



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2014-0116194
 (43) 공개일자 2014년10월01일

- | | |
|---|--|
| (51) 국제특허분류(Int. Cl.) H04N 19/13 (2014.01) H04N 19/18 (2014.01) (21) 출원번호 10-2014-7022370 (22) 출원일자(국제) 2013년01월11일 심사청구일자 없음 (85) 번역문제출일자 2014년08월11일 (86) 국제출원번호 PCT/US2013/021234 (87) 국제공개번호 WO 2013/106710 국제공개일자 2013년07월18일 (30) 우선권주장 13/738,534 2013년01월10일 미국(US) (뒷면에 계속) | (71) 출원인 켈컴 인코포레이티드 미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775 (72) 발명자 세레진 바담 미국 92121 캘리포니아주 샌디에고 모어하우스 드라이브 5775 솔레 로할스 호엘 미국 92121 캘리포니아주 샌디에고 모어하우스 드라이브 5775 카르체비츠 마르타 미국 92121 캘리포니아주 샌디에고 모어하우스 드라이브 5775 (74) 대리인 특허법인코리아나 |
|---|--|

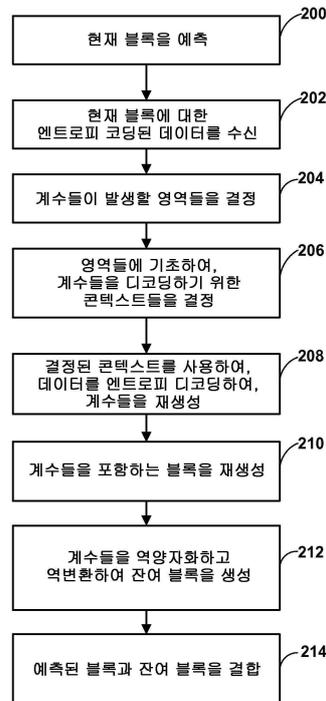
전체 청구항 수 : 총 37 항

(54) 발명의 명칭 비디오 코딩에서 변환 계수 데이터를 코딩하기 위한 콘텍스트들의 결정

(57) 요약

일 예에서, 비디오 데이터를 코딩하기 위한 디바이스는 변환 계수가 발생하는 비디오 블록의 영역에 기초하여, 비디오 블록의 변환 계수를 코딩하기 위한 콘텍스트를 결정하며, 결정된 콘텍스트를 사용하여 변환 계수를 엔트로피 코딩하도록 구성된 비디오 코더를 포함한다. 영역은 비디오 블록의 변환 계수들의 하나 이상의 상부-좌측 4×4 서브 블록들을 포함하는 제 1 영역, 및 제 1 영역 외부의 비디오 블록의 변환 계수들을 포함하는 제 2 영역 중 하나를 포함할 수도 있다.

대표도 - 도16



(30) 우선권주장

61/586,668 2012년01월13일 미국(US)

61/588,595 2012년01월19일 미국(US)

61/597,097 2012년02월09일 미국(US)

특허청구의 범위

청구항 1

비디오 데이터를 코딩하는 방법으로서,

변환 계수가 발생하는 비디오 블록의 영역에 기초하여 상기 비디오 블록의 변환 계수를 코딩하기 위한 콘텍스트를 결정하는 단계; 및

결정된 상기 콘텍스트를 사용하여 상기 변환 계수를 엔트로피 코딩하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 2

제 1 항에 있어서,

상기 영역은, 상기 비디오 블록의 변환 계수들의 하나 이상의 상부-좌측 4×4 서브 블록들을 포함하는 제 1 영역, 및 상기 제 1 영역 외부의 상기 비디오 블록의 변환 계수들을 포함하는 제 2 영역 중 하나를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 3

제 1 항에 있어서,

상기 영역은 상기 비디오 블록의 복수의 영역들 중 하나를 포함하며,

상기 영역들 각각은 상기 비디오 블록의 하나 이상의 서브 블록들의 각각의 세트들을 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 4

제 1 항에 있어서,

상기 변환 계수를 코딩하는 단계는, 상기 변환 계수와 연관된 유의도 정보, 상기 변환 계수의 레벨 정보, 및 상기 변환 계수와 연관된 부호 정보 중 하나 이상을 코딩하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 5

제 1 항에 있어서,

상기 비디오 블록은, 변환 유닛 (transform unit; TU), 예측 유닛 (prediction unit; PU), 코딩 유닛 (coding unit; CU), 최대 코딩 유닛 (largest coding unit; LCU), 및 블록들의 그룹 중 하나를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 6

제 1 항에 있어서,

상기 콘텍스트를 결정하는 단계는, 상기 영역에 기초하여 포지션-기반 콘텍스트 정보 및 이웃-기반 콘텍스트 정보 중 하나를 사용하여 상기 콘텍스트를 결정하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 7

제 1 항에 있어서,

상기 콘텍스트를 결정하는 단계는, 상기 영역에 기초하여 상기 비디오 블록에 대한 포지션-기반 콘텍스트에 적용된 오프셋을 결정하는 단계를 포함하며,

상기 영역에 대한 상기 오프셋은, 고정된 오프셋, 및 상기 비디오 블록의 사이즈, 상기 비디오 블록 내의 상기 변환 계수의 포지션, 및 상기 비디오 블록 내의 상기 변환 계수를 포함하는 서브 블록의 포지션 중 하나

이상에 의존하는 오프셋 중 하나인, 비디오 데이터를 코딩하는 방법.

청구항 8

제 1 항에 있어서,

상기 비디오 블록의 복수의 영역들로부터 상기 영역을 결정하는 단계를 더 포함하며,

상기 영역들 각각은 복수의 변환 유닛 (TU) 사이즈들의 각각의 사이즈에 대응하며,

상기 콘텍스트를 결정하는 단계는, 상기 영역과 그 영역과 동일한 사이즈를 갖는 TU 사이에 공유되는 콘텍스트를 선택하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 9

제 1 항에 있어서,

상기 비디오 블록의 복수의 영역들로부터 상기 영역을 결정하는 단계를 더 포함하며,

상기 영역들 각각은 복수의 변환 유닛 (TU) 사이즈들의 각각의 사이즈에 대응하며,

상기 콘텍스트를 결정하는 단계는 상이한 사이즈들의 2 이상의 TU들 사이에서 변환 계수들의 전용 포지션들에 대한 공유된 콘텍스트를 선택하는 단계를 포함하며,

상기 영역은 상기 상이한 사이즈들의 2 이상의 TU들 중 하나와 동일한 사이즈를 갖는, 비디오 데이터를 코딩하는 방법.

청구항 10

제 9 항에 있어서,

상기 변환 계수들의 전용 포지션들에 대한 상기 공유된 콘텍스트는, DC 계수들, 및 상기 상이한 사이즈들의 2 이상의 TU들 사이에 공유되는 높은 주파수 계수들 중 하나에 대한 콘텍스트를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 11

제 9 항에 있어서,

상기 변환 계수들의 전용 포지션들에 대한 상기 공유된 콘텍스트는, 4×4 변환 계수들의 사이즈를 갖는 제 1 TU 와 8×8 변환 계수들의 사이즈를 갖는 제 2 TU 사이의 공유된 콘텍스트를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 12

제 1 항에 있어서,

상기 비디오 블록은 비-정사각형 비디오 블록을 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 13

제 1 항에 있어서,

상기 변환 계수를 엔트로피 코딩하는 단계는, 콘텍스트 적응 바이너리 산술 코딩 (CABAC) 에 따라서 상기 결정된 콘텍스트를 사용하여 상기 변환 계수를 엔트로피 디코딩하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 14

제 1 항에 있어서,

상기 변환 계수를 엔트로피 코딩하는 단계는,

콘텍스트 적응 바이너리 산술 코딩 (CABAC) 에 따라서 상기 결정된 콘텍스트를 사용하여 상기 변환 계수를 엔트로피 인코딩하는 단계를 포함하는, 비디오 데이터를 코딩하는 방법.

청구항 15

비디오 데이터를 코딩하기 위한 디바이스로서,

변환 계수가 발생하는 비디오 블록의 영역에 기초하여 상기 비디오 블록의 변환 계수를 코딩하기 위한 콘텍스트를 결정하고, 결정된 상기 콘텍스트를 사용하여 상기 변환 계수를 엔트로피 코딩하도록 구성된 비디오 코더를 포함하는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 16

제 15 항에 있어서,

상기 영역은, 상기 비디오 블록의 변환 계수들의 하나 이상의 상부-좌측 4×4 서브 블록들을 포함하는 제 1 영역, 및 상기 제 1 영역 외부의 상기 비디오 블록의 변환 계수들을 포함하는 제 2 영역 중 하나를 포함하는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 17

제 15 항에 있어서,

상기 영역은 상기 비디오 블록의 복수의 영역들 중 하나를 포함하며, 상기 영역들 각각은 상기 비디오 블록의 하나 이상의 서브 블록들의 각각의 세트들을 포함하는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 18

제 15 항에 있어서,

상기 변환 계수를 코딩하기 위해, 상기 비디오 코더는 상기 변환 계수와 연관된 유의도 정보, 상기 변환 계수의 레벨 정보, 및 상기 변환 계수와 연관된 부호 정보 중 하나 이상을 코딩하도록 구성되는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 19

제 15 항에 있어서,

상기 비디오 블록은, 변환 유닛 (TU), 예측 유닛 (PU), 코딩 유닛 (CU), 최대 코딩 유닛 (LCU), 및 블록들의 그룹 중 하나를 포함하는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 20

제 15 항에 있어서,

상기 비디오 코더는, 상기 영역에 기초하여 포지션-기반 콘텍스트 정보 및 이웃-기반 콘텍스트 정보 중 하나를 사용하여 상기 콘텍스트를 결정하도록 구성되는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 21

제 15 항에 있어서,

상기 비디오 코더는 또한, 상기 영역에 기초하여 상기 비디오 블록에 대한 포지션-기반 콘텍스트에 적용된 오프셋을 결정하도록 구성되며,

상기 영역에 대한 상기 오프셋은, 고정된 오프셋, 및 상기 비디오 블록의 사이즈, 상기 비디오 블록 내의 상기 변환 계수의 포지션, 및 상기 비디오 블록 내의 상기 변환 계수를 포함하는 서브 블록의 포지션 중 하나 이상에 의존하는 오프셋 중 하나인, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 22

제 15 항에 있어서,

상기 비디오 코더는, 상기 변환 계수를 엔트로피 디코딩하도록 구성된 비디오 디코더를 포함하는, 비디오 데

이터를 코딩하기 위한 디바이스.

청구항 23

제 15 항에 있어서,

상기 비디오 코더는, 상기 변환 계수를 엔트로피 인코딩하도록 구성된 비디오 인코더를 포함하는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 24

비디오 데이터를 코딩하기 위한 디바이스로서,

변환 계수가 발생하는 비디오 블록의 영역에 기초하여 상기 비디오 블록의 변환 계수를 코딩하기 위한 콘텍스트를 결정하기 위한 수단; 및

결정된 상기 콘텍스트를 사용하여 상기 변환 계수를 엔트로피 코딩하기 위한 수단을 포함하는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 25

제 24 항에 있어서,

상기 영역은, 상기 비디오 블록의 변환 계수들의 하나 이상의 상부-좌측 4×4 서브 블록들을 포함하는 제 1 영역, 및 상기 제 1 영역 외부의 상기 비디오 블록의 변환 계수들을 포함하는 제 2 영역 중 하나를 포함하는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 26

제 24 항에 있어서,

상기 영역은 상기 비디오 블록의 복수의 영역들 중 하나를 포함하며,

상기 영역들 각각은 상기 비디오 블록의 하나 이상의 서브 블록들의 각각의 세트들을 포함하는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 27

제 24 항에 있어서,

상기 변환 계수를 코딩하기 위한 수단은, 상기 변환 계수와 연관된 유의도 정보, 상기 변환 계수의 레벨 정보, 및 상기 변환 계수와 연관된 부호 정보 중 하나 이상을 코딩하기 위한 수단을 포함하는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 28

제 24 항에 있어서,

상기 비디오 블록은, 변환 유닛 (TU), 예측 유닛 (PU), 코딩 유닛 (CU), 최대 코딩 유닛 (LCU), 및 블록들의 그룹 중 하나를 포함하는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 29

제 24 항에 있어서,

상기 콘텍스트를 결정하기 위한 수단은, 상기 영역에 기초하여 포지션-기반 콘텍스트 정보 및 이웃-기반 콘텍스트 정보 중 하나를 사용하여 상기 콘텍스트를 결정하기 위한 수단을 포함하는, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 30

제 24 항에 있어서,

상기 콘텍스트를 결정하기 위한 수단은, 상기 영역에 기초하여 상기 비디오 블록에 대한 포지션-기반 콘텍스트

트에 적용된 오프셋을 결정하기 위한 수단을 포함하며,

상기 영역에 대한 상기 오프셋은, 고정된 오프셋, 및 상기 비디오 블록의 사이즈, 상기 비디오 블록 내의 상기 변환 계수의 포지션, 및 상기 비디오 블록 내의 상기 변환 계수를 포함하는 서브 블록의 포지션 중 하나 이상에 의존하는 오프셋 중 하나인, 비디오 데이터를 코딩하기 위한 디바이스.

청구항 31

명령들을 저장하고 있는 컴퓨터 판독가능 저장 매체로서,

상기 명령들은, 실행되는 경우 프로세서로 하여금,

변환 계수가 발생하는 비디오 블록의 영역에 기초하여 상기 비디오 블록의 변환 계수를 코딩하기 위한 콘텍스트를 결정하게 하고;

결정된 상기 콘텍스트를 사용하여 상기 변환 계수를 엔트로피 코딩하게 하는, 컴퓨터 판독가능 저장 매체.

청구항 32

제 31 항에 있어서,

상기 영역은, 상기 비디오 블록의 변환 계수들의 하나 이상의 상부-좌측 4×4 서브 블록들을 포함하는 제 1 영역, 및 상기 제 1 영역 외부의 상기 비디오 블록의 변환 계수들을 포함하는 제 2 영역 중 하나를 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 33

제 31 항에 있어서,

상기 영역은 상기 비디오 블록의 복수의 영역들 중 하나를 포함하며, 상기 영역들 각각은 상기 비디오 블록의 하나 이상의 서브 블록들의 각각의 세트들을 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 34

제 31 항에 있어서,

상기 프로세서로 하여금, 상기 변환 계수를 코딩하게 하는 명령들은,

상기 프로세서로 하여금, 상기 변환 계수와 연관된 유의도 정보, 상기 변환 계수의 레벨 정보, 및 상기 변환 계수와 연관된 부호 정보 중 하나 이상을 코딩하게 하는 명령들을 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 35

제 31 항에 있어서,

상기 비디오 블록은, 변환 유닛 (TU), 예측 유닛 (PU), 코딩 유닛 (CU), 최대 코딩 유닛 (LCU), 및 블록들의 그룹 중 하나를 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 36

제 31 항에 있어서,

상기 프로세서로 하여금, 상기 콘텍스트를 결정하게 하는 명령들은,

상기 프로세서로 하여금, 상기 영역에 기초하여 포지션-기반 콘텍스트 정보 및 이웃-기반 콘텍스트 정보 중 하나를 사용하여 상기 콘텍스트를 결정하게 하는 명령들을 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 37

제 31 항에 있어서,

상기 프로세서로 하여금, 상기 콘텍스트를 결정하게 하는 명령들은, 상기 프로세서로 하여금, 상기 영역에 기초하여 상기 비디오 블록에 대한 포지션-기반 콘텍스트에 적용된 오프셋을 결정하게 하는 명령들을 포함하며,

상기 영역에 대한 상기 오프셋은, 고정된 오프셋, 및 상기 비디오 블록의 사이즈, 상기 비디오 블록 내의 상

기 변환 계수의 포지션, 및 상기 비디오 블록 내의 상기 변환 계수를 포함하는 서브 블록의 포지션 중 하나 이상에 의존하는 오프셋 중 하나인, 컴퓨터 판독가능 저장 매체.

명세서

기술 분야

[0001] 본 출원은 2012년 1월 13일자에 출원된 미국 가출원 제 61/586,668호, 2012년 1월 13일자에 출원된, 2012년 1월 19일자에 출원된 미국 가출원 번호 제 61/588,595호, 및 2012년 2월 9일자에 출원된 미국 가출원 제 61/597,097호에 우선권을 주장하며, 이의 각각은 본원에서 전체적으로 참고로 포함된다.

[0002] 기술 분야

[0003] 본 개시물은 비디오 코딩에 관한 것이다.

배경 기술

[0004] 디지털 비디오 능력들은, 디지털 텔레비전, 디지털 직접 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인 휴대정보 단말기들 (PDAs), 랩탑 또는 데스크탑 컴퓨터들, 태블릿 컴퓨터들, e-북 리더들, 디지털 카메라들, 디지털 리코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 무선 전화기들, 소위 "스마트폰들", 원격 화상회의 디바이스들, 비디오 스트리밍 디바이스들 등을 포함한, 광범위한 디바이스들에 포함될 수 있다. 디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, 파트 10, AVC (Advanced Video Coding), 현재 개발중인 HEVC (High Efficiency Video Coding) 표준, 및 이러한 표준들의 확장판들에 의해 정의된 표준들에서 설명되는 비디오 코딩 기법들과 같은, 비디오 압축 기법들을 구현한다. 비디오 디바이스들은 이러한 비디오 압축 기법들을 구현함으로써, 디지털 비디오 정보를 좀더 효율적으로 송신, 수신, 인코딩, 디코딩, 및/또는 저장할 수도 있다.

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

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

발명의 내용

해결하려는 과제

[0007] 일반적으로, 본 개시물은 예컨대, 콘텍스트-적응 바이너리 산술 코딩 (context-adaptive binary arithmetic coding; CABAC) 을 사용하여, 엔트로피 코딩하기 위한 비디오 데이터의 콘텍스트들을 결정하는 것과 관련된 기법들을 기술한다. CABAC 코딩은 일반적으로 여러 구문 엘리먼트들의 이진화된 표현들을 코딩할 때 콘텍스트를 결정하는 것을 수반한다. 구문 엘리먼트들의 예들은 변환 계수들이 유의한지 여부를 나타내는 데이터, 유의한 변환 계수들의 부호들, 및 유의한 변환 계수들에 대한 레벨 값들과 같은, 변환 계수들에 대한

데이터를 포함한다. 변환 계수들은 일반적으로 변환 유닛 (TU) 과 같은, 변환 블록의 계수들에 대응한다.

본 개시물은 변환 계수들이 발생하는 변환 블록의 영역들에 기초하여 변환 계수들을 코딩하기 위한 콘텍스트들을 결정하는 기법들을 기술한다.

- [0008] 일 예에서, 비디오 데이터를 코딩하는 방법은, 변환 계수가 발생하는 비디오 블록의 영역에 기초하여 비디오 블록의 변환 계수를 코딩하기 위한 콘텍스트를 결정하는 단계, 및 결정된 콘텍스트를 사용하여 변환 계수를 엔트로피 코딩하는 단계를 포함한다.
- [0009] 또 다른 예에서, 비디오 데이터를 코딩하기 위한 디바이스는, 변환 계수가 발생하는 비디오 블록의 영역에 기초하여, 비디오 블록의 변환 계수를 코딩하기 위한 콘텍스트를 결정하며, 결정된 콘텍스트를 사용하여 변환 계수를 엔트로피 코딩하도록 구성된 비디오 코더를 포함한다.
- [0010] 또 다른 예에서, 비디오 데이터를 코딩하기 위한 디바이스는, 변환 계수가 발생하는 비디오 블록의 영역에 기초하여 비디오 블록의 변환 계수를 코딩하기 위한 콘텍스트를 결정하기 위한 수단, 및 결정된 콘텍스트를 사용하여 변환 계수를 엔트로피 코딩하기 위한 수단을 포함한다.
- [0011] 또 다른 예에서, 컴퓨터 판독가능 저장 매체는, 실행되는 경우 프로세서로 하여금, 변환 계수가 발생하는 비디오 블록의 영역에 기초하여, 비디오 블록의 변환 계수를 코딩하기 위한 콘텍스트를 결정하며, 결정된 콘텍스트를 사용하여 변환 계수를 엔트로피 코딩하게 하는 명령들을 저장하고 있다.
- [0012] 또 다른 예에서, 비디오 데이터를 디코딩하는 방법은, 변환 계수가 비디오 블록의 DC 변환 계수인 것으로 결정되는 경우, 비디오 블록의 변환 계수가 DC 변환 계수인지 여부를 결정하는 단계, 비디오 블록의 사이즈에 관계 없이 DC 변환 계수인 변환 계수에 기초하여 변환 계수를 디코딩하기 위한 콘텍스트를 결정하는 단계, 및 결정된 콘텍스트를 사용하여 변환 계수를 엔트로피 디코딩하는 단계를 포함한다.
- [0013] 또 다른 예에서, 비디오 데이터를 디코딩하기 위한 디바이스는, 변환 계수가 비디오 블록의 DC 변환 계수인 것으로 결정되는 경우 비디오 블록의 변환 계수가 DC 변환 계수인지 여부를 결정하고, 비디오 블록의 사이즈에 관계 없이 DC 변환 계수인 변환 계수에 기초하여 변환 계수를 디코딩하기 위한 콘텍스트를 결정하며, 결정된 콘텍스트를 사용하여 변환 계수를 엔트로피 디코딩하도록 구성된 비디오 디코더를 포함한다.
- [0014] 또 다른 예에서, 비디오 데이터를 디코딩하는, 디바이스는 비디오 블록의 변환 계수가 DC 변환 계수인지 여부를 결정하기 위한 수단, 변환 계수가 비디오 블록의 DC 변환 계수인 것으로 결정되는 경우, 비디오 블록의 사이즈에 관계 없이 DC 변환 계수인 변환 계수에 기초하여 변환 계수를 디코딩하기 위한 콘텍스트를 결정하기 위한 수단, 및 결정된 콘텍스트를 사용하여 변환 계수를 엔트로피 디코딩하기 위한 수단을 포함한다.
- [0015] 또 다른 예에서, 컴퓨터 판독가능 저장 매체는, 실행되는 경우 프로세서로 하여금, 변환 계수가 비디오 블록의 DC 변환 계수인 것으로 결정되는 경우 비디오 블록의 변환 계수가 DC 변환 계수인지 여부를 결정하고, 비디오 블록의 사이즈에 관계 없이 DC 변환 계수인 변환 계수에 기초하여 변환 계수를 디코딩하기 위한 콘텍스트를 결정하며, 결정된 콘텍스트를 사용하여 변환 계수를 엔트로피 디코딩하게 하는 명령들을 저장하고 있다.
- [0016] 또 다른 예에서, 비디오 데이터를 인코딩하는 방법은, 변환 계수가 비디오 블록의 DC 변환 계수인 것으로 결정되는 경우, 비디오 블록의 변환 계수가 DC 변환 계수인지 여부를 결정하는 단계, 비디오 블록의 사이즈에 관계 없이 DC 변환 계수인 변환 계수에 기초하여 변환 계수를 인코딩하기 위한 콘텍스트를 결정하는 단계, 및 결정된 콘텍스트를 사용하여 변환 계수를 엔트로피 인코딩하는 단계를 포함한다.
- [0017] 또 다른 예에서, 비디오 데이터를 인코딩하기 위한 디바이스는, 변환 계수가 비디오 블록의 DC 변환 계수인 것으로 결정되는 경우 비디오 블록의 변환 계수가 DC 변환 계수인지 여부를 결정하고, DC 비디오 블록의 사이즈에 관계 없이 변환 계수인 변환 계수에 기초하여 변환 계수를 인코딩하기 위한 콘텍스트를 결정하며, 결정된 콘텍스트를 사용하여 변환 계수를 엔트로피 인코딩하도록 구성된 비디오 인코더를 포함한다.
- [0018] 또 다른 예에서, 비디오 데이터를 인코딩하기 위한 디바이스는, 비디오 블록의 변환 계수가 DC 변환 계수인지 여부를 결정하기 위한 수단, 변환 계수가 비디오 블록의 DC 변환 계수인 것으로 결정되는 경우, 비디오 블록의 사이즈에 관계 없이 DC 변환 계수인 변환 계수에 기초하여 변환 계수를 인코딩하기 위한 콘텍스트를 결정하기 위한 수단, 및 결정된 콘텍스트를 사용하여 변환 계수를 엔트로피 인코딩하기 위한 수단을 포함한다.
- [0019] 또 다른 예에서, 컴퓨터 판독가능 저장 매체는, 실행되는 경우 프로세서로 하여금, 변환 계수가 비디오 블록의 DC 변환 계수인 것으로 결정되는 경우 비디오 블록의 변환 계수가 DC 변환 계수인지 여부를 결정하고, DC 비디오 블록의 사이즈에 관계 없이 변환 계수인 변환 계수에 기초하여 변환 계수를 인코딩하기 위한 콘텍스트

를 결정하며, 결정된 콘텍스트를 사용하여 변환 계수를 엔트로피 인코딩하게 하는 명령들을 저장하고 있다.

