



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2020-0064171
(43) 공개일자 2020년06월05일

(51) 국제특허분류(Int. Cl.)
H04N 19/13 (2014.01) H04N 19/91 (2014.01)
(52) CPC특허분류
H04N 19/13 (2015.01)
H04N 19/91 (2015.01)
(21) 출원번호 10-2020-7015415(분할)
(22) 출원일자(국제) 2012년06월18일
심사청구일자 2020년05월28일
(62) 원출원 특허 10-2019-7018337
원출원일자(국제) 2012년06월18일
심사청구일자 2019년07월25일
(85) 번역문제출일자 2020년05월28일
(86) 국제출원번호 PCT/EP2012/061613
(87) 국제공개번호 WO 2012/172113
국제공개일자 2012년12월20일
(30) 우선권주장
61/497,794 2011년06월16일 미국(US)
61/508,506 2011년07월15일 미국(US)

(71) 출원인
지이 비디오 컴프레션, 엘엘씨
미국 뉴욕 12211 올버니 사우스우즈 블러바드 8
(72) 발명자
게오르게, 발레리
독일 10365 베를린 존-지크-슈트라세 24
브로스, 벤자민
독일 12055 베를린 브라운슈바이거 슈트라세 66
(뒷면에 계속)
(74) 대리인
윤의섭, 김수진

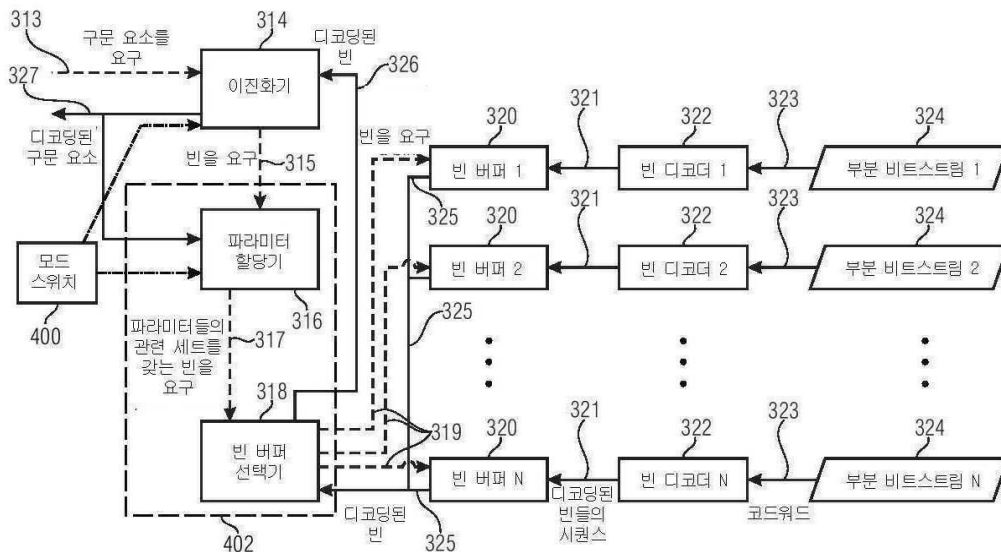
전체 청구항 수 : 총 13 항

(54) 발명의 명칭 움직임 벡터 차분값의 엔트로피 코딩

(57) 요약

수평과 수직 컴포넌트들의 이진화들을 사용하여 움직임 벡터 차분값들의 수평과 수직 컴포넌트들이 코딩되는 데이터스트림으로부터 비디오를 디코딩하기 위한 디코더가 설명되는데, 이진화들은 절단 값 아래의 수평과 수직 컴포넌트들의 도메인의 첫 번째 간격 내에서, 각각 수평과 수직 컴포넌트들의 절단된 단항 코드와, 절단 값을 포함 (뒷면에 계속)

대표도



하고 절단 값 위의 수평과 수직 컴포넌트들의 도메인의 두 번째 간격 내에서, 각각 절단 값을 위한 절단된 단항 코드 형태의 접두부호와 수평과 수직 컴포넌트들의 지수-골롬 코드 형태의 접미부호의 조합을 동일시한다. 엔트로피 인코더는 움직임 벡터 차분값들의 수평과 수직 컴포넌트들을 위하여, 움직임 벡터 차분값들의 수평과 수직 컴포넌트들에 통상적인, 절단된 단항 코드의 위치 당 정확하게 하나의 컨텍스트를 갖는 컨텍스트-적응적 이진 엔트로피 디코딩을 사용하는 데이터 스트림으로부터의 절단된 단항 코드 및, 움직임 벡터 차분값들의 이진화를 획득하기 위하여 일정한 등-확률 바이패스 방식을 사용하는 지수-골롬 코드를 유래하도록 구성된다. 디심벌라이저는 움직임 벡터 차분값들의 수평과 수직 컴포넌트들의 정수 값을 획득하기 위하여 움직임 벡터 차분값 구문 요소들의 이진화들을 탈이진화하도록 구성된다. 재구성자는 움직임 벡터 차분값들의 수평과 수직 컴포넌트들의 정수 값을 기초로 하여 비디오를 재구성하도록 구성된다.

(72) 발명자

키르호호퍼, 하이너

독일 10555 베를린 고츠코브스키슈트라쎄 5

마르페, 데트레브

독일 12161 베를린 수에드베스트코르소 70

뉘옌, 톱

독일 13055 베를린 란츠베르거 알리 217

프라이스, 마티아스

독일 10997 베를린 아이젠반슈트라쎄 20

지크만, 미샤

독일 10439 베를린 로텐베르크슈트라쎄 23

슈터게만, 얀

독일 10245 베를린 심플론슈트라쎄 55

비간트, 토마스

독일 14195 베를린 오토-아펠-슈트라쎄 52

명세서

청구범위

청구항 1

각각, 움직임 벡터 차분값들의 수평과 수직 컴포넌트들의 절대 값의 이진화를 사용하여 움직임 벡터 차분값들의 수평과 수직 컴포넌트들이 코딩되는 데이터 스트림으로부터 비디오를 디코딩하기 위한 디코더에 있어서, 상기 이진화는:

절대 값이 0보다 큰지 아닌지 여부를 특정하는 빈 지수(bin index) 0의 제 1 빈(first bin),

제 1 플래그가 1과 동일할 때 단지 코딩되고 절대 값이 1보다 큰지 아닌지 여부를 특정하는 빈 지수 1의 제 2 빈(second bin),

을 포함하는 접두부호, 및

절대 값이 2 이상일 때, 2 이상의 간격 내에서 절대 값의 나머지 값을 특정하는 1차의 Exp-Golomb 코드 형태인 접미부호,를 포함하며,

디코더는:

움직임 벡터 차분값들의 수평과 수직 컴포넌트들에 대해,

각각, 움직임 벡터 차분값들의 수평과 수직 컴포넌트들에 공통된, 접두부호의 제 1 및 제 2 빈들에 대한 정확히 하나의 콘텍스트와 함께 콘텍스트-적응적 이진 엔트로피 디코딩을 이용하여 데이터 스트림으로부터 제 1 및 제 2 빈들, 및

일정한 등-확률(equi-probability) 바이패스 방식을 이용하여 Exp-Golomb 코드,를 유도하여 움직임 벡터 차분값들의 이진화를 얻도록 구성되는 엔트로피 디코더;

움직임 벡터 차분값들의 수평 및 수직 컴포넌트들의 절대 값을 얻기 위해 절대 값의 이진화를 역이진화하도록 구성되는 디심벌라이저;

움직임 벡터 차분값들의 수평 및 수직 컴포넌트들의 절대 값에 기초하여 비디오를 재구성하도록 구성된 재구성자;를 포함하는, 디코더.

청구항 2

제 1항에 있어서, 상기 엔트로피 디코더는, 움직임 벡터 차분값들의 수평과 수직 컴포넌트들에 대해 공통된, 접두부호의 제 1 빈에 대한 정확히 하나의 제 1 콘텍스트, 및 움직임 벡터 차분값들의 수평과 수직 컴포넌트들에 대해 공통된, 접두부호의 제 2 빈에 대한 정확히 하나의 제 2 콘텍스트를 이용하도록 구성되며, 제 1 및 제 2 콘텍스트는 상이한 것을 특징으로 하는 디코더.

청구항 3

제 1항 또는 2항에 있어서,

상기 엔트로피 디코더는, 접두부호 중 현재 유도되는 빈에 대하여, 현재 유도되는 빈에 대해 선택된 콘텍스트와 연관된 현재 개연성 상태로부터, 현재 유도되는 빈에 의존하여 새로운 확률 상태로 이동시킴으로써 확률 상태 업데이트를 실행하도록 구성되는 것을 특징으로 하는 디코더.

청구항 4

제 1항 내지 3항 중 어느 한 항에 있어서, 상기 데이터 스트림은 깊이 맵을 인코딩한 것을 특징으로 하는 디코

더.

청구항 5

제 1항 내지 3항 중 어느 한 항에 있어서, 상기 데이터 스트림은 색 값들(color values)에 대응하는 정보 샘플들의 배치로서 비디오를 인코딩한 것을 특징으로 하는 디코더.

청구항 6

데이터 스트림으로 비디오를 인코딩하기 위한 인코더에 있어서,

예측된 움직임 벡터들의 예측 에러를 나타내기 위해 움직임 벡터 차분값들의 수평과 수직 컴포넌트들의 정수 값들을 설정하고 움직임 벡터들을 예측하여 움직임 벡터들을 예측적으로 코딩하며 움직임 벡터들을 사용하여 움직임 보상 예측에 의해 비디오를 예측적으로 코딩하도록 구성되는 구성자;

움직임 벡터 차분값들의 수평과 수직 컴포넌트들의 절대 값의 이진화를 획득하기 위해 움직임 벡터 차분값들의 수평과 수직 컴포넌트들의 절대 값을 이진화하도록 구성되는 심벌라이저;

상기 이진화는:

절대 값이 0보다 큰지 아닌지 여부를 특징하는 빈 지수(bin index) 0의 제 1 빈(first bin),

제 1 플래그가 1과 동일할 때 단지 코딩되고 절대 값이 1보다 큰지 아닌지 여부를 특징하는 빈 지수 1의 제 2 빈(second bin),

을 포함하는 접두부호, 및

절대 값이 2 이상일 때, 2 이상의 간격 내에서 절대 값의 나머지 값을 특징하는 1차의 Exp-Golomb 코드 형태인 접미부호,를 포함하며,

움직임 벡터 차분값들의 수평과 수직 컴포넌트들에 대해,

각각, 움직임 벡터 차분값들의 수평과 수직 컴포넌트들에 공통된, 접두부호의 제 1 및 제 2 빈들에 대한 정확히 하나의 컨텍스트와 함께 컨텍스트-적응적 이진 엔트로피 인코딩을 이용하여 데이터 스트림으로 제 1 및 제 2 빈들, 및

일정한 등-확률(equi-probability) 바이패스 방식을 이용하여 Exp-Golomb 코드,를 인코딩하도록 구성되는 엔트로피 인코더;를 포함하는, 인코더.

청구항 7

제 6항에 있어서, 상기 엔트로피 인코더는, 움직임 벡터 차분값들의 수평과 수직 컴포넌트들에 대해 공통된, 접두부호의 제 1 빈에 대한 정확히 하나의 제 1 컨텍스트, 및 움직임 벡터 차분값들의 수평과 수직 컴포넌트들에 대해 공통된, 접두부호의 제 2 빈에 대한 정확히 하나의 제 2 컨텍스트를 이용하도록 구성되며, 제 1 및 제 2 컨텍스트는 상이한 것을 특징으로 하는 인코더.

청구항 8

제 6항 또는 7항에 있어서,

상기 엔트로피 인코더는, 접두부호 중 현재 인코딩되는 빈에 대하여, 현재 인코딩되는 빈에 대해 선택된 컨텍스트와 연관된 현재 개연성 상태로부터, 현재 인코딩되는 빈에 의존하여 새로운 확률 상태로 이동시킴으로써 확률 상태 업데이트를 실행하도록 구성되는 것을 특징으로 하는 인코더.

청구항 9

제 6항 내지 8항 중 어느 한 항에 있어서, 상기 데이터 스트림으로 깊이 맵을 인코딩하도록 구성되는 것을 특징으로 하는 인코더.

청구항 10

제 6항 내지 8항 중 어느 한 항에 있어서, 상기 데이터 스트림으로 색 값(color values)들에 대응하는 정보 샘플들의 배치로서 비디오를 인코딩하도록 구성되는 것을 특징으로 하는 인코더.

청구항 11

비디오를 인코딩한 데이터 스트림에 있어서,

예측된 움직임 벡터들의 예측 에러를 나타내기 위해 움직임 벡터 차분값들의 수평과 수직 컴포넌트들의 정수 값들을 시그널링하는 데이터 스트림과 함께 움직임 벡터들을 예측하여 움직임 벡터들을 예측적으로 코딩하며 움직임 벡터들을 사용하는 움직임 보상 예측을 이용하며,

움직임 벡터 차분값들의 수평과 수직 컴포넌트들의 절대 값의 이진화의 형태로 움직임 벡터 차분값들의 수평과 수직 컴포넌트들의 절대 값이 데이터 스트림으로 코딩되며,

상기 이진화는:

절대 값이 0보다 큰지 아닌지 여부를 특징하는 빈 지수(bin index) 0의 제 1 빈(first bin),

제 1 플래그가 1과 동일할 때 단지 코딩되고 절대 값이 1보다 큰지 아닌지 여부를 특징하는 빈 지수 1의 제 2 빈(second bin),

을 포함하는 접두부호, 및

절대 값이 2 이상일 때, 2 이상의 간격 내에서 절대 값의 나머지 값을 특징하는 1차의 Exp-Golomb 코드 형태인 접미부호,를 포함하며,

각각, 움직임 벡터 차분값들의 수평과 수직 컴포넌트들에 공통된, 접두부호의 제 1 및 제 2 빈들에 대한 정확히 하나의 콘텍스트와 함께 콘텍스트-적응적 이진 엔트로피 인코딩을 이용하여 데이터 스트림으로 제 1 및 제 2 빈들이 엔트로피 코딩되며,

일정한 등-확률(equi-probability) 바이패스 방식을 이용하여 Exp-Golomb 코드가 데이터 스트림으로 엔트로피 인코딩되는, 데이터 스트림.

청구항 12

제 11항에 있어서, 깊이 맵을 인코딩한 데이터 스트림.

청구항 13

제 11항에 있어서, 색 값(color values)들에 대응하는 정보 샘플들의 배치로서 비디오를 인코딩한 데이터 스트림.

발명의 설명

기술 분야

[0001] 본 발명은 비디오 데이터를 코딩하기 위한 엔트로피 코딩 개념에 관한 것이다.

배경 기술

[0003] 종래에 많은 비디오 코덱들이 알려져 있다. 일반적으로, 이러한 코덱들은 비디오 콘텐츠를 표현하기 위하여 필요한 데이터의 양을 감소시키는데, 즉, 그것들은 데이터를 압축한다. 비디오 코딩의 맥락에서, 비디오 데이터의 압축은 바람직하게는 서로 다른 코딩 기법들을 연속하여 적용함으로써 달성된다: 픽처 콘텐츠(picture content)를 예측하기 위하여 움직임-보상 예측이 사용된다. 움직임 보상 예측뿐만 아니라 예측 잔류(residuum)로 결정되는 움직임 벡터들은 손실 없는 엔트로피 코딩의 대상이다. 데이터의 양을 더 감소시키기 위하여, 움직임 벡터들 자체는 예측의 대상이 되며 따라서 움직임 벡터 예측 잔류를 표현하는 움직임 벡터 차분값들만이 엔트로피 인코딩되어야만 한다. 예를 들면, H.264에서, 정보를 움직임 벡터 차분값들 상에 전송하기 위하여 방금 설명된 과정이 적용된다. 특히, 움직임 벡터 차분값들은 절단된(truncated) 단항 코드의 조합과 상응하는 빈 스트링 내로, 그리고 지수 골롬(Golomb) 코드 상의 특정 절단 값으로부터, 이진화된다. 지수 골롬 코드의 빈들은 0.5의 고정된 확률을 갖는 등-확률(equi-probability) 바이패스 방식을 사용하여 쉽게 코딩되나, 첫 번째 빈들을 위하여 일부 컨텍스트들이 제공된다. 절단 값(cutoff value)은 9가 되도록 선택된다. 따라서, 움직임 벡터 차분값들을 코딩하기 위하여 높은 양의 컨텍스트가 제공된다.

[0005] 그러나, 높은 수의 컨텍스트의 제공은 코딩 복잡도를 증가시킬 뿐만 아니라 코딩 효율에 부정적으로 영향을 미친다: 만일 컨텍스트가 너무 적게 방문되면, 확률 적용, 즉, 엔트로피 코딩의 원인 동안에 각각의 컨텍스트와 관련된 확률 추정의 적용은 효율적으로 실행하는데 실패한다. 따라서, 부적절하게 적용된 확률 적용은 실제 심벌(symbol) 통계를 추정한다. 게다가, 만일 이진화의 특정 빈을 위하여, 일부 컨텍스트들이 제공되면, 그것들 사이의 선택은 그러한 필요성이 디코딩 과정의 실행을 방해할 수 있는 이웃하는 빈들/구문 요소 값들의 검사를 필요하게 만들 수 있다. 다른 한편으로, 만일 컨텍스트의 수가 너무 낮게 제공되면, 매우 다양한 실제 심벌 통계들이 하나의 컨텍스트 내에 그룹화되며 따라서, 그러한 컨텍스트와 관련된 확률 추정은 그것과 관련된 빈들을 효율적으로 인코딩하는데 실패한다.

발명의 내용

해결하려는 과제

[0007] 따라서, 그러한 코딩 개념을 제공하는 것이 본 발명의 목적이다.

과제의 해결 수단

[0009] 본 발명의 목적은 첨부된 독립항들의 주제에 의해 달성된다.

[0011] 본 발명의 기본 발견은 움직임 벡터 차분값들을 이진화하기 위하여 절단된 단항 코드가 사용되는 것까지, 절단된 단항 코드의 단지 두 개의 빈 위치가 존재하도록 하기 위하여 2에 이르기까지 절단 값을 감소시킴으로써, 그리고 만일 절단 값으로부터 움직임 벡터 차분값들의 이진화를 위하여 지수 골롬 코드를 위한 하나의 순차가 사용되면, 그리고 만일 부가적으로, 각각 절단된 단항 코드의 두 개의 빈 위치를 위하여 정확하게 하나의 컨텍스트가 제공되면, 따라서 이웃하는 이미지 블록들의 빈들 또는 구문 요소들을 기초로 하는 컨텍스트 선택은 필요하지 않으며 이러한 빈 위치들에서 빈들의 블록들 내로의 너무 미세한 분류는 확률 적용이 적절하게 작용하도록 방지되면, 그리고 만일 수평 및 수직 컴포넌트들을 위하여 동일한 컨텍스트가 사용되고 그렇게 함으로써 너무 미세한 컨텍스트 세분화의 부정적인 효과들을 감소시키면, 움직임 벡터 차분값들의 엔트로피 코딩의 코딩 효율이 더 증가될 수 있다는 것이다.

발명의 효과

[0013]

또한, 움직임 벡터 차분값들의 엔트로피 코딩과 관련하여 방금 설명된 설정들은 이를 움직임 벡터들을 실행하고 전송되는 움직임 벡터 차분값들의 필요한 양을 감소시키는 개선된 방법과 결합할 때 특히 가치가 있다는 것이 알려졌다. 예를 들면, 움직임 벡터 예측변수(예측자, predictor)의 순차 리스트를 획득하기 위하여 다중 움직임 벡터 예측변수들이 제공될 수 있으며, 문제의 움직임 벡터 차분값에 의해 예측 잔여(residual)가 표현되는 실제 움직임 벡터 차분값을 결정하기 위하여 움직임 벡터의 이러한 리스트 내의 지수가 사용될 수 있다. 비록 사용되는 리스트 지수 상의 정보가 디코딩 면에서의 데이터 스트림으로부터 유도되어야만 하나, 움직임 벡터들의 전체 예측 양은 증가하며 따라서, 움직임 벡터 차분값들의 크기는 감소하며 따라서 코딩 효율은 더 증가하고 절단 값의 감소 및 움직임 벡터 차분값들의 수평과 수직 컴포넌트들을 위한 콘텍스트의 공용은 그러한 개선된 움직임 벡터 예측에 적합하다. 다른 한편으로, 데이터 스트림 내에 전송되는 움직임 벡터 차분값들의 수를 감소시키기 위하여 머징(merging)이 사용될 수 있다: 이를 위하여, 머징 정보가 블록들의 그룹으로 그룹화된 블록들의 세분화(subdivision)의 디코더 블록들에 신호를 전달하는 데이터 스트림 내에 전달될 수 있다. 움직임 벡터 차분값들은 그리고 나서 개별 블록들 대신에 이러한 병합된 그룹들의 유닛으로 데이터 스트림 내에 전송될 수 있으며, 그렇게 함으로써 전송되어야만 하는 움직임 벡터 차분값들의 수를 감소시킬 수 있다. 블록들의 이러한 클러스터링(clustering)이 이웃하는 움직임 벡터 차분값들 사이의 상호 상관을 감소시키기 때문에, 방금 언급된 하나의 빈 위치를 위한 일부 콘텍스트들의 제공의 생략은 이웃하는 움직임 벡터 차분값들에 따라 너무 미세한 분류로부터 콘텍스트들 내로의 엔트로피 코딩 전략을 방지한다. 오히려, 머징 개념은 이웃하는 블록들의 움직임 벡터 차분값들 사이의 상호 상관을 사용하며 따라서, 하나의 빈 위치를 위하여 하나의 콘텍스트면(수평 및 수직 컴포넌트들을 위하여 동일하게) 충분하다.

도면의 간단한 설명

[0014]

본 발명의 바람직한 실시 예들이 도면들과 관련하여 아래에 설명될 것이다.

도 1은 일 실시 예에 따른 인코더의 블록 다이어그램을 도시한다.

도 2a-2c는 블록들 내의 픽처와 같은 샘플 어레이의 서로 다른 세분들을 개략적으로 도시한다.

도 3은 일 실시 예에 따른 디코더의 블록 다이어그램을 도시한다.

도 4는 일 실시 예에 따른 인코더의 블록 다이어그램을 더 상세히 도시한다.

도 5는 일 실시 예에 따른 디코더의 블록 다이어그램을 더 상세히 도시한다.

도 6은 공간 도메인으로부터 스펙트럼 도메인 내로의 블록의 변환, 결과로서 생기는 변환 블록, 및 그것의 재변환을 개략적으로 도시한다.

도 7은 일 실시 예에 따른 인코더의 블록 다이어그램을 도시한다.

도 8은 일 실시 예에 따른, 도 8의 인코더에 의해 발생하는 디코딩 비트스트림에 적합한 디코더의 블록 다이어그램을 도시한다.

도 9는 일 실시 예에 따른 일 실시 예에 따른 다중화 부분 비트스트림들을 갖는 데이터 패킷을 나타내는 개략적인 다이어그램을 도시한다.

도 10은 또 다른 실시 예에 따라 고정된-크기 세그먼트들을 사용하는 대안의 세그멘테이션을 갖는 데이터 패킷을 나타내는 개략적인 다이어그램을 도시한다.

도 11은 일 실시 예에 따른 모드 스위칭을 제공하는 디코더를 도시한다.

도 12는 또 다른 실시 예에 따른 모드 스위칭을 제공하는 디코더를 도시한다.

도 13은 일 실시 예에 따라 도 11의 디코더에 꼭 맞는 인코더를 도시한다.

도 14는 일 실시 예에 따라 도 12의 디코더에 꼭 맞는 인코더를 도시한다.

도 15는 pStateCtx 및 fullCtxState/256^{**}E^{**}의 매핑을 도시한다.

도 16은 본 발명의 일 실시 예에 따른 디코더를 도시한다.

도 17은 본 발명의 일 실시 예에 따른 인코더를 도시한다.

도 18은 본 발명의 일 실시 예에 따른 움직임 벡터 차분값 이진화를 개략적으로 도시한다.

도 19는 일 실시 예에 따른 병합 개념을 개략적으로 도시한다.

도 20은 움직임 벡터 예측 전략을 개략적으로 도시한다.

발명을 실시하기 위한 구체적인 내용

- [0015] 도면들의 설명 동안에, 일부 이러한 도면들에서 발생하는 구성요소들은 이러한 도면들의 각각에서 동일한 참조 부호로 표시되며 불필요한 반복을 방지하기 위하여 이러한 구성요소들의 반복적인 설명은 기능성에 관한 한 방지된다. 그럼에도 불구하고, 하나의 도면에 대하여 제공되는 기능성과 설명들은 분명히 반대로 표시되지 않는 한 또한 다른 도면들에 적용되어야 한다.
- [0017] 다음에서, 우선, 도 1 내지 10을 참조하여, 일반적인 비디오 코딩 개념의 실시 예들이 설명된다. 도 1 내지 6은 구문 레벨 상의 비디오 코덱 작동의 일부분에 관한 것이다. 다음의 도면들 8 내지 10은 구문 요소 스트림의 데이터 스트림으로의 변환과 그 반대의 경우에 관한 코드의 일부분을 위한 실시 예에 관한 것이다. 그리고 나서, 본 발명의 특정 양상들과 실시 예들이 도 1 내지 10과 관련하여 대표적으로 나타내는 일반적인 개념의 가능한 구현 형태로 설명된다.
- [0019] 도 1은 본 발명의 양상들이 구현될 수 있는 인코더(10)를 위한 일 실시 예를 도시한다.
- [0021] 인코더는 정보 샘플들의 어레이(20)를 데이터 스트림 내로 인코딩한다. 정보 샘플들의 어레이는 예를 들면, 명도(brightness) 값들, 색 값들, 광도 값들, 채도(chroma) 값들 등과 상응하는 정보 샘플들을 표현할 수 있다. 그러나, 정보 샘플들은 또한 예를 들면, 광센서의 시간 등에 의해 발생하는 깊이 맵(depth map)인 샘플 어레이(20)의 경우에 깊이 값들일 수 있다.
- [0023] 인코더는 블록 기반 인코더이다. 즉, 인코더(10)는 샘플 어레이를 블록들(40)의 유닛들 내의 데이터 스트림(30) 내로 인코딩한다. 블록들(40)의 유닛들 내의 인코딩은 인코더(10)가 서로 완전히 독립적으로 이러한 블록들(40)을 인코딩하는 것을 반드시 의미하지는 않는다. 오히려, 인코더(10)는 나머지 블록들을 추정하거나 인트라(intra)-예측하기 위하여 이전에 인코딩된 블록들의 재구성을 할 수 있으며, 코딩 파라미터들의 설정을 위하여, 즉, 각각의 블록과 상응하는 각각의 샘플 어레이 영역이 코딩되는 방법을 설정하기 위하여 블록들의 입도(granularity)를 사용할 수 있다.
- [0025] 또한, 인코더(10)는 변환 코더이다. 즉, 인코더(10)는 공간 도메인으로부터 스펙트럼 도메인으로 각각의 블록(40) 내의 정보 샘플들을 전달하기 위하여 변환을 사용함으로써 블록들(40)을 인코딩한다. 고속 푸리에 변환(FFT)의 이산 코사인 변환(DCT) 등과 같은 이차원 변환 등이 사용될 수 있다. 바람직하게는, 블록들(40)은 정사각형 형태 또는 직사각형 형태이다.
- [0027] 샘플 어레이(20)의 도 1에 도시된 블록들(40) 내로의 세분화는 단지 설명 목적을 제공한다. 도 1은 비-오버래핑 방식으로 서로 인접해 있는 정사각형 또는 직사각형 블록들(40)의 정규 2차원 배치로 세분화되는 것으로서의 샘플 어레이(20)를 도시한다. 블록들(40)의 크기는 미리 결정될 수 있다. 즉, 인코더(10)는 데이터 스트림(30) 내의 블록들(40)의 블록 크기에 대한 정보를 디코딩 면으로 전달하지 않을 수 있다. 예를 들면, 디코더는 미리 결정된 블록 크기를 기대할 수 있다.
- [0029] 그러나, 일부 대안들이 가능하다. 예를 들면, 블록들은 서로 오버랩할 수 있다. 그러나, 오버래핑은 각각의 블

록이 어떠한 이웃한 블록에 의해 오버래핑되지 않는 부를 가질 정도로 또는 블록들의 각각의 샘플이 최대로, 미리 결정된 방향을 따라 현재 블록에 병렬(juxtaposition)로 배치되는 이웃한 블록들 중에서 하나의 블록에 의해 오버래핑되는 것과 같이 제한될 수 있다. 후자는 현재 블록을 완전히 포함하기 위하여 왼쪽과 오른쪽 이웃 블록들이 현재 블록을 오버랩할 수 있으나 서로 오버레이하지(overlay) 않을 수 있다는 것을 의미하며, 수직 및 대각선 방향의 이웃자리들 위하여도 동일하게 적용된다.

[0031] 또 다른 대안으로서, 샘플 어레이(20)의 블록들(40)로의 세분화는 비트스트림(30)을 거쳐 디코더 면으로 전달되도록 사용되는 세분화 상의 세분화 정보를 갖는 인코더(10)에 의해 샘플 어레이(20)의 콘텐츠에 적용될 수 있다.

[0033] 도 2a 내지 2c는 샘플 어레이(20)의 블록들(40)로의 세분화를 위한 서로 다른 실시 예를 도시한다. 도 2a는 샘플 어레이(20)의, 증가하는 크기로 40a, 40b, 40c 및 40d에서 표시되는 대표 블록들을 갖는, 서로 다른 크기들의 블록들로의 쿼드트리(quadtrees) 기반 세분화를 도시한다. 도 2a의 세분화에 따르면, 샘플 어레이(20)는 우선 트리 블록들(40d)의 정규 2차원 배치로 나뉘지며, 이는 차례로, 특정 트리 블록(40d)이 쿼드트리 구조에 따라 더 세분화될 수 있는지에 따라 그것들과 관련된 개별 세분화 정보를 갖는다. 블록(40)의 왼쪽에 대한 트리 블록은 바람직하게는 쿼드트리 구조에 따라 더 작은 블록들로 세분화된다. 인코더(20)는 도 2a의 실선과 쉐선으로 도시된 각각의 블록을 위한 하나의 2차원 변환을 실행할 수 있다. 바꾸어 말하면, 인코더(20)는 블록 세분화의 유닛들로 어레이(20)를 변환할 수 있다.

[0035] 쿼드트리 기반 세분화 대신에 더 일반적인 다중 트리 기반 세분화가 사용될 수 있으며 계층 레벨당 자식 노드(child node)들의 수는 서로 다른 계층 레벨들에서 다를 수 있다.

[0037] 도 2b는 세분화의 또 다른 실시 예를 도시한다. 도 2b에 따르면, 샘플 어레이(20)는 우선 서로 인접한 비-오버래핑 방식으로 정규 2차원 배치로 배치되는 매크로블록(40b, macroblock)들로 세분화되는데 각각의 매크로블록(40b)은 서로 다른 매크로블록들을 위하여 서로 다른 세분화 입도들을 달성하기 위하여 매크로블록이 세분화되지 않거나, 또는 만일 세분화되면, 정규 2차원 방식으로 동등한 크기의 서브-블록들로 세분화되는가에 따라 그것들과 관련된 세분화 정보를 갖는다. 그 결과는 40a, 40b 및 40a'에서 표시되는 서로 다른 크기의 대표들을 갖는 서로 다른 크기의 블록들(40)로의 샘플 어레이(20)의 세분화이다. 도 2a에서와 같이, 인코더(10)는 실선과 쉐선으로 도 2b에 도시된 각각의 블록들 상에 2차원 변환을 실행한다. 도 2c는 뒤에 설명될 것이다.

[0039] 도 3은 샘플 어레이(20)의 재구성된 버전(60)을 재구성하기 위하여 인코더(10)에 의해 발생되는 데이터 스트림(30)을 디코딩할 수 있는 디코더(50)를 도시한다. 디코더(50)는 데이터 스트림(30)으로부터 각각의 블록들(40)을 위한 변환 계수 블록을 추출하고 각각의 변환 계수 블록들 상에 역 변환을 실행함으로써 재구성된 버전(60)을 재구성한다.

[0041] 인코더(10)와 디코더(50)는 각각 변환 계수 블록들 상의 정보를 삽입하고, 데이터 스트림으로부터 이러한 정보를 추출하기 위하여 엔트로피 인코딩/디코딩을 실행하도록 구성될 수 있다. 서로 실시 예들에 따른 이와 관련된 상세한 설명이 뒤에 설명된다. 데이터 스트림(30)은 샘플 어레이(20)의 모든 블록(40)을 위하여 변환 계수 블록들 상의 정보를 반드시 포함하지는 않는다는 것을 이해하여야 한다. 오히려, 블록들(40)의 서브셋으로서 또 다른 방법으로 비트스트림(30) 내로 코딩될 수 있다. 예를 들면, 인코더(10)는 비트스트림(30) 내로의 삽입을 갖는 블록들(40)의 특정 블록을 위한 변환 계수 블록의 삽입을 삼가도록 결정할 수 있으며 대신에 대안의 코딩 파라미터들이 디코더(50)가 예측하거나 또는 그렇지 않으면 각각의 블록을 재구성된 버전(60) 내에 채우는 것을 가능하게 한다. 예를 들면, 인코더(10)는 텍스처(texture) 합성을 통하여 디코더에 의해 디코더 면에서 채워질 수 있는 샘플 어레이(20) 내의 블록들을 위치시키고 이를 비트스트림 내에 그에 알맞게 표시하기 위하여 텍스처 분석을 실행할 수 있다.

[0043] 다음의 도면들과 관련하여 설명되는 것과 같이, 변환 계수 블록들은 샘플 어레이(20)의 각각의 블록(40)의 오리지널 정보 샘플들이 스펙트럼 도메인 표현을 반드시 표현할 필요는 없다. 오히려, 그러한 변환 계수 블록은 각각의 블록(40)의 예측 잔류의 스펙트럼 도메인 표현을 표현한다. 도 4는 그러한 인코더를 위한 일 실시 예를 도시한다. 도 4의 인코더는 변환 스테이지(100), 엔트로피 코더(102), 역 변환 스테이지(104), 예측변수(110) 및 감산기(108, subtractor)뿐만 아니라 가산기(110)를 포함한다. 감산기(108), 변환 스테이지(100) 및 엔트로피 코더(102)는 도 4의 인코더의 입력(112)과 출력(114) 사이에 언급된 순서로 직렬로 연결된다. 역 변환 스테이지(104), 가산기(110) 및 예측변수(106)는 변환 스테이지 및 감산기(108)의 역 입력 사이에 언급된 순서로 연결되며, 예측변수(106)의 출력은 또한 가산기(110)의 또 다른 입력에 연결된다.

[0045] 도 4의 코더는 예측 변환 기반 코더이다. 즉, 입력(112)으로 들어가는 샘플 어레이(20)의 블록들이 표현 시간에서 현재 샘플 어레이(20)를 선행하거나 따라갈 수 있는 이전에 인코딩되고 재구성된 동일한 샘플 어레이(20)의 부들 또는 이전에 코딩되고 재구성된 다른 샘플 어레이들로부터 예측된다. 예측은 예측변수(106)에 의해 예측된다. 감산기(108)는 그러한 오리지널 블록으로부터 예측을 감산하고 변환 스테이지(100)는 예측 잔류들 상에 2차원 변환을 실행한다. 2차원 변환 자체 또는 변환 스테이지(100) 내부의 그 뒤의 측정은 변환 계수 블록들 내의 변환 계수들의 양자화에 이르게 할 수 있다. 양자화된 변환 계수 블록들은 예를 들면, 출력(114)에서 출력되는 결과로서 생기는 데이터 스트림을 갖는 엔트로피 인코더(102) 내의 엔트로피 인코딩에 의해, 손실 없이 코딩된다. 역 변환 스테이지(104)는 양자화된 잔류를 재구성하며, 가산기(110)는 차례로, 예측변수(106)가 앞서 언급된 현재 인코딩되는 예측 블록들을 예측할 수 있는가를 기초로 하여 재구성된 정보 샘플들을 획득하기 위하여 재구성된 잔류를 상응하는 예측과 결합한다. 예측변수(106)는 블록들을 예측하기 위하여 화면 내 예측 방식들 및 화면간 예측 방식들과 같은 서로 다른 예측 방식들을 사용할 수 있으며 데이터 스트림 내로의 삽입을 위하여 예측 파라미터들이 엔트로피 인코더(102)로 전송된다. 각각의 화면내 예측 블록의 움직임 데이터를 위하여, 각각의 움직임 데이터는 디코딩 면의 예측을 다시 행하는 것을 가능하게 하기 위하여 엔트로피 인코더(114)를 거쳐 비트스트림 내로 삽입된다. 픽처의 예측 블록을 위한 움직임 데이터는 예를 들면, 이웃하는 이미 인코딩된 예측 블록들의 움직임 벡터로부터 규정된 방법에 의해 유래하는 움직임 벡터 예측변수에 대하여 현재 예측 블록을 위한 움직임 벡터를 다르게 코딩하는 움직임 벡터 차분값을 표현하는 구문 요소를 포함하는 구문 부를 포함할 수 있다.

[0047] 즉, 도 4에 따르면, 변환 계수 블록들은 그것의 실제 정보 샘플들보다는 오히려 샘플 어레이의 잔류의 스펙트럼 표현을 표현한다. 즉, 도 4에 따르면, 구문 요소들의 시퀀스는 데이터 스트림(114) 내로 엔트로피 인코딩되기 위하여 엔트로피 인코더(102)로 들어갈 수 있다. 구문 요소들의 시퀀스는 화면간 예측 블록들을 위한 움직임 벡터 차분값 구문 요소들 및 중요한 변환 계수 레벨들의 위치들을 표시하는 유의계수 맵(significance map)에 관한 구문 요소들뿐만 아니라 변환 블록들을 위하여, 자체로 중요한 변환 계수 레벨들을 정의하는 구문 요소들을 포함할 수 있다.

[0049] 도 4의 실시 예를 위하여 일부 대안들이 존재한다는 것을 이해하여야 하며 이들 중 일부는 첨부된 도 4의 설명과 통합되는 본 명세서의 도입 부분에서 설명되었다.

[0051] 도 5는 도 4의 인코더에 의해 발생하는 데이터 스트림을 디코딩할 수 있는 디코더를 도시한다. 도 5의 디코더는 엔트로피 디코더(150), 역 변환 스테이지(152), 가산기(154) 및 예측변수(156)를 포함한다. 엔트로피 디코더(150), 역 변환 스테이지(152), 및 가산기(154)는 언급된 순서로 도 5의 디코더의 입력(158)과 출력(160) 사이에 직렬로 연결된다. 엔트로피 디코더(150)의 또 다른 출력은 차례로, 가산기(154)의 출력과 그것의 또 다른 입력 사이에 연결되는 예측변수(156)에 연결된다. 엔트로피 디코더(150)는 입력(158)에서 도 5의 디코더로 들어가는 데이터 스트림으로부터 변환 계수 블록들을 추출하며 역 변환은 잔류 신호를 획득하기 위하여 스테이지(152)에서 변환 계수 블록들에 적용된다. 잔류 신호는 출력(160)에서 샘플 어레이의 재구성되는 버전의 재구성되는 블록을 획득하기 위하여 가산기(154)에서 예측변수(156)로부터의 예측과 결합된다. 재구성된 버전을 기초로 하여, 예측변수(156)는 예측들을 발생시키며 그렇게 함으로써 인코더 면에서 예측변수(156)에 의해 실행된 예측

들을 재구성한다. 인코더 면에서 사용된 것과 동일한 예측들을 획득하기 위하여, 예측변수(156)는 엔트로피 디코더(150)가 또한 입력(158)에서 데이터 스트림으로부터 획득하는 예측 파라미터들을 사용한다.

[0053] 위에서 설명된 실시 예들에서, 잔류의 예측과 변환이 실행되는 공간 입도는 서로 동일할 필요가 없다는 것을 이해하여야 한다. 이는 도 2c에 도시된다. 이러한 도면은 실선을 갖는 예측 입도와 채선을 갖는 잔류 입도의 예측 블록들을 위한 세분화를 도시한다. 알 수 있는 것과 같이, 세분화들은 서로 독립적인 인코더에 의해 선택될 수 있다. 더 정확히 설명하면, 데이터 스트림 구문은 예측 세분화와 관계없이 잔류 세분화의 정의를 허용할 수 있다. 대안으로서, 잔류 세분화는 각각의 잔류 블록이 예측 블록의 적절한 서브셋과 동일하도록 예측 세분화의 확장일 수 있다. 이는 예를 들면, 다시 예측 입도가 실선으로 도시되고 잔류 입도가 채선으로 도시된, 도 2a와 도 2b 상에 도시된다. 즉, 도 2a-2c에서, 그것들과 관련된 동일한 참조 부호를 갖는 모든 블록은 하나의 2차원 변환이 실행될 수 있는 잔류 블록들일 수 있으며 예를 들면, 채선 블록들(40a)을 포함하는 더 큰 실선 블록들은 예측 파라미터 설정이 개별적으로 실행되는 예측 블록들일 수 있다.

