(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2018년05월31일
(11) 등록번호 10-1863177
(24) 등록일자 2018년05월25일

(51) 국제특허분류(Int. Cl.)
H04N 19/436 (2014.01) *H04N 19/13* (2014.01)
(52) CPC특허분류
H04N 19/436 (2015.01)
H04N 19/13 (2015.01)
(21) 출원번호 10-2017-7017592(분할)
(22) 출원일자(국제) 2013년01월21일
심사청구일자 2017년06월26일
(85) 번역문제출일자 2017년06월26일
(65) 공개번호 10-2017-0077295
(43) 공개일자 2017년07월05일
(62) 원출원 특허 10-2014-7034587
원출원일자(국제) 2013년01월21일
심사청구일자 2014년12월09일
(86) 국제출원번호 PCT/EP2013/051043
(87) 국제공개번호 WO 2013/107906
국제공개일자 2013년07월25일
(30) 우선권주장
61/588,849 2012년01월20일 미국(US)
(56) 선행기술조사문헌
Gordon Clare et al., 'Wavefront Parallel Processing for HEVC Encoding and Decoding', JCTVC of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 6th Meeting: Torino, IT, F274, (2011.07.14.-22.)
RADE KUTIL, 'OPTIMIZATION OF BITSTREAM ASSEMBLY IN PARALLEL MULTIMEDIA COMPRESSION', Scalable Computing: Practice and Experience, 2001
KR1020100072347 A
KR1020070064642 A

(73) 특허권자
지이 비디오 컴프레션, 엘엘씨
미국 뉴욕 12211 올버니 사우스우즈 블러바드 8
(72) 발명자
쉬를, 토마스
독일 10437 베를린 둔커슈트라쎄 72
게오르게, 발레리
독일 10365 베를린 존-지크-슈트라쎄 24
(뒷면에 계속)
(74) 대리인
윤의섭, 김수진

전체 청구항 수 : 총 22 항 심사관 : 조우연

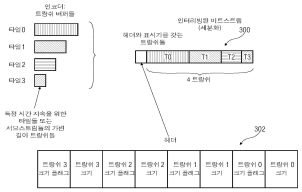
(54) 발명의 명칭 **병렬 처리, 전송 디멀티플렉서 및 비디오 비트스트림을 허용하는 코딩 개념**

(57) 요 약

슬라이스들, 파면 병렬 처리 서브스트림들 또는 타일들 내의 화상을 기술하고 슬라이스들, 파면 병렬 처리 서브스트림들 또는 타일들의 트랑쉬들 내의 인코더로부터 콘텍스트 적응 이진 산술 코딩을 사용하여 코딩되는 초기 바이트 시퀀스 페이로드는 트랑쉬 경계들을 가로질러 콘텍스트-적응적 이진 산술 코딩 확률 적응의 지속과 함께

(뒷면에 계속)

대 표 도



트랑쉬들로 분할되거나 또는 초핑된다. 이러한 측정에 의해, 슬라이스들, 파면 병렬 처리 서브스트림들 또는 타일들 내에 부가적으로 도입된 트랑쉬 경계들은 이러한 요소들의 엔트로피 코딩 효율의 감소에 이르지 않는다. 그러나, 다른 한편으로는, 트랑쉬들은 원래의 슬라이스들,파면 병렬 처리 서브스트림들 또는 타일들보다 작으며 따라서 그것들은 초핑되지 않은 원래 엔티티들, 즉, 슬라이스들, 파면 병렬 처리 서브스트림들 또는 타일들보다 낮은 지연으로 일찍 전송된다. 첫 번째 양상과 결합될 수 있는, 또 다른 양상에 따르면, 상응하는 서브스트림들 또는 타일들을 갖는 멀티-스레드 디코더의 역할을 병렬로 할 수 있도록 전송 디멀티플렉서가 네트워크 추상 계층 유닛들 내의 슬라이스들의 데이터를 상응하는 서브스트림들 또는 타일들에 할당하는 것을 가능하게 하기 위하여, 서브스트림 마커 네트워크 추상 계층 유닛들이 비디오 비트스트림의 네트워크 추상 계층 유닛들의 시퀀스 내에 사용된다.

(72) 발명자

그륀버그, 카르스텐

독일 13599 베를린 아딕케스슈트라쎄 43

키르호호퍼, 하이너

독일 10555 베를린 고츠코브스키슈트라쎄 5

헨켈, 아나스타샤

독일 10245 베를린 파렌하이트슈트라쎄 8

마르페, 데트레브

독일 12161 베를린 수에드베스트코르소 70

명 세 서

청구범위

청구항 1

디코더에 있어서,

WPP 서브스트림들이 분할되어져 내부에 도입되는 트랑쉬 경계들을 갖는 트랑쉬들의 인코더로부터, 화상의 최대 코딩 유닛(LCU) 열(row) 마다, 하나의 WPP 서브스트림과 함께 상기 WPP 서브스트림들 내의 화상을 기술하며, 콘텍스트 적응 이진 산술 코딩(CABAC)을 사용하여 코딩되어지는 초기 바이트 시퀀스 페이로드(raw byte sequence payload)를, 수신하고;

상기 WPP 서브스트림들 내에 도입되는 상기 트랑쉬 경계들에 걸쳐서 지속적인 콘텍스트 적응 이진 산술 코딩(CABAC) 확률 적응을 이용하여 상기 트랑쉬들을 엔트로피 디코딩하며; 그리고

예측 잔여 데이터의 재변환을 이용하여 상기 화상을 획득하기 위하여 초기 바이트 시퀀스 페이로드(RBSP)를 디코딩하도록 구성되어 있는 것을 특징으로 하는 디코더.

청구항 2

제1항에 있어서,

상기 트랑쉬들은 슬라이스 헤더들을 사용하여 패킷으로 패킷화되며,

상기 디코더는 상기 트랑쉬들을 수신하는데 있어서, 새로운 슬라이스를 수신하게 되면, CABAC 확률을 재설정하여 CABAC 확률 적응을 중지하거나, 또는 CABAC 확률 적응을 지속시키기 위해서, 새로운 슬라이스의 슬라이스 헤더 내의 플래그, 상기 새로운 슬라이스의 슬라이스 형태 또는 상기 새로운 슬라이스를 포함하는 네트워크 추상 계층(NAL) 유닛의 NAL 유닛 형태에 반응을 보이도록 구성되는 것을 특징으로 하는 디코더.

청구항 3

제1항 또는 제2항에 있어서,

상기 디코더는 상기 트랑쉬들을 수신하는데 있어서, 각각의 트랑쉬에 대하여, 각각의 상기 트랑쉬가 어떠한 WPP 서브스트림에 속하는지에 대하여 식별함으로써, 상기 트랑쉬들을 탈-인터리빙(deinterleave)하도록 구성되는 것을 특징으로 하는 디코더.

청구항 4

제1항 또는 제2항에 있어서,

각각의 패킷이, 상기 WPP 서브스트림들 또는 타일들 중에서 정의된 순서로 배치되는, 상기 화상의 각각의 상기 WPP 서브스트림의 하나의 트랑쉬, 또는 상기 화상의 상기 WPP 서브스트림의 서브셋을 포함하도록, 상기 트랑쉬들은 패킷으로 패킷화되며,

각각의 상기 패킷은, 각각의 상기 패킷으로 패킹된 상기 트랑쉬들의 위치들 또는 길이들을 나타내는 정보를 포함하는 헤더, 또는 서로 각각의 상기 패킷 내의 상기 트랑쉬들을 분리하는 마커들을 포함하며,

상기 디코더는,

상기 초기 바이트 시퀀스 페이로드를 수신하는데 있어서, 상기 패킷들 내의 상기 트랑쉬들을 액세스하기 위하여 상기 헤더들 또는 상기 마커들에 의해 포함되는 정보를 사용하도록 구성되는 것을 특징으로 하는 디코더.

청구항 5

제 4항에 있어서,

- 상기 WPP 서브스트림들 또는 타일들 중에서 정의된 순서에 따라 -

상기 화상의 상기 WPP 서브스트림들 또는 타일들의 첫 번째 트랑쉬들을 포함하는 패킷들은, 저 지연 특징 표시기(low delay feature indicator)를 포함하며,

- 상기 WPP 서브스트림들 또는 타일들 중에서 정의된 순서에 따라 -

상기 화상의 상기 WPP 서브스트림들 또는 타일들의 두 번째 또는 뒤따르는 트랑쉬들은, 연속 표시기(continuation indicator)를 포함하는 것을 특징으로 하는 디코더.

청구항 6

제 4항에 있어서,

상기 패킷들은 네트워크 추상 계층(NAL) 유닛들 또는 슬라이스들인 것을 특징으로 하는 디코더.

청구항 7

디코더에 있어서,

WPP 서브스트림들 내의 화상을 기술하며, 콘텍스트 적응 이진 산술 코딩(CABAC)을 사용하여 코딩되어 있는 초기 바이트 시퀀스 페이로드를, 상기 WPP 서브스트림들이 분할되어져 내부에 도입되는 트랑쉬 경계들을 갖는 트랑쉬들의 인코더로부터 수신하고;

상기 WPP 서브스트림 중 하나의 트랑쉬를 엔트로피 디코딩하는 것을 시작하는데 있어서, 상기 WPP 서브스트림의 또 다른 트랑쉬를 엔트로피 디코딩하는 단계의 종료 부분에서, 상기 콘텍스트 적응 이진 산술 코딩(CABAC) 확률을 채택하는 것에 의해, 상기 WPP 서브스트림들 내에 도입되는 상기 트랑쉬 경계들에 걸쳐서 지속적인 콘텍스트 적응 이진 산술 코딩(CABAC) 확률 적응을 이용하여 상기 트랑쉬들을 엔트로피 디코딩하며; 그리고,

예측 잔여 데이터의 재변환을 이용하여 상기 화상을 획득하기 위하여 초기 바이트 시퀀스 페이로드를 디코딩하도록 구성되어 있는 것을 특징으로 하는 디코더.

청구항 8

제1항 또는 제 7항에 있어서,

상기 초기 바이트 시퀀스 페이로드는, 상이한 관점들에 대응하는 계층으로 인코딩된 장면(scene)을 가지는 것을 특징으로 하는 디코더.

청구항 9

제1항 또는 제 7항에 있어서,

상기 초기 바이트 시퀀스 페이로드는, 계층으로 인코딩된 화상을 가지는 것을 특징으로 하는 디코더.

청구항 10

인코더에 있어서,

콘텍스트-적응 이진 산술 코딩(CABAC)을 사용하여 초기 바이트 시퀀스를 엔트로피 인코딩하면서, 화상의 최대 코딩 유닛(LCU) 열(row) 마다(per) 하나의 WPP 서브스트림을 갖는 WPP 서브스트림들 내의 화상을 기술하기 위하

여, 상기 화상을 인코딩하여, 초기 바이트 시퀀스 페이로드를 형성하고,

상기 WPP 서브스트림들이 분할되어져 내부에 도입되는 트랑쉬 경계들을 갖는 트랑쉬들 내의 상기 초기 바이트 시퀀스를 전송하며,

상기 WPP 서브스트림들 내에 도입되는 상기 트랑쉬 경계들에 걸쳐서 상기 엔트로피 인코딩을 시행하는 경우에 콘텍스트-적응 이진 산술 코딩(CABAC) 확률 적응을 계속하도록 구성되어 있으며,

상기 인코더는, 예측 잔여 데이터의 변환을 이용하여 상기 초기 바이트 시퀀스 페이로드를 형성하는 것을 특징으로 하는 인코더.

청구항 11

제 10항에 있어서,

상기 인코더는 상기 트랑쉬들이 최대 전송 유닛 크기와 일치하도록 상기 초기 바이트 시퀀스를 형성하도록 구성되는 것을 특징으로 하는 인코더.

청구항 12

제10항에 있어서,

상기 초기 바이트 시퀀스 페이로드는, 상이한 관점들에 대응하는 계층으로 인코딩된 장면을 가지는 것을 특징으로 하는 인코더.

청구항 13

제10항에 있어서,

상기 초기 바이트 시퀀스 페이로드는, 계층으로 인코딩된 화상을 가지는 것을 특징으로 하는 인코더.

청구항 14

비디오 비트스트림을 저장한 디지털 저장 매체에 있어서,

콘텍스트 적응 이진 산술(CABAC)을 사용하여 코딩되고, 화상의 최대 코딩 유닛 열 마다 하나의 WPP 서브스트림을 갖는 WPP 서브스트림들 내의 상기 화상을 기술하는 초기 바이트 시퀀스 페이로드를 포함하며,

상기 비디오 비트스트림은,

상기 WPP 서브스트림들 내에 도입되는 트랑쉬 경계들에 걸쳐, 지속적인 콘텍스트 적응 이진 산술 코딩(CABAC) 확률 적응을 이용함으로써, 상기 WPP 서브스트림들이 분할되어져 내부에 도입되는 트랑쉬 경계들을 갖는 상기 WPP 서브스트림의 트랑쉬들로 분해되며,

각각의 상기 트랑쉬는 각각의 상기 트랑쉬가 속하는 상기 WPP 서브스트림들이 순차적으로 분해되는 트랑쉬들 중에서 그 순위의 분명한 표시를 포함하며,

상기 초기 바이트 시퀀스 페이로드는 예측 잔여 데이터의 변환을 이용하여 내부에 상기 화상을 인코딩하는 것을 특징으로 하는, 비디오 비트스트림을 저장한 디지털 저장 매체.

청구항 15

제 14항에 있어서,

각각의 패킷이, 상기 WPP 서브스트림들 또는 타일들 중에서 정의된 순서로 배치되는, 상기 화상의 각각의 상기

WPP 서브스트림 또는 타일의 한 개의 트랑쉬, 또는 상기 화상의 상기 WPP 서브스트림들 또는 타일의 서브셋을 포함하도록, 상기 트랑쉬들은 패킷으로 패킷화되며,

각각의 상기 패킷은, 각각의 상기 패킷 내에 패킹된 상기 트랑쉬들의 위치들 또는 길이들을 나타내는 정보를 포함하는 헤더, 또는 서로 각각의 상기 패킷 내의 상기 트랑쉬들을 분리하는 마커들을 포함하는 것을 특징으로 하는, 비디오 비트스트림을 저장한 디지털 저장 매체.

청구항 16

제 14항 또는 15항에 있어서,

- 상기 WPP 서브스트림들 또는 타일들 중에서 정의된 순서에 따라 -

상기 화상의 상기 WPP 서브스트림들 또는 타일들의 첫 번째 트랑쉬들을 포함하는 패킷들은, 저 지연 특징 표시기를 포함하고,

- 상기 WPP 서브스트림들 또는 타일들 중에서 정의된 순서에 따라 -

상기 화상의 상기 WPP 서브스트림들 또는 타일들의 두 번째 또는 뒤따르는 트랑쉬들은 연속 표시기를 포함하는 것을 특징으로 하는, 비디오 비트스트림을 저장한 디지털 저장 매체.

청구항 17

제 15항에 있어서,

상기 패킷들은 네트워크 추상 계층(NAL) 유닛들 또는 슬라이스들인 것을 특징으로 하는, 비디오 비트스트림을 저장한 디지털 저장 매체.

청구항 18

제14항에 있어서,

상기 초기 바이트 시퀀스 페이로드는, 상이한 관점들에 대응하는 계층으로 인코딩된 장면(scene)을 가지는 것을 특징으로 하는, 비디오 비트스트림을 저장한 디지털 저장 매체.

청구항 19

제14항에 있어서,

상기 초기 바이트 시퀀스 페이로드는, 계층으로 인코딩된 화상을 가지는 것을 특징으로 하는, 비디오 비트스트림을 저장한 디지털 저장 매체.

청구항 20

디코딩 방법에 있어서,

WPP 서브스트림들이 분할되어 내부에 도입되는 트랑쉬 경계들을 갖는 상기 WPP 서브스트림들의 트랑쉬들 내의 인코더로부터, 화상의 최대 코딩 유닛 열 마다 하나의 WPP 서브스트림을 갖는 상기 WPP 서브스트림들 내의 화상을 기술하며, CABAC을 사용하여 코딩되는 초기 바이트 시퀀스 페이로드를, 수신하는 단계;

상기 WPP 서브스트림들 내부에 도입되는 상기 트랑쉬 경계들에 걸쳐서 지속적인 CABAC 확률 적응을 이용하여 상기 트랑쉬들을 엔트로피 디코딩하는 단계; 및,

예측 잔여 데이터의 재변환을 이용하여 상기 화상을 획득하기 위하여 초기 바이트 시퀀스 페이로드(RBSP)를 디

코딩하도록 구성되어 있는 것을 특징으로 하는 디코딩 방법.

청구항 21

인코딩 방법에 있어서,

CABAC을 이용하여 초기 바이트 시퀀스를 엔트로피 인코딩하면서, 화상의 최대 코딩 유닛 열 마다 하나의 WPP 서브스트림을 갖는 WPP 서브시스템들 내의 상기 화상을 기술하기 위하여, 상기 화상을 인코딩하여, 초기 바이트 시퀀스 페이로드를 형성하는 단계;

상기 WPP 서브스트림들이 분할되어 내부에 도입되는 트랑쉬 경계들을 갖는 트랑쉬들 내의 상기 초기 바이트 시퀀스를 전송하는 단계; 및

상기 WPP 서브스트림들 내에 도입되는 트랑쉬 경계들에 걸쳐서 상기 엔트로피 인코딩을 실행할 때에 CABAC 확률 적응을 지속하는 단계;를 포함하며,

상기 초기 바이트 시퀀스 페이로드는 예측 잔여 데이터의 변환을 이용하여 형성되는 것을 특징으로 하는, 인코딩 방법

청구항 22

컴퓨터 상에 구동될 때, 제 20항 내지 21항 중 어느 한 항에 따른 방법을 실행하기 위한 프로그램 코드를 가지는 컴퓨터 프로그램을 저장한 디지털 저장 매체.

