



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2016-0016841  
(43) 공개일자 2016년02월15일

- |  |  |
|--|--|
| <p>(51) 국제특허분류(Int. Cl.)<br/>H04N 19/597 (2014.01) H04N 13/02 (2006.01)</p> <p>(52) CPC특허분류(Coo. Cl.)<br/>H04N 19/597 (2015.01)<br/>H04N 13/0217 (2013.01)</p> <p>(21) 출원번호 10-2015-7034639</p> <p>(22) 출원일자(국제) 2014년05월30일<br/>심사청구일자 없음</p> <p>(85) 번역문제출일자 2015년12월04일</p> <p>(86) 국제출원번호 PCT/US2014/040272</p> <p>(87) 국제공개번호 WO 2014/194218<br/>국제공개일자 2014년12월04일</p> <p>(30) 우선권주장<br/>61/829,919 2013년05월31일 미국(US)<br/>14/290,603 2014년05월29일 미국(US)</p> | <p>(71) 출원인<br/>켈컴 인코퍼레이티드<br/>미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775</p> <p>(72) 발명자<br/>장 리<br/>미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775</p> <p>천 잉<br/>미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775</p> <p>카르체비츠 마르타<br/>미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775</p> <p>(74) 대리인<br/>특허법인코리아나</p> |
|--|--|

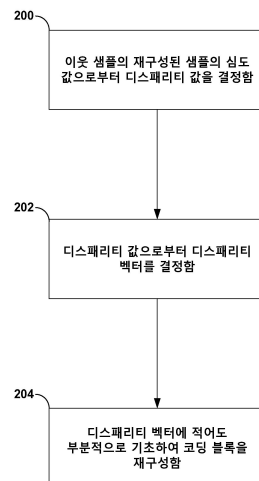
전체 청구항 수 : 총 30 항

(54) 발명의 명칭 심도 블록들의 디스패리티에 기초한 어드밴스드 심도 인터 코딩

### (57) 요약

하나의 예에서, 본 개시물은, 비디오 데이터의 픽처의 슬라이스의 각각의 코딩 유닛 (CU) 의 각각의 예측 유닛 (PU) 에 대해, 적어도 하나의 이웃 샘플의 적어도 하나의 재구성된 심도 샘플의 적어도 하나의 심도 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 값을 결정하는 단계를 포함하는 기법들에 관련된다. 이 기법들은 적어도 하나의 디스패리티 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 벡터를 결정하는 단계를 더 포함하고, 여기서 적어도 하나의 디스패리티 벡터는 각각의 PU 에 대한 각각의 CU 에 대한 것이다. 이 기법들은 적어도 하나의 디스패리티 벡터에 적어도 부분적으로 기초하여 각각의 PU 에 대한 각각의 CU 에 대한 코딩 블록을 재구성하는 단계를 더 포함한다.

대표도 - 도10





(52) CPC특허분류(Coo. Cl.)  
**H04N 13/0271** (2013.01)

---



## 특허청구의 범위

### 청구항 1

심도 뷰 데이터 (depth view data) 를 디코딩하는 방법으로서,

상기 심도 뷰 데이터의 픽처의 슬라이스의 각각의 코딩 유닛 (coding unit; CU) 의 각각의 예측 유닛 (prediction unit; PU) 에 대해:

적어도 하나의 이웃 샘플의 적어도 하나의 재구성된 심도 샘플의 적어도 하나의 심도 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 값을 결정하는 단계;

상기 적어도 하나의 디스패리티 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 벡터를 결정하는 단계로서, 상기 적어도 하나의 디스패리티 벡터는 상기 각각의 CU 에 대한 것인, 상기 적어도 하나의 디스패리티 벡터를 결정하는 단계; 및

상기 적어도 하나의 디스패리티 벡터에 적어도 부분적으로 기초하여 상기 각각의 CU 에 대한 코딩 블록을 재구성하는 단계

를 포함하는, 심도 뷰 데이터를 디코딩하는 방법.

### 청구항 2

제 1 항에 있어서,

상기 적어도 하나의 디스패리티 벡터를 결정하는 단계는, 2개 이상의 이웃 심도 샘플들에 적어도 부분적으로 기초하여 하나의 디스패리티 벡터를 결정하는 단계를 포함하고,

상기 방법은, 상기 2개 이상의 이웃 심도 샘플들 중 하나의 이웃 심도 샘플을 선택하는 단계를 더 포함하고,

상기 2개 이상의 이웃 심도 샘플들 중 하나의 이웃 심도 샘플을 선택하는 단계는,

선택된 상기 이웃 심도 샘플을 식별하는 인덱스 값을 결정하는 것으로서, 상기 인덱스 값은 블록 레벨에서 시그널링되는, 상기 선택된 상기 이웃 심도 샘플을 식별하는 인덱스 값을 결정하는 것, 및

상기 2개 이상의 이웃 심도 샘플들 중 하나를 선택하기 위해 상기 2개 이상의 이웃 심도 샘플들의 값들에 수학 함수를 적용하는 것

중 적어도 하나에 기초하는, 심도 뷰 데이터를 디코딩하는 방법.

### 청구항 3

제 1 항에 있어서,

상기 적어도 하나의 디스패리티 벡터를 결정하는 단계는, 2개 이상의 이웃 심도 샘플들에 적어도 부분적으로 기초하여 2개 이상의 디스패리티 벡터들을 결정하는 단계를 포함하고,

상기 방법은,

상기 2개 이상의 이웃 심도 샘플들에 기초하여 2개 이상의 디스패리티 값들을 상기 2개 이상의 디스패리티 벡터들로 컨버팅하는 단계를 더 포함하는, 심도 뷰 데이터를 디코딩하는 방법.

### 청구항 4

제 1 항에 있어서,

상기 적어도 하나의 디스패리티 벡터를 적어도 하나의 디스패리티 모션 벡터로 컨버팅하는 단계를 더 포함하는, 심도 뷰 데이터를 디코딩하는 방법.

### 청구항 5

제 1 항에 있어서,



상기 적어도 하나의 디스패리티 값을 결정하는 단계는, 상기 각각의 PU 의 2개 이상의 이웃 샘플들에 적어도 부분적으로 기초하여 2개 이상의 디스패리티 값들을 결정하는 단계를 더 포함하고,

상기 적어도 하나의 디스패리티 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 벡터를 결정하는 단계는, 상기 2개 이상의 디스패리티 값들에 적어도 부분적으로 기초하여 2개 이상의 디스패리티 벡터들을 결정하는 단계를 더 포함하고,

상기 PU 의 상부 좌측 샘플은 좌표들 (x, y) 을 가지며,

2개 이상의 상기 재구성된 심도 샘플들은,

상기 PU 의 상부 좌측 이웃 샘플,

상기 PU 의 상부 이웃 로우 (row),

상기 PU 의 좌측 이웃 컬럼 (column),

상기 각각의 PU 의 특정 이웃 블록에 로케이팅된 임의의 샘플,

좌표들 (x-1, y+1) 을 갖는 샘플,

좌표들 (x+1, y-1) 을 갖는 샘플, 그리고,

PU 사이즈가  $W \times H$  로 표기될 때, 좌표들 (x-1, y+H-1) 을 갖는 샘플 및 좌표들 (x+W-1, y-1) 을 갖는 샘플

중 적어도 2개 이상으로부터의 것인, 심도 뷰 데이터를 디코딩하는 방법.

#### 청구항 6

제 5 항에 있어서,

상기 적어도 하나의 디스패리티 값을 결정하는 단계는, 상기 각각의 CU 의 2개 이상의 이웃 샘플들에 적어도 부분적으로 기초하여 2개 이상의 디스패리티 값들을 결정하는 단계를 더 포함하고,

상기 각각의 CU 의 상부 좌측 샘플은 좌표들 (x, y) 을 가지며,

2개 이상의 상기 재구성된 심도 샘플들은,

상기 각각의 CU 의 상부 좌측 이웃 샘플,

상기 각각의 CU 의 상부 이웃 로우,

상기 각각의 CU 의 좌측 이웃 컬럼,

상기 각각의 CU 의 특정 이웃 블록에 로케이팅된 임의의 샘플,

좌표들 (x-1, y+1) 을 갖는 샘플,

좌표들 (x+1, y-1) 을 갖는 샘플, 그리고,

각각의 CU 사이즈가  $2N \times 2N$  으로 표기될 때, 좌표들 (x-1, y+2N-1) 을 갖는 샘플 및 좌표들 (x+2N-1, y-1) 을 갖는 샘플

중 적어도 2개 이상으로부터의 것인, 심도 뷰 데이터를 디코딩하는 방법.

#### 청구항 7

제 5 항에 있어서,

상기 2개 이상의 이웃 샘플들은 상기 PU 의 최대 코딩 유닛 (largest coding unit; LCU) 의 이웃 샘플들인, 심도 뷰 데이터를 디코딩하는 방법.

#### 청구항 8

제 1 항에 있어서,

상기 적어도 하나의 이웃 샘플은 상기 각각의 CU 의 이웃인, 심도 뷰 데이터를 디코딩하는 방법.



#### 청구항 9

제 1 항에 있어서,  
 상기 적어도 하나의 재구성된 심도 샘플은,  
 상기 PU 의 상부 좌측 이웃 샘플,  
 상기 PU 의 상부 이웃 로우,  
 상기 PU 의 좌측 이웃 컬럼, 및  
 상기 PU 의 특정 이웃 블록에 로케이팅된 임의의 샘플  
 중 적어도 하나로부터의 것인, 심도 뷰 데이터를 디코딩하는 방법.

#### 청구항 10

제 1 항에 있어서,  
 상기 적어도 하나의 디스패리티 벡터를 결정하는 단계는, 수행된 이웃 기반 디스패리티 벡터 유도 (neighbor-based disparity vector derivation; NBDV) 프로세스가 임의의 이용가능한 디스패리티 벡터를 식별하는 것을 실패했음에 응답하는, 심도 뷰 데이터를 디코딩하는 방법.

#### 청구항 11

제 1 항에 있어서,  
 상기 적어도 하나의 디스패리티 벡터에 기초하여 참조 블록을 식별하는 단계;  
 식별된 상기 참조 블록의 참조 인덱스 및 모션 벡터에 적어도 부분적으로 기초하여 상기 PU 의 현재 모션을 예측하는 단계; 및  
 상기 참조 인덱스 및 모션 벡터에 적어도 부분적으로 기초하여 결정된 후보를 병합 또는 어드밴스드 모션 벡터 예측 (advanced motion vector prediction; AMVP) 후보 리스트 내에 삽입하는 단계  
 를 더 포함하는, 심도 뷰 데이터를 디코딩하는 방법.

#### 청구항 12

제 1 항에 있어서,  
 상기 적어도 하나의 디스패리티 벡터를 결정하는 단계는, 카메라 파라미터들에 적어도 부분적으로 기초하여 상기 디스패리티 벡터를 결정하는 단계를 포함하는, 심도 뷰 데이터를 디코딩하는 방법.

#### 청구항 13

제 12 항에 있어서,  
 하나 이상의 카메라 파라미터들은 2개의 뷰들의 수평 변위를 포함하는, 심도 뷰 데이터를 디코딩하는 방법.

#### 청구항 14

비디오 데이터를 디코딩하기 위한 디바이스로서,  
 심도 뷰 데이터의 픽처와 연관된 데이터를 저장하도록 구성된 메모리; 및  
 상기 메모리와 통신하는 하나 이상의 프로세서들  
 을 포함하고,  
 상기 하나 이상의 프로세서들은,  
 상기 심도 뷰 데이터의 픽처의 슬라이스의 각각의 코딩 유닛 (CU) 의 각각의 예측 유닛 (PU) 에 대해:  
 적어도 하나의 이웃 샘플의 적어도 하나의 재구성된 심도 샘플의 적어도 하나의 심도 값에 적어도 부분적으로



기초하여 적어도 하나의 디스패리티 값을 결정하고;

상기 적어도 하나의 디스패리티 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 벡터를 결정하는 것으로서, 상기 적어도 하나의 디스패리티 벡터는 상기 각각의 CU 에 대한 것인, 상기 적어도 하나의 디스패리티 벡터를 결정하는 것을 행하며;

상기 적어도 하나의 디스패리티 벡터에 적어도 부분적으로 기초하여 상기 각각의 CU 에 대한 코딩 블록을 재구성하도록

구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

#### 청구항 15

제 14 항에 있어서,

상기 적어도 하나의 디스패리티 벡터를 결정하는 것은, 상기 디바이스가 2개 이상의 이웃 심도 샘플들에 적어도 부분적으로 기초하여 하나의 디스패리티 벡터를 결정하도록 구성되는 것을 포함하고,

상기 디바이스는 또한, 상기 2개 이상의 이웃 심도 샘플들 중 하나의 이웃 심도 샘플을 선택하도록 구성되고,

상기 2개 이상의 이웃 심도 샘플들 중 하나의 이웃 심도 샘플을 선택하는 것은,

선택된 상기 이웃 심도 샘플을 식별하는 인덱스 값을 결정하는 것으로서, 상기 인덱스 값은 블록 레벨에서 시그널링되는, 상기 선택된 상기 이웃 심도 샘플을 식별하는 인덱스 값을 결정하는 것, 및

상기 2개 이상의 이웃 심도 샘플들 중 하나를 선택하기 위해 상기 2개 이상의 이웃 심도 샘플들의 값들에 수학 함수를 적용하는 것

중 적어도 하나에 기초하는, 비디오 데이터를 디코딩하기 위한 디바이스.

#### 청구항 16

제 14 항에 있어서,

상기 디바이스는 또한,

상기 적어도 하나의 디스패리티 벡터를 적어도 하나의 디스패리티 모션 벡터로 컨버팅하도록 구성되는, 비디오 데이터를 디코딩하기 위한 디바이스.

#### 청구항 17

명령들을 저장한 컴퓨터 판독가능 저장 매체로서,

상기 명령들은, 실행될 때,

심도 뷰 데이터의 픽처의 슬라이스의 각각의 코딩 유닛 (CU) 의 각각의 예측 유닛 (PU) 에 대해:

적어도 하나의 이웃 샘플의 적어도 하나의 재구성된 심도 샘플의 적어도 하나의 심도 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 값을 결정하고;

상기 적어도 하나의 디스패리티 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 벡터를 결정하는 것으로서, 상기 적어도 하나의 디스패리티 벡터는 상기 각각의 CU 에 대한 것인, 상기 적어도 하나의 디스패리티 벡터를 결정하는 것을 행하며;

상기 적어도 하나의 디스패리티 벡터에 적어도 부분적으로 기초하여 상기 각각의 CU 에 대한 코딩 블록을 재구성하도록

비디오 디코딩 디바이스를 구성하는, 컴퓨터 판독가능 저장 매체.

#### 청구항 18

제 17 항에 있어서,

상기 적어도 하나의 디스패리티 벡터를 결정하는 것은, 상기 명령들이 또한, 2개 이상의 이웃 심도 샘플들에 적어도 부분적으로 기초하여 하나의 디스패리티 벡터를 결정하도록 상기 비디오 디코딩 디바이스를 구성하는 것을



포함하고,

상기 명령들은 또한,

상기 2개 이상의 이웃 심도 샘플들 중 하나의 이웃 심도 샘플을 선택하도록

상기 비디오 디코딩 디바이스를 구성하고,

상기 2개 이상의 이웃 심도 샘플들 중 하나의 이웃 심도 샘플을 선택하는 것은,

선택된 상기 이웃 심도 샘플을 식별하는 인덱스 값을 결정하는 것으로서, 상기 인덱스 값은 블록 레벨에서 시그널링되는, 상기 선택된 상기 이웃 심도 샘플을 식별하는 인덱스 값을 결정하는 것, 및

상기 2개 이상의 이웃 심도 샘플들 중 하나를 선택하기 위해 상기 2개 이상의 이웃 심도 샘플들의 값들에 수학적 함수를 적용하는 것

중 적어도 하나에 기초하는, 컴퓨터 판독가능 저장 매체.

#### 청구항 19

심도 뷰 데이터를 인코딩하는 방법으로서,

비디오 데이터의 픽처의 슬라이스의 각각의 코딩 유닛 (CU) 의 각각의 예측 유닛 (PU) 에 대해:

적어도 하나의 이웃 샘플의 적어도 하나의 재구성된 심도 샘플의 적어도 하나의 심도 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 값을 결정하는 단계;

상기 적어도 하나의 디스패리티 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 벡터를 결정하는 단계로서, 상기 적어도 하나의 디스패리티 벡터는 상기 각각의 CU 에 대한 것인, 상기 적어도 하나의 디스패리티 벡터를 결정하는 단계; 및

상기 적어도 하나의 디스패리티 벡터에 적어도 부분적으로 기초하여 상기 각각의 CU 에 대한 코딩 블록의 인코딩된 표현을 생성하는 단계

를 포함하는, 심도 뷰 데이터를 인코딩하는 방법.

#### 청구항 20

제 19 항에 있어서,

상기 적어도 하나의 디스패리티 벡터를 결정하는 단계는, 2개 이상의 이웃 심도 샘플들에 적어도 부분적으로 기초하여 하나의 디스패리티 벡터를 결정하는 단계를 포함하고,

상기 방법은, 상기 2개 이상의 이웃 심도 샘플들 중 하나의 이웃 심도 샘플을 선택하는 단계를 더 포함하고,

상기 2개 이상의 이웃 심도 샘플들 중 하나의 이웃 심도 샘플을 선택하는 단계는,

선택된 상기 이웃 심도 샘플을 식별하는 인덱스 값을 결정하는 것으로서, 상기 인덱스 값은 블록 레벨에서 시그널링되는, 상기 선택된 상기 이웃 심도 샘플을 식별하는 인덱스 값을 결정하는 것, 및

상기 2개 이상의 이웃 심도 샘플들 중 하나를 선택하기 위해 상기 2개 이상의 이웃 심도 샘플들의 값들에 수학적 함수를 적용하는 것

중 적어도 하나에 기초하는, 심도 뷰 데이터를 인코딩하는 방법.

#### 청구항 21

제 19 항에 있어서,

상기 적어도 하나의 디스패리티 벡터를 결정하는 단계는, 2개 이상의 이웃 심도 샘플들에 적어도 부분적으로 기초하여 2개 이상의 디스패리티 벡터들을 결정하는 단계를 포함하고,

상기 방법은,

상기 2개 이상의 이웃 심도 샘플들에 기초하여 2개 이상의 디스패리티 값들을 상기 2개 이상의 디스패리티 벡터



들로 컨버팅하는 단계를 더 포함하는, 심도 뷰 데이터를 인코딩하는 방법.

#### 청구항 22

제 19 항에 있어서,

상기 디스패리티 벡터를 디스패리티 모션 벡터로 컨버팅하는 단계를 더 포함하는, 심도 뷰 데이터를 인코딩하는 방법.

#### 청구항 23

제 19 항에 있어서,

상기 적어도 하나의 디스패리티 값을 결정하는 단계는, 상기 각각의 PU 의 2개 이상의 이웃 샘플들에 적어도 부분적으로 기초하여 2개 이상의 디스패리티 값들을 결정하는 단계를 더 포함하고,

상기 적어도 하나의 디스패리티 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 벡터를 결정하는 단계는, 상기 2개 이상의 디스패리티 값들에 적어도 부분적으로 기초하여 2개 이상의 디스패리티 벡터들을 결정하는 단계를 더 포함하고,

상기 PU 의 상부 좌측 샘플은 좌표들 (x, y) 을 가지며,

2개 이상의 상기 재구성된 심도 샘플들은,

상기 PU 의 상부 좌측 이웃 샘플,

상기 PU 의 상부 이웃 로우,

상기 PU 의 좌측 이웃 컬럼,

상기 각각의 CU 의 특정 이웃 블록에 로케이팅된 임의의 샘플,

좌표들 (x-1, y+1) 을 갖는 샘플,

좌표들 (x+1, y-1) 을 갖는 샘플, 그리고,

PU 사이즈가 WxH 로 표기될 때, 좌표들 (x-1, y+H-1) 을 갖는 샘플 및 좌표들 (x+W-1, y-1) 을 갖는 샘플

중 적어도 2개 이상으로부터의 것인, 심도 뷰 데이터를 인코딩하는 방법.

#### 청구항 24

제 23 항에 있어서,

상기 적어도 하나의 디스패리티 값을 결정하는 단계는, 상기 각각의 CU 의 2개 이상의 이웃 샘플들에 적어도 부분적으로 기초하여 2개 이상의 디스패리티 값들을 결정하는 단계를 더 포함하고,

상기 각각의 CU 의 상부 좌측 샘플은 좌표들 (x, y) 을 가지며,

2개 이상의 상기 재구성된 심도 샘플들은,

좌표들 (x-1, y+1) 을 갖는 샘플,

좌표들 (x+1, y-1) 을 갖는 샘플, 그리고,

각각의 CU 사이즈가 2Nx2N 으로 표기될 때, 좌표들 (x-1, y+2N-1) 을 갖는 샘플 및 좌표들 (x+2N-1, y-1) 을 갖는 샘플

중 적어도 2개 이상으로부터의 것인, 심도 뷰 데이터를 인코딩하는 방법.

#### 청구항 25

제 23 항에 있어서,

상기 2개 이상의 이웃 샘플들은 상기 PU 의 최대 코딩 유닛 (LCU) 의 이웃 샘플들인, 심도 뷰 데이터를 인코딩하는 방법.



#### 청구항 26

제 19 항에 있어서,  
 상기 적어도 하나의 재구성된 심도 샘플은,  
 상기 PU 의 상부 좌측 이웃 샘플,  
 상기 PU 의 상부 이웃 로우,  
 상기 PU 의 좌측 이웃 컬럼, 및  
 상기 PU 의 특정 이웃 블록에 로케이팅된 임의의 샘플  
 중 적어도 하나로부터의 것인, 심도 뷰 데이터를 인코딩하는 방법.

#### 청구항 27

제 19 항에 있어서,  
 상기 적어도 하나의 디스패리티 벡터를 결정하는 단계는, 수행된 이웃 기반 디스패리티 벡터 유도 (NBDV) 프로세스가 임의의 이용가능한 디스패리티 벡터를 식별하는 것을 실패했음에 응답하는, 심도 뷰 데이터를 인코딩하는 방법.

#### 청구항 28

제 19 항에 있어서,  
 상기 디스패리티 벡터에 기초하여 참조 블록을 식별하는 단계;  
 식별된 상기 참조 블록의 참조 인덱스 및 모션 벡터에 적어도 부분적으로 기초하여 상기 PU 의 현재 모션을 예측하는 단계; 및  
 상기 참조 인덱스 및 모션 벡터에 적어도 부분적으로 기초하여 결정된 후보를 병합 또는 어드밴스드 모션 벡터 예측 (AMVP) 후보 리스트 내에 삽입하는 단계  
 를 더 포함하는, 심도 뷰 데이터를 인코딩하는 방법.

#### 청구항 29

제 19 항에 있어서,  
 상기 디스패리티 값에 적어도 부분적으로 기초하여 디스패리티 벡터를 결정하는 것은, 상기 디스패리티 값 및 카메라 파라미터들에 적어도 부분적으로 기초하여 상기 디스패리티 벡터를 결정하는 것을 더 포함하는, 심도 뷰 데이터를 인코딩하는 방법.

#### 청구항 30

제 29 항에 있어서,  
 하나 이상의 카메라 파라미터들은 2개의 뷰들의 수평 변위를 포함하는, 심도 뷰 데이터를 인코딩하는 방법.

### 명세서

#### 기술분야

[0001] 본 출원은 2013년 5월 31일자로 출원된 미국 가출원 제61/829,919호의 이익을 주장하고, 그 전체 내용은 여기에 참조로 포함된다.

[0002] 기술분야

[0003] 본 개시물은 비디오 코딩 (즉, 비디오 데이터의 인코딩 및/또는 디코딩) 에 관한 것이다.

#### 배경 기술



[0004] 디지털 비디오 능력들은 디지털 텔레비전들, 디지털 다이렉트 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인 휴대 정보 단말기들(PDAs), 랩톱 또는 데스크톱 컴퓨터들, 태블릿 컴퓨터들, 전자책 판독기들(e-book readers), 디지털 카메라들, 디지털 레코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 라디오 전화기들, 소위 "스마트 폰들", 비디오 화상회의 디바이스들, 비디오 스트리밍 디바이스들 등을 포함하는 광범위한 디바이스들에 통합될 수 있다. 디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Part 10, 어드밴스드 비디오 코딩(Advanced Video Coding; AVC)에 의해 정의된 표준들, 현재 개발 중인 고효율 비디오 코딩(High Efficiency Video Coding; HEVC) 표준, 및 이러한 표준들의 확장들에서 설명되는 것들과 같은 비디오 압축 기법들을 구현한다. 비디오 디바이스들은 이러한 비디오 압축 기법들을 구현함으로써 디지털 비디오 정보를 더욱 효율적으로 송신, 수신, 인코딩, 디코딩, 및/또는 저장할 수도 있다.

