



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2015년06월10일

(11) 등록번호 10-1528085

(24) 등록일자 2015년06월04일

(51) 국제특허분류(Int. Cl.)

H04N 19/00 (2014.01)

(21) 출원번호 10-2013-7011349

(22) 출원일자(국제) 2011년09월30일

심사청구일자 2013년05월01일

(85) 번역문제출일자 2013년05월01일

(65) 공개번호 10-2013-0095289

(43) 공개일자 2013년08월27일

(86) 국제출원번호 PCT/US2011/054425

(87) 국제공개번호 WO 2012/045041

국제공개일자 2012년04월05일

(30) 우선권주장

13/249,020 2011년09월29일 미국(US)

(뒷면에 계속)

(56) 선행기술조사문헌

WO2011142817 A1*

KR1020050052523 A

JP2004064725 A

*는 심사관에 의하여 인용된 문헌

(73) 특허권자

퀄컴 인코포레이티드

미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775

(72) 발명자

솔레 로할스 호엘

미국 92121 캘리포니아주 샌디에고 모어하우스 드라이브 5775

조쉬 라잔 엘

미국 92121 캘리포니아주 샌디에고 모어하우스 드라이브 5775

카르체비츠 마르타

미국 92121 캘리포니아주 샌디에고 모어하우스 드라이브 5775

(74) 대리인

특허법인코리아나

전체 청구항 수 : 총 26 항

심사관 : 이상래

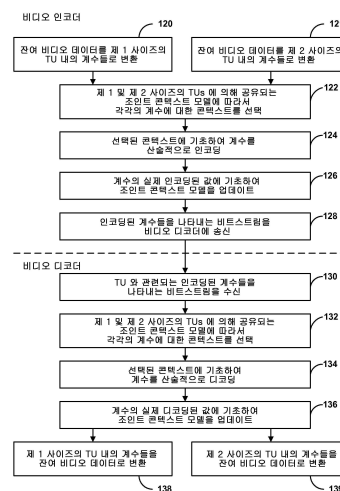
(54) 발명의 명칭 조인트 콘텍스트 모델을 사용한 계수들의 엔트로피 코딩

(57) 요약

이 개시물은 상이한 사이즈들을 가지는 변환 유닛들 사이에 공유되는 조인트 콘텍스트 모델을 사용하여 비디오 계수들의 엔트로피 인코딩 및 디코딩을 수행하는 기법들을 설명한다. 예를 들어, 조인트 콘텍스트 모델은 32x32 의 제 1 사이즈를 가지는 변환 유닛들과 16x16 의 제 2 사이즈를 가지는 변환 유닛들 사이에 공유될 수도

(뒷면에 계속)

대 표 도 - 도7



있다. 상이한 사이즈들을 가지는 변환 유닛들 사이에 공유되는 조인트 콘텍스트 모델을 사용하여 엔트로피 코딩을 수행하는 것은, 콘텍스트들 및 확률들을 저장하는 데 필요한 메모리의 양을 감소시킬 수도 있고, 콘텍스트 모델들을 유지하는 계산 비용들을 저감한다. 일 예에서, 조인트 콘텍스트 모델은 제 2 사이즈를 가지는 리테인된 계수 블록을 생성하도록 제로화된 계수들과 제 1 사이즈를 가지는 변환 유닛들과 제 2 사이즈를 가지는 변환 유닛들 사이에 공유될 수도 있다. 다른 예에서, 조인트 콘텍스트 모델은 제 1 사이즈를 가지는 변환 유닛들과 제 2 사이즈를 가지는 변환 유닛들 사이에 공유될 수도 있다.

(30) 우선권주장

13/249,079 2011년09월29일 미국(US)

61/389,170 2010년10월01일 미국(US)

명세서

청구범위

청구항 1

비디오 데이터의 변환 계수들을 엔트로피 코딩하기 위한 복수의 컨텍스트 모델들을 유지하는 단계로서, 상기 복수의 컨텍스트 모델들은 각각이 상이한 변환 유닛 사이즈에 사용되는 하나 이상의 컨텍스트 모델들 및 2 이상의 변환 유닛 사이즈들에 사용되는 적어도 하나의 조인트 컨텍스트 모델을 포함하는, 상기 복수의 컨텍스트 모델들을 유지하는 단계;

제 1 변환 유닛 또는 제 2 변환 유닛 중 하나의 변환 유닛에서의 변환 계수들을 엔트로피 코딩하기 위한, 제 1 사이즈를 가지는 상기 제 1 변환 유닛 및 제 2 사이즈를 가지는 상기 제 2 변환 유닛에 의해 공유되는 상기 조인트 컨텍스트 모델을 선택하는 단계로서, 상기 제 1 사이즈 및 상기 제 2 사이즈는 상이하고, 상기 변환 유닛들 각각은 변환 계수들의 복수의 블록들을 포함하는 잔여 (residual) 비디오 블록 데이터 유닛을 포함하는, 상기 조인트 컨텍스트 모델을 선택하는 단계;

상기 조인트 컨텍스트 모델에 따라 상기 제 1 변환 유닛 또는 상기 제 2 변환 유닛 중 상기 하나의 변환 유닛과 연관된 상기 변환 계수들에 대한 컨텍스트들을 선택하는 단계로서, 상기 변환 계수들에 대한 컨텍스트들을 선택하는 단계는, 상기 변환 유닛들 중 상기 하나의 변환 유닛의 이전에 코딩된 인근 변환 계수들의 값들에 기초하여 상기 변환 계수들 중 주어진 하나의 변환 계수에 상기 조인트 컨텍스트 모델에서의 컨텍스트를 할당하는 단계 및 상기 조인트 컨텍스트 모델에서의 상기 할당된 컨텍스트와 연관된 상기 변환 계수들 중 상기 주어진 하나의 변환 계수의 값에 대한 확률 추정을 결정하는 단계를 포함하는, 상기 변환 계수들에 대한 컨텍스트들을 선택하는 단계; 및

상기 선택된 컨텍스트들에 기초하여, 컨텍스트 적응적 이진 산술 코딩 (context adaptive binary arithmetic coding: CABAC) 을 이용하여 상기 변환 유닛들 중 상기 하나의 변환 유닛의 상기 변환 계수들을 엔트로피 코딩하는 단계를 포함하는, 비디오 데이터의 코딩 방법.

청구항 2

제 1 항에 있어서,

상기 조인트 컨텍스트 모델은 유효 맵 (significance map) 의 조인트 컨텍스트 모델을 포함하고,

상기 비디오 데이터의 코딩 방법은,

상기 조인트 컨텍스트 모델에 따라 상기 제 1 사이즈를 가지는 상기 제 1 변환 유닛 또는 상기 제 2 사이즈를 가지는 상기 제 2 변환 유닛 중 상기 하나의 변환 유닛과 연관된 상기 변환 계수들에 대한 상기 유효 맵의 컴포넌트들에 대한 컨텍스트들을 선택하는 단계; 및

상기 선택된 컨텍스트들에 기초하여 상기 변환 유닛들 중 상기 하나의 변환 유닛과 연관된 상기 유효 맵의 상기 컴포넌트들을 엔트로피 코딩하는 단계를 더 포함하는, 비디오 데이터의 코딩 방법.

청구항 3

삭제

청구항 4

제 1 항에 있어서,

상기 비디오 데이터의 코딩 방법은 비디오 데이터를 디코딩하는 방법을 포함하며,

상기 비디오 데이터의 코딩 방법은,

상기 제 1 사이즈를 가지는 상기 제 1 변환 유닛 또는 상기 제 2 사이즈를 가지는 상기 제 2 변환 유닛 중 상기 하나의 변환 유닛과 연관된 인코딩된 변환 계수들을 나타내는 비트스트림을 수신하는 단계;

상기 조인트 컨텍스트 모델에 따라 상기 인코딩된 변환 계수들에 대한 컨텍스트들을 선택하는 단계; 및
상기 선택된 컨텍스트들에 기초하여 상기 인코딩된 변환 계수들을 상기 변환 유닛들 중 상기 하나의 변환 유닛으로 엔트로피 디코딩하는 단계를 더 포함하는, 비디오 데이터의 코딩 방법.

청구항 5

제 4 항에 있어서,

상기 제 1 사이즈를 가지는 상기 제 1 변환 유닛 또는 상기 제 2 사이즈를 가지는 상기 제 2 변환 유닛 중 상기 하나의 변환 유닛의 상기 변환 계수들을 잔여 비디오 데이터로 역변환하는 단계를 더 포함하는, 비디오 데이터의 코딩 방법.

청구항 6

제 1 항에 있어서,

상기 비디오 데이터의 코딩 방법은 비디오 데이터를 인코딩하는 방법을 포함하며,

상기 비디오 데이터의 코딩 방법은,

잔여 비디오 데이터를, 상기 제 1 사이즈를 가지는 상기 제 1 변환 유닛 또는 상기 제 2 사이즈를 가지는 상기 제 2 변환 유닛 중 상기 하나의 변환 유닛 내의 상기 변환 계수들로 변환하는 단계;

상기 조인트 컨텍스트 모델에 따라 상기 변환 유닛들 중 상기 하나의 변환 유닛의 상기 변환 계수들에 대한 컨텍스트들을 선택하는 단계; 및

상기 선택된 컨텍스트들에 기초하여 상기 변환 유닛들 중 상기 하나의 변환 유닛의 상기 변환 계수들을 엔트로피 인코딩하는 단계를 더 포함하는, 비디오 데이터의 코딩 방법.

청구항 7

제 1 항에 있어서,

상기 제 1 사이즈의 상기 제 1 변환 유닛은 32x32 변환 유닛을 포함하고,

상기 제 2 사이즈의 상기 제 2 변환 유닛은 16x16 변환 유닛을 포함하는, 비디오 데이터의 코딩 방법.

청구항 8

제 1 항에 있어서,

상기 조인트 컨텍스트 모델에 따라 컨텍스트들을 선택하는 단계는,

상기 제 1 변환 유닛의 이전에 코딩된 인근 변환 계수들의 값들에 기초하여 상기 제 1 사이즈를 가지는 상기 제 1 변환 유닛의 제 1 변환 계수에 상기 조인트 컨텍스트 모델에서의 컨텍스트를 할당하는 단계;

상기 조인트 컨텍스트 모델에서의 상기 할당된 컨텍스트와 연관된 상기 제 1 변환 계수의 값에 대한 확률 추정을 결정하는 단계;

상기 제 1 변환 계수의 실제 코딩된 값에 기초하여 상기 조인트 컨텍스트 모델에서의 상기 할당된 컨텍스트와 연관된 상기 확률 추정을 업데이트하는 단계;

상기 제 2 변환 유닛의 이전에 코딩된 인근 변환 계수들의 값들에 기초하여 상기 제 2 사이즈를 가지는 상기 제 2 변환 유닛의 제 2 변환 계수에 상기 조인트 컨텍스트 모델에서의 컨텍스트와 동일한 컨텍스트를 할당하는 단계;

상기 조인트 컨텍스트 모델에서의 상기 할당된 동일한 컨텍스트와 연관된 상기 제 2 변환 계수의 값에 대한 확률 추정을 결정하는 단계; 및

상기 제 2 변환 계수의 실제 코딩된 값에 기초하여 상기 조인트 컨텍스트 모델에서의 상기 할당된 동일한 컨텍스트와 연관된 상기 확률 추정을 업데이트하는 단계를 포함하는, 비디오 데이터의 코딩 방법.

청구항 9

비디오 데이터의 변환 계수들을 엔트로피 코딩하기 위한 복수의 컨텍스트 모델들을 저장하되, 상기 복수의 컨텍스트 모델들은 각각이 상이한 변환 유닛 사이즈에 사용되는 하나 이상의 컨텍스트 모델들 및 2 이상의 변환 유닛 사이즈들에 사용되는 적어도 하나의 조인트 컨텍스트 모델을 포함하는, 메모리; 및

프로세서를 포함하고, 상기 프로세서는,

제 1 변환 유닛 또는 제 2 변환 유닛 중 하나의 변환 유닛에서의 변환 계수들을 엔트로피 코딩하기 위한, 제 1 사이즈를 가지는 상기 제 1 변환 유닛 및 제 2 사이즈를 가지는 제 2 변환 유닛에 의해 공유되는 상기 조인트 컨텍스트 모델을 선택하되, 상기 제 1 사이즈 및 상기 제 2 사이즈는 상이하고, 상기 변환 유닛들 각각은 변환 계수들의 복수의 블록들을 포함하는 잔여 (residual) 비디오 블록 데이터 유닛을 포함하고;

상기 조인트 컨텍스트 모델에 따라 상기 제 1 변환 유닛 또는 상기 제 2 변환 유닛 중 하나의 변환 유닛과 연관된 상기 변환 계수들에 대한 컨텍스트들을 선택하도록 구성되되, 상기 프로세서는 상기 변환 유닛들 중 상기 하나의 변환 유닛의 이전에 코딩된 인근 변환 계수들의 값들에 기초하여 상기 변환 계수들 중 주어진 하나의 변환 계수에 상기 조인트 컨텍스트 모델에서의 컨텍스트를 할당하고, 상기 조인트 컨텍스트 모델에서의 상기 할당된 컨텍스트와 연관된 상기 변환 계수들 중 상기 주어진 하나의 변환 계수의 값에 대한 확률 추정을 결정하도록 구성되고;

상기 선택된 컨텍스트들에 기초하여, 컨텍스트 적응적 이진 산술 코딩 (context adaptive binary arithmetic coding: CABAC) 을 이용하여 상기 변환 유닛들 중 상기 하나의 변환 유닛의 상기 변환 계수들을 엔트로피 코딩하도록 구성된 프로세서를 포함하는, 비디오 코딩 디바이스.

청구항 10

제 9 항에 있어서,

상기 조인트 컨텍스트 모델은 유효 맵의 조인트 컨텍스트 모델을 포함하고,

상기 프로세서는,

상기 조인트 컨텍스트 모델에 따라 상기 제 1 사이즈를 가지는 상기 제 1 변환 유닛 또는 상기 제 2 사이즈를 가지는 상기 제 2 변환 유닛 중 상기 하나의 변환 유닛과 연관된 상기 변환 계수들에 대한 상기 유효 맵의 컴포넌트들에 대한 컨텍스트들을 선택하고;

상기 선택된 컨텍스트들에 기초하여 상기 변환 유닛들 중 상기 하나의 변환 유닛과 연관된 상기 유효 맵의 상기 컴포넌트들을 엔트로피 코딩하도록 구성된, 비디오 코딩 디바이스.

청구항 11

삭제

청구항 12

제 9 항에 있어서,

상기 비디오 코딩 디바이스는 비디오 디코딩 디바이스를 포함하며,

상기 프로세서는,

상기 제 1 사이즈를 가지는 상기 제 1 변환 유닛 또는 상기 제 2 사이즈를 가지는 상기 제 2 변환 유닛 중 상기 하나의 변환 유닛과 연관된 인코딩된 변환 계수들을 나타내는 비트스트림을 수신하고;

상기 조인트 컨텍스트 모델에 따라 상기 인코딩된 변환 계수들에 대한 컨텍스트들을 선택하고;

상기 선택된 컨텍스트들에 기초하여 상기 인코딩된 변환 계수들을 상기 변환 유닛들 중 상기 하나의 변환 유닛으로 엔트로피 디코딩하도록 구성된, 비디오 코딩 디바이스.

청구항 13

제 12 항에 있어서,

상기 프로세서는,

상기 제 1 사이즈를 가지는 상기 제 1 변환 유닛 또는 상기 제 2 사이즈를 가지는 상기 제 2 변환 유닛 중 상기 하나의 변환 유닛의 상기 변환 계수들을 잔여 비디오 데이터로 역변환하도록 구성된, 비디오 코딩 디바이스.

청구항 14

제 9 항에 있어서,

상기 비디오 코딩 디바이스는 비디오 인코딩 디바이스를 포함하며,

상기 프로세서는,

잔여 비디오 데이터를, 상기 제 1 사이즈를 가지는 상기 제 1 변환 유닛 또는 상기 제 2 사이즈를 가지는 상기 제 2 변환 유닛 중 상기 하나의 변환 유닛 내의 상기 변환 계수들로 변환하고;

상기 조인트 컨텍스트 모델에 따라 상기 변환 유닛들 중 상기 하나의 변환 유닛의 상기 변환 계수들에 대한 콘텍스트들을 선택하고;

상기 선택된 콘텍스트들에 기초하여 상기 변환 유닛들 중 상기 하나의 변환 유닛의 상기 변환 계수들을 엔트로피 인코딩하도록 구성된, 비디오 코딩 디바이스.

청구항 15

제 9 항에 있어서,

상기 제 1 사이즈의 상기 제 1 변환 유닛은 32x32 변환 유닛을 포함하고,

상기 제 2 사이즈의 상기 제 2 변환 유닛은 16x16 변환 유닛을 포함하는, 비디오 코딩 디바이스.

청구항 16

제 9 항에 있어서,

상기 프로세서는,

상기 제 1 변환 유닛의 이전에 코딩된 인근 변환 계수들의 값들에 기초하여 상기 제 1 사이즈를 가지는 상기 제 1 변환 유닛의 제 1 변환 계수에 상기 조인트 컨텍스트 모델에서의 콘텍스트를 할당하고;

상기 조인트 컨텍스트 모델에서의 상기 할당된 콘텍스트와 연관된 상기 제 1 변환 계수의 값에 대한 확률 추정을 결정하고;

상기 제 1 변환 계수의 실제 코딩된 값에 기초하여 상기 조인트 컨텍스트 모델에서의 상기 할당된 콘텍스트와 연관된 상기 확률 추정을 업데이트하고;

상기 제 2 변환 유닛의 이전에 코딩된 인근 변환 계수들의 값들에 기초하여 상기 제 2 사이즈를 가지는 상기 제 2 변환 유닛의 상기 제 2 변환 계수에 상기 조인트 컨텍스트 모델에서의 콘텍스트와 동일한 콘텍스트를 할당하고;

상기 조인트 컨텍스트 모델에서의 상기 할당된 동일한 콘텍스트와 연관된 상기 제 2 변환 계수의 값에 대한 확률 추정을 결정하고;

상기 제 2 변환 계수의 실제 코딩된 값에 기초하여 상기 조인트 컨텍스트 모델에서의 상기 할당된 동일한 콘텍스트와 연관된 상기 확률 추정을 업데이트하도록 구성된, 비디오 코딩 디바이스.

청구항 17

비디오 데이터의 변환 계수들을 엔트로피 코딩하기 위한 복수의 컨텍스트 모델들을 유지하는 수단으로서, 상기 복수의 컨텍스트 모델들은 각각이 상이한 변환 유닛 사이즈에 사용되는 하나 이상의 컨텍스트 모델들 및 2 이상의 변환 유닛 사이즈들에 사용되는 적어도 하나의 조인트 컨텍스트 모델을 포함하는, 상기 복수의 컨텍스트 모델들을 유지하는 수단;

제 1 변환 유닛 또는 제 2 변환 유닛 중 하나의 변환 유닛에서의 변환 계수들을 엔트로피 코딩하기 위한, 제 1

사이즈를 가지는 상기 제 1 변환 유닛 및 제 2 사이즈를 가지는 제 2 변환 유닛에 의해 공유되는 상기 조인트 콘텍스트 모델을 선택하는 수단으로서, 상기 제 1 사이즈 및 상기 제 2 사이즈는 상이하고, 상기 변환 유닛들 각각은 변환 계수들의 복수의 블록들을 포함하는 잔여 (residual) 비디오 블록 데이터 유닛을 포함하는, 상기 조인트 콘텍스트 모델을 선택하는 수단;

상기 조인트 콘텍스트 모델에 따라 상기 제 1 변환 유닛 또는 상기 제 2 변환 유닛 중 상기 하나의 변환 유닛과 연관된 상기 변환 계수들에 콘텍스트들을 선택하는 수단; 및

상기 선택된 콘텍스트들에 기초하여, 콘텍스트 적응적 이진 산술 코딩 (context adaptive binary arithmetic coding: CABAC) 을 이용하여 상기 변환 유닛들 중 상기 하나의 변환 유닛의 상기 변환 계수들을 엔트로피 코딩하는 수단을 포함하고,

상기 변환 계수들에 대한 콘텍스트들을 선택하는 수단은, 상기 변환 유닛들 중 상기 하나의 변환 유닛의 이전에 코딩된 인근 변환 계수들의 값들에 기초하여 상기 변환 계수들 중 주어진 하나의 변환 계수에 대한 상기 조인트 콘텍스트 모델에서의 콘텍스트를 할당하는 수단 및 상기 조인트 콘텍스트 모델에서의 상기 할당된 콘텍스트와 연관된 상기 변환 계수들 중 상기 주어진 하나의 변환 계수의 값에 대한 확률 추정을 결정하는 수단을 포함하는, 비디오 코딩 디바이스.

청구항 18

제 17 항에 있어서,

상기 조인트 콘텍스트 모델은 유효 맵 (significance map) 의 조인트 콘텍스트 모델을 포함하고,

상기 비디오 코딩 디바이스는,

상기 조인트 콘텍스트 모델에 따라 상기 제 1 사이즈를 가지는 상기 제 1 변환 유닛 또는 상기 제 2 사이즈를 가지는 상기 제 2 변환 유닛 중 상기 하나의 변환 유닛과 연관된 상기 변환 계수들에 대한 상기 유효 맵의 컴포넌트들에 대한 콘텍스트들을 선택하는 수단; 및

상기 선택된 콘텍스트들에 기초하여 상기 변환 유닛들 중 상기 하나의 변환 유닛과 연관된 상기 유효 맵의 상기 컴포넌트들을 엔트로피 코딩하는 수단을 더 포함하는, 비디오 코딩 디바이스.

청구항 19

삭제

청구항 20

제 17 항에 있어서,

상기 비디오 코딩 디바이스는 비디오 디코딩 디바이스를 포함하며,

상기 비디오 코딩 디바이스는,

상기 제 1 사이즈를 가지는 상기 제 1 변환 유닛 또는 상기 제 2 사이즈를 가지는 상기 제 2 변환 유닛 중 상기 하나의 변환 유닛과 연관된 인코딩된 변환 계수들을 나타내는 비트스트림을 수신하는 수단;

상기 조인트 콘텍스트 모델에 따라 상기 인코딩된 변환 계수들에 대한 콘텍스트들을 선택하는 수단; 및

상기 선택된 콘텍스트들에 기초하여 상기 인코딩된 변환 계수들을 상기 변환 유닛들 중 상기 하나의 변환 유닛으로 엔트로피 디코딩하는 수단을 더 포함하는, 비디오 코딩 디바이스.

청구항 21

제 20 항에 있어서,

상기 제 1 사이즈를 가지는 상기 제 1 변환 유닛 또는 상기 제 2 사이즈를 가지는 상기 제 2 변환 유닛 중 상기 하나의 변환 유닛의 상기 변환 계수들을 잔여 비디오 데이터로 역변환하는 수단을 더 포함하는, 비디오 코딩 디바이스.

청구항 22

제 17 항에 있어서,

상기 비디오 코딩 디바이스는 비디오 인코딩 디바이스를 포함하며,

상기 비디오 코딩 디바이스는,

잔여 비디오 데이터, 상기 제 1 사이즈를 가지는 상기 제 1 변환 유닛 또는 상기 제 2 사이즈를 가지는 상기 제 2 변환 유닛 중 상기 하나의 변환 유닛 내의 상기 변환 계수들로 변환하는 수단;

상기 조인트 컨텍스트 모델에 따라 상기 변환 유닛들 중 상기 하나의 변환 유닛의 상기 변환 계수들에 대한 컨텍스트들을 선택하는 수단; 및

상기 선택된 컨텍스트들에 기초하여 상기 변환 유닛들 중 상기 하나의 변환 유닛의 상기 변환 계수들을 엔트로피 인코딩하는 수단을 더 포함하는, 비디오 코딩 디바이스.

청구항 23

제 17 항에 있어서,

상기 제 1 사이즈의 상기 제 1 변환 유닛은 32x32 변환 유닛을 포함하고,

상기 제 2 사이즈의 상기 제 2 변환 유닛은 16x16 변환 유닛을 포함하는, 비디오 코딩 디바이스.

