



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2023년09월18일
(11) 등록번호 10-2579286
(24) 등록일자 2023년09월12일

- (51) 국제특허분류(Int. Cl.)
 HO4N 19/61 (2014.01) G06F 17/14 (2006.01)
 HO4N 19/12 (2014.01) HO4N 19/13 (2014.01)
 HO4N 19/159 (2014.01) HO4N 19/176 (2014.01)
 HO4N 19/192 (2014.01) HO4N 19/463 (2014.01)
 HO4N 19/70 (2014.01) HO4N 19/91 (2014.01)
 HO4N 19/96 (2014.01)
- (52) CPC특허분류
 HO4N 19/61 (2015.01)
 G06F 17/147 (2013.01)
- (21) 출원번호 10-2020-7034287
- (22) 출원일자(국제) 2019년05월29일
 심사청구일자 2020년11월27일
- (85) 번역문제출일자 2020년11월27일
- (65) 공개번호 10-2021-0003238
- (43) 공개일자 2021년01월11일
- (86) 국제출원번호 PCT/AU2019/050534
- (87) 국제공개번호 WO 2020/000019
 국제공개일자 2020년01월02일
- (30) 우선권주장
 2018204786 2018년06월29일 오스트레일리아(AU)
- (56) 선행기술조사문헌
- (73) 특허권자
 캐논 가부시끼가이샤
 일본 도쿄도 오오따꾸 시모마루쵸 3쵸메 30방 2고
- (72) 발명자
 로즈완 크리스토퍼 제임스
 오스트레일리아 2138 뉴 사우스 웨일즈 콘코드 웨
 스트 빅토리아 애비뉴 유닛 1/22
- (74) 대리인
 장수길, 이중희

(뒷면에 계속)

전체 청구항 수 : 총 12 항

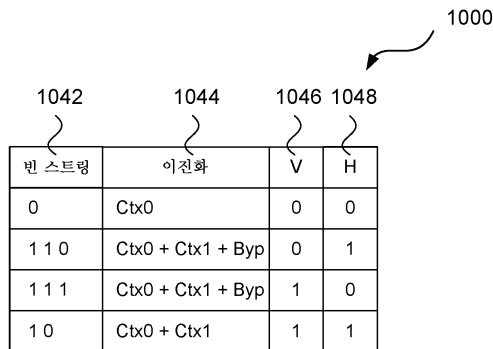
심사관 : 황수진

(54) 발명의 명칭 비디오 샘플들의 변환된 블록을 인코딩 및 디코딩하기 위한 방법, 장치 및 시스템

(57) 요약

본 개시내용은 비트스트림으로부터 이미지 프레임에서의 변환 블록을 디코딩하는 방법(1200)을 제공한다. 방법은 비트스트림으로부터 단일 절단된 단항 이진화를 디코딩하는 단계를 포함하고, 단일 절단된 단항 이진화는 이미지 프레임의 변환 블록의 수평 변환 및 수직 변환에 대해 사용된다. 그후, 방법은 디코딩된 단일 절단된 단항 이진화에 기초하여 수평 및 수직 변환의 타입을 결정하고, 결정된 타입의 수평 및 수직 변환을 이미지의 변환 블록에 적용함으로써 이미지 프레임에서의 변환 블록을 디코딩한다.

대표도 - 도10a



(52) CPC특허분류

HO4N 19/12 (2015.01)
HO4N 19/13 (2015.01)
HO4N 19/159 (2015.01)
HO4N 19/176 (2015.01)
HO4N 19/192 (2015.01)
HO4N 19/463 (2015.01)
HO4N 19/70 (2015.01)
HO4N 19/91 (2015.01)
HO4N 19/96 (2015.01)

(56) 선행기술조사문헌

US20180020218 A1*

Heiko Schwarz et al, Description of Core Experiment 7 (CE 7): Quantization and coefficient coding, JVET of ITU-T and ISO/IEC, 10th meeting, JVET-J1027-v4 (2018.05.18.) 1부.*

Yi-Wen Chen et al, Description of SDR, HDR and 360° video coding technology proposal by Qualcomm and Technicolor - low and high complexity versions, JVET of ITU-T and ISO/IEC, 10th meeting, JVET-J0021 (2018.04.14.) 1부.*

*는 심사관에 의하여 인용된 문헌

명세서

청구범위

청구항 1

비트스트림으로부터 이미지에서의 변환 블록을 디코딩하는 방법으로서:

CABAC(context adaptive binary arithmetic coder)를 사용함으로써 상기 비트스트림으로부터 적어도 단일 절단된 단항 코드워드(single truncated unary codeword)를 디코딩하는 단계;

상기 변환 블록에 대해, 상기 디코딩된 단일 절단된 단항 코드워드를 사용함으로써 수평 변환 타입을, 상기 디코딩된 단일 절단된 단항 코드워드를 사용함으로써 수직 변환 타입을, 결정하는 단계 - 상기 디코딩된 단일 절단된 단항 코드워드를 사용함으로써, 상기 수평 변환 타입과 상기 수직 변환 타입에 대해 상이한 변환 타입이 결정될 수 있음 -; 및

상기 결정된 수평 변환 타입 및 상기 결정된 수직 변환 타입을 상기 이미지에서의 변환 블록에 적용함으로써 상기 이미지에서의 변환 블록을 디코딩하는 단계를 포함하고,

상기 단일 절단된 단항 코드워드가 복수의 빈들로 구성되는 빈 스트링에 대응되는 경우, 복수의 정보가 CABAC의 컨텍스트들을 지정하기 위해 사용되고, 상기 복수의 빈들 중 서로 다른 빈에 각각 대응되는 상기 복수의 정보는 서로 상이한, 방법.

청구항 2

제1항에 있어서,

상기 수평 변환 타입 및 상기 수직 변환 타입은 변환 세트들과 변환 인덱스들 간의 매핑(mapping)에 의해 결정되고, 상기 변환 세트들 및 상기 변환 인덱스들은 상기 디코딩된 단일 절단된 단항 코드워드에 기초하여 결정되는, 방법.

청구항 3

제2항에 있어서,

상기 결정된 변환 세트들 중 하나는 변환 타입 DST-VII 및 DCT-VIII를 포함하는, 인덱스 0으로 식별되는 변환 세트인, 방법.

청구항 4

제1항에 있어서,

상기 결정된 수평 변환 타입의 수평 변환은, 상기 결정된 수직 변환 타입의 수직 변환이 상기 변환 블록에 대해 수행되기 전에 상기 이미지에서의 변환 블록에 대해 수행되는, 방법.

청구항 5

이미지에서의 변환 블록을 비트스트림으로 인코딩하는 방법으로서:

상기 변환 블록에 대한 수평 변환 타입을 결정하는 단계;

상기 변환 블록에 대한 수직 변환 타입을 결정하는 단계;

상기 결정된 수평 변환 타입 및 상기 결정된 수직 변환 타입을 상기 이미지에서의 변환 블록에 적용함으로써 상기 이미지에서의 변환 블록을 인코딩하는 단계; 및

CABAC(context adaptive binary arithmetic coder)를 사용함으로써 적어도 단일 절단된 단항 코드워드를 인코딩하는 단계를 포함하고,

디코딩하는 방법에서, 상기 수평 변환 타입은 상기 변환 블록에 대해 상기 단일 절단된 단항 코드워드를 사용함

으로써 결정되고, 상기 수직 변환 타입은 상기 변환 블록에 대해 상기 단일 절단된 단항 코드워드를 사용함으로써 결정되며,

상기 디코딩하는 방법에서 상기 단일 절단된 단항 코드워드를 사용함으로써, 상기 수평 변환 타입과 상기 수직 변환 타입에 대해 상이한 변환 타입이 결정될 수 있고,

상기 단일 절단된 단항 코드워드가 복수의 빈들로 구성되는 빈 스트링에 대응되는 경우, 복수의 정보가 CABAC의 컨텍스트들을 지정하기 위해 사용되고, 상기 복수의 빈들 중 서로 다른 빈에 각각 대응되는 상기 복수의 정보는 서로 상이한, 방법.

청구항 6

제5항에 있어서,

상기 수평 변환 타입 및 상기 수직 변환 타입은 변환 세트들과 변환 인덱스들 간의 매핑에 의해 결정되고, 상기 변환 세트들 및 상기 변환 인덱스들은 상기 단일 절단된 단항 코드워드에 기초하여 결정되는, 방법.

청구항 7

제6항에 있어서,

상기 결정된 변환 세트들 중 하나는 변환 타입 DST-VII 및 DCT-VIII를 포함하는, 인덱스 0으로 식별되는 변환 세트인, 방법.

청구항 8

제5항에 있어서,

상기 결정된 수평 변환 타입의 수평 변환은, 상기 결정된 수직 변환 타입의 수직 변환이 상기 변환 블록에 대해 수행되기 전에 상기 이미지에서의 변환 블록에 대해 수행되는, 방법.

청구항 9

비트스트림으로부터 이미지에서의 변환 블록을 디코딩하도록 구성된 장치로서:

CABAC(context adaptive binary arithmetic coder)를 사용함으로써 상기 비트스트림으로부터 적어도 단일 절단된 단항 코드워드를 디코딩하도록 구성된 디코딩 유닛; 및

상기 변환 블록에 대해, 상기 디코딩된 단일 절단된 단항 코드워드를 사용함으로써 수평 변환 타입을, 상기 디코딩된 단일 절단된 단항 코드워드를 사용함으로써 수직 변환 타입을, 결정하도록 구성된 결정 유닛을 포함하고,

상기 디코딩된 단일 절단된 단항 코드워드를 사용함으로써, 상기 수평 변환 타입과 상기 수직 변환 타입에 대해 상이한 변환 타입이 결정될 수 있고,

상기 디코딩 유닛은, 상기 결정된 수평 변환 타입 및 상기 결정된 수직 변환 타입을 상기 이미지에서의 변환 블록에 적용함으로써 상기 이미지에서의 변환 블록을 디코딩하도록 구성되고,

상기 단일 절단된 단항 코드워드가 복수의 빈들로 구성되는 빈 스트링에 대응되는 경우, 복수의 정보가 CABAC의 컨텍스트들을 지정하기 위해 사용되고, 상기 복수의 빈들 중 서로 다른 빈에 각각 대응되는 상기 복수의 정보는 서로 상이한, 장치.

청구항 10

이미지에서의 변환 블록을 비트스트림으로 인코딩하도록 구성된 장치로서:

상기 변환 블록에 대한 수평 변환 타입을 결정하고, 상기 변환 블록에 대한 수직 변환 타입을 결정하도록 구성된 결정 유닛; 및

상기 결정된 수평 변환 타입 및 상기 결정된 수직 변환 타입을 상기 이미지에서의 변환 블록에 적용함으로써 상기 이미지에서의 변환 블록을 인코딩하도록 구성된 인코딩 유닛을 포함하고,

상기 인코딩 유닛은, CABAC(context adaptive binary arithmetic coder)를 사용함으로써 적어도 단일 절단된

단항 코드워드를 인코딩하도록 구성되고,

디코딩 장치에서, 상기 수평 변환 타입은 상기 변환 블록에 대해 상기 단일 절단된 단항 코드워드를 사용함으로써 결정되고, 상기 수직 변환 타입은 상기 변환 블록에 대해 상기 단일 절단된 단항 코드워드를 사용함으로써 결정되며,

상기 디코딩 장치에서 상기 단일 절단된 단항 코드워드를 사용함으로써, 상기 수평 변환 타입과 상기 수직 변환 타입에 대해 상이한 변환 타입이 결정될 수 있고,

상기 단일 절단된 단항 코드워드가 복수의 bin들로 구성되는 bin 스트림에 대응되는 경우, 복수의 정보가 CABAC의 컨텍스트들을 지정하기 위해 사용되고, 상기 복수의 bin들 중 서로 다른 bin에 각각 대응되는 상기 복수의 정보는 서로 상이한, 장치.

청구항 11

컴퓨터 프로그램을 포함하는 비일시적 컴퓨터 판독가능 저장 매체로서,

상기 프로그램은, 컴퓨터에 의해 실행될 때 상기 컴퓨터로 하여금 비트스트림으로부터 이미지에서의 변환 블록을 디코딩하는 방법을 수행하도록 야기하고,

상기 방법은:

CABAC(context adaptive binary arithmetic coder)를 사용함으로써 상기 비트스트림으로부터 적어도 단일 절단된 단항 코드워드를 디코딩하는 단계;

상기 변환 블록에 대해, 상기 디코딩된 단일 절단된 단항 코드워드를 사용함으로써 수평 변환 타입을, 상기 디코딩된 단일 절단된 단항 코드워드를 사용함으로써 수직 변환 타입을, 모두 결정하는 단계 - 상기 디코딩된 단일 절단된 단항 코드워드를 사용함으로써, 상기 수평 변환 타입과 상기 수직 변환 타입에 대해 상이한 변환 타입이 결정될 수 있음 -; 및

상기 결정된 수평 변환 타입 및 상기 결정된 수직 변환 타입을 상기 이미지에서의 변환 블록에 적용함으로써 상기 이미지에서의 변환 블록을 디코딩하는 단계를 포함하고,

상기 단일 절단된 단항 코드워드가 복수의 bin들로 구성되는 bin 스트림에 대응되는 경우, 복수의 정보가 CABAC의 컨텍스트들을 지정하기 위해 사용되고, 상기 복수의 bin들 중 서로 다른 bin에 각각 대응되는 상기 복수의 정보는 서로 상이한, 비일시적 컴퓨터 판독가능 저장 매체.

청구항 12

컴퓨터 프로그램을 포함하는 비일시적 컴퓨터 판독가능 저장 매체로서,

상기 프로그램은, 컴퓨터에 의해 실행될 때 상기 컴퓨터로 하여금 이미지에서의 변환 블록을 비트스트림으로 인코딩하는 방법을 수행하도록 야기하고,

상기 방법은:

상기 변환 블록에 대한 수평 변환 타입을 결정하는 단계;

상기 변환 블록에 대한 수직 변환 타입을 결정하는 단계;

상기 결정된 수평 변환 타입 및 상기 결정된 수직 변환 타입을 상기 이미지에서의 변환 블록에 적용함으로써 상기 이미지에서의 변환 블록을 인코딩하는 단계; 및

CABAC(context adaptive binary arithmetic coder)를 사용함으로써 적어도 단일 절단된 단항 코드워드를 인코딩하는 단계를 포함하고,

디코딩하는 방법에서, 상기 수평 변환 타입은 상기 변환 블록에 대해 상기 단일 절단된 단항 코드워드를 사용함으로써 결정되고, 상기 수직 변환 타입은 상기 변환 블록에 대해 상기 단일 절단된 단항 코드워드를 사용함으로써 결정되며,

상기 디코딩하는 방법에서 상기 단일 절단된 단항 코드워드를 사용함으로써, 상기 수평 변환 타입과 상기 수직 변환 타입에 대해 상이한 변환 타입이 결정될 수 있고,

상기 단일 절단된 단항 코드워드가 복수의 빈들로 구성되는 빈 스트링에 대응되는 경우, 복수의 정보가 CABAC의 컨텍스트들을 지정하기 위해 사용되고, 상기 복수의 빈들 중 서로 다른 빈에 각각 대응되는 상기 복수의 정보는 서로 상이한, 비밀시적 컴퓨터 판독가능 저장 매체.

청구항 13

삭제

청구항 14

삭제

청구항 15

삭제

청구항 16

삭제

발명의 설명

기술 분야

[0001] 본 발명은 일반적으로 디지털 비디오 신호 처리에 관한 것으로, 특히, 비디오 샘플들의 변환된 블록을 인코딩 및 디코딩하기 위한 방법, 장치 및 시스템에 관한 것이다. 본 발명은 또한 비디오 샘플들의 변환된 블록을 인코딩 및 디코딩하기 위한 컴퓨터 프로그램이 기록된 컴퓨터 판독가능 매체를 포함하는 컴퓨터 프로그램 제품에 관한 것이다.

배경 기술

[0002] 비디오 데이터의 송신 및 저장을 위한 애플리케이션들을 포함하는, 비디오 코딩을 위한 많은 애플리케이션들이 현재 존재한다. 많은 비디오 코딩 표준들이 또한 개발되었고, 다른 비디오 코딩 표준들이 현재 개발 중이다. 비디오 코딩 표준화에서의 최근의 발전은 "JVET(Joint Video Experts Team)"라고 불리는 그룹을 형성하게 하였다. JVET(Joint Video Experts Team)는 "VCEG(Video Coding Experts Group)"로서 알려진 ITU(International Telecommunication Union)의 ITU-T(Telecommunication Standardisation Sector)의 SG16/Q6(Study Group 16, Question 6)의 멤버들, 및 "MPEG(Moving Picture Experts Group)"로서 또한 알려진 ISO/IEC JTC1/SC29/WG11(International Organisations for Standardisation/International Electrotechnical Commission Joint Technical Committee 1/Subcommittee 29/Working Group 11)의 멤버들을 포함한다.

[0003] JVET(Joint Video Experts Team)는 CfP(Call for Proposals)를 발행했고, 응답들은 미국 샌디에고에서 10번째 회의에서 분석되었다. 제출된 응답들은 현재의 최신 비디오 압축 표준, 즉 "HEVC(high efficiency video coding)"의 비디오 압축 능력보다 상당히 능가하는 능력을 보여주었다. 이러한 성과에 기초하여, VVC(versatile video coding)으로 명명되는 새로운 비디오 압축 표준을 개발하기 위한 프로젝트를 시작하기로 결정되었다. VVC는, 특히 비디오 포맷들이 (예를 들어, 더 높은 해상도 및 더 높은 프레임 레이트를 가져서) 용량이 증가하고 또한 대역폭 비용들이 비교적 높은, WAN을 통한 서비스 전달에 대한 증가하는 시장 수요를 다뤄야 함에 따라 점점 더 높아지는 압축 성능에 대한 지속적인 수요를 해결할 것으로 예상된다. 동시에, VVC는 동시대의 실리콘 공정에서 구현가능해야만 하고 또한 달성된 성능 대 구현 비용(예를 들어, 실리콘 면적, CPU 프로세서 부하, 메모리 이용률 및 대역폭의 관점에서) 간의 수용가능한 절충을 제공해야만 한다.

[0004] 비디오 데이터는 이미지 데이터의 프레임들의 시퀀스를 포함하며, 이들 각각은 하나 이상의 색 채널을 포함한다. 일반적으로, 하나의 원색 채널 및 2개의 2차 색 채널이 존재한다. 원색 채널은 일반적으로 "루마" 채널로 지칭되고, 2차 색 채널(들)은 일반적으로 "크로마" 채널로 지칭된다. 비디오 데이터가 전형적으로 RGB (적색-녹색-청색) 색 공간에서 디스플레이되지만, 색 공간은 3개의 각자의 성분 사이에 높은 상관 정도를 갖는다. 인코더 또는 디코더에 의해 보이는 비디오 데이터 표현은 종종 YCbCr과 같은 색 공간을 이용하고 있다. YCbCr은 Y(주요) 채널에서의 루마 및 Cb 및 Cr(보조) 채널들에서의 크로마에 집중한다. 또한, Cb 및 Cr 채널들은 루마 채널에 비해 더 낮은 레이트로 샘플링될 수 있는데, 예를 들어, 수평으로 절반이고 수직으로 절반이며,

"4:2:0 크로마 포맷"으로 알려져 있다.

[0005] VVC는 프레임들이 블록들로 분할되고 블록들이 특정 순서로 처리되는 "블록 기반" 코덱이다. 각각의 블록에 대해, 블록의 내용들의 예측이 생성되고, 예측과 인코더에 대한 입력으로서 보이는 실제 블록 내용들 사이의 차이 (또는 공간 도메인에서의 "잔차(residual)")의 표현이 형성된다. 차이는 DCT(Discrete Cosine Transform) 또는 다른 변환과 같은 순방향 1차 변환을 잔차 값들의 블록에 적용한 결과로 생기는 잔차 계수들의 시퀀스로서 코딩될 수 있다. 이 1차 변환은 잔차 샘플들의 블록에 별개로 적용되는데, 즉, 2차원 변환은 2개의 패스로 수행되어, 먼저 행들은 1차원 변환을 각각의 행에 적용함으로써 변환되고, 그 후 1차원 변환을 각각의 열에 적용하여 잔차 샘플들을 실질적으로 상관해제시키는 잔차 계수들의 블록을 산출함으로써 이 결과를 변환한다. 더욱이, 이 행 및 열 처리에서 이용하기 위한 다중 변환이 이용가능하여, 어느 것이 이용될지를 지시하기 위한 시그널링을 필요로 한다. 블록에서 상이한 변환들을 이용하는 옵션이 압축 이점을 제공하지만, 이것은 어느 변환이 수평으로 및 수직으로 이용될지를 선택하기 위해 인덱스들을 코딩할 필요성에 의해 약간 감소된다. 이 시그널링의 효율적인 코딩은 이 시그널링 오버헤드의 영향에 유익하다.

발명의 내용

해결하려는 과제

과제의 해결 수단

[0006] 본 발명의 목적은 기존 구성들의 하나 이상의 단점을 실질적으로 극복하거나 적어도 개선하는 것이다.

[0007] 본 개시내용의 양태는 비트스트림으로부터 이미지 프레임에서의 변환 블록을 디코딩하는 방법을 제공하는데, 이러한 방법은: 비트스트림으로부터 단일 절단된 단항 이진화(single truncated unary binarisation)를 디코딩하는 단계 - 단일 절단된 단항 이진화는 이미지 프레임의 변환 블록의 수평 변환 및 수직 변환을 위해 사용됨-; 디코딩된 단일 절단된 단항 이진화에 기초하여 수평 및 수직 변환의 타입을 결정하는 단계; 및 결정된 타입의 수평 및 수직 변환을 이미지의 변환 블록에 적용함으로써 이미지 프레임에서의 변환 블록을 디코딩하는 단계를 포함한다.