[0005] 비디오 압축 기법들은 비디오 시퀀스들에 내재하는 리던던시를 감소 또는 제거하기 위해 공간(인트라-픽처) 예측 및/또는 시간(인터-픽처) 예측을 수행한다. 블록 기반 비디오 코딩의 경우, 비디오 슬라이스(즉, 비디오 프레임 또는 비디오 프레임의 일부분)는 비디오 블록들로 파티셔닝될 수도 있다. 픽처의 인트라-코딩된(I) 슬라이스에서의 비디오 블록들은 동일한 픽처 내의 이웃 블록들에서의 참조 샘플들에 대한 공간 예측을 사용하여 인코딩된다. 픽처의 인터-코딩된(P 또는 B) 슬라이스에서의 비디오 블록들은 동일한 픽처 내의 이웃 블록들에서의 참조 샘플들에 대한 공간 예측 또는 다른 참조 픽처들 내의 참조 샘플들에 대한 시간 예측을 사용할 수도 있다. 픽처들은 프레임들이라고 지칭될 수도 있다. 참조 픽처들은 참조 프레임들이라고 지칭될 수도 있다.

[0006] 공간 또는 시간 예측은 코딩될 블록에 대한 예측 블록을 발생시킨다. 잔차 데이터는 코딩될 오리지널 블록과 예측 블록 사이의 픽셀 차이들을 나타낸다. 인터-코딩된 블록은 예측 블록을 형성하는 참조 샘플들의 블록을 가리키는 모션 벡터에 따라 인코딩되며, 잔차 데이터는 코딩된 블록과 예측 블록 사이의 차이를 나타낸다. 인트라-코딩된 블록은 인트라-코딩 모드 및 잔차 데이터에 따라 인코딩된다. 추가의 압축을 위해, 잔차 데이터는 픽셀 도메인으로부터 변환 도메인으로 변환되어, 잔차 계수들을 발생시킬 수도 있으며, 이 잔차 계수들은 그 후에 양자화될 수도 있다. 초기에 2차원 어레이로 배열된 양자화된 계수들은 계수들의 1차원 벡터를 생성하기 위해 스캐닝될 수도 있고, 엔트로피 코딩이 훨씬 더 많은 압축을 달성하기 위해 적용될 수도 있다.

[0007] 멀티뷰 코딩 비트스트림은 뷰들을, 예를 들어, 다수의 관점들로부터 인코딩함으로써 생성될 수도 있다. 개발되었거나 또는 개발 중에 있는 일부 3차원(3D) 비디오 표준들은 멀티뷰 코딩 양태들을 사용한다. 예를 들어, 상이한 뷰들은 3D 비디오를 지원하기 위해 좌안(left eye) 및 우안 뷰들을 송신할 수도 있다. 대안적으로, 일부 3D 비디오 코딩 프로세스들은 소위 멀티뷰 플러스 심도 코딩(multiview plus depth coding)을 적용할 수도 있다. 멀티뷰 플러스 심도 코딩에서, 3D 비디오 비트스트림은 텍스처 뷰 컴포넌트들뿐만 아니라, 심도 뷰 컴포넌트들도 포함할 수도 있다. 예를 들어, 각각의 뷰는 하나의 텍스처 뷰 컴포넌트 및 하나의 심도 뷰 컴포넌트를 포함할 수도 있다.

## 발명의 내용

### 과제의 해결 수단

[0008] 일반적으로, 본 개시물은 3D-HEVC 코덱에 의한 2개 이상의 뷰들 플러스 심도의 코딩을 포함하여, 어드밴스드 코덱들에 기초한 3D 비디오 코딩에 관한 것이다. 구체적으로는, 본 개시물은 심도 뷰 코딩을 위한 디스패리티 벡터들에 관련된 기법들을 설명한다.

[0009] 하나의 예에서, 본 개시물은 비디오 데이터를 디코딩하는 방법을 설명한다. 비디오 데이터의 픽처의 슬라이스의 각각의 코딩 유닛(coding unit; CU)의 각각의 예측 유닛(prediction unit; PU)에 대해, 비디오 데이터를 디코딩하는 방법은, 적어도 하나의 이웃 샘플의 적어도 하나의 재구성된 심도 샘플의 적어도 하나의 심도 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 값을 결정하는 단계를 포함한다. 또한, 이 방법은 적어도 하나의 디스패리티 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 벡터를 결정하는 단계를 포함하고, 여기서 적어도 하나의 디스패리티 벡터는 각각의 CU에 대한 것이다. 이 방법은 또한 적어도 하나의 디스패리티 벡터에 적어도 부분적으로 기초하여 각각의 CU에 대한 코딩 블록을 재구성하는 단계를 포함한다.

[0010] 다른 예에서, 본 개시물은 픽처와 연관된 데이터를 저장하도록 구성된 메모리 및 그 메모리와 통신하고 구성된 하나 이상의 프로세서들을 포함하는 비디오 코딩 디바이스에 관련되며, 비디오 데이터의 픽처의 슬라이스의 각



각의 CU 의 각각의 PU 에 대해, 하나 이상의 프로세서들은 적어도 하나의 이웃 샘플의 적어도 하나의 재구성된 심도 샘플의 적어도 하나의 심도 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 값을 결정하도록 구성된다. 또한, 하나 이상의 프로세서들은 적어도 하나의 디스패리티 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 벡터를 결정하도록 구성되고, 여기서 적어도 하나의 디스패리티 벡터는 각각의 CU 에 대한 것이다. 하나 이상의 프로세서들은 또한 적어도 하나의 디스패리티 벡터에 적어도 부분적으로 기초하여 각각의 CU 에 대한 코딩 블록을 재구성하도록 구성된다.

[0011]

다른 예에서, 본 개시물은, 실행될 때, 본 개시물에 따라 기법들을 수행하도록 비디오 디코딩 디바이스를 구성하는 명령들을 저장한 컴퓨터 판독가능 저장 매체에 관련된다. 비디오 데이터의 픽처의 슬라이스의 각각의 CU 의 각각의 PU 에 대해, 명령들은 적어도 하나의 이웃 샘플의 적어도 하나의 재구성된 심도 샘플의 적어도 하나의 심도 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 값을 결정하도록 비디오 디코딩 디바이스를 구성한다. 또한, 명령들은 적어도 하나의 디스패리티 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 벡터를 결정하도록 비디오 디코딩 디바이스를 구성하고, 여기서 적어도 하나의 디스패리티 벡터는 각각의 CU 에 대한 것이다. 명령들은 또한 적어도 하나의 디스패리티 벡터에 적어도 부분적으로 기초하여 각각의 CU 에 대한 코딩 블록을 재구성하도록 비디오 디코딩 디바이스를 구성한다.

[0012]

다른 예에서, 본 개시물은 비디오 데이터를 인코딩하는 방법을 설명한다. 비디오 데이터의 픽처의 슬라이스의 각각의 CU 의 각각의 PU 에 대해, 이 방법은 적어도 하나의 이웃 샘플의 적어도 하나의 재구성된 심도 샘플의 적어도 하나의 심도 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 값을 결정하는 단계를 포함한다. 이 방법은 또한 적어도 하나의 디스패리티 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 벡터를 결정하는 단계를 포함하고, 여기서 적어도 하나의 디스패리티 벡터는 각각의 CU 에 대한 것이다. 이 방법은 또한 적어도 하나의 디스패리티 벡터에 적어도 부분적으로 기초하여 각각의 CU 에 대한 코딩 블록의 인코딩된 표현을 생성하는 단계를 포함한다.

[0013]

다른 예에서, 본 개시물은 하나 이상의 프로세서들을 포함하는 비디오 코딩 디바이스에 관련된다. 비디오 데이터의 픽처의 슬라이스의 각각의 CU 의 각각의 PU 에 대해, 하나 이상의 프로세서들은 적어도 하나의 이웃 샘플의 적어도 하나의 재구성된 심도 샘플의 적어도 하나의 심도 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 값을 결정하도록 구성된다. 또한, 하나 이상의 프로세서들은 적어도 하나의 디스패리티 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 벡터를 결정하도록 구성되고, 여기서 적어도 하나의 디스패리티 벡터는 각각의 CU 에 대한 것이다. 하나 이상의 프로세서들은 또한 적어도 하나의 디스패리티 벡터에 적어도 부분적으로 기초하여 각각의 CU 에 대한 코딩 블록의 인코딩된 표현을 생성하도록 구성된다.

[0014]

다른 예에서, 본 개시물은, 실행될 때, 본 개시물에 따라 기법들을 수행하도록 비디오 인코딩 디바이스를 구성하는 명령들을 저장한 컴퓨터 판독가능 저장 매체에 관련된다. 비디오 데이터의 픽처의 슬라이스의 각각의 CU 의 각각의 PU 에 대해, 명령들은 적어도 하나의 이웃 샘플의 적어도 하나의 재구성된 심도 샘플의 적어도 하나의 심도 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 값을 결정하도록 비디오 디코딩 디바이스를 구성한다. 또한, 명령들은 적어도 하나의 디스패리티 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 벡터를 결정하도록 비디오 디코딩 디바이스를 구성하고, 여기서 적어도 하나의 디스패리티 벡터는 각각의 CU 에 대한 것이다. 명령들은 또한 적어도 하나의 디스패리티 벡터에 적어도 부분적으로 기초하여 각각의 CU 에 대한 코딩 블록의 인코딩된 표현을 생성하도록 비디오 디코딩 디바이스를 구성한다.

[0015]

본 개시물의 하나 이상의 예들의 상세들은 첨부 도면들 및 아래의 상세한 설명에서 제시된다. 다른 특성들, 목적들, 및 이점들은 이 설명, 도면들, 및 청구항들로부터 명백해질 것이다.

### 도면의 간단한 설명

[0016]

도 1 은 본 개시물에서 설명된 기법들을 활용할 수도 있는 일 예시적인 비디오 코딩 시스템을 예시하는 블록도이다.

도 2 는 일 예시적인 멀티-뷰 코딩 디코딩 순서를 예시하는 개념도이다.

도 3 은 본 개시물에서 설명된 기법들에 따른, 멀티-뷰 코딩을 위한 일 예시적인 예측 구조를 예시하는 개념도이다.

도 4 는 본 개시물에서 설명된 기법들에 따른, 이웃 기반 디스패리티 벡터 유도 (neighbor-based disparity vector derivation; NBDV) 를 위한 현재 예측 유닛 (PU) 의 공간 이웃 블록들을 예시하는 개념도이다.



도 5 는 본 개시물에서 설명된 기법들에 따른, 역방향 워핑 (backward warping) 에 기초하여 블록 기반 뷰 합성 예측 (block-based view synthesis prediction; BVSP) 을 하기 위한 참조 뷰로부터의 심도 블록 유도의 일 예시적인 가시화의 개념도이다.

도 6 은 본 개시물에서 설명된 기법들에 따른, 어드밴스드 잔차 예측 (advanced residual prediction; ARP) 의 일 예시적인 예측 구조를 예시하는 개념도이다.

도 7 은 본 개시물에서 설명된 기법들에 따른, 현재 심도 블록의 디스패리티 벡터를 유도하는데 사용된 예시적인 참조 샘플들을 예시하는 개념도이다.

도 8 은 본 개시물에서 설명된 기법들을 구현할 수도 있는 일 예시적인 비디오 인코더를 예시하는 블록도이다.

도 9 는 본 개시물에서 설명된 기법들을 구현할 수도 있는 일 예시적인 비디오 디코더를 예시하는 블록도이다.

도 10 은 본 개시물에서 하나 이상의 양태들에 따른 일 예시적인 디코딩 기법을 예시하는 흐름도이다.

도 11 은 본 개시물에서 하나 이상의 양태들에 따른 일 예시적인 인코딩 기법을 예시하는 흐름도이다.

### 발명을 실시하기 위한 구체적인 내용

[0017]

3D-HEVC 에서의 심도 코딩 (depth coding) 은, 인터-뷰 참조 픽처를 활용하는 통상적인 모션 보상인, HEVC 코딩 및 디스패리티 보상의 상위에 구축된다. 심도 정보를 보다 양호하게 나타내기 위해, 3D-HEVC 인트라 코딩에서, 몇몇의 향상 심도 코딩 툴들이 적용된다. 향상 심도 코딩 툴들은 모두 비-정사각형 파티션들로의 심도 블록들의 분리를 가능하게 한다. 이러한 파티션 기반 심도 인트라 코딩 모드들은 DMM (Depth Modeling Modes), RBC (Region Boundary Chain) 코딩 및 SDC (Simplified Depth Coding) 를 포함한다. DMM, RBC 또는 SDC 에서, 각각의 심도 PU 는 1개 또는 2개의 부분들로 분할될 수 있고, 각각의 부분은 상수 값, 즉, DC 값으로 나타낸다. 각각의 파티션에 대한 DC 값은 이웃 참조 샘플들을 사용하여 예측되고, 잔차 값은 추가로 코딩되어 예측 에러를 보상할 수도 있다.

[0018]

DMM 과 RBC 방법들 양쪽 모두가 심도 PU 를 2개의 부분들로 파티셔닝하지만, 이들은 파티셔닝 패턴의 표현에 대해서는 다르다. DMM 에서, 웨지렛 (wedgelet) 및 윤곽 (contour) 패턴들을 포함하는, 2개의 타입들의 파티셔닝 패턴들이 적용된다. 웨지렛 패턴은 심도 PU 를 특정한 직선으로 세그먼트화한다. 웨지렛과는 달리, RBC 는 명시적으로 일련의 연결된 체인들을 사용하여 파티셔닝 패턴을 나타내고, 각각의 체인은, 하나의 샘플과, 0 에서부터 7 까지 인덱싱되는 그의 8-연결성 샘플들 중 하나와의 연결일 수도 있고, 윤곽 패턴은 2개의 불규칙적인 파티션들로의 심도 PU 의 파티션을 지원할 수 있고, 그 각각은 개별 서브-영역들을 포함할 수도 있다. 심도 블록의 윤곽의 파티션은 코-로케이팅된 (co-located) 텍스처를 분석함으로써 계산될 수도 있다. 또한, 1) 웨지렛 패턴들의 미리 정의된 세트로부터 선택된 웨지렛 패턴 인덱스의 명시적 시그널링; 2) 웨지렛 패턴과 이웃 인트라 예측 모드를 사용하는 현재 웨지렛 패턴의 예측; 및 3) 재구성되는 코-로케이팅된 텍스처 블록에 기초한 파티셔닝 패턴의 유도를 포함하는, 파티셔닝 패턴을 시그널링하기 위한 상이한 방법들이 웨지렛 모드들에서 활용된다.

[0019]

MPI (Motion Parameter Inheritance) 는 병합 후보 리스트 생성을 변경함으로써 구현되었다. HEVC 병합 모드에 의해 생성된 오리지널 공간 및 시간 후보들에 부가적으로, 코-로케이팅된 텍스처 블록의 모션 정보로부터 생성될 수도 있는 하나의 부가적인 병합 후보가 현재 심도 블록의 병합 리스트 내에 부가될 수도 있다.

[0020]

어드밴스드 인터-뷰 코딩 스킴들은 심도 코딩에 적용되지 않는다. 심도 코딩에서, 디스패리티 벡터를 유도하기 위해, 텍스처 코딩에 사용되는, 이웃 기반 디스패리티 벡터 유도 (neighbor-based disparity vector derivation; NBDV) 스킴을 활용하는 것은 덜 정확하고 보다 많이 복잡할 수도 있다. 따라서, 본 개시물에서 설명된 기법들은 이웃 블록들에서 하나 이상의 재구성된 심도 샘플들로부터 현재 심도 블록에 대한 디스패리티 벡터를 생성한다. 유도된 디스패리티 벡터는 인터-뷰 모션 예측 및 어드밴스드 잔차 예측을 포함하는 어드밴스드 인터-뷰 코딩 툴들에 대해 활용될 수도 있다.

[0021]

심도 블록에 대한 디스패리티 벡터 유도는 PU 레벨 또는 CU 레벨에서 발생할 수 있다. CU-레벨 디스패리티 벡터 유도에서, CU 의 모든 PU들에 대해, 현재 PU 를 커버하는 코딩 유닛 (coding unit; CU) 의 이웃 블록들에서 재구성된 심도 샘플들 중 하나의 재구성된 심도 샘플의 심도 값이 활용되어 주어진 참조 뷰의 대응하는 블록에 대한 디스패리티 벡터로 컨버팅될 수도 있다. 디스패리티 값으로부터 디스패리티 벡터로의 컨버전은 카메라 파라미터들에 기초할 수도 있다. 참조 뷰에 각각 대응하는 다수의 디스패리티 벡터들이 유도될 수도



있다.

[0022] 예를 들어, 적어도 하나의 디스패리티 값을 결정함에 있어서, 시스템은 각각의 PU의 2개 이상의 이웃 샘플들에 적어도 부분적으로 기초하여 2개 이상의 디스패리티 값들을 결정할 수도 있다. 적어도 하나의 디스패리티 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 벡터를 결정함에 있어서, 시스템은 2개 이상의 디스패리티 값들에 적어도 부분적으로 기초하여 2개 이상의 디스패리티 벡터들을 결정할 수도 있다. PU의 상부 좌측 샘플은 좌표들  $(x, y)$  을 가지며, 2개 이상의 재구성된 심도 샘플들은 PU의 상부 이웃 로우(row), PU의 좌측 이웃 컬럼(column), 각각의 PU의 특정 이웃 블록에 로케이팅된 임의의 샘플, 좌표들  $(x-1, y+1)$  을 갖는 샘플, 좌표들  $(x+1, y-1)$  을 갖는 샘플, 그리고, PU의 사이즈가  $W \times H$  로 표기될 때, 좌표들  $(x-1, y+H-1)$  을 갖는 샘플 및 좌표들  $(x+W-1, y-1)$  을 갖는 샘플 중 적어도 2개 이상으로부터의 것이다.

[0023] PU를 커버하는 CU의 이웃 샘플들은 다음과 같이 선택될 수도 있다. 현재 PU를 커버하는 CU내의 상부 좌측 샘플의 좌표들은  $(x, y)$  로서 표기될 수도 있다. 하나의 예에서, 현재 PU를 커버하는 CU의 상부 좌측 이웃 샘플이 선택될 수도 있다. 상부 좌측 이웃 샘플은  $(x-1, y-1)$  의 좌표들을 갖는다. 부가적으로, 현재 PU를 커버하는 CU의 좌측 이웃 컬럼 또는 상부 이웃 로우에 로케이팅된 임의의 샘플이 사용될 수도 있다. 또한, 현재 PU를 커버하는 CU의 공간 이웃 블록들에 로케이팅된 임의의 샘플이 사용될 수도 있다.

[0024] 대안적으로, 현재 PU가 코딩 중일 때, 현재 PU를 커버하는 CU의 하나보다 많은 이용가능한 이웃 샘플들이 현재 PU에 대한 하나 이상의 디스패리티 벡터들을 유도하는데 사용될 수도 있다. 예를 들어, 좌표  $(x, y-1)$  를 갖는 다른 상부 샘플 및 좌표들  $(x-1, y)$  을 갖는 다른 좌측 샘플이 사용될 수도 있다. 부가적으로,  $(x-1, y+1)$  또는  $(x+1, y-1)$  의 좌표들을 갖는 샘플이 사용될 수도 있다. CU의 사이즈가  $2N \times 2N$  으로 표기될 수도 있을 때,  $(x-1, y+2N-1)$  및  $(x+2N-1, y-1)$  의 좌표들을 갖는 임의의 샘플이 사용될 수도 있다. 또한, 상부 이웃 로우, 좌측 이웃 컬럼의 임의의 샘플, 또는 현재 PU를 커버하는 CU의 공간 이웃 블록들의 임의의 샘플들이 사용될 수도 있다.

[0025] 예를 들어, 적어도 하나의 디스패리티 값을 결정함에 있어서, 시스템은 각각의 CU의 2개 이상의 이웃 샘플들에 적어도 부분적으로 기초하여 2개 이상의 디스패리티 값들을 결정할 수도 있다. 각각의 CU의 상부 좌측 샘플은 좌표들  $(x, y)$  을 가지며, 2개 이상의 재구성된 심도 샘플들은 각각의 CU의 상부 좌측 이웃 샘플, 각각의 CU의 상부 이웃 로우, 각각의 CU의 좌측 이웃 컬럼, 각각의 CU의 특정 이웃 블록에 로케이팅된 임의의 샘플, 좌표들  $(x-1, y+1)$  을 갖는 샘플, 좌표들  $(x+1, y-1)$  을 갖는 샘플, 그리고, 각각의 CU의 사이즈가  $2N \times 2N$  으로 표기될 때, 좌표들  $(x-1, y+2N-1)$  을 갖는 샘플 및 좌표들  $(x+2N-1, y-1)$  을 갖는 샘플 중 적어도 2개 이상으로부터의 것이다.

[0026] 대안적으로, 현재 PU가 코딩 중일 때, 현재 PU를 커버하는 최대 CU (largest CU; LCU)의 하나 이상의 이용가능한 이웃 샘플들이 현재 PU에 대한 하나 이상의 디스패리티 벡터들을 유도하는데 사용될 수도 있다. 일부 예들에서, 상술된 샘플들 중 임의의 샘플이 LCU에 대해 사용될 수도 있다.

[0027] PU-레벨 디스패리티 벡터 유도의 경우, 각각의 PU에 대해, 이웃 블록들에서 재구성된 심도 샘플들 중 하나의 재구성된 심도 샘플의 심도 값이 활용되어 주어진 참조 뷰의 대응하는 블록에 대한 디스패리티 벡터로 컨버팅될 수도 있다. 디스패리티 값으로부터 디스패리티 벡터로의 컨버전은 카메라 파라미터들에 기초할 수도 있다. 참조 뷰에 각각 대응하는 다수의 디스패리티 벡터들이 유도될 수도 있다.

[0028] 도 1은 본 개시물에서 설명된 기법들을 활용할 수도 있는 일 예시적인 비디오 코딩 시스템 (10)을 예시하는 블록도이다. 여기에 사용되는 바와 같이, 용어 "비디오 코더"는 비디오 인코더들과 비디오 디코더들 양쪽 모두를 포괄적으로 지칭한다. 본 개시물에서 용어를 "비디오 코딩" 또는 "코딩"은 비디오 인코딩 또는 비디오 디코딩을 포괄적으로 지칭할 수도 있다.

[0029] 도 1에 도시된 바와 같이, 비디오 코딩 시스템 (10)은 소스 디바이스 (12) 및 목적지 디바이스 (14)를 포함한다. 소스 디바이스 (12)는 인코딩된 비디오 데이터를 생성한다. 이에 따라, 소스 디바이스 (12)는 비디오 인코딩 디바이스 또는 비디오 인코딩 장치라고 지칭될 수도 있다. 목적지 디바이스 (14)는 소스 디바이스 (12)에 의해 생성된 인코딩된 비디오 데이터를 디코딩할 수도 있다. 이에 따라, 목적지 디바이스 (14)는 비디오 디코딩 디바이스 또는 비디오 디코딩 장치라고 지칭될 수도 있다. 소스 디바이스 (12) 및 목적지 디바이스 (14)는 비디오 코딩 디바이스들 또는 비디오 코딩 장치들의 예들일 수도 있다.

[0030] 소스 디바이스 (12) 및 목적지 디바이스 (14)는 데스크톱 컴퓨터들, 모바일 컴퓨팅 디바이스들, 노트북 (즉, 랩톱) 컴퓨터들, 태블릿 컴퓨터들, 셋-톱 박스들, 소위 "스마트" 폰들과 같은 전화기 핸드셋들, 텔레비전들, 카



메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 콘솔들, 차량내 컴퓨터들 등을 포함하는 광범위한 디바이스들을 포함할 수도 있다.

