



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2019-0059280  
(43) 공개일자 2019년05월30일

- (51) 국제특허분류(Int. Cl.)  
*H04N 19/117* (2014.01) *H04N 19/132* (2014.01)  
*H04N 19/137* (2014.01) *H04N 19/14* (2014.01)  
*H04N 19/147* (2014.01) *H04N 19/159* (2014.01)  
*H04N 19/174* (2014.01) *H04N 19/176* (2014.01)  
*H04N 19/182* (2014.01) *H04N 19/44* (2014.01)  
*H04N 19/70* (2014.01)
- (52) CPC특허분류  
*H04N 19/117* (2015.01)  
*H04N 19/132* (2015.01)
- (21) 출원번호 10-2019-7009394
- (22) 출원일자(국제) 2017년10월04일  
심사청구일자 없음
- (85) 번역문제출일자 2019년04월01일
- (86) 국제출원번호 PCT/US2017/055173
- (87) 국제공개번호 WO 2018/067722  
국제공개일자 2018년04월12일
- (30) 우선권주장  
62/404,126 2016년10월04일 미국(US)  
15/724,112 2017년10월03일 미국(US)

- (71) 출원인  
퀄컴 인코포레이티드  
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
- (72) 별명자  
카르체비츠 마르타  
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775  
장 리  
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775  
(뒷면에 계속)
- (74) 대리인  
특허법인코리아나

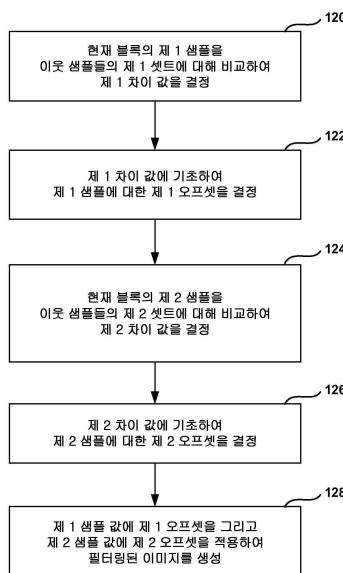
전체 청구항 수 : 총 36 항

(54) 발명의 명칭 **피크 샘플 적응형 오프셋**

**(57) 요 약**

비디오 코더는, 현재 블록의 제 1 샘플을 이웃 샘플들의 제 1 셋트에 대해 비교하여 제 1 차이 값을 결정하고; 그 제 1 차이 값에 기초하여 제 1 샘플에 대한 제 1 오프셋을 결정하며; 현재 블록의 제 2 샘플을 이웃 샘플들의 제 2 셋트에 대해 비교하여 제 1 차이 값과 동일한 부호를 가지는 제 2 차이 값을 결정하며; 그 제 2 차이 값에 기초하여, 제 1 샘플에 대한 오프셋과는 상이한, 제 2 샘플에 대한 오프셋을 결정하고; 그리고, 제 1 오프셋을 제 1 샘플 값에 그리고 제 2 오프셋을 제 2 샘플 값에 적용하여 필터링된 이미지를 생성한다.

**대 표 도** - 도10



(52) CPC특허분류

*HO4N 19/137* (2015.01)

*HO4N 19/14* (2015.01)

*HO4N 19/147* (2015.01)

*HO4N 19/159* (2015.01)

*HO4N 19/174* (2015.01)

*HO4N 19/176* (2015.01)

*HO4N 19/182* (2015.01)

*HO4N 19/44* (2015.01)

*HO4N 19/70* (2015.01)

(72) 발명자

**천 지안례**

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

**치엔 웨이-정**

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

---

**천 이-웬**

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

**자오 신**

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

## 명세서

### 청구범위

#### 청구항 1

비디오 데이터를 디코딩하는 방법으로서,

현재 블록의 제 1 샘플을 이웃 샘플들의 제 1 셋트에 대해 비교하여 제 1 차이 값을 결정하는 단계;

상기 제 1 차이 값에 기초하여 상기 제 1 샘플에 대한 제 1 오프셋을 결정하는 단계로서, 상기 제 1 오프셋은 제 1 비-제로 값인, 상기 제 1 샘플에 대한 제 1 오프셋을 결정하는 단계;

상기 현재 블록의 제 2 샘플을 이웃 샘플들의 제 2 셋트에 대해 비교하여 제 2 차이 값을 결정하는 단계로서, 상기 이웃 샘플들의 제 1 셋트는 상기 이웃 샘플들의 제 2 셋트가 상기 제 2 샘플에 대해 가지는 것과 동일한 상기 제 1 샘플에 대한 패턴을 가지며, 상기 제 1 차이 값 및 상기 제 2 차이 값은 동일한 부호를 가지는, 상기 제 2 차이 값을 결정하는 단계;

상기 제 2 차이 값에 기초하여 상기 제 2 샘플에 대한 제 2 오프셋을 결정하는 단계로서, 상기 제 2 오프셋은 상기 제 1 비-제로 값과는 상이한 제 2 비-제로 값인, 상기 제 2 샘플에 대한 제 2 오프셋을 결정하는 단계; 및

상기 제 1 오프셋을 제 1 샘플 값에 그리고 상기 제 2 오프셋을 제 2 샘플 값에 적용하여 필터링된 이미지를 생성하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

#### 청구항 2

제 1 항에 있어서,

상기 이웃 샘플들의 제 1 셋트 및 상기 이웃 샘플들의 제 2 셋트는 각각 2 개보다 많은 샘플들을 포함하는, 비디오 데이터를 디코딩하는 방법.

#### 청구항 3

제 1 항에 있어서,

상기 이웃 샘플들의 제 1 셋트 및 상기 이웃 샘플들의 제 2 셋트는 각각, 상기 현재 블록을 포함하는 현재 꼭처와는 상이한 꼭처에 있는 적어도 하나의 시간적 이웃하는 샘플을 포함하는, 비디오 데이터를 디코딩하는 방법.

#### 청구항 4

제 1 항에 있어서,

비디오 데이터의 상기 현재 블록에 대해, 이웃하는 샘플들의 로케이션들을 나타내는 신팩스 데이터를 수신하는 단계로서, 상기 웃하는 샘플들의 로케이션들은 상기 패턴을 정의하는, 상기 이웃하는 샘플들의 로케이션들을 나타내는 신팩스 데이터를 수신하는 단계; 및

상기 이웃하는 샘플들의 로케이션들을 나타내는 수신된 상기 신팩스 데이터에 기초하여 상기 이웃 샘플들의 제 1 셋트 및 상기 이웃 샘플들의 제 2 셋트를 결정하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

#### 청구항 5

제 1 항에 있어서,

비디오 데이터의 상기 현재 블록에 대해, 상기 현재 블록의 샘플들을 적어도 2 개의 그룹들로 그룹핑하기 위한 규칙들의 셋트를 나타내는 신팩스 데이터를 수신하는 단계로서, 제 1 그룹은 필터링될 샘플들을 포함하고 제 2 그룹은 필터링되지 않을 샘플들을 포함하는, 상기 규칙들의 셋트를 나타내는 신팩스 데이터를 수신하는 단계; 및

상기 제 1 그룹에 속하는 상기 제 1 샘플 및 상기 제 2 샘플을 결정하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

## 청구항 6

제 1 항에 있어서,

상기 현재 블록의 제 1 샘플을 이웃 샘플들의 제 1 셋트에 대해 비교하여 제 1 차이 값을 결정하는 단계는, 상기 현재 블록의 상기 제 1 샘플의 값을 상기 이웃 샘플들의 제 1 셋트의 평균 값에 대해 비교하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

## 청구항 7

제 1 항에 있어서,

상기 현재 블록의 제 1 샘플을 이웃 샘플들의 제 1 셋트에 대해 비교하여 제 1 차이 값을 결정하는 단계는, 상기 현재 블록의 상기 제 1 샘플의 값과 상기 이웃 샘플들의 제 1 셋트의 샘플 값들 사이의 값 차이들의 합계를 결정하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

## 청구항 8

제 1 항에 있어서,

상기 제 1 샘플의 값은 상기 이웃 샘플들의 제 1 셋트에서의 샘플들의 모든 값들보다 더 크고, 상기 제 2 샘플의 값은 상기 이웃 샘플들의 제 2 셋트에서의 샘플들의 모든 값들보다 더 큰, 비디오 데이터를 디코딩하는 방법.

## 청구항 9

제 1 항에 있어서,

상기 방법은 비디오 인코딩 프로세스의 일부로서 수행되는, 비디오 데이터를 디코딩하는 방법.

## 청구항 10

비디오 데이터를 디코딩하기 위한 디바이스로서,

상기 디바이스는,

상기 비디오 데이터를 저장하도록 구성된 메모리; 및

하나 이상의 프로세서들을 포함하고,

상기 하나 이상의 프로세서들은,

현재 블록의 제 1 샘플을 이웃 샘플들의 제 1 셋트에 대해 비교하여 제 1 차이 값을 결정하고;

상기 제 1 차이 값에 기초하여 상기 제 1 샘플에 대한 제 1 오프셋을 결정하는 것으로서, 상기 제 1 오프셋은 제 1 비-제로 값인, 상기 제 1 샘플에 대한 제 1 오프셋을 결정하는 것을 행하며;

상기 현재 블록의 제 2 샘플을 이웃 샘플들의 제 2 셋트에 대해 비교하여 제 2 차이 값을 결정하는 것으로서, 상기 이웃 샘플들의 제 1 셋트는 상기 이웃 샘플들의 제 2 셋트가 상기 제 2 샘플에 대해 가지는 것과 동일한 상기 제 1 샘플에 대한 패턴을 가지며, 상기 제 1 차이 값 및 상기 제 2 차이 값은 동일한 부호를 가지는, 상기 제 2 차이 값을 결정하는 것을 행하고;

상기 제 2 차이 값에 기초하여 상기 제 2 샘플에 대한 제 2 오프셋을 결정하는 것으로서, 상기 제 2 오프셋은 상기 제 1 비-제로 값과는 상이한 제 2 비-제로 값인, 상기 제 2 샘플에 대한 제 2 오프셋을 결정하는 것을 행하고; 그리고

상기 제 1 오프셋을 제 1 샘플 값에 그리고 상기 제 2 오프셋을 제 2 샘플 값에 적용하여 필터링된 이미지를 생성하도록

구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

## 청구항 11

제 10 항에 있어서,

상기 이웃 샘플들의 제 1 셋트 및 상기 이웃 샘플들의 제 2 셋트는 각각 2 개보다 많은 샘플들을 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

### 청구항 12

제 10 항에 있어서,

상기 이웃 샘플들의 제 1 셋트 및 상기 이웃 샘플들의 제 2 셋트는 각각, 상기 현재 블록을 포함하는 현재 블록과는 상이한 블록에 있는 적어도 하나의 시간적 이웃하는 샘플을 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

### 청구항 13

제 10 항에 있어서,

상기 하나 이상의 프로세서들은 추가적으로,

비디오 데이터의 상기 현재 블록에 대해, 이웃하는 샘플들의 로케이션들을 나타내는 신팩스 데이터를 수신하는 것으로서, 상기 웃하는 샘플들의 로케이션들은 상기 패턴을 정의하는, 상기 이웃하는 샘플들의 로케이션들을 나타내는 신팩스 데이터를 수신하는 것을 행하고; 그리고

상기 이웃하는 샘플들의 로케이션들을 나타내는 수신된 상기 신팩스 데이터에 기초하여 상기 이웃 샘플들의 제 1 셋트 및 상기 이웃 샘플들의 제 2 셋트를 결정하도록

구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

### 청구항 14

제 10 항에 있어서,

상기 하나 이상의 프로세서들은 추가적으로,

비디오 데이터의 상기 현재 블록에 대해, 상기 현재 블록의 샘플들을 적어도 2 개의 그룹들로 그룹핑하기 위한 규칙들의 셋트를 나타내는 신팩스 데이터를 수신하는 것으로서, 제 1 그룹은 필터링될 샘플들을 포함하고 제 2 그룹은 필터링되지 않을 샘플들을 포함하는, 상기 규칙들의 셋트를 나타내는 신팩스 데이터를 수신하는 것을 행하고; 그리고

상기 제 1 그룹에 속하는 상기 제 1 샘플 및 상기 제 2 샘플을 결정하도록

구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

### 청구항 15

제 10 항에 있어서,

상기 현재 블록의 제 1 샘플을 이웃 샘플들의 제 1 셋트에 대해 비교하여 제 1 차이 값을 결정하기 위해, 상기 하나 이상의 프로세서들은 추가적으로, 상기 현재 블록의 상기 제 1 샘플의 값을 상기 이웃 샘플들의 제 1 셋트의 평균 값에 대해 비교하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

### 청구항 16

제 10 항에 있어서,

상기 현재 블록의 제 1 샘플을 이웃 샘플들의 제 1 셋트에 대해 비교하여 제 1 차이 값을 결정하기 위해, 상기 하나 이상의 프로세서들은 추가적으로, 상기 현재 블록의 상기 제 1 샘플의 값과 상기 이웃 샘플들의 제 1 셋트의 샘플 값들 사이의 값 차이들의 합계를 결정하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

### 청구항 17

제 10 항에 있어서,

상기 제 1 샘플의 값은 상기 이웃 샘플들의 제 1 셋트에서의 샘플들의 모든 값들보다 더 크고, 상기 제 2 샘플의 값은 상기 이웃 샘플들의 제 2 셋트에서의 샘플들의 모든 값들보다 더 큰, 비디오 데이터를 디코딩하기 위한 디바이스.

#### 청구항 18

제 10 항에 있어서,

상기 디바이스는 비디오 인코딩 프로세스의 일부로서 비디오 데이터를 디코딩하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

#### 청구항 19

제 10 항에 있어서,

상기 디바이스는 무선 통신 디바이스를 포함하고,

인코딩된 비디오 데이터를 수신하도록 구성된 수신기를 더 포함하는, 비디오 데이터를 디코딩하기 위한 디바이스.

#### 청구항 20

제 19 항에 있어서,

상기 무선 통신 디바이스는 전화 핸드셋을 포함하고,

상기 수신기는, 상기 인코딩된 비디오 데이터를 포함하는 신호를 무선 통신 표준에 따라 복조하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

#### 청구항 21

명령들을 저장한 컴퓨터 판독가능 저장 매체로서,

상기 명령들은, 하나 이상의 프로세서들에 의해 실행될 때, 상기 하나 이상의 프로세서들로 하여금,

현재 블록의 제 1 샘플을 이웃 샘플들의 제 1 셋트에 대해 비교하여 제 1 차이 값을 결정하게 하고;

상기 제 1 차이 값에 기초하여 상기 제 1 샘플에 대한 제 1 오프셋을 결정하는 것으로서, 상기 제 1 오프셋은 제 1 비-제로 값인, 상기 제 1 샘플에 대한 제 1 오프셋을 결정하는 것을 행하게 하며;

상기 현재 블록의 제 2 샘플을 이웃 샘플들의 제 2 셋트에 대해 비교하여 제 2 차이 값을 결정하는 것으로서, 상기 이웃 샘플들의 제 1 셋트는 상기 이웃 샘플들의 제 2 셋트가 상기 제 2 샘플에 대해 가지는 것과 동일한 상기 제 1 샘플에 대한 패턴을 가지며, 상기 제 1 차이 값 및 상기 제 2 차이 값은 동일한 부호를 가지는, 상기 제 2 차이 값을 결정하는 것을 행하게 하고;

상기 제 2 차이 값에 기초하여 상기 제 2 샘플에 대한 제 2 오프셋을 결정하는 것으로서, 상기 제 2 오프셋은 상기 제 1 비-제로 값과는 상이한 제 2 비-제로 값인, 상기 제 2 샘플에 대한 제 2 오프셋을 결정하는 것을 행하게 하며; 그리고

상기 제 1 오프셋을 제 1 샘플 값에 그리고 상기 제 2 오프셋을 제 2 샘플 값에 적용하여 필터링된 이미지를 생성하게 하는, 컴퓨터 판독가능 저장 매체.

#### 청구항 22

제 21 항에 있어서,

상기 이웃 샘플들의 제 1 셋트 및 상기 이웃 샘플들의 제 2 셋트는 각각 2 개보다 많은 샘플들을 포함하는, 컴퓨터 판독가능 저장 매체.

#### 청구항 23

제 21 항에 있어서,

상기 이웃 샘플들의 제 1 셋트 및 상기 이웃 샘플들의 제 2 셋트는 각각, 상기 현재 블록을 포함하는 현재 블록에 있는 적어도 하나의 시간적 이웃하는 샘플을 포함하는, 컴퓨터 판독가능 저장 매체.

#### 청구항 24

제 21 항에 있어서,

실행될 때 상기 하나 이상의 프로세서들로 하여금,

비디오 데이터의 상기 현재 블록에 대해, 이웃하는 샘플들의 로케이션들을 나타내는 선택스 데이터를 수신하는 것으로서, 상기 웃하는 샘플들의 로케이션들은 상기 패턴을 정의하는, 상기 이웃하는 샘플들의 로케이션들을 나타내는 선택스 데이터를 수신하는 것을 행하게 하고; 그리고

상기 이웃하는 샘플들의 로케이션들을 나타내는 수신된 상기 선택스 데이터에 기초하여 상기 이웃 샘플들의 제 1 셋트 및 상기 이웃 샘플들의 제 2 셋트를 결정하게 하는

명령들을 추가로 저장한, 컴퓨터 판독가능 저장 매체.

#### 청구항 25

제 21 항에 있어서,

실행될 때 상기 하나 이상의 프로세서들로 하여금,

비디오 데이터의 상기 현재 블록에 대해, 상기 현재 블록의 샘플들을 적어도 2 개의 그룹들로 그룹핑하기 위한 규칙들의 셋트를 나타내는 선택스 데이터를 수신하는 것으로서, 제 1 그룹은 필터링될 샘플들을 포함하고 제 2 그룹은 필터링되지 않을 샘플들을 포함하는, 상기 규칙들의 셋트를 나타내는 선택스 데이터를 수신하는 것을 행하고; 그리고

상기 제 1 그룹에 속하는 상기 제 1 샘플 및 상기 제 2 샘플을 결정하게 하는

명령들을 추가로 저장한, 컴퓨터 판독가능 저장 매체.

#### 청구항 26

제 21 항에 있어서,

상기 현재 블록의 제 1 샘플을 이웃 샘플들의 제 1 셋트에 대해 비교하여 제 1 차이 값을 결정하기 위해, 상기 명령들은 상기 하나 이상의 프로세서들로 하여금, 상기 현재 블록의 상기 제 1 샘플의 값을 상기 이웃 샘플들의 제 1 셋트의 평균 값에 대해 비교하게 하는, 컴퓨터 판독가능 저장 매체.

#### 청구항 27

제 21 항에 있어서,

상기 현재 블록의 제 1 샘플을 이웃 샘플들의 제 1 셋트에 대해 비교하여 제 1 차이 값을 결정하기 위해, 상기 명령들은 상기 하나 이상의 프로세서들로 하여금, 상기 현재 블록의 상기 제 1 샘플의 값과 상기 이웃 샘플들의 제 1 셋트의 샘플 값들 사이의 값 차이들의 합계를 결정하게 하는, 컴퓨터 판독가능 저장 매체.

#### 청구항 28

제 21 항에 있어서,

상기 제 1 샘플의 값은 상기 이웃 샘플들의 제 1 셋트에서의 샘플들의 모든 값들보다 더 크고, 상기 제 2 샘플의 값은 상기 이웃 샘플들의 제 2 셋트에서의 샘플들의 모든 값들보다 더 큼, 컴퓨터 판독가능 저장 매체.

#### 청구항 29

비디오 데이터를 디코딩하기 위한 장치로서,

현재 블록의 제 1 샘플을 이웃 샘플들의 제 1 셋트에 대해 비교하여 제 1 차이 값을 결정하는 수단;

상기 제 1 차이 값을 기초하여 상기 제 1 샘플에 대한 제 1 오프셋을 결정하는 수단으로서, 상기 제 1 오프셋은

제 1 비-제로 값인, 상기 제 1 샘플에 대한 제 1 오프셋을 결정하는 수단;

상기 현재 블록의 제 2 샘플을 이웃 샘플들의 제 2 셋트에 대해 비교하여 제 2 차이 값을 결정하는 수단으로서, 상기 이웃 샘플들의 제 1 셋트는 상기 이웃 샘플들의 제 2 셋트가 상기 제 2 샘플에 대해 가지는 것과 동일한 상기 제 1 샘플에 대한 패턴을 가지며, 상기 제 1 차이 값 및 상기 제 2 차이 값은 동일한 부호를 가지는, 상기 제 2 차이 값을 결정하는 수단;

상기 제 2 차이 값에 기초하여 상기 제 2 샘플에 대한 제 2 오프셋을 결정하는 수단으로서, 상기 제 2 오프셋은 상기 제 1 비-제로 값과는 상이한 제 2 비-제로 값인, 상기 제 2 샘플에 대한 제 2 오프셋을 결정하는 수단; 및 상기 제 1 오프셋을 제 1 샘플 값에 그리고 상기 제 2 오프셋을 제 2 샘플 값에 적용하여 필터링된 이미지를 생성하는 수단을 포함하는, 비디오 데이터를 디코딩하기 위한 장치.

### 청구항 30

제 29 항에 있어서,

상기 이웃 샘플들의 제 1 셋트 및 상기 이웃 샘플들의 제 2 셋트는 각각 2 개보다 많은 샘플들을 포함하는, 비디오 데이터를 디코딩하기 위한 장치.

### 청구항 31

제 29 항에 있어서,

상기 이웃 샘플들의 제 1 셋트 및 상기 이웃 샘플들의 제 2 셋트는 각각, 상기 현재 블록을 포함하는 현재 뷍처와는 상이한 뷍처에 있는 적어도 하나의 시간적 이웃하는 샘플을 포함하는, 비디오 데이터를 디코딩하기 위한 장치.

### 청구항 32

제 29 항에 있어서,

비디오 데이터의 상기 현재 블록에 대해, 이웃하는 샘플들의 로케이션들을 나타내는 신팩스 데이터를 수신하는 수단으로서, 상기 웃하는 샘플들의 로케이션들은 상기 패턴을 정의하는, 상기 이웃하는 샘플들의 로케이션들을 나타내는 신팩스 데이터를 수신하는 수단; 및

상기 이웃하는 샘플들의 로케이션들을 나타내는 수신된 상기 신팩스 데이터에 기초하여 상기 이웃 샘플들의 제 1 셋트 및 상기 이웃 샘플들의 제 2 셋트를 결정하는 수단을 더 포함하는, 비디오 데이터를 디코딩하기 위한 장치.

### 청구항 33

제 29 항에 있어서,

비디오 데이터의 상기 현재 블록에 대해, 상기 현재 블록의 샘플들을 적어도 2 개의 그룹들로 그룹핑하기 위한 규칙들의 셋트를 나타내는 신팩스 데이터를 수신하는 수단으로서, 제 1 그룹은 필터링될 샘플들을 포함하고 제 2 그룹은 필터링되지 않을 샘플들을 포함하는, 상기 규칙들의 셋트를 나타내는 신팩스 데이터를 수신하는 수단; 및

상기 제 1 그룹에 속하는 상기 제 1 샘플 및 상기 제 2 샘플을 결정하는 수단을 더 포함하는, 비디오 데이터를 디코딩하기 위한 장치.

### 청구항 34

제 29 항에 있어서,

상기 현재 블록의 제 1 샘플을 이웃 샘플들의 제 1 셋트에 대해 비교하여 제 1 차이 값을 결정하는 수단은, 상기 현재 블록의 상기 제 1 샘플의 값을 상기 이웃 샘플들의 제 1 셋트의 평균 값에 대해 비교하는 수단을 포함하는, 비디오 데이터를 디코딩하기 위한 장치.

### 청구항 35

제 29 항에 있어서,

상기 현재 블록의 제 1 샘플을 이웃 샘플들의 제 1 셋트에 대해 비교하여 제 1 차이 값을 결정하는 수단은, 상기 현재 블록의 상기 제 1 샘플의 값과 상기 이웃 샘플들의 제 1 셋트의 샘플 값들 사이의 값 차이들의 합계를 결정하는 수단을 포함하는, 비디오 데이터를 디코딩하기 위한 장치.

