



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2017-0062464
(43) 공개일자 2017년06월07일

- (51) 국제특허분류(Int. Cl.)
H04N 19/52 (2014.01) **H04N 19/119** (2014.01)
H04N 19/146 (2014.01) **H04N 19/176** (2014.01)
H04N 19/182 (2014.01) **H04N 19/593** (2014.01)
H04N 19/61 (2014.01) **H04N 19/82** (2014.01)
- (52) CPC특허분류
H04N 19/52 (2015.01)
H04N 19/119 (2015.01)
- (21) 출원번호 10-2017-7008749
- (22) 출원일자(국제) 2015년08월19일
심사청구일자 없음
- (85) 번역문제출일자 2017년03월30일
- (86) 국제출원번호 PCT/US2015/045876
- (87) 국제공개번호 WO 2016/053494
국제공개일자 2016년04월07일
- (30) 우선권주장
14/503,669 2014년10월01일 미국(US)

- (71) 출원인
퀄컴 인코포레이티드
 미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
- (72) 발명자
미쉬라 아쉬쉬
 미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
모한타 아린답
 인도 110022 뉴델리 푸람 섹터-8 알.케이. 엔 150
- (74) 대리인
특허법인코리아나

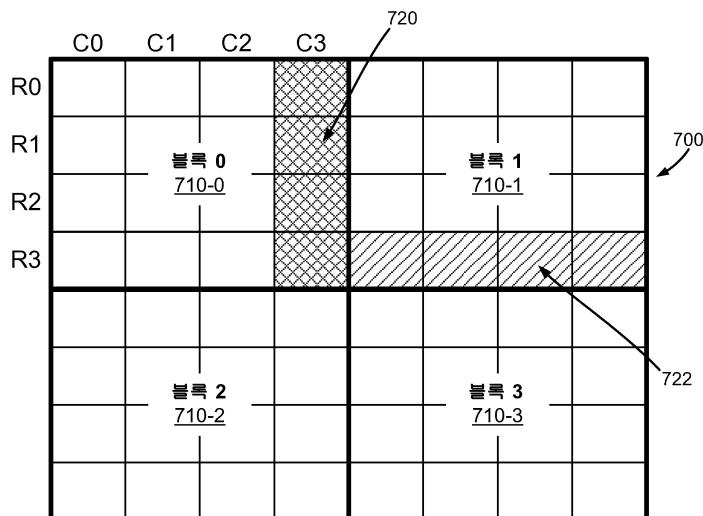
전체 청구항 수 : 총 30 항

(54) 발명의 명칭 **비디오 코딩을 위한 파이프라인식 인트라 예측 하드웨어 아키텍처**

(57) 요 약

공유된 비디오 콘텐츠의 품질 및 양이 증가함에 따라, 비디오 인코딩 표준들 및 기법들이 원거리통신 및 다른 네트워크들을 통한 대역폭 소비를 줄이기 위해 개발되고 있다. 대역폭 소비를 줄이기 위한 하나의 기법이 인트라 예측인데, 그것은 비디오 프레임들 내의 공간적 리던던시들을 이용한다. 각각의 비디오 프레임은 블록들로 세그먼트화될 수도 있고, 인트라 예측이 그 블록들에 적용될 수도 있다. 그러나, 일부 블록들의 인트라 예측은 다른 블록들의 완료 (예컨대, 복원)에 의존할 수도 있는데, 이는 병렬 프로세싱을 도전적이게 할 수 있다. 제공된 것은 다수의 블록들의 인트라 예측에 연관된 효율 및 스루풋을 개선하기 위한 예시적인 기법들이다.

대 표 도 - 도7



(52) CPC특허분류

HO4N 19/146 (2015.01)

HO4N 19/176 (2015.01)

HO4N 19/182 (2015.01)

HO4N 19/593 (2015.01)

HO4N 19/61 (2015.01)

HO4N 19/82 (2015.01)

명세서

청구범위

청구항 1

인트라 예측을 수행하는 방법으로서,

이웃 프로세싱 부에서, 제 1 복수의 이웃 화소들을 수신하는 단계;

상기 이웃 프로세싱 부에 의해, 인트라 예측 프로세싱 부에 의한 인트라 예측을 위한 상기 제 1 복수의 이웃 화소들을 준비하는 단계;

상기 인트라 예측 프로세싱 부에 의해, 준비된 상기 제 1 복수의 이웃 화소들을 수신하는 단계;

상기 인트라 예측 프로세싱 부에 의해, 상기 준비된 제 1 복수의 이웃 화소들을 사용하여 제 1 시구간 동안 제 1 복수의 인트라 예측된 화소들을 갖는 제 1 블록을 결정하는 단계;

상기 이웃 프로세싱 부에 의해, 제 2 블록의 인트라 예측을 위한 제 2 복수의 이웃 화소들을 수신하는 단계;

상기 이웃 프로세싱 부에 의해, 상기 제 1 시구간의 적어도 일부 동안 인트라 예측을 위한 상기 제 2 복수의 이웃 화소들을 준비하는 단계; 및

상기 인트라 예측 프로세싱 부에 의해, 준비된 상기 제 2 복수의 이웃 화소들을 사용하여 상기 제 1 시구간을 바로 뒤따르는 제 2 시구간 동안 제 2 복수의 인트라 예측된 화소들을 갖는 제 2 블록을 결정하는 단계를 포함하며,

상기 제 1 블록에서의 상기 제 1 복수의 인트라 예측된 화소들 중 적어도 일부의 인트라 예측된 화소들은 상기 제 2 블록을 결정하기 위해 사용되는 상기 제 2 복수의 이웃 화소들 중 적어도 일부의 이웃 화소들과 대응하는, 인트라 예측을 수행하는 방법.

청구항 2

제 1 항에 있어서,

상기 제 2 시구간은 상기 제 1 시구간의 실질적으로 직후인, 인트라 예측을 수행하는 방법.

청구항 3

제 1 항에 있어서,

상기 제 1 복수의 이웃 화소들의 상기 준비하는 단계는 상기 제 1 복수의 이웃 화소들을 패딩 및 필터링하는 단계를 포함하는, 인트라 예측을 수행하는 방법.

청구항 4

제 1 항에 있어서,

상기 제 1 및 제 2 블록들은 공통 인트라 예측 모드에 둘 다 연관되는 변환 블록들인, 인트라 예측을 수행하는 방법.

청구항 5

제 1 항에 있어서,

상기 인트라 예측 프로세싱 부는 수 N의 프로세싱 엔진들을 포함하고, 각각의 프로세싱 엔진은 상기 제 1 시구간의 각각의 사이클 동안 상이한 화소를 인트라 예측하는, 인트라 예측을 수행하는 방법.

청구항 6

제 5 항에 있어서,

상기 제 1 블록은 M 개 화소들 바이 M 개 화소들의 사이즈를 갖고, 상기 인트라 예측 프로세싱 부가 상기 제 1 블록을 결정하는 상기 제 1 시구간은 대략 $M \times M/N$ 개 사이클들 동안 지속되는, 인트라 예측을 수행하는 방법.

청구항 7

제 6 항에 있어서,

상기 제 2 복수의 이웃 셀들의 상기 준비하는 단계는 상기 제 1 블록의 상기 결정하는 단계가 끝난 지 대략 $2M/N$ 개 사이클들 후에 시작하는, 인트라 예측을 수행하는 방법.

청구항 8

제 5 항에 있어서,

상기 이웃 프로세싱 부는 사이클당 $2N$ 개 화소들의 판독 대역폭을 갖는 이웃 베퍼로부터 상기 제 1 복수 및 제 2 복수의 이웃 화소들을 수신하는, 인트라 예측을 수행하는 방법.

청구항 9

제 6 항에 있어서,

상기 제 1 복수의 이웃 화소들은 $4M+1$ 개 화소들을 포함하는, 인트라 예측을 수행하는 방법.

청구항 10

제 1 항에 있어서,

상기 제 1 블록의 상기 결정하는 단계는 인트라 예측된 화소들의 중대한 행 또는 열을 상기 제 1 블록의 나머지를 결정하기 전에 결정하는 단계를 포함하며, 인트라 예측된 화소들의 상기 중대한 행 또는 열은 상기 제 2 블록을 결정하는데 사용된 상기 제 2 복수의 이웃 화소들 중 적어도 일부의 이웃 화소들과 대응하는, 인트라 예측을 수행하는 방법.

청구항 11

인트라 예측을 수행하는 프로세서로서,

인트라 예측을 위한 제 1 복수 및 제 2 복수의 이웃 화소들을 수신 및 준비하도록 동작 가능한 이웃 프로세싱 부; 및

준비된 상기 제 1 복수의 이웃 화소들을 사용하여 제 1 시구간 동안 제 1 복수의 인트라 예측된 화소들을 갖는 제 1 블록을 결정하도록 동작 가능한 인트라 예측 프로세싱 부로서, 상기 인트라 예측 프로세싱 부는 또한 준비된 상기 제 2 복수의 이웃 화소들을 사용하여 제 2 시구간 동안 제 2 복수의 인트라 예측된 화소들을 갖는 제 2 블록을 결정하도록 동작 가능한, 상기 인트라 예측 프로세싱 부를 포함하며,

상기 제 1 블록에서의 상기 제 1 복수의 인트라 예측된 화소들 중 적어도 일부의 인트라 예측된 화소들은 상기 제 2 블록을 결정하기 위해 사용되는 상기 제 2 복수의 이웃 화소들 중 적어도 일부의 이웃 화소들과 대응하며; 그리고

상기 이웃 프로세싱 부는 또한 상기 제 1 시구간의 적어도 일부 동안 상기 제 2 복수의 이웃 화소들을 준비하도록 동작 가능한, 인트라 예측을 수행하는 프로세서.

청구항 12

제 11 항에 있어서,

상기 제 2 시구간은 상기 제 1 시구간의 실질적으로 직후인, 인트라 예측을 수행하는 프로세서.

청구항 13

제 11 항에 있어서,

상기 이웃 프로세싱 부는 또한 상기 제 1 복수의 이웃 화소들을 패딩 및 필터링하도록 동작 가능한, 인트라 예

측을 수행하는 프로세서.

청구항 14

제 11 항에 있어서,

상기 제 1 및 제 2 블록들은 공통 인트라 예측 모드에 둘 다 연관되는 변환 블록들인, 인트라 예측을 수행하는 프로세서.

청구항 15

제 11 항에 있어서,

상기 인트라 예측 프로세싱 부는 수 N 의 프로세싱 엔진들을 포함하고, 각각의 프로세싱 엔진은 상기 제 1 시구간의 각각의 사이클 동안 상이한 화소를 인트라 예측하도록 동작 가능한, 인트라 예측을 수행하는 프로세서.

청구항 16

제 15 항에 있어서,

상기 제 1 블록은 M 개 화소들 바이 M 개 화소들의 사이즈를 갖고, 상기 제 1 시구간은 대략 M^2/N 개 사이클들 동안 지속되는, 인트라 예측을 수행하는 프로세서.

청구항 17

제 16 항에 있어서,

상기 이웃 프로세싱 부는 상기 인트라 예측 프로세싱 부가 상기 제 1 블록을 결정하는 것을 끝낸 지 대략 $2M/N$ 개 사이클들 후 상기 제 2 복수의 이웃 셀들을 준비하는 것을 시작하도록 동작 가능한, 인트라 예측을 수행하는 프로세서.

청구항 18

제 15 항에 있어서,

상기 제 1 복수 및 제 2 복수의 이웃 화소들을 저장하도록 동작 가능한 이웃 베퍼를 더 포함하며, 상기 이웃 베퍼는 사이클당 $2N$ 개 화소들의 판독 대역폭을 갖는, 인트라 예측을 수행하는 프로세서.

청구항 19

제 16 항에 있어서,

상기 제 1 복수의 이웃 화소들은 $4M+1$ 개 화소들을 포함하는, 인트라 예측을 수행하는 프로세서.

청구항 20

제 11 항에 있어서,

상기 인트라 예측 프로세싱 부는 또한 상기 제 1 블록에서의 인트라 예측된 화소들의 중대한 행 또는 열을 상기 제 1 블록의 나머지를 결정하기 전에 결정하도록 동작 가능하며, 인트라 예측된 화소들의 상기 중대한 행 또는 열은 상기 제 2 블록을 결정하는데 사용된 상기 제 2 복수의 이웃 화소들 중 적어도 일부의 이웃 화소들과 대응하는, 인트라 예측을 수행하는 프로세서.

청구항 21

비디오 디코더로서,

인트라 예측을 수행하도록 동작 가능한 인트라 예측 모듈로서,

인트라 예측을 위한 제 1 복수 및 제 2 복수의 이웃 화소들을 수신 및 준비하도록 동작 가능한 이웃 프로세싱 부; 및

준비된 상기 제 1 복수의 이웃 화소들을 사용하여 제 1 시구간 동안 제 1 복수의 인트라 예측된 화소들을 갖는 제 1 블록을 결정하도록 동작 가능한 인트라 예측 프로세싱 부로서, 상기 인트라 예측 프로세싱 부는 또한 준비

된 상기 제 2 복수의 이웃 화소들을 사용하여 제 2 시간 동안 제 2 복수의 인트라 예측된 화소들을 갖는 제 2 블록을 결정하도록 동작 가능한, 상기 인트라 예측 프로세싱 부를 포함하며,

상기 이웃 프로세싱 부는 또한 상기 제 1 시구간의 적어도 일부 동안 상기 제 2 복수의 이웃 화소들을 준비하도록 동작 가능한, 상기 인트라 예측 모듈;

제 1 잔차 블록과 제 2 잔차 블록을 제공하도록 동작 가능한 역 변환 모듈; 및

제 1 복원된 블록을 결정하기 위해 상기 제 1 블록과 상기 제 1 잔차 블록을 가산하도록 동작 가능한 합산기로서, 상기 합산기는 또한 제 2 복원된 블록을 결정하기 위해 상기 제 2 블록과 상기 제 2 잔차 블록을 가산하도록 동작 가능한, 상기 합산기를 포함하며,

상기 제 1 복원된 블록은 상기 제 2 블록을 결정하기 위해 사용되는 상기 제 2 복수의 이웃 화소들 중 적어도 일부의 이웃 화소들을 포함하는, 비디오 디코더.

청구항 22

제 21 항에 있어서,

상기 제 1 및 제 2 복원된 블록들을 참조 프레임의 부분들로서 저장하도록 동작 가능한 참조 프레임 버퍼를 더 포함하며,

상기 참조 프레임은 후속 모션 보상, 인트라 예측, 및 프레젠테이션 디바이스 상의 디스플레이 중 적어도 하나를 위해 사용될 수도 있는, 비디오 디코더.

청구항 23

제 21 항에 있어서,

상기 인트라 예측 프로세싱 부는 수 N 의 프로세싱 엔진들을 포함하고, 각각의 프로세싱 엔진은 상기 제 1 시구간의 각각의 사이클 동안 상이한 화소를 인트라 예측하도록 동작 가능한, 비디오 디코더.

청구항 24

제 23 항에 있어서,

상기 제 1 블록은 M 개 화소들 바이 M 개 화소들의 사이즈를 갖고, 상기 제 1 시구간은 대략 $M*M/N$ 개 사이클들 동안 지속되는, 비디오 디코더.

청구항 25

제 23 항에 있어서,

상기 제 1 복수 및 제 2 복수의 이웃 화소들을 저장하도록 동작 가능한 이웃 버퍼를 더 포함하며, 상기 이웃 버퍼는 사이클당 $2N$ 개 화소들의 판독 대역폭을 갖는, 비디오 디코더.

청구항 26

제 23 항에 있어서,

상기 이웃 버퍼는 $16N$ 의 RAM 폭을 갖는 이웃 RAM을 포함하는, 비디오 디코더.

청구항 27

제 24 항에 있어서,

상기 제 1 복수의 이웃 화소들은 $4M+1$ 개 화소들을 포함하는, 비디오 디코더.

청구항 28

제 21 항에 있어서,

상기 인트라 예측 프로세싱 부는 또한 상기 제 1 블록에서의 인트라 예측된 화소들의 중대한 행 또는 열을 상기 제 1 블록의 나머지를 결정하기 전에 결정하도록 동작 가능하며, 인트라 예측된 화소들의 상기 중대한 행 또는

열은 상기 제 2 블록을 결정하는데 사용된 상기 제 2 복수의 이웃 화소들 중 적어도 일부의 이웃 화소들과 대응하는, 비디오 디코더.

청구항 29

제 28 항에 있어서,

상기 역 변환 모듈은 또한 상기 중대한 행 또는 열에 대응하는 잔차 값들을 생성하도록 동작 가능한, 비디오 디코더.

청구항 30

제 29 항에 있어서,

상기 합산기는 또한 상기 제 2 복수의 이웃 화소들 중 적어도 일부의 이웃 화소들을 생성하기 위해 상기 중대한 행 또는 열과 상기 대응하는 잔차 값들을 가산하도록 동작 가능한, 비디오 디코더.

발명의 설명

배경기술

[0001]

1. 기술 분야

[0002]

본 출원은 대체로 비디오 인코더 및 비디오 디코더들에서의 인트라 예측에 관한 것이고, 더 구체적으로는, 파이프라이닝 기법들을 사용한 인트라 예측의 하드웨어 구현예들에 관한 것이다.

[0003]

2. 관련 기술

[0004]

비디오 코덱들, 이를테면 고 효율 비디오 코딩 (High Efficiency Video Coding, HEVC) 에서 사용되는 것들이, 모바일 디바이스들, 스마트 TV들, 및 많은 다른 비디오 애플리케이션들에서의 사용을 위해 비디오를 압축하는 다양한 기법들을 통합한다. 하나의 이러한 기법은 인트라 예측인데, 그것은 비디오 스트리밍을 위해 요구된 파일 사이즈 및/또는 대역폭을 감소시키기 위해 비디오 프레임 내의 공간적 리던던시들을 이용한다.