[0031] 목적지 디바이스 (14) 는 소스 디바이스 (12) 로부터 채널 (16) 을 통해 인코딩된 비디오 데이터를 수신할 수도 있다. 채널 (16) 은 인코딩된 비디오 데이터를 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 이동시키는 것이 가능한 하나 이상의 매체들 또는 디바이스들을 포함할 수도 있다. 하나의 예에서, 채널 (16) 은 소스 디바이스 (12) 로 하여금 인코딩된 비디오 데이터를 직접 목적지 디바이스 (14) 로 실시간으로 송신할 수 있게 하는 하나 이상의 통신 매체들을 포함할 수도 있다. 이 예에서, 소스 디바이스 (12) 는 무선 통신 프로토콜과 같은 통신 표준에 따라, 인코딩된 비디오 데이터를 변조할 수도 있으며, 변조된 비디오 데이터를 목적지 디바이스 (14) 로 송신할 수도 있다. 하나 이상의 통신 매체들은 무선 및/또는 유선 통신 매체들, 예컨대, 무선 주파수 (RF) 스펙트럼 또는 하나 이상의 물리적인 송신 라인들을 포함할 수도 있다. 하나 이상의 통신 매체들은 패킷 기반 네트워크, 예컨대, 근거리 네트워크, 광역 네트워크, 또는 글로벌 네트워크 (예를 들어, 인터넷) 의 일부를 형성할 수도 있다. 하나 이상의 통신 매체들은 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로의 통신을 용이하게 하는 다른 장비를 포함할 수도 있다.

[0032] 다른 예에서, 채널 (16) 은 소스 디바이스 (12) 에 의해 생성되는 인코딩된 비디오 데이터를 저장하는 저장 매체를 포함할 수도 있다. 이 예에서, 목적지 디바이스 (14) 는 저장 매체에, 예를 들어, 디스크 액세스 또는 카드 액세스를 통해 액세스할 수도 있다. 저장 매체는 블루레이 디스크들, DVD들, CD-ROM들, 플래시 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 다른 적합한 디지털 저장 매체들과 같은 다양한 로컬-엑세스되는 데이터 저장 매체들을 포함할 수도 있다.

[0033] 추가 예에서, 채널 (16) 은 소스 디바이스 (12) 에 의해 생성되는 인코딩된 비디오 데이터를 저장하는 파일 서버 또는 다른 중간 저장 디바이스를 포함할 수도 있다. 이 예에서, 목적지 디바이스 (14) 는 스트리밍 또는 다운로드를 통해 파일 서버 또는 다른 중간 저장 디바이스에 저장된 인코딩된 비디오 데이터에 액세스할 수도 있다. 파일 서버는 인코딩된 비디오 데이터를 저장하고 그 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로 송신하는 것이 가능한 서버의 타입일 수도 있다. 예시적인 파일 서버들은 (예를 들어, 웹사이트용) 웹 서버들, 파일 전송 프로토콜 (FTP) 서버들, NAS (network attached storage) 디바이스들, 및 로컬 디스크 드라이브들을 포함한다.

[0034] 목적지 디바이스 (14) 는 인코딩된 비디오 데이터에 인터넷 연결과 같은 표준 데이터 연결을 통해 액세스할 수도 있다. 데이터 연결들의 예시적인 타입들은 무선 채널들 (예를 들어, Wi-Fi 연결들), 유선 연결들 (예를 들어, DSL, 케이블 모뎀 등), 또는 파일 서버 상에 저장된 인코딩된 비디오 데이터에 액세스하는데 적합한 양쪽의 조합들을 포함할 수도 있다. 파일 서버로부터의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 이 양쪽의 조합일 수도 있다.

[0035] 본 개시물의 기법들은 무선 애플리케이션들 또는 설정들로 제한되지 않는다. 이 기법들은 오버-디-에어 텔레비전 브로드캐스트들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, 예를 들어, 인터넷을 통한 스트리밍 비디오 송신들, 데이터 저장 매체 상의 저장을 위한 비디오 데이터의 인코딩, 데이터 저장 매체 상에 저장된 비디오 데이터의 디코딩, 또는 다른 애플리케이션들과 같은 다양한 멀티미디어 애플리케이션들의 지원 하에서 비디오 코딩에 적용될 수도 있다. 일부 예들에서, 비디오 코딩 시스템 (10) 은 비디오 스트리밍, 비디오 플레이어, 비디오 브로드캐스팅, 및/또는 비디오 전화통신과 같은 애플리케이션들을 지원하기 위해 1-방향 또는 2-방향 비디오 송신을 지원하도록 구성될 수도 있다.

[0036] 도 1 은 단지 예이며, 본 개시물의 기법들은 인코딩 디바이스와 디코딩 디바이스 사이의 임의의 데이터 통신을 반드시 포함하지는 않는 비디오 코딩 설정들 (예를 들어, 비디오 인코딩 또는 비디오 디코딩) 에 적용할 수도 있다. 다른 예들에서, 데이터는 로컬 메모리로부터 추출되거나, 네트워크를 통해 스트리밍되거나 또는 기타 등등이다. 비디오 인코딩 디바이스는 데이터를 인코딩하여 메모리에 저장할 수도 있거나, 및/또는 비디오 디코딩 디바이스는 메모리로부터 데이터를 추출하여 디코딩할 수도 있다. 많은 예들에서, 인코딩 및 디코딩은 서로 통신하지 않지만 단순히 데이터를 메모리로 인코딩하거나 및/또는 메모리로부터 데이터를 추출하여 디코딩하는 디바이스들에 의해 수행될 수도 있다.

[0037] 도 1 의 예에서, 소스 디바이스 (12) 는 비디오 소스 (18), 비디오 인코더 (20), 및 출력 인터페이스 (22) 를 포함한다. 일부 예들에서, 출력 인터페이스 (22) 는 변조기/복조기 (모뎀) 및/또는 송신기를 포함할 수도 있다. 비디오 소스 (18) 는 비디오 캡처 디바이스, 예를 들어, 비디오 카메라, 이전에 캡처된 비디오 데이터를 포함하는 비디오 아카이브, 비디오 콘텐츠 제공자로부터 비디오 데이터를 수신하기 위한 비디오 공급 인터



페이스, 및/또는 비디오 데이터 생성하기 위한 컴퓨터 그래픽스 시스템, 또는 비디오 데이터의 이러한 소스들의 조합을 포함할 수도 있다.

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

[0039] 본 개시물의 기법들은 비디오 인코더 (20) 와 같은 비디오 인코더의 엘리먼트들에 의해 수행될 수도 있다. 이 기법에서, 비디오 데이터의 픽처의 슬라이스의 각각의 코딩 유닛의 각각의 예측 유닛에 대해, 비디오 인코더 (20) 는 적어도 하나의 이웃 샘플의 적어도 하나의 재구성된 심도 샘플의 적어도 하나의 심도 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 값을 결정할 수도 있다. 비디오 인코더 (20) 는 적어도 하나의 디스패리티 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 벡터를 결정할 수도 있고, 여기서 적어도 하나의 디스패리티 벡터는 각각의 코딩 유닛에 대한 것이다. 비디오 인코더 (20) 는 적어도 하나의 디스패리티 벡터에 적어도 부분적으로 기초하여 각각의 코딩 유닛에 대한 코딩 블록의 인코딩된 표현을 생성할 수도 있다.

[0040] 도 1 의 예에서, 목적지 디바이스 (14) 는 입력 인터페이스 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (32) 를 포함한다. 일부 예들에서, 입력 인터페이스 (28) 는 수신기 및/또는 모뎀을 포함한다. 입력 인터페이스 (28) 은 인코딩된 비디오 데이터를 채널 (16) 을 통해 수신할 수도 있다. 디스플레이 디바이스 (32) 는 목적지 디바이스 (14) 와 통합될 수도 있고 또는 그 외부에 있을 수도 있다. 일반적으로, 디스플레이 디바이스 (32) 는 디코딩된 비디오 데이터를 디스플레이한다. 디스플레이 디바이스 (32) 는 다양한 디스플레이 디바이스들, 예컨대, 액정 디스플레이 (LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (OLED) 디스플레이, 또는 다른 타입의 디스플레이 디바이스들을 포함할 수도 있다.

[0041] 본 개시물의 기법들은 비디오 디코더 (30) 와 같은 비디오 디코더의 엘리먼트들에 의해 수행될 수도 있다. 비디오 데이터의 픽처의 슬라이스의 각각의 코딩 유닛의 각각의 예측 유닛에 대해, 비디오 디코더 (30) 는 적어도 하나의 이웃 샘플의 적어도 하나의 재구성된 심도 샘플의 적어도 하나의 심도 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 값을 결정할 수도 있다. 비디오 디코더 (30) 는 적어도 하나의 디스패리티 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 벡터를 결정할 수도 있고, 여기서 적어도 하나의 디스패리티 벡터는 각각의 코딩 유닛에 대한 것이다. 비디오 디코더 (30) 는 적어도 하나의 디스패리티 벡터에 적어도 부분적으로 기초하여 각각의 코딩 유닛에 대한 코딩 블록을 재구성할 수도 있다.

[0042] 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들 (DSPs), 주문형 집적 회로들 (ASICs), 필드 프로그래밍가능 게이트 어레이들 (FPGAs), 이산 로직, 하드웨어, 또는 임의의 이들의 조합들과 같은 다양한 적합한 회로부 중 임의의 회로부로서 구현될 수도 있다. 기법들이 소프트웨어로 부분적으로 구현되면, 디바이스는 소프트웨어용 명령들을 적합한 비일시적 컴퓨터 판독가능 저장 매체에 저장할 수도 있으며, 본 개시물의 기법들을 수행하기 위해 그 명령들을 하드웨어에서 하나 이상의 프로세서들을 사용하여 실행할 수도 있다. (하드웨어, 소프트웨어, 하드웨어와 소프트웨어의 조합 등을 포함한) 전술한 것 중 임의의 것이 하나 이상의 프로세서들인 것으로 간주될 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 인코더들 또는 디코더들에 포함될 수도 있으며, 이들 중 어느 쪽이든 각각의 디바이스에서 결합된 인코더/디코더 (CODEC) 의 일부로서 통합될 수도 있다.

[0043] 본 개시물은 일반적으로 특정 정보를 비디오 디코더 (30) 와 같은 다른 디바이스로 "시그널링" 하는 비디오 인코더 (20) 에 관련될 수도 있다. 용어 "시그널링" 은 일반적으로 압축된 비디오 데이터를 디코딩하는데 사용되는 선택스 엘리먼트들 및/또는 다른 데이터의 통신을 지칭할 수도 있다. 이러한 통신은 실시간 또는 거의-실시간으로 일어날 수도 있다. 대안적으로, 이러한 통신은 특정 기간에 걸쳐 일어날 수도 있는데, 예컨대, 선택스 엘리먼트들을, 예를 들어, 로컬로 액세스가능한 저장 디바이스 또는 파일 서버 또는 스트리밍 서버를 통해 원격으로 액세스가능한 저장 매체와 같은 컴퓨터 판독가능 저장 매체에 인코딩시에 인코딩된 비트스트림으로 저장할 때에 일어날 수도 있으며, 이 선택스 엘리먼트들은 그 후에 이 매체에 저장된 후에 언제라도 디코딩 디바이스에 의해 취출될 수도 있다.

[0044] 일부 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 ISO/IEC MPEG-4 Visual 및 ITU-T H.264 (또한 ISO/IEC MPEG-4 AVC 로서 알려짐) 과 같은, 그의 스케일러블 비디오 코딩 (SVC) 확장, 멀티뷰 비디오 코딩 (MVC) 확장, 및 MVC 기반 3DV 확장을 포함한, 비디오 압축 표준에 따라 동작한다. 일부 경우들에서, MVC 기



반 3DV 를 준수하는 임의의 적법한 비트스트림은 MVC 프로파일, 예를 들어, 스테레오 하이 프로파일에 순응할 수도 있는 서브-비트스트림을 항상 포함한다. H.264/AVC 의 MVC 확장의 합동 초안은 「"Advanced video coding for generic audio visual services", ITU-T 권고안 H.264, 2010년 3월」에 설명되어 있다. 또한, H.264/AVC 에 대한 3차원 비디오 (3DV) 코딩 확장, 즉, AVC 기반 3DV 를 생성하려는 노력이 지속되고 있다. 다른 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 또는 ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual, 및 ITU-T H.264, ISO/IEC Visual 에 따라 동작할 수도 있다.

[0045]

도 1 의 예에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 ITU-T 비디오 코딩 전문가 그룹 (VCEG) 과 ISO/IEC 동화상 전문가 그룹 (MPEG) 의 비디오 코딩에 대한 합동 협력팀 (JCT-VC) 에 의해 현재 개발 중인 HEVC (High Efficiency Video Coding) 표준에 따라 동작할 수도 있다. "HEVC Working Draft 10" 이라고 지칭되는, 업커밍 HEVC 표준의 초안은 「Bross 등, "High Efficiency Video Coding (HEVC) text specification draft 10", ITU-T SG16 WP3 과 ISO/IEC JTC1/SC29/WG11 의 비디오 코딩에 대한 합동 협력팀 (JCT-VC), 12차 회의, 스위스 제네바, 2013년 1월」에 설명되어 있고, 이는 2013년 5월 31일자로 [http://phenix.it-sudparis.eu/jct/doc\\_end\\_user/documents/12\\_Geneva/wg11/JCTVC-L1003-v34.zip](http://phenix.it-sudparis.eu/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v34.zip) 으로부터 입수가능하다.

[0046]

또한, HEVC 에 대한 스케일러블 비디오 코딩, 멀티-뷰 코딩 및 3DV 확장들을 생성하려는 노력들이 지속되고 있다. HEVC 의 SVC 확장은 SHEVC 라고 지칭될 수도 있다. HEVC 의 3DV 확장은 HEVC 기반 3DV 또는 3D-HEVC 라고 지칭될 수도 있다. 3D-HEVC 는, 「Schwarz 등, "Description of 3D Video Coding Technology Proposal by Fraunhofer HHI (HEVC compatible configuration A), ISO/IEC JTC1/SC29/WG11, Doc. MPEG11/M22570, 스위스 제네바, 2011년 11월/12월, 이하 "m22570"」 및 「Schwarz 등, "Description of 3D Video Coding Technology Proposal by Fraunhofer HHI (HEVC compatible configuration B), ISO/IEC JTC1/SC29/WG11, Doc. MPEG11/M22571, 스위스 제네바, 2011년 11월/12월, 이하 "m22571"」에서 제안된 솔루션들에 적어도 부분적으로 기초한다. 3D-HEVC 에 대한 참조 소프트웨어 설명은 「Schwarz 등, "Test Model under Consideration for HEVC based 3D video coding", ISO/IEC JTC1/SC29/WG11 MPEG2011/N12559, 미국 새너제이, 2012년 2월」에서 입수가능하다. 참조 소프트웨어, 즉, HTM 버전 3.0 은, 2013년 5월 31일자로 [https://hevc.hhi.fraunhofer.de/svn/svn\\_3DVCSsoftware/tags/HTM-3.0/](https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSsoftware/tags/HTM-3.0/) 으로부터 입수가능하다.

[0047]

H.264/AVC, HEVC, 및 다른 비디오 코딩 표준들에서, 비디오 시퀀스는 일련의 픽처들을 포함할 수도 있다. 픽처들은 또한 "프레임들" 로서 지칭될 수도 있다. 픽처는  $S_L$ ,  $S_{Cb}$ , 및  $S_{Cr}$  로 표기되는, 3개의 샘플 어레이들을 포함할 수도 있다.  $S_L$  은 루마 샘플들의 2차원 어레이 (즉, 블록) 이다.  $S_{Cb}$  는 Cb 크로미넌스 샘플들의 2차원 어레이이다.  $S_{Cr}$  은 Cr 크로미넌스 샘플들의 2차원 어레이이다. 크로미넌스 샘플들은 또한 여기에서 "크로마" 샘플들로서 지칭될 수도 있다. 다른 경우들에서, 픽처는 단색일 수도 있으며 단지 루마 샘플들의 어레이를 포함할 수도 있다.

[0048]

H.264/AVC 에서, 각각의 픽처는 매크로블록 (MB) 들의 세트로 파티셔닝될 수도 있다. 매크로블록은 3개의 샘플 어레이들을 갖는 픽처의 크로마 샘플들의 2개의 대응하는 블록들 및 루마 샘플들의 16x16 블록, 또는 3개의 별개의 컬러 평면들을 사용하여 코딩되는 픽처 또는 단색 픽처의 샘플들의 16x16 블록이다.

[0049]

비디오 인코더 (20) 는 인터 예측자 또는 인트라 예측을 사용하여 매크로블록들을 인코딩할 수도 있다. 비디오 인코더 (20) 가 인터 예측을 사용하여 매크로블록을 인코딩할 때, 비디오 인코더 (20) 는 현재 픽처 (즉, 매크로블록을 포함하는 픽처) 이외의 하나 이상의 픽처들의 샘플들에 기초하여 매크로블록에 대한 하나 이상의 예측 블록들을 생성한다. 인터 예측을 사용하여 인코딩된 매크로블록은 인터 매크로블록이라고 지칭될 수도 있다. 비디오 인코더 (20) 가 인트라 예측을 사용하여 매크로블록을 인코딩할 때, 비디오 인코더 (20) 는 현재 픽처에서의 샘플들에 기초하여 매크로블록에 대한 하나 이상의 예측 블록들을 생성한다. 인트라 예측을 사용하여 인코딩된 매크로블록은 인트라 매크로블록이라고 지칭될 수도 있다.

[0050]

H.264/AVC 에서, 각각의 인터 매크로블록은 4개의 상이한 방법들로 파티셔닝될 수도 있다: 하나의 16x16 매크로블록 파티션, 2개의 16x8 매크로블록 파티션들, 2개의 8x16 매크로블록 파티션들, 또는 4개의 8x8 매크로블록 파티션들.

[0051]

하나의 MB 에서 상이한 MB 파티션들은 각각의 방향에 대한 상이한 참조 인덱스 값들을 가질 수도 있다 (즉, RefPicList0 또는 RefPicList1). MB 가 4개의 8x8 MB 파티션들로 파티셔닝되지 않을 때, MB 는 각각의 방향에서 전체 MB 파티션에 대한 단지 하나의 모션 벡터를 가질 수도 있다.



- [0052] MB 가 4개의 8x8 MB 파티션들로 파티셔닝될 때, 각각의 8x8 MB 파티션은 서브-블록들로 추가로 파티셔닝될 수 있다. 8x8 MB 파티션으로부터 서브-블록들을 획득하기 위한 4개의 상이한 방법들이 존재한다: 하나의 8x8 서브-블록, 2개의 8x4 서브-블록들, 2개의 4x8 서브-블록들, 또는 4개의 4x4 서브-블록들.
- [0053] 각각의 서브-블록은 각각의 예측 방향에서 상이한 모션 벡터를 가질 수 있다. 8x8 MB 파티션이 서브-블록들로 파티셔닝되는 방법은 서브-블록 파티션으로 지칭된다.
- [0054] 상술된 바와 같이, 멀티-뷰 코딩 (MVC) 은 H.264/AVC 의 확장이다. 멀티-뷰 코딩에서, 상이한 뷰포인트들로부터의 동일한 장면의 다수의 뷰들이 존재할 수도 있다. 용어 "액세스 유닛" 은 동일한 시간 인스턴스에 대응하는 픽처들의 세트를 지칭하기 위해 사용된다. 따라서, 비디오 데이터는 시간 경과에 따라 발생하는 일련의 액세스 유닛들로서 개념화될 수도 있다. "뷰 컴포넌트" 는 단일 액세스 유닛에서의 뷰의 코딩된 표현일 수도 있다. 본 개시물에서, "뷰" 는 동일한 뷰 식별자와 연관된 뷰 컴포넌트들의 시퀀스를 지칭할 수도 있다.
- [0055] VCEG 및 MPEG 의 3D 비디오 코딩에 대한 합동 협력팀 (JCT-3V) 이 HEVC 에 기초한 3DV 표준을 개발하고 있고, 그 표준화 노력들의 일부는 HEVC 에 기초한 멀티뷰 비디오 코덱 (MV-HEVC) 의 표준화 및 HEVC 에 기초한 3D 비디오 코딩 (3D-HEVC) 에 대한 다른 부분을 포함한다. 3D-HEVC 의 경우, 텍스처 및 심도 뷰들 양쪽에 대해, 코딩 유닛/예측 유닛 레벨에서 이들을 포함하는 새로운 코딩 틀들이 포함되고 지원될 수도 있다. 3D-HEVC 에 대한 최근 소프트웨어 3D-HTM (3D-HTM 버전 6.0) 이 2013년 5월 31일자로 [https://hevc.hhi.fraunhofer.de/svn/svn\\_3DVCSoftware/tags/HTM-6.0/](https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-6.0/) 에서 입수가능하다.
- [0056] 3D-HEVC 의 작업 초안뿐만 아니라 최근 참조 소프트웨어 설명은 2013년 5월 31일자로 [http://phenix.it-sudparis.eu/jct2/doc\\_end\\_user/documents/3\\_Geneva/wg11/JCT3V-C1005-v2.zip](http://phenix.it-sudparis.eu/jct2/doc_end_user/documents/3_Geneva/wg11/JCT3V-C1005-v2.zip) 에서 입수가능하다. Gerhard Tech, Krzysztof Wegner, Ying Chen, Sehoon Yea, "3D-HEVC Test Model 3", JCT3V-C1005\_spec\_d1, ITU-T SG16 WP3 및 ISO/IEC JTC1/SC29/WG11 의 3D 비디오 코딩 확장 개발에 대한 합동 협력팀, 3차 회의: 스위스 제네바, 2013년 1월 17-23일.
- [0057] 픽처의 인코딩된 표현을 생성하기 위해, 비디오 인코더 (20) 는 코딩 트리 유닛 (CTU) 들의 세트를 생성할 수도 있다. CTU들 각각은 루마 샘플들의 코딩 트리 블록, 크로마 샘플들의 2개의 대응하는 코딩 트리 블록들, 및 코딩 트리 블록들의 샘플들을 코딩하는데 사용되는 선택스 구조들일 수도 있다. 코딩 트리 블록은 샘플들의 NxN 블록일 수도 있다. CTU 는 또한 "트리 블록" 또는 "최대 코딩 유닛" (LCU) 이라고 지칭될 수도 있다. HEVC 의 CTU들은 H.264/AVC 와 같은 다른 비디오 코딩 표준들의 매크로블록들과 대략 유사할 수도 있다. 그러나, CTU 는 특정 사이즈로 반드시 제한되지는 않으며, 하나 이상의 코딩 유닛 (CU) 들을 포함할 수도 있다. 슬라이스는 래스터 스캔에서 연속적으로 순서화된 정수개의 CTU들을 포함할 수도 있다.
- [0058] 본 개시물은 샘플들의 하나 이상의 블록들 및 그 샘플들의 하나 이상의 블록들의 샘플들을 코딩하는데 사용되는 선택스 구조들을 지칭하기 위해 용어 "비디오 유닛" 또는 "비디오 블록" 을 사용할 수도 있다. 비디오 유닛들의 예시적인 타입들은 CTU들, CU들, PU들, 변환 유닛 (TU) 들, 매크로블록들, 매크로블록 파티션들 등을 포함할 수도 있다.
- [0059] 코딩된 CTU 를 생성하기 위해, 비디오 인코더 (20) 는 CTU 의 코딩 트리 블록들에 대한 쿼드-트리 파티셔닝을 재귀적으로 수행하여, 코딩 트리 블록들을 코딩 블록들, 그에 따라 명칭 "코딩 트리 유닛들" 로 분할할 수도 있다. 코딩 블록은 샘플들의 NxN 블록이다. CU 는 루마 샘플들의 코딩 블록 및 루마 샘플 어레이, Cb 샘플 어레이 및 Cr 샘플 어레이를 갖는 픽처의 크로마 샘플들의 2개의 대응하는 코딩 블록들, 및 코딩 블록들의 샘플들을 코딩하는데 사용되는 선택스 구조들일 수도 있다. 비디오 인코더 (20) 는 CU 의 코딩 블록을 하나 이상의 예측 블록들로 파티셔닝할 수도 있다. 예측 블록은 동일한 예측이 적용되는 샘플들의 직사각형 (즉, 정사각형 또는 비-정사각형) 블록일 수도 있다. CU 의 예측 유닛 (prediction unit; PU) 은 루마 샘플들의 예측 블록, 픽처의 크로마 샘플들의 2개의 대응하는 예측 블록들, 및 예측 블록 샘플들을 예측하는데 사용되는 선택스 구조들일 수도 있다. 비디오 인코더 (20) 는 CU 의 각각의 PU 의 루마, Cb 및 Cr 예측 블록들에 대한 예측 루마, Cb 및 Cr 블록들을 생성할 수도 있다.
- [0060] 비디오 인코더 (20) 는 인트라 예측 또는 인터 예측을 사용하여 PU 에 대한 예측 블록들을 생성할 수도 있다. 비디오 인코더 (20) 가 PU 의 예측 블록들을 생성하기 위해 인트라 예측을 사용하는 경우, 비디오 인코더 (20) 는 PU 와 연관된 픽처의 디코딩된 샘플들에 기초하여 PU 의 예측 블록들을 생성할 수도 있다.
- [0061] 비디오 인코더 (20) 가 PU 의 예측 블록들을 생성하기 위해 인터 예측을 사용하는 경우, 비디오 인코더 (20) 는



PU 와 연관된 픽처 이외의 하나 이상의 픽처들의 디코딩된 샘플들에 기초하여 PU 의 예측 블록들을 생성할 수도 있다. 인터 예측은 단방향 인터 예측 (즉, 단-예측) 또는 양방향 인터 예측 (즉, 양-예측) 일 수도 있다.