### 청구항 36

제 29 항에 있어서,

상기 제 1 샘플의 값은 상기 이웃 샘플들의 제 1 셋트에서의 샘플들의 모든 값들보다 더 크고, 상기 제 2 샘플의 값은 상기 이웃 샘플들의 제 2 셋트에서의 샘플들의 모든 값들보다 더 큰, 비디오 데이터를 디코딩하기 위한 장치.

### 발명의 설명

#### 기술 분야

[0001] 본 출원은 2016년 10월 4일 출원된 미국 가 특허 출원 제 62/404,126 호의 이익을 주장하며, 그 전체 내용은 참조에 의해 본원에 통합된다.

#### 기술 분야

[0003] 본 개시는 비디오 인코딩 및 비디오 디코딩에 관한 것이다.

### 배경 기술

#### 배경

[0005] 디지털 비디오 능력들은 디지털 텔레비전들, 디지털 직접 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인용 디지털 보조기들 (PDA들), 랩탑 또는 데스크탑 컴퓨터들, 태블릿 컴퓨터들, e-북 리더들, 디지털 카메라들, 디지털 레코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 무선 전화기들, 소위 "스마트 폰들", 비디오 텔레컨퍼런싱 디바이스들, 비디오 스트리밍 디바이스들 등을 포함한, 광범위한 디바이스들에 통합될 수 있다. 디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, 파트 10, 어드밴스드 비디오 코딩 (AVC)에 의해 정의된 표준들, 최근에 마무리된 고 효율 비디오 코딩 (HEVC) 표준, 및 그러한 표준들의 확장들에서 설명된 기술들과 같은 비디오 압축 기술들을 구현한다. 비디오 디바이스들은 그러한 비디오 압축 기법들을 구현함으로써 디지털 비디오 정보를 더 효율적으로 송신, 수신, 인코딩, 디코딩, 및/또는 저장할 수도 있다.

[0006] 비디오 압축 기법들은 비디오 시퀀스들에 내재한 리던던시를 감소 또는 제거하기 위해 공간 (인트라-픽처) 예측 및/또는 시간 (인터-픽처) 예측을 수행한다. 블록 기반 비디오 코딩에 대해, 비디오 슬라이스 (즉, 비디오 프레임 또는 비디오 프레임의 일부)는 비디오 블록들로 파티셔닝될 수도 있으며, 이 비디오 블록들은 또한 트리블록들, 코딩 유닛들 (CU들) 및/또는 코딩 노드들로서 지칭될 수도 있다. 픽처의 인트라-코딩된 (I) 슬라이스에서의 비디오 블록들은 동일 픽처의 이웃 블록들에서의 참조 샘플들에 대한 공간 예측을 이용하여 인코딩된다. 픽처의 인터-코딩된 (P 또는 B) 슬라이스에서의 비디오 블록들은 동일 픽처의 이웃 블록들에서의 참조 샘플들에 대한 공간 예측, 또는 다른 참조 픽처들에서의 참조 샘플들에 대한 시간 예측을 이용할 수도 있다. 픽처들은 프레임들로서 지칭될 수도 있으며, 참조 픽처들은 참조 프레임들로 지칭될 수도 있다.

[0007] 공간 또는 시간 예측은 코딩될 블록에 대한 예측 블록을 발생시킨다. 잔차 데이터는 코딩될 원래의 블록과 예측 블록 간의 픽셀 차이들을 나타낸다. 인터-코딩된 블록은 예측 블록을 형성하는 참조 샘플들의 블록을 포인팅하는 모션 벡터에 따라 인코딩되고, 잔차 데이터는 코딩된 블록과 예측 블록 간의 차이를 나타낸다. 인트라-코딩된 블록은 인트라-코딩 모드 및 잔여 데이터에 따라 인코딩된다. 추가의 압축을 위하여, 잔차 데이터는 픽셀 도메인으로부터 변환 도메인으로 변환되어 잔차 변환 계수들을 초래할 수도 있고, 그 후 이들은 양자화될 수도 있다. 2차원 어레이로 초기에 배열되는 양자화된 변환 계수들은 변환 계수들의 1차원 벡터를 생성하기 위해 스캐닝될 수도 있으며, 훨씬 더 많은 압축을 달성하도록 엔트로피 코딩이 적용될 수도 있다.

### 발명의 내용

- [0008] 이 개시물은 피크 SAO로서 본 명세서에서 지칭되는, 샘플 적응형 오프셋 (sample adaptive offset; SAO) 필터링의 일 타입을 기술한다. 이 개시물에서 기술된 바와 같이 피크 SAO를 구현하는 비디오 코더는 샘플의 값을 2 개보다 많은 이웃하는 샘플들 (neighboring samples)에 대해 비교함으로써 샘플에 대한 오프셋을 결정할 수도 있다. 일부 예들에서, 피크 SAO를 구현하는 비디오 코더는 샘플들과 각각의 이웃 샘플들 (neighbor samples) 사이의 차이의 크기에 기초하여 블록의 상이한 샘플들의 상이한 오프셋들을 결정할 수도 있다.
- [0009] 하나의 예에서, 비디오 데이터를 디코딩하는 방법은, 현재 블록의 제 1 샘플을 이웃 샘플들의 제 1 셋트에 대해 비교하여 제 1 차이 값을 결정하는 단계; 그 제 1 차이 값에 기초하여 제 1 샘플에 대한 제 1 오프셋을 결정하는 단계로서, 상기 제 1 오프셋은 제 1 비-제로 값인, 상기 제 1 샘플에 대한 제 1 오프셋을 결정하는 단계; 현재 블록의 제 2 샘플을 이웃 샘플들의 제 2 셋트에 대해 비교하여 제 2 차이 값을 결정하는 단계로서, 상기 이웃 샘플들의 제 1 셋트는 이웃 샘플들의 제 2 셋트가 제 2 샘플에 대해 가지는 것과 동일한 제 1 샘플에 대한 패턴을 가지며, 제 1 차이 값 및 제 2 차이 값은 동일한 부호를 가지는, 상기 제 2 차이 값을 결정하는 단계; 제 2 차이 값에 기초하여 제 2 샘플에 대한 제 2 오프셋을 결정하는 단계로서, 제 2 오프셋은 제 1 비-제로 값과는 상이한 제 2 비-제로 값인, 상기 제 2 샘플에 대한 제 2 오프셋을 결정하는 단계; 및, 제 1 오프셋을 제 1 샘플 값에 그리고 제 2 오프셋을 제 2 샘플 값에 적용하여 필터링된 이미지를 생성하는 단계를 포함한다.
- [0010] 다른 예에서, 비디오 데이터를 디코딩하기 위한 디바이스는, 비디오 데이터를 저장하도록 구성된 메모리 및 하나 이상의 프로세서들을 포함하고, 상기 하나 이상의 프로세서들은, 현재 블록의 제 1 샘플을 이웃 샘플들의 제 1 셋트에 대해 비교하여 제 1 차이 값을 결정하고; 그 제 1 차이 값에 기초하여 제 1 샘플에 대한 제 1 오프셋을 결정하는 것으로서, 제 1 오프셋은 제 1 비-제로 값인, 상기 제 1 샘플에 대한 제 1 오프셋을 결정하는 것을 행하며; 현재 블록의 제 2 샘플을 이웃 샘플들의 제 2 셋트에 대해 비교하여 제 2 차이 값을 결정하는 것으로서, 이웃 샘플들의 제 1 셋트는 이웃 샘플들의 제 2 셋트가 제 2 샘플에 대해 가지는 것과 동일한 제 1 샘플에 대한 패턴을 가지며, 제 1 차이 값 및 제 2 차이 값은 동일한 부호를 가지는, 상기 제 2 차이 값을 결정하는 것을 행하고; 제 2 차이 값에 기초하여 제 2 샘플에 대한 제 2 오프셋을 결정하는 것으로서, 제 2 오프셋은 제 1 비-제로 값과는 상이한 제 2 비-제로 값인, 상기 제 2 샘플에 대한 제 2 오프셋을 결정하는 것을 행하고; 그리고, 제 1 오프셋을 제 1 샘플 값에 그리고 제 2 오프셋을 제 2 샘플 값에 적용하여 필터링된 이미지를 생성하도록 구성된다.
- [0011] 또 다른 예에서, 컴퓨터 판독가능 매체는, 하나 이상의 프로세서들에 의해 실행될 때 그 하나 이상의 프로세서들로 하여금, 현재 블록의 제 1 샘플을 이웃 샘플들의 제 1 셋트에 대해 비교하여 제 1 차이 값을 결정하게 하고; 그 제 1 차이 값에 기초하여 제 1 샘플에 대한 제 1 오프셋을 결정하는 것으로서, 제 1 오프셋은 제 1 비-제로 값인, 상기 제 1 샘플에 대한 제 1 오프셋을 결정하는 것을 행하게 하며; 현재 블록의 제 2 샘플을 이웃 샘플들의 제 2 셋트에 대해 비교하여 제 2 차이 값을 결정하는 것으로서, 이웃 샘플들의 제 1 셋트는 이웃 샘플들의 제 2 셋트가 제 2 샘플에 대해 가지는 것과 동일한 제 1 샘플에 대한 패턴을 가지며, 제 1 차이 값 및 제 2 차이 값은 동일한 부호를 가지는, 상기 제 2 차이 값을 결정하는 것을 행하게 하고; 제 2 차이 값에 기초하여 제 2 샘플에 대한 제 2 오프셋을 결정하는 것으로서, 제 2 오프셋은 제 1 비-제로 값과는 상이한 제 2 비-제로 값인, 상기 제 2 샘플에 대한 제 2 오프셋을 결정하는 것을 행하게 하며; 그리고, 제 1 오프셋을 제 1 샘플 값에 그리고 제 2 오프셋을 제 2 샘플 값에 적용하여 필터링된 이미지를 생성하게 하는, 명령들을 저장한다.
- [0012] 또 다른 예에서, 비디오 데이터를 디코딩하기 위한 장치는, 현재 블록의 제 1 샘플을 이웃 샘플들의 제 1 셋트에 대해 비교하여 제 1 차이 값을 결정하는 수단; 그 제 1 차이 값에 기초하여 제 1 샘플에 대한 제 1 오프셋을 결정하는 수단으로서, 제 1 오프셋은 제 1 비-제로 값인, 상기 제 1 샘플에 대한 제 1 오프셋을 결정하는 수단; 현재 블록의 제 2 샘플을 이웃 샘플들의 제 2 셋트에 대해 비교하여 제 2 차이 값을 결정하는 수단으로서, 이웃 샘플들의 제 1 셋트는 이웃 샘플들의 제 2 셋트가 제 2 샘플에 대해 가지는 것과 동일한 제 1 샘플에 대한 패턴을 가지며, 제 1 차이 값 및 제 2 차이 값은 동일한 부호를 가지는, 상기 제 2 차이 값을 결정하는 수단; 제 2 차이 값에 기초하여 제 2 샘플에 대한 제 2 오프셋을 결정하는 수단으로서, 제 2 오프셋은 제 1 비-제로 값과는 상이한 제 2 비-제로 값인, 상기 제 2 샘플에 대한 제 2 오프셋을 결정하는 수단; 및, 제 1 오프셋을 제 1 샘플 값에 그리고 제 2 오프셋을 제 2 샘플 값에 적용하여 필터링된 이미지를 생성하는 수단을 포함한다.
- [0013] 본 개시의 하나 이상의 양태들의 상세들은 첨부 도면들 및 이하의 설명에서 전개된다. 이 개시물에서 기술된 기법들의 다른 특징들, 목적들, 및 이점들은 상세한 설명 및 도면들로부터, 그리고 청구항들로부터 명백해질 것이다.

## 도면의 간단한 설명

[0014] 도 1 은 이 개시물에서 기술된 기법들을 이용할 수도 있는 일 예시적인 비디오 인코딩 및 디코딩 시스템을 나타내는 블록도이다.

도 2 는 HEVC 디코더의 일 예시적인 블록도를 나타낸다.

도 3a 내지 도 3d 는 에지 오프셋 (EO) 샘플 분류를 위한 4 개의 1-D 방향 패턴들을 나타낸다.

도 4 는 현재 픽셀 및 그 현재 픽셀의 이웃하는 픽셀들의 일 예를 나타낸다.

도 5 는 현재 픽셀 및 그 현재 픽셀의 이웃하는 픽셀들의 일 예를 나타낸다.

도 6a 및 도 6b 는 8-선택된 이웃하는 샘플들을 갖는 2 개의 샘플들의 예들을 나타낸다.

도 7 은 이 개시물에서 기술된 기법들을 구현할 수도 있는 일 예시적인 비디오 인코더를 나타내는 블록도이다.

도 8 은 이 개시물에서 기술된 기법들을 구현할 수도 있는 일 예시적인 비디오 디코더를 나타내는 블록도이다.

도 9 는 이 개시물의 기법들을 수행하기 위한 필터 유닛의 일 예시적인 구현을 나타낸다.

도 10 은 이 개시물의 기법들에 따른 일 예시적인 비디오 디코딩 프로세스를 나타내는 흐름도이다.

### 발명을 실시하기 위한 구체적인 내용

[0015] 이 개시물은 인-루프 코딩의 일부로서 포스트-프로세싱 스테이지에서, 또는 예측 스테이지에서 사용될 수 있을 필터링 방법에 관련된 기법들을 기술한다. 이 개시물의 기법들은 HEVC (High Efficiency Video Coding) 과 같은 기준의 비디오 코덱들 내로 구현될 수도 있거나 현재 개발 중인 H.266 표준과 같은 미래의 비디오 코딩 표준을 위한 효율적인 코딩 틀일 수도 있다.

[0016] 샘플 적응형 오프셋 (SAO) 필터링은 비디오 코딩에서 사용되는 루프 필터링의 일 타입이다. 일반적으로, 비디오 프레임 (예컨대, 재구성된 이미지)에서의 샘플들에 대한 오프셋 값들의 부가는 일부 경우들에서, 인코딩된 비디오 데이터를 저장 또는 송신하기 위해 필요한 비트 오버헤드를 크게 증가시킬 수 없이 코딩을 향상시킬 수도 있다. 잠재적으로 SAO 필터링으로부터 초래되는 코딩에서의 향상은, 예를 들어, 비디오 데이터를 인코딩할 때 보다 양호한 레이트-왜곡 (rate-distortion; RD) 트레이드오프가 달성될 수도 있다는 것, 및/또는, 디코딩된 비디오 이미지가 주어진 비트 레이트에 대해 원래의 비디오 이미지에 비해 더 적은 왜곡을 포함한다는 것일 수도 있다. SAO 기법들은 에지 메트릭들, 밴드 메트릭들, 또는 다른 타입들의 메트릭들과 같은 샘플 (또는 블록) 분류 메트릭들에 따라 상이한 샘플들 (또는 샘플들의 블록들)에 상이한 오프셋 값들이 적용되도록 허용한다.

[0017] 일부 구현들에서, SAO 필터 유닛은, 일반적으로 이 개시물에서 밴드 오프셋 (band offset; BO) 필터링 및 에지 오프셋 (edge offset; EO) 필터링으로서 지칭되는, 2 가지 타입들의 SAO 필터링을 수행하도록 구성될 수도 있다. SAO 필터는 또한, 때때로, 그 자체가 제 3 타입의 SAO 필터링으로 고려될 수 있는 오프셋 없음을 적용할 수도 있다. SAO 필터에 의해 적용되는 오프셋 필터링의 타입은 비디오 디코더에 명시적으로 또는 암시적으로 시그널링될 수도 있다. EO 필터링을 적용할 때, 샘플들은 코딩 트리 유닛 (CTU)의 에지 정보에 기초하여 분류될 수 있고, 오프셋은 에지 분류에 기초하여 샘플들에 대해 결정될 수 있다. 이하에서 보다 자세히 설명되는 바와 같이, 통상적으로 에지-기반 SAO 의 4 가지 변형들이 존재하고, 샘플의 값은 샘플의 8 개의 이웃하는 샘플들 중 2 개에 대해 비교된다. 비교를 위해 어느 2 개의 샘플들이 사용되는지는 에지-기반 오프셋의 어느 변형이 사용되는지에 의존한다. 즉, 비교를 위해 사용되는 2 개의 샘플들을 고르기 위해, 명시적으로/암시적으로 시그널링되는 EO 패턴이 사용된다. 이 개시물에서, 이웃하는 샘플들에 대한 샘플의 크기 차이에 기초하여, 오프셋이 샘플 값에 부가된다.

[0018] BO 필터링을 적용할 때, 샘플들은 강도 값과 같은 샘플 값에 기초하여 상이한 밴드들로 분류될 수 있고, 각 밴드는 연관된 오프셋을 갖는다. 밴드는 샘플 값들의 범위를 포함한다. 예를 들어, 0에서부터 255 까지의 범위인 샘플 값들은 (0 내지 31 로 라벨링된) 32 개의 동일한 밴드들로 나누어질 수도 있고, 샘플 값들 0-7 은 제 1 밴드이고, 샘플 값들 8-15 는 제 2 밴드이며, 샘플 값들 16-23 은 제 3 밴드이며, 모든 32 개의 밴드들에 대해 이러한 식이다. 밴드들은, 복수의 상이한 오프셋 값들 중에서, 어느 특정 오프셋 값을 샘플 또는 샘플들의 그룹에 적용할지를 결정하기 위해 사용될 수 있다. 예를 들어, 샘플이 (상기 예에서 제 2 밴드, 즉, 값들 8-15, 내에 있는) 10 의 값을 갖는 경우에, 제 2 밴드와 연관된 오프셋이 샘플 값에 부가될 수 있다.

[0019] 다양한 밴드들에 대한 오프셋들을 시그널링 및 생성하는 목적들을 위해, 밴드들은 2 개 이상의 그룹들로 그룹핑

될 수도 있다. B0 필터링을 위해, 샘플들은, 예를 들어, 상술된 바와 같이 32 개의 밴드들 (밴드들 0-31)로 그룹핑될 수도 있고, 그 밴드들은 2 개의 그룹들로 그룹핑될 수도 있다 (예컨대, 16 개 밴드들의 2 개의 그룹들, 4 개의 밴드들의 하나의 그룹 및 28 개 밴드들의 하나의 그룹, 8 개 밴드들의 하나의 그룹 및 24 개 밴드들의 하나의 그룹, 또는 다른 이러한 그룹핑들). 밴드들의 그룹핑은, 밴드들에 대한 오프셋 값들이 인코딩된 비디오 비트스트림에서 시그널링되는 순서를 결정하기 위해 사용될 수 있고, 및/또는, 특정 밴드들이 제로 외의 오프셋 값을 갖는지를 결정하기 위해 사용될 수 있다. 밴드들에 대한 오프셋들은 차분 코딩 기법들을 이용하여 시그널링될 수도 있고, 이 차분 코딩 기법들에서 현재 값은 현재 값과 이전 값 사이의 차이로서 시그널링된다. 시그널링 오버헤드를 감소시키기 위해서, 오직 작은 수의 밴드들 (예컨대, 4 개의 밴드들) 만이 비-제로 (non-zero) 오프셋 값들과 연관될 수도 있는 한편, 모든 다른 밴드들은 제로의 오프셋 값을 갖는 것으로 가정된다.

[0020] HEVC 표준에서 구현되는 바와 같이, SAO 필터링은 슬라이스 레벨에서 인에이블 및 디스에이블될 수도 있고, 또한, SAO 필터링은 루마 및 크로마 샘플들에 대해 별도로 인에이블 및 디스에이블될 수도 있다. 예를 들어, 비디오 데이터의 특정 슬라이스에 대해, 루마 및 크로마 샘플들 양자가 SAO 필터링될 수도 있고; 루마 샘플도 크로마 샘플도 SAO 필터링되지 않을 수도 있고; 또는 루마 및 크로마 샘플들 중 하나가 SAO 필터링될 수도 있는 한편, 다른 것은 SAO 필터링되지 않는다. SAO 필터링이 슬라이스에 대해 인에이블될 때, SAO 타입 및 오프셋 값들이 코딩 트리 유닛 (CTU) 레벨로 시그널링된다. 상기 소개된 바와 같이, SAO 타입은 BO 필터링, EO 필터링, 또는 무 SAO 필터링을 포함할 수 있다. 따라서, 특정 슬라이스에 대해 SAO 필터링이 인에이블되는 경우에도, 그 슬라이스의 일부 CTU 들은 SAO 필터링되지 않을 수도 있다 (즉, 일부 CTU 들은 무 SAO 필터링의 SAO 타입을 가질 것이다). 일부 경우들에서, SAO 타입 및 오프셋 값들은 일련의 병합 플래그들과 함께 시그널링될 수도 있고, 여기서, 제 1 병합 플래그에 대한 참 (true) 값은 CTU 에 대한 오프셋 타입 및 오프셋 값들이 좌측 이웃하는 CTU로부터 유래됨을 나타내고, 제 2 병합 플래그에 대한 참 값은 SAO 타입 및 오프셋 값들이 상부 이웃하는 CTU로부터 유래됨을 나타낸다. 양 병합 플래그들이 거짓 (false) 인 경우에는, 새로운 SAO 타입 및 새로운 오프셋 값들이 CTU 에 대해 전송된다.

[0021] 비록 HEVC 가 슬라이스 레벨에서 SAO 필터링을 인에이블 및 디스에이블시키고 CTU 레벨에서 SAO 정보를 시그널링하지만, 본 개시의 기법들은 이러한 시그널링이 상이한 레벨에서 일어나는 SAO 구성들로 또는 모두 상이한 시그널링 메커니즘들을 이용하는 SAO 구성들로 구현될 수도 있음이 고려된다.

[0022] SAO 정보 (예컨대, SAO 타입 및 복수의 오프셋 값들) 가 CTU 에 대해 시그널링된다. SAO 타입에 기초하여, 비디오 디코더는 복수의 오프셋 값들 중에서 어느 오프셋 값을 각 샘플에 대해 부가할지를 샘플 단위로 결정한다. 이하에서 보다 자세히 설명되는 바와 같이, 특정 샘플 또는 샘플들의 블록 (예컨대, CTU 에서 2x2 블록)에 대해 어느 오프셋 값을 적용할지의 결정은 샘플들의 블록의 특정 샘플에 대해 결정된 에지 및 밴드 분류들에 기초하여 결정될 수도 있다.

[0023] 이 개시물은, 본 명세서에서 피크 SAO 로서 지칭되는, 새로운 버전의 SAO 를 기술한다. 피크 SAO 는 상술된 EO 또는 BO 필터링과 함께 사용되는 추가적인 SAO 타입일 수도 있고, 또는, 상술된 SAO 필터와는 별개로 새로운 코딩 틀로서 구현될 수도 있다. 피크 SAO 를 구현할 때, 비디오 디코더는, 샘플들 및 그것의 이웃하는 샘플들의 관계가 동일한 경우에도, 비디오 데이터의 현재 블록에서 2 개 이상의 샘플들에 상이한 오프셋들을 적용할 수도 있다. 피크 SAO 의 하나의 예로서, 비디오 디코더는 샘플들이 모든 4 개의 이웃하는 샘플들보다 조금 더 큰 샘플 값 (예컨대, 루미넌스 값 또는 크로미넌스 값) 을 갖는 것을 결정하기 위해 제 1 샘플을 4 개의 이웃하는 샘플들에 대해 비교할 수도 있다. 제 1 샘플이 약간 더 큰 샘플 값을 갖는 것에 기초하여, 비디오 디코더는 제 1 샘플에 제 1 오프셋을 적용할 수도 있다. 제 2 샘플에 대해, 비디오 디코더는 샘플이 모든 4 개의 이웃하는 샘플들보다 훨씬 더 큰 샘플 값을 갖는 것을 결정하기 위해 제 2 샘플들 4 개의 유사하게 위치된 이웃하는 샘플들에 대해 비교할 수도 있다. 제 2 샘플이 훨씬 더 큰 샘플 값을 갖는 것에 기초하여, 비디오 디코더는 제 2 샘플에, 제 1 샘플에 대해 사용된 오프셋과는 상이한 오프셋을 적용할 수도 있다. 그 상이한 오프셋은, 예를 들어, 제 1 오프셋과는 상이한 단일의 오프셋 값일 수도 있고, 또는, 제 1 오프셋 플러스 추가적인 오프셋과 같이 2 개의 오프셋들의 결합일 수도 있다.

