(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2019년03월06일
(11) 등록번호 10-1955143
(24) 등록일자 2019년02월27일

(51) 국제특허분류(Int. Cl.)
H04N 19/13 (2014.01) *H04N 19/70* (2014.01)
(52) CPC특허분류
H04N 19/13 (2015.01)
H04N 19/70 (2015.01)
(21) 출원번호 10-2017-7027822(분할)
(22) 출원일자(국제) 2012년01월12일
심사청구일자 2017년10월27일
(85) 번역문제출일자 2017년09월28일
(65) 공개번호 10-2017-0117217
(43) 공개일자 2017년10월20일
(62) 원출원 특허 10-2017-7013696
원출원일자(국제) 2012년01월12일
심사청구일자 2017년05월19일
(86) 국제출원번호 PCT/EP2012/050431
(87) 국제공개번호 WO 2012/095488
국제공개일자 2012년07월19일
(30) 우선권주장
61/432,884 2011년01월14일 미국(US)
(56) 선행기술조사문헌
WO2008129021 A2
M. Winken, et al. Description of video coding technology proposal by Fraunhofer HHI. JCT-VC of ITU-T and ISO/IEC. JCTVC-A116. Apr. 13, 2010, pp.1-44
KR101741296 B1*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
지이 비디오 컴프레션, 엘엘씨
미국 뉴욕 12211 올버니 사우스우즈 블러바드 8
(72) 발명자
마르페, 데틀레브
독일연방공화국, 12161 베를린, 수트베스트코르소 70
웅웅엔, 퉁
독일연방공화국, 13055 베를린, 란트스버거 알레 217
(뒷면에 계속)
(74) 대리인
한양특허법인

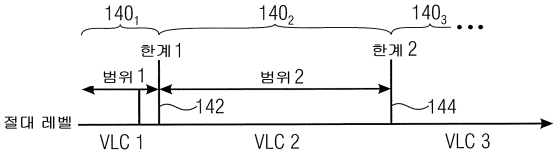
전체 청구항 수 : 총 14 항 심사관 : 이상래

(54) 발명의 명칭 **엔트로피 인코딩 및 디코딩 방식**

(57) 요 약

각각의 신택스 요소의 값 범위를 n개의 분할 시퀀스로 분해하며, 각각의 분할 내에 있는 z의 성분을 개별적으로 VCL 코딩에 의해 적어도 하나로 코딩하며, PIPE나 엔트로피 코딩에 의해 적어도 하나로 코딩하는 것은 중간 코딩 오버헤드에서 압축 효율을 상당히 증가시키는데 사용되며, 이는 사용된 코딩 방식이 신택스 요소 통계에 더 양호하게 적응될 수 있기 때문이다. 따라서, 실시예에 따라, 신택스 요소는 각각의 수(n)의 소스 심벌로 분해되며, 상기 각각의 수(n)의 소스 심벌은, n개의 분할(140_{1-3})의 시퀀스 중 어떤 것에, 각각의 신택스 요소의 값 범위가 세분되며 각각의 신택스 요소의 값(z)이 속하는지에 의존하여, 상기 각각의 수의 소스 심벌(s_i)의 값의 합이 z를 산출하며, n>1인 경우, 모든 i=1...n-1에 대해, s_i의 값이 i번째 분할의 범위에 대응하게 된다.

대 표 도 - 도1b



(72) 발명자

쉬바르츠, 하이코

독일연방공화국, 13187 베를린, 뉘에 숀홀처 스트라쎄 12

비간드, 토마스

독일연방공화국, 14195 베를린, 오토-아펠-스트라쎄 52

명 세 서

청구범위

청구항 1

신택스 요소의 적어도 서브그룹의 신택스 요소를 각각의 수(n)의 소스 심벌(s_i)(i=1,...,n)로 개별적으로 분해함으로써, N개의 분할(140_{1-3})의 시퀀스로 세분되는 값의 범위를 갖는, 상기 신택스 요소의 시퀀스(138)를 소스 심벌의 시퀀스(106)로 전환하도록 구성되는 분해기(136)로서, 상기 각각의 수(n)의 소스 심벌은, 상기 N개의 분할(140_{1-3})의 시퀀스 중 어떤 분할에 상기 각각의 신택스 요소의 값(z)이 속하는지에 의존하여, 상기 각각의 수(n)의 소스 심벌(s_i)의 값의 합이 z를 산출하며, n>1인 경우, 모든 i=1...n-1에 대해, s_i의 값이 i번째 분할의 범위에 대응하게 되는, 분해기(136);

상기 소스 심벌의 시퀀스(106)를 소스 심벌의 제1 서브시퀀스(108)와 소스 심벌의 제2 서브시퀀스(110)로 세분하여, x가 {1,...,N}의 제1 서브세트의 일원인 소스 심벌 s_x는 모두 상기 제1 서브시퀀스(108) 내에 포함되고, y가 상기 제1 서브세트와 공통되지 않은 {1,...,N}의 제2 서브세트의 일원인 소스 심벌 s_y가 모두 상기 제2 서브시퀀스(110) 내에 포함되게 하도록 구성되는 서브디바이더(100);

상기 제1 서브시퀀스(108)의 소스 심벌을 심벌-방식으로(symbol-wisely) 인코딩하도록 구성되는 VLC 인코더(102); 및

상기 소스 심벌의 제2 서브시퀀스(110)를 인코딩하도록 구성되는 산술 인코더(104)를 포함하고,

상기 신택스 요소의 상기 서브그룹의 값(z)들은 변환 계수 레벨들 또는 움직임 벡터들의 절대값들이고,

상기 분해기는, 이전에 인코딩된 소스 심벌에 따라 상기 분할 사이의 하나 이상의 한계를 조정하도록 구성되는, 엔트로피 인코딩 장치.

청구항 2

청구항 1에 있어서, 상기 신택스 요소는 비디오를 스케일링 가능한 방식으로 인코딩하는, 엔트로피 인코딩 장치.

청구항 3

제1 비트스트림(206)의 코드워드로부터 소스 심벌의 제1 서브시퀀스(204)의 소스 심벌을 코드워드-방식으로 재구성하도록 구성되는 VLC 디코더(200);

소스 심벌의 제2 서브시퀀스(208)를 재구성하도록 구성되는 산술 디코더(202); 및

신택스 요소의 적어도 서브그룹의 신택스 요소에 대해, 신택스 요소의 값(z)을 얻기 위해, 소스 심벌(s_i)의 값이 i번째 분할의 범위에 대응하는 한, 각각의 수의 소스 심벌(s_i)의 값을 1부터 n까지 합함으로써, N개의 분할(140_{1-3})의 시퀀스 중 어떤 분할에, 각각의 신택스 요소의 값 범위가 세분되며 각각의 신택스 요소의 값(z)이 속하는지에 의존하여, 상기 각각의 수(n)의 소스 심벌(s_i)(i=1,...,n)을 결정하여, 상기 각각의 수(n)의 소스 심벌로부터 각 신택스 요소를 개별적으로 구성함으로써, 상기 소스 심벌의 제1 서브시퀀스(204)와 상기 소스 심벌의 제2 서브시퀀스(208)로부터 상기 N개의 분할(140_{1-3})의 시퀀스로 세분되는 값의 범위를 갖는, 상기 신택스 요소의 시퀀스(226)를 구성하도록 구성되는 구성기(224)를 포함하며,

상기 구성기(224)는, 상기 제1 서브시퀀스(204)로부터 x가 {1,...,N}의 제1 서브세트의 일원인 모든 소스 심벌 s_x와 상기 제2 서브시퀀스(208)로부터 y가 상기 제1 서브세트와 공통되지 않은 {1,...,N}의 제2 서브세트의 일원인 모든 소스 심벌 s_y를 회수(retrieve)하도록 구성되고,

상기 신택스 요소의 상기 서브그룹의 값(z)은 절대값들이고,

상기 구성기는 이전에 재구성된 소스 심벌에 따라 상기 분할 사이의 하나 이상의 한계를 조정하도록 구성되는, 엔트로피 디코딩 장치.

청구항 4

청구항 3에 있어서, 상기 제2 서브세트는 {1}이며, 상기 N개의 분할 시퀀스는, p번째 분할이 모든 p, q∈{1..N}(p>q)에 대해 q번째 분할보다 더 큰 값의 값의 범위를 커버하도록 배치되는, 엔트로피 디코딩 장치.

청구항 5

청구항 4에 있어서, N=3인, 엔트로피 디코딩 장치.

청구항 6

삭제

청구항 7

청구항 3 내지 청구항 5 중 어느 한 항에 있어서, 소스 심벌의 시퀀스(218)를 얻기 위해 상기 소스 심벌의 제1 서브시퀀스(204)와 상기 소스 심벌의 제2 서브시퀀스를 재결합하도록 구성되는 재결합기(220)를 더 포함하는, 엔트로피 디코딩 장치.

청구항 8

청구항 3에 있어서, 상기 신택스 요소는 비디오를 스케일링 가능한 방식으로 디코딩하는, 엔트로피 디코딩 장치.

청구항 9

신택스 요소의 적어도 서브그룹의 신택스 요소를 각각의 수(n)의 소스 심벌(s_i)(i=1,...,n)로 개별적으로 분해함으로써, N개의 분할(140_{1-3})의 시퀀스로 세분되는 값의 범위를 갖는, 상기 신택스 요소의 시퀀스(138)를 소스 심벌의 시퀀스(106)로 전환하는 단계로서, 상기 각각의 수(n)의 소스 심벌은, 상기 N개의 분할(140_{1-3})의 시퀀스 중 어떤 분할에 상기 각각의 신택스 요소의 값(z)이 속하는지에 의존하여, 상기 각각의 수의 소스 심벌(s_i)의 값의 합이 z를 산출하며, n>1인 경우, 모든 i=1...n-1에 대해, s_i의 값이 i번째 분할의 범위에 대응하게 되는, 단계;

상기 소스 심벌의 시퀀스(106)를 소스 심벌의 제1 서브시퀀스(108)와 소스 심벌의 제2 서브시퀀스(110)로 세분하여, x가 {1,...,N}의 제1 서브세트의 일원인 소스 심벌 s_x는 모두 상기 제1 서브시퀀스(108) 내에 포함되고, y가 상기 제1 서브세트와 공통되지 않은 {1,...,N}의 제2 서브세트의 일원인 소스 심벌 s_y가 모두 상기 제2 서브시퀀스(110) 내에 포함되는 단계;

VLC 인코딩에 의해, 상기 제1 서브시퀀스(108)의 소스 심벌을 심벌-방식으로 인코딩하는 단계; 및

산술 인코딩에 의해, 상기 소스 심벌의 제2 서브시퀀스(110)를 인코딩하는 단계를 포함하며,

상기 신택스 요소의 상기 서브그룹의 값(z)들은 변환 계수 레벨들 또는 움직임 벡터들의 절대값들이고,

개별적으로 분해함으로써 전환하는 상기 단계는 이전에 인코딩된 소스 심벌에 따라 상기 분할 사이의 하나 이상의 한계를 조정하는 단계를 포함하는, 엔트로피 인코딩 방법.

청구항 10

청구항 9에 있어서, 상기 신택스 요소는 비디오를 스케일링 가능한 방식으로 인코딩하는, 엔트로피 인코딩 방법.

청구항 11

VLC 디코딩에 의해, 제1 비트스트림(206)의 코드워드로부터 소스 심벌의 제1 서브시퀀스(204)의 소스 심벌을 코

드워드-방식으로 재구성하는 단계;

산술 디코딩에 의해, 소스 심벌의 제2 서브시퀀스(208)를 재구성하는 단계; 및

신택스 요소의 적어도 서브그룹의 신택스 요소에 대해, 상기 신택스 요소의 값(z)을 얻기 위해, 소스 심벌(s_i)의 값이 i번째 분할의 범위에 대응하는 한, 각각의 수의 소스 심벌(s_i)의 값을 1부터 n까지 합함으로써, N개의 분할(140_{1-3})의 시퀀스 중 어떤 분할에, 각각의 신택스 요소의 값 범위가 세분되며 각각의 신택스 요소의 값(z)이 속하는지에 의존하여, 상기 각각의 수(n)의 소스 심벌(s_i)(i=1,...,n)을 결정하여, 상기 각각의 수(n)의 소스 심벌로부터 각 신택스 요소를 개별적으로 구성함으로써, 상기 소스 심벌의 제1 서브시퀀스(204)와 상기 소스 심벌의 제2 서브시퀀스(208)로부터 N개의 분할(140_{1-3})의 시퀀스로 세분된 값의 범위를 가지는 신택스 요소의 시퀀스(226)를 구성하는 단계를 포함하며,

상기 구성하는 단계는, 상기 제1 서브시퀀스(204)로부터 x가 {1,...,N}의 제1 서브세트의 일원인 모든 소스 심벌 s_x와 상기 제2 서브시퀀스(208)로부터 y가 상기 제1 서브세트와 공통되지 않은 {1,...,N}의 제2 서브세트의 일원인 모든 소스 심벌 s_y를 회수하는 단계를 포함하며,

상기 신택스 요소의 상기 서브그룹의 값(z)은 절대값들이고,

상기 구성하는 단계는, 이전에 재구성된 소스 심벌에 따라 하나 이상의 상기 분할 사이의 경계를 조정하는 단계를 포함하는, 엔트로피 디코딩 방법.

청구항 12

청구항 11에 있어서, 상기 신택스 요소는 비디오를 스케일링 가능한 방식으로 디코딩하는, 엔트로피 디코딩 방법.

청구항 13

컴퓨터 상에서 실행될 때, 청구항 9 또는 청구항 11에 기재된 방법을 실행하는 프로그램 코드를 갖는 컴퓨터 프로그램이 저장된, 컴퓨터가 읽을 수 있는 저장 매체.

청구항 14

디지털 저장 매체로서, 청구항 9에 기재된 엔트로피 인코딩 방법에 의해 인코딩된 데이터 스트림이 저장된, 디지털 저장 매체.

청구항 15

청구항 14에 있어서, 상기 데이터 스트림은 비디오를 스케일링 가능한 방식으로 인코딩하는, 디지털 저장 매체.

청구항 16

삭제

청구항 17

삭제

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

청구항 21

삭제

청구항 22

삭제

청구항 23

삭제

청구항 24

삭제

청구항 25

삭제

청구항 26

삭제

청구항 27

삭제

청구항 28

삭제

청구항 29

삭제

청구항 30

삭제

청구항 31

삭제

청구항 32

삭제

청구항 33

삭제

청구항 34

삭제

청구항 35

삭제

청구항 36

삭제

청구항 37

삭제

청구항 38

삭제

청구항 39

삭제

청구항 40

삭제

청구항 41

삭제

청구항 42

삭제

청구항 43

삭제

청구항 44

삭제

청구항 45

삭제

청구항 46

삭제

청구항 47

삭제

청구항 48

삭제

청구항 49

삭제

청구항 50

삭제

청구항 51

삭제

발명의 설명

기 술 분 야

[0001] 본 발명은 엔트로피 인코딩 및 디코딩에 관한 것이며, 예컨대 비디오 및 오디오 압축과 같은 응용에 이용될 수 있다.

배 경 기 술

[0002] 엔트로피 코딩은, 일반적으로, 무손실 데이터 압축의 가장 일반적인 형태로서 간주될 수 있다. 무손실 압축의 목적은 원 데이터 표현에 필요한 것보다 더 소수의 비트로 그러나 정보 손실 없이도 이산 데이터를 표현하는 것이다. 이산 데이터는 텍스트, 그래픽스, 영상, 비디오, 오디오, 음성, 팩스, 의료 데이터, 기상 데이터, 금융 데이터, 또는 임의의 다른 형태의 디지털 데이터의 형태로 제공될 수 있다.

[0003] 엔트로피 코딩에서, 기저의 이산 데이터 소스의 특정한 높은 레벨의 특징은 종종 무시된다. 결국, 임의의 데이터 소스는 소정의 m-진 알파벳의 값을 가지며 대응하는 (이산) 확률 분포($\{p_1,\ldots,p_m\}$)의 특징을 갖는 소스 심벌의 시퀀스로서 주어지는 것으로 간주된다. 이들 추상적인 세팅에서, 심벌당 비트 단위의 예상 코드워드 길이 면에서 엔트로피 코딩 방법의 하한은 다음의 엔트로피로 주어진다.

[0004] $$H = -\sum_{i=1}^{m} p_i \log_2 p_i \qquad \text{(A1)}$$

[0005] 허프만 코드(Huffman code) 및 산술 코드가 (특정한 면에서의) 엔트로피 한계를 근사화할 수 있는 실제 코드의 잘 알려져 있는 예이다. 고정된 확률 분포의 경우, 허프만 코드는 상대적으로 구성하기 쉽다, 허프만 코드의 가장 매력적인 속성은, 이것의 구현이 VLC(Variable-Length Code) 표의 사용에 의해 효율적으로 실현될 수 있다는 점이다. 그러나 시변 소스 통계, 즉 변화하는 심벌 확률을 처리할 때, 허프만 코드 및 그 대응하는 VLC 표의 적응은 구현 비용 면에서뿐만 아니라 알고리즘 복잡도 면에서 꽤 힘이 든다. 또한, p_k>0.5인 지배적인 알파벳 값을 갖는 경우에, (런 렝쓰 코딩(run length coding)과 같은 임의의 알파벳 확장을 사용하지 않고도) 대응하는 허프만 코드의 중복도는 상당히 클 수 있다. 허프만 코드의 다른 단점은, 고차 확률 모델링을 다룰 경우에 복수 세트의 VLC 표가 필요할 수 있다는 점에 의해 주어진다. 다른 한편으로, 산술 코딩은, VLC 보다 상당히 더 복잡하면서도, 더 왜곡된 확률 분포의 경우에서뿐만 아니라 적응적이며 더 고차인 확률 모델링을 다룰 때 더 일관되고 충분히 처리하는 장점을 제공한다. 실제로, 이러한 특징은 기본적으로, 산술 코딩이 적어도 개념적으로는 확률 추정의 임의의 소정 값을 다소 직접적인 방식으로 결과 코드워드의 부분에 매핑하는 메커니즘을 제공한다는 사실로부터 유래한다. 그러한 인터페이스가 제공됨으로써, 산술 코딩은 한편으론 확률 모델링 및 확률 추정의 작업과 다른 한편으론 실제 엔트로피 코딩, 즉 심벌의 코드워드로의 매핑 사이에 명백한 분리를 허용한다.

[0006] 산술 코딩 및 VLC 코딩에 대한 대안으로 PIPE 코딩이 있다. 더 정밀하게 말해, PIPE 코딩에서, 단위 간격은 작은 세트의 공통되지 않은(disjoint) 확률 간격으로 분할되어 랜덤 심벌 변수의 확률 추정을 따라 코딩 처리를 파이프라이닝한다. 이 분할에 따라, 임의의 알파벳 크기를 갖는 이산 소스 심벌의 입력 시퀀스는 알파벳 심벌의 시퀀스에 매핑될 수 있으며, 알파벳 심벌 각각은 하나의 특정한 확률 간격에 할당되며, 이 간격은 이제 특별히 전용인 엔트로피 인코딩 처리에 의해 인코딩된다. 각각의 간격이 고정된 확률로 표현됨에 따라, 확률 간격 분할 엔트로피(PIPE: Probability Interval Partitioning Entropy) 코딩 처리가 간단한 변수 간 길이 코드의 설계 및 적용을 기반으로 할 수 있다. 확률 모델링은 고정되거나 적응적일 수 있다. 그러나, PIPE 코딩은 산술 코딩보다 상당히 덜 복잡한 반면, VLC 코딩보다 여전히 더 복잡하다.

[0007] 그러므로 산술 코딩 및 VLC 코딩 모두의 장점을 이미 결합한 PIPE 코딩과도 비교하더라도, 한편으론 코딩 복잡도와 다른 한편으론 압축 효율 사이의 더 양호한 절충을 달성하게 하는 엔트로피 코딩 방식을 갖는 것이 유리할 것이다.

[0008] 또한, 일반적으로, 중간 코딩 복잡도에서 더 양호한 복잡도 효율 자체를 달성할 수 있게 하는 엔트로피 코딩 방식을 직접 갖는 것이 유리할 것이다.

발명의 내용

해결하려는 과제

[0009] 따라서, 본 발명의 목적은 전술한 수요를 충족하는, 즉 한편으론 코딩 복잡도와 다른 한편으론 압축 효율 사이의 더 양호한 절충을 달성할 수 있게 하는 엔트로피 코딩 컨셉을 제공하는 것이다.

과제의 해결 수단

[0010] 이러한 목적은 청구항 1 또는 청구항 23에 기재된 장치, 청구항 49 또는 청구항 50에 기재된 방법, 및 청구항 51에 기재된 컴퓨터 프로그램에 의해 달성된다.

[0011] 본 발명은, 각각의 분할 내에 있는 신택스 요소 값(z)의 구성요소를 적어도 하나는 VLC 코딩에 의해, 적어도 하나는 PIPE 또는 산술 코딩 또는 임의의 다른 엔트로피 코딩 방법에 의해 코딩하여, 각각의 신택스 요소의 값 범위를 n개의 분할 시퀀스로 분해하면, 사용된 코딩 방식이 신택스 요소 통계에 더 양호하게 적응될 수 있다는 점에서 중간 코딩 오버헤드에서 압축 효율을 상당히 증가시킬 수 있다는 아이디어를 기반으로 한다. 따라서, 본 발명의 실시예에 따라, 신택스 요소는 각각의 개수(n)의 소스 심벌(s_i)(i=1....n)로 분해되며, 각각의 개수(n)의 소스 심벌은, n개의 분할(140_{1-3}) 중 어떤 시퀀스에, 각각의 신택스 요소의 값 범위가 세분되며 각각의 신택스 요소의 값(z)이 속하는지에 의존하여, 각각의 개수의 소스 심벌(s_i)의 값의 합이 z를 산출하며, n>1인 경우, 모든 i=1...n-1에 대해, s_i의 값이 i번째 분할의 범위에 대응하게 된다.

[0012] 또한, VLC 코딩 외에 PIPE 코딩을 사용하는 본 발명의 일부 실시예는, 알파벳 심벌이 그 확률 분포 추정에 따라 복수의 특별한 엔트로피 인/디코더 중에서 분포되게 하는 PIPE 코딩의 분류 특성 외에, 인코딩될 소스 심벌을 VLC 코딩을 겪게 되는 제1 서브스트림과, -PIPE 코더의 심벌 알파벳, 즉 알파벳 심벌의 시퀀스로서 표현되는-PIPE 코딩을 겪게 되는 제2 서브스트림으로 세분되게 하는 추가 분류 스테이지가 제공된다면, 한편으론 코딩 복잡도와 다른 한편으론 압축 효율 사이의 더 양호한 절충을 달성할 수 있다는 점을 이용한다. 이 조치에 의해, 본 출원의 명세서의 도입부에서 앞서 개괄적으로 기재한 결합이 없이 적절한 심벌 확률 분포, 즉 VLC 코딩에 의해 효율적으로 코딩되기에 적절한 확률 분포를 갖는 소스 심벌을 더 높은 값의 분할과 같은 VLC 코딩된 심벌로서 분류할 수 있는 반면, 더 낮은 값의 분할과 같은 다른 소스 심벌은 PIPE 코딩된 심벌로서 취급되고 PIPE 코딩을 겪게 될 수 있으며, 그 코딩 복잡도는 VLC 코딩보다 높지만, 더 양호한 압축 효율을 갖는다.

[0013] 실시예에 따라, 신택스 요소를 각각의 정수 개의 소스 심벌로 분해하고, 그 심벌 중 하나는 VLC 코딩된 심벌로서 취급하고 다른 하나는 PIPE 코딩된 심벌로서 취급할 수 있다.

[0014] 본 발명의 바람직한 양상은 포함된 독립 청구범위의 요지이다.

[0015] 본 발명의 바람직한 실시예는 도면에 관해 후술될 것이다.

도면의 간단한 설명

[0016] 도 1a는 실시예에 따른 엔트로피 인코딩 장치의 블록도이다.

도 1b는 실시예에 따른 신택스 요소의 소스 심벌로의 가능한 분해를 예시하는 개략도이다.

도 1c는 실시예에 따른 신택스 요소를 소스 심벌로 분해할 때의 도 1a의 분해기(decomposer)의 가능한 동작 모드를 예시한 흐름도이다.

도 2a는 실시예에 따른 엔트로피 디코딩 장치의 블록도이다.

도 2b는 실시예에 따른 신택스 요소를 소스 심벌로부터 구성할 때의 도 2a의 구성기의 가능한 동작 모드를 예시한 흐름도이다.

도 3은 도 1에 사용할 수 있는 실시예에 따른 PIPE 인코더의 블록도이다.

도 4는 도 3의 PIPE 인코더에 의해 생성된 비트스트림을 도 2에 사용된 실시예에 따라 디코딩하는데 적절한 PIPE 디코더의 블록도이다.

도 5는 실시예에 따른 다중화된 부분 비트스트림을 갖는 데이터 패킷을 예시하는 개략도이다.

도 6은 추가 실시예에 따른 고정 길이 세그먼트를 사용한 대안적인 세그멘테이션(segmentation)을 갖는 데이터 패킷을 예시한 개략도이다.

도 7은 부분 비트스트림 인터리빙을 사용한 실시예에 따른 PIPE 인코더의 블록도이다.

도 8은 실시예에 따른 도 7의 인코더 측에서의 코드워드 버퍼의 상태에 대한 예를 개략적으로 예시한다.

도 9는 부분 비트스트림 인터리빙을 사용한 실시예에 따른 PIPE 디코더의 블록도이다.

도 10은 단일 코드워드 세트를 사용한 코드워드 인터리빙을 사용한 실시예에 따른 PIPE 디코더의 블록도이다.

도 11은 고정 길이 비트 시퀀스의 인터리빙을 사용한 실시예에 따른 PIPE 인코더의 블록도이다.

도 12는 실시예에 따른 도 11의 인코더 측에서의 글로벌 비트 버퍼의 상태에 대한 예를 개략적으로 예시한다.

도 13은 고정 길이 비트 시퀀스의 인터리빙을 사용한 실시예에 따른 PIPE 디코더의 블록도이다.

도 14는 균일한 확률 분포를 (0, 0.5]에서 갖는 K=4로의 최적의 확률 간격 이산화를 예시한 그래프이다.

도 15는 허프만 알고리즘에 의해 얻은 관련 가변 길이 코드 및 p=0.38의 LPB 확률에 대한 이진수 이벤트 트리를 예시한 개략도이다.

도 16은 최대 수의 표 엔트리(L_m)가 주어진 경우의 최적의 코드(C)에 대한 상대적인 비트율 증가($\rho(p, C)$)를 모을 수 있는 그래프이다.

도 17은 K=12개의 간격으로의 이론적으로 최적인 확률 간격 분할에 대한 율 증가와 L_m=65의 최대 개수의 표 엔트리를 갖는 V2V 코드로의 실제 설계를 예시한 그래프이다.

도 18은 삼진(ternary) 선택 트리의 완전 이진 선택 트리로의 전환의 예를 예시한 개략도이다.

도 19는 실시예에 따른 인코더(왼쪽) 및 디코더(오른쪽)를 포함하는 시스템의 블록도이다.

도 20은 실시예에 따른 엔트로피 인코딩 장치의 블록도이다.

도 21은 실시예에 따른 엔트로피 디코딩 장치의 블록도이다.

도 22는 다른 실시예에 따른 엔트로피 인코딩 장치의 블록도이다.

도 23은 실시예에 따른 도 22의 인코더 측에서의 글로벌 비트 버퍼의 상태에 대한 예를 예시하는 개략도이다.

도 24는 본 출원의 다른 실시예에 따른 엔트로피 디코딩 장치의 블록도이다.

발명을 실시하기 위한 구체적인 내용

[0017] 본 출원의 여러 실시예를 도면에 관해 다음의 설명에서 기재하기 전에, 동일한 참조 번호는 도면에 걸쳐서 사용되어 이들 도면에서 동일하거나 등가의 요소를 표시하며, 이전 도면 중 어느 하나에서 제공된 이들 요소에 대한 설명은, 이전 설명이 현재의 도면의 설명과 충돌하지 않는 한, 그 다음의 도면 중 어느 것에도 적용되어야 한다.

[0018] 도 1a는 본 출원의 실시예에 따른 엔트로피 인코딩 장치이다. 이 장치는 서브디바이드(100), VLC 인코더(102) 및 PIPE 인코더(104)를 포함한다.

[0019] 서브디바이더(100)는 소스 심벌의 시퀀스(106)를 소스 심벌의 제1 서브시퀀스(108)와 소스 심벌의 제2 서브시퀀스(110)로 세분하도록 구성된다. VLC 인코더(102)는 서브디바이더(100)의 제1 출력에 연결된 입력을 가지며, 제1 서브시퀀스(108)의 소스 심벌을 제1 비트스트림(112)을 형성하는 코드워드로 심벌-방식으로 전환하도록 구성된다. VLC 인코더(102)는 룩업표를 포함할 수 있으며, 개별적으로 소스 심벌을 인덱스로서 사용할 수 있어서, 소스 심벌마다 각각의 코드워드를 룩업 표에서 룩업할 수 있다. VLC 인코더는 후자의 코드워드를 출력하며, 서브시퀀스(110)에서 다음의 소스 심벌로 진행하여, 각 코드워드가 서브스퀀스(110) 내의 소스 심벌 중 정확히 하나와 관련되는 코드워드 시퀀스를 출력한다. 코드워드는 상이한 길이를 가질 수 있으며, 코드워드가 다른 코드워드 중 어느 것과 프레픽스를 형성하지 않도록 한정될 수 있다. 또한, 룩업표는 정적(static)일 수 있다.