단-예측 또는 양-예측을 수행하기 위해, 비디오 인코더 (20) 는 현재 슬라이스에 대한 제 1 참조 픽처 리스트 (RefPicList0) 및 제 2 참조 픽처 리스트 (RefPicList1) 를 생성할 수도 있다. 참조 픽처 리스트들 각각은 하나 이상의 참조 픽처들을 포함할 수도 있다. 단-예측을 사용할 때, 비디오 인코더 (20) 는 참조 픽처 내의 참조 로케이션을 결정하기 위해 RefPicList0 및 RefPicList1 중 어느 하나 또는 양쪽에서 참조 픽처들을 탐색할 수도 있다. 또한, 단-예측을 사용할 때, 비디오 인코더 (20) 는, 참조 로케이션에 대응하는 샘플들에 적어도 부분적으로 기초하여, PU 에 대한 예측 샘플 블록들을 생성할 수도 있다. 또한, 단-예측을 사용할 때, 비디오 인코더 (20) 는 PU 의 예측 블록과 참조 로케이션 사이의 공간 변위를 나타내는 단일 모션 벡터를 생성할 수도 있다. PU 의 예측 블록과 참조 로케이션 사이의 공간 변위를 나타내기 위해, 모션 벡터는 PU 의 예측 블록과 참조 로케이션 사이의 수평 변위를 특징하는 수평 컴포넌트를 포함할 수도 있으며, PU 의 예측 블록과 참조 로케이션 사이의 수직 변위를 특징하는 수직 컴포넌트를 포함할 수도 있다.

[0062]

PU 를 인코딩하기 위해 양-예측을 사용할 때, 비디오 인코더 (20) 는 RefPicList0 에서의 참조 픽처에서 제 1 참조 로케이션을, 그리고 RefPicList1 에서의 참조 픽처에서 제 2 참조 로케이션을 결정할 수도 있다. 비디오 인코더 (20) 는 그 후에, 제 1 및 제 2 참조 로케이션들에 대응하는 샘플들에 적어도 부분적으로 기초하여, PU 에 대한 예측 블록들을 생성할 수도 있다. 또한, PU 를 인코딩하기 위해 양-예측을 사용할 때, 비디오 인코더 (20) 는 PU 의 예측 블록과 제 1 참조 로케이션 사이의 공간 변위를 나타내는 제 1 MV, 및 PU 의 예측 블록과 제 2 참조 로케이션 사이의 공간 변위를 나타내는 제 2 MV 를 생성할 수도 있다.

[0063]

비디오 인코더 (20) 가 CU 의 하나 이상의 PU들에 대한 예측 루마, Cb 및 Cr 블록들을 생성한 후에, 비디오 인코더 (20) 는 CU 에 대한 루마 잔차 블록을 생성할 수도 있다. CU 의 루마 잔차 블록에서의 각각의 샘플은 CU 의 예측 루마 블록들 중 하나에서의 루마 샘플과 CU 의 오리지널 루마 코딩 블록에서의 대응하는 샘플 사이의 차이를 나타낸다. 부가적으로, 비디오 인코더 (20) 는 CU 에 대한 Cb 잔차 블록을 생성할 수도 있다.

CU 의 Cb 잔차 블록에서의 각각의 샘플은 CU 의 예측 Cb 블록들 중 하나에서의 Cb 샘플과 CU 의 오리지널 Cb 코딩 블록에서의 대응하는 샘플 사이의 차이를 나타낼 수도 있다. 비디오 인코더 (20) 는 또한 CU 에 대한 Cr 잔차 블록을 생성할 수도 있다. CU 의 Cr 잔차 블록에서의 각각의 샘플은 CU 의 예측 Cr 블록들 중 하나에서의 Cr 샘플과 CU 의 오리지널 Cr 코딩 블록에서의 대응하는 샘플 사이의 차이를 나타낼 수도 있다.

[0064]

또한, 비디오 인코더 (20) 는 쿼드-트리 파티셔닝을 사용하여 CU 의 루마, Cb 및 Cr 잔차 블록들을 하나 이상의 루마, Cb 및 Cr 변환 블록들로 분해할 수도 있다. 변환 블록은 동일한 변환이 적용되는 샘플들의 직사각형 블록일 수도 있다. CU 의 변환 유닛 (TU) 은 루마 샘플들의 변환 블록, 크로마 샘플들의 2개의 대응하는 변환 블록들, 및 변환 블록 샘플들을 변환하는데 사용되는 신택스 구조들일 수도 있다. 따라서, CU 의 각각의 TU 는 루마 변환 블록, Cb 변환 블록, 및 Cr 변환 블록과 연관될 수도 있다. TU 와 연관된 루마 변환 블록은 CU 의 루마 잔차 블록의 서브-블록일 수도 있다. Cb 변환 블록은 CU 의 Cb 잔차 블록의 서브-블록일 수도 있다. Cr 변환 블록은 CU 의 Cr 잔차 블록의 서브-블록일 수도 있다.

[0065]

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

[0066]

계수 블록 (예를 들어, 루마 계수 블록, Cb 계수 블록 또는 Cr 계수 블록) 을 생성한 후에, 비디오 인코더 (20) 는 계수 블록을 양자화할 수도 있다. 양자화는 일반적으로 변환 계수들을 나타내는데 사용되는 데이터의 양을 가능한 한 감소시키기 위해 변환 계수들이 양자화되어, 추가적인 압축을 제공하는 프로세스를 지칭한다. 또한, 비디오 인코더 (20) 는 변환 계수들을 역양자화하고 그 변환 계수들에 역변환을 적용하여 픽처의 CU들의 TU들의 변환 블록들을 재구성할 수도 있다. 비디오 인코더 (20) 는 CU 의 PU들의 예측 블록들 및 CU 의 TU들의 재구성된 변환 블록들을 사용하여 CU 의 코딩 블록들을 재구성할 수도 있다. 픽처의 각각의 CU 의 코딩 블록들을 재구성함으로써, 비디오 인코더 (20) 는 픽처를 재구성할 수도 있다. 비디오 인코더 (20) 는 재구성된 픽처들을 디코딩된 픽처 버퍼 (DPB) 에 저장할 수도 있다. 비디오 인코더 (20) 는 인터 예측 및 인트라 예측을 위해 DPB 에서의 재구성된 픽처들을 사용할 수도 있다.

[0067]

비디오 인코더 (20) 가 계수 블록을 양자화한 후에, 비디오 인코더 (20) 는 양자화된 변환 계수들을 나타내는



신택스 엘리먼트들을 엔트로피 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 양자화된 변환 계수들을 나타내는 신택스 엘리먼트들에 대해 컨텍스트-적응 2진 산술 코딩 (CABAC) 을 수행할 수도 있다. 비디오 인코더 (20) 는 엔트로피 인코딩된 신택스 엘리먼트들을 비트스트림에서 출력할 수도 있다.

[0068]

비디오 인코더 (20) 는 코딩된 픽처들 및 연관된 데이터의 표현을 형성하는 비트들의 시퀀스를 포함하는 비트스트림을 출력할 수도 있다. 비트스트림은 네트워크 추상화 계층 (NAL) 유닛들의 시퀀스를 포함할 수도 있다.

NAL 유닛들 각각은 NAL 유닛 헤더를 포함하며 미가공 바이트 시퀀스 페이로드 (raw byte sequence payload; RBSP) 를 캡슐화한다. NAL 유닛 헤더는 NAL 유닛 타입 코드를 나타내는 신택스 엘리먼트를 포함할 수도 있다. NAL 유닛의 NAL 유닛 헤더에 의해 특정된 NAL 유닛 타입 코드는 NAL 유닛의 타입을 나타낸다. RBSP 는 NAL 유닛 내에 캡슐화된 정수개의 바이트들을 포함하는 신택스 구조일 수도 있다. 일부 경우들에서, RBSP 는 제로 비트들을 포함한다.

[0069]

상이한 타입들의 NAL 유닛들이 상이한 타입들의 RBSP들을 캡슐화할 수도 있다. 예를 들어, NAL 유닛의 제 1 타입은 픽처 파라미터 세트 (PPS) 에 대한 RBSP 를 캡슐화할 수도 있고, NAL 유닛의 제 2 타입은 코딩된 슬라이스에 대한 RBSP 를 캡슐화할 수도 있고, NAL 유닛의 제 3 타입은 보충 강화 정보 (SEI) 에 대한 RBSP 를 캡슐화할 수도 있으며, 기타 등등을 캡슐화할 수도 있다. PPS 는 0개 이상의 전체 코딩된 픽처들에 적용하는 신택스 엘리먼트들을 포함할 수도 있는 신택스 구조이다. (파라미터 세트들 및 SEI 메시지들에 대한 RBSP들과는 대조적으로) 비디오 코딩 데이터에 대한 RBSP들을 캡슐화하는 NAL 유닛들은 비디오 코딩 계층 (VCL) NAL 유닛들이라고 지칭될 수도 있다. 코딩된 슬라이스를 캡슐화하는 NAL 유닛은 코딩된 슬라이스 NAL 유닛이라고 여기에서 지칭될 수도 있다. 코딩된 슬라이스에 대한 RBSP 는 슬라이스 헤더 및 슬라이스 데이터를 포함할 수도 있다.

[0070]

비디오 디코더 (30) 는 비트스트림을 수신할 수도 있다. 부가적으로, 비디오 디코더 (30) 는 비트스트림을 파싱하여 비트스트림으로부터 신택스 엘리먼트들을 디코딩할 수도 있다. 비디오 디코더 (30) 는 비트스트림으로부터 디코딩된 신택스 엘리먼트들에 적어도 부분적으로 기초하여 비디오 데이터의 픽처들을 재구성할 수도 있다. 비디오 데이터를 재구성하기 위한 프로세스는 일반적으로 비디오 인코더 (20) 에 의해 수행된 프로세스와는 상반될 수도 있다. 예를 들어, 비디오 디코더 (30) 는 PU들의 모션 벡터들을 사용하여 현재 CU 의 PU들에 대한 예측 블록을 결정할 수도 있다. 비디오 디코더 (30) 는 PU들의 모션 벡터들 또는 모션 벡터를 사용하여 PU들에 대한 예측 블록들을 생성할 수도 있다.

[0071]

부가적으로, 비디오 디코더 (30) 는 현재 CU 의 TU들과 연관된 계수 블록들을 역양자화할 수도 있다. 비디오 디코더 (30) 는 계수 블록들에 대한 역변환들을 수행하여 현재 CU 의 TU들과 연관된 변환 블록들을 재구성할 수도 있다. 비디오 디코더 (30) 는 현재 CU 의 PU들에 대한 예측 샘플 블록들의 샘플들을 현재 CU 의 TU들의 변환 블록들의 대응하는 샘플들에 부가함으로써 현재 CU 의 코딩 블록들을 재구성할 수도 있다. 픽처의 각각의 CU 에 대해 코딩 블록들을 재구성함으로써, 비디오 디코더 (30) 는 그 픽처를 재구성할 수도 있다. 비디오 디코더 (30) 는 다른 픽처들의 디코딩에 있어서의 사용을 위해 및/또는 출력을 위해 디코딩된 픽처들을 디코딩된 픽처 버퍼에 저장할 수도 있다.

[0072]

비디오 코더 (예를 들어, 비디오 인코더 (20) 또는 비디오 디코더 (30)) 가 픽처의 현재 슬라이스를 코딩하기 시작할 때, 비디오 코더는 제 1 참조 픽처 리스트 (즉, List 0) 를 초기화할 수도 있다. 또한, 현재 슬라이스가 B 슬라이스이면, 비디오 코더는 제 2 참조 픽처 리스트 (즉, List 1) 를 초기화할 수도 있다. 본 개시물은 List 0 을 "RefPicList0" 으로서 지칭할 수도 있고 List 1 을 "RefPicList1" 로서 지칭할 수도 있다. 비디오 코더가 참조 픽처 리스트 (예를 들어, List 0 또는 List 1) 를 초기화한 후에, 비디오 코더는 참조 픽처 리스트에서의 참조 픽처들의 순서를 변경할 수도 있다. 다시 말해서, 비디오 코더는 참조 픽처 리스트 변경 (RPLM) 프로세스를 수행할 수도 있다. 비디오 코더는 참조 픽처들의 순서를, 하나의 특정의 참조 픽처가 참조 픽처 리스트에서 하나보다 많은 위치에 나타날 수도 있는 경우를 포함하여, 임의의 순서로 변경할 수도 있다.

[0073]

일부 경우들에서, 비디오 인코더 (20) 는 병합 모드 또는 어드밴스드 모션 벡터 예측 (advanced motion vector prediction; AMVP) 모드를 사용하여 PU 의 모션 정보를 시그널링할 수도 있다. 다시 말해서, HEVC 에서, 모션 파라미터들의 예측을 위한 2개의 모드들이 존재하며, 하나는 병합 모드이고 다른 하나는 AMVP 이다. PU 의 모션 정보는 PU 의 모션 벡터(들) 및 PU 의 참조 인덱스(들) 를 포함할 수도 있다. 비디오 인코더 (20) 가 병합 모드를 사용하여 현재 PU 의 모션 정보를 시그널링할 때, 비디오 인코더 (20) 는 병합 후보 리스트 (즉, 모션 벡터 예측자 (motion vector predictor; MVP) 후보 리스트) 를 생성한다. 다시 말해서, 비디오



인코더 (20) 는 모션 벡터 예측자 리스트 구성 프로세스를 수행할 수도 있다. 병합 후보 리스트는 병합 후보들 (즉, MVP 후보들) 의 세트를 포함한다. 병합 후보 리스트는 현재 PU 에 공간적으로 또는 시간적으로 이웃하는 PU들의 모션 정보를 나타내는 병합 후보들의 세트를 포함할 수도 있다. 즉, 병합 모드에서, 모션 파라미터들 (예를 들어, 참조 인덱스들, 모션 벡터들 등) 의 후보 리스트는, 후보가 공간 및 시간 이웃 블록들로부터의 것일 수 있는 경우에 구성된다.

[0074]

또한, 병합 모드에서, 비디오 인코더 (20) 는 병합 후보 리스트로부터 병합 후보를 선택할 수도 있고, 선택된 병합 후보에 의해 나타낸 모션 정보를 현재 PU 의 모션 정보로서 사용할 수도 있다. 비디오 인코더 (20) 는 선택된 병합 후보의 병합 후보 리스트에서의 위치를 시그널링할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 후보 리스트 내에 인덱스를 송신함으로써, 선택된 모션 벡터 파라미터들을 시그널링할 수도 있다. 비디오 디코더 (30) 는 비트스트림으로부터 후보 리스트 내의 인덱스 (즉, 병합 후보 인덱스) 를 획득할 수도 있다.

부가적으로, 비디오 디코더 (30) 는 동일한 병합 후보 리스트를 생성할 수도 있고, 선택된 병합 후보의 위치의 표시에 기초하여, 선택된 병합 후보를 결정할 수도 있다. 비디오 디코더 (30) 는 그 후에 선택된 병합 후보의 모션 정보를 사용하여 현재 PU 에 대한 예측 블록들을 생성할 수도 있다. 즉, 비디오 디코더 (30) 는, 후보 리스트 인덱스에 적어도 부분적으로 기초하여, 후보 리스트에서, 선택된 후보를 결정할 수도 있고, 여기서 선택된 후보는 현재 PU 에 대한 모션 벡터를 특정한다. 이러한 방법으로, 디코더 측에서, 일단 인덱스가 디코딩된다면, 인덱스가 가리키는 대응하는 블록의 모든 모션 파라미터들은 현재 PU 에 의해 상속된 것이다.

[0075]

스킵 모드는 병합 모드와 유사하다. 스킵 모드에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 병합 후보 리스트를, 비디오 인코더 (20) 및 비디오 디코더 (30) 가 병합 모드에서 병합 후보 리스트를 사용하는 동일한 방법으로 생성하여 사용한다. 그러나, 비디오 인코더 (20) 가 스킵 모드를 사용하여 현재 PU 의 모션 정보를 시그널링할 때, 비디오 인코더 (20) 는 현재 PU 에 대한 어떠한 잔차 데이터도 시그널링하지 않는다. 이에 따라, 비디오 디코더 (30) 는 병합 후보 리스트에서의 선택된 후보의 모션 정보에 의해 나타낸 참조 블록을 PU 에 대한 예측 블록으로서 사용할 수도 있다.

[0076]

AMVP 모드는 비디오 인코더 (20) 가 후보 리스트를 생성하고 후보들의 리스트로부터 후보를 선택한다는 점에서 병합 모드와 유사하다. 그러나, 비디오 인코더 (20) 가 AMVP 모드를 사용하여 현재 PU 의 모션 정보를 시그널링할 때, 비디오 인코더 (20) 는 또한 후보 리스트에서 선택된 후보의 위치를 시그널링하는 것에 부가적으로 참조 인덱스 및 현재 PU 에 대한 모션 벡터 차이 (MVD) 를 시그널링할 수도 있다. 현재 PU 에 대한 MVD 는 현재 PU 의 모션 벡터와 AMVP 후보 리스트로부터 선택된 AMVP 후보의 모션 벡터 사이의 차이를 나타낼 수도 있다. 단-예측에서, 비디오 인코더 (20) 는 현재 PU 에 대한 하나의 참조 인덱스 및 하나의 MVD 를 시그널링할 수도 있다. 양-예측에서, 비디오 인코더 (20) 는 현재 PU 에 대한 2개의 참조 인덱스들 및 2개의 MVD들을 시그널링할 수도 있다. 이러한 방법으로, 비디오 인코더 (20) 는 후보 리스트 내에 인덱스를 송신함으로써 선택된 모션 벡터들을 시그널링할 수도 있고, 참조 인덱스 값들 및 MVD들을 시그널링할 수도 있다. 다시 말해서, 현재 PU 에 대한 모션 벡터를 나타내는 비트스트림에서의 데이터는 참조 인덱스, 후보 리스트에 대한 인덱스, 및 MVD 를 나타내는 데이터를 포함할 수도 있다.

[0077]

또한, 현재 PU 의 모션 정보가 AMVP 모드를 사용하여 시그널링될 때, 비디오 디코더 (30) 는 비트스트림으로부터 현재 PU 에 대한 MVD 및 후보 리스트 인덱스를 획득할 수도 있다. 비디오 디코더 (30) 는 동일한 AMVP 후보 리스트를 생성할 수도 있고, AMVP 후보 리스트에서 선택된 후보의 위치의 표시에 기초하여, 선택된 후보를 결정할 수도 있다. 비디오 디코더 (30) 는 MVD 를 선택된 후보에 의해 나타낸 모션 벡터에 부가함으로써 현재 PU 의 모션 벡터를 복구할 수도 있다. 즉, 비디오 디코더 (30) 는 그 선택된 후보에 의해 나타낸 모션 벡터 및 MVD 에 적어도 부분적으로 기초하여 현재 PU 의 모션 벡터를 결정할 수도 있다. 비디오 디코더 (30) 는 그 후에 현재 PU 의 복구된 모션 벡터 또는 모션 벡터들을 사용하여 현재 PU 에 대한 예측 블록들을 생성할 수도 있다.

[0078]

현재 PU 와 시간적으로 이웃하는 PU (즉, 현재 PU 와는 상이한 시간 인스턴스에 있는 PU) 의 모션 정보에 기초하고 있는 AMVP 후보 리스트 또는 병합 후보 리스트에서의 후보는 시간 모션 벡터 예측자 (TMVP) 라고 지칭될 수도 있다. TMVP 는 HEVC 의 코딩 효율을 개선시키기 위해 사용되고, 다른 코딩 툴들과는 상이하다. TMVP 를 결정하기 위해, 비디오 코더는 우선, 현재 PU 와 코-로케이팅되는 PU 를 포함하는 참조 픽처를 식별할 수도 있다. 즉, 비디오 코더는 디코딩된 픽처 버퍼에서의, 더욱 구체적으로는 참조 픽처 리스트에서의 프레임의 모션 벡터에 액세스한다. 다시 말해서, 비디오 코더는 코-로케이팅된 픽처를 식별할 수도 있다.

[0079]

인터 예측된 슬라이스에서, TMVP 가 전체 코딩된 비디오 시퀀스에 대해 인에이블될 때 (시퀀스 파라미터 세트에



서의 `sps_temporal_mvp_enable_flag` 가 1 로 설정됨), 비디오 인코더 (20) 는 슬라이스 헤더에서의 `slice_temporal_mvp_enable_flag` 를 시그널링하여 TMVP 가 현재 슬라이스에 대해 인에이블되는지 여부를 나타낼 수도 있다.

[0080] 현재 픽처의 현재 슬라이스가 B 슬라이스 (즉, 양방향으로 인터 예측된 PU들을 포함하도록 허용되는 슬라이스) 일 경우, 비디오 인코더 (20) 는 슬라이스 헤더에서, 코-로케이팅된 픽처가 `RefPicList0` 또는 `RefPicList1` 로부터의 것인지 여부를 나타내는 신택스 엘리먼트 (예컨대, `collocated_from_10_flag`) 를 시그널링할 수도 있다.

비디오 디코더 (30) 가 코-로케이팅된 픽처를 포함하는 참조 픽처 리스트를 식별한 후에, 비디오 디코더 (30) 는 식별된 참조 픽처 리스트에서 픽처 (즉, 코-로케이팅된 픽처) 를 식별하기 위해, 슬라이스 헤더에서 시그널링될 수도 있는 다른 신택스 엘리먼트 (예를 들어, `collocated_ref_idx`) 를 사용할 수도 있다.

[0081] 비디오 코더는 코-로케이팅된 픽처를 체크함으로써 코-로케이팅된 PU 를 식별할 수도 있다. TMVP 는 코-로케이팅된 PU 를 포함하는 CU 의 우측-하부 PU 의 모션 정보, 또는 이 PU 를 포함하는 CU 의 중심 PU들 내의 우측-하부 PU 의 모션 정보 중 어느 하나를 나타낼 수도 있다. 코-로케이팅된 PU 를 포함하는 CU 의 우측-하부 PU 는 PU 의 예측 블록의 하부-우측 샘플의 바로 아래와 우측의 로케이션을 커버하는 PU 일 수도 있다. 다시 말해서, TMVP 는, 참조 픽처에 있으며 현재 PU 의 하부 우측 코너와 코-로케이팅되어 있는 로케이션을 커버하는 PU 의 모션 정보를 나타낼 수도 있고, 또는 TMVP 는, 참조 픽처에 있으며 현재 PU 의 중심과 코-로케이팅되어 있는 로케이션을 커버하는 PU 의 모션 정보를 나타낼 수도 있다.

[0082] 상기 프로세스에 의해 식별된 모션 벡터들이 병합 모드 또는 AMVP 모드에 대한 모션 후보를 생성하기 위해 사용될 때, (POC 값에 의해 반영된) 시간적 로케이션에 기초하여 모션 벡터들이 스케일링될 수도 있다. 예를 들어, 비디오 코더는 현재 픽처와 참조 픽처의 POC 값들 사이의 차이가 더 작을 때보다, 현재 픽처와 참조 픽처의 POC 값들 사이의 차이가 더 클 때에 더 큰 양만큼 모션 벡터의 크기를 증가시킬 수도 있다.

