



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2015-0015462  
(43) 공개일자 2015년02월10일

- (51) 국제특허분류(Int. Cl.)  
*H04N 19/51* (2014.01) *H04N 19/176* (2014.01)
- (21) 출원번호 10-2014-7032261
- (22) 출원일자(국제) 2013년04월19일  
심사청구일자 없음
- (85) 번역문제출일자 2014년11월18일
- (86) 국제출원번호 PCT/US2013/037442
- (87) 국제공개번호 WO 2013/159038  
국제공개일자 2013년10월24일
- (30) 우선권주장  
13/865,894 2013년04월18일 미국(US)  
61/636,554 2012년04월20일 미국(US)
- (71) 출원인  
퀄컴 인코포레이티드  
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
- (72) 발명자  
천 익  
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775  
장 리  
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775  
카르체비츠 마르타  
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
- (74) 대리인  
특허법인코리아나

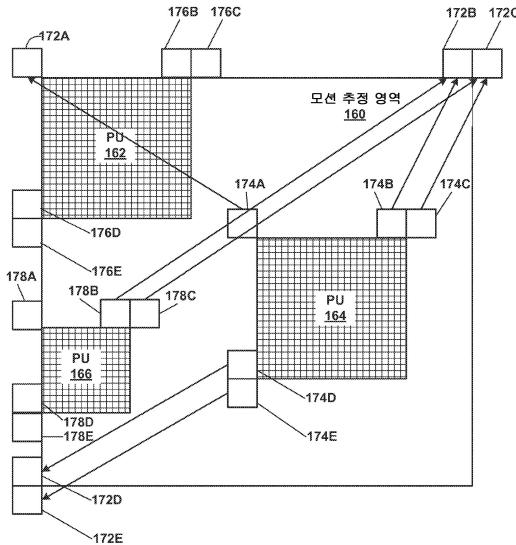
전체 청구항 수 : 총 50 항

(54) 발명의 명칭 비디오 코딩에 있어 인터-뷰 예측을 위한 디스패리티 벡터 생성

**(57) 요 약**

본 예에서, 비디오 코더 (예컨대, 비디오 인코더 또는 비디오 디코더) 는, 포함 블록 내에 있는 비디오 데이터의 현재 블록이 디스패리티 모션 벡터를 이용하여 코딩된다고 결정하고, 현재 블록에 이웃하는 블록이 또한 포함 블록 내에 있다는 결정에 기초하여, 후보 리스트에서 이웃하는 블록을 포함 블록 외부에 있고 포함 블록에 이웃하는 블록으로 대체하고, 후보 리스트에서 복수의 블록들 중 하나로부터 디스패리티 모션 벡터 예측자를 선택하고, 디스패리티 모션 벡터 예측자에 기초하여 디스패리티 모션 벡터를 코딩하도록 구성된다. 이러한 방법으로, 본 개시물의 기법들은 포함 블록 내 블록들이 별별로 코딩될 수 있게 할 수도 있다.

**대 표 도 - 도8**



## 특허청구의 범위

### 청구항 1

비디오 데이터를 디코딩하는 방법으로서,

포함 블록 내에 있는 비디오 데이터의 현재 블록이 디스패리티 모션 벡터를 이용하여 인코딩되는 것을 결정하는 단계;

상기 현재 블록에 이웃하는 블록이 또한 상기 포함 블록 내에 있다는 결정에 기초하여, 후보 리스트에서 상기 이웃하는 블록을 상기 포함 블록 외부에 있고 상기 포함 블록에 이웃하는 블록으로 대체하는 단계;

상기 후보 리스트에서 복수의 블록들 중 하나로부터 디스패리티 모션 벡터 예측자를 선택하는 단계; 및

상기 디스패리티 모션 벡터 예측자에 기초하여 상기 디스패리티 모션 벡터를 디코딩하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

### 청구항 2

제 1 항에 있어서,

상기 포함 블록은 모션 추정 영역 (MER) 을 포함하고,

상기 현재 블록은 코딩 유닛 (CU) 및 예측 유닛 (PU) 중 하나를 포함하는, 비디오 데이터를 디코딩하는 방법.

### 청구항 3

제 2 항에 있어서,

상기 대체하는 단계는, 상기 이웃하는 블록이 상기 현재 블록에 좌하측 이웃하는 블록에 대응하는 경우, 상기 현재 블록에 좌하측 이웃하는 블록을 상기 MER 에 좌하측 이웃하는 블록으로 대체하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

### 청구항 4

제 2 항에 있어서,

상기 대체하는 단계는, 상기 이웃하는 블록이 상기 현재 블록에 좌하측 이웃하는 블록에 대응하는 경우, 상기 MER 에 좌측 이웃하는 블록이 상기 현재 블록에 좌하측 이웃하는 블록과 동일한 수직 위치를 갖도록, 상기 현재 블록에 좌하측 이웃하는 블록을 상기 MER 에 좌측 이웃하는 블록으로 대체하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

### 청구항 5

제 2 항에 있어서,

상기 대체하는 단계는, 상기 이웃하는 블록이 상기 현재 블록에 우상측 이웃하는 블록에 대응하는 경우, 상기 현재 블록에 우상측 이웃하는 블록을 상기 MER 에 우상측 이웃하는 블록으로 대체하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

### 청구항 6

제 2 항에 있어서,

상기 대체하는 단계는, 상기 이웃하는 블록이 상기 현재 블록에 우상측 이웃하는 블록에 대응하는 경우, 상기 MER 에 상부 이웃하는 블록이 상기 현재 블록에 우상측 이웃하는 블록과 동일한 수평 위치를 갖도록, 상기 현재 블록에 우상측 이웃하는 블록을 상기 MER 에 상부 이웃하는 블록으로 대체하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

### 청구항 7

제 2 항에 있어서,

상기 대체하는 단계는, 상기 이웃하는 블록이 상기 현재 블록에 좌상측 이웃하는 블록에 대응하는 경우, 상기 현재 블록에 좌상측 이웃하는 블록을 상기 MER 에 좌상측 이웃하는 블록으로 대체하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

#### 청구항 8

제 2 항에 있어서,

상기 대체하는 단계는, 상기 이웃하는 블록이 상기 현재 블록에 좌상측 이웃하는 블록에 대응하는 경우, 상기 MER 에 좌측 이웃하는 블록이 상기 현재 블록에 좌상측 이웃하는 블록과 동일한 수직 위치를 갖도록, 상기 현재 블록에 좌상측 이웃하는 블록을 상기 MER 에 좌측 이웃하는 블록으로 대체하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

#### 청구항 9

제 2 항에 있어서,

상기 대체하는 단계는, 상기 이웃하는 블록이 상기 현재 블록에 좌상측 이웃하는 블록에 대응하는 경우, 상기 MER 에 상부 이웃하는 블록이 상기 현재 블록에 좌상측 이웃하는 블록과 동일한 수평 위치를 갖도록, 상기 현재 블록에 좌상측 이웃하는 블록을 상기 MER 에 상부 이웃하는 블록으로 대체하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

#### 청구항 10

제 1 항에 있어서,

상기 포함 블록은 코딩 유닛 (CU) 을 포함하고,

상기 현재 블록은 예측 유닛 (PU) 을 포함하는, 비디오 데이터를 디코딩하는 방법.

#### 청구항 11

제 1 항에 있어서,

상기 디스패리티 모션 벡터를 이용하여 비디오 데이터의 상기 현재 블록을 디코딩하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

#### 청구항 12

제 1 항에 있어서,

상기 디스패리티 모션 벡터를 디코딩하는 단계는,

모션 벡터 차이 값을 디코딩하는 단계; 및

상기 디스패리티 모션 벡터를 재생하기 위해 상기 모션 벡터 차이 값을 상기 디스패리티 모션 벡터 예측자에 가산하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

#### 청구항 13

비디오 데이터를 디코딩하는 디바이스로서,

포함 블록 내에 있는 비디오 데이터의 현재 블록이 디스패리티 모션 벡터를 이용하여 인코딩되는 것을 결정하고, 상기 현재 블록에 이웃하는 블록이 또한 상기 포함 블록 내에 있다는 결정에 기초하여, 후보 리스트에서 상기 이웃하는 블록을 상기 포함 블록 외부에 있고 상기 포함 블록에 이웃하는 블록으로 대체하고, 상기 후보 리스트에서 복수의 블록들 중 하나로부터 디스패리티 모션 벡터 예측자를 선택하고, 상기 디스패리티 모션 벡터 예측자에 기초하여 상기 디스패리티 모션 벡터를 디코딩하도록 구성된 비디오 디코더를 포함하는, 비디오 데이터를 디코딩하는 디바이스.

#### 청구항 14

제 13 항에 있어서,

상기 포함 블록은 모션 추정 영역 (MER) 을 포함하고,

상기 현재 블록은 코딩 유닛 (CU) 및 예측 유닛 (PU) 중 하나를 포함하는, 비디오 데이터를 디코딩하는 디바이스.

### 청구항 15

제 14 항에 있어서,

상기 비디오 디코더는, 상기 이웃하는 블록이 상기 현재 블록에 좌하측 이웃하는 블록에 대응하는 경우, 상기 현재 블록에 좌하측 이웃하는 블록을 상기 MER 에 좌하측 이웃하는 블록으로 대체하도록 구성되는, 비디오 데이터를 디코딩하는 디바이스.

### 청구항 16

제 14 항에 있어서,

상기 비디오 디코더는, 상기 이웃하는 블록이 상기 현재 블록에 좌하측 이웃하는 블록에 대응하는 경우, 상기 MER 에 좌측 이웃하는 블록이 상기 현재 블록에 좌하측 이웃하는 블록과 동일한 수직 위치를 갖도록, 상기 현재 블록에 좌하측 이웃하는 블록을 상기 MER 에 좌측 이웃하는 블록으로 대체하도록 구성되는, 비디오 데이터를 디코딩하는 디바이스.

### 청구항 17

제 14 항에 있어서,

상기 비디오 디코더는, 상기 이웃하는 블록이 상기 현재 블록에 우상측 이웃하는 블록에 대응하는 경우, 상기 현재 블록에 우상측 이웃하는 블록을 상기 MER 에 우상측 이웃하는 블록으로 대체하도록 구성되는, 비디오 데이터를 디코딩하는 디바이스.

### 청구항 18

제 14 항에 있어서,

상기 비디오 디코더는, 상기 이웃하는 블록이 상기 현재 블록에 우상측 이웃하는 블록에 대응하는 경우, 상기 MER 에 상부 이웃하는 블록이 상기 현재 블록에 우상측 이웃하는 블록과 동일한 수평 위치를 갖도록, 상기 현재 블록에 우상측 이웃하는 블록을 상기 MER 에 상부 이웃하는 블록으로 대체하도록 구성되는, 비디오 데이터를 디코딩하는 디바이스.

### 청구항 19

제 14 항에 있어서,

상기 비디오 디코더는, 상기 이웃하는 블록이 상기 현재 블록에 좌상측 이웃하는 블록에 대응하는 경우, 상기 현재 블록에 좌상측 이웃하는 블록을 상기 MER 에 좌상측 이웃하는 블록으로 대체하도록 구성되는, 비디오 데이터를 디코딩하는 디바이스.

### 청구항 20

제 14 항에 있어서,

상기 비디오 디코더는, 상기 이웃하는 블록이 상기 현재 블록에 좌상측 이웃하는 블록에 대응하는 경우, 상기 MER 에 좌측 이웃하는 블록이 상기 현재 블록에 좌상측 이웃하는 블록과 동일한 수직 위치를 갖도록, 상기 현재 블록에 좌상측 이웃하는 블록을 상기 MER 에 좌측 이웃하는 블록으로 대체하도록 구성되는, 비디오 데이터를 디코딩하는 디바이스.

### 청구항 21

제 14 항에 있어서,

상기 비디오 디코더는, 상기 이웃하는 블록이 상기 현재 블록에 좌상측 이웃하는 블록에 대응하는 경우, 상기 MER 에 상부 이웃하는 블록이 상기 현재 블록에 좌상측 이웃하는 블록과 동일한 수평 위치를 갖도록, 상기 현재 블록에 좌상측 이웃하는 블록을 상기 MER 에 상부 이웃하는 블록으로 대체하도록 구성되는, 비디오 데이터를 디코딩하는 디바이스.

### 청구항 22

제 13 항에 있어서,

상기 포함 블록은 코딩 유닛 (CU) 을 포함하고,

상기 현재 블록은 예측 유닛 (PU) 을 포함하는, 비디오 데이터를 디코딩하는 디바이스.

### 청구항 23

제 13 항에 있어서,

상기 비디오 디코더는 상기 디스패리티 모션 벡터를 이용하여 비디오 데이터의 상기 현재 블록을 디코딩하도록 더 구성되는, 비디오 데이터를 디코딩하는 디바이스.

### 청구항 24

제 13 항에 있어서,

상기 비디오 디코더는, 모션 벡터 차이 값을 디코딩하고, 상기 모션 벡터 차이 값을 상기 디스패리티 모션 벡터 예측자에 가산하여, 상기 디스패리티 모션 벡터를 재생하도록 구성되는, 비디오 데이터를 디코딩하는 디바이스.

### 청구항 25

제 13 항에 있어서,

상기 디바이스는,

집적 회로;

마이크로프로세서; 및

무선 통신 디바이스

중 적어도 하나를 포함하는, 비디오 데이터를 디코딩하는 디바이스.

### 청구항 26

비디오 데이터를 디코딩하는 디바이스로서,

포함 블록 내에 있는 비디오 데이터의 현재 블록이 디스패리티 모션 벡터를 이용하여 인코딩되는 것을 결정하는 수단;

상기 현재 블록에 이웃하는 블록이 또한 상기 포함 블록 내에 있다는 결정에 기초하여, 후보 리스트에서 상기 이웃하는 블록을 상기 포함 블록 외부에 있고 상기 포함 블록에 이웃하는 블록으로 대체하는 수단;

상기 후보 리스트에서 복수의 블록들 중 하나로부터 디스패리티 모션 벡터 예측자를 선택하는 수단; 및

상기 디스패리티 모션 벡터 예측자에 기초하여 상기 디스패리티 모션 벡터를 디코딩하는 수단을 포함하는, 비디오 데이터를 디코딩하는 디바이스.

### 청구항 27

제 26 항에 있어서,

상기 포함 블록은 모션 추정 영역 (MER) 을 포함하고,

상기 현재 블록은 코딩 유닛 (CU) 및 예측 유닛 (PU) 중 하나를 포함하는, 비디오 데이터를 디코딩하는 디바이스.

**청구항 28**

명령들을 저장하고 있는 컴퓨터 관독가능 저장 매체로서,  
 상기 명령들은, 실행되는 경우, 하나 이상의 프로세서들로 하여금,  
 포함 블록 내에 있는 비디오 데이터의 현재 블록이 디스패리티 모션 벡터를 이용하여 인코딩되는 것을 결정하게  
 하고;  
 상기 현재 블록에 이웃하는 블록이 또한 상기 포함 블록 내에 있다는 결정에 기초하여, 후보 리스트에서 상기  
 이웃하는 블록을 상기 포함 블록 외부에 있고 상기 포함 블록에 이웃하는 블록으로 대체하게 하며;  
 상기 후보 리스트에서 복수의 블록들 중 하나로부터 디스패리티 모션 벡터 예측자를 선택하게 하고;  
 상기 디스패리티 모션 벡터 예측자에 기초하여 상기 디스패리티 모션 벡터를 디코딩하게 하는, 컴퓨터 관독가능  
 저장 매체.

**청구항 29**

제 28 항에 있어서,  
 상기 포함 블록은 모션 추정 영역 (MER) 을 포함하고,  
 상기 현재 블록은 코딩 유닛 (CU) 및 예측 유닛 (PU) 중 하나를 포함하는, 컴퓨터 관독가능 저장 매체.

**청구항 30**

비디오 데이터를 인코딩하는 방법으로서,  
 디스패리티 모션 벡터를 이용하여 포함 블록 내의 비디오 데이터의 현재 블록을 인코딩하는 단계;  
 상기 현재 블록에 이웃하는 블록이 또한 상기 포함 블록 내에 있다는 결정에 기초하여, 후보 리스트에서 상기  
 이웃하는 블록을 상기 포함 블록 외부에 있고 상기 포함 블록에 이웃하는 블록으로 대체하는 단계;  
 상기 후보 리스트에서 복수의 블록들 중 하나로부터 디스패리티 모션 벡터 예측자를 선택하는 단계; 및  
 상기 디스패리티 모션 벡터 예측자에 기초하여 상기 디스패리티 모션 벡터를 인코딩하는 단계를 포함하는, 비디오  
 데이터를 인코딩하는 방법.

**청구항 31**

제 30 항에 있어서,  
 상기 포함 블록은 모션 추정 영역 (MER) 을 포함하고,  
 상기 현재 블록은 코딩 유닛 (CU) 및 예측 유닛 (PU) 중 하나를 포함하는, 비디오 데이터를 인코딩하는 방법.

**청구항 32**

제 31 항에 있어서,  
 상기 대체하는 단계는, 상기 이웃하는 블록이 상기 현재 블록에 좌하측 이웃하는 블록에 대응하는 경우, 상기  
 현재 블록에 좌하측 이웃하는 블록을 상기 MER 에 좌하측 이웃하는 블록으로 대체하는 단계를 포함하는, 비디오  
 데이터를 인코딩하는 방법.

**청구항 33**

제 31 항에 있어서,  
 상기 대체하는 단계는, 상기 이웃하는 블록이 상기 현재 블록에 좌하측 이웃하는 블록에 대응하는 경우, 상기  
 MER 에 좌측 이웃하는 블록이 상기 현재 블록에 좌하측 이웃하는 블록과 동일한 수직 위치를 갖도록, 상기 현재  
 블록에 좌하측 이웃하는 블록을 상기 MER 에 좌측 이웃하는 블록으로 대체하는 단계를 포함하는, 비디오 데이터  
 를 인코딩하는 방법.

#### 청구항 34

제 31 항에 있어서,

상기 대체하는 단계는, 상기 이웃하는 블록이 상기 현재 블록에 우상측 이웃하는 블록에 대응하는 경우, 상기 현재 블록에 우상측 이웃하는 블록을 상기 MER 에 우상측 이웃하는 블록으로 대체하는 단계를 포함하는, 비디오 데이터를 인코딩하는 방법.

#### 청구항 35

제 31 항에 있어서,

상기 대체하는 단계는, 상기 이웃하는 블록이 상기 현재 블록에 우상측 이웃하는 블록에 대응하는 경우, 상기 MER 에 상부 이웃하는 블록이 상기 현재 블록에 우상측 이웃하는 블록과 동일한 수평 위치를 갖도록, 상기 현재 블록에 우상측 이웃하는 블록을 상기 MER 에 상부 이웃하는 블록으로 대체하는 단계를 포함하는, 비디오 데이터를 인코딩하는 방법.

#### 청구항 36

제 31 항에 있어서,

상기 대체하는 단계는, 상기 이웃하는 블록이 상기 현재 블록에 좌상측 이웃하는 블록에 대응하는 경우, 상기 현재 블록에 좌상측 이웃하는 블록을 상기 MER 에 좌상측 이웃하는 블록으로 대체하는 단계를 포함하는, 비디오 데이터를 인코딩하는 방법.

#### 청구항 37

제 31 항에 있어서,

상기 대체하는 단계는, 상기 이웃하는 블록이 상기 현재 블록에 좌상측 이웃하는 블록에 대응하는 경우, 상기 MER 에 좌측 이웃하는 블록이 상기 현재 블록에 좌상측 이웃하는 블록과 동일한 수직 위치를 갖도록, 상기 현재 블록에 좌상측 이웃하는 블록을 상기 MER 에 좌측 이웃하는 블록으로 대체하는 단계를 포함하는, 비디오 데이터를 인코딩하는 방법.

#### 청구항 38

제 31 항에 있어서,

상기 대체하는 단계는, 상기 이웃하는 블록이 상기 현재 블록에 좌상측 이웃하는 블록에 대응하는 경우, 상기 MER 에 상부 이웃하는 블록이 상기 현재 블록에 좌상측 이웃하는 블록과 동일한 수평 위치를 갖도록, 상기 현재 블록에 좌상측 이웃하는 블록을 상기 MER 에 상부 이웃하는 블록으로 대체하는 단계를 포함하는, 비디오 데이터를 인코딩하는 방법.

#### 청구항 39

제 30 항에 있어서,

상기 포함 블록은 코딩 유닛 (CU) 을 포함하고,

상기 현재 블록은 예측 유닛 (PU) 을 포함하는, 비디오 데이터를 인코딩하는 방법.

#### 청구항 40

제 30 항에 있어서,

상기 디스패리티 모션 벡터를 인코딩하는 단계는,

상기 디스패리티 모션 벡터와 상기 디스패리티 모션 벡터 예측자 사이의 차이를 계산하는 단계; 및

상기 차이를 인코딩하는 단계를 포함하는, 비디오 데이터를 인코딩하는 방법.

#### 청구항 41

비디오 데이터를 인코딩하는 디바이스로서,

디스패리티 모션 벡터를 이용하여 포함 블록 내의 비디오 데이터의 현재 블록을 인코딩하고, 상기 현재 블록에 이웃하는 블록이 또한 상기 포함 블록 내에 있다는 결정에 기초하여, 후보 리스트에서 상기 이웃하는 블록을 상기 포함 블록 외부에 있고 상기 포함 블록에 이웃하는 블록으로 대체하며, 상기 후보 리스트에서 복수의 블록들 중 하나로부터 디스패리티 모션 벡터 예측자를 선택하고, 상기 디스패리티 모션 벡터 예측자에 기초하여 상기 디스패리티 모션 벡터를 인코딩하도록 구성된 비디오 인코더를 포함하는, 비디오 데이터를 인코딩하는 디바이스.

#### **청구항 42**

제 41 항에 있어서,

상기 포함 블록은 모션 추정 영역 (MER) 을 포함하고,

상기 현재 블록은 코딩 유닛 (CU) 및 예측 유닛 (PU) 중 하나를 포함하는, 비디오 데이터를 인코딩하는 디바이스.

#### **청구항 43**

제 42 항에 있어서,

상기 비디오 인코더는, 상기 이웃하는 블록이 상기 현재 블록에 좌하측 이웃하는 블록에 대응하는 경우, 상기 현재 블록에 좌하측 이웃하는 블록을 상기 MER 에 좌하측 이웃하는 블록으로 대체하도록 구성되는, 비디오 데이터를 인코딩하는 디바이스.

#### **청구항 44**

제 42 항에 있어서,

상기 비디오 인코더는, 상기 이웃하는 블록이 상기 현재 블록에 좌하측 이웃하는 블록에 대응하는 경우, 상기 MER 에 좌측 이웃하는 블록이 상기 현재 블록에 좌하측 이웃하는 블록과 동일한 수직 위치를 갖도록, 상기 현재 블록에 좌하측 이웃하는 블록을 상기 MER 에 좌측 이웃하는 블록으로 대체하도록 구성되는, 비디오 데이터를 인코딩하는 디바이스.

#### **청구항 45**

제 42 항에 있어서,

상기 비디오 인코더는, 상기 이웃하는 블록이 상기 현재 블록에 우상측 이웃하는 블록에 대응하는 경우, 상기 현재 블록에 우상측 이웃하는 블록을 상기 MER 에 우상측 이웃하는 블록으로 대체하도록 구성되는, 비디오 데이터를 인코딩하는 디바이스.

#### **청구항 46**

제 42 항에 있어서,

상기 비디오 인코더는, 상기 이웃하는 블록이 상기 현재 블록에 우상측 이웃하는 블록에 대응하는 경우, 상기 MER 에 상부 이웃하는 블록이 상기 현재 블록에 우상측 이웃하는 블록과 동일한 수평 위치를 갖도록, 상기 현재 블록에 우상측 이웃하는 블록을 상기 MER 에 상부 이웃하는 블록으로 대체하도록 구성되는, 비디오 데이터를 인코딩하는 디바이스.

#### **청구항 47**

제 42 항에 있어서,

상기 비디오 인코더는, 상기 이웃하는 블록이 상기 현재 블록에 좌상측 이웃하는 블록에 대응하는 경우, 상기 현재 블록에 좌상측 이웃하는 블록을 상기 MER 에 좌상측 이웃하는 블록으로 대체하도록 구성되는, 비디오 데이터를 인코딩하는 디바이스.

#### **청구항 48**

제 42 항에 있어서,

상기 비디오 인코더는, 상기 이웃하는 블록이 상기 현재 블록에 좌상측 이웃하는 블록에 대응하는 경우, 상기 MER 에 좌측 이웃하는 블록이 상기 현재 블록에 좌상측 이웃하는 블록과 동일한 수직 위치를 갖도록, 상기 현재 블록에 좌상측 이웃하는 블록을 상기 MER 에 좌측 이웃하는 블록으로 대체하도록 구성되는, 비디오 데이터를 인코딩하는 디바이스.

#### 청구항 49

제 42 항에 있어서,

상기 비디오 인코더는, 상기 이웃하는 블록이 상기 현재 블록에 좌상측 이웃하는 블록에 대응하는 경우, 상기 MER 에 상부 이웃하는 블록이 상기 현재 블록에 좌상측 이웃하는 블록과 동일한 수평 위치를 갖도록, 상기 현재 블록에 좌상측 이웃하는 블록을 상기 MER 에 상부 이웃하는 블록으로 대체하도록 구성되는, 비디오 데이터를 인코딩하는 디바이스.

#### 청구항 50

제 41 항에 있어서,

상기 디바이스는,

집적 회로;

마이크로프로세서; 및

무선 통신 디바이스

중 적어도 하나를 포함하는, 비디오 데이터를 인코딩하는 디바이스.

### 명세서

#### 기술 분야

[0001] 본 출원은 2012년 4월 20일자에 출원된 미국 가출원번호 제 61/636,554호의 이익을 주장하며, 이의 전체 내용들이 본원에 참고로 포함된다.

#### 기술 분야

[0003] 본 개시물은 비디오 코딩에 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, 태블릿 컴퓨터들, e-북 리더들, 디지털 카메라들, 디지털 레코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 무선 전화기들, 소위 "스마트폰들", 원격 화상회의 디바이스들, 비디오 스트리밍 디바이스들 등을 포함한, 광범위한 디바이스들에 포함될 수 있다. 디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, 파트 10, AVC (ADVANCED VIDEO CODING), 현재 개발중인 HEVC (HIGH EFFICIENCY VIDEO CODING) 표준, 및 이러한 표준들의 확장판들에 의해 정의된 표준들에서 설명되는 비디오 코딩 기법들을 같은, 비디오 코딩 기법들을 구현한다. HEVC는 ITU-T 비디오 코딩 전문가 그룹 (VCEG) 과 ISO/IEC 동화상 전문가 그룹 (MPEG) 의 비디오 코딩에 관한 합동 작업팀 (JCT-VC)에 의해 개발되고 있다. HEVC의 초안은 [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) 으로부터 입수 가능하다. H.264/AVC는 2010년 3월, ITU-T 권고안 H.264, "Advanced video coding for generic audiovisual services"에 설명되어 있는 스케일러블 비디오 코딩 (SVC) 및 멀티뷰 비디오 코딩 (MVC) 확장판을 포함한다. 비디오 디바이스들은 이러한 비디오 코딩 기법들을 구현함으로써, 디지털 비디오 정보를 좀 더 효율적으로 송신하거나, 수신하거나, 인코딩하거나, 디코딩하거나, 및/또는 저장할 수도 있다.

[0005] 비디오 코딩 기법들은 비디오 시퀀스들에 고유한 리던던시를 감소시키거나 또는 제거하기 위해, 공간 (인트라-화상) 예측 및/또는 시간 (인터-화상) 예측을 포함한다. 블록-기반 비디오 코딩에 있어, 비디오 슬라이스

(예컨대, 비디오 프레임 또는 비디오 프레임의 부분)은 비디오 블록들로 파티셔닝될 수도 있으며, 이 비디오 블록들은 또한 트리블록들, 코딩 유닛들 (CU들) 및/또는 코딩 노드들로서 지칭될 수도 있다. 화상의 인트라-코딩된 (I) 슬라이스에서 비디오 블록들은 동일한 화상에서 이웃하는 블록들에서의 참조 샘플들에 대한 공간 예측을 이용하여 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 이용할 수도 있다. 화상들은 프레임들로 지칭될 수 있으며, 참조 화상들은 참조 프레임들로서 지칭될 수도 있다.

[0006] 공간 또는 시간 예측은 코딩되는 블록에 대한 예측 블록을 초래한다. 잔차 데이터는 코딩되는 원래 블록과 예측 블록 사이의 픽셀 차이들을 나타낸다. 인터-코딩된 블록은 예측 블록을 형성하는 참조 샘플들의 블록을 가리키는 모션 벡터, 및 코딩된 블록과 예측 블록 사이의 차이를 나타내는 잔차 데이터에 따라서 인코딩된다. 인트라-코딩된 블록은 인트라-코딩 모드 및 잔차 데이터에 따라서 인코딩된다. 추가적인 압축을 위해, 잔차 데이터는 픽셀 도메인으로부터 변환 도메인으로 변환되어, 잔차 변환 계수들을 초래할 수도 있으며, 이 잔차 변환 계수는 그후 양자화될 수도 있다. 처음에 2차원 어레이로 배열된, 양자화된 변환 계수들은 변환 계수들의 1차원 벡터를 생성하기 위해 스캐닝될 수도 있으며, 엔트로피 코딩이 더욱 더 많은 압축을 달성하기 위해 적용될 수도 있다.

### 발명의 내용

#### 과제의 해결 수단

[0007] 일반적으로, 본 개시물은 차기 HEVC (HIGH EFFICIENCY VIDEO CODING) 표준의 기법들을 수행하도록 구성된 코덱들과 같은, 진보된 코덱들에 기초한, 멀티뷰 비디오 코딩과 관련된 기법들을 기술한다. 이러한 진보된 코덱들은 HEVC에 따라서 2개의 이상의 뷰들을 코딩하도록 구성될 수도 있으며, 여기서, 뷰들의 이미지들은 3차원의 (3D) 비디오 효과를 생성하기 위해 실질적으로 동시에 렌더링될 수도 있다. 좀더 구체적으로는, 본 개시물은 디스패리티 모션 벡터 (disparity motion vector) 생성과 관련된 기법들을 기술한다.