청구항 24

비디오 데이터를 코딩하기 위한 명령들을 포함하는 컴퓨터 판독가능 매체로서,

상기 명령들은, 실행될 때, 프로세서로 하여금,

비디오 데이터의 변환 계수들을 엔트로피 코딩하기 위한 복수의 컨텍스트 모델들을 유지하되, 상기 복수의 컨텍스트 모델들은 각각이 상이한 변환 유닛 사이즈에 사용되는 하나 이상의 컨텍스트 모델들 및 2 이상의 변환 유닛 사이즈들에 사용되는 적어도 하나의 조인트 컨텍스트 모델을 포함하게 하고;

제 1 변환 유닛 또는 제 2 변환 유닛 중 하나의 변환 유닛에서의 변환 계수들을 엔트로피 코딩하기 위한, 제 1 사이즈를 가지는 상기 제 1 변환 유닛 및 제 2 사이즈를 가지는 제 2 변환 유닛에 의해 공유되는 상기 조인트 컨텍스트 모델을 선택하되, 상기 제 1 사이즈 및 상기 제 2 사이즈는 상이하고, 상기 변환 유닛들 각각은 변환 계수들의 복수의 블록들을 포함하는 잔여 (residual) 비디오 블록 데이터 유닛을 포함하게 하고;

상기 조인트 컨텍스트 모델에 따라 상기 제 1 변환 유닛 또는 상기 제 2 변환 유닛 중 상기 하나의 변환 유닛과 연관된 상기 변환 계수들에 대한 컨텍스트들을 선택하게 하고;

상기 선택된 컨텍스트들에 기초하여, 컨텍스트 적응적 이진 산술 코딩 (context adaptive binary arithmetic coding: CABAC) 을 이용하여 상기 변환 유닛들 중 상기 하나의 변환 유닛의 상기 변환 계수들을 엔트로피 코딩하게 하며;

상기 명령들은, 상기 프로세서로 하여금 추가로, 상기 변환 유닛들 중 상기 하나의 변환 유닛의 이전에 코딩된 인근 변환 계수들의 값들에 기초하여 상기 변환 계수들 중 주어진 하나의 변환 계수에 상기 조인트 컨텍스트 모델에서의 컨텍스트를 할당하게 하고, 상기 조인트 컨텍스트 모델에서의 상기 할당된 컨텍스트와 연관된 상기 변환 계수들 중 상기 주어진 하나의 변환 계수의 값에 대한 확률 추정을 결정하게 하는, 컴퓨터 판독가능 매체.

청구항 25

제 24 항에 있어서,

상기 조인트 컨텍스트 모델은 유효 맵 (significance map) 의 조인트 컨텍스트 모델을 포함하고,

상기 컴퓨터 판독가능 매체는, 상기 프로세서로 하여금,

상기 조인트 컨텍스트 모델에 따라 상기 제 1 사이즈를 가지는 상기 제 1 변환 유닛 또는 상기 제 2 사이즈를 가지는 상기 제 2 변환 유닛 중 상기 하나의 변환 유닛과 연관된 상기 변환 계수들에 대한 상기 유효 맵의 컴포넌트들에 대한 컨텍스트들을 선택하게 하는 명령들; 및

상기 선택된 컨텍스트들에 기초하여 상기 유닛들 중 상기 하나의 변환 유닛과 연관된 상기 유효 맵의 상기 컴포

넌트들을 엔트로피 코딩하게 하는 명령들을 더 포함하는, 컴퓨터 판독가능 매체.

청구항 26

삭제

청구항 27

제 24 항에 있어서,

상기 명령들은 비디오 데이터를 디코딩하기 위한 명령들을 포함하고,

상기 컴퓨터 판독가능 매체는, 상기 프로세서로 하여금,

상기 제 1 사이즈를 가지는 상기 제 1 변환 유닛 또는 상기 제 2 사이즈를 가지는 상기 제 2 변환 유닛 중 상기 하나의 변환 유닛과 연관된 인코딩된 변환 계수들을 나타내는 비트스트림을 수신하게 하고;

상기 조인트 컨텍스트 모델에 따라 상기 인코딩된 변환 계수들에 대한 컨텍스트들을 선택하게 하고;

상기 선택된 컨텍스트들에 기초하여 상기 인코딩된 변환 계수들을 상기 제 1 또는 제 2 변환 유닛 중 상기 하나의 변환 유닛으로 엔트로피 디코딩하게 하는 명령들을 더 포함하는, 컴퓨터 판독가능 매체.

청구항 28

제 27 항에 있어서,

상기 프로세서로 하여금, 상기 제 1 사이즈를 가지는 상기 제 1 변환 유닛 또는 상기 제 2 사이즈를 가지는 상기 제 2 변환 유닛 중 상기 하나의 변환 유닛의 상기 변환 계수들을 잔여 비디오 데이터로 역변환하게 하는 명령들을 더 포함하는, 컴퓨터 판독가능 매체.

청구항 29

제 24 항에 있어서,

상기 명령들은 비디오 데이터를 인코딩하기 위한 명령들을 포함하며,

상기 컴퓨터 판독가능 매체는, 상기 프로세서로 하여금,

잔여 비디오 데이터를, 상기 제 1 사이즈를 가지는 상기 제 1 변환 유닛 또는 상기 제 2 사이즈를 가지는 상기 제 2 변환 유닛 중 상기 하나의 변환 유닛 내의 상기 변환 계수들로 변환하게 하고;

상기 조인트 컨텍스트 모델에 따라 상기 변환 유닛들 중 상기 하나의 변환 유닛의 상기 변환 계수들에 대한 컨텍스트들을 선택하게 하고;

상기 선택된 컨텍스트들에 기초하여 상기 변환 유닛들 중 상기 하나의 변환 유닛의 상기 변환 계수들을 엔트로피 인코딩하게 하는 명령들을 더 포함하는, 컴퓨터 판독가능 매체.

청구항 30

제 24 항에 있어서,

상기 제 1 사이즈의 상기 제 1 변환 유닛은 32x32 변환 유닛을 포함하고,

상기 제 2 사이즈의 상기 제 2 변환 유닛은 16x16 변환 유닛을 포함하는, 컴퓨터 판독가능 매체.

청구항 31

삭제

청구항 32

삭제

청구항 33

삭제

청구항 34

삭제

청구항 35

삭제

청구항 36

삭제

청구항 37

삭제

청구항 38

삭제

청구항 39

삭제

청구항 40

삭제

청구항 41

삭제

청구항 42

삭제

청구항 43

삭제

청구항 44

삭제

발명의 설명

기술 분야

[0001] 본 개시물은 비디오 코딩에 관한 것이고, 더 구체적으로는, 비디오 코딩을 위한 엔트로피 코딩에 관한 것이다.

배경 기술

[0002] 디지털 비디오 성능들은, 디지털 텔레비전, 디지털 직접 브로드캐스트 시스템, 무선 브로드캐스트 시스템, 개인 휴대 정보 단말기 (PDA), 랩톱 또는 데스크톱 컴퓨터, 디지털 카메라, 디지털 리코딩 디바이스, 디지털 미디어 플레이어, 비디오 게이밍 디바이스, 비디오 게임 콘솔, 셀룰러 또는 위성 무선 전화, 비디오 원격 회의 디바이스 등을 포함하는 광범위한 디바이스들 내에 포함될 수 있다. 디지털 비디오 디바이스들은, MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Part 10, AVC (Advanced Video Coding), 또는 부상하는 HEVC (High Efficiency Video Coding) 표준에 의해 정의되는 표준들 및 이러한 표준들의 확장들에서 설명되는 것들과 같은 비디오 압축 기법들을 구현한다.

[0003] 비디오 압축 기법들은 비디오 시퀀스들에 내재하는 리턴던시를 감소시키거나 제거하는 공간 예측 및/또는 시간

예측을 수행한다. 블록 기반 비디오 코딩의 경우, 비디오 프레임 또는 슬라이스는 비디오 블록들 또는 코딩 유닛들(CUs)로 파티셔닝될 수도 있다. CUs은 하나 이상의 예측 유닛들(PUs)로 더 파티셔닝되어 CU에 대한 예측 비디오 데이터를 결정하도록 할 수도 있다. 비디오 압축 기법들은 또한 CUs를 잔여 비디오 블록 데이터의 하나 이상의 변환 유닛들(TUs)로 파티셔닝할 수도 있으며, 이러한 TUs은 코딩될 비디오 블록과 예측 비디오 데이터 사이의 차이를 나타낸다. 2차원 이산 코사인 변환(DCT)과 같은 선형 변환들이 잔여 비디오 블록 데이터를 픽셀 도메인으로부터 주파수 도메인으로 변환하여 추가 압축을 달성하도록 TU에 적용될 수도 있다.

[0004]

변환에 이어서, TU 내의 변환 계수들은 양자화를 통해 더 압축될 수도 있다. 양자화 다음으로, 엔트로피 코딩 유닛은, TU의 계수들의 2차원 어레이를 스캔하여 엔트로피 인코딩될 수 있는 연속 벡터를 생성하도록 TU의 사이즈와 관련된 지그-재그 스캔 또는 다른 스캔 순서를 적용할 수도 있다. 그 후, 엔트로피 코딩 유닛은, 계수들의 연속 벡터를 엔트로피 코딩한다. 예를 들어, 엔트로피 코딩 유닛은 컨텍스트 적응적 가변 길이 코딩(context adaptive variable length coding: CAVLC), 컨텍스트 적응적 이진 산술 코딩(context adaptive binary arithmetic coding: CABAC), 또는 다른 엔트로피 코딩 기법을 수행할 수도 있다. 컨텍스트 적응적 엔트로피 코딩의 경우에 있어서, 엔트로피 코딩 유닛은, TU의 사이즈와 관련된 컨텍스트 모델에 따라서, TU 내의 계수들 각각에 대한 컨텍스트들을 선택할 수도 있다. 그 후, 계수들은 선택된 컨텍스트들에 기초하여 엔트로피 코딩될 수도 있다.

발명의 내용

과제의 해결 수단

[0005]

일반적으로, 이 개시물은 상이한 사이즈들을 가지는 변환 유닛들 사이에 공유되는 조인트 컨텍스트 모델을 사용하여 잔여 비디오 데이터의 블록과 관련되는 변환 계수들의 엔트로피 인코딩 및 디코딩을 수행하는 기법들을 설명한다. 예를 들어, 조인트 컨텍스트 모델은 32x32의 제 1 사이즈를 가지는 변환 유닛들과 16x16의 제 2 사이즈를 가지는 변환 유닛들 사이에 공유될 수도 있다. 몇몇 경우들에 있어서, 2개 초과 사이즈들의 변환 유닛들이, 동일한 조인트 컨텍스트 모델을 공유할 수도 있다. 일 예로서, 조인트 컨텍스트 모델은, 변환 유닛에 대한 유효 맵의 조인트 컨텍스트 모델일 수도 있다. 다른 예들에서, 조인트 컨텍스트 모델은 다른 코딩 정보 또는 신택스 엘리먼트들과 관련될 수도 있다.

[0006]

부상하는 고효율 비디오 코딩(High Efficiency Video Coding: HEVC) 표준에서, 코딩 유닛(CU)은 변환을 위한 잔여 비디오 데이터를 포함하는 하나 이상의 변환 유닛들(TUs)을 포함할 수도 있다. 추가 변환 유닛 사이즈들, 예컨대 32x32 내지 128x128은 비디오 코딩 효율을 향상시키도록 제안되었지만, 각각의 추가 변환 유닛 사이즈들에 대한 컨텍스트 모델들을 유지하기 위해 증가하는 메모리 및 계산 요건들을 역시 초래한다. 상이한 사이즈들을 가지는 변환 유닛들 사이에 공유되는 조인트 컨텍스트 모델을 사용하여 엔트로피 코딩을 수행하면, 비디오 인코딩 및 디코딩 디바이스들 상에 컨텍스트들 및 확률들을 저장하는 데 필요한 메모리의 양을 감소시킬 수도 있고, 비디오 인코딩 및 디코딩 디바이스들 상에 컨텍스트 모델들을 유지하는 계산 비용들을 저감할 수도 있다.

[0007]

몇몇 예들에서, 이 기법들은, 2차원 변환들을 수행할 때 더 큰 변환 유닛 사이즈들에 대한 중간 버퍼링 요건들을 역시 감소시킬 수도 있다. 그 경우에 있어서, 이 기법들은, 각각의 방향의 2차원 변환을 적용하여 리테인된 계수 블록을 생성한 후에 제 1 사이즈의 변환 유닛에 포함된 더 높은 주파수의 변환 계수 서브세트의 제로화, 즉 값들을 제로로 설정하는 것을 포함한다. 이 예에서, 엔트로피 코딩을 위한 조인트 컨텍스트 모델은 리테인된 계수 블록을 생성하도록 제로화된 계수들 및 제 1 사이즈를 가지는 변환 유닛과 본래 제 2 사이즈를 가지는 변환 유닛 사이에 공유될 수도 있다. 몇몇 경우들에 있어서, 리테인된 계수 블록은 제 2 사이즈와 동일한 사이즈를 가질 수도 있다. 다른 예들에서, 엔트로피 코딩을 위한 조인트 컨텍스트 모델은 제 1 사이즈를 가지는 변환 유닛과 제 2 사이즈를 가지는 변환 유닛 사이에 공유될 수도 있다.

[0008]

일 예에서, 본 개시물은, 리테인된 계수 블록을 생성하도록 제로화되는 계수들을 가지는 제 1 사이즈의 변환 유닛과 제 2 사이즈를 가지는 변환 유닛에 의해 공유되는 조인트 컨텍스트 모델을 유지하는 단계를 포함하는 비디오 데이터 디코딩 방법을 설명하며, 여기서 제 1 사이즈와 제 2 사이즈는 상이하다. 또한, 이 방법은 리테인된 계수 블록을 가지는 제 1 사이즈 및 조인트 컨텍스트 모델에 따른 제 2 사이즈 중 하나를 가지는 변환 유닛과 관련된 계수들에 대한 컨텍스트들을 선택하는 단계, 및 선택된 컨텍스트에 기초하여 코딩 프로세스에 따라서 변환 유닛과 관련된 계수들을 엔트로피 코딩하는 단계를 포함한다.

- [0009] 다른 예에서, 본 개시물은, 리테인된 계수 블록을 생성하도록 제로화된 계수들을 가지는 제 1 사이즈의 변환 유닛 및 제 2 사이즈를 가지는 변환 유닛에 의해 공유되는 조인트 콘텍스트 모델을 저장하는 메모리를 포함하는 비디오 코딩 디바이스를 설명하며, 여기서 제 1 사이즈와 제 2 사이즈는 상이하다. 비디오 코딩 디바이스는, 조인트 콘텍스트 모델을 유지하고, 리테인된 계수 블록을 가지는 제 1 사이즈 및 조인트 콘텍스트 모델에 따른 제 2 사이즈 중 하나를 가지는 변환 유닛과 관련된 계수들에 대한 콘텍스트들을 선택하고, 선택된 콘텍스트들에 기초하여 코딩 프로세스에 따라서 변환 유닛과 관련된 계수들을 엔트로피 코딩하도록 구성된 프로세서를 더 포함한다.
- [0010] 다른 예에서, 본 개시물은, 리테인된 계수 블록을 생성하도록 제로화된 계수들을 가지는 제 1 사이즈의 변환 유닛 및 제 2 사이즈를 가지는 변환 유닛에 의해 공유되는 조인트 콘텍스트 모델을 유지하기 위한 수단을 포함하는 비디오 코딩 디바이스를 설명하며, 여기서 제 1 사이즈와 제 2 사이즈는 상이하다. 비디오 코딩 디바이스는 또한 리테인된 계수 블록을 가지는 제 1 사이즈 및 조인트 콘텍스트 모델에 따른 제 2 사이즈 중 하나를 가지는 변환 유닛과 관련된 계수들에 대한 콘텍스트들을 선택하기 위한 수단, 및 선택된 콘텍스트들에 기초하여 코딩 프로세스에 따라서 변환 유닛과 관련되는 계수들을 엔트로피 코딩하기 위한 수단을 포함한다.
- [0011] 다른 예에서, 본 개시물은, 실행될 때, 프로세서로 하여금, 리테인된 계수 블록을 생성하도록 제로화되는 계수들을 가지는 제 1 사이즈의 변환 유닛 및 제 2 사이즈를 가지는 변환 유닛에 의해 공유되는 조인트 콘텍스트 모델을 유지하게 하는 명령들을 포함하는 컴퓨터 판독가능 매체를 설명하며, 여기서 제 1 사이즈와 제 2 사이즈는 상이하다. 이 명령들은 또한 프로세서로 하여금, 리테인된 계수 블록을 가지는 제 1 사이즈 및 조인트 콘텍스트 모델에 따른 제 2 사이즈 중 하나를 가지는 변환 유닛과 관련된 계수들에 대한 콘텍스트를 선택하게 하고, 선택된 콘텍스트들에 기초하여 코딩 프로세스에 따라서 변환 유닛과 관련된 계수들을 엔트로피 코딩하게 한다.
- [0012] 다른 예에서, 본 개시물은, 제 1 사이즈를 가지는 변환 유닛 및 제 2 사이즈를 가지는 변환 유닛에 의해 공유되는 조인트 콘텍스트 모델을 유지하는 단계를 포함하는 비디오 데이터 코딩 방법을 설명하며, 여기서 제 1 사이즈와 제 2 사이즈는 상이하다. 이 방법은 또한 조인트 콘텍스트 모델에 따른 제 2 사이즈 및 제 1 사이즈 중 하나를 가지는 변환 유닛과 관련된 계수들에 대한 콘텍스트들을 선택하는 단계, 및 선택된 콘텍스트들에 기초하여 코딩 프로세스에 따라서 변환 유닛의 계수들을 엔트로피 코딩하는 단계를 포함한다.
- [0013] 추가 예에서, 본 개시물은, 제 1 사이즈를 가지는 변환 유닛 및 제 1 사이즈와는 상이한 제 2 사이즈를 가지는 변환 유닛에 의해 공유되는 조인트 콘텍스트 모델을 저장하는 메모리, 및 조인트 콘텍스트 모델을 유지하고, 조인트 콘텍스트 모델에 따른 제 2 사이즈 및 제 1 사이즈 중 하나를 가지는 변환 유닛과 관련된 계수들에 대한 콘텍스트를 선택하고, 선택된 콘텍스트들에 기초하여 코딩 프로세스에 따라서 변환 유닛의 계수들을 엔트로피 코딩하도록 구성된 프로세서를 포함하는 비디오 코딩 디바이스를 설명한다.
- [0014] 다른 예에서, 본 개시물은, 제 1 사이즈를 가지는 변환 유닛 및 제 2 사이즈를 가지는 변환 유닛에 의해 공유되는 조인트 콘텍스트 모델을 유지하기 위한 수단을 포함하는 비디오 코딩 디바이스를 설명하며, 여기서 제 1 사이즈와 제 2 사이즈는 상이하다. 이 비디오 코딩 디바이스는 또한 조인트 콘텍스트 모델에 따른 제 2 사이즈 및 제 1 사이즈 중 하나를 가지는 변환 유닛과 관련된 계수들에 대한 콘텍스트들을 선택하기 위한 수단, 및 선택된 콘텍스트들에 기초하여 코딩 프로세스에 따라서 변환 유닛의 계수들을 엔트로피 코딩하기 위한 수단을 포함한다.
- [0015] 다른 예에서, 본 개시물은, 비디오 데이터를 코딩하기 위한 명령들을 포함하는 컴퓨터 판독가능 매체로서, 실행될 때, 프로세서로 하여금, 제 1 사이즈를 가지는 변환 유닛 및 제 1 사이즈와는 상이한 제 2 사이즈를 가지는 변환 유닛에 의해 공유되는 조인트 콘텍스트 모델을 유지하게 하고, 조인트 콘텍스트 모델에 따른 제 2 사이즈 및 제 1 사이즈 중 하나를 가지는 변환 유닛과 관련된 계수들에 대한 콘텍스트들을 선택하게 하고, 선택된 콘텍스트들에 기초하여 코딩 프로세스에 따라서 변환 유닛의 계수들을 엔트로피 코딩하게 하는 명령들을 포함하는 컴퓨터 판독가능 매체를 설명한다.

도면의 간단한 설명

- [0016] 도 1 은, 상이한 사이즈들을 가지는 변환 유닛들 사이에 공유되는 조인트 콘텍스트 모델을 사용하여 비디오 계수들의 엔트로피 코딩을 수행하는 기법들을 이용할 수도 있는 예시적 비디오 인코딩 및 디코딩 시스템을 예시한 블록도이다.
- 도 2 는, 조인트 콘텍스트 모델을 사용하여 비디오 계수들을 엔트로피 인코딩하는 기법들을 구현할 수도 있는

예시적 비디오 인코더를 예시한 블록도이다.

도 3a 및 도 3b 는 제 1 사이즈를 가지는 변환 유닛으로부터 제 2 사이즈를 가지는 리테인된 계수 블록들의 정사각형 및 직사각형 영역들을 각각 예시한 개념도들이다.

도 4 는 조인트 컨텍스트 모델을 사용하여 비디오 계수들을 엔트로피 디코딩하는 기법들을 구현할 수도 있는 예시적 비디오 디코더를 예시한 블록도이다.

도 5 는 조인트 컨텍스트 모델에 따라서 비디오 계수들에 대한 컨텍스트들을 선택하도록 구성된 예시적 엔트로피 인코딩 유닛을 예시한 블록도이다.

도 6 은 조인트 컨텍스트 모델에 따라서 비디오 계수들에 대한 컨텍스트들을 선택하도록 구성된 예시적 엔트로피 디코딩 유닛을 예시한 블록도이다.

도 7 은 제 1 사이즈를 가지는 제 1 변환 유닛과 제 2 사이즈를 가지는 제 2 변환 유닛 사이에 공유되는 조인트 컨텍스트 모델을 사용하여 비디오 계수들을 엔트로피 인코딩 및 엔트로피 디코딩하는 예시적 동작을 예시한 플로우차트이다.

도 8 은 제 1 사이즈를 가지는 제 1 변환 유닛과 제로화된 계수들 (coefficients zeroed out) 사이에 공유되는 조인트 컨텍스트 모델을 사용하여 비디오 계수들을 엔트로피 인코딩 및 엔트로피 디코딩하는 예시적 동작을 예시한 플로우차트이다.

발명을 실시하기 위한 구체적인 내용

[0017]

전반적으로, 이 개시물은 상이한 사이즈들을 가지는 변환 유닛들 사이에 공유되는 조인트 컨텍스트 모델을 사용하여 잔여 비디오 데이터의 블록과 관련된 변환 계수들의 엔트로피 인코딩 및 엔트로피 디코딩을 수행하는 기법들을 설명한다. 예를 들어, 조인트 컨텍스트 모델은 32x32 의 제 1 사이즈를 가지는 변환 유닛들과 16x16 의 제 2 사이즈를 가지는 변환 유닛들 사이에 공유될 수도 있다. 부상하는 고효율 비디오 코딩 (High Efficiency Video Coding: HEVC) 표준에서, 코딩 유닛 (CU) 은 잔여 비디오 데이터를 포함하는 하나 이상의 변환 유닛들 (TUs) 을 포함할 수도 있다. 변환 전, 잔여 비디오 데이터는 공간 도메인에 잔여 픽셀 값들을 포함한다. 변환 후, 잔여 비디오 데이터는 변환 도메인에 잔여 변환 계수들을 포함한다. 추가 변환 유닛 사이즈들, 예컨대 32x32 내지 128x128 은 비디오 코딩 효율을 향상시키도록 제안되어 오고 있지만, 역시 메모리 및 계산 요건들을 증가시켜 추가 변환 유닛 사이즈들 각각에 대한 컨텍스트 모델들을 유지시킨다. 상이한 사이즈들을 가지는 변환 유닛들 사이에 공유되는 조인트 컨텍스트 모델을 사용하여 엔트로피 코딩을 수행하는 것은, 비디오 인코딩 및 디코딩 디바이스들 상에서 컨텍스트들 및 확률들을 저장하는 데 필요한 메모리의 양을 감소시킬 수도 있고, 비디오 인코딩 및 디코딩 디바이스들에서 컨텍스트 모델들을 유지시키는 계산 비용을 저감할 수도 있다.

[0018]

일부 예들에서, 기법들은 또한 2 차원 변환을 수행할 때 더 큰 변환 유닛 사이즈들에 대한 중간 버퍼링 요건들을 감소시킬 수도 있다. 기법들은, 2 차원 변환의 각각의 방향이 리테인된 계수 블록을 생성하도록 적용된 후, 제 1 사이즈의 변환 유닛에 포함되는 변환 계수들의 더 높은 주파수 서브세트의 제로화, 즉 값들을 제로로 설정하는 것을 포함한다. 그 후, 비디오 인코딩 및 디코딩 디바이스들은, 2 차원 변환의 적용하는 각각의 방향, 즉 로우들 및 컬럼들 사이에서 감소한 수의 계수들을 버퍼링할 수도 있다. 더 높은 주파수의 계수들이 제 1 사이즈를 가지는 변환 유닛으로부터 제로화될 때, 리테인된 계수 블록에 포함된 계수들은, 본래 제 2 사이즈인 변환 유닛에 포함된 계수들과 유사한 확률 통계를 갖는다. 이 경우, 엔트로피 코딩에 대한 조인트 컨텍스트 모델은 본래 제 2 사이즈를 가지는 리테인된 계수 블록 및 변환 유닛들을 생성하도록 제로화된 계수들과 함께 제 1 사이즈를 가지는 변환 유닛들 사이에서 공유될 수도 있다. 몇몇 경우들에 있어서, 리테인된 계수 블록은 제 2 사이즈와 동일한 사이즈를 가질 수도 있다. 다른 경우들에 있어서, 리테인된 계수 블록은 제 1 사이즈 및 제 2 사이즈 양측 모두와는 상이한 제 3 사이즈와 동일한 사이즈를 가질 수도 있다.

