(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2025년03월06일
(11) 등록번호 10-2777573
(24) 등록일자 2025년03월04일

(51) 국제특허분류(Int. Cl.)
H04N 19/70 (2014.01) *H04N 19/105* (2014.01)
H04N 19/117 (2014.01) *H04N 19/17* (2014.01)
H04N 19/577 (2014.01) *H04N 19/82* (2014.01)
(52) CPC특허분류
H04N 19/70 (2015.01)
H04N 19/105 (2015.01)
(21) 출원번호 10-2021-7027093
(22) 출원일자(국제) 2020년03월09일
심사청구일자 2023년02월20일
(85) 번역문제출일자 2021년08월25일
(65) 공개번호 10-2021-0135502
(43) 공개일자 2021년11월15일
(86) 국제출원번호 PCT/CN2020/078393
(87) 국제공개번호 WO 2020/182093
국제공개일자 2020년09월17일
(30) 우선권주장
PCT/CN2019/077429 2019년03월08일 중국(CN)
(56) 선행기술조사문헌
US20150124865 A1*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
두인 비전 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
장, 카이
미국 캘리포니아 90066 로스엔젤레스 웨스트 제퍼슨 블러바드 12655번지 6층 스위트 137호
장, 리
미국 캘리포니아 90066 로스엔젤레스 웨스트 제퍼슨 블러바드 12655번지 6층 스위트 137호
(뒷면에 계속)
(74) 대리인
특허법인 무한

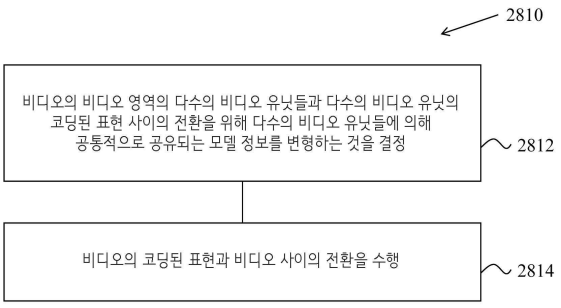
전체 청구항 수 : 총 18 항 심사관 : 최재륜

(54) 발명의 명칭 **비디오 처리에서 변형 정보의 시그널링**

(57) 요 약

하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 간의 전환을 수행하는 단계를 포함하고, 코딩된 표현은 하나 이상의 비디오 영역들 중 일부의 인-루프 변형(ILR)에 적용 가능한 변형 모델 정보를 포함하고, 변형 모델 정보는 제1 도메인 및 제2 도메인에서의 표현, 및/또는 채도 비디오 유닛의 채도 잔차의 스케일링에 기초하여 비디오 영역의 비디오 유닛의 재구성에 대한 정보를 제공하고, 변형 모델 정보는 대응하는 빈으로부터 절대 델타 코드워드 값을 명시하는 제2 신택스 요소를 나타내기 위해 이용되는 비트 수를 도출하는 제1 신택스 요소를 포함하는 파라미터 세트를 포함하고, 제1 신택스 요소는 임계값보다 작은 값을 가지는 비디오 처리 방법이 제공된다.

대 표 도 - 도28a



(52) CPC특허분류
H04N 19/117 (2015.01)
H04N 19/17 (2015.01)
H04N 19/577 (2015.01)
H04N 19/82 (2015.01)
(72) 발명자
리우, 홍빈
중국 베이징 100080 하이뎬 디스트릭트 지춘 로드 넘버 63 차이나 새틀라이트 커뮤니케이션 타워 진르터우탸오 우체국
왕, 위에
중국 베이징 100080 하이뎬 디스트릭트 지춘 로드 넘버 63 차이나 새틀라이트 커뮤니케이션 타워 진르터우탸오 우체국

명 세 서

청구범위

청구항 1

비디오의 비디오 영역의 현재 채도 비디오 블록과 상기 비디오의 비트스트림 간의 전환 동안, 스케일링 프로세스가 현재 채도 비디오 블록의 채도 잔차 샘플들에 적용되는지를 결정하는 단계; 및

적어도 하나의 스케일링 인자 varScale에 기초하여 상기 채도 잔차 샘플들에 대해 상기 스케일링 프로세스를 적용하여 상기 전환을 수행하는 단계

를 포함하고,

상기 스케일링 프로세스에서, 상기 채도 잔차 샘플들은 상기 현재 채도 비디오 블록을 재구성하기 위해 사용되기 전에 스케일링되고,

상기 적어도 하나의 스케일링 인자 varScale는 오프셋 기반 평균 연산(offset-based average operation)에 의해 특정 휘도 샘플들을 사용하여 도출된 변수 invAvgLuma에 기초하여 도출되고, 상기 변수 invAvgLuma는 상기 비디오의 색상 형식에 의존하여 계산되고, 상기 비디오의 상기 색상 형식은, 4:4:4 형식, 4:2:0 형식 또는 4:2:2 형식이고,

상기 invAvgLuma의 계산된 값은 상기 특정 휘도 샘플들을 결정하기 위해 사용되는 위치 변수들에 의존하고, 상기 위치 변수들의 값들은 상기 비디오의 상기 색상 형식을 사용하여 결정되고, 상기 위치 변수들의 다른 값들은, 4:4:4 형식, 4:2:0 형식 및 4:2:2 형식을 포함하는 다른 색상 형식들에 대응하고,

상기 스케일링 프로세스는 제1 조각별 선형 모델(piecewise linear model)에 기초하고, 인덱스 변수는 상기 변수 invAvgLuma가 속하는 조각을 식별하고, 상기 적어도 하나의 스케일링 인자 varScale는 상기 인덱스에 기초하여 도출되는,

비디오 데이터 처리 방법.

청구항 2

제1항에 있어서,

상기 비디오 영역의 휘도 비디오 블록에 대해,

1) 상기 휘도 비디오 블록의 예측 샘플들이 원본 도메인에서 변형된 도메인으로 전환되는, 상기 휘도 비디오 블록에 대한 순방향 매핑 프로세스(forward mapping process); 또는

2) 상기 변형된 도메인의 상기 휘도 비디오 블록의 재구성된 샘플들이 상기 원본 도메인으로 전환되는, 상기 순방향 매핑 프로세스의 역 연산(inverse operation)인 역 매핑 프로세스(inverse mapping process)

중 적어도 하나가 수행되는,

비디오 데이터 처리 방법.

청구항 3

제2항에 있어서,

제2 조각별 선형 모델은,

상기 순방향 매핑 프로세스 동안 상기 휘도 비디오 블록의 상기 예측 샘플들을 특정 값들로 매핑하기 위해 사용되는,

비디오 데이터 처리 방법.

청구항 4

제2항에 있어서,

상기 역 매핑 프로세스에서 생성된 상기 원본 도메인의 상기 휘도 비디오 블록의 전환된 재구성된 샘플들에 필터링 프로세스가 적용되는,

비디오 데이터 처리 방법.

청구항 5

제3항에 있어서,

상기 제2 조각별 선형 모델에 대한 제1 신택스 요소 및 제2 신택스 요소가 상기 비트스트림에 포함되고,

상기 제1 신택스 요소에 1을 더한 값은 상기 제2 신택스 요소의 표현에 사용되는 비트 수를 명시하고,

상기 제2 신택스 요소는 상기 특정 값들과 연관된 i번째 빈(bin)에 대한 절대 델타 코드 워드 값(absolute delta code word value)을 명시하는,

비디오 데이터 처리 방법.

청구항 6

제5항에 있어서,

상기 제1 신택스 요소의 값은,

임계값보다 작은,

비디오 데이터 처리 방법.

청구항 7

제6항에 있어서,

상기 임계값은,

비트 깊이(bit depth)에 의존하는,

비디오 데이터 처리 방법.

청구항 8

제1항에 있어서,

상기 제1 조각별 선형 모델은 변형된 도메인에서 상기 비디오 영역의 휘도 비디오 블록의 재구성된 샘플들을 원본 도메인으로 전환하는 순방향 매핑 프로세스의 역 연산인 역 매핑 프로세스에 사용되고,

상기 휘도 비디오 블록에 대한 상기 순방향 매핑 프로세스에서, 상기 휘도 비디오 블록의 예측 샘플들은 상기 원본 도메인에서 상기 변형된 도메인으로 전환되는,

비디오 데이터 처리 방법.

청구항 9

제2항에 있어서,

제3 신택스 요소 및 제4 신택스 요소는 상기 비트스트림에 포함되고,

상기 제3 신택스 요소는 최대 허용 빈 인덱스(maximum allowed bin index)와 상기 순방향 매핑 프로세스에서 사용되는 최대 빈 인덱스(maximum bin index) 간의 차이를 명시하고,

상기 제4 신택스 요소는 상기 순방향 매핑 프로세스에서 사용되는 최소 빈 인덱스를 명시하는,

비디오 데이터 처리 방법.

청구항 10

제9항에 있어서,

상기 제3 신택스 요소의 값은,

0부터 상기 최대 허용 빈 인덱스까지의 범위 내에 있는,

비디오 데이터 처리 방법.

청구항 11

제9항에 있어서,

상기 제4 신택스 요소의 값은,

상기 순방향 매핑 프로세스에서 사용되는 상기 최대 빈 인덱스보다 작거나 같은,

비디오 데이터 처리 방법.

청구항 12

제9항에 있어서,

상기 제4 신택스 요소의 값은,

0부터 상기 순방향 매핑 프로세스에서 사용되는 상기 최대 빈 인덱스까지의 범위 내에 있는,

비디오 데이터 처리 방법.

청구항 13

제9항에 있어서,

상기 최대 허용 빈 인덱스는,

15와 같은,

비디오 데이터 처리 방법.

청구항 14

제1항에 있어서,

상기 전환은,

상기 현재 채도 비디오 블록을 상기 비트스트림으로 인코딩하는 것을 포함하는,

비디오 데이터 처리 방법.

청구항 15

제1항에 있어서,

상기 전환은,

비트스트림으로부터 상기 현재 채도 비디오 블록을 디코딩하는 것을 포함하는,

비디오 데이터 처리 방법.

청구항 16

프로세서 및 명령어들이 있는 비일시적 메모리를 포함하는 비디오 데이터 처리 장치에 있어서, 상기 프로세서에 의한 실행 시 상기 명령어들은 상기 프로세서가,

비디오의 비디오 영역의 현재 채도 비디오 블록과 상기 비디오의 비트스트림 간의 전환 동안, 스케일링 프로세스가 현재 채도 비디오 블록의 채도 잔차 샘플들에 적용되는지를 결정하고,

적어도 하나의 스케일링 인자 varScale에 기초하여 상기 채도 잔차 샘플들에 대해 상기 스케일링 프로세스를 적

용하여 상기 전환을 수행하도록 하고,

상기 스케일링 프로세스에서, 상기 채도 잔차 샘플들은 상기 현재 채도 비디오 블록을 재구성하기 위해 사용되기 전에 스케일링되고,

상기 적어도 하나의 스케일링 인자 varScale는 오프셋 기반 평균 연산에 의해 특정 휘도 샘플들을 사용하여 도출된 변수 invAvgLuma에 기초하여 도출되고, 상기 변수 invAvgLuma는 상기 비디오의 색상 형식에 의존하여 계산되고, 상기 비디오의 상기 색상 형식은, 4:4:4 형식, 4:2:0 형식 또는 4:2:2 형식이고,

상기 invAvgLuma의 계산된 값은 상기 특정 휘도 샘플들을 결정하기 위해 사용되는 위치 변수들에 의존하고, 상기 위치 변수들의 값들은 상기 비디오의 상기 색상 형식을 사용하여 결정되고, 상기 위치 변수들의 다른 값들은 4:4:4 형식, 4:2:0 형식 및 4:2:2 형식을 포함하는 다른 색상 형식들에 대응하고,

상기 스케일링 프로세스는 제1 조각별 선형 모델에 기초하고, 인덱스 변수는 상기 변수 invAvgLuma가 속하는 조각을 식별하고, 상기 적어도 하나의 스케일링 인자 varScale는 상기 인덱스에 기초하여 도출되는,

비디오 데이터 처리 장치.

청구항 17

프로세서가

비디오의 비디오 영역의 현재 채도 비디오 블록과 상기 비디오의 비트스트림 간의 전환 동안, 스케일링 프로세스가 현재 채도 비디오 블록의 채도 잔차 샘플들에 적용되는지를 결정하고,

적어도 하나의 스케일링 인자 varScale에 기초하여 상기 채도 잔차 샘플들에 대해 상기 스케일링 프로세스를 적용하여 상기 전환을 수행하도록 하는 명령어들을 저장하고,

상기 스케일링 프로세스에서, 상기 채도 잔차 샘플들은 상기 현재 채도 비디오 블록을 재구성하기 위해 사용되기 전에 스케일링되고,

상기 적어도 하나의 스케일링 인자 varScale는 오프셋 기반 평균 연산에 의해 특정 휘도 샘플들을 사용하여 도출된 변수 invAvgLuma에 기초하여 도출되고, 상기 변수 invAvgLuma는 상기 비디오의 색상 형식에 의존하여 계산되고, 상기 비디오의 상기 색상 형식은, 4:4:4 형식, 4:2:0 형식 또는 4:2:2 형식이고,

상기 invAvgLuma의 계산된 값은 상기 특정 휘도 샘플들을 결정하기 위해 사용되는 위치 변수들에 의존하고, 상기 위치 변수들의 값들은 상기 비디오의 상기 색상 형식을 사용하여 결정되고, 상기 위치 변수들의 다른 값들은 4:4:4 형식, 4:2:0 형식 및 4:2:2 형식을 포함하는 다른 색상 형식들에 대응하고,

상기 스케일링 프로세스는 제1 조각별 선형 모델에 기초하고, 인덱스 변수는 상기 변수 invAvgLuma가 속하는 조각을 식별하고, 상기 적어도 하나의 스케일링 인자 varScale는 상기 인덱스에 기초하여 도출되는,

비일시적 컴퓨터 판독 가능 저장 매체.

청구항 18

비디오 처리 장치에 의해 수행되는 방법에 의해 생성된 비디오의 비트스트림을 저장하는 비일시적 컴퓨터 판독 가능 기록 매체에 있어서, 상기 방법은,

비디오의 비디오 영역의 현재 채도 비디오 블록의 채도 잔차 샘플들에 스케일링 프로세스가 적용되는지를 결정하는 단계; 및

적어도 하나의 스케일링 인자 varScale에 기초하여 상기 채도 잔차 샘플들에 상기 스케일링 프로세스를 적용하여 상기 비트스트림을 생성하는 단계를 포함하고,

상기 스케일링 프로세스에서, 상기 채도 잔차 샘플들은 상기 현재 채도 비디오 블록을 재구성하기 위해 사용되기 전에 스케일링되고,

상기 적어도 하나의 스케일링 인자 varScale는 오프셋 기반 평균 연산에 의해 특정 휘도 샘플들을 사용하여 도출된 변수 invAvgLuma에 기초하여 도출되고, 상기 변수 invAvgLuma는 상기 비디오의 색상 형식에 의존하여 계산되고, 상기 비디오의 상기 색상 형식은, 4:4:4 형식, 4:2:0 형식 또는 4:2:2 형식이고,

상기 invAvgLuma의 계산된 값은 상기 특정 휘도 샘플들을 결정하기 위해 사용되는 위치 변수들에 의존하고, 상기 위치 변수들의 값들은 상기 비디오의 상기 색상 형식을 사용하여 결정되고, 상기 위치 변수들의 다른 값들은 4:4:4 형식, 4:2:0 형식 및 4:2:2 형식을 포함하는 다른 색상 형식들에 대응하고,

상기 스케일링 프로세스는 제1 조각별 선형 모델에 기초하고, 인덱스 변수는 상기 변수 invAvgLuma가 속하는 조각을 식별하고, 상기 적어도 하나의 스케일링 인자 varScale는 상기 인덱스에 기초하여 도출되는,

비일시적 컴퓨터 판독 가능 기록 매체.

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

발명의 설명

기 술 분 야

본 출원은 2020년 3월 9일에 출원된 국제 특허 출원 PCT/CN2020/078393에 기초하고, 2019년 3월 8일에 출원된 국제 특허 출원 번호 PCT/CN2019/077429에 대한 우선권 및 이익을 주장한다. 전술한 모든 특허 출원은 그 전체가 참고로 본 명세서에 포함된다.

이 특허 문서는 비디오 코딩 기술, 장치 및 시스템에 관한 것이다.

배 경 기 술

현재, 더 나은 압축비들(compression ratios)을 제공하거나 더 낮은 복잡성 또는 병렬화된 실행들(parallelized implementations)을 허용하는 비디오 코딩(coding) 및 디코딩(decoding) 방식들(schemes)을 제공하기 위해 현재(current) 비디오 코덱 기술들(video codec technologies)의 성능을 개선하려는 노력들이 진행 중(underwa

y)이다. 업계 전문가들은 최근 몇 가지 새로운 비디오 코딩 도구들을 제안했고, 그 효과성(effectivity)을 결정하기 위한 테스트가 현재 진행 중이다.

발명의 내용

과제의 해결 수단

디지털 비디오 코딩, 특히 양자화(quantization) 단계 시그널링(signaling) 및 비디오 코딩에서 다른 도구들과의 블록 기반 인-루프 변형(in-loop reshaping)의 상호작용(interaction)들과 관련된 장치, 시스템 및 방법. HEVC와 같은 기존의(existing) 비디오 코딩 표준(video coding standard)에 적용되거나, 완성될 표준(다목적 비디오 코딩(Versatile Video Coding))에 적용될 수 있다. 미래의 비디오 코딩 표준이나 비디오 코덱에도 적용 가능(applicable)하다.

하나의 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하기 위해 사용될 수 있다. 이 방법은 다음을 포함한다: 비디오의 비디오 영역(video region)의 다수의 비디오 유닛들(multiple video units)과 다수의 비디오 유닛들의 코딩된 표현(coded representation) 간의 전환을 위해, 다수의 비디오 유닛들에 의해 공통적으로 공유되는 변형 모델 정보(reshaping model information)를 결정하는 단계; 및 비디오의 코딩된 표현과 비디오 간의 전환(conversion)을 수행하는 단계. 변형 모델 정보는 제1 도메인(domain) 및 제2 도메인에서 비디오 샘플들을 구성하고/하거나 채도 비디오 유닛(chroma video unit)의 채도 잔차(chroma residue)를 스케일링(scaling)하기 위한 정보를 제공한다.

다른 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하기 위해 사용될 수 있다. 방법은 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 간의 전환을 수행하기 위해, 비디오의 비트-깊이의 함수(function of a bit-depth)로서 변형 모델 정보의 변수 값(a value of a variable)을 결정하는 단계, 및 결정에 기초하여 전환을 수행하는 단계를 포함하고, 변형 정보는 하나 이상의 비디오 영역들 중 일부의 인-루프 변형(ILR)에 적용 가능하고, 변형 정보는 제1 도메인 및 제2 도메인에서의 표현, 및/또는 채도 비디오 유닛의 채도 잔차의 스케일링에 기초하여 비디오 영역의 비디오 유닛의 재구성(reconstruction)에 대한 정보를 제공한다.

다른 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하기 위해 사용될 수 있다. 방법은 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 간의 전환을 수행하는 단계를 포함하고, 코딩된 표현은 하나 이상의 비디오 영역들 중 일부의 인-루프 변형(ILR)에 적용 가능한 변형 모델 정보를 포함하고, 변형 모델 정보는 제1 도메인 및 제2 도메인에서의 표현, 및/또는 채도 비디오 유닛의 채도 잔차의 스케일링에 기초하여 비디오 영역의 비디오 유닛의 재구성에 대한 정보를 제공하고, 변형 모델 정보는 초기화 규칙(initialization rule)에 기초하여 초기화된다.

다른 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하기 위해 사용될 수 있다. 방법은 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 간의 전환을 위해, 인 루프 변형(ILR)을 활성화(enable)할지 또는 비활성화(disable)할지를 결정하는 단계; 및 결정에 기초하여 전환을 수행하는 단계를 포함하고, 코딩된 표현은 하나 이상의 비디오 영역들 중 일부의ILR에 적용 가능한 변형 모델 정보를 포함하고, 변형 모델 정보는 제1 도메인 및 제2 도메인에서의 표현, 및/또는 채도 비디오 유닛의 채도 잔차의 스케일링에 기초하여 비디오 영역의 비디오 유닛의 재구성에 대한 정보를 제공하고, 결정하는 단계는, 변형 모델 정보가 초기화되지 않은 경우, ILR을 비활성화하도록 결정한다.

다른 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하기 위해 사용될 수 있다. 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 간의 전환을 수행하는 단계를 포함하고, 코딩된 표현은 하나 이상의 비디오 영역들 중 일부의 인 루프 변형(ILR)에 적용 가능한 변형 모델 정보를 포함하고, 변형 모델 정보는 제1 도메인 및 제2 도메인에서의 표현, 및/또는 채도 비디오 유닛의 채도 잔차의 스케일링에 기초하여 비디오 영역의 비디오 유닛의 재구성에 대한 정보를 제공하고, 변형 모델 정보는 비디오 영역이 특정 코딩 유형을 사용하여 코딩되는 경우에만 코딩된 표현에 포함된다.

다른 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하기 위해 사용될 수 있다. 방법은 비디오의 제1 비디오 영역과 제1 비디오 영역의 코딩된 표현 간의 전환을 위해, 제2 비디오 영역으로부터의 변형 정보가 규칙에 기초하여 전환에 사용 가능한지 여부를 결정하는 단계; 및 결정에 따라 전환을 수행하는 단계를 포함한다.

다른 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하기 위해 사용될 수 있다. 방법은 현재 비디오 영역이 인트라 코딩(intra coding)을 사용하여 코딩되도록 비디오의 비디오 영역과 비디오 영역이 코딩된 표현 사이의 전환을 수행하는 단계를 포함하고, 코딩된 표현은 비디오 영역 레벨(video region level)에서 코딩된 표현의 플래그 값(value of a flag)에 기초하여 조건부로(conditionally) 코딩된 표현의 변형 모델 정보를 명시하는(specifying) 형식 규칙(format rule)에 따른다.

다른 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하기 위해 사용될 수 있다. 방법은 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 간의 전환을 수행하는 단계를 포함하고, 코딩된 표현은 하나 이상의 비디오 영역들 중 일부의 인-루프 변형(ILR)에 적용 가능한 변형 모델 정보를 포함하고, 변형 모델 정보는 제1 도메인 및 제2 도메인에서의 표현, 및/또는 채도 비디오 유닛의 채도 잔차의 스케일링에 기초하여 비디오 영역의 비디오 유닛의 재구성에 대한 정보를 제공하고 및 변형 모델 정보는 허용된(allowed) 최대 빈 인덱스(maximum bin index)와 재구성에 사용될 최대 빈 인덱스 간의 차이를 명시하는 신택스 요소(syntax element)를 포함하는 파라미터 세트(parameter set)를 포함하고, 파라미터는 범위 내에 있다.

다른 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하기 위해 사용될 수 있다. 방법은 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 간의 전환을 수행하는 단계를 포함하고, 코딩된 표현은 하나 이상의 비디오 영역들 중 일부의 인-루프 변형(ILR)에 적용 가능한 변형 모델 정보를 포함하고, 변형 모델 정보는 제1 도메인 및 제2 도메인에서의 표현, 및/또는 채도 비디오 유닛의 채도 잔차의 스케일링에 기초하여 비디오 영역의 비디오 유닛의 재구성에 대한 정보를 제공하고, 및 변형 모델 정보는 재구성에 사용될 최대 빈 인덱스를 포함하는 파라미터 세트를 포함하고, 최대 빈 인덱스는 재구성에 사용될 최소 빈 인덱스 및 무부호 정수(unsigned integer)이고, 최소 빈 인덱스 이후에 시그널링되는 신택스 요소의 합과 동일한 제1 값으로 도출(derived)된다.

다른 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하기 위해 사용될 수 있다. 방법은 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 간의 전환을 수행하는 단계를 포함하고, 코딩된 표현은 하나 이상의 비디오 영역들 중 일부의 인-루프 변형(ILR)에 적용 가능한 변형 모델 정보를 포함하고, 변형 모델 정보는 제1 도메인 및 제2 도메인에서의 표현, 및/또는 채도 비디오 유닛의 채도 잔차의 스케일링에 기초하여 비디오 영역의 비디오 유닛의 재구성에 대한 정보를 제공하고, 변형 모델 정보는 대응하는 빈으로부터 절대 델타 코드워드 값(absolute delta codeword value)을 명시하는 제2 신택스 요소를 나타내기 위해 이용되는 다수의(a number of) 비트들을 도출하는 제1 신택스 요소를 포함하는 파라미터 세트를 포함하고, 제1 신택스 요소는 임계값(threshold)보다 작은 값을 갖는다.

다른 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하기 위해 사용될 수 있다. 방법은 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 간의 전환을 수행하는 단계를 포함하고, 코딩된 표현은 하나 이상의 비디오 영역들 중 일부의 인-루프 변형(ILR)에 적용 가능한 변형 모델 정보를 포함하고, 변형 모델 정보는 제1 도메인 및 제2 도메인에서의 표현, 및/또는 채도 비디오 유닛의 채도 잔차의 스케일링에 기초하여 비디오 영역의 비디오 유닛의 재구성에 대한 정보를 제공하고, 변형 모델 정보는 ILR에서 사용되는 i번째 빈의 기울기(slope)를 나타내고, (i-1)번째 파라미터에 기초한 값을 가지는 i번째 파라미터를 포함하는 파라미터 세트를 포함하고, i는 양의 정수(positive integer)이다.

다른 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하기 위해 사용될 수 있다. 방법은 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 간의 전환을 수행하는 단계를 포함하고, 코딩된 표현은 하나 이상의 비디오 영역들 중 일부의 인-루프 변형(ILR)에 적용 가능한 변형 모델 정보를 포함하고, 변형 모델 정보는 제1 도메인 및 제2 도메인에서의 표현, 및/또는 채도 비디오 유닛의 채도 잔차의 스케일링에 기초하여 비디오 영역의 비디오 유닛의 재구성에 대한 정보를 제공하고, ILR을 위해 사용되는 변형 모델 정보는 시그널링되지 않은 reshape_model_bin_delta_sign_CW [i]을 포함하는 파라미터 세트를 포함하고, reshape_model_bin_delta_sign_CW [i]는 항상 양수이다.

다른 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하기 위해 사용될 수 있다. 방법은 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 간의 전환을 수행하는 단계를 포함하고, 코딩된 표현은 하나 이상의 비디오 영역들 중 일부의 인-루프 변형(ILR)에 적용 가능한 변형 모델 정보를 포함하고, 변형 모델 정보는 제1 도메인 및 제2 도메인에서의 표현, 및/또는 채도 비디오 유닛의 채도 잔차의 스케일링에 기초하여 비디오 영역의 비디오 유닛의 재구성에 대한 정보를 제공하고, 변형 모델 정보는 비디오 영역의 색상 형식(color format)에 따른 스케일링을 위한 휘도 값들(luma values)을 사용하기 위한 파라미터

invAvgLuma를 포함하는 파라미터 세트를 포함한다.

다른 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하기 위해 사용될 수 있다. 방법은 비디오의 현재 비디오 블록과 비디오의 코딩된 표현 간의 전환을 수행하는 단계를 포함하고, 전환은 재구성된 영상 휘도 샘플들을 수정된 재구성된 영상 휘도 샘플들로 변환(transform)하기 위한 영상 역 매핑 프로세스(picture inverse mapping process)를 포함하고, 영상 역 매핑 프로세스는 상한(upper bound) 및 하한(lower bound)이 별도로 설정되는 클리핑(clipping)을 포함한다.

다른 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하기 위해 사용될 수 있다. 방법은 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 간의 전환을 수행하는 단계를 포함하고, 코딩된 표현은 하나 이상의 비디오 영역들 중 일부의 인-루프 변형(ILR)에 적용 가능한 변형 모델 정보를 포함하고, 변형 모델 정보는 제1 도메인 및 제2 도메인에서의 표현, 및/또는 채도 비디오 유닛의 채도 잔차의 스케일링에 기초하여 비디오 영역의 비디오 유닛의 재구성에 대한 정보를 제공하고, 및 변형 모델 정보는 Pivot[i]<=T가 되도록 제한되는 피벗 수량(pivot quantity)을 포함하는 파라미터 세트를 포함한다.

다른 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하기 위해 사용될 수 있다. 방법은 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 간의 전환을 수행하는 단계를 포함하고, 코딩된 표현은 인 루프 변형(ILR)에 적용 가능한 정보 및 제1 도메인 및 제2 도메인에서의 표현 및/또는 채도 비디오 유닛의 스케일링 채도 잔차에 기초하여 비디오 영역의 비디오 유닛의 재구성을 위한 파라미터들을 제공하고, 및 채도 양자화 파라미터(quantization parameter)(QP) 는 그 값이 각 블록 또는 변환 유닛에 대해 도출되는 오프셋(offset)을 갖는다.

다른 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하기 위해 사용될 수 있다. 방법은 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 간의 전환을 수행하는 단계를 포함하고, 코딩된 표현은 인 루프 변형(ILR)에 적용 가능한 정보 및 제1 도메인 및 제2 도메인에서의 표현 및/또는 채도 비디오 유닛의 스케일링 채도 잔차에 기초하여 비디오 영역의 비디오 유닛의 재구성을 위한 파라미터들을 제공하고, 및 휘도 양자화 파라미터(QP)는 그 값이 각 블록 또는 변환 유닛에 대해 도출되는 오프셋을 갖는다.

위에 개시된 방법들 중 하나 이상은 인코더(encoder) 측 구현 또는 디코더(decoder) 측 실행일 수 있다.

또한, 대표적인 양태에서, 프로세서 및 그에 대한 명령어들이 있는 비일시적 메모리를 포함하는 비디오 시스템의 장치가 개시된다. 프로세서에 의한 실행 시에 명령어들은 프로세서가 개시된 방법 중 임의의 하나 이상을 실행하게 한다.

또한, 비일시적 컴퓨터 판독 가능 매체에 저장된 컴퓨터 프로그램 제품에 있어서, 개시된 방법 중 임의의 하나 이상을 수행하기 위한 프로그램 코드를 포함하는 컴퓨터 프로그램 제품이 개시된다.

개시된 기술의 상기 및 기타 양태 및 특징은 도면, 설명 및 청구항에서 보다 상세하게 설명된다.

도면의 간단한 설명

도 1은 병합 후보 리스트를 구성하는 예를 도시한다.

도 2는 공간적 후보들의 위치들의 예를 도시한다.

도 3은 공간적 병합 후보들의 중복 검사를 받는 후보 쌍들의 예를 도시한다.

도 4a 및 도 4b는 현재 블록의 크기 및 형태에 따른 제2 예측 유닛(PU)의 위치의 예들을 도시한다.

도 5는 시간적 병합 후보들에 대한 모션 벡터 스케일링의 예를 도시한다.

도 6은 시간적 병합 후보들에 대한 후보 위치들의 예를 도시한다.

도 7은 결합된 이중 예측 병합 후보를 생성하는 예를 도시한다.

도 8은 모션 벡터 예측 후보들을 구성하는 예를 도시한다.

도 9는 공간적 모션 벡터 후보들에 대한 모션 벡터 스케일링의 예를 도시한다.

도 10은 대체 시간적 모션 벡터 예측(ATMVP)의 예를 도시한다.

도 11은 공간적-시간적 모션 벡터 예측의 예를 도시한다.

도 12는 로컬 조도 보상 파라미터들을 도출하기 위한 이웃한 샘플들의 예를 도시한다.

도 13a 및 13b는 각각 4-파라미터 아핀 모델 및 6-파라미터 아핀 모델과 관련된 예시들을 도시한다.

도 14는 서브-블록 당 아핀 모션 벡터 필드의 예를 도시한다.

도 15a 및 15b는 각각 4-파라미터 아핀 모델 및 6-파라미터 아핀 모델의 예들을 도시한다.

도 16은 상속된 아핀 후보들에 대한 아핀 상호 모드에 대한 모션 벡터 예측의 예를 도시한다.

도 17은 구성된 아핀 후보에 대한 아핀 상호 모드에 대한 모션 벡터 예측의 예를 도시한다.

도 18a 및 18b는 아핀 병합 모드와 관련된 예시들을 도시한다.

도 19는 아핀 병합 모드에 대한 후보 위치들의 예들을 도시한다.

도 20은 UMVE(ultimate vector expression) 탐색 프로세스의 일 예를 도시한다.

도 21은 UMVE 탐색점의 일 예를 도시한다.

도 22는 디코더 측 모션 비디오 개선(DMVR)의 예를 도시한다.

도 23은 변형 단계에 의한 디코딩의 블록도 흐름도를 도시한다.

도 24는 양방향 필터의 샘플들의 예들을 도시한다.

도 25는 가중치 계산에 이용되는 창 샘플들(windowed samples)의 예들을 도시한다.

도 26은 예시적인 스캔 패턴을 도시한다.

도 27a 및 27b는 본 명세서에 설명된 시각적 미디어 처리를 실행하기 위한 하드웨어 플랫폼의 예의 블록도들이다.

도 28a 내지 28e는 개시된 기술의 일부 실행들에 기초한 비디오 처리를 위일 예시적인 방법들의 흐름도들을 도시한다.