[0008] 일 예에서, 비디오 데이터를 디코딩하는 방법은, 현재 블록에 이웃하는 블록이 또한 포함 블록 내에 있다는 결정에 기초하여, 포함 블록 내에 있는 비디오 데이터의 현재 블록이 디스패리티 모션 벡터를 이용하여 인코딩되는 것을 결정하는 단계; 후보 리스트에서 이웃하는 블록을, 포함 블록 외부에 있고 포함 블록에 이웃하는 블록으로 대체하는 단계; 후보 리스트에서 복수의 블록들 중 하나로부터 디스패리티 모션 벡터 예측자를 선택하는 단계; 및 디스패리티 모션 벡터 예측자에 기초하여 디스패리티 모션 벡터를 디코딩하는 단계를 포함한다.

[0009] 또 다른 예에서, 비디오 데이터를 디코딩하는 디바이스는, 현재 블록에 이웃하는 블록이 또한 포함 블록 내에 있다는 결정에 기초하여, 포함 블록 내에 있는 비디오 데이터의 현재 블록이 디스패리티 모션 벡터를 이용하여 인코딩되는 것을 결정하고; 후보 리스트에서 이웃하는 블록을, 포함 블록 외부에 있고 포함 블록에 이웃하는 블록으로 대체하고; 후보 리스트에서 복수의 블록들 중 하나로부터 디스패리티 모션 벡터 예측자를 선택하고; 그리고 디스패리티 모션 벡터 예측자에 기초하여 디스패리티 모션 벡터를 디코딩하도록 구성된 비디오 디코더를 포함한다.

[0010] 또 다른 예에서, 비디오 데이터를 디코딩하는 디바이스는, 포함 블록 내에 있는 비디오 데이터의 현재 블록이 디스패리티 모션 벡터를 이용하여 인코딩되는 것을 결정하는 수단; 현재 블록에 이웃하는 블록이 또한 포함 블록 내에 있다는 결정에 기초하여, 후보 리스트에서 이웃하는 블록을, 포함 블록 외부에 있고 포함 블록에 이웃하는 블록으로 대체하는 수단; 후보 리스트에서 복수의 블록들 중 하나로부터 디스패리티 모션 벡터 예측자를 선택하는 수단; 및 디스패리티 모션 벡터 예측자에 기초하여 디스패리티 모션 벡터를 디코딩하는 수단을 포함한다.

[0011] 또 다른 예에서, 컴퓨터 판독가능 저장 매체는, 실행되는 경우, 하나 이상의 프로세서들로 하여금, 포함 블록 내에 있는 현재 블록에 이웃하는 블록이 또한 포함 블록 내에 있다는 결정에 기초하여, 비디오 데이터의 현재 블록이 디스패리티 모션 벡터를 이용하여 인코딩되는 것을 결정하고; 후보 리스트에서 이웃하는 블록을, 포함 블록 외부에 있고 포함 블록에 이웃하는 블록으로 대체하고; 후보 리스트에서 복수의 블록들 중 하나로부터 디스패리티 모션 벡터 예측자를 선택하고; 그리고 디스패리티 모션 벡터 예측자에 기초하여 디스패리티 모션 벡터를 디코딩하도록 하는 명령들을 안에 저장하고 있다.

[0012] 또 다른 예에서, 비디오 데이터를 인코딩하는 방법은, 디스패리티 모션 벡터를 이용하여 포함 블록 내 비디오

데이터의 현재 블록을 인코딩하는 단계; 현재 블록에 이웃하는 블록이 또한 포함 블록 내에 있다는 결정에 기초하여, 후보 리스트에서 이웃하는 블록을, 포함 블록 외부에 있고 포함 블록에 이웃하는 블록으로 대체하는 단계; 후보 리스트에서 복수의 블록들 중 하나로부터 디스페리티 모션 벡터 예측자를 선택하는 단계; 및 디스페리티 모션 벡터 예측자에 기초하여 디스페리티 모션 벡터를 인코딩하는 단계를 포함한다.

[0013] 또 다른 예에서, 비디오 데이터를 인코딩하는 디바이스는, 디스페리티 모션 벡터를 이용하여 포함 블록 내 비디오 데이터의 현재 블록을 인코딩하고; 현재 블록에 이웃하는 블록이 또한 포함 블록 내에 있다는 결정에 기초하여, 후보 리스트에서 이웃하는 블록을, 포함 블록 외부에 있고 포함 블록에 이웃하는 블록으로 대체하고; 후보 리스트에서 복수의 블록들 중 하나로부터 디스페리티 모션 벡터 예측자를 선택하고; 그리고 디스페리티 모션 벡터 예측자에 기초하여 디스페리티 모션 벡터를 인코딩하도록 구성된 비디오 인코더 포함한다.

[0014] 하나 이상의 예들의 세부 사항들이 첨부도면 및 아래의 상세한 설명에서 개시된다. 다른 특성들, 목적들, 및 이점들은 설명 및 도면들로부터, 그리고 청구항들로부터 명백히 알 수 있을 것이다.

### 도면의 간단한 설명

[0015] 도 1 은 인터-뷰 예측을 위한 디스페리티 벡터들을 생성하는 기법들을 이용할 수도 있는 예시적인 비디오 인코딩 및 디코딩 시스템을 예시하는 블록도이다.

도 2 는 인터-뷰 예측을 위한 디스페리티 벡터들을 생성하는 기법들을 구현할 수도 있는 비디오 인코더의 일 예를 예시하는 블록도이다.

도 3 은 인터-뷰 예측을 위한 디스페리티 벡터들을 생성하는 기법들을 구현할 수도 있는 비디오 디코더의 일 예를 예시하는 블록도이다.

도 4 는 예시적인 멀티뷰 비디오 코딩 (MVC) 예측 패턴을 예시하는 개념도이다.

도 5 는 현재 예측 유닛 (PU) 및 공간적으로 이웃하는 블록들 ( $A_0$ ,  $A_1$ ,  $B_0$ ,  $B_1$ , 및  $B_2$ ) 의 일 예를 예시하는 블록도이다.

도 6 은 상이한 유형들인, 여러 화상들의 그룹들 (GOPs) 의 화상들을 예시하는 개념도이다.

도 7 은 블록들의 라인에서 모션 데이터 압축을 예시하는 개념도이다.

도 8 은 PU 를 포함하는 모션 추정 범위 (MER) 외부에 있는 대체물들에 대한 예를 예시하는 개념도이다.

도 9 는 디스페리티 벡터들이 체크되는 PU 의 일 예를 예시하는 개념도이다.

도 10 은 PU 를 포함하는 MER 외부에 있는 대체물들에 대한 대안 예를 예시하는 개념도이다.

도 11 은 PU 에 대한 디스페리티 벡터들에 대한 하나 이상의 후보 블록들이 PU 를 포함하는 CU 내에 있는 예를 예시하는 개념도이다.

도 12 는 디스페리티 생성 블록 유닛에 대한 디스페리티 벡터 생성의 일 예를 예시하는 개념도이다.

도 13 은 하나의 영역으로서 결합될 때 이웃하는 블록들의 스캐닝 순서에 대한 예를 예시하는 개념도이다.

도 14 는 시간 디스페리티 벡터 체킹을 위한 예시적인 방법을 예시하는 개념도이다.

도 15 는 본 개시물의 기법에 따른, 디스페리티 모션 벡터를 인코딩하는 예시적인 방법을 예시하는 플로우차트이다.

도 16 은 본 개시물의 기법에 따른, 디스페리티 모션 벡터를 디코딩하는 예시적인 방법을 예시하는 플로우차트이다.

### 발명을 실시하기 위한 구체적인 내용

[0016] 일반적으로, 비디오 코딩에서, 비디오 코더들은 공간 예측 (또는, 인트라-예측) 또는 시간 예측 (또는, 인터-예측) 을 이용하여 시퀀스에서의 화상들을 예측 코딩한다. 3차원의 (3D) 효과를 달성하기 위해, 2개의 (또는, 더이상의) 화상들은 상이한 뷰들로부터의 상이한 화상들이 뷰어의 눈들의 각각에 의해 인지되도록, 실질적으로 동시에, 제시될 수도 있다. 특정의 카메라 각도 또는 수평 카메라 위치에 대응하는 시퀀스에서의 화상들은 3D 비디오 데이터의 뷰를 지칭한다. 3D 비디오 코딩에서, 화상들은 시간 예측에 더해서, 또는 대안적으로,

인터-뷰 예측될 수도 있다.

[0017] ISO/IEC 동화상 전문가 그룹 (MPEG) 과 ITU-T 비디오 코딩 전문가 그룹 (VCEG) 의 JCT-VC (Joint Collaborative Team on Video Coding) 에서 고-효율 비디오 코딩 (HEVC) 에 대해 진행중인 비디오 코딩 표준화 노력이 존재한다. HEVC 의 작업 초안은 [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 입수 가능한, 2012년 2월, 미국, 캘리포니아주, 산호세: 8차 회의, JCTVC-H1003, "High efficiency video coding (HEVC) text specification draft 6" 에 설명되어 있다.

[0018] 본 개시물의 기법들은 일반적으로 3DV 로서 또한 지칭되는 HEVC-기반의 3D 비디오 코딩과 관련된다. 특히, 본 개시물은 3DV에서 뷰 합성 모드를 가능하게 하는 기법들을 제공한다. 아래에서 더욱더 자세하게 설명되는 바와 같이, HEVC에서, 비디오 인코더들 및 비디오 디코더들과 같은, 비디오 코더들은, 코딩 유닛들 (CU들) 로서 지칭되는 블록-기반 유닛들을 이용하여 화상들을 코딩한다. CU들은 여러 코딩 모드들을 이용하여 코딩 될 수도 있다.

[0019] 디스패리티-보상되는 예측 (DCP) 으로서 지칭되는 하나의 코딩 모드에서, (CU 또는 CU 의 예측 유닛 (PU) 과 같은) 비디오 데이터의 블록은 디스패리티 벡터를 이용하여 예측될 수도 있다. 디스패리티 벡터는 현재 CU 를 포함하는 화상과 동일한 시간 인스턴스로부터 상이한 뷰에서 가깝게-매칭하는 블록을 식별할 수도 있다. "디스패리티" 는 일반적으로 오브젝트의 이미지들을 캡쳐하거나 생성할 때 카메라 관점들의 수평 오프셋에 의해 유발되는 장면에서 본질적으로 동일한 오브젝트의 2개의 블록들의 수평 오프셋을 지칭한다. 디스패리티 벡터는 일반적으로 상이한 뷰에서의 화상에 대한 CU 또는 CU 의 PU 를 코딩하는데 사용되는 벡터에 대응한다.

[0020] 디스패리티 벡터 구성을 수행하는 종래의 HEVC 기법들은 어떤 문제들에 직면하였다. 예를 들어, 모션 추정 영역 (MER) 이 사용될 때, 공간 디스패리티 벡터를 획득하는데 사용되는 공간 이웃하는 블록들은 그들 블록들이 동일한 MER 에 존재하고 있음으로 인해, 이용불가능할지도 모른다. 이 경우, 최종 디스패리티 벡터는 이용 불가능하거나 또는 덜 정확할 수도 있다. 또한, 일부 인터-뷰 예측 기법들, 예컨대, 잔차 예측은 CU 레벨에서 이루어지며, 반면 일부 인터-뷰 예측 기법, 예컨대, 모션 예측은 PU 레벨에서 이루어진다. 따라서, 디스패리티 벡터 생성은 블록 유닛들의 상이한 레벨들에 적용되는 상이한 인터-뷰 예측 기법들로 인해 다수회 이루어질 필요가 있을 수도 있다. MER 은 MER 이 디스패리티 모션 벡터 예측이 수행되고 있는 CU 또는 PU 를 포함할 때, 포함 블록으로서 지칭될 수도 있으며, 반면 CU 는 CU 가 디스패리티 모션 벡터 예측이 수행되고 있는 PU 를 포함할 때, 포함 블록으로서 지칭될 수도 있다.

[0021] 더욱이, 설령 디스패리티 벡터 구성이 각각의 PU 에 대해 필요로 할지도 모르더라도, 이 PU 의 시간 디스패리티 벡터들은 동일한 CU 에서 다른 PU들의 시간 디스패리티 벡터들과 유사할 수도 있다. 더욱이, 시간 디스패리티 벡터들을 체크할 때, 다수의 PU들이 체크될 수도 있다. 설령 인터-뷰 참조 화상의 블록들이 시간 (temporal) 으로서 간주될 수 있을 때에도, 모션 벡터들은  $16 \times 16$  블록 유닛들로서 반드시 압축되지는 않는다.

[0022] 본 개시물의 기법들은 이들 문제들 중 하나 이상을 극복하는 것을 목적으로 한다. 이들 기법들은 현재 디스패리티 벡터 생성 기법들을 단순하여 통합할 수도 있다. 일부 예들에서, MER 이 사용되고 MER 이 PU 보다 더 클 때, PU 에 대한 인터-뷰 예측은 현재 MER 에 이웃하는 블록들의 공간 디스패리티 벡터들에 기초할 수도 있다. 일부 예들에서, 설령 인터-뷰 예측이 PU 레벨에서 이루어질 때에도, 디스패리티 벡터 생성은 PU 를 포함하는 CU 에 대해 이루어질 수 있다. 일부 예들에서, PU 에 대한 더 정확한 디스패리티 벡터를 생성하기 위해, CU 레벨에서 생성된 디스패리티 벡터는 개선될 수도 있다. 개선은 현재 PU 를 포함하는 CU 의 공간 이웃하는 블록들을 추가로 단지 체킹함으로써, 이루어질 수도 있다. 대안적으로 또는 추가적으로, 개선은 현재 PU 를 포함하는 MER 의 공간 이웃하는 블록들을 단지 추가로 체킹함으로써, 이루어질 수도 있다. 대안적으로 또는 추가적으로, 개선은 현재 PU 의 공간 이웃하는 블록들을 단지 추가로 체킹함으로써, 이루어질 수도 있다.

[0023] 도 1 은 인터-뷰 예측을 위한 디스패리티 벡터들을 생성하는 기법들을 이용할 수도 있는 예시적인 비디오 인코딩 및 디코딩 시스템을 예시하는 블록도 (10) 이다. 도 1 에 나타낸 바와 같이, 시스템 (10) 은 목적지 디바이스 (14) 에 의해 추후에 디코딩되는 인코딩된 비디오 데이터를 제공하는 소스 디바이스 (12) 를 포함한다. 특히, 소스 디바이스 (12) 는 비디오 데이터를 목적지 디바이스 (14) 에 컴퓨터 판독가능 매체 (16) 를 통해서 제공한다. 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 데스크탑 컴퓨터들, 노트북 (즉, 랩탑) 컴퓨터들, 태블릿 컴퓨터들, 셋-탑 박스들, 소위 "스마트" 폰들과 같은 전화기 핸드셋들, 소위 "스마트" 페드들, 텔레비전, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 콘솔들, 비디오 스트리밍

디바이스 등을 포함한, 광범위한 디바이스들 중 임의의 디바이스를 포함할 수도 있다. 일부의 경우, 소스 디바이스 (12) 및 목적지 디바이스 (14)는 무선 통신용으로 탑재될 수도 있다.

[0024] 목적지 디바이스 (14)는 디코딩되는 인코딩된 비디오 데이터를 컴퓨터 판독가능 매체 (16)를 통해서 수신할 수도 있다. 컴퓨터 판독가능 매체 (16)는 인코딩된 비디오 데이터를 소스 디바이스 (12)로부터 목적지 디바이스 (14)로 이동시킬 수 있는 임의 종류의 매체 또는 디바이스를 포함할 수도 있다. 일 예에서, 컴퓨터 판독가능 매체 (16)는 소스 디바이스 (12)로 하여금 인코딩된 비디오 데이터를 직접 목적지 디바이스 (14)로 실시간으로 송신할 수 있게 하는 통신 매체를 포함할 수도 있다. 인코딩된 비디오 데이터는 무선 통신 프로토콜과 같은 통신 표준에 따라서 변조되어 목적지 디바이스 (14)로 송신될 수도 있다. 통신 매체는 무선 주파수 (RF) 스펙트럼 또는 하나 이상의 물리적인 송신 라인들과 같은, 임의의 무선 또는 유선 통신 매체를 포함할 수도 있다. 통신 매체는 근거리 네트워크, 광역 네트워크, 또는 글로벌 네트워크, 예컨대 인터넷과 같은 패킷-기반 네트워크의 일부를 형성할 수도 있다. 통신 매체는 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (12)로부터 목적지 디바이스 (14)로 통신을 용이하게 하는데 유용할 수도 있는 임의의 다른 장비를 포함할 수도 있다.

[0025] 일부 예들에서, 인코딩된 데이터는 출력 인터페이스 (22)로부터 저장 디바이스로 출력될 수도 있다. 이와 유사하게, 인코딩된 데이터는 입력 인터페이스에 의해 저장 디바이스로부터 액세스될 수도 있다. 저장 디바이스는 하드 드라이브, Blu-ray 디스크들, DVD들, CD-ROM들, 플래시 메모리, 휘발성 또는 비-휘발성 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 임의의 다른 적합한 디지털 저장 매체들과 같은 다양한 분산된 또는 로컬 액세스되는 데이터 저장 매체들 중 임의의 데이터 저장 매체를 포함할 수도 있다. 추가 예에서, 저장 디바이스는 소스 디바이스 (12)에 의해 생성된 인코딩된 비디오를 저장할 수도 있는 파일 서버 또는 또 다른 중간 저장 디바이스에 대응할 수도 있다. 목적지 디바이스 (14)는 저장된 비디오 데이터에 저장 디바이스로부터 스트리밍 또는 다운로드를 통해서 액세스할 수도 있다. 파일 서버는 인코딩된 비디오 데이터를 저장하고 그 인코딩된 비디오 데이터를 목적지 디바이스 (14)로 송신하는 것이 가능한 임의 종류의 서버일 수도 있다. 예시적인 파일 서버들은 웹 서버 (예컨대, 웹사이트용), FTP 서버, NAS (NETWORK ATTACHED STORAGE) 디바이스들, 또는 로컬 디스크 드라이브를 포함한다. 목적지 디바이스 (14)는 인터넷 접속을 포함한, 임의의 표준 데이터 접속을 통해서, 인코딩된 비디오 데이터에 액세스할 수도 있다. 이것은 파일 서버 상에 저장된 인코딩된 비디오 데이터에 액세스하는데 적합한, 무선 채널 (예컨대, Wi-Fi 접속), 유선 접속 (예컨대, DSL, 케이블 모뎀 등), 또는 양쪽의 결합을 포함할 수도 있다. 저장 디바이스로부터의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 이들의 조합일 수도 있다.

[0026] 본 개시물의 기법들은 반드시 무선 애플리케이션들 또는 설정들에 한정되지는 않는다. 이 기법들은 오버-디-에어 텔레비전 브로드캐스트들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, 인터넷 스트리밍 비디오 송신들, 예컨대 HTTP (DASH)를 통한 동적 적용 스트리밍, 데이터 저장 매체 상에 인코딩된 디지털 비디오, 데이터 저장 매체 상에 저장된 디지털 비디오의 디코딩, 또는 다른 애플리케이션들과 같은, 다양한 멀티미디어 애플리케이션들 중 임의의 애플리케이션의 지원 하에 비디오 코딩에 적용될 수도 있다. 일부 예들에서, 시스템 (10)은 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅, 및/또는 비디오 전화 통신과 같은, 지원 애플리케이션들로의 1-방향 또는 2-방향 비디오 송신을 지원하도록 구성될 수도 있다.

[0027] 도 1의 예에서, 소스 디바이스 (12)는 비디오 소스 (18), 비디오 인코더 (20), 및 출력 인터페이스 (22)를 포함한다. 목적지 디바이스 (14)는 입력 인터페이스 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (32)를 포함한다. 본 개시물에 따르면, 소스 디바이스 (12)의 비디오 인코더 (20)는 인터-뷰 예측을 위한 디스패리티 벡터들을 생성하는 기법들을 적용하도록 구성될 수도 있다. 다른 예들에서, 소스 디바이스 및 목적지 디바이스는 다른 구성요소들 또는 배열들을 포함할 수도 있다. 예를 들어, 소스 디바이스 (12)는 비디오 데이터를 외부 카메라와 같은 외부 비디오 소스 (18)로부터 수신할 수도 있다. 이와 유사하게, 목적지 디바이스 (14)는 통합된 디스플레이 디바이스를 포함하는 대신, 외부 디스플레이 디바이스와 인터페이스할 수도 있다.

[0028] 도 1의 예시된 시스템 (10)은 단지 일 예이다. 인터-뷰 예측을 위한 디스패리티 벡터들을 생성하는 기법들은 임의의 디지털 비디오 인코딩 및/또는 디코딩 디바이스에 의해 수행될 수도 있다. 일반적으로, 본 개시물의 기법들은 비디오 인코딩 디바이스에 의해 수행되지만, 그 기법들은 또한 "코덱"으로서 일반적으로 지칭되는, 비디오 인코더/디코더에 의해 수행될 수도 있다. 더욱이, 본 개시물의 기법들은 또한 비디오 프리프로세서에 의해 수행될 수도 있다. 소스 디바이스 (12) 및 목적지 디바이스 (14)는 단지 코딩 디바이스들의 예들이며, 여기서, 소스 디바이스 (12)는 목적지 디바이스 (14)로의 송신을 위해 코딩된 비디오 데이터를 생

성한다. 일부 예들에서, 디바이스들 (12, 14) 은 디바이스들 (12, 14) 의 각각이 비디오 인코딩 및 디코딩 구성요소들을 포함하도록, 실질적으로 대칭적 방식으로 동작할 수도 있다. 그러므로, 시스템 (10) 은 예컨대, 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅, 또는 비디오 전화 통신을 위해, 비디오 디바이스들 (12, 14) 사이에 1-방향 또는 2-방향 비디오 송신을 지원할 수도 있다.

[0029] 소스 디바이스 (12) 의 비디오 소스 (18) 는 비디오 카메라와 같은 비디오 캡쳐 디바이스, 이전에 캡쳐된 비디오를 포함하는 비디오 아카이브, 및/또는 비디오 콘텐츠 제공자로부터 비디오를 수신하는 비디오 공급 인터페이스를 포함할 수도 있다. 추가 대안적인 예로서, 비디오 소스 (18) 는 컴퓨터 그래픽스-기반의 데이터를 소스 비디오, 또는 라이브 비디오, 아카이브된 비디오, 및 컴퓨터 생성된 비디오의 조합으로서 생성할 수도 있다.

어떤 경우, 비디오 소스 (18) 가 비디오 카메라이면, 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 소위 카메라 폰들 또는 비디오 폰들을 형성할 수도 있다. 그러나, 위에서 언급한 바와 같이, 본 개시물에서 설명하는 기법들은 비디오 코딩에 일반적으로 적용가능할 수도 있으며, 무선 및/또는 유선 애플리케이션들에 적용될 수도 있다. 각 경우, 캡쳐되거나, 사전-캡쳐되거나, 또는 컴퓨터-생성된 비디오는 비디오 인코더 (20) 에 의해 인코딩될 수도 있다. 인코딩된 비디오 정보는 그후 출력 인터페이스 (22) 에 의해 컴퓨터 판독가능 매체 (16) 상으로 출력될 수도 있다.

[0030] 컴퓨터 판독가능 매체 (16) 는 무선 브로드캐스트 또는 유선 네트워크 송신과 같은 일시성 매체, 또는 하드 디스크, 플래시 드라이브, 컴팩트 디스크, 디지털 비디오 디스크, Blu-ray 디스크, 또는 다른 컴퓨터 판독가능 매체들과 같은 저장 매체들 (즉, 비일시성 저장 매체들) 을 포함할 수도 있다. 일부 예들에서, 네트워크 서버 (미도시) 는 인코딩된 비디오 데이터를 소스 디바이스 (12) 로부터 수신하고 그 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로, 예컨대, 네트워크 송신을 통해서 제공할 수도 있다. 이와 유사하게, 디스크 스탬핑 서비스와 같은 매체 생산 서비스의 컴퓨팅 디바이스는 인코딩된 비디오 데이터를 소스 디바이스 (12) 로부터 수신하고 그 인코딩된 비디오 데이터를 포함하는 디스크를 제조할 수도 있다. 따라서, 컴퓨터 판독가능 매체 (16) 는 여러 예들에서, 여러 형태들의 하나 이상의 컴퓨터 판독가능 매체들을 포함하는 것으로 이해될 수도 있다.

[0031] 목적지 디바이스 (14) 의 입력 인터페이스 (28) 는 컴퓨터 판독가능 매체 (16) 로부터 정보를 수신한다. 컴퓨터 판독가능 매체 (16) 의 정보는 블록들 및 다른 코딩된 유닛들, 예컨대, GOP들의 특성을 및/또는 프로세싱을 기술하는 신택스 엘리먼트들을 포함하는, 비디오 인코더 (20) 에 의해 정의되고 또한 비디오 디코더 (30) 에 의해 사용되는, 신택스 정보를 포함할 수도 있다. 디스플레이 디바이스 (32) 는 그 디코딩된 비디오 데이터를 사용자에게 디스플레이하며, 음극선관 (CRT), 액정 디스플레이 (LCD), 페라로마 디스플레이, 유기 발광 다이오드 (OLED) 디스플레이, 또는 또 다른 유형의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들 중 임의의 디바이스를 포함할 수도 있다.

[0032] 비디오 인코더 (20) 및 비디오 디코더 (30) 는 현재 개발중인 HEVC (HIGH EFFICIENCY VIDEO CODING) 표준과 같은 비디오 코딩 표준에 따라서 동작할 수도 있으며, HEVC 테스트 모델 (HM) 에 따를 수도 있다. 이의 대안으로, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 MPEG 4, 파트 10, AVC (ADVANCED VIDEO CODING) 로서 대안적으로 지정되는 ITU-T H.264 표준, 또는 이런 표준들의 확장판들과 같은 다른 사유 (proprietary) 또는 산업 표준들에 따라서 동작할 수도 있다. 본 개시물의 기법들은, 그러나, 임의의 특정의 코딩 표준에 한정되지 않는다. 비디오 코딩 표준들의 다른 예들은 MPEG-2 및 ITU-T H.263 를 포함한다. 도 1 에 나타내지는 않지만, 일부 양태들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 오디오 인코더 및 디코더와 각각 통합될 수도 있으며, 오디오 및 비디오 양쪽의 인코딩을 공통 데이터 스트림 또는 별개의 데이터 스트림들로 처리하기에 적합한 MUX-DEMUX 유닛들, 또는 다른 하드웨어 및 소프트웨어를 포함할 수도 있다. 적용가능한 경우, MUX-DEMUX 유닛들은 ITU H.223 멀티플렉서 프로토콜, 또는 다른 프로토콜들, 예컨대, 사용자 데이터그램 프로토콜 (UDP) 을 따를 수도 있다.

[0033] ITU-T H.264/MPEG-4 (AVC) 표준은 ISO/IEC 동화상 전문가 그룹 (MPEG) 과 함께, ITU-T 비디오 코딩 전문가 그룹 (VCEG) 에 의해 조인트 비디오 팀 (JVT) 으로서 알려진 공동 파트너쉽의 성과로서, 정식화되었다. 일부 양태들에서, 본 개시물에서 설명하는 기법들은 일반적으로 H.264 표준에 따르는 디바이스들에 적용될 수도 있다. H.264 표준은 2005 년 3월, ITU-T 스터디 그룹에 의한, ITU-T 권고안 H.264, Advanced Video Coding for generic audiovisual services 에 설명되어 있으며, 본원에서 H.264 표준 또는 H.264 사양, 또는 H.264/AVC 표준 또는 사양으로서 지정될 수도 있다. 조인트 비디오 팀 (JVT) 은 H.264/MPEG-4 AVC 에 대한 확장판들에 대해 계속 노력을 기울이고 있다.

[0034] 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들

(DSPs), 주문형 집적회로들 (ASICs), 필드 프로그래밍가능 게이트 어레이들 (FPGAs), 이산 로직, 소프트웨어, 하드웨어, 펌웨어 또는 임의의 이들의 조합들과 같은, 다양한 적합한 인코더 회로 중 임의의 회로로 구현될 수도 있다. 이 기법들이 소프트웨어로 부분적으로 구현되는 경우, 디바이스는 본 개시물의 기법들을 수행하기 위해 소프트웨어용 명령들을 적합한 비일시성 컴퓨터 판독가능 매체에 저장하고, 그 명령들을 하드웨어에서 하나 이상의 프로세서들을 이용하여 실행할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 인코더들 또는 디코더들에 포함될 수도 있으며, 이들 중 어느 쪽이든 각각 디바이스에서 결합된 인코더/디코더 (코덱) 의 일부로서 통합될 수도 있다.

[0035] JCT-VC 는 HEVC 표준의 개발에 노력을 들이고 있다. HEVC 표준화 노력들은 HEVC 테스트 모델 (HM) 로서 지정되는 비디오 코딩 디바이스의 진화 모델 (evolving model) 에 기초한다. HM 은 예컨대, ITU-T H.264/AVC 에 따른 기존 디바이스들에 관련된 비디오 코딩 디바이스들의 여러 추가적인 능력들을 가정한다. 예를 들어, H.264 는 9개의 인트라-예측 인코딩 모드들을 제공하는 반면, HM 은 33개 만큼이나 많은 인트라-예측 인코딩 모드들을 제공할 수도 있다.

