



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2015-0135410  
(43) 공개일자 2015년12월02일

(51) 국제특허분류(Int. Cl.)	(71) 출원인
HO4N 19/593 (2014.01) HO4N 19/436 (2014.01)	퀄컴 인코포레이티드
HO4N 19/467 (2014.01) HO4N 19/70 (2014.01)	미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
(52) CPC특허분류	(72) 발명자
HO4N 19/593 (2015.01)	조쉬 라잔 랙스맨
HO4N 19/436 (2015.01)	미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
(21) 출원번호 10-2015-7030330	솔레 로할스 호엘
(22) 출원일자(국제) 2014년03월25일	미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
심사청구일자 없음	카르체비츠 마르타
(85) 번역문제출일자 2015년10월21일	미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
(86) 국제출원번호 PCT/US2014/031728	(74) 대리인
(87) 국제공개번호 WO 2014/160712	특허법인코리아나
국제공개일자 2014년10월02일	
(30) 우선권주장	
61/805,094 2013년03월25일 미국(US)	
(뒷면에 계속)	

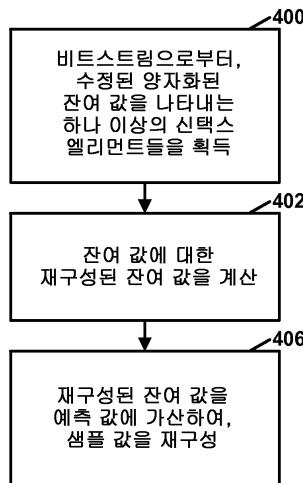
전체 청구항 수 : 총 36 항

(54) 발명의 명칭 변환이 스kip될 때 손실 코딩을 위한 인트라 예측 모드들

**(57) 요 약**

비디오 코더는 손실 코딩을 사용하여 코딩된 블록의 잔여 데이터에 잔여 차분 펠스 코드 변조 기법을 적용한다. 블록은 잔여 데이터에 대한 변환의 적용 없이 코딩될 수도 있다.

**대 표 도** - 도9b



(52) CPC특허분류

*HO4N 19/467* (2015.01)  
*HO4N 19/70* (2015.01)

(30) 우선권주장

61/809,199	2013년04월05일	미국(US)
61/809,203	2013년04월05일	미국(US)
61/809,811	2013년04월08일	미국(US)
61/809,870	2013년04월08일	미국(US)
61/810,179	2013년04월09일	미국(US)
61/810,218	2013년04월09일	미국(US)
61/843,144	2013년07월05일	미국(US)
14/223,874	2014년03월24일	미국(US)

## 명세서

### 청구범위

#### 청구항 1

비디오 데이터를 디코딩하는 방법으로서,

잔여 값들의 블록을 생성하는 단계로서,

상기 블록은 변환 스윕 블록이고, 상기 블록을 생성하는 단계는,

$M$  이 상기 블록의 높이이고  $N$  이 상기 블록의 폭인,  $0 \leq i \leq (M - 1)$  및  $0 \leq j \leq (N - 1)$ 에 대해, 상기 블록에서 잔여 값  $r_{i,j}$ 에 대한 재구성된 잔여 값  $Q(r_{i,j})$ 를 계산하는 단계를 포함하고,

상기 블록이 수직 인트라 예측 모드를 사용하여 코딩되면,  $Q(r_{i,j})$ 는 다음과 같이 정의되며:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & i=0, 0 \leq j \leq (N-1), \\ Q(\tilde{r}_{i,j}) + Q(r_{i-1,j}) & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases},$$

$Q(\tilde{r}_{i,j})$  는 수정된 잔여 값  $\tilde{r}_{i,j}$ 의 양자화된 버전을 나타내고, 상기 수정된 잔여 값  $\tilde{r}_{i,j}$ 는 상기 잔여 값  $r_{i,j}$ 의 수정된 버전이며,  $Q(r_{i-1,j})$ 는 상기 잔여 값  $r_{i,j}$ 의 한 칼럼 좌측에 있는 잔여 값에 대한 재구성된 잔여 값이고,

상기 블록이 수평 인트라 예측 모드를 사용하여 코딩되면,  $Q(r_{i,j})$ 는 다음과 같이 정의되며:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & 0 \leq i \leq (M-1), j=0, \\ Q(\tilde{r}_{i,j}) + Q(r_{i,j-1}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases},$$

$Q(r_{i,j-1})$ 은 상기 잔여 값  $r_{i,j}$ 의 한 로우 상부에 있는 잔여 값에 대한 재구성된 잔여 값인, 상기 잔여 값들의 블록을 생성하는 단계; 및

$0 \leq i \leq (M-1)$  및  $0 \leq j \leq (N-1)$ 에 대해, 상기 재구성된 잔여 값  $Q(r_{i,j})$ 를 예측 값에 가산하여 샘플 값을 재구성하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

#### 청구항 2

제 1 항에 있어서,

비트스트림으로부터, 수정된 양자화된 잔여 값  $\tilde{r}_{i,j}$ 을 나타내는 하나 이상의 신택스 엘리먼트들을 획득하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

#### 청구항 3

제 1 항에 있어서,

상기 블록이 상기 수직 인트라 예측 모드를 사용하여 코딩되면, 상기 수정된 잔여 값  $\tilde{r}_{i,j}$ 는 다음과 같이 정의되며:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ r_{i,j} - Q(r_{(i-1),j}), & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases},$$

상기 블록이 상기 수평 인트라 예측 모드를 사용하여 코딩되면, 상기 수정된 잔여 값  $\tilde{r}_{i,j}$  는 다음과 같이 정의 되는:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & 0 \leq i \leq (M-1), j=0 \\ r_{i,j} - Q(r_{i,(j-1)}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}, \text{ 비디오 데이터를 디코딩하는 방법.}$$

#### 청구항 4

제 1 항에 있어서,

상기 블록이 HEVC (High Efficiency Video Coding) 작업 초안 10에서 정의된 바와 같이 22와 30 사이의 인트라 예측 모드를 사용하여 코딩되면, 상기 양자화된 수정된 잔여 값  $Q(\tilde{r}_{i,j})$  는 다음과 같이 정의되고:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & i=0, 0 \leq j \leq (N-1), \\ Q(\tilde{r}_{i,j}) + Q(r_{i-1,j}) & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases},$$

상기 블록이 HEVC 작업 초안 10에서 정의된 바와 같이 6과 14 사이의 인트라 예측 모드를 사용하여 코딩되면,  $Q(\tilde{r}_{i,j})$  는 다음과 같이 정의되는:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & 0 \leq i \leq (M-1), j=0, \\ Q(\tilde{r}_{i,j}) + Q(r_{i,j-1}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}, \text{ 비디오 데이터를 디코딩하는 방법.}$$

#### 청구항 5

제 1 항에 있어서,

상기 제 2 블록이 대각선 인트라 예측 모드를 사용하여 코딩되면,  $Q(r_{i,j})$  는 다음과 같이 정의되며:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & i=0, 0 \leq j \leq (N-1) \\ Q(\tilde{r}_{i,j}), & 1 \leq i \leq (M-1), j=0 \\ Q(\tilde{r}_{i,j}) + Q(r_{i-1,j-1}), & 1 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases},$$

$Q(r_{i-1,j-1})$  은 상기 잔여 값  $r_{i,j}$  의 한 칼럼 좌측 및 한 로우 상부에 있는 잔여 값에 대한 재구성된 잔여 값인, 비디오 데이터를 디코딩하는 방법.

#### 청구항 6

제 5 항에 있어서,

상기 수정된 잔여 값  $\tilde{r}_{i,j}$  는 다음과 같이 정의되며:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ r_{i,j}, & 1 \leq i \leq (M-1), j=0 \\ r_{i,j} - Q(r_{i-1,j-1}), & 1 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases},$$

$Q(r_{i-1,j-1})$  은 상기 잔여 값  $r_{i,j}$  의 한 칼럼 좌측 및 한 로우 상부에 있는 잔여 값  $r_{i-1,j-1}$  에 대한 재구성된 잔여 값인, 비디오 데이터를 디코딩하는 방법.

### 청구항 7

제 1 항에 있어서,

상기 잔여 값  $r_{i,j}$  는 비트-시프트된 잔여 값인, 비디오 데이터를 디코딩하는 방법.

### 청구항 8

비디오 데이터를 인코딩하는 방법으로서,

$M$  이 블록의 높이이고  $N$  이 상기 블록의 폭이고, 상기 블록은 변환 스윕 블록인,  $0 \leq i \leq (M - 1)$  및  $0 \leq j \leq (N - 1)$  에 대해,

잔여 값  $r_{i,j}$  에 대한 수정된 잔여 값  $\tilde{r}_{i,j}$  를 결정하는 단계로서,

상기 블록이 수직 인트라 예측 모드를 사용하여 코딩되면,  $\tilde{r}_{i,j}$  는 다음과 같이 정의되며:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i = 0, 0 \leq j \leq (N - 1) \\ r_{i,j} - Q(r_{(i-1),j}), & 1 \leq i \leq (M - 1), 0 \leq j \leq (N - 1) \end{cases}$$

$Q(r_{(i-1),j})$  는 상기 잔여 값  $r_{i,j}$  의 한 칼럼 좌측에 있는 잔여 값  $r_{i-1,j}$  에 대한 재구성된 잔여 값을 나타내고,

상기 블록이 수평 인트라 예측 모드를 사용하여 코딩되면,  $\tilde{r}_{i,j}$  는 다음과 같이 정의되며:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & 0 \leq i \leq (M - 1), j = 0 \\ r_{i,j} - Q(r_{i,(j-1)}) & 0 \leq i \leq (M - 1), 1 \leq j \leq (N - 1) \end{cases}$$

$Q(r_{i,(j-1)})$  는 상기 잔여 값  $r_{i,j}$  의 한 로우 상부에 있는 잔여 값  $r_{i,j-1}$  에 대한 재구성된 잔여 값을 나타내는,

상기 잔여 값  $r_{i,j}$  에 대한 수정된 잔여 값  $\tilde{r}_{i,j}$  를 결정하는 단계; 및

상기 수정된 잔여 값  $\tilde{r}_{i,j}$  을 양자화하여 양자화된 수정된 잔여 값  $Q(\tilde{r}_{i,j})$  을 생성하는 단계를 포함하는, 비디오 데이터를 인코딩하는 방법.

### 청구항 9

제 8 항에 있어서,

상기 양자화된 수정된 잔여 값  $Q(\tilde{r}_{i,j})$  을 비트스트림으로 시그널링하는 단계를 더 포함하는, 비디오 데이터를 인코딩하는 방법.

### 청구항 10

제 8 항에 있어서,

재구성된 잔여 값  $Q(r_{i,j})$  를 계산하는 단계를 더 포함하고,

상기 블록이 상기 수직 인트라 예측 모드를 사용하여 코딩되면, 상기 재구성된 잔여 값  $Q(r_{i,j})$  는 다음과 같이

정의되고:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & i=0, 0 \leq j \leq (N-1), \\ Q(\tilde{r}_{i,j}) + Q(r_{i-1,j}) & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases},$$

상기 블록이 상기 수평 인트라 예측 모드를 사용하여 코딩되면, 상기 재구성된 잔여 값  $Q(r_{i,j})$  는 다음과 같이 정의되는:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & 0 \leq i \leq (M-1), j=0, \\ Q(\tilde{r}_{i,j}) + Q(r_{i,j-1}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}, \text{ 비디오 데이터를 인코딩하는 방법.}$$

### 청구항 11

제 10 항에 있어서,

상기 재구성된 잔여 값  $Q(r_{i,j})$  를 예측 값에 가산하여 샘플 값을 재구성하는 단계; 및

다른 블록들의 인트라 예측 또는 인터 예측에 상기 샘플 값을 사용하는 단계를 더 포함하는, 비디오 데이터를 인코딩하는 방법.

### 청구항 12

제 8 항에 있어서,

상기 블록이 HEVC (High Efficiency Video Coding) 작업 초안 10 에서 정의된 바와 같이 22 와 30 사이의 인트라 예측 모드를 사용하여 코딩되면, 상기 양자화된 수정된 잔여 값  $Q(\tilde{r}_{i,j})$  는 다음과 같이 정의되고:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & i=0, 0 \leq j \leq (N-1), \\ Q(\tilde{r}_{i,j}) + Q(r_{i-1,j}) & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases},$$

상기 블록이 HEVC 작업 초안 10 에서 정의된 바와 같이 6 과 14 사이의 인트라 예측 모드를 사용하여 코딩되면,

$Q(\tilde{r}_{i,j})$  는 다음과 같이 정의되는:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & 0 \leq i \leq (M-1), j=0, \\ Q(\tilde{r}_{i,j}) + Q(r_{i,j-1}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}, \text{ 비디오 데이터를 인코딩하는 방법.}$$

### 청구항 13

제 8 항에 있어서,

상기 블록이 대각선 인트라 예측 모드를 사용하여 코딩되면,  $\tilde{r}_{i,j}$  는 다음과 같이 정의되며:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ r_{i,j}, & 1 \leq i \leq (M-1), j=0 \\ r_{i,j} - Q(r_{i-1,j-1}), & 1 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases},$$

$Q(r_{i-1,j-1})$  은 상기 잔여 값  $r_{i,j}$  의 한 칼럼 좌측 및 한 로우 상부에 있는 잔여 값  $r_{i-1,j-1}$  에 대한 재구성된 잔여 값인, 비디오 데이터를 인코딩하는 방법.

### 청구항 14

제 13 항에 있어서,

상기 잔여 값  $r_{i,j}$ 에 대한 재구성된 잔여 값  $Q(r_{i,j})$ 를 계산하는 단계를 더 포함하고,

상기 블록이 상기 대각선 인트라 예측 모드를 사용하여 코딩되면, 상기 재구성된 잔여 값  $Q(r_{i,j})$ 는 다음과 같이 정의되며:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & i=0, 0 \leq j \leq (N-1) \\ Q(\tilde{r}_{i,j}), & 1 \leq i \leq (M-1), j=0 \\ Q(\tilde{r}_{i,j}) + Q(r_{i-1,j-1}), & 1 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases},$$

$Q(r_{i-1,j-1})$ 은 상기 잔여 값  $r_{i,j}$ 의 한 칼럼 좌측 및 한 로우 상부에 있는 잔여 값의 재구성된 버전인, 비디오 테이터를 인코딩하는 방법.

### 청구항 15

제 8 항에 있어서,

상기 잔여 값  $r_{i,j}$ 는 비트-시프트된 잔여 값인, 비디오 데이터를 인코딩하는 방법.

### 청구항 16

비디오 디코딩 장치로서,

데이터를 저장하는 메모리; 및

하나 이상의 프로세서들을 포함하고,

상기 하나 이상의 프로세서들은,

잔여 값들의 블록을 생성하는 것으로서,

상기 블록은 변환 스윕 블록이고, 상기 잔여 값들의 상기 블록을 생성하기 위해, 상기 하나 이상의 프로세서들은,

M이 상기 블록의 높이이고 N이 상기 블록의 폭인  $0 \leq i \leq (M-1)$  및  $0 \leq j \leq (N-1)$ 에 대해, 상기 블록에서 잔여 값  $r_{i,j}$ 에 대한 재구성된 잔여 값  $Q(r_{i,j})$ 을 계산하고,

상기 블록이 수직 인트라 예측 모드를 사용하여 코딩되면,  $Q(r_{i,j})$ 는 다음과 같이 정의되며:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & i=0, 0 \leq j \leq (N-1), \\ Q(\tilde{r}_{i,j}) + Q(r_{i-1,j}) & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases},$$

$\tilde{Q}(\tilde{r}_{i,j})$ 는 수정된 잔여 값  $\tilde{r}_{i,j}$ 의 양자화된 버전을 나타내고, 상기 수정된 잔여 값  $\tilde{r}_{i,j}$ 는 상기 잔여 값  $r_{i,j}$ 의 수정된 버전이며,  $Q(r_{i-1,j})$ 는 상기 잔여 값  $r_{i,j}$ 의 한 칼럼 좌측에 있는 잔여 값에 대한 재구성된 잔여 값이고,

상기 블록이 수평 인트라 예측 모드를 사용하여 코딩되면,  $Q(r_{i,j})$ 는 다음과 같이 정의되며:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & 0 \leq i \leq (M-1), j=0, \\ Q(\tilde{r}_{i,j}) + Q(r_{i,j-1}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases},$$

$Q(r_{i,j-1})$ 은 상기 잔여 값  $r_{i,j}$ 의 한 로우 상부에 있는 잔여 값에 대한 재구성된 잔여 값인, 상기 잔여 값들의 블록을 생성하며;

$0 \leq i \leq (M - 1)$  및  $0 \leq j \leq (N - 1)$ 에 대해, 상기 잔여 값  $r_{i,j}$ 를 예측 값에 가산하여 샘플 값을 재구성하도록 구성되는, 비디오 디코딩 장치.

### 청구항 17

제 16 항에 있어서,

상기 하나 이상의 프로세서들은 비트스트림으로부터, 수정된 양자화된 잔여 값  $Q(\tilde{r}_{i,j})$ 를 나타내는 하나 이상의 선택스 엘리먼트들을 획득하도록 구성되는, 비디오 디코딩 장치.

### 청구항 18

제 16 항에 있어서,

상기 블록이 상기 수직 인트라 예측 모드를 사용하여 코딩되면, 상기 수정된 잔여 값  $\tilde{r}_{i,j}$ 는 다음과 같이 정의되고:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i = 0, 0 \leq j \leq (N-1) \\ r_{i,j} - Q(r_{(i-1),j}), & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases},$$

상기 블록이 상기 수평 인트라 예측 모드를 사용하여 코딩되면, 상기 수정된 잔여 값  $\tilde{r}_{i,j}$ 는 다음과 같이 정의되는:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & 0 \leq i \leq (M-1), j = 0 \\ r_{i,j} - Q(r_{i,(j-1)}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}, \text{비디오 디코딩 장치.}$$

### 청구항 19

제 16 항에 있어서,

상기 블록이 HEVC (High Efficiency Video Coding) 작업 초안 10에서 정의된 바와 같이 22와 30 사이의 인트라 예측 모드를 사용하여 코딩되면, 상기 양자화된 수정된 잔여 값  $Q(\tilde{r}_{i,j})$ 는 다음과 같이 정의되고:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & i = 0, 0 \leq j \leq (N-1), \\ Q(\tilde{r}_{i,j}) + Q(r_{i-1,j}) & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases},$$

상기 블록이 HEVC 작업 초안 10에서 정의된 바와 같이 6과 14 사이의 인트라 예측 모드를 사용하여 코딩되면,  $Q(\tilde{r}_{i,j})$ 는 다음과 같이 정의되는:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & 0 \leq i \leq (M-1), j = 0, \\ Q(\tilde{r}_{i,j}) + Q(r_{i,j-1}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}, \text{비디오 디코딩 장치.}$$

### 청구항 20

제 16 항에 있어서,

상기 제 2 블록이 대각선 인트라 예측 모드를 사용하여 코딩되면,  $Q(r_{i,j})$ 는 다음과 같이 정의되며:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & i=0, 0 \leq j \leq (N-1) \\ Q(\tilde{r}_{i,j}), & 1 \leq i \leq (M-1), j=0 \\ Q(\tilde{r}_{i,j}) + Q(r_{i-1,j-1}), & 1 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}$$

$Q(r_{i-1,j-1})$  은 상기 잔여 값  $r_{i,j}$  의 한 칼럼 좌측 및 한 로우 상부에 있는 잔여 값에 대한 재구성된 잔여 값인, 비디오 디코딩 장치.

### 청구항 21

제 20 항에 있어서,

상기 수정된 잔여 값  $\tilde{r}_{i,j}$  는 다음과 같이 정의되는:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ r_{i,j}, & 1 \leq i \leq (M-1), j=0 \\ r_{i,j} - Q(r_{i-1,j-1}), & 1 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}$$

, 비디오 디코딩 장치.

### 청구항 22

제 16 항에 있어서,

상기 잔여 값  $r_{i,j}$  는 비트-시프트된 잔여 값인, 비디오 디코딩 장치.

### 청구항 23

제 16 항에 있어서,

상기 비디오 디코딩 장치는,

집적 회로;

마이크로프로세서; 및

코더를 포함하는 무선 통신 디바이스 중 적어도 하나를 포함하는, 비디오 디코딩 장치.

### 청구항 24

비디오 인코딩 장치로서,

데이터를 저장하는 메모리; 및

하나 이상의 프로세서들을 포함하고,

상기 하나 이상의 프로세서들은,

$M$  이 블록의 높이이고  $N$  이 상기 블록의 폭이고, 상기 블록은 변환 스Kim 블록인,  $0 \leq i \leq (M-1)$  및  $0 \leq j \leq (N-1)$  에 대해, 상기 하나 이상의 프로세서들이,

잔여 값  $r_{i,j}$  에 대한 수정된 잔여 값  $\tilde{r}_{i,j}$  을 결정하는 것으로서,

상기 블록이 수직 인트라 예측 모드를 사용하여 코딩되면,  $\tilde{r}_{i,j}$  는 다음과 같이 정의되며:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ r_{i,j} - Q(r_{(i-1),j}), & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases}$$

$Q(r_{(i-1),j})$  는 상기 잔여 값  $r_{i,j}$  의 한 칼럼 좌측에 있는 잔여 값  $r_{i-1,j}$  에 대한 재구성된 잔여 값을 나타내고,

상기 블록이 수평 인트라 예측 모드를 사용하여 코딩되면,  $\tilde{r}_{i,j}$  는 다음과 같이 정의되며:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & 0 \leq i \leq (M-1), j=0 \\ r_{i,j} - Q(r_{i,(j-1)}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases},$$

$Q(r_{i,(j-1)})$  는 상기 잔여 값  $r_{i,j}$  의 한 로우 상부에 있는 잔여 값  $r_{i,j-1}$ 에 대한 재구성된 잔여 값을 나타내는,

잔여 값  $r_{i,j}$ 에 대한 수정된 잔여 값  $\tilde{r}_{i,j}$ 을 결정하며;

상기 수정된 잔여 값  $\tilde{r}_{i,j}$ 을 양자화하여 양자화된 수정된 잔여 값  $Q(\tilde{r}_{i,j})$ 을 생성하도록 구성되는, 비디오 인코딩 장치.

### 청구항 25

제 24 항에 있어서,

상기 양자화된 수정된 잔여 값  $Q(\tilde{r}_{i,j})$ 을 비트스트림으로 시그널링하는 것을 더 포함하는, 비디오 인코딩 장치.

### 청구항 26

제 24 항에 있어서,

상기 하나 이상의 프로세서들은 재구성된 잔여 값  $Q(r_{i,j})$ 를 계산하도록 구성되고,

상기 블록이 상기 수직 인트라 예측 모드를 사용하여 코딩되면, 상기 재구성된 잔여 값  $Q(r_{i,j})$ 은 다음과 같이 정의되고:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & i=0, 0 \leq j \leq (N-1), \\ Q(\tilde{r}_{i,j}) + Q(r_{i-1,j}) & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases},$$

상기 블록이 상기 수평 인트라 예측 모드를 사용하여 코딩되면, 상기 재구성된 잔여 값  $Q(r_{i,j})$ 은 다음과 같이 정의되는:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & 0 \leq i \leq (M-1), j=0, \\ Q(\tilde{r}_{i,j}) + Q(r_{i,j-1}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}, \text{비디오 인코딩 장치.}$$

### 청구항 27

제 26 항에 있어서,

상기 하나 이상의 프로세서들은,

상기 재구성된 잔여 값  $Q(r_{i,j})$ 를 예측 값에 가산하여 샘플 값을 재구성하며;

다른 블록들의 인트라 예측 또는 인터 예측을 위해 상기 샘플 값을 사용하도록 구성되는, 비디오 인코딩 장치.

### 청구항 28

제 24 항에 있어서,

상기 블록이 HEVC (High Efficiency Video Coding) 작업 초안 10 에서 정의된 바와 같이 22 와 30 사이의 인트라 예측 모드를 사용하여 코딩되면, 상기 양자화된 수정된 잔여 값  $Q(\tilde{r}_{i,j})$  는 다음과 같이 정의되고:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & i=0, 0 \leq j \leq (N-1), \\ Q(\tilde{r}_{i,j}) + Q(r_{i-1,j}) & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases}$$

상기 블록이 HEVC 작업 초안 10 에서 정의된 바와 같이 6 과 14 사이의 인트라 예측 모드를 사용하여 코딩되면,  $Q(\tilde{r}_{i,j})$  는 다음과 같이 정의되는:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & 0 \leq i \leq (M-1), j=0, \\ Q(\tilde{r}_{i,j}) + Q(r_{i,j-1}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}, \text{ 비디오 인코딩 장치.}$$

### 청구항 29

제 24 항에 있어서,

상기 블록이 대각선 인트라 예측 모드를 사용하여 코딩되면,  $\tilde{r}_{i,j}$  는 다음과 같이 정의되며:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ r_{i,j}, & 1 \leq i \leq (M-1), j=0 \\ r_{i,j} - Q(r_{i-1,j-1}), & 1 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}$$

$Q(r_{i-1,j-1})$  은 상기 잔여 값  $r_{i,j}$  의 한 칼럼 좌측 및 한 로우 상부에 있는 잔여 값  $r_{i-1,j-1}$  에 대한 재구성된 잔여 값인, 비디오 인코딩 장치.

### 청구항 30

제 29 항에 있어서,

상기 하나 이상의 프로세서들은,

상기 잔여 값  $r_{i,j}$  에 대한 재구성된 잔여 값  $Q(r_{i,j})$  를 계산하는 것으로서,

상기 블록이 상기 대각선 인트라 예측 모드를 사용하여 코딩되면, 상기 재구성된 잔여 값  $Q(r_{i,j})$  는 다음과 같이 정의되며:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & i=0, 0 \leq j \leq (N-1) \\ Q(\tilde{r}_{i,j}), & 1 \leq i \leq (M-1), j=0 \\ Q(\tilde{r}_{i,j}) + Q(r_{i-1,j-1}), & 1 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}$$

$Q(r_{i-1,j-1})$  은 상기 잔여 값  $r_{i,j}$  의 한 칼럼 좌측 및 한 로우 상부에 있는 잔여 값의 재구성된 버전인, 상기 잔여 값  $r_{i,j}$  에 대한 재구성된 잔여 값  $Q(r_{i,j})$  를 계산하며;

상기 재구성된 잔여 값  $r_{i,j}$  를 예측 값에 가산하여 샘플 값을 재구성하도록 구성되는, 비디오 인코딩 장치.

### 청구항 31

제 24 항에 있어서,

상기 잔여 값  $r_{i,j}$  는 비트-시프트된 잔여 값인, 비디오 인코딩 장치.

**청구항 32**

제 24 항에 있어서,  
상기 비디오 인코딩 장치는,  
집적 회로;  
마이크로프로세서; 및  
코더를 포함하는 무선 통신 디바이스 중 적어도 하나를 포함하는, 비디오 인코딩 장치.

**청구항 33**

비디오 디코딩 장치로서,  
잔여 값들의 블록을 생성하는 수단으로서,  
상기 블록은 변환 스kip 블록이고, 상기 블록을 생성하는 것은,  
M 이 상기 블록의 높이이고 N 이 상기 블록의 폭인, 0 ≤ i ≤ (M - 1) 및 0 ≤ j ≤ (N - 1)에 대해, 상기 블록에서 잔여 값  $r_{i,j}$ 에 대한 재구성된 잔여 값  $Q(r_{i,j})$ 를 계산하는 것을 포함하고,  
상기 블록이 수직 인트라 예측 모드를 사용하여 코딩되면,  $Q(r_{i,j})$ 는 다음과 같이 정의되며:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & i=0, 0 \leq j \leq (N-1), \\ Q(\tilde{r}_{i,j}) + Q(r_{i-1,j}) & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases},$$

$Q(\tilde{r}_{i,j})$  는 수정된 잔여 값  $\tilde{r}_{i,j}$ 의 양자화된 버전을 나타내고, 상기 수정된 잔여 값  $\tilde{r}_{i,j}$ 는 상기 잔여 값  $r_{i,j}$ 의 수정된 버전이며,  $Q(r_{i-1,j})$ 는 상기 잔여 값  $r_{i,j}$ 의 한 칼럼 좌측에 있는 잔여 값에 대한 재구성된 잔여 값이고,

상기 블록이 수평 인트라 예측 모드를 사용하여 코딩되면,  $Q(r_{i,j})$ 는 다음과 같이 정의되며:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & 0 \leq i \leq (M-1), j=0, \\ Q(\tilde{r}_{i,j}) + Q(r_{i,j-1}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases},$$

$Q(r_{i,j-1})$ 은 상기 잔여 값  $r_{i,j}$ 의 한 로우 상부에 있는 잔여 값에 대한 재구성된 잔여 값인, 상기 잔여 값들의 블록을 생성하는 수단; 및

0 ≤ i ≤ (M - 1) 및 0 ≤ j ≤ (N - 1)에 대해, 상기 재구성된 잔여 값  $Q(r_{i,j})$ 를 예측 값에 가산하여 샘플 값을 재구성하는 수단을 포함하는, 비디오 디코딩 장치.

**청구항 34**

비디오 인코딩 장치로서,  
M 이 블록의 높이이고 N 이 상기 블록의 폭이고, 상기 블록은 변환 스kip 블록인, 0 ≤ i ≤ (M - 1) 및 0 ≤ j ≤ (N - 1)에 대해,

잔여 값  $r_{i,j}$ 에 대한 수정된 잔여 값  $\tilde{r}_{i,j}$ 을 결정하는 수단으로서,

상기 블록이 수직 인트라 예측 모드를 사용하여 코딩되면,  $\tilde{r}_{i,j}$ 는 다음과 같이 정의되며:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ r_{i,j} - Q(r_{(i-1),j}), & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases}$$

$Q(r_{(i-1),j})$  는 상기 잔여 값  $r_{i,j}$  의 한 칼럼 좌측에 있는 잔여 값  $r_{i-1,j}$ 에 대한 재구성된 잔여 값을 나타내고,

상기 블록이 수평 인트라 예측 모드를 사용하여 코딩되면,  $\tilde{r}_{i,j}$ 는 다음과 같이 정의되며:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & 0 \leq i \leq (M-1), j=0 \\ r_{i,j} - Q(r_{i,(j-1)}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}$$

$Q(r_{i,(j-1)})$  는 상기 잔여 값  $r_{i,j}$ 의 한 로우 상부에 있는 잔여 값  $r_{i,j-1}$ 에 대한 재구성된 잔여 값을 나타내는,

상기 잔여 값  $r_{i,j}$ 에 대한 수정된 잔여 값  $\tilde{r}_{i,j}$ 을 결정하는 수단; 및

상기 수정된 잔여 값  $\tilde{r}_{i,j}$ 을 양자화하여 양자화된 수정된 잔여 값  $Q(\tilde{r}_{i,j})$ 을 생성하는 수단을 포함하는, 비디오 인코딩 장치.

### 청구항 35

명령들이 저장되어 있는 컴퓨터 판독가능 데이터 저장 매체로서,

상기 명령들은, 실행되는 경우 비디오 디코딩 장치로 하여금, 잔여 값들의 블록을 생성하게 하는 것으로서,

상기 블록은 스kip 블록이고, 상기 블록을 생성하는 것은,

M이 상기 블록의 높이이고 N이 상기 블록의 폭인,  $0 \leq i \leq (M-1)$  및  $0 \leq j \leq (N-1)$ 에 대해, 상기 블록에서 잔여 값  $r_{i,j}$ 에 대한 재구성된 잔여 값  $Q(r_{i,j})$ 를 계산하는 것을 포함하고,

상기 블록이 수직 인트라 예측 모드를 사용하여 코딩되면,  $Q(r_{i,j})$ 는 다음과 같이 정의되며:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & i=0, 0 \leq j \leq (N-1), \\ Q(\tilde{r}_{i,j}) + Q(r_{i-1,j}) & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases}$$

$Q(\tilde{r}_{i,j})$ 는 수정된 잔여 값  $\tilde{r}_{i,j}$ 의 양자화된 버전을 나타내고, 상기 수정된 잔여 값  $\tilde{r}_{i,j}$ 는 상기 잔여 값  $r_{i,j}$ 의 수정된 버전이며,  $Q(r_{i-1,j})$ 는 상기 잔여 값  $r_{i,j}$ 의 한 칼럼 좌측에 있는 잔여 값에 대한 재구성된 잔여 값이고,

상기 블록이 수평 인트라 예측 모드를 사용하여 코딩되면,  $Q(r_{i,j})$ 는 다음과 같이 정의되며:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & 0 \leq i \leq (M-1), j=0, \\ Q(\tilde{r}_{i,j}) + Q(r_{i,j-1}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}$$

$Q(r_{i,j-1})$ 은 상기 잔여 값  $r_{i,j}$ 의 한 로우 상부에 있는 잔여 값에 대한 재구성된 잔여 값인, 상기 잔여 값들의 블록을 생성하게 하며;

$0 \leq i \leq (M-1)$  및  $0 \leq j \leq (N-1)$ 에 대해, 상기 재구성된 잔여 값  $Q(r_{i,j})$ 를 예측 값에 가산하여 샘플 값을 재구성하게 하는, 컴퓨터 판독가능 데이터 저장 매체.

### 청구항 36

명령들이 저장되어 있는 컴퓨터 관독가능 데이터 저장 매체로서,  
상기 명령들은, 실행되는 경우 비디오 인코딩 장치로 하여금,  
 $M$  이 블록의 높이이고  $N$  이 상기 블록의 폭이고, 상기 블록은 변환 스윕 블록인,  $0 \leq i \leq (M - 1)$  및  $0 \leq j \leq (N - 1)$ 에 대해,

잔여 값  $r_{i,j}$ 에 대한 수정된 잔여 값  $\tilde{r}_{i,j}$ 을 결정하게 하는 것으로서,

상기 블록이 수직 인트라 예측 모드를 사용하여 코딩되면,  $\tilde{r}_{i,j}$ 는 다음과 같이 정의되며:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i = 0, 0 \leq j \leq (N - 1) \\ r_{i,j} - Q(r_{(i-1),j}), & 1 \leq i \leq (M - 1), 0 \leq j \leq (N - 1) \end{cases},$$

$Q(r_{(i-1),j})$ 는 상기 잔여 값  $r_{i,j}$ 의 한 칼럼 좌측에 있는 잔여 값  $r_{i-1,j}$ 에 대한 재구성된 잔여 값을 나타내고,

상기 블록이 수평 인트라 예측 모드를 사용하여 코딩되면,  $\tilde{r}_{i,j}$ 는 다음과 같이 정의되며:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & 0 \leq i \leq (M - 1), j = 0 \\ r_{i,j} - Q(r_{i,(j-1)}) & 0 \leq i \leq (M - 1), 1 \leq j \leq (N - 1) \end{cases},$$

$Q(r_{i,(j-1)})$ 는 상기 잔여 값  $r_{i,j}$ 의 한 로우 상부에 있는 잔여 값  $r_{i,j-1}$ 에 대한 재구성된 잔여 값을 나타내는,

상기 잔여 값  $r_{i,j}$ 에 대한 수정된 잔여 값  $\tilde{r}_{i,j}$ 을 결정하게 하며;

상기 수정된 잔여 값  $\tilde{r}_{i,j}$ 을 양자화하여 양자화된 수정된 잔여 값  $Q(\tilde{r}_{i,j})$ 을 생성하게 하는, 컴퓨터 관독가능 데이터 저장 매체.

## 발명의 설명

### 기술 분야

[0001] 본 출원은 2013년 3월 25일자에 출원된 미국 가특허 출원번호 제 61/805,094호, 2013년 4월 5일자에 출원된 미국 가특허 출원번호 제 61/809,199호, 2013년 4월 5일자에 출원된 미국 가특허 출원번호 제 61/809,203호, 2013년 4월 8일자에 출원된 미국 가특허 출원번호 제 61/809,811호, 2013년 4월 8일자에 출원된 미국 가특허 출원번호 제 61/809,870호, 2013년 4월 9일자에 출원된 미국 가특허 출원번호 제 61/810,179호, 2013년 4월 9일자에 출원된 미국 가특허 출원번호 제 61/810,218호, 및 2013년 7월 5일자에 출원된 미국 가특허 출원번호 제 61/843,144호의 이익을 주장하고, 이의 각각의 전체 내용이 참조로 포함된다.

### 기술 분야

[0003] 본 개시물은 비디오 코딩 및 압축에 관한 것이다.

### 배경 기술

[0004] 디지털 비디오 능력들은 디지털 텔레비전, 디지털 직접 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인 휴대정보 단말기들 (PDAs), 랩탑 또는 데스크탑 컴퓨터들, 디지털 카메라들, 디지털 리코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 무선 전화기들, 원격 화상회의 디바이스들 등을 포함한, 광범위한 디바이스들에 포함될 수 있다. 디지털 비디오 디바이스들은 디지털 비디오 정보를 좀더 효율적으로 송수신하고 저장하기 위해, MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, 파트 10, AVC (Advanced Video Coding), HEVC (High Efficiency Video Coding) 표준에 의해 정

의된 표준들, 및 이런 표준들의 확장판들에 설명된 것들과 같은 비디오 압축 기법들을 구현한다.

[0005] 비디오 압축 기법들은 비디오 시퀀스들에 고유한 리던던시를 감소시키거나 또는 제거하기 위해 공간 (인트라-화상) 예측 및/또는 시간 (인터-화상) 예측을 수행한다. 블록-기반 비디오 코딩에 있어, 비디오 슬라이스는 비디오 블록들로 파티셔닝될 수도 있고, 이 비디오 블록들은 또한 트리블록들, 코딩 유닛들 (CUs) 및/또는 코딩 노드들로서 지칭될 수도 있다. 화상의 인트라-코딩된 (I) 슬라이스에서 비디오 블록들은 동일한 화상에서 이웃하는 블록들에서의 참조 샘플들에 대한 공간 예측을 사용하여 인코딩된다. 화상의 인터-코딩된 (P 또는 B) 슬라이스에서 비디오 블록들은 동일한 화상에서 이웃하는 블록들에서의 참조 샘플들에 대한 공간 예측, 또는 다른 참조 화상들에서의 참조 샘플들에 대한 시간 예측을 사용할 수도 있다. 화상들은 프레임들로 지칭될 수 있고, 참조 화상들은 참조 프레임들로서 지칭될 수도 있다.

## 발명의 내용

### 과제의 해결 수단

[0006] 일반적으로, 본 개시물의 기법들은 비디오 코딩에서 인트라 예측에 관한 것이다. 본원에서 설명하는 바와 같이, 비디오 코더는 손실 코딩을 사용하여 코딩된 블록의 잔여 데이터에 잔여 차분 펄스 코드 변조 기법 (residual differential pulse code modulation technique)를 적용한다. 블록은 잔여 데이터에 대한 변환의 적용 없이 코딩될 수도 있다.

[0007] 일 예에서, 본 개시물은 비디오 데이터를 디코딩하는 방법을 기술하고, 상기 방법은 잔여 값들의 블록을 생성하는 단계를 포함하고, 여기서, 블록은 변환 스kip 블록이며; 상기 블록을 생성하는 단계는,  $0 \leq i \leq (M - 1)$  및  $0 \leq j \leq (N - 1)$ 에 대해, 블록에서 잔여 값  $r_{i,j}$ 에 대한 재구성된 잔여 값  $Q(r_{i,j})$ 를 계산하는 단계, 및  $0 \leq i \leq (M - 1)$  및  $0 \leq j \leq (N - 1)$ 에 대해, 재구성된 잔여 값  $Q(r_{i,j})$ 를 예측 값에 가산하여 샘플 값을 재구성하는 단계를 포함하고, 여기서,  $M$ 은 블록의 높이이고  $N$ 은 블록의 폭이고, 여기서, 블록이 수직 인트라 예

측 모드를 사용하여 코딩되면,  $Q(r_{i,j})$ 는 
$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & i=0, 0 \leq j \leq (N-1), \\ Q(\tilde{r}_{i,j}) + Q(r_{i-1,j}) & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases}$$
 과 같이 정의되고, 블록이 수평 인트라 예측 모드를 사용하여 코딩되면,  $Q(r_{i,j})$ 는

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & 0 \leq i \leq (M-1), j=0, \\ Q(\tilde{r}_{i,j}) + Q(r_{i,j-1}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}$$
 과 같이 정의되고, 여기서,  $\tilde{Q}(\tilde{r}_{i,j})$ 는 수정된

잔여 값  $\tilde{r}_{i,j}$ 의 양자화된 베타를 나타내고, 수정된 잔여 값  $\tilde{r}_{i,j}$ 는 잔여 값  $r_{i,j}$ 의 수정된 베타이고,  $Q(r_{i,j})$ 는 잔여 값  $r_{i,j}$ 의 한 칼럼 좌측에 있는 잔여 값에 대한 재구성된 잔여 값이고, 여기서,  $Q(r_{i,j-1})$ 은 잔여 값  $r_{i,j}$ 의 한 로우 상부에 있는 잔여 값에 대한 재구성된 잔여 값이다.

[0008] 또 다른 예에서, 본 개시물은 비디오 데이터를 인코딩하는 방법을 기술하고, 상기 방법은,  $0 \leq i \leq (M - 1)$  및  $0 \leq j \leq (N - 1)$ 에 대해, 잔여 값  $r_{i,j}$ 에 대한 수정된 잔여 값  $\tilde{r}_{i,j}$ 을 결정하는 단계; 및 수정된 잔여

값  $\tilde{r}_{i,j}$ 을 양자화하여 양자화된 수정된 잔여 값  $Q(\tilde{r}_{i,j})$ 을 생성하는 단계를 포함하고, 여기서,  $M$ 은 블록의 높이이고  $N$ 은 블록의 폭이고, 여기서, 블록은 변환 스kip 블록이고, 여기서, 블록이 수직 인트라 예측 모드를

사용하여 코딩되면,  $\tilde{r}_{i,j}$ 는 
$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ r_{i,j} - Q(r_{(i-1),j}), & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases}$$
 과 같이

정의되며, 블록이 수평 인트라 예측 모드를 사용하여 코딩되면,  $\tilde{r}_{i,j}$ 는

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & 0 \leq i \leq (M-1), j=0 \\ r_{i,j} - Q(r_{i,(j-1)}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}$$

과 같이 정의되고, 여기서,  $Q(r_{(i-1,j)})$  는 잔여 값  $r_{i,j}$  의 한 칼럼 좌측에 있는 잔여 값  $r_{i-1,j}$  에 대한 재구성된 잔여 값을 나타내고, 여기서,  $Q(r_{i,(j-1)})$  는 잔여 값  $r_{i,j}$  의 한 로우 상부에 있는 잔여 값  $r_{i,j-1}$  에 대한 재구성된 잔여 값을 나타낸다.

[0009] 또 다른 예에서, 본 개시물은, 비디오 디코딩 장치를 기술하고, 상기 비디오 디코딩 장치는 데이터를 저장하는 메모리; 및 잔여 값들의 블록을 생성하도록 구성된 하나 이상의 프로세서들을 포함하고, 여기서, 블록은 변환 스킵 블록이고, 블록을 생성하기 위해, 하나 이상의 계산기들은  $0 \leq i \leq (M-1)$  및  $0 \leq j \leq (N-1)$ 에 대해, 블록에서 잔여 값  $r_{i,j}$  에 대한 재구성된 잔여 값  $Q(r_{i,j})$  를 계산하며,,  $0 \leq i \leq (M-1)$  및  $0 \leq j \leq (N-1)$ 에 대해, 샘플 값을 재구성하기 위해 잔여 값  $r_{i,j}$  를 예측 값에 가산하고, 여기서,  $M$  은 블록의 높이이고  $N$  은 블록의 폭이고, 여기서, 블록이 수직 인트라 예측 모드를 사용하여 코딩되면,  $Q(r_{i,j})$  는 다음과 같이 정의

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & i=0, 0 \leq j \leq (N-1), \\ Q(\tilde{r}_{i,j}) + Q(r_{i-1,j}) & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases}$$

되며: 그리고, 블록이 수평 인트라 예측 모드를 사용하여 코딩되면,  $Q(r_{i,j})$  는 다음과 같이 정의되며:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & 0 \leq i \leq (M-1), j=0, \\ Q(\tilde{r}_{i,j}) + Q(r_{i,j-1}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}$$

, 여기서,  $Q(\tilde{r}_{i,j})$  는 수정된 잔여 값  $\tilde{r}_{i,j}$  의

양자화된 버전을 나타내고, 수정된 잔여 값  $\tilde{r}_{i,j}$  는 잔여 값  $r_{i,j}$  의 수정된 버전이고,  $Q(r_{i-1,j})$  는 잔여 값  $r_{i-1,j}$  의 한 칼럼 좌측에 있는 잔여 값에 대한 재구성된 잔여 값이고, 여기서,  $Q(r_{i,j-1})$  은 잔여 값  $r_{i,j}$  의 한 로우 상부에 있는 잔여 값에 대한 재구성된 잔여 값이다.

