(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2023년11월24일
(11) 등록번호 10-2606010
(24) 등록일자 2023년11월21일

(51) 국제특허분류(Int. Cl.)
*H04N 19/82* (2014.01) *H04N 19/117* (2014.01)
*H04N 19/132* (2014.01) *H04N 19/42* (2014.01)
*H04N 19/70* (2014.01)
(52) CPC특허분류
*H04N 19/82* (2015.01)
*H04N 19/117* (2015.01)
(21) 출원번호 10-2021-7028519
(22) 출원일자(국제) 2020년03월24일
심사청구일자 2023년03월02일
(85) 번역문제출일자 2021년09월06일
(65) 공개번호 10-2021-0135513
(43) 공개일자 2021년11월15일
(86) 국제출원번호 PCT/CN2020/080827
(87) 국제공개번호 WO 2020/192644
국제공개일자 2020년10월01일
(30) 우선권주장
PCT/CN2019/079395 2019년03월24일 중국(CN)

(73) 특허권자
**베이징 바이트댄스 네트워크 테크놀로지 컴퍼니, 리미티드**
중국, 베이징 100041, 스징산 디스트릭트, 스싱로드, 넘버 30, 넘버 3빌딩, 2층, 룸 비-0035
**바이트댄스 아이엔씨**
미국, 90066 캘리포니아 로스엔젤레스 스위트룸 137호 6층 제퍼슨 블러바드 웨스트 12655
(72) 발명자
**장, 리**
미국 캘리포니아 90066 로스엔젤레스 웨스트 제퍼슨 블러바드 12655번지 6층 스위트 137호
**장, 카이**
미국 캘리포니아 90066 로스엔젤레스 웨스트 제퍼슨 블러바드 12655번지 6층 스위트 137호
*(뒷면에 계속)*
(74) 대리인
**특허법인 무한**

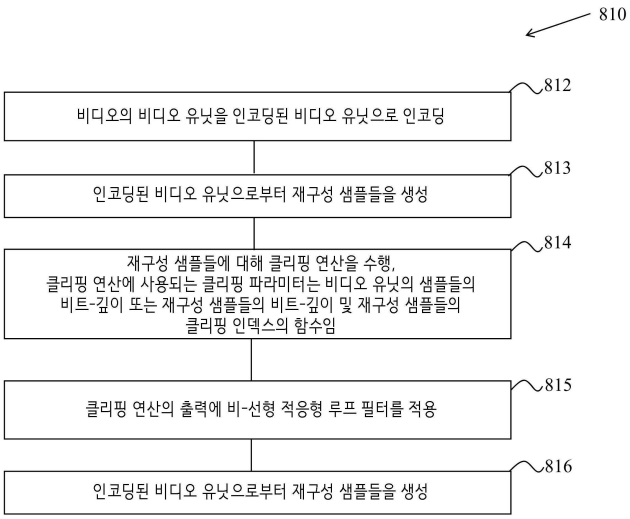
전체 청구항 수 : 총 15 항 심사관 : 김건우

(54) 발명의 명칭 **비디오 처리에서 비-선형 적응형 루프 필터링**

## (57) 요 약

비디오 처리를 위한 장치, 시스템 및 방법이 설명된다. 예시적인 양태에서, 비디오 처리를 위한 방법은 비디오의 비디오 유닛을 인코딩된 비디오 유닛으로 인코딩하는 단계; 인코딩된 비디오 유닛으로부터 재구성 샘플들을 생성하는 단계; 재구성 샘플들에 대해 클리핑 연산을 수행하는 단계 - 클리핑 연산에 사용되는 클리핑 파라미터는 비디오 유닛의 샘플들의 비트-깊이 또는 재구성 샘플들의 비트-깊이 및 재구성 샘플들의 클리핑 인덱스의 함수임 -; 클리핑 연산의 출력에 비-선형 적응형 루프 필터를 적용하는 단계; 및 인코딩된 비디오 유닛을 사용하여 비디오의 코딩된 표현을 생성하는 단계를 포함한다.

## 대 표 도



(52) CPC특허분류
*H04N 19/132* (2015.01)
*H04N 19/42* (2015.01)
*H04N 19/70* (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 현재 비디오 영역에 대해 비-선형 적응형 루프 필터링 연산(non-linear adaptive loop filtering operation)이 적용되었는지 결정하는 단계 - 상기 현재 비디오 영역은 코딩 트리 블록임 -;

상기 현재 비디오 영역에 대한 적어도 하나의 제1 필터링 인덱스를 도출하고 상기 적어도 하나의 제1 필터링 인덱스에 기초하여 하나 이상의 제1 필터링 파라미터들을 도출하는 단계;

상기 적어도 하나의 제1 필터링 인덱스에 기초하여 클리핑 파라미터 세트로부터 하나 이상의 제1 클리핑 파라미터들을 도출하는 단계;

상기 하나 이상의 제1 클리핑 파라미터들에 기초하여 상기 비-선형 적응형 루프 필터링 연산의 일부인 클리핑 연산을 수행하는 단계 - 상기 하나 이상의 제1 클리핑 파라미터들은 상기 하나 이상의 제1 필터링 파라미터들을 적용하기 전에 상기 현재 비디오 영역의 현재 샘플 값의 참조 샘플 값들과 상기 현재 샘플 값 사이의 차이들을 클리핑하기 위해 사용됨 -; 및

상기 비-선형 적응형 루프 필터링 연산에 기초하여 상기 현재 비디오 영역과 상기 비디오의 비트스트림 사이의 전환을 수행하는 단계를 포함하고,

상기 클리핑 파라미터 세트는 미리 정의된 테이블에 기초한 클리핑 인덱스 및 비트-깊이(bit-depth)에 기초하여 구성되고,

상기 미리 정의된 테이블에서, 주어진 비트-깊이에 대한 제1 클리핑 인덱스에 대응하는 하나의 클리핑 파라미터의 값은 다른 비트-깊이에 대한 상기 제1 클리핑 인덱스에 대응하는 다른 클리핑 파라미터의 값에 기초하여 도출되는,

비디오 데이터 처리 방법.

### 청구항 2

제1항에 있어서,

상기 클리핑 파라미터 세트의 클리핑 파라미터 값들을 도출하기 위해 시프팅 연산(shifting operation)이 사용되는,

비디오 데이터 처리 방법.

### 청구항 3

제1항에 있어서,

상기 클리핑 인덱스는,

상기 비트스트림의 필드와 연관되는,

비디오 데이터 처리 방법.

### 청구항 4

제1항에 있어서,

상기 클리핑 인덱스는,

미리-정의된 규칙을 이용하여 결정되는,

비디오 데이터 처리 방법.

### 청구항 5

제1항에 있어서,

상기 클리핑 파라미터 세트에 대해, 상기 클리핑 인덱스의 주어진 값 하에서 다른 비트-깊이를 위해 다른 클리핑 파라미터 값들이 사용되는,

비디오 데이터 처리 방법.

**청구항 6**

제1항에 있어서,

상기 미리 정의된 테이블에서, 상기 하나의 클리핑 파라미터의 값은,

상기 다른 클리핑 파라미터의 값에 시프팅 연산을 적용하는 것에 기초하여 도출되는,

비디오 데이터 처리 방법.

**청구항 7**

제1항에 있어서,

상기 적어도 하나의 제1 필터링 인덱스는,

상이한 방향들에서의 다중 샘플 차이들(multiple sample differences)에 기초하여 도출되는,

비디오 데이터 처리 방법.

**청구항 8**

제7항에 있어서,

상기 현재 비디오 영역은 다중 MXM 비디오 영역으로 분할되고,

상이한 방향들에서의 상기 다중 샘플 차이들은 모든 MXM 비디오 서브-영역에 대해 도출되고,

M은 2 또는 4인,

비디오 데이터 처리 방법.

**청구항 9**

제8항에 있어서,

상이한 방향들에서의 상기 다중 샘플 차이들은,

1:N 서브샘플링 레이트에 기초하여 도출되고,

N은 1보다 큰,

비디오 데이터 처리 방법.

**청구항 10**

제1항에 있어서,

상기 하나 이상의 제1 필터링 파라미터들 및 상기 하나 이상의 제1 클리핑 파라미터들은,

동일한 적응 파라미터 세트 식별(adaptation parameter set identification)에 기초하여 도출되는,

비디오 데이터 처리 방법.

**청구항 11**

제1항에 있어서,

상기 전환은,

상기 현재 비디오 영역을 상기 비트스트림으로 인코딩하는 것을 포함하는,

비디오 데이터 처리 방법.

**청구항 12**

제1항에 있어서,

상기 전환은,

상기 비트스트림으로부터 상기 현재 비디오 영역을 디코딩하는 것을 포함하는,

비디오 데이터 처리 방법.

**청구항 13**

명령어들을 포함하는 비-일시적인 메모리(non-transitory memory) 및 프로세서를 포함하는 비디오 데이터 처리 장치에 있어서, 상기 명령어들은 상기 프로세서에 의해 실행될 때 상기 프로세서가,

비디오의 현재 비디오 영역에 대해 비-선형 적응형 루프 필터링 연산이 적용되었는지 결정하고 - 상기 현재 비디오 영역은 코딩 트리 블록임 -;

상기 현재 비디오 영역에 대한 적어도 하나의 제1 필터링 인덱스를 도출하고 상기 적어도 하나의 제1 필터링 인덱스에 기초하여 하나 이상의 제1 필터링 파라미터들을 도출하고;

상기 적어도 하나의 제1 필터링 인덱스에 기초하여 클리핑 파라미터 세트로부터 하나 이상의 제1 클리핑 파라미터들을 도출하고;

상기 하나 이상의 제1 클리핑 파라미터들에 기초하여 상기 비-선형 적응형 루프 필터링 연산의 일부인 클리핑 연산을 수행하고- 상기 하나 이상의 제1 클리핑 파라미터들은 상기 하나 이상의 제1 필터링 파라미터들을 적용하기 전에 상기 현재 비디오 영역의 현재 샘플 값의 참조 샘플 값들과 상기 현재 샘플 값 사이의 차이들을 클리핑하기 위해 사용됨 -; 및

상기 비-선형 적응형 루프 필터링 연산에 기초하여 상기 현재 비디오 영역과 상기 비디오의 비트스트림 사이의 전환을 수행하도록 하고,

상기 클리핑 파라미터 세트는 미리 정의된 테이블에 기초한 비트-깊이(bit-depth) 및 클리핑 인덱스에 기초하여 구성되고,

상기 미리 정의된 테이블에서, 주어진 비트-깊이에 대한 제1 클리핑 인덱스에 대응하는 하나의 클리핑 파라미터의 값은 다른 비트-깊이에 대한 상기 제1 클리핑 인덱스에 대응하는 다른 클리핑 파라미터의 값에 기초하여 도출되는,

비디오 데이터 처리 장치.

**청구항 14**

프로세서가

비디오의 현재 비디오 영역에 대해 비-선형 적응형 루프 필터링 연산이 적용되었는지 결정하고 - 상기 현재 비디오 영역은 코딩 트리 블록임 -;

상기 현재 비디오 영역에 대한 적어도 하나의 제1 필터링 인덱스를 도출하고 상기 적어도 하나의 제1 필터링 인덱스에 기초하여 하나 이상의 제1 필터링 파라미터들을 도출하고;

상기 적어도 하나의 제1 필터링 인덱스에 기초하여 클리핑 파라미터 세트로부터 하나 이상의 제1 클리핑 파라미터들을 도출하고;

상기 하나 이상의 제1 클리핑 파라미터들에 기초하여 상기 비-선형 적응형 루프 필터링 연산의 일부인 클리핑 연산을 수행하고- 상기 하나 이상의 제1 클리핑 파라미터들은 상기 하나 이상의 제1 필터링 파라미터들을 적용하기 전에 상기 현재 비디오 영역의 현재 샘플 값의 참조 샘플 값들과 상기 현재 샘플 값 사이의 차이들을 클리핑하기 위해 사용됨 -; 및

상기 비-선형 적응형 루프 필터링 연산에 기초하여 상기 현재 비디오 영역과 상기 비디오의 비트스트림 사이의

전환을 수행

하도록 하는 명령어들(instructions)을 저장하고,

상기 클리핑 파라미터 세트는 미리 정의된 테이블에 기초한 비트-깊이(bit-depth) 및 클리핑 인덱스에 기초하여 구성되고,

상기 미리 정의된 테이블에서, 주어진 비트-깊이에 대한 제1 클리핑 인덱스에 대응하는 하나의 클리핑 파라미터의 값은 다른 비트-깊이에 대한 상기 제1 클리핑 인덱스에 대응하는 다른 클리핑 파라미터의 값에 기초하여 도출되는,

비-일시적인 컴퓨터 판독 가능한 저장 매체.

**청구항 15**

비디오 처리 장치에 의해 수행되는 방법에 의해 생성된 비디오의 비트스트림을 저장하는 비-일시적인 컴퓨터 판독 가능한 기록 매체에 있어서,

상기 방법은,

비디오의 현재 비디오 영역에 대해 비-선형 적응형 루프 필터링 연산이 적용되었는지 결정하는 단계 - 상기 현재 비디오 영역은 코딩 트리 블록임 -;

상기 현재 비디오 영역에 대한 적어도 하나의 제1 필터링 인덱스를 도출하고 상기 적어도 하나의 제1 필터링 인덱스에 기초하여 하나 이상의 제1 필터링 파라미터들을 도출하는 단계;

상기 적어도 하나의 제1 필터링 인덱스에 기초하여 클리핑 파라미터 세트로부터 하나 이상의 제1 클리핑 파라미터들을 도출하는 단계;

상기 하나 이상의 제1 클리핑 파라미터들에 기초하여 상기 비-선형 적응형 루프 필터링 연산의 일부인 클리핑 연산을 수행하는 단계 - 상기 하나 이상의 제1 클리핑 파라미터들은 상기 하나 이상의 제1 필터링 파라미터들을 적용하기 전에 상기 현재 비디오 영역의 현재 샘플 값의 참조 샘플 값들과 상기 현재 샘플 값 사이의 차이들을 클리핑하기 위해 사용됨 -; 및

상기 클리핑 연산에 기초하여 상기 비디오의 비트스트림을 생성하는 단계

를 포함하고,

상기 클리핑 파라미터 세트는 미리 정의된 테이블에 기초한 비트-깊이(bit-depth) 및 클리핑 인덱스에 기초하여 구성되고,

상기 미리 정의된 테이블에서, 주어진 비트-깊이에 대한 제1 클리핑 인덱스에 대응하는 하나의 클리핑 파라미터의 값은 다른 비트-깊이에 대한 상기 제1 클리핑 인덱스에 대응하는 다른 클리핑 파라미터의 값에 기초하여 도출되는,

비-일시적인 컴퓨터 판독 가능한 기록 매체.

**청구항 16**

삭제

**청구항 17**

삭제

**청구항 18**

삭제

**청구항 19**

삭제

**청구항 20**

삭제

**청구항 21**

삭제

**청구항 22**

삭제

**청구항 23**

삭제

**청구항 24**

삭제

**청구항 25**

삭제

**청구항 26**

삭제

## *발명의 설명*

### *기 술 분 야*

[0001] 본 출원은 2019년 3월 24일에 출원된 국제 특허 출원 번호 PCT/CN2019/079395에 대한 우선권 및 이익을 주장하는 2020년 3월 24일에 출원된 국제 특허 출원 PCT/CN2020/080827의 국내 단계이다. 법에 따른 모든 목적들을 위해, 전술한 출원들의 전체는 본 출원 개시의 일부로서 참고로 포함된다.

[0002] 이 특허 문서는 비디오 처리 기술, 장치 및 시스템에 관한 것이다

### *배 경 기 술*

[0003] 비디오 압축(video compression)의 발전에도 불구하고, 디지털 비디오는 여전히 인터넷 및 기타 디지털 통신 네트워크들에서 가장 큰 대역폭 사용을 차지한다. 비디오를 수신하고 디스플레이할 수 있는 연결된 사용자 장치들의 수가 증가함에 따라 디지털 비디오 사용에 대한 대역폭 수요는 계속 증가할 것으로 예상된다

## *발명의 내용*

### *해결하려는 과제*

### *과제의 해결 수단*

[0004] 디지털 비디오 처리, 및 예를 들어 비디오 처리를 위한 비-선형 적응형 루프 필터링(non-linear adaptive loop filtering)과 관련된 장치, 시스템 및 방법이 설명된다. 설명된 방법은 기존의 비디오 코딩 표준(예를 들어, 고효율 비디오 코딩(High Efficiency Video Coding, HEVC))과 미래의 비디오 코딩 표준 (예를 들어, 다목적 비디오 코딩(Versatile Video Coding, VVC)) 또는 코덱 모두에 적용될 수 있다.

하나의 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하기 위해 사용될 수 있다. 이 방법은 비디오의 비디오 유닛(video unit)을 인코딩된 비디오 유닛(encoded video unit)으로 인코딩(encoding)하는 단계; 인코딩된 비디오 유닛으로부터 재구성 샘플(reconstruction sample)들을 생성하는 단계; 재구성 샘플들에 대해 클리핑 연산(clipping operation)을 수행하는 단계 - 클리핑 연산에 사용되는 클리핑 파라미터(clipping

parameter)는 비디오 유닛의 샘플들의 비트-깊이 또는 재구성 샘플들의 비트-깊이 및 재구성 샘플들의 클리핑 인덱스의 함수임 -; 클리핑 연산의 출력에 비-선형 적응형 루프 필터(non-linear adaptive loop filter)를 적용하는 단계; 및 인코딩된 비디오 유닛을 사용하여 비디오의 코딩된 표현(coded representation)을 생성하는 단계를 포함한다.