[0036] 일반적으로, HM 의 작업 모델은 비디오 프레임 또는 화상이 "코딩 트리 유닛들" 또는 "CTUs" 로서 또한 지칭되는, 루마 샘플 및 크로마 샘플 양쪽을 포함하는, 트리블록들 또는 최대 코딩 유닛들 (LCU) 의 시퀀스로 분할될 수도 있다고 기술한다. 비트스트림 내 신택스 데이터는 LCU 에 대한 사이즈를 정의할 수도 있으며, 이 최대 코딩 유닛은 꾹셀들의 개수의 관점에서 최대 코딩 유닛이다. 슬라이스는 코딩 순서에서 다수의 연속되는 트리블록들을 포함한다. 비디오 프레임 또는 화상은 하나 이상의 슬라이스들로 파티셔닝될 수도 있다. 각각의 트리블록은 큐드트리에 따라 코딩 유닛들 (CU들) 로 분할될 수도 있다. 일반적으로, 큐드트리 데이터 구조는 CU 당 하나의 노드를 포함하며, 동시에, 루트 노드는 트리블록에 대응한다. CU 가 4개의 서브-CU들로 분할되면, CU 에 대응하는 노드는 4개의 리프 노드들을 포함하며, 그 리프 노드 각각은 서브-CU들 중 하나에 대응한다.

[0037] 큐드트리 데이터 구조의 각각의 노드는 대응하는 CU 에 대해 신택스 데이터를 제공할 수도 있다. 예를 들어, 큐드트리에서 노드는 그 노드에 대응하는 CU 가 서브-CU들로 분할되는지의 여부를 나타내는 분할 플래그를 포함할 수도 있다. CU 에 대한 신택스 엘리먼트들은 회귀적으로 정의될 수도 있으며, CU 가 서브-CU들로 분할되는지의 여부에 의존할 수도 있다. CU 가 추가로 분할되지 않으면, 리프-CU 로서 지칭된다. 본 개시물에서, 리프-CU 의 4개의 서브-CU들은 또한 원래 리프-CU 의 명시적인 분할이 없더라도 리프-CU들로 지칭될 것이다. 예를 들어,  $16 \times 16$  사이즈에서 CU 가 추가로 분할되지 않으면, 4개의  $8 \times 8$  서브-CU들이 또한  $16 \times 16$  CU 가 전혀 분할되지 않았더라도 리프-CU들로서 지칭될 것이다.

[0038] CU 는 CU 가 사이즈 구별을 갖지 않는다는 점을 제외하고는, H.264 표준의 매크로블록과 유사한 목적을 갖는다. 예를 들어, 트리블록은 4개의 자식 노드들 (또한, 서브-CU들로서 지칭됨) 로 분할될 수도 있으며, 각각의 자식 노드는 결국 부모 노드일 수도 있으며 또 다른 4개의 자식 노드들로 분할될 수도 있다. 큐드트리의 리프 노드로서 지칭되는, 최종, 미분할된 자식 노드는 또한 리프-CU 로서 지칭되는, 코딩 노드를 포함한다. 코딩 된 비트스트림과 연관되는 신택스 데이터는 최대 CU 깊이로서 지칭되는, 트리블록이 분할될 수도 있는 최대 횟수를 정의할 수도 있으며, 또한 코딩 노드들의 최소 사이즈를 정의할 수도 있다. 따라서, 비트스트림은 또한 최소 코딩 유닛 (SCU) 을 정의할 수도 있다. 본 개시물은 HEVC 의 상황에서, CU, PU, 또는 TU 중 임의의 것, 또는 다른 표준들의 상황에서 유사한 데이터 구조들 (예컨대, H.264/AVC 에서의 매크로블록들 및 그의 서브-블록들) 을 지칭하는데 용어 "블록" 을 사용한다.

[0039] CU 는 코딩 노드, 및 이 코딩 노드와 연관되는 변환 유닛들 (TU들) 및 예측 유닛들 (PU들) 을 포함한다. CU 의 사이즈는 코딩 노드의 사이즈에 대응하며 정사각형 형태이어야 한다. CU 의 사이즈는  $8 \times 8$  꾹셀들로부터  $64 \times 64$  꾹셀들 이상의 최대치를 갖는 트리블록의 사이즈까지 이를 수도 있다. 각각의 CU 는 하나 이상의 PU 들 및 하나 이상의 TU들을 포함할 수도 있다. CU 와 연관되는 신택스 데이터는 예를 들어, 하나 이상의 PU 들로의 CU 의 파티셔닝을 기술할 수도 있다. 파티셔닝 모드들은 CU 가 스킵되는지 또는 직접 모드 인코딩될지, 인트라-예측 모드 인코딩될지, 또는 인터-예측 모드 인코딩될지 여부의 사이에 상이할 수도 있다. PU들은 비-정사각형의 형태로 파티셔닝될 수도 있다. CU 와 연관되는 신택스 데이터는 또한 예를 들어, 큐드트리에 따른 하나 이상의 TU들로의 CU 의 파티셔닝을 기술할 수도 있다. TU 는 정사각형 또는 비-정사각형 (예컨대, 직사각형) 의 형태일 수 있다.

[0040] HEVC 표준은 TU들에 따라서 변환들을 허용하며, 이 TU들은 상이한 CU들에 대해 상이할 수도 있다. TU들은 일반적으로 파티셔닝된 LCU 에 대해 정의된 주어진 CU 내 PU들의 사이즈에 기초하여 사이징되지만, 이것이 항상

그런 것은 아니다. TU들은 일반적으로 PU들과 동일한 사이즈이거나 또는 그보다 작다. 일부 예들에서, CU에 대응하는 잔차 샘플들은 "잔차 퀘드 트리" (RQT)로서 알려진 퀘드트리 구조를 이용하여 더 작은 유닛들로 세분될 수도 있다. RQT의 리프 노드들은 변환 유닛들 (TU들)로서 지칭될 수도 있다. TU들과 연관되는 픽셀 차이 값들은 변환 계수들을 생성하기 위해 변환될 수도 있으며, 그 변환 계수들은 양자화될 수도 있다.

[0041] 리프-CU는 하나 이상의 예측 유닛들 (PUs)을 포함할 수도 있다. 일반적으로, PU는 대응하는 CU의 모두 또는 부분에 대응하는 공간 영역을 나타내며, PU에 대한 참조 샘플을 취출하기 위한 데이터를 포함할 수도 있다. 더욱이, PU는 예측에 관련된 데이터를 포함한다. 예를 들어, PU가 인트라-모드 인코딩될 때, PU에 대한 데이터는 잔차 퀘드트리 (RQT)에 포함될 수도 있으며, PU에 대응하는 TU에 대한 인트라-예측 모드를 기술하는 데이터를 포함할 수도 있다. 또 다른 예로서, PU가 인터-모드 인코딩될 때, PU는 PU에 대한 하나 이상의 모션 벡터들을 정의하는 데이터를 포함할 수도 있다. PU에 대한 모션 벡터를 정의하는 데이터는 예를 들어, 모션 벡터의 수평 성분, 모션 벡터의 수직 성분, 모션 벡터에 대한 해상도 (예컨대, 1/4 픽셀 정밀도 또는 1/8 픽셀 정밀도), 모션 벡터가 가리키는 참조 화상, 및/또는 모션 벡터에 대한 참조 화상 리스트 (예컨대, 리스트 0, 리스트 1, 또는 리스트 C)를 기술할 수도 있다.

[0042] 하나 이상의 PU들을 갖는 리프-CU는 또한 하나 이상의 변환 유닛들 (TU들)을 포함할 수도 있다. 변환 유닛들은 위에서 설명한 바와 같이, RQT (또한, TU 퀘드트리 구조로서 지칭됨)를 이용하여 규정될 수도 있다. 예를 들어, 분할 플래그는 리프-CU가 4개의 변환 유닛들로 분할되는지의 여부를 나타낼 수도 있다. 그 후, 각각의 변환 유닛은 추가적인 서브-TU들로 추가로 분할될 수도 있다. TU가 추가로 분할되지 않을 때, 리프-CU로서 지칭될 수도 있다. 일반적으로, 인트라 코딩에 있어, 리프-CU에 속하는 모든 리프-TU들은 동일한 인트라 예측 모드를 공유한다. 즉, 동일한 인트라-예측 모드가 일반적으로 리프-CU의 모든 TU들에 대해 예측된 값을 계산하기 위해 적용된다. 인트라 코딩에 있어, 비디오 인코더는 각각 리프-TU에 대한 잔차 값을 인트라 예측 모드를 이용하여, TU에 대응하는 CU의 부분과 원래 블록 사이의 차이로서 계산할 수도 있다. TU는 PU의 사이즈에 반드시 제한될 필요는 없다. 따라서, TU들은 PU 보다 더 크거나 또는 더 작을 수도 있다. 인트라 코딩에 있어, PU는 동일한 CU에 대한 대응하는 리프-TU와 콜로케이팅될 수도 있다 (collocated). 일부 예들에서, 리프-TU의 최대 사이즈는 대응하는 리프-CU의 사이즈에 대응할 수도 있다.

[0043] 더욱이, 리프-CU들의 TU들은 또한 잔차 퀘드트리들 (RQTs)로서 지칭되는, 각각의 퀘드트리 데이터 구조들과 연관될 수도 있다. 즉, 리프-CU는 리프-CU가 어떻게 TU들로 파티셔닝되는지를 나타내는 퀘드트리를 포함할 수도 있다. TU 퀘드트리의 루트 노드는 일반적으로 리프-CU에 대응하는 반면, CU 퀘드트리의 루트 노드는 일반적으로 트리블록 (또는, LCU)에 대응한다. 분할되지 않은 RQT의 TU들은 리프-TU들로서 지칭된다. 일반적으로, 본 개시물은 달리 언급하지 않는 한, 리프-CU 및 리프-TU를 지칭하기 위해, 각각 용어들 CU 및 TU를 사용한다.

[0044] 비디오 시퀀스는 일반적으로 비디오 프레임들 또는 화상들의 시리즈를 포함한다. 화상들의 그룹 (GOP)은 일반적으로 비디오 화상들의 하나 이상의 시리즈를 포함한다. GOP는 GOP의 헤더, 화상들의 하나 이상의 헤더, 또는 다른 곳에, GOP에 포함된 다수의 화상을 기술하는 신팩스 데이터를 포함할 수도 있다. 화상의 각각의 슬라이스는 각각의 슬라이스에 대한 인코딩 모드를 기술하는 슬라이스 신팩스 데이터를 포함할 수도 있다. 비디오 인코더 (20)는 일반적으로 비디오 데이터를 인코딩하기 위해 개개의 비디오 슬라이스들을 내비디오 블록들에 대해 동작한다. 비디오 블록은 CU 내 코딩 노드에 대응할 수도 있다. 비디오 블록들은 고정 또는 가변 사이즈들을 가질 수도 있으며, 규정된 코딩 표준에 따라서 사이즈가 상이할 수도 있다.

[0045] 일 예로서, HM은 여러 PU 사이즈들에서 예측을 지원한다. 특정의 CU의 사이즈가  $2N \times 2N$ 이라고 가정하면, HM은  $2N \times 2N$  또는  $N \times N$ 의 PU 사이즈들에서는 인트라-예측을, 그리고  $2N \times 2N$ ,  $2N \times N$ ,  $N \times 2N$ , 또는  $N \times N$ 의 대칭적인 PU 사이즈들에서는 인터-예측을 지원한다. HM은 또한  $2N \times nU$ ,  $2N \times nD$ ,  $nL \times 2N$ , 및  $nR \times 2N$ 의 PU 사이즈들에서의 인터-예측에 대해 비대칭적인 파티셔닝을 지원한다. 비대칭적인 파티셔닝에서, CU의 하나의 방향은 파티셔닝되지 않지만, 다른 방향은 25% 및 75%로 파티셔닝된다. 25% 파티션에 대응하는 CU의 부분은 "상부 (Up)", "하부 (Down)", "좌측 (Left)", 또는 "우측 (Right)"의 표시가 뒤따르는 "n"으로 표시된다. 따라서, 예를 들어, " $2N \times nU$ "는 최상부에서  $2N \times 0.5N$  PU로 그리고 최저부에서  $2N \times 1.5N$  PU로 수평으로 파티셔닝된  $2N \times 2N$  CU를 지칭한다.

[0046] 본 개시물에서, " $N \times N$ " 및 " $N$  곱하기  $N$ "은 수직 및 수평 치수들의 관점에서 비디오 블록의 픽셀 치수들, 예컨

대,  $16 \times 16$  픽셀들 또는 16 곱하기 16 픽셀들을 지정하기 위해 상호교환가능하게 사용될 수도 있다. 일반적으로,  $16 \times 16$  블록은 수직 방향으로 16개의 픽셀들 ( $y = 16$ ) 및 수평 방향으로 16개의 픽셀들 ( $x = 16$ ) 을 가질 것이다. 이와 유사하게,  $N \times N$  블록은 일반적으로 수직 방향으로  $N$  개의 픽셀들 및 수평 방향으로  $N$  개의 픽셀들을 가지며, 여기서  $N$  은 음이 아닌 정수 값을 나타낸다. 블록에서 픽셀들은 로우들 및 칼럼들로 배열될 수도 있다. 더욱이, 블록들은 수직 방향에서와 같이 수평 방향에서 동일한 픽셀들의 개수를 반드시 가질 필요는 없다. 예를 들어, 블록들은  $N \times M$  픽셀들을 포함할 수도 있으며, 여기서  $M$  은 반드시  $N$  과 같을 필요는 없다.

[0047]

**CU** 의 PU들을 이용한 인트라-예측 또는 인터-예측 코딩 이후, 비디오 인코더 (20) 는 **CU** 의 **TU**들에 대한 잔차 데이터를 계산할 수도 있다. PU들은 공간 도메인 (또한, 픽셀 도메인으로 지칭됨) 에서 예측 픽셀 데이터를 생성하는 방법 또는 모드를 기술하는 십екс 데이터를 포함할 수도 있으며, **TU**들은 이산 코사인 변환 (DCT), 정수 변환, 웨이블릿 변환, 또는 잔차 비디오 데이터에 개념적으로 유사한 변환과 같은 변환의 적용 이후 변환 도메인에서의 계수들을 포함할 수도 있다. 잔차 데이터는 미인코딩된 화상의 픽셀들과 PU들에 대응하는 예측 값들 사이의 픽셀 차이들에 대응할 수도 있다. 비디오 인코더 (20) 는 **CU** 에 대한 잔차 데이터를 포함하는 **TU**들을 형성하고, 그후 그 **TU**들을 변환하여, 그 **CU** 에 대한 변환 계수들을 생성할 수도 있다.

[0048]

변환 계수들을 생성하는 임의의 변환들 이후, 비디오 인코더 (20) 는 변환 계수들의 양자화를 수행할 수도 있다. 양자화는 일반적으로 계수들을 나타내는데 사용되는 데이터의 양을 가능한 한 감축하기 위해 변환 계수들이 양자화되는 프로세스를 지칭하며, 추가적인 압축을 제공한다. 양자화 프로세스는 그 계수들의 일부 또는 모두와 연관되는 비트 심도를 감소시킬 수도 있다. 예를 들어,  $n$ -비트 값은 양자화 동안  $m$ -비트 값까지 절사될 수도 있으며, 여기서,  $n$  은  $m$  보다 더 크다.

[0049]

양자화 이후, 비디오 인코더는 변환 계수들을 스캐닝하여, 양자화된 변환 계수들을 포함하는 2차원 매트릭스로부터 1차원 벡터를 생성할 수도 있다. 스캐닝은 어레이의 전면에서 더 높은 에너지 (따라서, 더 낮은 주파수) 계수들을 배치하고, 그리고 어레이의 후면에서 더 낮은 에너지 (따라서, 더 높은 주파수) 계수들을 배치하도록 설계될 수도 있다. 일부 예들에서, 비디오 인코더 (20) 는 엔트로피 인코딩될 수 있는 직렬화된 벡터를 생성하기 위해, 미리 정의된 스캐닝 순서를 이용하여, 양자화된 변환 계수들을 스캐닝할 수도 있다. 다른 예들에서, 비디오 인코더 (20) 는 적응적 스캐닝을 수행할 수도 있다. 양자화된 변환 계수들을 스캐닝하여 1차원 벡터를 형성한 후, 비디오 인코더 (20) 는 예컨대, 컨텍스트-적응 가변 길이 코딩 (CAVLC), 컨텍스트-적응 2진 산술 코딩 (CABAC), 십екс-기반의 컨텍스트-적응 2진 산술 코딩 (SBAC), 확률 간격 파티셔닝 엔트로피 (PIPE) 코딩 또는 또 다른 엔트로피 인코딩 방법론에 따라서, 1차원 벡터를 엔트로피 인코딩할 수도 있다.

비디오 인코더 (20) 는 또한 비디오 데이터를 디코딩할 때에 비디오 디코더 (30) 에 의해 사용하기 위한 인코딩된 비디오 데이터와 연관되는 십екс 엘리먼트들을 엔트로피 인코딩할 수도 있다.

[0050]

CABAC 를 수행하기 위해, 비디오 인코더 (20) 는 컨텍스트 모델 내 컨텍스트를 송신되는 심볼에 할당할 수도 있다. 컨텍스트는 예를 들어, 심볼의 이웃하는 값들이 비-제로인지 여부에 관련될 수도 있다. CAVLC 를 수행하기 위해, 비디오 인코더 (20) 는 송신되는 심볼에 대해 가변 길이 코드를 선택할 수도 있다. VLC 에서의 코드워드들은 상대적으로 더 짧은 코드들이 더 가능성 있는 심볼들에 대응하지만, 더 긴 코드들이 덜 가능성 있는 심볼들에 대응하도록, 구성될 수도 있다. 이와 같이, VLC 의 사용은 예를 들어, 송신되는 각각의 심볼에 대해 동일-길이 코드워드들을 사용하는 것을 넘어서 비트 절감을 달성할 수도 있다. 확률 결정은 그 심볼에 할당된 컨텍스트에 기초할 수도 있다.

[0051]

비디오 인코더 (20) 및 비디오 디코더 (30) 는 본 개시물의 기법들 중 임의의 기법 또는 모두를 수행하도록 구성될 수도 있다. 예를 들어, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 비디오 데이터를 모션 추정 영역 (MER) 을 이용하여 병렬로 코딩하도록 구성될 수도 있다. 본 개시물의 기법들에 따르면, 비디오 인코더 (20) 및 비디오 디코더 (30) 가 현재 코딩중인 PU 보다 더 큰 MER 을 이용할 때, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 MER 내 블록들보다는, 현재 MER 에 이웃하는 블록들의 공간 디스패리티 벡터들에 기초하여, PU 에 대한 인터-뷰 예측을 수행하도록 구성될 수도 있다. 이러한 기법들의 예들은 도 8 - 도 10 에 대해 아래에서 더 자세히 설명된다.

이에 추가적으로 또는 대안적으로, 비디오 인코더 (20) 및 비디오 디코더 (30) 가 PU 레벨에서 인터-뷰 예측을 수행하도록 구성될 때, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 그럼에도 불구하고, PU 를 포함하는 CU 에 대한 디스패리티 벡터 생성을 수행할 수도 있다. 이러한 기법들의 예들은 도 11 에 대해 아래에서 더 자세히 설명된다. 이와 유사하게, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 PU 에 대한 더 정확한 디스

패리티 벡터를 생성하기 위해 CU 레벨에서 생성된 디스패리티 벡터를 개선하도록 구성될 수도 있다. 예를 들어, 비디오 인코더 (20) 및 비디오 디코더 (30)는 개선을 위해, 현재 PU를 포함하는 CU의 공간 이웃하는 블록들을 체크할 수도 있다. 또 다른 예로서, 비디오 인코더 (20) 및 비디오 디코더 (30)는 개선을 위해, 현재 PU를 포함하는 MER의 공간 이웃하는 블록들을 체크할 수도 있다. 또한, 또 다른 예로서, 비디오 인코더 (20) 및 비디오 디코더 (30)는 개선을 위해, 현재 PU의 공간 이웃하는 블록들을 체크할 수도 있다.

[0053] 따라서, 비디오 인코더 (20)는 디스패리티 모션 벡터를 이용하여 포함 블록 내 비디오 데이터의 현재 블록을 인코딩하도록 구성될 수도 있다. 예를 들어, 비디오 인코더 (20)는 인터-뷰 예측을 이용하여 현재 블록을 인코딩하는 것이 최상의, 또는 적어도 허용가능한, 레이트-왜곡 최적화 (RDO) 특성들을 산출한다고 결정할 수도 있다. 더욱이, 현재 블록은 포함 블록, 예컨대 모션 추정 영역 (MER) 내에서 생성할 수도 있다. 이의 대안으로, 현재 블록은 CU 내에서 생성하는 PU를 포함할 수도 있다.

[0054] 어쨌든, 현재 블록에 이웃하는 블록이 또한 포함 블록 내에 있다는 결정에 기초하여, 비디오 인코더 (20)는 후보 리스트에서 이웃하는 블록을, 포함 블록 외부에 있고 포함 블록에 이웃하는 블록으로 대체할 수도 있다. 즉, 모션 벡터 인코딩 (예컨대, 인터-뷰 예측의 경우, 디스패리티 모션 벡터 인코딩)을 수행하려는 목적을 위해, 비디오 인코더 (20)는 모션 벡터 예측자 후보들의 리스트를 포함한, 후보 리스트를 구성할 수도 있다. 일반적으로, 비디오 인코더 (20)는 현재 블록에 공간 및/또는 시간 이웃하는 블록들로부터 모션 벡터들을 리스트에 포함시킬 후보 모션 벡터 예측자들로서 선택할 수도 있다. 본 개시물의 기법들에 따르면, 비디오 인코더 (20)는 포함 블록 내 비디오 데이터의 병렬 인코딩 (그리고 유사하게는, 병렬 디코딩)을 가능하게 하기 위해, 포함 블록 내 블록들의 모션 벡터들을 포함 블록 외부에 있는 블록들의 모션 벡터들로 대체할 수도 있다.

[0055] 비디오 인코더 (20)는 그후 후보 리스트에서 복수의 블록들 중 하나로부터 디스패리티 모션 벡터 예측자를 선택할 수도 있다. 예를 들어, 비디오 인코더 (20)는 (예컨대, 벡터 차이 계산들에 의해 계산되는 바와 같은) 현재 블록에 대한 디스패리티 모션 벡터에 가장 가깝게 매칭하는 후보 모션 벡터 예측자를 중 하나를 리스트에서 선택할 수도 있다. 비디오 인코더 (20)는 그후 디스패리티 모션 벡터 예측자에 기초하여 디스패리티 모션 벡터를 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20)는 병합 모드 또는 진보된 모션 벡터 예측 (AMVP) 기법들을 이용하여 디스패리티 모션 벡터를 인코딩할 수도 있다.

[0056] 병합 모드에서, 비디오 인코더 (20)는 비디오 디코더 (30)로 하여금 그 이웃하는 블록의 모션 정보를 현재 블록에 대한 모션 정보로서 사용할 수 있게 하기 위해, 모션 정보가 취출되어질 이웃하는 블록을 식별하는 병합 인덱스 값을 시그널링할 수도 있다. 본 개시물의 기법들에 따르면, 시그널링된 이웃하는 블록이 포함 블록 외부에 있으면, 이러한 시그널링은 비디오 디코더 (30)로 하여금 현재 블록에 대한 모션 정보로서 사용할 대체된 블록에 대한 모션 정보를 취출하게 하도록 프롬프트할 수도 있다. 병합 인덱스에 대응하는 블록으로부터 상속된 모션 정보는 참조 화상 리스트 및 참조 화상에 대응하는 참조 화상 인덱스를 더 포함할 수도 있다. HEVC에 대한 하이-레벨 신택스 (HLS)-단독 확장판을 이용하기 위해, 비디오 인코더 (20)는 인터-뷰 참조 화상들을 장기 참조 화상들로서, 그리고 인트라-뷰 참조 화상들을 단기 참조 화상들로서 시그널링하도록 구성될 수도 있으며, 이 경우, 참조 화상 인덱스 및 참조 화상 리스트 시그널링은 단일 뷰 코딩을 위한 HEVC의 기법들과 실질적으로 동일하거나 또는 동일하게 유지할 수도 있다.

[0057] AMVP에서, 비디오 인코더 (20)는, 병합 모드에 대한 병합 인덱스와 실질적으로 동일한 방법으로 이웃하는 블록에 대응할 수도 있는, 모션 정보가 취출될 이웃하는 블록을 식별하는 AMVP 인덱스를 시그널링할 수도 있다. 그러나, 비디오 인코더 (20)는 디스패리티 모션 벡터와 AMVP 인덱스에 대응하는 블록의 모션 벡터 예측자 사이의 차이를 나타내는 모션 벡터 차이 값을 추가적으로 인코딩할 수도 있다. 본 개시물의 기법들에 따르면, AMVP 인덱스가 포함 블록 내에 또한 있는 현재 블록에 이웃하는 블록에 대응하는 경우, 모션 벡터 예측자를 정의하는 모션 정보는 포함 블록 외부에 있는 블록으로부터 실제로 취출될 수도 있다. 비디오 인코더 (20)는 또한 참조 화상 리스트 및 참조 인덱스를 나타내는 데이터를 시그널링할 수도 있다. 더욱이, 비디오 디코더 (30)는 디스패리티 모션 벡터를 이용하여 현재 블록을 인코딩할 수도 있다.

[0058] 따라서, 비디오 디코더 (30)는 비디오 인코더 (20)의 기법과는 실질적으로 반대인 기법을 수행하도록 구성될 수도 있다. 예를 들어, 비디오 디코더 (30)는, 현재 블록에 이웃하는 블록이 또한 포함 블록 내에 있다는 결정에 기초하여, 포함 블록 내에 있는 비디오 데이터의 현재 블록이 디스패리티 모션 벡터를 이용하여 인코딩되는 것을 결정하고, 후보 리스트에서 이웃하는 블록을, 포함 블록 외부에 있고 포함 블록에 이웃하는 블록으로 대체하고, 후보 리스트에서 복수의 블록들 중 하나로부터 디스패리티 모션 벡터 예측자를 선택하고, 그리고 디스패리티 모션 벡터 예측자에 기초하여 디스패리티 모션 벡터를 디코딩하도록 구성될 수도 있다.

[0059]

예를 들어, 비디오 디코더 (30) 는 현재 블록이 디스패리티 모션 벡터를 이용하여 인코딩된다는 것을 나타내는 시그널링 데이터를 수신할 수도 있다. 비디오 디코더 (30) 는 추가로 포함 블록 내 현재 블록의 위치 뿐만 아니라, 현재 블록에 이웃하는 블록들의 위치들을 결정하여, 이웃하는 블록들 또는 포함 블록 외부에 있고 포함 블록에 이웃하는 블록들을 후보 리스트에 추가할지의 여부를 결정할 수도 있다.

[0060]

비디오 디코더 (30) 는 디스패리티 모션 벡터를 디코딩하는데 이웃하는 블록들 중 어느 블록을 사용할지를 식별하는 식별자 (예컨대, 병합 인덱스 또는 AMVP 인덱스) 를 수신할 수도 있다. 본 개시물의 기법들에 따르면, 시그널링된 이웃하는 블록이 현재 블록을 포함하는 포함 블록 내에 있으면, 비디오 디코더 (30) 는 포함 블록 외부에 있는 포함 블록에 이웃하는 블록으로부터 모션 정보를 실제로 수신할 수도 있다. 비디오 디코더 (30) 는 그후 식별자에 대응하는 블록으로부터 취출된 모션 벡터 예측자를 이용하여 디스패리티 모션 벡터를 디코딩할 수도 있다. 이와 유사하게, 비디오 디코더 (30) 는 디스패리티 모션 벡터를 이용하여 현재 블록을 디코딩할 수도 있다.

[0061]

비디오 인코더 (20) 및 비디오 디코더 (30) 각각은, 적용 가능한 경우, 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들 (DSPs), 주문형 집적회로들 (ASICs), 필드 프로그래밍 가능 게이트 어레이들 (FPGAs), 이산 로직 회로, 소프트웨어, 하드웨어, 펌웨어 또는 임의의 이들의 조합들과 같은, 다양한 적합한 인코더 또는 디코더 회로 중 임의의 회로로서 구현될 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 인코더들 또는 디코더들에 포함될 수도 있으며, 이를 중 어느 쪽이든 결합된 비디오 인코더/디코더 (코덱) 의 일부로서 통합될 수도 있다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 를 포함하는 디바이스는 집적 회로, 마이크로프로세서, 및/또는 무선 통신 디바이스, 예컨대 셀룰러 전화기를 포함할 수도 있다.

[0062]

도 2 는 인터-뷰 예측을 위한 디스패리티 벡터들을 생성하는 기법들을 구현할 수도 있는 비디오 인코더 (20) 의 일 예를 예시하는 블록도이다. 비디오 인코더 (20) 는 비디오 슬라이스들 내 비디오 블록들의, 시간 인트라-코딩, 시간 인터-코딩, 및 인터-뷰 코딩을 수행할 수도 있다. 시간 인트라 코딩은 주어진 비디오 프레임 또는 화상 내 비디오에서 공간 리던던시를 감소시키거나 또는 제거하기 위해 공간 예측에 의존한다. 시간 인터-코딩은 비디오 시퀀스의 인접 프레임들 또는 화상들 내 비디오에서 시간 리던던시를 감소시키거나 또는 제거하기 위해 시간 예측에 의존한다. 인터-뷰 코딩은 관점에서 수평으로 시프트될 수도 있는 또 다른 뷰로부터 화상들에서의 리던던시를 감소시키거나 또는 제거하기 위해, 인터-뷰 예측에 의존한다. 시간 인트라-모드 (I 모드) 는 여러 공간 기반의 코딩 모드들 중 임의의 코딩 모드를 지칭할 수도 있다. 단방향 예측 (P 모드) 또는 양방향-예측 (B 모드) 과 같은, 시간 인터-모드들은 여러 시간-기반의 코딩 모드들 중 임의의 코딩 모드를 지칭할 수도 있다. 인터-뷰 모드들은 디스패리티 모션 벡터 또는 다른 이러한 인터-뷰 코딩 기법들의 사용을 지칭할 수도 있다. 일반적으로, 인터-뷰 예측이 시간 인터-코딩과 실질적으로 유사한 방법으로 수행될 수도 있다.