[0055] 위의 실시 예들은 공통으로 (전류 또는 오리지널) 샘플들의 블록이 디코더 면에서 변환 계수 블록 내로 변환되고 이는 차례로, 디코더 면에서 재구성된 샘플들의 블록 내로 역변환되는 사실을 갖는다. 이는 도 6에 도시된다. 도 6은 샘플들(202)의 블록을 도시한다. 도 6의 경우에 있어서, 이러한 블록(200)은 바람직하게는 정사각형 및 크기가 4×4 샘플들(202)이다. 샘플들(202)은 수평 방향(x)과 수직 방향(y)을 따라 규칙적으로 배치된다. 위에서 설명된 2차원 변환(T)에 의해, 블록(200)은 스펙트럼 도메인 내로, 주로 변환 계수들(206)의 블록(204) 내로 변환되며, 변환 블록(204)은 블록과 동일한 크기이다. 즉, 변환 블록(204)은 두 수직 방향과 수직 방향으로, 블록(200)이 샘플들을 갖는 것만큼 많은 변환 계수들(206)을 갖는다. 그러나, 변환(T)이 스펙트럼 변환이기 때문에, 변환 블록(204) 내의 변환 계수들(206)의 위치들은 공간 위치들과 상응하지 않고 오히려 블록(200)의 콘텍스트의 스펙트럼 컴포넌트들과 상응한다. 특히, 변환 블록(204)의 수평 축은 이를 따라 수평 방향에서의 스펙트럼 주파수가 단조적으로(monotonically) 증가하는 축과 상응하며 수직 축은 이를 따라 수직 방향에서의 공간 주파수가 단조적으로 증가하는 축과 상응하는데 이산 코사인 컴포넌트 변환 계수는 블록(204)의 구석-여기서 바람직하게는 상부 왼쪽 구석- 내에 위치되며 따라서, 하부 오른쪽 구석에서, 두 수평과 수직 방향에서 가장 높은 주파수와 상응하는 변환 계수(206)가 위치된다. 공간 방향, 특정 변환 계수(206)가 속하는 공간 주파수의 무시는 일반적으로 상부 왼쪽 구석으로부터 하부 오른쪽 구석으로 증가한다. 역 변환(T-1)에 의해, 변환 블록(204)은 블록(200)의 카피(208)를 다시 획득하기 위하여, 스펙트럼 도메인으로부터 공간 도메인으로 재전달된다. 변환 동안에 양자화/손실이 도입되지 않는 경우에서, 재구성은 완벽할 수 있다.

[0057] 위에서 이미 설명된 것과 같이, 도 6으로부터 블록(200)의 더 큰 블록 크기들이 결과로서 생기는 스펙트럼 표현(204)의 스펙트럼 해상도를 증가시킨다는 것을 알 수 있다. 다른 한편으로는, 양자화 잡음이 전체 블록(204)에 걸쳐 확산하는 경향이 있고 따라서 블록들(200) 내의 분리되고 매우 국한적인 대상은 양자화 잡음에 기인하여 오리지널 블록(200)에 대하여 재변환된 블록의 편차들에 이르게 하는 경향이 있다. 그러나, 더 큰 블록들을 사용하는 주 장점은 한편으로는 중요한, 즉, 비-제로 (양자화된) 변환 계수들, 즉, 레벨들 사이의 비율 및 다른 한편으로는 무의(insignificant) 계수들의 수가 작은 블록들과 비교하여 큰 블록들 내에서 감소될 수 있으며 그렇게 함으로써 더 나은 코딩 효율을 가능하게 할 수 있다는 것이다. 바꾸어 말하면, 흔히, 중요한 변환 계수 레벨들, 즉, 제로로 양자화되지 않는 계수들은 변환 블록(204)에 걸쳐 드물게 분포한다. 이 때문에, 아래에 더 상세히 설명될 실시 예들에 따르면, 중요한 변환 계수 레벨들의 위치들은 유의계수 맵에 의해 데이터 스트림 내에 신호가 보내진다. 이와는 별도로, 중요한 변환 계수의 값들, 즉, 양자화된 변환 계수들의 경우의 변환 계수 레벨들은 데이터 스트림 내에 전송된다.

[0059] 위에서 설명된 모든 인코더와 디코더는 따라서 구문 요소들의 특정 구문을 처리하도록 구성된다. 즉, 변환 계수 레벨들과 같은 앞서 언급된 구문 요소들, 변환 블록들의 유의계수 맵에 관한 구문 요소들, 화면간-예측 블록들에 관한 구문 요소들 등은 규정된 방법으로 데이터 스트림 내에 연속적으로 배치되는 것으로 가정된다. 그러한 규정된 방법은 예를 들면, H.264 표준 또는 다른 비디오 코덱들에서, 그것이 행해지는 한 슈도(pseudo) 코드의 형태로 표현될 수 있다.

- [0061] 또한 바꾸어 말하면, 위의 설명은 주로 미디어 데이터, 여기서는 바람직하게는 비디오 데이터의, 그것들 중에서 특정 구문 형태들, 그것의 의미와 순차를 규정하는 미리 정의된 구문 구조에 따른 구문 요소들의 시퀀스로의 전환을 처리하였다. 도 4와 5의 엔트로피 인코더와 엔트로피 디코더는 아래에 설명되는 것과 같이 작동하고 구성될 수 있다. 이들은 구문 요소 시퀀스와 데이터 스트림, 즉, 심벌 또는 비트스트림 사이의 전환에도 책임이 있다.
- [0063] 일 실시 예에 따른 엔트로피 인코더가 도 7에 도시된다. 인코더는 구문 요소들(301)의 스트림을 손실 없이 두 개 또는 그 이상의 부분 비트스트림(312)으로 전환한다.
- [0065] 본 발명의 바람직한 실시 예에서, 각각의 구문 요소(301)는 하나 또는 그 이상의 카테고리의 세트의 카테고리, 즉, 구문 요소 형태와 관련된다. 일례로서, 카테고리들은 구문 요소의 형태를 지정할 수 있다. 하이브리드 비디오 코딩의 맥락에서, 개별 카테고리는 매크로블록 코딩 방식들, 블록 코딩 방식들, 레퍼런스 픽처 지수들, 움직임 벡터 차분값들, 세분화 플래그들, 코딩된 블록 플래그들, 양자화 파라미터들, 변환 계수 레벨들 등과 관련될 수 있다. 오디오, 음성, 텍스트, 문서, 또는 일반적인 데이터 코딩과 같은 다른 적용 영역들에서, 구문 요소들의 서로 다른 범주화가 가능하다.
- [0067] 일반적으로, 각각의 구문 요소는 값의 유한 또는 계수가능 무한 세트의 값을 취할 수 있는데, 가능한 구문 요소 값들의 세트는 서로 다른 구문 요소 카테고리들을 위하여 다를 수 있다. 예를 들면, 이진 구문 요소들뿐만 아니라 정수 값 구문 요소가 존재한다.
- [0069] 인코딩과 디코딩의 복잡도의 감소 및 서로 다른 구문 요소들과 구문 요소 카테고리들을 위한 일반적인 인코딩과 디코딩 디자인의 허용을 위하여 구문 요소들(301)은 이진 결정들의 순차 세트로 전환되고 이러한 이진 결정들은 그리고 나서 단순한 이진 코딩 알고리즘들에 의해 처리된다. 따라서, 이진화기(302, binarizer)는 각각의 구문 요소(301)의 값을 빈들(303)의 시퀀스(또는 스트림 또는 워드) 상으로 전단사 방식으로(bijectively) 매핑한다. 빈들(303)의 시퀀스는 순차 이진 결정들의 세트를 표현한다. 각각의 빈(303) 또는 이진 결정은 두 값들의 세트 중 하나의 값, 예를 들면, 값들 0과 1 중 하나를 취할 수 있다. 특정 구문 요소 카테고리를 위한 이진화 전략은 가능한 구문 요소 값들의 세트 및/또는 특정 카테고리를 위한 구문 요소의 다른 특성들에 의존할 수 있다.
- [0071] 테이블 1은 계수가능 무한 세트들을 위한 3가지 바람직한 이진화 전략을 도시한다. 계수가능 무한 세트들을 위한 이진화 전략은 또한 구문 요소 값들의 유한 세트들을 위하여 적용될 수 있다. 특히 구문 요소 값들의 큰 유한 세트들을 위하여, 비능률(빈들의 사용되지 않는 시퀀스들로부터 야기하는)이 무시될 수 있으나, 그러한 이진화 전략들의 보편성은 복잡도와 메모리 요구와 관련하여 장점을 제공한다. 구문 요소 값들의 작은 유한 세트들을 위하여, 흔히 이진화 전략을 가능한 심벌 값들의 수에 적용하는 것이 바람직하다.
- [0073] 테이블 2는 8 값의 유한 세트들을 위한 3가지 바람직한 이진화 전략을 도시한다. 유한 세트들을 위한 이진화 전략들은 빈 시퀀스들의 유한 세트들이 중복이 없는 코드를 표현하는(및 잠재적으로 빈 시퀀스들을 기록하는) 방법으로 빈들의 일부 시퀀스를 변형함으로써 계수가능 무한 세트들을 위한 보편적인 단항 이진화 전략들로부터 유래할 수 있다. 일례로서, 테이블 2의 절단된 단항 이진화 전략은 보편적인 단항 이진화의 구문 요소(7)를 위한 빈 시퀀스를 변형함으로써 생성되었다(테이블 1 참조). 테이블 2의 순차 0의 절단되고 기록된 지수-골롬(exponential-Golomb) 이진화는 보편적인 지수-골롬 순차 0 이진화의 구문 요소 7을 위한 빈 시퀀스를 변형하고(테이블 1 참조) 빈 시퀀스들을 기록함으로써(심벌 7을 위한 절단된 빈 시퀀스는 심벌 1로 지정되었다) 생성되었다. 구문 요소들의 유한 세트들을 위하여, 테이블 2의 마지막 칼럼에 예시된 것과 타이, 비-체계적/비-보편적 이진화 전략들을 사용하는 것이 또한 가능하다.

[0075] 테이블 1. 계수가능 무한 세트들(또는 큰 유한 세트들)의 이진화 예들.

심별 값	단항 이진화	지수-골롬 순차 0 이진화	지수-골롬 순차 1 이진화
0	1	1	10
1	01	010	11
2	001	011	0100
3	0001	0010 0	0101
4	0000 1	0010 1	0110
5	0000 01	0011 0	0111
6	0000 001	0011 1	0010 00
7	0000 0001	0001 000	0010 01
...	...	...	...

[0076]

[0078] 테이블 2. 유한 세트들의 이진화 예들.

심별 값	절단된 단항 이진화	절단되고 레코딩된 지수-골롬 순차 0 이진화	비-체계적 이진화
0	1	1	000
1	01	000	001
2	001	010	01
3	0001	011	1000
4	0000 1	0010 0	1001
5	0000 01	0010 1	1010
6	0000 001	0011 0	1011 0
7	0000 000	0011 1	1011 1

[0079]

[0081] 이진화기(302)에 의해 생성된 bin들의 시퀀스의 각각의 bin(303)은 순차적인 순서로 파라미터 할당기(304, parameter assigner) 내로 제공된다. 파라미터 할당기는 하나 또는 그 이상의 파라미터의 세트를 각각의 bin(303)에 할당하고 파라미터들(305)의 관련 세트를 갖는 bin을 출력한다. 파라미터들의 세트는 인코더와 디코더에서 정확하게 동일한 방법으로 결정된다. 파라미터들의 세트는 하나 또는 그 이상의 다음의 파라미터로 구성될 수 있다.

[0083] 특히, 파라미터 할당기(304)는 현재 bin(303)에 할당하도록 구성될 수 있다. 예를 들면, 파라미터 할당기(304)는 현재 bin(303)을 위한 이용가능한 콘텍스트 지수들 중 하나를 선택할 수 있다. 현재 bin(303)을 위한 콘텍스트들의 이용가능한 세트는 bin의 형태에 의존할 수 있으며, 차례로, 현재 bin(303)이 이것의 이진화의 일부분인, 구문 요소(301)의 형태/카테고리 및 후자의 이진화 내의 현재 bin(303)의 위치에 의해 정의될 수 있다. 이용가능한 콘텍스트 세트 중에서의 콘텍스트 선택은 이전의 bin들 및 후자와 관련된 구문 요소들에 의존할 수 있다. 이러한 각각의 콘텍스트는 그것과 관련된 확률 모델, 즉, 현재 bin을 위한 두 가능한 값 중 하나를 위한 확률의 추정을 위한 측정을 갖는데, 확률 모델은 부가적으로 두 가능한 bin 값 중 어느 것이 현재 bin을 위한 덜 가능성이 있거나 또는 더 가능성이 있는 bin을 표현하는지를 위한 추정을 지정하는 식별자에 의해 정의된다. 현재 bin을 위하여 단지 하나의 콘텍스트가 이용가능한 경우에 있어서, 콘텍스트 선택은 제거될 수 있다. 아래에 더 상세히 설명될 것과 같이, 파라미터 할당기(304)는 또한 각각의 콘텍스트에 속한 각각의 bin들의 실제 bin 통계에 대한 다양한 콘텍스트와 관련된 확률 모델들을 적용하기 위하여 확률 모델 적용을 실행할 수 있다.

[0085] 또한 아래에 더 상세히 설명될 것과 같이, 파라미터 할당기(304)는 활성화되는 고효율 방식 또는 낮은 복잡도

방식에 따라 다르게 작동할 수 있다. 두 방식들에서 확률 모델은 아래에 설명될 것과 같이 현재 빈(303)을 빈 인코더들(310) 중 어느 하나에 관련되나, 파라미터 할당기(304)의 작용의 방식은 낮은 복잡도 방식에서 덜 복잡한 경향이 있으며, 코딩 효율은 빈 통계에 더 정확하게 적용되도록 개별 빈들(303)의 개별 인코더들(310)로의 관련을 야기하는 파라미터 할당기(304)에 기인하여 고효율 방식에서 증가하며, 그렇게 함으로써 낮은 복잡도 방식에 대한 엔트로피를 최적화한다.

[0087] 파라미터 할당기(304)의 출력되는 파라미터들(305)의 관련 세트를 갖는 각각의 빈은 빈 버퍼 선택기(306, bin buffer selector)에 제공된다. 빈 버퍼 선택기(306)는 입력 빈 값과 관련 파라미터들(305)을 기초로 하여 입력 빈(305)의 값을 잠재적으로 변경하며 잠재적으로 변경된 값을 갖는 출력 빈(307)을 두 개 이상의 빈 버퍼들(308) 중 어느 하나 내로 제공한다. 출력 빈(307)이 보내지는 빈 버퍼(308)는 입력 빈(305)의 값 및/또는 관련 파라미터들(305)의 값을 기초로 하여 결정된다.

[0089] 본 발명의 바람직한 실시 예에서, 빈 버퍼 선택기는 빈의 값을 변경하지 않는데, 즉, 출력 빈(307)은 항상 입력 빈(305)과 동일한 값을 갖는다. 본 발명의 또 다른 바람직한 실시 예에서, 빈 버퍼 선택기(306)는 입력 빈 값 및 현재 빈을 위한 두 개의 가능한 빈 값들 중 어느 하나를 위한 확률의 추정을 위한 관련 측정을 기초로 하여 출력 빈 값(307)을 결정한다. 본 발명의 바람직한 실시 예에서, 만일 현재 빈을 위한 두 개의 가능한 빈 값들 중 어느 하나를 위한 확률의 측정이 특정 임계값보다 작으면(또는 작거나 동일하면), 출력 빈 값(307)은 입력 빈 값(305)과 동일하게 설정되고, 만일 현재 빈을 위한 두 개의 가능한 빈 값들 중 어느 하나를 위한 확률의 측정이 특정 임계값과 동일하거나 크면(또는 크면), 출력 빈 값(307)은 변경된다(즉, 입력 빈 값과 반대로 설정된다). 본 발명의 또 다른 바람직한 실시 예에서, 만일 현재 빈을 위한 두 개의 가능한 빈 값들 중 어느 하나를 위한 확률의 측정이 특정 임계값보다 크면(또는 크거나 동일하면), 출력 빈 값(307)은 입력 빈 값(305)과 동일하게 설정되고, 만일 현재 빈을 위한 두 개의 가능한 빈 값들 중 어느 하나를 위한 확률의 측정이 특정 임계값과 작거나 동일하면(또는 작으면), 출력 빈 값(307)은 변경된다(즉, 입력 빈 값과 반대로 설정된다). 본 발명의 바람직한 실시 예에서, 임계값은 두 가능한 빈 값을 위하여 추정된 확률을 위하여 0.5의 값과 상응한다.

[0091] 본 발명의 또 다른 바람직한 실시 예에서, 빈 버퍼 선택기(306)는 입력 빈 값(305) 및 두 가능한 빈 값 중 어느 하나가 현재 빈을 위하여 덜 가능성이 있거나 또는 더 가능성이 있는 빈 값을 표현하는 추정을 지정하는 관련 식별자를 기초로 하여 출력 빈 값(307)을 결정한다. 본 발명의 바람직한 실시 예에서, 만일 식별자가 두 가능한 빈 값의 제 1 세트가 현재 빈을 위한 덜 가능성 있는(또는 더 가능성 있는) 빈 값을 표현하면 출력 빈 값은 입력 빈 값(305)과 동일하게 설정되며, 만일 식별자가 두 가능한 빈 값의 제 2 세트가 현재 빈을 위한 덜 가능성 있는(또는 더 가능성 있는) 빈 값을 표현하면 출력 빈 값은 변경된다(즉, 입력 빈 값과 반대로 설정된다).

[0093] 본 발명의 바람직한 실시 예에서, 빈 버퍼 선택기(306)는 현재 빈을 위한 두 개의 가능한 빈 값 중 어느 하나를 위한 확률의 추정을 위한 관련 측정을 기초로 하여 출력 빈이 보내지는 빈 버퍼(308)를 결정한다. 본 발명의 바람직한 실시 예에서, 두 개의 가능한 빈 값 중 어느 하나를 위한 확률의 추정을 위한 관련 측정을 위한 가능한 값들의 세트는 유한하며 빈 버퍼 선택기(306)는 정확하게 하나의 빈 버퍼(308)를 두 개의 가능한 빈 값 중 어느 하나를 위한 확률의 추정을 위한 각각의 가능한 값과 관련시키는 테이블을 포함하며, 두 개의 가능한 빈 값 중 어느 하나를 위한 확률의 추정을 위한 측정을 위한 서로 다른 값들은 동일한 빈 버퍼(308)와 관련될 수 있다. 본 발명의 또 다른 바람직한 실시 예에서, 두 개의 가능한 빈 값 중 어느 하나를 위한 확률의 추정을 위한 측정을 위한 서로 다른 값들은 동일한 빈 버퍼(308)와 관련될 수 있다. 본 발명의 바람직한 실시 예에서, 두 개의 가능한 빈 값 중 어느 하나를 위한 확률의 추정을 위한 측정을 위한 반대편 측정(반대편 측정은 P 및 1-P를 추정하는 확률을 표현하는 것이다)을 갖는 입력 빈들(303)은 동일한 빈 버퍼(308) 내로 제공된다. 본 발명의 또 다른 바람직한 실시 예에서, 특정 빈 버퍼를 갖는 현재 빈을 위한 두 개의 가능한 빈 값 중 어느 하나를 위한 확률의 추정을 위한 측정의 관련성은 예를 들면, 생성된 부분적인 비트스트림들이 유사한 비트 레이트들을 갖는 것을 보장하기

위하여 시간에 따라 적용된다. 또한 아래에, 간격 지수는 또한 파이프(pipe) 지수로 불릴 것인데, 더 가능한 빈 값을 나타내는 세분화 지수 및 플래그와 함께 파이프 지수는 실제 확률 모델, 즉, 확률 추정을 색인화한다.

[0095] 본 발명의 또 다른 바람직한 실시 예에서, 빈 버퍼 선택기(306)는 현재 빈을 위한 덜 가능성이 있거나 또는 더 가능성이 있는 빈 값을 위한 확률의 추정을 위한 관련 측정을 기초로 하여 출력 빈(307)이 보내지는 빈 버퍼(308)를 결정한다. 본 발명의 바람직한 실시 예에서, 덜 가능성이 있거나 또는 더 가능성이 있는 빈 값을 위한 확률의 추정을 위한 가능한 값들의 세트는 유한하며 빈 버퍼 선택기(306)는 정확하게 하나의 빈 버퍼(308)를 덜 가능성이 있거나 또는 더 가능성이 있는 빈 값을 위한 확률의 추정의 각각의 가능한 값과 관련시키는 테이블을 포함하며, 덜 가능성이 있거나 또는 더 가능성이 있는 빈 값을 위한 확률의 추정을 위한 측정을 위한 서로 다른 값들은 동일한 빈 버퍼(308)와 관련된다. 본 발명의 또 다른 바람직한 실시 예에서, 덜 가능성이 있거나 또는 더 가능성이 있는 빈 값을 위한 확률의 추정을 위한 측정을 위한 가능한 값들의 범위는 다수의 간격으로 분할되며, 빈 버퍼 선택기(306)는 덜 가능성이 있거나 또는 더 가능성이 있는 빈 값을 위한 확률의 추정을 위한 현재 측정을 위한 간격 지수를 결정하며, 빈 버퍼 선택기(306)는 정확하게 하나의 빈 버퍼(308)를 간격 지수를 위한 각각의 가능한 값과 관련시키는 테이블을 포함하며, 간격 지수를 위한 서로 다른 값들은 동일한 빈 버퍼(308)와 관련될 수 있다. 본 발명의 또 다른 바람직한 실시 예에서, 특정 빈 버퍼를 갖는 현재 빈을 위한 덜 가능성이 있거나 또는 더 가능성이 있는 빈 값을 위한 확률의 추정을 위한 측정을 위한 관련성은 예를 들면, 생성된 부분적인 비트스트림들이 유사한 비트 레이트들을 갖는 것을 보장하기 위하여 시간에 따라 적용된다.

[0097] 두 개 또는 그 이상의 빈 버퍼들(308) 각각은 정확하게 하나의 빈 인코더(310)와 연결되며 각각의 빈 인코더는 단지 하나의 빈 버퍼(308)와 연결된다. 각각의 빈 인코더(310)는 관련된 빈 버퍼(308)로부터 빈들을 판독하며 빈들(309)의 시퀀스를 비트들의 시퀀스를 표현하는 코드워드(311) 내로 전환한다. 빈 버퍼들(308)은 선입선출(first-in-first-out) 버퍼들을 표현하며, 뒤에 빈 버퍼(308) 내로 제공되는 빈들은 빈 버퍼 내로 일찍(순차적 순서로) 제공된 빈들 전에 인코딩되지 않는다. 특정 빈 인코더(310)의 출력인 코드워드들(311)은 특정 부분 비트스트림(312)으로 판독된다. 전체 인코딩 알고리즘은 구문 요소들을 두 개 또는 그 이상의 부분적인 비트스트림(312) 내로 전환하며, 부분적인 비트스트림들의 수는 빈 버퍼들과 빈 인코더들의 수와 동일하다. 본 발명의 바람직한 실시 예에서, 빈 인코더(310)는 빈들(309)의 가변 수를 비트들의 가변 수의 코드워드(311) 내로 전환한다. 본 발명의 위에서 설명되고 아래에 설명될 실시 예들의 한가지 장점은 빈들의 인코딩이 병렬로(예를 들면, 확률 측정들의 서로 다른 그룹들을 위하여) 수행될 수 있다는 것이며, 이는 일부 구현들을 위하여 처리 시간을 감소시킨다.

[0099] 본 발명의 실시 예들의 또 다른 장점은 빈 인코더들(310)에 의해 수행되는 빈 인코딩이 파라미터들(305)의 서로 다른 세트들을 위하여 구체적으로 디자인된다는 것이다. 특히, 빈 인코딩과 인코딩은 추정된 확률들의 서로 다른 그룹들을 위하여 최적화될 수 있다(코딩 효율 및/또는 복잡도와 관련하여). 한편으로는, 이는 코딩 효율의 향상을 허용한다. 본 발명의 바람직한 실시 예에서, 빈 인코더들(310)은 현재 빈을 위한 두 개의 가능한 빈 값들(305) 중 어느 하나를 위한 확률의 추정을 위한 측정들의 서로 다른 그룹들을 위하여 서로 다른 인코딩 알고리즘들(즉, 빈 시퀀스들의 코드워드들 상으로의 매핑)을 구현한다. 본 발명의 또 다른 바람직한 실시 예에서, 빈 인코더들(310)은 현재 빈을 위한 덜 가능성이 있거나 또는 더 가능성이 있는 빈 값을 위한 확률의 추정을 위한 측정들의 서로 다른 그룹들을 위하여 서로 다른 인코딩 알고리즘들을 구현한다.

[0101] 본 발명의 바람직한 실시 예에서, 빈 인코더들(310, 또는 하나 또는 그 이상의 빈 인코더들)은 입력 빈들(309)의 시퀀스들을 코드워드들(310) 상에 직접적으로 매핑하는 엔트로피 인코더들을 표현한다. 그러한 매핑들은 효율적으로 구현되고 복잡한 산술 코딩 엔진을 필요로 하지 않는다. 빈들의 시퀀스들 상으로의 코드워드들의 역매핑은(디코더에서 수행되는 것과 같이) 입력 시퀀스의 완벽한 디코딩을 보장하기 위하여 독특해야만 하나, 코드워드들(310) 상으로의 빈 시퀀스들(309)의 매핑은 반드시 독특할 필요는 없는데, 즉, 빈들의 특정 시퀀스가 코드워드들의 하나 이상의 시퀀스 상에 매핑될 수 있는 것이 가능하다. 본 발명의 바람직한 실시 예에서, 입력 빈들(309)의 시퀀스들의 코드워드들(310) 상으로의 매핑은 전단사이다. 본 발명의 또 다른 바람직한 실시 예에서, 빈 인코더들(310, 또는 하나 또는 그 이상의 빈 인코더들)은 입력 빈들(309)의 가변 길이 시퀀스들을 가변

길이 코드워드들(310) 상에 직접적으로 매핑하는 엔트로피 인코더들을 표현한다. 본 발명의 바람직한 실시 예에서, 출력 코드워드들은 일반적인 허프만(huffman) 코드들 또는 표준(canonical) 허프만 코드들과 같은 중복 없는 코드들을 표현한다.

[0103] 빈 시퀀스들의 반복 없는 코드들로의 전단사 매핑을 위한 두 가지 예가 테이블 3에 도시된다. 본 발명의 또 다른 바람직한 실시 예에서, 출력 코드워드들은 에러 검출과 에러 복구에 적합한 반복 코드들을 표현한다. 본 발명의 또 다른 바람직한 실시 예에서, 출력 코드워드들은 구문 요소들을 암호화하기에 적합한 암호화 코드(encryption code)들을 표현한다.

[0105] 테이블 3. 빈 시퀀스들과 코드워드들 사이의 매핑들을 위한 예들.

빈들의 시퀀스 (빈 순차는 왼쪽부터 오른쪽으로이다)	코드워드들 (빈 순차는 왼쪽부터 오른쪽으로이다)
0000 0000	1
0000 0001	0000
0000 001	0001
0000 01	0010
0000 1	0011
0001	0100
001	0101
01	0110
1	0111
빈들의 시퀀스 (빈 순차는 왼쪽부터 오른쪽으로이다)	코드워드들 (빈 순차는 왼쪽부터 오른쪽으로이다)
000	10
01	11
001	010
11	011
1000 0	0001
1001	0010
1010	0011
1000 1	0000 0
1011	0000 1

[0106]

[0108] 본 발명의 또 다른 바람직한 실시 예에서, 빈 인코더들(310, 또는 하나 또는 그 이상의 빈 인코더들)은 입력 빈들(309)의 가변 길이 시퀀스들을 고정 길이 코드워드들(310) 상으로 직접적으로 매핑하는 엔트로피 인코더들을 표현한다. 본 발명의 또 다른 바람직한 실시 예에서, 빈 인코더들(310, 또는 하나 또는 그 이상의 빈 인코더들)은 입력 빈들(309)의 고정 길이 시퀀스들을 가변 길이 코드워드들(310) 상으로 직접적으로 매핑하는 엔트로피 인코더들을 표현한다.

[0110] 본 발명의 일 실시 예에 따른 디코더가 도 8에 도시된다. 디코더는 기본적으로 인코더의 역 작동들을 실행하며, 따라서 (이전에 인코딩된) 구문 요소들(327)의 시퀀스가 두 개 또는 그 이상의 부분적인 비트스트림들(324)의 세트로부터 디코딩된다. 디코더는 2가지 서로 다른 처리 흐름을 포함한다: 인코더의 데이터 흐름을 복제하는 데이터 요구(data request), 및 인코더 데이터 흐름의 역을 표현하는 데이터 흐름. 도 8의 도면에서, 선택 화살표들은 데이터 요구 흐름을 나타내며, 실선 화살표는 데이터 흐름을 나타낸다. 디코더의 빌딩 블록은 기본적으로 인코더의 데이터 블록들을 복제하나, 역 작동들을 구현한다.

- [0112] 구문 요소의 디코딩은 이진화기(314)에 보내지는 새로운 디코딩된 구문 요소(313)를 위한 요구에 의해 트리거된다. 본 발명의 바람직한 실시 예에서, 새로운 디코딩된 구문 요소(313)를 위한 각각의 요구는 하나 또는 그 이상의 카테고리의 세트의 하나의 카테고리과 관련된다. 구문 요소를 위한 요구와 관련된 카테고리는 인코딩 동안에 상응하는 구문 요소와 관련되었던 카테고리과 동일하다.
- [0114] 이진화기(314)는 구문 요소(313)를 위한 요구들을 파라미터 할당기(316)로 보내지는 빈을 위한 하나 또는 그 이상의 요구 내로 매핑한다. 이진화기(314)에 의해 파라미터 할당기(316)로 보내지는 빈을 위한 요구에 대한 최종 응답으로서, 이진화기(314)는 빈 버퍼 선택기(318)로부터 디코딩된 빈(326)을 받는다. 이진화기(314)는 디코딩된 빈들(326)의 수신된 시퀀스를 요구된 구문 요소를 위한 특정 이진화 전략과 비교하며, 만일 디코딩된 빈들(326)의 수신된 시퀀스가 구문 요소의 이진화와 일치하면, 이진화기는 그것의 빈 버퍼를 비우고 새로운 디코딩된 심벌의 요구에 대한 최종 응답으로서 디코딩된 구문 요소를 출력한다. 만일 디코딩된 빈들의 이미 수신된 시퀀스가 요구된 구문 요소를 위한 이진화 전략을 위한 빈 시퀀스들 중 어느 하나와도 일치하지 않으면, 이진화기는 디코딩된 빈들의 시퀀스가 요구된 구문 요소를 위한 이진화 전략을 위한 빈 시퀀스들 중 어느 하나와 일치할 때까지 빈을 위한 또 다른 요구를 파라미터 할당기로 보낸다. 구문 요소를 위한 각각의 요구를 위하여, 디코더는 상응하는 구문 요소를 인코딩하기 위하여 사용되었던 이진화 전략과 동일한 이진화를 사용한다. 특정 구문 요소 카테고리를 위한 이진화 전략은 가능한 구문 요소 값들의 세트 및/또는 특정 카테고리를 위한 구문 요소들의 다른 특성들에 의존할 수 있다.
- [0116] 파라미터 할당기(316)는 하나 또는 그 이상의 파라미터들의 세트를 빈을 위한 각각의 요구에 할당하고 파라미터들의 관련 세트를 갖는 빈을 위한 요구를 빈 버퍼 선택기로 보낸다. 파라미터 할당기에 의해 요구된 빈에 할당되는 파라미터들의 세트는 인코딩 동안에 상응하는 빈에 할당되었던 것과 동일하다. 파라미터들의 세트는 도 7의 인코더 설명에서 언급된 하나 또는 그 이상의 파라미터들로 구성될 수 있다.
- [0118] 본 발명의 바람직한 실시 예에서, 파라미터 할당기(316)는 빈을 위한 각각의 요구를 할당기(304)가 수행했던 것과 동일한 파라미터들, 즉, 현재 요구된 빈을 위한 덜 가능성이 있거나 또는 더 가능성이 있는 빈 값을 위한 확률의 추정을 위한 측정 및 두 가능한 빈 값 중 어느 것이 현재 요구된 빈을 위한 덜 가능성이 있거나 또는 더 가능성이 있는 빈 값을 표현하는가의 추정을 지정하는 식별자와 같이, 현재 요구된 빈을 위한 두 가능한 빈 값 중 어느 하나를 위한 확률의 추정을 위한 콘텍스트 및 그것과 관련된 측정과 관련시킨다.
- [0120] 파라미터 할당기(316)는 하나 또는 그 이상의 이미 디코딩된 심벌을 기초로 하여 하나 또는 그 이상의 위에서 언급된 확률 측정(현재 요구된 빈을 위한 두 개의 가능한 빈 값들 중 어느 하나의 확률의 추정을 위한 측정, 현재 요구된 빈을 위한 덜 가능성이 있거나 또는 더 가능성이 있는 빈 값을 위한 확률의 추정을 위한 측정, 두 가능한 빈 값들 중 어느 것이 현재 요구된 빈을 위한 덜 가능성이 있거나 또는 더 가능성이 있는 빈 값을 표현하는지의 추정을 지정하는 식별자)을 결정할 수 있다. 하나의 빈을 위한 특정 요구를 위한 확률 측정들의 결정은 상응하는 빈을 위한 인코더에서의 처리를 복제한다. 확률 측정들을 결정하기 위하여 사용되는 디코딩된 심벌들은 동일한 심벌 카테고리의 하나 또는 그 이상의 이미 디코딩된 심벌, 이웃하는 시간 및/또는 공간 위치들(구문 요소를 위한 현재 요구와 관련된 데이터 세트와 관련하여)의 데이터 세트(샘플들의 블록들 또는 그룹들과 같이)와 상응하는 동일한 심벌 카테고리의 하나 또는 그 이상의 이미 디코딩된 심벌들, 또는 동일하거나 및/또는 이웃하는 공간 및/또는 시간 위치들(구문 요소를 위한 현재 요구와 관련된 데이터 세트와 관련하여)의 데이터 세트와 상응하는 서로 다른 심벌 카테고리들의 하나 또는 그 이상의 이미 디코딩된 심벌들을 포함할 수 있다.
- [0122] 파라미터 할당기(316)의 출력인 파라미터들(317)의 관련 세트를 갖는 하나의 빈을 위한 각각의 요구는 빈 버퍼 선택기(318) 내로 제공된다. 파라미터들(317)의 관련 세트를 기초로 하여, 빈 버퍼 선택기(318)는 하나의 빈(319)을 위한 요구를 두 개 또는 그 이상의 빈 버퍼들(320) 중 어느 하나에 보내고 선택된 빈 버퍼(320)로부터 디코딩된 빈(325)을 수신한다. 디코딩된 입력 빈(325)은 잠재적으로 변경되고 디코딩된 출력 빈(326, 잠재적으로 변경된 값을 갖는)은 파라미터들(317)의 관련 세트를 갖는 하나의 빈을 위한 요구에 대한 최종 응답으로서

이진화기(314)에 보내진다.

[0124] 하나의 빈을 위한 요구가 전달되는 빈 버퍼(320)는 인코더 면에서 빈 버퍼 선택기의 출력 빈이 보내진 빈 버퍼와 동일한 방법으로 선택된다.

[0126] 본 발명의 바람직한 실시 예에서, 빈 버퍼 선택기(318)는 현재 요구된 빈을 위한 관련 측정을 기초로 하여 하나의 빈(319)을 위한 요구가 보내지는 빈 버퍼(320)를 결정한다. 본 발명의 바람직한 실시 예에서, 두 개의 가능한 빈 값 중 어느 하나를 위한 확률의 추정을 위한 측정을 위한 가능한 값들의 세트는 유한하며 빈 버퍼 선택기(318)는 정확하게 하나의 빈 버퍼(320)를 두 개의 가능한 빈 값 중 어느 하나를 위한 확률의 추정의 각각의 가능한 값과 관련시키는 테이블을 포함하며, 두 개의 가능한 빈 값 중 어느 하나를 위한 확률의 추정을 위한 측정을 위한 서로 다른 값들은 동일한 빈 버퍼(320)와 관련될 수 있다. 본 발명의 또 다른 바람직한 실시 예에서, 두 개의 가능한 빈 값 중 어느 하나를 위한 확률의 추정을 위한 측정을 위한 가능한 값들의 범위는 다수의 간격으로 할당되며, 빈 버퍼 선택기(318)는 두 개의 가능한 빈 값 중 어느 하나를 위한 확률의 추정을 위한 현재 측정을 위한 간격 지수를 결정하며, 빈 버퍼 선택기(318)는 정확하게 하나의 빈 버퍼(320)를 간격 지수를 위한 각각의 가능한 값과 관련시키는 테이블을 포함하며, 간격 지수를 위한 서로 다른 값들은 동일한 빈 버퍼(320)와 관련될 수 있다. 본 발명의 바람직한 실시 예에서, 두 개의 가능한 빈 값 중 어느 하나를 위한 확률의 추정을 위한 반대편 측정들(반대편 측정들은 확률 추정들 P 및 $1-P$ 을 표현하는 것들이다)을 갖는 빈들(317)을 위한 요구들은 동일한 빈 버퍼(320)로 전달된다. 본 발명의 또 다른 바람직한 실시 예에서, 현재 빈 요구를 위한 두 개의 가능한 빈 값 중 어느 하나를 위한 확률의 추정을 위한 측정의 특정 빈 버퍼와의 관련성은 시간에 따라 적용된다.

[0128] 본 발명의 또 다른 바람직한 실시 예에서, 빈 버퍼 선택기(318)는 현재 요구된 빈을 위한 덜 가능성이 있거나 또는 더 가능성이 있는 빈 값들의 확률의 추정을 위한 관련 측정을 기초로 하여 하나의 빈(319)을 위한 요구가 보내지는 빈 버퍼(320)를 결정한다. 본 발명의 바람직한 실시 예에서, 덜 가능성이 있거나 또는 더 가능성이 있는 빈 값들의 확률의 추정을 위한 관련 측정을 위한 가능한 값들의 세트는 유한하며 빈 버퍼 선택기(318)는 정확하게 하나의 빈 버퍼(320)를 덜 가능성이 있거나 또는 더 가능성이 있는 빈 값들의 확률의 추정의 각각의 가능한 값과 관련시키는 테이블을 포함하며, 덜 가능성이 있거나 또는 더 가능성이 있는 빈 값들의 확률의 추정을 위한 측정을 위한 서로 다른 값들은 동일한 빈 버퍼(320)와 관련될 수 있다. 본 발명의 또 다른 바람직한 실시 예에서, 덜 가능성이 있거나 또는 더 가능성이 있는 확률의 추정을 위한 측정을 위한 가능한 값들의 범위는 다수의 간격으로 분할되며, 빈 버퍼 선택기(318)는 덜 가능성이 있거나 또는 더 가능성이 있는 확률의 추정을 위한 현재 측정을 위한 간격 지수를 결정하며, 빈 버퍼 선택기(318)는 정확하게 하나의 빈 버퍼(320)를 간격 지수를 위한 각각의 가능한 값과 관련시키는 테이블을 포함하며, 간격 지수를 위한 서로 다른 값들은 동일한 빈 버퍼(320)와 관련될 수 있다. 본 발명의 또 다른 바람직한 실시 예에서, 현재 빈 요구를 위한 덜 가능성이 있거나 또는 더 가능성이 있는 확률의 추정을 위한 측정의 특정 빈 버퍼와의 관련성은 시간에 따라 적용된다.

[0130] 선택된 빈 버퍼(320)로부터 디코딩된 빈(325)을 받은 후에, 빈 버퍼 선택기(318)는 잠재적으로 입력 빈(325)을 변경하고 출력 빈(326, 잠재적으로 변경된 값을 갖는)을 이진화기(314)로 보낸다. 빈 버퍼 선택기(318)의 입력/출력 빈 매핑은 인코더 면에서의 빈 버퍼 선택기의 입력/출력 빈 매핑의 역이다.

[0132] 본 발명의 바람직한 실시 예에서, 빈 버퍼 선택기(318)는 빈의 값을 변경하지 않는데, 즉, 출력 빈(326)은 항상 입력 빈(325)과 동일한 값을 갖는다. 본 발명의 또 다른 바람직한 실시 예에서, 빈 버퍼 선택기(318)는 입력 빈 값들(325) 및 하나의 빈(317)을 위한 요구와 관련된 현재 요구된 빈을 위한 두 개의 가능한 빈 값 중 어느 하나를 위한 확률의 추정을 위한 측정을 기초로 하여 출력 빈 값(326)을 결정한다. 본 발명의 바람직한 실시 예에서, 출력 빈 값(326)은 만일 현재 요구된 빈을 위한 두 개의 가능한 빈 값 중 어느 하나를 위한 확률을 위한 측정이 특정 임계값보다 작으면(또는 작거나 동일하면) 입력 빈 값(325)과 동일하게 설정되며 만일 현재 요구된 빈을 위한 두 개의 가능한 빈 값 중 어느 하나를 위한 확률을 위한 측정이 특정 임계값보다 크거나 동일하면(또는 크면), 출력 빈 값(326)은 변경된다(즉, 입력 빈 값과 반대로 설정된다). 본 발명의 또 다른 바람직한

실시 예에서, 출력 빈 값(326)은 만일 현재 요구된 빈을 위한 두 개의 가능한 빈 값 중 어느 하나를 위한 확률을 위한 측정이 특정 임계값보다 크면(또는 크거나 동일하면) 입력 빈 값(325)과 동일하게 설정되며 만일 현재 요구된 빈을 위한 두 개의 가능한 빈 값 중 어느 하나를 위한 확률을 위한 측정이 특정 임계값보다 작거나 동일하면(또는 작으면), 출력 빈 값(326)은 변경된다(즉, 입력 빈 값과 반대로 설정된다). 본 발명의 바람직한 실시 예에서, 임계값은 두 가능한 빈 값을 위한 추정 확률을 위한 0.5의 값과 상응한다.

[0134] 본 발명의 또 다른 바람직한 실시 예에서, 빈 버퍼 선택기(318)는 입력 빈 값(325) 및 두 가능한 빈 값들 중 어느 것이 빈(317)을 위한 요구와 관련된, 현재 빈 요구를 위한 덜 가능성이 있거나 또는 더 가능성이 있는 빈 값을 표현하는지를 추정을 지정하는 식별자를 기초로 하여 출력 빈 값(326)을 결정한다. 본 발명의 바람직한 실시 예에서, 만일 식별자가 두 가능한 빈 값 중 첫 번째가 현재 빈 요구를 위한 덜 가능성이 있는(또는 더 가능성이 있는) 빈 값을 표현하는 것으로 지정하면 출력 빈 값(326)은 입력 빈 값(325)과 동일하게 설정되며, 만일 식별자가 두 가능한 빈 값 중 두 번째가 현재 빈 요구를 위한 덜 가능성이 있는(또는 더 가능성이 있는) 빈 값을 표현하는 것으로 지정하면 출력 빈 값(326)은 변경된다(즉, 이는 입력 빈 값과 반대로 설정된다).