발명을 실시하기 위한 구체적인 내용

1. HEVC/H.265의 비디오 코딩

비디오 코딩 표준은 주로 잘 알려진 ITU-T 및 ISO/IEC 표준의 개발을 통해 발전해 왔다. ITU-T는 H.261 및 H.263을 생성하고 ISO/IEC는 MPEG-1 및 MPEG-4 Visual을 생성했고, 두 조직들은 H.262/MPEG-2 비디오 및 H.264/MPEG-4 고급 비디오 코딩(AVC) 및 H.265/HEVC 표준을 공동으로 제작했다. H.262 이후로, 비디오 코딩 표준은 시간적 예측과 변환 코딩이 사용되는 하이브리드 비디오 코딩 구조를 기반에 기초한다. HEVC를 넘어선 미래 비디오 코딩 기술을 탐색하기 위해 2015년 VCEG와 MPEG가 공동으로 JVET(Joint Video Exploration Team)를 설립했다. 그 이후로 JVET에서 많은 새로운 방법을 채택하고 JEM(Joint Exploration Model)이라는 참조 소프트웨어에 적용했다. 2018년 4월, VCEG(Q6/16)와 ISO/IEC JTC1 SC29/WG11(MPEG) 간의 JVET(Joint Video Expert Team)가 HEVC에 비해 50% 비트 전송률(bitrate) 감소를 목표로 하는 VVC 표준 작업을 위해 만들어졌다. VVC 초안의 최신 버전, 즉 다목적 비디오 코딩(초안 2)은 다음에서 찾을 수 있다.

http://phenix.it-sudparis.eu/jvet/doc_end_user/documents/11_Ljubljana/wg11/JVET-K1001-v7.zip

VTM이라는 VVC의 최신 참조 소프트웨어는 다음에서 찾을 수 있다.

https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/tags/VTM-2.1

2.1. HEVC/H.265의 상호 예측(Inter prediction)

각각의 상호-예측된(inter-predicted) PU는 하나 또는 두 개의 참조 영상 리스트들(reference picture lists)에 대한 모션 파라미터들(motion parameters)을 갖는다. 모션 파라미터들은 모션 벡터와 참조 영상 인덱스를 포함한다. 두 개의 참조 영상 리스트들 중 하나의 사용은 inter_pred_idc를 사용하여 시그널링될 수도 있다. 모션 벡터들은 예측 변수들(predictors)에 대한 델타들로서, 명확하게(explicitly) 코딩될 수 있다.

CU(coding unit)가 스킵 모드(skip mode)로 코딩될 때, 하나의 PU는 CU와 연관되고 중요한 잔차 계수들(significant residual coefficients), 코딩된 모션 벡터 델타 또는 참조 영상 인덱스가 없다. 병합 모드

(merge mode)가 명시되어 현재 PU에 대한 모션 파라미터들이 공간적(spatial) 및 시간적(temporal) 후보들을 포함하는 이웃한 PU들로(neighbouring PUs)부터 획득된다. 병합 모드는 스킵 모드뿐만 아니라 상호-예측된 전체 PU에 적용될 수 있다. 병합 모드의 대안은 모션 파라미터들의 명확한 전송(explicit transmission)이다. 여기서 모션 벡터(더 정확하게 말하면 모션 벡터 예측 변수와 비교하여 모션 벡터 차이들(motion vector differences)(MVD)), 각 참조 영상 리스트 및 참조 영상 리스트 사용(usage)에 대한 대응하는 참조 영상 인덱스 각 PU마다 명확하게 시그널링된다. 이러한 모드는 본 개시에서 AMVP(Advanced Motion Vector Prediction)로 명명된다.

시그널링이 2개의 참조 영상 리스트들 중 하나가 사용될 것임을 나타내는 경우, PU는 샘플들의 한 블록으로부터 생성된다. 이것을 "단방향 예측(uni-prediction)"이라고 한다. 단방향 예측은 P-슬라이스들(P-slices)과 B-슬라이스들(B-slices) 전체에 적용 가능하다.

시그널링이 참조 영상 리스트들 전체가 사용될 것임을 나타내는 경우, PU는 샘플들의 2개 블록들로부터 생성된다. 이를 "이중-예측(bi-prediction)"이라고 한다. 이중-예측은 B-슬라이스들에만 적용 가능하다.

다음 텍스트는 HEVC에 명시된 상호 예측 모드들에 대한 세부 정보들을 제공한다. 설명은 병합 모드로 시작된다.

2.1.1. 참조 영상 리스트

HEVC에서, 텀 상호 예측(term inter prediction)은 현재 디코딩된(decoded) 영상 이외의 참조 영상의 데이터 요소들(예를 들어, 샘플 값들 또는 모션 벡터들)로부터 도출된 예측을 나타내기 위해 사용된다. H.264/AVC에서와 같이 영상은 다수의 참조 영상들로부터 예측될 수 있다. 상호 예측에 사용되는 참조 영상들은 하나 이상의 참조 영상 리스트들로 구성된다. 참조 인덱스는 리스트의 참조 영상들 중 예측 신호를 생성하기 위해 이용해야 하는 참조 영상들을 식별한다.

단일(single) 참조 영상 리스트인 List 0은 P 슬라이스에 사용되고, 두 개의 참조 영상 리스트들인 List 0과 List 1은 B 슬라이스에 사용된다. List 0/1에 포함된 참조 영상은 캡처/표시 순서 측면에서 과거 및 미래 영상들로부터의 것일 수 있다.

2.1.2. 병합 모드

2.1.2.1. 병합 모드의 후보들 도출(Derivation)

병합 모드를 사용하여 PU가 예측되는 경우, 병합 후보들의 리스트의 항목(entry)을 가리키는 인덱스(index)가 비트스트림(bitstream)에서 분석(parse)되고 모션 정보를 탐색하기 위해 이용된다. 이 리스트의 구성은 HEVC 표준에 명시되어 있고, 다음 단계들의 순서에 따라 요약될 수 있다.

o 단계 1: 초기 후보들 도출

o 단계 1.1: 공간적 후보들 도출

o 단계 1.2: 공간적 후보들에 대한 중복 검사(Redundancy check)

o 단계 1.3: 시간적 후보들 도출

o 단계 2: 추가(Additional) 후보들 삽입

o 단계 2.1: 이중 예측 후보들 생성

o 단계 2.2: 제로 모션 후보들 삽입

이 단계들은 또한 도 1에 개략적으로 도시되어 있다. 공간적 병합 후보 도출의 경우, 서로 다른 5개의 위치들에 위치한 후보들 중 최대 4개의 병합 후보들이 선택된다. 시간적 병합 후보 도출의 경우, 두 후보들 중 최대 하나의 병합 후보가 선택된다. 디코더에서 각 PU에 대한 일정한 수(constant number)의 후보가 가정되기 때문에, 단계 1에서 획득된 후보들의 수가 슬라이스 헤더(slice header)에서 시그널링되는 최대 병합 후보 수(MaxNumMergeCand)에 도달하지 않으면 추가 후보들이 생성된다. 후보들의 수가 일정하기 때문에 최상의(best) 병합 후보의 인덱스는 TU(truncated unary binarization)를 사용하여 인코딩된(encoded)다. CU의 크기가 8인 경우, 현재 CU의 전체 PU들은 2NX2N 예측 유닛의 병합 후보 리스트와 동일한 단일 병합 후보 리스트를 공유한다.

다음에서, 전술한 단계들과 연관된 연산들이 상세하게 설명된다.

2.1.2.2. 공간적 후보들 도출

공간적 병합 후보들의 도출에 있어서, 도 2에 도시된 위치들에 위치한 후보들 중 최대 4개의 병합 후보들이 선택된다. 도출 순서는 A1, B1, B0, A0 및 B2이다. 위치 B2는 위치 A1, B1, B0, A0의 PU가 적용 가능하지 않거나 (예: 다른 슬라이스 또는 타일에 속하기 때문에) 또는 인트라 코딩된 경우에만 고려된다. 위치 A1의 후보가 추가된 후, 나머지 후보들의 추가는 중복 검사를 거쳐 동일한 모션 정보를 가지는 후보들을 리스트에서 제외하여 코딩 효율성(efficiency)을 향상시킨다. 계산 복잡성(computational complexity)을 줄이기 위해 언급된 중복 검사에서 전체 가능한 후보 쌍들이 고려되지는 않는다. 대신에 도 3에서 화살표로 연결된 쌍들만이 고려되고, 중복 검사에 사용되는 대응하는 후보가 동일한 모션 정보가 아닌 경우에만 후보가 리스트에 추가된다. 중복된 모션 정보(duplicate motion information)의 또 다른 소스는 2Nx2N과 다른 분할들과 연관된 "제2 PU"이다. 예로서, 도 4a 및 4b는 각각 NX2N 및 2NXN의 경우들에 대한 제2 PU를 도시한다. 현재 PU가 NX2N으로 분할되는 경우, 위치 A1의 후보는 리스트 구성을 위해 고려되지 않는다. 사실, 이 후보를 추가함으로써 동일한 모션 정보를 가지는 2개의 예측 유닛들이 생성될 것이고, 이는 코딩 유닛에서 단지 하나의 PU를 갖는 것으로 중복된다. 유사하게, 현재 PU가 2NXN으로 분할되는 경우, 위치 B1은 고려되지 않는다.

2.1.2.3. 시간적 후보들 도출

이 단계에서는 하나의 후보만 리스트에 추가된다. 특히, 이 시간적 병합 후보의 도출에서, 스케일링된 모션 벡터는 주어진 참조 영상 리스트 내에서 현재 영상의 POC 차이가 가장 작은 영상에 속하는 같은 위치에 있는 PU에 기초하여 도출된다. 같은 위치에 있는(co-located) PU의 도출에 사용되는 참조 영상 리스트는 슬라이스 헤더에서 명확하게 시그널링된다. 시간적 병합 후보에 대한 스케일링된 모션 벡터는 POC 거리들, tb 및 td를 사용하여 같은 위치에 있는 PU의 모션 벡터로부터 스케일링된 도 5의 점선으로 도시된 바와 같이 획득되고, 여기서 tb는 현재 영상의 참조 영상과 현재 영상 간의 POC 차이로 정의되고 td는 같은 위치에 있는 영상의 참조 영상과 같은 위치에 있는 영상 간의 POC 차이로 정의된다. 시간적 병합 후보의 참조 영상 인덱스는 0으로 설정된다. 스케일링 프로세스의 실제 실현(practical realization)은 HEVC 사양(specification)에 설명되어 있다. B-슬라이스의 경우, 하나는 참조 영상 리스트 0에 대한 것이고 다른 하나는 참조 영상 리스트 1에 대한 2개의 모션 벡터들이 획득되고 결합되어 이중-예측 병합 후보가 된다.

참조 프레임에 속하는 같은 위치에 있는 PU(Y)에서, 시간적 후보에 대한 위치는 도 6에 도시된 바와 같이 후보 C0와 C1 사이에서 선택된다. 위치 C0의 PU가 적용 가능하지 않거나, 인트라 코딩되거나, 현재 코딩 트리 유닛(tree unit)(CTU a/k/a LCU, 최대 코딩 유닛) 행의 외부에 있는 경우 위치 C1이 사용된다. 그렇지 않으면 위치 C0이 시간적 병합 후보의 도출에 사용된다.

2.1.2.4. 추가 후보들 삽입

공간적, 시간적 병합 후보들 외에도, 병합 후보들의 두 가지 추가 유형들이 있다. 결합된 이중 예측 병합 후보와 제로 병합 후보이다. 결합된 이중 예측 병합 후보들은 공간적 및 시간적 병합 후보들을 이용하여 생성된다. 결합된 이중 예측 병합 후보는 B-슬라이스에만 사용된다. 결합된 이중 예측 후보들은 초기 후보의 제1 참조 영상 리스트 모션 파라미터들을 다른 후보의 제2 참조 영상 리스트 모션 파라미터들과 결합함으로써 생성된다. 이 두 튜플들(tuples)이 서로 다른 모션 가설들(hypotheses)을 제공하는 경우 새로운 이중 예측 후보를 형성한다. 예로서, 도 7은 mvL0 및 refIdxL0 또는 mvL1 및 refIdxL1을 가지는 원본 리스트(original list)(왼쪽)의 두 후보들이 최종 리스트(final list)(오른쪽)에 추가된 결합된 이중 예측 병합 후보를 생성하기 위해 사용되는 경우를 도시한다. 이러한 추가 병합 후보들을 생성하는 것으로 간주되는 조합들에 관한 많은 규칙들이 있다.

병합 후보들 리스트의 나머지 항목들을 채우기 위해 제로 모션 후보들이 삽입되므로 MaxNumMergeCand 용량(capacity)에 도달한다. 이러한 후보들은 제로 공간적 변위(spatial displacement)를 가지고, 참조 영상 인덱스가 0에서부터 시작하여 새로운 제로 모션 후보가 리스트에 추가될 때마다 증가한다. 마지막으로 이러한 후보들에 대해서는 중복 검사가 수행되지 않는다.

2.1.3. AMVP

AMVP는 모션 파라미터들의 명확한 전송에 사용되는 이웃한 PU와 모션 벡터의 시공간적(spatio-temporal) 상관(correlation)을 활용한다. 각 참조 영상 리스트에 대해, 모션 벡터 후보 리스트는 먼저 시간적으로 이웃하는 PU 위치들 위의 왼쪽의 가용성(availability)을 확인하고, 중복 후보들을 제거하고, 후보 리스트를 일정한 길이

로 만들기 위해 제로 벡터를 추가하여 구성된다. 그런 다음 인코더는 후보 리스트로부터 최상의 예측 변수를 선택하고 선택한 후보를 나타내는 대응하는 인덱스를 전송할 수 있다. 병합 인덱스 시그널링과 유사하게, 최상의 모션 벡터 후보의 인덱스는 절단된 단항(truncated unary)을 사용하여 인코딩된다. 이 경우 인코딩될 최댓값은 2이다(예를 들어, 도 8 참조). 다음 섹션들에서는 모션 벡터 예측 후보의 도출 프로세스에 대한 세부 정보들이 제공된다.

2.1.3.1. AMVP 후보들 도출

도 8은 모션 벡터 예측 후보의 도출 프로세스를 요약한 것이다.

모션 벡터 예측에서는 두 가지 유형들의 모션 벡터 후보들이 고려된다: 공간적 모션 벡터 후보와 시간적 모션 벡터 후보. 공간적 모션 벡터 후보 도출을 위해, 2개의 모션 벡터 후보들은 도 2에 도시된 바와 같이 5개의 상이한 위치들에 위치한 각 PU의 모션 벡터들에 기초하여 최종적으로 도출된다.

시간적 모션 벡터 후보 도출을 위해, 하나의 모션 벡터 후보는 두 개의 서로 다른 위치들에 기반하여 도출된 두 개의 후보들 중에서 선택된다. 시공간적 후보들의 제1 리스트가 만들어진 후 리스트에서 중복된 모션 벡터 후보들이 제거된다. 잠재적(potential) 후보들의 수가 2보다 큰 경우, 연관된 참조 영상 리스트 내의 참조 영상 인덱스가 1보다 큰 모션 벡터 후보들은 리스트에서 제거된다. 시공간적 모션 벡터 후보들의 개수가 2개 미만인 경우 제로 모션 벡터 후보들이 리스트에 추가된다.

2.1.3.2. 공간적 모션 벡터 후보들

공간적 모션 벡터 후보들의 도출에서, 5개의 잠재적 후보들 중 최대 2개의 후보들이 고려되고, 이는 도 2에 도시된 위치들에 위치한 PU들로부터 도출되고, 그 위치들은 모션 병합의 위치들과 동일하다. 현재 PU의 왼쪽에 대한 도출 순서는 A0, A1 및 스케일링된 A0, 스케일링된 A1으로 정의된다. 현재 PU의 위쪽에 대한 도출 순서는 B0, B1, B2, 스케일링된 B0, 스케일링된 B1, 스케일링된 B2로 정의된다. 따라서 각 측면에 대해 모션 벡터 후보로 사용할 수 있는 4개의 경우들이 있고, 2개의 경우들은 공간적 스케일링을 사용할 필요가 없고 2개의 경우들은 공간적 스케일링이 사용된다. 네 가지 다른 경우들을 요약하면 다음과 같다.

* 공간적 스케일링 없음

- (1) 동일한 참조 영상 리스트 및 동일한 참조 영상 인덱스(동일한 POC)

- (2) 다른 참조 영상 리스트, 그러나 동일한 참조 영상(동일한 POC)

* 공간적 스케일링

- (3) 동일한 참조 영상 리스트, 그러나 다른 참조 영상(다른 POC)

- (4) 다른 참조 영상 리스트 및 다른 참조 영상(다른 POC)

공간적 스케일링이 없는 경우들을 먼저 확인한 다음 공간적 스케일링을 수행한다. 공간적 스케일링은 참조 영상 리스트에 관계없이 이웃한 PU의 참조 영상과 현재 PU의 참조 영상의 POC가 다를 때 고려된다. 왼쪽 후보들의 전체 PU들이 적용 가능하지 않거나 인트라 코딩된 경우, 위 모션 벡터에 대한 스케일링은 왼쪽 및 위쪽 MV(motion vector) 후보들의 병렬 도출을 돕기 위해 허용된다. 그렇지 않으면 위의 모션 벡터에 대해 공간적 스케일링이 허용되지 않는다.

공간적 스케일링 프로세스에서, 이웃한 PU의 모션 벡터는 도 9에 도시된 바와 같이 시간적 스케일링과 유사한 방식으로 스케일링된다. 주요 차이점은 현재 PU의 참조 영상 리스트와 인덱스가 입력으로 제공된다는 것이다; 실제 스케일링 프로세스는 시간적 스케일링 프로세스와 동일하다.

2.1.3.3. 시간적 모션 벡터 후보들

참조 영상 인덱스 도출을 제외하고, 시간적 병합 후보들의 도출을 위한 전체 프로세스들은 공간적 모션 벡터 후보들의 도출과 동일하다(도 6 참조). 참조 영상 인덱스는 디코더에 시그널링된다.

2.2. JEM의 서브-CU 기반 모션 벡터 예측 방법

QTBT가 있는 JEM에서 각 CU는 각 예측 방향에 대해 최대 하나의 모션 파라미터들의 세트를 가질 수 있다. 큰 CU를 서브-CU들로 분할하고 큰 CU의 전체 서브-CU들에 대한 모션 정보를 도출함으로써 두 개의 서브-CU 레벨 모션 벡터 예측 방법이 인코더에서 고려된다. 대체 시간적 모션 벡터 예측(Alternative temporal motion vector

prediction)(ATMVP) 방법은 각 CU가 배치된 참조 영상의 현재 CU보다 작은 다수의 블록들로부터 모션 정보의 다수의 세트들을 가져올 수 있도록 한다. 공간적-시간적 모션 벡터 예측(STMVP) 방법에서 서브-CU들의 모션 벡터들은 시간적 모션 벡터 예측 변수와 공간적 이웃한 모션 벡터를 사용하여 재귀적(recursively)으로 도출된다.

서브-CU 모션 예측을 위한 더 정확한 모션 필드를 보존하기 위해, 참조 프레임들에 대한 모션 압축(motion compression)은 현재 비활성화되어 있다.

2.2.1. 대체 시간적 모션 벡터 예측

도 10은 대체 시간적 모션 벡터 예측(ATMVP)의 예를 도시한다. 대체 시간적 모션 벡터 예측(ATMVP) 방법에 있어서, 모션 벡터들 시간적 모션 벡터 예측(TMVP)은 현재 CU보다 작은 블록들로부터 모션 정보(모션 벡터들 및 참조 인덱스들을 포함)의 다수의 세트들을 가져옴으로써 수정된다. 서브-CU들은 정사각형 NXN 블록들이다(N은 기본값(default)에 의해 4로 설정됨).

ATMVP는 2개의 단계들로 CU 내 서브-CU들의 모션 벡터들을 예측한다. 제1 단계는 소위(so-called) 시간적 벡터를 사용하여 참조 영상에서 대응되는 블록을 식별하는 것이다. 참조 영상을 모션 소스 영상이라고 한다. 제2 단계는 현재 CU를 서브-CU들로 분할하고 각 서브-CU에 대응하는 블록에서 각 서브-CU의 참조 인덱스들과 모션 벡터들을 획득하는 것이다.

제1 단계에서, 현재 CU의 공간적 이웃한 블록들의 모션 정보에 의해 참조 영상과 대응하는 블록이 결정된다. 이웃한 블록들의 반복적인 스캐닝 프로세스를 피하기 위해 현재 CU의 병합 후보 리스트에서 제1 병합 후보를 사용한다. 제1 적용 가능한 모션 벡터와 연관된 참조 인덱스는 시간적 벡터 및 모션 소스 영상에 대한 인덱스로 설정된다. 이러한 방식으로 ATMVP에서는 TMVP에 비해 대응하는 블록을 더 정확하게 식별할 수 있고, 대응하는 블록(때때로 병치된(collocated) 블록이라고도 함)은 현재 CU에 대해 항상 오른쪽-하단 또는 중심 위치에 있다.

제2 단계에서, 현재 CU의 좌표에 시간적 벡터를 추가함으로써, 서브-CU의 대응하는 블록은 모션 소스 영상의 시간적 벡터에 의해 식별된다. 각 서브-CU에 대해, 대응하는 블록의 모션 정보(중심 샘플을 포함하는 가장 작은 모션 그리드)는 서브-CU에 대한 모션 정보를 도출하기 위해 이용된다. 대응하는 NXN 블록의 모션 정보를 확인한 후, HEVC의 TMVP와 동일한 방식으로 현재 서브-CU의 모션 벡터들 및 참조 인덱스들로 전환되고, 모션 스케일링 및 기타 절차들이 적용된다. 예를 들어, 디코더는 저지연 조건(low-delay condition)(즉, 현재 영상의 전체 참조 영상들의 POC들이 현재 영상의 POC보다 작음)이 충족되는지 여부를 확인하고 가능하게는 모션 벡터 MVx(참조 영상 리스트 X에 대응하는 모션 벡터)를 사용하여 각 서브-CU에 대한 모션 벡터 MVy (X는 0 또는 1이고 Y는 1-X임)를 예측한다.

2.2.2. 시공간적 모션 벡터 예측(STMVP)

이 방법에서 서브-CU들의 모션 벡터들은 래스터(raster) 스캔 순서에 따라 재귀적으로 도출된다. 도 11은 이러한 개념을 예시한다. 4개의 4X4 서브-CU들 A, B, C 및 D를 포함하는 8X8 CU를 생각해 본다. 현재 프레임에서 이웃하는 4x4 블록들은 a, b, c 및 d로 레이블이 명시된다.

서브-CU A에 대한 모션 도출은 2개의 공간적 이웃들을 식별하는 것으로 시작한다. 제1 이웃은 서브-CU A 위의 NXN 블록(블록 c)이다. 이 블록 c가 적용 가능하지 않거나, 인트라 코딩된 경우 서브-CU A 위의 다른 NXN 블록들이 확인된다(블록 c에서 시작하여 왼쪽에서부터 오른쪽으로). 제2 이웃은 서브-CU A의 왼쪽에 있는 블록(블록 b)이다. 블록 b가 적용 가능하지 않거나 인트라 코딩된 경우 서브-CU A의 왼쪽의 다른 블록들이 확인된다(블록 b에서 시작하여 상단에서부터 하단으로). 각 리스트에 대한 이웃한 블록들에서 획득된 모션 정보는 주어진 리스트에 대한 제1 참조 프레임으로 스케일링된다. 다음으로, 서브-블록 A의 시간적 모션 벡터 예측 변수(TMVP) 는 HEVC에 명시된 것과 동일한 TMVP 도출 절차에 따라 도출된다. 위치 D에 병치된 블록의 모션 정보를 가져와 그에 따라 스케일링한다. 마지막으로, 모션 정보를 탐색하고 스케일링한 후 적용 가능한 전체 모션 벡터들(최대 3개)가 각 참조 리스트에 대해 개별적으로 평균화된다. 평균화된 모션 벡터는 현재 서브-CU의 모션 벡터로 할당된다.

2.2.3. 서브-CU 모션 예측 모드 시그널링

서브-CU 모드들은 추가 병합 후보들로 활성화되고, 모드들에 신호를 보내는 데 필요한 추가 신택스 요소는 없다. ATMVP 모드와 STMVP 모드를 나타내기 위해 각 CU의 병합 후보들 리스트에 2개의 추가 병합 후보들이 추가된다. 시퀀스 파라미터 세트가 ATMVP 및 STMVP가 활성화됨을 나타내는 경우, 최대 7개의 병합 후보들이 사용된다. 추가 병합 후보들의 인코딩 논리(encoding logic)는 HM의 병합 후보들과 동일하다. 즉, P 또는 B 슬라

이스의 각 CU에 대해 두 개의 추가 병합 후보들에 대해 두 번 더 RD 확인들이 필요하다.

JEM에서, 병합 인덱스의 전체 빈들은 CABAC에 의해 컨텍스트(context) 코딩된다. HEVC에서는 제1 빈만 컨텍스트 코딩하고 나머지 빈들은 컨텍스트 바이패스 코딩(context by-pass coded)된다.

2.3. JEM에서 로컬 조도 보상(Local illumination compensation in JEM)

로컬 조도 보상(LIC)은 스케일링 인자(scaling factor) a 및 오프셋 b를 사용하여 조도 변화들에 대한 선형 모델(linear model)에 기초한다. 그리고 각 상호-모드(inter-mode)로 코딩된 코딩 유닛(CU)에 대해 적응적으로 활성화 또는 비활성화된다.

LIC가 CU에 적용되는 경우, 현재 CU의 이웃한 샘플들과 대응하는 참조 샘플들을 사용하여 파라미터들 a 및 b를 도출하기 위해 최소 오차 자승법(least square error method)이 사용된다. 보다 구체적으로, 도 12에 도시된 바와 같이, CU의 서브샘플링된(2:1 서브샘플링) 이웃한 샘플들 및 참조 영상에서 대응하는 샘플들(현재 CU 또는 서브-CU의 모션 정보로 식별됨)이 사용된다.

2.3.1. 예측 블록들의 도출

IC 파라미터들은 각 예측 방향에 대해 별도로 도출되고 적용된다. 각 예측 방향에 대해, 디코딩된 모션 정보로 제1 예측 블록이 생성된 다음, LIC 모델을 적용하여 임시 예측 블록(temporary prediction block)이 획득된다. 이후, 2개의 임시 예측 블록들을 이용하여 최종 예측 블록을 도출한다.

CU가 병합 모드로 코딩될 때, LIC 플래그(flag)는 병합 모드에서 모션 정보 복사와 유사한 방식으로 이웃한 블록으로부터 복사되고, 그렇지 않으면, LIC가 적용되는지 여부를 나타내기 위해 CU에 대해 LIC 플래그가 시그널링된다.

영상에 대해 LIC가 활성화되는 경우, CU에 LIC가 적용되는지 여부를 결정하기 위해 추가 CU 레벨 RD 확인이 필요하다. CU에 대해 LIC가 활성화되는 경우, 절대 차이의 평균-제거 합(mean-removed sum of absolute difference)(MR-SAD) 및 절대 아다마르-변환 차이의 평균-제거 합(mean-removed sum of absolute Hadamard-transformed difference)(MR-SATD)은, SAD 및 SATD 대신, 정수 펠 모션 탐색(integer pel motion search) 및 분수 펠 모션 탐색(fractional pel motion search)에 각각 사용된다.

인코딩 복잡도를 줄이기 위해 JEM에는 다음과 같은 인코딩 방식이 적용된다.

* 현재 영상과 참조 영상들 사이에 명백한 조도 변화가 없는 경우, 전체 영상에 대해 LIC가 비활성화된다. 이를 식별하기 위해, 현재 영상의 히스토그램(histogram)들과 현재 영상의 전체 참조 영상이 인코더에서 계산된다. 현재 영상과 현재 영상의 전체 참조 영상 간의 히스토그램 차이가 주어진 임계값보다 작은 경우, 현재 영상에 대해 LIC가 비활성화된다. 그렇지 않으면 현재 영상에 대해 LIC가 활성화된다.

2.4. VVC의 상호 예측 방법

MVD 시그널링을 위한 적응형 모션 벡터 차이 해상도(Adaptive Motion Vector Difference Resolution)(AMVR), 아핀 예측 모드(affine prediction mode), 삼각 예측 모드(Triangular prediction mode)(TPM), ATMVP, 일반화된 이중 예측(Generalized Bi-Prediction)(GBI), 양방향 광학 흐름(Bi-directional Optical flow)(BIO)과 같은 상호 예측 개선을 위한 몇 가지 새로운 코딩 도구들이 있다.

2.4.1. VVC의 코딩 블록 구조

VVC에서, 영상을 정사각형 또는 직사각형 블록들로 분할하기 위해 쿼드트리(QuadTree)/이진트리(BinaryTree)/다중트리(MulitpleTree)(QT/BT/TT) 구조가 채택된다.

QT/BT/TT 외에 별도의 트리(일명 듀얼 코딩 트리(Dual coding tree))도 I-프레임들용 VVC에서 채택된다. 별도의 트리를 사용하여 코딩 블록 구조는 휘도 및 채도 구성요소(component)들에 대해 별도로 시그널링된다.

2.4.2 적응형 모션 벡터 차이 해상도

HEVC에서, 모션 벡터 차이(MVD)들(모션 벡터와 PU의 예측된 모션 벡터 사이)는 use_integer_mv_flag가 슬라이스 헤더에서 0과 같은 경우, 1/4 휘도 샘플들의 유닛들로 시그널링된다. VVC에서는 적응형 모션 벡터 차이 해상도(AMVR)가 도입된다. VVC에서 MVD는 1/4 휘도 샘플들, 정수 휘도 샘플들 또는 4개의 휘도 샘플들(즉, 1/4-펠, 1-펠, 4-펠) 유닛들로 코딩될 수 있다. MVD 해상도는 코딩 유닛(CU) 레벨에서 제어되고, MVD 해상도 플래그들

은 적어도 하나의 0이 아닌 MVD 구성요소들을 가지는 각 CU에 대해 조건부로 시그널링된다.

적어도 하나의 0이 아닌 MVD 구성요소들을 가지는 CU의 경우, 1/4 휘도 샘플 MV 정밀도(precision)가 CU에서 사용되는지 여부를 나타내기 위해 제1 플래그가 시그널링된다. 제1 플래그(1과 같음)가 1/4 휘도 샘플 MV 정밀도가 사용되지 않음을 나타내는 경우, 정수 휘도 샘플 MV 정밀도 또는 4 휘도 샘플 MV 정밀도가 사용되는지 여부를 나타내기 위해 다른 플래그가 시그널링된다.

CU의 제1 MVD 해상도 플래그가 0이거나 CU에 대해 코딩되지 않은 경우(CU의 전체 MVD들이 0임을 의미), 1/4 휘도 샘플 MV 해상도는 CU에 사용된다. CU가 정수 휘도 샘플 MV 정밀도 또는 4휘도 샘플 MV 정밀도를 사용하는 경우, CU에 대한 AMVP 후보 리스트의 MVP들은 대응하는 정밀도로 반올림된다.

2.4.3 아핀 모션 보상 예측

HEVC에서 모션 보상 예측(Motion Compensation Prediction)(MCP)을 위해 변환 모션 모델(translation motion model)만 적용된다. 현실 세계에는 확대/축소(zoom in/out), 회전(rotation), 원근감 모션들(perspective motions) 및 기타 불규칙한 모션들(irregular motions)과 같은 다양한 종류의 모션이 있다. VVC에서 단순화된 아핀 변환 모션 보상 예측은 4-파라미터 아핀 모델과 6-파라미터 아핀 모델에 적용된다. 도 13a 및 도 13b에 도시된 바와 같이, 블록의 아핀 모션 필드는 각각 4-파라미터 아핀 모델에 대한 2개의 제어점 모션 벡터(control point motion vector)(CPMV)들과 6-파라미터 아핀 모델에 대한 3개의 CPMV들로 설명된다.

블록의 모션 벡터 필드(motion vector field)(MVF)는 식 (1)의 4-파라미터 아핀 모델(여기서 4-파라미터는 변수 a, b, e 및 f로 정의됨)과 식 (2)의 6-파라미터 아핀 모델(여기서 4-파라미터는 변수 a, b, c, d, e 및 f로 정의됨)을 사용하여 다음 방정식들로 설명된다.