[0020] PIPE 인코더(104)는 서브디바이더(100)이 제2 출력에 더 연결된 입력을 가지며, 알파벳 심벌 시퀀스의 형태로 표현된 소스 심벌의 제2 서브시퀀스(110)를 인코딩하도록 구성되며, 알파벳 심벌 시퀀스의 이전 알파벳 심벌 내에 포함된 정보를 기반으로, 각각의 알파벳 심벌이 가질 수 있는 가능 값 사이의 확률 분포의 추정에 대한 측정치를 알파벳 심벌 시퀀스의 각 알파벳 심벌에 할당하도록 구성된 할당기(114), 각각의 엔트로피 인코더에 전송된 알파벳 심벌을 각각의 제2 비트스트림(118)으로 전환하도록 각각 구성되는 복수의 엔트로피 인코더(116), 및 제2 서브시퀀스(110)의 각 알파벳 심벌을 복수의 엔트로피 인코더(116) 중 선택된 하나에 전송하도록 구성된 선택기(120)를 포함하며, 그러한 선택은, 각각의 알파벳 심벌에 할당된 확률 분포의 추정에 대한 전술한 측정치에 의존한다. 소스 심벌과 알파벳 심벌 사이의 관련성은, 각 알파벳 심벌이 서브시퀀스(110)의 정확히 하나의 소스 심벌과 고유하게 관련되어 서로 바로 뒤따를 수 있는 알파벳 심벌 시퀀스의 가능한 다른 알파벳 심벌과 함께 이

하나의 소스 심벌을 나타내도록 이루어질 수 있다.

[0021] 더 상세하게 후술될 바와 같이, 소스 심벌의 시퀀스(106)는 파싱 가능한 비트스트림의 신택스 요소의 시퀀스일 수 있다. 파싱 가능한 비트스트림은, 예컨대, 변환 계수 레벨, 움직임 벡터, 움직임 화상 참조 인덱스, 스케일 인자, 오디오 엔빌로프 에너지 값 등을 나타내는 신택스 요소와 스케일링 가능하거나 스케일링 불가능한 방식으로 비디오 및/또는 오디오 컨텐츠를 나타낼 수 있다. 신택스 요소는 특히 상이한 타입 또는 카테고리를 가질 수 있으며, 동일한 타입의 신택스 요소는 예컨대 파싱 가능한 비트스트림 내에서 동일한 의미를 갖지만 상이한 화상, 상이한 매크로블록, 상이한 스펙트럼 성분 등과 같은 상이한 부분에 대한 것인 반면, 상이한 타입의 신택스 요소는 움직임 벡터와 같이 비트스트림 내에서 상이한 의미를 가질 수 있으며, 움직임 예측 잔여를 나타내는 변환 계수 레벨을 나타내는 신택스 요소와는 상이한 의미를 갖는다.

[0022] 서브디바이더(100)는 신택스 요소의 타입에 따라 서브디비전(subdivision)을 실행하도록 구성된다. 즉, 서브디바이더(100)는 제1 타입 그룹의 신택스 요소를 제1 서브시퀀스(108)에 전송할 수 있고, 제1 그룹과 차별화되는 제2 타입 그룹의 신택스 요소를 제2 서브시퀀스(110)에 전송할 수 있다. 서브디바이더(100)에 의해 실행된 서브디비전은, 서브시퀀스(108) 내의 신택스 요소의 심벌 통계가 VLC 인코더(102)에 의해 VLC 인코딩되기에 적절하도록, 즉 본 출원의 명세서의 도입부에서 개괄적으로 기재한 바와 같이 VLC 인코딩의 사용 및 특정한 심벌 통계에 대한 그 적정성에 대한 제약에도 거의 최소 엔트로피를 가능케 하도록 설계될 수 있다. 다른 한편으로, 서브디바이더(100)는 모든 다른 신택스 요소를 제2 서브시퀀스(110)에 전송할 수 있어서, VLC 인코딩에 적절하지 않은 심벌 통계를 갖는 이들 신택스 요소는 더 복잡하지만 더 효율적인-압축비 면에서- PIPE 인코더(104)로 인코딩된다.

[0023] 다음의 도면에 대한 더 상세한 실시예의 경우에서도, PIPE 인코더(104)는 서브시퀀스(110)의 각 신택스 요소를 알파벳 심벌의 각각의 부분 시퀀스에 개별적으로 매핑하여 함께 알파벳 심벌의 전술한 시퀀스(124)를 형성하도록 구성된 심벌라이저(122)를 포함할 수 있다. 다시 말해, 심벌라이저(122)는, 예컨대 서브시퀀스(110)의 소스 심벌이 이미 알파벳 심벌의 각각의 부분 시퀀스로서 나타나는 경우 존재하지 않을 수 있다. 심벌라이저(122)는, 예컨대 서브시퀀스(110) 내의 소스 심벌이 상이한 알파벳, 특히 상이한 수의 가능한 알파벳 심벌을 갖는 알파벳인 경우에 유리하다. 즉, 이 경우, 심벌라이저(122)는 서브스트림(110) 내에 도착한 심벌의 알파벳을 하모나이즈(harmonize)할 수 있다. 심벌라이저(122)는 예컨대 서브시퀀스(110) 내에 도착한 심벌을 이진화하도록 구성된 이진화기로서 구현될 수 있다.

[0024] 전술한 바와 같이, 신택스 요소는 상이한 타입일 수 있다. 이것은 서브스트림(110) 내의 신택스 요소에 대해서도 사실일 수 있다. 심벌라이저(122)는 이때 상이한 타입의 신택스 요소마다 상이한, 이진화 방식과 같은 심벌라이징 매핑 방식을 사용하여 서브시퀀스(110)의 신택스 요소의 개별 매핑을 실행하도록 구성될 수 있다. 특정한 이진화 방식의 예는 다음의 설명에서 일진 이진화 방식(unary binarization scheme), 예컨대 순서 0 또는 순서 1의 exp-골롬(Golomb) 이진화 방식, 트렁케이트된(truncated) 일진 이진화 방식, 트렁케이트되고 재정렬된 exp-골롬 순서 0 이진화 방식 또는 비-체계적인 이진화 방식과 같은 것으로 제공된다.

[0025] 따라서, 엔트로피 인코더(116)는 이진 알파벳에서 동작하도록 구성될 수 있다. 마지막으로, 심벌라이저(122)는 도 1a에 도시된 바와 같이 PIPE 인코더(104) 자체의 일부분으로서 간주될 수 있음을 주목해야 한다. 택일적으로, 그러나 이진화기는 PIPE 인코더 외부에 있는 것으로 간주될 수 있다.

[0026] 후자의 경우와 유사하게, 할당기(114)는, 심벌라이저(122)와 선택기(120) 사이에 직렬로 연결되도록 도시될지라도, 택일적으로는 심벌라이저(124)의 출력과 선택기(120)의 제1 입력 사이에 연결되는 것으로 간주될 수 있으며, 할당기(114)의 출력은 도 3에 관해 후에 기술될 바와 같이 선택기(120)의 다른 입력에 연결됨을 주목해야 한다. 실제로, 할당기(114)는 확률 분포의 추정에 대한 전술한 추정치를 갖는 각 알파벳 심벌을 수반한다.

[0027] 도 1a의 엔트로피 인코딩 장치의 출력에 관한 한, 출력은 VLC 인코더(102)에 의해 출력된 제1 비트스트림(112)과 복수의 엔트로피 인코더(116)에 의해 출력된 복수의 제2 비트스트림(118)으로 구성된다. 더 후술될 바와 같이, 모든 이들 비트스트림은 병렬로 송신될 수 있다. 택일적으로, 이들 비트스트림은 인터리버(128)에 의해 공통 비트스트림(126)으로 인터리빙될 수 있다. 도 22 내지 도 24의 실시예는 그러한 비트스트림 인터리빙을 갖는 예를 도시한다. 도 1에 더 도시되는 바와 같이, PIPE 인코더(104) 자체는 그 자신의 인터리버(130)를 포함할 수 있어서, 복수의 제2 비트스트림(118)을 공통 PIPE 코딩된 비트스트림(132)으로 인터리빙할 수 있다. 그러한 인터리버(130)에 대한 가능성은 도 5 내지 도 13에 관해 기술할 다음의 실시예로부터 유도할 수 있다. 비트스트림(132)과 비트스트림(112)은 병렬 구성으로 도 1a의 엔트로피 인코딩 장치의 출력을 나타낼 수 있다. 택일적으로, 다른 인터리버(134)가 두 비트스트림을 인터리빙할 수 있으며, 그러한 경우에, 인터리버(130 및

134)는 하나의 2-스테이지 인터리버(128)의 두 스테이지를 형성할 것이다.

[0028] 전술한 바와 같이, 서브디바이더(100)는 신택스-요소-방식의 세분을 실행할 수 있다. 즉, 서브디바이더(100)가 동작하는 소스 심벌은 전체 신택스 요소일 것이거나, 택일적으로는 서브디바이더(100)는 신택스 요소 단위로 동작할 수 있다.

[0029] 그러나 대안적인 실시예에 따라, 도 1a의 엔트로피 인코딩 장치는, 파싱 가능한 비트스트림(138) 내의 신택스 요소를 서브디바이더(100)에 진입하는 소스 심벌 시퀀스(106)의 소스 심벌 중 하나 이상으로 개별적으로 분해하기 위해 분해기(136)를 포함할 수 있다. 특히, 분해기(136)는, 각 신택스 요소를 각각의 정수의 소스 심벌로 개별적으로 분해함으로써 신택스 요소의 시퀀스(138)를 소스 심벌의 시퀀스(106)로 전환하도록 구성될 수 있다. 정수는 신택스 요소 사이에서 변할 수 있다. 특히, 신택스 요소 중 일부는 분해기(136)에 의해 불변인 채로 남겨질 수 도 있지만, 다른 신택스 요소는 정확히 2개, 또는 적어도 2개의 소스 심벌로 분해된다. 서브바이더(100)는 그렇게 분해된 신택스 요소의 소스 심벌 중 하나를 소스 심벌의 제1 서브시퀀스(108)에 전송하고, 그 분해된 신택스 요소의 소스 심벌 중 다른 하나를 소스 심벌의 제2 서브시퀀스(110)에 전송하도록 구성될 수 있다. 전술한 바와 같이, 비트스트림(138) 내의 신택스 요소는 상이한 타입을 가질 수 있고, 분해기(136)는 신택스 요소의 타입에 따라 개별 분해를 실행하도록 구성될 수 있다. 분해기(136)는 바람직하게는, 정수 개수의 소스 심벌로부터 모든 신택스 요소에 대해 공통인 각각의 신택스 요소로의, 디코딩 측에서 후에 사용되는 미리 결정된 고유한 역 매핑이 존재하도록 신택스 요소의 개별 분해를 실행한다.

[0030] 예컨대, 분해기(136)는 파싱 가능한 비트스트림(138)의 신택스 요소(z)를 2개의 소스 심벌(x 및 y)로 분해하여, z=x+y, z=x-y, z=x · y 또는 z=x:y가 되도록 구성될 수 있다. 이 조치에 의해, 서브디바이더(100)는 신택스 요소를 2개의 구성요소, 즉 소스 심벌 스트림(106)의 소스 심벌로 분해할 수 있으며, 그 심벌 중 하나는 x와 같이 압축 효율 면에서 VLC 인코딩되기에 적절할 수 있으며, 다른 하나는 y와 같이 VLC 인코딩에 적절하지 않으며 그러므로 제1 서브스트림(108)이 아닌 제2 서브스트림(110)으로 전달된다. 분해기(136)에 의해 사용되는 분해는 바이젝티브(bijective)가 될 필요는 없다. 그러나 전술한 바와 같이, 분해가 바이젝티브가 아닌 경우 분해기(136)가 선택할 수 있는 가능한 분해의 신택스 요소의 고유한 회수를 가능케 하는 역 매핑이 존재해야 한다.

[0031] 이제까지, 상이한 신택스 요소를 다루는데 있어서의 상이한 가능성에 대해 기재하였다. 그러한 신택스 요소나 경우가 존재하는지에 대해서는 부가적이다. 본 명세서에 기재한 실시예는 그러나 다음의 원리에 따라 분해기(136)에 의해 분해되는 신택스 요소를 주로 다룬다.

[0032] 도 1b에 도시된 바와 같이, 분해기(136)는 스테이지에서 파싱 가능한 비트스트림(138)의 특정 신택스 요소(z)를 분해하도록 구성된다. 둘 이상의 스테이지가 존재할 수 있다. 스테이지는 신택스 요소(z)의 값 범위를 도 1c에 도시된 바와 같이 둘 이상의 인접한 세부 간격 또는 세부 범위로 분리하기 위한 것이다. 신택스 요소의 값 범위는 두 개의 무한 종점, 단지 하나를 가질 수 있거나, 명백한 종점(definite endpoint)을 가질 수 있다. 도 1c에서, 신택스 요소의 값 범위는 예시적으로는 세 개의 분할(140_{1-3})로 세분된다. 도 1b에 도시된 바와 같이, 신택스 요소가 제1 분할(140_1)의 경계(142), 즉 분할(140_1 및 140_2)을 분리하는 상한 이상이라면, 신택스 요소는 제1 분할(1401)의 경계 한계 1로부터 감산되어 z는 다시 이것이 제2 분할(142_2)의 경계(144), 즉 분할(140_2 및 140_3)을 분리하는 상한 이상인지에 대해 체크된다. z'가 경계(144) 이상이라면, z'는 제2 분할 (140_2)의 경계 한계 2로부터 감산되어 z"를 얻는다. z가 한계 1 미만인 제1 경우에, 신택스 요소(z)는 플레인(plain)에서 서브디바이더(100)에 전송된다. z가 한계 1과 한계 2 사이에 있는 경우, 신택스 요소(z)는 z=한계 1+z'인 투플(tuple)(한계 1, z')로서 서브디바이더(100)로 전송되며, z가 한계 2를 초과한 경우에, 신택스 요소(z)는 z=한계 1+한계 2+z'인 트리플렛(triplet)(한계 1, 한계 2-한계 1, z')으로서 서브디바이더(100)에 전송된다. 제1(또는 단독) 구성요소, 즉 z 또는 한계 1이 서브디바이더(100)에 의해 코딩될 제1 소스 심벌을 형성하며, 제2 구성요소, 즉 z' 또는 한계 2-한계 1은 존재하는 경우 서브디바이더(100)에 의해 코딩될 제2 소스 심벌을 형성하며, 제3 구성요소, 즉 z"는 존재하는 경우 서브디바이더(100)에 의해 코딩될 제3 소스 심벌을 형성한다. 따라서, 도 1b 및 도 1c의 실시예에 따라, 신택스 요소는 1 내지 3개의 소스 심벌 중 임의의 심벌에 의해 매핑되지만, 더 적거나 많은 최대 개수의 소스 심벌로의 일반화는 상기 설명으로부터 쉽게 유도할 수 있으며, 그러한 대안은 또한 다음의 설명에서 기재될 것이다.

[0033] 어느 경우에도, 이들 모든 상이한 구성요소나 결과적인 소스 심벌은 코딩 대안 중에서 코딩될 하기 실시예에 따른다. 이들 중 적어도 하나가 서브디바이더에 의해 PIPE 코더(104)에 전송되며, 마지막으로 그들 중 다른 하나는 VLC 코더(102)에 전송된다.

[0034] 특정한 유리한 실시예를 아래에서 더 상세하게 개괄적으로 설명한다.

[0035] 엔트로피 인코딩 장치에 대한 실시예를 앞서 기재한 후, 엔트로피 디코딩 장치의 실시예를 도 2a와 관련하여 기재한다. 도 2a의 엔트로피 디코딩 장치는 VLC 디코더(200)와 PIPE 디코더(202)를 포함한다. VLC 디코더(200)는 제1 비트스트림(206)의 코드워드로부터 제1 서브시퀀스(204)의 소스 심벌을 코드-방식으로 재구성하도록 구성된다. 제1 비트스트림(206)은 도 1의 비트스트림(112)과 같으며, 도 1a의 서브시퀀스(108)에 관한 한 동일한 사실이 서브시퀀스(204)에도 적용된다. PIPE 디코더(202)는 알파벳 심벌 시퀀스 형태로 표현된 소스 심벌의 제2 서브시퀀스(208)를 재구성하도록 구성되며, 복수의 엔트로피 디코더(210), 할당기(212) 및 선택기(214)를 포함한다. 복수의 엔트로피 디코더(210)는 제2 비트스트림(216) 중 각각의 하나를 알파벳 심벌의 시퀀스 중 알파벳 심벌로 전환하도록 구성된다. 할당기(212)는, 각각의 알파벳 심벌이 가질 가능한 값 사이의 확률 분포의 추정의 측정치를, 알파벳 심벌의 시퀀스의 이전에 재구성된 알파벳 심벌 내에 포함된 정보를 기반으로 해서, 재구성될 소스 심벌의 제2 서브시퀀스(208)를 표현하는 알파벳 심벌의 시퀀스의 각 알파벳 심벌에 할당하도록 구성된다. 이를 위해, 할당기(212)는 선택기(214)의 출력과 그 입력 사이에서 직렬로 연결될 수 있는 반면, 선택기(214)의 다른 입력은 각각 그에 연결되는 엔트로피 디코더(210)의 출력을 갖는다. 선택기(214)는 복수의 엔트로피 디코더(210) 중 선택된 하나로부터 알파벳 심벌의 시퀀스 중 각 알파벳 심벌을 회수하도록 구성되며, 그러한 선택은 각각의 알파벳 심벌에 할당된 측정치에 따른다. 다시 말해, 할당기(212)와 함께 선택기(214)는, 알파벳 심벌의 시퀀스의 이전 알파벳 심벌 내에 포함된 정보를 조사하여 얻은 엔트로피 디코더(210) 중의 순서로 엔트로피 디코더(210)에 의해 얻은 알파벳 심벌을 회수하도록 동작한다. 다시 말해, 할당기(212)와 선택기(214)는, 알파벳 심벌 사이에서 알파벳 심벌의 원래 순서로 재구성할 수 있다. 그 다음 알파벳 심벌을 예상함과 함께, 할당기(212)는 각각의 알파벳 심벌에 대한 확률 분포의 추정의 전술한 측정치를 결정할 수 있고, 이것을 사용하여, 선택기(214)는 엔트로피 디코더(210) 중에서 선택하여 이 알파벳 심벌의 실제 값을 회수한다. 더 정밀하게 말하면, 더 상세하게 후술될 바와 같이, PIPE 디코더(202)는, 알파벳 심벌을 순차적으로 요청하고 있는 알파벳 심벌 요청에 응답하여 알파벳 심벌의 시퀀스의 형태로 표현된 소스 심벌의 서브시퀀스(208)를 재구성하도록 구성될 수 있고, 할당기(212)는, 재구성될 소스 심벌의 제2 서브시퀀스(208)을 표현하는 알파벳 심벌의 시퀀스의 알파벳 심벌에 대한 각 요청에, 각각의 알파벳 심벌이 가질 수 있는 가능한 값 중의 확률 분포의 추정의 전술한 측정치를 할당하도록 구성될 수 있다. 따라서, 선택기(214)는, 재구성될 소스 심벌의 제2 서브시퀀스(208)를 표현하는 알파벳 심벌의 시퀀스의 알파벳 심벌에 대한 각 요청에 대해, 복수의 엔트로피 디코더(210) 중 선택된 하나로부터 알파벳 심벌의 시퀀스의 각각의 알파벳 심벌을 회수하도록 구성될 수 있으며, 이러한 선택은 각각의 알파벳 심벌에 대한 각각의 요청에 할당된 측정치에 의존한다. 한편으론 디코딩 측에서의 요청과 다른 한편으론 인코딩 측에서의 데이터 흐름 또는 인코딩 사이의 용어 색인(concordance)은 도 4에 관련하여 더 상세하게 개괄적으로 설명할 것이다.

[0036] 소스 심벌의 제1 서브시퀀스(214)와 소스 심벌의 제2 서브시퀀스(208)은 소스 심벌의 하나의 공통 시퀀스(210)를 공통적으로 형성하므로, 도 2a의 엔트로피 디코딩 장치는, 부가적으로, 소스 심벌의 공통 시퀀스(218)를 얻기 위해 제1 서브시퀀스(204)와 제2 서브시퀀스(208)를 재결합하도록 구성된 재결합기(220)를 포함할 수 있다. 소스 심벌의 이 공통 시퀀스(208)는 도 1a의 시퀀스(106)의 재구성을 산출한다.

[0037] 도 1에 관련하여 앞서 제시한 기재에 따라, 제1 및 제2 서브시퀀스(204 및 208)의 소스 심벌은 파싱 가능한 비트스트림의 신택스 요소일 수 있다. 이 경우, 재결합기(220)는, 신택스 요소 사이에서 순서를 정의하는 일부 파싱 규칙에 의해 규정된 순서로 제1 및 제2 서브시퀀스(204 및 208)를 통해 도착한 소스 심벌을 인터리빙함으로써 신택스 요소의 시퀀스(218)의 이 파싱 가능한 비트스트림을 재구성하도록 구성될 수 있다. 특히, 신택스 요소는, 전술한 바와 같이, 상이한 타입일 수 있으며, 재결합기(220)는 서브스트림(204)을 통해 VLC 디코더(200)로부터 제1 타입 그룹의 신택스 요소를 회수 또는 요청하고, 서브스트림(208)을 통해 PIPE 디코더(202)로부터 제2 타입의 신택스 요소를 회수 또는 요청하도록 구성될 수 있다. 따라서, 바로 앞에서 기재한 파싱 규칙이 제1 그룹 내의 타입의 신택스 요소가 라인에서 그 다음이라고 지시한다면, 재결합기(202)는 서브시퀀스(204)의 실제 소스 심벌을 공통 시퀀스(218)에 삽입하고, 그 밖에 서브시퀀스(208)로부터의 것을 삽입한다.

[0038] 마찬가지로, PIPE 디코더(202)는 선택기(214)의 출력과 재결합기(220)의 입력 사이에 연결된 디심벌라이저(222)를 포함할 수 있다. 도 1과 관련하여 앞서 기재된 것과 유사하게, 디심벌라이저(222)는 PIPE 디코더(202)의 외부에 있는 것으로 간주될 수 있고, 택일적으로는 재결합기(202) 뒤, 즉 재결합기(220)의 출력 측에 배치될 수 있다. 디심벌라이저(222)는, 알파벳 심벌의 부분 시퀀스의 단위에서, 선택기(214)에 의해 출력된 알파벳 심벌(224)의 시퀀스를 소스 심벌, 즉 서브시퀀스(208)의 신택스 요소에 리매핑하도록 구성될 수 있다. 재결합기(220)와 유사하게, 디심벌라이저(222)는 알파벳 심벌의 가능한 부분 시퀀스의 구성에 대해 알고 있다. 특히, 디

심벌라이저(222)는, 이들 최근에 수신한 알파벳 심벌이 각각의 신택스 요소의 각각의 값과 관련된 알파벳 심벌의 유효 부분 시퀀스를 산출하는지에 대해서나 그런 경우가 아닌지, 그리고 어떤 알파벳 심벌이 그 다음에 분실 중인지에 대해 확인하기 위해 선택기(214)로부터 최근에 수신된 알파벳 심벌을 분석할 수 있다. 다시 말해, 디심벌라이저(222)는, 언제라도, 각각의 신택스 요소의 수신을 끝내기 위해 추가 알파벳 심벌이 선택기(214)로부터 수신되어야 하는지에 대해 그리고 따라서 선택기(214)에 의해 출력된 알파벳 심벌 중 각각의 하나가 어떤 신택스 요소에 속해 있는지를 안다. 이를 위해, 디심벌라이저(222)는 상이한 타입의 신택스 요소에 대해 상이한 심벌라이징 (디)매핑 방식을 사용할 수 있다. 유사하게, 할당기(212)는 선택기(214)에 의해 엔트로피 디코더(210) 중 임의의 것으로부터 회수될 현재 알파벳 심벌의 신택스 요소의 각각 하나에 대한 관계를 알며, 따라서, 즉 관련된 신택스 요소 타입에 따라서, 이 알파벳 심벌의 확률 분포의 추정의 전술한 측정치를 설정할 수 있다. 또한, 할당기(212)는 현재의 알파벳 심벌의 동일한 부분 시퀀스에 속한 상이한 알파벳 심벌 사이를 구별할 수 있고, 이들 알파벳 심벌에 대해 확률 분포의 추정의 측정치를 상이하게 설정할 수 있다. 이에 관한 세부 내용은 아래에서 더 상세하게 기재할 것이다. 본 명세서에서 기재한 바와 같이, 할당기(212)는 컨텍스트를 알파벳 심벌에 할당하도록 구성될 수 있다. 이러한 할당은 현재 신택스 요소의 알파벳 심벌의 부분 시퀀스 내의 신택스 요소 타입 및/또는 위치에 의존할 수 있다. 할당기(212)가 컨텍스트를 선택기(214)에 의해 엔트로피 디코더(210) 중 어느 하나로부터 회수될 현재의 알파벳 심벌에 할당하자마자, 알파벳 심벌은, 각 컨텍스트가 그와 관련된 추정의 측정치를 가짐에 따라 그와 관련된 확률 분포의 추정의 측정치를 본래 가질 수 있다. 또한, 컨텍스트-및 그와 관련된 확률 분포의 추정의 측정치-는 이제까지 엔트로피 디코더(210)로부터 회수된 각각의 컨텍스트의 알파벳 심벌의 실제 통계에 따라 적응될 수 있다. 이에 관한 세부 내용은 더 상세하게 후술될 것이다.

[0039] 도 1의 상기 설명과 유사하게, 신택스 요소에서 서브시퀀스(204 및 208)의 전술한 소스 심벌 사이의 상관관계는 일대일 상관관계가 아닐 수 있다. 오히려, 신택스 요소는, 결국 신택스 요소 사이에서 다르지만, 어느 경우에도 하나의 신택스 요소에 대해 적어도 1보다 큰 숫자를 갖는 정수 개의 소스 심벌로 분해되었을 수 있다. 앞서 주의한 바와 같이, 다음의 실시예는 이러한 종류의 신택스 요소의 처리를 주로 다루며, 다른 종류의 신택스 요소는 더는 존재하지 않을 수 있다.

[0040] 바로 앞에서 기재했던 신택스 요소를 처리하기 위해, 도 2a의 엔트로피 디코딩 장치는 도 1a의 분해기(136)에 의해 실행된 분리를 다시 행하도록 구성된 구성기(224)를 포함할 수 있다. 특히, 구성기(224)는 시퀀스(218)의 소스 심벌로부터 신택스 요소의 시퀀스(226)를 구성하거나, 재결합기(220)가 손실된 경우, 제1 서브시퀀스(204)에 속한 정수 개수의 소스 심벌의 소스 심벌 중 하나와, 제2 서브시퀀스(208)에 속한 동일한 신택스 요소의 정수 개수의 소스 심벌의 소스 심벌 중 다른 하나로 각각의 정수 개수의 소스 심벌로부터 각 신택스 요소를 개별적으로 구성함으로써 서브시퀀스(204 및 208)를 구성하도록 구성될 수 있다. 이러한 조치에 의해, 특정한 신택스 요소가 PIPE 디코딩 경로를 통해야 하는 나머지 구성요소로부터 VLC 디코딩에 적절한 구성요소를 분리하도록 디코더 측에서 분리될 수 있다. 상기 논의와 유사하게, 신택스 요소는 상이한 타입을 가질 수 있으며, 구성기(224)는 신택스 요소의 타입에 따라 개별 구성을 실행하도록 구성될 수 있다. 특히, 구성기(224)는, 각각의 신택스 요소의 정수 개수의 소스 심벌을 논리적으로나 수학적으로 결합함으로써 각각의 신택스 요소를 얻도록 구성될 수 있다. 예컨대, 구성기(224)는, 각 신택스 요소마다, 하나의 신택스 요소의 제1 및 제2 소스 심벌에 +, -, : 또는 · 를 적용하도록 구성될 수 있다.

[0041] 전술한 바와 같이, 이후에 논의할 설명은 그러나 도 1b 및 도 1c에 따른 분해기(136)와 그에 관해 기술된 대안에 의해 분해된 신택스 요소를 주로 다룬다. 도 2a는 이들 신택스 요소를 그들의 소스 심벌(218)로부터 재구성하기 위해 구성기(224)가 어떻게 기능하는지를 도시한다.

[0042] 도 2b에 도시된 바와 같이, 구성기(224)는 본 예에서는 1 내지 3 중 어느 하나인 x를 갖는 입력 소스 심벌(s_1 내지 s_x)로부터 스테이지에서 그러한 신택스 요소(z)를 구성하도록 구성된다. 둘 이상의 스테이지가 존재할 수 있다. 도 2b에 도시된 바와 같이, 구성기(224)는 미리 z를 제1 심벌(s_1)로 설정하고, z가 제1 한계와 같은지를 체크한다. 그런 경우가 아니라면, z가 찾아졌다. 그 밖의 경우에, 구성기(224)는 소스 심벌 스트림(218)의 그 다음 소스 심벌(s_2)을 z에 더하며, 다시 이 z가 한계 2와 같은지를 체크한다. 그렇지 않다면, z가 찾아졌다. 그렇지 않은 경우, 구성기(224)는 소스 심벌 스트림(218)의 그 다음 소스 심벌(s_3)을 z에 더하여, z를 그 최종 형태로 얻는다. 더 적거나 많은 최대 개수의 소스 심벌에 대한 일반화는 상기 설명으로부터 쉽게 유도할 수 있으며, 그러한 대안은 또한 다음에서 기재될 것이다.