다른 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하기 위해 사용될 수 있다. 이 방법은 비디오의 비디오 유닛을 나타내는 인코딩된 비디오 유닛에 대해 비디오의 코딩된 표현을 파싱(parsing)하는 단계; 인코딩된 비디오 유닛으로부터 비디오 유닛의 재구성 샘플들을 생성하는 단계; 재구성 샘플들에 대해 클리핑 연산을 수행하는 단계 - 클리핑 연산에 사용되는 클리핑 파라미터는 비디오 유닛의 샘플들의 비트-깊이 또는 재구성 샘플들의 비트-깊이 및 재구성 샘플들의 클리핑 인덱스의 함수임 - ; 및 최종 디코딩된 비디오 유닛(final decoded video unit)을 생성하기 위해 클리핑 연산의 출력에 비-선형 적응형 루프 필터를 적용하는 단계를 포함한다.

다른 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하기 위해 사용될 수 있다. 이 방법은 하나 이상의 비디오 영역(video region)들을 포함하는 비디오의 코딩된 표현과 비디오 사이의 전환(conversion)을 수행하는 단계; 및 비-선형 적응형 루프 필터를 사용하여 비디오 영역의 비디오 유닛의 재구성을 필터링하기 위한 클리핑 파라미터를 결정하는 단계를 포함하고, 결정하는 단계는, 비디오 및/또는 비디오 영역 및/또는 비디오 유닛의 코딩된 정보에 기초한다.

다른 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하기 위해 사용될 수 있다. 이 방법은 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 사이의 전환을 수행하는 단계; 및 비-선형 적응형 루프 필터를 사용하여 비디오 영역의 비디오 유닛의 재구성을 필터링하기 위한 클리핑 파라미터를 결정하는 단계를 포함하고, 클리핑 파라미터는, 색상 표현 형식의 함수이다.

다른 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하기 위해 사용될 수 있다. 이 방법은 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 사이의 전환을 수행하는 단계; 및 비-선형 적응형 루프 필터를 사용하여 비디오 영역의 비디오 유닛의 재구성을 필터링하기 위한 클리핑 파라미터를 결정하는 단계를 포함하고, 클리핑 파라미터는, 제1 도메인 및 제2 도메인에서 비디오 유닛의 표현에 기초하여 비디오 유닛을 재구성하기 위해 및/또는 채도 비디오 유닛(chroma video unit)의 채도 잔차(chroma residue)를 스케일링하기 위해 인-루프 변형(in-loop reshaping, ILR)이 적용되는지 여부에 의존(depend)한다.

다른 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하기 위해 사용될 수 있다. 이 방법은 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 사이의 전환을 수행하는 단계를 포함하고, 코딩된 표현은, 비-선형 적응형 루프 필터를 사용하여 비디오 영역의 비디오 유닛의 재구성을 필터링하기 위한 클리핑 파라미터를 제공하는 제1 부가 정보(side information)를 포함하고, 제1 부가 정보는, 비-선형 적응형 루프 필터에서 사용되는 필터 계수(filter coefficient)들을 나타내는 제2 부가 정보와 함께 시그널링된다.

다른 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하기 위해 사용될 수 있다. 이 방법은 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 사이의 전환을 수행하는 단계를 포함하고, 코딩된 표현은, 비-선형 적응형 루프 필터를 사용하여 비디오 영역의 비디오 유닛의 재구성을 필터링하기 위한 클리핑 파라미터의 다중 세트들(multiple sets)을 나타내는 부가 정보를 포함한다.

다른 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하기 위해 사용될 수 있다. 이 방법은 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 사이의 전환을 수행하는 단계를 포함하고, 코딩된 표현은, 비-선형 적응형 루프 필터를 사용하여 비디오 영역의 채도 비디오 유닛의 재구성을 필터링하기 위한 하나 이상의 클리핑 파라미터들을 제공하는 부가 정보를 포함하고, 하나 이상의 클리핑 파라미터들은, 색상 형식(color format)에 의존한다.

다른 대표적인 양태에서, 개시된 기술은 비디오 처리를 위한 방법을 제공하기 위해 사용될 수 있다. 이 방법은 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 사이의 전환을 수행하는 단계를 포함하고, 코딩된 표현은, 적응형 루프 필터를 사용하여 비디오 영역의 비디오 유닛의 재구성을 필터링하기 위한 클리핑 파라미터를 제공하는 부가 정보를 포함하고, 수행하는 단계는, 비디오 영역 레벨에서 샘플 차이들에 클리핑 연산을 적용하는 것에 의해 필터링된 비디오 유닛을 생성하는 단계를 포함한다.

또 다른 대표적인 양태에서, 상술된 방법은 프로세서-실행가능 코드(processor-executable code)의 형태로 구현

되고 컴퓨터 판독가능 프로그램 매체(computer-readable program medium)에 저장된다.

또 다른 대표적인 양태에서, 상술된 방법을 수행하도록 구성되거나 동작가능한 디바이스가 개시된다. 장치는 이 방법을 구현하도록 프로그래밍된 프로세서를 포함할 수 있다.

또 다른 대표적인 양태에서, 상술된 방법 중 하나 이상은 인코더(encoder) 측 구현 또는 디코더(decoder) 측 구현일 수 있다.

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

[0005] 삭제

[0006] 삭제

[0007] 삭제

[0008] 삭제

[0009] 삭제

[0010] 삭제

[0011] 삭제

[0012] 삭제

[0013] 삭제

[0014] 삭제

[0015] 삭제

[0016] 삭제

[0017] 삭제

## 도면의 간단한 설명

도 1은 비디오 코딩을 위한 인코더 블록도의 예를 도시한다.

도 2a, 2b 및 2c는 기하학적 변환-기반 적응형 루프 필터(GALF) 필터 형상들의 예들을 도시한다.

도 3은 GALF 인코더 결정을 위한 흐름 그래프의 예를 도시한다.

도 4a 내지 도 4d는 적응형 루프 필터(ALF) 분류를 위한 예시적인 서브샘플링된 라플라시안 계산들을 도시한다.

도 5는 휘도 필터 형상의 예를 도시한다.

도 6은 와이드 비디오 그래픽 배열(Wide Video Graphic Array, WVGA) 시퀀스의 영역 분할의 예를 도시한다.

도 7은 변형에 의한 디코딩 흐름의 예시적인 흐름도를 도시한다.

도 8a 내지 도 8c는 개시된 기술의 일부 구현에 따른 비디오 처리를 위한 예시적인 방법의 흐름도를 도시한다.

도 9는 개시된 기술의 일부 구현들에 따른 비디오 처리를 위한 예시적인 방법의 흐름도를 도시한다.

도 10a 및 10b는 본 문서에 설명된 비디오 처리 기술을 구현하기 위한 하드웨어 플랫폼의 예의 블록도들이다.

## *발명을 실시하기 위한 구체적인 내용*

[0019] 삭제

[0020] 삭제

[0021] 삭제

[0022] 삭제

[0023] 삭제

[0024] 삭제

[0025] 삭제

[0026] 삭제

[0027] 삭제

[0028] 삭제

[0029] 삭제

[0030] 삭제

[0031] 삭제

[0032] 삭제

[0033] 삭제

[0034] 삭제

[0035] 삭제

[0036] 삭제

[0037] 삭제

[0038] 삭제

[0039] 삭제

[0040] 삭제

[0041] 삭제

[0042] 삭제

[0043] 삭제

[0044] 삭제

[0045] 삭제

[0046] 삭제

[0047] 삭제

[0048] 삭제

[0049] 삭제

[0050] 고해상도 비디오에 대한 수요가 증가함에 따라 비디오 코딩 방법 및 기술은 현대 기술에 유비쿼터스(ubiquitous)이다. 비디오 코덱(Video codec)들은 일반적으로 디지털 비디오를 압축하거나 압축 해제하는 전자 회로 또는 소프트웨어를 포함하고 더 높은 코딩 효율성을 제공하기 위해 지속적으로 개선되고 있다. 비디오 코덱은 압축되지 않은 비디오를 압축된 형식(compressed format)으로 또는 그 반대로 전환한다. 비디오 품질(video quality), 비디오를 나타내기 위해 이용되는 데이터의 양(비트 전송률에 의해 결정됨), 인코딩 및 디코딩 알고리즘의 복잡성, 데이터 손실 및 오류에 대한 민감도(sensitivity to data losses and errors), 편집 용이성(ease of editing), 랜덤 액세스(random access) 및 종단 간 지연(end-to-end delay)(지연시간(latency)) 사이에는 복잡한 관계가 있다. 압축된 형식은 일반적으로 표준 비디오 압축 사양, 예를 들어 HEVC 표준(H.265 또는 MPEG-H Part 2라고도 함), 완성될 VVC 표준, 또는 다른 현재 및/또는 미래의 비디오 코딩 표준을 따른다.

일부 실시예들에서, 미래의 비디오 코딩 기술은 공동 탐사 모델(Joint Exploration Model, JEM)으로 알려진 참조 소프트웨어를 사용하여 탐색된다. JEM에서 서브-블록 기반 예측은 아핀 예측(affine prediction), 대체 시간적 모션 벡터 예측(alternative temporal motion vector prediction, ATMVP), 공간적-시간적 모션 벡터 예측(spatial-temporal motion vector, STMVP), 양방향 광학 흐름(bi-directional optical flow, BIO), 프레임 레이트 상향 전환(Frame-Rate Up Conversion, FRUC), 로컬 적응형 모션 벡터 해상도(Locally Adaptive Motion Vector Resolution, LAMVR), 중첩된 블록 모션 보상(Overlapped Block Motion Compensation, OBMC), 로컬 조명 보상(Local Illumination Compensation, LIC) 및 디코더 측 모션 벡터 개선(Decoder-side Motion Vector Refinement, DMVR)과 같은 여러 코딩 도구에서 채택된다.

개시된 기술의 실시예들은 런타임 성능(runtime performance)을 향상시키기 위해 기존의 비디오 코딩 표준(예를 들어, HEVC, H.265) 및 미래의 표준에 적용될 수 있다. 섹션 표제들은(Section headings) 설명의 가독성을 향상시키기 위해 본 문서에서 사용되고 논의 또는 실시예들(및/또는 구현들)을 개별 섹션으로만 제한하지 않는다.

## 1 색상 공간 및 채도 서브샘플링의 예들

색상 모델(color model)(또는 색상 시스템(color system))이라고도 하는 색상 공간(Color space)은 색상 범위를 숫자들의 튜플들(tuples of numbers), 일반적으로 3개 또는 4개 값 또는 색상 구성요소들(예: RGB)로 간단히 설명하는 추상적 수학 모델(abstract mathematical model)이다. 기본적으로 색상 공간은 좌표계(coordinate system)와 서브-공간의 정교화(elaboration)이다.

비디오 압축을 위해 가장 많이 사용되는 색상 공간은 YCbCr과 RGB이다.

YCBCR 또는 Y'CBCR라고 쓰이기도 하는 YCbCr, Y'CbCr 또는 Y Pb/Cb Pr/Cr은 비디오 및 디지털 사진 시스템들에서 색상 이미지 파이프라인의 일부로 사용되는 색상 공간들의 제품군(family of color spaces)이다. Y'는 휘도 구성요소(luma component)이고 CB 및 CR은 청색차(blue-difference) 및 적색차(red-difference) 채도 구성요소(chroma component)들이다. Y'(프라임 포함)는 휘도인 Y와 구별된다. 즉, 감마 보정된(gamma corrected) RGB 프라이머리(RGB primary)들에 기초하여 빛의 강도(light intensity)가 비선형적으로 인코딩된다.

채도 서브샘플링은 휘도보다 색상 차이에 대한 인간 시각 시스템의 낮은 선명도(acuity)를 이용하여 휘도 정보보다 채도 정보에 대해 더 낮은 해상도를 구현함으로써 이미지들을 인코딩하는 실행이다.

### 1.1 4:4:4 색상 형식

3개의 Y'CbCr 구성요소들 각각은 동일한 샘플 레이트(sample rate)를 가지므로 채도 서브샘플링이 없다. 이 방식은 고급 필름 스캐너들(high-end film scanners)과 영화 후반 제작(cinematic post production)에 종종 사용된다.

### 1.2 4:2:2 색상 형식

2개의 채도 구성요소들은 휘도의 샘플 레이트의 절반으로 샘플링되고, 예를 들어 수평 채도 해상도는 절반이 된다. 이것은 시각적인 차이가 거의 또는 전혀 없이 압축되지 않은 비디오 신호의 대역폭을 1/3로 감소시킨다.

### 1.3 4:2:0 색상 형식

4:2:0에서 수평 샘플링은 4:1:1에 비해 2배이지만 Cb 및 Cr 채널은 이 방식에서 각 대체 라인에서만 샘플링되므로 수직 해상도는 절반이다. 따라서 데이터 레이트는 동일하다. Cb와 Cr은 수평 및 수직으로 각각 2의 비율로 서브샘플링된다. 가로 및 세로 위치가 다른 4:2:0 방식 세 가지 변형들이 있다.

- MPEG-2에서 Cb와 Cr은 수평으로 나란히 배치된다. Cb 및 Cr은 (틈새에 배치되어(sited interstitially)) 수

직 방향으로 픽셀들 사이에 배치된다.

- JPEG/JFIF, H.261, 및 MPEG-1에서, Cb 및 Cr은 대체 휘도 샘플들 사이의 중간인 틈새에 배치된다.

- 4:2:0 DV에서는 Cb와 Cr이 수평방향으로 함께 배치(so sited)된다. 수직 방향으로, 그들은 대체 라인(alternating line)들에 함께 배치된다.

**2 일반적인 비디오 코덱의 코딩 흐름의 예들**

도 1은 VVC의 인코더 블록도의 예를 도시한다. 도 1에 도시된 바와 같이, 비디오 인코딩의 흐름은 인트라 예측 및/또는 모션 추정(motion estimation)/모션 보상(ME/MC)이 된 입력 비디오로 시작할 수 있다. 이러한 연산들은 비디오의 이전에 코딩된 부분의 재구성된 사본으로부터의 피드백을 사용한다. 인트라 예측 및/또는 ME/MC의 출력은 변환 연산(T)과 양자화 연산(quantization operation, Q)을 통해 차등적으로 처리되며, 이는 출력 코딩된 표현으로 엔트로피(entropy) 코딩된다. 피드백 루프에서, 인코딩된 표현(Q 블록의 출력)은 역양자화 연산(inverse quantization operation, IQ)을 거친 다음, 인코딩된 비디오 블록들의 재구성 샘플들을 생성하기 위한 역변환(inverse transform, IT) 연산을 거칠 수 있다.

재구성된 샘플들은 참조 샘플, 또는 참조 블록, 또는 추가 인코딩에 사용되는 참조 화상(reference picture)을 생성하기 위해 다양한 "인-루프" 필터링 연산들을 통해 추가로 처리될 수 있다. 인-루프 필터링 프로세스 체인에는 3개의 인-루프 필터링 블록인 디블로킹 필터(deblocking filter, DF), 샘플 적응형 오프셋(sample adaptive offset, SAO) 및 ALF가 포함된다. 미리 정의된 필터들을 사용하는 DF와 달리, SAO 및 ALF는 오프셋을 추가하고, 오프셋들 및 필터 계수들을 시그널링하는 코딩된 부가 정보와 함께 유한 임펄스 응답(finite impulse response, FIR) 필터를 각각 적용하여, 원본 샘플들과 재구성된 샘플들 간의 평균 제곱 오차들을 줄이기 위해 현재 화상의 원본 샘플들을 활용한다. ALF는 각 화상의 마지막 처리 단계에 위치하고 이전 단계들에서 생성된 인공 결함(artifact)들을 잡아서 수정하는 도구라고 볼 수 있다.

디코더 측에서, 디코딩 및 재구성된 비디오 샘플들을 생성하기 위해 몇몇 인코딩 연산들이 역순으로 수행된다. 예를 들어, 도 1을 참조하면, 비디오 유닛의 재구성 샘플을 생성하기 위해, 디코더는 엔트로피 코딩의 출력인 코딩된 표현을 파싱할 수 있고, 역양자화(IQ) 연산 및 역변환(IT) 연산을 통해 전달되는 비디오의 인코딩 유닛들 또는 블록들을 획득할 수 있다. 최종 디코딩된 비디오 유닛은 비디오 인코더의 피드백 루프에 대해 위에서 설명된 바와 같은 인-루프 필터링 연산을 적용함으로써 생성될 수도 있다.

**3 JEM의 기하학적 변환-기반 적응형 루프 필터의 예들**

JEM에서는 블록-기반 필터 적응을 가지는 기하학적 변환-기반 적응형 루프 필터(GALF)가 적용된다. 휘도 구성요소의 경우 로컬 기울기(local gradient)들의 방향과 활성도(activity)에 기초하여 2x2 블록마다 25개의 필터들 중 하나가 선택된다.

**3.1 필터 형상의 예들**

JEM에서, 최대 3개의 다이아몬드 필터 형상들(각각 도 2a, 2b 및 2c에 도시된 5x5 다이아몬드, 7x7 다이아몬드 및 9x9 다이아몬드에 대해)이 휘도 구성요소에 대해 선택될 수 있다. 휘도 구성요소에 사용되는 필터 형상을 나타내기 위해 화상 레벨에서 인덱스가 시그널링된다. 화상의 채도 구성요소들에는 항상 5X5 다이아몬드 형상이 사용된다.