수학식 1

$$\begin{cases} mv^h(x,y) = ax - by + e = \dfrac{(mv_1^h - mv_0^h)}{w}x - \dfrac{(mv_1^v - mv_0^v)}{w}y + mv_0^h \\ mv^v(x,y) = bx + ay + f = \dfrac{(mv_1^v - mv_0^v)}{w}x + \dfrac{(mv_1^h - mv_0^h)}{w}y + mv_0^v \end{cases}$$

수학식 2

$$\begin{cases} mv^h(x,y) = ax + cy + e = \dfrac{(mv_1^h - mv_0^h)}{w}x + \dfrac{(mv_2^h - mv_0^h)}{h}y + mv_0^h \\ mv^v(x,y) = bx + dy + f = \dfrac{(mv_1^v - mv_0^v)}{w}x + \dfrac{(mv_2^v - mv_0^v)}{h}y + mv_0^v \end{cases}$$

여기서 $(mv^h_0,\ mv^h_0)$은 왼쪽 상단 모서리 제어점의 모션 벡터이고, $(mv^h_1,\ mv^h_1)$은 오른쪽 상단 모서리 제어점의 모션 벡터이고, $(mv^h_2,\ mv^h_2)$는 왼쪽 하단 모서리 제어점의 모션 벡터이고, 3개의 모션 벡터들은 전체 제어점 모션 벡터(CPMV)들이라고 하고, (x, y)는 현재 블록 내의 왼쪽 상단 샘플에 대한 대표점의 좌표를 나타내고, $(mv^h(x,y),\ mv^v(x,y))$는 (x,y)에 위치한 샘플에 대해 도출된 모션 벡터이다. CP 모션 벡터들은 (아핀 AMVP 모드에서와 같이) 시그널링되거나 즉석에서 (아핀 병합 모드에서와 같이) 도출될 수 있다. w와 h는 현재 블록의 너비와 높이이다. 실제로 나눗셈은 반올림 연산과 함께 오른쪽-시프트로 구현된다. VTM에서 대표점은 서브-블록의 중심 위치로 정의되고, 예를 들어, 현재 블록 내에서 왼쪽 상단 샘플에 대한 서브-블록의 왼쪽 상단 좌표가 (xs, ys)일 때 대표점의 좌표는 (xs+2, ys+2)로 정의된다. 각 서브-블록(즉, VTM에서 4x4)에 대해 대표점은

전체 서브-블록에 대한 모션 벡터를 도출하기 위해 활용된다.

모션 보상 예측을 더 단순화하기 위해 서브-블록 기반 아핀 변환 예측이 적용된다. 각 MXN(현재 VVC에서는 M과 N이 전체 4로 설정됨) 서브-블록의 모션 벡터를 도출하기 위해, 도 14에 도시된 바와 같이, 각 서브-블록의 중심 샘플의 모션 벡터는 수학식 1 및 2에 따라 계산되고, 1/16 분수 정확도로 반올림된다. 그런 다음 1/16-펠에 대한 모션 보상 보간 필터(motion compensation interpolation filter)들이 적용되어 도출된 모션 벡터로 각 서브-블록의 예측을 생성한다. 1/16-펠에 대한 보간 필터는 아핀 모드에 의해 도입된다.

MCP 이후에는 각 서브-블록의 높은 정확도 모션 벡터를 반올림하여 일반 모션 벡터와 동일한 정확도로 저장한다.

2.4.3.1. 아핀 예측의 시그널링

병진 모션 모델(translational motion model)과 유사하게, 아핀 예측으로 인한 부가 정보를 시그널링하기 위한 두 가지 모드들이 있다. AFFINE_INTER 및 AFFINE_MERGE 모드들이다.

2.4.3.2. AF_INTER 모드

너비와 높이가 전체 8보다 큰 CU들의 경우 AF_INTER 모드를 적용할 수 있다. CU 레벨의 아핀 플래그는 AF_INTER 모드가 사용되는지 여부를 나타내기 위해 비트스트림에서 시그널링된다.

이 모드에서, 각 참조 영상 리스트(List 0 또는 List 1)에 대해 다음 순서로 3가지 유형들의 아핀 모션 예측 변수들로 아핀 AMVP 후보 리스트가 구성되고, 각각의 후보는 현재 블록의 추정된 CPMV들을 포함한다. 인코더 측에서 발견된 최상의 CPMV(예: 도 17의 mv_0 mv_1 mv_2)와 추정된 CPMV의 차이들을 시그널링한다. 또한, 추정된 CPMV들이 도출되는 아핀 AMVP 후보의 인덱스가 더 시그널링된다.

1) 상속된 아핀 모션 예측 변수들(Inherited affine motion predictors)

확인 순서는 HEVC AMVP 리스트 구성에서 공간적 MVP들의 순서와 유사하다. 첫째, 왼쪽의 상속된 아핀 모션 예측 변수는 아핀 코딩되고 현재 블록에서와 동일한 참조 영상을 가지는 {A1, A0}의 제1 블록에서 도출된다. 둘째, 위의 상속된 아핀 모션 예측 변수는 아핀 코딩되고 현재 블록에서와 동일한 참조 영상을 가지는 {B1, B0, B2}의 제1 블록에서 도출된다. 5개의 블록들 A1, A0, B1, B0, B2가 도 16에 도시되어 있다.

이웃한 블록이 아핀 모드로 코딩된 것으로 발견되면, 이웃한 블록을 커버하는 코딩 유닛의 CPMV들은 현재 블록의 CPMV들의 예측 변수들을 도출하기 위해 이용된다. 예를 들어, A1이 비-아핀 모드로 코딩되고 A0이 4-파라미터 아핀 모드로 코딩된 경우 왼쪽 상속된 아핀 MV 예측 변수는 A0에서 도출된다. 이 경우, 도 18b에서 상단 왼쪽의 CPMV에 대한 MV_0^N 및 상단 오른쪽의 CPMV에 대한 MV_1^N에 의해 도시된 바와 같이, A0를 커버하는 CU의 CPMV들은, 상단-왼쪽(좌표 (x0, y0) 포함), 상단-오른쪽(좌표 (x1, y1) 포함), 및 하단-오른쪽 위치들(좌표 (x2, y2) 포함)에 대한 MV_0^C, MV_1^C,MV_2^C에 의해 도시된 바와 같이, 현재 블록의 추정된 CPMV들을 도출하기 위해 활용된다.

1) 구성된 아핀 모션 예측 변수들

구성된 아핀 모션 예측 변수는 도 17에 도시된 바와 같이 동일한 참조 영상을 갖는 이웃하는 상호 코딩된 블록으로부터 도출되는 제어점 모션 벡터(CPMV)들로 구성된다. 현재 아핀 모션 모델이 4-파라미터 아핀인 경우, CPMV들의 수는 2이고, 그렇지 않으면 현재 아핀 모션 모델이 6-파라미터 아핀인 경우 CPMV들의 수는 3이다. 왼쪽 상단 CPMV$\overline{mv}_0$은 상호 코딩되고 현재 블록과 동일한 참조 영상을 가지는 그룹 {A, B, C}의 제1 블록에서 MV에 의해 도출된다. 오른쪽 상단 CPMV$\overline{mv}_1$은 상호 코딩되고 현재 블록과 동일한 참조 영상을 가지는 그룹 {D, E}의 제1 블록에서 MV에 의해 도출된다. 왼쪽 하단 CPMV$\overline{mv}_2$는 상호 코딩되고 현재 블록과 동일한 참조 영상을 가지는 그룹 {F, G}의 제1 블록에서 MV에 의해 도출된다.

- 현재 아핀 모션 모델이 4-파라미터 아핀인 경우, 그런 다음 $\overline{mv}_0$ 및 $\overline{mv}_1$이 전체 발견되는 경우에만 구성된 아핀 모션 예측 변수가 후보 리스트에 삽입되고, 즉, $\overline{mv}_0$ 및 $\overline{mv}_1$은 현재 블록의 왼쪽 상단(좌표 (x0, y0) 사용), 오른쪽 상단(좌표 (x1, y1) 사용) 위치들에 대한 추정 CPMV들로 사용된다.

- 현재 아핀 모션 모델이 6-파라미터 아핀인 경우, 그런 다음 $\overline{mv}_0$ 및 $\overline{mv}_1$이 전체 발견되는 경우에만 구성된

아핀 모션 예측 변수가 후보 리스트에 삽입되고, 즉, $\overline{mv}_0$, $\overline{mv}_1$ 및 $\overline{mv}_2$은 현재 블록의 왼쪽 상단(좌표 (x0, y0) 사용), 오른쪽 상단(좌표 (x1, y1) 사용) 및 오른쪽 하단(좌표 (x2, y2) 사용) 위치들에 대한 추정 CPMV들로 사용된다.

구성된 아핀 모션 예측 변수를 후보 리스트에 삽입하는 경우, 가지 치기를 하는 프로세스(pruning process)가 적용되지 않는다.

3) 일반 AMVP 모션 예측 변수

아핀 모션 예측 변수의 수가 최대에 도달할 때까지 다음이 적용된다.

적용 가능한 경우 전체 CPMV들을 $\overline{mv}_2$와 동일하게 설정하여 아핀 모션 예측 변수를 도출한다.

적용 가능한 경우 전체 CPMV들을 $\overline{mv}_1$과 동일하게 설정하여 아핀 모션 예측 변수를 도출한다.

적용 가능한 경우 전체 CPMV들을 $\overline{mv}_0$과 동일하게 설정하여 아핀 모션 예측 변수를 도출한다.

적용 가능한 경우 전체 CPMV들을 HEVC TMVP와 동일하게 설정하여 아핀 모션 예측 변수를 도출한다.

전체 CPMV들을 0 MV로 설정하여 아핀 모션 예측 변수를 도출한다.

$\overline{mv}_i$는 구성된 아핀 모션 예측 변수에서 이미 도출되었다.

AF_INTER 모드에서, 4/6-파라미터 아핀 모드가 사용되는 경우, 2/3 제어점이 필요하므로 도 15a에 도시된 바와 같이 이러한 제어점에 대해 2/3 MVD를 코딩해야 한다. JVET-K0337에서는 MV를 다음과 같이 도출하는 것을 제안한다. 즉, mvd_0으로부터 mvd_1과 mvd_2를 예측한다.

수학식 3

$$mv_0 = \overline{mv}_0 + mvd_0$$

$$mv_1 = \overline{mv}_1 + mvd_1 + mvd_0$$

$$mv_2 = \overline{mv}_2 + mvd_2 + mvd_0$$

여기서 도 15b에 도시된 바와 같이, $\overline{mv}_i$, mvd_i 및 mv_1은 예측 모션 벡터, 모션 벡터 차이 및 각각 왼쪽 상단 픽셀(i = 0), 오른쪽 상단 픽셀(i = 1) 또는 왼쪽 하단 픽셀(i = 2)의 모션 벡터이다.

2개의 모션 벡터들(예: mvA(xA, yA) 및 mvB(xB, yB))의 추가는 개별적으로 두 구성요소들의 합과 같다. 즉, newMV = mvA + mvB이고 newMV의 두 구성요소들은 각각 (xA + xB)와 (yA + yB)로 설정된다.

2.4.3.3. AF_MERGE 모드

CU가 AF_MERGE 모드에서 적용되는 경우, 유효한 이웃한 재구성된 블록들에서 아핀 모드로 코딩된 첫 번째 블록을 가져온다. 그리고 후보 블록에 대한 선택 순서는 도 18a에 도시된 바와 같이 왼쪽, 위, 오른쪽 위, 왼쪽 아래에서 왼쪽 위(A, B, C, D, E 순서로 표시됨)이다. 예를 들어, 이웃한 왼쪽 하단 블록이 도 18b에서 A0로 표시된 바와 같이 아핀 모드로 코딩되는 경우, 블록 A를 포함하는 이웃한 CU/PU의 왼쪽 상단, 오른쪽 상단 및 왼쪽 하단 모서리의 제어점(CP) 모션 벡터 mv_0^N, $mv1^N$ 및 mv_2^N 이 반출된다. 그리고 현재 CU/PU에서 왼쪽 상단/오른쪽 상단/왼쪽 하단의 모션 벡터 mv_0^C, mv_1^C 및 mv_2^C(6-파라미터 아핀 모델에만 사용됨)는 mv_0^N, $mv1^N$ 및 mv_2^N에 기초하여 계산된다. VTM-2.0에서, 왼쪽 상단 모서리에 위치한 서브 블록(예: VTM의 4x4 블록)은 mv0을 저장하고, 현재 블록이 아핀 코딩된 경우 오른쪽 상단 모서리에 위치한 서브 블록은 mv1을 저장한다는 것을 유의해야 한다. 현재 블록이 6-파라미터 아핀 모델로 코딩된 경우 왼쪽 하단 모서리에 있는 서브-블록은 mv2를 저장한다. 그렇지 않으면(4-파라미터 아핀 모델 사용) LB는 mv2'를 저장한다. 다른 서브-블록들은 MC에 사용

된 MV를 저장한다.

현재 CU mv_0^C, mv_1^C 및 mv_2^C 의 CPMV를 도출한 후, 단순화된 아핀 모션 모델 수학식 1과 2에 따라 현재 CU의 MVF를 생성한다. 현재 CU가 AF_MERGE 모드로 코딩되었는지 여부를 식별하기 위해, 적어도 하나의 이웃한 블록이 아핀 모드로 코딩된 경우 비트스트림에서 아핀 플래그가 시그널링된다.

JVET-L0142 및 JVET-L0632에서, 아핀 병합 후보 리스트는 다음 단계로 구성된다.

1) 상속된 아핀 후보들 삽입

상속된 아핀 후보는 후보가 유효한 이웃한 아핀 코딩된 블록의 아핀 모션 모델로부터 도출됨을 의미한다. 최대 2개의 상속된 아핀 후보들은 이웃한 블록들의 아핀 모션 모델로부터 도출되어 후보 리스트에 삽입된다. 왼쪽 예측 변수의 경우, 스캔 순서는 {A0, A1}이다; 위의 예측 변수에 대해 스캔 순서는 {B0, B1, B2}이다.

2) 구성된 아핀 후보들 삽입

아핀 병합 후보 리스트에서 후보들의 수가 MaxNumAffineCand(예를 들어, 5)보다 작은 경우, 구성된 아핀 후보들을 후보 리스트에 삽입한다. 구성된 아핀 후보는 각 제어점의 이웃한 모션 정보를 결합하여 후보를 구성하는 것을 의미한다.

a) 제어점에 대한 모션 정보는 도 19에 도시된 특정 공간적 이웃들 및 시간적 이웃으로부터 먼저 도출된다. CPk(k=1, 2, 3, 4)는 k번째 제어점을 나타낸다. A0, A1, A2, B0, B1, B2 및 B3은 CPk를 예측하기 위한 공간적 위치들이다(k=1, 2, 3). T는 CP4를 예측하기 위한 시간적 위치이다.

CP1, CP2, CP3 및 CP4의 좌표들은 각각 (0, 0), (W, 0), (H, 0) 및 (W, H)이고, 여기서 W 및 H는 현재 블록의 너비와 높이이다.

각 제어점의 모션 정보는 다음과 같은 우선 순위에 따라 획득된다.

- CP1의 경우 검사 우선 순위는 B2->B3->A2이다. 적용 가능한 경우 B2가 사용된다. 그렇지 않고 B2가 적용 가능하지 않으면 B3이 사용된다. B2와 B3을 전체 적용할 수 없는 경우 A2가 사용된다. 3개의 후보들을 전체 적용할 수 없는 경우 CP1의 모션 정보를 얻을 수 없다.

- CP2의 경우 검사 우선 순위는 B1->B0이다.

- CP3의 경우 검사 우선 순위는 A1->A0이다.

- CP4의 경우 T가 사용된다.

a) 둘째, 제어점들의 조합들을 사용하여 아핀 병합 후보를 구성한다.

I. 파라미터 아핀 후보를 구성하기 위해서는 3개의 제어점들의 모션 정보가 필요하다. 3개의 제어점들은 다음 4가지 조합들({CP1, CP2, CP4}, {CP1, CP2, CP3}, {CP2, CP3, CP4}, {CP1, CP3, CP4}) 중 하나로부터 선택될 수 있다. {CP1, CP2, CP3}, {CP2, CP3, CP4}, {CP1, CP3, CP4} 조합들은 왼쪽 상단, 오른쪽 상단 및 왼쪽 하단 제어점들로 표시되는 6-파라미터 션 모델로 전환된다.

II. 4-파라미터 아핀 후보들을 구성하기 위해서 두 제어점들의 모션 정보가 필요하다. 두 제어점들은 두 가지 조합들({CP1, CP2}, {CP1, CP3}) 중 하나로부터 선택될 수 있다. 두 조합들은 왼쪽 상단 및 오른쪽 상단 제어점들로 표시되는 4파라미터 모션 모델로 전환된다.

III. 구성된 아핀 후보들의 조합들은 다음 순서로 후보 리스트에 삽입된다: {CP1, CP2, CP3}, {CP1, CP2, CP4}, {CP1, CP3, CP4}, {CP2, CP3, CP4}, {CP1, CP2}, {CP1, CP3}

i. 각 조합에 대해 각 CP에 대한 리스트 X의 참조 인덱스들이 확인되고 전체 동일한 경우, 이 조합은 리스트 X에 대한 유효한 CPMV들을 갖는다. 조합에 리스트 0과 리스트 1 전체에 대해 유효한 CPMV들이 없으면 이 조합은 유효하지 않은 것으로 표시된다. 그렇지 않으면 유효하며 CPMV들은 서브-블록 병합 리스트에 포함된다.

3) 모션 벡터들이 0인 패딩

아핀 병합 후보 리스트에서 후보들의 수가 5개 미만이면 리스트가 가득 찰 때까지 참조 인덱스들이 0인 0 모션 벡터를 후보 리스트에 삽입한다.

보다 구체적으로, 서브-블록 병합 후보 리스트의 경우, MV들이 (0, 0)으로 설정되고, 예측 방향이 단방향 예측으로 리스트 0(P 슬라이스의 경우)으로부터 설정되고, 이중 예측(B 슬라이스의 경우)으로 설정된다.

2.4.4. 모션 벡터 차이들과 병합(Merge with motion vector differences)(MMVD)

JVET-L0054에서, 궁극적인 모션 벡터 표현(ultimate motion vector expression)(UMVE, MMVD로도 알려짐)이 제시된다. UMVE는 제안된 모션 벡터 표현 방식으로 스킵 또는 병합 모드들 에 사용된다.

UMVE는 VVC의 일반 병합 후보 리스트에 포함된 것과 동일한 병합 후보를 재사용한다. 병합 후보들 중 기본 후보(base candidate)가 선택될 수 있고, 제안된 모션 벡터 표현 방법에 의해 더 확장된다.

UMVE는 시작점, 모션 크기 및 모션 방향을 이용하여 MVD를 표현내는 새로운 모션 벡터 차이(MVD) 표현 방법을 제공한다.

이 제안 기술은 병합 후보 리스트를 그대로 사용한다. 그러나 기본값 병합 유형(default merge type)(MRG_TYPE_DEFAULT_N)인 후보들만 UMVE의 확장에 대해 고려된다.

기본 후보 인덱스는 시작점을 정의한다. 기본 후보 인덱스는 다음과 같이 리스트에 있는 후보들 중 최상의 후보를 나타낸다.

표 1

기본 후보 IDX	0	1	2	3
N번째 MVP	1st MVP	2nd MVP	3rd MVP	4th MVP

표 1. 기본 후보 IDX기본 후보들의 수가 1인 경우 기본 후보 IDX는 시그널링되지 않는다.

거리 인덱스는 모션 크기 정보이다. 거리 인덱스는 시작점 정보로부터 미리 정의된 거리를 나타낸다. 미리 정의된 거리는 다음과 같다:

표 2

거리 IDX	0	1	2	3	4	5	6	7
픽셀 거리	1/4-펠	1/2-펠	1-펠	2-펠	4-펠	8-펠	16-펠	32-펠

표 2 거리 IDX방향 인덱스는 시작점에 대한 MVD의 방향을 나타낸다. 방향 인덱스는 아래와 같이 4가지 방향들을 나타낼 수 있다.

표 3

방향 IDX	00	01	10	11
x-축	+	-	N/A	N/A
y-축	N/A	N/A	+	-

표 3. 방향 IDXUMVE 플래그는 스킵 플래그 또는 병합 플래그를 보낸 직후에 시그널링된다. 스킵 또는 병합 플래그가 참이면, UMVE 플래그가 분석된다. UMVE 플래그가 1이면 UMVE 신택스들이 분석된다. 단, 1이 아니면 AFFINE 플래그를 분석한다. AFFINE 플래그가 1이면 AFFINE 모드이고, 1이 아니면 스킵/병합 인덱스는 VTM의 스킵/병합 모드에 대해 분석된다.

UMVE 후보들로 인한 추가적인 라인 버퍼(Additional line buffer)는 필요하지 않는다. 소프트웨어의 스킵/병합 후보가 직접 기본 후보로 사용되기 때문이다. 입력된 UMVE 인덱스를 사용하여 모션 보상 직전에 MV의 보완(supplement)을 결정한다. 이를 위해 긴 라인 버퍼를 보유(hold)할 필요가 없다.

현재의 공통 테스트 조건에서, 병합 후보 리스트의 제1 또는 제2 병합 후보가 기본 후보로 선택될 수 있다.

UMVE는 MV 차이들과의 병합(Merge with MV Differences)(MMVD)라고도 알려져 있다.

2.4.5. 디코더-측 모션 벡터 개선(Decoder-side Motion Vector Refinement)(DMVR)

이중 예측 연산에서, 하나의 블록 영역의 예측을 위해 list0의 모션 벡터(MV)와 list1의 모션 벡터(MV)를 각각 사용하여 형성된 두 개의 예측 블록들이 결합되어 하나의 예측 신호를 형성한다. 디코더-측 모션 벡터 개선(DMVR) 방법에서는 이중 예측의 두 모션 벡터들이 더 개선된다.

JEM 설계에서 모션 벡터들은 양방향 템플릿 매칭 프로세스에 의해 개선된다. 양방향 템플릿 매칭은 추가적인 모션 정보의 전송 없이 개선된 MV를 획득하기 위해 참조 영상들의 재구성 샘플들과 양방향 템플릿 사이의 왜곡 기반 탐색을 수행하기 위해 디코더에 적용되었다. 예가 도 22에 도시되어 있다. 양방향 템플릿은 도 22에 도시된 바와 같이 list0의 초기 MV0 및 list1의 MV1에서 각각 2개의 예측 블록의 가중 조합(weighted combination)(즉, 평균)으로 생성된다. 템플릿 매칭 연산은 생성된 템플릿과 참조 영상의 샘플 영역(초기 예측 블록 주변) 사이의 비용 측정값 계산으로 구성된다. 두 개의 참조 영상들 각각에 대해 최소 템플릿 비용을 산출하는 MV는 원본을 대체하기 위해 대응하는 리스트의 업데이트된 MV로 간주된다. JEM에서는 각 리스트에 대해 9개의 MV 후보들을 탐색한다. 9개의 MV 후보들에는 원본 MV와 8개의 주변 MV들이 포함되고, 수평 또는 수직 방향 또는 둘 전체에서 원본 MV에 대한 하나의 휘도 샘플 오프셋이 있다. 마지막으로 두 개의 새로운 MV들, 즉 도 22에 도시된 MV0' 및 MV1'은 최종 이중 예측 결과들을 생성하기 위해 사용된다. 절대 차이 합계(sum of absolute differences)(SAD)가 비용 측정으로 사용된다. 하나의 주변 MV에 의해 생성된 예측 블록의 비용을 계산하는 경우, 반올림된 MV(정수 펠로)는 실제 MV 대신 예측 블록을 획득하기 위해 실제로 사용된다.

DMVR의 프로세스를 더욱 단순화하기 위해 JVET-M0147은 JEM에서 설계에 대한 몇 가지 변경 사항들을 제안했다. 보다 구체적으로, VTM-4.0(곧 출시 예정)에 채택된 DMVR 설계는 다음과 같은 주요 기능들을 가지고 있다.

* list0과 list1 사이의 (0,0) 위치 SAD로 조기 종료

* DMVR의 블록 크기 W*H>=64 && H>=8

* CU 크기 > 16*16의 DMVR에 대해 CU를 여러 16x16 서브-블록들로 분할

* 참조 블록 크기(W+7)*(H+7)(휘도 용)

* 25포인트 SAD 기반 정수-펠 탐색(예: (+-) 2개 개선 탐색 범위, 단일 단계)

* 쌍선형 보간(Bilinear-interpolation) 기반 DMVR

* list0과 list1 간의 MVD 미러링을 통해 양방향 일치 허용

* "파라메트릭 오류 표면 방정식" 기반 서브-픽셀 개선

* 휘도/채도 MC w/ 참조 블록 패딩(필요한 경우)

* MC 및 TMVP에만 사용되는 개선된 MV들

2.4.6. 결합된 인트라 및 상호 예측

JVET-L0100에서 다중 가설 예측(multi-hypothesis prediction)이 제안되고, 결합된 인트라 및 상호 예측은 다중 가설들을 생성하는 한 방법이다.

다중 가설 예측이 인트라 모드를 개선하기 위해 적용되는 경우, 다중 가설 예측은 하나의 인트라 예측과 하나의 병합 인덱싱된 예측을 결합한다. 병합 CU에서 플래그가 참일 때 인트라 후보 리스트에서 인트라 모드를 선택하기 위해 병합 모드에 대해 하나의 플래그가 시그널링된다. 휘도 구성요소의 경우, 인트라 후보 리스트는 DC, 평면, 수평, 수직 모드의 4가지 인트라 예측 모드들로부터 도출되고, 인트라 후보 리스트의 크기는 블록 형태에 따라 3 또는 4가 될 수 있다. CU 너비가 CU 높이의 두 배보다 크면 가로 모드는 인트라 모드 리스트에서 제외되고 CU 높이가 CU 너비의 두 배보다 크면 세로 모드는 인트라 모드 리스트에서 제거된다. 인트라 모드 인덱스에 의해 선택된 하나의 인트라 예측 모드와 병합 인덱스에 의해 선택된 하나의 병합 인덱스 예측은 가중 평균을 이용하여 결합된다. 채도 구성요소의 경우 추가 시그널링 없이 항상 DM이 적용된다. 예측들을 결합하기 위한 가중치들은 다음과 같이 설명된다. DC 또는 평면 모드가 선택되거나 CB 너비 또는 높이가 4보다 작은 경우 동일한 가중치들이 적용된다. CB 너비 및 높이가 4 이상인 CB의 경우 수평/수직 모드가 선택되는 경우, 먼저 하나의 CB가 수직/수평으로 4개의 동일한 영역들로 분할된다. (w_intra_i, w_inter_i)로 표시된 각 가중치 세트, 여기서 i는 1에서 4이고 $(w_intra_1, w_inter_1) = (6, 2)$, $(w_intra_2, w_inter_2) = (5, 3)$, $(w_intra_3, w_inter_3) = (3, 5)$, $(w_intra_4, w_inter_4) = (2, 6)$은 대응되는 영역에 적용된다. (w_intra_1, w_inter_1)은 참조 샘플들

에 가장 가까운 영역이고 (w_intra_4, w_inter_4)는 참조 샘플들에서 가장 먼 영역이다. 그런 다음, 2개의 가중 예측들을 합산하고 3비트를 오른쪽 쉬프팅하여 결합 예측이 계산될 수 있다. 또한, 예측 변수들의 인트라 가설에 대한 인트라 예측 모드는 다음 이웃한 CU들의 참조를 위해 저장될 수 있다.

2.5 JVET-M0427의 인-루프 변형(ILR)

인-루프 변형(ILR)은 채도 스케일링을 사용한 휘도 매핑(Luma Mapping with Chroma Scaling)(LMCS)으로도 알려져 있다.

인-루프 변형(ILR)의 기본 개념은 원본(제1 도메인에서) 신호(예측/재구성 신호)를 제2 도메인(변형된 도메인)으로 변환하는 것이다.

인루프 휘도 변형 변수(luma reshaper)는 한 쌍의 순람표(look-up table)(LUT)들로 구현되지만 두 LUT들 중 하나만 시그널링되면, 다른 하나는 시그널링된 LUT으로부터 계산될 수 있다. 각 LUT는 1차원 10비트 1024 항목 매핑 테이블(1D-LUT)이다. 하나의 LUT는 입력 휘도 코드 값들 Y_i를 변경된 값들 Y_r: $Y_r = FwdLUT[Y_i]$에 매핑하는 순방향 LUT인 FwdLUT이다. 다른 LUT는 변경된 코드 값들 Y_r을 $\hat{Y}_i$: $\hat{Y}_i = InvLUT[Y_r]$에 매핑하는 역 LUT인 InvLUT이다. ($\hat{Y}_i$는 Y_i의 재구성 값들을 나타낸다.)

2.5.1 PWL 모델

개념적으로, 조각별 선형(piece-wise linear)(PWL)은 다음과 같은 방식으로 구현된다.

x1, x2를 2개의 입력 피벗 포인트라고 하고, y1, y2를 한 조각에 대한 대응하는 출력 피벗 포인트들이라고 하자. x1과 x2 사이의 입력 값 x에 대한 출력 값 y는 다음 방정식으로 보간될 수 있다:

수학식 4

$$y = ((y2-y1)/(x2-x1)) * (x-x1) + y1$$

고정 소수점 구현에서 방정식은 다음과 같이 다시 쓸 수 있다:

수학식 5

$$y = ((m * x + 2^{FP_PREC-1}) >> FP_PREC) + c$$

여기서 m은 스칼라, c는 오프셋, FP_PREC는 정밀도를 명시하는 상수 값이다.

CE-12 소프트웨어에서 PWL 모델은 1024 항목 FwdLUT 및 InvLUT 매핑 테이블들을 미리 계산하기 위해 이용된다. 그러나 PWL 모델은 구현들이 LUT들을 미리 계산하지 않고도 즉석에서 동일한 매핑 값들을 계산할 수 있도록 한다.

2.5.2. 테스트 CE12-2

2.5.2.1. 휘도 변형

인-루프 휘도 변형(즉, 제안의 CE12-2)의 테스트 2는 슬라이스 간 재구성에서 블록 유닛 인트라 예측을 위한 디코딩 지연 시간(latency)도 제거하는 더 낮은 복잡성 파이프라인을 제공한다. 인트라 예측은 상호 및 인트라 슬라이스 전체에 대해 재구성된 도메인에서 수행된다. 인트라 예측은 슬라이스 유형에 관계없이 항상 변형된 영역에서 수행된다. 이와 같이 구성하면 이전 TU 재구성이 완료된 직후 인트라 예측을 시작할 수 있다. 이러한 배치는 또한 슬라이스 종속(slice dependent) 대신 인트라 모드에 대한 통합 프로세스를 제공할 수 있다. 도 23은 모드에 기초한 CE12-2 디코딩 프로세스의 블록도를 나타낸다.

CE12-2는 또한 CE12-1의 32-조각 PWL 모델 대신에 휘도 및 채도 잔차 스케일링에 대한 16-조각 PWL 모델을 테스

트한다.

CE12-2에서 인-루프 휘도 변형 변수를 사용한 슬라이스 간 재구성(밝은 녹색 음영 블록은 변형된 도메인의 신호를 나타낸다: 휘도 잔차, 인트라 휘도 예측 및 인트라 휘도 재구성)

2.5.2.2. 휘도 종속 채도 잔차 스케일링

휘도 종속 채도 잔차 스케일링은 고정 소수점 정수 연산으로 구현된 곱셈의 프로세스이다. 채도 잔차 스케일링은 채도 신호와 휘도 신호의 상호작용을 보상한다. 채도 잔차 스케일링은 TU 레벨에서 적용된다. 보다 구체적으로 다음이 적용된다:

- 인트라의 경우 재구성된 휘도가 평균화된다.

- 인터(상호)의 경우 예측 휘도가 평균화된다.

평균은 PWL 모델에서 인덱스를 식별하기 위해 이용된다. 인덱스는 스케일링 인자 cScaleInv를 식별한다. 채도 잔차에 대응하는 숫자를 곱한다.

채도 스케일링 인자는 재구성된 휘도 값보다는 순방향 매핑된 예측 휘도 값으로부터 계산된다는 점에 유의한다.

2.5.2.3. ILR 부가 정보 신호

파라미터들은 (현재) 타일 그룹 헤더(tile group header)(ALF와 유사)에서 전송된다. 이들은 40-100비트를 필요로 한다고 한다.

다음 표는 JVET-L1001의 버전 9에 기초한다. 추가할 신택스는 밑줄이 그어진 굵은 기울임꼴 글꼴로 아래에 강조 표시되어 있다.

7.3.2.1 시퀀스 파라미터에서 RBSP 신택스는 다음과 같이 설정될 수 있다.