[0019]

다른 예들에서, 제 1 사이즈를 가지는 제 1 변환 유닛에 포함된 계수들은, 제 1 변환 유닛 내에 더 높은 주파수의 계수들을 제로화하지 않고서도, 제 2 사이즈의 제 2 변환 유닛에 포함된 계수들과 유사한 확률 통계를 가질 수도 있다. 이것은, 더 높은 주파수의 계수들이, 엔트로피 코딩에 대한 인근 계수들의 확률 통계에 미치는 효과가 무시될 정도로 잔여 비디오 데이터가 거의 없음을 나타낼 수도 있기 때문에 가능하다. 이 경우에 있어서, 엔트로피 코딩에 대한 조인트 컨텍스트 모델은 제 1 사이즈를 가지는 변환 유닛들과 제 2 사이즈를 가지는 변환 유닛들 사이에서 공유될 수도 있다.

- [0020] 도 1 은, 상이한 사이즈들을 가지는 변환 유닛들 사이에 공유되는 조인트 콘텍스트 모델을 사용하여 비디오 계수들의 엔트로피 코딩을 수행하는 기법들을 이용할 수도 있는 예시적 비디오 인코딩 및 디코딩 시스템을 예시한 블록도이다. 도 1 에 도시된 바와 같이, 시스템 (10) 은 인코딩된 비디오를 저장할 수도 있고, 및/또는 인코딩된 비디오를 통신 채널 (16) 을 통해 목적지 디바이스 (14) 로 송신할 수도 있는 소스 디바이스 (12) 를 포함한다. 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 서로의 실시간 액티브 통신에 반드시 참여하는 것은 아닐 수도 있다. 몇몇 경우들에 있어서, 소스 디바이스 (12) 는, 목적지 디바이스 (14) 가 디스크 (disk) 액세스를 통해 회망하는 대로 액세스할 수도 있는 저장 매체에 인코딩된 비디오 데이터를 저장할 수도 있고 또는 목적지 디바이스 (14) 가 스트리밍을 통해 회망하는 대로 액세스할 수도 있는 파일 서버에 인코딩된 비디오 데이터를 저장할 수도 있다. 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 광범위한 디바이스들 중 임의의 것을 포함할 수도 있다. 몇몇 경우들에 있어서, 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 통신 채널 (16) 을 통해 비디오 정보를 통신할 수 있는 무선 통신 디바이스들을 포함할 수도 있으며, 이 경우의 통신 채널 (16) 은 무선이다.
- [0021] 그러나, 조인트 콘텍스트 모델을 사용한 비디오 계수들의 엔트로피 코딩에 관한 본 개시물의 기법들은, 반드시 무선 애플리케이션들 또는 세팅들로 제한되는 것은 아니다. 예를 들어, 이들 기법들은 OTA (over-the-air) 텔레비전 브로드캐스트, 케이블 텔레비전 송신, 위성 텔레비전 송신, 인터넷 비디오 송신, 저장 매체 상에 인코딩되는 인코딩된 디지털 비디오, 또는 다른 시나리오들에 적용할 수도 있다. 따라서, 통신 채널 (16) 은 인코딩된 비디오 데이터의 송신에 적합한 무선 또는 유선 매체들의 임의의 조합을 포함할 수도 있으며, 디바이스들 (12, 14) 은, 이동 전화, 스마트폰, 디지털 미디어 플레이어, 셋톱 박스, 텔레비전, 디스플레이, 데스크톱 컴퓨터, 휴대용 컴퓨터, 태블릿 컴퓨터, 게이밍 콘솔, 휴대용 게이밍 디바이스 등과 같은 다양한 유선 또는 무선 미디어 디바이스들 중 임의의 것을 포함할 수도 있다.
- [0022] 도 1 의 예에서, 소스 디바이스 (12) 는 비디오 소스 (18), 비디오 인코더 (20), 변조기/복조기 (모뎀) (22) 및 송신기 (24) 를 포함한다. 목적지 디바이스 (14) 는 수신기 (26), 모뎀 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (32) 를 포함한다. 다른 예들에서, 소스 디바이스 및 목적지 디바이스는 다른 컴포넌트들 또는 배열물들을 포함할 수도 있다. 예를 들어, 소스 디바이스 (12) 는, 외부 카메라, 비디오 저장 아카이브, 컴퓨터 그래픽 소스 등과 같은 외부 비디오 소스 (18) 로부터 비디오 데이터를 수신할 수도 있다. 마찬가지로, 목적지 디바이스 (14) 는 통합형 디스플레이 디바이스를 포함하는 것이 아니라 외부 디스플레이 디바이스와 인터페이스할 수도 있다.
- [0023] 도 1 의 예시된 시스템 (10) 은 단지 일 예에 불과하다. 다른 예들에서는, 임의의 디지털 비디오 인코딩 및/또는 디코딩 디바이스가 조인트 콘텍스트 모델을 사용하여 비디오 계수들의 엔트로피 코딩에 대해 개시된 기법들을 수행할 수도 있다. 이 기법들은 또한 비디오 인코더/디코더 (통상, "코덱" 이라고 지칭됨) 에 의해 수행될 수도 있다. 게다가, 이 개시물의 기법들은 또한 비디오 전처리기에 의해서 수행될 수도 있다. 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 단지 이러한 코딩 디바이스들의 예들에 불과하며, 이 소스 디바이스 (12) 는 목적지 디바이스 (14) 로의 송신을 위한 코딩된 비디오 데이터를 생성한다. 몇몇 예들에서, 디바이스들 (12, 14) 은 각각의 디바이스들 (12, 14) 이 비디오 인코딩 및 디코딩 컴포넌트들을 포함하도록 실질적으로 대칭인 방식으로 동작할 수도 있다. 따라서, 시스템 (10) 은, 예컨대 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅, 또는 비디오 전화용의 비디오 디바이스들 (12, 14) 간의 단방향 또는 양방향 비디오 송신을 지원할 수도 있다.
- [0024] 소스 디바이스 (12) 의 비디오 소스 (18) 는, 비디오 카메라, 사전 캡처 비디오를 포함하는 비디오 아카이브, 및/또는 비디오 콘텐츠 제공자로부터의 비디오 피드와 같은 비디오 캡처 디바이스를 포함할 수도 있다. 다른 대안으로서, 비디오 소스 (18) 는 소스 비디오로서 또는 라이브 비디오, 아카이브된 비디오, 및 컴퓨터 생성 비디오의 조합으로서 컴퓨터 그래픽 기반 데이터를 생성할 수도 있다. 몇몇 경우들에 있어서, 비디오 소스 (18) 가 비디오 카메라인 경우, 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 소위 카메라 폰 또는 비디오 폰을 형성할 수도 있다. 그러나, 전술한 바와 같이, 이 개시물에서 설명되는 기법들은 비디오 코딩에 전반적으로 적용 가능할 수도 있고, 무선 및/또는 유선 애플리케이션들에 적용될 수도 있다. 각각의 경우에 있어서, 캡처, 사전 캡처, 또는 컴퓨터 생성 비디오는 비디오 인코더 (20) 에 의해 인코딩될 수도 있다. 그 후, 인코딩된 비디오 정보는 통신 표준에 따라서 모뎀 (22) 에 의해 변조될 수도 있고, 송신기 (24) 를 통해 목적지 디바이스 (14) 에 송신될 수도 있다. 모뎀 (22) 은 다양한 믹서들, 필터들, 증폭기들, 또는 신호 변조용으로 설계된 다른 컴포넌트들을 포함할 수도 있다. 송신기 (24) 는, 증폭기들, 필터들, 및 하나 이상의 안테나들을 포함하는, 데이터 송신용으로 설계된 회로들을 포함할 수도 있다.

- [0025] 이 개시물에 따르면, 소스 디바이스 (12) 의 비디오 인코더 (20) 는 조인트 컨텍스트 모델을 사용하여 비디오 계수들을 엔트로피 코딩하는 기법들을 적용하도록 구성될 수도 있다. 인코딩된 비디오 프레임의 코딩 유닛 (CU) 은 잔여 비디오 데이터를 포함하는 하나 이상의 변환 유닛들 (TUs) 을 포함할 수도 있다. 변환 전, 잔여 비디오 데이터는 공간 도메인에 잔여 픽셀 값들을 포함한다. 변환 후, 잔여 비디오 데이터는 변환 도메인에 잔여 변환 계수들을 포함한다. 비디오 인코더 (20) 는 상이한 사이즈들을 가지는 변환 유닛들 사이에 공유되는 조인트 컨텍스트 모델을 유지할 수도 있고, 조인트 컨텍스트 모델에 따라서 변환 유닛들 중 하나와 관련된 계수들에 대한 컨텍스트들을 선택할 수도 있다. 그 후, 비디오 인코더 (20) 는 선택된 컨텍스트들에 기초하여 계수들을 엔트로피 인코딩할 수도 있다.
- [0026] 일 예로서, 조인트 컨텍스트 모델은 32x32 의 제 1 사이즈를 가지는 변환 유닛들과 16x16 의 제 2 사이즈를 가지는 변환 유닛들 사이에 공유될 수도 있다. 다른 예들에서, 2 개 초과 사이즈들의 변환 유닛들이 동일한 조인트 컨텍스트 모델을 공유할 수도 있다. 또한, 2 개 이상의 사이즈들의 변환 유닛들은 TUs 에 대한 컨텍스트 모델들 중 일부 또는 모두를 공유할 수도 있다. 일 경우에 있어서, 조인트 컨텍스트 모델은 TU 에 대한 유효 맵의 조인트 컨텍스트 모델일 수도 있다. 다른 경우들에 있어서, 조인트 컨텍스트 모델은 다른 코딩 정보 또는 선택스 엘리먼트들과 관련될 수도 있다. 따라서, 이 기법들은 비디오 인코더 (20) 상에 컨텍스트들 및 확률들을 저장하는 데 필요한 메모리의 양을 감소시킬 수도 있고, 비디오 인코더 (20) 상에 컨텍스트 모델들을 유지하는 계산 비용들을 저감할 수도 있다.
- [0027] 일 예에서, 비디오 인코더 (20) 는 제로화될 수도 있는데, 다시 말해 제로와 동일한 값들을 설정할 수도 있으며, 더 높은 주파수 서브세트의 변환 계수들은, 2 차원 변환의 각각의 방향이 리테인된 계수 블록을 생성하도록 적용된 후, 제 1 사이즈의 변환 유닛에 포함된다. 이 경우에 있어서, 이 기법들은, 2 차원 변환의 적용하는 각각의 방향, 즉 로우들 및 컬럼들 사이에서 버퍼링되는 계수들의 수를 감소시킬 수도 있다. 고주파수의 계수들이 변환 유닛으로부터 제로화될 때, 리테인된 계수 블록에 포함된 계수들은, 본래 제 2 사이즈인 변 호나 유닛에 포함된 계수들과 유사한 확률 통계를 갖는다. 이 예에서, 비디오 인코더 (20) 는, 제로화되어 리테인된 계수 블록을 생성하는 계수들과 함께 제 1 사이즈를 가지는 변환 유닛들 및 제 2 사이즈를 가지는 변환 유닛들에 의해 공유되는 조인트 컨텍스트 모델을 유지할 수도 있고, 조인트 컨텍스트 모델에 따라서, 리테인된 계수 블록을 가지는 제 1 사이즈 및 제 2 사이즈 중 하나의 변환 유닛의 계수들에 대한 컨텍스트들을 선택할 수도 있다. 몇몇 경우들에 있어서, 리테인된 계수 블록은 제 2 사이즈와 동일한 사이즈를 가질 수도 있다. 다른 경우들에 있어서, 리테인된 계수 블록은 제 1 사이즈 및 제 2 사이즈 양측 모두와는 상이한 제 3 사이즈와 동일한 사이즈를 가질 수도 있다.
- [0028] 다른 예에서, 제 1 사이즈를 가지는 제 1 변환 유닛에 포함된 계수들은, 제 1 변환 유닛 내에 고주파수의 계수들을 제로화하는 일 없이도, 제 2 사이즈를 가지는 제 2 변환 유닛에 포함된 계수와 유사한 확률 통계를 가질 수도 있다. 이것은, 고주파수의 계수들이, 엔트로피 코딩에 대한 인근 계수들의 확률 통계에 미치는 효과가 무시될 정도로 잔여 비디오 데이터가 거의 없음을 나타낼 수도 있기 때문에 가능하다. 이 예에서, 비디오 인코더 (20) 는 제 1 사이즈를 가지는 변환 유닛들 및 제 2 사이즈를 가지는 변환 유닛들에 의해 공유되는 조인트 컨텍스트 모델을 유지할 수도 있고, 조인트 컨텍스트 모델에 따라서 제 1 및 제 2 사이즈 중 하나의 변환 유닛 내의 계수들에 대한 컨텍스트를 선택할 수도 있다. 몇몇 경우들에 있어서, 제 1 사이즈 및 제 2 사이즈의 변환 유닛들 내의 고주파수의 계수들만이 조인트 컨텍스트 모델을 공유할 수도 있다. 제 1 사이즈의 변환 유닛 내의 저주파수의 계수들, 예컨대 DC 컴포넌트들 및 이웃 계수들은 제 2 사이즈의 변환 유닛 내의 저주파수의 계수들과는 상이한 컨텍스트 모델을 사용할 수도 있다.
- [0029] 목적지 디바이스 (14) 의 수신기 (26) 는 채널 (16) 을 통해 정보를 수신하며, 모뎀 (28) 은 정보를 복조한다. 채널 (16) 을 통해 통신되는 정보는 비디오 인코더 (20) 에 의해 정의된 선택스 정보를 포함할 수도 있고, 이 정보는 또한 비디오 디코더 (30) 에 의해서도 사용되며, 코딩 유닛들 (CUs), 예측 유닛들 (PUs), 변환 유닛들 (TUs), 또는 코딩된 비디오 (예컨대, 비디오 슬라이스들, 비디오 프레임들, 및 화상 비디오 시퀀스들 또는 화상 비디오 그룹들 (GOPs) 의 다른 유닛들의 특성들 및/또는 프로세싱을 설명하는 선택스 엘리먼트들을 포함한다. 디스플레이 디바이스 (32) 는 디코딩된 비디오 데이터를 사용자에게 디스플레이하며, 음극선관 (CRT), 액정 디스플레이 (LCD), 플라즈마 디스플레이, 및 유기 발광 다이오드 (OLED) 디스플레이, 또는 다른 유형의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들 중 임의의 것을 포함할 수도 있다.
- [0030] 이 개시물에 따르면, 목적지 디바이스 (14) 의 비디오 디코더 (30) 는 조인트 컨텍스트 모델을 사용하여 비디오 계수들을 엔트로피 디코딩하는 기법들을 적용하도록 구성될 수도 있다. 디코딩된 비디오 프레임의 CU 는, 변환 전과 변환후의 잔여 비디오 데이터를 포함하는 하나 이상의 TUs 을 포함할 수도 있다. 비디오 디코더

(30) 는 상이한 사이즈들을 가지는 변환 유닛들 사이에 공유되는 조인트 컨텍스트 모델을 유지할 수도 있고, 조인트 컨텍스트 모델에 따라 변환 유닛들 중 하나와 관련된 계수들에 대한 컨텍스트들을 선택할 수도 있다. 그 후, 비디오 디코더 (30) 는 선택된 컨텍스트들에 기초하여 계수들을 엔트로피 디코딩할 수도 있다.

[0031]

일 예로서, 조인트 컨텍스트 모델은 32x32 의 제 1 사이즈를 가지는 변환 유닛들과 16x16 의 제 2 사이즈를 가지는 변환 유닛들 사이에 공유될 수도 있다. 전술한 바와 같이, 다른 예들에서는, 2 개 초과 사이즈들의 변환 유닛들이, 동일한 조인트 컨텍스트 모델을 공유할 수도 있다. 또한, 2 이상의 사이즈들의 변환 유닛들이, TUs 에 대한 컨텍스트 모델들 중 일부 또는 모두를 공유할 수도 있다. 일 경우에 있어서, 조인트 컨텍스트 모델은 TU 에 대한 유효 맵의 조인트 컨텍스트 모델일 수도 있다. 다른 경우들에 있어서, 조인트 컨텍스트 모델은 TU 에 대한 중요 맵의 조인트 컨텍스트 모델일 수도 있다. 따라서, 이 기법들은 비디오 디코더 (30) 상에 컨텍스트들 및 확률들을 저장하는 데 필요한 메모리의 양을 감소시킬 수도 있고, 비디오 디코더 (30) 상에 컨텍스트 모델들을 유지하는 계산 비용들을 저감할 수도 있다.

[0032]

일 예에서, 비디오 디코더 (30) 는, 리테인된 계수 블록과 관련된 인코딩된 계수들 및 제 1 사이즈를 가지는 변환 유닛으로부터의 제로화된 계수들을 나타내는 비트스트림을 수신할 수도 있다. 고주파수의 계수들이 변환 유닛으로부터 제로화될 때, 리테인된 계수 블록에 포함된 계수들은, 본래 제 2 사이즈인 변환 유닛에 포함된 계수들과 유사한 확률 통계를 갖는다. 이 예에서, 비디오 디코더 (30) 는, 리테인된 계수 블록을 생성하도록 제로화된 계수들과 함께 제 1 사이즈를 가지는 변환 유닛들 및 제 2 사이즈를 가지는 변환 유닛들에 의해 공유되는 조인트 컨텍스트 모델을 유지할 수도 있고, 조인트 컨텍스트 모델에 따라서, 리테인된 계수 블록을 가지는 제 1 사이즈 및 제 2 사이즈 중 하나의 변환 유닛과 관련된 인코딩된 계수들에 대한 컨텍스트들을 선택할 수도 있다. 몇몇 경우들에 있어서, 리테인된 계수 블록은 제 2 사이즈와 동일한 사이즈를 가질 수도 있다. 다른 경우들에 있어서, 리테인된 계수 블록은 제 1 사이즈 및 제 2 사이즈와는 상이한 제 3 사이즈와 동일한 사이즈를 가질 수도 있다.

[0033]

다른 예에서, 비디오 디코더 (30) 는 제 1 사이즈 및 제 2 사이즈 중 하나를 가지는 변환 유닛과 관련된 인코딩된 계수들을 나타내는 비트스트림을 수신할 수도 있다. 제 1 사이즈를 가지는 제 1 변환 유닛에 포함된 계수들은, 제 1 변환 유닛 내의 고주파수의 계수들을 제로화하지 않고도, 제 2 사이즈의 제 2 변환 유닛에 포함된 계수들과 유사한 확률 통계를 가질 수도 있다. 이 예에서, 비디오 디코더 (30) 는 제 1 사이즈 및 제 2 사이즈를 가지는 변환 유닛들에 의해 공유되는 조인트 컨텍스트 모델을 유지할 수도 있고, 조인트 컨텍스트 모델에 따라서, 제 1 사이즈 및 제 2 사이즈 중 하나의 변환 유닛과 관련된 인코딩된 계수들에 대한 컨텍스트들을 선택할 수도 있다. 몇몇 경우들에 있어서, 제 1 사이즈 및 제 2 사이즈의 변환 유닛들 내의 고주파수의 계수들만이 조인트 컨텍스트 모델을 공유할 수도 있다. 제 1 사이즈의 변환 유닛 내의 저주파수의 계수들, 예컨대 DC 컴포넌트들 및 인근 계수들은 제 2 사이즈의 변환 유닛 내의 주파수의 계수들과는 상이한 컨텍스트 모델을 사용할 수도 있다.

[0034]

도 1 의 예에서, 통신 채널 (16) 은, 무선 주파수 (RF) 스펙트럼 또는 하나 이상의 물리적 송신 선들과 같은 임의의 무선 또는 유선 통신 매체, 또는 무선 매체와 유선 매체의 임의의 조합을 포함할 수도 있다. 통신 채널 (16) 은, 근거리 통신망, 광역 통신망, 또는 인터넷과 같은 글로벌 네트워크와 같은 패킷 기반 네트워크의 일부를 형성할 수도 있다. 통신 채널 (16) 은, 일반적으로, 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 데이터를 송신하기 위한, 유선 또는 무선 매체들의 임의의 적합한 조합을 포함하는, 임의의 적합한 통신 매체, 또는 상이한 통신 매체들의 조합을 나타낸다. 통신 채널 (16) 은 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로의 통신을 용이하게 하는 데 유용할 수도 있는 임의의 다른 장비를 포함할 수도 있다. 전술한 바와 같이, 몇몇 경우들에 있어서, 소스 디바이스 (12) 및 목적지 디바이스 (14) 는 통신 채널 (16) 을 통한 실시간 액티브 통신에 참여하지 않을 수도 있다. 예를 들어, 소스 디바이스 (12) 는, 그 대신에, 목적지 디바이스 (14) 가 디스크 (disk) 액세스를 통해 희망하는 대로 액세스할 수도 있는 저장 매체에 인코딩된 비디오 데이터를 저장할 수도 있고, 또는 목적지 디바이스 (14) 가 스트리밍을 통해 희망하는 대로 액세스할 수도 있는 파일 서버에 인코딩된 비디오 데이터를 저장할 수도 있다.

[0035]

비디오 인코더 (20) 및 비디오 디코더 (30) 는, 부상하는 고효율 비디오 코딩 (HEVC) 표준 또는 ITU-T H.264 표준과 같은 비디오 압축 표준 (대안으로, MPEG-4, Part 10, Advanced Video Coding (AVC) 이라고 지칭됨) 에 따라 동작할 수도 있다. 그러나, 이 개시물의 기법들은 임의의 특정 코딩 표준으로 제한되지 않는다. 다른 예들은 MPEG-2 및 ITU-T H.263 을 포함한다. 도 1 에는 도시되어 있지 않지만, 몇몇 양태들에 있어서, 비디오 인코더 (20) 및 비디오 디코더 (30) 는 각각 오디오 인코더 및 디코더와 통합될 수도 있으며, 적절한 MUX-DEMUX 유닛들 또는 다른 하드웨어 및 소프트웨어를 포함하여 공통 데이터 스트림 또는 개별 데이터 스트림

에서의 오디오 및 비디오 양측 모두의 인코딩을 취급할 수도 있다. 적용 가능하다면, MUX-DEMUX 유닛들은 ITU H.223 멀티플렉서 프로토콜에 또는 사용자 데이터그램 프로토콜 (user datagram protocol: UDP) 과 같은 다른 프로토콜들에 따를 수도 있다.

[0036]

HEVC 표준화 노력들은 HEVC 테스트 모델 (HM) 이라고 지칭되는 비디오 코딩 디바이스 모델에 기초한다. HM 은, 예컨대 ITU-T H.264/ AVC 에 따라서 기존 디바이스들에 대한 비디오 코딩 디바이스들의 여러 부가 기능들을 가정한다. HM 은 비디오 데이터의 블록을 코딩 유닛 (CU) 이라 지칭한다. 비트스트림 내의 신택스 데이터는 최대 코딩 유닛 (LCU) 을 정의할 수도 있으며, 이 최대 코딩 유닛은 픽셀들의 수와 관련하여 최대 코딩 유닛이다. 일반적으로 CU 는, CU 가 사이즈 특이성을 갖지 않는다는 점을 제외하면, H.264 표준의 매크로블록과 유사한 목적을 갖는다. 따라서, CU 는 서브-CUs 로 분할될 수도 있다. 일반적으로, 이 개시물에서 CU 에 대한 지칭들은 화상 (picture) 의 최대 코딩 유닛 또는 LCU 의 서브-CU 를 지칭할 수도 있다. LCU 는 서브-CUs 로 분할될 수도 있고, 각각의 서브-CU 는 서브-CUs 로 더 분할될 수도 있다. 비트스트림에 대한 신택스 데이터는, CU 깊이라고 지칭되는, LCU 가 분할될 수도 있는 최대 횟수를 정의할 수도 있다. 따라서, 비트스트림은 또한 최소 코딩 유닛 (SCU) 을 정의할 수도 있다.

[0037]

더 이상 분할되지 않는 CU (즉, LCU 의 잎 (leaf) 노드) 는 하나 이상의 예측 유닛들 (PUs) 을 포함할 수도 있다. 일반적으로 PU 는 대응하는 CU 의 전부 또는 일부를 나타내며, PU 에 대한 기준 샘플을 추출하기 위한 데이터를 포함한다. 예를 들어, PU 가 인트라-모드 인코딩될 때, PU 는 PU 에 대한 인트라-예측 모드를 설명하는 데이터를 포함할 수도 있다. 다른 예로서, PU 가 인터-모드 인코딩될 때, PU 는 PU 에 대한 모션 벡터를 정의하는 데이터를 포함할 수도 있다. 모션 벡터를 정의하는 데이터는, 예를 들어 모션 벡터의 수평 컴포넌트, 모션 벡터의 수직 컴포넌트, 모션 벡터의 분해능 (예컨대, 1/2 픽셀 정밀도, 1/4 픽셀 정밀도, 또는 1/8 픽셀 정밀도), 모션 벡터가 지적하는 기준 프레임, 및/또는 모션 벡터에 대한 기준 프레임 리스트 (예컨대, 리스트 0 또는 리스트 1) 를 설명할 수도 있다. PU(s) 를 정의한 CU 에 대한 데이터는, 예를 들어 CU 를 하나 이상의 PUs 파티셔닝하는 것을 설명할 수도 있다. 파티셔닝 모드들은, CU 가 스킵 또는 직접 모드 인코딩되는지, 인트라-예측 모드 인코딩되는지, 또는 인터-예측 모드 인코딩되는지의 사이에서 상이할 수도 있다.