[0005]

비디오 프레임이 화소들의 블록들로 일반적으로 세그먼트화되고, 주어진 블록이 그것 바로 주위의 블록들 및 화소들과 종종 고도로 상관된다. 따라서, 블록 내의 화소들은 이미 계산된 이웃 화소들로부터 보간될 수 있다. 의존성의 성질은 인트라 예측 모드로서 저장될 수도 있다. 일단 인트라 예측 모드가 결정되면, 보간된 화소들 및 실제 화소들 간의 차이들 (예컨대, 잔차들) 만이 인코딩될 것이 필요하다. 이를 잔차들은 후속 프로세싱 스테이지들에서 인코딩되는데, 후속 프로세싱 스테이지들은 이산 코사인 변환 (discrete cosine transformation, DCT), 양자화, 및 엔트로피 코딩을 포함한다. 공간적 리던던시를 고려함으로써, 인트라 예측은 전체 압축 비율을 크게 증가시킬 수 있다.

[0006]

HEVC에서, 일부 참조 프레임들 (예컨대, 인트라 코딩식 뷐처들 또는 I-프레임들) 이 인트라 예측 (인터 예측과는 대조적임) 에 의해 전적으로 인코딩되고 디코딩된다. 참조 프레임들은 복수의 코딩 유닛들 (coding units, CU) 을 포함할 수도 있고, 각각의 CU는 사이즈 $M \times M$ 의 변환 유닛들 (transform units, TU) 로 나누어질 수도 있다. CU는 크로마 또는 루마 정보에 연관될 수도 있는 코딩 블록들 (coding blocks, CB) 로 추가로 나누어질 수도 있다. 유사하게, TU들은 변환 블록들 (transform blocks, TB) 로 나누어질 수도 있다. 인트라 예측 모드들은 TB 레벨에 계산적으로 적용될 수도 있다.

[0007]

HEVC 표준은, $4M+1$ 이웃 화소들이 인트라 예측을 수행하는데 사용될 수도 있고, 이를 이웃 화소들은 그러므로 주어진 블록 (예컨대, TU) 에 대한 인트라 예측이 시작되기 전에 다른 방도들로 패딩, 필터링, 및 잠재적으로 컨디셔닝되어야만 한다는 것을 묘사한다. 블록의 인트라 예측이 완료된 후, 다음 블록의 인트라 예측이 시작된다. 두 개의 블록들은 순차적으로 프로세싱될 수도 있는데, 제 2 블록의 인트라 예측이 제 1 블록 및/또는 다른 이전에 계산된 블록들의 인트라 예측으로부터 도출되는 복원된 화소들에 일반적으로 의존할 수도 있기 때문이다. 따라서, HEVC 표준은 이 스테이지에서 어느 정도의 각각의 블록의 순차적 프로세싱을 본질적으로 요구하며, 이는 많은 다른 비디오 표준들에 대해서도 적용된다.

발명의 내용

[0008]

위에서 설명된 순차적 프로세싱의 요건이 주어진다고 하면, 본 개시물은 인트라 예측의 연속적인 블록들 (예컨

대, 코딩 유닛 내의 연속적인 변환 유닛들) 간의 레이턴시를 줄이는 것을 목표로 한다. 제 2 블록의 인트라 예측은 종종 제 1 블록에 연관된 모든 복원된 화소에 의존하는 것이 아니라, 제 1 및 제 2 블록들의 상대 위치에 따라 달라지는, 단일의 중대한 행 또는 열에만 의존한다는 것에 주의한다. 각각의 블록이, 통상적으로 행하지는 바와 같이 레스터 스캔을 사용하여 인트라 예측된다면, 중대한 행 또는 열은 제 1 블록의 프로세싱 사이클의 맨 끝을 향하여 인트라 예측된다. 중대한 행 또는 열의 화소들의 복원, 필터링, 패딩, 및/또는 다른 컨디셔닝이 많은 프로세싱 사이클들을 필요로 할 수 있으므로, 제 2 블록의 인트라 예측은 지연된다. HEVC에서, 각각의 코딩 유닛은 256 개까지의 변환 유닛들을 포함할 수 있고, 그래서 이들 지연들은 전체 성능을 줄이도록 빠르게 합산될 수 있다.

[0009] 제안된 하드웨어 아키텍처는 제 1 블록에서의 중대한 행들 및 열들에 대한 화소 생성을 우선시 (prioritization) 하도록 동작 가능한 인트라 예측 프로세싱 부를 이용할 수도 있다. 이웃 프로세싱 부가 중대한 행들 및 열들에 대응하는 복원된 화소들을 수신할 수도 있고 제 1 블록의 인트라 예측이 완료되기 전에 제 2 블록의 인트라 예측을 위한 이웃 화소들을 추가로 준비 (예컨대, 패딩 및 필터링) 할 수도 있다. 이는 인트라 예측 프로세싱 부가 블록들의 각각 간을 빠르게 전환하는 것을 허용한다. 이웃 프로세싱 및 인트라 예측을 인터리빙함으로써, 인트라 예측의 연속적인 블록들 (예컨대, 코딩 유닛 내에 있는 인접한 변환 유닛들) 간의 레이턴시는 제거되거나 또는 감소된다.

[0010] 더욱이, 복수의 인트라 예측 엔진들은 주어진 블록을 프로세싱하기 위해 병렬로 동작할 수도 있다. 인트라 예측 엔진들의 수 (N) 는 이웃 화소들을 저장하는데 사용되는 이웃 버퍼 (예컨대, 이웃 RAM을 포함함)의 관독 대역폭 (예컨대, 사이클당 $2N$ 개 화소들)으로 스케일링될 수도 있다. 대체로, 인트라 예측 엔진들에 의해 사용되는 메모리의 최대 관독 레이트는 인트라 예측 엔진들이 화소들을 복원을 위해 (그리고, 예컨대, 후속 블록들의 인트라 예측을 위해) 출력하는 레이트에 근접하여 일치될 수도 있다. 개시된 시스템들 및 방법들은 그에 따라 관독 대역폭에 연관된 공통 병목을 관리하도록 스케일링할 수도 있다.

도면의 간단한 설명

[0011] 본 개시물의 특징들, 양태들, 및 실시형태들은 첨부된 도면들에 연계하여 설명되는데, 도면들 중에서:

도 1은 본 개시물의 원리들에 따라 파이프라인식 인트라 예측을 위한 기법들을 이용하도록 구성될 수도 있는 비디오 인코딩 및 디코딩 시스템의 일 예를 예시하는 블록도를 도시하며;

도 2는 개시된 원리들에 따라 구현될 수도 있는 비디오 인코더의 일 예를 예시하는 블록도를 도시하며;

도 3은 개시된 원리들에 따라 구현될 수도 있는 비디오 디코더의 일 예를 예시하는 블록도를 도시하며;

도 4는 인트라 예측을 위한 다양한 모드들을 예시하는 개략도를 도시하며;

도 5는 이웃하는 블록들에 대한 블록의 의존성들을 예시하는 개략도를 도시하며;

도 6은 인트라 예측 서브시스템을 예시하는 블록도를 도시하며;

도 7은 인트라 예측을 위해 선택되는 코딩 블록을 예시하는 블록도를 도시하며; 그리고

도 8은 인터리브식 인트라 예측 기법을 예시하는 타이밍도를 도시한다.

이들 예시적인 도면들 및 실시형태들은 본 출원에서 제시된 임의의 청구항들에 의해 언급된 발명의 주제 (subject matter)의 기재된 상세한 설명을 제공하는 것이다. 이들 예시적인 도면들 및 실시형태들은 임의의 이러한 청구항들의 범위를 제한하는데 사용되지 않아야 한다.

게다가, 비록 유사한 참조 번호들이 유사한 구조들을 언급하는데 편의를 위해 사용될 수도 있지만, 다양한 예시적인 실시예들의 각각은 별개의 변형들인 것으로 간주될 수도 있다.

발명을 실시하기 위한 구체적인 내용

[0012] 도 1은 본 개시물의 원리들에 따라 파이프라인식 인트라 예측을 위한 기법들을 이용하도록 구성될 수도 있는 비디오 인코딩 및 디코딩 시스템 (10)의 일 예를 예시하는 블록도를 도시한다. 도 1에 도시된 바와 같이, 시스템 (10)은 인코딩된 비디오를 통신 채널 (16)을 통해 목적지 디바이스 (14)로 송신하는 소스 디바이스 (12)를 포함한다. 인코딩된 비디오는 저장 매체 (34) 또는 파일 서버 (36) 상에 또한 저장될 수도 있고 원하는 대로 목적지 디바이스 (14)에 의해 액세스될 수도 있다. 소스 디바이스 (12)와 목적지 디바이스

(14) 는 데스크톱 컴퓨터들, 노트북 (즉, 랙톱) 컴퓨터들, 태블릿 컴퓨터들, 셋톱 박스들, 전화기 핸드셋들 (예컨대, 스마트폰들 또는 다른 개인용 통신 디바이스들), 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 콘솔들, 모바일 디바이스들 등을 포함한 매우 다양한 디바이스들 중 임의의 것을 포함할 수도 있다. 많은 경우들에서, 이러한 디바이스들은 무선 통신, 유선 통신, 또는 둘 다를 위해 갖추어질 수도 있다. 따라서, 통신 채널 (16) 은 무선 채널 (예컨대, Wi-Fi 접속), 유선 채널 (예컨대, DSL, 케이블 모뎀, 이더넷, USB), 또는 인코딩된 비디오 데이터의 송신에 적합한 무선 및 유선 채널들의 조합을 포함할 수도 있다.

[0013] 본 개시물의 원리들은 다양한 멀티미디어 애플리케이션들, 이를테면 OTA (over-the-air) 텔레비전 브로드캐스트들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, 스트리밍 비디오 송신들 (예컨대, 인터넷을 통함), 데이터 저장 매체 상의 저장을 위한 디지털 비디오의 인코딩, 데이터 저장 매체 상에 저장된 디지털 비디오의 디코딩, 또는 다른 애플리케이션들 중 임의의 것의 지원 하의 비디오 코딩에 적용될 수도 있다.

[0014] 일부 실시형태들에서, 디바이스들 (12, 14) 은 디바이스들 (12, 14) 의 각각이 비디오 인코딩 및 디코딩 컴포넌트들을 포함하도록 실질적으로 대칭적인 방식으로 동작할 수도 있다. 따라서, 시스템 (10) 은 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅, 또는 비디오폰 통화 (video telephony) 와 같은 애플리케이션들을 지원하기 위해 단방향 비디오 송신 및 양방향 비디오 송신 둘 다를 지원하도록 구성될 수도 있다.

[0015] 도 1의 실시형태에서, 소스 디바이스 (12) 는 비디오 소스 (18), 비디오 인코더 (20), 변조기/복조기 (모뎀) (22) 및 송신기 (24) 를 포함한다. 비디오 소스 (18) 는 비디오 캡처 디바이스 (예컨대, 비디오 카메라), 이전에 캡처된 비디오를 담고 있는 비디오 아카이브, 비디오 콘텐츠 제공자로부터 비디오를 수신하도록 동작 가능한 비디오 퍼드 인터페이스, 및 컴퓨터 그래픽 데이터를 소스 비디오로서 생성하는 컴퓨터 그래픽 시스템의 임의의 조합 또는 개개의 사례들을 포함할 수도 있다. 일부 실시형태들에서, 비디오 소스 (18) 는 비디오 카메라일 수도 있고, 소스 디바이스 (12) 와 목적지 디바이스 (14) 는 카메라 폰들 또는 비디오 폰들을 형성할 수도 있다. 그러나, 본 개시물에서 설명되는 기법들은 일반적으로 비디오 코딩에 적용가능하고, 무선 및/또는 유선 애플리케이션들에 적용될 수도 있다.

[0016] 캡처된, 사전 캡처된 (pre-captured), 또는 컴퓨터 생성된 비디오는 비디오 인코더 (20) 에 의해 인코딩될 수도 있다. 인코딩된 비디오 정보는 통신 표준, 이를테면 무선 통신 프로토콜에 따라 모뎀 (22) 에 의해 변조될 수도 있고, 목적지 디바이스 (14) 에게 송신기 (24) 를 통해 송신될 수도 있다. 모뎀 (22) 은 다양한 막서들, 필터들, 증폭기들, 및 신호 변조를 위해 설계된 다른 컴포넌트들을 포함할 수도 있다. 송신기 (24) 는 증폭기들, 필터들, 및 하나 이상의 안테나들을 포함하는, 데이터를 송신하기 위해 설계된 회로들을 포함할 수도 있다.

[0017] 비디오 인코더 (20) 에 의해 인코딩되는 캡처된, 사전 캡처된, 또는 컴퓨터 생성된 비디오는 나중의 소비를 위해 저장 매체 (34) 또는 파일 서버 (36) 상에 또한 저장될 수도 있다. 저장 매체 (34) 는 블루레이 디스크들, DVD들, CD-ROM들, 플래시 메모리, 또는 인코딩된 비디오를 저장하는 임의의 다른 적합한 디지털 저장 매체들을 포함할 수도 있다. 저장 매체 (34) 상에 저장된 인코딩된 비디오는 그 다음에 디코딩 및 플레이백을 위해 목적지 디바이스 (14) 에 의해 액세스될 수도 있다.

[0018] 파일 서버 (36) 는 인코딩된 비디오를 저장하고 그 인코딩된 비디오를 목적지 디바이스 (14) 에게 송신할 수 있는 임의의 유형의 서버일 수도 있다. 예의 파일 서버들은 웹 서버 (예컨대, 웹사이트 용), FTP 서버, 네트워크 접속 스토리지 (network attached storage, NAS) 디바이스들, 로컬 디스크 드라이브, 또는 인코딩된 비디오 데이터를 저장하고 그것을 목적지 디바이스에게 송신할 수 있는 임의의 다른 유형의 디바이스를 포함한다.

파일 서버 (36) 로부터의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 양쪽 모두의 조합일 수도 있다. 파일 서버 (36) 는 목적지 디바이스 (14) 에 의해, 인터넷 접속을 포함한 임의의 표준 데이터 접속을 통해 액세스될 수도 있다. 이는 무선 채널 (예컨대, Wi-Fi 접속), 유선 접속 (예컨대, DSL, 케이블 모뎀, 이더넷, USB), 또는 파일 서버 상에 저장된 인코딩된 비디오 데이터에 액세스하기에 적합한 양쪽 모두의 조합을 포함할 수도 있다.

[0019] 도 1의 예에서, 목적지 디바이스 (14) 는 수신기 (26), 모뎀 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (32) 를 포함한다. 수신기 (26) 는 채널 (16) 을 통해 정보를 수신하고, 모뎀 (28) 은 비디오 디코더 (30) 를 위한 복조된 비트스트림을 생성하기 위해 그 정보를 복조한다. 채널 (16) 을 통해 통신되는 정보는 비디오 데이터를 디코딩함에 있어서 비디오 디코더 (30) 에 의한 사용을 위해 비디오 인코더 (20) 에 의해 생성된 다양한 신택스 정보를 포함할 수도 있다. 그런 신택스는 저장 매체 (34) 또는 파일 서버 (36) 상에 저장

되는 인코딩된 비디오 데이터에 또한 포함될 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30)의 각각은 비디오 데이터를 인코딩 또는 디코딩할 수 있는 각각의 인코더-디코더 (CODEC)의 부분을 형성할 수도 있다.

[0020] 디스플레이 디바이스 (32)는 목적지 디바이스 (14)와 통합되거나, 또는 그것 외부에 있을 수도 있다. 일부 예들에서, 목적지 디바이스 (14)는 통합형 디스플레이 디바이스를 포함할 수도 있고 외부 디스플레이 디바이스와 인터페이싱하도록 또한 구성될 수도 있다. 다른 예들에서, 목적지 디바이스 (14)는 디스플레이 디바이스일 수도 있다. 일반적으로, 디스플레이 디바이스 (32)는 디코딩된 비디오 데이터를 사용자에게 디스플레이하고, 음극선관 (cathode ray, CRT), 액정 디스플레이 (liquid crystal display, LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (organic light emitting diode, OLED) 디스플레이, 또는 다른 유형의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들 중 임의의 것을 포함할 수도 있다.

[0021] 도 1에 도시된 실시형태에서, 통신 채널 (16)은 임의의 무선 또는 유선 통신 매체, 이를테면 라디오 주파수 (radio frequency, RF) 스펙트럼 또는 하나 이상의 물리적 송신 라인들, 또는 무선 및 유선 매체들의 임의의 조합을 포함할 수도 있다. 통신 채널 (16)은 패킷 기반 네트워크, 이를테면 국부 영역 네트워크, 광역 네트워크, 또는 글로벌 네트워크 (예컨대, 인터넷)의 부분을 형성할 수도 있다. 통신 채널 (16)은, 비디오 데이터를 소스 디바이스 (12)로부터 목적지 디바이스 (14)로 또는 소스 디바이스 (12)와 목적지 디바이스 (14) 간에 양방향적으로 송신하기 위한, 임의의 적합한 통신 매체, 또는 상이한 통신 매체들의 콜렉션을 일반적으로 나타낸다. 통신 채널 (16)은 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (12)와 목적지 디바이스 (14) 간의 통신을 용이하게 하는데 유용할 수도 있는 임의의 다른 장비를 포함할 수도 있다.

