (x, yM2) 및 (x, yM1)은 행 yL에서의 샘플들 (x, yL)을 사용하여 패딩될 수 있다.

[0399] 3. 일 예에서, yL이 CtbSizeY - 5와 같고 행 yP3, yP2 및 yP1이 가용하지 않는 경우, 행 yP3, yP2 및 yP1에서의 샘플들 (x, yP3), (x, yP2) 및 (x, yP1)은 행 yL에서의 샘플들 (x, yL)을 사용하여 패딩될 수 있다. 한편, 대응하는 행 yM2 및 yM1에서의 샘플들 (x, yM2) 및 (x, yM1)은 행 yL에서의 샘플들 (x, yL)을 사용하여 패딩될 수 있다.

[0400] c. 도 29는 (X_c, Y_c)에 위치한 현재 크로마 샘플을 필터링하는 경우 가용하지 않은 샘플(ALF 가상 경계 위, C0에 의해 표시) 및 그것에 대응하는 샘플(C7에 의해 표시)의 위치의 예를 도시한다.

[0401] d. 일 예에서, CC-ALF/크로마 ALF/루마 ALF/기타 종류의 필터링 방법들을 위해 ALF 가상 경계에서 미러링된 패딩을 활성화할지 비활성화할지 여부는 시퀀스 헤더/픽처 헤더/SPS/VPS/DPS/PPS/APS/슬라이스 헤더/타일 그룹 헤더와 같은 시퀀스 레벨/픽처 레벨/슬라이스 레벨/타일 그룹 레벨에서 시그널링될 수 있다.

[0402] e. 일 예에서, ALF 가상 경계에서 반복 패딩 및/또는 미러링된 패딩을 활성화할지 또는 비활성화할지 여부는 코딩된 정보에 의존적일 수 있다.

[0403] i. 일 예에서, 상기 코딩된 정보는 CTU/CTB 크기와 같은 블록 크기를 가리킬 수 있다.

[0404] 1) 일 예에서, CTU/CTB 크기가 T = 32/64/128와 같은 T 이상인 경우, 미러링된 패딩은 ALF 가상 경계에서 활용될 수 있다.

[0405] 2) 일 예에서, CTU/CTB 크기가 T = 4/8/16과 같은 T 이하인 경우, 반복 패딩은 ALF 가상 경계에서 활용될 수 있다.

[0406] 2. 상기 항에서, 수직 패딩은 수평 패딩으로 대체될 수 있다.

[0407] a. 또는, 더 나아가, 어느 패딩 방향(수직 또는 수평)이 사용될 것인지는 경계가 수평 경계인지 수직 경계인지 여부에 의존할 수 있다.

[0408] b. 또는, 더 나아가, 수직 거리는 수평 거리로 대체될 수 있다.

[0409] 3. 1항의 미러링된 패딩 방법은 픽처/서브픽처/슬라이스/타일/경계 및/또는 360도 경계에 대해 활용될 수 있다.

[0410] 일반적인 해결수단들

[0411] 4. 상기 개시된 방법들을 적용할지 여부 및/또는 어떻게 적용할지는 시퀀스 헤더/픽처 헤더/SPS/VPS/DPS/PPS/APS/슬라이스 헤더/타일 그룹 헤더와 같은 시퀀스 레벨/픽처 레벨/슬라이스 레벨/타일 그룹 레벨에서 시그널링될 수 있다.

[0412] 5. 상기 개시된 방법들을 적용할지 여부 및/또는 어떻게 적용할지는 컬러 포맷, 싱글/듀얼 트리 분할, 샘플의 위치(가령, CU/CTU에 상대적인 위치)와 같은 코딩된 정보에 의존적일 수 있다

[0413] 표 4-1. 수직 루마 샘플 위치 yL 및 applyAlfLineBufBoundary에 따른 yP1의 사양

[0414]

Condition	yP1
(yL = = CtbSizeY - 4) && (applyAlfLineBufBoundary = = 1)	0
(yL = = CtbSizeY - 5) && (applyAlfLineBufBoundary = = 1)	0
Otherwise	1

[0415] 표 4-2. 수직 루마 샘플 위치 yL 및 applyAlfLineBufBoundary에 따른 yP1의 사양

[0416]

Condition	yP1
(yL = = CtbSizeY - 4) && (applyAlfLineBufBoundary = = 1)	1
(yL = = CtbSizeY - 5) && (applyAlfLineBufBoundary = = 1)	0
Otherwise	1

[0417] 표 4-3. 수직 루마 샘플 위치 yL 및 applyAlfLineBufBoundary에 따른 yM1, yP1 및 yP2의 사양

Condition	yM1	yP1	yP2
(yL = = CtbSizeY - 3) && (applyAlfLineBufBoundary = = 1)	-1	1	1
(yL = = CtbSizeY - 4) && (applyAlfLineBufBoundary = = 1)	0	0	0
(yL = = CtbSizeY - 5) && (applyAlfLineBufBoundary = = 1)	0	0	0
(yL = = CtbSizeY - 6) && (applyAlfLineBufBoundary = = 1)	-1	1	1
Otherwise	-1	1	2

[0418]

[0419] 표 4-4. 수직 루마 샘플 위치 yL 및 applyAlfLineBufBoundary에 따른 yM1, yP1 및 yP2의 사양

Condition	yM1	yP1	yP2
(yL = = CtbSizeY - 4) && (applyAlfLineBufBoundary = = 1)	0	1	1
(yL = = CtbSizeY - 5) && (applyAlfLineBufBoundary = = 1)	0	0	0
(yL = = CtbSizeY - 6) && (applyAlfLineBufBoundary = = 1)	0	1	1
Otherwise	-1	1	2

[0420]

[0421] 표 4-5. 수직 루마 샘플 위치 yL 및 applyAlfLineBufBoundary에 따른 yM2, yM1, yP1 및 yP2의 사양

Condition	yM2	yM1	yP1	yP2
(yL = = CtbSizeY - 3) && (applyAlfLineBufBoundary = = 1)	-1	-1	1	1
(yL = = CtbSizeY - 4) && (applyAlfLineBufBoundary = = 1)	0	0	0	0
(yL = = CtbSizeY - 5) && (applyAlfLineBufBoundary = = 1)	0	0	0	0
(yL = = CtbSizeY - 6) && (applyAlfLineBufBoundary = = 1)	-1	-1	1	1
Otherwise	-2	-1	1	2

[0422]

[0423] 표 4-6. 수직 루마 샘플 위치 yL 및 applyAlfLineBufBoundary에 따른 yM2, yM1, yP1, yP2 및 yP3의 사양

Condition	yM2	yM1	yP1	yP2	yP3
(yL = = CtbSizeY - 2) && (applyAlfLineBufBoundary = = 1)	-2	-1	1	2	2
(yL = = CtbSizeY - 3) && (applyAlfLineBufBoundary = = 1)	-1	-1	1	1	1
(yL = = CtbSizeY - 4) && (applyAlfLineBufBoundary = = 1)	0	0	0	0	0
(yL = = CtbSizeY - 5) && (applyAlfLineBufBoundary = = 1)	0	0	0	0	0
(yL = = CtbSizeY - 6) && (applyAlfLineBufBoundary = = 1)	-1	-1	1	1	1
(yL = = CtbSizeY - 7) && (applyAlfLineBufBoundary = = 1)	-2	-1	1	2	2
Otherwise	-2	-1	1	2	3

[0424]

[0425] 표 4-7. 수직 루마 샘플 위치 yL 및 applyAlfLineBufBoundary에 따른 yM2, yM1, yP1, yP2 및 yP3의 사양

Condition	yM2	yM1	yP1	yP2	yP3
(yL = = CtbSizeY - 3) && (applyAlfLineBufBoundary = = 1)	-1	-1	1	2	2
(yL = = CtbSizeY - 4) && (applyAlfLineBufBoundary = = 1)	0	0	1	1	1
(yL = = CtbSizeY - 5) && (applyAlfLineBufBoundary = = 1)	0	0	0	0	0
(yL = = CtbSizeY - 6) && (applyAlfLineBufBoundary = = 1)	0	0	1	1	1
(yL = = CtbSizeY - 7) && (applyAlfLineBufBoundary = = 1)	-1	-1	1	2	2
Otherwise	-2	-1	1	2	3

[0426]

[0427] **5. 실시예들**

[0428] 삭제 및 추가들을 보여줌으로써 변경사항들이 강조된다.

[0429] **5.1. 실시예#1**

[0430] JVET-P0080에 명시된 작업 초안은 아래와 같이 변경될 수 있다.

[0431] **x.x.x.x크로마 샘플들의 블록에 대한 크로스 컴포넌트 필터링 프로세스**

[0432] 이 프로세스의 입력들은 아래와 같다.

[0433] - 루마 적응적 루프 필터링 프로세스 이전의 재구성된 루마 픽처 샘플 어레이 $recPicture_L$,

[0434] - 필터링된 재구성된 크로마 픽처 샘플 어레이 $alfPicture_C$,

[0435] - 현재 픽처의 좌측 상단 샘플에 상대적인 크로마 샘플들의 현재 블록의 좌측 상단 샘플을 명시하는 크로마 위치 (xC, yC),

[0436] - 크로마 샘플들의 블록의 폭 ccAlfWidth,

[0437] - 크로마 샘플들의 블록의 높이 ccAlfHeight,

[0438] - 크로스 컴포넌트 필터 계수들 CcAlfCoeff[j] (j = 0..13)

[0439] 이 프로세스의 출력은 수정된 필터링된 재구성된 크로마 픽처 샘플 어레이 ccAlfPicture이다.

[0440] 코딩 트리 블록 루마 위치(xCtb, yCtb)는 아래와 같이 도출된다.

[0441]
$$xCtb = (((xC * SubWidthC) >> CtbLog2SizeY) << CtbLog2SizeY \quad (8\text{-}1229)$$
$$yCtb = (((yC * SubHeightC) >> CtbLog2SizeY) << CtbLog2SizeY \quad (8\text{-}1229)$$

[0442] 필터링된 재구성된 크로마 샘플들ccAlfPicture[xC + x][yC + y]의 도출을 위해, 샘플들의 현재 크로마 블록 내의 각 재구성된 크로마 샘플$alfPicture_C$[xC + x][yC + y] (x = 0..ccAlfWidth - 1, y = 0..ccAlfHeight - 1)은 아래와 같이 필터링된다.

[0443] - 크로마 위치 (xC + x, yC + y)의 현재 크로마 샘플에 상응하는 루마 위치(xL, yL)는 ((xC + x) * SubWidthC, (yC + y) * SubHeightC)와 같게 설정된다.

[0444] - 어레이 $recPicture_L$ 내부의 루마 위치들(h_{xL+i}, v_{yL+j}) (i = -2..2, j = -2..3)은 아래와 같이 도출된다.

[0445] - 만약

[0446] pps_loop_filter_across_virtual_boundaries_disabled_flag가 1과 같고, 임의의 n = 0..pps_num_ver_virtual_boundaries - 1에 대해, PpsVirtualBoundariesPosX[n] % CtbSizeY가 0과 같지 않고, xL - PpsVirtualBoundariesPosX[n]가 0 이상 3 미만인 경우, 다음이 적용된다.

[0447]
$$h_{xL+i} = Clip3(PpsVirtualBoundariesPosX[n], pic_width_in_luma_samples - 1, xL + i) \quad (8\text{-}1229)$$

[0448] - 그렇지 않고, 만약

[0449] pps_loop_filter_across_virtual_boundaries_disabled_flag가 1과 같고, 임의의 n = 0..pps_num_ver_virtual_boundaries - 1에 대해, PpsVirtualBoundariesPosX[n] % CtbSizeY가 0과 같지 않고, PpsVirtualBoundariesPosX[n]- xL가 0 보다 크고 4 미만이면, 다음이 적용된다.

[0450]
$$h_{x+i} = Clip3(0, PpsVirtualBoundariesPosX[n] - 1, xL + i) \quad (8\text{-}1230)$$

[0451] - 그렇지 않은 경우, 다음이 적용된다.

[0452] $h_{x+i} = Clip3(0, pic_width_in_luma_samples - 1, xL + i)$ (8-1231)

[0453] - 만약

[0454] pps_loop_filter_across_virtual_boundaries_disabled_flag가 1과 같고, 임의의 n = 0..pps_num_hor_virtual_boundaries - 1에 대해, PpsVirtualBoundariesPosY[n] % CtbSizeY가 0과 같지 않고 yL - PpsVirtualBoundariesPosY[n]가 0 이상 3 미만인 경우, 다음이 적용된다.

[0455] $v_{y+j} = Clip3(PpsVirtualBoundariesPosY[n], pic_height_in_luma_samples - 1, yL + j)$ (8-1232)

[0456] - 그렇지 않고, 만약

[0457] pps_loop_filter_across_virtual_boundaries_disabled_flag가 1과 같고, 임의의 n = 0..pps_num_hor_virtual_boundaries - 1에 대해, PpsVirtualBoundariesPosY[n] % CtbSizeY가 0과 같지 않고 PpsVirtualBoundariesPosY[n] - yL이 0 보다 크고 4 보다 작으면, 다음이 적용된다.

[0458] $v_{y+j} = Clip3(0, PpsVirtualBoundariesPosY[n] - 1, yL + j)$ (8-1233)

[0459] - 그렇지 않은 경우, 다음이 적용된다.

[0460] $v_{y+j} = Clip3(0, pic_height_in_luma_samples - 1, yL + j)$ (8-1234)

[0461] - 변수 clipLeftPos, clipRightPos, clipTopPos 및 clipBottomPos는 (xCtb, yCtb) 및 (xL - xCtb, yL - yCtb)를 입력들로 하여 8.8.5.5 절에 명시된 바와 같이 ALF 경계 위치 도출 프로세스를 호출하는 것에 의해 도출된다

[0462] - 수직 샘플 위치 오프셋 yM2, yM1, yP1, yP2 및 yP3는 표 4-1에 명시된다. 수직 루마 샘플 위치 yL 및 applyAlfLineBufBoundary에 따른 yP1의 사양

[0463]

Condition	yP1
(yL = = CtbSizeY - 4) && (applyAlfLineBufBoundary = = 1)	0
(yL = = CtbSizeY - 5) && (applyAlfLineBufBoundary = = 1)	0
Otherwise	1

[0464] 표 4-2. 수직 루마 샘플 위치 yL 및 applyAlfLineBufBoundary에 따른 yP1의 사양

[0465]

Condition	yP1
(yL = = CtbSizeY - 4) && (applyAlfLineBufBoundary = = 1)	1
(yL = = CtbSizeY - 5) && (applyAlfLineBufBoundary = = 1)	0
Otherwise	1

[0466] - 수직 루마 샘플 위치 yL, clipLeftPos 및 clipRightPos에 따른

[0467] - 수평 샘플 위치 오프셋들 xM1, xM2, xP1 및 xP2는 수평 루마 샘플 위치 xL, clipLeftPos 및 clipRightPos에 따라 표 y-yyyy에 명시된다.

[0468] - 변수 curr은 아래와 같이 도출된다.

[0469] $curr = alfPicture_C[xC + x, yC + y]$ (8-1286)

[0470] - 크로스 컴포넌트 필터 계수들 f[j]의 어레이는 j = 0..13로 아래와 같이 도출된다.

[0471] $$f[j] = CcAlfCoeff[j] \quad (8\text{-}1287)$$

[0472] - 변수 sum은 아래와 같이 도출된다.

$$\begin{aligned} sum = {} & f[0] * recPicture_L[h_x, V_{y+yM2}] + \\ & f[1] * recPicture_L[h_{x+xM1}, V_{y+yM1}] + \\ & f[2] * recPicture_L[h_x, V_{y+yM1}] + \\ & f[3] * recPicture_L[h_{x+xP1}, V_{y+yM1}] + \\ & f[4] * recPicture_L[h_{x+xM2}, V_y] + \\ & f[5] * recPicture_L[h_{x+xM1}, V_y] + \\ & f[6] * recPicture_L[h_x, V_y] + \\ & f[7] * recPicture_L[h_{x+xP1}, V_y] + \\ & f[4] * recPicture_L[h_{x+xP2}, V_y] + \\ & f[4] * recPicture_L[h_{x+xM2}, V_{y+yP1}] + \\ & f[8] * recPicture_L[h_{x+xM1}, V_{y+yP1}] + \\ & f[9] * recPicture_L[h_x, V_{y+yP1}] + \\ & f[10] * recPicture_L[h_{x+xP1}, V_{y+yP1}] + \\ & f[4] * recPicture_L[h_{x+xP2}, V_{y+yP1}] + \\ & f[11] * recPicture_L[h_{x+xM1}, V_{y+yP2}] + \\ & f[12] * recPicture_L[h_x, V_{y+yP2}] + \\ & f[13] * recPicture_L[h_{x+xP1}, V_{y+yP2}] + \\ & f[0] * recPicture_L[h_x, V_{y+yP3}] + \end{aligned} \quad (8\text{-}1289)$$

[0473] $$sum = curr + (sum + 64) >> 7) \quad (8\text{-}1290)$$

[0474] - 수정된 필터링된 재구성된 크로마 픽처 샘플 어레이 ccAlfPicture[xC + x][yC + y]는 아래와 같이 도출된다.

[0475] $$ccAlfPicture[xC + x][yC + y] = Clip3(0, (1 << BitDepth_C) - 1, sum) \quad (8\text{-}1291)$$

[0476] *표 x-xx - 수직 루마 샘플 위치 yL, clipTopPos 및 clipBottomPos에 따른 yM1, yM2, yP1, yP2 및 yP3의 사양*

[0477] 아래 표를 삭제한다.

[0478]

Condition	yM2	yM1	yP1	yP2	yP3
yL = = clipTopPos + 1	−1	−1	1	2	3
yL = = clipTopPos	0	0	1	2	3
yL = = clipBottomPos − 1	−2	−1	0	0	0
yL = = clipBottomPos − 2	−2	−1	1	1	1
yL = = clipBottomPos − 3	−2	−1	1	2	2
Otherwise	−2	−1	1	2	3

[0479] 상기 표를 대신하여 아래 표를 추가한다.

Condition	yM2	yM1	yP1	yP2	yP3
yL = = clipTopPos + 1	-1	-1	1	2	2
yL = = clipTopPos	0	0	1	1	1
yL = = clipBottomPos – 1	0	0	0	0	0
yL = = clipBottomPos – 2	0	0	1	1	1
yL = = clipBottomPos – 3	-1	-1	1	2	2
Otherwise	–2	-1	1	2	3

[0480]

[0481] 표 y-yy - 수평 루마 샘플 위치 xL, clipLeftPos 및 clipRightPos에 따른 xM1, xM2, xP1 및 xP2의 사양

Condition	xM2	xM1	xP1	xP2
xL = = clipLeftPos + 1	–1	–1	1	2
xL = = clipLeftPos	0	0	1	2
xL = = clipRightPos – 1	–2	–1	0	0
xL = = clipRightPos – 2	–2	–1	1	1
Otherwise	–2	–1	1	2

[0482]

[0483] **5.2. 실시예 #2**

[0484] JVET-P0080에 명시된 작업 초안은 아래와 같이 변경될 수 있다.

[0485] **x.x.x.x 크로마 샘플들의 블록에 대한 크로스 컴포넌트 필터링 프로세스**

[0486] 이 프로세스의 입력들은 아래와 같다.

[0487] - 루마 적응적 루프 필터링 프로세스 이전의 재구성된 루마 픽처 샘플 어레이 $recPicture_L$,

[0488] - 필터링된 재구성된 크로마 픽처 샘플 어레이 $alfPicture_C$,

[0489] - 현재 픽처의 좌측 상단 샘플에 상대적인 크로마 샘플들의 현재 블록의 좌측 상단 샘플을 명시하는 크로마 위치 (xC, yC),

[0490] - 크로마 샘플들의 블록의 폭 ccAlfWidth,

[0491] - 크로마 샘플들의 블록의 높이 ccAlfHeight,

[0492] - 크로스 컴포넌트 필터 계수들 CcAlfCoeff[j] (j = 0..13)

[0493] 이 프로세스의 출력은 수정된 필터링된 재구성된 크로마 픽처 샘플 어레이 ccAlfPicture이다.

[0494] 코딩 트리 블록 루마 위치(xCtb, yCtb)는 아래와 같이 도출된다.

$$xCtb = (((xC * SubWidthC) >> CtbLog2SizeY) << CtbLog2SizeY \quad (8\text{-}1229)$$
$$yCtb = (((yC * SubHeightC) >> CtbLog2SizeY) << CtbLog2SizeY \quad (8\text{-}1229)$$

[0495]

[0496] 필터링된 재구성된 크로마 샘플들 ccAlfPicture[xC + x][yC + y]의 도출을 위해, 샘플들의 현재 크로마 블록 내의 각 재구성된 크로마 샘플$alfPicture_C$[xC + x][yC + y] (x = 0..ccAlfWidth - 1, y = 0..ccAlfHeight - 1)은 아래와 같이 필터링된다.

[0497] - 크로마 위치 (xC + x, yC + y)의 현재 크로마 샘플에 상응하는 루마 위치(xL, yL)는 ((xC + x) * SubWidthC, (yC + y) * SubHeightC)와 같게 설정된다.

[0498] - 어레이 $recPicture_L$ 내부의 루마 위치들($h_{xL + i}$, $v_{yL + j}$) (i = -2..2, j = -2..3)은 아래와 같이

도출된다.

[0499] - 만약

[0500] pps_loop_filter_across_virtual_boundaries_disabled_flag가 1과 같고, 임의의 n = 0..pps_num_ver_virtual_boundaries - 1에 대해, PpsVirtualBoundariesPosX[n] % CtbSizeY가 0과 같지 않고, xL - PpsVirtualBoundariesPosX[n]가 0 이상 3 미만인 경우, 다음이 적용된다.

[0501] h_{xL+i} = Clip3(PpsVirtualBoundariesPosX[n], pic_width_in_luma_samples - 1, xL + i) (8-1229)

[0502] - 그렇지 않고, 만약

[0503] pps_loop_filter_across_virtual_boundaries_disabled_flag가 1과 같고, 임의의 n = 0..pps_num_ver_virtual_boundaries - 1에 대해, PpsVirtualBoundariesPosX[n] % CtbSizeY가 0과 같지 않고, PpsVirtualBoundariesPosX[n]- xL가 0 보다 크고 4 미만이면, 다음이 적용된다.

[0504] h_{x+i} = Clip3(0, PpsVirtualBoundariesPosX[n] - 1, xL + i) (8-1230)

[0505] - 그렇지 않은 경우, 다음이 적용된다.

[0506] h_{x+i} = Clip3(0, pic_width_in_luma_samples - 1, xL + i) (8-1231)

[0507] - 만약

[0508] pps_loop_filter_across_virtual_boundaries_disabled_flag가 1과 같고, 임의의 n = 0..pps_num_hor_virtual_boundaries - 1에 대해, PpsVirtualBoundariesPosY[n] % CtbSizeY가 0과 같지 않고 yL - PpsVirtualBoundariesPosY[n]가 0 이상 3 미만인 경우, 다음이 적용된다.

[0509] v_{y+j} = Clip3(PpsVirtualBoundariesPosY[n], pic_height_in_luma_samples - 1, yL + j) (8-1232)

[0510] - 그렇지 않고, 만약

[0511] pps_loop_filter_across_virtual_boundaries_disabled_flag가 1과 같고, 임의의 n = 0..pps_num_hor_virtual_boundaries - 1에 대해, PpsVirtualBoundariesPosY[n] % CtbSizeY가 0과 같지 않고 PpsVirtualBoundariesPosY[n] - yL이 0 보다 크고 4 보다 작으면, 다음이 적용된다.

[0512] v_{y+j} = Clip3(0, PpsVirtualBoundariesPosY[n] - 1, yL + j) (8-1233)

[0513] - 그렇지 않은 경우, 다음이 적용된다.