[0043] 어느 경우에도, 이들 모든 상이한 구성요소나 결과적인 소스 심벌은 아래의 실시예에 따라 코딩 대안 중에서 코

딩된다. 이들 중 적어도 하나는 서브디바이더에 의해 PIPE 코더(104)에 전송되며, 그 중 마지막 다른 하나는 VLC 코더(102)에 전송된다.

[0044] 특정한 유리한 실시예를 이하에서 더 상세하게 개괄적으로 논의할 것이다. 이들 세부 내용은, 소스 심벌을 인코딩하는데 사용될 수 있는 엔트로피 VLC 및 PIPE 코딩 방식과 신택스 요소의 값 범위를 분리하는 유리한 가능성을 주로 다룬다.

[0045] 또한, 도 1에 관해 또한 앞서 기재했던 바와 같이, 도 2a의 엔트로피 디코딩 장치는 인터리빙된 비트스트림(228)에 의해 인터리빙된 형태로나 개별적으로 제1 비트스트림(206) 뿐만 아니라 복수의 제2 비트스트림(216)을 수신하도록 구성될 수 있다. 후자의 경우, 도 2a의 엔트로피 디코딩 장치는, 인터리빙된 비트스트림(228)을 디인터리빙하여 한편으론 제1 비트스트림(206)을 얻고 다른 한편으론 복수의 제2 비트스트림(216)을 얻도록 구성되는 디인터리버(230)를 포함할 수 있다. 도 1의 앞선 논의와 유사하게, 디인터리버(230)는, 두 스테이지, 즉 인터리빙된 비트스트림(228)을 두 부분, 즉 한편으론 비트스트림(206)과 다른 한편으론 제2 비트스트림(216)의 인터리빙된 형태(234)로 디인터리빙하기 위한 디인터리버(232)와, 후자의 비트스트림(234)을 디인터리빙하여 개별 비트스트림(216)을 얻는 디인터리버(236)로 세분될 수 있다.

[0046] 도 1a 및 도 2a는 한편으론 엔트로피 인코딩 장치와, 다른 한편으론 도 1의 엔트로피 인코딩 장치에 의해 얻은 인코딩 결과를 디코딩하는데 적절한 엔트로피 디코딩 장치를 도시하였다. 도 1a 및 도 2에 도시한 요소 중 많은 것에 관한 세부 내용은 추가 도면과 관련하여 더 상세하게 기재될 것이다. 따라서, 다음의 기재에서 이들 세부 내용을 참조할 것이며, 이들 세부 내용은, 이들 내용이 상기 기재한 실시예에서 개별적으로 구현될 수 있는 한, 도 1a 및 도 2의 실시예에 개별적으로 또한 적용되는 것으로서 간주되어야 한다. 단지 인터리버(132)와 디인터리버(234)에 관해서, 일부 추가로 주의를 해야 한다. 특히, 비트스트림(112 및 118)의 인터리빙은, 비트스트림이 송신되기 위해 한 채널로 다중화되어야 할 경우에 유리할 수 있다. 이 경우, 일부 최대 디코딩 지연에 따르는 것과 같은 충족되어야 할 특정 조건에 따르기 위해서 한편으론 VLC 비트스트림(112)과 다른 한편으론 PIPE 인코딩 비트스트림(118)을 인터리빙하는 것이 유리할 수 있다. 다시 말해, 신택스 요소와 소스 심벌 각각의 시간 사이의 상대적인 시간 변위는 한편으론 디코딩 측에서 회수될 수 있고, 다른 한편으론 파싱 가능한 비트스트림에서의 그 위치에 따른 시간적인 상대적 변위가 특정 최대 지연을 초과하지 않는 것이 필요할 수 있다. 이러한 문제를 해결하기 위한 많은 대안을 후술할 것이다. 이들 가능성 중 하나는 알파벳 심벌 시퀀스를 코드워드로 매핑하도록 구성된 가변 길이 코더 타입을 갖는 엔트로피 인코더(116)와, 역 매핑을 행하는 엔트로피 디코더(210)를 수반한다. VLC 비트스트림(112)과 PIPE 비트스트림(118)의 코드워드는, 이들 비트스트림 중 임의의 비트스트림의 코드워드가 다른 비트스트림 중 임의의 비트스트림의 임의의 코드워드의 프레픽스(prefix)가 되지 않도록 선택되어(그러나 반드시 이렇게 되어야 할 필요는 없음), 코드워드 경계가 디코더 측에서 고유하게 결정 가능한 상태로 남게 된다. 어느 경우에도, 각각 인터리버(128)는, 각각 선택기(120)에 의해 복수의 엔트로피 인코더(116)에 전송된 알파벳 심벌의 시퀀스(124)의 알파벳 심벌이 각각의 엔트로피 인코더(116)에서 각각의 코드워드에 매핑될 새로운 심벌 시퀀스의 시작이 되고, 제2 비트스트림(108)의 새로운 소스 심벌이 VLC 인코더(102)에 의해 매핑되는 순서에 따라 순차적인 순서로 제1 비트스트림(112)과 제2 비트스트림(118) 내의 코드워드의 코드워드 엔트리 시퀀스를 예비하고 버퍼링하도록 구성될 수 있다. 다시 말해, 인터리버(128)는 비트스트림(112)의 코드워드를 공통 비트스트림(126)내로, 각각 서브스트림(108) 및 소스 심벌 스트림(106) 내의 그 순서로 VLC 인코딩에 의해 얻어졌던 소스 심벌의 순서로 삽입한다. 엔트로피 인코더(116)에 의해 출력된 코드워드는 VLC 비트스트림(112)의 코드워드 중 연속되는 코드워드 사이의 공통 비트스트림(126)에 삽입된다. 할당기(114)와 선택기(120) 각각에 의한 알파벳 심벌의 PIPE 인코딩 분류로 인해, 엔트로피 인코더(116)의 코드워드 각각은 인코딩된 서브스트림(110)의 상이한 소스 심벌의 알파벳 심벌을 그 내부에 갖고 있다. 서로 간에 및 비트스트림(112)의 VLC 코드워드에 대한 공통 비트스트림(126) 내의 PIPE 인코딩된 비트스트림(118)의 코드워드의 위치는 각각 각 코드워드에서 인코딩된 제1 알파벳 심벌, 즉 시간적으로 가장 오래된 심벌에 의해 결정된다. 알파벳 심벌 스트림(124)에서 비트스트림(118)의 코드워드로 인코딩되는 이들 1차 알파벳 심벌의 순서가 공통 비트스트림(126) 내의 비트스트림(118)의 코드워드의 서로 간의 순서를 결정하고, 비트스트림(112)의 VLC 코드워드에 대해, 비트스트림(118)의 코드워드로 인코딩된 이들 1차 알파벳 심벌이 속한 소스 심벌이 비트스트림(118) 중 임의의 비트스트림의 각각의 코드워드가 비트스트림(112)의 어떤 연속하는 코드워드 사이에서 위치하게 되는지를 결정한다. 특히, 비트스트림(118) 중 임의의 비트스트림의 각각의 코드워드가 그 사이에 위치하게 될 연속하는 VLC 코드워드는, 비트스트림(118)의 각각의 코드워드로 인코딩된 각각의 1차 알파벳 심벌이 속한 세분되지 않은 소스 심벌 스트림(106)에 따라 서브스트림(110)의 소스 심벌이 그 사이에 위치하는 코드워드이다. 인터리버(128)는 인터리빙된 코드워드의 공통 비트스트림(126)을 얻기 위해 순차적인 순서로 전술한 코드워드 엔트리에 입력된 코드워드를 제거하도록 구성될 수 있다. 이미 전술된 바와 같이, 엔트로피 인코더(116)는 그 코드워

드를 각각의 엔트로피 인코더(116)를 위해 예비한 코드워드 엔트리로 순차적으로 입력하도록 구성될 수 있고, 선택기(120)는, 제1 서브스트림(108)과 제2 서브스트림(110)의 소스 심벌이 소스 심벌의 시퀀스(106) 내에서 인터리빙되었던 순서를 유지하는 순서로 제2 서브스트림(110)의 소스 심벌을 표현하는 알파벳 심벌을 전송하도록 구성될 수 있다.

[0047] 엔트로피 인코더(116) 중 특정한 인코더들이 좀처럼 선택되지 않아서, 그처럼 매우 드물게 사용되는 엔트로피 인코더(116) 내의 유효한 코드워드를 얻는데 오랜 시간이 걸리는 상황을 대처하기 위해 추가 조치를 제공할 수 있다. 그러한 조치의 예를 더 상세하게 후술할 것이다. 특히, 엔트로피 인코더(116)와 함께 인터리버(128)는, 이 경우에는, 이제까지 수집된 그 알파벳 심벌과, 전술한 코드워드 엔트리 내로 입력되었던 코드워드 각각을, 플러싱 절차 시간을 디코딩 측에서 예상하거나 에뮬레이트(emulate)할 수 있는 방식으로 플러싱(flushing)하도록 구성될 수 있다.

[0048] 디코딩 측에서, 디인터리버(230)는 역의 개념으로서 동작할 수 있다: 전술한 파싱 방식에 따라, 디코딩될 그 다음 소스 심벌이 VLC 코딩된 심벌일 때마다, 공통 비트스트림(228) 내의 현재의 코드워드는 VLC 코드워드로서 간주되고 비트스트림(206) 내에서 VLC 디코더(200)로 전송된다. 다른 한편, 서브스트림(208)의 PIPE 인코딩된 심벌 중 임의의 심벌에 속한 알파벳 심벌 중 임의의 심벌이 1차 알파벳 심벌일 때마다, 즉 비트스트림(216) 중 각각의 비트스트림의 코드워드의 각각의 엔트로피 디코더(210)에 의한 각각의 알파벳 심벌 시퀀스로의 새로운 매핑을 필요로 할 때마다, 공통 비트스트림(228)의 현재의 코드워드는 PIPE 인코딩된 코드워드로 간주되며 각각의 엔트로피 디코더(210)에 전송된다. 그 다음 코드워드 경계의 검출, 즉 각각 디코더(200 및 202) 중 임의의 디코더로 막 전송되었던 코드워드의 끝으로부터, 들어오는 인터리빙된 비트스트림(228) 내의 코드워드의 끝으로의 그 다음 코드워드의 확장의 검출은 연기될 수 있으며, 앞서 개괄적으로 설명한 규칙에 따라 이 그 다음 코드워드의 전용 수신자인 디코더(200 및 202)의 인지 하에서 실행될 수 있다: 이러한 인지를 기반으로 하여, 수신자 디코더에 의해 사용된 코드북이 알려져 있고, 각각의 코드워드는 검출 가능하다. 만약 다른 한편으로 코드북이, 코드워드 경계가 200 및 202 중 수신자 디코더에 대한 선험적인(a-priori) 인지 없이 검출되도록 설계된다면, 코드워드 분리는 병렬로 실행될 수 있다. 어느 경우에도, 인터리빙으로 인해, 소스 심벌은 엔트로피 디코딩된 형태로, 즉 소스 심벌로서, 합리적으로 지연된 그 정확한 순서로 디코더에서 이용 가능하다.

[0049] 엔트로피 인코딩 장치 및 각각의 엔트로피 디코딩 장치에 대한 전술한 실시예를 기재한 후, 전술한 PIPE 인코더 및 PIPE 디코더에 대한 더 상세한 실시예를 이제 기재할 것이다.

[0050] 실시예에 따른 PIPE 인코더를 도 3에 예시한다. 동일한 인코더가 도 1a에서 PIPE 인코더로서 사용될 수 있다. PIPE 인코더는 소스 심벌(1)의 스트림을 둘 이상의 부분 비트스트림(12)의 세트로 손실 없이 전환한다. 각 소스 심벌(1)은 하나 이상의 카테고리나 타입의 세트의 카테고리나 타입과 관련될 수 있다. 예컨대, 카테고리는 소스 심벌 타입을 명시할 수 있다. 하이브리드 비디오 코딩 환경에서, 개별적인 카테고리는 매크로블록 코딩 모드, 블록 코딩 모드, 참조 화상 인덱스, 움직임 벡터차, 서브디비전 플래그, 코딩된 블록 플래그, 양자화 파라미터, 변환 계수 레벨 등과 관련될 수 있다. 오디오, 음성, 문자, 문서, 또는 일반 데이터 코딩과 같은 다른 응용 분야에서, 소스 심벌의 상이한 분류가 가능하다.

[0051] 일반적으로, 각각의 소스 심벌은 유한한 또는 계수 가능한 무한 값 세트의 값을 가질 수 있으며, 여기서 가능한 소스 심벌 값의 세트는 상이한 소스 심벌 카테고리마다 상이할 수 있다. 인코딩 및 디코딩 알고리즘의 복잡도를 줄이고, 상이한 소스 심벌 및 소스 심벌 카테고리에 대해 일반적인 인코딩 및 디코딩 설계를 허용하기 위해, 소스 심벌(1)은 이진 결정(binary decision)의 순서가 정해진 세트로 전환되며, 이들 이진 결정은 그 후 간단한 이진 코딩 알고리즘에 의해 처리된다. 그러므로 이진화기(2)는 각 소스 심벌(1)의 값을 이진수(3)의 시퀀스(또는 열) 상에 바이젝티브 매핑한다(bijectively mapping). 이진수(3)의 시퀀스는 순서가 정해진 이진 결정의 세트를 나타낸다. 각 이진수(3) 또는 이진 결정은 두 값의 세트 중 하나의 값, 예컨대 값 0 및 1 중 하나를 가질 수 있다. 이진 방식은 상이한 소스 심벌 카테고리마다 상이할 수 있다. 특정한 소스 심벌 카테고리에 대한 이진화 방식은 가능한 소스 심벌 값의 세트 및/또는 특정 카테고리에 대한 소스 심벌의 다른 속성에 따라 다를 수 있다. 표 1은 계수 가능한 무한 세트에 대한 세 가지 예시적인 이진화 방식을 예시한다. 계수 가능한 무한 세트에 대한 이진화 방식은 또한 심벌 값의 유한 세트에 적용할 수 있다. 특히 심벌 값의 큰 유한 세트의 경우, (미사용된 이진수 시퀀스로 인한) 비효율성을 무시할 수 있지만, 그러한 이진화 방식의 범용성은 복잡도 및 메모리 요건 면에서 장점을 제공한다. 심벌 값의 작은 유한 세트의 경우, 이진화 방식을 가능한 심벌 값의 수로 적응시키는 것이 (코딩 효율 면에서) 종종 바람직하다. 표 2는 8개의 값의 유한 세트에 대한 세 가지 예시적인 이진화 방식을 예시한다. 유한 세트에 대한 이진화 방식은, 이진수 시퀀스의 유한 세트가 중복 없는 코드(및 잠재적으로는 이진수 시퀀스 순서 재정렬)을 나타내는 방식으로 이진수의 일부 시퀀스를 변경함으로써 계수 가능한 무한

세트에 대한 범용 이진화 방식으로부터 유도할 수 있다. 예로서, 표 2의 트렁케이트된 일진 이진화 방식을 범용 일진 이진의 소스 심벌(7)에 대한 이진수 시퀀스를 변경함으로써 만들었다(표 1 참조). 표 2의 순서 0의 트렁케이트되고 순서 재정렬된 Exp-골롬 이진화는 범용 Exp-골롬 순서 0 이진화의 소스 심벌(7)의 이진수 시퀀스를 변경함으로써(표 1 참조) 그리고 이 이진수 시퀀스의 순서를 재정렬함으로써 만들어졌다(심벌(7)에 대한 트렁케이트된 이진수 시퀀스가 심벌 1에 할당되었다.). 심벌의 유한 세트의 경우, 표 2의 마지막 열에서 예시한 바와 같이, 비-체계적/비-범용 이진화 방식을 사용할 수 도 있다.

표 1

[0052]

심벌 값	일진 이진화	Exp-골롬 순서 0 이진화	Exp-골롬 순서 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
...	...	...	...

[0053] 표 1: 계수 가능한 무한 세트(또는 큰 유한 세트)에 대한 이진화 예

표 2

[0054]

심벌 값	트렁케이트된 일진 이진화	트렁케이트되고 순서 재정렬된 Exp-골롬 순서 0 이진화	비-체계적인 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

[0055] 표 2: 유한 세트에 대한 이진화 예

[0056] 이진화기(2)에 의해 만들어진 이진수의 시퀀스의 각각의 이진수(3)는 순차적인 순서로 파라미터 할당기(4)에 공급된다. 파라미터 할당기는 하나 이상의 파라미터 세트를 각각의 이진수(3)에 할당하여, 관련된 파라미터 세트(5)를 갖는 이진수를 출력한다. 파라미터 세트는 인코더 및 디코더에서 정확히 동일한 방식으로 결정된다. 파라미터 세트는 다음의 파라미터 중 하나 이상으로 구성될 수 있다:

[0057] - 현재의 이진수에 대한 두 개의 가능한 이진수 값 중 하나에 대한 확률의 추정에 대한 측정치,

[0058] - 현재의 이진수에 대한 더 작은 확률이나 더 큰 확률의 이진수 값에 대한 확률의 추정에 대한 측정치,

[0059] - 두 개의 가능한 이진수 값 중 어떤 값이 현재의 이진수에 대한 더 작은 확률이나 더 큰 확률의 이진수 값을 나타내는지에 대한 추정을 명시하는 식별자,

[0060] - 관련된 소스 심벌의 카테고리,

[0061] - 관련된 소스 심벌의 중요도에 대한 측정치,

[0062] - (예컨대, 시간적, 공간적, 또는 용량적(volumetric) 데이터 세트에서의) 관련된 심벌의 위치에 대한 측정치,

[0063] - 이진수 또는 관련된 소스 심벌에 대한 채널 코드 보호를 명시하는 식별자,

[0064] - 이진수 또는 관련된 소스 심벌에 대한 암호화 방식을 명시하는 식별자,

[0065] - 관련된 심벌에 대한 등급을 명시하는 식별자,

[0066] - 관련된 소스 심벌에 대한 이진수의 시퀀스에서의 이진수 숫자.

[0067] 실시예에서, 파라미터 할당기(4)는 각각의 이진수(3 및 5)를 현재의 이진수에 대한 두 개의 가능한 이진수 값 중 하나에 대한 확률의 추정에 대한 측정치와 관련시킨다. 파라미터 할당기(4)는 각각의 이진수(3 및 5)를 현재의 이진수에 대한 더 작은 확률 또는 더 큰 확률의 이진수 값에 대한 확률의 추정에 대한 측정치와, 두 개의 가능한 이진수 값 중 어떤 값이 현재의 이진수에 대한 더 작은 확률이나 더 큰 확률의 이진수 값을 나타내는지에 대한 추정을 명시하는 식별자와 관련시킨다. 더 작은 확률이나 더 큰 확률의 이진수 값에 대한 확률과, 두 개의 가능한 이진수 값 중 어떤 값이 더 작은 또는 더 큰 확률의 이진수 값을 나타내는지를 명시하는 식별자가 두 개의 가능한 이진수 값 중 하나의 확률에 대한 등가의 측정치임을 주목해야한다.

[0068] 다른 실시예에서, 파라미터 할당기(4)는 각각의 이진수(3 및 5)를 현재의 이진수에 대한 두 개의 가능한 이진수 값 중 하나에 대한 확률의 추정에 대한 측정치 및 하나 이상의 다른 파라미터(상기 게재한 파라미터 중 하나 이상일 수 있음)와 관련시킨다. 다른 실시예에서, 파라미터 할당기(4)는 각각의 이진수(3 및 5)를 현재의 이진수에 대한 더 작은 확률이나 더 큰 확률의 이진수 값에 대한 확률의 추정에 대한 측정치, 두 개의 가능한 이진수 값 중 어떤 값이 현재의 이진수에 대한 더 작은 확률이나 더 큰 확률의 이진수 값을 나타내는지에 대한 추정을 명시하는 식별자, 및 하나 이상의 다른 파라미터(상기 게재한 파라미터 중 하나 이상일 수 있음)와 관련시킬 수 있다.

[0069] 실시예에서, 파라미터 할당기(4)는, 하나 이상의 이미 인코딩된 심벌의 세트를 기반으로 하여, 상기 언급한 확률 측정치(현재의 이진수에 대한 두 개의 가능한 이진수 값 중 하나에 대한 확률의 추정에 대한 측정치, 현재의 이진수에 대한 더 작은 확률이나 더 큰 확률의 이진수 값에 대한 확률의 추정에 대한 측정치, 및 두 개의 가능한 이진수 값 중 어떤 값이 현재의 이진수에 대한 더 작은 확률이나 더 큰 확률의 이진수 값을 나타내는지에 대한 추정을 명시하는 식별자) 중 하나 이상을 결정한다. 확률 측정치를 결정하는데 사용되는 인코딩된 심벌은 동일한 심벌 카테고리의 하나 이상의 이미 인코딩된 심벌, (현재의 소스 심벌과 관련된 데이터 세트에 관련하여) 이웃한 공간 및/또는 시간 위치의 (블록이나 샘플 그룹과 같은) 데이터 세트에 대응하는 동일한 심벌 카테고리의 하나 이상의 이미 인코딩된 심벌, 또는 (현재의 소스 심벌과 관련된 데이터 세트에 관련하여) 동일한 및/또는 이웃한 공간 및/또는 시간 위치의 데이터 세트에 대응하는 상이한 심벌 카테고리의 하나 이상의 이미 인코딩된 심벌을 포함한다.

[0070] 파라미터 할당기(4)의 출력인 관련된 파라미터 세트를 갖는 각각의 이진수(5)는 이진수 버퍼 선택기(6)에 공급된다. 이진수 버퍼 선택기(6)는 잠재적으로는, 입력 이진수 값 및 관련된 파라미터(5)를 기반으로 하여 입력된 이진수(5)의 값을 변경하여, 출력 이진수(7)(잠재적으로 변경된 값을 가짐)를 둘 이상의 이진수 버퍼(8) 중 하나에 공급한다. 출력 이진수(7)가 전송된 이진수 버퍼(8)는 입력 이진수(5)의 값 및/또는 관련된 파라미터(5)의 값을 기반으로 하여 결정된다.

[0071] 실시예에서, 이진수 버퍼 선택기(6)는 이진수의 값을 변경하지 못한다, 즉 출력 이진수(7)는 입력 이진수(5)와 항상 동일한 값을 갖는다.

[0072] 다른 실시예에서, 이진수 버퍼 선택기(6)는 입력 이진수 값(5)과, 현재의 이진수에 대한 두 개의 가능한 이진수 값 중 하나에 대한 확률의 추정에 대한 관련 측정치를 기반으로 하여 출력 이진수 값(7)을 결정한다. 실시예에서, 출력 이진수 값(7)은, 현재의 이진수에 대한 두 개의 가능한 이진수 값 중 하나에 대한 확률에 대한 측정치가 특정한 임계치 미만이라면(또는 그 이하라면), 입력 이진수 값(5)과 동일하게 설정되고, 현재의 이진수에 대한 두 개의 가능한 이진수 값 중 하나에 대한 확률에 대한 측정치가 특정한 임계치 이상이라면(또는 그것을 초과한다면), 출력 이진수 값(7)은 변경된다(즉, 이 값은 입력 이진수 값의 반대로 설정된다). 다른 실시예에서, 출력 이진수 값(7)은, 현재의 이진수에 대한 두 개의 가능한 이진수 값 중 하나에 대한 확률에 대한 측정치가 특정 임계치를 초과한다면(또는 그 이상이라면), 입력 이진수 값(5)과 동일하게 설정되며, 현재의 이진수에 대한 두 개의 가능한 이진수 값 중 하나에 대한 확률에 대한 측정치가 특정 임계치 이하라면(또는 그 미만이라면), 출력 이진수 값(7)은 변경된다(즉, 이 값은 입력 이진수 값의 정반대로 설정된다). 실시예에서, 임계치는 두 가능한 이진수 값에 대한 추정된 확률에 대해 0.5의 값에 대응한다.

[0073] 이진수 버퍼 선택기(6)는, 입력 이진수 값(5)과,두 개의 가능한 이진수 값 중 어떤 값이 현재의 이진수에 대한 더 작은 확률이나 더 큰 확률의 이진수 값을 나타내는지에 대한 추정을 명시하는 관련 식별자를 기반으로 하여 출력 이진수 값(7)을 결정할 수 있다. 실시예에서, 출력 이진수 값(7)은, 식별자가, 두 개의 가능한 이진수 값 중 제1 값이 현재의 이진수에 대한 더 작은 확률(또는 더 큰 확률)의 이진수 값을 나타냄을 명시한다면 입력 이진수 값(5)과 동일하게 설정되며, 식별자가 두 개의 가능한 이진수 값 중 제2 값이 현재의 이진수에 대한 더 작은 확률(또는 더 큰 확률)의 이진수 값을 나타냄을 명시한다면, 출력 이진수 값(7)은 변경된다(즉, 이 값은 입력 이진수 값의 정반대로 설정된다).

[0074] 이진수 버퍼 선택기(6)는, 출력 이진수(7)가 전송될 이진수 버퍼(8)를, 현재의 이진수에 대한 두 개의 가능한 이진수 값 중 하나에 대한 확률의 추정에 대한 관련 측정치를 기반으로 하여 결정할 수 있다. 실시예에서, 두 개의 가능한 이진수 값 중 하나에 대한 확률의 추정에 대한 측정치에 대한 가능한 값의 세트는 유한하며, 이진수 버퍼 선택기(6)는 정확히 하나의 이진수 버퍼(8)를 두 개의 가능한 이진수 값 중 하나에 대한 확률의 추정에 대한 각각의 가능한 값과 관련시키는 표를 포함하며, 두 개의 가능한 이진수 값 중 하나에 대한 확률의 추정에 대한 측정치에 대한 상이한 값은 동일한 이진수 버퍼(8)와 관련될 수 있다. 다른 실시예에서, 두 개의 가능한 이진수 값 중 하나에 대한 확률의 추정에 대한 측정치에 대한 가능한 값의 범위는 다수의 간격으로 분할되며, 이진수 버퍼 선택기(6)는, 두 개의 가능한 이진수 값 중 하나에 대한 확률의 추정에 대한 현재의 측정치에 대한 간격 인덱스를 결정하고, 이진수 버퍼 선택기(6)는 정확히 하나의 이진수 버퍼(8)를 간격 인덱스에 대한 각 가능한 값과 관련시키는 표를 포함하며, 간격 인덱스에 대한 상이한 값은 동일한 이진수 버퍼(8)와 관련될 수 있다. 바람직한 실시예에서, 두 개의 가능한 이진수 값 중 하나에 대한 확률의 추정에 대한 정반대 측정치를 갖는 입력 이진수(5)(정반대 측정치는 확률 추정(P 및 1-P)을 나타내는 측정치임)가 동일한 이진수 버퍼(8)에 공급된다. 다른 바람직한 실시예에서, 현재의 이진수에 대한 두 개의 가능한 이진수 값 중 하나에 대한 확률의 추정에 대한 측정치의 특정 이진수 버퍼와의 관련은 만들어진 부분 비트스트림이 유사한 비트 레이트를 가짐을 보장하기 위해 예컨대 시간에 따라 적응된다.

[0075] 이진수 버퍼 선택기(6)는 출력 이진수(7)가 전송될 이진수 버퍼(8)를, 현재의 이진수에 대한 더 작은 확률이나 더 큰 확률의 이진수 값에 대한 확률의 추정에 대한 관련 측정치를 기반으로 하여 결정할 수 있다. 바람직한 실시예에서, 더 작은 확률이나 더 큰 확률의 이진수 값에 대한 확률의 추정에 대한 관련 측정치에 대한 가능한 값의 세트는 유한하며, 이진수 버퍼 선택기(6)는 정확히 하나의 이진수 버퍼(8)를 더 작은 확률이나 더 큰 확률의 이진수 값에 대한 확률의 추정의 각 가능한 값과 관련시키는 표를 포함하며, 더 작은 확률이나 더 큰 확률의 이진수 값에 대한 확률의 추정에 대한 측정치에 대한 상이한 값은 동일한 이진수 버퍼(8)와 관련될 수 있다. 다른 실시예에서, 더 작은 확률이나 더 큰 확률의 이진수 값에 대한 확률의 추정에 대한 측정치에 대한 가능한 값의 범위는 다수의 간격으로 분할되며, 이진수 버퍼 선택기(6)는, 더 작은 확률이나 더 큰 확률의 이진수 값에 대한 확률의 추정에 대한 현재의 측정치에 대한 간격 인덱스를 결정하고, 이진수 버퍼 선택기(6)는 정확히 하나의 이진수 버퍼(8)를 간격 인덱스에 대한 각 가능한 값과 관련시키는 표를 포함하며, 간격 인덱스에 대한 상이한 값은 동일한 이진수 버퍼(8)와 관련될 수 있다. 바람직한 실시예에서, 현재의 이진수에 대한 더 작은 확률이나 더 큰 확률의 이진수 값에 대한 확률의 추정에 대한 측정치의 특정 이진수 버퍼와의 관련은 만들어진 부분 비트스트림이 유사한 비트 레이트를 가짐을 보장하기 위해 예컨대 시간에 따라 적응된다.