[0010] 또 다른 예에서, 본 개시물은 비디오 디코딩 장치를 기술하고, 상기 비디오 디코딩 장치는 잔여 값들의 블록을 생성하는 수단; 및  $0 \leq i \leq (M-1)$  및  $0 \leq j \leq (N-1)$  샘플 값을 재구성하기 위해 잔여 값  $r_{i,j}$  를 예측 값에 가산하는 수단을 포함하고, 여기서, 블록은 변환 스킵 블록이고, 여기서, 블록을 생성하는 것은  $0 \leq i \leq (M-1)$  및  $0 \leq j \leq (N-1)$ 에 대해, 블록에서 잔여 값  $r_{i,j}$  에 대한 재구성된 잔여 값  $Q(r_{i,j})$  를 계산하는 것을 포함하고, 여기서,  $M$  은 블록의 높이이고  $N$  은 블록의 폭이고, 여기서, 블록이 수직 인트라 예측 모드를 사용하여 코딩되면,  $Q(r_{i,j})$  는 다음과 같이 정의되며:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & i=0, 0 \leq j \leq (N-1), \\ Q(\tilde{r}_{i,j}) + Q(r_{i-1,j}) & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases}$$

, 그리고, 블록이 수평 인트라 예측 모드를 사용하여 코딩되면,  $Q(r_{i,j})$  는 다음과 같이 정의되며:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & 0 \leq i \leq (M-1), j=0, \\ Q(\tilde{r}_{i,j}) + Q(r_{i,j-1}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}$$

여기서,  $Q(\tilde{r}_{i,j})$  는 수정된 잔여 값  $\tilde{r}_{i,j}$  의 양자화된 버전을 나타내고, 수정된 잔여 값  $\tilde{r}_{i,j}$  는 잔여 값  $r_{i,j}$  의 수정된 버전이고,  $Q(r_{i-1,j})$  는 잔여 값  $r_{i-1,j}$  의 한 칼럼 좌측에 있는 잔여 값에 대한 재구성된 잔여 값이고, 여기서,  $Q(r_{i,j-1})$  은 잔여 값  $r_{i,j}$  의 한 로우 상부에 있는 잔여 값에 대한 재구성된 잔여 값이다.

[0011] 또 다른 예에서, 본 개시물은 비디오 인코딩 장치를 기술하고, 상기 비디오 인코딩 장치는, 데이터를 저장하는 메모리; 및  $0 \leq i \leq (M-1)$  및  $0 \leq j \leq (N-1)$ 에 대해, 하나 이상의 프로세서들이 잔여 값  $r_{i,j}$  에 대한

수정된 잔여 값  $\tilde{r}_{i,j}$  을 결정하며, 수정된 잔여 값  $\tilde{r}_{i,j}$  을 양자화하여 양자화된 수정된 잔여 값  $Q(\tilde{r}_{i,j})$  을 생성하도록 구성된 하나 이상의 프로세서들을 포함하고, 여기서,  $M$  은 블록의 높이이고  $N$  은 블록의 폭이고, 여

기서, 블록은 변환 스kip 블록이고, 여기서, 블록이 수직 인트라 예측 모드를 사용하여 코딩되면,  $\tilde{r}_{i,j}$  는 다음

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ r_{i,j} - Q(r_{(i-1),j}), & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases}, \text{ 그리고 블록이 수평 인트라}$$

예측 모드를 사용하여 코딩되면,  $\tilde{r}_{i,j}$  는 다음과 같이 정의되며:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & 0 \leq i \leq (M-1), j=0 \\ r_{i,j} - Q(r_{i,(j-1)}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}, \text{ 여기서, } Q(r_{(i-1),j}) \text{ 는 잔여 값 } r_{i,j} \text{ 의 한 칼럼 좌측에 있는 잔여 값 } r_{i-1,j} \text{ 에 대한 재구성된 잔여 값을 나타내고, 여기서, } Q(r_{i,(j-1)}) \text{ 는 잔여 값 } r_{i,j} \text{ 의 한 로우 상부에 있는 잔여 값 } r_{i,j-1} \text{ 에 대한 재구성된 잔여 값을 나타낸다.}$$

[0012] 또 다른 예에서, 본 개시물은 비디오 인코딩 장치를 기술하고, 상기 비디오 인코딩 장치는,  $0 \leq i \leq (M-1)$

및  $0 \leq j \leq (N-1)$  에 대해, 잔여 값  $r_{i,j}$  에 대한 수정된 잔여 값  $\tilde{r}_{i,j}$  을 결정하는 수단; 및 수정된 잔여

값  $\tilde{r}_{i,j}$  을 양자화하여 양자화된 수정된 잔여 값  $Q(\tilde{r}_{i,j})$  을 생성하는 수단을 포함하고, 여기서,  $M$  은 블록의 높이이고  $N$  은 블록의 폭이고, 블록은 변환 스kip 블록이며: 여기서, 블록이 수직 인트라 예측 모드를 사용하여

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ r_{i,j} - Q(r_{(i-1),j}), & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases}, \text{ 그}$$

리고 블록이 수평 인트라 예측 모드를 사용하여 코딩되면,  $\tilde{r}_{i,j}$  는 다음과 같이 정의되며:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & 0 \leq i \leq (M-1), j=0 \\ r_{i,j} - Q(r_{i,(j-1)}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}, \text{ 여기서, } Q(r_{(i-1),j}) \text{ 는 잔여 값 } r_{i,j} \text{ 의 한 칼럼 좌측에 있는 잔여 값 } r_{i-1,j} \text{ 에 대한 재구성된 잔여 값을 나타내고, 여기서, } Q(r_{i,(j-1)}) \text{ 는 잔여 값 } r_{i,j} \text{ 의 한 로우 상부에 있는 잔여 값 } r_{i,j-1} \text{ 에 대한 재구성된 잔여 값을 나타낸다.}$$

[0013] 또 다른 예에서, 본 개시물은 명령들이 저장되어 있는 컴퓨터 판독가능 저장 매체를 기술하고, 상기 명령들은, 실행되는 경우 비디오 디코딩 장치로 하여금, 잔여 값들의 블록을 생성하게 하며,,  $0 \leq i \leq (M-1)$  및  $0 \leq j \leq (N-1)$  에 대해, 재구성된 잔여 값  $Q(r_{i,j})$  를 예측 값에 가산하여 샘플 값을 재구성하게 하고, 여기서, 블록은 변환 스kip 블록이고, 블록을 생성하는 것은  $0 \leq i \leq (M-1)$  및  $0 \leq j \leq (N-1)$  에 대해, 블록에서 잔여 값  $r_{i,j}$  에 대한 재구성된 잔여 값  $Q(r_{i,j})$  를 계산하는 것을 포함하고, 여기서,  $M$  은 블록의 높이이고  $N$  은 블록의 폭이고, 여기서, 블록이 수직 인트라 예측 모드를 사용하여 코딩되면,  $Q(r_{i,j})$  는 다음과 같이 정의되며:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & i=0, 0 \leq j \leq (N-1), \\ Q(\tilde{r}_{i,j}) + Q(r_{i-1,j}) & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases}, \text{ 그리고, 블록이 수평 인트라 예측 모드를 사용하여}$$

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & 0 \leq i \leq (M-1), j=0, \\ Q(\tilde{r}_{i,j}) + Q(r_{i,j-1}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}, \text{ 코딩되면, } Q(r_{i,j}) \text{ 는 다음과 같이 정의되며: }$$

여기서,  $Q(\tilde{r}_{i,j})$  는 수정된 잔여 값  $\tilde{r}_{i,j}$  의 양자화된 버전을 나타내고, 수정된 잔여 값  $\tilde{r}_{i,j}$  는 잔여 값  $r_{i,j}$  의 수정된 버전이고,  $Q(r_{i-1,j})$  는 잔여 값  $r_{i,j}$  의 한 칼럼 좌측에 있는 잔여 값에 대한 재구성된 잔여 값이고, 여기서,  $Q(r_{i,j-1})$  은 잔여 값  $r_{i,j}$  의 한 로우 상부에 있는 잔여 값에 대한 재구성된 잔여 값이다.

[0014]

또 다른 예에서, 본 개시물은 명령들이 저장되어 있는 컴퓨터 관독가능 저장 매체를 기술하고, 상기 명령들은, 실행되는 경우 비디오 인코딩 장치로 하여금,  $0 \leq i \leq (M - 1)$  및  $0 \leq j \leq (N - 1)$ 에 대해, 잔여 값  $r_{i,j}$

에 대한 수정된 잔여 값  $\tilde{r}_{i,j}$ 을 결정하게 하며, 수정된 잔여 값  $\tilde{r}_{i,j}$ 을 양자화하여 양자화된 수정된 잔여 값

$Q(\tilde{r}_{i,j})$ 을 생성하게 하고, 여기서,  $M$ 은 블록의 높이이고  $N$ 은 블록의 폭이고, 여기서, 블록은 변환 스킵 블록이고, 여기서, 블록이 수직 인트라 예측 모드를 사용하여 코딩되면,  $\tilde{r}_{i,j}$ 는 다음과 같이 정의되며:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i = 0, 0 \leq j \leq (N-1) \\ r_{i,j} - Q(r_{(i-1),j}), & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases}, \text{ 그리고, 블록이 수평 인트라 예측 모드를 사용}$$

하여 코딩되면,  $\tilde{r}_{i,j}$ 는 다음과 같이 정의되며:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & 0 \leq i \leq (M-1), j = 0 \\ r_{i,j} - Q(r_{(i-1),j}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}, \text{ 여기서, } Q(r_{(i-1),j}) \text{는 잔여 값 } r_{i,j} \text{의 한 칼럼 좌측에 있는 잔여 값 } r_{i-1,j} \text{에 대한 재구성된 잔여 값을 나타내고, 여기서, } Q(r_{(i-1),j}) \text{는 잔여 값 } r_{i,j} \text{의 한 로우 상부에 있는 잔여 값 } r_{i,j-1} \text{에 대한 재구성된 잔여 값을 나타낸다.}$$

[0015]

본 개시물의 하나 이상의 예들의 세부 사항들은 첨부도면 및 아래의 상세한 설명에서 개시된다. 다른 특성들, 목적들, 및 이점들은 설명, 도면들, 및 청구범위로부터 명백히 알 수 있을 것이다.

### 도면의 간단한 설명

[0016]

도 1은 본 개시물의 기법들을 사용할 수도 있는 예시적인 비디오 코딩 시스템을 예시하는 블록도이다.

도 2는 사이즈  $M$ (높이)  $\times N$ (폭)의 블록을 예시하는 개념도이다.

도 3은 예시적인 인트라 예측 모드 방향들을 예시하는 개념도이다.

도 4는 비디오 코딩에서의 예측에 사용될 수도 있는 예시적인 샘플들을 예시하는 개념도이다.

도 5a는 근사-수직 모드들 (near-vertical modes)에 대한 잔여 차분 월스 코드 변조 (DPCM) 방향을 나타낸다.

도 5b는 근사-수평 모드들 (near-horizontal modes)에 대한 잔여 DPCM 방향을 나타낸다.

도 6은 본 개시물의 기법들을 구현할 수도 있는 예시적인 비디오 인코더를 예시하는 블록도이다.

도 7은 본 개시물의 기법들을 구현할 수도 있는 예시적인 비디오 디코더를 예시하는 블록도이다.

도 8a는 본 개시물의 하나 이상의 기법들에 따른, 비디오 인코더의 예시적인 동작을 예시하는 플로우차트이다.

도 8b는 본 개시물의 하나 이상의 기법들에 따른, 비디오 인코더의 예시적인 동작을 예시하는 플로우차트이다.

도 9a는 본 개시물의 하나 이상의 기법들에 따른, 비디오 디코더의 예시적인 동작을 예시하는 플로우차트이다.

도 9b는 본 개시물의 하나 이상의 기법들에 따른, 비디오 디코더의 예시적인 동작을 예시하는 플로우차트이다.

도 10a는 본 개시물의 하나 이상의 기법들에 따른, 부호 데이터 은닉 (sign data hiding)를 위한 예시적인 비디오 인코더 동작을 예시하는 플로우차트이다.

도 10b는 본 개시물의 하나 이상의 기법들에 따른, 부호 데이터 은닉을 위한 예시적인 비디오 디코더 동작을 예시하는 플로우차트이다.

### 발명을 실시하기 위한 구체적인 내용

[0017]

일반적으로, 본 개시물은 HEVC (High Efficiency Video Coding) 및 다른 비디오 코딩 표준들에서 인트라 예측을 향상시키는 기법들을 기술한다. 인트라 예측은 현재의 화상에서의 샘플 값들에 기초하여, 현재의 화상의

비디오 블록에 대한 예측 블록을 생성하는 프로세스이다. 따라서, 현재의 화상의 비디오 블록이 인트라 예측을 사용하여 인코딩될 때, 비디오 인코더는 비디오 블록에 대한 예측 블록을 생성하거나 또는 아니면 식별하기 위해 다른 화상들로부터의 샘플 값들을 사용하지 않는다.

[0018] 예측 블록을 생성한 후, 비디오 인코더는 예측 블록을 사용하여, 잔여 샘플들의 블록 (즉, 잔여 블록) 을 결정할 수도 있다. 잔여 블록에서의 잔여 샘플들은 예측 블록에서의 샘플들과, 비디오 블록의 대응하는 원래 샘플들 사이의 차이를 나타낼 수도 있다. 비디오 인코더는 잔여 블록에 변환을 적용함으로써 변환 계수 블록을 생성할 수도 있다. 이 변환은 잔여 샘플들을 광센 도메인으로부터 변환 도메인으로 변환할 수도 있다.

비디오 인코더는 그 후 변환 계수들의 비트 심도들을 감소시키기 위해 변환 계수 블록에서의 변환 계수들을 양자화할 수도 있다. 비디오 인코더는 양자화된 변환 계수들을 나타내는 신택스 엘리먼트들을 엔트로피 인코딩하고, 최종 엔트로피 인코딩된 신택스 엘리먼트들을 비트스트림에 포함시킬 수도 있다.

[0019] 비디오 디코더는 이 프로세스의 역방향을 수행할 수도 있다. 즉, 비디오 디코더는 비트스트림에서의 신택스 엘리먼트들을 엔트로피 디코딩하여, 양자화된 변환 계수들을 결정할 수도 있다. 비디오 디코더는 그 후 양자화된 변환 계수들을 역양자화하여, 변환 계수들을 결정할 수도 있다. 더욱이, 비디오 디코더는 변환 계수들에 역변환을 적용하여 잔여 블록을 결정할 수도 있다. 게다가, 비디오 디코더는 예측 블록을 (예컨대, 인트라 예측을 사용하여) 결정할 수도 있다. 비디오 디코더는 예측 블록에서의 샘플들 및 잔여 블록에서의 대응하는 잔여 샘플들을 사용하여, 비디오 블록의 샘플들을 재구성할 수도 있다.

[0020] 변환의 적용 및 양자화의 사용은 정보 손실을 초래한다. 따라서, 비디오 디코더에 의해 재구성된 비디오 블록의 샘플들은 비디오 블록의 원래 샘플들과 동일한 정밀도의 레벨을 갖지 않을 수도 있다. 따라서, 변환의 적용과 양자화의 사용은 "손실" 코딩의 유형일 수도 있다. 일부의 경우, 비디오 인코더는 비디오 블록을 무손실 인코딩을 사용하여 인코딩할 수도 있다. 비디오 인코더가 무손실 인코딩을 사용하여 비디오 블록을 인코딩할 때, 비디오 인코더는 잔여 샘플들에 변환을 적용하지 않고, 잔여 샘플들을 양자화하지 않는다. 이와 유사하게, 비디오 디코더는 역양자화 또는 역변환을 적용하지 않는다. 그 결과, 비디오 디코더에 의해 재구성된 비디오 블록의 샘플들은 비디오 블록의 원래 샘플들과 동일한 정밀도의 레벨을 가질 수도 있다.

[0021] 다른 경우, 비디오 인코더는 비디오 인코더가 잔여 샘플들에 변환을 적용하지 않지만 잔여 샘플들을 양자화하는 일종의 손실 코딩의 유형을 수행할 수도 있다. 이와 유사하게, 비디오 디코더는 잔여 샘플들에 역양자화를 적용하지만, 잔여 샘플들에 역변환을 적용하지 않을 수도 있다. 비디오 인코더가 잔여 샘플들에 여전히 양자화를 적용하기 때문에, 비디오 디코더에 의해 재구성되는 샘플들은 원래 샘플들보다 더 작은 정밀도를 가질 수도 있고, 그러나 정밀도 손실은 변환이 적용된 경우보다 잠재적으로 더 적을 수도 있다.

[0022] 위에서 나타낸 바와 같이, 비디오 코더 (예컨대, 비디오 인코더 또는 비디오 디코더) 는 인트라 예측을 사용하여, 예측 블록을 생성할 수도 있다. 좀더 구체적으로, 비디오 코더는 복수의 사용 인트라 예측 모드들 중에서 특정의 인트라 예측 모드를 사용하여, 예측 블록을 생성한다. HEVC 및 다른 비디오 코딩 표준들에서, 인트라 예측 모드들은 복수의 방향 인트라 예측 모드들, 평면 인트라 예측 모드, 및 DC 인트라 예측 모드를 포함한다. 일반적으로, 비디오 코더가 평면 인트라 예측 모드를 사용하여 예측 블록을 생성할 때, 예측 블록의 샘플들은 선형 투영들 (linear projections) 의 조합에 기초하여 결정될 수도 있다. 비디오 코더가 DC 인트라 예측 모드를 사용하여 예측 블록을 생성할 때, 비디오 코더는 DC 인트라 예측 값을 결정할 수도 있다. DC 인트라 예측 값은 예측 블록의 좌측 에지 및 상부 에지에 인접한 샘플들의 평균 값일 수도 있다. 비디오 코더는 예측 블록에서의 각각의 샘플 값을 DC 인트라 예측 값과 동일하게 설정할 수도 있다.

[0023] 본 개시물의 일부 기법들은 비디오 코더가 무손실 코딩을 사용할 때 DC 인트라 예측 모드에 대한 향상을 제공한다. 무손실 코딩에서, 비디오 인코더는 DC 인트라 예측 모드를 사용할 때 샘플들의 원래 값들을 사용하여, 예측 블록에서의 샘플들의 값들을 결정할 수도 있다. 손실 코딩에서, 비디오 디코더는 DC 인트라 예측을 사용할 때 샘플들의 원래 값들에 액세스하지 않고 예측 블록에서의 샘플들의 값들을 결정한다. 그러나, 무손실 코딩에서, 비디오 디코더는 DC 인트라 예측을 사용할 때 샘플들의 재구성된 값들에 액세스하여, 예측 블록에서의 값들을 결정한다. 무손실 코딩에서, 샘플들의 재구성된 값들은 샘플들의 원래 값들과 동일하다.

[0024] 본원에서 설명하는 바와 같이, 비디오 코더는 예측 블록을 생성할 수도 있다. 예측 블록을 생성하는 것의 일부로서, 비디오 코더는 현재의 샘플의 DC 예측을 위해, 예측 블록의 현재의 로우에서 현재의 샘플의 좌측에 있는 무손실로 재구성된 샘플 및 현재의 로우의 상부에 있는 예측 블록의 로우에 대한 무손실로 재구성된 샘플 중 적어도 하나를 사용할 수도 있다. 더욱이, 일부 경우, 이것은 비디오 디코더로 하여금, 예측 블록에서의 샘플 값들의 결정을 파이프라인 (pipeline) 가능하게 할 수도 있다.

[0025] 더욱이, 위에서 나타낸 바와 같이, 비디오 인코더는 변환 스킵 코딩으로서 지칭될 수도 있는, 양자화가 사용되지만 변환은 스kip되는 손실 코딩의 유형을 수행할 수도 있다. 본 개시물의 하나 이상의 추가적인 기법들에 따르면, 비디오 인코더는 잔여 차분 펠스 코드 변조 (DPCM) 의 유형을 적용하여, 코딩을 위해 비-변환되지만 양자화된, 잔여 샘플들을 준비할 수도 있다. 이 유형의 잔여 DPCM 은 본 개시물에서 다른 어딘가에 자세하게 설명된다. 손실 인트라 코딩에서 DPCM 을 사용하는 다른 제안들과는 대조적으로, 본 개시물에서 설명되는 잔여 DPCM 의 이 유형은 비디오 인코더 및/또는 비디오 디코더의 처리량을 증가시킬 수도 있다.

[0026] 위에서 나타낸 바와 같이, 비디오 인코더는 양자화된 변환 계수들을 나타내는 십택스 엘리먼트들을 엔트로피 인코딩할 수도 있다. 무손실 코딩 또는 손실 코딩에서, 변환이 스kip될 때, 동일한 십택스 엘리먼트들이 잔여 샘플들을 나타내기 위해 사용될 수도 있다. HEVC 및 다른 비디오 코딩 표준들에서, 변환 계수 또는 잔여 샘플을 나타내는 십택스 엘리먼트들은 변환 계수 또는 잔여 샘플이 양인지 또는 음인지 여부를 나타내는 부호 십택스 엘리먼트를 포함할 수도 있다. 일부의 경우, 변환 계수 또는 잔여 샘플이 양인지 또는 음인지 여부를 나타내기 위해 부호 십택스 엘리먼트들을 포함하는 것이 불필요할 수도 있다. 대신, 변환 계수 또는 잔여 샘플이 양인지 또는 음인지 여부를 나타내는 정보가 변환 계수 또는 잔여 샘플에 대한 다른 십택스 엘리먼트들의 값들에 내장될 수도 있다. 부호 십택스 엘리먼트들을 시그널링하는 대신, 이러한 정보를 다른 십택스 엘리먼트들의 값들에 내장하는 것은, 부호 데이터 은닉으로서 지칭될 수도 있다.

[0027] 그러나, 부호 데이터 은닉은, 변환이 스kip되고 평면 인트라 예측 모드, DC 인트라 예측 모드 (예컨대, 예측 블록에서의 샘플들에 대응하는 재구성된 샘플들이 예측 블록에서의 예측 샘플들의 값을 결정하는데 사용되는 DC 인트라 예측 모드), 또는 잔여 DPCM 이 사용되는 손실 코딩을 사용하여 코딩되는 블록들에 대해 구현하는 것이 어려울 수도 있다. 더욱이, 변환 스킵 코딩에서, 부호 데이터 은닉은 잔여 DPCM 이 적용될 때 형성되는 잔여 값들에 에러들을 도입할 수도 있다. 이러한 에러들은 후속 잔여 샘플들로 전파하여, 성능의 열화를 초래 할 수도 있다. 따라서, 본 개시물의 하나 이상의 기법들에 따르면, 부호 데이터 은닉은, 부호 데이터 은닉이 이러한 블록들에 대해 인에이블된다면 하나 이상의 십택스 엘리먼트들이 나타내더라도, 이러한 블록들에 대해 규범적으로 디스에이블될 수도 있다.

[0028] 예를 들어, 일부 예들에서, 비디오 디코더는, 현재 블록이 잔여 데이터에 대한 변환의 적용 없이 손실 코딩을 사용하여 생성되고 잔여 DPCM 이 사용되는 인트라 예측 모드를 사용하여 현재 블록이 인트라 예측되면, 부호 데이터 은닉이 현재 블록에 대해 디스에이블된다면 결정한다. 이러한 예들에서, 부호 데이터 은닉이 현재 블록에 대해 디스에이블될 때, 비디오 디코더는 비트스트림으로부터, 블록에서의 각각의 개별의 유효 값 (significant value) 에 대해, 각각의 유효 값이 양인지 또는 음인지 여부를 나타내는 각각의 십택스 엘리먼트를 획득할 수도 있다.

[0029] 도 1 은 본 개시물의 기법들을 사용할 수도 있는 예시적인 비디오 코딩 시스템 (10) 을 예시하는 블록도이다. 본원에서 설명하는 바와 같이, 용어 "비디오 코더" 는 비디오 인코더들 및 비디오 디코더들 양쪽을 포괄적으로 지칭한다. 본 개시물에서, 용어들 "비디오 코딩" 또는 "코딩" 은 비디오 인코딩 또는 비디오 디코딩을 포괄적으로 지칭할 수도 있다.

[0030] 도 1 에 나타낸 바와 같이, 비디오 코딩 시스템 (10) 은 소스 디바이스 (12) 및 목적지 디바이스 (14) 를 포함 한다. 소스 디바이스 (12) 는 인코딩된 비디오 데이터를 생성한다. 따라서, 소스 디바이스 (12) 는 비디오 인코딩 디바이스 또는 비디오 인코딩 장치로서 지칭될 수도 있다. 목적지 디바이스 (14) 는 소스 디바이스 (12) 에 의해 생성된 인코딩된 비디오 데이터를 디코딩할 수도 있다. 따라서, 목적지 디바이스 (14) 는 비디오 디코딩 디바이스 또는 비디오 디코딩 장치로서 지칭될 수도 있다. 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 비디오 코딩 디바이스들 또는 비디오 코딩 장치들의 예들일 수도 있다.

[0031] 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 데스크탑 컴퓨터들, 모바일 컴퓨팅 디바이스들, 노트북 (예컨대, 랩탑) 컴퓨터들, 태블릿 컴퓨터들, 셋-탑 박스들, 소위 "스마트" 폰들과 같은 전화기 핸드셋들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 콘솔들, 자동차용 컴퓨터들, 또는 기타 등등을 포함한, 광범위한 디바이스들을 포함할 수도 있다.

[0032] 목적지 디바이스 (14) 는 인코딩된 비디오 데이터를 소스 디바이스 (12) 로부터 채널 (16) 을 통해 수신할 수도 있다. 채널 (16) 은 인코딩된 비디오 데이터를 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 이동시키는 것이 가능한 하나 이상의 매체들 또는 디바이스들을 포함할 수도 있다. 일 예에서, 채널 (16) 은 소스 디바이스 (12) 로 하여금, 인코딩된 비디오 데이터를 직접 목적지 디바이스 (14) 로 실시간으로 송신 가능하게 하는 하나 이상의 통신 매체들을 포함할 수도 있다. 이 예에서, 소스 디바이스 (12) 는 무선 통신 프로토콜

과 같은 통신 표준에 따라서, 인코딩된 비디오 데이터를 변조할 수도 있고, 변조된 비디오 데이터를 목적지 디바이스 (14)로 송신할 수도 있다. 하나 이상의 통신 매체들은 무선 및/또는 유선 통신 매체들, 예컨대 무선 주파수 (RF) 스펙트럼 또는 하나 이상의 물리적인 송신 라인들을 포함할 수도 있다. 하나 이상의 통신 매체들은 근거리 네트워크, 광역 네트워크, 또는 글로벌 네트워크 (예컨대, 인터넷)과 같은, 패킷-기반 네트워크의 일부를 형성할 수도 있다. 채널 (16)은 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (12)로부터 목적지 디바이스 (14)로 통신을 용이하게 하는 다른 장비와 같은, 여러 유형들의 디바이스들을 포함할 수도 있다.

[0033] 또 다른 예에서, 채널 (16)은 소스 디바이스 (12)에 의해 생성되는 인코딩된 비디오 데이터를 저장하는 저장 매체를 포함할 수도 있다. 이 예에서, 목적지 디바이스 (14)는, 예컨대, 디스크 액세스 또는 카드 액세스를 통해 저장 매체에 액세스할 수도 있다. 저장 매체는 Blu-ray 디스크들, DVDs, CD-ROMs, 플레이시 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 다른 적합한 디지털 저장 매체들과 같은 다양한 로컬-액세스되는 데이터 저장 매체들을 포함할 수도 있다.

[0034] 추가 예에서, 채널 (16)은 소스 디바이스 (12)에 의해 생성된 인코딩된 비디오 데이터를 저장하는 파일 서버 또는 또 다른 중간 저장 디바이스를 포함할 수도 있다. 이 예에서, 목적지 디바이스 (14)는 스트리밍 또는 다운로드를 통해 파일 서버 또는 다른 중간 저장 디바이스에 저장된 인코딩된 비디오 데이터에 액세스할 수도 있다. 파일 서버는 인코딩된 비디오 데이터를 저장하고 그 인코딩된 비디오 데이터를 목적지 디바이스 (14)로 송신가능한 서버의 형태일 수도 있다. 예시적인 파일 서버들은 (예컨대, 웹사이트용) 웹 서버들, 파일 전송 프로토콜 (FTP) 서버들, NAS (network attached storage) 디바이스들, 로컬 디스크 드라이브들 등을 포함한다.

[0035] 목적지 디바이스 (14)는 인터넷 접속과 같은 표준 데이터 접속을 통해 인코딩된 비디오 데이터에 액세스할 수도 있다. 데이터 접속들의 예시적인 유형들은 무선 채널들 (예컨대, Wi-Fi 접속들), 유선 접속들 (예컨대, DSL, 케이블 모뎀, 등), 또는 파일 서버 상에 저장된 인코딩된 비디오 데이터에 액세스하는데 적합한 양쪽의 조합들을 포함할 수도 있다. 파일 서버로부터의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 이 양쪽의 조합일 수도 있다.

[0036] 본 개시물의 기법들은 무선 애플리케이션들 또는 설정들에 한정되지 않는다. 이 기법들은 오버-디-에어 텔레비전 브로드캐스트들, 케이블 텔레비전 송신들, 예컨대, 인터넷을 통한 위성 텔레비전 송신들, 데이터 저장 매체 상의 저장을 위한 비디오 데이터의 인코딩, 데이터 저장 매체 상에 저장된 비디오 데이터의 디코딩, 또는 다른 애플리케이션들과 같은, 다양한 멀티미디어 애플리케이션들의 지원 하에서, 비디오 코딩에 적용될 수도 있다. 일부 예들에서, 비디오 코딩 시스템 (10)은 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅, 및/또는 비디오 전화 통신과 같은, 지원 애플리케이션들로의 일방향 또는 양방향 비디오 송신을 지원하도록 구성될 수도 있다.

[0037] 도 1의 예에서, 소스 디바이스 (12)는 비디오 소스 (18), 비디오 인코더 (20), 및 출력 인터페이스 (22)를 포함한다. 일부 예들에서, 출력 인터페이스 (22)는 변조기/복조기 (모뎀) 및/또는 송신기를 포함할 수도 있다. 비디오 소스 (18)은 비디오 캡쳐 디바이스, 예컨대, 비디오 카메라, 이전에-캡쳐된 비디오 데이터를 포함하는 비디오 아카이브, 비디오 컨텐츠 제공자로부터 비디오 데이터를 수신하는 비디오 공급 인터페이스, 및/또는 비디오 데이터를 생성하기 위한 컴퓨터 그래픽스 시스템, 또는 이런 비디오 데이터의 소스들의 조합을 포함할 수도 있다.

[0038] 비디오 인코더 (20)는 비디오 소스 (18)로부터의 비디오 데이터를 인코딩할 수도 있다. 일부 예들에서, 소스 디바이스 (12)는 인코딩된 비디오 데이터를 목적지 디바이스 (14)로부터 출력 인터페이스 (22)를 통해 직접 송신한다. 다른 예들에서, 인코딩된 비디오 데이터는 또한 디코딩 및/또는 플레이백을 위한 목적지 디바이스 (14)에 의한 추후 액세스를 위해 저장 매체 또는 파일 서버 상으로 저장될 수도 있다.

[0039] 도 1의 예에서, 목적지 디바이스 (14)는 입력 인터페이스 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (32)를 포함한다. 일부 예들에서, 입력 인터페이스 (28)은 수신기 및/또는 모뎀을 포함한다. 입력 인터페이스 (28)은 인코딩된 비디오 데이터를 채널 (16)을 통해 수신할 수도 있다. 디스플레이 디바이스 (32)는 목적지 디바이스 (14)와 통합되거나 또는 그 외부에 있을 수도 있다. 일반적으로, 디스플레이 디바이스 (32)는 디코딩된 비디오 데이터를 디스플레이한다. 디스플레이 디바이스 (32)는 다양한 디스플레이 디바이스들, 예컨대 액정 디스플레이 (LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (OLED) 디스플레이, 또는 또 다른 유형의 디스플레이 디바이스를 포함할 수도 있다.

[0040] 도 1 은 단지 예이고, 본 개시물의 기법들은 비디오 인코딩 디바이스와 비디오 디코딩 디바이스 사이의 임의의 데이터 통신을 반드시 포함할 필요가 없는 비디오 코딩 설정들 (예컨대, 비디오 인코딩 또는 비디오 디코딩)에 적용할 수도 있다. 다른 예들에서, 데이터는 로컬 메모리로부터 취출되어, 네트워크 등을 통해 스트리밍된다. 비디오 인코딩 디바이스는 데이터를 인코딩하여 메모리에 저장할 수도 있거나, 및/또는 비디오 디코딩 디바이스는 메모리로부터 데이터를 취출하여 디코딩할 수도 있다. 많은 예들에서, 비디오 인코딩 및 디코딩은 서로 통신하지 않지만, 메모리에 데이터를 간단히 인코딩하거나 및/또는 메모리로부터 데이터를 취출하여 디코딩하는 디바이스들에 의해 수행된다.

[0041] 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들 (DSPs), 주문형 집적 회로들 (ASICs), 필드-프로그래밍가능 게이트 어레이들 (FPGAs), 이산 로직, 하드웨어, 또는 임의의 이들의 조합들과 같은, 다양한 적합한 회로 중 임의의 회로로서 구현될 수도 있다. 기법들이 소프트웨어로 부분적으로 구현되면, 디바이스는 소프트웨어용 명령들을 적합한 비일시적 컴퓨터 판독가능 저장 매체에 저장할 수도 있고, 본 개시물의 기법들을 수행하기 위해 그 명령들을 하드웨어에서 하나 이상의 프로세서들을 사용하여 실행할 수도 있다. (하드웨어, 소프트웨어, 하드웨어와 소프트웨어의 조합 등을 포함한) 전술한 것 중 임의의 것이 하나 이상의 프로세서들로 간주될 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 인코더들 또는 디코더들에 포함될 수도 있고, 이를 중 어느 쪽이든 각각의 디바이스에서 결합된 인코더/디코더 (CODEC) 의 일부로서 통합될 수도 있다.

[0042] 본 개시물은 일반적으로 어떤 정보를 "시그널링하는" 비디오 인코더 (20) 을 참조할 수도 있다. 용어 "시그널링"은 일반적으로 압축된 비디오 데이터를 디코딩하는데 사용되는 선택스 엘리먼트들 및/또는 다른 데이터의 통신을 지칭할 수도 있다. 이런 통신은 실시간 또는 거의-실시간으로 일어날 수도 있다. 대안적으로, 이러한 통신은 예컨대, 선택스 엘리먼트들을, 인코딩 시에 인코딩된 비트스트림으로, 컴퓨터 판독가능 저장 매체에 저장할 때 일어날지도 모르는 어떤 기간에 걸쳐서 일어날 수도 있고, 이 선택스 엘리먼트들은 그 후 이 매체에 저장된 후에 비디오 디코딩 디바이스가 임의의 시간에 취출할 수도 있다.

[0043] 일부 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 HEVC (High Efficiency Video Coding) 표준과 같은, 비디오 압축 표준에 따라서 동작한다. "HEVC 작업 초안 6" 으로 지칭되는, HEVC 표준의 초안은, ITU-T SG16 WP3 과 ISO/IEC JTC1/SC29/WG11 의 JCT-VC (Joint Collaborative Team on Video Coding), 7차 회의: 2011년 11월, 스위스, 제네바, Bross 등, "High efficiency video coding (HEVC) text specification draft 6" 에 설명되어 있고, 이의 전체 내용이 본원에 참고로 포함된다. 2013년 4월 5일 현재, HEVC 작업 초안 6 은 [http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/8\\_San%20Jose/wg11/JCTVC-H1003-v22.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/8_San%20Jose/wg11/JCTVC-H1003-v22.zip)로부터 다운로드 가능하다. "HEVC 작업 초안 9" 로서 지칭되는, HEVC 표준의 또 다른 초안은, ITU-T SG16 WP3 과 ISO/IEC JTC1/SC29/WG11 의 JCT-VC (Joint Collaborative Team on Video Coding), 11차 회의: 2012년, 10월, 중국, 상하이, Bross 등, "HEVC Working Draft 9" 에 설명되어 있고, 이의 전체 내용이 본원에 참고로 포함된다. 2014년 3월 24일 현재, HEVC 작업 초안 9 는 [http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/11\\_Shanghai/wg11/JCTVC-K1003-v8.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/11_Shanghai/wg11/JCTVC-K1003-v8.zip)로부터 다운로드 가능하다. "HEVC 작업 초안 10" 으로서 지칭되는, HEVC 의 또 다른 초안은, ITU-T SG16 WP3 과 ISO/IEC JTC1/SC29/WG11 의 JCT-VC (Joint Collaborative Team on Video Coding), 12차 회의: 2013년 1월 14-23일, 스위스, 제네바, Bross 등, "High Efficiency Video Coding (HEVC) text specification draft 10 (for FDIS & Consent)" 에 설명되어 있고, 이의 전체 내용이 본원에 참고로 포함된다. 2014년 3월 24일 현재, HEVC 작업 초안 10 은 [http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/12\\_Geneva/wg11/JCTVC-L1003-v20.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v20.zip)로부터 입수가능하다. 본 개시물의 기법들은, 그러나, 임의의 특정의 코딩 표준 또는 기법에 한정되지 않는다.

[0044] 더욱이, 범위 확장 사양 (range extension specification) 은 HEVC 에 대해 개발 중에 있다. 범위 확장 사양은 4: 0: 0, 4: 2: 0, 4: 2: 2, 및 4: 4: 4 크로마 샘플링과 같은, 대체 샘플링 모드들에 대해 제공한다. ITU-T SG 16 WP 3 과 ISO/IEC JTC 1/SC 29/WG 11 의 JCT-VC (Joint Collaborative Team on Video Coding), 13차 회의: 2013년 4월 18-26일, 대한민국, 인천, Flynn 등, "High Efficiency Video Coding (HEVC) Range Extensions text specification: Draft 3", (이하, "JCTVC-M1005\_v2") 는 HEVC 에 대한 범위 확장 사양의 초안이다. 2014년 3월 24일 현재, JCTVC-M1005\_v2 는 [http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/13\\_Incheon/pending/JCTVC-M1005-v2.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/13_Incheon/pending/JCTVC-M1005-v2.zip)로부터 입수가능하다. JCTVC-M1005\_v2 의 전체 내용이 참조로 포함된다.

[0045] 위에서 간단히 언급한 바와 같이, 비디오 인코더 (20) 는 비디오 데이터를 인코딩한다. 비디오 데이터는 하나 이상의 화상들을 포함할 수도 있다. 화상들의 각각은 비디오의 일부를 형성하는 정지 화상이다. 비

디오 인코더 (20) 가 비디오 데이터를 인코딩할 때, 비디오 인코더 (20) 는 비트스트림을 생성할 수도 있다. 비트스트림은 비디오 데이터의 코딩된 표현을 형성하는 비트들의 시퀀스를 포함할 수도 있다. 비트스트림은 코딩된 화상들 및 연관된 데이터를 포함할 수도 있다. 코딩된 화상은 화상의 코딩된 표현이다. 연관되는 데이터는 시퀀스 파라미터 세트들 (SPSs), 화상 파라미터 세트들 (PPSs), 및 다른 신택스 구조들을 포함할 수도 있다. SPS 는 제로 또는 더 이상의 화상들의 시퀀스들에 적용가능한 파라미터들을 포함할 수도 있다. PPS 는 제로 또는 더 이상의 화상들에 적용가능한 파라미터들을 포함할 수도 있다.

[0046] 화상은  $S_L$ ,  $S_{Cb}$  및  $S_{Cr}$  로 나타내는, 3개의 샘플 어레이들을 포함할 수도 있다.  $S_L$  은 루마 샘플들의 2차원 어레이 (즉, 블록) 이다. 루마 샘플들은 또한 본원에서 "Y" 샘플들로서 지칭될 수도 있다.  $S_{Cb}$  는  $Cb$  색차 샘플들의 2차원 어레이이다.  $S_{Cr}$  은  $Cr$  색차 샘플들의 2차원 어레이이다. 색차 샘플들은 또한 본원에서 "크로마" 샘플들로서 지칭될 수도 있다.  $Cb$  색차 샘플들은 본원에서 "U 샘플들"로서 지칭될 수도 있다.  $Cr$  색차 샘플들은 본원에서 "V 샘플들"로서 지칭될 수도 있다.

[0047] 일부 예들에서, 비디오 인코더 (20) 는 화상의 크로마 어레이들 (즉,  $S_{Cb}$  및  $S_{Cr}$ ) 을 다운샘플링할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 YUV 4: 2: 0 비디오 포맷, YUV 4: 2: 2 비디오 포맷, 또는 4: 4: 4 비디오 포맷을 사용할 수도 있다. YUV 4: 2: 0 비디오 포맷에서, 비디오 인코더 (20) 는 크로마 어레이들이 루마 어레이의 높이의  $\frac{1}{2}$  및 폭의  $\frac{1}{2}$  이 되도록, 크로마 어레이들을 다운샘플링할 수도 있다. YUV 4: 2: 2 비디오 포맷에서, 비디오 인코더 (20) 는 크로마 어레이들이 루마 어레이의 폭의  $\frac{1}{2}$  및 동일한 높이가 되도록, 크로마 어레이들을 다운샘플링할 수도 있다. YUV 4: 4: 4 비디오 포맷에서, 비디오 인코더 (20) 는 크로마 어레이들을 다운샘플링하지 않는다.

[0048] 화상의 인코딩된 표현을 생성하기 위해, 비디오 인코더 (20) 는 코딩 트리 유닛들 (CTUs) 의 세트를 생성할 수도 있다. CTUs 의 각각은 루마 샘플들의 코딩 트리 블록, 크로마 샘플들의 2개의 대응하는 코딩 트리 블록들, 및 코딩 트리 블록들의 샘플들을 코딩하는데 사용되는 신택스 구조들일 수도 있다. 코딩 트리 블록은 샘플들의 NxN 블록일 수도 있다. CTU 는 또한 "트리 블록" 또는 "최대 코딩 유닛" (LCU) 로서 지칭될 수도 있다. HEVC 의 CTUs 는 H.264/AVC 와 같은, 다른 표준들의 매크로블록들과 대략 유사할 수도 있다. 그러나, CTU 는 특정의 사이즈에 반드시 제한되지 않고, 하나 이상의 코딩 유닛들 (CUs) 를 포함할 수도 있다.

[0049] 화상을 인코딩하는 것의 일부로서, 비디오 인코더 (20) 는 화상의 각각의 슬라이스 (즉, 코딩된 슬라이스들) 의 인코딩된 표현들을 생성할 수도 있다. 코딩된 슬라이스를 생성하기 위해, 비디오 인코더 (20) 는 CTUs 의 시리즈를 인코딩할 수도 있다. 본 개시물은 CTU 의 인코딩된 표현을 코딩된 CTU 로서 지칭할 수도 있다.

일부 예들에서, 슬라이스들의 각각은 정수의 코딩된 CTUs 를 포함한다.

[0050] 코딩된 CTU 를 생성하기 위해, 비디오 인코더 (20) 는 코딩 트리 블록들을 코딩 블록들, 따라서 이름 "코딩 트리 유닛들"로 분할하기 위해 CTU 의 코딩 트리 블록들에 관해 큐드-트리 파티셔닝을 회귀적으로 수행할 수도 있다. 코딩 블록은 샘플들의 NxN 블록이다. CU 는 루마 샘플 어레이,  $Cb$  샘플 어레이 및  $Cr$  샘플 어레이를 갖는 화상의 루마 샘플들의 코딩 블록 및 크로마 샘플들의 2개의 대응하는 코딩 블록들, 및 코딩 블록들의 샘플들을 코딩하는데 사용되는 신택스 구조들일 수도 있다. 단색 화상들, 또는 별개의 칼라 평면들을 사용하여 코딩되는 화상들에서, CU 는 샘플들의 단일 코딩 블록 및 코딩 블록을 코딩하는데 사용되는 신택스 구조들을 포함할 수도 있다. 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) 블록일 수도 있다. CU 의 예측 유닛 (PU) 는 루마 샘플들의 예측 블록, 화상의 크로마 샘플들의 2개의 대응하는 예측 블록들, 및 예측 블록 샘플들을 예측하는데 사용되는 신택스 구조들일 수도 있다. 비디오 인코더 (20) 는 CU 의 각각의 PU 의 루마,  $Cb$  및  $Cr$  예측 블록들에 대한 예측 루마,  $Cb$  및  $Cr$  블록들을 생성할 수도 있다. 단색 화상들, 또는 별개의 칼라 평면들을 사용하여 코딩되는 화상들에서, CU 는 샘플들의 단일 코딩 블록 및 코딩 블록을 코딩하는데 사용되는 신택스 구조들을 포함할 수도 있다.

[0051] 비디오 인코더 (20) 는 인트라 예측 또는 인터 예측을 사용하여, PU 에 대한 예측 블록들을 생성할 수도 있다. 비디오 인코더 (20) 가 PU 의 예측 블록들을 생성하기 위해 인트라 예측을 사용하면, 비디오 인코더 (20) 는 PU 와 연관되는 화상의 디코딩된 샘플들에 기초하여 PU 의 예측 블록들을 생성할 수도 있다.

[0052] 비디오 인코더 (20) 가 PU 의 예측 블록들을 생성하기 위해 인터 예측을 사용하면, 비디오 인코더 (20) 는 PU 와 연관되는 화상 이외의 하나 이상의 화상들의 디코딩된 샘플들에 기초하여, PU 의 예측 블록들을 생성할 수도 있다. 비디오 인코더 (20) 는 PU 의 예측 블록들을 생성하기 위해 단방향-예측 또는 양방향-예측을 사용할

수도 있다. 비디오 인코더 (20) 가 PU 에 대한 예측 블록들을 생성하기 위해 단방향-예측을 사용할 때, PU 는 단일 모션 벡터를 가질 수도 있다. 비디오 인코더 (20) 가 PU 에 대한 예측 블록들을 생성하기 위해 단방향-예측을 사용할 때, PU 는 2개의 모션 벡터들을 가질 수도 있다.

[0053] 비디오 인코더 (20) 가 CU 의 하나 이상의 PUs 에 대한 예측 블록들 (예컨대, 루마, Cb 및 Cr 블록들) 을 생성한 후, 비디오 인코더 (20) 는 CU 에 대한 잔여 블록을 생성할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 CU 에 대한 루마 잔여 블록을 생성할 수도 있다. CU 의 루마 잔여 블록에서의 각각의 샘플은 CU 의 예측 루마 블록들 중 하나에서의 루마 샘플과 CU 의 원래 루마 코딩 블록에서의 대응하는 샘플 사이의 차이를 나타낸다. 게다가, 비디오 인코더 (20) 는 CU 에 대한 Cb 잔여 블록을 생성할 수도 있다. CU 의 Cb 잔여 블록에서의 각각의 샘플은 CU 의 예측 Cb 블록들 중 하나에서의 Cb 샘플과 CU 의 원래 Cb 코딩 블록에서의 대응하는 샘플 사이의 차이를 나타낼 수도 있다. 비디오 인코더 (20) 는 또한 CU 에 대한 Cr 잔여 블록을 생성할 수도 있다. CU 의 Cr 잔여 블록에서의 각각의 샘플은 CU 의 예측 Cr 블록들 중 하나에서의 Cr 샘플과 CU 의 원래 Cr 코딩 블록에서의 대응하는 샘플 사이의 차이를 나타낼 수도 있다.