[0063]

도 2 에 나타낸 바와 같이, 비디오 인코더 (20) 는 인코딩되는 비디오 프레임 내 현재 비디오 블록을 수신한다. 도 2 의 예에서, 비디오 인코더 (20) 는 모드 선택 유닛 (40), 참조 화상 메모리 (64), 합산기 (50), 변환 프로세싱 유닛 (52), 양자화 유닛 (54), 및 엔트로피 인코딩 유닛 (56) 을 포함한다. 모드 선택 유닛 (40) 은, 결과적으로, 모션 보상 유닛 (44), 모션 추정 유닛 (42), 인트라-예측 유닛 (46), 및 파티션 유닛 (48) 을 포함한다. 비디오 블록 재구성을 위해, 비디오 인코더 (20) 는 또한 역양자화 유닛 (58), 역변환 유닛 (60), 및 합산기 (62) 를 포함한다. 디블록킹 필터 (도 2 에 미도시) 가 또한 블록 경계들을 필터링하여 재구성된 비디오로부터 블록킹 현상 아티팩트들을 제거하기 위해 포함될 수도 있다. 원활 경우, 디블록킹 필터는 일반적으로 합산기 (62) 의 출력을 필터링할 것이다. (인 루프 또는 사후 루프에서) 추가적인 필터들이 또한 디블록킹 필터에 추가하여 사용될 수도 있다. 이러한 필터들은 간결성을 위해 도시되지 않지만, 그러나 원활 경우, 합산기 (50) 의 출력을 (인-루프 필터로서) 필터링할 수도 있다.

[0064]

인코딩 프로세스 동안, 비디오 인코더 (20) 는 코딩될 비디오 프레임 또는 슬라이스를 수신한다. 프레임 또는 슬라이스는 다수의 비디오 블록들로 분할될 수도 있다. 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 은 시간 예측을 제공하기 위해, 하나 이상의 참조 프레임들에서 하나 이상의 블록들에 대한, 수신된 비디오 블록의 인터-예측 코딩 (시간 뷰 이든 또는 인터-뷰 이든) 을 수행한다. 인트라-예측 유닛 (46) 은 대안적으로, 공간 예측을 제공하기 위해, 코딩되는 블록과 동일한 프레임 또는 슬라이스에서의 하나 이상의 이웃하는 블록들에 대한 수신된 비디오 블록의 인트라-예측 코딩을 수행할 수도 있다. 비디오 인코더 (20) 는 예컨대, 비디오 테이터의 각각의 블록에 대해 적합한 코딩 모드를 선택하기 위해, 다수의 코딩 패스들 (passes) 을 수행할 수도 있다.

[0065]

더욱이, 파티션 유닛 (48)은 이전 코딩 패스들에서의 이전 파티셔닝 방식들의 평가에 기초하여, 비디오 데이터의 블록들을 서브-블록들로 파티셔닝할 수도 있다. 예를 들어, 파티션 유닛 (48)은 레이트-왜곡 분석 (예컨대, 레이트-왜곡 최적화)에 기초하여, 처음에 프레임 또는 슬라이스를 LCU들로 파티셔닝하고, LCU들의 각각을 서브-CU들로 파티셔닝할 수도 있다. 모드 선택 유닛 (40)은 서브-CU들로의 LCU의 파티셔닝을 나타내는 큐드트리 데이터 구조를 추가로 생성할 수도 있다. 큐드트리의 리프-노드 CU들은 하나 이상의 PU들 및 하나 이상의 TU들을 포함할 수도 있다.

[0066]

모드 선택 유닛 (40)은 코딩 모드들 중 하나 (시간 인트라-코딩, 시간 인터-코딩, 또는 인터-뷰 코딩)를, 예컨대, 여러 결과들에 기초하여 선택할 수도 있으며, 그리고, 최종 인트라- 또는 인터-코딩된 (시간 또는 인터-뷰) 블록을 합산기 (50)에 제공하여, 잔차 블록 데이터를 생성하고, 그리고, 합산기 (62)에 제공하여 참조 프레임으로서의 사용을 위해 인코딩된 블록을 재구성한다. 모드 선택 유닛 (40)은 또한 모션 벡터들 (예컨대, 시간 모션 벡터들 또는 디스패리티 모션 벡터들), 인트라-모드 표시자들, 파티션 정보, 및 다른 이러한 신팩스 정보와 같은 신팩스 엘리먼트들을, 엔트로피 인코딩 유닛 (56)에 제공한다.

[0067]

모션 추정 유닛 (42) 및 모션 보상 유닛 (44)은 고도로 통합될 수도 있지만, 개념적인 목적들을 위해 별개로 예시된다. 모션 추정 유닛 (42)에 의해 수행되는 모션 추정은 모션 벡터들을 생성하는 프로세스이며, 이 프로세스는 비디오 블록들에 대한 모션을 추정한다. 디스패리티 모션 벡터는 일반적으로 상이한 뷰들에 대한 약간 상이한 카메라 관점들로부터 기인하는, 상이한 뷰들로부터의 대응하는 화상들의 2개의 블록들 사이의 수평 시프트, 즉, 디스패리티를 나타낸다. 모션 벡터는, 예를 들어, 현재 프레임 (또는, 다른 코딩된 유닛) 내 코딩중인 현재 블록에 대한 참조 프레임 (또는, 다른 코딩된 유닛) 내 예측 블록에 상대적인, 현재 비디오 프레임 또는 화상 내 비디오 블록의 PU의 변위를 나타낼 수도 있다. 예측 블록은 픽셀 차이의 관점에서 코딩되는 블록에 가깝게 매칭하는 것으로 발견되는 블록이며, SAD (sum of absolute difference), SSD (sum of square difference), 또는 다른 차이 메트릭들에 의해 결정될 수도 있다. 일부 예들에서, 비디오 인코더 (20)는 참조 화상 메모리 (64)에 저장된 참조 화상들의 서브-정수 픽셀 위치들에 대한 값들을 계산할 수도 있다. 예를 들어, 비디오 인코더 (20)는 참조 화상의 1/4 픽셀 위치들, 1/8 픽셀 위치들, 또는 다른 분수 픽셀 위치들의 값들을 내삽할 수도 있다. 따라서, 모션 추정 유닛 (42)은 풀 픽셀 위치들 및 분수 픽셀 위치들에 대해, 모션 탐색을 수행하고, 분수 픽셀 정밀도를 가진 모션 벡터를 출력할 수도 있다.

[0068]

모션 추정 유닛 (42)은 PU의 위치를 참조 화상의 예측 블록의 위치와 비교함으로써, 인터-코딩된 (시간 또는 인터-뷰) 슬라이스에서 비디오 블록의 PU에 대한 모션 벡터를 계산한다. 참조 화상은 제 1 참조 화상 리스트 (리스트 0) 또는 제 2 참조 화상 리스트 (리스트 1)로부터 선택될 수도 있으며, 이 리스트 각각은 하나 이상의 참조 화상 메모리 (64)에 저장된 참조 화상들을 식별한다. 일부 예들에서, 인터-뷰 참조 화상들은 장기 참조 화상들로서 시그널링될 수도 있으며, 반면 시간 참조 화상들은 단기 참조 화상들로서 시그널링될 수도 있다. 이의 대안으로, 인터-뷰 참조 화상들은 단기 참조 화상들로서 시그널링될 수도 있으며, 반면 시간 참조 화상들은 장기 참조 화상들로서 시그널링될 수도 있다. 모션 추정 유닛 (42)은 그 계산된 모션 벡터를 엔트로피 인코딩 유닛 (56) 및 모션 보상 유닛 (44)으로 전송한다.

[0069]

모션 보상 유닛 (44)에 의해 수행되는 모션 보상은 모션 추정 유닛 (42)에 의해 결정된 모션 벡터에 기초하여 예측 블록을 폐치하거나 또는 생성하는 것을 수반할 수도 있다. 또, 모션 추정 유닛 (42) 및 모션 보상 유닛 (44)은 일부 예들에서, 기능적으로 통합될 수도 있다. 현재 비디오 블록의 PU에 대한 모션 벡터를 수신하자 마자, 모션 보상 유닛 (44)은 모션 벡터가 참조 화상 리스트들 중 하나에서 가리키는 예측 블록을 케이트할 수도 있다. 합산기 (50)는 이하에서 설명하는 바와 같이, 코딩중인 현재 비디오 블록의 픽셀 값들로부터 예측 블록의 픽셀 값들을 감산하여 픽셀 차이 값들을 형성함으로써, 잔차 비디오 블록을 형성한다. 일반적으로, 모션 추정 유닛 (42)은 루마 성분들에 대해 모션 추정을 수행하고, 모션 보상 유닛 (44)은 크로마 성분들 및 루마 성분들 양쪽에 대해 루마 성분들에 기초하여 계산된 모션 벡터들을 이용한다. 모드 선택 유닛 (40)은 또한 비디오 슬라이스의 비디오 블록들을 디코딩할 때에 비디오 디코더 (30)에 의한 사용을 위해 비디오 블록들 및 비디오 슬라이스와 연관되는 신팩스 엘리먼트들을 생성할 수도 있다.

[0070]

게다가, 비디오 인코더 (20)는 모션 벡터를, 예컨대, 병합 모드 또는 AMVP 중 하나를 이용하여 인코딩할 수도 있다. 본 개시물의 기법들에 따르면, 현재 블록이 포함 블록 (예컨대, MER) 내에 포함되는 경우, 포함 블록이 병렬로 코딩되는 것이 가능한 영역을 정의한다고 가정하면, 비디오 인코더 (20)는 현재 블록에 이웃하는 블록이 포함 블록 내에 있을 때, 포함 블록에 이웃하는 블록의 모션 벡터를 현재 블록에 적용할 수도 있다. 즉, 현재 블록이 포함 블록 내에 있을 때, 그리고 현재 블록에 이웃하는 블록이 또한 포함 블록 내에 있을 때, 비디오 인코더 (20)는 모션 벡터 코딩의 목적들을 위해, 현재 블록에 이웃하는 블록을, 포함 블록 외부에 있고

포함 블록에 이웃하는 블록으로, 대체할 수도 있다.

[0071] 본 개시물에 따른 모션 벡터 코딩 기법들의 설명의 목적들을 위해, 용어 "이웃하는 블록"은 이웃하는 블록이 현재 블록을 포함하는 블록 외부에 있을 때 현재 블록에 이웃하는 블록을 지칭하는 것으로 이해되어야 하지만, 그렇지 않고 이웃하는 블록이 포함 블록 내에 있을 때는 포함 블록에 이웃하고 외부에 있는 블록을 지칭하는 것으로 이해되어야 한다. 병합 모드에서, 비디오 인코더 (20)는 이웃하는 블록의 모션 벡터를 현재 블록 (예컨대, 현재 PU)에 적용할 수도 있다. AMVP 모드에서, 비디오 인코더 (20)는 현재 블록에 대한 모션 벡터와 이웃하는 블록의 모션 벡터 사이의 모션 벡터 차이를 계산할 수도 있으며, 엔트로피 인코딩 유닛 (56)은 모션 벡터 차이를 나타내는 데이터를 인코딩할 수도 있다. 더욱이, 병합 모드에서, 엔트로피 인코딩 유닛 (56)은 이웃하는 블록을 식별하는 병합 인덱스를 정의하는 데이터를 인코딩할 수도 있으며, 반면 AMVP에서, 엔트로피 인코딩 유닛 (56)은 이웃하는 블록을 식별하는 AMVP 인덱스를 정의하는 데이터를 인코딩할 수도 있다. 병합 인덱스 및 AMVP 인덱스는 이웃하는 블록 식별자들의 예들을 나타낸다

[0072] 좀더 자세하게 설명하면, 비디오 인코더 (20)는 현재 블록 (예컨대, 현재 PU)에 대한 하나 이상의 공간 이웃들에 대한 식별자들을 포함하는, 후보 리스트를 구성할 수도 있다. 공간 이웃들의 예들이 아래 도 5에 도시된다. 그러나, 현재 블록이 포함 블록 내에 있다고 (그리고, 포함 블록이 병렬 코딩을 가능하게 하는 영역을 정의하고 있다고) 가정하면, 비디오 인코더 (20)는 포함 블록 내에 또한 포함되는 현재 블록에 이웃하는 블록들 중 임의의 블록에 대한 대체물들을 결정할 수도 있다. 대체 방식들의 예들이 도 8 내지 도 11에 도시된다.

[0073] 일반적으로, 비디오 인코더 (20)는 유사하게 포지셔닝된 포함 블록 외부의 블록을 포함 블록 내에 있는 현재 블록에 대한 이웃하는 블록으로 대체할 수도 있다. 예를 들어, 일부 예들에서, 포함 블록 내 이웃하는 블록이 현재 블록의 특정의 모서리에 있으면, 대체된 블록은 포함 블록의 동일한 모서리에 있을 수도 있다. 또 다른 예로서, 이웃하는 블록이 현재 블록의 좌측면 상에 있으면, 대체된 블록은 동일한 수직 위치를 갖지만, 대체된 블록을 포함 블록의 외부에, 그러나, 바로 옆에 배치하는 수평 위치를 가질 수도 있다. 이와 유사하게, 이웃하는 블록이 현재 블록의 상부측 상에 있으면, 대체된 블록은 동일한 수평 위치를 갖지만 대체된 블록을 포함 블록의 외부에, 바로 옆에 배치하는 수직 위치를 가질 수도 있다. 비디오 인코더 (20)는, 마치 이웃하는 블록이 포함 블록 내에 있었고 현재 블록에 이웃하고 있는 것처럼, 포함 블록에 있는 이웃하는 블록을 나타내는 데이터를 인코딩할 수도 있다. 도 3에 대해 아래에서 설명되는 바와 같이, 그러나, 그렇지 않고 이웃하는 블록이 포함 블록 내에 있을 때, 비디오 디코더 (30)는 코딩된 데이터가 포함 블록 외부에 있는 이웃하는 블록에 대응한다고 결정하도록 구성될 수도 있다.

[0074] 인트라-예측 유닛 (46)은 위에서 설명한 바와 같이, 모션 추정 유닛 (42) 및 모션 보상 유닛 (44)에 의해 수행되는 인터-예측에 대한 대안으로서, 현재 블록을 인트라-예측할 수도 있다. 특히, 인트라-예측 유닛 (46)은 현재 블록을 인코딩하는데 사용할 인트라-예측 모드를 결정할 수도 있다. 일부 예들에서, 인트라-예측 유닛 (46)은 예컨대, 별개의 인코딩 과정들 동안 여러 인트라-예측 모드들을 이용하여 현재 블록을 인코딩할 수도 있으며, 인트라-예측 유닛 (46) (또는, 일부 예들에서는, 모드 선택 유닛 (40))은 테스트된 모드들로부터 사용할 적합한 인트라-예측 모드를 선택할 수도 있다.

[0075] 예를 들어, 인트라-예측 유닛 (46)은 여러 테스트된 인트라-예측 모드들에 대한 레이트-왜곡 분석을 이용하여 레이트-왜곡 값들을 계산하고, 그 테스트된 모드들 중에서 최상의 레이트-왜곡 특성들을 갖는 인트라-예측 모드를 선택할 수도 있다. 레이트-왜곡 분석은 일반적으로 인코딩된 블록과 그 인코딩된 블록을 생성하기 위해 인코딩되었던 원래의 미인코딩된 블록 사이의 왜곡의 양 (또는, 에러) 뿐만 아니라, 그 인코딩된 블록을 생성하는데 사용된 비트레이트 (즉, 다수의 비트들)를 결정한다. 인트라-예측 유닛 (46)은 여러 인코딩된 블록들에 대한 왜곡들 및 레이트들로부터 비율들 (비들)을 계산하여, 어느 인트라-예측 모드가 그 블록에 대해 최상의 레이트-왜곡 값을 나타내는지를 결정할 수도 있다.

[0076] 블록에 대한 인트라-예측 모드를 선택한 후, 인트라-예측 유닛 (46)은 블록에 대한 그 선택된 인트라-예측 모드를 나타내는 정보를 엔트로피 인코딩 유닛 (56)에 제공할 수도 있다. 엔트로피 인코딩 유닛 (56)은 그 선택된 인트라-예측 모드를 나타내는 정보를 인코딩할 수도 있다. 비디오 인코더 (20)는 복수의 인트라-예측 모드 인덱스 테이블들 및 복수의 수정된 인트라-예측 모드 인덱스 테이블들 (또한, 코드워드 맵핑 테이블들로서 지칭됨)을 포함할 수도 있는 그 송신되는 비트스트림 구성 데이터에, 여러 블록들에 대한 인코딩 컨텍스트들의 정의들, 및 가장 가능성있는 인트라-예측 모드, 인트라-예측 모드 인덱스 테이블 및 컨텍스트들의 각각에 사용할 수정된 인트라-예측 모드 인덱스 테이블의 표시들을 포함할 수도 있다.

[0077]

비디오 인코더 (20)는 코딩중인 원래 비디오 블록으로부터, 모드 선택 유닛 (40)으로부터의 예측 데이터를 감산함으로써 잔차 비디오 블록을 형성한다. 합산기 (50)는 이 감산 동작을 수행하는 구성요소 또는 구성요소들을 나타낸다. 변환 프로세싱 유닛 (52)은 이산 코사인 변환 (DCT) 또는 개념적으로 유사한 변환과 같은 변환을 잔차 블록에 적용하여, 잔차 변환 계수 값들을 포함하는 비디오 블록을 생성한다. 변환 프로세싱 유닛 (52)은 DCT 와 개념적으로 유사한 다른 변환들을 수행할 수도 있다. 웨이블릿 변환들, 정수 변환들, 서브밴드 변환들 또는 다른 유형들의 변환들이 또한 이용될 수 있다. 어쨌든, 변환 프로세싱 유닛 (52)은 그 변환을 잔차 블록에 적용하여, 잔차 변환 계수들의 블록을 생성한다. 변환은 잔차 정보를 픽셀 값 도메인으로부터 주파수 도메인과 같은 변환 도메인으로 변환할 수도 있다. 변환 프로세싱 유닛 (52)은 최종 변환 계수들을 양자화 유닛 (54)으로 전송할 수도 있다. 양자화 유닛 (54)은 비트 레이트를 추가로 감소시키기 위해 변환 계수들을 양자화한다. 양자화 프로세스는 그 계수들의 일부 또는 모두와 연관되는 비트 심도를 감소시킬 수도 있다. 양자화의 정도는 양자화 파라미터를 조정함으로써 변경될 수도 있다. 일부 예들에서, 양자화 유닛 (54)은 그후 양자화된 변환 계수들을 포함하는 매트릭스의 스캐닝을 수행할 수도 있다. 이의 대안으로, 엔트로피 인코딩 유닛 (56)은 그 스캐닝을 수행할 수도 있다.

[0078]

양자화 이후, 엔트로피 인코딩 유닛 (56)은 양자화된 변환 계수들을 엔트로피 코딩한다. 예를 들어, 엔트로피 인코딩 유닛 (56)은 컨텍스트 적응 가변 길이 코딩 (CAVLC), 컨텍스트 적응 2진 산술 코딩 (CABAC), 신택스-기반의 컨텍스트-적응 2진 산술 코딩 (SBAC), 확률 간격 파티셔닝 엔트로피 (PIPE) 코딩 또는 또 다른 엔트로피 코딩 기법을 수행할 수도 있다. 컨텍스트-기반의 엔트로피 코딩의 경우, 컨텍스트는 이웃하는 블록들에 기초할 수도 있다. 엔트로피 인코딩 유닛 (56)에 의한 엔트로피 코딩 이후, 인코딩된 비트스트림은 또 다른 디바이스 (예컨대, 비디오 디코더 (30))로 송신되거나 또는 추후 송신 또는 취출을 위해 아카이브될 수도 있다.

[0079]

역양자화 유닛 (58) 및 역변환 유닛 (60)은 역양자화 및 역변환을 각각 적용하여, 예컨대, 참조 블록으로 추후 사용을 위해, 픽셀 도메인에서 잔차 블록을 재구성한다. 모션 보상 유닛 (44)은 잔차 블록을 참조 화상 메모리 (64)의 프레임들 중 하나의 예측 블록에 가산함으로써 참조 블록을 계산할 수도 있다. 모션 보상 유닛 (44)은 또한 하나 이상의 내삽 필터들을 그 재구성된 잔차 블록에 적용하여, 모션 추정에 사용하기 위한 서브-정수 픽셀 값들을 계산할 수도 있다. 합산기 (62)는 재구성된 잔차 블록을 모션 보상 유닛 (44)에 의해 생성되는 모션 보상된 예측 블록에 가산하여, 참조 화상 메모리 (64)에의 저장을 위한 재구성된 비디오 블록을 생성한다. 재구성된 비디오 블록은 후속 비디오 프레임에서 블록을 인터-코딩하기 위해 모션 추정 유닛 (42) 및 모션 보상 유닛 (44)에 의해 참조 블록으로서 사용될 수도 있다.

[0080]

도 2 의 비디오 인코더 (20)는 디스패리티 모션 벡터를 이용하여 포함 블록 내 비디오 데이터의 현재 블록을 인코딩하고, 현재 블록에 이웃하는 블록이 또한 포함 블록 내에 있다는 결정에 기초하여, 후보 리스트에서 이웃하는 블록을, 포함 블록 외부에 있고 포함 블록에 이웃하는 블록으로 대체하고, 후보 리스트에서 복수의 블록들 중 하나로부터 디스패리티 모션 벡터 예측자를 선택하고, 그리고 디스패리티 모션 벡터 예측자에 기초하여 디스패리티 모션 벡터를 인코딩하도록 구성된 비디오 인코더의 일 예를 나타낸다.

[0081]

도 3 은 인터-뷰 예측을 위한 디스패리티 벡터들을 생성하는 기법들을 구현할 수도 있는 비디오 디코더 (30)의 일 예를 예시하는 블록도이다. 도 3 의 예에서, 비디오 디코더 (30)는 엔트로피 디코딩 유닛 (70), 모션 보상 유닛 (72), 인트라 예측 유닛 (74), 역양자화 유닛 (76), 역변환 유닛 (78), 참조 화상 메모리 (82) 및 합산기 (80)를 포함한다. 비디오 디코더 (30)는 일부 예들에서, 비디오 인코더 (20)에 대해 설명된 인코딩 과정과는 일반적으로 반대인 디코딩 과정을 수행할 수도 있다 (도 2). 모션 보상 유닛 (72)은 엔트로피 디코딩 유닛 (70)으로부터 수신된 시간 또는 디스패리티 모션 벡터들에 기초하여 예측 데이터를 생성할 수도 있으며, 반면 인트라-예측 유닛 (74)은 디코딩 유닛 (70)으로부터 수신된 인트라-예측 모드 표시자들에 기초하여 예측 데이터를 생성할 수도 있다.

[0082]

디코딩 프로세스 동안, 비디오 디코더 (30)는 인코딩된 비디오 슬라이스의 비디오 블록들 및 연관되는 신택스 엘리먼트들을 나타내는 인코딩된 비디오 비트스트림을 비디오 인코더 (20)로부터 수신한다. 비디오 디코더 (30)의 엔트로피 디코딩 유닛 (70)은 그 비트스트림을 엔트로피 인코딩하여, 양자화된 계수들, 모션 벡터들 또는 인트라-예측 모드 표시자들, 및 다른 신택스 엘리먼트들을 생성한다. 엔트로피 디코딩 유닛 (70)은 인터-예측 (시간 또는 인터-뷰)에 관련된 모션 벡터들 (시간 및/또는 디스패리티) 및 다른 신택스 엘리먼트들을 모션 보상 유닛 (72)으로 포워딩한다. 비디오 디코더 (30)는 신택스 엘리먼트들을 비디오 슬라이스 레벨 및/또는 비디오 블록 레벨에서 수신할 수도 있다.

[0083]

비디오 슬라이스가 인트라-코딩된 (I) 슬라이스로서 코딩될 때, 인트라 예측 유닛 (74)은 시그널링된 인트라 예측 모드 및 현재 프레임 또는 화상의 이전에 디코딩된 블록들로부터의 데이터에 기초하여, 현재 비디오 슬라이스의 비디오 블록에 대한 예측 데이터를 생성할 수도 있다. 비디오 프레임이 인터-코딩된 (즉, 시간 B, 시간 P, 또는 인터-뷰) 슬라이스로서 코딩될 때, 모션 보상 유닛 (72)은 엔트로피 디코딩 유닛 (70)으로부터 수신된 모션 벡터들 및 다른 신택스 엘리먼트들에 기초하여, 현재 비디오 슬라이스의 비디오 블록에 대한 예측 블록들을 생성한다. 예측 블록들은 참조 화상 리스트들 중 하나 내 참조 화상을 중 하나로부터 생성될 수도 있다. 비디오 디코더 (30)는 디플트 구성 기법들을 이용하여, 참조 화상 메모리 (82)에 저장된 참조 화상들에 기초하여, 참조 프레임 리스트들, 즉, 리스트 0 및 리스트 1를 구성할 수도 있다.

[0084]

모션 보상 유닛 (72)은 모션 벡터들 및 다른 신택스 엘리먼트들을 파싱하여 현재 비디오 슬라이스의 비디오 블록에 대한 예측 정보를 결정하고, 그리고, 그 예측 정보를 이용하여, 디코딩중인 현재 비디오 블록에 대한 예측 블록들을 생성한다. 예를 들어, 모션 보상 유닛 (72)은 그 수신된 신택스 엘리먼트들 중 일부를 이용하여, 비디오 슬라이스의 비디오 블록들을 코딩하는데 사용되는 예측 모드 (예컨대, 인트라-예측, 시간 인터-예측, 또는 인터-뷰 예측), 인터-예측 슬라이스 유형 (예컨대, B 슬라이스, P 슬라이스, 또는 GPB 슬라이스), 슬라이스에 대한 참조 화상 리스트들 중 하나 이상에 대한 구성 정보, 슬라이스의 각각의 인터-인코딩된 비디오 블록에 대한 모션 벡터들, 슬라이스의 각각의 인터-코딩된 비디오 블록에 대한 인터-예측 상태, 및 현재 비디오 슬라이스에서 비디오 블록들을 디코딩하기 위한 다른 정보를 결정한다.

[0085]

본 개시물의 기법들에 따르면, 모션 보상 유닛 (72)은 현재 블록에 대한 모션 정보를 재생하는 방법을 나타내는 신택스 엘리먼트들을 수신할 수도 있다. 모션 보상 유닛 (72)은 현재 블록에 대한 모션 정보를 디코딩하는데 사용되는 모션 정보로부터 공간 또는 시간 이웃하는 블록의 식별자를 나타내는 하나 이상의 신택스 엘리먼트들을 수신할 수도 있다. 이러한 신택스 엘리먼트들은 예를 들어, 병합 인덱스 또는 AMVP 인덱스를 포함할 수도 있다. 현재 블록은 포함 블록 내에 있을 수도 있으며 (예컨대, 여기서 현재 블록은 MER 내 CU 또는 PU, 또는 CU 내 PU를 포함할 수도 있다), 신택스 엘리먼트들에 의해 표시되는 공간 이웃하는 블록은 포함 블록 내에 또한 있는 현재 블록에 이웃하는 블록에 대응할 수도 있다.

[0086]

이러한 경우, 본 개시물의 기법들에 따르면, 모션 보상 유닛 (72)은 식별자 (여기서, 식별자는 신택스 엘리먼트들 중 하나, 예컨대, 병합 인덱스 또는 AMVP 인덱스에 대응할 수도 있다)에 의해 표시된 현재 블록에 이웃하는 블록을, 포함 블록에 이웃하고 포함 블록 외부에 있는 블록으로 대체할 수도 있다. 예를 들어, 식별자가 현재 블록의 모서리에서의 이웃하는 블록에 대응하면, 모션 보상 유닛 (72)은 포함 블록의 대응하는 모서리에서의 블록을, 예컨대, 도 8에 대해 아래에서 설명되는 바와 같이 대체할 수도 있다. 이의 대안으로, 식별자가 현재 블록의 모서리에서의 이웃하는 블록에 대응하면, 모션 보상 유닛 (72)은 예컨대, 도 10에 대해 아래에서 설명되는 바와 같이, 유사한 수평 또는 수직 포지션에 있으나, 수직으로 또는 수평으로 쉬프트된 블록이 현재 블록의 외부에 있도록 대체할 수도 있다. 일반적으로, 비디오 디코더 (30)는 모션 정보가 적절히 디코딩될 수 있도록, 비디오 인코더 (20)에 의해 이용되는 방법과 동일한 방법으로, 이러한 현재 블록에 이웃하는 블록들을 포함 블록에 이웃하는 블록들로 대체하도록 구성될 수도 있다.

[0087]

좀 더 자세하게 설명하면, 모션 보상 유닛 (72)은 모션 정보가 취출되어질 수도 있는 하나 이상의 후보 이웃하는 블록들 (공간 또는 시간)을 포함하는 후보 리스트를 조합할 수도 있다. 본 개시물의 기법들에 따르면, 비디오 디코더 (30)는 현재 블록에 이웃하는 블록들이 현재 블록을 포함하는 포함 블록 내에 있을 때, 후보 리스트에서, 현재 블록에 이웃하는 블록들을 포함 블록에 이웃하는 블록들로, 대체할 수도 있다. 예측 모션 정보 (예컨대, 병합 인덱스 또는 AMVP 인덱스)를 취출할 블록을 식별하는 식별자는 후보 리스트에서의 엔트리에 대응할 수도 있다.

[0088]

포함 블록은 병렬 디코딩 영역을 정의할 수도 있다. 포함 블록 외부에 있는 데이터를 이용하여 (예컨대, 병합 모드 또는 AMVP 모드를 이용하여) 현재 블록에 대한 모션 정보를 예측함으로써, 포함 블록 내 정보는 디코딩 현재 블록에 대한 모션 정보에 앞서 디코딩될 필요가 없다. 이러한 방법으로, 모션 보상 유닛 (72)은 포함 블록 내 2개 이상의 블록들에 대한 모션 정보를 디코딩할 수도 있다. 따라서, 포함 블록은 병렬 디코딩 영역을 정의하는 것으로 지칭될 수도 있다. 즉, 본 개시물의 기법들은 비디오 디코더 (30)로 하여금 포함 블록 내 데이터를 병렬로 디코딩 가능하게 하여, 포함 블록 내 2개 이상의 블록들의 실질적으로 동시적인 디코딩을 가능하게 할 수도 있다.