[0022] 고 효율 비디오 코딩 (HEVC) 표준은, ITU-T 비디오 코딩 전문가 그룹 (Video Coding Experts Group, VCEG) 및 ISO/IEC 동 화상 전문가 그룹 (Motion Picture Experts Group, MPEG)의 JCT-VC (Joint Collaboration Team on Video Coding)에 의해 개발되고 있다. "HEVC 규격 초안 7"이라고 지칭되는 HEVC 표준의 최근 초안은 2012년 6월 7일 현재 http://phenix.it-sudparis.eu/jct/doc_end_user/documents/9_Geneva/wg11/JCTVC-I1003-v3로부터 다운로드 가능하다. HEVC 규격 초안 7에 대한 완전한 언급은 문서 JCTVC-I1003, 『Bross et al., "High Efficiency Video Coding (HEVC) Text Specification Draft 7," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 9th Meeting: Geneva, Switzerland, April 27, 2012 to May 7, 2012』이다. "HEVC 규격 초안 (WD) 8"이라고 지칭되는 HEVC 표준의 추가적인 최근 초안이, http://phenix.int-evry.fr/jct/doc_end_user/documents/10_Stockholm/wg11/JCTVC-J1003-v8.zip으로부터 입수 가능하다. 승인된 HEVC 규격은 2013년 4월 현재 <http://www.itu.int/rec/T-REC-H.265~201304-I>에서 발견될 수 있다. 본 단락에서 언급된 참조문헌들의 각각은 본 명세서에 그 전부가 참조로 포함된다.

[0023] HEVC 표준은 비디오 프레임 또는 블록이 코딩 트리 유닛들 (coding tree units, CTU)이라고 또한 지칭될 수도 있는 최대 코딩 유닛들 (largest coding units, LCU)의 시퀀스로 나누어질 수도 있다는 것을 설명한다. 각각의 CTU는 복수의 코딩 트리 블록들 (예컨대, 하나의 루마 CTB 및 두 개의 크로마 CTB들)을 포함할 수도 있다. 각각의 CTU는 쿼드트리에 따라 코딩 유닛 (CU)들로 또한 분할될 수도 있다. 일반적으로, 쿼드트리 데이터 구조가 CU당 하나의 노드를 포함하며 루트 노드가 CTU에 대응한다. 예를 들어, CTU의 쿼드트리가 네 개의 자식 노드들 (이것들은 서브 CU들에 연관될 수도 있음)로 분할될 수도 있고, 이를 자식 노드들 중 하나 이상은 게다가 다른 네 개의 자식 노드들로 분할되는 부모 노드일 수도 있다. 리프 노드라고 또한 지칭되는 각각의 최종, 쿼드트리의 비분할 자식 노드는, 리프 CU와 대응할 수도 있다. 일반적으로, CU가 일부 비-HEVC 표준들 (예컨대, H.264 표준)의 매크로블록과는, 몇 가지 차이들 (예컨대, CU들이 사이즈가 가변할 수도 있다)과 함께, 유사한 목적을 갖는다. 따라서, "CU" 및 "매크로블록"이란 용어들은, 일반론의 어떠한 손실도 없이, 본 문서에서 교환적으로 사용될 수도 있다.

[0024] 각각의 CU는 코딩 노드뿐만 아니라 그 코딩 노드에 연관된 하나 이상의 예측 유닛들 (prediction units, PU들) 및 하나 이상의 변환 유닛들 (TU들)을 포함할 수도 있다. CU 사이즈는 코딩 노드의 사이즈에 대응할 수도 있고 형상이 일반적으로 정사각형일 수도 있다. 예를 들어, CU 사이즈는 8x8 화소들부터 CTU의 사이즈까지의 범위일 수도 있는데, CTU의 사이즈는 최대 사이즈 (예컨대, 64x64 개 화소들)를 가질 수도 있다.

[0025] 본 개시물에서, "MxM" 및 "M 바이 M"은 각각 수평 및 수직 치수들의 측면에서 비디오 블록의 화소 치수들을 교환적으로 지칭 (예컨대, 16x16 개 화소들 또는 16 바이 16 개 화소들)하는데 사용될 수도 있다. 일반적으로, 16x16 블록이 수평 방향의 16 개 화소들 ($x = 16$)과 수직 방향의 16 개 화소들 ($y = 16$)을 가질 것이다. 비슷하게, MxM 블록이 일반적으로 수평 방향의 M 개 화소들과 수직 방향의 M 개 화소들을 가지며, 여기서 M

은 음이 아닌 정수 값을 나타낸다. 블록에서의 화소들은 행들 및 열들로 배열될 수도 있다. 블록들 (예컨대, HEVC에서의 TU들)에 대한 공통의 사이즈들이 4x4, 8x8, 16x16, 및 32x32를 포함할 수도 있다. 그러나, 블록들은 수직 방향에서와 동일한 수의 화소들을 수평 방향에서 반드시 가질 필요는 없다. 예를 들어, 블록들은 P가 M과 반드시 동일하지는 않은 MxP 개 화소들을 포함할 수도 있다.

[0026] 리프 CU에 연관된 선택스 데이터는, 하나 이상의 TU들로의 리프 CU의 구획화를 설명할 수도 있다. 일부 예들에서, 리프 CU에 대응하는 잔차 샘플들은 "잔차 쿼드 트리 (residual quad tree)" (RQT)로서 알려진 쿼드트리 구조를 사용하여 세분될 수도 있는데, 여기서 RQT의 리프 노드들이 TU들에 연관된다. 다시 말하면, 리프 CU가 TU들로 구획화되는 방법을 나타내는 쿼드트리를 리프 CU는 포함할 수도 있다. RQT (예컨대, TU 쿼드트리)가 리프 CU에 일반적으로 대응하는 루트 노드를 가질 수도 있는 한편, CU 쿼드트리의 루트 노드는 LCU에 일반적으로 대응할 수도 있다. 따라서, 하나 이상의 TU 쿼드트리들은 CU 쿼드트리 내에 네스팅될 수도 있다.

[0027] 분할되지 않는 RQT의 자식 노드들은 리프 TU들과 연관될 수도 있다. 일반적으로, 본 개시물은 달리 언급되지 않는 한, CU 및 TU라는 용어들을 리프 CU 및 리프 TU를 각각 지칭하기 위해 사용한다.

[0028] HEVC 표준은 TU 사이즈가 CU들 간에 가변할 수도 있는 TU 유닛 기반의 변환들을 허용한다. TU가 형상이 정사각형일 수 있거나 또는 정사각형이 아닐 (예컨대, 직사각형일) 수도 있다. 게다가, TU들은 인트라 예측 동안 통상적으로는 주어진 CU 내의 PU들과는 동일한 사이즈이거나 또는 그러한 PU들보다 작지만, TU들은 인터 예측된 CU의 대응하는 PU들보다 더 클 수도 있다. TU들에 연관된 화소 차이 값들은 양자화될 수도 있는 변환 계수들을 생성하기 위해 변환될 수도 있다.

[0029] CU에 연관된 선택스 데이터는, PU들이 예측에 관련된 데이터를 포함하는 하나 이상의 PU들로의 CU의 구획화를 또한 설명할 수도 있다. 일반적으로, PU가 대응하는 CU의 전부 또는 부분에 대응하는 공간 영역을 나타내고, PU에 대한 참조 샘플을 취출하기 위한 데이터를 포함할 수도 있다. 가용 PU 구획화 모드들은 CU가 스kip되는지 또는 직접 모드 인코딩되는지, 인트라 예측 모드 인코딩되는지, 또는 인터 예측 모드 인코딩되는지에 의존하여 가변할 수도 있다. 예를 들어, CU가 인트라 예측 모드 인코딩되는 경우, 그 CU는 CU와 동일한 사이즈를 갖는 단일 PU 또는 CU의 사이즈 각각의 4 분의 1인 네 개의 정사각형 PU들과 대응할 수도 있다. 게다가, PU에 대한 데이터 (예컨대, 인트라 예측 모드 표시자)는 RQT에 포함될 수도 있어서, 그 데이터는 PU에 대응하는 하나 이상의 TU들에 적용될 수도 있다. 일부 시나리오들에서, PU가 형상이 비-정사각형 (예컨대, 직사각형) 일 수도 있다.

[0030] 일반적으로, 인트라 코딩 (예컨대, 인트라 예측 코딩)에 대해, CU에 속한 모든 TU들은 동일한 인트라 예측 모드를 공유한다. 다시 말하면, CU가 인트라 예측 코딩을 위한 PU와 동일한 사이즈일 수도 있음으로써, CU의 모든 TU들에 대해 예측된 값들을 계산하는 공통 인트라 예측 모드를 가질 수도 있다. 일부 시나리오들에서, 인트라 코딩된 CU가 네 개의 PU들로 나누어질 수도 있고, CU는 적어도 네 개의 TU들로 또한 나누어질 수도 있다. 따라서, 인트라 예측 동안, TU들은 사이즈가 대응하는 PU보다 더 작거나 또는 그 PU와 동일하다. 인트라 코딩을 위해, 비디오 인코더가 인트라 예측 모드를 사용하여 각각의 TU에 대한 잔차 값을, 그 TU에 대응하는 CU의 부분 및 원래의 블록 간의 차이로서 계산할 수도 있다.

[0031] 인터 예측 동안, TU가 사이즈가 공통의 CU 내의 PU보다 더 크거나, 더 작거나, 또는 그 PU와 동일할 수도 있다. 따라서, TU가 대응하는 PU들의 각각보다 커서, TU가 복수의 PU들에 대응할 수도 있다. 대안적으로, PU가 대응하는 TU들의 각각보다 커서, PU가 복수의 TU들에 대응할 수도 있다. 그러나, PU와 TU 중 어느 것도 대응하는 CU보다 크지 않을 수도 있다.

[0032] 비디오 인코더 (20) 와 비디오 디코더 (30)는 비디오 압축 표준, 이를테면 HEVC 표준 또는 HEVC 표준에 대한 확장본들에 따라 동작할 수도 있다. 대안으로 또는 추가적으로, 비디오 인코더 (20) 와 비디오 디코더 (30)는, 다르게는 MPEG-4, 파트 10, 고급 비디오 코딩 (Advanced Video Coding, AVC) 이라고 지칭되는 ITU-T (International Telecommunication Union Telecommunication Standardization Sector) H.264 표준과 같은 다른 독점 또는 산업 표준들, 또는 그런 표준들의 확장본들에 따라 동작할 수도 있다. 다른 예들은 VP9, MPEG-2 및 ITU-T H.263을 포함할 수도 있다. 본 개시물의 기법들은, 그러나, 임의의 특정 코딩 표준 또는 기법으로 제한되지 않는다.

[0033] 비록 도 1에 도시되진 않았지만, 일부 양태들에서, 비디오 인코더 (20) 와 비디오 디코더 (30)는 각각이 오디오 인코더 및 디코더와 통합될 수도 있고, 적절한 멀티플렉싱 및 디멀티플렉싱 모듈들, 또는 다른 하드웨어 및/ 또는 소프트웨어를 포함하여, 공통 데이터 스트림 또는 개별 데이터 스트림들에서의 오디오 및 비디오 양쪽 모

두의 인코딩을 핸들링할 수도 있다. 적용 가능하다면, 일부 실시형태들에서, 멀티플렉싱 및 디멀티플렉싱 모듈들은 ITU H.223 멀티플렉서 프로토콜, 또는 사용자 데이터그램 프로토콜 (user datagram protocol, UDP)과 같은 다른 프로토콜들에 부합할 수도 있다.

[0034] 비디오 코더가, 본 개시물에서 설명된 바와 같이, 비디오 인코더, 비디오 디코더, 또는 둘 다를 지칭할 수도 있다. 비디오 인코더 (20) 와 비디오 디코더 (30) 각각은 다양한 적합한 코더 회로, 이를테면 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들 (digital signal processors, DSP들), 주문형 집적회로들 (application specific integrated circuits, ASIC들), 필드 프로그램가능 게이트 어레이들 (field programmable gate arrays, FPGA들), 개별 로직, 소프트웨어, 하드웨어, 펌웨어 또는 그것들의 임의의 조합들 중 임의의 것으로서 구현될 수도 있다. 그 기법들이 소프트웨어에서 적어도 부분적으로 구현되는 경우, 디바이스가 본 개시물의 기법들을 수행하기 위해, 적합한 비일시적 컴퓨터 관독가능 매체 내에 소프트웨어에 대한 명령들을 저장하고 하나 이상의 프로세서들을 사용하여 하드웨어에서 그 명령들을 실행할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 의 각각은 하나 이상의 인코더들 또는 디코더들 내에 포함될 수도 있고, 그것들 중 어느 하나는 결합형 인코더/디코더 (코덱) 의 일부로서 각각의 디바이스 내에 통합될 수도 있다.

[0035] 비디오 인코더 (20) 및 비디오 디코더 (30) 중 하나 또는 둘 다가 인트라 예측 프로세스를 개선하기 위해 본 개시물에서 설명되는 기법들 중 임의의 것 또는 모두를 구현할 수도 있다.

[0036] 도 2는 개시된 원리들에 따라 구현될 수도 있는 비디오 인코더 (20) 의 일 예를 예시하는 블록도를 도시한다. 비디오 인코더 (20) 는 개선된 인트라 예측을 위한 개시된 기법들 중 임의의 것 또는 모두를 수행하도록 구성될 수도 있다. 비디오 인코더 (20) 는 예시를 목적으로 HEVC의 맥락에서, 하지만 개선된 인트라 예측 프로세스로부터 이익을 얻을 수도 있는 다른 코딩 표준들 또는 방법들에 관한 본 개시물의 제한 없이, 설명될 것이다.

[0037] 비디오 인코더 (20) 는 비디오 프레임들 내의 코딩 유닛들의 인트라 코딩 및 인터 코딩을 수행할 수도 있다. 인트라 코딩은 공간적 예측에 의존하여, 주어진 비디오 프레임 내의 비디오에서의 공간적 리던던시를 감소시키거나 또는 제거한다. 인터 코딩은 시간적 예측에 의존하여, 비디오 시퀀스의 현재 프레임 및 이전에 코딩된 프레임들 사이의 시간적 리던던시를 감소시키거나 또는 제거한다. 이전에 코딩된 프레임들은 비디오 시퀀스에서의 (예컨대, 그룹 (group of pictures, GOP) 내의) 현재 프레임 전 및/또는 후일 수도 있다.

인트라 모드 (I-모드) 는 여러 공간 기반 비디오 압축 모드들 중 임의의 것을 지칭할 수도 있다. 단방향 예측 모드 (P 모드) 또는 양방향 예측 모드 (B 모드) 와 같은 인터 모드들은 여러 시간 기반 비디오 압축 모드들 중 임의의 것을 지칭할 수도 있다.

[0038] 도 2에 도시된 바와 같이, 비디오 인코더 (20) 는 인코딩될 비디오 프레임 내의 현재 비디오 블록을 수신한다. 도 2의 예에서, 비디오 인코더 (20) 는 모션 보상 모듈 (44), 모션 추정 모듈 (42), 인트라 예측 모듈 (46), 참조 프레임 버퍼 (64), 합산기 (50), 변환 모듈 (52), 양자화 모듈 (54), 및 엔트로피 인코딩 모듈 (56) 을 포함한다. 변환 모듈 (52) 은 변환 (예컨대, 이산 코사인 변환 (DCT)) 을 하나 이상의 변환 유닛들 (TU) 을 포함하는 잔차 데이터의 블록에 적용하는데, 각각의 TU는 복수의 변환 블록들 (예컨대, 하나의 루마 TB 및 두 개의 크로마 TB) 을 포함할 수도 있다. 비디오 블록 복원을 위해, 비디오 인코더 (20) 는 역 양자화 모듈 (58), 역 변환 모듈 (60), 및 합산기 (62) 를 또한 포함한다. 블록화제거 필터 (도시되지 않음) 가 블록 경계들을 필터링하여 복원된 비디오로부터 블록현상 (blockiness) 아티팩트들을 제거하기 위해 또한 포함될 수도 있다. 원한다면, 블록화제거 필터는 통상 합산기 (62) 의 출력을 필터링할 것이다. 추가적인 필터들 (인 루프 또는 포스트 루프) 이 블록화제거 필터에 부가하여 또한 사용될 수도 있다. 추가적인 필터들이 간략화를 위해 도시되지 않지만, 그것들은 비디오 인코더 (20) 내의 다양한 로케이션들에서, 이를테면 합산기 (50) 의 출력에서 루프 내 필터링을 위해 포함될 수도 있다.

[0039] 비록 예시의 편의를 위해 별개의 기능성 모듈들로서 도시되지만, 비디오 인코더 (20) 내의 컴포넌트들 중 임의의 컴포넌트의 구조 및 기능은 서로 고도로 통합될 수도 있다. 예를 들어, 일부 실시형태들에서, 모션 추정 모듈 (42) 은 모션 보상 모듈 (44) 과 고도로 통합될 수도 있다.

[0040] 인코딩 프로세스 동안, 비디오 인코더 (20) 는 코딩될 비디오 프레임 또는 슬라이스를 수신한다. 그 프레임 또는 슬라이스는 다수의 비디오 블록들, 예컨대, 최대 코딩 유닛들 (LCU들) 로 나누어질 수도 있다. 모션 추정 모듈 (42) 과 모션 보상 모듈 (44) 은 시간적 압축을 제공하기 위해 참조 프레임 버퍼 (64) 에 저장된 하나 이상의 참조 프레임들에서의 하나 이상의 블록들을 기준으로 수신된 비디오 블록의 인터 예측 코딩을 수행한다. 인트라 예측 모듈 (46) 은 공간적 압축을 제공하기 위해 코딩될 블록과 동일한 프레임 또는 슬라이스에

서의 하나 이상의 이웃하는 블록들을 기준으로 수신된 비디오 블록의 인트라 예측 코딩을 수행할 수도 있다. 인트라 예측 모듈 (46) 은, 이 문서에서 나중에 논의되는 바와 같이, 개시된 원리들에 따라 개선될 수도 있다.