청구항 23

삭제

발명의 설명

기 술 분 야

[0001] 본 발명은 진화하는 고효율 비디오 코딩(High Efficiency Video Coding, HEVC)에서와 같이 병렬 처리(Parallel processing), 전송 디멀티플렉서(transport demultiplexer) 및 비디오 비트스트림을 허용하는 코딩 개념에 관한 것이다.

배 경 기 술

[0002] 인코더와 디코더의 병렬화는 고효율 비디오 코딩 표준뿐만 아니라 비디오 해상도의 예상되는 증가에 의한 증가하는 처리 필요 조건 대문에 매우 중요하다. 다중-코어 아키텍처(multi-core architecture)들이 다양한 범위의 현대 전자 장치들에서 이용가능하게 되었다. 그 결과, 다중-코어 아키텍처들의 사용을 가능하게 하는 효율적인 방법들이 필요하다.

[0003] 최대 코딩 유닛(largest coding unit, LCU)의 인코딩 또는 디코딩은 콘텍스트 적응 이진 산술 코딩(CABAC) 확률들이 각각의 이미지의 특이성들에 적용되는, 래스터 스캔(raster scan)에서 발생한다. 인접한 최대 코딩 유닛들 사이에 공간 의존도가 존재한다. 각각의 최대 코딩 유닛은 예를 들면, 모션-벡터, 예측, 인트라 예측 등의 서로 다른 컴포넌트들 때문에, 그것의 왼쪽, 위, 위-왼쪽 및 위-오른쪽 이웃 최대 코딩 유닛들에 의존한다. 디코딩에서의 병렬화를 가능하도록 하기 위하여, 이러한 의존성들은 일반적으로 최신 적용들에서 인터럽트될(interrupted) 필요가 있거나 또는 인터럽트된다.

[0004] 일부 병렬화 개념, 주로 엔트로피 슬라이스를 사용하는 파면(wavefront) 처리[3], 서브스트림들[2], [4], [11], 또는 타일(tile)들[5]을 사용하는 파면 병렬 처리(wavefront parallel processing, WPP) 운영들이 제안되어 왔다. 후자는 디코더 또는 인코더에서 병렬화를 허용하기 위하여 반드시 파면 처리와 결합될 필요는 없다. 이러한 관점에서 보면, 타일들이 파면 병렬 처리와 유사하다. 엔트로피 슬라이스의 뒤따르는 연구를 위한 본 발명자들

의 초기 동기요인은 코딩 효율 손실을 낮추고 따라서 인코더와 디코더에서의 병렬화 접근법을 위한 비트스트림에 대한 부담을 감소시키는, 기술을 실행하는 것이다.

[0005] 특히 최대 코딩 유닛들의 사용의 더 나은 이해를 제공하기 위하여, 먼저 H.264/AVC의 구조를 볼 수 있다[1].

[0006] H.264/AVC에서의 코딩된 비디오 시퀀스는 네트워크 추상 계층(network abstract layer, NAL) 유닛 스트림 내에 수집되는 일련의 액세스 유닛들로 구성되며 그것들은 단지 하나의 시퀀스 파라미터 세트를 사용한다. 각각의 비디오 시퀀스는 독립적으로 디코딩될 수 있다. 코딩된 시퀀스는 코딩된 화상(picture)들의 시퀀스로 구성된다. 코딩된 프레임은 전체 프레임 또는 단일 필드일 수 있다. 각각의 화상은 고정된 크기의 매크로블록(macroblock, 고효율 비디오 코딩에서 최대 코딩 유닛들)으로 분할된다. 일부 매크로블록들 또는 최대 코딩 유닛들은 하나의 슬라이스로 함께 병합될 수 있다. 따라서 화상은 하나 또는 그 이상의 슬라이스의 수집이다. 이러한 데이터 분리의 목적은 다른 슬라이스들로부터의 데이터의 사용 없이 슬라이스에 의해 표현되는, 화상의 영역 내의 샘플들의 독립적인 디코딩을 허용하는 것이다.

[0007] 종종 "엔트로피 슬라이스"로서 언급되는 기술[3]은 종래의 슬라이스를 부가적인 서브-슬라이스들로 분할하는 것이다. 특히, 이는 단일 슬라이스의 엔트로피 코딩된 데이터의 슬라이싱을 의미한다. 슬라이스 내의 엔트로피 슬라이스들의 배치는 서로 다른 다양성을 가질 수 있다. 가장 간단한 것은 프레임 내의 최대 코딩 유닛들/매크로블록들의 각각의 열(row)을 하나의 엔트로피 슬라이스로서 사용하는 것이다. 엔트로피 슬라이스들로서 대안의, 칼럼들 또는 개별 영역들이 사용될 수 있는데, 이는 인터럽트되고 서로, 예를 들면 도 1의 슬라이스 1에 대하여 토글될(toggled) 수 있다.

[0008] 엔트로피 슬라이스 개념의 명백한 목적은 디코딩 처리의 시간을 향상시키기 위하여, 즉, 처리 속도를 올리기 위하여 병렬 중앙처리유닛(CPU)/그래픽처리유닛(GPU) 및 다중-코어 아키텍쳐들의 사용을 가능하게 하는 것이다. 현재의 슬라이스는 다른 슬라이스 데이터의 참조 없이 파스되거나(parsed) 또는 재구성될 수 있는 파티션들로 분할될 수 있다. 비록 엔트로피 슬라이스 접근법으로 몇 가지 장점들이 달성될 수 있으나, 그렇게 함으로써 일부 불이익이 나타난다.

[0009] 엔트로피 슬라이스 개념은 [2], [10], [11]에서 제안된 것과 같이 서브스트림 파면 처리로 더 확대되었으며 부분적으로 [5]와 통합되었다. 여기서 서브스트림들의 반복 전략이 엔트로피 슬라이스와 비교하여 라인당 향상된 엔트로피 상태 초기화를 갖는 것을 정의된다.

[0010] 타일 개념은 코딩되려는 화상 정보의 분리를 허용하며, 각각이 타일은 고유의 래스터 스캔 순서를 갖는다. 타일은 프레임 내에 반복되는, 공통 구조에 의해 정의된다. 타일은 또한 최대 코딩 유닛들 또는 코딩 유닛들과 관련하여 특정 칼럼 폭 및 라인 높이를 가질 수 있다. 타일들은 또한 독립적으로 인코딩될 수 있으며 또한 디코더 스레드(decoder thread)들이 액세스 유닛의 타일들을 완전히 또는 독립적인 방법으로 일부 코딩 운영 단계, 즉, 엔트로피 코딩 및 변환 코딩을 위하여 처리할 수 있는 같이, 그것들이 다른 타일들과의 공동 처리를 필요로 하지 않는 방법으로 인코딩될 수 있다.

[0011] 따라서 타일은 후자의 경우에 있어서, 예를 들면, 고효율 비디오 코딩 코덱의 필터링 단계에서 타일 인코더들뿐만 아니라 디코더들을 완전히 또는 병렬 방법과 독립적으로 실행하도록 허용한다.

[0012] 비디오 통신 시스템 또는 유사시스템들의 캡처링(capturing), 인코딩, 전송, 디코딩 및 프레젠테이션 체인에서의 병렬화 기술을 완전히 사용하기 위하여, 통신 참여자 사이의 전송 및 액세스는 전체 종단 간 지연 인젝션(end-to-end delay injection)을 위한 중요하고 시간 소비적 단계이다. 만일 타일들, 서브스트림들 또는 엔트로피 슬라이스들과 같은, 병렬화 기술들을 사용하면, 이는 특히 문제가 된다.

[0013] 파면 병렬 처리 서브스트림들의 데이터 접근법들은 파티션들의 코딩된 데이터가 만일 처리되면, 데이터 지역성(data locality)을 갖지 않는다는 사실을 나타내는데, 즉 액세스 유닛의 단일 스레드 디코딩(single threaded decodong)은 그 다음의 파면 병렬 처리 서브스트림 라인의 액세스하기 위하여 잠재적으로 큰 메모리 부분들을 넘어 점프할 필요가 있다. 멀티-스레드 디코딩 시스템은 완전한 병렬 방법으로 작업하기 위하여, 특정 데이터, 즉, 파면 병렬 처리 서브스트림들 상에서 전송을 기다릴 필요가 있으며, 따라서 파면 처리를 이용한다.

[0014] 비디오-스트리밍(video-streaming)에 있어서, 높은 해상도(풀(full)-HD, 쿼드-HD 등)는 전송되어야만 하는 높은 데이터 양에 이르게 한다. 영상 회의(＜145 ms), 또는 게임 애플리케이션(＜145 ms)과 같은, 이른바 저-지연 사용의 경우인, 분초를 다투는 시나리오들을 위하여, 매우 낮은 종단 간 지연들이 필요하다. 따라서, 전송 시간은 임계 인자가 된다. 영상 회의 적용을 위하여, ASDL의 업-로드 링크(up-load link)가 고려된다. 여기서, 대개 I-

프레임들을 언급하는, 이른바 스트림의 랜덤 액세스 지점들이 전송 동안에 장애를 야기하는 후보군들일 것이다.

[0015] 고효율 비디오 코딩은 디코더에서뿐만 아니라 디코더 면에서 이른바 파면 처리뿐만 아니라 타일 처리를 허용한다. 이는 엔트로피 슬라이스들, 파면 병렬 처리 서브스트림들, 또는 심지어 그것들의 조합의 사용에 의해 이용 가능해진다. 병렬 처리는 또한 병렬 타일 인코딩 및 디코딩에 의해 허용된다.

[0016] "비-병렬화 표적화(non-parallelization targeting)"의 경우에 있어서, 전체 슬라이스의 데이터는 즉시 전달될 수 있으며, 따라서 슬라이스들의 마지막 코딩 유닛(CU)은 만일 전송되었으면, 디코더에 의해 액세스 가능하다. 만일 단일 스레드 디코더가 존재하면, 이는 문제가 되지 않는다.

[0017] 멀티-스레드(multi-threaded)의 경우에 있어서, 만일 다중 중앙 처리 유닛 또는 코어가 사용될 수 있으면, 인코딩된 데이터가 파면-디코더 또는 타일-디코더 스레드들에 도착하자마자 디코딩 과정을 시작하려고 할 것이다.

[0018] 따라서, 코딩 효율의 덜 심각한 감소를 갖는 병렬 처리 환경들에서의 코딩 지연의 감소를 가능하게 하는 개념들을 갖는 것이 바람직할 수 있다.

발명의 내용

해결하려는 과제

[0019] 따라서, 병렬 처리 환경에서 그러한 더 효율적이고 낮은 지연 코딩을 가능하게 하는 코딩 개념, 전송 디멀티플렉싱 개념 및 비디오 비트스트림을 제공하는 것이 본 발명의 목적이다.

과제의 해결 수단

[0020] 이러한 목적은 첨부된 독립항들의 주제에 의해 달성된다.

[0021] 본 발명의 첫 번째 양상에 따르면, 슬라이스들, 파면 병렬 처리 서브스트림들 또는 타일들 내의 화상을 설명하고 콘텍스트-적응 이진 산술 코딩을 사용하여 코딩된 초기 바이트 시퀀스 페이로드(raw byte sequence payload, RBSP)는 트랑쉬(tranche) 경계들을 가로질러 콘텍스트-적응적 이진 산술 코딩 확률 적응의 지속과 함께 트랑쉬들로 분할되거나 또는 초핑된다(chopped). 이러한 측정에 의해, 슬라이스들, 파면 병렬 처리 서브스트림들 또는 타일들 내에 부가적으로 도입된 트랑쉬 경계들은 이러한 요소들의 엔트로피 코딩 효율의 감소에 이르지 않는다. 그러나, 다른 한편으로는, 트랑쉬들은 원래의 슬라이스들,파면 병렬 처리 서브스트림들 또는 타일들보다 작으며 따라서 그것들은 초핑되지 않은 원래 엔티티들, 즉, 슬라이스들, 파면 병렬 처리 서브스트림들 또는 타일들보다 낮은 지연으로 일찍 전송된다.

발명의 효과

[0022] 첫 번째 양상과 결합될 수 있는, 또 다른 양상에 따르면, 상응하는 서브스트림들 또는 타일들을 갖는 멀티-스레드(multi-threaded) 디코더의 역할을 병렬로 할 수 있도록 전송 디멀티플렉서가 네트워크 추상 계층 유닛들 내의 슬라이스들의 데이터를 상응하는 서브스트림들 또는 타일들에 할당하는 것을 가능하게 하기 위하여, 서브스트림 마커 네트워크 추상 계층 유닛들이 비디오 비트스트림의 네트워크 추상 계층 유닛들의 시퀀스 내에 사용된다.

도면의 간단한 설명

[0023] 본 발명의 구현들은 종속항들의 주제이다. 또한, 본 발명의 바람직한 실시 예들이 도면들과 관련하여 아래에 더 상세히 설명된다.

도 1은 엔트로피 슬라이스들의 가능한 콤파운드를 개략적으로 도시한다.

도 2는 세 개의 슬라이스에 확산된 세 개의 타일을 개략적으로 도시한다.

도 3은 4개의 가변 길이 트랑쉬 순환 인터리빙(interleaving) 전략의 트랑쉬들의 인터리빙 실시 예를 개략적으로 도시한다.

도 4는 엔트로피 슬라이스 데이터의 인코딩, 세그멘테이션(segmentation), 인터리빙 및 디코딩을 개략적으로 도시한다.

도 5는 항상 마커 코드들을 사용하고 실제 슬라이스 데이터를 다중 네트워크 추상 계층 유닛들로 확산하는 4개

의 가변 길이 트랑쉬 순환 인터리빙 전략의 트랑쉬들의 인터리빙 실시 예를 개략적으로 도시한다. 파티션이 존재하지 않더라도 마커들이 사용된다. 이는 트랑쉬 식별자(tranche identifier)를 사용하고, 그 뒤에 트랑쉬 수를 나타내는, 마커를 사용하여 더 향상될 수 있다. 이는 순환 방식을 위하여 필요한 것과 같이, 항상 마커를 보내는 필요성을 쓸모없게 한다.

도 6은 네트워크 추상 계층 유닛 구문을 나타내는 슈도코드(pseudocode)의 테이블을 도시한다.

도 7은 시퀀스 파라미터 세트 구문을 나타내는 슈도코드의 테이블을 도시한다.

도 8은 저 지연 슬라이스 계층(low delay slice layer) 초기 바이트 시퀀스 페이로드를 나타내는 슈도코드의 테이블을 도시한다.

도 9는 슬라이스 헤더 구문을 나타내는 슈도코드의 테이블을 도시한다.

도 10은 서브스트림 마커 구문을 나타내는 슈도코드의 테이블을 도시한다.

도 11은 엔트로피 슬라이스 데이터의 간단한 캡슐화를 위한 일 실시 예를 개략적으로 도시한다(AF는 MPEG-2 전송 스트림(Transport Stream, TS) 적응 필드(Adaptation Field)이다).

도 12는 엔트로피 슬라이스 데이터의 단일 기본 스트림 캡슐화를 위한 또 다른 실시 예를 개략적으로 도시한다.

도 13은 엔트로피 슬라이스 데이터의 패킹된(packed) 다중-기본 스트림 캡슐화를 개략적으로 도시한다.

도 14는 단일 기본 스트림을 위한 전송 디멀티플렉서를 나타내는 개략적인 블록 다이어그램을 도시한다.

도 15는 다중-기본 스트림을 위한 전송 디멀티플렉서를 나타내는 개략적인 블록 다이어그램을 도시한다.

도 16은 인코더를 나타내는 개략적인 블록 다이어그램을 도시한다.

도 17은 디코더를 나타내는 개략적인 블록 다이어그램을 도시한다.

도 18은 디코더에 의해 실행되는 단계들의 플로차트를 도시한다.

도 19는 실시간 전송 프로토콜(Real-time Transport Protocol, RTP)을 사용하는 다중-기본 스트림을 위한 일 실시 예를 개략적으로 도시한다.

발명을 실시하기 위한 구체적인 내용

[0024] 병렬 디코더 스레드가 프레임이 데이터를 시작하고 종료하는 시간을 감소시키기 위하여, 아래의 실시 예들은 저 지연 인터리빙 접근법에 의한 작은 트랑쉬(tranche) 내로의 하나 또는 그 이상의 타일의 데이터 혹은 하나 또는 그 이상의 파면 병렬 처리 서브스트림의 데이터와 같은, 병렬화를 위하여 구조화된, 데이터의 세그멘테이션을 사용한다.

[0025] 따라서 인코더는 최대 코딩 유닛 혹은 인코더로부터 디코더로의 전송 경로를 거쳐 디코더에 대한 트랑쉬 형태의 그것들의 서브스트림 또는 타일 또는 부분의 적어도 바이드 정렬 부분에 상응하는, 데이터를 전송할 수 있다.