[0038]

하나 이상의 PUs 를 가지는 CU 는 또한 하나 이상의 변환 유닛들 (TUs) 을 포함할 수도 있다. PU 를 사용한 예측에 이어서, 비디오 인코더는 PU 에 대응하는 CU 의 부분에 대한 잔여 값들을 계산할 수도 있다. TUs 에 포함된 잔여 값들은, 변환 계수들로 변환되고, 그 후에 양자화되고, 스캐닝되어 엔트로피 코딩에 대한 직렬 변환 계수들을 생성하도록 할 수도 있는 픽셀 차이 값들에 대응한다. TU 는 반드시 PU 의 사이즈로 제한되는 것은 아니다. 따라서, TUs 는 동일한 CU 에 대한 대응하는 PUs 보다 클 수도 있고 또는 그보다 작을 수도 있다. 몇몇 예들에서, TU 의 최대 사이즈는 대응하는 CU 의 사이즈일 수도 있다. 이 개시물은 "비디오 블록" 이라는 용어를 사용하여 CU, PU 또는 TU 중 임의의 것을 지칭한다.

[0039]

비디오 인코더 (20) 및 비디오 디코더 (30) 는, 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들 (DSPs), 주문형 반도체들 (ASICs), 필드 프로그래밍가능 게이트 어레이들 (FPGAs), 이산 로직, 소프트웨어, 하드웨어, 펌웨어 또는 이들의 임의의 조합들과 같은 다양한 적합한 인코더 회로기기 중 임의의 것으로서 구현될 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30) 의 각각은 하나 이상의 인코더들 또는 디코더들에 포함될 수도 있으며, 이들 각각은 각각의 카메라, 컴퓨터, 모바일 디바이스, 가입자 디바이스, 브로드캐스트 디바이스 셋톱 박스, 서버 등에서의 결합형 인코더/디코더 (코덱) 의 일부로서 통합될 수도 있다.

[0040]

화상들의 비디오 시퀀스 또는 화상들의 그룹 (GOP) 은 일반적으로 일련의 비디오 프레임들을 포함한다. GOP 는 GOP 의 헤더, GOP 의 하나 이상의 프레임들의 헤더, 또는 GOP 내에 포함된 프레임들의 수를 설명하는 어느 다른 곳에 신택스 데이터를 포함할 수도 있다. 각각의 프레임은 각각의 프레임에 대한 인코딩 모드를 설명하는 프레임 신택스 데이터를 포함할 수도 있다. 비디오 인코더 (20) 는 일반적으로 비디오 데이터를 인코딩하기 위해 개별적인 비디오 프레임들 내의 비디오 블록들 상에서 동작한다. 비디오 블록은 CU 또는 CU 의 PU 에 대응할 수도 있다. 비디오 블록들은 고정된 사이즈 또는 가변 사이즈들을 가질 수도 있으며, 특정 코딩 표준에 따라서 사이즈 면에서 상이할 수도 있다. 각각의 비디오 프레임은 복수의 슬라이스들을 포함할 수도 있다. 각각의 슬라이스는, 하나 이상의 PUs 를 포함할 수도 있는, 복수의 CUs 를 포함할 수도 있다.

[0041]

예로서, HEVC 테스트 모델 (HM) 은 다양한 CU 사이즈들 면에서 예측을 지원한다. LCU 의 사이즈는 신택스 정보에 의해 정의될 수도 있다. 특정 CU 의 사이즈가 $2N \times 2N$ 이라고 가정하면, HM 은 $2N \times 2N$ 또는 $N \times N$ 의 사이즈들에서 인트라-예측을 지원하고, $2N \times 2N$, $2N \times N$, $N \times 2N$, 또는 $N \times N$ 의 대칭 사이즈들에서 인터-예측을 지원한다. HM 은 또한 $2N \times nU$, $2N \times nD$, $nL \times 2N$, 및 $nR \times 2N$ 의 인터-예측에 대한 비대칭 분할을 지원한다.

비대칭 분할에서, CU 의 일 방향은 분할되지 않지만, 다른 방향은 25% 및 75% 로 분할된다. 분할된 25% 에 대응하는 CU 의 부분은 "상측", "하측", "좌측", 또는 "우측" 의 표시가 이어지는 "n" 에 의해 나타내진다. 따라서, 예를 들어, "2Nx n U" 은 상측에서 2Nx0.5N PU 으로 그리고 하측에서 Nx1.5N PU 와 수평으로 분할된 2Nx2N CU 를 지칭한다.

[0042]

이 개시물에서, " $N \times N$ " 및 " N 바이 N " 은 수평 및 수직 디멘전들, 예컨대 16x16 픽셀들 또는 16x16 픽셀들과 관련하여, 비디오 블록의 픽셀 디멘전들 (예컨대, CU, PU, 또는 TU) 을 지칭하는 데 상호 교환 가능하게 사용될 수도 있다. 16x16 블록은 수직 방향 ($y=16$) 으로 16 개의 픽셀들을 가질 것이고, 수평 방향 ($x=16$) 으로 16 개의 픽셀들을 가질 것이다. 마찬가지로, $N \times N$ 블록은 수직 방향으로 N 개의 픽셀들을 갖고, 수평 방향으로 N 개의 픽셀들을 가지며, 여기서 N 은 음수가 아닌 정수 값을 나타낸다. 블록 내의 픽셀들은 로우들 및 컬럼들로 배열될 수도 있다. 또한, 블록들은 반드시 수직 방향에서 수평 방향에서와 동일한 수의 픽셀들을 가질 필요는 없다. 예를 들어, 블록들은 $N \times M$ 개의 픽셀들을 가지는 직사각형 영역들을 포함할 수도 있으며, M 은 반드시 N 과 동일할 필요는 없다.

[0043]

CU 에 대한 PU 를 생성하기 위한 인트라-예측 또는 인터-예측 코딩에 이어서, 비디오 인코더 (20) 는 CU 에 대한 하나 이상의 TUs 를 생성하기 위해 잔여 데이터를 계산할 수도 있다. 잔여 데이터는 CU 의 PU 의 인코딩되지 않은 화상의 픽셀들과 예측 값들 사이의 픽셀 차이들에 대응할 수도 있다. 비디오 인코더 (20) 는 CU 에 대한 잔여 데이터를 포함하는 하나 이상의 TUs 를 형성할 수도 있다. 그 후, 비디오 인코더 (20) 는 TUs 를 변환할 수도 있다. 이산 코사인 변환 (DCT), 정수 변환, 웨이블릿 변환, 또는 개념적으로 유사한 변환과 같은 변환의 적용 이전에, CU 의 TUs 는 픽셀 도메인에 픽셀 차이 값들을 포함하는 잔여 비디오 데이터를 포함할 수도 있다. 변환의 적용에 이어서, TUs 는 주파수 도메인에서 잔여 비디오 데이터를 나타내는 변환 계수들을 포함할 수도 있다.

[0044]

변환 계수들을 생성하기 위한 임의의 변환들에 이어서, 변환 계수들의 양자화가 수행될 수도 있다. 양자화는, 일반적으로, 변환 계수들이, 계수들을 나타내는 데 사용되는 데이터의 양을 가능하게는 감소시키도록 양자화되는 프로세스를 지칭한다. 양자화 프로세스는 계수들 중 일부 또는 모두와 관련된 비트 깊이를 감소시킬 수도 있다. 예를 들어, n -비트 값은 양자화 동안 m -비트 값으로 버림 (round down) 될 수도 있다.

[0045]

비디오 인코더 (20) 는 지그-재그 스캔, 수평 방향 스캔, 수직 방향 스캔, 또는 양자화된 변환 계수들을 스캔하여 엔트로피 인코딩될 수 있는 직렬 벡터를 생성하도록 하는, TU 의 사이즈와 관련된 다른 스캔 순서를 적용할 수도 있다. 몇몇 예들에서, 비디오 인코더 (20) 는, 사전 정의된 스캔 순서를 이용하여, 양자화된 변환 계수들을 스캔할 수도 있다. 다른 예들에서, 비디오 인코더 (20) 는 적응적 스캔을 수행할 수도 있다. 양자화된 변환 계수들을 스캔하여 1 차원 벡터를 형성한 후, 비디오 인코더 (20) 는, 예컨대 콘텍스트 적응적 가변 길이 코딩 (context adaptive variable length coding: CAVLC), 콘텍스트 적응적 이진 산술 코딩 (context adaptive binary arithmetic coding: CABAC), 또는 다른 엔트로피 인코딩 방법에 따라서, 1 차원 벡터를 엔트로피 인코딩할 수도 있다.

[0046]

콘텍스트 적응적 엔트로피 코딩을 수행하기 위해, 비디오 인코더 (20) 는, 예를 들어 인근 계수들의 값들이 비제로 (non-zero) 인지 아닌지의 여부와 관련된 수도 있는, 콘텍스트 모델에 따라서 각각의 계수에 콘텍스트를 할당해야 한다. 그 후, 비디오 인코더 (20) 는 콘텍스트 모델에서 할당된 콘텍스트와 관련된 계수들에 대한 코딩 프로세스들을 결정한다. 통상, 비디오 인코더 (20) 는 구현된 비디오 압축 표준에 의해 지원되는 TUs 의 상이한 사이즈들 각각에 대한 별도의 콘텍스트 모델들을 유지해야 한다. HEVC 표준의 경우, 추가 변환 유닛 사이즈들, 예컨대 32x32 내지 128x128 은 비디오 코딩 효율을 향상시키도록 제안되었지만, 추가 TU 사이즈들도 또한 추가 변환 유닛 사이즈들의 각각에 대한 콘텍스트 모델들을 유지하기 위해 메모리 및 계산 요건들을 증가시킨다.

[0047]

이 개시물의 기법들에 따르면, 적응적 엔트로피 코딩을 수행하기 위해, 비디오 인코더 (20) 는 상이한 사이즈들의 TUs 사이에 공유되는 조인트 콘텍스트 모델에 따라서 계수들에 대한 콘텍스트들을 선택할 수도 있다. 더 구체적으로, 비디오 인코더 (20) 는 TU 의 사전 인코딩된 인근 계수들의 값들에 기초하여 그 TU 의 소정 계수에 조인트 콘텍스트 모델 내의 콘텍스트를 할당할 수도 있다. 할당된 콘텍스트는 TU 에 의해 공유되는 조인트 콘텍스트 모델에 의해 정의된 기준들에 기초하여 선택될 수도 있다. 비디오 인코더 (20) 는 조인트 콘텍스트 모델에서 할당된 콘텍스트와 관련된 계수들에 대한 코딩 프로세스를 결정할 수도 있다. 그 후, 비디오 인코더 (20) 는 결정된 확률 추정에 기초하여 계수를 엔트로피 인코딩한다. 예를 들어, CABAC 의 경우에 있어서, 비디오 인코더 (20) 는 조인트 콘텍스트 모델에서 할당된 콘텍스트와 관련된 계수의 값 (예컨대, 0 또는

1) 에 대한 확률 추정을 결정할 수도 있다. 그 후, 비디오 인코더 (20) 는 계수의 실제 코딩된 값에 기초하여 조인트 컨텍스트 모델에서 할당된 컨텍스트와 관련된 확률 추정을 업데이트한다.

[0048]

일 예로서, 비디오 인코더 (20) 는, 본래 제 1 사이즈를 가지는 제 1 변환 유닛 또는 본래 제 2 사이즈를 가지는 제 2 변환 유닛 중 어느 하나 내에서 리테인된 계수 블록과 관련된 계수들에 대해 동일한 조인트 컨텍스트 모델을 사용하여 컨텍스트들을 선택할 수도 있다. 또한, 비디오 인코더 (20) 는, 본래 제 1 사이즈를 가지는 변환 유닛 또는 본래 제 2 사이즈를 가지는 제 2 변환 유닛 중 어느 하나 내에서의 리테인된 계수 블록의 계수들의 실제 코딩된 값들에 기초하여 조인트 컨텍스트 모델에서 선택된 컨텍스트들과 관련된 확률 추정들을 업데이트할 수도 있다. 다른 예로서, 비디오 인코더 (20) 는 제 1 사이즈를 가지는 제 1 변환 유닛 또는 제 2 사이즈를 가지는 변환 유닛 중 어느 하나와 관련된 계수들에 대한 동일한 조인트 컨텍스트 모델을 사용하여 컨텍스트들을 선택할 수도 있다. 이 경우에 있어서, 그 후, 비디오 인코더 (20) 는 제 1 사이즈의 제 1 변환 유닛 또는 제 2 사이즈의 제 2 변환 유닛 중 어느 하나의 계수들의 실제 코딩된 값들에 기초하여 조인트 컨텍스트 모델에서 선택된 컨텍스트들과 관련된 확률 추정들을 업데이트할 수도 있다.

[0049]

어느 경우에서든, 2 개 이상의 사이즈들의 변환 유닛들 사이에 조인트 컨텍스트 모델을 공유하는 것은, 비디오 인코더 (20) 상에 컨텍스트들 및 확률들을 저장하는 데 필요한 메모리의 양을 감소시킬 수도 있다. 또한, 조인트 컨텍스트 모델을 공유하는 것은, 비디오 슬라이스의 시작부에서 모든 컨텍스트 모델들을 리셋하는 것을 포함하는, 비디오 인코더 (20) 상에 컨텍스트 모델들을 유지하는 계산 비용들을 역시 감소시킬 수도 있다.

[0050]

비디오 인코더 (20) 는 또한 예측 정보를 나타내는 선택스 엘리먼트들을 엔트로피 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는, CUs, PUs, 및 TUs 의 사이즈들을 포함한 비디오 블록 정보, 인트라-모드 예측에 대한 모션 벡터 정보, 및 CABAC 에 대한 유효 계수 맵 정보, 즉 유효 계수들의 포지션을 나타내는 1들 (ones) 및 0들 (zeros) 의 맵을 나타내는 선택스 엘리먼트들을 엔트로피 인코딩할 수도 있다. 비디오 디코더 (30) 는 비디오 인코더 (20) 와 본질적으로 대칭인 방식으로 동작할 수도 있다.

[0051]

도 2 는 조인트 컨텍스트 모델을 사용하여 비디오 계수들을 엔트로피 인코딩하는 기법들을 구현할 수도 있는 예시적 비디오 인코더를 예시한 블록도이다. 비디오 인코더 (20) 는 비디오 프레임들 내의 코딩 유닛들의 인트라-코딩 및 인터-코딩을 수행할 수도 있다. 인트라-코딩은 소정 비디오 프레임 내의 비디오에서 공간 리던던시를 감소시키거나 제거하는 공간 예측에 의존한다. 인터-코딩은 비디오 시퀀스의 인접 프레임들 내의 비디오에서 시간 리던던시를 감소시키거나 제거하는 시간 예측에 의존한다. 인트라-모드 (I 모드) 는 여러 공간 기반 압축 모드들 중 임의의 것을 지칭할 수도 있다. 단방향 예측 (P 모드), 양방향 예측 (B 모드), 또는 일반화 P/B 예측 (GPB 모드) 와 같은 인터-모드들은 여러 시간 기반 압축 모드들 중 임의의 것을 지칭할 수도 있다.

[0052]

도 2 의 예에서, 비디오 인코더 (20) 는 모드 선택 유닛 (38), 예측 유닛 (40), 합산 유닛 (50), 변환 유닛 (52), 양자화 유닛 (54), 엔트로피 인코딩 유닛 (56), 및 기준 프레임 메모리 (64) 를 포함한다. 예측 유닛 (40) 은 모션 추정 유닛 (42), 모션 보상 유닛 (44), 및 인트라 예측 유닛 (46) 을 포함한다. 비디오 블록 재구성에 대해, 비디오 인코더 (20) 는 또한 역양자화 유닛 (58), 역변환 유닛 (60), 및 합산 유닛 (52) 도 포함한다. 적응적 루프 필터 (ALF) 및 샘플 적응적 오프셋 (SAO) (도 2 에는 도시되지 않음) 과 같은 디블로킹 필터 또는 다른 인-루프 필터들이 또한 포함되어, 재구성된 비디오로부터 블로킹 아티팩트들 (blockiness artifacts) 을 제거하도록 블록 경계들을 필터링할 수도 있다. 원한다면, 디블로킹 필터는 일반적으로 합산 유닛 (62) 의 출력을 필터링할 것이다.

[0053]

도 2 에 도시된 바와 같이, 비디오 인코더 (20) 는 인코딩될 비디오 프레임 또는 슬라이스 내의 비디오 블록을 수신한다. 프레임 또는 슬라이스는 다수의 비디오 블록들 또는 CUs 로 분할될 수도 있다. 모드 선택 유닛 (38) 은 여러 결과들에 기초하여 비디오 블록에 대한 코딩 모드들, 즉 인트라 모드 또는 인터 모드 중 하나를 선택할 수도 있다. 그 후, 예측 유닛 (40) 은, 결과로서 생성된 인트라-코딩 또는 인터-코딩 예측 블록을 합산 유닛 (50) 에 제공하여 잔여 블록 데이터를 생성하고, 그리고 그 인트라-코딩 또는 인터-코딩 예측 블록을 합산 유닛 (62) 에 제공하여 기준 프레임 내의 기준 블록으로서의 사용을 위한 인코딩된 블록을 재구성한다.

[0054]

예측 유닛 (40) 내의 인트라 예측 유닛 (46) 은 코딩될 비디오 블록과 동일한 프레임 내의 하나 이상의 인근 블록들에 대해 비디오 블록의 인트라-예측 코딩을 수행한다. 예측 유닛 (40) 내의 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 은 기준 프레임 메모리 (64) 에 저장된 하나 이상의 기준 프레임들 내의 하나 이상의 기준 블록들에 대해 비디오 블록의 인터-예측 코딩을 수행한다. 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 은

고도로 집적될 수도 있지만, 개념적 목적들을 위해 개별적으로 예시된다. 모션 추정 유닛 (42) 에 의해 수행되는 모션 추정은, 비디오 블록들에 대한 모션을 추정하는, 모션 벡터들을 생성하는 프로세스이다. 모션 벡터는, 예를 들어 비디오 블록의 변위, 또는 기준 블록에 대한 현재 비디오 프레임 내의 PU, 또는 기준 프레임 내의 PU 를 나타낼 수도 있다. 기준 블록은, SAD (sum of absolute difference), SSD (sum of square difference), 또는 다른 차이 메트릭들에 의해 결정될 수도 있는 픽셀 차이와 관련하여, 코딩될 비디오 블록 또는 PU 와 밀접하게 매칭하는 것으로 밝혀지는 블록이다.

[0055] 모션 추정 유닛 (42) 은 계산된 모션 벡터를 모션 보상 유닛 (44) 에 전송한다. 모션 보상 유닛 (44) 에 의해 수행되는 모션 보상은, 모션 추정에 의해 결정된 모션 벡터에 기초하여, 예측 블록을 페치 또는 생성하는 것을 수반할 수도 있다. 비디오 인코더 (20) 는 코딩될 비디오 블록으로부터 예측 블록을 감산함으로써 잔여 비디오 블록을 형성한다. 합산 유닛 (50) 은 이 감산 연산을 수행하는 컴포넌트 또는 컴포넌트들을 나타낸다.

[0056] 모션 보상 유닛 (44) 은 비디오 시퀀스 레벨, 비디오 프레임 레벨, 비디오 슬라이스 레벨, 비디오 CU 레벨, 또는 비디오 PU 레벨 중 하나 이상에서의 예측 정보를 나타내도록 정의된 신택스 엘리먼트들을 생성할 수도 있다. 예를 들어, 모션 보상 유닛 (44) 은 CUs, PUs, 및 TUs 의 사이즈들을 포함하는 비디오 블록 정보, 및 인트라-모드 예측에 대한 모션 벡터 정보를 나타내는 신택스 엘리먼트들을 생성할 수도 있다.

[0057] 현재 비디오 블록으로부터 예측 블록을 감산함으로써 비디오 인코더 (20) 가 잔여 비디오 블록을 형성한 후, 변환 유닛 (52) 은 잔여 블록으로부터 하나 이상의 TUs 를 형성할 수도 있다. 변환 유닛 (52) 은, 이산 코사인 변환 (DCT), 정수 변환, 웨이블릿 변환, 또는 개념적으로 유사한 변환과 같은 변환을 TU 에 적용하여, 잔여 변환 계수들을 포함하는 비디오 블록을 생성할 수도 있다. 변환은 잔여 블록을 픽셀 도메인으로부터 주파수 도메인과 같은 변환 도메인으로 전환될 수도 있다. 더 구체적으로, 변환의 적용 이전에, TU 는 픽셀 도메인에 잔여 비디오 데이터를 포함할 수도 있고, 변환의 적용 후, TU 는 주파수 도메인에서 잔여 비디오 데이터를 나타내는 변환 계수들을 포함할 수도 있다.

[0058] 몇몇 예들에서, 변환 유닛 (52) 은 2 차원의 분리가능 변환을 포함할 수도 있다. 변환 유닛 (52) 은, 먼저 TU 내의 잔여 비디오 데이터의 로우들에, 즉 제 1 방향으로, 1 차원 변환을 적용하고, 그 후에, TU 내의 잔여 비디오 데이터의 컬럼들에, 즉 제 2 방향에 1 차원 변환을 적용함으로써, 또는 순서를 역으로 적용함으로써, TU 에 2 차원 변환을 적용할 수도 있다. 일 예로서, TU 는 32x32 TU 를 포함할 수도 있다. 변환 유닛 (52) 은, 먼저 32-포인트 1 차원 변환을 TU 내의 픽셀 데이터의 각각의 로우에 적용하여 중간 변환 계수들의 32x32 TU 를 생성하고, 두 번째로, 32-포인트 1 차원 변환을 TU 내의 중간 변환 계수들의 각각의 컬럼에 적용하여 변환 계수들의 32x32 TU 를 생성할 수도 있다.

[0059] TU 내의 잔여 비디오 데이터에 제 1 방향으로 1 차원 변환을 적용한 후, 비디오 인코더 (20) 는 제 2 방향으로 1 차원 변환을 적용하기 위한 중간 변환 계수들을 버퍼링한다. 전술한 바와 같이, HEVC 표준에서, 더 큰 변환 유닛 사이즈들, 예컨대 32x32 내지 128x128 은 비디오 코딩 효율을 향상시키도록 제안되었다. 그러나, 더 큰 TU 사이즈들은 또한 2 차원 변환에 대한 중간 버퍼링 요건들을 증가시킬 것이다. 예를 들어, 32x32 TU 의 경우에 있어서, 비디오 인코더 (20) 는 제 1 방향으로의 1 차원 변환 후에 1024 개의 중간 변환 계수들을 버퍼링해야 할 것이다.

[0060] 더 큰 TU 사이즈들에 대한 중간 버퍼링 요건들을 감소시키기 위해, 이 개시물에 설명된 기법들은, 2 차원 변환의 각각의 방향이 적용된 후, 제 1 사이즈의 TU 에 포함된 고주파수의 변환 계수 서브세트를 제로화하는 것을 포함한다. 이 방식으로, 변환 유닛 (52) 은 TU 의 제 1 사이즈보다 작은 제 2 사이즈를 가지는, TU 내에 리테인된 계수 블록을 생성할 수도 있다.

[0061] 제로화하는 프로세스는, 제로와 동일한 TU 내의 변환 계수 서브세트의 값들을 세팅하는 것을 포함한다. 제로화되는 변환 계수들은 계산되지 않거나 폐기된다; 그 대신, 제로화된 변환 계수들은 단순히 제로로 설정되고 저장하거나 인코딩할 어떠한 값도 갖지 않는다. 이 개시물에 따르면, 제로화된 변환 계수들은 일반적으로 TU 에서 리테인된 저주파수의 변환 계수들에 대해 더 높은 주파수의 변환 계수들이다. 고주파수의 변환 계수들은, 대개는 인코딩될 비디오 블록과 예측 블록 사이의 매우 작은 픽셀 차이들에 대응하는 잔여 비디오 데이터를 나타낸다. 따라서, 고주파수 변환 계수들은, 값들을 제로로 설정하는 것이, 디코딩된 비디오 품질에 대해 무시할 수 있는 효과를 가질 정도로 잔여 비디오 데이터를 거의 포함하지 않을 수도 있다.

