



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2016-0106616  
(43) 공개일자 2016년09월12일

- |   |  |
|---|--|
| <p>(51) 국제특허분류(Int. Cl.)<br/> <i>H04N 19/593</i> (2014.01) <i>H04N 19/11</i> (2014.01)<br/> <i>H04N 19/154</i> (2014.01) <i>H04N 19/176</i> (2014.01)<br/> <i>H04N 19/186</i> (2014.01) <i>H04N 19/44</i> (2014.01)<br/> <i>H04N 19/513</i> (2014.01) <i>H04N 19/52</i> (2014.01)</p> <p>(52) CPC특허분류<br/> <i>H04N 19/593</i> (2015.01)<br/> <i>H04N 19/11</i> (2015.01)</p> <p>(21) 출원번호 10-2016-7020057<br/> (22) 출원일자(국제) 2015년01월09일<br/> 심사청구일자 없음<br/> (85) 번역문제출일자 2016년07월21일<br/> (86) 국제출원번호 PCT/US2015/010783<br/> (87) 국제공개번호 WO 2015/106078<br/> 국제공개일자 2015년07월16일<br/> (30) 우선권주장<br/> 61/925,647 2014년01월09일 미국(US)<br/> (뒷면에 계속)</p> | <p>(71) 출원인<br/> <b>켈컴 인코포레이티드</b><br/> 미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775</p> <p>(72) 발명자<br/> <b>천 잉</b><br/> 미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775<br/> <b>광 차오</b><br/> 미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775<br/> (뒷면에 계속)</p> <p>(74) 대리인<br/> <b>특허법인코리어나</b></p> |
|---|--|

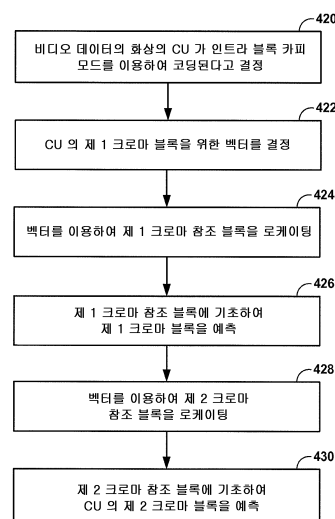
전체 청구항 수 : 총 43 항

(54) 발명의 명칭 예측 블록으로부터의 인트라 예측

(57) 요약

비디오 데이터를 코딩하기 위한 디바이스는 비디오 데이터를 저장하도록 구성된 메모리 및 하나 이상의 프로세서들을 포함하는 비디오 코더를 포함하고 상기 하나 이상의 프로세서들은, 비디오 데이터의 화상의 코딩 유닛이 인트라 블록 카피 모드를 이용하여 코딩되는 것을 결정하고; 코딩 유닛의 제 1 크로마 블록을 위한 벡터를 결정하고; 벡터를 이용하여 제 1 크로마 참조 블록을 로케이팅(locating) 하는 것으로서, 제 1 크로마 참조 블록은 화상에 있는, 상기 제 1 크로마 참조 블록을 로케이팅하고; 제 1 크로마 참조 블록에 기초하여 제 1 크로마 블록을 예측하고; 벡터를 이용하여 제 2 크로마 참조 블록을 로케이팅하는 것으로서, 제 2 크로마 참조 블록은 화상에 있는, 상기 제 2 크로마 참조 블록을 로케이팅하고; 제 2 크로마 참조 블록에 기초하여 코딩 유닛의 제 2 크로마 블록을 예측하도록 구성된다.

대표도 - 도9



(52) CPC특허분류

**HO4N 19/154** (2015.01)

**HO4N 19/176** (2015.01)

**HO4N 19/186** (2015.01)

**HO4N 19/44** (2015.01)

**HO4N 19/513** (2015.01)

**HO4N 19/52** (2015.01)

(72) 발명자

**장 리**

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

**솔레 로할스 호엘**

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

**카르체비츠 마르타**

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

(30) 우선권주장

61/926,224 2014년01월10일 미국(US)

14/592,819 2015년01월08일 미국(US)

## 명세서

### 청구범위

#### 청구항 1

비디오 데이터를 디코딩하는 방법으로서,

상기 비디오 데이터의 화상의 코딩 유닛이 인트라 블록 카피 모드를 이용하여 코딩되는 것을 결정하는 단계;

상기 코딩 유닛과 연관된 복수의 벡터들로부터 상기 코딩 유닛의 제 1 크로마 블록을 위한 벡터를 결정하는 단계;

상기 벡터를 이용하여 제 1 크로마 참조 블록을 로케이팅하는 단계로서, 상기 제 1 크로마 참조 블록은 상기 화상에 있는, 상기 제 1 크로마 참조 블록을 로케이팅하는 단계;

상기 제 1 크로마 참조 블록에 기초하여 상기 제 1 크로마 블록을 예측하는 단계;

상기 제 1 크로마 참조 블록을 로케이팅하는데 이용되었던 동일한 벡터를 이용하여, 동일한 코딩 유닛의 제 2 크로마 블록을 위한 제 2 크로마 참조 블록을 로케이팅하는 단계로서, 상기 제 2 크로마 참조 블록은 상기 화상에 있고, 상기 제 1 크로마 참조 블록과는 상이한, 상기 제 2 크로마 참조 블록을 로케이팅하는 단계; 및

상기 제 2 크로마 참조 블록에 기초하여 상기 코딩 유닛의 상기 제 2 크로마 블록을 예측하는 단계

를 포함하는, 비디오 데이터를 디코딩하는 방법.

#### 청구항 2

제 1 항에 있어서,

상기 코딩 유닛은 2개 이상의 예측 유닛들을 포함하고, 상기 제 1 크로마 블록은 상기 코딩 유닛의 제 1 예측 유닛에 대응하고, 상기 제 2 크로마 블록은 상기 코딩 유닛의 제 2 예측 유닛에 대응하고, 상기 코딩 유닛의 상기 제 1 크로마 블록을 위한 상기 벡터를 결정하는 단계는 상기 제 1 예측 유닛을 위한 벡터를 획득하는 단계를 포함하는, 비디오 데이터를 디코딩하는 방법.

#### 청구항 3

제 1 항에 있어서,

상기 코딩 유닛은 2개 이상의 예측 유닛들을 포함하고, 각각의 예측 유닛은 연관된 벡터를 갖고,

상기 방법은,

상기 코딩 유닛의 제 1 예측 유닛과 연관된 상기 벡터를 이용하여 제 1 루마 참조 블록을 로케이팅하는 단계;

상기 코딩 유닛의 제 2 예측 유닛과 연관된 상기 벡터를 이용하여 제 2 루마 참조 블록을 로케이팅하는 단계;

상기 제 1 예측 유닛과 연관된 상기 벡터를 이용하여 상기 제 1 크로마 참조 블록을 로케이팅하는 단계; 및

상기 제 1 예측 유닛과 연관된 상기 벡터를 이용하여 상기 제 2 크로마 참조 블록을 로케이팅하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

#### 청구항 4

제 1 항에 있어서,

상기 코딩 유닛을 위한 크로마 샘플링 포맷은 4:2:0 이고, 상기 복수의 벡터들로부터 결정된 상기 벡터는 상기 코딩 유닛의 하단 우측 블록과 연관되는, 비디오 데이터를 디코딩하는 방법.

#### 청구항 5

제 4 항에 있어서,

상기 제 2 크로마 블록은 상기 코딩 유닛의 상단 우측 블록이고,

상기 방법은,

상기 벡터를 이용하여 제 3 참조 블록을 로케이팅하는 단계로서, 상기 제 3 참조 블록은 상기 화상에 있는, 상기 제 3 참조 블록을 로케이팅하는 단계;

제 1 참조 블록에 기초하여 제 3 크로마 블록을 예측하는 단계로서, 상기 제 3 크로마 블록은 상기 코딩 유닛의 상단 좌측 블록인, 상기 제 3 크로마 블록을 예측하는 단계;

상기 벡터를 이용하여 제 4 참조 블록을 로케이팅하는 단계로서, 상기 제 4 참조 블록은 상기 화상에 있는, 상기 제 4 참조 블록을 로케이팅하는 단계;

제 2 참조 블록에 기초하여 상기 코딩 유닛의 제 4 크로마 블록을 예측하는 단계로서, 상기 제 4 크로마 블록은 상기 코딩 유닛의 하단 좌측 블록인, 상기 제 4 크로마 블록을 예측하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

## 청구항 6

제 1 항에 있어서,

상기 코딩 유닛을 위한 크로마 샘플링 포맷은 4:2:2 이고, 상기 제 1 크로마 블록은 상단 우측 블록이고 상기 제 2 크로마 블록은 상기 코딩 유닛의 상단 좌측 블록인, 비디오 데이터를 디코딩하는 방법.

## 청구항 7

제 6 항에 있어서,

상기 코딩 유닛의 제 3 크로마 블록을 위한 제 2 벡터를 결정하는 단계;

상기 제 2 벡터를 이용하여 제 3 참조 블록을 로케이팅하는 단계로서, 상기 제 3 참조 블록은 상기 화상에 있는, 상기 제 3 참조 블록을 로케이팅하는 단계;

상기 제 3 크로마 블록을 위한 예측자로서 상기 제 3 참조 블록을 이용하는 단계;

상기 벡터를 이용하여 제 4 참조 블록을 로케이팅하는 단계로서, 상기 제 4 참조 블록은 상기 화상에 있는, 상기 제 4 참조 블록을 로케이팅하는 단계;

상기 코딩 유닛의 제 4 크로마 블록을 위한 예측자로서 상기 제 4 참조 블록을 이용하는 단계를 더 포함하는, 비디오 데이터를 디코딩하는 방법.

## 청구항 8

제 1 항에 있어서,

상기 코딩 유닛은 잔차 데이터를 포함하는, 비디오 데이터를 디코딩하는 방법.

## 청구항 9

비디오 데이터를 인코딩하는 방법으로서,

화상의 코딩 유닛의 제 1 크로마 블록을 위한 벡터를 상기 코딩 유닛과 연관된 복수의 벡터들로부터 결정하는 단계;

상기 벡터를 이용하여 제 1 크로마 참조 블록을 로케이팅하는 단계로서, 상기 제 1 크로마 참조 블록은 상기 화상에 있는, 상기 제 1 크로마 참조 블록을 로케이팅하는 단계;

상기 제 1 크로마 참조 블록에 기초하여 상기 제 1 크로마 블록을 예측하는 단계;

상기 제 1 크로마 참조 블록을 로케이팅하는데 이용되었던 동일한 벡터를 이용하여, 동일한 코딩 유닛의 제 2 크로마 블록을 위한 제 2 크로마 참조 블록을 로케이팅하는 단계로서, 상기 제 2 크로마 참조 블록은 상기 화상에 있고, 상기 제 1 크로마 참조 블록과는 상이한, 상기 제 2 크로마 참조 블록을 로케이팅하는 단계;

상기 제 2 크로마 참조 블록에 기초하여 상기 코딩 유닛의 제 2 크로마 블록을 예측하는 단계; 및

인코딩된 상기 비디오 데이터에의 포함을 위해, 상기 화상의 상기 코딩 유닛이 인트라 블록 카피 모드를 이용하여 코딩된다는 표시 및 상기 코딩 유닛의 각 루마 블록을 위한 벡터의 표시를 생성하는 단계를 포함하는, 비디오 데이터를 인코딩하는 방법.

#### 청구항 10

제 9 항에 있어서,

상기 코딩 유닛은 2개 이상의 예측 유닛들을 포함하고, 상기 제 1 크로마 블록은 제 1 예측 유닛에 대응하고, 상기 제 2 크로마 블록은 제 2 예측 유닛에 대응하고, 상기 코딩 유닛의 상기 제 2 크로마 블록을 위한 상기 벡터를 결정하는 단계는 상기 제 1 예측 유닛을 위한 벡터를 획득하는 단계를 포함하는, 비디오 데이터를 인코딩하는 방법.

#### 청구항 11

제 9 항에 있어서,

상기 코딩 유닛은 2개 이상의 예측 유닛들을 포함하고, 각각의 예측 유닛은 연관된 벡터를 갖고,

상기 방법은,

제 1 예측 유닛과 연관된 상기 벡터를 이용하여 제 1 루마 참조 블록을 로케이팅하는 단계;

제 2 예측 유닛과 연관된 상기 벡터를 이용하여 제 2 루마 참조 블록을 로케이팅하는 단계;

상기 제 1 예측 유닛과 연관된 상기 벡터를 이용하여 상기 제 1 크로마 참조 블록을 로케이팅하는 단계; 및

상기 제 1 예측 유닛과 연관된 상기 벡터를 이용하여 상기 제 2 크로마 참조 블록을 로케이팅하는 단계를 더 포함하는, 비디오 데이터를 인코딩하는 방법.

#### 청구항 12

제 9 항에 있어서,

상기 코딩 유닛을 위한 크로마 샘플링 포맷은 4:2:0 이고, 상기 복수의 벡터들로부터 결정된 상기 벡터는 상기 코딩 유닛의 하단 우측 블록과 연관되는, 비디오 데이터를 인코딩하는 방법.

#### 청구항 13

제 12 항에 있어서,

상기 제 2 크로마 블록은 상기 코딩 유닛의 상단 우측 블록이고,

상기 방법은,

상기 벡터를 이용하여 제 3 참조 블록을 로케이팅하는 단계로서, 상기 제 3 참조 블록은 상기 화상에 있는, 상기 제 3 참조 블록을 로케이팅하는 단계;

제 1 참조 블록에 기초하여 제 3 크로마 블록을 예측하는 단계로서, 상기 제 3 크로마 블록은 상기 코딩 유닛의 상단 좌측 블록인, 상기 제 3 크로마 블록을 예측하는 단계;

상기 벡터를 이용하여 제 4 참조 블록을 로케이팅하는 단계로서, 상기 제 4 참조 블록은 상기 화상에 있는, 상기 제 4 참조 블록을 로케이팅하는 단계; 및

제 2 참조 블록에 기초하여 상기 코딩 유닛의 제 4 크로마 블록을 예측하는 단계로서, 상기 제 4 크로마 블록은 상기 코딩 유닛의 하단 좌측 블록인, 상기 제 4 크로마 블록을 예측하는 단계를 더 포함하는, 비디오 데이터를 인코딩하는 방법.

#### 청구항 14

제 9 항에 있어서,

상기 코딩 유닛을 위한 크로마 샘플링 포맷은 4:2:2 이고, 상기 제 1 크로마 블록은 상단 우측 블록이고 상기

제 2 크로마 블록은 상기 코딩 유닛의 상단 좌측 블록인, 비디오 데이터를 인코딩하는 방법.

#### 청구항 15

제 14 항에 있어서,

상기 코딩 유닛의 제 3 크로마 블록을 위한 제 2 벡터를 결정하는 단계;

상기 제 2 벡터를 이용하여 제 3 참조 블록을 로케이팅하는 단계로서, 상기 제 3 참조 블록은 상기 화상에 있는, 상기 제 3 참조 블록을 로케이팅하는 단계;

상기 제 3 크로마 블록을 위한 예측자로서 상기 제 3 참조 블록을 이용하는 단계;

상기 벡터를 이용하여 제 4 참조 블록을 로케이팅하는 단계로서, 상기 제 4 참조 블록은 상기 화상에 있는, 상기 제 4 참조 블록을 로케이팅하는 단계; 및

상기 코딩 유닛의 제 4 크로마 블록을 위한 예측자로서 상기 제 4 참조 블록을 이용하는 단계를 더 포함하는, 비디오 데이터를 인코딩하는 방법.

#### 청구항 16

제 9 항에 있어서,

상기 코딩 유닛은 잔차 데이터를 포함하는, 비디오 데이터를 인코딩하는 방법.

#### 청구항 17

비디오 데이터를 코딩하기 위한 디바이스로서,

비디오 데이터를 저장하도록 구성된 메모리;

하나 이상의 프로세서들을 포함하는 비디오 코더를 포함하고,

상기 하나 이상의 프로세서들은,

상기 비디오 데이터의 화상의 코딩 유닛이 인트라 블록 카피 모드를 이용하여 코딩되는 것을 상기 코딩 유닛과 연관된 복수의 벡터들로부터 결정하고;

상기 코딩 유닛의 제 1 크로마 블록을 위한 벡터를 결정하고;

상기 벡터를 이용하여 제 1 크로마 참조 블록을 로케이팅하는 것으로서, 상기 제 1 크로마 참조 블록은 상기 화상에 있는, 상기 제 1 크로마 참조 블록을 로케이팅하고;

상기 제 1 크로마 참조 블록에 기초하여 상기 제 1 크로마 블록을 예측하고;

상기 제 1 크로마 참조 블록을 로케이팅하는데 이용되었던 동일한 벡터를 이용하여, 동일한 코딩 유닛의 제 2 크로마 블록을 위한 제 2 크로마 참조 블록을 로케이팅하는 것으로서, 상기 제 2 크로마 참조 블록은 상기 화상에 있고, 상기 제 1 크로마 참조 블록과는 상이한, 상기 제 2 크로마 참조 블록을 로케이팅하고; 및

상기 제 2 크로마 참조 블록에 기초하여 상기 코딩 유닛의 제 2 크로마 블록을 예측하도록 구성되는, 비디오 데이터를 코딩하기 위한 디바이스.

#### 청구항 18

제 17 항에 있어서,

상기 코딩 유닛은 2개 이상의 예측 유닛들을 포함하고, 상기 제 1 크로마 블록은 제 1 예측 유닛에 대응하고, 상기 제 2 크로마 블록은 제 2 예측 유닛에 대응하고, 상기 코딩 유닛의 상기 제 2 크로마 블록을 위한 상기 벡터를 결정하는 것은 상기 제 1 예측 유닛을 위한 벡터를 획득하는 것을 포함하는, 비디오 데이터를 코딩하기 위한 디바이스.

#### 청구항 19

제 17 항에 있어서,

상기 코딩 유닛은 2개 이상의 예측 유닛들을 포함하고, 각각의 예측 유닛은 연관된 벡터를 갖고,  
 상기 하나 이상의 프로세서들은 또한  
 상기 제 1 예측 유닛과 연관된 상기 벡터를 이용하여 상기 제 1 루마 참조 블록을 로케이팅하고;  
 상기 제 2 예측 유닛과 연관된 상기 벡터를 이용하여 상기 제 2 루마 참조 블록을 로케이팅하고;  
 상기 제 1 예측 유닛과 연관된 상기 벡터를 이용하여 상기 제 1 크로마 참조 블록을 로케이팅하고; 및  
 상기 제 1 예측 유닛과 연관된 상기 벡터를 이용하여 상기 제 2 크로마 참조 블록을 로케이팅하도록 구성되는,  
 비디오 데이터를 코딩하기 위한 디바이스.

#### 청구항 20

제 17 항에 있어서,  
 상기 코딩 유닛을 위한 크로마 샘플링 포맷은 4:2:0 이고, 상기 복수의 벡터들로부터 결정된 상기 벡터는 상기 코딩 유닛의 하단 우측 블록과 연관되는, 비디오 데이터를 코딩하기 위한 디바이스.

#### 청구항 21

제 20 항에 있어서,  
 상기 제 2 크로마 블록은 상기 코딩 유닛의 상단 우측 블록이고,  
 상기 하나 이상의 프로세서들은 또한,  
 상기 벡터를 이용하여 제 3 참조 블록을 로케이팅하는 것으로서, 상기 제 3 참조 블록은 상기 화상에 있는, 상기 제 3 참조 블록을 로케이팅하고;  
 제 1 참조 블록에 기초하여 제 3 크로마 블록을 예측하는 것으로서, 상기 제 3 크로마 블록은 상기 코딩 유닛의 상단 좌측 블록인, 상기 제 3 크로마 블록을 예측하고;  
 상기 벡터를 이용하여 제 4 참조 블록을 로케이팅하는 것으로서, 상기 제 4 참조 블록은 상기 화상에 있는, 상기 제 4 참조 블록을 로케이팅하고; 및  
 제 2 참조 블록에 기초하여 상기 코딩 유닛의 제 4 크로마 블록을 예측하는 것으로서, 상기 제 4 크로마 블록은 상기 코딩 유닛의 하단 좌측 블록인, 상기 제 4 크로마 블록을 예측하도록 구성되는, 비디오 데이터를 코딩하기 위한 디바이스.

#### 청구항 22

제 17 항에 있어서,  
 상기 코딩 유닛을 위한 크로마 샘플링 포맷은 4:2:2 이고, 상기 제 1 크로마 블록은 상단 우측 블록이고 상기 제 2 크로마 블록은 상기 코딩 유닛의 상단 좌측 블록인, 비디오 데이터를 코딩하기 위한 디바이스.

#### 청구항 23

제 22 항에 있어서,  
 상기 하나 이상의 프로세서들은 또한  
 상기 코딩 유닛의 제 3 크로마 블록을 위한 제 2 벡터를 결정하고;  
 상기 제 2 벡터를 이용하여 제 3 참조 블록을 로케이팅하는 것으로서, 상기 제 3 참조 블록은 상기 화상에 있는, 상기 제 3 참조 블록을 로케이팅하고;  
 상기 제 3 크로마 블록을 위한 예측자로서 상기 제 3 참조 블록을 이용하고;  
 상기 벡터를 이용하여 제 4 참조 블록을 로케이팅하는 것으로서, 상기 제 4 참조 블록은 상기 화상에 있는, 상기 제 4 참조 블록을 로케이팅하고; 및  
 상기 코딩 유닛의 제 4 크로마 블록을 위한 예측자로서 상기 제 4 참조 블록을 이용하도록 구성되는, 비디오 테

이터를 코딩하기 위한 디바이스.

#### 청구항 24

제 17 항에 있어서,

상기 코딩 유닛은 잔차 데이터를 포함하는, 비디오 데이터를 코딩하기 위한 디바이스.

#### 청구항 25

제 17 항에 있어서,

상기 디바이스는

마이크로프로세서;

집적 회로 (IC); 및

비디오 인코더를 포함하는 무선 통신 디바이스

중 하나를 포함하는, 비디오 데이터를 코딩하기 위한 디바이스.

#### 청구항 26

제 17 항에 있어서,

상기 비디오 코더는 비디오 인코더를 포함하는, 비디오 데이터를 코딩하기 위한 디바이스.

#### 청구항 27

제 17 항에 있어서,

상기 비디오 코더는 비디오 디코더를 포함하는, 비디오 데이터를 코딩하기 위한 디바이스.

#### 청구항 28

비디오 데이터를 코딩하기 위한 장치로서,

상기 비디오 데이터의 화상의 코딩 유닛이 인트라 블록 카피 모드를 이용하여 코딩되는 것을 상기 코딩 유닛과 연관된 복수의 벡터들로부터 결정하는 수단;

상기 코딩 유닛의 제 1 크로마 블록을 위한 벡터를 결정하는 수단;

상기 제 1 크로마 참조 블록을 로케이팅하는데 이용되었던 동일한 벡터를 이용하여, 동일한 코딩 유닛의 제 2 크로마 블록을 위한 제 2 크로마 참조 블록을 로케이팅하는 수단으로서, 상기 제 2 크로마 참조 블록은 상기 화상에 있고, 상기 제 1 크로마 참조 블록과는 상이한, 상기 제 2 크로마 참조 블록을 로케이팅하는 수단;

상기 제 1 크로마 참조 블록에 기초하여 상기 제 1 크로마 블록을 예측하는 수단;

상기 벡터를 이용하여 제 2 크로마 참조 블록을 로케이팅하는 수단으로서, 상기 제 2 크로마 참조 블록은 상기 화상에 있는, 상기 제 2 크로마 참조 블록을 로케이팅하는 수단; 및

상기 제 2 크로마 참조 블록에 기초하여 상기 코딩 유닛의 제 2 크로마 블록을 예측하는 수단

을 포함하는, 비디오 데이터를 코딩하기 위한 장치.

#### 청구항 29

제 28 항에 있어서,

상기 코딩 유닛은 2개 이상의 예측 유닛들을 포함하고, 상기 제 1 크로마 블록은 제 1 예측 유닛에 대응하고, 상기 제 2 크로마 블록은 제 2 예측 유닛에 대응하고, 상기 코딩 유닛의 상기 제 2 크로마 블록을 위한 상기 벡터를 결정하는 수단은 상기 제 1 예측 유닛을 위한 벡터를 획득하는 수단을 포함하는, 비디오 데이터를 코딩하기 위한 장치.



### 청구항 30

제 28 항에 있어서,

상기 코딩 유닛은 2개 이상의 예측 유닛들을 포함하고, 각각의 예측 유닛은 연관된 벡터를 갖고,

상기 장치는

제 1 예측 유닛과 연관된 상기 벡터를 이용하여 제 1 루마 참조 블록을 로케이팅하는 수단;

제 2 예측 유닛과 연관된 상기 벡터를 이용하여 제 2 루마 참조 블록을 로케이팅하는 수단;

상기 제 1 예측 유닛과 연관된 상기 벡터를 이용하여 상기 제 1 크로마 참조 블록을 로케이팅하는 수단; 및

상기 제 1 예측 유닛과 연관된 상기 벡터를 이용하여 상기 제 2 크로마 참조 블록을 로케이팅하는 수단을 더 포함하는, 비디오 데이터를 코딩하기 위한 장치.

### 청구항 31

제 28 항에 있어서,

상기 코딩 유닛을 위한 크로마 샘플링 포맷은 4:2:0 이고, 상기 복수의 벡터들로부터 결정된 상기 벡터는 상기 코딩 유닛의 하단 우측 블록과 연관되는, 비디오 데이터를 코딩하기 위한 장치.

### 청구항 32

제 31 항에 있어서,

상기 제 2 크로마 블록은 상기 코딩 유닛의 상단 우측 블록이고,

상기 장치는,

상기 벡터를 이용하여 제 3 참조 블록을 로케이팅하는 수단으로서, 상기 제 3 참조 블록은 상기 화상에 있는, 상기 제 3 참조 블록을 로케이팅하는 수단;

제 1 참조 블록에 기초하여 제 3 크로마 블록을 예측하는 수단으로서, 상기 제 3 크로마 블록은 상기 코딩 유닛의 상단 좌측 블록인, 상기 제 3 크로마 블록을 예측하는 수단;

상기 벡터를 이용하여 제 4 참조 블록을 로케이팅하는 수단으로서, 상기 제 4 참조 블록은 상기 화상에 있는, 상기 제 4 참조 블록을 로케이팅하는 수단; 및

제 2 참조 블록에 기초하여 상기 코딩 유닛의 제 4 크로마 블록을 예측하는 수단으로서, 상기 제 4 크로마 블록은 상기 코딩 유닛의 하단 좌측 블록인, 상기 제 4 크로마 블록을 예측하는 수단을 더 포함하는, 비디오 데이터를 코딩하기 위한 장치.

### 청구항 33

제 28 항에 있어서,

상기 코딩 유닛을 위한 크로마 샘플링 포맷은 4:2:2 이고, 상기 제 1 크로마 블록은 상단 우측 블록이고 상기 제 2 크로마 블록은 상기 코딩 유닛의 상단 좌측 블록인, 비디오 데이터를 코딩하기 위한 장치.

### 청구항 34

제 33 항에 있어서,

상기 코딩 유닛의 제 3 크로마 블록을 위한 제 2 벡터를 결정하는 수단;

상기 제 2 벡터를 이용하여 제 3 참조 블록을 로케이팅하는 수단으로서, 상기 제 3 참조 블록은 상기 화상에 있는, 상기 제 3 참조 블록을 로케이팅하는 수단;

상기 제 3 크로마 블록을 위한 예측자로서 상기 제 3 참조 블록을 이용하는 수단;

상기 벡터를 이용하여 제 4 참조 블록을 로케이팅하는 수단으로서, 상기 제 4 참조 블록은 상기 화상에 있는, 상기 제 4 참조 블록을 로케이팅하는 수단; 및

상기 코딩 유닛의 제 4 크로마 블록을 위한 예측자로서 상기 제 4 참조 블록을 이용하는 수단을 더 포함하는, 비디오 데이터를 코딩하기 위한 장치.

#### 청구항 35

제 28 항에 있어서,

상기 코딩 유닛은 잔차 데이터를 포함하는, 비디오 데이터를 코딩하기 위한 장치.

#### 청구항 36

명령들을 저장한 컴퓨터 판독가능 저장 매체로서,

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

비디오 데이터의 화상의 코딩 유닛이 인트라 블록 카피 모드를 이용하여 코딩되는 것을 상기 코딩 유닛과 연관된 복수의 벡터들로부터 결정하게 하고;

상기 코딩 유닛의 제 1 크로마 블록을 위한 벡터를 결정하게 하고;

상기 제 1 크로마 참조 블록을 로케이팅하는데 이용되었던 동일한 벡터를 이용하여, 동일한 코딩 유닛의 제 2 크로마 블록을 위한 제 2 크로마 참조 블록을 로케이팅하게 하는 것으로서, 상기 제 2 크로마 참조 블록은 상기 화상에 있고, 상기 제 1 크로마 참조 블록과는 상이한, 상기 제 2 크로마 참조 블록을 로케이팅하게 하고;

상기 제 1 크로마 참조 블록에 기초하여 상기 제 1 크로마 블록을 예측하게 하고;

상기 벡터를 이용하여 제 2 크로마 참조 블록을 로케이팅하게 하는 것으로서, 상기 제 2 크로마 참조 블록은 상기 화상에 있는, 상기 제 2 크로마 참조 블록을 로케이팅하게 하고; 및

상기 제 2 크로마 참조 블록에 기초하여 상기 코딩 유닛의 제 2 크로마 블록을 예측하게 하는, 컴퓨터 판독가능 저장 매체.

#### 청구항 37

제 36 항에 있어서,

상기 코딩 유닛은 2개 이상의 예측 유닛들을 포함하고, 상기 제 1 크로마 블록은 제 1 예측 유닛에 대응하고, 상기 제 2 크로마 블록은 제 2 예측 유닛에 대응하고, 상기 코딩 유닛의 상기 제 2 크로마 블록을 위한 상기 벡터를 결정하는 것은 상기 제 1 예측 유닛을 위한 벡터를 획득하는 것을 포함하는, 컴퓨터 판독가능 저장 매체.

#### 청구항 38

제 36 항에 있어서,

상기 코딩 유닛은 2개 이상의 예측 유닛들을 포함하고, 각각의 예측 유닛은 연관된 벡터를 갖고, 상기 컴퓨터 판독가능 저장 매체는, 하나 이상의 프로세서들에 의해 실행될 때 상기 하나 이상의 프로세서들로 하여금,

제 1 예측 유닛과 연관된 상기 벡터를 이용하여 상기 제 1 루마 참조 블록을 로케이팅하게 하고;

제 2 예측 유닛과 연관된 상기 벡터를 이용하여 상기 제 2 루마 참조 블록을 로케이팅하게 하고;

상기 제 1 예측 유닛과 연관된 상기 벡터를 이용하여 상기 제 1 크로마 참조 블록을 로케이팅하게 하고; 및

상기 제 1 예측 유닛과 연관된 상기 벡터를 이용하여 상기 제 2 크로마 참조 블록을 로케이팅하게 하는 추가의 명령들을 저장한, 컴퓨터 판독가능 저장 매체.

#### 청구항 39

제 36 항에 있어서,

상기 코딩 유닛을 위한 크로마 샘플링 포맷은 4:2:0 이고, 상기 복수의 벡터들로부터 결정된 상기 벡터는 상기 코딩 유닛의 하단 우측 블록과 연관되는, 컴퓨터 판독가능 저장 매체.