[0041] 모드 선택 모듈 (40) 은 각각의 모드에 대한 레이트 왜곡 분석에 기초하여, 코딩 모드들 중 하나 (예컨대, 인트라 또는 인터) 를 선택할 수도 있다. 이 선택에 기초하여, 결과적인 인트라 코딩된 또는 인터 코딩된 블록은 잔차 블록을 나타내는 잔차 데이터를 생성하는 합산기 (50) 에 그리고 참조 프레임에서의 사용을 위한 인코딩된 블록을 복원하는 합산기 (62) 에 제공될 수도 있다. 모드 선택 모듈 (40) 은 신팩스 엘리먼트들, 이를 테면 인트라 예측 모드 표시자들, 모션 벡터들, 구획 정보, 및 다른 이러한 신팩스 정보를 엔트로피 인코딩 모듈 (56) 에게 또한 제공할 수도 있다. 일부 비디오 프레임들은 I-프레임들로서 지정될 수도 있는데, I-프레임에서의 모든 블록들은 인트라 예측 모드에서 인코딩된다. 일부 경우들에서, 인트라 예측 모듈 (46) 은, 예를 들어, 모션 추정 모듈 (42) 에 의해 수행된 모션 검색이 블록의 충분한 예측을 초래하지 않는 경우, P-프레임 또는 B-프레임에서의 블록의 인트라 예측 인코딩을 또한 수행할 수도 있다.

[0042] 위에서 언급된 바와 같이, 모션 추정 모듈 (42) 과 모션 보상 모듈 (44) 은 고도로 통합될 수도 있지만 개념상의 목적들을 위해 별개로 예시된다. 모션 추정은 비디오 블록들에 대한 모션을 추정하는, 모션 벡터들을 생성하는 프로세스이다. 모션 벡터가, 예를 들어, 참조 프레임에서의 참조 샘플을 기준으로 현재 프레임에서의 예측 유닛 (PU) 의 변위를 표시할 수도 있는데, PU가 복수의 예측 블록들 (예컨대, 하나의 루마 PB 및 두 개의 크로마 PB들) 을 포함할 수도 있다. 참조 샘플이 화소 차이 메트릭의 측면에서 코딩되고 있는 PU와 근접하여 일치한다는 것이 발견된 블록일 수도 있다. 화소 차이 메트릭은 차의 절대값 합 (sum of absolute difference, SAD), 차의 제곱 합 (sum of square difference, SSD), 또는 다른 차이 메트릭들을 수반할 수도 있다.

[0043] 모션 추정 모듈 (42) 은 인터 코딩된 프레임의 PU에 대한 모션 벡터를 그 PU와 참조 프레임 버퍼 (64) 에 저장된 참조 프레임의 참조 샘플들을 비교함으로써 계산한다. 일부 예들에서, 비디오 인코더 (20) 는 참조 프레임 버퍼 (64) 에 저장된 참조 프레임들의 부 정수 (sub-integer) 화소 위치들에 대한 값들을 계산할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 참조 프레임의 1/4 화소 위치들, 1/8 화소 위치들, 또는 다른 분수 (fractional) 화소 위치들의 값들을 계산할 수도 있다. 그러므로, 모션 추정 모듈 (42) 은 풀 (full) 및 분수 화소 위치들 양쪽 모두에 대하여 모션 검색을 수행하고 분수 화소 정밀도를 갖는 모션 벡터를 출력할 수도 있다. 일부 실시형태들에서, 모션 검색은 루마 PB들을 사용할 수도 있는데, 이는 모션을 충분히 나타낼 수도 있다. 모션 추정 모듈 (42) 은 계산된 모션 벡터를 엔트로피 인코딩 모듈 (56) 및 모션 보상 모듈 (44) 로 전송할 수도 있다.

[0044] 모션 보상 모듈 (44) 에 의해 수행되는 모션 보상은 모션 추정 모듈 (42) 에 의해 결정될 수도 있는 모션 벡터에 기초하여 PU에 대한 값들을 폐치하는 것 또는 생성하는 것을 수반할 수도 있다. 현재 비디오 블록의 PU에 대한 모션 벡터를 수신 시, 모션 보상 모듈 (44) 은 참조 프레임들 중 하나의 참조 프레임에서 모션 벡터가 가리키는 대응 블록을 찾을 수도 있다. 합산기 (50) 는, 아래에서 논의되는 바와 같이, 코딩되고 있는 현재 비디오 블록의 화소 값들을 감산하여, 화소 차이 값들을 형성함으로써, 잔차 블록을 나타내는 잔차 데이터를 형성한다. 대체로, 모션 추정 모듈 (42) 은 모션 벡터들을 계산하기 위해 루마 성분들에 대하여 모션 추정을 수행하고, 모션 보상 모듈 (44) 은 크로마 성분들 및 루마 성분들 양쪽 모두에 대해 모션 벡터들을 사용한다.

[0045] 인트라 예측 모듈 (46) 은 모션 추정 모듈 (42) 및 모션 보상 모듈 (44) 에 의해 수행된 인터 예측에 대한 대안으로서, 수신된 블록을 인트라 예측 인코딩할 수도 있다. 인트라 예측 모듈 (46) 은 블록에 대한 좌우, 상하 인코딩 순서를 가정하여, 이웃하는 이전에 코딩된 블록, 예컨대, 현재 블록의 상측의, 상측 및 우측의, 상측 및 좌측의, 좌측의, 또는 하측 및 좌측의 블록들에 대하여 수신된 블록 (예컨대, 변환 블록) 을 인코딩할 수도 있다. 인트라 예측 모듈 (46) 은 다양한 상이한 인트라 예측 모드들로 구성될 수도 있다. 예를 들어, 인트라 예측 모듈 (46) 은 인코딩되고 있는 코딩 유닛의 사이즈에 기초하여, 특정한 수의 방향성 예측 모드 (directional prediction mode) 들, 예컨대, 33 가지 방향성 예측 모드들로 구성될 수도 있다. 인트라 예측 모듈 (46) 은 다른 유형들의 예측 모드들, 예컨대, 평면 예측 모드 (planar prediction mode) 및 DC 예측 모드로 또한 구성될 수도 있다.

[0046] 인트라 예측 모듈 (46) 은, 예를 들어, 다양한 인트라 예측 모드들에 대한 에러 값들을 계산하고 최저 에러 값 또는 다른 메트릭에 의한 최고 스코어를 산출하는 모드를 선택함으로써 인트라 예측 모드를 선택할 수도 있다. 모드 선택들은 PU 레벨에서 이루어질 수도 있다. 방향성 예측 모드들은 공간적으로 이웃하는 화소들 (또

한 "이웃 화소들"이라고 지칭됨)의 값들을 조합하고 조합된 값들을 연관된 PU에서의 하나 이상의 화소 위치들에 적용하는 기능들을 포함할 수도 있다. 일단 PU에서의 모든 화소 위치들에 대한 값들이 계산되었다면, 인트라 예측 모듈 (46)은 연관된 PU 및 인코딩될 수신된 블록 간의 화소 차이들에 기초하여 예측 모드에 대한 에러 값을 계산할 수도 있다. 일부 실시형태들에서, 인트라 예측 모듈은 어떤 모드가 최저 에러 값 또는 다른 메트릭에 의한 최고 레이팅을 산출하는지를 결정하기 위해 모든 인트라 예측 모드를 테스트할 수도 있다. 일부 실시형태들에서, 인트라 예측 모듈 (46)은 허용가능 에러 값 또는 다른 메트릭에 의한 충분한 레이팅을 산출하는 인트라 예측 모드가 결정되기까지 인트라 예측 모드들을 테스트하는 것을 계속할 수도 있다. 가용 인트라 예측 모드들의 서브셋트를 테스트하는 것은 요구된 프로세싱 레이턴시 및 프로세싱 대역폭을 감소시킬 수도 있다.

[0047]

인트라 예측 모듈 (46)은 다양한 인트라 예측 모드들을 테스팅하는 경우 레이트 왜곡 분석을 대안적으로 또는 추가적으로 사용할 수도 있고, 각각의 PU에 대해 (테스트된 모드들 중에서) 최상의 레이트 왜곡 특성들을 갖는 인트라 예측 모드를 선택할 수도 있다. 레이트 왜곡 분석은 인코딩된 블록 및 대응하는 원래의, 비인코딩된 블록 간의 왜곡 (예컨대, 에러)의 양, 뿐만 아니라 인코딩된 블록을 생성하는데 필요한 비트들의 수 (및 결과적인 비트레이트)를 일반적으로 결정한다. 인트라 예측 모듈 (46)은 어떤 인트라 예측 모드가 그 블록에 대한 최상의 레이트 왜곡 값을 나타내는지를 결정하기 위해 다양한 인코딩된 블록들에 대한 왜곡들 및 레이트들로부터 비율들을 계산할 수도 있다. 인트라 예측 모드를 선택한 후, 인트라 예측 모듈 (46)은 선택된 인트라 예측 모드에 연관된 PU를 합산기 (50)에게 전송할 수도 있다. 선택된 인트라 예측 모드의 표시자가 비디오 인코더 (20)에 의해 출력되는 인코딩된 비디오 데이터에 포함되기 위해 엔트로피 인코딩 모듈 (56)에게 전송될 수도 있다.

[0048]

모드 선택 페이즈 동안, 인트라 예측 모듈 (46)은 참조 프레임 버퍼 (64)에 저장된 정보를 사용하여 잔차 블록들의 값들을 인트라 예측할 수도 있다. 인트라 예측 계산은 블록 레벨에서 (예컨대, 하나의 변환 유닛 (TU) 또는 변환 블록 (TB) 씩 따로따로) 일어날 수도 있는데, 일부 TU들은 인트라 예측 모드가 컴퓨팅되는 대응하는 코딩 유닛들 (CU) 및 예측 유닛들 (PU) 보다 더 작을 수도 있다. 더욱이, CU 내의 하나의 TU의 인트라 예측은 CU 내에 또한 있는 이웃하는 TU들의 인트라 예측 및 복원에 따라 달라질 수도 있다. 개시된 원리들에 따라, 인트라 예측 엔진이 연속적인 블록들 (예컨대, TU들)의 개선된 파이프라인을 인트라 예측 동안 허용할 수도 있다. 이들 원리들은 이 문서에서 나중에 논의된다.

[0049]

비디오 인코더 (20)는 코딩되고 있는 원래의 비디오 블록으로부터 모션 보상 모듈 (44) 또는 인트라 예측 모듈 (46)에 의해 계산된 예측 데이터를 감산함으로써 잔차 블록을 나타내는 잔차 데이터를 결정한다. 합산기 (50)는 이 감산 연산을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다. 잔차 블록은 화소 차이 값들의 2 차원 매트릭스에 대응할 수도 있다. 잔차 블록에서의 값들은 인트라 예측된 또는 인터 예측된 블록 내의 그리고 코딩될 원래의 블록 내의 병치된 (co-located) 화소들의 값들 간의 차이들 (예컨대, 에러)에 대응할 수도 있다. 그 차이들은 코딩되는 블록의 유형에 의존하여 크로마 또는 루마 차이들일 수도 있다.

[0050]

인트라 예측 동안, 부분적으로 인트라 예측된 CU 또는 PU의 블록 사이즈로 된 (예컨대, TU 사이즈로 된) 부분들은 합산기 (50), 변환 모듈 (52), 양자화 모듈 (54), 역 양자화 모듈 (58), 역 변환 모듈 (60), 합산기 (62), 및 참조 프레임 버퍼 (64)를 통해 전파할 수도 있다. 따라서, 제 2 블록 (예컨대, TU)의 인트라 예측은, 심지어 제 1 및 제 2 블록들이 공통 CU 또는 PU에 연관되더라도, 이미 인코딩되어 있는 제 1 블록에 연관된 참조 값들을 이용할 수도 있다.

[0051]

변환 모듈 (52)은 잔차 데이터로부터 TU들을 형성할 수도 있는데, 각각의 TU는 복수의 변환 블록들 (예컨대, 하나의 루마 TB 및 두 개의 크로마 TB들)을 포함할 수도 있다. 변환 모듈 (52)은 이미지 변환 (예컨대, 2 차원 변환), 이를테면 이산 코사인 변환 (DCT), 이산 사인 변환 (discrete sine transform, DST), 이산 푸리에 변환 (discrete Fourier transform, DFT), 아다마르 (Hadamard) 변환, 웨이브릿 변환, 정수 변환, 서브대역 변환, 또는 다른 유형의 변환을 TU에 적용하여, 변환 계수들을 포함하는 비디오 블록을 형성한다. TU에 연관된 복수의 TB들은 동시에 또는 연속하여 컴퓨팅될 수도 있다.

[0052]

변환 모듈 (52)은 결과적인 변환 계수들을 양자화 모듈 (54)로 전송할 수도 있으며, 양자화 모듈은 그 다음에 변환 계수들을 양자화할 수도 있다. 양자화는, 변환 계수들이 이산 값들의 감소된 세트로 변환되는 프로세스를 일반적으로 지칭하는데, 이는 정밀도를 대가로 증가된 효율 (예컨대, 더 높은 압축)을 초래한다. 양자화 레벨은 적응적으로 결정될 수도 있어서, 더 낮은 에너지 계수들이 더 낮은 에너지 계수들보다 적게 양자화 된다. 게다가, 더 낮은 에너지 계수들의 일부는 양자화 모듈 (54)에 의해 양자화된 후에 "0"의 값으로 설

정될 수도 있다. 이는 아래에서 설명되는 바와 같이, 엔트로피 인코딩에서 개선된 압축을 허용할 수도 있다.

[0053] 양자화를 뒤이어, 비디오 인코더 (20)는 변환 계수들을 스캔하여, 양자화된 변환 계수들을 갖는 2차원 매트릭스로부터 1차원 어레이를 생성할 수도 있다. 스캔은 상대적으로 더 높은 에너지를 종종 갖는 더 낮은 주파수 계수들을 어레이의 전면에 더 가까이 배치하고 더 높은 주파수 계수들을 어레이의 뒷면에 더 가까이 배치하도록 설계될 수도 있다. 일부 예들에서, 비디오 인코더 (20)는 미리 정의된 스캔 순서 (예컨대, 지그재그 스캔 패턴)를 사용하여 변환 계수들을 스캔할 수도 있다. 다른 예들에서, 비디오 인코더 (20)는 적응적 스캔을 수행할 수도 있고, 선택된 스캔 순서(들)는 데이터 스트림으로 인코딩될 수도 있다. 스캐닝은, 예컨대, 엔트로피 인코딩 모듈 (56)에 의해 수행될 수도 있다.

[0054] 일단 변환 계수들이 스캔되면, 엔트로피 인코딩 모듈 (56)은 콘텍스트 적응 가변 길이 코딩 (context-adaptive variable-length coding, CAVLC), 확률 간격 파티셔닝 엔트로피 (probability interval partitioning entropy, PIPE) 코딩, 또는 콘텍스트 기반 적응 이진 산술 코딩 (context-based adaptive binary arithmetic coding, CABAC)과 같은 엔트로피 코딩을 계수들에 적용할 수도 있다. 덧붙여서, 엔트로피 인코딩 모듈 (56)은 모션 벡터 (MV) 정보와 비디오 디코더에서 비디오 데이터를 디코딩에 있어서 유용한 다양한 선택스 엘리먼트들 중 임의의 선택스 엘리먼트를 인코딩할 수도 있다. 이들 선택스 엘리먼트들은 인코딩된 비디오 데이터를 복원하는 경우 비디오 디코더에 의해 사용될 수도 있다. 엔트로피 코딩 모듈 (56)에 의한 엔트로피 코딩에 뒤이어, 결과적인 인코딩된 비디오는 다른 디바이스, 이를테면 비디오 디코더로 송신되거나, 또는 나중의 송신 또는 취출을 위해 보관될 수도 있다.

[0055] 역 양자화 모듈 (58)과 역 변환 모듈 (60)은 각각 역 양자화 및 역 변환을 적용하여, 예컨대, 참조 프레임 버퍼 (64)에 저장되는 참조 프레임을 위한 참조 블록을 생성하기 위해 화소 도메인에서의 잔차 블록을 복원한다.

모듈들 (58, 60)은 비디오 디코더의 부분들을 시뮬레이션하는데 사용될 수도 있고, 그에 따라 비디오 디코더에 존재하는 대응하는 모듈들과 실질적으로 유사할 수도 있다. 모션 보상 모듈 (44)은, 합산기 (62)와 연계하여, 잔차 블록을 참조 프레임 버퍼 (64)의 프레임들 중 하나의 프레임의 예측 블록에 가산함으로써 참조 블록을 계산할 수도 있다. 모션 보상 모듈 (44)은 하나 이상의 보간 필터들을 복원된 잔차 블록에 또한 적용하여 모션 추정에서의 사용을 위한 부정수 화소 값들을 계산할 수도 있다. 합산기 (62)는 복원된 잔차 블록을 모션 보상 모듈 (44)에 의해 생성된 모션 보상된 예측 블록에 가산하여, 참조 프레임 버퍼 (64)에서의 저장을 위한 복원된 비디오 블록을 생성할 수도 있다. 참조 프레임 버퍼 (64)는 디코더에서 수신되는 것과 실질적으로 동일한 비디오 정보를 유지할 수도 있으며, 이는 인트라 예측 모드들 및 모션 벡터들이 적용되는 시간 동안, 비디오 인코더 (20)가 디코더에 의해 보이는 이 공통 "상태" 정보에 기초하여 인트라 예측 모드들 및 모션 벡터들을 정확히 인코딩하는 것을 허용할 수도 있다.

[0056] 참조 프레임 버퍼 (64)에 저장되는 복원된 비디오 블록은 다른 비디오 프레임에서의 블록을 인터 코딩하기 위한 참조 블록으로서 모션 추정 모듈 (42) 및 모션 보상 모듈 (44)에 의해 사용될 수도 있다. 덧붙여, 또는 대안적으로, 복원된 비디오 블록은 동일한 비디오 프레임에서의 블록을 인트라 코딩하기 위해 인트라 예측 모듈 (46)에 의해 사용될 수도 있다. 일부 시나리오들에서, 특정 참조 블록이 인트라 예측 또는 인터 예측 중 어느 하나를 위해 사용되지 않을 수도 있지만, 단지, 다른 유용한 참조 블록들을 갖는 참조 프레임의 일부일 수도 있다.