[0008] 본 개시내용의 또 다른 양태는 컴퓨터 애플리케이션 프로그램들을 포함하는 비일시적 컴퓨터 판독가능 매체를 제공하고, 프로그램들은 컴퓨터로 하여금 비트스트림으로부터 이미지 프레임에서 변환 블록을 디코딩하는 방법을 수행하게 하기 위해 컴퓨터 장치에 의해 실행가능하고, 방법은: 비트스트림으로부터 단일 절단된 단항 이진화를 디코딩하는 단계 - 단일 절단된 단항 이진화는 이미지 프레임의 변환 블록의 수평 변환 및 수직 변환을 위해 사용됨 -; 디코딩된 단일 절단된 단항 이진화에 기초하여 수평 및 수직 변환의 타입을 결정하는 단계; 및 결정된 타입의 수평 및 수직 변환을 이미지의 변환 블록에 적용함으로써 이미지 프레임에서의 변환 블록을 디코딩하는 단계를 포함한다.

[0009] 다른 양태들도 개시된다.

도면의 간단한 설명

[0010] 본 발명의 적어도 하나의 실시예가 다음의 도면들 및 부록들을 참조하여 이제 설명될 것이다.

도 1은 비디오 인코딩 및 디코딩 시스템을 도시한 개략 블록도이다.

도 2a 및 도 2b는 도 1의 비디오 인코딩 및 디코딩 시스템 중 하나 또는 둘 다가 실시될 수 있는 범용 컴퓨터 시스템의 개략 블록도를 형성한다.

도 3은 비디오 인코더의 기능 모듈들을 도시하는 개략 블록도이다.

도 4는 비디오 디코더의 기능 모듈을 도시하는 개략 블록도이다.

도 5는 VVC의 트리 구조에서 블록의 하나 이상의 블록으로의 이용가능한 분할들을 도시하는 개략 블록도이다.

도 6은 VVC의 트리 구조에서 블록의 하나 이상의 블록으로의 허용된 분할들을 달성하는 데이터흐름의 개략도이다.

도 7은 CTU(coding tree unit)의 다수의 CU(coding unit)로의 예시적 분할이다.

도 8a는 인트라 예측 모드들을 도시하는 도면이다.

도 8b는 변환 인덱스에 따라 변환 세트로부터 특정 변환으로의 매핑을 도시하는 표이다.

도 8c는 인트라 예측 모드들로부터 수평 방향 및 수직 방향에서의 변환 블록에 대한 변환 세트들의 매핑을 도시하는 표이다.

도 9는 도 3의 비디오 인코더 또는 도 4의 비디오 디코더의 역변환 모듈을 도시하는 개략 블록도이다.

도 10a는 절단된 단항 선택스 요소와 수평으로 및 수직으로의 변환 인덱싱을 위한 변환 인덱스들 사이의 관계를 도시하는 도면이다.

도 10b는 절단된 단항 선택스 요소와 수평 및 수직으로의 변환 인덱싱을 위한 변환 인덱스들 사이의 대안적인 관계를 도시하는 도면이다.

도 11은 잔차 계수들의 변환 블록을 비트스트림으로 인코딩하기 위해 명시적 변환 선택에 따라 1차 변환을 수행하는 방법의 흐름도이다.

도 12는 명시적 변환 선택에 따라 1차 역변환에 의해 비트스트림으로부터 잔차 계수들의 변환 블록을 디코딩하는 방법의 흐름도이다.

발명을 실시하기 위한 구체적인 내용

- [0011] [최상의 모드를 포함하는 상세한 설명]
- [0012] 첨부 도면들 중 임의의 하나 이상에서 동일한 참조 번호들을 갖는 단계들 및/또는 특징들이 참조되는 경우, 그 단계들 및/또는 특징들은, 반대 의도가 나타나지 않는 한, 이 설명의 목적을 위해 동일한 기능(들) 또는 동작(들)을 갖는다.
- [0013] 도 1은 비디오 인코딩 및 디코딩 시스템(100)의 기능 모듈을 도시하는 개략 블록도이다. 시스템(100)은 압축 효율을 개선하고 및/또는 감소된 구현 비용을 달성하기 위해 계수 스캐닝 방법을 활용할 수 있다. 시스템(100)은 소스 디바이스(110) 및 목적지 디바이스(130)를 포함한다. 통신 채널(120)은 인코딩된 비디오 정보를 소스 디바이스(110)로부터 목적지 디바이스(130)로 통신하기 위해 사용된다. 일부 구성들에서, 소스 디바이스(110) 및 목적지 디바이스(130) 어느 하나 또는 둘 다는 각자의 모바일 폰 핸드셋들 또는 "스마트폰들"을 포함할 수 있으며, 이 경우에 통신 채널(120)은 무선 채널이다. 다른 구성에서, 소스 디바이스(110) 및 목적지 디바이스(130)는 비디오 회의 장비를 포함할 수 있으며, 이 경우에 통신 채널(120)은 전형적으로 인터넷 접속과 같은 유선 채널이다. 더욱이, 소스 디바이스(110) 및 목적지 디바이스(130)는 OTA(over-the-air) 텔레비전 방송들을 지원하는 디바이스들, 케이블 텔레비전 애플리케이션들, 인터넷 비디오 애플리케이션들(스트리밍을 포함함) 및 인코딩된 비디오 데이터가 파일 서버에서의 하드 디스크 드라이브들과 같은 일부 컴퓨터 관독가능 저장 매체상에서 캡처되는 애플리케이션들을 포함하는 광범위한 디바이스들 중 임의의 것을 포함할 수 있다.
- [0014] 도 1에 도시된 바와 같이, 소스 디바이스(110)는 비디오 소스(112), 비디오 인코더(114) 및 송신기(116)를 포함한다. 비디오 소스(112)는 전형적으로 이미징 센서와 같은, 캡처된 비디오 프레임 데이터의 소스((113)으로 도시됨), 비일시적 기록 매체상에 저장된 이전에 캡처된 비디오 시퀀스, 또는 원격 이미징 센서로부터의 비디오 피드를 포함한다. 비디오 소스(112)는 또한, 예를 들어, 컴퓨터 디바이스, 예를 들어, 태블릿 컴퓨터상에서 실행되는 운영 체제 및 다양한 애플리케이션들의 비디오 출력을 디스플레이하는 컴퓨터 그래픽 카드의 출력일 수 있다. 비디오 소스(112)로서 이미징 센서를 포함할 수 있는 소스 디바이스들(110)의 예들은 스마트 폰들, 비디오 캠코더들, 전문 비디오 카메라들, 및 네트워크 비디오 카메라들을 포함한다. 비디오 인코더(114)는 도 3을 참조하여 더 설명되는 바와 같이 비디오 소스(112)로부터의 캡처된 프레임 데이터(화살표(113)로 표시됨)를 비트스트림(화살표(115)로 표시됨)으로 변환(또는 "인코딩") 한다. 비트스트림(115)은 인코딩된 비디오 데이터(또는 "인코딩된 비디오 정보")로서 통신 채널(120)을 통해 송신기(116)에 의해 송신된다. 또한, 비트스트림(115)이 나중에 통신 채널(120)을 통해 송신될 때까지 또는 통신 채널(120)을 통한 전송 대신에 "플레이스" 메모리 또는 하드 디스크 드라이브와 같은 비일시적 저장 디바이스(122)에 저장되는 것이 가능하다.
- [0015] 목적지 디바이스(130)는 수신기(132), 비디오 디코더(134) 및 디스플레이 디바이스(136)를 포함한다. 수신기(132)는 통신 채널(120)로부터 인코딩된 비디오 데이터를 수신하고 수신된 비디오 데이터를 비트스트림(화살표

(133)으로 표시됨)으로서 비디오 디코더(134)에 전달한다. 비디오 디코더(134)는 이어서 화살표(135)로 표시된 디코딩된 프레임 데이터를 디스플레이 디바이스(136)에 출력한다. 디스플레이 디바이스(136)의 예는 음극선관, 스마트폰, 태블릿 컴퓨터, 컴퓨터 모니터에서와 같은 또는 독립형 텔레비전 세트에서와 같은 액정 디스플레이를 포함한다. 소스 디바이스(110) 및 목적지 디바이스(130) 각각의 기능이 단일 디바이스로 구현되는 것도 가능하며, 그 예는 이동 전화 핸드셋 및 태블릿 컴퓨터를 포함한다.

[0016] 앞서 언급된 예시적인 디바이스들에도 불구하고, 소스 디바이스(110) 및 목적지 디바이스(130) 각각은, 전형적으로 하드웨어와 소프트웨어 컴포넌트들의 조합을 통해 범용 컴퓨팅 시스템 내에 구성될 수 있다. 도 2a는 컴퓨터 시스템(200)을 예시하는데, 이 컴퓨터 시스템은: 컴퓨터 모듈(201); 키보드(202), 마우스 포인터 디바이스(203), 스캐너(226), 비디오 소스(112)로서 구성될 수 있는 카메라(227), 및 마이크로폰(280)과 같은 입력 디바이스들; 및 프린터(215), 디스플레이 디바이스(136)로서 구성될 수 있는 디스플레이 디바이스(214), 및 라우드스피커(217)를 포함하는 출력 디바이스들을 포함한다. 외부 모뎀(Modulator-Demodulator, Modem) 송수신기 디바이스(216)는 접속(221)을 통해 통신 네트워크(220)로/로부터 통신하기 위해 컴퓨터 모듈(201)에 의해 사용될 수 있다. 통신 채널(120)을 나타낼 수 있는 통신 네트워크(220)는 인터넷, 셀룰러 통신 네트워크, 또는 사실 WAN과 같은 광역 네트워크(WAN)일 수 있다. 접속(221)이 전화선인 경우, 모뎀(216)은 전통적인 "다이얼-업" 모뎀일 수 있다. 대안적으로, 접속(221)이 고용량(예컨대, 케이블 또는 광학) 접속인 경우, 모뎀(216)은 광대역 모뎀일 수 있다. 무선 모뎀은 또한 통신 네트워크(220)로의 무선 접속을 위해 사용될 수 있다. 송수신기 디바이스(216)는 송신기(116) 및 수신기(132)의 기능을 제공할 수 있고, 통신 채널(120)은 접속(221)에서 구현될 수 있다.

[0017] 컴퓨터 모듈(201)은 전형적으로 적어도 하나의 프로세서 유닛(205) 및 메모리 유닛(206)을 포함한다. 예를 들어, 메모리 유닛(206)은 반도체 랜덤 액세스 메모리(RAM) 및 반도체 판독 전용 메모리(ROM)를 가질 수 있다. 컴퓨터 모듈(201)은 또한 비디오 디스플레이(214), 라우드스피커들(217) 및 마이크로폰(280)에 결합되는 오디오-비디오 인터페이스(207); 키보드(202), 마우스(203), 스캐너(226), 카메라(227) 및 선택적으로 조이스틱 또는 다른 휴먼 인터페이스 디바이스(도시되지 않음)에 결합되는 I/O 인터페이스(213); 및 외부 모뎀(216) 및 프린터(215)에 대한 인터페이스(208)를 포함하는 다수의 입력/출력(I/O) 인터페이스를 포함한다. 오디오-비디오 인터페이스(207)로부터 컴퓨터 모니터(214)로의 신호는 일반적으로 컴퓨터 그래픽 카드의 출력이다. 일부 구현들에서, 모뎀(216)은 컴퓨터 모듈(201) 내에, 예를 들어, 인터페이스(208) 내에 통합될 수 있다. 컴퓨터 모듈(201)은 또한, LAN(Local Area Network)으로 알려진 로컬 영역 통신 네트워크(local-area communications network)(222)에 대한 접속(223)을 통한 컴퓨터 시스템(200)의 결합을 허용하는 로컬 네트워크 인터페이스(211)를 갖는다. 도 2a에 예시된 바와 같이, 로컬 통신 네트워크(222)는 또한 접속(224)을 통해 광역 네트워크(220)에 결합될 수 있으며, 접속은 전형적으로 소위 "방화벽" 디바이스 또는 유사한 기능의 디바이스를 포함할 것이다. 로컬 네트워크 인터페이스(211)는 Ethernet™ 회로 카드, Bluetooth™ 무선 구성 또는 IEEE 802.11 무선 구성을 포함할 수 있고; 그러나, 다수의 다른 타입의 인터페이스가 인터페이스(211)에 대해 실시될 수 있다. 로컬 네트워크 인터페이스(211)는 또한 송신기(116) 및 수신기(132)의 기능을 제공할 수 있고, 통신 채널(120)은 또한 로컬 통신 네트워크(222)에서 구현될 수 있다.

[0018] I/O 인터페이스들(208 및 213)은 직렬 및 병렬 접속성 중 어느 하나 또는 둘 다를 제공할 수 있으며, 전자는 전형적으로 USB(Universal Serial Bus) 표준에 따라 구현되고 대응하는 USB 커넥터들(도시되지 않음)을 갖는다. 저장 디바이스들(209)이 제공되고 전형적으로 하드 디스크 드라이브(HDD)(210)를 포함한다. 플로피 디스크 드라이브 및 자기 테이프 드라이브(도시되지 않음)와 같은 다른 저장 디바이스들도 사용될 수 있다. 광 디스크 드라이브(212)는 전형적으로 데이터의 비휘발성 소스로서 기능하도록 제공된다. 예를 들어, 광학 디스크들(예를 들어, CD-ROM, DVD, Blu-ray Disc™), USB-RAM, 휴대용, 외부 하드 드라이브들, 및 플로피 디스크들과 같은 휴대용 메모리 디바이스들은 컴퓨터 시스템(200)에 대한 데이터의 적절한 소스들로서 사용될 수 있다. 전형적으로, HDD(210), 광 드라이브(212), 네트워크들(220 및 222) 중 임의의 것은 또한 비디오 소스(112)로서, 또는 디스플레이(214)를 통한 재생을 위해 저장된 디코딩된 비디오 데이터에 대한 목적지로서 동작하도록 구성될 수 있다. 시스템(100)의 소스 디바이스(110) 및 목적지 디바이스(130), 또는 시스템(100)의 소스 디바이스(110) 및 목적지 디바이스(130)는 컴퓨터 시스템(200)에서 구현될 수 있다.

[0019] 컴퓨터 모듈(201)의 컴포넌트들(205 내지 213)은 전형적으로 상호 접속된 버스(204)를 통해, 그리고 관련 기술 분야의 통상의 기술자에게 알려진 컴퓨터 시스템(200)의 종래의 동작 모드를 야기하는 방식으로 통신한다. 예를 들어, 프로세서(205)는 접속(218)을 이용하여 시스템 버스(204)에 결합된다. 마찬가지로, 메모리(206) 및 광 디스크 드라이브(212)는 접속들(219)에 의해 시스템 버스(204)에 결합된다. 설명된 구성이 실시될 수 있는

컴퓨터의 예는 IBM-PC 및 호환 컴퓨터들, Sun SPARCstation, Apple MacTM 또는 유사한 컴퓨터 시스템을 포함한다.

- [0020] 적절한 경우 또는 원하는 경우, 비디오 인코더(114) 및 비디오 디코더(134)는 물론이고 이하에서 설명되는 방법들이 컴퓨터 시스템(200)을 사용하여 구현될 수 있으며, 여기서 비디오 인코더(114), 비디오 디코더(134) 및 설명된 방법들은 컴퓨터 시스템(200) 내에서 실행가능한 하나 이상의 소프트웨어 애플리케이션 프로그램(233)으로서 구현될 수 있다. 특히, 컴퓨터 시스템(200) 내에서 수행되는 소프트웨어(233)에서의 명령어들(231)(도 2b 참조)에 의해 비디오 인코더(114), 비디오 디코더(134) 및 설명된 방법들의 단계들이 이루어진다. 소프트웨어 명령어들(231)은 각각이 하나 이상의 특정 작업을 수행하기 위한 하나 이상의 코드 모듈로서 형성될 수 있다. 소프트웨어는 또한 2개의 별개의 부분으로 분할될 수 있고, 여기서 제1 부분 및 대응하는 코드 모듈들은 설명된 방법들을 수행하고, 제2 부분 및 대응하는 코드 모듈들은 제1 부분과 사용자 사이의 사용자 인터페이스를 관리한다.
- [0021] 소프트웨어는 예를 들어, 다음에 설명되는 저장 디바이스들을 포함하는 컴퓨터 판독가능 매체에 저장될 수 있다. 소프트웨어는 컴퓨터 판독가능 매체로부터 컴퓨터 시스템(200) 내로 로딩되고, 그 후 컴퓨터 시스템(200)에 의해 실행된다. 컴퓨터 판독가능 매체상에 기록된 그러한 소프트웨어 또는 컴퓨터 프로그램을 갖는 컴퓨터 판독가능 매체는 컴퓨터 프로그램 제품이다. 컴퓨터 시스템(200)에서의 컴퓨터 프로그램 제품의 사용은 바람직하게는 비디오 인코더(114), 비디오 디코더(134) 및 설명된 방법들을 구현하기 위한 유리한 장치를 달성한다.
- [0022] 소프트웨어(233)는 전형적으로 HDD(210) 또는 메모리(206)에 저장된다. 소프트웨어는 컴퓨터 판독가능 매체로부터 컴퓨터 시스템(200) 내로 로딩되고, 컴퓨터 시스템(200)에 의해 실행된다. 따라서, 예를 들어, 소프트웨어(233)는 광학 디스크 드라이브(212)에 의해 판독되는 광학적으로 판독가능한 디스크 저장 매체(예를 들어, CD-ROM)(225)상에 저장될 수 있다.
- [0023] 일부 예들에서, 애플리케이션 프로그램들(233)은 하나 이상의 CD-ROM(225)상에 인코딩되어 사용자에게 공급되어 대응하는 드라이브(212)를 통해 판독될 수 있거나, 대안적으로 네트워크들(220 또는 222)로부터 사용자에게 의해 판독될 수 있다. 또한, 소프트웨어는 또한 다른 컴퓨터 판독가능 매체로부터 컴퓨터 시스템(200) 내로 로딩될 수 있다. 컴퓨터 판독가능 저장 매체는 실행 및/또는 처리를 위해 컴퓨터 시스템(200)에 기록된 명령어 및/또는 데이터를 제공하는 임의의 비일시적 유형의 저장 매체를 지칭한다. 이러한 저장 매체의 예는, 이러한 디바이스가 컴퓨터 모듈(201)의 내부 또는 외부에 있는지 아니든지 간에, 플로피 디스크, 자기 테이프, CD-ROM, DVD, Blu-ray DiscTM, 하드 디스크 드라이브, ROM 또는 집적 회로, USB 메모리, 광자기 디스크, 또는 PCMCIA 카드 등과 같은 컴퓨터 판독가능 카드를 포함한다. 소프트웨어, 애플리케이션 프로그램들, 명령어 및/또는 비디오 데이터 또는 인코딩된 비디오 데이터를 컴퓨터 모듈(401)에 제공하는 데 또한 참여할 수 있는 일시적 또는 비유형의(non-tangible) 컴퓨터 판독가능 송신 매체들의 예들은 무선 또는 적외선 송신 채널들뿐만 아니라 또 다른 컴퓨터 또는 네트워크화된 디바이스에의 네트워크 접속, 및 웹사이트들상에 기록된 이메일 송신 및 정보 등을 포함하는 인터넷 또는 인트라넷들 등을 포함한다.
- [0024] 위에서 언급된 애플리케이션 프로그램들(233)의 제2 부분 및 대응하는 코드 모듈들은 디스플레이(214)상에 렌더링되거나 다른 방식으로 표현될 하나 이상의 그래픽 사용자 인터페이스(GUI)를 구현하도록 실행될 수 있다. 전형적으로 키보드(202) 및 마우스(203)의 조작을 통해, 컴퓨터 시스템(200) 및 애플리케이션의 사용자는 GUI(들)와 연관된 애플리케이션들에 제어 커맨드 및/또는 입력을 제공하기 위해 기능적으로 적응가능한 방식으로 인터페이스를 조작할 수 있다. 라우드스피커들(217)을 통해 출력되는 음성 프롬프트들 및 마이크로폰(280)을 통해 입력되는 사용자 음성 커맨드들을 이용하는 오디오 인터페이스와 같은 다른 형식들의 기능적으로 적응가능한 사용자 인터페이스들이 또한 구현될 수 있다.
- [0025] 도 2b는 프로세서(205) 및 "메모리"(234)의 상세한 개략 블록도이다. 메모리(234)는 도 2a의 컴퓨터 모듈(201)에 의해 액세스될 수 있는 (HDD(209) 및 반도체 메모리(206)를 포함하는) 모든 메모리 모듈들의 논리적 모음을 나타낸다.
- [0026] 컴퓨터 모듈(201)이 초기에 전원이 켜지면, POST(power-on self-test) 프로그램(250)이 실행된다. POST 프로그램(250)은 전형적으로 도 2a의 반도체 메모리(206)의 ROM(249)에 저장된다. 소프트웨어를 저장하는 ROM(249)과 같은 하드웨어 디바이스는 때때로 펌웨어라고 지칭된다. POST 프로그램(250)은 적절한 기능을 보장하기 위해 컴퓨터 모듈(201) 내의 하드웨어를 조사하고, 정확한 동작을 위해, 프로세서(205), 메모리(234)(209, 206), 및 또한 전형적으로 ROM(249)에 저장된 기본 입출력 시스템 소프트웨어(BIOS) 모듈(251)을 검사한다. 일

단 POST 프로그램(250)이 성공적으로 실행되면, BIOS(251)는 도 2a의 하드 디스크 드라이브(210)를 활성화시킨다. 하드 디스크 드라이브(210)의 활성화는 하드 디스크 드라이브(210)상에 상주하는 부트스트랩 로더 프로그램(252)이 프로세서(205)를 통해 실행되게 야기한다. 이는 운영 체제(253)를 RAM 메모리(206) 내에 로드하고, 그 상에서 운영 체제(253)는 동작을 시작한다. 운영 체제(253)는 프로세서 관리, 메모리 관리, 디바이스 관리, 저장소 관리, 소프트웨어 애플리케이션 인터페이스, 및 일반 사용자 인터페이스를 포함하는, 다양한 하이 레벨 기능들을 이행하기 위해, 프로세서(205)에 의해 실행가능한 시스템 레벨 애플리케이션이다.