#### 청구항 40

제 39 항에 있어서,

상기 제 2 크로마 블록은 상기 코딩 유닛의 상단 우측 블록이고, 상기 컴퓨터 판독가능 저장 매체는 상기 하나 이상의 프로세서들에 의해 실행될 때 상기 하나 이상의 프로세서들로 하여금,

상기 벡터를 이용하여 제 3 참조 블록을 로케이팅하게 하는 것으로서, 상기 제 3 참조 블록은 상기 화상에 있는, 상기 제 3 참조 블록을 로케이팅하게 하고;

제 1 참조 블록에 기초하여 제 3 크로마 블록을 예측하게 하는 것으로서, 상기 제 3 크로마 블록은 상기 코딩 유닛의 상단 좌측 블록인, 상기 제 3 크로마 블록을 예측하게 하고;

상기 벡터를 이용하여 제 4 참조 블록을 로케이팅하게 하는 것으로서, 상기 제 4 참조 블록은 상기 화상에 있는, 상기 제 4 참조 블록을 로케이팅하게 하고; 및

제 2 참조 블록에 기초하여 상기 코딩 유닛의 제 4 크로마 블록을 예측하게 하는 것으로서, 상기 제 4 크로마 블록은 상기 코딩 유닛의 하단 좌측 블록인, 상기 제 4 크로마 블록을 예측하게 하는 추가의 명령들을 저장한, 컴퓨터 판독가능 저장 매체.

#### 청구항 41

제 36 항에 있어서,

상기 코딩 유닛을 위한 크로마 샘플링 포맷은 4:2:2 이고, 상기 제 1 크로마 블록은 상단 우측 블록이고 상기 제 2 크로마 블록은 상기 코딩 유닛의 상단 좌측 블록인, 컴퓨터 판독가능 저장 매체.

#### 청구항 42

제 41 항에 있어서,

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

상기 코딩 유닛의 제 3 크로마 블록을 위한 제 2 벡터를 결정하게 하고;

상기 제 2 벡터를 이용하여 제 3 참조 블록을 로케이팅하게 하는 것으로서, 상기 제 3 참조 블록은 상기 화상에 있는, 상기 제 3 참조 블록을 로케이팅하게 하고;

상기 제 3 크로마 블록을 위한 예측자로서 상기 제 3 참조 블록을 이용하게 하고;

상기 벡터를 이용하여 제 4 참조 블록을 로케이팅하게 하는 것으로서, 상기 제 4 참조 블록은 상기 화상에 있는, 상기 제 4 참조 블록을 로케이팅하게 하고; 및

상기 코딩 유닛의 제 4 크로마 블록을 위한 예측자로서 상기 제 4 참조 블록을 이용하게 하는 추가의 명령들을 저장한, 컴퓨터 판독가능 저장 매체.

#### 청구항 43

제 36 항에 있어서,

상기 코딩 유닛은 잔차 데이터를 포함하는, 컴퓨터 판독가능 저장 매체.

### 발명의 설명

### 기술 분야

[0001] 본원은 하기 U.S. 가출원들의 각각의 혜택을 주장하며, 이들의 각각의 전체 내용은 참조에 의해 본원에 인용된다:

[0002] 2014년 1월 9일자로 출원된 U.S. 가출원 번호 No. 61/925,647;

[0003] 2014년 1월 10일자로 출원된 U.S. 가출원 번호 No. 61/926,224.

[0004] 기술분야

[0005] 본 개시는 비디오 코딩에 관한 것이고, 보다 구체적으로는, 다른 비디오 블록들에 기초한 비디오 블록들의 예측에 관한 것이다.

## 배경 기술

[0006] 디지털 비디오 능력들은 디지털 텔레비전들, 디지털 다이렉트 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인 휴대정보 단말기 (PDA) 들, 랩탑 또는 데스크탑 컴퓨터들, 태블릿 컴퓨터들, 전자책 단말기, 디지털 카메라들, 디지털 레코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 무선 전화기들, 소위 "스마트 폰", 원격 화상회의 디바이스들, 비디오 스트리밍 디바이스들 등을 포함한, 광범위한 디바이스들에 포함될 수 있다. 디지털 비디오 디바이스들은, MPEG-2, MPEG-4, ITU-T H.263 또는 ITU-T H.264/MPEG-4, Part 10, AVC (Advanced Video Coding) 에 의해 정의되는 표준들, 그리고 현재 개발 중인 HEVC (High Efficiency Video Coding) 표준, 및 그러한 표준들의 확장들에서 설명된 것들과 같은 비디오 압축 기법들을 구현한다. 비디오 디바이스들은, 그러한 비디오 압축기법들을 구현함으로써 보다 효율적으로 디지털 비디오 정보를 송신, 수신, 인코딩, 디코딩, 및/또는 저장할 수도 있다.

[0007] 비디오 압축 기법들은, 비디오 시퀀스들에 내재하는 중복성 (redundancy) 을 감소시키거나 또는 제거하기 위해 공간 (인트라-화상) 예측 및/또는 시간 (인터-화상) 예측을 수행한다. 블록 기반 비디오 코딩을 위해, 비디오 슬라이스 (즉, 비디오 프레임, 또는 비디오 프레임의 일부) 는 비디오 블록들로 파티셔닝될 수도 있고, 이 비디오 블록들은 트리 블록들, 코딩 유닛 (CU) 들, 및/또는 코딩 노드들로도 지칭될 수도 있다. 화상의 인트라-코딩된 (I) 슬라이스에서의 비디오 블록들은 동일한 화상에서 이웃 블록들에 있는 참조 샘플들에 대한 공간 예측을 이용하여 인코딩된다. 화상의 인터-코딩된 (P 또는 B) 슬라이스에서의 비디오블록들은 동일한 화상에서의 이웃 블록들에 있는 참조 샘플들에 대한 공간 예측, 또는 다른 참조 화상들에서의 참조 샘플들에 대한 시간 예측을 이용할 수도 있다. 화상들은 프레임들로 지칭될 수도 있고, 참조 화상들은 참조 프레임들로 지칭될 수도 있다.

[0008] 공간 또는 시간 예측은 코딩될 블록을 위한 예측 블록을 초래한다. 잔차 데이터는 코딩될 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 표시하는 잔차 데이터에 따라 인코딩된다. 인트라-코딩된 블록은 인트라-코딩 모드 및 잔차 데이터에 따라 인코딩된다. 추가 압축을 위해, 잔차 데이터는 픽셀 도메인으로부터 변환 도메인으로 변환되어, 잔차 변환 계수들을 초래할 수도 있으며, 다음으로 이들은 양자화될 수도 있다. 초기에 2 차원 행렬로 배열된, 양자화된 변환 계수들은, 변환 계수들의 1 차원 벡터를 생성하기 위하여 스캔될 수도 있고, 엔트로피 코딩이 적용되어 훨씬 더 많은 압축을 달성할 수도 있다.

## 발명의 내용

[0009] 개요

[0010] 일반적으로, 본 개시는, IntraBC 또는 IntraMC 모드로서 지칭될 수도 있는 동일한 화상 내에 비디오 데이터의 예측 블록들로부터 비디오 데이터의 블록들의 인트라 예측을 위한 모드를 포함하는 비디오 데이터를 코딩하기 위한 기법들을 설명한다.

[0011] 일 예에서, 비디오 데이터를 디코딩하는 방법은 비디오 데이터의 화상의 코딩 유닛이 인트라 블록 카피 모드를 이용하여 코딩되는 것을 결정하는 단계; 코딩 유닛과 연관된 복수의 벡터들로부터 코딩 유닛의 제 1 크로마 블록을 위한 벡터를 결정하는 단계; 벡터를 이용하여 제 1 크로마 참조 블록을 로케이팅 (locating) 하는 단계로서, 제 1 크로마 참조 블록은 화상에 있는, 상기 제 1 크로마 참조 블록을 로케이팅하는 단계; 제 1 크로마 참조 블록에 기초하여 제 1 크로마 블록을 예측하는 단계; 벡터를 이용하여 제 2 크로마 참조 블록을 로케이팅하는 단계로서, 제 2 크로마 참조 블록은 화상에 있는, 상기 제 2 크로마 참조 블록을 로케이팅하는 단계; 및 제 2 크로마 참조 블록에 기초하여 코딩 유닛의 제 2 크로마 블록을 예측하는 단계를 포함한다.

[0012] 다른 예에서, 비디오 데이터를 인코딩하는 방법은 화상의 코딩 유닛의 제 1 크로마 블록을 위한 벡터를 코딩 유닛과 연관된 복수의 벡터들로부터 결정하는 단계; 벡터를 이용하여 제 1 크로마 참조 블록을 로케이팅 (locating) 하는 단계로서, 제 1 크로마 참조 블록은 화상에 있는, 상기 제 1 크로마 참조 블록을 로케이팅하는 단계; 제 1 크로마 참조 블록에 기초하여 제 1 크로마 블록을 예측하는 단계; 벡터를 이용하여 제 2 크로마 참조 블록을 로케이팅하는 단계로서, 제 2 크로마 참조 블록은 화상에 있는, 상기 제 2 크로마 참조 블록을 로케이팅하는 단계; 제 2 크로마 참조 블록에 기초하여 코딩 유닛의 제 2 크로마 블록을 예측하는 단계; 및 인코딩

된 비디오 데이터에의 포함을 위해, 화상의 코딩 유닛이 인트라 블록 카피 모드를 이용하여 코딩된다는 표시 및 코딩 유닛의 각 루마 블록을 위한 벡터의 표시를 생성하는 단계를 포함한다.

[0013] 다른 예에서, 비디오 데이터를 코딩하기 위한 장치는 비디오 데이터의 화상의 코딩 유닛이 인트라 블록 카피 모드를 이용하여 코딩되는 것을 코딩 유닛과 연관된 복수의 벡터들로부터 결정하는 수단; 코딩 유닛의 제 1 크로마 블록을 위한 벡터를 결정하는 수단; 벡터를 이용하여 제 1 크로마 참조 블록을 로케이팅 (locating) 하는 수단로서, 제 1 크로마 참조 블록은 화상에 있는, 상기 제 1 크로마 참조 블록을 로케이팅하는 수단; 제 1 크로마 참조 블록에 기초하여 제 1 크로마 블록을 예측하는 수단; 벡터를 이용하여 제 2 크로마 참조 블록을 로케이팅 하는 수단로서, 제 2 크로마 참조 블록은 화상에 있는, 상기 제 2 크로마 참조 블록을 로케이팅하는 수단; 및 제 2 크로마 참조 블록에 기초하여 코딩 유닛의 제 2 크로마 블록을 예측하는 수단을 포함한다.

[0014] 비디오 데이터를 코딩하기 위한 디바이스는 비디오 데이터를 저장하도록 구성된 메모리 및 하나 이상의 프로세서들을 포함하는 비디오 코더를 포함하고 상기 하나 이상의 프로세서들은, 비디오 데이터의 화상의 코딩 유닛이 인트라 블록 카피 모드를 이용하여 코딩되는 것을 코딩 유닛과 연관된 복수의 벡터들로부터 결정하고; 코딩 유닛의 제 1 크로마 블록을 위한 벡터를 결정하고; 벡터를 이용하여 제 1 크로마 참조 블록을 로케이팅 (locating) 하는 것으로서, 제 1 크로마 참조 블록은 화상에 있는, 상기 제 1 크로마 참조 블록을 로케이팅하고; 제 1 크로마 참조 블록에 기초하여 제 1 크로마 블록을 예측하고; 벡터를 이용하여 제 2 크로마 참조 블록을 로케이팅하는 것으로서, 제 2 크로마 참조 블록은 화상에 있는, 상기 제 2 크로마 참조 블록을 로케이팅하고; 제 2 크로마 참조 블록에 기초하여 코딩 유닛의 제 2 크로마 블록을 예측하도록 구성된다.

[0015] 다른 예에서, 컴퓨터 판독가능 저장 매체는 명령들을 저장하고 그 명령들은 하나 이상의 프로세서들에 의해 실행될 때, 하나 이상의 프로세서로 하여금, 비디오 데이터의 화상의 코딩 유닛이 인트라 블록 카피 모드를 이용하여 코딩되는 것을 코딩 유닛과 연관된 복수의 벡터들로부터 결정하게 하고; 코딩 유닛의 제 1 크로마 블록을 위한 벡터를 결정하게 하고; 벡터를 이용하여 제 1 크로마 참조 블록을 로케이팅 (locating) 하게 하는 것으로서, 제 1 크로마 참조 블록은 화상에 있는, 상기 제 1 크로마 참조 블록을 로케이팅하게 하고; 제 1 크로마 참조 블록에 기초하여 제 1 크로마 블록을 예측하게 하고; 벡터를 이용하여 제 2 크로마 참조 블록을 로케이팅하게 하는 것으로서, 제 2 크로마 참조 블록은 화상에 있는, 상기 제 2 크로마 참조 블록을 로케이팅하게 하고; 제 2 크로마 참조 블록에 기초하여 코딩 유닛의 제 2 크로마 블록을 예측하게 한다.

[0016] 하나 이상의 예들의 상세는 첨부 도면 및 아래의 설명에 제시되어 있다. 다른 특징, 목적 및 이점들은 상세한 설명 및 도면, 그리고 청구항들로부터 분명해질 것이다.

## 도면의 간단한 설명

[0017] 도 1 은 본 개시에 기재된 기법들을 이용할 수도 있는 일 예의 비디오 인코딩 및 디코딩 시스템을 예시하는 블록도이다.

도 2는 본 개시의 기법들에 따라 현재 화상 내의 비디오 데이터의 현재 블록을 예측하기 위한 현재 블록 내의 비디오 데이터의 일 예의 예측 블록을 예시하는 개념도이다.

도 3은, 비디오 데이터의 예측 블록이 선택될 수도 있는, 의도된 영역을 정의하기 위한 일 예의 기법을 예시하는 개념도이다.

도 4는 비디오 데이터의 현재 블록, 및 비디오 데이터의 현재 블록을 위한 후보 예측 벡터들이 도출될 수도 있는 비디오 데이터의 이웃 블록들의 일 예의 개념도이다.

도 5a는 병합 모드를 위한 공간 이웃 모션 벡터 후보들의 예를 보여준다.

도 5b는 진보 모션 벡터 예측 (AMVP) 모드를 위한 공간 이웃 모션 벡터 후보들의 예를 보여준다.

도 6은 본 개시에 기재된 기법들을 구현할 수도 있는 일 예의 비디오 인코더를 예시하는 블록도이다.

도 7은 본 개시에 기재된 기법들을 구현할 수도 있는 일 예의 비디오 디코더를 예시하는 블록도이다.

도 8은 본 개시의 기법들에 따라 비디오 데이터를 인코딩하는 일 예의 방법을 예시하는 흐름도이다.

도 9는 본 개시의 기법들에 따라 비디오 데이터를 디코딩하는 일 예의 방법을 예시하는 흐름도이다.

## 발명을 실시하기 위한 구체적인 내용

- [0018] 상세한 설명
- [0019] 비디오 시퀀스는 일반적으로 화상들의 시퀀스로서 표현된다. 통상적으로, 블록 기반 코딩 기법들은 개개의 화상들의 각각을 코딩하는데 이용된다. 즉, 각각의 화상은 블록들로 분할되고, 블록들의 각각은 개별적으로 코딩된다. 비디오 데이터의 블록을 코딩하는 것은 일반적으로, 블록을 위한 예측 값을 형성하는 것 및 잔차 값, 즉 원래 블록과 예측 값 사이의 차이를 코딩하는 것을 수반한다. 특히, 비디오 데이터의 원래 블록은 픽셀 값들의 매트릭스를 포함하고, 예측 값은 예측 픽셀 값들의 매트릭스를 포함한다. 잔차 값은 원래 블록의 픽셀 값들과 예측 픽셀 값들 사이의 픽셀 바이 픽셀 (pixel-by-pixel) 차이에 대응한다.
- [0020] 비디오 데이터의 블록을 위한 예측 기법들은 일반적으로, 인트라 예측 또는 인터 예측으로서 카테고리화된다. 인트라 예측, 또는 공간 예측은 일반적으로, 이전에 코딩된 블록들의 부분인 이웃하는 픽셀 값들로부터 블록을 예측하는 것을 수반한다. 인터 예측, 또는 시간 예측은 일반적으로, 이전에 코딩된 화상들 (예를 들어, 프레임들 또는 슬라이스들)의 픽셀 값들로부터 블록을 예측하는 것을 수반한다.
- [0021] 원격 데스크톱, 원격 게이밍, 무선 디스플레이, 자동차 인포테인먼트 (infotainment), 클라우드 컴퓨팅 및 다른 것들과 같은 많은 응용들이 일상 생활에 있어서 일상화되고 있다. 이들 응용들에 있어서 비디오 콘텐츠는 보통, 자연 콘텐츠, 텍스트, 인공 그래픽, 및 다른 콘텐츠들의 조합이다. 텍스트 및 인공 그래픽 영역들에서, 반복된 패턴들 (이를테면, 문자 (character), 아이콘, 심볼 등)이 종종 존재한다.
- [0022] 인트라 블록 카피 (IntraBC) 또는 인트라 모션 보상 (IntraMC)으로 지칭될 수도 있는 동일한 화상 내의 비디오 데이터의 예측 블록들로부터 비디오 데이터의 블록들의 인트라 예측은, 비디오 코더로 하여금 그러한 리던던시 (redundancy) 및 인트라 프레임 코딩 효율성을 향상시키는 것을 가능하게 할 수도 있는 기법이다. 일부 비디오 코딩 기법들에서, 비디오 코더들은, 현재 비디오 블록의 예측을 위해 동일한 화상 내의 비디오 데이터의 현재 블록의 바로 상부 또는 하부에 있거나 또는 직접 수평적으로 라인을 이루는 이전에 재구성된 비디오 데이터의 블록들을 이용할 수도 있다. 즉, 비디오 데이터의 화상 또는 프레임이 2-D 그리드 상에 부과되면, 비디오 데이터의 각각의 블록은 x 값 및 y 값들의 고유 범위를 차지한다. 따라서, 일부 비디오 코더들은, x 값들의 동일한 (즉, 현재 비디오 블록과 수직으로 라인을 이루는) 세트, 또는 y 값들의 동일한 (즉, 현재 비디오 블록과 수평적으로 라인을 이루는) 세트만을 공유하는 동일한 화상에서 이전에 코딩된 비디오 데이터의 블록들에 기초하여 비디오 데이터의 현재 블록을 예측할 수도 있다.
- [0023] 비디오 코더는, 반드시 비디오 데이터의 현재 블록의 바로 상부 또는 좌측 (또는 바로 우측 또는 하부)에 있을 필요가 있는 것은 아닌 동일한 프레임, 즉 동일한 화상 내의 비디오 데이터의 이전에 재구성된 블록으로부터 현재 비디오 블록을 예측하는 것이 유리할 수도 있다. 예측 세트에 더 많은 비디오 블록들을 포함시킴으로써, 비디오 코더는 현재 비디오 블록의 더 정확한 예측을 달성함으로써, 코딩 효율을 증가시킬 수도 있다.
- [0024] 일반적으로, 본 개시는, IntraBC 또는 IntraMC 모드로서 지칭될 수도 있는 동일한 화상 내에 비디오 데이터의 예측 블록들로부터 비디오 데이터의 블록들의 인트라 예측을 위한 모드를 포함하는 비디오 데이터를 코딩하기 위한 기법들을 설명한다. 본 개시의 IntraBC 또는 IntraMC 기법들은 비디오 데이터의 현재 블록을 위한 비디오 데이터의 예측 블록을 식별하는 단계를 포함할 수도 있다. 비디오 데이터의 예측 블록은, 예를 들어, 비디오 데이터의 현재 블록과 동일한 화상 내의 비디오 데이터의 재구성된 블록에 대응할 수도 있다. 비디오 데이터의 예측 블록은, 화상 내에 있는 의도된 영역내에서, 예를 들어, 비디오 데이터의 현재 블록의 상부, 상부 우측, 상부 좌측, 및/또는 좌측 구역내에서 기인할 수도 있다. 비디오 데이터의 예측 블록은 현재 비디오 블록의 바로 상부 또는 바로 좌측 중 어느 한쪽에 있는 것으로 한정되지 않고, 결과적으로 현재 블록에 대하여 예측 블록을 식별하는데 이용된 벡터는 반드시 1차원 벡터인 것은 아니다. 대신에, 비디오 데이터의 예측 블록을 식별 또는 결정하기 위하여, 비디오 코더는, 비디오 데이터의 현재 블록에 대해 수평 변위 성분 및 수직 변위 성분을 포함하는 2차원 벡터를 정의하는 하나 이상의 선택스 엘리먼트들을 코딩할 수도 있다. 2차원 벡터는 블록 벡터, 오프셋 벡터, 또는 모션 벡터로 지칭될 수도 있고, 예를 들어, 현재 블록의 상단 좌측 코너에 관하여 예측 블록을 식별하는데 이용될 수도 있다.
- [0025] 아래에서 더 자세하게 설명될 바처럼, 코딩 유닛들로 지칭되는 비디오 데이터의 블록들은 다수의 예측 유닛들을 포함할 수도 있다. 각각의 예측 유닛은 따로 예측될 수도 있다. 따라서, 각각의 예측 유닛에 대해 블록 벡터는 비트스트림에서 시그널링될 수도 있고, 비디오 디코더는 예측 유닛의 한 루마 성분 및 예측 유닛의 2개 크로마 성분들을 예측하기 위하여 그 블록 벡터를 이용할 수도 있다. 비디오 데이터의 크로마 서브샘플링 포맷에 따라, 비디오 디코더는 예측 유닛의 2개 크로마 성분들을 예측하기 위하여 블록 벡터를 스케일링할 수도 있다. 크로마 성분들을 예측하기 위한 블록 벡터의 스케일링은 가독성을 돕기 위하여 다수회 반복되지 않는



다. 크로마 성분들을 예측하기 위한 그러한 블록 벡터의 스케일링은 크로마 성분들을 예측할 때 일어날 수도 있지만, 그러한 블록 벡터의 스케일링은 크로마 성분들을 예측하는 모든 경우에 필요한 것은 아닐 수도 있다는 것이 이해되어야 한다.

[0026] 기존 기법들에 따라, 상이한 블록 벡터는 코딩 유닛의 각각의 예측 유닛을 위한 참조 블록들을 식별하는데 이용될 수도 있다. 예를 들어, 코딩 유닛의 제 1 예측 유닛을 위한 제 1 블록 벡터는 제 1 예측 유닛의 루마 성분들을 위한 루마 참조 블록 (즉, 루마 샘플들의 예측 블록) 을 식별하는데 이용될 수도 있고, 제 1 예측 유닛의 2개의 크로마 성분들을 위한 크로마 참조 블록들 (즉, 크로마 샘플들의 2개의 대응하는 크로마 블록들) 을 식별하는데 이용될 수도 있다. 코딩 유닛의 제 2 예측 유닛을 위한 제 2 블록 벡터는 제 2 예측 유닛의 루마 성분들을 위한 루마 참조 블록 (즉, 루마 샘플들의 예측 블록) 을 식별하는데 이용될 수도 있고, 제 2 예측 유닛의 2개의 크로마 성분들을 위한 크로마 참조 블록들 (즉, 크로마 샘플들의 2개의 대응하는 크로마 블록들) 을 식별하는데 이용될 수도 있고 기타 등등이다.

[0027] 하지만, 본 개시의 기법들에 따르면, 상이한 블록 벡터들이 코딩 유닛의 각각의 예측 유닛의 루마 성분들을 예측하는데 이용될 수도 있는 한편, 동일한 블록 벡터는 코딩 유닛의 예측 유닛들의 2개 이상의 크로마 성분들을 예측하는데 이용될 수도 있다. 예를 들어, 위에서 처럼, 코딩 유닛의 제 1 예측 유닛을 위한 제 1 블록 벡터는 제 1 예측 유닛의 루마 성분들을 위한 루마 참조 블록 (즉, 루마 샘플들의 예측 블록) 을 식별하는데 이용될 수도 있고, 제 1 예측 유닛의 2개의 크로마 성분들을 위한 크로마 참조 블록들 (즉, 크로마 샘플들의 2개의 대응하는 크로마 블록들) 을 식별하는데 이용될 수도 있다. 하지만, 제 2 예측 유닛의 2개 크로마 성분들에 대해, (위의 예에서 처럼) 제 2 블록 벡터를 이용하기 보다는, 동일한 제 1 블록 벡터가 크로마 참조 블록들을 식별하는데 이용된다. 제 2 블록 벡터는 여전히, 제 2 예측 유닛의 루마 성분들을 위한 루마 참조 블록을 식별하는데 이용될 수도 있다.

[0028] 보다 구체적으로, 본 개시는 화상 또는 슬라이스 내에 예측된 블록들로 블록들의 코딩을 지원하기 위한 기법들을 설명한다. 제안된 기법들은, 예를 들어, 부상하고 있는 HEVC 범위 확장들 (RCEx) 과 함께 이용될 수도 있는데, 이는 높은 비트 깊이 (예를 들어, 8 비트 초과) 및 높은 크로마 샘플링 포맷 (예를 들어, 4:4:4 및 4:2:2 를 포함) 을 지원할 수도 있다. 본 개시의 기법들은 또한 스크린 콘텐츠 코딩 (screen content coding) 에 적용될 수도 있다.

[0029] 여기서 이용된, 용어 "비디오 코더" 는 총칭적으로 비디오 인코더 및 비디오 디코더 양자 모두를 지칭한다. 본 개시에서, 용어 "비디오 코딩" 또는 "코딩" 은 총칭적으로 비디오 인코딩 또는 비디오 디코딩을 지칭할 수도 있다. 따라서, 다르게 언급되지 않으면, 코딩과 관련하여 설명된 기법들은 비디오 인코더 또는 비디오 디코더 중 어느 하나에 의해 수행될 수도 있다는 것이 가정되어야 한다. 본원의 일부에서, 어떤 기법들은 비디오 디코딩 또는 비디오 디코더와 관련하여 설명될 수도 있다. 하지만, 그러한 기법들은 비디오 인코딩에 적용가능하지 않거나 또는 비디오 인코더에 의해 수행되지 않을 수도 있다고 가정되어서는 안된다. 예를 들어, 그러한 기법들은 비디오 데이터를 어떻게 인코딩하는지를 결정하는 부분으로서 수행될 수도 있거나 또는 비디오 인코더에서 비디오 디코딩 루프의 부분으로서 수행될 수도 있다.

[0030] 본 개시에서 이용된 바처럼, 용어 현재 블록은, 이미 코딩되었거나 또는 아직 코딩되어야 하는 블록과는 대조적으로, 현재 코딩되고 있는 블록을 지칭한다. 마찬가지로, 현재 코딩 유닛, 예측 유닛, 또는 변환 유닛은, 현재 코딩되고 있는 코딩 유닛, 예측 유닛, 또는 변환 유닛을 지칭한다.

[0031] 도 1은 본원에 기재된 기법들 중의 하나 이상을 구현할 수도 있는 일 예의 비디오 인코딩 및 디코딩 시스템을 예시하는 블록도이다. 도 1에 도시된 바처럼, 시스템 (10) 은, 목적지 디바이스 (14) 에 의해 디코딩될 인코딩된 비디오 데이터를 제공하는 소스 디바이스 (12) 를 포함한다. 특히, 소스 디바이스 (12) 는, 컴퓨터 판독가능 매체 (16) 를 통해 목적지 디바이스 (14) 로 비디오 데이터를 제공한다. 소스 디바이스 (12) 및 목적지 디바이스 (14) 는, 데스크탑 컴퓨터들, 노트북 (즉, 랩톱) 컴퓨터들, 태블릿 컴퓨터들, 셋톱 박스들, 전화기 핸드셋 이블테면 소위 "스마트" 폰들, 소위 "스마트" 패드, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 재생기들, 비디오 게이밍 콘솔들, 비디오 스트리밍 디바이스 등을 포함한, 광범위한 디바이스들 중 어느 것을 포함할 수도 있다. 일부 경우들에서, 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 무선 통신을 위해 갖추어질 수도 있다.

[0032] 목적지 디바이스 (14) 는, 컴퓨터 판독가능 매체 (16) 를 통해 디코딩될 인코딩된 비디오 데이터를 수신할 수도 있다. 컴퓨터 판독가능 매체 (16) 는, 인코딩된 비디오 데이터를 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 이동시킬 수 있는 임의의 타입의 매체 또는 디바이스를 포함할 수도 있다. 일 예에서, 컴퓨터

판독가능 매체 (16) 는, 소스 디바이스 (12) 로 하여금 실시간으로 직접 목적지 디바이스 (14) 로, 인코딩된 비디오 데이터를 송신할 수 있게 하기 위한 통신 매체를 포함할 수도 있다. 인코딩된 비디오 데이터는, 무선 통신 프로토콜 등의 통신 표준에 따라 변조되고, 목적지 디바이스 (14) 로 송신될 수도 있다. 통신 매체는 임의의 무선 또는 유선 통신 매체, 이를테면, 무선 주파수 (RF) 스펙트럼 또는 하나 이상의 물리적 송신 라인들을 포함할 수도 있다. 통신 매체는, 로컬 영역 네트워크, 와이드 영역 네트워크, 또는 인터넷과 같은 글로벌 네트워크 등의 패킷 기반 네트워크의 부분을 형성할 수도 있다. 통신 매체는 라우터, 스위치, 기지국, 또는 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 통신을 가능하게 하는데 유용할 수도 있는 임의의 다른 장비를 포함할 수도 있다.

[0033] 일부 예들에서, 컴퓨터 판독가능 매체 (16) 는 저장 디바이스를 포함할 수도 있고, 소스 디바이스는 인코딩된 비디오 데이터를 출력 인터페이스 (22) 를 통해 저장 디바이스에 출력할 수도 있다. 유사하게, 인코딩된 데이터는 입력 인터페이스 (28) 에 의해 저장 디바이스로부터 액세스될 수도 있다. 저장 디바이스는, 하드 드라이브, 블루레이 디스크, DVD, CD-ROM, 플래시 메모리, 휘발성 또는 비휘발성 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 임의의 다른 적합한 디지털 저장 매체 등의 다양한 분산형 또는 로컬적으로 액세스되는 데이터 저장 매체 중 어느 것을 포함할 수도 있다. 예를 들어, 디스크 스탬핑 설비 등의 매체 제조 설비의 컴퓨팅 디바이스는, 소스 디바이스 (12) 로부터 인코딩된 비디오 데이터를 수신하고 그 인코딩된 비디오 데이터를 포함하는 디스크를 제조할 수도 있다. 그러므로, 컴퓨터 판독가능 매체 (16) 는, 다양한 예들에서, 다양한 형태들의 하나 이상의 컴퓨터 판독가능 매체를 포함하는 것으로 이해될 수도 있다. 다른 예에서, 저장 디바이스는, 소스 디바이스 (12) 에 의해 생성되는 인코딩된 비디오를 저장할 수도 있는, 파일 서버 또는 또 다른 중간 저장 디바이스에 대응할 수도 있다. 목적지 디바이스 (14) 는, 스트리밍 또는 다운로드를 통해 저장 디바이스로부터 저장된 비디오 데이터에 액세스할 수도 있다. 파일 서버는, 인코딩된 비디오 데이터를 저장하고 그 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로 송신할 수 있는 임의의 타입의 서버일 수도 있다. 예의 파일 서버들은, (예를 들어, 웹사이트용) 웹 서버, FTP 서버, NAS (network attached storage) 디바이스, 또는 로컬 디스크 드라이브를 포함한다. 목적지 디바이스 (14) 는, 인터넷 접속을 포함한, 임의의 표준 데이터 접속을 통해 인코딩된 비디오 데이터에 액세스할 수도 있다. 이것은, 파일 서버 상에 저장된 인코딩된 비디오 데이터에 액세스하는데 적합한 무선 채널 (예를 들어, Wi-Fi 접속), 유선 접속 (예를 들어, DSL, 케이블 모뎀 등), 또는 양자 모두의 조합을 포함할 수도 있다. 저장 디바이스로부터 인코딩된 비디오 데이터의 송신은, 스트리밍 송신, 다운로드 송신 또는 이들의 조합일 수도 있다.