[0026] 트랑쉬들이 완전한 파면 병렬 처리 서브스트림 또는 타일보다 작거나 및/또는 전송 경로의 실제 최대 전송 유닛(MTU)에 적용될 수 있기 때문에, 다중 파면 병렬 처리 서브스트림 또는 타일의 트랑쉬들은 디코더 면에서의 디코딩하는, 완전한 액세스 유닛의 완결 이전에, 인코더와 디코더 사이의 전송 유닛 내에 배치되도록 하기 위하여, 액세스 유닛의 완전한 파면 병렬 처리 서브스트림들 또는 타일들의 순차적 전송을 사용할 때보다 상당히 일찍 시작할 수 있다.

[0027] 이는 확실히 트랑쉬들의 빠른 전송 및 디코더에서 병렬 디코딩 과정의 이른 시작을 야기한다. 이러한 접근법은 또한 만일 그 다음 프레임의 슬라이스(들) 또는 엔트로피 슬라이스(들)가 프레임 간 참조들의 이용가능성 때문에 그 다음의 프레임의 엔트로피 슬라이스의 디코딩을 위하여 필요한 정보를 기초로 하여 예를 들면, 파면 방식으로 이미 디코딩될 수 있으면, 프레임 경계들을 넘어 적용될 수 있다. 디코딩 순서에 성공한 프레임의 이미 디코딩할 수 있는 데이터는 최대 허용/신호전달 움직임 벡터 길이 혹은 선행 프레임(들)에 대한 데이터 부분들의 의존성을 나타내는 스트림, 또는 파라미터 세트와 같은, 시퀀스-고정 위치 내에 신호전달되도록 사용된 위치를 나타내는 고정 참조 전략 내의 부가적인 정보로부터 유래될 수 있다.

[0028] 화상은 최대 코딩 유닛-열(들) 당 하나의 엔트로피 슬라이스로, 혹은 파면 병렬 처리 서브스트림 또는, 엔트로피 슬라이스 내에 더 포함될 수 있는 열 당 하나의 파면 병렬 처리 서브시스템과 같은 조합을 사용하여 인코딩

될 수 있다. 그러한 데이터 구조들은 디코더 면에서 파면 병렬 처리 기술의 사용에 필요하다. 그렇지 않으면 타일들이 병렬 처리를 허용하도록 사용될 수 있다.

[0029] 인코딩 과정 동안에, 파면 병렬 처리 스트림들 또는 타일들을 포함하는, 각각의 슬라이스의 비트스트림은 인코더와 디코더 사이의 최대 전송 유닛 크기를 일치시키기 위하여 다양한 크기의 트랑쉬들로 분할될 수 있다. 그리고 나서 결과로서 생긴 트랑쉬들은 인터리빙되고 전송을 통과할 수 있으며 최대 전송 유닛의 패킷들에 들어갈 수 있다.

[0030] 디코더 면에서의 처리를 허용하기 위하여, 각각의 트랑쉬 앞과 뒤에, 마커 코드(marker code)가 삽입될 수 있다. 고효율 비디오 코딩에 적절한 마커 코드는 "0x00 00 02"일 수 있으며, 이는 시작 코드 에뮬레이션(emulation) 방지도 통과할 수 있다. 다중 트랑쉬를 포함하는 패킷의 수용 후에, 수신기 또는 디코더는 부가적인 파싱(parsing) 단계를 필요로 하지 않도록 하기 위하여 시작 코드 에뮬레이션 방지 과정 동안에 실제 포함된 비트스트림을 파싱할 수 있다. 예를 들면, 트랑쉬 식별을 위한 두 가지 방식이 존재할 수 있다. 1과 동일한 tranche_id를 갖는 트랑쉬부터 n과 동일한 tranche_id를 갖는 트랑쉬까지, 항상 트랑쉬들의 순환(cyclic) 배치가 존재할 수 있다. 이는 두 번째 일반적인 방법에 대한 안전한 신호전송 데이터일 수 있다. 대안의 방법은 예를 들면, 8 비트 값으로서, tranche_id를 나타내는, 마커 다음의 특정 헤더일 수 있다.

[0031] 저 지연 캡슐화을 위한 인터리빙 전략의 사용 상에서 디코더에 정보를 전송하기 위하여, 네트워크 추상 계층 유닛 헤더 내에 low_delay_flaf가 존재할 수 있다.

[0032] 또 다른 방식은 전송 계층 상의 인터리빙 및 디인터리빙(탈-인터리빙, deinterleaving)일 수 있는데, 즉, 디코딩 과정 외부에 실시간 전송 프로토콜[8][9][13] 또는 MPEG-2 전송 스트림[7] 계층이 존재할 수 있다.

[0033] 따라서, 헤더는 존재하는 트랑쉬 당 바이트들에서의 크기 정보를 포함하는 플래그에 의해 트랑쉬의 존재를 나타내는, 패킷의 앞쪽에 놓일 수 있다. 전송 계층이 디코딩 과정으로부터 분리되기 때문에 마커 코드를 통합할 필요가 없을 수도 있는데, 그 이유는 전송 계층의 부가적인 정보가 그러한 데이터를 디코더를 통과시키기 전에 그래도 제거될 필요가 있기 때문이다. 전송 계층은 그리고 나서 또한 디코더로의 비트스트림 전송을 위하여 데이터를 재정렬한다.

[0034] 가변 길이 헤더는 추가의 멀티플렉싱 계층 상에서 사용될 수 있다. 이러한 멀티플렉싱 계층은 또한 코덱의 부분일 수 있으며 디코더 내의 실제 초기 바이트 시퀀스 데이터(RBSP) 액세스 전에 도입될 수 있다. 한 가지 헤더 전략을 도 3에서 찾을 수 있다. 그러나 또한 길이뿐만 아니라 그것의 표시기(indicator)를 나타내는 각각의 트랑쉬 앞에 직접적으로 헤더가 존재할 수 있다. 위에서 이미 설명된 것과 같이 표시기를 비트스트림 구조들에 매핑하는 필요성이 여전히 존재한다.

[0035] 트랑쉬 크기는 또한 일정한 크기, 예를 들면, 트랑쉬 당 x 바이트일 수 있다. 이는 도 4에 도시된 것과 같은, 단순한 멀티플렉싱 전략을 야기한다.

[0036] 세크먼트들의 일정한 크기는 그것의 가변 길이 때문에 비트스트림의 끝에서 문제를 야기한다.

[0037] 두 가지의 일반적인 가능한 해결책이 존재한다. 한 가자는 순환 x-바이트 세크먼트들의 발생(일반적으로 슬라이스의 비트스트림 표현은 바이트-정렬된다) 및 각각의 디코더-엔진에 의한 바이트들의 소비의 제어인데, 즉, 디코더는 엔트로피 슬라이스의 완성 또는 마커 코드의 포함을 발견한다.

[0038] 두 번째 방법은 트랑쉬들이 도면에 도시된 것과 같이 헤더에서 가변 길이일 때, 트랑쉬 길이들의 신호전송이다.

[0040] *세그먼트의 크기와 인터리빙 방식은 하나의 SEI-메시지 또는 SPS 내에 신호전송될 수 있다.

[0041] 전송 전략이 도 4에 도시된다.

[0042] 또 다른 흥미 있는 방법은 네트워크 추상 계층 또는 슬라이스 패킷과 같은 패킷 내의 트랑쉬들의 세트의 끝에서의 완결 코드(finalizing code)들 또는 마커 코드들의 사용이다. 이러한 경우에 있어서, 가변 길이 세그먼트들이 가능하며, 따라서 비트스트림의 완전한 파싱이 필요하다. 여기서 메모리 액세스를 제한하기 위하여, 멀티플렉싱을 위한 이러한 부가적인 파싱 과정은 네트워크 추상 계층 내에 포함된 초기 바이트 시퀀스 페이로드 데이터를 액세스하기 전에 첫 번째 단계로서 필요한, 시작 코드 에뮬레이션 방지 파싱과 결합될 수 있다. 그러한 마커 개요가 도 5에 도시된다.

[0043] 여기서의 개념은 데이터를 트랑쉬 내로 삽입하는 동안에, 인터리빙 방식으로 실제 슬라이스, 엔트로피 슬라이스 또는 이와 유사한 것과 같은 높은 레벨 구조를 파면 병렬 처리 서브스트림들 또는 타일들과 같은, 그것이 포함

된 낮은 레벨 데이터 구조로 분할하는 것이다. 각각 낮은 레벨 구조에 속하는, 이러한 트랑쉬들은 특정 네트워크 추상 계층 유닛, 도면에 "NAL 유닛 #1"을 도시된 것과 같이, 플래그 또는 슬라이스 형태에 의한 저 지연 인터리빙 접근법을 나타내는 저 지연 인터리빙 플래그 또는 슬라이스 또는 경량 슬라이스 헤더에 의한 부가적인 시그널링을 갖는 네트워크 추상 계층 유닛일 수 있는, 저 지연 패킷 내로 삽입되며, 따라서 디코더는 디코더에서 오리지널/탈-인터리빙된 순서로 트랑쉬들의 순차적 처리를 사용하는, "단일" 스레드 디코더를 위하여 레코딩 기능을 적용하도록 알려진다. 저 지연 특징을 얻도록 다중 패킷들에 대하여 인터리빙된 트랑쉬들로서 실제 슬라이스의 데이터를 분할하기 위하여, 전송 계층은 저 지연 인터리빙된 데이터를 포함하는 네트워크 추상 계층 유닛을 최대 전송 유닛 크기의 네트워크 패킷들에 단편화할 수 있다. 실제 슬라이스 데이터의 다중 네트워크 추상 계층 유닛들로의 단편화(fragmentation)는 또한 코딩 계층에 의해 직접적으로 적용될 수 있으며, 따라서 "NAL 유닛 #2"를 위하여 도 5에 도시된 것과 같이, 슬라이스의 연속을 포함하는 그러한 형태의 네트워크 추상 계층 유닛에 신호를 보내기 위한 필요성이 존재한다. 네트워크 추상 계층 유닛들과 같은, 다중 패킷 내의 인터리빙된 데이터의 완결을 검출하기 위하여, 또한 도면에 "NAL 유닛 #2"를 위하여 도시된 것과 같이 특정 완결 코드 또는 슬라이드 또는 네트워크 추상 계층 헤더에서의 완료를 나타내는 플래그의 필요성이 존재할 수 있다.

[0044] 네트워크 추상 계층 패킷들을 손실하는 경우에 있어서, 또한 손실들을 검출하는 필요성이 존재한다. 이는 헤더, 예를 들면, 포함된 트랑쉬들, 또는 단지 특정 트랑쉬(#1)의 제 1 멀티플렉스 버퍼(MB)들과 같은, 경량 슬라이스 헤더 내의 부가적인 정보에 의해 적용될 수 있다. 파면 병렬 처리 서브스트림들을 위한 오프셋들 또는 트랑쉬의 실제 크기와 같은 정보를 가질 때, 누군가는 또한 완료 코드를 갖는 네트워크 추상 계층 유닛 및 선행 네트워크 추상 계층 유닛들을 수신한 후에 온전성 검사(sanity check)를 수행하기 위하여 이러한 크기 값들(특정 파면 병렬 처리 서브스트림 또는 타일의 오프셋 값들)을 사용할 수 있다.

[0045] 즉, 설명된 것과 같이, 각각의 패킷(300)이 파면 병렬 처리 서브스트림들 또는 타일들 중에서 정의된 순서(#)로 배치되는, 화상의 각각의 파면 병렬 처리 서브스트림 또는 타일, 혹은 화상의 각각의 파면 병렬 처리 서브스트림 또는 타일의 서브셋의 하나의 트랑쉬(T#)를 포함하는 것과 같은 방식으로(그 이유는 예를 들면, 특정 파면 병렬 처리 서브스트림 또는 타일이 선행 패킷들에 의해 이미 완전히 전달되었기 때문에), 트랑쉬들이 패킷들(300) 내로 패킷화될 수 있으며, 각각의 패킷은 각각의 패킷(300) 내로 패킹된 트랑쉬들(T#)의 위치들 및/또는 길이들을 나타내는 정보를 포함하는 헤더(302), 또는 서로 각각의 패킷(300) 내의 트랑쉬들(T#)을 분리하는 마커들(304)을 포함하며, 디코더는 초기 바이트 시퀀스 페이로드를 수신하는데 있어서, 트랑쉬들을 패킷들 내에 액세스하기 위하여 헤더들(302) 또는 마커들(304)에 의해 포함되는 정보를 사용하도록 구성될 수 있다. 첫 번째(파면 병렬 처리 서브스트림들 또는 타일들 중에서 정의된 순서에 따라) 트랑쉬들을 포함하는 패킷들(300a)은 저 지연 특징 표시기(306)를 포함할 수 있으며, 화상의 파면 병렬 처리 서브스트림들 또는 타일들의 두 번째 또는 뒤따르는(파면 병렬 처리 서브스트림들 또는 타일들 중에서 정의된 순서에 따라) 트랑쉬들(T#)은 연속 표시기(continuation indicator, 308)를 포함할 수 있다. 패킷(300)은 네트워크 추상 계층 유닛들 또는 슬라이스들일 수 있다.

[0046] 다음에서, 트랑쉬들 내로의 저 지연 인터리빙을 위한 구문 및 의미(semantic)를 신호전달하기 위한 일 실시 예가 제공된다.

[0047] 그럼에도 불구하고, 파면 병렬 처리 서브스트림 또는 타일의 데이터와 같은, 트랑쉬 데이터의 분할이 또한 위에 설명된 것과 같이 슬라이스 레벨 상에 또는 아래에 적용될 수 있다.

[0048] 이제, 부가적인 처리 단계들을 감소시키기 위하여 시작 코드 에뮬레이션 방지를 위한 파싱과 결합될 수 있는, 접근법이 도시된다. 따라서, 고효율 비디오 코딩 코덱의 초기 바이트 시퀀스 페이로드 레벨에서 인터리빙이 적용된다.

[0049] 트랑쉬는 저 지연 데이터 액세스를 위하여 네트워크 추상 계층 유닛 페이로드 섹션 내에 인터리빙되려는 섹션으로의 초기 바이트 시퀀스 페이로드 데이터의 분할로서 알 수 있다. 트랑쉬의 완결은 코드 0x000002에 의해 표시될 수 있으며 8비트 트랑쉬 식별자 tranche_id가 뒤따를 수 있다. 트랑쉬들을 순환 방식으로 인터리빙될 수 있으며, 따라서 트랑쉬 종료 코드(end code)는 분명하게 유래되는, tranhe_id이 뒤따르지 않는다. 단일 트랑쉬 내의 초기 바이트 시퀀스 페이로드 데이터는 타일의 데이터, 서브스트림의 데이터, 슬라이스의 데이터 또는 엔트로피 슬라이스의 데이터와 상응한다.

[0050] 네트워크 추상 계층 유닛 구문에서, 트랑쉬들의 순환 배치인, "low delay encapsulation_flag"뿐만 아니라 부가적인 식별자 "tranche_id"를 거쳐 트랑쉬의 표시 그 다음에 네트워크 추상 계층 유닛 헤더 내의 "low delay cyclic_flag"와 같은 플래그를 거쳐 마커 코더에 의해 표시되는 것과 같은 저 지연 인터리빙을 위하여 두 가지

방식이 허용될 수 있다. 이러한 두 가지 플래그는 또한 시퀀스 파라미터 세트들, 또는 APS에 존재할 수 있다. 순환 트랑쉬 배치를 위하여, "num_low_delay_tranches"로서 SPS 내에 제공되는 것과 같은, 파싱 동안에 트랑쉬들의 수를 알기 위한 필요성이 여전히 존재한다.

[0051] 네트워크 추상 계층 유닛에서 인터리빙된 "LD_rbsp_byte"는 파서(parser)와 레코더에 의해 네트워크 추상 계층 구문 내의 마지막 포-루프(for-loop)에서의 실제 순차적 초기 바이트 시퀀스 페이로드 순서에 판독된다.

[0052] for(i=0,i++,i<num_low_delay_tranches){

[0053] for(j=0,J++,J<NumBytesInRBSP[i]){

[0054] rbsp_byte[NumBytesInRBSP++]=LD_rbsp_byte[j][i]

[0055] }

[0056] 또한 "low_delay_tranche_lenght_minus"에서 표시된 것과 같이 순환 배치된 트랑쉬들의 고정된 크기를 위하여 SPS 또는 APS 내의 분명한 신호전달의 존재할 수 있다. 후자는 네트워크 추상 계층 유닛 구문 실시 예에서 사용되지 않았으나, 만일 도 4에 도시된 것과 같이 패킷화를 가지고 있다는 것을 고려하면 확실히 알 수 있다. 도 6의 네트워크 추상 계층 유닛 구문에서, 도 5에 도시되고 위에 설명된 것과 같은 패킷화가 기본이었다.

[0057] 슬라이스 및/또는 네트워크 추상 계층 유닛들과 같은, 다중 패킷들에 대한 트랑쉬들의 이러한 인터리빙을 허용하도록 하기 위하여, 이미 수신된 네트워크 추상 계층 유닛들의 초기 바이트 시퀀스 페이로드 데이터로의 반복 액세스를 갖도록 트랑쉬들을 위한 LD_rbsp_byte의 어레이와 같은, 글로벌 버퍼(global buffer)를 위한 필요 조건이 존재할 수 있다.