[0136] 위에서 설명된 것과 같이, 빈 버퍼 선택기는 하나의 빈(319)을 위한 요구를 두 개 또는 그 이상의 빈 버퍼들(320) 중 어느 하나로 보낸다. 빈 버퍼들(320)은 연결된 빈 디코더들(322)로부터 디코딩된 빈들(321)의 시퀀스들이 제공되는, 선입선출 버퍼들을 표현한다. 빈 버퍼 선택기(318)로부터 빈 버퍼(320)에 보내지는 하나의 빈(309)을 위한 요구에 대한 응답으로서, 빈 버퍼(320)는 빈 버퍼(320) 내로 우선 제공된 그것의 콘텐츠의 빈을 제거하고 이를 빈 버퍼 선택기(318)로 보낸다. 더 일찍 빈 버퍼(320)로 보내진 빈들은 더 일찍 제거되고 빈 버퍼 선택기(318)로 보내진다.

[0138] 두 개 또는 그 이상의 버퍼들(320) 각각은 정확하게 하나의 빈 디코더(322)와 연결되고 각각의 빈 디코더는 단지 하나의 빈 버퍼(320)에 연결된다. 각각의 빈 디코더(322)는 개별의 부분적인 비트스트림(324)로부터 비트들의 시퀀스들을 표현하는 코드워드들(323)을 판독한다. 빈 디코더는 코드워드들(323)을 연결된 빈 버퍼(320)로 보내지는 빈들(321)의 시퀀스 내로 전환한다. 전체 인코딩 알고리즘은 두 개 또는 그 이상의 부분적인 비트스트림(312)을 다수의 디코딩된 구문 요소들 내로 전환하며, 부분적인 비트스트림들의 수는 빈 버퍼들과 빈 인코더들의 수와 동일하며 구문 요소들의 디코딩은 새로운 구문 요소들을 위한 요구들에 의해 트리거된다. 본 발명의 바람직한 실시 예에서, 빈 디코더(322)는 비트들의 가변 수의 코드워드들(323)을 빈들(321)의 가변 수의 시퀀스로 전환한다. 본 발명의 실시 예들의 한가지 장점은 두 개 또는 그 이상의 부분적인 비트스트림들로부터 빈들의 디코딩이 병렬로(예를 들면, 확률 측정들의 서로 다른 그룹들을 위하여) 수행될 수 있다는 것이며, 이는 일부 구현들을 위하여 처리 시간을 감소시킨다.

[0140] 본 발명의 실시 예들의 또 다른 장점은 빈 디코더들(322)에 의해 수행되는 빈 디코딩이 파라미터들(317)의 서로 다른 세트들을 위하여 구체적으로 디자인될 수 있다는 것이다. 특히, 빈 인코딩과 인코딩은 추정된 확률들의 서로 다른 그룹들을 위하여 최적화될 수 있다(코딩 효율 및/또는 복잡도와 관련하여). 한편으로는, 이는 유사한 인코딩/디코딩 복잡도를 갖는 최신 엔트로피 코딩 알고리즘들에 대하여 코딩 효율의 향상을 허용한다. 본 발명의 바람직한 실시 예에서, 빈 디코더들(322)은 현재 빈 요구를 위한 두 개의 가능한 빈 값(317) 중 어느 하나를 위한 확률의 추정을 위한 측정들의 서로 다른 그룹들을 위하여 서로 다른 디코딩 알고리즘들(즉, 빈 시퀀스들의 코드워드들 상으로의 매핑)을 구현한다. 본 발명의 또 다른 바람직한 실시 예에서, 빈 디코더들(322)은 현재 요구된 빈을 위한 덜 가능성이 있거나 또는 더 가능성이 있는 빈 값을 위한 확률의 추정을 위한 측정들의 서로 다른 그룹들을 위하여 서로 다른 디코딩 알고리즘들을 구현한다.

[0142] 빈 디코더들(322)은 인코더 면에서 상응하는 빈 인코더들의 역 매핑을 한다.

[0144] 본 발명의 바람직한 실시 예에서, 빈 디코더들(322, 또는 하나 또는 그 이상의 빈 디코더)은 코드워드들(323)을 빈들(321)이 시퀀스들 상에 직접적으로 매핑하는 엔트로피 디코더들을 표현한다. 그러한 매핑은 효율적으로 구

현될 수 있으며 복잡한 산술 코딩 엔진을 요구하지 않는다. 빈들의 시퀀스들 상으로의 코드워드들의 매핑은 독특해야만 한다. 본 발명의 바람직한 실시 예에서, 빈들(321)의 시퀀스들 상으로의 코드워드들(323)의 매핑은 전단사이다. 본 발명의 바람직한 실시 예에서, 빈 디코더들(322, 또는 하나 또는 그 이상의 디코더)은 가변 길이 코드워드들(323)을 빈들(321)의 가변 길이 시퀀스들 내로 직접적으로 매핑하는 엔트로피 디코더들을 표현한다. 본 발명의 바람직한 실시 예에서, 입력 코드워드들은 일반적인 허프만 코드들 또는 표준 허프만 코드들과 같은 중복 없는 코드들을 표현한다. 빈 시퀀스들의 반복 없는 코드들로의 전단사 매핑을 위한 두 가지 예가 테이블 3에 도시된다.

[0146] 본 발명의 또 다른 바람직한 실시 예에서, 빈 디코더들(322, 또는 하나 또는 그 이상의 빈 디코더들)은 고정 길이 코드워드들(323)을 빈들(321)의 가변 길이 시퀀스들 상으로 직접적으로 매핑하는 엔트로피 디코더들을 표현한다. 본 발명의 또 다른 바람직한 실시 예에서, 빈 디코더들(322, 또는 하나 또는 그 이상의 빈 디코더들)은 가변 길이 코드워드들(323)을 빈들(321)의 고정 길이 시퀀스들 상으로 직접적으로 매핑하는 엔트로피 디코더들을 표현한다.

[0148] 따라서, 도 7과 8은 심벌들(3)의 시퀀스를 인코딩하기 위한 인코더 및 이를 재구성하기 위한 디코더를 위한 실시 예를 도시한다. 인코더는 다수의 파라미터들(305)을 심벌들의 시퀀스의 각각의 심벌에 할당하도록 구성되는 할당기(304)를 포함한다. 할당은 현재 심벌들이 속하고 구문 요소들(1)의 구문 구조에 따라, 현재 기대되며 차례로, 기대가 이전의 구문 요소들(1)과 심벌들(3)로부터 추론할 수 있는 표현(이진화와 같은)에 대한 구문 요소(1)의 카테고리들과 같이 심벌들의 시퀀스의 이전의 심벌들 내에 포함된 정보를 기초로 한다. 또한, 인코더는 각각 각각의 엔트로피 인코더에 전달된 심벌들(3)을 각각의 비트스트림 내로 전환하도록 구성되는 복수의 엔트로피 인코더(10), 및 각각의 심벌(3)을 복수의 엔트로피 인코더(10) 중 선택된 하나에 전달하도록 구성되는 선택기(306)를 포함하며, 선택은 각각의 심벌에 할당되는 파라미터들(305)의 수에 따른다. 할당기(304)는 각각의 선택기(502)를 생산하기 위하여 선택기(206) 내에 통합되는 것으로서 생각될 수 있다.

[0150] 심벌들의 시퀀스를 재구성하기 위한 디코더는 각각 각각의 비트스트림(323)을 심벌들(321) 내로 전환하도록 구성되는, 복수의 엔트로피 디코더(322); 다수의 파라미터들(317)을 심벌들의 시퀀스(도 8의 326 및 327 참조)의 이전에 재구성된 심벌들 내에 포함된 정보를 기초로 하여 재구성되는 심벌들의 시퀀스의 각각의 심벌(315)에 할당하도록 구성되는 할당기(316); 및 복수의 엔트로피 디코더(322) 중 선택된 하나로부터 재구성되는 심벌들의 시퀀스의 각각의 심벌을 검색하도록 구성되는 선택기(318);를 포함하는데, 선택은 각각의 심벌에 정의되는 파라미터들의 수에 따른다. 할당기(316)는 각각의 심벌에 할당되는 파라미터들의 수가 각각의 심벌이 추정할 수 있는 가능한 심벌 값들 중에서의 분포의 확률의 추정을 위한 측정을 포함하는 것과 같이 구성될 수 있다. 다시, 할당기(316) 및 선택기(318)는 하나의 블록, 선택기(402) 내로 통합되는 것으로 생각될 수 있다. 재구성되는 심벌들의 시퀀스는 이진 알파벳일 수 있으며 할당기(316)는 확률 분포의 추정이 이진 알파벳의 두 가능한 빈 값들 중 덜 가능성이 있거나 또는 더 가능성이 있는 빈 값의 확률의 추정을 위한 측정으로 구성될 수 있으며, 두 가능한 빈 값 중 어느 하나를 위한 추정을 지정하는 식별자는 덜 가능성이 있거나 또는 더 가능성이 있는 빈 값을 표현한다. 할당기(316)는 또한 그것과 관련된 각각의 확률 분포 추정을 갖는 각각의 컨텍스트와 함께 재구성되는 심벌들의 시퀀스의 이전에 재구성된 심벌들 내에 포함되는 정보를 기초로 하여 컨텍스트를 재구성되는 심벌들(315)의 시퀀스의 각각의 심벌에 내부로 할당하도록 구성될 수 있으며, 각각의 컨텍스트가 할당되는 이전에 재구성된 심벌들의 심벌 값들을 기초로 하여 각각의 컨텍스트를 위한 확률 분포 추정을 실제 심벌 통계에 적용하도록 구성될 수 있다. 컨텍스트는 비디오 또는 픽처 코딩에서, 또는 금융 적용들의 경우에서의 테이블들에서와 같이 구문 요소들이 속하는 위치들의 공간 관계 또는 이웃을 고려할 수 있다. 그리고 나서, 각각의 심벌을 위한 확률 분포의 추정을 위한 측정은 확률 분포의 추정을 위한 측정(부분적인 비트스트림(312)을 지수화하는 파이프 지수)을 획득하기 위하여 복수의 확률 분포 추정 대표 중 어느 하나(세분화 지수를 잘라냄)에 대하여 각각의 심벌(세분화 지수와 함께 파이프 지수에 의해 지수화되는 아래의 실시 예들에서)로 할당된 컨텍스트와 관련된 확률 분포 추정을 양자화하거나, 또는 각각의 테이블 내의 지수로서 이를 사용하는 것과 같이 각각의 심벌에 할당된 컨텍스트와 관련된 확률 분포 추정을 기초로 하여 결정될 수 있다. 선택기는 복수의 엔트로피 디코더와 복수의 확률 분포 추정 대표 사이에 전단사 관련성이 정의되는 것과 같이 구성될 수 있다. 선택기(18)는 시간에 따라, 심벌들의 시퀀스의 이전에 재구성된 심벌들에 따라 미리 결정된 결정론적 방법으로 확률 분포 추정

들의 범위로부터 복수의 확률 분포 추정 대표로 양자화 매핑을 변경하도록 구성될 수 있다. 즉, 선택기(318)는 양자화 단계 크기들, 즉, 개별 엔트로피 디코더들과 전단사로 관련된 개별 확률 지수들 상으로 매핑된 확률 분포들의 간격들을 변경할 수 있다. 복수의 엔트로피 디코더(322)는 차례로, 양자화 매핑의 변경에 응답하여 심벌들의 비트스트림들 내로의 그것의 전환하는 방법을 적용하도록 구성될 수 있다. 예를 들면, 각각의 엔트로피 디코더(322)는 각각의 확률 분포 추정 양자화 간격 내의 특정 확률 분포 추정을 위하여 최적화될 수 있는데, 즉, 각각의 확률 분포 추정 양자화 간격 내의 특정 확률 분포 추정을 위한 최적 압축률을 가질 수 있으며, 최적화되기 위하여 후자의 변경 상에 각각의 확률 분포 추정 양자화 간격 내의 이러한 특정 확률 분포 추정의 위치를 적용하도록 하기 위하여 그것의 코드워드/심벌 시퀀스 매핑을 변경할 수 있다. 선택기는 심벌들이 복수의 엔트로피 디코더로부터 검색되는 비율이 덜 분산되는 것과 같이 양자화 매핑을 변경하도록 구성될 수 있다. 이진화기(314)와 관련하여, 만일 구문 요소가 이미 이진수이면 이는 제거될 수 있다는 것을 이해하여야 한다. 또한, 디코더(322)의 형태에 따라, 버퍼들(320)의 존재가 반드시 필요하지는 않다. 또한, 버퍼들은 디코더들 내에 통합될 수 있다.

[0152] 유한 구문 요소 시퀀스들의 종료

[0154] 본 발명의 바람직한 실시 예에서, 인코딩과 디코딩은 구문 요소들의 유한 세트를 위하여 수행된다. 흔히, 비디오 시퀀스의 스틸 이미지, 프레임 또는 필드, 이미지의 조각(slice), 비디오 시퀀스의 프레임 또는 필드의 조각, 혹은 연속적인 오디오 샘플들의 세트 등과 같은 데이터의 특정 양이 코딩된다. 구문 요소들이 유한 세트를 위하여, 일반적으로, 인코더 면에서 생성되는 부분적인 비트스트림들은 종료되어야만 하는데, 즉, 모든 구문 요소는 전송되거나 또는 저장된 부분적인 비트스트림들로부터 디코딩될 수 있는 것이 보장되어야만 한다. 마지막 빈이 상응하는 빈 버퍼(308) 내로 삽입된 후에, 빈 인코더(310)는 완전한 코드워드가 부분적인 비트스트림(312)에 관독되는 것을 보장해야만 한다. 만일 빈 인코더(310)가 빈 시퀀스들의 코드워드들 상으로의 직접적인 매핑을 구현하는 엔트로피 인코더를 표현하면, 마지막 빈을 빈 버퍼 내로 관독한 후에 빈 버퍼 내에 저장된 빈 시퀀스는 코드워드와 관련된 빈 시퀀스를 표현하지 않을 수 있다(즉, 코드워드들과 관련된 두 개 또는 그 이상의 빈 시퀀스들의 접두부호(prefix)를 표현할 수 있다). 이는 코드워드가 관독될 때까지 특정 또는 임의의 값을 갖는 빈들을 빈 버퍼 내로 삽입함으로써 수행될 수 있다. 본 발명의 바람직한 실시 예에서, 빈 인코더는 최소 길이를 갖는 코드워드들 중 어느 하나를 선택한다(관련 빈 시퀀스는 접두부호로서 빈 버퍼 내의 빈 시퀀스를 포함해야만 한다는 특성에 더하여). 디코더 면에서, 빈 디코더(322)는 부분적인 비트스트림 내의 마지막 코드워드를 위하여 필요한 것보다 더 많은 빈들을 디코딩할 수 있는데, 이러한 빈들은 빈 버퍼 선택기(118)에 의해 요구되지 않으며 버려지고 무시된다. 심벌들의 유한 세트의 디코딩은 디코딩된 구문 요소들을 위한 요구들에 의해 제어된다. 만일 데이터의 양을 위하여 또 다른 구문 요소가 요구되지 않으면, 디코딩은 종료된다.

[0156] 부분적인 비트스트림들의 전송과 멀티플렉스

[0158] 인코더에 의해 생성되는 부분적인 비트스트림들(312)은 개별적으로 전송될 수 있거나 또는 단일 비트스트림 내로 멀티플렉스될 수 있거나, 또는 부분적인 비트스트림들의 코드워드들은 단일 비트스트림 내로 끼워질 수 있다.

[0160] 본 발명의 일 실시 예에서, 데이터의 양을 위한 각각의 부분적인 비트스트림은 하나의 데이터 패킷에 관독된다. 데이터의 양은 비디오 시퀀스의 스틸 픽처, 필드 또는 프레임, 스틸 픽처의 조각, 비디오 시퀀스의 필드 또는 프레임의 조각, 혹은 오디오 샘플들의 프레임 등과 같은 구문 요소들의 임의의 세트일 수 있다.

[0162] 본 발명의 또 다른 바람직한 실시 예에서, 데이터 양을 위한 두 개 또는 그 이상의 부분적인 비트스트림들 또는 데이터의 양을 위한 모든 부분적인 비트스트림은 데이터 패킷 내로 멀티플렉싱된다. 멀티플렉싱된 부분적인 비트스트림들을 포함하는 데이터 패킷의 구조가 도 9에 도시된다.

[0164] 데이터 패킷(400)은 각각의 부분적인 비트스트림의 데이터를 위한(데이터의 고려되는 양을 위한) 헤더(header) 및 하나의 파티션으로 구성된다. 데이터의 패킷의 헤더(400)는 데이터 패킷(의 나머지)의 비트스트림 데이터(402)의 세그먼트들 내로의 분할을 위한 표시들을 포함한다. 분할을 위한 표시들 외에, 헤더는 부가 정보를 포함할 수 있다. 본 발명의 바람직한 실시 예에서, 데이터 패킷의 분할을 위한 표시들은 비트들, 또는 바이트들 또는 비트들의 배수 또는 바이트들의 배수의 유닛들 내의 데이터 세그먼트의 시작의 위치들이다. 본 발명의 바람직한 실시 예에서, 데이터 세그먼트들의 시작의 위치들은 데이터 패킷의 시작에 관한, 또는 헤더의 단부에 관하거나 혹은 이전 데이터 패킷의 시작에 관한, 데이터 패킷의 헤더 내의 절대값으로서 코딩된다. 본 발명의 또 다른 바람직한 실시 예에서, 데이터 세그먼트들의 시작의 위치들은 서로 다르게 코딩되는데, 즉, 데이터 세그먼트의 실제 시작 및 데이터 세그먼트의 시작을 위한 사이의 차이만이 코딩된다. 예측은 데이터 패킷의 전체 크기, 헤더의 크기, 데이터 패킷 내의 데이터 세그먼트의 수, 선행 데이터 세그먼트들의 시작의 위치와 같은 이미 알려지거나 또는 전송된 정보를 기초로 하여 유래될 수 있다. 본 발명의 바람직한 실시 예에서, 첫 번째 데이터 패킷의 시작의 위치는 코딩되지 않고, 데이터 패킷 헤더의 크기를 기초로 하여 추론된다. 디코더 면에서, 전송된 분할 표시들은 데이터 세그먼트들의 시작을 유래하기 위하여 사용된다. 데이터 세그먼트들은 그리고 나서 부분 비트스트림들로서 사용되고 데이터 세그먼트들 내에 포함된 데이터는 순차적인 순서로 상응하는 빈 디코더들 내로 제공된다.

[0166] 부분 비트스트림들을 데이터 패킷 내로 멀티플렉싱하기 위한 일부 대안들이 존재한다. 특히 부분 비트스트림들의 크기가 매우 유사한 경우를 위하여, 필요한 부가 정보를 감소시킬 수 있는, 하나의 대안이 도 10에 도시된다. 데이터 패킷의 페이로드, 즉, 그것의 헤더(411)가 없는 데이터 패킷(410)은 미리 정의된 방법으로 세그먼트들(412) 내로 분할된다. 예로서, 데이터 패킷 페이로드는 동일한 크기의 세그먼트들 내로 분할될 수 있다. 그때 각각의 세그먼트는 부분 비트스트림 또는 부분 비트스트림(413)의 첫 번째 부분과 관련된다. 만일, 부분 비트스트림이 관련 데이터 세그먼트보다 크면, 그것의 나머지(414)는 다른 데이터 세그먼트들의 단부에서 사용되지 않은 공간 내로 위치된다. 이는 비트스트림의 나머지 부분이 부가 정보를 감소시키는, 역 순서로(데이터 세그먼트의 단부에서 시작하여) 삽입되는 방법으로 수행될 수 있다. 부분 비트스트림들의 나머지들의 데이터 세그먼트들과의 관련성 및, 하나 이상의 나머지가 데이터 세그먼트에 추가될 때, 하나 또는 그 이상의 나머지를 위한 시작 부는 비트스트림 내부로, 즉, 데이터 패킷 헤더 내로 신호가 전달되어야만 한다.

[0168] 가변 길이 코드워드들의 인터리빙(interleaving)

[0170] 일부 적용들을 위하여, 위에서 설명된 하나의 데이터 패킷 내의 부분 비트스트림들의 멀티플렉싱(구문 요소들의 양을 위한)은 다음의 단점들을 가질 수 있다: 한편으로는 작은 데이터 패킷들을 위하여, 분할을 신호전달하는 부가 정보를 위한 비트들의 수는 부분 비트스트림들 내의 실제 데이터에 대하여 중요할 수 있는데, 이는 결국은 코딩 효율을 감소시킨다. 다른 한편으로는, 멀티플렉싱은 저지연을 필요로 하는 적용들에(예를 들면, 비디오 회의 적용들에) 적합하지 않을 수 있다. 설명된 멀티플렉싱과 함께, 인코더는 부분 비트스트림들이 완전히 생성되기 전에 데이터 패킷의 전송을 시작할 수 없는데, 그 이유는 분할들의 시작이 위치들이 전에 알려지지 않기 때문이다. 게다가, 일반적으로, 디코더는 그것이 데이터 패킷의 디코딩을 시작할 수 있기 전에 마지막 데이터 세그먼트의 시작을 받을 때까지 기다려야만 한다. 비디오 회의 시스템들과 같은 적용들을 위하여, 이러한 지연들은 그러한 적용들을 위하여 중요한, 일부 비디오 픽처들의 시스템의 부가적인 전체 지연에 더해질 수 있다(특히 전송 비트 레이트에 가까운 비트 레이트들을 위하여, 그리고 픽처를 인코딩/디코딩하기 위하여 두 픽처 사이의 시간 간격을 거의 필요로 하는 인코더들/디코더들을 위하여). 특정 적용들을 위한 단점들을 극복하기 위하여, 본 발명의 바람직한 실시 예의 인코더는 둘 또는 그 이상의 빈에 의해 발생하는 코드워드들이 단일 비트스트림 내로 인터리브되는 것과 같은 방법으로 구성될 수 있다. 인터리브된 코드워드들을 갖는 비트스트림은 디코더에 직접적으로 보내질 수 있다(작은 버퍼 지연을 무시할 때, 아래 참조). 디코더 면에서, 둘 또는 그 이상의 빈 디코더들은 디코딩 순서로 비트스트림으로부터 직접적으로 코드워드들을 판독한다; 디코딩은 첫 번째 받은 비트로 시작될 수 있다. 게다가, 부분 비트스트림들의 멀티플렉싱(또는 인터리빙)을 신호전달하는데 어떠한 부가 정보도 필요로 하지 않는다. 디코더 복잡도를 감소시키는 또 다른 방법은 빈 디코더들(322)이 글로벌 비트 버퍼로부터 가변-길이 코드워드들을 판독하지 않을 때 달성될 수 있으나, 대신에 그것들은 항상 글로벌 비트 버퍼로부터

비트들의 고정-길이 시퀀스들을 판독하고 비트들의 이러한 고정 길이 시퀀스들을 로컬 비트 버퍼에 추가하며, 각각의 빈 디코더(322)는 분리된 로컬 비트 버퍼와 연결된다. 가변-길이 코드워드들은 그리고 나서 로컬 비트 버퍼로부터 판독된다. 따라서, 가변-길이 코드워드들의 파싱(parsing)은 병렬로 수행될 수 있으며, 비트들의 고정-길이 시퀀스들의 액세스만이 동기화된 방법으로 수행되어야만 하나, 그러한 비트들의 고정-길이 시퀀스들의 액세스는 일반적으로 매우 빠르며, 따라서 일부 아키텍처(architecture)들을 위하여 전체 디코딩 복잡도가 감소될 수 있다. 특정 로컬 비트 버퍼로 보내지는 빈들의 고정된 수는 서로 다른 로컬 비트 버퍼를 위하여 다를 수 있으며 이는 또한 빈 디코더, 빈 버퍼, 또는 비트 버퍼 내의 이벤트들과 같은 특정 파라미터들에 따라, 시간에 따라 변할 수 있다. 그러나, 특정 액세스에 의해 판독되는 비트들의 수는 가변-길이 코드워드들의 판독에 대한 중요한 차이인, 특정 액세스 동안에 판독되는 실제 비트들에 의존하지 않는다. 비트들의 고정-길이 시퀀스들의 판독은 빈 버퍼들, 빈 디코더들, 또는 로컬 빈 버퍼들 내의 특정 이벤트들에 의해 트리거된다. 예로서, 연결된 빈 버퍼에 존재하는 비트들의 수가 미리 정의된 임계값 아래에 포함될 때 비트들의 새로운 고정-길이 시퀀스의 판독을 요구하는 것이 가능하며, 서로 다른 비트 버퍼들을 위하여 서로 다른 임계값들이 사용될 수 있다. 인코더에서, 빈들의 고정-길이 시퀀스들이 동일한 순서로 디코더 면에서 비트스트림으로부터 그것들이 판독되는, 비트스트림 내로 삽입되는 것이 보장되어야만 한다. 또한 고정-길이 시퀀스들의 이러한 인터리빙을 위에서 설명된 것들과 유사한 저지연 제어와 결합하는 것이 가능하다. 다음에서, 비트들의 고정 길이 시퀀스들의 인터리빙을 위한 바람직한 실시 예가 설명된다. 후자의 인터리빙 전략들에 대한 더 상세설명을 위하여, 국제특허 WO2011/128268A1이 참조된다.

[0172] 이전의 코딩도 비디오 데이터를 압축하기 위하여 사용되는 실시 예들을 설명한 후에, 한편으로는 압축률과 다른 한편으로는 록-업 테이블과 계산 오버헤드 사이의 뛰어난 균형과 관련하여 특히 효율적인 구현을 제공하는 본 발명의 실시 예들을 구현하기 위한 또 다른 실시 예가 설명된다. 특히, 다음의 실시 예들은 개별적으로 비트스트림들을 엔트로피 코딩하고 확률 추정의 부들을 효율적으로 포함하도록 하기 위하여 계산적으로 덜 복잡한 가변 길이 코드들의 사용을 가능하게 한다. 아래에 설명되는 실시 예에서, 심벌들은 이진 유형이고 아래에 나타내는 가변 길이 코드 코드들은 예를 들면, $[0;0.5]$ 내에서 확장하는 R_{PS} 에 의해 표현되는 확률 추정을 포함한다.

[0174] 특히, 아래에 설명되는 다음의 실시 예들은 각각 도 7 내지 17에서의 개별 엔트로피 코더들(310)과 디코더들(322)을 위한 가능한 구현들을 설명한다. 그것들은 빈들, 즉 이진 심벌들의 코딩에 적합한데, 그 이유는 그것들이 이미지 또는 비디오 압축 적용들에서 발생하기 때문이다. 따라서, 이러한 실시 예들은 또한 그러한 이진 심벌들이 각각 인코딩되는 빈들(307) 및 디코딩되는 비트스트림들(324)의 의 하나 또는 그 이상의 스트림 내로 분할되는 이미지 또는 비디오 코딩에 적용가능하며, 각각의 그러한 빈 스트림은 베르누이(Bernoulli) 과정의 실현으로서 고려될 수 있다. 아래에 설명되는 실시 예들은 빈 스트림들을 인코딩하기 위하여 하나 또는 그 이상의 아래에 설명되는 다양한 이른바 가변-대-가변-코드들(v2v-코드들)을 사용한다. 가변-대-가변-코드는 동일한 수의 코드 워드들을 갖는 두 무-접두부호 코드, 일차, 및 이차 무-접두부호(prefix-free) 코드로서 고려될 수 있다. 아래에 설명되는 실시 예들에 따라, 인코더들(310)과 디코더들(322)의 적어도 일부는 다음과 같이 운용된다: 빈들(307)의 특정 시퀀스를 인코딩하기 위하여, 일차 무-접두부호 코드의 코드 워드가 버퍼(308)로부터 판독될 때마다, 이차 무-접두부호 코드의 상응하는 코드 워드가 비트 스트림(312)에 판독된다. 그러한 비트 스트림(324)을 디코딩하기 위하여 동일한 과정이 사용된다. 일차 및 이차 무-접두부호 코드에서는 교환된다. 즉, 비트스트림(324)을 디코딩하기 위하여, 이차 무-접두부호코드의 코드 워드가 각각의 비트 스트림(324)으로부터 판독될 때마다, 일차 무-접두부호 코드의 상응하는 코드-워드가 버퍼(320)에 판독된다.

[0176] 바람직하게는, 아래에 설명되는 코드들은 록-업 테이블을 필요로 하지 않는다. 코드들은 유한 상태 기계들 형태로 구현될 수 있다. 여기에 나타내는 가변-대-가변-코드들은 코드 워드들을 위한 큰 테이블을 저장할 필요가 없는 것과 같은 간단한 구성 규칙들에 의해 발생될 수 있다. 대신에, 인코딩 또는 디코딩을 수행하기 위하여 간단한 알고리즘이 사용될 수 있다. 이러한 구성 규칙들이 아래에 설명되며 그러한 구성 규칙들 중 2가지가 파라미터화될 수 있다. 그것들은 앞서 언급된 확률 간격의 차이 또는 서로소(disjoint) 부들을 포함하며, 따라서, 병렬의 모든 3 코드(인코더들(11)/디코더들(22)의 서로 다른 것들을 위한 각각), 또는 그것들 중 2개)와 같이, 만약 함께 사용하면 특히 바람직하다. 아래에 설명되는 구성 규칙들로, 임의의 확률(p)을 갖는 베르누이 과정들을 위하여, 코드들 중 하나가 초과 코드 길이에 대하여 잘 실행하는 것과 같이, 가변-대-가변-코드들의 세트를

디자인하는 것이 가능하다.

[0178] 위에서 설명된 것과 같이, 각각 스트림들(312 및 324)의 인코딩과 디코딩은 각각의 스트림을 위하여 독립적으로 또는 인터리브드(interleaved) 방식으로 실행될 수 있다. 그러나, 가변-대-가변-코드들의 존재하는 클래스들에 특정하지 않으며, 따라서 다음에서의 각각의 3가지 구성을 위하여 특정 코드워드의 인코딩과 디코딩만이 설명된다. 그러나, 인터리빙 솔루션들에 대한 위의 모든 실시 예들은 또한 각각 현재 설명되는 코드들 또는 인코더들(310)과 디코더들(322)과 결합가능하다.

[0180] 구성 규칙 1: '단항 빈 파이프' 코드들 또는 인코더/디코더들(310 및 322)

[0182] 단항 빈 파이프 코드들(파이프 = 확률 간격 분할 엔트로피(probability interval partitioning entropy, PIPE, 이하 파이프로 표기)은 이른바 '빈 파이프' 코드들, 즉, 각각 앞서 언급된 확률 범위 [0;0.5]의 특정한 확률 서브-간격에 속하는 이진 심벌 통계의 데이터를 전달하는, 개별 비트스트림들(12 및 24) 중 어느 하나의 코딩에 적합한 코드들의 특별한 버전이다. 빈 파이프 코드들이 구성이 먼저 설명된다. 빈 파이프 코드는 적어도 3개의 코드 워드들을 갖는 어떠한 접두부호 없는 코드로부터 구성될 수 있다. 가변-대-가변 코드를 형성하기 위하여 이는 일차 및 이차 코드로서 접두부호 없는 코드를 사용하나, 이차 접두부호 없는 코드의 두 코드워드들은 상호 교환된다. 이는 두 코드 워드들을 제외하고는, 빈들이 변경되지 않는 접두부호 없는 코드로 판독된다는 것을 의미한다. 이러한 기술로, 하나의 접두부호 없는 코드만이 정보와 함께 저장하는데 필요한데, 두 코드 워드들은 상호교환되며 따라서, 메모리 소비가 감소된다. 이는 서로 다른 길이의 코드 워드들을 교환하는데만 의미가 있는데 그 이유는 그렇지 않으면, 비트 스트림이 비트 스트림(비트 스트림의 단부에서 발생할 수 있는 효과들을 무시하는)과 동일한 길이를 가질 수 있기 때문이다.

[0184] 이러한 구성 규칙 때문에, 빈 파이프 코드들의 두드러진 특성은, 만일 일차 및 이차 접두부호 없는 코드가 상호 교환되면(코드 워드들의 매핑이 유지되는 동안에), 결과로서 생기는 가변-대-가변-코드는 오리지널 가변-대-가변-코드와 동일하다는 것이다. 따라서, 인코딩 알고리즘과 디코딩 알고리즘은 빈-파이프 코드들을 위하여 동일하다.

[0186] 단항 빈 파이프 코드는 특별한 접두부호 없는 코드로부터 구성된다. 이러한 특별한 접두부호 없는 코드는 다음과 같이 구성된다: 우선, n 단항 코드 워드들로 구성되는 접두부호 없는 코드가 '01', '001', '0001'로 시작하여 n 코드워드가 생산될 때까지 발생되는데, n은 단항 빈 파이프 코드를 위한 파라미터이다. 가장 긴 코드 워드로부터, 트레일링(trailing) 1이 제거된다. 이는 절단된 단항 코드(그러나 코드 워드 '0'이 없는)와 상응한다. 그리고 나서, n-1 단항 코드 워드들이 '10', '110', '1110'으로 시작하여 n-1 코드워드가 생산될 때까지 발생된다. 이러한 가장 긴 코드 워드로부터, 트레일링 0이 제거된다. 단항 빈 파이프 코드를 발생시키기 위하여 이러한 두 접두부호 없는 코드들의 조합 세트가 입력으로서 사용된다. 상호교환되는 두 코드 워드들만이 0들로 구성되고 1들로 구성된다.

[0188] n=4를 위한 예:

[0189] 수 일차 이차

[0190] 1 0000 111

[0191] 2 0001 0001

[0192] 3 001 001

[0193] 4 01 01

[0194] 5 10 10

- [0195] 6 110 110
- [0196] 7 111 0000
- [0198] 구성 규칙 2: '라이스에 대한 단항' 코드들 및 라이스에 대한 단항 인코더/디코더들(10 및 22)
- [0200] 라이스 코드들에 대한 단항은 일차 코드로서 절단된 단항 코드를 사용한다. 즉, 단항 코드 워드들은 '1', '01', '001'로 시작하여 2^n+1 코드 워드들이 발생되고 가장 긴 코드 워드가 형성될 때까지 발생되며, 트레일링 1은 제거된다. n 은 라이스 코드에 대한 단항의 파라미터이다. 이차 접두부호 없는 코드는 다음과 같이 일차 무-접두부호 코드의 코드 워드로부터 구성된다. 0들만으로 구성되는 일차 코드 워드에 대하여, 코드 워드 '1'이 할당된다. 다른 모든 코드 워드들은 일차 무-접두부호 코드의 상응하는 코드 워드의 0들의 수의 n -비트 이진 표현을 갖는 연속되는 코드 워드 '0'으로 구성된다.
- [0201] $n=3$ 을 위한 예:
- [0202] 수 일차 이차
- [0203] 1 1 0000
- [0204] 2 01 0001
- [0205] 3 001 0010
- [0206] 4 0001 0011
- [0207] 5 00001 0100
- [0208] 6 000001 0101
- [0209] 7 0000001 0110
- [0210] 8 00000001 0111
- [0211] 9 00000000 1
- [0212] 이는 무한 단항 코드를 라이스 파라미터 2^n 을 갖는 라이스 코드에 매핑하는 것과 동일하다는 것에 유의하여야 한다.
- [0214] 구성 규칙 3: '3 빈' 코드
- [0215] 3 빈 코드는 다음과 같이 주어진다:
- [0216] 수 일차 이차
- [0217] 1 000 0
- [0218] 2 001 100
- [0219] 3 010 101
- [0220] 4 100 110
- [0221] 5 110 11100
- [0222] 6 101 11101
- [0223] 7 011 11110
- [0224] 8 111 11111

- [0226] 이는 일차 코드(심벌 시퀀스들)가 고정된 길이를 가지며(항상 3 빈) 코드 워드들이 1들의 수를 선행함으로써 분류되는 특성을 갖는다.
- [0228] 3 빈 코드의 효율적인 구현이 다음에 설명된다. 3 빈 코드를 위한 인코더와 디코더는 다음의 방법으로 테이블들을 저장하지 않고 구현될 수 있다.
- [0230] 인코더(10 중의 어느 하나)에서, 빈 스트림(즉, 7)으로부터 3 빈들이 판독된다. 만일 이러한 3 빈들이 정확하게 하나의 1을 포함하면, 코드 워드 '1'이 비트 스트림에 판독되고 1의 위치(00을 갖는 오른쪽으로부터 시작하여)의 이진 표현으로 구성되는 두 빈들이 뒤따른다. 반일, 3 빈들이 정확하게 하나의 0을 포함하면, 비트스트림에 코드 워드 '111'이 판독되고 0의 위치(00을 갖는 오른쪽으로부터 시작하여)의 이진 표현으로 구성되는 두 빈들이 뒤따른다. 나머지 코드 워드들 '000' 및 '111'이 각각 '0' 및 '11111'에 매핑된다.
- [0232] 디코더(22 중의 어느 하나)에서, 각각의 비트스트림(24)으로부터 하나의 빈 또는 비트가 판독된다. 만일 이것이 '0'과 동일하면, 코드 워드 '000'이 빈 스트림(21)에 디코딩된다. 만일 이것이 '1'과 동일하면, 두 개 이상의 빈이 비트 스트림(24)로부터 판독된다. 만일 이러한 두 비트가 '11'과 동일하지 않으면, 그것들은 수의 이진 표현으로서 해석되고 두 0과 하나의 1은 수에 의해 1의 위치가 결정되는 것과 같이 비트 스트림에 디코딩된다. 만일 두 비트가 '11'과 동일하면, 두 개 이상의 비트가 판독되고 수의 이진 표현으로서 해석된다. 만일 이러한 수가 3보다 작으면, 두 1과 하나의 0이 디코딩되고 수는 0의 위치를 결정한다. 만일 그것이 3과 동일하면, '111'이 빈 스트림에 디코딩된다.
- [0234] 단항 빈 파이프 코드들의 효율적인 구현이 다음에 설명된다. 단항 빈 파이프 코드들을 위한 인코더와 디코더는 카운터(counter)를 사용함으로써 효율적으로 구현될 수 있다. 빈 파이프 코드들의 구조 때문에, 빈 파이프 코드들의 인코딩과 디코딩은 구현하기가 쉽다:
- [0236] 인코더(10 중의 어느 하나)에서, 만일 코드 워드의 첫 번째 빈이 '0'과 동일하면, 빈들은 '1'이 발생하거나 n 0들이 판독될 때까지(코드 워드의 첫 번째 '0'을 포함하여) 처리된다. 만일 '1'이 발생되면, 판독된 빈들은 비트 스트림에 변하지 않은 채로 기록된다. 그렇지 않으면(즉, n 0들이 판독되었으면), $n-1$ 1들이 비트 스트림에 기록된다. 만일 코드 워드의 첫 번째 빈이 '1'과 동일하면, 빈들은 '0'이 발생하거나 $n-1$ 1들이 판독될 때까지(코드 워드의 첫 번째 '1'을 포함하여) 처리된다. 만일 '0'이 발생되면, 판독된 빈들은 비트 스트림에 변하지 않은 채로 기록된다. 그렇지 않으면(즉, $n-1$ 1들이 판독되었으면), n 0들이 비트 스트림에 기록된다.
- [0238] 디코더(322 중의 어느 하나)에서, 인코더에서와 동일한 알고리즘이 사용되는데, 그 이유는 위에서 설명된 것과 같이 이는 빈 파이프 코드들에서도 동일하기 때문이다.
- [0240] 라이스 코드들에 대한 단항의 효율적인 구현이 다음에 설명된다. 라이스 코드들에 대한 단항을 위한 인코더와 디코더는 이제 설명될 것과 같은 카운터를 사용함으로써 효율적으로 구현될 수 있다.
- [0242] 인코더(310 중의 어느 하나)에서, 빈들은 하나의 1이 발생하거나 또는 2^n 0들이 판독될 때까지 빈 스트림으로부터 판독된다(즉, 7). 0들의 수가 계수된다. 만일 계수된 수가 2^n 과 동일하면, 코드 워드 '1'이 비트 스트림에 기록된다. 그렇지 않으면, '0'이 기록되고 그 뒤에 n 비트들로 기록되는, 계수된 수의 단항 표현이 뒤따른다.
- [0244] 디코더(322 중의 어느 하나)에서, 하나의 비트가 판독된다. 만일 그것이 '1'과 동일하면, 2^n 0들이 빈 스트림에

디코딩된다. 만일 그것이 '0'과 동일하면, n 이상의 비트가 판독되고 수의 이진 표현으로서 해석된다. 0들의 이러한 수는 빈 스트림에 디코딩되고, 그 뒤에 '1'이 뒤따른다.

[0246]

바꾸어 말하면, 방금 언급된 실시 예들은 심벌들의 시퀀스의 이전의 심벌들 내에 포함된 정보를 기초로 하여 심벌들의 시퀀스의 각각의 시퀀스에 다수의 파라미터들(305)을 할당하도록 구성되는 할당기(316); 각각, 각각의 엔트로피 인코더(310)로 전달된 심벌들(307)을 각각의 비트스트림(312)으로 변환하도록 구성되는 복수의 엔트로피 인코더(310); 및 각각의 심벌(303)을 복수의 엔트로피 인코더(10) 중 선택된 하나에 전달하도록 구성되며, 선택은 각각의 심벌(303)에 할당되는 파라미터들(305)의 수에 따르는 선택기(6);를 포함하는, 심벌들(303)의 시퀀스를 인코딩하기 위한 인코더를 설명한다. 방금 설명된 실시 예들에 따르면, 엔트로피 인코더들의 적어도 하나의 첫 번째 서브셋은 심벌들(307)의 스트림 내의 가변 길이들의 심벌 시퀀스들을 비트스트림(312) 내에 삽입되는 가변 길이들의 코드워드들로 매핑하도록 구성되는 가변 길이 인코더일 수 있으며, 첫 번째 서브셋의 각각의 엔트로피 코더들(310)은 일차 무-접두부호의 코드 워드들의 거의 2개가 이차 무-접두부호 코드의 동일한 코드 워드들에 매핑되며 일차 및 이차 무-접두부호 코드들의 두 코드 워드들은 서로 다른 길이를 가지며 교환 방식으로 서로 상에 매핑되는 것과 같이, $(2n-1) \geq 3$ 코드 워드들을 갖는 일차 무-접두부호의 코드 워드들이 일차 접두부호 코드와 동일한 이차 무-접두부호 코드의 코드워드들에 매핑되는 전단사 매핑 규칙을 사용하며, 엔트로피 인코더들은 위에서 언급된 확률 간격의 서로 다른 부를 포함하도록 하기 위하여 서로 다른 n 을 사용할 수 있다. 첫 번째 무-접두부호 코드는 첫 번째 무-접두부호의 코드들이 $(a,b)_2, (a,a,b)_3, \dots, (a, \dots, a,b)_n, (b,a)_2, (b,b,a)_3, \dots, (b, \dots, b,a)_{n-1}, (b, \dots, b)_{n-1}$ 이며, 교환 방식으로 서로 상으로 매핑되는 두 코드워드는 $b \neq a$ 이고 $a, b \in \{0,1\}$ 을 갖는 $(a, \dots, a)_n$ 및 $(b, \dots, b)_{n-1}$ 인 것과 같이 구성된다. 대안들도 실현 가능하다.