[0062] 일 예로서, 변환 유닛 (52) 은, 제 1 방향의, 예컨대 로우 방향 (row-wise) 의 1 차원 변환을 32x32 TU 내의

잔여 비디오 데이터에 적용할 수도 있고, 변환으로부터 출력된 중간 변환 계수들의 절반을 제로화할 수도 있다. 그 후, 비디오 인코더 (20) 는 단지 중간 변환 계수들의 리테인된 절반만을 버퍼링하면 된다. 그러면, 변환 유닛 (52) 은, 제 2 방향의, 예컨대 컬럼 방향 (column-wise) 의 1 차원 변환을 32×32 TU 내의 리테인된 중간 변환 계수들에 적용할 수도 있다. 이 방식으로, 변환 유닛 (52) 은, 본래 사이즈 32×32 의 TU 내에서 사이즈 16×16 을 가지는 유효 계수들의 리테인된 계수 블록을 생성할 수도 있다.

[0063]

전술한 예에서, 변환 유닛 (52) 은 16×16 리테인된 계수 블록, 즉 TU 의 본래 사이즈의 $1/4$ 를 생성하도록 구성되었다. 다른 경우들에 있어서, 변환 유닛 (52) 은, 코딩 프로세스에 대한 코딩 복잡도 요건들에 의존하여, 더 크거나 더 작은 백분율의 계수들을 제로화함으로써 상이한 사이즈를 가지는 리테인된 계수 블록을 생성하도록 구성될 수도 있다. 또한, 몇몇 경우들에 있어서, 변환 유닛 (52) 은 직사각형 영역을 가지는 리테인된 계수 블록을 생성하도록 구성될 수도 있다. 이 경우에 있어서, 이 기법들은, 직사각형 영역의 더 짧은 변 (즉, 더 작은 리테인된 변환 계수들) 의 방향으로 1 차원 변환을 적용함으로써 중간 버퍼링 요건들에서의 추가 감소를 제공한다. 이 방식으로, 비디오 인코더 (20) 는, 직사각형 영역의 더 긴 변의 방향으로 1 차원 변환을 적용하기 전에 중간 변환 계수들의 절반 미만을 버퍼링할 수도 있다. 정사각형 및 직사각형 영역의 리테인된 계수 블록들 양측 모두에 대한 제로화 프로세스들은 도 3a 및 도 3b 와 관련하여 더 상세히 설명된다.

[0064]

변환 유닛 (52) 은 결과로서 생성된 변환 계수들을 양자화 유닛 (54) 에 전송할 수도 있다. 양자화 유닛 (54) 은 변환 계수들을 양자화하여 비트 레이트를 더 감소시킨다. 양자화 프로세스는 계수들 중 일부 또는 모두와 관련된 비트 깊이를 감소시킬 수도 있다. 양자화 정도는 양자화 파라미터를 조절함으로써 수정될 수도 있다. 엔트로피 인코딩 유닛 (56) 또는 양자화 유닛 (54) 은, 양자화된 변환 계수들을 포함하는 TU 의 스캔을 수행할 수도 있다. 엔트로피 인코딩 유닛 (56) 은 엔트로피 인코딩될 수 있는 직렬 벡터를 생성하도록 지그-재그 스캔 또는 TU 의 사이즈와 관련된 다른 스캔 순서를 적용할 수도 있다.

[0065]

본래 제 1 사이즈인 TU 의 계수들이 제로화되어 제 2 사이즈를 가지는 리테인된 계수 블록을 생성하는 일 예에서, 엔트로피 인코딩 유닛 (56) 은 제 2 사이즈의 TU 에 대한 스캔 순서를 사용하여, 리테인된 계수들을 스캔할 수도 있다. 이 경우에 있어서, 엔트로피 인코딩 유닛 (56) 은 16×16 스캔 순서를 본래 32×32 의 사이즈인 TU 내에 사이즈 16×16 을 가지는 리테인된 계수 블록에 적용할 수도 있다. 본래 제 1 사이즈인 TU 의 계수들이 제로화되어 제 2 사이즈를 가지는 리테인된 계수 블록을 생성하는 다른 예에서, 엔트로피 인코딩 유닛 (56) 은, 리테인된 계수 블록에 포함되지 않은 TU 의 계수들을 스킵하도록 수정된 제 1 사이즈의 TU 에 대한 스캔 순서를 사용하여 리테인된 계수들을 스캔할 수도 있다. 이 경우에 있어서, 엔트로피 인코딩 유닛은, 본래 사이즈 32×32 의 TU 내의 모든 제로화된 계수들을 스킵함으로써 사이즈 16×16 을 가지는 리테인된 계수 블록에 32×32 스캔 순서를 적용할 수도 있다.

[0066]

양자화된 변환 계수들을 스캔하여 1 차원 벡터를 형성한 후, 엔트로피 인코딩 유닛 (56) 은 양자화된 변환 계수들의 벡터를 엔트로피 코딩한다. 예를 들어, 엔트로피 인코딩 유닛 (56) 은, CABAC, CAVLC 과 같은 콘텍스트 적응적 엔트로피 인코딩 또는 다른 엔트로피 코딩 기법을 수행할 수도 있다. 엔트로피 인코딩 유닛 (56) 에 의한 엔트로피 인코딩에 이어서, 인코딩된 비트스트림은, 비디오 디코더 (30) 와 같은 비디오 디코더에 송신될 수도 있고, 또는 추후 송신이나 추출을 위해 아카이브될 수도 있다.

[0067]

콘텍스트 적응적 엔트로피 코딩을 수행하기 위해, 엔트로피 인코딩 유닛 (56) 은, 예를 들어 인근 계수들의 값들이 비제로 (non-zero) 인지 아닌지의 여부와 관련될 수도 있는, 콘텍스트 모델에 따라서 각각의 계수에 콘텍스트를 할당한다. 엔트로피 인코딩 유닛 (56) 은 또한 콘텍스트 모델에서 할당된 콘텍스트와 관련된 계수에 대한 코딩 프로세스를 결정한다. 그 후, 엔트로피 인코딩 유닛 (56) 은 할당된 콘텍스트들에 기초하여 계수들을 엔트로피 인코딩한다. 예를 들어, CABAC 의 경우에 있어서, 엔트로피 인코딩 유닛 (56) 은 조인트 콘텍스트 모델에서 할당된 콘텍스트와 관련된 계수의 값 (예컨대, 0 또는 1) 에 대한 확률 추정을 결정할 수도 있다. 그 후, 엔트로피 인코딩 유닛 (56) 은 계수의 실제 인코딩된 값에 기초하여 콘텍스트에서 할당된 콘텍스트와 관련된 확률 추정을 업데이트한다.

[0068]

통상, 비디오 인코더 (20) 는 구현된 비디오 압축 표준에 의해 지원되는 상이한 사이즈들의 TUs 의 각각에 대한 별도의 콘텍스트 모델들을 유지한다. HEVC 표준에 대해, 추가의 변환 유닛 사이즈들, 예컨대 32×32 내지 128×128 은 비디오 코딩 효율을 향상시키도록 제안되었지만, 추가 TU 사이즈들은 또한 추가 변환 유닛 사이즈들의 각각에 대한 콘텍스트 모델들을 유지하기 위한 메모리 및 계산적 요건들이 증가하게 된다. 몇몇 경우에 있어서, 더 큰 TU 사이즈들은, 더 큰 TU 사이즈들에 대한 증가한 수의 콘텍스트들을 유지하도록 메모리 및 계산 요건을 증가시킬 수도 있다.

- [0069] 이 개시물의 교시사항들에 따르면, 엔트로피 인코딩 유닛 (56) 은 상이한 사이즈들을 가지는 변환 유닛들 사이에 공유되는 조인트 컨텍스트 모델을 사용하여 비디오 계수들의 엔트로피 인코딩을 수행하도록 구성될 수도 있다. 이 기법들은, 주포, CABAC 의 엔트로피 인코딩 기법에 대해서 설명된다. 그러나, 몇몇 경우들에 있어서, 이 기법들은 또한 다른 컨텍스트 적응적 엔트로피 코딩 기법들에 적용될 수도 있다. 2 개 이상의 사이즈들의 변환 유닛들 사이에 조인트 컨텍스트 모델을 공유하는 것은, 비디오 인코더 (20) 상에 컨텍스트들 및 확률들을 저장하는 데 필요한 메모리의 양을 감소시킬 수도 있다. 또한, 조인트 컨텍스트 모델을 공유하는 것은, 또한 비디오 슬라이스의 시작부에서 모든 컨텍스트 모델들을 리셋하는 것을 포함하는, 비디오 인코더 (20) 상에 컨텍스트 모델들을 유지하는 계산 비용들을 저감할 수도 있다. CABAC 의 경우에 있어서, 이 기법들은 또한 계수들의 실제 코딩된 값들에 기초하여 컨텍스트 모델들의 확률 추정들을 계속적으로 업데이트하는 계산 비용을 저감할 수도 있다.
- [0070] 이 개시물에 따르면, 엔트로피 인코딩 유닛 (56) 은 상이한 사이즈들을 가지는 변환 유닛들 사이에 공유되는 조인트 컨텍스트 모델을 유지할 수도 있고, 조인트 컨텍스트 모델에 따라서 변환 유닛들 중 하나와 관련된 계수들에 대한 컨텍스트들을 선택할 수도 있다. 그 후, 엔트로피 인코딩 유닛 (56) 은 선택된 컨텍스트들에 기초하여 변환 유닛들 내의 유효 계수들을 엔트로피 인코딩할 수도 있다. 일 예로서, 조인트 컨텍스트 모델은 32x32 의 제 1 사이즈를 가지는 변환 유닛들과 16x16 의 제 2 사이즈를 가지는 변환 유닛들 사이에 공유될 수도 있다. 몇몇 경우들에 있어서, 2 개 초과 사이즈들의 변환 유닛들이 동일한 조인트 컨텍스트 모델을 공유할 수도 있다. 일 예에서, 조인트 컨텍스트 모델은 변환 유닛들에 대한 유효 맵들의 조인트 컨텍스트 모델일 수도 있다. 다른 예들에서, 조인트 컨텍스트 모델은 다른 코딩 정보 또는 신텍스 엘리먼트들과 관련될 수도 있다. 조인트 컨텍스트 모델들을 사용하는 CABAC 인코딩 프로세스는 도 5 와 관련하여 더 상세히 설명된다.
- [0071] 일 예에서, 엔트로피 인코딩 유닛 (56) 은 리테인된 계수 블록을 생성하도록 제로화된 계수들과 함께 제 1 사이즈를 가지는 TU 및 본래 제 2 사이즈를 가지는 TU 사이에 공유될 수도 있다. 몇몇 경우들에서, 리테인된 계수 블록은 제 2 사이즈와 동일한 사이즈를 가질 수도 있다. 예를 들어, 고주파수의 계수들이 제 1 사이즈의 TU 로부터 제로화될 때, 제 2 사이즈의 리테인된 계수 블록에 포함된 계수들은, 본래 제 2 사이즈인 TU 에 포함된 계수들과 유사한 확률 통계를 갖는다. 이 경우에 있어서, 제 2 블록 사이즈의 리테인된 계수 블록이, 본래 제 1 사이즈인 TU 의 계수들을 제로화함으로써 생성될 때, 엔트로피 인코딩 유닛 (56) 은 조인트 컨텍스트 모델에 따라서 리테인된 계수 블록의 계수들에 대한 컨텍스트들을 선택할 수도 있다. 그 후, 엔트로피 인코딩 유닛 (56) 은 선택된 컨텍스트들에 기초하여 리테인된 계수 블록 내의 유효 계수들을 엔트로피 인코딩한다. CABAC 의 경우에 있어서, 엔트로피 인코딩 유닛 (56) 은 또한 계수들의 실제 코딩된 값들에 기초하여 조인트 컨텍스트 모델에서 선택된 컨텍스트들과 관련된 확률 추정들을 업데이트한다.
- [0072] 다른 예에서, 엔트로피 인코딩 유닛 (56) 은 제 1 사이즈를 가지는 제 1 TU 및 제 2 사이즈를 가지는 제 2 TU 에 의해 공유되는 조인트 컨텍스트 모델을 유지할 수도 있다. 몇몇 경우들에 있어서, 제 1 사이즈를 가지는 제 1 TU 에 포함된 계수들은, 제 1 TU 내의 고주파수의 계수들을 제로화하는 일 없이도, 제 2 사이즈를 가지는 제 2 TU 에 포함된 계수들과 유사한 확률 통계를 가질 수도 있다. 이것은, 고주파수의 계수들이, 엔트로피 코딩에 대한 인근 계수들의 확률 통계에 미치는 영향이 무시될 수 있을 정도로 잔여 비디오 데이터가 거의 없음을 나타낼 수도 있기 때문에 가능하다. 이 예에서, 엔트로피 인코딩 유닛 (56) 은 조인트 컨텍스트 모델에 따라서 제 1 및 제 2 사이즈 중 하나의 TU 내의 계수들에 대한 컨텍스트들을 선택할 수도 있다. 그 후, 엔트로피 인코딩 유닛 (56) 은 선택된 컨텍스트들에 기초하여 TU 내의 유효 계수들을 엔트로피 인코딩한다. 그러면, CABAC 의 경우에 있어서, 엔트로피 인코딩 유닛 (56) 은 계수들의 실제 코딩된 값들에 기초하여 조인트 컨텍스트 모델에서 선택된 컨텍스트들과 관련된 확률 추정들을 업데이트할 수도 있다.
- [0073] 엔트로피 인코딩 유닛 (56) 은 또한 코딩될 비디오 블록에 대한 다른 예측 정보 및 모션 벡터들을 나타내는 신텍스 엘리먼트들을 엔트로피 인코딩할 수도 있다. 예를 들어, 엔트로피 인코딩 유닛 (56) 은, 이 개시물에서 설명되는 기법들을 이용하여, 유효 맵, 즉 TU 에서 유효 계수들의 위치를 나타내는 0들과 1들의 맵을 나타내는 신텍스 엘리먼트들을 엔트로피 인코딩할 수도 있다. 엔트로피 인코딩 유닛 (56) 은 인코딩된 비트스트림에서의 송신을 위해 모션 보상 유닛 (44) 에 의해 생성된 적절한 신텍스 엘리먼트들을 포함하는 헤더 정보를 또한 구성할 수도 있다. 비디오 인코더 (20) 는 CUs, PUs, 및 TUs 의 사이즈들을 포함하는 비디오 블록 정보, 및 인트라-모드 예측을 위한 모션 벡터 정보를 나타내는 신텍스 엘리먼트들을 엔트로피 인코딩할 수도 있다. 신텍스 엘리먼트들을 엔트로피 인코딩하기 위해, 엔트로피 인코딩 유닛 (56) 은, 신텍스 엘리먼트들을 하나 이상의 이전 비트들로 이진화함으로써, 그리고 컨텍스트 모델에 따라서 각각의 이전 비트에 대한 콘

스트들을 선택함으로써 CABAC 를 수행할 수도 있다.

- [0074] 역양자화 유닛 (58) 및 역변환 유닛 (60) 은 각각 역양자화 및 역변환을 적용하여 기준 프레임의 기준 블록으로서 추후에 사용하기 위해 픽셀 도메인에 잔여 블록을 재구성한다. 합산 유닛 (62) 은 재구성된 잔여 블록을 모션 보상 유닛 (44) 에 의해 생성된 예측 블록에 더하여, 기준 프레임 메모리 (64) 에 저장되는 기준 블록을 생성한다. 기준 블록은 모션 추정 유닛 (42) 및 모션 보상 유닛 (44) 에 의해 기준 블록으로서 사용되어 후속 비디오 프레임의 블록을 인터-예측하도록 할 수도 있다.
- [0075] 도 3a 및 도 3b 는 제 1 사이즈를 가지는 변환 유닛 내에 제 2 사이즈를 가지는 리테인된 계수 블록들의 정사각형 및 직사각형 영역들을 각각 예시한 개념도들이다. 도 3a 및 도 3b 의 개념 예들에서는, 상이한 박스들이 변환 적용 후의 TU 내에서의 변환 계수들을 나타낸다. 해칭 표시된 영역들 내의 계수들은 리테인된 변환 계수들을 포함하고, 해칭 표시되지 않은 영역들 (즉, 백색 또는 음영 표시 박스들) 에 포함되지 않은 계수들은 2 차원 변환 동안에 제로화된 계수들을 포함한다.
- [0076] 전술한 바와 같이, 이 개시물의 기법들은, 각각의 방향의 2 차원 변환 후, 제 1 사이즈의 TU 에 포함된 고주파수의 변환 계수 서브세트를 제로화함으로써, 더 큰 TU 사이즈들, 예컨대 32x32 내지 128x128 에 대한 중간 버퍼링 요건들을 감소시킨다. 이 방식으로, 도 2 로부터의 변환 유닛 (52) 은 제 1 사이즈의 TU 보다 작은 제 2 사이즈를 가지는 TU 내에 리테인된 계수 블록을 생성할 수도 있다.
- [0077] 제로화하는 프로세스는, TU 내의 변환 계수 서브세트의 값들을 제로와 동일하게 설정하는 것을 포함한다. 제로화된 변환 계수들은 계산되지 않거나 무시되며; 그 대신, 제로화된 변환 계수들은 간단히 제로와 동일하게 설정되고, 저장 또는 인코딩할 값을 갖지 않는다. 이 개시물에 따르면, 제로화된 변환 계수들은 일반적으로 TU에서 리테인된 변환 계수들에 대한 고주파수의 변환 계수들이다. 고주파수의 변환 계수들은 인코딩될 비디오 블록과 예측 블록 사이의 매우 작은 픽셀 차이들에 대응하는 잔여 비디오 데이터를 나타낸다. 따라서, 고주파수의 변환 계수들은, 제로와 동일한 값들을 설정하는 것이, 디코딩된 비디오 품질에 대한 무시할 만한 효과를 가질 정도로 잔여 비디오 데이터를 거의 포함하지 않을 수도 있다.
- [0078] 도 3a 는 16x16 의 제 1 사이즈를 가지는 TU (70) 및 TU (70) 내에서 제 2 사이즈 8x8 의 최종 정사각형 영역을 가지는 리테인된 계수 블록 (74) 을 예시한다. 리테인된 계수 블록 (74) 의 사이즈 및 형상은 코딩 프로세스에 대한 코딩 복잡도 요건들에 기초하여 선택될 수도 있다. 이 예에서, 도 2 로부터의 변환 유닛 (52) 은, 본래 사이즈 16x16 의 TU (70) 내에 사이즈 8x8 의 정사각형 영역을 가지는 리테인된 계수 블록 (74) 을 생성하도록 구성될 수도 있다. 리테인된 계수 블록 (74) 을 생성하기 위해, 변환 유닛 (52) 은, 각각의 방향, 즉 로우들 및 컬럼들의 2 차원 변환을 적용한 후, 더 높은 주파수 계수들의 절반을 제로화할 수도 있다. 다른 경우들에 있어서, 변환 유닛 (52) 은 코딩 프로세스에 대한 코딩 복잡도 요건들에 의존하여 더 큰 서브세트 또는 더 작은 서브세트의 계수들을 제로화하도록 구성될 수도 있다.
- [0079] 먼저, 변환 유닛 (52) 은 TU (70) 에서 잔여 비디오 데이터의 로우들에 1 차원 변환을 적용할 수도 있고, 변환으로부터 출력된 중간 변환 계수들의 서브세트 (이 경우에는, 절반) 를 제로화할 수도 있다. 도 3a 의 예시된 예에서, 리테인된 중간 변환 계수들은, TU (70) 의 본래의 16x16 사이즈의 절반과 동일한 16x8 의 직사각형을 가지는 리테인된 중간 계수 블록 (73)(즉, TU (70) 의 음영 블록들) 에 포함된다. 제로화된 서브세트 (즉, TU (70) 의 백색 블록들) 는 TU (70) 의 리테인된 중간 계수 블록 (73) 내의 계수들보다 더 높은 주파수 값들을 가지는 계수들을 포함할 수도 있다. 예시된 예에서, 변환 유닛 (52) 은 TU (70) 의 각각의 로우에서 8 개의 최고 주파수 값들을 가지는 계수들의 절반을 제로화한다. 이 제로화 프로세스는 TU (70) 내의 16x8 의 직사각형 영역을 가지는 리테인된 중간 계수 블록 (73) 을 초래한다. 다른 예들에서, 리테인된 중간 계수 블록 (73) 의 영역은 상이한 사이즈 또는 형상을 포함할 수도 있다. 변환 유닛 (52) 은 16x16 TU (70) 에서 모든 계수들 중 최고 주파수 값들을 가지는 계수들의 절반을 제로화할 수도 있다. 이 제로화 프로세스는 TU (70) 의 상위 좌측 코너에 있는 직사각형 영역을 가지는 리테인된 중간 계수 블록을 초래할 수도 있다.
- [0080] 제 1 방향의 변환으로부터 출력된 중간 변환 계수들의 절반을 제로화함으로써, 비디오 인코더 (20) 는, 제 2 방향의 변환을 적용하기 전에, 리테인된 중간 계수 블록 (73) 내의 계수들만을 버퍼링하면 된다. 제로화된 서브세트 (즉, TU (70) 에서 백색 블록들) 내의 계수들은 저장하거나 변환하거나 인코딩할 어떠한 값도 갖지 않는다. 이 방식으로, 이 기법들은 2 차원 변환들을 수행할 때의 중간 버퍼링 요건들을 감소시킬 수도 있다. 이것은, HEVC 표준에 대해 제안된 더 큰 변환 유닛 사이즈들, 예컨대 32x32 내지 128x128 에 특히 유용할 수도 있다.

- [0081] 버퍼링 후, 변환 유닛 (52) 은, 리테인된 중간 계수 블록 (73) 의 리테인된 중간 변환 계수들의 컬럼들에 1 차원 변환을 적용할 수도 있고, 변환으로부터 출력된 변환 계수들의 서브세트 (이 경우에는, 절반) 를 제로화할 수도 있다. 도 3a 의 예시된 예에서, 리테인된 변환 계수들은, TU (70) 의 본래 16x16 사이즈의 1/4 와 동일한 사이즈 8x8 의 정사각형 영역을 가지는 리테인된 계수 블록 (74)(즉, TU (70) 에서 해칭 표시된 블록들) 에 포함된다. 제로화된 서브세트 (즉, 리테인된 중간 계수 블록 (73) 에서 해칭 표시되지 않은 정사각형들) 는 TU (70) 의 리테인된 계수 블록 (74) 내의 계수들보다 더 높은 주파수 값들을 가지는 계수들을 포함할 수도 있다. 예시된 예에서, 변환 유닛 (52) 은 리테인된 중간 계수 블록 (73) 의 각각의 컬럼에서 8 개의 최고 주파수 값들을 가지는 계수들의 절반을 제로화한다. 다른 예들에서, 변환 유닛 (52) 은 16x8 리테인된 중간 계수 블록 (73) 의 모든 계수들 중 최고 주파수 값들을 가지는 계수들의 절반을 제로화할 수도 있다. 어느 경우이든, 제로화 프로세스는 TU (70) 내의 8x8 의 정사각형 영역을 가지는 리테인된 계수 블록 (74) 을 초래한다.
- [0082] 도 3b 는 TU (76) 내의 제 2 사이즈 4x16 의 최종 직사각형 영역을 가지는 리테인된 계수 블록 (78) 및 16x16 의 제 1 사이즈를 가지는 TU (76) 를 예시한다. 리테인된 계수 블록 (78) 의 사이즈 및 형상은 코딩 프로세스에 대한 코딩 복잡도 요건에 기초하여 선택될 수도 있다. 더 구체적으로, 리테인된 계수 블록 (78) 의 최종 직사각형 영역은, 인트라-코딩 모드, 스캔 패턴, 및 리테인된 계수 블록 (78) 에 대한 최종 유효 계수의 포지션 중 적어도 하나에 기초하여 선택될 수도 있다.
- [0083] 도 3b 의 예시된 예에서, 변환 유닛 (52) 은 본래 사이즈 16x16 인 TU (76) 내의 사이즈 4x16 의 직사각형 영역을 가지는 리테인된 계수 블록 (78) 을 생성하도록 구성될 수도 있다. 변환 유닛 (52) 은 먼저 리테인된 계수 블록 (78) 의 최종 직사각형 영역의 최단 변 (예컨대, 컬럼들) 의 방향으로 TU (76) 에 적용될 수도 있다. 리테인된 계수 블록 (78) 을 생성하기 위해, 변환 유닛 (52) 은 제 1 방향 (예컨대, 컬럼들) 의 2 차원 변환을 적용한 후 계수들의 3/4 를 제로화할 수도 있다. 다른 경우들에 있어서, 변환 유닛 (52) 은 코딩 프로세스에 대한 코딩 복잡도 요건들에 의존하여 더 큰 서브세트 또는 더 작은 서브세트의 계수들을 제로화하도록 구성될 수도 있다.
- [0084] 직사각형 영역이 리테인된 계수 블록에 대해 선택될 때, 이 기법들은 직사각형 영역의 최단 변 (즉, 더 적은 리테인된 변환 계수들) 의 방향으로 1 차원 변환을 먼저 적용함으로써 중간 버퍼링 요건들에서의 추가 감소를 제공할 수도 있다. 이 방식으로, 비디오 인코더 (20) 는, 직사각형 영역의 더 긴 변의 방향으로 1 차원 변환을 적용하기 전, 중간 변환 계수들의 절반 미만을 버퍼링할 수도 있다. 도 3b 의 예시된 예에서, 리테인된 계수 블록 (78) 의 높이 (H) 는 직사각형 영역의 폭 (W) 보다 작으므로, 더 적은 중간 변환 계수들이 수직 방향으로 리테인될 것이다. 따라서, 변환 유닛 (52) 은 먼저 TU (76) 의 컬럼들에 변환을 적용하여, 비디오 인코더 (20) 가, TU (76) 의 로우들에 변환을 적용하기 전, 중간 변환 계수들의 절반 미만 (이 경우에는, 1/4) 을 버퍼링할 수도 있다.
- [0085] 더 구체적으로, 변환 유닛 (52) 은 TU (76) 에서 잔여 비디오 데이터의 컬럼들에 1 차원 변환을 적용할 수도 있고, 변환으로부터 출력된 중간 변환 계수들의 서브세트 (이 경우에는, 3/4) 를 제로화할 수도 있다. 도 3b 의 예시된 예에서, 리테인된 중간 변환 계수들은, TU (76) 의 본래 16x16 사이즈의 1/4 과 동일한 4x16 의 직사각형 영역을 가지는 리테인된 중간 계수 블록 (77)(즉, TU (76) 에서 음영 블록들) 에 포함된다. 제로화된 서브세트 (즉, TU (76) 의 백색 블록들) 는 TU (76) 의 리테인된 중간 계수 블록 (77) 내의 계수들보다 더 높은 주파수 값들을 가지는 계수들을 포함할 수도 있다. 예시된 예에서, 변환 유닛 (52) 은 TU (76) 의 각각의 컬럼에서 12 개의 최고 주파수 값들을 가지는 계수들의 3/4 를 제로화한다. 제로화 프로세스는 TU (76) 내의 4x16 의 직사각형 영역을 가지는 리테인된 중간 계수 블록 (77) 을 초래한다.
- [0086] 제 1 방향의 변환으로부터 출력된 중간 변환 계수들의 3/4 를 제로화함으로써, 비디오 인코더 (20) 는, 제 2 방향의 변환을 적용하기 전에, 리테인된 중간 계수 블록 (77) 내의 계수들만을 버퍼링하면 된다. 제로화된 서브세트 (즉, TU (76) 의 백색 블록들) 내의 계수들은 저장하거나 변환하거나 인코딩할 어떠한 값도 갖지 않는다. 이 방식으로, 이 기법들은 2 차원 변환들을 수행할 때의 중간 버퍼링 요건들을 감소시킬 수도 있다. 이것은, HEVC 표준에 대해 제안된 더 큰 변환 유닛 사이즈들, 예컨대 32x32 내지 128x128 에 특히 유용할 수도 있다.
- [0087] 버퍼링 후, 변환 유닛 (52) 은 리테인된 중간 계수 블록 (77) 에서 계수들의 로우들에 1 차원 변환을 적용할 수도 있다. 이 예에서, 변환 유닛 (52) 은, TU (76) 가 16x16 의 본래 사이즈의 1/4 까지 이미 제로화되어 있기 때문에 변환으로부터 출력된 변환 계수들 중 어떠한 것도 제로화하지 않을 수도 있다. 도 3a 의 예시된

예에서, 리테인된 변환 계수들은, 리테인된 중간 계수 블록 (77) 과 동일한 4x16 의 직사각형 영역을 가지는 리테인된 계수 블록 (78)(즉, TU (76) 의 해칭 표시된 블록들) 에 포함된다. 제로화 프로세스는 TU (70) 내의 4x16 의 직사각형 영역을 가지는 리테인된 계수 블록 (78) 을 초래한다.