[0514] v_{y+j} = Clip3(0, pic_height_in_luma_samples - 1, yL + j) (8-1234)

[0515] - 변수 clipLeftPos, clipRightPos, clipTopPos 및 clipBottomPos는 (xCtb, yCtb) 및 (xL - xCtb, yL - yCtb)를 입력들로 하여 8.8.5.5 절에 명시된 바와 같이 ALF 경계 위치 도출 프로세스를 호출하는 것에 의해 도출된다.

[0516] 수직 샘플 위치 오프셋 yM2, yM1, yP1, yP2 및 yP3는 표 4-1에 명시된다. 수직 루마 샘플 위치 yL 및 applyAlfLineBufBoundary에 따른 yP1의 사양

[0517]

Condition	yP1
(yL = = CtbSizeY - 4) && (applyAlfLineBufBoundary = = 1)	0
(yL = = CtbSizeY - 5) && (applyAlfLineBufBoundary = = 1)	0
Otherwise	1

[0518] 표 4-2. 수직 루마 샘플 위치 yL 및 applyAlfLineBufBoundary에 따른 yP1의 사양

Condition	yP1
(yL = = CtbSizeY - 4) && (applyAlfLineBufBoundary = = 1)	1
(yL = = CtbSizeY - 5) && (applyAlfLineBufBoundary = = 1)	0
Otherwise	1

[0519]

[0520] - 수직 루마 샘플 위치 yL, clipLeftPos 및 clipRightPos에 따른

[0521] - 수평 샘플 위치 오프셋들 xM1, xM2, xP1 및 xP2는 수평 루마 샘플 위치 xL, clipLeftPos 및 clipRightPos에 따라 표 y-yyyy에 명시된다.

[0522] - 변수 curr은 아래와 같이 도출된다.

[0523] $$curr = alfPicture_C[xC + x, yC + y] \quad (8\text{-}1286)$$

[0524] - 크로스 컴포넌트 필터 계수들 f[j]의 어레이는 j = 0..13로 아래와 같이 도출된다.

[0525] $$f[j] = CcAlfCoeff[j] \quad (8\text{-}1287)$$

[0526] - 변수 sum은 아래와 같이 도출된다.

$$\begin{aligned} sum = {} & f[0] * recPicture_L[h_x, v_{y+yM2}] + \\ & f[1] * recPicture_L[h_{x+xM1}, v_{y+yM1}] + \\ & f[2] * recPicture_L[h_x, v_{y+yM1}] + \\ & f[3] * recPicture_L[h_{x+xP1}, v_{y+yM1}] + \\ & f[4] * recPicture_L[h_{x+xM2}, v_y] + \\ & f[5] * recPicture_L[h_{x+xM1}, v_y] + \\ & f[6] * recPicture_L[h_x, v_y] + \\ & f[7] * recPicture_L[h_{x+xP1}, v_y] + \\ & f[4] * recPicture_L[h_{x+xP2}, v_y] + \\ & f[4] * recPicture_L[h_{x+xM2}, v_{y+yP1}] + \\ & f[8] * recPicture_L[h_{x+xM1}, v_{y+yP1}] + \\ & f[9] * recPicture_L[h_x, v_{y+yP1}] + \\ & f[10] * recPicture_L[h_{x+xP1}, v_{y+yP1}] + \\ & f[4] * recPicture_L[h_{x+xP2}, v_{y+yP1}] + \\ & f[11] * recPicture_L[h_{x+xM1}, v_{y+yP2}] + \\ & f[12] * recPicture_L[h_x, v_{y+yP2}] + \\ & f[13] * recPicture_L[h_{x+xP1}, v_{y+yP2}] + \\ & f[0] * recPicture_L[h_x, v_{y+yP3}] + \end{aligned} \quad (8\text{-}1289)$$

$$sum = curr + (sum + 64) >> 7) \quad (8\text{-}1290)$$

[0527]

[0528] - 수정된 필터링된 재구성된 크로마 픽처 샘플 어레이 ccAlfPicture[xC + x][yC + y]는 아래와 같이 도출된다.

[0529] $$ccAlfPicture[xC + x][yC + y] = Clip3(0, (1 << BitDepth_C) - 1, sum) \quad (8\text{-}1291)$$

[0530] *표 x-xx - 수직 루마 샘플 위치 yL, clipTopPos 및 clipBottomPos에 따른 yM1, yM2, yP1, yP2 및 yP3의 사양*

[0531] 아래 표를 삭제한다.

Condition	yM2	yM1	yP1	yP2	yP3
yL = = clipTopPos + 1	−1	−1	1	2	3
yL = = clipTopPos	0	0	1	2	3
yL = = clipBottomPos − 1	−2	−1	0	0	0
yL = = clipBottomPos − 2	−2	−1	1	1	1
yL = = clipBottomPos − 3	−2	−1	1	2	2
Otherwise	−2	−1	1	2	3

[0532]

[0533] 아래 표를 추가한다

Condition	yM2	yM1	yP1	yP2	yP3
yL = = clipTopPos + 2	-2	-1	1	2	2
yL = = clipTopPos + 1	-1	-1	1	1	1
yL = = clipTopPos	0	0	0	0	0
yL = = clipBottomPos − 1	0	0	0	0	0
yL = = clipBottomPos − 2	0	-1	1	1	1
yL = = clipBottomPos − 3	-2	-1	1	2	2
Otherwise	−2	-1	1	2	3

[0534]

[0535] *표 y-yy - 수평 루마 샘플 위치 xL, clipLeftPos 및 clipRightPos에 따른 xM1, xM2, xP1 및 xP2의 사양*

Condition	xM2	xM1	xP1	xP2
xL = = clipLeftPos + 1	−1	−1	1	2
xL = = clipLeftPos	0	0	1	2
xL = = clipRightPos − 1	−2	−1	0	0
xL = = clipRightPos – 2	−2	−1	1	1
Otherwise	−2	−1	1	2

[0536]

[0537] **5.3. 실시예 #3**

[0538] JVET-P1008에 명시된 작업 초안은 아래와 같이 변경될 수 있다.

[0539] **x.x.x.x 크로마 샘플들의 블록에 대한 크로스 컴포넌트 필터링 프로세스**

[0540] 이 프로세스의 입력들은 아래와 같다.

[0541] - 루마 적응적 루프 필터링 프로세스 이전의 재구성된 루마 픽처 샘플 어레이 $recPicture_L$,

[0542] - 필터링된 재구성된 크로마 픽처 샘플 어레이 $alfPicture_C$,

[0543] - 현재 픽처의 좌측 상단 샘플에 상대적인 현재 크로마 코딩 트리 블록의 좌측 상단 샘플을 명시하는 크로마 위치 (xCtbC, yCtbC),

[0544] - 크로마 샘플들의 블록의 폭 ccAlfWidth,

[0545] - 크로마 샘플들의 블록의 높이 ccAlfHeight,

[0546] - 크로스 컴포넌트 필터 계수들 CcAlfCoeff[j] (j = 0..7)

[0547] 이 프로세스의 출력은 수정된 필터링된 재구성된 크로마 픽처 샘플 어레이 ccAlfPicture이다.

[0548] 코딩 트리 블록 루마 위치(xCtb, yCtb)는 아래와 같이 도출된다

[0549] $$xCtb = ((xCtbC * SubWidthC) >> CtbLog2SizeY) << CtbLog2SizeY \quad (8\text{-}1229)$$
$$yCtb = ((yCtbC * SubHeightC) >> CtbLog2SizeY) << CtbLog2SizeY \quad (8\text{-}1229)$$

[0550] 필터링된 재구성된 크로마 샘플들ccAlfPicture[xCtbC + x][yCtbC + y]의 도출을 위해, 샘플들의 현재 크로마 블록 내의 각 재구성된 크로마 샘플alfPicture$_C$[xCtbC + x][yCtbC + y] (x = 0..ccAlfWidth - 1, y = 0..ccAlfHeight - 1)은 아래와 같이 필터링된다.

[0551] - 크로마 위치 (xCtbC + x, yCtbC + y)의 현재 크로마 샘플에 상응하는 루마 위치(xL, yL)는 ((xCtbC + x) * SubWidthC, (yCtbC + y) * SubHeightC)와 같게 설정된다.

[0552] - 어레이 recPicture$_L$ 내부의 루마 위치들(h_{xL+i}, v_{yL+j}) (i = -1..1, j = -1..2)은 아래와 같이 도출된다.

[0553] - 만약 pps_loop_filter_across_virtual_boundaries_disabled_flag가 1과 같고, 임의의 n = 0..pps_num_ver_virtual_boundaries - 1에 대해, PpsVirtualBoundariesPosX[n] % CtbSizeY가 0과 같지 않고, xL - PpsVirtualBoundariesPosX[n]가 0 이상 3 미만인 경우, 다음이 적용된다.

[0554] $$h_{xL+i} = Clip3(PpsVirtualBoundariesPosX[n], pic_width_in_luma_samples - 1, xL + i) \quad (8\text{-}1229)$$

[0555] - 그렇지 않고, 만약

[0556] pps_loop_filter_across_virtual_boundaries_disabled_flag가 1과 같고, 임의의 n = 0..pps_num_ver_virtual_boundaries - 1에 대해, PpsVirtualBoundariesPosX[n] % CtbSizeY가 0과 같지 않고, PpsVirtualBoundariesPosX[n]- xL가 0 보다 크고 4 미만이면, 다음이 적용된다.

[0557] $$h_{x+i} = Clip3(0, PpsVirtualBoundariesPosX[n] - 1, xL + i) \quad (8\text{-}1230)$$

[0558] - 그렇지 않은 경우, 다음이 적용된다.

[0559] $$h_{x+i} = Clip3(0, pic_width_in_luma_samples - 1, xL + i) \quad (8\text{-}1231)$$

[0560] - 만약 pps_loop_filter_across_virtual_boundaries_disabled_flag가 1과 같고, 임의의 n = 0..pps_num_hor_virtual_boundaries - 1에 대해, PpsVirtualBoundariesPosY[n] % CtbSizeY가 0과 같지 않고 yL - PpsVirtualBoundariesPosY[n]가 0 이상 3 미만인 경우, 다음이 적용된다.

[0561] $$v_{y+j} = Clip3(PpsVirtualBoundariesPosY[n], pic_height_in_luma_samples - 1, yL + j) \quad (8\text{-}1232)$$

[0562] - 그렇지 않고, 만약

[0563] pps_loop_filter_across_virtual_boundaries_disabled_flag가 1과 같고, 임의의 n = 0..pps_num_hor_virtual_boundaries - 1에 대해, PpsVirtualBoundariesPosY[n] % CtbSizeY가 0과 같지 않고 PpsVirtualBoundariesPosY[n] - yL이 0 보다 크고 4 보다 작으면, 다음이 적용된다.

[0564] $$v_{y+j} = Clip3(0, PpsVirtualBoundariesPosY[n] - 1, yL + j) \quad (8\text{-}1233)$$

[0565] - 그렇지 않은 경우, 다음이 적용된다.

[0566] $$v_{y+j} = Clip3(0, pic_height_in_luma_samples - 1, yL + j) \quad (8\text{-}1234)$$

[0567] - 변수 clipLeftPos, clipRightPos, clipTopPos 및 clipBottomPos는 (xCtb, yCtb) 및 (xL - xCtb, yL - yCtb)를 입력들로 하여 8.8.5.5 절에 명시된 바와 같이 ALF 경계 위치 도출 프로세스를 호출하는 것에 의해 도출된다.

[0568] - 수직 샘플 위치 오프셋 yM1, yP1 및 yP2는 표 4-1에 명시된다. 수직 루마 샘플 위치 yL 및

applyAlfLineBufBoundary에 따른 yP1의 사양

Condition	yP1
(yL = = CtbSizeY − 4) && (applyAlfLineBufBoundary = = 1)	0
(yL = = CtbSizeY − 5) && (applyAlfLineBufBoundary = = 1)	0
Otherwise	1

[0569]

[0570] 표 4-2. 수직 루마 샘플 위치 yL 및 applyAlfLineBufBoundary에 따른 yP1의 사양

Condition	yP1
(yL = = CtbSizeY − 4) && (applyAlfLineBufBoundary = = 1)	1
(yL = = CtbSizeY − 5) && (applyAlfLineBufBoundary = = 1)	0
Otherwise	1

[0571]

[0572] - 수직 루마 샘플 위치 yL, clipLeftPos 및 clipRightPos에 따른

[0573] - 수평 샘플 위치 오프셋들 xM1 및 xP1는 수평 루마 샘플 위치 xL, clipLeftPos 및 clipRightPos에 따라 표 y-yyyy에 명시된다.

[0574] - 변수 curr는 다음과 같이 도출된다

[0575] $$curr = alfPicture_C[xCtbC + x, yCtbC + y] \quad (8\text{-}1286)$$

[0576] - 크로스 컴포넌트 필터 계수들의 어레이 f[j]는 j = 0..7로 아래와 같이 도출된다.

[0577] $$f[j] = CcAlfCoeff[j] \quad (8\text{-}1287)$$

[0578] - 변수 sum은 아래와 같이 도출된다.

$$\begin{aligned} sum = {} & f[0] * recPicture_L[h_x, v_{y+yM1}] + \\ & f[1] * recPicture_L[h_{x+xM1}, v_y] + \\ & f[2] * recPicture_L[h_x, v_y] + \\ & f[3] * recPicture_L[h_{x+xP1}, v_y] + \\ & f[4] * recPicture_L[h_{x+xM1}, v_{y+yP1}] + \\ & f[5] * recPicture_L[h_x, v_{y+yP1}] + \\ & f[6] * recPicture_L[h_{x+xP1}, v_{y+yP1}] + \\ & f[7] * recPicture_L[h_x, v_{y+yP2}] \end{aligned} \quad (8\text{-}1289)$$

$$sum = Clip3(-(1 << (BitDepth_C - 1)), (1 << (BitDepth_C - 1)) - 1, sum) \quad (8\text{-}1290)$$

$$sum = curr + (sum + 64) >> (7 + (BitDepth_Y - BitDepth_C)) \quad (8\text{-}1290)$$

[0579]

[0580] - 수정된 필터링된 재구성된 크로마 픽처 샘플 어레이 ccAlfPicture[xCtbC + x][yCtbC + y]는 다음과 같이 도출된다.

[0581] $$ccAlfPicture[xCtbC + x][yCtbC + y] = Clip3(0, (1 << BitDepth_C) - 1, sum) \quad (8\text{-}1291)$$

[0582] *표 x-xx - 수직 루마 샘플 위치 yL, clipTopPos 및 clipBottomPos에 따른 yM1, yP1 및 yP2의 사양*

[0583] 아래 표를 삭제한다.

Condition	yM1	yP1	yP2
yL = = clipTopPos + 1	−1	1	1
yL = = clipTopPos	0	0	1
yL = = clipBottomPos − 1	0	0	1
yL = = clipBottomPos − 2	−1	1	1
Otherwise	−1	1	2

[0584]

[0585] 아래 표를 추가한다.

Condition	yM1	yP1	yP2
yL = = clipTopPos	0	1	1
yL = = clipBottomPos − 1	0	0	0
yL = = clipBottomPos − 2	0	1	1
Otherwise	−1	1	2

[0586]

[0587] *표 y-yy - 수평 루마 샘플 위치 xL, clipLeftPos 및 clipRightPos에 따른 xM1 및 xP1의 사양*

Condition	xM1	xP1
xL = = clipLeftPos	0	0
xL = = clipRightPos − 1	0	0
xL = = clipRightPos – 2	–1	1
Otherwise	–1	1

[0588]

[0589] **5.4. 실시예 #4**

[0590] JVET-P1008에 명시된 작업 초안은 아래와 같이 변경될 수 있다.

[0591] **x.x.x.x 크로마 샘플들의 블록에 대한 크로스 컴포넌트 필터링 프로세스**

[0592] 이 프로세스의 입력들은 아래와 같다.

[0593] - 루마 적응적 루프 필터링 프로세스 이전의 재구성된 루마 픽처 샘플 어레이 $recPicture_L$,

[0594] - 필터링된 재구성된 크로마 픽처 샘플 어레이 $alfPicture_C$,

[0595] - 현재 픽처의 좌측 상단 샘플에 상대적인 현재 크로마 코딩 트리 블록의 좌측 상단 샘플을 명시하는 크로마 위치 (xCtbC, yCtbC),

[0596] - 크로마 샘플들의 블록의 폭 ccAlfWidth,

[0597] - 크로마 샘플들의 블록의 높이 ccAlfHeight,

[0598] - 크로스 컴포넌트 필터 계수들 CcAlfCoeff[j] (j = 0..7)

[0599] 이 프로세스의 출력은 수정된 필터링된 재구성된 크로마 픽처 샘플 어레이ccAlfPicture이다.

[0600] 코딩 트리 블록 루마 위치(xCtb, yCtb)는 아래와 같이 도출된다.

[0601]
$$xCtb = ((xCtbC * SubWidthC) >> CtbLog2SizeY) << CtbLog2SizeY \quad (8\text{-}1229)$$
$$yCtb = ((yCtbC * SubHeightC) >> CtbLog2SizeY) << CtbLog2SizeY \quad (8\text{-}1229)$$

[0602] 필터링된 재구성된 크로마 샘플들ccAlfPicture[xCtbC + x][yCtbC + y]의 도출을 위해, 샘플들의 현

재 크로마 블록 내의 각 재구성된 크로마 샘플$alfPicture_C$[xCtbC + x][yCtbC + y] (x = 0..ccAlfWidth - 1, y = 0..ccAlfHeight - 1)은 아래와 같이 필터링된다.

[0603] - 크로마 위치 (xCtbC + x, yCtbC + y)의 현재 크로마 샘플에 상응하는 루마 위치(xL, yL)는 ((xCtbC + x) * SubWidthC, (yCtbC + y) * SubHeightC)와 같게 설정된다.

[0604] - 어레이 $recPicture_L$ 내부의 루마 위치들(h_{xL+i}, v_{yL+j}) (i = -1..1, j = -1..2)은 아래와 같이 도출된다.

[0605] - 만약 pps_loop_filter_across_virtual_boundaries_disabled_flag가 1과 같고, 임의의 n = 0..pps_num_ver_virtual_boundaries - 1에 대해, PpsVirtualBoundariesPosX[n] % CtbSizeY가 0과 같지 않고, xL - PpsVirtualBoundariesPosX[n]가 0 이상 3 미만인 경우, 다음이 적용된다.

$$h_{xL+i} = Clip3(PpsVirtualBoundariesPosX[n], pic_width_in_luma_samples - 1, xL + i) \quad (8\text{-}1229)$$

[0606]

[0607] - 그렇지 않고, 만약

[0608] pps_loop_filter_across_virtual_boundaries_disabled_flag가 1과 같고, 임의의 n = 0..pps_num_ver_virtual_boundaries - 1에 대해, PpsVirtualBoundariesPosX[n] % CtbSizeY가 0과 같지 않고, PpsVirtualBoundariesPosX[n]- xL가 0 보다 크고 4 미만이면, 다음이 적용된다.

$$h_{x+i} = Clip3(0, PpsVirtualBoundariesPosX[n] - 1, xL + i) \quad (8\text{-}1230)$$

[0609]

[0610] - 그렇지 않은 경우, 다음이 적용된다.

$$h_{x+i} = Clip3(0, pic_width_in_luma_samples - 1, xL + i) \quad (8\text{-}1231)$$

[0611]

[0612] - 만약 pps_loop_filter_across_virtual_boundaries_disabled_flag가 1과 같고, 임의의 n = 0..pps_num_hor_virtual_boundaries - 1에 대해, PpsVirtualBoundariesPosY[n] % CtbSizeY가 0과 같지 않고 yL - PpsVirtualBoundariesPosY[n]가 0 이상 3 미만인 경우, 다음이 적용된다.

$$v_{y+j} = Clip3(PpsVirtualBoundariesPosY[n], pic_height_in_luma_samples - 1, yL + j) \quad (8\text{-}1232)$$

[0613]

[0614] - 그렇지 않고, 만약

[0615] pps_loop_filter_across_virtual_boundaries_disabled_flag가 1과 같고, 임의의 n = 0..pps_num_hor_virtual_boundaries - 1에 대해, PpsVirtualBoundariesPosY[n] % CtbSizeY가 0과 같지 않고 PpsVirtualBoundariesPosY[n] - yL이 0 보다 크고 4 보다 작으면, 다음이 적용된다

$$v_{y+j} = Clip3(0, PpsVirtualBoundariesPosY[n] - 1, yL + j) \quad (8\text{-}1233)$$

[0616]

[0617] - 그렇지 않은 경우, 다음이 적용된다.

$$v_{y+j} = Clip3(0, pic_height_in_luma_samples - 1, yL + j) \quad (8\text{-}1234)$$

[0618]

[0619] - 변수 clipLeftPos, clipRightPos, clipTopPos 및 clipBottomPos는 (xCtb, yCtb) 및 (xL - xCtb, yL - yCtb)를 입력들로 하여 8.8.5.5 절에 명시된 바와 같이 ALF 경계 위치 도출 프로세스를 호출하는 것에 의해 도출된다.

[0620] - 수직 샘플 위치 오프셋 yM1, yP1 및 yP2는 표 4-1에 명시된다. 수직 루마 샘플 위치 yL 및 applyAlfLineBufBoundary에 따른 yP1의 사양

Condition	yP1
(yL = = CtbSizeY - 4) && (applyAlfLineBufBoundary = = 1)	0
(yL = = CtbSizeY - 5) && (applyAlfLineBufBoundary = = 1)	0
Otherwise	1

[0621]

[0622] 표 4-2. 수직 루마 샘플 위치 yL 및 applyAlfLineBufBoundary에 따른 yP1의 사양

Condition	yP1
(yL = = CtbSizeY - 4) && (applyAlfLineBufBoundary = = 1)	1
(yL = = CtbSizeY - 5) && (applyAlfLineBufBoundary = = 1)	0
Otherwise	1

[0623]

[0624] - 수직 루마 샘플 위치 yL, clipLeftPos 및 clipRightPos에 따른

[0625] - 수평 샘플 위치 오프셋들 xM1 및 xP1는 수평 루마 샘플 위치 xL, clipLeftPos 및 clipRightPos에 따라 표 y-yyyy에 명시된다.

[0626] - 변수 curr는 다음과 같이 도출된다.

[0627] $$curr = alfPicture_C[xCtbC + x, yCtbC + y] \quad (8\text{-}1286)$$

[0628] - 크로스 컴포넌트 필터 계수들의 어레이 f[j]는 j = 0..7로 아래와 같이 도출된다.

[0629] $$f[j] = CcAlfCoeff[j] \quad (8\text{-}1287)$$

[0630] - 변수 sum은 아래와 같이 도출된다.

$$\begin{aligned} sum = {} & f[0] * recPicture_L[h_x, v_{y+yM1}] + \\ & f[1] * recPicture_L[h_{x+xM1}, v_y] + \\ & f[2] * recPicture_L[h_x, v_y] + \\ & f[3] * recPicture_L[h_{x+xP1}, v_y] + \\ & f[4] * recPicture_L[h_{x+xM1}, v_{y+yP1}] + \\ & f[5] * recPicture_L[h_x, v_{y+yP1}] + \\ & f[6] * recPicture_L[h_{x+xP1}, v_{y+yP1}] + \\ & f[7] * recPicture_L[h_x, v_{y+yP2}] \end{aligned} \quad (8\text{-}1289)$$

$$sum = Clip3(-(1 << (BitDepth_C - 1)), (1 << (BitDepth_C - 1)) - 1, sum) \quad (8\text{-}1290)$$

[0631] $$sum = curr + (sum + 64) >> (7 + (BitDepth_Y - BitDepth_C)) \quad (8\text{-}1290)$$

[0632] - 수정된 필터링된 재구성된 크로마 픽처 샘플 어레이 ccAlfPicture[xCtbC + x][yCtbC + y]는 아래와 같이 도출된다.