[0076] 둘 이상의 이진수 버퍼(8) 각각은 정확히 하나의 이진수 인코더(10)와 연결되며, 각 이진수 인코더는 하나의 이진수 버퍼(8)와만 연결된다. 각 이진수 인코더(10)는 관련된 이진수 버퍼(8)로부터 이진수를 판독하고, 이진수(9)의 시퀀스를, 비트 시퀀스를 나타내는 코드워드(11)로 전환한다. 이진수 버퍼(8)는 선입선출 버퍼를 나타내며, 따라서 (순차적인 순서에서) 이진수 버퍼(8)에 나중에 공급된 비트가 (순차적인 순서에서) 이진수 버퍼에 더 일찍 공급된 이진수 이전에 인코딩되지 않는다. 특정한 이진수 인코더(10)의 출력인 코드워드(11)는 특정한 부분 비트스트림(12)에 기록된다. 전체 인코딩 알고리즘은 소스 심벌(1)을 둘 이상의 부분 비트스트림(12)으로 전환하며, 부분 비트스트림의 수는 이진수 버퍼 및 이진수 인코더의 수와 같다. 바람직한 실시예에서, 이진수 인코더(10)는 가변 수의 이진수(9)를 가변 수의 비트의 코드워드(11)로 전환한다. 상기 및 이후에 개괄적으로 기재된(될) PIPE 인코딩 실시예의 한 가지 장점은, 이진수 인코딩이 (예컨대, 상이한 확률 측정치의 그룹마다) 병렬로 행해질 수 있어서, 이점이 여러 구현에 대한 처리 시간을 줄인다는 점이다.

[0077] PIPE 코딩의 다른 장점은, 이진수 인코더(10)에 의해 행해진 이진수 인코딩이 상이한 파라미터 세트(5)마다 상세하게 설계될 수 있다는 점이다. 특히, 이진수 인코딩 및 인코딩은 상이한 추정된 확률 그룹마다 (코딩 효율 및/또는 복잡도 면에서) 최적화될 수 있다. 한편으론, 이로 인해, 유사한 코딩 효율을 갖는 산술 코딩 알고리즘에 비해 인코딩/디코딩 복잡도를 감소시킬 수 있다. 다른 한편으론, 이로 인해, 유사한 인코딩/디코딩 복잡도를

갖는 VLC 코딩 알고리즘에 비해 코딩 효율을 개선시킬 수 있다. 실시예에서, 이진수 인코더(10)는, 현재의 이진수에 대한 두 개의 가능한 이진수 값(5) 중 하나에 대한 확률의 추정에 대한 측정치의 상이한 그룹마다 상이한 인코딩 알고리즘(즉, 이진수 시퀀스의 코드워드로의 매핑)을 구현한다. 다른 실시예에서, 이진수 인코더(10)는 현재의 이진수에 대한 더 작은 확률 또는 더 큰 확률의 이진수 값에 대한 확률의 추정에 대한 측정치의 상이한 그룹마다 상이한 인코딩 알고리즘을 구현한다. 다른 실시예에서, 이진수 인코더(10)는 상이한 암호화 방식마다 상이한 인코딩 알고리즘을 구현한다. 다른 실시예에서, 이진수 인코더(10)는 현재의 이진수에 대한 두 개의 가능한 이진수 값(5) 중 하나에 대한 확률의 추정에 대한 측정치 그룹과 채널 보호 코드의 상이한 조합마다 상이한 인코딩 알고리즘을 구현한다. 다른 실시예에서, 이진수 인코더(10)는 현재의 이진수에 대한 더 작은 확률이나 더 큰 확률의 이진수 값(5)에 대한 확률의 추정에 대한 측정치의 그룹 및 채널 보호 코드의 상이한 조합마다 상이한 인코딩 알고리즘을 구현한다. 다른 실시예에서, 이진수 인코더(10)는 현재의 이진수에 대한 두 개의 가능한 이진수 값(5) 중 하나에 대한 확률의 추정에 대한 측정치의 그룹 및 암호화 방식의 상이한 조합마다 상이한 인코딩 알고리즘을 구현한다. 다른 바람직한 실시예에서, 이진수 인코더(10)는 현재의 이진수에 대한 더 작은 확률이나 더 큰 확률의 이진수 값(5)에 대한 확률의 추정에 대한 측정치 그룹 및 암호화 방식의 상이한 조합마다 상이한 인코딩 알고리즘을 구현한다.

[0078] 이진수 인코더(10)-또는 이진수 인코더 중 하나 이상-는 이진 산술 인코딩 엔진을 나타낼 수 있다. 다른 실시예에서, 이진수 인코더 중 하나 이상은 이진 산술 코딩 엔진을 나타내며, 소정의 이진수 버퍼의 대표적인 LPS/LPB 확률(P_{LPS})로부터 대응한 코드 간격 폭(R_{LPS})으로의 매핑-즉, 예컨대 코드 간격의 하한을 식별하는 현재의 간격 오프셋(L)과 현재의 간격 폭(R)에 의해 한정되는 이진 산술 코딩 엔진의 내부 상태의 간격 세분-은 표 룩업을 사용하여 실현된다. 다른 바람직한 실시예에서, 소정의 이진수 버퍼에 관련된 각 표-기반 이진 산술 코딩 엔진의 경우, K개의 대표적인 간격 폭 값$\{Q_0,...,Q_{K-1}\}$이 R_{LPS}를 표현하기 위해 사용되며, K 및 대표적인 간격 폭 값 $\{Q_0,...,Q_{K-1}\}$의 선택은 이진수 버퍼에 의존한다. K>1을 선택한 경우, 이진수의 산술 인코딩은 현재의 간격 폭(R)을 {0,...,K-1}에서의 값을 갖는 양자화 인덱스(q)에 매핑하는 세부 단계와, 룩업 표로부터의 대응하는 부분 간격 폭 값(Q_q)에 인덱스로서 q를 사용하여 접근함으로써 내부 세분을 실행하는 세부 단계를 수반할 수 있다. K=1을 선택한 경우, 즉 단 하나의 대표적인 간격 폭 값(Q_0)이 주어진 경우에, 이 값(Q_0)은 단일 재정규화(renormalization) 사이클 내에서 대응하는 이진수 버퍼에 진입한 복수의 MPS/MPB 값의 디코딩을 허용하기 위해 그의 파워로서 선택될 수 있다. 각 산술 코딩 엔진의 결과적인 코드워드는 개별적으로 송신, 패킷화 또는 저장될 수 있거나, 후술될 바와 같이 송신 또는 저장용으로 인터리빙될 수 있다.

[0079] 즉, 이진 산술 코딩 엔진(10)은 이진수를 그 이진수 버퍼(8)에서 코딩할 때 다음의 단계를 실행할 수 있다:

[0080] 1. 이진수 버퍼로부터 valLPS를 수신함(여기서 고려된 각각의 이진 산술 코딩 엔진(10)은 "이진수"를 수신하도록 선택되었으며(또는, 다시 말해 "이진수"가 각각의 이진 산술 코딩 엔진(10)과 관련되었으며), 그 이유는 p_state[bin]과 같은 확률 분포 추정이 이 이진 산술 코딩 엔진(10)과 관련되었기 때문임을 상기한다)

[0081] 2. R의 양자화:

[0082] q_index=Qtab[R>>q](또는 일부 다른 형태의 양자화)

[0083] 3. R_{LPS} 및 R의 결정:

[0085] *R_{LPS}=Rtab[q_index](p_state가 여기서 언급되지 않으며, 이는 고려된 이진 산술 코딩 엔진(10)에 대해 이것이 정해져 있기 때문이며, 즉 p_state[encoder], Rtab은 여기서 p[p_state[encoder]·Q[q_index]에 대해 미리 계산된 값을 저장하였음을 주목해야 한다)

[0086] R=R-R_{LPS}[즉, R은 "이진수"가 MPS였던것처럼 미리 업데이트된다]

[0087] 4. 새로운 부분 간격의 계산:

[0088] 만약 (이진수=1-ValMPS)라면,

[0089] L← L+R

[0090] R←R_{LPS}

[0091] 5. L 및 R의 재정규화, 비트를 기록함,

[0092] 여기서 q_index는 Qtab으로부터 판독된 양자화 값의 인덱스를 기재하고,

[0093] p_state는 (이진 산술 코딩 엔진(10)에 대해 고정된) 현재의 상태를 기재하고,

[0094] R_{LPS}는 LPS에 대응하는 간격 폭을 기재하며,

[0095] valMPS는 MPS에 대응하는 비트의 값을 기재한다.

[0096] 따라서, 이진 산술 디코딩 엔진(22)은 이진수 버퍼(20)로 출력되는 이진수를 디코딩할 때 다음의 단계를 실행할 수 있다.

[0097] 1. 이진수에 대한 요청을 수신함(여기서 고려된 각각의 이진 산술 디코딩 엔진(22)은 "이진수"를 디코딩하도록 선택되었으며(또는, 다시 말해 "이진수"가 각각의 이진 산술 인코딩 엔진(22)과 관련되었으며), 그 이유는 p_state[bin]과 같은 확률 분포 추정이 이 이진 산술 디코딩 엔진(22)과 관련되었기 때문임을 상기한다)

[0098] 2. R의 양자화:

[0099] q_index=Qtab[R>>q](또는 일부 다른 형태의 양자화)

[0100] 3. R_{LPS} 및 R의 결정:

[0101] R_{LPS}=Rtab[q_index](p_state가 여기서 언급되지 않으며, 이는 고려된 이진 산술 디코딩 엔진(22)에 대해 이것이 고정되어 있기 때문이며, 즉 p_state[encoder], Rtab은 여기서 p[p_state[encoder]·Q[q_index]에 대해 미리 계산된 값을 저장하였음을 주목해야 한다)

[0102] R=R-R_{LPS}[즉, R은 "이진수"가 MPS였던것처럼 미리 업데이트된다]

[0103] 4. 부분 간격의 위치에 따른 이진수의 결정:

[0104] 만약 (V≥R)라면,

[0105] 이진수←1-valMPS(이진수는 LPS로서 디코딩되고; 이진수 버퍼 선택기(18)는 이 이진수 정보 및 valMPS를 사용하여 실제 이진수 값을 얻을 것이다)

[0106] V← V-R

[0107] R←R_{LPS}

[0108] 그렇지 않다면

[0109] 이진수←valMPS(이진수는 MPS로서 디코딩되고; 이진수 버퍼 선택기(18)는 이 이진수 정보 및 valMPS를 사용하여 실제 이진수 값을 얻을 것이다)

[0110] 5. R의 재정규화, 1비트를 판독하고 V를 업데이트함,

[0111] q_index는 Qtab으로부터 판독된 양자화 값의 인덱스를 기재하고,

[0112] p_state는 (이진 산술 디코딩 엔진(22)에 대해 고정된) 현재의 상태를 기재하고,

[0113] R_{LPS}는 LPS에 대응하는 간격 폭을 기재하고,

[0114] valMPS는 MPS에 대응하는 비트의 값을 기재하며,

[0115] V는 현재의 부분 간격의 내부로부터의 값을 기재한다.

[0116] 이진수 인코더(10)-또는 이진수 인코더 중 하나 이상-는, 입력 이진수의 시퀀스(9)를 직접 코드워드(10)에 매핑하는 엔트로피 인코더를 나타낼 수 있다. 그러한 매핑은 효율적으로 구현될 수 있으며, 복잡한 산술 코딩 엔진을 필요로 하지 않는다. (디코더에서 행해졌던 것처럼) 코드워드의 이진수 시퀀스로의 역매핑은 입력 시퀀스의 완벽한 디코딩을 보장하기 위해 고유한 것이어야 하지만, 이진수 시퀀스(9)의 코드워드(10)로의 매핑은 반드시 고유할 필요는 없다, 즉 이진수의 특정한 시퀀스는 코드워드의 하나보다 많은 시퀀스에 매핑될 수 있다. 실시예에서, 입력 이진수 시퀀스(9)는 코드워드(10)에 바이젝티브 매핑된다. 다른 바람직한 실시예에서, 이진수 인코더(10)-또는 이진수 인코더 중 하나 이상-는 입력 이진수의 가변 길이 시퀀스(9)를 가변 길이 코드워드(10)에 직접 매핑하는 엔트로피 인코더를 나타낸다. 실시예에서, 출력 코드워드는 일반 허프만 코드나 정규(canonical)

허프만 코드와 같은 중복되지 않는 코드를 나타낸다.

[0117] 이진수 시퀀스의 중복되지 않는 코드로의 바이젝티브 매핑의 두 예가 표 3에 예시되어 있다. 다른 실시예에서, 출력 코드워드는 에러 검출 및 에러 복구에 적절한 중복 코드를 나타낸다. 다른 바람직한 실시예에서, 출력 코드워드는 소스 심벌을 암호화하는데 적절한 암호화 코드를 나타낸다.

표 3

[0118]

이진수 시퀀스 (이진수 순서는 왼쪽에서 오른쪽으로 임)	코드워드 (비트 순서는 왼쪽에서 오른쪽으로 임)
0000 0000	1
0000 0001	0000
0000 001	0001
0000 01	0010
0000 1	0011
0001	0100
001	0101
01	0110
1	0111
이진수 시퀀스 이진수 순서는 왼쪽에서 오른쪽으로 임)	코드워드 (비트 순서는 왼쪽에서 오른쪽으로 임)
000	10
01	11
001	010
11	011
1000 0	0001
1001	0010
1010	0011
1000 1	0000 0
1011	0000 1

[0119] 다른 실시예에서, 이진수 인코더(10)-또는 이진수 인코더 중 하나 이상-는 입력 이진수의 가변 길이 시퀀스(9)를 고정 길이 코드워드(10)에 직접 매핑하는 엔트로피 인코더를 나타낸다. 다른 실시예에서, 이진수 인코더(10)-또는 이진수 인코더 중 하나 이상-는 입력 이진수의 고정된 길이의 시퀀스(9)를 가변 길이 코드워드(10)에 직접 매핑하는 엔트로피 인코더를 나타낸다.

[0120] 실시예에 따른 PIPE 디코더를 도 4에 예시한다. 디코더는 도 3의 인코더의 역동작을 기본적으로 실행하여, (이전에 인코딩된) 소스 심벌(27)의 시퀀스를 둘 이상의 부분 비트스트림(24)의 세트로부터 디코딩한다. 디코더는 두 개의 상이한 처리 흐름을 포함한다: 인코더의 데이터 흐름을 복제하는 데이터 요청의 흐름과, 인코더 데이터 흐름의 역을 나타내는 데이터 흐름. 도 4의 예시에서, 점선 화살표는 데이터 요청 흐름을 나타내는 반면, 실선 화살표는 데이터 흐름을 나타낸다. 디코더의 빌딩 블록은 기본적으로 인코더의 빌딩 블록을 복제한 것이지만, 역 동작을 구현한다.

[0121] 소스 심벌의 디코딩은, 이진화기(14)에 전송된 새 디코딩된 소스 심벌(13)에 대한 요청에 의해 트리거된다. 실시예에서, 새 디코딩된 소스 심벌(13)에 대한 각 요청은 하나 이상의 카테고리 세트의 카테고리와 관련된다. 소스 심벌에 대한 요청과 관련된 카테고리는 인코딩 동안 대응하는 소스 심벌과 관련되었던 카테고리와 동일하다.

[0122] 이진화기(14)는 소스 심벌에 대한 요청(13)을 파라미터 할당기(16)에 전송된 이진수에 대한 하나 이상의 요청에 매핑한다. 이진화기(14)에 의해 파라미터 할당기(16)에 전송된 이진수에 대한 요청에 대한 최종 응답으로서, 이진화기(14)는 이진수 버퍼 선택기(18)로부터 디코딩된 이진수(26)를 수신한다. 이진화기(14)는 디코딩된 이진수(26)의 수신된 시퀀스를 요청된 소스 심벌에 대한 특정 이진화 방식의 이진수 시퀀스와 비교하고, 디코딩된 이진수(26)의 수신된 시퀀스가 소스 심벌의 이진화와 매칭한다면, 이진화기는 그 이진수 버퍼를 비우고, 디코딩된 소스 심벌을 새 디코딩된 심벌에 대한 요청에 대한 최종 응답으로서 출력한다. 디코딩된 이진수의 이미 수신된 시퀀스가 요청된 소스 심벌에 대한 이진화 방식에 대한 이진수 시퀀스 중 임의의 시퀀스와 매칭하지 않는다면, 이진화기는, 디코딩된 이진수의 시퀀스가 요청된 소스 심벌에 대한 이진화 방식의 이진수 시퀀스 중 하나와 매칭할 때까지 이진수에 대한 다른 요청을 파라미터 할당기에 전송한다. 소스 심벌에 대한 각 요청에 대해, 디코

더는 대응하는 소스 심벌을 인코딩하는데 사용되었던 동일한 이진화 방식을 사용한다. 이진화 방식은 상이한 소스 심벌 카테고리마다 상이할 수 있다. 특정한 소스 심벌 카테고리에 대한 이진화 방식은 특정한 카테고리에 대한 가능한 소스 심벌 값의 세트 및/또는 소스 심벌의 다른 속성에 의존할 수 있다.

[0123] 파라미터 할당기는 하나 이상의 파라미터의 세트를 이진수에 대한 각각의 요청에 할당하여, 관련된 파라미터 세트를 갖는 이진수에 대한 요청을 이진수 버퍼 선택기에 전송한다. 파라미터 할당기에 의해 요청된 이진수에 할당되는 파라미터 세트는, 인코딩 동안 대응하는 이진수에 할당되었던 것과 동일하다. 파라미터 세트는 인코더 기재에서 언급한 파라미터 중 하나 이상으로 구성될 수 있다.

[0124] 파라미터 할당기(16)는 이진수에 대한 각 요청을 현재의 요청된 이진수에 대한 두 개의 가능한 이진수 값 중 하나에 대한 확률의 추정에 대한 측정치와 관련시킬 수 있다. 특히, 파라미터 할당기(16)는 이진수에 대한 각 요청을, 현재의 요청된 이진수에 대한 더 작은 확률이나 더 큰 확률의 이진수 값에 대한 확률의 추정에 대한 측정치와, 두 개의 가능한 이진수 값 중 어떤 값이 현재의 요청된 이진수에 대한 더 작은 확률이나 더 큰 확률의 이진수 값을 나타내는지에 대한 추정을 명시하는 식별자와 관련시킬 수 있다.

[0125] 다른 실시예에서, 파라미터 할당기(16)는 이진수에 대한 각 요청(15 및 17) 을 현재의 요청된 이진수에 대한 두 개의 가능한 이진수 값 중 하나에 대한 확률의 추정에 대한 측정치 및 하나 이상의 다른 파라미터와 관련시킨다. 다른 바람직한 실시예에서, 파라미터 할당기(16)는 이진수에 대한 각 요청(15 및 17)을, 현재의 요청된 이진수에 대한 더 작은 확률이나 더 큰 확률의 이진수 값에 대한 확률의 추정에 대한 측정치와, 두 개의 가능한 이진수 값 중 어떤 값이 현재의 요청된 이진수에 대한 더 작은 확률이나 더 큰 확률의 이진수 값을 나타내는지에 대한 추정을 명시하는 식별자와, 하나 이상의 다른 파라미터(상기 게재한 파라미터 중 하나 이상을 포함할 수 있음)관련시킬 수 있다.

[0126] 실시예에서, 파라미터 할당기(16)는, 하나 이상의 이미 디코딩된 심벌의 세트를 기반으로 하여, 상기 언급한 확률 측정치(현재의 요청된 이진수에 대한 두 개의 가능한 이진수 값 중 하나에 대한 확률의 추정에 대한 측정치, 현재의 요청된 이진수에 대한 더 작은 확률이나 더 큰 확률의 이진수 값에 대한 확률의 추정에 대한 측정치, 및 두 개의 가능한 이진수 값 중 어떤 값이 현재의 요청된 이진수에 대한 더 작은 확률이나 더 큰 확률의 이진수 값을 나타내는지에 대한 추정을 명시하는 식별자) 중 하나 이상을 결정한다. 이진수에 대한 특정한 요청에 대한 확률 측정치의 결정은 대응하는 이진수에 대한 인코더에서의 처리를 복제한다. 확률 측정치를 결정하는데 사용되는 디코딩된 심벌은 동일한 심벌 카테고리의 하나 이상의 이미 디코딩된 심벌, (소스 심벌에 대한 현재의 요청과 관련된 데이터 세트에 관련하여) 이웃한 공간 및/또는 시간 위치의 (블록이나 샘플 그룹과 같은) 데이터 세트에 대응하는 동일한 심벌 카테고리의 하나 이상의 이미 디코딩된 심벌, 또는 (소스 심벌에 대한 현재의 요청과 관련된 데이터 세트에 관련하여) 동일한 및/또는 이웃한 공간 및/또는 시간 위치의 데이터 세트에 대응하는 상이한 심벌 카테고리의 하나 이상의 이미 디코딩된 심벌을 포함할 수 있다.

[0127] 파라미터 할당기(16)의 출력인 관련된 파라미터 세트를 갖는 이진수에 대한 각 요청(17)은 이진수 버퍼 선택기(18)에 공급된다. 관련된 파라미터 세트(17)를 기반으로 하여, 이진수 버퍼 선택기(18)는 이진수에 대한 요청(19)을 둘 이상의 이진수 버퍼(20) 중 하나에 전송하며, 선택된 이진수 버퍼(20)로부터 디코딩된 이진수(25)를 수신한다. 디코딩된 입력 이진수(25)는 잠재적으로 변경되어, 디코딩된 출력 이진수(26)-잠재적인 변경된 값을 가짐-가 관련된 파라미터 세트를 갖는 이진수에 대한 요청(17)에 대한 최종 응답으로서 이진화기(14)에 전송된다.

[0128] 이진수에 대한 요청이 전송되는 이진수 버퍼(20)는, 인코더 측에서 이진수 버퍼 선택기의 출력 이진수가 전송되었던 이진수 버퍼와 동일한 방식으로 선택된다.

[0129] 실시예에서, 이진수 버퍼 선택기(18)는, 이진수 버퍼에 대한 요청(19)이 전송된 이진수 버퍼(20)를 현재의 요청된 이진수에 대한 두 개의 가능한 이진수 값 중 하나에 대한 확률의 추정에 대한 관련 측정치를 기반으로 하여 결정한다. 실시예에서, 두 개의 가능한 이진수 값 중 하나에 대한 확률의 추정에 대한 측정치에 대한 가능한 값의 세트는 유한하며, 이진수 버퍼 선택기(18)는 정확히 하나의 이진수 버퍼(20)를 두 개의 가능한 이진수 값 중 하나에 대한 확률의 추정의 각 가능한 값과 관련시키는 표를 포함하며, 두 개의 가능한 이진수 값 중 하나에 대한 확률의 추정에 대한 측정치에 대한 상이한 값은 동일한 이진수 버퍼(20)와 관련될 수 있다. 다른 실시예에서, 두 개의 가능한 이진수 값 중 하나에 대한 확률의 추정에 대한 측정치에 대한 가능한 값의 범위는 다수의 간격으로 분할되며, 이진수 버퍼 선택기(18)는, 두 개의 가능한 이진수 값 중 하나에 대한 확률의 추정에 대한 현재의 측정치에 대한 간격 인덱스를 결정하고, 이진수 버퍼 선택기(18)는 정확히 하나의 이진수 버퍼(20)를 간격 인덱스에 대한 각 가능한 값과 관련시키는 표를 포함하며, 간격 인덱스에 대한 상이한 값은 동일한

이진수 버퍼(20)와 관련될 수 있다. 두 개의 가능한 이진수 값 중 하나에 대한 확률의 추정에 대한 정반대 측정치를 갖는 이진수에 대한 요청(17)은 동일한 이진수 버퍼(20)에 전송될 수 있다(정반대 측정치는 확률 추정(P 및 1-P)을 나타내는 측정치임). 또한, 현재의 이진수 요청에 대한 두 개의 가능한 이진수 값 중 하나에 대한 확률의 추정에 대한 측정치의 특정 이진수 버퍼와의 관련성은 시간에 따라 적응될 수 있다.

[0130] 이진수 버퍼 선택기(18)는, 이진수에 대한 요청(19)이 전송될 이진수 버퍼(20)를, 현재의 요청된 이진수에 대한 더 작은 확률이나 더 큰 확률의 이진수 값에 대한 확률의 추정에 대한 관련 측정치를 기반으로 하여 결정할 수 있다. 실시예에서, 더 작은 확률이나 더 큰 확률의 이진수 값에 대한 확률의 추정에 대한 측정치에 대한 가능한 값의 세트는 유한하며, 이진수 버퍼 선택기(18)는 정확히 하나의 이진수 버퍼(20)를 더 작은 확률이나 더 큰 확률의 이진수 값에 대한 확률의 추정의 각 가능한 값과 관련시키는 표를 포함하며, 더 작은 확률이나 더 큰 확률의 이진수 값에 대한 확률의 추정에 대한 측정치에 대한 상이한 값은 동일한 이진수 버퍼(20)와 관련될 수 있다. 실시예에서, 더 작은 확률이나 더 큰 확률의 이진수 값에 대한 확률의 추정에 대한 측정값에 대한 가능한 값의 범위는 다수의 간격으로 분할되며, 이진수 버퍼 선택기(18)는, 더 작은 확률이나 더 큰 확률의 이진수 값에 대한 확률의 추정에 대한 현재의 측정치에 대한 간격 인덱스를 결정하고, 이진수 버퍼 선택기(18)는 정확히 하나의 이진수 버퍼(20)를 간격 인덱스에 대한 각 가능한 값과 관련시키는 표를 포함하며, 간격 인덱스에 대한 상이한 값은 동일한 이진수 버퍼(20)와 관련될 수 있다. 다른 바람직한 실시예에서, 현재의 이진수 요청에 대한 더 작은 확률이나 더 큰 확률의 이진수 값에 대한 확률의 추정에 대한 측정치의 특정 이진수 버퍼와의 관련성은 시간에 따라 적응된다.

[0131] 선택된 이진수 버퍼(20)로부터 디코딩된 이진수(25)를 수신한 후, 이진수 버퍼 선택기(18)는 입력 이진수(25)를 잠재적으로 변경하여, 출력 이진수(26)-잠재적으로 변경된 값을 가짐-를 이진화기(14)에 전송한다. 이진수 버퍼 선택기(18)의 입출력 이진수 매핑은 인코더 측에서의 이진수 버퍼 선택기의 입출력 이진수 매핑의 역이다.

[0132] 이진수 버퍼 선택기(18)는 이진수의 값을 변경하지 않도록 구성될 수 있다, 즉 출력 이진수(26)는 항상 입력 이진수(25)와 동일한 값을 갖는다.

[0133] 이진수 버퍼 선택기(18)는 입력 이진수 값(25)과, 이진수에 대한 요청(17)과 관련된 현재의 요청된 이진수에 대한 두 개의 가능한 이진수 값 중 하나에 대한 확률의 추정에 대한 측정치를 기반으로 하여 출력 이진수 값(26)을 결정할 수 있다. 실시예에서, 출력 이진수 값(26)은, 현재의 이진수 요청에 대한 두 개의 가능한 이진수 값 중 하나에 대한 확률에 대한 측정치가 특정한 임계치 미만이라면(또는 그 이하라면), 입력 이진수 값(25)과 동일하게 설정되고, 현재의 이진수 요청에 대한 두 개의 가능한 이진수 값 중 하나에 대한 확률에 대한 측정치가 특정한 임계치 이상이라면(또는 그것을 초과한다면), 출력 이진수 값(26)은 변경된다(즉, 이 값은 입력 이진수 값의 반대로 설정된다). 다른 실시예에서, 출력 이진수 값(26)은, 현재의 이진수 요청에 대한 두 개의 가능한 이진수 값 중 하나에 대한 확률에 대한 측정치가 특정 임계치를 초과한다면(또는 그 이상이라면), 입력 이진수 값(25)과 동일하게 설정되며, 현재의 이진수 요청에 대한 두 개의 가능한 이진수 값 중 하나에 대한 확률에 대한 측정치가 특정 임계치 이하라면(또는 그 미만이라면), 출력 이진수 값(26)은 변경된다(즉, 이 값은 입력 이진수 값의 정반대로 설정된다). 바람직한 실시예에서, 임계치는 두 가능한 이진수 값에 대한 추정된 확률에 대해 0.5의 값에 대응한다.

[0134] 다른 실시예에서, 이진수 버퍼 선택기(18)는, 입력 이진수 값(25)과, 두 개의 가능한 이진수 값 중 어떤 값이 이진수에 대한 요청(17)과 관련된, 현재의 이진수 요청에 대한 더 작은 확률이나 더 큰 확률의 이진수 값을 나타내는지에 대한 추정을 명시하는 식별자를 기반으로 하여 출력 이진수 값(26)을 결정할 수 있다. 바람직한 실시예에서, 출력 이진수 값(26)은, 식별자가, 두 개의 가능한 이진수 값 중 제1 값이 현재의 이진수에 대한 더 작은 확률(또는 더 큰 확률)의 이진수 값을 나타냄을 명시한다면 입력 이진수 값(25)과 동일하게 설정되며, 식별자가 두 개의 가능한 이진수 값 중 제2 값이 현재의 이진수 요청에 대한 더 작은 확률(또는 더 큰 확률)의 이진수 값을 나타냄을 명시한다면, 출력 이진수 값(26)은 변경된다(즉, 이 값은 입력 이진수 값의 정반대로 설정된다).