[0088]

도 4 는 조인트 컨텍스트 모델을 사용하여 비디오 계수들을 엔트로피 디코딩하는 기법들을 구현할 수도 있는 예시적 비디오 디코더를 예시한 블록도이다. 도 4 의 예에서, 비디오 디코더 (30) 는 엔트로피 디코딩 유닛 (80), 예측 유닛 (81), 역양자화 유닛 (86), 역변환 유닛 (88), 합산 유닛 (90), 및 기준 프레임 메모리 (92) 를 포함한다. 예측 유닛 (81) 은 모션 보상 유닛 (82) 및 인트라 예측 유닛 (84) 을 포함한다. 비디오 디코더 (30) 는, 몇몇 예들에서, 비디오 인코더 (20, 도 2) 에 대해 설명된 인코딩 패스에 대해 전반적으로 상호적인 디코딩 패스를 수행할 수도 있다.

[0089]

디코딩 프로세스 동안, 비디오 디코더 (30) 는 인코딩된 비디오 프레임들 또는 슬라이스들을 나타내는 인코딩된 비디오 비트스트림 및 비디오 인코더 (20) 로부터의 코딩 정보를 나타내는 신택스 엘리먼트들을 수신한다. 엔트로피 디코딩 유닛 (80) 은 TU 내의 양자화된 변환 계수들을 생성하도록 비트스트림들 엔트로피 디코딩한다. 예를 들어, 엔트로피 디코딩 유닛 (80) 은 CABAC, CAVLC 과 같은 컨텍스트 적응적 엔트로피 디코딩 또는 다른 엔트로피 코딩 기법들을 수행할 수도 있다. 엔트로피 디코딩 유닛 (80) 은, 이 개시물에서 설명되는 기법들을 이용하여, 유효 맵, 즉 TU 에서 유효 계수들의 포지션을 나타내는 1들과 0들의 맵을 포함하는 신택스 엘리먼트들을 또한 엔트로피 디코딩한다. 엔트로피 디코딩 유닛 (80) 은 또한 모션 벡터들 및 다른 예측 신택스 엘리먼트들을 엔트로피 디코딩할 수도 있다.

[0090]

컨텍스트 적응적 엔트로피 디코딩을 수행하기 위해, 엔트로피 디코딩 유닛 (80) 은, 예를 들어 이전에 디코딩된 인근 계수들의 값들이 비제로인지의 여부와 관련될 수도 있는, 컨텍스트 모델에 따라서 비트스트림에 나타내지는 각각의 인코딩된 계수의 컨텍스트를 할당한다. 엔트로피 디코딩 유닛 (80) 은 또한 컨텍스트 모델에서 할당된 컨텍스트와 관련되는 인코딩된 계수에 대한 코딩 프로세스를 결정한다. 그 후, 엔트로피 디코딩 유닛 (80) 은 할당된 컨텍스트들에 기초하여 계수들을 엔트로피 디코딩한다. CABAC 의 경우에 있어서, 엔트로피 디코딩 유닛 (80) 은 컨텍스트 모델에서 할당된 컨텍스트와 관련되는 인코딩된 계수의 값 (예컨대, 0 또는 1) 에 대한 확률 추정을 결정한다. 그 후, 엔트로피 디코딩 유닛 (80) 은 계수의 실제 디코딩된 값에 기초하여 컨텍스트 모델에서 할당된 컨텍스트와 관련된 확률 추정을 업데이트한다.

[0091]

통상, 비디오 디코더 (30) 는 구현된 비디오 압축 표준에 의해 지원되는 상이한 사이즈들의 TUs 의 각각에 대한 별도의 컨텍스트 모델들을 유지해야 한다. HEVC 표준에 대해, 추가 변환 유닛 사이즈들, 예컨대 32x32 내지 128x128 은 비디오 코딩 효율을 향상시키도록 제안되었지만, 추가 TU 사이즈들은 또한 추가 변환 유닛 사이즈들 각각에 대한 컨텍스트 모델들을 유지하기 위해 증가한 메모리 및 계산 요건들을 초래한다.

[0092]

이 개시물의 기법들에 따르면, 엔트로피 디코딩 유닛 (80) 은 상이한 사이즈들을 가지는 변환 유닛들 사이에 공유되는 조인트 컨텍스트 모델을 사용하여 비디오 계수들의 엔트로피 디코딩을 수행하도록 구성될 수도 있다. 이 기법들은 주로 CABAC 의 엔트로피 디코딩 기법에 대해 설명된다. 그러나, 몇몇 경우들에 있어서, 이 기법들은 또한 다른 컨텍스트 적응적 엔트로피 코딩 기법들에 적용될 수도 있다. 2 개 이상의 사이즈들의 변환 유닛들 사이에 조인트 컨텍스트 모델을 공유하면, 비디오 디코더 (30) 상에 컨텍스트들 및 확률들을 저장하는 데 필요한 메모리의 양을 감소시킬 수도 있다. 또한, 조인트 컨텍스트 모델을 공유하는 것은, 비디오 슬라이스의 시작부에서 모든 컨텍스트 모델들을 리셋하는 것을 포함하는, 비디오 디코더 (30) 상에 컨텍스트 모델들을 유지하는 계산 비용들을 역시 감소시킬 수도 있다. CABAC 의 경우에 있어서, 이 기법들은 또한 계수들의 실제 코딩된 값들에 기초하여 컨텍스트 모델들의 확률 추정들을 계속해서 업데이트하는 계산 비용들을 역시 감소시킬 수도 있다.

[0093]

이 개시물에 따르면, 엔트로피 디코딩 유닛 (80) 은 상이한 사이즈들을 가지는 변환 유닛들 사이에 공유되는 조인트 컨텍스트 모델을 유지할 수도 있고, 조인트 컨텍스트 모델에 따라서 변환 유닛들 중 하나와 관련된 계수들에 대한 컨텍스트들을 선택할 수도 있다. 그 후, 엔트로피 디코딩 유닛 (80) 은 선택된 컨텍스트들에 기초하여 변환 유닛들과 관련된 유효 계수들을 엔트로피 디코딩할 수도 있다. 일 예로서, 조인트 컨텍스트 모델은 32x32 의 제 1 사이즈를 가지는 변환 유닛들과 16x16 의 제 2 사이즈를 가지는 변환 유닛들 사이에 공유될 수도 있다. 몇몇 경우들에 있어서는, 2 개 초과 사이즈들의 변환 유닛들이 동일한 조인트 컨텍스트 모델을 공유할 수도 있다. 일 예에서, 조인트 컨텍스트 모델은 변환 유닛들에 대한 유효 맵들의 조인트 컨텍스트 모델일 수도 있다. 다른 예들에서, 조인트 컨텍스트 모델은 다른 코딩 정보 또는 신택스 엘리먼트들과 관련될 수도 있다. CABAC 디코딩 프로세스는 도 6 과 관련하여 더 상세히 설명된다.

- [0094] 일 예에서, 엔트로피 디코딩 유닛 (80) 은 리테인된 계수 블록을 생성하도록 제로화된 계수들과 제 1 사이즈를 가지는 TU 및 본래의 제 2 사이즈를 가지는 TU 에 의해 공유되는 조인트 컨텍스트 모델을 유지할 수도 있다. 몇몇 경우들에 있어서, 리테인된 계수 블록은 제 2 사이즈와 동일한 사이즈를 가질 수도 있다. 예를 들어, 이 경우에 있어서, 엔트로피 디코딩 유닛 (80) 은 조인트 컨텍스트 모델에 따라 본래 제 1 사이즈를 가지는 TU 내에서 제 2 사이즈를 가지는 리테인된 계수 블록의 계수들에 대한 컨텍스트들을 선택할 수도 있다. 그 후, 엔트로피 디코딩 유닛 (80) 은 선택된 컨텍스트들에 기초하여, 인코딩된 유효 계수들을, 제 1 사이즈를 가지는 TU 내의 리테인된 계수 블록으로 산술적으로 디코딩한다. CABAC 의 경우에 있어서, 엔트로피 디코딩 유닛 (80) 은 또한 계수들의 실제 디코딩된 값들에 기초하여 조인트 컨텍스트 모델에서 선택된 컨텍스트들과 관련된 확률 추정들을 업데이트한다.
- [0095] 다른 예에서, 엔트로피 디코딩 유닛 (80) 은 제 1 사이즈를 가지는 제 1 TU 및 제 2 사이즈를 가지는 제 2 TU 에 의해 공유되는 조인트 컨텍스트 모델을 유지할 수도 있다. 이 예에서, 엔트로피 디코딩 유닛 (80) 은 조인트 컨텍스트 모델에 따라 제 1 및 제 2 사이즈 중 하나의 TU 와 관련되는 인코딩된 계수들에 대한 컨텍스트들을 선택할 수도 있다. 그 후, 엔트로피 디코딩 유닛 (80) 은 선택된 컨텍스트들에 기초하여, 인코딩된 유효 계수들을 TU 로 엔트로피 디코딩한다. CABAC 의 경우에 있어서, 그 후, 엔트로피 디코딩 유닛 (80) 은 계수들의 실제 디코딩된 값들에 기초하여 조인트 컨텍스트 모델에서 선택된 컨텍스트들과 관련된 확률 추정들을 업데이트할 수도 있다. 어느 경우이든, 엔트로피 디코딩 유닛 (80) 은 제 1 사이즈 또는 제 2 사이즈 중 어느 하나의 TUs 내의 디코딩된 양자화된 변환 계수들을 역양자화 유닛 (86) 에 포워드한다.
- [0096] 역양자화 유닛 (86) 은, 전술한 바와 같이, 엔트로피 디코딩 유닛 (80) 에 의해 TUs 로 디코딩된 양자화된 변환 계수들을 역양자화, 즉 양자화 해제한다. 역양자화 프로세스는, 양자화 정도를 결정하도록 각각의 비디오 블록 또는 CU 에 대해 비디오 인코더 (20) 에 의해 계산된 양자화 파라미터 QP 의 사용을 포함할 수도 있고, 마찬가지로, 적용되어야 하는 역양자화의 정도를 포함할 수도 있다. 역변환 유닛 (88) 은, 픽셀 도메인에 잔여 비디오 데이터를 생성하기 위해, TUs 내의 변환 계수들에, 역변환, 예컨대, 역DCT, 역정수 변환, 역웨이블릿 변환, 또는 개념적으로 유사한 역변환 프로세스를 적용한다.
- [0097] 몇몇 예들에서, 역변환 유닛 (88) 은 2 차원 개별 변환을 포함할 수도 있다. 역변환 유닛 (88) 은, 먼저 TU 내의 변환 계수들의 로우들에 1 차원 변환을 적용하고, 그 후, TU 내의 변환 계수들의 컬럼들에 1 차원 역변환을 적용함으로써 TU에서 변환 계수들에 대해 제 1 방향으로 1 차원 역변환을 적용함으로써, TU 에 2 차원 변환을 적용할 수도 있으며, 그 반대 순서로 적용할 수도 있다. TU 의 변환 계수들에 제 1 방향으로 1 차원 역변환을 적용한 후, 비디오 디코더 (30) 는 제 2 방향으로 1 차원 역변환을 적용하기 위한 중간 잔여 데이터를 버퍼링한다. 전술한 바와 같이, HEVC 표준에서, 더 큰 변환 유닛 사이즈들, 예컨대 32x32 내지 128x128 은 비디오 코딩 효율을 향상시키도록 제안되었다. 그러나, 더 큰 TU 사이즈들은 2 차원 변환들에 대한 증가한 중간 버퍼링 요건들을 역시 초래할 것이다.
- [0098] 더 큰 TU 사이즈들에 대한 중간 버퍼링 요건들을 감소시키기 위해, 이 개시물에 설명된 기법들은 도 2 로부터의 비디오 인코더 (20) 를 사용하여 TU 에 포함된 고주파수의 변환 계수 서브셋을 제로화하는 것을 포함할 수도 있다. TU 내의 제로화된 변환 계수들은 간단히 제로와 동일하게 설정되며, 저장하거나 변환하거나 인코딩할 어떠한 값도 갖지 않는다. 따라서, 엔트로피 디코딩 유닛 (80) 은, 본래의 제 1 사이즈를 가지는 TU 내에서 제 2 사이즈를 가지는 리테인된 계수 블록과 관련되는 인코딩된 계수들을 나타내는 인코딩된 비트스트림을 수신한다. 엔트로피 디코딩 유닛 (80) 은, 계수들을, 제 1 사이즈를 가지는 TU 내의 리테인된 계수 블록으로 디코딩한다. 그 후, TU 는 제 2 사이즈의 리테인된 계수 블록 내의 계수들 및 TU 내의 나머지 계수들을 나타내는 제로들을 포함한다.
- [0099] 이 방식으로, 비디오 인코더 (20) 에서 변환 계수들을 제로화하는 프로세스는 또한 비디오 디코더 (30) 에서 역변환을 수행할 때 더 큰 TU 사이즈들에 대한 중간 버퍼링 요건들을 역시 감소시킬 수도 있다. 일 예로서, 역변환 유닛 (88) 은 제 1 방향, 예컨대 로우 방향의 1 차원 역변환을, 사이즈 32x32 를 가지는 TU 내에서 사이즈 16x16 을 가지는 리테인된 계수 블록의 변환 계수들에 적용할 수도 있다. 로우 역변환 후, 비디오 디코더 (30) 는, TU 의 절반, 즉 32x16 개의 계수들만을 포함하는, 리테인된 계수 블록 내의 계수들로부터 변환된 중간 잔여 데이터만을 버퍼링하면 된다. 역변환 유닛 (88) 은 제 2 방향, 예컨대 컬럼 방향의 1 차원 역변환을, TU 내의 중간 잔여 데이터에 적용할 수도 있다. 이 방식으로, 역변환 유닛 (88) 은, 사이즈 16x16 을 가지는 리테인된 계수 블록에 잔여 데이터를 포함하고 TU 에 나머지 잔여 데이터를 나타내도록 제로들을 추가함으로써 본래 사이즈 32x32 의 TU 를 생성할 수도 있다.

- [0100] 엔트로피 디코딩 유닛 (80) 은 디코딩된 모션 벡터들 및 다른 예측 선택스 엘리먼트들을 예측 유닛 (81) 에 포워드한다. 비디오 디코더 (30) 는 비디오 예측 유닛 레벨, 비디오 코딩 유닛 레벨, 비디오 슬라이스 레벨, 비디오 프레임 레벨, 및/또는 비디오 시퀀스 레벨에서 선택스 엘리먼트들을 수신할 수도 있다. 비디오 프레임이 인트라-코딩된 프레임으로서 코딩될 때, 예측 유닛 (81) 의 인트라 예측 유닛 (84) 은 현재 프레임의 이전 디코딩된 블록들로부터의 데이터에 기초하여 현재 비디오 프레임의 비디오 블록들에 대한 예측 데이터를 생성한다. 비디오 프레임이 인터-코딩된 프레임으로서 코딩될 때, 예측 유닛 (81) 의 모션 보상 유닛 (82) 는 엔트로피 디코딩 유닛 (80) 으로부터 수신되는 디코딩된 모션 벡터들에 기초하여 현재 비디오 프레임의 비디오 블록들에 대한 예측 블록들을 생성한다. 예측 블록들은 기준 프레임 메모리 (92) 에 저장된 기준 프레임의 하나 이상의 기준 블록들에 대해 생성될 수도 있다.
- [0101] 모션 보상 유닛 (82) 은 모션 벡터들 및 다른 예측 선택스를 파싱함으로써 디코딩되는 비디오 블록에 대한 예측 정보를 결정하고, 디코딩되는 현재 비디오 블록에 대한 예측 블록들을 생성한다. 예를 들어, 모션 보상 유닛 (82) 은 수신된 선택스 엘리먼트들 중 일부를 사용하여, 현재 프레임을 인코딩하는 데 사용되는 CUs 의 사이즈들, 프레임의 각각의 CU 가 분할되는 방법을 설명하는 분할 정보, 각각의 분할이 인코딩되는 방식을 나타내는 모드들 (예컨대, 인트라- 또는 인터-예측), 및 인터-예측 슬라이스 유형 (예컨대, B 슬라이스, P 슬라이스, 또는 GPB 슬라이스), 기준 프레임 리스트 구성 커맨드들, 기준 프레임들에 적용되는 보간 필터들, 프레임의 각각의 비디오 블록에 대한 모션 벡터들, 모션 벡터들과 관련된 비디오 파라미터 값들, 및 현재 비디오 프레임을 디코딩할 다른 정보를 결정한다.
- [0102] 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. 합산 유닛 (90) 은 이 합산 연산을 수행하는 컴포넌트 또는 컴포넌트들일 수도 있다. 원한다면, 디블로킹 필터는 블로킹 아티팩트들을 제거하기 위해, 디코딩된 블록들을 필터링하도록 적용될 수도 있다. 그 후, 디코딩된 비디오 블록들은, 후속 모션 보상에 대한 기준 프레임들의 기준 블록들을 제공하는, 기준 프레임 메모리 (92) 의 기준 프레임들에 저장된다. 기준 프레임 메모리 (92) 는 또한 도 1 의 디스플레이 디바이스 (32) 와 같은 디스플레이 디바이스 상에서의 프레젠테이션을 위한 디코딩된 비디오를 생성한다.
- [0103] 도 5 는 조인트 컨텍스트 모델에 따라 비디오 계수들에 대한 컨텍스트들을 선택하도록 구성된 도 2 로부터의 엔트로피 인코딩 유닛 (56) 의 일 예를 예시한 블록도이다. 엔트로피 인코딩 유닛 (56) 은 컨텍스트 모델링 유닛 (94), 산술 인코딩 유닛 (96), 및 조인트 컨텍스트 모델 저장소 (98) 를 포함한다. 전술한 바와 같이, 이 개시물의 기법들은 상이한 사이즈들을 가지는 변환 유닛들 사이에 공유되는 조인트 컨텍스트 모델을 사용하여 비디오 계수들의 엔트로피 인코딩을 수행하는 것에 관한 것이다. 구체적으로, 이 기법들은 여기서 CABAC 인코딩 프로세스에 대해 설명된다.
- [0104] 엔트로피 인코딩 유닛 (56) 은 조인트 컨텍스트 모델 저장소 (98) 내에서 상이한 사이즈들을 가지는 변환 유닛들 사이에 공유되는 조인트 컨텍스트 모델을 유지한다. 일 예로서, 조인트 컨텍스트 모델 저장소 (98) 는 제 2 사이즈를 가지는 리테인된 계수 블록을 생성하도록 제로화된 계수들과 제 1 사이즈를 가지는 TU 및 본래의 제 2 사이즈를 가지는 TU 에 의해 공유되는 조인트 컨텍스트 모델을 저장할 수도 있다. 다른 예로서, 조인트 컨텍스트 모델 저장소 (98) 는 제 1 사이즈를 가지는 TU 및 제 2 사이즈를 가지는 TU 에 의해 공유되는 조인트 컨텍스트 모델을 정의할 수도 있다. 몇몇 경우들에 있어서, 제 1 사이즈는 32x32 를 포함할 수도 있고, 제 2 사이즈는 16x16 를 포함할 수도 있다.
- [0105] 컨텍스트 모델링 유닛 (94) 은 엔트로피 인코딩을 위한 벡터로 스캐닝된 제 1 사이즈 또는 제 2 사이즈 중 어느 하나의 TU 와 관련된 변환 계수들을 수신한다. 그 후, 컨텍스트 모델링 유닛 (94) 은 조인트 컨텍스트 모델에 따라 TU 의 이전에 인코딩된 인근 계수들의 값들에 기초하여 TU 의 계수들 각각에 컨텍스트를 할당한다. 더 구체적으로, 컨텍스트 모델링 유닛 (94) 은 이전에 인코딩된 인근 계수들의 값들이 비제로인지의 여부에 따라서 컨텍스트들을 할당할 수도 있다. 할당된 컨텍스트는 조인트 컨텍스트 모델에서의 컨텍스트 인덱스, 예컨대 context(i) (여기서, i=0, 1, 2, ..., N) 를 지칭할 수도 있다.
- [0106] 컨텍스트가 계수에 할당된 후, 컨텍스트 모델링 유닛 (94) 은 조인트 컨텍스트 모델에서 할당된 컨텍스트와 관련된 계수의 값 (예컨대, 0 또는 1) 에 대한 확률 추정을 결정할 수도 있다. 각각의 상이한 컨텍스트 인덱스는 그 컨텍스트 인덱스를 가지는 계수의 값에 대한 확률 추정과 관련된다. 컨텍스트 모델링 유닛 (94) 에 의해 CABAC 에 대해 수행된 확률 추정은 유한 상태 기계 (FSM) 를 사용하는 테이블 구동 추정기에 기초할 수도 있다. 각각의 컨텍스트에 대해, FSM 은 소정의 계수가 0 또는 1 의 값을 가지는 최상의 확률 추정으로서 현

제 상태를 제공하고 과거 컨텍스트 값을 트래킹함으로써 관련 확률 추정을 유지한다. 예를 들어, 확률 상태가 0 내지 127 의 범위에 있다면, 상태 0 은 0 의 값을 가지는 계수의 확률이 0.9999 라는 것을 의미할 수도 있고, 상태 127 은 0 의 값을 가지는 계수의 확률이 0.0001 이라는 것을 의미할 수도 있다.

[0107] 산술 인코딩 유닛 (96) 은 할당된 컨텍스트와 관련되는 결정된 계수의 확률 추정에 기초하여 계수를 산술적으로 인코딩한다. 이 방식으로, 산술 인코딩 유닛 (96) 은 조인트 컨텍스트 모델에 따라 제 1 사이즈 또는 제 2 사이즈 중 어느 하나의 TU 와 관련된 산술적으로 인코딩된 계수들을 나타내는 인코딩된 비트스트림을 생성한다.