표 4

seq_parameter_set_rbsp( ) {	설명어(Descriptor)
sps_seq_parameter_set_id	ue(v)
intra_only_constraint_flag	u(1)
max_bitdepth_constraint_idc	u(4)
max_chroma_format_constraint_idc	u(2)
frame_only_constraint_flag	u(1)
no_qtbtt_dual_tree_intra constraint_flag	u(1)
no_sao_constraint_flag	u(1)
no_alf_constraint_flag	u(1)
no_pcm_constraint_flag	u(1)
no_temporal_mvp_constraint_flag	u(1)
no_sbtmvp_constraint_flag	u(1)
no_amvr_constraint_flag	u(1)
no_cclm_constraint_flag	u(1)
no_affine_motion_constraint_flag	u(1)
no_ladf_constraint_flag	u(1)
no_dep_quant_constraint_flag	u(1)
no_sign_data_hiding_constraint_flag	u(1)
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)
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
log2_max_pic_order_cnt_lsb_minus4	ue(v)

qtbtt_dual_tree_intra_flag	ue(v)
log2_ctu_size_minus2	ue(v)
log2_min_luma_coding_block_size_minus2	ue(v)
partition_constraints_override_enabled_flag	ue(v)
sps_log2_diff_min_qt_min_cb_intra_tile_group_luma	ue(v)
sps_log2_diff_min_qt_min_cb_inter_tile_group	ue(v)
sps_max_mtt_hierarchy_depth_inter_tile_groups	ue(v)
sps_max_mtt_hierarchy_depth_intra_tile_groups_luma	ue(v)
if(sps_max_mtt_hierarchy_depth_intra_tile_groups_luma != 0) {	
sps_log2_diff_max_bt_min_qt_intra_tile_group_luma	ue(v)
sps_log2_diff_max_tt_min_qt_intra_tile_group_luma	ue(v)
}	
if(sps_max_mtt_hierarchy_depth_inter_tile_groups != 0) {	
sps_log2_diff_max_bt_min_qt_inter_tile_group	ue(v)
sps_log2_diff_max_tt_min_qt_inter_tile_group	ue(v)
}	
if(qtbtt_dual_tree_intra_flag) {	
sps_log2_diff_min_qt_min_cb_intra_tile_group_chroma	ue(v)
sps_max_mtt_hierarchy_depth_intra_tile_groups_chroma	ue(v)
if (sps_max_mtt_hierarchy_depth_intra_tile_groups_chroma != 0) {	
sps_log2_diff_max_bt_min_qt_intra_tile_group_chroma	ue(v)
sps_log2_diff_max_tt_min_qt_intra_tile_group_chroma	ue(v)
}	
}	
sps_sao_enabled_flag	u(1)
sps_alf_enabled_flag	u(1)
pcm_enabled_flag	u(1)
if(pcm_enabled_flag) {	
pcm_sample_bit_depth_luma_minus1	u(4)
pcm_sample_bit_depth_chroma_minus1	u(4)
log2_min_pcm_luma_coding_block_size_minus3	ue(v)
log2_diff_max_min_pcm_luma_coding_block_size	ue(v)
pcm_loop_filter_disabled_flag	u(1)
}	
sps_ref_wraparound_enabled_flag	u(1)
if(sps_ref_wraparound_enabled_flag)	
sps_ref_wraparound_offset	ue(v)
sps_temporal_mvp_enabled_flag	u(1)
if(sps_temporal_mvp_enabled_flag)	
sps_sbtmvp_enabled_flag	u(1)
sps_amvr_enabled_flag	u(1)
sps_bdof_enabled_flag	u(1)
sps_cclm_enabled_flag	u(1)
sps_mts_intra_enabled_flag	u(1)
sps_mts_inter_enabled_flag	u(1)
sps_affine_enabled_flag	u(1)
if(sps_affine_enabled_flag)	
sps_affine_type_flag	u(1)
sps_gbi_enabled_flag	u(1)
sps_cpr_enabled_flag	u(1)
sps_ciip_enabled_flag	u(1)
sps_triangle_enabled_flag	u(1)

sps_ladf_enabled_flag	u(1)
if (sps_ladf_enabled_flag) {	
sps_num_ladf_intervals_minus2	u(2)
sps_ladf_lowest_interval_qp_offset	se(v)
for(i = 0; i < sps_num_ladf_intervals_minus2 + 1; i++) {	
sps_ladf_qp_offset[i]	se(v)
sps_ladf_delta_threshold_minus1[i]	ue(v)
}	
}	
sps_reshaper_enabled_flag	*u(1)*
rbsp_trailing_bits( )	
}	

7.3.3.1에서 일반 타일 그룹 헤더 신택스는 다음과 같이 밑줄 친 굵은 기울임꼴 텍스트의 삽입에 의해 수정될 수 있다.

표 5

tile_group_header( ) {	설명어
tile_group_pic_parameter_set_id	ue(v)
if(NumTilesInPic > 1) {	
tile_group_address	u(v)
num_tiles_in_tile_group_minus1	ue(v)
}	
tile_group_type	ue(v)
tile_group_pic_order_cnt_lsb	u(v)
if(partition_constraints_override_enabled_flag) {	
partition_constraints_override_flag	ue(v)
if(partition_constraints_override_flag) {	
tile_group_log2_diff_min_qt_min_cb_luma	ue(v)
tile_group_max_mtt_hierarchy_depth_luma	ue(v)
if(tile_group_max_mtt_hierarchy_depth_luma != 0)	
tile_group_log2_diff_max_bt_min_qt_luma	ue(v)
tile_group_log2_diff_max_tt_min_qt_luma	ue(v)
}	
if(tile_group_type = = I && qtbtt_dual_tree_intra_flag) {	
tile_group_log2_diff_min_qt_min_cb_chroma	ue(v)
tile_group_max_mtt_hierarchy_depth_chroma	ue(v)
if(tile_group_max_mtt_hierarchy_depth_chroma != 0)	
tile_group_log2_diff_max_bt_min_qt_chroma	ue(v)
tile_group_log2_diff_max_tt_min_qt_chroma	ue(v)
}	
}	
}	
}	
if (tile_group_type != I) {	
if(sps_temporal_mvp_enabled_flag)	
tile_group_temporal_mvp_enabled_flag	u(1)
if(tile_group_type = = B)	
mvd_l1_zero_flag	u(1)
if(tile_group_temporal_mvp_enabled_flag) {	
if(tile_group_type = = B)	
collocated_from_l0_flag	u(1)
}	

six_minus_max_num_merge_cand	ue(v)
if(sps_affine_enable_flag)	
five_minus_max_num_subblock_merge_cand	ue(v)
}	
tile_group_qp_delta	se(v)
if(pps_tile_group_chroma_qp_offsets_present_flag) {	
tile_group_cb_qp_offset	se(v)
tile_group_cr_qp_offset	se(v)
}	
if(sps_sao_enabled_flag) {	
tile_group_sao_luma_flag	u(1)
if(ChromaArrayType != 0)	
tile_group_sao_chroma_flag	u(1)
}	
if(sps_alf_enabled_flag) {	
tile_group_alf_enabled_flag	u(1)
if(tile_group_alf_enabled_flag)	
alf_data( )	
}	
if(tile_group_type = = P | | tile_group_type = = B) {	
num_ref_idx_l0_active_minus1	ue(v)
if(tile_group_type = = B)	
num_ref_idx_l1_active_minus1	ue(v)
}	
dep_quant_enabled_flag	u(1)
if(!dep_quant_enabled_flag)	
sign_data_hiding_enabled_flag	u(1)
if(deblocking_filter_override_enabled_flag)	
deblocking_filter_override_flag	u(1)
if(deblocking_filter_override_flag) {	
tile_group_deblocking_filter_disabled_flag	u(1)
if(!tile_group_deblocking_filter_disabled_flag) {	
tile_group_beta_offset_div2	se(v)
tile_group_tc_offset_div2	se(v)
}	
}	
if(num_tiles_in_tile_group_minus1 > 0) {	
offset_len_minus1	ue(v)
for(i = 0; i < num_tiles_in_tile_group_minus1; i++)	
entry_point_offset_minus1[i]	u(v)
}	
if (sps_reshaper_enabled_flag) {	
tile_group_reshaper_model_present_flag	*u(1)*
if (tile_group_reshaper_model_present_flag)	
tile_group_reshaper_model ( )	
tile_group_reshaper_enable_flag	*u(1)*
if (tile_group_reshaper_enable_flag && (!(qtbtt_dual_tree_intra_flag && tile_group_type == I)))	
tile_group_reshaper_chroma_residual_scale_flag	*u(1)*
}	
byte_alignment( )	
}	

새로운 신택스 테이블 타일 그룹 변형 모델은 다음과 같이 추가될 수 있다:

표 6

tile_group_reshaper_model ( ) {	설명어
reshaper_model_min_bin_idx	ue(v)
reshaper_model_delta_max_bin_idx	ue(v)
reshaper_model_bin_delta_abs_cw_prec_minus1	ue(v)
for (i = reshaper_model_min_bin_idx; i <= reshaper_model_max_bin_idx; i++) {	
reshape_model_bin_delta_abs_CW [i]	u(v)
if (reshaper_model_bin_delta_abs_CW[i]) > 0)	
reshaper_model_bin_delta_sign_CW_flag[i]	u(1)
}	
}	

일반 시퀀스 파라미터 세트 RBSP 의미론(semantics)에서 다음 의미론이 추가될 수 있다:1과 동일한 **sps_reshaper_enabled_flag**는 변형 변수가 코딩된 비디오 시퀀스(coded video sequence)(CVS)에서 사용되는 것을 명시한다. sps_reshaper_enabled_flag가 0이면 변형 변수가 CVS에서 사용되지 않음을 명시한다.

타일 그룹 헤더 신택스에서, 다음 의미론이 추가될 수 있다:

1과 동일한 **tile_group_reshaper_model_present_flag**는 tile_group_reshaper_model()이 타일 그룹 헤더에 존재함을 명시한다. 0과 동일한 tile_group_reshaper_model_present_flag는 tile_group_reshaper_model()이 타일 그룹 헤더에 존재하지 않음을 명시한다. tile_group_reshaper_model_present_flag가 존재하지 않는 경우 0과 동일한 것으로 유추된다.

1과 동일한 **tile_group_reshaper_enabled_flag**는 현재 타일 그룹에 대해 변형 변수가 활성화됨을 명시한다. tile_group_reshaper_enabled_flag가 0이면 현재 타일 그룹에 대해 변형 변수가 활성화되지 않음을 명시한다. tile_group_reshaper_enable_flag가 존재하지 않는 경우 0과 동일한 것으로 유추된다.

tile_group_reshaper_model() 신택스는 다음과 같이 추가될 수 있다:

reshape_model_min_bin_idx는 변형 변수 구성 프로세스에서 사용될 최소 빈(또는 조각) 인덱스를 명시한다. reshape_model_min_bin_idx의 값은 0에서 MaxBinIdx까지의 범위에 있어야 한다. MaxBinIdx의 값은 15와 같아야 한다.

reshape_model_delta_max_bin_idx는 최대 허용 빈(또는 조각) 인덱스 MaxBinIdx에서 변형 변수 구성 프로세스에 사용할 최대 빈 인덱스를 뺀 값을 명시한다. reshape_model_max_bin_idx의 값은 MaxBinIdx - reshape_model_delta_max_bin_idx와 동일하게 설정된다.

reshaper_model_bin_delta_abs_cw_prec_minus1 더하기 1은 신택스 reshape_model_bin_delta_abs_CW[i]의 표현에 사용되는 비트 수를 명시한다.

reshape_model_bin_delta_abs_CW[i]는 i번째 빈에 대한 절대 델타 코드워드 값을 명시한다.

reshaper_model_bin_delta_sign_CW_flag[i]는 다음과 같이 reshape_model_bin_delta_abs_CW[i]의 부호를 명시한다:

- reshape_model_bin_delta_sign_CW_flag[i]가 0이면 대응하는 변수 RspDeltaCW[i]는 양수 값이다.

- 그렇지 않으면(reshape_model_bin_delta_sign_CW_flag[i]가 0이 아님), 대응하는 변수 RspDeltaCW[i]는 음수 값이다.

reshape_model_bin_delta_sign_CW_flag[i]가 존재하지 않는 경우, 0과 동일한 것으로 추론된다.

변수 RspDeltaCW[i] = (1 2*reshape_model_bin_delta_sign_CW [i]) * reshape_model_bin_delta_abs_CW [i];

변수 RspCW[i]는 다음 단계들에 따라 도출된다:

변수 OrgCW는 (1 << BitDepthY)/(MaxBinIdx + 1)과 동일하게 설정된다.

- reshaper_model_min_bin_idx < = i <= reshaper_model_max_bin_idx인 경우,

RspCW[i] = OrgCW + RspDeltaCW[i]이고,

- Otherwise, RspCW[i] = 0

BitDepthY의 값이 10인 경우 RspCW[i]의 값은 32 내지 2 * OrgCW - 1의 범위에 있을 것이다.

0에서 MaxBinIdx + 1(포함)의 범위에 있는 i를 갖는 변수 InputPivot[i]는 다음과 같이 도출된다.

InputPivot[i] = i * OrgCW

0에서 MaxBinIdx + 1(포함)의 범위에 있는 i를 갖는 변수 ReshapePivot[i], 0에서 MaxBinIdx(포함)의 범위에 있는 i를 갖는 변수 ScaleCoef[i] 및 InvScaleCoeff[i]는 다음과 같이 도출된다:

shiftY = 14

ReshapePivot[0] = 0;

for(i = 0; i <= MaxBinIdx ; i++) {

ReshapePivot[i + 1] = ReshapePivot[i] + RspCW[i]

ScaleCoef[i] = (RspCW[i] * (1 << shiftY) + (1 << (Log2(OrgCW) - 1))) >> (Log2(OrgCW))

if (RspCW[i] == 0)

InvScaleCoeff[i] = 0

else

InvScaleCoeff[i] = OrgCW * (1 << shiftY) / RspCW[i]

}

0에서 MaxBinIdx(포함)의 범위에 있는 i를 갖는 변수 ChromaScaleCoef[i]는 다음과 같이 도출될 수 있다:

ChromaResidualScaleLut[64] = {16384, 16384, 16384, 16384, 16384, 16384, 16384, 8192, 8192, 8192, 8192, 5461, 5461, 5461, 5461, 4096, 4096, 4096, 4096, 3277, 3277, 3277, 3277, 2731, 2731, 2731, 2731, 2341, 2341, 2341, 2048, 2048, 2048, 1820, 1820, 1820, 1638, 1638, 1638, 1638, 1489, 1489, 1489, 1489, 1365, 1365, 1365, 1365, 1260, 1260, 1260, 1260, 1170, 1170, 1170, 1170, 1092, 1092, 1092, 1092, 1024, 1024, 1024, 1024};

shiftC = 11

- if (RspCW[i] == 0)

ChromaScaleCoef [i] = (1 << shiftC)

- Otherwise (RspCW[i] != 0),

ChromaScaleCoef[i] = ChromaResidualScaleLut[RspCW[i] >> 1]

병합 및 인트라 예측을 위한 가중 샘플 예측 프로세스와 관련하여 다음 텍스트를 추가할 수 있다. 추가는 밑줄이 그어진 기울임꼴로 표시된다.

8.4.6.6 병합 및 인트라 예측 결합을 위한 가중 샘플 예측 프로세스

이 프로세스에 대한 입력은 다음과 같다:

- 현재 코딩 블록의 너비 cbWidth,

- 현재 코딩 블록의 높이 cbHeight,

- 두 개의 (cbWidth)x(cbHeight) 배열들 predSamplesInter 및 predSamplesIntra,

- 인트라 예측 모드 predModeIntra,

- 색상 구성요소 인덱스를 명시하는 변수 cIdx.

이 프로세스의 출력은 예측 샘플 값들의 (cbWidth)x(cbHeight) 배열 predSamplesComb이다.

변수 bitDepth는 다음과 같이 도출된다.

- cIdx가 0이면 bitDepth는 BitDepthY와 동일하게 설정된다.

- 그렇지 않으면 bitDepth는 BitDepthC와 동일하게 설정된다.

x = 0..cbWidth - 1 및 y = 0..cbHeight - 1인 예측 샘플들 predSamplesComb[x][y]는 다음과 같이 도출된다.

- 가중치 w는 다음과 같이 도출된다.

 - predModeIntra가 INTRA_ANGULAR50인 경우 w는 y와 동일한 nPos 및 cbHeight와 동일한 nSize로 표 7에 명시된다.

 - 그렇지 않고 predModeIntra가 INTRA_ANGULAR18이면 w는 nPos가 x와 같고 nSize가 cbWidth와 같은 표 7에 명시된다

 - 그렇지 않으면 w는 4로 설정된다.

- cIdx가 0이면 predSamplesInter는 다음과 같이 도출된다.

- tile_group_reshaper_enabled_flag가 1인 경우,

shiftY = 14

idxY = predSamplesInter[x][y] >> Log2(OrgCW)

predSamplesInter [x][y] = Clip1Y (ReshapePivot[idxY]

*+ (ScaleCoeff[idxY] *(predSamplesInter[x][y] InputPivot[idxY])*

+ (1 << (shiftY - 1)) >> shiftY) *(8 xxx)*

- 그렇지 않은 경우(tile_group_reshaper_enabled_flag는 0과 같음)

predSamplesInter [x][y] = predSamplesInter [x][y]

- 예측 샘플 predSamplesComb[x][y]는 다음과 같이 도출된다.

predSamplesComb[x][y] = (w * predSamplesIntra[x][y] + (8 740)

 (8 - w) * predSamplesInter[x][y]) >> 3)

표 7

0 <= nP < (nS / 4)	(nS / 4) <= nP < (nS / 2)	(nS / 2) <= nP < (3 *nS / 4)	(3 *nS / 4) <= nP < nS
6	5	3	2

표 7 - 위치 nP와 크기 nS의 함수로서의 w의 명시밑줄 굵게 기울임꼴로 표시된 다음 텍스트는 영상 재구성 프로세스에서 추가될 수 있다.

8.5.5 영상 재구성 프로세스

이 프로세스에 대한 입력은 다음과 같다.

- 현재 영상 구성요소의 왼쪽 상단 샘플에 상대적인 현재 블록의 왼쪽 상단 샘플을 명시하는 위치(xCurr, yCurr),

- 현재 블록의 너비와 높이를 각각 명시하는 변수들 nCurrSw 및 nCurrSh,

- 현재 블록의 색상 구성요소를 명시하는 변수 cIdx,

- 현재 블록의 예측 샘플들을 명시하는 (nCurrSw)x(nCurrSh) 배열 predSamples,

- 현재 블록의 잔차 샘플들을 명시하는 (nCurrSw)x(nCurrSh) 배열 resSamples.

색상 구성요소 cIdx의 값에 따라 다음이 할당된다:

- cIdx가 0이면 recSamples는 재구성된 영상 샘플 어레이 SL에 대응하고 함수 clipCidx1은 Clip1Y에 대응한다.

- 그렇지 않고 cIdx가 1이면 recSamples는 재구성된 채도 샘플 어레이 SCb에 대응하고 함수 clipCidx1은 Clip1C에 대응한다.

- 그렇지 않으면(cIdx가 2임) recSamples는 재구성된 채도 샘플 어레이 SCr에 대응하고 함수 clipCidx1은 Clip1C에 대응한다.

tile_group_reshaper_enabled_flag의 값이 1과 같을 때 위치(xCurr, yCurr)에서 재구성된 샘플 어레이 recSamples의 (nCurrSw)x(nCurrSh) 블록은 8.5.5.1절에 명시된 매핑 프로세스로 도출된다. 그렇지 않으면 위치(xCurr, yCurr)에서 재구성된 샘플 배열 recSamples의 (nCurrSw)x(nCurrSh) 블록은 다음과 같이 도출된다.

recSamples[xCurr + i][yCurr + j] = clipCidx1(predSamples[i][j] + resSamples[i][j]) (8 xxx)

with i = 0..nCurrSw - 1, j = 0..nCurrSh - 1

8.5.5.1 매핑 프로세스를 통한 영상 재구성

이 절은 매핑 프로세스로 영상 재구성을 명시한다. 휘도 샘플 값에 대한 매핑 프로세스를 사용한 영상 재구성은 8.5.5.1.1에 명시되어 있다. 채도 샘플 값에 대한 매핑 프로세스를 사용한 영상 재구성은 8.5.5.1.2에 명시되어 있다.

8.5.5.1.1 휘도 샘플 값에 대한 매핑 프로세스를 통한 영상 재구성

이 프로세스에 대한 입력들은 다음과 같다:

- 현재 블록의 휘도 예측 샘플들을 명시하는 (nCurrSw)x(nCurrSh) 배열 predSamples,

- 현재 블록의 휘도 잔차 샘플들을 명시하는 (nCurrSw)x(nCurrSh) 배열 resSamples.

이 프로세스의 출력은 다음과 같다:

- (nCurrSw)x(nCurrSh) 매핑된 휘도 예측 샘플 어레이 predMapSamples,

- (nCurrSw)x(nCurrSh) 재구성된 휘도 샘플 어레이 recSamples.

predMapSamples는 다음과 같이 도출된다:

- If (CuPredMode[xCurr][yCurr] = = MODE_INTRA) || (CuPredMode[xCurr][yCurr] = = MODE_CPR) || (CuPredMode[xCurr][yCurr] = = MODE_INTER && mh_intra_flag[xCurr][yCurr])

predMapSamples[xCurr + i][yCurr + j] = predSamples[i][j] (8 xxx)

with i = 0..nCurrSw - 1, j = 0..nCurrSh - 1

- otherwise ((CuPredMode[xCurr][yCurr] = = MODE_INTER && !mh_intra_flag[xCurr][yCurr])) 다음이 적용된다.

shiftY = 14

idxY = predSamples[i][j] >> Log2(OrgCW)

predMapSamples[xCurr + i][yCurr + j] = ReshapePivot[idxY]

+ (ScaleCoeff[idxY] *(predSamples[i][j] InputPivot[idxY])

+ (1 << (shiftY - 1))) >> shiftY (8 xxx)

with i = 0..nCurrSw - 1, j = 0..nCurrSh - 1

recSamples는 다음과 같이 도출된다.

recSamples[xCurr + i][yCurr + j] = Clip1Y (predMapSamples[xCurr + i][yCurr + j]+ resSamples[i][j]) (8 xxx)

i = 0..nCurrSw - 1, j = 0..nCurrSh - 1

8.5.5.1.2 채도 샘플 값에 대한 매핑 프로세스를 통한 영상 재구성

이 프로세스에 대한 입력들은 다음과 같다.

- 현재 블록의 매핑된 휘도 예측 샘플들을 명시하는 (nCurrSwx2)x(nCurrShx2) 배열 매핑된 predMapSamples,

- 현재 블록의 채도 예측 샘플들을 명시하는 (nCurrSw)x(nCurrSh) 배열 predSamples,

- (nCurrSw)x(nCurrSh) 배열 resSamples는 현재 블록의 채도 잔차 샘플들을 명시한다.

이 프로세스의 출력은 재구성된 채도 샘플 어레이 recSamples이다.

recSamples는 다음과 같이 도출된다:

- If (!tile_group_reshaper_chroma_residual_scale_flag || ((nCurrSw)x(nCurrSh) <= 4))

recSamples[xCurr + i][yCurr + j] = Clip1C (predSamples[i][j] + resSamples[i][j]) (8 xxx)

with i = 0..nCurrSw - 1, j = 0..nCurrSh - 1

- otherwise (tile_group_reshaper_chroma_residual_scale_flag && ((nCurrSw)x(nCurrSh) > 4)), 다음이 적용된다.

변수 varScale은 다음과 같이 도출된다.

1. invAvgLuma = Clip1Y((S_iS_j predMapSamples[(xCurr << 1) + i][(yCurr << 1) + j] + nCurrSw * nCurrSh *2) / (nCurrSw * nCurrSh *4))

2. 변수 idxYInv는 샘플 값 invAvgLuma의 입력과 함께 *8.5.6.2절*에 명시된 대로 조각별 함수 인덱스의 식별을 포함하여 도출된다.

3. varScale = ChromaScaleCoef[idxYInv]

recSamples는 다음과 같이 도출된다.

- If tu_cbf_cIdx [xCurr][yCurr] equal to 1, the following applies:

shiftC = 11

recSamples[xCurr + i][yCurr + j] = ClipCidx1 (predSamples[i][j] + Sign(resSamples[i][j])

* ((Abs(resSamples[i][j]) * varScale + (1 << (shiftC - 1))) >> shiftC)) (8 xxx)

with i = 0..nCurrSw - 1, j = 0..nCurrSh - 1

- Otherwise (tu_cbf_cIdx[xCurr][yCurr] equal to 0)

recSamples[xCurr + i][yCurr + j] = ClipCidx1(predSamples[i][j]) (8 xxx)

with i = 0..nCurrSw - 1, j = 0..nCurrSh - 1

8.5.6 영상 역 매핑 프로세스

이 절은 tile_group_reshaper_enabled_flag의 값이 1일 때 적용된다(invoked). 입력은 재구성된 영상 휘도 샘플 어레이 SL이고 출력은 역 매핑 프로세스 후에 수정된 재구성된 영상 휘도 샘플 어레이 S'L이다.

휘도 샘플 값에 대한 역 매핑 프로세스는 8.4.6.1에 명시되어 있다.

8.5.6.1 휘도 샘플 값들의 영상 역 매핑 프로세스

이 프로세스에 대한 입력들은 현재 영상의 왼쪽 상단 휘도 샘플에 상대적인 휘도 샘플 위치를 명시하는 휘도 위

치(xP, yP)이다.

이 프로세스의 출력은 역 매핑된 휘도 샘플 값 invLumaSample이다.

invLumaSample의 값은 다음과 같은 순서화된 단계들을 적용하여 도출된다.

1. 변수 idxYInv는 휘도 샘플 값 SL[xP][yP]의 입력으로 8.5.6.2절에 명시된 바와 같이 조각별 함수 인덱스의 식별을 적용하여 도출된다.

2. reshapeLumaSample의 값은 다음과 같이 도출된다.

shiftY = 14

invLumaSample = InputPivot[idxYInv] + (InvScaleCoeff[idxYInv] *(SL[xP][yP] ReshapePivot[idxYInv]) + (1 << (shiftY - 1))) >> shifty (8 xx)

3. clipRange = ((reshape_model_min_bin_idx > 0) && (reshape_model_max_bin_idx < MaxBinIdx));

- clipRange가 1이면 다음이 적용된다:

minVal = 16 << (BitDepthY - 8)

maxVal = 235<< (BitDepthY - 8)

invLumaSample = Clip3(minVal, maxVal, invLumaSample)

- else(clipRange는 0과 같음),

invLumaSample = ClipCidx1(invLumaSample),

8.5.6.2 휘도 구성요소들에 대한 조각별 함수 인덱스 식별

이 프로세스에 대한 입력들은 휘도 샘플 값 S이다.

이 프로세스의 출력은 샘플 S가 속한 조각을 식별하는 인덱스 iS이다. 변수 idxS는 다음과 같이 도출된다:

```
for( idxS = 0, idxFound = 0; idxS <= MaxBinIdx; idxS++ ) {
    if( (S < ReshapePivot[ idxS + 1 ] ) {
        idxFound = 1
        break
    }
}
```

ID idxS를 찾는 대체 구현은 다음과 같다:

```
if (S < ReshapePivot[ reshape_model_min_bin_idx ])
    idxS = 0
else if (S >= ReshapePivot[ reshape_model_max_bin_idx ])
    idxS  = MaxBinIdx
else
    idxS = findIdx ( S, 0, MaxBinIdx + 1, ReshapePivot[ ] )

function idx = findIdx (val, low, high, pivot[ ]) {
    if ( high – low <= 1 )
        idx = low
    else {
        mid = ( low + high) >> 1
        if (val <  pivot [mid] )
            high = mid
        else
            low = mid
        idx = findIdx (val, low, high, pivot[])
    }
}
```

2.5.2.4. ILR의 사용

인코더 측에서 각 영상(또는 타일 그룹)은 먼저 변형된 도메인으로 전환된다. 그리고 전체 코딩 프로세스는 변형된 도메인에서 수행된다. 인트라 예측의 경우 이웃한 블록은 변형된 도메인에 있다. 상호 예측을 위해, (디코딩된 영상 버퍼로부터 원본 도메인으로부터 생성된) 참조 블록들이 먼저 변형된 도메인으로 전환된다. 그런 다음 잔차가 생성되어 비트스트림으로 코딩된다.

전체 영상(또는 타일 그룹)가 인코딩/디코딩을 완료한 후 변형된 도메인의 샘플들을 원본 도메인으로 전환한 다음 디블로킹 필터(deblocking filter) 및 기타 필터들을 적용한다.

예측 신호에 대한 순방향 변형은 다음과 같은 경우들에 비활성화된다.

- 현재 블록이 인트라 코딩된다.

- 현재 블록은 CPR(현재 영상 참조, 일명 인트라 블록 복사(intra block copy)(IBC))로 코딩된다.

- 현재 블록이 결합된 인트라-인터(상호) 모드(Combined Inter-Intra Mode) (CIIP)로 코딩되고 인트라 예측 블록에 대해 순방향 변형이 비활성화된다.

2.6. 가상 파이프라이닝 데이터 유닛들(Virtual Pipelining Data Units)(VPDU)

가상 파이프라이닝 데이터 유닛(VPDU)들은 영상에서 비중첩(non-overlapping) MxM-luma(L)/NxN-chroma(C) 유닛들로 정의된다. 하드웨어 디코더에서 연속적인 VPDU들은 동시에 여러 파이프라인 단계들에서 처리된다; 다른 단계들은 다른 VPDU들을 동시에 처리한다. VPDU 크기는 대부분의 파이프라인 단계들에서 버퍼 크기에 대략 비례하므로 VPDU 크기를 작게 유지하는 것이 매우 중요하다고 한다. HEVC 하드웨어 디코더들에서 VPDU 크기는 최대 변환 블록(transform block)(TB) 크기로 설정된다. 최대 TB 크기를 32x32-L/16x16-C(HEVC에서와 같이)에서부터 64x64-L/32x32-C(현재 VVC에서와 같이)로 확대하면 코딩 이득(gain)들을 가져올 수 있고, HEVC와 비교하여 예상되는 VPDU 크기(64x64-L/32x32-C)의 4배이다. 그러나 쿼드트리(QT) 코딩 유닛(CU) 분할 외에 추가적인 코딩 이득들을 달성하기 위해 VVC에서 상황 트리(Ternary Tree)(TT) 및 이진 트리(Binary Tree)(BT) 분할을 적용할 수 있고, 128x128-L/64x64-C 코딩 트리 블록(coding tree block)(CTU)들을 재귀적으로 사용하므로 HEVC에 비해 VPDU 크기(128x128-L/64x64-C)가 16배 증가한다고 한다.

VVC의 현재 설계에서, VPDU 크기는 64x64-L/32x32-C로 정의된다.

2.7. APS

적응 파라미터 세트(Adaptation Parameter Set)(APS)는 ALF 파라미터들을 전달하기 위해 VVC에서 채택된다. 타일 그룹 헤더에는 ALF가 활성화되는 경우, 조건부로 존재하는 aps_id가 포함된다. APS에는 aps_id 및 ALF 파라미터가 포함되어 있다. 새로운 NUT(AVC 및 HEVC에서와 같은 NAL 유닛 유형) 값이 APS에 할당된다(JVET-M0132으로부터). VTM-4.0의 공통 테스트 조건들은 (나타나기 위해) aps_id = 0을 사용하고 각 영상과 함께 APS를 보내는 것이 좋다. 현재 APS ID 값의 범위는 0..31이고 APS는 영상 간에 공유될 수 있다(영상 내의 다른 타일 그룹들에서 다를 수 있음). ID 값이 있는 경우 고정 길이(fixed-length)로 코딩되어야 한다. 동일한 영상 내에서 다른 콘텐츠와 ID 값들을 재사용할 수 없다.

2.8. 재구성 후 필터들(Post-reconstruction filters)

2.8.1 확산 필터(diffusion filter)(DF)

JVET-L0157에서, CU의 인트라/인터(상호) 예측 신호가 확산 필터들에 의해 추가로 수정될 수 있는 확산 필터가 제안된다.

2.8.1.1. 균일 확산 필터(Uniform Diffusion Filter)

균일 확산 필터는 아래에 정의된 h^{I} 또는 h^{IV}로 명시되는 고정된 마스크로 예측 신호를 변환하여 구현된다.

예측 신호 자체 외에도, 블록의 왼쪽 및 위의 재구성된 샘플들의 한 라인이 필터링된 신호에 대한 입력으로 사용되고, 여기서 이러한 재구성된 샘플들의 사용은 블록 간에서 회피될 수 있다.

pred를 인트라 또는 모션 보상 예측에 의해 획득된 주어진 블록에 대일 예측 신호라고 하자. 필터의 경계점(boundary point)들을 처리하기 위해서는 예측 신호를 예측 신호 pred_ext로 확장해야 한다. 이 확장된 예측은 두 가지 방법으로 형성될 수 있다: 중간 단계(intermediate step)로서 블록 왼쪽과 위쪽에 재구성된 샘플들의

한 라인이 예측 신호에 추가된 다음 결과 신호가 전체 방향들로 미러링된다. 또는 예측 신호 자체만 전체 방향들로 미러링된다. 후자의 확장은 상호 블록들에 사용된다. 이 경우, 예측 신호 자체만이 확장된 예측 신호 pred_ext에 대한 입력을 포함한다.

필터 h^I가 사용되는 경우, 앞서 언급한 경계 확장을 사용하여 예측 신호 pred를 $h^I * pred$로 대체하는 것이 제안된다. 여기에서 필터 마스크 h^I는 다음과 같이 주어진다.

$$h^I = (0.25)^4 \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 6 & 0 & 16 & 0 & 6 & 0 & 0 \\ 0 & 4 & 0 & 24 & 0 & 24 & 0 & 4 & 0 \\ 1 & 0 & 16 & 0 & 36 & 0 & 16 & 0 & 1 \\ 0 & 4 & 0 & 24 & 0 & 24 & 0 & 4 & 0 \\ 0 & 0 & 6 & 0 & 16 & 0 & 6 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

필터 h^{IV}가 사용되는 경우 pred 예측 신호를 $h^{IV} * pred$로 대체하는 것이 제안된다.

여기서 필터 h^{IV}는 $h^{IV} = h^I * h^I * h^I * h^I$로 주어진다.