[0024] 전통적인 SAO 에서, 비디오 디코더는, 제 1 및 제 2 샘플이 그것들의 이웃하는 샘플들에 대해 동일한 관계 (즉, 모두보다 더 큼) 를 가지기 때문에, 상술된 제 1 및 제 2 샘플들 양자에 동일한 오프셋을 적용할 것이다. 하지만, 피크 SAO 에서, 비디오 디코더는, 제 1 샘플의 값은 그것의 이웃하는 샘플들보다 오직 약간 더 큰 반면에 제 2 샘플의 값은 그것의 이웃하는 샘플들보다 훨씬 더 크기 때문에, 제 1 및 제 2 샘플들에 상이한 오프셋들을 적용한다. 또한, 전통적인 SAO 에서, 샘플은 오직 2 개의 이웃하는 샘플들에 대해 비교되는 반면, 피

크 SAO 에서는, 샘플은 4 또는 8 개의 이웃들과 같이 다른 수들의 이웃하는 샘플들에 대해 비교될 수도 있다. 피크 SAO 의 이들 및 다른 양태들은 이하에서 보다 자세히 설명될 것이다.

[0025] 그리하여, 전통적인 SAO 기법들은 이웃하는 샘플들에 관해 동일한 특성들에 대해 상이한 크기들을 갖는 상이한 샘플들에 대해 상이한 오프셋들을 선택하기 위한 입도 (granularity) 를 제공하지 않는다. 예를 들어, 전통적인 SAO 에서, 비디오 코더는 이웃 샘플들에 비해 상대적으로 훨씬 더 큰 값을 갖는 샘플에 대해 이웃 샘플들에 비해 상대적으로 약간 더 큰 값을 갖는 샘플에 동일한 오프셋을 적용한다. 이 개시물은 보다 양호한 레이트-왜곡 트레이트오프를 갖는 비디오 코더들을 잠재적으로 초래하는 비디오 코더들에 적용가능한 예시적인 기법들을 기술한다. 실제로, 이 개시물에서 기술된 비디오 코더들은 이웃 샘플들과 블록 기반 그들 각각의 차이들의 상이한 샘플들에 대해 상이한 오프셋들을 허용할 수도 있다.

[0026] 본 개시의 다양한 기법들이 비디오 코더에 관하여 설명될 수도 있고, 이 비디오 코더는 비디오 인코더 또는 비디오 디코더 중 어느 일방을 지칠할 수도 있는 일반적인 용어이도록 의도된다. 달리 명시적으로 진술되지 않는 한, 비디오 인코더 또는 비디오 디코더에 관해 설명된 기법들은 비디오 인코더 또는 비디오 디코더의 타방에 의해 수행될 수 없다고 가정되어서는 아니된다. 예를 들어, 많은 경우들에서, 비디오 디코더는 인코딩된 비디오 데이터를 디코딩하기 위해 비디오 인코더와 동일한, 또는 대로는 상호 역의 코딩 기법을 수행한다. 많은 경우들에서, 비디오 인코더는 또한 비디오 디코딩 루프를 포함하고, 따라서 비디오 인코더는 비디오 데이터를 인코딩하는 것의 일부로서 비디오 디코딩을 수행한다. 따라서, 달리 진술되지 않는 한, 비디오 인코더에 관해 이 개시물에서 기술된 기법들은 또한 비디오 인코더에 의해 수행될 수도 있고, 그 역도 가능하다.

[0027] 이 개시물은 도한 현재 블록, 현재 픽처 등과 같은 용어들을 사용한다. 이 개시물의 맥락에서, “현재 (current)” 라는 용어는, 예를 들어, 이전에 또는 이미 코딩된 블록 또는 픽처, 또는 아직 코딩되지 않은 블록 또는 픽처에 대한 반대로서, 현재 코딩되고 있는 블록 또는 픽처를 식별하도록 의도된다.

[0028] 이 개시물은 설명의 용이성을 위해 HEVC 기술을 이용하여 다양한 비디오 코딩 기법들을 설명할 것이다. 하지만, 이 개시물의 기법들은 HEVC 에 한정되는 것으로 가정되어서는 아니되고, 사실, 이 개시물의 기법들은 HEVC 및 그것의 확장들에 대한 후속 표준들에서 구현될 수도 있음이 명시적으로 고려된다.

[0029] 도 1 은 이 개시물에서 기술된 기술들을 이용할 수도 있는 예시적인 비디오 인코딩 및 디코딩 시스템 (10) 을 나타내는 블록도이다. 도 1 에 도시된 바와 같이, 시스템 (10) 은, 목적지 디바이스 (14) 에 의해 더 나중 시간에 디코딩될 인코딩된 비디오 데이터를 생성하는 소스 디바이스 (12) 를 포함한다. 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 데스크탑 컴퓨터들, 노트북 (즉, 랩탑) 컴퓨터들, 태블릿 컴퓨터들, 셋탑 박스들, 소위 “스마트” 폰들과 같은 전화기 핸드셋들, 소위 “스마트” 패드들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 콘솔들, 비디오 스트리밍 디바이스 등을 포함한 광범위한 디바이스들 중 임의의 디바이스를 포함할 수도 있다. 일부 경우들에서, 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 무선 통신을 위해 갖추어질 수도 있다.

[0030] 목적지 디바이스 (14) 는 디코딩될 인코딩된 비디오 데이터를 링크 (16) 를 통해 수신할 수도 있다. 링크 (16) 는 인코딩된 비디오 데이터를 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 이동 가능한 임의의 타입의 매체 또는 디바이스를 포함할 수도 있다. 일 예에 있어서, 링크 (16) 는, 소스 디바이스 (12) 로 하여금 인코딩된 비디오 데이터를 직접 목적지 디바이스 (14) 로 실시간으로 송신할 수 있게 하는 통신 매체를 포함할 수도 있다. 인코딩된 비디오 데이터는, 무선 통신 프로토콜 등의 통신 표준에 따라 변조되고, 목적지 디바이스 (14) 로 송신될 수도 있다. 통신 매체는 무선 주파수 (RF) 스펙트럼 또는 하나 이상의 물리적 송신 라인들과 같은 임의의 무선 또는 유선 통신 매체를 포함할 수도 있다. 통신 매체는 로컬 영역 네트워크, 광역 네트워크, 또는 인터넷과 같은 글로벌 네트워크와 같은 패킷 기반 네트워크의 부분을 형성할 수도 있다. 통신 매체는 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로의 통신을 용이하게 하는데 유용할 수도 있는 임의의 다른 장비를 포함할 수도 있다.

[0031] 다른 예에서, 인코딩된 데이터는 출력 인터페이스 (22) 로부터 저장 디바이스 (26) 로 출력될 수도 있다. 유사하게, 인코딩된 데이터는 입력 인터페이스에 의해 저장 디바이스 (26) 로부터 액세스될 수도 있다. 저장 디바이스 (26) 는 하드 드라이브, 블루-레이 디스크들, DVD들, CD-ROM들, 플래시 메모리, 휴발성 또는 비휘발성 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 임의의 다른 적합한 디지털 저장 매체들과 같은 다양한 분산된 또는 국부적으로 액세스된 데이터 저장 매체들 중 임의의 데이터 저장 매체를 포함할 수도 있다.

추가의 예에 있어서, 저장 디바이스 (26) 는, 소스 디바이스 (12) 에 의해 생성된 인코딩된 비디오를 보유할 수도 있는 파일 서버 또는 다른 중간 저장 디바이스에 대응할 수도 있다. 목적지 디바이스 (14) 는 저장 디

바이스 (26)로부터의 저장된 비디오 데이터에 스트리밍 또는 다운로드를 통해 액세스할 수도 있다. 파일 서버는, 인코딩된 비디오 데이터를 저장하고 그 인코딩된 비디오 데이터를 목적지 디바이스 (14)로 송신할 수 있는 임의의 유형의 서버일 수도 있다. 예시적인 파일 서버들은 웹 서버 (예를 들어, 웹 사이트용), FTP 서버, 네트워크 접속형 저장 (NAS) 디바이스들, 또는 로컬 디스크 드라이브를 포함한다. 목적지 디바이스 (14)는, 인터넷 접속을 포함한, 임의의 표준 데이터 접속을 통해 인코딩된 비디오 데이터에 액세스할 수도 있다. 이는 파일 서버 상에 저장되는 인코딩된 비디오 데이터에 액세스하기에 적합한 무선 채널 (예를 들어, Wi-Fi 커넥션), 유선 커넥션 (예를 들어, DSL, 케이블 모뎀 등), 또는 이를 양자의 조합을 포함할 수도 있다.

인코딩된 비디오 데이터의 저장 디바이스 (26)로부터의 송신은 스트리밍 송신, 다운로드 송신, 또는 이들 양자의 조합일 수도 있다.

[0032] 본 개시의 기법들은 무선 애플리케이션들 또는 설정들로 반드시 한정되는 것은 아니다. 그 기법들은, 공중 경유 (over-the-air) 텔레비전 브로드캐스트들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, 예를 들어, 인터넷을 통한 스트리밍 비디오 송신들, 데이터 저장 매체 상의 저장을 위한 디지털 비디오의 인코딩, 데이터 저장 매체 상에 저장된 디지털 비디오의 디코딩, 또는 다른 애플리케이션들과 같은 다양한 멀티미디어 애플리케이션들 중 임의의 애플리케이션들의 지원으로 비디오 코딩에 적용될 수도 있다. 일부 예들에서, 시스템 (10)은 비디오 스트리밍, 비디오 재생, 비디오 브로드캐스팅, 및/또는 화상 통화 (video telephony) 와 같은 애플리케이션들을 지원하기 위하여 일방향 (one-way) 또는 양방향 (two-way) 비디오 송신을 지원하도록 구성될 수도 있다.

[0033] 도 1의 예에서, 소스 디바이스 (12)는 비디오 소스 (18), 비디오 인코더 (20), 및 출력 인터페이스 (22)를 포함한다. 일부 경우들에서, 출력 인터페이스 (22)는 변조기/복조기 (모뎀) 및/또는 송신기를 포함할 수도 있다. 소스 디바이스 (12)에 있어서, 비디오 소스 (18)는 비디오 캡처 디바이스, 예를 들어, 비디오 카메라, 이전에 캡처된 비디오를 포함하는 비디오 아카이브, 비디오 콘텐츠 제공자로부터 비디오를 수신하기 위한 비디오 피드 인터페이스, 및/또는 컴퓨터 그래픽스 데이터를 소스 비디오로서 생성하기 위한 컴퓨터 그래픽스 시스템과 같은 소스, 또는 그러한 소스들의 조합을 포함할 수도 있다. 일 예로서, 비디오 소스 (18)가 비디오 카메라이면, 소스 디바이스 (12) 및 목적지 디바이스 (14)는 소위 카메라 폰들 또는 비디오 폰들을 형성 할 수도 있다. 하지만, 본 개시에서 설명된 기법들은 일반적으로 비디오 코딩에 적용가능할 수도 있으며, 무선 및/또는 유선 애플리케이션들에 적용될 수도 있다.

[0034] 캡처되거나 사전-캡처되거나 또는 컴퓨터 생성된 비디오는 비디오 인코더 (20)에 의해 인코딩될 수도 있다. 인코딩된 비디오 데이터는 소스 디바이스 (12)의 출력 인터페이스 (22)를 통해 목적지 디바이스 (14)로 직접 송신될 수도 있다. 인코딩된 비디오 데이터는 또한 (또는 대안적으로), 디코딩 및/또는 재생을 위한 목적지 디바이스 (14) 또는 다른 디바이스들에 의한 더 나중의 액세스를 위해 저장 디바이스 (26) 상으로 저장될 수도 있다.

[0035] 목적지 디바이스 (14)는 입력 인터페이스 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (32)를 포함한다. 일부 경우들에 있어서, 입력 인터페이스 (28)는 수신기 및/또는 모뎀을 포함할 수도 있다. 목적지 디바이스 (14)의 입력 인터페이스 (28)는 인코딩된 비디오 데이터를 링크 (16)를 통해 수신한다. 링크 (16)를 통해 통신되는 또는 저장 디바이스 (26) 상에 제공되는 인코딩된 비디오 데이터는, 비디오 데이터를 디코딩함에 있어서 비디오 디코더 (30)와 같은 비디오 디코더에 의한 사용을 위해 비디오 인코더 (20)에 의해 생성된 다양한 선택스 엘리먼트들을 포함할 수도 있다. 그러한 선택스 엘리먼트들에는, 통신 매체 상에서 송신되거나 저장 매체 상에 저장되거나 또는 파일 서버에 저장되는 인코딩된 비디오 데이터와 함께 포함될 수도 있다.

[0036] 디스플레이 디바이스 (32)는 목적지 디바이스 (14)와 통합되거나 그 외부에 있을 수도 있다. 일부 예들에 있어서, 목적지 디바이스 (14)는 통합된 디스플레이 디바이스를 포함할 수도 있고, 또한, 외부 디스플레이 디바이스와 인터페이싱하도록 구성될 수도 있다. 다른 예들에 있어서, 목적지 디바이스 (14)는 디스플레이 디바이스일 수도 있다. 일반적으로, 디스플레이 디바이스 (32)는 디코딩된 비디오 데이터를 사용자에게 디스플레이하며, 이는 액정 디스플레이 (liquid crystal display; LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (organic light emitting diode; OLED) 디스플레이, 또는 다른 유형의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들 중 임의의 것을 포함할 수도 있다.

[0037] 비디오 인코더 (20) 및 비디오 디코더 (30)는 최근에 마무리된 고 효율 비디오 코딩 (HEVC) 표준과 같은 비디오 압축 표준에 따라 동작할 수도 있고, HEVC 테스트 모델 (HM)에 부합할 수도 있다. 비디오 인코더 (20)

및 비디오 디코더 (30)는 추가적으로, ITU-T VCEG (Video Coding Experts Group) 및 ISO/IEC MPEG (Motion Picture Experts Group)의 JCT-VC (Joint Collaboration Team on Video Coding) 및 JCT-3V (Joint Collaboration Team on 3D Video Coding Extension Development)에 의해 개발된, 범위 확장과 같은 HEVC 확장, 멀티뷰 확장 (MV-HEVC), 또는 스케일러블 확장 (SHVC)에 따라 동작할 수도 있다.

[0038] 비디오 인코더 (20) 및 비디오 디코더 (30)는 또한, ISO/IEC MPEG-4, Part 10, AVC (Advanced Video Coding)로서 대안적으로 지정되는, ITU-T H.264 표준과 같은 다른 독점적 또는 산업 표준들, 또는 SVC (Scalable Video Coding) 및 MVC (Multi-view Video Coding) 확장들과 같은 이러한 표준들의 확장들에 따라 동작할 수도 있다.

하지만, 본 개시의 기법들은 임의의 특정 코딩 표준으로 한정되지 않는다. 비디오 압축 표준들의 다른 예들은 ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 또는 ISO/IEC MPEG-2 Visual, ITU-T H.263, 및 ISO/IEC MPEG-4 Visual을 포함한다.

[0039] ITU-T VCEG (Q6/16) 및 ISO/IEC MPEG (JTC 1/SC 29/WG 11)는, (스크린 콘텐츠 코딩 및 하이-다이내믹-레인지 코딩을 위한 그것의 현재의 확장들 및 단기 확장들을 포함하는) 현재의 HEVC 표준의 것을 현저하게 초과하는 압축 능력을 갖는 미래의 비디오 코딩 기술의 표준화에 대한 잠재적인 필요성을 지금 연구하고 있다. 그 그룹들은 이 영역에서 그들의 전문가들에 의해 제안된 압축 기술 설계들을 평가하기 위해 JVET (Joint Video Exploration Team)로서 알려진 연합 공동작업 노력에서 이 팀구 활동에 대해 함께 작업하고 있다. JVET는 2015년 10월 19-21 일 동안 처음 만났다. 참조 소프트웨어의 하나의 버전, 즉, JEM 2 (Joint Exploration Model 2)는 [https://jvet.hhi.fraunhofer.de/svn/svn\\_HMJEMSoftware/tags/HM-16.6-JEM-2.0/](https://jvet.hhi.fraunhofer.de/svn/svn_HMJEMSoftware/tags/HM-16.6-JEM-2.0/) 으로부터 다운로드될 수 있다. JEM-2에 대한 알고리즘은 「J. Chen, E. Alshina, G. J. Sullivan, J.-R. Ohm, J. Boyce, "Algorithm description of Joint Exploration Test Model 2", JVET-B1001, San Diego, Mar. 2016」에서 설명되고, 그 설명은 참조에 의해 본원에 통합된다. 참조 소프트웨어의 다른 버전, 즉, JEM 3 (Joint Exploration Model 3)는 [https://jvet.hhi.fraunhofer.de/svn/svn\\_HMJEMSoftware/tags/HM-16.6-JEM-3.0/](https://jvet.hhi.fraunhofer.de/svn/svn_HMJEMSoftware/tags/HM-16.6-JEM-3.0/) 으로부터 다운로드될 수 있다. JEM3에 대한 알고리즘 설명은 또한 JVET-C1001로서 지정될 수도 있고, 참조에 의해 본원에 통합된다.

[0040] 비록 도 1에서는 도시되지 않았지만, 일부 양태들에서, 비디오 인코더 (20) 및 비디오 디코더 (30)는 오디오 인코더 및 디코더와 각각 통합될 수도 있으며, 오디오 및 비디오 양자의 인코딩을 공통 데이터 스트림 또는 별개의 데이터 스트림들로 처리하기 위해 적절한 MUX-DEMUX 유닛들 또는 다른 하드웨어 및 소프트웨어를 포함할 수도 있다. 적용가능하다면, 일부 예들에 있어서, MUX-DEMUX 유닛들은 ITU H.223 멀티플렉서 프로토콜, 또는 사용자 데이터그램 프로토콜 (UDP)과 같은 다른 프로토콜들에 부합할 수도 있다.

[0041] 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서 (DSP)들, 주문형 집적회로(ASIC)들, 필드 프로그래머블 게이트 어레이(FPGA)들, 이산 로직, 소프트웨어, 하드웨어, 펌웨어 또는 이들의 임의의 조합들과 같은 다양한 적합한 인코더 회로 또는 디코더 회로 중 임의의 회로로서 구현될 수도 있다. 기법들이 부분적으로 소프트웨어로 구현될 경우, 디바이스는 적합한 비일시적인 컴퓨터 판독가능 매체에 소프트웨어에 대한 명령들을 저장하고, 본 개시의 기법들을 수행하기 위해 하나 이상의 프로세서들을 사용하는 하드웨어에서 명령들을 실행할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30)의 각각은 하나 이상의 인코더들 또는 디코더들에 포함될 수도 있으며, 이들 중 어느 하나는 개별 디바이스에 있어서 결합된 인코더/디코더 (CODEC)의 부분으로서 통합될 수도 있다.

[0042] HEVC 및 다른 비디오 코딩 규격들에 있어서, 비디오 시퀀스는 통상적으로 픽처들의 시리즈를 포함한다. 픽처들은 "프레임들"로서 또한 지정될 수도 있다. 하나의 예시적인 접근으로, 픽처는  $S_L$ ,  $S_{Cb}$ , 및  $S_{Cr}$ 로서 표기되는 3 개의 샘플 어레이들을 포함할 수도 있다. 이러한 일 예시적인 접근에서,  $S_L$ 은 루마 샘플들의 2-차원 어레이 (즉, 블록)이다.  $S_{Cb}$ 는 Cb 크로미넌스 샘플들의 2-차원 어레이이다.  $S_{Cr}$ 는 Cr 크로미넌스 샘플들의 2-차원 어레이이다. 크로미넌스 샘플들은 또한, 본 명세서에서 "크로마 (chroma)" 샘플들로서 지정될 수도 있다. 다른 경우들에서, 픽처는 단색 (monochrome) 일 수도 있고, 루마 샘플들의 어레이만을 포함할 수도 있다.

[0043] 픽처의 인코딩된 표현을 생성하기 위하여, 비디오 인코더 (20)는 코딩 트리 유닛 (CTU)들의 셋트를 생성할 수도 있다. CTU들의 각각은 루마 샘플들의 코딩 트리 블록, 크로마 샘플들의 2 개의 대응 코딩 트리 블록들, 및 코딩 트리 블록들의 샘플들을 코딩하는데 사용된 신팩스 구조들을 포함할 수도 있다. 단색 픽처들 또는 3 개의 별개의 컬러 평면들을 갖는 픽처들에 있어서, CTU는 단일의 코딩 트리 블록, 및 그 코딩 트리 블록의

샘플들을 코딩하는데 사용되는 신택스 구조들을 포함할 수도 있다. 코딩 트리 블록은 샘플들의 NxN 블록일 수도 있다. CTU 는 또한 "트리 블록" 또는 "최대 코딩 유닛" (LCU) 으로서 지정될 수도 있다. HEVC 의 CTU 들은 H.264/AVC 와 같은 다른 표준들의 매크로블록들과 대략 유사할 수도 있다. 그러나, CTU 는 반드시 특정한 크기로 제한되지는 않고, 하나 이상의 코딩 유닛 (CU) 들을 포함할 수도 있다. 슬라이스는 레스터 스캔 순서에서 연속적으로 순서화된 정수 개의 CTU를 포함할 수도 있다.

[0044] 코딩된 CTU 를 생성하기 위하여, 비디오 인코더 (20) 는 CTU 의 코딩 트리 블록들에 대해 큐드트리 파티셔닝을 재귀적으로 수행하여, 코딩 트리 블록들을 코딩 블록들, 따라서, 일명 "코딩 트리 유닛들"로 분할할 수도 있다.

코딩 블록은 샘플들의 NxN 블록일 수도 있다. CU 는 루마 샘플들의 코딩 블록과, 루마 샘플 어레이, Cb 샘플 어레이, 및 Cr 샘플 어레이를 가지는 픽처의 크로마 샘플들의 2 개의 대응하는 코딩 블록들과, 코딩 블록들의 샘플들을 코딩하기 위하여 이용된 신택스 구조들을 포함할 수도 있다. 단색 픽처들 또는 3 개의 별개의 컬러 평면들을 갖는 픽처들에 있어서, CU 는 단일의 코딩 블록, 및 그 코딩 블록의 샘플들을 코딩하는데 사용된 신택스 구조들을 포함할 수도 있다.

[0045] 비디오 인코더 (20) 는 CU 의 코딩 블록을 하나 이상의 예측 블록들로 파티셔닝할 수도 있다. 예측 블록은, 동일한 예측이 적용되는 샘플들의 직사각형 (즉, 정사각형 또는 비-정사각형) 블록이다. CU 의 예측 유닛 (PU) 은 루마 샘플들의 예측 블록, 크로마 샘플들의 2 개의 대응하는 예측 블록들, 및 예측 블록들을 예측하기 위하여 이용된 신택스 구조들을 포함할 수도 있다. 단색 픽처들 또는 3 개의 별개의 컬러 평면들을 갖는 픽처들에 있어서, PU 는 단일의 예측 블록, 및 그 예측 블록을 예측하는데 사용되는 신택스 구조들을 포함할 수도 있다. 비디오 인코더 (20) 는 CU 의 각각의 PU 의 루마, Cb 및 Cr 예측 블록들에 대한 예측성 루마, Cb, 및 Cr 블록들을 생성할 수도 있다.

[0046] 비디오 인코더 (20) 는 인트라 예측 또는 인터 예측을 이용하여 PU에 대한 예측된 비디오 블록을 생성할 수도 있다. 비디오 인코더 (20) 가 인트라 예측을 이용하여 PU 의 예측 블록들을 생성하는 경우, 비디오 인코더 (20) 는 PU 와 연관된 픽처의 디코딩된 샘플들에 기초하여 PU 의 예측 블록들을 생성할 수도 있다. 비디오 인코더 (20) 가 인터 예측을 이용하여 PU 의 예측 블록들을 생성하는 경우, 비디오 인코더 (20) 는 PU 와 연관된 픽처 이외의 하나 이상의 픽처들의 디코딩된 샘플들에 기초하여 PU 의 예측 블록들을 생성할 수도 있다.