[0248]

바꾸어 말하면, 엔트로피 인코더들의 각각의 첫 번째 서브셋은 전달된 심벌들을 각각의 비트스트림 내의 각각의 엔트로피 인코더로 변환하는데 있어서, (1.1) 이 경우에 각각의 엔트로피 인코더가 코드워드를 첫 번째 심벌과 동일하고 그 다음에 심벌 b 까지 각각의 엔트로피 인코더에 전달되는 다음의 심벌들과 동일한, 각각의 비트스트림에 기록하도록 구성되는, $b \neq a$ 및 $a, b \in \{0,1\}$ 을 갖는 b 가 첫 번째 심벌 다음의 그 다음의 $n-1$ 심벌들 내에 발생하는지, (1.2) 이 경우에 각각의 엔트로피 인코더가 하나의 코드워드를 $(b, \dots, b)_{n-1}$ 과 동일한 각각의 비트스트림에 기록하도록 구성되는, 첫 번째 심벌 다음의 그 다음의 $n-1$ 심벌들 내에 어떠한 b 도 발생하지 않는지에 대하여 결정하기 위하여 (1) 이 경우에 각각의 엔트로피 인코더가 각각의 엔트로피 인코더로 전달되는 다음의 심벌들을 조사하도록 구성되는, 첫 번째 심벌이 $a \in \{0,1\}$ 과 동일한지에 대하여 결정하기 위하여; 또는 (2.1) 이 경우에는 각각의 엔트로피 인코더가 첫 번째 심벌과 동일하고 그 다음에 심벌 a 까지 각각의 엔트로피 인코더로 전달되는 다음의 심벌들과 동일한, 코드워드를 각각의 비트스트림에 기록하도록 구성되는, a 가 첫 번째 심벌 다음의 그 다음의 $n-2$ 심벌 내에 발생하는지, (2.2) 이 경우에는 각각의 엔트로피 인코더가 $(a, \dots, a)_n$ 과 동일한, 코드워드를 각각의 비트스트림에 기록하는, 어떠한 a 도 그 다음의 $n-2$ 심벌 내에 발생하지 않는지에 대하여 결정하기 위하여 (2) 이 경우에는 각각의 엔트로피 인코더가 각각의 엔트로피 인코더로 전달되는 다음의 심벌들을 조사하도록 구성되는, 첫 번째 심벌이 b 와 동일한지에 대하여 결정하기 위하여, 각각의 엔트로피 인코더로 전달되는 첫 번째 심벌을 조사하도록 구성될 수 있다.

[0250]

부가적으로 또는 대안으로서, 엔트로피 인코더들(10)의 두 번째 서브셋은 각각 가변 길이들의 심벌 시퀀스들을 고정 길이들의 코드워드에 매핑하도록 구성되는 가변 길이 인코더일 수 있으며, 두 번째 서브셋의 각각의 엔트로피 코더들은 일차 절단된 단항 코드의 코드워드($bb \dots b$)가 이차 무-접두부호 코드의 코드워드(c) 상으로 매핑되고 일차 절단된 단항 코드의 모든 다른 코드워드들 $\{(a), (ba), (bba), \dots, (b, \dots, ba)\}$ 은 접두부호로서 $c \neq d$ 이고 $c, d \in \{0,1\}$ 을 갖는 (d) 및 접미부호로서 n -비트 워드를 갖는 코드워드들 상으로 매핑되는 것과 같이, $b \neq a$ 이고 $a, b \in \{0,1\}$ 을 갖는 형태 $\{(a), (ba), (bba), \dots, (b, \dots, ba), (bb \dots b)\}$ 의 $2^n + 1$ 코드 워드들을 갖는 일차 절단된 단항 코드의 코드 워드들이 이차 무-접두부호 코드의 코드 워드들 상에 매핑하는 전단사 매핑 규칙을 사용하며, 엔트로피 인코더들은 서로 다른 n 을 사용한다. 엔트로피 인코더들의 각각의 두 번째 서브셋은 n -비트 워드가 일차 절단된 단항 코드의 각각의 코드워드 내의 b '들의 수의 n -비트 표현인 것과 같이 구성될 수 있다. 그러나, 대안들이 실현 가능하다.

[0252] 다시, 각각의 인코더(10)의 운용의 방식의 관점으로부터, 엔트로피 인코더들이 두 번째 세트는 각각의 엔트로피 인코더로 전달된 심벌들을 각각의 비트스트림으로 전환하는데 있어서, a 가 발생할 때까지, 또는 각각의 엔트로피 인코더로 전달된 심벌들의 시퀀스 내의 n' 들의 수가 2^n 에 도달할 때까지 각각의 엔트로피 인코더로 전달된 심벌들의 시퀀스 내의 n' 들의 수를 계수하도록 구성될 수 있으며, 시퀀스의 모든 2^n 심벌은 b 이며, (1) 만일 b' 들이 수가 2^n 과 동일하면, 각각의 비트스트림에 대한 이차 무-접두부호 코드의 코드워드로서 c 를 $c \in \{0,1\}$ 로 기록하며, (2) 만일 b' 들이 수가 2^n 보다 낮으면, 접두부호로서 $c \neq d$ 과 $d \in \{0,1\}$ 을 갖는 (d) 및 접미부호로서 b' 들의 수에 딸 결정되는 n -비트 워드를 갖는, 각각의 비트스트림에 대한 이차 무-접두부호 코드의 코드워드를 기록한다.

[0254] 또한 부가적으로 또는 대안으로서, 엔트로피 디코더들 중 미리 결정된 하나는 각각 가변 길이들의 코드워드들을 고정 길이들의 심벌 시퀀들에 매핑하도록 구성되는 가변 길이 코더들일 수 있으며, 미리 결정된 엔트로피 코더는 $a \in \{0,1\}$ 을 갖는 일차 코드의 코드워드 $(aaa)_3$ 가 $c \in \{0,1\}$ 을 갖는 코드워드 (c) 상으로 매핑되는 것과 같이 일차 부호의 길이 3의 23 코드 워드들이 이차 무-접두부호 코드의 코드 워드들로 매핑되는 전단사 매핑 규칙을 사용하며, $b \neq a$ 및 $b \in \{0,1\}$ 을 갖는 정확하게 하나의 b 를 갖는 일차 코드의 모든 3개의 코드워드는 접두부호로서 $c \neq d$ 및 $d \in \{0,1\}$ 을 갖는 (d) 및 접미부호로서 2-비트의 첫 번째 세트 중에서 각각의 첫 번째 2-비트 워드를 갖는 코드워드들 상으로 매핑되며, 정확하게 하나의 a 를 갖는 일차 코드의 모든 3개의 코드워드는 접두부호로서 d 를 갖는 코드워드들 상으로 매핑되며, 접미부호로서, 첫 번째 2-비트 워드의 연결은 첫 번째 세트의 요소가 아닐 수 있고 두 번째 2-비트 워드는 두 번째 세트의 요소가 아닐 수 있으며, 코드워드 $(bbb)_3$ 은 접두부호로서 (d)를 갖는 하나의 코드워드 상으로 매핑되며, 접미부호로서, 첫 번째 2-비트 워드의 연결은 첫 번째 세트의 요소가 아닐 수 있고 두 번째 2-비트 워드는 두 번째 세트의 요소가 아닐 수 있다. 정확하게 하나의 b 를 갖는 일차 코드의 코드워드들의 첫 번째 2-비트 워드는 일차 코드의 각각의 코드워드 내의 b 의 위치의 2-비트 표현일 수 있으며, 정확하게 하나의 a 를 갖는 일차 코드의 코드워드들의 두 번째 2-비트 워드는 일차 코드의 각각의 코드워드 내의 a 의 위치의 2-비트 표현일 수 있다. 그러나, 대안들도 실현 가능하다.

[0256] 다시, 엔트로피 인코더들 중 미리 결정된 하나는 미리 결정된 엔트로피 인코더로 전달되는 심벌들을 각각의 비트스트림으로 전환하는데 있어서, (1) 이 경우에는 미리 결정된 엔트로피 인코더가 각각의 비트스트림에 코드워드 (c)를 기록하도록 구성되는, 트리플릿이 a' 들로 구성되는지, (2) 이 경우에는 미리 결정된 엔트로피 인코더가 각각의 비트스트림에, 접두부호로서 (d) 및 접미부호로서 트리플릿 내의 b 의 위치의 2-비트 표현을 갖는 코드워드를 기록하도록 구성되는, 트리플릿이 정확하게 하나의 b 를 포함하는지, (3) 이 경우에는 미리 결정된 엔트로피 인코더가 각각의 비트스트림에, 접두부호로서 (d) 및 접미부호로서 첫 번째 세트의 요소가 아닐 수 있는 첫 번째 2-비트의 연결과 트리플릿 내의 a 의 위치의 2-비트 표현을 갖는 코드워드를 기록하도록 구성되는, 트리플릿이 정확하게 하나의 a 를 포함하는지, 또는 (4) 이 경우에는 미리 결정된 엔트로피 인코더가 각각의 비트스트림에, 접두부호로서 (d) 및 접미부호로서 첫 번째 세트의 요소가 아닐 수 있는 첫 번째 2-비트의 연결과 두 번째 세트의 요소가 아닐 수 있는 첫 번째 2-비트를 갖는 코드워드를 기록하도록 구성되는, 트리플릿이 b' 들로 구성되는지에 대하여 트리플릿들 내의 미리 결정된 엔트로피 인코더에 대한 심벌들을 조사하도록 구성될 수 있다.

[0258] 디코딩 면과 관련하여, 방금 언급된 실시 예들은 각각 각각의 비트스트림을 심벌들(321)로 전환하도록 구성되는 복수의 엔트로피 디코더(322); 다수의 파라미터를 심벌들의 시퀀스의 이전에 재구성된 심벌들 내에 포함된 정보를 기초로 하여 재구성되는 심벌들의 시퀀스의 각각의 심벌(326)에 할당하도록 구성되는 할당기(316); 및 복수의 엔트로피 디코더 중 선택된 하나로부터 재구성되는 심벌들의 시퀀스의 각각의 심벌(325)을 검색하도록 구성되는 선택기(318);를 포함하며 선택은 각각의 심벌에 정의되는 파라미터들의 수에 의존하는, 심벌들(326)이 시퀀스를 재구성하기 위한 디코더를 개시한다. 방금 설명된 실시 예들에 따르면, 엔트로피 디코더들(322)의 적어도 하나의 첫 번째 서브셋은 각각 가변 길이들의 코드워드들을 가변 길이들의 심벌 시퀀스들에 매핑하도록 구성되는 가변 길이 디코더들이며, 첫 번째 서브셋의 각각의 엔트로피 디코더들(22)은 일차 무-접두부호 코드의 코

드 워드들의 거의 2개가 이차 무-접두부호 코드워드의 동일한 코드 워드들에 매핑하나 일차 및 이차 무-접두부호 코드들의 두 코드 워드는 서로 다른 길이를 가지며 교환 방식으로 서로 상으로 매핑되는 것과 같이 $(2n-1) \geq 3$ 코드워드들을 갖는 일차 무-접두부호 코드의 코드 워드들이 일차 접두부호 코드와 동일한 이차 무-접두부호 코드의 코드워드들에 매핑되는 전단사 매칭 규칙을 사용하며, 엔트로피 인코더들은 서로 다른 n 이다. 첫 번째 무-접두부호 코드는 첫 번째 무-접두부호 코드의 코드워드들이 $(a,b)_2, (a,a,b)_3, \dots, (a,\dots,a,b)_n, (a,\dots,a)_2, (b,b,a)_3, \dots, (b,\dots,b,a)_{n-1}, (b,\dots,b)_{n-1}$ 이고, 교환 방식으로 서로 매핑되는 두 코드워드들은 $(a,\dots,a)_n$ 및 $(b,\dots,b)_n$ 일 수 있으며, $b \neq a$ 이고 $a, b \in \{0,1\}$ 이다. 그러나, 대안들이 실현 가능하다.

[0260] 엔트로피 디코더들의 각각의 첫 번째 서브셋은 각각의 비트스트림을 심벌들로 전환하는데 있어서, (1) 이 경우에 있어서, (1.1) 이 경우에는 각각의 엔트로피 디코더가 첫 번째 비트와 동일하고 그 다음에 비트 b 까지 각각의 비트스트림의 비트들과 동일한, 심벌 시퀀스를 재구성하도록 구성되는, $b \neq a$ 이고 $b \in \{0,1\}$ 을 갖는 b 가 첫 번째 비트 다음의 그 다음의 $n-1$ 비트들 내에서 발생하는지, 또는 (1.2) 이 경우에는 각각의 엔트로피 디코더가 $(b,\dots,b)_{n-1}$ 과 동일한, 심벌 시퀀스를 재구성하도록 구성되는, 어떠한 b 도 첫 번째 비트 다음의 그 다음의 $n-1$ 비트들 내에서 발생하지 않는지를 결정하기 위하여 각각의 엔트로피는 각각의 비트스트림의 다음의 비트들을 결정하도록 구성되는, 첫 번째 비트가 $a \in \{0,1\}$ 과 동일한지를 결정하기 위하여, 또는 (2) 이 경우에 있어서, (2.1) 이 경우에는 각각의 엔트로피 디코더가 첫 번째 비트와 동일하고 그 다음에 심벌 a 까지 각각의 비트스트림의 비트들과 동일한, 심벌 시퀀스를 재구성하도록 구성되는, 그 다음의 $n-2$ 비트들 내에서 발생하는지, 또는 (2.2) 이 경우에는 각각의 엔트로피 디코더가 $(a,\dots,a)_n$ 과 동일한, 심벌 시퀀스를 재구성하도록 구성되는, 어떠한 a 도 첫 번째 비트 다음의 그 다음의 $n-2$ 비트들 내에서 발생하지 않는지를 결정하기 위하여 각각의 엔트로피는 각각의 비트스트림의 다음의 비트들을 결정하도록 구성되는, 첫 번째 비트가 $a \in \{0,1\}$ 과 동일한지를 결정하기 위하여, 각각의 비트스트림의 첫 번째 비트를 조사하도록 구성될 수 있다.

[0262] 부가적으로, 또는 대안으로서, 엔트로피 디코더들(322)의 적어도 두 번째 세트는 이차 무-접두부호 코드의 코드 워드 (c) 가 일차 절단된 코드의 코드워드($bb\dots b$) 상으로 매핑되고 접두부호로서 $c \neq d$ 와 $c, d \in \{0,1\}$ 및 접미부호로서 n -비트 워드를 갖는 (d) 를 갖는 코드워드들이 일차 절단된 단항 코드의 다른 코드워드들 $\{(a), (ba), (bba), \dots, (b\dots ba)\}$ 의 각각의 하나에 매핑되는 것과 같이, 이차 무-접두부호 코드의 코드 워드들이 일차 절단된 단항 코드 상으로 매핑되는 $b \neq a$ 및 $a, b \in \{0,1\}$ 을 갖는 형태 $\{(a), (ba), (bba), \dots, (b\dots ba), (bb\dots b)\}$ 의 2^{n+1} 코드 워드들을 갖는 전단사 매핑 규칙을 사용하는 두 번째 서브셋의 각각의 엔트로피 디코더들로, 각각 고정 길이들의 코드워드들을 가변 길이들의 심벌 시퀀스들에 매핑하도록 구성되는 가변 길이 디코더일 수 있으며, 엔트로피 디코더들은 서로 다른 n 을 사용한다.

[0264] 엔트로피 디코더들의 각각의 두 번째 세트는 각각 고정 길이들의 코드워드들을 가변 길이들의 심벌 시퀀스에 매핑하도록 구성되고, 각각의 엔트로피 디코더의 비트스트림을 심벌들로 전환하는데 있어서, (1) 이 경우에 각각의 엔트로피 디코더가 $b \in \{0,1\}$ 을 갖는 $(bb\dots b)_2^n$ 과 동일한 심벌 시퀀스를 재구성하도록 구성되는, $c \in \{0,1\}$ 을 갖는, 각각의 비트스트림의 첫 번째 비트가 c 와 동일한지; 또는 (2) 이 경우에 각각의 엔트로피 디코더는 첫 번째 비트 다음의, 각각의 비트스트림의 n 추가의 비트들로부터 n -비트 워드를 결정하고 $b \neq a$ 이고 $b \in \{0,1\}$ 을 가지며 b 들의 수는 n -비트 단어에 의존하는 형태 $\{(a), (ba), (bba), \dots, (b\dots ba), (bb\dots b)\}$ 인 그 것으로부터 심벌 시퀀스를 재구성하도록 구성되는, 각각의 비트스트림의 첫 번째 비트가 $c \neq d$ 이고 $c, d \in \{0,1\}$ 을 갖는 d 와 동일한지를 결정하기 위하여 각각의 비트스트림의 첫 번째 비트를 조사하도록 구성되는 가변 길이 디코더일 수 있다.

[0266] 부가적으로 또는 대안으로서, 엔트로피 디코더들 중 미리 결정된 하나는 각각 가변 길이들의 코드워드들을 고정 길이들의 심벌 시퀀스들에 매핑하도록 구성되는 가변 길이 디코더들일 수 있으며, 미리 결정된 엔트로피 디코더는 $c \in \{0,1\}$ 을 갖는 코드워드가 $a \in \{0,1\}$ 을 갖는 일차 코드의 코드워드 $(aaa)_3$ 에 매핑되는 것과 같이 이차 무-접두부호 코드의 코드 워드 (c) 가 일차 코드의 길이 3의 2^3 코드 워드들에 매핑되는 전단사 매핑 규칙을 사용하

며, 접두부호로서 $c \neq d$ 및 $d \in \{0,1\}$ 을 갖는 (d)를 갖는 코드워드들 및 접미부호로서 3개의 2-비트 워드들의 첫 번째 세트 중에서 각각의 첫 번째 2-비트는 정확하게 하나의 a를 갖는 일차 코드의 모든 3개의 코드워드 상으로 매핑되며, 접두부호로서 (d)를 갖는 코드워드 및 첫 번째 2-비트의 연결은 첫 번째 세트의 요소가 아닐 수 있고 두 번째 2-비트는 두 번째 세트의 요소가 아닐 수 있는데 그 이유는 접미부호가 코드워드 (bbb)₃ 상으로 매핑되기 때문이다. 정확하게 하나의 b를 갖는 일차 코드의 코드워드들의 첫 번째 2-비트 워드는 일차 코드의 각각의 코드워드 내의 b의 위치의 2-비트 표현일 수 있으며, 정확하게 하나의 a를 갖는 일차 코드의 코드워드들의 두 번째 2-비트 워드는 일차 코드의 각각의 코드워드 내의 a의 위치의 2-비트 표현일 수 있다. 그러나, 대안들도 실현 가능하다.

[0268] 엔트로피 디코더들 중 미리 결정된 하나는 각각 가변 길이들의 코드워드들을 각각의 3개의 심벌의 심벌 시퀀스들로 매핑하도록 구성되고, 각각의 엔트로피 디코더의 비트스트림을 심벌들로 전환하는데 있어서, (1) 이 경우에 미리 결정된 엔트로피 디코더가 0 {0,1}을 갖는 (aaa)₃와 동일한 심벌 시퀀스를 재구성하도록 적용되는, 각각의 비트스트림의 첫 번째 비트가 미리 결정된 엔트로피 디코더가 $c \in \{0,1\}$ 을 갖는 c와 동일한지를, 또는 (2) 이 경우에 미리 결정된 엔트로피 디코더가 첫 번째 비트 다음의, 각각의 비트스트림의 2개의 추가의 비트들로부터 첫 번째 2-비트를 결정하도록 구성되는, 각각의 비트스트림의 첫 번째 비트가 $c \neq d$ 이고 $d \in \{0,1\}$ 을 갖는 d와 동일한지를 결정하기 위하여 각각의 비트스트림의 첫 번째 비트를 조사하도록 구성되며, (2.1) 이 경우에 미리 결정된 엔트로피 디코더는 $b \neq a$ 이고 $b \in \{0,1\}$ 을 가지며 각각의 심벌 시퀀스 내의 b의 위치는 첫 번째 2-비트 워드에 의존하는, 정확하게 하나의 b를 갖는 심벌 시퀀스를 재구성하도록 구성되는, 첫 번째 2-비트 워드가 3개의 2-비트 단어들의 첫 번째 세트의 요소가 없는지를, 또는 (2.2) 이 경우에 미리 결정된 엔트로피 디코더는 첫 번째 2-비트 워드가 결정된 2개의 비트들 다음의, 각각의 비트스트림의 2개의 추가의 비트들로부터 두 번째 2-비트 워드를 결정하도록 구성되는, 첫 번째 2-비트 워드가 첫 번째 세트의 요소인지를 결정하기 위하여 첫 번째 2-비트 워드를 조사하도록 구성되며, (3.1) 이 경우에 미리 결정된 엔트로피 디코더는 각각의 심벌 시퀀스 내의 a의 위치가 두 번째 2-비트 워드에 의존하는, 정확하게 하나의 a를 갖는 심벌 시퀀스를 재구성하도록 구성되는, 두 번째 2-비트 워드가 3개의 2-비트 단어들의 두 번째 세트의 요소가 없는지를, 또는 (3.2) 이 경우에 미리 결정된 엔트로피 디코더는 (bbb)₃와 동일한 심벌 시퀀스를 재구성하도록 구성되는, 두 번째 2-비트 워드가 3개의 2-비트 워드들의 두 번째 세트의 요소인지를 결정하기 위하여 두 번째 2-비트 워드를 조사하도록 구성되는, 가변 길이 디코더일 수 있다.

[0270] 이제, 비디오 코딩 전략의 일반적인 개념을 설명한 후에, 위의 실시 예들과 관련하여 본 발명의 실시 예들이 설명된다. 바꾸어 말하면, 아래에 설명되는 실시 예들은 위의 전략들의 사용에 의해 구현될 수 있으며, 마찬가지로, 위의 코딩 전략들은 아래에 설명되는 실시 예들을 사용하고 이용할 수 있다.

[0272] 도 7 내지 9와 관련하여 설명된 위의 실시 예들에서, 도 1 내지 6의 엔트로피 인코더와 디코더들은 파이프 개념에 따라 구현되었다. 하나의 특정 실시 예는 산술 단일-확률 상태 인코더/디코더들(310 및 322)을 사용하였다. 아래에 설명될 것과 같이, 대안의 실시 예에 따라, 엔터티들(306-310) 및 상응하는 엔터티들(318 내지 322)은 통상의 엔트로피 인코딩 엔진에 의해 대체될 수 있다. 예로서, 단지 하나의 통상의 상태(R 및 L)를 관리하고 모든 심벌을 하나의 통상의 비트스트림 내로 인코딩하는, 산술 인코딩 엔진을 상상하면, 그렇게 함으로써, 병렬 처리에 대한 존재하는 파이프 개념의 바람직한 양상들을 포기하나, 아래에 설명되는 것과 같이 부분 비트스트림들의 인터리빙이 필요성을 방지한다. 그렇게 함으로써, 업데이트(테이블 록-업과 같은)에 의해 콘텍스트들의 확률이 추정되는 확률 상태들의 수는 확률 간격 세분이 실행되는 확률 상태들의 수보다 높을 수 있다. 즉, 테이블(Rtab) 내로의 지수화 전에 확률 간격 모델 폭 값의 양자화와 유사하게, 또한 확률 상태 지수가 양자화될 수 있다. 단일 인코더/디코더들(310 및 322)을 위한 가능한 구현을 위한 위의 설명은 콘텍스트-적응적 이진 산술 인코딩/디코딩 엔진들과 같은 엔트로피 인코더/디코더들(318-322/306-310)이 구현의 예를 위하여 확대될 수 있다.

[0274] 더 정확히 설명하면, 일 실시 예에 따라, 파라미터 할당기(여기서는 콘텍스트 할당기로서 작용하는)의 출력에 부착되는 엔트로피 인코더는 다음의 방법으로 운용될 수 있다:

- [0275] 0. 할당기(304)는 확률 파라미터와 함께 빈 값을 전달한다. 확률은 pstate_current[bin]이다.
- [0276] 1. 따라서, 엔트로피 인코딩 엔진은: 1) valLPS, 2) 빈, 및 3) 확률 분포 추정 pState_current[bin]을 받는다. pState_current[bin]은 Rtab의 구별가능한 확률 상태 지수들의 수보다 더 많은 상태들을 가질 수 있다. 만일 그렇다면, pState_current[bin]은 p_state 즉, 그리고 나서 테이블 Rtab을 액세스하도록 사용되는 지수를 획득하기 위하여 예를 들면, 1 이상 및 바람직하게는 2 또는 3인 m을 갖는 m LSB들의 무시에 의한 것과 같이 양자화될 수 있다. 그러나, 양자화는 제거될 수 있는데, 즉, p_state는 pState_currnet[bin]일 수 있다.
- [0277] 2. 그리고 나서, R의 양자화가 실행된다(위에서 설명된 것과 같이: p_state의 모든 구별가능한 값들을 위하여 하나의 R(및 하나의 통상적 비트스트림을 갖는 상응하는 L)이 사용/관리되거나, 또는 p_stste의 구별가능한 값 당 하나의 R(및 R/L 쌍 당 관련 부분 비트스트림을 갖는 상응하는 L)이 그러한 값 당 하나의 빈 인코더(310)를 갖는 것과 상응할 수 있다).
- [0278] $q_index = Qtab[R \gg q]$ (또는 일부 다른 형태의 양자화)
- [0279] 3. 그리고 나서, R_{LPS} 의 R의 결정이 실행된다:
- [0280] $R_{LPS} = Rtab[p_state][q_index]$; Rtab은 $p[p_state] \cdot Q[q_index]$ 를 위한 그 안에 저장된 미리 계산된 값들을 갖는다.
- [0281] $R = R - R_{LPS}$ [즉, R은 '빈'이 더 가능성이 있는 심벌이었던 것처럼 예비로 미리 업데이트된다]
- [0282] 4. 새로운 부분 간격의 계산:
- [0283] if (bin = 1-valMPS) then
- [0284] $L \leftarrow L + R$
- [0285] $L \leftarrow RR_{LPS}$
- [0286] 5. 비트들을 기록하는, L과 R의 재구성.
- [0288] 유사하게, 파라미터 할당기(여기서는 콘텍스트 할당기로서 작용하는)의 출력에 부착되는 엔트로피 디코더는 다음의 방법으로 운용될 수 있다:
- [0289] 0. 할당기(304)는 확률 파라미터와 함께 빈 값을 전달한다. 확률은 pstate_current[bin]이다.
- [0290] 1. 따라서, 엔트로피 디코딩 엔진은: 1) valLPS, 및 2) 확률 분포 추정 pState_current[bin]을 받는다. pState_current[bin]은 Rtab의 구별가능한 확률 상태 지수들의 수보다 더 많은 상태들을 가질 수 있다. 만일 그렇다면, pState_current[bin]은 p_state 즉, 그리고 나서 테이블 Rtab을 액세스하도록 사용되는 지수를 획득하기 위하여 1 이상 및 바람직하게는 2 또는 3인 m을 갖는 m LSB들의 무시에 의한 것과 같이 양자화될 수 있다. 그러나, 양자화는 제거될 수 있는데, 즉, p_state는 pState_currnet[bin]일 수 있다.
- [0291] 2. 그리고 나서, R의 양자화가 실행된다(위에서 설명된 것과 같이: p_state의 모든 구별가능한 값들을 위하여 하나의 R(및 하나의 통상적 비트스트림을 갖는 상응하는 L)이 사용/관리되거나, 또는 p_stste의 구별가능한 값 당 하나의 R(및 R/L 쌍 당 관련 부분 비트스트림을 갖는 상응하는 L)이 그러한 값 당 하나의 빈 인코더(310)를 갖는 것과 상응할 수 있다).
- [0292] $q_index = Qtab[R \gg q]$ (또는 일부 다른 형태의 양자화)
- [0293] 3. 그리고 나서, R_{LPS} 의 R의 결정이 실행된다:
- [0294] $R_{LPS} = Rtab[p_state][q_index]$; Rtab은 $p[p_state] \cdot Q[q_index]$ 를 위한 그 안에 저장된 미리 계산된 값들을 갖는다.
- [0295] $R = R - R_{LPS}$ [즉, R은 '빈'이 더 가능성이 있는 심벌이었던 것처럼 예비로 미리 업데이트된다]
- [0296] 4. 부분 간격의 위치에 따른 빈의 결정:

- [0297] if ($V \geq R$) then
- [0298] bin $\leftarrow 1 - \text{valMPS}$
- [0299] $R \leftarrow R_{LPS}$
- [0300] else
- [0301] bin $\leftarrow 1 - \text{valMPS}$ (빈은 더 가능성이 있는 심벌로서 디코딩된다; 실제 빈 값은 이러한 빈 정보와 valMPS에 의해 획득된다)
- [0302] 5. 하나의 비트를 판독하고 V를 업데이트하는, R의 재구성.
- [0304] 위에서 설명된 것과 같이, 할당기(4)는 pState_current[bin]을 각각의 빈에 할당한다. 컨텍스트 선택을 기초로 하여 연관이 수행된다. 즉, 할당기(4)는 차례로 그것과 관련된 pState_current[bin]을 갖는 컨텍스트 지수(ctxIdx)를 선택할 수 있다. 언제나 확률 업데이트가 실행될 수 있으며, 확률(pState_current[bin])은 코딩된 비트의 값에 따라 실행된다:
- [0305] if (bit = 1 - valMPS) then
- [0306] pState_current $\leftarrow$ Next_State_LPS[pState_current]
- [0307] if (pState_current = 0) then valMPS $\leftarrow 1 - \text{valMPS}$
- [0308] else
- [0309] pState_current $\leftarrow$ Next_State_MPS[pState_current]
- [0311] 만일 하나 이상의 컨텍스트가 제공되면, 적응성은 컨텍스트 방법으로 수행되는데, 즉, 코딩을 위하여 pState_current[ctxIdx]가 사용되고 그리고 나서 현재 빈 값(각각, 인코딩되거나 디코딩된)이 업데이트된다.
- [0313] 아래에 더 상세히 설명될 것과 같이, 이제 설명되는 실시 예들에 따르면, 인코더와 디코더는 서로 다른 방식들, 주로 낮은 복잡도, 및 고효율 방식을 운용하기 위하여 선택적으로 구현될 수 있다. 이는 주로 다음의 파이프 코딩(그리고 나서 낮은 복잡도 및 고효율 파이프 방식을 언급하는)과 관련하여 설명되나, 복잡도 계산 내용의 설명은 하나의 통상적인 컨텍스트-적응적 산술 인코더/디코더를 사용하는 실시 예와 같은, 엔트로피 인코딩/디코딩 엔진들의 다른 구현들에 전달될 수 있다.
- [0315] 아래에서 설명되는 실시 예들에 따라 두 엔트로피 코딩 방식 모두 다음을 공유한다:
- [0316] • 동일한 구문과 의미(각각 구문 요소 시퀀스(301 및 327)를 위하여)
- [0317] • 모든 구문 요소를 위한 동일한 이진화 전략(CABAC(컨텍스트-적응적 이진 산술 코딩, 이하 CABAC로 표기)를 위하여 현재 지정된 것과 같이)(즉, 이진화기들은 활성화되는 방식과 관계없이 운용될 수 있다)
- [0318] • 동일한 파이프 코드들의 사용(즉, 빈 인코더들/디코더들은 활성화되는 방식과 관계없이 운용될 수 있다)
- [0319] • 8 비트 확률 모델 초기화 값들의 사용(CABAC를 위하여 현재 지정된 16 비트 확률 모델 초기화 값들 대신에)
- [0321] 일반적으로 말하면, 각각의 빈을 위한 파이프 경로(312)의 선택의 복잡도와 간은, 복잡도를 처리에 있어서 낮은 복잡도-파이프는 고효율-파이프와 다르다.
- [0323] 예를 들면, 낮은 복잡도 방식은 다음의 제약 하에서 운용될 수 있다: 각각의 빈(binIdx)을 위하여, 정확하게 하

나의 확률 모델, 즉 하나의 컨텍스트 지수가 존재할 수 있다. 즉, 낮은 복잡도 파이프에서는 어떠한 컨텍스트 선택/적용이 제공되지 않는다. 그러나, 아래에 더 설명되는 것과 같이, 잔여 코딩을 위하여 사용되는 것과 같은 특정 구문 요소가 컨텍스트들을 사용하여 코딩될 수 있다. 게다가, 모든 확률 모델은 비-적응적인데, 즉, 모든 모델은 적절한 모델 확률들(조각 형태 및 조각 양자화 파라미터의 선택에 따라)로 각각의 조각의 시작에서 초기화될 수 있으며, 조각의 처리를 통하여 고정되어 유지될 수 있다. 예를 들면, 컨텍스트 모델링과 코딩 모두를 위하여, 8개의 서로 다른 파이프 코드들(310/322)과 상응하는 8개의 서로 다른 모델 확률이 제공될 수 있다. 잔여 코딩을 위한 특정 구문 요소들, 즉, 그 의미가 아래에 더 상세히 설명되는, `significance_coeff_flag` 및 `coeff_abs_level_greaterX(X=1,2)`은 (적어도) 예를 들면 4 구문 요소들의 그룹들이 동일한 확률 모델로 인코딩/디코딩되는 것과 같이 확률 모델들에 할당될 수 있다. 컨텍스트-적응적 가변 길이 코딩과 비교하여, 낮은 복잡도-파이프 방식은 대략 동일한 R-D 성능 및 동일한 처리율을 달성한다.

[0325] **고효율-파이프는 다음의 차이들과 함께 H.264의 CAVAC와 개념적으로 유사하게** 구성될 수 있다: 이진 산술 코딩(binary arithmetic coding, BAC)은 파이프 코딩에 의해(낮은 복잡도-파이프 경우에서와 동일하게) 대체된다. 각각의 확률 모델, 즉, 각각의 컨텍스트 지수는 파이프 지수 및 세분화 지수에 의해 표현될 수 있으며, 0...7 범위 내의 값들을 갖는 파이프 지수는 8개의 서로 다른 파이프 코드들의 모델 확률을 표현한다. 이러한 변화는 상태들의 내부 표현에만 영향을 미치고 상태 기계(즉, 확률 추정) 자체에는 영향을 미치지 않는다. 아래에 더 상세히 설명될 것과 같이, 확률 모델들의 초기화는 위에서 설명된 것과 같이 8 비트 초기화 값들을 사용할 수 있다. 구문 요소들 `coeff_abs_level_greaterX` ($X=1, 2$), `coeff_abs_level_minus3`, 및 `coeff_sign_flag` (이들의 의미는 아래의 설명으로부터 더 자명해질 것이다)의 후방 스캐닝(backward scanning)은 전방 스캔(예를 들면, 유의성 맵 코딩에서 사용되는)과 동일한 스캐닝 경로를 따라 실행될 수 있다. `coeff_abs_level_greaterX(X=1, 2)`의 코딩을 위한 컨텍스트 유도는 또한 단순화될 수 있다. CABAC와 비교하여, 제안된 고효율-파이프는 더 나은 처리능력에서 대략 동일한 R-D 성능을 달성한다.

[0327] 방금 언급된 방식들이 예를 들면, 이들이 서로 다른 방식으로 운용하는 것과 같이 앞서 언급된 컨텍스트-적응성 이진 산술 인코딩/디코딩 엔진을 제공함으로써 쉽게 발생된다는 것을 쉽게 알 수 있다.

[0329] 따라서, 본 발명의 첫 번째 양상에 따른 일 실시 예에 따르면, 데이터 스트림을 디코딩하기 위한 디코더는 도 11에 도시된 것과 같이 구성될 수 있다. 디코더는 비디오 데이터와 같은 미디어 데이터가 코딩되는 인터리브드 비트스트림(340)과 같은 데이터스트림(401)을 디코딩하기 위한 것이다. 디코더는 데이터 스트림(401)에 따라 낮은 복잡도 방식 또는 고효율 방식을 활성화하도록 구성되는 모드 스위치(400)를 포함한다. 이를 위하여, 데이터 스트림(401)은 활성화되는 하나인 낮은 복잡도 방식의 경우에 1의 이진 값을 가지며, 활성화되는 하나의 고효율 방식의 경우에 0의 이진 값을 갖는, 이진 구문 요소와 같은, 구문 요소를 포함할 수 있다. 명백하게, 이진 값과 코딩 방식 사이의 관련성은 전환될 수 있으며, 두 개의 가능한 값 이상을 갖는 비-이진 구문 요소가 또한 사용될 수 있다. 두 방식 사이의 실제 선택이 각각의 구문 요소 이전에 아직 분명하지 않기 때문에, 이러한 구문 요소는 예를 들면, 고정된 확률 추정 또는 확률 모델로 인코딩되거나 혹은 그대로, 즉, 바이패스 방식을 사용하여 데이터스트림(401) 내로 판독되는 데이터스트림(401)의 일부 리딩 헤더(leading header) 내에 포함될 수 있다.

[0331] 또한, 도 11의 디코더는 각각 데이터스트림(401) 내의 코드워드들을 심벌들의 부분적인 시퀀스들(321)로 전환하도록 구성되는 복수의 엔트로피 디코더(322)를 포함한다. 위에 설명된 것과 같이, 디-인터리버(de-interleaver, 404)는 한편으로는 디코더들(322)의 입력들 및 다른 한편으로는 데이터스트림(401)이 적용되는, 도 11의 디코더의 입력 사이에 연결될 수 있다. 또한 위에서 이미 설명된 것과 같이, 각각의 엔트로피 디코더(322)는 각각의 확률 간격과 관련될 수 있으며, 다양한 엔트로피 디코더들의 확률 간격들은 절대 심벌 값들보다는 더 가능성이 있는 심벌과 덜 가능성이 있는 심벌로 처리하는 엔트로피 디코더들(322)의 경우에 0 내지 1 또는 0 내지 0.5의 전체 확률 간격을 포함한다. 이러한 문제에 관한 상세한 설명은 위에서 설명되었다. 뒤에, 디코더들(322)의 수는 각각의 디코더에 할당되는 파이프 지수를 갖는 8이라고 가정되나, 다른 수도 또한 실현 가능하다. 또한, 다음에서 바람직하게는 `pipe_id` 0인, 이러한 코더들 중 어느 하나는 등-확률 통계를 갖는 빈들을 위하여 최적화되는데, 즉, 그것들의 빈 값은 동등하게 아마도 1과 0으로 가정한다. 이러한 디코더는 단지 빈들을 전달할 수 있

다. 각각의 인코더(310)는 동일하게 운용된다. 각각 선택기들(402 및 502)에 의해 가장 가능성 있는 빈 값, valMPS의 값에 따른 어떠한 빈 조각도 제거될 수 있다. 바꾸어 말하면, 각각의 부분 스트림의 엔트로피는 이미 최적이다.

[0333] 또한 도 11의 디코더는 복수의 엔트로피 디코더(322) 중 선택된 하나로부터 심벌들의 시퀀스(326)의 각각의 심벌을 검색하도록 구성되는 선택기(402)를 포함한다. 위에 설명된 것과 같이, 선택기(402)는 파라미터 할당기(316)와 선택기(318) 내로 분리될 수 있다. 디심벌라이저(314)는 구문 요소들의 시퀀스(327)를 획득하기 위하여 심벌들의 시퀀스(326)를 탈심벌화하도록 구성된다. 재구성자(404)는 구문 요소들(327)이 시퀀스를 기초로 하여 미디어 데이터(405)를 재구성하도록 구성된다. 선택기(402)는 화살표(406)에 의해 나타낸 것과 같이 낮은 복잡도 방식과 고효율 방식 중 활성화된 어느 하나에 따라 선택을 실행하도록 구성된다.

[0335] 위에서 이미 설명된 것과 같이, 재구성자(404)는 고정된 구문과 구문 요소들의 의미들 상에서 운용하는, 즉, 모드 스위치(400)에 의한 방식 선택과 관련하여 고정된, 예측 블록 기반 비디오 디코더의 일부일 수 있다. 즉, 재구성자(404)의 구성은 모드 스위치 능력으로 고생하지 않는다. 더 정확히 설명하면, 재구성자(404)는 모드 스위치(400)에 의해 제공되는 모드 스위치능력에 기인하는 구현 오버헤드를 증가시키지 않으며 적어도 잔여 데이터와 관련하여 기능성 및 예측 데이터는 스위치(400)에 의해 선택되는 방식과 관계없이 그대로 남는다. 그러나, 엔트로피 디코더들(322)과 관련하여 동일하게 적용한다. 이러한 모든 디코더(322)는 두 방식에서 재사용되며, 따라서 비록 도 11의 디코더가 두 방식, 낮은 복잡도와 고효율 방식에 호환되나 부가적인 구현 오버헤드는 존재하지 않는다.

[0337] 부가 양상으로서 도 11의 디코더는 하나의 방식 또는 나머지 방식에서 독립된 데이터스트림들 상에서만 운용할 수 있지 않다는 것을 이해하여야 한다. 오히려, 도 11의 디코더뿐만 아니라 데이터스트림(401)은 그에 알맞게 방식 선택을 잠금-루프 제어하기 위하여 디코더로부터 인코더로 피드백 채널의 사용과 함께 배터리 상태 등과 같은 외부 또는 환경 조건들에 따라 예를 들면, 디코딩 면에서 코딩 복잡도를 제어하기 위하여 비디오 또는 일부 오디오 피스(piece) 동안에서와 같은 미디어 데이터의 하나의 피스 동안에 두 방식 사이의 전환이 가능할 수 있는 것과 같이 구성될 수 있다.