[0034] 본 개시의 기법들은 무선 애플리케이션들 또는 세팅들에 반드시 한정되는 것은 아니다. 그 기법들은, 공중 경유 (over-the-air) 텔레비전 브로드캐스트, 케이블 텔레비전 송신, 위성 텔레비전 송신, DASH (dynamic adaptive streaming over HTTP) 와 같은 인터넷 스트리밍 비디오 송신, 데이터 저장 매체 상에 인코딩되는 디지털 비디오, 데이터 저장 매체 상에 저장된 디지털 비디오의 디코딩, 또는 다른 애플리케이션 등의 다양한 멀티미디어 애플리케이션들 중 어느 것을 지원하는 비디오 코딩에 적용될 수도 있다. 일부 예들에서, 시스템 (10) 은, 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅 및/또는 화상 통화등의 애플리케이션들을 지원하기 위하여 일방향 또는 양방향 비디오 송신을 지원하도록 구성될 수도 있다.

[0035] 도 1은 예일뿐이고 본 개시의 기법들은 인코딩과 디코딩 디바이스들 간의 임의의 데이터 통신을 반드시 포함하는 것은 아닌 비디오 코딩 세팅들 (예를 들어, 비디오 인코딩 또는 비디오 디코딩) 에 적용될 수도 있다. 다른 예들에서, 데이터는 로컬 메모리로부터 추출되거나, 네트워크 상에서 스트리밍되는 등이다. 비디오 인코딩 디바이스는 데이터를 메모리로 인코딩 및 저장할 수도 있거나, 및/또는 비디오 디코딩 디바이스는 메모리로부터 데이터를 추출 및 디코딩할 수도 있다. 많은 예들에서, 인코딩 및 디코딩은, 서로 통신하는 것이 아니라, 단순히 메모리에 데이터를 인코딩하거나 및/또는 메모리로부터 데이터를 추출 및 디코딩하는 디바이스들에 의해 수행된다.

[0036] 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. 다른 예들에서, 소스 디바이스 및 목적지 디바이스는 다른 컴포넌트들 또는 배열 (arrangement) 들을 포함할 수도 있다. 예를 들어, 소스 디바이스 (12) 는 외부 카메라와 같은 외부 비디오 소스 (18) 로부터 비디오 데이터를 수신할 수도 있다. 마찬가지로, 목적지 디바이스 (14) 는 통합된 디스플레이 디바이스 (32) 를 포함하기 보다는, 외부 디스플레이 디바이스 (32) 와 인터페이스 접속할 수도 있다.

[0037] 도 1의 예시된 시스템 (10) 은 하나의 예일 뿐이다. 임의의 디지털 비디오 인코딩 및/또는 디코딩 디바이스가 본원에 기재된 기법들을 수행할 수도 있다. 일반적으로 본 개시의 기법들은 비디오 인코딩 디바이스에



의해 수행되지만, 그 기법들은 또한 "코덱 (CODEC)" 으로서 통상적으로 지칭되는, 비디오 인코더/디코더에 의해 수행될 수도 있다. 더욱이, 본 개시의 기법들은 또한 비디오 프리프로세서에 의해 수행될 수도 있다. 소스 디바이스 (12) 및 목적지 디바이스 (14) 는, 소스 디바이스 (12) 가 목적지 디바이스 (14) 로의 송신을 위해 코딩된 비디오 데이터를 생성하는 그러한 코딩 디바이스들의 예들일 뿐이다. 몇몇 예들에서, 디바이스들 (12, 14) 은, 디바이스들 (12, 14) 의 각각이 비디오 인코딩 및 디코딩 컴포넌트들을 포함하도록 실질적으로 대칭적인 방식으로 동작할 수도 있다. 그러므로, 시스템 (10) 은 예를 들면, 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅 또는 화상 통화를 위해, 비디오 디바이스들 (12, 14) 간의 일방향 또는 양방향 비디오 송신을 지원할 수도 있다.

[0038] 소스 디바이스 (12) 의 비디오 소스 (18) 는 비디오 카메라와 같은 비디오 캡처 디바이스, 이전에 캡처된 비디오를 포함하는 비디오 아카이브 (video archive), 및/또는 비디오 콘텐츠 제공자로부터 비디오를 수신하기 위한 비디오 피드 인터페이스 (video feed interface) 를 포함할 수도 있다. 다른 대안으로서, 비디오 소스 (18) 는 라이브 비디오, 보관된 비디오 및 컴퓨터 생성된 비디오의 조합, 또는 소스 비디오로서 컴퓨터 그래픽스 기반 데이터를 생성할 수도 있다. 일부 경우들에서, 비디오 소스 (18) 가 비디오 카메라이면, 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 소위 카메라 폰들 또는 비디오 폰들을 형성할 수도 있다. 하지만, 위에서 언급된 바처럼, 본 개시에 설명된 기법들은, 일반적으로 비디오 코딩에 적용가능할 수도 있고, 무선 및/또는 유선 애플리케이션들에 적용될 수도 있다. 각 경우에서, 캡처되거나, 미리 캡처되거나, 또는 컴퓨터 생성된 비디오는 비디오 인코더 (20) 에 의해 인코딩될 수도 있다. 다음으로, 인딩된 비디오 정보는 컴퓨터 관독가능 매체 (16) 상으로 출력 인터페이스 (22) 에 의해 출력될 수도 있다. 일부 예들에서, 출력 인터페이스 (22) 는 변조기/복조기 (모뎀) 및/또는 송신기를 포함할 수도 있다.

[0039] 목적지 디바이스 (14) 의 입력 인터페이스 (28) 는 컴퓨터 관독가능 매체 (16) 로부터 정보를 수신한다. 일부 예들에서, 입력 인터페이스 (28) 는 수신기 및/또는 모뎀을 포함한다. 컴퓨터 관독가능 매체 (16) 의 정보는, 블록들 및 다른 코딩된 유닛들, 예를 들어, 슬라이스들, 화상들, 화상들의 그룹 (GOP) 들 또는 비디오 데이터의 프로세싱 및/또는 특징을 기술하고, 비디오 디코더 (30) 에 의해 이용되는, 인코딩된 비디오 데이터 및 비디오 인코더 (20) 에 의해 정의된 다른 선택 정보들을 포함할 수도 있는, 인코딩된 비디오 비트스트림을 포함할 수도 있다. 디스플레이 디바이스 (32) 는 디코딩된 비디오 데이터를 이용자에게 표시하고, 음극선관 (CRT), 액정 디스플레이 (LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (OLED) 디스플레이, 또는 다른 타입의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들 중 어느 것을 포함할 수도 있다.

[0040] 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서 (DSP) 들, 주문형 반도체 (ASIC), 필드 프로그램가능 게이트 어레이 (FPGA), 이산 로직, 소프트웨어, 하드웨어, 펌웨어 또는 이들의 임의의 조합과 같은 다양한 적합한 인코더 회로 중 어느 것으로서 구현될 수도 있다. 그 기법들이 부분적으로 소프트웨어로 구현될 때, 디바이스는 적합한 비일시적 컴퓨터 관독가능 매체에 그 소프트웨어를 위한 명령들을 저장하고 본 개시의 기법들을 수행하기 위하여 하나 이상의 프로세서들을 이용하여 하드웨어에서 그 명령들을 실행할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 의 각각은 하나 이상의 인코더들 또는 디코더들에 포함될 수도 있는데, 이들 중 어느 쪽이 각각의 디바이스에서 결합된 인코더/디코더 (CODEC) 의 부분으로서 통합될 수도 있다. 비록 도 1에 도시되지는 않았지만, 일부 양태들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 각각 오디오 인코더 및 디코더와 통합될 수도 있고, 적절한 MUX-DEMUX 유닛들 또는 다른 하드웨어 및 소프트웨어를 포함하여, 공통 데이터 스트림 또는 분리된 데이터 스트림들에서 오디오 및 비디오 양자 모두의 인코딩을 핸들링 (handling) 할 수도 있다. 적용가능하다면, MUX-DEMUX 유닛들은 ITU H.223 멀티플렉서 프로토콜 또는 다른 프로토콜들 이를테면 사용자 데이터그램 프로토콜 (UDP) 을 따를 수도 있다.

[0041] 본 개시는 일반적으로 어떤 정보를 또 다른 디바이스, 이를테면 비디오 디코더 (30) 로 "시그널링" 하는 비디오 인코더 (20) 와 관련될 수도 있다. 용어 "시그널링" 은 일반적으로, 압축된 비디오 데이터를 디코딩하는데 이용되는 선택스 엘리먼트들 및/또는 다른 데이터의 통신을 지칭할 수도 있다. 그러한 통신은 실시간 또는 근실시간으로 일어날 수도 있다. 다르게는, 그러한 통신은, 인코딩 시에 인코딩된 비트스트림에서 컴퓨터 관독가능 저장 매체에 선택스 엘리먼트들을 저장할 때 일어날 수도 있고, 다음으로 이 매체에 저장된 후에 임의의 시간에 디코딩 디바이스에 의해 취출될 수도 있는 바처럼 시간 기간에 걸쳐 일어날 수도 있다.

[0042] 비디오 인코더 (20) 및 비디오 디코더 (30) 는, 하나 이상의 비디오 코딩 표준들에 따라 동작하도록 구성되는 것으로서, 예시의 목적을 위해, 본 개시에 설명되어 있다. 하지만, 본 개시의 기법들은 반드시 어느 특정

코딩 표준에 한정될 필요는 없고, 다양한 상이한 코딩 표준들을 위해 적용될 수도 있다.

[0043] 비디오 인코더 (20) 및 비디오 디코더 (30) 는 비디오 코딩 표준, 이를테면, 임의의 확장, 수정 또는 추가를 포함하는 현재 개발중인 HEVC (High Efficiency Video Coding) 표준에 따라 동작할 수도 있고, HEVC 테스트 모델 (HM) 에 따를 수도 있다. 다르게는, 비디오 인코더 (20) 및 비디오 디코더 (30) 는, 다르게는 MPEG4, Part 10, AVC (Advanced Video Coding) 으로서도 지칭되는, ITU-T H.264 표준과 같은 다른 사유 (proprietary) 또는 산업 표준들 또는 그러한 표준들의 확장들에 따라 동작할 수도 있다. 하지만, 본 개시의 기법들은 임의의 특정 코딩 표준에 한정되지 않는다. 비디오 코딩 표준들의 다른 예들은 MPEG-2 및 ITU-T H.263 를 포함한다.

[0044] 다른 특허 또는 산업 표준들의 예들은 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 MPEG-4 AVC 으로서도 알려짐) 을 포함하며, 그의 SVC (Scalable Video Coding) 및 MVC (Multiview Video Coding) 확장들, 또는 그러한 표준들의 확장, 수정 또는 추가들을 포함한다. 게다가, H.264/AVC 에 대한 3차원 비디오 (3DV) 코딩 확장, 즉 AVC-기반 3DV 를 생성하기 위한 노력이 진행되고 있다. 즉, 예시적인 비디오 코딩 표준들은 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 MPEG-4 AVC 으로서도 알려짐) 을 포함하며, 그의 SVC (Scalable Video Coding) 및 MVC (Multi-view Video Coding) 확장들을 포함한다. H.264 표준은 ITU-T 연구 그룹에 의한 2005년 3월자의, ITU-T Recommendation H.264, Advanced Video Coding for generic audiovisual services 에 기술되어 있는데, 이는 여기서 H.264 표준 또는 H.264 규격 (specification), 또는 H.264/AVC 표준 또는 규격으로 지칭될 수도 있다. JVT (Joint Video Team) 는 H.264/MPEG-4 AVC 에 대한 확장들에 대해 계속 작업하고 있다. MVC 의 최근 합동 초안이 “Advanced video coding for generic audiovisual services,” ITU-T Recommendation H.264 (2010년 3월) 에 기재되어 있다.

[0045] 또한, ITU-T VCEG (Video Coding Experts Group) 및 ISO/IEC MPEG (Motion Picture Experts Group) 의 JCT-VC (Joint Collaboration Team on Video Coding) 에 의해 개발된, 새로 개발된 비디오 코딩 표준, 즉 HEVC (High-Efficiency Video Coding) 이 있다. 도 1의 예에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 HEVC 에 따라 동작할 수도 있다. 2013 년 10월 17일자로 [http://phenix.int-evry.fr/jct/doc\\_end\\_user/documents/12\\_Geneva/wg11/JCTVC-L1003-v34.zip](http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v34.zip) 로 이용가능하고 전체 내용이 참조에 의해 인용되는, Bross 등의 “고효율 비디오 코딩 (HEVC) 텍스트 규격 초안 10 (FDIS & Last Call 용) [ITU-T SG 16 WP 3 및 ISO/IEC JTC 1/SC 29/WG 11 의 JCT-VC (Joint Collaborative Team on Video Coding), 12차 미팅: 스위스, 제네바, 2013 년 1월 14-23일] 문헌 JCTVC-L1003\_v34 (이하 “JCTVC-L1003-v34”) 이 HEVC 의 최근의 초안이다. JCTVC-L1003-v34 에 정의된 HEVC 의 버전은 HEVC 버전 1 또는 “HEVC v1” 으로 지칭될 수도 있다. [http://phenix.int-evry.fr/jct/doc\\_end\\_user/current\\_document.php?id=8143](http://phenix.int-evry.fr/jct/doc_end_user/current_document.php?id=8143) 로부터 이용가능하고 전부 내용이 참조에 의해 인용되는 McCann 등의 “고효율 비디오 코딩 (HEVC) 테스트 모델 12 (HM 12) 인코더 기술,” 문헌 JCTVC-N1002 이 HEVC 의 최근의 인코더 기술 (description) 이다.

[0046] 3D 서비스들을 지원하는 HEVC 의 2개 확장들은, ITU-T VCEG (Video Coding Experts Group) 및 ISO/IEC MPEG (Motion Picture Experts Group) 의 JCT-3V (Joint Collaboration Team on 3D Video coding) 에 의해 개발되었다. 2개 확장들은 즉, 각각 MV-HEVC 및 3D-HEVC 이다. 도 1의 예에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 MV-HEVC 및/또는 3D-HEVC 에 따라 동작할 수도 있다.

[0047] MV-HEVC 는 블록 레벨 설계를 변경함이 없이 멀티 (텍스처) 뷰들의 코딩을 지원한다. [http://phenix.it-sudparis.eu/jct2/doc\\_end\\_user/documents/5\\_Vienna/wg11/JCT3V-E1004-v6.zip](http://phenix.it-sudparis.eu/jct2/doc_end_user/documents/5_Vienna/wg11/JCT3V-E1004-v6.zip)로부터 이용가능하고 전체 내용이 참조에 의해 인용되는 Tech 등의, “MV-HEVC 초안 텍스트 5,” [ITU-T SG 16 WP 3 및 ISO/IEC JTC 1/SC 29/WG 11 의 3D 비디오 코딩 확장 개발에 관한 공동 협력 팀] 문헌 번호. JCT3V-E1004-v6 이, MV-HEVC 의 최근 초안이다.

[0048] 3D-HEVC 는 멀티뷰 비디오 플러스 깊이 포맷을 코딩하고 HEVC 코딩 모듈들에 추가로 구축된 새로운 코딩 툴들을 포함한다. 새로 도입된 코딩 툴들은 텍스처 코딩 및 깊이 코딩 양자 모두에 적용가능하다. [http://phenix.it-sudparis.eu/jct2/doc\\_end\\_user/documents/5\\_Vienna/wg11/JCT3V-E1001-v3.zip](http://phenix.it-sudparis.eu/jct2/doc_end_user/documents/5_Vienna/wg11/JCT3V-E1001-v3.zip) 로부터 이용가능하고 전체 내용이 참조에 의해 인용되는 Tech 등의, “3D-JEVC 초안 텍스트 1,” [ITU-T SG 16 WP 3 및 ISO/IEC JTC 1/SC 29/WG 11 의 3D 비디오 코딩 확장 개발에 관한 공동 협력 팀, 5차 미팅: 오스트리아, 비엔나, 2013년 7월 27일 - 8월 2일] 문헌 번호. JCT3V-E1001-v3 는, 3D-HEVC 의 최근 초안이다. 3D-HEVC

를 위한 최근의 소프트웨어 3D-HTM 은 하기 링크로부터 다운로드될 수 있다:

- [0049] [3D-HTM 버전 8.0]: [https://hevc.hhi.fraunhofer.de/svn/svn\\_3DVCSoftware/tags/HTM-8.0/](https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-8.0/). 전체 내용이 참조에 의해 인용되는, 최근의 소프트웨어 기술, Zhang 등의, "3D-HEVC 테스트 모델 5," [ITU-T SG 16 WP 3 및 ISO/IEC JTC 1/SC 29/WG 11 의 3D 비디오 코딩 확장 개발에 관한 공동 협력 팀, 5차 미팅: 오스트리아, 비엔나, 2013년 7월 27일 - 8월 2일] 문헌 번호: JCT3V-E1005 는, [http://phenix.it-sudparis.eu/jct2/doc\\_end\\_user/current\\_document.php?id=1360](http://phenix.it-sudparis.eu/jct2/doc_end_user/current_document.php?id=1360)로부터 이용가능하다.
- [0050] HEVC 표준의 추가 개발 및 확장들은 HEVC 범위 확장들을 포함한다. HEVC 범위 확장들의 일 예는, Flynn 등의 "고효율 비디오 코딩 (HEVC) 범위 확장 텍스트 규격: 초안 4", [ITU-T SG 16 WP 3 및 ISO/IEC JTC 1/SC 29/WG 11 의 JCT-VC (Joint Collaborative Team on Video Coding), 문헌 JCTVC-N1005\_v3, 13차 미팅: 한국, 인천, 2013년 4월 18-26] 에 기재되어 있고 이는 참조에 의해 본원에 전부 인용되고 다음에서 이용가능하다: [http://phenix.int-evry.fr/jct/doc\\_end\\_user/current\\_document.php?id=8139](http://phenix.int-evry.fr/jct/doc_end_user/current_document.php?id=8139).
- [0051] HEVC 표준의 추가 개발 및 확장들은 스크린 콘텐츠 코딩 (SCC) 확장들을 포함한다.
- [0052] HEVC 및 다른 비디오 코딩 규격들에서, 비디오 시퀀스는 통상적으로 일련의 화상 (picture) 들을 포함한다. 화상들은 또한, "프레임" 들로 지칭될 수도 있다. 화상은,  $S_L$ ,  $S_{Cb}$  및  $S_{Cr}$  로 표기되는, 3개의 샘플 어레이들을 포함할 수도 있다.  $S_L$  는 루마 샘플들의 2차원 어레이 (즉, 블록) 이다.  $S_{Cb}$  는 Cb 크로미넌스 샘플들의 2차원 어레이이다.  $S_{Cr}$  는 Cr 크로미넌스 샘플들의 2차원 어레이이다. 크로미넌스 샘플들은 또한, 본원에서 "크로마" 샘플들로 지칭될 수도 있다. 다른 사례들에서, 화상은 단색 (monochrome) 일 수도 있고, 루마 샘플들의 어레이만을 포함할 수도 있다.
- [0053] HEVC 표준화 노력들은 HEVC 테스트 모델 (HM) 로 지칭되는 비디오 코딩 디바이스의 진화 모델에 기초한다. HM 은 예컨대, ITU-T H.264/AVC 에 따른 기존 디바이스들에 비하여 비디오 코딩 디바이스들의 여러 추가적인 능력들을 상정한다. 예를 들어, H.264 는 9개의 인트라 예측 인코딩 모드들을 제공하지만, HM 는 무려 33개의 인트라 예측 인코딩 모드들을 제공할 수도 있다.
- [0054] 일반적으로, HM 의 작업 모델은 비디오 인코더 (20) 가, 최대 코딩 유닛 (LCU) 또는 트리블록들로도 지칭되는 코딩 트리 유닛 (CTU) 들의 시퀀스로 비디오 프레임 또는 화상을 분할할 수도 있다는 것을 기술한다. CTU 들의 각각은 루마 샘플들의 코딩 트리 블록 (CTB), 크로마 샘플들의 2개의 대응하는 코딩 트리 블록들, 및 코딩 트리 블록들의 샘플들을 코딩하는데 이용된 선택 구조들을 포함할 수도 있다. 단색 화상 또는 분리된 색 평면들을 포함하는 화상에서, CTU 는 단일 코딩 트리 블록 및 그 단일 코딩 트리 블록의 샘플들을 코딩하는데 이용된 선택 구조들을 포함할 수도 있다.
- [0055] 비트스트림 내의 선택 데이터는 CTU 를 위한 크기를 정의할 수도 있으며, 이는 픽셀들의 수의 측면에서 최대 코딩 유닛이다. CTU 는 샘플들의  $N \times N$  블록일 수도 있다. HEVC 의 CTU 들은 대체로, H.264/AVC 와 같은 다른 비디오 코딩 표준들의 매크로블록들에 유사할 수도 있다. 하지만, CTU 가 특정 크기로 반드시 한정되는 것은 아니고 하나 이상의 코딩 유닛들 (CU) 들을 포함할 수도 있다.
- [0056] 비디오 프레임 또는 화상은, 하나 이상의 슬라이스들로 파티셔닝될 수도 있다. 슬라이스는, 코딩 또는 스캐닝 순서에서 다수의 연속적인 CTU 들을 포함한다. 각각의 CTU 는 쿼드트리에 따라 코딩 유닛 (CU) 들로 스플릿될 수도 있다. CU 는 샘플들의  $N \times N$  블록일 수도 있다. 일반적으로, 쿼드트리 데이터 구조는, 트리블록에 대응하는 루트 노드와, CU 당 하나의 노드를 포함한다. CU 가 4 개의 서브 CU 들로 스플릿되는 경우, 그 CU 에 대응하는 노드는 4 개의 리프 노드들을 포함하고, 이들의 각각은 서브 CU 들 중 하나에 대응한다. CU 로도 지칭될 수도 있는, CU 또는 서브-CU 는 루마 샘플 어레이, Cb 샘플 어레이 및 Cr 샘플 어레이를 갖는 화상의 루마 샘플들의 코딩 블록 및 크로마 샘플들의 2개의 대응하는 코딩 블록들, 및 그 코딩 블록들의 샘플들을 코딩하는데 이용된 선택 구조를 포함할 수도 있다. 단색 화상 또는 분리된 색 평면들을 포함하는 화상에서, CU 는 단일 코딩 블록 및 그 단일 코딩 블록의 샘플들을 코딩하는데 이용된 선택 구조들을 포함할 수도 있다.
- [0057] 쿼드트리 데이터 구조의 각각의 노드는, 대응하는 CU 를 위한 선택 데이터를 제공할 수도 있다. 예를 들어, 쿼드트리에서의 노드는, 그 노드에 대응하는 CU 가 서브 CU 들로 스플릿되는지 여부를 표시하는, 스플릿 플래그 (split flag) 를 포함할 수도 있다. CU 를 위한 선택 엘리먼트들은 회귀적으로 정의될 수도 있고, CU 가 서브 CU 들로 스플릿되는지 여부에 의존할 수도 있다. CU가 더 스플릿되지 않으면, 그것은 리프 CU

(leaf-CU) 로 지칭된다. 본 개시에서, 리프 CU의 4개 서브 CU 들은 또한, 원래 리프 CU 의 명시적 스플리팅 (explicit splitting) 이 없더라도, 리프 CU들로 지칭될 수도 있다. 예를 들어, 16x16 크기의 CU 가 더 스플릿되지 않으면, 16x16 CU 가 스플릿되지 않았지만 4개의 8x8 서브 CU들이 또한 리프 CU 들로 지칭될 수도 있다. 코딩된 비트스트림과 연관된 선택스 데이터는, 최대 CU 깊이로도 지칭되는, 트리블록이 스플릿될 수도 있는 최대 횟수를 정의할 수도 있고, 또한 코딩 노드들의 최소 크기를 정의할 수도 있다. 따라서, 비트스트림은 또한 최소 코딩 유닛 (smallest coding unit; SCU) 을 정의할 수도 있다.

[0058] CU 는 코딩 노드 그리고 그 코딩 노드와 연관된 예측 유닛 (PU) 들 및 변환 유닛 (TU) 들을 포함한다. CU 의 크기는 코딩 노드의 크기에 대응하고 형상이 정사각형이어야 한다. CU 의 크기는 8x8 픽셀들로부터, 최대 64x64 픽셀들 이상인 트리블록의 크기에 이르기까지의 범위일 수도 있다. 일반적으로, 비디오 인코더 (20) 및 비디오 디코더 (30) 는, 하나의 모드, 예를 들어, 인트라 예측 또는 인터 예측으로 각각의 CU 를 코딩한다.

[0059] 각각의 CU 는 하나 이상의 PU 들 및 하나 이상의 TU 들을 포함할 수도 있다. CU 와 연관된 선택스 데이터는, 예를 들어, CU 를 하나 이상의 PU 들로 파티셔닝하는 것을 기술할 수도 있다. 파티셔닝 모드들은, CU 가 스킵 또는 직접 모드 인코딩되는지, 인트라 예측 모드 인코딩되는지, 또는 인터 예측 모드 인코딩되는지 간에 달라질 수도 있다. PU 들은 형상이 비정사각형 (예를 들어, 직사각형) 으로 파티셔닝될 수도 있다. CU 와 연관된 선택스 데이터는 또한, 예를 들어, 쿼드트리에 따라 CU 를 하나 이상의 TU 들로 파티셔닝하는 것을 기술할 수도 있다. TU 는 형상이 정사각형 또는 비정사각형 (예를 들어, 직사각형) 일 수 있다.

[0060] CU 의 PU 및 TU 들은 화상의 루마 샘플들의 예측 블록, 크로마 샘플들의 2개 대응하는 예측 블록들, 및 예측 블록 샘플들을 예측하는데 이용된 선택스 구조들을 포함할 수도 있다. 비디오 인코더 (20) 는, CU 의 각각의 PU 를 위해 예측 루마, Cb, 및 Cr 블록들을 생성할 수도 있다. 단색 화상 또는 분리된 색 평면들을 포함하는 화상에서, PU 는 단일 예측 블록 및 그 단일 예측 블록을 예측하는데 이용된 선택스 구조들을 포함할 수도 있다.

[0061] 비디오 인코더 (20) 및 비디오 디코더 (30) 는 다양한 크기를 갖는 PU들을 지원할 수도 있다. 특정 CU 의 크기가 2Nx2N 이라고 가정하면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 2Nx2N 또는 NxN 의 PU 크기들에서 인트라 예측, 그리고 2Nx2N, 2NxN, Nx2N, 또는 NxN 의 대칭적 PU 크기에서의 인터 예측을 지원할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 또한, 2NxnU, 2NxnD, nLx2N, 및 nRx2N 의 PU 크기에서의 인터 예측을 위한 비대칭적 파티셔닝을 지원할 수도 있다. 비대칭적 파티셔닝에서, CU 의 일 방향은 파티셔닝되지 않는 반면, 타 방향은 25% 및 75% 으로 파티셔닝된다. 25% 파티션에 대응하는 CU 의 부분은 “n” 다음에 “상 (Up)”, “하 (Down)”, “좌 (Left)”, 또는 “우 (Right)” 의 표시에 의해 표시된다. 따라서, 예를 들어, 2NxnU” 는, 상단의 2Nx0.5N PU 및 하단의 2Nx1.5N PU 로 수평적으로 파티셔닝되는 2Nx2N CU 를 지칭한다. HEVC 에서, 최소 PU 크기들은 8x4 및 4x8 이다.

[0062] 본 개시에서, “NxN” 그리고 “N 바이 N” 은, 수직 및 수평 차원 (dimension) 들의 면에서 비디오 블록의 픽셀 차원들, 예를 들면, 16x16 픽셀들 또는 16 바이 16 픽셀들을 지칭하는데 상호교환가능하게 이용될 수도 있다. 일반적으로, 16x16 블록은, 수직 방향에서 16 픽셀들 (y = 16) 그리고 수평 방향에서 16 픽셀들 (x = 16) 을 갖는다. 마찬가지로, NxN 블록은 일반적으로 수직 방향에서 N 픽셀들 그리고 수평 방향에서 N 픽셀들을 갖고, 여기서 N 은 음이 아닌 정수 값을 나타낸다. 블록에서 픽셀들은 행과 열들로 배열될 수도 있다. 더욱이, 블록들은 수직 방향과 동일한 수의 수평 방향 픽셀들을 반드시 가질 필요는 없다. 예를 들면, 블록들은 NxM 픽셀들을 포함할 수도 있고, 여기서 M은 N과 반드시 동일한 것은 아니다.

[0063] 일반적으로, PU 는, 대응하는 CU 의 전부 또는 일부에 대응하는 공간 구역을 나타내고, PU 를 위해 참조 샘플을 추출하기 위한 데이터를 포함할 수도 있다. 더욱이, PU 는 예측에 관한 데이터를 포함한다. 예를 들어, PU 가 인트라 모드 인코딩될 때, PU 를 위한 데이터는 PU 에 대응하는 예측 블록을 위한 인트라 예측 모드를 기술할 수도 있다.

[0064] 또 다른 예로서, PU 가 인터 모드 인코딩될 때, PU 는 PU 를 위한 하나 이상의 모션 벡터들을 정의하는 데이터를 포함할 수도 있다. PU 를 위한 모션 벡터를 정의하는 데이터는, 예를 들어, 모션 벡터의 수평 성분, 모션 벡터의 수직 성분, 모션 벡터의 해상도 (예를 들어, 1/4 픽셀 정밀도 또는 1/8 픽셀 정밀도) 를 기술할 수도 있다. PU 는 또한, 참조 화상을 포함하는 참조 화상 리스트 내에 인덱스와 같이 모션 벡터가 가리키는 참조 화상을 식별하는 데이터를 포함시킬 수도 있다.