[0020] 또 다른 예에서, 비디오 데이터를 디코딩하는 방법은, 현재의 서브 블록에 대한 하나 이상의 이웃하는 서브 블록들의 코딩된 서브 블록 플래그들에 대한 값들을 결정하는 단계, 코딩된 서브 블록 플래그들에 대한 값들에 기초하여 현재의 서브 블록의 변환 계수를 디코딩하기 위한 콘텍스트를 결정하는 단계, 및 결정된 콘텍스트를 사용하여 변환 계수를 엔트로피 디코딩하는 단계를 포함한다.

[0021] 또 다른 예에서, 비디오 데이터를 디코딩하기 위한 디바이스는, 현재의 서브 블록에 대한 하나 이상의 이웃하는 서브 블록들의 코딩된 서브 블록 플래그들에 대한 값들을 결정하고, 코딩된 서브 블록 플래그들에 대한 값들에 기초하여 현재의 서브 블록의 변환 계수를 디코딩하기 위한 콘텍스트를 결정하며, 결정된 콘텍스트를 사용하여 변환 계수를 엔트로피 디코딩하도록 구성된 비디오 디코더를 포함한다.

[0022] 또 다른 예에서, 비디오 데이터를 디코딩하기 위한 디바이스는, 현재의 서브 블록에 대한 하나 이상의 이웃하는 서브 블록들의 코딩된 서브 블록 플래그들에 대한 값들을 결정하기 위한 수단, 코딩된 서브 블록 플래그들에 대한 값들에 기초하여 현재의 서브 블록의 변환 계수를 디코딩하기 위한 콘텍스트를 결정하기 위한 수단, 및 결정된 콘텍스트를 사용하여 변환 계수를 엔트로피 디코딩하기 위한 수단을 포함한다.

[0023] 또 다른 예에서, 컴퓨터 판독가능 저장 매체는, 실행되는 경우 프로세서로 하여금, 현재의 서브 블록에 대한 하나 이상의 이웃하는 서브 블록들의 코딩된 서브 블록 플래그들에 대한 값들을 결정하고, 코딩된 서브 블록 플래그들에 대한 값들에 기초하여 현재의 서브 블록의 변환 계수를 디코딩하기 위한 콘텍스트를 결정하며, 결정된 콘텍스트를 사용하여 변환 계수를 엔트로피 디코딩하게 하는 명령들을 저장하고 있다.

[0024] 또 다른 예에서, 비디오 데이터를 인코딩하는 방법은, 현재의 서브 블록에 대한 하나 이상의 이웃하는 서브 블록들의 코딩된 서브 블록 플래그들에 대한 값들을 결정하는 단계, 코딩된 서브 블록 플래그들에 대한 값들에 기초하여 현재의 서브 블록의 변환 계수를 인코딩하는 콘텍스트를 결정하는 단계, 및 결정된 콘텍스트를 사용하여 변환 계수를 엔트로피 인코딩하는 단계를 포함한다.

[0025] 또 다른 예에서, 비디오 데이터를 인코딩하기 위한 디바이스는, 현재의 서브 블록에 대한 하나 이상의 이웃하는 서브 블록들의 코딩된 서브 블록 플래그들에 대한 값들을 결정하고, 코딩된 서브 블록 플래그들에 대한 값들에 기초하여 현재의 서브 블록의 변환 계수를 인코딩하는 콘텍스트를 결정하며, 결정된 콘텍스트를 사용하여 변환 계수를 엔트로피 인코딩하도록 구성된 비디오 인코더를 포함한다.

[0026] 또 다른 예에서, 비디오 데이터를 인코딩하기 위한 디바이스는, 현재의 서브 블록에 대한 하나 이상의 이웃하는 서브 블록들의 코딩된 서브 블록 플래그들에 대한 값들을 결정하기 위한 수단, 코딩된 서브 블록 플래그들에 대한 값들에 기초하여 현재의 서브 블록의 변환 계수를 인코딩하는 콘텍스트를 결정하기 위한 수단, 및 결정된 콘텍스트를 사용하여 변환 계수를 엔트로피 인코딩하기 위한 수단을 포함한다.

[0027] 또 다른 예에서, 컴퓨터 판독가능 저장 매체는, 실행되는 경우 프로세서로 하여금, 현재의 서브 블록에 대한 하나 이상의 이웃하는 서브 블록들의 코딩된 서브 블록 플래그들에 대한 값들을 결정하고, 코딩된 서브 블록 플래그들에 대한 값들에 기초하여 현재의 서브 블록의 변환 계수를 인코딩하는 콘텍스트를 결정하며, 결정된 콘텍스트를 사용하여 변환 계수를 엔트로피 인코딩하게 하는 명령들을 저장하고 있다.

[0028] 하나 이상의 예들의 세부 사항들이 첨부도면 및 아래의 상세한 설명에서 개시된다. 다른 특성들, 목적들, 및 이점들은 설명 및 도면들로부터, 그리고 청구항들로부터 명백히 알 수 있을 것이다.

도면의 간단한 설명

[0029] 도 1 은 본 개시물에서 설명되는 인터-예측 기법들을 사용할 수도 있는 예시적인 비디오 인코딩 및 디코딩 시스템을 예시하는 블록도이다.

도 2 는 본 개시물에서 설명되는 인터-예측 기법들을 구현할 수도 있는 예시적인 비디오 인코더를 예시하는 블록도이다.

도 3 은 본 개시물에서 설명되는 인터-예측 기법들을 구현할 수도 있는 예시적인 비디오 디코더를 예시하는 블록도이다.

도 4 는 비디오 블록에서의 변환 계수들과, 비디오 블록과 연관된 유의도 맵 사이의 관계를 예시하는 개념도이다.

도 5a 내지 도 5d 는 지그-재그 스캐닝 순서, 수평 스캐닝 순서, 수직 스캐닝 순서, 및 대각선 스캐닝 순서를

사용하여 스캐닝되는 비디오 데이터의 블록들의 예들을 제시하는 개념도들이다.

도 6 은 변환 계수 코딩을 위한 서브 블록들로 분할되는 예시적인 비디오 블록을 제시하는 개념도이다.

도 7 은 역방향 대각선 스캐닝 순서를 사용하여 스캐닝되는 비디오 블록에서의 계수들의 유의도 맵에 대한 콘텍스트 모델을 정의하는데 사용되는 예시적인 5-지점 서포트를 제시하는 개념도이다.

도 8a 및 도 8b 는 5-지점 서포트 내의 콘텍스트 의존성을 제시하는 개념도들이다.

도 9a 및 도 9b 는 2 이상의 영역들로의 비디오 블록의 예시적인 분할들을 제시하는 개념도들이다.

도 10 은 비디오 블록의 각각의 영역에 대한 이웃- 또는 포지션-기반 콘텍스트들의 예시적인 할당을 제시하는 개념도이다.

도 11 은 비디오 블록의 각각의 영역에 대한 콘텍스트 오프셋들의 예시적인 할당을 제시하는 개념도이다.

도 12 는 기존 콘텍스트 모델들과 연관되는 TU 사이즈들에 기초한, 2 이상의 영역들로의 비디오 블록의 예시적인 임베딩되는 분할 (embedded division) 을 제시하는 개념도이다.

도 13a 및 도 13b 는 2 이상의 영역들로의 비디오 블록의 예시적인 분할들을 제시하는 개념도들이다.

도 14a 및 도 14b 는 비디오 블록의 각각의 영역에 대한 콘텍스트 오프셋들의 예시적인 할당을 제시하는 개념도들이다.

도 15 는 현재의 블록을 인코딩하는 예시적인 방법을 제시하는 플로우차트이다.

도 16 은 비디오 데이터의 현재의 블록을 디코딩하는 예시적인 방법을 제시하는 플로우차트이다.

발명을 실시하기 위한 구체적인 내용

[0030]

일반적으로, 본 개시물은 예컨대, 콘텍스트-적용 바이너리 산술 코딩 (CABAC) 을 사용하여, 엔트로피 코딩하기 위한 비디오 데이터의 콘텍스트들을 결정하는 것과 관련된 기법들을 기술한다. CABAC 코딩은 일반적으로 여러 구문 (syntax) 엘리먼트들의 이진화된 표현들을 코딩할 때 콘텍스트를 결정하는 것을 수반한다. 구문 엘리먼트들은 예를 들어, 변환 계수들이 유의한지 여부를 나타내는 데이터, 유의한 변환 계수들의 부호들, 및 유의한 변환 계수들에 대한 레벨 값들과 같은, 변환 계수들에 대한 데이터를 포함한다. 변환 계수들은 일반적으로 변환 유닛 (TU) 과 같은, 변환 블록의 계수들에 대응한다. 본 개시물은 변환 계수들이 발생하는 변환 블록의 영역들에 기초하여, 변환 계수들을 코딩하기 위한 콘텍스트들을 결정하는 기법들을 기술한다.

[0031]

일반적으로, 본 개시물의 기법들에 따르면, 비디오 코더는 변환 계수가 발생하는 영역에 기초하여, 변환 계수를 코딩하기 위한 콘텍스트를 결정하고 그 후 그 결정된 콘텍스트를 사용하여 변환 계수를 엔트로피 코딩하도록 구성될 수도 있다. 비디오 블록은 영역들로 다양한 방법들로 분할될 수도 있다. 도 9a 및 도 11 은 비디오 블록이 하나 이상의 상부-좌측 서브 블록들 (예컨대, 4×4 서브 블록들) 을 포함하는 제 1 영역, 및 제 1 영역 외부의 서브 블록들을 포함하는 제 2 영역으로 분할되는 예들을 제시한다. 도 9b 는 비디오 블록이 대각선 방향을 따라서 영역들로 분할되는 예를 제시한다. 도 10 은 비디오 블록이 4분위수들 (quartiles) 로 분할되고, 상부-좌측 4분위수가 상부-좌측 4분위수의 상부-좌측 부분의 서브 블록들을 포함하는 제 1 서브-영역, 및 제 1 서브-영역 외부의 상부-좌측 4분위수의 서브 블록들을 포함하는 제 2 서브-영역으로 추가로 분할되는 예를 제시한다. 도 12 는 비디오 블록이 비디오 블록 사이즈들 (예컨대, 4×4, 8×8, 16×16, 및 32×32) 에 대응하는 영역들로 분할되는 예를 제시한다. 도 13a 는 비디오 블록이 수평 직사각형의 영역들로 분할되는 예를 제시한다. 도 13b 는 비디오 블록이 수직 직사각형의 영역들로 분할되는 예를 제시한다. 이들 도면들은 아래에서 더욱더 자세하게 설명된다.

[0032]

여러 예들에서, 비디오 코더는 변환 계수들을 코딩하기 위한 콘텍스트를 여러 방법들로, 예컨대, 변환 계수가 발생하는 영역에 기초하여 결정하도록 구성될 수도 있다. 예를 들어, 비디오 코더는 일부 영역들에 대한 포지션-기반 콘텍스트 정보 또는 다른 영역들에 대한 이웃-기반 콘텍스트 정보를 사용하여 콘텍스트를 결정하도록 구성될 수도 있다. 일부 예들에서, 특정의 영역 내의 모든 변환 계수들은 그 영역에 기초하여 결정된, 동일한 콘텍스트를 사용하여 코딩될 수도 있다. 다른 예들에서, 영역 내의 변환 계수들에 대한 콘텍스트들은 콘텍스트 이웃에 기초하여 결정될 수도 있다. 또한 다른 예들에서, 비디오 코더는 변환 계수가 발생하는 영역에 기초하여 콘텍스트에 적용된 오프셋을 결정할 수도 있다. 즉, 영역들 각각은 콘텍스트에

적용되는 특성의 컨텍스트 오프셋과 연관될 수도 있다.

[0033] 본 개시물의 기법들은 대역폭 소비를 감소시켜, 변환 계수들에 대한 구문 엘리먼트들을 코딩할 때 비트들의 절감들을 초래할 수도 있다. 이러한 구문 엘리먼트들은 (대응하는 변환 계수가 유의한, 즉, 난-제로인지 여부를 나타내는) 유의 계수 플래그, 유의 계수들의 부호, 유의 계수가 1 보다 큰 절대값을 갖는지 여부의 표시, 1 보다 큰 절대값을 가진 유의 계수가 2 보다 큰 절대값을 갖는지 여부의 표시, 및/또는 2 보다 큰 절대값들을 갖는 계수들에 대한 나머지 레벨 값 중 임의의 것 또는 모두를 포함할 수도 있다.

[0034] 도 1 은 본 개시물에서 설명하는 기법들을 활용할 수도 있는 예시적인 비디오 인코딩 및 디코딩 시스템 (10) 을 예시하는 블록도이다. 도 1 에 나타낸 바와 같이, 시스템 (10) 은 목적지 디바이스 (14) 에 의해 추후 디코딩될 인코딩된 비디오 데이터를 생성하는 소스 디바이스 (12) 를 포함한다. 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 데스크탑 컴퓨터들, 노트북 (즉, 랩탑) 컴퓨터들, 태블릿 컴퓨터들, 셋-탑 박스들, 소위 "스마트" 폰들과 같은 전화기 핸드셋들, 소위 "스마트" 패드들, 텔레비전, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 콘솔들, 비디오 스트리밍 디바이스 등을 포함한, 광범위한 디바이스들 중 임의의 디바이스를 포함할 수도 있다. 일부의 경우, 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 무선 통신용으로 탑재될 수도 있다.

[0035] 목적지 디바이스 (14) 는 디코딩될 인코딩된 비디오 데이터를 링크 (16) 를 통해서 수신할 수도 있다. 링크 (16) 는 인코딩된 비디오 데이터를 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 이동시킬 수 있는 임의의 유형의 매체 또는 디바이스를 포함할 수도 있다. 일 예에서, 링크 (16) 는 소스 디바이스 (12) 로 하여금 인코딩된 비디오 데이터를 직접 목적지 디바이스 (14) 로 실시간으로 송신할 수 있게 하는 통신 매체를 포함할 수도 있다. 인코딩된 비디오 데이터는 무선 통신 프로토콜과 같은 통신 표준에 따라서 변조되어 목적지 디바이스 (14) 로 송신될 수도 있다. 통신 매체는 무선 주파수 (RF) 스펙트럼 또는 하나 이상의 물리적인 송신 라인들과 같은, 임의의 무선 또는 유선 통신 매체를 포함할 수도 있다. 통신 매체는 근거리 네트워크, 광역 네트워크, 또는 글로벌 네트워크, 예컨대 인터넷과 같은 패킷-기반 네트워크의 일부를 형성할 수도 있다. 통신 매체는 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 통신을 용이하게 하는데 유용할 수도 있는 임의의 다른 장비를 포함할 수도 있다.

[0036] 이의 대안으로, 인코딩된 데이터는 출력 인터페이스 (22) 로부터 저장 디바이스 (34) 로 출력될 수도 있다. 이와 유사하게, 인코딩된 데이터는 입력 인터페이스에 의해 저장 디바이스 (34) 로부터 액세스될 수도 있다. 저장 디바이스 (34) 는 하드 드라이브, 블루-레이 디스크들, DVD들, CD-ROM들, 플래시 메모리, 휘발성 또는 비-휘발성 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 임의의 다른 적합한 디지털 저장 매체들과 같은 다양한 분산된 또는 로컬 액세스되는 데이터 저장 매체들 중 임의의 데이터 저장 매체를 포함할 수도 있다. 추가 예에서, 저장 디바이스 (34) 는 소스 디바이스 (12) 에 의해 생성된 인코딩된 비디오를 유지할 수도 있는 파일 서버 또는 또 다른 중간 저장 디바이스에 대응할 수도 있다. 목적지 디바이스 (14) 는 저장된 비디오 데이터에 저장 디바이스 (34) 로부터 스트리밍 또는 다운로드를 통해서 액세스할 수도 있다. 파일 서버는 인코딩된 비디오 데이터를 저장하고 그 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로 송신하는 것이 가능한 임의 종류의 서버일 수도 있다. 예시적인 파일 서버들은 웹 서버 (예컨대, 웹사이트용), FTP 서버, NAS (network attached storage) 디바이스들, 또는 로컬 디스크 드라이브를 포함한다. 목적지 디바이스 (14) 는 인터넷 접속을 포함한, 임의의 표준 데이터 접속을 통해서, 인코딩된 비디오 데이터에 액세스할 수도 있다. 이것은 파일 서버 상에 저장된 인코딩된 비디오 데이터에 액세스하는데 적합한, 무선 채널 (예컨대, Wi-Fi 접속), 유선 접속 (예컨대, DSL, 케이블 모뎀 등), 또는 양자의 조합을 포함할 수도 있다. 저장 디바이스 (34) 로부터의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 이 양자의 조합일 수도 있다.

[0037] 본 개시물의 기법들은 반드시 무선 애플리케이션들 또는 설정들에 한정되지는 않는다. 이 기법들은 공중 경유 (over-the-air) 텔레비전 브로드캐스트들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, 예컨대, 인터넷을 통한 스트리밍 비디오 송신들, 데이터 저장 매체 상에의 저장을 위한 디지털 비디오의 인코딩, 데이터 저장 매체 상에 저장된 디지털 비디오의 디코딩, 또는 다른 애플리케이션들과 같은, 다양한 멀티미디어 애플리케이션들 중 임의의 애플리케이션의 지원 하에 비디오 코딩에 적용될 수도 있다. 일부 예들에서, 시스템 (10) 은 비디오 스트리밍, 비디오 플레이어백, 비디오 브로드캐스팅, 및/또는 비디오 전화 통신과 같은, 지원 애플리케이션들로의 1-방향 또는 2-방향 비디오 송신을 지원하도록 구성될 수도 있다.

[0038] 도 1 의 예에서, 소스 디바이스 (12) 는 비디오 소스 (18), 비디오 인코더 (20) 및 출력 인터페이스 (22) 를 포함한다. 일부의 경우, 출력 인터페이스 (22) 는 변조기/복조기 (모뎀) 및/또는 송신기를 포함할 수도

있다. 소스 디바이스 (12) 에서, 비디오 소스 (18) 는 비디오 캡처 디바이스, 예컨대, 비디오 카메라, 이전에 캡처된 비디오를 포함하는 비디오 아카이브, 비디오 콘텐츠 제공자로부터 비디오를 수신하는 비디오 공급 인터페이스, 및/또는 컴퓨터 그래픽스 데이터를 소스 비디오로서 발생하는 컴퓨터 그래픽스 시스템과 같은 소스, 또는 이러한 소스들의 조합을 포함할 수도 있다. 일 예로서, 비디오 소스 (18) 가 비디오 카메라이면, 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 소위 카메라 폰들 또는 비디오 폰들을 형성할 수도 있다. 그러나, 본 개시물에서 설명하는 기법들은 비디오 코딩에 일반적으로 적용가능할 수도 있으며, 무선 및/또는 유선 애플리케이션들에 적용될 수도 있다.

[0039] 캡처되거나, 사전-캡처되거나, 또는 컴퓨터-생성된 비디오는 비디오 인코더 (20) 에 의해 인코딩될 수도 있다. 인코딩된 비디오 데이터는 소스 디바이스 (12) 의 출력 인터페이스 (22) 를 통해서 목적지 디바이스 (14) 로 바로 송신될 수도 있다. 인코딩된 비디오 데이터는 또한 (또는, 대안적으로) 디코딩 및/또는 플레이백을 위한, 목적지 디바이스 (14) 또는 다른 디바이스들에 의한 추후 액세스를 위해, 저장 디바이스 (34) 상에 저장될 수도 있다.

[0040] 목적지 디바이스 (14) 는 입력 인터페이스 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (32) 를 포함한다. 일부의 경우, 입력 인터페이스 (28) 는 수신기 및/또는 모뎀을 포함할 수도 있다. 목적지 디바이스 (14) 의 입력 인터페이스 (28) 는 인코딩된 비디오 데이터를 링크 (16) 를 통해서 수신한다. 링크 (16) 를 통해서 통신되거나, 또는 저장 디바이스 (34) 상에 제공되는 인코딩된 비디오 데이터는, 비디오 데이터를 디코딩할 때에, 비디오 디코더 (30) 와 같은 비디오 디코더에 의해 사용하기 위한, 비디오 인코더 (20) 에 의해 생성되는 다양한 구문 엘리먼트들을 포함할 수도 있다. 이러한 구문 엘리먼트들은 통신 매체 상으로 송신되거나, 저장 매체 상에 저장되거나, 또는 파일 서버에 저장된 인코딩된 비디오 데이터와 함께 포함될 수도 있다.

[0041] 디스플레이 디바이스 (32) 는 목적지 디바이스 (14) 와 통합되거나 또는 그 외부에 있을 수도 있다. 일부 예들에서, 목적지 디바이스 (14) 는 통합된 디스플레이 디바이스를 포함하며, 또한 외부 디스플레이 디바이스와 인터페이스하도록 구성될 수도 있다. 다른 예들에서, 목적지 디바이스 (14) 는 디스플레이 디바이스일 수도 있다. 일반적으로, 디스플레이 디바이스 (32) 는 그 디코딩된 비디오 데이터를 사용자에게 디스플레이 하며, 액정 디스플레이 (LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (OLED) 디스플레이, 또는 또 다른 유형의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들 중 임의의 디바이스를 포함할 수도 있다.