[0054] 더욱이, 비디오 인코더 (20) 는 퀼드-트리 파티셔닝을 사용하여, CU 의 잔여 블록들을 변환 블록들로 분해할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 퀼드-트리 파티셔닝을 사용하여, CU 의 루마, Cb, 및 Cr 잔여 블록들을 루마, Cb, 및 Cr 변환 블록들로 분해할 수도 있다. 변환 블록은 동일한 변환이 적용되는 샘플들의 직사각형의 블록일 수도 있다. CU 의 변환 유닛 (TU) 는 루마 샘플들의 변환 블록, 크로마 샘플들의 2개의 대응하는 변환 블록들, 및 변환 블록 샘플들을 변환하는데 사용되는 선택스 구조들일 수도 있다. 따라서, CU 의 각각의 TU 는 루마 변환 블록, Cb 변환 블록, 및 Cr 변환 블록과 연관될 수도 있다. TU 와 연관되는 루마 변환 블록은 CU 의 루마 잔여 블록의 서브-블록일 수도 있다. Cb 변환 블록은 CU 의 Cb 잔여 블록의 서브-블록일 수도 있다. Cr 변환 블록은 CU 의 Cr 잔여 블록의 서브-블록일 수도 있다. 단색 화상들, 또는 별개의 칼라 평면들을 사용하여 인코딩되는 화상들에서, TU 는 단일 변환 블록 및 변환 블록 샘플들을 변환하는데 사용되는 선택스 구조들을 포함할 수도 있다. TU 사이즈는 TU 의 변환 블록의 사이즈일 수도 있다.

[0055] 비디오 인코더 (20) 는 하나 이상의 변환들을 TU 의 변환 블록에 적용하여, TU 에 대한 계수 블록을 생성할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 하나 이상의 변환들을 TU 의 루마 변환 블록에 적용하여, TU 에 대한 루마 계수 블록을 생성할 수도 있다. 계수 블록은 변환 계수들의 2차원 어레이일 수도 있다. 변환 계수는 스칼라 양일 수도 있다. 또한, 비디오 인코더 (20) 는 하나 이상의 변환들을 TU 의 Cb 변환 블록에 적용하여, TU 에 대한 Cb 계수 블록을 생성할 수도 있다. 비디오 인코더 (20) 는 하나 이상의 변환들을 TU 의 Cr 변환 블록에 적용하여, TU 에 대한 Cr 계수 블록을 생성할 수도 있다.

[0056] 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 계수들을 나타내는데 사용되는 데이터의 양을 가능한 한 감축하기 위해 변환 계수들이 양자화되는 프로세스를 지칭하고, 추가적인 압축을 제공한다. 비디오 인코더 (20) 가 계수 블록을 양자화한 후, 비디오 인코더 (20) 는 양자화된 변환 계수들을 나타내는 선택스 엘리먼트들을 엔트로피 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 양자화된 변환 계수들을 나타내는 선택스 엘리먼트들에 관해 컨텍스트-적응 2진 산술 코딩 (CABAC) 를 수행할 수도 있다. 비디오 인코더 (20) 는 엔트로피-인코딩된 선택스 엘리먼트들을 비트스트림으로 출력할 수도 있다.

[0057] 비디오 디코더 (30) 는 비디오 인코더 (20) 에 의해 생성되는 비트스트림을 수신할 수도 있다. 게다가, 비디오 디코더 (30) 는 비트스트림을 파싱하여, 비트스트림으로부터 선택스 엘리먼트들을 디코딩할 수도 있다. 비디오 디코더 (30) 는 비트스트림으로부터 디코딩된 선택스 엘리먼트들에 적어도 부분적으로 기초하여 비디오 데이터의 화상들을 재구성할 수도 있다. 비디오 데이터를 재구성하는 프로세스는 일반적으로 비디오 인코더 (20) 에 의해 수행되는 프로세스와 반대일 수도 있다. 예를 들어, 비디오 디코더 (30) 는 현재의 CU 의 PUs 에 대한 예측 블록들을 결정하기 위해 PUs 의 모션 벡터들을 사용할 수도 있다. 게다가, 비디오 디코더 (30) 는 현재의 CU 의 TUs 과 연관되는 변환 계수 블록들을 역양자화할 수도 있다. 비디오 디코더 (30) 는 변환 계수 블록들에 관해 역변환들을 수행하여, 현재의 CU 의 TUs 와 연관되는 변환 블록들을 재구성 할 수도 있다. 비디오 디코더 (30) 는 현재의 CU 의 PUs 에 대한 예측 블록들의 샘플들을 현재의 CU 의 TUs 의 변환 블록들의 대응하는 샘플들에 가산함으로써, 현재의 CU 의 코딩 블록들을 재구성할 수도 있다. 화상의 각각의 CU 에 대해 코딩 블록들을 재구성함으로써, 비디오 디코더 (30) 는 그 화상을 재구성할 수도 있다.

[0058] 위에서 나타낸 바와 같이, 비디오 인코더 (20) 또는 비디오 디코더 (30) 과 같은, 비디오 코더는, 인트라 예측을 사용하여, 현재의 PU에 대한 예측 블록을 생성할 수도 있다. 비디오 코더가 현재의 PU에 대한 예측 블록을 생성하기 위해 인트라 예측을 사용할 때, 비디오 코더는 참조 샘플들의 세트를 사용하여 예측 블록에서의 샘플들의 값들을 결정할 수도 있다. 예를 들어, HEVC 인트라 예측에서, 상부 및 좌측면 이웃하는 블록들로부터 이미 재구성된 샘플들이 예측에 사용될 수도 있다. 이들 재구성된 샘플들은 참조 샘플들로서 지정될 수도 있다.

[0059] 도 2 는 HEVC 인트라 예측을 위한 블록의 참조 샘플들을 예시한다. 다시 말해서, 도 2 는 사이즈 M (높이) × N (폭)의 블록을 예시하는 개념도이다. 도 2 에서, M은 로우들을 나타내고, N은 칼럼들을 나타낸다. 더욱이, 도 2 에서, 블록의 샘플들은  $P_{i,j}$ 로 나타내어지고, 여기서,  $0 \leq i \leq (M - 1)$  및  $0 \leq j \leq (N - 1)$ 이다. 본 개시물에서, 용어 "샘플들"은 입력 성분 (예컨대, RGB 화상에서의 R, G, 또는 B, YCbCr 화상들에서의 Y, Cb, 또는 Cr, 등)의 원래 픽셀 값들 또는 칼라 변환을 입력 성분들에 적용한 후 성분의 샘플 값들을 지칭할 수 있다. 도 2 의 예에서, 참조 픽셀들은  $P_{-1,j}$ 로 나타내어지고, 여기서  $-1 \leq j < 2N$ 이며,  $P_{i,-1}$ 로 나타내어지고, 여기서  $-1 \leq i < 2M$ 이다.

[0060] 도 2 의 예에 나타낸 바와 같이, 참조 샘플들은 현재의 PU의 좌측에 있는 참조 샘플들의 세트 및 현재의 PU의 상부에 있는 참조 샘플의 세트를 포함할 수도 있다. 본 개시물은 현재의 PU의 상부에 있는 참조 샘플들의 세트를 상부 예측자로서 지칭할 수도 있다. 본 개시물은 현재의 PU의 좌측에 있는 참조 샘플들의 세트를 좌측 예측자로서 지칭할 수도 있다. 다시 말해서, HEVC 인트라 예측에서, 상부 및 좌측면 이웃하는 블록들로부터 이미 재구성된 샘플들이 예측에 사용된다 ("상부" 이웃하는 블록은 또한 "상부" 이웃하는 블록으로서 불려질 수도 있다). 이들 샘플들은 참조 샘플들로서 지정된다. 일부 예들에서, 참조 픽셀들이 사용불가능하면, HEVC 를 사용하는 비디오 코더는 특정의 패딩 프로세스를 사용하여, 손실된 (missing) 참조 샘플들을 생성할 수도 있다.

[0061] 비디오 코더가 인트라 예측을 사용하여 예측 블록을 생성할 때, 비디오 코더는 복수의 가용 인트라 예측 모드들 중 인트라 예측 모드에 따라서 예측 블록을 생성할 수도 있다. 인트라 예측 모드들은 복수의 방향 (즉, 각도) 인트라 예측 모드들을 포함할 수도 있다. 예를 들어, HEVC 의 일부 버전들에서, 33개의 방향 인트라 예측 모드들이 존재한다. 방향 인트라 예측 모드들의 각각은 상이한 방향에 대응한다. 도 3 은 예시적인 인트라 예측 모드 방향들을 예시하는 개념도이다. 비디오 코더가 방향 인트라 예측 모드에 따라서 예측 블록을 생성할 때, 비디오 코더는 예측 블록의 각각의 샘플 각각에 대해, 방향 인트라 예측 모드에 대응하는 방향에서의 각각의 샘플과 정렬되는 참조 샘플의 값 (또는, 참조 샘플들의 가중치 조합) 을 각각의 샘플에 할당할 수도 있다. 비디오 코더가 방향 (즉, 각도) 인트라 예측 모드를 사용하여 현재 블록에 대한 예측 블록을 생성할 때, 비디오 코더는 각도 인트라 예측을 수행하고 있다고 말해질 수도 있다.

[0062] 더욱이, HEVC 의 일부 버전들에서, 인트라 예측 모드들은 DC 인트라 예측 모드를 포함한다. 이러한 HEVC 의 버전들에서, 비디오 코더가 DC 인트라 예측을 사용하여 예측 블록을 생성할 때, 비디오 코더는 참조 샘플들의 평균 값을 결정할 수도 있다. 비디오 코더는 그 후 예측 블록에서의 각각의 샘플이 결정된 평균 값을 갖는다고 결정할 수도 있다. 따라서, 비디오 코더가 DC 인트라 예측을 사용하여 예측 블록을 생성할 때, 예측 블록의 모든 샘플들은 동일한 값을 갖는다. 예를 들어, 모든 참조 샘플들이 사용가능하도록, 패딩 프로세스가 완료되었다고 가정한다. 이 예에서, 도 2 의 예에 나타낸 4×4 블록에 대해, DC 예측은 다음과 같이 형성될 수도 있다:

$$(\sum_{i=0}^3 P_{i,-1} + \sum_{j=0}^3 P_{-1,j} + 4) \gg 3, \quad (1)$$

[0064] 여기서,  $\gg$  는 비트 단위 우측 시프트 동작을 나타낸다.

[0065] HEVC 의 일부 버전들에서, 인트라 예측 모드들은 평면 인트라 예측 모드를 포함한다. 비디오 코더가 평면 인트라 예측 모드를 사용하여 PU에 대한 예측 블록을 생성할 때, 비디오 코더는 이웃하는 샘플들의 세트,  $p[x][y]$  를 결정할 수도 있고, 여기서,  $x = -1, y = -1..nTbs*2-1$  및  $x = 0..nTbS*2-1, y = -1$  이고, 여기서  $nTbS$  는 현재의 PU의 사이즈이다. 더욱이,  $predSamples[x][y]$  는 예측 블록의 위치  $x, y$ 에서의 샘플의 값을 나타낼 수도 있다. 비디오 코더는 예측 블록의 샘플들을 다음과 같이 결정할 수도 있다:

$$\begin{aligned}
 \text{predSamples}[x][y] = & ((nTbS - 1 - x) * p[-1][y] + & (2) \\
 & (x + 1) * p[nTbS][-1] + \\
 & (nTbs - 1 - y) * p[x][-1] + \\
 & (y + 1) * p[-1][nTbS] + nTbS) \gg (\log_2(nTbS) + 1)
 \end{aligned}$$

[0066]

[0067] 일반적으로, 평면 인트라 예측 모드가 사용될 때, 예측 블록의 샘플의 값은 그 값의 2개의 선형 내삽들의 평균이다. 제 1 선형 내삽에서,  $x$ 의 값들이 예측 블록의 로우를 가로질러 좌측에서 우측으로 증가함에 따라서, 로우의 좌측에 있는 참조 샘플에 부여되는 가중치는 감소하는 반면, 예측 블록의 우상부 모서리의 상부 및 우측에 있는 참조 샘플에 부여되는 가중치는 증가한다. 제 2 선형 내삽에서,  $y$ 의 값들이 예측 블록의 칼럼 아래쪽으로 증가함에 따라서, 칼럼의 상부에 있는 참조 샘플에 부여되는 가중치는 감소하는 반면, 예측 블록의 좌하단 모서리의 하측 및 좌측에 있는 샘플에 부여되는 가중치는 증가한다.

[0068]

[0068] 도 2 의 예에서, 평면 인트라 예측은 샘플들  $P_{-1,j}$  및  $P_{M,-1}$ 을 사용하여, 수직 방향에서 단방향-선형 예측을 생성할 수도 있고, 여기서  $0 \leq j \leq (N - 1)$ 이다. 이와 유사하게, 샘플들  $P_{i,-1}$  및  $P_{-1,N}$ 은 수평 방향에서 단방향-선형 예측을 생성하는데 사용될 수도 있고, 여기서  $0 \leq i \leq (M - 1)$ 이다. 마지막으로, 이 예에서, 수평 및 수직 예측들은 평균될 수도 있다 (또는, 가능하게는 또 다른 수학적 연산과 결합될 수도 있다). 예를 들어, 평면 예측된 값들이  $T_{i,j}$ 로 표기된다하고, 블록이 정사각형, 즉,  $M = N$ 이라고 가정한다. 이 예에서,  $T_{i,j}$ 는 다음과 같이 결정될 수도 있다:

$$T_{i,j}^V = (M - i) * P_{-1,j} + i * P_{M,-1},$$

$$T_{i,j}^H = (N - j) * P_{i,-1} + j * P_{-1,N} \quad \text{및} \quad (3)$$

$$T_{i,j} = (T_{i,j}^V + T_{i,j}^H + N) \gg (\log_2 N + 1).$$

[0072]

[0072] 이 예에서, \* 는 곱을 나타내고, >> 는 비트 단위 우측 시프트 동작을 나타내고, 위첨자들 H 및 V 는 수평 및 수직 방향들에서의 예측들을 각각 나타낸다.

[0073]

[0073] 일부의 경우, 비디오 인코더 (20) 및 비디오 디코더 (30)는 본원에서 설명하는 바와 같이, 무손실 코딩 모드를 구현한다. 일반적으로, 비디오 인코더 (20)가 블록을 인코딩할 때, 비디오 인코더 (20)는 (예컨대, 이산 코사인 변환을 사용하여) 변환하고 블록에 대한 잔여 데이터 (즉, 예측 에러)를 양자화한다. 다시 말해서, 예측 에러가 변환되고 양자화된다. 그러나, 비디오 인코더 (20)가 무손실 코딩을 사용하여 블록 (예컨대, CU)를 인코딩할 때, 비디오 인코더 (20)는 블록에 대한 잔여 데이터에 변환 또는 양자화를 적용하지 않을 수도 있다. 다시 말해서, (예컨대, CU 또는 전체 화상에 대한) 무손실 코딩 모드에서, 변환 및 양자화 단계들은 스킵될 수도 있다. 대신, 비디오 인코더 (20)는 양자화된 변환 계수들과 동일한 방법으로 잔여 데이터의 샘플 값들을 처리할 수도 있다. 예를 들어, 비디오 인코더 (20)는 잔여 데이터의 샘플 값들을 나타내는 신택스 엘리먼트들을 엔트로피 디코딩하고 최종 데이터를 비트스트림에 포함시킬 수도 있다. 따라서, 잔여 데이터는 변환 또는 양자화로 인해, 어떤 정보의 손실을 겪지 않는다.

[0074]

[0074] 이와 유사하게, 비디오 디코더 (30)가 무손실 인코딩을 사용하여 인코딩되어진 블록 (예컨대, CU)를 디코딩하는 일부 경우들에서, 비디오 디코더 (30)는 블록에 대한 잔여 데이터에 역양자화 또는 역변환들을 적용하지 않을 수도 있다. 대신, 비디오 디코더 (30)는 잔여 데이터의 샘플 값들을 나타내는 신택스 엘리먼트들을 엔트로피 디코딩하고 그 후 잔여 데이터의 샘플 값들에 적어도 부분적으로 기초하여 블록의 샘플 값들을 재구성 할 수도 있다.

[0075]

[0075] 본 개시물의 여러 예시적인 기법들은 무손실 인코딩에 관한 것이다. 본원에서 설명하는 바와 같이, 예측을 위해 이웃하는 블록들로부터의 참조 샘플들을 사용하는 대신, 현재 블록으로부터의 샘플들이 향상된 예측을 위해 사용될 수 있다. 예를 들어, 본 개시물의 여러 예시적인 기법들은 HEVC 표준에서 무손실 코딩을 위해 인트라 DC 예측 모드에 적용가능할 수도 있는 변경들을 기술한다. 더욱이, 본 개시물의 여러 예시적인 기법들은 HEVC 표준에서 무손실 코딩을 위해 인트라 평면 예측 모드에 적용가능할 수도 있는 변경들을 기술한다. 본 개시물의 기법들은 또한 다른 유형들의 예측에 적용가능할 수도 있고, 또한 다른 코딩 표준들에 적용가능할 수도 있다.

[0076] 무손실 코딩에서, 예측을 위해 이웃하는 블록들로부터의 참조 샘플들을 사용하는 대신, 현재 블록으로부터의 샘플들이 향상된 예측을 위해 사용될 수 있다. 예를 들어, 변환이 스kip될 때 무손실 코딩 모드들 뿐만 아니라 손실 코딩 모드들에 있어 각도 인트라 예측을 위한 기법들은, ITU-T SG 16 WP3 및 ISO/IEC JTC1/SC29/WG11의 JCT-VC (Joint Collaborative Team on Video Coding), 5차 회의, 2011년, 3월 16-23일, 스위스, 제네바, Lan 등, "Intra and inter coding tools for screen contents", 문서 JCTVC-E145 (이하, "JCTVC-E145")에 제시되어 있다. 더욱이, 무손실 코딩에 있어, 변환 및 양자화 단계들은 스kip되고, 따라서, 인트라 DC 모드를 사용할 때 예측 블록의 샘플들을 결정하는 프로세스를 더 향상시키는 것이 가능할 수도 있다.

[0077] 본 개시물의 하나 이상의 기법들에 따르면, 예측 블록의 계산을 위해, 샘플들이 로우들을 따라서 래스터 스캐닝으로 프로세싱되고 있는 것으로 가정될 수도 있다. 그러나, 다른 예들에서, 동일한 기법들이 칼럼들에 따른 래스터 스캐닝으로 (또는, 심지어, 그 기법들의 병렬화의 어떤 잠재성을 상실하지만, 대각선 또는 지그-재그 스캐닝들로) 확장될 수 있다. 어떤 변환 또는 양자화도 무손실 모드에서의 예측 에러에 적용되지 않을 때, 예측 에러 (즉, 잔여)의 엔트로피 디코딩 이후, 원래 샘플이 무손실로 재구성될 수 있다고 가정될 수도 있다. 그러므로, 무손실 코딩의 맥락에서, 용어 "원래 샘플" 또는 "원래 샘플 값"은 실제 원래 샘플 값들 또는 재구성된 샘플 값들 (즉, 비-잔여 샘플들)을 지칭할 수도 있다. 로우들을 따른 래스터 스캐닝으로 인해, 현재의 로우로부터 현재의 샘플의 좌측에 있는 모든 샘플들 뿐만 아니라 이전 로우들로부터의 모든 샘플들도 DC 예측을 위해 사용가능하다. 본 개시물의 하나 이상의 기법들은 DC 인트라 예측 모드를 사용한 인트라 예측을 향상시키기 위해 이것을 사용한다.

[0078] 본 개시물은 여러 예시적인 인트라 예측 모드들을 기술한다. 본 개시물의 인트라 예측 모드들은 평면 인트라 예측 모드를 대체할 수 있거나 또는 평면 인트라 예측 모드들로서 이해될 수 있다. 따라서, 본 개시물의 인트라 예측 모드들이 무손실 코딩을 위해 HEVC에서의 현재의 평면 모드를 대체할 수도 있다. 본 개시물의 예들에서 설명되는 세부 사항들은 본 개시물의 다른 예들의 하나 이상의 세부 사항들과 결합될 수도 있다. 즉, 세부 사항들은 또 다른 예들을 달성하기 위해 매우 다양한 상이한 방법들 중 임의의 방법으로 결합될 수도 있다.

[0079] 비디오 코더 (예컨대, 비디오 인코더 (20) 또는 비디오 디코더 (30)) 이 DC 인트라 예측 모드를 사용하여 예측 블록을 생성하는 본 개시물의 일부 예들에 따르면, 비디오 코더는 예측 블록의 샘플들을 래스터 스캐닝 순서로 프로세싱할 수도 있다. 비디오 코더가 예측 블록의 샘플을 프로세싱할 때, 비디오 코더는 예측 블록의 샘플의 인과적 이웃을 사용하여, 샘플에 대한 DC 예측 값을 형성할 수도 있다. 일반적으로, 예측 블록에서 샘플의 인과적 이웃은 이미 결정되어 있는 예측 블록에서의 샘플들에 대응하는 재구성된 샘플들 (예컨대, 비-잔여, 비-예측 샘플들)의 세트이다. 예를 들면, 예측 블록의 좌상부 샘플로부터 시작해서 로우들을 따라 래스터 스캐닝 순서를 사용할 때, 예측 블록에서 샘플의 인과적 이웃은 샘플의 상부 및 좌측 로케이션들에 대응하는 재구성된 샘플들을 포함할 수도 있다. 하나의 이런 예에서, 비디오 코더가 DC 인트라 예측 모드를 사용하여 예측 블록을 생성할 때, 비디오 코더는 현재의 샘플  $P_{i,j}$ 에 대한 DC 예측 값,  $DC_{i,j}$ 를 다음과 같이 계산할 수도 있고, 여기서  $0 \leq i \leq (M - 1)$  및  $0 \leq j \leq (N - 1)$ 이다:

$$DC_{i,j} = (P_{i,j-1} + P_{i-1,j} + 1) \gg 1. \quad (4)$$

[0080] 또는 유사하게,

$$DC_{i,j} = (P_{i,j-1} + P_{i-1,j}) \gg 1. \quad (5)$$

[0081] 따라서, 방정식들 (4) 및 (5)에 있어, 예측 블록에서 각각의 샘플 각각에 대해, 비디오 인코더 (20)는 각각의 샘플에 대한 DC 인트라 예측 값 (즉,  $DC_{i,k}$ )를, 각각의 샘플의 상부에 있는 샘플 (즉,  $P_{i,j-1}$ )의 재구성된 값과, 각각의 샘플의 좌측에 있는 샘플의 재구성된 값 (즉,  $P_{i-1,j}$ )의 평균으로서 결정할 수도 있다. 비디오 인코더 (20)가 무손실 코딩을 사용하고 있기 때문에, 각각의 샘플의 좌측에 있는 샘플의 재구성된 값 및 각각의 샘플의 상부에 있는 샘플의 재구성된 값은 각각의 샘플의 좌측에 있는 샘플의 원래 값들 및 각각의 샘플의 상부에 있는 샘플의 원래 값과 동일하다. 이와 유사하게, 방정식들 (4) 및 (5)에 있어, 예측 블록에서 각각의 샘플 각각에 대해, 비디오 디코더 (30)는 각각의 샘플에 대한 DC 인트라 예측 값 (즉,  $DC_{i,k}$ )를, 각각의 샘플의 상부에 있는 샘플 (즉,  $P_{i,j-1}$ )의 재구성된 값과, 각각의 샘플의 좌측에 있는 샘플의 재구성된 값 (즉,  $P_{i-1,j}$ )의 평균으로서 결정할 수도 있다. 비디오 디코더 (30)가 무손실 코딩을 사용하고 있기 때문에, 각각의 샘플의

상부에 있는 샘플의 재구성된 값 및 각각의 샘플의 좌측에 있는 샘플의 재구성된 값은 각각의 샘플의 상부에 있는 샘플의 원래 값 및 각각의 샘플의 좌측에 있는 샘플의 원래 값과 동일하다.

[0084] 참조 샘플들에 대한 레스터 스캐닝 및 패딩 프로세스로 인해, 재구성된 샘플들  $P_{i,j-1}$  및  $P_{i-1,j}$ 는 항상 사용가능할 수도 있다. 다시 말해서, 현재의 샘플의 상부에 있는 샘플 및 현재의 샘플의 좌측에 있는 참조 샘플은 항상 사용가능할 수도 있다. 더욱이, DC 예측 값  $DC_{i,j}$ 를 계산하는데 사용되는 우측 시프팅 동작의 비선형 성질로 인해, 비디오 디코더 (30)가 다수의 샘플들을 병렬로 프로세싱하기가 어려울 수도 있다. 예를 들어,  $DC_{i,j+1} = (P_{i,j} + P_{i-1,j+1} + 1) \gg 1$ , 또는

$$DC_{i,j+1} = ((R_{i,j} + ((P_{i,j-1} + P_{i-1,j} + 1) \gg 1) + P_{i-1,j+1} + 1) \gg 1 \text{ 과 같이 표현될 수 있다.}$$

[0085] 상기 방정식들에서,  $R_{i,j}$ 는 로케이션 ( $i,j$ )에서의 샘플에 대한 예측 잔여이다. 우측 비트-시프트가 비선형 프로세스이기 때문에,  $DC_{i,j}$ 의 계산을 종료하기 전에  $DC_{i,j+1}$ 을 계산하는 방법이 전혀 존재하지 않을 수도 있다. 예를 들어, 비디오 디코더 (30)가 예측 블록의 다수의 샘플들을 병렬로 프로세싱하는 것이 불가능할 수도 있다.

[0086] 비디오 디코더 (30)가 무손실 코딩을 사용하여 코딩되는 현재 블록의 코딩된 표현을 디코딩할 때, 비디오 디코더 (30)는 비트스트림으로부터, 현재 블록의 잔여 샘플 값들을 나타내는 십택스 엘리먼트들을 획득할 수도 있다. 따라서, 현재 블록이 무손실 코딩을 사용하여 코딩될 때, 비디오 디코더 (30)는 현재 블록의 잔여 샘플 값들을 결정하기 위해 역양자화 또는 역변환을 적용할 필요가 없다. 비트스트림으로부터 십택스 엘리먼트들을 획득하는 것은 십택스 엘리먼트들을 엔트로피 디코딩하는 것을 수반할 수도 있다. 따라서, 현재 블록이 무손실 또는 손실 코딩을 사용하여 코딩될 때, 현재 블록의 예측 잔여들 (즉, 예측 에러)가 엔트로피 디코딩되어 있다고 가정될 수도 있다.

[0087] 따라서, 현재 블록이 무손실 또는 손실 코딩을 사용하여 코딩될 때, 예측 잔여들은, 현재 블록에 대한 예측 블록의 샘플들에 대해 DC 예측 값들을 결정할 때에 비디오 디코더 (30)가 사용하는 샘플들의 재구성된 값들을 결정할 때에 사용하기 위해 사용가능하다. 예측 잔여들이 이미 엔트로피 디코딩되어 있다고 가정하면, 하나의 샘플 지연으로 상이한 로우들에서의 샘플들의 프로세싱을 파이프라인하는 것이 가능할 수도 있다. 따라서, 본 개시물의 하나 이상의 기법들에 따르면, 비디오 디코더 (30)는 제 1 로우로부터의 하나의 샘플이 재구성된 이후 샘플들의 제 2 로우의 프로세싱을 시작할 수도 있다. 이러한 방법으로, 비디오 디코더 (30)는 샘플들의 다수의 로우들을 병렬로 프로세싱할 수도 있다. 그러므로, 예측 블록을 생성하는 것의 일부로서, 비디오 코더는 예측 블록의 상이한 로우들에서 샘플들의 프로세싱을 파이프라인할 수도 있고, 여기서, DC 예측을 위한 하나의 사이클 지연이 예측 블록의 로우들 사이에 존재한다.

[0088] 또 다른 예에서, 비디오 코더 (예컨대, 비디오 인코더 (20) 또는 비디오 디코더 (30))은 DC 예측 값  $DC_{i,j}$ 를 다음과 같이 계산할 수도 있고, 여기서  $0 \leq i \leq (M - 1)$  및  $0 \leq j \leq (N - 1)$ 이다:

$$DC_{i,j} = (P_{i,j-1} + P_{i-1,j} - P_{i-1,j-1}). \quad (6)$$

[0089] [0090] 따라서, 이 예에서, 예측 블록의 각각의 샘플 각각에 대해, 비디오 인코더 (20)는 각각의 샘플에 대한 DC 예측 값 (즉  $DC_{i,j}$ )를, 각각의 샘플의 상부에 있는 샘플 (즉,  $P_{i,j-1}$ )의 재구성된 값과 각각의 샘플의 좌측에 있는 샘플 (즉,  $P_{i-1,j}$ )의 재구성된 값의 합계, 마이너스, 각각의 샘플의 직 상부 및 좌측에 있는 샘플 (즉,  $P_{i-1,j-1}$ )의 원래 값으로서 결정할 수도 있다. 이와 유사하게, 예측 블록의 각각의 샘플 각각에 대해, 비디오 디코더 (30)는 각각의 샘플에 대한 DC 예측 값 (즉  $DC_{i,j}$ )를, 각각의 샘플의 상부에 있는 샘플 (즉,  $P_{i,j-1}$ )의 재구성된 값과 각각의 샘플의 좌측에 있는 샘플 (즉,  $P_{i-1,j}$ )의 재구성된 값의 합계, 마이너스, 각각의 샘플의 직 상부 및 좌측에 있는 샘플 (즉,  $P_{i-1,j-1}$ )의 재구성된 값으로서 결정할 수도 있다. 무손실 코딩이 사용되고 있기 때문에, 샘플들의 재구성된 값은 샘플들의 원래 값들과 동일하다.

[0091] 이 예에서, 추가적인 로직을 사용함으로써 로우에서의 다수의 샘플 값들을 프로세싱하는 것을 더 간단히 할 수도 있다. 예를 들어, 비디오 코더는  $P_{i,j}$ 를 기다리지 않고 예측 블록의 현재의 샘플  $P_{i,j}$ 에 대한  $DC_{i,j+1}$ 을 다음과 같이 계산할 수 있다.  $DC_{i,j+1}$ 은 다음과 같이 표현될 수 있다:

[0092]  $DC_{i,j+1} = ((P_{i,j-1} + P_{i-1,j} - P_{i-1,j-1}) + r_{i,j} + P_{i-1,j+1} - P_{i-1,j}) \quad (7)$

[0093] 방정식 (7) 은 다음과 같이 다시 쓸 수도 있다:

[0094]  $DC_{i,j+1} = \sum_{n=0}^j r_{i,j-n} + P_{i-1,j+1} + P_{i,-1} + P_{i-1,-1}. \quad (8)$

[0095] 방정식들 (7) 및 (8) 에 있어,  $r_{i,j}$  는 샘플  $P_{i,j}$  에 대한 예측 에러 잔여이다. 따라서, 이 예에서, 로케이션  $(i,j+1)$  (즉,  $DC_{i,j+1}$ ) 에서의 특정의 샘플에 대한 DC 인트라 예측 값의 계산은 특정의 샘플의 좌측에 있는 샘플들의 재구성된 값들에 의존하지 않는다. 대신, 특정의 샘플에 대한 DC 인트라 예측 값  $DC_{i,j+1}$  의 계산은 직상부의 샘플에 대한 재구성된 값 뿐만 아니라, 현재의 샘플의 좌측에 있는 모든 샘플들, 및 동일한 로우 및 상부에 있는 로우에서의 참조 샘플들에 대한 잔여 값들에 의존할 수도 있다. 이것은 전체 로우에 대한 잔여들이 이미 디코딩되어 있다고 가정하면, (비디오 인코더 (20) 뿐만 아니라) 비디오 디코더 (30) 으로 하여금, 블록의 로우에서의 모든 샘플들에 대한 DC 예측 값들을 병렬로 계산가능하게 할 수도 있다.

[0096] 무슨 실 코딩이 사용되고 있기 때문에, 샘플들 (즉,  $P_{i,j-1}, P_{i-1,j}$ , 및  $P_{i-1,j-1}$ ) 의 재구성된 값은 샘플들의 원래 값들과 동일하다. 본 개시물에서 다른 어딘가에 추가로 설명된 바와 같이, 이 기법은 순실 코딩에 적용될 수도 있다. 그 경우, 병렬화를 유지하기 위해, DC 예측을 위해 좌측에 있는 샘플들에 대한 비클리핑된 재구성된 값을 사용하는 것이 필요하다. 상부에 있는 로우로부터의 재구성된 샘플들은 클리핑되거나 또는 클리핑되지 않을 수도 있다. 예를 들어, 8-비트 비디오 시퀀스에 대해, 재구성된 샘플들은 간격 [0, 255] 으로 클리핑된다.

[0097] 또 다른 예에서, 좌측, 좌상부, 상부 및 우상부 샘플들이 DC 인트라 예측 값들을 결정할 때에 사용된다. 이 예에서, 비디오 코더는 다음과 같이 예측 블록의 현재의 샘플  $P_{i,j}$  에 대한 DC 예측 값  $DC_{i,j}$  를 계산할 수도 있고, 여기서  $0 \leq i \leq (M - 1)$  및  $0 \leq j \leq (N - 1)$  이다:

[0098]  $DC_{i,j} = (P_{i,j-1} + P_{i-1,j} + P_{i-1,j-1} + P_{i-1,j+1} + 2) \gg 2, \quad (9)$

[0099] 또는 유사하게,

[0100]  $DC_{i,j} = (P_{i,j-1} + P_{i-1,j} + P_{i-1,j-1} + P_{i-1,j+1}) \gg 2. \quad (10)$

[0101] 따라서, 방정식들 (9) 및 (10) 의 예들에 있어, 예측 블록의 각각의 샘플 각각에 대해, 각각의 샘플에 대한 DC 인트라 예측 값 (즉,  $DC_{i,j}$ ) 는 각각의 샘플의 상부에 있는 재구성된 샘플 (즉,  $P_{i,j-1}$ ), 각각의 샘플의 좌측에 있는 재구성된 샘플 (즉,  $P_{i-1,j}$ ), 각각의 샘플의 상부 및 좌측에 있는 재구성된 샘플 (즉,  $P_{i-1,j-1}$ ), 및 각각의 샘플의 상부 및 우측에 있는 재구성된 샘플 (즉,  $P_{i-1,j+1}$ ) 의 평균이다. 무슨 실 코딩이 사용되고 있기 때문에, 샘플들 (즉,  $P_{i,j-1}, P_{i-1,j}, P_{i-1,j-1}$ , 및  $P_{i-1,j+1}$ ) 의 재구성된 값들은 샘플들의 원래 값들과 동일하다.

[0102] DC 인트라 예측 값들을 결정하는데 방정식들 (9) 또는 (10) 을 사용하는 예들에서, 최종 칼럼 ( $j = (N - 1), i > 0$ ) 에서의 샘플들에 대해, 우상부 샘플은 사용불가능하다. 이를 극복하기 위해, 비디오 코더는 상부 및 우상부 샘플들 (즉, 각각,  $P_{i-1,j}$  및  $P_{i-1,j+1}$ ) 이 동일한 값을 가진다고 가정할 수도 있다. 비디오 코더가 DC 인트라 예측 값들을 결정하는데 방정식들 (9) 또는 (10) 을 사용하는 또 다른 예에서, 비디오 코더는 단지 가용 샘플들만을 사용하기 위해 DC 예측을 수정할 수도 있다. 일부 예들에서, 샘플은 샘플이 현재의 슬라이스 또는 화상의 경계를 내에 있지 않거나, 또는 아직 코딩되어 있지 않으면, 사용불가능할 수도 있다.

[0103] 더욱이, 본 개시물의 또 다른 예에 따르면, 비디오 코더는 TU 사이즈보다 작은 블록 사이즈에 대해 DC 인트라 예측을 수행할 수도 있다. 예를 들어, TU 사이즈에 관계없이, 비디오 코더는  $2 \times 2$  블록들에 대해 DC 인트라 예측을 수행할 수도 있다. 비디오 코더는 예측 블록의  $2 \times 2$  블록들을 레스터 스캐닝 순서로 프로세싱할 수도 있다. 이 예에서, 샘플들  $P_{2i,2j}, P_{2i,2j+1}, P_{2i+1,2j}$ , 및  $P_{2i+1,2j+1}$  에 대해, 비디오 코더는 DC 인트라 예측 값을 다음과 같이 계산한다:

[0104]  $(P_{2i-1,2j} + P_{2i-1,2j+1} + P_{2i,2j-1} + P_{2i+1,2j-1} + 2) \gg 2, \quad (11)$

[0105] 또는 유사하게,

$$(P_{2i-1,2j} + P_{2i-1,2j+1} + P_{2i,2j-1} + P_{2i+1,2j-1}) \gg 2. \quad (12)$$

[0106] 이 예에서,  $0 \leq i \leq ((M / 2) - 1)$  및  $0 \leq j \leq ((N / 2) - 1)$  이고, 여기서, M은 블록의 높이이고 N은 블록의 폭이다. 더욱이, 이 예에서, M 및 N이 둘 다 짹수라고 가정된다. 이 예에서, 비디오 코더는 4개의 샘플들을 병렬로 프로세싱할 수 있다. 이 예에서, 비디오 코더는  $2 \times 2$  블록의 4개의 샘플들의 각각의 DC 인트라 예측 값들을 병렬로 결정 가능할 수도 있다. 유사한 예들에서, 비디오 코더는  $2 \times 2$  블록들 대신,  $4 \times 4$  블록들 또는  $8 \times 8$  블록들을 사용할 수도 있다.

[0107] 본 개시물의 또 다른 예에 따르면, 정상 DC 예측을 수행한 후 잔여들 사이의 상관이 사용된다. 예를 들어,  $r_{i,j}$  를, HEVC (예컨대, HEVC 작업 초안 10)에 규정된 바와 같이 DC 인트라 예측을 수행한 후 예측 잔여들이라 하고, 여기서,  $0 \leq i \leq (M - 1)$  및  $0 \leq j \leq (N - 1)$  이다. 예를 들어,  $r_{i,j}$  는 상기 방정식 (1)에서 설명된 바와 같이 DC 인트라 예측을  $4 \times 4$  블록에 대해 수행한 후 예측 잔여 값일 수도 있다. 이 예에서, 비디오 코더는 그 후 중간 값을  $s_{i,j}$  를 생성할 수도 있고, 여기서,  $0 \leq i \leq (M - 1)$  및  $0 \leq j \leq (N - 1)$  이다. 비디오 코더는 중간 값을  $s_{i,j}$  를 다음과 같이 생성할 수도 있다:

$$s_{i,j} = r_{i,2j}, \quad s_{i,(j+(N/2))} = r_{i,2j} - r_{i,2j+1} \quad (13)$$

[0109] 상기 방정식 (13)에서,  $0 \leq i \leq (M - 1)$ ,  $0 \leq j \leq ((N/2) - 1)$  이다.

[0110] 비디오 코더는 그 후 수정된 잔여 값을  $t_{i,j}$  를 다음과 같이 생성할 수도 있고, 여기서  $0 \leq i \leq (M - 1)$  및  $0 \leq j \leq (N - 1)$  이다:

$$t_{i,j} = s_{2i,j}, \quad t_{(i+(M/2)),j} = s_{2i,j} - s_{2i+1,j} \quad (14)$$

[0111] 방정식 (14)에서,  $0 \leq i \leq ((M / 2) - 1)$  및  $0 \leq j \leq (N - 1)$  이다.

[0112] 비디오 인코더는 표준적인 HEVC (예컨대, HEVC 작업 초안 10)에서 설명된 바와 같이, 수정된 잔여들,  $t_{i,j}$  를 엔트로피 인코딩할 수도 있다. 디코더 측에서 (예컨대, 비디오 디코더 (30)에서), 이 프로세스는 역전된다. 예를 들어, 비디오 디코더 (30)는 다음을 결정할 수도 있고,

$$s_{2i,j} = t_{i,j}, \quad s_{2i+1,j} = t_{i,j} - t_{(i+(M/2)),j}, \quad (15)$$

[0113] 여기서,  $0 \leq i \leq ((M / 2) - 1)$  및  $0 \leq j \leq (N - 1)$  이다. 비디오 디코더 (30)는 또한 다음을 결정 할 수도 있고,

$$r_{i,2j} = s_{i,j}, \quad r_{i,2j+1} = s_{i,j} - s_{i,(j+(N/2))}, \quad (16)$$

[0114] 여기서,  $0 \leq i \leq ((M - 1)$  및  $0 \leq j \leq ((N / 2) - 1)$  이다. 이 예는 M 및 N이 둘 다 짹수라고 가정 한다.

[0115] 본 개시물의 또 다른 예에서, 잠재적으로 더 나은 예측자가 간단한 차이를 취하는 대신, 사용될 수 있다. 예를 들어, 비디오 코더는  $s_{i,j}$  를 다음과 같이 결정할 수도 있다:

$$s_{i,j} = r_{i,2j+1}, \quad s_{i,(j+(N/2))} = P_{i,2j} - ((P_{i,2j-1} + P_{i,2j+1} + 1) \gg 1) \quad (17)$$

$$t_{i,j} = s_{2i+1,j}, \quad t_{(i+(M/2)),j} = s_{2i,j} - ((s_{2i+1,j} + s_{2i+1,j} + 1) \gg 1) \quad (18)$$

[0116] 상기 방정식 (17)에서,  $0 \leq i \leq M$ ,  $0 \leq j < (N/2)$  이다. 상기 방정식 (18)에서,  $0 \leq i \leq M/2$ ,  $0 \leq j < N$  이다.

[0117] 본 개시물의 여러 다른 예들에서 설명되는 기법들이 변환에 스kip될 때 손실 코딩할 때에 DC 인트라 예측 모드를 향상시키기 위해 적용될 수 있다. 다시 말해서, 본 개시물의 여러 다른 예들은 비디오 인코더가 변환 블록의 잔여 샘플들에 변환을 적용하지 않지만 변환 블록의 잔여 샘플들을 양자화할 때 DC 인트라 예측 모드를 향상

시키기 위해 적용될 수도 있다. 예를 들어, 상기 단락 [0083]에서 설명된 예에서, 인과적 이웃이 현재의 샘플에 대한 DC 예측 값을 계산하기 위해 사용된다. 이 예에서, 무손실 코딩의 경우에 행해지는 바와 같이 DC 예측 값을 계산하는데 원래 샘플 값들을 사용하는 대신, 인과적 이웃에서 재구성된 (양자화된) 샘플 값들이 사용될 수도 있다. 변환의 적용이 스kip되기 때문에, 인과적 이웃에서의 재구성된 값들이 사용가능하다.

병렬화 이점들을 유지하기 위해서, 클리핑 동작이 전체 로우에 대한 프로세싱이 완료될 때까지 현재의 로우로부터의 재구성된 값들에 적용되지 않는다는 점에 유의해야 한다. 상부의 로우에 대해서는, 클리핑된 또는 비클리핑된 재구성된 값들이 사용될 수도 있다.

[0124] 이와 유사하게, 상기 단락 [0092]에서 설명된 바와 같이, TU는 더 작은 블록들 (예컨대,  $2 \times 2$  블록들)로 분할되고, DC 예측 값이 각각의 더 작은 블록에 대해 계산된다. 무손실 코딩의 경우에 행해지는 바와 같이 DC 예측 값을 계산하는데 원래 샘플 값들을 사용하는 대신, 재구성된 (양자화된) 샘플 값들이 변환이 스kip되는 손실 코딩의 경우에 사용될 수도 있다.

[0125] 손실 코딩의 기법들은 본 개시물에서 위에서 설명된다. 일부 이러한 기법들에 따르면,  $P_{i,j}$  가, 원래 샘플 값들을 나타낸다고 가정하고, 여기서  $0 \leq i \leq (M - 1)$  및  $0 \leq j \leq (N - 1)$  이다. 더욱이,  $Q(P_{i,j})$  가  $P_{i,j}$ 의 양자화된 버전을 나타낸다고 하자. 그 후, 손실 코딩을 사용하는 본 개시물의 추가적인 예에 따라서, 비디오 코더는 DC 예측 값  $DC_{i,j}$ 를 다음과 같이 계산할 수도 있다:

$$DC_{i,j} = (Q(P_{i,j-1}) + Q(P_{i-1,j}) - Q(P_{i-1,j-1})) \quad (\text{방정식 DC1})$$

[0126] 방정식 DC1은 샘플 값들 (즉,  $P_{i,j-1}$ ,  $P_{i-1,j}$ , 및  $P_{i-1,j-1}$ )이 양자화된 후 역양자화된다는 점을 제외하고는, 상기 방정식 (6)과 유사하다는 점에 유의한다. 본 개시물은 이러한 샘플들을 원래 샘플들의 양자화된 샘플들 또는 양자화된 버전들로서 지칭할 수도 있다. 따라서, 방정식 DC1에 있어, 예측 블록의 각각의 샘플 각각에 대해, 비디오 코더는 각각의 샘플에 대한 DC 예측 값을, 각각의 샘플의 상부에 있는 원래 샘플의 양자화된 버전 (즉,  $Q(P_{i,j-1})$ )과 각각의 샘플의 좌측에 있는 원래 샘플의 양자화된 버전 (즉,  $Q(P_{i-1,j-1})$ )의 합계, 마이너스, 각각의 샘플의 상부 및 좌측에 있는 원래 샘플의 양자화된 버전 (즉,  $Q(P_{i-1,j-1})$ )로서 계산할 수도 있다. 방정식 DC1에 있어, 유형  $Q(P_{i,j})$ 의 항들은 재구성된 샘플들이다. 비디오 코더는 그 후 예측 잔여를  $r_{i,j} = P_{i,j} - DC_{i,j}$ 로서 계산할 수도 있다. 다시 말해서, 잔여 값  $r_{i,j}$ 는 샘플 값  $P_{i,j}$ , 마이너스, 대응하는 DC 인트라 예측 값  $DC_{i,j}$ 와 같다. 양자화 및 역양자화 이후 재구성된 잔여는  $Q(r_{i,j})$ 로 나타내어진다.