- [0065] 인터 예측은 단방향 (즉, 단예측) 또는 양방향 (즉, 양예측) 일 수도 있다. 단예측 또는 양예측을 수행하기 위하여, 비디오 인코더 (20) 및 비디오 디코더는 제 1 참조 화상 리스트 (RefPicList0) 및 제 2 참조 화상 리스트 (RefPicList1) 를 현재 화상을 위해 생성할 수도 있다. 참조 화상 리스트들의 각각은 하나 이상의 참조 화상들을 포함할 수도 있다. 참조 화상 리스트 (즉, 이용가능하다면, RefPicList0 및 RefPicList1) 가 구성된 후, 참조 화상 리스트에 대한 참조 인덱스는 참조 화상 리스트에 포함된 임의의 참조 화상을 식별하는데 이용될 수 있다.
- [0066] 참조 화상은 시간적, 예를 들어, 디스플레이, 순서에서의 이전 화상, 향후 화상, 또는 2개 이상의 이전에 인코딩된 화상들로부터 예측들의 조합일 수도 있다. 비디오 코더들 예를 들어, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 화상의 시간 순서를 식별하기 위하여 화상 순서 카운트 (POC) 를 이용한다. 비디오 코더들은 또한, 참조 화상 리스트 구성 및 모션 벡터 스케일링을 위해 화상들의 POC 값들을 이용한다. 비디오 인코더 (20) 또는 비디오 디코더 (30) 는, 참조 화상 메모리, 예를 들어, 참조 화상 메모리 (368) (도 6) 및 참조 화상 메모리 (396) (도 7) 에 참조 화상들을 저장할 수도 있다.
- [0067] 현재 비디오 블록, 예를 들어, PU 가 본 개시의 기법들에 따라 IntraBC 를 이용하여 코딩될 때, 블록을 위한 (블록 벡터, 오프셋 벡터, 모션 벡터 또는 변위 벡터로 지칭될 수도 있는) 2차원 벡터를 정의하는 데이터는, 예를 들어, 벡터의 수평 성분, 벡터의 수직 성분, 벡터를 위한 해상도 (예를 들어, 1/4 픽셀 정밀도 또는 1/8 픽셀 정밀도) 를 기술할 수도 있다. 하지만, 본 개시의 기법들에 따라 IntraBC 를 이용하여 예측되는 PU 의 데이터는, 현재 비디오 블록과 동일한 프레임 또는 화상 내에 참조 블록이 있으므로, 모션 벡터가 가리키는 참조 화상을 식별할 필요가 없다.
- [0068] 비디오 인코더 (20) 가, CU 의 예측 루마, Cb, 및 Cr 블록들을 하나 이상의 PU 들을 위해 생성한 후에, 비디오 인코더 (20) 는 CU 를 위한 루마 잔차 블록을 생성할 수도 있다. CU 의 루마 잔차 블록에 있는 각각의 샘플은 CU 의 PU 의 예측 루마 블록에 있는 루마 샘플과 CU 의 원래 루마 코딩 블록에 있는 대응하는 샘플 사이의 차이를 표시한다. 또한, 비디오 인코더 (20) 는 CU 를 위한 Cb 잔차 블록을 생성할 수도 있다. CU 의 Cb 루마 잔차 블록에 있는 각각의 샘플은 CU 의 PU 의 예측 Cb 블록에 있는 Cb 샘플과 CU 의 원래 Cb 코딩 블록에 있는 대응하는 샘플 사이의 차이를 표시한다. 또한, 비디오 인코더 (20) 는 CU 를 위한 Cr 잔차 블록을 생성할 수도 있다. CU 의 Cr 루마 잔차 블록에 있는 각각의 샘플은 CU 의 PU 의 예측 Cr 블록에 있는 Cr 샘플과 CU 의 원래 Cr 코딩 블록에 있는 대응하는 샘플 사이의 차이를 표시한다.
- [0069] 하나 이상의 PU 들을 갖는 리프 CU 는 하나 이상의 변환 유닛 (TU) 들을 또한 포함할 수도 있다. CU 의 TU 는 루마 샘플들의 변환 블록, 크로마 샘플들의 2개 대응하는 변환 블록들, 및 변환 블록 샘플들을 변환하는데 이용된 신택스 구조들을 포함할 수도 있다. 비디오 인코더 (20) 는 양자화될 수도 있는 변환 계수들을 생성하기 위하여 TU 들과 연관된 잔차 블록 내의 픽셀 차이 값들을 변환할 수도 있다. 비디오 인코더 (20) 는, 예를 들어, 이산 코사인 변환 (DCT), 정수 변환, 웨이블릿 변환, 또는 개념적으로 유사한 변환을 잔차 비디오 데이터에 적용할 수도 있다.
- [0070] TU들은, 위에서 논의된 바처럼, (TU 쿼드트리 구조로도 지칭되는) RQT를 이용하여 명시될 수도 있다. 예를 들어, 스플릿된 플래그는 리프 CU 가 4개의 변환 유닛들로 스플릿되는지 여부를 나타낼 수도 있다. 다음으로, 각 변환 유닛은, 추가 서브 TU들로 더 스플릿될 수도 있다. TU가 더 스플릿되지 않을 때, 그것은 리프 TU 로 지칭될 수도 있다. TU 는 반드시 PU 의 크기로 제한되는 것은 아니다. 따라서, TU 는 PU 보다 더 크거나 더 작을 수도 있다. 인트라 코딩을 위해, PU 는 동일한 CU 에 대해 대응하는 리프 TU 와 병치될 수도 있다. 일부 예들에서, 리프 TU 의 최대 크기는, 대응하는 리프 CU 의 크기에 대응할 수도 있다. 일반적으로, 본 개시는, 다르게 언급되지 않는 한, 리프 CU 및 리프 TU 를 지칭하기 위하여 용어 CU 및 TU 를 각각 이용한다.
- [0071] 본 개시는 본원에 기재된 샘플들의 하나 이상의 블록들의 샘플들을 코딩하는데 이용된 샘플들 및 신택스 구조들의 임의의 하나 이상의 블록들을 지칭하기 위하여 용어 "비디오 유닛", "비디오 블록" 또는 "비디오 데이터의 블록" 을 이용할 수도 있다. 비디오 블록들의 예의 타입들은 HEVC 의 맥락에서 CTU들, CU들, PU들, 또는 TU 들, 또는 다른 표준들의 맥락에서 유사한 데이터 구조들 (예를 들어, H.264/AVC 매크로블록들 또는 이들의 매크로블록 파티션) 을 포함할 수도 있다.
- [0072] 변환 계수들을 생성하기 위한 임의의 변환 다음에, 비디오 인코더 (20) 는 변환 계수들의 양자화를 수행할 수도 있다. 일반적으로 양자화는, 변환 계수들이 양자화되어 그 계수들을 나타내는데 이용된 데이터의 양을 감소시킬 수 있으며, 추가 압축을 제공하는 프로세스를 지칭한다. 양자화 프로세스는 계수들의 일부 또는 전부

와 연관된 비트 깊이를 감소시킬 수도 있다. 예를 들면,  $n$  비트 값은 양자화 동안  $m$  비트 값으로 반내림(round down) 될 수도 있고, 여기서  $n$ 은  $m$ 보다 더 크다.

[0073] 양자화 다음에, 비디오 인코더는 변환 계수들을 스캔하여, 양자화된 변환 계수들을 포함하는 2차원 매트릭스로부터 1차원 벡터를 생성할 수도 있다. 그 스캔은 더 높은 에너지 (그리고 따라서 더 낮은 주파수) 계수들을 어레이의 전방에 두고 더 낮은 에너지 (그리고 따라서 더 높은 주파수) 계수들을 어레이의 후방에 두도록 설계될 수도 있다. 일부 예들에서, 비디오 인코더 (20) 는, 미리정의된 스캔 순서를 이용하여 양자화된 변환 계수들을 스캔함으로써 엔트로피 인코딩될 수 있는 직렬화된 벡터를 생성할 수도 있다. 다른 예들에서, 비디오 인코더 (20) 는 적응적 스캔을 수행할 수도 있다. 양자화된 변환 계수들을 스캔하여 1차원 벡터를 형성한 후에, 비디오 인코더 (20) 는 1차원 벡터를, 예를 들어, CAVLC (context-adaptive variable length coding), CABAC (context-adaptive binary arithmetic coding), SBAC (syntax-based context-adaptive binary arithmetic coding), PIPE (Probability Interval Partitioning Entropy) 코딩 또는 또 다른 엔트로피 인코딩 방법론에 따라, 엔트로피 인코딩할 수도 있다. 비디오 인코더 (20) 는 또한, 비디오 데이터를 디코딩함에 있어서 비디오 디코더 (30) 에 의한 이용을 위해 인코딩된 비디오 데이터와 연관된 신택스 엘리먼트들을 엔트로피 인코딩할 수도 있다.

[0074] CABAC 을 수행하기 위하여, 비디오 인코더 (20) 는, 송신될 심볼에 콘텍스트 모델 내의 콘텍스트를 배정할 수도 있다. 콘텍스트는, 예를 들어, 심볼의 이웃하는 값들이 비영(non-zero) 인지 여부에 관한 것일 수도 있다. CAVLC 을 수행하기 위하여, 비디오 인코더 (20) 는, 송신될 심볼을 위해 가변 길이 코드를 선택할 수도 있다. VLC 에서의 코드워드들은, 상대적으로 더 짧은 코드들이 더 높은 확률 심볼들에 대응하는 한편, 더 긴 코드들이 더 적은 확률 심볼들에 대응하도록 구성될 수도 있다. 이런 식으로, VLC 의 이용은, 예를 들어, 송신될 각 심볼에 동일 길이 코드워드들을 이용하는 것에 비해, 비트 절약(bit savings) 을 달성할 수도 있다. 확률 결정은, 심볼에 할당된 콘텍스트에 기초할 수도 있다.

[0075] 또한, 비디오 인코더 (20) 는 인코딩된 화상들을, 예를 들어, 잔차 데이터를 역 양자화 및 역 변환함으로써, 디코딩하고 잔차 데이터와 예측 데이터를 조합할 수도 있다. 이런 식으로, 비디오 인코더 (20) 는 비디오 디코더 (30) 에 의해 수행되는 디코딩 프로세스를 시뮬레이션할 수 있다. 그러므로, 비디오 인코더 (20) 및 비디오 디코더 (30) 양자 모두는 실질적으로 동일한 디코딩된 비디오 데이터, 예를 들어, 화상들 또는 화상들로부터의 블록들에 대해, 인트라 화상, 인터 화상 또는 IntraBC 예측에서의 이용을 위해, 액세스한다.

[0076] 비디오 인코더 (20) 는, 신택스 엘리먼트들을 포함한, 코딩된 데이터 및 연관된 데이터의 표현을 형성하는 비트들의 시퀀스를 포함하는 인코딩된 비디오 비트스트림을 출력할 수도 있다. 비트스트림은 네트워크 추상 계층(network abstraction layer; NAL) 유닛들의 시퀀스를 포함할 수도 있다. NAL 유닛들의 각각은, NAL 유닛 헤더를 포함할 수도 있고, RBSP (raw byte sequence payload) 를 캡슐화(encapsulate) 할 수도 있다. NAL 유닛 헤더는, NAL 유닛 타입 코드를 표시하는 신택스 엘리먼트를 포함할 수도 있다. NAL 유닛의 NAL 유닛 헤더에 의해 명시되는 NAL 유닛 타입 코드는 NAL 유닛의 타입을 표시한다. RBSP 는 NAL 유닛 내에 캡슐화되는 정수 개의 바이트들을 포함하는 신택스 구조를 포함할 수도 있다. 일부 사례들에서, RBSP 는 제로 비트들을 포함한다.

[0077] 상이한 타입들의 NAL 유닛들은 상이한 타입들의 RBSP 들을 캡슐화할 수도 있다. 예를 들어, 제 1 타입의 NAL 유닛은, 파라미터 세트를 위해 RBSP 를 캡슐화할 수도 있고, 제 2 타입의 NAL 유닛은 코딩된 슬라이스를 위해 RBSP 를 캡슐화할 수도 있고, 제 3 타입의 NAL 유닛은, SEI (Supplemental Enhancement Information) 를 위해 RBSP 를 캡슐화할 수도 있는 등이다. (파라미터 세트들 및 SEI 메시지들을 위한 RBSP 들과는 반대로) 비디오 코딩 데이터를 위해 RBSP 를 캡슐화하는 NAL 유닛들은 비디오 코딩 계층(VCL) NAL 유닛들로 지칭될 수도 있다. 코딩된 슬라이스를 캡슐화하는 NAL 유닛은 코딩된 슬라이스 NAL 유닛으로 본원에서 지칭될 수도 있다. 코딩된 슬라이스를 위한 RBSP 는 슬라이스 헤더 및 슬라이스 데이터를 포함할 수도 있다.

[0078] 비디오 인코더 (20) 는 인코딩된 비디오 비트스트림에서, 인코딩된 비디오 데이터에 추가하여, 비디오 데이터의 특정 블록, 또는 이의 그룹을 디코딩하는 법을 비디오 디코더에 알리는 신택스 엘리먼트들을 포함시킬 수도 있다. 비디오 인코더 (20) 는 다양한 신택스 구조들에 신택스 엘리먼트들을, 그것이 가리키는 비디오 구조의 타입 (예를 들어, 시퀀스, 화상, 슬라이스, 블록), 및 그의 값이 얼마나 자주 변경될 수도 있는지에 따라, 포함시킬 수도 있다. 예를 들어, 비디오 인코더 (20) 는, 비디오 파라미터 세트 (VPS), 시퀀스 파라미터 세트 (SPS) 또는 화상 파라미터 세트 (PPS) 와 같은 파라미터 세트들에 신택스 엘리먼트들을 포함시킬 수도 있다. 다른 예들로서, 비디오 인코더 (20) 는 SEI 메시지들 및 슬라이스 헤더들에 신택스 엘리먼트들을 포함시킬 수

도 있다.

- [0079] 일반적으로, 비디오 디코더 (30) 는 비디오 인코더에 의해 수행되는 인코딩 프로세스의 역 (inverse) 인 디코딩 프로세스를 수행할 수도 있다. 예를 들어, 비디오 디코더 (30) 는 양자화된 비디오 데이터를 엔트로피 인코딩하기 위하여 비디오 인코더에 의해 이용되는 엔트로피 인코딩 기법들의 역을 이용하여 엔트로피 디코딩을 수행할 수도 있다. 비디오 디코더 (30) 는 또한, 비디오 인코더 (20) 에 의해 채용되는 양자화 기법들의 역을 이용하여 비디오 데이터를 역 양자화할 수도 있고, 양자화되는 변환 계수들을 산출하기 위하여 비디오 인코더 (20) 에 의해 이용되는 변환의 역을 수행할 수도 있다. 다음으로, 비디오 디코더 (30) 는 결과적인 잔차 블록들을 인접 참조 비디오 데이터 (인트라 예측), 또 다른 화상으로부터의 예측 블록들 (인터 예측), 또는 동일한 화상 (IntraBC) 으로부터의 예측 블록들에 적용하여 궁극적인 디스플레이를 위한 비디오 블록을 산출할 수도 있다. 비디오 디코더 (30) 는, 비디오 디코더 (30) 에 의해 수신되는 비트스트림에서의 인코딩된 비디오 데이터로 비디오 인코더 (20) 에 의해 제공되는 선택스 엘리먼트들에 기초하여 비디오 인코더 (20) 에 의해 수행되는 다양한 프로세스들의 역을 수행하도록 구성, 명령, 제어 또는 지시될 수도 있다.
- [0080] 각각의 화상은 루마 성분 및 하나 이상의 크로마 성분들을 포함할 수도 있다. 따라서, 여기에 기재된 블록 기반 인코딩 및 디코딩 동작들은, 루마 또는 크로마 픽셀 값들을 포함하거나 또는 이들에 연관된 블록들, 예를 들어, CU들, PU들 및 TU들에 동등하게 적용가능할 수도 있다.
- [0081] Budagavi 등의, “AHG8: 인트라 모션 보상을 이용한 비디오 코딩,” 문헌: JCTVC-M0350 [ITU-T SG 16 WP 3 및 ISO/IEC JTC 1/SC 29/WG 11 의 JCT-VC (Joint Collaborative Team on Video Coding, 13차 미팅: 한국, 인천, 2013년 4월 18-26 일] (이하, “JCT-VC M0350”) 에 보고된 바처럼, (IntraBC 로도 지칭되는) IntraMC 는 인트라 화상 리던던시를 제거하고 인트라 프레임 코딩 효율을 향상시키는 것을 가능하게 하는 전용 기법이다. JCT-VC M0350 는 참조에 의해 전부 인용되고 [http://phenix.int-evry.fr/jct/doc\\_end\\_user/current\\_document.php?id=7601](http://phenix.int-evry.fr/jct/doc_end_user/current_document.php?id=7601) 로부터 다운로드에 이용가능하다. JCT-VC M0350 에 따르면, IntraMC 는 다음을 코딩하는 것을 포함한다: (1) 현재 비디오 블록, 예를 들어 CU 로부터 변위된, 예를 들어 동일한 프레임 또는 화상 내의 블록인, 예측 신호의 위치를 표시하는 1차원 오프셋 또는 변위 벡터 (여기서 블록 벡터, 모션 벡터, 또는 “MV” 로도 불린다) 와 함께 (2) 잔차 신호. IntraMC 를 이용하는 CU 들 또는 다른 블록들에 대해, 예측 신호들이 동일한 화상에서 이미 재구성된 구역으로부터 획득된다.
- [0082] 도 2는, 본 개시에 따라 동일한 화상 내의 비디오 데이터의 예측 블록들로부터 비디오 데이터의 블록들의 인트라 예측을 위한 모드에 따라, 예를 들어 본 개시의 기법들에 따른 IntraBC 모드에 따라, 현재 화상 (103) 내의 비디오 데이터 (102) 의 현재 블록을 예측하기 위한 일 예의 기법을 예시하는 개념도이다. 도 2는 현재 화상 (103) 내의 비디오 데이터 (104) 의 예측 블록을 예시한다. 비디오 코더, 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는, 본 개시의 기법들에 따른 IntraBC 모드에 따라 현재 비디오 블록 (102) 을 예측하기 위하여 예측 비디오 블록 (104) 을 이용할 수도 있다.
- [0083] 비디오 인코더 (20) 는, 비디오 데이터의 이전에 재구성된 블록들의 세트로부터 현재 비디오 블록 (102) 을 예측하기 위한 예측 비디오 블록 (104) 을 선택한다. 비디오 인코더 (20) 는, 또한 인코딩된 비디오 비트스트림에 포함된 비디오 데이터를 역 양자화 및 역 변환하고, 비디오 데이터의 재구성된 블록들을 예측하는데 이용된 예측 블록들과 결과적인 잔차 블록들을 합산하는 것에 의해, 비디오 데이터의 블록들을 재구성한다. 도 2의 예에서, “의도된 구역” (intended area) 또는 “래스터 구역” (raster area) 으로도 지칭될 수도 있는 화상 (103) 내의 의도된 영역 (108) 은 이전에 재구성된 비디오 블록들의 세트를 포함한다. 비디오 인코더 (20) 는, 아래에서 더 자세하게 설명되어 있는 바처럼, 다양한 방식으로 화상 (103) 내의 의도된 영역 (108) 을 정의할 수도 있다. 비디오 인코더 (20) 는, 의도된 영역 (108) 내의 다양한 비디오 블록들에 기초하여 현재 비디오 블록 (102) 을 예측하고 코딩하는 상대적인 효율성 및 정확성의 분석에 기초하여 의도된 영역 (108) 에 있는 비디오 블록들 중에서 현재 비디오 블록 (102) 을 예측하기 위하여 예측 비디오 블록 (104) 을 선택할 수도 있다.
- [0084] 비디오 인코더 (20) 는 현재 비디오 블록 (102) 에 대해 예측 비디오 블록 (104) 의 위치 또는 변위를 나타내는 2차원 벡터 (106) 를 결정한다. 2차원 블록 벡터 (106) 는, 현재 비디오 블록 (102) 에 대해 예측 비디오 블록 (104) 의 수평 및 수직 변위를 각각 나타내는 수평 변위 성분 (112) 및 수직 변위 성분 (110) 을 포함한다. 비디오 인코더 (20) 는, 인코딩된 비디오 비트스트림에서, 2차원 블록 벡터 (106) 를 식별 또는 정의하는, 예를 들어, 수평 변위 성분 (112) 및 수직 변위 성분 (110) 을 정의하는, 하나 이상의 선택스 엘리먼트들을 포함할 수도 있다. 비디오 디코더 (30) 는 하나 이상의 선택스 엘리먼트들을 디코딩하여 2차원 블록

벡터 (106) 를 결정하고, 결정된 벡터를 이용하여 현재 비디오 블록 (102) 을 위한 예측 비디오 블록 (104) 을 식별할 수도 있다.

[0085] 일부 예들에서, 2차원 블록 벡터 (106) 의 해상도는 정수 픽셀일 수 있다, 예를 들어, 정수 픽셀 해상도를 갖는 것으로 제약될 수 있다. 그러한 예들에서, 수평 변위 성분 (112) 및 수직 변위 성분 (110) 의 해상도는 정수 픽셀일 수도 있다. 그러한 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 현재 비디오 블록 (102) 을 위한 예측자를 결정하기 위하여 예측 비디오 블록 (104) 의 픽셀 값들을 보간할 필요가 없다.

[0086] 다른 예들에서, 수평 변위 성분 (112) 및 수직 변위 성분 (110) 중의 하나 또는 양자 모두의 해상도는 서브 픽셀일 수 있다. 예를 들어, 성분들 (112 및 114) 중의 하나는 정수 픽셀 해상도를 가질 수도 있는 반면, 다른 것은 서브 픽셀 해상도를 갖는다. 일부 예들에서, 수평 변위 성분 (112) 및 수직 변위 성분 (110) 의 양자 모두의 해상도는 서브 픽셀일 수 있지만, 수평 변위 성분 (112) 및 수직 변위 성분 (110) 은 상이한 해상도를 가질 수도 있다.

[0087] 일부 예들에서, 비디오 코더, 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 수평 변위 성분 (112) 및 수직 변위 성분 (110) 의 해상도를 특정 레벨, 예를 들어, 블록 레벨, 슬라이스 레벨 또는 화상 레벨 적응화에 기초하여 적응시킨다. 예를 들어, 비디오 인코더 (20) 는, 수평 변위 성분 (112) 및 수직 변위 성분 (110) 이 정수 픽셀 해상도인지 또는 정수 픽셀 해상도가 아닌지를 표시하는 플래그를 슬라이스 레벨에서, 예를 들어, 슬라이스 헤더에서 시그널링할 수도 있다. 플래그가 수평 변위 성분 (112) 및 수직 변위 성분 (110) 이 정수 픽셀 해상도가 아니라고 표시하면, 비디오 디코더 (30) 는 해상도가 서브 픽셀 해상도라고 추론할 수도 있다. 일부 예들에서, 반드시 플래그인 것은 아닌 하나 이상의 선택스 엘리먼트들이 수평 변위 성분 (112) 및/또는 수직 변위 성분 (110) 의 집합적 또는 개별적 해상도들을 표시하기 위하여 비디오 데이터의 각각의 슬라이스 또는 다른 유닛에 대해 송신될 수도 있다.

[0088] 또 다른 예들에서, 플래그 또는 선택스 엘리먼트 대신에, 비디오 인코더 (20) 는 해상도 컨텍스트 정보로부터 수평 변위 성분 (112) 및/또는 수직 변위 성분 (110) 의 해상도에 기초하여 설정될 수도 있고, 비디오 디코더 (30) 는 해상도 컨텍스트 정보로부터 수평 변위 성분 (112) 및/또는 수직 변위 성분 (110) 의 해상도를 추론할 수도 있다. 예들로서, 해상도 컨텍스트 정보는, 현재 비디오 블록 (102) 을 포함하는 화상 또는 화상들의 시퀀스를 위한, 색공간 (예를 들어, YUV, RGB 등), 특정 컬러 포맷 (예를 들어, 4:4:4, 4:2:2, 4:2:0 등), 프레임 크기, 프레임 레이트, 또는 양자화 파라미터 (QP) 를 포함할 수도 있다. 적어도 일부 예들에서, 비디오 코더는, 이전에 코딩된 프레임들 또는 화상들에 관련된 정보에 기초하여, 수평 변위 성분 (112) 및/또는 수직 변위 성분 (110) 의 해상도를 결정할 수도 있다. 이런 식으로, 수평 변위 성분 (112) 의 해상도 및 수직 변위 성분 (110) 에 대한 해상도는 사전 정의되거나, 시그널링되거나, 다른 사이트 정보 (예를 들어, 해상도 컨텍스트 정보) 로부터 추론되거나, 또는 이미 코딩된 프레임들에 기초할 수도 있다.

[0089] 현재 비디오 블록 (102) 은 CU, 또는 CU 의 PU 일 수도 있다. 일부 예들에서, 비디오 코더, 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는, IntraBC 에 따라 예측되는 CU 를 다수의 PU 들로 스플릿할 수도 있다. 그러한 예들에서, 비디오 코더는, CU 의 PU 들의 각각에 대해 각각의 (예를 들어, 상이한) 2차원 벡터 (106) 를 결정할 수도 있다. 예를 들어, 비디오 코더는 2Nx2N CU 를 2개의 2NxN PU들, 2개의 Nx2N PU들, 또는 4개의 NxN PU 들로 스플리팅할 수도 있다. 다른 예들로서, 비디오 코더는 2Nx2N CU 를 ((N/2)xN + (3N/2)xN) PU들, ((3N/2)xN + (N/2)xN) PU들, (Nx(N/2) + Nx(3N/2)) PU들, (Nx(3N/2) + Nx(N/2)) PU들, 4개의 (N/2)x2N PU들, 또는 4개의 2Nx(N/2) PU들로 스플리팅할 수도 있다. 일부 예들에서, 비디오 코더는 2Nx2N PU 를 이용하여 2Nx2N CU 를 예측할 수도 있다.

[0090] 현재 비디오 블록 (102) 은 루마 비디오 블록, 또는 루마 비디오 블록에 대응하는 크로마 비디오 블록일 수도 있다. 일부 예들에서, 비디오 인코더 (20) 는 단지 루마 비디오 블록들을 위한 2차원 벡터들 (106) 을 정의하는 하나 이상의 선택스 엘리먼트들을 인코딩된 비디오 비트스트림으로 인코딩할 수도 있다. 그러한 예들에서, 비디오 디코더 (30) 는, 루마 블록을 위해 시그널링되는 2차원 벡터에 기초하여 루마 블록에 대응하는 하나 이상의 크로마 블록들의 각각에 대해 2차원 벡터들 (106) 을 도출할 수도 있다.

[0091] 컬러 포맷, 예를 들어, 컬러 샘플링 포맷 또는 크로마 샘플링 포맷에 따라, 비디오 코더는 루마 비디오 블록에 대하여 대응하는 크로마 비디오 블록들을 다운샘플링할 수도 있다. 컬러 포맷 4:4:4 은 다운샘플링을 포함하지 않으며, 크로마 블록들이 루마 블록과 수평 및 수직 방향들에서 같은 수의 샘플들을 포함한다는 것을 의미한다. 컬러 포맷 4:2:2 는 수평 방향에서 다운샘플링되며, 루마 블록에 대해 크로마 블록들에서 수평 방향에 절반의 샘플들이 있다는 것을 의미한다. 컬러 포맷 4:2:0 는 수평 및 수직 방향들에서 다운샘플링되며,



루마 블록에 대해 크로마 블록들에서 수평 및 수직 방향들에 절반의 샘플들이 있다는 것을 의미한다.

[0092] 비디오 코더들이 대응하는 루마 블록들을 위한 벡터들 (106) 에 기초하여 크로마 비디오 블록들을 위한 벡터들 (106) 을 결정하는 예들에서, 비디오 코더들은 루마 벡터를 수정할 필요가 있을 수도 있다. 예를 들어, 루마 벡터 (106) 는 수평 변위 성분 (112) 및/또는 수직 변위 성분 (110) 이 홀수 개의 픽셀들인 정수 해상도를 갖고, 컬러 포맷이 4:2:2 또는 4:2:0 이면, 컨버팅된 루마 벡터는 대응하는 크로마 블록에서 정수 픽셀 위치를 가리키지 않을 수도 있다. 그러한 예들에서, 비디오 코더들은 대응하는 크로마 블록을 예측하기 위한 크로마 벡터로서 이용하기 위해 루마 벡터를 스케일링할 수도 있다. 일부 예들에서, 비디오 인코더 (20) 는, 의도된 영역 (108) 을 정의하거나, 또는 컨버팅된 벡터를 스케일링하여, 크로마 블록을 예측하는데 이용된 컨버팅된 루마 벡터 (106) 가, 재구성되지 않은 또는 인루프 필터링된 예측 크로마 블록들을 가리키지 않을 수도 있다. 본 개시에서, 크로마 블록을 위한 벡터의 스케일링은 명시적으로 언급되지 않지만, 그러한 스케일링이 일어나지 않는 것으로 가정되지 않아야 한다. 크로마 블록을 위한 벡터의 스케일링은, 본 개시에 설명된 예들에서 명시적으로 설명되지 않더라도 일어날 수도 있다 (하지만, 모든 경우에 반드시 그러한 것은 아닐 수도 있다).

[0093] 도 3은, 비디오 인코더 (20) 가 비디오 데이터의 현재 블록을 예측하기 위하여 비디오 데이터의 예측 블록을 선택할 수도 있는, 의도된 영역을 정의하기 위한 일 예의 기법을 예시하는 개념도이다. 도 3에 의해 예시된 예에서, 비디오 인코더 (20) 는 비디오 데이터 (122) 의 현재 블록을 예측 및 인코딩한다. 비디오 인코더 (20) 는 현재 비디오 블록 (122) 을 예측하기 위하여 의도된 영역 (128) 내에서 비디오 데이터 (124) 의 예측 블록을 선택한다. 비디오 인코더 (20) 는, 현재 비디오 블록 (122) 에 대한 예측 비디오 블록 (124) 의 변위를 표시하는, 수평 변위 성분 (132) 및 수직 변위 성분 (130) 을 포함하는, 2차원 벡터 (126) 를 결정한다. 비디오 인코더 (20) 는 2차원 벡터 (126) 를 정의하는 인코딩된 비디오 비트스트림에서 하나 이상의 신택스 엘리먼트들을 인코딩한다.

[0094] 일부 예들에서, 비디오 인코더 (20) 는, 특히 비디오 디코더 (30) 에서의, 구현 및 프로세싱 복잡성이 감소될 수 있도록, 의도된 영역 (128) 을 정의, 예를 들어, 높이, 폭 또는 다른 치수와 같은 의도된 영역의 크기를 정의한다. 그렇게 함에 있어서, 비디오 인코더 (20) 는 2차원 벡터 (126) 의 크기를 한정, 예를 들어, 수직 변위 성분 (130) 및/또는 수평 변위 성분 (132) 의 크기를 한정할 수도 있다. 일부 예들에서, 비디오 인코더 (20) 는 비디오 인코더 (20) 및 비디오 디코더 (30) 에 의해 비디오 데이터를 병렬 프로세싱하는 것을 가능하게 하기 위해 의도된 영역 (128) 을 한정할 수도 있다. 일부 예들에서, 비디오 인코더 (20) 는, 인루프 필터링, 이를테면 디블록킹 및 샘플 적응적 오프셋 (SAO) 필터링 없이, 그리고 비디오 코더의 메모리 요구사항들을 지나치게 부담 지우거나 또는 그러한 인루프 필터링의 적용을 지연시킴 없이 예측 비디오 블록들의 이용을 가능하게 하기 위하여 의도된 영역 (128) 을 한정할 수도 있다.

[0095] 도 3에 예시된 바처럼, 현재 비디오 블록 (122) 은 현재 LCU (134) 내에 있다. 도 3은 또한, 현재 LCU 의 좌측 이웃하는 (좌측) LCU (136) 를 예시한다. 좌측 LCU (136) 가 도 3에 예시되어 있는데 왜냐하면 화상의 비디오 블록들은 통상적으로, 상부 좌측에서 하부 우측으로의 래스터 스캔 순서에서 인코딩되기 때문이다. 화상의 비디오 블록들이 상이한 순서에서 코딩되는 예들에서, 좌측 LCU (136) 에 관한 이하의 논의는 현재 LCU (134) 의 상이한, 이웃하는 LCU 에 적용될 수도 있다.