[0042] 비디오 인코더 (20) 및 비디오 디코더 (30) 는 현재 개발중인 HEVC (High Efficiency Video Coding) 표준과 같은 비디오 압축 표준에 따라서 동작할 수도 있으며, HEVC 테스트 모델 (HM) 에 따를 수도 있다. 이의 대안으로, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 MPEG 4, 파트 10, AVC (Advanced Video Coding) 으로서 대안적으로 지칭되는 ITU-T H.264 표준, 또는 이러한 표준들의 확장판들과 같은, 다른 사유 또는 산업 표준들에 따라서 동작할 수도 있다. 표준들의 확장판들은 예를 들어, 스케일러블 비디오 코딩 (SVC), 멀티뷰 비디오 코딩 (MVC), 심도 정보를 코딩하는 것과 같은 3차원 (3D) 코딩 등을 포함한다. 본 개시물의 기법들은, 그러나, 임의의 특정의 코딩 표준 또는 표준 확장판에 한정되지 않는다. 비디오 압축 표준들의 다른 예들은 MPEG-2 및 ITU-T H.263 을 포함한다.

[0043] 도 1 에 나타내지는 않지만, 일부 양태들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 오디오 인코더 및 디코더와 각각 통합될 수도 있으며, 오디오와 비디오 양자의 인코딩을 공통 데이터 스트림 또는 별개의 데이터 스트림들로 처리하기 위해 적합한 MUX-DEMUX 유닛들, 또는 다른 하드웨어 및 소프트웨어를 포함할 수도 있다. 적용가능한 경우, 일부 예들에서, MUX-DEMUX 유닛들은 ITU H.223 멀티플렉서 프로토콜, 또는 다른 프로토콜들, 예컨대, 사용자 데이터그램 프로토콜 (UDP) 을 따를 수도 있다.

[0044] 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들 (DSP들), 주문형 집적회로들 (ASIC들), 필드 프로그래밍가능 게이트 어레이들 (FPGA들), 이산 로직, 소프트웨어, 하드웨어, 펌웨어 또는 임의의 이들의 조합들과 같은, 다양한 적합한 인코더 회로 중 임의의 회로로 구현될 수도 있다. 이 기법들이 소프트웨어로 부분적으로 구현되는 경우, 디바이스는 본 개시물의 기법들을 수행하기 위해 소프트웨어용 명령들을 적합한 비일시성 컴퓨터 판독가능 매체에 저장하고, 그 명령들을 하드웨어에서 하나 이상의 프로세서들을 사용하여 실행할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 각각은 하나 이상의 인코더들 또는 디코더들에 포함될 수도 있으며, 이들 중 어느 쪽이든 각각 디바이스에서 결합된 인코더/디코더 (코덱) 의 일부로서 통합될 수도 있다.

[0045] JCT-VC 는 HEVC 표준의 개발에 노력을 들이고 있다. HEVC 표준화 노력들은 HEVC 테스트 모델 (HM) 로서

지칭되는 비디오 코딩 디바이스의 진화 모델 (evolving model) 에 기초한다. HM 은 예컨대, ITU-T H.264/AVC 에 따른 기존 디바이스들에 관련된 비디오 코딩 디바이스들의 여러 추가적인 능력들을 가정한다. 예를 들어, H.264 는 9개의 인트라-예측 인코딩 모드들을 제공하는 반면, HM 은 33개 만큼이나 많은 인트라-예측 인코딩 모드들을 제공할 수도 있다

[0046] 일반적으로, HM 의 작업 모델은 비디오 프레임 또는 화상이 루마 샘플 및 크로마 샘플 양자를 포함하는 트리블록들 또는 최대 코딩 유닛들 (LCU) 의 시퀀스로 분할될 수도 있다고 기술한다. 트리블록은 H.264 표준의 매크로블록과 유사한 목적을 갖는다. 슬라이스는 코딩 순서에서 다수의 연속되는 트리블록들을 포함한다. 비디오 프레임 또는 화상은 하나 이상의 슬라이스들로 파티셔닝될 수도 있다. 각각의 트리블록은 쿼드트리에 따라 코딩 유닛들 (CU들) 로 분할될 수도 있다. 예를 들어, 트리블록은, 쿼드트리의 루트 노드 처럼, 4개의 자식 노드들로 분할될 수도 있으며, 각각의 자식 노드는 결국 부모 노드일 수도 있으며 또 다른 4개의 자식 노드들로 분할될 수도 있다. 쿼드트리의 리프 노드와 같은, 최종, 미분할된 자식 노드는 코딩 노드, 즉, 코딩된 비디오 블록을 포함한다. 코딩된 비트스트림과 연관되는 구문 데이터는 최대 힛수를 정의할 수도 있으며 트리블록이 분할될 수도 있으며, 또한 코딩 노드들의 최소 사이즈를 정의할 수도 있다.

[0047] CU 는 코딩 노드, 및 이 코딩 노드와 연관되는 변환 유닛들 (TU들) 및 예측 유닛들 (PU들) 을 포함한다. CU 의 사이즈는 코딩 노드의 사이즈에 대응하며 정사각형 형태이어야 한다. CU 의 사이즈는 8×8 픽셀들로부터 64×64 픽셀들 이상의 최대치를 갖는 트리블록의 사이즈까지 이를 수도 있다. 각각의 CU 는 하나 이상의 PU들 및 하나 이상의 TU들을 포함할 수도 있다. CU 와 연관되는 구문 데이터는 예를 들어, 하나 이상의 PU들로의 CU 의 파티셔닝을 기술할 수도 있다. 파티셔닝 모드들은 CU 가 스킵되는지 또는 직접 모드 인코딩될지, 인트라-예측 모드 인코딩될지, 또는 인터-예측 모드 인코딩될지 여부의 사이에 상이할 수도 있다. PU들은 비-정사각형의 형태로 파티셔닝될 수도 있다. CU 와 연관되는 구문 데이터는 또한 예를 들어, 쿼드트리에 따른 하나 이상의 TU들로의 CU 의 파티셔닝을 기술할 수도 있다. TU 는 정사각형 또는 비-정사각형의 형태일 수 있다.

[0048] HEVC 표준은 TU들에 따라서 변환들을 허용하며, 이 TU들은 상이한 CU들 에 대해 상이할 수도 있다. TU들은 일반적으로 파티셔닝된 LCU 에 대해 정의된 주어진 CU 내의 PU들의 사이즈에 기초하여 사이징되지만, 이것이 항상 그런 것은 아니다. TU들은 일반적으로 PU들과 동일한 사이즈이거나 또는 그보다 작다. 일부 예들에서, CU 에 대응하는 잔여 샘플들은 "잔여 쿼드 트리" (RQT) 로서 알려진 쿼드트리 구조를 사용하여 더 작은 유닛들로 세분될 수도 있다. RQT 의 리프 노드들은 변환 유닛들 (TU들) 로서 지칭될 수도 있다. TU들과 연관되는 픽셀 차이 값들은 변환 계수들을 생성하기 위해 변환될 수도 있으며, 그 변환 계수들은 양자화될 수도 있다.

[0049] 일반적으로, PU 는 예측 프로세스에 관련되는 데이터를 포함한다. 예를 들어, PU 가 인트라-모드 인코딩 될 때, PU 는 그 PU 에 대한 인트라-예측 모드를 기술하는 데이터를 포함할 수도 있다. 또 다른 예로서, PU 가 인터-모드 인코딩 될 때, PU 는 그 PU 에 대한 모션 벡터를 정의하는 데이터를 포함할 수도 있다. PU 에 대한 모션 벡터를 정의하는 데이터는 예를 들어, 모션 벡터의 수평 성분, 모션 벡터의 수직 성분, 모션 벡터에 대한 해상도 (예컨대, 1/4 픽셀 정밀도 또는 1/8 픽셀 정밀도), 모션 벡터가 가리키는 참조 화상, 및/또는 모션 벡터에 대한 참조 화상 리스트를 기술할 수도 있다.

[0050] 일반적으로, TU 는 변환 및 양자화 프로세스들에 사용된다. 하나 이상의 PU들을 갖는 주어진 CU 는 또한 하나 이상의 TU들을 포함할 수도 있다. 예측 이후, 비디오 인코더 (20) 는 PU 에 대응하는 잔여 값들을 계산할 수도 있다. 잔여 값들은 엔트로피 코딩을 위한 직렬화된 변환 계수들을 생성하기 위해 변환 계수들로 변환되고, 양자화되며, TU들을 사용하여 스캐닝될 수도 있는 픽셀 차이 값들을 포함한다. 본 개시물은 일반적으로 CU 의 코딩 노드를 지칭하기 위해 용어 "비디오 블록" 을 사용한다. 일부 특정의 경우들에서, 본 개시물은 또한 트리블록, 즉, LCU, 또는 코딩 노드 및 PU들 및 TU들을 포함하는 CU 를 지칭하기 위해 용어 "비디오 블록" 을 사용할 수도 있다.

[0051] 비디오 시퀀스는 일반적으로 비디오 프레임들 또는 화상들의 시리즈를 포함한다. 화상들의 그룹 (GOP) 은 일반적으로 일련의 하나 이상의 비디오 화상들을 포함한다. GOP 는 GOP 의 헤더, 화상들의 하나 이상의 헤더, 또는 다른 곳에, GOP 에 포함된 다수의 화상들을 기술하는 구문 데이터를 포함할 수도 있다. 화상의 각각의 슬라이스는 각각의 슬라이스에 대한 인코딩 모드를 기술하는 슬라이스 구문 데이터를 포함할 수도 있다. 비디오 인코더 (20) 는 일반적으로 비디오 데이터를 인코딩하기 위해 개개의 비디오 슬라이스들 내의 비디오 블록들에 대해 동작한다. 비디오 블록은 CU 내의 코딩 노드에 대응할 수도 있다. 비디오

블록들은 고정 또는 가변 사이즈들을 가질 수도 있으며, 규정된 코딩 표준에 따라서 사이즈가 상이할 수도 있다.

[0052] 일 예로서, HM 은 여러 PU 사이즈들에서 예측을 지원한다. 특정의 CU 의 사이즈가 $2N \times 2N$ 이라고 가정하면, HM 은 $2N \times 2N$ 또는 $N \times N$ 의 PU 사이즈들에서 인트라-예측을, 그리고 $2N \times 2N$, $2N \times N$, $N \times 2N$, 또는 $N \times N$ 의 대칭적인 PU 사이즈들에서 인터-예측을 지원한다. HM 은 또한 $2N \times nU$, $2N \times nD$, $nL \times 2N$, 및 $nR \times 2N$ 의 PU 사이즈들에서의 인터-예측에 대해 비대칭적인 파티셔닝을 지원한다. 비대칭적인 파티셔닝에서, CU 의 하나의 방향은 파티셔닝되지 않지만, 다른 방향은 25% 및 75% 로 파티셔닝된다. 25% 파티션에 대응하는 CU 의 부분은 "상부 (Up)", "하부 (Down)", "좌측 (Left)", 또는 "우측 (Right)" 의 표시가 뒤따르는 "n" 으로 표시된다. 따라서, 예를 들어, " $2N \times nU$ " 는 최상부에서 $2N \times 0.5N$ PU 로 그리고 최저부에서 $2N \times 1.5N$ PU 로 수평으로 파티셔닝된 $2N \times 2N$ CU 를 지칭한다.

[0053] 본 개시물에서, " $N \times N$ " 및 "N 곱하기 N" 은 수직 및 수평 치수들의 관점에서 비디오 블록의 픽셀 치수들, 예컨대, 16×16 픽셀들 또는 16 곱하기 16 픽셀들을 지칭하기 위해 상호교환가능하게 사용될 수도 있다. 일반적으로, 16×16 블록은 수직 방향으로 16개의 픽셀들 ($y = 16$) 및 수평 방향으로 16개의 픽셀들 ($x = 16$) 을 가질 것이다. 이와 유사하게, $N \times N$ 블록은 일반적으로 수직 방향으로 N 개의 픽셀들 및 수평 방향으로 N 개의 픽셀들을 가지며, 여기서 N 은 음이 아닌 정수 값을 나타낸다. 블록에서 픽셀들은 로우들 및 칼럼들로 배열될 수도 있다. 더욱이, 블록들은 수직 방향에서와 같이 수평 방향에서 동일한 픽셀들의 개수를 반드시 가질 필요는 없다. 예를 들어, 블록들은 $N \times M$ 픽셀들을 포함할 수도 있으며, 여기서 M 은 반드시 N 과 같을 필요는 없다.

[0054] CU 의 PU들을 사용한 인트라-예측 또는 인터-예측 코딩 이후, 비디오 인코더 (20) 는 CU 의 TU들에 대한 잔여 데이터를 계산할 수도 있다. PU들은 공간 도메인 (또한, 픽셀 도메인으로 지칭됨) 에서 픽셀 데이터를 포함할 수도 있으며, TU들은 이산 코사인 변환 (DCT), 정수 변환, 웨이블릿 변환, 또는 잔여 비디오 데이터에 개념적으로 유사한 변환과 같은 변환의 적용 이후 변환 도메인에서 계수들을 포함할 수도 있다. 잔여 데이터는 인코딩되지 않은 화상의 픽셀들과 PU들에 대응하는 예측 값들 사이의 픽셀 차이들에 대응할 수도 있다. 비디오 인코더 (20) 는 CU 에 대한 잔여 데이터를 포함하는 TU들을 형성하고, 그 후 그 TU들을 변환하여, 그 CU 에 대한 변환 계수들을 생성할 수도 있다.

[0055] 변환 계수들을 생성하는 임의의 변환들 이후, 비디오 인코더 (20) 는 변환 계수들의 양자화를 수행할 수도 있다. 양자화는 일반적으로 계수들을 나타내는데 사용되는 데이터의 양을 가능한 한 감축하기 위해 변환 계수들이 양자화되는 프로세스를 지칭하며, 추가적인 압축을 제공한다. 양자화 프로세스는 그 계수들의 일부 또는 모두와 연관되는 비트 심도를 감소시킬 수도 있다. 예를 들어, n-비트 값은 양자화 동안 m-비트 값까지 절사될 수도 있으며, 여기서, n 은 m 보다 더 크다.

[0056] 일부 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 엔트로피 인코딩될 수 있는 직렬화된 벡터를 생성하기 위해, 미리 정의된 스캐닝 순서를 사용하여, 양자화된 변환 계수들을 스캐닝할 수도 있다. 다른 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 적응적 스캐닝을 수행할 수도 있다. 양자화된 변환 계수들을 스캐닝하여 1차원 벡터를 형성한 후, 또는 스캐닝 동안, 비디오 인코더 (20) 는 1차원 벡터를 예컨대, 컨텍스트 적응 가변 길이 코딩 (CAVLC), 컨텍스트 적응 바이너리 산술 코딩 (CABAC), 구문-기반 컨텍스트-적용 바이너리 산술 코딩 (SBAC), 확률 간격 파티셔닝 엔트로피 (PIPE) 코딩 또는 또 다른 엔트로피 인코딩 방법론에 따라서, 엔트로피 인코딩할 수도 있다. 비디오 디코더 (30) 는 계수들을 엔트로피 디코딩하고, 역양자화 프로세스 및 역변환 프로세스를 수행하여 잔여 데이터를 재생하며, 잔여 데이터와 예측 데이터를 합산하여 디코딩된 비디오 데이터를 생성할 수도 있다. 비디오 인코더 (20) 는 또한 비디오 데이터를 디코딩할 때에 비디오 디코더 (30) 에 의해 사용하기 위한 인코딩된 비디오 데이터와 연관되는 구문 엘리먼트들을 엔트로피 인코딩할 수도 있다.

[0057] CABAC 을 수행하기 위해, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 컨텍스트 모델 내의 컨텍스트를 코딩되는 심볼에 할당할 수도 있다. 컨텍스트는 예를 들어, 심볼의 이웃하는 값들이 1-제로인지 여부에 관련될 수도 있다. 본 개시물의 기법들에 따르면, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 변환 계수가 발생하는 비디오 블록의 영역에 기초하여, 변환 계수를 엔트로피 코딩 (예컨대, 엔트로피 인코딩 또는 엔트로피 디코딩) 하기 위한 컨텍스트를 결정하도록 구성될 수도 있다.

[0058] 비디오 인코더 (20) 및 비디오 디코더 (30) 는 비디오 블록들 (예컨대, 변환 유닛들) 에 대한 여러 영역들의 정의들로 구성될 수도 있다. 예를 들어, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 비디오 블록들의

여러 사이즈들에 대한 영역들의 정의들로 구성될 수도 있다. 일부 예들에서, 비디오 인코더 (20) 는 비디오 블록을 영역들로 분할하는 방법을 결정하고, 블록이 어떻게 분할되는지를 나타내는 데이터를 코딩할 수도 있다. 영역들 각각은 각각의 영역 내에서 발생하는 변환 계수들에 대한 컨텍스트를 결정하기 위한, 각각의 값 및/또는 기법과 연관될 수도 있다.

[0059] 예를 들어, 비디오 블록의 특정의 영역은 이웃-기반 컨텍스트 결정 방식과 연관될 수도 있지만, 비디오 블록의 또 다른 영역은 포지션-기반 컨텍스트 결정 방식과 연관될 수도 있다. 또 다른 예로서, 비디오 블록의 영역은 그 영역에 위치한 변환 계수들에 대해 결정된 컨텍스트에 적용된 오프셋과 연관될 수도 있다. 동일한 비디오 블록의 상이한 영역들은 컨텍스트를 계산하기 위한, 상이한 오프셋 값들 및/또는 상이한 기법들과 연관될 수도 있다.

[0060] 일 예로서, 비디오 블록은 2개의 상이한 영역들, 즉, 비디오 블록의 상부-좌측 모서리에서 하나 이상의 서브 블록들 (예컨대, 4×4 변환 계수 서브 블록들) 을 포함하는 제 1 영역, 및 제 1 영역에 포함되지 않는 비디오 블록의 다른 서브 블록들을 포함하는 제 2 영역을 포함할 수도 있다. 좀더 구체적으로는, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 서브 블록의 x- 및 y-좌표를 결정하고, x 와 y 의 총합을 임계값과 비교함으로써, 서브 블록이 제 1 영역 또는 제 2 영역에 있는지 여부를 결정할 수도 있다. x 와 y 의 총합이 임계치 미만이면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 서브 블록이 제 1 영역에 있다고 결정할 수도 있으며, 그렇지 않으면, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 서브 블록이 제 2 영역에 있다고 결정할 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 계수들이 제 1 영역의 서브 블록 또는 제 2 영역의 서브 블록에 있는지 여부에 기초하여, 비디오 블록의 계수들에 대한 컨텍스트를 결정할 수도 있다.

[0061] 예를 들어, 일부 영역들에서, 컨텍스트는 고정된 컨텍스트일 수도 있으며, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 그 고정된 컨텍스트를 사용하여 이러한 영역들에서의 변환 계수들을 코딩한다. 즉, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 동일한 컨텍스트를 그 영역에서의 모든 변환 계수들에 적용할 수도 있다. 이의 대안으로, 그 영역 내의 서브 블록들 각각은 컨텍스트를 결정하는 동일한 방법 (예컨대, 고정된 컨텍스트 방법) 과 연관될 수도 있지만, 그 영역에서의 상이한 서브 블록들은 상이한 고정된 컨텍스트들을 가질 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 그 영역에서의 서브 블록의 포지션에 기초하여 서브 블록에 대한 고정된 컨텍스트를 결정할 수도 있다. 또한, 또 다른 예로서, 고정된 컨텍스트들은 그 영역 내의 개개의 변환 계수 포지션들에 할당될 수도 있다. 즉, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 비디오 블록, 서브 블록, 및/또는 영역 내의 변환 계수의 포지션에 기초하여, 영역 내의 변환 계수를 코딩하기 위한 컨텍스트를 결정할 수도 있다.

[0062] 또 다른 예로서, 일부 영역들에서, 컨텍스트 모델은 이웃하는 서브 블록들에 따라서 정의될 수도 있다. 예를 들어, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 특정의 영역 내의 각각의 서브 블록에 대한 컨텍스트들의 세트들로 구성될 수도 있다. 즉, 그 영역에서의 각각의 서브 블록은 컨텍스트들의 각각의 세트와 연관될 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 는 각각의 서브 블록에서의 각각의 변환 계수에 대한 컨텍스트들의 세트 중에서, 적합한 컨텍스트를 선택할 수도 있다. 하나의 서브 블록에 대한 컨텍스트들의 세트는 또 다른 서브 블록에 대한 컨텍스트들의 세트와는 상이할 수도 있다.

[0063] 또한, 또 다른 예로서, 대응하는 서브 블록에서 임의의 유의한 (즉, 난-제로) 계수들이 존재하는 여부를 나타내는, 그 영역에서의 각각의 서브 블록에 대한 개개의 플래그들이 코딩될 수도 있다. 이들 플래그들은 코딩된 서브 블록 플래그들로서 지칭될 수도 있다. 이러한 플래그들은 서브 블록들에서의 변환 계수들을 코딩하기 위한 컨텍스트를 선택하는데 사용될 수도 있다. 예를 들어, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 하나 이상의 이웃하는 서브 블록들의 플래그들의 값들에 기초하여, 서브 블록에서의 변환 계수들을 코딩하기 위한 컨텍스트를 결정할 수도 있다. 예를 들어, 플래그들은 0 또는 1 의 이진 값들을 가질 수도 있으며, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 우측-이웃하는 서브 블록 및 아래-이웃하는 서브 블록 (또한, 최저부-이웃하는 서브 블록으로서 지칭됨) 에 대한 플래그 값들의 총합에 기초하여, 현재의 서브 블록에서의 변환 계수들을 코딩하기 위한 컨텍스트를 결정할 수도 있다. 다른 공식들이 또한 서브 블록에 대한 컨텍스트를 계산하는데 사용될 수도 있다.

[0064] 비디오 인코더 (20) 및 비디오 디코더 (30) 는 본 개시물의 기법들 중 임의의 기법 또는 모두를, 단독으로 또는 임의의 조합으로 구현하도록 구성될 수도 있다. 이들 기법들의 하나의 예시적인 조합은, 비디오 인코더 (20) 및 비디오 디코더 (30) 가 변환 유닛을 서브 블록들 (예컨대, 4×4 픽셀 서브 블록들) 로 분할하고, 그 후 서브 블록에서의 변환 계수의 포지션에 기초하여, 그리고, 하나 이상의 이웃하는 서브 블록들, 예컨대, 좌측-이웃하는 서브 블록 및 최저부-이웃하는 서브 블록에 대한 코딩된 블록 플래그들에 기초하여, 서브 블록

의 특정의 변환 계수의 데이터를 코딩하기 위한 콘텍스트를 결정하도록 구성될 수도 있는 것이다.

[0065] 비디오 인코더 (20) 및 비디오 디코더 (30) 는 이들 여러 예들에서 결정된 콘텍스트들을 사용하여, 변환 계수들을 나타내는 하나 이상의 구문 엘리먼트들을 코딩하도록 구성될 수도 있다. 변환 계수들은 여러 유형들의 구문 엘리먼트들을 포함할 수도 있다. 예를 들어, 변환 계수는 그 변환 계수가 0-제로 값을 갖는지 (즉, 유의한지) 여부를 나타내는 유의 계수 플래그 (significant_coeff_flag) 를 포함할 수도 있다. 변환 계수가 유의하면, 변환 계수는 변환 계수의 값이 0 보다 크거나 또는 작는지 여부를 나타내는 부호 값 (예컨대, coeff_sign_flag) 및 변환 계수의 절대값이 1 보다 큰지 여부를 나타내는 값 (예컨대, coeff_abs_level_greater1_flag) 을 포함할 수도 있다. 변환 계수가 1 보다 큰 절대값을 가지면, 변환 계수는 변환 계수가 2 보다 큰 절대값을 갖는지 여부를 나타내는 값 (예컨대, coeff_abs_level_greater2_flag) 을 포함할 수도 있다. 변환 계수가 2 보다 큰 절대값을 가지면, 변환 계수는 변환 계수의 절대값 마이너스 2 를 나타내는 값 (예컨대, coeff_abs_level_remaining) 을 포함할 수도 있다.