[0027] 운영 체제(253)는 컴퓨터 모듈(201)상에서 실행되는 각각의 프로세스 또는 애플리케이션이 또 다른 프로세스에 할당된 메모리와 충돌하지 않고 실행될 충분한 메모리를 갖도록 보장하기 위해 메모리(234)(209, 206)를 관리한다. 또한, 도 2a의 컴퓨터 시스템(200)에서 이용가능한 상이한 타입의 메모리는 각각의 프로세스가 효과적으로 실행될 수 있도록 적절하게 사용되어야만 한다. 따라서, 모아진 메모리(234)는 (달리 언급되지 않는 한) 메모리의 특정 세그먼트들이 어떻게 할당되는지를 예시하도록 의도된 것이 아니라, 오히려 컴퓨터 시스템(200)에 의해 액세스가능한 메모리의 일반적인 뷰 및 그러한 것이 어떻게 사용되는지를 제공하도록 의도된다.

[0028] 도 2b에 도시된 바와 같이, 프로세서(205)는 제어 유닛(239), 산술 로직 유닛(ALU)(240), 및 때때로 캐시 메모리라고 불리는 로컬 또는 내부 메모리(248)를 포함하는 다수의 기능 모듈을 포함한다. 캐시 메모리(248)는 전형적으로 레지스터 섹션에 다수의 저장 레지스터(244-246)를 포함한다. 하나 이상의 내부 버스(241)가 이들 기능 모듈들을 기능적으로 상호접속시킨다. 프로세서(205)는 전형적으로 또한 접속(218)을 이용하여 시스템 버스(204)를 통해 외부 디바이스들과 통신하기 위한 하나 이상의 인터페이스(242)를 갖는다. 메모리(234)는 접속(219)을 이용하여 버스(204)에 결합된다.

[0029] 애플리케이션 프로그램(233)은 조건부 분기 및 루프 명령어들을 포함할 수 있는 명령어들의 시퀀스(231)를 포함한다. 프로그램(233)은 또한 프로그램(233)의 실행에 사용되는 데이터(232)를 포함할 수 있다. 명령어들(231) 및 데이터(232)는 제각기 메모리 위치들(228, 229, 230 및 235, 236, 237)에 저장된다. 명령어들(231) 및 메모리 위치들(228-230)의 상대적 크기에 좌우되어, 특정 명령어는 메모리 위치(230)에 도시된 명령어에 의해 묘사된 바와 같이 단일 메모리 위치에 저장될 수 있다. 대안적으로, 명령어는 메모리 위치들(228 및 229)에 도시된 명령어 세그먼트들에 의해 묘사된 바와 같이, 그 각각이 개별 메모리 위치에 저장되는 다수의 부분으로 세그먼트화될 수 있다.

[0030] 일반적으로, 프로세서(205)는 그 안에서 실행되는 명령어들의 세트를 제공받는다. 프로세서(205)는, 프로세서(205)가 명령어들의 또 다른 세트를 실행함으로써 그에 반응하는 후속 입력을 대기한다. 각각의 입력은 입력 디바이스들(202, 203) 중 하나 이상에 의해 생성된 데이터, 네트워크들(220, 202) 중 하나에 걸쳐 외부 소스로부터 수신된 데이터, 저장 디바이스들(206, 209) 중 하나로부터 검색된 데이터, 또는 대응하는 판독기(212)에 삽입된 저장 매체(225)로부터 검색된 데이터를 포함하는 다수의 소스 중 하나 이상으로부터 제공될 수 있으며, 이들 모두는 도 2a에 묘사되어 있다. 명령어들의 세트의 실행은 일부 경우들에서 데이터의 출력이라는 결과를 낳을 수 있다. 실행은 또한 데이터 또는 변수들을 메모리(234)에 저장하는 것을 수반할 수 있다.

[0031] 비디오 인코더(114), 비디오 디코더(134), 및 설명된 방법은 대응하는 메모리 위치들(255, 256, 257)에서 메모리(234)에 저장되는 입력 변수들(254)을 사용할 수 있다. 비디오 인코더(114), 비디오 디코더(134), 및 설명된 방법은 대응하는 메모리 위치들(262, 263, 264)에서 메모리(234)에 저장되는 출력 변수들(261)을 생성한다. 중간 변수들(258)은 메모리 위치들(259, 260, 266, 및 267)에 저장될 수 있다.

[0032] 도 2b의 프로세서(205)를 참조하면, 레지스터들(244, 245, 246), 산술 논리 유닛(ALU)(240), 및 제어 유닛(239)은 함께 작업하여 프로그램(233)을 구성하는 명령어 세트에서의 모든 명령어마다에 대해 "페치, 디코드, 및 실행" 사이클들을 수행하는 데 필요한 마이크로연산들의 시퀀스들을 수행한다. 각각의 페치, 디코드, 및 실행 사이클은 다음을 포함한다:

[0033] (a) 메모리 위치(228, 229, 230)로부터 명령어(231)를 페치 또는 판독하는 페치 동작;

[0034] (b) 제어 유닛(239)이 어느 명령어가 페치되었는지를 결정하는 디코드 동작; 및

[0035] (c) 제어 유닛(239) 및/또는 ALU(240)가 명령어를 실행하는 실행 동작.

[0036] 그 후, 다음 명령어에 대한 추가 페치, 디코드, 및 실행 사이클이 실행될 수 있다. 유사하게, 제어 유닛(239)이 메모리 위치(232)에 값을 저장하거나 기입하는 저장 사이클이 수행될 수 있다.

[0037] 설명된 도 12 및 도 13의 방법에서의 각각의 단계 또는 서브 프로세스는 프로그램(233)의 하나 이상의 세그먼트

와 연관되어 있으며, 프로그램(233)의 주목한 세그먼트들에 대한 명령어 세트에서의 모든 명령어마다에 대한 페치, 디코드, 및 실행 사이클들을 수행하기 위해 함께 작업하는 프로세서(205) 내의 레지스터 섹션(244, 245, 247), ALU(240), 및 제어 유닛(239)에 의해 전형적으로 수행된다.

[0038] 도 3은 비디오 인코더(114)의 기능 모듈을 도시하는 개략 블록도이다. 도 4는 비디오 디코더(134)의 기능 모듈을 도시하는 개략 블록도이다. 일반적으로, 데이터는, 블록들을 고정 크기의 서브 블록들로 분할하는 것과 같이 샘플들 또는 계수들의 그룹들로, 또는 어레이들로서 비디오 인코더(114) 및 비디오 디코더(134) 내에서의 기능 모듈들 간에 전달된다. 비디오 인코더(114) 및 비디오 디코더(134)는 도 2a 및 도 2b에 도시된 바와 같이 범용 컴퓨터 시스템(200)을 이용하여 구현될 수 있고, 여기서 다양한 기능 모듈들은 컴퓨터 시스템(200) 내의 전용 하드웨어에 의해, 하드 디스크 드라이브(205)상에 상주하고 프로세서(205)에 의한 그 실행 시에 제어되는 소프트웨어 애플리케이션 프로그램(233)의 하나 이상의 소프트웨어 코드 모듈과 같은 컴퓨터 시스템(200) 내에서 실행가능한 소프트웨어에 의해 구현될 수 있다. 대안적으로, 비디오 인코더(114) 및 비디오 디코더(134)는 전용 하드웨어와 컴퓨터 시스템(200) 내에서 실행가능한 소프트웨어의 조합에 의해 구현될 수 있다. 비디오 인코더(114), 비디오 디코더(134) 및 설명된 방법들은 설명된 방법들의 기능들 또는 서브 기능들을 수행하는 하나 이상의 집적 회로와 같은 전용 하드웨어로 대안적으로 구현될 수 있다. 그러한 전용 하드웨어는 그래픽 처리 유닛(GPU)들, 디지털 신호 프로세서(DSP)들, 주문형 표준 제품(ASSP)들, 주문형 집적 회로(ASIC)들, 필드 프로그래머블 게이트 어레이(FPGA)들 또는 하나 이상의 마이크로프로세서 및 연관된 메모리들을 포함할 수 있다. 특히, 비디오 인코더(114)는 모듈들(322-386)을 포함하고, 비디오 디코더(134)는 소프트웨어 애플리케이션 프로그램(233)의 하나 이상의 소프트웨어 코드 모듈로서 각각 구현될 수 있는 모듈들(420-496)을 포함한다.

[0039] 도 3의 비디오 인코더(114)가 VVC(versatile video coding) 비디오 인코딩 파이프라인의 일레이지만, 본 명세서에 설명된 처리 단계들을 수행하기 위해 다른 비디오 코덱들도 사용될 수 있다. 비디오 인코더(114)는 일련의 프레임들과 같은 캡처된 프레임 데이터(113)를 수신하고, 각각의 프레임은 하나 이상의 색 채널을 포함한다. 블록 파티셔너(block partitioner)(310)는 먼저 프레임 데이터(113)를 일반적으로 "코딩 트리 유닛(CTU)들"이라고 지칭되는 영역들로 분할하는데, 이는 일반적으로 형상이 정사각형이고 CTU들에 대한 특정 크기가 사용되도록 구성된다. 코딩 트리 유닛들의 크기는, 예를 들어, 64x64, 128x128, 또는 256x256 루마 샘플들일 수 있다. 블록 파티셔너(310)는 각각의 CTU를 하나 이상의 코딩 유닛(CU)으로 더 분할하며, CU들은 다양한 크기를 가지며, 이는 정사각형 및 비정사각형 중횡비를 모두를 포함할 수 있다. 따라서, 현재 블록(312), "코딩 유닛"(CU)이 블록 파티셔너(310)로부터 출력되어, CTU의 하나 이상의 블록에 걸친 반복에 따라 진행된다. 그러나, CU의 개념은 블록 파티셔너(310)로부터 생기는 블록 파티셔닝에 한정되지 않는다. 비디오 디코더(134)는 또한 손실 압축 기법들의 사용으로 인해 전형적으로 블록 파티셔너(310)로부터의 블록들의 근사인 CU들을 생성한다고 할 수 있다. 더욱이, 비디오 인코더(114)는 비디오 디코더(134)에서 보이는 것과 동일한 근사를 갖는 CU들을 생성하여, 비디오 디코더(134)에서의 블록 예측 방법들에 이용가능한 샘플 데이터에 대한 정확한 지식을 가능하게 한다. CTU들을 CU들로 파티셔닝하기 위한 옵션들은 도 5 및 도 6을 참조하여 아래에 더 설명된다.

[0040] 프레임 데이터(113)의 제1 분할로부터 생기는 코딩 트리 유닛(CTU)들은 래스터 스캔 순서로 스캐닝될 수 있고, 하나 이상의 "슬라이스"로 그룹화된다. 프레임 데이터(113)가 전형적으로 다중의 색 채널을 포함하기 때문에, CTU들 및 CU들은 블록 파티셔너(310)의 동작으로부터 정의된 블록 영역과 중첩되는 모든 색 채널들로부터의 샘플들과 연관된다. CU는 하나 이상의 코딩 블록(CB)을 포함한다고 할 수 있고, 각각의 CB는 CU와 동일한 블록 영역을 점유하지만 프레임 데이터(113)의 색 채널들 각각의 것과 연관된다. 루마 채널에 비해 크로마 채널들의 잠재적으로 상이한 샘플링 레이트로 인해, 크로마 채널들에 대한 CB들의 치수들은 루마 채널들에 대한 CB들의 치수들과 상이할 수 있다. 4:2:0 크로마 포맷을 이용할 때, CU의 크로마 채널들의 CB들은 CU의 루마 채널에 대한 CB의 폭 및 높이의 절반의 치수들을 갖는다.

[0041] 블록 파티셔너(310)로부터 생기는 모든 CU들에 걸쳐 반복함에 있어서, 비디오 인코더(114)는 각각의 블록, 예를 들어, 블록(312)에 대해 화살표(320)로 표시된 "예측 유닛(prediction unit, PU)"을 생성한다. PU(320)는 연관된 CU(312)의 콘텐츠의 예측이다. 감산기 모듈(322)은 PU(320)와 CU(312) 사이의, (324)로 표시된 차이(또는, 차이가 공간 도메인에 있는 것을 지칭하는 "잔차(residual)")를 생성한다. 차이(324)는 PU(320) 및 CU(312)에서의 대응하는 샘플들 간의 블록 크기 차이이다. 차이(324)는 변환되고, 양자화되고, 화살표(336)로 표시된 변환 유닛(transform unit, TU)으로서 표현된다. PU(320)는 전형적으로 많은 가능한 후보 PU들 중 "최상의" 결과적인 것으로서 선택된다. 후보 PU는 비디오 인코더(114)에게 이용가능한 예측 모드들 중 하나로부터 생기는 PU이다. 각각의 후보 PU는 대응하는 변환 유닛을 초래한다. 변환 유닛(336)은 차이(324)의 양자화되고 변환된 표현이다. 디코더(114)에서의 예측된 PU와 조합될 때, 변환 유닛(336)은 비트스트림에서의 추가적인 시

그널링을 희생하여 디코딩된 CU들과 원래의 블록들(312) 사이의 차이를 감소시킨다.

- [0042] 따라서, 각각의 후보 PU는 연관된 코딩 비용(레이트) 및 연관된 차이(또는 "왜곡")를 갖는다. 코딩 레이트(비용)는 전형적으로 비트로 측정된다. 블록의 코딩 왜곡은 전형적으로 SAD(sum of absolute differences) 또는 SSD(sum of squared differences)와 같은 샘플 값들에서의 차이로서 추정된다. 각각의 후보 PU로부터 생기는 추정치는 차이(324)를 이용하여 모드 선택기(386)에 의해 결정되어 (화살표(388)로 표시된) 인트라 예측 모드를 결정한다. 각각의 후보 예측 모드 및 대응하는 잔차 코딩과 연관된 코딩 비용의 추정은 잔차의 엔트로피 코딩 보다 상당히 낮은 비용으로 수행될 수 있다. 따라서, 레이트 왜곡 의미에서 최적의 모드를 결정하기 위해 다수의 후보 모드가 평가될 수 있다.
- [0043] 최적 모드를 결정하는 것은 전형적으로 라그랑주 최적화(Lagrangian optimisation)의 변동을 사용하여 달성된다. 인트라 예측 모드(388)의 선택은 전형적으로 특정 인트라 예측 모드의 적용으로부터 생기는 잔차 데이터에 대한 코딩 비용을 결정하는 것을 수반한다. 코딩 비용은 "변환된 차이들의 합"을 사용함으로써 근사화될 수 있으며, 그에 의해 하다마드(Hadamard) 변환과 같은 비교적 간단한 변환이 이용되어 추정되고 변환된 잔차 비용을 획득한다. 비교적 간단한 변환들을 사용하는 구현들의 경우, 단순화된 추정 방법으로부터 생기는 비용이 그렇지 않으면 전체 평가로부터 결정되었을 실제 비용들과 단조적으로 관련된다면, 단순화된 추정 방법은 비디오 인코더(114)에서 복잡도의 감소를 가지며 동일한 결정(즉, 인트라 예측 모드)을 행하기 위해 사용될 수 있다. 가능하게는 잔차 데이터의 코딩에 이용가능한 추가 모드 결정들로부터 생기는, 추정된 비용과 실제 비용 사이의 관계에서 가능한 비 단조성(non-monotonicity)을 허용하기 위해, 단순화된 추정 방법이 최상의 후보들의 리스트를 생성하기 위해 사용될 수 있다. 최상의 후보들의 리스트는 임의의 수의 것일 수 있다. 후보들 각각에 대한 잔차 데이터를 코딩하기 위한 최적의 모드 선택들을 확립하기 위해 최상의 후보들을 이용하여 더 완전한 검색이 수행될 수 있어서, 다른 모드 결정들과 함께 인트라 예측 모드의 최종 선택을 허용한다.
- [0044] 다른 모드 결정들은 "변환 스킵(transform skip)"으로 알려진, 1차 및 2차 변환들을 스킵하는 능력을 포함한다. 변환들을 스킵하는 것은 변환 기저 함수들로서의 표현을 통해 감소된 코딩 비용에 대한 적절한 상관을 결여하는 잔차 데이터에 적합하다. 비교적 단순한 컴퓨터 생성 그래픽과 같은 특정 타입의 콘텐츠는 유사한 거동을 나타낼 수 있다.
- [0045] 모드 선택기 모듈(386)과 연관된 또 다른 모드 결정은 명시적 다중 변환 인덱스(화살표(390)로 표시됨)의 선택이다. 명시적 다중 변환 인덱스(390)는 4개의 가능한 값을 갖는다. 이들 4개의 가능한 값은 (도 10a의 열들(1046 및 1048)에 도시된) 2개의 인덱스의 조합이며, 여기서 각각의 인덱스는 0 내지 1의 범위에 있다. 제1 인덱스는 수평 변환 인덱스 H와 관련되고 제2 인덱스는 수직 변환 인덱스 V와 관련된다. 수평 변환 인덱스 H는 수평 변환 선택으로도 알려져 있고, 수직 변환 인덱스 V는 수직 변환 선택으로도 알려져 있다. 명시적 다중 변환 인덱스(390)의 선택 및 코딩이 도 10을 참조하여 설명된다.
- [0046] 라그랑주 또는 유사한 최적화 처리는 (블록 파티셔너(310)에 의한) CTU의 CU들로의 최적의 파티셔닝을 선택할 뿐만 아니라 복수의 가능성으로부터 최상의 예측 모드를 선택하는데 채택될 수 있다. 모드 선택기 모듈(386)에서 후보 모드들의 라그랑주 최적화 프로세스의 적용을 통해, 최저 비용 측정치를 갖는 인트라 예측 모드가 최상의 모드로서 선택된다. 최상의 모드는 선택된 인트라 예측 모드(388)이고, 또한 엔트로피 인코더(338)에 의해 비트스트림(115)으로 인코딩된다. 모드 선택기 모듈(386)의 동작에 의한 인트라 예측 모드(388)의 선택은 블록 파티셔너(310)의 동작으로 확장된다. 예를 들어, 인트라 예측 모드(388)의 선택을 위한 후보는 주어진 블록에 적용가능한 모드들, 및 주어진 블록과 집합적으로 병치된 다중의 더 작은 블록에 적용가능한 추가적 모드들을 포함할 수 있다. 이러한 경우들에서, 후보들의 선택의 프로세스는 암시적으로 또한 CTU의 CU들로의 최상의 계층적 분해를 결정하는 프로세스이다.
- [0047] 엔트로피 인코더(338)는 신택스 요소의 가변 길이 코딩 및 신택스 요소의 산술 코딩 모두를 지원한다. 컨텍스트 적응 2진 산술 코딩 프로세스를 이용하여 산술 코딩이 지원된다. 산술 코딩된 신택스 요소들은 하나 이상의 "빈(bin)"의 시퀀스들로 구성된다. 빈들은 비트들과 같이 "0" 또는 "1"의 값을 갖는다. 그러나, 빈들은 비트스트림(115)에서 이산 비트들로서 인코딩되지 않는다. 빈들은 "컨텍스트"로 알려져 있는, 연관된 가능성 있는 값 및 연관된 확률을 가진다. 코딩될 실제 빈(bin)이 가능성 있는 값과 매칭될 때, "최대 확률 심볼(most probable symbol, MPS)"이 코딩된다. 최대 확률 심볼을 코딩하는 것은 소비된 비트들의 면에서 볼 때 비교적 저렴하다. 코딩될 실제 빈이 가능성 있는 값과 매칭되지 않을 때, "최소 확률 심볼(least probable symbol, LPS)"이 코딩된다. 최소 확률 심볼을 코딩하는 것은 소비된 비트들의 면에서 볼 때 비교적 높은 비용을 갖는다. 빈 코딩 기술은 "0" 대 "1"의 확률이 스큐(skew)되는 빈들의 효율적인 코딩을 가능하게 한다. 2개의

가능한 값(즉, "플래그")을 갖는 선택 요소의 경우, 단일 빈이 적절하다. 많은 가능한 값들을 갖는 선택 요소들의 경우, 빈들의 시퀀스가 필요하다. 그 후, 시퀀스에서의 더 이른 빈들의 값에 기초하여 시퀀스에서의 더 나중의 빈들의 존재가 결정될 수 있다. 또한, 각각의 빈은 하나보다 많은 컨텍스트와 연관될 수 있고, 특정한 컨텍스트의 선택은 선택 요소에서의 더 이른 빈들, 이웃 선택 요소들(즉, 이웃 블록들로부터의 것들)의 빈 값들 등에 의존한다. 빈이 코딩될 때마다, 컨텍스트는 새로운 빈 값에 적응하도록 업데이트된다. 이와 같이, 2진 산술 코딩 방식은 적응적이라고 말한다.