[0083] 3D-HEVC 는 상이한 뷰포인트들로부터의 동일한 장면의 다수의 뷰들을 제공한다. 3D-HEVC 에 대한 표준화 노력들 중 일부는 HEVC 에 기초한 멀티뷰 비디오 코덱의 표준화를 포함한다. 이와 유사하게, HEVC 기반 3DV 에서는, 상이한 뷰들로부터의 재구성된 뷰 컴포넌트들에 기초한 인터-뷰 예측이 가능하게 된다. H.264/AVC 에서의 MVC 와 같이, 3D-HEVC 는 인터-뷰 모션 예측 (IMP) 을 지원한다. 3D-HEVC 에서, IMP 는 표준 HEVC 에서 사용된 모션 보상과 유사하고, 동일한 또는 유사한 신택스 엘리먼트들을 활용할 수도 있다. 그러나, 비디오 코더가 PU 상에서 IMP 를 수행할 때, 비디오 코더는, 참조 픽처로서, PU 와 동일한 액세스 유닛에 있지만 상이한 뷰에 있는 픽처를 사용할 수도 있다. 이에 반해, 종래의 모션 보상은 단지 참조 픽처들로서 상이한 액세스 유닛들에서의 픽처들을 사용한다. 따라서, 3D-HEVC 에서, 의존적 뷰에서의 블록의 모션 파티셔닝들은, 동일한 액세스 유닛의 다른 뷰들에서의 이미 코딩된 모션 파라미터들에 기초하여 예측 또는 추론된다.

[0084] 비디오 코더는 현재 PU 의 모션 정보가 병합 모드 또는 AMVP 모드를 사용하여 시그널링될 때 후보 리스트 (예를 들어, 병합 후보 리스트 또는 AMVP 후보 리스트) 를 생성할 수도 있다. 3D-HEVC 에서, 후보 리스트는 후보 리스트에서의 다른 후보들과 동일한 방식으로 사용될 수도 있는 인터-뷰 예측 후보를 포함할 수도 있다. 인터-뷰 예측 후보는 참조 픽처의 PU (즉, 참조 PU) 의 모션 정보를 특징한다. 참조 픽처는 현재 PU 와 동일한 액세스 유닛에 있을 수도 있지만, 현재 PU 와는 상이한 뷰에 있다. 참조 PU 를 결정하기 위해, 비디오 코더는 현재 PU 에 대한 디스패리티 벡터를 결정하기 위한 디스패리티 벡터 구성 프로세스를 수행할 수도 있다. 현재 PU 에 대한 디스패리티 벡터는 참조 텍스처 픽처 내의 로케이션과 현재 PU 사이의 수평 공간 변위를 나타낼 수도 있다. 참조 PU 는 디스패리티 벡터에 의해 나타난 로케이션을 커버하는 참조 텍스처 픽처의 PU 일 수도 있다.

[0085] 디스패리티 모션 벡터는 인터-뷰 참조 픽처 내의 로케이션을 가리키는 모션 벡터이다. 인터-뷰 참조 픽처는 현재 PU 와 동일한 액세스 유닛에 있지만 상이한 뷰에 있는 텍스처 픽처이다. 공간 디스패리티 벡터 ("SDV") 는 현재 PU 에 공간적으로 이웃하는 PU 의 디스패리티 모션 벡터이다. 다시 말해서, SDV 는, 공간적으로 이웃하는 PU 에 의해 특정되고 인터-뷰 참조 픽처에서의 로케이션을 나타내는 모션 벡터이고, 여기서 공간적으로 이웃하는 PU 는 현재 PU 에 공간적으로 이웃한다. 시간 디스패리티 벡터 ("TDV") 는, 현재 PU 와 동일한 뷰에서 그리고 현재 PU 와는 상이한 액세스 유닛에서, 현재 PU 와 코-로케이팅된 PU 의 디스패리티 모션 벡터이다. 다시 말해서, TDV 는 임의의 참조 픽처 또는 인터-뷰 픽처에서 동일한 액세스 유닛과 코-로케이팅된 PU, 코-로케이팅된 LCU 로부터의 디스패리티 모션 벡터일 수도 있다. 대안적으로, TMVP 에 의해 생성된 모션 벡터 또는 TMVP 에 대해 사용된 픽처로부터의 코-로케이팅된 PU 의 모션 벡터가 디스패리티 벡터인 경우, 이것은 또한 TDV 로서 취급된다. 현재 PU 의 공간적으로 이웃하거나 시간적으로 이웃하는 PU 가 인터-



뷰 모션 예측을 사용하여 코딩되는 경우, 공간적으로 이웃하거나 시간적으로 이웃하는 PU 의 디스패리티 벡터는 목시적 디스패리티 벡터 ("IDV") 이다.

[0086]

비디오 코더가 IMP 에 대해 직접적으로 SDV들, TDV들, 또는 IDV들로부터 선택된 디스패리티 벡터를 사용할 수도 있다. 위에서 나타낸 바와 같이, 비디오 인코더는 병합/스킵 모드 또는 AMVP 모드를 사용하여 현재 PU 의 모션 정보를 시그널링할 때 현재 PU 에 대한 MVP 후보 리스트들을 생성할 수도 있다. 비디오 코더는 인터-뷰 참조 픽처에서 참조 PU 를 결정하기 위해 선택된 디스패리티 벡터 후보에 의해 특정된 디스패리티 벡터를 사용할 수도 있다. 비디오 코더는 그 후에 병합 모드 또는 AMVP 모드에 대한 MV 후보 리스트들에서 인터-뷰 예측 MV 후보로서 참조 PU 의 모션 정보를 포함할 수도 있다.

[0087]

선택스 엘리먼트 `part_mode` 는 현재 CU 의 파티셔닝 모드를 특정한다. HEVC 에서 `part_mode` 의 값은 다음과 같이 제한된다. 하나의 CU 의 코딩 모드가 `MODE_INTRA` 와 동일한 경우, `part_mode` 는 0 또는 1 과 동일해야 한다. 그렇지 않다면, 하나의 CU 의 코딩 모드가 `MODE_INTER` 와 동일한 경우, 다음이 적용된다. 현재 CU 의 사이즈가 최소 CU 의 사이즈보다 더 크고 비대칭 모션 파티션이 가능하게 되는 경우, `part_mode` 는 0 내지 2 의 범위에 그리고 4 내지 7 의 범위에 있어야 한다. 그렇지 않다면, 현재 CU 의 사이즈가 최소 CU 의 사이즈보다 더 크고 비대칭 모션 파티션이 불가능하게 되는 경우, `part_mode` 는 0 내지 2 의 범위에 있어야 한다. 그렇지 않다면, 현재 CU 의 사이즈가 8 과 동일한 경우, `part_mode` 의 값은 0 내지 2 의 범위에 있어야 한다. 그렇지 않다면 (현재 CU 의 사이즈가 8 보다 더 크다), `part_mode` 의 값은 0 내지 3 의 범위에 있어야 한다.

[0088]

`part_mode` 와 변수 `PartMode` 의 연관된 값 사이의 관계가 다음의 테이블 1 에 정의된다. `part_mode` 가 존재하지 않을 때, 현재 CU 의 파티션 모드는 `PART_2Nx2N` 과 동일한 것으로 추론된다는 것에 주목한다.

하나의 CU 의 코딩 모드	<code>part_mode</code>	<code>PartMode</code>
<code>MODE_INTRA</code>	0	<code>PART_2Nx2N</code>
	1	<code>PART_NxN</code>
<code>MODE_INTER</code>	0	<code>PART_2Nx2N</code>
	1	<code>PART_2NxN</code>
	2	<code>PART_Nx2N</code>
	3	<code>PART_NxN</code>
	4	<code>PART_2NxnU</code>
	5	<code>PART_2NxnD</code>
	6	<code>PART_nLx2N</code>
	7	<code>PART_nRx2N</code>

[0089]

테이블 1: 예측 모드 및 파티셔닝 타입에 대한 명칭 연관

[0090]

[0091]

모든 뷰들에 대해 텍스처 우선 코딩 순서를 사용하는 3D-HEVC 에서의 디스패리티 벡터 유도 방법을 위해 NBDV 가 사용된다. 현재 3D-HEVC 설계에서, NBDV 는 또한 참조 뷰의 심도 맵으로부터 심도 데이터를 추출하기 위해 사용된다.

[0092]

2개의 뷰들 사이의 디스패리티의 추정자를 위해 디스패리티 벡터가 사용된다. 이웃 블록들이 비디오 코딩에 있어서 거의 동일한 모션/디스패리티 정보를 공유하기 때문에, 현재 블록은 이웃 블록들에서의 모션 벡터 정보를 양호한 예측자로서 사용할 수 있다. 이러한 생각에 따라, NBDV 는 상이한 뷰들에서 디스패리티 벡터를 추정하기 위해 이웃 디스패리티 정보를 사용한다.

[0093]

몇몇 공간 및 시간 이웃 블록들이 우선 정의된다. 이들 각각은 그 후에 현재 블록과 후보 블록 사이의 상관의 우선순위에 의해 결정된 미리 정의된 순서로 체크된다. 일단 디스패리티 모션 벡터 (즉, 이 모션 벡터가 인터-뷰 참조 픽처를 가리킨다) 가 후보들에서 발견된다면, 디스패리티 모션 벡터는 디스패리티 벡터로 컨버팅된다. 이웃 블록들의 2개 세트들이 활용된다. 하나의 세트는 공간 이웃 블록들로부터의 것이고 다른 세트는 시간 이웃 블록들로부터의 것이다.

[0094]

3D-HEVC 는 L. Zhang 등, "3D-CE5.h: Disparity vector generation results", 문서 JCT3V-A0097 에서 제안된 NBDV 방법을 우선 채용하였다. J. Sung 등, "3D-CE5.h: Simplification of disparity vector derivation



for HEVC-based 3D video coding", 문서 JCT3V-A0126 에서의 단순화된 NBDV 에 목시적 디스패리티 벡터들이 포함되었다. J. Kang 등, "3D-CE5.h related: Improvements for disparity vector derivation", 문서 번호 JCT3V-B0047 에서, NBDV 는 디코딩된 픽처 버퍼에 저장된 목시적 디스패리티 벡터들을 제거함으로써 더욱 단순화되지만, 또한 RAP 픽처 선택으로 코딩 이득이 개선된다.

[0095] 도 2 는 일 예시적인 멀티-뷰 디코딩 순서를 예시하는 개념도이다. 도 2 의 예에서, 각각의 정사각형은 뷰 컴포넌트에 대응한다. 정사각형들의 컬럼들은 액세스 유닛들에 대응한다. 각각의 액세스 유닛은 시간 인스턴스의 모든 뷰들의 코딩된 픽처들을 포함하도록 정의될 수도 있다. 정사각형들의 로우들은 뷰들에 대응한다. 도 2 의 예에서, 액세스 유닛들은 T0...T8 로 라벨링되며, 뷰들은 S0...S8 로 라벨링된다. 액세스 유닛의 각각의 뷰 컴포넌트가 다음 액세스 유닛의 임의의 뷰 컴포넌트 이전에 디코딩되기 때문에, 도 2 의 디코딩 순서는 시간 우선 코딩으로서 지칭될 수도 있다. 액세스 유닛들의 디코딩 순서는 뷰들의 출력 또는 디스플레이 순서와 동일하지 않을 수도 있다.

[0096] 멀티-뷰 코딩은 인터-뷰 예측을 지원한다. 인터-뷰 예측은 H.264/AVC, HEVC, 또는 다른 비디오 코딩 표준들에 사용되는 인터 예측과 유사하며, 유사한 선택스 엘리먼트들을 사용할 수도 있다. 그러나, 비디오 코더가 (매크로블록과 같은) 현재 비디오 유닛에 대한 인터-뷰 예측을 수행할 때, 비디오 코더는 참조 픽처로서, 현재 비디오 유닛과 동일한 액세스 유닛에 있지만 상이한 뷰에 있는 픽처를 사용할 수도 있다. 이에 반해, 종래의 인터 예측은 단지 상이한 액세스 유닛들에서의 픽처들을 참조 픽처들로서 사용한다.

[0097] 멀티-뷰 코딩에서, 뷰는, 비디오 디코더 (예를 들어, 비디오 디코더 (30)) 가 임의의 다른 뷰에서의 픽처들에 대한 참조 없이도 뷰에서의 픽처들을 디코딩할 수 있는 경우, "베이스 뷰" 라고 지칭될 수도 있다. 비-베이스 뷰들 중 하나에서이 픽처를 코딩할 때, 픽처가 상이한 뷰에 있지만 비디오 코더가 현재 코딩하고 있는 픽처와 동일한 시간 인스턴스 (즉, 액세스 유닛) 내에 있는 경우, (비디오 인코더 (20) 또는 비디오 디코더 (30) 와 같은) 비디오 코더는 픽처를 참조 픽처 리스트 (예를 들어, RefPicList0 또는 RefPicList1) 내에 추가할 수도 있다. 다른 인터 예측 참조 픽처들과 같이, 비디오 코더는 참조 픽처 리스트의 임의의 위치에 인터-뷰 예측 참조 픽처를 삽입할 수도 있다.

[0098] 도 3 은 본 개시물에서 설명된 기법들에 따른, 멀티-뷰 코딩을 위한 일 예시적인 예측 구조를 예시하는 개념도이다. 도 3 의 멀티-뷰 예측 구조는 시간 및 인터-뷰 예측을 포함한다. 도 3 의 예에서, 각각의 정사각형은 뷰 컴포넌트에 대응한다. "I" 로 라벨링된 정사각형들은 인트라 예측된 뷰 컴포넌트들이다. "P" 로 라벨링된 정사각형들은 단방향으로 인터 예측된 뷰 컴포넌트들이다. "B" 및 "b" 로 라벨링된 정사각형들은 양방향으로 인터 예측된 뷰 컴포넌트들이다. "b" 로 라벨링된 정사각형들은 "B" 로 라벨링된 정사각형들을 참조 픽처들로서 사용할 수도 있다. 제 1 정사각형으로부터 제 2 정사각형으로 가리키는 화살표는 제 1 정사각형이 제 2 정사각형에 대한 참조 픽처로서 인터 예측에서 이용가능한 것을 나타낸다. 도 3 에서 수직 화살표들로 나타낸 바와 같이, 동일한 액세스 유닛의 상이한 뷰들에서의 뷰 컴포넌트들은 참조 픽처들로서 이용가능할 수도 있다. 동일한 액세스 유닛의 다른 뷰 컴포넌트에 대한 참조 픽처로서의 액세스 유닛의 하나의 뷰 컴포넌트의 사용은 인터-뷰 예측이라고 지칭될 수도 있다. 따라서, 멀티-뷰 비디오 코딩을 위한 통상적인 MVC 예측 (각각의 뷰 내의 인터-픽처 예측과 인터-뷰 예측 양쪽 모두를 포함함) 구조가 도 3 에 도시되고, 여기서 예측들은 화살표들로 나타내며, 지시도달 오브젝트는 예측 참조를 위해 지시출발 오브젝트를 사용한다.

[0099] H.264/AVC 의 MVC 확장에서, 인터-뷰 예측은, H.264/AVC 모션 보상의 선택스를 사용하지만, 상이한 뷰에서의 픽처가 참조 픽처로서 사용되는 것을 허용하는 디스패리티 모션 보상에 의해 지원될 수도 있다. 2개의 뷰들의 코딩은 또한, H.264/AVC 의 MVC 확장에 의해 지원될 수도 있다. H.264/AVC 의 MVC 확장의 이점들 중 하나는, MVC 인코더가 2개보다 많은 뷰들을 3D 비디오 입력으로서 취급할 수도 있고 MVC 디코더가 이러한 멀티뷰 표현을 디코딩할 수도 있다는 점이다. 결과적으로, MVC 디코더를 갖는 임의의 렌더러 (renderer) 는 2개보다 많은 뷰들을 갖는 3D 비디오 콘텐츠들을 예상할 수도 있다.

[0100] H.264/AVC 의 MVC 확장에서, 인터-뷰 예측은 동일한 액세스 유닛에서의 (즉, 동일한 시간 인스턴스를 갖는) 픽처들 중에서 허용된다. 다시 말해서, MVC 에서, 인터-뷰 예측은 뷰들 사이의 상관을 제거하기 위해 동일한 액세스 유닛의 (즉, 동일한 시간 인스턴스를 갖는) 상이한 뷰들로부터 캡처된 픽처들 중에서 수행된다. 비-베이스 뷰들 중 하나에서의 픽처를 코딩할 때, 픽처가 상이한 뷰에 있지만 동일한 시간 인스턴스를 갖는 경우, 픽처는 참조 픽처 리스트 내에 추가될 수도 있다. 다시 말해서, 인터-뷰 예측으로 코딩된 픽처는 다른 비-베이스 뷰들의 인터-뷰 예측을 위해 참조 픽처 리스트 내에 추가될 수도 있다. 인터-뷰 예측 참조 픽처는,



임의의 인터 예측 참조 픽처처럼, 참조 픽처 리스트의 임의의 위치에 놓일 수 있다.

- [0101] 또한, 멀티-뷰 비디오 코딩의 맥락에서는, 2개의 타입들의 모션 벡터들이 존재할 수도 있다. 하나의 타입의 모션 벡터는 시간 참조 픽처들을 가리키는 정상 모션 벡터이고, 대응하는 시간 인터 예측은 모션-보상된 예측(MCP)이다. 다른 타입의 모션 벡터는 상이한 뷰에서의 픽처들(즉, 인터-뷰 참조 픽처들)을 가리키는 디스패리티 모션 벡터이고, 대응하는 인터 예측은 디스패리티-보상된 예측(DCP)이다.
- [0102] 도 4는 본 개시물에서 설명된 기법들에 따른, 이웃 기반 디스패리티 벡터 유도(NBDV)를 위한 현재 예측 유닛(PU)의 공간 이웃 블록들을 예시하는 개념도이다. 현재 NBDV에서, 5개의 공간 이웃 블록들이 현재 PU에 대한 디스패리티 벡터 유도를 위해 사용된다. 5개의 공간 이웃 블록들은, A0, A1, B0, B1, 및 B2로 표기되는, 현재 PU의 하부-좌측, 좌측, 상부-우측, 상부 및 상부-좌측 블록들이다. 5개의 공간 이웃 블록들이 HEVC에서 병합 모드들에 사용된 것들과 동일하다는 것에 주목해야 한다. 따라서, 어떠한 부가적인 메모리 액세스도 요구되지 않는다.
- [0103] 시간 이웃 블록들을 체크하기 위해, 후보 픽처의 구성 프로세스가 우선 수행된다. 현재 뷰로부터 2개까지의 참조 픽처들은 후보 픽처들로서 취급될 수도 있다. 코-로케이팅된 참조 픽처가 후보 픽처 리스트에 우선 삽입되고, 그 후에 나머지 후보 픽처들이 참조 인덱스의 오름차순으로 후속된다. 양쪽 참조 픽처 리스트들에서 동일한 참조 인덱스를 갖는 참조 픽처들이 이용가능할 때, 코-로케이팅된 픽처의 동일한 참조 픽처 리스트에서의 하나는 다른 하나에 선행한다. 후보 픽처 리스트에서의 각각의 후보 픽처에 대해, 시간 이웃 블록들을 유도하기 위해 3개의 후보 영역들이 결정된다.
- [0104] 블록이 인터-뷰 모션 예측으로 코딩될 때, 상이한 뷰에서 대응하는 블록을 선택하기 위해 디스패리티 벡터가 유도될 필요가 있을 수도 있다. 유도된 디스패리티 벡터라고도 또한 지칭되는 명시적 디스패리티 벡터(IDV)는, 인터-뷰 모션 예측에서 유도된 디스패리티 벡터라고 지칭된다. 블록이 모션 예측으로 코딩되더라도, 유도된 디스패리티 벡터는 다음 블록을 코딩할 목적으로 폐기되지 않는다. NBDV 프로세스는 시간 이웃 블록들에서의 디스패리티 모션 벡터들, 공간 이웃 블록들에서의 디스패리티 모션 벡터들, 그리고 그 후에 명시적 디스패리티 벡터를 차례로 체크할 수도 있다. 일단, 디스패리티 벡터가 발견된다면, 프로세스가 종료된다.
- [0105] 이전 프로세스 동안 어떠한 디스패리티 모션 벡터도 또는 IDV도 발견되지 않을 때, NBDV는 이용불가능한 것으로서 마킹된다. 이 경우, NBDV가 이용불가능한 디스패리티 벡터를 리턴한 것으로 간주될 수 있다. 그러나, NBDV가 이용불가능한 결과를 리턴하더라도, 디스패리티 벡터를 요구하는 다른 코딩 프로세스들에 의해 제로 디스패리티 벡터가 사용될 수 있다. 3D-HEVC는 참조 뷰의 심도가 액세스되도록 허용한다. 이 경우, 심도는 또한 역방향 뷰 합성 예측(backward view synthesis prediction; BVSP)을 위해 사용될 디스패리티 모션 벡터 또는 디스패리티 벡터를 세밀화하기 위해 사용될 수 있다.
- [0106] 이용가능한 디스패리티 벡터가 NBDV 프로세스로부터 유도될 때, 이것은 참조 뷰의 심도 맵으로부터 심도 데이터를 추출함으로써 더욱 세밀화된다. 세밀화 프로세스는 2개의 단계들을 포함한다. 첫 번째로, 베이스 뷰와 같은 이전에 코딩된 참조 심도 뷰에서 유도된 디스패리티 벡터를 사용함으로써 대응하는 심도 블록이 로케이팅된다. 대응하는 심도 블록의 사이즈는 현재 PU의 것과 동일하다. 두 번째로, 대응하는 심도 블록의 4개의 코너 픽셀들로부터의 하나의 심도 값이 선택되고, 세밀화된 디스패리티 벡터의 수평 컴포넌트로 컨버팅된다. 디스패리티 벡터의 수직 컴포넌트가 변경되지 않는다.
- [0107] 이용가능한 디스패리티 벡터를 유도하기 위한 상기 프로세스는 또한 NBDV-R(NBDV refinement) 또는 Do-NBDV(depth oriented NBDV)로 불린다. 그러나, NBDV가 이용가능한 디스패리티 벡터를 제공하지 않을 때(따라서, NBDV의 결과가 이용불가능함), 상기 NBDV-R 프로세스는 스킵되고 제로 디스패리티 벡터가 바로 리턴된다.
- [0108] 세밀화된 디스패리티 벡터가 인터-뷰 모션 예측을 위해 사용되는 한편 세밀화되지 않은 디스패리티 벡터가 인터-뷰 잔차 예측을 위해 사용된다는 것에 주목한다. 부가적으로, 세밀화된 디스패리티 벡터는 이것이 역방향 VSP 모드로 코딩되는 경우 하나의 PU의 모션 벡터로서 저장된다.
- [0109] HEVC가 AMVP 및 병합 모드들에 대한 후보 리스트의 개념을 도입하기 때문에, 3D-HEVC에서, 새로운 후보들을 도입함으로써 인터-뷰 모션 예측이 실현된다. AMVP 후보 리스트 생성을 변경시키는 활동들이 있었지만, 그것을 행하는 이익이 작은 것으로 간주되어서 AMVP 모드가 현재 3D-HEVC에서 미변경된 채로 유지되었다. 인터-뷰 모션 예측의 주요한 실현이 병합 모드의 변경들에 의해 달성된다.
- [0110] 3D-HEVC에서, 6개까지의 병합 후보들이 병합 후보 리스트 내에 삽입될 수 있다. HEVC에서 병합 모드의 디코딩 프로세스들에 의해 리스트 내에 삽입된 후보들을 유지하는 동안, 2개의 부가적인 후보들이 병합 후보 리스트



트 내에 부가될 수도 있다. 이들은 참조 블록의 모션 정보에 액세스하는 것으로부터 유도된 후보, 즉, 인터-뷰 후보 및 현재 블록의 디스패리티 벡터로부터 컨버팅된 후보이다.

[0111] NBDV 프로세스로부터 유도된 디스패리티 벡터 및 그의 연관된 참조 뷰 id 에 의해 로케이팅된 참조 블록의 모션 벡터들을 재사용함으로써 인터-뷰 후보가 유도된다. 또한, 병합 후보의 참조 인덱스는 모션 벡터가 참조 뷰를 가리키는 것과 동일한 액세스 유닛 (시간 인스턴스) 내에 있는 픽처를 (현재 픽처의 참조 픽처 리스트에서) 식별하도록 설정된다.

[0112] 비디오 코더는 다른 후보를 생성하기 위해 인터-뷰 참조 픽처를 식별하는 참조 인덱스를 이용하여 디스패리티 벡터를 디스패리티 모션 벡터로 컨버팅한다. 이 후보는 인터-뷰 후보의 가용성에 관계없이 삽입된다. HEVC 에서 병합 프로세스와 유사하게, 비디오 코더는, 도 4 에 도시된 바와 같이,  $A_1$  과  $B_1$  에 의해 표기된 공간 이웃들로부터의 후보들만을 단지 비교함으로써, 부가적인 후보들에 프루닝 (pruning) 을 적용한다.