[0089]

모션 보상 유닛 (72)은 또한 내삽 필터들에 기초하여 내삽을 수행할 수도 있다. 모션 보상 유닛 (72)은 비디오 블록들의 인코딩 동안 비디오 인코더 (20)에 의해 사용되는 것과 같은 내삽 필터들을 이용하여, 참조

블록들의 서브-정수 픽셀들에 대해 내삽된 값들을 계산할 수도 있다. 이 경우, 모션 보상 유닛 (72)은 수신된 신택스 엘리먼트들로부터 비디오 인코더 (20)에 의해 사용되는 내삽 필터들을 결정하고 그 내삽 필터들을 이용하여 예측 블록들을 생성할 수도 있다.

[0090] 역양자화 유닛 (76)은 비트스트림으로 제공되어 엔트로피 디코딩 유닛 (70)에 의해 디코딩되는 양자화된 변환 계수들을 역양자화한다, 즉 양자화 해제한다. 역양자화 프로세스는 양자화의 정도를 결정하기 위해, 그리고, 이와 유사하게, 적용되어야 하는 역양자화의 정도를 결정하기 위해, 비디오 슬라이스에서의 각각의 비디오 블록에 대한, 비디오 디코더 (30)에 의해 계산된 양자화 파라미터 QPy의 사용을 포함할 수도 있다. 역변환 유닛 (78)은 역 DCT, 역 정수 변환, 또는 개념적으로 유사한 역변환 프로세스와 같은 역변환을 변환 계수들에 적용하여, 픽셀 도메인에서 잔차 블록들을 생성한다.

[0091] 모션 보상 유닛 (72)이 모션 벡터들 및 다른 신택스 엘리먼트들에 기초하여 현재 비디오 블록에 대한 예측 블록을 생성한 후, 비디오 디코더 (30)는 역변환 유닛 (78)으로부터의 잔차 블록들을 모션 보상 유닛 (72)에 의해 생성된 대응하는 예측 블록들과 합산함으로써, 디코딩된 비디오 블록을 형성한다. 합산기 (80)는 이 합산 동작을 수행하는 구성요소 또는 구성요소들을 나타낸다. 원활 경우, 블록킹 현상 아티팩트들 (blockiness artifacts)을 제거하기 위해 디블록킹 필터가 또한 그 디코딩된 블록들을 필터링하는데 적용될 수도 있다. (코딩 루프 중에 또는 코딩 루프 이후에) 다른 루프 필터들이 또한 픽셀 전환들 (pixel transitions)을 평활화하거나 또는 아니면 비디오 품질을 향상시키기 위해 사용될 수도 있다. 주어진 프레임 또는 화상에서 디코딩된 비디오 블록들은 그후 참조 화상 메모리 (82)에 저장되며, 이 메모리는 후속 모션 보상을 위해 사용되는 참조 화상들을 저장한다. 참조 화상 메모리 (82)는 또한 도 1의 디스플레이 디바이스 (32)와 같은 디스플레이 디바이스 상에의 추후 프리젠테이션을 위해, 디코딩된 비디오를 저장한다.

[0092] 이러한 방법으로, 도 3의 비디오 디코더 (30)는 현재 블록에 이웃하는 블록이 또한 포함 블록 내에 있다는 결정에 기초하여, 포함 블록 내에 있는 비디오 데이터의 현재 블록이 디스패리티 모션 벡터를 이용하여 인코딩되는 것을 결정하고, 후보 리스트에서 이웃하는 블록을, 포함 블록 외부에 있고 포함 블록에 이웃하는 블록으로 대체하고, 후보 리스트에서 복수의 블록들 중 하나로부터 디스패리티 모션 벡터 예측자를 선택하고, 그리고 디스패리티 모션 벡터 예측자에 기초하여 디스패리티 모션 벡터를 디코딩하도록 구성된 비디오 디코더의 일 예를 나타낸다.

[0093] 도 4는 예시적인 MVC 예측 패턴을 예시하는 개념도이다. 멀티-뷰 비디오 코딩 (MVC)은 ITU-T H.264/AVC의 확장판이다. 유사한 기법이 HEVC에 적용될 수도 있다. 도 4의 예에서, (뷰 ID들 "S0" 내지 "S7"를 갖는) 8개의 뷰들이 예시되며, 12개의 시간 로케이션들 ("T0" 내지 "T11")이 각각의 뷰에 대해 예시된다. 즉, 도 4에서 각각의 로우는 뷰에 대응하지만, 각각의 칼럼은 시간 로케이션을 나타낸다.

[0094] MVC가 H.264/AVC 디코더들에 의해 디코딩 가능한 소위, 베이스 뷰를 갖고 또한 스테레오 뷰 쌍이 MVC에 의해 억제될 수 있지만, MVC의 하나의 이점은 2개보다 많은 뷰들을 3D 비디오 입력으로 이용하고 다수의 뷰들에 의해 표현되는 이 3D 비디오를 디코딩하는 예를 지원할 수 있다는 점이다. MVC 디코더를 갖는 클라이언트의 렌더러는 다수의 뷰들을 가진 3D 비디오 콘텐츠를 기대할 수도 있다.

[0095] 전형적인 MVC 디코딩 순서는 시간-우선 코딩으로서 지칭된다. 액세스 유닛은 하나의 출력 시간 인스턴스에 대한 모든 뷰들의 코딩된 화상들을 포함할 수도 있다. 예를 들어, 시간 T0의 화상들의 각각은 공통 액세스 유닛에 포함될 수도 있으며, 시간 T1의 화상들의 각각은 제 2의, 공통 액세스 유닛 등에 포함될 수도 있다. 디코딩 순서는 출력 또는 디스플레이 순서와 반드시 동일하지는 않다.

[0096] 도 4에서의 프레임들은 대응하는 프레임이 인트라-코딩되는지 (즉, I-프레임), 또는 (즉, P-프레임과 같이) 하나의 방향으로 또는 (즉, B-프레임과 같이) 다수의 방향들로 인터-코딩되는지를 나타내는 문자를 포함하는 음영 처리된 블록을 이용하여, 도 4에서 각각의 로우 및 각각의 칼럼의 교차점에 표시된다. 일반적으로, 예측들은 화살표들로 표시되며, 여기서, 지시도달 프레임은 예측 참조를 위해 지시출발 오브젝트를 이용한다. 예를 들어, 시간 로케이션 T0에서 뷰 S2의 P-프레임은 시간 로케이션 T0에서 뷰 S0의 I-프레임으로부터 예측된다.

[0097] 단일 뷰 비디오 인코딩에서와 같이, 멀티뷰 비디오 코딩 비디오 시퀀스의 프레임들은 상이한 시간 로케이션들에서 프레임들에 대해 예측 인코딩될 수도 있다. 예를 들어, 시간 로케이션 T1에서 뷰 S0의 b-프레임은 시간 로케이션 T0에서의 뷰 S0의 I-프레임으로부터 자신에게 지시되는 화살표를 가지며, 이 화살표는 b-프레임이 I-프레임으로부터 예측된다는 것을 나타낸다. 게다가, 그러나, 멀티뷰 비디오 인코딩의 상황에서, 프레

임들은 인터-뷰 예측될 수도 있다. 즉, 뷰 성분은 참조를 위해 다른 뷰들에서의 뷰 성분들을 이용할 수도 있다. MVC에서, 예를 들어, 인터-뷰 예측은 마치 또 다른 뷰에서의 뷰 성분이 인터-예측 참조인 것처럼 실현된다. 잠재적인 인터-뷰 참조들은 시퀀스 파라미터 세트 (SPS) MVC 확장판으로 시그널링되며, 인터-예측 또는 인터-뷰 예측 참조들의 유연한 순서정렬을 가능하게 하는 참조 화상 리스트 구성 프로세스에 의해 수정될 수 있다.

[0098] H.264/AVC의 MVC 확장판에서, 일 예로서, 인터-뷰 예측은 디스패리티 모션 보상에 의해 지원되며, 그 디스패리티 모션 보상은 H.264/AVC 모션 보상의 선택을 이용하지만, 상이한 뷰에서의 화상으로 하여금 참조 화상으로서 사용될 수 있게 한다. 2개의 뷰들의 코딩은 입체적인 뷰들로서 일반적으로 지정되는 MVC에 의해 지원될 수 있다. MVC의 이점들 중 하나는 MVC 인코더가 2개보다 많은 뷰들을 3D 비디오 입력으로서 취할 수 있고 MVC 디코더가 이러한 멀티뷰 표현을 디코딩할 수 있다는 점이다. 따라서, MVC 디코더를 가진 렌더링 디바이스는 3D 비디오 콘텐츠를 2개보다 많은 뷰들로 예상할 수도 있다.

[0099] MVC에서, 인터-뷰 예측 (IVP)은 동일한 액세스 유닛에서의 (즉, 동일한 시간 인스턴스를 가진) 화상을 사이에 허용된다. 액세스 유닛은 일반적으로, 공통 시간 인스턴스에 대한 모든 뷰 성분들 (예컨대, 모든 NAL 유닛들)을 포함하는 데이터의 유닛이다. 따라서, MVC에서, 인터-뷰 예측은 동일한 액세스 유닛에서의 화상을 사이에 허용된다. 비-베이스 뷰들 중 하나에서의 화상을 코딩할 때, 상이한 뷰에 있지만 동일한 시간 인스턴스를 (예컨대, 동일한 POC 값, 따라서, 동일한 액세스 유닛에서) 가지면, 화상이 참조 화상 리스트에 추가될 수도 있다. 인터-뷰 예측 참조 화상은 마치 유사한 임의의 인터 예측 참조 화상처럼 참조 화상 리스트의 임의의 위치에 삽입될 수도 있다.

[0100] 멀티-뷰 비디오 코딩의 상황에서, 2종류의 모션 벡터들이 존재한다. 하나는 시간 참조 화상을 가리키는 정규 모션 벡터들이고, 대응하는 인터 예측 모드는 모션 보상되는 예측 (MCP)로서 지정된다. 다른 하나는 상이한 뷰에서의 화상들을 가리키는 디스패리티 모션 벡터들이며, 대응하는 인터-뷰 예측 모드는 디스패리티-보상되는 예측 (DCP)로서 지정된다.

[0101] 종래의 HEVC에서는, 모션 파라미터들의 예측을 위한 2개의 모드들이 존재하는데: 하나는 병합 모드이고, 다른 하나는 진보된 모션 벡터 예측 (AMVP)이다. 병합 모드에서, 모션 파라미터들 (참조 화상들, 및 모션 벡터들)의 후보 리스트가 구성되며, 여기서, 후보는 공간 또는 시간 이웃하는 블록들로부터 유래할 수 있다. 공간적으로 및 시간적으로 이웃하는 블록들은 후보 리스트, 즉, 모션 예측 정보가 선택될 수도 있는 후보들의 세트를 형성할 수도 있다. 따라서, 모션 예측 정보로서 선택된 모션 파라미터들은 후보 리스트로의 인덱스를 코딩함으로써 시그널링될 수도 있다. 디코더 측에서, 일단 인덱스가 디코딩되면, 인덱스가 가리키는 대응하는 블록의 모든 모션 파라미터들이 병합 모드에서 상속될 수도 있다.

[0102] AMVP, 따라서 종래의 HEVC에서, 각각의 모션 가설에 대한 모션 벡터 예측자들의 후보 리스트는 코딩된 참조 인덱스에 기초하여 유도된다. 이 리스트는 시간 참조 화상에서 동일 장소에 배치된 블록의 이웃하는 블록의 모션 파라미터들에 기초하여 유도되는 시간 모션 벡터 예측자 뿐만 아니라, 동일한 참조 인덱스와 연관되는 이웃하는 블록들의 모션 벡터들을 포함한다. 선택된 모션 벡터들은 후보 리스트로의 인덱스를 송신함으로써 시그널링된다. 게다가, 참조 인덱스 값들 및 모션 벡터 차이들이 또한 시그널링된다.

[0103] 게다가, 2개의 모션 데이터 압축 방법들은 HEVC에 정의되어 있다. 이를 방법들은 아래에서 도 6에 대해 더욱 자세히 설명된다.

[0104] 도 4는 인터-뷰 예측의 여러 예들을 제공한다. 뷰 S1의 프레임들, 도 4의 예에서, 뷰 S1의 상이한 시간 로케이션들에서의 프레임들로부터 예측될 뿐만 아니라, 동일한 시간 로케이션들에서의 뷰들 S0 및 S2의 프레임들의 프레임들로부터 인터-뷰 예측되는 것으로 예시된다. 예를 들어, 시간 로케이션 T1에서 뷰 S1의 b-프레임은 시간 로케이션들 T0 및 T2에서의 뷰 S1의 B-프레임들의 각각 뿐만 아니라, 시간 로케이션 T1에서의 뷰들 S0 및 S2의 b-프레임들로부터 예측된다.

[0105] 도 4의 예에서, 대문자 "B" 및 소문자 "b"는 상이한 인코딩 방법론들보다는, 프레임들 사이의 상이한 계층적 관계들을 나타내도록 의도된다. 일반적으로, 대문자 "B" 프레임들은 소문자 "b" 프레임들보다 예측 계층에서 상대적으로 더 높다. 도 4는 또한 상이한 레벨들의 음영을 이용한 예측 계층에서의 변형 예들을 예시하며, 여기서, 더 많은 양의 음영친 (즉, 상대적으로 더 어두운) 프레임들이 예측 계층에서 더 적은 음영을 갖는 그들 프레임들보다 더 높다 (즉, 상대적으로 더 밝다). 예를 들어, 도 4에서의 모든 I-프레임들은 풀 음영으로 예시되며, P-프레임들은 다소 밝은 음영을 가지며, B-프레임들 (및 소문자 b-프레임들)은 서로에 대해 여

러 레벨들의 음영을 가지나, P-프레임들 및 I-프레임들의 음영보다 항상 더 밝다.

[0106] 일반적으로, 예측 계층은, 계층에서 상대적으로 더 높은 그들 프레임들이 계층에서 상대적으로 낮은 프레임들의 디코딩 동안 참조 프레임들으로 사용될 수 있도록, 예측 계층에서 상대적으로 더 높은 프레임들이 계층에서 상대적으로 낮은 디코딩 프레임들 이전에 디코딩되어야 한다는 점에서, 뷰 순서 인덱스들에 관련된다. 뷰 순서 인덱스는 액세스 유닛에서의 뷰 성분들의 디코딩 순서를 나타내는 인덱스이다. 뷰 순서 인덱스들은 H.264/AVC (MVC 수정안) 의 부록 H에 규정된 바와 같은, SPS MVC 확장판에 시사되어 있다. SPS에서, 각각의 인덱스  $i$ 에 대해, 대응하는  $view\_id$ 가 시그널링된다. 일부 예들에서, 뷰 성분들의 디코딩 구성요소들은 뷰 순서 인덱스의 오름 차순을 따를 것이다. 모든 뷰들이 제시되면, 뷰 순서 인덱스들은 0으로부터  $num\_views\_minus\_1$  까지 연속되는 순서이다.

[0107] 이러한 방법으로, 참조 프레임들로서 사용된 프레임들은 참조 프레임들을 참조하여 인코딩된 프레임들을 디코딩하기 전에 디코딩될 수도 있다. 뷰 순서 인덱스는 액세스 유닛에서의 뷰 성분들의 디코딩 순서를 나타내는 인덱스이다. 각각의 뷰 순서 인덱스  $I$ 에 대해, 대응하는  $view\_id$ 가 시그널링된다. 뷰 성분들의 디코딩은 뷰 순서 인덱스들의 오름 차순을 따른다. 모든 뷰들이 제시되면, 뷰 순서 인덱스들의 세트는 제로로부터 뷰들의 전체 개수 하나 미만까지의 연속적으로 순서화된 세트를 포함할 수도 있다.

[0108] 계층의 동일한 레벨들에서의 어떤 프레임들에 대해, 디코딩 순서는 서로에 대해 문제가 되지 않을 수도 있다. 예를 들어, 시간 로케이션 T0에서 뷰 S0의 I-프레임은 시간 로케이션 T0에서 뷰 S2의 P-프레임에 대한 참조 프레임으로서 사용되며, 그 다음으로, 시간 로케이션 T0에서 뷰 S4의 P-프레임에 대한 참조 프레임으로서 사용된다. 따라서, 시간 로케이션 T0에서 뷰 S0의 I-프레임은, 시간 로케이션 T0에서 뷰 S4의 P-프레임 이전에 디코딩되어야 하는, 시간 로케이션 T0에서 뷰 S2의 P-프레임 이전에 디코딩되어야 한다. 그러나, 뷰 S1과 뷰 S3 사이에, 디코딩 순서는, 뷰들 S1 및 S3가 예측을 위해 서로에 의존하지 않고 대신, 예측 계층에서 더 높은 뷰들로부터 오직 예측되기 때문에, 문제되지 않는다. 더욱이, 뷰 S1은 뷰 S1이 뷰들 S0 및 S2 이후에 디코딩되는 한, 뷰 S4 이전에 디코딩될 수도 있다.

[0109] 이러한 방법으로, 계층적 순서정렬이 뷰들 S0 내지 S7을 기술하기 위해 사용될 수도 있다. 뷰 SA가 뷰 SB 이전에 디코딩되어야 한다는 것을 표기 SA > SB가 의미한다고 하자. 이 표기를 이용하면, 도 4의 예에서, S0 > S2 > S4 > S6 > S7이다. 또한, 도 4의 예에 대해, S0 > S1, S2 > S1, S2 > S3, S4 > S3, S4 > S5, 및 S6 > S5이다. 이들 요구사항들을 위반하지 않는 뷰들에 대한 임의의 디코딩 순서가 가능하다. 따라서, 많은 상이한 디코딩 순서들이 단지 어떤 한계를 하에서 가능하다.

[0110] 본 개시물의 기법들에 따르면, 비디오 인코더(20) 및 비디오 디코더(30)는 블록(예컨대, 포함 블록) 내 데이터를 별별로 코딩하도록 구성될 수도 있다. 이를 달성하기 위해, 비디오 인코더(20) 및 비디오 디코더(30)는 포함 블록 내 데이터를 이용하여 포함 블록 내 다른 데이터를 예측하도록 구성될 수도 있다. 예를 들어, 비디오 인코더(20) 및 비디오 디코더(30)는 현재 블록 외부에 있는 블록들로부터의 예측 모션 정보를 이용하여, 포함 블록 내, 현재 블록의 모션 정보를 코딩하도록 구성될 수도 있다. 따라서, 비디오 인코더(20) 및 비디오 디코더(30)는 예컨대, 시간 및/또는 디스패리티 모션 벡터들을 코딩하기 위한 모션 예측 후보들을 대체할 수도 있다. 이러한 방법으로, 비디오 인코더(20) 및 비디오 디코더(30)는 본 개시물의 기법들을 이용하여, 예컨대, 도 4의 시간 및/또는 디스패리티 모션 벡터들을 인코딩할 수도 있다.

[0111] 도 5는 현재 PU 및 공간적으로 이웃하는 블록들( $A_0, A_1, B_0, B_1$ , 및  $B_2$ )의 일 예를 나타내는 블록도이다. 도 5는 현재 PU와 그의 공간적으로 이웃하는 PU들 사이의 관계를 도시한다. 이 설명의 목적들을 위해, 다음 심볼들이 정의된다:

[0112] 루마 로케이션( $xP, yP$ ) (즉, 휘도 성분에서) 이 현재 화상의 좌상측 샘플에 대한, 현재 PU의 좌상측 루마 샘플을 규정하는데 사용될 수도 있다.

[0113] 변수들 nPSW 및 nPSH는 루마에 대한, 즉, 휘도 성분에서 현재 PU의 폭 및 높이를 표시할 수도 있다.

[0114] 이 예에서, 현재 화상의 좌상측 샘플에 대한, 현재 예측 유닛 N의 좌상측 루마 샘플은( $xN, yN$ )이다. 그러면, ( $xN, yN$ ) (여기서, N은  $A_0, A_1, B_0, B_1$  또는  $B_2$ 로 대체됨)는 ( $xP - 1, yP + nPSH$ ), ( $xP - 1, yP + nPSW - 1$ ), ( $xP + nPSW, yP - 1$ ), ( $xP + nPSW - 1, yP - 1$ ) 또는 ( $xP - 1, yP - 1$ )로서 각각 정의될 수도 있다.

[0115] 위에서 언급한 바와 같이, 일부 예들에서, 비디오 인코더(20) 또는 비디오 디코더(30)와 같은, 비디오 코더

는, 시간 모션 벡터 예측을 수행할 수도 있다. 즉, 현재 PU에 대한 모션 벡터는 시간 모션 벡터 예측자(TMVP)와 관련하여 코딩될 수도 있다. TMVP를 획득하기 위해, 먼저, 비디오 코더는 동일 장소에 배치된 화상(예컨대, 현재 화상과 동일한 액세스 유닛 내 화상)을 식별할 수도 있다. 현재 화상의 슬라이스가 B 슬라이스이면, collocated\_from\_10\_flag는 동일 장소에 배치된 화상이 RefPicList0 또는 RefPicList1로부터 유래하는지 여부를 나타내기 위해 슬라이스 헤더에서 시그널링될 수도 있다.

[0116] 게다가, B 슬라이스에 대해, 결합된 리스트(RefPicListC)는 최종 참조 화상 리스트들(RefPicList0 및 RefPicList1)가 구성되어진 이후에 구성될 수도 있다. 결합된 리스트는 참조 화상 리스트 변경 신택스이 결합된 리스트에 제시되면, 추가로 변경될 수 있다.

[0117] 참조 화상 리스트가 식별된 후, 슬라이스 헤더에서 시그널링된, collocated\_ref\_idx가 리스트에서의 화상에서 화상을 식별하는데 사용될 수도 있다. 동일 장소에 배치된 PU가 그후 동일 장소에 배치된 화상을 체킹함으로써 식별될 수도 있다. 이 PU를 포함하는 CU의 우하측 PU의 모션, 또는 이 PU를 포함하는 CU의 중심 PU들 내 우하측 PU의 모션이 사용될 수도 있다.

[0118] 상기 프로세스에 의해 식별되는 모션 벡터들이 AMVP 또는 병합 모드에 대한 모션 후보를 생성하는데 사용될 때, 그들은(화상들에 대한 POC(화상 순서 카운트) 값들에 의해 반영되는) 시간 로케이션에 기초하여 스케일링될 수도 있다. HEVC에서, 화상 파라미터 세트(PPS)는 플래그 enable\_temporal\_mvp\_flag를 포함한다. 0과 동일한 temporal\_id를 가진 특정의 화상이 0과 동일한 enable\_temporal\_mvp\_flag를 갖는 PPS를 지칭할 때, DPB에서의 모든 참조 화상들은 "시간 모션 벡터 예측용으로 미사용됨"으로 표시될 수도 있으며, 디코딩 순서에서 그 특정의 화상 이전 화상들로부터의 어떤 모션 벡터도 디코딩 순서에서 특정의 화상 이후 특정의 화상 또는 화상의 디코딩에서 시간 모션 벡터 예측자로서 사용되지 않을 것이다.

[0119] 도 6은 상이한 유형들인, 여러 화상들의 그룹들(GOPs)의 화상들을 예시하는 개념도이다. 특히 도 6은 이전 GOP(100), 현재 GOP(102), 및 다음 GOP(104)를 나타낸다. 이전 GOP(100)는 화상들(106, 108)을 포함한다. 현재 GOP(102)는 화상들(110-124)을 포함한다(여기서, 도 6의 예에서, 화상(124)은 깨끗한 무작위 액세스 화상(CRA)의 일 예를 나타낸다). 다음 GOP(104)는 화상들(126, 128)을 포함한다.

[0120] 화상들(106-128)의 각각은 대응하는 화상의 POC 값을 나타내는 숫자 값을 포함한다. 디코딩 순서는 POC 값들으로 표시된 바와 같은 디스플레이 순서와 상이할 수도 있다는 점에 유의해야 한다. 도 6에서 화살표들은 잠재적인 예측들을 나타낸다. 특히, 화살표가 가리키는 화상은 화살표가 출발하는 화상에 의해 참조 화상으로서 사용될 수도 있다. 예를 들어, (POC 값 16을 가진) 화상(108)은 (POC 값 20을 가진) 화상(116)을 예측하기 위한 참조 화상으로서 사용될 수도 있다.

[0121] HEVC에서, 네트워크 추상화 계층(NAL) 유닛 유형에 의해 식별될 수 있는 4개의 화상 유형들이 존재한다. 이들은 동시 디코딩 리프레시(IDR) 화상, 깨끗한 무작위 액세스(CRA) 화상, 시간 계층 액세스(TLA) 화상, 및 IDR, CRA 또는 TLA 화상이 아닌 코딩된 화상이다. 이 예에서, POC 값 24를 가진 화상은 CRA 화상이다.

[0122] IDR 및 코딩된 화상들은 H.264/AVC 사양으로부터 실질적으로 상속되는 화상 유형들이다. CRA 및 TLA 화상 유형들은 HEVC에서 새로운 것이며 H.264/AVC 사양에서 이용불가능하다. CRA 화상은 IDR 화상들을 삽입하는 것보다 더 효율적인 비디오 시퀀스의 중간에서 임의의 무작위 액세스 지점으로부터 시작하는 디코딩을 용이하게 하는 화상 유형이다. HEVC에서, 이들 CRA 화상들로부터 시작하는 비트스트림은 또한 적합 비트스트림들(conforming bitstreams)이다. TLA 화상은 유효한 시간 계층 스위칭 지점들을 나타내기 위해서 사용될 수 있는 화상 유형이다.

[0123] (예컨대, 네트워크를 통한, 예컨대 HTTP(DASH)를 통한 동적 적응 스트리밍, 실시간 전송 프로토콜(RTP) 또는 실시간 스트리밍 프로토콜(RTSP)에 따른) 브로드캐스팅 및 스트리밍과 같은, 비디오 애플리케이션들에서는, 사용자들이 상이한 채널들 사이에서 스위칭하고 최소 지역으로 비디오의 특정의 부분들로 점프하는 중요한 특징이 요구된다. 이 특징은 비디오 비트스트림들에서 규칙적인 간격들로 무작위 액세스 화상들을 가짐으로써 사용가능하게 될 수도 있다. H.264/AVC 및 HEVC 양쪽에 규정된, IDR 화상이 무작위 액세스용으로 사용될 수 있다. 그러나, IDR 화상이 코딩된 비디오 시퀀스를 시작하고 항상 디코딩 화상 버퍼(DPB)를 청소하기 때문에, 디코딩 순서에서 IDR 다음 화상들은 IDR 화상 이전에 디코딩된 화상들을 참조로서 이용할 수 없다. 그 결과, 무작위 액세스를 위한 IDR 화상들에 의존하는 비트스트림들은 현저하게 낮은 코딩 효율(예컨대, 6%)을 가질 수 있다. 코딩 효율을 향상시키기 위해, HEVC에서 CRA 화상들은 디코딩 순서에서 CRA 화상에 뒤따

르지만 출력 순서에서 그것에 선행하는 화상들로 하여금, CRA 화상 이전에 디코딩된 화상들을 참조 화상들로서 이용할 수 있게 한다.

[0124] CRA 화상 (화상 (124)) 둘레의 예시적인 예측 구조가 도 6에 도시된다. (POC 24를 가진) CRA 화상 (124)은 디코딩 순서에서 CRA 화상 (124)을 뒤따르며 출력 순서에서 CRA 화상 (124)보다 선행하는, (POC 값들 17 내지 23을 가진) 다른 화상들 (110-122)을 포함하는 현재 GOP (102)에 속한다. 즉, 도 6의 예에서 (POC 값들 17 내지 23을 가지는) 화상들 (110-122)이 화상 (124)으로부터, 또는 화상 (124)으로부터 결국 의존하는 화상으로부터 의존할 수도 있기 때문에, 화상 (124)은 화상들 (110-122)에 앞서 디코딩되어야 한다. 화상들 (110-122)은 따라서 CRA 화상 (124)의 선두 화상들로 지칭되며, 디코딩이 현재 CRA 화상 이전에 IDR 또는 CRA 화상으로부터 시작하면 정확하게 디코딩될 수 있다. 그러나, 선두 화상들은 CRA 화상 (124)으로부터의 무작위 액세스가 일어날 때 정확히 디코딩될 수 없다. 즉, 도 6의 예에 나타낸 바와 같이, 선두 화상들은 화상 (108)과 같은, 비트스트림에서의 이전 화상들로부터 의존할 수도 있다. 그러므로, 선두 화상들 (110-122)은 무작위 액세스 디코딩 동안 일반적으로 폐기된다.

[0125] 디코딩이 시작하는 장소에 따라서 이용불가능할 수도 있는 참조 화상들로부터의 에러 전파를 방지하기 위해, 디코딩 순서 및 출력 순서 양쪽에서, CRA 화상 (124)을 뒤따르는, 도 6에 나타낸 바와 같은 다음 GOP (104)에서의 모든 화상들은, 참조를 위해 (POC 값들 17-23을 가진 선두 화상들 (110-122)을 포함하는) 디코딩 순서 또는 출력 순서에서 CRA 화상 (124)보다 선행하는 임의의 화상을 이용불가능하도록, 코딩될 수도 있다.

[0126] 유사한 무작위 액세스 기능들이 H.264/AVC에서 복구 지점 SEI 메시지에 의해 지원된다. H.264/AVC 디코더 구현예는 이 기능을 지원하거나 또는 지원하지 않을 수도 있다. HEVC에서, CRA 화상에서 시작하는 비트스트림은 적합 비트스트림으로서 간주된다. 비트스트림이 CRA 화상에서 시작할 때, CRA 화상의 선두 화상들은 이용불가능한 참조 화상들을 참조할 수도 있으며, 따라서 정확히 디코딩되지 않을 수 있다. 그러나, HEVC는 시작 CRA 화상의 선두 화상들이 출력되지 않는다는 것을, 따라서 이름 "깨끗한 무작위 액세스"를 규정한다. 비트스트림 순응성 요구사항의 확립을 위해, HEVC는 비-출력 선두 화상들의 디코딩에 이용불가능한 참조 화상들을 생성하는 디코딩 프로세스를 규정한다. 그러나, 적합 디코더 구현예들은, 디코딩 프로세스가 비트스트림의 시작으로부터 수행될 때와 비교하여 동일한 출력을 생성하는 한, 그 디코딩 프로세스를 따르도록 할 필요가 없다.