[0339] 따라서 도 11의 디코더는 선택되는 낮은 복잡도 방식 또는 선택되는 고효율 방식의 경우에 있어서, 두 경우에 유사하게 운용된다. 재구성자(404)는 구문 요소들을 사용하여 재구성을 실행하고 일부 구문 구조 규정을 처리하거나 준수함으로써 미리 결정된 구문 요소 형태의 현재 구문 요소를 요구한다. 디심벌라이저(314)는 재구성자(404)에 의해 요구되는 구문 요소를 위한 유효한 이진화를 생산하기 위하여 다수의 빈을 요구한다. 명백하게, 이진 알파벳의 경우에, 디심벌라이저(314)에 의해 실행되는 이진화는 각각의 빈/심벌(326)을 재구성자(404)로 통과시키도록 감소되는데 그 이유는 이진 구문 요소가 현재 요구되었기 때문이다.

[0341] 그러나, 선택기(404)는 모드 스위치(400)에 의해 선택되는 방식에 관계없이 작용한다. 선택기(402)의 운용의 방식은 고효율 방식에서 더 복잡한 경향이 있고, 낮은 복잡도 방식에서 덜 복잡한 경향이 있다. 게다가, 다음의 설명은 덜 복잡한 방식에서 선택기(402)의 운용이 방식이 엔트로피 디코더들(322)로부터 일련의 심벌들을 검색하는데 엔트로피 디코더들(322) 중에서의 선택을 변경하는 비율을 감소시키는 경향이 있다. 바꾸어 말하면, 낮은 복잡도 방식에서, 즉시 일련의 심벌들이 복수의 엔트로피 디코더(322) 중에서 동일한 엔트로피 디코더로부터 검색되는 증가된 확률이 존재한다. 이는 차례로, 엔트로피 디코더들(322)로부터 심벌들의 빠른 검색을 허용한다. 고효율 방식에서, 차례로, 선택기(402)의 운용의 방식은 각각의 선택된 엔트로피 디코더(322)와 관련된 확률 간격이 선택기(402)에 의해 현재 검색되는 심벌의 실제 심벌 통계에 더 근접하게 맞는 엔트로피 디코더들(322) 중에서의 선택에 이르게 하는 경향이 있으며, 그렇게 함으로써 고효율 방식에 따라 각각의 데이터 스트림을 발생시킬 때 인코딩 면에서 더 나은 압축률을 생산한다.

[0343] 예를 들면, 두 방식에서 선택기(402)의 서로 다른 행동은 다음과 같이 실현될 수 있다. 예를 들면, 선택기(40

2)는 미리 결정된 심벌을 위하여, 활성화되는 고효율 방식의 경우에 이전에 검색된 심벌들에 따라, 그리고 활성화되는 낮은 복잡도의 경우에 심벌들의 시퀀스의 어떠한 이전에 검색된 심벌들과도 관계없이 복수의 엔트로피 디코더(322) 중에서의 선택을 실행하도록 구성될 수 있다. 심벌들의 시퀀스(326)의 이전에 검색된 심벌들의 의존성은 컨텍스트 적응성 및/또는 확률 적응성에 기인할 수 있다. 두 적응성은 선택기(402) 내의 낮은 복잡도 동안에 꺼질 수 있다.

[0345] 또 다른 실시 예에 따르면, 데이터스트림(401)은 조각들, 프레임들, 픽처들의 그룹, 프레임 시퀀스들 등과 같은 일련의 부들 내로 구조화될 수 있으며, 심벌들의 시퀀스의 각각의 심벌은 복수의 심벌 형태 중 각각의 하나와 관련될 수 있다. 이러한 경우에 있어서, 선택기(402)는 현재 부 내의 미리 결정된 심벌 형태의 심벌들을 위하여, 활성화되는 고효율 방식의 경우에 현재 부 내의 미리 결정된 심벌 형태의 심벌들의 시퀀스의 이전에 검색된 심벌들에 따라 선택을 변경하고, 활성화되는 낮은 복잡도 방식의 경우에 현재 부 내에서 일정한 선택을 남기도록 구성될 수 있다. 즉, 선택기(402)는 미리 결정된 심벌 형태를 위하여 엔트로피 디코더들(322) 중에서의 선택을 변경하도록 허용될 수 있으나, 이러한 변경들은 일련의 부들 사이의 전이 사이에서 발생하도록 제한된다. 이러한 측정에 의해, 실제 심벌 통계의 평가는 시간 인스턴스들을 거의 발생하지 않도록 제한되나, 코딩 복잡도는 대부분의 시간 내에서 감소된다.

[0347] 또한, 심벌들의 시퀀스(326)의 각각의 심벌은 복수의 심벌 형태 중 각각의 하나와 관련될 수 있으며, 선택기(402)는 미리 결정된 심벌 형태의 미리 결정된 심벌을 위하여, 심벌들의 시퀀스(326)의 이전에 검색된 심벌들에 따라 복수의 컨텍스트 중 어느 하나를 선택하고 활성화되는 고효율 방식의 경우에 미리 결정된 심벌에 따라 선택된 컨텍스트와 관련된 확률 모델의 업데이트와 함께 선택된 컨텍스트와 관련된 확률 모델에 따라 엔트로피 디코더들(322) 중에서의 선택을 실행하며, 심벌들의 시퀀스(326)의 이전에 검색된 심벌들에 따라 복수의 컨텍스트 중 어느 하나의 선택을 실행하고 활성화되는 낮은 복잡도 방식의 경우에 선택된 컨텍스트 상수와 관련된 확률 모델의 남김(leaving)과 함께 선택된 컨텍스트와 관련된 확률 모델에 따라 엔트로피 디코더들(322) 중에서의 선택을 실행하도록 구성될 수 있다. 즉, 선택기(402)는 두 방식에서 특정 구문 요소 형태와 관련하여 컨텍스트 적응을 사용할 수 있으나, 낮은 복잡도 방식의 경우에 확률 적응성을 억제할 수 있다.

[0349] 대안으로서, 확률 적응성을 완전히 억제하는 대신에, 선택기(402)는 단지 고효율 방식에 관하여 낮은 복잡도 방식의 확률 적응성의 업데이트 비율을 감소시킬 수 있다.

[0351] 또한, 낮은 복잡도-파이프-특정 양상들, 즉, 낮은 복잡도 방식들의 양상들은 바꾸어 말하면 다음과 같이 설명될 수 있다. 특히, 비-적응적 확률 모델들이 낮은 복잡도 방식에서 사용될 수 있다. 비-적응적 확률 모델은 하드코딩된(hardcoded), 즉, 전체 일정한 확률을 가질 수 있거나, 또는 그것의 확률은 조각만의 처리를 통하여 고정되며 따라서 조각 형태 또는 양자화 파라미터, 즉, 예를 들면, 각각의 조각을 위하여 데이터스트림(401) 내에 신호가 전달되는 양자화 파라미터에 의존하여 설정될 수 있다. 동일한 bin들에 할당되는 연속적인 bin들이 고정된 확률 모델을 따른다고 가정함으로써, 한 단계로 그러한 bin들의 일부를 디코딩하는 것이 가능한데 그 이유는 그것들이 동일한 파이프 코드를 사용하여, 즉, 동일한 엔트로피 디코더를 사용하여 인코딩되고, 각각 bin을 디코딩한 후에 확률 업데이트가 생략되기 때문이다. 확률 업데이트들의 생략은 인코딩과 디코딩 과정 동안의 운용들을 절약하며, 따라서 또한 복잡도 감소와 하드웨어 디자인에서의 유의성 단순화에 이르게 한다.

[0353] 비-적응적 제한은 bin들의 특정 수가 이러한 모델을 사용하여 인코딩/디코딩된 후에 확률 업데이트들이 허용되는 것과 같은 방법으로 모든 또는 일부 선택된 확률 모델을 위하여 덜해질 수 있다. 적절한 업데이트 간격은 확률 적응을 허용하며 일부 bin들을 즉시 디코딩하는 능력을 갖는다.

[0355] 다음에서, 낮은 복잡도-파이프 및 고효율-파이프의 가능한 통상적이고 복잡도-확정성 양상들의 더 상세한 설명이 제공된다. 특히, 다음에서, 동일한 방법으로 또는 복잡도-확장성(scalable) 방법으로 낮은 복잡도-파이프 방식과 고효율-파이프 방식을 위하여 사용될 수 있는 양상들이 설명된다. 복잡도-확장성은 특정 부분을 제거하

나 또는 그것들을 덜 복잡한 것으로 대체함으로써 낮은 복잡도 경우가 고효율 경우로부터 유도되는 것을 의미한다. 그러나, 그것들과 함께 처리하기 전에, 도 11의 실시 예는 위에서 설명된 컨텍스트 적응성 이진 산술 인코딩/디코딩 실시 예 상으로 쉽게 전달될 수 있다는 것을 이해하여야 한다: 선택기(402)와 엔트로피 디코더들(322)는 데이터스트림(401)을 직접적으로 수신하고 데이터스트림으로부터 현재 유도되는 bin을 위한 컨텍스트를 선택할 수 있는 컨텍스트 적응성 이진 산술 디코더 내로 압축될 수 있다. 이는 특히 컨텍스트 적응성 및/또는 확률 적응성에 있어서 사실이다. 두 기능들/적응성들은 낮은 복잡도 방식 동안에 끼질 수 있거나, 또는 더 느긋하게 디자인될 수 있다.

[0357]

예를 들면, 도 11의 실시 예의 구현에서, 엔트로피 디코더들(322)을 포함하는 파이프 엔진 엔트로피 코딩 단계는 8개의 체계적인 가변-대-가변-코드들을 사용할 수 있는데, 즉, 각각의 엔트로피 디코더(322)는 위에서 설명된 가변-대-가변 형태일 수 있다. 체계적인 가변-대-가변 코드들을 사용하는 파이프 코딩 개념은 가변-대-가변 코드들의 수를 한정함으로써 단순화된다. 컨텍스트-적응적 이진 산술 디코더의 경우에 있어서, 확률 세분화를 위하여 가변-대-가변 코드들은 서로 다른 컨텍스트들을 위하여 동일한 확률 상태들을 관리할 수 있으며 서로 다른 컨텍스트를 위하여 이를(또는 그것의 양자화된 버전) 사용할 수 있다. CABAC 또는 확률 모델 상태들, 즉, 확률 업데이트를 위하여 사용되는 상태들의 파이프 지수들 또는 R탭 내의 록-업을 위한 확률 지수들로의 매핑이 테이블 A와 같이 도시될 수 있다.

CABAC 상태	파이프 지수	CABAC 상태	파이프 지수
0	0	32	5
1		33	
2		34	
3	1	35	
4		36	
5		37	
6		38	
7		39	
8		40	
9		41	
10	2	42	
11		43	
12		44	
13		45	
14		46	
15	3	47	6
16		48	
17		49	
18		50	
19		51	
20		52	
21		53	
22	4	54	
23		55	
24		56	
25		57	
26		58	
27		59	
28		60	
29		61	
30		62	
31			7

[0359]

[0360]

테이블 A: CABAC 상태들의 파이프 지수들로의 매핑

[0362]

이러한 변형된 전략은 복잡도-확정성 비디오 코딩 접근법을 위한 기초로서 사용될 수 있다. 확률 방식 적용을 실행할 때, 선택기(402) 또는 컨텍스트-적응성 이진 산술 디코더는 각각 파이프 디코더(322)를 선택할 수 있는데, 즉, 사용되는 파이프 지수 및 테이블 A에 도시된 매핑을 사용하여 현재 디코딩되려는 심볼과 관련된(컨텍스트를 통한 것과 같이) 확률 상태 지수(여기서는 바람직하게는 0 내지 62 범위의)를 기초로 하여, 각각 R탭 내의

확률 지수를 유도할 수 있으며, 예를 들면, 각각 더 가능성이 있는 심벌 및 덜 가능성이 있는 심벌의 경우에 방문되는 그 다음의 확률 상태 지수를 나타내는 특정한 테이블 워크(walk) 전이 값들을 사용하는 현재 디코딩되는 심벌에 따라 이러한 확률 상태 지수를 업데이트할 수 있다. 낮은 복잡도 방식의 경우에 있어서, 후자의 업데이트는 제거될 수 있다. 전체적으로 고정된 확률 모델들이 경우에 매핑도 없어질 수 있다.

[0364] 그러나, 임의 엔트로피 코딩 설정이 사용될 수 있으며 작은 적용과 함께 본 발명에서의 기술이 또한 사용될 수 있다.

[0366] 도 11의 위의 설명은 일반적으로 구문 요소들과 구문 요소 형태들을 언급하였다. 다음에서, 변환 계수 레벨들의 복잡도 구성가능 코딩이 설명된다.

[0368] 예를 들면, 재구성자(404)는 활성화되는 고효율 방식 또는 낮은 복잡도 방식과 관계없이 구문 요소들의 시퀀스의 일부를 기초로 하여 변환 계수 레벨들(202)의 변환 블록(200)을 재구성하도록 구성될 수 있으며, 구문 요소들의 시퀀스(327)의 일부는 비-인터리브드 방식으로, 변환 블록(200) 내의 비-제로 변환 계수 레벨들의 위치들을 나타내는 유의성 맵을 정의하며, 그 다음에 비-제로 변환 계수 레벨들을 정의하는 레벨 구문 요소들이 뒤따른다. 특히, 다음의 요소들이 포함될 수 있다: 변환 블록 내의 마지막 비-제로 변환 계수 레벨의 위치를 나타내는 단부 위치 구문 요소들(last_significant_pos_x, last_significant_pos_y); 함께 유의성 맵을 정의하고 DC 위치로부터 변환 블록(200) 내의 마지막 비-제로 변환 계수 레벨에 이르게 하는 1차원 경로(274)를 따라 각각의 위치를 위하여, 각각의 위치에서 변환 계수 레벨이 비-제로인지 아닌지에 대하여 나타내는 제 1 구문 요소들(coeff_significant_flag); 제 1 이진 구문 요소들에 따라 비-제로 변환 계수 레벨이 위치되는 1차원 경로의 각각의 위치를 위하여, 각각의 위치에서 변환 계수 레벨이 1보다 큰지에 대하여 나타내는 제 2 구문 요소들(coeff_abs_greater1); 및 제 1 이진 구문 요소들에 따라 1보다 큰 변환 계수 레벨이 위치되는 1차원 경로의 각각의 위치를 위하여, 각각의 위치에서 각각의 변환 계수 레벨이 1을 초과하는 양을 나타내는 제 3 구문 요소들(coeff_abs_greater_2).

[0370] 단부 위치 구문 요소들, 제 1, 제 2 및 제 3 구문 위치들 중에서의 순차는 고효율 방식과 낮은 복잡도 방식을 위하여 동일하며, 선택기(402)는 디심벌라이저가 끝 부분 위치 구문 요소들, 제 1, 제 2 및/또는 제 3 구문 위치들을 획득하는 심벌들을 위하여 엔트로피 디코더들(322) 중에서의 선택을 실행하도록 구성될 수 있다.

[0372] 특히, 선택기(402)는 디심벌라이저(314)가 제 1 구문 요소들과 제 2 구문 요소들을 획득하는 심벌들의 서브시퀀스 중에서 미리 결정된 심벌 형태의 심벌들을 위하여, 심벌들의 서브시퀀스 중에서 미리 결정된 심벌 형태의 이전에 검색된 심벌들에 따라 복수의 컨텍스트 중 하나를 미리 결정된 심벌 형태의 각각의 심벌을 위하여 선택하고, 활성화되는 고효율 방식의 경우에 선택된 컨텍스트와 관련된 확률 모델에 따라 선택을 실행하며, 활성화되는 낮은 복잡도의 경우에 서브시퀀스의 연이은 지속적인 서브파트에 대하여 선택이 일정한 것과 같이 부분 상수(piece wise constant) 방식으로 선택을 실행하도록 구성될 수 있다. 위에 설명된 것과 같이, 서브파트들은 1차원 경로(274)를 따라 측정될 때 각각의 서브파트가 확장하는 이치들의 수, 또는 현재 컨텍스트 내에 이미 코딩된 각각의 형태의 구문 요소들의 수로 측정될 수 있다. 즉, 이진 구문 요소들, 예를 들면, coeff_significant_flag, coeff_abs_greater1 및 coeff_abs_greater2는 고효율 방식에서 선택된 컨텍스트의 확률 모델을 기초로 하여 디코더(322)의 선택과 함께 적응적으로 코딩된다. 확률 적용이 또한 사용된다. 낮은 복잡도 방식에서, 각각의 이진 구문 요소들 coeff_significant_flag, coeff_abs_greater1 및 coeff_abs_greater2를 위하여 사용되는 서로 다른 컨텍스트들이 또한 존재한다. 그러나, 각각의 구문 요소를 위하여, 컨텍스트는 그 다음의 전이에서 컨텍스트의 변화와 함께 경로(274)를 따라 제 1 부를 위하여 고정되며, 경로(274)를 따라 바로 따른다. 예를 들면, 각각의 부는 각각의 위치를 위하여 각각의 구문 요소가 존재하는지와 관계없이 블록(200) 길이의 4, 8, 16 위치들로 정의될 수 있다. 예를 들면, coeff_abs_greater1 및 coeff_abs_greater2는 유효 위치들, 즉, coeff_significant_flag가 1인 위치들을 위하여 존재한다. 대안으로서, 각각의 위치는 따라서 결과로서 생기는 각각의 부가 블록들의 높은 수를 넘어 확장하는지와 관계없이, 4, 8, 16 구문 요소들 길이로 정의될 수 있다. 예를 들면, coeff_abs_greater1 및 coeff_abs_greater2는 단지 유의 위치들을 위하여 존재하

며, 따라서 4개의 구문 요소들의 부들은 이 위치에서 각각의 레벨 때문에 *coeff_abs_greater1* 및 *coeff_abs_greater2* 중 어느 것도 제로가 아닌 것과 같이 그러한 어떠한 구문 요소도 전송되지 않는 경로(274)를 따라 그것들 사이의 위치 때문에 각각 4 블록 위치들 이상을 넘어 확장할 수 있다.

[0374] 선택기(402)는 디심벌라이저가 제 1 구문 요소들과 제 2 구문 요소들을 획득하는 심벌들의 서브시퀀스 중에서 미리 결정된 심벌 형태의 심벌들을 위하여, 미리 결정된 심벌 값을 갖고 동일한 서브파트에 속하는, 심벌들의 시퀀스 내의 미리 결정된 심벌 형태의 이전에 검색된 심벌들의 수, 또는 동일한 서브파트에 속하는, 심벌들의 시퀀스 내의 미리 결정된 심벌 형태의 이전에 검색된 심벌들의 수에 따라 복수의 컨텍스트 중 어느 하나를 미리 결정된 심벌 형태의 각각의 심벌을 위하여 선택하도록 구성될 수 있다. 위의 특정 실시 예들에 따라 첫 번째 대안은 *coeff_abs_greater1*에 있어서 사실이었으며 이차 대안은 *coeff_abs_greater2*에 있어서 사실이었다.

[0376] 또한, 제 1 이진 구문 요소들에 따라, 1보다 큰 변환 계수 레벨이 위치되는 1차원 경로의 각각의 위치를 위하여, 각각의 위치에서 각각의 변환 계수 레벨이 1을 초과하는 양을 나타내는 제 3 구문 요소들은 정수 값의 구문 요소들, 즉 *coeff_abs_minus3*을 포함할 수 있으며, 디심벌라이저(314)는 심벌 시퀀스 단어들을 정수 값의 구문 요소들의 공-도메인에 매핑하기 위하여 제어 파라미터에 의해 제어가능한 매핑 기능을 사용하고, 만일 고효율 방식이 활성화되면 이전의 제 3 구문 요소의 정수 값의 구문 요소들에 따라 정수 값이 구문 요소 당 제어 파라미터를 설정하며, 활성화되는 낮은 복잡도의 경우에 서브시퀀스의 일련의 연속적인 서브파트들에 대하여 설정이 일정한 것과 같이 부분 상수방식으로 설정을 실행하도록 구성될 수 있으며, 선택기(402)는 두 고효율 방식과 낮은 복잡도 방식에서, 동일한 확률 분포와 관련된, 정수 값이 구문 요소들 상으로 매핑된 심벌 시퀀스 단어들이 심벌들을 위하여 엔트로피 디코더들(322) 중 미리 결정된 어느 하나를 선택하도록 구성될 수 있다. 즉, 디심벌라이저가 선택된 방식에 따라 운용될 수 있더라도, 스위치(400)가 점선(407)에 의해 도시된다. 제어 파라미터의 부분 상수 설정 대신에, 디심벌라이저는 예를 들면 현재 조각 동안에 제어 파라미터를 일정하게 유지할 수 있거나 또는 시간에 맞게 포괄적으로 일정하게 유지할 수 있다.

[0378] 그 다음에, 복잡도-확장성 컨텍스트 모델링이 설명된다.

[0380] 컨텍스트 모델 지수의 유도를 위한 상부와 왼쪽 이웃의 동일한 구문 요소의 평가는 통상의 접근법이며 예를 들면, 움직임 벡터 차분값 구문 요소를 위하여 고효율의 경우에 흔히 사용된다. 그러나, 이러한 평가는 더 나은 저장을 필요로 하며 구문 요소의 직접적인 코딩을 허용하지 않는다. 또한, 높은 코딩 성능을 달성하기 위하여, 더 많은 이용가능한 이웃들이 평가될 수 있다.

[0382] 바람직한 실시 예에서, 이웃 정사각형 또는 직사각형 블록들 혹은 예측 유닛들의 구문 요소들을 평가하는 모든 컨텍스트 단계는 하나의 컨텍스트 모델에 고정된다. 이는 컨텍스트 모델 선택 단계에서의 적응성의 불능화와 동일하다. 그러한 바람직한 실시 예를 위하여, 이진화 후에 빈 스트링의 빈 지수에 따른 컨텍스트 모델 선택은 CABAC를 위한 현재 디자인과 비교하여 변형되지 않는다. 또 다른 바람직한 실시 예에서, 구문 요소들을 위한 고정된 컨텍스트 모델에 더하여, 또한 서로 다른 빈 지수를 위한 컨텍스트 모델이 고정된다. 본 설명은 변환 계수 레벨들의 코딩과 관련된 움직임 벡터 차분값과 구문 요소들을 위한 이진화 및 컨텍스트 모델 선택을 포함하지 않는다는 것을 이해하여야 한다.

[0384] 바람직한 실시 예에서, 왼쪽 이웃의 평가만이 허용된다. 이는 처리 체인(chain)에서 감소된 버퍼에 이르게 하는데 그 이유는 마지막 블록 또는 코딩 유닛 라인이 더 이상 저장되어서는 안 되기 때문이다. 또 다른 바람직한 실시 예에서, 동일한 코딩 유닛에 놓인 이웃들만이 평가된다.

[0386] 바람직한 실시 예에서, 모든 이용가능한 이웃이 평가된다. 예를 들면, 상부와 왼쪽 이웃에 더하여, 가용성의 경우에 상부 왼쪽, 상부 오른쪽, 및 하부 왼쪽 이웃이 평가된다.

[0388] 즉, 도 11의 선택기(402)는 미디어 데이터의 미리 결정된 블록과 관련하여 미리 결정된 심벌을 위하여, 복수의 컨텍스트 중 어느 하나를 선택하고 선택된 컨텍스트와 관련된 확률 모델에 따라 엔트로피 디코더들(322) 중에서의 선택을 실행하기 위하여 활성화된 고효율 방식의 경우에 미디어 데이터의 서로 다른 이웃하는 블록들의 높은 수와 관련하여 심벌들의 시퀀스의 이전에 검색된 심벌들을 사용하도록 구성될 수 있다. 즉, 이웃하는 블록들은 시간들 및/또는 공간 도메인으로 이웃할 수 있다. 예를 들면, 도 1 내지 3에 공간적으로 이웃하는 블록들을 볼 수 있다. 그때, 선택기(402)는 낮은 복잡도 방식과 비교하여 고효율 방식의 경우에서 이웃하는 블록들의 높은 수와 관련하여 이전에 검색된 심벌들 또는 구문 요소들을 기초로 하여 접촉 적용을 실행하기 위하여 모드 스위치(400)에 의해 방식 선택에 반응할 수 있으며 그렇게 함으로써 방금 설명된 것과 같은 저장 오버헤드를 감소시킨다.

[0390] 그 다음에, 일 실시 예에 따른 움직임 벡터 차분값들의 감소된 복잡도 코딩이 설명된다.

[0392] H.264/AVC 비디오 코덱 표준에서, 매크로블록과 관련된 움직임 벡터는 현재 매크로블록의 움직임 벡터와 중간 움직임 벡터 예측변수 사이의 차이(움직임 벡터 차분값, mvd)의 차이의 시그널링에 의해 전송된다. 엔트로피 코더로서 CABAC가 사용될 때, 움직임 벡터 차이는 다음과 같이 코딩된다. 정수 값의 움직임 벡터 차분값는 절대 및 부호 부분으로 나뉜다. 절대 부분은 결과로서 생기는 빈 스트링의 접두부호와 접미부호로서 언급되는, 절단된 단항 및 3차 지수-골롬의 조합을 사용하여 이진화된다. 절단된 단항 이진화와 관련된 빈들은 컨텍스트 모델들을 사용하여 코딩되며, 지수-골롬 이진화와 관련된 빈들은 바이패스 방식에서, 즉, CABAC를 갖는 0.5의 고정 확률로 코딩된다. 단항 이진화는 다음과 같이 작동한다. 움직임 벡터 차분값의 절대 정수 값을 n 이라 하고, 그리고 나서 결과로서 생기는 빈 스트링은 n 번의 '1' 및 하나의 트레일링 '0'으로 구성된다., 예를 들면, $n=4$ 일 때, 빈 스트링은 '11110'이다. 절단된 단항 경우에 있어서, 한계가 존재하며 만일 값이 이러한 한계를 초과하면, 빈 스트링은 $n+1$ 번의 '1'로 구성된다. 움직임 벡터 차이의 경우를 위하여, 한계는 9와 동일하다. 이는 만일 움직임 벡터 차분값가 9번의 '1'을 야기하는 9와 동일하거나 또는 9보다 큰 절대값이 코딩되면, 빈 스트링은 지수-골롬 이진화를 갖는 접두부호 및 접미부호로 구성된다는 것을 의미한다. 절단된 단항 부분을 위한 컨텍스트 모델링은 다음과 같이 수행된다. 빈 스트링의 첫 번째 빈을 위하여, 만일 이용가능하면 상부 및 왼쪽 매크로블록들로부터 절대 움직임 벡터 차이 값들이 취해진다(만일 이용할 수 없으며, 값은 0인 것으로 추론된다). 만일 특정 컴포넌트(수평 또는 수직방향)를 위한 합이 2보다 크면, 두 번째 컨텍스트 모델이 선택되며, 만일 절대 합이 32보다 크면, 세 번째 컨텍스트 모델이 선택되며, 그렇지 않으면(절대 합이 3보다 작으면) 첫 번째 컨텍스트 모델이 선택된다. 게다가, 컨텍스트 모델들은 각각의 컴포넌트를 위하여 서로 다르다. 빈 스트링의 두 번째 빈을 위하여, 4번째 컨텍스트 모델이 사용되고 단항 부분들의 나머지 빈들을 위하여 5번째 컨텍스트 모델이 사용된다. 절대 움직임 벡터 차분값가 9와 동일하거나 이보다 클 때, 예를 들면, 절단된 단항 부분의 모든 빈은 '1'과 동일하며, 절대 움직임 벡터 차와 9 사이의 차이는 3차 지수-골롬 이진화를 갖는 바이패스 방식으로 코딩된다. 마지막 단계에서, 움직임 벡터 차분값의 부호는 바이패스 방식으로 코딩된다.

[0394] 엔트로피 코더로서 CABAC를 사용할 때 움직임 벡터 차이를 위한 최신 코딩 기술은 고효율 비디오 코딩((HEVC) 프로젝트의 현재 테스트 모델(TM)에서 지정된다. 고효율 비디오 코딩에서, 블록 크기들은 다양하며 움직임 벡터에 의해 지정되는 형태는 예측 유닛(prediction unit, PU)으로서 언급된다. 상부 왼쪽 이웃의 예측 유닛 크기는 현재 예측 유닛과 다른 형태와 크기를 가질 수 있다. 따라서, 관련될 때마다, 상부와 왼쪽 이웃의 정의는 이제 현재 예측 유닛의 상부-왼쪽 노서리의 상부와 왼쪽 이웃으로서 언급된다. 코딩 자체를 위하여, 일 실시 예에 따라 첫 번째 빈을 위한 유도 과정만이 변경될 수 있다. 이웃들로부터의 움직임 벡터의 절대 합의 평가 대신에, 각각의 이웃이 개별적으로 평가될 수 있다. 만일 이웃의 절대 움직임 벡터가 이용가능하고 16보다 크면, 첫 번째 빈을 위한 컨텍스트 모델 지수는 동일한 수의 컨텍스트 모델을 야기하여 증가될 수 있으며, 나머지 절대 움직임 벡터 차분값 레벨과 부호의 코딩은 정확하게 H.264/AVC와 동일하다.

[0396] 움직임 벡터 차분값의 코딩에 대하여 위에서 설명된 기술에 있어서, 컨텍스트 모델로 9 빈들까지 코딩되어야 하며, 움직임 벡터 차분값의 나머지 값은 부호 정보와 함께 낮은 복잡도의 바이패스 방식으로 코딩될 수 있다. 이

러한 본 발명의 실시 예는 증가된 수의 바이패스를 야기하는 컨텍스트 모델들의 수를 감소시키고 움직임 벡터 차분값의 코딩을 위하여 필요한 컨텍스트 모델들의 수를 감소시키기 위한 기술을 설명한다. 이를 위하여, 절단 값이 9에서 1 또는 2로 감소된다. 이는 절대 움직임 벡터 차분값이 0보다 큰지를 지정하는 첫 번째 빈만이 컨텍스트 모델을 사용하여 코딩되거나 또는 절대 움직임 벡터 차분값이 0과 1보다 큰지를 지정하는 첫 번째 및 두 번째 빈이 컨텍스트 모델을 사용하여 코딩되며, 나머지 값은 바이패스 방식으로 및/또는 가변 길이 코딩 코드를 사용하여 코딩되는 것을 의미한다. 가변 길이 코딩 코드를 사용하는(단항 또는 절단된 단항 코드를 사용하지 않는) 이진화로부터 야기되는 모든 빈은 낮은 복잡도의 바이패스 방식을 사용하여 코딩된다. 파이프의 경우에 있어서, 비트스트림 내로의, 그리고 비트스트림으로부터의 직접적인 삽입이 가능하다. 게다가, 만약 있다면, 첫 번째 빈을 위한 더 나은 컨텍스트 모델 선택을 유도하기 위하여 상부와 왼쪽 이웃의 서로 다른 정의가 사용될 수 있다.

[0398]

바람직한 실시 예에서, 지수-골롬 코드들은 절대 움직임 벡터 차분값 컴포넌트들의 나머지 부분을 이진화하도록 사용된다. 이를 위하여, 지수-골롬 코드의 순차는 가변적이다. 지수-골롬 코드의 순차는 다음과 같이 유도된다. 첫 번째 빈을 위한 컨텍스트 모델, 및 따라서 그러한 컨텍스트 모델이 지수가 유도되고 코딩된 후에, 지수-골롬 이진화 부분을 위한 순차로서 지수가 사용된다. 이러한 바람직한 실시 예에서, 첫 번째 빈을 위한 컨텍스트 모델은 이는 지수-골롬 코드의 순차로서 사용되는, 지수 0-2를 야기하는 1-3 범위 내에 존재하며, 이는 지수-골롬 코드의 순차로서 사용될 수 있다. 이러한 바람직한 실시 예는 고효율 경우를 위하여 사용될 수 있다.

[0400]

절대 움직임 벡터 차분값의 코딩에서 5 컨텍스트의 두 배를 사용하는 위에서 설명된 기술에 대한 대안에 있어서, 9 단항 코드 이진화 빈들을 코딩하기 위하여, 14 컨텍스트 모델들(각각의 컴포넌트를 위하여 7)이 또한 사용될 수 있다. 예를 들면, 단항 부분의 첫 번째와 두 번째 빈은 이전에 설명된 것과 같이 4개의 서로 다른 컨텍스트로 코딩될 수 있고, 세 번째 빈을 위하여 5번째 컨텍스트가 사용될 수 있으며, 4번째 빈과 관련하여 6번째 컨텍스트가 사용될 수 있으며, 5번째부터 9번째 빈은 7번째 컨텍스트를 사용하여 코딩된다. 따라서, 이러한 경우에 있어서 14개의 컨텍스트가 필요할 수 있으며, 낮은 복잡도의 바이패스 방식으로 나머지 값만이 코딩될 수 있다. 증가된 수의 바이패스를 야기하는 컨텍스트 모델들로 코딩되는 빈들의 수를 감소시키고 움직임 벡터 차분값의 코딩을 위하여 필요한 컨텍스트 모델들의 수를 감소시키는 기술은 예를 들면, 9로부터 1 또는 2로와 같이, 절단 값을 감소시키는 것이다. 이는 절대 움직임 벡터 차분값이 제로보다 큰지를 지정하는 첫 번째 빈만이 컨텍스트 모델을 사용하여 코딩될 수 있거나 혹은 절대 움직임 벡터 차분값이 제로와 1보다 큰지를 지정하는 첫 번째 및 두 번째 빈이 각각의 컨텍스트 모델을 사용하여 코딩될 수 있으며, 나머지 값은 가변 길이 코딩 코드로 코딩된다는 것을 의미한다. 가변 길이 코딩 코드를 사용하는 이진화를 야기하는 모든 빈은 낮은 복잡도의 바이패스 방식을 사용하여 코딩된다. 파이프의 경우에 있어서, 비트스트림 내로의, 그리고 비트스트림으로부터의 직접적인 삽입이 가능하다. 게다가, 설명된 실시 예는 첫 번째 빈을 위한 더 나은 컨텍스트 모델 선택을 유도하기 위하여 상부 및 왼쪽 이웃의 또 다른 정의를 사용한다. 이에 더하여, 컨텍스트 모델링은 첫 번째 또는 첫 번째와 두 번째 빈을 위하여 필요한 컨텍스트 모델들이 수가 감소되어 또 다른 메모리 감소에 이르게 하는 방법으로 변형된다. 또한, 위의 이웃들과 같은 이웃들의 평가는 이웃들의 움직임 벡터 차분값 값들의 저장을 위하여 필요한 라인 버퍼/메모리의 절약을 야기할 수 없다. 마지막으로, 컴포넌트들의 코딩 순차는 두 컴포넌트들을 위한 점두부호 빈들(즉, 컨텍스트 모델들로 코딩되는 빈들)의 코딩을 허용하고 그 뒤에 바이패스 빈들의 코딩을 허용하는 방식으로 분할될 수 있다.

[0402]

바람직한 실시 예에서, 절대 움직임 벡터 차분값 컴포넌트들의 나머지 부분을 이진화하기 위하여 지수-골롬 코드들이 사용된다. 이를 위하여, 지수-골롬 코드의 순차는 가변적이다. 지수-골롬 코드의 순차는 다음과 같이 유도될 수 있다. 첫 번째 빈을 위한 컨텍스트 모델, 따라서 그러한 컨텍스트 모델이 지수가 유도된 후에, 지수는 지수-골롬 이진화를 위한 순차로서 사용된다. 이러한 바람직한 실시 예에서, 첫 번째 빈을 위한 컨텍스트 모델은 지수-골롬 코드의 순차로서 사용될 수 있을, 지수 0-2를 야기하는 1-3의 범위 내에 존재한다. 이러한 바람직한 실시 예는 고효율 경우를 위하여 사용될 수 있으며 컨텍스트 모델이 수는 6으로 감소된다. 다시 컨텍스트 모델들의 수를 감소시키기 위하여, 따라서 메모리를 절약하기 위하여, 또 다른 바람직한 실시 예에서 수평 및 수직 컴포넌트들이 동일한 컨텍스트 모델을 공유할 수 있다. 그러한 경우에 있어서, 3개의 컨텍스트 모델만이 필요하다. 게다가, 본 발명의 또 다른 바람직한 실시 예에서의 평가를 위하여 왼쪽 이웃만이 고려될 수

있다. 이러한 바람직한 실시 예에서, 임계값은 변형되지 않을 수 있다(예를 들면, 0 또는 1의 지수-골롬 파라미터를 야기하는 16의 단일 임계값 혹은 0 또는 2의 지수-골롬 파라미터를 야기하는 32의 단일 임계값). 이러한 바람직한 실시 예는 움직임 벡터 차분값의 저장을 위하여 필요한 라인 버퍼를 절약한다. 그러한 바람직한 실시 예를 위하여, 움직임 벡터 차분값의 코딩을 위하여 총 3개의 컨텍스트 모델이 필요하며 가능한 지수-골롬 파라미터는 0-2 범위에 존재한다. 또 다른 바람직한 실시 예에서, 임계값은 16 및 32와 동일하다. 다시, 설명된 실시 예는 고효율 경우에 적합하다.

[0404] 본 발명의 또 다른 바람직한 실시 예에서, 절단 값은 9로부터 2로 감소된다. 이러한 바람직한 실시 예에서, 첫 번째 빈과 두 번째 빈은 컨텍스트 모델들을 사용하여 코딩될 수 있다. 첫 번째 빈을 위한 컨텍스트 모델 선택은 최신 또는 위의 바람직한 실시 예에서 설명된 변형된 방법으로 수행될 수 있다. 두 번째 빈을 위하여, 최신 기술로서 개별 컨텍스트 모델이 선택된다. 또 다른 바람직한 실시 예에서, 두 번째 빈을 위한 컨텍스트 모델은 왼쪽 이웃의 움직임 벡터 차분값을 평가함으로써 선택된다. 이러한 경우를 위하여, 컨텍스트 모델 지수는 첫 번째 빈을 위한 것과 동일하며, 이용가능한 컨텍스트 모델들은 첫 번째 빈을 위한 것과 다르다. 통틀어 6개의 컨텍스트 모델이 필요하다(컴포넌트들이 컨텍스트 모델들을 공유한다는 것에 유의하여야 한다). 다시, 지수-골롬 파라미터는 첫 번째 빈의 선택된 컨텍스트 모델 지수에 의존할 수 있다. 본 발명의 또 다른 바람직한 실시 예에서, 지수-골롬 파라미터는 두 번째 빈의 컨텍스트 모델 지수에 의존한다. 본 발명의 설명된 실시 예들은 고효율 경우를 위하여 사용될 수 있다.

[0406] 본 발명의 또 다른 바람직한 실시 예에서, 두 빈을 위한 컨텍스트 모델들은 고정되고 왼쪽 또는 위의 이웃들을 평가함으로써 유도되지 않는다. 이러한 바람직한 실시 예를 위하여, 컨텍스트 모델의 총 수는 2와 동일하다. 본 발명의 또 다른 바람직한 실시 예에서, 첫 번째 빈과 두 번째 빈은 동일한 컨텍스트 모델을 공유한다. 그 결과, 움직임 벡터 차분값의 코딩을 위하여 하나의 컨텍스트 모델만이 필요하다. 본 발명의 두 바람직한 실시 예들에서, 지수-골롬 파라미터는 고정되고 1과 동일하다. 설명된 본 발명의 바람직한 실시 예는 두 고효율 및 낮은 복잡도 구성에 적합하다.

[0408] 또 다른 바람직한 실시 예에서, 지수-골롬 부분의 순차는 첫 번째 빈의 컨텍스트 모델 지수와 관계없이 유도된다. 이러한 경우에 있어서, 지수-골롬 부분을 위한 순차를 유도하기 위하여 H.264/AVC의 통상의 컨텍스트 모델 선택의 절대 합이 사용된다. 이러한 바람직한 실시 예는 고효율 경우를 위하여 사용될 수 있다.

[0410] 또 다른 바람직한 실시 예에서, 지수-골롬 코드들의 순차는 고정되고 0으로 설정된다. 또 다른 바람직한 실시 예에서, 지수-골롬 코드들의 순차는 고정되고 1로 설정된다. 바람직한 실시 예에서, 지수-골롬 코드들의 순차는 2로 고정된다. 또 다른 바람직한 실시 예에서, 지수-골롬 코드들의 순차는 3으로 고정된다. 또 다른 바람직한 실시 예에서, 지수-골롬 코드들의 순차는 현재 예측 유닛의 형태와 크기에 따라 고정된다. 설명된 바람직한 실시 예들은 낮은 복잡도 경우를 위하여 사용될 수 있다. 지수-골롬 부분의 고정된 순차는 컨텍스트 모델들로 코딩된 빈들의 감소된 수로 고려된다는 것을 이해하여야 한다.

[0412] 바람직한 실시 예에서, 이웃들은 다음과 같이 정의된다. 위의 예측 유닛을 위하여, 모든 예측 유닛은 현재 예측 유닛을 포함하고 가장 큰 움직임 벡터를 갖는 예측 유닛으로 고려된다. 이는 또한 왼쪽 이웃을 위하여 수행된다. 모든 예측 유닛은 현재 예측 유닛이 평가되고 가장 큰 움직임 벡터를 갖는 예측 유닛이 사용되는 것을 포함한다. 또 다른 바람직한 실시 예에서, 모든 예측 유닛으로부터 평균 절대 움직임 벡터 값은 상부 및 왼쪽 경계를 포함하고 현재 예측 유닛은 첫 번째 빈을 유도하도록 사용된다.

[0414] 위의 본 발명의 바람직한 실시 예들을 위하여, 코딩 순차를 다음과 같이 변경하는 것이 가능하다. 움직임 벡터 차분값은 수평 및 수직 방향을 위하여 차례로 지정되어야만 한다(또는 반대로). 따라서 두 개의 빈 스트림이 코딩되어야만 한다. 엔트로피 코딩 엔진을 위한 모드 스위칭의 수를 최소화하기 위하여, 첫 번째 단계에서 두 컴포넌트를 위한 컨텍스트 모델로 코딩된 빈들을 코딩하고 그 다음에 두 번째 단계에서 바이패스 방식으로 코딩된

빈들을 코딩하는 것이 가능하다. 이는 단지 레코딩이라는 것에 유의하여야 한다.