[0113] JCT3V-C0152 에서 제안된 역방향-워핑 VSP (backward-warping VSP; BVSP) 접근법은 3D-AVC 에서 블록 기반 VSP 와 동일하다. 이들 2개의 기법들 양쪽 모두가 역방향-워핑 및 블록 기반 VSP 를 사용하여 모션 벡터 차이들의 송신을 피하게 하고 더욱 정밀한 모션 벡터들을 사용하게 한다. 그러나, 구현 상세들은 상이한 플랫폼들로 인해 상이하다. BVSP 라는 용어는 또한 3D-HEVC 에서 역방향-워핑 VSP 접근법을 나타내는데 사용된다.

[0114] 3D-HTM 에서, 텍스처 우선 코딩이 일반 테스트 조건들에서 적용된다. 따라서, 하나의 비-베이스 텍스처 뷰를 디코딩할 때 대응하는 비-베이스 심도 뷰가 이용불가능하고 심도 정보가 추정되어 BVSP 를 수행하는데 사용된다.

[0115] 도 5 는 본 개시물에서 설명된 기법들에 따른, 역방향 워핑에 기초하여 블록 기반 뷰 합성 예측 (block-based view synthesis prediction; BVSP) 을 하기 위한 참조 뷰로부터의 심도 블록 유도의 일 예시적인 가시화의 개념도이다. 도 5 는 참조 뷰로부터의 심도 블록이 로케이팅된 후에 BVSP 예측을 위해 사용되는 방법의 3 단계들을 예시한다. 블록에 대한 심도 정보를 추정하기 위해, 이웃 블록들로부터 디스패리티 벡터를 우선 유도하는 것이 제안된다. 그 후에, 유도된 디스패리티 벡터는 참조 뷰로부터 심도 블록을 획득하는데 사용된다. HTM 5.1 테스트 모델은 NBDV (이웃 블록 디스패리티 벡터) 로서 알려진 디스패리티 벡터 예측자를 유도하기 위한 프로세스를 포함한다. (dvx, dvy) 가 NBDV 함수로부터 식별된 디스패리티 벡터를 표기하고 현재 블록 위치가 (blockx, blocky) 로서 표기되는 것으로 한다. 심도 블록은 참조 뷰의 심도 이미지에서 (blockx+dvx, blocky+dvy) 에서 페치될 수도 있다. 페치된 심도 블록은 현재 PU 의 동일한 사이즈를 가질 것이고, 이것은 그 후에 현재 PU 에 대한 역방향 워핑을 행하는데 사용될 수도 있다. BVSP 가 시퀀스에서 인에이블되는 경우, 인터-뷰 모션 예측을 위한 NBDV 프로세스가 변경된다.

[0116] BVSP 가 인에이블될 때 NBDV 프로세스에 대한 하나의 차이는, 시간 이웃 블록들 각각에 대해, 이것이 디스패리티 모션 벡터를 사용하는 경우, 디스패리티 모션 벡터가 디스패리티 벡터로서 리턴되는 것을 포함한다. 디스패리티 벡터는 심도 정보 섹션에 액세스하면서 NBDV 의 세밀화에서 설명된 바와 같이 더욱 세밀화된다.

[0117] 다른 차이는, 공간 이웃 블록들 각각에 대해, 참조 픽처 리스트 0 및 참조 픽처 리스트 1 에 대해 차례로 다음을 적용한다는 것이다: 공간 이웃 블록이 디스패리티 모션 벡터를 사용하는 경우, 디스패리티 모션 벡터는 디스패리티 벡터로서 리턴되고, 이것은 상술된 바와 같이 더욱 세밀화된다. 그렇지 않다면, 공간 이웃 블록이 BVSP 모드를 사용하는 경우, 연관된 모션 벡터가 디스패리티 벡터로서 리턴된다. 디스패리티 벡터는 상술된 것과 유사한 방법으로 더욱 세밀화된다. 그러나, 4개의 코너 픽셀들보다는 오히려 대응하는 심도 블록의 모든 픽셀들로부터 최대 심도 값이 선택되고, 세밀화된 디스패리티 벡터의 수직 컴포넌트가 0 으로 설정된다.

[0118] 또 다른 차이는, 공간 이웃 블록들 각각에 대해, 이것이 IDV 를 사용하는 경우, IDV 는 디스패리티 벡터로서 리턴되고 이것은 상술된 바와 같이 더욱 세밀화되는 것을 포함한다. 추가의 차이는, 어떠한 디스패리티 모션 벡터도 이용가능하지 않은 경우, 세밀화 프로세스가 적용되지 않고, 디스패리티 벡터가 제로 벡터로서 유도되는 것을 포함한다.

[0119] 디스패리티 모션 벡터 유도 프로세스의 경우, BVSP 모드로 코딩된 하나의 PU 내의 각각의 서브-영역 (4x4 블록) 에 대해, 상술된 세밀화된 디스패리티 벡터를 이용하여 참조 심도 뷰에서 대응하는 4x4 심도 블록이 첫 번째로 로케이팅된다. 두 번째로, 대응하는 심도 블록에서 16개의 심도 픽셀들의 최대 값이 선택된다. 세 번째로, 최대 값이 디스패리티 모션 벡터의 수평 컴포넌트로 컨버팅된다. 디스패리티 모션 벡터의 수직 컴포넌



트가 0 으로 설정된다.

- [0120] 도 6 은 본 개시물에서 설명된 기법들에 따른, 어드밴스드 잔차 예측 (advanced residual prediction; ARP) 의 일 예시적인 예측 구조를 예시하는 개념도이다. 현재 3D-HEVC 에서, 2개의 뷰들의 잔차 신호 사이의 상관을 더욱 효율적으로 활용하기 위해, 인터-뷰 잔차 예측은 소위 어드밴스드 잔차 예측 (ARP) 에 의해 실현되었다. ARP 에서, 도 6 에 도시된 바와 같이, 참조 뷰에 대한 잔차 픽처를 유지하고 잔차 픽처에서 참조 블록 내의 잔차를 직접적으로 예측하는 대신에, 디스패리티 벡터로 식별된 참조 블록의 잔차는 온-더-플라이 (on-the-fly) 로 생성된다.
- [0121] 도 6 에 도시된 바와 같이, 비-베이스 뷰에서 현재 블록의 잔차 (40) 를 양호하게 예측하기 위해, 비디오 코더 는 디스패리티 벡터 (44) 를 사용하여 참조 블록 (42) 을 식별할 수도 있다. 참조 블록 (42) 의 재구성된 신호와 예측 신호 (43) 사이의 잔차를 유도하기 위해 참조 블록 (42) 의 모션 보상이 호출된다. ARP 모드 가 호출될 때, 예를 들어, 비-베이스 뷰의 참조 픽처에서의 블록 (41) 으로부터의 모션 보상에 의해 생성된, 비-베이스 뷰의 예측 신호 위에 예측된 잔차가 부가된다. ARP 모드의 이점들 중 하나는 (ARP 에 대한 잔차를 생성할 때) 참조 블록 (42) 에 의해 사용된 모션 벡터 (45) 가 현재 블록 (40) 의 모션 벡터 (46) 와 정렬되어서, 현재 블록의 잔차 신호가 더욱 정밀하게 예측될 수 있다는 것이다. 따라서, 잔차의 에너지는 상당히 감소될 수 있다.
- [0122] 베이스 (예를 들어, 참조) 와 비-베이스 뷰들 사이의 양자화 차이가 더 작은 예측 정확성을 초래할 수도 있기 때문에, 참조 뷰로부터 생성된 잔차에 2개의 가중 팩터들이 적응적으로 적용될 수도 있다: 0.5 및 1.
- [0123] 베이스 (예를 들어, 참조) 뷰에서의 부가적인 모션 보상이 메모리 액세스 및 계산들의 상당한 증가를 요구할 수도 있기 때문에, 코딩 효율의 보다 적은 희생으로 설계를 더욱 실체적으로 만드는 몇몇 방법들이 구현될 수도 있다. 첫 번째로, 특히 비디오 인코더 (20) 와 같은 인코더에서 연산들을 감소시키기 위해 PU 가  $2N \times 2N$  으로 코딩될 때 ARP 모드가 단지 인에이블된다. 두 번째로, ARP 모드로 코딩된 블록들에 대한 메모리 액세스 를 상당히 감소시키기 위해 참조 블록과 현재 블록 양쪽 모두의 모션 보상을 위해 바이리니어 필터들이 사용될 수도 있다. 세 번째로, 캐시 효율을 개선시키기 위해, 모션 벡터들이 비-베이스 뷰에서 상이한 픽처들을 가리킬 수도 있지만, 베이스 뷰에서의 참조 픽처는 고정된다. 이 경우, 현재 블록의 모션 벡터는 픽처 거리에 의해 기초하여 스케일링될 필요가 있을 수도 있다.
- [0124] 도 7 은 본 개시물에서 설명된 기법들에 따른, 현재 심도 블록의 디스패리티 벡터를 유도하는데 사용된 예시적인 참조 샘플들을 예시하는 개념도이다. 도 7 은 현재 PU 에 대응하는 현재 심도 블록 (50) 을 포함한다. 현재 심도 블록 (50) 은 폭 (W) 및 높이 (H) 를 갖는다. 3개의 이웃 참조 샘플들이 현재 심도 블록에 대해 예시된다.
- [0125] 하나의 예에서, 현재 심도 블록 (50) 의 상부 좌측 이웃 샘플 (54) 이 선택된다. 도 7 에 도시된 바와 같이, 현재 심도 블록 (50) 내의 상부 좌측 샘플 (52) 이 (x, y) 의 좌표를 갖는 경우 상부 좌측 이웃 샘플 (54) 은 (x-1, y-1) 의 좌표를 갖는다. 대안적으로, 현재 심도 블록 (50) 의 상부 이웃 로우 또는 좌측 이웃 컬럼에 로케이팅된 임의의 샘플이 사용될 수도 있다. 대안적으로, 현재 심도 블록 (50) 의 공간 이웃 블록들에 로케이팅된 임의의 샘플이 사용될 수도 있다.
- [0126] 대안적으로, 현재 심도 블록 (50) 이 코딩 중일 때, 하나보다 많은 이용가능 이웃 샘플들이 현재 심도 블록 (50) 에 대한 하나 이상의 디스패리티 벡터들을 유도하는데 사용될 수도 있다. 예를 들어, 좌표 (x, y-1) 를 갖는 다른 상부 샘플 (56) 및 좌표 (x-1, y) 를 갖는 다른 좌측 샘플 (58) 이 사용될 수도 있다. 대안적으로, (x-1, y+1) 또는 (x+1, y-1) 의 좌표를 갖는 샘플이 사용될 수도 있다. 대안적으로, 도 7 에 도시된 좌측 이웃 코너 또는 상부 이웃 코너로서, 현재 심도 블록 (50) 의 사이즈가  $W \times H$  로 표기되는 경우, (x-1, y+H-1) 및 (x+W-1, y-1) 의 좌표를 갖는 샘플들이 사용될 수도 있다. 대안적으로, 현재 심도 블록 (50) 의 상부 이웃 로우의 임의의 샘플들이 사용될 수도 있다. 대안적으로, 현재 심도 블록 (50) 의 좌측 이웃 컬럼의 임의의 샘플들이 사용될 수도 있다. 대안적으로, 현재 심도 블록 (50) 의 공간 이웃 블록들의 임의의 샘플들이 사용될 수도 있다.
- [0127] 상술된 CU-레벨 및 PU-레벨 솔루션들에 적용가능하다면, 다른 대안들이 블록-레벨 디스패리티 벡터 유도에 적용될 수도 있다. 하나보다 많은 이웃 심도 샘플들이 디스패리티 벡터를 유도하는데 사용될 때, 하나의 샘플이 다음 수단 중 하나를 통해 선정될 수도 있다. 첫 번째로, 다수의 샘플들 중 어떤 것이 디스패리티 벡터로 컨버팅되도록 선정되는지를 나타내는 인덱스가 블록 (PU 또는 CU) 레벨에서 시그널링될 수도 있다.



- [0128] 두 번째로, 디스패리티 벡터들의 수가 컨버팅될 수도 있다. 그 수는 심도 샘플들의 수와 동일할 수도 있다. 다수의 디스패리티 벡터들은 인터-뷰 예측을 위해 사용된 예측자들을 생성하는데 사용된다. 예를 들어, 다수의 모션 벡터 후보들이 생성되어 병합 후보 리스트 내에 삽입될 수 있다.
- [0129] 세 번째로, 수학 함수, 예를 들어, 중앙, 최대 또는 평균 함수가 다수의 심도 샘플들의 값들에 적용될 수도 있고, 이 함수의 출력 값이 고유 디스패리티 벡터로 컨버팅되도록 사용된다. 예를 들어, CU-레벨 디스패리티 벡터 유도 프로세스에서, 현재 심도 블록 (50) 을 커버하는 CU 내의 상부 좌측 샘플의 좌표들은 (x, y) 로서 표기되고 CU 사이즈는  $2N \times 2N$  이며, 좌표 (x-1, y-1), (x-1, y) 및 (x, y-1) 를 갖는 3개의 샘플들의 중앙 값이 사용된다. 다른 예에서, 좌표 (x-1, y-1), (x-1, y+2N), (x-1, y+2N-1), (x+2N, y-1) 및 (x+2N-1, y-1) 를 갖는 5개의 샘플들의 최대 값이 사용될 수도 있다.
- [0130] 예를 들어, 적어도 하나의 디스패리티 값을 결정함에 있어서, 시스템은 각각의 CU 의 2개 이상의 이웃 샘플들에 적어도 부분적으로 기초하여 2개 이상의 디스패리티 값들을 결정할 수도 있다. 각각의 CU 의 상부 좌측 샘플은 좌표들 (x, y) 을 가지며, 2개 이상의 재구성된 심도 샘플들은 각각의 CU 의 상부 좌측 이웃 샘플, 각각의 CU 의 상부 이웃 로우, 각각의 CU 의 좌측 이웃 컬럼, 각각의 CU 의 특정 이웃 블록에 로케이팅된 임의의 샘플, 좌표들 (x-1, y+1) 을 갖는 샘플, 좌표들 (x+1, y-1) 을 갖는 샘플, 그리고, 각각의 CU 사이즈가  $2N \times 2N$  으로 표기될 때, 좌표들 (x-1, y+2N-1) 을 갖는 샘플 및 좌표들 (x+2N-1, y-1) 을 갖는 샘플 중 적어도 2개 이상으로부터의 것이다.
- [0131] 디스패리티 벡터를 유도하는데 사용된 샘플을 선정하기 위해 수학 함수를 적용하는 다른 예에서, 3개의 샘플들의 중앙 값이 사용된다. 예를 들어, PU-레벨 디스패리티 벡터 유도 프로세스의 경우, 현재 PU 내의 상부 좌측 샘플의 좌표들은 (x, y) 로서 표기되고 PU 사이즈는  $W \times H$  로 표기되며, 좌표 (x-1, y-1), (x-1, y) 및 (x, y-1) 를 갖는 3개의 샘플들의 중앙 값이 사용된다. 다른 예에서, 좌표 (x-1, y-1), (x-1, y+H), (x-1, y+H-1), (x+W, y-1) 및 (x+W-1, y-1) 를 갖는 5개의 샘플들의 최대 값이 사용될 수도 있다.
- [0132] 예를 들어, 적어도 하나의 디스패리티 값을 결정함에 있어서, 시스템은 각각의 PU 의 2개 이상의 이웃 샘플들에 적어도 부분적으로 기초하여 2개 이상의 디스패리티 값들을 결정할 수도 있다. 적어도 하나의 디스패리티 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 벡터를 결정함에 있어서, 시스템은 2개 이상의 디스패리티 값들에 적어도 부분적으로 기초하여 2개 이상의 디스패리티 벡터들을 결정할 수도 있다. PU 의 상부 좌측 샘플은 좌표들 (x, y) 을 가지며, 2개 이상의 재구성된 심도 샘플들은 PU 의 상부 이웃 로우, PU 의 좌측 이웃 컬럼, 각각의 CU 의 특정 이웃 블록에 로케이팅된 임의의 샘플, 좌표들 (x-1, y+1) 을 갖는 샘플, 좌표들 (x+1, y-1) 을 갖는 샘플, 그리고, PU 사이즈가  $W \times H$  로 표기될 때, 좌표들 (x-1, y+H-1) 을 갖는 샘플 및 좌표들 (x+W-1, y-1) 을 갖는 샘플 중 적어도 2개 이상으로부터의 것이다.
- [0133] 상기 예들은 3D-HEVC 의 텍스처 뷰 코딩에 사용된 NBDV 프로세스와 결합될 수도 있다. NBDV 프로세스와 동일한 프로시저가 우선 적용될 수도 있다. 디스패리티 벡터가 NBDV 프로세스로부터 발견되지 않는 경우, 상기 방법들 중 하나가 추가로 적용되어 현재 심도 블록에 대한 디스패리티 벡터를 획득할 수도 있다.
- [0134] 일부 예들에서, 하나 이상의 샘플들은 대안에서 단지 설명된 그룹들로부터 선택될 수도 있다. 다른 예들에서, 하나 이상의 샘플들은 그룹들 중 임의의 것으로부터 선택될 수도 있다.
- [0135] 심도 블록에 대한 디스패리티 벡터는 다수의 방법들로 활용될 수도 있다. 예를 들어, 디스패리티 벡터는 참조 블록을 식별하는데 사용될 수도 있고 참조 블록의 모션 정보는 현재 블록의 현재 모션을 예측하는데 사용된다. 더욱 구체적으로는, 참조 블록의 모션 벡터는 병합 또는 AMVP 후보 리스트 내에 삽입될 새로운 후보를 생성하는데 사용될 수도 있다. 부가적으로, 디스패리티 벡터는 디스패리티 모션 벡터로 컨버팅되는데 사용될 수도 있다. 이러한 컨버팅된 후보는 병합 또는 AMVP 후보 리스트 내에 삽입될 수 있다. 또한, 디스패리티 벡터는, ARP 가 현재 심도 블록에 대해 적용될 수 있게 하는 참조 블록을 식별하는데 사용될 수도 있다.
- [0136] 도 8 은 본 개시물의 기법들을 구현할 수도 있는 일 예시적인 비디오 인코더 (20) 를 예시하는 블록도이다. 도 8 은 설명의 목적들을 위해 제공되며 본 개시물에서 폭넓게 예시되고 설명된 바와 같이 기법들의 제한으로 간주되지 않아야 한다. 설명의 목적들을 위해, 본 개시물은 HEVC 코딩의 맥락에서 비디오 인코더 (20) 를 설명한다. 그러나, 본 개시물의 기법들은 다른 코딩 표준들 또는 방법들에 적용가능할 수도 있다.
- [0137] 도 8 의 예에서, 비디오 인코더 (20) 는 예측 프로세싱 유닛 (100), 잔차 생성 유닛 (102), 변환 프로세싱 유닛 (104), 양자화 유닛 (106), 역양자화 유닛 (108), 역변환 프로세싱 유닛 (110), 재구성 유닛 (112), 필터 유닛 (114), 디코딩된 픽처 버퍼 (116), 및 엔트로피 인코딩 유닛 (118) 을 포함한다. 예측 프로세싱 유닛 (100)



은 인터-예측 프로세싱 유닛 (120) 및 인트라-예측 프로세싱 유닛 (126) 을 포함한다. 인터-예측 프로세싱 유닛 (120) 은 모션 추정 유닛 (122) 및 모션 보상 유닛 (124) 을 포함한다. 다른 예들에서, 비디오 인코더 (20) 는 더 많거나, 더 적거나, 또는 상이한 기능 컴포넌트들을 포함할 수도 있다.

[0138] 비디오 인코더 (20) 는 비디오 데이터를 수신할 수도 있다. 비디오 인코더 (20) 는 비디오 데이터의 픽처의 슬라이스에서 각각의 CTU 를 인코딩할 수도 있다. CTU들 각각은 픽처의 동일-사이즈로 된 루마 코딩 트리 블록 (CTB) 들 및 대응하는 CTB들과 연관될 수도 있다. CTU 를 인코딩하는 것의 일부로서, 예측 프로세싱 유닛 (100) 은 쿼드-트리 파티셔닝을 수행하여, CTU 의 CTB들을 점진적으로 더 작은 블록들로 분할할 수도 있다. 더 작은 블록들은 CU들의 코딩 블록들일 수도 있다. 예를 들어, 예측 프로세싱 유닛 (100) 은 CTU 와 연관된 CTB 를 4개의 동일-사이즈로 된 서브-블록들로 파티셔닝하고, 서브-블록들 중 하나 이상을 4개의 동일-사이즈로 된 서브-서브-블록들로, 그리고 기타 등등으로 파티셔닝할 수도 있다.

[0139] 비디오 인코더 (20) 는 CU들의 인코딩된 표현들 (즉, 코딩된 CU들) 을 생성하기 위해 CTU 의 CU들을 인코딩할 수도 있다. CU 를 인코딩하는 것의 일부로서, 예측 프로세싱 유닛 (100) 은 CU 의 하나 이상의 PU들 중에서 CU 와 연관된 코딩 블록들을 파티셔닝할 수도 있다. 따라서, 각각의 PU 는 루마 예측 블록 및 대응하는 크로마 예측 블록들과 연관될 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 다양한 사이즈들을 갖는 PU들을 지원할 수도 있다. CU 의 사이즈는 CU 의 루마 코딩 블록의 사이즈를 지칭할 수도 있으며, PU 의 사이즈는 PU 의 루마 예측 블록의 사이즈를 지칭할 수도 있다. 특정 CU 의 사이즈가  $2N \times 2N$  이라고 가정하면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 인트라 예측에 대해서는  $2N \times 2N$  또는  $N \times N$  의 PU 사이즈들을, 그리고 인터 예측에 대해서는  $2N \times 2N$ ,  $2N \times N$ ,  $N \times 2N$ ,  $N \times N$ , 또는 이와 유사한 것의 대칭 PU 사이즈들을 지원할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 또한 인터 예측에 대해서  $2N \times nU$ ,  $2N \times nD$ ,  $nL \times 2N$ , 및  $nR \times 2N$  의 PU 사이즈들에 대한 비대칭적인 파티셔닝을 지원할 수도 있다.

[0140] 인터-예측 프로세싱 유닛 (120) 은 CU 의 각각의 PU 에 대해 인터 예측을 수행함으로써 PU 에 대한 예측 데이터를 생성할 수도 있다. PU 에 대한 예측 데이터는 PU 의 예측 블록들 및 PU 에 대한 모션 정보를 포함할 수도 있다. 인터-예측 프로세싱 유닛 (120) 은 PU 가 I 슬라이스, P 슬라이스, 또는 B 슬라이스 내에 있는지에 따라 CU 의 PU 에 대해 상이한 동작들을 수행할 수도 있다. I 슬라이스에서, 모든 PU들은 인트라 예측된다. 따라서, PU 가 I 슬라이스에 있는 경우, 인터-예측 프로세싱 유닛 (120) 은 PU 에 대한 인터 예측을 수행하지 않는다. 따라서, I-모드에서 인코딩된 비디오 블록들의 경우, 동일한 프레임 내의 이전에 인코딩된 이웃 블록들로부터의 공간 예측을 사용하여 예측 블록이 형성된다.

[0141] P 슬라이스에서의 PU들이 인트라 예측되거나 또는 단방향으로 인터 예측될 수도 있다. 예를 들어, PU 가 P 슬라이스 내에 있는 경우, 모션 추정 유닛 (122) 은 PU 에 대한 참조 영역에 대한 참조 픽처들의 리스트 (예를 들어, "RefPicList0") 에서 참조 픽처들을 탐색할 수도 있다. PU 에 대한 참조 영역은, 참조 픽처 내에서, PU 의 예측 블록들에 가장 가깝게 대응하는 샘플 블록들을 포함하는 영역일 수도 있다. 모션 추정 유닛 (122) 은 PU 에 대한 참조 영역을 포함하는 참조 픽처의 RefPicList0 에서의 위치를 나타내는 참조 인덱스를 생성할 수도 있다. 부가적으로, 모션 추정 유닛 (122) 은 PU 의 예측 블록과 참조 영역과 연관된 참조 로케이션 사이의 공간 변위를 나타내는 모션 벡터를 생성할 수도 있다. 예를 들어, 모션 벡터는 현재 디코딩된 픽처에서의 좌표들로부터 참조 픽처에서의 좌표들까지 오프셋을 제공하는 2차원 벡터일 수도 있다. 모션 추정 유닛 (122) 은 참조 인덱스 및 모션 벡터를 PU 의 모션 정보로서 출력할 수도 있다. 모션 보상 유닛 (124) 은 PU 의 모션 벡터에 의해 나타낸 참조 로케이션에서의 실제 또는 내삽된 샘플들에 기초하여 PU 의 예측 블록들을 생성할 수도 있다.