[0047] 비디오 인코더 (20) 가 CU 의 하나 이상의 PU들에 대한 예측 루마, Cb, 및 Cr 블록들을 생성한 후, 비디오 인코더 (20) 는 CU 에 대한 루마 잔차 블록을 생성할 수도 있다. CU 의 루마 잔차 블록에서의 각각의 샘플은 CU 의 예측성 루마 블록들 중 하나에서의 루마 샘플과 CU 의 오리지널 루마 코딩 블록에서의 대응하는 샘플 간의 차이를 나타낸다. 또한, 비디오 인코더 (20) 는 CU 에 대한 Cb 잔차 블록을 생성할 수도 있다. CU 의 Cb 잔차 블록에서의 각각의 샘플은 CU 의 예측성 Cb 블록들 중 하나에서의 Cb 샘플과 CU 의 오리지널 Cb 코딩 블록에서의 대응하는 샘플 간의 차이를 나타낼 수도 있다. 비디오 인코더 (20) 는 또한, CU 에 대한 Cr 잔차 블록을 생성할 수도 있다. CU 의 Cr 잔차 블록에서의 각각의 샘플은 CU 의 예측성 Cr 블록들 중 하나에서의 Cr 샘플과 CU 의 오리지널 Cr 코딩 블록에서의 대응하는 샘플 간의 차이를 나타낼 수도 있다.

[0048] 더욱이, 비디오 인코더 (20) 는 큐드 트리 파티셔닝을 이용하여, CU 의 루마, Cb, 및 Cr 잔차 블록들을 하나 이상의 루마, Cb, 및 Cr 변환 블록들로 분해할 수도 있다. 변환 블록은, 동일한 변환이 적용되는 샘플들의 직사각형 (예컨대, 정사각형 또는 비-정사각형) 블록이다. CU 의 변환 유닛 (TU) 은 루마 샘플들의 변환 블록, 크로마 샘플들의 2 개의 대응하는 변환 블록들, 및 변환 블록 샘플들을 변환하기 위하여 이용된 신택스 구조들을 포함할 수도 있다. 따라서, CU 의 각각의 TU 는 루마 변환 블록, Cb 변환 블록, 및 Cr 변환 블록과 연관될 수도 있다. TU 와 연관된 루마 변환 블록은 CU 의 루마 잔차 블록의 서브-블록일 수도 있다. Cb 변환 블록은 CU 의 Cb 잔차 블록의 서브-블록일 수도 있다. Cr 변환 블록은 CU 의 Cr 잔차 블록의 서브-블록일 수도 있다. 단색 픽처들 또는 3 개의 별개의 컬러 평면들을 갖는 픽처들에 있어서, TU 는 단일의 변환 블록, 및 그 변환 블록의 샘플들을 변환하는데 사용되는 신택스 구조들을 포함할 수도 있다.

[0049] 비디오 인코더 (20) 는 TU 의 루마 변환 블록에 하나 이상의 변환들을 적용하여 TU 에 대한 루마 계수 블록을 생성할 수도 있다. 계수 블록은 변환 계수들의 2-차원 어레이일 수도 있다. 변환 계수는 스칼라 양일 수도 있다. 비디오 인코더 (20) 는 TU 의 Cb 변환 블록에 하나 이상의 변환들을 적용하여 TU 에 대한 Cb 계수 블록을 생성할 수도 있다. 비디오 인코더 (20) 는 TU 의 Cr 변환 블록에 하나 이상의 변환들을 적용하여 TU 에 대한 Cr 계수 블록을 생성할 수도 있다.

[0050] CTU들, CU들, PU들, 및 Tu들을 갖는 상기 블록 구조는 일반적으로 HEVC 에서 사용된 블록 구조를 기술한다. 하지만, 다른 비디오 코딩 표준들은 상이한 블록 구조들을 사용할 수도 있다. 하나의 예로서, 비록 HEVC

는 Pu들 및 Tu들이 상이한 사이즈들 또는 형상들을 갖도록 허용하지만, 다른 비디오 코딩 표준들은 예측 블록들 및 변환 블록들이 동일한 사이즈를 갖도록 요구할 수도 있다. 본 개시의 기법들은 HEVC의 블록 구조에 한정되지 아니하고, 다른 블록 구조들과 호환가능할 수도 있다.

[0051] 계수 블록 (예컨대, 루마 계수 블록, Cb 계수 블록, 또는 Cr 계수 블록)을 생성한 후, 비디오 인코더 (20)는 계수 블록을 양자화할 수도 있다. 양자화는 일반적으로, 변환 계수들이 그 변환 계수들을 나타내는데 사용되는 데이터의 양을 가능하게는 감소시키도록 양자화되어 추가 압축을 제공하는 프로세스를 지칭한다. 비디오 인코더 (20)가 계수 블록을 양자화한 후, 비디오 인코더 (20)는 양자화된 변환 계수들을 표시하는 신택스 엘리먼트들을 엔트로피 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20)는 양자화된 변환 계수들을 나타내는 신택스 엘리먼트들에 대해 컨텍스트-적용 바이너리 산술 코딩 (CABAC)을 수행할 수도 있다.

[0052] 비디오 인코더 (20)는, 코딩된 꾹처들 및 연관된 데이터의 표현을 형성하는 비트들의 시퀀스를 포함하는 비트스트림을 출력할 수도 있다. 비트스트림은 네트워크 추상화 계층 (NAL) 유닛들의 시퀀스를 포함할 수도 있다. NAL 유닛은, NAL 유닛 내의 데이터의 타입의 표시, 및 필요에 따라 에뮬레이션 방지 비트들로 산재된 RBSP (raw byte sequence payload)의 형태로 그 데이터를 포함하는 바이트들을 포함하는 신택스 구조이다. NAL 유닛들 각각은 NAL 유닛 헤더를 포함하고 RBSP를 캡슐화한다. NAL 유닛 헤더는, NAL 유닛 타입 코드를 나타내는 신택스 엘리먼트를 포함할 수도 있다. NAL 유닛의 NAL 유닛 헤더에 의해 명시된 NAL 유닛 타입 코드는 NAL 유닛의 타입을 나타낸다. RBSP는, NAL 유닛 내에서 캡슐화되는 정수 개수의 바이트들을 포함하는 신택스 구조일 수도 있다. 일부 경우들에서, RBSP는 제로 비트들을 포함한다.

[0053] 상이한 타입들의 NAL 유닛들은 상이한 타입들의 RBSP들을 캡슐화할 수도 있다. 예를 들어, 제 1 타입의 NAL 유닛은 PPS에 대한 RBSP를 캡슐화할 수도 있고 제 2 타입의 NAL 유닛은 코딩된 슬라이스에 대한 RBSP를 캡슐화할 수도 있고, 제 3 타입의 NAL 유닛은 SEI 메시지들에 대한 RBSP를 캡슐화할 수 있는 등이다. (파라미터 셋트들 및 SEI 메시지들에 대한 RBSP들에 대한 것과는 반대로) 비디오 코딩 데이터에 대한 RBSP들을 캡슐화하는 NAL 유닛들은 VCL NAL 유닛들로 지칭될 수도 있다.

[0054] 비디오 디코더 (30)는 비디오 인코더 (20)에 의해 생성된 비트스트림을 수신할 수도 있다. 또한, 비디오 디코더 (30)는 비트스트림을 파싱하여, 그 비트스트림으로부터 신택스 엘리먼트들을 획득할 수도 있다. 비디오 디코더 (30)는 비트스트림으로부터 획득된 신택스 엘리먼트들에 적어도 부분적으로 기초하여 비디오 데이터의 꾹처들을 재구성할 수도 있다. 비디오 데이터를 재구성하기 위한 프로세스는 일반적으로, 비디오 인코더 (20)에 의해 수행되는 프로세스에 대해 상호 역일 수도 있다. 또한, 비디오 디코더 (30)는 현재 CU의 TU들과 연관된 계수 블록들을 역양자화할 수도 있다. 비디오 디코더 (30)는 계수 블록들에 대해 역변환들을 수행하여 현재 CU의 TU들과 연관된 변환 블록들을 재구성할 수도 있다. 비디오 디코더 (30)는 현재 CU의 PU들에 대한 예측성 블록들의 샘플들을, 현재 CU의 TU들의 변환 블록들의 대응하는 샘플들에 부가함으로써, 현재 CU의 코딩 블록들을 재구성할 수도 있다. 꾹처의 각각의 CU에 대한 코딩 블록들을 재구성함으로써, 비디오 디코더 (30)는 꾹처를 재구성할 수도 있다.

[0055] HEVC 및 JEM 기법들의 양태들이 이제 논의될 것이다. 도 2는 HEVC 디코더의 일 예시적인 블록도를 나타낸다. 도 2에서 도시된 비디오 디코더는 상술된 비디오 디코더 (30)에 대응할 수도 있고, 이하에서 보다 자세히 설명될 것이다. HEVC는 디-블록킹 필터 (de-blocking filter; DBF) 및 SAO를 포함하는 2 가지 인루프 필터들을 채용한다. HEVC 디코딩 및 SAO에 관한 추가적인 상세들은 「C. Fu, E. Alshina, A. Alshin, Y. Huang, C. Chen, Chia. Tsai, C. Hsu, S. Lei, J. Park, W. Han, “Sample adaptive offset in the HEVC standard,” IEEE Trans. Circuits Syst. Video Technol., 22(12): 1755-1764 (2012)」에서 설명된다.

[0056] 예시된 바와 같이, DBF에 대한 입력은, 재구성 블록으로부터의 출력으로 도시된 바와 같이, 인트라 또는 인터 예측 후의 재구성된 이미지일 수도 있다. DBF는 코딩된 블록 경계들에서의 아티팩트들의 겸출을 수행하고, 선택된 필터를 적용함으로써 그것들을 줄인다. H.264/AVC 디블록킹 필터에 비해, HEVC 디블록킹 필터는 시각적 아티팩트들의 현저한 감소를 여전히 달성하면서도 더 낮은 계산적 복잡도 및 더 양호한 병렬적 프로세싱을 갖는다. 추가적인 예들에 대해서는, 「A. Norkin, G. Bjontegaard, A. Fuldsseth, M. Narroschke, M. Ikeda, K. Andersson, Minhua Zhou, G. Van der Auwera, “HEVC Deblocking Filter,” IEEE Trans. Circuits Syst. Video Technol., 22(12): 1746 - 1754 (2012)」를 참조하라.

[0057] SAO에 대한 입력은 디블록킹 필터로부터의 출력으로 도시된 바와 같이, 디블록킹 필터링을 호출한 후의 재구성된 이미지일 수도 있다. SAO의 개념은, 먼저 구역 샘플들을 선택된 분류자로 다수의 카테고리들로 분류하

고, 각각의 카테고리에 대한 오프셋을 획득하며, 그 다음에, 카테고리의 각 샘플에 대해 오프셋을 부가함으로써, 구역의 평균 샘플 왜곡을 감소시키기 위한 것이고, 여기서, 구역의 오프셋 및 분류자 인덱스는 비트스트림에서 코딩된다. HEVC에서, 구역 (SAO 파라미터를 시그널링을 위한 유닛)은 CTU 이도록 정의된다.

[0058] 낮은 복잡도의 요건들을 만족시킬 수 있는 2 가지 SAO 타입들이 HEVC에서 채택된다. 그들 2 가지 타입들은 위에서 소개된 바와 같이 EO 및 BO이다. SAO 타입의 인덱스가 코딩된다 ([0, 2]의 범위 내에 있음). SAO 타입의 시그널링의 맥락에서, SAO를 수행하지 않는 것 또는 SAO를 생략하는 것은 SAO의 제3 타입으로 고려될 수도 있다. EO에 있어서, 샘플 분류는 1-D 방향 패턴들: 수평적, 수직적,  $135^\circ$ , 대각선, 및  $45^\circ$  사선에 따라 현재 샘플들과 이웃하는 샘플들 사이의 비교에 기초한다.

[0059] 도 3a 내지 도 3d는 EO 샘플 분류를 위한 4 개의 1-D 방향 패턴드를 도시한다: 수평적 (도 3a, EO 클래스 = 0), 수직적 (도 3b, EO 클래스 = 1),  $135^\circ$  사선 (도 3c, EO 클래스 = 2), 및  $45^\circ$  사선 (도 3d, EO 클래스 = 3). SAO에 관련된 추가적인 상세들은 「C. Fu, E. Alshina, A. Alshin, Y. Huang, C. Chen, Chia. Tsai, C. Hsu, S. Lei, J. Park, W. Han, “Sample adaptive offset in the HEVC standard,” IEEE Trans. Circuits Syst. Video Technol., 22(12): 1755-1764 (2012)」에서 기술된다.

[0060] 선택된 EO 패턴에 따라, 테이블 I에서 edgeIdx에 의해 표시된 5 개의 카테고리들이 추가로 정의된다. 0~3과 동일한 edgeIdx에 대해, 오프셋의 크기가 시그널링되는 한편, 부호 플래그가 암시적으로 코딩된다, 즉, edgeIdx에 대한 네거티브 오프셋은 0 또는 1과 동일하고 edgeIdx에 대한 포지티브 오프셋은 2 또는 3과 동일하다. 4와 동일한 edgeIdx에 대해, 오프셋은 항상 0으로 설정되고, 이는 이 경우에 대해 연산이 필요하지 않음을 의미한다.

## 표 1

표 I: EO에 대한 분류

카테고리 (edgeIdx)	조건
0	$c < a \ \&\& c < b$
1	$(c < a \ \&\& c == b) \    \ (c == a \ \&\& c < b)$
2	$(c > a \ \&\& c == b) \    \ (c == a \ \&\& c > b)$
3	$c > a \ \&\& c > b$
4	상기 어느 것도 아님

[0062] BO에 대해, 샘플 분류는 샘플 값들에 기초한다. 각각의 컬러 컴포넌트는 BO 타입 SAO 필터링을 위한 분류에 대한 그 자신의 SAO 파라미터들을 가질 수도 있다. BO는 동일한 밴드의 모든 샘플들에 하나의 오프셋이 부가됨을 의미한다. 샘플 값 범위는 32개의 밴드들로 동등하게 분할된다. 0에서부터 255까지의 범위인 8비트 샘플들에 대해, 밴드의 폭은 8이고, 8k에서부터 8k+7까지의 샘플 값들은 밴드 k에 속하고, 여기서, k는 0에서부터 31까지의 범위이다. 밴드에서의 원래의 샘플들과 재구성된 샘플들 사이의 평균 차이(즉, 밴드의 오프셋)가 디코더(예컨대, 비디오 디코더(30))에 시그널링된다. 오프셋 부호들에 대한 제약은 존재하지 않는다. 오직 4개의 연속적인 밴드들의 오프셋들 및 시작 밴드 포지션만이 디코더(예컨대, 비디오 디코더(30))에 시그널링된다.

[0063] 부 정보의 시그널링을 위해, 부 정보를 감소시키기 위해, 다수의 CTU들은 SAO 파라미터들을 공유하도록 함께 병합될 수 있다 (sao\_merge\_left\_flag를 1과 동일하게 설정하는 것을 통해) 상부 CTU로부터 파라미터들을 카피하는 것 또는 sao\_merge\_up\_flag를 1과 동일하게 설정하는 것을 통해) 좌측 CTU로부터 파라미터들을 카피하는 것 중 어느 일방).

[0064] 선택스 테이블들

## 표 2

코딩 트리 유닛 선택스

coding_tree_unit( ) {	디스크립터
xCtb = ( CtbAddrInRs % PicWidthInCtbsY ) << CtbLog2SizeY	

yCtb = ( CtbAddrInRs / PicWidthInCtbsY ) << CtbLog2SizeY	
if( slice_sao_luma_flag    slice_sao_chroma_flag )	
sao( xCtb >> CtbLog2SizeY, yCtb >> CtbLog2SizeY )	
coding_quadtree( xCtb, yCtb, CtbLog2SizeY, 0 )	
}	

### 표 3

#### 샘플 적용형 오프셋 선택스

sao( rx, ry ) {	디스크립터
if( rx > 0 ) {	
leftCtbInSliceSeg = CtbAddrInRs > SliceAddrRs	
leftCtbInTile = TileId[ CtbAddrInTs ] == TileId[ CtbAddrRsToTs[ CtbAddrInRs - 1 ] ]	
if( leftCtbInSliceSeg && leftCtbInTile )	
sao_merge_left_flag	ae(v)
}	
if( ry > 0 && !sao_merge_left_flag ) {	
upCtbInSliceSeg = ( CtbAddrInRs - PicWidthInCtbsY ) >= SliceAddrRs	
upCtbInTile = TileId[ CtbAddrInTs ] ==	
TileId[ CtbAddrRsToTs[ CtbAddrInRs - PicWidthInCtbsY ] ]	
if( upCtbInSliceSeg && upCtbInTile )	
sao_merge_up_flag	ae(v)
}	
if( !sao_merge_up_flag && !sao_merge_left_flag )	
for( cIdx = 0; cIdx < ( ChromaArrayType != 0 ? 3 : 1 ); cIdx++ )	
if( ( slice_sao_luma_flag && cIdx == 0 )	
( slice_sao_chroma_flag && cIdx > 0 ) ) {	
if( cIdx == 0 )	
sao_type_idx_luma	ae(v)
else if( cIdx == 1 )	
sao_type_idx_chroma	ae(v)
if( SaoTypeIdx[ cIdx ][ rx ][ ry ] != 0 ) {	
for( i = 0; i < 4; i++ )	
sao_offset_abs[ cIdx ][ rx ][ ry ][ i ]	ae(v)
if( SaoTypeIdx[ cIdx ][ rx ][ ry ] == 1 ) {	
for( i = 0; i < 4; i++ )	
if( sao_offset_abs[ cIdx ][ rx ][ ry ][ i ] != 0 )	
sao_offset_sign[ cIdx ][ rx ][ ry ][ i ]	ae(v)
sao_band_position[ cIdx ][ rx ][ ry ]	ae(v)
} else {	
if( cIdx == 0 )	
sao_eo_class_luma	ae(v)
if( cIdx == 1 )	
sao_eo_class_chroma	ae(v)
}	
}	
}	
}	
}	

#### 시맨틱스

[0067] sao\_merge\_left\_flag 가 1 과 동일하다는 것은 선택스 엘리먼트들 sao\_type\_idx\_luma, sao\_type\_idx\_chroma, sao\_band\_position, sao\_eo\_class\_luma, sao\_eo\_class\_chroma, sao\_offset\_abs, 및 sao\_offset\_sign 이 좌측 코딩 트리 블록의 대응하는 선택스 엘리먼트들로부터 도출됨을 명시한다. Sao\_merge\_left\_flag 가 0 과 동

일하다는 것은 이들 선택스 엘리먼트들이 좌측 코딩 트리 블록의 대응하는 선택스 엘리먼트들로부터 도출되지 않음을 명시한다. `sao_merge_left_flag` 가 존재하지 않을 때, 0 과 동일한 것으로 추론된다.

[0069] `sao_merge_up_flag` 가 1 과 동일하다는 것은 선택스 엘리먼트들 `sao_type_idx_luma`, `sao_type_idx_chroma`, `sao_band_position`, `sao_eo_class_luma`, `sao_eo_class_chroma`, `sao_offset_abs`, 및 `sao_offset_sign` 이 상부 코딩 트리 블록의 대응하는 선택스 엘리먼트들로부터 도출됨을 명시한다. `sao_merge_up_flag` 가 0 과 동일하다는 것은 이들 선택스 엘리먼트들이 상부 코딩 트리 블록의 대응하는 선택스 엘리먼트들로부터 도출되지 않음을 명시한다. `sao_merge_up_flag` 가 존재하지 않을 때, 0 과 동일한 것으로 추론된다.

[0070] `sao_type_idx_luma` 는 루마 컴포넌트에 대한 오프셋 탑입을 명시한다. 어레이 `SaoTypeIdx[ cIdx ][ rx ][ ry ]` 는 컬러 컴포넌트 `cIdx` 에 대해 로케이션 (`rx`, `ry`) 에서의 코딩 트리 블록에 대해 표 7-8에서 명시된 바와 같은 오프셋 탑입을 명시한다. `SaoTypeIdx[ 0 ][ rx ][ ry ]` 의 값은 다음과 같이 도출된다:

[0071] - `sao_type_idx_luma` 이 존재하는 경우, `SaoTypeIdx[ 0 ][ rx ][ ry ]` 은 `sao_type_idx_luma` 와 동일하게 설정된다.

[0072] - 그렇지 않은 경우 (`sao_type_idx_luma` 가 존재하지 않는 경우), `SaoTypeIdx[ 0 ][ rx ][ ry ]` 는 다음과 같이 도출된다:

[0073] - `sao_merge_left_flag` 가 1 과 동일한 경우, `SaoTypeIdx[ 0 ][ rx ][ ry ]` 은 `SaoTypeIdx[ 0 ][ rx - 1 ][ ry ]` 과 동일하게 설정된다.

[0074] - 그렇지 않은 경우, `sao_merge_up_flag` 가 1 과 동일한 경우, `SaoTypeIdx[ 0 ][ rx ][ ry ]` 은 `SaoTypeIdx[ 0 ][ rx ][ ry - 1 ]` 과 동일하게 설정된다.

[0075] - 그렇지 않은 경우, `SaoTypeIdx[ 0 ][ rx ][ ry ]` 은 0 과 동일하게 설정된다.

[0076] `sao_type_idx_chroma` 는 크로마 컴포넌트에 대한 오프셋 탑입을 명시한다. `SaoTypeIdx[ cIdx ][ rx ][ ry ]` 의 값들은 1..2 과 동일한 `cIdx` 에 대해 다음과 같이 도출된다:

[0077] - `sao_type_idx_chroma` 가 존재하는 경우, `SaoTypeIdx[ cIdx ][ rx ][ ry ]` 는 `sao_type_idx_chroma` 와 동일하게 설정된다.

[0078] - 그렇지 않은 경우 (`sao_type_idx_chroma` 가 존재하지 않는 경우), `SaoTypeIdx[ cIdx ][ rx ][ ry ]` 는 다음과 같이 도출된다:

[0079] - `sao_merge_left_flag` 가 1 과 동일한 경우, `SaoTypeIdx[ cIdx ][ rx ][ ry ]` 는 `SaoTypeIdx[ cIdx ][ rx - 1 ][ ry ]` 과 동일하게 설정된다.

[0080] - 그렇지 않은 경우, `sao_merge_up_flag` 가 1 과 동일한 경우, `SaoTypeIdx[ cIdx ][ rx ][ ry ]` 은 `SaoTypeIdx[ cIdx ][ rx ][ ry - 1 ]` 과 동일하게 설정된다.

[0081] - 그렇지 않은 경우, `SaoTypeIdx[ cIdx ][ rx ][ ry ]` 은 0 과 동일하게 설정된다.

#### 표 4

표 7-8 - SAO 탑입의 명세

<code>SaoTypeIdx[ cIdx ][ rx ][ ry ]</code>	SAO 탑입 (정보성)
0	적용 안됨
1	밴드 오프셋
2	에지 오프셋

[0083] `sao_offset_abs[ cIdx ][ rx ][ ry ][ i ]` 는 컬러 컴포넌트 `cIdx` 에 대해 로케이션 (`rx`, `ry`) 에서의 코딩 트리 블록에 대한 `i`-번째 카테고리의 오프셋 값은 명시한다.

[0084] `sao_offset_abs[ cIdx ][ rx ][ ry ][ i ]` 이 존재하지 않을 때, 다음과 같이 추론된다:

[0085] - `sao_merge_left_flag` 가 1 과 동일한 경우, `sao_offset_abs[ cIdx ][ rx ][ ry ][ i ]` 는

sao\_offset\_abs[ cIdx ][ rx - 1 ][ ry ][ i ] 과 동일한 것으로 추론된다.

[0086] - 그렇지 않은 경우, sao\_merge\_up\_flag 가 1 과 동일한 경우, sao\_offset\_abs[ cIdx ][ rx ][ ry ][ i ] 는 sao\_offset\_abs[ cIdx ][ rx ][ ry - 1 ][ i ] 과 동일한 것으로 추론된다.

[0087] - 그렇지 않은 경우, sao\_offset\_abs[ cIdx ][ rx ][ ry ][ i ] 는 0 과 동일한 것으로 추론된다.

[0088] sao\_offset\_sign[ cIdx ][ rx ][ ry ][ i ] 는 컬러 컴포넌트 cIdx 에 대해 로케이션 ( rx, ry ) 에서의 코딩 트리 블록에 대한 i-번째 카테고리의 오프셋 값의 부호를 명시한다.