[0633] $$ccAlfPicture[xCtbC + x][yCtbC + y] = Clip3(0, (1 << BitDepth_C) - 1, sum) \quad (8\text{-}1291)$$

[0634] *표 x-xx - 수직 루마 샘플 위치 yL, clipTopPos 및 clipBottomPos에 따른 yM1, yP1 및 yP2의 사양*

[0635] 아래 표를 삭제한다.

Condition	yM1	yP1	yP2
yL = = clipTopPos + 1	−1	1	1
yL = = clipTopPos	0	0	1
yL = = clipBottomPos − 1	0	0	1
yL = = clipBottomPos − 2	−1	1	1
Otherwise	−1	1	2

[0636]

[0637] 아래 표를 추가한다.

Condition	yM1	yP1	yP2
yL = = clipTopPos + 1	-1	1	1
yL = = clipTopPos	0	0	0
yL = = clipBottomPos − 1	0	0	0
yL = = clipBottomPos − 2	-1	1	1
Otherwise	−1	1	2

[0638]

[0639] *표 y-yy - 수평 루마 샘플 위치 xL, clipLeftPos 및 clipRightPos에 따른 xM1 및 xP1의 사양*

Condition	xM1	xP1
xL = = clipLeftPos	0	0
xL = = clipRightPos − 1	0	0
xL = = clipRightPos – 2	–1	1
Otherwise	–1	1

[0640]

[0641] 도 32는 본 문서에 개시된 다양한 기술들이 구현될 수 있는 예시적인 비디오 처리 시스템(1900)을 도시하는 블록도이다. 다양한 구현들은 시스템(1900)의 일부 또는 전체 구성요소들을 포함할 수 있다. 시스템(1900)은 비디오 콘텐트를 수신하기 위한 입력(1902)을 포함할 수 있다. 상기 비디오 콘텐트는 8 또는 10 비트 다중 컴포넌트 픽셀 값들과 같은 로(raw) 또는 비압축 포맷으로 수신될 수 있거나 또는 압축 또는 인코딩된 포맷일 수 있다. 입력(1902)은 네트워크 인터페이스, 주변 버스 인터페이스 또는 스토리지 인터페이스를 표시할 수 있다. 네트워크 인터페이스의 예들은 이더넷, 수동 광학 네트워크 (PON: passive optical network) 등과 같은 유선 인터페이스들 및 와이파이 또는 셀룰러 인터페이스들과 같은 무선 인터페이스들을 포함할 수 있다.

[0642] 시스템(1900)은 본 문서에 기술된 다양한 코딩 또는 인코딩 방법들을 구현할 수 있는 코딩 구성요소(1904)를 포함할 수 있다. 코딩 구성요소(1904)는 상기 비디오의 코딩된 표현을 산출하기 위해 코딩 구성요소(1904)의 입력(1902)으로부터 출력까지의 비디오의 평균 비트레이트를 감소시킬 수 있다. 따라서, 코딩 기술들은 종종 비디오 압축 또는 비디오 트랜스코딩 기술로 불린다. 코딩 구성요소(1904)의 출력은 컴포넌트(1906)에 의해 표현되는 것처럼, 저장되거나 또는 연결된 통신을 통해 전송될 수 있다. 입력(1902)에서 수신된 상기 비디오의 저장되거나 통신된 (또는 코딩된) 비트스트림 표현은 컴포넌트(1908)에 의해 사용되어 픽셀 값들을 생성하거나 또는 디스플레이 인터페이스(1910)에 전송되는 디스플레이 가능한 비디오를 생성할 수 있다. 상기 비트스트림 표현으로부터 사용자가 시청가능한 비디오를 생성하는 프로세스는 종종 비디오 압축해제라 불린다. 더 나아가, 특정 비디오 처리 동작들이 "코딩" 동작 또는 툴로 불리는 반면, 상기 코딩 툴 또는 동작들은 인코더 측에서 사용되며 상기 코딩의 결과들을 반전시키는 대응하는 디코딩 툴 또는 동작들은 디코더가 실시할 것이라는 것이 이해될 것이다.

[0643] 주변 버스 인터페이스 또는 디스플레이 인터페이스의 예들은 범용 직렬 버스(USB: universal serial bus) 또는 고해상도 멀티미디어 인터페이스 (HDMI: high definition multimedia interface) 또는 디스플레이포트 등을 포

함할 수 있다. 스토리지 인터페이스의 예들은 SATA(직렬 고급 기술 연결), PCI, IDE 인터페이스 등을 포함한다. 본 문서에서 개시되는 기술들은 이동 전화, 노트북, 스마트폰, 또는 디지털 데이터 처리 및/또는 비디오 디스플레이를 실행할 수 있는 기타 디바이스들과 같은 다양한 전자 디바이스들에서 구현될 수 있다.

[0644] 도 33은 비디오 처리 장치(3600)의 블록도이다. 상기 장치(3600)는 본 명세서에서 개시되는 하나 또는 그 이상의 방법들을 실시하는 데에 사용될 수 있다. 장치(3600)는 스마트폰, 테블릿, 컴퓨터, 사물인터넷(IoT) 수신기 등에서 구현될 수 있다. 장치(3600)는 하나 또는 그 이상의 프로세서들(3602), 하나 또는 그 이상의 메모리들(3604), 그리고 비디오 처리 하드웨어(3606)을 포함할 수 있다. 프로세서(들)(3602)는 본 문서에서 개시되는 하나 또는 그 이상의 방법들을 실시하도록 구성될 수 있다. 메모리(메모리들)(3604)은 본 명세서에 개시된 방법들 및 기술들을 실시하는 데에 사용되는 데이터 및 코드를 저장하는 데에 사용될 수 있다. 비디오 처리 하드웨어(3606)는 본 문서에서 개시되는 일부 기술들을 하드웨어 회로에서 실행할 수 있다.

[0645] 도 35는 본 개시의 기술들을 활용할 수 있는 예시적인 비디오 코딩 시스템(100)을 도시하는 블록도이다.

[0646] 도 35에서 보이는 바와 같이, 비디오 코딩 시스템(100)은 소스 디바이스(110) 및 목적지 디바이스(120)를 포함할 수 있다. 소스 디바이스(110)는 인코딩된 비디오 데이터를 생성시키며, 비디오 인코딩 디바이스로 불릴 수 있다. 목적지 디바이스(120)는 소스 디바이스(110)에 의해 생성된 상기 인코딩된 비디오 데이터를 디코딩 할 수 있으며, 비디오 디코딩 디바이스로 불릴 수 있다.

[0647] 소스 디바이스(110)는 비디오 소스(112), 비디오 인코더(114) 및 입력/출력(I/O) 인터페이스(116)를 포함할 수 있다.

[0648] 비디오 소스(112)는 비디오 캡처 디바이스, 비디오 콘텐트 제공기로부터 비디오 데이터를 수신하기 위한 인터페이스, 그리고/또는 비디오 데이터를 생성하기 위한 컴퓨터 그래픽 시스템과 같은 소스, 또는 이러한 소스들의 조합을 포함할 수 있다. 상기 비디오 데이터는 하나 또는 그 이상의 픽처들을 포함할 수 있다. 비디오 인코더(114)는 비디오 소스(112)로부터의 상기 비디오 데이터를 인코딩하여 비트스트림을 생성한다. 상기 비트스트림은 상기 비디오 데이터의 코딩된 표현을 형성하는 비트들의 시퀀스를 포함할 수 있다. 상기 비트스트림은 코딩된 픽처들 및 연관된 데이터를 포함할 수 있다. 코딩된 픽처는 픽처의 코딩된 표현이다. 연관된 데이터는 시퀀스 파라미터 세트들, 픽처 파라미터 세트들 및 기타 신택스 구조들을 포함할 수 있다. I/O 인터페이스(116)는 변조기/복조기 (모뎀) 및/또는 송신기를 포함할 수 있다. 인코딩된 비디오 데이터는 네트워크(130a)를 통해 I/O 인터페이스(116)를 거쳐 목적지 디바이스(120)로 직접 전송될 수 있다. 상기 인코딩된 비디오 데이터는 또한 목적지 디바이스(120)에 의한 접근을 위해 스토리지 매체/서버(130b)에 저장될 수 있다.

[0649] 목적지 디바이스(120)는 I/O 인터페이스(126), 비디오 디코더(124) 및 디스플레이 디바이스(122)를 포함할 수 있다.

[0650] I/O 인터페이스(126)는 수신기 및/또는 모뎀을 포함할 수 있다. I/O 인터페이스(126)는 소스 디바이스(110) 또는 스토리지 매체/서버(130b)로부터 인코딩된 비디오 데이터를 획득할 수 있다. 비디오 디코더(124)는 상기 인코딩된 비디오 데이터를 디코딩할 수 있다. 디스플레이 디바이스(122)는 상기 디코딩된 비디오 데이터를 사용자에게 보여줄 수 있다. 디스플레이 디바이스(122)는 목적지 디바이스(120)와 통합될 수 있거나 또는 외부 디스플레이 디바이스와 접속하도록 구성될 목적지 디바이스(120)의 외부에 있을 수 있다.

[0651] 비디오 인코더(114) 및 비디오 디코더(124)는 HEVC(고효율 비디오 코딩) 표준, VVC(다기능 비디오 코딩) 표준 및 기타 현행 및/또는 추가 표준들과 같은 비디오 압축 표준에 따라 작동할 수 있다.

[0652] 도 36은 도 35에 도시된 시스템(100) 내의 비디오 인코더(114)일 수 있는 비디오 인코더(200)의 일 예를 도시하는 블록도이다.

[0653] 비디오 인코더(200)는 본 개시의 기술들 중 어느 하나 또는 전부를 실시하도록 구성될 수 있다. 도 36의 예에서, 비디오 인코더(200)는 복수의 기능적 구성요소들을 포함한다. 본 개시에서 설명되는 기술들은 비디오 인코더(200)의 다양한 구성요소들 사이에 공유될 수 있다. 일부 예들에서, 프로세서는 본 개시에서 설명되는 기술들 중 어느 하나 또는 전부를 실시하도록 구성될 수 있다.

[0654] 비디오 인코더(200)의 기능적 구성요소들은 분할부(201), 모드 선택부(203), 모션 추정부(204), 모션 보상부(205) 및 인트라 예측부(206)를 포함할 수 있는 예측부(202), 잔차 생성부(207), 변환부(208), 양자화부(209), 역양자화부(210), 역변환부(211), 복원부(212), 버퍼(213), 그리고 엔트로피 인코딩부(214)를 포함할 수 있다.

[0655] 다른 예들에서, 비디오 인코더(200)는 더 많은 수의, 적은 수의 또는 상이한 기능적 구성요소들을 포함할 수 있

다. 일 예에서, 예측부(202)는 인트라 블록 카피(IBC) 유닛을 포함한다. 상기 IBC 유닛은 적어도 하나의 참조 픽처가 현재 비디오 블록이 위치하는 픽처인 IBC 모드로 예측을 수행할 수 있다.

[0656] 게다가, 모션 추정부(204) 및 모션 보상부(205)와 같은 몇몇 구성요소들은 고도로 통합될 수 있지만, 설명의 목적을 위해 도36의 예에서는 별도로 도시되었다.

[0657] 분할부(201)는 픽처를 하나 또는 그 이상의 비디오 블록들로 분할할 수 있다. 비디오 인코더(200) 및 비디오 디코더(300)는 다양한 비디오 블록 크기들을 지원할 수 있다.

[0658] 모드 선택부(203)는 예를 들어 오류 결과들에 기반하여, 인터 또는 인트라 코딩 모드들 중 하나를 선택할 수 있으며, 그 결과로 인트라 또는 인터 코딩된 블록을 잔차 생성부(207)에 제공하여 잔차 블록 데이터를 생성하도록 하고, 복원부(212)에 제공하여 인코딩된 블록을 참조 픽처로 사용하기 위해 복원하도록 할 수 있다. 몇몇 예들에서, 모드 선택부(203)는 예측이 인트라 예측 신호 및 인터 예측 신호에 기반하는 인트라 인터 결합 예측(CIIP) 모드를 선택할 수 있다. 모드 선택부(203)는 또한 인터 예측의 경우 블록에 대한 모션 벡터를 위한 해상도(예를 들어, 서브 픽셀 또는 정수 픽셀 정밀도)를 선택할 수 있다.

[0659] 현재 비디오 블록에 인터 예측을 실시하기 위해, 모션 추정부(204)는 버퍼(213)에서부터 상기 현재 비디오 블록까지 하나 또는 그 이상의 참조 프레임들을 비교하여 상기 현재 비디오 블록에 대한 모션 정보를 생성할 수 있다. 모션 보상부(205)는 상기 모션 정보 및 버퍼(213)로부터 상기 현재 블록에 관련된 픽처가 아닌 픽처들의 디코딩된 샘플들에 기반하여 상기 현재 비디오에 대한 예측 비디오 블록을 결정할 수 있다.

[0660] 모션 추정부(204) 및 모션 보상부(205)는 현재 비디오 블록에 대해, 예를 들어 상기 현재 비디오 블록이 I 슬라이스, P 슬라이스 또는 B 슬라이스에 있는지 여부에 의존하여, 상이한 동작들을 실시할 수 있다.

[0661] 일부 예들에서, 모션 추정부(204)는 현재 비디오 블록에 대해 단방향(uni-directional) 예측을 실시할 수 있으며, 모션 추정부(204)는 현재 비디오 블록을 위한 참조 비디오 블록에 대해 리스트 0 또는 리스트 1의 참조 픽처들을 탐색할 수 있다. 그리고 나서, 모션 추정부(204)는 참조 비디오 블록을 포함하는 리스트 0 또는 리스트 0에서 참조 픽처를 지시하는 참조 인덱스를 그리고 상기 현재 비디오 블록과 상기 참조 비디오 블록 사이의 공간적 이동(displacement)을 지시하는 모션 벡터를 생성할 수 있다. 모션 추정부(204)는 상기 참조 인덱스, 예측 방향 지시자 그리고 상기 모션 벡터를 상기 비디오 블록의 모션 정보로 출력할 수 있다. 모션 보상부(205)는 상기 현재 비디오 블록의 상기 모션 정보에 의해 지시되는 상기 참조 비디오 블록에 기반하여 상기 현재 블록의 예측 비디오 블록을 생성할 수 있다.

[0662] 다른 예들에서, 모션 추정부(204)는 현재 비디오 블록에 대해 양방향(bi-directional) 예측을 실시할 수 있으며, 모션 추정부(204)는 현재 비디오 블록을 위한 참조 비디오 블록에 대해 리스트 0의 참조 픽처들을 탐색할 수 있고, 현재 비디오 블록을 위한 다른 참조 비디오 블록에 대해 리스트 1의 참조 픽처들을 탐색할 수 있다. 그리고 나서, 모션 추정부(204)는 참조 비디오 블록들을 포함하는 리스트 0 또는 리스트 0에서 참조 픽처들을 지시하는 참조 인덱스를 그리고 상기 현재 비디오 블록과 상기 참조 비디오 블록들 사이의 공간적 이동들(displacements)을 지시하는 모션 벡터들을 생성할 수 있다. 모션 추정부(204)는 상기 참조 인덱스들 및 상기 현재 비디오 블록의 상기 모션 벡터들을 상기 비디오 블록의 모션 정보로 출력할 수 있다. 모션 보상부(205)는 상기 현재 비디오 블록의 상기 모션 정보에 의해 지시되는 상기 참조 비디오 블록들에 기반하여 상기 현재 블록의 예측 비디오 블록을 생성할 수 있다.

[0663] 일부 예들에서, 모션 추정부(204)는 디코더의 디코딩 처리를 위한 모션 정보의 풀 세트를 출력할 수 있다.

[0664] 일부 예들에서, 모션 추정부(204)는 현재 비디오에 대한 모션 정보의 풀 세트를 출력하지 않을 수 있다. 오히려, 모션 추정부(204)는 다른 비디오 블록의 모션 정보를 참조하여 현재 비디오 블록의 모션 정보를 시그널링할 수 있다. 예를 들어, 모션 추정부(204)는 현재 비디오 블록의 모션 정보가 이웃 비디오 블록의 모션 정보와 충분히 유사하다고 판단할 수 있다.

[0665] 일 예에서, 모션 추정부(204)는 현재 비디오 블록과 관련된 신택스 구조에서 현재 비디오 블록이 다른 비디오 블록과 동일한 모션 정보를 가지고 있다는 것을 비디오 디코더(300)에게 지시하는 값을 지시할 수 있다.

[0666] 다른 예에서, 모션 추정부(204)는 현재 비디오 블록과 관련된 신택스 구조에서 다른 비디오 블록 및 모션 벡터 차분(MVD: motion vector difference)을 식별할 수 있다. 상기 모션 벡터 차분은 현재 비디오 블록과 지시되는 비디오 블록의 모션 벡터 사이의 차분을 지시한다. 비디오 디코더(300)는 지시되는 비디오 블록의 모션 벡터 및 모션 벡터 차분을 이용하여 현재 비디오 블록의 모션 벡터를 결정할 수 있다.

[0667] 위에서 논의된 것처럼, 비디오 인코더(200)는 모션 벡터를 예측적으로 시그널링할 수 있다. 비디오 인코더(200)에 의해 실시될 수 있는 예측적 시그널링 기술들의 두 가지 예에는 향상된 모션 벡터 예측(AMVP: advanced motion vector prediction)과 머지 모드 시그널링이 있다.

[0668] 인트라 예측부(206)는 현재 비디오 블록에 대해 인트라 예측을 실시할 수 있다. 인트라 예측부(206)가 현재 비디오 블록에 대해 인트라 예측을 실시하는 경우, 인트라 예측부(206)는 동일한 픽처의 다른 비디오 블록들의 디코딩된 샘플들에 기반하여 현재 비디오 블록에 대한 예측 데이터를 생성할 수 있다. 현재 비디오 블록에 대한 예측 데이터는 예측된 비디오 블록 및 다양한 신택스 요소들을 포함할 수 있다.

[0669] 잔차 생성부(207)는 현재 비디오 블록에서 현재 비디오 블록의 예측 비디오 블록(들)을 차감하여(예를 들어, 마이너스 기호로 지시) 현재 비디오 블록에 대한 잔차 데이터를 생성할 수 있다. 현재 비디오 블록의 상기 잔차 데이터는 현재 비디오 블록의 샘플들의 상이한 샘플 구성요소들에 해당하는 잔차 비디오 블록들을 포함할 수 있다.

[0670] 다른 예들에서, 가령 스킵 모드에서, 현재 비디오 블록에 대한 잔차 데이터가 없을 수 있으며, 잔차 생성부(207)는 감산 동작을 실시하지 않을 수 있다.

[0671] 변환 처리부(208)는 하나 또는 그 이상의 변환들을 현재 비디오 블록과 관련된 잔차 비디오 블록에 적용하여 현재 비디오 블록에 대한 하나 또는 그 이상의 변환 계수 비디오 블록들을 생성할 수 있다.

[0672] 변환 처리부(208)가 현재 비디오 블록과 관련된 변환 계수 비디오 블록을 생성한 후, 양자화부(209)는 현재 비디오 블록과 관련된 하나 또는 그 이상의 양자화 파라미터(QP: quantization parameter) 값들에 기반하여 현재 비디오 블록과 관련된 상기 변환 계수 비디오 블록을 양자화 할 수 있다.

[0673] 역양자화부(210) 및 역변환부(211)는 역양자화 및 역변환을 상기 변환 계수 비디오 블록에 각각 적용하여 상기 변환 계수 비디오 블록으로부터 잔차 비디오 블록을 복원할 수 있다. 복원부(212)는 상기 복원된 잔차 비디오 블록을 예측부(202)에 의해 생성된 하나 또는 그 이상의 예측 비디오 블록들에 해당하는 샘플들에 더하여 버퍼(213)에 저장하기 위해 현재 블록과 관련된 복원 비디오 블록을 생성할 수 있다.

[0674] 복원부(212)가 상기 비디오 블록을 복원한 후에, 루프 필터링 동작이 상기 비디오 블록에서 비디오 블로킹 아티팩트들을 감소시키기 위해 실시될 수 있다.

[0675] 엔트로피 인코딩부(214)는 비디오 인코더(200)의 다른 기능적 구성요소들로부터 데이터를 수신할 수 있다. 엔트로피 인코딩부(214)가 상기 데이터를 수신할 때, 엔트로피 인코딩부(214)는 하나 또는 그 이상의 엔트로피 인코딩 동작들을 실시하여 엔트로피 인코딩된 데이터를 생성하고 상기 엔트로피 인코딩된 데이터를 포함하는 비트스트림을 출력할 수 있다.

[0676] 도 37은 도 35에 도시된 시스템(100) 내의 비디오 디코더(114)일 수 있는 비디오 디코더(300)의 일 예를 도시하는 블록도이다.

[0677] 비디오 디코더(300)는 본 개시의 기술들 중 어느 하나 또는 전부를 실시하도록 구성될 수 있다. 도 37의 예에서, 비디오 디코더(300)는 복수의 기능적 구성요소들을 포함한다. 본 개시에서 설명되는 기술들은 비디오 디코더(300)의 다양한 구성요소들 사이에 공유될 수 있다. 일부 예들에서, 프로세서는 본 개시에서 설명되는 기술들 중 어느 하나 또는 전부를 실시하도록 구성될 수 있다.

[0678] 도 37의 예에서, 비디오 디코더(300)는 엔트로피 디코딩부(301), 모션 보상부(302), 인트라 예측부(303), 역양자화부(304), 역변환부(305), 복원부(306), 그리고 버퍼(307)를 포함한다. 일부 예들에서, 비디오 디코더(300)는 비디오 인코더(200) (도 36)과 관련하여 설명된 인코딩 패스에 일반적으로 상반된 디코딩 패스를 실시할 수 있다.

[0679] 엔트로피 디코딩부(301)는 인코딩된 비트스트림을 검색할 수 있다. 상기 인코딩된 비트스트림은 엔트로피 코딩된 비디오 데이터(예를 들어, 비디오 데이터의 인코딩된 블록들)을 포함할 수 있다. 엔트로피 디코딩부(301)는 상기 엔트로피 코딩된 비디오 데이터를 디코딩할 수 있으며, 엔트로피 디코딩된 비디오 데이터로부터, 모션 보상부(302)는 모션 벡터들, 모션 벡터 정밀도 및 참조 픽처 리스트 인덱스들을 포함하는 모션 정보 및 기타 모션 정보를 결정할 수 있다. 예를 들어, 모션 보상부(302)는 AMVP 및 머지 모드를 실행하여 이러한 정보를 결정할 수 있다.

[0680] 모션 보상부(302)는 가능한 경우 보간 필터들에 기반한 보간을 실시하여 움직임 보상된 블록들을 산출할 수 있

다. 서브 픽셀 정밀도와 함께 사용될 보간 필터들에 대한 식별자들은 신택스 요소들에 포함될 수 있다.

[0681] 모션 보상부(302)는 비디오 엔코더(200)가 비디오 블록의 인코딩 동안 사용한 것과 같이 보간 필터들을 사용하여 참조 블록의 서브-정수 픽셀들에 대한 보간된 값들을 계산할 수 있다. 모션 보상부(302)는 비디오 인코더(200)가 사용한 상기 보간 필터들을 검색된 신택스 정보에 따라 결정할 수 있으며, 상기 보간 필터들을 사용하여 예측적 블록들을 생성할 수 있다.