[0066] 비디오 인코더 (20) 및 비디오 디코더 (30) 의 CABAC 코더는 본 개시물의 기법에 따라 결정된 콘텍스트들을 사용하여 이 값들 중 임의의 값 또는 모두를 코딩할 수도 있다. 추가적으로, 또는 대안적으로, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 본 개시물의 기법에 따라 결정된 콘텍스트를 사용하여, 최종 유의 계수의 포지션을 나타내는 데이터 (예컨대, last_significant_coeff_x_prefix, last_significant_coeff_x_suffix, last_significant_coeff_y_prefix, 및 last_significant_coeff_y_suffix) 를 코딩할 수도 있다.

[0067] 비디오 인코더 (20) 및 비디오 디코더 (30) 는 본 개시물에서 설명하는 기법들 중 임의의 하나 이상을, 단독으로 또는 임의의 조합으로 수행하도록 구성될 수도 있다. 변환 계수가 발생하는 비디오 블록의 영역에 기초하여 비디오 블록의 변환 계수를 코딩하기 위한 콘텍스트를 결정하며, 그 결정된 콘텍스트를 사용하여 변환 계수를 엔트로피 코딩하는 여러 기법들이 아래에 설명된다. 이러한 기법들의 예들은 아래에서 도 9 내지 도 14 와 관련하여 설명된다. 일반적으로, 결정된 콘텍스트를 사용하여 변환 계수를 코딩하는 것은 결정된 콘텍스트를 사용하여 변환 계수의 하나 이상의 구문 엘리먼트들을 코딩하는 것을 포함한다. 콘텍스트를 결정하는 것은 일반적으로 변환 계수가 발생하는 영역을 결정하는 것, 및 그 영역에 기초하여, 콘텍스트를 결정하는 것을 포함한다. 예를 들어, 영역은 특정의 콘텍스트 또는 콘텍스트들의 세트와 연관될 수도 있거나, 및/또는 콘텍스트를 결정하기 위한 하나 이상의 기법들과 연관될 수도 있다.

[0068] 도 2 는 본 개시물에서 설명되는 인터-예측 기법들을 구현할 수도 있는 예시적인 비디오 인코더 (20) 를 예시하는 블록도이다. 비디오 인코더 (20) 는 비디오 슬라이스들 내의 비디오 블록들의 인트라-코딩 및 인터-코딩을 수행할 수도 있다. 인트라-코딩은 주어진 비디오 프레임 또는 화상 내의 비디오에서 공간 리던던시를 감소시키거나 또는 제거하기 위해, 공간 예측에 의존한다. 인터-코딩은 비디오 시퀀스의 인접 프레임들 또는 화상들 내의 비디오에서 시간 리던던시를 감소시키거나 또는 제거하기 위해 시간 예측에 의존한다. 인트라-모드 (I 모드) 는 여러 공간 기반의 압축 모드들 중 임의의 압축 모드를 지칭할 수도 있다. 단방향 예측 (P 모드) 또는 양방향-예측 (B 모드) 과 같은 인터-모드들은 여러 시간-기반의 압축 모드들 중 임의의 모드를 지칭할 수도 있다.

[0069] 도 2 의 예에서, 비디오 인코더 (20) 는 모드 선택 유닛 (35), 예측 프로세서 (41), 참조 화상 메모리 (64), 합산기 (50), 변환 프로세싱 유닛 (52), 양자화 유닛 (54), 및 엔트로피 인코딩 유닛 (56) 을 포함한다. 예측 프로세서 (41) 는 모션 추정 유닛 (42), 모션 보상 유닛 (44), 및 인트라 예측 유닛 (46) 을 포함한다. 비디오 블록 재구성을 위해, 비디오 인코더 (20) 는 또한 역양자화 유닛 (58), 역변환 유닛 (60), 및 합산기 (62) 를 포함한다. 디블록킹 필터 (도 2 에 미도시) 가 또한 블록 경계들을 필터링하여 재구성된 비디오로부터 블로킹 현상 아티팩트들을 제거하기 위해 포함될 수도 있다. 원할 경우, 디블록킹 필터는 일반적으로 합산기 (62) 의 출력을 필터링할 것이다. 추가적인 루프 필터들 (인 루프 또는 포스트 루프) 이 또한 디블록킹 필터에 추가하여 사용될 수도 있다.

[0070] 도 2 에 나타낸 바와 같이, 비디오 인코더 (20) 는 비디오 데이터를 수신하고, 모드 선택 유닛 (35) 은 그 데이터를 비디오 블록들로 파티셔닝한다. 이 파티셔닝은 또한 슬라이스들, 타일들, 또는 다른 더 큰 유닛들 뿐만 아니라, 예컨대, LCU들 및 CU들의 쿼드트리 구조에 따라서 파티셔닝하는 비디오 블록으로 파티셔닝하는 것을 포함할 수도 있다. 비디오 인코더 (20) 는 일반적으로 인코딩되는 비디오 슬라이스 내의 비디오 블록들을 인코딩하는 컴포넌트들을 예시한다. 슬라이스는 다수의 비디오 블록들로 (그리고, 아마도, 타일들로서 지칭되는 비디오 블록들의 세트들로) 분할될 수도 있다. 예측 프로세서 (41) 는 여러 결과들 (예컨대, 코딩 레이트 및 왜곡의 레벨) 에 기초하여, 현재의 비디오 블록에 대해, 복수의 인트라 코딩 모드들 중 하나 또는 복수의 인터 코딩 모드들 중 하나와 같은, 복수의 가능한 코딩 모드들 중 하나를 선택할 수도

있다. 예측 프로세서 (41) 는 최종 인트라- 또는 인터-코딩된 블록을, 합산기 (50) 에 제공하여 잔여 블록 데이터를 생성하며, 합산기 (62) 에 제공하여, 참조 화상으로서 사용하기 위한 인코딩된 블록을 재구성할 수도 있다.

[0071] 예측 프로세서 (41) 내의 인트라 예측 유닛 (46) 은 코딩되는 현재의 블록과 동일한 프레임 또는 슬라이스에서의 하나 이상의 이웃하는 블록들에 대해, 현재의 비디오 블록의 인트라-예측 코딩을 수행하여, 공간 압축을 제공할 수도 있다. 예측 프로세서 (41) 내의 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 은 하나 이상의 참조 화상들에서의 하나 이상의 예측 블록들에 대해, 현재의 비디오 블록의 인터-예측 코딩을 수행하여, 시간 압축을 제공한다.

[0072] 모션 추정 유닛 (42) 은 비디오 시퀀스에 대한 미리 결정된 패턴에 따라서 비디오 슬라이스에 대한 인터-예측 모드를 결정하도록 구성될 수도 있다. 미리 결정된 패턴은 시퀀스에서의 비디오 슬라이스들을 P 슬라이스들, B 슬라이스들 또는 GPB 슬라이스들로서 지정될 수도 있다. 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 은 고도로 통합될 수도 있지만, 개념적인 목적들을 위해 별개로 예시된다. 모션 추정 유닛 (42) 에 의해 수행되는 모션 추정은 모션 벡터들을 생성하는 프로세스이며, 이 프로세스는 비디오 블록들에 대한 모션을 추정한다. 모션 벡터는, 예를 들어, 참조 화상 내의 예측 블록에 대한, 현재의 비디오 프레임 또는 화상 내의 비디오 블록의 PU 의 변위를 나타낼 수도 있다.

[0073] 예측 블록은 픽셀 차이의 관점에서 코딩되는 비디오 블록의 PU 에 가깝게 매칭하는 것으로 발견되는 블록이며, SAD (sum of absolute difference), SSD (sum of square difference), 또는 다른 차이 메트릭들에 의해 결정될 수도 있다. 일부 예들에서, 비디오 인코더 (20) 는 참조 화상 메모리 (64) 에 저장된 참조 화상들의 서브-정수 픽셀 포지션들에 대한 값들을 계산할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 참조 화상의 1/4 픽셀 포지션들, 1/8 픽셀 포지션들, 또는 다른 분수 픽셀 포지션들의 값들을 내삽할 수도 있다. 따라서, 모션 추정 유닛 (42) 은 풀 픽셀 포지션들 및 분수 픽셀 포지션들에 대해, 모션 탐색을 수행하고, 분수 픽셀 정밀도를 가진 모션 벡터를 출력할 수도 있다.

[0074] 모션 추정 유닛 (42) 은 PU 의 포지션을 참조 화상의 예측 블록의 포지션과 비교함으로써 인터-코딩된 슬라이스에서의 비디오 블록의 PU 에 대한 모션 벡터를 계산한다. 참조 화상은 제 1 참조 화상 리스트 (리스트 0) 또는 제 2 참조 화상 리스트 (리스트 1) 로부터 선택될 수도 있으며, 이 리스트 각각은 하나 이상의 참조 화상 메모리 (64) 에 저장된 참조 화상들을 식별한다. 모션 추정 유닛 (42) 은 그 계산된 모션 벡터를 엔트로피 인코딩 유닛 (56) 및 모션 보상 유닛 (44) 으로 전송한다.

[0075] 모션 보상 유닛 (44) 에 의해 수행되는 모션 보상은, 가능한 한, 서브-픽셀 정밀도까지 내삽들을 수행함으로써, 모션 추정에 의해 결정된 모션 벡터에 기초하여 예측 블록을 폐칭하거나 또는 생성하는 것을 수반할 수도 있다. 현재의 비디오 블록의 PU 에 대한 모션 벡터를 수신하자마자, 모션 보상 유닛 (44) 은 모션 벡터가 참조 화상 리스트들 중 하나에서 가리키는 예측 블록을 위치시킬 수도 있다. 모션 보상 유닛 (44) 은 또한 비디오 슬라이스의 비디오 블록들을 디코딩할 때에 비디오 디코더 (30) 에 의한 사용을 위해 비디오 블록들 및 비디오 슬라이스와 연관되는 구문 엘리먼트들을 생성할 수도 있다.

[0076] 인트라 예측 유닛 (46) 은 위에서 설명한 바와 같이, 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 에 의해 수행되는 인터-예측에 대한 대안으로서, 현재의 블록을 인트라-예측할 수도 있다. 특히, 인트라 예측 유닛 (46) 은 현재의 블록을 인코딩하는데 사용할 인트라-예측 모드를 결정할 수도 있다. 일부 예들에서, 인트라 예측 유닛 (46) 은 예컨대, 별개의 인코딩 과정들 동안 여러 인트라-예측 모드들을 사용하여 현재의 블록을 인코딩할 수도 있으며, 인트라 예측 유닛 (46) (또는, 일부 예들에서, 모드 선택 유닛 (35)) 은 테스트된 모드들로부터 사용할 적합한 인트라-예측 모드를 선택할 수도 있다. 예를 들어, 인트라 예측 유닛 (46) 은 여러 테스트된 인트라-예측 모드들에 대한 레이트-왜곡 분석을 사용하여 레이트-왜곡 값들을 계산하고, 그 테스트된 모드들 중에서 최상의 레이트-왜곡 특성들을 갖는 인트라-예측 모드를 선택할 수도 있다. 레이트-왜곡 분석은 일반적으로 인코딩된 블록과 그 인코딩된 블록을 생성하기 위해 인코딩되었던 원래의 인코딩되지 않은 블록 사이의 왜곡의 양 (또는, 에러) 뿐만 아니라, 그 인코딩된 블록을 생성하는데 사용되는 비트 레이트 (즉, 비트들의 수) 를 결정한다. 인트라 예측 유닛 (46) 은 여러 인코딩된 블록들에 대한 왜곡들 및 레이트들로부터 비율들 (비들) 을 계산하여, 어느 인트라-예측 모드가 그 블록에 대해 최상의 레이트-왜곡 값을 나타내는 지를 결정할 수도 있다.

[0077] 임의의 경우, 블록에 대한 인트라-예측 모드를 선택한 후, 인트라 예측 유닛 (46) 은 블록에 대한 그 선택된 인트라-예측 모드를 나타내는 정보를 엔트로피 인코딩 유닛 (56) 에 제공할 수도 있다. 엔트로피 인코딩

유닛 (56) 은 본 개시물의 기법에 따라서 그 선택된 인트라-예측 모드를 나타내는 정보를 인코딩할 수도 있다. 비디오 인코더 (20) 는 복수의 인트라-예측 모드 인덱스 테이블들 및 복수의 수정된 인트라-예측 모드 인덱스 테이블들 (또한, 코드워드 맵핑 테이블들로서 지칭됨) 을 포함할 수도 있는 그 송신된 비트스트림 구성 데이터에, 여러 블록들에 대한 인코딩 컨텍스트들의 정의들, 및 가장 가능성있는 인트라-예측 모드, 인트라-예측 모드 인덱스 테이블 및 컨텍스트들의 각각에 사용할 수정된 인트라-예측 모드 인덱스 테이블의 표시들을 포함할 수도 있다.

[0078] 예측 프로세서 (41) 가 현재의 비디오 블록에 대한 예측 블록을 인터-예측 또는 인트라-예측을 통해서 생성한 후, 비디오 인코더 (20) 는 현재의 비디오 블록으로부터 예측 블록을 감산함으로써 잔여 비디오 블록을 형성한다. 합산기 (50) 는 이 계산을 수행하는 유닛을 나타낸다. 잔여 블록에서의 잔여 비디오 데이터는 하나 이상의 TU들에 포함되어, 변환 프로세싱 유닛 (52) 에 적용될 수도 있다. 변환 프로세싱 유닛 (52) 은 일반적으로 잔여 비디오 데이터를 픽셀 도메인으로부터 주파수 도메인과 같은 변환 도메인으로 변환한다. 변환 프로세싱 유닛 (52) 은 이산 코사인 변환 (DCT) 또는 개념적으로 유사한 변환과 같은 변환을 사용하여 잔여 비디오 데이터를 잔여 변환 계수들로 변환할 수도 있다. 이의 대안으로, 변환 프로세싱 유닛 (52) 은 2-차원 (2-D) 변환을 (수평 및 수직 방향 양자에서) TU들에서의 잔여 데이터에 적용할 수도 있다.

[0079] 변환 프로세싱 유닛 (52) 은 최종 변환 계수들을 양자화 유닛 (54) 으로 전송할 수도 있다. 양자화 유닛 (54) 은 변환 계수들을 양자화하여, 비트 레이트를 추가로 감소시킨다. 양자화 프로세스는 그 계수들의 일부 또는 모두와 연관되는 비트 심도를 감소시킬 수도 있다. 양자화의 정도는 양자화 파라미터를 조정함으로써 변경될 수도 있다.

[0080] 양자화 이후, 엔트로피 인코딩 유닛 (56) 은 양자화된 변환 계수들을 엔트로피 인코딩한다. 예를 들어, 엔트로피 인코딩 유닛 (56) 은 컨텍스트 적응 가변 길이 코딩 (CAVLC), 컨텍스트 적응 바이너리 산술 코딩 (CABAC), 구문-기반 컨텍스트-적응 바이너리 산술 코딩 (SBAC), 확률 간격 파티셔닝 엔트로피 (PIPE) 코딩 또는 또 다른 엔트로피 인코딩 방법론 또는 기법을 수행할 수도 있다. 이러한 엔트로피 인코딩은 일반적으로, 양자화된 변환 계수들 (일반적으로, 본원에서는 간결성을 위해 "변환 계수들" 로서 간단히 지칭됨) 을 한 번 이상 스캐닝하고, 대응하는 변환 계수들이 유의한지, 1 또는 2 보다 큰 절대값, 절대값 (또는, 그의 부분, 예컨대, 2 보다 큰 부분) 및 유의 계수들의 부호를 갖는지 여부를 나타내는 구문 엘리먼트들과 같은, 변환 계수들에 대한 구문 엘리먼트들을 각각의 스캐닝 동안, 엔트로피 코딩하는 것을 포함한다.

[0081] 본 개시물의 기법들에 따르면, 엔트로피 인코딩 유닛 (56) 은 변환 계수가 발생하는 비디오 블록의 영역에 기초하여, 비디오 블록 (예컨대, 변환 유닛) 의 변환 계수를 코딩 (즉, 엔트로피 인코딩) 하기 위한 컨텍스트를 결정할 수도 있다. 예를 들어, 스캐닝 동안, 엔트로피 인코딩 유닛 (56) 은 비디오 블록에서의 변환 계수의 포지션을 결정하고, 그 포지션이 발생하는 영역을 결정할 수도 있다. 게다가, 엔트로피 인코딩 유닛 (56) 은 비디오 블록에 대한 영역들을 정의하는 구성 데이터를 포함할 수도 있다.

[0082] 예를 들어, 엔트로피 인코딩 유닛 (56) 은 임계값을 사용하여 구성될 수도 있다. 이 예에서, 엔트로피 인코딩 유닛 (56) 은 변환 계수의 포지션을 정의하는 x- 및 y-좌표들이 임계값보다 큰 총합 (즉, x+y) 을 가지는지 여부를 결정할 수도 있다. 제 1 영역은, 이 예에서, x- 및 y-좌표 값들의 총합이 임계값 미만인 변환 계수들에 대응하며, 제 2 영역은 x- 및 y-좌표 값들의 총합이 임계값보다 크거나 동일한 변환 계수들에 대응한다. 이의 대안으로, 다수의 임계값들이 다수의 영역들을 정의하는데 사용될 수도 있다. 이러한 식으로 정의된 영역들의 일 예가 아래에서 더욱더 자세하게 설명되는 도 9b 에 도시된다.

[0083] 또 다른 예로서, 엔트로피 인코딩 유닛 (56) 은 비디오 블록에서의 변환 계수를 포함하여, 서브 블록의 포지션을 결정하도록 구성될 수도 있다. 서브 블록은 4×4 변환 계수 서브 블록에 대응할 수도 있다. 즉, 비디오 블록은 복수의 비-중첩하는 서브 블록들을 포함할 수도 있으며, 각각의 서브 블록들은 동일한 사이즈, 예컨대, 4×4 변환 계수들을 갖는다. 서브 블록에 대한 영역을 결정하기 위해, 엔트로피 인코딩 유닛 (56) 은 서브 블록의 x- 및 y-좌표 (예컨대, 서브 블록의 상부-좌측 변환 계수와 같은, 서브 블록의 특정한 변환 계수) 의 총합을 임계값과 비교할 수도 있다. x- 및 y-좌표들의 총합이 임계값 미만인지 여부는 서브 블록의 변환 계수들이 제 1 영역 또는 제 2 영역에 포함되는지 여부를 나타낼 수도 있다.

[0084] 예를 들어, C_{ij} 가 포지션 (i, j) 에서 상부-좌측 변환 계수를 갖는 서브 블록의 포지션을 나타내며, 여기서, x=i 및 y=j 라고 하자. 또, T 가 임계값을 정의한다고 하자. 엔트로피 인코딩 유닛 (56) 은 다음의 사 코드를 사용하여 서브 블록의 변환 계수들이 발생하는 영역을 결정할 수도 있다:

[0085] (i+j<T) ? region1 : region2.

- [0086] 이 예에서, $i+j$ 가 T 미만 (즉, 서브 블록의 x - 및 y -좌표들의 총합이 임계값 미만) 일 때, 엔트로피 인코딩 유닛 (56) 은 서브 블록의 모든 변환 계수들이 영역 1 에서 발생한다고 결정하고, 반면, $i+j$ 가 T 보다 크거나 또는 동일할 (즉, 서브 블록의 x - 및 y -좌표들의 총합이 임계값보다 크거나 또는 동일할) 때, 엔트로피 인코딩 유닛 (56) 은 서브 블록의 모든 변환 계수들이 영역 2 에서 발생한다고 결정한다. 영역들의 이들 및 다른 예들이 도 9 내지 도 14 와 관련하여 아래에서 더 자세히 설명된다.
- [0087] 엔트로피 인코딩 유닛 (56) 은 여러 방법들에서의 영역들에 기초하여 컨텍스트들을 결정하도록 구성될 수도 있다. 예를 들어, 엔트로피 인코딩 유닛 (56) 은 변환 계수를 코딩하기 위한 컨텍스트를, 변환 계수가 발생하는 영역에 기초하여, 비디오 블록에서의 변환 계수의 로케이션 또는 변환 계수가 발생하는 4×4 서브 블록의 포지션을 사용하여, 결정할 수도 있다.
- [0088] 이의 대안으로, 컨텍스트 모델은 이웃하는 4×4 서브 블록들에 따라서 정의될 수도 있다. 예를 들어, 엔트로피 인코딩 유닛 (56) 은 각각의 4×4 서브 블록에 가용 컨텍스트들의 각각의 세트를 할당하고, 예컨대, 서브 블록에서의 변환 계수의 포지션에 기초하여, 서브 블록에서 코딩되는 현재의 변환 계수에 대한 컨텍스트들 중 하나를 선택할 수도 있다. 컨텍스트들의 세트들은, 각각의 서브 블록이 가용 컨텍스트들의 상이한 세트를 갖도록, 각각의 서브 블록들에 할당될 수도 있다. 또한, 또 다른 예로서, 엔트로피 인코딩 유닛 (56) 은 컨텍스트를 $ctx = \text{Right4x4SubBlockFlag} + \text{Bottom4x4SubBlockFlag}$ 로서 계산할 수도 있다. 이 경우, $\text{Right4x4SubBlockFlag}$ 는 우측-이웃하는 서브 블록에 대한 코딩된 서브 블록 플래그를 나타내지만, $\text{Bottom4x4SubBlockFlag}$ 는 최저부-이웃하는 코딩된 서브 블록 플래그에 대한 코딩된 서브 블록 플래그를 나타낸다.
- [0089] 일부 예들에서, 엔트로피 인코딩 유닛 (56) 은 오프셋을 변환 계수를 엔트로피 인코딩하기 위한 결정된 컨텍스트에 적용할 수도 있으며, 변환 계수가 발생하는 영역에 기초하여, 적용할 오프셋을 추가로 결정할 수도 있다. 즉, 엔트로피 인코딩 유닛 (56) 은 베이스 컨텍스트를 동일한 일반적인 방법으로 2 이상의 영역들의 계수들에 대해 계산할 수도 있으며, 그러나 상이한 영역들은 상이한 대응하는 오프셋 값들을 가질 수도 있다. 따라서, 엔트로피 인코딩 유닛 (56) 은 그 영역이 맵핑되는 오프셋 (즉, 그 영역이 연관되는 오프셋) 에 기초하여, 오프셋을 계산된 컨텍스트 값에 적용할 수도 있다.
- [0090] 엔트로피 인코딩 유닛 (56) 은 변환 계수가 (일반적으로 변환 블록의 상부-좌측 모서리에 나타내어지는) DC (direct current) 변환 계수를 결정하고, 변환 계수가 DC 변환 계수인지 여부 뿐만 아니라, 변환 계수가 발생하는 영역에 기초하여, 변환 계수를 코딩하기 위한 컨텍스트를 선택할 수도 있다. 예를 들어, 엔트로피 인코딩 유닛 (56) 은 전용 포지션들 (dedicated positions) 에 대한 공유된 컨텍스트들 (shared contexts) 을 사용하여, 변환 계수들에 대한 컨텍스트들을 결정할 수도 있다. 즉, 공유된 컨텍스트는 특정의 포지션, 예컨대, 서브 블록의 상부-좌측 모서리에서 발생하는 모든 변환 계수들에 적용되는 동일한 컨텍스트를 포함할 수도 있다. 따라서, 공유된 컨텍스트는 다른 서브 블록들의 상부-좌측 포지션에서 발생하는 비-DC 변환 계수들과는 대조적으로, DC 변환 계수를 코딩할 때 적용되는 특정의 컨텍스트의 표시를 추가로 포함할 수도 있다.
- [0091] 이에 추가적으로 또는 대안적으로, 공유된 컨텍스트는 블록들의 특정의 포지션들에서 발생하는 변환 계수들에 대한 상이한 사이즈들의 블록들 사이에서 공유된 컨텍스트들을 포함할 수도 있다. 예를 들어, 엔트로피 인코딩 유닛 (56) 은 임의의 사이즈, 예컨대, 4×4 , 8×8 , 16×16 등의 비디오 블록들 (예컨대, TU들) 의 DC 변환 계수들을 코딩할 때 동일한 컨텍스트를 적용하도록 구성될 수도 있다. 즉, 엔트로피 인코딩 유닛 (56) 은 임의의 사이즈의 블록들에 대해, DC 변환 계수를 DC 변환 계수를 코딩하기 위한 동일한 컨텍스트 데이터에 맵핑하는 데이터를 포함할 수도 있다. 즉, 엔트로피 인코딩 유닛 (56) 은 코딩중인 현재의 비디오 블록의 사이즈에 관계 없이, DC 변환 계수에 대해 결정된 컨텍스트를 사용하여, DC 변환 계수를 코딩하도록 구성될 수도 있다. 일반적으로, DC 변환 계수는 비디오 블록의 상부-좌측 계수이다.
- [0092] 엔트로피 인코딩 유닛 (56) 에 의한 엔트로피 인코딩에 이어서, 인코딩된 비트스트림은 비디오 디코더 (30) 로 송신되거나, 또는 비디오 디코더 (30) 에 의한 추후 송신 또는 취출을 위해 아카이브될 수도 있다. 엔트로피 인코딩 유닛 (56) 은 또한 코딩중인 현재의 비디오 슬라이스에 대한, 모션 벡터들, 인트라-모드 표시들, 및 다른 구문 엘리먼트들을 엔트로피 인코딩할 수도 있다.
- [0093] 역양자화 유닛 (58) 및 역변환 유닛 (60) 은 역양자화 및 역변환을 각각 적용하여, 참조 화상의 참조 블록으로 추후 사용을 위해 픽셀 도메인에서 잔여 블록을 재구성한다. 모션 보상 유닛 (44) 은 잔여 블록을 참조 화상 리스트들 중 하나 내의 참조 화상들 중 하나의 예측 블록에 가산함으로써 참조 블록을 계산할 수도