[0135] 전술한 바와 같이, 이진수 버퍼 선택기는 이진수에 대한 요청(19)을 두 개 이상의 이진수 버퍼(20) 중 하나에 전송한다. 이진수 버퍼(20)는 선입선출 버퍼를 나타내며, 이 버퍼에는 연결된 이진수 디코더(22)로부터 디코딩된 이진수 시퀀스(21)가 공급된다. 이진수 버퍼 선택기(18)로부터 이진수 버퍼(20)에 전송된 이진수에 대한 요청(19)에 대한 응답으로, 이진수 버퍼(20)는 이진수 버퍼(20)에 먼저 공급되었던 그 내용의 이진수를 제거하고, 그것을 이진수 버퍼 선택기(18)에 전송한다. 이진수 버퍼(20)에 더 일찍 전송된 이진수는 더 일찍 제거되고 이진수 버퍼 선택기(18)에 전송된다.

[0136] 두 개 이상의 이진수 버퍼(20) 각각은 정확히 하나의 이진수 디코더(22)와 연결되며, 각 이진수 디코더는 하나의 이진수 버퍼(20)와만 연결된다. 각 이진수 디코더(22)는 비트 시퀀스를 나타내는 코드워드(23)를 별도의 부분 비트스트림(24)으로부터 판독한다. 이진수 디코더는 코드워드(23)를, 연결된 이진수 버퍼(20)에 전송되는 이진수 시퀀스(21)로 전환한다. 전체 디코딩 알고리즘은 둘 이상의 부분 비트스트림(24)을 다수의 디코딩된 소스 심벌로 전환하며, 부분 비트스트림의 수는 이진수 버퍼 및 이진수 디코더의 수와 동일하고, 소스 심벌의 디코딩은 새로운 소스 심벌에 대한 요청에 의해 트리거된다. 실시예에서, 이진수 디코더(22)는 가변 수의 비트의 코드워드(23)를 가변 수의 이진수의 시퀀스(21)로 전환한다. 상기 PIPE 실시예의 한 가지 장점은, 둘 이상의 부분 비트스트림으로부터의 이진수의 디코딩이 (예컨대, 상이한 그룹의 확률 측정치마다) 병렬로 행해질 수 있고, 이 점이 여러 구현에 대한 처리 시간을 줄인다는 점이다.

[0137] 상기 PIPE 디코딩 실시예의 다른 장점은 이진수 디코더(22)에 의해 행해진 이진수 디코딩이 상세하게는 상이한 세트의 파라미터(17)마다 설계될 수 있다는 점이다. 특히, 이진수 인코딩 및 인코딩은 상이한 추정된 확률 그룹마다 (코딩 효율 및/또는 복잡도 면에서) 최적화될 수 있다. 한편으론, 이로 인해, 유사한 코딩 효율을 갖는 산술 코딩 알고리즘에 비해 인코딩/디코딩 복잡도를 감소시킬 수 있다. 다른 한편으론, 이로 인해, 유사한 인코딩/디코딩 복잡도를 갖는 VLC 코딩 알고리즘에 비해 코딩 효율을 개선시킬 수 있다. 실시예에서, 이진수 디코더(22)는, 현재의 이진수 요청에 대한 두 개의 가능한 이진수 값(17) 중 하나에 대한 확률의 추정에 대한 측정치의 상이한 그룹마다 상이한 디코딩 알고리즘(즉, 비트 시퀀스의 코드워드로의 매핑)을 구현한다. 다른 실시예에서, 이진 디코더(22)는 현재의 요청된 이진수에 대한 더 작은 확률 또는 더 큰 확률의 이진수 값에 대한 확률의 추정에 대한 상이한 그룹의 측정치마다 상이한 디코딩 알고리즘을 구현한다. 다른 실시예에서, 이진수 디코더(22)는 상이한 채널 보호 코드마다 상이한 디코딩 알고리즘을 구현한다. 다른 실시예에서, 이진수 디코더(22)는 상이한 암호화 방식마다 상이한 디코딩 알고리즘을 구현한다. 다른 실시예에서, 이진수 디코더(22)는 현재의 요청된 이진수에 대한 두 개의 가능한 이진수 값(17) 중 하나에 대한 확률의 추정에 대한 측정치 그룹과 채널 보호 코드의 상이한 조합마다 상이한 디코딩 알고리즘을 구현한다. 다른 실시예에서, 이진수 디코더(22)는 현재의 요청된 이진수에 대한 더 작은 확률이나 더 큰 확률의 이진수 값(17)에 대한 확률의 추정에 대한 측정치의 그룹 및 채널 보호 코드의 상이한 조합마다 상이한 디코딩 알고리즘을 구현한다. 다른 실시예에서, 이진수 디코더(22)는 현재의 요청된 이진수에 대한 두 개의 가능한 이진수 값(17) 중 하나에 대한 확률의 추정에 대한 측정치의 그룹 및 암호화 방식의 상이한 조합마다 상이한 디코딩 알고리즘을 구현한다. 다른 바람직한 실시예에서, 이진수 디코더(22)는 현재의 요청된 이진수에 대한 더 작은 확률이나 더 큰 확률의 이진수 값(17)에 대한 확률의 추정에 대한 측정치 그룹 및 암호화 방식의 상이한 조합마다 상이한 디코딩 알고리즘을 구현한다.

[0138] 이진수 디코더(22)는 인코더 측에서 대응하는 이진수 인코더의 역 매핑을 행한다.

[0139] 실시예에서, 이진수 디코더(22)-또는 하나 이상의 이진수 디코더-는 이진 산술 디코딩 엔진을 나타낸다.

[0140] 다른 실시예에서, 이진수 디코더(22)-또는 하나 이상의 이진수 디코더-는 코드워드(23)를 이진수의 시퀀스(21)에 직접 매핑하는 엔트로피 디코더를 나타낸다. 그러한 매핑은 효율적으로 구현될 수 있으며, 복잡한 산술 코딩 엔진을 필요로 하지 않는다. 코드워드의 이진수 시퀀스로의 매핑은 고유한 것이어야 한다. 실시예에서, 코드워드(23)는 이진수 시퀀스(21)에 바이젝티브 매핑된다. 다른 실시예에서, 이진수 디코더(10)-또는 하나 이상의 이진수 디코더-는 가변 길이 코드워드(23)를 이진수의 가변 길이 시퀀스(21)에 직접 매핑하는 엔트로피 디코더를 나타낸다. 실시예에서, 입력 코드워드는 일반 허프만 코드나 정규 허프만 코드와 같은 중복되지 않는 코드를 나타낸다. 중복되지 않는 코드의 이진수 시퀀스로의 바이젝티브 매핑의 두 예가 표 3에 예시되어 있다. 다른 실시예에서, 입력 코드워드는 에러 검출 및 에러 복구에 적절한 중복 코드를 나타낸다. 다른 실시예에서, 입력 코드워드는 암호화 코드를 나타낸다.

[0141] 이진수 디코더(22)-또는 하나 이상의 이진수 디코더-는, 고정 길이 코드워드(23)를 가변 길이의 이진수 시퀀스(21)로 직접 매핑할 수 있는 엔트로피 디코더를 나타낼 수 있다. 택일적으로, 이진수 디코더(22)-또는 하나 이상의 이진수 디코더-는 가변 길이 코드워드(23)를 고정 길이 이진수 시퀀스(21)로 직접 매핑하는 엔트로피 디코더를 나타낸다.

[0142] 따라서, 도 3 및 도 4는 소스 심벌 시퀀스(1)를 인코딩하는 PIPE 인코더와 이 시퀀스를 재구성하는 PIPE 디코더에 대한 실시예를 도시했다. 즉, 도 3의 PIPE 인코더는 심벌라이저(122)로서 동작하는 이진화기(2), 할당기(114)로서 동작하는 파라미터 할당기(4), 선택기(120)로서 동작하는 이진수 버퍼 선택기(6), 도 1a에서 비트스트림(118)에 대응하는 비트스트림(12)을 각각 출력하는 각각의 엔트로피 인코더(116)로서 동작하는 직렬로 연결된 이진수 버퍼(8)와 이진수 인코더(10)의 쌍을 갖는 도 1a의 PIPE 인코더(104)로서 사용될 수 있다. 도 3 및

도 1의 비교로부터 명백해지는 바와 같이, 도 1a의 할당기(114)는 심벌라이저(122)의 출력측보다는 그 입력측에 택일적으로 연결될 수 있는 입력을 가질 수 있다. 유사하게, 도 4의 PIPE 디코더는, 도 2에서 비트스트림(216)에 대응하는 부분 비트스트림(24), 개별 엔트로피 디코더(210)에 대응하는 직렬로 연결된 버퍼(20)와 이진수 디코더(22)의 쌍, 선택기(214)로서 동작하는 이진수 버퍼 선택기(18), 할당기(212)로서 동작하는 파라미터 할당기(16) 및 디심벌라이저(222)로서 동작하는 이진화기(14)를 갖는 도 2a에서의 PIPE 디코더(202)로서 동작할 수 있다. 다시, 도 2a와 도 4 사이의 비교를 통해, 디심벌라이저(222), 할당기(212) 및 선택기(214) 사이의 상호연결이 상이하게 구성될 수 있어서, 택일적인 실시예에 따르면, 도 2a의 연결이 도 4에 도시한 연결에 대응하도록 수정되게 됨이 분명해 진다.

[0143] 도 3의 PIPE 인코더는, 알파벳 심벌 시퀀스(3)의 각 알파벳 심벌에 다수의 파라미터(5)를 할당하도록 구성된 할당기(4)를 포함한다. 이 할당은, 현재의 알파벳 심벌이 속해 있고, 신택스 요소(1)의 신택스 구조에 따라 현재 예상되는 표현법-이진화와 같은-으로의 신택스 요소(1)의 분류와 같은 알파벳 심벌 시퀀스의 이전 알파벳 심벌 내에 포함된 정보를 기반으로 하며, 그러한 예상은 이제 이전 신택스 요소(1) 및 알파벳 심벌(3)의 이력으로부터 유도 가능하다. 또한, 인코더는 각각의 엔트로피 인코더에 전송된 알파벳 심벌(3)을 각각의 비트스트림(12)으로 전환하도록 각각 구성되는 복수의 엔트로피 인코더(10)와, 각 알파벳 심벌(3)을 복수의 엔트로피 인코더(10) 중 선택된 하나에 전송하도록 구성된 선택기(6)를 포함하며, 그러한 선택은 각각의 알파벳 심벌(3)에 할당된 다수의 파라미터(5)에 의존한다. 도 4의 PIPE 디코더는 각각의 비트스트림(23)을 알파벳 심벌(21)로 전환하도록 각각 구성되는 복수의 엔트로피 디코더(22), 알파벳 심벌 시퀀스의 이전에 재구성된 알파벳 심벌(도 4의 26 및 27 참조) 내에 포함된 정보를 기반으로 하여 재구성될 알파벳 심벌 시퀀스의 각 알파벳 심벌(15)에 다수의 파라미터(17)를 할당하도록 구성된 할당기(16), 및 복수의 엔트로피 디코더(22) 중 선택된 하나로부터 재구성될 알파벳 심벌 시퀀스의 각 알파벳 심벌을 회수하도록 구성되는 선택기(18)를 포함하며, 그러한 선택은 각각의 알파벳 심벌에 한정된 다수의 파라미터에 의존한다. 할당기(16)는, 각 알파벳 심벌에 할당된 다수의 파라미터가, 각각의 알파벳 심벌이 가질 수 있는 가능한 알파벳 심벌 값 사이의 분포 확률의 추정에 대한 측정치를 포함하거나 그러한 측정치가 되도록 구성될 수 있다. 재구성될 알파벳 심벌 시퀀스는 이진 알파벳으로 되어 있을 수 있고, 할당기(16)는, 확률 분포의 추정이, 이진 알파벳의 두 개의 가능한 이진수 값 중 더 작은 확률이나 더 큰 확률의 이진수 값의 확률의 추정에 대한 측정치와, 두 가능한 이진수 값 중 어떤 것이 더 작은 확률이나 더 큰 확률의 이진수 값을 나타내는지를 명시하는 식별자로 구성되도록 구성될 수 있다. 할당기(16)는, 컨텍스트를, 관련된 각각의 확률 분포 추정을 갖는 각 컨텍스트로 재구성될 알파벳 심벌 시퀀스의 이전에 재구성된 알파벳 심벌 내에 포함된 정보를 기반으로 하여 재구성될 알파벳 심벌 시퀀스(15)의 각 알파벳 심벌에 내부적으로 할당하고, 각각의 컨텍스트가 할당된 이전에 재구성된 알파벳 심벌의 심벌 값을 기반으로 하여 각 컨텍스트에 대한 확률 분포 추정을 실제 심벌 통계에 적응시키도록 더 구성될 수 있다. 컨텍스트는, 비디오 또는 화상 코딩에서나, 심지어는 금융 응용의 경우 표에서와 같이 신택스 요소가 속하는 위치의 공간 관계나 이웃을 고려할 수 있다. 그 후, 각 알파벳 심벌에 대한 확률 분포의 추정에 대한 측정치가, 확률 분포의 추정에 대한 측정치를 얻기 위해, 복수의 확률 분포 추정 대표치 중 하나로의 각각의 알파벳 심벌로 할당된 컨텍스트와 관련된 확률 분포 추정을 양자화함으로써와 같이 각각의 알파벳 심벌에 할당된 콘텍스트와 관련된 확률 분포 추정을 기반으로 하여 결정될 수 있다. 선택기는, 서젝티브(surjective) 관련성은 복수의 엔트로피 인코더와 복수의 확률 분포 추정 대표값 사이에 한정되도록, 즉 각 엔트로피 인코더가 관련된 적어도 하나의 확률 분포 추정 대표값을 갖지만, 하나보다 많은 확률 분포 추정 대표값이 하나의 엔트로피 인코더와 관련될 수 있도록 구성될 수 있다. 실시예에 따라, 관련은 심지어 바이젝티브(bijective)일 수 있다. 선택기(18)는, 시간이 지남에 따라, 알파벳 심벌 시퀀스의 이전에 재구성된 알파벳 심벌에 따라 미리 결정된 결정적인 방식으로 양자화 매핑을 확률 분포 추정 범위로부터 복수의 확률 분포 추정 대표값으로 변경하도록 구성될 수 있다. 즉, 선택기(18)는 양자화 간격 크기, 즉 이제 개별 엔트로피 디코더와 서젝티브하게 관련될 수 있는 개별 확률 인덱스로 매핑된 확률 분포의 간격을 변경할 수 있다. 이제, 복수의 엔트로피 디코더(22)는 양자화 매핑의 변경에 응답하여 알파벳 심벌을 비트스트림으로 전환하는 그 방식을 적응시키도록 구성될 수 있다. 예컨대, 각 엔트로피 디코더(22)는 최적화될 수 있다, 즉 각각의 확률 분포 추정 양자화 간격 내의 특정한 확률 분포 추정에 대해 최적 압축율을 가질 수 있으며, 각각의 확률 분포 추정 양자화 간격을 최적화되도록 변경하면 이 간격 내에서 이 특정한 확률 분포 추정의 위치를 적응시키기 위해 그 코드워드/심벌 시퀀스 매핑을 변경할 수 있다. 선택기는, 알파벳 심벌이 복수의 엔트로피 디코더로부터 회수될 율이 덜 분산되게 되도록 양자화 매핑을 변경하도록 구성될 수 있다. 이진화기(14)에 관해, 신택스 요소가 이미 이진수라면, 이것은 사라질 수 있음을 주의해야 한다. 또한, 디코더(22)의 타입에 따라, 버퍼(20)의 존재는 필요치 않다. 또한, 버퍼는 디코더 내에 통합될 수 있다.

[0144] 지금까지, 도 1a 및 도 2에서 PIPE 인코더(104)와 PIPE 디코더(202)에 대한 더 상세한 실시예를, 도 1a 및 도 2

의 장치에서 쉽게 구현된다면 VLC 및 PIPE 부분 비트스트림이 병렬로 전달되는 병렬 비트스트림 출력을 야기하는 도 3 및 도 4를 참조하여, 앞서 기재하였다. 다음의 설명에서, 존재하고, 그 후, 병렬로 VLC 비트스트림을 따라 송신될 PIPE 부분 비트스트림을 결합하는 방법이나, 두 비트스트림, 즉 VLC 비트스트림 및 인터리빙된 PIPE 비트스트림을 2차로 인터리빙함에 대한 가능성을 기재한다.

[0145] **유한 소스 심벌 시퀀스의 종료**

[0146] PIPE 인코더 및 디코더에 대한 실시예에서, 소스 심벌의 유한 세트에 대해 인코딩 및 디코딩이 행해진다. 종종, 정지영상, 비디오 시퀀스의 프레임 또는 필드, 이미지 슬라이스, 비디오 시퀀스의 프레임 또는 필드의 슬라이스, 또는 연속적인 오디오 샘플 세트 등과 같은 특정 양의 데이터를 코딩한다. 소스 심벌의 유한 세트의 경우, 일반적으로, 인코더 측에서 만들어진 부분 비트스트림은 종료되어야 한다, 즉 모든 소스 심벌이 송신되거나 저장된 부분 비트스트림으로부터 디코딩될 수 있음을 보장해야 한다. 마지막 이진수가 대응하는 이진수 버퍼(8)에 삽입된 이후, 이진수 인코더(10)는 전체 코드워드가 부분 비트스트림(12)에 기록됨을 보장해야 한다. 이진수 인코더(10)가 이진수 산술 코딩 엔진을 나타낸다면, 산술 코드워드는 종료되어야 한다. 이진수 인코더(10)가 코드워드로의 이진수 시퀀스의 직접 매핑을 구현하는 엔트로피 인코더를 나타낸다면, 마지막 이진수를 이진수 버퍼에 기록한 이후 이진수 버퍼에 저장된 이진수 시퀀스는 코드워드와 관련되는 이진수 시퀀스를 나타내지 않을 수 있다(즉, 이 시퀀스는 코드워드와 관련되는 둘 이상의 이진수 시퀀스의 프레픽스를 나타낼 수 있다). 그러한 경우, 이진수 버퍼에서 이진수 시퀀스를 프레픽스로서 포함하는 이진수 시퀀스와 관련된 코드워드 중 임의의 것은 부분 비트스트림에 기록되어야 한다(이진수 버퍼가 플러싱되어야 한다). 이러한 구성은, 특정한 또는 임의의 값을 갖는 이진수를 코드워드가 기록될 때까지 이진수 버퍼에 삽입함으로써 행해질 수 있다. 바람직한 실시예에서, (관련된 이진수 시퀀스가 프레픽스로서 이진수 버퍼에서 이진수 시퀀스를 포함해야 한다는 속성에 추가하여) 이진수 인코더는 최소 길이를 갖는 코드워드 중 하나를 선택한다. 디코더 측에서, 이진수 디코더(22)는 부분 비트스트림에서 마지막 코드워드에 필요한 것보다 더 많은 이진수를 디코딩할 수 있고, 이들 이진수는 이진수 버퍼 선택기(18)에 의해 요청되지 않아서 버려지고 무시된다. 유한 세트의 심벌의 디코딩은 디코딩된 소스 심벌에 대한 요청에 의해 제어되고, 추가 소스 심벌이 데이터 양에 대해 요청되지 않는다면, 디코딩이 종료된다.

[0147] **부분 비트스트림의 송신 및 다중화**

[0148] PIPE 인코더에 의해 만들어진 부분 비트스트림(12)은 개별적으로 송신될 수 있거나, 단일 비트스트림으로 다중화될 수 있거나, 부분 비트스트림의 코드워드가 단일 비트스트림으로 인터리빙될 수 있다.

[0149] 실시예에서, 데이터 양에 대한 각 부분 비트스트림을 하나의 데이터 패킷에 기록한다. 데이터 양은 정지 화상, 비디오 시퀀스의 필드나 프레임, 정지 화상의 슬라이스, 비디오 시퀀스의 필드나 프레임의 슬라이스, 또는 오디오 샘플의 프레임 등과 같은 소스 심벌의 임의의 세트일 수 있다.

[0150] 다른 바람직한 실시예에서, 다량의 데이터에 대한 부분 비트스트림 중 둘 이상이나, 다량의 데이터에 대한 모든 부분 비트스트림을 하나의 데이터 패킷으로 다중화한다. 다중화된 부분 비트스트림을 포함하는 데이터 패킷의 구조를 도 5에 예시한다. 즉, 도 5에 도시한 데이터 패킷은 중간 인터리빙된 스트림(132 및 234) 각각의 일부가 될 수 있다.

[0151] 데이터 패킷(300)은 헤더와 (고려된 다량의 데이터에 대한) 각 부분 비트스트림의 데이터에 대한 하나의 분할로 구성된다. 데이터 패킷의 헤더(300)는 데이터 패킷(그 나머지)의 비트스트림 데이터(302)의 세그먼트로의 분할에 대한 지시를 포함한다. 분할에 대한 지시 외에, 헤더는 추가 정보를 포함할 수 있다. 실시예에서, 데이터 패킷의 분할에 대한 지시는 비트나 바이트나 복수의 비트나 복수의 바이트 단위의 데이터 세그먼트의 시작의 위치이다. 실시예에서, 데이터 세그먼트의 시작의 위치는, 데이터 패킷의 시작에 대한 또는 헤드의 끝에 대한 또는 이전 데이터 패킷의 시작에 대한 데이터 패킷의 헤더에서 절대값으로 코딩된다. 다른 실시예에서, 데이터 세그먼트의 시작의 위치는 차별적으로 코딩된다. 즉, 데이터 세그먼트의 실제 시작과 데이터 세그먼트의 시작에 대한 예측 사이의 차이만이 코딩된다. 예측은 데이터 패킷의 전체 크기, 헤더의 크기, 데이터 패킷에서의 데이터 세그먼트의 개수, 이전 데이터 세그먼트의 시작의 위치와 같은 이미 알려져 있거나 송신된 정보를 기반으로 하여 유도할 수 있다. 실시예에서, 제1 데이터 패킷의 시작의 위치는 코딩되는 것이 아니라 데이터 패킷 헤더의 크기를 기반으로 추론된다. 디코더 측에서, 송신된 분할 지시는 데이터 세그먼트의 시작을 유도하는데 사용된다. 데이터 세그먼트는 그 후 부분 비트스트림으로서 사용되며, 데이터 세그먼트에 포함된 데이터가 순차적인 순서로 대응하는 이진수 디코더에 공급된다.

[0152] 부분 비트스트림(12)을 데이터 패킷으로 다중화하기 위한 여러 가지 대안이 있다. 특히 부분 비트스트림의 크기가 매우 유사한 경우에 대해 필요한 부차적인 정보를 감소시킬 수 있는 한 대안을 도 6에 예시한다. 데이터 패킷의 페이로드, 즉 그 헤더(311) 이외의 데이터 패킷(310)을 미리 한정된 방식으로 세그먼트(312)로 분할된다. 예로서, 데이터 패킷의 페이로드가 동일한 크기의 세그먼트로 분할될 수 있다. 그 후, 각 세그먼트는 부분 비트스트림과 또는 부분 비트스트림(313)의 제1 부분과 관련된다. 부분 비트스트림이 관련된 데이터 세그먼트보다 크다면, 그 나머지(314)는 다른 데이터 세그먼트의 끝에서 미사용 공간에 배치된다. 이러한 구성은, 비트스트림의 나머지 부분이 (데이터 세그먼트의 끝으로부터 시작해서) 역순으로 삽입되는 방식으로 행해질 수 있으며, 이 점은 부차적인 정보를 감소시킨다. 부분 비트스트림의 나머지의 데이터 세그먼트로의 관련성 및, 하나보다 많은 나머지가 데이터 세그먼트에 더해질 때, 나머지 중 하나 이상에 대한 시작점은 비트스트림 내부에서, 예컨대 데이터 패킷 헤더에서 신호화되어야 한다.

[0153] **가변 길이 코드워드의 인터리빙**

[0154] 일부 응용의 경우, 하나의 데이터 패킷에서 (소스 심벌 양에 대한) 부분 비트스트림(12)의 전술한 다중화는 다음의 단점이 있을 수 있다: 한편으론, 작은 데이터 패킷의 경우, 분할을 신호화하기 위해 필요한 부차적인 정보에 대한 비트 수가 부분 비트스트림에서 실제 데이터에 비해 상당해질 수 있으며, 이점은 코딩 효율을 결국 감소시킨다. 다른 한편으론, 다중화가 (예컨대, 비디오 회의 응용의 경우) 작은 지연을 필요로 하는 응용에 적절하지 않을 수 있다. 기재한 다중화로, PIPE 인코더는, 부분 비트스트림이 완전히 만들어지기 이전에 데이터 패킷의 송신을 시작할 수 없으며, 이는 분할의 시작의 위치가 이전에 알려져 있지 않기 때문이다. 더 나아가, 일반적으로, PIPE 디코더는, 데이터 패킷의 디코딩을 시작할 수 있기 전에 마지막 데이터 세그먼트의 시작을 수신할 때까지 대기해야 한다. 비디오 회의 시스템과 같은 응용의 경우, 이들 지연은 (화상을 인코딩/디코딩하기 위해 두 화상 사이의 시간 간격을 거의 필요로 하는 인코더/디코더에 대해서 및 송신 비트율에 가까운 비트율에 대해) 여러 비디오 화상의 시스템의 추가 전체 지연에 합해질 수 있으며, 이점은 그러한 응용에서 중요하다. 특정 응용에 대한 단점을 극복하기 위해, PIPE 인코더는, 둘 이상의 이진수 인코더에 의해 생성되는 코드워드가 단일 비트스트림으로 인터리빙되는 방식으로 구성될 수 있다. 인터리빙된 코드워드를 갖는 비트스트림은 (작은 버퍼 지연을 무시할 경우, 아래 참조) 직접 디코더에 전송될 수 있다. PIPE 디코더 측에서, 둘 이상의 이진수 디코더는 디코딩 순서로 비트스트림으로부터 직접 코드워드를 판독하며, 디코딩은 수신된 제1 비트로 시작할 수 있다. 게다가, 부차적인 정보는 부분 비트스트림의 다중화(또는 인터리빙)를 신호화하는데 필요치 않다.

[0155] 코드워드 인터리빙을 갖는 PIPE 인코더의 기본 구조를 도 7에 도시한다. 이진수 인코더(10)는 코드워드를 부분 비트스트림에 직접 기록하는 것이 아니라, 단일 코드워드 버퍼(29)와 연결되며, 이 버퍼로부터 코드워드가 코딩 순서로 비트스트림(34)에 기록된다. 이진수 인코더(10)는 하나 이상의 새 코드워드 버퍼 엔트리에 대한 요청(28)을 코드워드 버퍼(29)에 전송하며, 후에 코드워드(30)를 코드워드 버퍼(29)에 전송하며, 코드워드는 예비 버퍼 엔트리에 저장된다. 코드워드 버퍼(29)의 (일반적으로 가변 길이) 코드워드(31)가, 대응하는 비트(33)를 생성된 비트스트림(34)에 기록하는 코드워드 기록기(32)에 의해 접근된다. 코드워드 버퍼(29)는 선입선출 버퍼로서 동작하여, 더 일찍 예비된 코드워드 엔트리가 비트스트림에 더 일찍 기록된다.

[0156] 다른 일반화에서, 복수의 코드워드 버퍼와 부분 비트스트림(12)이 가능하며, 코드워드 버퍼의 수는 이진수 인코더의 수 미만이다. 이진수 인코더(10)는 하나 이상의 코드워드를 코드워드 버퍼(29)에 예비하여, 코드워드 버퍼에서의 하나 이상의 코드워드의 예비가 연결된 이진수 버퍼(8)에서의 특정 이벤트에 의해 트리거된다. 실시예에서, 코드워드 버퍼(29)는, PIPE 디코더가 도 1a에서 132와 도 2에서 134에 각각 대응하는 비트스트림(34)을 즉각 디코딩할 수 있는 방식으로 동작한다. 코드워드가 비트스트림에 기록되는 코딩 순서는 대응하는 코드워드가 코드워드 버퍼에 예비되는 순서와 동일하다. 실시예에서, 각 이진수 인코더(10)는 하나의 코드워드를 예비하며, 이 예비는 연결된 이진수 버퍼에서의 특정한 이벤트에 의해 트리거된다. 다른 실시예에서, 각 이진수 인코더(10)는 하나 보다 많은 코드워드를 예비하며, 이 예비는 연결된 이진수 버퍼에서의 특정한 이벤트에 의해 트리거된다. 다른 실시예에서, 이진수 인코더(10)는 상이한 양의 코드워드를 예비하며, 여기서 특정한 이진수 인코더에 의해 예비된 코드워드의 양은 특정한 이진수 인코더 및/또는 특정한 이진수 인코더/이진수 버퍼의 다른 속성(관련 확률 측정치, 이미 기록된 비트의 개수 등)에 의존할 수 있다.