[0682] 모션 보상부(302)는 인코딩된 비디오 시퀀스의 프레임(들) 및/또는 슬라이스(들)을 인코딩 하는 데에 사용된 블록들의 크기들을 판단하기 위한 일부 신택스 정보, 인코딩된 비디오 시퀀스의 픽처의 각 매크로블록이 어떻게 분할되는지를 기술하는 분할 정보, 각 파티션이 어떻게 인코딩되었는지를 지시하는 모드들, 각각의 인트라 인코딩된 블록에 대한 하나 또는 그 상의 참조 프레임들 (및 참조 프레임 리스트들), 그리고 인코딩된 비디오 시퀀스를 디코딩하기 위한 기타 정보를 이용할 수 있다.

[0683] 인트라 예측부(303)는 공간적으로 인접한 블록들로부터 예측 블록을 형성하기 위해 예를 들어 비트스트림에서 수신된 인트라 예측 모드들을 이용할 수 있다. 역양자화부(303)는 비트스트림에서 제공되며 엔트로피 디코딩부(301)에 의해 디코딩된 양자화된 비디오 블록 계수들을 역 양자화(즉 양자화 해제)한다. 역변환부(305)는 역변환을 적용한다.

[0684] 복원부(306)는 모션 보상부(202) 또는 인트라 예측부(303)에 의해 생성된 해당 예측 블록들과 잔차 블록들을 합계하여 디코딩된 블록들을 형성할 수 있다. 요구되는 경우, 디블로킹 필터 또한 블록화 아티팩트(blockiness artifacts)를 제거하기 위해 디코딩된 블록들의 필터링에 적용될 수 있다. 그리고 나서, 디코딩된 비디오 블록들은 버퍼(307)에 저장되며, 버퍼는 후속 모션 보상/인트라 예측을 위한 참조 블록들을 제공하고, 또한 디스플레이 장치상에 제시하기 위한 디코딩된 비디오를 산출한다.

[0685] 일부 실시예들은 아래의 조항 기반(clause-based) 포맷을 사용하여 기술될 수 있다. 첫 번째 세트의 조항들은 이전 섹션 (예를 들어, 실시예들의 항목 1)에서 논의된 기술들의 예시적인 실시예들을 보여준다.

[0686] 1. 비디오 처리 방법으로서 (예를 들어 도 34의 방법(3400)), 비디오의 비디오 유닛과 상기 비디오의 코딩된 표현 간의 변환을 위해, 기준에 근거하여, 크로스 컴포넌트 적응적 루프 필터 동작을 사용할 것을 결정하는 단계(3402)를 포함하고, 여기서 상기 크로스 컴포넌트 적응적 루프 필터는 가용하지 않은 루마 샘플들을 위한 미러링된 패딩 기술을 사용하고, 상기 결정에 기반하여 상기 변환을 실시하는 단계(3404)를 포함한다. 본 문서는 크로스 컴포넌트 적응적 루프 필터의 다양한 실시예들, 그것의 동작 및 미러링된 패딩 기술들, 그리고 가상 버퍼 경계에 대한 그것들의 관계 또한 개시한다.

[0687] 2. 조항 1에 있어서, 상기 미러링된 패딩 기술이 상기 가용하지 않은 루마 샘플의 하나 또는 그 이상의 해당 루마 샘플들을 도출하기 위해 더 사용되는, 비디오 처리 방법.

[0688] 3. 조항 2에 있어서, 상기 하나 또는 그 이상의 해당 루마 샘플들은 대표 루마 샘플로부터 상기 하나 또는 그 이상의 해당 루마 샘플들의 거리 또는 상기 대표 루마 샘플로부터 상기 가용하지 않은 샘플의 거리에 기반하여 결정되는, 비디오 처리 방법.

[0689] 4. 조항 3에 있어서, 상기 대표 루마 샘플은, 상기 크로스 컴포넌트 적응적 루프 필터링 기술이 사용되는 크로마 샘플의 위치에 해당하는, 비디오 처리 방법.

[0690] 5. 조항 3에 있어서, 상기 대표 루마 샘플의 위치는 상기 비디오의 컬러 포맷에 의존하는, 비디오 처리 방법.

[0691] 6. 조항 3 내지 조항 4 중 어느 것에 있어서, 상기 거리는 상기 하나 또는 그 이상의 루마 샘플들을 포함하는 제2 방향 상의 픽셀 라인과 상기 대표 샘플을 포함하는 행 사이의 제1 방향 상의 거리에 해당하는, 비디오 처리 방법.

[0692] 7. 조항 6에 있어서, C는 상기 대표 루마 샘플이 위치하는 상기 제2 방향 상의 중앙 라인을 나타내고, M은 상기 가용하지 않은 샘플이 위치하는 상기 제2 방향 상의 라인을 나타내고, N은 상기 하나 또는 그 이상의 루마 샘플들이 위치하는 상기 제2 방향 상의 라인을 나타내고, 여기서 C, M 및 N은 양의 정수들이고, M은 N과 같지 않으며, 그리고 상기 미러링된 패딩 기술이 상기 크로스 컴포넌트 적응적인 루프 필터의 크기 및 모양에 기반하여 적용되는, 비디오 처리 방법.

[0693] 8. 조항 1에 있어서, 상기 크로스 컴포넌트 적응적 루프 필터는 KxL 필터 모양을 가지고, 이때, K는 짝수

이고 L은 양의 정수이며, 그리고 상기 미러링된 패딩 기술은, 상기 크로스 컴포넌트 적응적 루프 필터의 가상 경계로부터 M 또는 N 만큼 떨어져 제2 방향 상의 라인에 위치하고 상기 가상 경계 근처의 최근접 샘플 라인으로부터 패딩되는 가용하지 않는 샘플들을 패딩하는 단계를 포함하는, 비디오 처리 방법.

[0694] 9. 조항 3에 있어서, 상기 크로스 컴포넌트 적응적 루프 필터의 가상 경계가 상기 제2 방향 상에서 상기 대표 루마 샘플 아래에 있는 경우, 상기 제2 방향 상에서 상기 가상 경계 위에(above) 있는 최근접 라인을 사용하여 상기 가용하지 않은 샘플들을 패딩하는, 비디오 처리 방법.

[0695] 10. 조항 3에 있어서, 가용하지 않은 샘플이 행 M에 위치하고, M은 C 보다 작은 정수이고, C는 상기 대표 루마 샘플의 상기 제2 방향 상의 중앙 라인을 지시하는 정수인 경우, d(C, M) = d(N, C) - 오프셋이거나, 이때 오프셋은 정수 값, 또는 d(C, M) < d(N, C)일 때, 이때 d( )는 거리 함수, 상기 제2 방향 상의 라인 N에 위치하는 샘플은 상기 해당 샘플들로 판단되는, 비디오 처리 방법.

[0696] 아래 조항들은 이전 섹션(예를 들어, 항목 2)에서 논의된 기술들의 예시적인 실시예들을 보여준다.

[0697] 11. 조항 1내지 조항 10 중 어느 것에 있어서, 상기 제1 방향은 수직 방향이고 상기 제2 방향은 수평 방향인, 비디오 처리 방법.

[0698] 12. 조항 1내지 조항 10 중 어느 것에 있어서, 상기 제1 방향은 수평 방향이고 상기 제2 방향은 수직 방향인, 비디오 처리 방법.

[0699] 13. 조항 11내지 조항 12 중 어느 것에 있어서, 상기 제1 방향 및 상기 제2 방샹의 오리엔테이션은 상기 가상 버퍼의 경계의 오리엔테이션에 의존하는, 비디오 처리 방법.

[0700] 아래 조항들은 이전 섹션(예를 들어, 항목 3)에서 논의된 기술들의 예시적인 실시예들을 보여준다.

[0701] 14. 조항 1내지 조항 13 중 어느 것에 있어서, 상기 비디오 유닛은 비디오 픽처, 비디오 서브픽처, 비디오 슬라이스, 비디오 타일, 또는 비디오의 360도 경계를 포함하는, 비디오 처리 방법.

[0702] 15. 조항 1내지 조항 14 중 어느 것에 있어서, 상기 변환을 실시하는 단계는 상기 비디오를 인코딩하여 상기 코딩된 표현을 생성하는 것을 포함하는, 비디오 처리 방법.

[0703] 16. 조항 1내지 조항 14 중 어느 것에 있어서, 상기 변환을 실시하는 단계는 상기 코딩된 표현을 파싱하고 디코딩하여 상기 비디오를 생성하는 것을 포함하는, 비디오 처리 방법.

[0704] 상기 개시된 조항들에서, 상기 오리엔테이션은 수평 또는 수직 일 수 있으며, 그에 따라 상기 제1 방향 및 상기 제2 방향은 픽셀 열들 및 픽셀 행들로 지칭되는 수직 또는 수평 방향들일 수 있다.

[0705] 17. 비디오 디코딩 장치로서, 조항 1 내지 조항 16 중 하나 또는 그 이상에서의 방법을 실시하도록 구성되는 프로세서를 포함하는 비디오 처리 장치.

[0706] 18. 비디오 인코딩 장치로서, 조항 1 내지 조항 16 중 하나 또는 그 이상에서의 방법을 구현하도록 구성되는 프로세서를 포함하는 비디오 인코딩 장치.

[0707] 19. 컴퓨터 프로그램 제품으로, 코드를 안에 저장하며, 상기 코드는, 프로세서에 의해 실행시에, 상기 프로세서로 하여금 조항 1 내지 조항 16 중 어느 것에서의 방법을 실시하도록 하는, 컴퓨터 프로그램 제품.

[0708] 20. 본 문서에 기술되는 방법, 장치 또는 시스템.

[0709] 두 번째 세트의 조항들은 이전 섹션들 (예를 들어, 실시예들의 예시들의 항목 1 내지 항목 3)에서 논의된 기술들의 예시적인 실시예들을 보여준다.

[0710] 1. 비디오 처리 방법으로서 (예를 들어 도 38A의 방법(3810)), 비디오의 비디오 유닛과 상기 비디오의 비트스트림 표현 간의 변환을 위해, 상기 비디오 유닛으로의 루프 필터링 툴 적용 동안 가용하지 않은 루마 샘플을 패딩하기 위한 미러링된 패딩 프로세스를 활성화할 것인지 여부를 결정하는 단계(3812) 그리고 상기 결정에 기반하여 상기 변환을 실시하는 단계(3814)를 포함하는, 비디오 처리 방법.

[0711] 2. 조항 1에 있어서, 상기 루프 필터링 툴은 크로스 컴포넌트 적응적 루프 필터링(CC-ALF: cross-component adaptive loop filtering) 툴을 포함하는, 비디오 처리 방법.

[0712] 3. 조항 1에 있어서, 상기 루프 필터링 툴은 적응적 루프 필터링(ALF: adaptive loop filtering) 툴을 포함하는, 비디오 처리 방법.

[0713] 4. 조항 1에 있어서, 상기 미러링된 패딩 프로세스는, 제2 샘플이 가용한 경우에도, 상기 루프 필터링 툴에서 필터 서포트 영역 내의 제1 샘플의 해당(corresponding) 샘플인 상기 제2 샘플을 패딩하는 단계를 포함하고, 상기 제1 샘플은 가용하지 않고 패딩되어야 하는, 비디오 처리 방법.

[0714] 5. 조항 1에 있어서, 상기 미러링된 패딩 프로세스는 상기 가용하지 않은 루마 샘플 및 상기 가용하지 않은 루마 샘플에 해당하는 하나 또는 그 이상의 루마 샘플들을 도출하는 데에 더 이용되는, 비디오 처리 방법.

[0715] 6. 조항 5에 있어서, 해당 루마 샘플은, 대표 루마 샘플로부터 상기 해당 루마 샘플의 거리 및/또는 상기 대표 루마 샘플로부터 상기 가용하지 않은 샘플의 거리에 기반하여 결정되는, 비디오 처리 방법.

[0716] 7. 조항 6에 있어서, 상기 대표 루마 샘플은 필터링될 크로마 샘플의 대응(collocated) 루마 샘플로 정의되는, 비디오 처리 방법.

[0717] 8. 조항 7에 있어서, 상기 크로마 샘플의 상기 대응 루마 샘플의 위치는 상기 비디오의 컬러 포맷에 의존하는, 비디오 처리 방법.

[0718] 9. 조항 8에 있어서, (x, y)에 위치하는 상기 크로마 샘플의 상기 대응 루마 샘플은 4:2:0인 컬러 포맷에서 (2x, 2y)에 위치하는 것으로 정의되는, 비디오 처리 방법.

[0719] 10. 조항 8에 있어서, (x, y)에 위치하는 상기 크로마 샘플의 상기 대응 루마 샘플은 4:2:2인 컬러 포맷에서 (2x, y)에 위치하는 것으로 정의되는, 비디오 처리 방법.

[0720] 11. 조항 8에 있어서, (x, y)에 위치하는 상기 크로마 샘플의 상기 대응 루마 샘플은 4:4:4인 컬러 포맷에서 (x, y)에 위치하는 것으로 정의되는, 비디오 처리 방법.

[0721] 12. 조항 6에 있어서, 상기 거리는 상기 해당 루마 샘플을 포함하는 제1 행 및 상기 대표 루마 샘플을 포함하는 제2 행 사이의 제1 방향 상의 거리를 참조하는, 비디오 처리 방법.

[0722] 13. 조항 6에 있어서, 상기 거리는 상기 해당 루마 샘플을 포함하는 제2 방향 상의 픽셀 라인과 상기 대표 루마 샘플을 포함하는 행 사이의 제1 방향 상의 차이로 계산되는, 비디오 처리 방법.

[0723] 14. 조항 13에 있어서, C는 상기 대표 루마 샘플이 위치하는 상기 제2 방향 상의 중앙 라인을 나타내고, M은 상기 가용하지 않은 샘플이 위치하는 상기 제2 방향 상의 라인을 나타내고, N은 상기 해당 루마 샘플이 위치하는 상기 제2 방향 상의 라인을 나타내고, 이때 C, M 및 N은 양의 정수들이고, M은 N과 같지 않은, 비디오 처리 방법.

[0724] 15. 조항 5에 있어서, 상기 미러링된 패딩 프로세스에서 패딩될 상기 하나 또는 그 이상의 해당 루마 샘플들은 상기 CC-ALF 툴에 의해 사용되는 필터의 모양에 의해 활용되는 샘플들의 행들의 수에 기반하는, 비디오 처리 방법.

[0725] 16. 조항 5에 있어서, 상기 가용하지 않은 루마 샘플은 행 M에 위치하고, 행 N에 위치하는 샘플들은 패딩될 상기 하나 또는 그 이상의 해당 루마 샘플들로 판단되며, 그리고 d(C, M) = d (N, C)인, 그에 따라 d(x, y)는 행 x와 행 y 사이의 거리를 나타내고 M, C 및 N은 양의 정수들인, 비디오 처리 방법.

[0726] 17. 조항 6에 있어서, 1) 중앙 루마 샘플이 상기 대표 루마 샘플로 선택되는 경우 및 2) 상기 CC-ALF 툴이 K x L 필터 모양을 가지는 경우, 그에 따라 K는 홀수이고 L은 양의 정수, 상기 미러링된 패딩 프로세스는 적응적 루프 필터링(ALF: adaptive loop filtering) 툴의 적용 동안 사용되는 것에 해당하는, 비디오 처리 방법.

[0727] 18. 조항 6에 있어서, 상기 CC-ALF 툴이 K x l 필터 모양을 가지고 상기 가용하지 않은 루마 샘플이 가상 경계로부터 위의 행 M 또는 아래의 행 N에 위치하는 경우, 상기 미러링된 패딩 프로세스는 상기 가상 경계로부터 떨어진 행 N 또는 M에 위치한 상기 하나 또는 그 이상의 해당 루마 샘플들을 상기 행 N 위 또는 행 M 아래의 최근접 샘플로부터 패딩하는 단계를 포함하며, 그에 따라 K는 홀수이고, L, M 및 N은 양의 정수들인, 비디오 처리 방법.

[0728] 19. 조항 18에 있어서, 상기 가상 경계가 상기 대표 루마 샘플 위에 있는 경우 상기 가용하지 않은 루마 샘플은 상기 가상 경계 아래의 최근접 행을 사용하여 패딩되고, 상기 가상 경계가 상기 대표 루마 샘플 위에 있는 경우 상기 하나 또는 그 이상의 해당 루마 샘플들은 상기 하나 또는 그 이상의 해당 루마 샘플들을 포함하는 행 위의 최근접 행을 사용하여 패딩되는, 비디오 처리 방법.

[0729] 20. 조항 18에 있어서, 상기 가상 경계가 상기 대표 루마 샘플 아래에 있는 경우 상기 가용하지 않은 루마

샘플은 상기 가상 경계 위의 최근접 행을 사용하여 패딩되고, 상기 가상 경계가 상기 대표 루마 샘플 아래에 있는 경우 상기 하나 또는 그 이상의 해당 루마 샘플들은 상기 하나 또는 그 이상의 해당 루마 샘플들을 포함하는 행 아래의 최근접 행을 사용하여 패딩되는, 비디오 처리 방법.

[0730] 21. 조항 19 또는 20에 있어서, K=2, yL=0, yP1=1이고, 상기 가상 경계가 CtbSizeY-4와 같으며, 그에 따라 yL 및 yP1은 두 샘플 행들의 y 좌표들이고 CtbSizeY는 코딩 트리 유닛(CTU: coding tree unit)의 크기를 나타내는, 비디오 처리 방법.

[0731] 22. 조항 21에 있어서, yL이 CtbSizeY - 4와 같고 yL 위의 행이 가용하지 않은 경우, yP1에서 행 내의 샘플 (x, yP1)은 yL에서 행 내의 샘플 (x, yL)을 사용하여 패딩되는, 비디오 처리 방법.

[0732] 23. 조항 19 또는 20에 있어서, K=4, yM1=-1, yL=0, yP1=1, yP2=2이고, 상기 가상 경계가 CtbSizeY-4와 같으며, 그에 따라 yM1, yL, yP1 및 yP2는 네 개 샘플 행들의 y 좌표들이고 CtbSizeY는 코딩 트리 유닛(CTU: coding tree unit)의 크기를 나타내는, 비디오 처리 방법.

[0733] 24. 조항 23에 있어서, yL이 CtbSizeY - 3과 같고 yM1 위의 행이 가용하지 않은 경우, yP2에서 행 내의 샘플 (x, yP2)는 yP1에서 행 내의 샘플 (x, yP1)을 사용하여 패딩되는, 비디오 처리 방법.

[0734] 25. 조항 23에 있어서, yL이 CtbSizeY - 4와 같고 yM1 위의 행이 가용하지 않은 경우, yM1에서 행 내의 샘플 (x, yM1)은 yL에서 행 내의 샘플 (x, yL)을 사용하여 패딩되고, yP2 및 yP1에서 행들 내의 샘플들 (x, yP2) 및 (x, yP1)은 yL에서 행 내의 샘플 (x, yL)을 사용하여 패딩되는, 비디오 처리 방법.

[0735] 26. 조항 19 또는 20에 있어서, K=6, yM2=-2, yM1=-1, yL=0, yP1=1, yP2=2, yP3=3이고, 상기 가상 경계가 CtbSizeY-4와 같으며, 그에 따라 yM2, yM1, yL, yP1, yP2 및 yP3는 여섯 개 샘플 행들의 y 좌표들이고 CtbSizeY는 코딩 트리 유닛(CTU: coding tree unit)의 크기를 나타내는, 비디오 처리 방법.

[0736] 27. 조항 26에 있어서, yL이 CtbSizeY - 2과 같고 yM2 위의 행이 가용하지 않은 경우, yP3에서 행 내의 샘플 (x, yP3)는 yP2에서 행 내의 샘플 (x, yP2)를 사용하여 패딩되는, 비디오 처리 방법.

[0737] 28. 조항 26에 있어서, yL이 CtbSizeY - 3과 같고 yM2 위의 행 및 yM2에서의 행이 가용하지 않은 경우, yM2에서 행 내의 샘플 (x, yM2)는 yM1에서 행 내의 샘플 (x, yM1)을 사용하여 패딩되고, yP3 및 yP2에서 행들 내의 샘플들 (x, yP3) 및 (x, yP2)는 yP1에서 행 내의 샘플 (x, yP1)을 사용하여 패딩되는, 비디오 처리 방법.

[0738] 29. 조항 26에 있어서, yL이 CtbSizeY - 4와 같고 yM2 위의 행, yM2에서의 행 그리고 yM1에서의 행이 가용하지 않은 경우, yM2 및 yM1에서 행들 내의 샘플들 (x, yM2) 및 (x, yM1)은 yL에서 행 내의 샘플 (x, yL)을 사용하여 패딩되고, yP3, yP2 및 yP1에서 행들 내의 샘플들 (x, yP3), (x, yP2) 및 (x, yP1)은 yL에서의 상기 행 내의 샘플 (x, yL)을 사용하여 패딩되는, 비디오 처리 방법.

[0739] 30. 조항 21에 있어서, yL이 CtbSizeY - 5와 같고 yP1에서의 행이 가용하지 않은 경우, yP1에서 행 내의 샘플 (x, yP1)은 yL에서 행 내의 샘플 (x, yL)을 사용하여 패딩되는, 비디오 처리 방법.

[0740] 31. 조항 23에 있어서, yL이 CtbSizeY - 6와 같고 yP2에서의 행이 가용하지 않은 경우, yP2에서 행 내의 샘플 (x, yP2)은 yP1에서 행 내의 샘플 (x, yP1)을 사용하여 패딩되는, 비디오 처리 방법.

[0741] 32. 조항 23에 있어서, yL이 CtbSizeY - 5와 같고yP2 및 yP1 에서의 행들이 가용하지 않은 경우, yP1에서 행 내의 샘플 (x, yP1) 및 yP2에서 행 내의 샘플 (x, yP2)가 yL에서 행 내의 샘플 (x, yL)을 사용하여 패딩되고, yM1에서 행 내의 샘플 (x, yM1)은 yL에서의 행 내의 샘플 (x, yL)을 사용하여 패딩되는, 비디오 처리 방법.

[0742] 33. 조항 26에 있어서, yL이 CtbSizeY - 7과 같고 yP3에서의 행이 가용하지 않은 경우, yP3에서 상기 행 내의 샘플 (x, yP3)는 yP2에서 행 내의 샘플 (x, yP2)를 사용하여 패딩되는, 비디오 처리 방법.

[0743] 34. 조항 26에 있어서, yL이 CtbSizeY - 6와 같고yP3에서의 행 및 yP2 에서의 행이 가용하지 않은 경우, yP3에서 상기 행 내의 샘플 (x, yP3) 및 yP2에서 상기 행 내의 샘플 (x, yP2)가 yP1에서 행 내의 샘플 (x, yP1)을 사용하여 패딩되고, 샘플 (x, yM2)는 yM1에서 행 내의 샘플 (x, yM1)을 사용하여 패딩되는, 비디오 처리 방법.

[0744] 35. 조항 26에 있어서, yL이 CtbSizeY - 5와 같고yP3, yP2 및 yP1에서의 행들이 가용하지 않은 경우, yP3에서 행 내의 샘플 (x, yP3), yP2에서 행 내의 샘플 (x, yP2), 그리고 yP1에서 행 내의 샘플 (x, yP1)이 yL에

서 행 내 샘플 (x, yL)을 사용하여 패딩되고, yM2에서 행 내 샘플 (x, yM2) 및 yM1에서 행 내 샘플 (x, yM1)은 yL에서 상기 행 내의 샘플 (x, yL)을 사용하여 패딩되는, 비디오 처리 방법.

[0745] 36. 조항 5에 있어서, 상기 가용하지 않은 루마 샘플은 행 M에 위치하고, 행 N에 위치하는 샘플들은 패딩될 상기 하나 또는 그 이상의 해당 루마 샘플들로 판단되며, 그리고 d(C, M) = d (N, C) - 오프셋 또는 d(C, M) < d(N, C)인, 그에 따라 d(x, y)는 행 x와 행 y 사이의 거리를 나타내고, 오프셋은 정수이며, M, C 및 N은 양의 정수들인, 비디오 처리 방법.