있다. 모션 보상 유닛 (44) 은 또한 하나 이상의 내삽 필터들을 그 재구성된 잔여 블록에 적용하여, 모션 추정에 사용하기 위한 서브-정수 픽셀 값들을 계산할 수도 있다. 합산기 (62) 는 재구성된 잔여 블록을 모션 보상 유닛 (44) 에 의해 생성된 모션 보상된 예측 블록에 가산하여, 참조 화상 메모리 (64) 에의 저장을 위한 참조 블록을 생성한다. 참조 블록은 후속 비디오 프레임 또는 화상에서 블록을 인터-예측하기 위해 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 에 의해 참조 블록으로서 사용될 수도 있다.

[0094] 이 방식으로, 비디오 인코더 (20) 는 변환 계수가 발생하는 비디오 블록의 영역에 기초하여, 비디오 블록의 변환 계수를 코딩하기 위한 컨텍스트를 결정하며, 결정된 컨텍스트를 사용하여 변환 계수를 엔트로피 코딩하도록 구성된 비디오 코더의 일 예를 나타낸다. 영역은 비디오 블록의 변환 계수들의 하나 이상의 상부-좌측 4×4 서브 블록들을 포함하는 제 1 영역, 및 제 1 영역 외부의 비디오 블록의 변환 계수들을 포함하는 제 2 영역 중 하나를 포함할 수도 있다.

[0095] 도 3 은 본 개시물에서 설명되는 인터-예측 기법들을 구현할 수도 있는 예시적인 비디오 디코더 (30) 를 예시하는 블록도이다. 도 3 의 예에서, 비디오 디코더 (30) 는 엔트로피 디코딩 유닛 (80), 예측 프로세서 (81), 역양자화 유닛 (86), 역변환 유닛 (88), 합산기 (90), 및 참조 화상 메모리 (92) 를 포함한다. 예측 프로세서 (81) 모션 보상 유닛 (82) 및 인트라 예측 유닛 (84) 을 포함한다. 비디오 디코더 (30) 는 일부 예들에서, 도 2 로부터의 비디오 인코더 (20) 에 대해 설명된 인코딩 과정과는 일반적으로 반대인 디코딩 과정을 수행할 수도 있다.

[0096] 디코딩 프로세스 동안, 비디오 디코더 (30) 는 인코딩된 비디오 슬라이스의 비디오 블록들 및 연관되는 구문 엘리먼트들을 나타내는 인코딩된 비디오 비트스트림을 비디오 인코더 (20) 로부터 수신한다. 비디오 디코더 (30) 의 엔트로피 디코딩 유닛 (80) 은 비트스트림을 엔트로피 디코딩하여, 양자화된 계수들, 모션 벡터들, 및 다른 구문 엘리먼트들을 생성한다. 엔트로피 디코딩 유닛 (80) 은 모션 벡터들, 인트라-모드 표시들, 및 다른 예측-관련되는 구문 엘리먼트들을 예측 프로세서 (81) 로 포워딩한다. 엔트로피 디코딩 유닛 (80) 은 양자화된 계수들을, 블록 (예컨대, TU) 의 유형으로, 역양자화 유닛 (86) 으로 포워딩한다. 비디오 디코더 (30) 는 구문 엘리먼트들을 비디오 슬라이스 레벨 및/또는 비디오 블록 레벨에서 수신할 수도 있다.

[0097] 특히, 본 개시물의 기법들에 따르면, 엔트로피 디코딩 유닛 (80) 은 변환 계수들이 발생하는 블록의 영역에 기초하여, 변환 계수들을 엔트로피 디코딩하기 위한 컨텍스트를 결정할 수도 있다. 구체적으로 설명하면, 엔트로피 디코딩 유닛 (80) 은, 일단 변환 계수가 그 블록 내에 포지셔닝되면, 변환 계수가 발생할 블록의 영역에 기초하여 컨텍스트를 결정할 수도 있다. 엔트로피 디코딩 유닛 (80) 은 도 9 내지 도 14 와 관련하여 아래에서 설명되는 바와 같은 영역들, 또는 다른 이러한 영역들을 결정하도록 구성될 수도 있다. 예를 들어, 도 9a 에 나타난 바와 같이, 엔트로피 디코딩 유닛 (80) 은 변환 계수가 블록의 상부-좌측 모서리에서의 하나 이상의 서브 블록들을 포함하는 제 1 영역, 또는 제 1 영역 외부의 서브 블록들을 포함하는 제 2 영역에 발생할지 여부를 결정하며, 변환 계수가 제 1 영역 또는 제 2 영역에서 발생하는지 여부에 기초하여 컨텍스트를 결정하도록 구성될 수도 있다.

[0098] 이와 유사하게, 엔트로피 디코딩 유닛 (80) 은, 엔트로피 디코딩 유닛 (80) 이 각각의 영역에서 계수들과 연관되는 컨텍스트를 계산하거나 또는 결정하는 하나 이상의 여러 기법들로 구성될 수도 있다는 점에서, 그 영역에 기초하여 컨텍스트를 결정할 수도 있다. 즉, 각각의 영역은 컨텍스트를 계산하거나 또는 결정하는 하나 이상의 기법들과 연관될 수도 있다. 예를 들어, 영역은 하나 이상의 변환 계수들 사이에 공유되는 컨텍스트와 연관될 수도 있다. 또 다른 예로서, 영역은 그 영역의 서브 블록들 사이에 공유되는 컨텍스트들과 연관될 수도 있다. 또한 또 다른 예로서, 영역은 그 영역에서의 변환 계수에 대해 계산된 컨텍스트 값에 적용된 오프셋 값과 연관될 수도 있다. 엔트로피 디코딩 유닛 (80) 은 이들 또는 본원에서 설명되는 다른 기법들을 사용하여, 변환 계수가 발생하는 영역에 기초하여, 변환 계수를 디코딩하기 위한 컨텍스트를 결정하도록 구성될 수도 있다. 엔트로피 디코딩 유닛 (80) 은 그 후 결정된 컨텍스트를 사용하여 변환 계수를 엔트로피 디코딩할 수도 있다.

[0099] 이에 추가적으로 또는 대안적으로, 공유된 컨텍스트는 블록들의 특성의 포지션들에서 발생하는 변환 계수들에 대한 상이한 사이즈들의 블록들 사이에서의 공유된 컨텍스트들을 포함할 수도 있다. 예를 들어, 엔트로피 디코딩 유닛 (80) 은 임의의 사이즈, 예컨대, 4×4, 8×8, 16×16, 또는 기타 등등의 비디오 블록들 (예컨대, TU들) 의 DC 변환 계수들을 코딩할 때 동일한 컨텍스트를 적용하도록 구성될 수도 있다. 즉, 엔트로피 디코딩 유닛 (80) 은 임의의 사이즈의 블록들에 대한, DC 변환 계수들, DC 변환 계수를 코딩하기 위한 동일한 컨텍스트 데이터에 맵핑하는 데이터를 포함할 수도 있다. 즉, 엔트로피 디코딩 유닛 (80) 은 코딩중인 현

재의 비디오 블록의 사이즈에 관계 없이, DC 변환 계수에 대해 결정된 컨텍스트를 사용하여, DC 변환 계수를 코딩하도록 구성될 수도 있다. 일반적으로, DC 변환 계수는 비디오 블록의 상부-좌측 계수이다.

[0100] 비디오 슬라이스가 인트라-코딩된 (I) 슬라이스로서 코딩될 때, 예측 프로세서 (81) 의 인트라 예측 유닛 (84) 은 시그널링된 인트라 예측 모드 및 현재의 프레임 또는 화상의 이전에 디코딩된 블록들로부터의 데이터에 기초하여, 현재의 비디오 슬라이스의 비디오 블록에 대한 예측 데이터를 생성할 수도 있다. 비디오 프레임이 인터-코딩된 (즉, B, P 또는 GPB) 슬라이스로서 코딩될 때, 예측 프로세서 (81) 의 모션 보상 유닛 (82) 은 엔트로피 디코딩 유닛 (80) 으로부터 수신된 모션 벡터들 및 다른 구문 엘리먼트들에 기초하여, 현재의 비디오 슬라이스의 비디오 블록에 대한 예측 블록들을 생성한다. 예측 블록들은 참조 화상 리스트들 중 하나 내의 참조 화상들 중 하나로부터 생성될 수도 있다. 비디오 디코더 (30) 는 디폴트 구성 기법들을 사용하여, 참조 화상 메모리 (92) 에 저장된 참조 화상들에 기초하여, 참조 프레임 리스트들, 즉, 리스트 0 및 리스트 1 를 구성할 수도 있다.

[0101] 모션 보상 유닛 (82) 은 모션 벡터들 및 다른 구문 엘리먼트들을 파싱하여 현재의 비디오 슬라이스의 비디오 블록에 대한 예측 정보를 결정하며, 그 예측 정보를 사용하여, 디코딩중인 현재의 비디오 블록에 대한 예측 블록들을 생성한다. 예를 들어, 모션 보상 유닛 (82) 은 그 수신된 구문 엘리먼트들 중 일부를 사용하여, 비디오 슬라이스의 비디오 블록들을 코딩하는 사용되는 예측 모드 (예컨대, 인트라- 또는 인터-예측), 인터-예측 슬라이스 유형 (예컨대, B 슬라이스, P 슬라이스, 또는 GPB 슬라이스), 슬라이스에 대한 참조 화상 리스트들 중 하나 이상에 대한 구성 정보, 슬라이스의 각각의 인터-인코딩된 비디오 블록에 대한 모션 벡터들, 슬라이스의 각각의 인터-코딩된 비디오 블록에 대한 인터-예측 상태, 및 다른 정보를 결정하여, 현재의 비디오 슬라이스에서의 비디오 블록들을 디코딩한다.

[0102] 모션 보상 유닛 (82) 은 또한 내삽 필터들에 기초하여 내삽을 수행할 수도 있다. 모션 보상 유닛 (82) 은 비디오 블록들의 인코딩 동안 비디오 인코더 (20) 에 의해 사용되는 것과 같은 내삽 필터들을 사용하여, 참조 블록들의 서브-정수 픽셀들에 대해 내삽된 값들을 계산할 수도 있다. 이 경우, 모션 보상 유닛 (82) 은 수신된 구문 엘리먼트들로부터 비디오 인코더 (20) 에 의해 사용되는 내삽 필터들을 결정하고 그 내삽 필터들을 사용하여 예측 블록들을 생성할 수도 있다.

[0103] 역양자화 모듈 (86) 은 비트스트림으로 제공되며 엔트로피 디코딩 유닛 (80) 에 의해 디코딩된 양자화된 변환 계수들을 역양자화한다, 즉, 양자화 해제한다. 역양자화 프로세스는 양자화의 정도와, 마찬가지로, 적용되어야 하는 역양자화의 정도를 결정하기 위해, 비디오 슬라이스에서의 각각의 비디오 블록에 대해 비디오 인코더 (20) 에 의해 계산된 양자화 파라미터의 사용을 포함할 수도 있다. 역변환 유닛 (88) 은 역 DCT, 역 정수 변환, 또는 개념적으로 유사한 역변환 프로세스와 같은 역변환을 변환 계수들에 적용하여, 픽셀 도메인에서 잔여 블록들을 생성한다.

[0104] 일부의 경우, 역변환 유닛 (88) 은 2차원 (2-D) 역변환을 (수평 및 수직 방향 양자에서) 계수들에 적용할 수도 있다. 본 개시물의 기법들에 따르면, 역변환 유닛 (88) 은 수평 1차원 (1-D) 역변환, 수직 1-D 역변환, 또는 무 변환 (no transform) 을 TU들의 각각에서의 잔여 데이터에 대신 적용할 수도 있다. 비디오 인코더 (20) 에서 잔여 데이터에 적용된 변환의 유형은 적합한 역변환의 유형을 변환 계수들에 적용하기 위해 비디오 디코더 (30) 로 시그널링될 수도 있다.

[0105] 모션 보상 유닛 (82) 이 모션 벡터들 및 다른 구문 엘리먼트들에 기초하여 현재의 비디오 블록에 대한 예측 블록을 생성한 후, 비디오 디코더 (30) 는 역변환 유닛 (88) 으로부터의 잔여 블록들을 모션 보상 유닛 (82) 에 의해 생성된 대응하는 예측 블록들과 합산함으로써, 디코딩된 비디오 블록을 형성한다. 합산기 (90) 는 이 합산 동작을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다. 원할 경우, 블록킹 현상 아티팩트들 (blockiness artifacts) 을 제거하기 위해 디블록킹 필터가 또한 그 디코딩된 블록들을 필터링하는데 적용될 수도 있다. (코딩 루프 중에 또는 코딩 루프 이후에) 다른 루프 필터들이 또한 픽셀 전환들 (pixel transitions) 을 평활화하거나 또는 아니면 비디오 품질을 향상시키기 위해 사용될 수도 있다. 주어진 프레임 또는 화상에서 디코딩된 비디오 블록들은 그 후 참조 화상 메모리 (92) 에 저장되며, 이 메모리는 후속 모션 보상을 위해 사용되는 참조 화상들을 저장한다. 참조 화상 메모리 (92) 는 또한 도 1 의 디스플레이 디바이스 (32) 와 같은 디스플레이 디바이스 상에의 추후 프리젠테이션을 위해, 디코딩된 비디오를 저장한다.

[0106] 이 방식으로, 비디오 디코더 (30) 는 변환 계수가 발생하는 비디오 블록의 영역에 기초하여, 비디오 블록의 변환 계수를 코딩하기 위한 컨텍스트를 결정하며, 결정된 컨텍스트를 사용하여 변환 계수를 엔트로피 코딩하도록 구성되는 비디오 코더의 일 예를 나타낸다. 영역은 비디오 블록의 변환 계수들의 하나 이상의 상부-

좌측 4×4 서브 블록들을 포함하는 제 1 영역, 및 제 1 영역 외부의 비디오 블록의 변환 계수들을 포함하는 제 2 영역 중 하나를 포함할 수도 있다.

[0107] 도 4 는 비디오 블록에서의 변환 계수들과, 비디오 블록과 연관된 유의도 맵 사이의 관계를 예시하는 개념도이다. 도 4 에 예시된 바와 같이, 유의도 맵은 비디오 블록에서, 유의 계수 값, 즉, 제로보다 큰 값의 각각의 인스턴스를 나타내기 위해 "1" 을 포함한다. 유의도 맵은 디코딩되는 비디오 블록에서의 유의한, 즉, 제로보다 큰, 계수들의 로케이션을 결정하기 위해, 비디오 디코더 (30) 와 같은 비디오 디코더에 의해 디코딩가능한 비트스트림으로, 시그널링될 수도 있다. 좀더 구체적으로는, 비디오 블록 내의 최종 난-제로 계수의 포지션이 비트스트림으로 시그널링될 수도 있다. 비디오 블록에서의 최종 난-제로 계수의 포지션은 비디오 블록에 사용되는 스캐닝 순서에 의존한다. 추가적인 구문 엘리먼트들이 알려진 또는 알려질 수 있는 스캐닝 순서에 따라서 최종 난-제로 계수에 대한, 다른 유의 계수들을 나타내기 위해 시그널링될 수도 있다.

[0108] 도 5a 내지 도 5d 는 지그-재그 스캐닝 순서, 수평 스캐닝 순서, 수직 스캐닝 순서, 및 대각선 스캐닝 순서를 사용하여 스캐닝되는 비디오 데이터의 블록들의 예들을 예시하는 개념도들이다. 도 5a 내지 도 5d 에 나타낸 바와 같이, 비디오 데이터, 예컨대, CU 의 TU 의 8×8 블록은 원들로 표시된, 대응하는 블록 포지션들에서의 64 개의 변환 계수들을 포함할 수도 있다. 이 예에서, 블록들 (100, 102, 104 및 106) 는 각각 8×8 의 사이즈를 가지며, 따라서, 앞에서 설명한 예측 기법들을 사용하여 발생된 64 개의 변환 계수들을 포함한다.

[0109] 본 개시물에서 설명하는 기법들에 따르면, 블록들 (100, 102, 104 및 106) 의 각각에서의 64 개의 변환 계수들은 2-D 변환, 수평 1-D 변환, 및 수직 1-D 변환 중 하나를 사용하여 변환되었거나 또는 역변환될 수도 있거나, 또는 변환 계수들은 전혀 변환되지 않을 수도 있다. 변환되든 안되든, 비디오 블록들 (100, 102, 104 및 106) 의 각각에서의 계수들은 지그-재그 스캐닝 순서, 수평 스캐닝 순서, 수직 스캐닝 순서, 및 대각선 스캐닝 순서 중 하나를 사용하여 엔트로피 코딩하는 준비로 스캐닝된다.

[0110] 도 5a 에 나타낸 바와 같이, 블록 (100) 과 연관되는 스캐닝 순서는 지그-재그 스캐닝 순서이다. 지그-재그 스캐닝 순서는 비디오 인코더 (20) 또는 비디오 디코더 (30) 와 같은, 비디오 코더로 하여금, 블록 (100) 의 양자화된 변환 계수들을 도 5a 에 화살표들로 나타낸 바와 같이 대각선으로 스캐닝가능하게 한다. 도 5d 에서와 유사하게, 대각선 스캐닝 순서는 비디오 코더로 하여금, 블록 (106) 의 양자화된 변환 계수들을 도 5d 에 화살표들로 나타낸 바와 같이 대각선 방식으로 스캐닝가능하게 한다. 도 5b 및 도 5c 에 나타낸 바와 같이, 블록들 (102 및 104) 과 연관되는 스캐닝 순서들은 각각 수평 스캐닝 순서 및 수직 스캐닝 순서이다. 또한 도 5b 및 도 5c 에 화살표들로 나타낸 바와 같이, 수평 스캐닝 순서는 비디오 코더로 하여금 스캐닝 블록 (102) 의 양자화된 변환 계수들을 수평 라인 단위로, 또는 "래스터" 방식으로 스캐닝가능하며, 한편 수직 스캐닝 순서는 비디오 코더로 하여금 블록 (104) 의 양자화된 변환 계수들을 수직 라인 단위로, 또는 "순환 라스터" 방식으로 스캐닝가능하게 한다.

[0111] 다른 예들에서, 위에서 설명한 바와 같이, 블록은 블록들 (100, 102, 104 및 106) 의 사이즈보다 더 작은 또는 더 큰 사이즈를 가질 수도 있으며, 더 많거나 또는 더 적은 양자화된 변환 계수들 및 대응하는 블록 포지션들을 포함할 수도 있다. 이들 예들에서, 특정의 블록과 연관되는 스캐닝 순서는 비디오 코더로 하여금 블록의 양자화된 변환 계수들을 도 5a 내지 도 5d 의 8×8 블록들의 예들에 나타낸 바와 같은 실질적으로 유사한 방법으로 스캐닝가능하게 할 수도 있으며, 예컨대, 4×4 블록 또는 16×16 블록이 앞에서 설명한 스캐닝 순서들 중 임의의 스캐닝 순서에 뒤따라서 스캐닝될 수도 있다.

[0112] 도 5a 내지 도 5d 에서는 스캐닝들의 방향이 일반적으로 낮은-주파수 계수들로부터 높은-주파수 계수들로 진행되는 것으로 나타내지만, 다른 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 스캐닝이 높은-주파수 계수들로부터 낮은-주파수 계수들로 진행할 수도 있는 역 스캐닝 순서를 수행하도록 구성될 수도 있다. 즉, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 계수들을 도 5a 내지 도 5d 에 나타낸 것의 역방향 순서로 스캐닝할 수도 있다.

[0113] 도 6 은 변환 계수 코딩을 위한 서브 블록들로 분할되는 예시적인 비디오 블록 (110) 을 예시하는 개념도이다. 현재의 HM 에서, 서브 블록 컨셉이 변환 계수 코딩을 위해 사용된다. 비디오 코더는 결정된 서브 블록 사이즈보다 큰 임의의 변환 유닛 (TU) 을 서브 블록들로 하위 분할될 수도 있다. 예를 들어, 비디오 블록 (110) 은 4개의 4×4 서브 블록들로 분할된다.