[0058] 완결 코드를 수신한 후에 오류 내성(error resilience)을 허용하도록 하기 위하여, 또는 만일 트랑쉬를 위하여 수신된 바이트들의 수의 합이 트랑쉬 크기와 동일하면, 이는 포함된 트랑쉬 데이터를 위하여 제공된 것과 같은 오프셋 값으로부터, 즉, 문제의 트랑쉬가 일부인 각각의 파면 병렬 처리 서브스트림 또는 터일에 대한 데이터로부터 유래될 수 있다.

[0059] 인터리빙된 저 지연 트랑쉬들 내에 배치되는 파면 병렬 처리 서브스트림들의 중요한 필요 조건은 트랑쉬(n+1)에 의해 트랑쉬(n)로부터의 데이터만이 액세스되는 것인데, 이는 이미 트랑쉬(n) 내에 제공되고 디코더에서 이미 저장되었거나 이용가능하다는 것이다. 슬라이스 레벨 상의 재정렬/탈-인터리빙을 위한 저 지연 슬라이스 계층 초기 바이트 시퀀스 페이로드 구문은 다음과 같이 디자인될 수 있다. 특히, 그러한 경우에 있어서 구문은 네트워크 추상 계층 유닛 층 상에서와 거의 동일한 행동을 가져야만 하나, 재정렬은 슬라이스 레벨 상에서 정의되어야만 한다. 도 8은 저 지연 슬라이스 계층 초기 바이트 시퀀스 페이로드 구문을 도시한다.

[0060] 인터리빙된 트랑쉬들을 패킷화하기 위하여 슬라이스 헤더를 사용하는 경우에, 만일 콘텍스트 적응 이진 산술 코딩 상태를 재설정하지 않고, 새로운 슬라이스를 수신하면, 코덱 레벨에서 표시하기 위한 필요 조건이 존재할 수 있는데, 그 이유는 트랑쉬들의 엔트로피 코딩, 예를 들면, 파면 병렬 처리 서브스트림이 중단되어서는 안 되기 때문이다. 슬라이스 내에 콘텍스트 적응 이진 산술 코딩을 재설정하지 않는 것은 슬라이스 헤더 내에 "no_cabac_reset_flag"로서 표시된다. 도시된 슬라이스 헤더는 저 지연 슬라이스들에 적합하며, 따라서 또한 entropy_slice 특징들이 존재하여야만 한다. 상응하는 슬라이스 헤더 구문이 도 9에 도시된다.

[0061] 전송 계층은 코딩 계층 내의 다수의 서브스트림/타일/트랑쉬(전송 계층 상에서, 본 발명의 발명자들은 서브스트림, 타일, 서브스트림 또는 타일의 부분, 혹은 유사한 기능을 갖는 비트스트림의 일부분에 의해 표현될 수 있는 추상적 엔티티(abstract entity)를 추정하는데, 병렬 디코딩 또는 점진적 디코더 리프레시(gradual decoder refresh)를 허용한다)가 서로에 관계없이 독립적으로 처리될 수 있는지의 사실을 기초로 하여 디코더 유닛(들)으로 전달되는 데이터의 스케줄링의 최적화를 가능하게 한다. 한 가지 가능성은 트랑쉬들을 병렬로 최소 지연을 갖는 일부 디코딩 유닛들에 보내기 시작하는 것이다. 비트스트림은 전송 계층 상에서 개별적으로 처리될 수 있는 가장 작은 아이템들인 네트워크 추상 계층 유닛들의 시퀀스로 구성된다. 결론적으로, 전송 계층 상의 다음의 처리 방법들은 개별 슬라이스 또는 엔트로피 슬라이스 네트워크 추상 계층 유닛들 내에 포함된 서브스트림들/타일들/트랑쉬들을 기초로 한다.

[0062] 전송 계층은 또한 만일 코딩 계층이 점진적 디코더 리프레시를 사용하는지의 사실을 기초로 하여 디코더 성능 및 오류 내성을 최적화하여야만 한다. 한 가지 옵션은 예를 들면, 전송 오류들 때문에 비트스트림의 이전 부분들이 올바르게 수신되지 않았거나 또는 예를 들면, 전송 채널들 사이의 스위치 때문에 전혀 수신되지 않았으면

비트스트림의 무관한 부분들을 드로핑(dropping)하는 것이다.

[0063] 그러한 이용/최적화를 허용하기 위하여 서로 다른 정보가 전송 계층 상에 신호가 보내진다.

[0064] 일반적인 부가 정보는 기술자(descriptor)를 사용하여 신호가 보내진다:

[0065] - 서브스트림들/타일들의 수, "1"은 전체 비디오 프레임을 포함하는 단지 하나의 스트림/타일을 의미한다.

[0066] - 모든 서브스트림/타일에 공통적인 정보, 예를 들면, 만일 모든 서브스트림/타일이 동일한 크기이거나 버퍼 요구량이 동일할 때

[0067] - 각각의 서브스트림/타일에 관한 개별 정보,

[0068] - 점진적 디코더 리프레시 단계들의 수, "1"은 점진적 디코더 리프레시가 사용되지 않는 것을 의미한다

[0069] - 이러한 서브스트림들/타일들이 저 지연 병렬 처리를 허용하는지를 나타내는 플래그.

[0070] 만일 서브스트림들/타일들의 수가 >1이면, 구문 요소들은 특정 서브스트림/타일을 포함하는 각각의 데이터 블록 앞의 스트림 내에 삽입된다. 이러한 구문 요소들은 네트워크 추상 계층 유닛 구문을 후속하게 하나, 아래에서 서브스트림 마커들로서 언급되는, 코딩 계층에 의해 사용되지 않는 독특한 유닛 형태(예를 들면, nal_unit_type=0x19 또는 nal_unit_type=0x1F)를 사용한다.

[0071] 이러한 구문 요소들은 마커들로서 사용되고 서브스트림/타일을 식별하는 적어도 하나의 데이터 필드를 후속하게 하는 데이터 블록에 관한 정보를 지닌다.

[0072] 만일 점진적 디코더 리프레시 단계들의 수가 >1이면, 이러한 구문 요소들은 또한 서브스트림/타일이 인트라 코딩된지를 나타내는(점진적 디코더 리프레시를 허용하는) 플래그를 지닌다.

[0073] 상응하는 구문이 도 10에 도시된다. 다음의 제약이 적용될 수 있다:

[0074] **forbidden_zero_bit**는 0과 동일하여야만 한다.

[0075] **nal_ref_flag**는 0과 동일하여야만 한다.

[0076] **nal_unit_type**은 0x19와 동일하여야만 한다.

[0077] **substream_ID** : 동일한 화상에 속하는 각각의 또 다른 슬라이스 또는 엔트로피 슬라이스에 의해 증가되는, 화상에 속하는 제 1 슬라이스를 위하여 0으로 시작하는 카운터 값(counter value)

[0078] **is_intra** : 만일 '1'이면, 그 다음의 네트워크 추상 계층 유닛은 인트라 코딩된 슬라이스 또는 인트라 코딩된 엔트로피 슬라이스를 포함한다.

[0079] 전송 멀티플렉스 내의 비디오 스트림의 캡슐화를 위한 방법이 도 11에 도시되는데, 각각의 슬라이스 또는 엔트로피 슬라이스는 전송 스트림 패킷들의 정수 내에 개별적으로 전송된다. 만일 페이로드의 크기가 고정된 크기의 전송 스트림 패킷들 내의 이용가능한 바이트들과 정확하게 맞지 않으면, 마지막 전송 스트림 패킷은 적응 필드를 포함한다.

[0080] MPEG-2 전송 스트림의 기본 스트림의 유사한 행동이 또한 도 19에 도시된 것과 같이 실시간 전송 프로토콜의 실시간 전송 프로토콜 세션 또는 실시간 전송 프로토콜 스트림에 의해 제공된다는 것을 이해하여야 한다. 실시간 전송 프로토콜[8]에서, 실시간 전송 프로토콜 스트림에 표시된 것과 같이 미디어 형태 또는 페이로드 형태에 의해 식별되는[12])는 그것의 고유의 실시간 전송 프로토콜 세션 내에 포함될 수 있으며, 실시간 전송 프로토콜 세션은 (IP) 네트워크 주소, (UDP) 포트뿐만 아니라, 소스 식별자(source identifier, SSRC)에 의해 식별된다. SDP 내에 표시되는 것과 같은 미디어 세션은 각각 서로 다른 미디어 형태를 포함하는, 다중 실시간 전송 프로토콜 세션을 포함할 수 있다. 그러나, 서로 다른 실시간 전송 프로토콜 스트림들 내에 동일한 미디어 스트림(예를 들면, 비디오)을 전송하는 것이 가능한데, 실시간 전송 프로토콜 스트림들은 동일한 실시간 전송 프로토콜 세션(아래의 1과 유사한) 내에 포함될 수 있거나 또는 그들 고유의 실시간 전송 프로토콜 세션들(아래의 2와 유사한) 내에 포함될 수 있다. 도 19는 2의 경우를 도시한다.

[0081] 실시간 전송 프로토콜 페이로드 포맷들[9][13]은 [9][13]에서 설명된 것과 같이 오류 내성 목적을 위하여 고의로 디코딩 순서를 벗어나 전송되는 경우에 수신기에서 네트워크 추상 계층 유닛들의 디코딩 순서를 복구하도록 허용하는, 디코딩 순서 번호((decoding order number, DON)를 갖는다. 따라서, 부가적인 마커들(MKR)은 필요하지 않다. 파면 병렬 처리 서브스트림들 또는 타일들의 트랑쉬들이 인코딩 과정으로부터 이용가능하게 될 때 파

면 병렬 처리 서브스트림들 또는 타일들의 트랑쉬들을 순서대로 전송하는 경우에, 그것들을 단일 디코더에 제공하기 전에 트랑쉬들의 디코딩 순서를 복구하기 위하여 디코딩 순서 번호가 또한 사용될 수 있다. 그러나 이러한 경우에 있어서, 디코딩 과정 전의 개별 탈-인터리빙 과정 때문에 디코더에서 부가적인 지연이 도입될 수 있다. 여기에 설명되는 시스템은 인코딩된 트랑쉬들을 서로 다른 파면 병렬 처리 서브스트림들 또는 타일들의 디코딩 과정에 직접적으로 제공할 수 있으며 데이터는 수신기에 도달한다. 파면 병렬 처리 서브스트림들 또는 타일들과 관련된 트랑쉬들의 식별은 슬라이스 세그먼트의 슬라이스 세그먼트 헤드 내의 슬라이스 주소 및 실시간 전송 프로토콜 헤더 내의 실시간 전송 프로토콜 시퀀스 번호에 의해 표시된 것과 같이 패킷들의 전송 순서에 의해 유래될 수 있다. 이러한 시나리오에서, 디코딩 순서 번호는 단지 하위 호환성(backward compatibility)을 위하여, 즉, 그것들이 도착할 때 디코딩 순서를 벗어나 보내진 파면 병렬 처리 서브스트림들 또는 타일들의 트랑쉬들을 디코딩하는 향상된 능력을 제공하지 않는 디코더들을 위하여 사용된다. 디코딩 순서를 벗어나 트랑쉬 데이터를 송신하는 것은 단지 파면 병렬 처리 서브스트림 및 타일들 레벨과 관련하여, 즉, 전송된 데이터 내에 적용되며, 단일 파면 병렬 처리 서브스트림 또는 타일의 트랑쉬들은 디코딩 순서로 전송되며, 서로 다른 파면 병렬 처리 서브스트림들 또는 타일들의 트랑쉬들은 인터리빙된다.

[0082] 두 가지 가능한 옵션이 존재한다:

[0083] 1. 모든 슬라이스 및 엔트로피 슬라이스들이 동일한 기본 스트림 내에 포함되는데, 즉, 동일한 패킷 식별자(PID)가 그러한 비디오 스트림의 모든 전송 스트림 패킷에 할당된다; 뒤의 내용에서 이러한 방법은 단일 기본 스트림 캡슐화에 언급된다.

[0084] 2. 서로 다른 패킷 식별자들이 동일한 비디오 비트스트림의 슬라이스들 및 엔트로피 슬라이스들에 할당된다; 뒤의 내용에서 이러한 방법은 다중-기본 스트림 캡슐화에 언급된다.

[0085] 도 11은 만일 첫 번째 옵션이 모든 기본 스트림을 위하여 동일한 패킷 식별자의 설정에 의한 더 일반적인 구조의 특별한 경우로 간주되면 두 옵션 모두에 유효하다.

[0086] 단일 기본 스트림 내의 캡슐화를 위한 더 효율적인 방법이 도 12에 도시된다. 여기서, 화상 당 최대한 하나의 적응 필드가 필요하다.

[0087] 다중-기본 스트림 내의 캡슐화를 위한 더 효율적인 방법이 도 13에 도시된다. 여기서, 적응 필드가 방지되며, 그 대신에 또 다른 슬라이스, 예를 들면 그 다음 화상의 병치 타일(collocated tile)이 동일한 전송 스트림 패킷 내에서 즉시 시작한다.

[0088] 멀티-스레드 디코더를 표적화하는 하나의 단일 기본 스트림으로의 캡슐화를 위한 전송 디멀티플렉서의 가능한 구조가 도 14에 도시된다. 도면의 엔트로피 슬라이스는 특정 파면 병렬 처리 서브스트림 또는 타일의 데이터를 포함할 수 있다.

[0089] 전송 버퍼(TB)는 전송 패킷에 속하는 데이터를 수집하고 이를 멀티플렉스 버퍼(MB)에 전달한다. 멀티플렉스 버퍼의 출력에서, 네트워크 추상 계층 유닛 헤더들이 평가되고 서브스트림 마커들이 드로핑되며, 서브스트림 마커 내에 전달된 데이터가 저장된다. 각각의 슬라이스 또는 엔트로피 슬라이스의 데이터는, 일단 디코더 스레드가 이용가능하면 멀티-스레드 디코더에 의해 당겨지는 개별 슬라이스 버퍼(SB) 내에 저장된다.

[0090] 멀티-스레드 디코더를 표적화하는 다수의 기본 스트림들로의 캡슐화를 위한 전송 디멀티플렉서의 가능한 구조가 도 15에 도시된다.

[0091] 다시 말하면 위에 설명된 개념들이 아래에 다시 설명된다. 따라서, 아래의 설명은 위의 설명의 부가적인 상세내용과 개별적으로 조합가능하다.

[0092] 도 16은 본 발명의 일 실시 예에 따른 인코더의 일반적인 구조를 도시한다. 인코더(10)는 멀티-스레드 방식으로 또는 그렇지 않으면 단지 단일-스레드로 운영할 수 있도록 구현될 수 있다. 즉, 인코더(10)는 예를 들면, 다수의 중앙 처리 유닛 커널을 사용하여 구현될 수 있다. 바꾸어 말하면, 인코더(10)는 병렬 처리를 제공할 수 있으나, 반드시 그럴 필요는 없다. 본 발명의 코딩 개념은 병렬 처리 인코더들이 압축 효율을 위해하지(compromising) 않고 병렬 처리를 효율적으로 적용하는 것을 가능하게 한다. 병렬 처리 능력과 관련하여, 유사한 문(statement)들이 디코더를 위하여 유효하며, 이는 도 17과 관련하여 뒤에 설명된다.

[0093] 인코더(10)는 비디오 인코더이나, 일반적으로 인코더(10)는 또한 화상 인코더일 수 있다. 비디오(14)의 화상(14)은 입력(16)에서 인코더(10)로 들어가는 것으로 도시된다.

[0094] 인코더는 하이브리드 인코더일 수 있는데, 즉, 감산기(subtractor)와 같은, 잔여 결정기(residual determiner, 22)에 의해 획득된 것과 같은 예측 잔여(20)는 변환/탈양자화 모듈(24) 내의 이산 코사인 변환(DCT)과 같은 스펙트럼 분해와 같은 변환 및 양자화의 대상이 된다. 이에 따라 획득된 양자화된 잔여(26)는 엔트로피 코더(28) 내의 엔트로피 코딩, 주로 콘텍스트 적응 이진 산술 코딩의 대상이 된다. 디코더를 위하여 이용가능한 것과 같은 잔여의 재구성가능한 버전, 즉, 탈양자화되고 재변환된 잔여 신호(30)는 변환 및 탈양자화 모듈(31)에 의해 복구되고, 결합기(combiner, 33)에 의해 예측기(predictor, 18)의 예측 신호(32)와 결합되며, 그렇게 함으로써 화상(12)의 재구성을 야기한다. 그러나 인코더(10)는 블록 기반으로 운영된다. 따라서, 재구성된 신호(34)는 블록 경계들에서의 불연속성에 의해 영향을 받게 되며, 따라서 어떠한 예측기(18)가 실질적으로 인코딩된 화상들을 예측하는지를 기초로 하여 참조 화상(reference picture, 38)을 생성하기 위하여 필터(36)가 재구성된 신호(34)에 적용될 수 있다. 도 16의 쇄선에 의해 도시된 것과 같이, 그러나 예측기(18)는 또한 필터(36) 없이 직접적으로 재구성된 신호(34)를 이용하거나 또는 중간 버전을 이용할 수 있다. 화상 코딩의 경우에 있어서, 필터(36)는 제외될 수 있다.