[0127] HEVC에서, 적합 비트스트림은 거의 IDR 화상들을 포함하지 않으며, 그 결과, 코딩된 비디오 시퀀스 또는 불완전한 코딩된 비디오 시퀀스의 서브세트를 포함할 수도 있다는 점에 유의할 가치가 있다.

[0128] 도 7은 블록들의 라인에서 모션 데이터 압축을 예시하는 개념도이다. 특히, 블록들은  $16 \times 16$  블록 경계들 (150A-150E)에 의해 분할된다. 도 7의 예에서, 음영처리된 블록들에 대한 모션 정보는 라인 베퍼에 저장될 수도 있다. 미음영처리된 (Unshaded) 블록들은 이웃하는 블록의 모션 정보를 공유할 수도 있다. 따라서, 블록들 (152A-152I)의 쌍들의 각각은 모션 정보를 공유할 수도 있다. HEVC에서 모션 벡터 스토리지 압축 방법은 병합/스킵 모드 및 AMVP에 대한 시간 모션 벡터들을 저장하고 로드하는데 소요되는 메모리 대역폭을 감소시키도록 설계된다.  $16 \times 16$  압축 방법은 기본적으로 모션 벡터 필드를 4의 인자 만큼 수평 및 수직 방향 양쪽으로 다운-샘플링하여, 각각의  $16 \times 16$  영역에 대해 동일한 모션 벡터를 초래한다.

[0129] 모션 데이터는 일반적으로 (예컨대, 빠른 디스플레이 시간 (List0)을 가진 화상으로, 늦은 디스플레이 시간 (List1)을 가진 화상들으로, 또는 양방향으로의) 인터 예측 방향, 참조 화상 인덱스 (예컨대, 참조 화상 리스트로의 인덱스), 그리고, 수평 및 수직 성분을 포함할 수도 있는 모션 벡터 (MV)를 포함한다. 상부 LCU로 우의 모션 데이터에 액세스하는데 요구되는 각각의 PU에 대해, 단지 라인 베퍼에서 압축된 모션 데이터가 액세스될 수 있다. 도 7은 하나의 라인의 2:1 모션 데이터 압축을 나타낸다. 모든 4개의  $4 \times 4$  블록들에 대해, 제 1 및 최종  $4 \times 4$  블록들의 모션 데이터는 라인 베퍼에 저장된다. 제 1 블록의 모션 데이터는 처음 2개의 블록들의 모션 데이터를 나타낸다. 최종 블록의 모션 데이터는 최종 2개의 블록들의 모션 데이터를 나타낸다. 이러한 방법으로, 단지 메모리의 절반이 요구된다.

[0130] HEVC에서, LCU는 별별 모션 추정 영역들 (MERs)로 분할될 수도 있으며, 단지 현재 PU 와는 상이한 MERs에 속하는 그들 이웃하는 PU들로 하여금, 병합/스킵 MVP 리스트 구성 프로세스에 포함될 수 있게 할 수도 있다. MER의 사이즈는 화상 파라미터 세트 (PPS)로 log2\_parallel\_merge\_level\_minus2로서 시그널링될 수도 있다. MER 사이즈가  $N \times N$ 보다 클 때, 여기서,  $2N \times 2N$ 은 가장 작은 CU 사이즈이며, MER은 공간 이웃하는 블록은, 현재 PU와 동일한 동일한 MER 내에 있으면, 이용불가능한 것으로 간주되는 방법으로, 적용될 수도 있다.

[0131]

MPEG 에서 HEVC-기반의 3D 비디오 코딩 (3D-HEVC) 코덱은 m22570 (2011년 11월, 스위스, 제네바, ISO/IEC JTC1/SC29/WG11 MPEG2011/m22570, Heiko Schwarz, Christian Bartnik, Sebastian Bosse, Heribert Brust, Tobias Hinz, Haricharan Lakshman, Detlev Marpe, Philipp Merkle, Karsten Muller, Hunn Rhee, Gerhard Tech, Martin Winken, Thomas Wiegand, "Description of 3D Video Technology Proposal by Fraunhofer HHI (HEVC compatible; configuration A)") 및 m22571 에서 제안된 솔루션들에 기초한다. 3D-HEVC 에 대한 참조 소프트웨어의 설명은 2012년 2월, 미국, 산호세, ISO/IEC JTC1/SC29/WG11 MPEG2011/N12559, Heiko Schwarz, Krzysztof Wegner, "Test Model under Consideration for HEVC based 3D video coding" 로서 입수가능하다. HTM 으로서 지칭되는, 참조 소프트웨어는 [https://hevc.hhi.fraunhofer.de/svn/svn\\_3DVCSoftware/trunk](https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/trunk) 에서 입수가능하다.

[0132]

대리인 명부 번호 제 121833U1/1212:178USP1호 및 제 121833U2/1212:178USP2호, 2012년 3월 14일자에 출원된 미국 가출원 번호 제 61/610,961호, 및 2012년 4월 11일자에 출원된 제 61/623,041호, Chen 등에 의한, "Disparity Vector Construction Method for 3D-HEVC" 에는, 공간 디스파리티 벡터들 (SDV), 시간 디스파리티 벡터들 (TDV) 또는 암시적인 디스파리티 벡터들 (IDV; Implicit Disparity Vector)로부터의 디스파리티 벡터 생성 (구성) 방법이 인터-뷰 모션 예측용으로 설명되어 있다. 그 안에 정의된 바와 같이, SDV 는 인터-뷰 참조를 가리키고 있는 현재 PU 의 공간 이웃하는 PU 의 모션 벡터이다. TDV 는 동일한 액세스 유닛을 가진 임의의 참조 화상 또는 인터-뷰 화상에서 동일 장소에 배치된 PU 또는 동일 장소에 배치된 LCU로부터의 디스파리티 모션 벡터일 수도 있다. 이의 대안으로, TMVP 에 사용되는 화상으로부터의 동일 장소에 배치된 PU 의 모션 벡터 또는 TMVP 에 의해 생성된 모션 벡터가 디스파리티 벡터이면, 또한 TDV 로서 취급될 수도 있다. 현재 PU 의 공간 또는 시간 이웃하는 PU 가 인터-뷰 모션 예측에 사용되면, 이웃하는 PU 에 사용된 디스파리티 벡터는 IDV 일 수도 있다. 상기 3종류로부터의 디스파리티 벡터들이 디스파리티 벡터 후보 리스트를 구성하는데 사용될 수도 있으며 그들 중 하나가 일부 기준들에 기초하여 선택될 수도 있으며 IMP 용으로 직접 사용될 수도 있다.

[0133]

인터-뷰 모션 예측은 AMVP 모드 및 병합 모드 양쪽에 적용될 수 있다. 각각의 PU 에 대해, 인터-뷰 모션 예측 (DVIVMP) 을 위한 디스파리티 벡터는 구성된 디스파리티 벡터이며 인터-뷰 모션 예측용으로 사용될 수도 있다. 인터-뷰 모션 예측을 포함하기 위해, AMVP 모드는 인터-뷰 모션 벡터 예측자가 후보 리스트에 추가될 수도 있는 방법으로 확장되었다. 현재 블록의 중간의 샘플에 대한 심도 추정에 기초하여, 디스파리티 벡터 및 참조 뷰에서의 참조 블록은 위에서 설명한 바와 같이 결정될 수도 있다. 현재 블록에 대한 참조 인덱스 가 인터-뷰 참조 화상을 참조하면, 인터-뷰 모션 벡터 예측자는 대응하는 디스파리티 벡터와 동일하게 설정된다. 현재 참조 인덱스가 시간 참조 화상을 참조하고 참조 블록이 현재 참조 인덱스와 동일한 액세스 유닛을 참조하는 모션 가설을 이용하면, 이 모션 가설과 연관되는 모션 벡터는 인터-뷰 모션 벡터 예측자로서 사용될 수도 있다. 다른 경우, 인터-뷰 모션 벡터 예측자는 무효한 것으로 표시되고 모션 벡터 예측자 후보들의 리스트에 포함되지 않을 수도 있다.

[0134]

AMVP 모드와 유사하게, 모션 파라미터들의 후보 리스트는 인터-뷰 모션 예측을 이용하여 획득될 수도 있는 모션 파라미터 세트에 의해 확장될 수도 있다. 각각의 잠재적인 모션 가설에 대해, 참조 화상 리스트의 처음 2개의 참조 인덱스들이 주어진 순서로 조사될 수도 있다. 참조 인덱스 0 에 대한 모션 벡터 후보는 AMVP 모드에 대한 것과 동일한 방법으로 유도될 수도 있다. 유도된 모션 벡터가 유효하면, 참조 인덱스 0 및 유도된 모션 벡터가 고려되는 가설에 사용될 수도 있다. 그렇지 않으면, 참조 인덱스 1 는 동일한 방법으로 테스트 될 수도 있다. 또한 무효한 모션 벡터를 초래하면, 모션 가설은 이용불가능한 것으로 표시될 수도 있다. 시간 예측을 더 선호하기 위해, 참조 인덱스들이 테스트되는 순서는 제 1 인덱스가 인터-뷰 참조 화상을 참조하면 역전될 수도 있다. 모든 잠재적인 모션 가설들이 이용불가능한 것으로 표시되면, 인터-뷰 후보가 선택되지 않을 수 있다.

[0135]

인터-뷰 잔차 예측 (IRP) 이 3D-HEVC 에서 이용될 수 있다. 현재 화상의 대응하는 심도 맵 (추정된 심도 맵 일 수도 있음) 에 기초하여, 디스파리티 벡터가 현재 블록에 대해 결정될 수도 있다. 디스파리티 벡터가 그 후 참조 뷰 (잔차 참조 블록) 에서 잔차 블록을 로케이트하는 사용될 수도 있다. 잔차 예측이 블록에 대해 사용가능하게 될 때, 잔차 참조 블록이 이 블록에 대한 모션 보상된 예측자 및 시그널링된 잔차에 더하여, 추가될 수도 있다.

[0136]

2012년 4월 9일자에 출원된, 미국 가출원번호 제 61/621,929호, 대리인 사건번호 제 122144P1호/1212:203USP1 를 가지는, Chen 등에 의한, "Motion Parameter Prediction and Coding for Multiview Video" 에서, 공간 및 시간 이웃하는 블록들로부터의 디스파리티 벡터 생성 방법이 IMP 및 IRP 양쪽에 대해 설명되어 있다. 솔루

선들 중 하나는 다음과 같이 정의된다.

[0137] 후보 공간 이웃하는 블록들은 AMVP 동안 체크된 것과 동일할 수도 있으며, 그들의 순서는 도 5 에 도시된 바와 같이  $A_0$ ,  $A_1$ ,  $B_0$ ,  $B_1$  또는  $B_2$  이다. 순서화된 공간 이웃하는 블록들 중 하나의 모션 벡터가 디스패리티 모션 벡터이면, 디스패리티 벡터를 생성하는데 사용될 수도 있다. 시간 이웃하는 블록들의 모션 벡터들로부터의 시간 디스패리티 벡터 (TDV) 는 참조 화상들 (후보 화상들로 지칭됨) 의 세트에 기초하여 유도될 수 있으며, 각각의 후보 화상에 대해, 상이한 블록들의 세트가 체크될 수도 있다.

[0138] 후보 화상들은 참조 인덱스의 오름 차순으로 체크될 수도 있다. 하나의 참조 인덱스가 RefPicList0 및 RefPicList1 에서의 2개의 화상들에 대응할 때, RefPicListX 로부터의 화상은 다른 화상보다 선행하며, 여기서, X 는 다른 화상보다 선행하는 collocated\_from\_10\_flag 와 동일하다. 3개의 후보 영역들은 각각의 후보 화상에 대해 결정될 수도 있다. 영역이 다수의 PU들 또는  $16 \times 16$  영역들을 커버할 때, 이러한 후보 영역의 각각의  $16 \times 16$  영역의 PU들 또는 좌상측  $4 \times 4$  블록들은 래스터 스캐닝 순서로 스캐닝될 수도 있다. 그렇지 않으면, 좌상측  $4 \times 4$  블록이 디스패리티 벡터 유도용으로 선택될 수도 있다. CPU 는 현재 PU 의 동일 장소에 배치된 영역에 대응할 수도 있으며; CLCU 는 현재 PU 의 동일 장소에 배치된 영역을 커버링하는 최대 코딩 유닛 (LCU) 에 대응하며; 그리고 BR 은 우하측  $4 \times 4$  블록에 대응할 수도 있다. 디스패리티 벡터 생성 프로세스는 디스패리티 벡터가 IMP 또는 IRP 에 대해 각각 사용될 때, PU 또는 CU 레벨에서 수행될 수도 있다.

[0139] 가장 효율적인 코딩 툴들 중 하나는 인터-뷰 모션 예측 (IMP) 이며, 여기서, 의존적인 뷰에서 블록의 모션 파라미터들이 동일한 액세스 유닛의 다른 뷰들에서의 이미 코딩된 모션 파라미터들에 기초하여 예측되거나 또는 추론된다.

[0140] 도 8 은 PU 를 포함하는 모션 추정 범위 (MER) 외부에 있는 대체물들에 대한 예를 예시하는 개념도이다. 도 8 의 예에서, PU들 (162, 164, 및 166) 은 MER (160) 내에 있다. 즉, MER (160) 은 PU들 (162, 164, 및 166) 을 포함하는 포함 블록의 일 예를 나타낸다. 본 개시물의 기법들에 따르면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 과 같은, 비디오 코더는 공간 디스패리티 벡터 체킹을 수행하도록 구성될 수도 있다. 일부 예들에서, 비디오 코더는 PU 를 코딩할 때 MER 을 이용할 수도 있다. 이러한 방법으로, 비디오 코더는 공간 디스패리티 벡터들을 획득하는데 MER 내부에 있는 블록들을 이용할 수 없을지도 모른다. 이 경우, 대체 블록들이 사용될 수도 있다.

[0141] PU들 (162, 164, 166) 은 도 5 의 블록들 ( $A_0$ ,  $A_1$ ,  $B_0$ ,  $B_1$ , 및  $B_2$ ) 에 대응하는 이웃하는 블록들을 갖는다. 예를 들어, PU (162) 는 이웃하는 블록들 (172A, 176B, 176C, 176D, 및 176E) 에 이웃한다. PU (164) 는, 또 다른 예로서, 이웃하는 블록들 (174A-174E) 에 이웃한다. 또한 또 다른 예로서, PU (166) 는 이웃하는 블록들 (178A-178E) 에 이웃한다. 게다가, MER (160) 은 그 자신의 이웃하는 블록들, 즉, 블록들 (172A-172E) 을 갖는다. 일반적으로, 끝에 문자 "A" 를 갖는 이웃하는 블록들은 도 5 의 블록  $B_2$  에 대응하며, 끝에 문자 "B" 를 갖는 이웃하는 블록들은 도 5 의 블록  $B_1$  에 대응하며, 끝에 문자 "C" 를 갖는 이웃하는 블록들은 도 5 의 블록  $B_0$  에 대응하며, 끝에 문자 "D" 를 갖는 이웃하는 블록들은 도 5 의 블록  $A_1$  에 대응하며, 그리고, 끝에 문자 "E" 를 갖는 이웃하는 블록들은 도 5 의 블록  $A_0$  에 대응한다.

[0142] PU 의  $A_0$  또는  $A_1$  이웃하는 (좌하측) 블록들이, 도 8 에 도시된 바와 같이, (블록들 (174D 및 174E) 와 같이) MER (160) 내부에 있을 때, 그들 블록들은 이용불가능한 것으로 간주될 수도 있다. 따라서 MER (160) (블록들 (172D 및 172E)) 의  $A_0$  또는  $A_1$  이웃하는 블록들은 대체물로서 사용될 수도 있다. 도 8 에서 PU (164) 에 대해 나타낸 바와 같이, PU (164) 의  $A_0$  및  $A_1$  블록들은 블록들 (174D 및 174E) 이며, 그들이 PU (164) 를 또한 포함하는 MER (160) 내에 있기 때문에, 이 예에서 이용불가능한 것으로 간주된다. 따라서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 블록 (174D) 을 블록 (172D) 으로, 그리고, 블록 (174E) 을 블록 (172E) 으로 대체할 수도 있다.

[0143] 이러한 방법으로, 블록들 (174D 및 174E) 은 현재 블록에 좌하측 이웃하는 블록들 (이 예에서는, PU (164)) 의 예들을 나타낸다. 더욱이, 이 예에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 블록들 (174D 및 174E) 을 각각 블록들 (172D 및 172E) 로 대체할 수도 있다. 이 대체는 현재 블록에 좌하측 이웃하는 블록 (예컨대, PU (164) 의 블록들 (174D, 174E)) 을 MER (160) 에 좌측 이웃하는 블록 (예컨대, 블록들 (172D, 172E)) 로 대체하는 일 예를 나타낸다.

[0144] PU 의  $B_0$  또는  $B_1$  이웃하는 (우상측) 블록들이 내부 MER (160) 내부에 있을 때, 그들 블록들은 이용불가능한 것으로 간주될 수도 있다. 예를 들어, PU (166) 에 이웃하는 이웃하는 블록들 (178B, 178C), 및 PU (164) 에 이웃하는 이웃하는 블록들 (174B, 174C) 은 이용불가능한 것으로 간주될 수도 있다. 따라서, MER (160) 의  $B_0$  또는  $B_1$  이웃하는 블록들 (이 예에서, 블록들 (172B, 172C)) 이 대체물들로서 사용될 수도 있다. 도 8 에서 PU (166) 에 대해 나타낸 바와 같이, 블록들 (178B 및 178C) 은 그들이 PU (166) 를 또한 포함하는 MER (160) 내에 있기 때문에, 이 예에서 이용불가능하다. 따라서, 비디오 인코더 (29) 및 비디오 디코더 (30) 는 이웃하는 블록들 (178B, 178C) 을 이웃하는 블록들 (172B, 172C) (MER (160) 의 이웃하는 블록들) 로 대체할 수도 있다. 이와 유사하게, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 PU (164) 에 이웃하는 이웃하는 블록들 (174B, 174C) 을 이웃하는 블록들 (172B, 172C) (MER (160) 의 이웃하는 블록들) 로 대체할 수도 있다.

[0145] 이 대체는, 현재 블록을 또한 포함하는 포함 블록에서 이웃하는 블록 (예컨대, PU들 (164, 166) 중 하나) 가 현재 블록에 우상측 이웃하는 블록 (예컨대, 블록들 (174B, 174C, 178B, 178C) 중 하나) 에 대응할 때, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 현재 블록 (예컨대, PU들 (164, 166) 중 하나) 에 우상측 이웃하는 블록 (예컨대, 블록들 (174B, 174C, 178B, 178C) 중 하나) 를 MER (160) 에 우상측 이웃하는 블록 (예컨대, 블록들 (172B, 172C) 중 하나) 로 대체할 수도 있는, 예를 나타낸다. 이웃하는 블록들 사이에 도 8 에서 도시된 화살표들은 대체들을 나타낸다.

[0146] PU 의  $B_2$  이웃하는 (좌상측) 블록들이 PU (예컨대, PU (164) 에 이웃하는 블록 (174A)) 를 또한 포함하는 MER (160) 내에 있을 때, 이웃하는 블록은 이용불가능한 것으로 간주될 수도 있다. 따라서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 MER (160) 의  $B_2$  이웃하는 블록 (예컨대, 이웃하는 블록 (172A)) 을 PU (164) 의 이웃하는 블록 (174A) 에 대한 대체물로서 대체할 수도 있다. 이 대체는, 이웃하는 블록이 현재 블록에 좌상측 이웃하는 블록에 대응하는 경우, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 현재 블록에 좌상측 이웃하는 블록을 MER 에 좌상측 이웃하는 블록으로 대체할 수도 있는 예를 나타낸다.

[0147] PU 의 모든 이웃하는 블록들이 PU 를 포함하는 MER 외부에 있을 때, 어떤 대체도 요구되지 않는다. 이것은 도 8 의 예에서 PU (162) 에 이웃하는 블록들 (172A, 176B-176E) 의 경우도 마찬가지이다.

[0148] 도 9 는 디스패리티 벡터들이 체크되는 PU (210) 의 일 예를 예시하는 개념도이다. 도 9 는 각각의 위치 정의 아래에서의 화살표들 및 번호들을 이용하여 상대적인 이웃하는 위치들을 예시한다. PU (210) 는 이웃하는 공간 블록들 (212-220) 및 이웃하는 시간 블록 (222) 을 갖는다. 디스패리티 벡터 생성을 수행하는 비디오 코더가 위에서 설명한, Chen 등에 의한, "Motion Parameter Prediction and Coding for Multiview Video" 에서 설명된 바와 같이, 도 9 의 공간 이웃 디스패리티 벡터들, 예컨대, 시간 디스패리티 벡터들에 대해서, 다른 유형들의 디스패리티 벡터들을 체크할 수도 있다는 점에 유의한다. 도 9 의 예는 공간 디스패리티 벡터들이 AMVP 또는 MERGE 에서 체크되는 것들과 동일하게 생성하는, 이웃하는 블록들 (212-220) 로부터 체크될 때의 경우를 기술한다. 이러한 경우, MER 이 사용될 때, 대체들은 도 8 과 관련하여 위에서 설명한 바와 같이, 도입될 수도 있다. 다른 프로세스들은 변경되지 않고 유지될 수도 있다.

[0149] 이의 대안으로, 공간 이웃하는 블록들이 MER 내부에 있을 때, MER 의 경계에 있는 수평으로 또는 수직으로 정렬된 블록들이, MER 내부에 있는 것들을 체크하는 대신 체크될 수도 있다. (블록들  $A_0$ ,  $A_1$  에 각각 대응하는) 블록들 (220, 218) 에 대해, MER 의 좌측 경계에서의 수평으로 정렬된 블록이 대체물로서 사용될 수도 있다. (블록들  $B_0$ ,  $B_1$  에 각각 대응하는) 블록들 (216, 214) 에 대해, MER 의 상부 경계에서의 수직으로 정렬된 블록이 대체물로서 사용될 수도 있다. (블록  $B_2$  에 대응하는) 블록 (212) 에 대해, MER 의 좌측 경계에서의 수평으로 정렬된 블록, 및 MER 의 상부 경계에서의 수직으로 정렬된 블록 양쪽이 대체물로서 사용될 수 있다.

[0150] 도 10 은 PU 를 포함하는 MER 외부에 있는 대체물들에 대한 대안 예를 예시하는 개념도이다. 도 10 의 예는 PU들 (232, 236) 을 포함하는 MER (230) 을 예시한다. PU (232) 는 공간 이웃하는 블록들 (234A-234E) 을 갖는다. PU (236) 는 공간 이웃하는 블록들 (238A-238E) 을 갖는다. 도 10 에 도시된 바와 같이, PU 의  $A_0$  또는  $A_1$  이웃하는 블록들이 PU 를 포함하는 MER 내부에 있을 때, 그들은 이용불가능한 것으로 간주될 수도 있다. 따라서, 수평 방향으로 MER 외부에 있는 정렬된 블록 (즉, 수직으로 정렬된 블록들) 은 대체물로서 사용될 수도 있다. 도 10 에서 PU (232) 에 대해 나타낸 바와 같이, PU (232) 의 (블록들  $A_0$  및  $A_1$  에 대응하는) 블록들 (234D 및 234E) 는, 그들이 PU (232) 를 또한 포함하는 MER (230) 내에 있기 때문에, 이용불가능

하다. 따라서, 비디오 인코더 (20) 및 비디오 디코더 (30)는 블록들 (234D, 234E)을 각각 블록들 (244D, 244E)로 대체할 수도 있다. 이 예에서, 블록들 (244D, 244E)은 블록들 (234D, 234E)과 동일한 수직 위치를 갖지만, MER (230) 외부에 있고 MER (230)에 이웃한다.

[0151] 이러한 방법으로, 블록들 (234D 및 234E)을 블록들 (244D 및 244E)로 대체하는 것은, 현재 블록에 이웃하는 블록 (예컨대, PU (232))이 현재 블록에 좌하측 이웃하는 블록 (블록들 (234D, 234E))에 대응할 때, 비디오 인코더 (20) 및 비디오 디코더 (30)는, MER에 좌측 이웃하는 블록이 현재 블록에 좌하측 이웃하는 블록과 동일한 수직 위치를 갖도록, 현재 블록에 좌하측 이웃하는 블록 (예컨대, 블록들 (234D, 234E) 중 하나)을 MER (230)에 좌측 이웃하는 블록 (예컨대, 블록들 (244D, 244E) 중 각각의 하나)으로 대체할 수도 있는 예를 나타낸다. 특히, 이 예에서, 블록 (244D)은 블록 (234D)과 동일한 수직 위치를 가지며, 블록 (244E)은 블록 (234E)과 동일한 수직 위치를 갖는다.

[0152] PU의  $B_0$  또는  $B_1$ 이웃하는 (우상측) 블록들이 PU를 포함하는 MER 내부에 있을 때, 그들은 이용불가능한 것으로 간주될 수도 있다. 따라서, 수직 방향으로 MER 외부에 있는 정렬된 블록 (즉, 수평으로 정렬된 블록들)은 대체물로서 사용될 수도 있다. 도 10에서 PU (236)에 대해 나타낸 바와 같이, 예를 들어, PU (236)에 이웃하는 블록들 (238B, 238C)은 그들이 PU (236)를 포함하는 MER (230) 내부에 있기 때문에, 이용불가능하다.

따라서, 비디오 인코더 (20) 및 비디오 디코더 (30)는 블록들 (238B, 238C)을 블록들 (246B, 246C)로 대체할 수도 있으며, 여기서, 블록들 (246B, 246C)은 블록들 (238B, 238C)과 동일한 수평 위치를 각각 갖지만, MER (230) 외부에 블록들 (246B, 246C)을 배치하는 수직 위치를 갖는다. 이것은 또한 PU (232), 특히, PU (232)에 이웃하는 블록들 (234B, 234C)의 경우도 마찬가지이며, 그 때문에 비디오 인코더 (20) 및 비디오 디코더 (30)가 각각 블록들 (244B, 244C)을 대체할 수도 있다.

[0153] 따라서, 블록들 (238B, 238C, 234B, 234C)을 블록들 (246B, 246C, 244B, 244C)로 각각 대체하는 것은, 현재 블록에 이웃하는 블록이 현재 블록에 우상측 이웃하는 블록에 대응하는 경우, 비디오 인코더 (20) 및 비디오 디코더 (30)는, MER에 상부 이웃하는 블록이 현재 블록에 우상측 이웃하는 블록과 동일한 수평 위치를 갖도록, 현재 블록에 우상측 이웃하는 블록을 MER에 상부 이웃하는 블록으로 대체할 수도 있는 예를 나타낸다.

[0154] PU의  $B_2$ 이웃하는 (좌상측) 블록이 MER 내부에 있을 때, 블록은 이용불가능한 것으로 간주될 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30)는 수평 또는 수직으로 바로 MER (230)의 외부에 정렬된 블록들 중 하나를 대체할 수도 있다. 이것은 도 10에서 PU (232)에 이웃하는 블록 (234A)의 경우도 마찬가지이며, 이 때문에 블록들 (240, 242)이 대체물로서 사용될 수도 있다. 블록 (234A)을 블록 (240)으로 대체하는 것은, 현재 블록에 이웃하는 블록이 현재 블록에 좌상측 이웃하는 블록에 대응하는 경우, 비디오 인코더 (20) 및 비디오 디코더 (30)가, MER에 좌측 이웃하는 블록이 현재 블록에 좌상측 이웃하는 블록과 동일한 수직 위치를 갖도록, 현재 블록에 좌상측 이웃하는 블록을 MER에 좌측 이웃하는 블록으로 대체할 수도 있는 예를 나타낸다. 블록 (234A)을 블록 (242)으로 대체하는 것은, 현재 블록에 이웃하는 블록이 현재 블록에 좌상측 이웃하는 블록에 대응하는 경우, MER에 상부 이웃하는 블록이 현재 블록에 좌상측 이웃하는 블록과 동일한 수평 위치를 갖도록, 현재 블록에 좌상측 이웃하는 블록을 MER에 상부 이웃하는 블록으로 대체하는 예를 나타낸다.

[0155] 이러한 방법으로, 도 8의 예에 대한 대안으로서, 비디오 인코더 (20) 및 비디오 디코더 (30)에 의해 수행되는 디스패리티 모션 생성 방법은 여러 이웃하는 블록들을 체크할 수도 있다. 이 경우, PU의 이웃하는 블록이 MER 내부에 있을 때, 비디오 인코더 (20) 및 비디오 디코더 (30)는 MER 내부에 있는 PU에 이웃하는 블록을, PU를 포함하는 MER의 유사한 로케이션 또는 방향에서 이웃하는 블록으로 대체할 수도 있다.

[0156] 도 11은 PU에 대한 디스패리티 벡터들에 대한 하나 이상의 후보 블록들이 PU를 포함하는 CU 내에 있는 예를 예시하는 개념도이다. 이 예는 CU 레벨 디스패리티 생성에의 PU 레벨 디스패리티 벡터 생성의 통합에 관한 것이다. 이 예에서, 주어진 PU(예컨대, 현재 PU (260))에 있어, 공간 이웃하는 블록들 (266, 274, 278)을 체크하는 대신, 단지 현재 PU (260)를 포함하는 CU (280)의 공간 이웃하는 블록들이 체크될 수도 있다.

[0157] 도 11에 나타낸 예에 있어, 체크되었어야 하는 현재 PU (260)에 이웃하는 블록들 (266, 274, 278)은, CU (280)의 이웃하는 블록이 아닐지도 모르거나, 또는 CU의  $B_0$ ,  $B_1$ ,  $B_2$ ,  $A_0$ ,  $A_1$ 로 케이션들 중 임의의 로케이션에 속하지 않을지도 모르기 때문에, 이제는 체크되지 않는다 (로케이션들  $B_0$ ,  $B_1$ ,  $B_2$ ,  $A_0$ ,  $A_1$ 은 도 9의 설명에 관련하여 정의된다). 디스패리티 벡터 생성은 위에서 설명한, Chen 등에 의한, "Motion Parameter Prediction and Coding for Multiview Video"에서 설명된 바와 같이, 다른 유형들의 디스패리티 벡터들, 예컨

대, 시간 디스패리티 벡터들을 체크할 수도 있다. 이러한 방법으로, 이 예는 PU들 및 CU들에 대한 공간 이웃하는 블록들을 통합할 수도 있어, 이러한 공간 이웃하는 블록들은 공간 디스패리티 벡터 체킹에 동일한 프로세스를 공유하여 동일한 결과들을 얻을 수 있다.