[0746] 37. 조항 5에 있어서, 상기 가용하지 않은 루마 샘플은 행 M에 위치하고, 행 N에 위치하는 샘플들은 패딩될 상기 하나 또는 그 이상의 해당 루마 샘플들로 판단되며, 그리고 d(M, C) = d(C, N) - 오프셋 또는 d(C, M) < d(N, C)인, 그에 따라 d(x, y)는 행 x와 행 y 사이의 거리를 나타내고, 오프셋은 정수이며, M, C 및 N은 양의 정수들인, 비디오 처리 방법.

[0747] 38. 조항 36 또는 37에 있어서, 상기 오프셋은 1과 같은, 비디오 처리 방법.

[0748] 39. 조항 36에 있어서, 상기 가용하지 않은 루마 샘플이 가상 경계로부터 위의 행 M 또는 아래의 행 N에 위치하는 경우, 상기 미러링된 패딩 프로세스는 상기 가상 경계 아래의 행 N 또는 상기 가상 경계 위의 행 M에 위치한 상기 하나 또는 그 이상의 해당 루마 샘플들을 상기 행 N 위 또는 행 M 아래의 최근접 샘플로부터 패딩하는 단계를 포함하며, 그에 따라 M 및 N은 양의 정수들인, 비디오 처리 방법.

[0749] 40. 조항 39에 있어서, 상기 가상 경계가 상기 대표 루마 샘플 아래에 있는 경우 상기 가용하지 않은 루마 샘플은 상기 가상 경계 위의 최근접 행을 사용하여 패딩되고, 상기 하나 또는 그 이상의 해당 루마 샘플들은 상기 하나 또는 그 이상의 해당 루마 샘플들을 포함하는 행 아래의 최근접 행을 사용하여 패딩되는, 비디오 처리 방법.

[0750] 41. 조항 39에 있어서, 상기 가상 경계가 상기 대표 루마 샘플 위에 있는 경우 상기 가용하지 않은 루마 샘플은 상기 가상 경계 아래의 최근접 행을 사용하여 패딩되고, 상기 하나 또는 그 이상의 해당 루마 샘플들은 상기 하나 또는 그 이상의 해당 루마 샘플들을 포함하는 행 위의 최근접 행을 사용하여 패딩되는, 비디오 처리 방법.

[0751] 42. 조항 40 또는 41에 있어서, K=2, yL=0, yP1=1이고, 상기 가상 경계가 CtbSizeY-4와 같으며, 그에 따라 yL 및 yP1은 두 샘플 행들의 y 좌표들이고 CtbSizeY는 코딩 트리 유닛(CTU: coding tree unit)의 크기를 나타내는, 비디오 처리 방법.

[0752] 43. 조항 42에 있어서, yL이 CtbSizeY - 5와 같고 yP1에서의 행이 가용하지 않은 경우, yP1에서 행 내의 샘플 (x, yP1)은 yL에서 행 내의 샘플 (x, yL)을 사용하여 패딩되는, 비디오 처리 방법.

[0753] 44. 조항 40 또는 41에 있어서, K=4, yM1=-1, yL=0, yP1=1, yP2=2이고, 상기 가상 경계가 CtbSizeY-4와 같으며, 그에 따라 yM1, yL, yP1 및 yP2는 네 개 샘플 행들의 y 좌표들이고 CtbSizeY는 코딩 트리 유닛(CTU: coding tree unit)의 크기를 나타내는, 비디오 처리 방법.

[0754] 45. 조항 44에 있어서, yL이 CtbSizeY - 4와 같고 yM1 위의 행 및 yM1에서의 행이 가용하지 않은 경우, yM1에서 행 내의 샘플 (x, yM1)는 yL에서 행 내의 샘플 (x, yL)을 사용하여 패딩되고, yP2에서 행 내의 샘플 (x, yP2)는 yP1에서 행 내의 샘플 (x, yP1)을 사용하여 패딩되는, 비디오 처리 방법.

[0755] 46. 조항 44에 있어서, yL이 CtbSizeY - 6와 같고 yP2에서의 행이 가용하지 않은 경우, yP2에서 상기 행 내의 샘플 (x, yP2)은 yP1에서 행 내의 샘플 (x, yP1)을 사용하여 패딩되는, 비디오 처리 방법.

[0756] 47. 조항 44에 있어서, yL이 CtbSizeY - 5와 같고yP2 및 yP1 에서의 행들이 가용하지 않은 경우, 행 yP2에서 샘플 (x, yP2)가 yL에서 행 내의 샘플 (x, yL)을 사용하여 패딩되고, yM1에서 행 내의 샘플 (x, yM1)은 yL에서의 상기 행 내의 샘플 (x, yL)을 사용하여 패딩되는, 비디오 처리 방법.

[0757] 48. 조항 40 또는 41에 있어서, K=6, yM2=-2, yM1=-1, yL=0, yP1=1, yP2=2, yP3=3이고, 상기 가상 경계가 CtbSizeY-4와 같으며, 그에 따라 yM2, yM1, yL, yP1, yP2 및 yP3는 여섯 개 샘플 행들의 y 좌표들이고 CtbSizeY는 코딩 트리 유닛(CTU: coding tree unit)의 크기를 나타내는, 비디오 처리 방법.

[0758] 49. 조항 48에 있어서, yL이 CtbSizeY - 3과 같고 yM2 위의 행 및 yM2에서의 행이 가용하지 않은 경우, yM2에서 행 내의 샘플 (x, yM2)는 yM1에서 행 내의 샘플 (x, yM1)을 사용하여 패딩되고, yP3에서 행 내의 샘플 (x, yP3))는 yP2에서 행 내의 샘플 (x, yP2)을 사용하여 패딩되는, 비디오 처리 방법.

[0759] 50. 조항 48에 있어서, yL이 CtbSizeY - 4와 같고 yM2 위의 행, yM2에서의 행 그리고 yM1에서의 행이 가용하지 않은 경우, yM2 및 yM1에서 행들 내의 샘플들 (x, yM2) 및 (x, yM1)은 yL에서 행 내의 샘플 (x, yL)을 사용하여 패딩되고, yP3 및 yP2에서 행들 내의 샘플들 (x, yP3) 및 (x, yP2)는 yP1에서의 행 내의 샘플 (x, yP1)을 사용하여 패딩되는, 비디오 처리 방법.

[0760] 51. 조항 48에 있어서, yL이 CtbSizeY - 7과 같고 yP3에서의 행이 가용하지 않은 경우, yP3에서 상기 행 내의 샘플 (x, yP3)는 yP2에서 행 내의 샘플 (x, yP2)를 사용하여 패딩되고, yM2에서 행 내의 샘플 (x, yM2)는 yM1에서 행 내의 샘플 (x, yM1)을 사용하여 패딩되는, 비디오 처리 방법.

[0761] 52. 조항 48에 있어서, yL이 CtbSizeY - 6와 같고yP3에서의 행 및 yP2 에서의 행이 가용하지 않은 경우, yP3에서 상기 행 내의 샘플 (x, yP3) 및 yP2에서 상기 행 내의 샘플 (x, yP2)가 yP1에서 행 내의 샘플 (x, yP1)을 사용하여 패딩되고, yM2에서 행 내의 샘플 (x, yM2) 및 yM1에서 행 내의 샘플 (x, yM1)은 yL에서 행 내의 샘플 (x, yL)을 사용하여 패딩되는, 비디오 처리 방법.

[0762] 53. 조항 48에 있어서, yL이 CtbSizeY - 5와 같고yP3, yP2 및 yP1에서의 행들이 가용하지 않은 경우, yP3에서 행 내의 샘플 (x, yP3), yP2에서 행 내의 샘플 (x, yP2), 그리고 yP1에서 행 내의 샘플 (x, yP1)이 yL에서 행 내 샘플 (x, yL)을 사용하여 패딩되고, yM2에서 행 내 샘플 (x, yM2) 및 yM1에서 행 내 샘플 (x, yM1)은 yL에서 상기 행 내의 샘플 (x, yL)을 사용하여 패딩되는, 비디오 처리 방법.

[0763] 54. 조항 1에 있어서, 상기 결정의 결과가 시퀀스 레벨, 픽처 레벨, 슬라이스 레벨 또는 타일 그룹 레벨에서 상기 비트스트림 표현에 포함되는, 비디오 처리 방법.

[0764] 55. 이전 조항들 중 어느 것에 있어서, 상기 제1 방향이 수직 방향이고, 상기 제2 방향이 수평 방향인, 비디오 처리 방법.

[0765] 56. 이전 조항들 중 어느 것에 있어서, 상기 제1 방향이 수평 방향이고, 상기 제2 방향이 수직 방향인, 비디오 처리 방법.

[0766] 57. 조항 55 또는 56에 있어서, 상기 제1 방향 및 상기 제2 방향의 오리엔테이션은 가상 버퍼의 경계의 오리엔테이션에 의존하는, 비디오 처리 방법.

[0767] 58. 비디오 처리 방법으로서 (예를 들어, 도 38B의 방법(3820)), 비디오의 비디오 유닛과 상기 비디오의 비트스트림 표현 간의 변환을 위해, 상기 비디오 유닛의 코딩된 정보에 기반하여 가상 경계에 위치하는 샘플을 패딩하기 위해 반복 패딩 프로세스 및/또는 미러링된 패딩 프로세스를 적용할 것인지 여부를 결정하는 단계(3822) 그리고 상기 결정에 기반하여 상기 변환을 실시하는 단계(3824)를 포함하는, 비디오 처리 방법.

[0768] 59. 조항 58에 있어서, 상기 코딩된 정보는 코딩 트리 유닛(CTU) 또는 코딩 트리 블록(CTB)인 상기 비디오 유닛의 크기를 포함하는, 비디오 처리 방법.

[0769] 60. 조항 59에 있어서, 상기 미러링된 패딩 프로세스는 상기 CTU 또는 CTB 크기가 T 이상인 경우에 적용되며, 그에 따라 T는 양의 정수인, 비디오 처리 방법.

[0770] 61. 조항 59에 있어서, 상기 반복 패딩 프로세스는 상기 CTU 또는 CTB 크기가 T 이하인 경우에 적용되며, 그에 따라 T는 양의 정수인, 비디오 처리 방법.

[0771] 62. 이전 조항들 중 어느 것에 있어서, 상기 비디오 유닛은 상기 비디오의 픽처, 서브픽처, 슬라이스, 타일 또는 360도 경계를 포함하는, 비디오 처리 방법.

[0772] 63. 이전 조항들 중 어느 것에 있어서, 상기 CC-ALF 툴에서, 비디오 성분의 상기 비디오 유닛의 동일한 값들이 다른 비디오 성분의 상기 비디오 유닛의 동일한 값들로부터 예측되는, 비디오 처리 방법.

[0773] 64. 조항 1 내지 조항 63 중 어느 것에 있어서, 상기 변환은 상기 비디오를 상기 비트스트림 표현으로 인코딩하는 단계를 포함하는, 비디오 처리 방법.

[0774] 65. 조항 1 내지 조항 63 중 어느 것에 있어서, 상기 변환은 상기 비트스트림 표현으로부터 상기 비디오를 디코딩하는 단계를 포함하는, 비디오 처리 방법.

[0775] 66. 비디오 처리 장치로서, 제1항 내지 제65항 중 어느 하나 또는 그 이상에서의 방법을 실시하도록 구성되는 프로세서를 포함하는 비디오 처리 장치.

[0776] 67. 실행시에 프로세서로 하여금 조항 1 내지 조항 65 중 어느 하나 또는 그 이상에서의 방법을 실시하도록 하는 프로그램 코드를 저장하는 컴퓨터 판독가능 매체.

[0777] 68. 상기 기술된 방법들 중 어느 것에 따라 생성된 코딩된 표현 또는 비트스트림 표현을 저장하는 컴퓨터 판독가능 매체.

[0778] 본 문서에서, 용어 "비디오 처리"는 비디오 인코딩, 비디오 디코딩, 비디오 압축 또는 비디오 압축 해제를 가리킬 수 있다. 예를 들어, 비디오 압축 알고리즘들은 비디오의 픽셀 표현으로부터 상응하는 비트스트림 표현으로 변환하거나 그 반대의 경우에 적용될 수 있다. 현재 비디오 블록의 비트스트림 표현은, 예를 들어, 신택스의 의해 정의되는 대로, 비트스트림 내에서 같은 위치에 있거나(co-located) 상이한 곳에 분산되어 있는 비트들에 해당할 수 있다. 예를 들어, 매크로블록은 변환되고 코딩된 오류 잔차 값들로 그리고 헤더들 및 비트스트림의 다른 필드들의 비트들을 사용하여 인코딩될 수 있다.

[0779] 개시된 그리고 기타 솔루션들, 예들, 실시예들, 모듈들 및 본 문서에 기술된 기능적 동작들은 디지털 전자 회로, 본 문서에 개시된 구조들, 그것들의 구조적 등가물들을 포함하는 컴퓨터 소프트웨어, 펌웨어, 또는 하드웨어 또는 이들 중의 하나 이상의 조합으로 구현될 수 있다. 개시된 그리고 기타 실시예들은 하나 이상의 컴퓨터 프로그램 제품, 즉, 데이터 프로세싱 장치에 의한 실행 또는 그의 동작의 제어를 위해 컴퓨터 판독가능 매체 상에 인코딩되는 컴퓨터 프로그램 명령들의 하나 이상의 모듈로서 구현될 수 있다. 컴퓨터 판독가능 매체는 머신-판독가능 저장 장치, 머신-판독가능 저장 기판, 메모리 장치, 머신-판독가능 전파 신호에 영향을 주는 물질의 구성 또는 그들 중의 하나 이상의 조합일 수 있다. 용어 "데이터 처리 장치"는 데이터 처리를 위한 모든 장치, 디바이스 및 머신을 포함하며, 예를 들어 프로그램 가능한 프로세서, 컴퓨터 또는 다중 프로세서 또는 컴퓨터를 포함한다. 장치는 하드웨어에 더하여 문제의 컴퓨터 프로그램에 대한 실행 환경을 생성하는 코드, 예를 들어 프로세서 펌웨어, 프로토콜 스택, 데이터베이스 관리 시스템, 운영 체제, 또는 이들 중 하나의 조합을 구성하는 코드를 포함할 수 있다. 전파되는 신호는 인공적으로 생성된 신호, 예를 들어, 머신 생성 전기, 광, 또는 전자기 신호이며, 이는 적절한 수신기 장치로의 전송을 위한 정보를 인코딩하기 위해 생성된다.

[0780] 컴퓨터 프로그램 (프로그램, 소프트웨어, 소프트웨어 애플리케이션, 스크립트 또는 코드라고도 알려짐)은 컴파일 되거나 해석된 언어를 포함하여 모든 형태의 프로그래밍 언어로 작성될 수 있으며, 독립형 프로그램 또는 컴퓨터 환경에서 사용되기에 적합한 모듈, 컴포넌트, 서브루틴 또는 다른 유닛으로서 임의의 형태로 배치될 수 있다. 컴퓨터 프로그램은 파일 시스템 내의 파일에 반드시 대응되는 것은 아니다. 프로그램은 다른 프로그램들 또는 데이터를 보유하는 파일의 일부(예를 들어 마크업 언어 문서에 저장된 하나 이상의 스크립트) 또는 문제의 프로그램 전용인 단일 파일 또는 다중 편성(coordinated) 파일들(예를 들어, 하나 이상의 모듈들, 서브 프로그램들 또는 코드의 부분들을 저장하는 파일들)에 저장될 수 있다. 컴퓨터 프로그램은, 하나의 컴퓨터 상에서 또는 한 사이트에 위치하거나 다수의 사이트에 걸쳐 분배되고 통신 네트워크에 의해 상호 접속되는 다수의 컴퓨터 상에서 실행되도록 배치될 수 있다.

[0781] 본 문서에 설명된 프로세스들 및 논리 흐름들은 입력 데이터에 대해 작동하고 출력을 생성함으로써 기능들을 수행하기 위해 하나 이상의 컴퓨터 프로그램들을 실행하는 하나 이상의 프로그래밍가능한 프로세서들에 의해 수행될 수 있다. 프로세스들 및 논리 흐름들 그리고 장치는 FPGA (필드 프로그래밍 가능 게이트 어레이) 또는 ASIC (애플리케이션 특정 집적 회로)와 같은 특수 목적 논리회로로 실시될 수 있고, 구현될 수도 있다.

[0782] 컴퓨터 프로그램의 실행에 적합한 프로세서들은 예를 들어 범용 및 특수 목적의 마이크로 프로세서들과 임의의 종류의 디지털 컴퓨터의 임의의 하나 이상의 프로세서들을 포함한다. 일반적으로, 프로세서는 판독 전용 메모리(ROM), 또는 랜덤 액세스 메모리(RAM), 또는 둘 다로부터 명령어들과 데이터를 수신할 것이다. 컴퓨터의 필수 구성요소는 명령들을 실행하는 프로세서 및 명령들 및 데이터를 저장하는 하나 이상의 메모리 디바이스들이다. 일반적으로, 컴퓨터는 데이터를 저장하기 위한 하나 이상의 대용량 저장 디바이스들, 예를 들어, 자기 디스크, 광자기 디스크, 또는 광디스크를 포함하거나, 또는 이 디바이스들과 데이터를 송수신하기 위하여 동작적으로(operatively) 결합될 것이다. 하지만 컴퓨터는 이러한 디바이스를 구비할 필요는 없다. 컴퓨터 프로그램 명령어들과 데이터를 저장하기 적합한 컴퓨터 판독가능 매체에는, 예를 들어, 반도체 메모리 디바이스(예를 들어, EPROM, EEPROM, 플래시 메모리 디바이스), 자기 디스크(예를 들어, 내부 하드디스크, 착탈식 디스크), 광자기 디스크 및 CD ROM과 DVD-ROM 디스크를 포함하는 모든 형태의 비휘발성 메모리, 매체 및 메모리 디바이스가 포함된다. 프로세서와 메모리는 전용 논리 회로에 의해 보완되거나 또는 전용 논리 회로에 통합될 수 있다.

[0783] 본 특허문서가 다수의 특정한 세부사항을 포함하고 있지만, 이는 주제의 범위나 청구할 사항의 범위에 대한 어떠한 제한으로도 이해되어서는 안 되며, 특정한 기술들의 특정한 실시예들에 고유할 수 있는 특징들의 설명으로

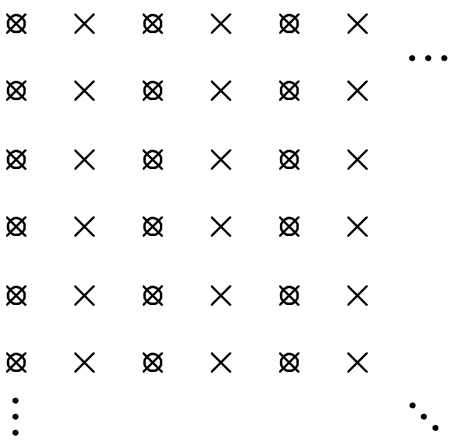
서 이해되어야 한다. 별개의 실시예들의 맥락에서 본 특허 문서에서 설명된 소정의 특징들은 조합되어 단일 실시예로 구현될 수도 있다. 반대로, 단일 실시예의 맥락에서 설명한 다양한 특징들은 복수의 실시예에서 별개로 구현되거나 어떤 적당한 하위 조합으로도 구현 가능하다. 또한, 앞에서 특징들이 특정 조합에서 동작하는 것으로서 설명되고 그와 같이 애초에 청구되기까지 하였지만, 청구된 조합으로부터의 하나 또는 그 이상의 특징들은 일부 경우에 해당 조합으로부터 삭제될 수 있으며, 청구된 조합은 하위 조합이나 하위 조합의 변형으로 될 수 있다.

[0784] 마찬가지로, 동작들이 도면들에서 특정한 순서로 묘사되고 있지만, 이는 그러한 동작들이 바람직한 결과를 얻기 위해, 도시한 특정 순서나 순차적인 순서로 수행되어야 한다거나, 도시된 모든 동작들이 수행되어야 한다는 것을 의미하지 않는다. 더욱이, 본 특허 문서에 기재된 실시예들에 있어서 다양한 시스템 구성요소들의 분리는 모든 실시예에서 그러한 분리를 요구하는 것으로 이해되어서는 안 된다.

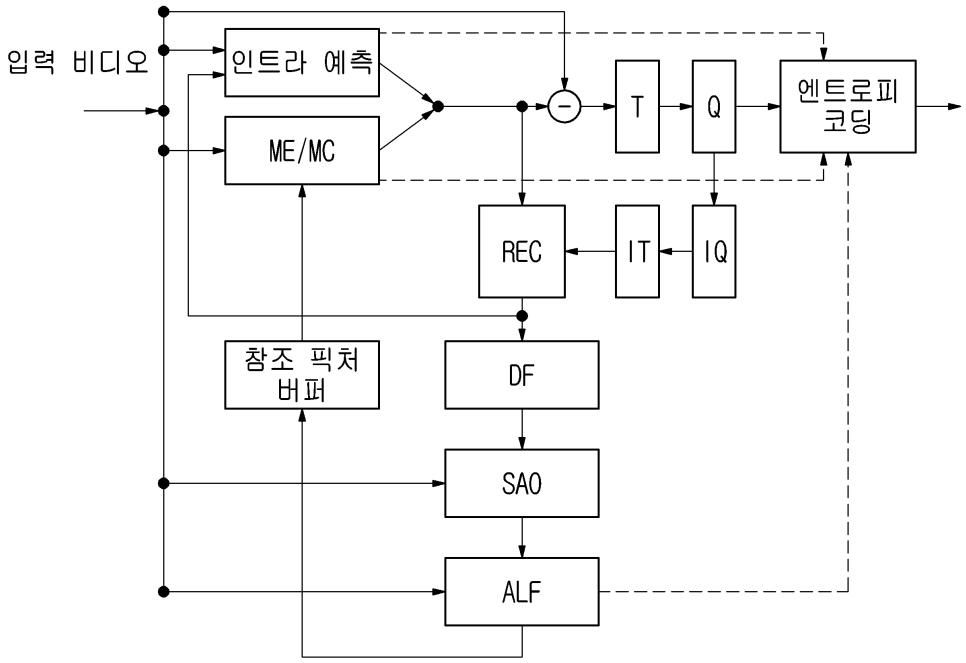
[0785] 단지 몇 가지 구현들 및 예시들이 설명되었으며, 기타 구현, 개선 및 변형은 본 특허 문서에 설명되고 도시된 것에 기초하여 이루어질 수 있다.