[0057] 도 3은 개시된 원리들에 따라 구현될 수도 있는 비디오 디코더 (30)의 일 예를 예시하는 블록도를 도시한다. 비디오 디코더 (30)는 본 개시물의 기법들 중 임의의 것 또는 모두를 수행하도록 구성될 수도 있다.

[0058] 도 3의 예에서, 비디오 디코더 (30)는 엔트로피 디코딩 모듈 (70), 모션 보상 모듈 (72), 인트라 예측 모듈 (74), 역 양자화 모듈 (76), 역 변환 모듈 (78), 참조 픽처 버퍼 (82) 및 합산기 (80)를 포함한다. 비디오 디코더 (30)는, 일부 실시형태들에서, 도 2에서 설명된 바와 같은 비디오 인코더 (20)에 관해 설명된 인코딩 과정에 일반적으로 역인 디코딩 과정을 수행할 수도 있다.

[0059] 비록 예시의 편의를 위해 별개의 기능성 모듈들로서 도시되지만, 비디오 디코더 (30) 내의 컴포넌트들 중 임의의 컴포넌트의 구조 및 기능은 서로 고도로 통합될 수도 있다. 예를 들어, 일부 실시형태들에서, 엔트로피 디코딩 모듈 (70)은 역 양자화 모듈 (76)과 고도로 통합될 수도 있다.

[0060] 디코딩 프로세스 동안, 비디오 디코더 (30)는, 비디오 인코더에 의해 생성된 바와 같은, 인코딩된 비디오 슬라이스의 비디오 블록들 및 연관된 선택스 엘리먼트들을 나타내는 인코딩된 비디오 비트스트림을 수신한다.

그 비트스트림은, 도 1에 도시된 바와 같이, 파일 서버로부터, 저장 매체로부터, 및 모뎀을 통해 비디오 인코더로부터를 포함하는, 다양한 소스들로부터 수신될 수도 있다.

[0061] 엔트로피 디코딩 모듈 (70)은 비트스트림을 생성했던 비디오 인코더의 엔트로피 인코딩 모듈에 의해 사용된 스캔 순서를 미러링하는 스캔 (예컨대, 역 지그재그 스캔 패턴)을 사용하여, 수신된 값들을 스캔할 수도 있다.

덧붙여, 또는 대안으로, 스캔 순서는 예측 모드, 블록 사이즈, 변환, 또는 다른 특성들과 같은 코딩된 비디오의 특성들에 기초하여 비디오 디코더 (30)에 의해 유추될 수도 있다. 스캔 순서가 어떻게 결정되더라도, 엔트로피 디코딩 모듈 (70)은 수신된 1차원 어레이를 2차원 매트릭스로 스캔하기 위해 스캔 순서의 역을 사용할 수도 있다. 엔트로피 디코딩 모듈 (70)은 그 다음에 인코딩된 비디오를 엔트로피 디코딩할 수도 있는데, 그 엔트로피 디코딩 모듈은 도 2의 엔트로피 인코딩 모듈 (56)에 의해 사용되는 것의 역인 프로세스를 수반할 수도 있다.

[0062] 비디오 디코더 (30)는 비디오 슬라이스 레벨 및/또는 비디오 블록 레벨에서 신택스 엘리먼트들을 수신할 수도 있다. 엔트로피 디코딩 모듈 (70)은 양자화된 계수들, 모션 벡터들, 인트라 예측 모드 표시자들, 및 다른 신택스 엘리먼트들을 추출하기 위해 비트스트림을 추가로 분석할 수도 있다. 모션 벡터들은 모션 보상 모듈 (72)에 의해 수신될 수도 있고 인트라 예측 모드 표시자들은 인트라 예측 모듈 (74)에 의해 수신될 수도 있다.

[0063] 슬라이스가 인트라 코딩식 (I) 슬라이스로서 코딩되는 경우, 인트라 예측 모듈 (74)은 현재 비디오 슬라이스의 비디오 블록에 대한 예측 데이터를 현재 프레임 또는 꾹처의 이전에 디코딩된 블록들로부터의 인트라 예측 모드 표시자 및 데이터에 기초하여 생성할 수도 있다. 이 프로세스는 도 4와 도 5 및 수반되는 설명들에서 더욱 상세히 다루어진다.

[0064] 비디오 프레임이 인터 코딩된 (예컨대, B, P, 또는 GPB) 슬라이스로서 코딩되는 경우, 모션 보상 모듈 (72)은 모션 벡터들 및 다른 신택스 정보에 기초하여, 현재 비디오 슬라이스의 비디오 블록에 대한 예측 유닛들 (PU)을 생성할 수도 있다. 그 PU들은 참조 꾹처 리스트들 중 하나의 참조 꾹처 리스트 내의 참조 꾹처들 중 하나의 참조 꾹처로부터 생성될 수도 있다. 비디오 디코더 (30)는 참조 프레임 버퍼 (82)에 저장된 참조 꾹처들에 기초하여 디폴트 구축 기법들을 사용하여, 참조 프레임 리스트들 (예컨대, HEVC 표준에 의해 정의된 바와 같은 List 0 및 List 1)을 구축할 수도 있다. 모션 보상 모듈 (72)은 모션 벡터들 및 다른 신택스 정보를 파악함으로써 비디오 블록에 대한 예측 정보를 결정할 수도 있다. 이 예측 정보는 디코딩되고 있는 현재 비디오 블록에 대한 PU들을 생성하는데 사용될 수도 있다. 예를 들어, 모션 보상 모듈 (72)은 예측 모드 (예컨대, 인트라 또는 인터), 인터 예측 슬라이스 유형 (예컨대, B 슬라이스, P 슬라이스, 또는 GPB 슬라이스), 그 슬라이스에 대한 하나 이상의 참조 꾹처 리스트들에 대한 구축 정보, 그 슬라이스의 각각의 인터 인코딩된 비디오 블록에 대한 모션 벡터들, 그 슬라이스의 각각의 인터 코딩된 비디오 블록에 대한 인터 예측 스테이터스, 및 현재 비디오 슬라이스에서의 비디오 블록들을 디코딩하기 위한 다른 정보를 결정할 수도 있다.

[0065] 모션 보상 모듈 (72)은 보간 필터들에 기초하여 보간을 또한 수행할 수도 있다. 모션 보상 모듈 (72)은 비디오 블록들을 인코딩했던 비디오 인코더에 의해 사용된 것들에 유사한 보간 필터들을 사용하여, 참조 블록들의 부 정수 화소들에 대한 보간된 값들을 계산할 수도 있다. 일부 실시형태들에서, 모션 보상 모듈 (72)은 수신된 신택스 엘리먼트들로부터 비트스트림을 생성했던 비디오 인코더에 의해 사용된 보간 필터들을 결정하고 그 보간 필터들을 사용하여 예측 블록들을 생성할 수도 있다.

[0066] 역 양자화 모듈 (76)은 비트스트림에서 제공되고 엔트로피 디코딩 모듈 (70)에 의해 디코딩된 양자화된 변환 계수들을 역 양자화 (예컨대, 탈양자화 (dequantize)) 한다. 역 양자화 프로세스는 양자화 정도와, 마찬가지로, 원래의 변환 계수들을 결정하기 위해 적용될 수도 있는 역 양자화 정도를 결정하기 위해 비디오 슬라이스에서의 각각의 비디오 블록에 대해 비디오 디코더 (30)에 의해 계산된 양자화 파라미터 (QP_Y)의 사용을 포함할 수도 있지만, 일부 양자화 에러는 존재할 수도 있다.

[0067] 역 변환 모듈 (78)은 화소 또는 화소 차이 도메인에서의 잔차 데이터를 형성하기 위해 역 변환 (예컨대, 역 DCT)을 적용할 수도 있다. 역 변환은 도 2의 변환 모듈 (52)에 의해 사용된 것의 역일 수도 있다. 잔차 데이터는 변환 유닛들 (TU)과, 따라서 디코딩되고 있는 변환 블록들 (TB)과 동일한 사이즈를 갖는 잔차 블록들에서 컴퓨팅될 수도 있다.

[0068] 모션 보상 모듈 (72) 및/또는 인트라 예측 모듈 (74)이 코딩 유닛 (CU)의 블록 사이즈로 된 (예컨대, TU 사이즈로 된) 부분을 생성한 후, 비디오 디코더 (30)는 역 변환 모듈 (78)로부터의 잔차 블록과 모션 보상 모듈

(72) 및/또는 인트라 예측 모듈 (74)에 의해 생성된 대응하는 블록 사이즈로 된 예측 데이터를 합산함으로써 디코딩된 비디오 블록을 형성한다. 합산기 (80)는, 이 합산 동작을 수행하고 참조 프레임 버퍼 (82)에서의 결과들을 저장하는 컴포넌트 또는 컴포넌트들을 나타낸다.

[0069] 인트라 예측 동안, 합산기 (80)는 각각의 역 변환된 블록과 참조 프레임 버퍼 (82)로부터의 대응하는 블록을 반복적으로 합산할 수도 있는데, 대응하는 블록은 역 변환된 블록과 동일한 사이즈일 수도 있고, 적어도 부분적으로는, 인트라 예측 모드 표시자에 기초하여 선택될 수도 있다. 따라서, 참조 프레임 버퍼에 기입되는 디코딩된 블록들은 TU들과 대응할 수도 있고 TU들과 동일한 사이즈를 가질 수도 있다. TU 사이즈로 된 디코딩된 블록들을 반복적으로 복원하고 참조 프레임 버퍼 (82)에 저장함으로써, 완전한, 디코딩된 CU가 참조 프레임 버퍼 (82)에서 결정될 수도 있다.

[0070] 본 개시물의 하나의 목적은 TU 및 CU 사이즈들이 상이하여서, 인트라 예측 프로세싱이 추가로 병렬화되는 시나리오들을 최적화하는 것이다.

[0071] 원한다면, 블록화제거 필터가 블록현상 아티팩트들을 제거하기 위하여 디코딩된 블록들을 필터링하는데 적용될 수도 있다. 다른 루프 필터들 (예컨대, 코딩 루프 내 또는 코딩 루프 뒤에 있음)이 화소 전환 (transition)들을 부드럽게 하기 위해, 또는 그렇지 않으면 비디오 품질을 개선하기 위해 또한 사용될 수도 있다. 주어진 프레임 또는 퍽처에서의 디코딩된 비디오 블록들은 그 다음에 참조 프레임 버퍼 (82)에 저장된다. 참조 프레임 버퍼 (82)는 후속 모션 보상 및/또는 인트라 예측을 위해 사용될 수도 있는 이 및 다른 참조 프레임들 또는 퍽처들을 저장할 수도 있다. 참조 프레임 버퍼 (82)는 디스플레이 디바이스 상의 프레젠테이션을 위해 디코딩된 비디오를 또한 저장할 수도 있다.

[0072] 도 4는 인트라 예측을 위한 다양한 모드들을 예시하는 개략도를 도시한다. 구체적으로는, 도 4는 HEVC 코더들을 위해 사용될 수도 있는 서른다섯 개 인트라 예측 모드들을 예시한다. 도 4는 예시를 목적으로 HEVC의 맥락에 있지만, 개선된 인트라 예측 프로세스로부터 이익을 얻을 수도 있는 다른 코딩 표준들 또는 방법들에 관한 본 개시물의 제한은 없다.

[0073] 이미지들 및 비디오 프레임들은 공간적으로 용장성인 데이터를 일반적으로 포함한다. 공간적 리던던시는 주어진 화소가 주위의 화소들 중 적어도 하나의 화소에 상관될 수도 있다는 것을 의미한다. 인트라 예측이 블록 내의 화소들이 이웃 화소들에 기초하여 보간되는 것을 허용한다. 선택된 이웃 화소들은 블록 내의 의존성들을 제거하기 위해서 이미 계산된 다른 블록들에 위치됨으로써, 그 블록 내의 다수의 화소들의 병렬 인트라 예측을 촉진할 수도 있다. 예를 들어, 방향성 모드를 사용하는 경우, 이웃 화소들은 계산되고 있는 블록 외부에 있는, 모드 정의된 방향에서의 가장 가까운 화소들로서 선택될 수도 있다. 다른 블록들로부터의 화소들을 사용하는 이 개념은 아래의 도 5에서 추가로 예시된다.

[0074] 도 4에 도시된 바와 같이, 인트라 예측 값이 인트라 예측 모드에 기초하여 화소 (401)에 대해 계산될 수도 있다. 인트라 예측 모드는 비디오 인코더에서 예측 유닛별 기반으로 컴퓨팅될 수도 있다. 예측 유닛이 대응하는 코딩 유닛 및 그것의 연관된 코딩 블록들과 동일한 사이즈인 경우, 코딩 블록들 내의 각각의 변환 블록은 공통 인트라 예측 모드를 가질 수도 있다. 인트라 예측 모드 표시자들이 다른 비디오 데이터와 함께 비트스트림 속에 인코딩될 수도 있어서, 모드 표시자들은 디코딩 프로세스 동안 디코더에 의해 사용될 수도 있다. 일부 코딩 유닛들 (예컨대, 인터 코딩되는 것들)이 대응하는 인트라 예측 모드 표시자들을 갖지 않을 수도 있다.

[0075] 모드 0은 화소 (401)가 HEVC 표준에 의해 정의된 평면 함수에 따라 계산되는 평면 코딩 (planar coding) 기법을 위해 사용될 수도 있다. 모드 1은 화소 (401)가 이웃 화소들의 평균으로서 계산되는 DC 코딩 기법을 위해 사용될 수도 있다. 모드 2 내지 모드 34는 방향성 코딩 (directional coding) 기법들을 위해 사용될 수도 있다. 여기서, 모드들 (2~34)에 연관된 화살표들은 화소 (401)의 값을 결정하기 위한 이웃 화소들을 찾는 검색 방향을 표시한다. 화살표가 두 개의 이웃 화소들 사이를 가리키는 상황들에서, 가중 보간 함수가 두 개의 이웃 화소들의 가중된 평균일 수도 있는 중간 값을 결정하기 위해 사용될 수도 있다.

[0076] HEVC에서, 인트라 예측 스킁은 블록들을 좌에서 우로 그리고 그 다음에 상단에서 하단으로 (예컨대, 영어 책을 읽는 경우와 같음), 또는 상단에서 하단으로 그리고 그 다음에 좌에서 우로 (예컨대, 일부 전통적인 동아시아 스크립트들을 읽는 경우와 같음) 계산하는 것을 수반하도록 가정될 수도 있다. 본 기술분야의 통상의 기술자에 의해 이해될 바와 같이, 블록 계산 순서를 회전시키거나 또는 그렇지 않으면 변경하는 것이 가능할 것이다 (예컨대, 그래서 블록들이 우에서 좌로, 하단에서 상단으로 인트라 예측된다). 그들 시나리오들에서, 방향

성 인트라 예측 모드들은 이미 계산된 블록들의 사용을 더 잘 촉진하기 위해 변경될 (예컨대, 회전될) 수도 있다.

[0077] 선택된 모드에 의존하여, 블록에서의 화소들에 대한 인트라 예측이 하나 이상의 다른 블록들에서의 이웃 화소들에 따라 달라질 수도 있다. 예를 들어, 인트라 예측 모드 26이 선택되는 경우, 화소 (401)를 포함하는 블록의 인트라 예측은 그 블록 상측의 이웃 블록에서부터의 하단 행에서의 화소에 의존할 수도 있다. 다른 예에서, 인트라 예측 모드 10이 선택되는 경우, 화소 (401)를 포함하는 블록의 인트라 예측은 그 블록 우측의 이웃 블록의 최우측 열에서의 화소에 의존할 수도 있다. 이들 의존성들의 결과로서, 인트라 예측은 이웃하는 블록들에 별도로 임의적으로 (arbitrarily) 적용되지 않을 수도 있다. 화살표들에 의해 도시된 바와 같이, 이웃 화소들을 제공하는 다른 블록들은 인트라 예측이 수행되는 현재 블록의 상측, 우상측, 좌상측, 좌측, 및 좌하측에 있을 수도 있다.

[0078] 도 5는 이웃하는 블록들에 대한 블록 (511)의 의존성들을 예시하는 개략도를 도시한다. 각각의 블록은, 예를 들어, 인트라 예측될 공통 예측 유닛 내에 있는 변환 유닛을 나타낼 수도 있다. 그 블록들은 도면에서 더 두꺼운 경계선들에 의해 구분된다. 각각의 블록은 4x4 개 화소들의 사이즈를 가질 수도 있다. 예를 들어, 블록 (511)은 "a" 내지 "p"로 라벨표시된 16 개 화소들을 포함할 수도 있다.

[0079] 블록들 (500, 501, 502, 510 및 520)은 완전하게 도시되지 않는다. 블록 (501)은 그것의 바닥 행에서 화소들 ("A", "B", "C", 및 "D")을 포함하며; 블록 (502)은 그것의 바닥 행에서 화소들 ("E", "F", "G", 및 "H")을 포함하며; 블록 (510)은 그것의 최우측 열에서 화소들 ("I", "J", "K", 및 "L")을 포함하며; 블록 (520)은 그것의 최우측 열에서 화소들 ("M", "N", "O", 및 "P")을 포함하고; 블록 (500)은 그것의 우하단 모서리에서 화소 "R"을 포함한다.