[0095] 예측기(18)는 화상(12)의 특정 블록들을 예측하기 위하여 서로 다른 예측 방식들 중에서 선택할 수 있다. 이전에 코딩된 화상들을 기초로 하여 어떠한 블록이 예측되는가에 따른 시간적 예측 방식, 동일한 화상의 이전에 코딩된 블록들을 기초로 하여 어떠한 블록이 예측되는가에 따른 공간적 예측 방식, 낮은 공간 해상도 또는 또 다른 관점에서와 같은, 낮은 계층에서 이러한 신을 나타내는 상응하는 화상을 기초로 하여 높은 공간 해상도 또는 또 다른 관점에서와 같은, 높은 계층에서의 신(scene)을 나타내는 화상의 어떠한 블록이 예측되는가에 따른 계층-간 예측 방식이 존재할 수 있다.

[0096] 양자화된 잔여 데이터, 즉, 변환 계수 레벨들 및 다른 잔여 데이터뿐만 아니라, 예를 들면, 예측기(18)에 의해 결정된 것과 같은 화상(12)의 개별 블록들을 위한 예측 방식들과 예측 파라미터들을 포함하는 코딩 방식을 컴파일(compile)하기 위하여 특정 구문이 사용되며 이러한 구문 요소들은 엔트로피 코더(28)에 의한 엔트로피 코딩의 대상이 된다. 엔트로피 코더(28)에 의한 출력으로서 이에 따라 획득된 데이터 스트림은 초기 바이트 시퀀스 페이로드(40)로 불린다.

[0097] 도 16의 인코더(10)의 요소들은 도 16에 도시된 것과 같이 상호연결된다.

[0098] 도 17은 도 17의 인코더에 꼭 맞는, 즉, 초기 바이트 시퀀스 페이로드를 디코딩할 수 있는 디코더를 도시한다. 도 17의 디코더는 일반적으로 참조 번호 (50)으로 표시되고 엔트로피 디코더(52), 재변환/탈양자화 모듈(54), 결합기(56), 필터(58) 및 예측기(predictor, 60)를 포함한다. 엔트로피 디코더(52)는 초기 바이트 시퀀스 페이로드(40)를 수신하며 잔여 신호(62) 및 코딩 파라미터들(64)을 복구하기 위하여 콘텍스트-적용 이진 산술 디코딩을 사용하여 엔트로피 디코딩을 실행한다. 재변환/탈양자화 모듈(64)은 잔여 데이터(62)를 탈양자화하고 재변환하며 이에 따라 획득된 잔여 신호를 결합기(56)에 전달한다. 결합기(56)는 또한 차례로, 예측 신호(66)와 잔여 신호(65)를 결합함으로써 결합기(56)에 의해 결정된 재구성된 신호(68)를 기초로 하는 코딩 파라미터들(64)을 사용하여 예측 신호(66)를 형성하는 예측기(60)로부터 예측 신호를 수신한다. 도 16과 관련하여 위에서 이미 설명된 것과 같이, 예측기(60)는 대안으로서 또는 부가적으로, 재구성된 신호(68)의 필터링된 버전 또는 그것의 일부 중간 버전을 사용할 수 있다. 디코더(50)의 출력(70)에서 최종적으로 재생산되고 출력되는 화상은 유사하게 조합 신호(68)의 필터링되지 않은 버전 또는 그것의 일부 필터링된 버전 상에서 결정될 수 있다.

[0099] 타일 개념에 따르면, 화상(12)은 타일들로 분할되고 적어도 이러한 타일들 내의 블록들의 예측들은 공간적 예측을 위한 기본으로서, 동일한 타일과 관련된 데이터만을 사용하도록 제한된다. 이러한 측정에 의해 적어도 예측은 각각의 타일을 위하여 병렬로 개별적으로 실행될 수 있다. 단지 설명의 목적을 위하여, 도 16은 화상(12)을 9개의 타일로 분할된 것으로 도시한다. 도 16에 도시된 것과 같이 각각의 타일의 9개 블록으로의 분할은 또한 단지 바람직한 일례이다. 또한, 완전성을 위하여, 타일들을 개별적으로 코딩하는 방법은 공간적 예측(인트라 예측)에 한정되지 않을 수 있다는 것에 유의하여야 한다. 오히려, 타일의 경계들을 가로지른 각각의 타일의 코딩 파라미터들의 어떠한 예측 및 각각의 타일의 경계들을 가로질러 각각의 타일의 엔트로피 코딩에서의 콘텍스트 선택의 어떠한 의존성은 또한 동일한 타일의 데이터에만 의존하도록 제한하기 위하여 금지될 수 있다. 따라서, 디코더는 방금 언급된 운영들을 주로 타일들의 유닛 내에서 병렬로 실행할 수 있다.

[0100] 일부 전송 채널을 거쳐 전송되도록 하기 위하여, 구문 요소들은 엔트로피 코더(28)에 의해 슬라이스-방식으로 엔트로피 코딩되어야만 한다. 이를 위하여, 엔트로피 코더(28)는 우선 제 1 타일의 블록들의 횡단을 갖는 타일들의 블록들을 스캔하고, 그리고 나서 타일 순서에서 그 다음 타일의 블록들보다 선행한다. 래스터 스캔 순서는 예를 들면, 각각 타일들 내의 블록들 및 타일들을 스캔하도록 사용될 수 있다. 슬라이스들은 그리고 나서 전송

을 위하여 가장 작은 유닛들인 네트워크 추상 계층 유닛들로 패킹된다. 슬라이스를 엔트로피 코딩하기 전에, 엔트로피 코더(28)는 그것의 콘텍스트 적응 이진 산술 코딩 확률들, 즉, 그러한 슬라이스의 구문 요소를 산술적으로 코딩하도록 사용되는 확률들을 초기화한다. 엔트로피 디코더(52)는 동일하게 수행하는데, 즉 슬라이스 시작에서 그것의 확률들을 초기화한다. 그러나, 각각의 초기화는 엔트로피 코딩 효율에 부정적으로 영향을 미치는데 그 이유는 확률들이 다양한 콘텍스트의 실제 부호 확률 통계에 연속적으로 적용되고 따라서 콘텍스트 적응 이진 산술 코딩 확률들의 재설정이 적용된 상태로부터의 편차를 나타내기 때문이다. 통상의 지식을 가진 자들에 알려진 것과 같이, 엔트로피 코딩은 확률들은 실제 부호 확률 통계들과 일치할 때만 최적 압축에 이른다.

[0101] 따라서, 본 발명의 일 실시 예에 따른 디코더는 도 18에 도시된 것과 같이 운영된다. 디코더는 단계 80에서 타일들의 트랑쉬들에서, 타일들(82) 내의 화상(12)을 설명하는 초기 바이트 시퀀스 페이로드를 수신한다. 도 18에서, 타일 순서(84) 내의 제 1 타일(82)은 바람직하게는, 각각 바람직하게는 그러한 타일 내의 블록들의 시퀀스의 서브-시퀀스를 덮는 두 개의 트랑쉬(86a 및 86b)로 절단되거나 또는 분할되도록 도시된다. 그리고 나서, 단계 82에서, 트랑쉬(86a 및 86b)들은 엔트로피 디코딩된다. 그러나, 트랑쉬들(86a 및 86b)를 엔트로피 디코딩하는데 있어서, 콘텍스트 적응 이진 산술 코딩 확률 적응이 트랑쉬 경계들을 가로질러 계속된다. 즉, 트랑쉬(86a)를 디코딩하는 동안에, 콘텍스트 적응 이진 산술 코딩 확률들이 실제 부호 통계들에 연속적으로 적용되고 트랑쉬(86a)를 엔트로피 디코딩하는 말미에서의 상태는 트랑시(86a)를 엔트로피 디코딩하기 시작할 때 적용된다. 단계 90에서, 화상(12)을 획득하기 위하여 이에 따라 엔트로피 디코딩된 초기 바이트 시퀀스 페이로드가 디코딩된다.

[0102] 타일들(82) 내부에 위치되는 트랑쉬 경계들(92)을 가로질러 연속적인 콘텍스트 적응 이진 산술 코딩 확률 적응 때문에, 이러한 트랑쉬 경계들은 타일들(82)로의 화상(12)의 분할을 넘어 엔트로피 코딩 효율에 부정적으로 영향을 미치지 않는다. 다른 한편으로는, 타일 병렬 처리가 여전히 가능하다. 그밖에, 트랑쉬들을 개별적으로 전송하는 것이 가능하며, 트랑쉬들이 완전한 타일들(82)보다 작기 때문에 각각의 타일의 제 1 트랑쉬가 수신되고 엔트로피 디코딩되자마자 단계 90에서 각각의 타일의 디코딩을 시작하는 것이 가능하다.

[0103] 도 16 내지 18의 설명은 주로 타일들의 사용에 관한 것이었다. 위에 설명된 것과 같이, 타일들은 화상의 공간적 분할을 야기한다. 타일들과 유사하게, 슬라이스들이 또한 화상을 공간적으로 분할한다. 따라서, 슬라이스들은 또한 병렬 인코딩/디코딩을 가능하게 하기 위한 수단이다. 타일들과 유사하게, 슬라이스들이 개별적으로 디코딩할 수 있도록 예측 등은 금지된다. 따라서, 도 16 내지 18의 설명은 또한 슬라이스들의 트랑쉬들로의 분할에 유효하다.

[0104] 파면 병렬 처리를 사용할 때에도 동일하게 적용된다. 파면 병렬 처리 서브스트림들은 또한 주로 파면 병렬 처리로의, 화상(12)의 공간적 분할을 나타낸다. 타일들 및 슬라이스들과 대조적으로, 파면 병렬 처리 서브스트림들은 예측들 상에 제한들을 부과하지 않으며 파면 병렬 처리 서브스트림을 가로질러 선택들을 접촉한다.

[0105] 파면 병렬 처리 서브스트림들은 도 4에 도시된 것과 같이, 최대 코딩 유닛 열들과 같은 블록 열들을 따라 확장하고, 병렬 처리를 가능하도록 하기 위하여 파면 병렬 처리 서브스트림들(도 4 참조) 중에서 정의된 것과 같은 순서로 콘텍스트 적응 이진 산술 코딩 엔트로피 코딩과 관련하여 단지 하나의 절충이 만들어지며, 첫 번째 파면 병렬 처리 서브스트림을 제외하고 각각의 파면 병렬 처리 서브스트림(92)을 위하여 콘텍스트 적응 이진 산술 코딩 확률들은 완전히 재설정되지 않고 채택되거나, 혹은 각각의 파면 병렬 처리 서브스트림을 위하여, 도 4에 도시된 것과 같이 왼쪽 면에서와 같은 화상(12)의 동일한 면에서 시작하는 최대 코딩 유닛 순서로, 그것들의 두 번째 최대 코딩 유닛까지 바로 앞의 파면 병렬 처리 서브스트림을 엔트로피 디코딩한 후에 결과로서 생기는 콘텍스트 적응 이진 산술 코딩과 동일하도록 설정된다. 따라서, 파면 병렬 처리 서브스트림들 사이의 일부 코딩 지연을 따름으로써, 이러한 파면 병렬 처리 서브스트림들(92)은 병렬로 디코딩될 수 있으며 따라서 화상(12)이 병렬로, 즉, 동시에 디코딩된 부분들은 왼쪽으로부터 오른쪽으로 경사진 방식으로 화상을 가로질러 이동하는 일종의 파면(92)을 형성한다.

[0106] 즉, 도 16 내지 18의 설명을 파면 병렬 처리 서브스트림들에 전달하는데 있어서, 어떠한 파면 병렬 처리 서브스트림(92, 도 4)도 또한 각각의 파면 병렬 처리 서브스트림(92)의 내부에 이러한 트랑쉬들(98a 및 98b) 사이의 경계(100)에서 콘텍스트 적응 이진 산술 코딩 확률 적응을 방해하지 않고 트랑쉬들(98a 및 98b)로 분할될 수 있으며, 그렇게 함으로써 두 트랑쉬(98a 및 98b)의 개별 전송가능성에 기인하는 엔트로피 코딩 효율과 관련하여 불이익을 방지하고 파면 병렬 처리를 사용하는 능력을 유지하며 트랑쉬들이 완전한 파면 병렬 처리 서브스트림들(92)보다 작기 때문에 이러한 파면 병렬 처리를 일찍 시작하는 것을 가능하게 한다.

[0107] 도 1 내지 15와 관련하여 위에 설명된 것과 같이, 네트워크 추상 계층 유닛들로 패킷화된 트랑쉬들을 전송하기

위한 일부 가능성이 존재한다. 그러한 트랑쉬들 또는 서브스트림들의 타일들 또는 서브스트림들 또는 슬라이스들이 각각의 서브스트림 또는 타일의 n번 째 트랑쉬를 선행하고 트랑쉬 경계들에 국한하도록 허용하는 정보를 나타내는 헤더를 갖는 산술적으로 코딩된 도메인 내의 트랑쉬들로 분할된 도 3이 참조된다. 또 다른 실시 예는 도 9에 나타내었다. 도 9에서, 타일들 또는 파면 병렬 처리 서브스트림들의 트랑쉬들로의 분할은 타일 또는 파면 병렬 처리 서브스트림 경계에서 시작하는, 즉, 타일 또는 파면 병렬 처리 서브스트림의 시작에서 시작하는 슬라이스 구조를 약간 변경함으로써 수행되었으며, 그렇게 함으로써 일반적인 콘텍스트 적응 이진 산술 코딩 확률 초기화/재설정을 야기한다. 그러나, 타일 또는 파면 병렬 처리 서브스트림의 내부에서 시작하는 트랑쉬들을 지니는 슬라이스들은 1로 설정되는 no_cabac_reset_flag를 가지며, 그렇게 함으로써 앞서 설명된 콘텍스트 적응 이진 산술 코딩 확률 적응의 지속을 야기한다.

[0108] 수신 단계(80)에서 발생하는, 탈-인터리빙이 관련되는 한, 각각의 트랑쉬를 위하여 어떠한 파면 병렬 처리 서브스트림 또는 타일에 각각의 트랑쉬가 속하는지가 결정된다. 예를 들면, 현재 화상의 파면 병렬 처리 서브스트림 또는 타일의 수를 통한 라운드-로빈(round-robin) 순환과 같이 서로 다른 가능성들이 위에 설명되었다. 대안으로서, 트랑쉬들을 전송하기 위하여 슬라이스 헤더들을 사용하는 경우에, 슬라이스 헤더들은 현재 화상(12) 내의 각각의 슬라이스의 시작을 국한하도록 허용하는 표시를 포함할 수 있다.

[0109] 이와 관련하여, 슬라이스들, 파면 병렬 처리 서브스트림들 또는 타일들의 트랑쉬들로의 분해는 각각의 슬라이스, 파면 병렬 처리 서브스트림 또는 타일 내의 정의되는 디코딩 순서를 따라 실행되는 것으로 알려져 있는데; 즉, 각각의 슬라이스, 파면 병렬 처리 서브스트림 또는 타일 내에, 각각의 슬라이스, 파면 병렬 처리 서브스트림 또는 타일에 의해 공간적으로 덮혀진 화상의 부분은 각각의 슬라이스, 파면 병렬 처리 서브스트림 또는 타일 내로 코딩되거나, 또는 그러한 디코딩 순서로 이들로부터 디코딩되며, 각각의 슬라이스, 파면 병렬 처리 서브스트림 또는 타일의 각각의 트랑쉬는 디코딩 순서를 따라 각각의 슬라이스, 파면 병렬 처리 서브스트림 또는 타일의 연속적인 부분을 덮는다. 이러한 방식에 의해, 동일한 슬라이스, 파면 병렬 처리 서브스트림 또는 타일에 속하는 트랑쉬들 중에서, 순서, 주로 코딩/디코딩의 순서가 정의되고, 각각의 트랑쉬는 그러한 순서 내에 순위(rank)를 갖는다. 화상의 파면 병렬 처리 서브스트림들 또는 타일들로의 분할이 디코더로 신호전달되기 때문에, 디코더는 분할에 대하여 알고 있다. 따라서, 각각의 트랑쉬를 예를 들면, 각각의 파면 병렬 처리 서브스트림 또는 타일과 연관시키기 위하여, 만일 각각의 트랑쉬가 각각의 트랑쉬가, 각각의 트랑쉬가 이들의 일부분인 타일/파면 병렬 처리 서브스트림의 코딩/디코딩 순서를 사용하여 연속적으로 화상을 덮는 위치 상에 시작 위치를 식별하는 시작 주소를 가지면 이는 충분할 수 있다. 예를 들면, 특정 타일 또는 파면 병렬 처리 서브스트림에 속하는 트랑쉬들 중에서의 순서도 전송 디멀티플렉서에서, 또는 시작 위치를 사용하는 디코더에 의해 재구성될 수 있다. 그러나, 재분류(resorting)를 위하여, 실시간 전송 프로토콜 전송과 관련하여 위에 설명된 것과 같이 하부 개방형 시스템 간 상호 접속(Open Systems Interconnection, OSI) 계층들의 전송 패킷 헤더들의 정보가 또한 디코딩 순서 번호, 즉 디코딩 순서 번호와 같이 사용될 수 있다. 방금 언급된 형태의 전송 디멀티플렉서는 동일한 파면 병렬 처리 서브스트림 또는 타일의 트랑쉬들의 데이터를 하나의 슬라이스 버퍼 내에 저장하고 서로 다른 파면 병렬 처리 서브스트림들 또는 타일들과 관련된 트랑쉬들의 데이터를 서로 다른 슬라이스 버퍼들 내에 저장하기 위하여 위에 논의된 전송 디멀티플렉서와 유사하게 구성될 수 있다. 위에 언급된 것과 같이, 슬라이스 구조, 즉, 슬라이스 헤더는 트랑쉬들을 전달하도록 사용될 수 있다.