2.8.1.2. 방향성 확산 필터(Directional Diffusion Filter)

신호 적응형 확산 필터들을 사용하는 대신, 방향성 필터들인 수평 필터 h^{hor} 및 수직 필터 h^{ver}는 여전히 고정된 마스크를 가지고 사용된다. 보다 정확하게는, 이전 섹션의 마스크 h^I에 대응하는 균일 확산 필터링은 수직 또는 수평 방향으로만 적용되도록 간단히 제한된다. 수직 필터는 예측 신호에 고정된 필터 마스크

$$h_{ver} = (0.5)^4 \begin{pmatrix} 1 \\ 0 \\ 4 \\ 0 \\ 6 \\ 0 \\ 4 \\ 0 \\ 1 \end{pmatrix}$$

를 적용하여 구현하고 수평 필터는 전치된 마스크(transposed mask) $h_{hor} = h_{ver}^t$를 사용하여 구현된다.

2.8.2. 양방향 필터(Bilateral filter)(BF)

양방향 필터는 JVET-L0406에서 제안되며 0이 아닌 변환 계수들과 슬라이스 양자화 파라미터가 17보다 큰 휘도 블록들에 항상 적용된다. 따라서 양방향 필터의 사용을 시그널링할 필요가 없다. 양방향 필터가 적용된 경우 역변환 직후 디코딩된 샘플에 대해 수행된다. 또한, 필터 파라미터들, 즉 가중치들은 코딩된 정보에서 명확하게 도출된다.

필터링 프로세스는 다음과 같이 정의된다.

수학식 6

$$P'_{0,0} = P_{0,0} + \sum_{k=1}^{K} W_k\left(abs(P_{k,0} - P_{0,0})\right) \times \left(P_{k,0} - P_{0,0}\right)$$

$P_{0,0}$은 현재 샘플의 강도이고 $P'_{0,0}$는 현재 샘플의 수정된 강도이며 $P_{k,0}$ 및 W_k는 각각 k-번째 이웃한 샘플에 대한 강도 및 가중치 파라미터이다. 하나의 현재 샘플과 그것의 4개의 이웃한 샘플들(즉, K=4)의 예가 도 24에 도시되어 있다.

보다 구체적으로, k-번째 이웃한 샘플과 연관된 가중치 $W_k(x)$는 다음과 같이 정의된다.

수학식 7

$$W_k(x) = Distance_k \times Range_k(x)$$

여기서, $Distance_k$는 다음과 같고,

수학식 8

$$Distance_k = e^{\left(-\frac{10000}{2\sigma_d^2}\right)} \Big/ 1 + 4 * e^{\left(-\frac{10000}{2\sigma_d^2}\right)}$$

$Range_k(x)$는 다음과 같으며,

수학식 9

$$Range_k(x) = e^{\left(-\frac{x^2}{8*(QP-17)*(QP-17)}\right)}$$

σ_d는 코딩된 모드 및 코딩 블록 크기에 의존한다. 설명된 필터링 프로세스는 인트라 코딩된 블록에 적용되고 TU가 추가로 분할될 때 상호 코딩된 블록에 적용되어 병렬 처리가 가능하다.

비디오 신호의 통계적 특성을 더 잘 포착하고 필터의 성능을 향상시키기 위해, 수학식 7에서 나온 가중치 함수는 코딩 모드 및 블록 분할의 파라미터(최소 크기)에 따라 표 8에 표로 표시된 σ_d 파라미터에 의해 조정된다.

표 8

Min (블록 너비, 블록 높이)	인트라 모드	상호 모드
4	82	62
8	72	52
기타	52	32

표 8 다양한 블록 크기 및 코딩 모드들에 대한 σ_d 값TU가 분할되지 않는 경우, 상호 코딩된 블록들에 대해 코딩 성능을 더욱 향상시키기 위해, 현재 샘플과 이웃한 샘플들 중 하나 사이의 강도 차이는 현재 샘플과 이웃한 샘플을 포함하는 두 창들 사이의 대표적인 강도 차이로 대체된다. 따라서 필터링 프로세스 방정식은 다음과 같이 수정된다:

수학식 10

$$P'_{0,0} = P_{0,0} + \sum_{k=1}^{N} W_k \left(\frac{1}{M}\sum_{m=-M/2}^{M/2} abs(P_{k,m} - P_{0,m})\right) \times (P_{k,0} - P_{0,0})$$

여기서 $P_{k,m}$ 및 $P_{0,m}$은 각각 $P_{k,0}$ 및 $P_{0,0}$을 중심으로 하는 창들 내 m번째 샘플 값을 나타낸다. 이 제안에서 창 크기는 3X3으로 설정된다. $P_{2,0}$ 및 $P_{0,0}$을 포함하는 두 개의 창들의 예가 도 25에 도시되어 있다.

2.8.3. 아다마르 변환 도메인 필터(Hadamard transform domain filter)(HF)

JVET-K0068에서 재구성 후 CU 레벨에 적용되고 곱셈이 없는 구현이 있는 1D 아다마르 변환 도메인의 인-루프 필터이다. 제안된 필터는 미리 정의된 조건을 충족하는 전체 CU 블록들에 적용되고 필터 파라미터는 코딩된 정보로부터 도출된다.

제안된 필터링은 4x4 블록들과 슬라이스 양자화 파라미터가 17보다 큰 경우를 제외하고 변환 계수들이 0이 아닌 휘도 재구성된 블록들에 항상 적용된다. 필터 파라미터는 코딩된 정보로부터 명확하게 도출된다. 제안된 필터가 적용된 경우 역변환 직후 디코딩된 샘플들에 대해 수행된다.

재구성된 블록 픽셀 처리의 각 픽셀에 대해 다음 단계들을 포함한다:

* 스캔 패턴에 따라 현재 하나를 포함하여 처리 픽셀 이웃한 4개의 인접 픽셀들을 스캔한다.

* 읽기 픽셀들(read pixels)의 4포인트 아다마르 변환

* 다음 공식에 기초한 스펙트럼 필터링:

수학식 11

$$F(i,\sigma) = \frac{R(i)^2}{R(i)^2 + \sigma^2} * R(i)$$

여기서 (i)는 아다마르 스펙트럼에서 스펙트럼 구성요소의 인덱스이고, R(i)는 인덱스에 대응하는 재구성된 픽셀의 스펙트럼 구성요소고, σ는 다음 식을 사용하여 코덱 양자화 파라미터 QP에서 도출된 필터링 파라미터이다.

수학식 12

$$\sigma = 2^{(1+0.126*(QP-27))}$$

스캔 패턴의 예는 도 26에 도시되어 있다.

CU 경계에 놓인 픽셀의 경우, 전체 필요한 픽셀들이 현재 CU 내에 있도록 스캔 패턴이 조정된다.

3. 기존 구현들의 단점들

ILR의 현재 설계에는 다음과 같은 문제들이 있을 수 있다.

1. 변형 모델 정보가 시퀀스에서 시그널링되지 않았을 수 있지만 tile_group_reshaper_enable_flag는 현재 슬라이스(또는 타일 그룹)에서 1과 동일하게 설정된다.

2. 저장된 변형 모델은 참조로 사용할 수 없는 슬라이스(또는 타일 그룹)에서 올 수 있다.

3. 하나의 영상은 여러 슬라이스(또는 타일 그룹)로 분할될 수 있으며, 각 슬라이스(또는 타일 그룹)는 변형 모델 정보를 시그널링할 수 있다.

4. 일부 값들과 범위들(예: RspCW [i]의 범위)는 비트 깊이가 10인 경우에만 정의된다.

5. reshape_model_delta_max_bin_idx가 잘 제한되지 않는다.

6. reshaper_model_bin_delta_abs_cw_prec_minus1이 잘 제한되지 않는다.

7. ReshapePivot[i]는 1<<BitDepth -1보다 클 수 있다.

8. 고정된 클리핑 파라미터들(즉, 최소값이 0이고 최대값이 (1 << BD) -1임)는 ILR의 사용을 고려하지 않고 사용된다. 여기서 BD는 비트 깊이를 나타낸다.

9. 채도 구성요소에 대한 변형 연산은 4:2:0 색상 형식만 고려한다.

10. 각 휘도 값이 상이하게 변형될 수 있는 휘도 구성요소와 달리, 채도 구성요소에 대해 하나의 인자만 선택되고 사용된다. 이러한 스케일링은 추가적인 복잡성을 줄이기 위해 양자화/역양자화 단계에 병합될 수 있다.

11. 영상 역 매핑 프로세스에서 클리핑은 상한과 하한을 별도로 고려할 수 있다.

12. i에 대한 ReshapePivot[i]가 reshaper_model_min_bin_idx < = i <= reshaper_model_max_bin_idx의 범위에 있지 않는다. reshaper_model_max_bin_idx가 제대로 설정되지 않았다.

4. 예시적인 실시예들 및 기술들

이하에서 설명하는 구체적인 실시예들은 일반적인 개념을 설명하기 위일 예들로서 고려되어야 한다. 이러한 실시예들은 좁게 해석되어서는 안 된다. 또한, 이들 실시예들은 임의의 방식으로 조합될 수 있다.

1. 시퀀스를 디코딩하기 전에 변형 모델을 초기화하는 것이 좋다.

a. 대안적으로, 재구성 모델은 I-슬라이스(또는 영상 또는 타일 그룹)를 디코딩하기 전에 초기화된다.

b. 대안적으로, 재구성 모델은 순간 디코딩 리프레시(Instantaneous Decoding Refresh)(IDR) 슬라이스(또는 영상 또는 타일 그룹)를 디코딩하기 전에 초기화된다.

c. 대안적으로, 재구성 모델은 클린 랜덤 액세스(clean random access)(CRA) 슬라이스(또는 영상 또는 타일 그룹)를 디코딩하기 전에 초기화된다.

d. 대안적으로, 재구성 모델은 인트라 랜덤 액세스 포인트(intra random access point)(I-RAP) 슬라이스(또는 영상 또는 타일 그룹)를 디코딩하기 전에 초기화된다. I-RAP 슬라이스들 (또는 영상들 또는 타일 그룹들)은 IDR 슬라이스들 (또는 영상들 또는 타일 그룹들) 및/또는 CRA 슬라이스들(또는 영상 또는 타일 그룹) 및/또는 끊어진 링크 액세스(broken link access)(BLA) 슬라이스들(또는 영상 또는 타일 그룹들)을 포함할 수 있다.

e. 변형 모델을 초기화하는 한 가지 예에서 OrgCW는 (1 << BitDepthY) / (MaxBinIdx + 1)과 동일하게 설정된다. ReshapePivot[i] = InputPivot [i] = i * OrgCW for i = 0, 1,..., MaxBinIdx;

f. 재구성 모델을 초기화하는 한 가지 예에서 ScaleCoef[i] = InvScaleCoeff[i]= 1 << shiftY for i = 0, 1,..., MaxBinIdx;

g. 변형 모델을 초기화하는 한 가지 예에서 OrgCW는 (1 << BitDepthY) / (MaxBinIdx + 1)과 동일하게 설정된다. RspCW[i] = i = 0, 1,..., MaxBinIdx에 대한 OrgCW.

h. 대안적으로, 기본 재구성 모델 정보는 시퀀스 레벨(예: SPS) 또는 영상 레벨(예: PPS)에서 시그널링될 수 있으며 재구성 모델은 기본 모델로 초기화된다.

i. 또는 변형 모델이 초기화되지 않은 경우 ILR이 비활성화되는 것으로 제한된다.

2. 재구성 모델 정보(예: tile_group_reshaper_model()의 정보)는 I-슬라이스(또는 영상 또는 타일 그룹)에서만 시그널링될 수 있다고 제안된다.

a. 대안적으로, 재구성 모델 정보(예: tile_group_reshaper_model()의 정보)는 IDR-슬라이스(또는 영상 또는 타일 그룹)에서만 시그널링될 수 있다.

b. 대안적으로, 재구성 모델 정보(예: tile_group_reshaper_model()의 정보)는 CRA-슬라이스(또는 영상 또는 타일 그룹)에서만 시그널링될 수 있다.

c. 대안적으로, 재구성 모델 정보(예: tile_group_reshaper_model()의 정보)는 I-RAP 슬라이스(또는 영상 또는 타일 그룹)에서만 시그널링될 수 있다. I-RAP 슬라이스들 (또는 영상들 또는 타일 그룹들)은 IDR 슬라이스들 (또는 영상들 또는 타일 그룹들) 및/또는 CRA 슬라이스들(또는 영상 또는 타일 그룹) 및/또는 끊어진 링크 액세스(BLA) 슬라이스들(또는 영상 또는 타일 그룹들)을 포함할 수 있다.

d. 대안적으로, 재구성 모델 정보(예: tile_group_reshaper_model()의 정보)는 시퀀스 레벨(예: SPS) 또는 영상 레벨(예: PPS) 또는 APS에서 시그널링될 수 있다.

3. 특정 영상 유형(IRAP 영상과 같은)에 걸쳐 영상/슬라이스/타일 그룹의 재구성 정보를 활용하는 것을 허용하지 않는 것이 제안된다.

a. 일 예에서, I 슬라이스(또는 영상 또는 타일 그룹)가 제1 슬라이스(또는 영상 또는 타일 그룹) 이후에 제2 슬라이스(또는 영상 또는 타일 그룹) 이전에 전송되거나 제2 슬라이스(또는 영상 또는 타일 그룹) 자체가 I 슬라이스(또는 영상 또는 타일 그룹)인 경우 제1 슬라이스(또는 영상 또는 타일 그룹)에서 시그널링된 재구성 모델 정보는 제2 슬라이스(또는 영상 또는 타일 그룹)에서 사용될 수 없다.

b. 대안적으로, IDR 슬라이스(또는 영상 또는 타일 그룹)가 제1 슬라이스(또는 영상 또는 타일 그룹) 이후에 제2 슬라이스(또는 영상 또는 타일 그룹) 이전에 전송되거나 제2 슬라이스(또는 영상 또는 타일 그룹) 자체가 IDR 슬라이스(또는 영상 또는 타일 그룹)인 경우 제1 슬라이스(또는 영상 또는 타일 그룹)에서 시그널링된 재구성 모델 정보는 제2 슬라이스(또는 영상 또는 타일 그룹)에서 사용될 수 없다.

c. 대안적으로, CRA 슬라이스(또는 영상 또는 타일 그룹)가 제1 슬라이스(또는 영상 또는 타일 그룹) 이후에 제2 슬라이스(또는 영상 또는 타일 그룹) 이전에 전송되거나 제2 슬라이스(또는 영상 또는 타일 그룹) 자체가 CRA 슬라이스(또는 영상 또는 타일 그룹)인 경우 제1 슬라이스(또는 영상 또는 타일 그룹)에서 시그널링된 재구성 모델 정보는 제2 슬라이스(또는 영상 또는 타일 그룹)에서 사용될 수 없다.

d. 대안적으로, I-RAP 슬라이스(또는 영상 또는 타일 그룹)가 제1 슬라이스(또는 영상 또는 타일 그룹) 이후에 제2 슬라이스(또는 영상 또는 타일 그룹) 이전에 전송되거나 제2 슬라이스(또는 영상 또는 타일 그룹) 자체가 I-RAP 슬라이스(또는 영상 또는 타일 그룹)인 경우 제1 슬라이스(또는 영상 또는 타일 그룹)에서 시그널링된 재구성 모델 정보는 제2 슬라이스(또는 영상 또는 타일 그룹)에서 사용될 수 없다. I-RAP 슬라이스들 (또는 영상들 또는 타일 그룹들)은 IDR 슬라이스들 (또는 영상들 또는 타일 그룹들) 및/또는 CRA 슬라이스들(또는 영상 또는 타일 그룹) 및/또는 끊어진 링크 액세스(BLA) 슬라이스들(또는 영상 또는 타일 그룹들)을 포함할 수 있다.

4. 일 예에서, I-슬라이스(또는 영상, 또는 타일 그룹)에서 플래그가 시그널링된다. 플래그가 X이면 재구성 모델 정보가 이 슬라이스(또는 영상 또는 타일 그룹)에서 시그널링되고, 그렇지 않으면 이 슬라이스(또는 영상 또는 타일 그룹)를 디코딩하기 전에 재구성 모델이 초기화된다. 예를 들어, X = 0 또는 1이다.

a. 대안적으로, IDR-슬라이스(또는 영상 또는 타일 그룹)에서 플래그가 시그널링된다. 플래그가 X이면 재구성 모델 정보가 이 슬라이스(또는 영상 또는 타일 그룹)에서 시그널링되고, 그렇지 않으면 이 슬라이스(또는 영상 또는 타일 그룹)를 디코딩하기 전에 재구성 모델이 초기화된다. 예를 들어, X = 0 또는 1이다.

b. 대안적으로, CRA-슬라이스(또는 영상 또는 타일 그룹)에서 플래그가 시그널링된다. 플래그가 X이면 재구성 모델 정보가 이 슬라이스(또는 영상 또는 타일 그룹)에서 시그널링되고, 그렇지 않으면 이 슬라이스(또는 영상 또는 타일 그룹)를 디코딩하기 전에 재구성 모델이 초기화된다. 예를 들어, X = 0 또는 1이다.

c. 대안적으로, I-RAP-슬라이스(또는 영상 또는 타일 그룹)에서 플래그가 시그널링된다. 플래그가 X이면 재구성 모델 정보가 이 슬라이스(또는 영상 또는 타일 그룹)에서 시그널링되고, 그렇지 않으면 이 슬라이스(또는 영상 또는 타일 그룹)를 디코딩하기 전에 재구성 모델이 초기화된다. 예를 들어, X = 0 또는 1이다. I-RAP 슬라이스들 (또는 영상들 또는 타일 그룹들)은 IDR 슬라이스들 (또는 영상들 또는 타일 그룹들) 및/또는 CRA 슬라이스들(또는 영상 또는 타일 그룹) 및/또는 끊어진 링크 액세스(BLA) 슬라이스들(또는 영상 또는 타일 그룹들)을 포함할 수 있다.

5. 일 예에서, 하나의 영상이 여러 슬라이스(또는 타일 그룹)로 분할되는 경우 각 슬라이스(또는 타일 그룹)는 동일한 재구성 모델 정보를 공유해야 한다.

a. 일례로, 하나의 영상이 여러 개의 슬라이스(또는 타일 그룹)로 분할된 경우, 제1 슬라이스(또는 타일 그룹)만이 재구성 모델 정보를 시그널링할 수 있다.

6. 변형에 사용되는 변수들은 비트 깊이에 따라 초기화, 조작 및 제한되어야 한다.

a. 일 예에서 MaxBinIdx = f(BitDepth) 여기서 f는 함수이다. 예: $MaxBinIdx=4*2^{(BitDepth-8)}-1$

b. 일 예에서, RspCW [i]는 g(BitDepth)에서 2 * OrgCW - 1의 범위에 있어야 한다. 예를 들어, RspCW [i]는 $8*2^{(BitDepth-8)}$에서 2 * OrgCW - 1의 범위에 있어야 한다.

7. 일 예에서, ScaleCoef[i] = InvScaleCoeff[i]= 1 << shiftY if RspCW[i]가 0과 같으면 for i = 0, 1,..., MaxBinIdx.

8. reshape_model_delta_max_bin_idx는 0 내지 MaxBinIdx-reshape_model_min_bin_idx의 범위에 있어야 함을 제

안한다.

a. 대안적으로 reshape_model_delta_max_bin_idx는 0 내지MaxBinIdx의 범위에 있어야 한다.

b. 일 예에서, reshape_model_delta_max_bin_idx는 0 내지 MaxBinIdx-reshape_model_min_bin_idx의 범위로 클리핑된다.

c. 일 예에서 reshape_model_delta_max_bin_idx는 0 내지 MaxBinIdx의 범위로 클리핑된다.

d. 일 예에서 reshaper_model_min_bin_idx는 reshaper_model_max_bin_idx보다 작거나 같아야 한다.

e. 일 예에서 reshaper_model_max_bin_idx는 reshaper_model_min_bin_idx내지 MaxBinIdx의 범위로 클리핑된다.

f. 일 예에서 reshaper_model_min_bin_idx는 0 내지 reshaper_model_max_bin_idx의 범위로 클리핑된다.

g. 위의 제한들 중 하나 또는 일부는 준수하는 비트-스트림에 필요할 수 있다.

9. reshape_model_max_bin_idx가 reshape_model_min_bin_idx +reshape_model_delta_maxmin_bin_idx와 동일하게 설정되도록 제안된다. 여기서, 무부호 정수인 reshape_model_delta_maxmin_bin_idx는 reshape_model_min_bin_idx 이후에 시그널링되는 신택스 요소이다.

b. 일 예에서 reshape_model_delta_maxmin_bin_idx는 0내지 MaxBinIdx-reshape_model_min_bin_idx의 범위에 있어야 한다.

c. 일 예에서 reshape_model_delta_maxmin_bin_idx는 1 내지 MaxBinIdx-reshape_model_min_bin_idx의 범위에 있어야 한다.

d. 일 예에서 reshape_model_delta_maxmin_bin_idx는 0 내지 MaxBinIdx-reshape_model_min_bin_idx의 범위로 클리핑되어야 한다.

e. 일 예에서 reshape_model_delta_maxmin_bin_idx는 1 내지 MaxBinIdx-reshape_model_min_bin_idx의 범위로 클리핑되어야 한다.

f. 위의 제한들 중 하나 또는 일부는 준수하는 비트-스트림에 필요할 수 있다.

10. reshaper_model_bin_delta_abs_cw_prec_minus1은 임계값 T보다 작아야 한다고 제안된다.

a. 일 예에서, T는 6 또는 7과 같은 고정된 숫자일 수 있다.

b. 일 예에서, T는 비트 깊이에 의존할 수 있다.

c. 위의 제한들은 준수하는 비트-스트림에 필요할 수 있다.

11. RspCW[i]는 RspCW[i -1], 즉 RspCW[i] = RspCW[i -1] + RspDeltaCW[i]에 의해 예측될 수 있다고 제안된다.

a. 일 예에서, RspCW[i]는 reshaper_model_min_bin_idx < = i <= reshaper_model_max_bin_idx인 경우에 RspCW[i -1]에 의해 예측될 수 있다.

b. 일 예에서, RspCW[i]는 OrgCW에 의해 예측되고, 즉 i가 0인 경우에 RspCW[i] = OrgCW + RspDeltaCW[i]이다.

c. 일 예에서, RspCW[i]는 OrgCW에 의해 예측되고, 즉 i가 reshaper_model_min_bin_idx와 동일한 경우 RspCW[i] = OrgCW + RspDeltaCW[i]이다.

12. reshape_model_bin_delta_sign_CW[i]는 절대 시그널링되지 않고 RspDeltaCW[i] = reshape_model_bin_delta_abs_CW[i]는 항상 양수임을 제안한다.

a. 일 예에서, RspCW[i] = MinV + RspDeltaCW[i].

i. 일 예에서 MinV = 32;

ii. 일 예에서 MinV = g(BitDepth)이다. 예를 들어 MinV = 8*2(BitDepth-8)이다.

13. invAvgLuma 계산은 색상 형식에 의존할 수 있다.

a. 일 예에서, invAvgLuma = Clip1Y((S_iS_j predMapSamples[(xCurr << scaleX) + i][(yCurr << scaleY) + j] + ((nCurrSw<<scaleX) * (nCurrSh<<scaleY) >>1)) / ((nCurrSw<<scaleX) * (nCurrSh<<scaleY)))

i. scaleX=scaleY=1 for 4:2:0 format;

ii. scaleX=scaleY=0 for 4:4:4 format;

iii. scaleX = 1and scaleY = 0 for 4:2:2 format.

14. 영상 역 매핑 프로세스에서 클리핑은 상한과 하한을 별도로 고려할 수 있음을 제안한다.

a. 일 예에서, invLumaSample = Clip3(minVal, maxVal, invLumaSample), minVal, maxVal은 다른 조건들에 따라 계산된다.

i. 예를 들어 minVal = T1 << (BitDepth - 8) if reshape_model_min_bin_idx > 0; 그렇지 않으면 minVal=0; 예: T1 = 16.

ii. 예를 들어, maxVal = T2 << (BitDepth - 8) if reshape_model_max_bin_idx < MaxBinIdx; 그렇지 않으면 maxVal =(1<< BitDepth)-1; 예: T2 = 235. 다른 예에서 T2 = 40.

15. ReshapePivot[i]는 ReshapePivot[i]<=T로 제한되어야 한다고 제안된다. 예: T = (1<< BitDepth)-1.

a. 예를 들어, Reshape Pivot[in + 1] = min(Reshape Pivot[i] + RspCW[i], T).

16. 채도 구성요소들에 대한 각 픽셀 도메인 잔차 값을 변형하는 대신, 채도 QP 오프셋(dChromaQp로 표시됨)이 각 블록 또는 TU에 대해 암시적으로 도출될 수 있고, 채도 QP에 추가될 수 있다. 이러한 방식으로 채도 구성요소들의 변형은 양자화/역양자화 프로세스로 병합된다.

a. 일 예에서, dChromaQp는 repLumaVal로 표시되는 대표적인 휘도 값에 기초하여 도출될 수 있다.

b. 일 예에서, repLumaVal은 블록 또는 TU의 부분 또는 전체 휘도 예측 값을 사용하여 도출될 수 있다.

c. 일 예에서, repLumaVal은 블록 또는 TU의 휘도 재구성된 값의 일부 또는 전체를 사용하여 도출될 수 있다.

d. 일 예에서, repLumaVal은 부분 또는 전체 휘도 예측 또는 블록 또는 TU의 재구성된 값의 평균으로서 도출될 수 있다.

e. ReshapePivot[idx] <= repLumaVal < ReshapePivot[idx + 1]라고 가정하면 InvScaleCoeff[idx]를 사용하여 dChromaQp를 도출할 수 있다.

i. 일 예에서, dQp는 argmin abs(2^(x/6 + shiftY) - InvScaleCoeff[idx]), x = -N..., M으로 선택될 수 있다. 예를 들어, N = M = 63이다.

ii. 일 예에서, dQp는 argmin abs(1 - (2^(x/6 + shiftY) / InvScaleCoeff[idx])), x = -N..., M으로 선택될 수 있다. 예를 들어 N = M = 63이다.

iii. 일 예에서, 다른 InvScaleCoeff[idx] 값에 대해, dChromaQp가 미리 계산되어 순람표에 저장될 수 있다.

17. 휘도 구성요소들에 대한 각 픽셀 도메인 잔차 값을 변형하는 대신, 휘도 QP 오프셋(dQp로 표시됨)이 각 블록에 대해 암시적으로 도출될 수 있고 휘도 QP에 추가될 수 있다. 이러한 방식으로 휘도 구성요소의 재구성은 양자화/역양자화 프로세스에 병합된다.

a. 일 예에서, dQp는 repLumaVal로 표시되는 대표적인 휘도 값에 기초하여 도출될 수 있다.

b. 일 예에서, repLumaVal은 블록 또는 TU의 부분 또는 전체 휘도 예측 값을 사용하여 도출될 수 있다.

c. 일 예에서, repLumaVal은 블록 또는 TU의 부분 또는 전체 휘도 예측 값의 평균으로서 도출될 수 있다.

d. idx = repLumaVal / OrgCW라고 가정하면 InvScaleCoeff[idx]를 사용하여 dQp를 도출할 수 있다.

i. 일 예에서, dQp는 argmin abs(2^(x/6 + shiftY) - InvScaleCoeff[idx]), x = -N..., M으로 선택될 수 있다. 예를 들어, N = M = 63이다.

ii. 일 예에서, dQp는 argmin abs(1 - (2^(x/6 + shiftY) / InvScaleCoeff[idx])), x = -N..., M으로 선택될 수 있다. 예를 들어 N = M = 63이다.

iii. 일 예에서, 상이한 InvScaleCoeff[idx] 값에 대해, dQp는 미리 계산되어 순람표에 저장될 수 있다.

이 경우, dChromaQp는 dQp와 동일하게 설정될 수 있다.

5. 개시된 기술의 예시적인 구현들

도 27a는 비디오 처리 장치(2700)의 블록도이다. 장치(2700)는 본 명세서에서 설명된 방법들 중 하나 이상을 구현하기 위해 이용될 수 있다. 장치(2700)는 스마트폰, 태블릿, 컴퓨터, 사물인터넷(IoT) 수신기 등으로 구현될 수 있다. 장치(2700)는 하나 이상의 프로세서들(2702), 하나 이상의 메모리들(2704) 및 비디오 처리 하드웨어(2706)를 포함할 수 있다. 프로세서(들)(2702)는 본 명세서에 설명된 하나 이상의 방법들을 구현하도록 구성될 수 있다. 메모리(메모리들)(2704)는 본 명세서에서 설명된 방법들 및 기술들을 구현하기 위해 이용되는 데이터 및 코드를 저장하기 위해 이용될 수 있다. 비디오 처리 하드웨어(2706)는 하드웨어 회로에서 본 명세서에 설명된 일부 기술들을 구현하기 위해 이용될 수 있고, 부분적으로 또는 완전히 프로세서들(2702)(예를 들어, 그래픽 프로세서 코어 GPU 또는 다른 신호 처리 회로)의 일부일 수 있다.

도 27b는 개시된 기술들이 구현될 수 있는 비디오 처리 시스템의 블록도의 다른 예이다. 도 27b는 여기에 개시된 다양한 기술들이 구현될 수 있는 예시적인 비디오 처리 시스템(4100)을 도시하는 블록도이다. 다양한 구현은 시스템(4100)의 구성요소들 중 일부 또는 전체를 포함할 수 있다. 시스템(4100)은 비디오 콘텐츠를 수신하기 위한 입력(4102)을 포함할 수 있다. 비디오 콘텐츠는 원시 또는 압축되지 않은 형식, 예를 들어 8 또는 10 비트 다중 구성요소 픽셀 값으로 수신될 수 있거나 압축 또는 인코딩된 형식일 수 있다. 입력(4102)은 네트워크 인터페이스, 이웃한 버스 인터페이스, 또는 저장 인터페이스를 나타낼 수 있다. 네트워크 인터페이스의 예들로는 이더넷, 수동 광 네트워크(passive optical network)(PON) 등과 같은 유선 인터페이스와 Wi-Fi 또는 셀룰러 인터페이스와 같은 무선 인터페이스가 있다.

시스템(4100)은 본 명세서에 설명된 다양한 코딩 또는 인코딩 방법들을 구현할 수 있는 코딩 구성요소(4104)를 포함할 수 있다. 코딩 구성요소(4104)는 비디오의 코딩된 표현을 생성하기 위해 입력(4102)으로부터 코딩 구성요소(4104)의 출력으로 비디오의 평균 비트 전송률을 감소시킬 수 있다. 따라서 코딩 기술들은 비디오 압축 또는 비디오 트랜스코딩 기술들이라고도 한다. 코딩 구성요소(4104)의 출력은 구성요소(4106)에 의해 표현되는 바와 같이, 연결된 통신을 통해 저장되거나 전송될 수 있다. 입력(4102)에서 수신된 비디오의 저장되거나 통신된 비트스트림(또는 코딩된) 표현은 디스플레이 인터페이스(4110)로 전송되는 픽셀 값들 또는 디스플레이 가능한 비디오를 생성하기 위해 구성요소(4108)에 의해 사용될 수 있다. 비트스트림 표현에서 사용자가 볼 수 있는 비디오를 생성하는 프로세스를 비디오 압축 해제라고도 한다. 또한 특정 비디오 처리 작업을 "코딩" 연산 또는 도구라고 하는 반면, 코딩 도구 또는 동작은 인코더에서 사용되며, 코딩의 결과를 역전시키는 대응하는 디코딩 도구 또는 동작은 디코더에 의해 수행될 것임을 이해할 것이다.

주변기기 버스 인터페이스 또는 디스플레이 인터페이스의 예들로는 USB(Universal Serial Bus) 또는 HDMI(High Definition Multimedia Interface) 또는 디스플레이 포트 등이 포함될 수 있다. 저장 인터페이스의 예들은 SATA(Serial Advanced Technology Attachment), PCI, IDE 인터페이스 등을 포함한다. 본 명세서에 기술된 기술들은 디지털 데이터 처리 및/또는 비디오 디스플레이를 수행할 수 있는 휴대폰, 랩톱, 스마트폰 또는 기타 장치와 같은 다양한 전자 장치에서 구현될 수 있다.

본 명세서에서 "비디오 처리"라는 용어는 비디오 인코딩, 비디오 디코딩, 비디오 압축 또는 비디오 압축 해제를 의미할 수 있다. 예를 들어, 비디오 압축 알고리즘은 비디오의 픽셀 표현에서 대응하는 비트스트림 표현으로 또는 그 반대로 전환하는 동안 적용될 수 있다. 현재 비디오 블록의 비트스트림 표현은, 예를 들어, 신택스에 의해 정의된 바와 같이 비트스트림 내의 다른 위치에 함께 배치되거나 확산되는 비트에 대응할 수 있다. 예를 들어, 매크로블록은 변환되고 코딩된 오류 잔차 값의 관점에서 그리고 또한 헤더의 비트 및 비트스트림의 다른 필드를 사용하여 인코딩될 수 있다.

개시된 방법 및 기술은 본 명세서에 개시된 기술의 사용을 허용함으로써 스마트폰, 랩탑, 데스크탑 및 유사한 장치와 같은 비디오 처리 장치 내에 통합된 비디오 인코더 및/또는 디코더 실시예에 이점이 있음을 인식할 것이다. .