[0157] 실시예에서, 코드워드 버퍼는 다음과 같이 동작한다. 새로운 이진수(7)가 특정한 이진수 버퍼(8)에 전송되고, 이진수 버퍼에 이미 저장된 이진수의 개수가 0이며, 특정한 이진수 버퍼와 관련된 이진수 인코더에 대한 코드워드 버퍼에 예비된 코드워드가 현재 없다면, 연결된 이진수 인코더(10)는 코드워드 버퍼에 요청을 전송하며, 이에 의해, 하나 이상의 코드워드 엔트리가 특정한 이진수 인코더에 대한 코드워드 버퍼(29)에 예비된다. 코드워드 엔트리는 가변 개수의 비트를 가질 수 있고, 버퍼 엔트리에서 비트 개수에 대한 상한 임계치는 보통 대응하

는 이진수 인코더에 대한 최대 코드워드 크기에 의해 주어진다. (코드워드 엔트리나 코드워드 엔트리들이 예비되어진) 이진수 인코더에 의해 생성된 그 다음 코드워드나 그 다음 코드워드들은 코드워드 버퍼의 예비된 엔트리 또는 엔트리들에 저장된다. 특정한 이진수 인코더에 대한 코드워드 버퍼에서의 모든 예비된 버퍼 엔트리가 코드워드로 채워지며, 그 다음 이진수가 특정한 이진수 인코더와 연결된 이진수 버퍼에 전송된다면, 하나 이상의 새 코드워드가 특정한 이진수 버퍼 등에 대한 코드워드 버퍼에 예비된다. 코드워드 버퍼(29)는 특정한 방식으로 선입선출 버퍼를 나타낸다. 버퍼 엔트리는 순차적인 순서로 예비된다. 대응하는 버퍼 엔트리가 더 일찍 예비되어진 코드워드가 더 일찍 비트스트림에 기록된다. 코드워드 기록기(32)는, 연속해서나 코드워드(30)가 코드워드 버퍼(29)에 기록된 이후 코드워드 버퍼(29)의 상태를 체크한다. 제1 버퍼 엔트리가 전체 코드워드를 포함한다면(즉, 버퍼 엔트리가 예비되는 것이 아니라, 코드워드를 포함한다면), 대응하는 코드워드(31)와 대응하는 버퍼 엔트리를 코드워드 버퍼(20)로부터 제거하고, 코드워드의 비트(33)를 비트스트림에 기록한다. 이러한 처리를, 제1 버퍼 엔트리가 코드워드를 포함하지 않을 때까지(즉, 예비되거나 사용중이지 않을 때까지) 반복된다. 디코딩 처리의 끝에서, 즉 고려된 다량의 데이터의 모든 소스 심벌이 처리되었다면, 코드워드 버퍼를 플러싱해야 한다. 그러한 플러싱 처리의 경우, 다음의 절차가 각각의 이진수 버퍼/이진수 인코더에 제1 단계로서 적용된다. 이진수 버퍼가 이진수를 포함한다면, 특정한 또는 임의의 값을 갖는 이진수를, 결과적인 이진수 시퀀스가 코드워드와 관련된 이진수 시퀀스를 나타낼 때까지 더하고(앞서 주목한 바와 같이, 이진수를 더하는 하나의 바람직한 방법은, 이진수 버퍼의 원래의 내용을 프레픽스로서 포함하는 이진수 시퀀스와 관련되는 최단 가능 코드워드를 생성하는 이진수 값-또는 이들 값 중 하나-을 더하는 것이다), 그 후, 코드워드가 대응하는 이진수 인코더에 대한 그 다음 예비된 버퍼 엔트리에 기록되며, (대응하는) 이진수 버퍼가 비워진다. 하나 보다 많은 버퍼 엔트리가 하나 이상의 이진수 인코더에 예비된다면, 코드워드 버퍼는 여전히 예비된 코드워드 엔트리를 포함할 수 있다. 그 경우, 이들 코드워드 엔트리는 대응하는 이진수 인코더에 대한 임의의 그러나 유효한 코드워드로 채워진다. 바람직한 실시예에서, 최단 유효 코드워드 또는(복수 개가 있다면) 최단 유효 코드워드 중 하나를 삽입한다. 끝으로, 코드워드 버퍼에서의 모든 남은 코드워드를 비트스트림에 기록한다.

[0158] 코드워드 버퍼의 상태에 대한 두 개의 예를 도 8에 예시한다. 예(a)에서, 코드워드 버퍼는, 코드워드로 채워진 2개의 엔트리와 5개의 예비 엔트리를 포함한다. 게다가, 그 다음 사용 중이지 않은 버퍼 엔트리를 표기한다. 제1 엔트리는 코드워드로 채워진다(즉, 이진수 인코더(2)는 코드워드를 이전에 예비된 엔트리에 막 기록했다). 다음 단계로, 이 코드워드는 코드워드 버퍼로부터 제거되고 비트스트림에 기록될 것이다. 그 후, 이진수 인코더(3)에 대한 제1 예비된 코드워드가 제1 버퍼 엔트리가 되지만, 이 엔트리는 코드워드 버퍼로부터 제거될 수 없으며, 이는 이 엔트리는 단지 예비된 것이며, 코드워드가 이 엔트리에 기록되지 않았기 때문이다. 예(b)에서, 코드워드 버퍼는 코드워드로 채워지는 3개의 엔트리와 4개의 예비 엔트리를 포함한다. 제1 엔트리는 예비로서 표기되며, 따라서 코드워드 기록기는 코드워드를 비트스트림에 기록할 수 없다. 3개의 코드워드가 코드워드 버퍼에 포함될지라도, 코드워드 기록기는, 코드워드가 이진수 인코더(3)에 대한 제1 예비된 버퍼 엔트리에 기록될 때까지 대기해야 한다. 코드워드는, 디코더 측에서 처리를 반전시킬 수 있기 위해서(하기 부분 참조), 이들 코드워드가 예비된 순서로 기록되어야 함을 주목해야 한다.

[0159] 코드워드 인터리빙을 갖는 PIPE 디코더의 기본 구조를 도 9에 도시한다. 이진수 디코더(10)는 별도의 부분 비트스트림으로부터 직접 코드워드를 판독하지 않지만, 비트 버퍼(38)에 연결되며, 이 버퍼(38)로부터 코드워드(37)가 코딩 순서로 판독된다. 비트 버퍼(38)가 반드시 필요한 것은 아니며, 이는 코드워드가 직접 비트스트림으로부터 판독될 수 도 있기 때문임을 주목해야 한다. 비트 버퍼(38)에는 주로 처리 체인의 명백하게 분리된 상이한 양상을 위해서 예시적으로 포함된다. 따라서 도 2의 비트스트림(234)에 대응하는 인터리빙된 코드워드를 갖는 비트스트림(40)의 비트(39)는 선입선출 버퍼를 나타내는 비트 버퍼(38)에 순차적으로 삽입된다. 특정한 이진수 디코더(22)가 하나 이상의 이진수 시퀀스에 대한 요청(35)을 수신한다면, 이진수 디코더(22)는 비트에 대한 요청(36)을 통해 비트 버퍼(38)로부터 하나 이상의 코드워드(37)를 판독한다. PIPE 디코더는 즉시 소스 심벌을 디코딩할 수 있다. (전술한 바와 같이) PIPE 인코더가, 코드워드 버퍼를 적절히 동작함으로써 코드워드가 이들 코드워드가 이진수 디코더에 의해 요청된 비트스트림에 동일한 순서로 기록됨을 보장해야 함을 주목해야 한다. PIPE 디코더에서, 전체 디코딩 처리는 소스 심벌에 대한 요청에 의해 트리거된다. 특정한 이진수 인코더에 의해 인코더 측에서 예비된 코드워드의 개수와 대응하는 이진수 디코더에 의해 판독된 코드워드의 개수로서 파라미터는 동일해야 한다.

[0160] 다른 일반화에서, 복수의 코드워드 버퍼와 부분 비트스트림이 가능하며, 여기서 비트 버퍼의 개수는 이진수 디코더의 개수 미만이다. 이진수 디코더(22)는 하 나의 시간 인스턴트에서 비트 버퍼(38)로부터 하나 이상의 코드워드를 판독하여, 비트 버퍼로부터의 하나 이상의 코드워드의 판독이 연결된 이진수 버퍼(20)에서 특정 이벤트에 의해 트리거된다. 실시예에서, 디코더는, 이진수에 대한 요청(19)이 특정한 이진수 버퍼(20)에 전송되고 이

진수 버퍼가 임의의 이진수를 포함하지 않을 때 하나 이상의 코드워드가 판독되는 방식으로 동작한다. 그러나 다른 이벤트, 예컨대 이진수 버퍼의 이진수의 개수가 미리 한정된 임계치 미만일 경우 코드워드의 판독을 트리거할 수 도 있다. 실시예에서, 각 이진수 디코더(22)는 하나의 코드워드를 판독하며, 판독은 연결된 이진수 버퍼에서의 특정 이벤트에 의해 트리거된다. 다른 실시예에서, 각 이진수 디코더(22)는 하나보다 많은 코드워드를 판독하며, 판독은 연결된 이진수 버퍼에서 특정 이벤트에 의해 트리거된다. 다른 실시예에서, 이진수 디코더(22)는 상이한 양의 코드워드를 판독하며, 특정한 이진수 디코더에 의해 판독된 코드워드 양은 (관련 확률 측정치, 이미 판독된 비트 수 등과 같은) 특정한 이진수 디코더/이진수 버퍼의 다른 속성 및/또는 특정한 이진수 디코더에 의존할 수 있다.

[0161] 실시예에서, 비트 버퍼로부터의 코드워드의 판독은 다음과 같이 동작한다. 새로운 이진수 요청(19)이 이진수 버퍼 선택기(18)로부터 특정한 이진수 버퍼(20)에 전송되고, 이진수 버퍼에서의 이진수의 개수가 0이라면, 연결된 이진수 디코더(22)는 비트 버퍼(38)로의 비트 요청(36)을 통해 비트 버퍼(38)로부터 하나 이상의 코드워드(37)를 판독한다. 이진수 디코더(22)는 판독된 코드워드(37)를 이진수 시퀀스(21)로 전환하여, 이들 이진수 시퀀스를 연결된 이진수 버퍼(20)에 저장한다. 이진수에 대한 요청(19)에 대한 마지막 응답으로서, 제1 삽입된 이진수가 이진수 버퍼(20)로부터 제거되어 이진수 버퍼 선택기(18)에 전송된다. 다른 이진수 요청의 응답으로서, 이진수 버퍼의 나머지 이진수는 이진수 버퍼가 비워질 때까지 제거된다. 추가 이진수 요청이 비트 버퍼 등으로부터 하나 이상의 새로운 코드워드를 판독하도록 이진수 디코더를 트리거한다. 비트 버퍼(38)는 미리 한정된 크기의 선입선출 버퍼를 나타내며, 비트스트림(40)으로부터의 비트(39)로 연속해서 채워진다. 코드워드가 이들이 디코딩 처리에 의해 요청되는 것과 동일한 방식으로 비트스트림에 기록됨을 보장하기 위해, 인코더 측에서의 코드워드 버퍼는 전술한 방식으로 동작할 수 있다.

[0162] 따라서, 복수의 엔트로피 디코더 각각은, 고정 길이의 코드워드를 가변 길이의 심벌 시퀀스에 매핑하도록 구성된 가변 길이 디코더일 수 있으며, 코드워드 버퍼(43)의 출력과 같은 코드워드 엔트리가 인터리빙된 코드워드의 단일 스트림을 수신하기 위해 제공될 수 있다. 복수의 엔트로피 디코더(22)는, 복수의 엔트로피 디코더로부터 선택기(18)에 의해 회수된 대로 재구성될 심벌 시퀀스의 심벌이 각각의 엔트로피 디코더에서 새로운 코드워드로부터 매핑될 새로운 심벌 시퀀스를 결국 야기하는 순서에 따라 순차적인 순서로 코드워드 엔트리로부터 코드워드를 회수하도록 구성될 수 있다.

[0163] **작은 지연 제약을 가진 가변 길이 코드워드의 인터리빙**

[0164] PIPE 코딩에 대한 기술한 코드워드 인터리빙은 임의의 분할 정보가 부차적인 정보로서 전송되는 것을 요구하지 않는다. 코드워드가 비트스트림에서 인터리빙되므로, 지연은 일반적으로 작다. 그러나 (예컨대, 코드워드 버퍼에서 저장된 최대 비트 수에 의해 명시되는) 특정한 지연 제약을 따름을 보장하지는 않는다. 더 나아가, 코드워드 버퍼에 대한 필요한 버퍼 크기는 이론적으로는 매우 크게 될 수 있다. 도 8(b)에서의 예를 고려할 때, 추가 이진수가 이진수 버퍼(3)에 전송되지 않으며, 따라서 이진수 인코더(3)는, 데이터 패킷의 끝에서의 플러싱 처리가 적용될 때까지 임의의 새로운 코드워드를 코드워드 버퍼에 전송하지 않는 것이 가능할 수 있다. 그 후 이진수 인코더(1 및 2)에 대한 모든 코드워드는, 이들이 비트스트림에 기록될 수 있기 전에 데이터 패킷의 끝까지 대기해야 할 것이다. 이러한 단점은 추가 메커니즘을 PIPE 인코딩 처리에 (및 또한 후술될 PIPE 디코딩 처리에) 더함으로써 회피할 수 있다. 그 추가 메커니즘의 기본 개념은, 지연 또는 지연의 상한에 관련된 측정치(이후 참조)가 명시한 임계치를 초과한다면, 제1 예비된 버퍼 엔트리가 (데이터 패킷의 끝에서와 유사한 메커니즘을 사용하여) 대응하는 이진수 버퍼를 플러싱함으로써 채워진다는 점이다. 그러한 메커니즘에 의해, 대기 중인 버퍼 엔트리의 개수는, 관련된 지연 측정치가 명시한 임계치 미만일 때까지 감소한다. 디코더 측에서, 지연 제약에 따르기 위해 인코더 측에 삽입되었던 이진수는 버려져야 한다. 이렇게 이진수를 버리기 위해서, 기본적으로 인코더 측에서와 동일한 메커니즘이 사용될 수 있다. 다음의 설명에서, 그러한 지연 제어에 대한 두 개의 예를 기재한다.

[0165] 일 실시예에서, 지연(또는 지연의 상한)에 대한 측정치는 코드워드 버퍼에서 활성 버퍼 엔트리의 개수이며, 여기서 활성 버퍼 엔트리의 개수는 예비된 버퍼 엔트리의 개수+코드워드를 포함하는 버퍼 엔트리의 개수이다. 제1 버퍼 엔트리는 항상 예비된 버퍼 엔트리이거나 사용 중이지 않은 버퍼 엔트리이며, 이는 제1 버퍼 엔트리가 코드워드를 포함하면, 이 코드워드는 비트스트림에 기록되기 때문임을 주목해야 한다. 예컨대, (응용에 의해 결정되는) 최대 허용 버퍼 지연이 D개의 비트이며, 모든 이진수 인코더에 대한 최대 코드워드 크기가 L이라면, 지연 제약을 위반하지 않고도 코드워드 버퍼에 포함될 수 있는 코드워드의 최대 개수에 대한 하한은 N=D/L에 의해 계산할 수 있다. 비트 단위의 지연 측정치(D)는 시스템에 의해서 필요한 것이 아니라, 코드워드의 최대 개수(N)는 인코더 및 디코더 모두에 알려져야 한다. 실시예에서, 코드워드 버퍼 엔트리 최대 개수(N)는 응용에 의해 정해

진다. 다른 실시예에서, 코드워드 버퍼 엔트리의 최대 개수(N)는 비트스트림 내부에서, 예컨대 데이터 패킷의 헤더(또는 슬라이드 헤더)에서, 또는 비트스트림에 포함되는 파라미터 세트에서 신호화된다. 이진수 인코더(10)가 하나 이상의 새로운 버퍼 엔트리의 예비를 위한 요청을 코드워드 버퍼(29)에 전송한다면, 새로운 코드워드 버퍼 엔트리가 예비되기 전에 다음의 처리를 수행한다(즉, 복수의 코드워드 버퍼 엔트리가 하나의 요청에 의해 예비된다면, 복수 회 수행된다): 현재 활성인 버퍼 엔트리의 개수+1(그 다음에 예비될 버퍼 엔트리를 고려하여)가 코드워드 버퍼 엔트리의 최대 개수(N)보다 크다면, (예비된) 제1 버퍼 엔트리가, 현재 활성인 버퍼 엔트리의 개수+1이 코드워드 버퍼 엔트리의 최대 개수(N) 이하가 될 때까지 다음의 설명에서 기재된 처리에 의해 플러싱된다. 예비된 버퍼 엔트리의 플러싱은 데이터 패킷의 끝에서의 플러싱과 유사하다: 대응하는 제1 버퍼 엔트리를 예비한 이진수 인코더(10)는 결과적인 이진수 시퀀스가 코드워드와 관련되는 이진수 시퀀스를 나타낼 때까지 특정한 또는 임의의 값을 갖는 이진수를 연결된 이진수 버퍼(8)에 더함으로써 플러싱되고, 코드워드는 그 후 예비된 버퍼 엔트리에 기록되며, (이진수 버퍼를 비우고 이전에 예비된 버퍼 엔트리를 제거하면서) 끝으로 비트스트림에 더해진다. 전술한 바와 같이, 이진수를 이진수 버퍼에 더하는 한 가지 바람직한 방식은 최단 가능 코드워드를 생성하는 이들 이진수를 더하는 것이다. 디코더 측에서, 유사한 처리가 지연 제약을 따르기 위해 더해진 이진수를 버리기 위해 수행된다. 그러므로 디코더는 비트 버퍼로부터 판독한 코드워드를 계수하는 카운터(C)를 유지한다(이 카운터는 비트 버퍼에 유지될 수 있다). 이 카운터(C)는 데이터 패킷의 디코딩의 시작에서 초기화되고(예컨대 0으로), 코드워드가 판독된 이후 1씩 증가한다. 게다가, 각 이진수 디코더(22)는, 마지막 코드워드가 대응하는 이진수 디코더(22)에 의해 판독되기 전에 코드워드 카운터(C)의 값을 저장하는 카운터(Cx)를 포함한다. 즉, 특정한 이진수 디코더(22)가 새로운 코드워드를 판독할 때, 그 카운터(Cx)는 제1 단계로서 C로 설정되고, 그 후 코드워드가 비트 버퍼로부터 판독된다. 이진수에 대한 요청(19)이 특정한 이진수 버퍼(20)에 전송되고, 전체 코드워드(C)와 연결된 이진수 디코더(22)의 카운터(Cx) 사이의 차(C-Cx)가 코드워드 버퍼 엔트리의 최대 개수(N)보다 클 때, 특정한 이진수 버퍼(20)에 현재 저장된 모든 이진수가 버려지고, 무시된다. 이 추가 단계 외에, 디코딩은 전술한 바와 같이 동작한다. 이진수에 대한 요청(19)이 전송된 이진수 버퍼(20)가 (모든 이진수가 이미 제거되었기 때문에 또는 작은 지연 메커니즘이 이진수 요청이 수신된 이후 제1 단계에서 모든 이진수를 버렸기 때문에) 비워진다면, 연결된 이진수 디코더(22)는 비트 버퍼(38) 등으로부터 하나 이상의 새로운 코드워드를 판독한다.

[0166] 다른 실시예에서, 지연(또는 지연의 상한)에 대한 측정치는 코드워드 버퍼에서 활성 버퍼 엔트리에 대한 최대 코드워드 길이의 합이며, 여기서 특정한 버퍼 엔트리에 대한 최대 코드워드 길이는 그 버퍼 엔트리와 관련되는 이진수 디코드에 의존한다. 예시로서, 버퍼 엔트리에 대한 최대 코드워드 길이는 예에서 6으로 나타내 진다. 다시, 제1 버퍼 엔트리는 항상 예비된 버퍼 엔트리 또는 사용 중이지 않은 버퍼 엔트리이며, 이는 제1 버퍼 엔트리가 코드워드를 포함하는 경우 이 코드워드가 비트스트림에 기록되기 때문임을 다시 주목해야 한다. (응용에 의해 결정되는) 최대 허용 버퍼 지연은 D비트이다. 이 최대 버퍼 지연(D)은 인코더 및 디코더 모두에게 알려져야 한다. 바람직한 실시예에서, 최대 버퍼 지연(D)은 응용에 의해 정해진다. 바람직한 실시예에서, 최대 버퍼 지연(D)은 비트스트림 내부에서, 예컨대 데이터 패킷의 헤더(또는 슬라이스 헤더) 또는 비트스트림에 포함되는 파라미터 세트에서 신호화된다. 이것은 비트, 또는 바이트, 또는 복수의 비트, 또는 복수의 바이트의 단위로 신호화될 수 있다. 이진수 인코더(10)가 하나 이상의 새로운 버퍼 엔트리의 예비를 위한 요청을 코드워드 버퍼(29)에 전송한다면, 다음의 처리를 새로운 코드워드 버퍼 엔트리가 예비되기 전에 수행된다(즉, 복수의 코드워드 버퍼 엔트리가 하나의 요청에 의해 예비된다면 복수 회 수행된다).

[0167] 모든 현재 활성인 버퍼 엔트리에 대한 최대 코드워드 길이의 합+예비될 버퍼 엔트리에 대한 최대 코드워드 길이가 최대 버퍼 지연(D)보다 크다면, (예비된) 제1 버퍼 엔트리는, 모든 활성인 버퍼 엔트리에 대한 최대 코드워드 길이의 합+예비될 버퍼 엔트리에 대한 최대 코드워드 길이가 최대 버퍼 지연(D) 이하일 때까지 전술한 처리에 의해 플러싱된다. 예컨대, 도 8(b)의 예를 고려하자. 모든 현재의 활성인 버퍼 엔트리에 대한 최대 코드워드 길이의 합이 29이다. 최대 버퍼 지연(D)이 32와 같게 설정된다고 가정하자. 새로운 버퍼 엔트리가 최대 코드워드 길이가 3인 이진수 인코더(2)에 의해 예비된다면, 제1 버퍼 엔트리는 플러싱되지 않으며, 이는 29+3이 32보다 크지 않기 때문이다. 그러나 그 다음 버퍼 엔트리가 최대 코드워드 길이가 7인 이진수 인코더(1)에 의해 예비된다면, 제1 버퍼 엔트리는 플러싱되며, 이는 29+7이 32보다 크기 때문이다. 예비한 버퍼 엔트리의 플러싱는 전술한 바와 같이 (특정한 또는 임의의 값을 갖는 이진수를 대응하는 이진수 버퍼에 더함으로써) 행해진다.

[0168] 디코더 측에서, 유사한 처리가 지연 제약을 따르기 위해 더해졌던 이진수를 버리기 위해 수행된다. 그러므로 디코더는 비트 버퍼로부터 판독된 코드워드에 대한 최대 코드워드 길이를 계수하는 카운터(C)를 유지한다(이 카운터는 비트 버퍼에 유지될 수 있다). 상이한 이진수 디코더와 관련되는 최대 코드워드 길이는 상이할 수 있음을 주목해야 한다. 카운터(C)는 데이터 패킷의 디코딩의 시작에서 (예컨대, 0으로) 초기화되며, 코드워드가 판독된

이후 증가한다. 이 카운터는 판독된 코드워드의 실제 길이만큼 증가하는 것이 아니라, 그 최대 길이만큼 증가한다. 즉, 코드워드가 특정한 이진수 디코더에 의해 판독되고, 특정한 이진수 디코더에 의해 사용된 코드워드 표와 관련된 최대 코드워드 길이가 Lx라면(상이한 이진수 디코더가 상이한 최대 코드워드 길이와 관련될 수 있다면), 카운터는 Lx만큼 증가한다. 전체 카운터(C) 외에, 각 이진수 디코더(22)는 마지막 코드워드가 대응하는 이진수 디코더(22)에 의해 판독되었기 전 코드워드 카운터(C)의 값을 저장하는 카운터(Cx)를 포함한다. 즉, 특정한 이진수 디코더(22)가 새로운 코드워드를 판독할 때, 그 카운터(Cx)는 제1 단계로서 C와 동일하게 설정되며, 그 후 코드워드는 비트 버퍼로부터 판독된다. 이진수에 대한 요청(19)이 특정한 이진수 버퍼(20)에 전송되고, 연결된 이진수 디코더(22)의 카운터(Cx)와 전체 카운터(C) 사이의 차이(C-Cx)가 최대 버퍼 지연(D)보다 클 때, 특정한 이진수 버퍼(20)에 현재 저장된 모든 이진수가 버려지고, 무시된다. 이 추가 단계 외에, 디코딩은 전술한 바와 같이 동작한다. 이진수에 대한 요청(19)이 전송된 이진수 버퍼(20)가 (모든 이진수가 이미 제거되었기 때문에 또는 작은 지연 메커니즘이 이진수 요청이 수신된 이후 제1 단계에서 모든 이진수를 버렸기 때문에) 비워진다면, 연결된 이진수 디코더(22)는 비트 버퍼(38) 등으로부터 하나 이상의 새로운 코드워드를 판독한다.

[0169] 따라서, 복수의 엔트로피 디코더(22)와 선택기(18)는 재구성될 심벌 시퀀스(29)를 형성하는데 참여하지 않기 위해 심벌 시퀀스의 서픽스를 간헐적으로 버리도록 구성될 수 있다. 간헐적으로 버리는 구성은, 코드워드 엔트리로부터 각각의 엔트리 디코더의 두 연속적인 코드워드 회수 사이에 복수의 엔트로피 디코더에 의해 코드워드 엔트리로부터 회수되었던 코드워드의 개수가 미리 결정된 기준을 충족하는 이벤트에서 실행될 수 있다. 복수의 엔트로피 인코더와 코드워드 버퍼는 이제 현재 전송되었지만 아직 매핑되지 않은 심벌을 프레픽스로서 갖는 돈케어(don't-care) 심벌에 의해 현재 전송되었지만 아직 매핑되지 않은 심벌을 간헐적으로 유효 심벌 시퀀스로 연장하고, 그에 따라 연장된 심벌 시퀀스를 코드워드에 매핑하고, 그에 따라 얻은 코드워드를 예비된 코드워드 엔트리에 입력하며 코드워드 엔트리를 플러싱하도록 구성될 수 있다. 간헐적인 연장, 입력 및 플러싱는 예비된 코드워드 엔트리의 개수+여기에 입력된 코드워드를 갖는 코드워드 엔트리의 개수가 미리 결정된 기준을 충족하는 이벤트에서 일어날 수 있다. 미리 결정된 기준은 복수의 인코더/디코더 쌍의 코드워드의 최대 길이를 고려할 수 있다.

[0170] 일부 아키텍처의 경우, 코드워드 인터리빙을 위한 전술한 바람직한 실시예는 결국 디코딩 복잡도 면에서 단점을 야기할 수 있다. 도 9에 예시한 바와 같이, 모든 이진수 디코더(22)는 단일 비트 버퍼(38)로부터 코드워드(일반적인 경우, 가변 길이 코드워드)를 판독한다. 코드워드의 판독은 병렬로 행해질 수 없으며, 이는 코드워드가 정확한 순서로 판독되어야 하기 때문이다. 이것이 의미하는 점은, 특정한 이진수 디코더가 다른 이진수 디코더가 코드워드의 판독을 마칠 때까지 대기해야 한다는 점이다. 가변 길이 코드워드의 판독의 복잡도가 (부분적으로 병렬화된) 디코딩 처리의 나머지에 대해 상당할 때, 가변 길이 코드워드의 이러한 접근은 전체 디코딩 처리에 대해 병목 구성이 될 수 있다. 단일 비트 버퍼로부터의 접근의 복잡도를 감소시키기 위해 채용할 수 있는 기재한 실시예의 일부 변형이 있으며, 이들 중 몇 가지를 다음에서 기재할 것이다. 하나의 바람직한 실시예에서, (예컨대, 중복되지 않은 프레픽스 코드를 나타내는) 단일 코드워드 세트가 존재하며, 각 이진수 디코더(22)에 사용되는 코드워드 세트는 단일 코드워드 세트의 서브세트이다. 상이한 이진수 디코더(22)가 단일 코드워드 세트의 상이한 서브세트를 사용할 수 있음을 주목해야 한다. 이진수 디코더(22) 중 일부에 의해 사용되는 코드워드 세트가 동일하더라도, 이진수 시퀀스와의 그 관련성은 상이한 이진수 디코더(22)마다 상이하다. 특정한 실시예에서, 동일한 코드워드 세트가 모든 이진수 디코더(22)에 사용된다. 모든 이진수 디코더에 대한 코드워드 세트를 서브세트로서 포함하는 단일 코드워드 세트를 갖는다면, 코드워드의 파싱은 이진수 디코더 외부에서 행해질 수 있으며, 이것은 코드워드 접근의 복잡도를 감소시킬 수 있다. PIPE 인코딩 처리는 전술한 처리에 비해 변하지 않는다. 변경된 PIPE 디코딩 처리를 도 10에 예시한다. 단일 코드워드 판독기에는 비트스트림(40)으로부터 비트(46)가 공급되며, -일반적으로 가변 길이의- 코드워드를 파싱한다. 판독된 코드워드(44)가 선입선출 버퍼를 나타내는 코드워드 버퍼(43)에 삽입된다. 이진수 디코더(22)는 하나 이상의 코드워드(41)에 대한 요청을 코드워드 버퍼(43)에 전송하며, 이 요청에 대한 응답으로서, 하나 이상의 코드워드가 코드워드 버퍼로부터 (순차적인 순서로) 제거되어, 대응하는 이진수 디코더(22)에 전송된다. 이 실시예로, 잠재적으로 복잡한 코드워드 파싱이 배경 처리에서 행해질 수 있으며, 이진 디코더를 대기할 필요가 없다는 점을 주목해야 한다. 이진수 디코더는 이미 파싱한 코드워드에 접근하며, 잠재적으로 복잡한 코드워드 파싱은 더 이상 전체 버퍼에 대한 요청의 일부가 아니다. 대신 이미 파싱된 코드워드가 이진수 디코더에 전송되며, 이러한 구성은, 코드워드 인덱스만이 이진수 디코더에 전송되는 방식으로 구현될 수 도 있다.