[0114] 도 6 의 예시된 예에서, 비디오 코더는 비디오 블록 (110) 을 4×4 서브 블록들로 분할한다. 다른 예들에

서, 비디오 코더는 비디오 블록들을 다른 사이즈들, 예컨대, 8×8, 16×16 등의 서브 블록들로 분할할 수도 있다. 비디오 코더가 프레임 또는 슬라이스의 모든 TU들에 대해 동일한 서브 블록 사이즈를 사용하면, 서브 블록 사이즈들에 의해 획득되는 균일성으로 인해, 하드웨어 구현에서 이득들이 획득될 수도 있다. 예를 들어, 모든 프로세싱이 TU 사이즈에 관계 없이, 이러한 서브 블록들에서 분할될 수도 있다. 그러나, 균일한 서브 블록 사이즈가 본 개시물의 기법들을 수행하는데 필수적은 아니다.

[0115] 계수 코딩을 위해, 비디오 코더는 도 6 상에 나타낸 바와 같이, 대각선 스캐닝 순서를 사용하여, 비디오 블록 (110) 의 각각의 4×4 서브 블록을 스캐닝할 수도 있다. 일부 예들에서, 비디오 코더는 각각의 서브 블록의 변환 계수들을 스캐닝하기 위해 통합된 (unified) 스캐닝을 사용할 수도 있다. 이 경우, 동일한 스캐닝 순서는 유의도 정보, 즉, 유의도 맵, 계수 레벨들, 부호 등에 대해 사용된다. 제 1 예에서, 도 6 에 나타낸 바와 같이, 비디오 코더는 대각선 스캐닝을 사용하여 변환 계수들을 스캐닝할 수도 있다. 또 다른 예에서, 비디오 코더는 도 6 에 나타낸 것의 반대인 순서로, 예컨대, 하부 우측 모서리에서 시작하여 상부 좌측 모서리로 진행되는 역방향 대각선 스캐닝으로, 변환 계수들을 스캐닝할 수도 있다. 다른 예들에서, 비디오 코더는 지그-재그, 수평, 또는 수직 스캐닝을 사용하여 변환 계수들을 스캐닝할 수도 있다. 다른 스캐닝 방향들/방위들이 또한 가능하다.

[0116] 설명의 용이성을 위해, 본 개시물은 비디오 블록의 서브 블록들을 4×4 서브 블록들인 것으로 기술한다. 본 개시물의 기법들은, 그러나, 또한 상이한 사이즈들, 예컨대, 8×8, 16×16 등의 서브 블록들에 대해 적용될 수도 있다. 모든 4×4 블록에 대해, significant_coeffgroup_flag 가 코딩되며, 서브 블록에서 적어도 하나의 널-제로 계수가 존재하면, 이 플래그는 1로 설정되고, 그렇지 않으면, 제로와 동일하다. significant_coeffgroup_flag 가 주어진 서브 블록에 대해 널-제로이면, 4×4 서브 블록은 역방향 대각선 순서로 스캐닝되며, significant_coeff_flag 는 계수의 유의도를 나타내기 위해 서브 블록의 모든 계수에 대해 코딩된다. 이들 플래그들의 그룹은 비디오 블록에 대한 유의도 맵으로서 지칭될 수도 있다. 일부 예에서, 유의도 맵을 명시적으로 시그널링하는 대신에, significant_coeffgroup_flag 는 이웃하는 4×4 서브 블록 플래그들을 사용하거나, 또는 4×4 서브 블록이 최종 계수 또는 DC 계수를 포함할 때, 내재적으로 유도될 수도 있다. 계수들의 절대값들, 즉, 계수 레벨들이 코딩된다.

[0117] 도 6 에서 스캐닝의 방향은 일반적으로 낮은-주파수 계수들로부터 높은-주파수 계수들로 진행되는 것으로 도시되지만, 다른 예들에서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 스캐닝이 높은-주파수 계수들로부터 낮은-주파수 계수들로 진행할 수도 있는 역 스캐닝 순서를 수행하도록 구성될 수도 있다. 즉, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 계수들을 도 6 에 나타낸 것의 역방향 순서로 스캐닝할 수도 있다.

[0118] 도 7 은 역방향 대각선 스캐닝 순서를 사용하여 스캐닝되는 비디오 블록 (112) 에서 계수들의 유의도 맵에 대한 콘텍스트들의 선택을 위한 콘텍스트 모델을 정의하는데 사용되는 예시적인 5-지점 서포트 이웃을 예시하는 개념도이다. 위에서 언급한 바와 같이, 콘텍스트-적응적 코딩을 위해, 변환 계수들은 변환 계수가 0 의 값 또는 1 의 값을 가질 확률들을 기술하는 콘텍스트 모델에 기초하여 코딩될 수도 있다. 유의도 맵 코딩과 관련하여, 콘텍스트 모델은 특정의 변환 계수가 유의한지, 즉, 널-제로인지 여부의 확률들을 기술한다.

[0119] 유의도 (significance) 맵 코딩을 위해, 5-지점 서포트 S 가 비디오 블록 (112) 의 변환 계수들의 유의도 맵을 코딩하기 위해 콘텍스트 모델을 정의하는데 사용될 수도 있다. 5-지점 서포트는 "콘텍스트 서포트 이웃", 또는 간단히 "서포트 이웃" 으로서 지칭될 수도 있다. 즉, 비디오 코더는 서포트를 고려하여, 현재의 포지션의 유의도가 1 또는 제로일 확률을 결정할 수도 있다. 콘텍스트 서포트 이웃은 현재의 계수들을 코딩하기 위한 콘텍스트들로서 사용될 수도 있는 (예컨대, 유의도 정보를 포함할 수도 있는) 이웃하는 계수들을 정의한다. 본 개시물의 일부 예들에 따르면, 콘텍스트 서포트 이웃은 블록 또는 서브 블록 내의 상이한 계수 포지션들에 대해 상이할 수도 있다.

[0120] 도 7 에 나타낸 예에서, 5-지점 서포트 S 는 둘러싸인 점으로 표현되는 현재 또는 "타겟" 포지션에 대해, 정사각형으로 둘러싸인 점으로 표현된다. 콘텍스트 모델 Ctx (아래 수식 (1)) 는 서포트의 모든 지점에서의 유의한 플래그들의 총합으로서 정의될 수도 있으며, 여기서, 유의도 플래그는 대응하는 변환 계수가 널-제로이면, "1" 로 설정되고 그렇지 않으면 "0" 으로 설정될 수도 있다.

$$Ctx = \sum_{p \in S} (coef_p \neq 0) \tag{1}$$

[0122] 따라서, 유의도 플래그 카운트는 서포트 원소 개수 (cardinality) 보다 작거나 또는 동일할 수 있다. ctx

의 값은 반드시 미가공 컨텍스트 값일 필요가 없으며, 그러나, 특정의 계수에 대한 데이터를 코딩하는데 사용되는 컨텍스트를 유도하기 위해, 베이스 컨텍스트 값에 오프셋의 유형으로 적용될 수도 있다.

[0123] 그러나, 도 7 에 나타낸 서포트 S 는 하나 보다 많은 변환 계수에 대한 컨텍스트 (예컨대, 변환 계수와 연관된 유의도 정보) 를 병렬로 계산할 때에 ("병렬 유의도 컨텍스트 계산" 또는 간단히 "병렬 컨텍스트 계산" 으로서 지칭됨) 적합하지 않을 수도 있다. 예를 들어, 도 7 에 나타낸 서포트 S 를 사용하는 것은, 서포트 S 에서의 모든 데이터가 컨텍스트들의 병렬 계산을 가능하게 하는데 이용 가능해야 하기 때문에, 유의도 정보에 대한 컨텍스트들을 병렬로 계산하는 비디오 코더의 능력을 방해할 수도 있다. 일부의 경우, 도 8a 과 관련하여 아래에서 설명되는 바와 같이, 코더는 서포트 S 에서의 서포트 엘리먼트가 서포트 S 에서의 또 다른 서포트 엘리먼트에 대한 컨텍스트를 결정하기 전에 코딩을 종료하기를 대기하도록 강요 받을 수도 있다. 이 지연은 유의도 정보를 효율적으로 프로세싱하는 비디오 코더의 능력을 감소시킨다.

[0124] 도 8a 및 도 8b 는 5-지점 서포트 내의 컨텍스트 의존성을 예시하는 개념도들이다. 예를 들어, 둘러싼 포지션 (circled position) 에 대한 유의도 컨텍스트를 계산하기 위해, (도 8a 에 도시된) 마름모꼴로 도시된 서포트 S 내의 포지션의 유의도 플래그를 파싱하는 것이 필요할 수도 있다. 이러한 파싱은, 마름모꼴이 스캐닝 순서에서 둘러싼 엘리먼트 바로 앞에 포지션되기 때문에, 2개의 계수들의 유의도 컨텍스트들을 병렬로 계산하라는 요구 사항이 존재하면, 지연을 도입할 수도 있다. 즉, 둘러싼 포지션이 마름모꼴로 표시된 포지션에 의존하고, 따라서 마름모꼴로 표시된 포지션이 둘러싼 포지션에 대한 컨텍스트를 결정하기 전에 코딩되어야 하기 때문에, 둘러싼 포지션의 컨텍스트는 마름모꼴로 표시된 포지션과 동시에 계산될 수 없다.

[0125] 이 의존성을 결정하기 위해, 어떤 엘리먼트들이 서포트 S 로부터 제거되어, 그 서포트를 소위 "홀" (도 8b에 도시된, 삼각형으로 둘러싸인 채워지지 않은 점) 로 만들 수도 있다. 예를 들어, 홀에서의 유의도 플래그는 스킵되며, 컨텍스트 계산에 고려되지 않는다 (즉, 제로인 것으로 추정됨). 따라서, 홀 포지션에서의 유의도 플래그를 파싱할 필요가 없다. 5-지점 서포트 형태는 더 나은 병렬 프로세싱을 가능하게 하기 위해 그 포지션에 의존한다.

[0126] 도 9a 및 도 9b 는 2 이상의 영역들로의 비디오 블록의 예시적인 분할들을 예시하는 개념도들이다. 현재의 HM 에서, 이웃 컨텍스트 모델링은 5-지점 서포트를 가진 8×8 보다 큰 TU 사이즈들 (즉, 16×16, 32×32 및 비-정사각형 변환 사이즈들 16×4, 4×16, 32×8 및 8×32) 에 대해 사용된다. 그러나, 5-지점 서포트에 의한 컨텍스트 모델링은 더 큰 블록 사이즈들에서 컨텍스트 계산들의 복잡성을 증가시킬 수도 있다. 도 9a 의 영역 (R1) 은 비디오 블록의 변환 계수들의 하나 이상의 상부-좌측 4×4 서브 블록들을 포함하는 영역의 일 예를 나타내지만, 도 9a 의 영역 (R2) 는 영역 (R1) 외부의 비디오 블록의 변환 계수들을 포함하는 영역의 일 예를 나타낸다. 도 9a 는 또한 복수의 영역들이 하나 이상의 서브 블록들의 각각의 세트들을 포함하는 예를 나타낸다.

[0127] 본 개시물에서 설명하는 기법들에 따르면, 비디오 인코더 (20) 또는 비디오 디코더 (30) 와 같은, 비디오 코더는, 비디오 블록을 (예컨대, 도 9a 및 도 9b 에 나타낸 바와 같이) 영역들 (R) 로 분할하고 상이한 영역들 각각에 대해 상이한 컨텍스트 할당 프로시저들을 사용할 수도 있다. 예를 들어, 일부 영역들은 고정된 또는 포지션-기반 컨텍스트를 사용할 수도 있으며, 일부 영역들은 이웃-기반 컨텍스트를 사용할 수도 있다. 도 9a 에 예시된 바와 같이, 영역들은 전체 서브 블록들이 하나의 영역 또는 또 다른 영역에 포함되도록, 4×4 서브 블록들에 기초할 수도 있다. 또한, 영역들로의 분할은 일부 예들에서 플렉서블할 수 있다. 도 9b 에 예시된 바와 같이, 비디오 블록은 서브 블록들의 부분들이 2개의 상이한 영역들에 포함될 수 있도록, 영역들을 대각선 방향으로 분할될 수도 있다. 다른 예들에서, 분할은 계수 포지션들 또는 이 계수를 포함하는 4×4 서브 블록의 포지션에 의존해도 된다.

[0128] 일부 예들에서, 컨텍스트는 비디오 블록에서의 계수 포지션에 따라서, 또는 이 계수를 포함하는 4×4 서브 블록의 포지션에 따라서 정의될 수도 있다. 이의 대안으로, 컨텍스트 모델은 이웃 4×4 서브 블록들에 따라서 정의될 수도 있다. 예를 들어, 동일한 4×4 서브 블록 내의 모든 계수는 하나 또는 여러 컨텍스트들을 사용할 수 있으며, 다음 4×4 서브 블록의 계수들은 또한 하나 또는 여러 컨텍스트들을 사용할 수 있다. 그러나, 하나의 4×4 서브 블록의 컨텍스트들은 이전 4×4 서브 블록 기반 컨텍스트들과는 상이해도 된다. 이의 대안으로, 컨텍스트들은 그 이웃에 따라서 $Ctx = Right4x4SubBlockFlag + Bottom4x4SubBlockFlag$, 또는 유사한 공식들로 계산되어도 된다. 또, $Right4x4SubBlockFlag$ 는 (예컨대, 우측-이웃하는, 4×4 서브 블록이 적어도 하나의 년-제로 계수를 포함하는지 여부를 나타내는) 우측-이웃하는 서브 블록에 대한 코딩된 서브 블록 플래그를 나타낼 수도 있으며, $Bottom4x4SubBlockFlag$ 는 (예컨대, 최저부 (bottom)-이웃하는, 4×4 서브 블록이 적어도 하나의 년-제로 계수를 포함하는지 여부를 나타내는) 우측-이웃하는 서브 블록에 대한 코

딩된 서브 블록 플래그를 나타낼 수도 있다.

[0129] 도 10 은 비디오 블록의 각각의 영역에 대한 이웃- 또는 포지션-기반 컨텍스트들의 예시적인 할당을 예시하는 개념도이다. 도 10 에 예시된 바와 같이, 컨텍스트들의 하이브리드 유형이 예를 들어, 컨텍스트들이 이웃 기반일 수 있는 일부 영역들에 대해, 그리고 고정되거나 또는 포지션 기반일 수 있는 동일한 비디오 블록의 일부 영역들에 대해 역시 사용되어도 된다. 포지션-기반 접근법의 잠재적인 이점은 컨텍스트를 계수적 방식 (coefficient-wise manner) 으로 계산하는 것이 불필요하다는 점이다. 대신에, 비디오 코더는 그 영역에서의 모든 계수들이 동일한 컨텍스트를 갖도록, 컨텍스트를 영역에서의 모든 계수들에 대해 한 번 계산할 수도 있다. 도 10 은 복수의 영역들이 하나 이상의 서브 블록들의 각각의 세트를 포함하는 예를 나타낸다.

[0130] 좌표들 (x, y) 을 가진 계수에 대해, 영역들은 계수 포지션에 따라서 정의될 수 있다. 예를 들어, 조건 (x + y >= 임계치) 이 참이면, 비디오 코더는 대응하는 계수가 영역 (R2) 내에서 발생한다고 결정할 수도 있으며; 그렇지 않고, 그 조건이 참이 아니면, 비디오 코더는 대응하는 계수가 영역 (R1) 내에서 발생한다고 결정한다. 이와 유사하게, 좌표들이 4x4 서브 블록들에 기초하여 영역들에 할당될 수 있다. (X, Y) 좌표들을 가진 서브 블록에 대해, 영역들은 4x4 서브 블록 포지션에 따라서 정의될 수 있다. 예를 들어, 조건 (X + Y >= 임계치) 이 참이면, 비디오 코더는 대응하는 계수가 영역 (R2) 내에서 발생한다고 결정할 수도 있으며; 그렇지 않으면, 비디오 코더는 대응하는 계수가 영역 (R1) 내에서 발생한다고 결정할 수도 있다. 임계치는 4, 5, 6, 7 또는 8 과 동일한 정수와 같은, 일부 미리 정의된 값에 고정될 수도 있거나, 또는 비디오 블록, 예컨대 TU 사이즈에 의존할 수도 있다.

[0131] 이 방식으로, 도 10 은 변환 계수가 발생하는 영역에 기초하여, 그 영역한 포지션-기반 컨텍스트 정보 및 이웃-기반 컨텍스트 정보 중 하나를 사용하여, 변환 계수를 코딩하기 위한 컨텍스트를 비디오 코더가 결정하도록 구성될 수도 있는 예를 나타낸다. 특히, 변환 계수가 제 1 영역에 있으면, 비디오 코더는 제 1 컨텍스트 결정 접근법을 사용하여, 변환 계수를 코딩하기 위한 컨텍스트를 결정할 수도 있다. 변환 계수가 제 2 영역에 있으면, 비디오 코더는 제 2 컨텍스트 결정 접근법을 사용하여 변환 계수를 코딩하기 위한 컨텍스트를 결정할 수도 있으며, 여기서, 제 2 컨텍스트 결정 접근법은 제 1 컨텍스트 결정 접근법과 상이하고, 제 1 영역은 제 2 영역과 상이하다. 일 예에서, 제 1 및 제 2 영역들은 중첩하지 않는다. 또, 제 1 및 제 2 컨텍스트 결정 접근법들의 예들은 포지션-기반 컨텍스트 정보 및 이웃-기반 컨텍스트 정보의 사용을 포함한다.

[0132] 도 11 은 비디오 블록의 각각의 영역에 대한 컨텍스트 오프셋들의 예시적인 할당을 예시하는 개념도이다. 컨텍스트 모델은 상이한 영역들에 대해 별개일 수도 있지만, 그러나 여전히 컨텍스트 계산을 위해 동일한 방법을 사용한다. 즉, 비디오 코더는 변환 계수를 코딩하기 위한 컨텍스트를 계산하는 하나의 방법으로 구성될 수도 있지만, 변환 계수가 발생하는 영역에 기초하여 결정된 상이한 컨텍스트 모델들을 포함할 수도 있다.

[0133] 예를 들어, 컨텍스트는 이웃에 기초하여 계산되지만, 오프셋을 사용하는 상이한 영역들에 대해 계산될 수도 있다. 각각의 영역에 대한 오프셋은 비디오 블록 사이즈, 비디오 블록 또는 서브 블록에서의 계수 포지션, 및 비디오 블록에서의 서브 블록 포지션 중 하나 이상으로 고정되거나 또는 의존할 수도 있다. 도 11 의 영역 (R1) 은 비디오 블록의 변환 계수들의 하나 이상의 상부-좌측 4x4 서브 블록들을 포함하는 영역의 또 다른 예를 나타내지만, 도 11 의 영역 (R2) 는 영역 (R1) 외부의 비디오 블록의 변환 계수들을 포함하는 영역의 또 다른 예를 나타낸다. 도 11 은 또한 복수의 영역들이 하나 이상의 서브 블록들의 각각의 세트들을 포함하는 일 예이다.

[0134] 오프셋에 의해, 컨텍스트는 수식 (2) 에 따라서 계산될 수도 있다.

$$Ctx = offset(region) + \sum_{p \in S} (coef_p \neq 0) \tag{2}$$

[0135] 이의 대안으로, 비디오 코더는 Ctx 를 입력으로서 사용하는 함수, 예를 들어, Ctx = (Ctx + 1) >> 1 에 따라서 컨텍스트를 계산할 수도 있다.

[0137] 영역-기반 오프셋들의 일 예가 도 11 상에 도시되며, 여기서, 영역들 (R1 및 R2) 은 4x4 서브 블록들에 기초하여 정의되며, 오프셋들은 영역들 (R1 및 R2) 에 대해 상이하다. 오프셋 값들 (offset1 및 offset2) 은 임의의 정수들일 수 있으며, 예를 들어 offset1 = 0, offset2 = 3 이다. 다른 예에서, 영역들로의 다른

분할들이 또한 가능하며, 2개보다 많은 영역들로의 분할들이 또한 가능하다.