도 28a는 비디오 처리의 예시적인 방법(2810)에 대한 흐름도이다. 방법(2810)은 단계(2812)에서 비디오의 비디오 영역의 다수의 비디오 유닛들과 다수의 비디오 유닛의 코딩된 표현 사이의 전환을 위해 다수의 비디오 유닛들에 의해 공통적으로 공유되는 모델 정보를 변형하는 것을 결정하는 단계를 포함한다. 방법(2810)은 단계(2814)에서 비디오의 코딩된 표현과 비디오 사이의 전환을 수행하는 단계를 더 포함한다. 도 28b는 비디오 처

리의 예시적인 방법(2820)에 대한 흐름도이다. 방법(2820)은, 단계(2822)에서, 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 간의 전환을 위해, 비디오의 비트-깊이의 함수로서 모델 정보를 변형하는 변수의 값을 결정하는 단계를 포함한다. 방법(2820)은 단계(2824)에서 결정에 기초하여 전환을 수행하는 단계를 더 포함한다.

도 28c는 비디오 처리의 예시적인 방법(2830)에 대한 흐름도이다. 방법(2830)은 단계(2832)에서, 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 사이의 전환에 대해, 인 루프 변형(ILR)을 활성화할지 또는 비활성화할지 여부를 결정하는 단계를 포함한다. 방법(2830)은 단계(2834)에서 결정에 기초하여 전환을 수행하는 단계를 더 포함한다. 일부 구현들에서, 결정은 변형 모델 정보가 초기화되지 않은 경우 ILR을 비활성화하기로 결정한다.

도 28d는 비디오 처리의 예시적인 방법(2840)에 대한 흐름도이다. 방법(2840)은 단계(2842)에서 비디오의 제1 비디오 영역과 제1 비디오 영역의 코딩된 표현 사이의 변환에 대해 규칙에 기초한 전환을 위해 제2 비디오 영역으로부터의 변형 정보가 사용가능한지를 결정하는 단계를 포함한다. 방법(2840)은 단계(2844)에서 결정에 따라 전환을 수행하는 단계를 더 포함한다.

도 28e는 비디오 처리의 예시적인 방법(2850)에 대한 흐름도이다. 방법(2850)은 단계(2852)에서 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 사이의 전환을 수행하는 단계를 포함한다. 일부 구현들에서, 코딩된 표현은 하나 이상의 비디오 영역들 중 일부의 인-루프 변형(ILR)에 적용 가능한 모델 정보를 변형하는 것을 포함한다. 일부 구현들에서, 변형 모델 정보는 제1 도메인 및 제2 도메인에서의 비디오 유닛의 표현 및/또는 채도 비디오 유닛의 스케일링 채도 잔차에 기초하여 비디오 영역의 비디오 유닛의 재구성을 위한 정보를 제공한다. 일부 구현들에서, 변형 모델 정보는 초기화 규칙에 기초하여 초기화된다. 일부 구현들에서, 변형 모델 정보는 비디오 영역이 특정 코딩 유형을 사용하여 코딩되는 경우에만 코딩된 표현에 포함된다. 일부 구현들에서, 전환은 현재 비디오 영역이 특정 코딩 유형을 사용하여 코딩되도록 비디오의 현재 비디오 영역과 현재 비디오 영역의 코딩된 표현 사이에서 수행되고, 코딩된 표현은 비디오 영역 레벨에서 코딩된 표현의 플래그 값에 기초하여 코딩된 표현의 모델 정보를 조건부로 변형하도록 명시하는 형식 규칙을 준수한다.

일부 구현들에서, 변형 모델 정보는 제1 도메인 및 제2 도메인에서의 표현 및/또는 채도 비디오 유닛의 스케일링 채도 잔차에 기초하여 비디오 영역의 비디오 유닛의 재구성을 위한 정보를 제공한다. 일부 구현들에서, 변형 모델 정보는 허용된 최대 빈 인덱스와 재구성에서 사용될 최대 빈 인덱스 간의 차이를 명시하는 신택스 요소를 포함하는 파라미터 세트를 포함하고, 파라미터는 범위 내에 있다. 일부 구현들에서, 재구성 모델 정보는 재구성에 사용될 최대 빈 인덱스를 포함하는 파라미터 세트를 포함하고, 최대 빈 인덱스는 재구성에 사용될 최소 빈 인덱스와 무부호 정수이고 최소 빈 인덱스 이후에 시그널링되는 신택스 요소의 합과 동일한 제1 값으로서 도출된다.

일부 구현들에서, 변형 모델 정보는 대응하는 빈으로부터 절대 델타 코드워드 값을 명시하는 제2 신택스 요소를 나타내기 위해 이용되는 비트 수를 도출하는 제1 신택스 요소를 포함하는 파라미터 세트를 포함하고, 제1 신택스 요소는 임계값보다 작은 값을 갖는다. 일부 구현들에서, 변형 모델 정보는 ILR에서 사용되는 i번째 빈의 기울기를 나타내고 (i-1)번째 파라미터에 기초한 값을 갖는 i번째 파라미터를 포함하는 파라미터 세트를 포함하고, i는 양의 정수다. 일부 구현들에서, ILR에 사용되는 변형 모델 정보는 시그널링되지 않고 RspDeltaCW[i] = reshape_model_bin_delta_abs_CW [i]가 항상 양수인 reshape_model_bin_delta_sign_CW[i]를 포함하는 파라미터 세트를 포함한다. 일부 구현들에서, 변형 모델 정보는 비디오 영역의 색상 형식에 따른 스케일링을 위한 휘도 값들을 사용하기 위한 파라미터 invAvgLuma를 포함하는 파라미터 세트를 포함한다. 일부 구현들에서, 전환은 재구성된 영상 휘도 샘플들을 수정된 재구성된 영상 휘도 샘플들로 변환하기 위한 영상 역 맵핑 프로세스를 포함하고, 영상 역 맵핑 프로세스는 상한 및 하한이 서로 별도로 설정되는 클리핑을 포함한다. 일부 구현들에서, 변형 모델 정보는 Pivot[i]<=T가 되도록 제한되는 피벗 수량을 포함하는 파라미터 세트를 포함한다. 일부 구현들에서, 채도 양자화 파라미터(QP)는 그 값이 각각의 블록 또는 변환 유닛에 대해 도출되는 오프셋을 갖는다. 일부 구현들에서, 휘도 양자화 파라미터(QP)는 그 값이 각각의 블록 또는 변환 유닛에 대해 도출되는 오프셋을 갖는다.

다양한 기술들 및 실시예들은 다음 절-기반 형식을 사용하여 설명될 수 있다.

절들의 제1 세트는 이전 섹션에 나열된 개시된 기술들의 특정 특징들 및 양태들을 설명한다.

1. 시각적 미디어 처리 방법에 있어서, 현재 비디오 블록과 현재 비디오 블록의 비트스트림 표현 사이의 전환을

수행하는 단계를 포함하고, 전환 동안, 인-루프 변형 단계와 연관된 부가 정보에 따라 현재 비디오 블록의 표현을 제1 도메인에서 제2 도메인으로 변환하기 위해 인-루프 변형 단계가 사용되고, 인-루프 변형 단계는 변형 모델 정보에 부분적으로 기초하고, 전환 동안, 변형 모델 정보는 초기화 단계, 시그널링 단계, 또는 디코딩 단계 중 하나 이상에서 이용되는 방법.

2. 1절에 있어서, 디코딩 단계는 I-슬라이스, 영상, 또는 타일 그룹에 관한 것인 방법.

3. 1절에 있어서, 디코딩 단계는 순간 디코딩 리프레시(IDR) 슬라이스, 영상, 또는 타일 그룹에 관한 것인 방법.

4. 1절에 있어서, 디코딩 단계는 클린 랜덤 액세스(CRA) 슬라이스, 영상, 또는 타일 그룹에 관한 것인 방법.

5. 1절에 있어서, 디코딩 단계는 인트라 랜덤 액세스 포인트(I-RAP) 슬라이스, 영상, 또는 타일 그룹에 관한 것인 방법.

6. 1절에 있어서, I-RAP 슬라이스, 영상, 또는 타일 그룹은 IDR 슬라이스, 영상, 또는 타일 그룹, CRA 슬라이스, 영상, 또는 타일 그룹, 또는 끊어진 링크 액세스(BLA) 슬라이스, 영상 또는 타일 그룹을 포함하는 방법.

7. 1-6 절들 중 어느 하나 이상에 있어서, 초기화 단계는 디코딩 단계 이전에 발생하는 방법.

8. 1-7 절들 중 어느 하나 이상에 있어서, 초기화 단계는 OrgCW 수량을 (1 << BitDepthY) / (MaxBinIdx + 1)로 설정하는 단계를 포함하고, ReshapePivot[i] = InputPivot [i] = i * OrgCW, i = 0, 1,..., MaxBinIdx이고, OrgCW, BitDepthY, MaxBinIdx 및 ReshapePivot은 변형 모델 정보와 연관된 수량인 방법.

9. 1-7 절들 중 어느 하나 이상에 있어서, 초기화 단계는 수량을 ScaleCoef[i] = InvScaleCoeff[i]= 1 << shiftY, for i = 0, 1,..., MaxBinIdx로 설정하는 단계를 포함하고, ScaleCoef, InvScaleCoeff 및 shiftY는 변형 모델 정보와 연관된 수량인 방법.

10. 1-7 절들 중 어느 하나 이상에 있어서, 초기화 단계는 수량을 (1 << BitDepthY) / (MaxBinIdx + 1) 로 설정하는 단계를 포함하고, RspCW[i] = OrgCW for i = 0, 1,..., MaxBinIdx이고, a. OrgCW 는 (1 << BitDepthY) / (MaxBinIdx + 1)와 동일하게 설정되고, RspCW[i] = OrgCW, for i = 0, 1,..., MaxBinIdx이고, OrgCW, BitDepthY, MaxBinIdx 및 RspCW는 변형 모델 정보와 연관된 수량인 방법.

11. 1절에 있어서, 초기화 단계는 변형 모델 정보를 기본값으로 설정하는 단계를 포함하고, 시그널링 단계는 시퀀스 파라미터 세트(sequence parameter set)(SPS) 또는 영상 파라미터 세트(picture parameter set)(PPS)에 포함된 기본값을 시그널링하는 단계를 포함하는 방법.

12. 1절에 있어서, 변형 모델 정보가 초기화 단계 동안 초기화되지 않았다고 결정할 때, 인-루프 변형 단계를 비활성화하는 단계를 더 포함하는 방법.

13. 1절에 있어서, 시그널링 단계 동안, 변형 모델 정보는 I-슬라이스, 영상 또는 타일 그룹, IDR 슬라이스, 영상 또는 타일 그룹, I-RAP 슬라이스, 영상 또는 타일 그룹 중 어느 하나로 시그널링되는 방법.

14. 1절에 있어서, I-RAP 슬라이스, 영상, 또는 타일 그룹은 IDR 슬라이스, 영상, 또는 타일 그룹, CRA 슬라이스, 영상, 또는 타일 그룹, 또는 끊어진 링크 액세스(BLA) 슬라이스, 영상 또는 타일 그룹 중 하나 이상을 포함하는 방법.

15. 1절에 있어서, 시그널링 단계는 시퀀스 파라미터 세트(SPS) 또는 영상 파라미터 세트(PPS)에 포함된 재구성 모델 정보를 시그널링하는 단계를 포함하는 방법.

16. 시각적 미디어 처리 방법에 있어서, 현재 비디오 블록과 현재 비디오 블록의 비트스트림 표현 사이의 전환을 수행하는 단계를 포함하고, 전환 동안, 인-루프 변형 단계와 연관된 부가 정보에 따라 현재 비디오 블록의 표현을 제1 도메인에서 제2 도메인으로 변환하기 위해 인-루프 변형 단계가 사용되고, 인-루프 변형 단계는 변형 모델 정보에 부분적으로 기초하고, 전환 동안, 변형 모델 정보는 초기화 단계, 시그널링 단계, 또는 디코딩 단계 및 변형 모델 정보가 제1 영상에서 시그널링되는 경우 제2 영상에서 변형 모델 정보의 이용을 비활성화하는 단계 중 하나 이상에서 활용되는 방법.

17. 16절에 있어서, 중간 영상이 제1 영상 이후에 제2 영상보다 먼저 전송되면, 변형 모델 정보가 제1 영상에서 시그널링되는 경우, 제2 영상에서 변형 모델 정보의 이용을 비활성화하는 단계를 더 포함하는 방법.

18. 17절에 있어서, 제2 영상은 I 영상, IDR 영상, CRA 영상, 또는 I-RAP 영상인 방법.

19. 17-18절 중 어느 하나 이상에 있어서, 중간 영상은 I 영상, IDR 영상, CRA 영상, 또는 I-RAP 영상인 방법.

20. 16-19절 중 어느 하나 이상에 있어서, 영상은 타일 또는 그룹을 포함하는 방법.

21. 시각적 미디어 처리 방법에 있어서, 현재 비디오 블록과 현재 비디오 블록의 비트스트림 표현 사이의 전환을 수행하는 단계를 포함하고, 전환 동안, 인-루프 변형 단계와 연관된 부가 정보에 따라 현재 비디오 블록의 표현을 제1 도메인에서 제2 도메인으로 변환하기 위해 인-루프 변형 단계가 사용되고, 인-루프 변형 단계는 변형 모델 정보에 부분적으로 기초하고, 전환 동안, 변형 모델 정보는 초기화 단계, 시그널링 단계, 또는 디코딩 단계 중 하나 이상에서 이용되고, 시그널링 단계 동안 플래그에 기초하도록 영상에서 플래그를 시그널링하고, 변형 모델 정보는 영상으로 전송되고, 그렇지 않으면 변형 모델 정보는 디코딩 단계에서 영상이 디코딩되기 전에 초기화 단계에서 초기화되는 방법.

22. 21절에 있어서, 영상은 I 영상, IDR 영상, CRA 영상, 또는 I-RAP 영상인 방법.

23. 21-23절 중 어느 하나에 이상에 있어서, 영상은 타일 또는 그룹을 포함하는 방법.

24. 21절에 있어서, 플래그는 0 또는 1 값을 취하는 방법.

25. 시각적 미디어 처리 방법에 있어서, 현재 비디오 블록과 현재 비디오 블록의 비트스트림 표현 사이의 전환을 수행하는 단계를 포함하고, 전환 동안, 인-루프 변형 단계와 연관된 부가 정보에 따라 현재 비디오 블록의 표현을 제1 도메인에서 제2 도메인으로 변환하기 위해 인-루프 변형 단계가 사용되고, 인-루프 변형 단계는 변형 모델 정보에 부분적으로 기초하고, 전환 동안, 변형 모델 정보는 초기화 단계, 시그널링 단계, 또는 디코딩 단계 중 하나 이상에서 이용되고, 영상을 다수의 유닛들로 분할할 때, 다수의 유닛들 각각과 연관된 변형 모델 정보는 동일한 방법.

26. 시각적 미디어 처리 방법에 있어서, 현재 비디오 블록과 현재 비디오 블록의 비트스트림 표현 사이의 전환을 수행하는 단계를 포함하고, 전환 동안, 인-루프 변형 단계와 연관된 부가 정보에 따라 현재 비디오 블록의 표현을 제1 도메인에서 제2 도메인으로 변환하기 위해 인-루프 변형 단계가 사용되고, 인-루프 변형 단계는 변형 모델 정보에 부분적으로 기초하고, 전환 동안, 변형 모델 정보는 초기화 단계, 시그널링 단계, 또는 디코딩 단계 중 하나 이상에서 이용되고, 영상을 다수의 유닛들로 분할할 때, 변형 모델 정보는 다수의 유닛들 중 제1 유닛에서만 시그널링되는 방법.

27. 25-26 절 중 어느 하나 이상에 있어서, 유닛은 슬라이스 또는 타일 그룹에 대응하는 방법.

28. 시각적 미디어 처리 방법에 있어서, 현재 비디오 블록과 현재 비디오 블록의 비트스트림 표현 사이의 전환을 수행하는 단계를 포함하고, 전환 동안, 인-루프 변형 단계와 연관된 부가 정보에 따라 현재 비디오 블록의 표현을 제1 도메인에서 제2 도메인으로 변환하기 위해 인-루프 변형 단계가 사용되고, 인-루프 변형 단계는 변형 모델 정보에 부분적으로 기초하고, 전환 동안, 변형 모델 정보는 초기화 단계, 시그널링 단계, 또는 디코딩 단계 중 하나 이상에서 이용되고, 변형 모델 정보는 비트 깊이 값에 기초하여 조작되는 방법.

29. 28절에 있어서, 변형 모델 정보는 비트 깊이 값과 관련된 MaxBinIdx 변수를 포함하는 방법.

30. 28절에 있어서, 변형 모델 정보는 비트 깊이 값과 관련된 RspCW 변수를 포함하는 방법.

30. 29절에 있어서, 변형 모델 정보는 0 내지 MaxBinIdx 변수의 값 범위인 reshape_model_delta_max_bin_idx 변수를 포함하는 방법.

31. 29절에 있어서, 변형 모델 정보는 0내지 MaxBinIdx - reshape_model_min_bin_idx에 대응하는 값의 범위에서 클리핑되는 reshape_model_delta_max_bin_idx 변수를 포함하고, eshape_model_min_bin_idx는 변형 모델 정보의 다른 변수인 방법.

32. 29절에 있어서, 변형 모델 정보는, 0에서부터 시작하여 MaxBinIdx 변수까지의 범위에서 클리핑되는 reshape_model_delta_max_bin_idx 변수를 포함하는 방법.

33. 29절에 있어서, 변형 모델 정보는 reshaper_model_min_bin_idx 변수 및 reshaper_model_delta_max_bin_idx 변수를 포함하고, reshaper_model_min_bin_idx 변수는 reshaper_model_min_bin_idx 변수보다 작은 방법.

34. 29절에 있어서, 변형 모델 정보는reshaper_model_min_bin_idx 변수 및 reshaper_model_delta_max_bin_idx

변수를 포함하고, eshaper_model_max_bin_idx는 reshaper_model_min_bin_idx 내지 MaxBinIdx의 다양한 범위로 클리핑되는 방법.

35. 29절에 있어서, 변형 모델 정보는reshaper_model_min_bin_idx 변수 및 reshaper_model_delta_max_bin_idx 변수를 포함하고, reshaper_model_min_bin_idx는 0 내지 reshaper_model_max_bin_idx의 다양한 범위로 클리핑되는 방법.

36. 29절에 있어서, 변형 모델 정보는 임계값보다 작은 reshaper_model_bin_delta_abs_cw_prec_minus1 변수를 포함하는 방법.

37. 36절에 있어서, 임계값은 고정된 숫자인 방법.

38. 36절에 있어서, 임계값은 비트 깊이 값에 기초하는 방법.

39. 28절에 있어서, 변형 모델 정보는 invAvgLuma = Clip1Y((S_iS_j predMapSamples[(xCurr << scaleX) + i][(yCurr << scaleY) + j] + ((nCurrSw<<scaleX) * (nCurrSh<<scaleY) >>1)) / ((nCurrSw<<scaleX) * (nCurrSh<<scaleY)))로 계산된 invAvgLuma 변수를 포함하는 방법.

40. 28절에 있어서, 4:2:0 형식에 대해 scaleX=scaleY=1인 방법.

41. 28절에 있어서, 4:4:4 형식에 대해 scaleX=scaleY=1인 방법.

42. 28절에 있어서, 4:2:2 형식에 대해 scaleX = 1 및 scaleY = 0인 방법.

43. 28절에 있어서, 변형 모델 정보는, invLumaSample = Clip3(minVal, maxVal, invLumaSample)로서 계산된 invLumaSample 변수를 포함하는 방법.

44. 43절에 있어서, reshape_model_min_bin_idx > 0이면 minVal = T1 << (BitDepth - 8)이고, 그렇지 않으면 minVal=0인 방법.

45. 43절에 있어서, reshape_model_max_bin_idx < MaxBinIdx이면 maxVal = T2 << (BitDepth - 8)이고, 그렇지 않으면 maxVal =(1<< BitDepth)-1인 방법.

46. 44절에 있어서, T1=16인 방법.

47. 45절에 있어서, T2는 값 235 또는 40을 취하는 방법.

48. 28절에 있어서, 변형 모델 정보는, ReshapePivot[i]<=T가 되도록 하는 방식으로 제한되는 ReshapePivot 수량을 포함하는 방법.

49. 48절에 있어서, T는 T = (1<< BitDepth)-1로서 계산될 수 있고, BitDepth는 비트 깊이 값에 대응하는 방법.

50. 1 내지 49절 중 어느 하나 이상에 있어서, 비디오 처리는 인코더-측 구현인 방법.

51. 1 내지 49절 중 어느 하나 이상에 있어서, 비디오 처리는 디코더-측 구현인 방법.

52. 프로세서 및 그에 대한 명령어들이 있는 비일시적 메모리를 포함하는 비디오 시스템의 장치에 있어서, 프로세서에 의한 실행 시에 명령어들은 프로세서가 1절 내지 49절 중 어느 하나 이상의 방법을 실행하게 하는 장치.

53. 비일시적 컴퓨터 판독 가능 매체에 저장된 컴퓨터 프로그램 제품에 있어서, 1절 내지 49절 중 어느 하나 이상의 방법을 수행하기 위한 프로그램 코드를 포함하는 컴퓨터 프로그램 제품.

절들의 제2 세트는 예를 들어 예 1-7을 포함하는 이전 섹션에 나열된 개시된 기술들의 특정 특징들 및 양태들을 설명한다.

1. 비디오 처리 방법에 있어서, 비디오의 비디오 영역의 다수의 비디오 유닛들과 다수의 비디오 유닛들의 코딩된 표현 간의 전환을 위해, 다수의 비디오 유닛들에 의해 공통적으로 공유되는 변형 모델 정보를 결정하는 단계; 및 비디오의 코딩된 표현과 비디오 간의 전환을 수행하는 단계. 변형 모델 정보는 제1 도메인 및 제2 도메인에서 비디오 샘플들을 구성하고/하거나 채도 비디오 유닛의 채도 잔차를 스케일링하기 위한 정보를 제공하는 방법.

2. 1절에 있어서, 다수의 비디오 유닛들은 다수의 슬라이스들 또는 다수의 타일 그룹에 대응하는 방법.

3. 2절에 있어서, 다수의 비디오 유닛들은 동일한 하나의 영상과 연관되는 방법.

4. 1절에 있어서, 변형 모델 정보는 다수의 비디오 유닛들의 코딩된 표현에 한 번만 존재하는 방법.

5. 비디오 처리를 위한 방법에 있어서, 하나 이상의 비디오 영역을 포함하는 비디오의 코딩된 표현과 비디오 간의 전환을 위해, 비디오의 비트-깊이의 함수로서 모델 정보를 변형하는 변수의 값을 결정하는 단계, 및 결정에 기초하여 전환을 수행하는 단계를 포함하고, 변형 정보는 하나 이상의 비디오 영역들 중 일부의 인-루프 재형성(ILR)에 적용 가능하고, 변형 정보는 제1 도메인 및 제2 도메인에서의 표현, 및/또는 채도 비디오 유닛의 채도 잔차의 스케일링에 기초하여 비디오 영역의 비디오 유닛의 재구성에 대한 정보를 제공하는 방법.

6. 5절에 있어서, 변수는 비트 깊이 값과 관련된 MaxBinIdx 변수를 포함하는 방법.

7. 5절에 있어서, 변수는 비트 깊이 값과 관련된 RspCW 변수를 포함하는 방법.

8. 7절에 있어서, 변수 RspCW는 비디오 영역의 비디오 유닛의 재구성을 도출하기 위해 이용되는 변수 ReshapePivot을 도출하기 위해 이용되는 방법.

9. 5절에 있어서, RspCW[i]가 0내지 MaxBinIdx 범위에 있는 i에 대해 0과 같은 경우, 변수는 InvScaleCoeff[i]= 1 << shiftY를 만족하는 InvScaleCoef[i]를 포함하고, shiftY는 정밀도를 나타내는 정수인 방법.

10. 9절에 있어서, shifty는 11 또는 14인 방법.

11. 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 간의 전환을 수행하는 단계를 포함하고, 코딩된 표현은 하나 이상의 비디오 영역들 중 일부의 인-루프 변형(ILR)에 적용 가능한 변형 모델 정보를 포함하고, 변형 모델 정보는 제1 도메인 및 제2 도메인에서의 표현, 및/또는 채도 비디오 유닛의 채도 잔차의 스케일링에 기초하여 비디오 영역의 비디오 유닛의 재구성에 대한 정보를 제공하고, 변형 모델 정보는 초기화 규칙에 기초하여 초기화되는 방법.

12. 11절에 있어서, 변형 모델 정보의 초기화는 비디오의 시퀀스를 디코딩하기 전에 발생하는 방법.

13. 11절에 있어서, 변형 모델 정보의 초기화는 i) 인트라(I) 코딩 유형, ii) 순간 디코딩 리프레시(IDR) 코딩 유형을 사용하여 코딩된 비디오 영역, iii) 클린 랜덤 액세스(CRA) 코딩 유형을 사용하여 코딩된 비디오 영역, 또는 iv) 인트라 랜덤 액세스 포인트(I- RAP) 코딩 유형을 사용하여 코딩된 비디오 영역 중 적어도 하나를 디코딩하기 전에 발생하는 방법.

14. 13절에 있어서, 인트라 랜덤 액세스 포인트(I-RAP)를 사용하여 코딩된 비디오 영역은 i) 순간 디코딩 리프레시(IDR) 코딩 유형ii) 클린 랜덤 액세스(CRA) 코딩 유형을 사용하여 코딩된 비디오 영역, 및/또는 iii) 끊어진 링크 액세스(BLA) 코딩 유형을 사용하여 코딩된 비디오 영역을 사용하여 코딩된 비디오 영역 중 적어도 하나를 포함하는 방법.

15. 11절에 있어서, 변형 모델 정보의 초기화는, OrgCW 수량을 (1 << BitDepthY) / (MaxBinIdx + 1)로 설정하는 단계를 포함하고, ReshapePivot[i] = InputPivot [i] = i * OrgCW, i = 0, 1,..., MaxBinIdx이고, OrgCW, BitDepthY, MaxBinIdx 및 ReshapePivot은 변형 모델 정보와 연관된 수량인 방법.

16. 11절에 있어서, 변형 모델 정보의 초기화는, 수량을 ScaleCoef[i] = InvScaleCoeff[i]= 1 << shiftY, for i = 0, 1,..., MaxBinIdx로 설정하는 단계를 포함하고, ScaleCoef, InvScaleCoeff 및 shiftY는 변형 모델 정보와 연관된 수량인 방법.

17. 11절에 있어서, 변형 모델 정보의 초기화는, 수량을 (1 << BitDepthY) / (MaxBinIdx + 1) 로 설정하는 단계를 포함하고, RspCW[i] = OrgCW for i = 0, 1,..., MaxBinIdx이고, a. OrgCW 는 (1 << BitDepthY) / (MaxBinIdx + 1)와 동일하게 설정되고, RspCW[i] = OrgCW, for i = 0, 1,..., MaxBinIdx이고, OrgCW, BitDepthY, MaxBinIdx 및 RspCW는 변형 모델 정보와 연관된 수량인 방법.

18. 1절에 있어서, 변형 모델 정보의 초기화는 변형 모델 정보를 기본값으로 설정하는 단계를 포함하고, 기본값은 시퀀스 파라미터 세트(SPS) 또는 영상 파라미터 세트(PPS)에 포함되는 방법.

19. 비디오 처리 방법에 있어서, 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 간의 전환을 위해, 인 루프 변형(ILR)을 활성화할지 또는 비활성화할지를 결정하는 단계; 및 결정에 기초하여 전환을 수행하는 단계를 포함하고, 코딩된 표현은 하나 이상의 비디오 영역들 중 일부의ILR에 적용 가능한 변형 모델 정보를 포함하고, 변형 모델 정보는 제1 도메인 및 제2 도메인에서의 표현, 및/또는 채도 비디오 유닛의 채도

잔차의 스케일링에 기초하여 비디오 영역의 비디오 유닛의 재구성에 대한 정보를 제공하고, 결정하는 단계는, 변형 모델 정보가 초기화되지 않은 경우, ILR을 비활성화하도록 결정하는 방법.

20. 비디오 처리 방법에 있어서, 하나 이상의 비디오 영역을 포함하는 비디오의 코딩된 표현과 비디오 간의 전환을 수행하는 단계를 포함하고, 코딩된 표현은 하나 이상의 비디오 영역들 중 일부의 인 루프 변형(ILR)에 적용 가능한 변형 모델 정보를 포함하고, 변형 모델 정보는 제1 도메인 및 제2 도메인에서의 표현, 및/또는 채도 비디오 유닛의 채도 잔차의 스케일링에 기초하여 비디오 영역의 비디오 유닛의 재구성에 대한 정보를 제공하고, 변형 모델 정보는 비디오 영역이 특정 코딩 유형을 사용하여 코딩되는 경우에만 코딩된 표현에 포함되는 방법.

21. 20절에 있어서, 비디오 영역은 슬라이스 또는 영상 또는 타일 그룹이고, 특정 코딩 유형은 인트라(I) 코딩 유형인 방법.

22. 20절에 있어서, 비디오 영역은 슬라이스 또는 영상 또는 타일 그룹이고, 특정 코딩 유형은 순간 디코딩 리프레시(IDR) 코딩 유형인 방법.

23. 20절에 있어서, 비디오 영역은 슬라이스 또는 영상 또는 타일 그룹이고, 특정 코딩 유형은 클린 랜덤 액세스(CRA) 코딩 유형인 방법.

24. 20절에 있어서, 비디오 영역은 슬라이스 또는 영상 또는 타일 그룹이고, 특정 코딩 유형은 인트라 랜덤 액세스 포인트(I-RAP) 코딩 유형인 방법.

25. 24절에 있어서, 인트라 랜덤 액세스 포인트(I-RAP) 코딩 유형을 사용하여 코딩된 중간 비디오 영역은 i) 순간 디코딩 리프레시(IDR) 코딩 유형을 사용하여 코딩된 비디오 영역, ii) 클린 랜덤 액세스(CRA) 코딩 유형을 사용하여 코딩된 비디오 영역, 및/또는 iii) 끊어진 링크 액세스(BLA) 코딩 유형을 사용하여 코딩된 비디오 영역 중 적어도 하나를 포함하는 방법.

26. 20절에 있어서, 변형 모델 정보는, 시퀀스 레벨, 영상 레벨, 또는 적응형 파라미터 세트(APS)에 포함되는 방법.

27. 비디오 처리 방법에 있어서, 제1 비디오 영역과 제1 비디오 영역의 코딩된 표현 간의 전환을 위해, 제2 비디오 영역으로부터의 변형 정보가 규칙에 기초하여 전환에 사용 가능한지 여부를 결정하는 단계; 및 결정에 따라 전환을 수행하는 단계를 포함하는 방법.

28. 27절에 있어서, 변형 모델 정보는, 제1 도메인 및 제2 도메인에서의 표현, 및/또는 채도 비디오 유닛의 채도 잔차의 스케일링에 기초한 비디오 영역의 비디오 유닛의 재구성을 위해 사용되는 방법.

29. 27절에 있어서, 규칙은 코딩된 표현이 제1 비디오 영역과 제2 비디오 영역 사이의 특정 코딩 유형을 사용하여 코딩된 중간 비디오 영역(intervening video region)을 포함하는 경우에 제1 비디오 영역에 의한 변형 모델 정보의 사용을 허용하지 않는 방법.

30. 29절에 있어서, 중간 비디오 영역은 인트라(I) 코딩 유형을 사용하여 코딩되는 방법.

31. 29-30절 중 어느 하나에 있어서, 중간 비디오 영역은 순간 디코딩 리프레시)(IDR) 코딩 유형을 사용하여 코딩되는 방법.

32. 29-31절 중 어느 하나에 있어서, 중간 비디오 영역은 클린 랜덤 액세스(CRA) 코딩 유형을 사용하여 코딩되는 방법.

33. 29-32절 중 어느 하나에 있어서, 중간 비디오 영역은 인트라 랜덤 액세스 포인트(I-RAP) 코딩 유형을 사용하여 코딩되는 방법.

34. 33절에 있어서, 인트라 랜덤 액세스 포인트(I-RAP) 코딩 유형을 사용하여 코딩된 중간 비디오 영역은 i) 순간 디코딩 리프레시(IDR) 코딩 유형을 사용하여 코딩된 비디오 영역, ii) 클린 랜덤 액세스(CRA) 코딩 유형을 사용하여 코딩된 비디오 영역, 및/또는 iii) 끊어진 링크 액세스(BLA) 코딩 유형을 사용하여 코딩된 비디오 영역 중 적어도 하나를 포함하는 방법.

35. 27-34절 중 어느 하나에 있어서, 제1 비디오 영역, 제2 비디오 영역 및/또는 중간 비디오 영역은 슬라이스들에 대응하는 방법.

36. 27-35절 중 어느 하나에 있어서, 제1 비디오 영역, 제2 비디오 영역 및/또는 중간 비디오 영역은 영상들에

대응하는 방법.

37. 27-36절 중 어느 하나에 있어서, 제1 비디오 영역, 제2 비디오 영역 및/또는 중간 비디오 영역은 타일 그룹들에 대응하는 방법.