[0110] 그 다음에, 다시 말해서 그것들을 다시 설명하기 위하여 도 11 내지 15의 실시 예들이 참조된다. 이러한 도면들에서 설명된 것과 같이, 슬라이스들(Si)은 네트워크 추상 계층 유닛 헤더(112)를 포함하는 각각의 네트워크 추상 계층 유닛(110, 도 11 참조)을 갖는 네트워크 추상 계층 유닛들로 패킷화된다. 슬라이스들(Si)은 정상 슬라이스들 또는 도 9에 따른 트랑쉬들을 지니는 슬라이스들일 수 있다는 것을 이해하여야 한다. 따라서, 이러한 슬라이스들은 현재 화상의 하나의 파면 병렬 처리 서브스트림 또는 타일과 관련된 데이터만을, 주로 각각, i번 째 파면 병렬 처리 서브스트림 또는 타일만을 지닌다. 단편화를 거쳐, 네트워크 추상 계층 유닛들(110)은 전송 스트림 패킷들(114), 주로 그것들의 페이로드 섹션(116)을 거쳐 전송된다. 그렇게 함으로써, 각각의 네트워크 추상 계층 유닛(110)과 상응하는 슬라이스(Si)는 즉, 네트워크 추상 계층 유닛(110) 바로 다음의 슬라이스 바로 다음의 파면 병렬 처리 서브스트림 또는 타일이 속하는, i를 나타내는 각각의 서브스트림 마커(MKR)에 의해 선행된다.

[0111] 서로 다른 파면 병렬 처리 서브스트림들 또는 타일들에 속하는 슬라이스들을 지니는 네트워크 추상 계층 유닛들(110)은 도 11 내지 13에 설명된 것과 같이 하나 이상의 기본 스트림(ES) 또는 동일한 기본 스트림 상에 분포될 수 있다. 위에 언급된 것과 같이, "기본 스트림"은 또한 고유의 실시간 전송 프로토콜 세션 내의 개별 실시간 전송 프로토콜 스트림을 식별할 수 있다.

[0112] 도 14 및 15와 관련하여 설명된 것과 같이, 전송 디멀티플렉서는 멀티플렉스 버퍼(MB), 슬라이스 버퍼(SB) 및 전송 버퍼(TB)를 포함할 수 있다. 슬라이스 버퍼들(SB)들은 WPP 서브스트림들 또는 타일들 내의 화상의 병렬 디코딩을 허용하는 멀티-스레드 디코더(MTD)에 의해 당겨진다. 전송 버퍼(TB)는 비디오 비트 스트림의 미리 결정된 기본 스트림의 전송 스트림 패킷에 속하는 데이터를 수집하고 데이터를 멀티플렉스 버퍼(MB)에 전달하도록 구성된다. 전송 디멀티플럭서는 그리고 나서 멀티플렉스 버퍼(MB)의 출력에서 전송 스트림 패킷들로 패킷화된 네트워크 추상 계층 유닛 시퀀스의 네트워크 추상 계층 유닛들의 네트워크 추상 계층 유닛 헤더들을 평가하고, 서브스트림 마커 네트워크 추상 계층 유닛들 내에 운반된 서브스트림 마커 데이터의 저장과 함께 서브스트림 마커 네트워크 추상 계층 유닛들(MKR)을 드로핑하며, 이들의 데이터 필드가 하나의, 즉, 동일한 슬라이스 버퍼(SB) 내의 동일한 파면 병렬 처리 서브스트림 또는 타일을 식별하는 서브스트림 마커 네트워크 추상 계층 유닛들을 후속하게 하는 네트워크 추상 계층 유닛들 내의 서브스트림들 또는 타일들의 슬라이스들의 데이터, 및 이들의 데이터 필드가 서로 다른 슬라이스 버퍼들 내의 서로 다른 파면 병렬 처리 서브스트림들 또는 타일들을 식별하는 서브스트림 마커 네트워크 추상 계층 유닛들을 후속하게 하는 네트워크 추상 계층 유닛들 내의 파면 병렬 처리 서브스트림들 또는 타일들의 슬라이스들의 데이터를 저장한다. 도 15에 도시된 것과 같이, 전송 디멀티플렉서는 도 15에서 TS demux라 불리는 디멀티플렉서를 포함하며, 비디오 비트 스트림을 수신하고 비디오 비트 스트림의 전송 스트림 패킷들을 서로 다른 기본 스트림들로 분할하도록, 즉, 비디오 비트 스트림의 전송 스트림 패킷을 서로 다른 기본 스트림들에 분배하도록 구성된다. 디멀티플렉서는 각각의 기본 스트림이 다른 기본 스트림들의 전송 스트림 패킷들의 프로그램 관련 데이터(Program Associated Data, PAD)들과 다른 프로그램 관련 데이터의 전송 스트림 패킷들로 구성되도록 전송 스트림 패킷의 전송 스트림 헤더 내에 포함된 패킷 식별자들에 따라 이러한 분할 또는 분배를 실행한다.

[0113] 즉, 만일 도 9의 실시 예의 의미에서 슬라이스들이 트랑쉬들과 상응하면, 각각의 WPP 서브스트림 또는 타일의 상응하는 슬라이스 버퍼(SB)가 그 안에 포함된 데이터를 갖자마자 멀티-스레드 디코더는 현재 화상의 하나 이상의 WPP 서브스트림 또는 타일을 처리하기 시작할 수 있으며, 그렇게 함으로써 지연을 감소시킨다.

[0114] 장치의 맥락에서 일부 양상들이 설명되었으나, 이러한 양상들은 또한 블록 또는 장치가 방법 단계 또는 방법 단계의 특징과 상응하는, 상응하는 방법의 설명을 나타내는 것은 자명하다. 바람직하게는, 방법 단계의 맥락에서 설명된 양상들은 또한 상응하는 장치의 상응하는 블록 또는 아이템 또는 특징의 설명을 나타낸다. 방법 단계들의 일부 또는 모두는 예를 들면, 마이크로프로세서, 프로그램가능 컴퓨터 또는 전자 회로 같은 하드웨어 장치에 의해(또는 장치를 사용하여) 실행될 수 있다. 일부 실시 예들에서, 강장 중요한 방법 단계들 중 일부 또는 그 이상이 그러한 장치에 의해 실행될 수 있다.

[0115] 본 발명의 인코딩된 비트스트림은 디지털 저장 매체 상에 저장될 수 있거나 또는 무선 전송 매체 또는 인터넷과 같은 유선 전송 매체와 같은 전송 매체 상에 전송될 수 있다.

[0116] 따라서 위의 내용들은 그중에서도, 타일들, 파면 병력 처리 서브스트림들, 슬라이스들 또는 엔트로피 슬라이스들 내에 구조화된 것과 같이, 새로운 고효율 비디오 코딩에 의해 제공되는 것과 같은 구조화 비디오 데이터의 저 지연 캡슐화 및 전송을 위한 방법들을 설명한다. 그중에서도 엔트로피 슬라이스들/슬라이스들/타일들/서브스트림들의 인터리빙된 전송을 통한 병렬화된 인코더-트랜스미터-수신기-디코더 환경 내의 저 지연 전송을 허용하는 기술이 제시되었다. 본 명세서의 도입부에 나타낸 애로사항을 해결하고 전송과 디코딩 시간의 지연, 즉 종단간 지연을 최소화하기 위하여, 그중에서도 병렬 전송과 처리를 위하여 인터리빙된 엔트로피 슬라이스 전략을 위한 기술이 제시되었다.

[0117] 특정 구현 필요 조건에 따라, 본 발명의 실시 예들은 하드웨어 또는 소프트웨어에서 구현될 수 있다. 구현은 디지털 저장 매체, 예를 들면, 그 안에 저장되는 전자적으로 판독가능한 제어 신호들을 갖는, 플로피 디스크, DVD, 블루-레이, CD, RON, PROM, EPROM, EEPROM 또는 플래시 메모리를 사용하여 실행될 수 있으며, 이는 각각의 방법이 실행되는 것과 같이 프로그램가능 컴퓨터 시스템과 협력한다(또는 협력할 수 있다). 따라서, 디지털 저장 매체는 컴퓨터로 판독가능할 수 있다.

[0118] 본 발명에 따른 일부 실시 예들은 여기에 설명된 방법들 중 어느 하나가 실행되는 것과 같이, 프로그램가능 컴퓨터 시스템과 협력할 수 있는, 전자적으로 판독가능한 제어 신호들을 갖는 데이터 캐리어를 포함한다.

[0119] 일반적으로, 본 발명의 실시 예들은 프로그램 코드를 갖는 컴퓨터 프로그램 제품으로서 구현될 수 있으며, 프로그램 코드는 컴퓨터 프로그램 제품이 컴퓨터 상에서 구동할 때 방법들 중 어느 하나를 실행하도록 운영될 수 있다. 프로그램 코드는 예를 들면, 기계 판독가능 캐리어 상에 저장될 수 있다.

[0120] 다른 실시 예들은 기계 판독가능 캐리어 상에 저장되는, 여기에 설명된 방법들 중 어느 하나를 실행하기 위한 컴퓨터 프로그램을 포함한다.

[0121] 바꾸어 말하면, 본 발명의 일 실시 예는 따라서 컴퓨터 프로그램이 컴퓨터 상에 구동할 때, 여기에 설명된 방법들 중 어느 하나를 실행하기 위한 프로그램 코드를 갖는 컴퓨터 프로그램이다.

[0122] 본 발명의 또 다른 실시 예는 따라서 여기에 설명된 방법들 중 어느 하나를 실행하기 위한 컴퓨터 프로그램을 포함하는, 그 안에 기록되는 데이터 캐리어(또는 데이터 저장 매체,또는 컴퓨터 판독가능 매체)이다. 데이터 캐리어, 디지털 저장 매체 또는 기록 매체는 일반적으로 유형(tangible) 및/또는 비-전이형이다.

[0123] 본 발명의 또 다른 실시 예는 따라서 여기에 설명된 방법들 중 어느 하나를 실행하기 위한 컴퓨터 프로그램을 나타내는 데이터 스트림 또는 신호들의 시퀀스이다. 데이터 스트림 또는 신호들의 시퀀스는 예를 들면 데이터 통신 연결, 예를 들면 인터넷을 거쳐 전송되도록 구성될 수 있다.

[0124] 또 다른 실시 예는 여기에 설명된 방법들 중 어느 하나를 실행하도록 구성되거나 또는 적용되는, 처리 수단, 예를 들면 컴퓨터, 또는 프로그램가능 논리 장치를 포함한다.

[0125] 또 다른 실시 예는 그 안에 여기에 설명된 방법들 중 어느 하나를 실행하기 위한 컴퓨터 프로그램이 설치된 컴퓨터를 포함한다.

[0126] 본 발명에 따른 또 다른 실시 예는 여기에 설명된 방법들 중 어느 하나를 실행하기 위한 컴퓨터 프로그램을 수신기로 전송하도록(예를 들면, 전자적으로 또는 선택적으로) 구성되는 장치 또는 시스템을 포함한다. 수신기는 예를 들면, 컴퓨터, 이동 장치, 메모리 장치 등일 수 있다. 장치 또는 시스템은 예를 들면, 컴퓨터 프로그램을 수신기로 전송하기 위한 파일 서버를 포함한다.

[0127] 일부 실시 예들에서, 여기에 설명된 방법들 중 일부 또는 모두를 실행하기 위하여 프로그램가능 논리 장치(예를 들면, 필드 프로그램가능 게이트 어레이)가 사용될 수 있다. 일부 실시 예들에서, 필드 프로그램가능 게이트 어레이는 여기에 설명된 방법들 중 어느 하나를 실행하기 위하여 마이크로프로세서와 협력할 수 있다. 일반적으로, 방법들은 바람직하게는 어떠한 하드웨어 장치에 의해 실행된다.

[0128] 이에 설명된 실시 예들은 단지 본 발명의 원리들을 위한 설명이다. 여기에 설명된 배치들과 상세내용들의 변형과 변경은 통상의 지식을 가진 자들에 자명할 것이라는 것을 이해할 것이다. 따라서, 본 발명은 여기에 설명된 실시 예들의 설명에 의해 표현된 특정 상세내용이 아닌 특허 청구항의 범위에 의해서만 한정되는 것으로 의도된다.

[0130] 참고문헌

[0131] [1] Thomas Wiegand, Gary J. Sullivan, Gisle Bjontegaard, Ajay Luthra, "Overview of the H.264/AVC Video Coding Standard", IEEE Trans. Circuits Syst. Video Technol., vol. 13, N7, July 2003.

[0132] [2] JCTVC-E196, "Wavefront Parallel Processing", 5th JCT-VC Meeting, Geneva 2011.

[0133] [3] JCTVC-D070, "Lightweight slicing for entropy coding", 4th Meeting, Daegu, 2011.

[0134] [4] JCTVC-D073, "Periodic initialization for wavefront coding functionality", 4th Meeting, Daegu, 2011.

[0135] [5] HEVC WD5: Working Draft 5 of High-Efficiency Video Coding JTCVC-G1103, 5th JCT-VC Meeting, Geneva Meeting November 2011.

[0136] [6] JTCVC-D243, "Analysis of entropy slices approaches", 4th Meeting, Daegu, 2011.

[0137] [7] ISO/IEC 13818-1/2011, MPEG-2 Transport Stream including AMDs 1-6.

[0138] [8] IETF Real-time transport protocol, RTP RFC 3550.

[0139] [9] IETF RTP Payload Format, IETF RFC 6184.

[0140] [10] JCTVC-F275, Wavefront and Cabac Flush: Different Degrees of Parallelism Without Transcoding, Torino Meeting

[0141] [11] JCT-VC-F724, Wavefront Parallel Processing for HEVC Encoding and Decoding, Torino Meeting** at

end of description

[0142] [12] IETF Session Description Protocol (SDP), RFC 4566

[0143] [13] IETF RTP Payload Format for High Efficiency Video Coding, draft-schierl-payload-h265