[0089] sao\_offset\_sign[ cIdx ][ rx ][ ry ][ i ] 가 존재하지 않을 때, 다음과 같이 추론된다:

[0090] - Sao\_merge\_left\_flag 가 1 과 동일한 경우, sao\_offset\_sign[ cIdx ][ rx ][ ry ][ i ] 는 sao\_offset\_sign[ cIdx ][ rx - 1 ][ ry ][ i ] 과 동일한 것으로 추론된다.

[0091] - 그렇지 않은 경우, sao\_merge\_up\_flag 가 1 과 동일한 경우, sao\_offset\_sign[ cIdx ][ rx ][ ry ][ i ] 는 sao\_offset\_sign[ cIdx ][ rx ][ ry - 1 ][ i ] 과 동일한 것으로 추론된다.

[0092] - 그렇지 않은 경우, SaoTypeIdx[ cIdx ][ rx ][ ry ] 이 2 와 동일한 경우, 다음과 같이 추론된다:

[0093] - i 가 0 또는 1 과 동일한 경우, sao\_offset\_sign[ cIdx ][ rx ][ ry ][ i ] 는 0 과 동일한 것으로 추론된다.

[0094] - 그렇지 않은 경우 (i 가 2 또는 3 과 동일), sao\_offset\_sign[ cIdx ][ rx ][ ry ][ i ] 는 1 과 동일한 것으로 추론된다.

[0095] - 그렇지 않은 경우, sao\_offset\_sign[ cIdx ][ rx ][ ry ][ i ] i 는 0 과 동일한 것으로 추론된다.

[0096] 가변 log2offsetScale 은 다음과 같이 도출된다:

[0097] - cIdx 가 0 과 동일한 경우, log2offsetScale 은 log2\_sao\_offset\_scale\_luma 와 동일하게 설정된다.

[0098] - 그렇지 않은 경우 (cIdx 가 1 또는 2 와 동일), log2offsetScale 은 log2\_sao\_offset\_scale\_chroma 와 동일하게 설정된다.

[0099] 0 에서부터 4 까지 포함하는 범위인 i 에 대해 SaoOffsetVal[ cIdx ][ rx ][ ry ][ i ] 는 다음과 같이 도출된다:

[0100] SaoOffsetVal[ cIdx ][ rx ][ ry ][ 0 ] = 0

[0101] for( i = 0; i < 4; i++ )

[0102] SaoOffsetVal[ cIdx ][ rx ][ ry ][ i + 1 ] =

[0103] ( 1 - 2 \* sao\_offset\_sign[ cIdx ][ rx ][ ry ][ i ] ) \* (7-72)

[0104] sao\_offset\_abs[ cIdx ][ rx ][ ry ][ i ] << log2offsetScale

[0105] sao\_band\_position[ cIdx ][ rx ][ ry ] 는 SaoTypeIdx[ cIdx ][ rx ][ ry ] 가 1 과 동일할 때 샘플 범위의 밴드 오프셋의 변위를 명시한다.

[0106] sao\_band\_position[ cIdx ][ rx ][ ry ] 가 존재하지 않을 때, 다음과 같이 추론된다:

[0107] - sao\_merge\_left\_flag 가 1 과 동일한 경우, sao\_band\_position[ cIdx ][ rx ][ ry ] 는 sao\_band\_position[ cIdx ][ rx - 1 ][ ry ] 와 동일한 것으로 추론된다.

[0108] - 그렇지 않은 경우, sao\_merge\_up\_flag 가 1 과 동일한 경우, sao\_band\_position[ cIdx ][ rx ][ ry ] 는 sao\_band\_position[ cIdx ][ rx ][ ry - 1 ] 과 동일한 것으로 추론된다.

[0109] - 그렇지 않은 경우, sao\_band\_position[ cIdx ][ rx ][ ry ] 는 0 과 동일한 것으로 추론된다.

[0110] sao\_eo\_class\_luma 는 루마 컴포넌트에 대한 예지 오프셋 클래스를 명시한다. 아래로 SaoEoClass[ cIdx ][ rx ][ ry ] 는 컬러 컴포넌트 cIdx 에 대해 로케이션 ( rx, ry ) 에서의 코딩 트리 블록에 대해 표 7-9 에서 명시된 바와 같은 오프셋 타입을 명시한다. SaoEoClass[ 0 ][ rx ][ ry ] 의 값들은 다음과 같이 도출된다:

- [0111] - sao\_eo\_class\_luma 가 존재하는 경우, SaoEoClass[ 0 ][ rx ][ ry ] 는 sao\_eo\_class\_luma 와 동일하게 설정된다.
- [0112] - 그렇지 않은 경우 (sao\_eo\_class\_luma 가 존재하지 않는 경우), SaoEoClass[ 0 ][ rx ][ ry ] 는 다음과 같이 도출된다:
- [0113] - sao\_merge\_left\_flag 가 1 과 동일한 경우, SaoEoClass[ 0 ][ rx ][ ry ] 는 SaoEoClass[ 0 ][ rx - 1 ][ ry ] 과 동일하게 설정된다.
- [0114] - 그렇지 않은 경우, sao\_merge\_up\_flag 가 1 과 동일한 경우, SaoEoClass[ 0 ][ rx ][ ry ] 는 SaoEoClass[ 0 ][ rx ][ ry - 1 ] 과 동일하게 설정된다.
- [0115] - 그렇지 않은 경우, SaoEoClass[ 0 ][ rx ][ ry ] 는 0 과 동일하게 설정된다.
- [0116] sao\_eo\_class\_chroma 는 크로마 컴포넌트들에 대한 예지 오프셋 클래스를 명시한다. SaoEoClass[ cIdx ][ rx ][ ry ] 의 값들은 1..2 와 동일한 cIdx 에 대해 다음과 같이 도출된다:
- [0117] - sao\_eo\_class\_chroma 가 존재하는 경우, SaoEoClass[ cIdx ][ rx ][ ry ] 는 sao\_eo\_class\_chroma 와 동일하게 설정된다.
- [0118] - 그렇지 않은 경우 (sao\_eo\_class\_chroma 가 존재하지 않는 경우), SaoEoClass[ cIdx ][ rx ][ ry ] 는 다음과 같이 도출된다:
- [0119] - sao\_merge\_left\_flag 가 1 과 동일한 경우, SaoEoClass[ cIdx ][ rx ][ ry ] 는 SaoEoClass[ cIdx ][ rx - 1 ][ ry ] 과 동일하게 설정된다.
- [0120] - 그렇지 않은 경우, sao\_merge\_up\_flag 가 1 과 동일한 경우, SaoEoClass[ cIdx ][ rx ][ ry ] 는 SaoEoClass[ cIdx ][ rx ][ ry - 1 ] 과 동일하게 설정된다.
- [0121] - 그렇지 않은 경우, SaoEoClass[ cIdx ][ rx ][ ry ] 는 0 과 동일하게 설정된다.

## 표 5

표 7-9 - SAO 예지 오프셋 클래스의 명세

SaoTypeIdx[ cIdx ][ rx ][ ry ]	SAO 예지 오프셋 클래스 (정보성)
0	1D 0-도 예지 오프셋
1	1D 90-도 예지 오프셋
2	1D 135-도 예지 오프셋
3	1D 45-도 예지 오프셋

- [0122] 비디오 인코더 (20) 및 비디오 디코더 (30) 는 다양한 JEM 필터 필터링 기법들을 구현하도록 구성될 수도 있다. 그들 JEM 필터링 기법들의 양태들이 이제 설명될 것이다. 변형된 DB 및 HEVC SAO 에 추가하여, JEM 은 지오메트리 변환-기반 적응형 루프 필터링 (Geometry transformation-based Adaptive Loop Filtering; GALF) 이라 불리는 또 다른 필터링 방법을 포함하였다. GALF 는 몇가지 새로운 양태들을 도입함으로써 HEVC 스테이지에서 연구된 ALF 의 코딩 효율성을 향상시키는 것을 목적으로 한다. ALF 는 Wiener-기반 적응형 필터를 사용함으로써 원래의 샘플들과 디코딩된 샘플들 사이의 평균 제곱 오차 (mean square error) 를 최소화하는 것을 목적으로 하고 있다. 꾹처에서의 샘플들은 다수의 카테고리들로 분류되고, 각 카테고리에서의 샘플들은 그 다음에 그것들의 연관된 적응형 필터로 필터링된다. 평균 제곱 오차 및 오버헤드 사이의 드레이드오프를 최적화하기 위해 필터 계수들이 시그널링 또는 상속될 수도 있다. ALF 의 성능을 추가로 향상시키기 위해 지오메트리 변환-기반 ALF (GALF) 스킴 (scheme) 이 제안되었고, 이는 ALF 전에 재구성된 샘플들의 그라디언트의 배향에 의존하여 필터 지원 구역에서 샘플들에 적용될, 회전, 대각선 및 수직 플립과 같은 기하학적 변환들을 도입한다.

- [0123] ALF/GLF 에 대한 입력은 SAO 를 호출한 후의 재구성된 이미지 (예컨대, 도 2 에서 샘플 적응형 오프셋의 출력) 일 수도 있다. 「M. Karczewicz, L. Zhang, W.-J. Chien, X. Li, “EE2.5: Improvements on adaptive loop filter,” Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Doc. JVET-B0060, 2nd Meeting: San Diego, USA, 20 Feb. - 26 Feb. 2016」 및 「M. Karczewicz, L. Zhang, W.-J.

Chien, X. Li, "EE2.5: Improvements on adaptive loop filter," Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Doc. JVET-C0038, 3rd Meeting: Geneva, CH, 26 May - 1 June 2016」에서 기술된 바와 같이, 지오메트릭 변환-기반 ALF (GALF) 이 제안되었고, 그것은 JEM 의 가장 최근 버전, 즉, JEM 3.0 에 채택되었다. GALF 에서, 분류는 대각선 그라디언트가 고려되어 수정되고, 필터 계수들에 기하학적 변환들이 적용될 수 있을 것이다. 각각의  $2 \times 2$  블록은 그것의 방향성 및 활성도의 양자화된 값에 기초하여 25 개 클래스들 중 하나의 출력으로 카테고리화된다. 상세한 내용들은 다음의 서브-섹션들에서 설명된다.

[0125] 비디오 인코더 (20) 및 비디오 디코더 (30) 는 정해진 필터들로부터 필터들을 예측하도록 구성될 수도 있다. 또한, 코딩 효율성을 향상시키기 위해, 시간 예측이 이용가능하지 않을 때 (인트라 프레임들), 16 개의 정해진 필터들의 셋트가 각 클래스에 할당된다. 정해진 필터의 사용을 나타내기 위해, 각 클래스에 대한 플래그가 시그널링되고, 필요한 경우, 선택된 정해진 필터의 인덱스가 시그널링된다. 주어진 클래스에 대해 정해진 필터가 선택될 때에도, 적응형 필터  $f(k, l)$  의 계수들이 여전히 이 클래스에 대해 전송될 수 있고, 이 경우에 재구성된 이미지에 적용될 필터의 계수들은 계수들의 양 셋트들의 합이다. 클래스들의 넘버는, 상이한 정해진 필터들이 그것들에 대해 선택된 경우에도, 비트스트림에서 시그널링된 동일한 계수들을 공유할 수 있다. 2016년 2월 15일 출원된 미국 가 특허 출원 제 62/295,461 호 및 2016년 4월 19일 출원된 미국 가 특허 출원 제 62/324,776 호에서 설명된 바와 같이, 정해진 필터들은 또한 인터-코딩된 프레임들에 적용될 수도 있다.

[0126] 비디오 인코더 (20) 및 비디오 디코더 (30) 는 필터 계수들의 시간 예측을 수행하도록 구성될 수도 있다. 필터 계수들의 시간적 예측의 양태들이 이제 설명될 것이다. 참조 픽처들의 ALF 계수들이 저장되고 현재 픽처의 ALF 계수들로서 재사용되도록 허용된다. 현재 픽처는 참조 픽처들에 대해 저장된 ALF 계수들을 사용하기 위해 선택할 수도 있고, ALF 계수들 시그널링을 바이패스할 수도 있다. 이 경우에, 참조 픽처들 중 하나에 대한 인덱스만이 시그널링되고, 표시된 참조 픽처의 저장된 ALF 계수들은 단순히 현재 픽처에 대해 상속된다. 시간 예측의 사용을 나타내기 위해, 인덱스를 전송하기 전에 하나의 플래그가 먼저 코딩된다.

[0127] HEVC/JEM 의 설계는 몇가지 잠재적인 문제점들을 가질 수도 있다. 잠재적인 문제점의 하나의 예로서, SAO 는 오직 분류에 대한 1-D 방향 정보만을 취하고, 이는 코딩 성능을 제한할 수도 있다. 잠재적인 문제점의 다른 예로서, HEVC 에서의 SAO 의 EO 는 패턴 및 카테고리를 포함하는 2 레벨 시그널링을 이용한다. HEVC 에서의 SAO 에서, 패턴 내의 각각의 카테고리에 대해, 이웃 샘플들과 현재 샘플 사이에 얼마나 차이가 먼지를 고려함이 없이 샘플들에 오직 하나의 오프셋이 부가될 수도 있다. 잠재적인 문제점의 또 다른 예로서, SAO 파라미터들은 좌측/상부 방향에 위치되지 않은 2 개의 블록들에 대해 유사할 수 있을 것이라는 것이 관찰되었다. 따라서, 상이한 블록들로부터의 파라미터들의 유연한 병합을 허용하는 것은 추가적인 코딩 이득들을 가져올 수도 있다. 잠재적인 문제점의 또 다른 예로서, 몇몇 인-루프 필터링 방법들은 재구성된 이미지를 수회 프로세싱할 필요성이 있고, 이는 대역폭 및 계산 복잡도를 증가시킬 수도 있다.

[0128] 상기 소개된 문제점들을 잠재적으로 해결하기 위해, 이 개시물은, 피크 SAO 로서 이 개시물에서 지칭된, 새로운 필터링 방법에 대한 기법을 기술하고, 여기서, 하나보다 더 많은 오프셋을 도출하기 위해, 현재 샘플 및 선택된 이웃하는 샘플들의, 단지 차이가 존재하기 보다는, 차이들의 양이 이용될 수도 있다. 또한, 보다 많은 이웃하는 샘플들이 이용될 수도 있고, 분류를 위한 보다 복잡한 기준들이 도입될 수도 있다. 이 개시물에서 기술된 피크 SAO 에 대한 기법들은 (예컨대, EO 및 BO 에 추가하여) 추가적인 SAO 타입으로서 SAO 필터 내로 구현될 수도 있거나, EO 와 같은 기준의 SAO 타입에 대한 대체일 수도 있다. 제안된 기법들은 또한 기존의 SAO 기법들에도 적용가능할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 다음과 같은 기법들을 개별적으로, 또는 달리 진술되지 않는 한, 임의의 조합으로, 적용할 수도 있다.

[0129] 비디오 인코더 (20) 및 비디오 디코더 (30) 는 피크 SAO 필터링을 수행하도록 구성될 수도 있다. 피크 SAO 의 구조의 양태들이 이제 설명될 것이다. 하나의 예에 따르면, 피크 SAO 정보의 3 개의 카테고리들이, 예를 들어, 3 개의 신팩스 엘리먼트들, 또는 명시적 시그널링 및 명시적 시그널링을 필요로 하지 않는 도출 기법들의 조합 중 어느 일방을 이용하여, 비디오 인코더 (20) 로부터 비디오 디코더 (30) 로 시그널링될 수도 있다. 피크 SAO 정보의 제 1 카테고리는 타입 정보이다. 타입은 분류를 위해 사용될 수도 있고, 각 타입은 하나 이상의 연관된 그룹들 (피크 SAO 정보의 제 2 카테고리) 을 가질 수도 있다. 또한, 타입 내의 각각의 그룹에 대해, 하나 이상의 클래스들 (피크 SAO 정보의 제 3 카테고리) 이 정의될 수도 있다.

[0130] 타입들은 이웃하는 샘플들의 선택 및/또는 별도의 또는 공동의 시그널링의 규칙(들)을 정의한다. 예를 들어, 타입은, 필터 파라미터들의 선택을 위해 사용되는 이웃하는 샘플들의 템플릿 (예컨대, 오프셋 값들) 및 이웃하는 샘플들 중 얼마나 많은지를 정의할 수도 있다. 예들로서, 제 1 타입은 필터 파라미터들을 결정하기 위해 사용되어야 하는 4 개의 이웃하는 샘플들을 명시할 수도 있고; 제 2 타입은 필터 파라미터들을 결정하기 위해 사용되어야 하는 4 개의 상이한 이웃하는 샘플들을 명시할 수도 있으며; 제 3 타입은 필터 파라미터들을 결정하기 위해 사용되어야 하는 8 개의 이웃하는 샘플들을 명시할 수도 있다. 그룹은 필터링되어야 하는 샘플들의 규칙을 정의하고, 및/또는, 템플릿에서 이웃하는 샘플들을 선택하기 위한 규칙(들)을 정의한다. 일부 예들에서, 샘플이 어느 그룹으로도 분류되지 않는 경우에, 샘플은 필터링되지 않는다. 클래스는 샘플들을 수정하기 위해 어느 오프셋이 사용되어야하는지를 나타낸다. 예를 들어, 현재 샘플과 선택된 이웃하는 샘플들 사이의 샘플 차이들은 클래스 인덱스를 도출하기 위해 평가되고 사용될 수도 있다. 상이한 클래스는 그 자신의 오프셋 값을 가질 수도 있다.

[0131] 타입 정의의 양태들이 이제 보다 자세히 설명될 것이다. 타입은 필터 파라미터들의 선택을 위해 이웃하는 샘플들 중 어느 것이 사용되어야 하는지를 말한다. 하나의 예에 따르면, HEVC SAO 에서와 같이 1-D 방향에 따라 2 개의 공간적 이웃하는 샘플들을 항상 이용하는 대신에, 공간적, 및/또는 시간적 이웃들을 포함하는 보다 많은 이웃하는 샘플들이 타입, 및/또는 그룹 및/또는 클래스를 결정하기 위해 이용될 수도 있다. 이웃하는 샘플들의 정의는 또한, HEVC 에서 SAO 에 적용가능할 수도 있다. N 은 타입 결정을 위해 이용되는 이웃하는 샘플들의 수를 표시한다. N 은 포지티브 값이고, 예를 들어, 2 보다 더 클 수도 있다.

[0132] 다른 예에서, 이웃하는 샘플은 둘러싸는 샘플들의 가중된 평균과 같이, 이웃하는 샘플의 수개의 둘러싸는 샘플들에 의해 도출되는 값에 의해 표현될 수도 있다.

[0133] 또 다른 예에서, 이웃하는 샘플들은 템플릿 매칭 프로세스를 이용하여 주어진 픽처 구역 (검색 윈도우) 으로부터 선택될 수도 있다. 이들 샘플들을 식별하기 위해, 주어진 픽처 구역에서의 각 픽셀 또는 픽셀들의 선택된 셋트에 대해, 그것의 둘러싸는 샘플들은 현재 픽셀의 둘러싸는 샘플들에 대해 비교되고, 현재 샘플에 비해 유사한 (더 작은 차이의) 둘러싸는 샘플들을 갖는 그들 픽셀들이 이웃하는 샘플들로서 선택된다. 예를 들어, 도 4 에서,  $s_0$  및  $s_x$  의 이웃하는 샘플들은 현재 픽셀 X 에 대해,

$$SAD_0 = \sum_{i=0}^7 |D_0^i - C_i|$$

$$SAD_x = \sum_{i=0}^7 |D_x^i - C_i|$$

[0135] [0136]로서 비교된다. SAD 값들에 기초하여, 최소 SAD 값들 (예컨대,  $s_0$ ) 을 갖는 N (예컨대, 4 또는 8) 개의 샘플들이 그 다음에 이웃하는 샘플들로서 식별되고 사용된다.

[0137] 본 개시의 기법들에 따르면, 제 1 샘플 및 제 2 샘플에 대한 오프셋들을 결정하기 위해, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 현재 블록의 제 1 샘플을 이웃 샘플들의 제 1 셋트에 대해 비교하고, 현재 블록의 제 2 샘플을 이웃 샘플들의 제 2 셋트에 대해 비교할 수도 있고, 이웃 샘플들의 제 1 셋트는 이웃 샘플들의 제 2 셋트가 제 2 샘플에 대하여 가지는 것과 동일한 제 1 샘플에 대한 패턴을 갖는다. 도 4 를 참조하면, 픽셀들 C1, C3, C4, 및 C6 은, 픽셀들 DX1, DX3, DX4, 및 DX6 이 픽셀 SX 에 대하여 가지는 것과 동일한 픽셀 X 에 대한 패턴을 갖는다. 또 다른 예로서, 픽셀들 C2 및 C5 는, 픽셀들 DX2-DX5 이 픽셀 SX 에 대하여 가지는 것과 동일한 픽셀 X 에 대한 패턴을 갖는다. 또 다른 예로서, 픽셀들 C1-C8 은, 픽셀들 DX1-DX8 이 픽셀 SX 에 대하여 가지는 것과 동일한 픽셀 X 에 대한 패턴을 갖는다.

[0138] 하나의 예에서, N 은 4 또는 8 로 설정될 수도 있다. 다른 예에서, 상이한 타입 결정은 상이한 수의 이웃하는 샘플들을 이용할 수도 있다. 또 다른 예에서, 하나의 타입 내에서도, 상이한 그룹들은 상이한 이웃하는 샘플들을 이용할 수도 있다, 예컨대, N 은 상이할 수도 있다. 또 다른 예에서, 하나의 그룹 내에서도, 상이한 클래스들은 상이한 이웃하는 샘플들을 이용할 수도 있다, 예컨대, N 은 상이할 수도 있다.

[0139] 이웃하는 샘플들이 이용되는 상대적인 포지션들은 각 타입에 대해 사전-정의되고 정해질 수도 있다. 예를 들어, 4 와 동일한 N 에 대해, 상대적인 포지션들은, ( $i$  가 0..3 인  $C_i$  에 의해 표시된) 도 5 에서 도시된 바와

같이 좌측, 상부, 우측 및 하부이고, 한편, 8 과 동일한 N에 대해, 공간적 둘러싸는 샘플들이 이용된다. 다른 예들에서, 타입에 대한 이웃하는 샘플들의 포지션들은 슬라이스 내에서 적응적일 수도 있다. 예컨대, 그것은 인트라 예측 모드/변환 매트릭스/슬라이스 타입에 의존할 수도 있다. 다른 예들에서, 상대적인 포지션들의 표시들은 시퀀스 파라미터 셋트/픽처 파라미터 셋트/슬라이스 헤더/적응 파라미터 셋트에서 시그널링될 수도 있다.

[0140] 도 5는 이웃 샘플들  $C_i$  및 현재 샘플 C의 일 예를 나타낸다. 또 다른 예에서, 타입, 그룹 및/또는 클래스들을 결정하기 위해 이용되는 이웃하는 샘플들의 상대적인 포지션들은 도 5에서 도시된 바와 같은 예들에 한정되지 아니한다. 상대적인 포지션들은 임의의 사전-정의된 공간적 및/또는 시간적 이웃하는 샘플들일 수 있다.

[0141] 비디오 인코더 (20)는 얼마나 많은 이웃하는 샘플들이 필터링을 위해 사용되는지의 표시를 비디오 디코더 (30)에 시그널링할 수도 있다. 또 또 다른 예에서, 얼마나 많은 이웃하는 샘플들이 필터링을 위해 사용되는지의 최대 수의 표시가 시그널링될 수도 있다. 또 다른 예에서, 더욱이, 타입 내에 얼마나 많은 그룹들이 있는지의 표시가 추가로 시그널링될 수도 있다. 또 다른 예에서, 타입의 인덱스가 시그널링될 수도 있고, 이는 K에 의해 표시된 타입 내의 그룹 수 및 수 (및/또는 이웃하는 샘플들의 포지션들) 양자를 암시할 수도 있다. 이 경우에, 그룹 수들을 별도로 시그널링할 필요가 없다. 하나의 예에서, 타입 인덱스, N 및 K 사이의 관계는 아래 표 II에서 주어진다.