[0158] 따라서, CU (280)는 현재 PU (260)를 포함하는 포함 블록의 일 예를 나타낸다. 비디오 인코더 (20) 및 비디오 디코더 (30)는 현재 PU (260)의 모션 벡터를 코딩하기 위해, 블록 (266)을 블록 (264)으로, 블록 (274)을 블록 (272)으로, 그리고 블록 (278)을 블록 (276)으로 대체할 수도 있다. 이러한 방법으로, PU들 (260, 262)은 병렬로 코딩될 수도 있다.

[0159] 일부 예들에서, 비디오 코더는 MER 경우에 대해, 예컨대, MER 내부에 있는 PU에 대해 도 11과 관련하여 설명한 것들과 유사한 기법들을 수행할 수도 있으며, PU에 대한 공간 디스패리티 벡터 체킹의 프로세스들 및 결과들은 PU를 포함하는 MER에 대한 것과 유사할 수도 있다.

[0160] 일부 예들에서, MER 및 PU/CU 디스패리티 벡터 생성 기법들은 통합될 수도 있다. MER 및 CU의 최대 사이즈는 공간 디스패리티 벡터 체킹에 사용되는 디스패리티 생성 블록 유닛의 사이즈가 되도록 선택될 수도 있다.

상기 예들과 유사하게, MER 또는 CU 내부에 있는 각각의 PU 또는 CU는 공간 디스패리티 벡터 체킹을 위한 동일한 프로세스들 및 결과들을 공유할 수도 있다.

[0161] 예를 들어, 현재 CU 사이즈가  $16 \times 16$ 이고, 반면 MER의 사이즈가  $32 \times 32$ 이라고 가정한다. 이 경우, 공간 디스패리티 벡터들을 획득하는데 사용되는 디스패리티 생성 블록 유닛은  $32 \times 32$ 의 사이즈를 갖고 현재 PU를 커버하고, 이  $32 \times 32$  블록 유닛 내부에 있는 각각의 CU 및 PU는 공간 디스패리티 벡터 체킹을 위해 동일한 프로세스들 및 결과들을 공유한다.

[0162] 또 다른 예로서, 현재 CU 사이즈가  $16 \times 16$ 이고, 반면 MER의 사이즈가  $4 \times 4$ 이라고 가정한다. 이 경우, 공간 디스패리티 벡터들을 획득하는데 사용되는 디스패리티 생성 블록 유닛은  $16 \times 16$ 의 사이즈를 가지며 현재 PU를 커버하고, 이  $16 \times 16$  코딩 유닛 내부에 있는 각각의 PU는 공간 디스패리티 벡터 체킹을 위해 동일한 프로세스들 및 결과들을 공유한다.

[0163] 도 12는 디스패리티 생성 블록 유닛에 대한 디스패리티 벡터 생성의 일 예를 예시하는 개념도이다. 이 예에서, 디스패리티 생성 블록 유닛은 현재 CU 및 임의의 PU를 포함하는 MER 중 더 큰 것으로서 정의될 수도 있다. 디스패리티 생성 블록 유닛이  $2M \times 2M$  픽셀들의 사이즈를 갖고 좌상측 픽셀 좌표가  $(x0, y0)$ 이라고 가정한다. 디스패리티 생성 블록 유닛 (DGBU) (300)의 상부 이웃하는 ( $4 \times 4$ ) 블록들 (T0 312 및 T1 314에서) 및 좌측 이웃하는 블록들 (L0 316 및 L1 318에서)이 주어진 PU를 예측하는데 사용될 수도 있다.

[0164] 상부 이웃하는 블록들은 2개의 부분들, 즉, T0 312 및 T1 314로 분리될 수도 있으며, 여기서, T0 312는 DGBU (300) (M)의 폭의 제 1 절반 내의 블록들을 커버하고, T1 314는 DGBU (300) (M)의 폭의 제 2 절반을 포함하는 블록 및 DGBU (300)의 우상측 이웃하는 블록인 하나 더의 블록을 커버한다.

[0165] 좌측 이웃하는 블록들은 2개의 부분들, 즉, L0 316 및 L1 318로 분리될 수도 있으며, 여기서, L0 316는 DGBU (300) (M)의 높이의 제 1 절반 내의 블록들을 커버하고, T1 318는 DGBU (300) (M)의 높이의 제 2 절반을 포함하는 블록 및 DGBU (300)의 좌하측 이웃하는 것인 하나 더의 블록을 커버한다.

[0166] B 310로 표시된, 좌상측 블록은 PU 또는 CU의 로케이션에 따라서, 예컨대, 영역들 줌 0 (302), 줌 1 (304), 줌 2 (306), 줌 3 (308) 중 하나 내에, T0 312 또는 L0 314의 부분으로서 역시 존재할 수도 있다. 이의 대안으로, B 310는 T0 312에 항상 속할 수도 있다. 대안적으로 B는 L0 316에 항상 속할 수도 있다. T0 312, T1 314, L0 316, 및 L1 318은 따라서 이웃하는 영역들에 대해 표시될 수도 있다.

[0167] 체크되는 4개의 영역들의 순서는  $(dx, dy)$ 로서 표시되는, DGBU (300)의 좌상측 픽셀  $(x0, y0)$ 에 대한, 중앙 PU 또는 CU의 선택된 픽셀의 상대적인 로케이션에 의존할 수도 있다. 선택된 픽셀은 PU 또는 CU의 중간으로부터 유래할 수 있다. 먼저, DGBU (300)는 4개의 줌들 (줌 0 (302), 줌 1 (304), 줌 2 (306), 줌 3 (308))으로 분리될 수도 있으며, 각각은 다음과 같이 좌상측 픽셀 로케이션 및 우하측 픽셀 로케이션의 상대적인 좌표로 표현된다 (도 12에 도시됨): 줌 0 (302):  $(0, 0)$  및  $(M, M)$ ; 줌 1 (304):  $(M+1, 0)$  및  $(2M-1, M)$ ; 줌 2 (306):  $(0, M+1)$  및  $(M, 2M-1)$ ; 및 줌 3 (308):  $(M+1, M+1)$  및  $(2M-1, 2M-1)$ . 이의 대안으로, 선택된 픽셀은 PU 또는 CU의 좌상측 모서리일 수 있으며, 따라서 4개의 줌들이 재정의될 수도 있다.

[0168] T0 312, T1 314, L0 316, 및 L1 318의 순서는 다음과 같이 정의될 수도 있다:  $(dx, dy)$ 가 줌 0 (312)에 있고  $dx = dy$ 이면, 순서는 L0 316, T0 312 312, L1 318 318, T1 314 314일 수도 있다. 이의 대안으로, L0

316 316 및 T0 312 312 의 순서는 스위칭가능할 수도 있으며, L1 318 318 및 T1 314 314 의 순서는 스위칭가능 할 수도 있다. (dx, dy) 가 줌 0 (302) 에 있고  $dx < dy$  이면, 순서는 L0 316, T0 312, L1 318, T1 314 일 수도 있으며, 대안적으로는, 순서는 L0 316, L1 318, T0 312, T1 314 일 수 있다. (dx, dy) 가 줌 0 (302) 에 있고  $dx > dy$  이면, 순서는 T0 312, L0 316, T1 314, L1 318 일 수도 있으며, 대안적으로는, 순서는 T0 312, T1 314, L0 316, L1 318 일 수 있다. (dx, dy) 가 줌 1 (304) 에 있으면, 순서는 T1 314, T0 312, L0 316, L1 318 일 수도 있다. (dx, dy) 가 줌 2 (306) 에 있고  $dx = dy$  이면, 순서는 L1 318, T1 314, L0 316, T0 312 일 수도 있으며, 대안적으로는, L0 316 및 T0 312 의 순서는 스위칭가능할 수도 있으며 L1 318 및 T1 314 의 순서는 스위칭가능가능할 수도 있다. (dx, dy) 가 줌 3 (308) 에 있고  $dx < dy$  이면, 순서는 L1 318, T1 314, L0 316, T0 312 일 수도 있으며, 대안적으로는, 순서는 L1 318, T1 314, L0 316, T0 312 일 수 있다. (dx, dy) 가 줌 3 에 있고  $dx > dy$  이면, 순서는 T1 314, L1 318, T0 312, L0 316 일 수도 있으며, 대안적으로는, 순서는 T1 314, T0 312, L1 318, L0 316 일 수 있다.

[0169] 각각의 영역에 대해, 비디오 코더는 상부 영역 또는 좌측 영역에 속하는지 여부에 따라서, 모든 블록들 ( $4 \times 4$ ) 을 좌측으로부터 우측으로 또는 상부로부터 저부로 체크할 수도 있다. 일단 디스패리티 벡터가 이용가능하면, 디스패리티 벡터가 생성되고 전체 영역에 대한 스캐닝이 종료되며 생성 프로세스는 이 디스패리티 벡터를 반환할 수도 있다. 어떤 디스패리티 벡터도 이용불가능하면, 생성 프로세스는 공백 (empty) 을 반환할 수도 있다.

[0170] 디스패리티 벡터의 이용가능성 (availability) 은, 다음이 참이면, 즉, 블록에서 적어도 하나의 모션 벡터가 디스패리티 모션 벡터이고 디스패리티 모션 벡터가 참조 뷰 및 목표 뷰의 관점에서 목표 디스패리티 벡터와 매칭 하면, 이용가능한 것으로 정의될 수도 있다. 이의 대안으로, 디스패리티 벡터의 이용가능성은, 블록에서 적어도 하나의 모션 벡터가 디스패리티 모션 벡터이면 이용가능한 것으로 정의될 수도 있다.

[0171] 비디오 코더는 임의의 PU 에 대한 공간 디스패리티 벡터를 생성하도록 구성될 수도 있다. 먼저, 각각의 영역에 대한 디스패리티 벡터가 생성되어 일시적으로 저장될 수도 있으며, 따라서 이 예에서는 최고 4 개의 디스패리티 벡터들이 존재할 수 있다. 각각의 PU 또는 CU 에 대해, L0 316, L1 318, T0 312, T1 314 의 순서가 위에서 설명한 바와 같이 결정될 수도 있기 때문에, 각각의 영역은 주어진 순서로 체크될 수도 있다. 영역의 각각이 체크되어 (이미 공백으로서 생성되거나 또는 알려진) 그의 대응하는 디스패리티 벡터가 공백이 아닐 때, PU 또는 CU 의 디스패리티 벡터로 설정될 수도 있으며, 전체 프로세스가 중지될 수도 있다. 어떤 디스패리티 벡터도 발견되지 않으면, 전체 프로세스는 공백 (디스패리티 벡터를 이용불가능함) 을 반환할 수도 있다.

[0172] 비디오 코더는 또한 좌상측 블록 "B 310" 을 할당할 수도 있다. 블록 B 이 속하는 영역에 관련되는 위에서 언급한 하나의 대안은 영역들의 순서에 의존할 수 있다. 일 예에서, 비디오 코더는 어느 영역이 먼저 체크 되는지에 따라서 B 310 를 L0 316 또는 T0 312 에 할당한다.

[0173] 대안적인 예에서, L0 316 및 L1 318 은 L 에 결합될 수도 있으며, T0 312 및 T1 314 은 T 에 결합될 수도 있다. 한편, 적용되면, B 310 는 T 또는 L 로부터 제거될 수도 있다. (dx, dy) 는 여전히 선택된 픽셀의 상대적인 좌표일 수도 있다. 체크된 영역들의 순서는 B; ( $dx \leq dy$ ) 이면: L, 아니면 T; 그리고, ( $dx \leq dy$ ) 이면: T, 아니면 L 로서 정의될 수도 있다. 이의 대안으로, 체크된 영역들의 순서는 ( $dx \leq dy$ ) 이면: L, 아니면 T; ( $dx \leq dy$ ) 이면: T, 아니면 L; 및 B 일 수도 있다.

[0174] 또 다른 대안적인 예에서, L 또는 T 내부에서 스캐닝 순서는 꼭 좌측로부터 우측으로 또는 상부로부터 저부로는 아니다. (dx, dy) 가 여전히 선택된 픽셀의 상대적인 좌표이다.  $dx \leq M$  이면, T 에서 블록들의 스캐닝 순서는 좌측으로부터 우측으로 일 수도 있으며, 그렇지 않으면, 우측으로부터 좌측으로 일 수도 있다.  $dy \leq M$  이면, L 에서 블록들의 스캐닝 순서는 상부로부터 저부로 일 수도 있으며, 그렇지 않으면, 저부로부터 상부로 이다.

[0175] 또 다른 대안적인 예에서, L0 316, L1 318, T0 312 또는 T1 314 내에서의 스캐닝 순서는 꼭 좌측으로부터 우측으로 또는 상부로부터 저부로는 아니다. L0 316, L1 318, T0 312, 또는 T1 314 내에서의 순서는, 선택된 픽셀 ( $dx, dy$ ) 의 상대적인 좌표에 따라서 역전될지도 모른다. 예를 들어,  $M/2 < dx \leq M$  이면, T0 312 내에서의 블록들의 스캐닝 순서는 우측으로부터 좌측으로 일 수도 있다.

[0176] 또 다른 대안적인 예에서, L0 316, L1 318, T0 312 또는 T1 314 에서의 블록들 모두가 스캐닝되지는 않고, 단지 각각의 영역 (L0 316, L1 318, T0 312 및 T1 314) 의 중간의 블록이 스캐닝될 수도 있다. 모든 좌표들은

(x0,y0)에 상대적일 수도 있다. L0 316에 있어, 중간의 블록은 M/2의 상대적인 y-좌표에서 시작한다. L1 318에 있어, 중간의 블록은 M\*3/2의 상대적인 y-좌표에서 시작한다. T0 312에 있어, 중간의 블록은 M/2의 상대적인 x-좌표에서 시작한다. T1 314에 있어, 중간의 블록은 M\*3/2의 상대적인 x-좌표에서 시작한다. 이 경우, B가 추가적으로 추가될 수도 있다. 따라서, 전부 합해서, 단지 4개 또는 5개의 블록들이 디스패리티 벡터 생성 동안 공간 블록들에 대해 체크될 수도 있다.

[0177]

이의 대안으로, 위에서 설명한 바와 같이 단지 각각의 결합된 영역 (L 및 T)의 중간의 블록이 스캐닝될 수도 있다. L에 있어, 중간의 블록은 M의 상대적인 y-좌표에서 시작할 수도 있으며; T에 있어, 중간의 블록은 M의 상대적인 x-좌표에서 시작할 수도 있다. 게다가, B가 추가될 수도 있다. 따라서, 모두 합해서, 단지 2 또는 3개의 블록들이 체크될 수도 있다.

[0178]

이의 대안으로, 단지 각각의 영역 (L 및 T)의 중간의 블록이 스캐닝될 수도 있다. 여기서, L 또는 T는 B를 포함할 수도 있다.

[0179]

또 다른 대안적인 예에서, T0 312, T1 314, L0 316 및 L1 318의 순서는 디스패리티 생성 블록 유닛에서 모든 PU들 또는 CU들에 대해 고정될 수도 있다. 체킹 순서는 다음과 같을 수 있다: L1 318, L0 316, T1 314, T0 312, B. 이의 대안으로, 상기 순서의 임의의 대체가 적용될 수 있다. L1 318 또는 L0 316에서의 블록들에 대해, 체킹 순서는 상부로부터 저부로 또는 저부로부터 상부로 일 수 있다. T1 314 또는 T0 312에서의 블록들에 대해, 체킹 순서는 좌측으로부터 우측으로 또는 우측으로부터 좌측으로 일 수 있다.

[0180]

이의 대안으로, L 또는 T 내에서의 스캐닝 순서는 꼭 좌측으로부터 우측으로 또는 상부로부터 저부로가 아닐 수도 있다. 예를 들어, L은 우측으로부터 좌측으로 스캐닝될 수도 있으며, T는 저부로부터 우측으로 스캐닝될 수도 있다. 이의 대안으로, 스캐닝 순서의 임의의 다른 대체들이 적용될 수도 있다.

[0181]

도 13은 하나의 영역으로서 결합될 때 이웃하는 블록들의 스캐닝 순서에 대한 예를 예시하는 개념도이다. 도 13의 예는, 도 12의 L, T, 및 B 영역들이 영역 N(332)을 형성하기 위해 추가로 결합된다는 점을 제외하고는, 도 12의 상기 예들과 유사하다. 비디오 코더는 도 13에 나타낸 바와 같이, 저부로부터 상부로(도 12의 B 310 위치로) 그리고 그후 좌측으로부터 우측으로 N(332)을 스캐닝될 수도 있다.

[0182]

이의 대안으로, L, T, 및 B 영역들은 영역 N에 추가로 결합될 수도 있으며, N은 반대 순서로(예컨대, 우측에서 좌측으로, 그후 상하로) 스캐닝될 수도 있다.

[0183]

이의 대안으로, 순서는 우측으로부터 좌측으로(상부 블록들에서), 그리고, 상부로부터 저부로(좌측 블록들에서)로서 역전될 수 있으며, 원래 순서 또는 반대 순서인지의 여부는  $dx \leq dy$ 인지 여부에 의해 결정될 수 있다. 예를 들어, ( $dx \leq dy$ )이면, 비디오 코더는 원래 순서를 사용할 수도 있으며, 그렇지 않으면 비디오 코더는 반대 순서를 사용할 수도 있다.

[0184]

도 14는 시간 디스패리티 벡터 체킹을 위한 예시적인 방법을 예시하는 개념도이다. 도 14는 PU 레벨 디스패리티 벡터 생성이 공간 이웃하는 블록들에 대한 CU 레벨로 통합되는 예를 예시한다. 따라서, 이 예는 CU 레벨 디스패리티 생성에의 PU 레벨 디스패리티 벡터 생성의 통합에 관한 것이다.

[0185]

현재 화상(352)의 LCU(360)에서 CU(362)의 주어진 PU(364)에 있어, 여기서, PU(364)는 디스패리티 모션 벡터(366)를 이용하여 예측되며, 동일 장소에 배치된 CU(CCU)는 예컨대, 비디오 인코더(20) 또는 비디오 디코더(30)와 같은 비디오 코더에 의해 항상 체크될 수도 있다. 현재 화상(352)의 각각의 CU 또는 CU 내 각각의 PU에 있어, CCU(356)로서 표시된, CU의 동일 장소에 배치된 영역이 식별될 수도 있다. CLCU(354)로 표시된, CCU(356)를 포함하는 최대 CU, 및 (BR'(358)로 표시된) CCU(356)의 우하측  $4 \times 4$  블록이 도 14에 나타낸 바와 같이, 시간 이웃하는 블록들으로서 사용될 수도 있다.

[0186]

상기 3개의 영역들의 순서는 Chen 등에 의한 "Motion Parameter Prediction and Coding for Multiview Video"에서 언급한 것과 유사할 수 있다는 점에 유의한다. 예를 들어, 순서는 CCU(356), CLCU(354), 및 BR'(358), 또는 BR'(358), CCU(356), 및 CLCU(354)일 수도 있다.

[0187]

다수의 블록들을 포함하는 CCU 또는 CLCU를 체크할 때, 각각의 CCU, 또는 CLCU의  $16 \times 16$  블록들은 레스터-스캐닝 순서로 체크될 수도 있다. 일단 디스패리티 벡터가 이용가능하면, 체킹 프로세스가 종료될 수도 있다. 이의 대안으로, BR'는 CCU에서 중간의  $4 \times 4$  블록 중 하나로 대체될 수도 있다. 이의 대안으로, BR'는 CCU에서 임의의 다른  $4 \times 4$  블록로 대체될 수도 있다. 이의 대안으로, BR'는 이용불가능함 및 체크 안됨으로 설정될 수도 있다. 이의 대안으로, BR' 및 CCU 양쪽이 이용불가능함 및 체크 안됨으로 설정될 수도 있

다.

[0188] 디스패리티 벡터 생성이 Chen 등에 의한 "Motion Parameter Prediction and Coding for Multiview Video" 에서 설명된 바와 같이, 다른 유형들의 디스패리티 벡터들, 예컨대, 공간 디스패리티 벡터들을 체크할 수도 있다는 점에 유의한다. 도 14 의 예는 PU들 및 CU들에 대해 시간 이웃하는 블록들을 통합할 수도 있으며, 따라서 그들은 시간 디스패리티 벡터 체킹 및 동일한 결과들을 위해 동일한 프로세스를 공유할 수 있다. 후보 화상 선택 방법은 Chen 등에 의한 "Motion Parameter Prediction and Coding for Multiview Video" 에 설명되는 방법과 동일할 수 있다.

[0189] 대안으로서, CU 사이즈가  $16 \times 16$  보다 작으면, CCU 는 이 CU 를 포함하는  $16 \times 16$  블록으로 확장될 수도 있다. 나머지 프로세스들이 확장된 CCU 에도 동일하게 적용될 수도 있다.

[0190] 대안으로서, 프로세스는 Chen 등에 의한 "Motion Parameter Prediction and Coding for Multiview Video" 에서 설명된 프로세스와 유사할 수도 있으며, 그러나 PU 사이즈가  $16 \times 16$  보다 작으면, CPU 는 이 CU 를 포함하는  $16 \times 16$  블록으로 확장될 수도 있다. 나머지 프로세스들이 확장된 CCU 에도 동일하게 적용될 수도 있다.

[0191] 도 15 는 본 개시물의 기법에 따른, 디스패리티 모션 벡터를 인코딩하는 예시적인 방법을 예시하는 플로우차트이다. 비디오 인코더 (20) 와 관련하여 설명되었지만, 다른 디바이스들 또는 구성요소들이 도 15 의 방법을 수행하도록 구성될 수도 있는 것으로 이해되어야 한다. 더욱이, 오직 하나의 사례가 도시되지만, 도 15 의 방법의 다수의 사례들이 공통으로 포함하는 블록의 다수의 블록들 (예컨대, MER 또는 공통 CU 의 다수의 PU들) 에 대해 실질적으로 동시에, 예컨대, 병렬로 수행될 수도 있는 것으로 이해되어야 한다.

[0192] 이 예에서, 비디오 인코더 (20) 는 포함 블록에서의 현재 블록을 인터-뷰 예측을 이용하여 먼저 인코딩할 수도 있다 (380). 예를 들어, 여러 시험 인코딩 과정들을 수행한 후, 비디오 인코더 (20) 는 현재 블록이 인터-뷰 예측을 이용하여 인코딩되어야 한다고 궁극적으로 결정할 수도 있다. 따라서, 비디오 인코더 (20) 는 인터-뷰 참조 화상을 가리키는 디스패리티 모션 벡터를 이용하여 현재 블록을 인코딩할 수도 있다.

[0193] 비디오 인코더 (20) 는 그후 디스패리티 모션 벡터를 인코딩할 수도 있다. 이를 행하기 위해, 비디오 인코더 (20) 는 디스패리티 모션 벡터에 대한 후보 모션 벡터 예측자들을 포함하는 후보 리스트를 형성할 수도 있다. 특히, 본 개시물의 기법들에 따르면, 비디오 인코더 (20) 는 현재 블록의 특정의 이웃하는 블록이 현재 블록을 포함하는 포함 블록에 있는지의 여부를 결정할 수도 있다 (382). 이웃하는 블록이 포함 블록에 있지 않으면 (382 의 "아니오" 분기), 비디오 인코더 (20) 는 현재 블록의 이웃하는 블록을 후보 리스트에 추가할 수도 있다 (384). 한편, 이웃하는 블록이 현재 블록을 또한 포함하는 포함 블록에 있으면 (382 의 "예" 분기), 비디오 인코더 (20) 는 포함 블록의 이웃하는 블록을 후보 리스트에 추가할 수도 있다 (386). 즉, 비디오 인코더 (20) 는 현재 블록의 이웃하는 블록을 포함 블록의 이웃하는 블록으로 대체할 수도 있다. 비디오 인코더 (20) 는 예컨대, 도 8 내지 도 11 과 관련하여, 위에서 설명한 여러 예들, 또는 다른 대체 방식들에 따라서 이 대체를 수행할 수도 있다.

[0194] 비디오 인코더 (20) 는 그후 이전 이웃하는 블록이 후보 리스트에의 추가를 위해 고려되는 최종 이웃하는 블록이었는지의 여부를 결정할 수도 있다 (388). 이전 이웃하는 블록이 고려되는 최종 이웃하는 블록이 아니었으면 (388 의 "아니오" 분기), 비디오 인코더 (20) 는 다음 이웃하는 블록, 및 특히, 위에서 설명한 바와 같이, 현재 블록의 다음 이웃하는 블록을 포함 블록의 이웃하는 블록으로 대체할지의 여부를 고려할 수도 있다. 그러나, 이전 이웃하는 블록이 최종 이웃하는 블록이었으면 (388 의 "예" 분기), 비디오 인코더 (20) 는 일부 예들에서, 잠재적으로는, 후보 리스트를 가지치기 (pruning) 한 후, 후보 리스트로부터, 디스패리티 모션 벡터 예측자를 선택할 수도 있다 (390). 비디오 인코더 (20) 는 그후 디스패리티 모션 벡터 예측자를 이용하여, 예컨대, 병합 모드 또는 AMVP 모드를 이용하여, 디스패리티 모션 벡터를 인코딩할 수도 있다 (392).

[0195] 이러한 방법으로, 도 15 의 방법은, 디스패리티 모션 벡터를 이용하여 포함 블록 내 비디오 데이터의 현재 블록을 인코딩하는 단계, 현재 블록에 이웃하는 블록이 또한 포함 블록 내에 있다는 결정에 기초하여, 후보 리스트에서 이웃하는 블록을, 포함 블록 외부에 있고 포함 블록에 이웃하는 블록으로 대체하는 단계, 후보 리스트에서 복수의 블록들 중 하나로부터 디스패리티 모션 벡터 예측자를 선택하는 단계, 및 디스패리티 모션 벡터 예측자에 기초하여 디스패리티 모션 벡터를 인코딩하는 단계를 포함하는 방법의 일 예를 나타낸다.

[0196] 도 16 은 본 개시물의 기법에 따른, 디스패리티 모션 벡터를 디코딩하는 예시적인 방법을 예시하는 플로우차트이다. 비디오 디코더 (30) 와 관련하여 설명하였지만, 다른 디바이스들 또는 구성요소들은 도 16 의 방법을 수행하도록 구성될 수도 있는 것으로 이해되어야 한다. 더욱이, 오직 하나의 사례가 도시되지만, 도 16 의

방법의 다수의 사례들이 공통으로 포함하는 블록의 다수의 블록들 (예컨대, MER 또는 공통 CU 의 다수의 PU들)에 대해 실질적으로 동시에, 예컨대, 병렬로 수행될 수도 있는 것으로 이해되어야 한다.

[0197] 먼저, 비디오 디코더 (30) 는 포함 블록에서 현재 블록이 인터-뷰 예측을 이용하여 인코딩되는 것을 결정할 수도 있다 (400). 따라서, 비디오 디코더 (30) 는 디스패리티 모션 벡터가 디코딩되어야 한다고 결정할 수도 있다. 따라서, 비디오 디코더 (30) 는 위에서 도 15 와 관련하여 설명한 방법과 실질적으로 동일한 방법으로 구성할 수도 있다. 예를 들어, 비디오 디코더 (30) 는 디스패리티 모션 벡터에 대한 후보 모션 벡터 예측자들을 포함하는 후보 리스트를 형성할 수도 있다. 특히, 본 개시물의 기법들에 따르면, 비디오 디코더 (30) 는 현재 블록의 특정의 이웃하는 블록이 현재 블록을 포함하는 포함 블록에 있는지의 여부를 결정할 수도 있다 (402). 이웃하는 블록이 포함 블록에 있지 않으면 (402 의 "아니오" 분기), 비디오 디코더 (30) 는 현재 블록의 이웃하는 블록을 후보 리스트에 추가할 수도 있다 (404). 한편, 이웃하는 블록이 현재 블록을 또한 포함하는 포함 블록에 있으면 (402 의 "예" 분기), 비디오 디코더 (30) 는 포함 블록의 이웃하는 블록을 후보 리스트에 추가할 수도 있다 (406). 즉, 비디오 디코더 (30) 는 현재 블록의 이웃하는 블록을 포함 블록의 이웃하는 블록으로 대체할 수도 있다. 비디오 디코더 (30) 는 예컨대, 도 8 - 도 11 과 관련하여 위에서 설명한 여러 예들, 또는 다른 대체 방식들에 따라서 이 대체를 수행할 수도 있다.

[0198] 비디오 디코더 (30) 는 그후 이전 이웃하는 블록이 후보 리스트에의 추가를 위해 고려되는 최종 이웃하는 블록이었는지의 여부를 결정할 수도 있다 (408). 이전 이웃하는 블록이 고려되는 최종 이웃하는 블록이 아니었으면 (388 의 "아니오" 분기), 비디오 디코더 (30) 는 다음 이웃하는 블록, 및 특히, 위에서 설명한 바와 같이, 현재 블록의 다음 이웃하는 블록을 포함 블록의 이웃하는 블록으로 대체할지의 여부를 고려할 수도 있다. 그러나, 이전 이웃하는 블록이 최종 이웃하는 블록이었으면 (408 의 "예" 분기), 비디오 디코더 (30) 는 일부 예들에서, 잠재적으로는, 후보 리스트를 가지치기 (pruning) 한 후, 후보 리스트로부터, 디스패리티 모션 벡터 예측자를 선택할 수도 있다 (410). 예를 들어, 비디오 디코더 (30) 는 후보 리스트로의 인덱스를 나타내는, 병합 인덱스 또는 AMVP 인덱스를 디코딩하고, 후보 리스트로의 인덱스에 대응하는 이웃하는 블록의 모션 정보로부터 디스패리티 모션 벡터 예측자를 선택할 수도 있다.