[0048] 또한, 컨텍스트를 결여한 빈들("바이패스 빈들")이 인코더(114)에 의해 지원된다. 바이패스 빈들은 "0"과 "1" 사이의 등가 확률 분포(equiprobable distribution)를 가정하여 코딩된다. 컨텍스트의 부재는 메모리를 절약하고 복잡성을 감소시키므로, 특정한 빈에 대한 값들의 분포가 스쿼되지 않는 경우에 바이패스 빈들이 사용된다. 컨텍스트 및 적응을 채택하는 엔트로피 코더의 일례는 CABAC(context adaptive binary arithmetic coder)로서 본 기술 분야에 공지되어 있고 이 코더의 많은 변형들이 비디오 코딩에서 채택되었다.

[0049] 엔트로피 인코더(338)는 컨텍스트 코딩된 빈 및 (선택적으로) 바이패스 코딩된 빈의 조합을 사용하여 인트라 예측 모드(388)를 인코딩한다. 전형적으로, "최대 확률 모드들"의 리스트가 비디오 인코더(114)에서 생성된다. 최대 확률 모드들의 리스트는 전형적으로 3개 또는 6개의 모드와 같은 고정된 길이를 갖고, 더 이른 블록들에서 마주치는 모드들을 포함할 수 있다. 컨텍스트 코딩된 빈은 인트라 예측 모드가 최대 확률 모드들 중 하나인지를 나타내는 플래그를 인코딩한다. 인트라 예측 모드(388)가 최대 확률 모드들 중 하나인 경우, 예를 들어, 절단된 단항 빈 스트링(truncated unary bin string)을 사용하여, 어느 최대 확률 모드가 인트라 예측 모드(388)에 대응하는지를 나타내는 추가 시그널링이 인코딩된다. 그렇지 않으면, 인트라 예측 모드(388)는 최대 확률 모드 리스트에 존재하는 것들 이외의 인트라 예측 모드들을 표현하기 위해, 고정 길이 코드와 같은 대안 선택을 이용하여, "잔여 모드"로서 인코딩된다.

[0050] 엔트로피 인코더(338)는 또한 도 10을 참조하여 설명된 바와 같이 특정 코딩 유닛들 또는 변환 블록들에 대한 명시적인 다중 변환 인덱스(390)를 인코딩한다. 특히, 명시적인 다중 변환 인덱스(390)는 수직 변환 인덱스 V 및 수평 변환 인덱스 H를 포함한다. 수직 변환 인덱스 V 및 수평 변환 인덱스 H 둘 모두는 (위에 설명된 바와 같이) 모드 선택기(386)에 의해 선택된다. 수직 변환 인덱스 V와 수평 변환 인덱스 H의 조합에 기초하여(도 10a 참조), 엔트로피 인코더(338)는 대응하는 빈 스트링(도 10a의 열(1042)에 도시됨)을 비트스트림(115)으로 인코딩한다.

[0051] 멀티플렉서 모듈(384)은 테스트된 후보 예측 모드들로부터 선택하는, 결정된 최상의 인트라 예측 모드(388)에 따라 PU(320)를 출력한다. 후보 예측 모드들은 비디오 인코더(114)에 의해 지원되는 모든 생각 가능한 예측 모드를 포함할 필요는 없다. 예측 모드들은 대체로 2가지 범주로 나누어진다. 제1 범주는 "인트라 프레임 예측"(또는 "인트라 예측")이다. 인트라 프레임 예측에서, 현재 프레임으로부터 도출된 다른 샘플들을 사용하여 블록에 대한 예측이 생성된다. 제2 범주는 "인터 프레임 예측"(또는 "인터 예측")이다. 인트라 프레임 예측에서, 블록에 대한 예측은 (캡처되거나 디스플레이될 때 프레임들의 순서와 상이할 수 있는) 비트스트림에서의 코딩 프레임들의 순서로 현재 프레임에 선행하는 프레임으로부터의 샘플들을 사용하여 산출된다. 각각의 범주 내에서(즉, 인트라 예측 및 인터 예측), 상이한 기법들이 적용되어 PU를 생성할 수 있다. 예를 들어, 인트라 예측은 규정된 필터링 프로세스에 따라 PU를 생성하는 방향과 조합하여, 이전에 재구성된 샘플들의 인접한 행들 및 열들로부터의 값들을 사용할 수 있다. 대안적으로, PU는 적은 수의 파라미터를 사용하여 기술될 수 있다. 인터 예측 방법들은 움직임 파라미터들의 수 및 그들의 정밀도에 있어서 변할 수 있다. 움직임 파라미터들은 전형적으로 참조 프레임 오프셋 더하기 1개 또는 2개의 참조 프레임에 대한 변환(translation)을 포함하지만, 더 많은 프레임들, 특수 프레임들, 또는 스케일링 및 회전과 같은 복잡한 아핀 파라미터들을 포함할 수 있다. 그에 부가하여, 참조된 샘플 블록들에 기초하여 조밀한 움직임 추정치들을 생성하기 위해 미리 결정된 움직임 리파인먼트(refinement) 프로세스가 적용될 수 있다.

[0052] 최상의 PU(320)를 결정하고 선택하고, 감산기(322)에서의 원래의 샘플 블록으로부터 PU(320)를 감산하면, 최저 코딩 비용(324)을 갖는 잔차가 획득되고 손실 압축을 거친다. 손실 압축 프로세스는 변환, 양자화 및 엔트로피 코딩의 단계들을 포함한다. 변환 모듈(326)은 차이(324)에 제1 변환을 적용하고, 차이(324)를 주파수 도메인으로 변환하고, 화살표(328)로 나타낸 중간 변환 계수들을 생성한다. 제1 변환은 전형적으로 분리가능하여, 각각의 블록의 행들의 세트 및 이어서 열들의 세트를 변환한다. 행들 및 열들의 각각의 세트의 변환은 블록의 각각의 행 및 열에서 반복된 일차원 변환들을 적용함으로써 수행된다. 변환은 전체 블록에 걸쳐 있다.

[0053] 특히, 인트라 예측 모드(388) 및 명시적 다중 변환 인덱스(390)는 변환 모듈(326)에 의해 수행되는 1차 변환을

집합적으로 형성하는, 수평 일차원 변환 및 수직 일차원 변환을 선택하기 위해 사용된다. 수평 1차원 변환 및 수직 1차원 변환 각각은 분리가능 변환이다. 수평 1차원 변환 및 수직 1차원 변환은 전체 변환 블록에 걸쳐 수행되어, 잔차 샘플들을 실질적으로 상관 해제시킨다.

[0054] 변환 모듈(326)에 대한 이용가능한 변환 타입들은 DST-VII, DST-VIII, DCT-VIII, DST-I, 및 DCT-V를 포함한다. 이 변환들의 세트로부터, 도 8b를 참조하여 설명된 바와 같이, 3개의 "변환 세트"(842)가 정의된다. 수평 또는 수직 방향에서의 각각의 방향에 대해, 변환 세트(842)가, 도 8c를 참조하여 설명된 바와 같이, 인트라 예측 모드(388)에 따라 선택된다. 각각의 변환 세트(842)로부터, 도 11을 참조하여 설명된 바와 같이, 하나의 변환 타입이 선택되고 적용된다. 중간 변환 계수들(328)은 2차 변환 모듈(330)에 전달된다.

[0055] 2차 변환 모듈(330)은 전체 블록의 좌측 상부 4x4 또는 8x8 영역을 점유하는 중간 변환 계수들과 같은 중간 변환 계수들(328)의 서브세트에 대해 동작한다. 중간 변환 계수들(328)에서의 다른 변환 계수들은 변경되지 않은 채로 모듈(330)을 통과한다. 2차 변환 모듈(330)은 중간 변환 계수들(328)의 서브세트에 다양한 변환들 중 하나를 적용하여 화살표(332)로 표현된 변환 계수들을 산출한다. 2차 변환 모듈(330)은 역 2차 변환 모듈(344)의 방식과 유사한 방식으로 선택된 순방향 2차 변환을 적용한다. 2차 변환 모듈(330)에 이용가능한 변환들은 전형적으로 분리가능하지 않으며, 따라서 변환 모듈(326)에 대한 경우에서와 같이 2개의 스테이지(즉, 행들 및 열들)에서 수행될 수 없다. 2차 변환의 더 높은 복잡도, 및 1차 변환의 더 이른 상관 해제로 인해, 변환 블록의 좌측 상단 영역에 대해서만 2차 변환을 수행하는 것으로 충분하다. 2차 변환 모듈(330)에 의해 수행되는 변환은 예측 모드에 적어도 부분적으로 의존하여 선택될 수 있다. 또한, 비디오 인코더(114)는 레이트/왜곡 비용 평가에 기초한 선택을 위한 추가 후보들의 테스트로서 모듈(330)에서의 적용된 변환의 선택을 고려할 수 있다.

[0056] 변환 계수들(332)은 양자화기 모듈(334)에 전달된다. 모듈(334)에서, "양자화 파라미터"에 따른 양자화가 수행되어 화살표(336)로 나타낸 잔차 계수들을 산출한다. 양자화 파라미터는 주어진 변환 블록에 대해 일정하며, 따라서 변환 블록에 대한 잔차 계수들의 산출을 위한 균일한 스케일링이라는 결과를 낳는다. 또한, "양자화 행렬"의 적용에 의해 불균일한 스케일링이 가능하며, 그에 의해, 양자화 파라미터 및 전형적으로 변환 블록의 크기와 동일한 크기를 갖는 스케일링 행렬에서의 대응하는 엔트리의 조합으로부터 각각의 잔차 계수에 대해 적용되는 스케일링 인자가 도출된다. 양자화 행렬은 시그널링하는데 비용이 많이 들고 따라서 비트스트림(115)에서 (만일 있다면) 드물게만 코딩된다. 양자화 행렬의 코딩은 스케일링 인자들의 2차원 행렬을 비트스트림(115)으로 엔트로피 인코딩될 스케일링 인자들의 리스트로 변환할 것을 요구한다. 기존의 Z-오더 스캔(Z-order scan)은 이 목적을 위해 재사용될 수 있어서, 양자화 행렬을 인코딩하는 드물게 수행되는 동작에 대한 부가의 스캔 패턴을 지원하는 것과 연관된 오버헤드를 회피할 수 있다. 잔차 계수들(336)은 비트스트림(115)에서의 인코딩을 위해 엔트로피 인코더(338)에 공급된다. 전형적으로, 변환 블록의 잔차 계수들은 스캔 패턴에 따라 순서화된 값들의 리스트를 산출하기 위해 스케닝된다. 스캔 패턴은 일반적으로 변환 블록을 4x4 "서브 블록들"의 시퀀스로서 스케닝하여, 서브 블록들의 배열이 변환 블록의 크기에 의존하면서, 잔차 계수들의 4x4 세트들의 세분성(granularity)으로 정규 스케닝 동작을 제공한다. 추가적으로, 예측 모드 및 대응하는 블록 파티셔닝은 또한 비트스트림(115)에서 인코딩된다. 전술한 바와 같이, 비디오 인코더(114)는 비디오 디코더(134)에서 보이는 프레임 표현에 대응하는 프레임 표현에 액세스할 필요가 있다. 따라서, 잔차 계수들(336)은 또한 양자화 해제기 모듈(340)에 의해 역양자화되어, 화살표(342)로 나타낸 역 변환 계수들을 산출한다. 역 변환 계수들(342)은 역 2차 변환 모듈(344)을 통과한다. 역 2차 변환 모듈(344)은 선택된 2차 변환을 적용하여 화살표(346)로 나타낸 바와 같이 중간 역 변환 계수들을 산출한다. 중간 역변환 계수들(346)은 역변환 모듈(348)에 공급되어, 변환 유닛의, 화살표(350)로 나타낸 잔차 샘플들을 산출한다. 합산 모듈(352)은 잔차 샘플들(350) 및 PU(320)를 합산하여 CU의 재구성된 샘플들(화살표(354)로 표시됨)을 산출한다. 재구성된 샘플(354)은 참조 샘플 캐시(356) 및 인 루프(in-loop) 필터 모듈(368)로 전달된다. ASIC상의 정적 RAM을 사용하여 전형적으로 구현되는 참조 샘플 캐시(356)(따라서 고가의 오프 칩 메모리 액세스를 회피함)는 프레임에서의 후속 CU들에 대한 인트라 프레임 예측 블록들을 생성하기 위한 종속성들을 충족시키는데 필요한 최소 샘플 저장을 제공한다. 최소 종속성들은 CTU들의 다음 행에 의한 사용 및 CTU의 높이에 의해 그 범위가 설정되는 열 버퍼링을 위해, CTU들의 행의 하단을 따른 샘플들의 "라인 버퍼"를 전형적으로 포함한다. 참조 샘플 캐시(356)는 참조 샘플들(화살표(358)로 표현됨)을 참조 샘플 필터(360)에 공급한다. 샘플 필터(360)는 평활화 연산을 적용하여 필터링된 참조 샘플들(화살표(362)로 표시됨)을 산출한다. 필터링된 참조 샘플들(362)은 인트라 프레임 예측 모듈(364)에 의해 사용되어 화살표(366)로 표현된, 샘플들의 인트라 예측된 블록을 산출한다. 각각의 후보 인트라 예측 모드에 대해, 인트라 프레임 예측 모듈(364)은 샘플들의 블록, 즉 (366)을 생성한다.

- [0057] 인 루프 필터 모듈(368)은 재구성된 샘플들(354)에 몇 개의 필터링 스테이지를 적용한다. 필터링 스테이지들은 불연속성에 기인하는 아티팩트를 감소시키기 위해 CU 경계들에 정렬된 평활화를 적용하는 "디블로킹 필터(deblocking filter, DBF)"를 포함한다. 인 루프 필터 모듈(368)에 존재하는 또 다른 필터링 스테이지는 "적응성 루프 필터(adaptive loop filter, ALF)"이고, 이것은 위너(Wiener) 기반 적응성 필터를 적용하여 왜곡을 더 감소시킨다. 인 루프 필터 모듈(368)에서의 추가의 이용가능한 필터링 스테이지는 "SAO(sample adaptive offset)" 필터이다. SAO 필터는 먼저 재구성된 샘플들을 하나 또는 다중의 범주로 분류하고, 할당된 범주에 따라 샘플 레벨에서 오프셋을 적용함으로써 동작한다. 필터링된 샘플들(370)은 인 루프 필터들 모듈(368)로부터 출력된다. 필터링된 샘플들(370)은 프레임 버퍼(372)에 저장된다. 프레임 버퍼(372)는 전형적으로 여러 개(예를 들어, 최대 16개)의 픽처를 저장하는 용량을 가지며, 따라서 메모리(206)에 저장된다. 이와 같이, 프레임 버퍼(372)에의 액세스는 메모리 대역폭의 면에서 볼 때 비용이 많이 든다. 프레임 버퍼(372)는 참조 프레임들(화살표 374로 표현됨)을 움직임 추정 모듈(376) 및 움직임 보상 모듈(380)에 제공한다.
- [0058] 움직임 추정 모듈(376)은 다수의 "움직임 벡터"(378로 표시됨)를 추정하며, 각각은 프레임 버퍼(372)에서의 기준 프레임들 중 하나에서의 블록을 참조하는, 현재 CU의 위치로부터의 데카르트 공간 오프셋이다. (382로 표현되는) 참조 샘플들의 필터링된 블록이 각각의 움직임 벡터에 대해 산출된다. 필터링된 참조 샘플들(382)은 모드 선택기(386)에 의한 잠재적 선택을 위해 이용가능한 추가 후보 모드들을 형성한다. 더욱이, 주어진 CU에 대해, PU(320)는 하나의 참조 블록을 사용하여 형성될 수 있거나("단방향 예측") 또는 2개의 참조 블록을 사용하여 형성될 수 있다("양방향 예측"). 선택된 움직임 벡터에 대해, 움직임 보상 모듈(380)은 움직임 벡터들에서 서브 픽셀 정확도를 지원하는 필터링 프로세스에 따라 PU(320)를 산출한다. 이와 같이, 움직임 추정 모듈(376)(이것은 많은 후보 움직임 벡터들에 대해 동작함)은 생각건대 감소된 계산 복잡성을 달성하기 위해 움직임 보상 모듈(380)(이것은 선택된 후보에 대해서만 동작함)의 것과 비교하여 단순화된 필터링 프로세스를 수행할 수 있다.
- [0059] 도 3의 비디오 인코더(114)가 VVC(versatile video coding)를 참조하여 기술되어 있지만, 다른 비디오 코딩 표준 또는 구현이 또한 모듈들(310 내지 386)의 처리 스테이지들을 채택할 수 있다. 프레임 데이터(113)(및 비트스트림(115))는 또한 메모리(206), 하드 디스크 드라이브(210), CD-ROM, Blu-ray disk™ 또는 다른 컴퓨터 판독가능 저장 매체로부터 판독(또는 그것에 기입)될 수 있다. 또한, 프레임 데이터(113)(및 비트스트림(115))는 통신 네트워크(220) 또는 무선 주파수 수신기에 접속된 서버와 같은 외부 소스로부터 수신(또는 그것에 송신)될 수 있다.
- [0060] 비디오 디코더(134)는 도 4에 도시되어 있다. 도 4의 비디오 디코더(134)가 VVC(versatile video coding) 비디오 인코딩 파이프라인의 일레이지만, 본 명세서에 기술된 처리 스테이지들을 수행하기 위해 다른 비디오 코덱들도 사용될 수 있다. 도 4에서 알 수 있는 바와 같이, 비트스트림(133)은 비디오 디코더(134)에 입력된다. 비트스트림(133)은 메모리(206), 하드 디스크 드라이브(210), CD-ROM, Blu-ray disk™ 또는 다른 비일시적 컴퓨터 판독가능 저장 매체로부터 판독될 수 있다. 대안적으로, 비트스트림(133)은 통신 네트워크(220) 또는 무선 주파수 수신기에 접속된 서버와 같은 외부 소스로부터 수신될 수 있다. 비트스트림(133)은 디코딩될 캡처된 프레임 데이터를 나타내는 인코딩된 선택스 요소들을 포함한다.
- [0061] 비트스트림(133)은 엔트로피 디코더 모듈(420)에 입력된다. 엔트로피 디코더 모듈(420)은 비트스트림(133)으로부터 선택스 요소들을 추출하고 선택스 요소들의 값들을 비디오 디코더(134)에서의 다른 모듈들에 전달한다. 엔트로피 디코더 모듈(420)은 CABAC 알고리즘을 적용하여 비트스트림(133)으로부터 선택스 요소들을 디코딩한다. 디코딩된 선택스 요소들은 비디오 디코더(134) 내의 파라미터들을 재구성하기 위해 사용된다. 파라미터들은 잔차 계수들(화살표(424)로 표현됨) 및 인트라 예측 모드(458)와 명시적 다중 변환 인덱스(454)와 같은 모드 선택 정보를 포함한다. 모드 선택 정보는 또한 움직임 벡터들, 및 각각의 CTU의 하나 이상의 CU로의 파티셔닝과 같은 정보를 포함한다. 파라미터들은, 전형적으로 이전에 디코딩된 CU들로부터의 샘플 데이터와 조합하여, PU들을 생성하는데 이용된다.
- [0062] 비트스트림(133)은 또한 빈 스트링 열(1042)(도 10a 참조)로부터의 빈 스트링을 포함한다. 엔트로피 디코더 모듈(420)은 빈 스트링을 추출하고 (수직 변환 세트 인덱스 열(1046)로부터의) 수직 변환 세트 인덱스 및 (수평 변환 세트 인덱스 열(1048)로부터의) 수평 변환 세트 인덱스를 산출한다.
- [0063] 잔차 계수들(424)은 양자화 해제기 모듈(428)에 입력된다. 양자화 해제기 모듈(428)은 잔차 계수들(424)에 대해 역 스케일링을 수행하여 양자화 파라미터에 따라 재구성된 중간 변환 계수들(화살표(432)로 표현됨)을 생성한다. 비균일 역 양자화 행렬을 사용하는 것이 비트스트림(133)에 표시된다면, 비디오 디코더(134)는 비트스트

림(133)으로부터의 양자화 행렬을 스케일링 인자들의 시퀀스로서 관독하고, 잔차 계수들의 코딩에 사용되는 Z-오더 스캔에 따라 스케일링 인자들을 행렬로 배열한다. 그 후, 역 스케일링은 양자화 파라미터와 조합하여 양자화 행렬을 사용하여 재구성된 중간 변환 계수들을 생성한다. 또한 양자화 행렬 스케일링 인자들을 스케일링하기 위한 잔차 계수들에 대한 Z-오더 스캔의 이용은 추가적인 스캔 패턴 및 연관된 메모리의 존재 및 드물게 수행되는 스캔에 대한 복잡도 부담을 회피한다. 재구성된 중간 변환 계수들(432)은 역 2차 변환 모듈(436)에 전달된다. 역 2차 모듈(436)은 화살표(440)로 표현된, 재구성된 변환 계수들을 산출하기 위해 "2차 역 변환"을 수행한다. 2차 변환은 결정된 변환 블록 크기에 따라 수행된다. 재구성된 변환 계수들(440)은 도 9를 참조하여 설명된 바와 같이 역 변환 모듈(444)에 전달된다. 모듈(444)은 계수들을 주파수 도메인으로부터 공간 도메인으로 되돌려 변환한다. 변환 블록은 유의 잔차 계수들(significant residual coefficients) 및 비 유의 잔차 계수 값들에 실효적으로 기초한다. 모듈(444)의 동작의 결과는 화살표(448)로 표현된 잔차 샘플들의 블록이다. 잔차 샘플들(448)은 대응하는 CU와 크기가 동일하다. 잔차 샘플들(448)은 합산 모듈(450)에 공급된다. 합산 모듈(450)에서, 잔차 샘플들(448)은 디코딩된 PU(452)에 합산되어, 화살표(456)로 표현된 재구성된 샘플들의 블록을 산출한다. 재구성된 샘플들(456)은 재구성된 샘플 캐시(460) 및 인 루프 필터링 모듈(488)에 공급된다. 인 루프 필터링 모듈(488)은 (492)로 표현되는 프레임 샘플들의 재구성된 블록들을 산출한다. 프레임 샘플들(492)은 프레임 버퍼(496)에 기입된다.