**3.1.1 블록 분류**

각 2X2 블록은 25개의 클래스들 중 하나로 분류된다. 분류 인덱스 C는 방향성 D와 활성도 $\hat{A}$의 양자화된 값에 기초하여 다음과 같이 도출된다.

*수학식 1*

$$C = 5D + \hat{A}$$

D 및 $\hat{A}$를 계산하기 위해, 먼저 1차원 라플라시안(1-D Laplacian)을 사용하여 수평, 수직 및 두 대각선 방향

(diagonal direction)의 기울기들을 계산한다.

*수학식 2*

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

*수학식 3*

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

*수학식 4*

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

*수학식 5*

$$g_{d2} = \sum_{k=i-2}^{i+3} \sum_{j=j-2}^{j+3} D2_{k,l}, \cdot D2_{k,l}$$
$$= |2R(k,l) - R(k-1,l+1) - R(k+1,l-1)|$$

인덱스 i와 j는 2X2 블록에서 좌상단 샘플의 좌표들을 나타내고, R(i,j)는 좌표 (i,j)에서 재구성된 샘플을 나타낸다.

그런 다음 수평 및 수직 방향들의 기울기들의 D 최댓값과 최솟값은 다음과 같이 설정된다.

*수학식 6*

$$g_{h,v}^{max} = max(g_h, g_v), \cdots g_{h,v}^{min} = min(g_h, g_v).$$

두 대각선 방향들의 기울기의 최댓값과 최솟값은 다음과 같이 설정된다.

*수학식 7*

$$g_{d0,d1}^{max} = max(g_{d0}, g_{d1}), \cdots g_{d0,d1}^{min} = min(g_{d0}, g_{d1}).$$

방향성 D의 값을 도출하기 위해 이러한 값들을 서로 비교하고 두 개의 임계값들 $t_1$ 및 $t_2$를 사용한다.

**단계 1.** $g_{h,v}^{max} \leq t_1 \cdot g_{h,v}^{min}$ 및 $g_{d0,d1}^{max} \leq t_1 \cdot g_{d0,d1}^{min}$ 이 모두 참인 경우, D는 0으로 설정된다.

**단계 2.** $g_{h,v}^{max} / g_{h,v}^{min} > g_{d0,d1}^{max} / g_{d0,d1}^{min}$ 인 경우, 단계 3에서부터 계속하고, 그렇지 않으면 단계 4부터 계속한다.

**단계 3.** $g_{h,v}^{max} > t_2 \cdot g_{h,v}^{min}$ 이 참인 경우, D는 2로 설정되고, 그렇지 않으면, D는 1로 설정된다.

**단계 4.** $g_{d0,d1}^{max} > t_2 \cdot g_{d0,d1}^{min}$ 이 참인 경우, D는 4로 설정되고, 그렇지 않으면 D는 3으로 설정된다.

활성도 값 A는 다음과 같이 계산된다.

*수학식 8*

$$A = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} \left(V_{k,l} + H_{k,l}\right).$$

A는 0 내지 4까지의 범위로 더 양자화되고, 양자화된 값은 $\hat{A}$로 표시된다.

화상의 두 채도 구성요소들에 대해 분류 방법이 적용되지 않는다. 즉, 각 채도 구성요소에 대해 ALF 계수들의 단일 세트가 적용된다.

### 3.1.2 필터 계수들의 기하학적 변환들

각 2X2 블록을 필터링하기 전에, 회전 또는 대각선 및 수직 플리핑(diagonal and vertical flipping)와 같은 기하학적 변환들은 그 블록에 대해 계산된 기울기 값들에 의존하여 필터 계수들 f(k,l)에 적용된다. 이는 필터 지원 영역(filter support region)의 샘플에 이러한 변환들을 적용하는 것과 같다. 아이디어는 ALF가 적용되는 다른 블록들을, 방향성을 정렬하여 더 유사하게 만드는 것이다.

대각선, 수직 플리핑 및 회전을 포함하는 세 가지 기하학적 변환들이 도입된다.

*수학식 9*

대각선: $f_D(k,l) = f(l,k)$

수직 플리핑: $f_V(k,l) = f(k,K-l-1)$

회전: $f_R(k,l) = f(K-l-1,k)$

여기서, K는 필터의 크기이고, $0 \leq k, l \leq K-1$ 은 계수 좌표들로서 위치 (0,0)은 좌상단 모서리에 있고 위치 (K-1,K-1)은 우하단 모서리에 있도록 한다. 변환들은 그 블록에 대해 계산된 기울기 값들에 의존하여 필터 계수들 f(k, l)에 적용된다. 변환과 네 방향들의 네 기울기들 사이의 관계는 표 1에 요약되어 있다.

*표 1*

| 기울기 값들 | 변환 |
|---|---|
| $g_{d2} < g_{d1}$ 및 $g_h < g_v$ | 변환 없음 |
| $g_{d2} < g_{d1}$ 및 $g_v < g_h$ | 대각선 |
| $g_{d1} < g_{d2}$ 및 $g_h < g_v$ | 수직 플리핑 |
| $g_{d1} < g_{d2}$ 및 $g_v < g_h$ | 회전 |

표 1: 한 블록 및 변환들에 대해 계산된 기울기 매핑(mapping)

### 3.1.3 필터 파라미터들의 시그널링

JEM에서는, GALF 필터 파라미터는 첫 번째 CTU에 대해, 즉 슬라이스 헤더(slice header) 뒤 및 첫 번째 CTU의 SAO 파라미터 이전에 시그널링된다. 최대 25개 세트들의 휘도 필터 계수들이 시그널링될 수 있다. 비트 오버헤드(bits overhead)를 감소시키기 위해 다른 분류의 필터 계수들이 병합될 수 있다. 또한, 참조 화상의 GALF 계수가 저장되고 현재 화상의 GALF 계수로 재사용될 수 있다. 현재 화상은 참조 화상에 대해 저장된 GALF 계수를 사용하고 GALF 계수들 시그널링을 우회하도록 선택할 수 있다. 이 경우, 참조 화상들 중 하나에 대한 인덱스만 시그널링되고, 나타난(indicated) 참조 화상의 저장된 GALF 계수들은 현재 화상에 대해 상속된다.

GALF 시간적 예측을 지원하기 위해 GALF 필터 세트들의 후보 리스트가 유지된다. 새로운 시퀀스를 디코딩하기 시작할 때 후보 리스트는 비어 있다. 하나의 화상을 디코딩한 후, 대응하는 필터들의 세트가 후보 리스트에 추가될 수 있다. 후보 리스트의 크기가 최대 허용 값(즉, 현재 JEM에서 6)에 도달하면 새로운 필터들의 세트가 디코딩 순서에서 가장 오래된 세트를 덮어쓴다(overwrite). 즉, 선입선출(first-in-first-out, FIFO) 규칙을 적용하여 후보 리스트를 업데이트한다. 중복들을 방지하기 위해 대응하는 화상이 GALF 시간적 예측을 사용하지 않는 경우에만 세트가 리스트에 추가될 수 있다. 시간적 확장성(temporal scalability)을 지원하기 위해 필터 세트들의 다중 후보 리스트가 있고 각 후보 리스트는 시간적 레이어와 연결된다. 더 구체적으로, 시간적 레이어 인덱스(temporal layer index, TempIdx)에 의해 할당된 각 배열은 더 낮은 TempIdx와 동일한 이전에 디코딩된 화상의 필터 세트들을 구성할 수 있다. 예를 들어 k번째 배열은 TempIdx가 k와 연관되도록 할당되고 TempIdx가 k보다 작거나 같은 화상들로부터의 필터 세트들만 포함한다. 특정 화상을 코딩한 후 화상과 연관된 필터 세트들은 같거나 더 높은 TempIdx와 연관된 배열을 업데이트하기 위해 이용된다.

GALF 계수의 시간적 예측은 시그널링 오버헤드를 최소화하기 위해 상호 코딩된 프레임에 사용된다. 인트라 프레임의 경우 시간적 예측을 사용할 수 없고 각 클래스에 16개의 고정된 필터들의 세트가 할당된다. 고정된 필터의 사용을 나타내기 위해 각 클래스에 대한 플래그가 표시되고 필요한 경우 선택된 고정된 필터의 인덱스가 표시된다. 주어진 클래스에 대해 고정된 필터가 선택된 경우에도, 적응형 필터 f(k,l)의 계수들은 이 클래스에 대해 여전히 전송될 수 있고, 이 경우 재구성된 이미지에 적용될 필터의 계수들은 두 계수들의 세트들(both sets of coefficients)의 합이다.

휘도 구성요소의 필터링 프로세스는 CU 레벨에서 제어될 수 있다. GALF가 CU의 휘도 구성요소에 적용되는지 여부를 나타내기 위해 플래그(flag)가 시그널링된다. 채도 구성요소의 경우 GALF 적용 여부는 화상 레벨에서만 나타난다.

### 3.1.4 필터링 프로세스

디코더 측에서 블록에 대해 GALF가 활성화되면 블록 내의 각 샘플 R(i,j)가 필터링되어 아래와 같이 샘플 값 R'(i,j)가 생성된다. 여기서 L은 필터 길이, $f_{m,n}$은 필터 계수, f(k,l)은 코딩된 필터 계수를 나타낸다.

*수학식 10*

$$R'(i,j) = \sum_{k=-L/2}^{L/2} \sum_{l=-L/2}^{L/2} f(k,l) \times R(i+k, j+l)$$

### 3.1.5 인코더 측 필터 파라미터들에 대한 결정 프로세스

GALF에 대한 전체 인코더 결정 프로세스가 도 3에 도시되어 있다. 각 CU의 휘도 샘플들에 대해 인코더는 GALF가 적용되고 적절한 시그널링 플래그가 슬라이스 헤더에 포함되는지 여부를 결정한다. 채도 샘플의 경우 필터 적용 결정은 CU 레벨이 아닌 화상 레벨을 기반으로 이루어진다. 또한 화상에 대한 채도 GALF는 화상에 대해 휘도 GALF가 활성화되는 경우에만 확인된다.

## 4 VVC에서 기하학적 변환-기반 적응형 루프 필터의 예들

VVC의 GALF의 현재 설계는 JEM의 설계와 비교하여 다음과 같은 주요 변경 사항들이 있다.

1) 적응형 필터 형상이 제거된다. 휘도 구성요소에는 7x7 필터 형상만 허용되고 채도 구성요소에는 5x5 필터 형상이 허용된다.

2) ALF 파라미터들의 시간적 예측과 고정된 필터들의 예측이 모두 제거된다.

3) 각 CTU에 대해 ALF가 활성화 또는 비활성화되었는지 여부에 따라 하나의 비트 플래그가 시그널링된다.

4) 클래스 인덱스 계산은 2x2가 아닌 4x4 레벨로 수행된다. 또한 JVET-L0147에서 제안된 바와 같이 ALF 분류를 위한 서브샘플링된 라플라시안 계산 방법이 활용된다. 보다 구체적으로, 하나의 블록 내에서 각 샘플에 대해 수평/수직/45 대각선/135도 기울기들을 계산할 필요가 없다. 대신 1:2 서브샘플링이 활용된다.

### 5 AVS2의 영역-기반 적응형 루프 필터의 예들

ALF는 인-루프 필터링의 마지막 단계이다. 이 프로세스에는 두 단계들이 있다. 첫 번째 단계는 필터 계수 도출이다. 필터 계수들을 훈련시키기 위해, 인코더는 휘도 구성요소의 재구성된 픽셀들을 16개 영역으로 분류하고, 필터 계수들의 한 세트는 원본 프레임과 재구성된 프레임 사이의 평균 제곱 오차를 최소화하기 위해 위너-홉프 방정식(wiener-hopf equation)들을 사용하여 각 범주에 대해 훈련된다. 이러한 16개의 필터 계수들의 세트들 사이의 중복(redundancy)을 감소시키기 위해 인코더는 레이트 왜곡 성능(rate-distortion performance)에 기초하여 이들을 적응적으로 병합한다. 최대 16개의 서로 다른 필터 세트들을 휘도 구성요소에 할당할 수 있고 색차(chrominance) 구성요소들에는 하나만 할당할 수 있다. 두 번째 단계는 프레임 레벨과 LCU 레벨을 모두 포함하는 필터 결정이다. 먼저 인코더는 프레임 레벨 적응형 루프 필터링을 수행할지 여부를 결정한다. 프레임 레벨 ALF가 켜져 있으면 인코더는 LCU 레벨 ALF를 수행할지 여부를 추가로 결정한다.

### 5.1 필터 형상

AVS-2에 선택된 필터 형상은 휘도 및 채도 구성요소들 모두에 대해 도 5에 도시된 바와 같이 3x3 정사각형 형상이 중첩된 7x7 십자 형상이다. 도 5의 각 사각형은 샘플에 대응한다. 따라서 C8 위치의 샘플에 대한 필터링된 값을 도출하기 위해 총 17개의 샘플들이 사용된다. 계수들을 전송하는 것의 오버헤드를 고려하여, 점-대칭 필터(point-symmetrical filter)는 {C0, C1, . . . , C8}인 9개의 계수들만 남은 상태로 사용되고, 이는 필터 계수들의 수와 필터링 곱셈들의 수를 반으로 감소시킨다. 점-대칭 필터는 또한 하나의 필터링된 샘플에 대한 계산의 절반을 감소시킬 수 있다. 예를 들어, 하나의 필터링된 샘플에 대해 9개의 곱셈들 및 14개의 덧셈 연산들만 수행된다.

### 5.2 영역-기반 적응형 병합

다른 코딩 오차들을 적용하기 위해 AVS-2는 휘도 구성요소에 대해 영역 기반 다중 적응형 루프 필터들을 채택한다. 휘도 구성요소는 대략 동일한 크기(roughly-equal-size)의 기본 영역들 16개로 분할된다. 여기서 각각의 기본 영역은 도 6에 도시된 바와 같이 최대 코딩 유닛(largest coding unit, LCU) 경계들 정렬되고, 각 영역에 대해 하나의 위너 필터가 도출된다. 더 많은 필터들을 사용할수록 더 많은 왜곡들이 감소되지만 이러한 계수들을 인코딩하기 위해 이용되는 비트들은 필터들의 수와 함께 증가한다. 최고의 레이트-왜곡 성능을 달성하기 위해 이러한 영역들은 동일한 필터 계수들을 공유하는 더 적은 수의 더 큰 영역들로 병합될 수 있다. 병합 프로세스를 단순화하기 위해 각 영역에는 이미지 사전 상관관계(image prior correlation)들에 기초하여 수정된 힐베르트 순서(Hilbert order)에 따라 인덱스가 할당된다. 연속적인 인덱스들이 있는 두 영역들은 레이트-왜곡 비용에 기초하여 병합될 수 있다.

영역들 간의 매핑 정보는 디코더에 시그널링되어야 한다. AVS-2에서 기본 영역들의 수는 병합 결과를 나타내기 위해 이용되고 필터 계수들은 영역 순서(region order)에 따라 순차적으로 압축된다. 예를 들어 {0, 1}, {2, 3, 4}, {5, 6, 7, 8, 9} 및 남은 기본 영역들이 각각 하나의 영역으로 병합되는 경우, 이 병합 맵을 나타내기 위해 3개의 정수들(예: 2, 3, 5)만 코딩된다.

### 5.3 부가 정보의 시그널링

다중 스위치 플래그(Multiple switch flag)들도 사용된다. 시퀀스 스위치 플래그인 adaptive_loop_filter_enable은 적응형 루프 필터가 전체 시퀀스에 적용되는지 여부를 제어하기 위해 이용된다. 이미지 스위치 플래그들인 picture_alf_enble[i]은 대응하는 i번째 이미지 구성요소에 ALF가 적용되는지 여부를 제어한다. picture_alf_enble[i]이 활성화되는 경우에만 그 색상 구성요소에 대한 대응하는 LCU 레벨 플래그들

및 필터 계수들이 전송된다. LCU 레벨 플래그들인 lcu_alf_enable[k]는 대응하는 k번째 LCU에 대해 ALF가 활성화되는지 여부를 제어하고 슬라이스 데이터에 인터리빙된다. 다른 레벨 규제된 플래그(different level regulated flag)들의 결정은 모두 레이트-왜곡 비용에 기초한다. 높은 유연성은 ALF가 코딩 효율성을 훨씬 더 크게 향상시키도록 한다.

일부 실시예들에서, 휘도 구성요소에 대해 최대 16개의 필터 계수들의 세트들이 있을 수 있다.

일부 실시예들에서, 그리고 각각의 채도 구성요소(Cb 및 Cr)에 대해, 필터 계수들의 한 세트가 전송될 수 있다.

### 6 VTM-4의 GALF

VTM4.0에서 적응형 루프 필터의 필터링 프로세스는 다음과 같이 수행된다.

*수학식 11*

$$O(x,y) = \sum_{(i,j)} w(i,j).I(x+i,y+j)$$

여기서 샘플들 I(x+i,y+j)는 입력 샘플들이고, O(x,y)는 필터링된 출력 샘플(즉, 필터 결과)이고, w(i,j)는 필터 계수들을 나타낸다. 실제로 VTM4.0에서는 고정점(fixed point)의 정밀도 계산들을 위해 정수 산술(integer arithmetic)을 사용하여 구현된다.