표 6

표 II

타입의 인덱스	N (필터 파라미터 선택을 위한 이웃하는 샘플들의 수)	K (주어진 타입 인덱스에 대한 그룹들의 수)
1	4	1
2	4	2
3	8	4
4	4 또는 8	5

[0143] 그룹 정의의 양태들이 이제 설명될 것이다. 예를 들어, 현재 샘플보다 더 큰 값을 갖는 이웃하는 샘플들의 수 (또는 대표 이웃하는 샘플)를  $C_{lgr}(N)$ 에 의해 표시하고, 현재 샘플보다 더 작은 값을 갖는 이웃하는 샘플들의 수를  $C_{sm1}(N)$ 에 의해 표시하자. 그룹 정의를 위한 기준들은  $C_{lgr}(N)$  및/또는  $C_{sm1}(N)$ 의 값에 의존할 수도 있다.

[0144] 또 다른 예에서, 그것은 함수  $f(C_{lgr}(N), C_{sm1}(N))$ 에 의존할 수도 있고, 예컨대,  $f$ 는 최대 함수로서 정의된다.  $C_{lgr}(N)$ 이  $C_{sm1}(N)$  보다 더 클 때, 그것은 ‘라지 트렌드 (large trend)’로서 명명된다.  $C_{lgr}(N)$ 이  $C_{sm1}(N)$  보다 더 작을 때, 그것은 ‘스몰 트렌드 (small trend)’로서 명명된다. 상이한 포지션들에 위치된 2개의 샘플들에 대해, 하나가 라지 트렌드에 속하고 다른 하나는 스몰 트렌드에 속하지만, 함수의 리턴되는 값들은 동일한 경우에 (일 예가 도 6a 및 도 6b에 도시되고 이하에서 설명된다), 다음이 적용될 수도 있다. 하나의 예에서, 2개의 샘플들은 동일한 그룹 내인 것으로 분류될 수도 있다. 이 경우에, 더욱이, 클래스 인덱스들이 동일한 경우에, 대칭적 필터링 프로세스가 2개의 트렌드들에 대해 적용되고, 여기서, 오프셋은 동일한 한편, 라지 트렌드에 대해, 수정된 샘플 값이 원래의 값 플러스 오프셋으로 설정되고, 스몰 트렌드에 대해, 수정된 샘플 값이 원래의 값 마이너스 오프셋으로 설정된다. 또 다른 예에서, 2개의 샘플들은 2개의 그룹들/클래스들로 분류될 수도 있고, 각 그룹/클래스는 상이한 오프셋을 가질 수도 있다. 이 경우에, 필터링 프로세스는 비대칭적이다.

[0145] 도 6a 및 도 6b는 8-선택된 이웃하는 샘플들을 갖는 2개의 샘플들의 예들을 나타낸다. 도 6a는 50과 동일한 현재 샘플 값 및  $C_{lgr}(N) = 6$ 을 갖는 ‘라지 트렌드’의 일 예를 나타낸다. 도 6b는 93과 동일한 현재 샘플 값 및  $C_{sm1}(N) = 6$ 을 갖는 ‘스몰 트렌드’의 일 예를 나타낸다.

[0146] 하나의 예에서, 그룹 K의 총 수에 의존하여, 수개의 임계치들 (올림 차순으로  $T_0, T_1, \dots, T_K$ )이 사전-정의될

수도 있다. 다른 예들에서, 임계치들은 온더플라이 (on-the-fly)로 시그널링되거나 도출될 수도 있다. 그룹 인덱스는  $C_{lgr}(N)$ ,  $C_{sm}(N)$ 의 값을 임계 값에 대해 비교함으로써 도출될 수도 있다.

[0147] 샘플들이 다음과 같은 조건들 중 하나를 만족시키는 경우에 그것은  $i$ -번째 그룹으로 분류될 것이다:

[0148] -  $f(\bullet)$  는  $T_i$  보다 더 크고,  $f(\bullet)$  는  $T_{i+1}$  보다 더 크지 않음.

[0149] 반면에,  $f(\bullet)$  가  $T_0$  보다 더 작거나 동일한 경우, 샘플은 분류되지 않는다, 즉, 아무런 필터링 동작도 이 샘플에 적용되지 않을 것이다.

[0150] 하나의 예에서, 임계치들은 아래 표 III에서 나타낸 바와 같이 정의된다.

표 7

표 III

타입의 인덱스	N (필터 파라미터 선택을 위한 이웃하는 샘플들의 수)	K (주어진 타입 인덱스에 대한 그룹들의 수)	임계치들
1	4	1	$T_0 = 2, T_1 = 4$
2	4	2	$T_0 = 2, T_1 = 3, T_2 = 4$
3	8	4	$T_0 = 4, T_1 = 5, T_2 = 6, T_3 = 7, T_4 = 8$
4	4 또는 8	5	

[0152] 하나의 예에서, 더욱이, 그룹 결정은 상이한  $N_s$ 에 대한 (스몰/라지) 트렌드에 의존할 수도 있다. 하나의 예에서, 2 개의 상이한  $N_s$ 에 대해, 트렌드들이 상이한 경우, 상이한 임계치들이 이용될 수도 있다. 다른 예에서, 그것은  $f(\bullet)$ 에 추가로 의존할 수도 있다. 4 와 동일한 타입에 대한 일 예가 아래 표 IV에서 나타낸 바와 같이 주어진다.

표 8

표 IV

타입	K	임계치들
4	5	<p>그룹 0:</p> $!(f(C_{lgr}(4), C_{sm}(4)) > M \&\& \text{sign}(C_{lgr}(4)-C_{sm}(4)) == \text{sign}(C_{lgr}(8)-C_{sm}(8)))$ $\&\& f(C_{lgr}(8), C_{sm}(8)) > T_0 \&\& f(C_{lgr}(8), C_{sm}(8)) \leq T_1$ <p>하나의 예에서, <math>T_0 = 4, T_1 = 7</math>;</p> <p><math>i</math>-번째 그룹 (<math>i</math> 는 <math>1 \dots 4</math>):</p> $f(C_{lgr}(4), C_{sm}(4)) > M \&\& \text{sign}(C_{lgr}(4)-C_{sm}(4)) == \text{sign}(C_{lgr}(8)-C_{sm}(8))$ $\&\& f(C_{lgr}(8), C_{sm}(8)) > T_i \&\& f(C_{lgr}(8), C_{sm}(8)) \leq T_{i+1}$ <p>하나의 예에서, <math>T_1 = 4, T_2 = 5, T_3 = 6, T_4 = 7, T_5 = 8</math></p>

[0154] 또 다른 예에서, 4 와 동일한 타입에 대해, 6 개의 그룹들이 존재할 수도 있고, 각 그룹의 정의들은 아래 표 V에서 나타낸 바와 같이 정의될 수도 있다.

표 9

[0155]

표 V

타입	K	임계치들
4	6	<p>i-번째 그룹 (i 는 0/1):  <math>!(f(C_{lgr}(4), C_{sm1}(4)) &gt; M \&amp;\&amp; \text{sign}(C_{lgr}(4)-C_{sm1}(4)) == \text{sign}(C_{lgr}(8)-C_{sm1}(8)))</math>  <math>\&amp;\&amp; f(C_{lgr}(8), C_{sm1}(8)) &gt; T_i \&amp;\&amp; f(C_{lgr}(8), C_{sm1}(8)) \leq T_{i+1}</math></p> <p>하나의 예에서, <math>T_0 = 4, T_1 = 5, T_2 = 8;</math></p> <p>i-번째 그룹 (i 는 2...5):  <math>f(C_{lgr}(4), C_{sm1}(4)) &gt; M \&amp;\&amp; \text{sign}(C_{lgr}(4)-C_{sm1}(4)) == \text{sign}(C_{lgr}(8)-C_{sm1}(8))</math>  <math>\&amp;\&amp; f(C_{lgr}(8), C_{sm1}(8)) &gt; T_i \&amp;\&amp; f(C_{lgr}(8), C_{sm1}(8)) \leq T_{i+1}</math></p> <p>하나의 예에서, <math>T_2 = 4, T_3 = 5, T_4 = 6, T_5 = 7, T_6 = 8</math></p>

[0156]

또 다른 예에서, 얼마나 많은 이웃하는 샘플들이 현재 샘플과 동일한지에 관한 정보가 그룹 결정에 대한 고려사항으로 취해질 수도 있다. 또 다른 예에서, 더욱이, 그룹 결정은 CBF (모두 제로 계수들을 나타내기 위한, 코딩된 블록 플래그)/코딩된 모드 (인트라/인터)/변환 매트릭스/예측 방향 (단방향 또는 양방향)/예측 블록/현재 슬라이스의 시간적 레벨에 의존할 수도 있다.

[0157]

하나의 예에서, 어떤 타입에 대해  $N$ 의 다수의 값들이 허용될 때, 타입 4에 대해  $N$ 이 4 또는 8 중 어느 일방일 수도 있는 표 III 및 표 IV에 대해 상술된 바와 같이,  $N$ 이 그 타입 내의 그룹에 대해 사용되어야 하는 것을 말하기 위해 하나 이상의 인덱스들이 시그널링될 수도 있다.

[0158]

비디오 인코더 (20) 및 비디오 디코더 (30)는 클래스 인덱스를 오프셋 값과 연관시킬 수도 있다. 클래스 인덱스 결정에 대한 기준들이 이제 논의될 것이다. 클래스 인덱스들의 결정은 선택된 이웃하는 샘플들과 현재 샘플 사이의 차이들의 평균 값에 의존할 수도 있다. 하나의 예에서, 선택된 이웃하는 샘플들의 수는 타입 정의를 위해 사용된 이웃하는 샘플들의 총 수, 예컨대 상기 언급된  $N$ , 보다 더 크지 않을 수도 있다. 하나의 예에서, 선택된 이웃하는 샘플들은, 그것이 ‘라지 (또는 스몰) 트렌드일 때, 현재 샘플보다 더 커야 (또는 더 작아야) 한다. 선택된 샘플들을  $C_{s0}, C_{s1}, \dots, C_{sM-1}$ 인 것으로 표시하면, 클래스 인덱스는

$$(\sum_{m=0}^{M-1} (|C_{sm} - C|)) / M$$

로서 도출된다.  $M$ 은 비-제로 포지티브 수이다. 또 다른 예에서, 더욱이, 모든 선택된 샘플들 중에서 최대 및 최소 차이들을 갖는 샘플들이 추가로 제거된다, 즉, 클래스 인덱스의 결정에서 고려되지 않는다. 또 다른 예에서, 선택된 샘플들의 수는  $(\text{pow}(2, x))$ 와 동일할 것이어서, 나누기 연산이 시프트에 의해 대체될 수 있을 것이다. 또 다른 예에서, 오직 하나의 샘플이 선택될 수도 있다.

[0159]

또 다른 예에서, 클래스 인덱스는 선택된 이웃하는 샘플들과 현재 샘플 사이의 값 차이들의 합계에 의존할 수도 있다. 또 다른 예에서, 더욱이, 클래스 인덱스는 정규화 팩터 (normalization factor)에 의존할 수도 있다. 하나의 예에서, 클래스 인덱스는 다음과 같이 도출될 수도 있다.

$$(\sum_{m=0}^{M-1} (|C_{sm} - C|)) / M / NF$$

[0160]

여기서,  $NF$ 는 정규화 팩터를 나타낸다. 또 다른 예에서, 정규화 팩터는  $(\text{pow}(2, x))$ 와 동일할 수도 있고, 따라서, 클래스 인덱스는,

$$\frac{(\sum_{m=0}^{M-1} (|C_{sm} - C|)) / M}{(1 \ll NF)}$$

[0162]

로서 도출될 수도 있다. 하나의 예에서, 정규화 팩터는 타입 인덱스/그룹 인덱스/ $N$ 에 의존할 수도 있다. 정규화 팩터는 각 그룹에 대해 시그널링될 수도 있다. 다른 예들에서, 모든 그룹들이 동일한 정규화 팩터를 공유할 수도 있고, 여기서, 정규화 팩터가 공유되는지 또는 아닌지 여부를 나타내기 위해 플래그가 시그널링될 수도 있다.

[0164]

허용된 클래스들의 최대 수가 시그널링될 수도 있다. 하나의 예에서, 이 수는 모든 타입들/그룹들에 의해 공유된다. 다른 예들에서, 이 수는 온더플라이로 시그널링되거나 관련될 수도 있다. 또 다른 예에서,

더욱이, 도출된 클래스 인덱스가 허용된 클래스들의 최대 수를 초과할 때, 이들 샘플들은 함께 병합된다. 하나의 예에서, 클래스 인덱스는 다음과 같이 도출될 수도 있다:

$$[0165] \min(\maxNr - \text{IdxOffset}, \frac{(\sum_{m=0}^{M-1} (|C_{sm} - C|)) / M}{(1 \ll NF)})$$

[0166] 또는

$$[0167] \min(\maxNr - \text{IdxOffset}, \left( \sum_{m=0}^{M-1} (|C_{sm} - C|) \right) \gg NF)$$

[0168] 여기서,  $\text{IdxOffset}$  은 0 또는 1 과 동일할 수도 있다.

[0169] 비디오 인코더 (20)는 비디오 디코더 (30)에 오프셋 값들을 시그널링할 수도 있다. 피크 SAO에서의 오프셋 값들의 시그널링이 이제 논의될 것이다. 라지 트렌드 및 스몰 트렌드에 대해, 대응하는 포지티브 (positive) 및 네거티브 (negative) 오프셋은 동일한 크기를 공유할 수도 있다. 다른 예들에서, 상이한 오프셋 값들이 시그널링될 수도 있다. 오프셋들은 바이패스 코딩으로 코딩될 수도 있다. 다른 예들에서, 오프셋들은 컨텍스트 모델들로 코딩될 수도 있다. 단항/절단된 단항/고정-길이 이진화 방법이 오프셋의 크기를 코딩하기 위해 사용될 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30)는 동일한 그룹에서 (이웃하는 클래스 인덱스와 연관된) 하나의 오프셋과 그것의 이전에 코딩된 값 사이의 차이를 코딩함으로써 예측 코딩을 이용할 수도 있다. 더욱이, 하나의 플래그는 이 방법이 인에이블되는지 또는 아닌지 여부를 나타내기 위한 것이다. 상이한 그룹에서 (선택된 클래스 인덱스와 연관된) 하나의 오프셋과 그것의 이전에 코딩된 값 사이의 차이를 코딩함으로써 예측 코딩이 이용될 수도 있다. 하나의 예에서, 오프셋은 이전에 코딩된 슬라이스들에서 사용된 오프셋 값들로부터 상속되거나 예측될 수도 있다. 하나의 예에서, 오프셋 값들의 수개의 셋트들이 저장될 수도 있다. 그 셋트의 하나의 인덱스가 추가로 시그널링될 수도 있다. 하나의 예에서, 오프셋들은 시그널링 없이 온더플라이로 도출될 수도 있다. 하나의 예에서, 선택된 이웃하는 샘플들의 평균이 먼저 계산될 수도 있다. 오프셋은  $\min(\text{average}, \text{Thresh})$ 로 설정될 수도 있고, 여기서, 변수  $\text{Thresh}$ 는 사전-정의될 수도 있다. 고 비트-심도 입력에 대해, 정해진 시프트 값이 SPS/PPS/APS/슬라이스 헤더에서 사전-정의되거나 시그널링될 수도 있고,  $L$ 에 의해 표시된다. 이 경우에, 최종 오프셋은 (시그널링된/도출된 오프셋  $\ll L$ )과 동일하게 설정된다.

[0170] 비디오 인코더 (20) 및 비디오 디코더 (30)가 어떻게 피크 SAO를 적용할 수도 있는지에 관한 양태들이 이제 논의될 것이다. 하나의 예에서, 제안된 피크 SAO 방법은 오직 루마 컴포넌트에만 적용될 수도 있다. 다른 예들에서, 더욱이, 어떤 픽처/슬라이스/타일에서 루마 컴포넌트에 대해 피크 SAO가 인에이블되는지 또는 디스에이블되는지 여부를 나타내기 위해 하나의 플래그가 시그널링될 수도 있다. 또 다른 예에서, 안된 피크 SAO 방법은 모든 컬러 컴포넌트들에 대해 적용될 수도 있다. 하나의 예에서, 그것은 모든 3 개의 컬러 컴포넌트들에 대해 호출되거나 디스에이블될 수도 있다. 더욱이 어떤 픽처/슬라이스/타일/블록에서 모든 3 개의 컬러 컴포넌트들에 대해 피크 SAO가 인에이블되는지 또는 디스에이블되는지 여부를 나타내기 위해 하나의 플래그가 시그널링될 수도 있다. 또 다른 예에서, 각각의 컬러 컴포넌트는 피크 SAO를 인에이블/디스에이블하는 그 자신의 제어를 가질 수도 있다. 더욱이 어떤 픽처/슬라이스/타일/블록에서 컴포넌트에 대해 피크 SAO가 인에이블되는지 또는 디스에이블되는지 여부를 나타내기 위해 각 컴포넌트에 대해 하나의 플래그가 시그널링될 수도 있다. 또 다른 예에서, 루마 및 크로마 컴포넌트들에 대한 별도의 제어들이 사용될 수도 있다. 더욱이 어떤 픽처/슬라이스/타일/블록에서 컴포넌트에 대해 피크 SAO가 인에이블되는지 또는 디스에이블되는지 여부를 나타내기 위해 2 개의 크로마 컴포넌트들에 대해 하나의 플래그 및 루마 컴포넌트에 대해 하나의 플래그가 시그널링될 수도 있다. 상기 경우들에서, 하나/그보다 많은 크로마 컴포넌트들에 대해 피크 SAO를 인에이블하는 것의 표시들의 시그널링은 루마 컴포넌트에 대한 것에 의존할 수도 있다. 하나의 예에서, 오직 피크 SAO가 하나의 픽처/슬라이스/타일/블록의 루마 컴포넌트에 대해 적용될 때에만, 크로마 컴포넌트들에 대한 표시들이 시그널링될 수도 있다. 그렇지 않은 경우, 피크 SAO는 크로마 컴포넌트들에 대해 적용되지 않을 것이다.

[0171] 하나의 예에서, 특정 컬러 컴포넌트에 대해, 각각의 코딩 블록 트리/코딩 유닛/구역에 대해 피크 SAO가 적용되는지 또는 아닌지 여부를 나타내기 위해 온/오프 플래그들을 비디오 인코더 (20)는 인코딩할 수도 있고 비디오 디코더 (30)는 디코딩할 수도 있다.

[0172] 상술된 시스템 (10)과 같은 비디오 코딩 시스템에서의 피크 SAO의 사용이 이제 설명될 것이다. 설명되는

기법들은 비디오 코딩 시스템의 포스트-프로세싱, 즉, 아웃-루프로서 사용될 수도 있다. 다른 예에서, 제안된 방법은 하나의 별도의 인-루프 필터링 코딩 툴로서 사용될 수도 있다. 하나의 예에서, 그것은 DBF, SAO, 적용형 루프 필터 (ALF) 와 같은 모든 다른 필터링 코딩 툴들 전에 호출될 수도 있다. 다른 예에서, 그것은 DB 후에 그리고 SAO 전에, 또는 SAO 후에 하지만 ALF 전에와 같이, 필터링 코딩 툴들 사이에서 호출될 수도 있다. 또 다른 예에서, 그것은 ALF 후에와 같이, 모든 필터링 코딩 툴들 후에 호출될 수도 있다. 또 또 다른 예에서, 피크 SAO 는 하나의 블록의 예측 블록에 적용될 수도 있다, 예컨대, 예측 블록은 모션-보상된 블록이거나 인트라 예측으로부터 도출된다.

[0173] 또 다른 예에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는, SAO 의 HEVC 버전 또는 SAO 의 다른 버전을 포함하는, 다른 인-루프 필터링 코딩 툴(들)과 함께 본 개시의 피크 SAO 기법들을 구현할 수도 있다. 또 다른 예에서, 더욱이, 피크 SAO 및 공동으로 사용되는 인-루프 필터링 툴 (예컨대, SAO) 에 대한 입력들은 동일할 것이다. 또 다른 예에서, 더욱이, SAO 의 타입 또는 EO 의 타입들은 피크 SAO 의 사용을 커버하도록 확장될 수 있을 것이다.

[0174] SAO 에 관련된 기법들이 이제 설명될 것이다. 반대 조건들로 정의되는 2 개의 카테고리들에 대해, 2 개의 오프셋들 (예컨대, 하나는 포지티브이고 하나는 네거티브) 의 크기들은 동일하도록 강제된다. 따라서, 2 개 대신 오직 하나의 오프셋이 시그널링될 필요성이 있을 수도 있다. 하나의 예에서, EO 카테고리 1 에 대한 포지티브 오프셋들 및 EO 카테고리 4 에 대한 네거티브 오프셋들은 동일하도록 강제된다. 하나의 예에서, EO 카테고리 2 에 대한 포지티브 오프셋들 및 EO 카테고리 3 에 대한 네거티브 오프셋들은 동일하도록 강제된다.

[0175] 반대 조건들로 정의되는 2 개의 카테고리들에 대해, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 제 2 오프셋 (포지티브 또는 네거티브) 의 크기를 예측적으로 코딩할 수도 있다. 따라서, 2 개의 크기들의 차이만이 제 2 오프셋에 대해 코딩될 수도 있다. 또 다른 예에서, 오직 CTU 의 공간적 좌측 또는 상부로부터 SAO 파라미터들의 병합을 허용하는 대신에, 현재 블록의 파라미터들이 상부-좌측, 상부-우측 블록과 같이 다른 이웃하는 블록들로부터 상속될 수 있을 것임이 제안되었다. 또 다른 예에서, 더욱이, 하나의 블록의 파라미터들은 이웃하는 블록이 아닌 블록으로부터 카피될 수 있을 것이다. 또 다른 예에서, 하나의 블록의 파라미터들은 상이한 픽처에서 위치하는 블록으로부터 카피될 수 있을 것이다.

[0176] 본원에 설명된 기법들은 메모리 대역폭 절약을 달성할 수도 있다. 다수의 인-루프 필터링 방법들이 적용될 수도 있을 때, 적어도 2 개의 인-루프 필터링 방법들의 입력이 동일할 수도 있다. 즉, 분류는 2 개 이상의 필터링 방법들에 대해 동일한 입력에 기초한다. 하나의 예에서, DBF 를 호출한 후의 재구성된 이미지는 피크 SAO 및 SAO 양자에 대한 입력으로서 사용될 수도 있다. 하나의 예에서, 피크 SAO 를 호출한 후의 재구성된 이미지는 SAO 및 ALF/GALF 양자에 대한 입력으로서 사용될 수도 있다. 다른 예에서, 더욱이, 상이한 필터링 방법들의 필터링 프로세스들은 여전히 소정의 순서에 따라 수행될 수도 있다.

[0177] 도 7 은 본 개시에서 설명된 기법들을 구현할 수도 있는 예시적인 비디오 인코더 (20) 를 나타내는 블록도이다. 비디오 인코더 (20) 는 비디오 슬라이스들 내에서 비디오 블록들의 인트라 및 인터 코딩을 수행할 수도 있다. 인트라 코딩은 주어진 비디오 프레임 또는 픽처 내의 비디오에서 공간적 리던던시를 감소시키거나 제거하기 위해 공간 예측에 의존한다. 인터-코딩은 비디오 시퀀스의 인접한 프레임들 또는 픽처들 내 비디오에 있어서 시간 리던던시를 감소 또는 제거하기 위해 시간 예측에 의존한다. 인트라-모드 (I 모드) 는 여러 공간 기반의 압축 모드들 중 임의의 것을 지칭할 수도 있다. 단방향 예측 (P 모드) 또는 양방향 예측 (B 모드) 과 같은 인터-모드들은 수개의 시간 기반 압축 모드들 중 임의의 모드를 지칭할 수도 있다.

[0178] 도 7 의 예에 있어서, 비디오 인코더 (20) 는 비디오 데이터 메모리 (33), 파티셔닝 유닛 (35), 예측 프로세싱 유닛 (41), 합산기 (50), 변환 프로세싱 유닛 (52), 양자화 유닛 (54), 및 엔트로피 인코딩 유닛 (56) 을 포함한다. 예측 프로세싱 유닛 (41) 은 모션 추정 유닛 (MEU) (42), 모션 보상 유닛 (MCU) (44) 및 인트라 예측 유닛 (46) 을 포함한다. 비디오 블록 재구성을 위해, 비디오 인코더 (20) 는 또한 역양자화 유닛 (58), 역변환 프로세싱 유닛 (60), 합산기 (62), 필터 유닛 (64), 및 디코딩된 픽처 버퍼 (DPB) (66) 를 포함한다.