[0064] 재구성된 샘플 캐시(460)는 비디오 인코더(114)의 재구성된 샘플 캐시(356)와 유사하게 동작한다. 재구성된 샘플 캐시(460)는 (예를 들어, 전형적으로 온 칩 메모리인, 대신에 데이터(232)를 사용함으로써) 메모리(206) 없이 후속 CU들을 인트라 예측하는 데 필요한 재구성된 샘플에 대한 스토리지를 제공한다. 화살표(464)로 표현된 참조 샘플들은 재구성된 샘플 캐시(460)로부터 획득되고 참조 샘플 필터(468)에 공급되어 화살표(472)로 표시된 필터링된 참조 샘플들을 산출한다. 필터링된 참조 샘플들(472)은 인트라 프레임 예측 모듈(476)에 공급된다. 모듈(476)은, 비트스트림(133)에서 시그널링되고 엔트로피 디코더(420)에 의해 디코딩되는 인트라 예측 모드 파라미터(458)에 따라, 화살표(480)로 표현된 인트라 예측된 샘플들의 블록을 산출한다.

[0065] 인트라 예측이 현재 CU에 대해 비트스트림(133)에서 표시될 때, 인트라 예측된 샘플들(480)은 멀티플렉서 모듈(484)을 통해 디코딩된 PU(452)를 형성한다.

[0066] 인트라 예측이 현재 CU에 대해 비트스트림(133)에서 표시될 때, 움직임 보상 모듈(434)은 프레임 버퍼(496)로부터 샘플들의 블록을 선택하고 필터링하기 위해 움직임 벡터 및 참조 프레임 인덱스를 사용하여 인트라 예측된 샘플들의 블록(438)을 산출한다. 샘플들의 블록(498)은 프레임 버퍼(496)에 저장된 이전에 디코딩된 프레임으로부터 획득된다. 양방향 예측의 경우, 디코딩된 PU(452)에 대한 샘플들을 산출하기 위해 샘플들의 2개의 블록이 산출되고 함께 혼합된다. 프레임 버퍼(496)는 인 루프 필터링 모듈(488)로부터의 필터링된 블록 데이터(492)로 채워진다. 비디오 인코더(114)의 인 루프 필터링 모듈(368)에서와 같이, 인 루프 필터링 모듈(488)은 DBF, ALF 및 SAO 필터링 동작들 중 임의의 것, 적어도, 또는 전부를 적용한다. 인 루프 필터링 모듈(368)은 재구성된 샘플들(456)로부터 필터링된 블록 데이터(492)를 산출한다.

[0067] 도 5는 VVC(versatile video coding)의 트리 구조에서 블록의 하나 이상의 블록으로의 이용가능한 분할 또는 스플릿의 컬렉션(500)을 도시한 개략 블록도이다. 도 3을 참조하여 설명한 바와 같이, 컬렉션(500)에 도시된 분할들은 각각의 CTU를 라그랑주 최적화에 따라 하나 이상의 CU로 분할하기 위해 인코더(114)의 블록 파티셔너(310)에게 이용가능하다.

[0068] 컬렉션(500)은 정사각형 블록들이 다른, 가능하게는 비 정사각형 블록들로 분할되는 것만을 도시하지만, 도면(500)은 잠재적 분할들을 도시하고 포함 블록을 정사각형으로 제약하지는 않음을 이해해야 한다. 포함 블록이 비 정사각형인 경우, 분할로부터 생기는 블록들의 치수들은 포함 블록의 중첩비에 따라 스케일링된다. 블록 파티셔너(310)에 의한 하나 이상의 CU로의 CTU의 특정 세분(subdivision)은 CTU의 "코딩 트리"로 지칭된다. 본 개시내용의 맥락에서, 리프 노드(leaf node)는 세분 프로세스가 종료되는 노드이다. 세분 프로세스는 리프 노드에 대응하는 영역이 최소 코딩 유닛 크기와 동일할 때 종료되어야 한다. 최소 크기의 코딩 유닛이라는 결과를 낳는 리프 노드들은 코딩 트리의 가장 깊은 분해 레벨에 존재한다. 세분 프로세스는 또한 가장 깊은 분해 레벨 이전에 종료되어, 코딩 유닛들이 최소 코딩 유닛 크기보다 큰 결과를 낳는다.

[0069] 코딩 트리의 리프 노드들에는 더 이상의 세분 없이 CU들이 존재한다. 예를 들어, 리프 노드(510)는 하나의 CU를 포함한다. 코딩 트리의 논-리프(non-leaf) 노드들에는 2개 이상의 추가 노드로의 스플릿이 존재하며, 이들 각각은 하나의 CU를 포함하거나 또는 더 작은 영역들로의 추가 스플릿을 포함할 수 있다.

[0070] 쿼드-트리 스플릿(512)은 도 5에 도시된 바와 같이 포함 영역(containing region)을 4개의 동일 크기 영역으로

분할한다. HEVC와 비교하여, VVC는 수평 이진 스플릿(514) 및 수직 이진 스플릿(516)의 추가에 의해 추가적인 유연성을 달성한다. 스플릿들(514, 516) 각각은 포함 영역을 2개의 동일 크기 영역으로 분할한다. 분할은 포함 블록 내의 수평 경계(514) 또는 수직 경계(516)를 따른다.

- [0071] 3진(ternary) 수평 스플릿(518) 및 3진 수직 스플릿(520)의 추가에 의해 VVC에서 추가의 유연성이 달성된다. 3진 스플릿들(518 및 520)은 블록을, 포함 영역의 폭 또는 높이의 1/4 및 3/4를 따라 수평으로(518) 또는 수직으로(520) 경계를 이루는 3개의 영역으로 분할한다. 쿼드 트리, 2진 트리, 및 3진 트리의 조합은 "QTBT" 또는 대안적으로 MT(multi-tree)로서 지칭된다.
- [0072] 쿼드 트리만을 지원하고 따라서 정사각형 블록들만을 지원하는 HEVC와 비교하여, QTBT는 특히 2진 트리 및/또는 3진 트리 스플릿들의 가능한 재귀적 적용을 고려하면, 더 많은 가능한 CU 크기들이라는 결과를 낳는다. 특이한 (예를 들어, 비 정사각형) 블록 크기들에 대한 잠재성은 블록 폭 또는 높이가 4개 미만의 샘플이거나 또는 4개 샘플의 배수가 아닌 결과를 낳는 스플릿들을 제거하기 위해 스플릿 옵션들을 제약함으로써 감소될 수 있다. 일반적으로, 제약은 루마 샘플들을 고려하는데 적용될 것이다. 그러나, 제약은 또한 크로마 채널들에 대한 블록들에 개별적으로 적용될 수 있고, 예를 들어 프레임 데이터가 4:2:0 크로마 포맷일 때, 잠재적으로 루마 대 크로마에 대한 상이한 최소 블록 크기들이라는 결과를 낳는다.
- [0073] 도 6은 VVC에서 사용되는 QTBT(또는 "코딩 트리") 구조의 데이터 흐름(600)을 예시하는 개략적인 흐름도이다. QTBT 구조는 CTU의 하나 이상의 CU로의 분할을 정의하기 위해 각각의 CTU에 대해 사용된다. 각각의 CTU의 QTBT 구조는 비디오 인코더(114)에서의 블록 파티셔닝(310)에 의해 결정되고, 비트스트림(115)으로 인코딩되거나 또는 비디오 디코더(134)에서의 엔트로피 디코더(420)에 의해 비트스트림(133)으로부터 디코딩된다. 데이터 흐름(600)은, 도 5에 도시된 분할들에 따라, CTU를 하나 이상의 CU로 분할하기 위해 블록 파티셔닝(310)가 이용할 수 있는 허용가능한 조합들을 더 특징으로 한다.
- [0074] 계층의 최상위 레벨, 즉 CTU에서 시작하여, 0개 이상의 쿼드-트리 분할이 먼저 수행된다. 구체적으로, 블록 파티셔닝(310)에 의해 쿼드-트리(QT) 스플릿 결정(610)이 이루어진다. "1" 심볼을 반환하는 (610)에서의 결정은 쿼드-트리 스플릿(512)에 따라 현재 노드를 4개의 서브 노드로 스플릿하라는 결정을 나타낸다. 그 결과는, (620)에서와 같이, 4개의 새로운 노드의 생성이고, 각각의 새로운 노드에 대해, QT 스플릿 결정(610)으로 회귀한다. 각각의 새로운 노드는 래스터 (또는 Z-스캔) 순서로 고려된다. 대안적으로, QT 스플릿 결정(610)이 어떤 추가 스플릿도 수행되지 않을 것임을 나타내는 경우("0" 심볼을 반환함), 쿼드-트리 파티셔닝이 중단되고 멀티-트리(MT) 스플릿들이 후속하여 고려된다.
- [0075] 먼저, 블록 파티셔닝(310)에 의해 MT 스플릿 결정(612)이 행해진다. (612)에서, MT 스플릿을 수행하라는 결정이 표시된다. 결정(612)에서 "0" 심볼을 반환하는 것은, 노드를 서브 노드들로 추가로 스플릿하는 것이 더 이상 수행되지 않을 것임을 나타낸다. 노드의 추가 스플릿이 수행되지 않는 경우, 노드는 코딩 트리의 리프 노드이고 코딩 유닛(CU)에 대응한다. (622)에서, 리프 노드가 출력된다. 대안적으로, MT 스플릿(612)이 MT 스플릿을 수행하라는 결정을 나타내면("1" 심볼을 반환하면), 블록 파티셔닝(310)은 방향 결정(614)으로 진행한다.
- [0076] 방향 결정(614)은 수평("H" 또는 "0") 또는 수직("V" 또는 "1")으로서 스플릿된 MT의 방향을 나타낸다. 블록 파티셔닝(310)은 결정(614)이 수평 방향을 나타내는 "0"을 반환하는 경우, 결정(616)으로 진행한다. 블록 파티셔닝(310)은 결정(614)이 수직 방향을 나타내는 "1"을 반환하는 경우, 결정(618)으로 진행한다.
- [0077] 결정들(616 및 618) 각각에서, MT 스플릿을 위한 파티션의 수는 BT/TT 스플릿에서 2개(이진 스플릿 또는 "BT" 노드) 또는 3개(삼진 스플릿 또는 "TT")로서 표시된다. 즉, BT/TT 스플릿 결정(616)은 (614)로부터의 표시된 방향이 수평일 때 블록 파티셔닝(310)에 의해 이루어지고 BT/TT 스플릿 결정(618)은 (614)로부터의 표시된 방향이 수직일 때 블록 파티셔닝(310)에 의해 이루어진다.
- [0078] BT/TT 스플릿 결정(616)은 수평 스플릿이 "0"을 반환함으로써 표시되는 2진 스플릿(514)인지 "1"을 반환함으로써 표시되는 3진 스플릿(518)인지를 표시한다. BT/TT 스플릿 결정(616)이 2진 스플릿을 나타내는 경우, HBT CTU 노드 생성 단계(625)에서 2개의 노드가 2진 수평 스플릿(514)에 따라 블록 파티셔닝(310)에 의해 생성된다. BT/TT 스플릿(616)이 3진 스플릿을 나타내는 경우, HTT CTU 노드 생성 단계(626)에서 3개의 노드가 3진 수평 스플릿(518)에 따라 블록 파티셔닝(310)에 의해 생성된다.
- [0079] BT/TT 스플릿 결정(618)은 수직 스플릿이 "0"을 반환함으로써 표시되는 2진 스플릿(516)인지 "1"을 반환함으로써 표시되는 3진 스플릿(520)인지를 표시한다. BT/TT 스플릿(618)이 2진 스플릿을 나타내는 경우, VBT CTU 노드 생성 단계(627)에서, 수직 2진 스플릿(516)에 따라 2개의 노드가 블록 파티셔닝(310)에 의해 생성된다.

BT/TT 스플릿(618)이 3진 분할을 나타내는 경우, VTT CTU 노드 생성 단계(628)에서 3개의 노드가 수직 3진 스플릿(520)에 따라 블록 파티셔너(310)에 의해 생성된다. 단계들(625-628)로부터 생기는 각각의 노드에 대해, MT 스플릿 결정(612)으로의 데이터 흐름(600)의 회귀는 방향(614)에 의존하여 좌측 대 우측 또는 상단 대 하단 순서로 적용된다. 그 결과, 2진 트리 및 3진 트리 스플릿들이 다양한 크기들을 갖는 CU들을 생성하기 위해 적용될 수 있다.

[0080] 도 7a 및 도 7b는 CTU(710)의 다수의 코딩 유닛(CU)으로의 예시적인 분할(700)을 제공한다. 예시적인 CU(712)가 도 7a에 도시되어 있다. 도 7a는 CTU(710)에서의 CU들의 공간 배열을 도시한다. 예시적인 분할(700)은 또한 도 7b에서 코딩 트리(720)로서 도시되어 있다.

[0081] 도 7a의 CTU(710)에서의 각각의 논-리프 노드, 예를 들어, 노드들(714, 716 및 718)에서, (추가로 분할될 수 있거나 CU들일 수 있는) 포함된 노드들은 코딩 트리(720)에서의 열들로서 표현되는 노드들의 리스트들을 생성하기 위해 "Z-오더"로 스캐닝 또는 순회(traverse)된다. 쿼드-트리 스플릿의 경우, Z-오더 스캐닝은 좌측 상단에서 우측으로 이어서 좌측 하단에서 우측으로의 순서로 귀결된다. 수평 및 수직 스플릿들에 대해, Z-오더 스캐닝(순회)은 제각기 상단에서 하단으로 그리고 좌측에서 우측으로의 스캐닝으로 단순화된다. 도 7b의 코딩 트리(720)는 적용된 스캔 순서에 따라 모든 노드들 및 CU들을 열거한다. 각각의 스플릿은 리프 노드(CU)에 도달할 때까지 트리의 다음 레벨에서 2, 3 또는 4개의 새로운 노드의 리스트를 생성한다.

[0082] 블록 파티셔너(310)에 의해 이미지를 CTU들로 분해하고 추가로 CU들로 분해하고, 또한 도 3을 참조하여 설명된 바와 같이 CU들을 이용하여 각각의 잔차 블록(324)을 생성하였으므로, 잔차 블록들은 인코더(114)에 의해 순방향 변환 및 양자화를 겪게 된다. 결과적인 변환 블록들(TB들)(336)은 엔트로피 코딩 모듈(338)의 동작의 일부로서 잔차 계수들의 순차적 리스트를 형성하도록 후속적으로 스캐닝된다. 비트스트림(133)으로부터 변환 블록들을 획득하기 위해 비디오 디코더(134)에서 동등한 프로세스가 수행된다.

[0083] 도 8a는 인트라 예측 모드들(388 및 458)을 이용하여 표시될 수 있는 변환 블록에 대한 인트라 예측 모드들의 세트(800)를 도시한다. 도 8a에서, 67개의 인트라 예측 모드가 정의된다. 모드 0은 "평면" 인트라 예측 모드이고, 모드 1은 "DC" 인트라 예측 모드이며, 모드들 2-66은 "각도" 인트라 예측 모드이다. 평면 인트라 예측 모드(모드 0)는 예측 블록을 평면에 따른 샘플들, 즉 오프셋 및 수평으로 및 수직으로 기울기를 갖는 샘플들로 채운다. 평면 파라미터들은 이용가능한 경우 이웃 참조 샘플들로부터 획득된다. 마찬가지로, DC 인트라 예측 모드(모드 1)는, 또한 (이용가능한 경우) 이웃 참조 샘플들을 이용하여 예측 블록을 오프셋으로 채운다.

[0084] 각도 인트라 예측 모드들(모드들 2-66)은 65개의 방향 또는 "각도" 중 하나의 것에 정렬된 텍스처를 산출함으로써 블록을 채운다. 명료성을 위해, 모드들 2, 18, 34, 50 및 66인, 65개의 각도의 서브세트가 도 8a에 도시되어 있다. 각각의 모드에 대해, 이웃 참조 샘플들이 각도 인트라 예측 모드에 대해 화살표에 의해 표시된 방향으로 예측 블록을 채우는 텍스처를 산출하기 위해 사용된다. 도 8a에 명시적으로 도시되지 않은 추가 각도들은 중간 위치들(즉, 모드들 3-17, 19-33, 35-49, 및 51-65)에 있다. 제1 대칭은 도 8a로부터 각도 모드(34)를 따라 그리고 예측 블록의 좌측 상단으로부터 우측 하단으로의 대각선 방향으로 명백하다. 제1 대칭으로부터, 모드 2 내지 모드 34는 이 대각선 축을 따른 전치를 가지며, 모드 66에서 모드 34까지에 대응하는 것으로 도시된다.

[0085] 도 8b는 변환 인덱스(844)에 따라 변환 세트들(842) 각각으로부터 특정 변환 타입으로의 매핑(840)을 도시한다. 3개의 변환 세트(842)가 0, 1, 및 2의 각자의 변환 세트 인덱스들에 의해 정의되고 식별된다. 각각의 변환 세트(842)는 변환 인덱스(844)에 의해 선택가능한 2개의 변환 타입을 포함한다. 그 결과, 매핑(840)은 변환 세트들(842)의 변환 세트 인덱스 및 변환 인덱스(844)에 기초하여 주어진 변환 타입이 선택되는 것을 허용한다. 선택된 변환은 1차원 변환이다. 수평 및 수직 방향들에 대해 변환 타입의 별개의 선택이 수행된다. 변환의 크기(포인트들의 수)는 변환 블록의 치수에 대응한다. 즉, 변환의 크기는 수직 변환을 선택할 때 변환 블록의 높이에 대응하는 한편, 변환의 크기는 수평 변환을 선택할 때 변환 블록의 폭에 대응한다.

[0086] 도 8c는 변환 블록이 수평 방향("H") 또는 수직 방향("V")에 있는지에 기초하여 인트라 예측 모드들로부터 변환 블록에 대한 변환 세트들(842)의 변환 세트 인덱스로의 매핑(880)을 도시하는 표이다. 각각의 인트라 예측 모드에 대해, 3개의 변환 세트(842)와 연관된 변환 세트 인덱스들 중 하나는 수평 방향 H에 대해 선택되고 3개의 변환 세트(842)와 연관된 변환 세트 인덱스들 중 하나는 수직 방향 V에 대해 선택된다.

[0087] 예를 들어, 인트라 예측 모드 14는 수직 방향에서의 변환 블록에 대해 변환 세트 인덱스 0에 매핑되는 한편, 인트라 예측 모드 14는 수평 방향에서의 변환 블록에 대해 변환 세트 인덱스 2에 매핑된다. 도 8b의 매핑(840)에

도시된 바와 같이, 변환 세트 인덱스 0은, 변환 인덱스(844)에 의존하여 선택가능한 DST-VII 또는 DCT-VIII에 매핑된다. 또한 도 8b의 매핑(840)에 도시된 바와 같이, 변환 세트 인덱스 2는 변환 인덱스(844)에 의존하여 선택가능한 DST-VII 또는 DCT-V에 매핑된다. 따라서, 수직 방향에서의 변환 블록에 대해, 인트라 예측 모드 14는 변환 인덱스(844)에 의존하여 DST-VII 또는 DCT-VIII 변환을 사용한다. 따라서, 수평 방향에서의 변환 블록에 대해, 인트라 예측 모드 14는 변환 인덱스(844)에 의존하여 DST-VII 또는 DCT-V 변환을 사용한다. 변환 인덱스(844)는 명시적인 다중 변환 인덱스(390 또는 454)에 기초한다. 전술한 바와 같이, 명시적 다중 변환 인덱스(390)는 모드 선택기(306)에 의해 선택되고, 명시적 다중 변환 인덱스(454)는 비트스트림(133)으로부터 엔트로피 인코더(420)에 의해 도출된다.