부호의 설명

[0144] 10 : 인코더

12 : 화상

14 : 비디오

18 : 예측기

20 : 예측 잔여

22 : 잔여 결정기

24 : 변환/탈양자화 모듈

26 : 양자화된 잔여

28 : 엔트로피 코더

30 : 잔여 신호

31 : 변환 및 탈양자화 모듈

32 : 예측 신호

33 : 결합기

34 : 재구성된 신호

36 : 필터

38 : 참조 화상

40 : 초기 바이트 시퀀스 페이로드

50 : 디코더

52 : 엔트로피 디코더

54 : 재변환/탈양자화 모듈

56 : 결합기

58 : 필터

60 : 예측기

62 : 잔여 신호

64 : 코딩 파라미터

66 : 예측 신호

68 : 재구성된 신호

82 : 타일

86a, 86b : 트랑쉬

92 : 트랑쉬 경계

92 : 파면 병렬 처리 서브스트림

98a, 98b : 트랑쉬

100 : 경계

110 : 네트워크 추상 계층 유닛

112 : 네트워크 추상 계층 유닛 헤더

114 : 전송 스트림 패킷

116 : 페이로드 섹션

300 : 패킷

302 : 헤더

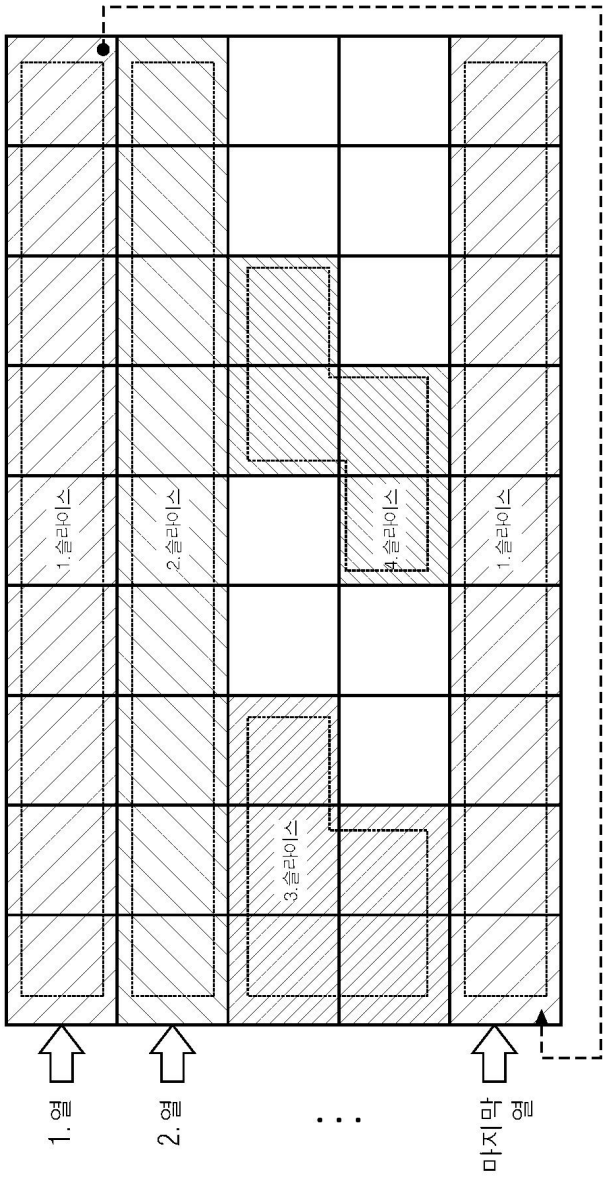
304 : 마커

306 : 저 지연 특징 표시기

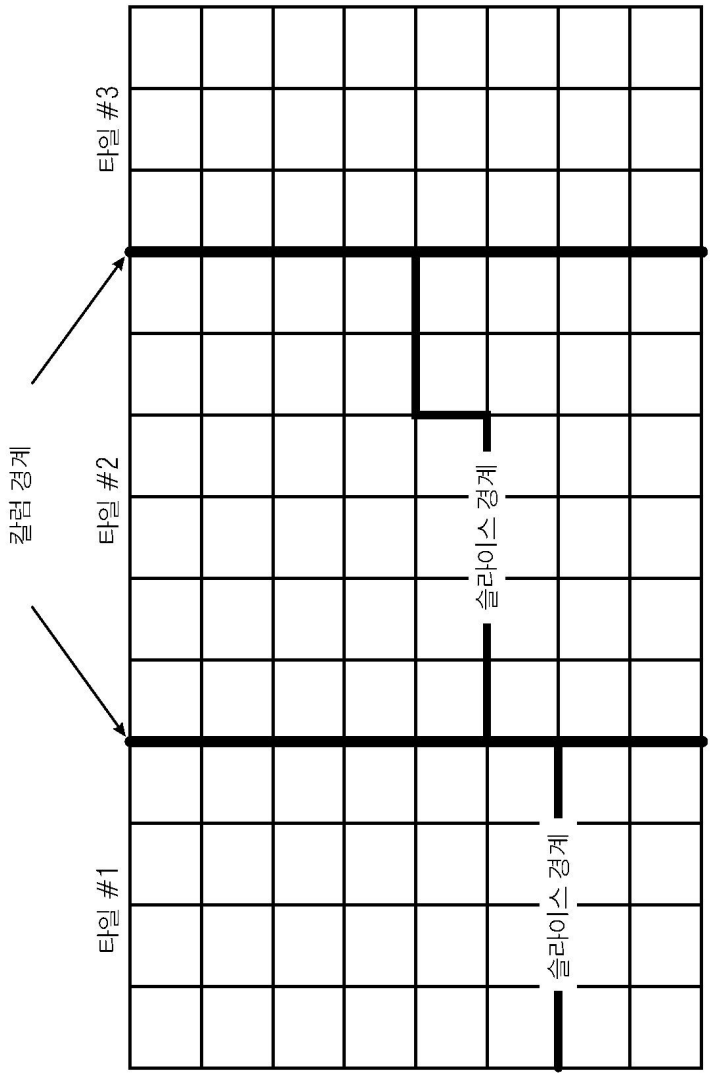
308 : 연속 표시기

도면

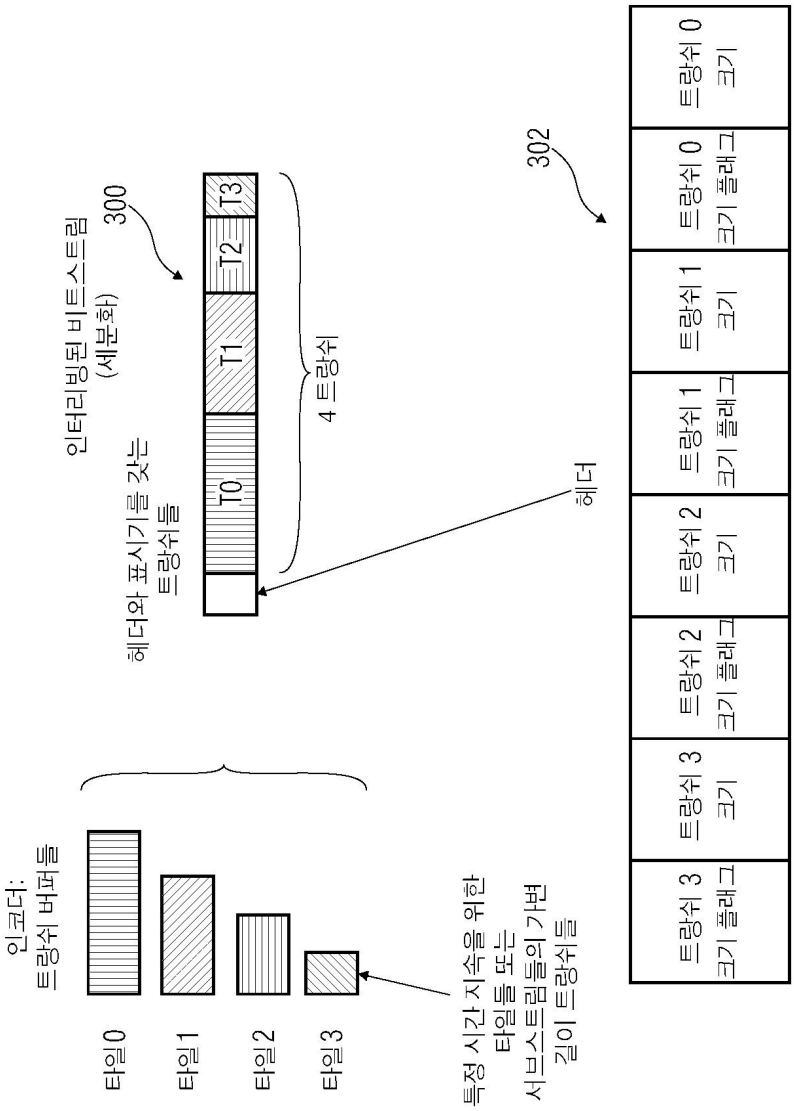
도면1



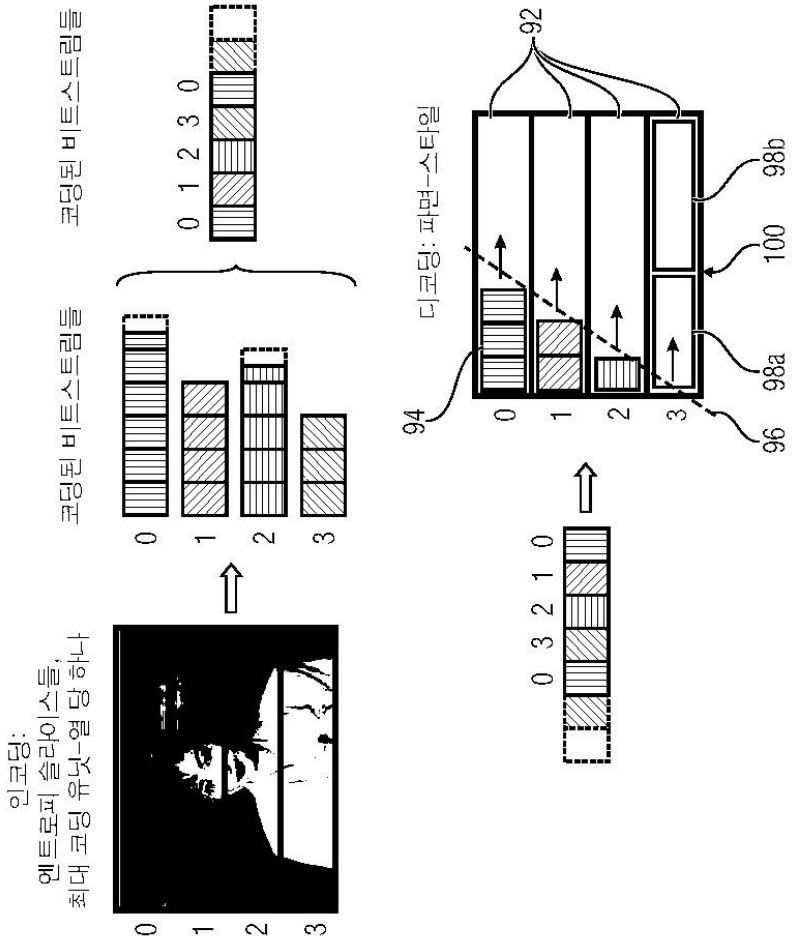
도면2



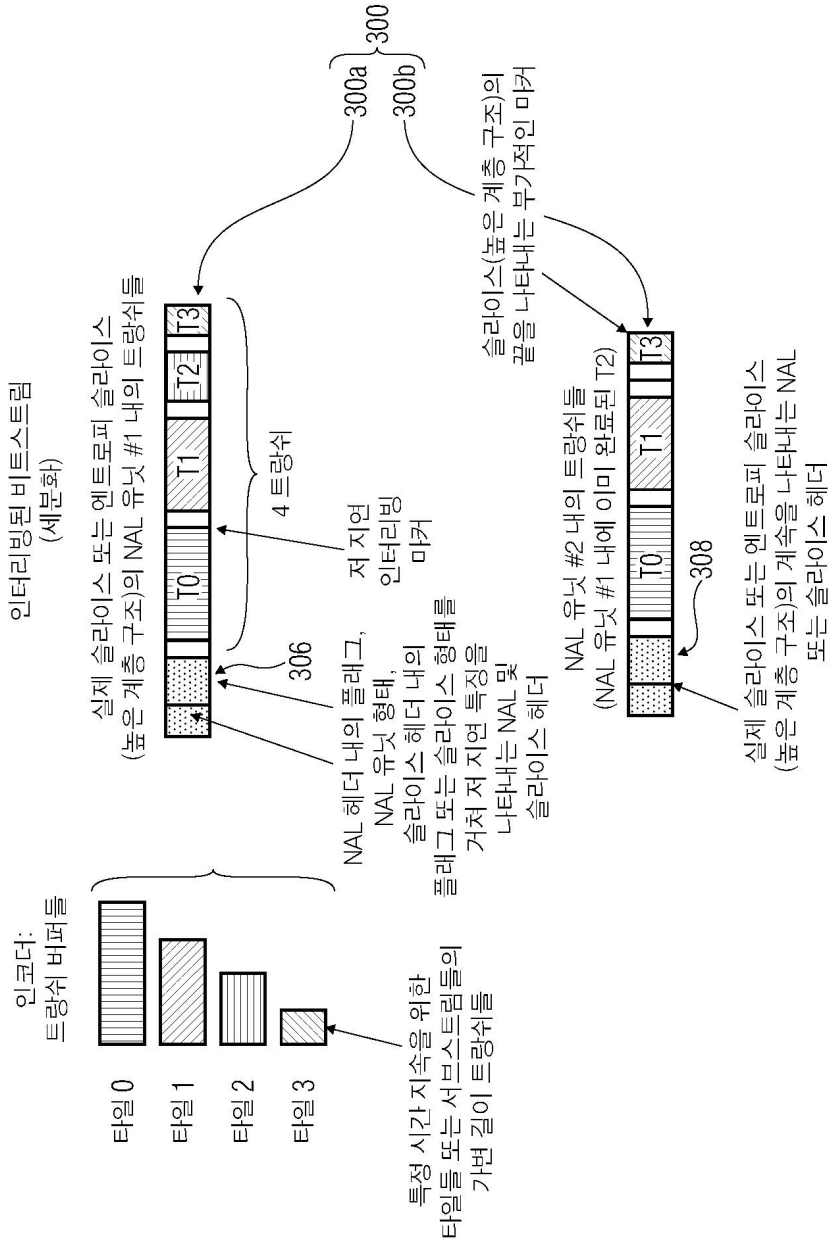
도면3


인코더:
트랑쉬 버퍼들
타일0
타일1
타일2
타일3
특정 시간 지속을 위한
타일들 또는
서브스트림들의 가변
길이 트랑쉬들
인터리빙된 비트스트림
(세분화)
300
헤더와 표시기를 갖는
트랑쉬들
T0
T1
T2
T3
4 트랑쉬
헤더
302
트랑쉬 3
크기 플래그
트랑쉬 3
크기
트랑쉬 2
크기 플래그
트랑쉬 2
크기
트랑쉬 1
크기 플래그
트랑쉬 1
크기
트랑쉬 0
크기 플래그
트랑쉬 0
크기

도면4


인코딩:
엔트로피 슬라이스들,
최대 코딩 유닛 열 당 하나
0 1 2 3
코딩된 비트스트림들
0 1 2 3
코딩된 비트스트림들
0 1 2 3 0
디코딩: 파면-스타일
0 3 2 1 0
92
94
96
98a
98b
100

도면5


인터리빙된 비트스트림
(세분화)
실제 슬라이스 또는 엔트로피 슬라이스
(높은 계층 구조)의 NAL 유닛 #1 내의 트랑쉬들
300a
300b
300
T0
T1
T2
T3
4 트랑쉬
306
저 지연
인터리빙
마커
NAL 헤더 내의 플래그,
NAL 유닛 형태,
슬라이스 헤더 내의
플래그 또는 슬라이스 형태를
거쳐 저 지연 특징을
나타내는 NAL 및
슬라이스 헤더
슬라이스(높은 계층 구조)의
끝을 나타내는 부가적인 마커
NAL 유닛 #2 내의 트랑쉬들
(NAL 유닛 #1 내에 이미 완료된 T2)
308
실제 슬라이스 또는 엔트로피 슬라이스
(높은 계층 구조)의 계속을 나타내는 NAL
또는 슬라이스 헤더
인코더:
트랑쉬 버퍼들
타일 0
타일 1
타일 2
타일 3
특정 시간 지속을 위한
타일들 또는 서브스트림들의
가변 길이 트랑쉬들

도면6a

nal unit(NumBytesInNALunit, num low delay tranches) {	기술자(Descriptor)
forbidden_zero_bit	f(1)
nal_ref_flag	u(1)
fnal_unit_type	u(6)
NumBytesInRBSP=0	
nalUnitHeaderBytes=1	
if(nal_unit_type==1||nal_unit_type== 4||nal_unit_type== 5) {	
temporal_id	u(3)
output_flag	u(1)
low_delay_encapsulation_flag	u(1)
low_delay_cyclic_flag	u(1)
reserved_one_2bits	u(2)
nalUnitHeaderBytes+=1	
}	
if(low_delay_encapsulation_flag !=1)	
{	
for(i=nalUnitHeaderBytes;i<NumBytesInNALunit; i++) {	
if(i+2<NumBytesInNALunit && next bits(24) == 0x000003) {	
rbsp_byte[NumBytesInRBSP++ **]**	b(8)
rbsp_byte[NumBytesInRBSP++ **]**	b(8)
i+=2	
emulation_prevention_three_byte /* equal to 0x03 */	f(8)
} else	
rbsp_byte[NumBytesInRBSP++ **]**	b(8)
}	
}	
else{	
for (i=0, i++, i<num_low_delay_tranches){	
NumLDBytesInRBSP[i]=0	
}	

도면6b

```
tranch_id=-1                                                          
for ( i=0, i++, i<num_low_delay_tranches){                            
  NumBytesInRBSP[i]=0                                                 
}                                                                     
for( i=nalUnitHeaderBytes;i<NumBytesInNALunit;i++ ) {                 
  if( i+2<NumBytesInNALunit && next_bits( 24 ) == 0x000002 ) {        
    if( low_delay_cyclic_flag) {                                      
      tranche_id=(tranche_id++) % num_low_delay_tranches              
      i+=2                                                            
    }                                                                 
    else{                                                             
      tranche_id                                              u(8)    
      i+=3                                                            
    }                                                                 
  }                                                                   
  else                                                                
  if( i+2<NumBytesInNALunit && next_bits( 24 ) == 0x000003 ) {        
    LD_rbsp_byte[ NumLDBytesInRBSP[tranche_id]++ ] [tranche_id]  b(8)  
    LD_rbsp_byte[ NumLDBytesInRBSP[tranche_id]++ ] [tranche_id]  b(8)  
    i+=2                                                              
    emultation_prevention_three_byte /* equal to 0x03 */        f(8)  
  } else                                                              
    LD_rbsp_byte[ NumLDBytesInRBSP[tranche_id]++ ] [tranche_id]  b(8)  
}                                                                     
for ( i=0, i++, i<num_low_delay_tranches){                            
  for ( i=0, i++, i< NumLDBytesInRBSP[i] ){                           
    rbsp_byte[ NumBytesInRBSP++]=LD_rbsp_byte[i][i]                   
  }                                                                   
}                                                                     
}                                                                     
}                                                                     
```

도면7

```
•••                                                                   
low_delay_encapsulaiton_present_flag                          u(1)    
if( low_delay_encapsulaiton_present_flag ==1){                        
  low_delay_cyclic_flag                                       u(1)    
  num_low_delay_tranches_flag                                 u(1)    
  if (num_low_delay_tranches_flag==1){                                
    num_low_delay_tranches                                    ue(v)   
      for ( i=0; i <num_low_delay_tranches; i++){                     
        low_delay_tranche_ lenght_minus1 [i]                  ue(v)   
      }                                                               
  }                                                                   
}                                                                     
•••                                                                   
```

도면8

low_delay_slice_layer_rbsp( ) {
slice_header( )
tranche_id=0
for (i=0, i++, i<num_low_delay_tranches){
tranche_slice_data( )
}
rbsp_slice_trailing_bits( )
	기술자
	(Descriptor)