[0142] B 슬라이스에서의 PU들이 인트라 예측되거나, 단방향으로 인터 예측되거나, 또는 양방향으로 인터 예측될 수도 있다. 따라서, PU 가 B 슬라이스 내에 있는 경우, 모션 추정 유닛 (122) 은 PU 에 대해 단-예측 또는 양-예측을 수행할 수도 있다. PU 에 대한 단-예측을 수행하기 위해, 모션 추정 유닛 (122) 은 PU 에 대한 참조 영역에 대한 RefPicList0 또는 제 2 참조 픽처 리스트 ("RefPicList1") 의 참조 픽처들을 탐색할 수도 있다. 모션 추정 유닛 (122) 은, PU 의 모션 정보로서, 참조 영역을 포함하는 참조 픽처의 RefPicList0 또는 RefPicList1 에서의 위치를 나타내는 참조 인덱스, PU 의 샘플 블록과 참조 영역과 연관된 참조 로케이션 사이의 공간 변위를 나타내는 모션 벡터, 및 참조 픽처가 RefPicList0 또는 RefPicList1 에 있는지 여부를 나타내는 하나 이상의 예측 방향 표시자들을 출력할 수도 있다. 모션 보상 유닛 (124) 은 PU 의 모션 벡터에 의해 나타낸 참조 영역에서의 실제 또는 내삽된 샘플들에 적어도 부분적으로 기초하여 PU 의 예측 블록들을 생성할 수도 있다.



- [0143] PU 에 대한 양방향 인터 예측을 수행하기 위해, 모션 추정 유닛 (122) 은 PU 에 대한 참조 영역에 대한 RefPicList0 에서의 참조 픽처들을 탐색할 수도 있으며, 또한 PU 에 대한 다른 참조 영역에 대한 RefPicList1 에서의 참조 픽처들을 탐색할 수도 있다. 모션 추정 유닛 (122) 은 참조 영역들을 포함하는 참조 픽처들의 RefPicList0 및 RefPicList1 에서의 위치들을 나타내는 참조 인덱스들을 생성할 수도 있다. 부가적으로, 모션 추정 유닛 (122) 은 참조 영역들과 연관된 참조 로케이션들과 PU 의 샘플 블록 사이의 공간 변위들을 나타내는 모션 벡터들을 생성할 수도 있다. PU 의 모션 정보는 PU 의 참조 인덱스들 및 MV들을 포함할 수도 있다. 모션 보상 유닛 (124) 은 PU 의 모션 벡터들에 의해 표시되는 참조 영역에서의 실제 또는 내삽된 샘플들에 적어도 부분적으로 기초하여 PU 의 예측 블록들을 생성할 수도 있다.
- [0144] 인트라-예측 프로세싱 유닛 (126) 은 PU 에 대해 인트라 예측을 수행함으로써 PU 에 대한 예측 데이터를 생성할 수도 있다. PU 에 대한 예측 데이터는 PU 에 대한 예측 블록들 및 다양한 신텍스 엘리먼트들을 포함할 수도 있다. 인트라-예측 프로세싱 유닛 (126) 은 I 슬라이스들, P 슬라이스들, 및 B 슬라이스들에서의 PU들에 대해 인트라 예측을 수행할 수도 있다.
- [0145] PU 에 대해 인트라 예측을 수행하기 위해, 인트라-예측 프로세싱 유닛 (126) 은 다수의 인트라 예측 모드들을 사용하여 PU 에 대한 예측 데이터의 다수의 세트들을 생성할 수도 있다. 인트라-예측 프로세싱 유닛 (126) 은 이웃 PU들의 샘플들에 기초하여 PU 에 대한 예측 블록들을 생성할 수도 있다. 이웃 PU들은 PU들, CU들, 및 CTU들에 대해 좌우, 상하 인코딩 순서를 가정하면, PU 의 상부, 상부 우측, 상부 좌측, 또는 좌측일 수도 있다. 인트라-예측 프로세싱 유닛 (126) 은 다양한 수의 인트라 예측 모드들을 사용할 수도 있다. 일부 예들에서, 인트라 예측 모드들의 수는 PU 의 예측 블록들의 사이즈에 의존할 수도 있다.
- [0146] 예측 프로세싱 유닛 (100) 은 PU들에 대해 인트라-예측 프로세싱 유닛 (120) 에 의해 생성된 예측 데이터, 또는 PU들에 대해 인트라-예측 프로세싱 유닛 (126) 에 의해 생성된 예측 데이터 중으로부터, CU 의 PU들에 대한 예측 데이터를 선택할 수도 있다. 일부 예들에서, 예측 프로세싱 유닛 (100) 은 예측 데이터의 세트들의 레이트/왜곡 메트릭들에 기초하여 CU 의 PU들에 대한 예측 데이터들을 선택한다. 선택된 예측 데이터의 예측 블록들은 여기에서 그 선택된 예측 블록들로서 지칭될 수도 있다.
- [0147] 잔차 생성 유닛 (102) 은 CU 의 루마, Cb 및 Cr 코딩 블록들 및 CU 의 PU들의 선택된 예측 루마, Cb 및 Cr 블록들에 기초하여, CU 의 루마, Cb 및 Cr 잔차 블록들을 생성할 수도 있다. 예를 들어, 잔차 생성 유닛 (102) 은 잔차 블록들에서의 각각의 샘플이 CU 의 코딩 블록에서의 샘플과 CU 의 PU 의 대응하는 선택된 예측 블록에서의 대응하는 샘플 사이의 차이와 동일한 값을 갖도록 CU 의 잔차 블록들을 생성할 수도 있다.
- [0148] 본 개시물의 기법들은, 비디오 인코더 (20) 의 잔차 생성 유닛 (102) 과 같은, 비디오 인코더의 엘리먼트들에 의해 수행될 수도 있지만, 다른 예들에서, 도 8 에 묘사되었거나 묘사되지 않은 비디오 인코더 (20) 의 다른 엘리먼트들이 본 개시물의 기법들을 수행할 수도 있다. 이 기법에서, 비디오 데이터의 픽처의 슬라이스의 각각의 코딩 유닛의 각각의 예측 유닛에 대해, 잔차 생성 유닛 (102) 은 적어도 하나의 이웃 샘플의 적어도 하나의 재구성된 심도 샘플의 적어도 하나의 심도 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 값을 결정할 수도 있다. 잔차 생성 유닛 (102) 은 적어도 하나의 디스패리티 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 벡터를 결정할 수도 있고, 여기서 적어도 하나의 디스패리티 벡터는 각각의 코딩 유닛에 대한 것이다. 잔차 생성 유닛 (102) 은 적어도 하나의 디스패리티 벡터에 적어도 부분적으로 기초하여 각각의 코딩 유닛에 대한 코딩 블록의 인코딩된 표현을 생성할 수도 있다.
- [0149] 변환 프로세싱 유닛 (104) 은 쿼드-트리 파티셔닝을 수행하여, CU 와 연관된 잔차 블록들을 CU 의 TU들과 연관된 변환 블록들로 파티셔닝할 수도 있다. 따라서, TU 는 루마 변환 블록 및 2개의 대응하는 크로마 변환 블록들과 연관될 수도 있다. CU 의 TU들의 루마 및 크로마 변환 블록들의 사이즈들 및 위치들은 CU 의 PU들의 예측 블록들의 사이즈들 및 위치들에 기초하거나 또는 기초하지 않을 수도 있다. "잔차 쿼드-트리" (RQT) 로 알려진 쿼드-트리 구조는 영역들 각각과 연관된 노드들을 포함할 수도 있다. CU 의 TU들은 RQT 의 리프 노드들에 대응할 수도 있다.
- [0150] 변환 프로세싱 유닛 (104) 은 하나 이상의 변환들을 TU 의 변환 블록들에 적용함으로써 CU 의 각각의 TU 에 대한 계수 블록들을 생성할 수도 있다. 변환 프로세싱 유닛 (104) 은 다양한 변환들을 TU 와 연관된 변환 블록에 적용할 수도 있다. 예를 들어, 변환 프로세싱 유닛 (104) 은 이산 코사인 변환 (DCT), 방향성 변환, 또는 개념적으로 유사한 변환을 변환 블록에 적용할 수도 있다. 일부 예들에서, 변환 프로세싱 유닛 (104) 은 변환들을 변환 블록에 적용하지 않는다. 이러한 예들에서, 변환 블록은 계수 블록으로서 취급될 수도 있다.



다.

- [0151] 양자화 유닛 (106) 은 계수 블록에서의 변환 계수들을 양자화할 수도 있다. 양자화 프로세스는 그 변환 계수들의 일부 또는 전부와 연관된 비트 심도를 감소시킬 수도 있다. 예를 들어,  $n$ -비트 변환 계수는 양자화 동안  $m$ -비트 변환 계수로 절사될 수도 있으며, 여기서  $n$  은  $m$  보다 더 크다. 양자화 유닛 (106) 은 CU 와 연관된 양자화 파라미터 (QP) 값에 기초하여 CU 의 TU 와 연관된 계수 블록을 양자화할 수도 있다. 비디오 인코더 (20) 는 CU 와 연관된 QP 값을 조정함으로써 CU 와 연관된 계수 블록들에 적용된 양자화의 정도를 조정할 수도 있다. 양자화는 정보의 손실을 도입할 수도 있어서, 양자화된 변환 계수들은 오리지널 정밀도들보다 더 낮은 정밀도를 가질 수도 있다.
- [0152] 역양자화 유닛 (108) 및 역변환 프로세싱 유닛 (110) 은 역양자화 및 역변환들을 계수 블록에 각각 적용하여, 계수 블록으로부터 잔차 블록을 재구성할 수도 있다. 재구성 유닛 (112) 은 그 재구성된 잔차 블록을 예측 프로세싱 유닛 (100) 에 의해 생성된 하나 이상의 예측 블록들로부터의 대응하는 샘플들에 부가하여, TU 와 연관된 재구성된 변환 블록을 생성할 수도 있다. 이 방법으로 CU 의 각각의 TU 에 대한 변환 블록들을 재구성함으로써, 비디오 인코더 (20) 는 CU 의 코딩 블록들을 재구성할 수도 있다.
- [0153] 필터 유닛 (114) 은 CU 와 연관된 코딩 블록들에서 블록킹 아티팩트들을 감소시키기 위해 하나 이상의 디블록킹 동작들을 수행할 수도 있다. 디코딩된 픽처 버퍼 (116) 는 필터 유닛 (114) 이 재구성된 코딩 블록들에 대해 하나 이상의 디블록킹 동작들을 수행한 후에 그 재구성된 코딩 블록들을 저장할 수도 있다. 인터-예측 유닛 (120) 은 그 재구성된 코딩 블록들을 포함하는 참조 픽처를 사용하여 다른 픽처들의 PU들에 대한 인터 예측을 수행할 수도 있다. 부가적으로, 인트라-예측 프로세싱 유닛 (126) 은 디코딩된 픽처 버퍼 (116) 에서의 재구성된 코딩 블록들을 사용하여 CU 와 동일한 픽처에서의 다른 PU들에 대한 인트라 예측을 수행할 수도 있다.
- [0154] 엔트로피 인코딩 유닛 (118) 은 비디오 인코더 (20) 의 다른 기능 컴포넌트들로부터 데이터를 수신할 수도 있다. 예를 들어, 엔트로피 인코딩 유닛 (118) 은 양자화 유닛 (106) 으로부터 계수 블록들을 수신할 수도 있으며 예측 프로세싱 유닛 (100) 으로부터 선택스 엘리먼트들을 수신할 수도 있다. 엔트로피 인코딩 유닛 (118) 은 데이터에 대해 하나 이상의 엔트로피 인코딩 동작들을 수행하여 엔트로피-인코딩된 데이터를 생성할 수도 있다. 예를 들어, 엔트로피 인코딩 유닛 (118) 은 콘텍스트-적용 가변 길이 코딩 (CAVLC) 동작, CABAC 동작, 변수-대-변수 (V2V) 길이 코딩 동작, 선택스-기반 콘텍스트-적용 2진 산술 코딩 (SBAC) 동작, 확률 간격 파티셔닝 엔트로피 (PIPE) 코딩 동작, 지수-골롬 인코딩 동작, 또는 다른 타입의 엔트로피 인코딩 동작을 데이터에 대해 수행할 수도 있다. 비디오 인코더 (20) 는 엔트로피 인코딩 유닛 (118) 에 의해 생성된 엔트로피-인코딩된 데이터를 포함하는 비트스트림을 출력할 수도 있다. 예를 들어, 비트스트림은 CU 에 대한 RQT 를 나타내는 데이터를 포함할 수도 있다. 비트스트림은 또한 엔트로피 인코딩되지 않은 선택스 엘리먼트들을 포함할 수도 있다.
- [0155] 도 9 는 본 개시물에서 설명된 기법들을 구현할 수도 있는 일 예시적인 비디오 디코더 (30) 를 예시하는 블록도이다. 도 9 는 설명의 목적들을 위해 제공되며 본 개시물에서 폭넓게 예시되고 설명된 바와 같이 기법들에 대한 제한이 아니다. 설명의 목적들을 위해, 본 개시물은 HEVC 코딩의 맥락에서 비디오 디코더 (30) 를 설명한다. 그러나, 본 개시물의 기법들은 다른 코딩 표준들 또는 방법들에 적용가능할 수도 있다.
- [0156] 도 9 의 예에서, 비디오 디코더 (30) 는 엔트로피 디코딩 유닛 (150), 예측 프로세싱 유닛 (152), 역양자화 유닛 (154), 역변환 프로세싱 유닛 (156), 재구성 유닛 (158), 필터 유닛 (160), 및 디코딩 픽처 버퍼 (162) 를 포함한다. 예측 프로세싱 유닛 (152) 은 모션 보상 유닛 (164) 및 인트라-예측 프로세싱 유닛 (166) 을 포함한다. 다른 예들에서, 비디오 디코더 (30) 는 더 많거나, 더 적거나, 또는 상이한 기능 컴포넌트들을 포함할 수도 있다.
- [0157] 엔트로피 디코딩 유닛 (150) 은 NAL 유닛들을 수신하고 NAL 유닛들을 파싱하여 선택스 엘리먼트들을 디코딩할 수도 있다. 엔트로피 디코딩 유닛 (150) 은 NAL 유닛들에서의 엔트로피-인코딩된 선택스 엘리먼트들을 엔트로피 디코딩할 수도 있다. 예측 프로세싱 유닛 (152), 역양자화 유닛 (154), 역변환 프로세싱 유닛 (156), 재구성 유닛 (158), 및 필터 유닛 (160) 은 비트스트림으로부터 추출된 선택스 엘리먼트들에 기초하여 디코딩된 비디오 데이터를 생성할 수도 있다.
- [0158] 비트스트림의 NAL 유닛들은 코딩된 슬라이스 NAL 유닛들을 포함할 수도 있다. 비트스트림을 디코딩하는 것의 일부로서, 엔트로피 디코딩 유닛 (150) 은 코딩된 슬라이스 NAL 유닛들로부터 선택스 엘리먼트들을 추출하여



엔트로피 디코딩할 수도 있다. 코딩된 슬라이스들 각각은 슬라이스 헤더 및 슬라이스 테이터를 포함할 수도 있다. 슬라이스 헤더는 슬라이스에 관련된 선택스 엘리먼트들을 포함할 수도 있다. 슬라이스 헤더에서의 선택스 엘리먼트들은 슬라이스를 포함하는 픽처와 연관된 PPS 를 식별하는 선택스 엘리먼트를 포함할 수도 있다.

[0159] 비트스트림으로부터 선택스 엘리먼트들을 디코딩하는 것에 부가적으로, 비디오 디코더 (30) 는 CU들에 대한 재구성 동작들을 수행할 수도 있다. CU 에 대한 재구성 동작을 수행하기 위해, 비디오 디코더 (30) 는 CU 의 각각의 TU 에 대한 재구성 동작을 수행할 수도 있다. CU 의 각각의 TU 에 대한 재구성 동작을 수행함으로써, 비디오 디코더 (30) 는 CU 의 잔차 블록들을 재구성할 수도 있다.

[0160] CU 의 TU 에 대한 재구성 동작을 수행하는 것의 일부로서, 역양자화 유닛 (154) 은 TU 와 연관되는 계수 블록들을 역양자화할 수도 있다, 즉, 양자화해제할 수도 있다. 역양자화 유닛 (154) 은 TU 의 CU 와 연관된 QP 값을 사용하여 양자화의 정도 그리고, 이와 마찬가지로, 역양자화 유닛 (154) 이 적용할 역양자화의 정도를 결정할 수도 있다.

[0161] 역양자화 유닛 (154) 이 계수 블록을 역양자화한 후에, 역변환 프로세싱 유닛 (156) 은 TU 와 연관된 잔차 블록을 생성하기 위해 하나 이상의 역변환들을 계수 블록에 적용할 수도 있다. 예를 들어, 역변환 프로세싱 유닛 (156) 은 역 DCT, 역 정수 변환, 역 카르넨-루베 (Karhunen-Loeve) 변환 (KLT), 역 회전 변환, 역 방향성 변환, 또는 다른 역변환을 계수 블록에 적용할 수도 있다.

[0162] PU 가 인트라 예측을 사용하여 인코딩되는 경우, 인트라-예측 프로세싱 유닛 (166) 은 인트라 예측을 수행하여 PU 에 대한 예측 블록들을 생성할 수도 있다. 인트라-예측 프로세싱 유닛 (166) 은 공간적으로 이웃하는 PU 들의 예측 블록들에 기초하여 PU 에 대한 예측 루마, Cb 및 Cr 블록들을 생성하기 위해 인트라 예측 모드를 사용할 수도 있다. 인트라-예측 프로세싱 유닛 (166) 은 비트스트림으로부터 디코딩된 하나 이상의 선택스 엘리먼트들에 기초하여 PU 에 대한 인트라 예측 모드를 결정할 수도 있다.

[0163] 예측 프로세싱 유닛 (152) 은 비트스트림으로부터 추출된 선택스 엘리먼트들에 기초하여 제 1 참조 픽처 리스트 (RefPicList0) 및 제 2 참조 픽처 리스트 (RefPicList1) 를 구성할 수도 있다. 또한, PU 가 인터 예측을 사용하여 인코딩되는 경우, 엔트로피 디코딩 유닛 (150) 은 PU 에 대한 모션 정보를 추출할 수도 있다. 모션 보상 유닛 (164) 은, PU 의 모션 정보에 기초하여, PU 에 대한 하나 이상의 참조 영역들을 결정할 수도 있다. 모션 보상 유닛 (164) 은, PU 에 대한 하나 이상의 참조 블록들에서의 샘플 블록들에 기초하여, PU 에 대한 예측 루마, Cb 및 Cr 블록들을 생성할 수도 있다.

[0164] 재구성 유닛 (158) 은 CU 의 루마, Cb 및 Cr 코딩 블록들을 재구성하기 위해, 적용가능한 경우, CU 의 TU들과 연관된 루마, Cb 및 Cr 변환 블록들 및 CU 의 PU들의 예측 루마, Cb 및 Cr 블록들, 즉, 인트라-예측 데이터 또는 인터-예측 데이터를 사용할 수도 있다. 예를 들어, 재구성 유닛 (158) 은 루마, Cb 및 Cr 변환 블록들의 샘플들을 예측 루마, Cb 및 Cr 블록들의 대응하는 샘플들에 부가하여 CU 의 루마, Cb 및 Cr 코딩 블록들을 재구성할 수도 있다.

[0165] 본 개시물의 기법들은, 비디오 디코더 (30) 의 재구성 유닛 (158) 에 의해 수행될 수도 있지만, 다른 예들에서, 도 9 에 묘사되었거나 묘사되지 않은 비디오 디코더 (30) 의 다른 엘리먼트들이 본 개시물의 기법들을 수행할 수도 있다. 비디오 데이터의 픽처의 슬라이스의 각각의 코딩 유닛의 각각의 예측 유닛에 대해, 재구성 유닛 (158) 은 적어도 하나의 이웃 샘플의 적어도 하나의 재구성된 심도 샘플의 적어도 하나의 심도 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 값을 결정할 수도 있다. 재구성 유닛 (158) 은 적어도 하나의 디스패리티 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 벡터를 결정할 수도 있고, 여기서 적어도 하나의 디스패리티 벡터는 각각의 코딩 유닛에 대한 것이다. 재구성 유닛 (158) 은 적어도 하나의 디스패리티 벡터에 적어도 부분적으로 기초하여 각각의 코딩 유닛에 대한 코딩 블록을 재구성할 수도 있다.

[0166] 필터 유닛 (160) 은 CU 의 루마, Cb 및 Cr 코딩 블록들과 연관된 블록킹 아티팩트들을 감소시키기 위해 디블록킹 동작을 수행할 수도 있다. 비디오 디코더 (30) 는 CU 의 루마, Cb 및 Cr 코딩 블록들을 디코딩된 픽처 버퍼 (162) 에 저장할 수도 있다. 디코딩된 픽처 버퍼 (162) 는 후속 모션 보상, 인트라 예측, 및 도 1 의 디스플레이 디바이스 (32) 와 같은 디스플레이 디바이스 상에의 프리젠테이션을 위해 참조 픽처들을 제공할 수도 있다. 예를 들어, 비디오 디코더 (30) 는, 디코딩된 픽처 버퍼 (162) 에서의 루마, Cb 및 Cr 블록들에 기초하여, 다른 CU들의 PU들에 대한 인트라 예측 또는 인터 예측 동작들을 수행할 수도 있다. 이러한 방법으로, 비디오 디코더 (30) 는, 비트스트림으로부터, 루마 계수 블록의 변환 계수 레벨들을 파싱하고, 변환 계수



레벨들을 역양자화하고, 변환을 변환 계수 레벨들에 적용하여 변환 블록을 생성하고, 그 변환 블록에 적어도 부분적으로 기초하여, 코딩 블록을 생성하고, 그리고 디스플레이를 위해 코딩 블록을 출력할 수도 있다.

[0167]

도 10 은 본 개시물에서 하나 이상의 양태들에 따른 일 예시적인 디코딩 기법을 예시하는 흐름도이다. 이 기법은, 비디오 디코더 (30) 의 재구성 유닛 (158) 과 같은, 비디오 디코더의 엘리먼트들에 의해 수행될 수도 있다. 이 기법으로, 비디오 데이터의 픽처의 슬라이스의 각각의 코딩 유닛의 각각의 예측 유닛에 대해, 재구성 유닛 (158) 은 적어도 하나의 이웃 샘플의 적어도 하나의 재구성된 심도 샘플의 적어도 하나의 심도 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 값을 결정할 수도 있다 (200). 재구성 유닛 (158) 은 적어도 하나의 디스패리티 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 벡터를 결정할 수도 있고, 여기서 적어도 하나의 디스패리티 벡터는 각각의 코딩 유닛에 대한 것이다 (202). 재구성 유닛 (158) 은 적어도 하나의 디스패리티 벡터에 적어도 부분적으로 기초하여 각각의 코딩 유닛에 대한 코딩 블록을 재구성할 수도 있다 (204).

[0168]

도 11 은 본 개시물에서 하나 이상의 양태들에 따른 일 예시적인 인코딩 기법을 예시하는 흐름도이다. 이 기법은, 비디오 인코더 (20) 의 잔차 생성 유닛 (102) 과 같은, 비디오 인코더의 엘리먼트들에 의해 수행될 수도 있다. 이 기법으로, 비디오 데이터의 픽처의 슬라이스의 각각의 코딩 유닛의 각각의 예측 유닛에 대해, 잔차 생성 유닛 (102) 은 적어도 하나의 이웃 샘플의 적어도 하나의 재구성된 심도 샘플의 적어도 하나의 심도 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 값을 결정할 수도 있다 (210). 잔차 생성 유닛 (102) 은 적어도 하나의 디스패리티 값에 적어도 부분적으로 기초하여 적어도 하나의 디스패리티 벡터를 결정할 수도 있고, 여기서 적어도 하나의 디스패리티 벡터는 각각의 코딩 유닛에 대한 것이다 (212). 잔차 생성 유닛 (102) 은 적어도 하나의 디스패리티 벡터에 적어도 부분적으로 기초하여 각각의 코딩 유닛에 대한 코딩 블록의 인코딩된 표현을 생성할 수도 있다 (214).