- [0088] 도 9는 비디오 인코더(114)의 역변환 모듈(348) 또는 비디오 디코더(134)의 역변환 모듈(444)을 도시한 개략 블록도이다. 양쪽 모듈(348, 444)의 동작은 동일하며, 이는 비디오 인코더(114)가 비디오 디코더(134)에서 수행된 잔차 계수들의 역양자화 및 역변환을 모델링할 필요성을 반영한다.
- [0089] 역변환 모듈들(348, 444) 각각은 변환 세트 테이블 모듈(910), 변환 인덱스 도출기 모듈(920), 수평 변환 선택기 모듈(940), 수직 변환 선택기 모듈(945), 수평 변환 모듈(960), 및 수직 변환 모듈(980)을 포함한다.
- [0090] 변환 세트 테이블 모듈(910)은 인트라 예측 모드(388 또는 458)를 수신한다. 역변환 모듈(348)로서 구현되는 경우, 변환 세트 테이블 모듈(910)은 모드 선택기(386)로부터 인트라 예측 모드(388)를 수신한다(도 3 참조). 역변환 모듈(444)로서 구현되는 경우, 변환 세트 테이블 모듈(910)은 엔트로피 인코더(420)로부터 인트라 예측 모드(458)를 수신한다(도 4 참조).
- [0091] 변환 세트 테이블 모듈(910)은 인트라 예측 모드(388 또는 458)로부터 (화살표(915)로 표현된) 수평 변환 세트 인덱스 및 (화살표(917)로 표현된) 수직 변환 세트 인덱스를 결정한다. 수평 변환 세트 인덱스(915) 및 수직 변환 세트 인덱스(917)는 매핑(880)에 따라 결정된다(도 8c 참조).
- [0092] 변환 인덱스 도출기(920)는 명시적인 다중 변환 인덱스(390 또는 454)를 수신한다. 역변환 모듈(348)로서 구현될 때, 변환 인덱스 도출기(920)는 모드 선택기(386)로부터 명시적 다중 변환 인덱스(390)를 수신한다(도 3 참조). 역변환 모듈(444)로서 구현될 때, 변환 인덱스 도출기(920)는 엔트로피 인코더(420)로부터 명시적 다중 변환 인덱스(454)를 수신한다(도 4 참조).
- [0093] 변환 인덱스 도출기(920)는 명시적 다중 변환 인덱스(390 또는 454)로부터 수평 변환 인덱스(920) 및 수직 변환 인덱스(925)를 추출한다. 수평 변환 인덱스(920) 및 수직 변환 인덱스(925)는 각각 0 내지 1의 범위에 있다.
- [0094] 수평 변환 선택기 모듈(940)은 (변환 세트 테이블 모듈(910)로부터) 수평 변환 세트 인덱스(915) 및 (변환 인덱스 도출기(920)로부터) 수평 변환 인덱스(920)를 수신한다. 수평 변환 선택기(940)는 매핑(840)(도 8b 참조)에 따라 수평 변환 세트 인덱스(915) 및 수평 변환 인덱스(920)로부터 선택된 수평 변환(화살표(950)로서 표현됨)을 산출한다. 수평 변환 세트 인덱스(915)는 변환 세트들(842) 중 하나를 선택하기 위해 사용되고 수평 변환 인덱스(920)는 변환 인덱스(844) 중 하나를 선택하기 위해 사용된다.
- [0095] 수직 변환 선택기 모듈(945)은 (변환 세트 테이블 모듈(910)로부터) 수직 변환 세트 인덱스(917) 및 (변환 인덱스 도출기(920)로부터) 수직 변환 인덱스(925)를 수신한다. 수직 변환 선택기 모듈(945)은 매핑(840)(도 8b 참조)에 따라 수직 변환 세트 인덱스(917) 및 수직 변환 인덱스(925)로부터 선택된 수직 변환(955)을 산출한다. 수직 변환 세트 인덱스(917)는 변환 세트들(842) 중 하나를 선택하는데 이용되고 수직 변환 인덱스(925)는 변환 인덱스(844) 중 하나를 선택하는데 이용된다.
- [0096] 수평 변환 모듈(960)은 선택된 수평 변환(950)을 중간 역변환 계수들(346) 또는 재구성된 변환 계수들(440)에 적용한다. 중간 역변환 계수들(346)은 역 2차 변환 모듈(344)(도 3 참조)로부터의 것이다. 재구성된 변환 계수들(440)은 역 2차 변환 모듈(436)(도 4 참조)로부터의 것이다.
- [0097] 변환 계수들(346 또는 440)은 선택된 수평 변환(950)에 따라 변환된다. 즉, 1차원 변환들의 세트를 갖는 선택된 수평 변환(950)은 변환 계수들(346 또는 440)의 어레이의 각각의 행에 적용된다. 각각의 1차원 변환은 변환 블록의 폭에 걸쳐 있다. 수평 변환 모듈(960)은 수평 변환된 변환 계수들(970)을 현재 변환 블록과 동일한 크기를 갖는 어레이로서 출력한다. 변환 계수들(970)은 수직 변환 모듈(980)에 공급된다. 설명의 편의를 위해, 변환 계수들(970)은 수평 변환된 변환 계수들(346 및 440) 둘 다를 지칭한다.
- [0098] 수직 변환 모듈(980)은 1차원 변환들의 세트를 갖는 선택된 수직 변환들(955)을 수평 변환된 변환 계수들(970)

의 각각의 열에 적용한다. 수직 변환 모듈(980)은 수직 선택된 변환(955)을 사용하여 각자의 수평 변환된 변환 계수들(346 또는 440)로부터 잔차 샘플들(350 또는 448)을 산출한다.

- [0099] 도 10a는 절단된 단항 선택 요소와 변환 선택을 위한 변환 인덱스들 사이의 관계를 보여주는 테이블(1000)을 도시한다. 변환 인덱스들은 수평 변환 인덱스 H(열(1048)에 도시됨) 및 수직 변환 인덱스 V(열(1046)에 도시됨)이다. 테이블(1000)은 빈 스트링 열(1042), 이진화 열(1044), 수직 변환 인덱스 열(1046), 및 수평 변환 인덱스 열(1048)을 포함한다.
- [0100] 빈 스트링 열(1042)은 최대값 3(즉, 0 내지 3의 범위)을 갖는 절단된 단항 이진화로부터의 4개의 가능한 빈 스트링을 갖는다. 인코더(114)의 경우, 열(1042)에 도시된 빈 스트링들은 엔트로피 인코더(338)가 비트스트림(115)으로 인코딩하는 빈 스트링들이다. 비트스트림(115)으로 인코딩될 빈 스트링은 모드 선택기(386)에 의해 선택된 수평 변환 인덱스 H 및 수직 변환 인덱스 V에 기초하여 선택된다.
- [0101] 디코더(134)의 경우, 열(1042)에 도시된 빈 스트링들은 비트스트림(133)으로부터 엔트로피 디코더(420)에 의해 디코딩되는 빈 스트링들이다. 이후, 엔트로피 디코더(420)는 디코딩된 빈 스트링에 기초하여 수평 변환 인덱스 H 및 수직 변환 인덱스 V를 출력한다.
- [0102] 이진화 열(1044)은 빈 스트링 열(1042)에서의 각각의 빈 스트링에 대한 조성을 포함한다. 이 명명법에서, "Ctx0" 및 "Ctx1"은 빈 스트링 열(1042)에서 대응하는 빈들을 코딩하는데 이용되는 2개의 별개의 CABAC 콘텐츠를 지칭한다. 용어 "Byp"는 빈 스트링 열(1042)에서의 대응하는 빈을 코딩하는데 이용되는 바이패스 코딩된 빈을 지칭한다.
- [0103] 수직 변환 인덱스 열(1046)은 수직 1D 변환들에 대한 변환 세트로부터 변환을 선택하기 위한 인덱스 값(0 또는 1임)을 포함한다. 수평 변환 인덱스 열(1048)은 수평 1D 변환들에 대한 변환 세트로부터 변환을 선택하기 위한 인덱스 값(0 또는 1임)을 포함한다. 테이블(1000)에서, 최단 빈 스트링이 "0" 및 "0"의 수직 및 수평 인덱스들의 경우에 할당된다. 최단 빈 스트링은 모드 선택기(386)에 의해 가장 빈번하게 선택된 경우에 대응한다. 그 다음, 두 번째 최단 빈 스트링은 "1" 및 "1"의 수직 및 수평 인덱스들의 경우에 할당된다. 이것은 모드 선택기(386)에 의해 두번째로 가장 빈번하게 선택된 경우에 대응한다. 이후, 수직 변환 인덱스와 수평 변환 인덱스의 조합에 대한 "01" 또는 "10"의 인덱스 값 조합의 경우들은 동일하게 불가능하고, 열(1044)에서 가장 긴 빈 스트링들이 할당된다. 더욱이, 이들 두 가지 경우 간의 구별은 바이패스 코딩된 빈을 사용하여 수행되어, 모드 선택기(386)에 의한 이들 두 가지 선택 간의 거의 동일한 발생 확률을 반영한다. 설명된 통계적 속성들은 변환 크기에 대해 주로 불변인 것으로 밝혀졌고, 따라서 절단된 단항 빈 스트링에서 그들의 위치에 따라 선택되는 이러한 컨텍스트들의 존재는 빈 값의 비대칭 확률 분포를 포착하기에 적합하다.
- [0104] 도 10b는 수평 및 수직 변환 인덱싱에 대한 절단된 단항 선택 요소와 변환 인덱스들 사이의 대안 관계를 보여주는 테이블(1050)이며, 절단된 단항 이진화의 제3 빈은 바이패스 코딩이 아니라 컨텍스트 코딩을 이용한다.
- [0105] 도 11은 잔차 계수들의 변환 블록을 비트스트림(115)으로 인코딩하기 위해 명시적 변환 선택에 따라 1차 변환을 수행하기 위한 방법(1100)의 흐름도이다. 방법(1100)은 구성된 FPGA, ASIC, 또는 ASSP와 같은 장치에 의해 구현될 수 있다. 또한, 방법(1100)은 프로세서(205)의 실행 하에서 비디오 인코더(114)에 의해 수행될 수 있다. 이에 따라, 방법(1100)은 컴퓨터 판독가능 저장 매체상에 및/또는 메모리(206)에 저장될 수 있다. 방법(1100)은 인트라 예측 모드 결정 단계(1110)에서 프로세서(205)로 시작된다.
- [0106] 방법(1100)은 인트라 예측 모드 단계(1110)에서 시작된다. 인트라 예측 모드 단계(1110)에서, 모드 선택기(386)는, 프로세서(205)의 실행 하에서, 현재 코딩 유닛과 연관된 예측 유닛에 대한 인트라 예측 모드(388)를 선택한다. 선택은 일반적으로 두 번의 패스로 수행된다. 제1 패스에서, 루마 예측 블록에 대한 모든 인트라 예측 모드들이 테스트된다. 각각의 모드에 대해, 잔차 비용은, 예를 들어, 하다마드(Hadamard) 변환과 같은 "절대 변환 차이들의 합" 방법을 사용하여 근사화된다. 이 테스트로부터, "최상의" (최저 왜곡의) 후보 예측 모드들의 리스트가 도출된다. 그 후, 후보 예측 모드들의 리스트에 대해 잔차 코딩의 전체 테스트가 수행된다. 전체 테스트의 결과로서, 하나의 인트라 예측 모드(388)가 현재 코딩 유닛에 대해 선택된다. 인트라 예측 모드(388)는 엔트로피 인코더(338)에 의해 비트스트림(115)으로 인코딩된다. 이후, 프로세서(205)에서의 제어는 인트라 예측 모드 단계(1110)로부터 변환 세트 결정 단계(1120)로 진행한다.
- [0107] 변환 세트 결정 단계(1120)에서, 역변환 모듈(348)은, 프로세서(205)의 실행 하에서, 결정된 인트라 예측 모드(388)를 이용하여 매핑(880)(도 8c 참조)에 따라 (도 9와 관련하여 전술한 바와 같은) 수평 변환 세트 인덱스(915) 및 (도 9와 관련하여 전술한 바와 같은) 수직 변환 세트 인덱스(917)를 결정한다. 수평 변환 세트 인덱

스(915)로부터, 매핑(840)에 따라 수평 변환에 대한 변환들의 세트가 결정된다. 수직 변환 세트 인덱스(917)로부터, 매핑(840)에 따라 수직 변환에 대한 변환들의 세트가 결정된다.

- [0108] 상기 예에서 설명된 바와 같이, 인트라 예측 모드(388)가 모드 14인 경우, 수직 변환에 대한 가능한 변환들은 DST-VII 또는 DCT-VIII이고 수평 변환에 대한 가능한 변환들은 DST-VII 또는 DCT-V이다.
- [0109] 그 후, 프로세서(205)에서의 제어는 변환 세트 결정 단계(1120)로부터 변환 테스트 단계(1130)로 진행한다.
- [0110] 변환 테스트 단계(1130)에서, 변환 모듈(326) 및 역변환 모듈(348)은, 프로세서(205)의 실행 하에서, 단계(1120)로부터의 수평 변환에 대한 변환들의 세트 및 수직 변환에 대한 변환들의 세트로부터의 변환들의 각각의 조합을 테스트한다. 인트라 예측 모드(388)가 모드 14인 예에서, DST-VII 및 DCT-VIII의 변환들은 수직 변환에 대해 테스트되고 DST-VII 또는 DCT-V의 변환들은 수평 변환에 대해 테스트된다.
- [0111] 진술한 바와 같이, 테스트는 변환 모듈(326) 및 역변환 모듈(348)에 의해 수행된다. 각각의 세트에서의 특정 변환들은 (수평 변환에 대한) 수평 변환 인덱스(920) 및 (수직 변환에 대한) 수직 변환 인덱스(925)를 사용하여 참조된다. 수직 및 수평 변환들의 각각의 조합은 잔차를 코딩하기 위한 왜곡을 결정하기 위해 테스트된다.
- [0112] 테스트 동안, 변환 모듈(326) 및 역변환 모듈(348)은 모듈의 "정상 동작"을 수행한다. 따라서, 역변환 모듈(348)에 의해 수행되는 모든 역변환 동작들에 대해, 변환 모듈(326)은 대응하는 순방향 변환 동작을 수행한다. 테스트 프로세스는 전체 변환을 수행하는 것과 비교하여 감소된 복잡도로 테스트를 수행하기 위해 근사화 방법들을 사용할 수 있다.
- [0113] 프로세서(205)에서의 제어는 변환 세트 테스트 단계(1130)로부터 변환 선택(1140)으로 진행한다.
- [0114] 변환 선택 단계(1140)에서, 잔차의 코딩을 위한 최저 왜곡을 야기하는 수직 및 수평 변환들의 조합이 선택된다. 인트라 예측 모드(388)가 모드 14인 예에서, 최저 왜곡을 제공하는 조합은 수직 변환에 대한 DCT-VIII 및 수평 변환에 대한 DST-VII일 수 있다. 프로세서(205)에서의 제어는 수평 변환(1150)을 적용하기 위해 변환 선택 단계(1140)로부터 진행한다.
- [0115] 수평 변환 적용 단계(1150)에서, 역변환 모듈(348)은 수평 변환을 수행한다. 특히, 수평 변환 모듈(960)은 선택된 수평 변환(950)을 수행한다. 인트라 예측 모드(388)가 모드 14인 예에서, DST-VII이 수평 변환을 위해 사용된다. 그 후 프로세서(205)에서의 제어는 수평 변환 적용(1150)으로부터 수직 변환 적용(1160)으로 진행한다.
- [0116] 수직 변환 적용 단계(1160)에서, 역변환 모듈(348)은 수직 변환을 수행한다. 특히, 수직 변환 모듈(980)은 선택된 수직 변환(955)을 수행한다. 인트라 예측 모드(388)가 모드 14인 예에서, DCT-VIII은 수직 변환을 위해 사용된다. 수평 변환 적용 단계(1150) 및 수직 변환 적용 단계(1160)의 결과로서, 2차 변환(즉, 모듈(330)의 동작)의 선택적인 적용 후에, 변환 계수들(332)이 산출된다. 그 후 프로세서(205)에서의 제어는 수직 변환 적용 단계(1160)로부터 잔차 계수 양자화 단계(1170)로 진행한다.
- [0117] 잔차 계수 양자화 단계(1170)에서, 양자화기 모듈(334)은 프로세서(205)의 실행 하에서 변환 계수들(332)을 양자화하여 양자화 파라미터에 따라 잔차 계수들(336)을 산출한다. 그 후, 프로세서(205)에서의 제어는 잔차 계수 양자화 단계(1170)로부터 잔차 계수 인코딩 단계(1180)로 진행한다.
- [0118] 잔차 계수 인코딩 단계(1180)에서, 엔트로피 인코더(338)는, 프로세서(205)의 실행 하에서, 잔차 계수들(336)을 비트스트림(115)으로 인코딩한다. 주어진 변환 블록에 대한 잔차 계수들(336)은 계수들의 2차원 어레이를 리스트로 스캐닝함으로써 코딩된다. 스캔은 일반적으로 변환 블록을 각각이 크기 4x4인 하나 이상의 서브 블록으로 분할한다(다른 크기도 가능하다). 각각의 서브블록에서의 유의 계수(significant coefficient)의 존재는 "코딩된 서브블록 플래그"로 시그널링되고 및, 적어도 하나의 유의 계수가 존재하는 경우 서브블록에서의 잔차 계수 당 하나의 유의 플래그로 시그널링된다. 코딩된 서브블록 플래그들 및 유의 플래그들은 컨텍스트 코딩된 bin들을 이용한다. "하나보다 큰 플래그" 및 "2개보다 큰 플래그"와 같은, 크기의 추가적인 표시들도 컨텍스트 코딩된 bin들을 이용하여 코딩된다. 마지막으로, 크기의 추가적인 표시들은 일반적으로 바이패스 코딩된 bin들을 갖는 골롬 라이스 코딩(Golomb Rice coding)을 사용하여 인코딩된다. 잔차 계수들(336)을 코딩하는데 있어서의 비트들에서의 비용은 그들의 크기들에 의해 주로 영향을 받는다. 최적의 EMT 인덱스들의 선택의 결과로서, 이러한 크기들은 약간 감소되어, 압축 효율의 개선으로 이어진다. 단계(1190)에서 수행되는 이러한 인덱스들의 이전화는 압축 효율을 더 개선한다. 프로세서(205)에서의 제어는 그 후 잔차 계수 인코딩 단계(1180)로부터 EMT bin 스트림 인코딩 단계(1190)로 진행한다.
- [0119] EMT bin 스트림 인코딩 단계(1190)에서, 엔트로피 인코더(338)는, 프로세서(205)의 실행 하에서, 도 10a를 참조

하여 설명된 바와 같이, 절단된 단항 이진화(최대 값 3), 컨텍스트 모델링, 및 빈 스트링들의 인덱스 값들로의 매핑을 이용하여 수평 변환 인덱스(920) 및 수직 변환 인덱스(925)를 비트스트림(115)으로 인코딩한다. 가장 빈번하게 선택된 인덱스들의 쌍(수평으로 "0" 및 수직으로 "0")이 가장 짧은 빈 스트링("0")을 사용하여 인코딩 되기 때문에, 각각의 인덱스에 대해 하나의 빈을 사용하는 간단한 경우보다 코딩 효율이 높다. 두 번째로 가장 빈번하게 선택된 인덱스들의 쌍(수평으로 "1" 및 수직으로 "1")은 두 번째로 짧은 빈 스트링, 즉 "1 0"을 사용하여 인코딩되고, 두 개의 거의 동일하게 가장 가능성이 낮은 조합들("1" 및 "0", "0" 및 "1")은 가장 긴 빈 스트링들("1 1 0" 및 "1 1 1")을 사용하여 인코딩된다. 제3 빈은, 컨텍스트 코딩을 이용한 활용에 이용가능한 통계적 비대칭이 없기 때문에, 바이패스 코딩을 이용한다. 이후 방법(1100)은 단계(1190)의 끝에서 종료된다.

[0120] 도 12는 비트스트림(133)으로부터 잔차 계수들의 변환 블록을 디코딩하기 위한 방법(1200)의 흐름도이다. 방법(1200)의 디코딩은 명시적 변환 선택에 따라 선택된 1차 역변환에 의해 수행된다. 방법(1200)은 구성된 FPGA, ASIC, 또는 ASSP와 같은 장치에 의해 구현될 수 있다. 또한, 방법(1200)은 프로세서(205)의 실행 하에서 비디오 디코더(134)에 의해 수행될 수 있다. 그에 따라, 방법(1200)은 컴퓨터 판독가능 저장 매체상에 및/또는 메모리(206)에 저장될 수 있다. 방법(1200)은 인트라 예측 모드 디코딩 단계(1210)에서 프로세서(205)로 시작한다.

[0121] 인트라 예측 모드 디코딩 단계(1210)에서, 엔트로피 디코더(420)는, 프로세서(205)의 실행 하에서, 비트스트림(133)으로부터 인트라 예측 모드(458)를 디코딩한다. 엔트로피 디코더(420)는 모드 선택기(386)에 의해 선택되었을 가능성이 가장 높게 여겨지는 모드들을 포함하는 최고 확률 모드들의 리스트를 산출한다. 컨텍스트 코딩된 빈은 최고 확률 모드들 중 하나 또는 최고 확률 모드 리스트에 있지 않은 다른 모드들("잔차 모드들")의 사용을 나타내기 위해 디코딩된다. 이후, 어느 최고 확률 모드가 사용 중인지 또는 어느 잔차 모드가 사용 중인지를 식별하기 위해 추가의 빈들이 디코딩된다. 그후 프로세서(205)에서의 제어는 인트라 예측 모드 디코딩 단계(1210)로부터 잔차 계수 디코딩 단계(1220)로 진행한다.