*수학식 12*

$$O(x,y) = \left( \sum_{i=-\frac{L}{2}}^{\frac{L}{2}} \sum_{j=-\frac{L}{2}}^{\frac{L}{2}} w(i,j).I(x+i,y+j) + 64 \right) \gg 7$$

여기서 L은 필터 길이를 나타내고 w(i,j)는 고정점 정밀도의 필터 계수들이다.

### 7 비-선형 적응형 루프 필터링(ALF)

#### 7.1 필터링 재공식화(filtering reformulation)

수학식 11은 코딩 효율성 영향 없이 다음 식으로 재공식화될 수 있다.

*수학식 13*

$$O(x,y) = I(x,y) + \sum_{(i,j)\neq(0,0)} w(i,j).\big(I(x+i,y+j) - I(x,y)\big)$$

여기서, w(i,j)는 수학식 11에서와 같은 필터 계수들이다[수학식 13에서 1과 동일한 반면 수학식 11에서 $1 - \Sigma_{(i,j)\neq(0,0)} w(i,j)$ 와 동일한 w(0,0)을 제외하고].

#### 7.2 수정된 필터

위의 (13)의 필터 공식을 사용하여, 필터링되는 현재 샘플 값(I(x,y))과 너무 다른 이웃한 샘플 값들(I(x+i,y+j))의 영향을 줄이기 위해 간단한 클리핑 기능을 사용하여 ALF를 보다 효율적으로 만들기 위해 비 선형성을 쉽게 도입할 수 있다.

이 제안에서 ALF 필터는 다음과 같이 수정된다:

*수학식 14*

$$O'(x,y) = I(x,y) + \sum_{(i,j)\neq(0,0)} w(i,j).K(I(x+i,y+j) - I(x,y), k(i,j))$$

여기서, $K(d,b) = \min(b, \max(-b,d))$는 클리핑 함수이고, k(i,j)는 (i,j) 필터 계수에 따라 달라지는 클리핑 파라미터들이다. 인코더는 최적의 k(i,j)를 찾기 위해 최적화를 수행한다.

JVET-N0242 구현에서, 클리핑 파라미터들 k(i,j)는 각 ALF 필터에 대해 명시되고 필터 계수당 하나의 클리핑 값이 시그널링된다. 이는 휘도 필터당 최대 12개의 클리핑 값들이 비트스트림에서 시그널링될 수 있고 채도 필터에 대해 최대 6개의 클리핑 값들이 시그널링될 수 있음을 의미한다.

시그널링 비용 및 인코더 복잡도를 제한하기 위해, 클리핑 값들의 평가를 가능한 값들의 작은 세트로 제한한다. 제안에서 INTER 및 INTRA 타일 그룹들에 대해 동일한 4개의 고정된 값들만 사용한다.

로컬 차이들의 분산(variance)이 채도보다 휘도에 대해 더 높기 때문에, 우리는 휘도 및 채도 필터에 대해 두 개의 다른 세트들을 사용한다. 또한 필요하지 않은 경우 클리핑을 비활성화할 수 있도록 각 세트에 최대 샘플 값(여기서는 10비트 비트-깊이의 경우 1024)을 포함한다.

JVET-N0242 테스트들에 사용된 클리핑 값들의 세트들은 표 2에 제공된다. 4개의 값들은 대수 도메인(logarithmic domain)에서 휘도의 경우 샘플 값들의 전체 범위(10비트로 코딩됨)와 채도의 경우 4에서 1024까지의 범위를 대략 균등하게 분할하여 선택된다.

보다 정확하게는, 클리핑 값들의 휘도 테이블은 다음 공식에 의해 획득되었다:

*수학식 15*

$$\text{AlfClip}_L = \left\{ \text{round}\left( \left( (M)^{\frac{1}{N}} \right)^{N-n+1} \right) \text{ for } n \in 1..N \right\}, \text{ with } M=2^{10} \text{ and } N=4$$

유사하게, 클리핑 값들의 채도 테이블들은 다음 공식에 따라 획득된다:

*수학식 16*

$$\text{AlfClip}_C = \left\{ \text{round}\left( A.\left( \left(\frac{M}{A}\right)^{\frac{1}{N-1}} \right)^{N-n} \right) \text{ for } n \in 1..N \right\}, \text{ with } M=2^{10}, N=4 \text{ and } A=4$$

*표 2*

| | INTRA/INTER 타일 그룹 |
|---|---|
| LUMA | { 1024, 181, 32, 6 } |
| CHROMA | { 1024, 161, 25, 4 } |

표 2: 승인된 클리핑 값들선택된 클리핑 값들은 위의 표 2의 클리핑 값의 인덱스에 대응하는 골롬 인코딩(Golomb encoding) 방식을 사용하여 "alf_data" 신택스(syntax) 요소에 코딩된다. 이 인코딩 방식은 필터 인덱스에 대한 인코딩 방식과 동일하다.

**8 JVET-M0427의 인-루프 변형(ILR)**

인-루프 변형(ILR)은 채도 스케일링(Chroma Scaling, LMCS)을 사용한 휘도 매핑으로도 알려져 있다.

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

으로 전환하는 것이다.

인-루프 휘도 변형 변수(in-loop luma reshaper)는 한 쌍의 룩-업 테이블(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 값들인 $Y_i$를, 변경된 값들인 $Y_r$: $Y_r = FwdLUT[Y_i]$에 매핑하는 순방향 LUT인 FwdLUT이다. 다른 LUT는 변경된 코드 값들인 $Y_r$을 $\hat{Y}_i$ : $\hat{Y}_i = InvLUT[Y_r]$에 매핑하는 역 LUT인 InvLUT이다. ($\hat{Y}_i$는 $Y_i$의 재구성 값들을 나타낸다.)

### 8.1 PWL 모델

개념적으로, 조각-별 선형(piece-wise linear, PWL)은 다음과 같은 방식으로 구현된다:

x1, x2를 2개의 입력 피봇 지점(pivot point)들이라고 하고, y1, y2를 한 조각에 대한 대응하는 출력 피봇 지점들이라고 하자. x1과 x2 사이의 입력 값 x에 대한 출력 값 y는 다음 방정식으로 보간될 수 있다.

*수학식 17*

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

고정점 구현에서 방정식은 다음과 같이 다시 써질 수 있다:

*수학식 18*

$$y = ((m * x + 2FP\_PREC-1) >> FP\_PREC) + c$$

여기서, m은 스칼라(scalar), c는 오프셋, FP_PREC는 정밀도를 명시하기 위한 상수 값이다.

CE-12 소프트웨어에서 PWL 모델은 1024-항목 FwdLUT 및 InvLUT 매핑 테이블들을 미리 계산하기 위해 사용된다; 그러나 PWL 모델은 구현들이 LUT들을 미리 계산하지 않고도 즉석에서(on-the-fly) 동일한 매핑 값들을 계산할 수 있도록 한다.

### 8.2 제4 VVC 미팅에서 CE12-2 테스트(Test CE12-2 in the 4th VVC meeting)

#### 8.2.1 휘도 변형

인-루프 휘도 변형(즉, 제안의 CE12-2)의 테스트 2는 인터 슬라이스 재구성(inter slice reconstruction)에서 블록-별(block-wise) 인트라 예측을 위한 디코딩 지연시간도 제거하는 더 낮은 복잡성 파이프라인을 제공한다. 인트라 예측은 인터 및 인트라 슬라이스 모두에 대해 변형된 도메인에서 수행된다.

인트라 예측은 슬라이스 타입(slice type)에 관계없이 항상 변형된 도메인에서 수행된다. 이러한 배치(arrangement)로, 이전 TU 재구성이 완료된 직후 인트라 예측을 시작할 수 있다. 이러한 배치는 또한 슬라이스 의존(slice dependent) 대신 인트라 모드에 대한 통합된 프로세스를 제공할 수 있다. 도 7은 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 테스트한다.

CE12-2에서 인-루프 휘도 변형 변수의 인터 슬라이스 재구성(밝은 음영된 블록들은 변형된 도메인의 신호를 나타냄: 휘도 잔차, 휘도 내 예측 및 휘도 내 재구성됨)

#### 8.2.2 휘도 의존 채도 잔차 스케일링

휘도 의존 채도 잔차 스케일링은 고정점 정수 연산으로 구현되는 승법(multiplicative) 프로세스이다. 채도 잔차 스케일링은 채도 신호와 휘도 신호의 상호 작용을 보상한다. 채도 잔차 스케일링은 TU 레벨에서 적용된다.

보다 구체적으로 다음이 적용된다.

- 인트라의 경우, 재구성된 휘도가 평균화된다.

- 인터의 경우, 예측 휘도가 평균화된다.

평균은 PWL 모델에서 인덱스를 식별하기 위해 이용된다. 인덱스는 스케일링 인자(factor) cScaleInv를 식별한다. 채도 잔차에 그 숫자를 곱한다.

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

### 8.2.3 ILR 부가 정보의 시그널링

파라미터는 (현재) 타일 그룹 헤더(ALF와 유사)로 전송된다. 이들은 40-100비트를 필요로 한다고 한다. 다음 스펙(spec)은 JVET-L1001 버전 9에 기초한다. 추가된 신택스는 아래에 기울임꼴로 강조 표시되어 있다.

*표 3*

*7.3.2.1 시퀀스 파라미터 세트 RBSP 신택스에서*

| seq_parameter_set_rbsp( ) { | Descriptor |
|---|---|
| sps_seq_parameter_set_id | ue(v) |
| ... | |
| 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( ) | |
| } | |

*표 4*

*7.3.3.1 일반적인 타일 그룹 헤더 신택스에서*

| tile_group_header( ) { | Descriptor |
|---|---|
| ... | |
| 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_chr oma_residual_scale_flag* | *u(1)* |
| *}* | |

| | |
|---|---|
| byte_alignment( ) | |
| } | |

표 5

새 신택스 테이블 타일 그룹 변형 변수 모델을 추가한다:

| tile_group_reshaper_model () { | Descriptor |
|---|---|
| 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 시맨틱(semantics)에서 다음 시맨틱을 추가한다:sps_reshaper_enabled_flag가 1이면 변형 변수가 코딩된 비디오 시퀀스(coded video sequence, CVS)에서 사용됨을 명시한다. sps_reshaper_enabled_flag가 0이면 변형 변수가 CVS에서 사용되지 않음을 명시한다

타일 그룹 헤더 신택스에 다음 시맨틱을 추가한다.

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

tile_group_reshaper_enabled_flag가 1이면 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_chroma_residual_scale_flag가 1이면 현재 타일 그룹에 대해 채도 잔차 스케일링이 활성화됨을 명시한다. tile_group_reshaper_chroma_residual_scale_flag이 0이면 채도 잔차 스케일링이 현재 타일 그룹에 대해 활성화되지 않음을 명시한다. tile_group_reshaper_chroma_residual_scale_flag가 존재하지 않는 경우, 0과 동일한 것으로 유추된다

tile_group_reshaper_model( ) 신택스 추가

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

reshape_model_delta_max_bin_idx는 최대 허용된 빈(또는 조각) 인덱스 MaxBinIdx에서 변형 변수 구성 프로세스에 사용될 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 값(absolute delta codeword value)을 명시한다.

reshaper_model_bin_delta_sign_CW_flag[ i ]는 다음과 같이 reshape_model_bin_delta_abs_CW[ i ]의 부호(sign)를 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 유추된다.

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

변수 RspCW[ i ]는 다음 단계에 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 ].

- 그렇지 않으면, RspCW[ i ] = 0이다.

RspCW [ i ]의 값은 BitDepthY의 값이 10과 같으면 32 내지 2 * OrgCW - 1의 범위에 있어야 한다.

0 내지 MaxBinIdx + 1(포함)의 범위에 있는 i가 있는 변수 InputPivot[ i ]는 다음과 같이 도출된다:

InputPivot[ i ] = i * OrgCW

i가 0 내지 MaxBinIdx + 1의 범위에 있는 변수 ReshapePivot[ i ], 0 내지 MaxBinIdx의 범위에 있는 i가 있는 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 ]
}
```

변수 ChromaScaleCoef[ i ]는 0 내지 MaxBinIdx(포함)의 범위에 있으며 다음과 같이 도출된다:

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]
```

#### 8.2.4 ILR의 사용

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

전체 화상(또는 타일 그룹)이 인코딩/디코딩을 완료한 후, 변형된 도메인의 샘플들이 원본 도메인으로 전환된 다음, 디블로킹 필터 및 기타 필터들이 적용된다.

예측 신호에 대한 순방향 변형은 다음과 같은 경우들에 비활성화된다:

- 현재 블록이 인트라 코딩됨

- 현재 블록은 CPR(현재 화상 참조(current picture referencing), 일명 인트라 블록 복사(intra block copy), IBC)로 코딩됨

- 현재 블록은 결합된 인터-인트라 모드(Combined Inter-Intra Mode, CIIP)로 코딩되고 순방향 변형은 인트라 예측 블록에 대해 비활성화됨

**9. 기존 구현들의 단점들**

JVET-N0242의 비-선형 ALF(NLALF) 설계에는 다음과 같은 문제들이 있다:

(1) 4:2:0 색상 형식을 위해 설계되었다. 4:4:4 색상 형식의 경우 휘도 및 채도 구성요소들이 비슷한 중요성을 가질 수 있다. NLALF를 더 잘 적용하는 방법은 알려져 있지 않다.

(2) 클리핑 값들은 10-비트 경우에 대해 설계되었다. 다른 비트-깊이에 대해 NLALF를 정의하는 방법은 아직 연구되지 않았다.

(3) 인-루프 변형 방법과 NLALF의 상호작용은 연구되지 않았다.

**10 비-선형 적응형 루프 필터링 개선을 위한 예시적인 방법**

현재 개시된 기술의 실시예들은 기존 구현의 단점들을 극복함으로써 더 높은 코딩 효율을 가지는 비디오 코딩을 제공한다. 개시된 기술에 기초한 비-선형 적응형 루프 필터링은 기존 및 미래의 비디오 코딩 표준 모두를 향상시킬 수 있고, 다양한 구현들에 대해 설명되는 다음 예시들에서 설명된다. 아래에 제공된 개시된 기술의 예시들은 일반적인 개념들을 설명하고, 제한하는 것으로 해석되지 않는다. 예에서, 반대로 명시적으로 나타내지 않는 한, 이러한 예에서 설명된 다양한 특징들이 결합될 수 있다.

1. NLALF에서 사용되는 파라미터들(예: 표 2에 정의된 클리핑 파라미터들)은 코딩된 정보에 의존할 수 있음이 제안된다.

a. NLALF에서 사용되는 파라미터들(예: 표 2에 정의된 클리핑 파라미터들)은 시간적 레이어 인덱스/저지연 체크 플래그(low delay check flag)/참조 화상들에 의존할 수 있음이 제안된다.

2. NLALF 파라미터들의 다중 세트들이 정의되거나 시그널링될 수 있다.

a. 대안적으로, 또한, NLALF 파라미터들의 다중 세트들이 시그널링될 때, 적응 파라미터 세트(Adaptation Parameter Set, APS)/타일 그룹 헤더/비디오 데이터 유닛과 같은 데이터 유닛들에서 시그널링될 수 있다.

b. 일 예에서, NLALF 파라미터들은 예측 방식(predictive way)으로 시그널링된다.

i. 예를 들어, 하나의 데이터 유닛(APS 또는 타일 그룹 또는 슬라이스와 같은)에서 시그널링된 한 세트의 NLALF 파라미터들은 동일한 데이터 유닛에서 시그널링되는 다른 NLALF 파라미터들의 세트에 의해 예측된다.

ii. 예를 들어, 하나의 데이터 유닛(APS 또는 타일 그룹, 또는 슬라이스와 같은)에서 시그널링된 한 세트의 NLALF 파라미터들은 다른 데이터 유닛에서 시그널링된 다른 NLALF 파라미터들의 세트에 의해 예측된다.

3. NLALF에서 사용되는 파라미터들(예: 표 2에 정의된 클리핑 파라미터들)은 NLALF를 적용하기 전에 재구성된 샘플들의 비트-깊이에 의존할 수 있음이 제안된다.

a. 또는, NLALF에서 사용되는 파라미터들(예: 표 2에 정의된 클리핑 파라미터들)은 인코딩/디코딩되기 전의 샘플들의 입력 비트-깊이에 의존할 수 있음이 제안된다.

b. 일 예에서, 하나의 주어진 비트-깊이에 대한 파라미터들은 다른 비트-깊이에 대해 할당된 것으로부터 도출될 수 있다.

i. 일 예에서, 비트-깊이에 따른 시프팅 연산(shifting operation)은 하나의 주어진 비트-깊이에 대한 파라미터를 도출하기 위해 적용될 수 있다.

4. NLALF에서 사용되는 파라미터들(예: 표 2에 정의된 클리핑 파라미터들)은 색상 표현 형식에 의존할 수 있음이 제안된다.

a. 일 예에서 RGB의 경우 G 색상 구성요소와 B/R 색상 구성요소들에 대해 동일한 인덱스를 가지는 파라미터이다.

5. NLALF에서 사용되는 파라미터들(예: 표 2에 정의된 클리핑 파라미터들)은 인-루프 변형(ILR) 방식의 적용 여부에 의존할 수 있음이 제안된다.

a. 일 예에서 ILR이 활성화되거나 비활성화되는 경우, 파라미터들이 다를 수 있다.