도면

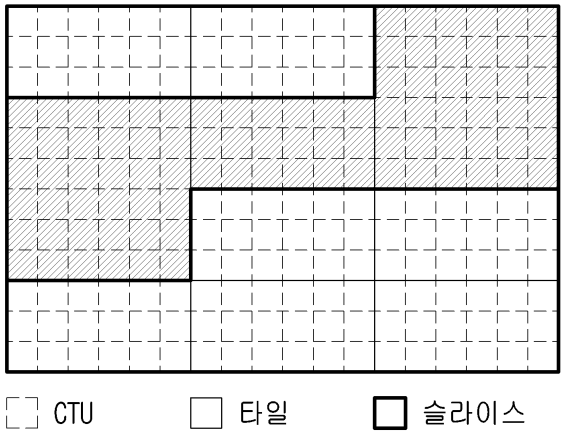
도면1



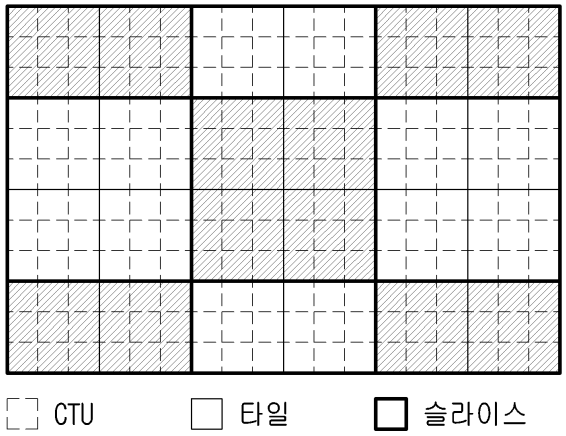
도면2


입력 비디오
인트라 예측
ME/MC
T
Q
엔트로피 코딩
REC
IT
IQ
참조 픽처 버퍼
DF
SAO
ALF

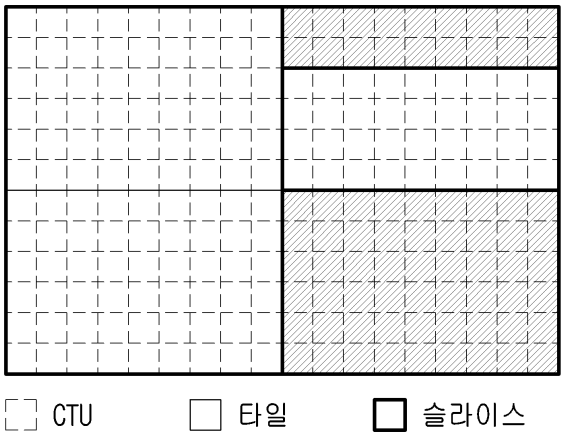
도면3


CTU
타일
슬라이스

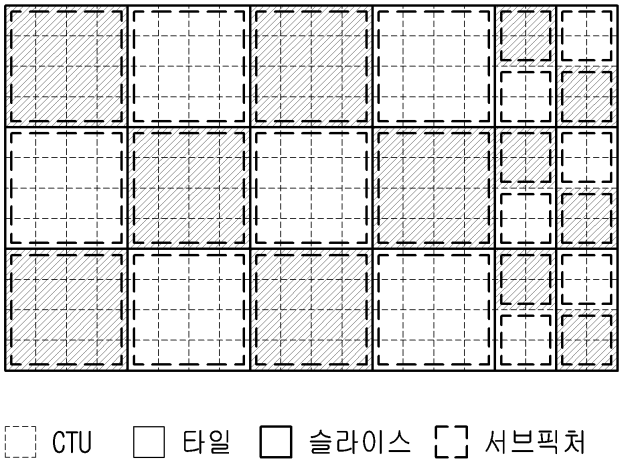
도면4


CTU
타일
슬라이스

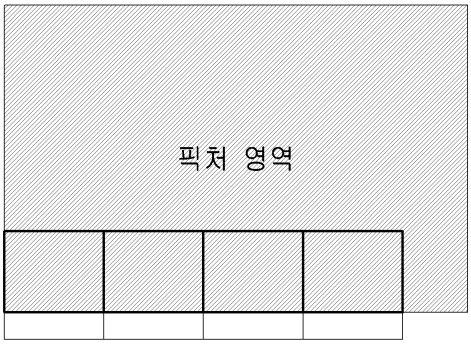
도면5



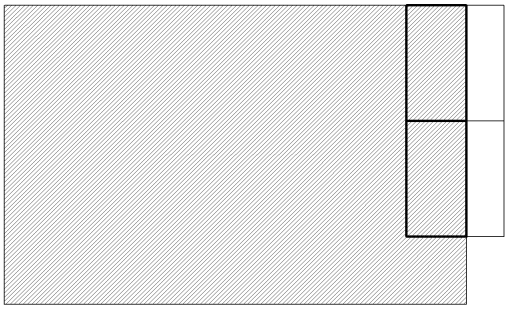
도면6



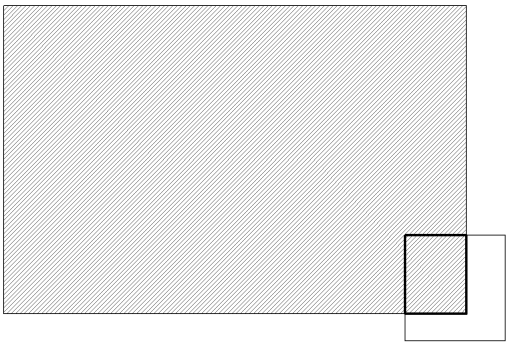
도면7a



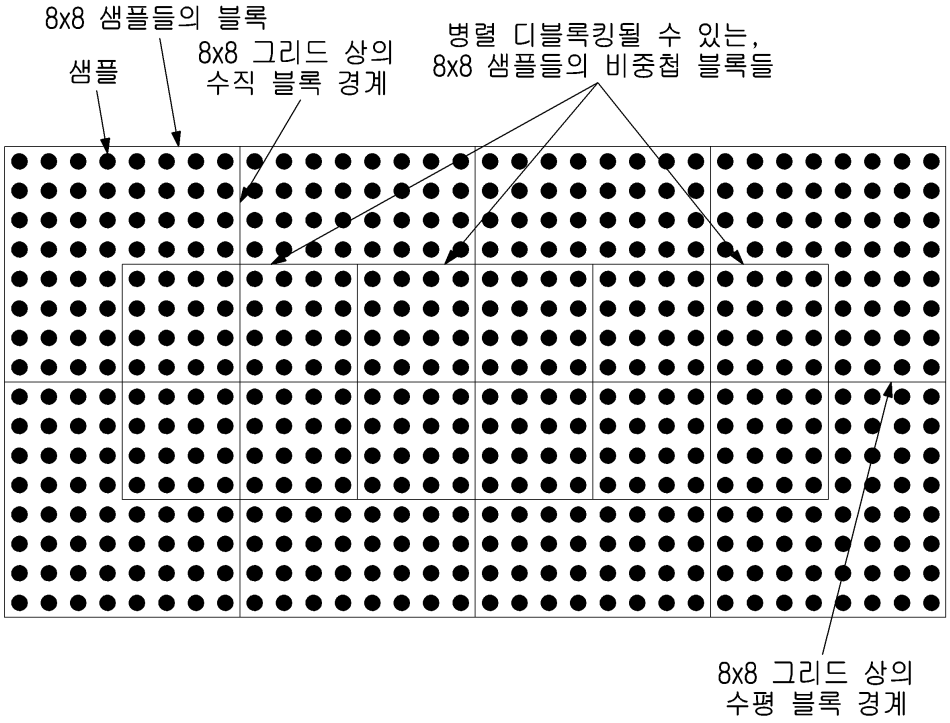
도면7b



도면7c



도면8



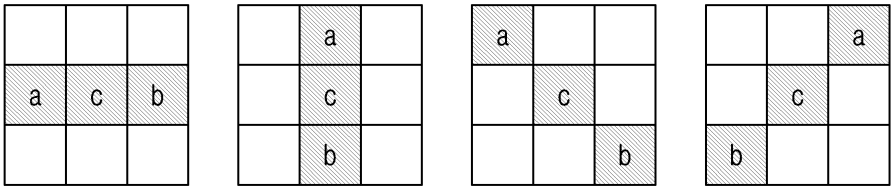
도면9

$p3_0$	$p2_0$	$p1_0$	$p0_0$	$q0_0$	$q1_0$	$q2_0$	$q3_0$
$p3_1$	$p2_1$	$p1_1$	$p0_1$	$q0_1$	$q1_1$	$q2_1$	$q3_1$
$p3_2$	$p2_2$	$p1_2$	$p0_2$	$q0_2$	$q1_2$	$q2_2$	$q3_2$
$p3_3$	$p2_3$	$p1_3$	$p0_3$	$q0_3$	$q1_3$	$q2_3$	$q3_3$
$p3_4$	$p2_4$	$p1_4$	$p0_4$	$q0_4$	$q1_4$	$q2_4$	$q3_4$
$p3_5$	$p2_5$	$p1_5$	$p0_5$	$q0_5$	$q1_5$	$q2_5$	$q3_5$
$p3_6$	$p2_6$	$p1_6$	$p0_6$	$q0_6$	$q1_6$	$q2_6$	$q3_6$
$p3_7$	$p2_7$	$p1_7$	$p0_7$	$q0_7$	$q1_7$	$q2_7$	$q3_7$

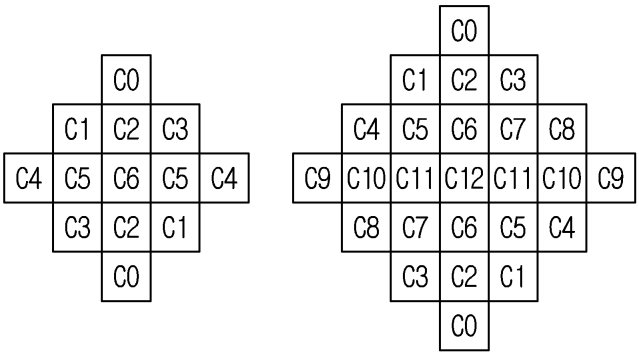
첫 번째 4 라인들

두 번째 4 라인들

도면10



도면11



도면12a

V		V		V		V	
	V		V		V		V
V		V		V		V	
	V		V		V		V
V		V		V		V	
	V		V		V		V
V		V		V		V	
	V		V		V		V

도면12b

H		H		H		H	
	H		H		H		H
H		H		H		H	
	H		H		H		H
H		H		H		H	
	H		H		H		H
H		H		H		H	
	H		H		H		H

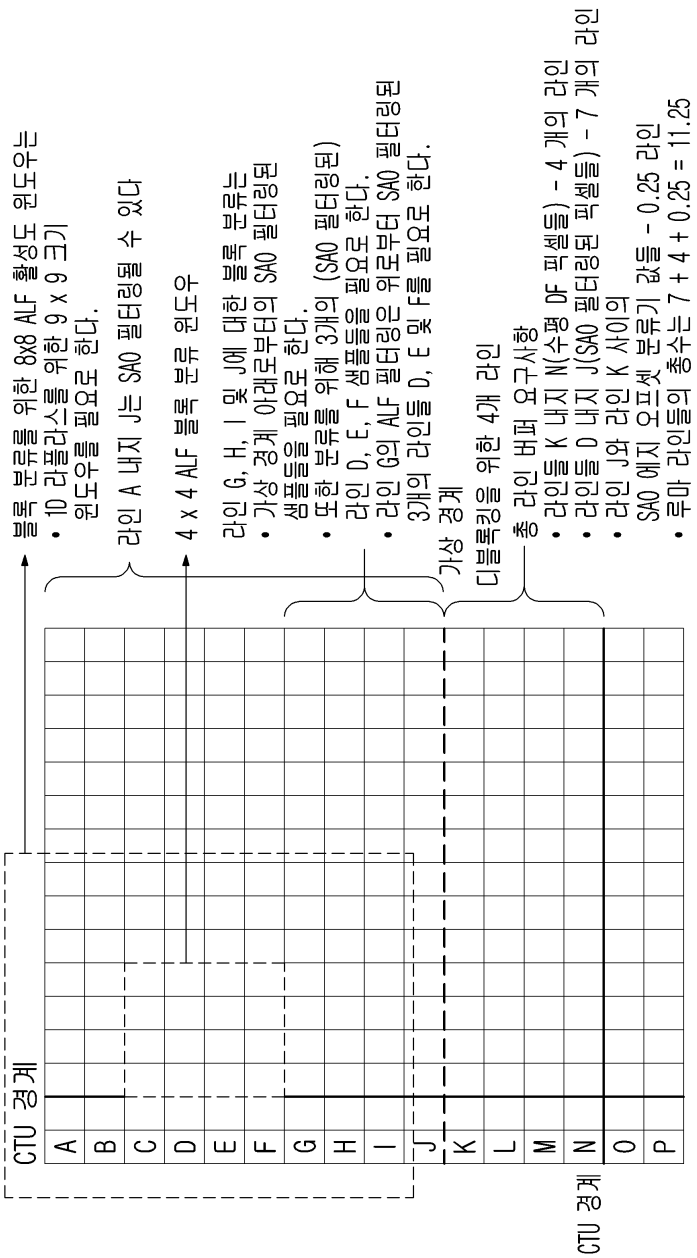
도면12c

D1		D1		D1		D1	
	D1		D1		D1		D1
D1		D1		D1		D1	
	D1		D1		D1		D1
D1		D1		D1		D1	
	D1		D1		D1		D1
D1		D1		D1		D1	
	D1		D1		D1		D1

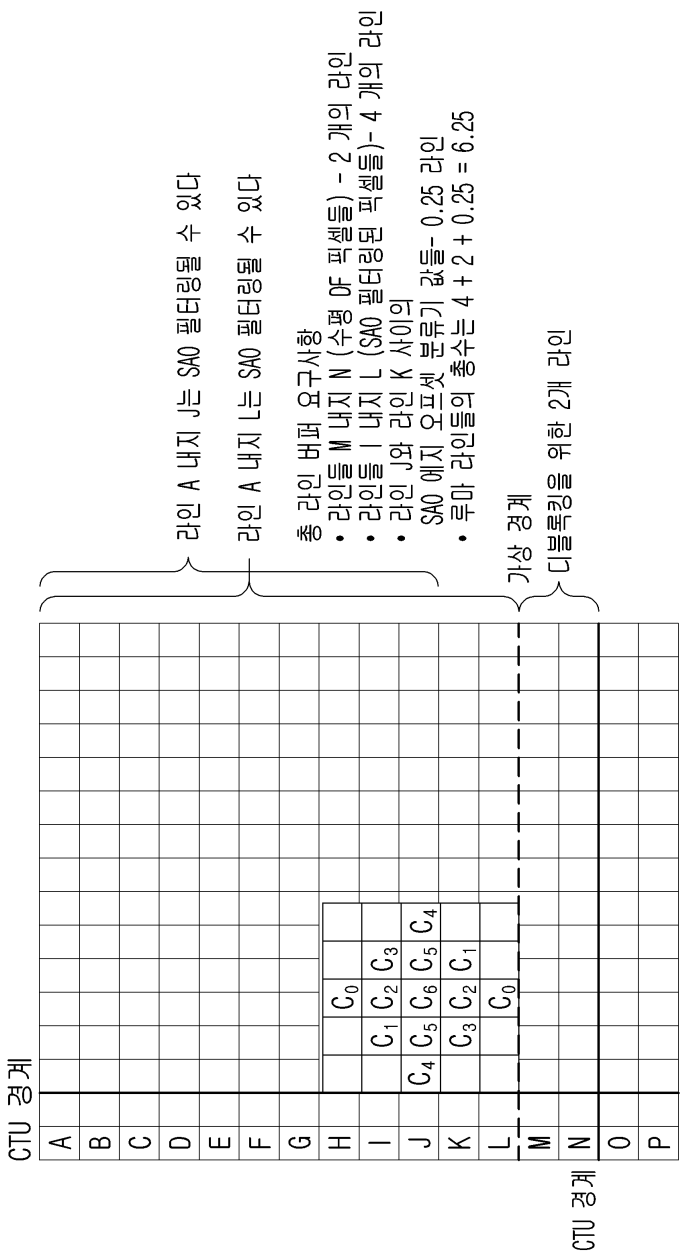
도면12d

D2		D2		D2		D2	
	D2		D2		D2		D2
D2		D2		D2		D2	
	D2		D2		D2		D2
D2		D2		D2		D2	
	D2		D2		D2		D2
D2		D2		D2		D2	
	D2		D2		D2		D2

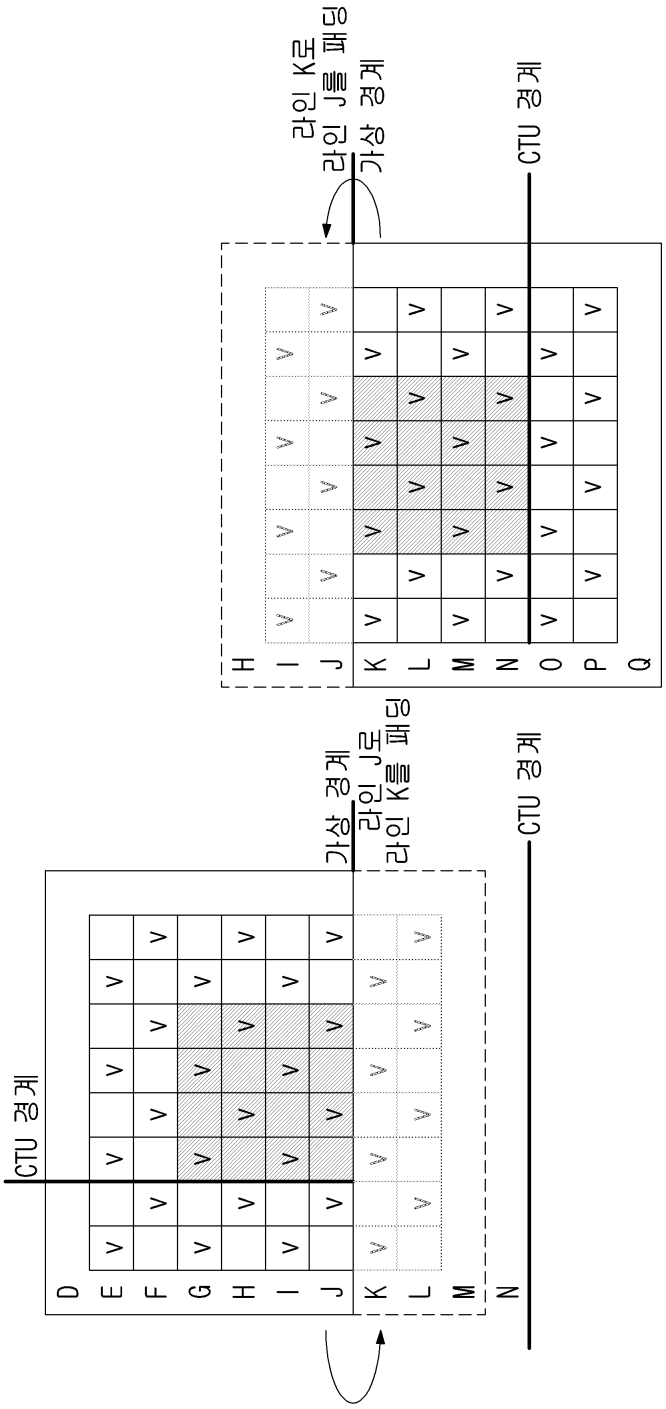
도면13



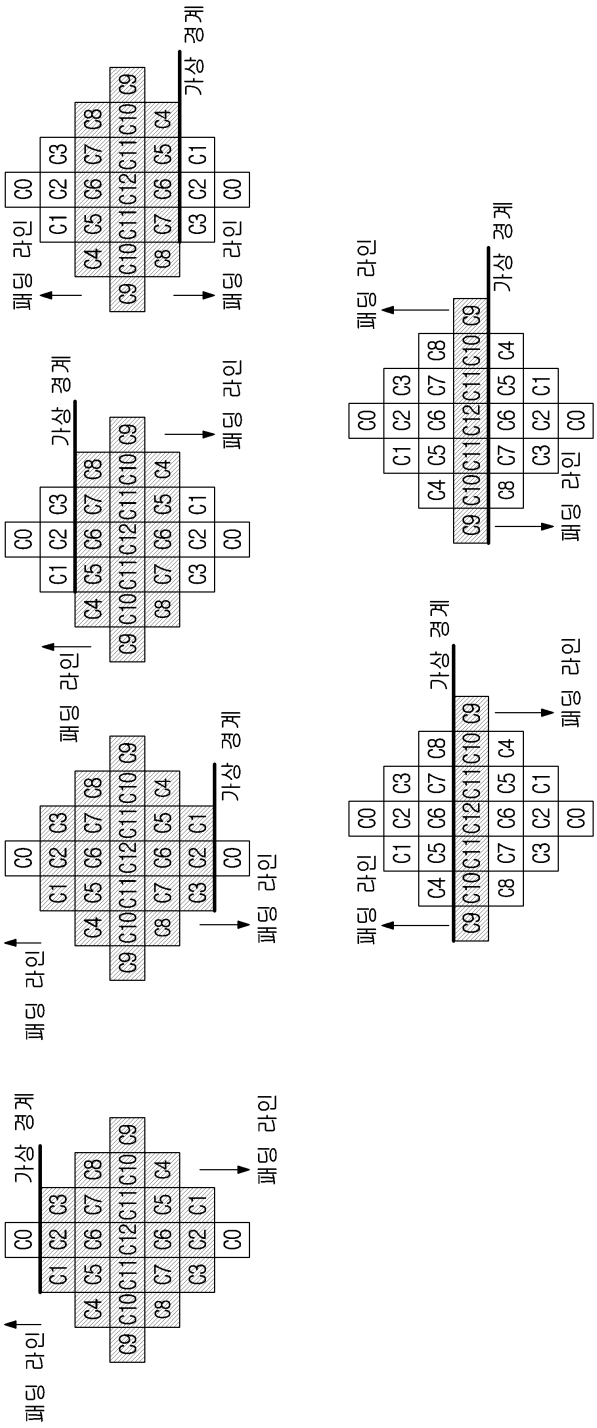
도면14


라인 A 내지 J는 SAO 필터링될 수 있다
라인 A 내지 L는 SAO 필터링될 수 있다
총 라인 버퍼 요구사항
• 라인들 M 내지 N (수평 DF 픽셀들) - 2 개의 라인
• 라인들 I 내지 L (SAO 필터링된 픽셀들)- 4 개의 라인
• 라인 J와 라인 K 사이의
SAO 에지 오프셋 분류기 값들- 0.25 라인
• 루마 라인들의 총수는 4 + 2 + 0.25 = 6.25
가상 경계
디블록킹을 위한 2개 라인
CTU 경계
CTU 경계
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
C0
C1 C2 C3
C4 C5 C6 C4
C3 C2 C1
C0

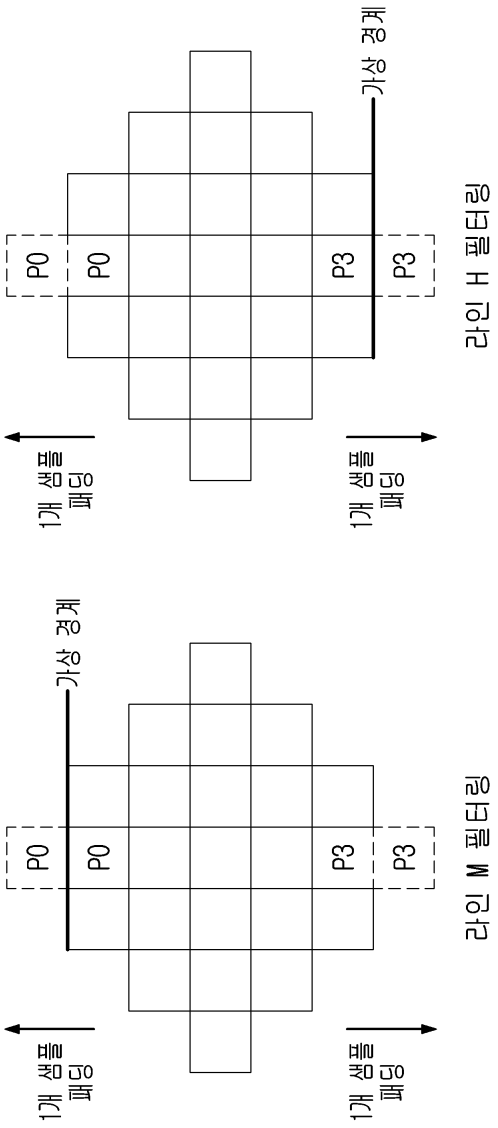
도면15


라인 K로 라인 J를 패딩
가상 경계
CTU 경계
H I J K L M N O P Q
가상 경계
라인 J로 라인 K를 패딩
CTU 경계
CTU 경계
D E F G H I J K L M N