- [0138] 도 12 는 기존 컨텍스트 모델들과 연관되는 TU 사이즈들에 기초한, 2 이상의 영역들로의 비디오 블록의 예시적인 임베딩되는 분할을 예시하는 개념도이다. 현재의 HM 에서 TU 의 여러 사이즈들 (4×4, 8×8, 16×16 및 32×32) 이 존재하므로, 더 큰 블록들의 분할은 도 12 에 예시된 바와 같이, 분할의 임베딩되는 스타일을 사용하여, 더 작은 TU 사이즈들에 따라서 이루어질 수 있다. 임베딩되는 분할에 대해, 컨텍스트 계산의 방법이 공유될 수도 있으며, 컨텍스트 모델 자체가 공유될 수도 있다.
- [0139] 예를 들어, TU 사이즈 32×32 에 대해, 4×4 TU 를 나타내는 영역 (R1) 에서, 컨텍스트 계산은 사이즈 4×4 의 실제 TU 에 대한 컨텍스트 계산을 위해 동일한 방법을 사용할 수도 있다. 게다가, 컨텍스트 모델이 사이즈 4×4 의 TU 와 사이즈 32×32 의 TU 의 R1 사이에 공유될 수도 있거나, 또는 오프셋이 사이즈 4×4 의 TU 에 대한 컨텍스트 모델에 적용될 수도 있다. R2 에 대해, 컨텍스트 계산 방법은 사이즈 8×8 의 TU 와 사이즈 32×32 의 TU 의 R2 사이에 공유될 수도 있다. R3 는 16×16 TU 영역을 나타내지만, R4 는 32×32 TU 영역을 나타낸다. 이 방법의 잠재적인 이점은 동일한 유닛들이 컨텍스트 계산들을 위해 사용될 수도 있으며, 임베딩되는 영역들과 TU들 사이의 추가적인 상관성이 고려될 수 있다는 점이다.
- [0140] 이의 대안으로, 임베딩되는 스타일 분할을 사용하여, 일부 유의도 맵 컨텍스트 모델들은 모든 TU들 또는 TU들의 일부 그룹 사이의 전용 포지션들에 대해 공유될 수도 있다. 예를 들어, DC 계수들에 대응하는, 컨텍스트 모델은 4×4 로부터 32×32 까지의 사이즈들을 가진 모든 TU들 사이에 공유될 수도 있다. 또 다른 예로서, 높은 주파수 계수들과 관련되는, 컨텍스트 모델은 모든 TU들 사이에 공유될 수도 있다. 이들 경우들에서, 사이즈 32×32 의 TU 에서, 4×4 TU 을 나타내는 영역 (R1) 은 사이즈들 4×4, 8×8, 16×16, 32×32 등 중 임의의 사이즈를 갖는 TU들과 동일한, DC 계수들 및/또는 높은 주파수 계수들에 대한 컨텍스트 모델을 사용할 수도 있다.
- [0141] 추가 예로서, 모든 TU들 사이에 공유하는 대신에, 위에서 설명한 계수들 (예컨대, DC 및/또는 높은 주파수 계수들) 의 컨텍스트 모델은 오직 모든 TU들의 서브세트 또는 그룹 사이에 공유될 수도 있다. 예를 들어, 계수의 컨텍스트 모델은 4×4 및 8×8 TU들과 같은, 2개의 사이즈들의 TU들 사이에 오직 공유될 수도 있다. 이 경우, 사이즈 32×32 의 TU 에서, 4×4 TU 을 나타내는 영역 (R1) 은 사이즈 4×4 및 8×8 를 갖는 TU들과 동일한, DC 계수들 및/또는 높은 주파수 계수들에 대한 컨텍스트 모델을 사용할 수도 있다.
- [0142] 이 방식으로, 도 12 의 예는 비디오 인코더 (20) 또는 비디오 디코더 (30) 와 같은, 비디오 코더가 비디오 블록의 복수의 영역들 중에서 변환 계수가 발생하는 영역을 결정하도록 구성될 수도 있는 예를 나타내며, 여기서, 영역들 각각은 복수의 변환 유닛 (TU) 사이즈들의 각각의 하나에 대응하며, 비디오 코더는 영역과 그 영역과 동일한 사이즈를 갖는 TU 사이에 공유되는 컨텍스트를 선택함으로써 컨텍스트를 결정한다.
- [0143] 도 12 는 또한, 비디오 인코더 (20) 또는 비디오 디코더 (30) 와 같은, 비디오 코더가 비디오 블록의 복수의 영역들 중에서 변환 계수가 발생하는 영역을 결정하도록 구성될 수도 있는 예를 나타내며, 여기서 영역들 각각은 복수의 변환 유닛 (TU) 사이즈들의 각각의 하나에 대응하며, 여기서 컨텍스트를 결정하기 위해 비디오 코더는 상이한 사이즈들의 2 이상의 TU들 사이에서의 변환 계수들의 전용 포지션들에 대한 공유된 컨텍스트를 선택하며, 영역은 상이한 사이즈들의 2 이상의 TU들 중 하나와 동일한 사이즈를 갖는다. 변환 계수들의 전용 포지션들에 대한 공유된 컨텍스트는 DC 계수들 및 상이한 사이즈들의 2 이상의 TU들 사이에 공유되는 높은 주파수 계수들 중 하나에 대한 컨텍스트를 포함할 수도 있다. 이에 추가적으로 또는 대안적으로, 변환 계수들의 전용 포지션들에 대한 공유된 컨텍스트는 4×4 변환 계수들의 사이즈를 갖는 제 1 TU 및 8×8 변환 계수들의 사이즈를 갖는 제 2 TU 사이에서의 공유된 컨텍스트를 포함할 수도 있다.
- [0144] 도 13a 및 도 13b 는 2 이상의 영역들로의 비디오 블록의 예시적인 분할들을 예시하는 개념도들이다. 영역들이 정사각형, 예컨대, 4×4, 서브 블록들에 기초하는 예들과 관련하여 위에서 설명한 방법과 유사한 방법으로, 본 개시물의 기법들은 또한 직사각형 형상의 서브 블록들에 기초하여, 비디오 블록, 예컨대 TU 를 2 이상의 영역들로 분할하는 분류 방법 (classification method) 을 기술한다. 예를 들어, 2×8 및 8×2 서브 블록들이 도 13a 및 도 13b 상에 나타난 바와 같은 계수들 스캐닝에 따라서 8×8 비디오 블록에 대해 사용될 수 있다. 이 예에서, 비디오 코더는 도 13a 에 나타난 블록에서의 계수들에 대해 수평 스캐닝을, 그리고 도 13b 에 나타난 블록에 대해 수직 스캐닝을 적용한다. 도 13a 및 도 13b 에 예시된 예들에서, 하나의 정사각형 블록은 하나의 단일 계수를 나타내며, 전체 비디오 블록의 사이즈는 8×8 이다.
- [0145] 본 개시물의 기법들에 따르면, 비디오 블록은 상이한 직사각형의 영역들, 예컨대 R1, R2, R3, 및 R4 로 분할될 수도 있다. 상이한 직사각형의 영역들의 각각은 상이한 컨텍스트 할당을 가질 수도 있다. 예를 들

어, 일부 영역들에 대해, 고정된 콘텍스트가 사용될 수도 있다. 이들 영역들은 도 13a 및 도 13b 에 나타나고 위에서 설명된 직사각형의 (예를 들어, 2×8 또는 8×2) 서브 블록들에 기초하여 형성될 수도 있다. 예를 들어, 콘텍스트는 비디오 블록에서의 계수 포지션에 따라서, 또는 이 계수를 포함하는 직사각형의 서브 블록의 포지션에 따라서 정의될 수 있다.

[0146] 이의 대안으로, 콘텍스트 모델은 이웃 직사각형 형상의 서브 블록들에 정의되어도 된다. 예를 들어, 동일한 직사각형의 서브 블록 내의 모든 계수는 하나 또는 여러 콘텍스트들을 사용할 수 있다. 게다가, 이웃하는 직사각형의 서브 블록의 계수들은 또한 하나 또는 여러 콘텍스트들을 사용할 수 있다. 그러나, 하나의 직사각형의 서브 블록의 콘텍스트들은 이전 직사각형의 서브 블록 기반 콘텍스트들과 상이할 수도 있다. 콘텍스트들의 하이브리드 유형이 예를 들어, 콘텍스트들이 이웃 기반일 수도 있는 일부 영역들에 대해, 그리고 고정되거나 또는 포지션 기반일 수 있는 동일한 비디오 블록의 일부 영역들에 대해, 역시 사용되어도 된다. 포지션 기반 접근법의 이점은 콘텍스트를 계수적으로 계산할 필요가 없어, 그 영역에 대해 한번 수행될 수 있다는 점이다. 또한, 분할은 계수 포지션들 또는 이 계수를 포함하는 직사각형의 서브 블록의 포지션에 의존해도 된다.

[0147] (x, y) 좌표들을 가진 계수에 대해, 영역들은 계수 포지션에 따라서 정의될 수 있다. 예를 들어, 조건 $(x + y \geq \text{임계치})$ 이 참이면, 이 계수는 영역 (R2) 에 할당될 수도 있으며; 그렇지 않으면, 영역 (R1) 에 할당될 수도 있다. 유사한 방법으로, 이것은 (X, Y) 좌표들을 가진 서브 블록에 대해 직사각형 형상의 서브 블록에 기초하여 이루어질 수 있으며, 영역들은 직사각형의 서브 블록 포지션에 따라서 정의될 수 있다. 예를 들어, 조건 $(X + Y \geq \text{임계치})$ 이 참이면, 이 계수는 영역 (R2) 에 할당될 수도 있으며, 그렇지 않으면, R1 에 할당될 수도 있다. 임계치는 (예컨대, 0 또는 1 과 동일한) 정수와 같은, 일부 미리 정의된 값에 고정될 수도 있거나, 또는 TU 사이즈에 의존해도 된다.

[0148] 이의 대안으로, 콘텍스트 모델은 상이한 영역들에 대해 상이할 수도 있지만, 콘텍스트 계산을 위해 동일한 방법을 여전히 사용할 수도 있다. 예를 들어, 콘텍스트는 이웃에 기초하지만, 오프셋을 사용하는 상이한 영역들에 대해 계산될 수도 있다. 오프셋은 고정되거나, 비디오 블록 사이즈에 의존하거나, 또는 비디오 블록 및/또는 직사각형의 서브 블록에서의 계수 포지션, 비디오 블록에서의 현재의 계수를 포함하는 직사각형의 서브 블록의 포지션, 또는 이들 조건들의 임의의 조합 중 하나 이상에 의존할 수 있다.

[0149] 오프셋에 의해, 콘텍스트는 수식 (3) 에 따라서 계산될 수도 있다.

$$C_{tx} = \text{offset}(\text{region}) + \sum_{p \in S} (\text{coef}_p \neq 0) \tag{3}$$

[0150] 이의 대안으로, 콘텍스트는 C_{tx} 를 입력으로서 사용하는 함수, 예를 들어 $C_{tx} = (C_{tx}+1) \gg 1$ 에 따라서 계산될 수도 있다.

[0151] 도 14a 및 도 14b 는 비디오 블록의 각각의 영역에 대한 콘텍스트 오프셋들의 예시적인 할당을 예시하는 개념도들이다. 이들 예들에서, 영역들 (R1 및 R2) 은 직사각형의 서브 블록들 및 스캐닝 방향에 기초하여 정의되며, 오프셋들은 영역들 (R1 및 R2) 에 대해 상이하다. 오프셋 값들 (offset1 및 offset2) 은 임의의 정수들일 수 있으며, 예를 들어, offset1 = 0, offset2 = 3 이다. 영역들로의 다른 분할들이 또한 가능하다. 예를 들어, 다수의 영역들은 2개보다 많을 수 있다. 계수 스캐닝 방향들에 따라서, 2×8 및 8×2 직사각형의 서브 블록들이 본 개시물에서 일 예로서 사용되었다는 점에 유의해야 한다. 유사한 방법들이 사이즈 M×N 를 가진 다른 직사각형-형상의 서브 블록들에 대해 제한 없이 사용될 수 있다.

[0152] 일반적으로, 본 개시물은 비디오 블록들의 대각선 기반의, 정사각형, 예컨대 4×4, 서브 블록 기반, 및 직사각형의, 예컨대, 2×8 및 8×2, 서브 블록 기반의 분할을 기술한다. 다른 예들에서, 다른 유형들의 분할이 가능하며, 분할은 상이한 사이즈들을 가진, 여러 형태들, 예컨대 직사각형, 정사각형, 삼각형 등에 기초하여 플렉서블할 수 있다. 본 개시물은 또한 비디오 블록들을 임의 개수의 영역들로 분할하는 것을 기술한다. 본 개시물은 정사각형 서브 블록, 직사각형의 서브 블록들에 기초하여, 또는 비디오 블록의 대각선 분할들과 같은 다른 그룹핑에 기초하여, 계수들을 영역들로 그룹핑하는 것을 추가로 기술한다. 위에서 설명된 임계치들 및 오프셋들은 또한 일 예로서 제공되며, 다른 값들 또는 이웃 의존성들이 사용될 수 있다.

[0153] 본 개시물에서 설명된 바와 같은 유사한 기법들이 비-정사각형 변환 유닛들 또는 다른 형태들의 유닛들에 대해 사용될 수 있다. 설명되는 기법들은 유의도 맵 코딩에, 그리고 변환 계수들의 빈 코딩 및 다른 구문에, 제한 없이 적용될 수도 있다. 게다가, 본 개시물은 일반적으로 비디오 블록들을 TU 블록들로서

지칭하며, 그러나 본 기법들은 TU들, PU들, CU들, LCU들 또는 블록들의 다른 그룹들 중 임의의 것에 적용될 수도 있다.

- [0155] 도 15 는 현재의 블록을 인코딩하는 예시적인 방법을 예시하는 플로우차트이다. 현재의 블록은 현재의 CU 또는 현재의 CU 의 부분을 포함할 수도 있다. 비디오 인코더 (20) (도 1 및 도 2) 에 대해 설명되었지만, 다른 디바이스들이 도 15 의 방법과 유사한 방법을 수행하도록 구성될 수도 있는 것으로 이해되어야 한다.
- [0156] 이 예에서, 비디오 인코더 (20) 는 먼저 현재의 블록을 예측한다 (150). 예를 들어, 비디오 인코더 (20) 는 현재의 블록에 대한 하나 이상의 예측 유닛들 (PU들) 을 계산할 수도 있다. 비디오 인코더 (20) 는 그 후 현재의 블록에 대한 잔여 블록을 계산하여, 예컨대 변환 유닛 (TU) 을 생성할 수도 있다 (152). 잔여 블록을 계산하기 위해, 비디오 인코더 (20) 는 원래, 코딩되지 않은 블록과 현재의 블록에 대한 예측된 블록 사이의 차이 (즉, 픽셀 단위 차이들) 를 계산할 수도 있다. 비디오 인코더 (20) 는 그 후 잔여 블록의 계수들을 변환하여 양자화할 수도 있다 (154). 다음으로, 비디오 인코더 (20) 는 잔여 블록의 양자화된 변환 계수들을 스캐닝할 수도 있다 (156).
- [0157] 스캐닝 동안, 비디오 인코더 (20) 는 현재의 계수가 발생하는 영역을 결정할 수도 있으며, 이 방식으로 비디오 인코더 (20) 는 여러 계수들이 발생하는 영역들을 결정할 수도 있다 (158). 본 개시물의 기법들에 따르면, 비디오 인코더 (20) 는 예를 들어, 계수들의 포지션들 또는 계수들이 발생하는 서브 블록들의 포지션들에 기초하여, 계수들이 발생하는 영역들을 결정할 수도 있다. 비디오 인코더 (20) 는 도 9 내지 도 14 와 관련하여 설명된 기법들, 또는 다른 유사한 기법들 중 임의의 기법을 사용하여 영역들을 결정할 수도 있다. 예를 들어, 도 9a 에 나타난 바와 같이, 비디오 인코더 (20) 는 계수가 하나 이상의 서브 블록들을 포함하는 제 1 영역, 또는 제 1 영역 외부의 서브 블록들을 포함하는 제 2 영역에서 발생하는지 여부를 결정하도록 구성될 수도 있다.
- [0158] 비디오 인코더 (20) 는 그 영역들에 기초하여, 계수들을 엔트로피 인코딩하기 위한 컨텍스트들을 추가로 결정할 수도 있다 (160). 즉, 비디오 인코더 (20) 는 계수가 발생하는 영역에 기초하여, 각각의 계수에 대해, 계수를 인코딩하기 위한 컨텍스트를 결정할 수도 있다. 예를 들어, 위에서 설명한 바와 같이, 비디오 인코더 (20) 는 계수가 발생하는 영역에 기초하여, 블록에서의 계수의 포지션, 블록에서의 계수를 포함하는 서브 블록의 포지션, 계산된 컨텍스트에 적용된 오프셋, 또는 기타 등등에 기초하여 컨텍스트를 결정할 수도 있다.
- [0159] 이와 유사하게, 비디오 인코더 (20) 는 결정된 컨텍스트들을 사용하여 계수들을 엔트로피 인코딩할 수도 있다 (162). 특히, 비디오 인코더 (20) 는 컨텍스트를 사용하여 계수들을 나타내는 하나 이상의 구문 엘리먼트들을 엔트로피 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 계수들에 대한 유의도 정보, 유의 계수들에 대한 레벨 정보, 및/또는 유의 계수들에 대한 부호 정보 중 하나 이상을 엔트로피 인코딩할 수도 있다. 유의도 정보는 significant_coeff_flag 데이터를 포함할 수도 있다. 레벨 정보는 coeff_abs_level_greater1_flag, coeff_abs_level_greater2_flag, 및 coeff_abs_level_remaining 를 포함할 수도 있다. 부호 정보는 coeff_sign_flag 를 포함할 수도 있다. 비디오 인코더 (20) 는 그 후 계수들에 대한 엔트로피 인코딩된 데이터를 출력할 수도 있다 (164).
- [0160] 이 방식으로, 도 15 의 방법은 변환 계수가 발생하는 비디오 블록의 영역에 기초하여 비디오 블록의 변환 계수를 코딩하기 위한 컨텍스트를 결정하는 단계, 및 결정된 컨텍스트를 사용하여 변환 계수를 엔트로피 코딩하는 단계를 포함하는 방법의 일 예를 나타낸다. 더욱이, 영역은 비디오 블록의 변환 계수들의 하나 이상의 상부-좌측 4×4 서브 블록들을 포함하는 제 1 영역, 및 제 1 영역 외부의 비디오 블록의 변환 계수들을 포함하는 제 2 영역 중 하나를 포함할 수도 있다.
- [0161] 도 16 은 비디오 데이터의 현재의 블록을 디코딩하는 예시적인 방법을 예시하는 플로우차트이다. 현재의 블록은 현재의 CU 또는 현재의 CU 의 부분을 포함할 수도 있다. 비디오 디코더 (30) (도 1 및 도 3) 와 관련하여 설명되지만, 다른 디바이스들이 도 16 의 방법과 유사한 방법을 수행하도록 구성될 수도 있는 것으로 이해되어야 한다.
- [0162] 비디오 디코더 (30) 는 예컨대, 인트라- 또는 인터-예측 모드를 사용하여 현재의 블록을 예측하여 (200), 현재의 블록에 대한 예측된 블록을 계산할 수도 있다. 비디오 디코더 (30) 는 또한 현재의 블록에 대응하는 잔여 블록의 계수들의 계수들에 대한 엔트로피 인코딩된 데이터와 같은, 현재의 블록에 대한 엔트로피 인코딩된 데이터를 수신할 수도 있다 (202).

- [0163] 본 개시물의 기법들에 따르면, 비디오 디코더 (30) 는 예컨대, 역 스캐닝 및 엔트로피 디코딩 프로세스 동안, 계수들이 발생할 영역들을 결정할 수도 있다 (204). 즉, 비디오 디코더 (30) 는 스캐닝 순서에서 이전에 디코딩된 변환 계수 및 다음 유의한 변환 계수의 포지션에 기초하여, 다음 변환 계수의 포지션을 결정할 수도 있다. 비디오 디코더 (30) 는 이 포지션이 발생하는 블록의 영역을 추가로 결정할 수도 있다. 비디오 디코더 (30) 는 유사하게 계수들 각각에 대한 영역들을 유사한 방법으로 결정할 수도 있다.
- [0164] 더욱이, 비디오 디코더 (30) 는 예를 들어, 계수들의 포지션들 또는 계수들이 발생할 서브 블록들의 포지션들에 기초하여, 계수들이 발생할 영역들을 결정할 수도 있다. 비디오 디코더 (30) 는 도 9 내지 도 14 와 관련하여 설명된 기법들, 또는 다른 유사한 기법들 중 임의의 기법을 사용하여 영역들을 결정할 수도 있다. 예를 들어, 도 9a 에 나타낸 바와 같이, 비디오 디코더 (30) 는 계수가 하나 이상의 서브 블록들을 포함하는 제 1 영역, 또는 제 1 영역 외부의 서브 블록들을 포함하는 제 2 영역에서 발생하는지 여부를 결정하도록 구성될 수도 있다.
- [0165] 더욱이, 비디오 디코더 (30) 는 결정된 영역들에 기초하여, 계수들을 디코딩하기 위한 콘텍스트들을 결정할 수도 있다 (206). 즉, 비디오 디코더 (30) 는 계수가 발생하는 영역에 기초하여, 각각의 계수에 대해, 계수를 디코딩하기 위한 콘텍스트를 결정할 수도 있다. 예를 들어, 위에서 설명한 바와 같이, 비디오 디코더 (30) 는 계수가 발생할 영역에 기초하여, 블록에서의 계수의 포지션, 블록에서의 계수를 포함하는 서브 블록의 포지션, 계산된 콘텍스트에 적용된 오프셋, 또는 기타 등등에 기초하여 콘텍스트를 결정할 수도 있다.
- [0166] 비디오 디코더 (30) 는 결정된 콘텍스트들을 사용하여 엔트로피 코딩된 데이터를 엔트로피 디코딩하여, 블록의 계수들을 재생할 수도 있다 (208). 특히, 비디오 디코더 (30) 는 콘텍스트를 사용하여 계수들을 나타내는 하나 이상의 구문 엘리먼트들을 엔트로피 디코딩할 수도 있다. 예를 들어, 비디오 디코더 (30) 는 계수들에 대한 유의도 정보, 유의 계수들에 대한 레벨 정보, 및/또는 유의 계수들에 대한 부호 정보 중 하나 이상을 엔트로피 디코딩할 수도 있다. 유의도 정보는 significant_coeff_flag 데이터를 포함할 수도 있다. 레벨 정보는 coeff_abs_level_greater1_flag, coeff_abs_level_greater2_flag, 및 coeff_abs_level_remaining 를 포함할 수도 있다. 부호 정보는 coeff_sign_flag 를 포함할 수도 있다. 비디오 디코더 (30) 는 그 후 그들의 각각의 포지션들에서 그 디코딩된 변환 계수들을 포함하도록 블록 (예컨대, TU) 을 재생할 수도 있다 (210). 즉, 위에서 설명한 바와 같이, 비디오 디코더 (30) 는 그 재생된 계수들을 역스캐닝하여, 양자화된 변환 계수들의 블록을 생성할 수도 있다.
- [0167] 비디오 디코더 (30) 는 그 후 그 계수들을 역양자화하고 역변환하여, 잔여 블록을 생성할 수도 있다 (212). 비디오 디코더 (30) 는 궁극적으로, 그 예측된 블록과 잔여 블록을 결합함으로써, 현재의 블록을 디코딩할 수도 있다 (214). 즉, 비디오 디코더 (30) 는 예측된 블록의 픽셀 값들을 잔여 블록의 동일 장소에 배치된 픽셀 값들과 수학적으로 합산하여, 원래 블록을 디코딩하고 재생할 수도 있다.
- [0168] 이 방식으로, 도 16 의 방법은 변환 계수가 발생하는 비디오 블록의 영역에 기초하여 비디오 블록의 변환 계수를 코딩하기 위한 콘텍스트를 결정하는 단계, 및 결정된 콘텍스트를 사용하여 변환 계수를 엔트로피 코딩하는 단계를 포함하는 방법의 일 예를 나타낸다. 더욱이, 영역은 비디오 블록의 변환 계수들의 하나 이상의 상부-좌측 4×4 서브 블록들을 포함하는 제 1 영역, 및 제 1 영역 외부의 비디오 블록의 변환 계수들을 포함하는 제 2 영역 중 하나를 포함할 수도 있다.
- [0169] 하나 이상의 예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현되는 경우, 그 기능들은 하나 이상의 명령들 또는 코드들로서, 컴퓨터 판독 가능 매체 상에 저장되거나 또는 컴퓨터 판독 가능 매체를 통해서 송신될 수도 있으며, 하드웨어-기반의 프로세싱 유닛에 의해 실행될 수도 있다. 컴퓨터 판독 가능 매체는 컴퓨터 판독 가능 저장 매체들을 포함할 수도 있으며, 이 컴퓨터 판독 가능 저장 매체들은 데이터 저장 매체와 같은 유형의 매체, 또는 예컨대, 통신 프로토콜에 따라서 한 장소로부터 다른 장소로의 컴퓨터 프로그램의 전송을 용이하게 하는 임의의 매체를 포함한 통신 매체들에 대응한다. 이 방식으로, 컴퓨터 판독 가능 매체들은 일반적으로 (1) 비일시성 유형의 컴퓨터 판독 가능 저장 매체 또는 (2) 신호 또는 캐리어 파와 같은 통신 매체에 대응할 수도 있다. 데이터 저장 매체는 본 개시물에서 설명하는 기법들의 구현을 위한 명령들, 코드 및/또는 데이터 구조들을 추출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 가용 매체들일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터 판독 가능 매체를 포함할 수도 있다.
- [0170] 일 예로서, 이에 한정하지 않고, 이러한 컴퓨터 판독 가능 저장 매체는 RAM, ROM, EEPROM, CD-ROM 또는 다른 광디스크 스토리지, 자기디스크 스토리지, 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 원하는 프

로그램 코드를 명령들 또는 데이터 구조들의 형태로 저장하는데 사용될 수 있고 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 접속이 컴퓨터 판독가능 매체로 적절히 지칭된다. 예를 들어, 동축 케이블, 광섬유 케이블, 이중 권선, 디지털 가입자 회선 (DSL), 또는 무선 기술들 예컨대 적외선, 무선, 및 마이크로파를 사용하여 명령들이 웹사이트, 서버, 또는 다른 원격 소스로부터 송신되는 경우, 동축 케이블, 광섬유 케이블, 이중 권선, DSL, 또는 무선 기술들, 예컨대 적외선, 무선, 및 마이크로파가 그 매체의 정의에 포함된다. 그러나, 컴퓨터 판독가능 저장 매체 및 데이터 저장 매체는 접속부들, 캐리어 파들, 신호들, 또는 다른 일시성 매체를 포함하지 않고, 그 대신에, 비-일시성 유형의 저장 매체로 송신되는 것으로 해석되어야 한다. 디스크 (disk) 및 디스크 (disc) 는, 본원에서 사용할 때, 콤팩트 디스크 (CD), 레이저 디스크, 광 디스크, 디지털 다기능 디스크 (DVD), 플로피 디스크 및 Blu-ray 디스크를 포함하며, 디스크들 (disks) 은 데이터를 자기적으로 보통 재생하지만, 디스크들 (discs) 은 레이저로 데이터를 광학적으로 재생한다. 앞에서 언급한 것들의 조합들이 또한 컴퓨터 판독가능 매체들의 범위 내에 포함되어야 한다.