[0171] **고정 길이 비트 시퀀스의 인터리빙**

[0172] PIPE 디코더 복잡도를 감소시키는 추가적인 방식은, 이진수 디코더(22)가 글로벌 비트 버퍼(38)로부터 가변 길

이 코드워드를 판독하는 것이 아니라, 대신 글로벌 비트 버퍼(38)로부터 고정 길이의 비트 시퀀스를 항상 판독하고 이들 고정 길이 비트 시퀀스를 로컬 비트 버퍼에 더할 때 달성될 수 있으며, 각 이진수 디코더(22)는 별도의 로컬 비트 버퍼와 연결된다. 가변 길이 코드워드는 그 후 로컬 비트 버퍼로부터 판독된다. 그러므로 가변 길이 코드워드의 파싱은 병렬로 행해질 수 있으며, 고정 길이 비트 시퀀스의 접근만이 동기화된 방식으로 행해져야 하지만, 고정 길이 비트 시퀀스의 그러한 접근은 보통 매우 빨라서, 전체 디코딩 복잡도는 일부 아키텍처의 경우 감소할 수 있다. 특정한 로컬 비트 버퍼에 전송된 고정된 수의 이진수는 상이한 로컬 비트 버퍼마다 상이할 수 있으며, 이진수 디코더, 이진수 버퍼 또는 비트 버퍼에서의 이벤트로서 특정 파라미터에 따라 시간이 지남에 따라 변할 수 도 있다. 그러나 특정한 접근에 의해 판독된 비트 수는 특정한 접근 동안 판독되는 실제 비트에 의존하지 않으며, 이점이 가변 길이 코드워드의 판독에 대한 중요한 차이점이다. 고정 길이 비트 시퀀스의 판독은 이진수 버퍼, 이진수 디코더, 로컬 비트 버퍼에서의 특정 이벤트에 의해 트리거된다. 예컨대, 연결된 비트 버퍼에 존재하는 비트 수가 미리 한정된 임계치 미만으로 떨어질 때 새로운 고정 길이 비트 시퀀스의 판독을 요청할 수 있으며, 상이한 임계치 값은 상이한 비트 버퍼마다 사용될 수 있다. 인코더에서, 고정 길이 이진수 시퀀스가 비트에 동일한 순서로 삽입됨을 보장해야 하며, 이들 시퀀스는 디코더 측에서 비트스트림으로부터 판독된다. 전술한 것과 유사하게 고정 길이 시퀀스의 인터리빙을 작은 지연 제어와 결합하는 것도 가능하다. 다음의 설명에서, 고정 길이 비트 시퀀스의 인터리빙에 대한 바람직한 실시예를 기재한다.

[0173] 도 11은, 둘 이상의 이진수 인코더에 대한 고정 길이 비트 시퀀스를 인터리빙하는 실시예에 대한 PIPE 인코더 구조의 예시를 도시한다. 도 7에 예시한 실시예와는 대조적으로, 이진수 인코더(10)는 단일 코드워드 버퍼에 연결되지 않는다. 대신, 각 이진수 인코더(10)는 별도의 비트 버퍼(48)와 연결되며, 이 버퍼(48)는 대응하는 부분 비트스트림에 대한 비트를 저장한다. 모든 비트 버퍼(48)는 글로벌 비트 버퍼(51)에 연결된다. 글로벌 비트 버퍼(51)는 비트 기록기(53)에 연결되며, 기록기(53)는 글로벌 비트 버퍼로부터 코딩/디코딩 순서로 비트(52)를 제거하여 제거된 비트(54)를 비트스트림(55)에 기록한다. 특정한 비트 버퍼(48) 또는 연결된 이진수 인코더(10) 또는 이진수 버퍼(8)에서의 특정한 이벤트 시에, 비트 버퍼(48)는 요청(49)을 글로벌 비트 버퍼(51)에 전송하여, 그에 의해 특정한 비트 수가 글로벌 비트 버퍼(51)에 예비된다. 고정 길이 비트 시퀀스의 예비에 대한 요청(49)이 순차적인 순서로 처리된다. 글로벌 비트 버퍼(51)는 특정한 방식으로 선입선출 버퍼를 나타내며, 더 일찍 예비된 비트가 비트스트림에 더 일찍 기록된다. 상이한 비트 버퍼(48)는 상이한 비트 양을 예비할 수 있으며, 이 비트 양은 이미 코딩된 심벌을 기반으로 하여 시간이 지남에 따라 변할 수 도 있지만, 특정한 요청에 의해 예비되는 비트 수는 요청이 글로벌 비트 버퍼에 전송된 시간에 알려짐을 주목해야 한다.

[0174] 특히, 비트 버퍼(48)와 글로벌 비트 버퍼(51)는 다음의 설명에서 기재된 대로 동작한다. 특정한 비트 버퍼(48)에 의해 예비된 비트 양을 Nx라고 표시한다. 이 비트 수(Nx)는 상이한 비트 버퍼(48)마다 상이할 수 있으며, 시간이 지남에 따라 변할 수 도 있다. 바람직한 실시예에서, 특정한 비트 버퍼(48)에 의해 예비된 비트 수(Nx)는 시간이 지남에 따라 고정된다. 고정된 비트 수(Nx)(49)의 예비는 비트 버퍼(48)에서의 비트 수(Mx), 예비 요청에 대한 비트 수(Nx), 및 관련된 최대 코드워드 길이(Lx)를 기반으로 하여 트리거된다. 각 이진수 인코더(10)는 상이한 최대 코드워드 길이(Lx)와 관련될 수 있음을 주목해야 한다. 이진수(7)가 특정한 이진수 버퍼(8)에 전송되고, 특정한 이진수 버퍼(8)가 비어있고, Nx 비트의 하나 보다 많은 시퀀스가 (이진수 인코더를 통해) 특정한 이진수 버퍼와 연결되는 비트 버퍼(48)에 대해 글로벌 비트 버퍼에 예비되지 않고, (이진수 인코더를 통해) 특정한 이진수 버퍼(8)와 연결되는 비트 버퍼(48)의 예비 요청에 의해 예비된 비트 수(Nx)와 이 비트 버퍼(48)에 현재 존재하는 비트 수(Mx) 사이의 차이(Nx-Mx)가 대응하는 이진수 인코더(10)와 관련되는 최대 코드워드 길이(Lx) 미만이라면, 연결된 비트 버퍼(49)는 Nx 비트의 예비를 위한 요청(49)을 글로벌 비트 버퍼(51)에 전송한다. 글로벌 비트 버퍼(51)는 특정한 비트 버퍼(48)를 위해 Nx 비트를 예비하며, 그 다음 예비를 위해 그 포인터를 증가시킨다. Nx 비트가 글로벌 비트 버퍼에 예비된 후, 이진수(7)는 이진수 버퍼(8)에 저장된다. 이 하나의 이진수가 코드워드와 관련된 이진수 시퀀스를 이미 나타낸다면, 이진수 인코더(10)는 이 이진수를 이진수 버퍼(8)로부터 제거하여 대응하는 코드워드(47)를 연결된 비트 버퍼(48)에 기록한다. 그렇지 않다면(이 하나의 이진수가 코드워드와 관련된 이진수 시퀀스를 이미 나타낸다면), 이진수 버퍼(8)가 코드워드와 관련된 이진수 시퀀스를 포함할 때까지, 추가 이진수(7)가 특정한 이진수 버퍼(8)에 의해 수용된다. 이 경우, 연결된 이진수 인코더(10)는 이진수 버퍼(8)로부터 이진수 시퀀스(9)를 제거하여, 대응하는 코드워드(47)를 연결된 비트 버퍼(48)에 기록한다. 비트 버퍼(48)의 결과적인 비트 수(Mx)가 예비된 비트 수(Nx) 이상이라면, 비트 버퍼(48)에 먼저 기록되어있던 Nx 비트는 글로벌 비트 버퍼(51)의 이전에 예비된 공간에 삽입된다. 특정한 이진수 버퍼(8)에 전송된 그 다음 이진수(7)의 경우, 앞서 명시한 바와 동일한 처리가 수행된다. 즉, 새로운 Nx 비트 수가 글로빌 비트 버퍼에 예비되어야 하는지를(Nx-Mx가 Lx미만인지) 먼저 체크하여, 이진수가 이진수 버퍼(8)에 삽입되는 등의 처리가 수행된다. 비트 기록기는 글로벌 비트 버퍼의 고정 길이 비트 시퀀스를 이들이 예비되었던 순서

로 기록한다. 글로벌 비트 버퍼(51)의 제1 길이 엔트리가 글로벌 비트 버퍼에 실제 삽입되었던 고정 길이 비트 시퀀스를 포함한다면(즉, 예비된 것이 아니라면), 비트 기록기(53)는 글로벌 비트 버퍼(51)로부터 이 비트 시퀀스(52)에 대한 비트를 제거하여 비트(54)를 비트스트림에 기록한다. 이 처리는, 글로벌 비트 버퍼의 제1 고정 길이 엔트리가 예비되거나 사용 중이지 않은 엔트리를 나타낼 때까지 반복된다. 글로벌 비트 버퍼에서의 제1 고정 길이 엔트리가 예비된 엔트리라면, 비트 기록기(53)는, 추가 비트(54)를 비트스트림(55)에 기록하기 전에 이 엔트리가 실제 비트로 채워질 때까지 대기한다.

[0175] 데이터 패킷의 끝에서, 이진수 버퍼는 전술한 바와 같이 플러싱된다. 게다가, 비트 버퍼가, 글로벌 비트 버퍼에서 모든 예비된 버퍼 엔트리가 채워져 비트스트림에 기록될 때까지 특정한 또는 임의의 값을 갖는 비트를 더함으로써 플러싱되어야 한다.

[0176] 도 12에서, 글로벌 비트 버퍼(51)의 가능한 상태에 대한 두 가지 예를 예시한다. 예(a)에서, 상이한 비트 버퍼/이진수 인코더가 상이한 비트 수를 예비하는 경우를 예시한다. 글로벌 비트 버퍼는 실제 기록된 고정 길이 비트 시퀀스를 갖는 3개의 엔트리와 예비된 고정 길이 비트 시퀀스를 갖는 4개의 엔트리를 포함한다. 제1 고정 길이 엔트리는 이미 (비트 버퍼/이진수 인코더(2)에 의해 막 삽입되었음이 분명한) 실제 비트를 포함하며, 이 엔트리(즉, 대응하는 8비트)는 제거될 수 있어서 비트스트림에 기록될 수 있다. 그 다음 엔트리는 이진수 인코더(3)를 위해 10비트를 예비하지만, 실제 비트는 아직 삽입되지 않았다. 이 엔트리는 비트스트림에 기록될 수 없으며, 실제 비트가 삽입될 때까지 대기해야 한다. 제2 예(b)에서, 모든 비트 버퍼/이진수 인코더는 동일한 비트 수(8비트)를 예비하였다. 글로벌 비트 버퍼는 8비트 시퀀스를 위한 4개의 예비와 3개의 실제 기록된 8비트 시퀀스를 포함한다. 제1 엔트리는 이진수 인코더(3)를 위한 8비트에 대한 예비를 포함한다. 임의의 새로운 비트가 비트스트림에 기록될 수 있기 전, 비트 기록기는, 비트 버퍼/이진수 인코더(3)가 8비트의 실제 값을 이 예비된 엔트리에 기록할 때까지 대기해야 한다.

[0177] 도 13은 고정 길이 비트 시퀀스를 인터리빙하는 본 발명의 실시예에 대한 PIPE 디코더 구조의 예시를 도시한다. 도 9에 도시한 실시예와 대조적으로, 이진수 디코더(22)는 단일 비트 버퍼에 연결되지 않는다. 대신, 각 이진수 디코더(22)는 별도의 비트 버퍼(58)와 연결되며, 이 버퍼(58)는 대응하는 부분 비트스트림으로부터 비트를 저장한다. 모든 비트 버퍼(58)는 글로벌 비트 버퍼(61)에 연결된다. 비트스트림(63)으로부터의 비트(62)는 글로벌 비트 버퍼(61)에 삽입된다. 특정한 비트 버퍼(58) 또는 연결된 이진수 디코더(22) 또는 이진수 버퍼(20)에서의 특정한 이벤트 시, 비트 버퍼(58)가 요청(59)을 글로벌 비트 버퍼(61)에 전송하여, 그에 의해 고정 길이 비트 시퀀스(60)가 글로벌 비트 버퍼(61)로부터 제거되어 특정한 비트 버퍼(58)에 삽입된다. 고정 길이 비트 시퀀스에 대한 요청(59)은 순차적인 순서로 처리된다. 글로벌 비트 버퍼(61)는 선입선출 버퍼를 나타내며, 글로벌 비트 버퍼에 더 일찍 삽입된 비트가 더 일찍 제거된다. 상이한 비트 버퍼(58)가 이미 디코딩된 심벌을 기반으로 하여 시간이 지남에 따라 변할 수 도 있는 상이한 비트 수를 요청할 수 있지만, 특정한 요청에 의해 요청된 비트 수는 요청이 글로벌 비트 버퍼에 전송된 시간에 알려져 있음을 주목해야 한다. 글로벌 비트 버퍼(61)가 반드시 필요하지는 않으며, 이는 코드워드가 비트스트림으로부터 직접 판독될 수 도 있기 때문임을 주목해야 한다. 글로벌 비트 버퍼(61)는 주로 처리 체인의 명백히 분리된 상이한 양상에 대해 예시에서 주로 포함된다.

[0178] 실시예에서, 비트 버퍼(58) 및 글로벌 비트 버퍼(61)는 다음의 설명에서 기재된 대로 동작한다. 특정한 비트 버퍼(58)에 의해 요청되고 판독되는 비트 양을 Nx로서 표시하며, 이것은 인코더 측에서 대응하는 비트 버퍼에 의해 글로벌 비트 버퍼에 기록되는 비트 양과 같다. 이 비트 수(Nx)는 상이한 비트 버퍼(58)마다 상이할 수 있으며, 시간이 지남에 따라 변할 수 도 있다. 본 발명의 바람직한 실시예에서, 특정한 비트 버퍼(58)에 의해 요청되고 판독되는 비트 수(Nx)는 시간이 지남에 따라 고정된다. 고정된 비트 수(Nx)(60)의 판독은 비트 버퍼(58)에서의 비트 수(Mx) 및 관련된 최대 코드워드 길이(Lx)를 기반으로 하여 트리거된다. 각 이진수 디코더(22)는 상이한 최대 코드워드 길이(Lx)와 관련될 수 있음을 주목해야 한다. 이진수에 대한 요청(19)이 특정한 이진수 버퍼(20)에 전송되고, 특정한 이진수 버퍼(20)가 비어있고, (이진수 디코더를 통해) 특정한 이진수 버퍼(20)와 연결되는 비트 버퍼(58)의 비트 수(Mx)가 대응하는 이진수 디코더(22)와 관련되는 최대 코드워드 길이(Lx) 미만이라면, 연결된 비트 버퍼(58)는 Nx 비트의 새로운 시퀀스를 위한 요청(59)을 글로벌 비트 버퍼(61)에 전송한다. 이 요청에 대한 응답으로서, 제1 Nx비트가 글로벌 비트 버퍼(61)로부터 제거되어 이 Nx 비트 시퀀스(60)는 요청이 전송되었던 비트 버퍼(58)에 전송된다. 끝으로, 이 Nx 비트의 시퀀스가 대응하는 비트 버퍼(58)에 더해진다. 그 후, 그 다음 코드워드(57)가 이 비트 버퍼로부터 판독되고, 연결된 이진수 디코더(22)는 관련된 이진수 시퀀스(21)를 연결된 이진수 버퍼(20)에 삽입한다. 이진수에 대한 원래의 요청(19)에 대한 마지막 응답으로서, 제1 이진수가 이진수 버퍼(20)로부터 제거되고, 이 디코딩된 이진수(25)가 이진수 버퍼 선택기(18)에 전송된다. 그 다음 이진수 요청(19)이 특정한 이진수 버퍼(20)에 전송되고, 이진수 버퍼가 비워질 때, 그 다음 비트가 이진수

버퍼(20)로부터 제거된다. 이진수 버퍼가 비어지지만, 연결된 비트 버퍼(58)에서의 비트 수(Mx)가 관련된 최대 코드워드 길이(Lx) 이상이라면, 그 다음 코드워드가 비트 버퍼로부터 판독되고, 새로운 이진수 시퀀스가 이진수 버퍼에 삽입되며, 이로부터 제1 비트가 제거되어 이진수 버퍼 선택기에 전송된다. 이진수 버퍼가 비어지고 연결된 비트 버퍼(58)에서의 비트 수(Mx)가 관련된 최대 코드워드 길이(Lx) 미만이라면, 그 다음 Nx 비트 시퀀스가 글로벌 비트 버퍼(61)로부터 판독되어 연결된 로컬 비트 버퍼(58)에 삽입되며, 그 다음 코드워드가 비트 버퍼로부터 판독되고, 새로운 이진수 시퀀스가 이진수 버퍼에 삽입되며, 시퀀스의 제1 이진수가 제거되어 이진수 버퍼 선택기에 전송된다. 모든 소스 심벌이 디코딩될 때까지 이 처리는 반복된다.

[0179] 데이터 패킷의 끝에서, 요청된 소스 심벌을 디코딩하기 위해 필요한 것보다 더 많은 이진수 및/또는 비트가 이진수 버퍼 및/또는 비트 버퍼에 삽입될 수 도 있다. 이진수 버퍼에서의 나머지 이진수 및 비트 버퍼에서의 나머지 비트는 버려지고 무시된다.

[0180] **작은 지연 제약을 갖는 고정 길이 비트 시퀀스의 인터리빙**

[0181] 고정 길이 비트 시퀀스의 인터리빙을 갖는 PIPE 인코더 및 디코더에 대한 기술한 실시예는 또한 앞서 기재한 인코더 버퍼 지연을 제어하는 방식과 결합할 수 있다. PIPE 코딩 개념은 앞서 기재한 지연 제어를 갖는 실시예에서와 동일하다. 지연 또는 지연의 상한(아래의 기재 참조)에 관련된 측정치가 명시한 임계치를 초과한다면, 제1 예비된 버퍼 엔트리는 (데이터 패킷의 끝에서와 유사한 메커니즘을 사용하여) 대응하는 이진수 버퍼를 플러싱함으로써 그리고 예비된 고정 길이 버퍼 엔트리의 모든 비트를 채우기 위해 추가 비트를 잠재적으로 기록함으로써 채워진다. 그러한 메커니즘에 의해, 대기 중인 버퍼 엔트리의 개수는, 관련 지연 측정치가 명시한 임계치 미만이 될 때까지 감소한다. 디코더 측에서, 지연 제약을 따르기 위해 인코더 측에서 삽입되었던 이진수 및 비트는 버려져야 한다. 이러한 이진수 및 비트의 버림을 위해, 기본적으로 인코더 측에서와 동일한 메커니즘을 사용할 수 있다.

[0182] 실시예에서, 지연(또는 지연의 상한)에 대한 측정치는 글로벌 비트 버퍼에서의 활성 버퍼 엔트리의 비트 수이며, 여기서 활성 버퍼 엔트리의 개수는 예비된 고정 길이 버퍼 엔트리의 개수+이미 기록된 비트를 포함하는 고정 길이 버퍼 엔트리의 개수이다. 제1 버퍼 엔트리는 항상 예비된 고정 길이 버퍼 엔트리나 사용 중이지 않은 버퍼 엔트리이며, 이는 제1 버퍼 엔트리가 기록된 비트를 포함한다면 이들 비트는 비트스트림에 기록되기 때문임을 주목해야 한다. (응용에 의해 결정되는) 최대 허용 버퍼 지연은 D비트로 하자. 이 최대 버퍼 지연(D)은 인코더 및 디코더 모두에 알려져야 한다. 본 발명의 바람직한 실시예에서, 최대 버퍼 지연(D)은 응용에 의해 정해진다. 본 발명의 다른 바람직한 실시예에서, 최대 버퍼 지연(D)은 비트스트림 내부에서, 예컨대 데이터 패킷의 헤더(또는 슬라이스 헤더)에서, 또는 비트스트림에 포함되는 파라미터 세트에서 신호화된다. 이것은 비트, 또는 바이트, 또는 복수의 비트, 또는 복수의 바이트 단위로 신호화될 수 있다. 이진수 인코더(10)가 새로운 고정 길이 비트 시퀀스의 예비를 위한 요청을 글로벌 비트 버퍼(51)에 전송한다면, 다음의 공정이, 새 고정 길이 버퍼 엔트리가 예비되기 전에 수행된다.

[0183] 글로벌 비트 버퍼에서의 활성 버퍼 엔트리의 비트 수+현재의 예비 요청에 의해 예비될 비트 수가 최대 버퍼 지연(D)을 초과한다면, (예비되는) 제1 버퍼 엔트리는, 글로벌 비트 버퍼에서의 활성 버퍼 엔트리의 비트 수+현재의 예비 요청에 의해 예비될 비트 수가 최대 버퍼 지연(D) 이상이 될 때까지 다음의 설명에 기재된 처리에 의해 플러싱된다. 예비된 고정 길이 버퍼 엔트리의 플러싱은 데이터 패킷의 끝에서의 플러싱과 유사하다. 대응하는 제1 버퍼 엔트리를 예비한 비트 버퍼(48)와 연결되는 이진수 인코더(10)는, 결과적인 이진수 시퀀스가 코드워드와 관련된 이진수 시퀀스를 나타낼 때까지 특정한 또는 임의의 값을 갖는 이진수를 연결된 이진수 버퍼(8)에 더함으로써 플러싱되고, 코드워드는 그 후 대응하는 비트 버퍼(48)에 삽입된다. 앞서 언급한 바와 같이, 이진수를 이진수 버퍼에 더하는 한 가지 바람직한 방식은 최단 가능 코드워드를 발생하는 이들 이진수를 더하는 것이다. 코드워드를 연결된 비트 버퍼에 기록하고 고정 길이 비트 시퀀스를 글로벌 비트 버퍼에 잠재적으로 삽입한 후, 비트 버퍼에 여전히 비트가 있다면(즉, 기록된 코드워드가 예비된 고정 길이 비트 시퀀스를 완전히 채우지 않았다면), 특정한 또는 임의의 값을 갖는 추가 비트가, 모든 비트가 비트 버퍼로부터 제거되어 예비된 버퍼 엔트리에 기록될 때까지 비트 버퍼에 더해진다. 끝으로, 이 처리의 끝에서, 전체 버퍼 엔트리(글로벌 비트 버퍼에서의 제1 고정 길이 엔트리)는 글로벌 비트 버퍼로부터 제거되어 비트스트림에 기록된다.

[0184] 디코더 측에서, 유사한 처리가 지연 제약을 따르기 위해 더해진 이진수 및 비트를 버리기 위해 수행된다. 그러므로 디코더는 글로벌 비트 버퍼로부터 판독한 비트를 계수하는 카운터(C)를 유지한다(이 카운터는 글로벌 비트 버퍼에 유지될 수 있다). 이 카운터(C)는 데이터 패킷의 디코딩의 시작에서 (예컨대 0으로) 초기화되고, 고정 길이 시퀀스가 판독된 이후 증가한다. Nx 비트의 고정 길이 시퀀스가 글로벌 비트 버퍼(61)로부터 판독된다면,

카운터(C)는 Nx만큼 증가한다. 전체 카운터(C) 외에, 각 비트 버퍼(58)는 카운터(Cx)를 포함하며, 카운터(Cx)는 마지막 고정 길이 비트 시퀀스가 대응하는 비트 버퍼(58)에 판독되었기 전 비트 카운터(C)의 값을 저장한다. 특정한 비트 버퍼(58)가 새로운 고정 길이 비트 시퀀스를 판독할 때, 그 카운터(Cx)는 제1 단계로서 C와 동일하게 설정된 후 고정 길이 비트 시퀀스는 글로벌 비트 버퍼(61)로부터 판독된다. 이진수에 대한 요청(19)이 특정한 이진수 버퍼(20)에 전송되고, 전체 카운터(C)와 연결된 비트 버퍼(58)의 카운터(Cx) 사이의 차(C-Cx)가 최대 버퍼 지연(D)보다 클 때, 특정한 이진수 버퍼(20)에 현재 저장된 모든 이진수와 연결된 비트 버퍼(58)에 저장된 모든 비트는 버려지고, 무시된다. 이 추가 단계 외에, 디코딩은 전술한 바와 같이 동작한다. 이진수에 대한 요청(19)이 전송된 이진수 버퍼(20)가 (모든 이진수가 이미 제거되었기 때문에 또는 작은 지연 메커니즘이 이진수 요청이 수신된 이후 제1 단계에서 모든 이진수를 버렸기 때문에) 비워진다면, 연결된 이진수 디코더(22)는 연결된 비트 버퍼(58)로부터 새로운 코드워드를 판독하려고 시도한다. 비트 버퍼(58)에서의 비트 수가 최대 코드워드 길이미만이라면, 새로운 고정 길이 비트 시퀀스가, 코드워드가 판독되는 등 전에 글로벌 비트 버퍼(61)로부터 판독된다.

[0185] 도 7 내지 도 13에 관해 이제까지 기재하였던 실시예는 한편으론 PIPE 인코더(104)와 다른 한편으론 PIPE 디코더(202) 사이의 인터리빙된 비트스트림 경로를 달성하기 위한 가능성에 관한 것이다. 도 1 및 도 2에 관해 전술한 바와 같이, 엔트로피 인코딩 및 디코딩 장치는 두 개의 별도의 채널-그 중 하나는 VLC 비트스트림(112)을 전달하고 그 중 다른 하나는 인터리빙된 PIPE 인코딩된 비트스트림을 전달함-에 의해 서로 연결될 수 있다. 그러나 PIPE 인코딩된 비트스트림(118)뿐만 아니라 VLC 비트스트림(112) 모두를 인터리빙할 가능성이 또한 있으며, 그러한 가능성은 도 20 내지 도 24를 참조하여 후술될 것이다. 그러나 그 이전에, 결과적인 개별 부분 간격을 개별 엔트로피 인코더(116) 및 엔트로피 디코더(210)에 각각 할당하여 확률 간격을 최적으로 세분하는 방법에 관한 세부 내용뿐만 아니라 PIPE 코딩 방식에 관한 수학적인 배경이 제공된다.

[0186] 이미 주목한 바와 같이, PIPE 코딩에서, 이산 심벌의 입력 시퀀스의 이벤트 공간은 이진수 확률 간격의 작은 세트에 매핑된다. 소스 심벌에 대한 확률 모델은 고정되거나 적응적일 수 있는 반면, 확률 간격을 사용한 엔트로피 코딩은 고정되어 유지되고 모델링 스테이지로부터 분리된다. 확률 간격 각각은, 허프만 코드의 복잡도 레벨을 갖는 매우 간단한 엔트로피 코드를 사용하여 코딩될 수 있다. 확률 간격 분할 엔트로피(PIPE: Probability Interval Partitioning Entropy)의 초과율은 순수한 산술 코딩의 초과율과 유사하다.

[0187] 엔트로피 코딩은, 일반적으로, 무손실 데이터 압축의 가장 일반적인 형태로서 간주될 수 있다. 무손실 압축은 이산 데이터를 원래의 데이터 표현에 필요한 것보다 더 적은 비트로 그러나 어떤 정보 손실 없이 표현하는 것을 목적으로 한다. 이산 데이터는 텍스트, 그래픽스, 영상, 비디오, 오디오, 음성, 팩시밀리, 의료 데이터, 기상 데이터, 금융 데이터, 또는 임의의 다른 디지털 데이터 형태의 형태로 제공될 수 있다. 많은 코딩 응용에서, 원래의 소스 데이터는 먼저 소위 코딩 심벌에 매핑되며, 이들 코딩 심벌은 그 후 엔트로피 코딩된다. 코딩 심벌에의 매핑은 양자화를 포함할 수 있으며, 그 경우, 전체 코딩 방식은 손실을 입게 된다. 코딩 심벌(s)은 M-진($M \geq 2$) 알파벳($A=\{a_0,...a_{M-1}\}$ 중 임의의 값을 가질 수 있다. 심벌(s)을 코딩하기 위해, 알파벳은 추정된 확률 질량 함수(pmf: probability mass function)$\{p_s(a_0),...p_s(a_{M-1})\}$와 관련되며, 이 pmf에서 고려되지 않은 코딩 심벌 사이의 모든 의존성은 무시한다. 이들 추상적인 세팅을 위해, 엔트로피

$$H_s = -\sum_{i=0}^{M-1} p_s(a_i)\log_2 p_s(a_i) \qquad \text{(B1)}$$

[0188]

[0189] 는, 엔트로피 코딩 기술로 달성할 수 있는, 심벌(s)을 코딩하기 위한, 심벌당 비트 단위인 예상 코드워드 길이에 대한 최대 하한이다. 수십 년간, 허프만 코딩 및 산술 코딩은 실제 엔트로피 코딩을 지배해왔다. 이들 코딩은 (특정한 면에서) 엔트로피 한계를 근사화할 수 있는 실제 코드의 잘 알려져 있는 예이다.

[0190] 고정된 확률 분포의 경우, 허프만 코드는 상대적으로 구성이 쉽다. 허프만 코드의 가장 매력적인 속성은, 이들 코드의 구현이 가변 길이 코드(VLC) 표의 사용에 의해 효율적으로 실현할 수 있다는 점이다. 그러나 시변 소스 통계를 다룰 때, 즉 심벌 확률을 변경할 때, 허프만 코드 및 그 대응하는 VLC 표의 적응은 구현 비용 면에서뿐만 아니라 알고리즘 복잡도 면에서 모두 꽤 힘든 것이다. 또한, $p_s(a_i)>0.5$인 지배적인 알파벳 값을 갖는 경우에, (런 렝쓰(run length) 코딩과 같은 임의의 알파벳 확장을 사용하지 않고도) 대응하는 허프만 코드의 중복도는 상당할 수 있다. 허프만 코드의 다른 단점은, 더 고차의 확률 모델링을 다루는 경우에, 복수의 VLC 표 세트가 요구될 수 있다는 점에 의해 주어진다.