[0416] 단항 또는 절단된 단항 이진화로부터 야기되는 빈들은 또한 값이 현재 빈 지수보다 큰지를 지정하는 빈 지수 당 하나의 플래그의 동등한 고정 길이 이진화에 의해 표현될 수 있다는 것을 이해하여야 한다. 예로써, 움직임 벡터 차분값의 절단된 단항 이진화를 위한 절단 값은 값들 0, 1, 2를 위한 코드워드들 0, 10, 11을 야기하는 2로 설정된다. 빈 지수 당 하나의 플래그를 갖는 상응하는 고정 길이 이진화에 있어서, 빈 지수 0(즉, 첫 번째 빈)을 위한 하나의 플래그는 절대 움직임 벡터 차분값 값이 0보다 큰지를 지정하고 빈 지수 1을 갖는 두 번째 빈을 위한 하나의 플래그는 절대 움직임 벡터 차분값 값이 1보다 큰지를 지정한다. 두 번째 플래그만이 코딩될 때 첫 번째 플래그가 1과 동일하면, 이는 동일한 코드워드들 0, 10, 11을 야기한다.

[0418] 그 다음에, 일 실시 예에 따른 확률 모델들의 내부 상태의 복잡도-확장성 표현이 설명된다.

[0420] 고효율-파이프 설정에서, 확률 모델의 내부 상태는 그것으로 빈을 인코딩한 후에 업데이트된다. 업데이트된 상태는 예전 상태 및 코딩된 빈의 값을 사용하는 상태 전이 테이블 룩업에 의해 유도된다. CABAC의 경우에 있어서, 확률 모델은 각각의 상태가 간격(0.0, 0.5)에서 하나의 모델 확률과 상응하는 63개의 서로 다른 상태를 취할 수 있다. 이러한 상태들 각각은 두 개이 모델 확률을 실현하기 위하여 사용된다. 상태에 할당된 확률에 더 하여, 1.0 - 확률이 또한 사용되고 valMPs라 불리는 플래그가 확률 또는 1.0 - 확률이 사용되는 정보를 저장한다.

[0422] 이는 총 126 상태들에 이르게 한다. 파이프 코딩 개념을 갖는 그러한 확률 모델을 사용하기 위하여, 각각의 126 상태들은 이용가능한 파이프 코더들 중 어느 하나에 매핑되는 것이 필요하다. 파이프의 현재 구현들에서, 이는 룩업-테이블을 사용함으로써 수행된다. 그러한 매핑의 일례가 테이블 A에 도시된다.

[0424] 다음에서, 일 실시 예가 내부 상태를 파이프 지수로 전환하도록 룩업 테이블을 사용하여 방지하기 위하여 확률 모델의 내부 상태가 어떻게 표현될 수 있는지를 설명한다. 확률 모델의 내부 상태 변수로부터 파이프 지수를 추출하기 위하여 오로지 일부 간단한 비트 마스킹 운용들이 필요하다. 확률 모델의 내부 상태의 이러한 신규의 복잡도-확장성 표현은 두 개의 레벨 방식으로 디자인된다. 낮은 복잡도 운용이 의무적인 적용들을 위하여 첫 번째 레벨만이 사용된다. 이는 관련 빈들을 인코딩하고 디코딩하는데 사용되는 파이프 지수 및 flag valMPs만을 설명한다. 설명된 파이프 엔트로피 코딩 전략의 경우에 있어서, 첫 번째 레벨은 8개의 서로 다른 확률 모델 사이를 구별하도록 사용될 수 있다. 따라서, 첫 번째 레벨은 파이프 지수를 위한 3 비트 및 valMPs를 위한 하나의 또 다른 비트가 필요할 수 있다. 두 번째 레벨로 첫 번째 레벨의 각각의 코스(coarse) 확률 범위들은 높은 해상도에서 확률들의 표현을 제공하는 일부 더 작은 간격들로 개선된다. 이러한 더 상세한 표현은 확률 추정기들의 더 정확한 운용을 가능하게 한다. 일반적으로, 이는 높은 RD 성능을 겨냥한 코딩 적용들에 적합하다. 예로서, 파이프의 사용을 갖는 확률 모델들의 내부 상태의 이러한 복잡도-확장성 표현은 다음과 같이 도시된다:

첫 번째 레벨				두 번째 레벨			
b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀
MPS	파이프 지수 (0-7)			세분화 지수 (0-15)			

[0425]

[0427] 첫 번째 및 두 번째 레벨은 단일 8 비트 메모리 내에 저장된다. 첫 번째 레벨(가장 유의한 비트 상의 더 가능성이 있는 심벌의 값을 갖는 파이프 지수를 정의하는 지수)을 저장하기 위하여 4 비트들이 필요하며 두 번째 레벨을 저장하기 위하여 또 다른 4 비트들이 필요하다. CABAC 확률 추정기의 행동을 구현하기 위하여, 각각의 파이프 지수는 얼마나 많은 CABAC 상태들이 파이프 지수 상에 매핑되었는가에 따라 특정 수의 허용된 세분화 지수를 갖는다. 예를 들면, 테이블 A에서의 매핑을 위하여, 테이블 B에 파이프 지수 당 CABAC 상태들의 수가 도시된다.

파이프 지수	0	1	2	3	4	5	6	7
CABAC 상태들의 수	3	7	5	7	10	14	16	1

[0429]

[0430]

테이블 B: 테이블 A의 예를 위한 파이프 지수 당 CABAC 상태들의 수

[0432]

빈의 인코딩 또는 디코딩 과정 동안에 파이프 지수 및 valMps는 간단한 비트 마스크(mask) 또는 비트 이동(shift) 운용들을 사용함으로써 직접적으로 액세스될 수 있다. 낮은 복잡도 코딩 과정들은 첫 번째 레벨의 4 비트들만을 필요로 하며 고효율 코딩 과정들은 CABAC 확률 추정기의 확률 모델 업데이트를 실행하기 위하여 두 번째 레벨의 4 비트들을 사용할 수 있다. 이러한 업데이트들 실행하기 위하여, 오리지널 테이블과 동일한 상태 전이들을 수행하나, 상태들의 복잡도-확장성 두-레벨 표현을 사용하는, 상태 전이 록업-테이블이 디자인될 수 있다. 오리지널 상태 전이 테이블은 63 요소들의 두 배로 구성된다. 각각의 빈 상태를 위하여, 이는 두 개의 출력 상태를 포함한다. 복잡도-확장성 표현을 사용할 때, 상태 전이(transition) 테이블은 테이블 크기의 수용가능한 증가인 128 요소의 2배로 구성된다. 이러한 증가는 세분화 지수를 사용하기 위하여 얼마나 많은 비트들이 사용되는가에 의존하며 정확하게 CABAC 확률 추정기의 행동을 대행하기 위하여 4개의 비트가 필요하다. 그러나, 각각의 파이프 지수를 위하여 단지 8 상태만이 허용되는 것과 같이 CABAC 상태들의 감소된 세트 상에서 운용할 수 있는, 서로 다른 확률 추정기가 사용될 수 있다. 따라서 메모리 소비는 세분화 지수를 표현하도록 사용되는 비트들의 수를 적용함으로써 코딩 과정의 주어진 복잡도 레벨과 일치될 수 있다. CABAC(64 확률 상태 지수가 존재하는)를 갖는 모델 확률들의 내부 상태와 비교하여 모델 확률들을 특정 파이프 코드에 매핑하기 위한 테이블 록업들의 사용이 방지되고 어떠한 다른 전환도 필요하지 않다.

[0434]

그 다음에, 일 실시 예에 따른 복잡도-확장성 컨텍스트 모델 업데이트가 설명된다.

[0436]

그러나, 복잡도-확장성 방법으로 컨텍스트 모델들의 업데이트를 수행하는 것이 가능하다. 즉, 컨텍스트 모델을 업데이트하는지에 대한 결정은 다양한 양상들을 기초로 할 수 있다. 예를 들면, 코더 설정은 예를 들면, 구문 요소 *coeff_significant_flag*의 컨텍스트 모델들 같은 특정 컨텍스트 모델들을 위하여 어떠한 업데이트도 수행할 수 없으며, 다른 모든 컨텍스트 모델을 위하여 항상 업데이트를 수행한다.

[0438]

바꾸어 말하면, 선택기(402)는 다수의 미리 결정된 심벌 형태의 각각의 심벌들을 위하여, 미리 결정된 심벌 형태들의 수가 고효율 방식과 비교하는 것보다 더 낮은 복잡도에서 낮은 것과 같이 각각의 확률 모델과 관련된 각각의 미리 결정된 심벌에 따라 엔트리 디코더들(322) 중에서의 선택을 실행하도록 구성될 수 있다.

[0440]

게다가, 컨텍스트 모델을 업데이트할 것인가를 제어하기 위한 표준은 예를 들면, 비트스트림 패킷의 크기, 지금까지 디코딩된 빈들의 수일 수 있거나, 또는 업데이트는 컨텍스트 모델을 위한 빈들의 특정한 고정 또는 가변 수를 코딩한 후에만 수행된다.

[0442]

컨텍스트 모델들을 업데이트할 것인가를 결정하기 위한 이러한 전략과 함께, 복잡도-확장성 컨텍스트 모델 업데이트가 구현될 수 있다. 이는 컨텍스트 모델 업데이트들이 수행되는 비트스트림 내의 빈들의 부를 증가시키거나 감소시키도록 허용한다. 컨텍스트 모델 업데이트들의 수가 높을수록, 코딩 효율은 더 뛰어나고 계산 복잡도가 높아진다. 따라서, 복잡도-확장성 컨텍스트 모델 업데이트는 설명된 전략으로 달성될 수 있다.

[0444]

바람직한 실시 예에서, 컨텍스트 모델 업데이트는 구문 요소들 *coeff_significant_flag*, *coeff_abs_greater1*, 및 *coeff_abs_greater2*를 제외하고 모든 구문 요소의 빈들을 위하여 수행된다.

- [0446] 또 다른 바람직한 실시 예에서, 콘텍스트 모델 업데이트는 *coeff_significant_flag*, *coeff_abs_greater1*, 및 *coeff_abs_greater2*만의 bin들을 위하여 수행된다.
- [0448] 또 다른 바람직한 실시 예에서, 콘텍스트 모델 업데이트는 조각의 인코딩 또는 디코딩이 시작할 때 모든 콘텍스트 모델을 위하여 수행된다. 특정한 미리 정의된 수의 변환 블록들이 처리된 후에, 콘텍스트 모델 업데이트는 조각의 말단이 도달할 때까지 모든 콘텍스트 모델을 위하여 사용할 수 없다.
- [0450] 예를 들면, 선택기(402)는 미리 결정된 심벌 형태의 심벌들을 위하여, 미리 결정된 심벌 형태의 심벌들이 업데이트와 함께 실행되는 심벌들의 시퀀스의 학습 단계의 길이가 고효율 방식과 비교하는 것보다 낮은 복잡도에서 더 짧은 것과 같이, 관련 확률 모델의 업데이트와 함께 또는 업데이트 없이 미리 결정된 심벌 형태와 관련된 확률 모델에 따라 엔트로피 디코더들(322) 중에서의 선택을 실행하도록 구성될 수 있다.
- [0452] 또 다른 바람직한 실시 예는 이전에 설명된 바람직한 실시 예와 동일하나, 이는 하나의 테이블이 모든 콘텍스트 모델의 "첫 번째 부분"(valMps 및 pipeIdx)를 저장하고 두 번째 테이블이 모든 콘텍스트 모델의 "두 번째 부분"(세분화Idx)을 저장하는 것과 같은 방법으로 콘텍스트 모델의 내부 상태의 복잡도-확장성 표현을 사용한다. 모든 콘텍스트 모델을 위하여 콘텍스트 모델 업데이트가 사용될 수 없는(이전의 바람직한 실시 예에서 설명된 것과 같이), 지점에서, "두 번째 부분"을 저장한 테이블은 더 이상 필요하지 않고 버려질 수 있다.
- [0454] 그 다음에, 일 실시 예에 따른 bin들의 시퀀스를 위한 콘텍스트 모델 업데이트가 설명된다.
- [0456] 낮은 복잡도-파이프 구성에서, *coeff_significant_flag*, *coeff_abs_greater1*, 및 *coeff_abs_greater2*을 gidxodml 구문 요소들의 bin들은 서브셋들로 그룹화된다. 각각의 서브셋을 위하여, 그것의 bin들을 인코딩하기 위하여 단일 콘텍스트 모델이 사용된다. 이러한 경우에 있어서, 콘텍스트 모델 업데이트는 이러한 시퀀스의 고정된 수의 bin들의 코딩 후에 수행될 수 있다. 이는 그 다음에 다중-bin 업데이트를 나타낸다. 그러나, 이러한 업데이트는 마지막으로 코딩된 bin과 콘텍스트 모델의 내부 상태만을 사용하는 업데이트와 다를 수 있다. 예를 들면, 코딩된 각각의 bin을 위하여, 하나의 콘텍스트 모델 업데이트 단계가 수행된다.
- [0458] 다음에서, 8 bin으로 구성되는 바람직한 서브셋의 인코딩을 위한 예들이 주어진다. 글자 'b'는 bin의 디코딩을 표시하고 글자 'u'는 콘텍스트 모델의 업데이트들을 표시한다. 낮은 복잡도-파이프 경우에 있어서, 콘텍스트 모델 업데이트들을 하지 않고 bin 디코딩만이 수행된다:
- [0459] b b b b b b b b
- [0461] 고효율-파이프 경우에 있어서, 각각의 bin의 디코딩 후에, 콘텍스트 모델 업데이트가 수행된다:
- [0462] b u b u b u b u b u b u b u
- [0464] 복잡도를 다소 감소시키기 위하여, bin들의 시퀀스 뒤에 콘텍스트 모델 업데이트가 수행될 수 있다(본 실시 예에서는 각각 4 bin 뒤에, 이러한 4 bin들의 업데이트들이 수행된다):
- [0465] b b b b u u u u b b b b u u u u
- [0467] 즉, 선택기(402)는 미리 결정된 심벌 형태의 심벌들을 위하여, 업데이트와 함께 미리 결정된 심벌 형태의 심벌

들을 위한 선택이 실행되는 주파수가 고효율 방식과 비교하는 것보다 낮은 복잡도에서 낮은 것과 같이, 관련 확률 모델의 업데이트와 함께 또는 업데이트 없이 미리 결정된 심벌 형태와 관련된 확률 모델에 따라 엔트로피 디코더들(322) 중에서의 선택을 실행하도록 구성될 수 있다.

[0469] 이러한 경우에 있어서, 4 빈들의 디코딩 후에, 방금 디코딩된 4 빈들을 기초로 하여 4 업데이트 단계가 뒤따른다. 이러한 4 업데이트 단계는 룩업 특별 룩업-테이블을 사용함으로써 하나의 단일 단계로 수행될 수 있다. 이러한 룩업 테이블은 4 빈들의 각각의 가능한 조합 및 컨텍스트 모델의 각각의 가능한 내부 상태를 위하여 4 종래의 업데이트 단계들 후에 결과로서 생기는 새로운 상태를 저장한다.

[0471] 특정 방식에서, 구문 요소(*coeff_significant_flag*)를 위하여 다중-빈 업데이트가 사용된다. 다른 모든 구문 요소의 빈들을 위하여, 어떠한 컨텍스트 업데이트 모델도 사용되지 않는다. 다중-빈 업데이트 단계가 수행되기 전에 코딩되는 빈들의 수는 n 으로 설정된다. 세트의 빈들의 수가 n 으로 나뉘질 수 없을 때, 1 내지 $n-1$ 빈들은 마지막 다중-빈 업데이트 후에 서브셋의 끝에 남는다. 이러한 각각의 빈들을 위하여, 이러한 빈들을 모두 코딩한 후에 종래의 단일-빈 업데이트가 수행된다. 수 n 은 1보다 큰 어떠한 양의 수일 수 있다. 또 다른 방식은 다중-빈 업데이트가 *coeff_significant_flag*, *coeff_abs_greater1*, 및 *coeff_abs_greater2*의 임의의 조합들(단기 *coeff_abs_greater1*만 대신에)을 위하여 수행되는 것을 제외하고는, 이전 방식과 동일할 수 있다. 따라서, 이러한 방식은 다른 것보다 더 복잡할 수 있다. 다른 모든 구문 요소(다중-빈 업데이트가 사용되지 않는)는 서브셋들 중 하나를 위하여 단일 빈 업데이트가 사용되고 나머지 서브셋을 위하여 어떠한 확률 모델 업데이트도 사용되지 않는 두 개의 서로소 서브셋으로 나뉜다. 어떤 가능한 서로소 서브셋도 유효하다(엔트로피 서브셋을 포함하여).

[0473] 대안의 실시 예에서, 다중-빈 업데이트는 다중 빈 업데이트 단계 이전에 즉시 코딩되는 마지막 m 빈들만을 기초로 할 수 있다. m 은 n 보다 작은 어떠한 자연수일 수 있다. 따라서, 디코딩은 다음과 같이 수행될 수 있다:

[0474] b b b b u u b b b b u u b b b b u u b b b b....

[0475] $n=4$ 이고 $m=2$.

[0477] 즉, 선택기(402)는 미리 결정된 심벌 형태의 심벌들을 위하여, n/m 비율이 고효율 방식과 비교하는 것보다 낮은 복잡도에서 높은 것과 같이, 미리 결정된 심벌 형태의 가장 최근의 m 심벌들을 기초로 하여 미리 결정된 형태의 관련 확률 모델의 모든 n -번째 심벌의 업데이트와 함께, 미리 결정된 심벌 형태와 관련된 확률 모델에 따라 엔트로피 디코더들(322) 중에서의 선택을 실행하도록 구성될 수 있다.

[0479] 또 다른 바람직한 실시 예에서, 구문 요소(*coeff_significant_flag*)를 위하여, 컨텍스트 모델들을 구문 요소들의 빈들에 할당하기 위하여 고효율-파이프 구성을 위하여 위에 설명된 것과 같이 로컬 템플릿을 사용하는 컨텍스트 모델링 전략이 사용될 수 있다. 그러나, 이러한 빈들을 위하여, 어떠한 컨텍스트 모델 업데이트도 사용되지 않는다.

[0481] 또한, 선택기(402)는 미리 결정된 심벌 형태의 심벌들을 위하여, 컨텍스트들의 수 및/또는 이전에 검색된 심벌들의 수가 고효율 방식과 비교하는 것보다 낮은 복잡도에서 낮은 것과 같이, 심벌들의 시퀀스의 이전에 검색된 심벌들의 수에 따라 컨텍스트들의 수 중 어느 하나를 선택하고 선택된 컨텍스트와 관련된 확률 모델에 따라 엔트로피 디코더들 중에서의 선택을 실행하도록 구성될 수 있다.

[0483] 8 비트 초기화 값들을 사용하는 확률 모델 초기화

[0484] 본 섹션은 종래의 비디오 코딩 표준 H.256/AVC의 경우에서와 같이 두 8 비트 값 대신에 이른바 8 비트 초기화 값을 사용하는 확률 모델들의 복잡도-확장성 내부 상태의 초기화 과정을 설명한다. 이는 H.264/AVC의 CABAC에서

의 확률 모델들을 위하여 사용되는 초기 값 쌍들에 비교할 수 있는 두 부분으로 구성된다. 두 부분은 양자화 파라미터로부터 특정 확률(예를 들면, 파이프 지수 형태의)을 표현하는, 확률 모델의 초기 상태를 계산하기 위하여 선형 방정식의 두 파라미터를 표현한다:

[0485] • 첫 번째 부분은 슬로프를 설명하며 인코딩 또는 디코딩 동안에 사용되는 양자화 파라미터와 관련하여 초기 상태의 의존성을 이용한다.

[0486] • 두 번째 부분은 주어진 양자화 파라미터에서의 파이프 지수뿐만 아니라 valMps를 정의한다.

[0488] 주어진 초기화 값을 사용하여 확률 모델을 초기화하기 위하여 두 가지 서로 다른 방식이 이용가능하다. 첫 번째 방식은 도식된 양자화 파라미터-독립 초기화이다. 이는 단지 파이프 지수 및 모든 양자화 파라미터를 위한 초기화 값의 두 번째 부분에 정의된 valMps만을 사용한다. 이는 슬로프가 0과 동일한 경우와 동일하다. 두 번째 방식은 양자화 파라미터-의존 초기화로 설명되며 이는 부가적으로 파이프 지수를 변경하고 세분화 지수를 정의하기 위하여 초기화 값의 첫 번째 부분의 슬로프를 사용한다. 8 비트 초기화 값의 두 부분은 다음과 같이 도식된다:

첫 번째 부분				두 번째 부분			
b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
슬로프 지수				파이프 확률 지수			

[0489]

[0491] 이는 두 개의 4 비트 부분들로 구성된다. 첫 번째 부분은 어레이 내에 저장되는 16개의 서로 다른 미리 정의된 슬로프 중에서 1을 가리키는 지수를 포함한다. 미리 정의된 슬로프들은 7개의 음의(negative) 슬로프(슬로프 지수 0-6), 제로와 동일한 하나의 슬로프(슬로프 지수 7) 및 8개의 양의 슬로프(슬로프 지수 8-15)로 구성된다. 슬로프들이 테이블 C에 도식된다.

슬로프 지수	0	1	2	3	4	5	6	7
슬로프 값	-239	-143	-85	-51	-31	-19	-11	0
슬로프 지수	8	9	10	11	12	13	14	15
슬로프 값	11	19	31	51	85	143	239	399

[0493]

[0494] 테이블 C;

[0496] 부동 소수점 연산(floating point operation)들의 사용을 방지하기 위하여 모든 값들은 256의 인자에 의해 매겨진다. 두 번째 부분은 확률 간격 $p=0$ 및 $p=1$ 사이의 valMps=1의 오름차순 확률을 구현하는 파이프 지수이다. 바꾸어 말하면, 파이프 코드 n 은 파이프 코드 $n-1$ 보다 높은 모델 확률에서 운용하여야만 한다. 모든 모델 확률을 위하여, 하나의 파이프 확률 지수가 이용가능하며 이는 확률 간격이 $Q=26$ 을 위하여 pvalMps=1의 확률을 포함하는 파이프 코드를 식별한다.

파이프 확률 지수	0	1	2	3	4	5	6	7
파이프 코더	UR5	UR4	UR3	UR2	TB	BP2	BP3	EP
움직임 벡터 차분값	0	0	0	0	0	0	0	0
파이프 확률 지수	8	9	10	11	12	13	14	15
파이프 코더	EP	BP3	BP2	TB	UR2	UR3	UR4	UR5
움직임 벡터 차분값	1	1	1	1	1	1	1	1

[0498]

[0499]

테이블 D: 초기화 값의 두 번째 부분의 파이프 코더들과 valMps로의 매핑: UR= 단항-대-라이스-코드, TB=3-빈-코드, BP=빈-파이프-코드, EP=동등한 확률(코딩되지 않은)

[0501]

$y = m * (QP - QPref) + 256 * b$ 형태의 간단한 선형 방정식을 계산함으로써 확률 모델들의 내부 상태의 초기화를 계산하기 위하여 양자화 파라미터 및 8 비트 초기화 값이 필요하다. m은 슬로프 지수(8 비트 초기화 값의 첫 번째 부분)의 사용에 의해 테이블 C로부터 취해지는 슬로프를 정의하고 b는 QPref=26(8 비트 초기화 값의 두 번째 부분: "파이프 확률 지수")에서 파이프 코더를 나타낸다는 것에 유의하여야 한다. 그때, valMPS는 1이고 만일 y가 2047보다 크면 파이프 지수는 $(y - 2048) \gg 8$ 과 동일하다. 그렇지 않으면, valMps는 0이고 파이프 지수는 $(2047 - y) \gg 8$ 과 동일하다. 세분화 지수는 만일 valMPS가 1과 동일하면 $((y - 2048) \& 255) * numStates \gg 8$ 과 동일하다. 두 경우에 있어서, numStates는 테이블 B에 도시된 것과 같이 파이프 지수의 CABAC 상태들의 수와 동일하다.

[0503]

위의 전략은 파이프 코더들과 조합하여 사용될 수 있을 뿐만 아니라 위에서 언급된 CABAC 전략들과 함께 사용될 수 있다. 파이프가 없을 때, 파이프 지수(즉, pState_current[bin]의 각각의 가장 중요한 빈들)당, CABAC 상태들, 즉, 그 사이에 확률 업데이트 내의 상태 전이가 실행되는 확률 상태들(pState_current[bin])은 그때 단지 실제로 양자화 파라미터에 따라 CANAC 상태의 피스-방식 선형 보간을 실현하는 파라미터들의 세트이다. 실제로, 게다가, 이러한 파이프-방식의 선형 보간은 또한 파라미터 numStates가 모든 파이프 지수를 위하여 동일한 값을 사용하는 경우에 사실상 사용할 수 없을 수 있다. 예를 들면, 모든 경우를 위한 numStates의 8로의 설정은 총 $16 * 8$ 상태를 생산하고 세분화 지수의 계산은 valMPS=1을 위한 $((y - 2048) \& 255) \gg 5$ 또는 valMPS=0을 위한 $((2047 - y) \& 255) \gg 5$ 로 단순화한다. 이러한 경우를 위하여, valMPS, 파이프 지수, 및 세분화 지수를 사용하는 표현의 다시 H.264/AVC의 오리진널 CABAC에 의해 사용되는 표현으로의 매핑은 매우 간단하다. CABAC 상태는 (파이프 지수 $\ll 3$) + 세분화 지수로서 주어진다. 이러한 양상은 도 16과 관련하여 아래에 더 설명된다.

[0505]

8 비트 초기화 값의 슬로프가 제로와 동일하지 않거나 또는 양자화 파라미터가 26과 동일하지 않으면, 인코딩 또는 디코딩 과정의 양자화 파라미터를 갖는 선형 방정식의 사용에 의해 내부 상태를 계산하는 것이 필요하다. 제로와 동일시키는 슬로프 또는 현재 코딩 과정의 양자화 파라미터가 26과 동일한 슬로프의 경우에 있어서 8 비트 초기화 값의 두 번째 부분이 확률 모델의 내부 상태를 초기화하기 위하여 직접적으로 사용될 수 있다. 그렇지 않으면, 결과로서 생기는 내부 상태의 소수 부분은 특정 파이프 코더의 한계들 사이의 선형 보간에 의한 고효율 코딩 적용들에서 세분화 지수를 결정하기 위하여 더 활용될 수 있다. 이러한 바람직한 실시 예에서 선형 보간은 간단히 소수 부분을 현재 파이프 코더를 위하여 이용가능한 세분화 지수들의 총 수와 곱하고 그 결과를 가장 가까운 정수 세분화 지수에 매핑함으로써 실행된다.

[0507]

확률 모델들의 내부 상태의 초기화의 과정은 파이프 확률 지수 상태들의 수와 관련하여 변경될 수 있다. 특히, 파이프 코더 E1을 사용하는 동일한 가능 방식의 이중 발생, 즉 1 또는 0인 더 가능성이 있는 심벌 사이를 구별

하기 위한 두 서로 다른 파이프 지수의 사용은 다음과 같이 방지될 수 있다. 다시, 과정은 조각 데이터의 파싱 동안에 적용될 수 있으며, 이러한 과정의 입력은 테이블 E에 도시된 것과 같이 예를 들면, 초기화되는 모든 콘 텍스트 모델을 위하여 비트 스트림 내에 전송될 수 있는, 8 비트 초기화 값일 수 있다.

[0509]

테이블 E: 확률 모델을 위한 무한 값의 8 비트들의 설정

	첫 번째 4 비트들				마지막 4 비트들			
초기 값 빈들	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀
변수	슬로프 지수				확률 지수			

[0510]

[0512]

처음의 4 비트는 슬로프 주를 정의하며 비트들 b₄-b₇을 마스킹함으로써 검색된다. 모든 슬로프 지수를 위하여 테이블 F에 슬로프(m)가 지정되고 표시된다.

[0514]

테이블 F. 슬로프 지수를 위한 변수 m의 값들

슬로프 지수	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
m	-39	-143	-85	-51	-31	-19	-11	0	11	19	31	51	85	143	239	399

[0515]

[0517]

비트들 b₀-b₃, 8 비트 초기 값의 마지막 4 비트들은 확률 지수를 식별하고 미리 정의된 양자화 파라미터에서의 확률을 설명한다. 확률 지수 0은 각각 값 0을 갖는 심벌들을 위한 가장 높은 확률을 나타내고, 확률 지수 14는 값 1을 갖는 심벌들을 위한 가장 높은 확률을 나타낸다. 테이블 G는 각각의 확률 지수를 위하여 상응하는 파이프 코더와 그것의 valMPs를 나타낸다.

[0519]

테이블 G. 초기 값의 마지막 4 부분의 파이프 코더들과 valMPs로의 매핑. UR = 단항-대-라이스-코드, TB = 3-빈-코드, BP = 빈-파이프-코드, EP = 동일한 확률(코딩되지 않은)

확률 지수	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
파이프 코더	UR5	UR4	UR3	UR2	TBC	BP2	BP3	EP	BP3	BP2	TBC	UR2	UR3	UR4	UR5
valMPs	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1

[0520]

[0522]

두 값과 함께 내부 상태의 계산은 $y = m \cdot x + 256 \cdot b$ 와 같은 선형 방정식을 사용함으로써 수행될 수 있는데, 여기서 m은 슬로프를 나타내고, x는 현재 조각의 양자화 파라미터를 나타내며 b는 아래의 설명에서 나타내는 것과 같이 확률 지수로부터 유래한다. 이러한 과정에서의 모든 값들은 부동 소수점 운용들의 사용을 방지하기 256의 인자에 의해 계산된다. 이러한 과정의 출력(y)은 현재 양자화 파라미터에서의 확률 모델의 내부 상태를 표현하며 8 비트 메모리 내에 저장된다. G에 도시된 것과 같이 내부 상태는 valMPs, 파이프 지수 및 세분화 지수로 구성된다.

[0524] 테이블 H. 확률 모델의 내부 상태의 설정

	첫 번째 4 비트들				마지막 4 비트들			
초기 값 빈들	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀
변수	valMps	파이프 지수			세분화 지수			

[0525]

[0527] 세분화 지수와 파이프 지수의 할당은 CABAC 확률 모델의 내부 상태(pStateCtx)와 유사하며 H 내에 표현된다.

[0529] 테이블 I. 파이프 지수, 세분화 지수 및 확률 상태 지수

파이프 지수	0			1							2											
세분화 지수	0	1	2	0	1	2	3	4	5	6	0	1	2	3	4							
CABAC 확률모델	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14							
파이프 지수	3							4														
세분화 지수	0	1	2	3	4	5	6	0	1	2	3	4	5	6	7	8	9					
CABAC 확률모델	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31					
파이프 지수	5																					
세분화 지수	0	1	2	3	4	5	6	7	8	9	10	11	12					13				
CABAC 확률모델	32	33	34	35	36	37	38	39	40	41	42	43	44					45				
파이프 지수	6																7					
세분화 지수	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		0				
CABAC 확률모델	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61		62				

[0530]

[0532] 바람직한 실시 예에서 확률 지수는 QP26에서 정의된다. 8 비트 초기화 값을 기초로 하여 내부 상태(valMps, 파이프 지수 및 세분화 지수)는 다음의 슈도-코드에서 설명되는 것과 같이 처리된다:

[0533] $n = (\text{probIdx} \ll 8) - m * 26$

[0534] $\text{fullCtxState} = \max(0, \min(3839, (m * \max(0, \min(51, \text{SliceQP}_v)$

[0535] $))) + n + 128)$

[0536] $\text{remCtxState} = \text{fullCtxState} \& 255$

[0537] $\text{preCtxState} = \text{fullCtxState} \gg 8$

[0538] $\text{if}(\text{preCtxState} < 8) \{$

```

[0539]   pipeIdx = 7   preCtxState
[0540]   valMPS = 0
[0541]   } else {
[0542]   pipeIdx = preCtxState - 8
[0543]   valMPS = 1
[0544]   }
[0545]   offset = { 3, 7, 5, 7, 10, 14, 16, 1 }
[0546]   if( pipeIdx == 0 ) {
[0547]   if( remCtxState <= 127 )
[0548]       remCtxState = 127 - remCtxState
[0549]   else
[0550]       remCtxState = remCtxState - 128
[0551]   refineIdx = ( ( remCtxState << 1 ) * offset ) >> 8
[0552]   } else {
[0553]   if( valMPS == 0 )
[0554]       remCtxState = 255 - remCtxState
[0555]   refineIdx = ( remCtxState * offset[pipeIdx] ) >> 8
[0556]   }

```

[0558] 슈도 코드에 도시된 것과 같이 세분화 지수는 파이프 지수의 간격 사이를 선형으로 보간하고 그 결과를 상응하는 세분화 지수에 양자화함으로써 계산된다. 오프셋은 각각의 파이프 지수를 위하여 세분화 지수의 총 수를 지정한다. fullCtxState/256의 간격 [7, 8]은 반으로 나뉜다. 간격 [7, 7.5]은 pipeIdx = 0 및 valMps = 0으로 매핑되며 간격 [7.5, 8]은 pipeIdx = 0 및 valMps = 1로 매핑된다. 도 16은 내부 상태를 유래하는 과정을 도시하고 fullCtxState/256의 pStateCtx로의 매핑을 나타낸다.

[0560] 슬로프는 파이프 지수 및 양자화 파라미터의 의존성을 표시한다는 것에 유의한다. 만일 8 빈 초기화 값의 슬로프 지수가 7과 동일하면 결과로서 생기는 내부 상태는 모든 조각 양자화 파라미터들을 위하여 동일하며 따라서 내부 상태의 초기화 과정은 조각의 현재 양자화 파라미터로부터 독립된다.

[0562] 즉, 선택기(402)는 두 방식, 낮은 복잡도와 고효율에 일반적일 수 있는 테이블 내의 지수로서 이러한 구문 요소를 사용하는 그 안에 포함되는 변환 계수 레벨들과 같이, 이러한 부의 데이터를 양자화하기 위하여 사용되는 양자화 단계 크기(QP)를 표시하는 구문 요소를 사용하여, 전체 스트림 또는 그 다음의 조각과 같은 데이터스트림의 다음의 부를 디코딩하는데 사용되는 파이프 지수들을 초기화할 수 있다. 테이블 D와 같은 테이블은 각각의 참조 QPref를 위하여, 각각의 심별 형태를 위한 파이프 지수들, 또는 각각의 심별 형태를 위한 다른 데이터를 포함할 수 있다. 현재 부의 실제 양자화 파라미터에 따라, 선택기는 폭(QP-QPref)을 곱하는 것에 의한 것과 같이, 실제 양자화 파라미터와 양자화 파라미터 자체에 의해 지수화되는 각각의 테이블 엔트리를 사용하여 파이프 지수 값을 계산할 수 있다. 낮은 복잡도 및 고효율 방식에서의 유일한 차이는 다음과 같다: 선택기는 고효율 방식과 비교하여 낮은 복잡도 방식의 경우 더 낮은 정확도로 결과를 계산한다. 선택기는 예를 들면, 단지 계산 결과의 정수 부분을 사용할 수 있다. 고효율 방식에서, 낮은 정확도 또는 정수 부분에 의해 표시되는 것과 같은 각각의 파이프지수를 위하여 이용가능한 세분화 지수 중의 하나를 선택하기 위하여 소수 부분(fractional part)과 같은, 가장 높은 정확도 나머지가 사용된다. 위에서 언급된 테이블 워크의 사용에 의한 것과 같은 확률 적

용성을 실행하기 위하여 세분화 지수는 고효율 방식에서(또한 잠재적으로 더 드물게 낮은 복잡도 방식에서) 사용된다. 높은 경계에서 현재 파이프 지수를 위한 이용가능한 지수들을 제거할 때, 세분화 지수의 최소화와 함께 가장 높은 파이프지수가 그 다음에 선택된다. 낮은 경계에서 현재 파이프 지수를 위한 이용가능한 지수들을 제거할 때, 새로운 파이프 지수를 위하여 이용가능한 최대 세분화 지수의 최대화와 함께 그 다음으로 낮은 파이프 지수가 그 다음에 선택된다. 세분화 지수와 함께 파이프 지수는 확률 상태를 정의하나, 부분 스트림들 중에서의 선택을 위하여, 선택기는 단지 파이프 지수를 사용한다. 세분화 지수는 단지 확률을 더 가깝게, 또는 더 정확하게 추적하는 역할을 한다.

[0564] 그러나, 위의 설명은 또한 복잡도 확장성이 도 7-10의 파이프 코딩 개념 또는 도 12에 도시된 것과 같은 디코더를 사용하는 CABAC와 관계없이 달성될 수 있다는 것을 나타낸다. 도 12의 디코더는 미디어 데이터가 코딩되는 데이터 스트림을 디코딩하기 위한 것이며, 데이터 스트림(601)에 따라 낮은-복잡도 방식 또는 고효율 방식을 활성화하도록 구성되는 모드 스위치(600)뿐만 아니라, 심벌 시퀀스 워드들의 정수 값 구문 요소들의 공-도메인으로의 매핑을 위하여, 제어 파라미터에 의해 제어가능한 매핑 기능을 사용하여 정수 값 구문 요소들을 획득하기 위하여 데이터 스트림으로부터 예를 들면, 직접적으로 또는 엔트로피 디코딩에 의해 획득된 심벌들의 시퀀스(603)를 디심벌라이저하도록 구성되는 디심벌라이저(602)를 포함한다. 재구성자(605)는 정수 값 구문 요소들을 기초로 하여 미디어 데이터(606)를 재구성하도록 구성된다. 디심벌라이저(602)는 화살표(607)에 의해 도시된 것과 같이, 고효율 방식이 활성화되는 경우에서의 제 1 비율로 제어 파라미터가 데이터 스트림에 따라 변경되고 제어 파라미터가 데이터 스트림에 관계없이 일정하거나 또는 낮은-복잡도 방식이 활성화되는 경우에서의 제 1 비율보다 낮은 제 2 비율로, 데이터 스트림에 따라 변경하는 것과 같이 탈심벌화를 실행하도록 구성된다. 예를 들면, 제어 파라미터는 이전에 탈심벌화된 심벌들에 따라 변경될 수 있다.

[0566] 위의 설명 중 일부는 도 12의 양상을 사용하였다. 시퀀스(327) 내의 구문 요소들 `coeff_abs_minus3` 및 움직임 벡터 차분값은 예를 들면, 407에 의해 표시된 것과 같이 선택된 방식에 따라 디심벌라이저(314) 내에 이진화되었으며, 재구성자(605)는 재구성을 위하여 이러한 구문 요소들을 사용하였다. 분명하게, 도 11과 19의 두 양상은 또한 쉽게 조합할 수 있으나, 도 12의 양상은 또한 다른 코딩 환경들과 결합될 수 있다.

[0568] 예를 들면, 위에서 설명된 움직임 벡터 차분값 코딩이 참조된다. 디심벌라이저(602)는 매핑 기능이 절단 값 아래의 정수 값 구문 요소들의 도메인의 첫 번째 간격 내의 매핑 및 절단 값을 위한 절단된 단항 코드 형태의 접두부호와 절단 값을 포함하고 절단 값 위의 정수 값 구문 요소들의 도메인의 두 번째 간격 내의 가변 길이 코딩 코드워드 형태의 접미부호의 조합을 실행하기 위하여 절단된 단항 코드를 사용하는 것과 같이 구성될 수 있으며, 디코더는 변경하는 확률 추정을 갖는 엔트로피 코딩을 사용하여 데이터 스트림(601)으로부터 절단된 단항 코드의 다수의 첫 번째 빈 및 일정한 등-확률 바이패스 방식을 사용하여 가변 길이 코딩 코드워드의 다수의 두 번째 빈을 유래하도록 구성되는 엔트로피 디코더(608)를 포함할 수 있다. 고효율 방식에서, 엔트로피 코딩은 화살표(609)에 의해 도시된 것과 같이 낮은 복잡도 코딩보다 더 복잡할 수 있다. 즉, 콘텍스트-적응성 및/또는 확률 적응성은 고효율 방식으로 적용되고 낮은 복잡도 방식으로 억제될 수 있거나, 또는 복잡도는 다양한 실시예들과 관련하여 위에서 다른 것과 같이, 다른 항들로 계산될 수 있다.

[0570] 미디어 데이터를 데이터 스트림 내로 인코딩하기 위하여, 도 11이 디코더와 꼭 맞는 인코더가 도 13에 도시된다. 이는 데이터 스트림(501) 내로 낮은-복잡도 방식 또는 고효율 방식의 활성을 신호전달하도록 구성되는 삽입기(500), 미디어 데이터(505)를 구문 요소들의 시퀀스(506) 내로 프리코딩(precoding)하도록 구성되는 구성자(504), 구문 요소들의 시퀀스(506)를 심벌들의 시퀀스(508) 내로 기호화하도록 구성되는 심벌라이저(symbolizer, 507), 각각 심벌들의 부분 시퀀스들을 데이터 스트림의 코드 워드들로 전환하도록 구성되는 복수의 엔트로피 인코더(310), 및 심벌들의 시퀀스(508)의 각각의 심벌을 복수의 엔트로피 인코더(310) 중 선택된 하나에 전달하도록 구성되는 선택기(502)를 포함할 수 있으며, 선택기(502)는 화살표(51)에 의해 도시된 것과 같이 낮은-복잡도 방식과 고효율 방식 중 활성화된 하나에 따라 선택을 실행하도록 구성된다. 인코더들(310)의 코드워드들을 위하여 선택적으로 인터리버(510)가 제공될 수 있다.