[0169]

하나 이상의 예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현되는 경우, 그 기능들은, 하나 이상의 명령들 또는 코드로서, 컴퓨터 판독가능 매체 상에 저장되거나 또는 컴퓨터 판독가능 매체를 통해 송신될 수도 있으며, 하드웨어 기반 프로세싱 유닛에 의해 실행될 수도 있다. 컴퓨터 판독가능 매체는 컴퓨터 판독가능 저장 매체를 포함할 수도 있으며, 이 컴퓨터 판독가능 저장 매체는 데이터 저장 매체와 같은 유형의 매체, 또는, 예를 들어, 통신 프로토콜에 따라 한 장소로부터 다른 장소로의 컴퓨터 프로그램의 전송을 용이하게 하는 임의의 매체를 포함한 통신 매체에 대응한다. 이러한 방법으로, 컴퓨터 판독가능 매체는 일반적으로 (1) 비일시적인 유형의 컴퓨터 판독가능 저장 매체 또는 (2) 신호 또는 캐리어 파와 같은 통신 매체에 대응할 수도 있다. 데이터 저장 매체는 본 개시물에서 설명된 기법들의 구현을 위한 명령들, 코드 및/또는 데이터 구조들을 추출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 가용 매체일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터 판독가능 매체를 포함할 수도 있다.

[0170]

제한이 아닌 예로서, 이러한 컴퓨터 판독가능 저장 매체는 RAM, ROM, EEPROM, CD-ROM 또는 다른 광 디스크 스토리지, 자기 디스크 스토리지, 또는 다른 자기 스토리지 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 맥락이 컴퓨터 판독가능 매체로 적절히 지칭된다. 예를 들어, 동축 케이블, 광섬유 케이블, 연선, 디지털 가입자 회선 (DSL), 또는 무선 기술들, 예컨대, 적외선, 라디오, 및 마이크로파를 사용하여 명령들이 웹사이트, 서버, 또는 다른 원격 소스로부터 송신되는 경우, 동축 케이블, 광섬유 케이블, 연선, DSL, 또는 무선 기술들, 예컨대, 적외선, 라디오, 및 마이크로파가 그 매체의 정의에 포함된다. 그러나, 컴퓨터 판독가능 저장 매체 및 데이터 저장 매체는 커넥션들, 반송파들, 신호들, 또는 다른 순시적 매체를 포함하지 않고, 그 대신에 비순시적 유형의 저장 매체에 관련된다는 것으로 이해되어야 한다. 디스크 (disk) 및 디스크 (disc) 는, 여기에 사용되는 바와 같이, 콤팩트 디스크 (CD), 레이저 디스크, 광 디스크, 디지털 다기능 디스크 (DVD), 플로피 디스크 및 블루레이 디스크를 포함하며, 디스크들 (disks) 은 데이터를 자기적으로 보통 재생하지만, 디스크들 (discs) 은 레이저로 데이터를 광학적으로 재생한다. 상기한 것의 조합들이 또한 컴퓨터 판독가능 매체의 범위 내에 포함되어야 한다.

[0171]

명령들은 하나 이상의 디지털 신호 프로세서들 (DSPs), 범용 마이크로프로세서들, 주문형 집적 회로들 (ASICs), 필드 프로그래밍가능 로직 어레이들 (FPGAs), 또는 다른 등가의 집적 또는 이산 로직 회로부와 같은 하나 이상의 프로세서들에 의해 실행될 수도 있다. 이에 따라, 용어 "프로세서" 는, 여기에 사용되는 바와 같이 전술한 구조 중 임의의 구조 또는 여기에서 설명된 기법들의 구현에 적합한 임의의 다른 구조를 지칭할 수도 있다. 부가적으로, 일부 양태들에서, 여기에 설명된 기능성은 전용 하드웨어 및/또는 인코딩 및 디코딩을 위해 구



성되는 소프트웨어 모듈들 내에 제공되거나, 또는 결합된 코덱에 포함될 수도 있다. 또한, 이 기법들은 하나 이상의 회로들 또는 로직 엘리먼트들로 완전히 구현될 수 있다.

[0172]

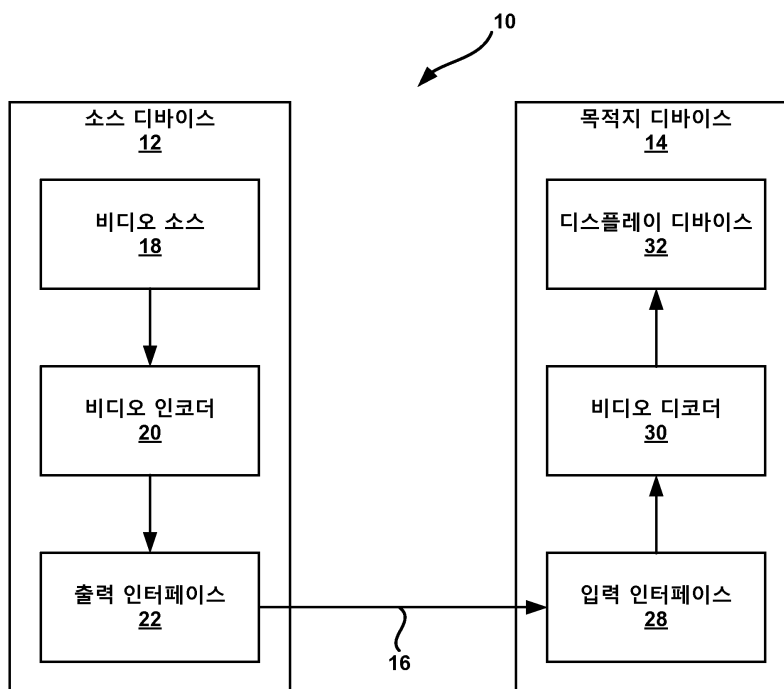
본 개시물의 기법들은 무선 핸드셋, 집적 회로 (IC) 또는 IC들의 세트 (예를 들어, 칩 세트) 를 포함한, 매우 다양한 디바이스들 또는 장치들로 구현될 수도 있다. 개시된 기법들을 수행하도록 구성된 디바이스들의 기능적 양태들을 강조하기 위해 다양한 컴포넌트들, 모듈들, 또는 유닛들이 본 개시물에서 설명되지만, 상이한 하드웨어 유닛들에 의한 실현을 반드시 필요로 하지는 않는다. 오히려, 상술된 바와 같이, 다양한 유닛들이 코덱 하드웨어 유닛에 결합되거나 또는 적합한 소프트웨어 및/또는 펌웨어와 함께 상술된 바와 같은 하나 이상의 프로세서들을 포함한, 상호작용하는 하드웨어 유닛들의 컬렉션으로 제공될 수도 있다.

[0173]

본 개시물의 다양한 예들이 설명되었다. 이들 및 다른 예들은 다음 청구항들의 범위 내에 있다.

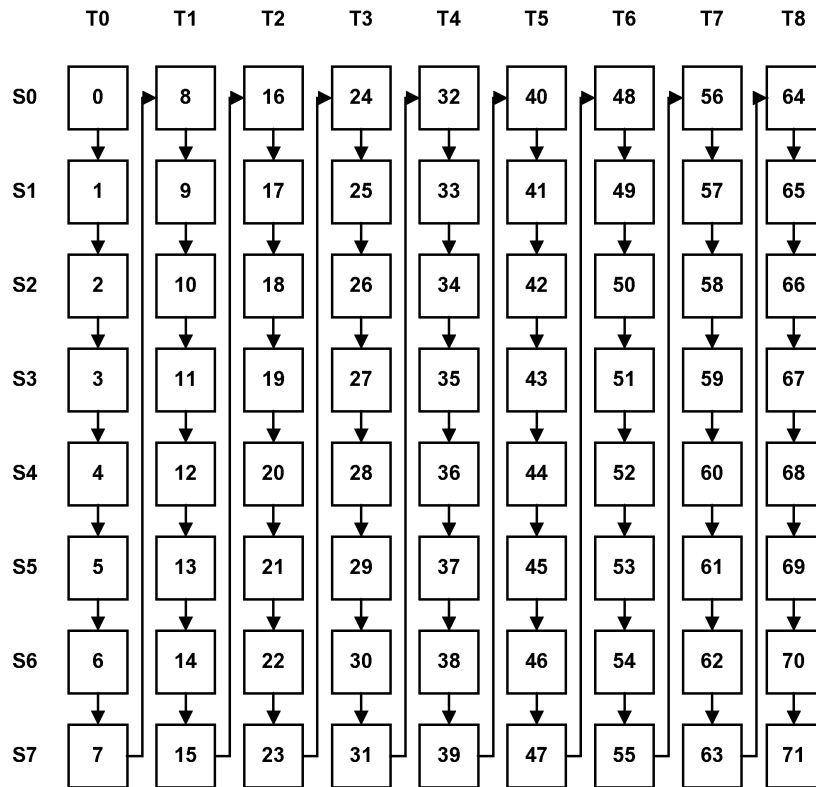
## 도면

### 도면1



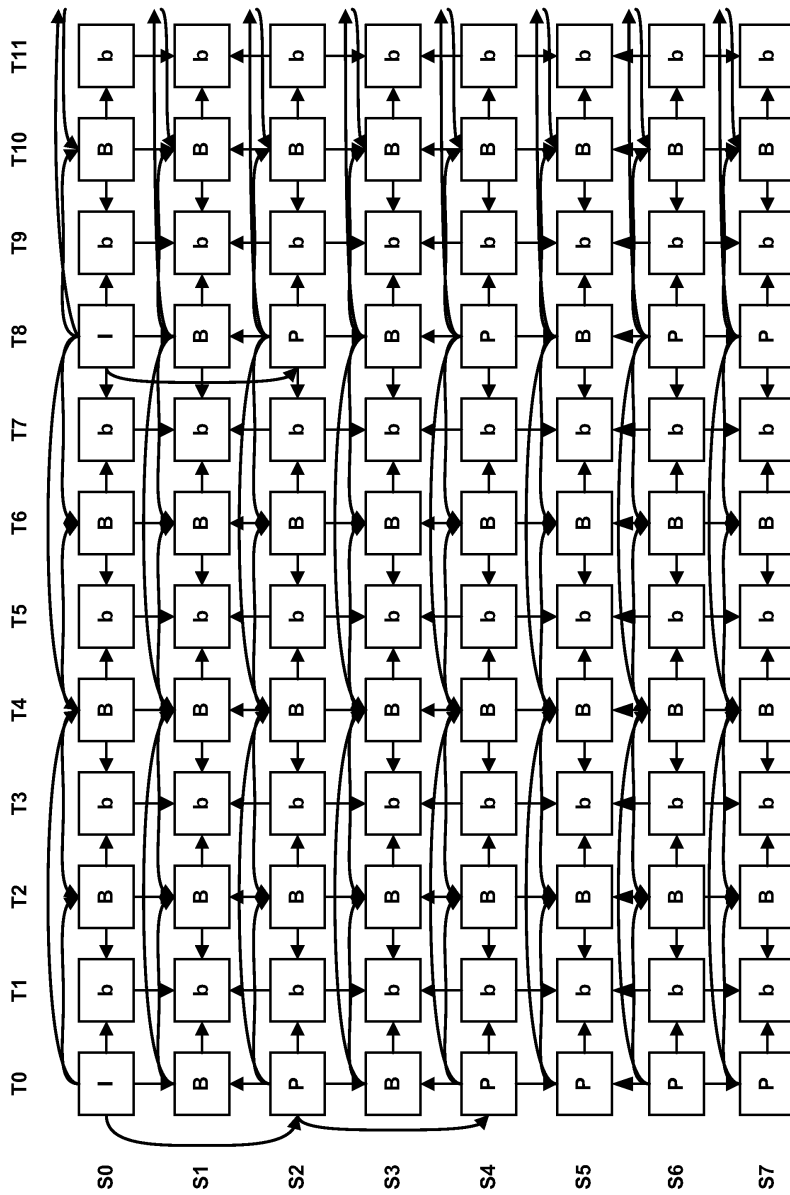


도면2



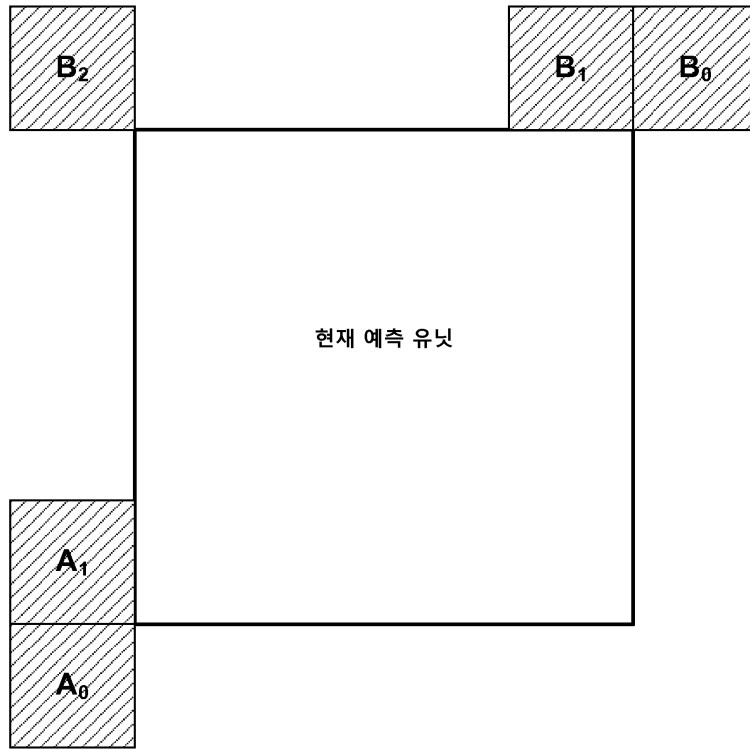


도면3



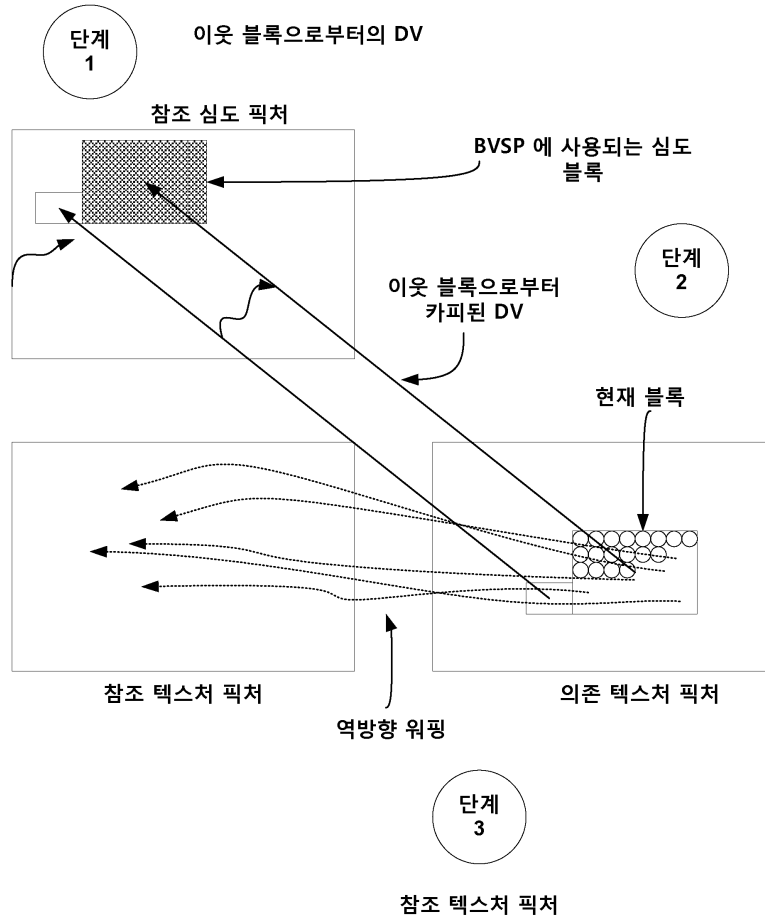


도면4

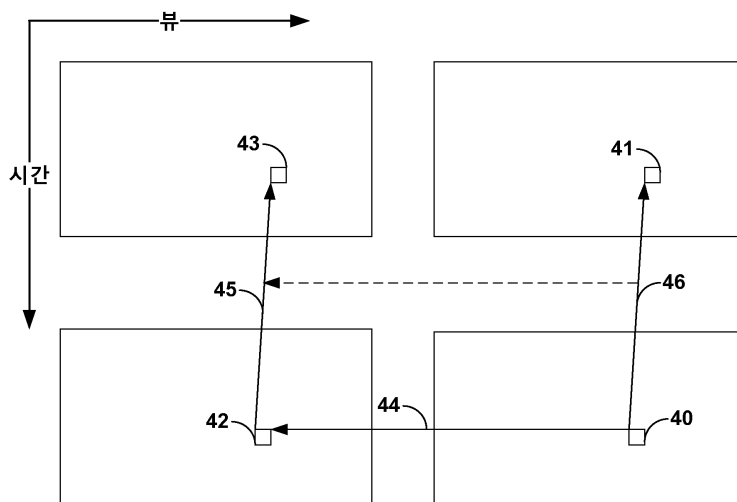




도면5

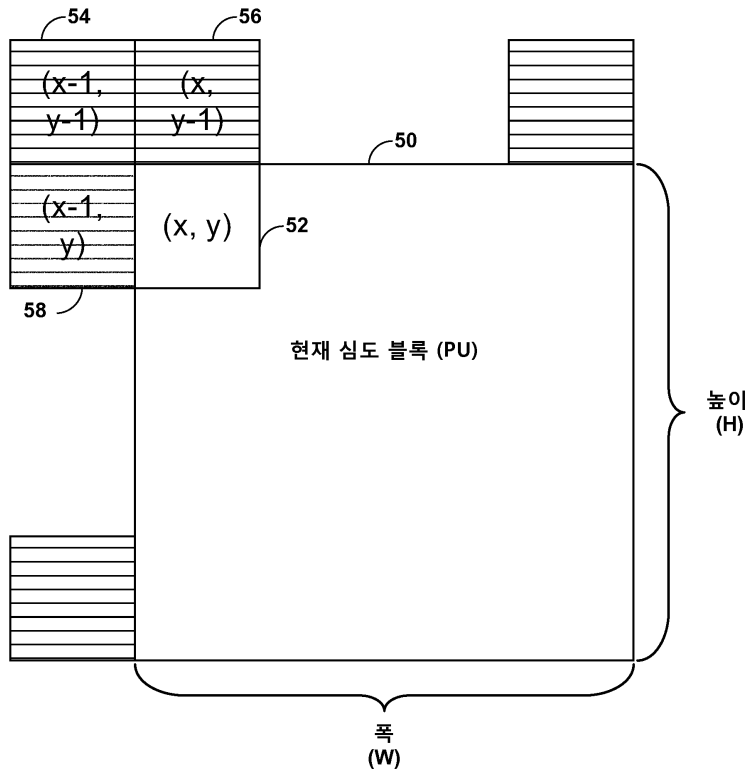


도면6



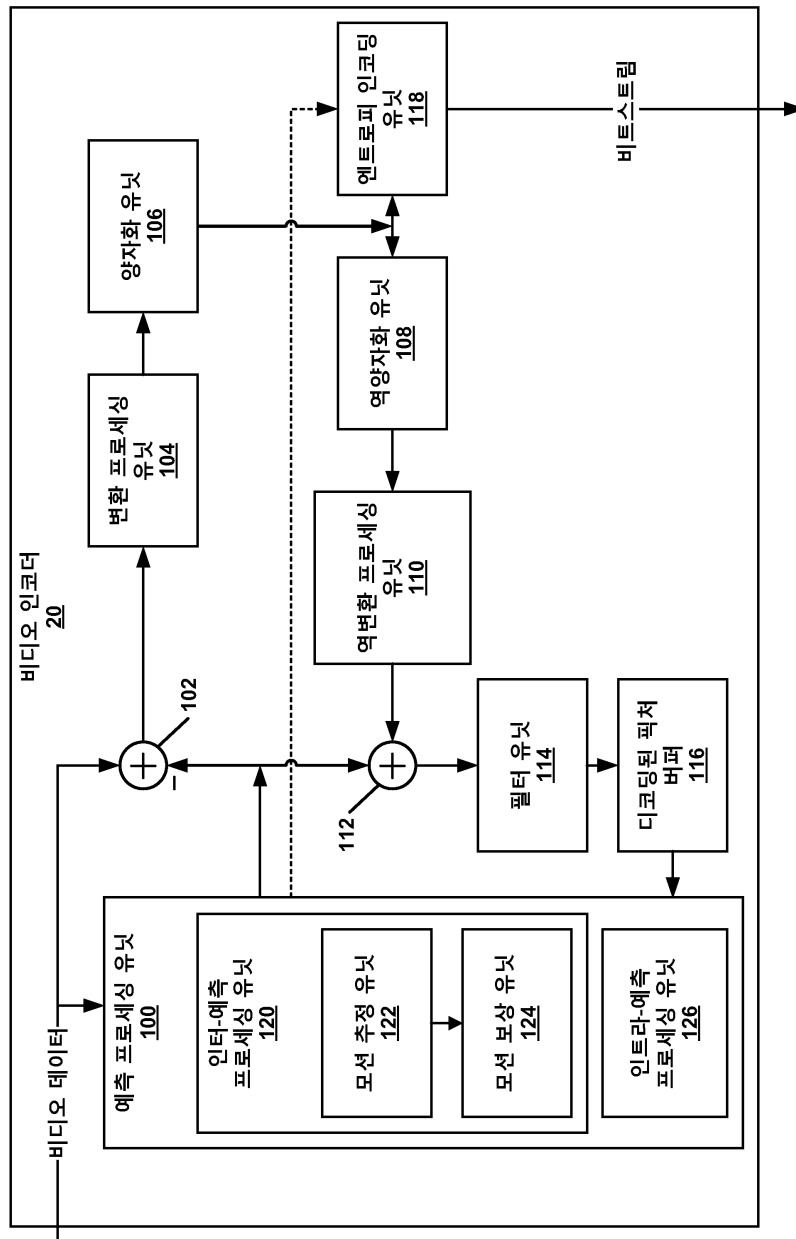


도면7



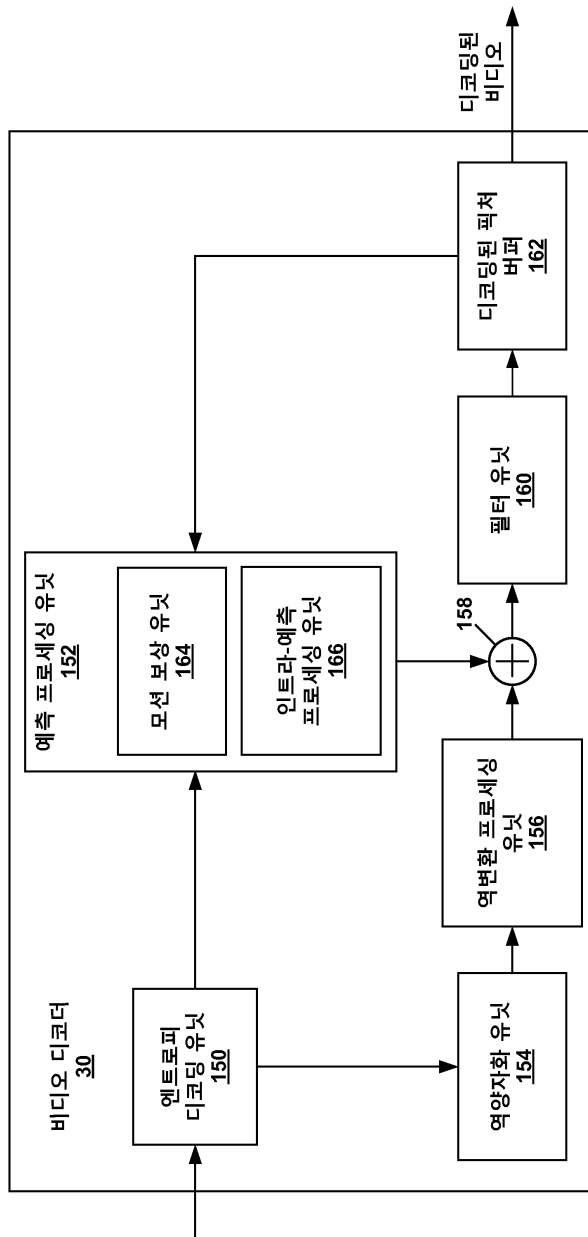


도면8





도면9





도면10





도면11