6. 필터 파라미터들(예: 필터 계수들)과 NLALF 파라미터들(예: 클리핑 파라미터들)을 함께 저장하는 것이 제안된다.

a. 일 예에서, 둘 다 APS에 저장될 수 있다.

b. 일 예에서, 하나의 비디오 데이터 유닛(예: CTU/영역/타일 그룹)이 하나의 APS와 연관된 필터 계수들을 사용하는 경우, 연관된 NLALF 파라미터들이 또한 이용될 수도 있다.

c. 대안으로, 하나의 비디오 데이터 유닛(예: CTU/영역/타일 그룹)을 코딩/디코딩하기 위해, 하나의 APS와 연결된 필터 계수들로부터 예측이 활성화되는 경우, 연관된 NLALF 파라미터들은 또한 동일한 APS로부터 하나의 비디오 데이터 유닛에 대한 NLALF 파라미터들을 예측하기 위해 이용될 수 있다.

7. 채도 색상 구성요소들에 대한 NLALF를 다루는 방법은 색상 형식에 의존할 수 있다.

a. 일 예에서, 하나의 주어진 색상 형식(4:4:4와 같은)에 대해, 2개의 채도 구성요소들은 상이한 NLALF 파라미터들을 사용할 수 있다.

8. ALF의 클리핑은 시퀀스 레벨, 화상 레벨, 슬라이스 레벨, 타일 그룹 레벨, 타일 레벨, CTU 레벨, CU 레벨 또는 블록 레벨에서 켜거나 끌 수 있다고 제안된다.

a. 예를 들어, ALF에서 클리핑을 켤지 여부는 SPS, PPS, 슬라이스 헤더, 타일 그룹 헤더, 타일, CTU, CU 또는 블록과 같은 디코더에 시그널링될 수 있다.

위에서 설명된 예들은 아래에 설명된 방법, 예를 들어, 비디오 디코더 또는 비디오 인코더에서 구현될 수도 있는 방법들(810 내지 840)의 맥락에서 통합될 수도 있다.

도 8a는 비디오 처리를 위한 예시적인 방법의 흐름도를 도시한다. 방법(810)은 단계(812)에서 비디오의 비디오 유닛을 인코딩된 비디오 유닛으로 인코딩하는 단계를 포함한다. 방법(810)은 단계(813)에서 인코딩된 비디오 유닛으로부터 재구성 샘플들을 생성하는 단계를 더 포함한다. 방법(810)은 단계(814)에서 재구성 샘플들에 대해 클리핑 연산을 수행하는 단계를 더 포함하고, 클리핑 연산에 사용되는 클리핑 파라미터는 비디오 유닛의 샘플들의 비트-깊이 또는 재구성 샘플들의 비트-깊이 및 재구성 샘플들의 클리핑 인덱스의 함수이다. 방법(810)은 단계(815)에서 클리핑 연산의 출력에 비-선형 적응형 루프 필터를 적용하는 단계를 더 포함한다. 방법(810)은 단계(816)에서 인코딩된 비디오 유닛을 사용하여 비디오의 코딩된 표현을 생성하는 단계를 더 포함한다.

도 8b는 비디오 처리를 위한 예시적인 방법의 흐름도를 도시한다. 방법(820)은 단계(822)에서 비디오의 비디오 유닛을 나타내는 인코딩된 비디오 유닛에 대해 비디오의 코딩된 표현을 파싱하는 단계를 포함한다. 방법(820)은 단계(823)에서 인코딩된 비디오 유닛으로부터 비디오 유닛의 재구성 샘플들을 생성하는 단계를 더 포함한다. 방법(820)은 단계(824)에서 재구성 샘플들에 대해 클리핑 연산을 수행하는 단계를 포함하고, 클리핑 연산에 사용되는 클리핑 파라미터는 비디오 유닛의 샘플들의 비트-깊이 또는 재구성 샘플들의 비트-깊이 및 재구성 샘플들의 클리핑 인덱스의 함수이다. 방법(820)은 단계(825)에서 최종 디코딩된 비디오 유닛을 생성하기 위해 클리핑 연산의 출력에 비-선형 적응형 루프 필터를 적용하는 단계를 더 포함한다.

도 8c는 비디오 처리를 위한 예시적인 방법의 흐름도를 도시한다. 방법(830)은 단계(832)에서 하나 이상의 비

디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 사이의 전환을 수행하는 단계를 포함한다. 방법(830)은 단계(834)에서 비-선형 적응형 루프 필터를 사용하여 비디오 영역의 비디오 유닛의 재구성을 필터링하기 위한 클리핑 파라미터를 결정하는 단계를 더 포함한다. 일부 구현들에서, 결정하는 단계는, 비디오 및/또는 비디오 영역 및/또는 비디오 유닛의 코딩된 정보에 기초한다. 일부 구현에서, 클리핑 파라미터는, 색상 표현 형식의 함수이다. 일부 구현에서, 클리핑 파라미터는, 제1 도메인 및 제2 도메인에서 비디오 유닛의 표현에 기초하여 비디오 유닛을 재구성하기 위해 및/또는 채도 비디오 유닛의 채도 잔차를 스케일링하기 위해 인-루프 변형(ILR)이 적용되는지 여부에 의존한다.

도 9는 비디오 처리를 위한 예시적인 방법의 흐름도를 도시한다. 방법(840)은 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 사이의 전환을 수행하는 단계를 포함한다. 일부 구현들에서, 코딩된 표현은, 비-선형 적응형 루프 필터를 사용하여 비디오 영역의 비디오 유닛의 재구성을 필터링하기 위한 클리핑 파라미터를 제공하는 제1 부가 정보를 포함하고, 제1 부가 정보는, 비-선형 적응형 루프 필터에서 사용되는 필터 계수들을 나타내는 제2 부가 정보와 함께 시그널링된다. 일부 구현들에서, 코딩된 표현은, 비-선형 적응형 루프 필터를 사용하여 비디오 영역의 비디오 유닛의 재구성을 필터링하기 위한 클리핑 파라미터의 다중 세트들을 나타내는 부가 정보를 포함한다. 일부 구현들에서, 코딩된 표현은, 비-선형 적응형 루프 필터를 사용하여 비디오 영역의 채도 비디오 유닛의 재구성을 필터링하기 위한 하나 이상의 클리핑 파라미터들을 제공하는 부가 정보를 포함하고, 하나 이상의 클리핑 파라미터들은, 색상 형식에 의존한다. 일부 구현들에서, 코딩된 표현은, 적응형 루프 필터를 사용하여 비디오 영역의 비디오 유닛의 재구성을 필터링하기 위한 클리핑 파라미터를 제공하는 부가 정보를 포함하고, 수행하는 단계는, 비디오 영역 레벨에서 샘플 차이들에 클리핑 연산을 적용하는 것에 의해 필터링된 비디오 유닛을 생성하는 단계를 포함한다.

## 11 개시된 기술의 예시적인 구현들

도 10a는 비디오 처리 장치(900)의 블록도이다. 장치(900)는 본 명세서에서 설명된 방법들 중 하나 이상을 구현하기 위해 이용될 수 있다. 장치(900)는 스마트폰, 태블릿, 컴퓨터, 사물인터넷(IoT) 수신기 등으로 구현될 수 있다. 장치(900)는 하나 이상의 프로세서들(902), 하나 이상의 메모리들(904) 및 비디오 처리 하드웨어(906)를 포함할 수 있다. 프로세서(들)(902)는 본 명세서에 설명된 하나 이상의 방법들(방법 800을 포함하지만 이에 국한되지 않음)을 구현하도록 구성될 수 있다. 메모리(메모리들)(904)는 본 명세서에서 설명된 방법들 및 기술들을 구현하기 위해 이용되는 데이터 및 코드를 저장하기 위해 이용될 수 있다. 비디오 처리 하드웨어(906)는 하드웨어 회로에서 본 명세서에 설명된 일부 기술들을 구현하기 위해 이용될 수 있다.

도 10b는 개시된 기술들이 구현될 수 있는 비디오 처리 시스템의 블록도의 다른 예이다. 도 10b는 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 연산은 인코더에서 사용되며, 코딩의 결과를 역전시키는 대응하는 디코딩 도구 또는 연산은 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 기

술들은 디지털 데이터 처리 및/또는 비디오 디스플레이를 수행할 수 있는 휴대폰, 랩톱, 스마트폰 또는 기타 장치와 같은 다양한 전자 장치에서 구현될 수 있다.

개시된 기술의 일부 실시예들은 비디오 처리 도구 또는 모드를 활성화하기 위한 결정 또는 결정을 내리는 것을 포함한다. 일 예에서, 비디오 처리 도구 또는 모드가 활성화되는 경우, 인코더는 비디오 블록의 처리에서 도구 또는 모드를 사용하거나 구현하지만 도구 또는 모드의 사용에 기초하여 결과 비트스트림을 반드시 수정할 필요는 없다. 즉, 비디오 블록으로부터 비디오의 비트스트림 표현으로의 전환은 결정에 기초하여 활성화되는 경우, 비디오 처리 도구 또는 모드를 사용한다. 다른 예에서, 비디오 처리 도구 또는 모드가 활성화되는 경우, 디코더는 비디오 처리 도구 또는 모드에 기초하여 비트스트림이 수정되었다는 지식으로 비트스트림을 처리할 것이다. 즉, 비디오의 비트스트림 표현으로부터 비디오 블록으로의 전환은 결정에 기초하여 활성화된 비디오 처리 도구 또는 모드를 사용하여 수행될 것이다.

개시된 기술의 일부 실시예들은 비디오 처리 도구 또는 모드를 비활성화하기 위한 결정 또는 결정을 내리는 것을 포함한다. 일 예에서, 비디오 처리 도구 또는 모드가 비활성화되는 경우, 인코더는 비디오 블록을 비디오의 비트스트림 표현으로 전환하기 위해 도구 또는 모드를 사용하지 않는다. 다른 예에서, 비디오 처리 도구 또는 모드가 비활성화되는 경우, 디코더는 결정에 기초하여 비활성화된 비디오 처리 도구 또는 모드를 사용하여 비트스트림이 수정되지 않았다는 지식으로 비트스트림을 처리할 것이다.

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.

일부 실시예들에서, 비디오 코딩 방법은 도 10a 또는 10b와 관련하여 설명된 바와 같이 하드웨어 플랫폼 상에서 구현되는 장치를 사용하여 구현될 수 있다.

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

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

1. 현재 비디오 블록의 코딩된 정보에 기초하여, 현재 비디오 블록에 대한 파라미터들의 세트를 결정하는 단계; 및 파라미터들의 세트를 사용하여 비-선형 필터링 연산을 수행하는 것에 기초하여, 대응하는 비트스트림 표현으로부터 현재 비디오 블록을 재구성하는 단계를 포함하는 비디오 데이터 처리 방법.

2. 1절에 있어서, 비-선형 필터링 연산은 비-선형 적응형 루프 필터링을 포함하는 방법.

3. 1절 또는 2절에 있어서, 파라미터들의 세트는 현재 비디오 블록의 휘도 구성요소 또는 채도 구성요소에 대한 적어도 하나의 클리핑 값을 포함하는 방법.

4. 3절에 있어서, 비-선형 필터링 연산은 채도 구성요소의 색상 형식에 기초하는 방법.

5. 1절 내지 3절 중 어느 하나에 있어서, 코딩된 정보는 시간적 레이어 인덱스 저지연 체크 플래그 또는 하나 이상의 참조 화상들을 포함하는 방법.

6. 1절 내지 3절 중 어느 하나에 있어서, 코딩된 정보는 비-선형 필터링 연산 이전의 재구성된 샘플들의 비트-깊이를 포함하는 방법.

7. 1절 내지 3절 중 어느 하나에 있어서, 코딩된 정보는 색상 표현 형식을 포함하는 방법.

8. 1절 내지 3절 중 어느 하나에 있어서, 코딩된 정보는 인-루프 변형(ILR) 방법을 적용하는 표시(indication)를 포함하는 방법.

9. 1절 내지 3절 중 어느 하나에 있어서, 대응하는 비트스트림 표현은 파라미터들의 세트를 포함하는 파라미터들의 다중 세트들을 포함하고, 파라미터들의 다중 세트들은 적응 파라미터 세트(APS), 타일 그룹 헤더 또는 하나 이상의 비디오 데이터 유닛들에서 시그널링되는 방법.

10. 1절 내지 3절 중 어느 하나에 있어서, 대응하는 비트스트림 표현은 비-선형 필터링 연산과 연관된 파라미터들 및 하나 이상의 필터 계수들의 세트를 포함하는 적응 파라미터 세트APS를 포함하는 방법.

11. 1절 또는 2절에 있어서, 파라미터들의 세트는 하나 이상의 클리핑 값들을 포함하고, 비-선형 필터링 연산은 시퀀스 레벨, 화상 레벨, 슬라이스 레벨, 타일 그룹 레벨, 타일 레벨, 코딩 트리 유닛(coding tree unit, CTU) 레벨, 코딩 유닛(coding unit, CU) 레벨 또는 블록 레벨에서 수행되는 방법.

12. 프로세서 및 그에 대한 명령어들이 있는 비일시적 메모리를 포함하는 비디오 시스템의 장치에 있어서, 프로세서에 의한 실행 시에 명령어들은 프로세서가 1절 내지 11절 중 하나 이상에 기재된 비디오 처리 방법을 실행하게 하는 장치.

13. 비일시적 컴퓨터 판독 가능 매체에 저장된 컴퓨터 프로그램 제품에 있어서, 1절 내지 11절 중 하나 이상에 기재된 비디오 처리 방법을 수행하기 위한 프로그램 코드를 포함하는 컴퓨터 프로그램 제품.

절들의 제2 세트는 예를 들어 예 1 및 3-5를 포함하는 이전 섹션에 나열된 개시된 기술들의 특정 특징 및 양태들을 설명한다.

1. 비디오의 비디오 유닛을 인코딩된 비디오 유닛으로 인코딩하는 단계; 인코딩된 비디오 유닛으로부터 재구성 샘플들을 생성하는 단계; 재구성 샘플들에 대해 클리핑 연산을 수행하는 단계 - 클리핑 연산에 사용되는 클리핑 파라미터는 비디오 유닛의 샘플들의 비트-깊이 또는 재구성 샘플들의 비트-깊이 및 재구성 샘플들의 클리핑 인덱스의 함수임 -; 클리핑 연산의 출력에 비-선형 적응형 루프 필터를 적용하는 단계; 및 인코딩된 비디오 유닛을 사용하여 비디오의 코딩된 표현을 생성하는 단계를 포함하는 비디오 데이터 처리 방법.

2. 1절에 있어서, 클리핑 인덱스는, 코딩된 표현에서 시그널링되는 방법.

3. 비디오의 비디오 유닛을 나타내는 인코딩된 비디오 유닛에 대해 비디오의 코딩된 표현을 파싱하는 단계; 인코딩된 비디오 유닛으로부터 비디오 유닛의 재구성 샘플들을 생성하는 단계; 재구성 샘플들에 대해 클리핑 연산을 수행하는 단계 - 클리핑 연산에 사용되는 클리핑 파라미터는 비디오 유닛의 샘플들의 비트-깊이 또는 재구성 샘플들의 비트-깊이 및 재구성 샘플들의 클리핑 인덱스의 함수임 - ; 및 최종 디코딩된 비디오 유닛을 생성하기 위해 클리핑 연산의 출력에 비-선형 적응형 루프 필터를 적용하는 단계를 포함하는 비디오 데이터 처리 방법.

4. 3절에 있어서, 클리핑 인덱스는 적어도 코딩된 표현의 필드에 기초하여 결정되는 방법.

5. 3절에 있어서, 클리핑 인덱스는, 미리-정의된 규칙을 이용하여 결정되는 방법.

6. 1절 또는 3절에 있어서, 비디오 유닛의 샘플들의 비트-깊이 또는 재구성 샘플들의 비트-깊이 및 재구성 샘플들의 클리핑 인덱스의 함수는 재구성 샘플들의 비트-깊이 또는 비디오 유닛의 비트-깊이에 기초하는 클리핑 인덱스의 주어진 값에 대해 다른 값들을 반환하는 함수인 방법.

7. 1절 또는 3절에 있어서, 클리핑 인덱스와 클리핑 파라미터 사이의 매핑은 재구성 샘플의 비트-깊이 또는 비디오 유닛의 비트-깊이에 의존하는 방법.

8. 1절 또는 3절에 있어서, 주어진 비트-깊이에 대한 제1 클리핑 인덱스에 대응하는 제1 클리핑 값은 제2 클리핑 인덱스에 대응하는 제2 클리핑 값에 기초하여 도출되는 방법.

9. 8절에 있어서, 다른 비트-깊이를 사용하는 시프팅 연산은 주어진 비트-깊이에 대한 클리핑 파라미터를 도출하기 위해 적용되는 방법.

10. 1절 내지 9절 중 어느 하나에 있어서, 코딩된 표현은 비-선형 적응형 루프 필터에서 사용되는 2개의 샘플 차이들의 상한 또는 하한(upper or lower bound)을 제어하는 클리핑 파라미터를 포함하는 방법.

11. 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 사이의 전환을 수행하는 단계; 및 비-선형 적응형 루프 필터를 사용하여 비디오 영역의 비디오 유닛의 재구성을 필터링하기 위한 클리핑 파라미터를 결정하는 단계를 포함하고, 결정하는 단계는 비디오 및/또는 비디오 영역 및/또는 비디오 유닛의 코딩된 정보에 기초하는 비디오 데이터 처리 방법.

12. 11절에 있어서, 코딩된 정보는 시간적 레이어 인덱스를 포함하는 방법.

13. 11절에 있어서, 코딩된 정보는 저지연 체크 플래그를 포함하는 방법.

14. 11절에 있어서, 코딩된 정보는 하나 이상의 참조 화상들을 포함하는 방법.