[0128] 이전 단락에서 설명된 예는 디코더 측 상에서 일부 바람직한 처리량 성질들을 가질 수도 있다. 예를 들어, 비디오 디코더 (30)가 블록의 로우 (또는, 칼럼)에서의 모든 샘플들에 대해 재구성된 샘플 값들을 동시에 계산하는 것이 가능할 수도 있다. 예를 들어, 비디오 디코더 (30)는 재구성된 샘플 값들을 다음과 같이 획득할 수도 있다:

$$Q(P_{i,j}) = Q(r_{i,j}) + DC_{i,j}, \text{ 또는}$$

$$Q(P_{i,j}) = (\sum_{n=0}^j Q(r_{i,j-n})) + Q(P_{i-1,j}) + Q(P_{i,-1}) + Q(P_{i-1,-1}) \quad (\text{방정식 DC2})$$

[0130] [0131] 방정식 DC2에 있어,  $Q(P_{i-1,j})$ 는 적합하게 클리핑될 수도 있는 재구성된 샘플을 나타낸다. 예를 들어, 재구성된 값을 8의 입력 비트 심도로 적합하게 클리핑하기 위해,  $Q(P_{i-1,j})$ 의 값들은 0과 255 사이에 클리핑된다.

[0132] 더욱이, 일부 예들에서,  $Q(P_{i-1,j})$ 의 비-클리핑된 버전들을 사용하는 것이 가능할 수도 있다. 다른 값들,  $Q(P_{i-1,-1})$  및  $Q(P_{i,-1})$ 은 이전에 재구성된 블록들에 속하고, 이미 클리핑되어 있다. 이러한 예들에서, 방정식 DC2에서 재구성된 샘플  $Q(P_{i-1,j})$ 는 클리핑되지 않지만, 처리량에 영향을 미치지 않고 적합하게 클리핑될 수 있다. 방정식 DC1에 규정되는 예측은, 비디오 디코더가 재구성을 위해 방정식 DC2를 사용하면, 단지 근사치이다. 방정식 DC1에서의 예측은, 클리핑된 버전인,  $Q(P_{i,j-1})$  대신, 비클리핑된 버전이 사용되기 때문에, 단지 근사치이다. 이러한 경우,  $Q(P_{i,j-1})$ 의 비클리핑된 버전은, DC 예측을 발생시켜 인코더와 디코더 사이에 드리프트를 피하기 위해서, 인코더 측 상에서도 또한 사용될 수도 있다. 클리핑된 버전을 사용하는 것이 가능하지만, 그 후 샘플들은 하나하나씩 재구성되어야 할 수도 있고, 이에 따라서 처리량에 영향을 미칠 수도

있다. 이것은, 그 경우, 방정식 DC1 이 재구성을 위해 사용되어야 할 수도 있기 때문이다. 이것은 샘플의 재구성이 샘플의 좌측에 있는 샘플의 재구성의 완료에 의존할 수도 있다는 것을 의미한다. 어떻게 샘플들의 로우가 병렬로 재구성될 수 있는지는 본 개시물에서 설명되었다. 유사한 프로세스가 칼럼에서의 모든 샘플들을 병렬로 재구성하기 위해 뒤이어질 수 있다. 더 적은 병렬성이 요망되면, 합계 항 (summation term) 은 더 작은 청크들로 쪼개질 수도 있고, 이에 의해 처리량을 잠재적으로 감소시키지만 샘플을 재구성하는데 요구되는 추가적인 동작들의 평균 개수를 감소시킬 수도 있다.

[0133] 따라서, 위에서 설명된 손실 코딩 예들 중 적어도 일부에서, 비디오 코더는 예측 블록을 생성할 수도 있다. 예측 블록을 생성하는 것의 일부로서, 비디오 코더는 제 1 재구성된 샘플 (예컨대,  $Q(P_{i-1,j})$ ) 및 제 2 재구성된 샘플 (예컨대,  $Q(P_{i,j-1})$ ) 중 적어도 하나를 예측 블록의 현재의 샘플의 DC 예측을 위해 사용할 수도 있다. 제 1 재구성된 샘플은 예측 블록의 현재의 로우에서 현재의 샘플의 좌측에 있는 샘플에 대응할 수도 있다. 제 2 재구성된 샘플은 현재의 로우의 상부에 있는 예측 블록의 로우에서의 샘플에 대응할 수도 있다. 비디오 코더는 예측 블록의 샘플들을 대응하는 잔여 샘플들에 가산함으로써, 손실 코딩을 사용하여 코딩된 코딩 블록을 재구성할 수도 있다.

[0134] 더욱이, 위에서 설명된 손실 코딩 예들 중 적어도 일부에서, 비디오 코더는 현재의 샘플의 상부에 있는 샘플에 대응하는 재구성된 값  $Q(P_{i,j-1})$  을 결정할 수도 있다. 게다가, 비디오 코더는 현재의 샘플의 좌측에 있는 샘플에 대응하는 재구성된 값  $Q(P_{i-1,j})$  을 결정할 수도 있다. 비디오 코더는 또한 현재의 샘플의 좌측 및 상부에 있는 샘플에 대응하는 재구성된 값  $Q(P_{i-1,j-1})$  을 결정할 수도 있다. 비디오 코더는 현재의 샘플  $P_{i,j}$  에 대한 DC 예측 값  $DC_{i,j}$  를 다음과 같이 계산할 수도 있다:

$$DC_{i,j} = (Q(P_{i,j-1}) + Q(P_{i-1,j}) - Q(P_{i-1,j-1})).$$

[0135] 위에서 설명된 손실 코딩 예들 중 적어도 일부에서, 비디오 코더는  $Q(P_{i,j-1})$ ,  $Q(P_{i-1,j})$ , 및  $Q(P_{i-1,j-1})$  중에서의 각각의 재구성된 값 각각에 대해, 각각의 재구성된 값을 다음 방법들 중 하나로 결정할 수도 있다. 첫째, 비디오 코더는 각각의 재구성된 값을, 주어진 샘플에 대한 역양자화된 잔여 값 (예컨대,  $Q(r_{i,j})$ ), 플러스, 대응하는 샘플에 대한 DC 예측 값 (예컨대,  $DC_{i,j}$ ) 로서 결정할 수도 있다. 둘째, 방정식 DC2 에서 설명된 바와 같이, 비디오 코더는 각각의 재구성된 값을, 주어진 샘플의 상부에 있는 샘플들에 대한 역양자화된 잔여 값들의 합계 (예컨대,  $\sum_{n=0}^j Q(r_{i,j-n})$ ), 플러스, 주어진 샘플의 좌측에 있는 샘플에 대응하는 재구성된 값 (예컨대,  $Q(P_{i-1,j})$ ), 플러스, 주어진 샘플을 포함하는 예측 블록의 칼럼의 최상부 샘플의 상부에 있는 참조 샘플에 대응하는 재구성된 값 (예컨대,  $Q(P_{i,-1})$ ), 플러스, 주어진 샘플의 좌측에 있는 예측 블록의 칼럼의 최상부 샘플의 상부에 있는 참조 샘플에 대응하는 재구성된 값 (예컨대,  $Q(P_{i-1,-1})$ ) 로서 결정할 수도 있다. 이 예에서, 주어진 샘플은 각각의 재구성된 값에 대응한다. 이러한 일부 예들에서, 비디오 코더는 주어진 샘플의 좌측에 있는 샘플에 대응하는 재구성된 값을 클리핑할 수도 있다.

[0137] 본 개시물의 또 다른 예는 평면 모드에 있어 예측 프로세스에 대한 변경을 제안한다. 평면 모드에 대한 방정식들 (예컨대, 방정식 (2)) 를 고려하는 경우, 최종 로우로부터의 원래 샘플들이  $P_{M-1}$  을 사용하는 대신, 수직 예측을 수행하기 위해 사용되면, 수직 예측이 더 정확할 수도 있다. 이와 유사하게, 최종 칼럼으로부터의 원래 샘플들이  $P_{-1,N}$  을 사용하는 대신, 수평 예측을 수행하기 위해 사용되면, 수평 예측이 더 정확할 수도 있다. 인트라 평면 모드에서 원래 샘플들의 사용은 본 개시물의 하나 이상의 예들 이면에 숨겨진 기본적인 아이디어이다.

[0138] 예를 들어, 비디오 인코더 (예컨대, 비디오 인코더 (20)) 은 최종 로우 및 칼럼에 대한 HEVC 평면 예측 (예컨대, HEVC 작업 초안 10 에서 설명된 바와 같은 평면 예측), 즉,  $T_{i,j}$  를 사용할 수도 있고, 여기서  $i = (M - 1)$  또는  $j = (N - 1)$  이다. 비디오 코더는 원래 샘플 값들로부터 예측 값들을 감산하여, 최종 로우 및 칼럼에 대한 잔여들을 생성할 수도 있다. 즉, 비디오 인코더는  $r_{i,j}$  를 결정할 수도 있고, 여기서  $i = (M - 1)$  또는  $j = (N - 1)$  이다. 비디오 인코더는 그 후 예측 값들  $T_{i,j}$  를 다음과 같이 생성할 수도 있고, 여기서  $0 \leq i \leq (M - 2)$  및  $0 \leq j \leq (N - 2)$  이다:

[0139]  $T_{i,j}^V = (M - i) * P_{-1,j} + i * P_{M-1,j},$

[0140]  $T_{i,j}^H = (N - j) * P_{i,-1} + j * P_{i,N-1}$  및

[0141]  $T_{i,j} = (T_{i,j}^V + T_{i,j}^H + N) \gg (\log_2 N + 1). \quad (19)$

[0142] 이 예는  $M = N$ 이라고 가정한다. 그러나, 직사각형의 블록들에 대한 이 개념의 확장은 직접적이다 (straightforward). 비디오 인코더는 원래 샘플 값들로부터 예측 값들을 감산함으로써 나머지 잔여들,  $r_{i,j}$  를 생성할 수도 있고, 여기서  $0 \leq i \leq (M - 2)$  및  $0 \leq j \leq (N - 2)$  이다. 비디오 인코더는 HEVC (예컨대, HEVC 작업 초안 10) 에서와 같이 잔여들,  $r_{i,j}$  의 전체 블록을 엔트로피 인코딩할 수도 있고, 여기서  $0 \leq i \leq (M - 1)$  및  $0 \leq j \leq (N - 1)$  이다.

[0143] 이 예에서, 디코더 측 상에서, 비디오 디코더 (예컨대, 비디오 디코더 (30)) 은 예측 잔여들의 전체 블록을 엔트로피 디코딩하여 잔여 값들  $r_{i,j}$  를 생성할 수도 있고, 여기서  $0 \leq i \leq (M - 1)$  및  $0 \leq j \leq (N - 1)$  이다.

비디오 디코더는 그 후 블록의 최종 로우 및 칼럼에서의 샘플들에 대해 HEVC 평면 예측을 수행할 수도 있다. 즉, 비디오 코더는 HEVC (예컨대, HEVC 작업 초안 10) 에 규정된 바와 같이  $T_{i,j}$  를 결정할 수도 있고, 여기서  $i = (M - 1)$  또는  $j = (N - 1)$  이다. 예를 들어, 비디오 코더는 상기 방정식 (3) 을 사용하여  $T_{i,j}$  를 결정할 수도 있다. 더욱이, 이 예에서, 비디오 디코더는 최종 로우 및 칼럼에 대한 잔여 값들을 상기 예측 값들에 가산하여, 최종 로우 및 칼럼에 대한 원래 샘플 값들을 재구성할 수도 있다. 그 후, 비디오 디코더는 정확히 상기 인코더 측 상에서와 같이 예측 값들  $T_{i,j}$  를 생성하고, 여기서,  $0 \leq i \leq (M - 2)$  및  $0 \leq j \leq (N - 2)$  이다. 비디오 디코더는 잔여 값들  $r_{i,j}$  를, 예측 값들에 가산하여, 블록에서의 나머지 샘플 위치들에 대한 샘플 값들을 재구성하고, 여기서  $0 \leq i \leq (M - 2)$  및  $0 \leq j \leq (N - 2)$  이다.

[0144] 본 개시물의 또 다른 예에서, (비디오 인코더 (20) 과 같은) 비디오 인코더는 최종 로우 및 칼럼 상에서의 엘리먼트에 대한 HEVC 평면 예측 (예컨대, HEVC 작업 초안 10 에서 설명된 바와 같은 평면 예측), 즉,  $T_{i,j}$  를 사용하고, 여기서,  $i = (M - 1)$  및  $j = (N - 1)$  이다. 이 예에서, 비디오 인코더는 그 위치에 대한 원래 샘플 값들로부터 그 위치에 대한 예측 값을 감산하여, 그 위치에 대한 잔여 값을 생성한다. 비디오 인코더는 그 후 최종 로우 및 칼럼의 엘리먼트들을 다음과 같이 이중 선형적으로 (bilinearly) 예측할 수도 있다:

$$T_{M-1,j} = ((N - j) * P_{M-1,-1} + j * P_{M-1,N-1}) \gg (\log_2 N),$$

[0145]  $T_{i,N-1} = ((M - i) * P_{-1,N-1} + i * P_{M-1,N-1}) \gg (\log_2 M). \quad (20)$

[0146] 더욱이, 이 예에서, 비디오 인코더는 예측 값들  $T_{i,j}$  를 다음과 같이 생성하고, 여기서  $0 \leq i \leq (M - 2)$  및  $0 \leq j \leq (N - 2)$  이다:

[0147]  $T_{i,j}^V = (M - i) * P_{-1,j} + i * P_{M-1,j},$

[0148]  $T_{i,j}^H = (N - j) * P_{i,-1} + j * P_{i,N-1}$  및

[0149]  $T_{i,j} = (T_{i,j}^V + T_{i,j}^H + N) \gg (\log_2 N + 1). \quad (21)$

[0150] 이 예에서, 비디오 인코더는 원래 샘플 값들로부터 예측 값들을 감산함으로써 잔여 값들  $r_{i,j}$  를 생성하고, 여기서  $0 \leq i \leq (M - 1)$  및  $0 \leq j \leq (N - 1)$  이다. 더욱이, 이 예에서, 비디오 인코더는 HEVC (예컨대, HEVC 작업 초안 10) 에서와 같이 잔여들,  $r_{i,j}$  의 전체 블록을 엔트로피 디코딩하고, 여기서  $0 \leq i \leq (M - 1)$  및  $0 \leq j \leq (N - 1)$  이다.

[0151] 본 개시물의 또 다른 예는 이전 예들에 대한 개선을 제공한다. 이 예에서, 본 개시물의 예측 기법들로 인한 잔여 배포는 변환에 채용될 때 비디오 압축에서의 일반적인 방향 (common one) 의 역방향인 경향이 있다. 일반적으로, 잔여는 낮은 주파수들에서 더 높은 값들을, 그리고 높은 주파수들에서 높은 예상된 값들을 갖는다.

상기 예들로부터 유래하는 잔여에 대해, 최종 로우 및 칼럼은 더 큰 값들을 갖는 경향이 있다. 변환 잔

여를 위해 설계되는 엔트로피 코딩 방법의 이점을 취하면서 성능을 향상시키는 접근법은 위에서 제공된 예들에서 예측으로부터 유래하는 잔여를 회전시키는 것이다. 즉, 잔여는 180도 회전되고, 따라서 좌상부 부분이 우하측이 되고, 반대의 경우도 마찬가지이다. 그 후, 이 회전된 잔여는 엔트로피 코딩된다. 대응하여, 디코더에서, 잔여가 획득되고 그 후 180도로 회전된다.

[0152] 본 개시물의 다른 예들에서, 평면 예측 프로세스는 다음과 같이 수정된다. 제 1 로우 및 칼럼에 대해, 예측은  $T_{i,j}$  및  $r_{i,j}$ 를 생성하기 위해 HEVC 평면 모드 (예컨대, HEVC 작업 초안 10에서 설명된 바와 같은 평면 모드)의 경우에서처럼 수행되고, 여기서,  $i = 0$  또는  $j = 0$ 이다. 이 예에서, 제 1 로우 및 칼럼에 대한 예측을 생성하는데 HEVC 방법 (예컨대, HEVC 작업 초안 10, 방정식 (3), 등에서 설명되는 방법)을 사용하는 대신, 다른 방법들이 사용된다. 예를 들어,  $P_{0,0}$ 은  $(P_{-1,0}+P_{0,-1}+1)>>1$ 로서 예측될 수 있다. 제 1 칼럼에서 나머지 샘플들은 동일한 로우에서의 좌측 샘플을 사용하여 예측될 수 있다. 이와 유사하게, 제 1 로우에서 나머지 샘플들은 동일한 칼럼에서의 상부 샘플을 사용하여 예측될 수 있다. 나머지 위치들에 대해, 평면 예측,  $T_{i,j}$ 는 다음과 같이 생성되고, 여기서  $1 \leq i \leq (M - 1)$  및  $1 \leq j \leq (N - 1)$ 이다:

$$[0153] T_{i,j}^V = P_{i-1,j} + w_v * (P_{i-1,j} - P_{i-2,j}),$$

$$[0154] T_{i,j}^H = P_{i,j-1} + w_h * (P_{i,j-1} - P_{i,j-2}) \text{ 및}$$

$$[0155] T_{i,j} = (T_{i,j}^V + T_{i,j}^H + 1) \gg 1. \quad (22)$$

[0156] 방정식 (22)에서,  $w_v$  및  $w_h$ 는 가중치들이다. 일부 예들에서, 0.5의 값이 비트-시프트로서 구현될 수 있으므로, 0.5의 값이  $w_v$  및  $w_h$  양쪽에 대해 사용된다. 나머지 잔여들 (즉,  $r_{i,j}$ )는 원래 샘플 값들로부터 예측값을 감산함으로써 생성되고, 여기서  $1 \leq i \leq (M - 1)$  및  $1 \leq j \leq (N - 1)$ 이다. 잔여들 (즉,  $r_{i,j}$ )의 전체 블록은 HEVC (예컨대, HEVC 작업 초안 10)에서와 같이 엔트로피 코딩될 수도 있고, 여기서  $0 \leq i \leq (M - 1)$  및  $0 \leq j \leq (N - 1)$ 이다.

[0157] 이 예에서, 디코더 측 상에서, 비디오 디코더 (예컨대, 비디오 디코더 (30))은 예측 잔여들의 전체 블록을 엔트로피 디코딩하여 잔여 값들 (즉,  $r_{i,j}$ )를 생성할 수도 있고, 여기서  $0 \leq i \leq (M - 1)$  및  $0 \leq j \leq (N - 1)$ 이다. 그 후, 비디오 디코더는 비디오 인코더에 의해 사용되는 임의의 다른 방법에 의해, HEVC (예컨대, HEVC 작업 초안 10)에 규정된 바와 같이 또는 위에서 설명한 바와 같이, 제 1 로우 및 칼럼에서의 샘플들에 대한 평면 예측 (즉,  $T_{i,j}$ )를 수행하고, 여기서,  $i = 0$  또는  $j = 0$ 이다. 더욱이, 이 예에서, 비디오 디코더는 제 1 로우 및 칼럼에 대한 잔여 값들을 상기 예측 값들에 가산하여, 제 1 로우 및 칼럼에 대한 원래 샘플 값들을 재구성한다. 그 후에, 비디오 디코더는 정확히 상기 인코더 측 상에서와 같이 예측 값들 (즉,  $T_{i,j}$ )를 생성하고, 여기서  $1 \leq i \leq (M - 1)$  및  $1 \leq j \leq (N - 1)$ 이다. 이 예에서, 비디오 디코더는 잔여 값들 (즉,  $r_{i,j}$ )를 예측 값들에 가산하여, 블록에서의 나머지 위치들에 대한 샘플 값들을 재구성하고, 여기서  $1 \leq i \leq (M - 1)$  및  $1 \leq j \leq (N - 1)$ 이다.

[0158] 상기 단락들은 무손실 코딩을 위해 평면 예측을 수행하는 상이한 방법을 기술한다. 본 개시물의 추가적인 예에서, 최우측 칼럼 및 하부 로우는 HEVC (예컨대, HEVC 작업 초안 10)에서 설명된 바와 같이 평면 예측 프로시저를 사용하여 예측된다. 이 예에서, 최우측 칼럼 및 하부 로우에 대한 원래 샘플 값들은 그 후 블록의 나머지 샘플들에 대한 평면 또는 각도 예측을 수행하는데 사용된다. 손실 코딩의 경우에 변환이 스kip될 때, 최우측 칼럼 및 하부 로우에 대한 원래 샘플 값들을 사용하는 대신, 재구성된 (양자화된) 샘플 값들이 나머지 샘플들에 대해 평면 또는 각도 예측을 수행하기 위해 사용될 수도 있다.

[0159] HEVC (예컨대, HEVC 작업 초안 10)에서, 제 1 로우 및 칼럼에 대한 DC 예측 값들이 필터링된다 (DC 예측 필터링). 이와 유사하게, 수평 및 수직 인트라 예측 모드들에 대해, 예측 값들의 제 1 로우 및 칼럼이, 각각, 필터링된다 (그레이디언트 필터링 (gradient filtering)). 위에서 설명된 방법들이 DC, 수평 또는 수직 인트라 예측 모드들에 대한 손실 코딩에 적용될 때, 그리고 변환이 스kip될 때, DC 예측 필터링 또는 그레이디언트 필터링은 스kip될 수도 있다.

[0160] 본 개시물의 또 다른 예에서, 평면 예측 모드를 변경하는 대신, 동일한 개념이 각도 인트라 예측 모드들에 적용된다. 각각의 각도 모드에 대해, 최종 로우 및 칼럼은 HEVC (예컨대, HEVC 작업 초안 10, 방정식 (3), 등)

에 규정된 바와 같이 예측된다. 그 후, 최종 로우 및 칼럼에 대한 원래 샘플 값들은 참조 샘플들에 더해서 인트라 예측을 수행하는데 사용된다. 도 4 는 예측을 위해 사용되는 샘플들을 나타낸다. 도 4 에서 음 영처리된 위치들은 예측을 수행하기 위해 참조 샘플들로서 사용되는 위치들이다. 특정의 예측 방향 및 특정의 위치에 대해, 참조 샘플들  $P_{-1,j}$ , (또는, 그들의 이중선형 내삽) 이, 예측 값으로서 사용되었더라면, 우측 칼럼으로부터의 원래 샘플 값들 (즉,  $P_{i,N-1}$ ) 이 대신 사용되고, 여기서  $N \leq j \leq 2N$  이고,  $0 \leq i \leq M - 1$  이다. 예측 각도가 최우측 칼럼을 인터셉트하는 위치가 작은 부분 (fraction) 이면, 이중선형 내삽법 또는 임의의 다른 적합한 내삽 방법이 사용될 수도 있다. 일 예로서,  $4 \times 4$  블록을 고려한다. 인트라 예측 모드 (34) 에 있어, 샘플 (2, 2) 에 대해, 대응하는 참조 샘플은  $P_{-1,6}$  일 것이다. 따라서, 모드 (34) 에 있어, HEVC 예측은  $T_{2,2} = P_{-1,5}$  일 것이다. 대신, 이 예에서,  $T_{2,2} = P_{1,3}$  이다.

[0161] 이와 유사하게, 특정의 예측 방향 및 특정의 위치에 대해, 참조 샘플들  $P_{i,-1}$ , (또는, 그들의 이중선형 내삽) 이 예측 값들로서 사용되었으면, 하부 로우로부터의 원래 샘플 값들 (즉,  $P_{M-1,j}$ ) 가 대신 사용되고,  $M \leq i \leq 2M$  이고,  $0 \leq j \leq N - 1$  이다. 예측 각도가 하부 로우를 인터셉트하는 위치가 작은 부분이면, 이중선형 내삽법 또는 또 다른 적합한 내삽 기법이 사용될 수도 있다. 일 예로서,  $4 \times 4$  블록을 고려한다. 이 예에서, 인트라 예측 모드 2 에 있어, 샘플 (2, 1) 에 대해, 대응하는 참조 샘플은  $P_{4,-1}$  이다. 따라서, 모드 2 에 대한 HEVC 예측은  $T_{2,1} = P_{4,-1}$  일 것이다. 대신, 이 예에서,  $T_{2,1} = P_{3,0}$  이다.

[0162] 본 개시물의 일부 추가적인 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 블록 내 다른 샘플 값들의 예측을 수행하기 위해 비디오 데이터의 블록 내 하나 이상의 원래 샘플 값들을 사용하는 코딩을 수행할 수도 있다. 원래 샘플 값들은 블록의 최종 로우 및 최종 칼럼 (예컨대, 그 블록의 하부 로우 및 최우측 칼럼)에 대응할 수도 있다. 또 다른 예로서, 원래 샘플 값들은 블록의 제 1 로우 및 제 1 칼럼 (예컨대, 상부 로우 및 최좌측 칼럼)에 대응할 수도 있다. 도 4 는 다른 샘플 값들의 예측을 수행하는데 사용되는 샘플 값들의 일 예를 예시한다. 이 예에서, 비디오 코더는 무손실 코딩 모드를 수행할 수도 있고, 무손실 코딩 모드는 평면 코딩 모드, 각도 인트라 코딩 모드, 또는 또 다른 모드를 포함할 수도 있다. 일부 예들에서, 이 기법들은 잔여 값들의 회전된 세트에 대한 엔트로피 코딩이 뒤따르는, 예측에 의해 생성되는 잔여 값들의 세트에 대한 회전 동작을 더 포함할 수도 있다.

[0163] 차분 펠스 코드 변조 (DPCM) 의 방법들은, ITU-T SG 16 WP 3 과 ISO/IEC JTC 1/SC 29/WG 11 의 JCT-VC (Joint Collaborative Team on Video Coding), 12차 회의, 2013년 1월 14~23일, 스위스, 제네바, Lee 등, "AHG7: Residual DPCM for HEVC lossless coding", 문서 번호. JCTVC-L0117 (이하, "JCTVC-L0117") 에 제안되어 있다.

JCTVC-L0117 은 무손실 코딩을 위해 HEVC에서 수평 (예컨대, 인트라 예측 모드 10) 및 수직 (예컨대, 인트라 예측 모드 (26)) 모드들에 대해 인트라 예측에 대한 향상을 제안하였다. 이 향상은 잔여 DPCM 으로 나타내었다. JCTVC-L0117 에서, 잔여 DPCM 은 CU 가 무손실로 코딩되고 있을 때 적용된다. 잔여 DPCM 의 기본적인 아이디어는 수직 모드에 대해서 현재의 픽셀을 예측하기 위해 상부 로우 픽셀을 사용하는 것, 및 수직 모드에 대해서 현재의 픽셀을 예측하기 위해 좌측 칼럼 픽셀을 사용하는 것이다.

[0164] JCTVC-L0117 에서 설명된 바와 같은, 잔여 DPCM 은 다음과 같이 설명될 수 있다. 사이즈  $M \times N$  (로우들)  $\times$   $N$  (칼럼들) 의 블록을 고려한다.  $r_{i,j}$  를, HEVC 작업 초안 10 에 규정된 바와 같이 인트라 예측을 수행한 후 예측 잔여라 하고, 여기서  $0 \leq i \leq (M - 1)$  및  $0 \leq j \leq (N - 1)$  이다. 이것은 도 5a 및 도 5b 에 도시된다. 블록은 임의의 성분 (예컨대, 루마, 크로마, R, G, B, 등) 을 나타낼 수 있다. JCTVC-L0117 에

서 제안되는 방법에서, 잔여 DPCM 은 잔여 샘플들에 적용되어, 인트라 예측 모드가 수직일 때 엘리먼트들  $\tilde{r}_{i,j}$  을 가진 수정된  $M \times N$  어레이  $\tilde{R}$  가 다음과 같이 획득된다:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i = 0, 0 \leq j \leq (N - 1) \\ r_{i,j} - r_{(i-1),j}, & 1 \leq i \leq (M - 1), 0 \leq j \leq (N - 1) \end{cases}, \quad (23)$$

[0166] 인트라 예측 모드가 수평일 때,  $\tilde{r}_{i,j}$  는 다음과 같이 획득된다:

$$[0167] \quad \tilde{r}_{i,j} = \begin{cases} r_{i,j}, & 0 \leq i \leq (M-1), j=0 \\ r_{i,j} - r_{i,(j-1)} & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases} \quad (24)$$

[0168]  $\tilde{R}$  의 수정된 잔여 샘플들이 원래 잔여 샘플들  $R$  대신, 비디오 디코더로 시그널링된다. 이것은 수직 예측 모드에 대해  $P_{i,j}$ 에 대한 예측으로서  $P_{i-1,j}$ 를 사용하는 것, 및 수평 예측 모드에 대해  $P_{i,j}$ 에 대한 예측으로서  $P_{i,j-1}$ 을 사용하는 것과 동등할 수도 있다.

[0169] 더욱이, JCTVC-L0117에서 제안된 방법에서, 디코더 측 상에서, 인트라 예측 모드가 수직 모드일 때, 원래 잔여 샘플들은 수정된 잔여 샘플들이 다음과 같이 파싱된 후 재구성될 수 있다:

$$[0170] \quad r_{i,j} = \sum_{k=0}^i \tilde{r}_{k,j}, \quad 0 \leq i \leq (M-1), 0 \leq j \leq (N-1) \quad , \quad (25)$$

[0171] 인트라 예측 모드가 수평 모드일 때, 원래 잔여 샘플들은 수정된 잔여 샘플들이 다음과 같이 파싱된 후 재구성될 수 있다:

$$[0172] \quad r_{i,j} = \sum_{k=0}^j \tilde{r}_{i,k}, \quad 0 \leq i \leq (M-1), 0 \leq j \leq (N-1) \quad . \quad (26)$$

[0173] 더욱이, 본 개시물은 수평 및 수직 인트라 예측 모드들에 대해 JCTVC-L0117에서 제안된 잔여 DPCM 기법들을 무손실 코딩을 위해 다른 각도 인트라 예측 모드들로 확장하는 방법들을 설명한다. 코딩이 무손실이기 때문에, 대응하는 예측 잔여들 뿐만 아니라, (인과적 코딩 순서에서) 원래 이웃하는 샘플들도 (변환 및 양자화가 스kip되기 때문에) 예측에 사용 가능하다.

[0174] 제 1 예에서, 잔여 DPCM 기법은 다른 각도 인트라 예측 모드들로 확장될 수도 있다. 도 3은 상이한 각도 예측 모드들 (2 내지 34)에 대한 인트라 예측 방향들을 나타낸다. 이하, 22와 30 사이의 모드를 고려한다. 이들 모드들의 각각에 대해, 예측 방향은 수직 (근사-수직)에 가까운 것으로 간주될 수도 있다. 번호들 22 및 30은 단지 예들이다. 다른 범위들 (예컨대, 24와 28 사이의 인트라 예측 모드들)도 또한 선택될 수도 있다. 이하, 잔여  $r_{i,j}$ 는, HEVC (예컨대, HEVC 작업 초안 10)에 규정된 바와 같이 계산된다고 간주하고, 여기서,  $0 \leq i \leq (M-1)$  및  $0 \leq j \leq (N-1)$ 이다. 즉, 예측은 각도 모드에 따라서 수행되고, 잔여  $r_{i,j}$ 를 얻기 위해 원래 샘플 값들로부터 감산된다. 잔여 DPCM은 엘리먼트들  $\tilde{r}_{i,j}$ 을 가진

수정된  $M \times N$  어레이  $\tilde{R}$ 를 다음과 같이 획득하기 위해, 정확히 JCTVC-L0117에서와 같이 잔여 샘플들에 적용될 수도 있다:

$$[0175] \quad \tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ r_{i,j} - r_{(i-1),j}, & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases} \quad (27)$$

[0176] 이와 유사하게, 이 제 1 예에서, 6과 14 사이의 방향 인트라 예측 모드를 고려한다. 이들 인트라 예측 모드들의 각각에 있어, 예측 방향은 수평 (즉, 근사-수평)에 가까운 것으로 간주할 수 있다. 번호들 6 및 14는 단지 예들이다. 다른 예들에서, 다른 범위들 (예컨대, 8과 12 사이의 인트라 예측 모드들)이 사용될 수도 있다. 이하, 잔여  $r_{i,j}$ 는, HEVC (예컨대, HEVC 작업 초안 10)에 규정된 바와 같이 계산된다고 간주하고, 여기서  $0 \leq i \leq (M-1)$  및  $0 \leq j \leq (N-1)$ 이다. 즉, 예측은 각도 모드에 따라서 수행되고, 예측은 잔여  $r_{i,j}$ 를 결정하기 위해 원래 샘플 값들로부터 감산된다. 잔여 DPCM은 다음과 같이 엘리먼트들

$\tilde{r}_{i,j}$ 을 가진 수정된  $M \times N$  어레이  $\tilde{R}$ 를 획득하기 위해 정확히 JCTVC-L0117에서와 같이 잔여 샘플들에 적용된다:

$$[0177] \quad \tilde{r}_{i,j} = \begin{cases} r_{i,j}, & 0 \leq i \leq (M-1), j=0 \\ r_{i,j} - r_{i,(j-1)}, & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases} \quad (28)$$

[0178] JCTVC-L0117 의 경우에서와 같이,  $\tilde{R}$  의 수정된 잔여 샘플들이 디코더로 시그널링된다.

[0179] 더욱이, 이 제 1 예에서, 디코더 측에서, 원래 잔여 샘플들은 수정된 잔여 샘플들이 다음과 같이 파싱된 후 재구성될 수 있다. 인트라 예측 모드가 근사-수직일 때 (예컨대, 모드들 22 내지 30), 원래 잔여 샘플들은 다음과 같이 재구성될 수 있다:

$$[0180] \quad r_{i,j} = \sum_{k=0}^i \tilde{r}_{k,j}, \quad 0 \leq i \leq (M-1), 0 \leq j \leq (N-1) \quad , \quad (29)$$

[0181] 인트라 예측 모드가 근사-수평일 때 (예컨대, 모드들 6 내지 14), 원래 잔여 샘플들은 다음과 같이 재구성될 수 있다:

$$[0182] \quad r_{i,j} = \sum_{k=0}^j \tilde{r}_{i,k}, \quad 0 \leq i \leq (M-1), 0 \leq j \leq (N-1) \quad . \quad (30)$$

[0183] 일단 잔여  $r_{i,j}$  가 계산되었으면, 잔여  $r_{i,j}$  는 원래 샘플들을 획득하기 위해 각도 모드에 따라서 수행되는 예측에 가산된다. 수평 및 수직 모드들에 대한 예측을 획득할 때, (수직 모드에 대해) 제 1 칼럼 또는 (수평 모드에 대해) 제 1 로우에 대한 예측에의 그레디언트 항 (gradient term)의 추가를 인에이블하거나 또는 디스에 이를하는 것이 가능할 수도 있다.

[0184] 잔여 DPCM 기법들이 무손실 코딩을 위해 각도 인트라 예측 모드들로 확장되는 제 2 예에서, 인트라 예측 모드 (18), 또는 인트라 예측 모드 (18)의 대각선 우하방 예측 방향에 가까운 예측 방향들을 가진 인트라 예측 모드들에 대해 (도 3 참조), 잔여는 다음 방법으로 수정될 수도 있다. 첫째, 잔여  $r_{i,j}$  는, HEVC 작업 초안 10에 규정된 바와 같이 계산될 수도 있고, 여기서  $0 \leq i \leq (M-1)$  및  $0 \leq j \leq (N-1)$  이다. 즉, 예측이 각도 모드에 따라서 수행될 수도 있고, 예측은 잔여  $r_{i,j}$  를 얻기 위해 원래 샘플 값들로부터 감산될 수도 있다.

다. 그 후, 엘리먼트들  $\tilde{r}_{i,j}$  을 가진 수정된  $M \times N$  어레이  $\tilde{R}$  는 다음과 같이 결정될 수도 있다:

$$[0185] \quad \tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ r_{i,j}, & 1 \leq i \leq (M-1), j=0 \\ r_{i,j} - r_{i-1,j-1}, & 1 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases} \quad . \quad (31)$$

[0186] 이 제 2 예에서,  $\tilde{R}$  의 수정된 잔여 샘플들이 디코더 (예컨대, 비디오 디코더 (30))로 시그널링된다. 디코더 측에서, 비디오 디코더는 비디오 디코더가 수정된 잔여 샘플들을 다음과 같이 파싱한 후 원래 잔여 샘플들을 재구성할 수도 있다. 인트라 예측 방향이 대각선 우하방 예측 방향에 가까울 때, 비디오 디코더는 원래 잔여 샘플들을 다음과 같이 재구성할 수도 있다:

$$[0187] \quad r_{i,j} = \begin{cases} \tilde{r}_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ \tilde{r}_{i,j}, & 1 \leq i \leq (M-1), j=0 \\ \tilde{r}_{i,j} + r_{i-1,j-1}, & 1 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases} \quad . \quad (32)$$

[0188] 더욱이, 이 제 2 예에서, 일단 잔여  $r_{i,j}$  가 계산되었으면,  $r_{i,j}$  는 원래 샘플들을 획득하기 위해 각도 모드에 따라서 수행되는 예측에 가산될 수도 있다. 잔여  $r_{i,j}$  를 계산할 때에,  $r_{i-1,j-1}$  이 사용될 필요가 있을 수도 있다. 이것은 실제 (true) 잔여들이 로우 단위로 또는 칼럼 단위로 계산되면, 항상 참일 수도 있다. 따라

서, 로우 (또는, 칼럼) 에서의 모든 샘플들에 대한 실제 잔여  $r_{i,j}$  를 병렬로 계산하는 것이 가능할 수도 있다.

[0189] 잔여 DPCM 기법들이 무손실 코딩을 위해 각도 인트라 예측 모드들로 확장되는 제 3 예에서, 인트라 예측 모드 (34) 또는 인트라 예측 모드 (34) 의 대각선 좌하방 예측 방향에 가까운 예측 방향들을 가진 인트라 예측 모드들에 대해 (도 3 참조), 잔여는 다음 방법으로 수정될 수도 있다. 첫째, 잔여  $r_{i,j}$  는, HEVC (예컨대, HEVC 작업 초안 10) 에서 규정된 바와 같이 계산될 수도 있고, 여기서  $0 \leq i \leq (M - 1)$  및  $0 \leq j \leq (N - 1)$  이다. 즉, 예측이 각도 모드에 따라서 수행될 수도 있고, 예측은 잔여  $r_{i,j}$  를 결정하기 위해 원래 샘플 값

들로부터 감산될 수도 있다. 그 후, 엘리먼트들  $\tilde{r}_{i,j}$  을 가진 수정된  $M \times N$  어레이  $\tilde{R}$  는 다음과 같이 결정될 수도 있다:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ r_{i,j}, & 1 \leq i \leq (M-1), j=(N-1) \\ r_{i,j} - r_{i-1,j+1}, & 1 \leq i \leq (M-1), 0 \leq j \leq (N-2) \end{cases} . \quad (33)$$

[0190] 더욱이, 이 제 3 예에서,  $\tilde{R}$  의 수정된 잔여 샘플들이 비디오 디코더 (예컨대, 비디오 디코더 (30)) 로 시그널링될 수도 있다. 디코더 측에서, 비디오 디코더는 비디오 디코더가 수정된 잔여 샘플들을 다음과 같이 파싱한 후 원래 잔여 샘플들을 재구성할 수도 있다. 인트라 예측 방향이 대각선 좌하방 예측 방향에 가까울 때 (모드 (34)), 비디오 디코더는 다음 방정식을 사용하여 원래 잔여 샘플들을 재구성할 수도 있다:

$$r_{i,j} = \begin{cases} \tilde{r}_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ \tilde{r}_{i,j}, & 1 \leq i \leq (M-1), j=(N-1) \\ \tilde{r}_{i,j} + r_{i-1,j+1}, & 1 \leq i \leq (M-1), 0 \leq j \leq (N-2) \end{cases} . \quad (34)$$

[0191] [0193] 이 제 3 예에서, 일단 잔여  $r_{i,j}$  가 계산되었으면,  $r_{i,j}$  는 원래 샘플들을 획득하기 위해 각도 모드에 따라서 수행되는 예측에 가산될 수도 있다. 잔여  $r_{i,j}$  를 계산할 때에,  $r_{i-1,j+1}$  이 사용될 필요가 있을 수도 있다. 이것은 실제 잔여들이 로우 단위로 계산되면, 항상 참일 수도 있다. 따라서, 로우에서의 모든 샘플들에 대한 실제 잔여  $r_{i,j}$  를 병렬로 계산하는 것이 가능할 수도 있다.

[0194] 잔여 DPCM 기법들이 무손실 코딩을 위해 각도 인트라 예측 모드들로 확장되는 제 4 예에서, 인트라 예측 모드 2 또는 모드 2 의 대각선 우상방 예측 방향에 가까운 예측 방향들을 가진 인트라 예측 모드들에 대해 (도 3 참조), 잔여는 다음 방법으로 수정될 수도 있다. 첫째, 잔여  $r_{i,j}$  는, HEVC (예컨대, HEVC 작업 초안 10) 에서 규정된 바와 같이 계산될 수도 있고, 여기서  $0 \leq i \leq (M - 1)$  및  $0 \leq j \leq (N - 1)$  이다. 즉, 예측은 각도 모드에 따라서 수행될 수도 있고, 예측은 잔여  $r_{i,j}$  를 결정하기 위해 원래 샘플 값들로부터 감산될 수도

있다. 그 후, 엘리먼트들  $\tilde{r}_{i,j}$  을 가진 수정된  $M \times N$  어레이  $\tilde{R}$  는 다음과 같이 결정될 수도 있다:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i=(M-1), 0 \leq j \leq (N-1) \\ r_{i,j}, & 0 \leq i \leq (M-2), j=0 \\ r_{i,j} - r_{i+1,j-1}, & 0 \leq i \leq (M-2), 1 \leq j \leq (N-1) \end{cases} . \quad (35)$$

[0195] [0196] 이 제 4 예에서,  $\tilde{R}$  의 수정된 잔여 샘플들은 비디오 디코더 (예컨대, 비디오 디코더 (30)) 로 시그널링될 수도 있다. 디코더 측에서, 비디오 디코더는 비디오 디코더가 수정된 잔여 샘플들을 다음과 같이 파싱한 후 원래 잔여 샘플들을 재구성할 수도 있다. 인트라 예측 방향이 대각선 우상방 예측 방향에 가까울 때 (모드 2), 비디오 디코더는 다음 방정식에 따라서 원래 잔여 샘플들을 재구성할 수도 있다:

$$[0197] r_{i,j} = \begin{cases} \tilde{r}_{i,j}, & i=(M-1), 0 \leq j \leq (N-1) \\ \tilde{r}_{i,j}, & 0 \leq i \leq (M-2), j=0 \\ \tilde{r}_{i,j} + r_{i+1,j-1}, & 0 \leq i \leq (M-2), 1 \leq j \leq (N-1) \end{cases} . \quad (36)$$

[0198] 더욱이, 이 제 4 예에서, 일단 잔여  $r_{i,j}$  가 계산되었으면, 잔여  $r_{i,j}$  는 원래 샘플들을 획득하기 위해 각도 모드에 따라서 수행되는 예측에 가산될 수도 있다. 잔여  $r_{i,j}$  를 계산할 때에,  $r_{i+1,j-1}$  이 사용될 필요가 있을 수도 있다. 이것은 실제 잔여들이 칼럼 단위로 계산되면, 항상 참일 수도 있다. 따라서, 칼럼에서의 모든 샘플들에 대한 실제 잔여  $r_{i,j}$  를 별별로 계산하는 것이 가능할 수도 있다. 잔여 DPCM 기법들이 무순실 코딩을 위해 각도 인트라 예측 모드들로 확장되는 제 1, 제 2, 제 3, 및 제 4 예들은, 각각의 실시형태에 대한 예측 모드들의 범위가 중첩하지 않는다면, 동시에 채용될 수도 있다.

[0199] 잔여 DPCM 기법들이 무순실 코딩을 위해 각도 인트라 예측 모드들로 확장되는 제 5 예에서 설명된 바와 같이, JCTVC-L0117 에서 제안되는 잔여 DPCM 방법에서, 수직 모드에 대해, 제 1 로우에 대한 잔여는 수정되지 않는다.

유사한 관측 (observation) 이 수평 모드에 대해 참이다. 그 경우, 제 1 칼럼에 대한 잔여는 수정되지 않는다. 유사한 관측이 잔여 DPCM 기법들이 무순실 코딩을 위해 각도 인트라 예측 모드들로 확장되는 제 1 예의 경우 근사-수직 및 근사-수평 모드들에 대해 참이다. 잔여 DPCM 기법들이 무순실 코딩을 위해 각도 인트라 예측 모드들로 확장되는 제 5 예는 잔여 DPCM 의 개념을, 수직 또는 근사-수직 인트라 예측 모드에 대해 제 1 로우로, 그리고 수평 또는 근사-수평 인트라 예측 모드에 대해 제 1 칼럼으로 확장하는 것을 제안한다.

수직 또는 근사-수직 모드를 고려한다. 이 경우, 제 1, 잔여  $r_{i,j}$  는, HEVC (예컨대, HEVC 작업 초안 10) 에서 규정된 바와 같이 계산될 수도 있고, 여기서  $0 \leq i \leq (M-1)$  및  $0 \leq j \leq (N-1)$  이다. 즉, 예측은 각도 모드에 따라서 수행될 수도 있고, 예측은 잔여  $r_{i,j}$  를 결정하기 위해 원래 샘플 값들로부터 감산될

수도 있다. 그 후, 엘리먼트들  $\tilde{r}_{i,j}$  을 가진 수정된  $M \times N$  어레이  $\tilde{R}$  는 다음과 같이 결정될 수도 있다:

$$[0200] \tilde{r}_{i,j} = r_{i,j} - r_{(i-1),j}, \quad 0 \leq i \leq (M-1), 0 \leq j \leq (N-1) . \quad (37)$$

[0201] 방정식 (37) 에서,  $r_{(i-1),j}$  는 상부 블록으로부터의 잔여를 지칭한다. 상부 블록이 사용불가능하거나 또는 상부 블록이 상이한 LCU 에 속하면, 제 1 로우 상에서 잔여 DPCM 을 수행하지 않는 것이 불가능할 수도 있다.