[0122] 잔차 계수 디코딩 단계(1220)에서, 엔트로피 디코더(420)는, 프로세서(205)의 실행 하에서, 잔차 계수들의 리스트를 디코딩하고 잔차 계수들을 변환 블록에 대한 어레이로 어셈블링한다. 어셈블리는 전형적으로 잔차 계수들을 4x4 서브 블록들로 그룹화하는 스캔 패턴을 따른다. 잔차 계수들의 리스트는 전형적으로 역방향으로 스캐닝 되는데, 이는 "마지막" 유의 잔차 계수로부터 되돌려 좌측 상단 잔차 계수("DC" 잔차 계수이지만, 이것은 코사인 함수에 기초하지 않은 변환들에 대한 블록 오프셋에 대응하지 않을 수 있음)를 향한다. 컨텍스트 코딩된 빈들은 각각의 서브 블록에서 유의 잔차 계수들의 존재를 표시하기 위해 디코딩되고, 만일 표시된다면, 추가적인 컨텍스트 코딩된 빈들이 서브 블록에서 각각의 잔차 계수의 유의성을 나타내기 위해 디코딩된다. 각각의 잔차 계수의 크기는 "1보다 큰" 또는 "2보다 큰" 플래그들을 디코딩하는 것을 추가 특징으로 할 수 있다. 마지막으로, 그 크기가 선행하는 컨텍스트 코딩된 빈들에 의해 결정되지 않는 잔차 계수들에 대해, "잔여 크기"는 바이패스 코딩된 빈들을 갖는 곱셈 라이스 코딩을 사용하여 디코딩된다. 이 디코딩의 결과로서, 변환 블록에 대한 잔차 계수들(424)이 비트스트림(133)으로부터 디코딩된다. 그후, 프로세서(205)에서의 제어는 잔차 계수 디코딩 단계(1220)로부터 잔차 계수 역양자화 단계(1230)로 진행한다.

[0123] 잔차 계수 역양자화 단계(1230)에서, 양자화 해제기 모듈(428)은, 프로세서(205)의 실행 하에서, 양자화 파라미터에 따라 잔차 계수들(424)을 스케일링함으로써 중간 변환 계수들(432)을 산출한다. 중간 변환 계수들(432)은, 중간 변환 계수들(432)을 산출하도록, 일반적으로 변환 블록의 좌측 상부 영역에만 적용되는 분리가능하지 않은 2차 변환의 적용에 의해 변환될 수 있다. 이 스케일링 동작의 결과로서, 중간 변환 계수들(432)은 비디오 인코더에서의 양자화 이전의 것들(즉, 변환 계수들(332))과 대략 동일한 크기를 갖는다. 분리가능하지 않은 역 2차 변환(즉, 역 2차 변환 모듈(436))의 선택적 적용을 통해, 재구성된 변환 계수들(440)이 중간 변환 계수들(432)로부터 획득된다. 그후, 프로세서(205)에서의 제어는 잔차 계수 역양자화 단계(1230)로부터 EMT 빈 스트링 디코딩 단계(1240)로 진행한다.

[0124] EMT 빈 스트링 디코딩 단계(1240)에서, 엔트로피 디코더(420)는, 프로세서(205)의 실행 하에서, 비트스트림(133)으로부터 절단된 단항 빈 스트링을 디코딩한다. 절단된 단항 빈 스트링은 도 10a의 열(1042)에 열거된 빈 스트링들 중 하나이다. 도 10a에 도시되고 전술된 바와 같이, 단항 빈 스트링은 (0 내지 3의 범위에 있는) 길이가 3개의 빈에 제한된다. 절단된 단항 빈 스트링은 길이가 1개 내지 3개의 빈이며, 제1 빈은 컨텍스트 코딩되고, 제2 빈(존재하는 경우)은 상이한 컨텍스트로 컨텍스트 코딩되며, 제3 빈(존재하는 경우)은 바이패스 코딩된다. 이러한 이진화는 도 10을 참조하여 설명된 바와 같다. 그후, 프로세서(205)에서의 제어는 EMT 빈 스트링 디코딩 단계(1240)로부터 EMT 인덱스 결정 단계(1250)로 진행한다.

- [0125] EMT 인덱스 결정 단계(1250)에서, 비디오 디코더(134)에서의 변환 인덱스 도출기(920)는, 프로세서(205)의 실행 하에서, 도 10a의 테이블(1000)에 따라 수평 변환 인덱스(920) 및 수직 변환 인덱스(925)를 결정한다. 이후 프로세서(205)에서의 제어는 EMT 인덱스 결정 단계(1250)로부터 변환 세트 결정 단계(1260)로 진행한다.
- [0126] 변환 세트 결정 단계(1260)에서, 역변환 모듈(444)은, 프로세서(205)의 실행 하에서, 매퍼(840)에서의 어느 변환 세트가 1차 역변환의 수평 및 수직 변환에 사용될 것인지를 결정한다. 특히, 인트라 예측 모드(458)를 이용하여, 변환 세트 테이블(910)은 수평 변환 세트(915) 및 수직 변환 세트(917)를 결정하는데 이용된다. 이후, 프로세서(205)에서의 제어는 변환 세트 결정 단계(1260)로부터 변환 선택 단계(1270)로 진행한다.
- [0127] 변환 선택 단계(1270)에서, 역변환 모듈(444)은, 프로세서(205)의 실행 하에서, 수평 변환(950) 및 수직 변환(955)을 결정한다. 수평 변환(950) 및 수직 변환(955)의 결정은 수평 변환 선택기(940) 및 수직 변환 선택기(945)에 의해 제각기 수행된다. 모듈들은, 도 9에 도시된 바와 같이, 입력들로서 수평 변환 세트 인덱스(915), 수직 변환 세트 인덱스(917), 수평 변환 인덱스(920), 및 수직 변환 인덱스(925)를 취하고, 도 8b에 따라, 선택된 수평 변환(950) 및 선택된 수직 변환(960)을 산출한다. 이후 프로세서(205)에서의 제어는 변환 선택 단계(1270)로부터 수평 변환 적용 단계(1280)로 진행한다.
- [0128] 수평 변환 적용 단계(1280)에서, 역변환 모듈(444)은, 프로세서(205)의 실행 하에서, 수평 변환 모듈(960)을 사용하여 선택된 수평 변환(950)을 적용한다. 선택된 수평 변환(950)은 재구성된 변환 계수들(440)의 각각의 행에 적용되어 수평 변환된 변환 계수들(970)을 산출한다. 수평 변환된 변환 계수들(970)은 수직 변환 모듈(980)에 공급된다. 이후 수직 변환 모듈(980)은 선택된 수직 변환(955)을 적용함으로써 잔차 샘플들(448)을 산출한다. 잔차 샘플들(950)은 또한, 도 4를 참조하여 기술된 바와 같이, 재구성된 샘플들을 산출하기 위해 인트라 예측된 PU와의 합산을 위해 사용된다. 그 후 방법(1200)은 종료된다.
- [0129] 방법들(1100 및 1200)의 또 다른 구성에서, 절단된 단항 빈 스트링의 이진화는 하나의 컨텍스트 코딩된 빈만이 사용되도록 수정된다. 특히, 도 10a를 참조하면, 이진화(1044)의 제1 빈은 컨텍스트 코딩된 채로 남아있는 반면, 이진화(1044)의 제2 빈은 컨텍스트 코딩 대신에 바이패스 코딩된 빈을 사용한다. 하나의 컨텍스트 코딩된 빈만을 사용하는 배열들은 압축 효율에서의 약간의 열화를 희생하여 구현 비용을 절약한다. 그러나, 하나의 컨텍스트 코딩된 빈만을 사용하는 구성은 약간 더 높은 처리량을 달성할 수 있다. 더 높은 처리량은, 엔트로피 인코더(338) 및 엔트로피 디코더(420)에서의 CABAC 프로세스가 적어도 컨텍스트 상태 업데이트가 없기 때문에 컨텍스트 코딩된 빈들보다 더 높은 레이트로 바이패스 코딩된 빈을 처리할 수 있기 때문이다.
- [0130] 도 10a를 참조하여 언급된 바와 같이, 제각기 "0" 및 "1"의; 또는 제각기 "1" 및 "0"의 값들을 갖는 수평 변환 인덱스 H 및 수직 변환 인덱스 V의 선택 확률은 대략 동일하다. 이와 같이, 도 10a에 도시된 것에 대한 대안적인 이진화는 제3의 바이패스 코딩된 빈의 사용을 반전시킨다. 제3 빈의 사용을 반전시키는 구성은 각자의 수직 및 수평 변환 인덱스들에 대해 "1 1 1"이 "0", "1"에 매핑되는 빈 스트링, 및 각자의 수직 및 수평 변환 인덱스들에 대해 "1", "0"에 매핑되는 빈 스트링 "1 1 0"을 갖는다. 이들 두 가지 선택 간의 거의 동일한 확률로 인해, 등가의 압축 이점이 달성된다.
- [0131] 수평 및 수직으로의 명시적 다중 변환 선택의 이진화의 결과로서, 특히, 그들이 각각의 인덱스 값의 그들의 예상 선택 확률에 따라 공동으로 코딩되는 양태의 결과로서, 수평 및 수직 인덱스들(예를 들어, 각각 하나의 컨텍스트 코딩된 빈)에 대해 독립적인 선택 요소들이 각각 사용되는 경우에 비해 코딩 효율이 개선된다.

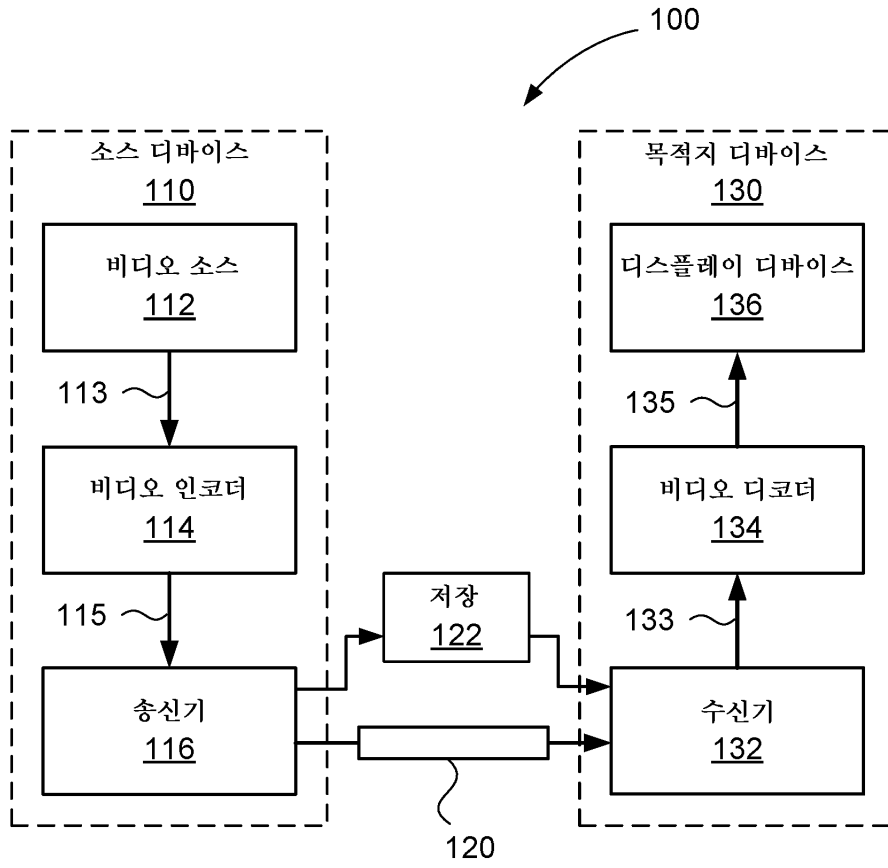
산업상 이용가능성

- [0133] 설명된 구성은 컴퓨터 및 데이터 처리 산업들에 적용가능하고, 특히 비디오 및 이미지 신호들과 같은 신호들의 인코딩 및 디코딩을 위한 디지털 신호 처리에 적용가능하여, 잔차 계수들의 비 국부화된 스캐닝으로 인해 메모리 대역폭의 관점에서 과도한 비용 없이 높은 압축 효율을 달성한다.
- [0134] 진술한 내용은 본 발명의 일부 실시예들만을 설명하고, 본 발명의 범위 및 사상을 벗어나지 않고 그에 대한 수정들 및/또는 변경들이 이루어질 수 있으며, 실시예들은 예시적인 것이고 제한적이지 않다.
- [0135] 본 명세서의 맥락에서, "포함하는(comprising)"이라는 단어는 "단지 그것으로 구성되는(consisting only of)"이 아니라, "주로 포함하지만 반드시 그것들만 포함하지는 않는" 또는 "갖는" 또는 "포함하는"을 의미한다. "포함하다(comprise)" 및 "포함하다(comprises)"와 같은, 단어 "포함하는(comprising)"의 변형들은 대응하여 변화된

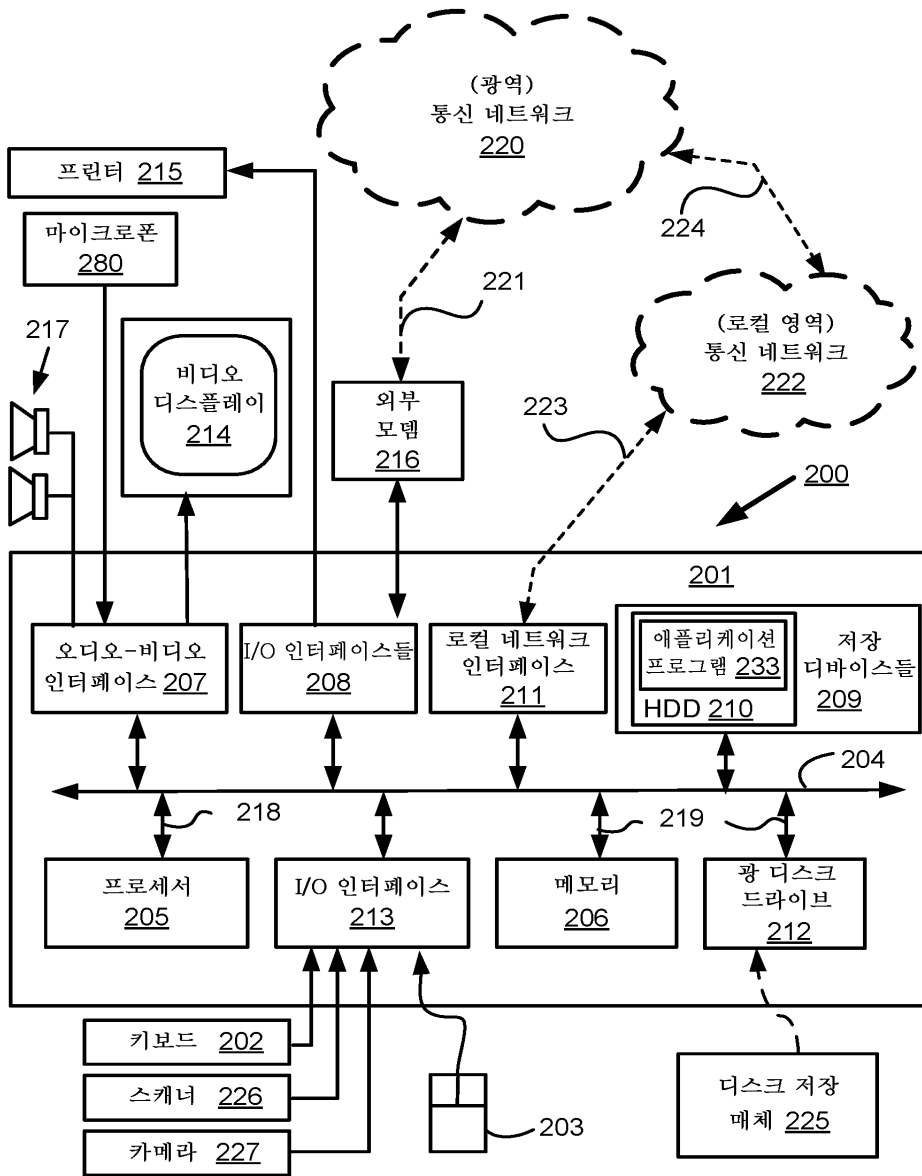
의미들을 갖는다.