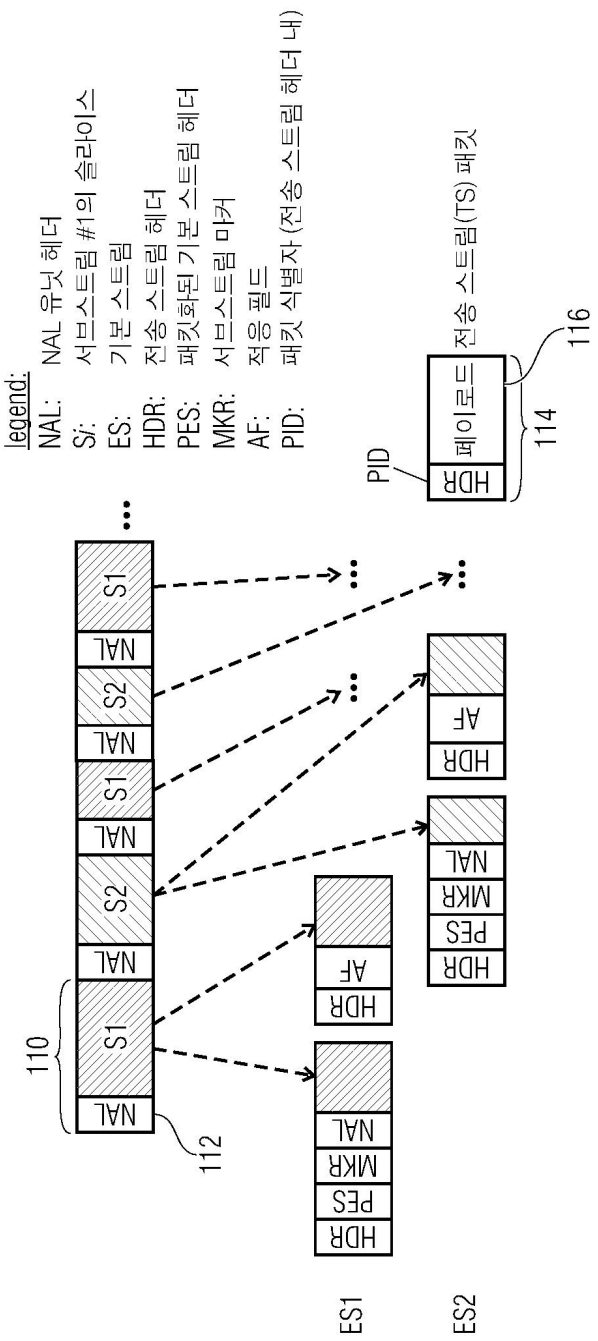
도면9

slice_header( ){	기술자 (Descriptor)
entropy_slice_flag	u(1)
low_delay_slice_flag	u(1)
....	
first_slice_in_pic_flag	u(1)
if(first_slice_in_pic_flag==0){	
slice_adress	u(v)
if(low_delay_slice_flag==1)	
no_cabac_reset_flag	u(1)
}	
....	
if(slice_type==P || slice_type==B)	
5_minus_max_num_merge_cand	ue(v)
for(i=0; i<num_substreams_minus1+1; i++){	
substream_length_mode	u(2)
substream_length[i]	u(v)
}	
}	

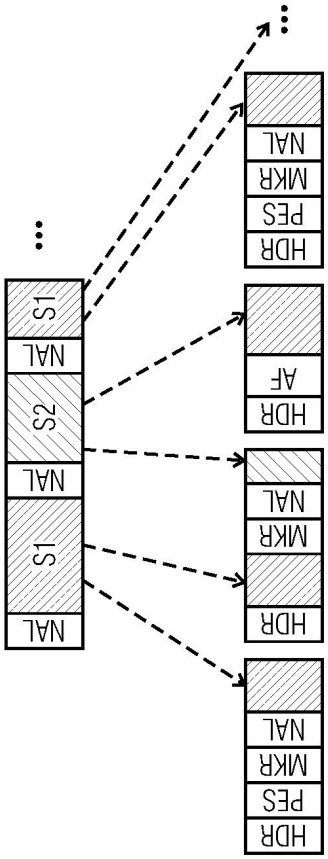
도면10

substream_marker(gradual_decoder_refresh_steps){	기술자 (Descriptor)
forbidden_zero_bit	f(1)
nal_ref_flag	u(1)
nal_unit_type	u(6)
substream_ID	u(8)
if(gradual_decoder_refresh_steps< 1)	
is_intra	u(1)
reserved_one_7bits	u(7)
}	
}	

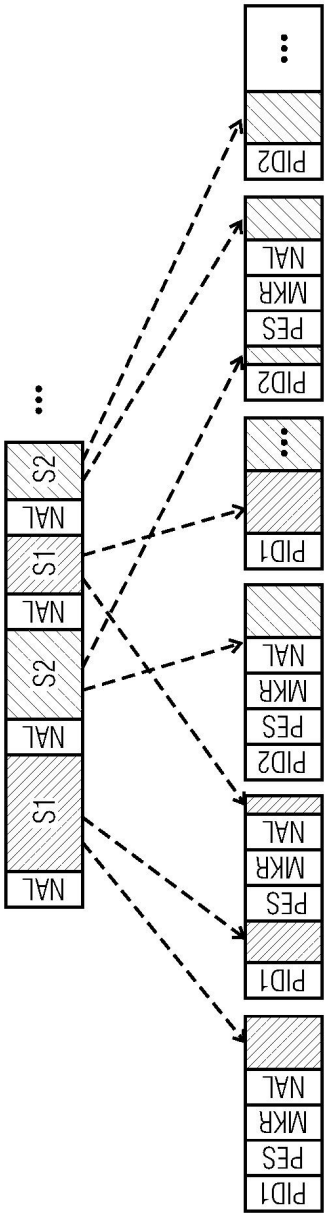
도면11


legend:
NAL: NAL 유닛 헤더
Si: 서브스트림 #i의 슬라이스
ES: 기본 스트림
HDR: 전송 스트림 헤더
PES: 패킷화된 기본 스트림 헤더
MKR: 서브스트림 마커
AF: 적응 필드
PID: 패킷 식별자 (전송 스트림 헤더 내)
110
112
114
116
PID
ES1
ES2
HDR
AF
PES
MKR
NAL
S1
S2
페이로드
전송 스트림(TS) 패킷

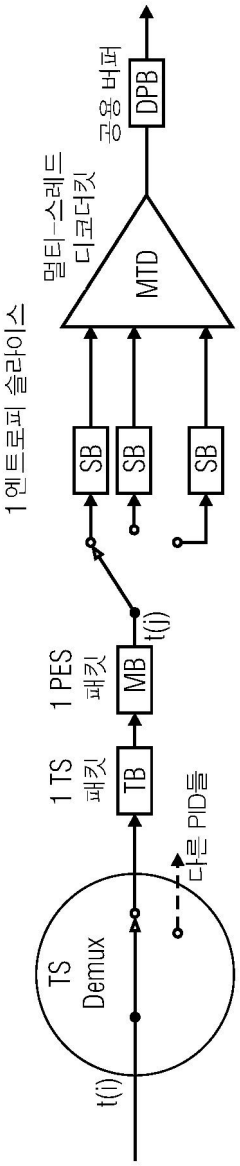
도면12


S1
NAL
S2
NAL
S1
NAL
HDR
PES
MKR
NAL
HDR
AF
HDR
NAL
MKR
HDR
HDR
PES
MKR
NAL

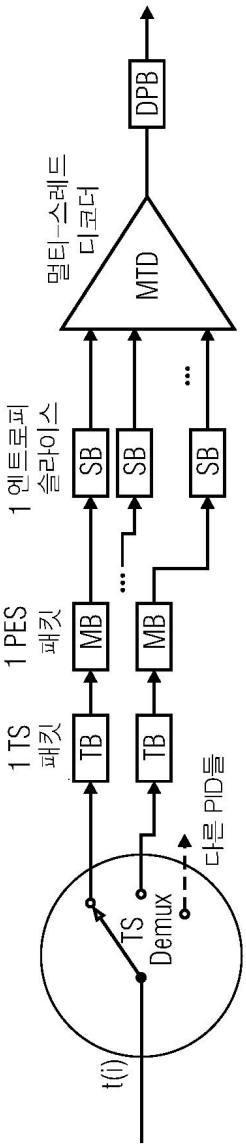
도면13


S2
NAL
S1
NAL
S2
NAL
S1
NAL
PID2
NAL
MKR
PES
PID2
PID1
NAL
MKR
PES
PID2
NAL
MKR
PES
PID1
NAL
MKR
PES
PID1

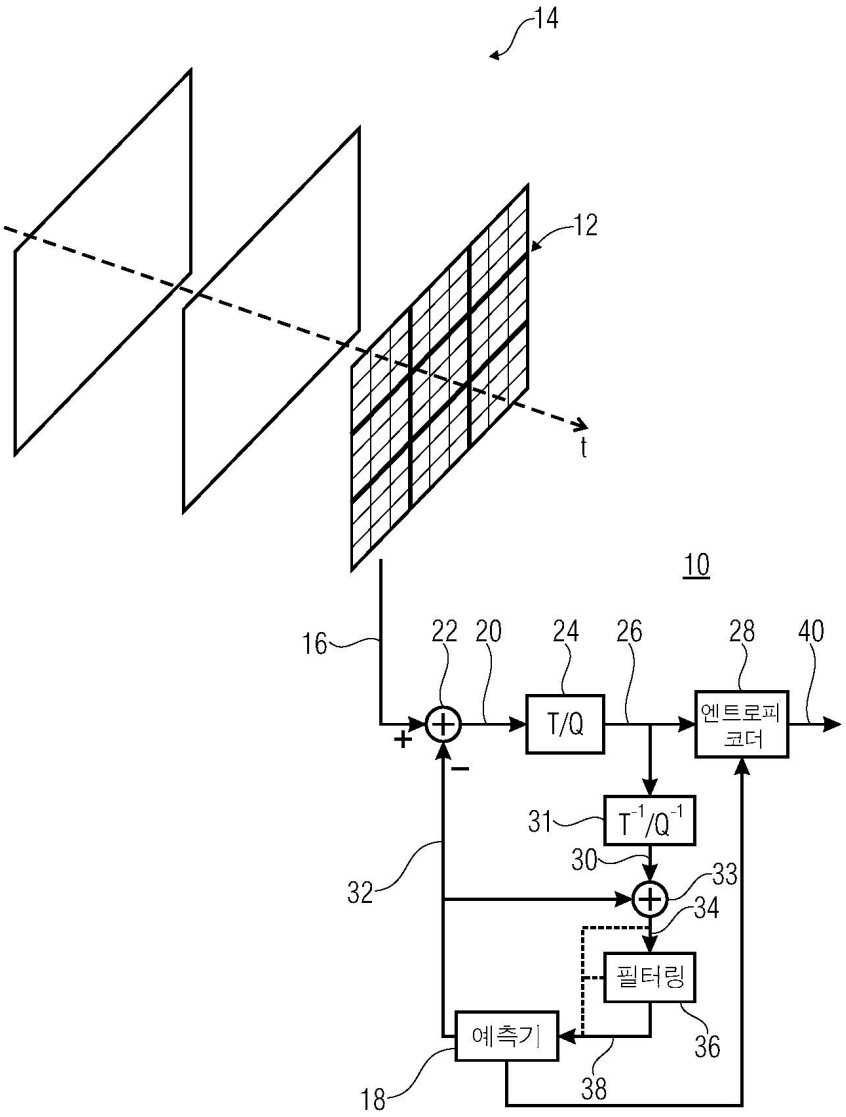
도면14


1 엔트로피 슬라이스
멀티-스레드 디코더킷
MTD
공용 버퍼
DPB
SB
SB
SB
1 PES 패킷
MB
t(i)
1 TS 패킷
TB
다른 PID들
TS Demux
t(i)

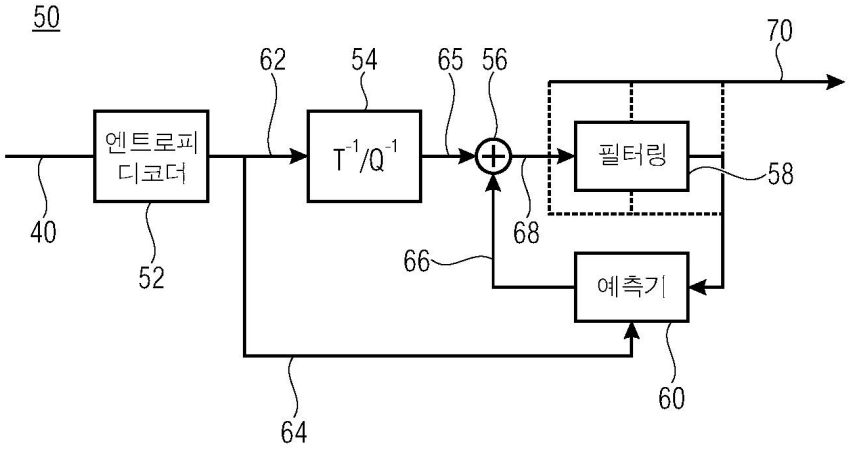
도면15


DPB
멀티-스레드 디코더
MTD
1 엔트로피 슬라이스
SB
SB
SB
1 PES 패킷
MB
MB
1 TS 패킷
TB
TB
다른 PID들
TS Demux
t(i)

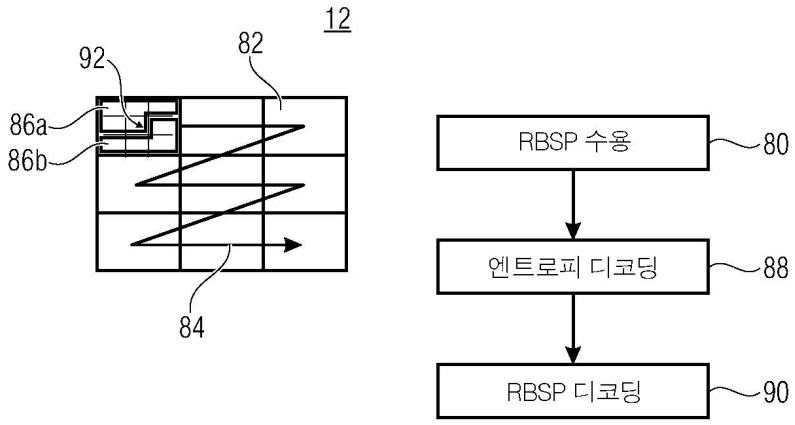
도면16


14
12
t
10
16
22
20
24
26
28
40
T/Q
엔트로피
코더
+
−
31
T^{-1}/Q^{-1}
30
33
32
34
필터링
36
예측기
18
38

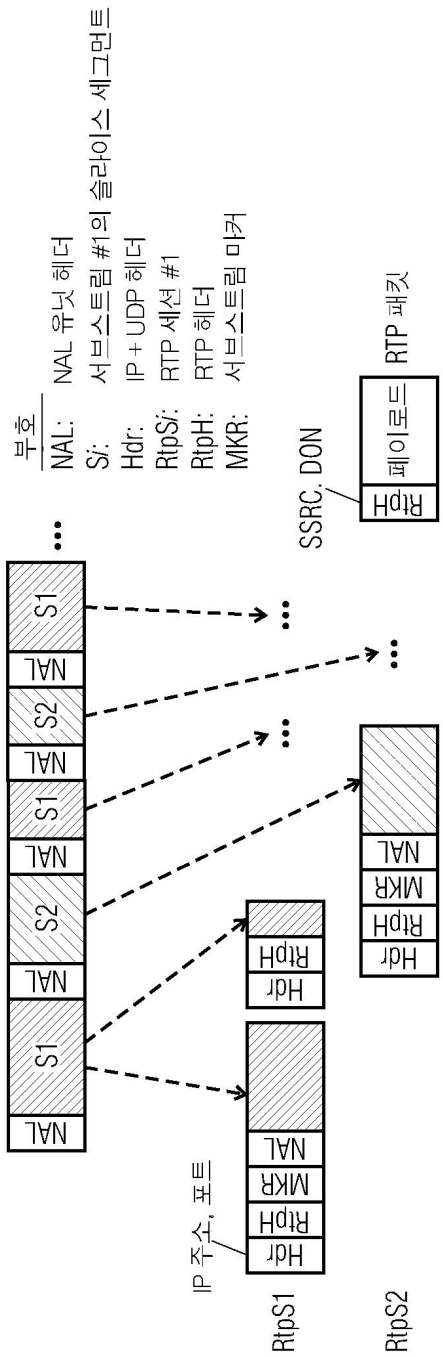
도면17


50
70
62
54
65
56
엔트로피
디코더
T^{-1}/Q^{-1}
필터링
58
40
52
68
66
예측기
64
60

도면18



도면19


부호
NAL: NAL 유닛 헤더
Si: 서브스트림 #1의 슬라이스 세그먼트
Hdr: IP + UDP 헤더
RtpSi: RTP 세션 #1
RtpH: RTP 헤더
MKR: 서브스트림 마커
S1
NAL
S2
NAL
S1
NAL
S2
NAL
S1
NAL
SSRC, DON
RtpH
페이로드
RTP 패킷
Hdr
RtpH
NAL
Hdr
RtpH
MKR
NAL
IP 주소, 포트
RtpS1
RtpS2