[0202] 이 제 5 예에서,  $\tilde{R}$  의 수정된 잔여 샘플들이 비디오 디코더로 시그널링될 수도 있다. 디코더 측에서, 비디오 디코더는 비디오 디코더가 수정된 잔여 샘플들을 다음과 같이 파싱한 후 원래 잔여 샘플들을 재구성할 수도 있다. 인트라 예측 모드가 수직 또는 근사-수직일 때, 원래 잔여 샘플들은 다음과 같이 재구성될 수도 있다:

$$[0203] r_{i,j} = r_{(i-1),j} + \sum_{k=0}^i \tilde{r}_{k,j}, \quad 0 \leq i \leq (M-1), 0 \leq j \leq (N-1) . \quad (38)$$

[0204] 유사한 전략이 다음과 같이 수평 및 근사-수평 모드들에 대해 사용될 수도 있다:

$$[0205] \tilde{r}_{i,j} = r_{i,j} - r_{i,(j-1)}, \quad 0 \leq i \leq (M-1), 0 \leq j \leq (N-1) , \text{ 및 } (39)$$

$$[0206] r_{i,j} = r_{i,-1} + \sum_{k=0}^j \tilde{r}_{i,k}, \quad 0 \leq i \leq (M-1), 0 \leq j \leq (N-1) . \quad (40)$$

[0207] 이 접근법은 위에서 설명한 바와 같이 잔여 DPCM 기법들이 무순실 코딩을 위해 각도 인트라 예측 모드들로 확장되는 제 2, 제 3, 및 제 4 예들과 유사한 방법으로 확장될 수도 있다.

[0208] 잔여 DPCM 기법들이 무순실 코딩을 위해 각도 인트라 예측 모드들로 확장되는 제 6 예에서, JCTVC-L0117 에서 제안된 잔여 DPCM 방법은 또한 DC 인트라 모드 (예컨대, HEVC 에서의 모드 1) 에, 그리고 평면 모드 (예컨대, HEVC 에서의 모드 0) 에 적용될 수 있다. 예를 들어, 수직 (또는, 수평) 잔여 예측은 DC 인트라 예측이 이

루어진 후 적용될 수도 있다. 더욱이, 수직 및 수평 잔여 예측이 둘 다 적용될 수도 있다, 즉, 제 1 DCPM의 출력에 먼저 수직 (수평) DCPM 을 적용하고 그 후 수평 (수직) DCPM 을 적용한다.

[0209] 잔여 DPCM 기법들이 무손실 코딩을 위해 각도 인트라 예측 모드들로 확장되는 제 7 예는 2개의 이상의 DPCMs 가 잔여 상에 적용될 수 있다는 점에서, 잔여 DPCM 기법들이 무손실 코딩을 위해 각도 인트라 예측 모드들로 확장되는 제 6 예와 유사하다. 예를 들어, 2개의 대각선 DPCMs 는 잔여 DPCM 기법들이 평면 모드, 그리고 그 후 수평 및 수직 DPCM 에 대한 무손실 코딩을 위해 각도 인트라 예측 모드들로 확장되는 제 3 및 제 4 예들에서 설명하였다.

[0210] 본 개시물에서 설명되는 여러 기법들은 또한 변환이 스kip될 때 손실 인트라 코딩으로 확장될 수도 있다. 예를 들어, HEVC (예컨대, HEVC 작업 초안 10) 에서,  $4 \times 4$  블록들에 대해 변환이 스kip될 수도 있다. 본 개시물에서, 용어 "변환 스kip 블록"은 변환의 적용이 스kip되는 블록에 대해 적용한다. 그러나, 잔여 DPCM 기법들이 무손실 코딩을 위해 각도 인트라 예측 모드들로 확장되는 위에서 설명된 제안된 기법들은 변환이 스kip되면, 더 높은 블록 사이즈들에 대해 사용될 수도 있다. 예를 들어, JCTVC-E145 는 변환이 스kip될 때 이들 기법들을 손실 인트라 코딩으로 확장하는 한 방법을 설명하였다. 일반적으로, JCTVC-E145 에서 제안되는 아이디어는, 도 2 에 나타낸 바와 같이, 인트라 예측을 위해 참조 샘플들을 사용하는 대신, 인과적 이웃들로부터의 재구성된 샘플 값들이 인트라 예측을 수행하기 위해 사용될 수 있다는 것이다. 그러나, 이것은 예측 프로세스가 각각의 샘플에 대해 반복되어야 하기 때문에, 계산상 손해가 많을 수 있다. 더욱이, 어떻게 이것이 DC 및 평면 인트라 예측 모드들에 대해 작용하는지에 관해서는 불분명하다.

[0211] 본 개시물의 하나 이상의 기법들은 손실 코딩을 위한 잔여 DPCM 에 관한 것이다. 예를 들어, 본 개시물의 일부 예들에서, 변환이 스kip되는 블록에 대한 근사-수평 및 근사-수직 인트라 예측 모드들이 고려된다. 본 개시물에서, 근사-수직 모드는 예측 방향이 근사 수직인 인트라 예측 모드로서 정의될 수도 있다. 근사-수직 모드들의 예들은 도 3 에 나타낸 바와 같이 22 와 30 사이의 모든 인트라 예측 모드들일 수도 있다. 이와 유사하게, 근사-수평 모드는 예측 방향이 근사 수평인 인트라 예측 모드로서 정의될 수도 있다. 근사-수평 인트라 예측 모드들의 예들은 도 3 에 나타낸 바와 같이 6 과 14 사이의 모든 인트라 예측 모드들일 수도 있다. 더욱이, 사이즈 M (로우들)  $\times$  N (칼럼들) 의 블록을 고려한다.  $r_{i,j}$  를 HEVC (예컨대, HEVC 작업 초안 10) 에 규정된 바와 같이 인트라 예측을 수행한 후 예측 잔여라 하고, 여기서  $0 \leq i \leq (M - 1)$  및  $0 \leq j \leq (N - 1)$  이다. 이것은 도 5a 및 도 5b 에 도시된다. 블록은 임의의 성분 (예컨대, 루마, 크로마, R, G, B, 등) 을 나타낼 수도 있다.

[0212] 구체적으로 설명하면, 도 5a 는 근사-수직 모드들에 대한 잔여 DPCM 방향을 나타낸다. 도 5b 는 근사-수평 모드들에 대한 잔여 DPCM 방향을 나타낸다. 도 5a 및 5b 에서 각각의 정사각형 각각은 각각의 잔여 샘플  $r_{i,j}$  에 대응한다. 도 5a 의 각각의 칼럼에서 수직 화살표들은 근사-수직 모드들에 대한 잔여 DPCM 방향을 나타낸다. 도 5b 의 각각의 로우에서 수평 화살표들은 근사-수평 모드들에 대한 잔여 DPCM 방향을 나타낸다. DPCM 방향은 잔여 DPCM 을 적용할 때 비디오 코더가 샘플들을 프로세싱하는 것에 따르는 방향 (예컨대, 수평 또는 수직) 이다.

[0213] 더욱이, 근사-수직 모드를 고려하고,  $Q(r_{i,j})$  가 잔여  $r_{i,j}$  의 양자화된 버전을 나타낸다고 하고, 여기서  $0 \leq i \leq (M - 1)$  및  $0 \leq j \leq (N - 1)$  이다. 잔여 값들의 블록은 R 로서 나타낼 수도 있다. 다시 말해서, 잔여  $r_{i,j}$  는 양자화 및 역양자화를 겪는다. 잔여 DPCM 은 다음과 같이  $M \times N$  어레이  $\tilde{R}$  를 획득하기 위해

잔여 샘플들에 적용될 수도 있다. 엘리먼트들  $\tilde{r}_{i,j}$  을 가진 수정된  $M \times N$  어레이  $\tilde{R}$  는 인트라 예측 모드가 수직일 때 다음과 같이 획득될 수도 있다:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i = 0, 0 \leq j \leq (N - 1) \\ r_{i,j} - Q(r_{(i-1),j}), & 1 \leq i \leq (M - 1), 0 \leq j \leq (N - 1) \end{cases}, \quad (41)$$

[0215] 인트라 예측 모드가 수평일 때,  $\tilde{r}_{i,j}$  는 다음과 같이 획득될 수도 있다:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & 0 \leq i \leq (M-1), j=0 \\ r_{i,j} - Q(r_{i,(j-1)}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases} . \quad (42)$$

[0217] 따라서, 상기 예에서, 인트라 예측 모드가 수직 또는 근사-수직일 때,  $\tilde{R}$  의 최좌측 칼럼에서의 수정된 잔여 값들은 R 의 최좌측 칼럼에서의 대응하는 잔여 값들과 동일하다. R 의 최좌측 칼럼의 우측에 있는 R 의 각각의 칼럼에서 각각의 잔여 값 (즉,  $r_{i,j}$ ) 각각에 대해, 비디오 인코더 (20) 는 대응하는 수정된 잔여 값  $\tilde{r}_{i,j}$  을, 각각의 잔여 값 (즉,  $r_{i,j}$ ), 마이너스, 각각의 잔여 값의 바로 좌측에서 발생하는 잔여 값에 대응하는 재구성된 잔여 값 (즉,  $Q(r_{(i-1),j})$ ) 과 동일하게 설정할 수도 있다. 이와 유사하게, 인트라 예측 모드가 수평 또는 근사-수평일 때,  $\tilde{R}$  의 상부 로우에서의 수정된 잔여 값들은 R 의 상부 로우에서의 대응하는 잔여 값들과 동일하다. 그러나, R 의 상부 로우의 아래에 있는 R 의 각각의 로우에서 각각의 잔여 값 (즉,  $r_{i,j}$ ) 각각에 대해, 비디오 인코더 (20) 는 대응하는 수정된 잔여 값 (즉,  $\tilde{r}_{i,j}$ ) 을, 각각의 잔여 값 (즉,  $r_{i,j}$ ), 마이너스, 각각의 잔여 값의 바로 상부에서 발생하는 잔여 값에 대응하는 재구성된 잔여 값 (즉,  $Q(r_{i,(j-1)})$ ) 과 동일하게 설정할 수도 있다.

[0218] 더욱이, 이 예에서, 수정된 잔여 샘플,  $\tilde{r}_{i,j}$  는 수정된 잔여 샘플의 양자화된 버전,  $Q(\tilde{r}_{i,j})$  을 생성하기 위해 양자화된다. 인트라 예측 모드가 수직일 때, 재구성된 잔여 샘플  $Q(r_{i,j})$  는 다음과 같이 계산된다:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & i=0, 0 \leq j \leq (N-1), \\ Q(\tilde{r}_{i,j}) + Q(r_{i-1,j}) & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases} . \quad (43)$$

[0220] 이것은 또한 다음과 같이 쓸 수도 있다:

$$Q(r_{i,j}) = \sum_{k=0}^i Q(\tilde{r}_{k,j}), \quad 0 \leq i \leq (M-1), 0 \leq j \leq (N-1)$$

[0222] 이와 유사하게, 모드가 수평일 때, 재구성된 잔여 샘플  $Q(r_{i,j})$  는 다음과 같이 계산된다:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & 0 \leq i \leq (M-1), j=0, \\ Q(\tilde{r}_{i,j}) + Q(r_{i,j-1}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases} . \quad (44)$$

[0224] 이것은 또한 다음과 같이 쓸 수 있다:

$$Q(r_{i,j}) = \sum_{k=0}^j Q(\tilde{r}_{i,k}), \quad 0 \leq i \leq (M-1), 0 \leq j \leq (N-1)$$

[0226] 따라서, 인트라 예측 모드가 수직 또는 근사-수직일 때, R 의 최좌측 칼럼에서의 잔여 값들에 대응하는 재구성된 잔여 값들은  $\tilde{R}$  의 최좌측 칼럼에서의 수정된 잔여 값들의 대응하는 양자화된 버전들과 동일하다. 그러나, R 의 최좌측 칼럼의 우측에 있는 R 의 각각의 칼럼에서 각각의 잔여 값 각각에 대해, 비디오 인코더 (20) 는 대응하는 재구성된 잔여 값을,  $\tilde{R}$  에서의 대응하는 수정된 잔여 값의 양자화된 버전 (즉,  $Q(\tilde{r}_{i,j})$ ) 과, 각각의 잔여 값의 바로 좌측에서 발생하는 잔여 값에 대응하는 재구성된 잔여 값 (즉,  $Q(r_{i-1,j})$ ) 의 합계와 동일하게 설정한다. 이와 유사하게, 인트라 예측 모드가 수평 또는 근사 수평일 때, R 의 상부 로우에서의 잔여 값들에 대응하는 재구성된 잔여 값들은  $\tilde{R}$  의 상부 로우에서의 수정된 잔여 값들의 대응하는 양자화된 버전들과 동일하다. 그러나, R 의 상부 로우의 아래에 있는 R 의 각각의 로우에서 R 의 각각의 잔여 값 각각에 대

해, 비디오 인코더 (20) 는 대응하는 재구성된 잔여 값을,  $\tilde{R}$  에서의 대응하는 수정된 잔여 값의 양자화된 버전 (즉,  $Q(\tilde{r}_{i,j})$ ) 과 각각의 잔여 값의 바로 상부에서 발생하는 R에서의 잔여 값에 대응하는 재구성된 잔여 값 (즉,  $Q(r_{i,j-1})$ ) 의 합계와 동일하게 설정할 수도 있다.

[0227] 디코더 측 상에서, 계산들 (43), 및 (44) 는 재구성된 잔여 샘플  $Q(r_{i,j})$  를 생성하기 위해 반복되고, 여기서  $0 \leq i \leq (M-1)$  및  $0 \leq j \leq (N-1)$  이다. 비디오 디코더 (30) 는 재구성된 잔여 샘플 값들을 원래 예측 값들에 가산하여, 재구성된 샘플 값들을 생성할 수도 있다. 예를 들어, 비디오 디코더 (30) 는 재구성된 잔여 샘플들  $Q(r_{i,j})$  를 예측 블록의 대응하는 샘플들에 가산하여, 현재 블록 (예컨대, 현재의 CU) 의 샘플 값들을 재구성할 수도 있다. DPCM 을 사용하여 인코딩되는 잔여 값들로부터 그 재구성된 잔여 값들을 결정하는 프로세스는 본원에서, "역 RDPCM" 으로서 지칭될 수도 있다. 보통, 비디오 인코더 (20) 및 비디오 디코더 (30) 양쪽은 재구성된 샘플들을 적합한 비트심도로 클리핑한다. 그러나, 이 경우, 클리핑 동작은 양자화된 잔여들을 재구성하기 위해 역 RDPCM 프로세스 이후에 수행된다.

[0228] 상기 예들은 대각선 방향들로도 또한 직접적인 방법 (straightforward manner) 으로 확장될 수도 있다. 예를 들어, 일 예에서, 인트라 예측 방향이 대각선 우하방 방향에 가까울 때 (예컨대, 인트라 예측 모드 (18)),

$\tilde{r}_{i,j}$  는 다음과 같이 결정될 수도 있다:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ r_{i,j}, & 1 \leq i \leq (M-1), j=0 \\ r_{i,j} - Q(r_{i-1,j-1}), & 1 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases} \quad (45)$$

[0229] [0230] 재구성된 잔여 샘플  $Q(r_{i,j})$  는 다음과 같이 결정될 수도 있다:

$$Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & i=0, 0 \leq j \leq (N-1) \\ Q(\tilde{r}_{i,j}), & 1 \leq i \leq (M-1), j=0 \\ Q(\tilde{r}_{i,j}) + Q(r_{i-1,j-1}), & 1 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases} \quad (46)$$

[0231] [0232] 다른 근사-대각선 모드들 (예컨대, 모드를 근사 좌하방 (모드 34) 및 근사 우상방 (모드 2)) 는 유사하게 취급될 수 있다.

[0233] 더욱이, 일부 예들에서, DPCM 을 재구성된 잔여들에 적용하는 대신, DPCM 은 재구성된 잔여들의 비트-시프트된 버전들에 적용될 수도 있다. 예를 들어, 변환 스윕 블록들에 대해, 비디오 디코더 (30) 는 각각의 샘플에 대해, 역양자화 동작, 7 만큼 좌측 시프트, 그 후 (예컨대, 오프셋을 가산한 후) (20 - 비트 심도) 만큼 우측 시프트를 수행할 수도 있다. 이 맵락에서, 비트 심도는 샘플 값 또는 변환 계수를 나타내기 위해 사용되는 비트수를 지칭할 수도 있다. 역양자화 동작에서, 오프셋을 가산한 후 bdShift 만큼 우측 시프트가 존재한다.

[0234] 예를 들어, JCTVC-M1005\_v2 의 섹션 8.6.2 (즉, HEVC 에 대한 범위 확장 사양의 초안) 은 TU 의 변환 블록의 변환 계수 레벨들의 어레이에 기초하여 잔여 샘플들의 어레이를 결정하는 스케일링 및 변환 프로세스를 기술한다. JCTVC-M1005\_v2 의 섹션 8.6.2 에서, 변환 계수 레벨들의 어레이는 변환 및 양자화가 변환 블록에 적용되지 않으면, 잔여 샘플 값들을 포함할 수도 있다. JCTVC-M1005\_v2 의 섹션 8.6.2 의 일부가 아래에 재현된다.

[0235] 잔여 샘플들 r 의  $(nTbS)x(nTbS)$  어레이는 다음과 같이 유도된다:

[0236] - cu\_transquant\_bypass\_flag 가 1 과 동일하면,  $(nTbS)x(nTbS)$  어레이 r 은 변환 계수들 TransCoeffLevel[ $xTbY$  ][ $yTbY$  ][ $cIdx$ ] 의  $(nTbS)x(nTbS)$  어레이와 동일하게 설정된다.

[0237] - 그렇지 않으면, 다음 순서화된 단계들이 적용된다:

[0238] 1. 하위 조항 8.6.3 에 규정된 바와 같은 변환 계수들에 대한 스케일링 프로세스가 변환 블록 로케이션 (

$xTbY$ ,  $yTbY$  ), 변환 블록의 사이즈  $nTbS$ , 칼라 성분 변수  $cIdx$ , 및 양자화 파라미터  $qP$  와 함께 입력들로서 호출되고, 출력은 스케일링된 변환 계수들  $d$  의  $(nTbS)x(nTbS)$  어레이이다.

[0239] 2. 잔여 샘플들  $r$  의  $(nTbS)x(nTbS)$  어레이는 다음과 같이 유도된다:

[0240] -  $\text{transform\_skip\_flag}[xTbY][yTbY][cIdx]$  가 1 과 동일하면, 잔여 샘플 어레이 값들  $r[x][y]$  은 다음과 같이 유도되고, 여기서,  $x = 0..nTbS - 1$ ,  $y = 0..nTbS - 1$  이다:

$$r[x][y] = (d[x][y] \ll 7) \quad (8-267)$$

[0242] - 그렇지 않으면 ( $\text{transform\_skip\_flag}[xTbY][yTbY][cIdx]$  가 0 과 동일하면), 하위 조항 8.6.4 에 규정된 바와 같은 스케일링된 변환 계수들에 대한 변환 프로세스가 변환 블록 로케이션 ( $xTbY$ ,  $yTbY$  ), 변환 블록의 사이즈  $nTbS$ , 칼라 성분 변수  $cIdx$ , 및 스케일링된 변환 계수들  $d$  의  $(nTbS)x(nTbS)$  어레이와 함께 입력들로서 호출되고, 출력은 잔여 샘플들  $r$  의  $(nTbS)x(nTbS)$  어레이이다.

[0243] 3. 변수  $bdShift$  는 다음과 같이 유도된다:

$$bdShift = (cIdx == 0) ? 20 - BitDepth_Y : 20 - BitDepth_C$$

$$(8-268)$$

[0246] 4. 잔여 샘플 값들  $r[x][y]$  은 다음과 같이 수정되고, 여기서,  $x = 0..nTbS - 1$ ,  $y = 0..nTbS - 1$  이다:

$$r[x][y] = (r[x][y] + (1 \ll (bdShift - 1))) \gg bdShift$$

$$(8-269)$$

[0249] 위에 나타낸 바와 같이, JCTVC-M1005\_v2 의 섹션 8.6.2 는, 변환 및 양자화가 변환 블록에 적용되면, JCTVC-M1005\_v2 의 섹션 8.6.3 이 호출된다고 규정한다. JCTVC-M1005\_v2 의 섹션 8.6.3 은 변환 계수들에 대한 스케일링 프로세스를 기술한다. JCTVC-M1005\_v2 의 섹션 8.6.3 이 아래에 재현된다.

[0250] 이 프로세스에의 입력들은 다음과 같다:

[0251] - 현재의 화상의 상부 좌측 루마 샘플에 대한, 현재의 루마 변환 블록의 좌상부 샘플을 규정하는 루마 로케이션 ( $xTbY$ ,  $yTbY$  ),

[0252] - 현재의 변환 블록의 사이즈를 규정하는 변수  $nTbS$ ,

[0253] - 현재 블록의 칼라 성분을 규정하는 변수  $cIdx$ ,

[0254] - 양자화 파라미터를 규정하는 변수  $qP$ .

[0255] 이 프로세스의 출력은 엘리먼트들  $d[x][y]$  를 가진 스케일링된 변환 계수들의  $(nTbS)x(nTbS)$  어레이  $d$  이다.

[0256] 변수  $bdShift$  는 다음과 같이 유도된다:

[0257] -  $cIdx$  가 0 과 동일하면,

$$bdShift = BitDepth_Y + \log_2(nTbS) - 5 \quad (8-270)$$

[0259] - 그렇지 않으면,

$$bdShift = BitDepth_C + \log_2(nTbS) - 5 \quad (8-271)$$

[0261] 리스트  $\text{levelScale}[ ]$  는  $\text{levelScale}[k] = \{40, 45, 51, 57, 64, 72\}$  로서 규정되고, 여기서,  $k = 0..5$  이다.

[0262] 스케일링된 변환 계수들  $d[x][y]$  의 유도를 위해, 다음이 적용되고, 여기서  $x = 0..nTbS - 1$ ,  $y = 0..nTbS - 1$  이다:

[0263] - 스케일링 인자  $m[x][y]$  는 다음과 같이 유도된다:

[0264] -  $\text{scaling\_list\_enabled\_flag}$  가 0 과 동일하면,

[0265]  $m[x][y] = 16$  (8-272)

[0266] - 그렇지 않으면 (scaling\_list\_enabled\_flag 가 1 과 동일하면),

[0267]  $m[x][y] = ScalingFactor[sizeId][matrixId][x][y]$  (8-273)

[0268] 여기서, sizeId 는  $(nTbS)x(nTbS)$  와 동일한 양자화 매트릭스의 사이즈에 대해 테이블 7-3 에 규정되어 있고, matrixId 는 sizeId, CuPredMode[ $xTbY$ ][ $yTbY$ ], 및 cIdx 에 대해, 테이블 7-4 에 각각 규정되어 있다.

[0269] - 스케일링된 변환 계수  $d[x][y]$  는 다음과 같이 유도된다:

$$\begin{aligned} d[x][y] &= \\ &\text{Clip3}(-32768, 32767, ((\text{TransCoeffLevel}[xTbY][yTbY][cIdx][x][y] *} \\ &m[x][y]) * \\ &\text{levelScale[qP%6] << (qP / 6) + (1 << (bdShift - 1))) >> bdShift) \end{aligned}$$

(8-274)

[0270] JCTVC-M1005\_v2 의 섹션 8.6.3 에 나타낸 바와 같이, 비디오 디코더 (30) 는 JCTVC-M1005\_v2 의 방정식 8-274 를 사용하여, 스케일링된 변환 계수,  $d[x][y]$  를 결정할 수도 있다. 방정식 8-274 에서, Clip3(...) 함수는 다음과 같이 정의되며:

$$\text{Clip3}(x, y, z) = \begin{cases} x &; z < x \\ y &; z > y \\ z &; 그 외 \end{cases}$$

[0272]

[0273] 본 개시물의 하나 이상의 기법들에 따르면, 역 DPCM 을 재구성된 잔여들에 적용하는 대신, 역 DPCM 이 좌측-시프트를 적용하기 전에 역양자화된 샘플 값들에 적용될 수도 있다. 이와 유사하게, 역변환 대신, 7 만큼의 좌측 시프트가 역양자화 이후에 적용될 수도 있다. 예를 들어, 잔여 샘플들의 어레이  $r[x][y]$  는 JCTVC-M1005\_v2 의 방정식 8-267 에 나타낸 바와 같이 유도될 수도 있고, 여기서,  $x = 0..nTbS-1$ ,  $y = 0..nTbS-1$  이다.

[0274] 일부 예들에서, 역 DPCM 은 역양자화 이후 또는 7 만큼 좌측 시프트 이후 샘플들에 적용된다. 이와 유사하게, 오프셋을 추가한 후 (20 - 비트 심도) 만큼의 우측 시프트가 잔여들을 재구성하기 위해 프로세스의 가장 끝에 제공된다. 예를 들어, 잔여 샘플들  $r[x][y]$  는, JCTVC-M1005\_v2 의 방정식 8-269 에 나타낸 바와 같이 수정될 수도 있고, 여기서,  $x = 0..nTbS-1$  및  $y = 0..nTbS-1$  이다.

[0275] 그 경우, 역 DPCM 은, 전체 우측 시프트가 (20 - 비트 심도) 에 달하도록, 나머지 우측 시프트를 적용하기 전에, 더 나은 정밀도를 유지하기 위해 (20 - 비트 심도) 보다 적은 양 만큼 우측 시프트한 후, 적용될 수도 있다.

[0276] 일부의 경우, 비디오 인코더 (20) 는 변환 블록의 잔여 샘플들에 변환을 적용하지 않지만, 변환 블록의 잔여 샘플들을 양자화한다. 다시 말해서, 비디오 인코더 (20) 는 변환이 스킵되는 손실 코딩의 유형을 적용할 수도 있다. 이러한 경우들에서, 비디오 디코더 (30) 는 비트스트림에서의 신택스 엘리먼트들로부터, 변환 블록의 양자화된 잔여 샘플들을 결정하고, 변환 블록의 양자화된 잔여 샘플들을 역양자화하여, 변환 블록에 역변환을 적용함이 없이 변환 블록의 잔여 샘플들을 재구성할 수도 있다.

[0277] 대안적으로, 그레디언트 필터링은 원래 샘플 값들 대신, 재구성된 (양자화된) 샘플 값을 사용하여, 본원에서 설명된 바와 같이, 수정될 수도 있다. HEVC (예컨대, HEVC 작업 초안 10) 에서, 수평 및 수직 모드들에 대한 예측을 획득할 때, 그레디언트 항은 (수직 모드에 대해) 제 1 칼럼 또는 (수평 모드에 대해) 제 1 로우에 대한 예측에 가산된다. 이것은 그레디언트 필터링으로서 지정될 수도 있다. 그레디언트 항의 추가는 잔여 DPCM 을 사용할 때 인에이블되거나 또는 디스에이블될 수 있다. 본 개시물은 잔여 예측이 사용될 때 수평 및 수직 인트라 예측 모드들에 대해 그레디언트 필터링에 대한 향상들을 제안한다.

[0278] 예를 들어, JCTVC-L0117 에서 제안되는 바와 같은 잔여 DPCM 이 사용되고 있다고 간주한다. HEVC 작업 초안

10 에서, 루마 성분들에 대해 그리고 32 미만인 블록 사이즈들에 대해, 수직 인트라 예측 모드에 대해, 샘플  $P_{i,0}$ ,  $0 \leq i \leq M-1$  에 대한 예측은 다음과 같이 주어진다:

$$\text{Clip}\left(P_{-1,0} + ((P_{i,-1} - P_{-1,-1}) \gg 1)\right)$$

[0279] [0280] 상기 방정식에서,  $\gg$  는 비트 단위 우측-시프트를 나타내고, 클립 동작은 예측 값들을 샘플 값들의 범위로 클리핑한다. 잔여 DPCM에서, 수직 인트라 예측 모드에 대해, 샘플  $P_{i,j}$ 에 대한 예측이  $P_{i-1,j}$  이기 때문에, 제 1 칼럼에서의 샘플들에 대한 그레디언트 항은  $((P_{i,-1} - P_{i-1,-1}) \gg 1)$  과 같이 수정된다. 따라서, 샘플  $P_{i,0}$ ,  $0 \leq i \leq M-1$ 에 대한 예측은 다음과 같다:

$$\text{Clip}\left(P_{i-1,0} + ((P_{i,-1} - P_{i-1,-1}) \gg 1)\right)$$

[0281] [0282] 더욱이, 그레디언트 필터링은 수직 인트라 예측 모드에 대해 다른 칼럼들로 다음과 같이 확장될 수도 있다. 좌측 및 좌상부 원래 샘플들이 무손실 모드에서 사용가능하기 때문에, 그레디언트 항은 수직 인트라 예측 모드에서 임의의 칼럼에 대한 예측에 가산될 수도 있다. 따라서, 수직 인트라 예측 모드에 대해, 샘플  $P_{i,j}$ ,  $0 \leq i \leq M-1$ ,  $0 \leq j \leq N-1$ 에 대한 예측은 다음으로 수정될 수도 있다:

$$\text{Clip}\left(P_{i-1,j} + ((P_{i,j-1} - P_{i-1,j-1}) \gg 1)\right)$$

[0283] [0284] 수직 인트라 예측 모드에 대한 수정된 그레디언트 필터링이 임의의 성분 및 임의의 블록 사이즈에 적용될 수도 있다.

[0285] 이 개념은 수평 인트라 예측 모드로도 또한 확장될 수도 있다. HEVC 작업 초안 10 에서, 루마 성분들에 대해 그리고 32 미만인 블록 사이즈들에 대해, 수평 인트라 예측 모드에 대해, 샘플  $P_{0,j}$ ,  $0 \leq j \leq N-1$ 에 대한 예측은 다음과 주어진다:

$$\text{Clip}\left(P_{0,-1} + ((P_{-1,j} - P_{-1,-1}) \gg 1)\right)$$

[0286] [0287] 상기 방정식에서,  $\gg$  는 비트 단위 우측-시프트를 나타내고, 클립 동작은 예측 값들을 샘플 값들의 범위로 클리핑한다. 잔여 DPCM에서, 수평 인트라 예측 모드에 대해, 샘플  $P_{i,j}$ 에 대한 예측이  $P_{i,j-1}$  이기 때문에, 제 1 로우에서의 샘플들에 대한 그레디언트 항은  $((P_{-1,j} - P_{-1,j-1}) \gg 1)$ 로 수정될 수도 있다. 샘플  $P_{0,j}$ ,  $0 \leq j \leq N-1$ 에 대한 예측은 다음에 의해 결정될 수도 있다:

$$\text{Clip}\left(P_{0,j-1} + ((P_{-1,j} - P_{-1,j-1}) \gg 1)\right)$$

[0288] [0289] 그레디언트 필터링은 수평 인트라 예측 모드에 대해 다른 칼럼들로 다음과 같이 확장될 수도 있다. 상부 및 좌상부 원래 샘플들이 무손실 모드에서 항상 사용가능하기 때문에, 그레디언트 항은 수평 인트라 예측 모드에서 임의의 로우에 대한 예측에 가산될 수도 있다. 따라서, 수평 인트라 예측 모드에 대해, 샘플  $P_{i,j}$ ,  $0 \leq i \leq M-1$ ,  $0 \leq j \leq N-1$ 에 대한 예측은 다음과 주어질 수도 있다:

$$\text{Clip}\left(P_{i,j-1} + ((P_{i-1,j} - P_{i-1,j-1}) \gg 1)\right)$$

[0290] [0291] JCTVC-M1005\_v2 의 섹션 7.3.2.3에서 설명된 바와 같이, PPS는 부호 데이터 은닉 인에이블 선택스 엘리먼트(예컨대, sign\_data\_hiding\_enabled\_flag)를 포함할 수도 있다. 더욱이, JCTVC-M1005\_v2의 섹션 7.3.8.11에 나타낸 바와 같이, 부호 데이터 은닉 인에이블 선택스 엘리먼트가 특정의 값(예컨대, 1)을 가질 때, 변환 계수들의 양/음의 부호들을 나타내는 선택스 엘리먼트들(예컨대, coeff\_sign\_flag 선택스 엘리먼트들)은 TU로부터 생략될 수도 있다. 따라서, 비디오 인코더(20)는 비트스트림으로, 부호 데이터 은닉이 현재의 화상, 따라서 현재의 화상 내 현재 블록에 대해 인에이블된다는 명시적 표시(예컨대,

`sign_data_hiding_enabled_flag`) 를 시그널링할 수도 있다. 이와 유사하게, 비디오 디코더 (30) 는 비트 스트림으로부터, 부호 데이터 은닉이 현재의 화상, 따라서 현재의 화상에서의 현재 블록에 대해 인에이블된다는 것을 나타내는 명시적 표시 (예컨대, `sign_data_hiding_enabled_flag`) 를 획득할 수도 있다. JCTVC-M1005\_v2 의 섹션 7.3.8.11 ① 아래에 재현된다.

residual_coding( x0, y0, log2TrafoSize, cIdx ) {	디스크립터
if( transform_skip_enabled_flag && !cu_transquant_bypass_flag && ( log2TrafoSize == 2 ) )	
<b>transform_skip_flag</b> [ x0 ][ y0 ][ cIdx ]	ae(v)
<b>last_sig_coeff_x_prefix</b>	ae(v)
<b>last_sig_coeff_y_prefix</b>	ae(v)
if( last_sig_coeff_x_prefix > 3 )	
<b>last_sig_coeff_x_suffix</b>	ae(v)
if( last_sig_coeff_y_prefix > 3 )	
<b>last_sig_coeff_y_suffix</b>	ae(v)
lastScanPos = 16	
lastSubBlock = ( 1 << ( log2TrafoSize - 2 ) ) * ( 1 << ( log2TrafoSize - 2 ) ) - 1	
do {	
if( lastScanPos == 0 ) {	
lastScanPos = 16	
lastSubBlock--	
}	
lastScanPos--	
xS = ScanOrder[ log2TrafoSize - 2 ][ scanIdx ][ lastSubBlock ][ 0 ]	
yS = ScanOrder[ log2TrafoSize - 2 ][ scanIdx ][ lastSubBlock ][ 1 ]	
xC = ( xS << 2 ) + ScanOrder[ 2 ][ scanIdx ][ lastScanPos ][ 0 ]	
yC = ( yS << 2 ) + ScanOrder[ 2 ][ scanIdx ][ lastScanPos ][ 1 ]	
} while( ( xC != LastSignificantCoeffX )    ( yC != LastSignificantCoeffY ) )	
for( i = lastSubBlock; i >= 0; i-- ) {	
xS = ScanOrder[ log2TrafoSize - 2 ][ scanIdx ][ i ][ 0 ]	
yS = ScanOrder[ log2TrafoSize - 2 ][ scanIdx ][ i ][ 1 ]	
inferSbDcSigCoeffFlag = 0	
if( ( i < lastSubBlock ) && ( i > 0 ) ) {	
<b>coded_sub_block_flag</b> [ xS ][ yS ]	ae(v)
inferSbDcSigCoeffFlag = 1	
}	
for( n = ( i == lastSubBlock ) ? lastScanPos - 1 : 15; n >= 0; n-- ) {	
xC = ( xS << 2 ) + ScanOrder[ 2 ][ scanIdx ][ n ][ 0 ]	
yC = ( yS << 2 ) + ScanOrder[ 2 ][ scanIdx ][ n ][ 1 ]	
if( coded_sub_block_flag[ xS ][ yS ] && ( n > 0    !inferSbDcSigCoeffFlag ) ) {	
<b>sig_coeff_flag</b> [ xC ][ yC ]	ae(v)
if( sig_coeff_flag[ xC ][ yC ] )	
inferSbDcSigCoeffFlag = 0	

}	
}	
firstSigScanPos = 16	
lastSigScanPos = -1	
numGreater1Flag = 0	
lastGreater1ScanPos = -1	
for( n = 15; n >= 0; n-- ) {	
xC = ( xS << 2 ) + ScanOrder[ 2 ][ scanIdx ][ n ][ 0 ]	
yC = ( yS << 2 ) + ScanOrder[ 2 ][ scanIdx ][ n ][ 1 ]	
if( sig_coeff_flag[ xC ][ yC ] ) {	
if( numGreater1Flag < 8 ) {	
coeff_abs_level_greater1_flag[ n ]	ae(v)
numGreater1Flag++	
if( coeff_abs_level_greater1_flag[ n ] && lastGreater1ScanPos == -1 )	
lastGreater1ScanPos = n	
}	
if( lastSigScanPos == -1 )	
lastSigScanPos = n	
firstSigScanPos = n	
}	
}	
signHidden = ( lastSigScanPos - firstSigScanPos > 3 && !cu_transquant_bypass_flag )	
if( lastGreater1ScanPos != -1 )	
coeff_abs_level_greater2_flag[ lastGreater1ScanPos ]	ae(v)
for( n = 15; n >= 0; n-- ) {	
xC = ( xS << 2 ) + ScanOrder[ 2 ][ scanIdx ][ n ][ 0 ]	
yC = ( yS << 2 ) + ScanOrder[ 2 ][ scanIdx ][ n ][ 1 ]	
if( sig_coeff_flag[ xC ][ yC ] && (!sign_data_hiding_enabled_flag    !signHidden    ( n != firstSigScanPos )) )	
coeff_sign_flag[ n ]	ae(v)
}	
numSigCoeff = 0	
sumAbsLevel = 0	
for( n = 15; n >= 0; n-- ) {	
xC = ( xS << 2 ) + ScanOrder[ 2 ][ scanIdx ][ n ][ 0 ]	
yC = ( yS << 2 ) + ScanOrder[ 2 ][ scanIdx ][ n ][ 1 ]	
if( sig_coeff_flag[ xC ][ yC ] ) {	

baseLevel = 1 + coeff_abs_level_greater1_flag[ n ] + coeff_abs_level_greater2_flag[ n ]	
if( baseLevel == ( ( numSigCoeff < 8 ) ? ( ( n == lastGreater1ScanPos ) ? 3 : 2 ) : 1 ) )	
coeff_abs_level_remaining[ n ]	ac(v)
TransCoeffLevel[ x0 ][ y0 ][ cIdx ][ xC ][ yC ] = ( coeff_abs_level_remaining[ n ] + baseLevel ) * ( 1 - 2 * coeff_sign_flag[ n ] )	
if( sign_data_hiding_enabled_flag && signHidden ) {	
sumAbsLevel += ( coeff_abs_level_remaining[ n ] + baseLevel )	
if( ( n == firstSigScanPos ) && ( ( sumAbsLevel % 2 ) == 1 ) )	
TransCoeffLevel[ x0 ][ y0 ][ cIdx ][ xC ][ yC ] = -TransCoeffLevel[ x0 ][ y0 ][ cIdx ][ xC ][ yC ]	
}	
numSigCoeff++	
}	
}	
}	

[0294]

JCTVC-M1005\_v2 의 섹션 7.4.3.3 에 규정된 바와 같이, 0 과 동일한 sign\_data\_hiding\_enabled\_flag 는 부호 비트 은닉 (즉, 부호 데이터 은닉) 이 디스에이블된다는 것을 규정한다. 1 과 동일한 sign\_data\_hiding\_enabled\_flag 는 부호 비트 은닉이 인에이블된다고 규정한다. 본 개시물은 용어들 "부호 비트 은닉" 과 "부호 데이터 은닉" 을 상호교환 가능하게 사용할 수도 있다. 더욱이, 비디오 코더는 변환 계수들을 나타내는 십진수 엘리먼트들을 서브-블록 스캐닝 순서에 따라서 편성할 수도 있다. 서브-블록 스캐닝 순서로, 비디오 코더는 변환 계수 블록을 서브-블록들로 분할하고, 각각의 서브-블록들은 폭이 4 개의 변환 계수들이고 높이가 4 개의 변환 계수들이다. 비디오 코더는 서브-블록들의 각각에 대해 부호 은닉 (sign hidden) 변수 (즉, signHidden) 을 결정할 수도 있다. 변환 계수 블록의 4x4 서브-블록에서의 제 1 유효 (즉, 비-제로) 변환 계수와 변환 계수 블록의 4x4 서브-블록에서의 최종 유효 변환 계수 사이의 위치들에서의 차이가 3 보다 크고 (즉, lastSignScanPos - firstSignScanPos > 3), 변환 및 양자화 단계들이 현재의 CU 에 대해 바이패스되며 (즉, 스킵되며) 않을 때 (즉, !cu\_transquant\_bypass\_flag), 비디오 코더는 4x4 서브-블록에 대한 부호 은닉 변수 (즉, signHidden) 이 1 과 동일하다고 결정할 수도 있다. 변환이 스킵되거나 또는 바이패스되면, 변환 계수들은 실제로 예측 잔여들이다.

4x4 서브-블록에서의 각각의 개별의 변환 계수에 대해, 각각의 변환 계수에 대한 부호 십진수 엘리먼트 (예컨대, coeff\_sign\_flag[n]) 는, 각각의 변환 계수가 유효하면 (즉, 비-제로이면) 그리고 sign\_data\_hiding\_enabled\_flag 가 부호 비트 은닉이 인에이블되지 않는다고 규정하거나, 4x4 서브-블록에 대한 부호 은닉 변수가 1 과 동일하지 않거나, 또는 각각의 변환 계수가 4x4 서브-블록에서의 제 1 유효 변환 계수가 아니면, 비트스트림으로 시그널링된다. 예를 들어, 유효 변환 계수의 부호는, sign\_data\_hiding\_enabled\_flag 가 부호 비트 은닉이 인에이블된다고 나타내고, 4x4 서브-블록의 제 1 유효 계수와 4x4 서브-블록의 최종 유효 계수 사이의 위치에서의 차이가 3 보다 더 크고, 4x4 서브-블록이 변환 및 양자화 바이패스 코딩 (예컨대, 무손실 코딩) 을 사용하여 생성되지 않았으며, 각각의 변환 계수가 4x4 서브-블록의 제 1 유효 변환 계수이면, 시그널링되지 않는다.

JCTVC-M1005\_v2 의 섹션 7.4.3.3 에 규정된 바와 같이, coeff\_sign\_flag[n] 는 스캐닝 위치 n 에 대한 변환 계수 레벨의 부호를 다음과 같이 규정한다. coeff\_sign\_flag[n] 이 0 과 동일하면, 대응하는 변환 계수 레벨은 양의 값을 갖는다. 그렇지 않으면 (coeff\_sign\_flag[n] 이 1 과 동일하면), 대응하는 변환 계수 레벨은 음의 값을 갖는다. coeff\_sign\_flag[n] 이 존재하지 않을 때, coeff\_sign\_flag[n] 은 0 과 동일한 것으로 추론된다.

부호 데이터 은닉이 인에이블될 때, 비디오 인코더 (20) 는 변환 계수의 값/잔여 샘플 값 자체에 변환 계수/잔여 샘플의 부호를 나타내는 데이터를 내장할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 패리티 정보가 4×4 서브블록에서 제 1 유효 계수가 양 또는 음인지 여부를 결정하기 위해 비디오 디코더 (30) 에 의해 사용될 수 있도록, 변환 계수/잔여 샘플을 나타내는 하나 이상의 비트들을 수정할 수도 있다. 예를 들어, 4×4 서브블록에서 변환 계수들의 절대값들의 합계가 짝수이면, 부호는 양으로 추론되고, 그렇지 않으면, 부호는 음으

로 추론된다. 일반적으로, 비디오 인코더 (20) 는  $4 \times 4$  서브블록에서 양자화된 변환 계수/잔여 샘플들 중 하나의 표현의 최하위 비트를 변경할 수도 있다. 그 결과, 사용자는 비디오 인코더 (20) 가 변환 계수/잔여 샘플의 부호를 나타내기 위해 변환 계수/잔여 샘플의 표현의 비트를 수정하는 것으로 인해서, 시각적 품질의 어려운 손실을 인지불가능할 수도 있다.

[0299] 손실 코딩하는 본 개시물의 기법들을 사용할 때, 비디오 인코더는 설명된 기법들이 적용되는 블록들에 대해 부호 데이터 은닉을 구현하는 것이 불가능할 수도 있다. 이들은 변환이 스kip되고 모드들이 평면 및/또는 DC 인트라 블록들이거나, 및/또는 잔여 DPCM 이 적용되는 모드들이다. 변환 계수/잔여 샘플의 부호를 나타내기 위해 변환 계수/잔여 샘플의 표현에서 비트들의 값들을 변경하는 비디오 인코더 (20) 에 의해 에러들이 도입될 수도 있다. 더욱이, 이를 기법들을 사용하여, 하나의 수정된 잔여 샘플에서의 에러가 후속 잔여 샘플들로 전파할 수도 있다. 이러한 에러들은, 잔여 DPCM 이 변환 계수들/잔여 샘플들을 결정하기 위해 다수의 변환 계수들/잔여 샘플들과 함께 가산하는 비디오 디코더 (30) 에 의존할 수도 있기 때문에, 잔여 DPCM 이 적용될 때 혼합될 수도 있다. 따라서, 부호 데이터 은닉은 실제로 성능의 열화를 초래할 수도 있다. 이러한 경우, 부호 데이터 은닉은 규범적으로 디스에이블될 수도 있다. 이것은, 부호 데이터 은닉이 사용되고 있다는 것을 비트스트림으로 (또는, 디풀트 선택으로) 나타내어지더라도, 부호 데이터 은닉이 어떤 블록들에 대해 디스에이블된다는 것을 의미한다.