[0108] 인코딩 후, 산술 인코딩 유닛 (96) 은 계수의 실제 인코딩된 값을 컨텍스트 모델링 유닛 (94) 에 역 공급하여, 조인트 컨텍스트 모델 저장소 (98) 내의 조인트 컨텍스트 모델에서 할당된 컨텍스트와 관련된 확률 추정을 업데이트한다. 컨텍스트 모델링 유닛 (94) 은 확률 상태들 사이에서 천이함으로써 조인트 컨텍스트 모델에서 할당된 컨텍스트에 대한 확률 업데이트를 수행한다. 예를 들어, 계수의 실제 코딩된 값이 0 이면, 더 낮은 상태로 천이함으로써, 계수 값이 0 과 동일할 확률이 증가할 수도 있다. 계수들의 실제 인코딩된 값들을 반영하도록 조인트 컨텍스트 모델의 확률 추정들을 계속해서 업데이트함으로써, 조인트 컨텍스트 모델에서 동일한 컨텍스트들에 할당된 장래 계수들에 대한 확률 추정들은 더 정확할 수도 있고, 산술 인코딩 유닛 (96) 에 의해 더욱 감소한 비트 인코딩을 초래할 수도 있다.

[0109] 제 1 예에서, 조인트 컨텍스트 모델 저장소 (98) 는 제 2 사이즈를 가지는 리테인된 계수 블록을 생성하도록 제로화된 계수들과 제 1 사이즈를 가지는 TU 및 본래의 제 2 사이즈를 가지는 TU 에 의해 공유되는 조인트 컨텍스트 모델을 저장할 수도 있다. 예를 들어, 조인트 컨텍스트 모델 저장소 (98) 는, 본래 사이즈 32x32를 가지는 TUs 및 본래 사이즈 16x16 을 가지는 TUs 내에서 사이즈 16x16을 가지는 리테인된 계수 블록들 사이에 공유되는 조인트 컨텍스트 모델을 저장할 수도 있다.

[0110] 사이즈 32x32 의 제 1 TU 내에서 사이즈 16x16 의 리테인된 계수 블록 내의 제 1 계수는 사이즈 16x16 의 리테인된 계수 블록을 생성하도록 제로화된 계수들을 가지는 사이즈 32x32 의 TU 및 본래 사이즈 16x16 의 TU 에 의해 공유되는 조인트 컨텍스트 모델에 컨텍스트(5) 를 할당받을 수도 있다. 그 후, 컨텍스트 모델링 유닛 (94) 은 조인트 컨텍스트 모델에서 할당된 컨텍스트(5) 와 관련된 제 1 계수의 값에 대한 확률 추정을 결정하고, 제 1 계수의 실제 인코딩된 값에 기초하여 조인트 컨텍스트 모델에서 할당된 컨텍스트(5) 와 관련된 확률 추정을 업데이트한다. 사이즈 16x16 의 제 2 TU 내의 제 2 계수는 또한 조인트 컨텍스트 모델에서 제 1 TU 내의 리테인된 계수 블록의 계수와 동일한 컨텍스트(5) 를 할당받을 수도 있다. 그 후, 컨텍스트 모델링 유닛 (94) 은 조인트 컨텍스트 모델에서 동일한 할당된 컨텍스트(5) 와 관련된 제 2 계수의 값에 대한 확률 추정을 결정하고, 제 2 계수의 실제 인코딩된 값에 기초하여 조인트 컨텍스트 모델에서 동일한 할당된 컨텍스트(5) 와 관련된 확률 추정을 업데이트한다.

[0111] 제 2 예에서, 조인트 컨텍스트 모델 저장소 (98) 는 제 1 사이즈를 가지는 TU 및 제 2 사이즈를 가지는 TU 에 의해 공유되는 조인트 컨텍스트 모델을 저장할 수도 있다. 예를 들어, 조인트 컨텍스트 모델 저장소 (98) 는 사이즈들 32x32 및 16x16을 가지는 TUs 에 의해 공유되는 조인트 컨텍스트 모델을 저장할 수도 있다. 사이즈 32x32 의 제 1 TU 내의 제 1 계수는 사이즈 32x32 및 16x16 의 TUs 에 의해 공유되는 조인트 컨텍스트 모델에서 컨텍스트(5) 를 할당받을 수도 있다. 그 후, 컨텍스트 모델링 유닛 (94) 은 조인트 컨텍스트 모델에서 할당된 컨텍스트(5) 와 관련된 제 1 계수의 값에 대한 확률 추정을 결정하고, 제 1 계수의 실제 인코딩된 값에 기초하여 조인트 컨텍스트 모델에서 할당된 컨텍스트(5) 와 관련된 확률 추정을 업데이트한다. 사이즈 16x16 의 제 2 TU 내의 제 2 계수는 또한 조인트 컨텍스트 모델에서 제 1 TU 의 계수와 동일한 컨텍스트(5) 를 할당받을 수도 있다. 그 후, 컨텍스트 모델링 유닛 (94) 은, 제 1 TU 의 계수와 동일한 컨텍스트(5) 와 관련된 조인트 컨텍스트 모델에서 제 2 계수의 값에 대한 확률 추정을 결정하고, 제 2 계수의 실제 인코딩된 값에 기초하여 조인트 컨텍스트 모델에서 동일한 할당된 컨텍스트(5) 와 관련된 확률 추정을 업데이트한다.

[0112] 도 6 은 조인트 컨텍스트 모델에 따라서 비디오 계수들에 대한 컨텍스트들을 선택하도록 구성된 엔트로피 디코딩 유닛 (80) 의 예를 예시한 블록도이다. 엔트로피 디코딩 유닛 (80) 은 산술 디코딩 유닛 (102), 컨텍스트 모델링 유닛 (104) 및 조인트 컨텍스트 모델 저장소 (106) 를 포함한다. 전술한 바와 같이, 상이한 사이즈들을 가지는 변환 유닛들 사이에 공유되는 조인트 컨텍스트 모델을 사용하여 비디오 계수들의 엔트로피 디코딩을 수행하는 것에 관한 것이다. 구체적으로, 이 기법들은 여기서 CABAC 디코딩 프로세스에 대해 설명된다. 엔트로피 디코딩 유닛 (80) 은 도 5 로부터의 엔트로피 인코딩 유닛 (56) 의 것과 본질적으로 대칭인 방식으로 동작할 수도 있다.

- [0113] 엔트로피 디코딩 유닛 (80) 은 조인트 콘텍스트 모델 저장소 (106) 내의 상이한 사이즈들을 가지는 변환 유닛들 사이에 공유되는 조인트 콘텍스트 모델을 유지한다. 조인트 콘텍스트 모델 저장소 (106) 내에 저장된 조인트 콘텍스트 모델은 도 5로부터 엔트로피 인코딩 유닛 (56) 에서의 조인트 콘텍스트 모델 저장소 (98) 에 저장된 조인트 콘텍스트 모델과 실질적으로 유사하다. 일 예로서, 조인트 콘텍스트 모델 저장소 (106) 는 제 2 사이즈를 가지는 리테인된 계수 블록을 생성하도록 제로화된 계수들과 제 1 사이즈를 가지는 TU 및 본래 제 2 사이즈를 가지는 TU 에 의해 공유되는 조인트 콘텍스트 모델을 저장할 수도 있다. 다른 예로서, 조인트 콘텍스트 모델 저장소 (106) 는 제 1 사이즈를 가지는 TU 및 제 2 사이즈를 가지는 TU 에 의해 공유되는 조인트 콘텍스트 모델을 저장할 수도 있다. 몇몇 경우들에 있어서, 제 1 사이즈는 32x32 를 포함할 수도 있고, 제 2 사이즈는 16x16 을 포함할 수도 있다.
- [0114] 산술 디코딩 유닛 (102) 은 제 1 사이즈 또는 제 2 사이즈 중 어느 하나의 TU 와 관련되는 인코딩된 변환 계수들을 나타내는 인코딩된 비트스트림을 수신한다. 산술 디코딩 유닛 (102) 은 비트스트림에 포함된 제 1 계수를 디코딩한다. 그 후, 콘텍스트 모델링 유닛 (104) 은 제 1 디코딩된 계수의 값에 기초하여 비트스트림에 포함된 후속 인코딩된 계수에 콘텍스트를 할당한다. 유사한 방식으로, 콘텍스트 모델링 유닛 (104) 은 조인트 콘텍스트 모델에 따라서 TU 의 이전에 디코딩된 인근 계수들의 값들에 기초하여 비트스트림에 포함된 인코딩된 계수들의 각각에 콘텍스트를 할당한다. 더 구체적으로, 콘텍스트 모델링 유닛 (104) 은 이전에 디코딩된 인근 계수들의 값들이 비제로인지의 여부에 따라서 콘텍스트들을 할당할 수도 있다. 할당된 콘텍스트는 조인트 콘텍스트 모델에서의 콘텍스트 인덱스라고 지칭될 수도 있다.
- [0115] 콘텍스트가 인코딩된 계수에 할당된 후, 콘텍스트 모델링 유닛 (104) 은 조인트 콘텍스트 모델에서 할당된 콘텍스트와 관련되는 인코딩된 계수의 값 (예컨대, 0 또는 1) 에 대한 확률 추정을 결정할 수도 있다. 각각의 상이한 콘텍스트 인덱스는 확률 추정과 관련된다. 콘텍스트 모델링 유닛 (104) 은 인코딩된 계수의 결정된 확률 추정을 산술 디코딩 유닛 (102) 에 공급한다. 그 후, 산술 디코딩 유닛 (102) 은 할당된 콘텍스트와 관련된 계수의 결정된 확률 추정에 기초하여, 인코딩된 계수를 산술적으로 디코딩한다. 이 방식으로, 산술 디코딩 유닛 (102) 은 조인트 콘텍스트 모델에 따라서 제 1 사이즈 또는 제 2 사이즈 중 어느 하나의 TU 내의 디코딩된 변환 계수들을 생성한다.
- [0116] 디코딩 후, 산술 디코딩 유닛 (102) 은 계수의 실제 디코딩된 값을 콘텍스트 모델링 유닛 (104) 에 공급되어, 조인트 콘텍스트 모델 저장소 (106) 내의 조인트 콘텍스트 모델에서 할당된 콘텍스트와 관련된 확률 추정을 업데이트한다. 조인트 콘텍스트 모델의 확률 추정을 계속해서 업데이트하여 계수들의 시계 디코딩된 값들을 반영함으로써, 조인트 콘텍스트 모델에서 동일한 콘텍스트들에 할당된 미래 계수들에 대한 확률 추정들은 더 정확할 수도 있고, 산술 디코딩 유닛 (102) 에 의해 더 감소한 비트 디코딩을 초래할 수도 있다.
- [0117] 제 1 예에서, 조인트 콘텍스트 모델 저장소 (106) 는 제 2 사이즈를 가지는 리테인된 계수 블록을 생성하도록 제로화된 계수들과 제 1 사이즈를 가지는 TU 및 본래 제 2 사이즈를 가지는 TU 에 의해 공유되는 조인트 콘텍스트 모델을 저장할 수도 있다. 예를 들어, 조인트 콘텍스트 모델 저장소 (106) 는, 본래 사이즈 32x32 를 가지는 TU 내의 사이즈 16x16을 가지는 리테인된 계수 블록 및 본래 사이즈 16x16을 가지는 TU 에 의해 공유되는 조인트 콘텍스트 모델을 저장할 수도 있다.
- [0118] 사이즈 32x32 의 제 1 TU 내의 사이즈 16x16 의 리테인된 계수 블록과 관련된 제 1 인코딩된 계수는 사이즈 16x16를 가지는 리테인된 계수 블록을 생성하도록 제로화된 계수들을 가지는 사이즈 32x32 의 TU 및 사이즈 16x16 의 TU 에 의해 공유되는 조인트 콘텍스트 모델에서 콘텍스트(5) 를 할당받을 수도 있다. 그 후, 콘텍스트 모델링 유닛 (104) 은 조인트 콘텍스트 모델에서 할당된 콘텍스트(5) 와 관련된 제 1 인코딩된 계수의 값에 대한 확률 추정을 결정하고, 제 1 계수의 실제 디코딩된 값에 기초하여 조인트 콘텍스트 모델에서 할당된 콘텍스트(5) 와 관련된 확률 추정을 업데이트한다. 사이즈 16x16 의 제 2 TU 와 관련된 제 2 인코딩된 계수는, 조인트 콘텍스트 모델 내에서, 제 1 TU 내의 리테인된 계수 블록과 관련된 제 1 인코딩된 계수와 동일한 콘텍스트(5) 를 할당받을 수도 있다. 그 후, 콘텍스트 모델링 유닛 (104) 은 조인트 콘텍스트 모델에서 동일한 할당된 콘텍스트(5) 와 관련된 제 2 인코딩된 계수의 값에 대한 확률 추정을 결정하고, 제 2 계수의 실제 디코딩된 값에 기초하여 조인트 콘텍스트 모델에서 동일한 할당된 콘텍스트(5) 와 관련된 확률 추정을 업데이트한다.
- [0119] 제 2 예에서, 조인트 콘텍스트 모델 저장소 (106) 는 제 1 사이즈를 가지는 TU 및 제 2 사이즈를 가지는 TU 에 의해 공유되는 조인트 콘텍스트 모델을 저장할 수도 있다. 예를 들어, 조인트 콘텍스트 모델 저장소 (106) 는 사이즈들 32x32 및 16x16을 가지는 TUs 에 의해 공유되는 조인트 콘텍스트 모델을 저장할 수도 있다. 사

이즈 32x32 의 제 1 TU 와 관련된 제 1 인코딩된 계수는 사이즈 32x32 의 TU 및 사이즈 16x16 의 TU 에 의해 공유되는 조인트 컨텍스트 모델에서 컨텍스트(5) 를 할당받을 수도 있다. 그 후, 컨텍스트 모델링 유닛 (104) 은 조인트 컨텍스트 모델에서 할당된 컨텍스트(5) 와 관련된 제 1 인코딩된 계수의 값에 대한 확률 추정 을 결정하고, 제 1 계수의 실제 디코딩된 값에 기초하여 조인트 컨텍스트 모델에서 할당된 컨텍스트(5) 와 관련된 확률 추정을 업데이트한다. 사이즈 16x16 의 제 2 TU 와 관련된 제 2 인코딩된 계수는, 조인트 컨텍스트 모델에서, 제 1 TU 와 관련된 제 1 인코딩된 계수와 동일한 컨텍스트(5) 를 할당받을 수도 있다. 그 후, 컨텍스트 모델링 유닛 (104) 은 조인트 컨텍스트 모델에서 동일한 할당된 컨텍스트(5) 와 관련된 제 2 인코딩된 계수의 값에 대한 확률 추정을 결정하고, 제 2 계수의 실제 디코딩된 값에 기초하여 조인트 컨텍스트 모델에서 동일한 할당된 컨텍스트(5) 와 관련된 확률 추정을 업데이트한다.

[0120] 도 7 은 제 1 사이즈를 가지는 제 1 변환 유닛과 제 2 사이즈를 가지는 제 2 변환 유닛 사이에 공유되는 조인트 컨텍스트 모델을 사용하여 비디오 계수들을 인코딩 및 디코딩하는 예시적 동작을 예시한 플로우차트이다. 예시된 동작은, 도 2 의 비디오 인코더 (20) 내에 있는 도 5 의 엔트로피 인코딩 유닛 (56) 및 도 4 의 비디오 디코더 (30) 내에 있는 도 6 의 엔트로피 디코딩 유닛 (80) 을 참조하여 설명되며, 다른 디바이스들이 유사한 기법들을 구현할 수도 있다.

[0121] 예시된 동작에서, 비디오 인코더 (20) 내의 엔트로피 인코딩 유닛 (56) 및 비디오 디코더 (30) 내의 엔트로피 디코딩 유닛 (80) 은 제 1 사이즈를 가지는 TU 및 제 2 사이즈를 가지는 TU 에 의해 공유되는 조인트 컨텍스트 모델을 유지할 수도 있다. 이 경우에 있어서, 제 1 사이즈, 예컨대 32x32 를 가지는 제 1 TU 에 포함된 계수들은, 제 1 TU 내의 고주파수의 계수들을 제로화하는 일 없이도, 제 2 사이즈, 예컨대 16x16 의 제 2 TU 에 포함되는 계수들과 유사한 확률 통계를 가질 수도 있다. 이것은, 고주파수의 계수들이, 엔트로피 코딩에 대한 인근 계수들의 확률 통계에 미치는 효과가 무시될 정도로 잔여 비디오 데이터가 거의 없음을 나타낼 때 가능하다.

[0122] 일 경우에 있어서, 비디오 인코더 (20) 는 변환 유닛 (52) 을 사용하여, 잔여 비디오 데이터를, 제 1 사이즈를 가지는 TU 내의 변환 계수들로 변환할 수도 있다 (120). 다른 경우에 있어서, 비디오 인코더 (20) 는 변환 유닛 (52) 을 사용하여, 잔여 비디오 데이터를, 제 2 사이즈를 가지는 TU 내의 변환 계수들로 변환할 수도 있다 (121). TU 가 제 1 사이즈의 것인지 또는 제 2 사이즈의 것인지와는 무관하게, 이 개시물의 기법들은 동일한 조인트 컨텍스트 모델에 따라서 엔트로피 인코딩 유닛 (56) 이 TU 내의 계수들을 엔트로피 인코딩하게 한다. 따라서, 이 기법들은, 비디오 인코더 (20) 상에서 컨텍스트들 및 확률들을 저장하는 데 필요한 메모리의 양을 감소시키고, 비디오 인코더 (20) 상에 컨텍스트 모델들을 유지하는 계산 비용들을 저감한다.

[0123] 엔트로피 인코딩 유닛 (56) 내의 컨텍스트 모델링 유닛 (94) 은 제 1 사이즈 및 제 2 사이즈 양측 모두를 가지는 TUs 에 의해 공유되는 조인트 컨텍스트 모델에 따라서 TU 내의 각각의 계수에 대한 컨텍스트를 선택한다 (122). 더 구체적으로, 컨텍스트 모델링 유닛 (94) 은 조인트 컨텍스트 모델에 따라서 TU 의 이전에 인코딩된 인근 계수들의 값들에 기초하여 TU 의 소정 계수에 컨텍스트를 할당한다. 그 후, 컨텍스트 모델링 유닛 (94) 은 조인트 컨텍스트 모델 저장소 (98) 내의 조인트 컨텍스트 모델에서 할당된 컨텍스트와 관련된 계수의 값 (예컨대, 0 또는 1) 에 대한 확률 추정을 결정할 수도 있다. 산술 인코딩 유닛 (96) 은 다음으로 계수에 대한 선택된 컨텍스트에 기초하여 계수를 산술적으로 인코딩한다 (124).

[0124] 인코딩 후, 산술 인코딩 유닛 (96) 은 계수들의 실제 인코딩된 값들을 컨텍스트 모델링 유닛 (94) 에 공급한다. 그 후, 컨텍스트 모델링 유닛 (94) 은 제 1 사이즈 또는 제 2 사이즈 중 어느 하나의 TU 내의 계수들의 실제 인코딩된 값들에 기초하여 조인트 컨텍스트 모델의 확률 추정을 업데이트할 수도 있다 (126). 비디오 인코더 (20) 는 제 1 사이즈 또는 제 2 사이즈 중 어느 하나의 TU 와 관련되는 인코딩된 계수들을 나타내는 비트스트림을 비디오 디코더(30) 에 송신한다 (128).

[0125] 비디오 디코더 (30) 는 제 1 사이즈 또는 제 2 사이즈 중 어느 하나를 가지는 TU 와 관련되는 인코딩된 계수들을 나타내는 비트스트림을 수신할 수도 있다 (130). TU 가 제 1 사이즈의 것인지 또는 제 2 사이즈의 것인지와는 무관하게, 이 개시물의 기법들은 동일한 조인트 컨텍스트 모델에 기초하여 엔트로피 디코딩 유닛 (80) 이 TU 와 관련된 계수들을 엔트로피 디코딩하게 한다. 따라서, 이 기법들은 비디오 디코더 (30) 상에 컨텍스트들 및 확률들을 저장하는 데 필요한 메모리의 양을 감소시킬 수도 있고, 비디오 디코더 (30) 상에 컨텍스트 모델들을 유지하는 계산 비용들을 저감할 수도 있다.

[0126] 엔트로피 디코딩 유닛 (80) 내의 컨텍스트 모델링 유닛 (104) 은 제 1 사이즈 및 제 2 사이즈 양측 모두를 가지는 TUs 에 의해 공유되는 조인트 컨텍스트 모델에 따라서 TU 와 관련되는 각각의 인코딩된 계수에 대한 컨텍스트

트를 선택한다 (132). 더 구체적으로, 콘텍스트 모델링 유닛 (104)은 조인트 콘텍스트 모델에 따라서 TU의 이전에 디코딩된 인근 계수들의 값들에 기초하여 TU와 관련된 후속 인코딩된 계수에 콘텍스트를 할당할 수도 있다. 그 후, 콘텍스트 모델링 유닛 (104)은 조인트 콘텍스트 모델 저장소 (106)내에 조인트 콘텍스트 모델에 할당된 콘텍스트와 관련되는 인코딩된 계수의 값 (예컨대, 0 또는 1)에 대한 확률 추정을 결정할 수도 있다. 콘텍스트 모델링 유닛 (104)은 인코딩된 계수에 대한 선택된 콘텍스트와 관련되는 결정된 확률 추정을 산술 디코딩 유닛 (102)에 공급한다. 산술 디코딩 유닛 (102)은 다음으로, 선택된 콘텍스트에 기초하여 제 1 사이즈 또는 제 2 사이즈 중 어느 하나의 TU에 인코딩된 계수를 산술적으로 디코딩한다 (134).

[0127]

디코딩 후, 산술 디코딩 유닛 (102)은 계수들의 실제 디코딩된 값들을 콘텍스트 모델링 유닛 (104)에 공급한다. 그 후, 콘텍스트 모델링 유닛 (104)은 제 1 사이즈 또는 제 2 사이즈 중 어느 하나의 TU내의 계수들의 실제 디코딩된 값들에 기초하여 조인트 콘텍스트 모델의 확률 추정을 업데이트할 수도 있다 (136). 일 경우에 있어서, 비디오 디코더 (30)는 역변환 유닛 (88)을 사용하여 제 1 사이즈를 가지는 TU내의 계수들을 잔여 비디오 데이터로 역변환할 수도 있다 (138). 다른 경우에 있어서, 비디오 디코더 (30)는 역변환 유닛 (88)을 사용하여 제 2 사이즈를 가지는 TU내의 계수들을 잔여 비디오 데이터로 역변환할 수도 있다 (139).

[0128]

도 8은 제 2 사이즈를 가지는 리테인된 계수 블록을 생성하도록 제로화된 계수들과 제 1 사이즈를 가지는 제 1 변환 유닛과 제 2 사이즈를 가지는 제 2 변환 유닛 사이에 공유되는 조인트 콘텍스트 모델을 사용하여 비디오 계수들을 엔트로피 인코딩 및 디코딩하는 예시적 동작을 예시한 플로우차트이다. 예시된 동작은 도 2의 비디오 인코더 (20)내의 도 5의 엔트로피 인코딩 유닛 (56) 및 도 4의 비디오 디코더 (30)내의 도 6의 엔트로피 디코딩 유닛 (80)을 참조하여 설명되지만, 다른 디바이스들이 유사한 기법들을 구현할 수도 있다.

[0129]

예시된 동작에서, 비디오 인코더 (20)내의 엔트로피 인코딩 유닛 (56) 및 비디오 디코더 (30)내의 엔트로피 디코딩 유닛 (80)은 제 2 사이즈를 가지는 리테인된 계수 블록을 생성하도록 제로화된 계수와 제 1 사이즈를 가지는 TU 및 제 2 사이즈를 가지는 TU에 의해 공유되는 조인트 콘텍스트 모델을 유지할 수도 있다. 이 경우에 있어서, 제 1 사이즈, 예컨대 32x32를 가지는 제 1 TU내의 제 2 사이즈, 예컨대 16x16을 가지는 리테인된 계수 블록에 포함된 계수들은 제 2 사이즈, 예컨대 16x16의 제 2 TU에 포함된 계수들과 유사한 확률 통계를 가질 수도 있다.