- [0572] 미디어 데이터를 데이터 스트림 내로 인코딩하기 위하여, 도 12에 꼭 맞는 인코더가 데이터 스트림(701) 내로 낮은-복잡도 방식 또는 고효율 방식의 활성을 신호전달하도록 구성되는 삽입기(700), 미디어 데이터(705)를 정수 값 구문 요소를 포함하는 구문 요소들의 시퀀스(706) 내로 프리코딩(precoding)하도록 구성되는 구성자(504), 및 정수 값 구문 요소들의 도메인을 심벌 시퀀스 워드들의 공-도메인에 매핑하기 위하여, 제어 파라미터에 의해 제어가능한 함수의 매핑을 사용하여 정수 값 구문 요소를 기호화하도록 구성되는 심벌라이저(707)를 포함하는 것과 같이 도 14에 도시되며, 심벌라이저(707)는 고효율 방식이 활성화되는 경우에 제 1 비율로 데이터 스트림에 따라 제어 파라미터가 변경되고 제어 파라미터는 화살표(708)에 의해 도시된 것과 같이 낮은-복잡도 방식이 활성화되는 경우에서의 제 1 비율보다 낮은 제 2 비율로, 데이터 스트림과 관계없이 일정하거나 또는 데이터 스트림에 따라 변경하는 것과 같이 기호화를 실행하도록 구성된다. 결과는 데이터스트림(701) 내로 코딩된다.
- [0574] 다시, 도 14의 실시 예는 위에 설명된 컨텍스트-적응적 이진 산술 인코딩/디코딩 실시 예 상으로 쉽게 전달될 수 있다는 것을 이해하여야 한다: 선택기(509)와 엔트로피 인코더들(310)은 데이터 스트림(401)을 직접적으로 출력하고 현재 데이터스트림으로부터 유래하는 빈을 위한 선택할 수 있는 컨텍스트-적응적 이진 산술 인코더 내로 압축할 수 있다. 이는 특히 컨텍스트 적응성 및/또는 확률 적응성에 있어서 사실이다.
- [0576] 위에서 대안의 실시 예들에 따라, 위의 실시 예들 중 일부와 관련하여 설명된 모드 스위칭 능력은 제거된다는 것을 간단히 설명하였다. 이를 명확히 하기 위하여, 단지 모드 스위칭 능력의 제거가 위의 실시 예들로부터 도 16의 실시 예를 구별하는 것에 대하여 위의 설명을 요약하는, 도 16이 참조된다. 게다가, 다음의 설명은 예를 들면, H.264와 비교하여 슬로프와 오프셋을 위한 덜 정확한 파라미터들을 사용하는 컨텍스트들의 확률 추정들의 초기화로부터 야기하는 장점들을 나타낼 것이다.
- [0578] 특히, 도 16은 움직임 벡터 차분값들의 수평과 수직 컴포넌트들이 움직임 벡터 차분값들의 수평과 수직 컴포넌트들의 이진화들을 사용하여 코딩되는 데이터 스트림(401)으로부터 비디오(405)를 디코딩하기 위한 디코더를 도시하며, 이진화들은 절단 값 아래의 수평과 수직 컴포넌트들의 도메인의 첫 번째 간격 내에서, 각각 수평과 수직 컴포넌트들의 절단된 단항 코드, 및 절단된 단항 코드 형태의 접두부호의 조합과 동일화한다. 각각 수평과 수직 컴포넌트들의 지수 골롬 코드의 형태의 절단 값과 접미부호는 수평과 수직 컴포넌트들의 도메인의 두 번째 간격 내에 그리고 절단 값 위에 포함되며, 절단 값은 2이고 지수 골롬 코드는 1의 순차를 갖는다. 디코더는 움직임 벡터 차분값들의 수평과 수직 컴포넌트들을 위하여, 절단된 단항 코드의 빈 위치 당 정확하게 하나의 컨텍스트를 갖는 컨텍스트-적응적 이진 엔트로피 디코딩을 사용하여 데이터 스트림으로부터 절단된 단항 코드, 및 움직임 벡터 차분값들의 이진화를 획득하기 위하여 일정한 등-확률 바이패스 방식(equi-probability bypass mode)를 사용하는 지수 골롬 코드를 유래하도록 구성되는 엔트로피 디코더(409)를 포함한다. 더 정확히 설명하면, 위에서 설명된 것과 같이 엔트로피 디코더(409)는 위에서 언급된 CABAC 전략과 같은 이진 엔트로피 디코딩을 사용하거나, 또는 이진 파이프 디코딩, 즉, 각각의 선택기/할당기와 함께 일부 병렬 운용 엔트로피 디코더들(322)을 포함하는 구성을 사용하여 데이터 스트림(401)으로부터 이진화들의 빈들(326)의 수를 유래하도록 구성될 수 있다. 디심벌라이저(314)는 움직임 벡터 차분값들의 수평과 수직 컴포넌트들의 정수 값들을 획득하기 위하여 움직임 벡터 차분값 구문 요소들의 이진화들을 역이진화하며, 재구성자(404)는 움직임 벡터 차분값들의 수평과 수직 컴포넌트들의 정수 값들을 기초로 하여 비디오를 재구성한다.
- [0579] 이를 더 상세히 설명하기 위하여, 대표적으로 하나의 움직임 벡터 차분값, 즉, 예측된 움직임 벡터와 실제/재구성된 움직임 벡터 사이의 예측 잔류를 표현하는 벡터를 도시한 도 18이 간단하게 참조된다. 수평과 수직 컴포넌트들(802_x , 및 802_y)이 또한 도시된다. 그것들은 픽셀 위치들의 단위, 즉, 픽셀 피치, 또는 픽셀 피치의 반 혹은 $1/4$ 등과 같은 서브픽셀 위치들의 단위로 전송될 수 있다. 수평과 수직 컴포넌트들($802_{x,y}$)은 정수 값이다. 그것들의 도메인은 제로부터 무한까지 도달한다. 부호 값은 개별적으로 처리될 수 있으며 여기서는 더 이상 고려되지 않는다. 바꾸어 말하면, 여기에 설명되는 설명은 움직임 벡터 차분값들($802_{x,y}$)의 크기에 초점을 맞춘다. 도메인은 804에 도시된다. 도 19는 서로 상에 수직으로 배치되는 컴포넌트($802_{x,y}$)의 가능한 값들과 관련된 도메인 축(804)의 오른쪽 면 상에서, 각각의 가능한 값이 매핑되는(이진화되는) 이진화들을 도시한다. 도시된 것과 같이, 절단된 단항 코드(806)만이 2의 절단 값 아래에 발생하며, 반면에 이진화는 접미부호로서, 또한 절단 값-1

위의 정수 값이 나머지를 위한 이진화를 계속하기 위하여 2의 절단 값과 동일하거나 또는 이보다 큰 가능한 값들로부터의 순차(808)의 지수 곱셈 코드를 갖는다. 모든 빈들을 위하여, 단지 두 개의 콘텍스트가 제공된다: 수평과 수직 컴포넌트들(802_{x,y})의 이진화의 첫 번째 빈 위치를 위한 하나, 및 두 수평과 수직 컴포넌트들(802_{x,y})의 절단된 단항 코드(806)의 두 번째 빈 위치를 위한 또 다른 하나. 지수 곱셈 코드(808)의 빈 위치를 위하여, 엔트로피 디코더(409)에 의해 등-확률 바이패스 방식이 사용된다. 즉, 두 빈 값들은 동등하게 확률을 발생하는 것으로 가정된다. 이러한 빈들을 위한 확률 추정은 고정된다. 그것에 비하여, 방금 언급된 절단된 단항 코드(806)의 빈들의 두 콘텍스트와 관련된 확률 추정이 디코딩 동안에 연속적으로 적용된다.

[0581]

위의 설명에 따라, 엔트로피 디코더(409)가 방금 언급된 작업들을 실행하기 위하여 어떻게 구현될 수 있는지에 대하여, 더 상세히 설명하기 전에, 설명은 이제 화살표(810)를 사용하여 도 18에 도시된 재-이진화를 갖는 코드들(106 및 108)의 빈들을 재-이진화함으로써 디심벌라이저(314)에 의해 획득되는 것과 같이 움직임 벡터 차분값들(800)과 그것들이 정수 값을 사용하는 재구성자(404)의 가능한 구현에 초점을 맞춘다. 특히, 재구성자(404)는 위에서 설명된 것과 같이, 데이터 스트림(401)으로부터 현재 재구성되는 픽처의, 이 중에서 적어도 일부가 움직임-보상 예측의 대상이 되는 블록들 내로의 세분과 관련한 정보를 검색할 수 있다. 도 19는 820에서 대표적으로 재구성되는 픽처 및 움직임-보상 예측이 822에서 그 안의 픽처 콘텐츠를 예측하도록 사용되는 픽처(120)의 앞서 언급된 세분의 블록들을 도시한다. 도 2a-2c와 관련하여 설명된 것과 같이, 블록들(122)의 세분과 크기들을 위하여 서로 다른 가능성이 존재한다. 이러한 블록들(122)을 위한 움직임 벡터 차분값(800)을 위한 전송을 방지하기 위하여, 재구성자(404)는 데이터 스트림이 세분이 고정된다는 사실에 더하여, 세분 정보에 더하여, 또는 세분 정보 없이, 부가적으로 머지(merge) 정보를 전송하는 머지 개념을 이용할 수 있다. 머지 정보는 어떠한 블록들(822)이 머지 그룹들을 형성하는지에 관하여 재구성자(404)로 신호를 전달한다. 이러한 측정에 의해, 재구성자(404)가 특정 움직임 벡터 차분값을 블록들(822)의 전체 머지 그룹에 적용하는 것이 가능하다. 자연적으로, 인코딩 면에서, 머지 정보의 전송은 세분 전송 오버헤드(만일 존재하면), 머지 정보 전송 오버헤드 및 머지 그룹들의 크기의 증가에 따라 감소하는 움직임 벡터 차분값 전송 오버헤드 사이의 균형의 대상이다. 다른 한편으로, 머지 그룹 당 블록들의 수의 증가는 이러한 머지 그룹을 위한 움직임 벡터 차분값의 각각의 머지 그룹의 개별 블록들의 실제 필요들로의 적용을 감소시키며 그렇게 함으로써 이러한 블록들의 움직임 벡터 차분값들의 덜 정확한 움직임-보상 예측들을 생산하고 예를 들면, 변환 계수 레벨 형태의 예측 잔류를 전송하기 위한 더 높은 전송 오버헤드를 필요로 한다. 따라서, 적절한 방식으로, 인코딩 면 상에 균형이 발견된다. 그러나, 어떠한 경우라도, 머지 개념은 덜 공간적 급간상관(inter-correlation)을 나타내는 머지 그룹들을 위한 움직임 벡터 차분값을 야기한다. 예를 들면, 멤버십의 특정 머지 그룹으로의 셰이딩(shading)에 의해 도시된 도 18이 참조된다. 확실히, 이러한 블록들 내의 픽처 콘텐츠의 실제 움직임은 각각의 블록들을 병합하도록 디코딩된 인코딩 면의 그것과 유사하였다. 그러나, 다른 머지 그룹들 내의 픽처 콘텐츠의 움직임의 상관성은 낮다. 따라서, 절단된 단항 코드(806)의 빈 당 단지 하나의 콘텍스트를 사용하는 제한은 엔트로피 코딩 효율에 부정적으로 영향을 주지 않는데 그 이유는 이웃하는 픽처 콘텐츠 움직임 사이의 공간적 급간상관을 위하여 머지 개념이 이미 충분히 수용하기 때문이다. 콘텍스트는 단지 빈 위치는 움직임 벡터 차분값 컴포넌트(802_{xy})의 이진화의 일부분이고 빈 위치는 2인 절단 값에 기인하여 1 또는 2인 사실을 기초로 하여 선택될 수 있다. 따라서, 다른 이미 디코딩된 빈들/구문 요소들/움직임 벡터 차분값 컴포넌트들(802_{xy})은 콘텍스트 선택에 영향을 미치지 않는다.

[0583]

유사하게, 재구성자(404)는 각각의 블록 또는 머지 그룹을 위하여 우선 어떠한 움직임 벡터 예측변수들의 리스트가 발생되고 그리고 나서 실제로 움직임 벡터 차분값을 예측하도록 사용되는 예측변수의 지수에 대한 데이터 스트림 정보 내로 분명하게 또는 함축적으로 전송하는 다중-가설 예측 개념(multi-hypothesis prediction concept)을 더(움직임 벡터들의 공간 및/또는 시간 예측을 넘어) 사용함으로써 움직임 벡터 차분값들에 의해 전달되는 정보 콘텐츠를 감소시키도록 구성될 수 있다. 예를 들면, 도 20에서의 비-음영 블록(122)이 참조된다. 재구성자(404)는 왼쪽으로부터, 상부로부터, 이 둘의 조합 등으로부터 같이 공간적으로 움직임 벡터의 예측 및 비디오의 이전에 디코딩된 픽처와 앞서 언급된 예측변수들의 또 다른 조합의 공동-위치된 부의 움직임 벡터로부터 움직임 벡터의 공간적 예측에 의한 것과 같이 이러한 블록의 움직임 벡터를 위하여 서로 다른 예측변수들을 제공할 수 있다. 이러한 예측변수들은 인코딩 면에서 예측할 수 있는 예측가능한 방식으로 재구성자(404)에 의해 분류된다. 이를 위하여 데이터스트림 내에 일부 정보가 전달되며 재구성자에 의해 사용된다. 즉, 예측변수들의 이러한 순차 리스트 중에서 어떤 예측변수가 이러한 블록의 움직임 벡터를 위하여 실제로 사용되어야

만 하는가에 대한, 일부 힌트가 데이터 스트림 내에 포함된다. 그러나, 지수가 우선 예측되고 그리고 나서 단지 그것의 예측이 전송되는 것이 또한 가능하다. 다른 가능성들이 또한 존재한다. 어떠한 경우에서도, 방금 언급된 예측 전략은 현재 블록의 움직임 벡터의 매우 정확한 예측을 가능하게 하며 따라서 움직임 벡터 차분값 내에 부과되는 정보 콘텐츠 요구가 감소된다. 따라서, 절단된 단항 코드(806)의 단지 두 빈들 상으로의 컨텍스트-적응적 엔트로피 코딩의 제한, 도 18과 관련하여 설명된 2로 내려간 절단 값의 감소뿐만 아니라, 1이 되는 지수 곱셈의 순차의 선택은 코딩 효율에 부정적으로 영향을 미치지 않는데, 그 이유는 움직임 벡터 차분값들은 높은 예측 효율에 기인하여, 움직임 벡터 차분값 컴포넌트(802x,y)의 높은 값들이 덜 빈번하게 방문되는 빈도 히스토그램을 나타내기 때문이다. 수평과 수직 컴포넌트들 사이의 구별의 어떠한 생략도 효율적인 예측에 들어맞는데, 그 이유는 예측이 예측 정확성이 높은 두 방향에서 동등하게 잘 운용되는 경향이 있기 때문이다.

[0585]

위의 설명에서, 도 1-15에 제공된 전체 내용은 또한 예를 들면, 디심벌라이저(314), 재구성자(404) 및 엔트로피 디코더(409)의 기능에 관한 한, 도 16에 도시된 엔티티들 상으로 전달가능하다는 것을 이해하는 것이 필수적이다. 그럼에도 불구하고, 완전성을 위하여, 이러한 내용들 중 일부가 아래에 다시 설명된다.

[0587]

방금 설명된 예측 전략의 더 나은 이해를 위하여, 도 20이 참조된다. 방금 설명된 것과 같이, 구성자(404)는 현재 블록(822) 또는 블록들의 현재 머지 그룹을 위한 서로 다른 예측변수들을 획득할 수 있으며, 이러한 예측변수들이 실선 벡터들(824)에 의해 도시된다. 예측변수들은 공간 및/또는 시간 예측에 의해 획득될 수 있는데, 부가적으로 산술 평균 운용 등이 사용될 수 있으며, 따라서 개별 예측변수들은 서로 동일하게 상관하는 방법으로 재구성자(404)에 의해 획득되었을 수 있다. 벡터들(826)이 획득된 것과 독립적으로, 재구성자(404)는 이러한 예측변수들(126)을 순차 리스트 내로 순차화하거나 또는 분류한다. 이는 도 21의 수들 1 내지 4에 의해 도시된다. 만일 분류 과정이 인코더와 디코더가 동시에 운용할 수 있도록 독특하게 결정될 수 있으면 이는 바람직하다. 그리고 나서, 방금 언급된 지수는 분명하게 또는 함축적으로, 데이터 스트림 중에서, 현재 블록 또는 머지 그룹을 위한 재구성자(404)에 의해 획득될 수 있다. 예를 들면, 제 2 예측변수 "2"가 선택되었을 수 있으며 재구성(404)는 움직임 벡터 차분값에 이러한 선택된 예측변수(126)를 더하며, 그렇게 함으로써 그리고 나서 움직임-압축 예측에 의해 현재 블록/머지 그룹의 컨텍스트를 예측하도록 사용되는, 최종적으로 재구성되는 움직임 벡터(128)를 생산한다. 머지 그룹의 경우에 있어서, 머지 그룹의 개별 블록들과 관련하여 움직임 벡터(128)를 더 세분화하기 위하여, 재구성자(404)가 머지 그룹의 블록들을 위하여 제공되는 또 다른 움직임 벡터 차분값을 포함하는 것이 가능할 수 있다.

[0589]

따라서, 도 16에 도시된 엔티티들의 구현들을 더 설명하면, 이는 이진 산술 디코딩 또는 이진 파이프 코딩을 사용하여 데이터 스트림(401)으로부터 절단된 단항 코드(806)를 유래하도록 구성되는 엔트로피 디코더(409)일 수도 있다. 또한, 엔트로피 디코더(409)는 절단된 단항 코드(806)의 두 위치를 위한 서로 다른 컨텍스트들, 또는 대안으로서 두 빈을 위한 동일한 컨텍스트를 사용하도록 구성될 수 있다. 엔트로피 디코더(409)는 확률 상태 업데이트를 실행하도록 구성될 수 있다. 엔트로피 디코더(409)는 현재 절단된 단항 코드(806) 외부에서 유래하는 빈을 위하여, 현재 유래된 빈을 위하여 선택된 컨텍스트와 관련된 현재 확률 상태로부터 현재 유래하는 빈에 따른 새로운 확률 상태로 전환함으로써 이를 수행한다. 위의 테이블들 Next_State_LPS 및 Next_State_MPS 위에서 열거된 다른 단계들 0 내지 5에 더하여 엔트로피 디코더에 의해 실행되는 것에 대한 테이블 룩-업이 참조된다. 위의 논의에서, 현재 확률 상태는 pstate_current에 의해 언급되었다. 이는 관심 있는 각각의 컨텍스트를 위하여 정의된다. 엔트로피 디코더(409)는 현재 확률 간격 폭 값, 즉 R을 양자화하고, 확률 간격 지수, q_index를 획득하기 위하여 현재 확률 간격을 표현하며, 차례로, 현재 확률 간격의 두 부분 간격들로의 세분을 획득하기 위하여, 현재 유래되는 빈을 위하여 선택된 컨텍스트와 관련된 현재 확률 상태에 따라, 확률 간격 지수와 확률 상태 지수, 즉, p_state를 사용하여 테이블 엔티티들 중에서 하나의 테이블 엔티티의 지수화에 의해 간격 세분을 실행함으로써 절단된 단항 코드(806) 중에서 현재 유래되는 빈을 이진 산술 디코딩하도록 구성될 수 있다. 위에서 설명된 실시 예들에서, 이러한 부분 간격들은 가장 가능하고 가장 덜 가능한 심벌과 관련되었다. 위에서 설명된 것과 같이, 엔트로피 디코더(409)는 예를 들면, 두 개 또는 세 개의, 8-비트 표현의 가장 중요한 비트들의 저장 및 현재 확률 간격 폭 값의 양자화로 현재 확률 폭 값(R)을 위한 8-비트 표현을 사용하도록 구성될 수 있다. 엔트로피 디코더(409)는 또한 현재 확률 간격의 내부로부터의 오프셋 값, 즉 V를 기초로 하여 두 부분 간격들 중에서 선택하고, 확률 간격 폭 값(R) 및 오프셋 상태 값을 업데이트하며, 선택된 부분 간격을 사용하여

현재 유래되는 빈의 값을 추론하고 업데이트된 확률 간격 폭 값(R) 및 데이터 스트림(401)으로부터의 계속적인 관독을 포함하는 오프셋 상태 값(V)의 재규격화(renormalization)를 실행하도록 구성될 수 있다. 엔트로피 디코더(409)는 예를 들면, 현재 확률 간격의 두 부분 간격들 내로의 세분을 획득하기 위하여 현재 확률 간격 폭 값을 이등분함으로써 지수 곱셈 코드 중에서 빈을 이진 산술 디코딩하도록 구성될 수 있다. 이등분은 고정된 확률 추정과 상응하고 0.5와 동일하다. 이는 간단한 비트 이동에 의해 구현될 수 있다. 엔트로피 디코더는 또한 각각의 움직임 벡터 차분값을 위하여, 각각의 움직임 벡터 차분값의 수평과 수직 컴포넌트들의 지수 곱셈 코드 이전에, 데이터 스트림(401)으로부터 각각의 움직임 벡터 차분값의 수평과 수직 컴포넌트들의 절단된 단항 코드를 유래하도록 구성될 수 있다. 이러한 측정에 의해, 엔트로피 디코더(409)는 높은 수의 빈들이 확률 추정이 고정된 빈들의 점수, 주로 0.5를 함께 형성하는 것을 이용할 수 있다. 이는 엔트로피 디코딩 과정을 가속화할 수 있다. 다른 한편으로는, 엔트로피 디코더(409)는 단지 우선 하나의 움직임 벡터 차분값의 수평과 수직 컴포넌트들을 유래함으로써 움직임 벡터 차분값들 사이의 순차를 유지하고 그리고 나서 그 다음 움직임 벡터 차분값의 수평과 수직 컴포넌트들을 계속 유래하는 것을 선호한다. 이러한 측정에 의해, 디코딩 엔티티, 즉, 도 16의 디코더 상에 부과되는 메모리 요구들은 감소되는데 그 이유는 디심벌라이저(314)가 또 다른 움직임 벡터 차분값들에 대한 스캔을 기다릴 필요 없이 즉시 움직임 벡터 차분값들의 탈-이진화를 계속하기 때문이다. 이는 콘텍스트 선택에 의해 가능할 수 있게 된다: 코드(806)의 빈 위치 당 단지 정확하게 하나의 콘텍스트가 이용가능하기 때문에, 어떠한 공간적 상관성도 검사될 필요가 없다.

[0591] 재구성자(404)는 위에서 설명된 것과 같이, 움직임 벡터의 수평 및 수직 컴포넌트들을 위한 예측변수들(126)을 획득하기 위하여 공간적으로 및/또는 시간적으로 수평 및/또는 수직 컴포넌트들을 예측할 수 있고 단순히 각각의 예측변수에 대한 움직임 벡터 차분값을 더하는 것에 의한 것과 같이, 움직임 벡터 차분값들의 수평 및 수직 컴포넌트들을 사용하여 예측변수들(826)을 개선함으로써 움직임 벡터들의 수평 및 수직 컴포넌트들을 재구성할 수 있다.

[0593] 또한, 재구성자(404)는 움직임 벡터들의 수평 및 수직 컴포넌트를 위한 예측의 순차 리스트를 획득하기 위하여 서로 다른 방식들로 움직임 벡터들의 수평 및 수직 컴포넌트들을 예측하고, 데이터 스트림으로부터 리스트 지수를 획득하며 리스트 지수가 움직임 벡터 차분값들의 수평 및 수직 컴포넌트들을 사용하여 가리키는 리스트의 예측변수를 개선함으로써 움직임 벡터들의 수평 및 수직 컴포넌트들을 재구성하도록 구성될 수 있다.

[0595] 또한, 위에서 이미 설명된 것과 같이, 재구성자(404)는 움직임 벡터들의 수평 및 수직 컴포넌트들(820x,y)을 비디오의 픽처들의 블록들 내로의 세분화에 의해 정의되는 공간적 입도에 적용함으로써 움직임 보상 예측을 사용하여 비디오를 재구성하도록 구성될 수 있으며, 재구성자(404)는 병합 그룹들 내로 블록들을 그룹화하고 병합 그룹들의 유닛에서, 이진화기(314)에 의해 획득되는 움직임 벡터 차분값들의 수평 및 수직 컴포넌트들(820x,y)의 정수 값들을 적용하기 위하여 데이터 스트림(401) 내에 존재하는 구문 요소들의 병합을 사용할 수 있다.

[0597] 재구성자(404)는 구문 요소들의 병합을 배제하는 데이터 스트림(401)의 일부로부터 블록들 내로 비디오 픽처들의 세분화를 유도할 수 있다. 재구성자(404)는 또한 관련 병합 그룹의 모든 블록들을 위한 미리 결정된 움직임 벡터의 수평 및 수직 컴포넌트들을 적용할 수 있거나, 또는 병합 그룹의 블록들과 관련된 움직임 벡터 차분값들의 수평 및 수직 컴포넌트들에 의해 이를 개선할 수 있다.

[0599] 완전성만을 위하여, 도 17은 도 16의 디코더에 맞는 인코더를 도시한다. 도 17의 인코더는 구성자(504), 심벌라이저(507) 및 엔트로피 인코더(513)를 포함한다. 인코더는 움직임 벡터들을 예측하고 예측된 움직임 벡터들의 예측 에러를 표현하기 위하여 움직임 벡터 차분값들의 수평 및 수직 컴포넌트들의 정수 값들(506)을 설정함으로써 움직임 벡터들을 사용하고 움직임 벡터들을 예측으로 코딩하는 움직임 보상 예측에 의해 비디오를 예측으로 코딩하도록 구성되는 구성자(504); 움직임 벡터 차분값들의 수평 및 수직 컴포넌트들의 이진화들(508)을 획득하기 위하여 정수 값들을 이진화하도록 구성되는 심벌라이저(507)를 구비하되, 이진화들은 절단 값 아래의 수평 및 수직 컴포넌트들의 도메인의 제 1 간격 내에서, 각각 수평 및 수직 컴포넌트들의 절단된 단항 코드와, 절단 값 위의 수평 및 수직 컴포넌트들의 도메인의 제 2 간격 내에서, 각각 절단 값을 위한 절단된 단항 코드 형태의

접두부호와 수평 및 수직 컴포넌트들의 지수-골롬 형태의 접미부호의 조합을 동일시하며, 절단 값은 2이고 지수-골롬 코드는 순차 1(차수 1, order one)을 가지며; 움직임 벡터 차분값들의 수평 및 수직 컴포넌트들을 위하여, 움직임 벡터 차분값들의 수평 및 수직 컴포넌트들에 일반적인, 절단된 단항 코드, 및 일정한 등-확률 바이패스 방식을 사용하는 지수-골롬 코드의 빈 위치 당 정확히 하나의 컨텍스트를 갖는 컨텍스트 적용성 이진 인코딩을 사용하여 절단된 단항 코드를 데이터 스트림 내로 인코딩하도록 구성되는 엔트로피 인코더(513)를 포함한다. 추가의 가능한 구현의 세부내용이 도 16의 디코더와 관련된 설명으로부터 도 17의 인코더 상으로 직접 이동될 수 있다.

- [0601] 비록 장치의 맥락에서 일부 양상들이 설명되었으나, 이러한 양상들은 또한 블록 또는 장치가 방법 단계 또는 방법 단계의 특징에 상응하는, 상응하는 방법의 설명을 나타내는 것이 자명하다. 유사하게, 방법 단계의 맥락에서 설명된 양상들은 또한 상응하는 장치의 상응하는 블록 또는 아이템 또는 특징을 나타낸다. 일부 또는 모든 방법 단계들은 예를 들면, 마이크로프로세서, 프로그램가능 컴퓨터 또는 전자 회로 같은, 하드웨어 장치에 의해(또는 사용하여) 실행될 수 있다. 일부 실시 예들에서, 일부 하나 또는 그 이상의 가장 중요한 방법 단계들이 그러한 장치에 의해 실행될 수 있다.
- [0603] 본 발명의 인코딩된 신호는 디지털 저장 매체 상에 저장될 수 있거나 또는 무선 전송 매체 혹은 인터넷과 같은 유선 전송 매체와 같은 전송 매체 상에 전송될 수 있다.
- [0605] 특정 구현 필요성에 따라, 본 발명의 실시 예들은 하드웨어 또는 소프트웨어에서 구현될 수 있다. 구현은 디지털 저장 매체, 예를 들면, 거기에 저장되는 전자적으로 판독가능한 신호들을 갖는, 플로피 디스크, DVD, CD, ROM, PROM, EPROM, EEPROM 또는 플래시 메모리를 사용하여 실행될 수 있는데, 이는 각각의 방법이 실행되는 것과 같이 프로그램가능 컴퓨터 시스템과 협력한다(또는 협력할 수 있다). 따라서 디지털 저장 매체는 컴퓨터 판독가능할 수 있다.
- [0607] 본 발명에 따른 일부 실시 예들은 여기에 설명된 방법들 중의 하나가 실행되는 것과 같이, 프로그램가능 컴퓨터 시스템과 협력할 수 있는, 전자적으로 판독가능한 제어 신호들을 갖는 비-일시적 데이터 캐리어를 포함한다.
- [0609] 일반적으로, 본 발명의 실시 예들은 프로그램 코드를 갖는 컴퓨터 프로그램 제품으로서 구현될 수 있는데, 프로그램 코드는 컴퓨터 프로그램 제품이 컴퓨터 상에 구동될 때 방법들 중의 하나를 실행하도록 작동할 수 있다. 프로그램 코드는 예를 들면 기계 판독가능 캐리어 상에 저장될 수 있다.
- [0611] 다른 실시 예들은 기계 판독가능 캐리어 상에 저장되는, 여기에 설명된 방법들 중의 하나를 실행하기 위한 컴퓨터 프로그램을 포함한다.
- [0613] 바꾸어 말하면, 따라서 본 발명의 방법의 일 실시 예는 컴퓨터 프로그램이 컴퓨터상에 구동할 때, 여기에 설명된 방법들 중의 하나를 실행하기 위한 프로그램 코드를 갖는 컴퓨터 프로그램이다.
- [0615] 본 발명의 방법의 또 다른 실시 예는 따라서 여기에 설명된 방법들 중의 하나를 실행하기 위하여 그것에 대해 기록된, 컴퓨터 프로그램을 포함하는 데이터 캐리어(또는 디지털 저장 매체, 또는 컴퓨터 판독가능 매체)이다. 데이터 캐리어, 디지털 저장 매체 또는 기록된 매체는 일반적으로 유형(tangible) 및/또는 비-일시적이다.
- [0617] 본 발명의 방법의 또 다른 실시 예는 따라서 여기에 설명된 방법들 중의 하나를 실행하기 위한 컴퓨터 프로그램을 표현하는 신호들의 데이터 스트림 또는 시퀀스이다. 신호들의 데이터 스트림 또는 시퀀스는 예를 들면 데이

터 통신 연결, 예를 들면 인터넷을 거쳐 전달되도록 구성될 수 있다.

- [0619] 또 다른 실시 예는 처리 수단들, 예를 들면, 여기에 설명된 방법들 중의 하나를 실행하거나 적용하도록 구성되는 컴퓨터, 또는 프로그램가능 논리 장치를 포함한다.
- [0621] 또 다른 실시 예는 여기에 설명된 방법들 중의 하나를 실행하기 위하여 거기에 설치된 컴퓨터 프로그램을 갖는 컴퓨터를 포함한다.
- [0623] 본 발명에 따른 또 다른 실시 예는 여기에 설명된 방법들 중 하나를 수신기에 실행하도록 컴퓨터 프로그램을 전달하도록(예를 들면, 전자적으로 또는 광학적으로) 구성되는 장치 또는 시스템을 포함한다. 수신기는 예를 들면, 컴퓨터, 이동 기기, 메모리 장치 등일 수 있다. 장치 또는 시스템은 컴퓨터 프로그램을 수신기에 전달하기 위한 파일 서버를 포함할 수 있다.
- [0625] 일부 실시 예들에서, 프로그램가능 논리 장치(예를 들면, 필드 프로그램가능 게이트 어레이)는 여기에 설명된 방법들의 기능들이 일부 또는 모두를 실행하도록 사용될 수 있다. 일부 실시 예들에서, 필드 프로그램가능 게이트 어레이는 여기에 설명된 방법들 중의 하나를 실행하기 위하여 마이크로프로세서와 협력할 수 있다. 일반적으로, 방법들은 바람직하게는 어떠한 하드웨어 장치에 의해 실행된다.
- [0627] 위에서 설명된 실시 예들은 단지 본 발명의 원리를 설명하기 위한 것이다. 여기에 설명된 배치들 및 내용들의 변형 및 변경들은 통상의 지식을 가진 자들에 자명할 것이라는 것을 이해하여야 한다. 따라서, 본 발명의 실시 예들의 설명에 의해 표현된 특정 상세 내용에 의한 것이 아니라 첨부된 청구항들의 범위에 의해서만 한정되는 것으로 의도된다.
- [0629] 본 분할출원은 최초 원출원의 청구범위를 실시예로서 아래에 기재하였다.
- [0631] [실시예 1]
- [0632] 수평과 수직 컴포넌트들의 이진화들을 사용하여 움직임 벡터 차분값들의 상기 수평과 수직 컴포넌트들이 코딩되는 데이터 스트림으로부터 비디오를 디코딩하기 위한 디코더에서, 상기 이진화들은 절단 값 아래의 상기 수평과 수직 컴포넌트들의 도메인의 첫 번째 간격 내에서, 각각 상기 수평과 수직 컴포넌트들의 절단된 단항 코드와, 상기 절단 값을 포함하고 상기 절단 값 위의 상기 수평과 수직 컴포넌트들의 상기 도메인의 두 번째 간격 내에서, 각각 상기 절단 값을 위한 상기 절단된 단항 코드 형태의 접두부호와 상기 수평과 수직 컴포넌트들의 지수-골롬 형태의 접미부호의 조합을 동일시하며, 상기 절단 값은 2이고 상기 지수-골롬 코드는 순차 1을 가지는 비디오를 디코딩하기 위한 디코더에 있어서:
- [0633] 상기 움직임 벡터 차분값들의 상기 수평과 수직 컴포넌트들을 위하여, 움직임 벡터 차분값들의 수평과 수직 컴포넌트들에 통상적인, 절단된 단항 코드의 위치 당 정확하게 하나의 콘텍스트를 갖는 콘텍스트-적응적 이진 엔트로피 디코딩을 사용하여 데이터 스트림으로부터의 상기 절단된 단항 코드 및, 움직임 벡터 차분값들의 이진화를 획득하기 위하여 일정한 등-확률 바이패스 방식을 사용하여 지수-골롬 코드를 유래하도록 구성되는 엔트로피 디코더;
- [0634] 상기 움직임 벡터 차분값들의 상기 수평과 수직 컴포넌트들의 정수 값들을 획득하기 위하여 상기 움직임 벡터 차분값 구문 요소들의 이진화들을 탈이진화하도록 구성되는 디실벌라이저;
- [0635] 상기 움직임 벡터 차분값들의 상기 수평과 수직 컴포넌트들의 정수 값들을 기초로 하여 상기 비디오를 재구성하도록 구성되는 재구성자;를 포함하는 것을 특징으로 하는 디코더.

- [0637] [실시예 2]
- [0638] 제 1실시예에 있어서, 상기 엔트로피 디코더(409)는 이진 산술 디코딩 또는 파이프 디코딩을 사용하여 상기 데이터 스트림(401)으로부터 상기 절단된 단항 코드(806)를 유래하도록 구성되는 것을 특징으로 하는 디코더.
- [0640] [실시예 3]
- [0641] 제 1실시예 또는 2실시예에 있어서, 상기 엔트로피 디코더(409)는 상기 절단된 단항 코드(806)의 상기 두 빈 위치들을 위하여 서로 다른 컨텍스트들을 사용하도록 구성되는 것을 특징으로 하는 디코더.
- [0643] [실시예 4]
- [0644] 제 1실시예 내지 3실시예 중 어느 한 실시예에 있어서, 상기 엔트로피 디코더(409)는 상기 절단된 단항 코드(806) 중에서 현재 유래하는 빈을 위하여, 상기 현재 유래되는 빈을 위하여 선택된 컨텍스트와 관련된 현재 확률 상태에서부터 상기 현재 유래되는 빈에 따라 새로운 확률 상태로 이동시킴으로써 확률 상태 업데이트를 실행하도록 구성되는 것을 특징으로 하는 디코더.
- [0646] [실시예 5]
- [0647] 제 1실시예 내지 4실시예 중 어느 한 실시예에 있어서, 상기 엔트로피 디코더(409)는 확률 간격 지수를 획득하기 위하여 현재 확률 간격을 표현하는 현재 확률 간격 폭 값을 양자화하고, 상기 현재 확률 간격의 두 부분 간격들 내로의 세분을 획득하기 위하여 현재 유래되는 상기 빈을 위하여 선택된 상기 컨텍스트와 관련된 현재 확률 상태에 따라 확률 간격 지수와 확률 상태 지수를 사용하여 테이블 엔트리 중에서 하나의 테이블 엔트리의 지수화에 의해 간격 세분을 실행함으로써, 상기 절단된 단항 코드(806) 중에서 현재 유래되는 빈을 이진 산술 디코딩하도록 구성되는 것을 특징으로 하는 디코더.
- [0649] [실시예 6]
- [0650] 제 5실시예에 있어서, 상기 엔트로피 디코더(409)는 상기 현재 확률 간격 폭 값을 위하여 8 비트 표현을 사용하고 상기 현재 확률 폭 값을 양자화하는데 상기 8 비트 표현의 2개 또는 3개의 가장 중요한 비트들을 저장하도록 구성되는 것을 특징으로 하는 디코더.
- [0652] [실시예 7]
- [0653] 제 5실시예 또는 6실시예에 있어서, 상기 엔트로피 디코더(409)는 상기 현재 확률 간격의 내부로부터의 오프셋 상태 값을 기초로 하여 상기 두 부분 간격들 중에서 선택하고, 상기 확률 간격 폭 값과 상기 오프셋 상태 값을 업데이트하며, 상기 선택된 부분 간격을 사용하여 현재 유래되는 상기 빈의 값을 추론하며, 상기 데이터 스트림(401)으로부터의 비트들의 판독의 연속을 포함하는 상기 업데이트된 확률 간격 폭 값과 상기 오프셋 상태 값의 재규격화를 실행하도록 구성되는 것을 특징으로 하는 디코더.
- [0655] [실시예 8]
- [0656] 제 5실시예 내지 7실시예 중 어느 한 실시예에 있어서, 상기 엔트로피 디코더(409)는 일정한 등-확률 바이패스 방식에서, 상기 현재 확률 간격의 두 부분 간격들 내로의 세분을 획득하기 위하여 상기 현재 확률 간격 폭 값을 이등분함으로써 상기 지수-골룰 코드 중에서 하나의 빈을 이진 산술 디코딩하도록 구성되는 것을 특징으로 하는 디코더.
- [0658] [실시예 9]

- [0659] 제 1실시에 내지 8실시에 중 어느 한 실시예에 있어서, 상기 엔트로피 디코더(409)는 각각의 움직임 벡터 차분값을 위하여, 상기 각각의 움직임 벡터 차분값의 상기 수평과 수직 컴포넌트들의 상기 지수-골롬 코드 전에, 상기 데이터 스트림으로부터 상기 각각의 움직임 벡터 차분값의 상기 수평과 수직 컴포넌트들의 상기 절단된 단항 코드를 유래하도록 구성되는 것을 특징으로 하는 디코더.
- [0661] [실시예 10]
- [0662] 제 1실시에 내지 9실시에 중 어느 한 실시예에 있어서, 상기 재구성자는 움직임 벡터들의 상기 수평과 수직 컴포넌트들을 위한 예측 변수를 획득하고 상기 움직임 벡터 차분값들의 상기 수평과 수직 컴포넌트들을 사용하여 상기 예측변수들(826)을 세분화함으로써 상기 움직임 벡터들의 상기 수평과 수직 컴포넌트들을 재구성하기 위하여 상기 움직임 벡터들의 상기 수평과 수직 컴포넌트들을 공간적으로 및/또는 시간적으로 예측하도록 구성되는 것을 특징으로 하는 디코더.
- [0664] [실시예 11]
- [0665] 제 1실시에 내지 10실시에 중 어느 한 실시예에 있어서, 상기 재구성자는 상기 움직임 벡터들의 수평과 수직 컴포넌트들을 위한 예측들의 순차 리스트를 획득하기 위하여 서로 다른 방법들로 상기 움직임 벡터들의 수평과 수직 컴포넌트들을 예측하고, 상기 데이터 스트림으로부터 리스트 지수를 획득하며, 상기 리스트 지수가 상기 움직임 벡터 차분값들의 상기 수평과 수직 컴포넌트들을 사용하여 나타내는 상기 예측변수를 세분화함으로써 상기 움직임 벡터들의 수평과 수직 컴포넌트들을 재구성하도록 구성되는 것을 특징으로 하는 디코더.
- [0667] [실시예 12]
- [0668] 제 10실시에 또는 11실시에에 있어서, 상기 재구성자는 상기 움직임 벡터들의 수평과 수직 컴포넌트들의 사용에 의해 움직임-보상 예측을 사용하여 상기 비디오를 재구성하도록 구성되는 것을 특징으로 하는 디코더.
- [0670] [실시예 13]
- [0671] 제 12실시에에 있어서, 상기 재구성자는 블록들 내의 상기 비디오의 픽처들의 세분화에 의해 정의되는 공간 입도에서 상기 움직임 벡터들의 수평과 수직 컴포넌트들의 적용에 의해 움직임-보상 예측을 사용하여 상기 비디오를 재구성하도록 구성되며, 상기 재구성자는 상기 블록들을 머지 그룹들 내로 분류하고 머지 그룹들의 유닛들로, 상기 탈이진화기에 의해 획득된 상기 움직임 벡터 차분값들의 상기 수평과 수직 컴포넌트들의 상기 정수 값들을 적용하기 위하여 상기 데이터 스트림 내에 존재하는 머지 구문 요소들을 사용하는 것을 특징으로 하는 디코더.
- [0673] [실시예 14]
- [0674] 제 13실시에에 있어서, 상기 재구성자는 상기 머지 구문 요소들을 제외하고 상기 데이터 스트림의 일부로부터 블록들 내의 상기 비디오의 픽처들의 세분을 유래하도록 구성되는 것을 특징으로 하는 디코더.
- [0676] [실시예 15]
- [0677] 제 13실시에 또는 14실시에에 있어서, 상기 재구성자는 관련된 머지 그룹의 모든 블록을 위하여 미리 결정된 움직임 벡터의 상기 수평과 수직 컴포넌트들을 적용하도록 구성되거나, 또는 상기 머지 그룹의 블록들과 관련된 상기 움직임 벡터 차분값들의 상기 수평과 수직 컴포넌트들에 의해 이를 세분화하도록 구성되는 것을 특징으로 하는 디코더.
- [0679] [실시예 16]

- [0680] 데이터 스트림 내로 비디오를 인코딩하기 위한 인코더에 있어서,
- [0681] 움직임 벡터들을 사용하는 움직임 보상 예측에 의해 상기 비디오를 예측으로 코딩하고, 상기 움직임 벡터들을 예측하고 상기 예측된 움직임 벡터들의 예측 오차를 표현하기 위하여 움직임 벡터 차분값들의 수평과 수직 컴포넌트들의 정수 값의 설정함으로써 상기 움직임 벡터들을 예측으로 코딩하도록 구성되는 것을 특징으로 하는 구성자;
- [0682] 상기 움직임 벡터 차분값들의 상기 수평과 수직 컴포넌트들의 이진화들을 획득하기 위하여 상기 정수 값들을 이진화하도록 구성되는 심벌라이저를 구비하되, 상기 이진화들은 절단 값 아래의 상기 수평과 수직 컴포넌트들의 도메인의 첫 번째 간격 내에서, 각각 상기 수평과 수직 컴포넌트들의 절단된 단항 코드와, 상기 절단 값을 포함하고 상기 절단 값 위의 상기 수평과 수직 컴포넌트들의 상기 도메인의 두 번째 간격 내에서, 각각 상기 절단 값을 위한 상기 절단된 단항 코드 형태의 접두부호와 상기 수평과 수직 컴포넌트들의 지수-골롬 형태의 접미부호의 조합을 동일시하며, 상기 절단 값은 2이고 상기 지수-골롬 코드는 순차 1을 가지며;
- [0683] 상기 움직임 벡터 차분값들의 상기 수평과 수직 컴포넌트들을 위하여, 움직임 벡터 차분값들의 수평과 수직 컴포넌트들에 통상적인, 절단된 단항 코드의 위치 당 정확하게 하나의 콘텍스트를 갖는 콘텍스트-적응적 이진 엔트로피 디코딩을 사용하여 데이터 스트림으로부터의 상기 절단된 단항 코드 및, 움직임 벡터 차분값들의 이진화를 획득하기 위하여 일정한 등-확률 바이패스 방식을 사용하여 지수-골롬 코드를 인코딩하도록 구성되는 엔트로피 인코더;를 포함하는 것을 특징으로 하는 인코더.
- [0685] [실시예 17]
- [0686] 제 16실시예에 있어서, 상기 엔트로피 인코더는 이진 산술 인코딩 또는 파이프 인코딩을 사용하여 상기 절단된 단항 코드를 상기 데이터 스트림 내로 인코딩하도록 구성되는 것을 특징으로 하는 인코더.
- [0688] [실시예 18]
- [0689] 제 16실시예 또는 17실시예에 있어서, 상기 엔트로피 인코더는 상기 절단된 단항 코드(806)의 상기 두 빈 위치들을 위하여 서로 다른 콘텍스트들을 사용하도록 구성되는 것을 특징으로 하는 인코더.
- [0691] [실시예 19]
- [0692] 제 16실시예 내지 18실시예 중 어느 한 실시예에 있어서, 상기 엔트로피 인코더는 상기 절단된 단항 코드 중에서 현재 인코딩하는 빈을 위하여, 상기 현재 인코딩되는 빈을 위하여 선택된 콘텍스트와 관련된 현재 확률 상태로부터 상기 현재 유래되는 빈에 따라 새로운 확률 상태로 이동시킴으로써 확률 상태 업데이트를 실행하도록 구성되는 것을 특징으로 하는 인코더.
- [0694] [실시예 20]
- [0695] 제 16실시예 내지 19실시예 중 어느 한 실시예에 있어서, 상기 엔트로피 인코더는 확률 간격 지수를 획득하기 위하여 현재 확률 간격을 표현하는 현재 확률 간격 폭 값을 양자화하고, 상기 현재 확률 간격의 두 부분 간격들 내로의 세분을 획득하기 위하여 현재 유래되는 상기 빈을 위하여 선택된 상기 콘텍스트와 관련된 현재 확률 상태에 따라 확률 간격 지수와 확률 상태 지수를 사용하여 테이블 엔트리 중에서 하나의 테이블 엔트리의 지수화에 의해 간격 세분을 실행함으로써, 상기 절단된 단항 코드(806) 중에서 현재 유래되는 빈을 이진 산술 디코딩하도록 구성되는 것을 특징으로 하는 인코더.
- [0697] [실시예 21]
- [0698] 제 20실시예에 있어서, 상기 엔트로피 인코더는 상기 현재 확률 간격 폭 값을 위하여 8 비트 표현을 사용하고 상기 현재 확률 폭 값을 양자화하는데 상기 8 비트 표현의 2개 또는 3개의 가장 중요한 비트들을 저장하도록 구성되는 것을 특징으로 하는 인코더.