[0080] 위에서 설명된 바와 같이, HEVC와 같은 일부 비디오 코딩 표준들은 블록 (511) 내의 화소들의 인트라 예측이 이웃하는 블록들 (500, 501, 502, 510 및 520) 중 적어도 하나의 이웃하는 블록 내의 이웃 화소들의 복원된 값들 중 적어도 일부의 복원된 값들에 의존한다는 것을 말한다. 이 의존성 성질은 인트라 예측 모드에 의해 특징화된다. 예를 들어, 수직 인트라 예측 모드 (예컨대, HEVC에서의 모드 26)가 선택된다면, 블록 (511)에서의 화소들 "a" 내지 "p"의 인트라 예측은 블록 (501)에서의 화소들 "A" 내지 "D"의 복원된 값들에 의존할 수도 있다. 복원된 화소 데이터는 블록 (501)의 인트라 예측이 완료된 후, 또는, 최소한, 화소들 "A" 내지 "D"의 인트라 예측이 완료된 후에 이용 가능하게 될 수도 있다. 따라서, 블록 (511)의 인트라 예측은, 특정한 인트라 예측 모드들에 대해, 블록 (501)의 인트라 예측이 적어도 부분적으로 완료된 후에 시작하는 것으로 제한될 수도 있다. 다른 예에서, 수평 인트라 예측 모드 (예컨대, HEVC에서의 모드 10)가 선택된다면, 블록 (511)의 인트라 예측은 블록 (510)에서의 화소들 "I" 내지 "L"의 복원된 값들에 의존할 수도 있다. 다른 예로서, HEVC에서의 모드 2와 같은 대각선 인트라 예측 모드가 선택된다면, 블록 (511)의 인트라 예측은 블록들 (510 및 520)에 위치되는 화소들 "I" 내지 "P"에 기초한 계산들을 수반할 수도 있다. HEVC에서의 모드 0 (DC 모드)와 같은 일부 인트라 예측 모드들은 심지어 블록들 (500, 501, 502, 510 및 520)의 각각으로부터의 값들을 (이용 가능하다면) 이용할 수도 있다.

[0081] 파일라인식 아키텍처에서, 모든 가용 인트라 예측 모드들은 지원될 것이 필요할 수도 있고, 그래서 블록들 (500, 501, 502, 510 및 520)의 적어도 부분적 복원은 511의 인트라 예측이 시작되기 전에 유익할 수도 있다. 아래에서 설명될 바와 같이, 코딩 블록이 이웃하는 블록들 중 하나 (예컨대, 블록 (520))에의 의존성을 제거하기 위해서 인트라 예측 모드에 기초하여 선택적으로 회전되거나 또는 전치 (transpose) 될 수도 있다.

[0082] 도 6은 인트라 예측 서브시스템 (600)을 예시하는 블록도를 도시한다. 서브시스템 (600)은 인트라 예측 모듈 (610), 역 변환 모듈 (620), 참조 프레임 버퍼 (630), 및 합산기 (640)를 포함한다. 서브시스템 (600)은 비디오 인코더 및 비디오 디코더들에 통합될 수도 있다. 예를 들어, 서브시스템 (600)이 도 2의 비디오 인코더 (20)에 통합된다면, 인트라 예측 모듈 (610)은 도 2의 인트라 예측 모듈 (46)을 나타낼 수도 있으며, 역 변환 모듈 (620)은 도 2의 역 변환 모듈 (60)을 나타낼 수도 있으며, 참조 프레임 버퍼 (630)는 도 2의 참조 프레임 버퍼 (64)를 나타낼 수도 있고, 합산기 (640)는 도 2의 합산기 (62)를 나타낼 수도 있다.

[0083] 유사하게, 서브시스템 (600)이 도 3의 비디오 디코더 (30)에 통합된다면, 인트라 예측 모듈 (610)은 도 3의 인트라 예측 모듈 (74)을 나타낼 수도 있으며, 역 변환 모듈 (620)은 도 3의 역 변환 모듈 (78)을 나타낼 수도 있으며, 참조 프레임 버퍼 (630)는 도 3의 참조 프레임 버퍼 (82)를 나타낼 수도 있고, 합산기 (640)는 도 3의 합산기 (80)를 나타낼 수도 있다. 서브시스템 (600)의 특정한 컴포넌트들 (예컨대, 참조 프레임

버퍼 (630)) 이 다른 서브시스템들에서 추가적으로 사용될 수도 있다.

[0084] 서브시스템 (600) 은, 심지어 주어진 블록의 인트라 예측이 인트라 예측 순서에서 주어진 블록에 바로 실행하는 블록에 적어도 부분적으로 의존할 수도 있더라도, 연속적인 블록들의 인트라 예측을 최적화하도록 설계될 수도 있다. 예를 들어, 각각의 블록의 인트라 예측은, 도 7에서 더 상세히 도시될 바와 같이, 중대한 행, 중대한 열 중 어느 하나에 대해서는 복원된 화소들에 의존할 수도 있거나, 또는 실행하는 블록에서의 어느 것에도 의존하지 않을 수도 있다.

[0085] 도 6으로 되돌아가면, 인트라 예측 모듈 (610) 은 이웃 버퍼 (612), 이웃 프로세싱 부 (614), 및 인트라 예측 프로세싱 부 (616) 를 포함할 수도 있다. 이웃 버퍼 (612) 는 이웃 RAM (613) 과 인터페이싱하는 로직 및/ 또는 회로를 사용하여 이웃 화소들을 저장 및 취출할 수도 있다. 이웃 화소들은 블록들을 인트라 예측할 경우 사용될 수도 있는 화소들을 포함할 수도 있다. HEVC의 맥락에서, 블록이 동일하거나 또는 더 큰 사이즈를 갖는 코딩 유닛 (및 예측 유닛) 에 연관될 수도 있는 변환 유닛을 나타낼 수도 있다. 화소들은 인트라 예측되고 있는 블록의 유형에 의존하여, 크로마 및/또는 루마 값들을 포함할 수도 있다. 여기서, 블록 사이즈는 M이 음이 아닌 정수 값인 MxM일 수도 있다. M에 대한 예시적인 값들은 4, 8, 16, 및 32를 포함한다.

[0086] 이웃 버퍼 (612) 에 저장된 화소들은, 그 화소들이 인트라 예측 프로세싱 부 (616) 에 의해 수신되기 전에, 다른 방도들로 필터링, 패딩, 및 잠재적으로 컨디셔닝될 것이 필요할 수도 있다. 이를 컨디셔닝 액션들은 인트라 예측 프로세싱 부 (616) 에 의한 프로세싱을 대비하여, 이웃 버퍼 (612) 로부터의 화소들을 수신하는 이웃 프로세싱 부 (614) 에 의해 적어도 부분적으로 수행될 수도 있다. 예를 들어, 이웃 프로세싱 부 (614) 는 MxM 블록을 프로세싱하는 인트라 예측 프로세싱 부 (616) 를 위해 4M + 1 개의 이웃 화소들을 준비할 수도 있다. 이를 이웃 화소들은, 예를 들어, 도 5에 도시된 바와 같이, 이웃 화소를 "A" 내지 "R"을 나타낼 수도 있다.

[0087] 도 6으로 되돌아가서, 인트라 예측 프로세싱 부 (616) 는 프로세싱 엔진들 (617-1 내지 617-N) (또한 PE 1 내지 PE N으로서 라벨표시됨) 을 포함할 수도 있는데, 그 프로세싱 엔진들은 비디오 인코딩 및/또는 비디오 디코딩 디바이스에서의 하드웨어로서 구현될 수도 있다. N 개의 프로세싱 엔진들 (617) 이 프로세싱되고 있는 블록 내의 상이한 화소들을 인트라 예측하기 위해 이웃 프로세싱 부 (614) 로부터 수신된 입력들의 공통 세트에 대해 각각 동시에 동작할 수도 있다. 프로세싱 엔진들 (617) 은 사이클 (예컨대, 클록 사이클) 당 하나의 화소를 각각 컴퓨팅할 수 있고, 각각의 프로세싱 엔진 (617) 은 블록 내의 상이한 화소 또는 화소들의 세트를 계산할 수도 있다. 따라서, 인트라 예측 엔진은 합산기 (640) 에 의해 수신될 수도 있는, 사이클당 N 개 화소들을 컴퓨팅할 수도 있다.

[0088] 합산기 (640) 는 인트라 예측 프로세싱 부 (616) 에 의해 출력되는 N 개의 인트라 예측된 화소들의 각각의 세트를 역 변환 모듈 (620) 에 의해 계산된 N 개의 대응 잔차 값들에 가산함으로써, N 개의 화소들을 복원할 수도 있다. 복원된 화소들은 참조 프레임 버퍼 (630) 에 저장될 수도 있다. 일부 복원된 화소들은 후속 블록을 인트라 예측하기 위해 사용될 수도 있고, 이를 복원된 화소들은 이웃 버퍼 (612) 에 추가적으로 저장될 수도 있다. 일부 실시형태들에서, 이웃 버퍼 (612) 는 이를 복원된 화소들을 참조 프레임 버퍼 (630) 로부터 판독하고, 일부 실시형태들에서, 합산기 (640) 는 이를 복원된 화소들을 참조 프레임 버퍼 (630) 및 이웃 버퍼 (612) 둘 다에게 출력한다.

[0089] 프로세싱 엔진들의 수 (N) 는, 이웃 버퍼 (612) 의 메모리 대역폭으로 스케일링할 수도 있다. 예를 들어, 일부 실시형태들에서, 이웃 버퍼의 판독 대역폭은 사이클당 2N 개의 화소들일 수도 있다. 그러나, 버퍼 판독 대역폭 대 프로세싱 엔진들의 수의 다른 비율들이 사용될 수도 있다. 게다가, 이웃 RAM (613) 의 형상은 판독 대역폭 및/또는 프로세싱 엔진들의 수에 관련할 수도 있다. 예를 들어, 이웃 RAM (613) 은 RAM 깊이가 192/N일 수도 있고 RAM 폭이 16N일 수도 있는 192/Nx16N의 형상 및 사이즈를 갖도록 구성될 수도 있다. 이웃 버퍼 (612) 가 RAM (613) 을 포함하는 것으로 도시되지만, 메모리의 다른 적합한 유형들이 부가적으로 또는 대안적으로 사용될 수도 있다.

[0090] 블록에서의 화소들을 인트라 예측하는 경우, 인트라 예측 프로세싱 부 (616) 는 후속 블록들 (예컨대, 현재 블록을 바로 뒤따르는 블록) 의 인트라 예측을 위해 사용될 화소들을 우선시할 수도 있다. 이를 화소들이 인트라 예측된 후, 그것들은 이웃 버퍼 (612) 에 (뿐만 아니라 참조 프레임 버퍼 (630) 에) 다시 저장될 수도 있는 복원된 화소들을 생성하기 위해 합산기 (640) 에서 대응하는 잔차 값들과 합산될 수도 있다. 복원된 화소들은 그 다음에 이웃 프로세싱 부 (614) 에 의해 인트라 예측 프로세싱 부 (616) 를 위한 다른 방도들로 패딩, 및 잠재적으로 컨디셔닝될 수도 있다. 화소들의 적합한 우선시로, 이웃 프로세싱 부 (614) 는 인트라

예측 프로세싱 부 (616) 가 현재 블록을 인트라 예측하는 것을 끝내기 전에 미리 다음 블록에 대한 복원된 잔차 값들을 가질 수도 있다. 따라서, 인트라 예측 프로세싱 부 (616) 는, 심지어 다음 블록이 현재 블록에 대한 의존성들을 갖는 경우에도, 그리고/또는 현재 블록과 다음 블록이 공통 코딩 유닛 내에 있는 경우에도, 현재 블록이 끝난 직후에 다음 블록을 프로세싱하는 것을 시작할 수도 있다. 이는 전통적인 수법들을 능가하는 상당한 개선이다.

[0091] 도 7은 인트라 예측을 위해 선택되는 코딩 블록 (700) 을 예시하는 블록도를 도시한다. 코딩 블록 (700) 은 복수의 변환 블록들 (710-0, 710-1, 710-2, 및 710-3) 을 가질 수도 있는데, 이 변환 블록들은 그 순서로 일반적으로 프로세싱될 수도 있다. 비-HEVC 맥락들에서, 코딩 블록 (700) 은 매크로블록 (700) 을 나타낼 수도 있고 변환 블록들 (710-0, 710-1, 710-2, 및 710-3) 은 더 일반적으로는 블록들 (710-0, 710-1, 710-2, 및 710-3) 을 나타낼 수도 있다. 인트라 예측 동안, 코딩 블록 (700) 은 대응하는 변환 블록들 (710) 의 각각에 의해 공유되는 단일 인트라 예측 모드를 가질 수도 있다. 블록들 (710) 은 복원 동안 참조 프레임 버퍼가 업데이트되게 하는 유닛일 수도 있다.

[0092] 일부 시나리오들에서, 변환 블록이 대응하는 코딩 블록과 동일한 사이즈일 수도 있는 한편, 다른 시나리오들 (예컨대, 도 7에 도시된 바와 같음) 에서, 변환 블록들은 연관된 코딩 블록 (그 자체는 단일 예측 블록과 연관될 수도 있음) 보다 더 작을 수도 있다. 따라서, 이웃하는 변환 블록들의 그룹이 동일한 인트라 예측 모드를 가질 수도 있다. 그 결과, 코딩 블록 내의 변환 블록들은 인트라 예측 동안 이전의 변환 블록들과는 규칙적인 의존성 패턴을 가질 수도 있다.

[0093] 블록 의존성의 일 예로서, 수평 인트라 예측 모드 또는 특정한 대각선 인트라 예측 모드들이 선택된다면, 블록 (710-1) (또한 블록 1로서 라벨표시됨) 의 인트라 예측은 블록 (710-0) (또한 블록 0으로 라벨 표시됨) 의 중대한 열 (720) (예컨대, 최우측 열) 의 복원에 의존할 수도 있다. 유사하게, 특정한 대각선 인트라 예측 모드들이 선택된다면, 블록 (710-2) (또한 블록 2로서 라벨표시됨) 의 인트라 예측은 블록 (710-1) 의 중대한 행 (722) (예컨대, 바닥 행) 의 복원에 의존할 수도 있다.

[0094] 중대한 열 (720) 은 인트라 예측 동안 블록 (710-1) 이 의존하는 블록 (710-0) 의 유일한 일부일 수도 있다. 다른 이웃 화소들 (예컨대, 블록 (710-1) 상측의 이웃 화소들) 은 코딩 블록 (700) 상측의 코딩 블록의 완료된 복원의 결과로서 이웃 버퍼에 이미 복원 및 저장될 수도 있다. 따라서, 중대한 열 (720) 은 블록 (710-0) 의 인트라 예측 동안 우선시될 수도 있다. 중대한 열 (720) 에 대한 복원된 화소들이 결정되는 경우, 블록 (710-1) 의 인트라 예측을 위한 준비 (예컨대, 필터링 및 패딩) 가 시작될 수도 있다. 이 준비는, 아래의 도 8에서 도시될 바와 같이, 블록 (710-0) 의 인트라 예측이 여전히 진행 중인 동안 일어나고 완료될 수도 있다. 따라서, 블록 (710-1) 은 인트라 예측 프로세싱 부가 블록 (710-0) 을 프로세싱하는 것을 끝낸 직후에 인트라 예측될 수도 있다.

[0095] 유사하게, 중대한 행 (722) 은 인트라 예측 동안 블록 (710-2) 이 의존하는 블록 (710-1) 의 유일한 일부일 수도 있다. 다른 이웃 화소들 (예컨대, 블록 (710-2) 좌측의 이웃 화소들) 은 코딩 블록 (700) 좌측의 코딩 블록의 완료된 복원의 결과로서 이웃 버퍼에 이미 복원 및 저장될 수도 있다. 따라서, 중대한 행 (722) 은 블록 (710-1) 의 인트라 예측 동안 우선시될 수도 있고, 블록 (710-2) 은 인트라 예측 프로세싱 부가 블록 (710-1) 을 프로세싱하는 것을 완료한 직후에 인트라 예측될 수도 있다.

[0096] 중대한 행 또는 열이 우선시되는 경우, N 개의 프로세싱 엔진들 (도 6에 도시된 바와 같음) 은 프로세싱할 상이한 화소 또는 화소들의 세트를 각각 선택할 수도 있다. MxM 블록에서의 단일 행 또는 열이 M 개의 화소들을 가질 수도 있으므로, M/N 개의 사이클들이 단일의 중대한 행 또는 열을 인트라 예측하기 위해 요구될 수도 있다. 이는 다음 블록에 대한 이웃 화소들을 준비하는 이웃 프로세싱 부에게 충분한 시간을 제공할 수도 있는 전체 블록을 프로세싱하기 위해 요구될 수도 있는 M*M/N 개 사이클들보다 실질적으로 더 작을 수도 있다.

[0097] 예를 들어, 4x4 블록인 것으로 도시되는 블록 (710-0) 에 대해 작동하는 두 개의 프로세싱 엔진들이 있다면, 제 1 프로세싱 엔진은 제 1 사이클 동안 열 C3 및 행 R0의 교차부에서의 화소에 대해 인트라 예측된 값을 계산할 수도 있다. 이 동일한 사이클 동안, 제 2 프로세싱 엔진은 열 C3 및 행 R1의 교차부에서의 화소에 대해 인트라 예측된 값을 계산할 수도 있다. 다음의 사이클에서, 제 1 및 제 2 프로세싱 엔진들은 각각 열 C3 및 행 R2의 교차부에서의 화소에 대한 그리고 열 C3 및 행 R3의 교차부에서의 화소에 대한 값을 보간할 수도 있다. 따라서, 두 개의 사이클들에서, 중대한 열 (720) 의 인트라 예측은 완료될 수도 있다. 중대한 열 (720) 은 그 다음에 프로세싱 엔진들이 나머지 열들을 보간하고 있는 시간 동안 다음 블록 (예컨대, 블록 (710-1)) 의 인트라 예측을 위해 복원되며, 저장되고, 준비될 수도 있다. 다르게 말하면, 이웃 프로세싱 부에 의

해 수행되는 이웃 프로세싱은 인트라 예측 프로세싱 부에 의해 수행되는 인트라 예측과 파이프라이닝된다.