[0096] 일부 예들에서, 비디오 인코더 (20) 는, 페치 (fetch) 된 예측 비디오 블록 (124) 이 현재 비디오 블록 (122) 과 동일한 LCU 내에, 즉, 현재 LCU (134) 내에 있도록 의도된 영역 (128) 을 한정할 수도 있다. 현재 LCU (134) 로 의도된 영역 (128) 을 한정하는 것은 비디오 코더들에 의한 LCU 들의 병렬 프로세싱을 가능하게 할 수도 있는데, 왜냐하면 비디오 코더의 프로세싱 유닛들은 현재 LCU 의 블록을 코딩할 때 또 다른 LCU 로부터 정보를 필요로 하지 않을 수도 있기 때문이다.

[0097] 의도된 영역 (128) 이 현재 LCU (134) 에 한정되는 일부 예들에서, 2차원 벡터 (126) 는, 현재 비디오 블록 (122) 이 현재 LCU (134) 의 가장 상단 블록인 경우에 수평 벡터로, 그리고 현재 비디오 블록 (122) 이 현재 LCU (134) 의 가장 좌측 블록인 경우에 수직 벡터로 한정될 수도 있다. 그러한 예들에서, 비디오 인코더 (20) 는, 2차원 벡터 (126) 의 수평 변위 성분 (132) 을 정의하는 하나 이상의 신택스 엘리먼트들을 인코딩할 수도 있고, 현재 비디오 블록 (122) 이 현재 LCU (134) 의 가장 상단 블록인 경우에, 제로가 되는, 2차원 벡터 (126) 의 수직 변위 성분 (130) 을 정의하는 하나 이상의 신택스 엘리먼트들을 인코딩할 필요가 없다. 유사하게, 비디오 인코더 (20) 는, 2차원 벡터 (126) 의 수직 변위 성분 (130) 을 정의하는 하나 이상의 신택스 엘리먼트들을 인코딩할 수도 있고, 현재 비디오 블록 (122) 이 현재 LCU (134) 의 가장 좌측 블록인 경우에, 제로

가 되는, 2차원 벡터 (126) 의 수평 변위 성분 (132) 을 정의하는 하나 이상의 선택스 엘리먼트들을 인코딩할 필요가 없다. 유사하게, 의도된 영역 (128) 이 현재 LCU (134) 로 한정되고 현재 비디오 블록 (122) 이 현재 LCU (134) 의 상단 좌측 유닛인 경우에, 2차원 벡터 (126) 의 수평 및 수직 성분들 (130, 132) 양자 모두는 제로가 되어 한다. 이 상황에 직면하는 일부 예들에서, 비디오 인코더 (20) 는 IntraBC 를 수행하지 않을 수도 있고, IntraBC 를 위한 임의의 선택스 엘리먼트들, 이를테면 2차원 벡터 (126) 를 표시하기 위한 선택스 엘리먼트들, 또는 현재 비디오 블록 (122) 이 IntraBC 에 따라 예측되는지 여부를 표시하는 플래그를 선출할 필요가 없다.

[0098] 의도된 영역이 현재 LCU 에 한정되고 현재 비디오 블록이 현재 LCU 에서 특정 위치들 내에 있을 때 IntraBC 에 대한 시그널링을 감소시키기 위한 이들 기법들은 또한, JCT-VC M0350 에서도 제안된 것들과 같은 1차원 모션 벡터들에 한정된 IntraBC 기법들을 위해 적용될 수도 있다. 예를 들어, 수직 모션 벡터가 현재 LCU 내에 있는 것으로 제한되고 현재 유닛이 가장 상단 유닛인 경우, 모션이 수직 또는 수평인지를 시그널링할 필요가 없다.

[0099] 일부 예들에서, 비디오 인코더 (20) 는 도 3에 예시된 바처럼, 의도된 영역 (128) 을, 현재 LCU (134) 및 하나 이상의 이웃하는 LCU 들, 예를 들어, 좌측 LCU (136) 에서 재구성된 구역의 부분 내에 있는 것으로 한정할 수도 있다. 이런 식으로, 비디오 인코더 (20) 는, 도 3에 예시된 바처럼, 현재 LCU (134) 에서 재구성된 블록들에 추가하여, 좌측 LCU (136) 와 같은 하나 이상의 이웃하는 LCU 들의 재구성된 구역에 있는 블록들만을 나타낼 수 있도록 2차원 벡터들 (126) 을 한정한다.

[0100] 일부 예들에서, 비디오 인코더 (20) 는 현재 LCU (134) 의 크기에 기초하여 의도된 영역 (128) 의 크기를 한정한다. 예를 들어, 도 3에 예시된 바처럼, 비디오 인코더 (20) 는, 의도된 영역 (128) 이 현재 LCU (134) 및 좌측 LCU (136) 을 넘어 수직으로 연장되지 않도록, 현재 LCU (134) 의 높이 (또는 수직 상한) 에 기초하여 의도된 영역 (128) 의 높이 (138) 를 한정할 수도 있다. LCU (또는 CTB) 의 크기는, 예를 들어, 슬라이스 헤더, 파라미터 세트, 또는 SEI 메시지를 통해, 인코딩된 비디오 비트스트림에서 비디오 디코더 (30) 로 비디오 인코더 (20) 에 의해 시그널링될 수도 있고, 그래서 LCU 크기에 기초한 의도된 영역에 대한 한계가 또한 비디오 디코더 (30) 로 효과적으로 시그널링될 수도 있다. 일부 예들에서, 비디오 인코더 (20) 는, 정수 개의 픽셀들에 기초하여 의도된 영역 (128) 의 현재 비디오 블록 (122) 의 좌측으로 크기, 예를 들어, 높이 (138) 또는 폭 (140) 을 한정한다. 예를 들어, 비디오 인코더 (20) 는, LCU들 (134 및 136) 의 폭에 대응할 수도 있는, 64 개와 같은 정수 개의 픽셀들로 현재 비디오 블록 (122) 의 좌측으로 폭 (140) 을 한정할 수도 있다.

[0101] 비디오 인코더 (20) 는 인루프 필터링, 이를테면 디블록킹 및 샘플 적응적 오프셋 (SAO) 필터링 없이 재구성된 비디오 블록들을 포함하도록 의도된 영역 (128) 을 한정할 수도 있다. 이런 식으로, 비디오 인코더 (20) 는, 현재 LCU (134) 에 추가하여, 인루프 필터링이 수행되지 않은, 좌측 LCU (136) 와 같은 이웃하는 LCU 의 재구성된 구역에 있는 블록들만을 나타낼 수 있도록 IntraBC 를 위한 2차원 벡터들을 한정할 수도 있다. 비디오 인코더 (20) 는, 의도된 영역 (128) 의 크기를 증가시킬 수 있지만, 그러한 증가는 의도된 영역에서 비디오 블록들의 인루프 필터링 및 추가 프로세싱을 지연시키거나 또는 인루프 필터링 전에 샘플들을 저장하기 위해 추가적인 메모리를 필요로 할 수도 있다. 따라서, 비디오 인코더 (20) 는 예측 정확성 및 효율성 사이의 밸런스를 달성하기 위하여, 예를 들어 본원에 기재된 바처럼, 의도된 영역 (128) 을 한정할 수도 있다.

[0102] 도 4는 비디오 데이터 (260) 의 현재 블록 및 비디오 데이터의 현재 블록을 위한 후보 예측 벡터들이 도출될 수도 있는 비디오 데이터 (262, 264, 266) 의 이웃하는 블록들의 일 예의 개념도이다. 일부 예들에서, 인터 예측을 위한 병합 모드 또는 AMVP (Advanced Motion Vector Prediction) 모드와 유사하게, 비디오 코더들은, IntraBC 를 이용하여, 이웃하는 비디오 블록들, 예를 들어, PU 들을 예측하는데 이용된 2차원 벡터들에 기초하여, 현재 비디오 블록, 예를 들어, PU 를 위한 예측 2차원 벡터를 결정할 수도 있다. 도 4 에 예시된 이웃하는 블록들은, 좌측 이웃하는 블록 (262), 상부 (상단) 이웃하는 블록 (264) 및 상부 좌측 (상단 좌측) 이웃하는 블록 (266) 을 포함한다. 도 4 에 예시된 이웃하는 블록들은 일 예이다. 다른 예들에서, 비디오 코더는, 인터 예측을 위한 병합 및 AMVP 모드들에 따라 고려될 수도 있는 임의의 이웃하는 블록과 같은 더 많거나, 더 적거나 및/또는 상이한 이웃 블록들의 벡터들을 고려할 수도 있다.

[0103] 예측자는 이웃하는 블록들 중의 선택된 하나로부터 2차원 블록 벡터의 수평 및/또는 수직 변위 성분일 수도 있다. 일부 예들에서, 예측자는 항상, 특정 이웃 유닛, 예를 들어, 상단 유닛 또는 좌측 유닛으로부터 기인한다. 일부 예들에서, 이웃하는 블록들 중의 어느 것이 현재 PU 를 위한 예측 2차원 블록 벡터를 제공하는지는 PU 형상 및/또는 PU 의 인덱스에 기초하여 결정된다. 일부 예들에서, 예측자는 복수의 이웃하는 유닛들로부터의 2차원 블록 벡터의 수평 및/또는 수직 성분들의 함수 (이를테면 평균 (mean) 또는 메디안 (median))

일 수 있다.

- [0104] 일반적으로, 이웃 블록이 예측 2차원 벡터를 제공할 수 없으면, 예측 2차원 벡터 (또는 후보 예측 2차원 벡터) 는 제로 또는 디폴트 2차원 벡터로 설정될 수도 있다. 일부 예들에서, 이웃하는 블록들의 벡터들만이, IntraBC 에 따라 이웃하는 비디오 블록이 예측될 때, 현재 비디오 블록을 위한 예측 벡터들로서 이용가능할 수도 있다. 즉, IntraMC 모드로 코딩되지 않은 이웃하는 블록들은 현재 비디오 블록을 위한 블록 벡터 예측을 위해 이용가능하지 않은 것으로 고려될 수도 있다. 예를 들어, 예측자가 항상, 좌측 이웃하는 블록으로부터 기인하고 좌측 이웃하는 블록이 IntraMC 모드를 이용하여 예측될 때, 그의 2차원 벡터가 예측 2차원 벡터로서 이용된다. 예측자는 항상, 좌측 이웃하는 블록으로부터 기인하고 좌측 이웃하는 블록이 IntraMC 모드로 예측되지 않을 때, 비디오 코더들은 현재 비디오 블록을 위한 예측 2차원 벡터로서 제로 벡터 또는 디폴트 벡터를 이용할 수도 있다.
- [0105] 일부 예들에서, 비디오 코더는, 이웃하는 비디오 블록들의 2차원 벡터를 현재 비디오 블록의 2차원 벡터의 예측을 위해 이용가능하지 않은 것으로, 그것이 현재 비디오 블록, 예를 들어, CU 또는 PU 와 동일한 LCU 또는 다른 더 크거나 또는 최대인 코딩 유닛에 있지 않은 경우에, 고려할 수도 있다. 일부 예들에서, 현재 비디오 블록, 예를 들어, PU 또는 CU 를 위한 현재 2차원 벡터가 현재 CU 또는 LCU 중의 첫번째 것인 경우, 현재 비디오 블록을 위한 예측 2차원 벡터는 제로 또는 디폴트 2차원 벡터로 설정될 수도 있다. 상부 이웃하는 CU 및 현재 CU 가 동일한 LCU 에 있지 않으면, 상부 이웃하는 CU 는 이용가능하지 않은 것으로 고려될 수 있다. 또한, 좌측 이웃하는 CU 및 현재 CU 가 동일한 LCU 에 있지 않으면, 좌측 이웃하는 CU 는 이용가능하지 않은 것으로 고려될 수 있다.
- [0106] 일부 예들에서, 인터 예측을 위한 병합 모드 및 AMVP 모드와 유사한 방식으로, 비디오 인코더 (20) 및 비디오 디코더 (30) 는, 다수의 이웃 블록들의 블록 벡터들을 포함하는 현재 비디오 블록을 위한 후보 예측 벡터들의 세트를 구성할 수도 있다. 그러한 예들에서, 어느 후보가 현재 비디오 블록을 위해 예측 2차원 벡터를 제공하는데 이용되는지를 시그널링하는 인덱스를 비디오 인코더 (20) 가 인코딩할 수도 있고, 비디오 디코더 (30) 가 디코딩할 수도 있다. 일부 예들에서, 비디오 코더들은, 예를 들어, 상이한 이웃 블록들로부터, 상이한 후보 벡터들에 기초하여 예측 2차원 벡터의 수평 변위 성분 및 수직 변위 성분을 각각 결정할 수도 있다. 그러한 예들에서, 비디오 코더들은 후보 리스트 수평 및 수직 변위 성분들로 각각의 인덱스를 코딩할 수도 있다.
- [0107] 일 예로서, IntraBC 모드에 따라 코딩된 현재 CU 에 대해, 그의 블록 벡터를 코딩하기 위하여, 비디오 코더들은 좌측 및 상단 이웃하는 CU 들을 체크한다. 그들 중 어느 것도 이용가능하지 않으면, 예를 들어, IntraBC 에 따라 코딩되지 않았으면, 비디오 코더는 현재 CU 의 예측 2차원 벡터가 (수평 및 수직 성분들 양자 모두에 대해) 제로 벡터라고 결정할 수도 있다. 이웃하는 CU 들 중의 하나만이 이용가능하면, 예를 들어, IntraBC 로 코딩되었으면, 비디오 코더들은, 이용가능한 이웃하는 CU 의 예측을 위해 이용된 벡터를 현재 CU 의 현재 2차원 벡터의 예측 2차원 벡터로서 이용한다. 양자 모두의 이웃하는 CU 들이 이용가능하면, 예를 들어, IntraBC 로 코딩되었고 그들의 2차원 벡터들이 동일하면, 이 2차원 벡터는 예측 2차원 벡터로서 이용된다. 양자 모두의 이웃하는 CU 들이 이용가능하면, 예를 들어, IntraBC 로 코딩되었고 그들의 블록 벡터들이 상이하면, 비디오 코더들은, 상단 이웃하는 CU 로부터의 블록 벡터 또는 좌측 이웃하는 CU 로부터의 블록 벡터가 현재 CU 를 위한 예측 2차원 벡터로서 이용되는지 여부를 나타내기 위하여 플래그 또는 다른 선택스 엘리먼트를 코딩할 수도 있다.
- [0108] 또 다른 예에서, 비디오 코더들은 (비제한적인 예로서) 세트  $(-w, 0), (-2w, 0), (-8, 0), (0, 0), (0, 8), (0, -h), (0, -2h)$  로부터 비롯될 수 있는 2개 이상의 디폴트 블록 벡터들을 정의할 수도 있고, 여기서  $w$  및  $h$  는 현재 비디오 블록, 예를 들어, CU 의 폭 및 높이이고, 제 1 성분은 수평 변위이고 제 2 성분은 수직 변위이다. 좌측 이웃하는 CU 및 상부 이웃하는 CU 가 이용가능하면, 비디오 코더들은 그들의 벡터들을 현재 CU 를 위한 제 1 및 제 2 예측 2차원 벡터들로서 이용할 수도 있다. 이들 중 하나가 이용가능하지 않으면, 비디오 코더들은 후보 예측 벡터들의 리스트에서 이용가능하지 않은 예측 벡터를 대체하기 위하여 디폴트 예측 벡터를 이용할 수도 있다. 이들 중 양자 모두가 이용가능하지 않으면, 비디오 코더들은 후보 예측 벡터들의 리스트에서 이용가능하지 않은 예측 벡터를 대체하기 위하여, 동일 또는 상이할 수도 있는, 2개의 디폴트 예측 벡터들을 이용할 수도 있다. 후보 리스트에서 디폴트 벡터들로 이용가능하지 않은 예측 벡터들을 대체하는 이점은 예측자들간의 선택을 위한 플래그가 또한 인코딩된 비디오 비트스트림에 포함되어, 비디오 디코더 (30) 가 이 플래그를 조건적으로 파싱 (conditionally parsing) 할 필요가 없을 수도 있다는 것이다. 비록 예들이 2개의 이웃하는 블록들로부터 도출된 2개의 예측 블록들을 포함하는 후보 리스트에 대하여 여기에서 설명되었지만, 블록

들/벡터들이 이용가능하면, 다른 예들에서 비디오 코더는 더 많거나 또는 더 적은 이웃 블록들을 고려할 수도 있고 더 많거나 또는 더 적은 예측 2차원 벡터들을 후보 리스트에 포함시킬 수도 있다.

[0109] 일부 예들에서, 각각의 LCU 에서 IntraBC 에 이용된 제 1 비디오 블록 및 대응하는 2차원 벡터에 대해, 비디오 코더들은 LCU 내에서 다른 블록들 및 연관된 벡터들을 위한 것보다는 예측 2차원 벡터를 결정하기 위해 상이한 도출 프로세스를 이용할 수도 있다. 예를 들어, 비디오 코더, 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는, LCU 에서 IntraBC 를 위해 이용된 제 1 비디오 블록 및 대응하는 2차원 벡터를 위한 예측 2차원 벡터가 디폴트 2차원 벡터라고 결정할 수도 있다. 디폴트 2차원 벡터는  $(-w, 0)$  일 수도 있고, 여기서  $w$  는 현재 비디오 블록, 예를 들어, CU 의 폭이다. 다른 예들에서, 디폴트 2차원 벡터는  $(-2w, 0)$ ,  $(-8, 0)$ ,  $(0, 0)$ ,  $(0, 8)$ ,  $(0, -h)$ ,  $(0, -2h)$  일 수도 있고, 여기서  $w$  및  $h$  는 현재 비디오 블록, 예를 들어, CU 의 폭 및 높이이다. 다른 예들에서, 비디오 코더, 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는, LCU 에서 IntraBC 를 위해 이용된 제 1 비디오 블록 및 대응하는 2차원 벡터를 위한 예측 2차원 벡터를 도출할 수도 있다. 일부 예들에서, 비디오 코더들은 상이한 크기들을 갖거나 또는 LCU 에서 상이한 위치에 있는 비디오 블록들, 예를 들어, CU 들에 대해 상이하게 LCU 에서 IntraBC 를 이용하여 코딩된 제 1 비디오 블록을 위한 예측 2차원 벡터들을 도출할 수도 있거나, 또는 다른 말로, 비디오 블록 크기 또는 LCU 에서의 위치에 기초하여 상이하게 LCU 에서의 IntraBC 를 이용하여 코딩되는 제 1 비디오 블록을 위해 예측 2차원 벡터들을 도출할 수도 있다.

[0110] 일부 예들에서, 비디오 코더들은 현재 비디오 블록의 2차원 벡터의 수평 및 수직 성분들을 결정하는데 이용되는 방법 또는 방법들을 플래그, 신택스 엘리먼트들에 기초하거나 또는 다른 정보에 기초하여 (이를테면 특정 색공간 (예를 들어, YUV, RGB 등)), 특정 컬러 포맷 (예를 들어, 4:2:0, 4:4:4 등), 프레임 크기, 프레임 레이트, 또는 양자화 파라미터 (QP), 또는 이전에 코딩된 프레임들에 기초하여) 선택할 수도 있다.

[0111] 일부 예들에서, 비디오 인코더 (20) 는, IntraMC 또는 IntraBC 모드로서 지칭될 수도 있는, 동일한 화상에 있는 예측 비디오 블록들에 기초하여 현재 비디오 블록들을 예측하기 위한 모드를 이용하여 비디오 블록, 예를 들어, CU 가 인코딩되는지 여부를 비디오 디코더 (30) 에 표시할 플래그, 예를 들어, IntraMC\_flag 또는 IntraBC\_flag 를 인코딩할 수도 있다. 플래그는, 예를 들어 본원에 기재된 바처럼, 하나 이상의 CU 들 또는 다른 블록들 또는 유닛들이 IntraMC 또는 IntraBC 모드를 이용하여 인코딩되는지 여부를 표시하는 값을 갖는 1-비트 플래그일 수도 있다. 비디오 코더들은 각각의 비디오 블록, 예를 들어, CU 또는 PU 에 대해 플래그의 값을 코딩할 수도 있거나, 또는 예를 들어 슬라이스 내에서, 또는 프레임 또는 화상, 또는 화상들의 시퀀스 내에서, 복수의 비디오 블록들의 각각에 플래그의 값을 코딩할 수도 있다. 비디오 코더들은, 예를 들어, 슬라이스 헤더, 파라미터 세트 (예를 들어, PPS), 또는 SEI 메시지에서 플래그를 코딩할 수도 있다.

[0112] 비디오 코더들은 바이패스 모드에서 플래그를 코딩할 수도 있거나, 또는 컨텍스트로 플래그를 산술 인코딩 (arithmetic encoding), 예를 들어, CABAC 인코딩할 수도 있다. 일부 예들에서, 비디오 코더들은, 이웃하는 비디오 블록들에 의존하지 않는 단일, 고정된 컨텍스트로 플래그를 산술 코딩, 예를 들어, CABAC 인코딩할 수도 있다. 다른 예들에서, 비디오 코더들은, 이웃하는 비디오 블록들로부터 도출되는 컨텍스트로 플래그를 산술 코딩, 예를 들어, CABAC 인코딩할 수도 있다.

[0113] 도 4를 참조하면, 비디오 코더들이 CABAC 컨텍스트를 이용하여 현재 비디오 블록 (260) 에 대한 플래그를 코딩할 때, 컨텍스트의 값은, 상단 (상부) 이웃하는 비디오 블록 (264), 상단-좌측 (상부-좌측) 이웃하는 비디오 블록 (266), 또는 좌측 이웃하는 비디오 블록 (262) 과 같은 이웃하는 비디오 블록들로부터 도출될 수 있다. 비디오 코더들은 이웃하는 비디오 블록들, 이를테면 상단 이웃하는 비디오 블록 (264) 을, 현재 비디오 블록과 동일한 LCU 내에 있을 때 현재 비디오 블록을 위한 컨텍스트를 도출하는데 이용가능한 것으로, 그리고 이웃하는 비디오 블록이 현재 LCU 의 밖에 있을 때 이용가능하지 않은 것으로 고려할 수도 있다.

[0114] 일 예로서, 현재 비디오 블록에 대해, 비디오 코더들은 컨텍스트 값 = (상단 IntraMC\_flag == 0) ? 0 : 1 + (좌측 IntraMC\_flag == 0) ? 0 : 1 을 도출할 수도 있다. 일부 예들에서, 비디오 코더들은 항상, 이웃하는 비디오 블록들 (262, 264, 266) 중의 특정 하나, 예를 들어, 좌측 이웃하는 비디오 블록 (262) 으로부터 현재 비디오 블록 (260) 을 위한 컨텍스트를 도출할 수도 있고, 현재 비디오 블록을 위한 컨텍스트 값은 (좌측 IntraMC\_flag == 0) ? 0 : 1 이다. 일부 예들에서, 비디오 코더들은, 비디오 블록, 예를 들어, CU 의 크기에 의존하는 컨텍스트 값을 이용하여 플래그를 코딩한다. 일부 예들에서, 상이한 크기들을 갖는 비디오 블록들은 상이한 컨텍스트 값들을 갖는다. 즉, 각각의 비디오 블록, 예를 들어, CU, 크기는 각각의 고유 컨텍스트 값과 연관될 수도 있다. 일부 예들에서, 상이한 미리정의된 크기들을 갖는 여러 비디오 블록들, 예를



들어, CU 들은 동일한 컨텍스트 값을 공유하고, 다른 크기들을 갖는 비디오 블록들은 하나 이상의 상이한 컨텍스트 값들을 이용한다. 즉, 비디오 코더들은 각각의 세트 또는 범위(들)의 가능한 비디오 블록, 예를 들어, CU, 크기들과 복수의 컨텍스트 값들의 각각을 연관시킬 수도 있다.

[0115] 일부 예들에서, 비디오 코더들, 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는, 단항 코드들에 기초하여 IntraBC 를 위한 2차원 벡터의 수평 변위 성분 및 수직 변위 성분 (예를 들어, 도 2에 있는 2차원 벡터 (106)의 수평 변위 성분 (112) 및 수직 변위 성분 (110))을 코딩할 수도 있다. 다른 예들에서, 비디오 코더들은 지수 골롬 (exponential Golomb) 또는 라이스 골롬 (Rice-Golomb) 코드들에 기초하여 수평 변위 성분 및 수직 변위 성분을 코딩할 수도 있다.

[0116] 일부 예들에서, 수평 및 수직 변위 성분들은 현재 비디오 블록의 상부 및 좌측의 구역들만을 나타낼 수도 있고, 비디오 코더들은 이들 값들에 대해 부호 비트 (sign bit) 들을 보유 또는 코딩할 필요가 없을 수도 있다. 일부 예들에서, 비디오 코더들은 현재 비디오 블록의 상부 및 좌측의 구역들이 현재 비디오 블록에 대하여 양의 방향들을 나타낼 수 있도록 참조의 프레임들을 구성할 수도 있다. 그러한 예들에서, 현재 비디오 블록의 상부 및/또는 좌측 비디오 블록들이 후보 예측 비디오 블록들로서 고려되면, 비디오 코더들은 부호 비트들을 보유 또는 코딩할 필요가 없을 수도 있는데, 왜냐하면 수평 변위 성분 및 수직 변위 성분의 모든 값들이 양의 (또는 음의) 값들을 나타내고 현재 비디오 블록의 상부 및/또는 좌측의 비디오 블록들을 표시한다고 사전-정의될 수도 있기 때문이다.

[0117] 일부 예들에서, 이들 2차원 벡터들의 최대 크기 (또는 잔차 2차원 벡터와 같은 하나 이상의 2차원 벡터들간의 차이)는, 예를 들어, 위에서 논의된 바처럼, 의도된 영역 및/또는 파이프라인 제약들의 정의에 기인하여, 작을 수도 있다. 그러한 예들에서, 비디오 인코더 (20)는 트렁케이티드 값들 (truncated value)로 이들 2차원 블록 벡터들을 2진화할 수도 있다. 일부 예들에서, 비디오 인코더 (20)는, 2차원 벡터들을 엔트로피 인코딩함에 있어서, 예를 들어, 2차원 벡터들의 수평 변위 성분 및 수직 변위 성분을 인코딩함에 있어서, 트렁케이티드 단항, 트렁케이티드 지수 골롬, 또는 트렁케이티드 라이스 코드들을 채용할 수도 있다.

[0118] 비디오 인코더 (20)가 여기에 기재된 다양한 트렁케이티드 인코딩 스킴들 중의 임의의 것에서 이용할 수도 있는 트렁케이션 값은, 예를 들어, LCU 크기에 기초하여, 일정 (constant) 할 수 있다. 일부 예들에서, 트렁케이션 값은 수평 변위 성분 및 수직 변위 성분에 대해 동일할 수도 있다. 다른 예들에서, 트렁케이션 값은 수평 변위 성분 및 수직 변위 성분에 대해 상이할 수도 있다.

[0119] 하나의 예시적인 예로서, LCU의 크기는 64, 예를 들어, 64x64 이고, 2차원 벡터들의 수직 성분들은, 도 3에 관하여 위에서 설명된 바처럼, LCU 내에 한정되면, 트렁케이션은 2차원 벡터의 수평 성분에 대해 63과 동일할 수 있고 2차원 벡터의 수직 성분에 대해  $63 - \text{MinCUsSize}$ 와 동일할 수 있다. 일부 예들에서, 트렁케이션 값은 LCU 내의 현재 비디오 블록의 위치에 따라 적응적일 수 있다. 예를 들어, 2차원 벡터의 수직 성분이 LCU 내에 한정되면, 비디오 인코더는 현재 비디오 블록의 상단 위치와 LCU의 상단 위치 사이의 차이로 벡터 이진화를 트렁케이팅할 수 있다.

[0120] 비디오 인코더 (20)는 바이패스 모드를 이용하여 2차원 벡터의 수평 및 수직 성분들의 2진화를 엔트로피 인코딩할 수도 있거나, 또는 2진화들을, 예를 들어, CABAC 컨텍스트로, 산술 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20)는 예를 들어, 도 3에 예시된 바처럼, 의도된 영역으로 예측 비디오 블록을 위한 서치를 한정하며, 2차원 벡터 (MV)의 성분들의 분포는 제로에 중심을 두지 않을 수도 있다. 예를 들어,  $MV_x$ 는 음이 되는 경향이 있는데, 왜냐하면 (동일 행에서) 현재 비디오 블록, 예를 들어, CU의 우측 상의 픽셀들이 인코딩/재구성되지 않았기 때문이다. 유사하게,  $MV_y$ 는 음이 되는 경향이 있는데, 왜냐하면 (동일 열에서) 현재 비디오 블록, 예를 들어, CU의 아래의 픽셀들이 인코딩/재구성되지 않았기 때문이다.

[0121] 바이패스 모드는 0 및 1에 대해 동일한 확률을 가정할 수도 있다. 부호에 대하여, 이것은 바이패스 모드가 양 또는 음이 되는 동일한 확률을 가진다는 것을 의미한다. 2차원 벡터의 성분들은 양 또는 음이 되는 동일한 확률을 갖지 않기 때문에, 비디오 인코더 (20)는, 부호를 컨텍스트로, 예를 들어, 0.5 외의 초기 확률로, 산술 인코딩, 예를 들어, CABAC 인코딩할 수도 있다.

[0122] 비디오 인코더 (20)가 2차원 벡터의 수평 성분 ( $MV_x$ )을 어떻게 인코딩할 수도 있는지의 일 예는 다음과 같다.  $MV_x$ 에 관하여 설명되었지만, 그 기법은 또한, 2차원 벡터의 수직 성분 ( $MV_y$ ), 또는 현재 벡터와 예측 벡터사이의 차이를 나타내는 잔차 2차원 벡터의 수평 또는 수직 성분들, 예를 들어,  $mvd_x$ 와  $mvd_y$ 를 인코딩하는데 적용될 수 있다.