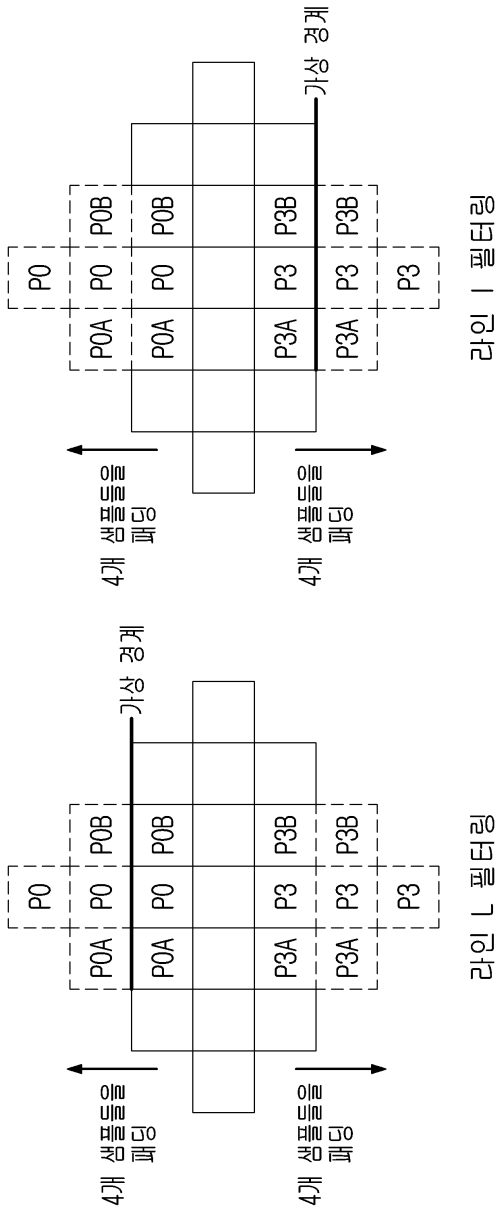
도면16



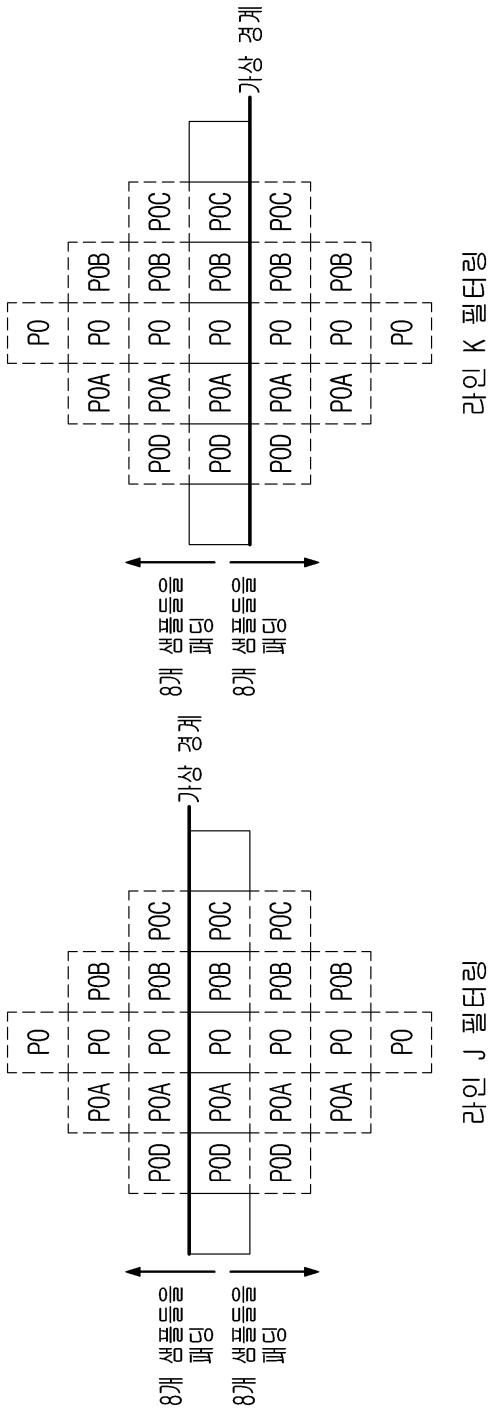
도면17a


라인 H 필터링
가상 경계
P0
P0
P3
P3
1개 샘플 패딩
1개 샘플 패딩
라인 M 필터링
가상 경계
P0
P0
P3
P3
1개 샘플 패딩
1개 샘플 패딩

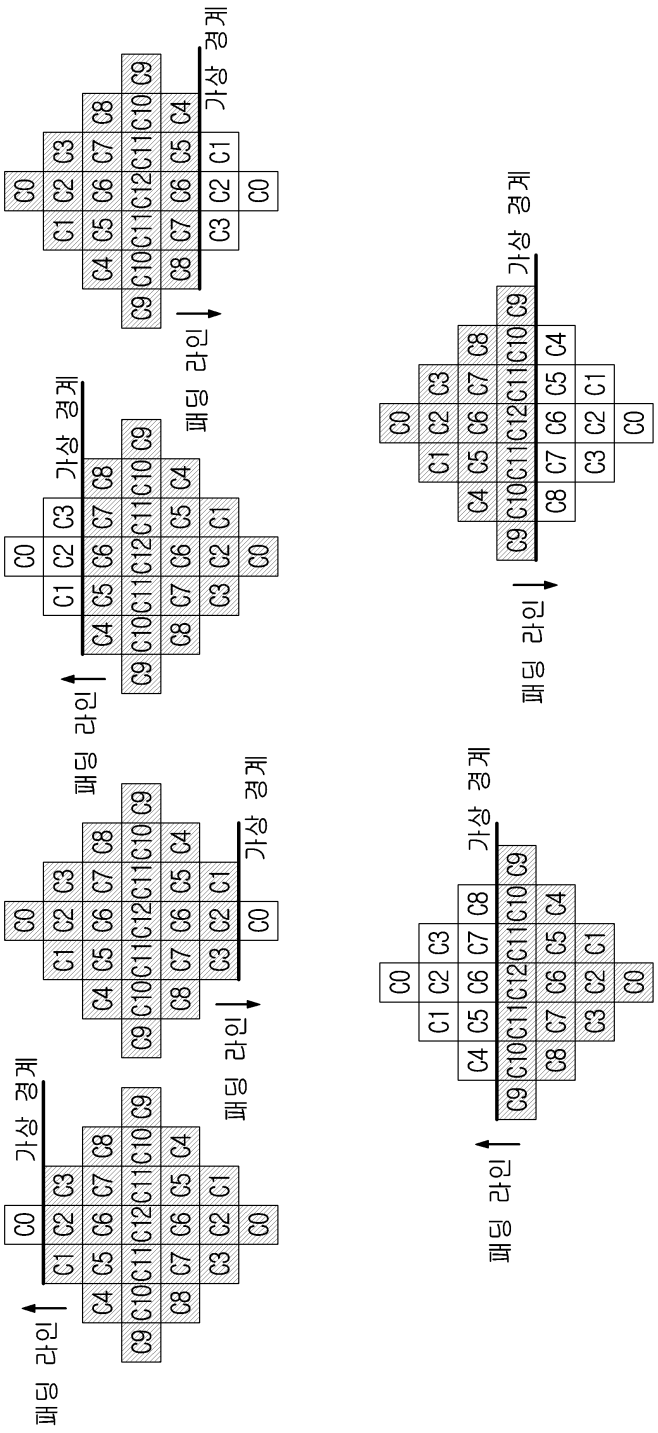
도면17b


라인 I 필터링
가상 경계
P0
P0A
P0
P0B
P3
P3A
P3
P3B
P3
4개 샘플들을 패딩
4개 샘플들을 패딩
라인 L 필터링
가상 경계
P0
P0A
P0
P0B
P3
P3A
P3
P3B
P3
4개 샘플들을 패딩
4개 샘플들을 패딩

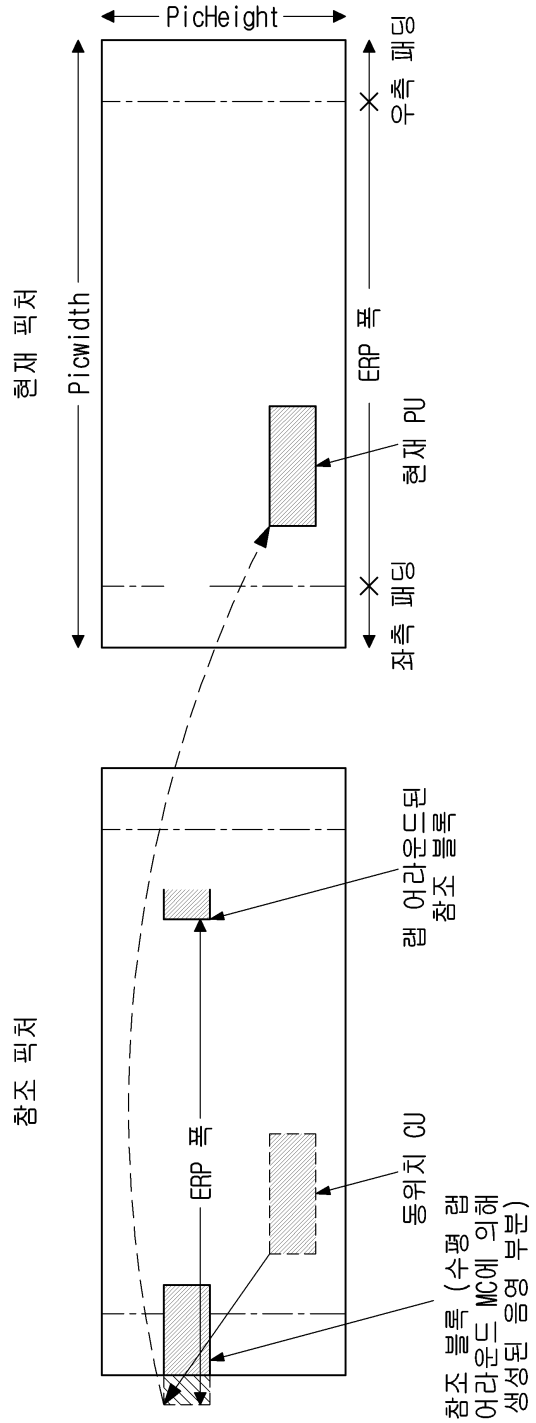
도면17c


가상 경계
8개 셀 폴더들을 편집
8개 셀 폴더들을 편집
라인 K 필터링
가상 경계
8개 셀 폴더들을 편집
8개 셀 폴더들을 편집
라인 J 필터링
P0
POA
POB
POC
POD

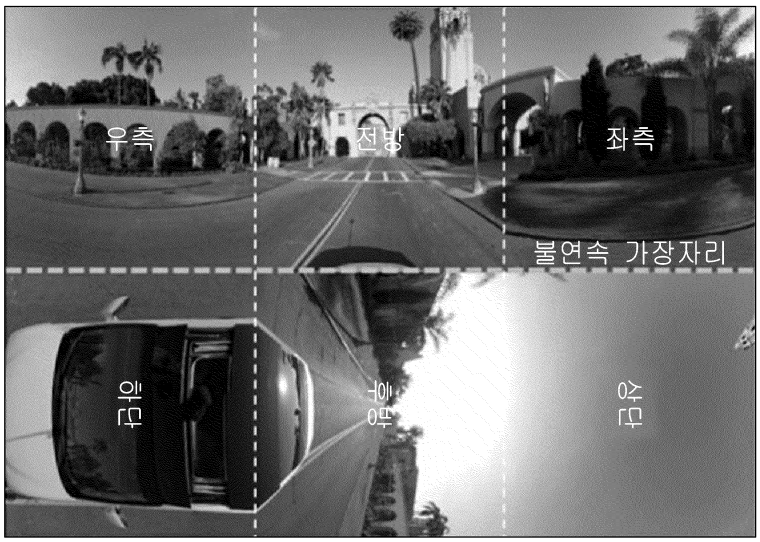
도면18



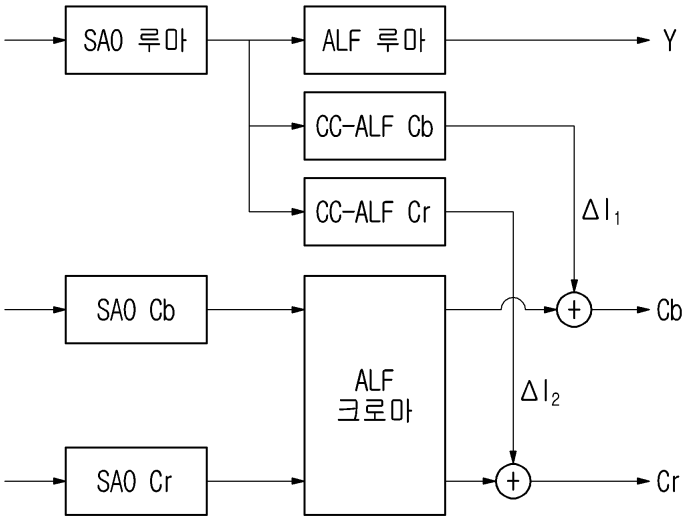
도면19


PicHeight
현재 픽처
Picwidth
우측 패딩
ERP 폭
현재 PU
좌측 패딩
참조 픽처
랩 어라운드된
참조 블록
ERP 폭
동위치 CU
참조 블록 (수평 랩
어라운드 MC에 의해
생성된 음영 부분)

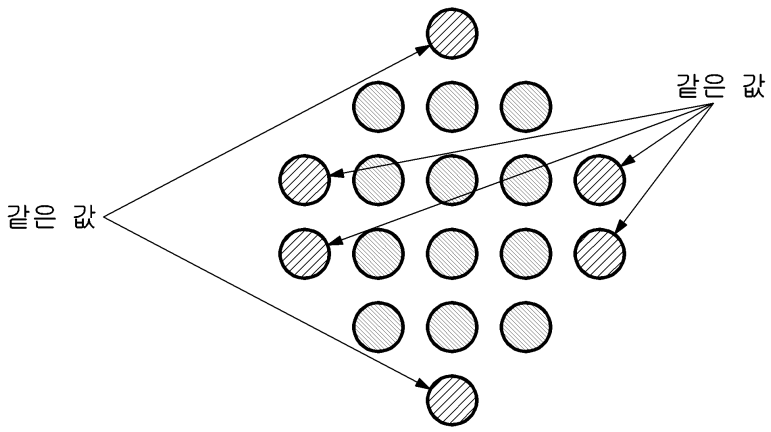
도면20



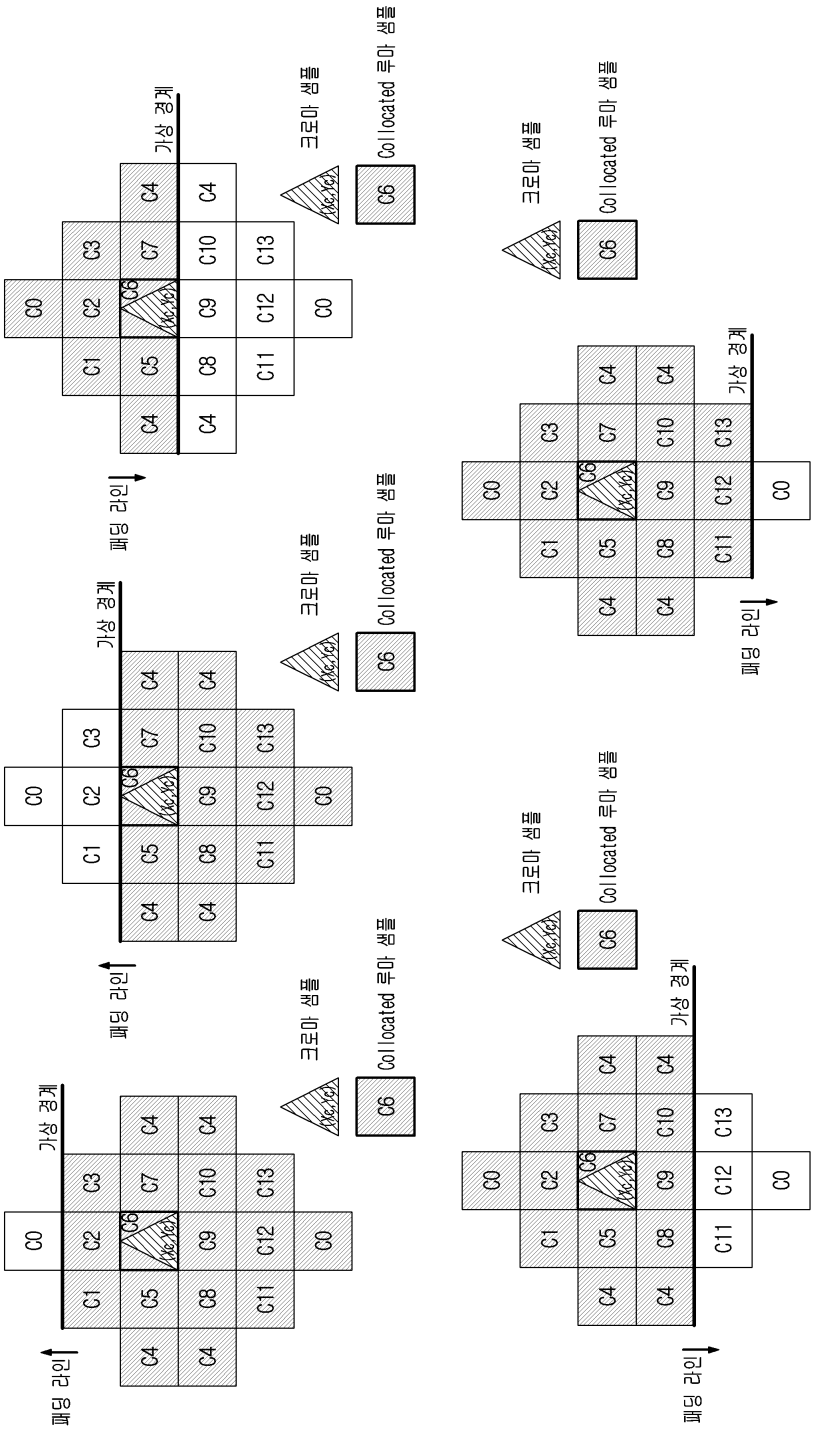
도면21a



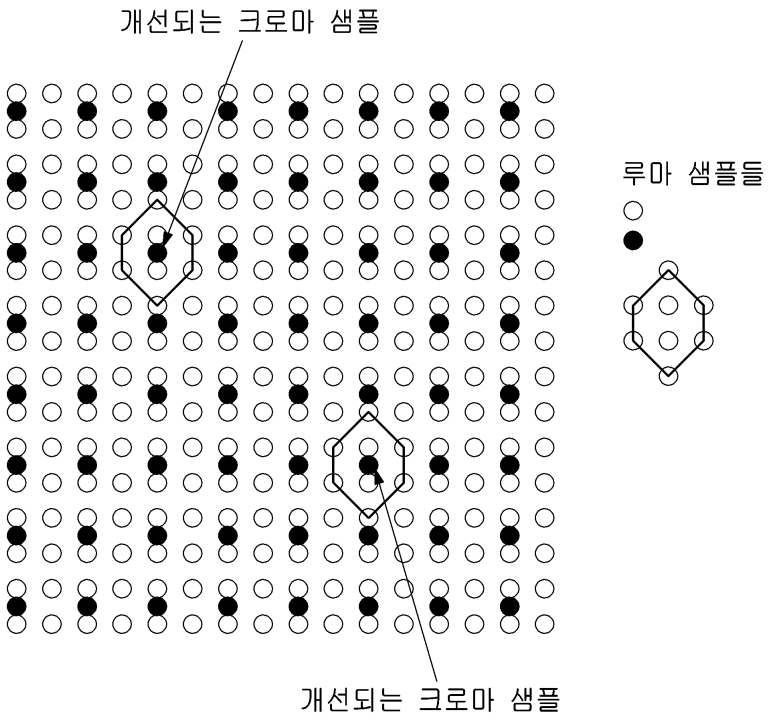
도면21b



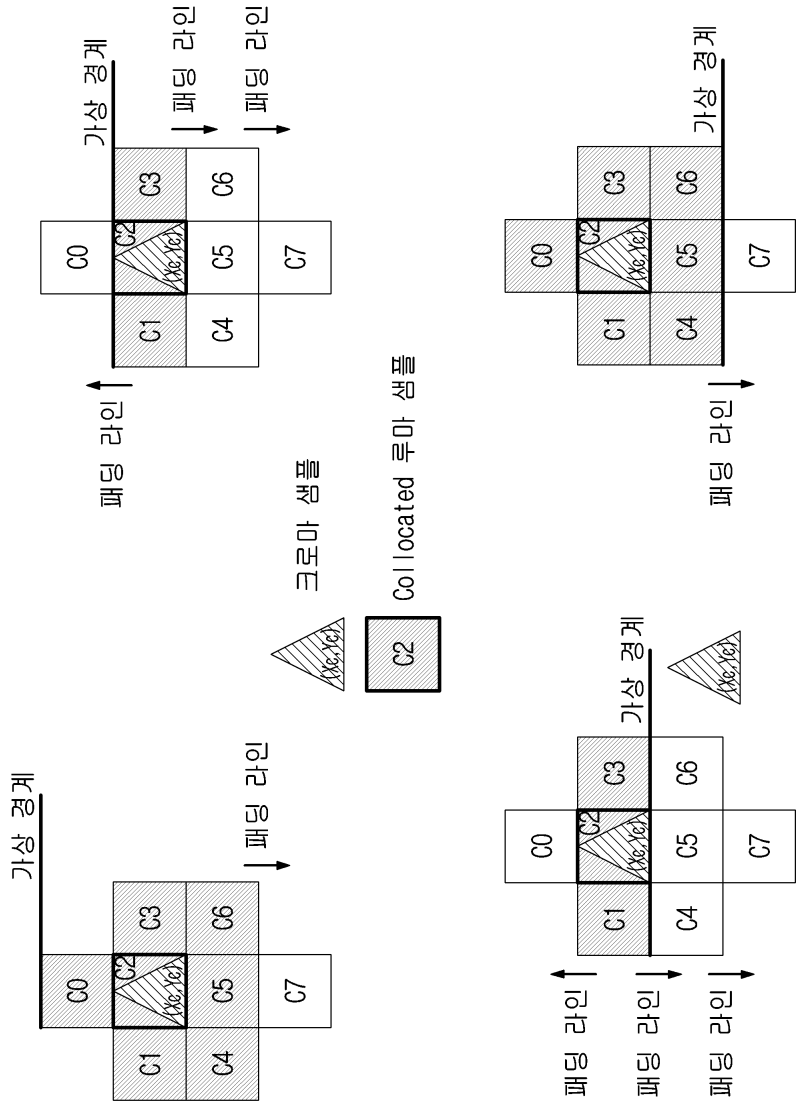
도면22


가상 경계
패딩 라인
크로마 샘플
Collocated 루마 샘플
C0
C1
C2
C3
C4
C5
C6
C7
C8
C9
C10
C11
C12
C13