15. 11절 내지 14절 중 어느 하나에 있어서, 비디오 영역은 동화상을 포함하는 방법.

16. 11절 내지 14절 중 어느 하나에 있어서, 비디오 유닛은 코딩 유닛을 포함하는 방법.

17. 11절 내지 16절 중 어느 하나에 있어서, 클리핑 파라미터는 비-선형 적응형 루프 필터에서 사용되는 2개의 샘플 차이들의 상한 또는 하한을 제어하는 방법.

18. 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 사이의 전환을 수행하는 단계; 및 비-선형 적응형 루프 필터를 사용하여 비디오 영역의 비디오 유닛의 재구성을 필터링하기 위한 클리핑 파라미터를 결정하는 단계를 포함하고, 클리핑 파라미터는 색상 표현 형식의 함수인 비디오 데이터 처리 방법.

19. 18절에 있어서, RGB 색상 형식에 대한 클리핑 파라미터는, 녹색 구성요소 및 청색 또는 적색 구성요소에 대해 동일한 인덱스를 가지는 방법.

20. 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 사이의 전환을 수행하는 단계; 및 비-선형 적응형 루프 필터를 사용하여 비디오 영역의 비디오 유닛의 재구성을 필터링하기 위한 클리핑 파라미터를 결정하는 단계를 포함하고, 클리핑 파라미터는 제1 도메인 및 제2 도메인에서 비디오 유닛의 표현에 기초하여 비디오 유닛을 재구성하기 위해 및/또는 채도 비디오 유닛의 채도 잔차를 스케일링하기 위해 인-루프 변형(ILR)이 적용되는지 여부에 의존하는 비디오 데이터 처리 방법.

21.1절 내지 21절 중 어느 하나에 있어서, 클리핑 파라미터는 비디오 유닛의 휘도 구성요소 또는 채도 구성요소에 대한 클리핑 값에 대응하는 방법.

22. 1절 내지 21절 중 어느 하나에 있어서, 전환 동안, 비디오 유닛의 재구성에 비-선형 적응형 루프 필터를 적용하여 필터링된 비디오 유닛을 생성하는 단계; 및 비디오의 다른 비디오 유닛의 예측을 결정하기 위해 필터링된 비디오 유닛을 사용하는 단계를 더 포함하는 방법.

23. 1절 내지 22절 중 어느 하나에 있어서, 전환을 수행하는 단계는 비디오로부터 코딩된 표현을 생성하는 단계를 포함하는 방법.

24. 1절 내지 22절 중 어느 하나에 있어서, 전환을 수행하는 단계는 코딩된 표현으로부터 비디오를 생성하는 단계를 포함하는 방법.

25. 프로세서 및 그에 대한 명령어들이 있는 비일시적 메모리를 포함하는 비디오 시스템의 장치에 있어서, 프로세서에 의한 실행 시에 명령어들은 프로세서가 1절 내지 24절 중 하나 이상에 기재된 비디오 처리 방법을 실행하게 하는 장치.

26. 비일시적 컴퓨터 판독 가능 매체에 저장된 컴퓨터 프로그램 제품에 있어서, 1절 내지 24절 중 하나 이상에 항에 기재된 비디오 처리 방법을 수행하기 위한 프로그램 코드를 포함하는 컴퓨터 프로그램 제품.

절들의 제3 세트는 예를 들어 예 2 및 6-8을 포함하는 이전 섹션에 나열된 개시된 기술들의 특정 특징 및 양태들을 설명한다.

1. 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 사이의 전환을 수행하는 단계를 포함하고, 코딩된 표현은 비-선형 적응형 루프 필터를 사용하여 비디오 영역의 비디오 유닛의 재구성을 필터링하기 위한 클리핑 파라미터를 제공하는 제1 부가 정보를 포함하고, 제1 부가 정보는 비-선형 적응형 루프 필터에서 사용되는 필터 계수들을 나타내는 제2 부가 정보와 함께 시그널링되는 비디오 데이터 처리 방법.

2. 1절에 있어서, 제1 부가 정보 및 제2 부가 정보는 동일한 적응 파라미터 세트에서 시그널링되는 방법.

3. 1절에 있어서, 적응 파라미터 세트와 연관된 필터 계수들 중 적어도 일부가 비디오 데이터 유닛에 의해 사용되는 경우, 적응 파라미터 세트와 연관된 클리핑 파라미터는 비디오 데이터 유닛에 의해서도 사용되는 방법.

4. 1절에 있어서, 적응 파라미터 세트와 연관된 필터 계수들 중 적어도 일부로부터의 예측이 비디오 데이터 유닛의 전환을 위해 활성화되는 경우, 적응 파라미터 세트와 연관된 파라미터는 적응 파라미터 세트로부터 다른 비디오 데이터 유닛에 대한 다른 파라미터를 예측하기 위해 사용되는 방법.

5. 3절 또는 4절에 있어서, 비디오 데이터 유닛은 코딩 트리 유닛, 비디오 영역 또는 타일 그룹인 방법.

6. 1절 내지 5절 중 어느 하나에 있어서, 파라미터는 비디오 유닛의 휘도 구성요소 또는 채도 구성요소에 대한 클리핑 값에 대응하는 방법.

7. 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 사이의 전환을 수행하는 단계를 포함

하고, 코딩된 표현은 비-선형 적응형 루프 필터를 사용하여 비디오 영역의 비디오 유닛의 재구성을 필터링하기 위한 클리핑 파라미터의 다중 세트들을 나타내는 부가 정보를 포함하는 비디오 데이터 처리 방법.

8. 7절에 있어서, 부가 정보는 클리핑 파라미터들의 다중 세트들을 포함하는 방법.

9. 7절에 있어서, 클리핑 파라미터들의 다중 세트들은 인코더 및 디코더에 알려져 있고, 부가 정보는 클리핑 파라미터들의 다중 세트들 중 하나 이상에 대한 인덱스를 포함하는 방법.

10. 7절에 있어서, 클리핑 파라미터들의 다중 세트들은 비디오 데이터 유닛 또는 비디오 유닛의 헤더에 포함되는 방법.

11. 10절에 있어서, 비디오 데이터 유닛은 적응 파라미터 세트, 타일 그룹 또는 슬라이스를 포함하는 방법.

12. 8절에 있어서, 데이터 유닛에서 시그널링된 클리핑 파라미터들의 다중 세트들 중 하나의 세트는 데이터 유닛에서 시그널링되는 클리핑 파라미터들의 다른 세트에 의해 예측되는 방법.

13. 7절에 있어서, 데이터 유닛에서 시그널링된 클리핑 파라미터들의 다중 세트들 중 하나의 세트는 다른 데이터 유닛에서 시그널링되는 클리핑 파라미터들의 다른 세트에 의해 예측되는 방법.

14. 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 사이의 전환을 수행하는 단계를 포함하고, 코딩된 표현은 비-선형 적응형 루프 필터를 사용하여 비디오 영역의 채도 비디오 유닛의 재구성을 필터링하기 위한 하나 이상의 클리핑 파라미터들을 제공하는 부가 정보를 포함하고, 하나 이상의 클리핑 파라미터들은 색상 형식에 의존하는 비디오 데이터 처리 방법.

15. 14절에 있어서, 특정한 색상 형식에 대해, 2개의 채도 구성요소들은 상이한 클리핑 파라미터들을 사용하는 방법.

16. 15절에 있어서, 특정한 색상 형식은 4:4:4인 방법.

17. 1절 내지 16절 중 어느 하나에 있어서, 전환 동안, 비디오 유닛의 재구성에 비-선형 적응형 루프 필터를 적용하여 필터링된 비디오 유닛을 생성하는 단계; 및 비디오의 다른 비디오 유닛의 예측을 결정하기 위한 필터링된 비디오 유닛을 사용하는 단계를 더 포함하는 방법.

18. 하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 사이의 전환을 수행하는 단계를 포함하고, 코딩된 표현은 적응형 루프 필터를 사용하여 비디오 영역의 비디오 유닛의 재구성을 필터링하기 위한 클리핑 파라미터를 제공하는 부가 정보를 포함하고, 수행하는 단계는 비디오 영역 레벨에서 샘플 차이들에 클리핑 연산을 적용하는 것에 의해 필터링된 비디오 유닛을 생성하는 단계를 포함하는 비디오 데이터 처리 방법.

19. 18절에 있어서, 비디오 영역 레벨은 시퀀스 레벨, 화상 레벨, 슬라이스 레벨, 타일 그룹 레벨, 타일 레벨, 코딩 트리 유닛 레벨, 코딩 유닛 레벨, 또는 블록 레벨인 방법.

20. 18절에 있어서, 클리핑 연산을 가능하게 하는 표시는 클리핑 연산을 활성화하는, 시퀀스 파라미터 세트(sequence parameter set, SPS), 화상 파라미터 세트(picture parameter set, PPS), 슬라이스 헤더, 타일 그룹 헤더(tile group header), 타일, 코딩 트리 유닛, 코딩 유닛 또는 블록에서 시그널링되는 방법.

21. 1절 내지 20절 중 어느 하나에 있어서, 비디오 영역은 동화상인 방법.

22. 1절 내지 20절 중 어느 하나에 있어서, 비디오 유닛은 코딩 유닛 또는 변환 유닛 또는 슬라이스 또는 코딩 트리 또는 코딩 트리 행(coding tree row)인 방법.

23. 1절 내지 22절 중 어느 하나에 있어서, 전환을 수행하는 단계는 현재 블록으로부터 코딩된 표현을 생성하는 단계를 포함하는 방법.

24. 1절 내지 22절 중 어느 하나에 있어서, 전환을 수행하는 단계는 코딩된 표현으로부터 현재 블록을 생성하는 단계를 포함하는 방법.

25. 프로세서 및 그에 대한 명령어들이 있는 비일시적 메모리를 포함하는 비디오 시스템의 장치에 있어서, 프로세서에 의한 실행 시에 명령어들은 프로세서가 1절 내지 24절 중 하나 이상에 기재된 비디오 처리 방법을 실행하게 하는 장치.

26. 비일시적 컴퓨터 판독 가능 매체에 저장된 컴퓨터 프로그램 제품에 있어서, 1절 내지 24절 중 하나 이상에 기재된 비디오 처리 방법을 수행하기 위한 프로그램 코드를 포함하는 컴퓨터 프로그램 제품.

전술한 내용으로부터, 현재 개시된 기술의 특정 실시예들이 예시의 목적들로 본 명세서에 설명되었지만, 본 발명의 범위를 벗어나지 않고 다양한 수정들이 이루어질 수 있음을 이해할 것이다. 따라서, 현재 개시된 기술은 첨부된 청구범위에 의한 경우를 제외하고는 제한되지 않는다.

이 특허 문서에 기술된 주제 및 기능적 동작들의 구현은 다양한 시스템, 디지털 전자 회로, 또는 이 명세서에 개시된 구조와 그 구조적 등가물을 포함하는 컴퓨터 소프트웨어, 펌웨어 또는 하드웨어, 또는 이들 중 하나 이상의 조합으로 구현될 수 있다. 본 명세서에 기술된 주제의 구현은 하나 이상의 컴퓨터 프로그램 제품, 즉, 데이터 처리 장치에 의해 실행되거나 데이터 처리 장치의 동작을 제어하기 위해 유형의 비일시적 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 유닛" 또는 "데이터 처리 장치"라는 용어는 예를 들어 프로그램 가능한 프로세서, 컴퓨터, 또는 다중 프로세서 또는 컴퓨터를 포함하여 데이터를 처리하기 위한 모든 장치, 디바이스 및 기계를 포함한다. 장치는 하드웨어 외에도 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.

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.

본 명세서에서 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). 프로세서와 메모리는 특수 목적 논리 회로에 의해 보완되거나 통합될 수 있다.

본 명세서는 도면과 함께 단지 예시적인 것으로 간주되어야 하며, 여기서 예시적인 것은 예시를 의미한다. 본 명세서에 사용된 바와 같이, 단수 형태 "a", "an" 및 "the"는 문맥이 명백하게 달리 나타내지 않는 한 복수 형태도 포함하는 것으로 의도된다. 추가적으로, "또는"의 사용은 문맥이 명백하게 달리 나타내지 않는 한 "및/또는"을 포함하도록 의도된다.

이 특허 문서에는 많은 세부 사항이 포함되어 있지만, 이는 모든 발명의 범위 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 적절한 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.