[0300] 일 예에서, 비디오 코더는, 비디오 코더가 블록에 변환을 적용하지 않거나, 블록이 평면 인트라 예측 모드 또는 DC 인트라 예측 모드를 사용하여 인트라 예측되거나, 또는 블록이 잔여 DPCM 이 적용되는 인트라 예측 모드를 사용하여 인트라 예측되면, 블록에 대한 부호 데이터 은닉을 디스에이블할 수도 있다. 예를 들어, 일부 예들에서, 부호 데이터 은닉은, 변환이 스kip되고 블록이 인트라-코딩되고 그리고 인트라 모드가 평면 인트라 예측 일 때 디스에이블될 수도 있다. 더욱이, 일부 예들에서, 부호 데이터 은닉은, 변환이 스kip되고 블록이 인트라-코딩되고 그리고 인트라 모드가 DC 인트라 예측일 때 디스에이블될 수도 있다. 더욱이, 일부 예들에서, 부호 데이터 은닉은, 변환이 스kip되고 블록이 인트라-코딩되고 그리고 인트라 모드가 잔여 DPCM 이 적용되는 모드일 때 디스에이블될 수도 있다.

[0301] 예를 들어, 비디오 인코더 (20) 는 현재 블록이 잔여 데이터로의 변환 (예컨대, 이산 코사인 변환, 방향성 변환, 또는 다른 변환) 의 적용 없이 손실 코딩을 사용하여 생성되고 현재 블록이 잔여 DPCM 이 사용되는 인트라 예측 모드를 사용하여 인트라 예측되면, 부호 데이터 은닉이 현재 블록에 대해 디스에이블된다고 결정할 수도 있다. 부호 데이터 은닉이 현재 블록에 대해 디스에이블될 때, 비디오 인코더 (20) 는 비트스트림에, 현재 블록에서의 각각의 개별의 유효 값에 대해, 각각의 유효 값이 양인지 또는 음인지 여부를 나타내는 각각의 선택스 엘리먼트를 포함시킬 수도 있다. 이와 유사하게, 이 예에서, 비디오 디코더 (30) 는, 현재 블록이 잔여 데이터에의 변환 (예컨대, 이산 코사인 변환, 방향성 변환, 또는 다른 변환) 의 적용 없이 손실 코딩을 사용하여 생성되고 잔여 DPCM 이 사용되는 인트라 예측 모드를 사용하여 현재 블록이 인트라 예측되면, 부호 데이터 은닉이 현재 블록에 대해 디스에이블된다고 결정할 수도 있다. 이러한 예들에서, 부호 데이터 은닉이 현재 블록에 대해 디스에이블될 때, 비디오 디코더 (30) 는 비트스트림으로부터, 블록에서의 각각의 개별의 유효 값에 대해, 각각의 유효 값이 양인지 또는 음인지 여부를 나타내는 각각의 선택스 엘리먼트를 획득한다.

[0302] 다른 예들에서, 부호 데이터 은닉은 변환이 스kip되고 블록이 인트라-코딩될 때 블록에 대해, 또는 모든 변환-스킵 블록들에 대해 디스에이블될 수도 있다. 따라서, 이 예에서, 부호 데이터 은닉은 어느 인트라 예측 모드가 사용되는지 여부에 관계없이 디스에이블될 수도 있다. 예를 들어, 비디오 코더는, 현재 블록이 현재 블록의 잔여 데이터에 대한 변환의 적용 없이 코딩되고 그리고 현재 블록이 DC 인트라 예측 모드 또는 평면 인트라 예측 모드를 사용하여 인트라 코딩되면, 부호 데이터 은닉이 현재 블록에 대해 디스에이블된다고 결정할 수도 있다.

[0303] 본 개시물이 잔여 DPCM 과 DC 및 평면에 대한 변경들에 대한 기법들을 설명하였다는 점에 유의해야 한다. 이를 기법들 중 임의의 하나 또는 조합이 사용될 수도 있다.

[0304] 도 6 은 본 개시물의 기법들을 구현할 수도 있는 예시적인 비디오 인코더 (20) 을 예시하는 블록도이다. 도 6 은 설명의 목적들을 위해 제공되며 본 개시물에서 넓게 예시되고 설명된 바와 같은 기법들의 한정으로 간주되지 않아야 한다. 설명의 목적을 위해, 본 개시물은 HEVC 코딩의 맥락에서 비디오 인코더 (20) 을 기술한다. 그러나, 본 개시물의 기법들은 다른 코딩 표준들 또는 방법들에 적용가능할 수도 있다.

[0305] 도 6 의 예에서, 비디오 인코더 (20) 는 예측 프로세싱 유닛 (100), 잔여 생성 유닛 (102), 변환 프로세싱 유닛 (104), 양자화 유닛 (106), 역양자화 유닛 (108), 역변환 프로세싱 유닛 (110), 재구성 유닛 (112), 웨이퍼 유닛

(114), 디코딩 화상 버퍼 (116), 및 엔트로피 인코딩 유닛 (118) 을 포함한다. 예측 프로세싱 유닛 (100) 은 인터-예측 프로세싱 유닛 (120) 및 인트라-예측 프로세싱 유닛 (126) 을 포함한다. 인터-예측 프로세싱 유닛 (120) 은 모션 추정 유닛 (122) 및 모션 보상 유닛 (124) 를 포함한다. 다른 예들에서, 비디오 인코더 (20) 는 더 많거나, 더 적거나, 또는 상이한 기능적 구성요소들을 포함할 수도 있다.

[0306] 비디오 인코더 (20) 는 비디오 데이터를 수신할 수도 있다. 비디오 인코더 (20) 는 비디오 데이터의 화상의 슬라이스에서 각각의 CTU 를 인코딩할 수도 있다. CTUs 의 각각은 화상의 동일-사이즈로된 루마 코딩 트리 블록들 (CTBs) 및 대응하는 CTBs 와 연관될 수도 있다. CTU 를 인코딩하는 것의 일부로서, 예측 프로세싱 유닛 (100) 은 퀘드-트리 파티셔닝을 수행하여, CTU 의 CTBs 를 계속해서-더 작은 블록들로 분할할 수도 있다. 더 작은 블록은 CUs 의 코딩 블록들일 수도 있다. 예를 들어, 예측 프로세싱 유닛 (100) 은 CTU 와 연관되는 CTB 를 4개의 동일-사이즈로된 서브-블록들로 파티셔닝하고, 서브-블록들 중 하나 이상을 4개의 동일-사이즈로된 서브-서브-블록들로, 그리고 기타 등등으로 파티셔닝할 수도 있다.

[0307] 비디오 인코더 (20) 는 CUs 의 인코딩된 표현들 (즉, 코딩된 CUs) 를 생성하기 위해 CTU 의 CUs 를 인코딩할 수도 있다. CU 를 인코딩하는 것의 일부로서, 예측 프로세싱 유닛 (100) 은 CU 의 하나 이상의 PUs 중에서 CU 와 연관되는 코딩 블록들을 파티셔닝할 수도 있다. 따라서, 각각의 PU 는 루마 예측 블록 및 대응하는 크로마 예측 블록들과 연관될 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 여러 사이즈들을 가지는 PUs 를 지원할 수도 있다. 위에서 나타낸 바와 같이, CU 의 사이즈는 CU 의 루마 코딩 블록의 사이즈를 지칭할 수도 있고, PU 의 사이즈는 루마 PU 의 예측 블록의 사이즈를 지칭할 수도 있다. 특정의 CU 의 사이즈가  $2Nx2N$  이라고 가정하면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 인트라 예측에 대해서는  $2Nx2N$  또는  $NxN$  의 PU 사이즈들을, 그리고 인터 예측에 대해서는  $2Nx2N$ ,  $2NxN$ ,  $Nx2N$ ,  $NxN$ , 또는 등등의 대칭 PU 사이즈들을 지원할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 또한 인터 예측에 대해서  $2NxN$ ,  $2NxN$ ,  $nLx2N$ , 및  $nRx2N$  의 PU 사이즈들에 대한 비대칭적인 파티셔닝을 지원할 수도 있다.

[0308] 인터-예측 프로세싱 유닛 (120) 은 CU 의 각각의 PU 에 대해 인터 예측을 수행함으로써, PU 에 대한 예측 데이터를 생성할 수도 있다. 예측 PU 에 대한 데이터는 PU 의 예측 샘플 블록들 (즉, 예측 블록들) 및 PU 의 모션 정보를 포함할 수도 있다. 인터-예측 유닛 (121) 은 PU 가 I 슬라이스, P 슬라이스, 또는 B 슬라이스 내에 있는지에 따라서, CU 의 PU 에 대해 상이한 동작들을 수행할 수도 있다. I 슬라이스에서, 모든 PUs 는 인트라 예측된다. 그러므로, PU 가 I 슬라이스에 있으면, 인터-예측 유닛 (121) 은 PU 에 대해 인터 예측을 수행하지 않는다. 따라서, I-모드에서 인코딩되는 블록들에 대해, 예측 블록은 동일한 프레임 내에서 이전에-인코딩된 이웃하는 블록들로부터의 공간 예측을 사용하여 형성된다.

[0309] PU 가 P 슬라이스 내에 있을 때, 모션 추정 유닛 (122) 은 참조 화상들의 리스트 (예컨대, "RefPicList0") 에서의 참조 화상들을 PU 에 대한 참조 영역에 대해 탐색할 수도 있다. PU 에 대한 참조 영역은 PU 의 샘플 블록들에 가장 가깝게 대응하는 샘플들 (예컨대, 샘플 블록들) 을 포함하는, 참조 화상 내, 영역일 수도 있다. 모션 추정 유닛 (122) 은 PU 에 대한 참조 영역을 포함하는 참조 화상의 RefPicList0 에서의 위치를 나타내는 참조 인덱스를 생성할 수도 있다. 게다가, 모션 추정 유닛 (122) 은 PU 의 코딩 블록과, 참조 영역과 연관되는 참조 로케이션 사이의 공간 변위를 나타내는 모션 벡터를 생성할 수도 있다. 예를 들어, 모션 벡터는 현재의 디코딩된 화상에서의 좌표들로부터 참조 화상에서의 좌표들까지 오프셋을 제공하는 2차원 벡터일 수도 있다. 모션 추정 유닛 (122) 은 참조 인덱스 및 모션 벡터를 PU 의 모션 정보로서 출력할 수도 있다. 모션 보상 유닛 (124) 은 PU 의 모션 벡터에 의해 나타내는 참조 로케이션에서의 실제 또는 내삽된 샘플들에 기초하여, PU 의 예측 샘플 블록들을 생성할 수도 있다.

[0310] PU 가 B 슬라이스 내에 있을 때, 모션 추정 유닛 (122) 은 PU 에 대해 단방향-예측 또는 양방향-예측을 수행할 수도 있다. PU 에 대한 단방향-예측을 수행하기 위해, 모션 추정 유닛 (122) 은 RefPicList0 또는 제 2 참조 화상 리스트 ("RefPicList1") 의 참조 화상들을 PU 에 대한 참조 영역에 대해 탐색할 수도 있다. 모션 추정 유닛 (122) 은 참조 영역을 포함하는 참조 화상의 RefPicList0 또는 RefPicList1 에서의 위치를 나타내는 참조 인덱스, PU 의 샘플 블록과 참조 영역과 연관되는 참조 로케이션 사이의 공간 변위를 나타내는 모션 벡터, 및 참조 화상이 RefPicList0 또는 RefPicList1 에 있는지 여부를 나타내는 하나 이상의 예측 방향 표시 (indication) 들을, PU 의 모션 정보로서 출력할 수도 있다. 모션 보상 유닛 (124) 은 PU 의 모션 벡터에 의해 나타내는 참조 영역에서의 실제 또는 내삽된 샘플들에 적어도 부분적으로 기초하여, PU 의 예측 샘플 블록들을 생성할 수도 있다.

[0311] PU 에 대한 양방향 인터 예측을 수행하기 위해, 모션 추정 유닛 (122) 은 RefPicList0 에서의 참조 화상들을 PU

에 대한 참조 영역에 대해 탐색할 수도 있고, 또한 RefPicList1에서의 참조 화상들을 PU에 대한 또 다른 참조 영역에 대해 탐색할 수도 있다. 모션 추정 유닛(122)은 참조 영역들을 포함하는 참조 화상들의 RefPicList0 및 RefPicList1에서의 위치를 나타내는 참조 화상 인덱스들(즉, 참조 인덱스들)을 생성할 수도 있다. 게다가, 모션 추정 유닛(122)은 참조 영역들과 연관되는 참조 로케이션과 PU의 샘플 블록 사이의 공간 변위들을 나타내는 모션 벡터들을 생성할 수도 있다. PU의 모션 정보는 PU의 참조 인덱스들 및 모션 벡터들을 포함할 수도 있다. 모션 보상 유닛(124)은 PU의 모션 벡터에 의해 나타내는 참조 영역에서의 실제 또는 내삽된 샘플들에 적어도 부분적으로 기초하여, PU의 예측 샘플 블록들을 생성할 수도 있다.

[0312] 본 개시물의 하나 이상의 기법들에 따르면, (인트라-예측 프로세싱 유닛(126)과 같은) 예측 프로세싱 유닛(100) 내 하나 이상의 유닛들은 본원에서 설명되는 기법들 중 하나 이상의 기법을 비디오 인코딩 프로세스의 일부로서 수행할 수도 있다.

[0313] 인트라-예측 프로세싱 유닛(126)은 PU에 대해 인트라 예측을 수행함으로써, PU에 대한 예측 데이터를 생성할 수도 있다. 예측 PU에 대한 데이터는 PU에 대한 예측 블록들 및 여러 선택스 엘리먼트들을 포함할 수도 있다. 인트라-예측 프로세싱 유닛(126)은 I 슬라이스들, P 슬라이스들, 및 B 슬라이스들에서의 PUs에 대해 인트라 예측을 수행할 수도 있다.

[0314] PU상에 인트라 예측을 수행하기 위해, 인트라-예측 프로세싱 유닛(126)은 다수의 인트라 예측 모드들을 사용하여, 예측 PU에 대한 다수의 데이터의 세트들을 생성할 수도 있다. 일부 인트라 예측 모드들을 사용하여 예측 PU에 대한 데이터의 세트를 생성하기 위해, 인트라-예측 프로세싱 유닛(126)은 이웃하는 PUs의 샘플 블록들로부터 PU의 샘플 블록들을 가로질러 인트라 예측 모드와 연관되는 방향으로 샘플들을 확장할 수도 있다. 이웃하는 PUs는 PUs, CUs, 및 CTUs에 대해 좌-우, 상-하 인코딩 순서를 가정하면, PU의 상측에, 우상부에, 좌상부에, 또는 좌측에 있을 수도 있다. 인트라-예측 프로세싱 유닛(126)은 다수의 인트라 예측 모드들, 예컨대, 33개의 방향 인트라 예측 모드들을 사용할 수도 있다. 일부 예들에서, 인트라 예측 모드들의 개수는 PU와 연관되는 영역의 사이즈에 의존할 수도 있다. 더욱이, 본원에서 설명된 바와 같은, 인트라-예측 프로세싱 유닛(126)은 본원에서 설명되는 무손실 코딩 모드들 및 변경들을 구현하여, 이러한 코딩 모드들을 향상시킬 수도 있다. 본 개시물의 일부 기법들에 따르면, 인트라-예측 프로세싱 유닛(126)은 블록 내 하나 이상의 원래 샘플 값들을 사용하여, 블록 내 다른 샘플 값들의 인트라 DC 예측을 수행할 수도 있다.

[0315] 예측 프로세싱 유닛(100)은 PUs에 대해 인터-예측 프로세싱 유닛(120)에 의해 생성된 예측 데이터, 또는 PUs에 대해 인트라-예측 프로세싱 유닛(126)에 의해 생성된 예측 데이터 중에서, CU의 PUs에 대한 예측 데이터를 선택할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛(100)은 예측 데이터의 세트들의 레이트/왜곡 메트릭들에 기초하여, CU의 예측 PU에 대한 데이터들을 선택한다. 선택된 예측 데이터의 예측 샘플 블록들은 본원에서, 선택된 예측 샘플 블록들로서 지칭될 수도 있다.

[0316] 잔여 생성 유닛(102)은 CU의 코딩 블록들(예컨대, 루마, Cb 및 Cr 코딩 블록들) 및 CU의 PUs의 선택된 예측 블록들(예컨대, 루마, Cb 및 Cr 블록들)에 기초하여, CU에 대한 잔여 블록들(예컨대, 루마, Cb 및 Cr 잔여 블록들)을 생성할 수도 있다. 예를 들어, 잔여 생성 유닛(102)은 잔여 블록들에서의 각각의 샘플이 CU의 코딩 블록에서의 샘플과 CU의 PU의 대응하는 선택된 예측 블록에서의 대응하는 샘플 사이의 차이와 동일한 값을 갖도록, CU의 잔여 블록들을 생성할 수도 있다.

[0317] 변환 프로세싱 유닛(104)은 퀘드-트리 파티셔닝을 수행하여, CU와 연관되는 잔여 블록들을 CU의 TUs와 연관되는 변환 블록들로 파티셔닝할 수도 있다. 따라서, TU는 루마 변환 블록 및 2개의 크로마 변환 블록들과 연관될 수도 있다. CU의 TUs의 루마 및 크로마 변환 블록들의 사이즈들 및 위치들은 CU의 PUs의 예측 블록들의 사이즈들 및 위치들에 기초하거나 또는 기초하지 않을 수도 있다. "잔여 퀘드-트리"(RQT)로서 알려진 퀘드-트리 구조는 그 영역들 각각과 연관되는 노드들을 포함할 수도 있다. CU의 TUs는 RQT의 리프 노드들에 대응할 수도 있다.

[0318] 변환 프로세싱 유닛(104)은 하나 이상의 변환들을 TU의 변환 블록들을 적용함으로써, CU의 각각의 TU에 대한 변환 계수 블록들을 생성할 수도 있다. 변환 프로세싱 유닛(104)은 여러 변환들을 TU와 연관되는 변환 블록에 적용할 수도 있다. 예를 들어, 변환 프로세싱 유닛(104)은 변환 블록에 이산 코사인 변환(DCT), 방향성 변환, 또는 개념적으로 유사한 변환을 적용할 수도 있다. 일부 예들에서, 변환 프로세싱 유닛(104)은 변환들을 변환 블록에 적용하지 않는다. 이러한 예들에서, 변환 블록은 변환 계수 블록으로서 취급될 수도 있다.

[0319] 양자화 유닛 (106)은 계수 블록에서의 변환 계수들을 양자화할 수도 있다. 양자화 프로세스는 그 변환 계수들의 일부 또는 모두와 연관되는 비트 심도를 감소시킬 수도 있다. 예를 들어, n-비트 변환 계수는 양자화 동안 m-비트 변환 계수로 절사될 수도 있고, 여기서, n은 m보다 더 크다. 양자화 유닛 (106)은 CU와 연관되는 양자화 파라미터 (QP) 값에 기초하여, CU의 TU와 연관되는 계수 블록을 양자화할 수도 있다. 비디오 인코더 (20)는 CU와 연관되는 QP 값을 조정함으로써 CU와 연관되는 계수 블록들에 적용되는 양자화의 정도를 조정할 수도 있다. 양자화는 정보의 손실을 도입할 수도 있고, 따라서 양자화된 변환 계수들은 원래 정밀도들보다 낮은 정밀도를 가질 수도 있다.

[0320] 본 개시물의 일부 예들에 따르면, 다음이  $0 \leq i \leq (M - 1)$  및  $0 \leq j \leq (N - 1)$ 에 대해 수행될 수도 있고, 여기서, M은 블록의 높이이고 N은 블록의 폭이다. 이러한 예들에서, 블록은 인트라 예측을 사용하여 생성되는 예측 블록에서의 샘플 값들과 원래 샘플들 값들 사이의 차이들을 나타내는 잔여 값들을 포함하는 잔여 블록이다. 더욱이, 이러한 예들에서, 블록은 변환 스킵 블록이다. 잔여 생성 유닛 (102)은 잔여 값  $r_{i,j}$

에 대한 수정된 잔여 값  $\tilde{r}_{i,j}$ 을 결정할 수도 있다. 블록이 수직 인트라 예측 모드를 사용하여 코딩되면,

$\tilde{r}_{i,j}$ 는 다음과 같이 정의되며:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i = 0, 0 \leq j \leq (N - 1) \\ r_{i,j} - Q(r_{(i-1),j}), & 1 \leq i \leq (M - 1), 0 \leq j \leq (N - 1) \end{cases}$$

[0321] [0322] 상기 방정식에서,  $Q(r_{(i-1),j})$ 는 잔여 값  $r_{i,j}$ 의 한 칼럼 좌측에 있는 잔여 값  $r_{i-1,j}$ 에 대한 재구성된 잔여 값을 나타낸다. 블록이 수평 인트라 예측 모드를 사용하여 코딩되면,  $\tilde{r}_{i,j}$ 는 다음과 같이 정의되며:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & 0 \leq i \leq (M - 1), j = 0 \\ r_{i,j} - Q(r_{i,(j-1)}) & 0 \leq i \leq (M - 1), 1 \leq j \leq (N - 1) \end{cases}$$

[0323] [0324] 상기 방정식에서,  $Q(r_{i,(j-1)})$ 은 잔여 값  $r_{i,j}$ 의 한 로우 상부에 있는 잔여 값  $r_{i,j-1}$ 에 대한 재구성된 잔여 값을 나타낸다. 양자화 유닛 (106)은 수정된 잔여 값  $\tilde{r}_{i,j}$ 을 양자화하여, 양자화된 수정된 잔여 값  $Q(\tilde{r}_{i,j})$ 을 생성한다.

[0325] 역양자화 유닛 (108) 및 역변환 프로세싱 유닛 (110)은 역양자화 및 역변환들을 계수 블록에 각각 적용하여, 계수 블록으로부터 잔여 블록을 재구성할 수도 있다. 재구성 유닛 (112)는 그 재구성된 잔여 블록을 예측 프로세싱 유닛 (100)에 의해 생성되는 하나 이상의 예측 블록들로부터의 대응하는 샘플들에 가산하여, TU와 연관되는 재구성된 변환 블록을 생성할 수도 있다. 이 방법으로 CU의 각각의 TU에 대한 변환 블록들을 재구성함으로써, 비디오 인코더 (20)는 CU의 코딩 블록들을 재구성할 수도 있다.

[0326] 필터 유닛 (114)은 필터 유닛 (114)은 디블록킹 블록들에서 블록킹 아티팩트들을 감소시키기 위해, 하나 이상의 디블록킹 동작들을 수행할 수도 있다. 디코딩 화상 버퍼 (116)는 버퍼 (116)는 14가 재구성된 코딩 블록들에 관해 하나 이상의 디블록킹 동작들을 수행한 후 그 재구성된 코딩 블록들을 저장할 수도 있다. 인터-예측 프로세싱 유닛 (120)은 그 구성된 코딩 블록들을 포함하는 참조 화상을 사용하여, 다른 화상들의 PUs에 관해 인터 측을 수행할 수도 있다. 게다가, 인트라-예측 프로세싱 유닛 (126)은 디코딩 화상 버퍼 (116)에서의 재구성된 코딩 블록들을 사용하여, CU와 동일한 화상에서의 다른 PUs에 관해 인트라 예측을 수행할 수도 있다.

[0327] 엔트로피 인코딩 유닛 (118)은 비디오 인코더 (20)의 다른 기능적 구성요소들로부터 데이터를 수신할 수도 있다. 예를 들어, 엔트로피 인코딩 유닛 (118)은 양자화 유닛 (106)으로부터 계수 블록들을 수신할 수도 있으며 예측 프로세싱 유닛 (100)으로부터 신택스 엘리먼트들을 수신할 수도 있다. 엔트로피 인코딩 유닛 (118)은 데이터에 대해 하나 이상의 엔트로피 인코딩 동작들을 수행하여, 엔트로피-인코딩된 데이터를 생성할 수도 있다. 예를 들어, 엔트로피 인코딩 유닛 (118)은 컨텍스트-적용 가변 길이 코딩 (CAVLC) 동작, CABAC 동작, 변수-대-변수 (V2V) 길이 코딩 동작, 신택스-기반 컨텍스트-적용 2진 산술 코딩 (SBAC) 동작, 확률 간격

파티셔닝 엔트로피 (PIPE) 코딩 동작, 지수-콜롬 (Exponential Golomb) 인코딩 동작, 또는 또다른 유형의 엔트로피 인코딩 동작을 데이터에 대해 수행할 수도 있다. 비디오 인코더 (20)는 엔트로피 인코딩 유닛 (118)에 의해 생성된 엔트로피-인코딩된 데이터를 포함하는 비트스트림을 출력할 수도 있다. 예를 들어, 비트스트림은 CU에 대한 RQT를 나타내는 데이터를 포함할 수도 있다.

[0328] 본 개시물의 일부 예들에 따르면, 엔트로피 인코딩 유닛 (118)은, 현재 블록이 잔여 데이터에 대한 변환의 적용 없이 생성되고 잔여 DPCM이 사용되는 인트라 예측 모드를 사용하여 현재 블록이 인트라 예측되면, 부호 데이터 은닉이 현재 블록에 대해 디스에이블된다면 결정할 수도 있다. 부호 데이터 은닉이 현재 블록에 대해 디스에이블될 때, 엔트로피 인코딩 유닛 (118)은 비트스트림에, 현재 블록에서의 값이 양 또는 음인지 여부를 나타내는 신택스 엘리먼트를 포함시킬 수도 있다.

[0329] 도 6에서의 엘리먼트 (130)는 무손실 코딩과 손실 코딩 간에 선택하는 스위치 (또는, 개념적인 스위치)를 나타낼 수도 있다. 제어 신호 (132)는 무손실 또는 손실 코딩을 결정하는 예측 프로세싱 유닛 (100)으로부터의 신호를 나타낼 수도 있고, 엘리먼트 (134)는 역변환 및 역양자화 프로세스들을 바이패스하는 디코딩 루프를 나타낼 수도 있다. 일부 예들에서, 무손실 코딩은 변환들 및 양자화를 배제한다. 다른 예들에서, 무손실 코딩은 변환들을 수행하고 오직 양자화 프로세스만을 배제한다. 또한 다른 예들에서, 무손실 코딩은 변환들 및 정량 (quantitation)의 사용으로 구현될 수도 있고, 그러나 양자화 파라미터는 임의의 양자화 데이터 손실을 피할 수 있도록 선택될 수도 있다. 이들 및 다른 예들은 본 개시물의 범위 내이다.

[0330] 엘리먼트들 (136 및 138)은 변환 스킵핑 모드를 구현하는데 사용될 수도 있는 스위치들 (또는, 개념적인 스위치들)을 나타낸다. 변환 스kipping 모드들에서, 잔여 데이터는 변환 프로세싱 유닛 (104)에 의해 변환되지 않고 양자화 유닛 (106)에 의해 양자화된다. 따라서, 엘리먼트 (136)의 대시 라인들 (dash lines)은 2개의 가능한 데이터 경로들을 나타낸다. 하나의 데이터에서, 잔여 데이터는 양자화 유닛 (106)에 의해 양자화되고, 다른 데이터 경로에서, 잔여 데이터는 양자화 유닛 (106)에 의해 양자화되지 않는다. 이와 유사하게, 비디오 인코더 (20)의 디코딩 루프에서, 잔여 데이터는 역양자화 유닛 (108)에 의해 역양자화되지만, 역변환 프로세싱 유닛 (110)에 의해 변환되지 않는다. 따라서, 엘리먼트 (138)의 대시 라인들은 잔여 데이터가 역양자화 유닛 (108)에 의해 역양자화되지만 역변환 프로세싱 유닛 (110)에 의해 변환되지 않는 대체 데이터 경로를 나타낸다.

[0331] 도 7은 본 개시물의 기법들을 구현하도록 구성된 예시적인 비디오 디코더 (30)을 예시하는 블록도이다. 도 7은 설명의 목적들을 위해 제공되며 본 개시물에 넓게 예시되고 설명된 것과 같은 기법들에 한정하는 것이 아니다. 설명의 목적을 위해, 본 개시물은 HEVC 코딩의 맥락에서의 비디오 디코더 (30)을 기술한다. 그러나, 본 개시물의 기법들은 다른 코딩 표준들 또는 방법들에 적용 가능할 수도 있다.

[0332] 도 7의 예에서, 비디오 디코더 (30)는 엔트로피 디코딩 유닛 (150), 예측 프로세싱 유닛 (152), 역양자화 유닛 (154), 역변환 프로세싱 유닛 (156), 재구성 유닛 (158), 필터 유닛 (160), 및 디코딩 화상 버퍼 (162)를 포함한다. 예측 프로세싱 유닛 (152)는 모션 보상 유닛 (164) 및 인트라-예측 프로세싱 유닛 (166)을 포함한다. 다른 예들에서, 비디오 디코더 (30)는 더 많거나, 더 적거나, 또는 상이한 기능적 구성요소들을 포함할 수도 있다.

[0333] 비디오 디코더 (30)는 비트스트림을 수신할 수도 있다. 엔트로피 디코딩 유닛 (150)은 비트스트림을 파싱하여, 비트스트림으로부터 신택스 엘리먼트들을 디코딩할 수도 있다. 엔트로피 디코딩 유닛 (150)은 비트스트림에서의 엔트로피-인코딩된 신택스 엘리먼트들을 엔트로피 디코딩할 수도 있다. 예측 프로세싱 유닛 (152), 역양자화 유닛 (154), 역변환 프로세싱 유닛 (156), 재구성 유닛 (158), 및 필터 유닛 (160)은 비트스트림으로부터 추출된 신택스 엘리먼트들에 기초하여, 디코딩된 비디오 데이터를 생성할 수도 있다.

[0334] 비트스트림은 일련의 NAL 유닛들을 포함할 수도 있다. 비트스트림의 NAL 유닛들은 코딩된 슬라이스 NAL 유닛들을 포함할 수도 있다. 비트스트림을 디코딩하는 것의 일부로서, 엔트로피 디코딩 유닛 (150)은 코딩된 슬라이스 NAL 유닛들로부터 신택스 엘리먼트들을 추출하여 엔트로피 디코딩할 수도 있다. 코딩된 슬라이스들 각각은 슬라이스 헤더 및 슬라이스 데이터를 포함할 수도 있다. 슬라이스 헤더는 슬라이스에 관련된 신택스 엘리먼트들을 포함할 수도 있다. 슬라이스 헤더에서의 신택스 엘리먼트들은 슬라이스를 포함하는 화상과 연관되는 PPS를 식별하는 신택스 엘리먼트를 포함할 수도 있다.

[0335] 본 개시물의 일부 예들에 따르면, 엔트로피 디코딩 유닛 (150)은 현재 블록이 잔여 데이터에 대한 변환의 적용 없이 생성되고 잔여 DPCM이 사용되는 인트라 예측 모드를 사용하여 현재 블록이 인트라 예측되면, 부호 데이터

은닉이 현재 블록에 대해 디스에이블된다고 결정할 수도 있다. 이러한 예들에서, 부호 데이터 은닉이 현재 블록에 대해 디스에이블될 때, 엔트로피 디코딩 유닛 (150)은 비트스트림으로부터, 블록에서의 각각의 개별의 유효 값에 대해, 각각의 유효 값이 양인지 또는 음인지 여부를 나타내는 각각의 신택스 엘리먼트를 획득한다.

[0336] 비트스트림으로부터 신택스 엘리먼트들을 디코딩하는 (즉, 획득하는) 것에 추가하여, 비디오 디코더 (30)는 비-파티셔닝된 CU에 대해 재구성 동작을 수행할 수도 있다. 비-파티셔닝된 CU에 대해 재구성 동작을 수행하기 위해, 비디오 디코더 (30)는 CU의 각각의 TU에 대해 재구성 동작을 수행할 수도 있다. CU의 각각의 TU에 대해 재구성 동작을 수행함으로써, 비디오 디코더 (30)는 CU의 잔여 블록들을 재구성할 수도 있다.

[0337] CU의 TU에 대해 재구성 동작을 수행하는 것의 일부로서, 역양자화 유닛 (154)은 TU와 연관되는 계수 블록들을 역양자화할 수도 있다, 즉, 양자화 해제할 수도 있다. 역양자화 유닛 (154)은 TU의 CU와 연관되는 QP 값을 사용하여, 적용할 역양자화 유닛 (154)에 대한 양자화의 정도 및 이와 유사하게, 역양자화의 정도를 결정할 수도 있다. 즉, 압축 비, 즉, 원래 시퀀스와 압축된 시퀀스를 나타내는데 사용되는 비트수의 비는 변환 계수들을 양자화할 때 사용되는 QP의 값을 조정함으로써 제어될 수도 있다. 압축 비는 또한 채용되는 엔트로피 코딩의 방법에 의존할 수도 있다.

[0338] 역양자화 유닛 (154)이유닛 (154)이역양자화한 후, 역변환 프로세싱 유닛 (156)은 TU와 연관되는 잔여 블록을 생성하기 위해, 하나 이상의 역변환들을 계수 블록에 적용할 수도 있다. 예를 들어, 역변환 프로세싱 유닛 (156)은 역 DCT, 역 정수 변환, 역 카루넨 루베 (Karhunen-Loeve) 변환 (KLT), 역 회전 변환, 역 방향 변환, 또는 또 다른 역변환을 계수 블록에 적용할 수도 있다.

[0339] PU가 인트라 예측을 사용하여 인코딩되면, 인트라-예측 프로세싱 유닛 (166)은 인트라 예측을 수행하여, PU에 대한 예측 블록들을 생성할 수도 있다. 인트라-예측 프로세싱 유닛 (166)은 인트라 예측 모드를 사용하여, 공간적으로-이웃하는 PUs의 예측 블록들에 기초하여 PU에 대한 예측 블록들 (예컨대, 예측 루마, Cb 및 Cr 블록들)을 생성할 수도 있다. 인트라-예측 프로세싱 유닛 (166)은 비트스트림으로부터 디코딩된 하나 이상의 신택스 엘리먼트들에 기초하여, PU에 대한 인트라 예측 모드를 결정할 수도 있다. 본 개시물의 일부 기법들에 따르면, 인트라-예측 프로세싱 유닛 (166)은 블록 내 하나 이상의 원래 샘플 값들을 사용하여, 블록 내 다른 샘플 값들의 인트라 DC 예측을 수행할 수도 있다. 즉, 인트라-예측 프로세싱 유닛 (166)은 예측 블록을 생성할 수도 있다. 예측 블록을 생성하는 것의 일부로서, 인트라-예측 프로세싱 유닛 (166)은 현재의 샘플의 DC 예측을 위해, 예측 블록의 현재의 로우에서 현재의 샘플의 좌측에 있는 무손실로 재구성된 샘플 및 현재의 로우의 상부에 있는 예측 블록의 로우에 대한 무손실로 재구성된 샘플 중 적어도 하나를 사용할 수도 있다.

[0340] 예측 프로세싱 유닛 (152)는 비트스트림으로부터 추출된 신택스 엘리먼트들에 기초하여, 제 1 참조 화상 리스트 (RefPicList0) 및 제 2 참조 화상 리스트 (RefPicList1)을 구성할 수도 있다. 더욱이, PU가 인터 예측을 사용하여 인코딩되면, 엔트로피 디코딩 유닛 (150)은 PU에 대한 모션 정보를 추출할 수도 있다. 모션 보상 유닛 (164)는 PU의 모션 정보에 기초하여, PU에 대한 하나 이상의 참조 영역들을 결정할 수도 있다. 모션 보상 유닛 (164)는 PU에 대한 하나 이상의 참조 블록들에서의 샘플들 블록들에 기초하여, PU에 대한 예측 블록들 (예컨대, 루마, Cb 및 Cr 블록들)을 생성할 수도 있다. 본 개시물의 하나 이상의 기법들에 따르면, (인트라 예측 프로세싱 유닛 (166)과 같은) 예측 프로세싱 유닛 (152) 내 하나 이상의 유닛들은 본원에서 설명하는 기법들을 비디오 디코딩 프로세스의 일부로서 수행할 수도 있다.

[0341] 재구성 유닛 (158)은 적용가능한 경우, CU의 TUs와 연관되는 변환 블록 (예컨대, 루마, Cb 및 Cr 변환 블록들) 및 CU의 PUs의 예측 블록들 (예컨대, 루마, Cb 및 Cr 예측 블록들), 즉, 인트라-예측 데이터 또는 인터-예측 데이터를 사용하여, CU의 코딩 블록들 (예컨대, 루마, Cb 및 Cr 코딩 블록들)을 재구성할 수도 있다.

예를 들어, 재구성 유닛 (158)은 루마, Cb 및 Cr 변환 블록들의 샘플들을 예측 루마, Cb 및 Cr 블록들의 대응하는 샘플들에 추가하여, CU의 루마, Cb 및 Cr 코딩 블록들을 재구성할 수도 있다.

[0342] 본 개시물의 일부 예들에 따르면, 엔트로피 디코딩 유닛 (150)은 잔여 값들의 블록을 생성할 수도 있다. 이 블록은 변환 스킵 블록일 수도 있다. 더욱이, 블록은 원래 샘플 값들과, 인트라 예측을 사용하여 생성되는 예측 블록에서의 샘플 값들 사이의 차이들을 나타내는 잔여 값들을 포함하는 잔여 블록일 수도 있다. 더욱이,  $0 \leq i \leq (M - 1)$  및  $0 \leq j \leq (N - 1)$ 에 대해, 역양자화 유닛 (154)은 블록에서 잔여 값  $r_{i,j}$ 에 대한 재구성된 잔여 값  $Q(r_{i,j})$ 를 계산할 수도 있고, 여기서, M은 블록의 높이이고 N은 블록의 폭이다. 블록이 수직 인트라 예측 모드를 사용하여 코딩되면 (또는, 일부 예들에서, 근사-수직 인트라 예측 모드),  $Q(r_{i,j})$

는 다음과 같이 정의되며:

$$[0343] Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & i=0, 0 \leq j \leq (N-1), \\ Q(\tilde{r}_{i,j}) + Q(r_{i-1,j}) & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases}$$

[0344] 상기 방정식에서,  $\tilde{Q}(\tilde{r}_{i,j})$  는 수정된 잔여 값  $\tilde{r}_{i,j}$  의 양자화된 버전을 나타내고, 수정된 잔여 값  $\tilde{r}_{i,j}$  는 잔여 값  $r_{i,j}$  의 수정된 버전이고,  $Q(r_{i-1,j})$  는 잔여 값  $r_{i,j}$  의 한 칼럼 좌측에 있는 잔여 값에 대한 (즉, 대응하는) 재구성된 잔여 값이다. 엔트로피 디코딩 유닛 (150)은 엔트로피 디코딩 유닛 (150)이  $Q(r_{i,j})$ 를 결정하는 방법과 동일한 방법으로,  $Q(r_{i-1,j})$ 를 미리 결정하였을 수도 있다. 블록이 수평 인트라 예측 모드를 사용하여 코딩되면,  $Q(r_{i,j})$ 는 다음과 같이 정의되며:

$$[0345] Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & 0 \leq i \leq (M-1), j=0, \\ Q(\tilde{r}_{i,j}) + Q(r_{i,j-1}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}$$

[0346] 상기 방정식에서,  $Q(r_{i,j-1})$ 은 잔여 값  $r_{i,j}$ 의 한 로우 상부에 있는 잔여 값에 대한 재구성된 잔여 값이다. 엔트로피 디코딩 유닛 (150)은 엔트로피 디코딩 유닛 (150)이  $Q(r_{i,j})$ 를 결정하는 방법과 동일한 방법으로,  $Q(r_{i,j-1})$ 을 미리 결정하였을 수도 있다. 재구성 유닛 (158)은 재구성된 잔여 값  $Q(r_{i,j})$ 를 예측 값에 가산하여, 샘플 값을 재구성할 수도 있다.

[0347] 필터 유닛 (160)은 CU의 코딩 블록들 (예컨대, 루마, Cb 및 Cr 코딩 블록들)과 연관되는 블록킹 아티팩트들을 감소시키기 위해 디블록킹 동작을 수행할 수도 있다. 비디오 디코더 (30)는 CU의 코딩 블록들 (예컨대, 루마, Cb 및 Cr 코딩 블록들)을 디코딩 화상 버퍼 (162)에 저장할 수도 있다. 디코딩 화상 버퍼 (162)는 후속 모션 보상, 인트라 예측, 및 도 1의 디스플레이 디바이스 (32)와 같은 디스플레이 디바이스 상에의 프리젠테이션을 위해, 참조 화상들을 제공할 수도 있다. 예를 들어, 비디오 디코더 (30)는 디코딩 화상 버퍼 (162)에서의 블록들 (예컨대, 루마, Cb 및 Cr 블록들)에 기초하여, 다른 CUs의 PUs에 대해 인트라 예측 또는 인터 예측 동작들을 수행할 수도 있다. 이러한 방법으로, 비디오 디코더 (30)는 비트스트림으로부터, 유효한 루마 계수 블록의 변환 계수 레벨들을 추출하고, 변환 계수 레벨들을 역양자화하고, 변환을 변환 계수 레벨들에 적용하여 변환 블록을 생성하고, 그 변환 블록에 적어도 부분적으로 기초하여, 코딩 블록을 생성하며, 디스플레이를 위해 코딩 블록을 출력할 수도 있다.

[0348] 엘리먼트 (170)는 손실 압축을 위한 정상 코딩 경로를 나타낼 수도 있고, 엘리먼트 (172)는 역변환 및 역양자화 프로세스들을 바이패스하는 바이패스 코딩 경로를 나타낼 수도 있다. 이들 상이한 경로들은 단지 예시적이고, 무손실 코딩은 어떤 바이패스 없이 수행될 수도 있다. 일부 예들에서, 무손실 코딩 변환들 및 양자화를 배제한다. 다른 예들에서, 무손실 코딩은 변환들을 수행하고 오직 양자화 프로세스만을 배제한다. 또한 다른 예들에서, 무손실 코딩은 변환들 및 정량의 사용으로 구현될 수도 있고, 그러나 양자화 파라미터가 임의의 양자화 데이터 손실을 피할 수 있도록 선택될 수도 있다. 엘리먼트 (174)는 변환 스킵핑 모드에 사용될 수도 있는 경로의 일 예를 나타낸다. 변환 스kipping 모드에서, 잔여 데이터는 역양자화 유닛 (154)에 의해 역양자화될 수도 있고, 그러나 역변환 프로세싱 유닛 (156)의 역변환은 스kip될 수도 있다. 이들 및 다른 예들은 본 개시물의 범위 내이다.

[0349] 도 8a는 본 개시물의 하나 이상의 기법들에 따른, 비디오 인코더 (20)의 예시적인 동작을 예시하는 플로우차트이다. 본 개시물의 다른 예들에서, 도 8a의 동작과 유사한 동작들은 더 많거나, 더 적거나, 또는 상이한 액션들을 포함할 수도 있다. 더욱이, 일부 예들에서, 도 8a의 동작의 하나 이상의 액션들은 생략되거나 또는 재배열될 수도 있다. 예를 들어, 도 8a 및 도 8b에서, 파선들은 일부 예들에서 수행되지 않는 액션들을 나타낸다.

[0350] 도 8a의 예에서, 비디오 인코더 (20)는 예측 블록을 생성할 수도 있다 (200). 예측 블록을 생성하는 것의 일부로서, 비디오 인코더 (20)는 현재의 샘플의 DC 예측을 위해, 예측 블록의 현재의 로우에서 현재의 샘플의 좌측에 있는 무손실로 재구성된 샘플, 및 현재의 로우의 상부에 있는 예측 블록의 로우에 대한 무손실로 재구성된 샘플 중 적어도 하나를 사용할 수도 있다 (202). 일부 예들에서, 비디오 디코더 (30)는 예측 블록에서

의 샘플들을 수평 래스터 스캐닝 순서, 수직 래스터 스캐닝 순서, 대각선 스캐닝 순서, 또는 지그-재그 스캐닝 순서로 프로세싱한다. 이러한 예들에서, 예측 블록에서의 샘플들을 프로세싱하는 것은 샘플들을 무손실로 재구성하는 것 뿐만 아니라, 샘플들에 대한 DC 예측들을 결정하는 것을 포함할 수도 있다. 더욱이, 도 8a의 예에서, 비디오 인코더 (20)는 코딩 블록에서의 샘플과 예측 블록에서의 대응하는 샘플 사이의 차이와 동일한 값을 갖는 잔여 샘플들을 생성할 수도 있다.

[0351] 도 8b 는 본 개시물의 하나 이상의 기법들에 따른, 비디오 디코더 (30)의 예시적인 동작을 예시하는 플로우차트이다. 본 개시물의 다른 예들에서, 도 8b의 동작과 유사한 동작들 더 많거나, 더 적거나, 또는 상이한 액션들을 포함할 수도 있다. 더욱이, 일부 예들에서, 도 8b의 동작의 하나 이상의 액션들은 생략되거나 또는 재배열될 수도 있다.

[0352] 도 8b의 예에서, 비디오 디코더 (30)는 예측 블록을 생성할 수도 있다 (250). 예측 블록을 생성하는 것의 일부로서, 비디오 디코더 (30)는 현재의 샘플의 DC 예측을 위해, 예측 블록의 현재의 로우에서 현재의 샘플의 좌측에 있는 무손실로 재구성된 샘플 및 현재의 로우의 상부에 있는 예측 블록의 로우에 대한 무손실로 재구성된 샘플 중 적어도 하나를 사용할 수도 있다 (252). 일부 예들에서, 비디오 디코더 (30)는 예측 블록에서의 샘플들을 수평 래스터 스캐닝 순서, 수직 래스터 스캐닝 순서, 대각선 스캐닝 순서, 또는 지그-재그 스캐닝 순서로 프로세싱한다. 이러한 예들에서, 예측 블록에서의 샘플들을 프로세싱하는 것은 샘플들에 대한 DC 예측들을 결정하는 것을 포함할 수도 있다. 더욱이, 도 8b의 예에서, 비디오 디코더 (30)는 예측 블록의 샘플들을 대응하는 잔여 샘플들에 추가함으로써 코딩 블록을 재구성할 수도 있다 (254).