[0098] 대체로, 인트라 예측 프로세싱 부는 예측을 위한 화소들을 선택적으로 그리고 유연하게 선택할 수도 있다. 예를 들어, 위에서 입증된 바와 같이, 인트라 예측 프로세싱 부는, 인트라 예측될 다음 블록 (예컨대, 변환 블록)의 상대 위치에 적어도 부분적으로 의존하여, 열들 또는 행들에 대해 동작할 수도 있다. 인트라 예측될 다음 블록이 현재 블록의 우측에 있다면, 현재 블록의 최우측 열이 우선시될 수도 있다. 최우측 열이 프로세싱된 후, 블록의 나머지는 열 관련 패턴 (column-wise pattern)으로 프로세싱될 수도 있다. 유사하게, 인트라 예측될 다음 블록이 현재 블록의 하측 또는 좌하측이면, 현재 블록의 바닥 행이 우선시될 수도 있다. 현재 블록의 나머지는 행 관련 패턴 (row-wise pattern)으로 프로세싱될 수도 있다.

[0099] 중대한 열들 및 행들의 우선시는 블록들의 상이한 인트라 예측 순서들에 대해 조정될 수도 있다. 더욱이, 일부 실시형태들에서, 복수의 열들 또는 행들이 단일 열 또는 행 대신 우선시될 수도 있다. 예를 들어, 이는 블록의 인트라 예측이 가장 가까운 및 두 번째 가장 가까운 행 또는 열 둘 다를 사용하는 경우 일어날 수도 있다.

[0100] 심지어 인트라 예측이 인트라 예측되고 있는 블록의 하측 및 좌측에 있는 블록들로부터의 화소들을 이용할 수도 있다는 것이 도 4 및 도 5에서 도시되었더라도, 블록 (710-1)은 블록 (710-2)에 대해 임의의 의존성을 갖는 것으로 도시되지 않았을 수도 있다. 그러나, 이 의존성이 블록 (710-1)의 인트라 예측을 불필요하게 중단 또는 지연시키는 것을 방지하기 위해, 코딩 블록 (700)은 이 유형의 의존성이 일어나는 경우 회전되거나 또는 전치될 수도 있다. 이는 인트라 예측 벡터를 완료된 (또는 적어도 부분적으로 완료된) 블록들 쪽으로 효과적으로 이동시킴으로써, 코딩 블록 내의 연속적인 변환 블록들의 최적화된 프로세싱을 촉진시킨다.

[0101] 도 7이 네 개의 변환 블록들을 갖는 코딩 블록을 도시하지만, 코딩 블록 내의 변환 블록들의 수는 가변할 수도 있다. 예를 들어, 코딩 블록이 1, 4, 16, 64, 또는 256 개의 변환 블록들에 연관될 수도 있다. 더욱이, 변환 블록이 4x4의 사이즈를 갖는 것으로 도시되지만, 다른 블록 사이즈들이 선택될 수도 있다. 예를 들어, HEVC에서, 변환 블록이 4x4, 8x8, 16x16, 또는 32x32의 사이즈를 가질 수도 있다.

[0102] 도 8은 인터리브식 인트라 예측 기법을 예시하는 타이밍도를 도시한다. 구체적으로, 그 타이밍 도는 제 1 블록 (블록 0)의 인트라 예측을 위한 제 1 타임라인 (801)과, 제 2 블록 (블록 1)의 인트라 예측을 위한 제 2 타임라인 (802)를 도시한다. HEVC의 맥락에서, 이를 블록들은 변환 블록들을 나타낼 수도 있다. 그 타이밍 도는 도 6에 도시된 인트라 예측 서브시스템의 실시형태에 연관되고, 블록 사이즈는 MxM인 것으로 가정된다. 모든 타이밍들은 당연히 예시적이고 그것들은 프로세싱 지연들 및 다른 요인들에 의존하여 가변할 수도 있다.

[0103] 시간 810에서, 이웃 프로세싱 부가 제 1 블록의 인트라 예측을 위해 사용될 수도 있는 이웃 화소들을 폐치하거나 또는 그렇지 않으면 수신할 수도 있다. 위에서 설명된 바와 같이, 4M+1 개의 이웃 화소들이, 예를 들어, 이웃 버퍼로부터 수신될 수도 있다. 여기서, "1"은 별도의 레지스터에 저장될 수도 있는 (그리고 타이밍 계산에 대해 고려되지 않을 수도 있는) 모서리 (예컨대, 좌측상단 모서리) 화소를 나타낸다. 그러나, 모서리 화소는 이웃 버퍼에 부가적으로 또는 대안으로 저장될 수도 있다. 버퍼로부터의 판독 대역폭이 사이클당 2N 개 화소들이면, 이웃 화소들의 모두를 판독하기 위해 이웃 프로세싱 부에 대해 대략 2M/N 개 사이클들이 걸릴 수도 있다. 이 액션은 시간 820에서 완료될 수도 있다.

[0104] 이웃 화소들이 폐치된 후, 그 이웃 화소들은 인트라 예측 프로세싱 부를 대비하여 다른 방식으로 패딩, 필터링 및/또는 컨디셔닝될 수도 있다. 이는 고정된 수량의 사이클들 (예컨대, 두 개의 사이클들) 이 걸릴 수도 있고 시간 830에서 완료될 수도 있다.

[0105] 시간 830에서, 인트라 예측 프로세싱 부는 중대한 행 또는 열로 시작하여, 제 1 블록을 인트라 예측하는 것을 시작할 수도 있다. MxM 블록에서의 중대한 행 또는 열은 M 개의 화소들을 가질 수도 있고, 인트라 예측 프로세싱 부는 사이클당 N 개 화소들을 프로세싱할 수도 있다. 따라서, 대략 M/N 개 사이클들 후, 중대한 행 또는 열은 인트라 예측될 수도 있다. 인트라 예측된 값들은 이웃 버퍼에 저장되는 복원된 화소들을 생성하기 위해 대응하는 잔차 값들과 합산될 수도 있어서, 그것들은 시간 840에서는 이웃 프로세싱 부에 의해 이미 수신되었다.

[0106] 시간 840에서, 이웃 프로세싱 부는 제 2 블록의 인트라 예측을 위해 사용될 수도 있는 복원된 이웃 화소들을 폐치하거나 또는 그렇지 않으면 수신할 수도 있다. 위에서 언급된 바와 같이, 수신된 화소들의 일부는 제 1 블록의 중대한 행 또는 열과 대응함으로써, 이웃 프로세싱 부가 준비되었고 제 1 블록에 대한 이웃 화소들을 인

트라 예측 프로세싱 부로 전송했던 시간 830과 이웃 프로세싱 부가 제 2 블록에 대한 이웃 화소들을 수신하는 것을 시작하는 시간 840 사이에 지연을 초래할 수도 있다.

[0107] 대략 2M/N 개 사이클들에 걸쳐, 이웃 프로세싱 부는 제 2 블록을 인트라 예측하기 위한 이웃 화소들의 각각을 수신할 수도 있는데, 이는 시간 850에서 완료될 수도 있다. 이웃 화소들을 패딩, 필터링, 및/또는 그렇지 않으면 컨디셔닝하기 위한 고정된 수량의 사이클들 (예컨대, 두 개의 사이클들) 후, 이웃 프로세싱 부는 인트라 예측 프로세싱 부를 위한 화소들을 준비하는 것을 끝낼 수도 있다.

[0108] 시간 860에서, 인트라 예측 프로세싱 부는 제 1 블록을 프로세싱하는 것을 완료할 수도 있고 제 2 블록을 프로세싱하는 것을 시작할 수도 있다. 다르게 말하면, 제 2 블록은 제 1 블록 바로 직후에 프로세싱될 수도 있는데, 이웃 프로세싱 부가 시간 860에는 제 2 블록에 대한 이웃 화소들로 준비될 수도 있기 때문이다. 일부 실시형태들에서, 이웃 프로세싱 부는 시간 860에 앞서 제 2 블록에 대한 이웃 화소들의 준비 (예컨대, 패딩 및 필터링) 를 완료할 수도 있다.

[0109] 인트라 예측 프로세싱 부의 관점에서, 제 1 블록은 이웃 프로세싱 부로부터 시간 830에 수신되고 시간 860에 종료하는 대략 $M \times M/N$ 개 사이클들에서 프로세싱된다. 제 1 블록이 완전히 프로세싱되자 마자, 인트라 예측 프로세싱 부는 제 2 블록을 프로세싱하는 것을, 이웃 프로세싱 부에 의해 준비된 대응하는 4M+1 이웃 화소들을 사용하여 시작할 수도 있다. 대략 $M \times M/N$ 개 사이클들의 다른 시구간 후에, 인트라 예측 프로세싱 부는 제 2 블록을 프로세싱하는 것을 (시간 870에) 완료하고, 제 3 블록을 프로세싱하는 것을 이웃 프로세싱 부에 의해 미리 준비될 수도 있는 대응하는 4M+1 이웃 화소들을 사용하여 즉시 시작할 수도 있다.

[0110] 이웃 프로세싱 부는 인트라 예측 프로세싱 부가 제 1 블록을 여전히 프로세싱하는 (예컨대, 인트라 예측하는) 시간 동안 제 2 블록에 대한 이웃 화소들을 프로세싱 (예컨대, 버퍼로부터의 판독, 패딩, 및 필터링) 할 수도 있다는 것이 주의될 수도 있다. 이런 식으로, 이웃 프로세싱 부는 인트라 예측 프로세싱 부와 파이프라인화될 (그리고 인터리빙될) 수도 있으며, 이는 인트라 예측 모듈 및 서브시스템의 효율 및 스루풋을 증가시킬 수 있다.

[0111] 종종, 파이프라인식 아키텍처들에서, 데이터가 최소 시구간들 동안 버퍼들에 남아 있어야 하는데, 정적 데이터가 비효율성을 및 더 낮은 스루풋으로 이어질 (또는 그러한 비효율성을 및 더 낮은 스루풋을 나타낼) 수도 있기 때문이다. 각각의 블록에 대한 프로세싱 사이클은 대략 $M \times M/N$ 개 사이클들이고, 그에 따라, 이는 또한 4M (또는 4M+1) 개 이웃 화소들의 각각의 세트가 이웃 버퍼에 저장될 수도 있는 평균 시간일 수도 있다는 것에 주의할 수도 있다. 이 단축된 저장 기간 (기존의 설계들과 비교됨) 은 개시된 원리들에 따라 설계된 인트라 예측 모듈들 및 서브시스템들에 대해 성취 가능한 증가된 스루풋 및 효율을 나타낸다.

[0112] 다시, 도 8에 도시된 액션들 또는 기간들 중 임의의 액션 또는 기간에 대한 사이클들의 정확한 수는 단지 예시적이고, 그것들은 프로세싱 및 다른 요인들로 인해 지연들에 따라 달라질 수도 있다.

[0113] 프로세싱 엔진들의 수 (N) 및 이웃 버퍼 판독 대역폭 (예컨대, 사이클당 2N 개 화소들) 은 상이한 성능 요건들에 따라 둘 다 스케일링될 수도 있다. 아래의 표 1은 개시된 원리들에 따라 다양한 구성들을 도시한다.

	구성 1	구성 2	구성 3	구성 4
32x32 코딩 유닛에 대한 사이클 카운트 예산	400	800	1600	3200
이웃 버퍼 사이즈	2x24x64	48x64	96x32	192x16
인트라 예측 프로세싱 부에서의 PE들의 수	8	4	2	1
이웃 버퍼 판독 대역폭	16	8	4	2
면적 (μm^2)	50000	36000	26550	20500

[0114]

표 1

[0116] 표 1에 도시된 값들은 28 nm 기술 노드를 사용하여 400 MHz로 실행하는 하드웨어 설계들에 관하여 측정되었거나 또는 그렇지 않으면 결정되었다. 기술 노드 및 동작 주파수는 순수하게 예시적이고 다른 기술 노드들 및/또는 동작 주파수들이 사용될 수도 있다.

[0117] 그 표는 프로세싱 엔진들 (PE들) 의 수뿐만 아니라 이웃 버퍼 판독 대역폭 (및, 예컨대, 이웃 RAM 폭) 이, 사이클 카운트 예산으로서 주어지는 성능 요건들이에 대해, 더 높은 성능 요건들을 수용하기 위해 증가할 수도 있

다는 것을 도시한다. 예를 들어, 구성 1은, 예컨대 울트라 고품위 (UHD) 해상도 (3,840 바이 2,160 개 화소 들 또는 그 이상) 을 갖는, 비디오를 인코딩 및/또는 디코딩하도록 설계된 하이엔드 (high-end) 코딩 디바이스 들에 대해 선택될 수도 있다. 구성 2 및 구성 3은 중간레벨 디바이스들 (예컨대, 1080p) 을 향하여 지향될 수도 있고, 구성 4는 로엔드 (low-end) 디바이스들 (예컨대, 720p) 을 향하여 지향될 수도 있다.

[0118] 개시된 원리들에 따르는 다양한 실시형태들이 위에서 설명되었지만, 그들 실시형태들은 예로서만 제시되고 제한하고 있지 않다는 것이 이해되어야 한다. 따라서, 본 개시물의 폭과 범위는 전술한 예시적인 실시형태들 중 어느 것에 의해 제한되지 않아야 하며, 다만 본 개시물로부터 발표된 청구항들 및 그것들의 동등물들에 따라서만 정의되어야 한다. 더욱이, 위의 장점들 및 특징들은 설명된 실시형태들에서 제공되지만, 위의 장점들 중 임의의 것 또는 모두를 달성하는 프로세스들 및 구조들에의 이러한 발표된 청구항들의 적용을 제한하지 않을 것이다.

[0119] 본 개시물에서 사용된 다양한 용어들은 본 기술 분야 내에서 특별한 의미들을 가진다. 특정 용어가 그 용어가 사용되는 맥락에 의존하여 이러한 "기술분야의 용어 (term of art)"로서 해석되어야 한다. "에 연결된", "와 통신하는", "에 연관된", 또는 다른 유사한 용어들이 통신들 및 접속들이 참조된 엘리먼트들 간에 직접적이거나 또는 참조된 엘리먼트들 간의 하나 이상의 중간물들을 통한다는 양쪽 모두의 상황들을 포함하도록 일반적으로 넓게 해석되어야 한다. 이들 및 다른 용어들은 그것들이 본 개시물에서 사용되는 맥락에 비추어 그리고 본 기술분야의 통상의 기술자가 개시된 맥락에서 그들 항들을 이해하는 바대로 해석되는 것이다. 위의 정의들은 개시된 맥락에 기초하여 그들 항들에 부여될 수도 있었던 다른 의미를 배제하지 않는다.

[0120] "그 시간에", "즉시", "동등한", "동안", "완료한", "동일한" 등과 같은 비교, 측정, 및 타이밍의 단어들은 "실질적으로 그 시간에", "실질적으로 즉시", "실질적으로 동등한", "실질적으로 동안", "실질적으로 완료한", "실질적으로 동일한" 등을 의미하는 것으로 이해되어야 하며, 여기서 "실질적으로"는 이러한 비교들, 측정들, 및 타이밍들이 암시적으로 또는 명시적으로 언급된 원하는 결과를 달성하기 위해 실현 가능하다는 것을 의미한다.

[0121] "블록"이라는 용어가 사용되는 경우, 이는 최대 코딩 유닛 (LCU), 코딩 트리 블록 (CTB), 코딩 유닛 (CU), 코딩 블록 (CB), 예측 유닛 (PU), 예측 블록 (PB), 변환 유닛 (TU), 변환 블록 (TB), H.264/AVC에 의해 정의된 바와 같은 서브블록, 및/또는 그것이 사용되는 맥락에 의존하는 유사한 데이터 구조들 중 임의의 것을 지칭할 수도 있다.

[0122] 다양한 발명의 주제가 HEVC에 관해 설명되지만, 이 발명의 주제는 현재 존재하는, 개발 중인, 및 아직 개발되지 않은 것들을 포함하는 다른 표준들에 또한 적용 가능할 수도 있다. 예를 들어, "CU" 및 "CB"라는 용어들이 사용되는 경우, 그것들은 그것들이 사용되는 맥락에 의존하여, H.264/AVC 표준에 의해 정의된 바와 같은 매크로 블록 및/또는 다른 인코딩 표준들의 또는 유사한 데이터 구조들을 또한 지칭할 수도 있다. 유사하게, "TU" 및 "TB"라는 용어들이 사용되는 경우, 그것들은 그것들이 사용되는 맥락에 의존하여, H.264/AVC 표준에 의해 정의된 바와 같은 서브블록 및/또는 다른 인코딩 표준들의 유사한 데이터 구조들을 또한 지칭할 수도 있다.