38. 비디오 처리 방법에 있어서, 현재 비디오 영역이 특정 코딩 유형을 사용하여 코딩되도록 비디오의 현재 비디오 영역과 현재 비디오 영역의 코딩된 표현 사이의 전환을 수행하는 단계를 포함하고, 코딩된 표현은, 비디오 영역 레벨에서 코딩된 표현의 플래그 값에 기초하여 코딩된 표현의 모델 정보를 조건부로 변형하도록 명시하는 형식 규칙을 준수하는 방법.

39. 38절에 있어서, 변형 모델 정보는 제1 도메인 및 제2 도메인에서의 표현 및/또는 채도 비디오 유닛의 스케일링 채도 잔차에 기초하여 비디오 영역의 비디오 유닛의 재구성을 위한 정보를 제공하는 방법.

40. 38절에 있어서, 플래그의 값이 제1 값을 가지는 경우에, 변형 모델 정보는 비디오 영역에서 시그널링되고, 그렇지 않으면 변형 모델 정보는 비디오 영역을 디코딩하기 전에 초기화되는 방법.

41. 38절에 있어서, 제1 값은 0 또는 1인 방법.

42. 38절에 있어서, 비디오 영역은 슬라이스 또는 영상 또는 타일 그룹이고, 특정 코딩 유형은 인트라(I) 코딩 유형인 방법.

43. 38절에 있어서, 비디오 영역은 슬라이스 또는 영상 또는 타일 그룹이고, 특정 코딩 유형은 순간 디코딩 리프레시(IDR) 코딩 유형인 방법.

44. 38절에 있어서, 비디오 영역은 슬라이스 또는 영상 또는 타일 그룹이고, 특정 코딩 유형은 클린 랜덤 액세스(CRA) 코딩 유형인 방법.

45. 38절에 있어서, 비디오 영역은 슬라이스 또는 영상 또는 타일 그룹이고, 특정 코딩 유형은 인트라 랜덤 액세스 포인트(I-RAP) 코딩 유형인 방법.

46. 45절에 있어서, 인트라 랜덤 액세스 포인트(I-RAP) 코딩 유형을 사용하여 코딩된 중간 비디오 영역은 i) 순간 디코딩 리프레시(IDR) 코딩 유형을 사용하여 코딩된 비디오 영역, ii) 클린 랜덤 액세스(CRA) 코딩 유형을 사용하여 코딩된 비디오 영역, 및/또는 iii) 끊어진 링크 액세스(BLA) 코딩 유형을 사용하여 코딩된 비디오 영역 중 적어도 하나를 포함하는 방법.

47. 1-46절 중 어느 하나에 있어서, 전환을 수행하는 단계는, 코딩된 표현으로부터 비디오를 생성하는 단계를 포함하는 방법.

48. 프로세서 및 그에 대한 명령어들이 있는 비일시적 메모리를 포함하는 비디오 시스템의 장치에 있어서, 프로세서에 의한 실행 시에 명령어들은 프로세서가 1절 내지 47절 중 어느 하나 이상의 방법을 실행하게 하는 장치.

49. 비일시적 컴퓨터 판독 가능 매체에 저장된 컴퓨터 프로그램 제품에 있어서, 1절 내지 47절 중 어느 하나 이상의 방법을 수행하기 위한 프로그램 코드를 포함하는 컴퓨터 프로그램 제품.

절들의 제3 세트는 예를 들어 예 8 및 9를 포함하는 이전 섹션에 나열된 개시된 기술의 특정 특징들 및 양태들을 설명한다.

1. 비디오 처리 방법에 있어서, 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 간의 전환을 수행하는 단계를 포함하고, 코딩된 표현은 하나 이상의 비디오 영역들 중 일부의 인-루프 변형(ILR)에 적용 가능한 변형 모델 정보를 포함하고, 변형 모델 정보는 제1 도메인 및 제2 도메인에서의 표현, 및/또는 채도 비디오 유닛의 채도 잔차의 스케일링에 기초하여 비디오 영역의 비디오 유닛의 재구성에 대한 정보를 제공하고 및 변형 모델 정보는 허용된 최대 빈 인덱스와 재구성에 사용될 최대 빈 인덱스 간의 차이를 명시하는 신택스 요소를 포함하는 파라미터 세트를 포함하고, 파라미터는 범위 내에 있는 방법.

2. 1절에 있어서, 신택스 요소는 0, 내지 재구성에 사용되는 허용된 최대 빈 인덱스와 최소 빈 인덱스 간의 차이의 범위 내에 있는 방법.

3. 1절에 있어서, 신택스 요소는 0 내지 허용된 최대 빈 인덱스의 범위 내에 있는 방법.

4. 1-3절 중 어느 하나에 있어서, 허용된 최대 빈 인덱스는, 15와 같은 방법.

5. 1 또는 4절에 있어서, 신택스 요소는 범위로 클리핑되는 방법.

6. 1절에 있어서, 최소 빈 인덱스는 최대 빈 인덱스보다 작거나 같은 방법.

7. 1절에 있어서, 최대 빈 인덱스는 최소 빈 인덱스 내지 허용된 최대 빈 인덱스의 범위로 클리핑되는 방법.

8. 1절에 있어서, 최소 빈 인덱스는 0 내지 최대 빈 인덱스의 범위로 클리핑되는 방법.

9. 비디오 처리 방법에 있어서, 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 간의 전환을 수행하는 단계를 포함하고, 코딩된 표현은 하나 이상의 비디오 영역들 중 일부의 인-루프 변형(ILR)에 적용 가능한 변형 모델 정보를 포함하고, 변형 모델 정보는 제1 도메인 및 제2 도메인에서의 표현, 및/또는 채도 비디오 유닛의 채도 잔차의 스케일링에 기초하여 비디오 영역의 비디오 유닛의 재구성에 대한 정보를 제공하고, 변형 모델 정보는 재구성에 사용될 최대 빈 인덱스를 포함하는 파라미터 세트를 포함하고, 및 최대 빈 인덱스는 재구성에 사용될 최소 빈 인덱스, 및 무부호 정수이고 최소 빈 인덱스 이후에 시그널링되는 신택스 요소의 합과 동일한 제1 값으로 도출되는 방법.

10. 9절에 있어서, 신택스 요소는 최대 빈 인덱스와 허용된 최대 빈 인덱스 간의 차이를 명시하는 방법.

11. 10절에 있어서, 신택스 요소는 0 내지 허용된 최소 빈 인덱스의 차이와 동일한 제2 값이 범위 내에 있는 방법.

12. 10절에 있어서, 신택스 요소는 1 내지 최소 빈 인덱스와 허용된 최대 빈 인덱스 간의 차이와 동일한 제2 값의 범위 내에 있는 방법.

13. 11절 또는 12절에 있어서, 신택스 요소는 범위로 클리핑되는 방법.

14. 9절에 있어서, 파라미터 세트의 신택스 요소들은 적합성 비트스트림(conformance bitstream)에서 요구되는 범위 내에 있는 방법.

15. 1절 내지 14절 중 어느 하나에 있어서, 전환을 수행하는 단계는, 코딩된 표현으로부터 비디오를 생성하는 단계를 포함하는 방법.

16. 프로세서 및 그에 대한 명령어들이 있는 비일시적 메모리를 포함하는 비디오 시스템의 장치에 있어서, 프로세서에 의한 실행 시에 명령어들은 프로세서가 1절 내지 15절 중 하나 이상에 기재된 비디오 처리 방법을 실행하게 하는 장치.

17. 비일시적 컴퓨터 판독 가능 매체에 저장된 컴퓨터 프로그램 제품에 있어서, 1절 내지 15절 중 하나 이상에 기재된 비디오 처리 방법을 수행하기 위한 프로그램 코드를 포함하는 컴퓨터 프로그램 제품.

절들의 제4 세트는 예를 들어 예10-17을 포함하는 이전 섹션에 나열된 개시된 기술들의 특정 특징들 및 양태들을 설명한다.

1. 비디오 처리 방법에 있어서, 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 간의 전환을 수행하는 단계를 포함하고, 코딩된 표현은 하나 이상의 비디오 영역들 중 일부의 인-루프 변형(ILR)에 적용 가능한 변형 모델 정보를 포함하고, 변형 모델 정보는 제1 도메인 및 제2 도메인에서의 표현, 및/또는 채도 비디오 유닛의 채도 잔차의 스케일링에 기초하여 비디오 영역의 비디오 유닛의 재구성에 대한 정보를 제공하고, 변형 모델 정보는 대응하는 빈으로부터 절대 델타 코드워드 값을 명시하는 제2 신택스 요소를 나타내기 위해 이용되는 다수의 비트들을 도출하는 제1 신택스 요소를 포함하는 파라미터 세트를 포함하고, 제1 신택스 요소는 임계값보다 작은 값을 가지는 방법.

2. 1절에 있어서, 제1 신택스 요소는 제2 신택스 요소를 나타내기 위해 이용되는 비트들의 수와 1 사이의 차이를 명시하는 방법.

3. 1절에 있어서, 임계값은 고정된 값을 가지는 방법.

4. 1절에 있어서 임계값은 값이 비트 깊이에 의존하는 변수를 가지는 방법.

5. 4절에 있어서, 임계값은 BitDepth - 1이고, BitDepth는 비트 깊이를 나타내는 방법.

6. 1절에 있어서, 제1 신택스 요소는 적합성 비트스트림 내에 있는 방법.

7. 비디오 처리 방법에 있어서, 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 간의 전환을 수행하는 단계를 포함하고, 코딩된 표현은 하나 이상의 비디오 영역들 중 일부의 인-루프 변형(ILR)에 적용 가능한 변형 모델 정보를 포함하고, 변형 모델 정보는 제1 도메인 및 제2 도메인에서의 표현, 및/또는 채도

비디오 유닛의 채도 잔차의 스케일링에 기초하여 비디오 영역의 비디오 유닛의 재구성에 대한 정보를 제공하고, 변형 모델 정보는 ILR에서 사용되는 i번째 빈의 기울기를 나타내고, (i-1)번째 파라미터에 기초한 값을 가지는 i번째 파라미터를 포함하는 파라미터 세트를 포함하고, i는 양의 정수인 방법.

8. 7절에 있어서, i번째 파라미터는 reshaper_model_min_bin_idx < = i <= reshaper_model_max_bin_idx의 경우에 (i-1)번째 파라미터로부터 예측되고, reshaper_model_min_bin_idx 및 reshaper_model_max_bin_index는 구성에 사용되는 최소 빈 인덱스 및 최대 빈 인덱스를 나타내는 방법.

9. 7절에 있어서, i번째 파라미터는 0과 동일한 i에 대한 OrgCW로부터 예측되는 방법.

10. 7절에 있어서, i번째 파라미터는 구성에 사용된 최소 빈 인덱스를 나타내는 reshaper_model_min_bin_idx의 값과 동일한 i에 대한 다른 파라미터로부터 예측되는 방법.

11. 비디오 처리 방법에 있어서, 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 간의 전환을 수행하는 단계를 포함하고, 코딩된 표현은 하나 이상의 비디오 영역들 중 일부의 인-루프 변형(ILR)에 적용 가능한 변형 모델 정보를 포함하고, 변형 모델 정보는 제1 도메인 및 제2 도메인에서의 표현, 및/또는 채도 비디오 유닛의 채도 잔차의 스케일링에 기초하여 비디오 영역의 비디오 유닛의 재구성에 대한 정보를 제공하고, ILR을 위해 사용되는 변형 모델 정보는 시그널링되지 않은 reshape_model_bin_delta_sign_CW [i]을 포함하는 파라미터 세트를 포함하고, reshape_model_bin_delta_sign_CW [i]는 항상 양수인 방법.

12. 11절에 있어서, 변수 CW[i]는 MinV 및 RspDeltaCW[i]의 합으로서 계산되는 방법.

13. 12절에 있어서, MinV는 32 또는 g(BitDepth)이고, BitDepth는 비트 깊이 값에 대응하는 방법.

14. 비디오 처리 방법에 있어서, 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 간의 전환을 수행하는 단계를 포함하고, 코딩된 표현은 하나 이상의 비디오 영역들 중 일부의 인-루프 변형(ILR)에 적용 가능한 변형 모델 정보를 포함하고, 변형 모델 정보는 제1 도메인 및 제2 도메인에서의 표현, 및/또는 채도 비디오 유닛의 채도 잔차의 스케일링에 기초하여 비디오 영역의 비디오 유닛의 재구성에 대한 정보를 제공하고, 변형 모델 정보는 비디오 영역의 색상 형식에 따른 스케일링을 위한 휘도 값들을 사용하기 위한 파라미터 invAvgLuma를 포함하는 파라미터 세트를 포함하는 방법.

15. 14절에 있어서, invAvgLuma는 invAvgLuma = Clip1Y((S_iS_j predMapSamples[(xCurr << scaleX) + i][(yCurr << scaleY) + j] + (cnt >>1)) / cnt로 계산되고, predMapSamples는 재구성된 휘도 샘플들을 나타내고, (xCurrY, yCurrY)는 현재 영상의 왼쪽 상단 채도 샘플에 대한 현재 채도 변환 블록의 왼쪽 상단 채도 샘플을 나타내고, (i, j)는 invAvgLuma를 도출하기 위해 관련된 휘도 샘플의 현재 채도 변환 블록의 왼쪽 상단 채도 샘플에 대한 상대적인 위치를 나타내고, cnt는 invAvgLuma를 도출하기 위해 포함된 휘도 샘플들의 수를 나타내는 방법.

16. 15절에 있어서, 4:2:0 형식에 대해 scaleX=scaleY=1인 방법.

17. 15절에 있어서, 4:4:4 형식에 대해 scaleX=scaleY=1인 방법.

18. 15절에 있어서, 4:2:2 형식에 대해 scaleX = 1 및 scaleY = 0인 방법.

19. 비디오 처리 방법에 있어서, 비디오의 현재 비디오 블록과 비디오의 코딩된 표현 간의 전환을 수행하는 단계를 포함하고, 전환은 재구성된 영상 휘도 샘플들을 수정된 재구성된 영상 휘도 샘플들로 변환하기 위한 영상 역 매핑 프로세스를 포함하고, 영상 역 매핑 프로세스는 상한 및 하한이 별도로 설정되는 클리핑을 포함하는 방법.

20. 19절에 있어서, 수정된 재구성된 영상 휘도 샘플들은, invLumaSample = Clip3(minVal, maxVal, invLumaSample), minVal, maxVal로서 계산되는 값 invLumaSample을 포함하는 방법. (피벗 수량)

21. 20절에 있어서, 구성에 사용될 최소 빈 인덱스가 0보다 큰 경우 minVal = T1 << (BitDepth - 8)이고 그렇지 않으면 minVal = 0인 방법.

22. 20절에 있어서, 구성에 사용될 최대 빈 인덱스가, 허용된 최대 빈 인덱스보다 작은 경우, maxVal = T2 << (BitDepth - 8)이고, 그렇지 않으면, maxVal = (1<< BitDepth)-1이고, BitDepth는 비트 깊이 값에 대응하는 방법.

23. 비디오 처리 방법에 있어서, 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 간의

전환을 수행하는 단계를 포함하고, 코딩된 표현은 하나 이상의 비디오 영역들 중 일부의 인-루프 변형(ILR)에 적용 가능한 변형 모델 정보를 포함하고, 변형 모델 정보는 제1 도메인 및 제2 도메인에서의 표현, 및/또는 채도 비디오 유닛의 채도 잔차의 스케일링에 기초하여 비디오 영역의 비디오 유닛의 재구성에 대한 정보를 제공하고, 및 변형 모델 정보는 Pivot[i]<=T가 되도록 제한되는 피벗 수량을 포함하는 파라미터 세트를 포함하는 방법.

24. 23절에 있어서, T는 T = (1<< BitDepth)-1로서 계산되고, BitDepth는 비트 깊이 값에 대응하는 방법.

25. 비디오 처리 방법에 있어서, 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 간의 전환을 수행하는 단계를 포함하고, 코딩된 표현은 인 루프 변형(ILR)에 적용 가능한 정보 및 제1 도메인 및 제2 도메인에서의 표현 및/또는 채도 비디오 유닛의 스케일링 채도 잔차에 기초하여 비디오 영역의 비디오 유닛의 재구성을 위한 파라미터들을 제공하고, 및 채도 양자화 파라미터(QP)는 그 값이 각 블록 또는 변환 유닛에 대해 도출되는 오프셋을 가지는 방법.

26. 25절에 있어서, 오프셋은 대표적인 휘도 값(repLumaVal)에 기초하여 도출되는 방법.

27. 26절에 있어서, 대표적인 휘도 값은 블록 또는 변환 유닛의 휘도 예측 값들의 일부 또는 전체를 사용하여 도출되는 방법.

28. 26절에 있어서, 대표적인 휘도 값은 블록 또는 변환 유닛의 부분 또는 전체 휘도 재구성된 값들을 사용하여 도출되는 방법.

29. 26절에 있어서, 대표적인 휘도 값은 블록 또는 변환 유닛의 휘도 예측 값들 또는 휘도 재구성된 값들의 일부 또는 전체의 평균으로서 도출되는 방법.

30. 26절에 있어서, ReshapePivot[idx] <= repLumaVal < ReshapePivot[idx + 1]이고, InvScaleCoeff[idx]는 오프셋을 도출하기 위해 이용되는 방법.

31. 30절에 있어서, 휘도 QP 오프셋은 argmin abs(2^(x/6 + shiftY) - InvScaleCoeff[idx]), x = -N..., M으로서 선택되고, N 및 M은 정수인 방법.

32. 30절에 있어서, 휘도 QP 오프셋은 argmin abs (1 - (2^(x/6 + shiftY) / InvScaleCoeff[idx])), x = -N..., M이고, N 및 M은 정수인 방법.

33. 30절에 있어서, 다양한 InvScaleCoeff[idx] 값들에 대해, 오프셋은 미리 계산되어 순람표에 저장되는 방법.

34. 비디오 처리 방법에 있어서, 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 간의 전환을 수행하는 단계를 포함하고, 코딩된 표현은 인 루프 변형(ILR)에 적용 가능한 정보 및 제1 도메인 및 제2 도메인에서의 표현 및/또는 채도 비디오 유닛의 스케일링 채도 잔차에 기초하여 비디오 영역의 비디오 유닛의 재구성을 위한 파라미터들을 제공하고, 및 휘도 양자화 파라미터(QP)는 그 값이 각 블록 또는 변환 유닛에 대해 도출되는 오프셋을 가지는 방법.

35. 34절에 있어서, 오프셋은 대표적인 휘도 값(repLumaVal)에 기초하여 도출되는 방법.

36. 34절에 있어서, 대표적인 휘도 값은 블록 또는 변환 유닛의 휘도 예측 값들의 일부 또는 전체를 사용하여 도출되는 방법.

37. 35 절에 있어서, 대표적인 휘도 값은 블록 또는 변환 유닛의 휘도 예측 값들의 일부 또는 전체의 평균으로서 도출되는 방법.

38. 35 절에 있어서, idx = repLumaVal / OrgCW이고 InvScaleCoeff[idx]는 오프셋을 도출하기 위해 이용되는 방법.

39. 38절에 있어서, 오프셋은 argmin abs (2^(x/6 + shiftY) - InvScaleCoeff[idx]), x = -N..., M으로 선택되고, N 및 M은 정수인 방법.

40. 38절에 있어서, 오프셋은 argmin abs (1 - (2^(x/6 + shiftY) / InvScaleCoeff[idx])), x = -N..., M으로 선택되고, N 및 M은 정수인 방법.

41. 38절에 있어서, 다양한 InvScaleCoeff[idx] 값들에 대해, 오프셋은 미리 계산되어 순람표에 저장되는 방

법.

42. 1-41절 중 어느 하나에 있어서, 전환을 수행하는 단계는, 코딩된 표현으로부터 비디오를 생성하는 단계를 포함하는 방법.

43. 프로세서 및 그에 대한 명령어들이 있는 비일시적 메모리를 포함하는 비디오 시스템의 장치에 있어서, 프로세서에 의한 실행 시에 명령어들은 프로세서가 1절 내지 42절 중 어느 하나 방법을 실행하게 하는 장치.

44. 비일시적 컴퓨터 판독 가능 매체에 저장된 컴퓨터 프로그램 제품에 있어서, 1절 내지 42절 중 어느 하나의 방법을 수행하기 위한 프로그램 코드를 포함하는 컴퓨터 프로그램 제품.

본 명세서에 기술된 개시 및 기타 솔루션, 예, 실시예, 모듈 및 기능적 동작은 디지털 전자 회로, 또는 이 문서에 개시된 구조와 그 구조적 등가물을 포함하는 컴퓨터 소프트웨어, 펌웨어 또는 하드웨어, 또는 이들 중 하나 이상의 조합으로 구현될 수 있다. 개시된 실시예 및 다른 실시예는 하나 이상의 컴퓨터 프로그램 제품, 즉, 데이터 처리 장치에 의해 실행되거나 데이터 처리 장치의 동작을 제어하기 위해 컴퓨터 판독 가능 매체 상에 인코딩된 컴퓨터 프로그램 명령어들의 하나 이상의 모듈로서 구현될 수 있다. 컴퓨터 판독가능 매체는 기계 판독가능 저장 장치, 기계 판독가능 저장 기판, 메모리 장치, 기계 판독가능 전파 신호에 영향을 미치는 물질의 구성, 또는 이들 중 하나 이상의 조합일 수 있다. "데이터 처리 장치"라는 용어는 예를 들어 프로그램 가능한 프로세서, 컴퓨터, 또는 다중 프로세서 또는 컴퓨터를 포함하여 데이터를 처리하기 위한 전체 장치, 장치 및 기계를 포함한다. 장치는 하드웨어에 추가하여 대응 컴퓨터 프로그램에 대한 실행 환경을 생성하는 코드, 예를 들어 프로세서 펌웨어, 프로토콜 스택, 데이터베이스 관리 시스템, 운영 체제, 또는 이들 중 하나 이상의 조합을 구성하는 코드를 포함할 수 있다. 전파된 신호는 인공적으로 생성된 신호, 예를 들어 기계 생성 전기, 광학 또는 전자기 신호로, 적절한 수신기 장치로의 전송을 위한 정보를 인코딩하기 위해 생성된다.

컴퓨터 프로그램(프로그램, 소프트웨어, 소프트웨어 응용 프로그램, 스크립트 또는 코드라고도 함)은 컴파일된 언어나 해석된 언어를 포함하여 전체 형태의 프로그래밍 언어로 작성할 수 있고, 컴퓨터 프로그램은 독립 실행형 프로그램 또는 모듈, 구성요소, 서브루틴 또는 컴퓨팅 환경에서 사용하기에 적합한 기타 유닛을 포함하여 임의의 형태로 배포될 수 있다. 컴퓨터 프로그램이 반드시 파일 시스템의 파일과 일치하는 것은 아니다. 프로그램은 대응 프로그램 전용 단일 파일 또는 여러 조정 파일(예: 하나 이상의 모듈, 서브 프로그램 또는 코드 부분을 저장하는 파일)에 다른 프로그램 또는 데이터(예: 마크업 언어 문서에 저장된 하나 이상의 스크립트)를 보유하는 파일의 일부에 저장할 수 있다. 컴퓨터 프로그램은 하나의 컴퓨터 또는 한 사이트에 있거나 여러 사이트에 분산되어 있고 통신 네트워크로 상호 연결된 여러 컴퓨터에서 실행되도록 배포될 수 있다.

이 문서에 설명된 프로세스 및 논리 흐름은 입력 데이터에 대해 작동하고 출력을 생성함으로써 기능을 수행하기 위해 하나 이상의 컴퓨터 프로그램을 실행하는 하나 이상의 프로그램 가능한 프로세서에 의해 수행될 수 있다.

프로세스 및 논리 흐름은 또한 특수 목적 논리 회로, 예를 들어 FPGA(필드 프로그램 가능 게이트 어레이(field programmable gate array)) 또는 ASIC(응용 특정 집적 회로(application specific integrated circuit))에 의해 수행될 수 있고 장치는 또한 구현될 수 있다.

컴퓨터 프로그램의 실행에 적합한 프로세서는 예를 들어 범용 및 특수 목적 마이크로프로세서, 및 임의의 종류의 디지털 컴퓨터의 임의의 하나 이상의 프로세서를 포함한다. 일반적으로 프로세서는 읽기 전용 메모리나 랜덤 액세스 메모리 또는 둘 다에서 명령어들과 데이터를 수신한다. 컴퓨터의 필수 요소는 명령어들을 수행하기 위한 프로세서와 명령어들 및 데이터를 저장하기 위한 하나 이상의 메모리 장치이다. 일반적으로, 컴퓨터는 또한 데이터를 저장하기 위한 하나 이상의 대용량 저장 장치, 예를 들어 자기, 광자기 디스크, 또는 광 디스크로부터 데이터를 수신하거나 이들로 데이터를 전송하거나 둘 전체를 포함하거나 작동 가능하게 연결된다. 그러나 컴퓨터에는 그러한 장치가 필요하지 않는다. 컴퓨터 프로그램 명령어들 및 데이터를 저장하기에 적합한컴퓨터 판독 가능 매체 및 데이터는 전체 형태의 비휘발성 메모리, 매체 및 예를 들어 반도체 메모리 장치를 포함하는 메모리 장치를 포함한다(예를 들어, EPROM, EEPROM 및 플래시 메모리 장치; 자기 디스크, 예를 들어 내부 하드 디스크 또는 이동식 디스크; 자기 광 디스크; 및 CD ROM 및 DVD-ROM 디스크). 프로세서와 메모리는 특수 목적 논리 회로에 의해 보완되거나 통합될 수 있다.

이 특허 문서에는 많은 세부 사항이 포함되어 있지만, 이는 청구 대상 또는 청구될 수 있는 항목의 범위에 대한 제한으로 해석되어서는 안 되며 오히려 특정 기술의 특정 실시예에 특정할 수 있는 기능에 대한 설명으로 해석되어서는 안 된다. 별도의 실시예와 관련하여 이 특허 문서에 설명된 특정 특징은 단일 실시예에서 조합하여 구현될 수도 있다. 역으로, 단일 실시예의 맥락에서 설명된 다양한 특징은 또한 개별적으로 또는 임의의 적절

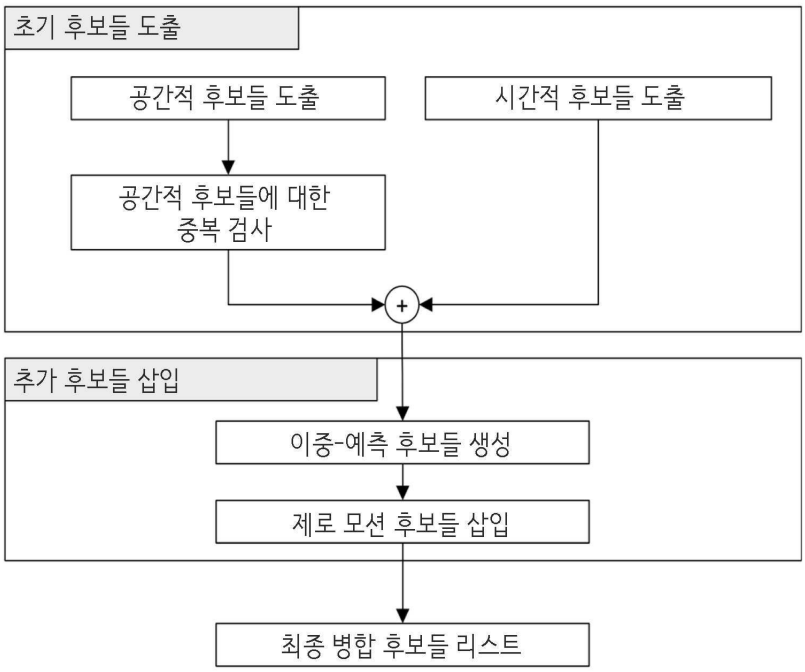
한 서브 조합으로 다중 실시예에서 구현될 수 있다. 게다가, 특징은 특정 조합으로 작용하는 것으로 위에서 설명될 수 있고 초기에 그렇게 주장될 수 있지만, 청구된 조합의 하나 이상의 특징은 일부 경우에 조합에서 제거될 수 있으며 청구된 조합은 서브 조합 또는 서브 조합의 변형에 관한 것일 수 있다.

유사하게, 작업은 도면에 특정 순서로 묘사되어 있지만, 이는 바람직한 결과를 달성하기 위해 그러한 작업이 표시된 특정 순서 또는 순차적인 순서로 수행되거나 전체 예시된 작업이 수행되어야 함을 요구하는 것으로 이해되어서는 안 된다. 더욱이, 이 특허 문서에 설명된 실시예에서 다양한 시스템 구성요소의 분리가 전체 실시예에서 그러한 분리를 요구하는 것으로 이해되어서는 안 된다.

소수의 구현들 및 예시들만이 설명되고, 이 특허 문서에 설명되고 예시된 것을 기반으로 다른 구현들, 개선들 및 변형들이 이루어질 수 있다.