[0353] 도 9a는 본 개시물의 하나 이상의 기법들에 따른, 비디오 인코더 (20)의 예시적인 동작을 예시하는 플로우차트이다. 본 개시물의 다른 예들에서, 도 9a의 동작과 유사한 동작들 더 많거나, 더 적거나, 또는 상이한 액션들을 포함할 수도 있다. 더욱이, 일부 예들에서, 도 9a의 동작의 하나 이상의 액션들은 생략되거나 또는 재배열될 수도 있다. 도 9a의 예는 도 6에 나타낸 구성요소들을 참조하여 설명된다. 그러나, 도 9a의 동작은 도 6에 나타낸 것 이외의 비디오 인코더들의 구성요소들 및 유형들에 의해 수행될 수도 있다.

[0354] 도 9a의 예에 나타낸 바와 같이, 비디오 인코더 (20)의 잔여 생성 유닛 (102)은 잔여 값들의 블록을 생성할 수도 있다 (350). 이 예에서, 잔여 값들의 블록은 변환 스킵 블록이다. 블록은 원래 샘플 값들과 인트라 예측을 사용하여 생성되는 예측 블록에서의 샘플 값들 사이의 차이들을 나타내는 잔여 값들을 포함하는 잔여 블록일 수도 있다. 비디오 코더는 블록의 각각의 로케이션 ( $i, j$ )에 대해, 도 9a의 나머지 액션들을 수행할 수도 있고, 여기서  $0 \leq i \leq (M - 1)$  및  $0 \leq j \leq (N - 1)$ 이고,  $M$ 은 블록의 높이이고  $N$ 은 블록의 폭이다.

[0355] 더욱이, 비디오 인코더 (20)의 잔여 생성 유닛 (102)은 블록에서 잔여 값  $r_{i,j}$ 에 대한 수정된 잔여 값  $\tilde{r}_{i,j}$ 을 결정할 수도 있다 (352). 블록이 수직 인트라 예측 모드, 또는 일부 예들에서, HEVC 작업 초안 10에서 정의된 바와 같은 22와 30 사이의 인트라 예측 모드를 사용하여 코딩되면,  $\tilde{r}_{i,j}$ 는 다음과 같이 정의되며:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i = 0, 0 \leq j \leq (N - 1) \\ r_{i,j} - Q(r_{(i-1),j}), & 1 \leq i \leq (M - 1), 0 \leq j \leq (N - 1) \end{cases}$$

[0356]  $Q(r_{(i-1),j})$ 는 잔여 값  $r_{i-1,j}$ 에 대한 (즉, 대응하는) 재구성된 잔여 값을 나타낸다. 블록이 수평 인트라 예

[0357] 측 모드, 또는, 일부 예들에서, 6과 14 사이의 인트라 예측 모드를 사용하여 코딩되면,  $\tilde{r}_{i,j}$ 는 다음과 같이 정의되며:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & 0 \leq i \leq (M - 1), j = 0 \\ r_{i,j} - Q(r_{i,(j-1)}) & 0 \leq i \leq (M - 1), 1 \leq j \leq (N - 1) \end{cases}$$

[0358]  $Q(r_{i,(j-1)})$ 은 잔여 값  $r_{i,j-1}$ 에 대한 재구성된 잔여 값을 나타낸다.

[0360] 게다가, 비디오 인코더 (20) 의 양자화 유닛 (106) 은 수정된 잔여 값  $\tilde{r}_{i,j}$  을 양자화하여, 양자화된 수정된 잔여 값  $Q(\tilde{r}_{i,j})$  을 생성할 수도 있다 (354). 비디오 인코더 (20) 는 양자화된 수정된 잔여 값  $Q(\tilde{r}_{i,j})$  을 비트스트림으로 시그널링할 수도 있다 (356). 예를 들어, 비디오 인코더 (20) 는 양자화된 수정된 잔여 값  $Q(\tilde{r}_{i,j})$  을 포함하는 하나 이상의 선택스 엘리먼트들을 생성할 수도 있다. 이 예에서, 비디오 인코더 (20) 의 엔트로피 인코딩 유닛 (118) 은 하나 이상의 선택스 엘리먼트들을 엔트로피 인코딩하고, 최종 데이터를 비트스트림에 포함시킬 수도 있다.

[0361] 더욱이, 도 9a 의 예에서, 비디오 인코더 (20) 는 재구성된 잔여 값  $Q(r_{i,j})$  를 계산할 수도 있다 (358). 일부 예들에서, 비디오 인코더 (20) 는 재구성된 잔여 값  $Q(r_{i,j})$  를 퍼드백 루프의 일부로서 계산하여, 추가적인 인트라 예측 또는 인터 예측에서의 사용을 위해, 재구성된 샘플 값들을 결정할 수도 있다. 블록이 수직 인트라 예측 모드 (또는, 일부 예들에서, 근사 수직 인트라 예측 모드) 를 사용하여 코딩되면, 재구성된 잔여 값  $Q(r_{i,j})$  는 다음과 같이 정의될 수도 있다:

$$[0362] Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & i = 0, 0 \leq j \leq (N-1), \\ Q(\tilde{r}_{i,j}) + Q(r_{i-1,j}) & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases}$$

[0363] 블록이 수평 인트라 예측 모드 (또는, 일부 예들에서, 근사 수평 인트라 예측 모드) 를 사용하여 코딩되면,  $Q(r_{i,j})$  는 다음과 같이 정의될 수도 있다:

$$[0364] Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & 0 \leq i \leq (M-1), j = 0, \\ Q(\tilde{r}_{i,j}) + Q(r_{i,j-1}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}$$

[0365] 블록이 변환 스킵 블록이기 때문에, 역변환 프로세싱 유닛 (110) 은 잔여 값  $r_{i,j}$  에 역변환을 적용하지 않는다. 그러므로, 비디오 인코더 (20) 의 재구성 유닛 (112) 는 재구성된 잔여 값  $Q(r_{i,j})$  를 예측 값에 가산하여, 재구성된 샘플 값을 결정할 수도 있다 (360). 예측 값은 예측 블록에서의 샘플일 수도 있다. 비디오 인코더 (20) 의 예측 프로세싱 유닛 (100) 은 다른 블록들의 인트라 예측 또는 인터 예측을 위해 재구성된 샘플 값을 사용할 수도 있다 (362).

[0366] 도 9b 는 본 개시물의 하나 이상의 기법들에 따른, 비디오 디코더 (30) 의 예시적인 동작을 예시하는 플로우차트이다. 본 개시물의 다른 예들에서, 도 9b 의 동작과 유사한 동작들 더 많거나, 더 적거나, 또는 상이한 액션들을 포함할 수도 있다. 더욱이, 일부 예들에서, 도 9b 의 동작의 하나 이상의 액션들은 생략되거나 또는 재배열될 수도 있다. 도 9b 의 예는 도 7 에 나타낸 구성요소들을 참조하여 설명된다. 그러나, 도 9b 의 동작은 도 7 에 나타낸 것과는 다른 비디오 디코더들의 구성요소들 및 유형들에 의해 수행될 수도 있다.

[0367] 비디오 디코더 (30) 는 변환 스킵 블록의 각각의 로케이션 ( $i, j$ ) 에 대해 도 9b 의 동작을 수행할 수도 있고, 여기서  $0 \leq i \leq (M-1)$  및  $0 \leq j \leq (N-1)$  이고, M 은 블록의 높이이고 N 은 블록의 폭이다. 블록은 원래 샘플 값들과 인트라 예측을 사용하여 생성되는 예측 블록에서의 샘플 값들 사이의 차이들을 나타내는 잔여 값들을 포함하는 잔여 블록일 수도 있다. 도 9a 의 예에 나타낸 바와 같이, 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (150) 은 비트스트림으로부터, 수정된 양자화된 잔여 값  $Q(\tilde{r}_{i,j})$  를 나타내는 하나 이상의 선택스 엘리먼트들을 획득할 수도 있다 (400). 엔트로피 디코딩 유닛 (150) 은 하나 이상의 선택스 엘리먼트들 중 일부 또는 모두를 엔트로피 디코딩할 수도 있다.

[0368] 더욱이, 도 9b 의 예에서, 비디오 디코더 (30) 의 역양자화 유닛 (154) 은 잔여 값  $r_{i,j}$  에 대한 재구성된 잔여 값  $Q(r_{i,j})$  를 계산할 수도 있다 (402). 일부 예들에서, 잔여 값  $r_{i,j}$  는 본 개시물에서 다른 어딘가에 설명된 바와 같은 비트-시프트된 잔여 값이다. 블록이 수직 인트라 예측 모드를 사용하여 코딩되면,  $Q(r_{i,j})$  는 다음과 같이 정의되며:

$$[0369] Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & i = 0, 0 \leq j \leq (N-1), \\ Q(\tilde{r}_{i,j}) + Q(r_{i-1,j}) & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases}$$

[0370]  $\tilde{Q}(\tilde{r}_{i,j})$  는 수정된 잔여 값  $\tilde{r}_{i,j}$  의 양자화된 버전을 나타내고, 수정된 잔여 값  $\tilde{r}_{i,j}$  는 잔여 값  $r_{i,j}$  의 수정된 버전이고,  $Q(r_{i-1,j})$  는 잔여 값  $r_{i,j}$  의 한 칼럼 좌측에 있는 잔여 값에 대한 재구성된 잔여 값이다. 블록이 수평 인트라 예측 모드를 사용하여 코딩되면,  $Q(r_{i,j})$  는 다음과 같이 정의되며:

$$[0371] Q(r_{i,j}) = \begin{cases} Q(\tilde{r}_{i,j}), & 0 \leq i \leq (M-1), j = 0, \\ Q(\tilde{r}_{i,j}) + Q(r_{i,j-1}) & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}$$

[0372]  $Q(r_{i,j-1})$  은 잔여 값  $r_{i,j}$  의 한 로우 상부에 있는 잔여 값에 대한 재구성된 잔여 값이다.

[0373] 블록이 수직 인트라 예측 모드를 사용하여 코딩되면, 수정된 잔여 값  $\tilde{r}_{i,j}$  는 다음과 같이 정의된다:

$$[0374] \tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i = 0, 0 \leq j \leq (N-1) \\ r_{i,j} - Q(r_{(i-1),j}), & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases}$$

[0375] 블록이 수평 인트라 예측 모드를 사용하여 코딩되면, 수정된 잔여 값  $\tilde{r}_{i,j}$  는 다음과 같이 정의된다:

$$[0376] \tilde{r}_{i,j} = \begin{cases} r_{i,j}, & 0 \leq i \leq (M-1), j = 0 \\ r_{i,j} - Q(r_{i,(j-1)}), & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}$$

[0377] 블록이 변환 스kip 블록이기 때문에, 역변환 프로세싱 유닛 (156)은 재구성된 잔여 값  $Q(r_{i,j})$ 에 역변환을 적용하지 않는다. 비디오 디코더 (30)의 재구성 유닛 (158)은 재구성된 잔여 값  $Q(r_{i,j})$ 를 예측 값에 가산하여, 샘플 값을 재구성할 수도 있다 (406). 예측 값은 예측 블록에서의 샘플일 수도 있다.

[0378] 도 10a는 본 개시물의 하나 이상의 기법들에 따른, 부호 데이터 은닉을 위한 예시적인 비디오 인코더 동작을 예시하는 플로우차트이다. 본 개시물의 다른 예들에서, 도 10a의 동작과 유사한 동작들은 더 많거나, 더 적거나, 또는 상이한 액션들을 포함할 수도 있다. 더욱이, 일부 예들에서, 도 10a의 동작의 하나 이상의 액션들은 생략되거나 또는 재배열될 수도 있다.

[0379] 도 10a의 예에서, 비디오 인코더 (20)는 비디오 데이터의 코딩된 표현을 형성하는 비트들의 시퀀스를 포함하는 비트스트림을 생성한다 (600). 비트스트림을 생성하는 것의 일부로서, 비디오 인코더 (20)는 현재 블록이 잔여 데이터에 대한 변환의 적용 없이 생성되고 현재 블록이 잔여 DPCM 기법이 사용되는 인트라 예측 모드를 사용하여 인트라 예측되면, 부호 데이터 은닉이 현재 블록에 대해 디스에이블된다고 결정할 수도 있다 (602).

도 10a의 맥락에서, 현재 블록은 잔여 DPCM 기법이 적용된 잔여 샘플들의 블록의 4x4 서브-블록일 수도 있다. 그 후에, 도 10a의 예에서, 비디오 인코더 (20)는 비트스트림을 출력할 수도 있다 (604).

[0380] 일부 예들에서, 부호 데이터 은닉이 현재 블록에 대해 디스에이블될 때, 비디오 인코더 (20)는 비트스트림으로, 현재 블록에서의 각각의 개별의 유효 잔여 값에 대해, 각각의 유효 잔여 값이 양 또는 음인지 여부를 나타내는 신택스 엘리먼트를 시그널링할 수도 있다. 이러한 예들에서, 부호 데이터 은닉이 현재 블록에 대해 디스에이블되지 않을 때, 비디오 인코더 (20)는 비트스트림으로, 현재 블록에서 적어도 하나의 유효 잔여 값 또는 변환 계수의 값이 양 또는 음인지 여부를 나타내는 신택스 엘리먼트를 시그널링하지 않을 수도 있다.

[0381] 도 10b는 본 개시물의 하나 이상의 기법들에 따른, 부호 데이터 은닉을 위한 예시적인 비디오 디코더 동작을 예시하는 플로우차트이다. 본 개시물의 다른 예들에서, 도 10b의 동작과 유사한 동작들은 더 많거나, 더 적거나, 또는 상이한 액션들을 포함할 수도 있다. 더욱이, 일부 예들에서, 도 10b의 동작의 하나 이상의

액션들은 생략되거나 또는 재배열될 수도 있다.

[0382] 도 10b 의 예에서, 비디오 디코더 (30) 는 비디오 데이터의 코딩된 표현을 형성하는 비트들의 시퀀스를 포함하는 비트스트림으로부터 신택스 엘리먼트들을 획득한다 (650). 비트스트림으로부터 신택스 엘리먼트들을 획득하는 것의 일부로서, 비디오 디코더 (30) 는, 현재 블록이 잔여 데이터에 대한 변환의 적용 없이 생성되고 현재 블록이 잔여 DPCM 기법이 사용되는 인트라 예측 모드를 사용하여 인트라 예측되면, 부호 데이터 은닉이 현재 블록에 대해 디스에이블된다고 결정할 수도 있다 (652). 도 10b 의 맥락에서, 현재 블록은 잔여 DPCM 기법이 적용된 잔여 샘플들의 블록의 4x4 서브-블록일 수도 있다. 그 후에, 도 10b 의 예에서, 비디오 디코더 (30) 는 비트스트림으로부터 획득된 신택스 엘리먼트들에 적어도 부분적으로 기초하여 비디오 데이터의 화상을 재구성할 수도 있다 (654).

[0383] 일부 예들에서, 부호 데이터 은닉이 현재 블록에 대해 디스에이블될 때, 비디오 디코더 (30) 는, 비트스트림으로부터, 현재 블록에서의 각각의 개별의 유효 잔여 값에 대해, 각각의 유효 잔여 값이 양 또는 음인지 여부를 나타내는 신택스 엘리먼트를 획득할 수도 있다. 이러한 예들에서, 부호 데이터 은닉이 현재 블록에 대해 디스에이블되지 않을 때, 비디오 디코더 (30) 는 비트스트림으로부터, 현재 블록에서 유효 잔여 값의 값이 양 또는 음인지 여부를 나타내는 신택스 엘리먼트를 획득하지 않을 수도 있다.

[0384] 다음 단락들은 본 개시물의 하나 이상의 기법들에 따른 예들의 제 1 시리즈를 제공한다.

[0385] 예 1. 비디오 데이터를 코딩하는 방법으로서, 상기 방법은 블록 내 다른 샘플 값들의 인트라 DC 예측을 수행하기 위해 비디오 데이터의 블록 내 하나 이상의 원래 샘플 값들을 사용하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

[0386] 예 2. 예 1 의 방법으로서, 상기 블록 내 다른 샘플 값들의 예측을 수행하기 위해 블록 내 하나 이상의 원래 샘플 값들을 사용하는 단계는 스캐닝 순서에서 이후에 발생하는 샘플 값들을 예측하기 위해 스캐닝 순서에서 이전에 발생하는 원래 샘플 값들을 사용하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

[0387] 예 3. 예 1 의 방법으로서, 상기 블록 내 다른 샘플 값들의 예측을 수행하기 위해 블록 내 하나 이상의 원래 샘플 값들을 사용하는 단계는 그 후에 스캐닝된 로우의 샘플 값들을 예측하기 위해 이전에 스캐닝된 로우의 원래 샘플 값들을 사용하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

[0388] 예 4. 예 1 의 방법으로서, 상기 블록 내 다른 샘플 값들의 예측을 수행하기 위해 블록 내 하나 이상의 원래 샘플 값들을 사용하는 단계는 샘플에 대한 샘플 값들을 예측하기 위해 샘플의 인과적 이웃들에 대응하는 원래 샘플 값들을 사용하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

[0389] 예 5. 예들 1 - 4 중 임의의 예의 방법으로서, 현재의 샘플  $P_{i,j}$ ,  $0 \leq i \leq (M-1)$ ,  $0 \leq j \leq (N-1)$  에 대한, DC 예측 값  $DC_{i,j}$  은  $DC_{i,j} = (P_{i,j-1} + P_{i-1,j+1}) \gg 1$  로서 계산되는, 비디오 데이터를 코딩하는 방법.

[0390] 예 6. 예들 1 - 4 중 임의의 예의 방법으로서, 현재의 샘플  $P_{i,j}$ ,  $0 \leq i \leq (M-1)$ ,  $0 \leq j \leq (N-1)$  에 대한, DC 예측 값  $DC_{i,j}$  은  $DC_{i,j} = (P_{i,j-1} + P_{i-1,j}) \gg 1$  로서 계산되는, 비디오 데이터를 코딩하는 방법.

[0391] 예 7. 예들 1 - 6 중 임의의 예의 방법으로서, DC 예측을 파이프라인되는 (pipelined) 방법으로 수행하는 단계를 더 포함하는, 비디오 데이터를 코딩하는 방법.

[0392] 예 8. 예 7 의 방법으로서, DC 예측을 위한 하나의 사이클 지연이 블록의 로우들 사이에 존재하는, 비디오 데이터를 코딩하는 방법.

[0393] 예 9. 예들 1 - 4 중 임의의 예의 방법으로서, DC 예측 값  $DC_{i,j}$ ,  $0 \leq i \leq (M-1)$ ,  $0 \leq j \leq (N-1)$  은  $DC_{i,j} = (P_{i,j-1} + P_{i-1,j-1})$  로서 계산되는, 비디오 데이터를 코딩하는 방법.

[0394] 예 10. 예 9 의 방법으로서,  $DC_{i,j+1}$  는  $P_{i,j}$  를 기다리지 않고  $DC_{i,j+1} = ((P_{i,j-1} + P_{i-1,j} - P_{i-1,j-1}) + r_{i,j} + P_{i-1,j+1} - P_{i-1,j})$  로서 계산될 수 있고, 여기서,  $r_{i,j}$  는 샘플  $P_{i,j}$  에 대한 예측 에러 잔여인, 비디오 데이터를 코딩하는 방법.

[0395] 예 11. 예들 1 - 4 중 임의의 예의 방법으로서, 좌측 샘플, 좌상부 샘플, 및 우상부 샘플 중 사용가능한 샘플들이 DC 예측을 위해 사용되는, 비디오 데이터를 코딩하는 방법.

[0396] 예 12. 예 11 의 방법으로서, DC 예측 값  $DC_{i,j}$ ,  $0 \leq i \leq (M-1)$ ,  $0 \leq j \leq (N-1)$  은  $DC_{i,j} = (P_{i,j-1} + P_{i-1,j} + P_{i-1,j-1} + P_{i-1,j+1})$

$P_{i,j+1} + P_{i-1,j} + P_{i-1,j-1} + P_{i-1,j+1}) \gg 2$  중 하나로 계산되는, 비디오 데이터를 코딩하는 방법.

[0397] 예 13. 예 12 의 방법으로서, 최종 칼럼에서의 샘플들 ( $j=(N-1), i>0$ )에 대해, 우상부 샘플은 사용불가능하고, 상부 및 우상부 샘플들 ( $P_{i-1,j}$  및  $P_{i-1,j+1}$ )은 동일한 값을 갖는, 비디오 데이터를 코딩하는 방법.

[0398] 예 14. 예 13 의 방법으로서, 상기 우상부 샘플은 DC 예측에 사용되지 않는, 비디오 데이터를 코딩하는 방법.

[0399] 예 15. 비디오 데이터를 코딩하는 방법으로서, 상기 방법은 변환 유닛 (TU) 사이즈보다 더 작은 블록 사이즈에 대해 DC 예측을 수행하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

[0400] 예 16. 예 15 의 방법으로서, TU 사이즈에 관계 없이, DC 예측은  $2\times 2$  블록 사이즈에 대해 수행되고, 적어도 일부 TU 사이즈들은  $2\times 2$  블록 사이즈보다 더 큼, 비디오 데이터를 코딩하는 방법.

[0401] 예 17. 예 15 또는 16 의 방법으로서, 상기 샘플들  $P_{2i,2j}, P_{2i,2j+1}, P_{2i+1,2j}$ , 및  $P_{2i+1,2j+1}$ 에 대해, DC 예측 값들은  $(P_{2i-1,2j} + P_{2i-1,2j+1} + P_{2i,2j-1} + P_{2i+1,2j-1} + 2) \gg 2$  또는  $(P_{2i-1,2j} + P_{2i-1,2j+1} + P_{2i,2j-1} + P_{2i+1,2j-1}) \gg 2$  중 하나로서 계산되고, 여기서,  $0 \leq i \leq ((M/2)-1)$ ,  $0 \leq j \leq ((N/2)-1)$  인, 비디오 데이터를 코딩하는 방법.

[0402] 예 18. 예 17 의 방법으로서, 상기 M 및 N 은 둘 다 짹수인, 비디오 데이터를 코딩하는 방법.

[0403] 예 19. 예 18 의 방법으로서, 4개의 샘플들이 별별로 프로세싱될 수 있는, 비디오 데이터를 코딩하는 방법.

[0404] 예 20. 비디오 데이터를 코딩하는 방법으로서, 상기 방법은 변환 유닛 (TU)의 사이즈에 관계없이  $2\times 2$  블록 사이즈에 대해 DC 예측을 수행하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

[0405] 예 21. 비디오 데이터를 코딩하는 방법으로서, 상기 방법은 정상 DC 예측을 수행한 후 잔여들 사이의 상관을 사용하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

[0406] 예 22. 예 21 의 방법으로서,  $r_{i,j}$ ,  $0 \leq i \leq (M-1)$ ,  $0 \leq j \leq (N-1)$  은 HEVC 표준에 따라 규정된 바와 같이 DC 예측을 수행한 후 예측 잔여들을 나타내고, 상기 방법은  $s_{i,j} = r_{i,2j}$ ,  $s_{i,(j+(N/2))} = r_{i,2j} - r_{i,2j+1}$ ,  $0 \leq i \leq (M-1)$ ,  $0 \leq j \leq ((N/2)-1)$ 에 따라서 중간 값들  $s_{i,j}$ ,  $0 \leq i \leq (M-1)$ ,  $0 \leq j \leq (N-1)$  을 생성하는 단계를 더 포함하는, 비디오 데이터를 코딩하는 방법..

[0407] 예 23. 예 22 의 방법으로서, 상기 방법은  $t_{i,j} = s_{2i,j} - t_{(i+(M/2)),j}$ ,  $0 \leq i \leq ((M/2)-1)$ ,  $0 \leq j \leq (N-1)$ 에 따라서 수정된 잔여들,  $t_{i,j}$ ,  $0 \leq i \leq (M-1)$ ,  $0 \leq j \leq (N-1)$  을 생성하는 단계를 더 포함하는, 비디오 데이터를 코딩하는 방법.

[0408] 예 24. 예 23 의 방법으로서, 상기 수정된 잔여들,  $t_{i,j}$  는 엔트로피-코딩되는, 비디오 데이터를 코딩하는 방법.

[0409] 예 25. 예 21 의 방법으로서, 상기 방법은 디코딩 동안 수행되고,

[0410]  $s_{2i,j} = t_{i,j}$ ,  $s_{2i+1,j} = t_{i,j} - t_{(i+(M/2)),j}$ ,  $0 \leq i \leq ((M/2)-1)$ ,  $0 \leq j \leq (N-1)$  및  $r_{i,2j} = s_{i,j}$ ,  $r_{i,2j+1} = s_{i,j} - s_{i,(j+(N/2))}$ ,  $0 \leq i \leq (M-1)$ ,  $0 \leq j \leq ((N/2)-1)$  인, 비디오 데이터를 코딩하는 방법.

[0411] 예 26. 예 25 의 방법으로서, 상기 M 및 N 은 둘 다 짹수인, 비디오 데이터를 코딩하는 방법.

$$\begin{aligned} s_{i,j} &= r_{i,2j+1}, & s_{i,(j+(N/2))} &= P_{i,2j} - ((P_{i,2j-1} + P_{i,2j+1} + 1) \gg 1), \\ 0 \leq i < M, 0 \leq j < (N/2), \\ t_{i,j} &= s_{2i+1,j}, & t_{(i+(M/2)),j} &= s_{2i,j} - ((s_{2i+1,j} + s_{2i+1,j} + 1) \gg 1), \end{aligned}$$

[0412] 예 27. 예 21 의 방법으로서,  $0 \leq i < M/2, 0 \leq j < N$ .

인, 비디오 데이터를 코딩하는 방법.

[0413] 예 28. 예들 1 - 27 의 임의의 조합의 방법.

[0414] 예 29. 예들 1 - 24 및 27 중 임의의 예 또는 이들의 조합들의 방법으로서, 상기 방법은 인코더에 의해 수행되고, 상기 코딩은 인코딩을 지칭하는, 비디오 데이터를 코딩하는 방법.

[0415] 예 30. 예들 1 - 21 및 25 - 27 중 임의의 예 또는 이들의 조합들의 방법으로서, 상기 방법은 디코더에 의해 수

행되고, 상기 코딩은 디코딩을 지칭하는, 비디오 테이터를 코딩하는 방법.

[0416] 예 31. 예들 1 - 27 중 임의의 예 또는 이들의 조합들의 방법을 수행하도록 구성된 시스템.

[0417] 예 32. 하나 이상의 프로세서들로 하여금, 예들 1 - 27 중 임의의 예 또는 이들의 조합들의 방법을 수행하게 하는 명령들을 저장하는 비일시적 컴퓨터 판독가능 저장 매체.

[0418] 예 33. 예들 1 - 24 및 27 중 임의의 예 또는 이들의 조합들의 방법을 수행하도록 구성된 비디오 인코딩 디바이스.

[0419] 예 34. 예들 1 - 21 및 25 - 27 중 임의의 예 또는 이들의 조합들의 방법을 수행하도록 구성된 비디오 디코딩 디바이스.

[0420] 예 35. 예들 1 - 24 및 27 중 임의의 예 또는 이들의 조합들의 방법의 단계들을 수행하는 수단을 포함하는 비디오 인코딩 디바이스.

[0421] 예 36. 예들 1 - 21 및 25 - 27 중 임의의 예 또는 이들의 조합들의 방법의 단계들을 수행하는 수단을 포함하는 비디오 디코딩 디바이스.

[0422] 다음 단락들은 본 개시물의 하나 이상의 기법들에 따른 예들의 제 2 시리즈를 제공한다.

[0423] 예 1. 비디오 데이터를 코딩하는 방법으로서, 상기 방법은 블록 내 다른 샘플 값들의 예측을 수행하기 위해 비디오 데이터의 블록 내 하나 이상의 원래 샘플 값들을 사용하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

[0424] 예 2. 예 1 의 방법으로서, 상기 블록 내 다른 샘플 값들의 예측을 수행하기 위해 블록 내 하나 이상의 원래 샘플 값들을 사용하는 단계는 다른 샘플 값들의 예측을 수행하기 위해 블록의 최종 로우 및 최종 칼럼에 대응하는 원래 샘플 값들을 사용하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

[0425] 예 3. 예 1 의 방법으로서, 상기 블록 내 다른 샘플 값들의 예측을 수행하기 위해 블록 내 하나 이상의 원래 샘플 값들을 사용하는 단계는 다른 샘플 값들의 예측을 수행하기 위해 블록의 제 1 로우 및 제 1 칼럼에 대응하는 원래 샘플 값들을 사용하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

[0426] 예 4. 예들 1 - 3중 임의의 예의 방법으로서, 도 4 는 다른 샘플 값들의 예측을 수행하는데 사용되는 샘플 값들의 샘플들 로케이션들을 예시하는, 비디오 데이터를 코딩하는 방법.

[0427] 예 5. 예들 1 - 4중 임의의 예의 방법으로서, 상기 방법은 무손실 코딩 모드에 대해 수행되는, 비디오 데이터를 코딩하는 방법.

[0428] 예 6. 예들 1 - 5중 임의의 예의 방법으로서, 상기 방법은 평면 코딩 모드에 대해 수행되는, 비디오 데이터를 코딩하는 방법.

[0429] 예 7. 예들 1 - 5중 임의의 예의 방법으로서, 상기 방법은 각도 인트라 코딩 모드에 대해 수행되는, 비디오 데이터를 코딩하는 방법.

[0430] 예 8. 예들 1 - 7중 임의의 예의 방법으로서, 상기 방법은 예측에 의해 생성되는 잔여 값들의 세트에 대해 회전 동작을 수행하는 단계; 및 잔여 값들의 회전된 세트에 대해 엔트로피 코딩을 수행하는 단계를 더 포함하는, 비디오 데이터를 코딩하는 방법.

[0431] 예 9. 예 8 의 방법으로서, 상기 잔여 값들의 세트는 변환된 값들인, 비디오 데이터를 코딩하는 방법.

[0432] 예 10. 예들 1 - 9중 임의의 예의 방법으로서, 원래 샘플 값들을 코딩하기 위해 예측 프로세스를 수행하는 단계를 더 포함하는, 비디오 데이터를 코딩하는 방법.

[0433] 예 11. 예들 1 - 10 의 임의의 조합의 방법.

[0434] 예 12. 예들 1 - 10 중 임의의 예 또는 이들의 조합들의 방법으로서, 상기 방법은 인코더에 의해 수행되고, 상기 코딩은 인코딩을 지칭하는, 비디오 테이터를 코딩하는 방법.

[0435] 예 13. 예들 1 - 10 중 임의의 예 또는 이들의 조합들의 방법으로서, 상기 방법은 디코더에 의해 수행되고, 상기 코딩은 디코딩을 지칭하는, 비디오 테이터를 코딩하는 방법.

[0436] 예 14. 예들 1 - 10 중 임의의 예 또는 이들의 조합들의 방법을 수행하도록 구성된 시스템.

- [0437] 예 15. 실행되는 경우, 하나 이상의 프로세서들로 하여금, 예들 1 - 10 중 임의의 예 또는 이들의 조합들의 방법을 수행하게 하는 명령들을 저장하는 비일시적 컴퓨터 판독가능 저장 매체.
- [0438] 예 16. 예들 1 - 10 중 임의의 예 또는 이들의 조합들의 방법을 수행하도록 구성된 비디오 인코딩 디바이스.
- [0439] 예 17. 예들 1 - 10 중 임의의 예 또는 이들의 조합들의 방법을 수행하도록 구성된 비디오 디코딩 디바이스.
- [0440] 예 18. 예들 1 - 10 중 임의의 예 또는 이들의 조합들의 방법의 단계들을 수행하는 수단을 포함하는, 비디오 인코딩 디바이스.
- [0441] 예 19. 예들 1 - 10 중 임의의 예 또는 이들의 조합들의 방법의 단계들을 수행하는 수단을 포함하는, 비디오 디코딩 디바이스.
- [0442] 예 20. 본 개시물에서 설명된 임의의 디바이스 또는 방법.
- [0443] 다음 단락들은 본 개시물의 하나 이상의 기법들에 따른, 예들의 제 3 시리즈를 제공한다.
- [0444] 예 1. 비디오 데이터를 코딩하는 방법으로서, 상기 방법은 잔여 샘플들의 수정된 어레이를 결정하는 단계; 수정된 잔여 샘플에 대해, 잔여 샘플의 역양자화된 버전을 결정하는 단계; 및 재구성된 값을 결정하기 위해 잔여 샘플의 역양자화된 버전을 예측 값에 가산하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.
- [0445] 예 2. 예 1의 방법으로서, DPCM 을 사용하여 역양자화된 잔여 값들을 코딩하는 단계를 더 포함하는, 비디오 데이터를 코딩하는 방법.
- [0446] 예 3. 예 1의 방법으로서, DPCM 을 사용하여 역양자화된 잔여 값들의 비트-시프트된 버전을 코딩하는 단계를 더 포함하는, 비디오 데이터를 코딩하는 방법.
- [0447] 예 4. 예들 1-3 중 임의의 예의 방법으로서, 상기 방법은 근사 수직 인트라-예측 모드에 대해 수행되는, 비디오 데이터를 코딩하는 방법.
- [0448] 예 5. 예들 1-3 중 임의의 예의 방법으로서, 상기 방법은 근사 수평 인트라-예측 모드에 대해 수행되는, 비디오 데이터를 코딩하는 방법.
- [0449] 예 6. 예들 1-5 중 임의의 예의 방법으로서, 부호 데이터 은닉을 디스에이블하는 단계를 더 포함하는, 비디오 데이터를 코딩하는 방법.
- [0450] 예 7. 예들 1-5 중 임의의 예의 방법으로서, 블록 유형에 기초하여, 부호 데이터 은닉을 디스에이블하는 단계를 더 포함하는, 비디오 데이터를 코딩하는 방법.
- [0451] 예 8. 예들 1-5 중 임의의 예의 방법으로서, 부호 데이터 은닉을 선택적으로 디스에이블하는 단계를 더 포함하는, 비디오 데이터를 코딩하는 방법.
- [0452] 예 9. 예들 1-8중 임의의 예의 방법으로서, 본 개시물에서 설명되는 임의의 기법을 더 포함하는, 비디오 데이터를 코딩하는 방법.
- [0453] 예 10. 비디오 데이터를 코딩하는 방법으로서, 상기 방법은 잔여 샘플들의 수정된 어레이를 결정하는 단계; 수정된 잔여 샘플에 대해, 잔여 샘플의 양자화된 버전을 결정하는 단계; 및 인코딩된 비트스트림으로 잔여 샘플의 양자화된 버전을 시그널링하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.
- [0454] 예 11. 예 10의 방법으로서, 상기 방법은 근사 수직 인트라-예측 모드에 대해 수행되는, 비디오 데이터를 코딩하는 방법.
- [0455] 예 12. 예 10의 방법으로서, 상기 방법은 근사 수평 인트라-예측 모드에 대해 수행되는, 비디오 데이터를 코딩하는 방법.
- [0456] 예 13. 예들 10-12 중 임의의 예의 방법으로서, 본 개시물에서 설명되는 임의의 기법을 더 포함하는, 비디오 데이터를 코딩하는 방법.
- [0457] 예 14. 예들 10-13 중 임의의 예 또는 이들의 조합들의 방법으로서, 상기 방법은 인코더에 의해 수행되고, 상기 코딩은 인코딩을 지칭하는, 비디오 데이터를 코딩하는 방법.
- [0458] 예 15. 예들 1 - 9 중 임의의 예 또는 이들의 조합들의 방법으로서, 상기 방법은 디코더에 의해 수행되고, 상기 코딩은 디코딩을 지칭하는, 비디오 데이터를 코딩하는 방법.

- [0459] 예 16. 예들 1 - 15 중 임의의 예 또는 이들의 조합들의 방법을 수행하도록 구성된 시스템.
- [0460] 예 17. 하나 이상의 프로세서들로 하여금, 예들 1 - 13 중 임의의 예 또는 이들의 조합들의 방법을 수행하게 하는 명령들을 저장하는 비일시적 컴퓨터 판독가능 저장 매체.
- [0461] 예 18. 예들 10 - 13 중 임의의 예 또는 이들의 조합들의 방법을 수행하도록 구성된 비디오 인코딩 디바이스.
- [0462] 예 19. 예들 1 - 9 중 임의의 예 또는 이들의 조합들의 방법을 수행하도록 구성된 비디오 디코딩 디바이스.
- [0463] 예 20. 예들 10-13 중 임의의 예 또는 이들의 조합들의 방법의 단계들을 수행하는 수단을 포함하는 비디오 인코딩 디바이스.
- [0464] 예 21. 예들 1 - 9 중 임의의 예 또는 이들의 조합들의 방법의 단계들을 수행하는 수단을 포함하는 비디오 디코딩 디바이스.
- [0465] 예 22. 본 개시물을 포함하는 본 개시물에서 설명되는 임의의 디바이스 또는 방법.
- [0466] 다음 단락들은 본 개시물의 하나 이상의 기법들에 따른 예들의 제 4 시리즈를 제공한다.
- [0467] 예 1. 비디오 데이터를 디코딩하는 방법으로서, 상기 방법은 무손실 코딩 및 인트라 예측을 사용하여 인코딩된 비디오 데이터의 블록을 수신하는 단계; 잔여 차분 펠스 코드 변조 (DPCM) 프로세스에 따라서 비디오 데이터의 무손실로 코딩된 블록으로부터 잔여 샘플들을 재구성하는 단계; 및 재구성된 비디오 샘플들을 생성하기 위해 잔여 샘플들을 사용하여 인트라 예측 모드에 따라서 인트라 예측을 수행하는 단계를 포함하고, 상기 인트라 예측 모드는 수직 인트라 예측 모드 및 수평 인트라 예측 모드 중 하나가 아닌, 비디오 데이터를 디코딩하는 방법.
- [0468] 예 2. 예 1 의 방법으로서, 상기 인트라 예측 모드는 근사-수직 인트라 예측 모드 및 근사-수평 인트라 예측 모드 중 하나인, 비디오 데이터를 디코딩하는 방법.
- [0469] 예 3. 예 2 의 방법으로서, HEVC 작업 초안 9 에 설명된 인트라 모드들과 같이, 상기 근사-수직 인트라 예측 모드는 인트라 예측 모드들 22 내지 30 중 하나이고, 상기 근사-수평 인트라 예측 모드는 인트라 예측 모드들 6 내지 14 중 하나인, 비디오 데이터를 디코딩하는 방법.
- [0470] 예 4. 예 2 의 방법으로서, 상기 잔여 DPCM 프로세스는 근사-수직 인트라 예측 모드들에 대한 수직 잔여 DPCM 프로세스이고, 상기 잔여 DPCM 프로세스는 근사-수평 인트라 예측 모드들에 대한 수평 잔여 DPCM 프로세스인, 비디오 데이터를 디코딩하는 방법.
- [0471] 예 5. 예 1 의 방법으로서, 상기 인트라 예측 모드는 대각선 우하방 인트라 예측 모드이고, 상기 잔여 DPCM 프로세스에 따라서 잔여 샘플들을 재구성하는 단계는 방정식
- $$r_{i,j} = \begin{cases} \tilde{r}_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ \tilde{r}_{i,j}, & 1 \leq i \leq (M-1), j=0 \\ \tilde{r}_{i,j} + r_{i-1,j-1}, & 1 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}$$
- 에 따라서 잔여 샘플들을 재구성하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.
- [0472] 여기서,  $r$  은 재구성된 잔여 샘플이고,  $\tilde{r}$  는 비디오 데이터의 무손실로 코딩된 블록의 샘플이고,  $M$  및  $N$  은 비디오 데이터의 블록의 사이즈를 정의하며,,  $i$  및  $j$  는 비디오 데이터의 블록 내 샘플의 로케이션을 정의하는, 비디오 데이터를 디코딩하는 방법.
- [0473] 예 6. 예 1 의 방법으로서, 상기 인트라 예측 모드는 대각선 좌하방 인트라 예측 모드이고, 상기 잔여 DPCM 프로세스에 따라서 잔여 샘플들을 재구성하는 단계는 방정식
- $$r_{i,j} = \begin{cases} \tilde{r}_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ \tilde{r}_{i,j}, & 1 \leq i \leq (M-1), j=(N-1) \\ \tilde{r}_{i,j} + r_{i-1,j+1}, & 1 \leq i \leq (M-1), 0 \leq j \leq (N-2) \end{cases}$$
- 에 따라서 잔여 샘플들을 재구성하는 단계를 포함하고, 여기서,  $r$  은 재구성된 잔여 샘플이고,  $\tilde{r}$  는 비디오 데이터의 무손실로 코딩된 블록의 샘플이고,  $M$  및  $N$  은 비디오 데이터의 블록의 사이즈를 정의하며,,  $i$  및  $j$  는

비디오 데이터의 블록 내 샘플의 로케이션을 정의하는, 비디오 데이터를 디코딩하는 방법.

[0474] 예 7. 예 1 의 방법으로서, 상기 인트라 예측 모드는 대각선 우상방 인트라 예측 모드이고, 상기 잔여 DPCM 프로세스에 따라서 잔여 샘플들을 재구성하는 단계는 방정식

$$r_{i,j} = \begin{cases} \tilde{r}_{i,j}, & i=(M-1), 0 \leq j \leq (N-1) \\ \tilde{r}_{i,j}, & 0 \leq i \leq (M-2), j=0 \\ \tilde{r}_{i,j} + r_{i+1,j-1}, & 0 \leq i \leq (M-2), 1 \leq j \leq (N-1) \end{cases}$$

에 따라서 잔여 샘플들을 재구성하는 단계를 포함하고,

여기서,  $r$  은 재구성된 잔여 샘플이고,  $\tilde{r}$  는 비디오 데이터의 무손실로 코딩된 블록의 샘플이고, M 및 N 은 비디오 데이터의 블록의 사이즈를 정의하며,, i 및 j 는 비디오 데이터의 블록 내 샘플의 로케이션을 정의하는, 비디오 데이터를 디코딩하는 방법.

[0475] 예 8. 예들 1 내지 7 의 임의의 조합의 방법.

[0476] 예 9. 예 1 의 방법으로서, 상기 인트라-예측 모드는 근사-수직 인트라 예측 모드이고, 상기 방법은 비디오 데이터의 무손실로 코딩된 블록의 제 1 로우에 대해, 잔여 DPCM 프로세스에 따라서 잔여 샘플들을 재구성하지 않는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

[0477] 예 10. 예 1 의 방법으로서, 상기 인트라 예측 모드는 근사-수평 인트라 예측 모드이고, 상기 방법은 비디오 데이터의 무손실로 코딩된 블록의 제 1 칼럼에 대해, 잔여 DPCM 프로세스에 따라서 잔여 샘플들을 재구성하지 않는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

[0478] 예 11. 예 1 의 방법으로서, 상기 인트라 예측 모드는 DC 인트라 예측 모드 및 평면 인트라 예측 모드 중 하나이고, 상기 잔여 DPCM 프로세스들에 따라서 잔여 샘플들을 재구성하는 단계는 수직 잔여 DPCM 프로세스 및 수평 DPCM 프로세스 중 하나에 따라서 잔여 샘플들을 재구성하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

[0479] 예 12. 예 1 의 방법으로서, 상기 인트라 예측 모드는 DC 인트라 예측 모드 및 평면 인트라 예측 모드 중 하나이고, 상기 잔여 DPCM 프로세스들에 따라서 잔여 샘플들을 재구성하는 단계는 수직 잔여 DPCM 프로세스 및 수평 DPCM 프로세스 양쪽에 따라서 잔여 샘플들을 재구성하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

[0480] 예 13. 예 1 의 방법으로서, 상기 인트라 예측 모드는 평면 인트라 예측 모드이고, 상기 잔여 DPCM 프로세스들에 따라서 잔여 샘플들을 재구성하는 단계는 대각선 DPCM 프로세스, 수평 DPCM 프로세스, 및 수직 DPCM 프로세스에 따라서 잔여 샘플들을 재구성하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

[0481] 예 14. 비디오 데이터를 인코딩하는 방법으로서, 상기 방법은, 비디오 데이터의 블록을 수신하는 단계; 샘플들의 예측 블록 및 잔여 샘플들을 생성하기 위해 인트라 예측 모드에 따라서 비디오 데이터의 블록에 대해 인트라 예측을 수행하는 단계로서, 상기 인트라 예측 모드는 수직 인트라 예측 모드 및 수평 인트라 예측 모드 중 하나가 아닌, 상기 인트라 예측을 수행하는 단계; 및 잔여 차분 웨스 코드 변조 (DPCM) 프로세스를 사용하여 잔여 샘플들로부터 비디오 데이터의 무손실로 코딩된 블록을 생성하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

[0482] 예 15. 예 14 의 방법으로서, 상기 인트라 예측 모드는 근사-수직 인트라 예측 모드 및 근사-수평 인트라 예측 모드 중 하나인, 비디오 데이터를 디코딩하는 방법.

[0483] 예 16. 예 15 의 방법으로서, 상기 근사-수직 인트라 예측 모드는 인트라 예측 모드들 22 내지 30 중 하나이고, 상기 근사-수평 인트라 예측 모드는 인트라 예측 모드들 6 내지 14 중 하나인, 비디오 데이터를 디코딩하는 방법.

[0484] 예 17. 예 15 의 방법으로서, 상기 잔여 DPCM 프로세스는 근사-수직 인트라 예측 모드들에 대한 수직 잔여 DPCM 프로세스이고, 상기 잔여 DPCM 프로세스는 근사-수평 인트라 예측 모드들에 대한 수평 잔여 DPCM 프로세스인, 비디오 데이터를 디코딩하는 방법.