- [0123]  $MV_x$  는 부호 값, 및  $(abs(MV_x))$  에 대한 2진화 스트링  $b_0 b_1 \dots$  에 의해 나타내어질 수도 있다. 제 1 빈  $b_0$  은  $abs(MV_x) > 0$  ( $b_0=1$ ) 인지 또는 그렇지 않은지 ( $b_0 = 0$ ) 를 표시한다. 비디오 인코더 (20) 는 컨텍스트로 CABAC 를 이용하여 제 1 빈  $b_0$  을 인코딩할 수도 있다.  $MV_x$  및  $MV_y$  에 대한  $b_0$  은 별개의 컨텍스트들을 가질 수도 있거나, 또는 동일한 컨텍스트들을 공유할 수도 있다. 일부 예들에서, IntraBC 의 MV 코딩에서  $i$  번째 빈은 인터 모션 보상의 블록 벡터 코딩에서  $i$  번째 빈들과 동일한 컨텍스트들을 공유한다. 일부 예들에서, IntraBC 의 MV 코딩 및 인터 모션 보상의 블록 벡터 코딩에서  $i$  번째 빈들은 컨텍스트들을 공유하지 않는다.
- [0124] 다음 빈들  $b_1 b_2 \dots$  은  $abs(MV_x) - 1$ 의 값을 나타내고, 비디오 인코더 (20) 는 바이패스 모드에서 파라미터 3으로 지수 곱셈 코드들을 이용하여 이들 빈들을 인코딩할 수도 있다. 일부 예들에서, 비디오 인코더 (20) 는 지수 곱셈 코드들의 다른 오더들, 예를 들어, 1, 2, 4, 5 를 이용한다. 일부 예들에서,  $b_1$  은  $abs(MV_x) = 1$  ( $b_1 = 1$ ) 또는 그렇지 않은지 ( $b_1 = 0$ ) 를 나타낸다. 일부 예들에서, 비디오 인코더 (20) 는 바이패스 모드로 또는 CABAC 컨텍스트로  $b_1$  을 인코딩할 수도 있다.
- [0125] 일부 예들에서,  $b_2 b_3 \dots$  은  $abs(MV_x) - 2$ 의 값을 나타내고, 비디오 인코더 (20) 는 바이패스 모드에서 파라미터 3, 또는 지수 곱셈 코드들의 다른 오더들로 지수 곱셈 코드들을 이용하여 이들 빈들을 인코딩할 수도 있다. 최종 빈은  $MV_x$  의 부호를 나타낼 수도 있고, 비디오 인코더 (20) 는 컨텍스트 없이 바이패스 모드에서 이 빈을 인코딩할 수도 있다. 다른 예들에서, 비디오 인코더 (20) 는, 하나 또는 다수의 컨텍스트들로 CABAC 를 이용하여 인코딩된 부호 빈을 인코딩할 수도 있다.  $MV_x$  및  $MV_y$  에 대한 부호 빈들은 별개의 컨텍스트들을 가질 수도 있거나, 또는 동일한 컨텍스트들을 공유할 수 있다.
- [0126] 일부 예들에서, 비디오 코더들, 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 IntraBC 모드의 적용, 및 플래그, 벡터 성분, 또는 다른 신택스의 연관된 코딩을, 특정 크기, 즉, 특정 크기 기준을 충족하는 비디오 블록들로 한정할 수도 있다. 크기 기준은, 최대 블록 크기 및/또는 최소 블록 크기 (예를 들어, IntraMC\_MaxSize 및 IntraMC\_MinSize) 중의 하나 또는 양자 모두일 수도 있다. 비디오 인코더 (20) 는, 예를 들어, 플래그 또는 다른 신택스 엘리먼트들을 이용하여, 인코딩된 비디오 비트스트림에서 비디오 디코더 (30) 에 하나 이상의 크기 기준을 표시할 수도 있거나, 또는 비디오 코더들은 다른 정보, 이를테면 특정 색공간 (예를 들어, YUV, RGB 등), 특정 컬러 포맷 (예를 들어, 4:2:0, 4:4:4 등), 프레임 크기, 프레임 레이트, 또는 양자화 파라미터 (QP) 에 기초하여, 또는 이전에 코딩된 프레임들에 기초하여, 크기 기준을 내재적으로 (implicitly) 결정할 수도 있다. 예를 들어, 작은 크기들을 위한 IntraBC 는 시스템의 메모리 대역폭 요구 사항들에 영향을 줄 수도 있고, 비디오 코더들은 크기 임계치보다 높은 블록들로 IntraBC 을 제한하기 위해 최소 블록 크기 기준을 적용할 수도 있다.
- [0127] 도 5a 및 도 5b 는 2개의 PU들 (즉, PU0 및 PU1) 들로 분할되는 CU 의 일 예를 보여준다. 도 5a 및 도 5b 는 HEVC AMVP 및 병합을 위한 공간적 이웃 도출의 예시 이미지를 보여준다. 공간적 이웃들은, 블록 벡터 코딩에, 이를테면 Intra BC AMVP 모드 또는 Intra BC 병합 모드를 위한 후보들을 결정하는데 이용될 수도 있다. 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 특정 PU (PU0) 에 대해, 도 5a 및 도 5b 에 도시된 이웃하는 블록들로부터 공간적 MV 후보들을 도출할 수도 있지만, 블록들로부터 후보들을 생성하는 방법들은 병합 및 AMVP 모드들에 대해 상이할 수도 있다. 병합 모드에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 도 5a 에 보여진 순서를 이용하여 4개에 이르기까지 공간적 MV 후보들을 도출할 수도 있다. 도 5a 에 보여진 바처럼, 순서는 다음과 같다: 좌측 (0), 상부 (1), 상부 우측 (2), 하부 좌측 (3), 및 상부 좌측 (4).
- [0128] AVMP 모드에서, 이웃하는 블록들은 다음 2개의 그룹들로 분할된다: 도 5b 에 나타낸 바처럼 블록 0 및 1 로 이루어지는 좌측 그룹, 및 블록 2, 3 및 4 로 이루어지는 상부 그룹. 각 그룹에 대해, 시그널링된 참조 인덱스에 의해 표시된 것과 동일한 참조 화상을 참조하는 이웃하는 블록에 있는 잠재적인 후보는 그룹의 최종 후보를 형성하기 위해 선택된 최고 우선순위를 가진다. 모든 이웃하는 블록들은 동일한 참조 화상을 가리키는 블록 벡터를 포함하지 않는 것이 가능할 수도 있다. 그러므로, 그러한 후보가 구해질 수 없으면, 첫번째 이용가능한 후보는 최종 후보를 형성하기 위하여 스케일링될 수도 있으며, 이렇게 하여 시간 거리 차이가 보상될 수 있다.
- [0129] Intra BC 를 위한 현재 기법들은 몇몇 잠재적인 문제들을 가질 수도 있다. 일 예에서, 크로마 샘플링 포맷이 4:4:4 이 아닐 때 (즉, 4:2:0 또는 4:2:2 중 어느 하나 일 때), 크로마 모션 보상 블록 크기는 2x2 만큼 작을 수도 있으며, 이렇게 하여 베이스 HEVC 규격에서 지원되는 최소 블록 크기보다 더 작을 수도 있다. 참조 블록들에 액세스할 때, 비디오 코더는 통상적으로 미리정의된 메모리 블록들 (예를 들어, 4x2, 8x2) 의 유닛들

에서 블록들을 판독한다. 따라서, 블록 크기가 미리정의된 메모리 블록 유닛들보다 더 작으면, 보상에서 이용되지 않을 수도 있는 추가의 픽셀들이 액세스되어, 메모리 액세스 효율을 떨어트린다.

[0130] 잠재적인 문제의 또 다른 예로서, 크로마 성분을 위한 루마 블록을 단순히 스케일링하는 것은 현재 CU 또는 PU 내의 영역을 가리키는 블록 벡터를 낳을 수도 있다. 또 다른 예로서, 블록 벡터 예측의 AMVP 방식은 Intra BC의 기존 구현들에서 잘 지원되지 않는다.

[0131] 본 개시는 Intra BC 예측을 향상시킬 수도 있는 기법들을 도입한다. 본 개시의 일 양태에 따르면, 코딩된 비디오 데이터가 4:2:0의 크로마 샘플링 포맷을 가질 때, 비디오 디코더 (30)는, 대응하는 루마 성분을 예측하는데 이용된 블록 벡터와는 상이한 블록 벡터를 크로마 성분을 예측하는데 이용할 수도 있다. 예를 들어, 제 1 루마 성분 및 그의 대응하는 제 1 크로마 성분을 위한 블록 벡터는 동일할 수도 있지만, 제 2 크로마 성분을 위한 블록 벡터는 그의 대응하는 제 2 루마 성분을 위한 블록 벡터와는 동일하지 않을 수도 있다. 오히려, 제 2 크로마 성분을 위한 블록 벡터는 제 1 루마 성분을 위한 블록 벡터와 동일할 수도 있다.

[0132] 비디오 디코더 (30)는 루마에 이용된 블록 벡터로부터 크로마를 위한 블록 벡터를 도출할 수도 있다. 예를 들어, 4개의 NxN PU 들로 분할된 2Nx2N CU에 대해, 비디오 인코더 (20)는, 현재 CU에서 4개의 PU/TU를 위해, 4개의 BV들, 즉 코딩 순서로 BV<sub>1</sub>, BV<sub>2</sub>, BV<sub>3</sub>, 및 BV<sub>4</sub>를 시그널링할 수도 있다. 본 개시의 기법들에 따르면, 비디오 디코더 (30)는 현재 CU의 루마 성분의 Intra BC 예측을 위해 그리고 전체 CU의 다수의 크로마 블록들의 IntraBC 예측을 위한 하나의 단일 블록 벡터의 도출을 위해 4개의 BV들을 이용할 수도 있다. 따라서, 4개의 PU/TU들을 포함하는 CU는 다수의 PU들의 크로마 블록들을 위해 (BV<sub>c</sub>로 표기되는) 하나의 공통 블록 벡터를 공유할 수도 있다.

[0133] 비디오 디코더 (30)는, 크로마 성분들을 위한 모션 보상 전에, HEVC 버전 1에서 행해진 바처럼 루마 블록 벡터를 크로마 블록 벡터로 컨버팅하거나, 또는 루마 블록 벡터를 크로마 블록 벡터로서 직접 이용하여, 4개의 루마 블록 벡터들로부터 공통 크로마 블록 벡터를 생성할 수도 있다. 비디오 디코더 (30)는 다음 방법들 중 하나의 방법에 의해 루마 블록 벡터 BV<sub>L</sub>를 도출할 수도 있다.

[0134] 1. 비디오 디코더 (30)는 BV<sub>1</sub>, BV<sub>2</sub>, BV<sub>3</sub>, BV<sub>4</sub> 중에서 최소 블록 벡터 BV<sub>L</sub>를 선택하고, BV<sub>L</sub>를, 예를 들어, 스케일링에 의해, BV<sub>c</sub>로 컨버팅한다. 설명의 목적을 위하여, 블록 벡터 BV<sub>a</sub>는 다음 가능한 방식들 중의 하나에서 다른 BV<sub>b</sub>보다 더 작은 것으로 정의된다.

[0135] a. BV<sub>a</sub> (BV<sub>a</sub>[0])의 수평 성분이 BV<sub>b</sub>의 그것보다 더 작은 경우

[0136] b. BV<sub>a</sub>의 수평 성분의 절대 값 (값 a에 대해, |x|로 표기됨)이 BV<sub>b</sub>의 그것보다 더 작은 경우

[0137] c. |BV<sub>a</sub>[0]| + |BV<sub>a</sub>[1]|이 |BV<sub>b</sub>[0]| + |BV<sub>b</sub>[1]|의 그것보다 더 작은 경우

[0138] d. BV<sub>a</sub> (BV<sub>a</sub>[1])의 수직 성분이 BV<sub>b</sub>의 그것보다 더 작은 경우

[0139] 2. 위의 최소 블록 벡터 선택들과 유사하게, 비디오 디코더 (30)는 4개의 블록 벡터들 중에서 최대 블록 벡터를 선택할 수도 있으며, 위에서 정의된 바처럼 하나의 벡터가 다른 것보다 더 작다는 다양한 정의들 중 하나를 갖는다.

[0140] 3. 비디오 디코더 (30)는, 루마 블록 벡터 BV<sub>L</sub>를 도출하기 위하여 4개의 루마 블록 벡터들에 대해 수평 및 수직 성분들을 평균화하는 것에 의해, 평균 블록 벡터를 결정할 수도 있고, 여기서 +2/-2를 추가하는 것의 라운딩 (rounding)은 평균화 동안 적용될 수도 있거나 또는 그렇지 않을 수도 있다. BV<sub>L</sub>은 공통 크로마 블록 벡터를 결정하기 위하여 스케일링될 수도 있다.

[0141] 4. 비디오 디코더 (30)는 BV<sub>1</sub>, BV<sub>2</sub>, BV<sub>3</sub>, 및 BV<sub>4</sub>로부터 하나의 블록 벡터를 BV<sub>L</sub>로서 선택할 수도 있다. 선택된 블록 벡터는, 예를 들어, 4개 중에서 고정된 하나의 블록 벡터, 이를테면 항상 BV<sub>1</sub>, 항상 BV<sub>2</sub> 등일 수도 있거나, 또는 비트스트림에서 명시적으로 시그널링될 수도 있다.

[0142] 본 개시의 다른 양태에 따르면, 코딩된 비디오 데이터가 4:2:2의 크로마 샘플링 포맷을 가질 때, 비디오 디코더 (30)는, 루마를 위해 이용된 BV와는 상이한 BV를 크로마를 위해 이용할 수도 있다. 그 대신에, 비디오 디코더 (30)는 루마를 위해 이용되는 BV (들)로부터 크로마를 위해 이용되는 블록 벡터를 도출할 수도 있

다. 예를 들어, IntraBC 가 PU/TU 로 확장될 수도 있고 PU/TU 가  $2N \times 2N$  CU 에 대해  $N \times N$  이라고 생각하면, 현재 CU 에서 2개의 PU/TU들에 대해, 2개의 BV 들, 즉, 코딩 순서로,  $BV_1$ , 및  $BV_2$ 이 있다. 위에 설명된 기법들과 유사하게, 최소 벡터, 최대 벡터, 또는 평균 벡터는 전체 CU 의 크로마 블록을 위해 현재 PU 의 2개 PU/TU 들의 2개 블록 벡터들로부터 생성될 수도 있다.

[0143] 본 개시의 일 기법에 따르면, 4개  $N \times N$  Intra BC PU/TU들로 코딩된  $2N \times 2N$  CU 에 대해, 비디오 디코더 (30) 는, 크로마 샘플링 포맷이 4:4:4 가 아닐 때,  $N \times N$  PU/TU 들의 블록 벡터들로부터 생성된 단일 블록 벡터로 크로마 성분을 예측할 수도 있다. 하지만, 비디오 디코더 (30) 는, PU 들의 루마 성분들을 예측하기 위해 4개의 별개의 블록 벡터들을 여전히 이용할 수도 있다. 예를 들어, 4개의  $N \times N$  PU 들로 분할된  $2N \times 2N$  CU 에 대해, 비디오 인코더 (20) 는 CU 의 각 PU 에 대해 하나씩, 4개의 별개의 블록 벡터들을 시그널링할 수도 있다. 비디오 디코더 (30) 는 그러한 4개의 별개의 블록 벡터들을 디코딩하고 그 4개의 별개의 블록 벡터들을 이용하여 4개의 PU 들의 루마 성분들을 예측할 수도 있다. 하지만, 4개의 PU 들의 크로마 성분들에 대하여, 비디오 디코더 (30) 는, 4개의 블록 벡터들 중 하나를 선택하고 그 하나의 블록 벡터를 이용하여 CU 의 PU 들 중 하나보다 많은 PU 들을 위한 크로마 성분들을 예측할 수도 있다. 일부 경우들에서, 비디오 디코더 (30) 는 CU 의 모든 4개의 PU 들을 위한 크로마 성분들을 예측하는데 하나의 블록 벡터를 이용할 수도 있다.

[0144] 다른 예에서, 4개  $N \times N$  Intra BC PU/TU들로 코딩된  $2N \times 2N$  CU 에 대해, 비디오 디코더 (30) 는, 크로마 샘플링 포맷이 4:4:4 외의 다른 것일 때,  $N \times N$  PU/TU 들의 블록 벡터들로부터 생성된 2개의 블록 벡터들로 크로마 성분을 예측할 수도 있다. 2개의 블록 벡터들의 각각은, 하나의  $N/2 \times N$  블록 또는  $N \times N/2$  블록을 예측하는데 이용될 수도 있다. 즉, 2개의 블록 벡터들의 각각은 CU 의 4개 PU 들 중 2개를 예측하는데 이용될 수도 있다. 비디오 디코더 (30) 는 체크되는 CU 의 PU 들의 4개 블록 벡터들 중 적어도 하나를 체크할 수도 있고 2개의 루마 블록 벡터들은 이들 블록 벡터들에 따라 도출된다. 보다 구체적으로, 예를 들어, 4개의 블록 벡터들이 비교되고 어느 다른 쌍보다 거리가 더 가까운 2개의 이웃하는 블록 벡터들이 선택될 수도 있다.

[0145] 일 예에서,  $\text{dis}(BV_0, BV_1)$ ,  $\text{dis}(BV_0, BV_2)$ ,  $\text{dis}(BV_1, BV_3)$  및  $\text{dis}(BV_2, BV_3)$  가 계산된다. (1) 가장 작은 것은  $2N \times N$  또는  $N \times 2N$  블록으로 조합될 수도 있는 쌍에 대응한다. 예를 들어,  $\text{dis}(BV_0, BV_1)$  이 가장 작으면, 상단 2개의  $N \times N$  블록들은  $2N \times N$  블록으로 조합된다. 남아있는 것들은 자동적으로 다른  $2N \times N$  블록이 된다. (2) 가장 큰 것은  $2N \times N$  또는  $N \times 2N$  블록으로 조합되지 않을 수도 있는 쌍에 대응한다. 예를 들어,  $\text{dis}(BV_0, BV_1)$  이 가장 크면, 좌측 2개의  $N \times N$  블록들이  $2N \times N$  블록으로 조합된다. 남아있는 것들은 자동적으로 다른  $2N \times N$  블록이 된다. (3) 각 조합된  $2N \times N$  블록에 대하여, 2개의 블록 벡터들이 프로세싱되고 단일 루마 블록 벡터를 도출한다. (4) 2개의 블록 벡터들의 거리는 간단히, 2개 블록 벡터들  $BV_a$  및  $BV_b$  에 대하여,  $(BV_a[0]-BV_b[0])$  의 절대 값 플러스  $(BV_a[1]-BV_b[1])$  의 절대 값으로서 정의될 수도 있다. 대안적으로, 음이 아닌 값들을 반환하는 다른 거리 정의들이 또한 적용될 수도 있다.

[0146] 본 개시의 다른 기법들에 따르면, 비디오 데이터가 4:2:0 또는 4:2:2 포맷에 따라 코딩될 때, 비디오 디코더 (30) 는 4개의  $N \times N$  PU/TU 들의 4개의 블록 벡터들로부터 하나의 모션을 도출할 수도 있다. 하나의 예에서, 비디오 디코더 (30) 는 하단 우측 블록 (즉, 하단 우측 PU) 를 위한 블록 벡터를 도출하고 이 블록 벡터를 이용하여 CU 의 4개 PU 들을 예측할 수도 있다. 하단 우측 PU 의 블록 벡터를 선택함으로써, 블록 벡터가 CU 의 또 다른 PU 를 가리키지 않는다는 것이 확보될 수도 있다.

[0147] 다른 예들에서, 비디오 디코더 (30) 는 하단 좌측 블록, 상단 우측 블록, 상단 좌측 블록의 블록 벡터가 이 CU 의 크로마 블록들을 위한 블록 벡터를 생성하는 유일한 것으로 선택되는 것을 선택할 수도 있다.

[0148] 본 개시의 다른 기법에 따르면, 비디오 데이터가 4:2:2 포맷에 따라 코딩될 때, 하나의 블록 벡터는 2개의 블록 벡터들로부터 도출될 수도 있다. 그러한 구현에서, 4개의  $N \times N$  PU들로 분할되는  $2N \times 2N$  CU 에 대해, 비디오 디코더 (30) 는 동일한 블록 벡터를 이용하여 CU 의 4개의 PU 들 중 2개를 예측할 수도 있다. 일 예에서, 비디오 디코더 (30) 는 상단 우측 PU 및 상단 좌측 PU 양자 모두를 예측하는데 상단 우측 블록 (즉, 상단 우측 PU) 의 블록 벡터를 이용할 수도 있다. 이 예에서, 비디오 디코더 (30) 는 하단 좌측 PU 및 하단 우측 PU 를 예측하는데 하단 우측 블록 (즉, 하단 우측 PU) 의 블록 벡터를 이용할 수도 있다. 다른 예에서, 비디오 디코더 (30) 는 CU 의 상단 2개 크로마 블록들을 예측하는데 상단 좌측 블록의 블록 벡터를 이용하고 하단 2개의 크로마 블록들을 예측하는데 하단 좌측 블록의 블록 벡터를 이용할 수도 있다.

[0149] 비디오 데이터가 4:2:0 또는 4:2:2 포맷에 따라 코딩될 때 그리고  $2N \times 2N$  CU 가  $2N \times N$  또는  $N \times 2N$  IntraBC PU/TU



들로 분할될 때, 비디오 디코더 (30) 는 상이한 블록 벡터들로 PU/TU 의 크로마 블록들을 예측할 수도 있다.

본 개시의 다른 기법에 따르면, 비디오 디코더 (30) 는 2개의 루마 블록 벡터들로부터 하나의 블록 벡터를 도출할 수도 있고 전체 CU 의 크로마 블록들 양자 모두를 예측하는데 하나의 블록 벡터를 이용할 수도 있다. 하나의 예에서, 비디오 디코더 (30) 는 ( $N \times 2N$  로 코딩될 때) 우측 블록의 블록 벡터, 또는 ( $2N \times N$  로 코딩될 때) 하단 블록의 블록 벡터를 CU 의 크로마 블록들을 위한 블록 벡터로서 선택할 수도 있다. 다른 예에서, 비디오 디코더 (30) 는 ( $N \times 2N$  로 코딩될 때) 좌측 블록의 블록 벡터, 또는 ( $2N \times N$  로 코딩될 때) 상단 블록의 블록 벡터를 CU 의 크로마 블록들을 예측하기 위한 블록 벡터로서 선택할 수도 있다.

[0150] 비디오 디코더 (30) 가 루마 PU들로부터 블록 벡터를 어떻게 도출하는지 (도출된 블록 벡터를 dBV 로 표기함) 에 상관 없이, 최종 크로마 블록 벡터를 형성하기 위하여, 비디오 디코더 (30) 는 도출된 루마 블록 벡터를 클립핑하여 리걸 블록 벡터 (legal block vector) 를 생성할 수도 있다. 일 예에서, 비디오 디코더 (30) 는, 크로마 블록 벡터가 동일한 PU/TU 내의 픽셀을 가리키지 않는 방식으로 클립핑을 수행할 수도 있다. 다른 예에서, 비디오 디코더 (30) 는, 크로마 블록 벡터가 동일한 CU 내의 픽셀을 가리키지 않는 방식으로 클립핑을 수행할 수도 있다.

[0151]  $2N \times 2N$  CU 가 4개의  $N \times N$  PU 들로 분할되는 경우에 대해, 비디오 디코더 (30) 는 수평 클립핑 동작 및 수직 클립핑 동작을 수행할 수도 있다. 비디오 디코더 (30) 는 수평 클립핑을 다음과 같이 적용할 수도 있다:

[0152] 수평 클립:  $\max(-\text{leftW}-(\text{CUx}\% \text{ctbW}), \text{dBV}[0])$ , 여기서

[0153] 1. leftW 는 예를 들어, 64 의 고정 값일 수 있거나, 또는 현재 CTU 의 폭, 또는 비트스트림에서, 예를 들어, 하이 레벨 선택스 이클테면 비디오 파라미터 세트, 시퀀스 파라미터 세트 및 화상 파라미터 세트에서 송신되는 임의의 다른 값일 수 있다.

[0154] 2. CUx 는 현재 CU 의 상단 좌측 픽셀의 수평 좌표이다.

[0155] 3. ctbW 는 CTB 의 폭이다

[0156] 수직 클립:  $\max(-\text{CUy}\% \text{ctbH}, \text{dBV}[1])$ , 여기서

[0157] 1. CUy 는 현재 CU 의 상단 좌측 픽셀의 수직 좌표이다.

[0158] 2. ctbH 는 CTB 의 높이이다.

[0159] 일부 예들에 따르면, 비디오 디코더 (30) 는, 예를 들어,  $2N \times 2N$  CU 가  $N \times 2N$  PU 들로 분할될 때 수평 클립핑만을 수행할 수도 있거나 또는  $2N \times 2N$  CU 가  $2N \times N$  PU 들로 분할될 때 수직 클립핑만을 수행할 수도 있다. 위에 설명된 경우들에서, 2개의  $N \times N$  블록들로부터 2개의 블록 벡터들이  $N \times 2N$  블록을 위해 조합되어야 할 때, 비디오 디코더 (30) 는 수평 클립핑만을 적용할 수도 있다. 위에 설명된 경우들에서, 2개의  $N \times N$  블록들로부터 2개의 블록 벡터들이  $2N \times N$  블록을 위해 조합되어야 할 때, 비디오 디코더 (30) 는 수직 클립핑만을 적용할 수도 있다.

[0160] 일부 구현들에서, 위에서 설명된 다양한 프로세스들은 어떤 CU 크기들에 대해서만 구현될 수도 있다. 예를 들어, 비디오 디코더 (30) 는 오직,  $8 \times 8$  이하인 CU 크기들에 대해 다수의 크로마 블록들을 예측하기 위해 단일 블록 벡터를 이용할 수도 있다. 여기서 설명된 기법들은, 예를 들어,  $32 \times 32$ ,  $16 \times 16$ , 또는  $8 \times 8$  와 같은 미리 정의된 크기 이하인 CU 크기들에 대해서만 적용될 수도 있다. 이 크기는 미리정의될 수도 있거나, 또는 대안적으로, sps/pps/슬라이스 레벨에서 시그널링될 수도 있다.

[0161] 도 6은 본 개시에 기재된 기법들을 구현할 수도 있는 일 예의 비디오 인코더를 예시하는 블록도이다. 비디오 인코더 (20) 는, 비디오 슬라이스들 내의 비디오 블록들의 인트라 코딩 및 인터 코딩을 수행할 수도 있다.

인트라 코딩은, 정해진 비디오 프레임 또는 화상 내의 비디오에서 공간적 중복성을 감소 또는 제거하기 위하여 공간적 예측에 의존한다. 인터 코딩은 비디오 시퀀스의 인접 프레임들 또는 화상들 내의 비디오에서 중복성을 감소 또는 제거하기 위하여 시간 또는 시점간 예측에 의존한다. 인트라 모드 (I 모드) 는 여러 공간 기반 압축 모드들 중 어느 것을 나타낼 수도 있다. 인터 모드들, 이클테면 단방향 예측 (P 모드) 또는 양방향 예측 (B 모드) 은, 여러 시간 기반 압축 모드들 중 어느 것으로 포함할 수도 있다. 비디오 인코더 (20) 는 또한, 동일한 화상 내에서 비디오 데이터의 예측 블록들로부터 비디오 데이터의 블록들의 인트라 예측을 위한 모드, 예를 들어, IntraBC 또는 IntraMC 모드를 여기에 기재된 바처럼 이용하도록 구성될 수도 있다.

[0162] 도 6 의 예에서, 비디오 인코더 (20) 는, 비디오 데이터 메모리 (334), 파티셔닝 유닛 (335), 예측 프로세싱 유닛 (341), 의도된 영역 메모리 (364), 필터 프로세싱 유닛 (366), 참조 화상 메모리 (368), 합산기 (350), 변

환 프로세싱 유닛 (352), 양자화 프로세싱 유닛 (354), 및 엔트로피 인코딩 유닛 (356) 을 포함한다. 예측 프로세싱 유닛 (341) 은 모션 추정 유닛 (342), 모션 보상 유닛 (344), 인트라 예측 프로세싱 유닛 (346), 및 인트라 블록 카피 (IntraBC) 유닛 (348) 을 포함한다. 비디오 블록 재구성을 위해, 비디오 인코더 (20) 는 또한 역 양자화 프로세싱 유닛 (358), 역 변환 프로세싱 유닛 (360), 및 합산기 (362) 를 포함한다.

[0163] 다양한 예들에서, 비디오 인코더 (20) 의 유닛은 본 개시의 기법들을 수행하도록 태스킹될 수도 있다. 또한, 일부 예들에서, 본 개시의 기법들은 비디오 인코더 (20) 의 유닛들 중의 하나 이상 중에서 분할될 수도 있다. 예를 들어, IntraBC 유닛 (348) 은 본 개시의 기법들을 단독으로, 또는 비디오 인코더의 다른 유닛들과 조합하여, 이를테면 모션 추정 유닛 (342), 모션 보상 유닛 (344), 인트라 예측 프로세싱 유닛 (346), 의도된 영역 메모리 (364), 및 엔트로피 인코딩 유닛 (356) 과 조합하여, 수행할 수도 있다.

[0164] 비디오 데이터 메모리 (334) 는 비디오 인코더 (20) 의 컴포넌트들에 의해 인코딩될 비디오 데이터를 저장할 수도 있다. 비디오 데이터 메모리 (334) 에 저장된 비디오 데이터는, 예를 들어, 비디오 소스 (18) 로부터 획득될 수도 있다. 의도된 영역 메모리 (364) 는, 예를 들어, 인트라- 또는 인터-코딩 모드들에서 비디오 인코더 (20) 에 의한 비디오 데이터를 인코딩하는데 이용하기 위한 참조 비디오 데이터를 저장하는 참조 블록 메모리일 수도 있다. 비디오 데이터 메모리 (334) 및 의도된 영역 메모리 (364) 는 동기 DRAM (SDRAM) 을 포함하는 동적 랜덤 액세스 메모리 (DRAM), 자기 저항 RAM (MRAM), 저항 RAM (RRAM), 또는 다른 타입의 메모리 디바이스들 등의 다양한 메모리 디바이스들 중의 어느 것에 의해 형성될 수도 있다. 비디오 데이터 메모리 (334) 및 의도된 영역 메모리 (364) 는 동일한 메모리 디바이스 또는 별개의 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에서, 비디오 데이터 메모리 (334) 는 비디오 인코더 (20) 의 다른 컴포넌트들과 함께 온칩 (on-chip) 될 수도 있거나, 또는 그러한 컴포넌트들에 대해 오프칩 (off-chip) 될 수도 있다.

[0165] 파티셔닝 유닛 (335) 은 비디오 데이터 메모리 (334) 로부터의 데이터를 비디오 블록들로 파티셔닝한다. 이 파티셔닝은 또한, 예를 들어, LCU 및 CU 들의 쿼드트리 구조에 따른, 비디오 블록 파티셔닝 뿐만 아니라 슬라이스들, 타일들 또는 다른 더 큰 유닛들로의 파티셔닝을 포함할 수도 있다. 비디오 인코더 (20) 는 일반적으로, 인코딩될 비디오 슬라이스 내의 비디오 블록들을 인코딩하는 컴포넌트들을 예시한다. 슬라이스는, 다수의 비디오 블록들로 (그리고 가능하게는 타일들로 지칭되는 비디오 블록들의 세트들로) 분할될 수도 있다.

[0166] 예측 프로세싱 유닛 (341) 은, 복수의 가능한 코딩 모드들 중 하나, 이를테면 복수의 인트라 코딩 모드들 중 하나, 복수의 인터 코딩 모드들 중 하나, 또는 IntraBC 모드를 본 개시에 설명된 기법들에 따라 여러 결과 (예를 들어, 코딩 레이트 및 왜곡의 레벨) 에 기초하여 선택할 수도 있다. 예측 프로세싱 유닛 (341) 은 결과적인 예측 블록을 합산기 (250) 에 제공하여 잔차 블록 데이터를 생성하고 합산기 (262) 에 제공하여 예를 들어, 참조 화상으로서, 다른 비디오 블록들의 예측에서의 이용을 위해 현재 블록을 재구성할 수도 있다.

[0167] 예측 프로세싱 유닛 (341) 내의 인트라 예측 프로세싱 유닛 (346) 은 코딩될 현재 블록과 동일한 프레임 또는 슬라이스에서의 하나 이상의 이웃 블록들에 대해 현재 비디오 블록의 인트라 예측 코딩을 수행하여 공간적 압축을 제공할 수도 있다. 예측 프로세싱 유닛 (341) 내의 모션 추정 유닛 (342) 및 모션 보상 유닛 (344) 은 하나 이상의 참조 화상들에서의 하나 이상의 예측 블록들에 대해 현재 비디오 블록의 인터 예측 코딩을 수행하여, 예를 들어, 시간적 압축을 제공한다.