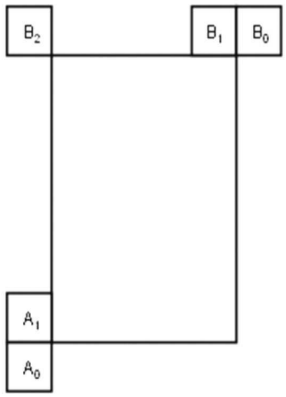
도면

도면1



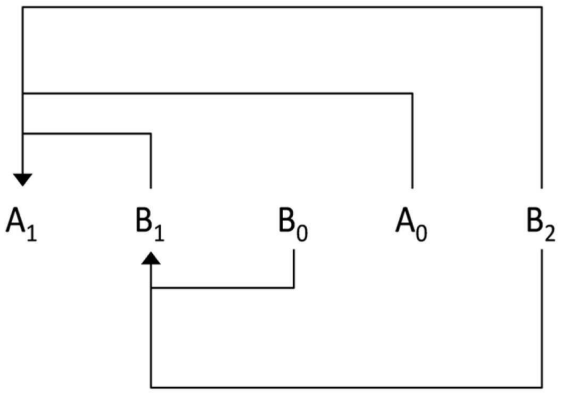
병합 후보들 리스트 구성에 대한 도출 프로세스

도면2



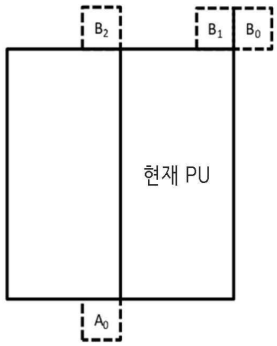
공간적 병합 후보들의 위치들

도면3



공간적 병합 후보들의 중복 검사를 위해 고려되는 후보 쌍들

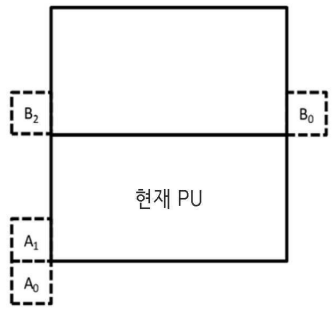
도면4a



Nx2N 및 2NxN 분할들의 제2 PU 위치들

Nx2N의 제2 PU

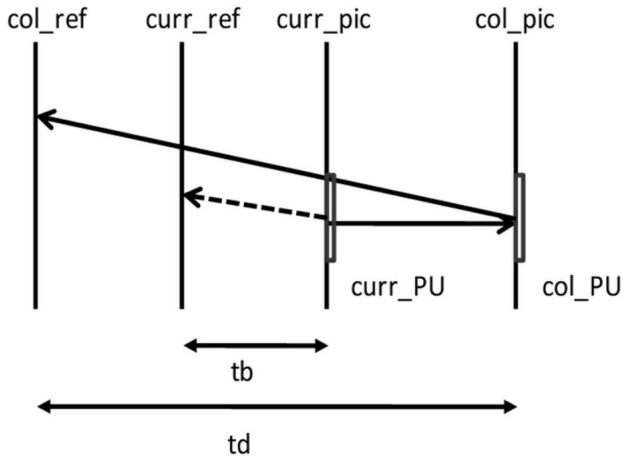
도면4b



Nx2N 및 2NxN 분할들의 제2 PU 위치들

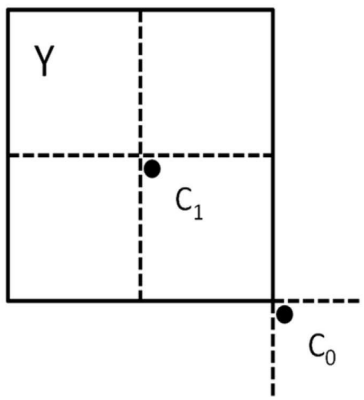
2NxN의 제2 PU

도면5



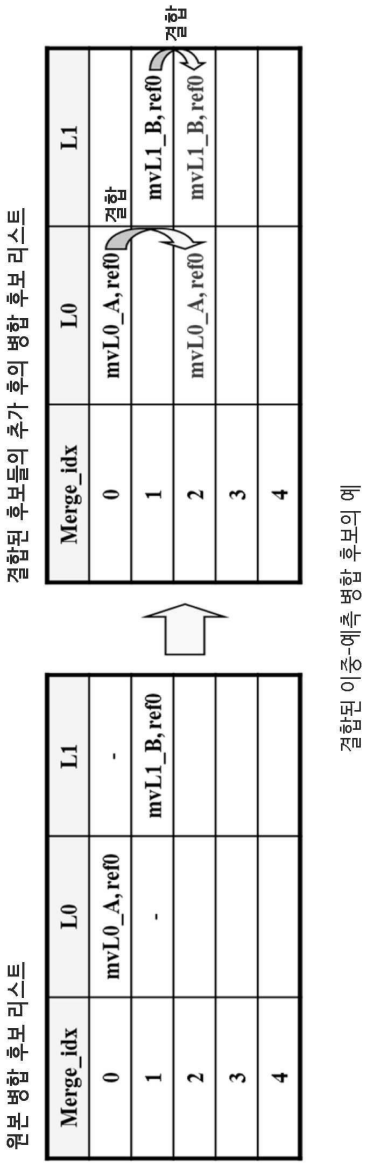
시간적 병합 후보에 대한 모션 벡터 스케일링의 예시

도면6

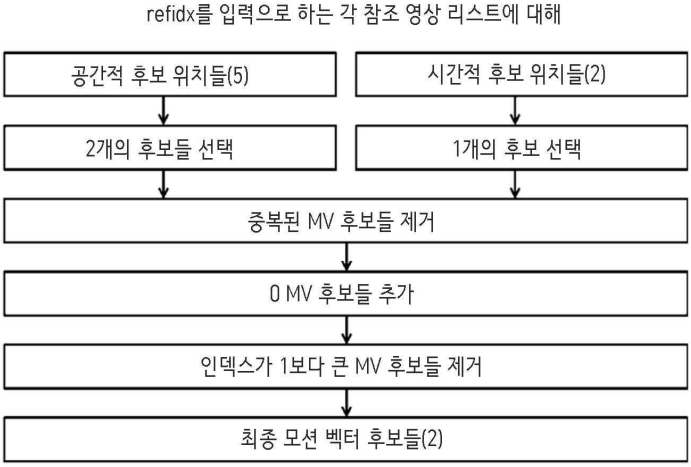


시간적 병합 후보 C0 및 C1의 후보 위치들

도면7

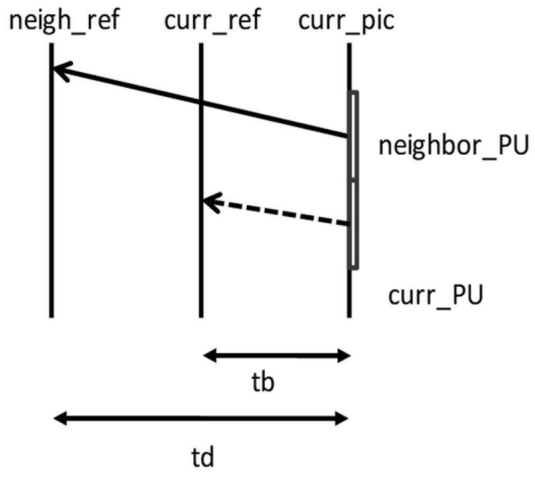


도면8



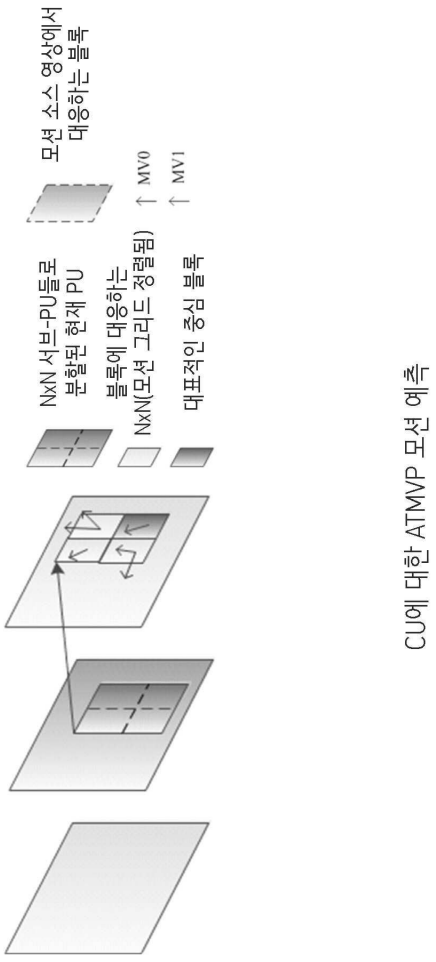
모션 벡터 예측 후보들에 대한 도출 프로세스

도면9



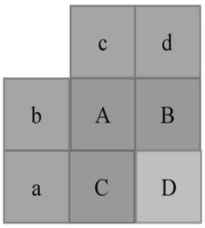
공간적 모션 벡터 후보에 대한 모션 벡터 스케일링의 예시

도면10



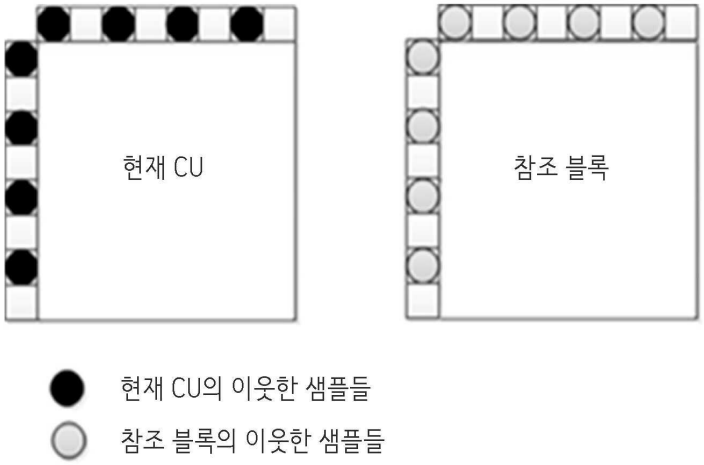
CU에 대한 ATMVP 모션 예측

도면11



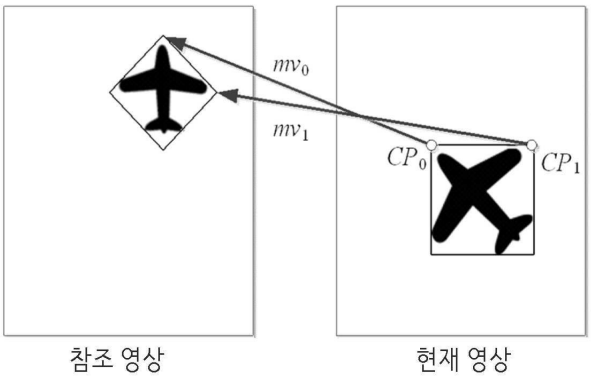
4개의 서브-블록들(A-D)과 하나의 CU 및 이웃한 블록들(a-d)의 예

도면12



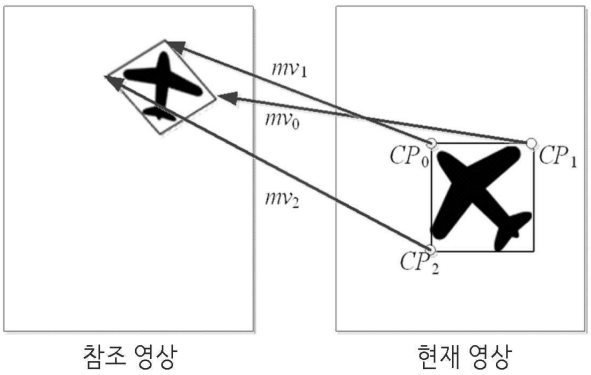
IC 파라미터들을 도출하는 데 사용되는 이웃한 샘플들

도면13a



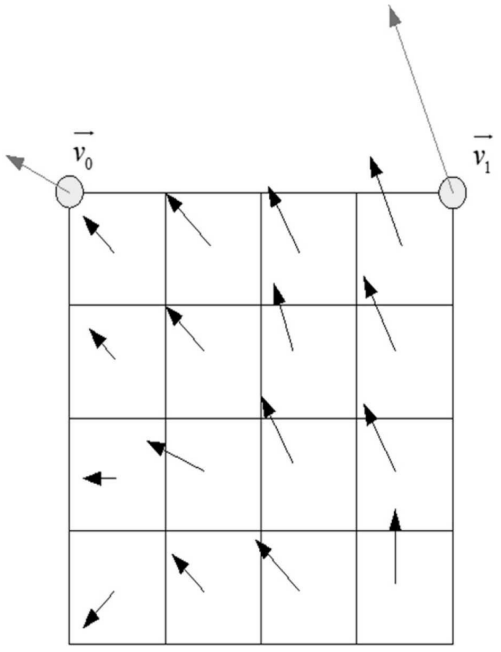
4 파라미터 아핀

도면13b



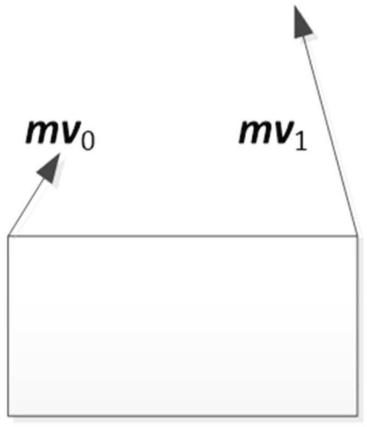
6 파라미터 아핀

도면14



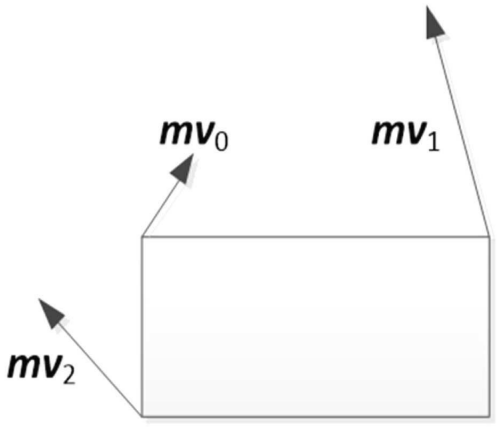
서브-블록 당 아핀 MVF의 예

도면15a



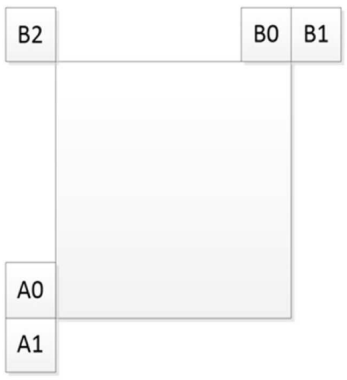
4-파라미터 아핀 모델

도면15b



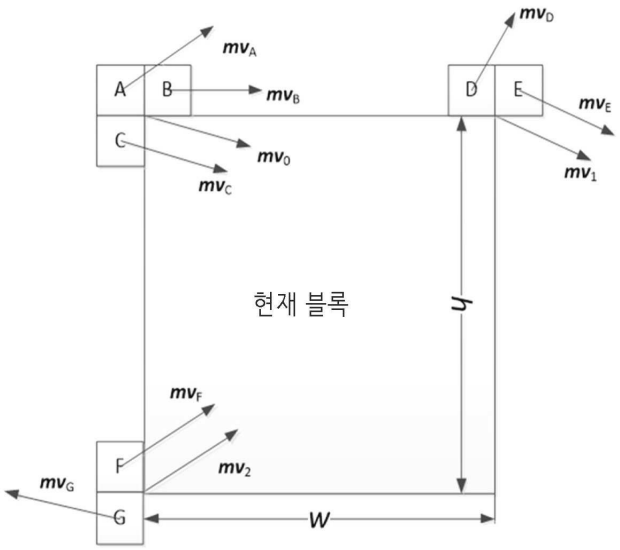
6-파라미터 아핀 모델

도면16



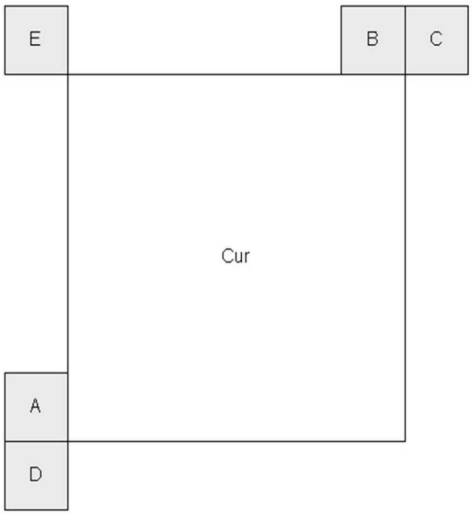
상속된 아핀 후보들에 대한 AF_INTER의 MVP

도면17



구성된 아핀 후보들에 대한 AF_INTER의 MVP

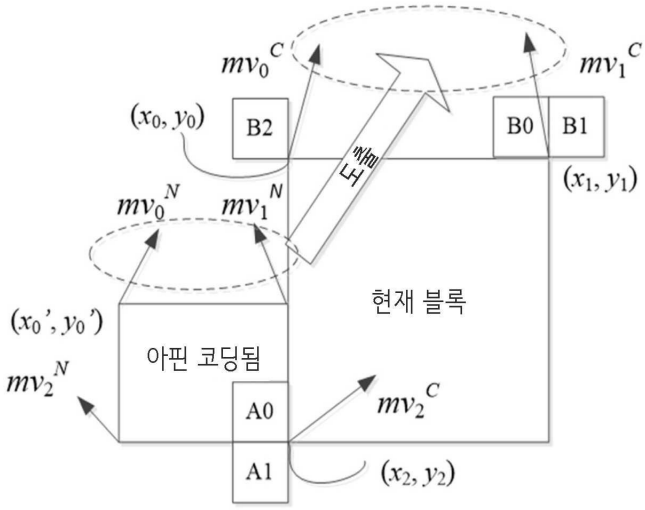
도면18a



AF_MERGE의 후보들

5개의 이웃한 블록들

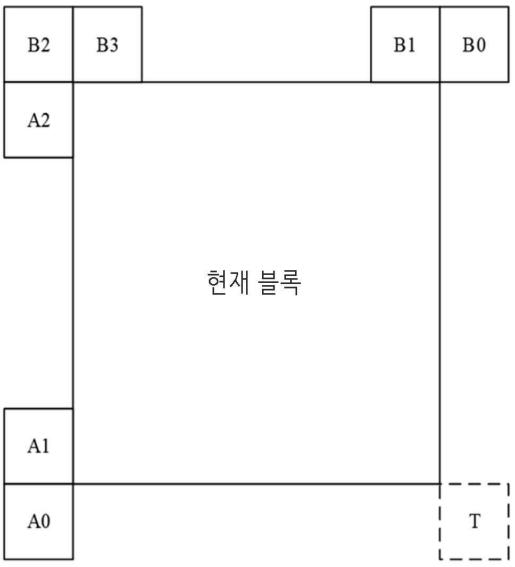
도면18b



AF_MERGE의 후보들

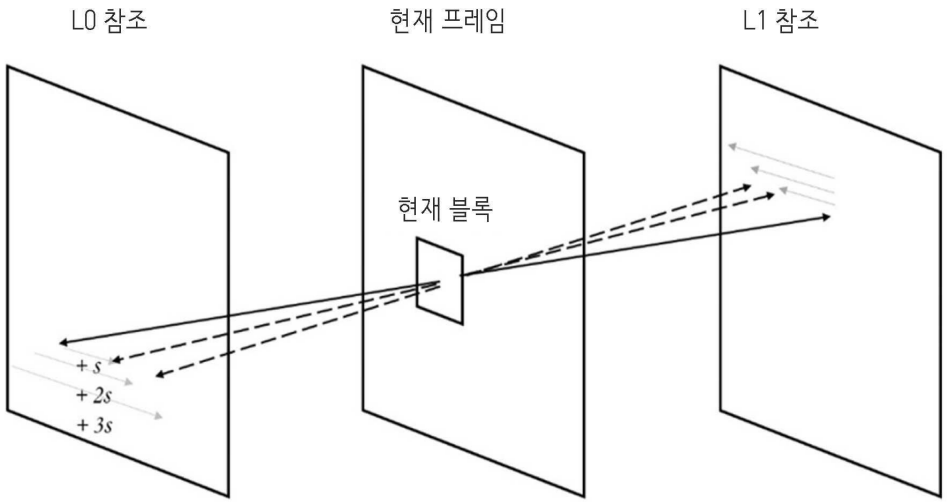
CPMV 예측 변수 도출

도면19



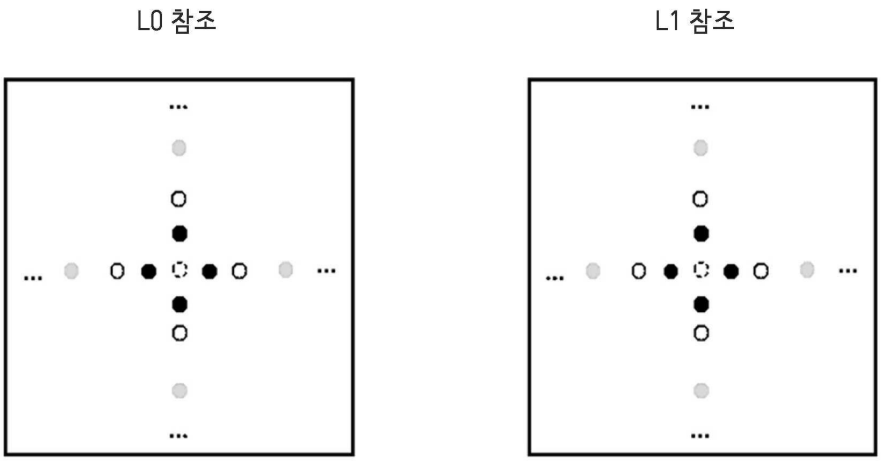
아핀 병합 모드에 대한 후보들 위치

도면20



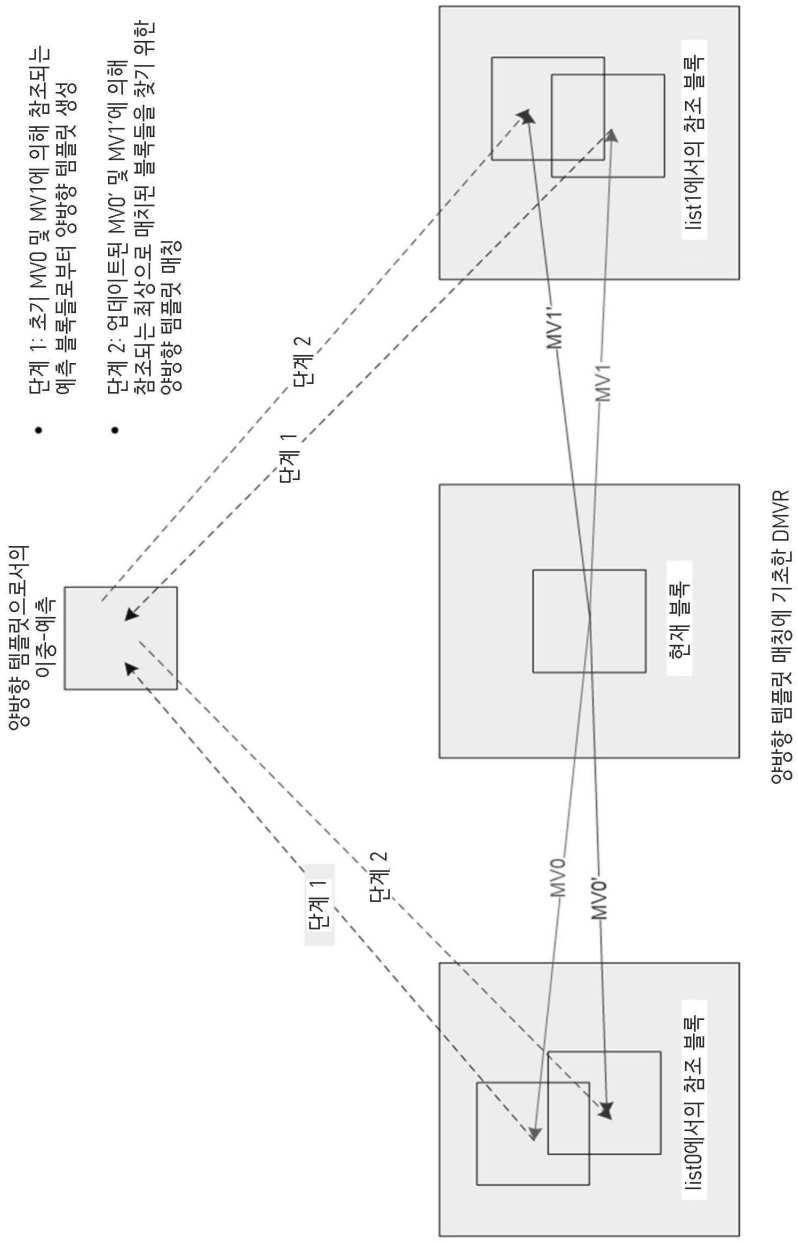
UMVE 탐색 프로세스

도면21

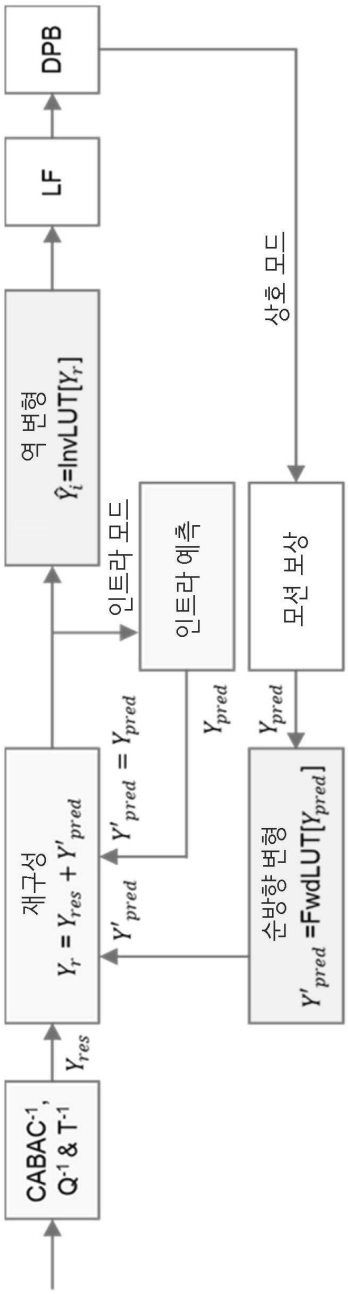


UMVE 탐색점

도면22

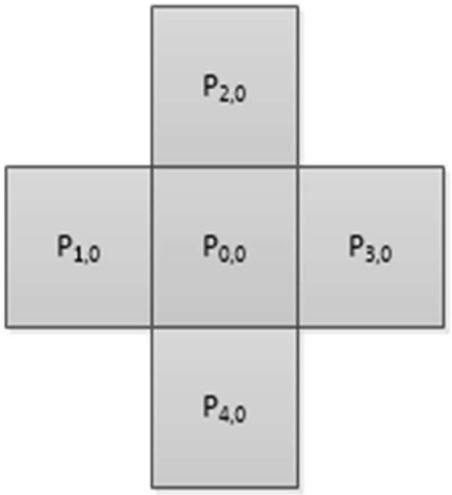

양방향 템플릿으로서의
이중-예측
• 단계 1: 초기 MV0 및 MV1에 의해 참조되는
예측 블록들로부터 양방향 템플릿 생성
• 단계 2: 업데이트된 MV0' 및 MV1'에 의해
참조되는 최상으로 매치된 블록들을 찾기 위한
양방향 템플릿 매칭
단계 1
단계 2
단계 1
단계 2
MV0
MV0'
MV1'
MV1
list0에서의 참조 블록
현재 블록
list1에서의 참조 블록
양방향 템플릿 매칭에 기초한 DMVR

도면23



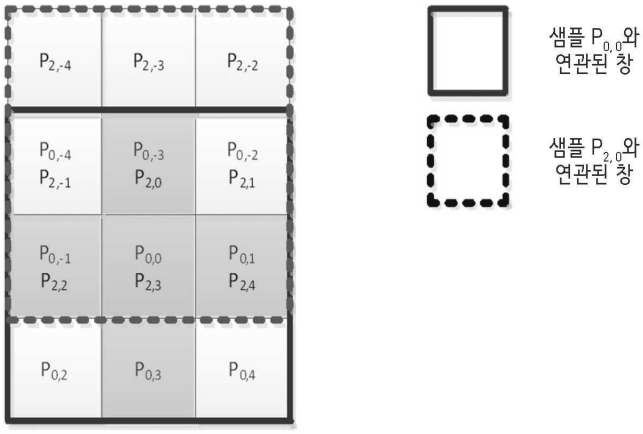
변형을 이용한 디코딩 흐름의 흐름도

도면24



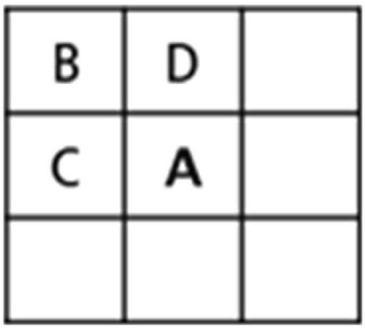
양방향 필터에서 이용되는 이웃한 샘플들

도면25



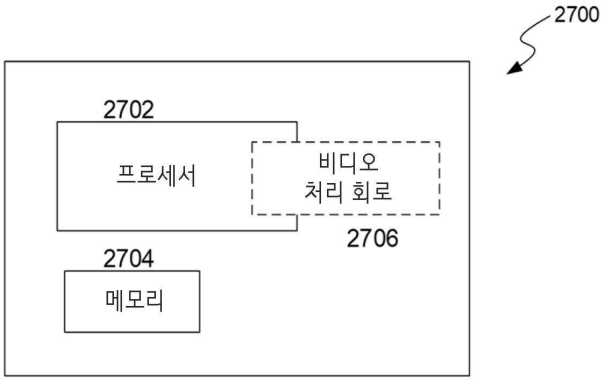
가중치 계산에 이용되는 2개의 샘플들을 포함하는 창들

도면26

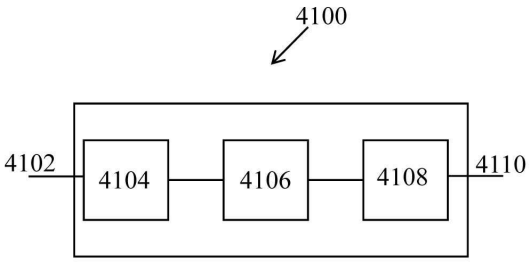


스캔 패턴. A-현재 픽셀, BCD-주변 픽셀들

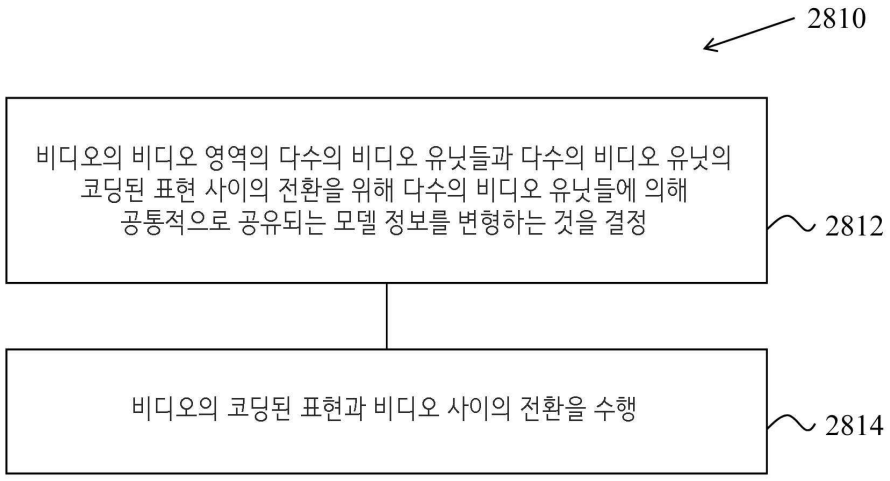
도면27a



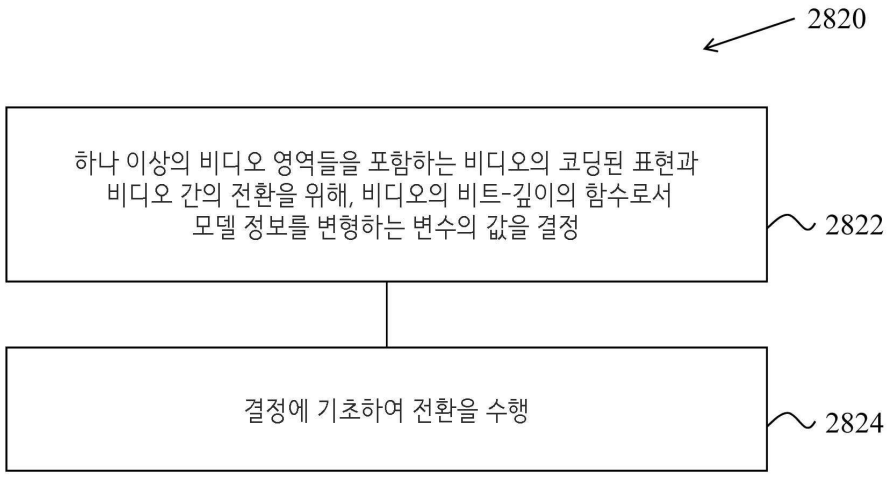
도면27b



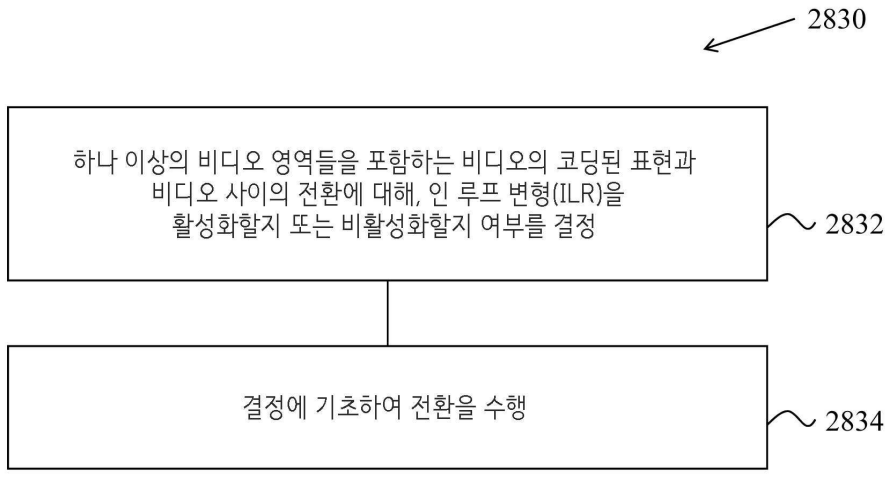
도면28a



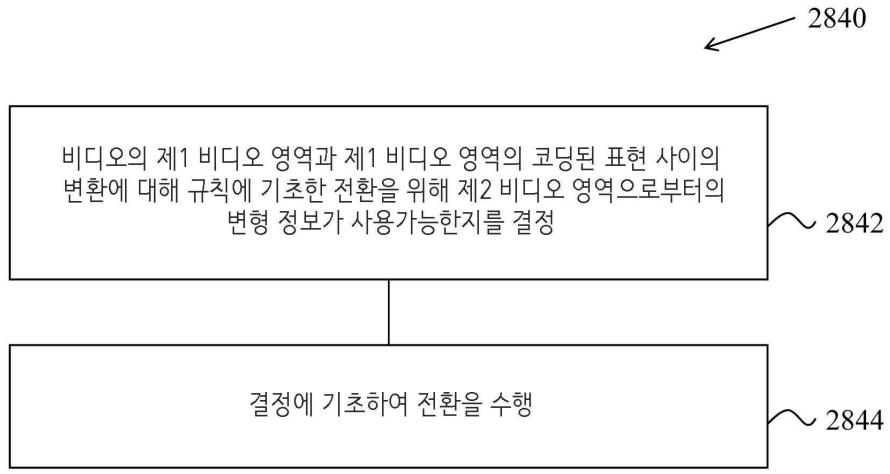
도면28b



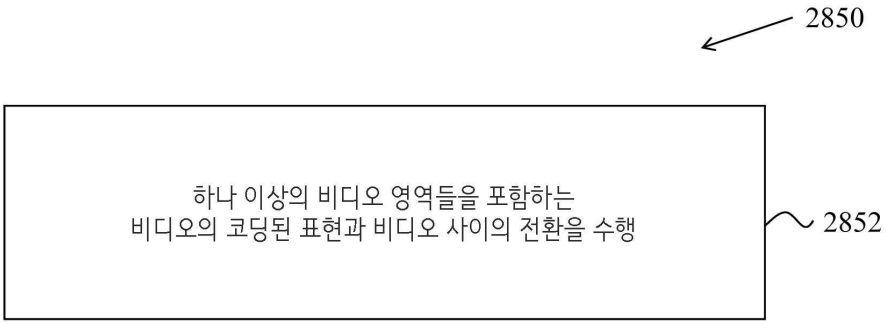
도면28c



도면28d


2840
비디오의 제1 비디오 영역과 제1 비디오 영역의 코딩된 표현 사이의 변환에 대해 규칙에 기초한 전환을 위해 제2 비디오 영역으로부터의 변형 정보가 사용가능한지를 결정
2842
결정에 기초하여 전환을 수행
2844

도면28e


2850
하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 사이의 전환을 수행
2852