[0485] 예 18. 예 14 의 방법으로서, 상기 인트라 예측 모드는 대각선 우하방 인트라 예측 모드이고, 상기 잔여 DPCM 프로세스를 사용하여 잔여 샘플들로부터 비디오 데이터의 무손실로 코딩된 블록을 생성하는 단계는 방정식

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ r_{i,j}, & 1 \leq i \leq (M-1), j=0 \\ r_{i,j} - r_{i-1,j-1}, & 1 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}$$

에 따라서 비디오 데이터의 무손실로 코딩된 블록을

생성하는 단계를 포함하고, 여기서,  $r$  은 잔여 샘플이고,  $\tilde{r}$  는 비디오 데이터의 무손실로 코딩된 블록의 샘플이고,  $M$  및  $N$  은 비디오 데이터의 블록의 사이즈를 정의하며,,  $i$  및  $j$  는 비디오 데이터의 블록 내 샘플의 로케이션을 정의하는, 비디오 데이터를 디코딩하는 방법.

[0486]

예 19. 예 14 의 방법으로서, 상기 인트라 예측 모드는 대각선 우하방 인트라 예측 모드이고, 상기 잔여 DPCM 프로세스를 사용하여 잔여 샘플들로부터 비디오 데이터의 무손실로 코딩된 블록을 생성하는 단계는 방정식

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ r_{i,j}, & 1 \leq i \leq (M-1), j=(N-1) \\ r_{i,j} - r_{i-1,j+1}, & 1 \leq i \leq (M-1), 0 \leq j \leq (N-2) \end{cases}$$

에 따라서 비디오 데이터의 무손실로 코딩된 블록을 생성

하는 단계를 포함하고, 여기서,  $r$  은 잔여 샘플이고,  $\tilde{r}$  는 비디오 데이터의 무손실로 코딩된 블록의 샘플이고,  $M$  및  $N$  은 비디오 데이터의 블록의 사이즈를 정의하며,,  $i$  및  $j$  는 비디오 데이터의 블록 내 샘플의 로케이션을 정의하는, 비디오 데이터를 디코딩하는 방법.

[0487]

예 20. 예 14 의 방법으로서, 상기 인트라 예측 모드는 대각선 우하방 인트라 예측 모드이고, 상기 잔여 DPCM 프로세스를 사용하여 잔여 샘플들로부터 비디오 데이터의 무손실로 코딩된 블록을 생성하는 단계는 방정식

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i=(M-1), 0 \leq j \leq (N-1) \\ r_{i,j}, & 0 \leq i \leq (M-2), j=0 \\ r_{i,j} - r_{i+1,j-1}, & 0 \leq i \leq (M-2), 1 \leq j \leq (N-1) \end{cases}$$

에 따라서 비디오 데이터의 무손실로 코딩된 블록을 생성

하는 단계를 포함하고, 여기서,  $r$  은 잔여 샘플이고,  $\tilde{r}$  는 비디오 데이터의 무손실로 코딩된 블록의 샘플이고,  $M$  및  $N$  은 비디오 데이터의 블록의 사이즈를 정의하며,,  $i$  및  $j$  는 비디오 데이터의 블록 내 샘플의 로케이션을 정의하는, 비디오 데이터를 디코딩하는 방법.

[0488]

예 21. 예들 14 내지 20 의 임의의 조합의 방법.

[0489]

예 22. 예 14 의 방법으로서, 상기 인트라-예측 모드는 근사-수직 인트라 예측 모드이고, 상기 방법은 비디오 데이터의 블록의 제 1 로우에 대해 잔여 DPCM 프로세스를 사용하여 비디오 데이터의 무손실로 코딩된 블록을 생성하지 않는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

[0490]

예 23. 예 14 의 방법으로서, 상기 인트라 예측 모드는 근사-수평 인트라 예측 모드이고, 상기 방법은 비디오 데이터의 블록의 제 1 칼럼에 대해 잔여 DPCM 프로세스를 사용하여 비디오 데이터의 무손실로 코딩된 블록을 생성하지 않는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

[0491]

예 24. 예 14 의 방법으로서, 상기 인트라 예측 모드는 DC 인트라 예측 모드 및 평면 인트라 예측 모드 중 하나이고, 상기 잔여 DPCM 프로세스들을 사용하여 잔여 샘플들로부터 비디오 데이터의 무손실로 코딩된 블록을 생성하는 단계는 수직 잔여 DPCM 프로세스 및 수평 DPCM 프로세스 중 하나에 따라서 잔여 샘플들로부터 비디오 데이터의 무손실로 코딩된 블록을 생성하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

[0492]

예 25. 예 14 의 방법으로서, 상기 인트라 예측 모드는 DC 인트라 예측 모드 및 평면 인트라 예측 모드 중 하나이고, 상기 잔여 DPCM 프로세스들을 사용하여 잔여 샘플들로부터 비디오 데이터의 무손실로 코딩된 블록을 생성하는 단계는 수직 잔여 DPCM 프로세스 및 수평 DPCM 프로세스 양쪽에 따라서 잔여 샘플들로부터 비디오 데이터의 무손실로 코딩된 블록을 생성하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

[0493]

예 26. 예 14 의 방법으로서, 상기 인트라 예측 모드는 평면 인트라 예측 모드이고, 상기 잔여 DPCM 프로세스들을 사용하여 잔여 샘플들로부터 비디오 데이터의 무손실로 코딩된 블록을 생성하는 단계는 대각선 DPCM 프로세스, 수평 DPCM 프로세스, 및 수직 DPCM 프로세스에 따라서 잔여 샘플들로부터 비디오 데이터의 무손실로 코딩된 블록을 생성하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

[0494] 예 27. 비디오 데이터를 디코딩하도록 구성된 장치로서, 상기 장치는 무손실 코딩 및 인트라 예측을 사용하여 인코딩된 비디오 데이터의 블록을 수신하는 수단; 잔여 차분 필스 코드 변조 (DPCM) 프로세스에 따라서 비디오 데이터의 무손실로 코딩된 블록으로부터 잔여 샘플들을 재구성하는 수단; 및 재구성된 비디오 샘플들을 생성하기 위해 잔여 샘플들을 사용하여 인트라 예측 모드에 따라서 인트라 예측을 수행하는 수단을 포함하고, 상기 인트라 예측 모드는 수직 인트라 예측 모드 및 수평 인트라 예측 모드 중 하나가 아닌, 비디오 데이터를 디코딩하도록 구성된 장치.

[0495] 예 28. 예 27 의 장치로서, 상기 인트라 예측 모드는 근사-수직 인트라 예측 모드 및 근사-수평 인트라 예측 모드 중 하나인, 비디오 데이터를 디코딩하도록 구성된 장치.

[0496] 예 29. 예 28 의 장치로서, 상기 근사-수직 인트라 예측 모드는 인트라 예측 모드들 22 내지 30 중 하나이고, 상기 근사-수평 인트라 예측 모드는 인트라 예측 모드들 6 내지 14 중 하나인, 비디오 데이터를 디코딩하도록 구성된 장치.

[0497] 예 30. 예 28 의 장치로서, 상기 잔여 DPCM 프로세스는 근사-수직 인트라 예측 모드들에 대한 수직 잔여 DPCM 프로세스이고, 상기 잔여 DPCM 프로세스는 근사-수평 인트라 예측 모드들에 대한 수평 잔여 DPCM 프로세스인, 비디오 데이터를 디코딩하도록 구성된 장치.

[0498] 예 31. 예 27 의 장치로서, 상기 인트라 예측 모드는 대각선 우하방 인트라 예측 모드이고, 상기 잔여 DPCM 프로세스에 따라서 잔여 샘플들을 재구성하는 수단은 방정식

$$r_{i,j} = \begin{cases} \tilde{r}_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ \tilde{r}_{i,j}, & 1 \leq i \leq (M-1), j=0 \\ \tilde{r}_{i,j} + r_{i-1,j-1}, & 1 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}$$

에 따라서 잔여 샘플들을 재구성하는 수단을 포함하고,

여기서,  $r$  은 재구성된 잔여 샘플이고,  $\tilde{r}$  는 비디오 데이터의 무손실로 코딩된 블록의 샘플이고,  $M$  및  $N$  은 비디오 데이터의 블록의 사이즈를 정의하며,,  $i$  및  $j$  는 비디오 데이터의 블록 내 샘플의 로케이션을 정의하는, 비디오 데이터를 디코딩하는 방법.

[0499] 예 32. 예 27 의 장치로서, 상기 인트라 예측 모드는 대각선 좌하방 인트라 예측 모드이고, 상기 잔여 DPCM 프

$$r_{i,j} = \begin{cases} \tilde{r}_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ \tilde{r}_{i,j}, & 1 \leq i \leq (M-1), j=(N-1) \\ \tilde{r}_{i,j} + r_{i-1,j+1}, & 1 \leq i \leq (M-1), 0 \leq j \leq (N-2) \end{cases}$$

로세스에 따라서 잔여 샘플들을 재구성하는 수단은 방정식

따라서 잔여 샘플들을 재구성하는 수단을 포함하고, 여기서,  $r$  은 재구성된 잔여 샘플이고,  $\tilde{r}$  는 비디오 데이터의 무손실로 코딩된 블록의 샘플이고,  $M$  및  $N$  은 비디오 데이터의 블록의 사이즈를 정의하며,,  $i$  및  $j$  는 비디오 데이터의 블록 내 샘플의 로케이션을 정의하는, 비디오 데이터를 디코딩하는 방법.

[0500] 예 33. 예 27 의 장치로서, 상기 인트라 예측 모드는 대각선 우상방 인트라 예측 모드이고, 상기 잔여 DPCM 프로세스에 따라서 잔여 샘플들을 재구성하는 수단은 방정식

$$r_{i,j} = \begin{cases} \tilde{r}_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ \tilde{r}_{i,j}, & 1 \leq i \leq (M-1), j=(N-1) \\ \tilde{r}_{i,j} + r_{i-1,j+1}, & 1 \leq i \leq (M-1), 0 \leq j \leq (N-2) \end{cases}$$

에 따라서 잔여 샘플들을 재구성하는 수단을 포함하고,

여기서,  $r$  은 재구성된 잔여 샘플이고,  $\tilde{r}$  는 비디오 데이터의 무손실로 코딩된 블록의 샘플이고,  $M$  및  $N$  은 비디오 데이터의 블록의 사이즈를 정의하며,  $i$  및  $j$  는 비디오 데이터의 블록 내 샘플의 로케이션을 정의하는, 비디오 데이터를 디코딩하는 방법.

[0501] 예 34. 예 27 내지 33 의 임의의 조합의 장치.

[0502] 예 35. 예 27 의 장치로서, 상기 인트라-예측 모드는 근사-수직 인트라 예측 모드이고, 상기 장치는, 비디오 데이터의 무손실로 코딩된 블록의 제 1 로우에 대해, 잔여 DPCM 프로세스에 따라서 잔여 샘플들을 재구성하지 않

는 수단을 더 포함하는, 비디오 데이터를 디코딩하도록 구성된 장치.

[0503] 예 36. 예 27 의 장치로서, 상기 인트라 예측 모드는 근사-수평 인트라 예측 모드이고, 상기 장치는, 비디오 데이터의 무손실로 코딩된 블록의 제 1 칼럼에 대해, 잔여 DPCM 프로세스에 따라서 잔여 샘플들을 재구성하지 않는 수단을 더 포함하는, 비디오 데이터를 디코딩하도록 구성된 장치.

[0504] 예 37. 예 27 의 장치로서, 상기 인트라 예측 모드는 DC 인트라 예측 모드 및 평면 인트라 예측 모드 중 하나이고, 상기 잔여 DPCM 프로세스들에 따라서 잔여 샘플들을 재구성하는 수단은 수직 잔여 DPCM 프로세스 및 수평 DPCM 프로세스 중 하나에 따라서 잔여 샘플들을 재구성하는 수단을 포함하는, 비디오 데이터를 디코딩하도록 구성된 장치.

[0505] 예 38. 예 27 의 장치로서, 상기 인트라 예측 모드는 DC 인트라 예측 모드 및 평면 인트라 예측 모드 중 하나이고, 상기 잔여 DPCM 프로세스들에 따라서 잔여 샘플들을 재구성하는 수단은 수직 잔여 DPCM 프로세스 및 수평 DPCM 프로세스 양쪽에 따라서 잔여 샘플들을 재구성하는 수단을 포함하는, 비디오 데이터를 디코딩하도록 구성된 장치.

[0506] 예 39. 예 27 의 장치로서, 상기 인트라 예측 모드는 평면 인트라 예측 모드이고, 상기 잔여 DPCM 프로세스들에 따라서 잔여 샘플들을 재구성하는 수단은 대각선 DPCM 프로세스, 수평 DPCM 프로세스, 및 수직 DPCM 프로세스에 따라서 잔여 샘플들을 재구성하는 수단을 포함하는, 비디오 데이터를 디코딩하도록 구성된 장치.

[0507] 예 40. 비디오 데이터를 인코딩하도록 구성된 장치로서, 상기 장치는, 비디오 데이터의 블록을 수신하는 수단; 잔여 샘플들을 생성하기 위해 인트라 예측 모드에 따라서 비디오 데이터의 블록에 대해 인트라 예측을 수행하는 수단으로서, 상기 인트라 예측 모드는 수직 인트라 예측 모드 및 수평 인트라 예측 모드 중 하나가 아닌, 상기 인트라 예측을 수행하는 수단; 및 잔여 차분 펄스 코드 변조 (DPCM) 프로세스를 사용하여 잔여 샘플들로부터 비디오 데이터의 무손실로 코딩된 블록을 생성하는 수단을 포함하는, 비디오 데이터를 인코딩하도록 구성된 장치.

[0508] 예 41. 예 40 의 장치로서, 상기 인트라 예측 모드는 근사-수직 인트라 예측 모드 및 근사-수평 인트라 예측 모드 중 하나인, 비디오 데이터를 인코딩하도록 구성된 장치.

[0509] 예 42. 예 41 의 장치로서, 상기 근사-수직 인트라 예측 모드는 인트라 예측 모드들 22 내지 30 중 하나이고, 상기 근사-수평 인트라 예측 모드는 인트라 예측 모드들 6 내지 14 중 하나인, 비디오 데이터를 인코딩하도록 구성된 장치.

[0510] 예 43. 예 41 의 장치로서, 상기 잔여 DPCM 프로세스는 근사-수직 인트라 예측 모드들에 대한 수직 잔여 DPCM 프로세스이고, 상기 잔여 DPCM 프로세스는 근사-수평 인트라 예측 모드들에 대한 수평 잔여 DPCM 프로세스인, 비디오 데이터를 인코딩하도록 구성된 장치.

[0511] 예 44. 예 40 의 장치로서, 상기 인트라 예측 모드는 대각선 우하방 인트라 예측 모드이고, 상기 잔여 DPCM 프로세스를 사용하여 잔여 샘플들로부터 비디오 데이터의 무손실로 코딩된 블록을 생성하는 수단은 방정식

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ r_{i,j}, & 1 \leq i \leq (M-1), j=0 \\ r_{i,j} - r_{i-1,j-1}, & 1 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}$$

에 따라서 비디오 데이터의 무손실로 코딩된 블록을 생성하는 수단을 포함하고, 여기서,  $r$  은 잔여 샘플이고,  $\tilde{r}$  는 비디오 데이터의 무손실로 코딩된 블록의 샘플이고,  $M$  및  $N$  은 비디오 데이터의 블록의 사이즈를 정의하며,,  $i$  및  $j$  는 비디오 데이터의 블록 내 샘플의 로케이션을 정의하는, 비디오 데이터를 디코딩하는 방법.

[0512] 예 45. 예 40 의 장치로서, 상기 인트라 예측 모드는 대각선 우하방 인트라 예측 모드이고, 상기 잔여 DPCM 프로세스를 사용하여 잔여 샘플들로부터 비디오 데이터의 무손실로 코딩된 블록을 생성하는 수단은 방정식

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i=0, 0 \leq j \leq (N-1) \\ r_{i,j}, & 1 \leq i \leq (M-1), j=(N-1) \\ r_{i,j} - r_{i-1,j+1}, & 1 \leq i \leq (M-1), 0 \leq j \leq (N-2) \end{cases}$$

에 따라서 비디오 데이터의 무손실로 코딩된 블록을 생성하는 수단을 포함하고, 여기서,  $r$  은 잔여 샘플이고,  $\tilde{r}$  는 비디오 데이터의 무손실로 코딩된 블록의 샘플이고,  $M$  및  $N$  은 비디오 데이터의 블록의 사이즈를 정의하며,,  $i$  및  $j$  는 비디오 데이터의 블록 내 샘플의 로케이션을

정의하는, 비디오 데이터를 디코딩하는 방법.

[0513] 예 46. 예 40 의 장치로서, 상기 인트라 예측 모드는 대각선 우하방 인트라 예측 모드이고, 상기 잔여 DPCM 프로세스를 사용하여 잔여 샘플들로부터 비디오 데이터의 무손실로 코딩된 블록을 생성하는 수단은 방정식

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i=(M-1), 0 \leq j \leq (N-1) \\ r_{i,j}, & 0 \leq i \leq (M-2), j=0 \\ r_{i,j} - r_{i+1,j-1}, & 0 \leq i \leq (M-2), 1 \leq j \leq (N-1) \end{cases}$$

에 따라서 비디오 데이터의 무손실로 코딩된 블록을 생성하는 수단을 포함하고, 여기서,  $r$  은 잔여 샘플이고,  $\tilde{r}$  는 비디오 데이터의 무손실로 코딩된 블록의 샘플이고,  $M$  및  $N$  은 비디오 데이터의 블록의 사이즈를 정의하며,,  $i$  및  $j$  는 비디오 데이터의 블록 내 샘플의 로케이션을 정의하는, 비디오 데이터를 디코딩하는 방법.

[0514] 예 47. 예들 40 내지 46 의 임의의 조합의 장치.

[0515] 예 48. 예 40 의 장치로서, 상기 인트라-예측 모드는 근사-수직 인트라 예측 모드이고, 상기 장치는, 비디오 데이터의 블록의 제 1 로우에 대해 잔여 DPCM 프로세스를 사용하여 비디오 데이터의 무손실로 코딩된 블록을 생성하지 않는 수단을 더 포함하는, 비디오 데이터를 인코딩하도록 구성된 장치.

[0516] 예 49. 예 40 의 장치로서, 상기 인트라 예측 모드는 근사-수평 인트라 예측 모드이고, 상기 장치는, 비디오 데이터의 블록의 제 1 칼럼에 대해 잔여 DPCM 프로세스를 사용하여 비디오 데이터의 무손실로 코딩된 블록을 생성하지 않는 수단을 더 포함하는, 비디오 데이터를 인코딩하도록 구성된 장치.

[0517] 예 50. 예 40 의 장치로서, 상기 인트라 예측 모드는 DC 인트라 예측 모드 및 평면 인트라 예측 모드 중 하나이고, 상기 잔여 DPCM 프로세스들을 사용하여 잔여 샘플들로부터 비디오 데이터의 무손실로 코딩된 블록을 생성하는 수단은 수직 잔여 DPCM 프로세스 및 수평 DPCM 프로세스 중 하나에 따라서 잔여 샘플들로부터 비디오 데이터의 무손실로 코딩된 블록을 생성하는 수단을 포함하는, 비디오 데이터를 인코딩하도록 구성된 장치.

[0518] 예 51. 예 40 의 장치로서, 상기 인트라 예측 모드는 DC 인트라 예측 모드 및 평면 인트라 예측 모드 중 하나이고, 상기 잔여 DPCM 프로세스들을 사용하여 잔여 샘플들로부터 비디오 데이터의 무손실로 코딩된 블록을 생성하는 수단은 수직 잔여 DPCM 프로세스 및 수평 DPCM 프로세스 양쪽에 따라서 잔여 샘플들로부터 비디오 데이터의 무손실로 코딩된 블록을 생성하는 수단을 포함하는, 비디오 데이터를 인코딩하도록 구성된 장치.

[0519] 예 52. 예 40 의 장치로서, 상기 인트라 예측 모드는 평면 인트라 예측 모드이고, 상기 잔여 DPCM 프로세스들을 사용하여 잔여 샘플들로부터 비디오 데이터의 무손실로 코딩된 블록을 생성하는 수단은 대각선 DPCM 프로세스, 수평 DPCM 프로세스, 및 수직 DPCM 프로세스에 따라서 잔여 샘플들로부터 비디오 데이터의 무손실로 코딩된 블록을 생성하는 수단을 포함하는, 비디오 데이터를 인코딩하도록 구성된 장치.

[0520] 예 53. 예들 1 내지 13 의 방법들의 임의의 조합을 수행하도록 구성된 비디오 디코더.

[0521] 예 54. 예들 14 내지 26 의 방법들의 임의의 조합을 수행하도록 구성된 비디오 인코더.

[0522] 예 55. 실행되는 경우, 비디오 데이터를 디코딩하도록 구성된 디바이스의 하나 이상의 프로세서들로 하여금, 예들 1 내지 13 의 방법들의 임의의 조합을 수행하게 하는 명령들을 저장하는 컴퓨터 관독가능 저장 매체.

[0523] 예 56. 실행되는 경우, 비디오 데이터를 인코딩하도록 구성된 디바이스의 하나 이상의 프로세서들로 하여금, 예들 14 내지 26 의 방법들의 임의의 조합을 수행하게 하는 명령들을 저장하는 컴퓨터 관독가능 저장 매체.

[0524] 다음 단락들은 본 개시물의 하나 이상의 기법들에 따른 예들의 제 5 시리즈를 제공한다.

[0525] 예 1. 비디오 데이터를 코딩하는 방법으로서, 상기 방법은 비디오 데이터의 블록의 수평 인트라 코딩을 위해 예측 샘플들을 생성하는 단계를 포함하고, 상기 비디오 데이터의 블록의 모든 칼럼에 대해, 예측 샘플들은 그레디언트 항을 포함하는, 비디오 데이터를 코딩하는 방법.

$$((P_{-1,j} - P_{-1,j-1}) \gg 1)$$

[0526] 예 2. 예 1 의 방법으로서, 초기 로우에서 예측 샘플들에 대한 그레디언트 항은 으로 주어지는, 비디오 데이터를 코딩하는 방법.

[0527] 예 3. 예들 1 - 2 중 임의의 예의 방법으로서, 상기  $P_{0,j}, 0 \leq j \leq N-1$ 에 대한 예측 샘플은  $\text{Clip}\left(P_{0,j-1} + ((P_{-1,j} - P_{-1,j-1}) \gg 1)\right)$ 인, 비디오 데이터를 코딩하는 방법.

[0528] 예 4. 예들 1 - 3 중 임의의 예의 방법으로서, 상기  $P_{i,j}, 0 \leq i \leq M-1, 0 \leq j \leq N-1$ 에 대한 예측 샘플들은  $\text{Clip}\left(P_{i,j-1} + ((P_{i-1,j} - P_{i-1,j-1}) \gg 1)\right)$ 으로 주어지는, 비디오 데이터를 코딩하는 방법.

[0529] 예 5. 예들 5 - 8 중 임의의 예의 방법으로서, 상기 방법은 무손실 수평 인트라 코딩에 적용되는, 비디오 데이터를 코딩하는 방법.

[0530] 예 6. 비디오 데이터를 코딩하는 방법으로서, 상기 방법은 비디오 데이터의 블록의 수직 인트라 코딩을 위해 예측 샘플들을 생성하는 단계를 포함하고, 상기 비디오 데이터의 블록의 모든 로우에 대해, 예측 샘플들은 그레디언트 항을 포함하는, 비디오 데이터를 코딩하는 방법.

[0531] 예 7. 예 6의 방법으로서, 초기 칼럼에서 예측 샘플들에 대한 그레디언트 항은  $((P_{i,-1} - P_{i-1,-1}) \gg 1)$ 으로 주어지는, 비디오 데이터를 코딩하는 방법.

[0532] 예 8. 예들 6 - 7 중 임의의 예의 방법으로서, 상기  $P_{i,j}, 0 \leq i \leq M-1, 0 \leq j \leq N-1$ 에 대한 예측 샘플은  $\text{Clip}\left(P_{i-1,0} + ((P_{i,-1} - P_{i-1,-1}) \gg 1)\right)$ 인, 비디오 데이터를 코딩하는 방법.

[0533] 예 9. 예들 6 - 8 중 임의의 예의 방법으로서, 상기  $P_{i,j}, 0 \leq i \leq M-1, 0 \leq j \leq N-1$ 에 대한 예측 샘플들은  $\text{Clip}\left(P_{i-1,j} + ((P_{i,j-1} - P_{i-1,j-1}) \gg 1)\right)$ 으로 주어지는, 비디오 데이터를 코딩하는 방법.

[0534] 예 10. 예들 6 - 9 중 임의의 예의 방법으로서, 상기 방법은 무손실 수직 인트라 코딩에 적용되는, 비디오 데이터를 코딩하는 방법.

[0535] 예 11. 예들 1 - 10의 임의의 조합의 방법.

[0536] 예 12. 예들 1 - 10 중 임의의 예 또는 이들의 조합들의 방법으로서, 상기 방법은 인코더에 의해 수행되고, 상기 코딩은 인코딩을 지칭하는, 비디오 데이터를 코딩하는 방법.

[0537] 예 13. 예들 1 - 10 중 임의의 예 또는 이들의 조합들의 방법으로서, 상기 방법은 디코더에 의해 수행되고, 상기 코딩은 디코딩을 지칭하는, 비디오 데이터를 코딩하는 방법.

[0538] 예 14. 예들 1 - 10 중 임의의 예 또는 이들의 조합들의 방법을 수행하도록 구성된 시스템.

[0539] 예 15. 실행되는 경우, 하나 이상의 프로세서들로 하여금, 예들 1 - 10 중 임의의 예 또는 이들의 조합들의 방법을 수행하게 하는 명령들을 저장하는 비일시적 컴퓨터 판독가능 저장 매체.

[0540] 예 16. 예들 1 - 10 중 임의의 예 또는 이들의 조합들의 방법을 수행하도록 구성된 비디오 인코딩 디바이스.

[0541] 예 17. 예들 1 - 10 중 임의의 예 또는 이들의 조합들의 방법을 수행하도록 구성된 비디오 디코딩 디바이스.

[0542] 예 18. 예들 1 - 10 중 임의의 예 또는 이들의 조합들의 방법의 단계들을 수행하는 수단을 포함하는 비디오 인코딩 디바이스.

[0543] 예 19. 예들 1 - 10 중 임의의 예 또는 이들의 조합들의 방법의 단계들을 수행하는 수단을 포함하는 비디오 디코딩 디바이스.

[0544] 하나 이상의 예들에서, 본원에서 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 상에 저장되거나 또는 컴퓨터 판독가능 매체를 통해 송신될 수도 있고, 하드웨어-기반의 프로세

성 유닛에 의해 실행될 수도 있다. 컴퓨터 관독가능 매체는 컴퓨터 관독가능 저장 매체들을 포함할 수도 있고, 이 컴퓨터 관독가능 저장 매체들은 데이터 저장 매체와 같은 유형의 매체, 또는 예컨대, 통신 프로토콜에 따라서 한 장소로부터 다른 장소로의 컴퓨터 프로그램의 전송을 용이하게 하는 임의의 매체를 포함한 통신 매체들에 대응한다. 이런 방법으로, 컴퓨터 관독가능 매체들은 일반적으로 (1) 비일시적 유형의 컴퓨터 관독가능 저장 매체, 또는 (2) 신호 또는 캐리어 파와 같은 통신 매체에 대응할 수도 있다. 데이터 저장 매체는 본 개시물에서 설명하는 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 제품은 컴퓨터 관독가능 매체를 포함할 수도 있다.

[0545] 일 예로서, 이에 한정하지 않고, 이런 컴퓨터 관독가능 저장 매체는 RAM, ROM, EEPROM, CD-ROM 또는 다른 광디스크 스토리지, 자기디스크 스토리지, 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 원하는 프로그램 코드를 명령들 또는 데이터 구조들의 형태로 저장하는데 사용될 수 있고 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 접속이 컴퓨터 관독가능 매체로 적절히 지정된다. 예를 들어, 동축 케이블, 광섬유 케이블, 연선, 디지털가입자 회선 (DSL), 또는 무선 기술들, 예컨대 적외선, 라디오, 및 마이크로파를 사용하여 명령들이 웹사이트, 서버, 또는 다른 원격 소스로부터 송신되는 경우, 동축 케이블, 광섬유 케이블, 연선, DSL, 또는 무선 기술들 예컨대 적외선, 라디오, 및 마이크로파가 그 매체의 정의에 포함된다. 그러나, 컴퓨터 관독가능 저장 매체 및 데이터 저장 매체는 접속부들, 반송파들, 신호들, 또는 다른 일시성 매체를 포함하지 않고, 그 대신, 비-일시성 유형의 저장 매체로 송신되는 것으로 해석되어야 한다.

디스크 (disk) 및 디스크 (disc) 는, 본원에서 사용할 때, 컴팩트 디스크 (CD), 레이저 디스크, 광 디스크, 디지털 다기능 디스크 (DVD), 플로피 디스크 및 Blu-ray 디스크를 포함하고, 디스크들 (disks) 은 데이터를 자기적으로 보통 재생하지만, 디스크들 (disks) 은 레이저로 데이터를 광학적으로 재생한다. 앞에서 언급한 것들의 결합들이 또한 컴퓨터 관독가능 매체들의 범위 내에 포함되어야 한다.

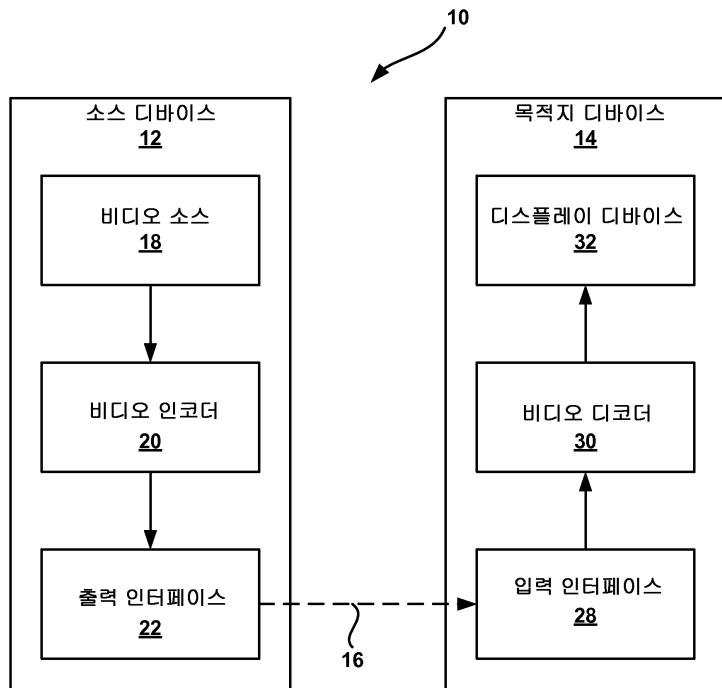
[0546] 명령들은 하나 이상의 디지털 신호 프로세서들 (DSPs), 범용 마이크로프로세서들, 주문형 집적회로들 (ASICs), 필드 프로그래밍가능 로직 어레이들 (FPGAs), 또는 다른 등가의 집적 또는 이산 로직 회로와 같은, 하나 이상의 프로세서들에 의해 실행될 수도 있다. 따라서, 용어 "프로세서" 는, 본원에서 사용될 때 전술한 구조 중 임의의 구조 또는 본원에서 설명하는 기법들의 구현에 적합한 임의의 다른 구조를 지칭할 수도 있다. 게다가, 일부 양태들에서, 본원에서 설명하는 기능 전용 하드웨어 및/또는 인코딩 및 디코딩을 위해 구성되는 소프트웨어 모듈들 내에 제공되거나, 또는 결합된 코덱에 포함될 수도 있다. 또한, 이 기법들은 하나 이상의 회로들 또는 로직 엘리먼트들로 전적으로 구현될 수 있다.

[0547] 본 개시물의 기법들은 무선 핸드셋, 집적 회로 (IC) 또는 IC들의 세트 (예컨대, 칩 세트) 를 포함한, 매우 다양한 디바이스들 또는 장치들로 구현될 수도 있다. 개시한 기법들을 수행하도록 구성되는 디바이스들의 기능적 양태들을 강조하기 위해서 여러 구성요소들, 모듈들, 또는 유닛들이 본 개시물에서 설명되지만, 상이한 하드웨어 유닛들에 의한 실현을 반드시 필요로 하지는 않는다. 대신, 위에서 설명한 바와 같이, 여러 유닛들이 코덱 하드웨어 유닛에 결합되거나 또는 적합한 소프트웨어 및/또는 펌웨어와 함께, 위에서 설명한 바와 같은 하나 이상의 프로세서들을 포함한, 상호작용하는 하드웨어 유닛들의 컬렉션으로 제공될 수도 있다.

[0548] 여러 예들이 설명되었다. 이들 및 다른 예들은 다음 청구항들의 범위 이내이다.

## 도면

## 도면1

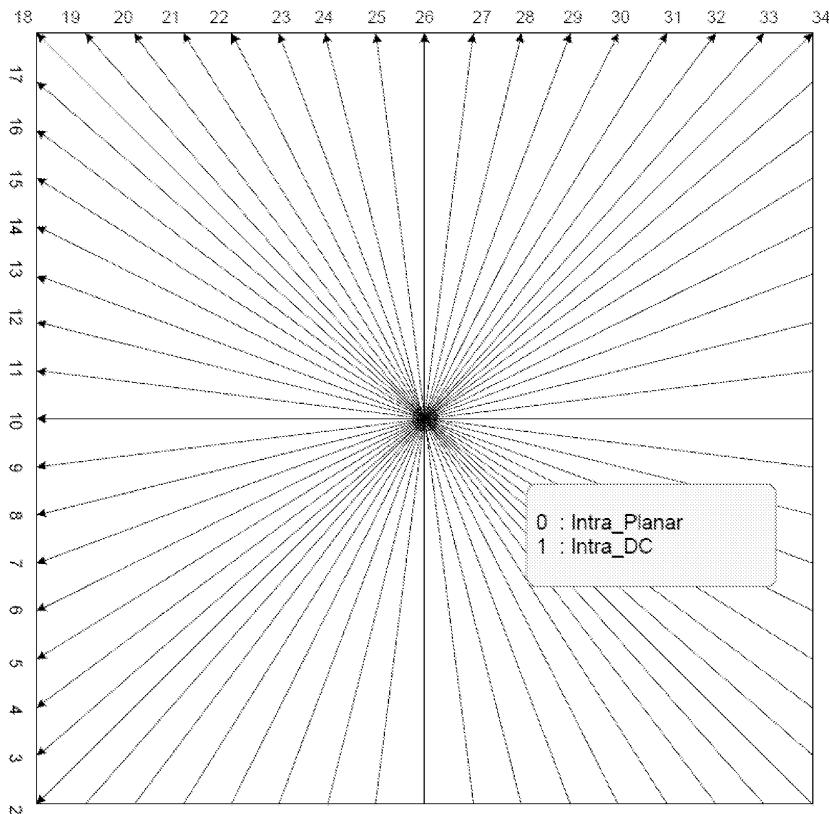


## 도면2

$P_{-1,-1}$	$P_{-1,0}$	$P_{-1,1}$	$\cdots$	$P_{-1,N-1}$	$P_{-1,N}$	$\cdots$		$P_{-1,2N-1}$
$P_{0,-1}$	$P_{0,0}$	$P_{0,1}$	$\cdots$	$P_{0,N-1}$				
$P_{1,-1}$	$P_{1,0}$				$\vdots$			
$\vdots$	$\vdots$							
$P_{M-1,-1}$	$P_{M-1,0}$	$\cdots$		$P_{M-1,N-1}$				
$P_{M,-1}$								
$\vdots$								
$P_{2M-1,-1}$								

본 도면에서 M=N=4

## 도면3



0 : Intra\_Planar  
1 : Intra\_DC

## 도면4

$P_{-1,-1}$	$P_{-1,0}$	$P_{-1,1}$	■ ■ ■	$P_{-1,N-1}$
$P_{0,-1}$	$P_{0,0}$	$P_{0,1}$	■ ■ ■	$P_{0,N-1}$
$P_{1,-1}$	$P_{1,0}$			■ ■ ■
■ ■ ■	■ ■ ■			
$P_{M-1,-1}$	$P_{M-1,0}$	■ ■ ■		$P_{M-1,N-1}$

본 도면에서 M=N=4

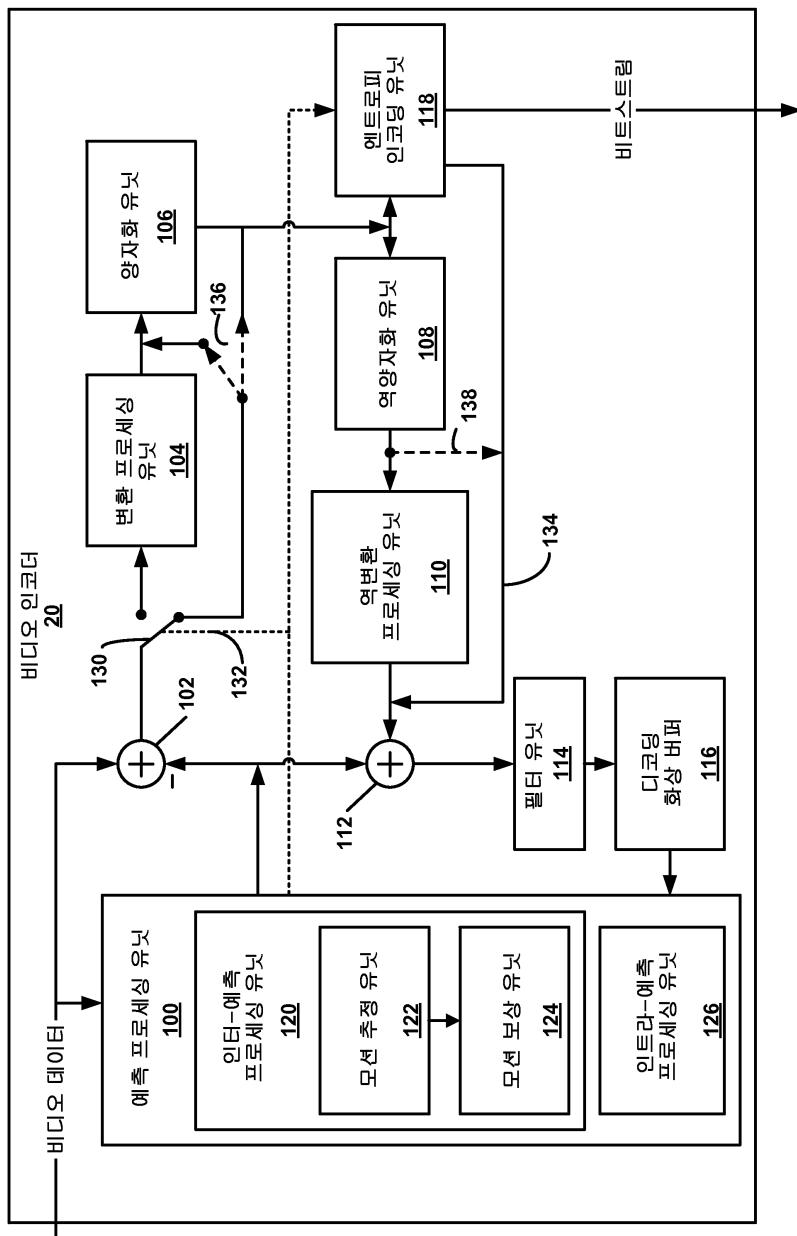
## 도면5a

$r_{0,0}$	$r_{0,1}$	$r_{0,2}$	...	$r_{0,(N-1)}$
$r_{1,0}$	$r_{1,1}$	$r_{1,2}$	...	$r_{1,(N-1)}$
$r_{2,0}$	$r_{2,1}$	$r_{2,2}$	...	$r_{2,(N-1)}$
...	...	...	...	...
$r_{(M-1),0}$	$r_{(M-1),1}$	$r_{(M-1),2}$	...	$r_{(M-1),(N-1)}$

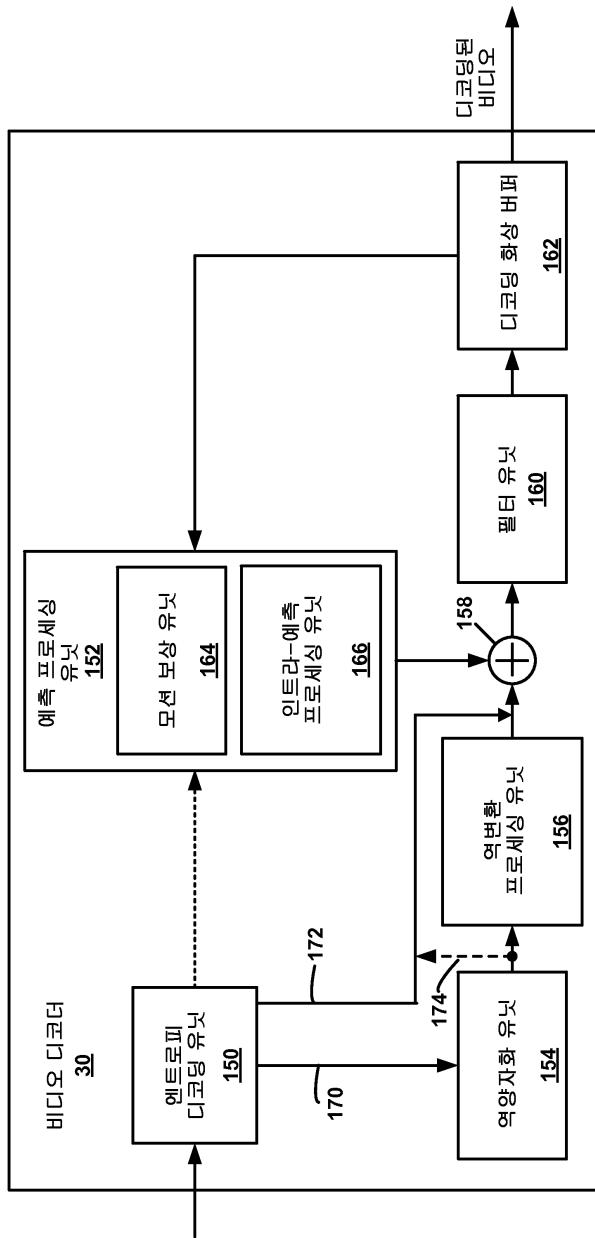
## 도면5b

$r_{0,0}$	$r_{0,1}$	$r_{0,2}$	...	$r_{0,(N-1)}$
$r_{1,0}$	$r_{1,1}$	$r_{1,2}$	...	$r_{1,(N-1)}$
$r_{2,0}$	$r_{2,1}$	$r_{2,2}$	...	$r_{2,(N-1)}$
...	...	...	...	...
$r_{(M-1),0}$	$r_{(M-1),1}$	$r_{(M-1),2}$	...	$r_{(M-1),(N-1)}$

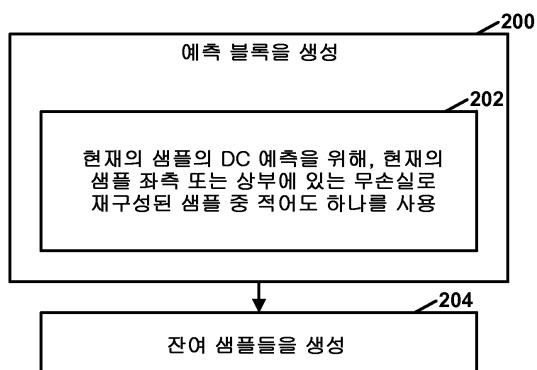
도면6



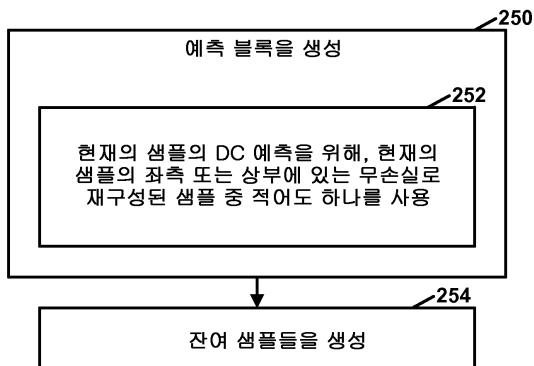
도면7



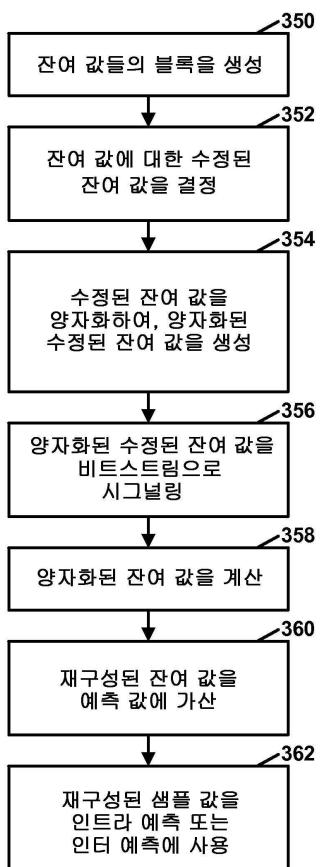
도면8a



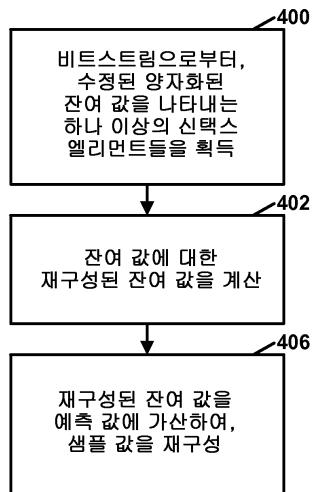
## 도면8b



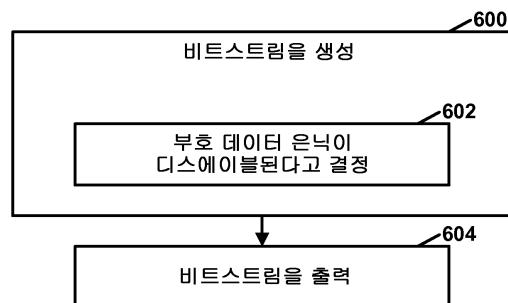
## 도면9a



## 도면9b



## 도면10a



## 도면10b