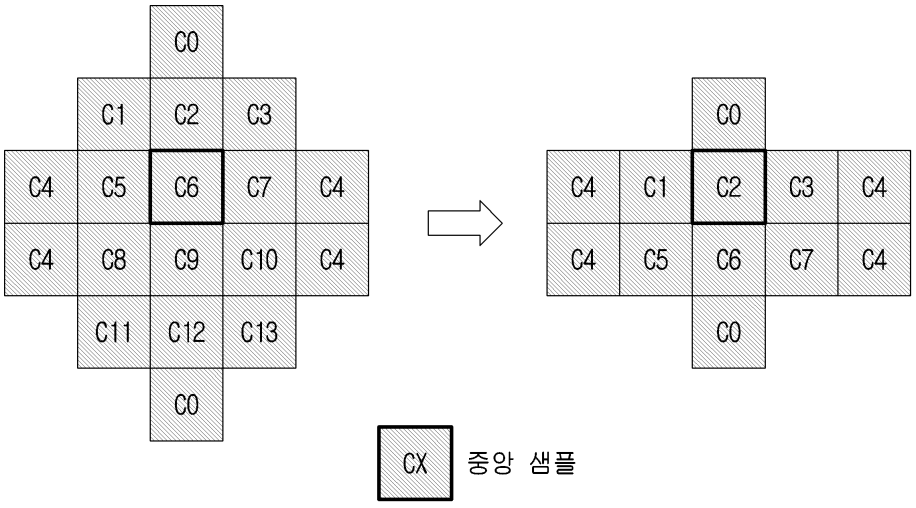
도면23



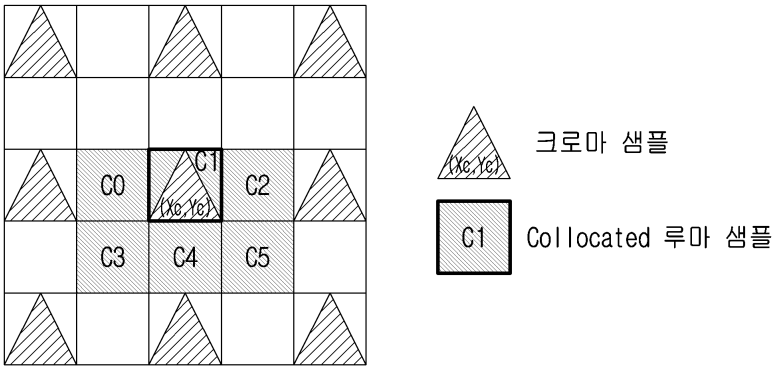
도면24


가상 경계
패딩 라인
패딩 라인
C0
C1
C2
C3
C4
C5
C6
C7
(xc,yc)
패딩 라인
크로마 샘플
Collocated 루마 샘플
C2
(xc,yc)
가상 경계
패딩 라인
가상 경계
패딩 라인
패딩 라인
패딩 라인

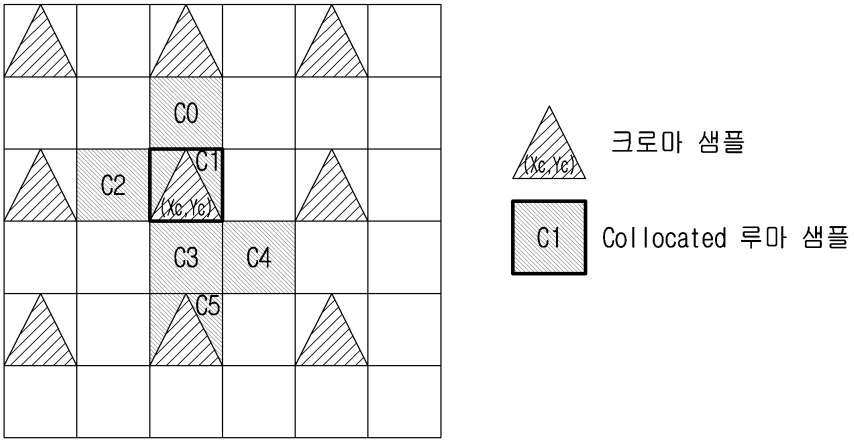
도면25


C0
C1
C2
C3
C4
C5
C6
C7
C8
C9
C10
C11
C12
C13
CX
중앙 샘플

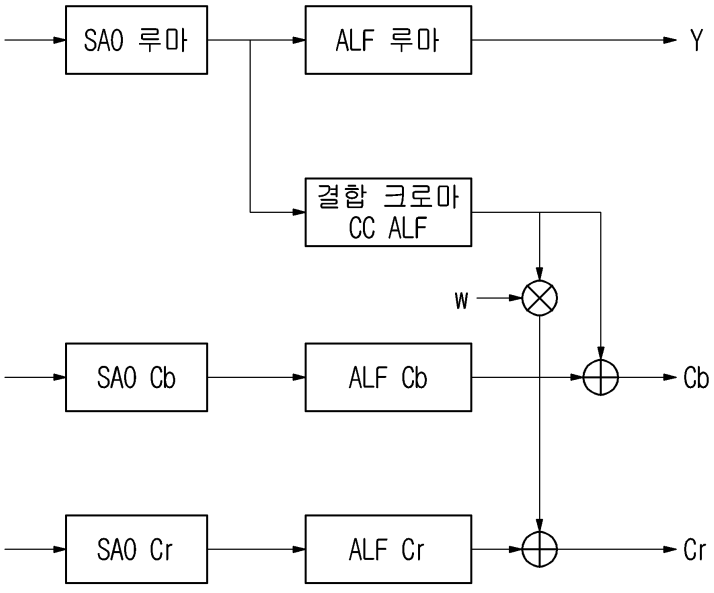
도면26



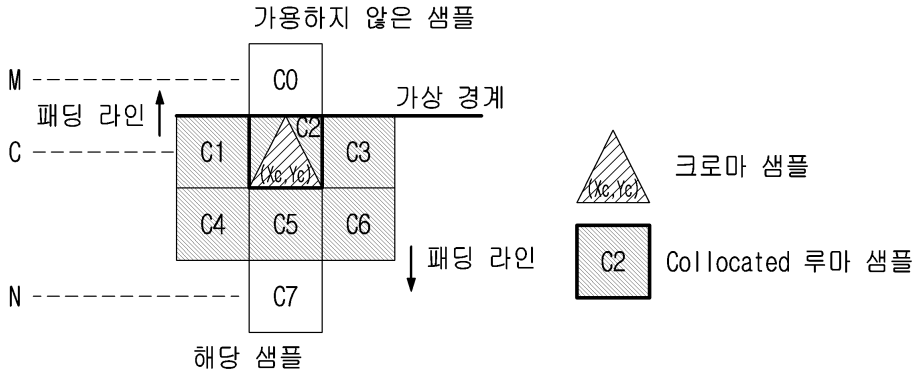
도면27



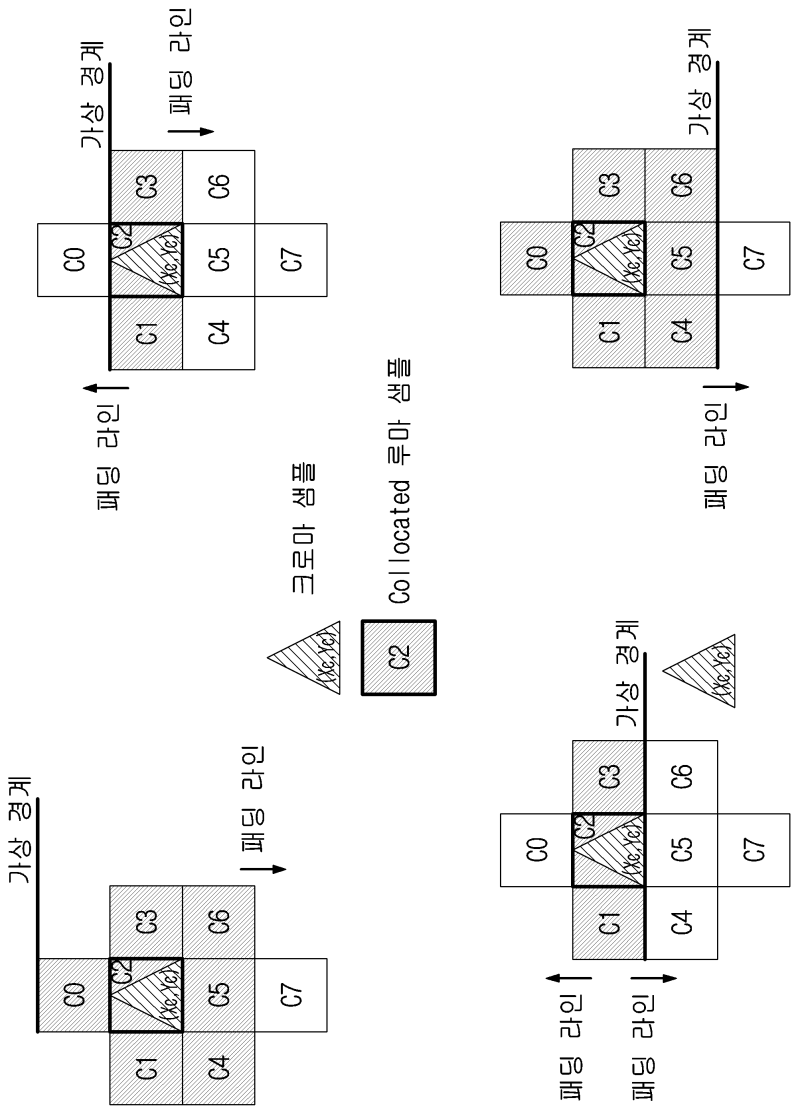
도면28



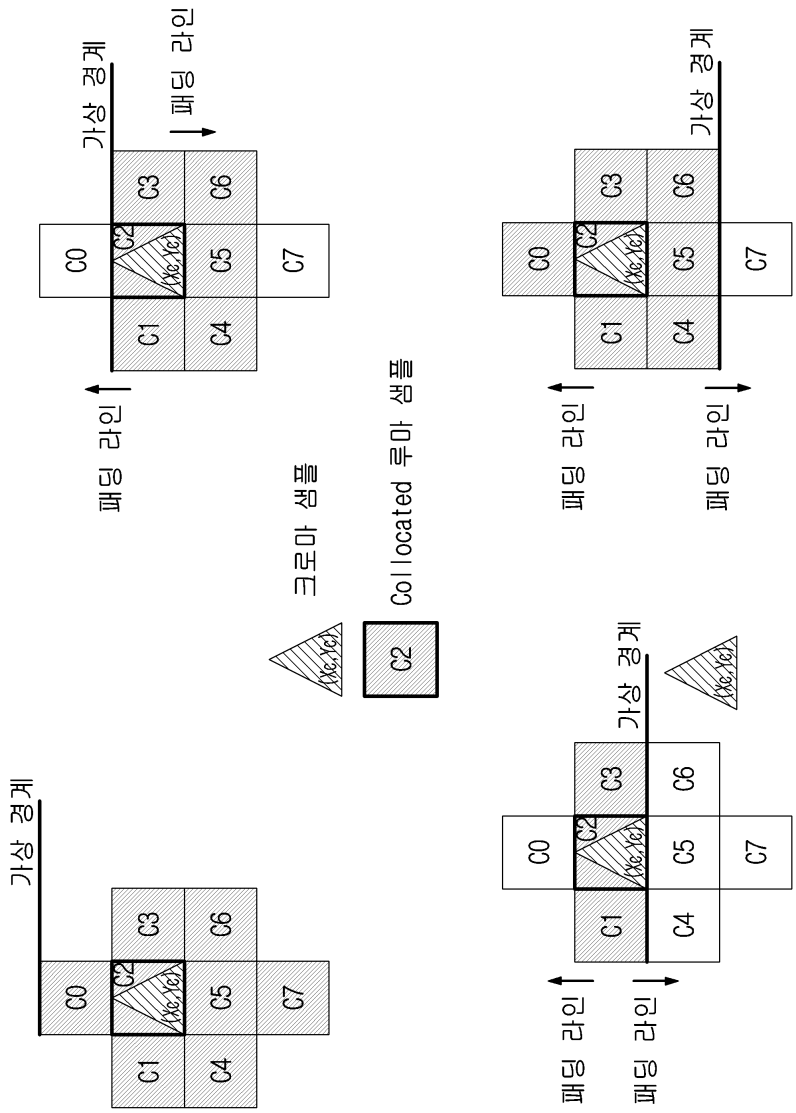
도면29


가용하지 않은 샘플
M
패딩 라인
C
N
C0
C1
C2
C3
C4
C5
C6
C7
(Xc,Yc)
가상 경계
패딩 라인
해당 샘플
크로마 샘플
C2
Collocated 루마 샘플

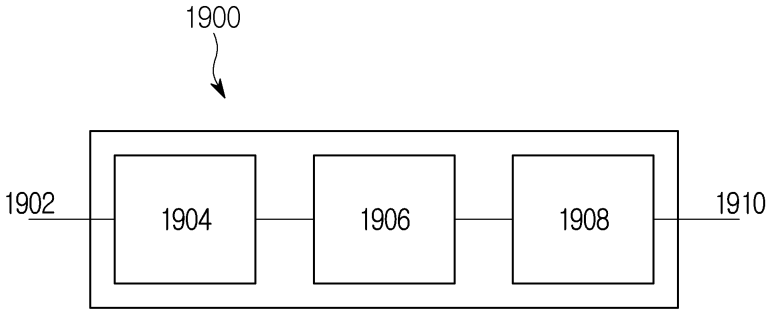
도면30


가상 경계
패딩 라인
C0
C1
C2
C3
C4
C5
C6
C7
(Xc,Yc)
크로마 샘플
Collocated 루마 샘플

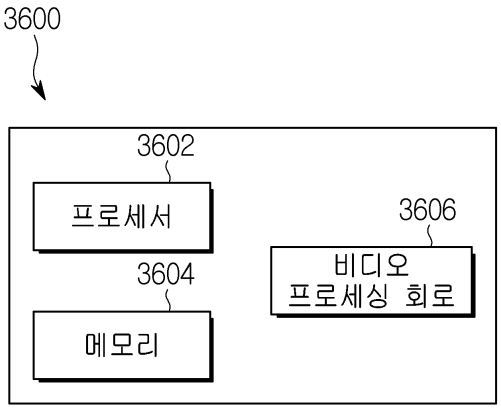
도면31


가상 경계
패딩 라인
C0
C1
C2
C3
C4
C5
C6
C7
(xc,yc)
패딩 라인
크로마 샘플
Collocated 루마 샘플
C2
가상 경계
패딩 라인
패딩 라인
가상 경계
패딩 라인
패딩 라인

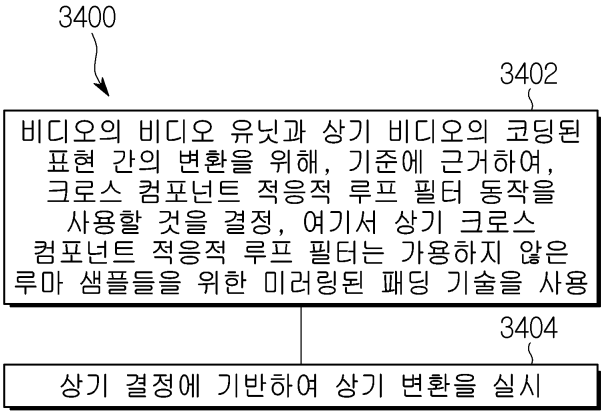
도면32


1900
1902
1904
1906
1908
1910

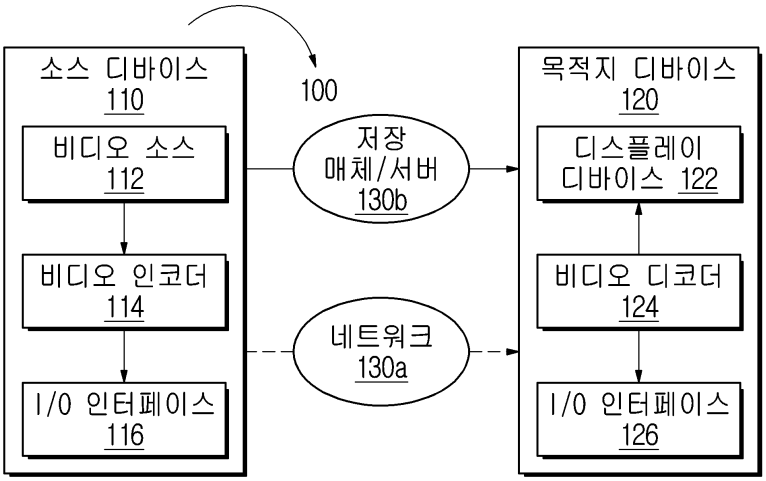
도면33



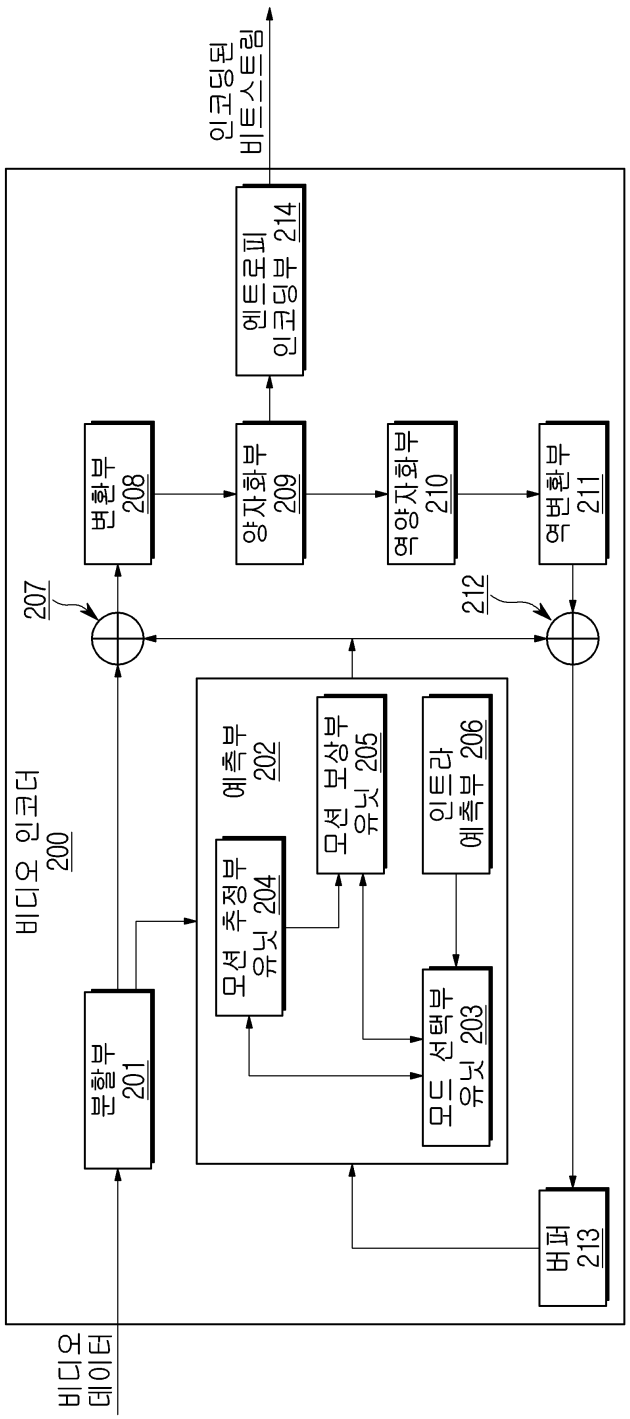
도면34



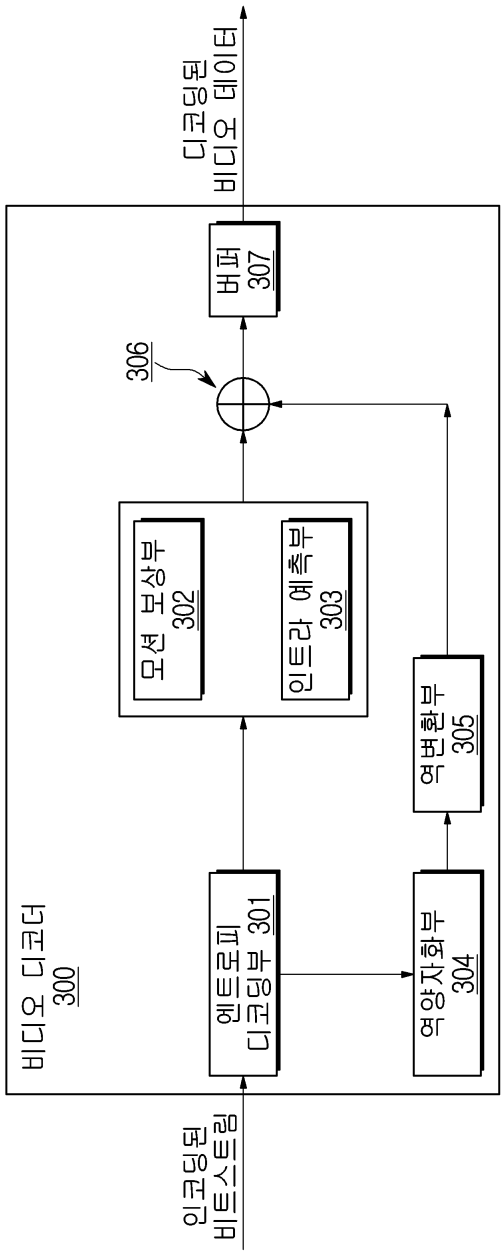
도면35



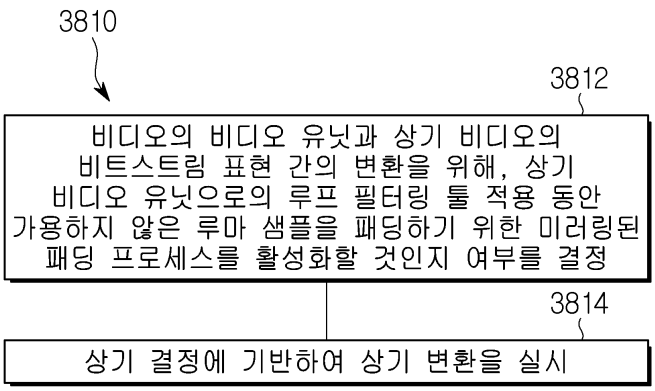
도면36


인코딩된
비트스트림
엔트로피
인코딩부 214
변환부
208
양자화부
209
역양자화부
210
역변환부
211
207
212
비디오 인코더
200
예측부
202
모션 보상부
유닛 205
인트라
예측부 206
모션 추정부
유닛 204
모드 선택부
유닛 203
분할부
201
버퍼
213
비디오
데이터

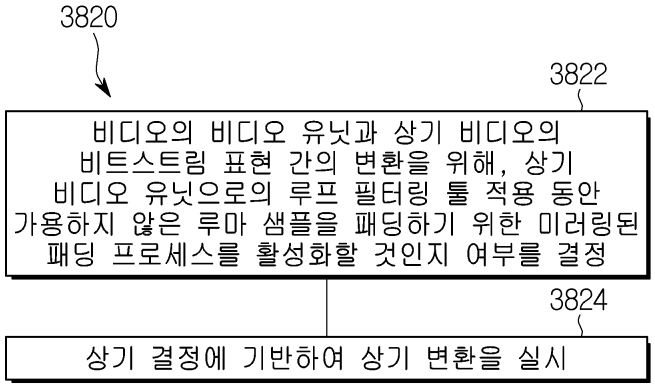
도면37


디코딩된
비디오 데이터
버퍼
307
306
모션 보상부
302
인트라 예측부
303
역변환부
305
비디오 디코더
300
엔트로피
디코딩부 301
역양자화부
304
인코딩된
비트스트림

도면38a


3810
3812
비디오의 비디오 유닛과 상기 비디오의
비트스트림 표현 간의 변환을 위해, 상기
비디오 유닛으로의 루프 필터링 툴 적용 동안
가용하지 않은 루마 샘플을 패딩하기 위한 미러링된
패딩 프로세스를 활성화할 것인지 여부를 결정
3814
상기 결정에 기반하여 상기 변환을 실시

도면38b


3820
3822
비디오의 비디오 유닛과 상기 비디오의 비트스트림 표현 간의 변환을 위해, 상기 비디오 유닛으로의 루프 필터링 툴 적용 동안 가용하지 않은 루마 샘플을 패딩하기 위한 미러링된 패딩 프로세스를 활성화할 것인지 여부를 결정
3824
상기 결정에 기반하여 상기 변환을 실시