[0051] 삭제

[0052] 삭제

[0053] 삭제

[0054] 삭제

[0055] 삭제

[0056] 삭제

[0057] 삭제

[0058] 삭제

[0059] 삭제

[0060] 삭제

[0061] 삭제

[0062] 삭제

[0063] 삭제

[0064] 삭제

[0065] 삭제

[0066] 삭제

[0067] 삭제

[0068] 삭제

[0069] 삭제

[0070] 삭제

[0071] 삭제

[0072] 삭제

[0073] 삭제

[0074] 삭제

[0075] 삭제

[0076] 삭제

[0077] 삭제

[0078] 삭제

[0079] 삭제

[0080] 삭제

[0081] 삭제

[0082] 삭제

[0083] 삭제

[0084] 삭제

[0085] 삭제

[0086] 삭제

[0087] 삭제

[0088] 삭제

[0089] 삭제

[0090] 삭제

[0091] 삭제

[0092] 삭제

[0093] 삭제

[0094] 삭제

[0095] 삭제

[0096] 삭제

[0097] 삭제

[0098] 삭제

[0099] 삭제

[0100] 삭제

[0101] 삭제

[0102] 삭제

[0103] 삭제

[0104] 삭제

[0105] 삭제

[0106] 삭제

[0107] 삭제

[0108] 삭제

[0109] 삭제

[0110] 삭제

[0111] 삭제

[0112] 삭제

[0113] 삭제

[0114] 삭제

[0115] 삭제

[0116] 삭제

[0117] 삭제

[0118] 삭제

[0119] 삭제

[0120] 삭제

[0121] 삭제

[0122] 삭제

[0123] 삭제

[0124] 삭제

[0125] 삭제

[0126] 삭제

[0127] 삭제

[0128] 삭제

[0129] 삭제

[0130] 삭제

[0131] 삭제

[0132] 삭제

[0133] 삭제

[0134] 삭제

[0135] 삭제

[0136] 삭제

[0137] 삭제

[0138] 삭제

[0139] 삭제

[0140] 삭제

[0141] 삭제

[0142] 삭제

[0143] 삭제

[0144] 삭제

[0145] 삭제

[0146] 삭제

[0147] 삭제

[0148] 삭제

[0149] 삭제

[0150] 삭제

[0151] 삭제

[0152] 삭제

[0153] 삭제

[0154] 삭제

[0155] 삭제

[0156] 삭제

[0157] 삭제

[0158] 삭제

[0159] 삭제

[0160] 삭제

[0161] 삭제

[0162] 삭제

[0163] 삭제

[0164] 삭제

[0165] 삭제

[0166] 삭제

[0167] 삭제

[0168] 삭제

[0169] 삭제

[0170] 삭제

[0171] 삭제

[0172] 삭제

[0173] 삭제

[0174] 삭제

[0175] 삭제

[0176] 삭제

[0177] 삭제

[0178] 삭제

[0179] 삭제

[0180] 삭제

[0181] 삭제

[0182] 삭제

[0183] 삭제

[0184] 삭제

[0185] 삭제

[0186] 삭제

[0187] 삭제

[0188] 삭제

[0189] 삭제

[0190] 삭제

[0191] 삭제

[0192] 삭제

[0193] 삭제

[0194] 삭제

[0195] 삭제

[0196] 삭제

[0197] 삭제

[0198] 삭제

[0199] 삭제

[0200] 삭제

[0201] 삭제

[0202] 삭제

[0203] 삭제

[0204] 삭제

[0205] 삭제

[0206] 삭제

[0207] 삭제

[0208] 삭제

[0209] 삭제

[0210] 삭제

[0211] 삭제

[0212] 삭제

[0213] 삭제

[0214] 삭제

[0215] 삭제

[0216] 삭제

[0217] 삭제

[0218] 삭제

[0219] 삭제

[0220] 삭제

[0221] 삭제

[0222] 삭제

[0223] 삭제

[0224] 삭제

[0225] 삭제

[0226] 삭제

[0227] 삭제

[0228] 삭제

[0229] 삭제

[0230] 삭제

[0231] 삭제

[0232] 삭제

[0233] 삭제

[0234] 삭제

[0235] 삭제

[0236] 삭제

[0237] 삭제

[0238] 삭제

[0239] 삭제

[0240] 삭제

[0241] 삭제

[0242] 삭제

[0243] 삭제

[0244] 삭제

[0245] 삭제

[0246] 삭제

[0247] 삭제

[0248] 삭제

[0249] 삭제

[0250] 삭제

[0251] 삭제

[0252] 삭제

[0253] 삭제

[0254] 삭제

[0255] 삭제

[0256] 삭제

[0257] 삭제

[0258] 삭제

[0259] 삭제

[0260] 삭제

[0261] 삭제

[0262] 삭제

[0263] 삭제

[0264] 삭제

[0265] 삭제

[0266] 삭제

[0267] 삭제

[0268] 삭제

[0269] 삭제

[0270] 삭제

[0271] 삭제

[0272] 삭제

[0273] 삭제

[0274] 삭제

[0275] 삭제

[0276] 삭제

[0277] 삭제

[0278] 삭제

[0279] 삭제

[0280] 삭제

[0281] 삭제

[0282] 삭제

[0283] 삭제

[0284] 삭제

[0285] 삭제

[0286] 삭제

[0287] 삭제

[0288] 삭제

[0289] 삭제

[0290] 삭제

[0291] 삭제

[0292] 삭제

[0293] 삭제

[0294] 삭제

[0295] 삭제

[0296] 삭제

[0297] 삭제

[0298] 삭제

[0299] 삭제

[0300] 삭제

[0301] 삭제

[0302] 삭제

[0303] 삭제

[0304] 삭제

[0305] 삭제

[0306] 삭제

[0307] 삭제

[0308] 삭제

[0309] 삭제

[0310] 삭제

[0311] 삭제

[0312] 삭제

[0313] 삭제

[0314] 삭제

[0315] 삭제

[0316] 삭제

[0317] 삭제

[0318] 삭제

[0319] 삭제

[0320] 삭제

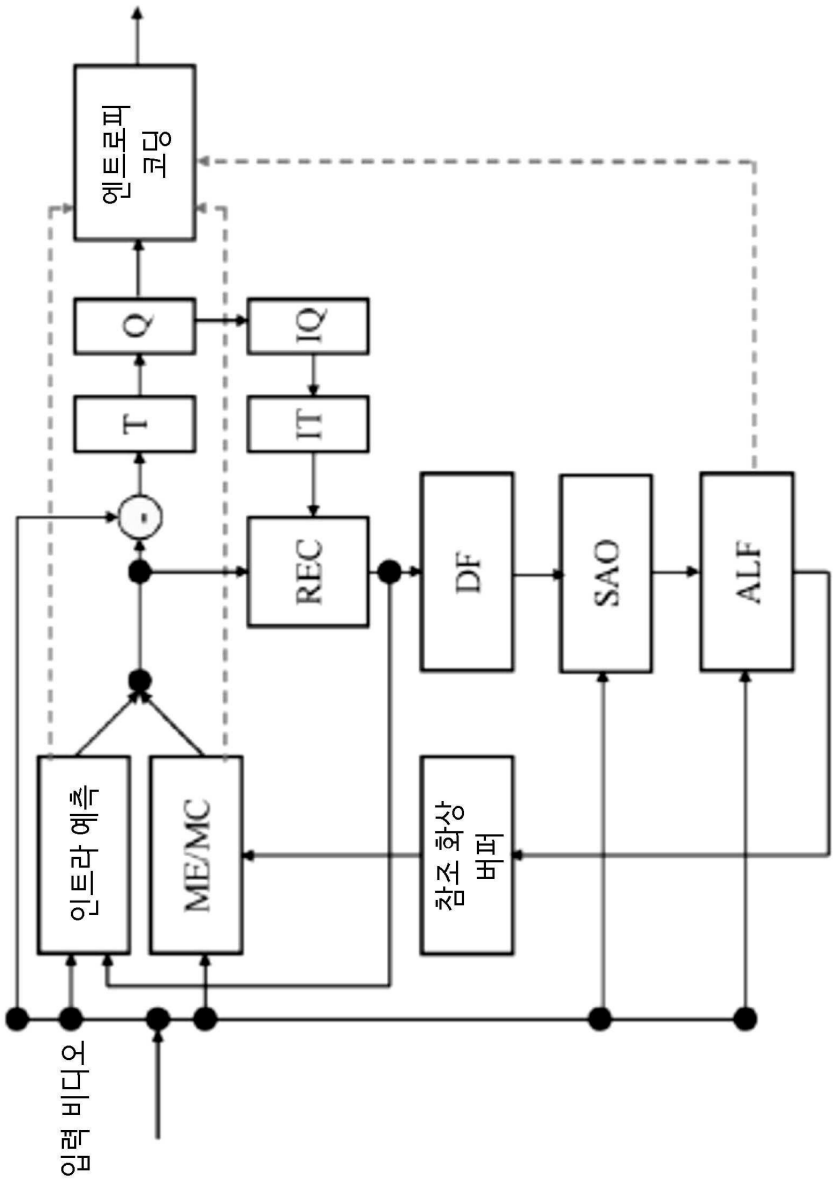
[0321] 삭제

[0322] 삭제

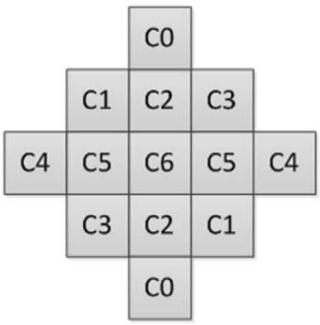
[0323] 삭제

도면

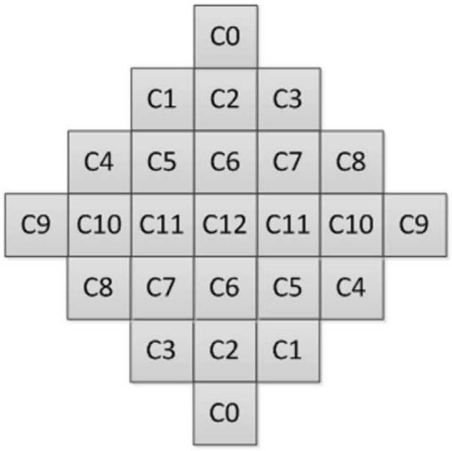
도면1


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

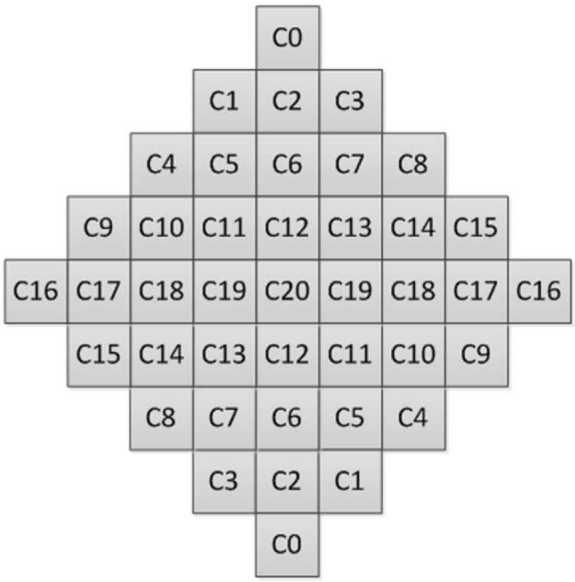
도면2a


C0
C1
C2
C3
C4
C5
C6
C5
C4
C3
C2
C1
C0

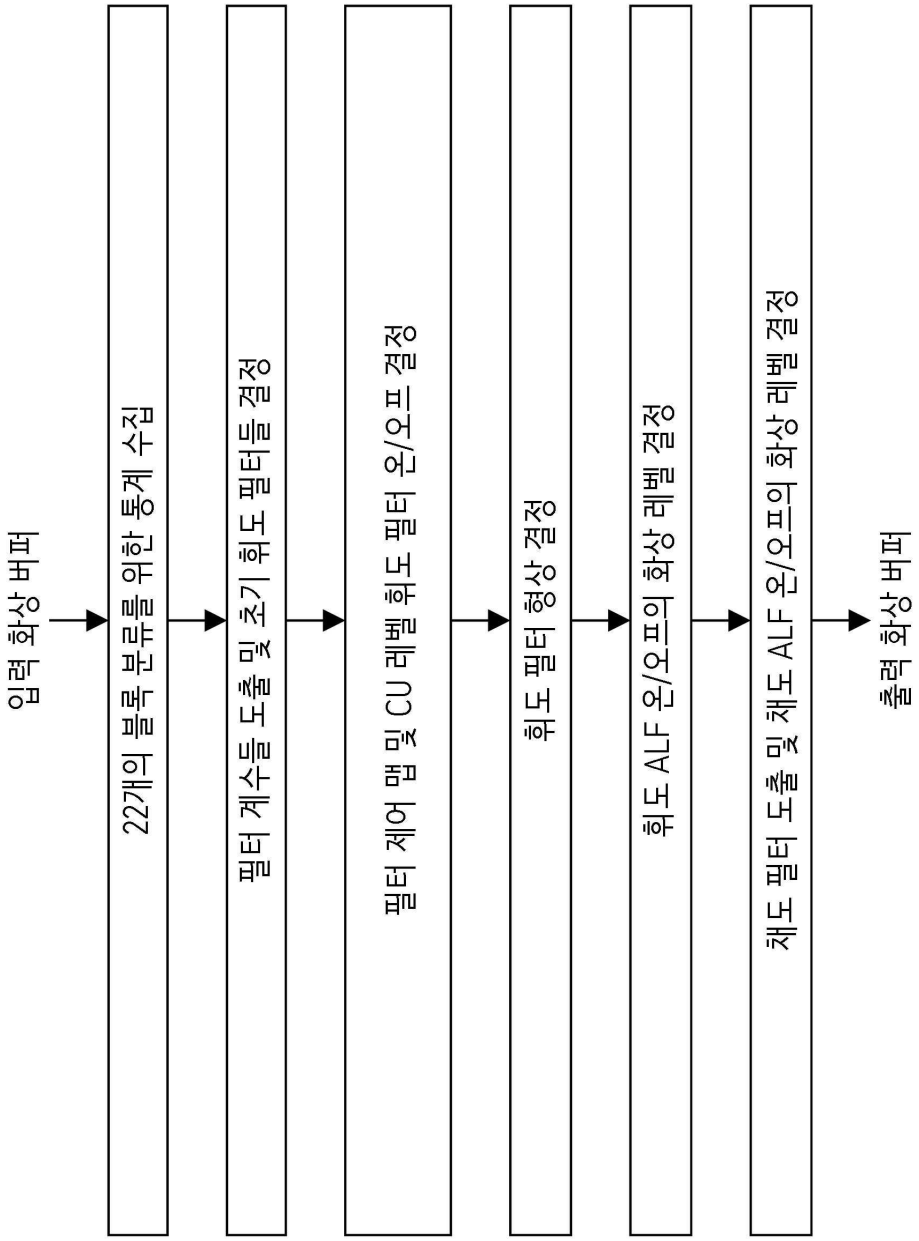
도면2b


C0
C1 C2 C3
C4 C5 C6 C7 C8
C9 C10 C11 C12 C11 C10 C9
C8 C7 C6 C5 C4
C3 C2 C1
C0

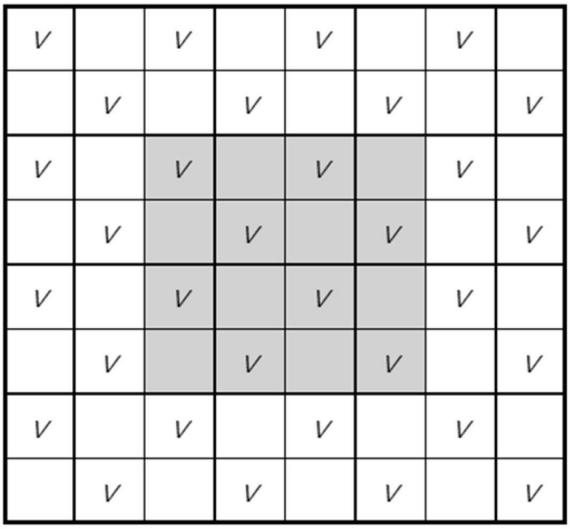
도면2c


C0
C1 C2 C3
C4 C5 C6 C7 C8
C9 C10 C11 C12 C13 C14 C15
C16 C17 C18 C19 C20 C19 C18 C17 C16
C15 C14 C13 C12 C11 C10 C9
C8 C7 C6 C5 C4
C3 C2 C1
C0

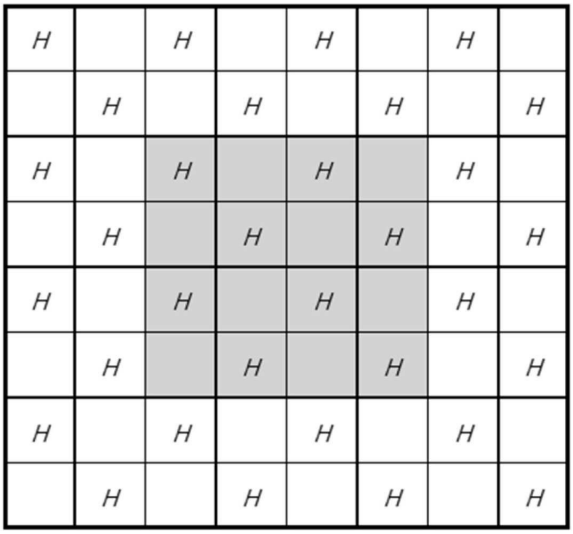
도면3


입력 화상 버퍼
22개의 블록 분류를 위한 통계 수집
필터 계수들 도출 및 초기 휘도 필터들 결정
필터 제어 맵 및 CU 레벨 휘도 필터 온/오프 결정
휘도 필터 형상 결정
휘도 ALF 온/오프의 화상 레벨 결정
채도 필터 도출 및 채도 ALF 온/오프의 화상 레벨 결정
출력 화상 버퍼