[0179] 도 7 에 도시된 바와 같이, 비디오 인코더 (20) 는 비디오 데이터를 수신하고, 그 수신된 비디오 데이터를 비디오 데이터 메모리 (33) 에 저장한다. 비디오 데이터 메모리 (33) 는 비디오 인코더 (20) 의 컴포넌트들에 의해 인코딩될 비디오 데이터를 저장할 수도 있다. 비디오 데이터 메모리 (33) 에 저장된 비디오 데이터는, 예를 들어, 비디오 소스 (18) 로부터 획득될 수도 있다. DPB (66) 는, 예컨대 인트라 또는 인터 코딩 모드들에서, 비디오 인코더 (20) 에 의해 비디오 데이터를 인코딩함에 있어서 사용하기 위한 참조 비디오

데이터를 저장하는 참조 픽처 메모리일 수도 있다. 비디오 데이터 메모리 (33) 및 DPB (66)는 다양한 메모리 디바이스들, 이를 테면, SDRAM (synchronous DRAM), MRAM (magnetoresistive RAM), RRAM (resistive RAM)을 포함하는 DRAM (Dynamic random access memory), 또는 다른 유형들의 메모리 디바이스들의 어느 것에 의해 형성될 수도 있다. 비디오 데이터 메모리 (33) 및 DPB (66)는 동일한 메모리 디바이스 또는 별도의 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에 있어서, 비디오 데이터 메모리 (33)는 비디오 인코더 (20)의 다른 컴포넌트들과 온-칩형이거나 또는 그들 컴포넌트들에 대하여 오프-칩형일 수도 있다.

파티셔닝 유닛 (35)은 비디오 데이터 메모리 (33)로부터 비디오 데이터를 수신하고, 그 비디오 데이터를 비디오 블록들로 파티셔닝한다. 이 파티셔닝은 또한, 슬라이스 (slice) 들, 타일 (tile) 들, 또는 다른 더 큰 유닛들로의 파티셔닝뿐만 아니라, 예컨대, LCU 들 및 CU 들의 큐드트리 구조에 따른 비디오 블록 파티셔닝을 포함할 수도 있다. 비디오 인코더 (20)는 인코딩되어야 할 비디오 슬라이스 내의 비디오 블록들을 인코딩하는 컴포넌트들을 일반적으로 예시한다. 슬라이스는 다수의 비디오 블록들로 (그리고 아마도 타일들로서) 지정되는 비디오 블록들의 셋트들로) 분할될 수도 있다. 예측 프로세싱 유닛 (41)은 여러 결과들 (예를 들어, 코딩 레이트 및 왜곡의 레벨)에 기초하여 현재의 비디오 블록에 대해 복수의 인트라 코딩 모드들 중 하나 또는 복수의 인터 코딩 모드들 중 하나와 같이 복수의 가능한 코딩 모드들 중 하나를 선택할 수도 있다. 예측 프로세싱 유닛 (41)은 결과적인 인트라- 또는 인터-코딩된 블록을 합산기 (50)에 제공하여 잔차 블록 데이터를 생성하고, 합산기 (62)에 제공하여 참조 픽처로서의 사용을 위한 인코딩된 블록을 재구성할 수도 있다.

예측 프로세싱 유닛 (41) 내의 인트라 예측 유닛 (46)은 공간 압축을 제공하기 위해, 코딩될 현재 블록과 동일한 프레임 또는 슬라이스에서의 하나 이상의 이웃 블록들에 관해 현재 비디오 블록의 인트라-예측 코딩을 수행할 수도 있다. 예측 프로세싱 유닛 (41) 내의 모션 추정 유닛 (42) 및 모션 보상 유닛 (44)은 시간 압축을 제공하기 위해 하나 이상의 참조 픽처들에서의 하나 이상의 예측 블록들에 관해 현재 비디오 블록의 인터-예측 코딩을 수행한다.

모션 추정 유닛 (42)은 비디오 시퀀스에 대한 미리결정된 패턴에 따라 비디오 슬라이스에 대해 인터 예측 모드를 결정하도록 구성될 수도 있다. 미리결정된 패턴은 비디오 슬라이스들을 P 슬라이스들 또는 B 슬라이스들로서 시퀀스로 지정할 수도 있다. 모션 추정 유닛 (42) 및 모션 보상 유닛 (44)은 고도로 집적화될 수도 있지만, 개념적 목적들을 위해 별개로 예시되어 있다. 모션 추정 유닛 (42)에 의해 수행된 모션 추정은 비디오 블록들을 위한 모션을 추정하는 모션 벡터들을 생성하는 프로세스이다. 모션 벡터는, 예를 들어, 참조픽처 내에서의 예측 블록에 관한 현재 비디오 프레임 또는 픽처 내의 비디오 블록의 PU의 변위를 표시할 수도 있다.

예측 블록은 픽셀 차이의 관점에서 코딩될 비디오 블록의 PU 와 밀접하게 매칭되도록 발견되는 블록이며, 이 픽셀 차이는 절대 차이의 합 (SAD), 제곱 차이의 합 (SSD), 또는 다른 차이 메트릭들에 의해 결정될 수도 있다.

일부 예들에 있어서, 비디오 인코더 (20)는 DPB (66)에 저장된 참조 픽처들의 서브-정수 픽셀 포지션들에 대한 값을 계산할 수도 있다. 예를 들어, 비디오 인코더 (20)는 참조 픽처의 1/4 픽셀 포지션들, 1/8 픽셀 포지션들, 또는 다른 분수의 픽셀 포지션들의 값을 보간할 수도 있다. 따라서, 모션 추정 유닛 (42)은 풀 픽셀 포지션들 및 분수 픽셀 포지션들에 관한 모션 탐색을 수행하고, 분수 픽셀 정밀도로 모션 벡터를 출력할 수도 있다.

모션 추정 유닛 (42)은 인터-코딩된 슬라이스에서의 비디오 블록의 PU에 대한 모션 벡터를, 그 PU의 포지션을 참조 픽처의 예측 블록의 포지션과 비교함으로써 계산한다. 참조 픽처는 제 1 참조 픽처 리스트 (List 0) 또는 제 2 참조 픽처 리스트 (List 1)로부터 선택될 수도 있고, 이들의 각각은 DPB (66) 내에 저장된 하나 이상의 참조 픽처들을 식별한다. 모션 추정 유닛 (42)은 계산된 모션 벡터를 엔트로피 인코딩 유닛 (56)과 모션 보상 유닛 (44)으로 전송한다.

모션 보상 유닛 (44)에 의해 수행된 모션 보상은, 모션 추정에 의해 결정된 모션 벡터에 기초하여 예측 블록을 폐칭 또는 생성하는 것을, 가능하게는 서브-픽셀 정밀도에 대한 보간들을 수행하는 것을 수반할 수도 있다. 현재 비디오 블록의 PU에 대한 모션 벡터를 수신할 시, 모션 보상 유닛 (44)은, 모션 벡터가 참조 픽처 리스트들 중 하나에 포인팅하는 예측 블록을 로케이팅할 수도 있다. 비디오 인코더 (20)는, 코딩되고 있는 현재 비디오 블록의 픽셀 값들로부터 예측 블록의 픽셀 값들을 감산하여 픽셀 차이 값들을 형성함으로써 잔차 비디오 블록을 형성한다. 픽셀 차이 값들은 블록에 대한 잔차 데이터를 형성하고, 루마 및 크로마 차이 컴포넌트들 양자를 포함할 수도 있다. 합산기 (50)는 이러한 감산 연산을 수행하는 컴포넌트 또는 컴포넌트들

을 나타낸다. 모션 보상 유닛 (44)은 또한, 비디오 슬라이스의 비디오 블록들을 디코딩함에 있어서 비디오 디코더 (30)에 의한 사용을 위해 비디오 블록들 및 비디오 슬라이스와 연관된 신택스 엘리먼트들을 생성할 수도 있다.

[0186] 예측 프로세싱 유닛 (41)이 인터-예측 또는 인트라-예측 중 어느 하나를 통해 현재 비디오 블록에 대한 예측 블록을 생성한 이후, 비디오 인코더 (20)는 현재 비디오 블록으로부터 예측 블록을 감산함으로써 잔차 비디오 블록을 형성한다. 잔차 블록에서의 잔차 비디오 데이터는 하나 이상의 TU 들 내에 포함될 수도 있고 변환 프로세싱 유닛 (52)에 적용될 수도 있다. 변환 프로세싱 유닛 (52)은 이산 코사인 변환 (discrete cosine transform; DCT) 또는 개념적으로 유사한 변환과 같은 변환을 이용하여 잔차 비디오 데이터를 잔차 변환 계수들로 변환한다. 변환 프로세싱 유닛 (52)은 잔차 비디오 데이터를 픽셀 도메인으로부터, 주파수 도메인과 같은 변환 도메인으로 변환할 수도 있다.

[0187] 변환 프로세싱 유닛 (52)은 결과적인 변환 계수들을 양자화 유닛 (54)으로 전송할 수도 있다. 양자화 유닛 (54)은 비트 레이트를 추가로 감소시키기 위해 변환 계수들을 양자화한다. 양자화 프로세스는 계수들의 일부 또는 전부와 연관된 비트 심도를 감소시킬 수도 있다. 양자화도는 양자화 파라미터를 조절함으로써 수 정될 수도 있다. 일부 예들에 있어서, 양자화 유닛 (54)은 그 다음에, 양자화된 변환 계수들을 포함하는 매트릭스의 스캔을 수행할 수도 있다. 또 다른 예에서, 엔트로피 인코딩 유닛 (56)은 스캔을 수행할 수도 있다.

[0188] 양자화에 이어서, 엔트로피 인코딩 유닛 (56)은 양자화된 변환 계수들을 엔트로피 인코딩한다. 예를 들어, 엔트로피 인코딩 유닛 (56)은 컨텍스트 적용 가변 길이 코딩 (CAVLC), 컨텍스트 적용 2 진 산술 코딩 (CABAC), 신택스-기반 컨텍스트-적용 2 진 산술 코딩 (SBAC), 확률 간격 파티셔닝 엔트로피 (PIPE) 코딩 또는 또 다른 엔트로피 인코딩 방법론 또는 기법을 수행할 수도 있다. 엔트로피 인코딩 유닛 (56)에 의한 엔트로피 인코딩에 이어서, 인코딩된 비트스트림은 비디오 디코더 (30)로 송신되거나, 또는 비디오 디코더 (30)에 의한 나중의 송신 또는 취출을 위해 아카이브될 수도 있다. 엔트로피 인코딩 유닛 (56)은 또한, 코딩되고 있는 현재의 비디오 슬라이스에 대한 모션 벡터들 및 다른 신택스 엘리먼트들을 엔트로피 인코딩할 수도 있다.

[0189] 역양자화 유닛 (58) 및 역변환 프로세싱 유닛 (60)은, 각각, 역양자화 및 역변환을 적용하여, 참조 픽처의 참조 블록으로서의 나중의 사용을 위해 픽셀 도메인에서 잔차 블록을 재구성한다. 모션 보상 유닛 (44)은, 참조 픽처 리스트들 중 하나 내에서의 참조 픽처들 중 하나의 참조 픽처의 예측 블록에 잔차 블록을 부가함으로써 참조 블록을 계산할 수도 있다. 모션 보상 유닛 (44)은 또한, 하나 이상의 보간 필터들을 재구성된 잔차 블록에 적용하여, 모션 추정에서의 사용을 위한 서브-정수 픽셀 값들을 계산할 수도 있다. 합산기 (62)는 재구성된 잔차 블록을, 모션 보상 유닛 (44)에 의해 생성된 모션 보상된 예측 블록에 가산하여, 재구성된 블록을 생성한다.

[0190] 필터 유닛 (64)은 재구성된 블록 (예컨대, 합산기 (62)의 출력)을 필터링하고, 그 필터링된 재구성된 블록을 참조 블록으로서의 사용들을 위해 DPB (66)에 저장한다. 참조 블록은, 후속 비디오 프레임 또는 픽처에서의 블록을 인터-예측하기 위해 참조 블록으로서 모션 추정 유닛 (42) 및 모션 보상 유닛 (44)에 의해 사용될 수도 있다. 필터 유닛 (64)은 디블록 필터링, SAO 필터링, 피크 SAO 필터링, ALF, 및/또는 GALF, 및/또는 다른 타입들의 루프 필터들과 같은 임의의 타입의 필터링을 수행할 수도 있다. 디블록 필터는, 예를 들어, 블록 경계들을 필터링하여 재구성된 비디오로부터 블록카니스 아티팩트들을 제거하기 위해 디블록킹 필터링을 적용할 수도 있다. 피크 SAO 필터는 전체적인 코딩 품질을 향상시키기 위해 재구성된 픽셀 값들에 오프셋들을 적용할 수도 있다. 부가적인 루프 필터들 (인 루프 (in loop) 또는 포스트 루프 (post loop))이 또한 사용될 수도 있다.

[0191] 도 8은 본 개시에서 설명된 기법들을 구현할 수도 있는 예시적인 비디오 디코더 (30)를 나타내는 블록도이다. 도 8의 비디오 디코더 (30)는, 예를 들어, 도 7의 비디오 인코더 (20)에 대해 상기 설명된 시그널링을 수신하도록 구성될 수도 있다. 도 8의 예에서, 비디오 디코더 (30)는, 예를 들어, 비디오 데이터 메모리 (78), 엔트로피 디코딩 유닛 (80), 예측 프로세싱 유닛 (81), 역 양자화 유닛 (86), 역 변환 프로세싱 유닛 (88), 합산기 (90) 및 DPB (94)를 포함한다. 예측 프로세싱 유닛 (81)은 모션 보상 유닛 (82) 및 인트라 예측 유닛 (84)을 포함한다. 비디오 디코더 (30)는, 일부 예들에서, 도 7로부터 비디오 인코더 (20)에 대해 설명된 인코딩 패스에 일반적으로 상호 역의 디코딩 패스를 수행할 수도 있다.

[0192] 디코딩 프로세스 동안, 비디오 디코더 (30)는 연관된 신택스 엘리먼트들 및 인코딩된 비디오 슬라이스의 비디오 블록들을 나타내는 인코딩된 비디오 비트스트림을 비디오 인코더 (20)로부터 수신한다. 비디오 디코더

(30) 는 수신된 인코딩된 비디오 비트스트림을 비디오 데이터 메모리 (78)에 저장한다. 비디오 데이터 메모리 (78)는, 비디오 디코더 (30)의 컴포넌트들에 의해 디코딩된 비디오 비트스트림과 같은 비디오 데이터를 저장할 수도 있다. 비디오 데이터 메모리 (78)에 저장된 비디오 데이터는, 예를 들어, 링크 (16)를 통해, 저장 디바이스 (26)로부터, 또는 카메라와 같은 로컬 비디오 소스로부터, 또는 물리적 데이터 저장 매체들을 액세스함으로써, 획득될 수도 있다. 비디오 데이터 메모리 (78)는, 인코딩된 비디오 비트스트림으로부터의 인코딩된 비디오 데이터를 저장하는 코딩된 픽처 베퍼 (CPB)를 형성할 수도 있다. DPB (94)는, 예를 들어, 인트라 또는 인터 코딩 모드들에서 비디오 디코더 (30)에 의해 비디오 데이터를 디코딩함에 있어서 사용하기 위해 참조 비디오 데이터를 저장하는 참조 픽처 메모리일 수도 있다. 비디오 데이터 메모리 (78) 및 DPB (94)는 DRAM, SDRAM, MRAM, RRAM, 또는 다른 타입들의 메모리 디바이스들과 같은 다양한 메모리 디바이스들의 임의의 것에 의해 형성될 수도 있다. 비디오 데이터 메모리 (78) 및 DPB (94)는 동일한 메모리 디바이스 또는 별도의 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에서, 비디오 데이터 메모리 (78)는 비디오 디코더 (30)의 다른 컴포넌트들과 온-칩형이거나 또는 그 컴포넌트들에 대하여 오프-칩형일 수도 있다.

[0193] 비디오 디코더 (30)의 엔트로피 디코딩 유닛 (80)은 비디오 데이터 메모리 (78)에 저장된 비디오 데이터를 엔트로피 디코딩하여, 양자화된 계수들, 모션 벡터들, 및 다른 신택스 엘리먼트들을 생성한다. 엔트로피 디코딩 유닛 (80)은 모션 벡터들 및 다른 신택스 엘리먼트들을 예측 프로세싱 유닛 (81)으로 포워딩한다. 비디오 디코더 (30)는 신택스 엘리먼트들을 비디오 슬라이스 레벨 및/또는 비디오 블록 레벨로 수신할 수도 있다.

[0194] 비디오 슬라이스가 인트라-코딩된 (I) 슬라이스로서 코딩될 때, 예측 프로세싱 유닛 (81)의 인트라 예측 유닛 (84)은 시그널링된 인트라 예측 모드와, 현재의 프레임 또는 픽처의 이전에 디코딩된 블록들로부터의 데이터에 기초하여, 현재의 비디오 슬라이스의 비디오 블록에 대한 예측 데이터를 생성할 수도 있다. 비디오 프레임이 인터-코딩된 슬라이스 (예컨대, B 슬라이스 또는 P 슬라이스)로서 코딩될 때, 예측 프로세싱 유닛 (81)의 모션 보상 유닛 (82)은 모션 벡터들과, 엔트로피 디코딩 유닛 (80)으로부터 수신된 다른 신택스 엘리먼트들에 기초하여, 현재의 비디오 슬라이스의 비디오 블록에 대한 예측 블록들을 생성한다. 예측 블록들은 참조 픽처 리스트들 중의 하나 내의 참조 픽처들 중의 하나로부터 생성될 수도 있다. 비디오 디코더 (30)는 DPB (94)에 저장된 참조 픽처들에 기초한 디폴트 구성 기법들을 이용하여 참조 프레임 리스트들, 즉, 리스트 0 및 리스트 1을 구성할 수도 있다.

[0195] 모션 보상 유닛 (82)은 모션 벡터들 및 다른 신택스 엘리먼트들을 과정함으로써 현재 비디오 슬라이스의 비디오 블록에 대한 예측 정보를 결정하고, 그 예측 정보를 사용하여, 디코딩되고 있는 현재 비디오 블록에 대한 예측 블록들을 생성한다. 예를 들어, 모션 보상 유닛 (82)은 수신된 신택스 엘리먼트들 중 일부를 사용하여, 비디오 슬라이스의 비디오 블록들을 코딩하는데 사용되는 예측 모드 (예를 들어, 인트라-예측 또는 인터-예측), 인터-예측 슬라이스 타입 (예를 들어, B 슬라이스 또는 P 슬라이스), 슬라이스에 대한 참조 픽처 리스트들 중 하나 이상에 대한 구성 정보, 슬라이스의 각각의 인터-인코딩된 비디오 블록에 대한 모션 벡터들, 슬라이스의 각각의 인터-코딩된 비디오 블록에 대한 인터-예측 상태, 및 현재 비디오 슬라이스에 있어서의 비디오 블록들을 디코딩하기 위한 다른 정보를 결정한다.

[0196] 모션 보상 유닛 (82)은 보간 필터들에 기초하여 보간을 또한 수행할 수도 있다. 모션 보상 유닛 (82)은 비디오 블록들의 인코딩 동안 비디오 인코더 (20)에 의해 이용되는 것과 같이 보간 필터들을 이용하여 참조 블록들의 부-정수 픽셀들에 대한 보간된 값들을 계산할 수도 있다. 이 경우, 모션 보상 유닛 (82)은 수신된 신택스 엘리먼트들로부터 비디오 인코더 (20)에 의해 이용되는 보간 필터들을 결정하고 보간 필터들을 이용하여 예측 블록들을 생성할 수도 있다.

[0197] 역 양자화 유닛 (86)은 비트스트림으로 제공되고 엔트로피 디코딩 유닛 (80)에 의해 디코딩된 양자화된 변환 계수들을 역 양자화, 즉, 양자화해제한다. 역양자화 프로세스는 적용되어야 할 양자화도 및, 마찬가지로, 역양자화도를 결정하기 위하여 비디오 슬라이스에서의 각각의 비디오 블록에 대해 비디오 인코더 (20)에 의해 계산된 양자화 파라미터의 이용을 포함할 수도 있다. 역변환 프로세싱 유닛 (88)은 픽셀 도메인 (pixel domain)에서 잔차 블록들을 생성하기 위하여, 역변환, 예컨대, 역 DCT, 역 정수 변환 (inverse integer transform), 또는 개념적으로 유사한 역 변환 프로세스를 변환 계수들에 적용한다.

[0198] 예측 프로세싱 유닛이 예를 들어 인트라 또는 인터 예측을 이용하여 현재 비디오 블록에 대한 예측 블록을 생성한 이후, 비디오 디코더 (30)는 역변환 프로세싱 유닛 (88)으로부터의 잔차 블록들을 모션 보상 유닛 (82)에

의해 생성된 대응하는 예측 블록들과 합산함으로써 재구성된 비디오 블록을 형성한다. 합산기 (90)는 이 합산 동작을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다.

[0199] 필터 유닛 (92)은 재구성된 블록 (예컨대, 합산기 (90)의 출력)을 필터링하고, 그 필터링된 재구성된 블록을 참조 블록으로서의 사용들을 위해 DPB (94)에 저장한다. 참조 블록은, 후속 비디오 프레임 또는 픽처에서의 블록을 인터-예측하기 위해 참조 블록으로서 모션 보상 유닛 (82)에 의해 사용될 수도 있다. 필터 유닛 (92)은 디블록 필터링, SAO 필터링, 퍼크 SAO 필터링, ALF, 및/또는 GALF, 및/또는 다른 타입들의 루프 필터들과 같은 임의의 타입의 필터링을 수행할 수도 있다. 디블록 필터는, 예를 들어, 블록 경계들을 필터링하여 재구성된 비디오로부터 블록키니스 아티팩트들을 제거하기 위해 디블록킹 필터링을 적용할 수도 있다. 퍼크 SAO 필터는 전체적인 코딩 품질을 향상시키기 위해 재구성된 픽셀 값들에 오프셋들을 적용할 수도 있다. 추가적인 루프 필터들 (인 루프 (in loop) 또는 포스트 루프 (post loop))이 또한 사용될 수도 있다.

[0200] 도 9는 필터 유닛 (92)의 일 예시적인 구현을 나타낸다. 필터 유닛 (64)은 동일한 방식으로 구현될 수도 있다. 필터 유닛들 (64 및 92)은, 가능하게는 비디오 인코더 (20) 또는 비디오 디코더 (30)의 다른 컴포넌트들과 함께, 본 개시의 기법들을 수행할 수도 있다. 도 9의 예에서, 필터 유닛 (92)은 디블록 필터 (102), 퍼크 SAO 필터 (104), 및 ALF/GLAF 필터 (106)를 포함한다. 퍼크 SAO 필터 (104)는, 예를 들어, 본 개시에서 기술된 방식으로 블록의 샘플들에 대한 오프셋 값들을 결정하도록 구성될 수도 있다.

[0201] 필터 유닛 (92)은 더 적은 필터들을 포함할 수도 있고 및/또는 추가적인 필터들을 포함할 수도 있다. 추가적으로, 도 9에서 도시된 특정 필터들은 상이한 순서로 구현될 수도 있다. (코딩 루프에 있어서 또는 코딩 루프 이후에) 다른 루프 필터들이 또한 픽셀 천이들을 평활하게 하거나 또는 그렇지 않으면 비디오 품질을 개선하기 위해 이용될 수도 있다. 그 후, 소정의 프레임 또는 픽처에 있어서의 디코딩된 비디오 블록들이 DPB (94)에 저장되고, 이 DPB는 후속적인 모션 보상을 위해 사용되는 참조 픽처들을 저장한다. DPB (94)는 도 1의 디스플레이 디바이스 (32)와 같은 디스플레이 디바이스 상으로의 나중의 프리젠테이션을 위해 디코딩된 비디오를 저장하는 추가적인 메모리의 일부일 수도 있거나 그 추가적인 메모리로부터 분리될 수도 있다.

[0202] 도 10은 본 개시에서 설명된 예시적인 비디오 디코딩 기법을 나타내는 흐름도이다. 도 10의 기법들은 비제한적으로 비디오 디코더 (30)와 같은 일반적인 비디오 디코더를 참조하여 설명될 것이다. 일부 경우들에서, 도 10의 기법들은 비디오 인코더 (20)의 디코딩 루프에 의해 수행될 수도 있다.