[0168] 모션 추정 유닛 (342) 은, 비디오 시퀀스에 대해 미리결정된 패턴에 따라 비디오 슬라이스를 위한 인터 예측 모드를 결정하도록 구성될 수도 있다. 모션 추정 유닛 (342) 및 모션 보상 유닛 (344) 은 고도로 통합될 수도 있지만, 개념적인 목적을 위해 따로따로 예시되어 있다. 모션 추정 유닛 (342) 에 의해 수행되는, 모션 추정은, 비디오 블록들을 위한 모션을 추정하는 모션 벡터들을 생성하는 프로세스이다. 예를 들어, 모션 벡터는, 참조 화상 내의 예측 블록에 대해 현재 비디오 프레임 또는 화상 내의 비디오 블록의 PU 의 변위를 나타낼 수도 있다. 유사하게, 본 개시의 기법들에 따른 IntraBC 에 이용되는 2차원 벡터들은 현재 비디오 프레임 또는 화상 내의 비디오 블록의 PU의 변위로서, 동일한 프레임 또는 화상 내의 예측 블록에 대한, 상기 PU의 변위를 나타낸다. IntraBC 유닛 (348) 은, 인터 예측을 위한 모션 추정 유닛 (342) 에 의한 모션 벡터들의 결정에 유사한 방식으로 IntraBC 코딩을 위해, 2차원 벡터들, 예를 들어, 블록 벡터 또는 모션 벡터들을 결정할 수도 있거나, 또는 2차원 벡터들을 결정하기 위하여 모션 추정 유닛 (342) 을 이용할 수도 있다.

[0169] 예를 들어, 인터 예측 또는 IntraBC 예측을 위한 IntraBC 유닛 및/또는 모션 추정 유닛 (342) 에 의해 식별되는, 예측 블록은, 절대 차이의 합 (sum of absolute difference; SAD), 제곱 차이의 합 (sum of square difference; SSD), 또는 다른 차이 메트릭들에 의해 결정될 수도 있는, 픽셀 차이에 관하여, 코딩될 비디오 블록의 PU에 밀접하게 매치하는 것으로 구해진 블록이다. 일부 예들에서, 비디오 인코더 (20) 는 참조 화상

메모리 (368) 에 저장된 참조 화상들의 서브 정수 픽셀 위치들 또는 의도된 영역 메모리 (364) 에 저장된 현재 화상의 의도된 영역을 위한 값들 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 보간할 수도 있다. 그러므로, 모션 추정 유닛 (342) 및/또는 IntraBC 유닛 (348) 은, 전 (full) 픽셀 위치들 그리고 분수 픽셀 위치들에 대해 예측 블록을 위한 서치를 수행하고 분수 픽셀 정밀도로 벡터를 출력할 수도 있다.

[0170] 모션 추정 유닛 (342) 은, PU 의 위치와 참조 화상의 예측 블록의 위치를 비교함으로써 인터 코딩된 슬라이스에서 비디오 블록의 PU를 위한 모션 벡터를 계산한다. 참조 화상은, 제 1 참조 화상 리스트 (리스트 0 또는 RefPicList0) 또는 제 2 참조 화상 리스트 (리스트 1 또는 RefPicList1)로부터 선택될 수도 있고, 이들의 각각은 참조 화상 메모리 (368) 에 저장된 하나 이상의 참조 화상들을 식별한다. 모션 추정 유닛 (342) 은 계산된 모션 벡터를 엔트로피 인코딩 유닛 (356) 및 모션 보상 유닛 (344) 으로 전송한다.

[0171] 모션 보상 유닛 (344) 에 의해 수행된 모션 보상은, 모션 추정에 의해 결정된 모션 벡터에 기초한 예측 블록의 페칭 (fetching) 또는 생성을 수반할 수도 있으며, 가능하게는 서브-픽셀 정밀도로 보간을 수행한다. 현재 비디오 블록의 PU를 위한 모션 벡터의 수신시에, 모션 보상 유닛 (344) 은, 모션 벡터가 참조 화상 리스트들 중 하나에서 가리키는 예측 블록을 로케이팅할 수도 있다.

[0172] 일부 예들에서, IntraBC 유닛 (348) 은 2차원 벡터들을 생성하고 모션 추정 유닛 (342) 및 모션 보상 유닛 (344) 에 대해 위에서 설명된 것과 유사한 방식으로 예측 블록들을 페칭할 수도 있지만, 그 예측 블록들은 현재 블록과 동일한 화상 또는 프레임이다. 다른 예들에서, IntraBC 유닛 (348) 은 모션 추정 유닛 (342) 및 모션 보상 유닛 (344) 을 전체적으로 또는 부분적으로 이용하여, 여기에 기재된 기법들에 따라 IntraBC 예측을 위한 그러한 기능들을 수행할 수도 있다. 어떤 경우든, IntraBC 에 대해, 예측 블록은, 절대 차이의 합 (SAD), 제곱 차이의 합 (SSD), 또는 다른 차이 메트릭들에 의해 결정될 수도 있는, 픽셀 차이에 관하여, 코딩될 블록과 밀접하게 매치하는 것으로 구해진 블록일 수도 있고 그 블록의 식별은 서브 정수 픽셀 위치들을 위한 값들의 계산을 포함할 수도 있다.

[0173] 예측 비디오 블록이 IntraBC 예측에 따라 동일한 화상으로부터 또는 인터 예측에 따라 상이한 화상으로부터 기인하는지간에, 비디오 인코더 (20) 는, 코딩되는 현재 비디오 블록의 픽셀 값들로부터 예측 블록의 픽셀 값들을 감산하여, 픽셀 차이 값들을 형성함으로써 잔차 비디오 블록을 형성할 수도 있다. 픽셀 차이 값들은, 블록을 위한 잔차 데이터를 형성하고, 루마 및 크로마 차이 성분들 양자 모두를 포함할 수도 있다. 합산기 (250) 는 이 감산 연산을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다. IntraBC 유닛 (348) 및/또는 모션 보상 유닛 (344) 은 또한, 비디오 슬라이스의 비디오 블록들을 디코딩함에 있어서 비디오 디코더 (30) 에 의한 이용을 위해 비디오 블록들 및 비디오 슬라이스와 연관된 신텍스 엘리먼트들을 생성할 수도 있다. 시스템 엘리먼트들은, 예를 들어, 예측 블록을 식별하는데 이용된 벡터를 정의하는 신텍스 엘리먼트들, 예측 모드를 표시하는 임의의 플래그들, 또는 본 개시의 기법들에 관하여 설명된 임의의 다른 신텍스를 포함할 수도 있다.

[0174] 인트라 예측 프로세싱 유닛 (346) 은, 상술된 바와 같이, 모션 추정 유닛 (342) 및 모션 보상 유닛 (344) 에 의해 수행되는 인터 예측, 또는 IntraBC 유닛 (348) 에 의해 수행되는 IntraBC 예측에 대한 대안으로서, 현재 블록을 인트라 예측할 수도 있다. 특히, 인트라 예측 프로세싱 유닛 (346) 은 현재 블록을 인코딩하는데 이용할 인트라 예측 모드를 결정할 수도 있다. 일부 예들에서, 인트라 예측 프로세싱 유닛 (346) 은, 예를 들어, 별개의 인코딩 패스들 동안에, 다양한 인트라 예측 모드들을 이용하여 현재 비디오 블록을 인코딩할 수도 있고, 인트라 예측 프로세싱 유닛 (346) (또는, 일부 예들에서, 예측 프로세싱 유닛 (341)) 은 테스트된 모드들로부터 이용할 적절한 인트라 예측 모드를 선택할 수도 있다. 예를 들어, 인트라 예측 프로세싱 유닛 (346) 은 다양한 테스트된 인트라 예측 모드들을 위해 레이트 왜곡 분석을 이용하여 레이트 왜곡 값들을 산출하고, 테스트된 모드들 중에서 최상의 레이트 왜곡 특성들을 갖는 인트라 예측 모드를 선택할 수도 있다. 레이트 왜곡 분석은 일반적으로, 인코딩된 블록과, 인코딩된 블록을 생성하기 위해 인코딩되었던 원래의 인코딩되지 않은 블록 사이의 왜곡 (또는 에러) 의 양 뿐만 아니라 인코딩된 블록을 생성하는데 이용된 비트 레이트 (즉, 비트들의 수) 를 결정한다. 인트라 예측 프로세싱 유닛 (346) 은 그 왜곡들로부터 비 (ratio) 및 여러 인코딩된 블록들을 위한 레이트들을 산출하여 어느 인트라 예측 모드가 블록을 위한 최상의 레이트 왜곡 값을 나타내는지를 결정할 수도 있다.

[0175] 어느 경우든, 블록에 대한 인트라 예측 모드를 선택한 후에, 인트라 예측 프로세싱 유닛 (346) 은 엔트로피 인코딩 유닛 (356) 에 블록에 대한 선택된 인트라 예측 모드를 나타내는 정보를 제공할 수도 있다. 엔트로피 인코딩 유닛 (356) 은 본 개시의 기법들에 따라 선택된 인트라-예측 모드를 나타내는 정보를 인코딩할 수도 있

다. 비디오 인코더 (20) 는 송신된 비트스트림에서 구성 데이터를 포함할 수도 있고, 이는 복수의 인트라 예측 모드 인덱스 테이블들 및 복수의 수정된 인트라 예측 모드 인덱스 테이블들 (코드워드 맵핑 테이블이라고도 한다), 다양한 블록들을 위한 인코딩 컨텍스트들의 정의들, 그리고 컨텍스트들의 각각을 위해 이용할 최고 확률 인트라 예측 모드의 표시들, 인트라 예측 모드 인덱스 테이블, 및 수정된 인트라 예측 모드 인덱스 테이블을 포함할 수도 있다.

[0176] 예측 프로세싱 유닛 (341) 이 인터-예측, 인트라-예측 또는 IntraBC 예측을 통해 현재 비디오 블록에 대한 예측 블록을 생성한 후에, 비디오 인코더 (20) 는 현재 비디오 블록으로부터 예측 블록을, 예를 들어, 합산기 (350) 를 통해, 가산함으로써 잔차 비디오 블록을 형성한다. 잔차 블록에서 잔차 비디오 데이터는 하나 이상의 TU 들에 포함될 수도 있고 변환 프로세싱 유닛 (352) 에 적용될 수도 있다. 변환 프로세싱 유닛 (352) 은 DCT (discrete cosine transform) 또는 개념적으로 유사한 변환과 같은 변환을 이용하여 잔차 변환 계수들로 잔차 비디오 데이터를 변환한다. 변환 프로세싱 유닛 (352) 은 잔차 비디오 데이터를 픽셀 도메인으로부터 변환 도메인, 이를테면 주파수 도메인으로 컨버팅할 수도 있다.

[0177] 변환 프로세싱 유닛 (352) 은 양자화 프로세싱 유닛 (354) 에 결과적인 변환 계수들을 전송할 수도 있다. 양자화 프로세싱 유닛 (354) 은 변환 계수들을 양자화하여 비트 레이트를 더 감소시킨다. 양자화 프로세싱 유닛 (354) 은 계수들의 일부 또는 전부와 연관된 비트 깊이를 감소시킬 수도 있다. 양자화의 정도는 양자화 파라미터를 조정함으로써 변경될 수도 있다. 다음으로, 일부 예들에서, 양자화 프로세싱 유닛 (354) 은 양자화된 변환 계수들을 포함하는 매트릭스의 스캔을 수행할 수도 있다. 대안적으로, 엔트로피 인코딩 유닛 (356) 은 스캔을 수행할 수도 있다.

[0178] 양자화 다음에, 엔트로피 인코딩 유닛 (356) 은 양자화된 변환 계수들을 엔트로피 인코딩한다. 예를 들어, 엔트로피 인코딩 유닛 (356) 은 CAVLC (context adaptive variable length coding), CABAC (context adaptive binary arithmetic coding), SBAC (syntax-based context-adaptive binary arithmetic coding), PIPE (probability interval partitioning entropy) 코딩 또는 또 다른 엔트로피 인코딩 방법론 또는 기법을 수행할 수도 있다. 엔트로피 인코딩 유닛 (356) 은, 예를 들어, IntraBC 모드에 따른 예측을 위한 벡터 성분들, 플래그들 및 다른 신택스 엘리먼트들을 포함한, 신택스 엘리먼트들을 인코딩하는 것 및 이진화를 위해 본원에 기재된 기법들 중 임의의 것을 수행할 수도 있다. 엔트로피 인코딩 유닛 (356) 에 의한 엔트로피 인코딩 다음에, 인코딩된 비디오 비트스트림은, 비디오 디코더 (30) 로 송신되거나 또는 비디오 디코더 (30) 에 의한 나중의 송신 또는 취출을 위해 보관될 수도 있다.

[0179] 역 양자화 프로세싱 유닛 (358) 및 역 변환 프로세싱 유닛 (360) 은 역 양자화 및 역 변환을 각각 적용하여, 다른 비디오 블록들의 예측을 위한 참조 블록으로서 나중에 사용하기 위해 픽셀 도메인에서 잔차 블록을 재구성한다. 모션 보상 유닛 (344) 및/또는 IntraBC 유닛 (348) 은, 참조 화상 리스트들 중의 하나 내의 참조 화상들 중의 하나의 예측 블록에 잔차 블록을 가산함으로써 참조 블록을 계산할 수도 있다. 모션 보상 유닛 (344) 및/또는 IntraBC 유닛 (348) 은 또한 하나 이상의 보간 필터들을 그 재구성된 잔차 블록에 적용하여, 모션 추정에 사용하기 위한 서브 정수 픽셀 값들을 계산할 수도 있다.

[0180] 합산기 (362) 는 재구성된 잔차 블록을 모션 보상 유닛 (344) 및/또는 IntraBC 유닛 (348) 에 의해 생성되는 모션 보상된 예측 블록에 가산하여 재구성된 비디오 블록을 생성한다. 의도된 영역 메모리 (364) 는, 여기에 기재된 바처럼, 비디오 인코더 (20), 예를 들어, IntraBC 유닛 (348) 에 의한 현재 비디오 블록의 IntraBC 를 위한 의도된 영역의 정의에 따라 재구성된 비디오 블록들을 저장한다. 의도된 영역 메모리 (364) 는 필터 프로세싱 유닛 (366) 에 의해 인루프 필터링되지 않은 재구성된 비디오 블록들을 저장할 수도 있다. 합산기 (362) 는 의도된 영역 메모리 (364) 와 병렬로 필터 프로세싱 유닛 (366) 에 재구성된 비디오 블록들을 제공할 수도 있거나, 또는 의도된 영역 메모리 (364) 는 IntraBC 를 위한 의도된 영역에 대한 필요성이 더 이상 없을 때 필터 프로세싱 유닛 (366) 으로 재구성된 비디오 블록들을 해제할 수도 있다. 어떤 경우든, IntraBC 유닛 (348) 은 현재 비디오 블록을 예측하기 위한 현재 비디오 블록과 동일한 화상 내의 예측 비디오 블록을 위한 의도된 영역 메모리 (364) 에서 재구성된 비디오 블록들을 서치할 수도 있다.

[0181] 필터 프로세싱 유닛 (366) 은 재구성된 비디오 블록들에 대한 인루프 필터링을 수행할 수도 있다. 인루프 필터링은, 재구성된 비디오로부터 블록키니스 아티팩트 (blockiness artifact) 를 제거하기 위해 블록 경계들을 필터링하기 위한 디블록 필터링을 포함할 수도 있다. 인루프 필터링은 또한, 재구성된 비디오를 향상시키기 위해 SAO 필터링을 포함할 수도 있다. 일부가 인루프 필터링될 수도 있는 재구성된 블록들은 참조 화상들로서 참조 화상 메모리 (368) 에 저장될 수도 있다. 참조 화상들은 후속 비디오 프레임 또는 화상에서 블록을



인터 예측하기 위한 예측 블록들로서 모션 추정 유닛 (342) 및 모션 보상 유닛 (344) 에 의해 이용될 수도 있는 재구성된 블록들을 포함할 수도 있다.

[0182] 이런 식으로, 비디오 인코더 (20) 는, 예를 들어, IntraBC 모드에 따라, 동일한 화상 내에서 비디오 데이터의 예측 블록들로부터 비디오 데이터의 블록들의 인트라 예측을 위한 본 개시의 예시적인 기법들을 구현하도록 구성될 수도 있다. 예를 들어, 비디오 인코더 (20) 는, 동일한 화상 내의 비디오 데이터의 예측 블록들로부터 비디오 데이터의 블록들의 인트라 예측을 위한 모드를 포함하는 비디오 데이터를 인코딩하는 방법을 수행하도록 구성된 비디오 인코더의 일 예일 수도 있으며, 상기 방법은 비디오 데이터의 현재 블록과 동일한 화상 내의 비디오 데이터의 이전에 인코딩된 블록들의 세트로부터 비디오 데이터의 현재 블록을 위한 비디오 데이터의 예측 블록을 선택하는 단계를 포함한다. 그 방법은, 2차원 벡터를 결정하는 단계로서, 상기 2차원 벡터는 수평 변위 성분 및 수직 변위 성분을 갖고, 상기 수평 변위 성분은 비디오 데이터의 예측 블록과 비디오 데이터의 현재 블록 사이의 수평 변위를 나타내고, 상기 수직 변위 성분은 비디오 데이터의 예측 블록과 비디오 데이터의 현재 블록 사이의 수직 변위를 나타내는, 상기 2차원 벡터를 결정하는 단계를 더 포함한다. 그 방법은, 비디오 데이터의 현재 블록 및 비디오 데이터의 예측 블록에 기초하여 잔차 블록을 결정하는 단계, 및 인코딩된 비디오 비트스트림에서, 2차원 벡터의 수평 변위 성분 및 수직 변위 성분 및 잔차 블록을 정의하는 하나 이상의 선택스 엘리먼트들을 인코딩하는 단계를 더 포함한다.

[0183] 비디오 인코더 (20) 는 또한, 인코딩된 비디오 비트스트림을 저장하도록 구성된 메모리, 및 하나 이상의 프로세서들을 포함하는 비디오 인코더의 일 예일 수도 있다. 하나 이상의 프로세서는, 비디오 데이터의 현재 블록과 동일한 화상 내의 비디오 데이터의 사전에 인코딩된 블록들의 세트로부터 비디오 데이터의 현재 블록을 위한 비디오 데이터의 예측 블록을 선택하도록 구성된다. 그 하나 이상의 프로세서들은 또한, 2차원 벡터를 결정하는 것으로서, 상기 2차원 벡터는 수평 변위 성분 및 수직 변위 성분을 갖고, 상기 수평 변위 성분은 비디오 데이터의 예측 블록과 비디오 데이터의 현재 블록 사이의 수평 변위를 나타내고, 상기 수직 변위 성분은 비디오 데이터의 예측 블록과 비디오 데이터의 현재 블록 사이의 수직 변위를 나타내는, 상기 2차원 벡터를 결정하도록 구성된다. 그 하나 이상의 프로세서는 또한, 비디오 데이터의 현재 블록 및 비디오 데이터의 예측 블록에 기초하여 잔차 블록을 결정하고, 및 인코딩된 비디오 비트스트림에서, 2차원 벡터의 수평 변위 성분 및 수직 변위 성분 및 잔차 블록을 정의하는 하나 이상의 선택스 엘리먼트들을 인코딩하도록 구성된다.

[0184] 도 7은 본 개시에 기재된 기법들을 구현할 수도 있는 일 예의 비디오 디코더를 예시하는 블록도이다. 도 7의 예에서, 비디오 디코더 (30) 는, 비디오 데이터 메모리 (378), 엔트로피 디코딩 유닛 (380), 예측 프로세싱 유닛 (381), 역 양자화 프로세싱 유닛 (386), 역 변환 프로세싱 유닛 (388), 합산기 (390), 의도된 영역 메모리 (392), 필터 프로세싱 유닛 (394), 및 참조 화상 메모리 (396) 를 포함한다. 예측 프로세싱 유닛 (381) 은, 모션 보상 유닛 (382), 인트라 예측 프로세싱 유닛 (384), 및 인트라 블록 카피 (IntraBC) 유닛 (385) 을 포함한다. 비디오 디코더 (30) 는, 일부 예들에서, 도 6로부터의 비디오 인코더 (20) 에 대해 설명된 인코딩 패스에 일반적으로 상반되는 디코딩 패스를 수행할 수도 있다.

[0185] 다양한 예들에서, 비디오 디코더 (30) 의 유닛은 본 개시의 기법들을 수행하도록 태스킹될 수도 있다. 또한, 일부 예들에서, 본 개시의 기법들은 비디오 디코더 (30) 의 유닛들 중의 하나 이상 중에서 분할될 수도 있다. 예를 들어, IntraBC 유닛 (385) 은 본 개시의 기법들을 단독으로, 또는 비디오 디코더 (30) 의 다른 유닛들과 조합하여, 이를테면, 모션 보상 유닛 (382), 인트라 예측 프로세싱 유닛 (384), 의도된 구역 메모리 (392), 및 엔트로피 디코딩 유닛 (380) 과 조합하여, 수행할 수도 있다.

[0186] 디코딩 프로세스 동안, 비디오 디코더 (30) 는, 비디오 인코더로부터 연관된 선택스 엘리먼트들 및 인코딩된 비디오 슬라이스의 비디오 블록들을 나타내는 인코딩된 비디오 비트스트림을 수신한다. 비디오 데이터 메모리 (378) 는 비디오 디코더 (30) 의 컴포넌트들에 의해 디코딩될 비디오 데이터, 이를테면 인코딩된 비디오 비트스트림을 저장할 수도 있다. 비디오 데이터 메모리 (378) 에 저장된 비디오 데이터는, 예를 들어, 컴퓨터 판독가능 매체 (16) 로부터, 예를 들어, 카메라와 같은 로컬 비디오 소스로부터, 비디오 데이터의 유선 또는 무선 네트워크 통신을 통해, 또는 물리적 데이터 저장 매체에 액세스하는 것에 의해, 획득될 수도 있다. 비디오 데이터 메모리 (378) 는 인코딩된 비디오 비트스트림으로부터 인코딩된 비디오 데이터를 저장하는 코딩된 화상 버퍼 (CPB) 를 형성할 수도 있다. 의도된 영역 메모리 (392) 는, 예를 들어, 인트라-블록 코딩 모드들에서 비디오 디코더 (30) 에 의한 비디오 데이터를 디코딩하는데 이용하기 위한 참조 비디오 데이터를 저장하는 참조 블록 메모리일 수도 있다. 비디오 데이터 메모리 (378) 및 의도된 영역 메모리 (392) 는 SDRAM 를 포함하는 DRAM, MRAM, RRAM, 또는 다른 타입의 메모리 디바이스들 등의 다양한 메모리 디바이스들 중의 어느 것에 의해 형성될 수도 있다. 비디오 데이터 메모리 (378) 및 참조 화상 메모리 (392) 는 동일한 메모리 디바이스 또

는 별개의 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에서, 비디오 데이터 메모리 (378) 는 비디오 디코더 (30) 의 다른 컴포넌트들과 함께 온칩 (on-chip) 일 수도 있거나, 또는 그러한 컴포넌트들에 대해 오프칩 (off-chip) 일 수도 있다.

[0187] 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (380) 은, 양자화된 계수들, 인터 예측을 위한 모션 벡터들, IntraBC 예측을 위한 2차원 벡터들, 및 본원에 기재된 다른 신택스 엘리먼트들을 생성하기 위하여 비트스트림을 엔트로피 디코딩한다. 엔트로피 디코딩 유닛 (380) 은, 예를 들어, IntraBC 모드에 따른 예측을 위한 벡터 성분들, 플래그들 및 다른 신택스 엘리먼트들을 포함한, 신택스 엘리먼트들을 인코딩하는 것 및 이진화를 위해 본원에 기재된 기법들 중 임의의 것의 역을 수행할 수도 있다. 엔트로피 디코딩 유닛 (380) 은 벡터들 및 다른 신택스 엘리먼트들을 예측 프로세싱 유닛 (381) 으로 포워딩한다. 비디오 디코더 (30) 는 시퀀스 레벨, 화상 레벨, 비디오 슬라이스 레벨 및/또는 비디오 블록 레벨에서 신택스 엘리먼트들을 수신할 수도 있다.

[0188] 비디오 슬라이스가 인트라 코딩된 (I) 슬라이스로서 코딩될 때, 예측 프로세싱 유닛 (381) 의 인트라 예측 프로세싱 유닛 (384) 은, 현재 프레임 또는 화상의 이전에 디코딩된 블록들로부터 시그널링된 인트라 예측 모드 및 데이터에 기초하여 현재 비디오 슬라이스의 비디오 블록을 위한 예측 데이터를 생성할 수도 있다. 비디오 프레임이 인터 코딩된 (즉, B 또는 P) 슬라이스로서 코딩될 때, 예측 프로세싱 유닛 (381) 의 모션 보상 유닛 (382) 은 엔트로피 디코딩 유닛 (280) 으로부터 수신된 모션 벡터들 및 다른 신택스 엘리먼트들에 기초하여 현재 비디오 슬라이스의 비디오 블록을 위한 예측 블록들을 생성한다. 예측 블록들은 참조 화상 리스트들 중의 하나의 리스트 내의 참조 화상들 중의 하나의 화상로부터 생성될 수도 있다. 비디오 디코더 (30) 는 참조 화상 메모리 (396) 에 저장된 참조 화상들에 기초하여 디폴트 (default) 구성 기법들 또는 임의의 다른 기법을 이용하여, 참조 프레임 리스트들, RefPicList0 및 RefPicList1 을 구성할 수도 있다. 비디오 블록이 본원에 기재된 IntraBC 모드에 따라 코딩될 때, 예측 프로세싱 유닛 (381) 의 IntraBC 유닛 (385) 은 엔트로피 디코딩 유닛 (380) 으로부터 수신된 2차원 벡터들 및 다른 신택스 엘리먼트들에 기초하여 현재 비디오 블록을 위한 예측 블록들을 생성한다. 예측 블록들은 비디오 인코더 (20) 에 의해 정의되는 현재 비디오 블록과 동일한 화상 내의 의도된 영역내에 있을 수도 있고 의도된 영역 메모리 (392) 로부터 추출될 수도 있다.

[0189] 모션 보상 유닛 (382) 및 IntraBC 유닛 (385) 은, 벡터들 및 다른 신택스 엘리먼트들을 파싱 (parsing) 하여 현재 비디오 슬라이스의 비디오 블록을 위한 예측 정보를 결정하고, 그 예측 정보를 이용하여 디코딩되고 있는 현재 비디오 블록을 위한 예측 블록들을 생성한다. 예를 들어, 모션 보상 유닛 (282) 은 수신된 신택스 엘리먼트들의 일부를 이용하여 비디오 슬라이스의 비디오 블록들을 코딩하는데 이용된 예측 모드 (예를 들어, 인트라 또는 인터 예측), 인터 예측 슬라이스 타입 (예를 들어, B 슬라이스, 또는 P 슬라이스), 슬라이스를 위한 참조 화상 리스트들의 하나 이상을 위한 구성 정보, 슬라이스의 각 인터 인코딩된 비디오 블록을 위한 모션 벡터들, 슬라이스의 각 인터 코딩된 비디오 블록을 위한 인터 예측 상태, 및 현재 비디오 슬라이스에서 비디오 블록들을 디코딩하기 위한 다른 정보를 결정한다. 유사하게, IntraBC 유닛 (385) 은, 현재 비디오 블록이 IntraBC 모드, 화상의 어느 비디오 블록들이 의도된 영역 내에 있고 의도된 영역 메모리 내에 저장되어야 하는지를 나타내는 의도된 영역 메모리 (392) 를 위한 구성 정보, 슬라이스의 각각의 IntraBC 예측된 비디오 블록에 대한 2차원 벡터들, 슬라이스의 각각의 IntraBC 예측된 비디오 블록에 대한 IntraBC 예측 상태, 및 현재 비디오 슬라이스에서 비디오 블록들을 디코딩하기 위한 다른 정보를 이용하여 예측되었다는 것을 결정하기 위하여, 수신된 신택스 엘리먼트들의 일부, 예를 들어, 플래그를 이용할 수도 있다.

[0190] 모션 보상 유닛 (382) 및 IntraBC 유닛 (385) 은 또한, 보간 필터들에 기초하여 보간을 수행할 수도 있다. 모션 보상 유닛 (382) 및 IntraBC 유닛 (385) 은 비디오 블록들의 인코딩 동안 비디오 인코더 (20) 에 의해 이용되는 보간 필터들을 이용하여 예측 블록들의 서브 정수 픽셀들을 위한 보간된 값들을 계산할 수도 있다. 이 경우에, 모션 보상 유닛 (382) 및 IntraBC 유닛 (385) 은 수신된 신택스 엘리먼트들로부터 비디오 인코더 (20) 에 의해 이용된 보간 필터들을 결정하고, 그 보간 필터들을 이용하여 예측 블록들을 생성할 수도 있다.

[0191] 역 양자화 프로세싱 유닛 (386) 은 비트스트림에서 제공되고 엔트로피 디코딩 유닛 (380) 에 의해 디코딩된 양자화된 변환 계수들을 역 양자화, 즉 탈양자화한다. 역 양자화 프로세스는, 양자화의 정도, 그리고, 마찬가지로, 적용되어야 하는 역 양자화의 정도를 결정하기 위해, 비디오 슬라이스에서 각 비디오 블록을 위해 비디오 인코더 (20) 에 의해 산출된 양자화 파라미터의 이용을 포함할 수도 있다. 역 변환 프로세싱 유닛 (388) 은, 픽셀 도메인에서 잔차 블록들을 생성하기 위해 변환 계수들에, 역 변환, 예를 들어, 역 DCT, 역 정수 변환, 또는 개념적으로 유사한 역 변환 프로세스를 적용한다.

[0192] 모션 보상 유닛 (382) 및 IntraBC 유닛 (385) 이 벡터들 및 다른 신택스 엘리먼트들에 기초하여 현재 비디오 블

록을 위한 예측 블록을 생성한 후에, 비디오 디코더 (30)는 역 변환 프로세싱 유닛 (388) 으로부터의 잔차 블록들과 모션 보상 유닛 (382) 및 IntraBC 유닛 (385) 에 의해 생성된 대응하는 예측 블록들을 합산함으로써 디코딩된 비디오 블록을 형성한다. 합산기 (390) 는 재구성된 비디오 블록들을 생성하기 위해 이 합산 연산을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다.

[0193] 의도된 영역 메모리 (392) 는, 여기에 기재된 바처럼, 비디오 인코더 (20) 에 의한 현재 비디오 블록의 IntraBC 를 위한 의도된 영역의 정의에 따라 재구성된 비디오 블록들을 저장한다. 의도된 영역 메모리 (392) 는 필터 프로세싱 유닛 (394) 에 의해 인루프 필터링되지 않은 재구성된 비디오 블록들을 저장할 수도 있다. 합산기 (390) 는 의도된 영역 메모리 (392) 와 병렬로 필터 프로세싱 유닛 (394) 에 재구성된 비디오 블록들을 제공할 수도 있거나, 또는 의도된 영역 메모리 (392) 는 IntraBC 를 위한 의도된 영역에 대한 필요성이 더 이상 없을 때 필터 프로세싱 유닛 (394) 으로 재구성된 비디오 블록들을 해제할 수도 있다. 어떤 경우든, IntraBC 유닛 (385) 은 의도된 영역 메모리 (392) 로부터 현재 비디오 블록을 위한 예측 비디오 블록을 취출한다.