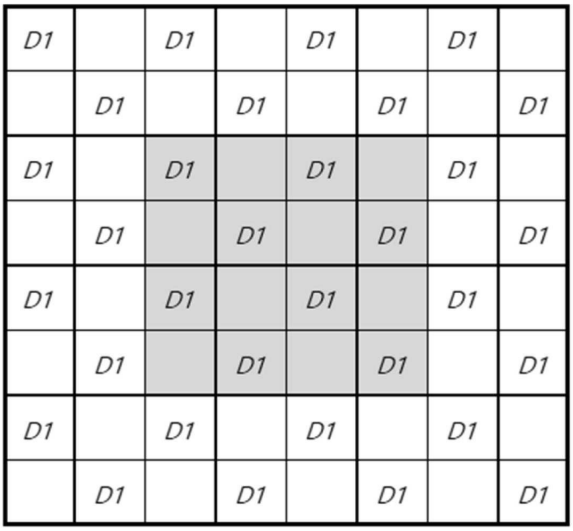
도면4a


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

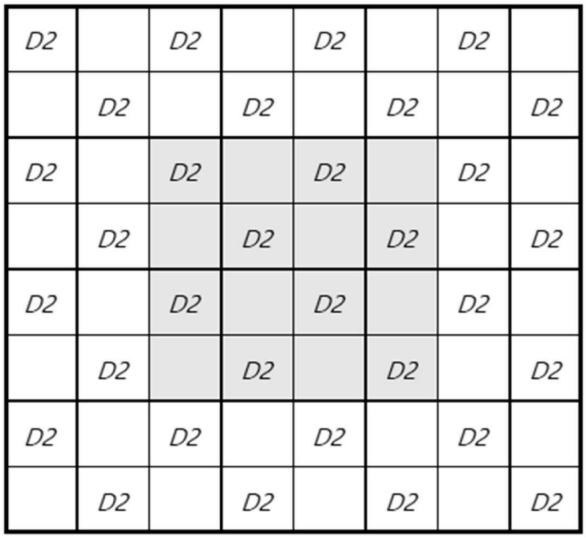
도면4b


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

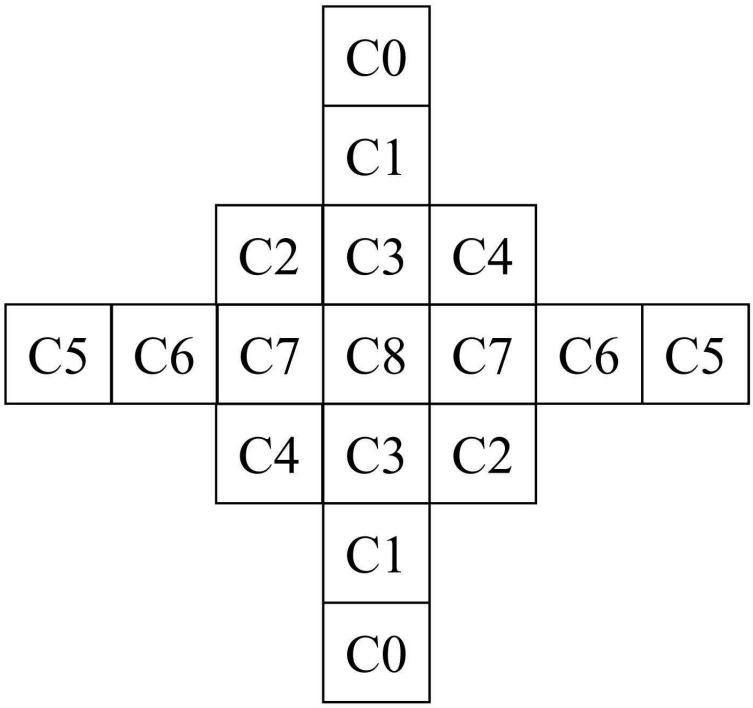
도면4c


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

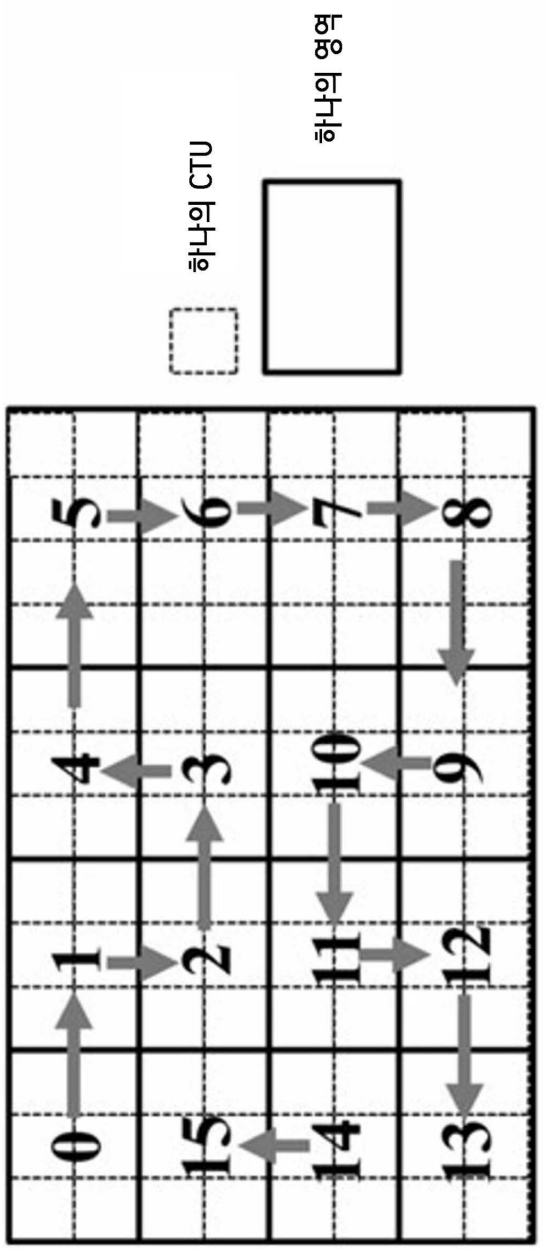
도면4d


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

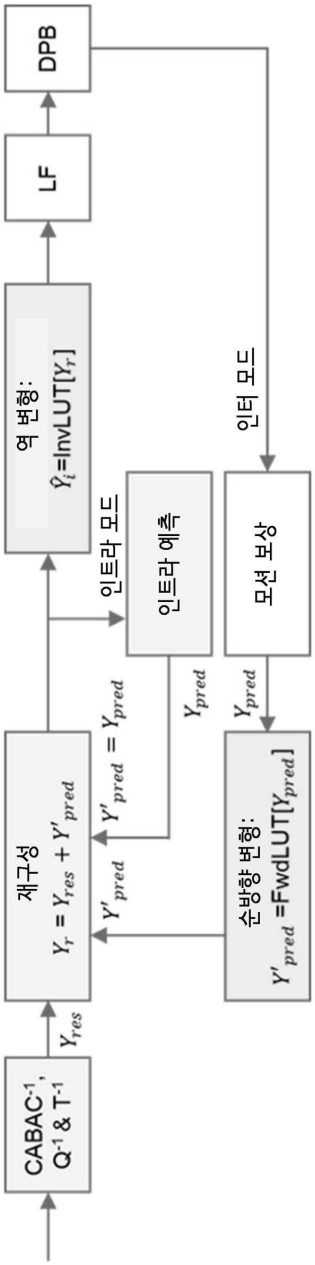
도면5



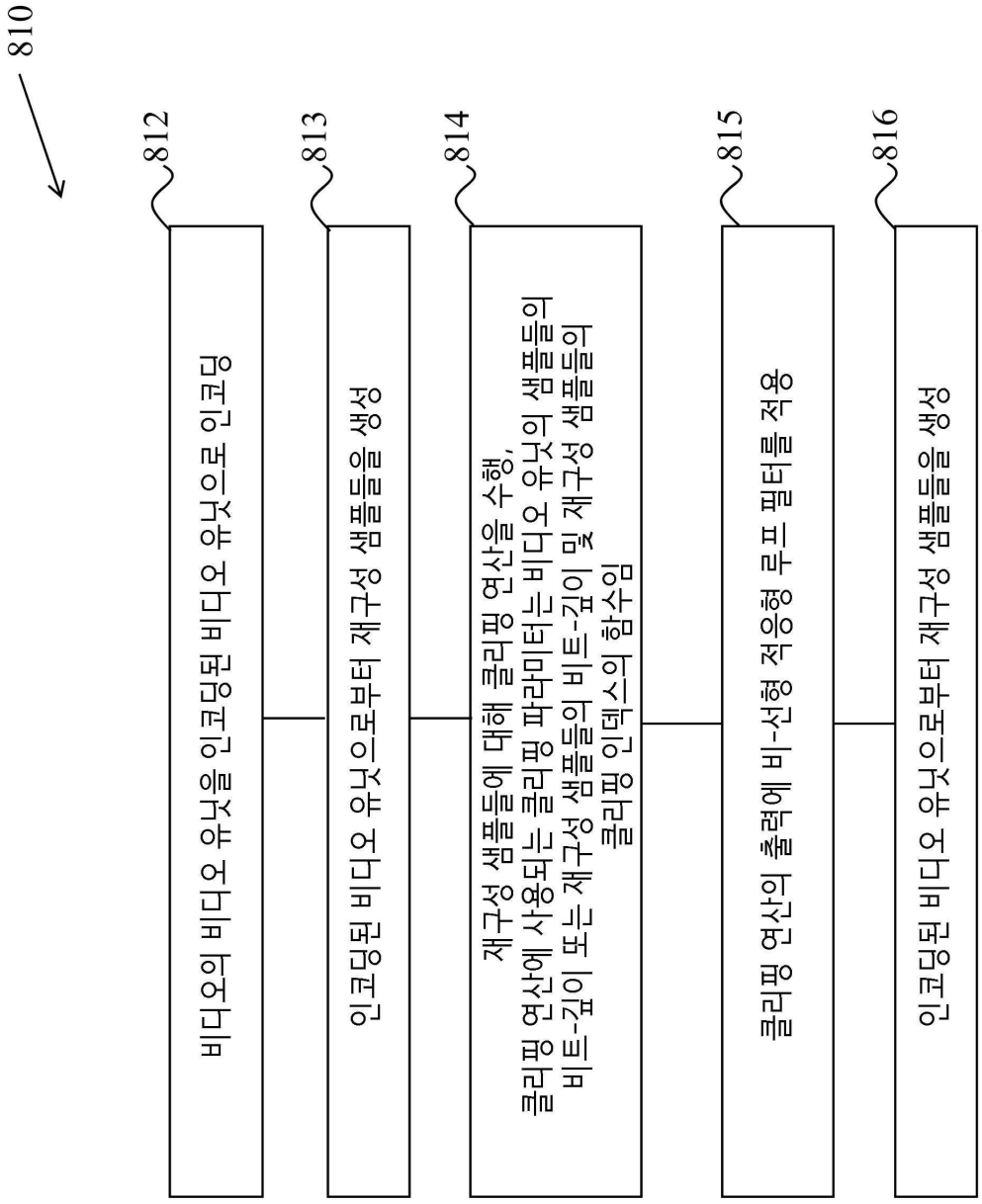
도면6



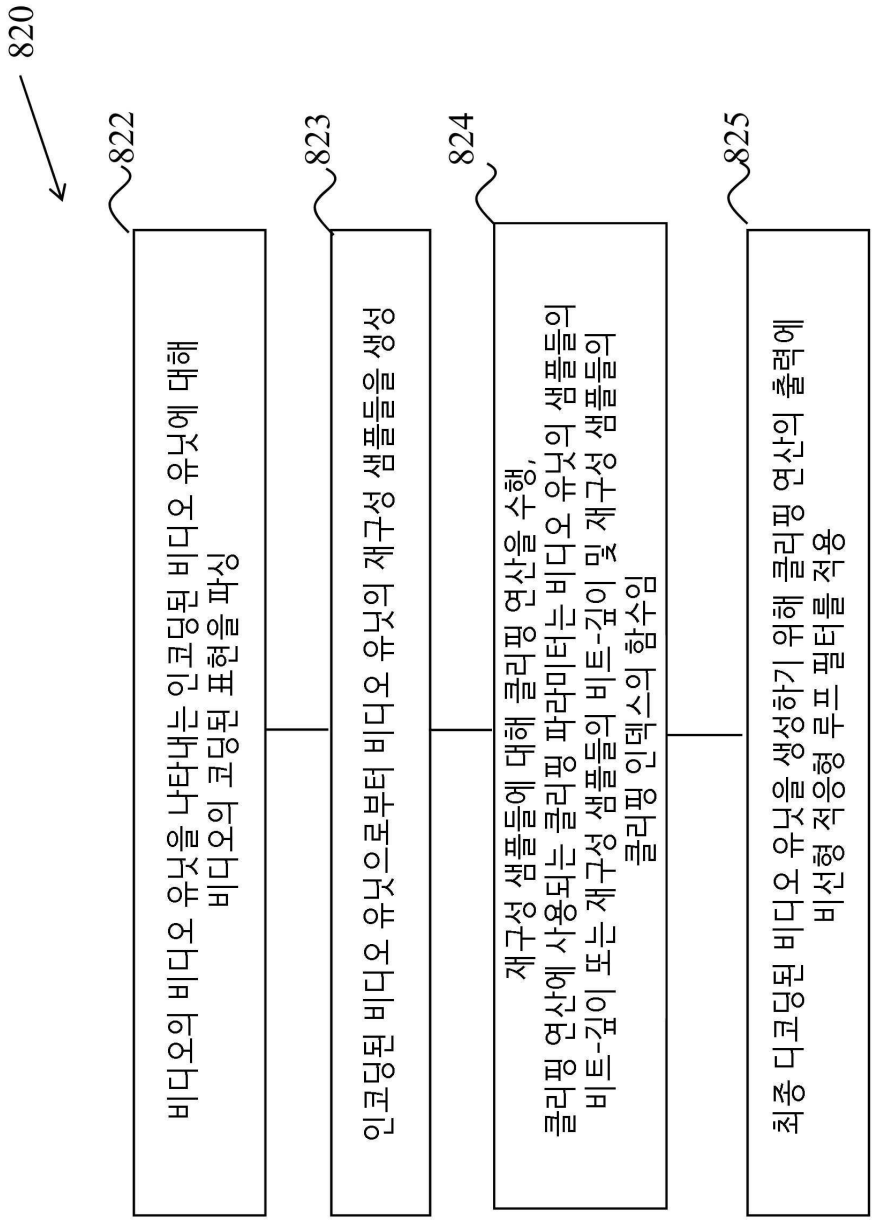
도면7


DPB
LF
역 변형:
$\hat{Y}_I$=InvLUT[$Y_r$]
인터 모드
인트라 모드
인트라 예측
모션 보상
$Y_{pred}$
$Y_{pred}$
재구성
$Y_r = Y_{res} + Y'_{pred}$
$Y'_{pred} = Y_{pred}$
$Y'_{pred}$
순방향 변형:
$Y'_{pred}$ =FwdLUT[$Y_{pred}$]
$Y_{res}$
CABAC$^{-1}$,
Q$^{-1}$ & T$^{-1}$

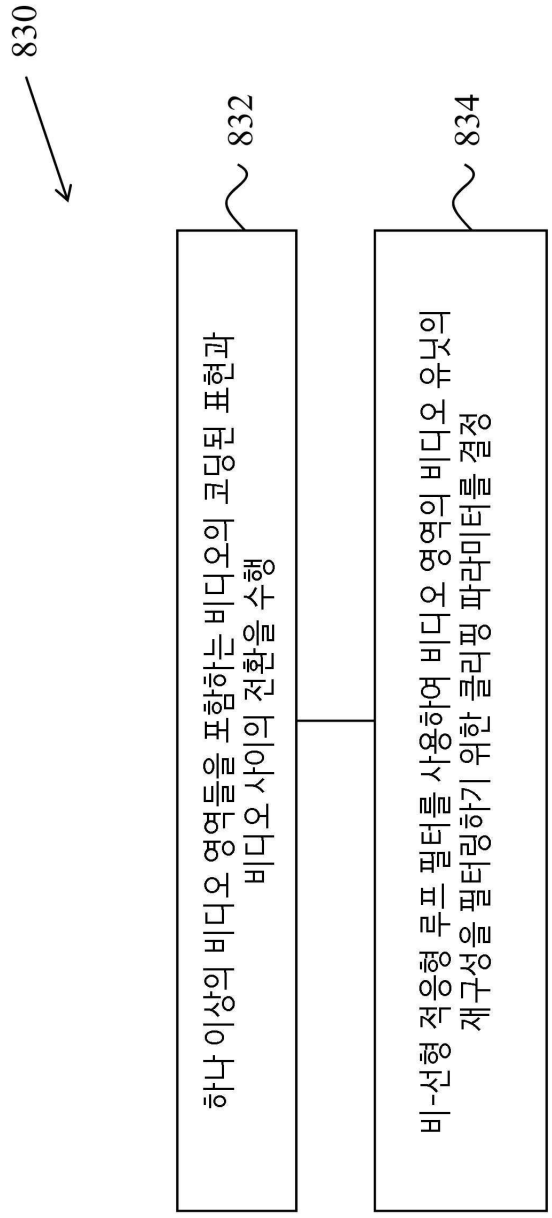
도면8a


810
812
비디오의 비디오 유닛을 인코딩된 비디오 유닛으로 인코딩
813
인코딩된 비디오 유닛으로부터 재구성 샘플들을 생성
814
재구성 샘플들에 대해 클리핑 연산을 수행,
클리핑 연산에 사용되는 클리핑 파라미터는 비디오 유닛의 샘플들의
비트-깊이 또는 재구성 샘플들의 비트-깊이 및 재구성 샘플들의
클리핑 인덱스의 함수임
815
클리핑 연산의 출력에 비-선형 적응형 루프 필터를 적용
816
인코딩된 비디오 유닛으로부터 재구성 샘플들을 생성

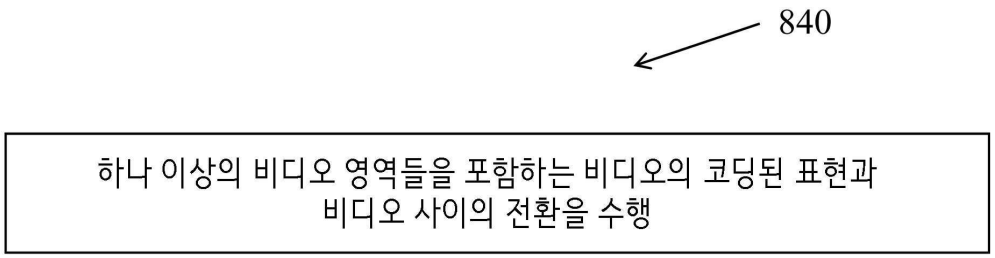
도면8b


820
822
비디오의 비디오 유닛을 나타내는 인코딩된 비디오 유닛에 대해
비디오의 코딩된 표현을 파싱
823
인코딩된 비디오 유닛으로부터 비디오 유닛의 재구성 샘플들을 생성
824
재구성 샘플들에 대해 클리핑 연산을 수행,
클리핑 연산에 사용되는 클리핑 파라미터는 비디오 유닛의 샘플들의
비트-깊이 또는 재구성 샘플들의 비트-깊이 및 재구성 샘플들의
클리핑 인덱스의 함수임
825
최종 디코딩된 비디오 유닛을 생성하기 위해 클리핑 연산의 출력에
비선형 적응형 루프 필터를 적용

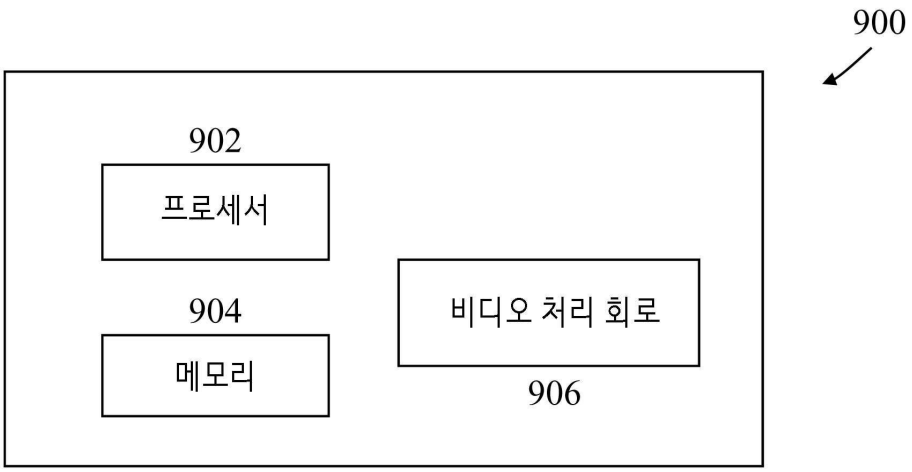
도면8c


830
832
834
하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 사이의 전환을 수행
비-선형 적응형 루프 필터를 사용하여 비디오 영역의 비디오 유닛의 재구성을 필터링하기 위한 클리핑 파라미터를 결정

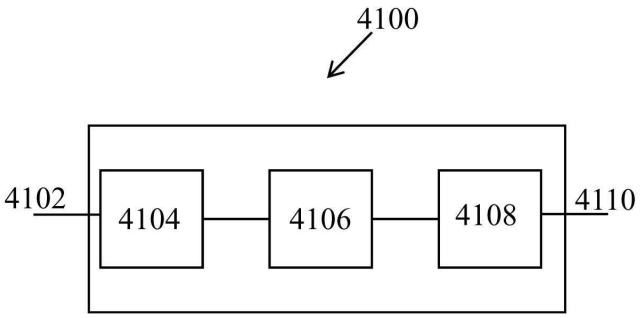
도면9


840
하나 이상의 비디오 영역들을 포함하는 비디오의 코딩된 표현과 비디오 사이의 전환을 수행

도면10a


900
902
프로세서
904
메모리
비디오 처리 회로
906

도면10b


4100
4102
4104
4106
4108
4110