[0203] 도 10의 예에서, 비디오 디코더는 현재 블록의 제 1 샘플을 이웃 샘플들의 제 1 셋트에 대해 비교하여 제 1 차이 값을 결정한다 (120). 제 1 차이 값은 제 1 샘플과 이웃 샘플들의 제 1 셋트 사이의 차이일 수도 있다. 비디오 데이터의 현재 블록에 대해, 비디오 디코더는 이웃하는 샘플들의 로케이션들을 나타내는 신택스 데이터를 수신할 수도 있고, 이웃하는 샘플들의 로케이션들은 패턴을 정의한다. 비디오 디코더는 이러한 이웃하는 샘플들의 로케이션들을 나타내는 신택스를, 예를 들어, 슬라이스 레벨, 또는 다른 레벨들, 슬라이스 레벨 아래 또는 위에서, 수신할 수도 있다. 하나의 예시적인 구현에서, 이웃하는 샘플들의 로케이션들을 나타내는 신택스 데이터는 인덱스 값을 포함하고, 비디오 디코더는 인덱스 값을 이웃하는 샘플들의 로케이션들의 셋트에 대해 맵핑한다.

[0204] 비디오 코더는, 예를 들어, 현재 블록의 제 1 샘플의 값을 이웃 샘플들의 제 1 셋트의 평균 값에 대해 비교하는 것, 현재 블록의 제 1 샘플의 값을 이웃 샘플들의 제 1 셋트의 가중된 평균 값에 대해 비교하는 것, 현재 블록의 제 1 샘플의 값과 이웃 샘플들의 제 1 셋트의 샘플 값들 사이의 값 차이들의 합계를 결정하는 것에 의해서, 또는 몇몇 다른 타입의 비교에 의해서, 현재 블록의 제 1 샘플을 이웃 샘플들의 제 1 셋트에 대해 비교하여 제 1 차이 값을 결정할 수도 있다.

[0205] 비디오 디코더는 제 1 차이 값에 기초하여 제 1 샘플에 대한 제 1 오프셋을 결정한다 (122). 비디오 디코더는 현재 블록의 제 2 샘플을 이웃 샘플들의 제 2 셋트에 대해 비교하여 제 2 차이 값을 결정한다 (124). 제 1 샘플과 유사하게, 제 2 샘플에 대해, 비디오 코더는, 현재 블록의 제 2 샘플의 값을 이웃 샘플들의 제 2 셋트의 평균 값에 대해 비교하는 것, 현재 블록의 제 2 샘플의 값을 이웃 샘플들의 제 2 셋트의 가중된 평균 값에 대해 비교하는 것, 현재 블록의 제 2 샘플의 값과 이웃 샘플들의 제 2 셋트의 샘플 값들 사이의 값 차이들의 합계를 결정하는 것에 의해서, 또는 몇몇 다른 타입의 비교에 의해서, 현재 블록의 제 2 샘플을 이웃 샘플들의 제 2 셋트에 대해 비교하여 제 2 차이 값을 결정할 수도 있다. 제 1 차이 값은 제 1 샘플과 이웃 샘플들의 제 2 셋트 사이의 차이이고, 제 1 차이 값 및 제 2 차이 값은 동일한 부호를 갖는다 (예컨대, 야암 모두 포지티브 이거나 양자 모두 네거티브이다). 이웃 샘플들의 제 1 셋트는 이웃 샘플들의 제 2 셋트가 제 2 샘플에 대하여 가지는 것과 동일한 제 1 샘플에 대한 패턴을 갖는다. 이웃 샘플들의 제 1 셋트 및 이웃 샘플들의 제 2

셋트는 각각, 2 개보다 많은 샘플들을 가지고, 이웃 샘플들의 제 1 셋트 및 이웃 샘플들의 제 2 셋트는 또한, 적어도 하나의 시간적 이웃하는 샘플들을 각각 포함할 수도 있다.

[0206] 비디오 디코더는 제 2 차이 값에 기초하여 제 2 샘플에 대한 제 2 오프셋을 결정한다 (126). 제 1 오프셋 및 제 2 오프셋 양자는 상이한, 비-제로 값들이다. 비디오 디코더는 제 1 오프셋을 제 1 샘플 값에 그리고 제 2 오프셋을 제 2 샘플 값에 적용하여 필터링된 이미지를 생성한다 (128). 비디오 데이터의 현재 블록에 대해, 비디오 디코더는 현재 블록의 그룹핑 샘플들에 대한 규칙들의 셋트를 나타내는 신택스 데이터를 적어도 2 개의 그룹들로 수신할 수도 있다. 제 1 그룹은 예를 들어 필터링될 샘플들을 포함하고, 제 2 그룹은 필터링되지 않을 샘플들을 포함할 수도 있다.

[0207] 도 10 의 예에서, 제 1 샘플의 값은 이웃 샘플들의 제 1 셋트에서의 모든 샘플들의 값들보다 더 클 수도 (또는 더 적을 수도) 있고, 제 2 샘플의 값은 또한, 이웃 샘플들의 제 2 셋트에서의 모든 샘플들의 값들보다 더 클 수도 (또는 더 적을 수도) 있다. 그럼에도 불구하고, 제 1 샘플 및 제 2 샘플이 이러한 특성을 공유함에도 불구하고, 비디오 디코더는 예를 들어 차이들의 크기에 기초하여 제 1 샘플 및 제 2 샘플에 상이한 오프셋들을 부가한다. 하나의 예로서, 제 1 샘플의 값은 이웃 샘플들의 제 1 셋트에서의 모든 샘플들의 값들보다 훨씬 더 클 수도 (또는 훨씬 더 적을 수도) 있는 한편, 제 2 샘플의 값은 이웃 샘플들의 제 2 셋트에서의 모든 샘플들의 값들보다 오직 약간 더 크다 (또는 약간 더 적다). 이 차이에 기초하여, 비디오 디코더는 제 1 샘플 및 제 2 샘플에 상이한 오프셋들을 적용할 수도 있다.

[0208] 하나 이상의 예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현되는 경우, 그 기능들은 하나 이상의 명령들 또는 코드로서 컴퓨터 판독가능 매체 상으로 저장 또는 전송되고 하드웨어 기반 프로세싱 유닛에 의해 실행될 수도 있다. 컴퓨터 판독가능 매체들은 데이터 저장 매체들과 같은 유형의 매체에 대응하는 컴퓨터 판독가능 저장 매체들, 또는 예를 들어, 통신 프로토콜에 따라 일 장소로부터 다른 장소로의 컴퓨터 프로그램의 전송을 용이하게 하는 임의의 매체를 포함하는 통신 매체들을 포함할 수도 있다. 이러한 방식으로, 컴퓨터 판독가능 매체들은 일반적으로 (1) 비일시적인 유형의 컴퓨터 판독가능 저장 매체들 또는 (2) 신호 또는 캐리어파와 같은 통신 매체에 대응할 수도 있다. 데이터 저장 매체들은 본 개시에서 설명된 기법들의 구현을 위한 명령들, 코드 및/또는 데이터 구조들을 취출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 가용 매체들일 수도 있다. 컴퓨터 프로그램 제품이 컴퓨터 판독가능 매체를 포함할 수도 있다.

[0209] 한정이 아닌 예로서, 그러한 컴퓨터 판독가능 저장 매체들은 RAM, ROM, EEPROM, CD-ROM 또는 다른 광학 디스크 저장부, 자기 디스크 저장부 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 원하는 프로그램 코드를 명령들 또는 데이터 구조들의 형태로 저장하는데 이용될 수 있고 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 커넥션이 컴퓨터 판독가능 매체로 적절히 명명된다. 예를 들어, 동축 케이블, 광섬유 케이블, 꼬임쌍선, 디지털 가입자 라인 (DSL), 또는 적외선, 무선, 및 마이크로파와 같은 무선 기술들을 이용하여 웹사이트, 서버, 또는 다른 원격 소스로부터 소프트웨어가 송신된다면, 동축 케이블, 광섬유 케이블, 꼬임쌍선, DSL, 또는 적외선, 무선, 및 마이크로파와 같은 무선 기술들은 매체의 정의에 포함된다. 하지만, 컴퓨터 판독가능 저장 매체들 및 데이터 저장 매체들은 커넥션들, 캐리어파들, 신호들, 또는 다른 일시적 매체들을 포함하지 않지만 대신 비일시적인 유형의 저장 매체들로 지향됨을 이해해야 한다. 본 원에서 이용된 디스크 (disk) 와 디스크 (disc) 는, 컴팩트 디스크(CD), 레이저 디스크, 광학 디스크, 디지털 다기능 디스크 (DVD), 플로피 디스크, 및 블루레이 디스크를 포함하며, 여기서 디스크 (disk) 들은 통상 자기적으로 데이터를 재생하는 반면, 디스크(disc) 들은 레이저들을 이용하여 광학적으로 데이터를 재생한다. 상기의 조합들이 또한, 컴퓨터 판독가능 매체들의 범위 내에 포함되어야 한다.

[0210] 명령들은, 하나 이상의 디지털 신호 프로세서들(DSPs), 범용 마이크로프로세서들, 주문형 집적 회로들(ASICs), 필드 프로그래머블 로직 어레이(FPGAs), 또는 다른 등가의 집적 또는 이산 로직 회로와 같은, 하나 이상의 프로세서들에 의해 실행될 수도 있다. 따라서, 본원에 사용된 용어 "프로세서" 는 전술한 구조 중 임의의 것 또는 본원에 설명된 기법들의 구현에 적합한 임의의 다른 구조를 지칭할 수도 있다. 또한, 일부 양태들에 있어서, 본 명세서에서 설명된 기능은 인코딩 및 디코딩을 위해 구성되거나 또는 결합된 코덱에서 통합된 전용 하드웨어 및/또는 소프트웨어 모듈들 내에 제공될 수도 있다. 또한, 그 기법들은 하나 이상의 회로들 또는 로직 엘리먼트들에서 완전히 구현될 수 있다.

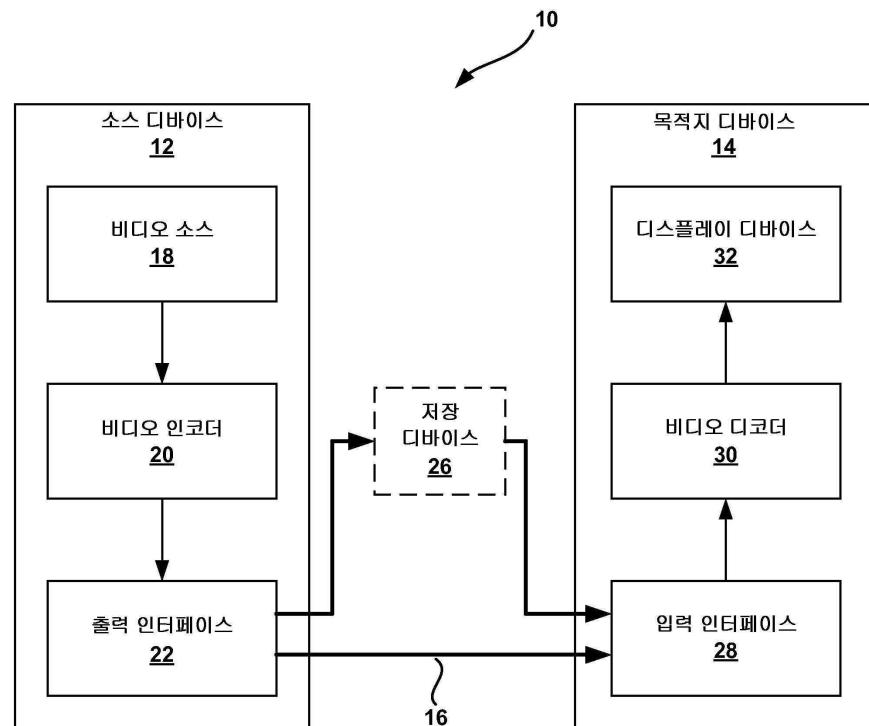
[0211] 본 개시의 기법들은 무선 핸드셋, 집적 회로 (IC) 또는 IC들의 셋트 (예를 들어, 칩 셋트) 를 포함하여 매우 다양한 디바이스들 또는 장치들에서 구현될 수도 있다. 다양한 컴포넌트들, 모듈들 또는 유닛들이, 개시된 기

술들을 수행하도록 구성된 디바이스들의 기능적인 양태들을 강조하기 위하여 본 개시에 설명되었지만, 상이한 하드웨어 유닛들에 의한 실현을 반드시 필요로 하는 것은 아니다. 오히려, 상술된 바처럼, 다양한 유닛들이 코덱 하드웨어 유닛에 결합될 수도 있거나, 또는 적합한 소프트웨어 및/또는 펌웨어와 함께, 상술된 하나 이상의 프로세서들을 포함하는 연동적인 (interoperative) 하드웨어 유닛들의 집합에 의해 제공될 수도 있다.

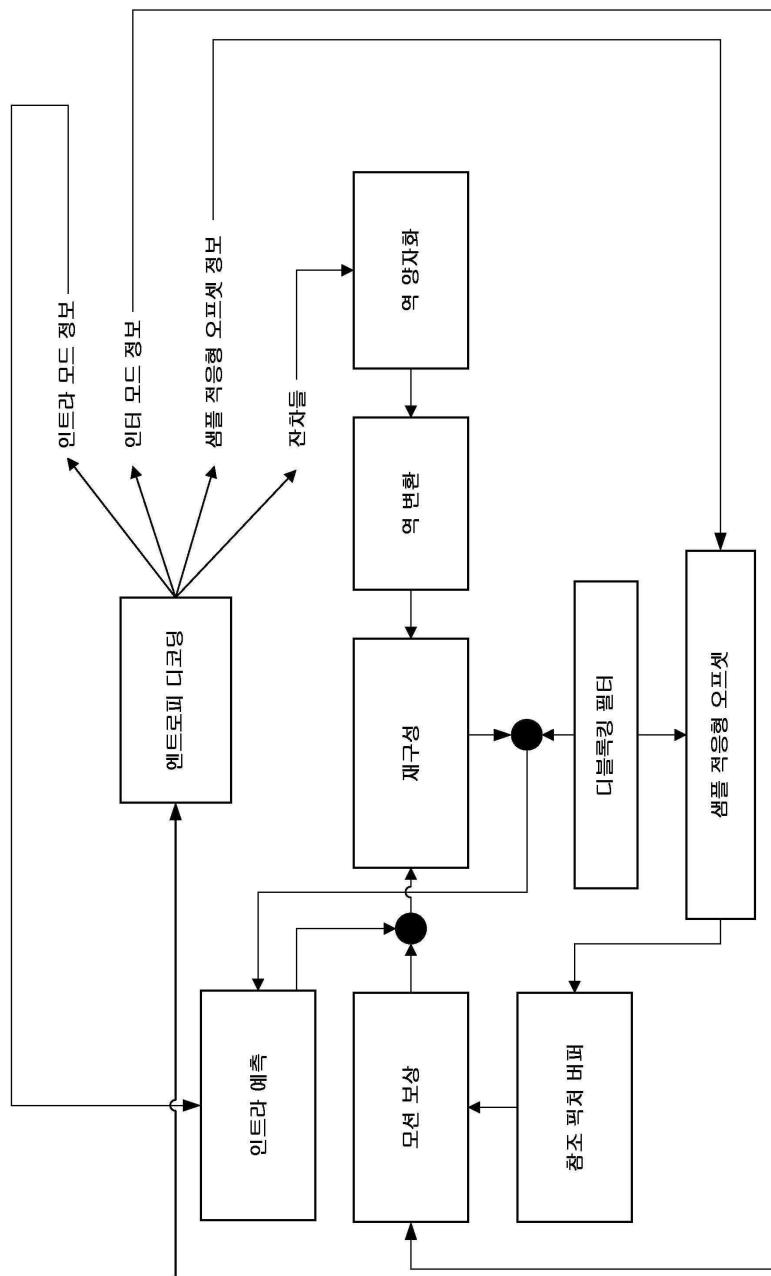
[0212] 다양한 예들이 설명되었다. 이들 및 다른 예들은 다음의 청구항들의 범위 내에 있다.

## 도면

### 도면1



도면2



도면3a

A	C	B

도면3b

	A	
	C	
	B	

도면3c

A		
	C	
		B

도면3d

		A
	C	
B		

도면4

D00	D01	D02
D03	s0	D04
D05	D06	D07

Dx0	Dx1	Dx2
Dx3	sx	Dx4
Dx5	Dx6	Dx7

C0	C1	C2
C3	X	C4
C5	C6	C7

검색 윈도우

## 도면5

C4	C1	C5
C0	C	C2
C6	C3	C7

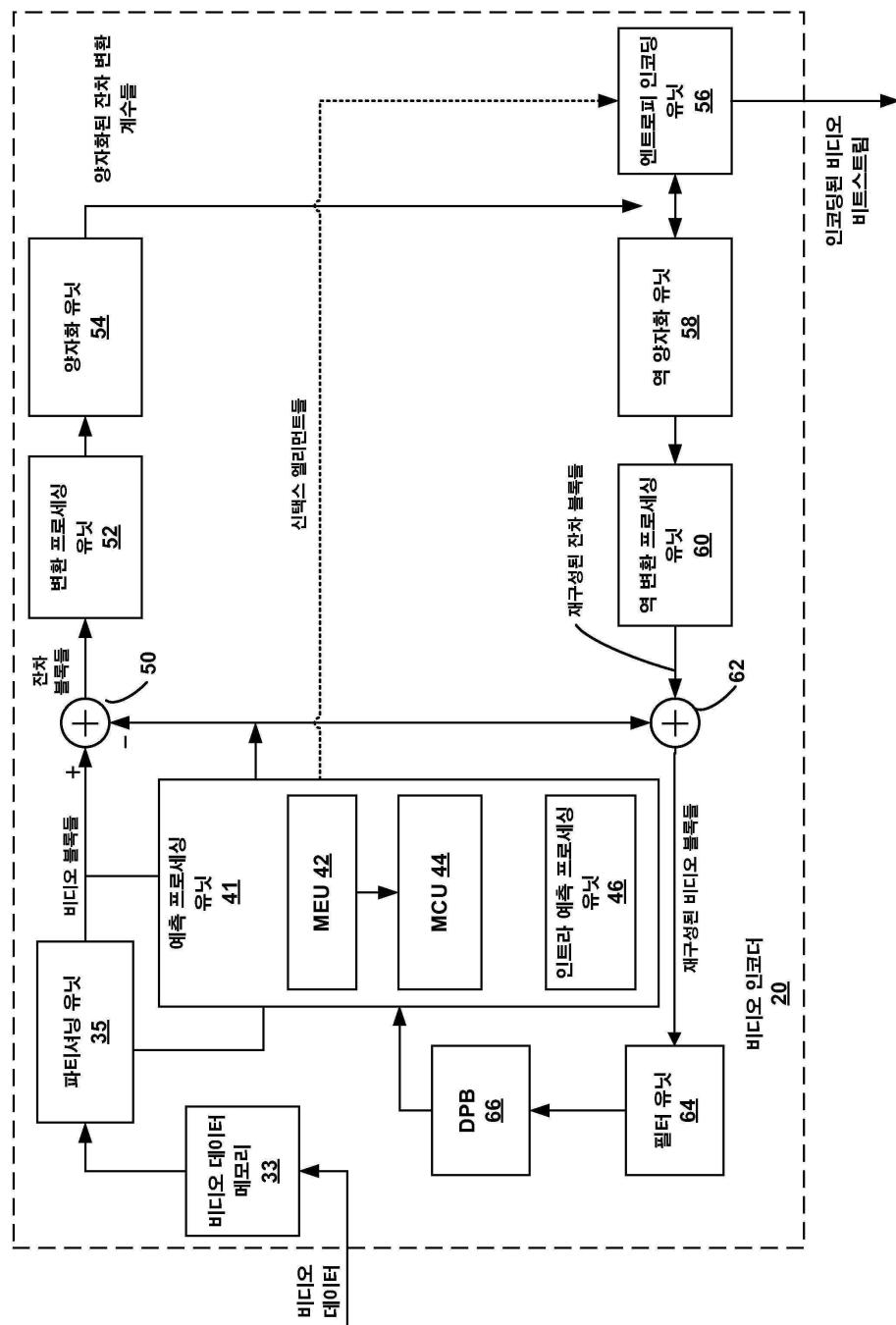
## 도면6a

52	52	53
51	50	53
52	48	47

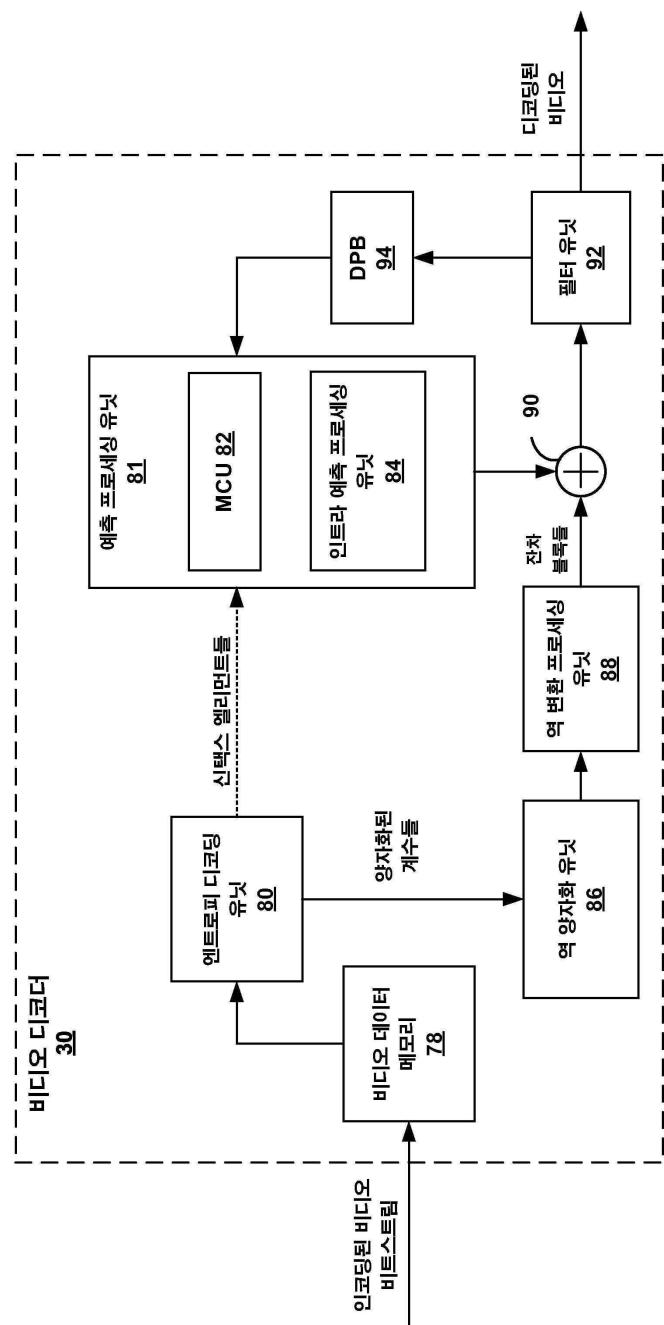
## 도면6b

90	90	95
91	93	93
92	91	91

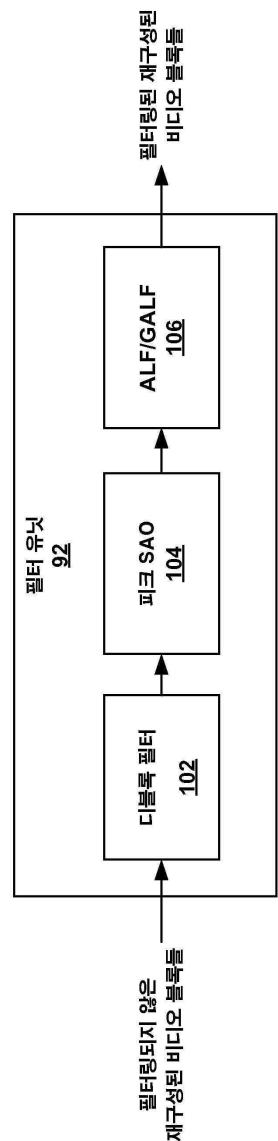
도면7



도면8



도면9



## 도면10