[0199] 비디오 디코더 (30) 는 그후 디스패리티 모션 벡터 예측자를 이용하여, 예컨대, 병합 모드 또는 AMVP 모드를 이용하여, 디스패리티 모션 벡터를 디코딩할 수도 있다 (412). 예를 들어, 병합 모드에서, 비디오 디코더 (30) 는 현재 블록에 대한 디스패리티 모션 벡터가 단계 (410) 에서 선택될 때의 디스패리티 모션 벡터 예측자와 동일하다고 결정할 수도 있다. 이의 대안으로, AMVP 모드에서, 비디오 디코더 (30) 는 모션 벡터 차이 값을 정의하는 데이터를 추가로 디코딩할 수도 있으며, 비디오 디코더 (30) 는 모션 벡터 차이 값을 디스패리티 모션 벡터 예측자에 가산함으로써 디스패리티 모션 벡터를 재생할 수도 있다.

[0200] 어쨌든, 비디오 디코더 (30) 는 그후 디코딩된 디스패리티 모션 벡터 예측자를 이용하여 현재 블록을 디코딩할 수도 있다 (414). 즉, 비디오 디코더 (30) 는 디스패리티 모션 벡터에 의해 식별되는 인터-뷰 참조 블록으로부터 예측된 블록을 생성하고, 양자화된 변환 계수들을 정의하는 십екс 앤리먼트들을 디코딩하고, 이를 계수들을 역양자화하고 역변환하여 잔차 블록을 재생하고, 그리고 (예측된 블록의 픽셀 값들을 잔차 블록의 픽셀 값들에 가산함으로써) 예측된 블록을 잔차 블록과 합산하여, 궁극적으로 현재 블록을 얻을 수도 있다.

[0201] 이러한 방법으로, 도 16 의 방법은, 현재 블록에 이웃하는 블록이 또한 포함 블록 내에 있다는 결정에 기초하여, 포함 블록 내에 있는 비디오 데이터의 현재 블록이 디스패리티 모션 벡터를 이용하여 인코딩되는 것을 결정하는 단계, 후보 리스트에서 이웃하는 블록을, 포함 블록 외부에 있고 포함 블록에 이웃하는 블록으로 대체하는 단계, 후보 리스트에서 복수의 블록들 중 하나로부터 디스패리티 모션 벡터 예측자를 선택하는 단계, 및 디스패리티 모션 벡터 예측자에 기초하여 디스패리티 모션 벡터를 디코딩하는 단계를 포함하는 방법의 일 예를 나타낸다.

[0202] 그 예에 따라서, 본원에서 설명되는 기법들 중 임의의 기법의 어떤 행위들 또는 이벤트들이 상이한 시퀀스로 수행될 수 있으며, 추가되거나, 병합되거나, 또는 모두 제외시킬 수도 있는 (예컨대, 모든 설명되는 행위들 또는 이벤트들이 기법들의 실시에 필수적인 것은 아닌) 것으로 인식되어야 한다. 더욱이, 어떤 예들에서, 행위들 또는 이벤트들은 순차적으로 보다는, 동시에, 예컨대, 멀티-쓰레드된 프로세싱, 인터럽트 프로세싱, 또는 다수의 프로세서들을 통해서 수행될 수도 있다.

[0203] 하나 이상의 예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현되는 경우, 그 기능들은 하나 이상의 명령들 또는 코드로서, 컴퓨터 판독 가능 매체 상에 저장되거나 또는 컴퓨터 판독 가능 매체를 통해서 송신될 수도 있으며, 하드웨어-기반의 프로세싱 유

닛에 의해 실행될 수도 있다. 컴퓨터 관독가능 매체는 컴퓨터 관독가능 저장 매체들을 포함할 수도 있으며, 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) 비일시성 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 사용 매체들일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터 관독가능 매체를 포함할 수도 있다.

[0204] 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 이용하여 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, 캐리어 파들, 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 포함하며, 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.

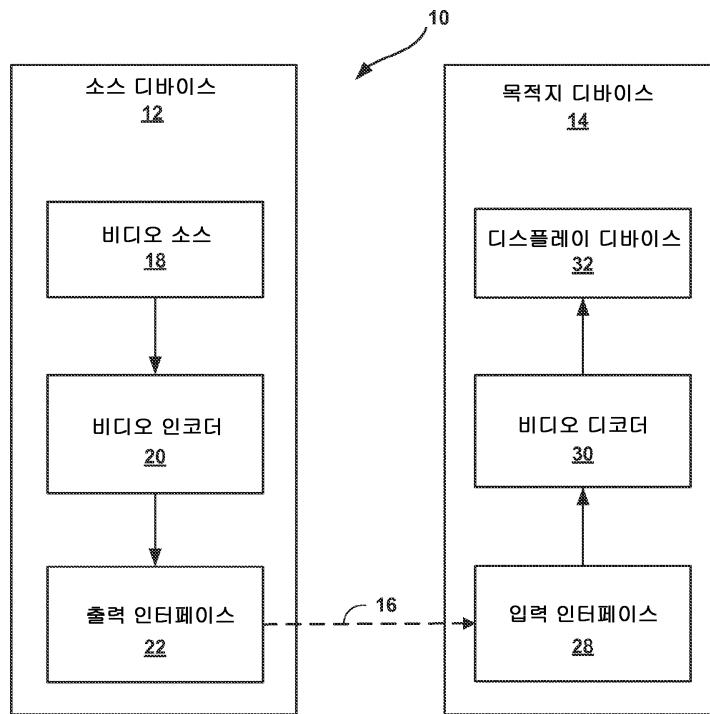
[0205] 명령들은 하나 이상의 디지털 신호 프로세서들 (DSP들), 범용 마이크로프로세서들, 주문형 집적회로들 (ASICs), 필드 프로그래밍가능 로직 어레이들 (FPGAs), 또는 다른 등가의 통합 또는 이산 로직 회로와 같은, 하나 이상의 프로세서들에 의해 실행될 수도 있다. 따라서, 용어 "프로세서" 는, 본원에서 사용될 때 전술한 구조 중 임의의 구조 또는 본원에서 설명하는 기법들의 구현에 적합한 임의의 다른 구조를 지칭할 수도 있다. 게다가, 일부 양태들에서, 본원에서 설명하는 기능 전용 하드웨어 및/또는 인코딩 및 디코딩을 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.

[0206] 본 개시물의 기법들은 무선 핸드셋, 집적 회로 (IC) 또는 IC들의 세트 (예컨대, 칩 셋) 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.

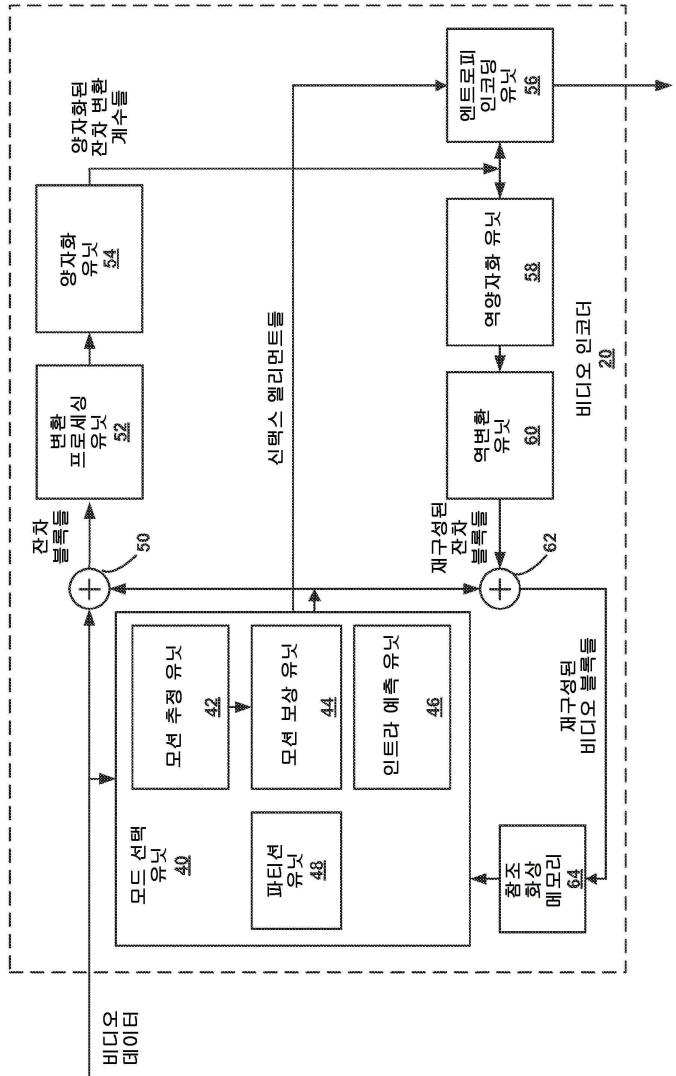
[0207] 여러 예들이 설명되었다. 이들 및 다른 예들은 다음 청구항들의 범위 이내이다.

도면

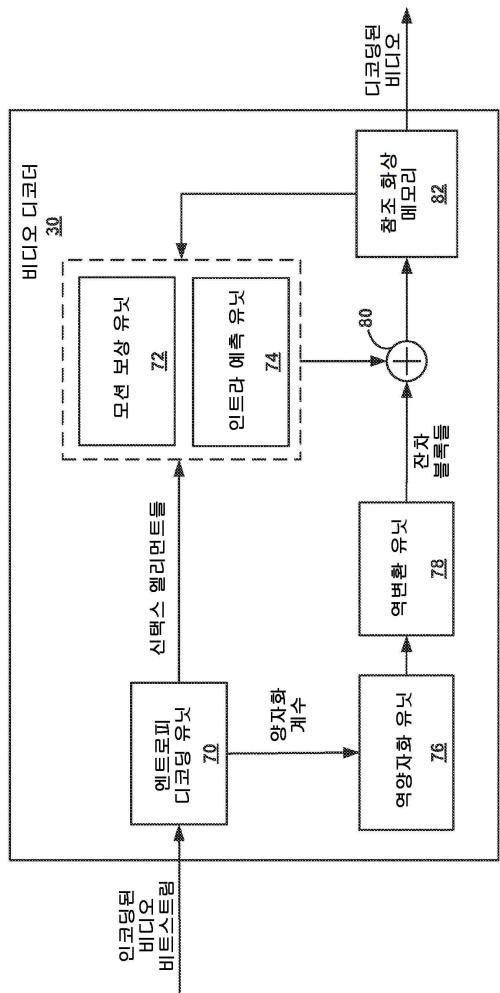
도면1



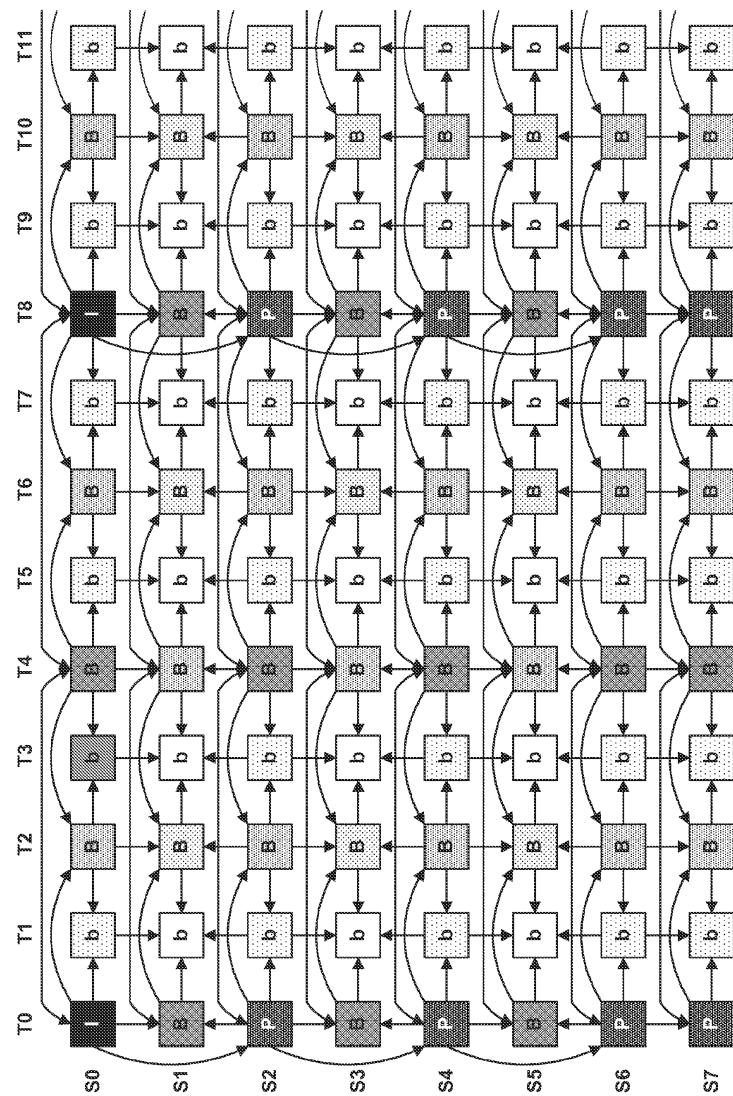
## 도면2



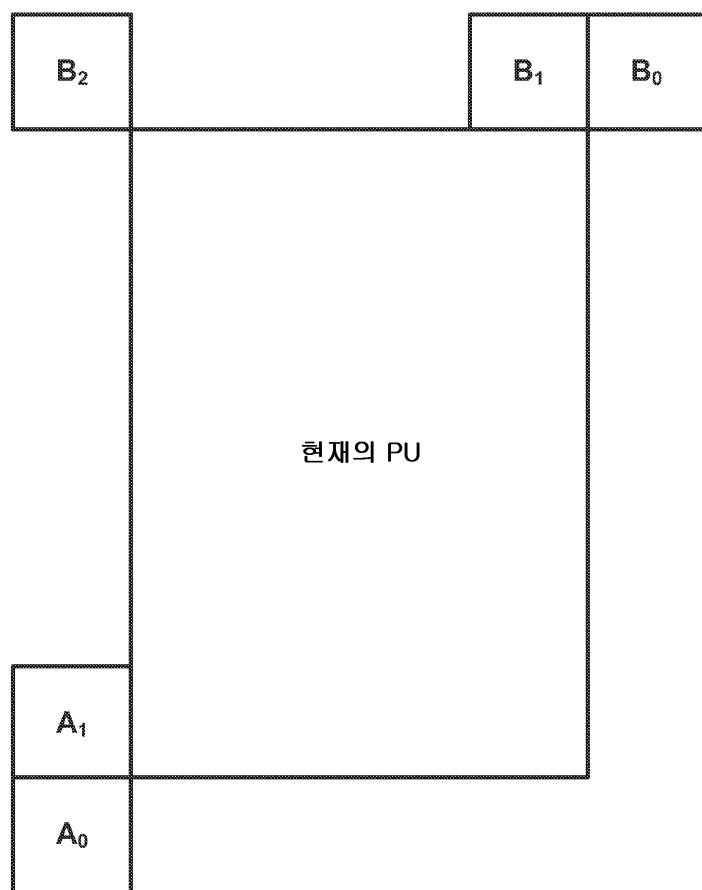
도면3



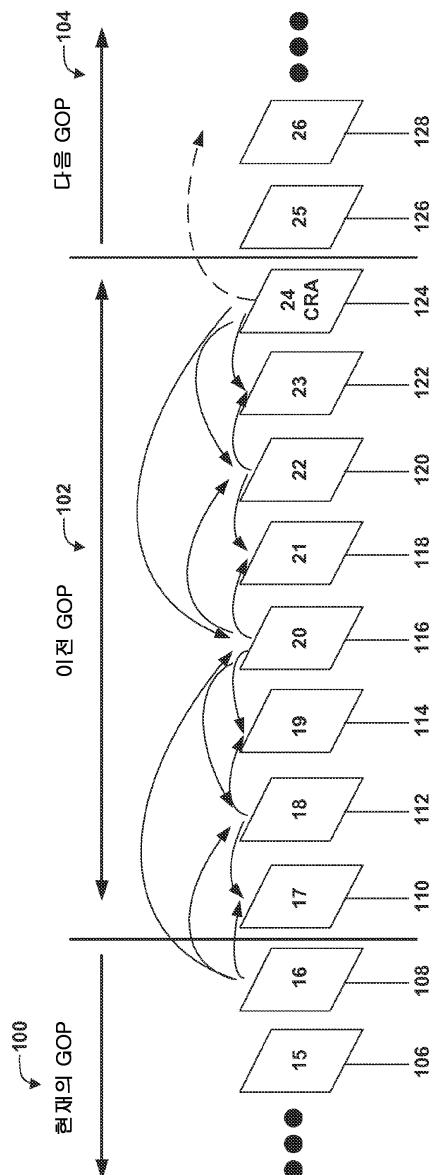
도면4



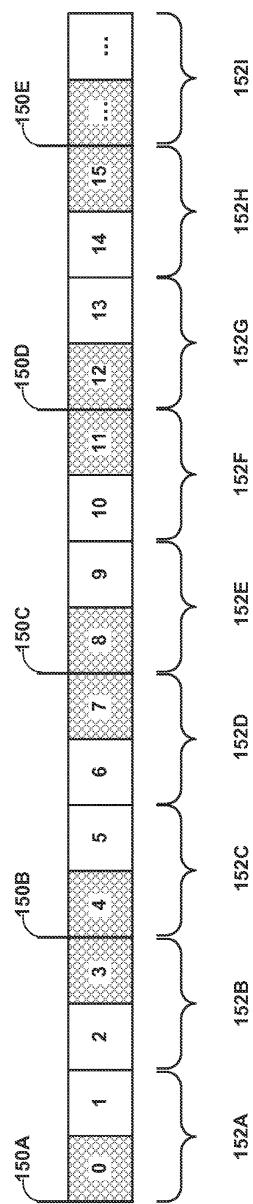
도면5



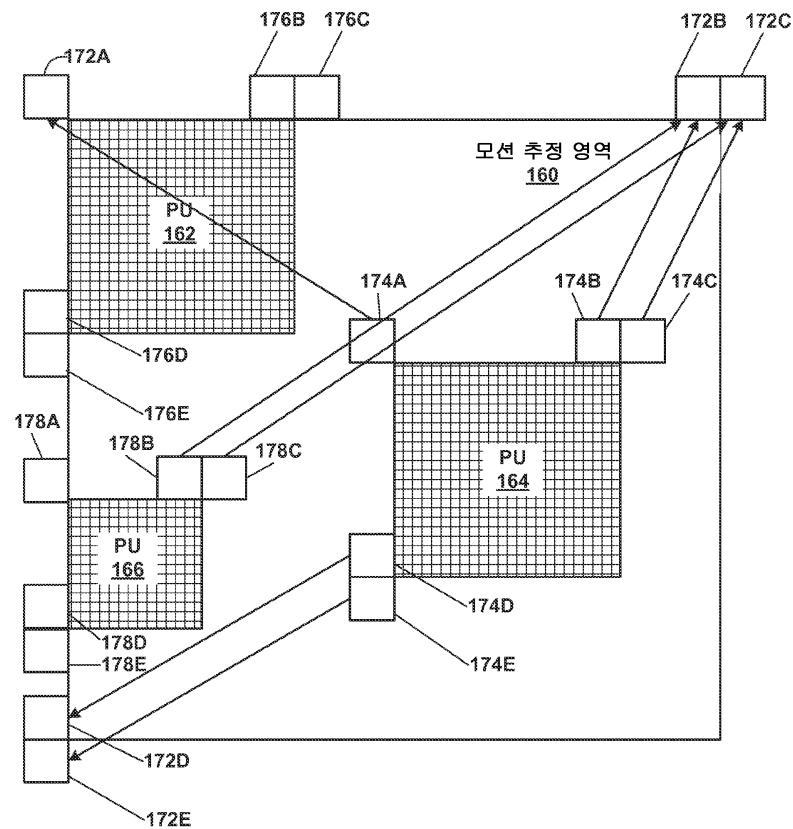
도면6



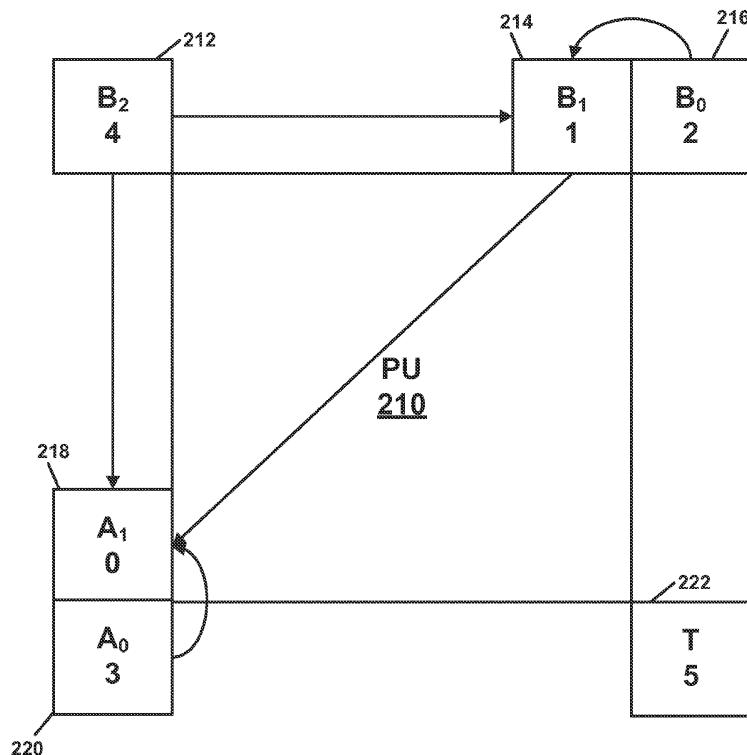
## 도면7



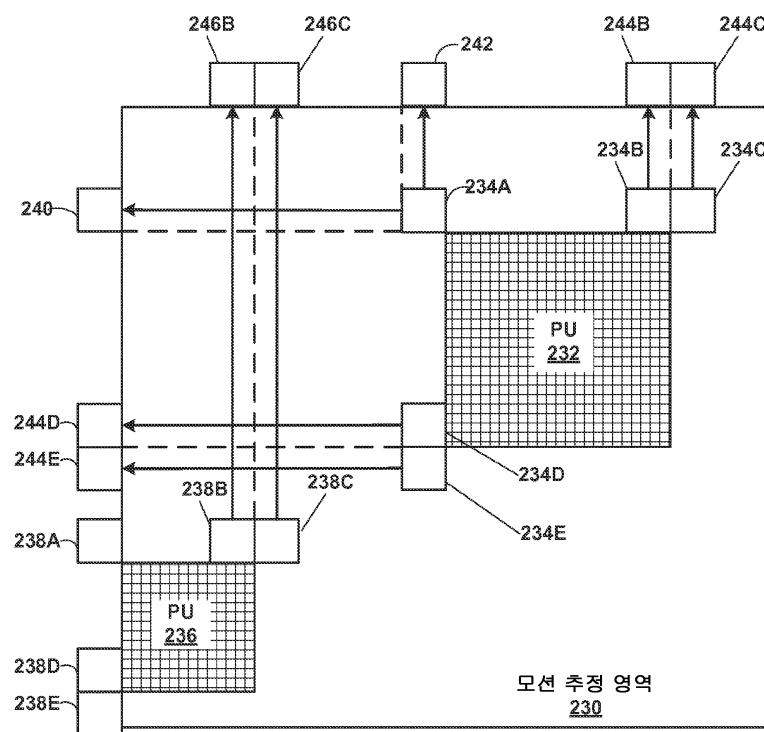
도면8



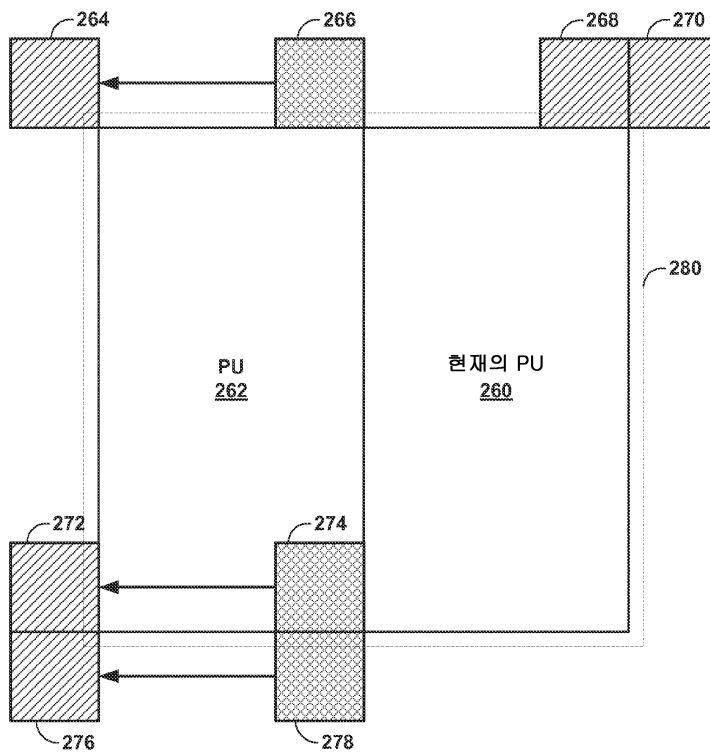
도면9



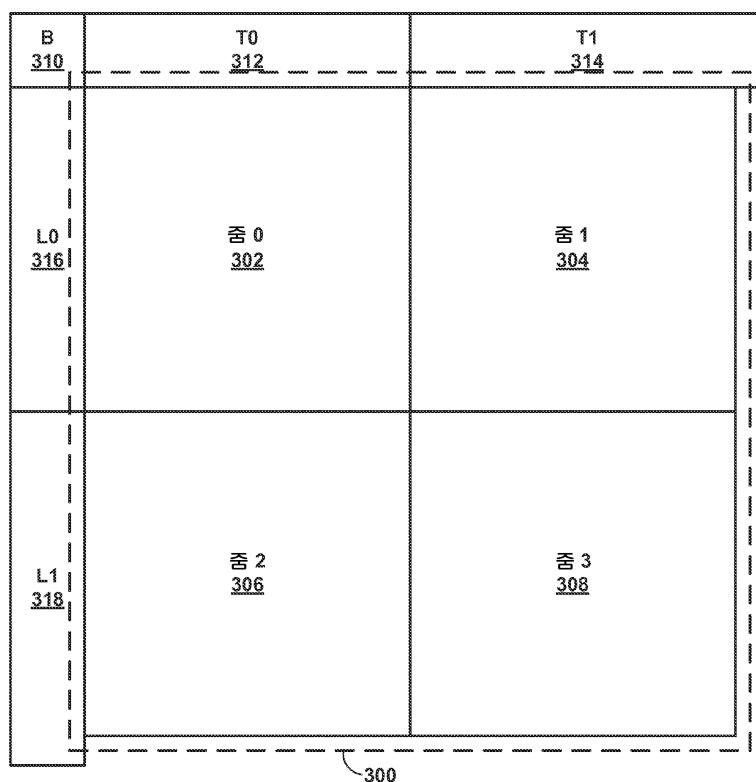
도면10



도면11



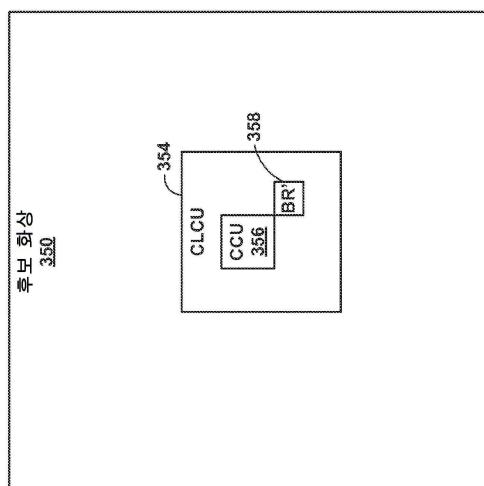
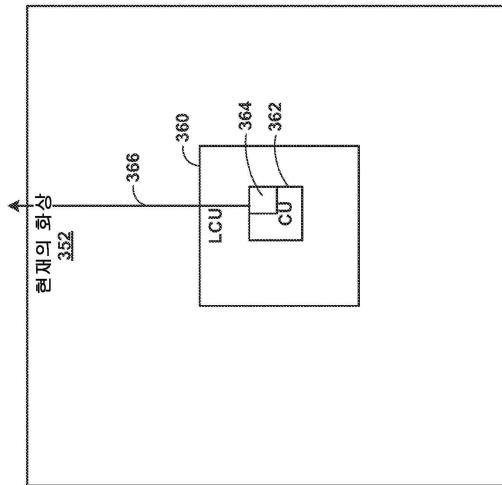
도면12



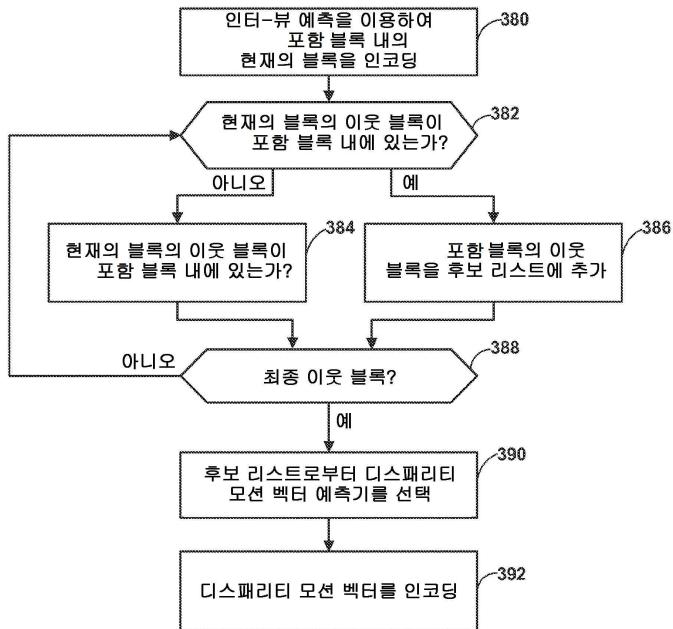
도면13



도면14



## 도면15



## 도면16