도면

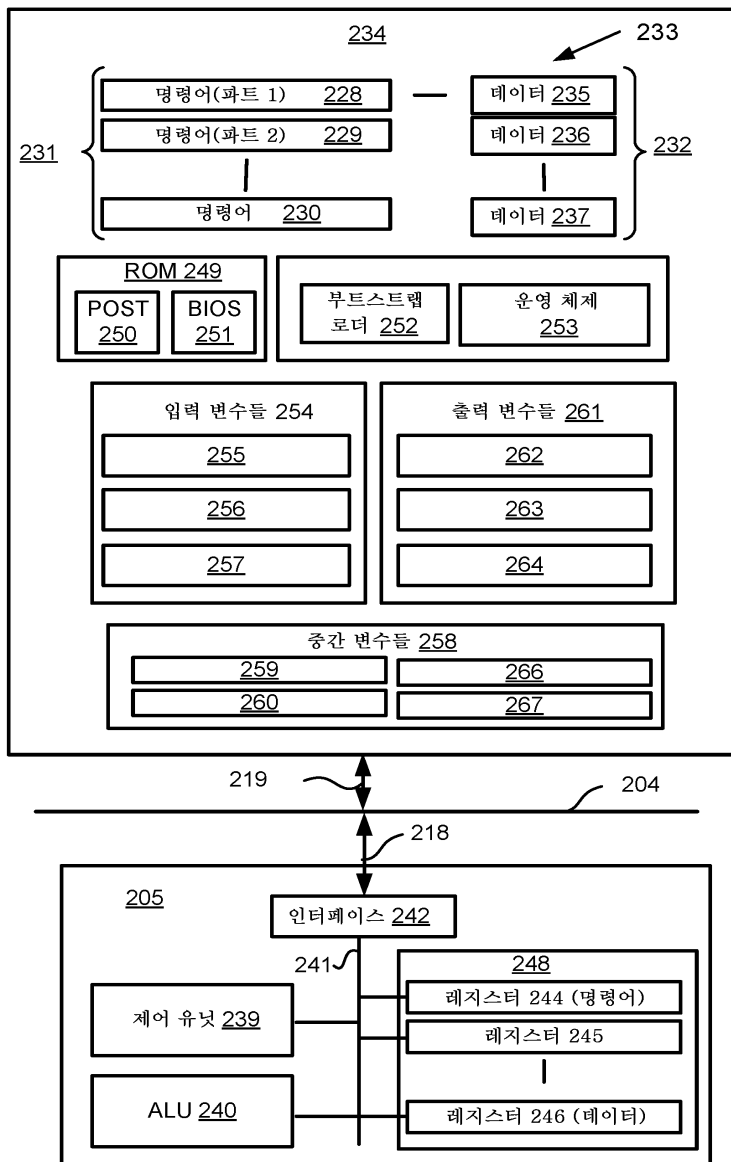
도면1



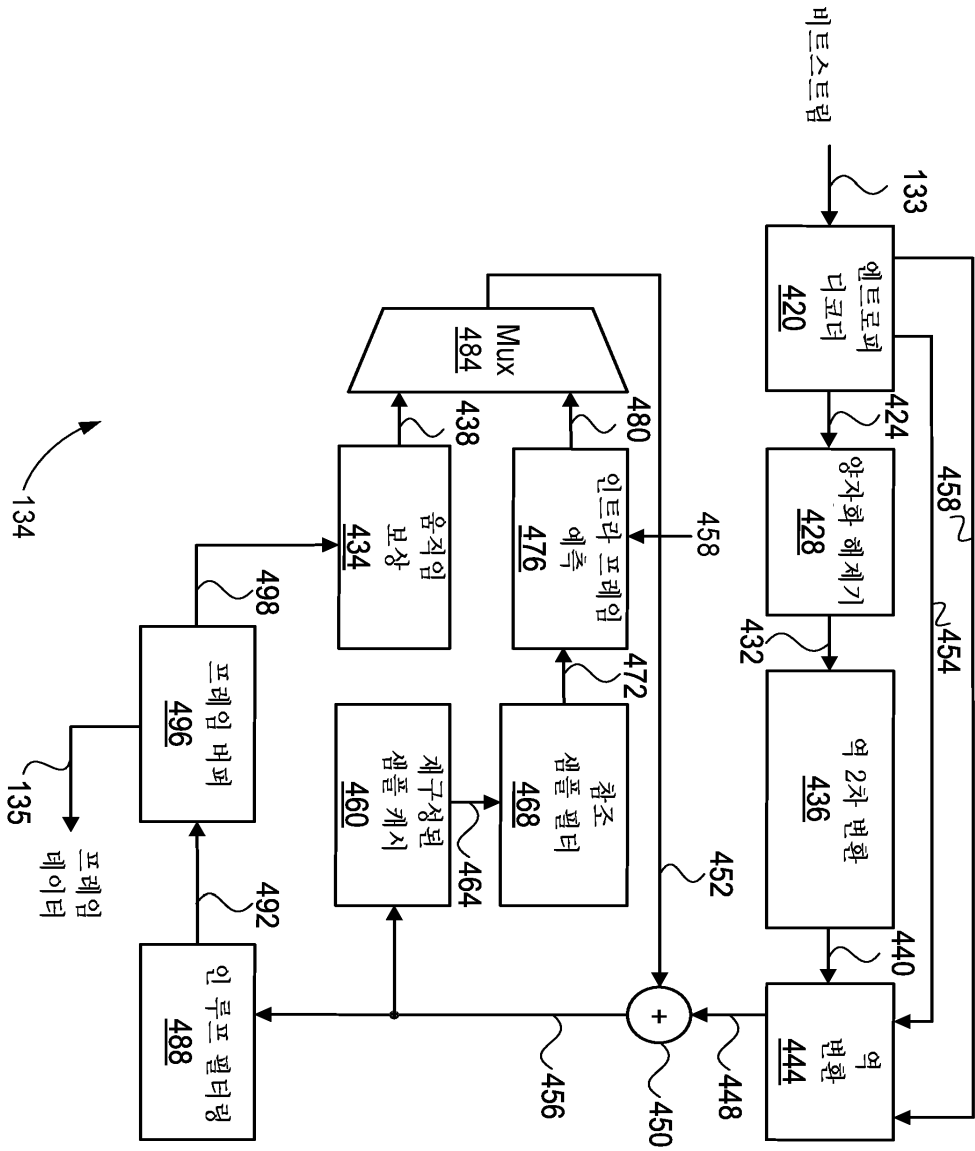
도면2a



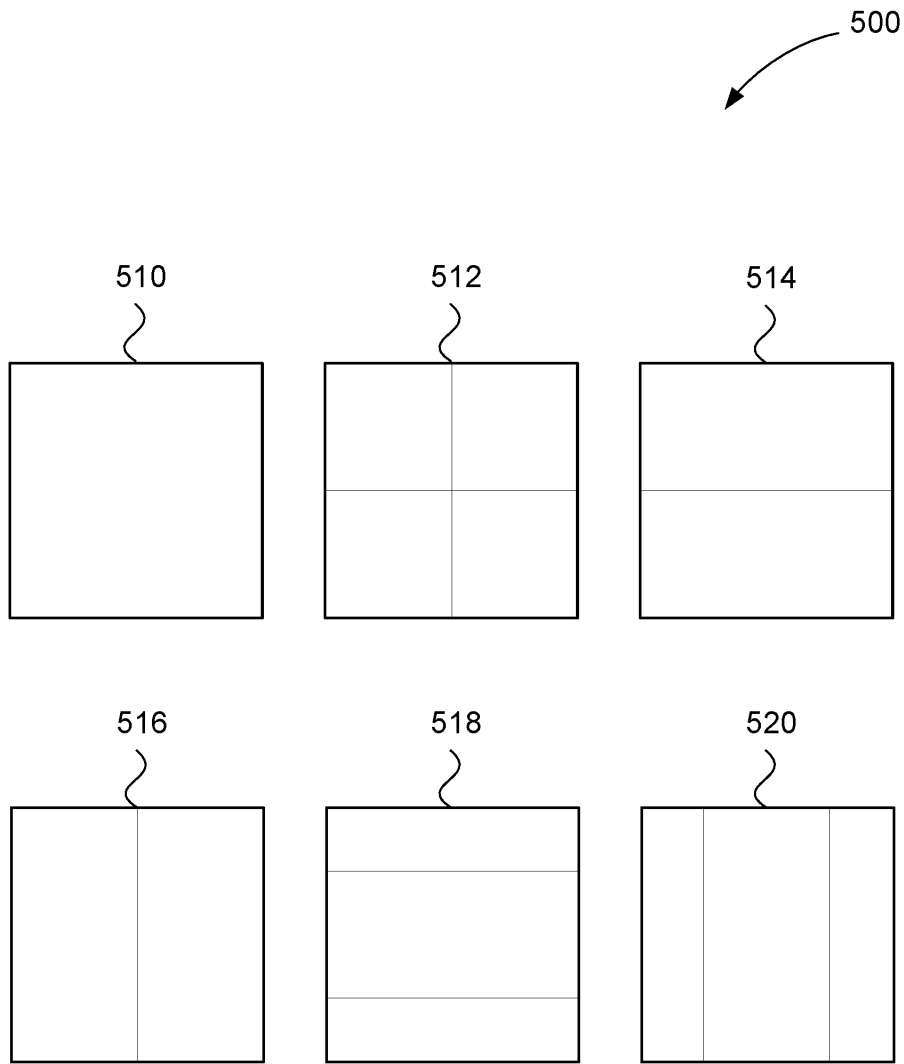
도면 2b



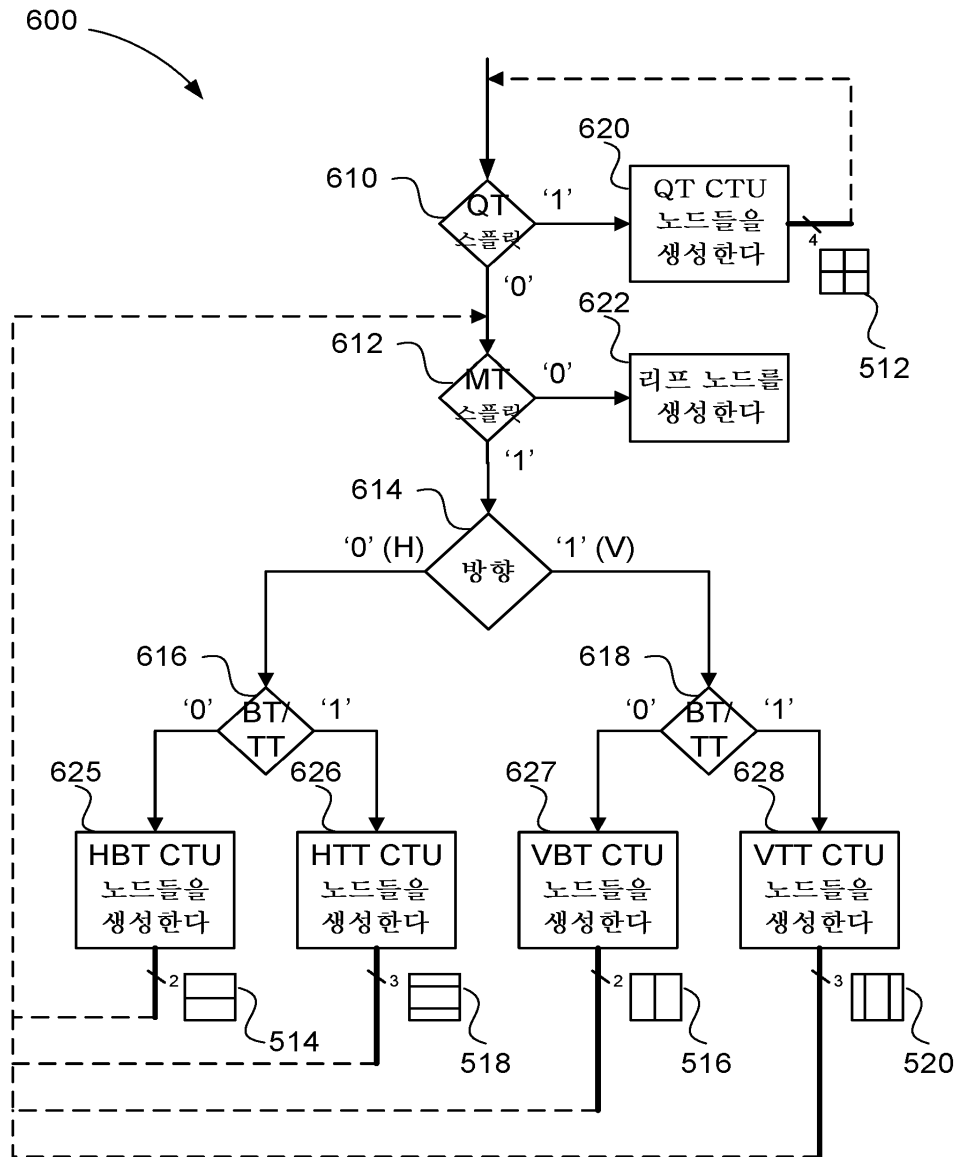
도면4



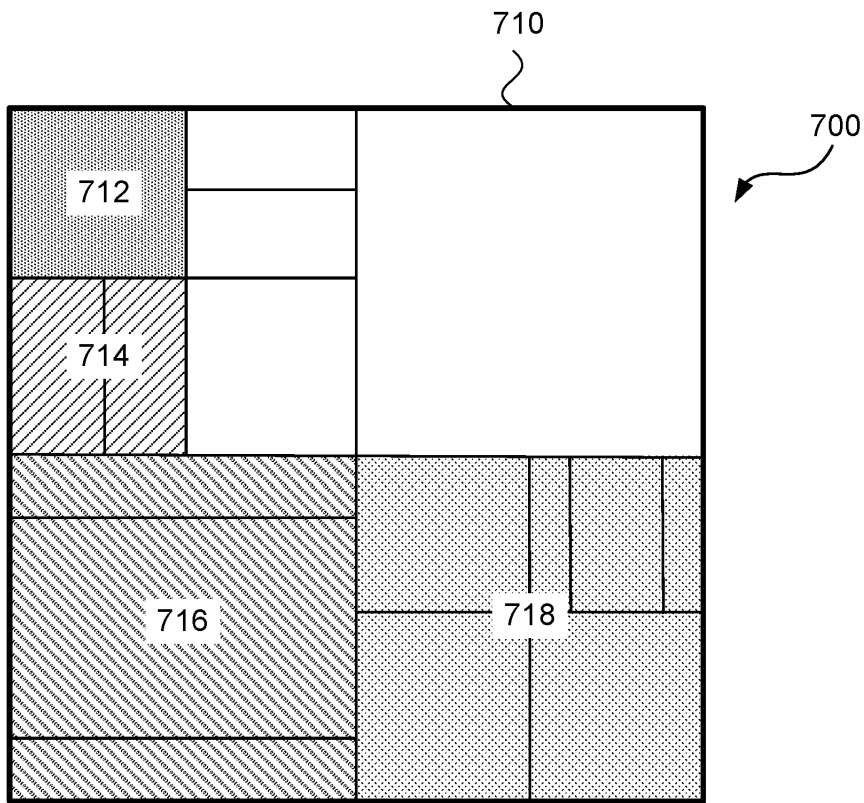
도면5



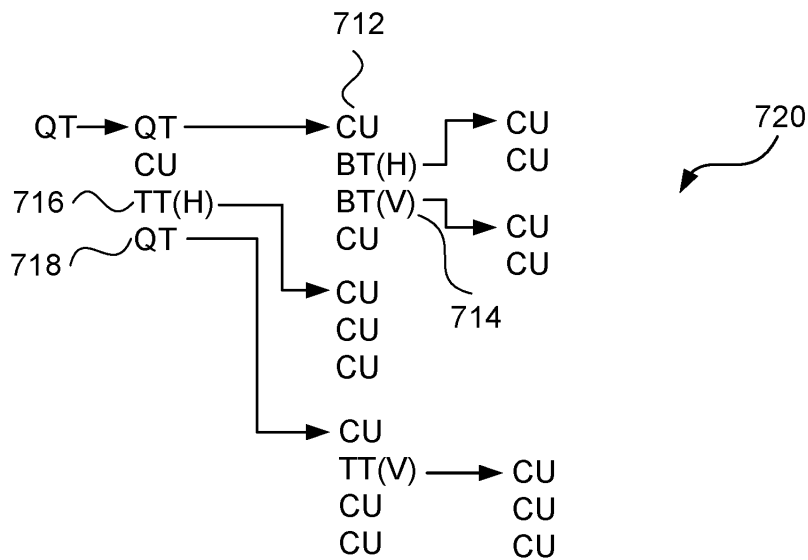
도면6



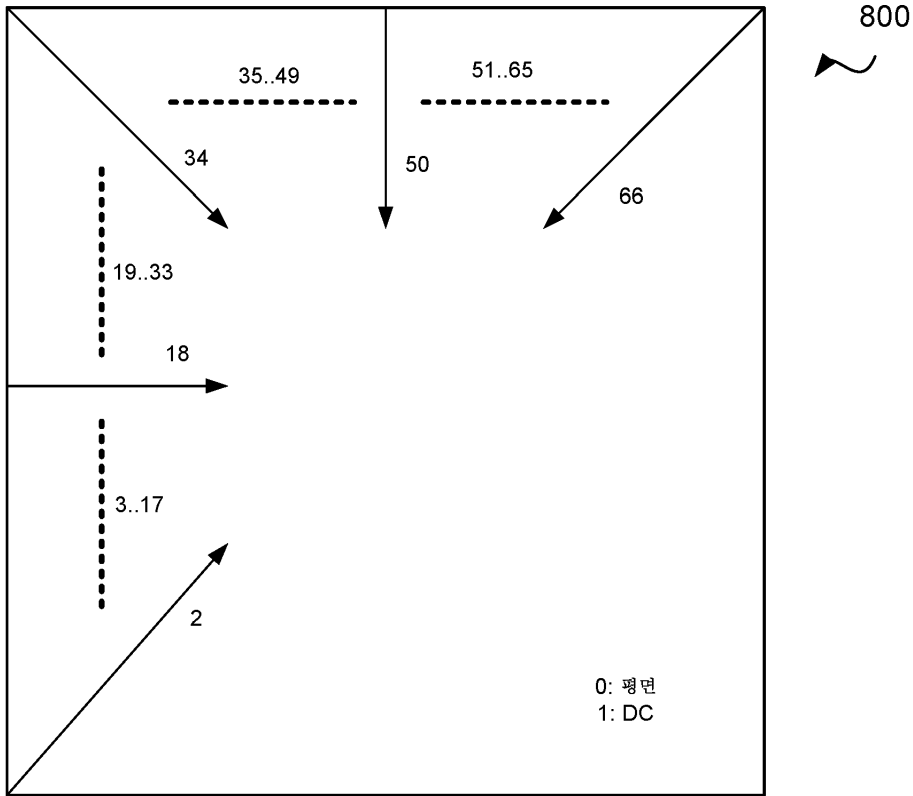
도면7a



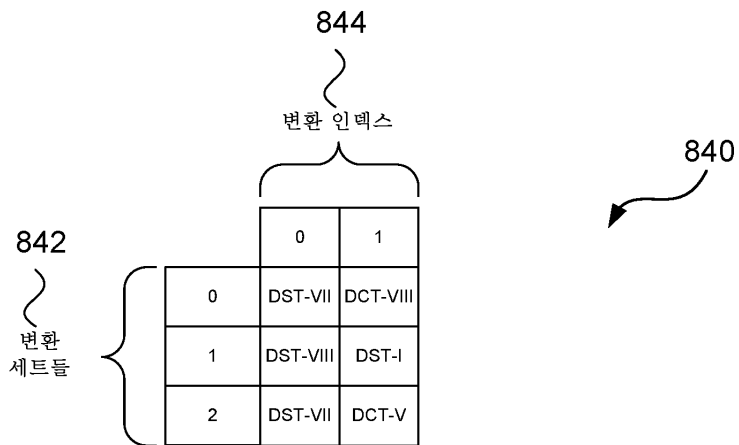
도면7b



도면8a



도면8b



도면8c

880



인트라 모드	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
V	2	1	0	1	0	1	0	1	0	1	0	1	0	1	0
H	2	1	0	1	0	1	0	1	0	1	0	1	0	1	2

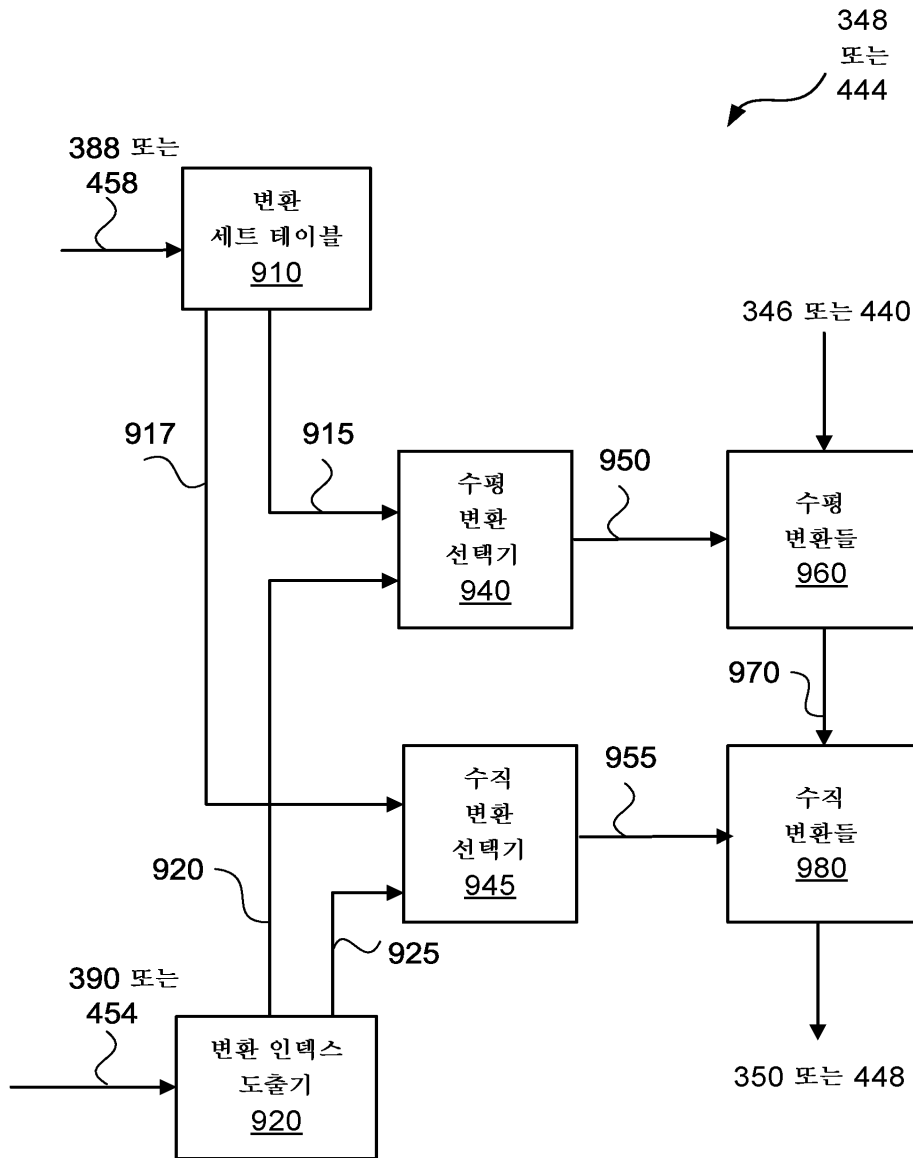
인트라 모드	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
V	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1
H	2	2	2	2	2	2	2	2	1	0	1	0	1	0	1

인트라 모드	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
V	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
H	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

인트라 모드	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
V	1	2	2	2	2	2	2	2	2	2	1	0	1	0	1
H	1	0	0	0	0	0	0	0	0	0	1	0	1	0	1

인트라 모드	60	61	62	63	64	65	66
V	0	1	0	1	0	1	0
H	0	1	0	1	0	1	0

도면9




도면10a

1000

1042	1044	1046	1048
빈 스트림	이진화	V	H
0	Ctx0	0	0
1 1 0	Ctx0 + Ctx1 + Byp	0	1
1 1 1	Ctx0 + Ctx1 + Byp	1	0
1 0	Ctx0 + Ctx1	1	1

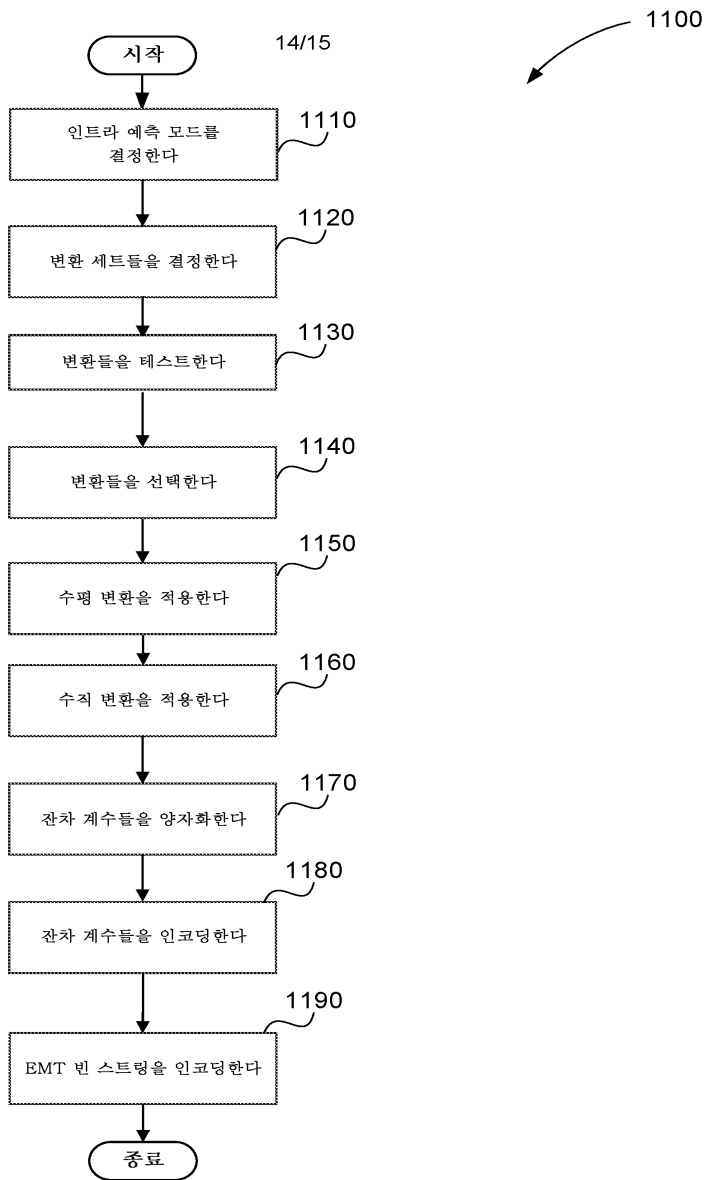
도면10b

1050



빈 스트링	이진화	V	H
0	Ctx0	0	0
1 0	Ctx0 + Ctx1	0	1
1 1 0	Ctx0 + Ctx1 + ctx2	1	0
1 1 1	Ctx0 + Ctx1 + Ctx2	1	1

도면11



도면12