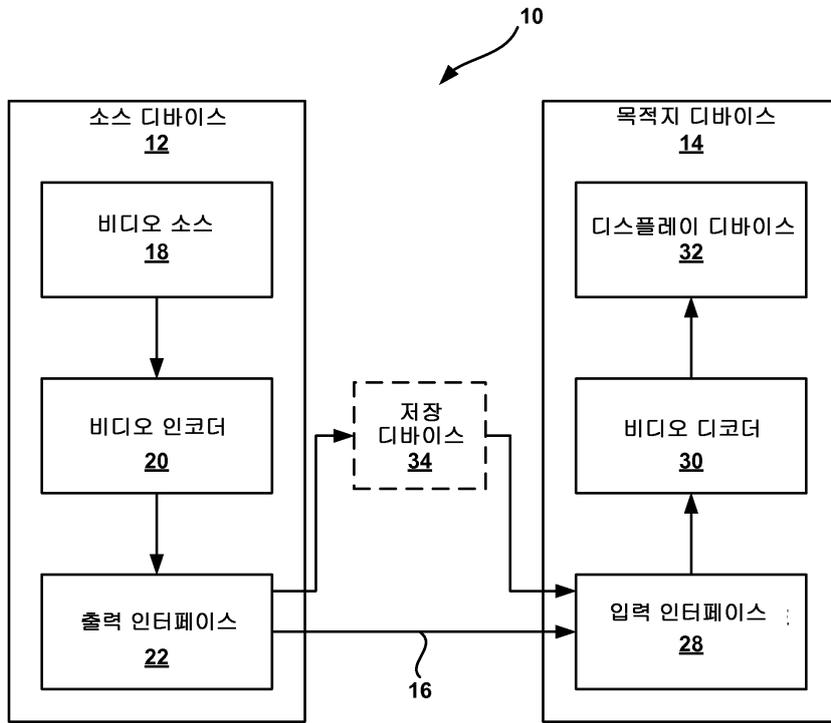
[0171] 명령들은 하나 이상의 디지털 신호 프로세서들 (DSPs), 범용 마이크로프로세서들, 주문형 집적회로들 (ASICs), 필드 프로그래밍가능 로직 어레이들 (FPGAs), 또는 다른 등가의 통합 또는 이산 로직 회로와 같은, 하나 이상의 프로세서들에 의해 실행될 수도 있다. 따라서, 용어 "프로세서" 는, 본원에서 사용될 때 전술한 구조 중 임의의 구조 또는 본원에서 설명하는 기법들의 구현에 적합한 임의의 다른 구조를 지칭할 수도 있다. 게다가, 일부 양태들에서, 본원에서 설명하는 기능은 전용 하드웨어 및/또는 인코딩 및 디코딩을 위해 구성되는 소프트웨어 모듈들 내에 제공되거나, 또는 결합된 코덱에 포함될 수도 있다. 또한, 이 기법들은 하나 이상의 회로들 또는 로직 엘리먼트들로 전적으로 구현될 수 있다.

[0172] 본 개시물의 기법들은 무선 핸드셋, 집적 회로 (IC) 또는 IC들의 세트 (예컨대, 칩 세트) 를 포함한, 매우 다양한 디바이스들 또는 장치들로 구현될 수도 있다. 개시한 기법들을 수행하도록 구성되는 디바이스들의 기능적 양태들을 강조하기 위해서 여러 컴포넌트들, 모듈들, 또는 유닛들이 본 개시물에서 설명되지만, 상이한 하드웨어 유닛들에 의한 실현을 반드시 필요로 하지는 않는다. 더 정확히 말하면, 위에서 설명한 바와 같이, 여러 유닛들이 코덱 하드웨어 유닛에 결합되거나 또는 적합한 소프트웨어 및/또는 펌웨어와 함께, 위에서 설명한 바와 같은 하나 이상의 프로세서들을 포함한, 상호작용하는 하드웨어 유닛들의 컬렉션으로 제공될 수도 있다.

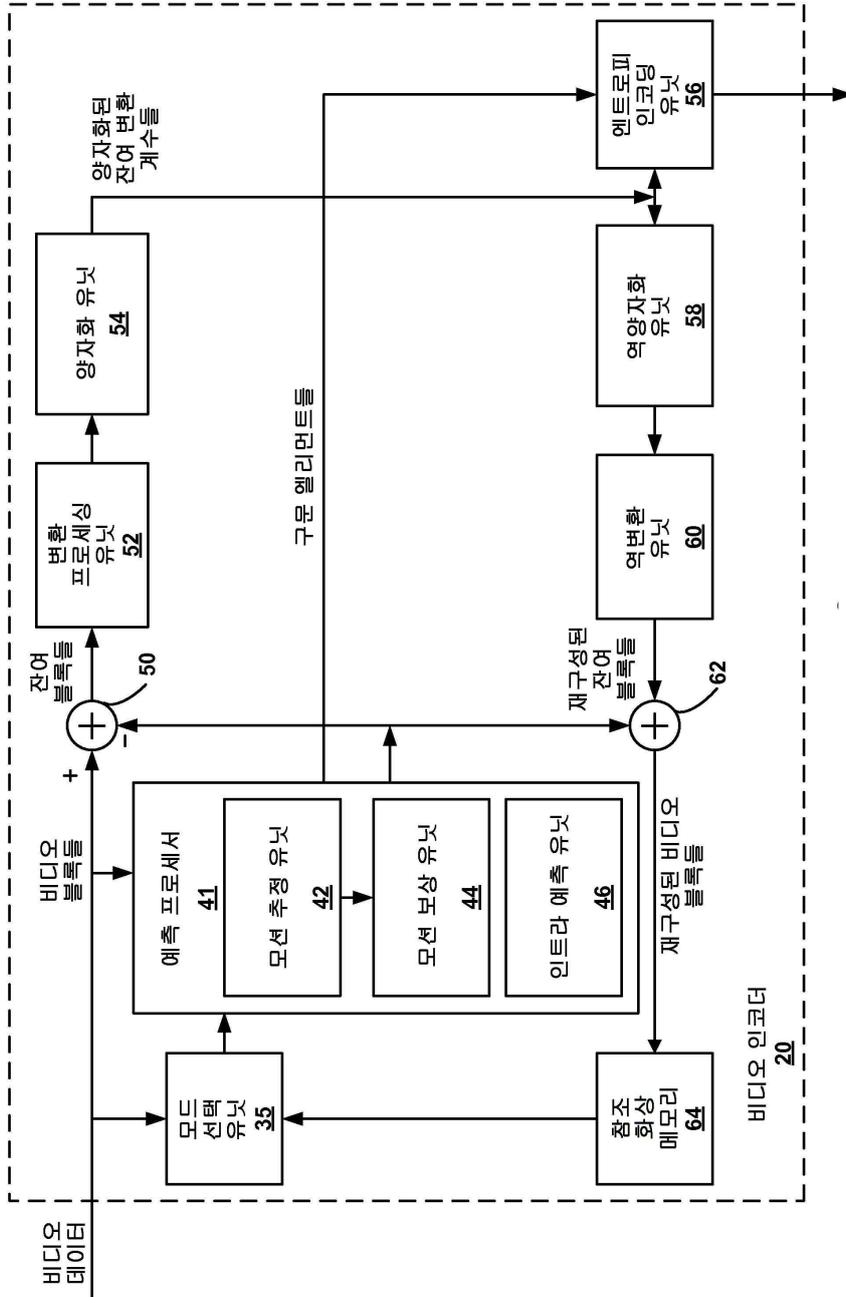
[0173] 여러 예들이 설명되었다. 이들 및 다른 예들은 다음 청구항들의 범위 이내이다.

도면

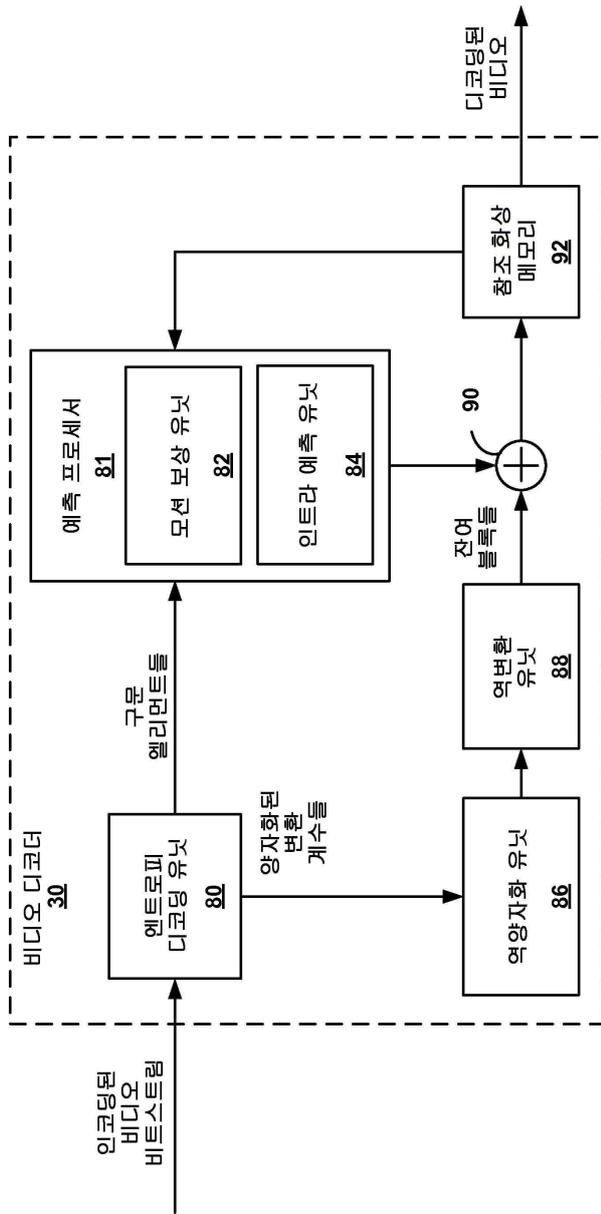
도면1



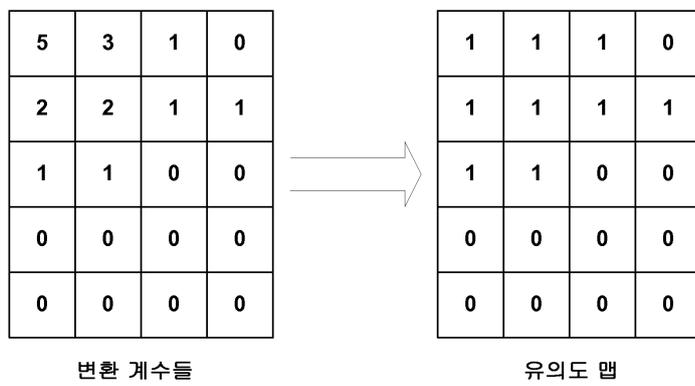
도면2



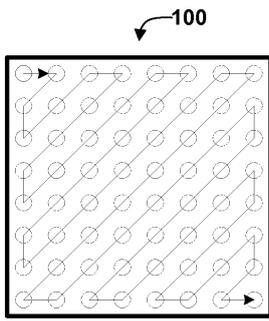
도면3



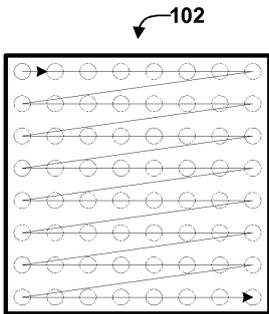
도면4



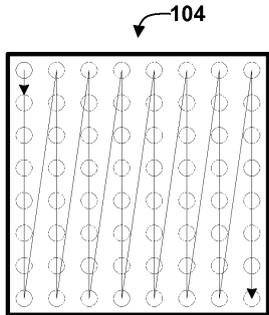
도면5a



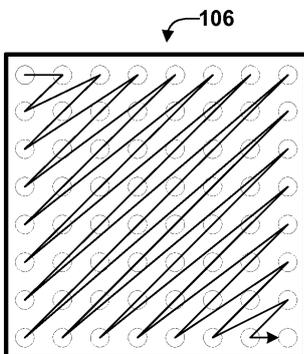
도면5b



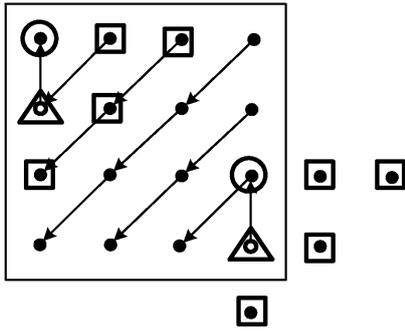
도면5c



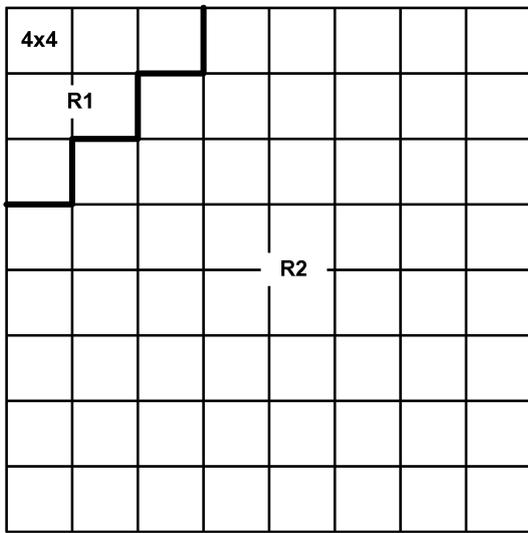
도면5d



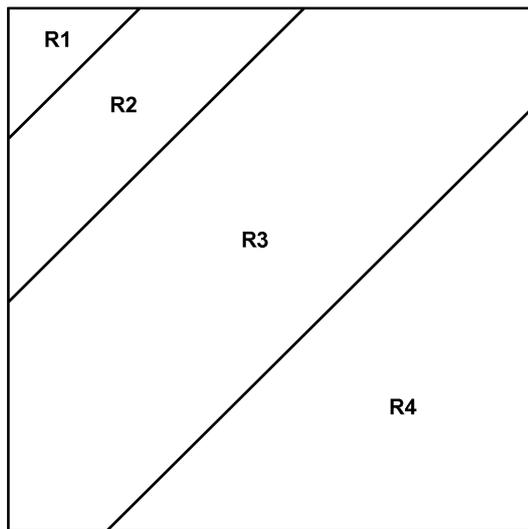
도면8b



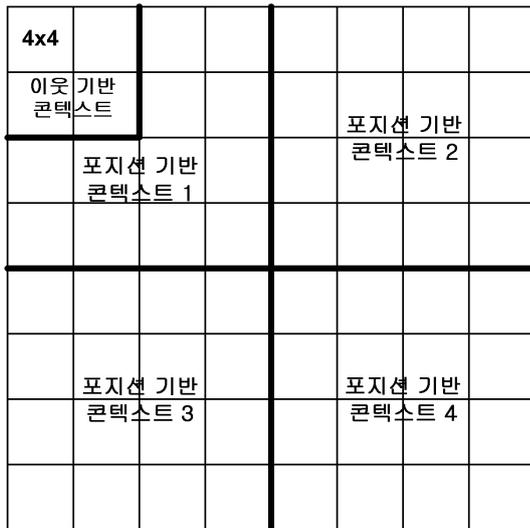
도면9a



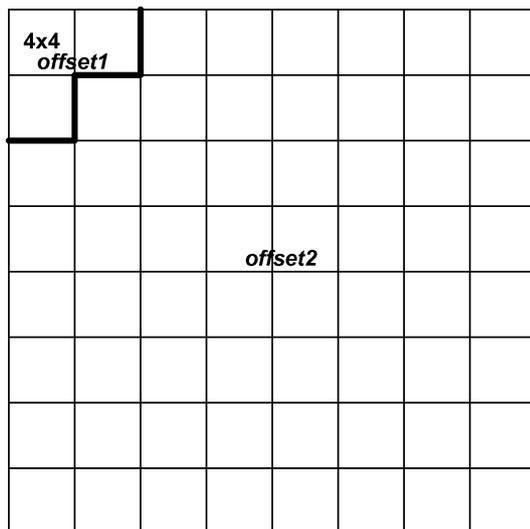
도면9b



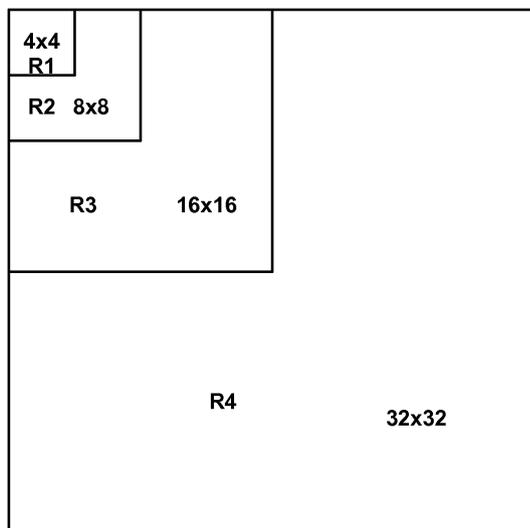
도면10



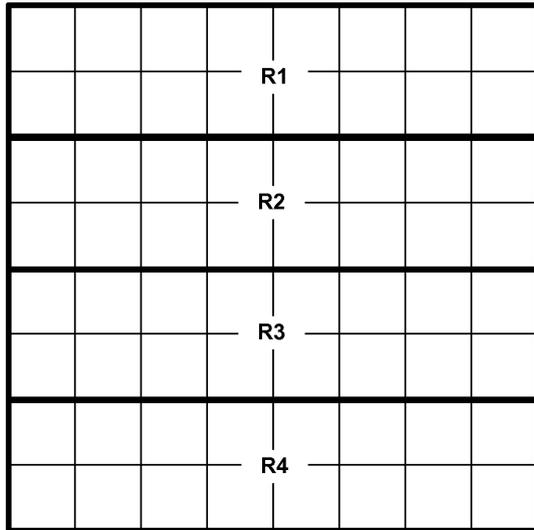
도면11



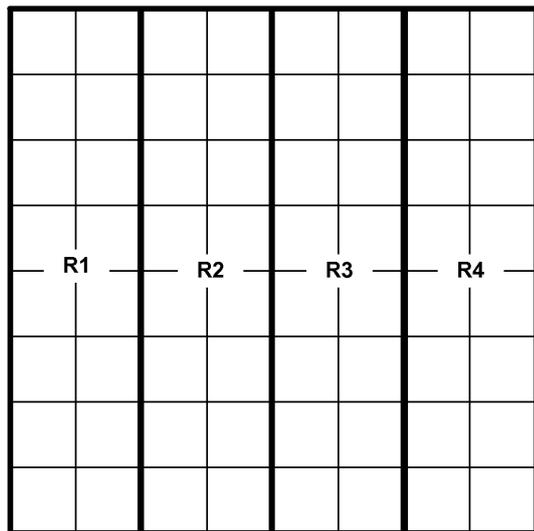
도면12



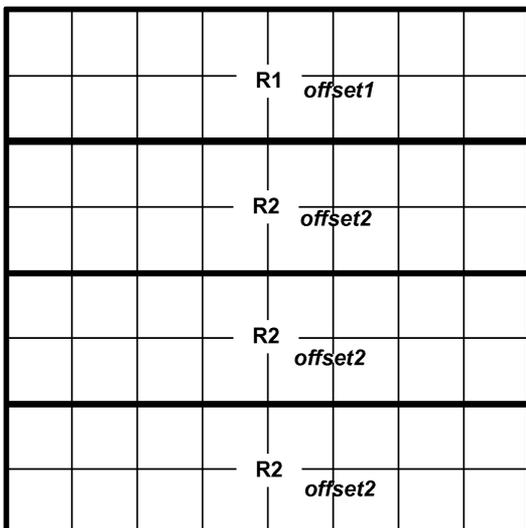
도면13a



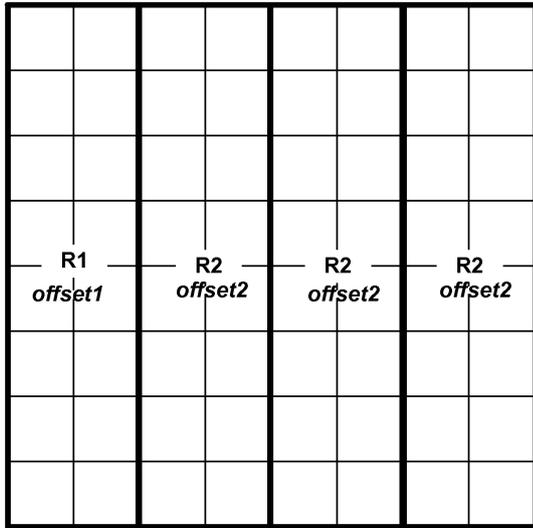
도면13b



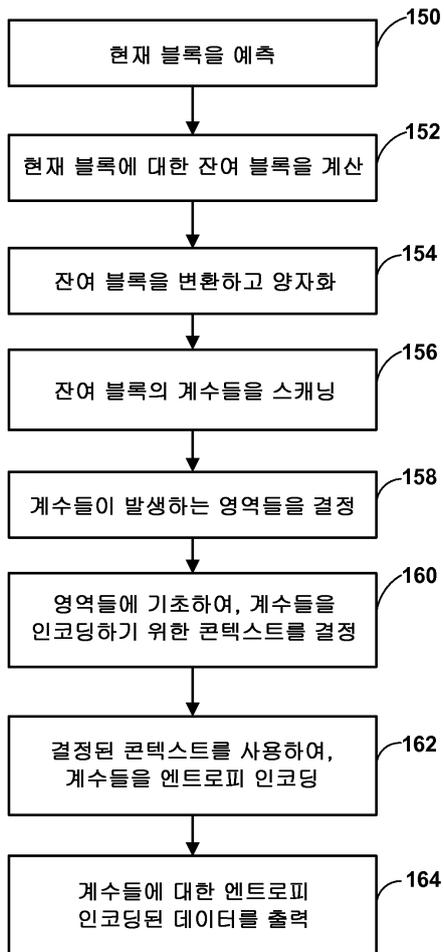
도면14a



도면14b



도면15



도면16