[0194] 필터 프로세싱 유닛 (394) 은 재구성된 비디오 블록들에 대한 인루프 필터링을 수행할 수도 있다. 인루프 필터링은, 재구성된 비디오로부터 블록키스 아티팩트 (blockiness artifact) 를 제거하기 위해 블록 경계들을 필터링하기 위한 더블 필터링을 포함할 수도 있다. 인루프 필터링은 또한, 재구성된 비디오를 향상시키기 위해 SAO 필터링을 포함할 수도 있다. 일부가 인루프 필터링될 수도 있는 재구성된 블록들은 참조 화상들로서 참조 화상 메모리 (368) 에 저장될 수도 있다. 참조 화상들은 후속 비디오 프레임 또는 화상에서 블록을 인터 예측하기 위한 예측 블록들로서 모션 보상 유닛 (382) 에 의해 이용될 수도 있는 재구성된 블록들을 포함할 수도 있다. 참조 화상 메모리 (396) 는 또한, 도 1의 디스플레이 디바이스 (32) 와 같은 디스플레이 디바이스 상에 나중에 표출하기 위해 디코딩된 비디오를 저장한다.

[0195] 이런 식으로, 비디오 디코더 (30) 는, 동일한 화상 내에서 비디오 데이터의 예측 블록들에 기초하여 비디오 데이터의 현재 블록들을 인트라 예측하기 위한 본 개시의 예시적인 기법들을 구현하도록 구성될 수도 있다. 예를 들어, 비디오 디코더 (30) 는 동일한 화상 내에서 비디오 데이터의 예측 블록들로부터 비디오 데이터의 블록들의 인트라 예측을 위한 모드를 포함하는 비디오 데이터를 디코딩하는 방법을 수행하도록 구성된 비디오 디코더의 일 예일 수도 있고, 그 방법은 인코딩된 비디오 비트스트림에서, 비디오 데이터의 현재 블록을 위한 2차원 벡터의 수평 변위 성분 및 수직 변위 성분 및 잔차 블록을 정의하는 하나 이상의 선택스 엘리먼트들을 수신하는 단계, 및 하나 이상의 선택스 엘리먼트들을 디코딩하는 단계를 포함한다. 그 방법은 2차원 벡터의 수평 변위 성분 및 수직 변위 성분을 정의하는 디코딩된 선택스 엘리먼트들에 기초하여 비디오 데이터의 현재 블록을 위한 비디오 데이터의 예측 블록을 결정하는 단계로서, 상기 비디오 데이터의 예측 블록은 비디오 데이터의 현재 블록과 동일한 화상 내에 있는 비디오 데이터의 재구성 블록인, 상기 예측 블록을 결정하는 단계, 및 비디오 데이터의 예측 블록 및 잔차 블록에 기초하여 비디오 데이터의 현재 블록을 재구성하는 단계를 포함한다.

[0196] 비디오 디코더 (30) 는 또한, 비디오 데이터를 인코딩하는 인코딩된 비디오 비트스트림을 저장하도록 구성된 메모리, 및 하나 이상의 프로세서들을 포함하는 비디오 디코더의 일 예일 수도 있다. 비디오 디코더의 하나 이상의 프로세서들은, 하나 이상의 선택스 엘리먼트들을 디코딩하고, 2차원 벡터의 수평 변위 성분 및 수직 변위 성분을 정의하는 디코딩된 선택스 엘리먼트들에 기초하여 비디오 데이터의 현재 블록을 위한 비디오 데이터의 예측 블록을 결정하는 것으로서, 상기 비디오 데이터의 예측 블록은 비디오 데이터의 현재 블록과 동일한 화상 내에 있는 비디오 데이터의 재구성 블록인, 상기 예측 블록을 결정하고, 그리고 비디오 데이터의 예측 블록 및 잔차 블록에 기초하여 비디오 데이터의 현재 블록을 재구성하도록 구성된다.

[0197] 도 8 는 본 개시에 기재된 기법에 따라 비디오 데이터를 인코딩하는 예시적인 방법을 예시하는 플로우차트이다. 도 8의 기법들은 비디오 인코더 (20) 와 같은 비디오 인코더에 관하여 설명될 것이다. 비디오 인코더 (20) 는 화상의 CU 의 제 1 크로마 블록을 위한 벡터를 코딩 유닛과 연관된 복수의 벡터들로부터 결정한다 (402). 비디오 인코더 (20) 는 벡터를 이용하여 제 1 크로마 참조 블록을 로케이팅한다 (404). 제 1 크로마 참조 블록은 CU 와 동일한 화상에 있을 수도 있다. 비디오 인코더 (20) 는 제 1 크로마 참조 블록에 기초하여 제 1 크로마 블록을 예측한다 (406). 비디오 인코더 (20) 는 벡터를 이용하여 제 2 크로마 참조 블록을 로케이팅한다 (408). 제 2 크로마 참조 블록은 CU 와 동일한 화상에 있을 수도 있다. 비디오 인코더 (20) 는 제 2 크로마 참조 블록에 기초하여 코딩 유닛의 제 2 크로마 블록을 예측한다 (410).

[0198] 비디오 인코더 (20) 는, 인코딩된 비디오 데이터에의 포함을 위해, 화상의 CU 가 인트라 블록 카피 모드를 이용

하여 코딩된다는 표시 및 CU 의 각 루마 블록을 위한 벡터의 표시를 생성한다 (412). 위에 설명된 바처럼, 비디오 인코더 (20) 는, 별개의 블록 벡터를 이용하여 루마 성분들의 각각을 예측하지만, 루마 성분들을 예측하는데 이용된 별개의 블록 벡터들로부터 결정된 공통 블록 벡터를 이용하여 CU 의 크로마 성분들을 예측할 수도 있다. 이에 따라, 비디오 인코더 (20) 는 CU 의 각 루마 성분을 위한 별개의 블록 벡터 표시를 인코딩된 비디오 데이터에 포함할 수도 있다.

- [0199] 도 9 는 본 개시에 기재된 기법에 따라 비디오 데이터를 디코딩하는 예시적인 방법을 예시하는 플로우차트이다. 도 9의 기법들은 비디오 디코더 (30) 와 같은 비디오 디코더에 관하여 설명될 것이다. 비디오 디코더 (30) 는 비디오 데이터의 화상의 CU 가 인트라 블록 카피 모드를 이용하여 코딩되는 것을 결정한다 (420). 비디오 디코더 (30) 는 CU 의 제 1 크로마 블록을 위한 벡터를 코딩 유닛과 연관된 복수의 벡터들로부터 결정한다 (422). 비디오 디코더 (30) 는 벡터를 이용하여 제 1 크로마 참조 블록을 로케이팅한다 (424). 제 1 크로마 참조 블록은 CU 와 동일한 화상에 있다. 비디오 디코더 (30) 는 제 1 크로마 참조 블록에 기초하여 제 1 크로마 블록을 예측한다 (426). 비디오 디코더 (30) 는 벡터를 이용하여 제 2 크로마 참조 블록을 로케이팅한다 (428). 제 2 크로마 참조 블록은 CU 와 동일한 화상에 있다. 비디오 디코더 (30) 는 제 2 크로마 참조 블록에 기초하여 CU 의 제 2 크로마 블록을 예측한다 (430).
- [0200] CU 는, 예를 들어, 2개 이상의 PU들을 포함할 수도 있으며, 제 1 크로마 블록은 제 1 PU 에 대응하고 제 2 크로마 블록은 제 2 PU 에 대응한다. 비디오 디코더 (30) 는, 제 1 PU 를 위한 벡터를 획득하는 것에 의해 CU 의 제 2 크로마 블록을 위한 벡터를 결정할 수도 있다.
- [0201] 일 예에서, CU 는 2개 이상의 PU 들을 포함할 수도 있고, 각 PU 는 연관된 벡터를 갖는다. 비디오 디코더 (30) 는 제 1 예측 유닛과 연관된 벡터를 이용하여 제 1 루마 참조 블록을 로케이팅할 수도 있다. 제 1 루마 참조 블록은 제 1 PU 에 대응할 수도 있다. 비디오 디코더 (30) 는 제 2 PU 와 연관된 벡터를 이용하여 제 2 루마 참조 블록을 로케이팅할 수도 있다. 제 2 루마 참조 블록은 제 2 PU 에 대응할 수도 있다. 비디오 디코더 (30) 는 제 1 PU 와 연관된 벡터를 이용하여 제 1 크로마 참조 블록을 로케이팅 및 제 1 PU 와 연관된 벡터를 이용하여 제 2 크로마 참조 블록을 로케이팅할 수도 있다.
- [0202] 일 예에서, CU 를 위한 크로마 샘플링 포맷이 4:2:0 일 때, 제 1 크로마 블록은 CU 의 하단 우측 블록일 수도 있다. 제 2 크로마 블록은 CU 의 상단 우측 블록일 수도 있고, 비디오 디코더 (30) 는 벡터를 이용하여 제 3 참조 블록을 로케이팅할 수도 있다. 제 3 참조 블록은 화상에 있을 수도 있다. 비디오 디코더 (30) 는 제 1 참조 블록에 기초하여 제 3 크로마 블록을 예측할 수도 있다. 제 3 크로마 블록은 CU 의 상단 좌측 블록일 수도 있다. 비디오 디코더 (30) 는 벡터를 이용하여 제 4 참조 블록을 로케이팅할 수도 있고 제 2 참조 블록에 기초하여 CU 의 제 4 크로마 블록을 예측할 수도 있다. 제 4 참조 블록은 화상에 있을 수도 있고, 제 4 크로마 블록은 CU 의 하단 좌측 블록일 수도 있다.
- [0203] 일 예에서, CU 를 위한 크로마 샘플링 포맷은 4:2:2 일 수도 있다. 제 1 참조 블록은 상단 우측 블록일 수도 있고, 제 2 크로마 블록은 CU 의 상단 좌측 블록일 수도 있다. 비디오 디코더 (30) 는 CU 의 제 3 크로마 블록을 위한 제 2 벡터를 결정할 수도 있고 제 2 벡터를 이용하여 제 3 참조 블록을 로케이팅할 수도 있다. 제 3 참조 블록은 화상에 있을 수도 있다. 비디오 디코더 (30) 는 제 3 크로마 블록을 위한 예측자로서 제 3 참조 블록을 이용할 수도 있다. 비디오 디코더 (30) 는 벡터를 이용하여 제 4 참조 블록을 로케이팅할 수도 있다. 제 4 참조 블록은 화상에 있을 수도 있다. 비디오 디코더 (30) 는 CU 의 제 4 크로마 블록을 위한 예측자로서 제 4 참조 블록을 이용할 수도 있다.
- [0204] 예에 따라, 여기에 기재된 기법들 중 어느 것의 특정 행위들 또는 이벤트들이 상이한 시퀀스에서 수행될 수 있거나, 추가될 수 있거나, 병합될 수 있거나, 또는 전부 생략될 수 있다 (예를 들어, 모든 설명된 행위들 또는 이벤트들이 그 기법들의 실시를 위해 필요한 것은 아니다) 는 것이 인식되어야 한다. 또한, 특정 예들에서, 행위들 또는 이벤트들은, 예를 들어, 순차적이기 보다는 멀티스레드 프로세싱, 인터럽트 프로세싱 또는 멀티플 프로세서들을 통해, 동시적으로 수행될 수도 있다.
- [0205] 하나 이상의 예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현되면, 그 기능들은 컴퓨터 판독가능 매체 상의 하나 이상의 명령 또는 코드로서 저장되거나 송신될 수도 있고 하드웨어 기반 프로세싱 유닛에 의해 실행될 수도 있다. 컴퓨터 판독가능 매체는, 예를 들면, 통신 프로토콜에 따라, 일 장소로부터 다른 장소로의 컴퓨터 프로그램의 전송을 가능하게 하는 임의의 매체를 포함하는 통신 매체 또는 데이터 저장 매체와 같은 유형의 매체에 대응하는 컴퓨터 판독가능 저장 매체를 포함할 수도 있다. 이런 방식으로, 컴퓨터 판독가능 매체는 일반적으로, (1) 비일시적인 유



형의 컴퓨터 판독가능 저장 매체 또는 (2) 신호 또는 캐리어 파와 같은 통신 매체에 대응할 수도 있다. 데이터 저장 매체는, 본 개시에서 설명된 기법들의 구현을 위해 명령들, 코드 및/또는 데이터 구조들을 추출하기 위하여 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.

[0206] 비한정적 예로서, 그러한 컴퓨터 판독가능 저장 매체는 RAM, ROM, EEPROM, CD-ROM 또는 다른 광학 디스크 저장, 자기 디스크 저장 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 명령 또는 데이터 구조의 형태로 원하는 프로그램 코드를 저장하는데 이용될 수 있고 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 접속이 컴퓨터 판독가능 매체로 적절히 칭해진다. 예를 들어, 명령들이 동축 케이블, 광섬유 케이블, 연선 (twisted pair), 디지털 가입자 라인 (DSL), 또는 적외선, 전파 (radio), 및 마이크로파와 같은 무선 기술을 이용하여 웹사이트, 서버, 또는 다른 원격 소스로부터 송신되면, 그 동축 케이블, 광섬유 케이블, 연선, DSL, 또는 적외선, 전파, 및 마이크로파와 같은 무선 기술은 매체의 정의 내에 포함된다. 하지만, 컴퓨터 판독가능 저장 매체 및 데이터 저장 매체는 접속, 캐리어 파, 신호 또는 다른 일시적 매체를 포함하는 것이 아니라, 대신에 비일시적, 유형의 저장 매체에 관련된다는 것이 이해되어야 한다. 여기에 이용된, 디스크 (disk) 및 디스크 (disc)는 콤팩트 디스크 (CD), 레이저 디스크, 광 디스크, DVD (digital versatile disc), 플로피 디스크 및 블루레이 디스크를 포함하며, 여기서, 디스크 (disk)는 보통 데이터를 자기적으로 재생하지만, 디스크 (disc)는 레이저를 이용하여 광학적으로 데이터를 재생한다. 또한, 상기의 조합은 컴퓨터 판독가능 매체의 범위 내에 포함되어야 한다.

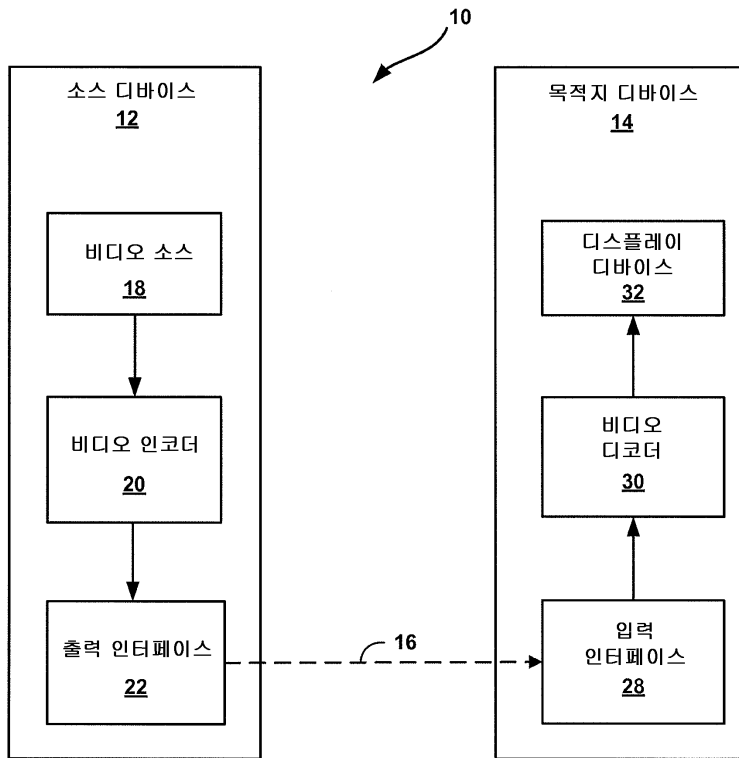
[0207] 명령들은 하나 이상의 프로세서, 이를테면 하나 이상의 DSP (digital signal processor), 범용 마이크로프로세서, ASIC (application specific integrated circuit), FPGA (field programmable logic array), 또는 다른 동등한 집적 또는 이산 로직 회로에 의해 실행될 수도 있다. 따라서, 본원에 이용된 용어 "프로세서"는 전술한 구조 중 임의의 것 또는 본원에 설명된 기법들의 구현에 적합한 임의의 다른 구조를 지칭할 수도 있다. 추가로, 일부 양태들에서, 여기서 설명된 기능은 인코딩 및 디코딩을 위해 구성된 전용 하드웨어 및/또는 소프트웨어 모듈들 내에 제공되거나 또는 결합된 코덱에 포함될 수도 있다. 또한, 그 기법들은 하나 이상의 회로 또는 로직 엘리먼트들에서 완전히 구현될 수 있다.

[0208] 본 개시의 기법들은 무선 핸드셋, 집적 회로 (IC) 또는 IC 들의 세트 (예를 들면, 칩 세트)를 포함하여, 광범위하게 다양한 디바이스들 또는 장치들에서 구현될 수도 있다. 다양한 컴포넌트들, 모듈들 또는 유닛들이, 개시된 기술들을 수행하도록 구성된 디바이스들의 기능적인 양태들을 강조하기 위하여 본 개시에 설명되었지만, 상이한 하드웨어 유닛들에 의한 실현을 반드시 필요로 하는 것은 아니다. 오히려, 상술된 바처럼, 다양한 유닛들이 코덱 하드웨어 유닛에 결합될 수도 있거나, 또는 적합한 소프트웨어 및/또는 펌웨어와 함께, 상술된 하나 이상의 프로세서들을 포함하는 연동적인 (interoperative) 하드웨어 유닛들의 집합에 의해 제공될 수도 있다.

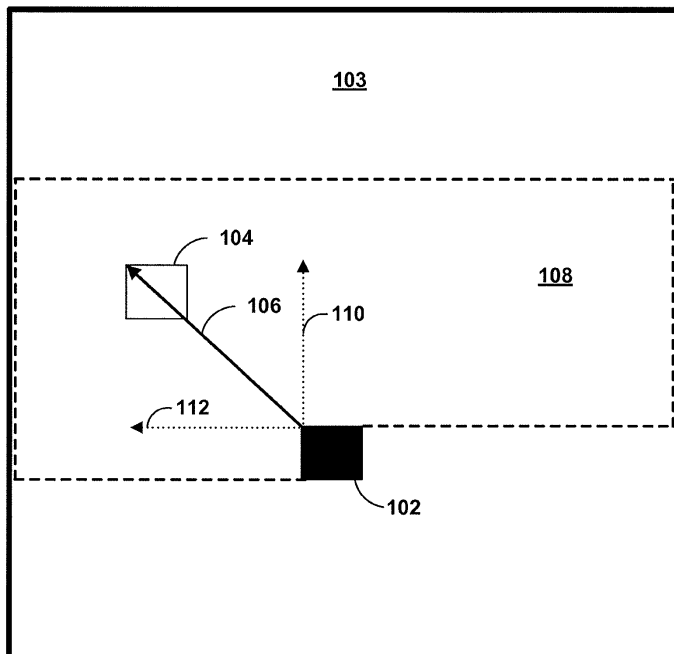
[0209] 다양한 예들이 설명되었다. 이들 및 다른 예들은 다음의 청구항들의 범위 내에 있다.

도면

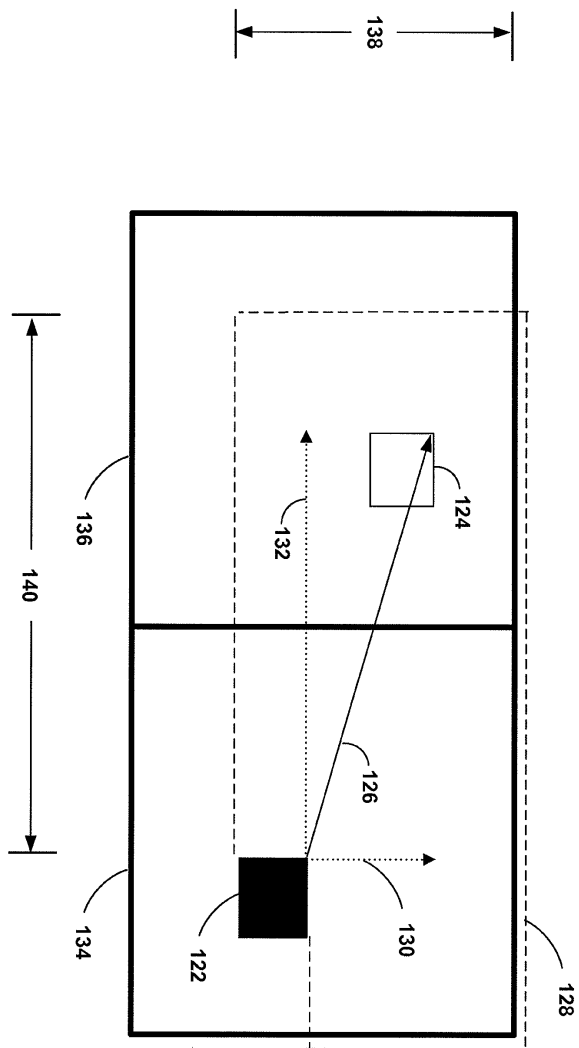
도면1



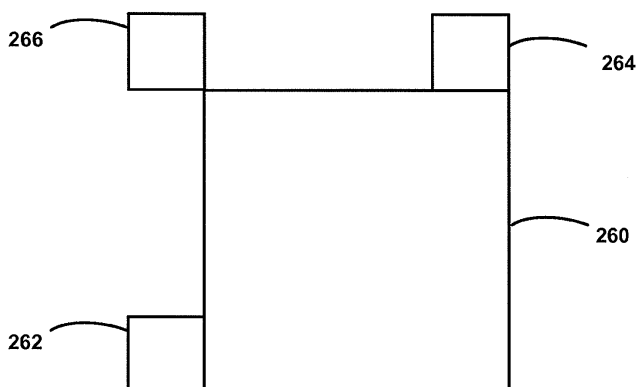
도면2



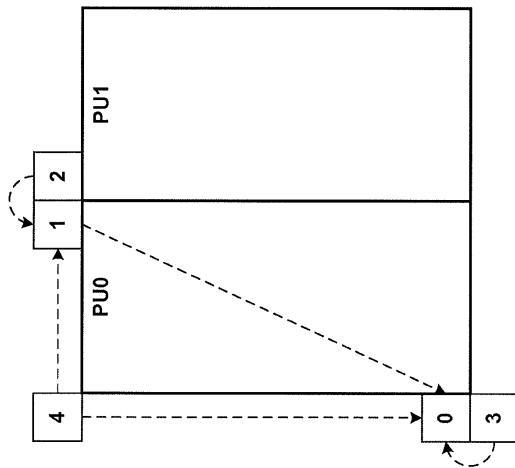
도면3



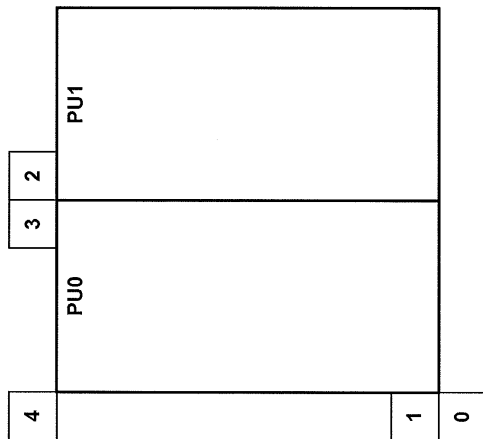
도면4



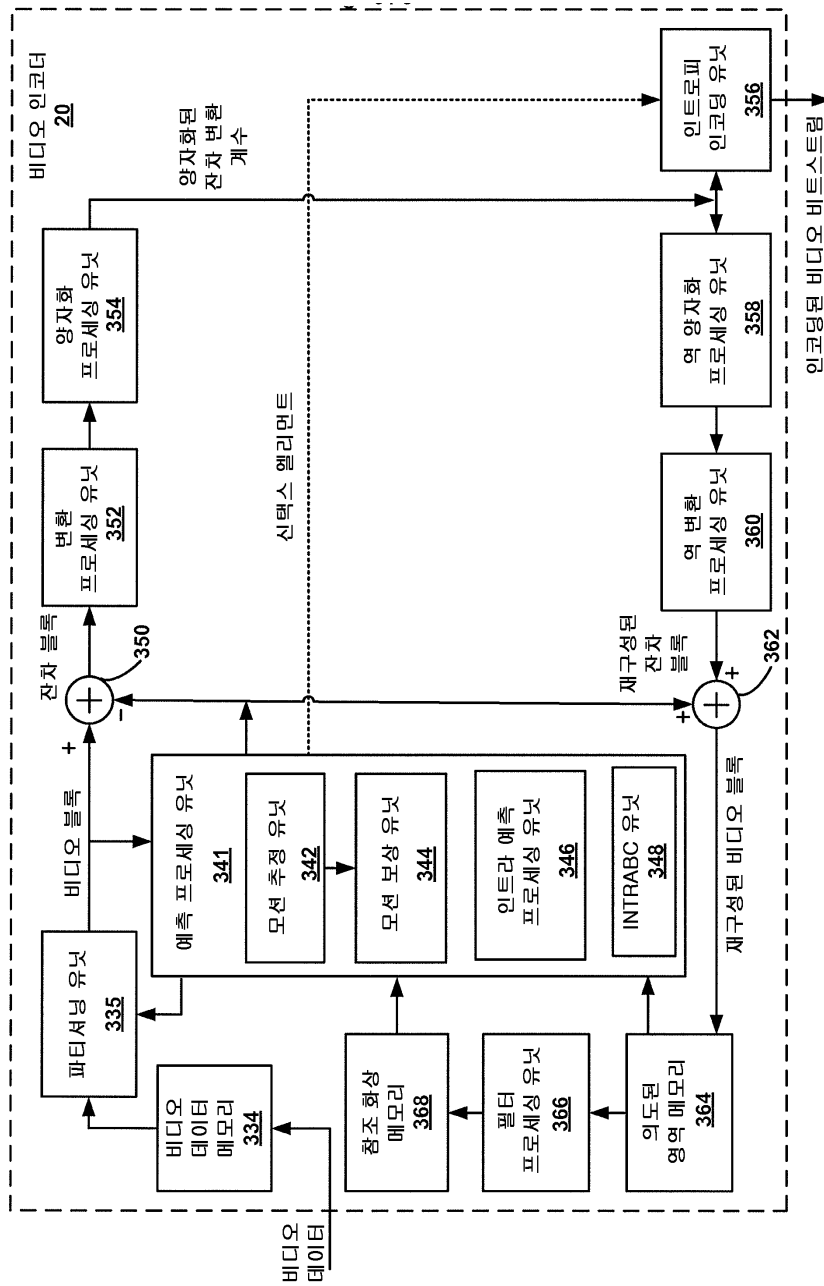
도면5a



도면5b

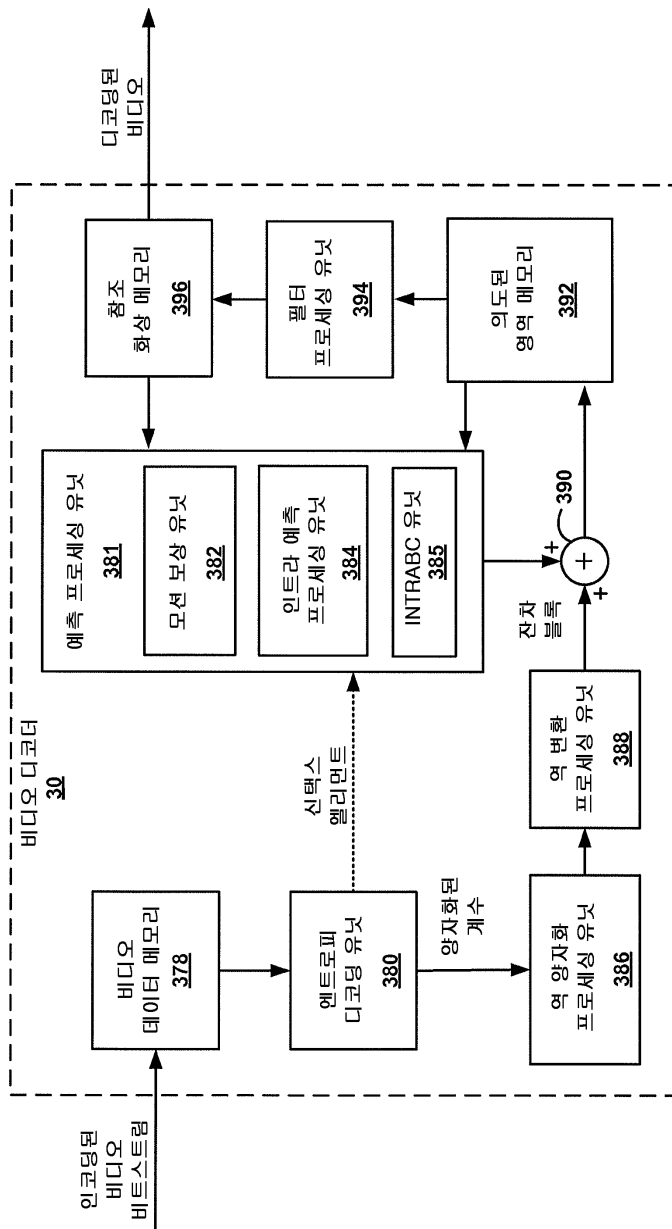


도면6

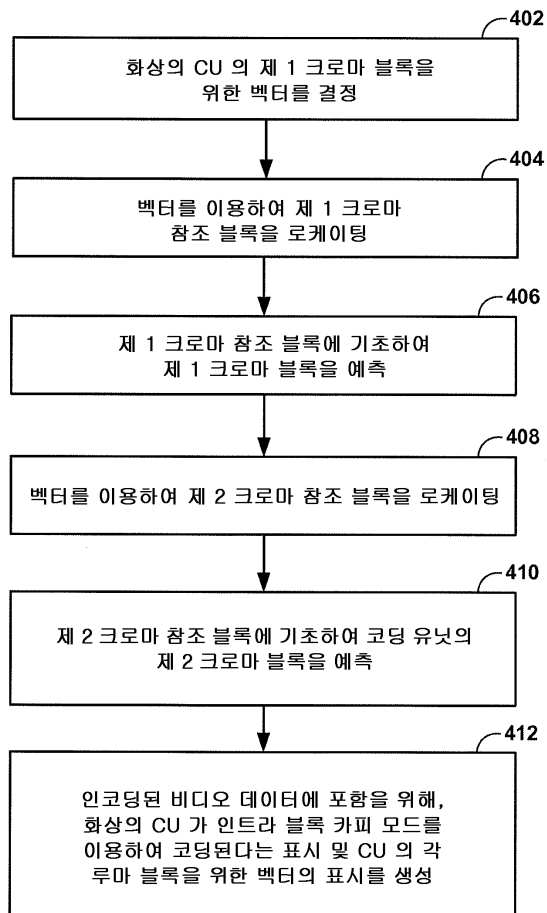




도면7



도면8



도면9