[0130]

일 경우에 있어서, 비디오 인코더 (20)는 변환 유닛 (52)을 사용하여, 잔여 비디오 데이터를, 제 1 사이즈를 가지는 TU내의 변환 계수들로 변환할 수도 있다 (140). 비디오 인코더 (20)는, 제 1 TU내의 제 2 사이즈를 가지는 리테인된 계수 블록을 생성하는 변환 후, 제 1 TU에 포함된 계수들의 서브세트를 제로화한다 (141). 제로화된 변환 계수들의 서브세트는 일반적으로 리테인된 계수블록내의 계수들에 대해 고주파수의 변환 계수들을 포함한다. 고주파수의 변환 계수들은, 값들을 제로와 동일하게 설정하는 것이, 디코딩된 비디오 품질에 대한 무시할만한 효과를 가질 정도로 잔여 비디오 데이터를 거의 포함하지 않을 수도 있다. 다른 경우에 있어서, 비디오 인코더 (20)는 변환 유닛 (52)을 사용하여, 잔여 비디오 데이터를, 제 2 사이즈를 가지는 TU내의 변환 계수들로 변환할 수도 있다 (142).

[0131]

TU가 본래 제 1 사이즈의 것인지 또는 제 2 사이즈의 것인지와는 무관하게, 이 개시물의 기법들은, 엔트로피 인코딩 유닛 (56)이, 동일한 조인트 콘텍스트 모델에 따라서 TU내의 리테인된 계수들을 엔트로피 인코딩하게 한다. 따라서, 이 기법들은, 비디오 인코더 (20)상에 콘텍스트들 및 확률들을 저장하는 데 필요한 메모리의 양을 감소시키고, 비디오 인코더 (20)상의 콘텍스트 모델들을 유지하는 계산 비용들을 저감한다.

[0132]

엔트로피 인코딩 유닛 (56)내의 콘텍스트 모델링 유닛 (94)은 제 2 사이즈로 제로화된 계수들과 제 1 사이즈를 가지는 TU 및 제 2 사이즈를 가지는 TU에 의해 공유되는 조인트 콘텍스트 모델에 따라서 TU내의 각각의 리테인된 계수에 대한 콘텍스트를 선택한다 (144). 더 구체적으로, 콘텍스트 모델링 유닛 (94)은 조인트 콘텍스트 모델에 따라서 리테인된 계수 블록의 이전에 인코딩된 인근 계수들의 값들에 기초하여 제 TU의 리테인된 계수 블록내의 소정 계수에 콘텍스트를 할당한다. 그 후, 콘텍스트 모델링 유닛 (94)은 조인트 콘텍스트 모델 저장소 (98)내의 조인트 콘텍스트 모델에서 할당된 콘텍스트와 관련된 계수의 값 (예컨대, 0 또는 1)에 대한 확률 추정을 결정할 수도 있다. 산술 인코딩 유닛 (96)은 다음으로 계수에 대한 선택된 콘텍스트에 기초하여 계수를 산술적으로 인코딩한다 (146).

[0133]

인코딩 후, 산술 인코딩 유닛 (96)은 계수들의 실제 인코딩된 값들을 콘텍스트 모델링 유닛 (94)에 공급한다. 그 후, 콘텍스트 모델링 유닛 (94)은 제 1 사이즈의 TU내의 제 2 사이즈의 리테인된 계수 블록 또는 본래 제 2 사이즈의 TU의 계수들의 실제 인코딩된 값들에 기초하여 조인트 콘텍스트 모델의 확률 추정을 업데이트할 수도 있다 (148). 비디오 인코더 (20)는 제 1 사이즈의 TU내의 리테인된 계수 블록 또는 제 2 사이즈의

TU 와 관련되는 인코딩된 계수들을 나타내는 비트스트림을 비디오 디코더 (30) 에 송신한다 (150).

[0134] 비디오 디코더 (30) 는 제 1 사이즈의 TU 내의 제 2 사이즈의 리테인된 계수 블록 또는 본래 제 2 사이즈의 TU 에 관련되는 인코딩된 계수들을 나타내는 비트스트림을 수신할 수도 있다 (152). TU 가 제 1 사이즈의 것인지 또는 제 2 사이즈의 것인지와는 무관하게, 이 개시물의 기법들은, 엔트로피 디코딩 유닛 (80) 이, 동일한 조인트 컨텍스트 모델에 기초하여 TU 와 관련된 계수들을 엔트로피 디코딩하게 한다. 따라서, 이 기법들은, 비디오 디코더 (30) 상에 컨텍스트들 및 확률들을 저장하는 데 필요한 메모리의 양을 감소시키고, 비디오 디코더 (30) 상에 컨텍스트 모델들을 유지하는 계산 비용들을 저감한다.

[0135] 엔트로피 디코딩 유닛 (80) 내의 컨텍스트 모델링 유닛 (104) 은, 제 2 사이즈를 가지는 리테인된 계수 블록을 생성하도록 제로화된 계수들과 제 1 사이즈를 가지는 TU 및 제 2 사이즈를 가지는 TU 에 의해 공유되는 조인트 컨텍스트 모델에 따라서 TU 와 관련된 각각의 계수에 대한 컨텍스트를 선택한다 (154). 더 구체적으로, 컨텍스트 모델링 유닛 (104) 은 조인트 컨텍스트 모델에 따라서 리테인된 계수 블록의 이전에 디코딩된 인근 계수들의 값들에 기초하여 제 1 TU 의 리테인된 계수 블록과 관련된 후속 인코딩된 계수에 컨텍스트를 할당할 수도 있다. 그 후, 컨텍스트 모델링 유닛 (104) 은 조인트 컨텍스트 모델 저장소 (106) 내의 조인트 컨텍스트 모델에서 할당된 컨텍스트와 관련되는 인코딩된 계수의 값 (예컨대, 0 또는 1) 에 대한 확률 추정을 결정할 수도 있다. 컨텍스트 모델링 유닛 (104) 은 인코딩된 계수에 대한 선택된 컨텍스트와 관련되는 결정된 확률을 산술 디코딩 유닛 (102) 에 공급한다. 산술 디코딩 유닛 (102) 은 다음으로, 인코딩된 계수들, 선택된 컨텍스트에 기초하여 제 1 사이즈의 TU 내의 리테인된 계수 블록 또는 제 2 사이즈의 TU 로 산술적으로 디코딩한다 (156).

[0136] 디코딩 후, 산술 디코딩 유닛 (102) 은 계수들의 실제 디코딩된 값들을 컨텍스트 모델링 유닛 (104) 에 공급한다. 그 후, 컨텍스트 모델링 유닛 (104) 은 제 1 사이즈의 TU 또는 본래 제 2 사이즈인 TU 내의 제 2 사이즈의 리테인된 계수 블록 내의 계수들의 실제 디코딩된 값들에 기초하여 조인트 컨텍스트 모델의 확률 추정을 업데이트할 수도 있다 (158). 일 경우에 있어서, 비디오 디코더(30) 는 역변환 유닛 (88) 을 사용하여, 제 1 사이즈를 가지는 TU 내의 제 2 사이즈를 가지는 리테인된 계수 블록의 계수들을 잔여 비디오 데이터로 역변환할 수도 있다 (160). 이 방식으로, 역변환 유닛 (88) 은, 제 2 사이즈를 가지는 리테인된 계수 블록에 잔여 데이터를 포함하고, TU 의 나머지 잔여 데이터를 나타내는 제로들을 추가함으로써 제 1 사이즈를 가지는 TU 를 생성할 수도 있다. 다른 경우에 있어서, 비디오 디코더 (30) 는 역변환 유닛 (88) 을 사용하여, 제 2 사이즈를 가지는 TU 내의 계수들을 잔여 비디오 데이터로 역변환할 수도 있다 (162).

[0137] 하나 이상의 예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현된다면, 이 기능들은 하나 이상의 명령들 또는 코드로서 컴퓨터 판독가능 매체 상에 저장되거나 이를 통하여 송신될 수 있고, 하드웨어 기반 프로세싱 유닛에 의하여 실행될 수도 있다. 컴퓨터 판독가능 매체는 예를 들어, 통신 프로토콜에 따라, 일 장소로부터 타 장소로의 컴퓨터 프로그램의 전송을 용이하게 하는 임의의 매체를 포함하는 통신 매체, 또는 데이터 저장 매체와 같은 유형의 매체에 대응하는, 컴퓨터 판독가능 저장 매체를 포함할 수도 있다. 이러한 방식으로, 컴퓨터 판독가능 매체는, 일반적으로 (1) 비일시적인 유형의 컴퓨터 판독가능 저장 매체 또는 (2) 신호 또는 반송파와 같은 통신 매체에 대응할 수도 있다. 데이터 저장매체는, 본 개시물에서 설명된 기법들의 구현을 위한 명령들, 코드 및/또는 데이터 구조들을 추출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 이용가능한 매체일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터 판독가능 매체를 포함할 수도 있다.

[0138] 제한이 아닌 일 예로, 이러한 컴퓨터 판독가능 저장 매체는 RAM, ROM, EEPROM, CD-ROM 또는 다른 광 디스크 저장, 자기 디스크 저장, 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 원하는 프로그램 코드를 명령들 또는 데이터 구조들의 형태로 저장하기 위하여 이용될 수 있고 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 접속도 적절하게 컴퓨터 판독가능 매체라고 칭한다. 예를 들어, 명령들이 동축 케이블, 광섬유 케이블, 연선 (twisted pair), 디지털 가입자 회선 (DSL), 또는 적외선, 라디오, 및/또는 마이크로파와 같은 무선 기술들을 이용하여 웹사이트, 서버, 또는 다른 원격 소스로부터 송신된다면, 매체의 정의에는 동축 케이블, 광섬유 케이블, 연선, DSL, 또는 적외선, 라디오, 및 마이크로파와 같은 무선 기술들이 포함된다. 그러나, 컴퓨터 판독가능 저장 매체 및 데이터 저장 매체가, 접속들, 반송파들, 신호들 또는 다른 일시적 매체를 포함하지 않고, 대신에 비일시적인, 유형의 저장 매체와 관련된다는 것이 이해되어야 한다. 본 명세서에서 사용되는 바와 같이, 디스크 (disk) 및 디스크 (disc) 는 콤팩트 디스크 (CD), 레이저 디스크, 광 디스크, 디지털 다기능 디스크 (DVD), 플로피 디스크 및 블루레이 디스크를 포함하며, 여기서 디스크 (disk) 들은 보통 데이터를 자기적으로 재생시키는 한편, 디스크 (disc) 들은 레이저를 이용하여 데이터를

광학적으로 재생시킨다. 상기의 조합들도 컴퓨터 판독가능 매체의 범위 내에 포함되어야 한다.

[0139]

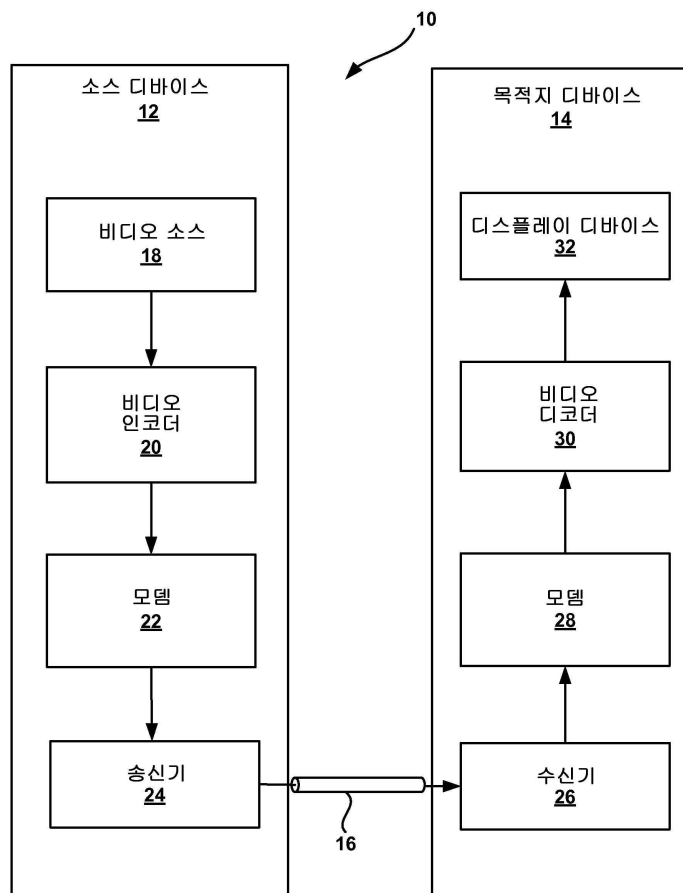
명령들은 하나 이상의 프로세서들, 이를 테면 하나 이상의 디지털 신호 프로세서 (DSP) 들, 범용 마이크로프로세서들, 주문형 집적회로 (ASIC) 들, 필드 프로그램가능 로직 어레이 (FPGA) 들, 또는 다른 등가의 집적 또는 이산 로직 회로에 의해 실행될 수도 있다. 따라서, 본 명세서에서 사용되는 바와 같이, 용어 "프로세서" 는 전술한 구조 또는 본 명세서에서 설명된 기법들의 구현에 적합한 임의의 다른 구조 중 임의의 것을 지칭할 수도 있다. 또한, 일부 양태들에서, 본 명세서에서 설명되는 기능성은 인코딩 및 디코딩을 위하여 구성되는 전용 하드웨어 및/또는 소프트웨어 모듈들 내에 제공되거나, 또는 결합된 코덱 (codec) 내에 통합될 수도 있다. 또한, 이 기술들은 하나 이상의 회로들 또는 로직 엘리먼트들 내에서 온전히 구현될 수 있다.

[0140]

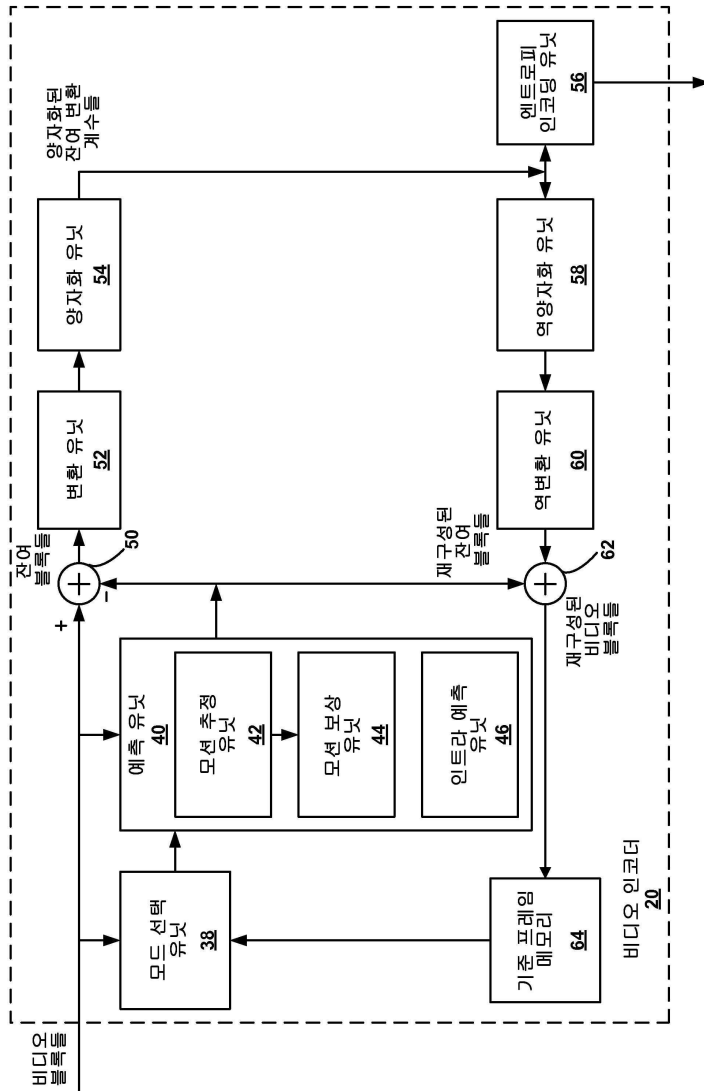
본 개시물의 기법들은, 무선 핸드셋, 집적회로 (IC) 또는 IC들의 세트 (예컨대, 칩 셋) 를 포함하는, 매우 다양한 디바이스들 또는 장치들에서 구현될 수도 있다. 다양한 컴포넌트들, 모듈들, 또는 유닛들은 개시된 기법들을 수행하도록 구성되는 디바이스들의 기능적 양태들을 강조하기 위하여 본 개시물에서 설명되지만, 반드시 상이한 하드웨어 유닛들에 의한 실현을 요구하는 것은 아니다. 오히려, 위에서 설명된 바와 같이, 다양한 유닛들은 코덱 하드웨어 유닛 내에 결합되거나 또는 적합한 소프트웨어 및/또는 펌웨어와 연계하여, 위에서 설명된 바와 같은 하나 이상의 프로세서들을 포함하는, 상호운용적 하드웨어 유닛들의 컬렉션에 의해 제공될 수도 있다.

도면

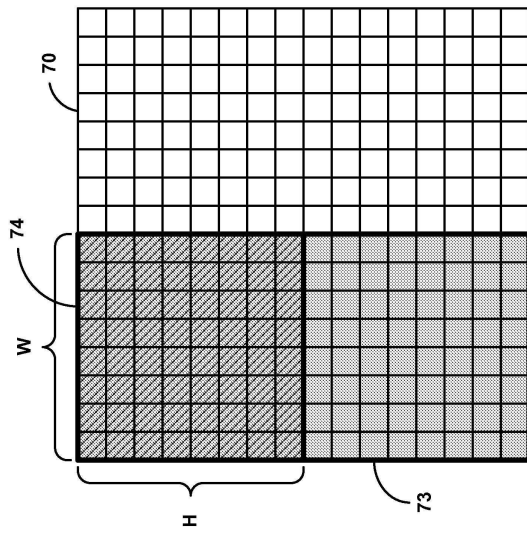
도면1



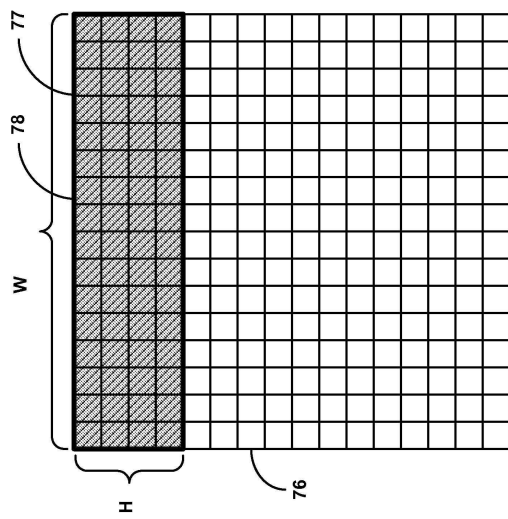
도면2



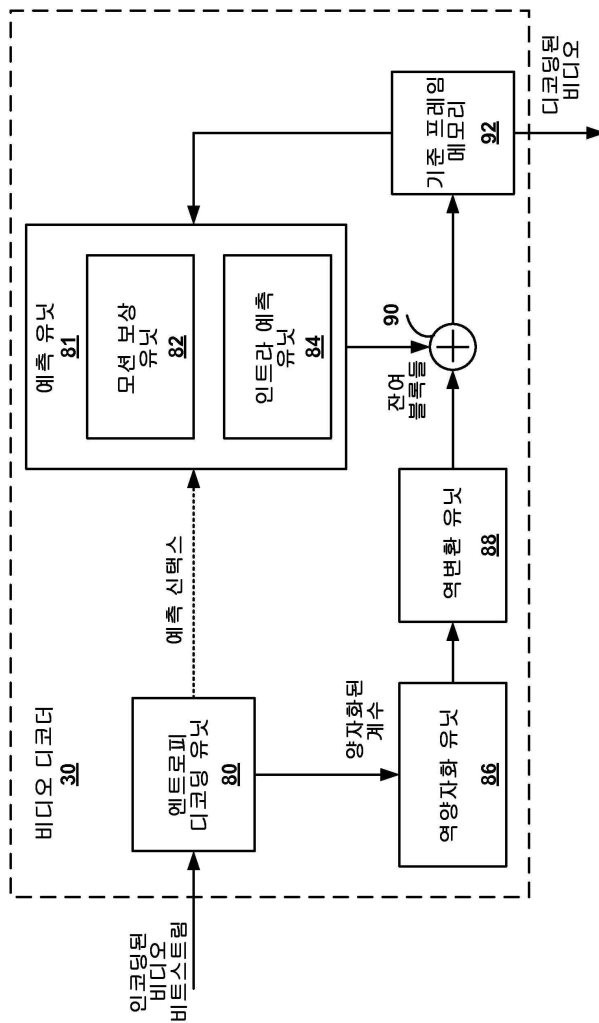
도면3a



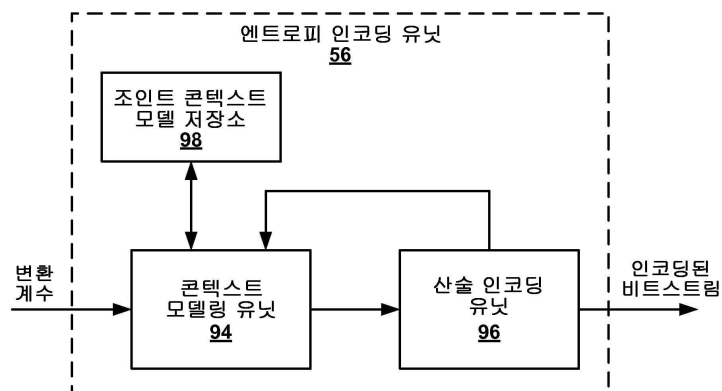
도면3b



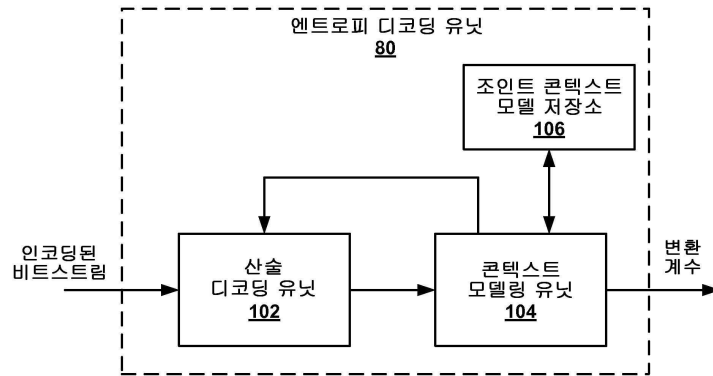
도면4



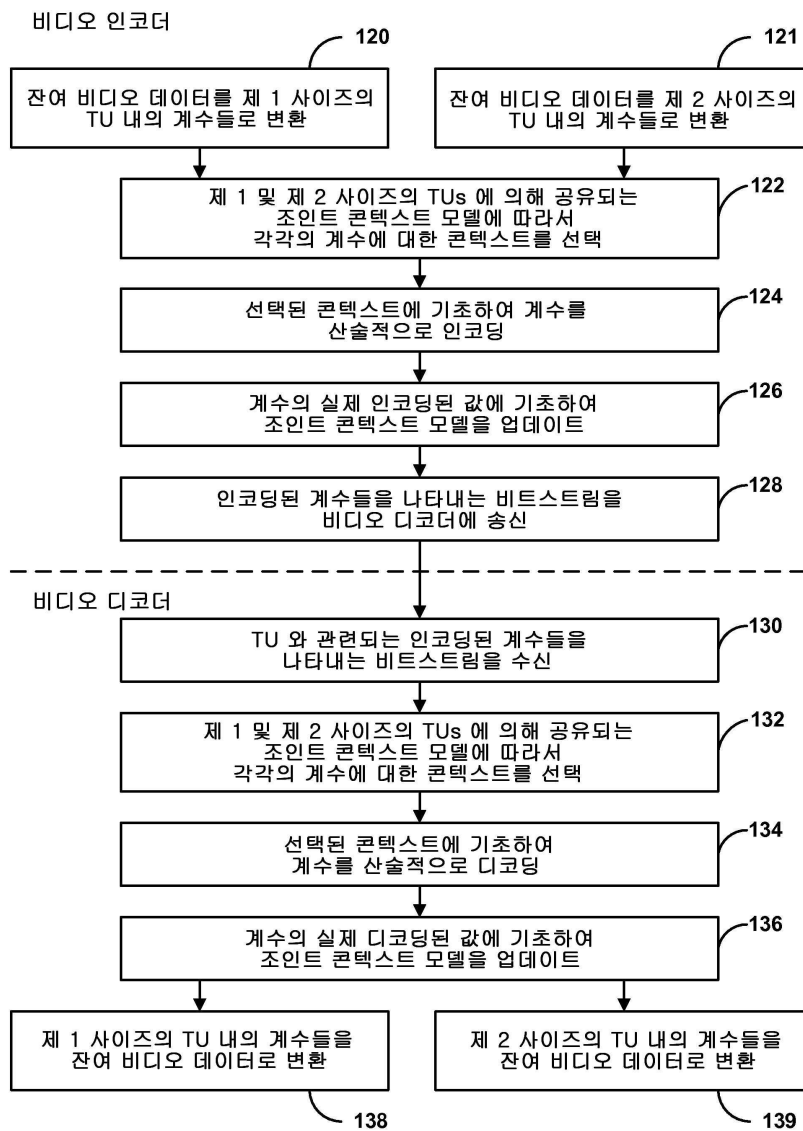
도면5



도면6



도면7



도면8