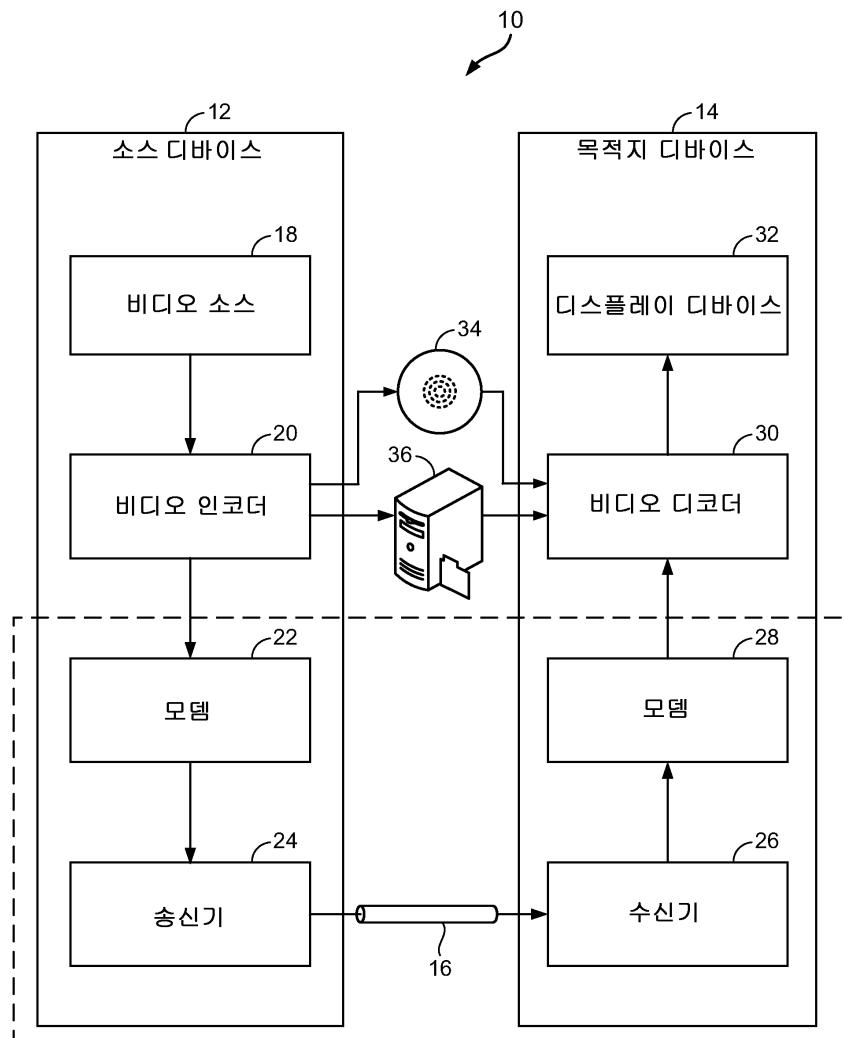
[0123] 게다가, "화소"라는 용어가 사용되는 경우, 이는 비디오 프레임 또는 이미지 내의 루마 값, 크로마 값, 또는 크로마 및 루마 값들의 조합을 지칭할 수도 있다.

[0124] 개시된 원리들은 파면 병렬 프로세싱 (wavefront parallel processing, WPP), 비디오 슬라이스들, 및 비디오타일들을 포함한 병렬성을 성취하기 위한 수많은 다른 기법들과 일반적으로 양립한다.

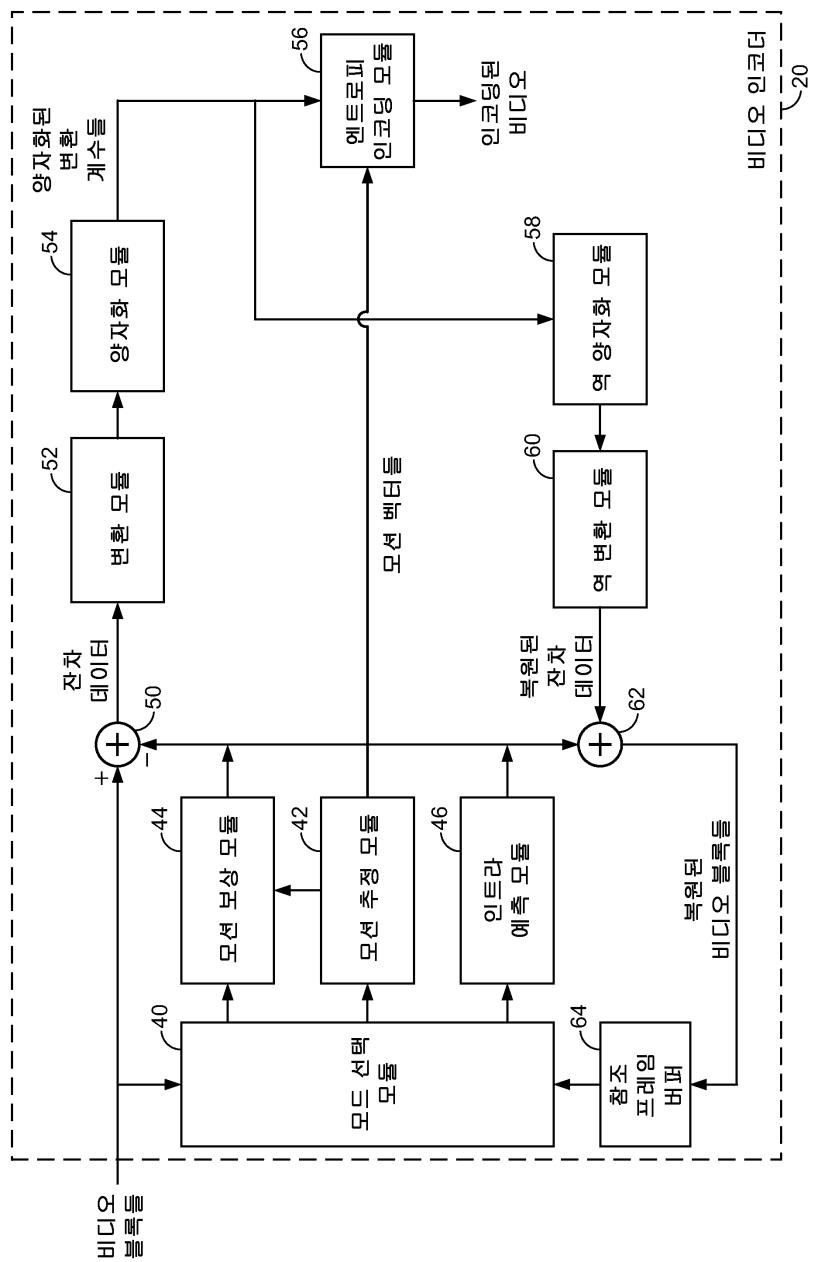
[0125] 덧붙여, 본 명세서에서의 섹션 제목들은 37 C.F.R. 1.77 하의 제안들과의 일관성을 위해 제공되거나 또는 그렇지 않으면 구조적인 신호들을 제공하는 것이다. 이들 제목들은 본 개시물에서 나올 수도 있는 임의의 청구항들에서 언급되는 발명의 주제를 제한하거나 또는 특징화하지 않을 것이다. 구체적으로 그리고 예로서, 비록 제목들이 "기술 분야"를 지칭하지만, 이러한 청구항들은 이른바 기술 분야를 설명하기 위해 이 제목 하에서 선택된 언어표현에 의해 제한되지 않아야 한다. 게다가, "배경"에서의 기술의 설명은 기술이 본 개시물에서의 임의의 발명의 주제에 대한 선행 기술이라는 것을 시인하는 것으로서 해석되는 것이 아니다. "요약"이 발표된 청구항들에서 언급되는 발명의 주제의 특징으로서 간주되는 것도 아니다. 더욱이, 단수형의 "발명"에 대한 본 개시물에서의 임의의 언급은 본 개시물에 단일의 신규성 요소 (point of novelty) 만 있다는 것을 주장하기 위해 사용되지 않아야 한다. 다수의 발명들이 본 개시물에서 발표되는 다수의 청구항들의 한정들에 따라 언급될 수도 있고, 이러한 청구항들은 본 발명(들), 및 그것에 의해 보호되는 그것들의 동등물들을 정의한다. 모든 경우들에서, 이러한 청구항들의 범위는 본 개시물에 비추어 그 청구항들 소유의 장점들에 관하여 고려될 것이지만, 본 명세서에서 언급되는 제목들에 의해 제한되지 않아야 한다.

도면

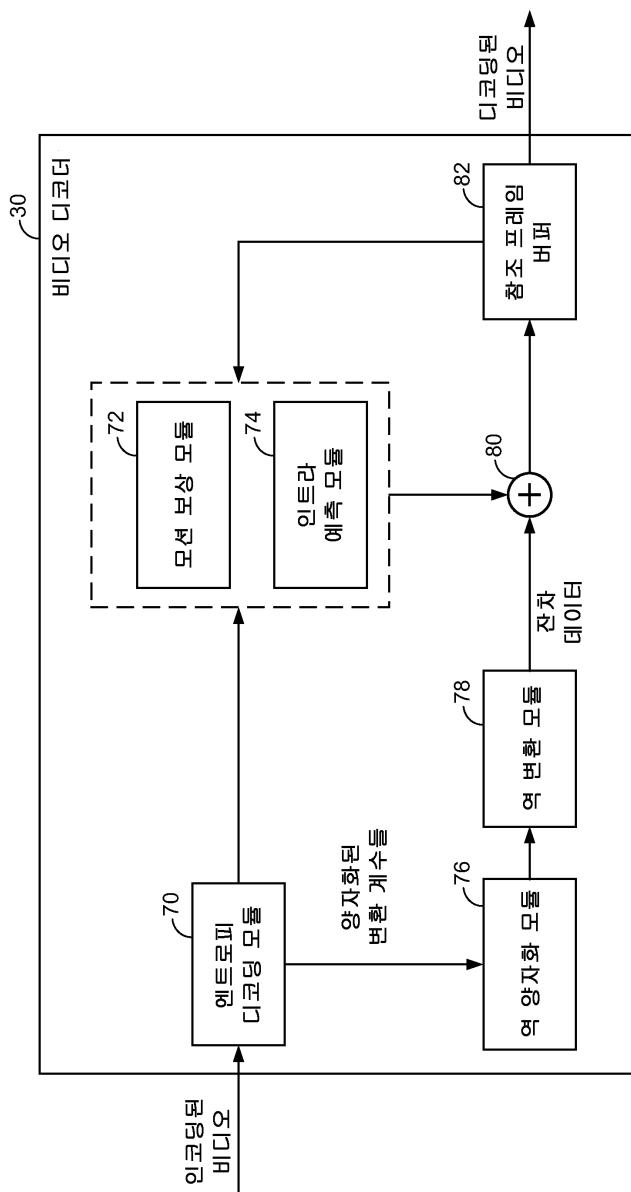
도면1



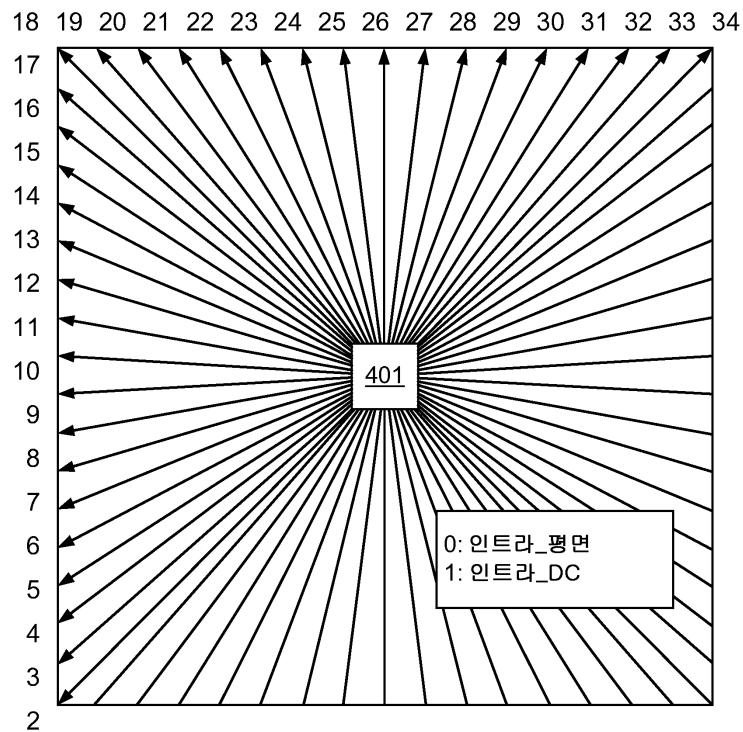
도면2



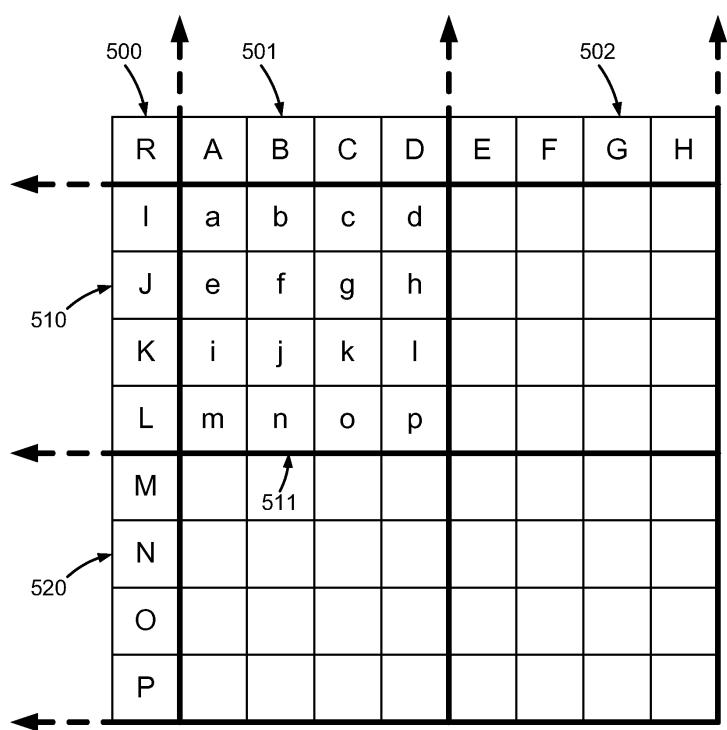
도면3



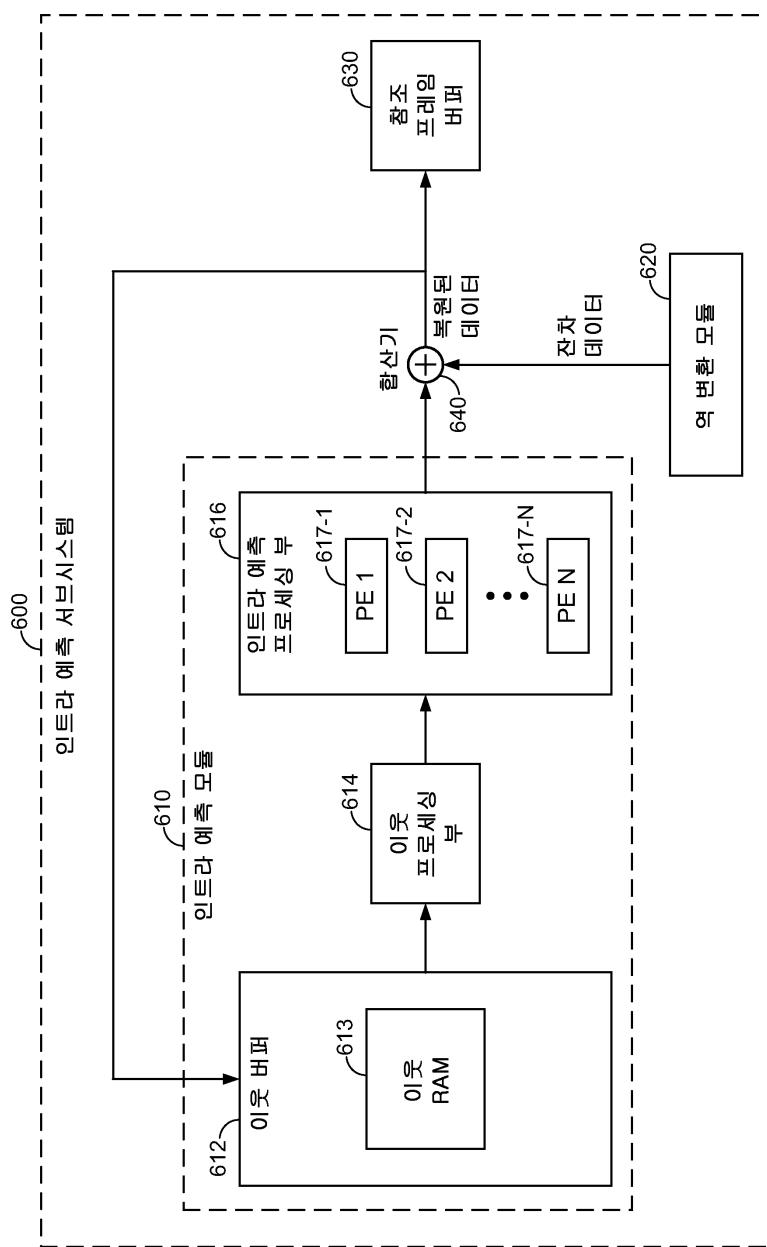
도면4



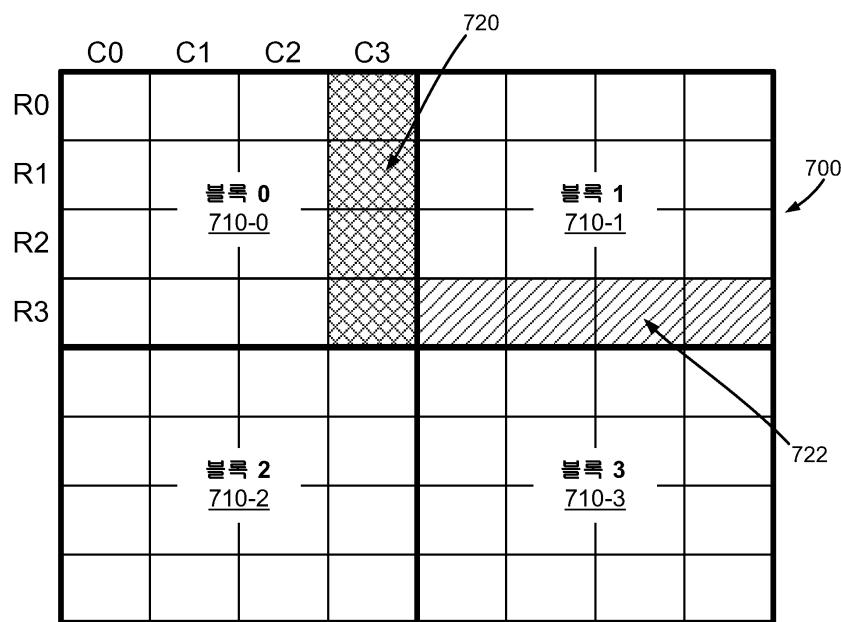
도면5



도면6



도면7



도면8