- [0700] [실시예 22]
- [0701] 제 20실시예 또는 21실시예에 있어서, 상기 엔트로피 인코더는 상기 현재 인코딩되는 bin의 정수 값을 기초로 하여 상기 두 부분 간격들 중에서 선택하고, 상기 확률 간격 폭 값과 상기 오프셋 상태 값을 업데이트하며, 상기 선택된 부분 간격을 사용하여 상기 확률 간격 폭 값과 확률 간격 오프셋을 업데이트하며, 상기 데이터 스트림 (401)으로부터의 비트들의 기록의 연속을 포함하는 상기 확률 간격 폭 값과 상기 오프셋 상태 값의 재규격화를 실행하도록 구성되는 것을 특징으로 하는 인코더.
- [0703] [실시예 23]
- [0704] 제 20실시예 내지 21실시예 중 어느 한 실시예에 있어서, 상기 엔트로피 인코더는 상기 현재 확률 간격의 두 부분 간격들 내로의 세분을 획득하기 위하여 상기 현재 확률 간격 폭 값을 이등분함으로써 상기 지수-골롬 코드 중에서 하나의 bin을 이진 산술 인코딩하도록 구성되는 것을 특징으로 하는 인코더.
- [0706] [실시예 24]
- [0707] 제 16실시예 내지 23실시예 중 어느 한 실시예에 있어서, 상기 엔트로피 인코더는 각각의 움직임 벡터 차분값을 위하여, 상기 각각의 움직임 벡터 차분값의 상기 수평과 수직 컴포넌트들의 상기 지수-골롬 코드 전에, 상기 각각의 움직임 벡터 차분값의 상기 수평과 수직 컴포넌트들의 상기 절단된 단항 코드를 상기 데이터 스트림 내로 인코딩하도록 구성되는 것을 특징으로 하는 인코더.
- [0709] [실시예 25]
- [0710] 제 16실시예 내지 24실시예 중 어느 한 실시예에 있어서, 상기 구성자는 움직임 벡터들의 상기 수평과 수직 컴포넌트들을 위한 예측 변수를 획득하기 위하여 상기 움직임 벡터들의 상기 수평과 수직 컴포넌트들을 공간적으로 및/또는 시간적으로 예측하고, 상기 움직임 벡터 차분값들의 상기 수평과 수직 컴포넌트들을 향하여 상기 예측변수들을 세분화하기 위하여 상기 움직임 벡터들의 상기 수평과 수직 컴포넌트들을 결정하도록 구성되는 것을 특징으로 하는 인코더.
- [0712] [실시예 26]
- [0713] 제 16실시예 내지 25실시예 중 어느 한 실시예에 있어서, 상기 구성자는 상기 움직임 벡터들의 수평과 수직 컴포넌트들을 위한 예측들의 순차 리스트를 획득하기 위하여 서로 다른 방법들로 상기 움직임 벡터들의 수평과 수직 컴포넌트들을 예측하고, 리스트 지수를 결정하며, 상기 리스트 지수를 나타내는 정보를 상기 데이터 스트림 내로 삽입하며, 상기 리스트 지수가 상기 움직임 벡터 차분값들의 상기 수평과 수직 컴포넌트들을 향하여 나타내는 상기 예측변수를 세분화하기 위하여 상기 움직임 벡터들의 수평과 수직 컴포넌트들을 결정하도록 구성되는 것을 특징으로 하는 인코더.
- [0715] [실시예 27]
- [0716] 제 16실시예 내지 26실시예 중 어느 한 실시예에 있어서, 상기 구성자는 블록들 내의 상기 비디오의 픽처들의 세분화에 의해 정의되는 공간 입도에서 상기 움직임 벡터들의 수평과 수직 컴포넌트들의 적용에 의해 움직임-보상 예측을 사용하여 상기 비디오를 코딩하도록 구성되며, 상기 구성자는 상기 블록들을 머지 그룹들 내로 분류하고 머지 그룹들의 유닛들로, 상기 탈이진화기에 의해 이진화의 대상이 되는 상기 움직임 벡터 차분값들의 상기 수평과 수직 컴포넌트들의 상기 정수 값을 적용하기 위하여 상기 데이터 스트림 내로 머지 구분 요소들을 결정하고 삽입하는 것을 특징으로 하는 인코더.

- [0718] [실시예 28]
- [0719] 제 27실시예에 있어서, 상기 구성자는 블록들 내의 상기 비디오의 픽처들의 세분을 상기 머징 구문 요소들을 제외하고 상기 데이터 스트림의 일부 내로 코딩하도록 구성되는 것을 특징으로 하는 인코더.
- [0721] [실시예 29]
- [0722] 제 27실시예 또는 28실시예에 있어서, 상기 구성자는 관련된 머지 그룹의 모든 블록을 위하여 미리 결정된 움직임 벡터의 상기 수평과 수직 컴포넌트들을 적용하도록 구성되거나, 또는 상기 머지 그룹의 블록들과 관련된 상기 움직임 벡터 차분값들의 상기 수평과 수직 컴포넌트들에 의해 이를 세분화하도록 구성되는 것을 특징으로 하는 인코더.
- [0724] [실시예 30]
- [0725] 수평과 수직 컴포넌트들의 이진화들을 사용하여 움직임 벡터 차분값들의 상기 수평과 수직 컴포넌트들이 코딩되는 데이터 스트림으로부터 비디오를 디코딩하기 위한 방법에서, 상기 이진화들은 절단 값 아래의 상기 수평과 수직 컴포넌트들의 도메인의 첫 번째 간격 내에서, 각각 상기 수평과 수직 컴포넌트들의 절단된 단항 코드와, 상기 절단 값을 포함하고 상기 절단 값 위의 상기 수평과 수직 컴포넌트들의 상기 도메인의 두 번째 간격 내에서, 각각 상기 절단 값을 위한 상기 절단된 단항 코드 형태의 접두부호와 상기 수평과 수직 컴포넌트들의 지수-골롬 형태의 접미부호의 조합을 동일시하며, 상기 절단 값은 2이고 상기 지수-골롬 코드는 순차 1을 가지는 비디오를 디코딩하기 위한 방법에 있어서:
- [0726] 상기 움직임 벡터 차분값들의 상기 수평과 수직 컴포넌트들을 위하여, 움직임 벡터 차분값들의 수평과 수직 컴포넌트들에 통상적인, 절단된 단항 코드의 위치 당 정확하게 하나의 컨텍스트를 갖는 컨텍스트-적응적 이진 엔트로피 디코딩을 사용하여 데이터 스트림으로부터의 상기 절단된 단항 코드 및, 움직임 벡터 차분값들의 이진화를 획득하기 위하여 일정한 등-확률 바이패스 방식을 사용하여 지수-골롬 코드를 유래하는 단계;
- [0727] 상기 움직임 벡터 차분값들의 상기 수평과 수직 컴포넌트들의 정수 값들을 획득하기 위하여 상기 움직임 벡터 차분값 구문 요소들의 이진화들을 탈이진화하는 단계;
- [0728] 상기 움직임 벡터 차분값들의 상기 수평과 수직 컴포넌트들의 정수 값들을 기초로 하여 상기 비디오를 재구성하는 단계;를 특징으로 하는 비디오를 디코딩하기 위한 방법
- [0730] [실시예 31]
- [0731] 비디오를 데이터 스트림 내로 인코딩하기 위한 인코더에 있어서,
- [0732] 움직임 벡터들을 사용하는 움직임 보상 예측에 의해 상기 비디오를 예측으로 코딩하는 단계 및 상기 움직임 벡터들을 예측하고 상기 예측된 움직임 벡터들의 예측 오차를 표현하기 위하여 움직임 벡터 차분값들의 수평과 수직 컴포넌트들의 정수 값의 설정함으로써 상기 움직임 벡터들을 예측으로 코딩하는 단계;
- [0733] 상기 움직임 벡터 차분값들의 상기 수평과 수직 컴포넌트들의 이진화들을 획득하기 위하여 상기 정수 값들을 이진화하는 단계를 구비하되, 상기 이진화들은 절단 값 아래의 상기 수평과 수직 컴포넌트들의 도메인의 첫 번째 간격 내에서, 각각 상기 수평과 수직 컴포넌트들의 절단된 단항 코드와, 상기 절단 값을 포함하고 상기 절단 값 위의 상기 수평과 수직 컴포넌트들의 상기 도메인의 두 번째 간격 내에서, 각각 상기 절단 값을 위한 상기 절단된 단항 코드 형태의 접두부호와 상기 수평과 수직 컴포넌트들의 지수-골롬 형태의 접미부호의 조합과 동일하며, 상기 절단 값은 2이고 상기 지수-골롬 코드는 순차 1을 가지며;
- [0734] 상기 움직임 벡터 차분값들의 상기 수평과 수직 컴포넌트들을 위하여, 움직임 벡터 차분값들의 수평과 수직 컴포넌트들에 통상적인, 절단된 단항 코드의 위치 당 정확하게 하나의 컨텍스트를 갖는 컨텍스트-적응적 이진 산술 인코딩을 사용하여 상기 절단된 단항 코드를 상기 데이터 스트림 및 일정한 등-확률 바이패스 방식을 사용하여 지수-골롬 코드 내로 인코딩하는 단계;를 포함하는 것을 특징으로 하는 인코더.

- [0736] [실시예 32]
- [0737] 컴퓨터 상에 구동할 때, 제 30실시예 내지 31실시예 중 어느 한 실시예에 따른 방법을 실행하기 위한 프로그램 코드를 갖는 컴퓨터 프로그램.

부호의 설명

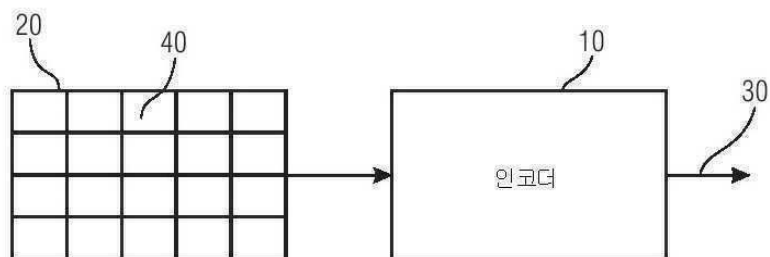
- [0739] 10 : 인코더
- 20 : 샘플 어레이
- 21 : 빈 스트림
- 22 : 디코더
- 24 : 비트스트림
- 30 : 데이터 스트림
- 40 : 블록
- 40b : 마크로블록
- 50 : 디코더
- 60 : 샘플 어레이의 재구성된 버전
- 100 : 변환 스테이지
- 102 : 엔트로피 코더
- 104 : 역 변환 스테이지
- 106 : 예측변수
- 108 : 감산기
- 110 : 가산기
- 112 : 입력
- 114 : 출력
- 120 : 픽처
- 122 : 블록
- 126 : 예측변수
- 128 : 움직임 벡터
- 150 : 엔트로피 디코더
- 152 : 역 변환 스테이지
- 154 : 가산기
- 156 : 예측변수
- 158 : 입력
- 160 : 출력
- 200 : 블록
- 202 : 샘플
- 204 : 변환 블록

206 : 변환 계수
 208 : 블록의 카피
 301 : 구문 요소
 302 : 이진화기
 303 : 빈
 304 : 파라미터 할당기
 305 : 파라미터
 306 : 빈 버퍼 선택기
 307 : 출력 빈
 308 : 빈 버퍼
 309 : 빈
 310 : 인코더
 311 : 코드워드
 312 : 비트스트림
 313 : 구문 요소
 314 : 디심벌라이저
 316 : 파라미터 할당기
 317 : 파라미터
 318 : 빈 버퍼 선택기
 319 : 빈
 320 : 빈 버퍼
 322 : 엔트로피 디코더
 325 : 입력 빈
 326 : 출력 빈
 327 : 구문 요소
 400 : 데이터 패킷
 401 : 데이터 스트림
 402 : 비트스트림 데이터
 404 : 재구성자
 405 : 미디어 데이터
 409 : 엔트로피 디코더
 411 : 헤더
 412 : 세그먼트
 413 : 부분 비트스트림
 500 : 삼입기
 501 : 데이터 스트림

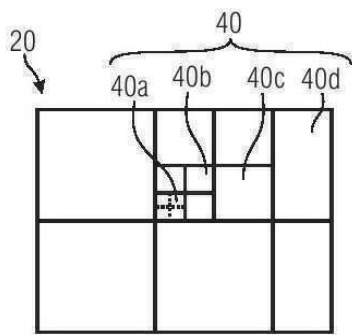
502 : 선택기
 504 : 구성자
 505 : 미디어 데이터
 506 : 구문 요소들의 시퀀스
 507 : 심벌라이저
 508 : 심벌들의 시퀀스
 510 : 인터리버
 600 : 모드 스위치
 601 : 데이터 스트림
 602 : 디심벌라이저
 603 : 시퀀스
 605 : 재구성자
 608 : 엔트로피 디코더
 700 : 삽입기
 701 : 데이터 스트림
 705 : 미디어 데이터
 706 : 요소들의 시퀀스
 707 : 심벌라이저
 800 : 움직임 벡터 차분값
 802_x: 수평 컴포넌트
 802_y : 수직 컴포넌트들
 802_{x,y} : 수평과 수직 컴포넌트
 804 : 도메인 축
 806 : 절단된 단항 코드
 808 : 지수 골룸 코드
 822 : 블록
 826 : 예측변수

도면

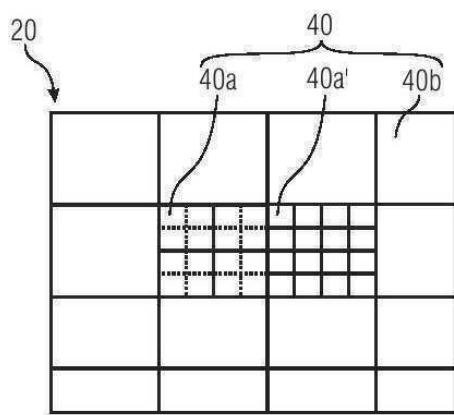
도면1



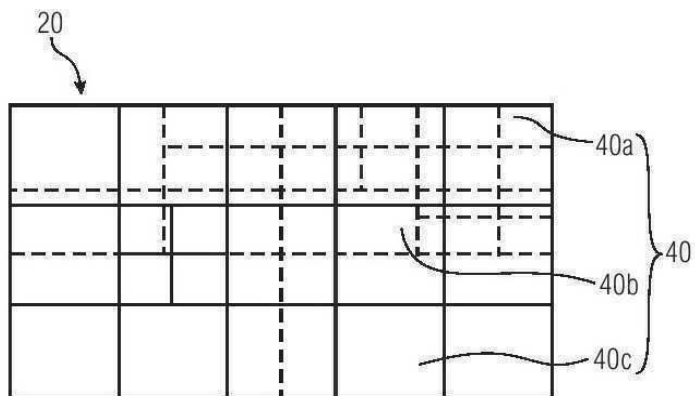
도면2a



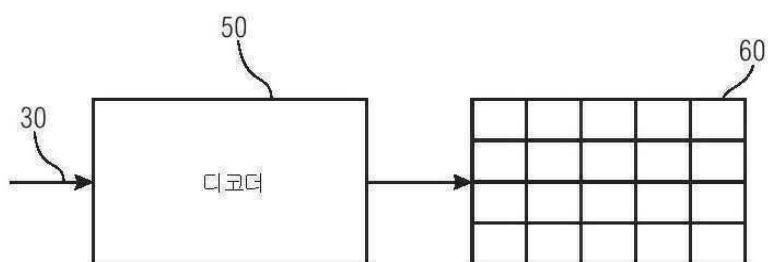
도면2b



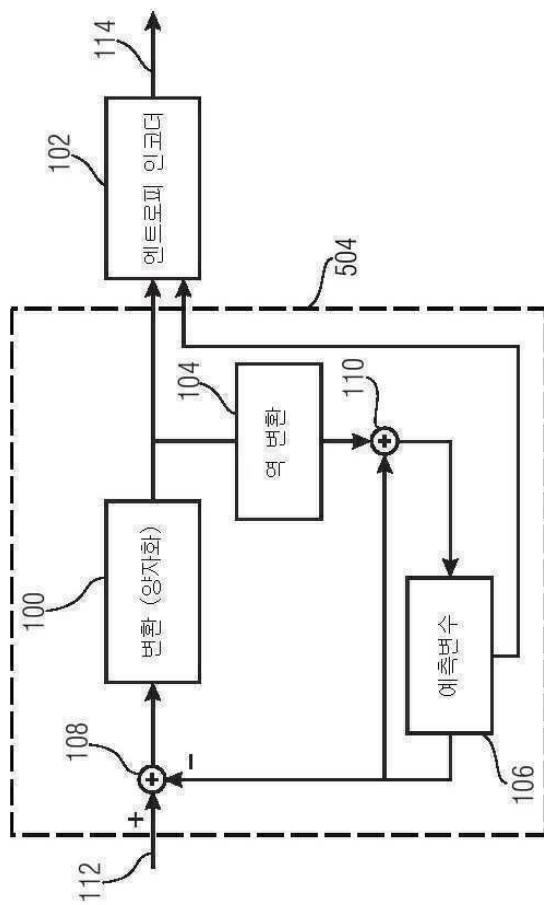
도면2c



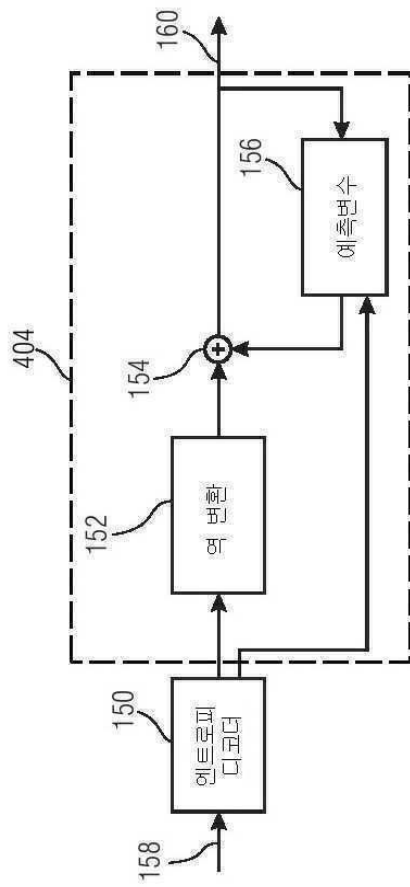
도면3



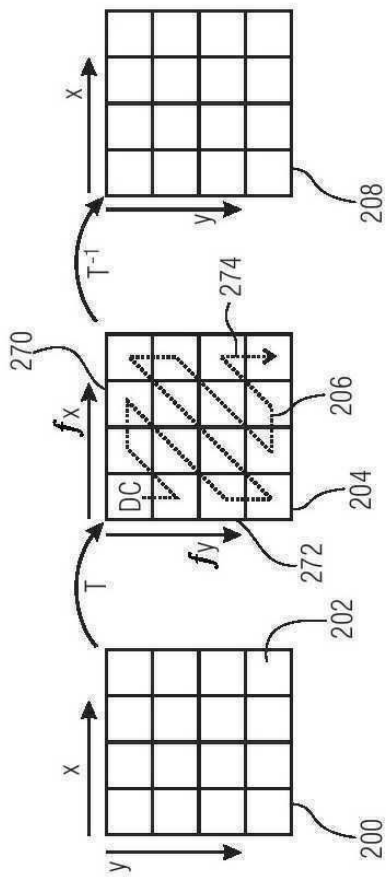
도면4



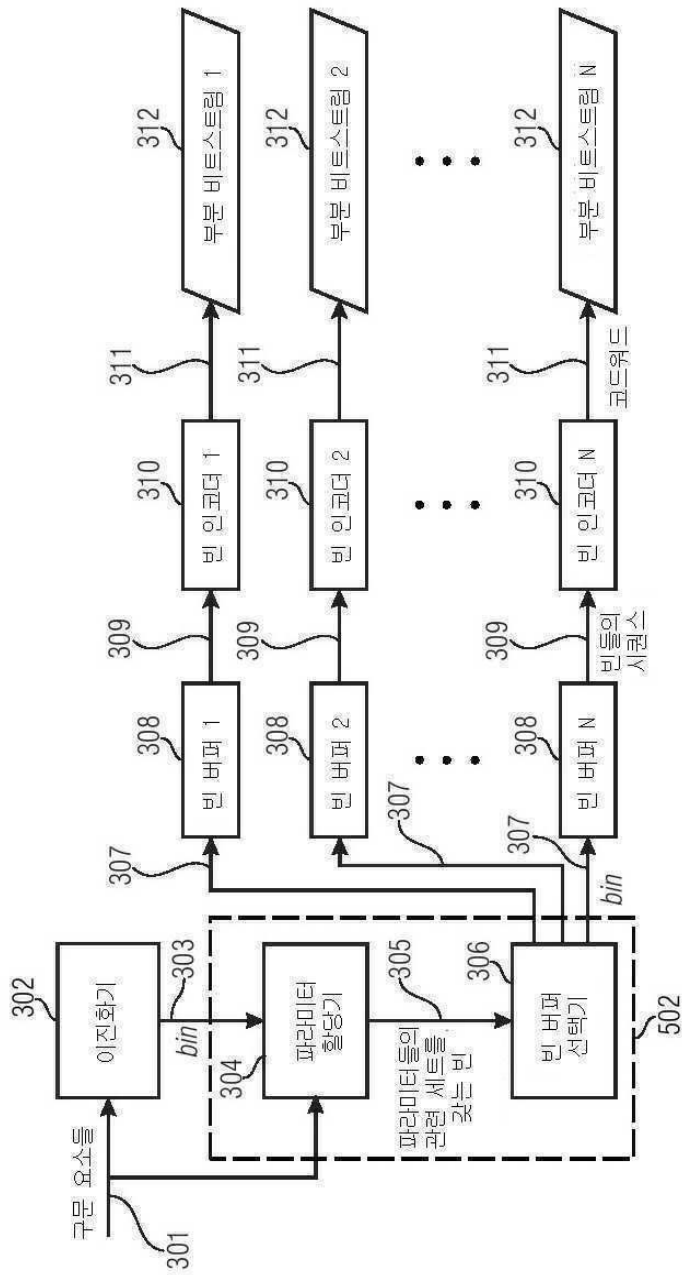
도면5



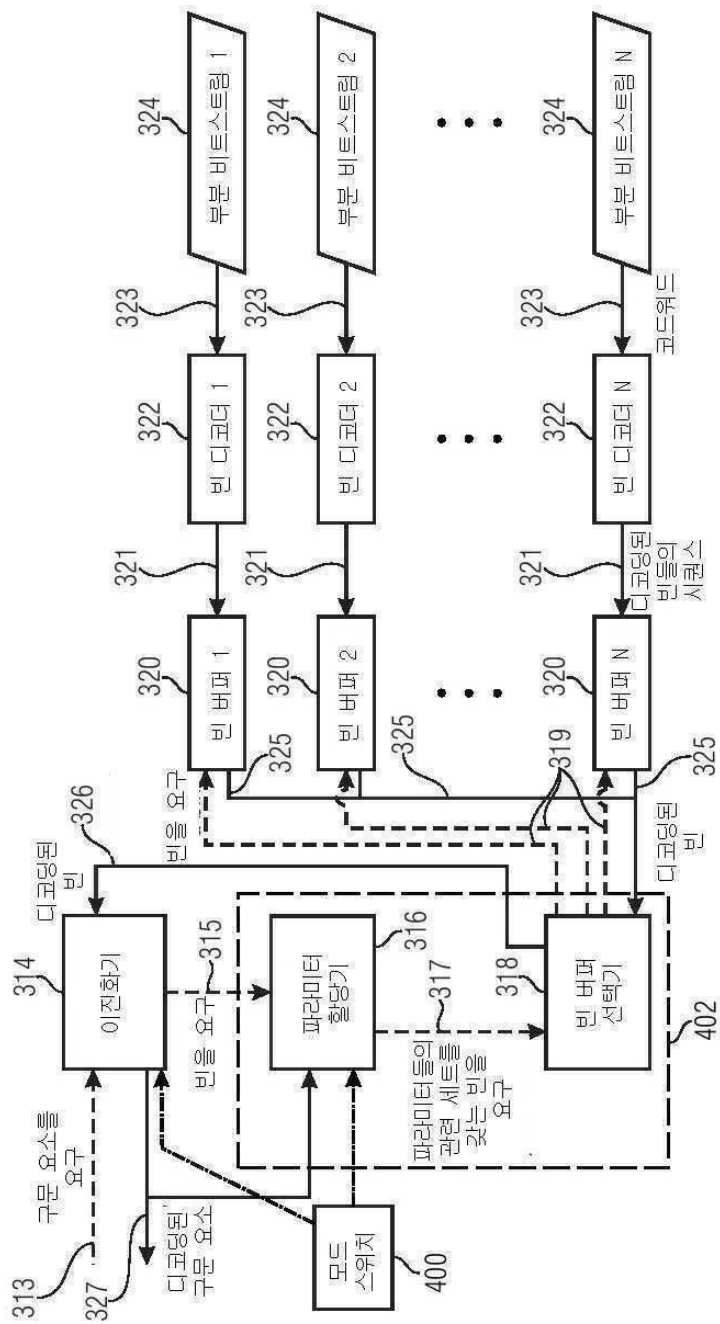
도면6



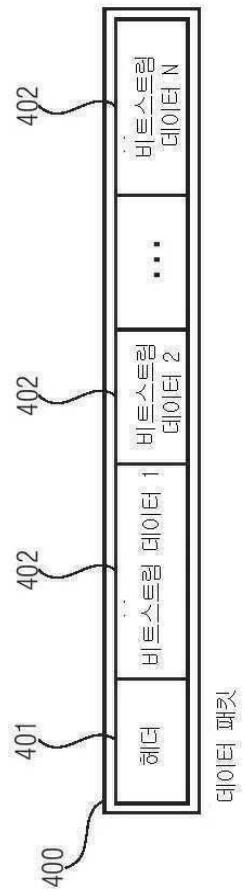
도면7



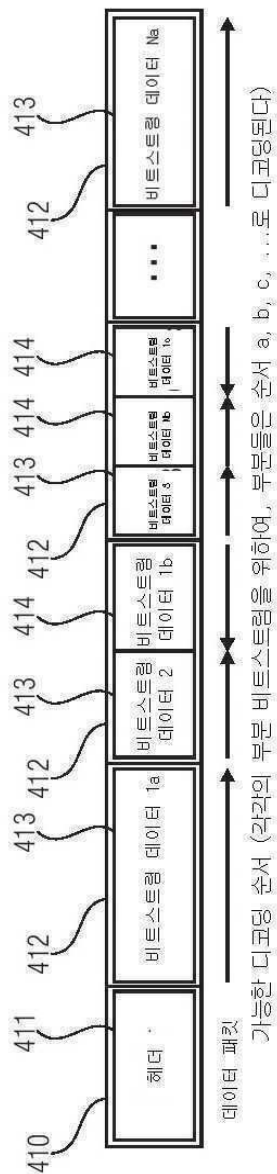
도면8



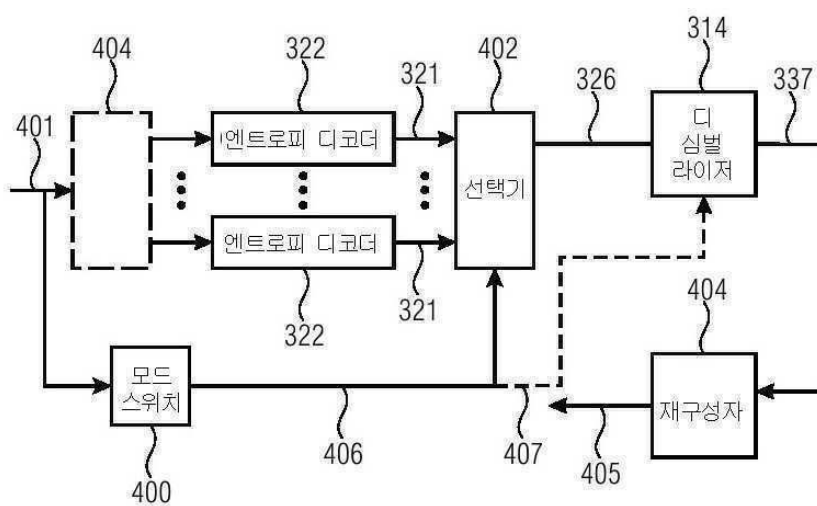
도면9



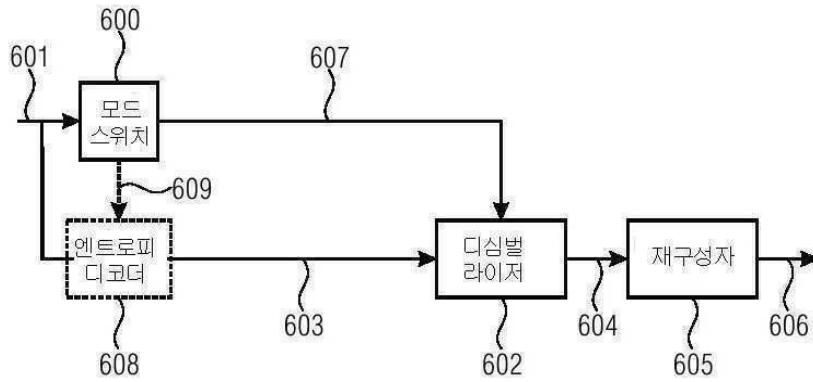
도면 10



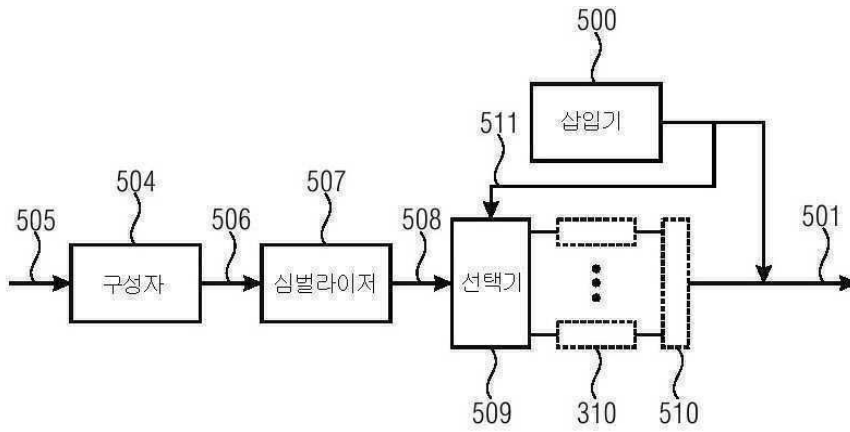
도면11



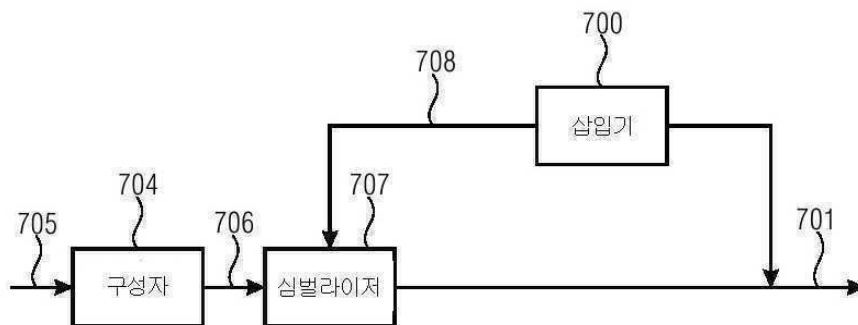
도면12



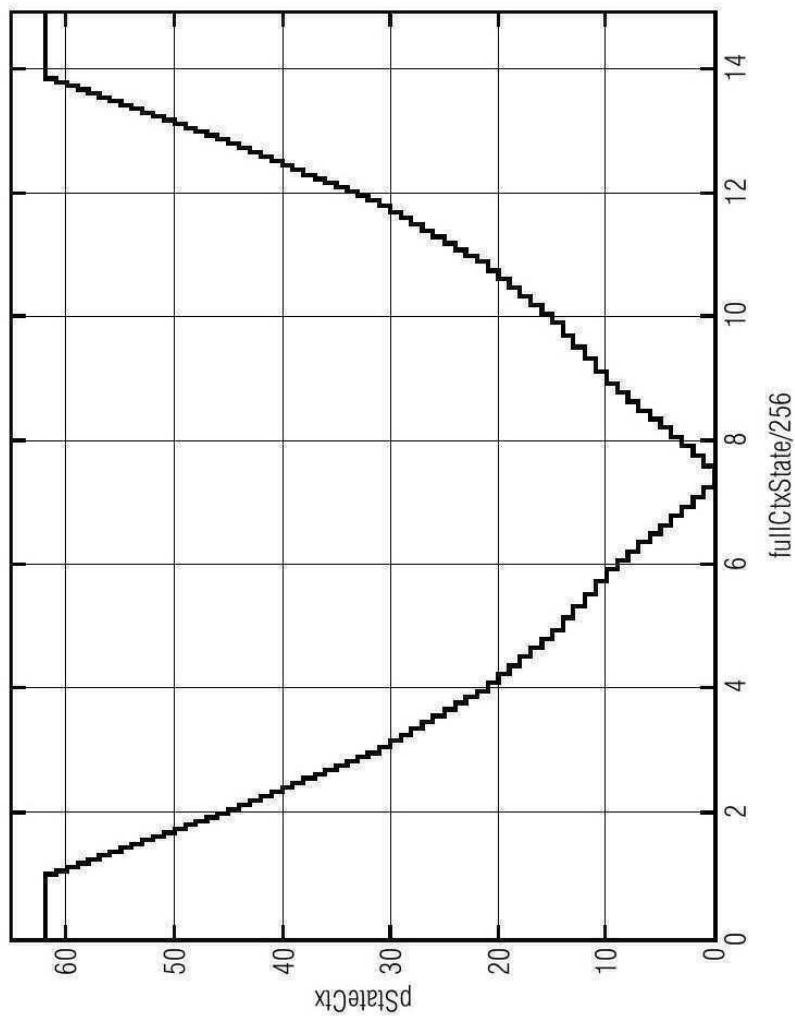
도면13



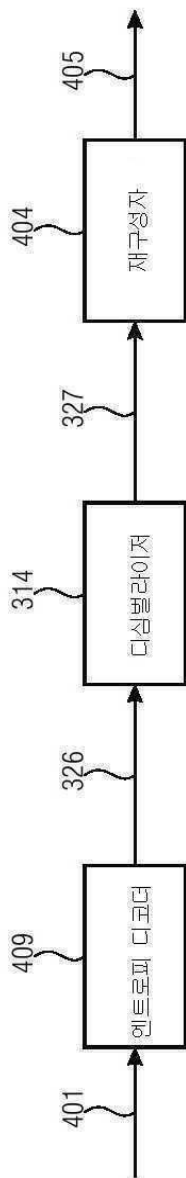
도면14



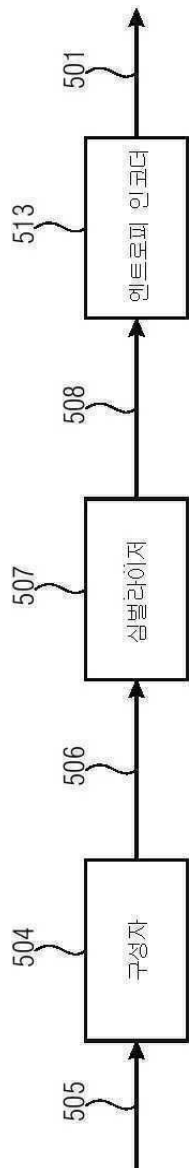
도면15



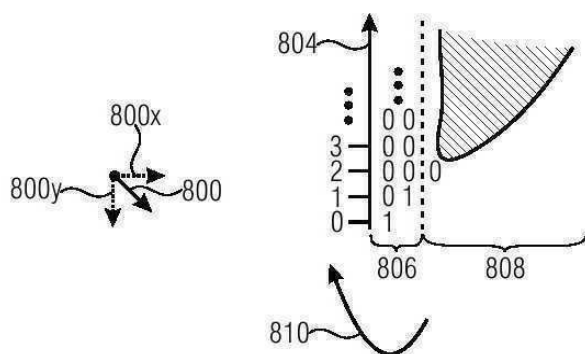
도면16



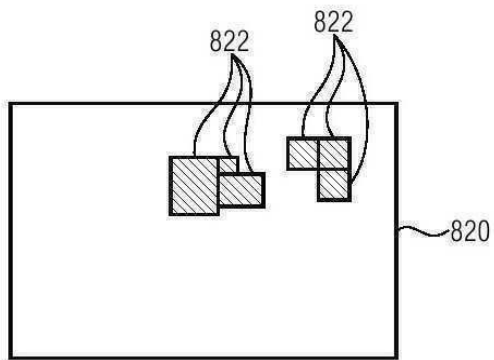
도면17



도면18



도면19



도면20