[0191] 다른 한편으로 산술 코딩은, VLC보다 상당히 더 복잡하면서도, 매우 왜곡된 확률 분포인 경우뿐만 아니라 적응 및 고차 확률 모델링에 대처할 때 더 일관되고 충분히 다룬다는 장점을 제공한다. 실제로, 이러한 특징은, 산술 코딩이, 적어도 개념적으로 확률 추정의 임의의 주어진 값을 다소 직접적인 방식으로 결과적인 코드워드의 일부분에 매핑하는 메커니즘을 제공한다는 점으로부터 기본적으로 유래한다. 그러한 인터페이스로 제공되므로, 산술 코딩은 한편으론 확률 모델링 및 확률 추정의 작업과 다른 한편으론 실제 엔트로피 코딩, 즉 심벌의 코드워드로의 매핑 사이의 명백한 분리를 허용한다.

[0192] 바로 앞에서 논의한 종래의 엔트로피 코딩 방식과는 달리, PIPE 코딩은 확률 간격 분할을 사용하며, 그 수학적 배경을 이후에 더 상세하게 기재할 것이다.

[0193] 코딩 심벌 시퀀스($\{s_0,\ldots,s_{N-1}\}$)를 고려한다. 각 심벌은 알파벳($s_i \in A_i$)으로부터 가져온다. 알파벳($A_i=\{a^i_0,a^i_1,\ldots\}$)은 각각 확률 추정($p_s(a^i_m)$)과 관련된 둘 이상의 문자를 포함한다. 확률 추정($p_s(a^i_m)$)은 인코더 및 디코더에 알려져 있고, 고정 또는 가변적일 수 있다. 가변 확률은 인코더 및 디코더에서 즉시 추정된다고 가정한다. 알파벳(A_i)은 심벌 시퀀스에 동일하거나 상이한 심벌 타입이 상이한 알파벳과 관련된다. 후자의 경우, 디코더는 시퀀스의 각 심벌의 알파벳을 알고 있다고 가정한다. 이러한 가정은, 실제 소스 코드 기재가 심벌 및 그 알파벳의 순서를 규정하는 신택스를 포함하므로 정당하다.

[0194] 심벌 시퀀스($\{s_0,\ldots,s_{N-1}\}$)는 또한 이진수로도 지칭되는 이진 심벌 시퀀스로 변환된다. 각 심벌(s_i)에 대해, 이진화

[0195] $$\boldsymbol{b}^i = \{b^i_0 \ldots\} = \gamma^i_b(s_i) \qquad \text{(B2)}$$

[0196] 는 알파벳 레터(a^i_m)의 이진수의 순서가 정해진 세트($\mathbf{b}^i_m$)로의 바이젝티브 매핑을 나타낸다. 이진화 매핑(γ^i_b)은 상이한 심벌(s_i)이나 심벌 카테고리마다 상이할 수 있다. 실제 심벌(s_i)에 대한 각 이진수 시퀀스(^{i})는 하나 이상의 이진수(b^i_k)로 구성된다. 디코더 측에서, 심벌(s_i)은 이진수 시퀀스($\mathbf{b}^i$)인 경우의 역 매핑($s_i=(\gamma^i_b)^{-1}(\mathbf{b}^i)$)에 의해 재구성될 수 있다. 이진화의 결과로, 소스 심벌 시퀀스($\{s_0,\ldots,s_{N-1}\}$)를 나타내는 이진수 시퀀스($\{b_0,\ldots b_{B-1}\}$)를 얻는다.

[0197] 모든 이진수(b_j)는 동일한 이진 알파벳($B=\{0,1\}$)과 관련되지만, $p^j_1=p^j_0$인 대응하는 이진 pmf($\{p^j_0, p^j_1\}$)는 보통 상이하다. 이진 pmf($\{p^j_0, p^j_1\}$)는 더 적은 확률의 이진수(LPB: Less Probable Bin) 값(b^j_{LPB}) 및 그 확률($p^j_{LPB}(P^j_{LPB}\leq 0.5)$)에 의해 기재될 수 있다. 이 이진수 확률 기술($\{b^j_{LPB}, p^j_{LPB}\}$)은 이진화 매핑(γ^i_b)인 경우에 심벌 알파벳에 대한 확률 추정($p_s(a^i_m)$)으로부터 직접 유도할 수 있다. 인코더 및 디코더 측에서 즉시 ($\{b^j_{LPB}, p^j_{LPB}\}$)을 직접 추정하는 것도 가능하다(그렇게 하는 것이 종종 바람직하다). 그러므로, 이진수는 신택스 및 이전에 코딩된 심벌이나 이진수를 기반으로 한 확률 모델(컨텍스트로도 지칭됨)과 관련될 수 있다. 또한 각 확률 모델의 경우, 확률 기술($\{b^j_{LPB}, p^j_{LPB}\}$)은 확률 모델로 코딩된 이진수의 값을 기반으로 하여 추정될 수 있다. 그러한 이진수 확률 모델링의 예는 H.264의 CABAC에 관해 기재된다.

[0198] 이진수 엔트로피 함수,

[0199] $$H(p) = -p\log_2(p) - (1-p)\log_2(1-p) \qquad \text{(B3)}$$

[0200] 는 p=0.5를 중심으로 대칭이므로, 동일한 이진 코더가, b^j_{LPB}의 값과는 독립적으로 동일한 LPB 확률(p^j_{LPB})과 관련된 모든 이진수를 코딩하는데 사용될 수 있다. 그러므로 이진수 시퀀스($\{b_0,\ldots b_{B-1}\}$)는 코딩 이진수 시퀀스($\{b^c_0,\ldots,b^c_{B-1}\}$)로 전환된다. 각 이진수($b_j$)의 경우, 대응하는 바이젝티브 매핑($\gamma^i_C$)은

[0201] $$b_j^c = \gamma_c^j(b_j) = b_j \oplus b_{LPB}^j \qquad \text{(B4)}$$

[0202] 으로 명시된다. 여기서 $\oplus$는 배타적 논리곱 연산자를 나타낸다. 디코더 측에서, 이진수(b_j)는 역 매핑($b_j=(\gamma_C^i)^{-1}(b_j^c)=b_j^c \oplus b_{LPB}^j$)에 의해 코딩 이진수($b_j^c$) 및 대응하는 LPB 값($b_{LPB}^j$)에 의해 재구성될 수 있다. 코딩 이진수($b_j^c=0$)는 대응하는 이진수($b_j$)의 값이 LPB 값($b_{LPB}^j$)과 동일함을 명시하며, 코딩 이진수($b_j^c=1$)는 대응하는 이진수($b_j$)의 값이 더 큰 확률의 이진수(MPB: More Probable Bin) 값($1-b_{LPB}^j$)과 같음을 명시한다.

[0203] 코딩 이진수 시퀀스($\{b_0^c,\ldots,b_{B-1}^c\}$)는 고유하게 소스 심벌 시퀀스($\{s_0,\ldots,s_{N-1}\}$)를 나타내며, 엔트로피 코딩에 사용될 수 있는 대응하는 확률 추정은 LPB 확률(p_{LPB}^j($p_{LPB}^j \le 0.5$임)에 의해 완벽하게 기재된다. 그러므로, 절반-개방된 간격((0,0.5])에서의 확률만이 코딩 이진수(b_i^c)에 대한 이진수 엔트로피 코더를 설계하는데 고려될 필요가 있다.

[0204] 실제 이진 엔트로피 코딩의 경우, 코딩 이진수 시퀀스($\{b_0^c,\ldots,b_{B-1}^c\}$)는 작은 수의 확률 간격($I_k$)에 투영된다. LPB 확률 간격((0,0.5])은 K개의 간격($Ik=(p_k, p_{k+1}]$)으로 분할된다.

[0205] $$\bigcup_{k=0}^{K-1} I_k = (0,0.5] \quad and \quad I_k \bigcap I_j = \varnothing \ for\ k \neq j \qquad \text{(B5)}$$

[0206] K개의 간격 세트는 K-1개의 간격 경계(p_k)(여기서, k=1,...,K-1)를 특징으로 한다. 일반성 손실이 없다면, 우리는 k=0,...K에 대해 $p_k<p_{k+1}$이라고 가정한다. 외부 간격 경계는 $p_0=0$ 및 $p_k=0.5$로 정해지고 주어진다. 간단한 비-적응성 이진 엔트로피 코더가 각 간격 I_k에 대해 설계된다. 관련된 LPB 확률($p_{LPB}^j \in I_k$)을 갖는 모든 코딩 이진수(b_j^c)가 간격(I_k)에 할당되어 대응하는 고정된 엔트로피 코더로 코딩된다.

[0207] 다음의 기재에서, 모든 이진수는 코딩 이진수(b_j^c)를 나타내며, 모든 확률(p)은 LPB 확률(p_{LPB}^j)이다.

[0208] 확률 간격 이산화의 코딩 효율에 대한 영향을 조사하기 위해, 우리는, 엔트로피 경계를 달성하는 고정된 확률에 대한 최적 엔트로피 코더를 설계할 수 있다고 가정한다. 각 확률 간격($I_k=(p_k, p_{k+1}]$)은 대표적인 확률($p_{Ik} \in I_k$)과 관련되며, 대응하는 최적 엔트로피 코더는 이 대표적인 확률에 대한 엔트로피 한계를 달성해야 한다. 이들 가정하에서, 간격 대표값(p_{Ik})에 대한 최적의 엔트로피 코더를 사용하여 확률(p)을 갖는 이진수를 코딩하기 위한 율은 다음과 같이 주어진다.

[0209] $$\begin{aligned} R(p,p_{I_k}) &= -p\log_2(p_{I_k}) - (1-p)\log_2(1-p_{I_k}) \\ &= H(p_{I_k}) + (p-p_{I_k})\,H'(p_{I_k}) \end{aligned} \qquad \text{(B6)}$$

[0210] 여기서 H(p)는 이진수 엔트로피 함수 3을 나타내며,

[0211] $$H'(p) = \log_2\left(\frac{1-p}{p}\right) \qquad \text{(B7)}$$

[0212] 이 그 제1 도함수이다. 우리는 또한 간격((0,0.5])에서의 확률의 분포가 f(p)($\int_0^{0.5} f(p)dp = 1$)로 주어진다고 가정한다. 그러면, 예상된 율은, 대응하는 대표값 확률($\{p_{Ik}\}$)을 갖는 K개의 간격($\{I_k\}$)의 소정의 세트에

대해, 이진수 당 비트 단위로, 다음과 같이 기재될 수 있다.

$$R = R(\{I_k\},\{p_{I_k}\}) = \sum_{k=0}^{K-1}\left(\int_{p_k}^{p_{k+1}} R(p,p_{I_k})\, f(p)\, dp\right) \quad \text{(B8)}$$

[0213]

[0214] 임의의 대표적인 확률(p_{Ik})(k=0,...,k-1)에 대한 제1 부분 도함수는 다음과 같이 주어진다.

$$\frac{\partial}{\partial p_{I_k}} R = \frac{p_{I_k}\int_{p_k}^{p_{k+1}} f(p)\, dp - \int_{p_k}^{p_{k+1}} p\, f(p)\, dp}{p_{I_k}(1-p_{I_k})\ln 2} \quad \text{(B9)}$$

[0215]

[0216] 수학식($\frac{\partial}{\partial p_{I_k}} R = 0$)은 정의(I_k)의 영역 내부의 대표적인 확률(p_{Ik})에 대해 단일 해,

$$p^*_{I_k} = \frac{\int_{p_k}^{p_{k+1}} p\, f(p)\, dp}{\int_{p_k}^{p_{k+1}} f(p)\, dp} \quad \text{(B10)}$$

[0217]

[0218] 를 갖는다. 이 해에 대한 제2 부분 도함수

$$\frac{\partial^2}{\partial p_{I_k}^2} R(p^*_{I_k}) = \frac{\int_{p_k}^{p_{k+1}} f(p)\, dp}{p^*_{I_k}(1-p^*_{I_k})\ln 2} \quad \text{(B11)}$$

[0219]

[0220] 는

$$\int_{p_k}^{p_{k+1}} f(p)\, dp > 0 \quad \text{(B12)}$$

[0221]

[0222] 인 경우, 항상 0보다 크다. 그러므로 조건(B12)이 충족되는 경우, 수학식 B10에서 주어진 값(p^*_{Ik})은 간격 경계(p_k 및 p_{k+1})에 의한 예상 전체 율(R)을 최소화하는 간격(I_k)에 대한 대표적인 확률이다. 그렇지 않다면, 이진수는 간격(I_k)에 투영되지 않으며, 대표적인 확률($p_{Ik}\in I_k$)은 전체 율(R)에 대한 영향이 없이 임의적으로 선택될 수 있지만, 그러한 구성은, 간격(I_k)이 엔트로피 코딩을 위해 사용될 수 없을 것이므로, 회피되어야 한다.

[0223] 최적 간격 경계에 대한 조건을 찾기 위해, 우리는 간격 경계(p_k)(k=1,...K-1)에 대해 예상 전체 율(R)의 제1 도함수를 조사한다. 모든($p\in[p_{Ik-1}, p_{Ik}]$)에 대해 f(p)>0인 경우, 수학식($\frac{\partial}{\partial p_k} R = 0$)은 정의($[p_{Ik-1}, p_{Ik}]$)의 영역 내부의 간격 경계(p_k)에 대해 단일 해,

$$p^*_k = \frac{H(p_{I_k}) - p_{I_k}H'(p_{I_k}) - H(p_{I_{k-1}}) + p_{I_{k-1}}H'(p_{I_{k-1}})}{H'(p_{I_{k-1}}) - H'(p_{I_k})} \quad \text{(B13)}$$

[0224]

[0225] 를 가지며, 이 해에 대한 제2 부분 도함수,

$$\frac{\partial^2}{\partial p_k^2} R(p^*_k) = f(p^*_k)\left(H'(p_{I_{k-1}}) - H'(p_{I_k})\right) \quad \text{(B14)}$$

[0226]

[0227] 는 항상 0보다 커서, p^*_k는 간격 대표값(p_{Ik-1} 및 p_{Ik})에 의해 예상 전체 율(R)을 최소화하는 내부 경계($p \in [p_{Ik-1}, p_{Ik}$))이다. f(p)=0에서 확률($[p_{Ik-1}, p_{Ik}$))이 존재한다면, 수학식($\frac{\partial}{\partial p_k} R = 0$)은 복수의 해를 갖지만, 추가 최적 해가 존재할 수 있더라도 수학식 B13에서 주어진 p^*_k는 여전히 최적이다.

[0229] *간격의 수(K)와 확률 분포(f(p))인 경우, 예상 전체 율(R)을 최소화하는 간격 대표값(p_{Ik})(k=0,...K-1)과 간격 경계(p_k)(k=1,...,K-1)는 k=0,...,K-1에 대해 조건 B12 하에서 수학식 B10 및 B13에 의해 주어진 수학식의 해를 구함으로써 얻을 수 있다. 이것은 다음의 반복적 알고리즘을 통해 달성할 수 있다.

[0230] 알고리즘 1:

[0231] 1) 모든 k=0,...K-1에 대해 조건 B12을 따르는 방식으로 모든 k=0,...,K-1에 대해 간격((0,0.5])을 K개의 임의의 간격($I_k=(p_k, p_{k+1}]$)($p_0=0$, $p_K=0.5$ 및 $p_k<p_{k+1}$)으로 분할.

[0232] 2) 수학식 B10에 따라 대표값(p_{Ik})(k=0,...,K-1)을 업데이트.

[0233] 3) 수학식 B13에 따라 간격 경계(p_k)(k=1,...,K-1)를 업데이트.

[0234] 4) 수렴할 때까지 이전 두 단계를 반복

[0235] 도 14는 기술한 알고리즘을 사용한 최적 간격 이산화의 예를 도시한다. 이 예에서, 우리는 $0<p\leq 0.5$에 대한 균일한 확률 분포(f(p)=2)를 가정하였고, 확률 간격(0,0.5])을 K=4개의 간격으로 분할했다. 확률 간격 이산화가 모든 $p \in (0,0.5]$에 대해 이진 엔트로피 함수(H(p))($A(p) \geq H(p)$)의 피스와이즈 선형 근사화(A(p))를 야기함을 알 수 있다.

[0236] 코딩 효율에 대한 간격 이산화의 영향에 대한 측정치로서, 엔트로피 한계에 대한 예상 전체 율 증가,

$$\bar{\rho} = \frac{R}{\int_0^{0.5} H(p)\, f(p)\, dp} - 1 \qquad \text{(B15)}$$

[0237]

[0238] 가 사용될 수 있다. 도 14의 실제 예의 경우, 엔트로피의 예상 값($\overline{H} \int_0^{0.5} H(p) f(p) dp$)은 1/(2ln2) 비트/이진수와 같고, 율 오버헤드($\overline{\rho}$)는 1.01%와 같다. 표 4는 선택된 수의 간격(K)에 대해, 균일한 확률 분포 및 선형 증가 확률 분포(f(p)=8p)($p \in (0,0.5]$)에 대한 율 오버헤드($\overline{\rho}_{uni}$ 및 $\overline{\rho}_{lin}$)를 게재한다.

표 4

[0239]

K	1	2	4	8	12	16
$\overline{\rho}_{uni}$ [%]	12.47	3.67	1.01	0.27	0.12	0.07
$\overline{\rho}_{lin}$ [%]	5.68	1.77	0.50	0.14	0.06	0.04

[0240] 표 4: 균일한 및 선형 증가 확률 분포에 대한 율 오버헤드 대 확률 간격 수

[0241] 본 절의 조사는, LPB 확률 간격((0,0.5])의 고정된 확률을 갖는 작은 수의 간격(예컨대 8 내지 10개의 간격)으로의 이산화가 코딩 효율에 매우 작은 영향을 줌을 보여준다.

[0242] 확률 간격에 대한 전술한 엔트로피 코딩으로 인해 고정된 확률을 사용한 개별 코더가 가능하게 된다.

[0243] 다음의 설명에서, 우리는 먼저 간단한 코드가 어떻게 고정된 확률에 대해 설계될 수 있는지를 설명한다. 이들 결과를 통해, 우리는 코드 설계 및 LPB 확률 간격((0,0.5])의 분할을 함께 최적화하는 알고리즘을 개발한다.

[0244] 고정된 확률($p=p_{lk}$)에 대한 엔트로피 코딩은 산술 코딩이나 가변 길이 코딩을 사용하여 행해질 수 있다. 후자의 경우, 다음의 접근법이 간단하고 매우 효율적일 것이다.

[0245] 우리는, 가변 비트 수가 가변 길이 코드워드에 매핑되게 하는 이진 엔트로피 코딩 방식을 고려한다. 고유 디코딩 가능성을 위해, 코드워드의 이진수 시퀀스로의 역 매핑은 고유해야 한다. 또한, 우리는 가능한 엔트로피 한계에 가까이 접근하는 코드를 설계하기 원하므로, 우리의 고려를 바이젝티브 매핑으로 제한한다. 그러한 바이젝티브 매핑은, 도 15에 도시한 바와 같이, 모든 리프(leaf) 노드가 코드워드와 관련되는 이진 트리에 의해 표현할 수 있다. 트리 에지는 이진 이벤트를 나타낸다. 도 15의 예에서, 하부 에지는 LBP 이진수 값을 나타내며, 상부 에지는 MPB 이진수 값을 나타낸다. 이진 트리는, 이것이 완전히 이진 트리라면, 즉 모든 노드가 리프이거나 두 개의 후손(descendant)을 갖는다면 이진수에 대한 프레픽스 코드를 나타낸다. 각 리프 노드는 주어진 LPB 확률(p)을 기반으로 한 확률과 관련된다. 루트 노드는 확률($p_{root}=1$)을 갖는다. 모든 다른 노드의 확률은 대응하는 조상(ancestor)의 확률을 LPB 후속에 대해서는 p와 그리고 MPB 후손에 대해서는 q=1-p와 곱함으로써 얻는다. 각 리프 노드(L_l)는 루트 노드로부터 리프 노드로의 LBP 에지의 수(a_l)와 MPB 에지의 수(b_l)를 특징으로 한다. 특정한 LPB 확률(p)에 대해, 리프 노드($L_l=\{a_l, b_l\}$)에 대한 확률(p_l)은

[0246] $$p_l = p^{a_l}(1-p)^{b_l} \qquad \text{(B16)}$$

[0247] 과 같다. 이진 트리(T)는 완전히 리프 노드의 수(L)와 관련 쌍($\{(a_l, b_l\})(l=0,\ldots,L-1)$의 수를 특징으로 한다.

[0248] 완전 이진 트리(T)와 LPB 확률(p)이라면, 코드워드의 리프 노드로의 최적의 할당은 허프만 알고리즘에 의해 얻을 수 있다. 결과적인 가변수의 비트의 가변 길이 코드워드(V2V)로의 매핑(C)은 리프 노드의 개수와 동일한 코드워드 개수(L)와, $l=0,\ldots,L-1$에 대한 튜플($\{a_l, b_l, l_l\}$)(l_l은 대응하는 리프 노드($L_l=\{a_l, b_l\}$)과 관련되는 코드워드 길이를 나타냄)을 특징으로 한다. 코드워드 길이($\{l_l\}$)라면 코드워드 할당에 대해 복수의 가능성이 있으며, 실제 코드워드 할당은, 코드워드가 고유하게 디코딩 가능한 프레픽스 코드를 나타내는 한, 중요하지 않다. 소정의 코드(C)와 LPB 확률(p)에 대한 비트/이진수 단위의 예상 율(R(p, C)은 예상 코드워드 길이와 코드워드당 예상 이진수의 수의 비이다.

[0249] $$R(p,C) = \frac{\sum_{l=0}^{L-1} p_l\, l_l}{\sum_{l=0}^{L-1} p_l (a_l + b_l)} = \frac{\sum_{l=0}^{L-1} p^{a_l}(1-p)^{b_l}\, l_l}{\sum_{l=0}^{L-1} p^{a_l}(1-p)^{b_l}(a_l + b_l)} \qquad \text{(B17)}$$

[0250] 코딩 설계는 종종 코드워드의 최대 개수(L), 코드워드 당 최대 이진수 수, 또는 최대 코드워드 길이와 같은 인자에 의해 제한되거나, 특정한 구조(예컨대, 최적 파싱을 허용하기 위한)의 코드로 제한된다. 우리가 특정한 응용에 대한 유용한 코드의 세트(S_C)가 주어진다고 가정하면, 특정한 LPB 확률(p)에 대한 최적 코드($C^* \in S_C$)는 예상 율(R(p,C))을 최소화함으로써 찾을 수 있다.

[0251] $$C^*(p) = \arg\min_{\forall C \in S_C} R(p,C) \qquad \text{(B18)}$$

[0252] 더 빠른 대안으로서, 최소화는 또한 소정의 이진수 트리 세트(S_T)보다 선행할 수 있으며, 각 트리에 대해, 허프만 알고리즘에 의해 얻은 단 하나의 V2V 코드(C)를 고려한다. 예로서, 우리는 리프 노드의 개수(L)가 소정의 최대치(L_m) 이하인 모든 이진 트리(T)를 고려함으로써 여러 LPB 확률(p)에 대한 V2V 코드를 설계했다. 도 16에서, 상대적인 율 증가($\rho(p,C^*(p)) = R(p,C^*(p))/H(p)$)는 선택된 최대 표 크기($L_m$)에 대한 LPB 확률(p)에 대해 도시된다. 율 증가($\rho(p)$)는 더 큰 표 크기를 허용함으로써 보통 감소할 수 있다. 더 큰 LPB 확률

의 경우, 8 내지 16개의 코드워드의 작은 표 크기(L)가 율 증가(ρ(p))를 합리적으로 작게 유지하기에는 보통 충분하지만, 더 작은 LPB 확률(예컨대, p<0.1)에 대해 더 큰 표 크기(L)를 필요로 한다.

[0253] 이전 절에서, 우리는 고정된 LPB 확률에 대한 최적 코드 및 코드 설계를 갖는 최적 확률 이산화를 고려했다. 그러나 일반적으로, 우리는 제한된 표 크기의 실제 V2V 코드를 갖는 엔트로피 한계를 달성할 수 없으므로, LPB 확률 간격((0,0.5])의 코드 설계 및 분할은 함께 최적화된 엔트로피 코딩 설계를 얻는데 고려되어야 한다.

[0254] 소정의 간격($I_k=(p_k,p_{k+1}]$)에 대해, 소정의 세트(S_C)의 코드(C_k)는, 이것이 소정의 간격에 대한 예상 율

$$R=\int_{p_k}^{p_{k+1}} R(p,C_k)f(p)dp$$

( )을 최소화한다면, 최적의 코드(C^*_k)이다.

$$C_k^* = \arg\min_{\forall C_k \in S_C} \int_{p_k}^{p_{k+1}} R(p,C_k)\, f(p)\, dp \qquad \text{(B19)}$$

[0255]

[0256] 실제 설계의 경우, 수학식 B10에 따라 간격(I_k)에 대한 최적의 대표 확률(p^*_{Ik})을 먼저 결정한 후, 수학식 B18에 따라 대표 확률(p^*_{Ik})에 대해 소정의 세트(S_C)에 대한 최적 코드(C^*_k)를 선택함으로써, 코딩 효율에 작은 영향을 미치면서, 수학식 B19에서의 적분의 최소화를 간략화할 수 있다.

[0257] 코드 세트(C_k)(k=0,...,K-1)라고 할 경우, 최적 간격 경계(p_k)(k=1,...,K-1)는 예상 전체 율을 최소화함으로써 유도할 수 있다.

$$R = R(\{p_k\},\{C_k\}) = \sum_{k=0}^{K-1}\left(\int_{p_k}^{p_{k+1}} R(p,C_k)\, f(p)\, dp\right) \qquad \text{(B20)}$$

[0258]

$$\frac{\partial}{\partial p_k} R = 0$$

[0259] 제1 도함수를 0인 간격 경계에 대해 ( )(k=1,...,K-1)로 설정하면, 다음을 산출한다.

$$p_k^* = p_k \quad with \quad R(p_k,C_{k-1}) = R(p_k,C_k) \qquad \text{(B21)}$$

[0260]

[0261] 수학식 B13에서와 유사하게, p^*_k는 항상 최적 해이지만, 확률 분포(f(p))에 따라서, 다른 최적 해가 존재할 수도 있음을 알 수 있다. 그러므로 각각 주어진 관련 코드(C_{k-1} 및 C_k)를 갖는 두 개의 간격(I_{k-1} 및 I_k) 사이의 최적 간격 경계(p^*_k)는 함수(R(p, C_{k-1}) 및 R(p, C_k))의 교차점이다.

[0262] 결국, 다음의 인터랙티브 알고리즘을 확률 간격의 개수(K)라고 할 때의 확률 간격 분할 및 관련 코드, 가능 코드(S_C)의 세트, 및 확률 분포(f(p))(p∈(0, 0.5])를 함께 유도하기 위해 사용될 수 있다.

[0263] 알고리즘 2:

[0264] 1) 3절에서 명시한 알고리즘 1을 사용하여 초기 확률 간격 경계(p_k)(k=0,...,K)를 유도.

[0265] 2) 수학식 B10에 따른 확률 간격(I_k)(k=0,...,K-1)에 대한 대표값(p_{Ik}) 유도.

[0266] 3) 수학식 B18에 따른 간격 대표값(p_{Ik})(k=0,...,K-1)에 대한 코드($C_k \in S_C$) 유도.

[0267] 4) 수학식 B21에 따른 간격 경계(p_k)(k=1,...,K-1)를 업데이트.

[0268] 5) 수렴할 때까지 이전 세 개의 단계를 반복.

[0269] 알고리즘 2의 단계 2 및 3은 또한 수학식 B19에 따른 간격 경계(p_k)(k=0,...,K)를 기반으로 하여 코드((C_k∈

S_C)(k=1,...,K)의 직접 미분으로 교체할 수 도 있다. 또한, 4.1절에서 언급한 바와 같이, 단계 3의 최소화는 또한 소정의 이진 트리 세트(S_T)보다 먼저 진행할 수 있으며, 이 세트에서 각 이진 트리(T)에 대해, 허프만 알고리즘에 의해 얻은 단 하나의 V2V 코드(C_k)를 고려한다.

[0270] 예로서, 우리는 알고리즘 2를 사용하여 K=12개의 확률 간격과 대응하는 V2V 코드로의 분할을 함께 유도하였다. 여기서, 알고리즘의 단계 3의 최소화는 소정의 이진 트리 세트(S_T)에 대해 등가의 최소화로 교체하였고, 이 세트에서 각 트리(T)에 대한 평가된 코드(C)는 허프만 알고리즘에 의해 얻었다. 우리는 최대 수(L_m=65)의 리프 노드를 갖는 트리(T) 및 그에 따라 최대 65개의 표 엔트리를 갖는 노드(C)를고려했다. 최대 16개의 리프 노드를 갖는 모든 이진 트리(T)는 최소화 시 평가하였다. 16개보다 많은 리프 노드를 갖는 트리의 경우, 우리는 더 작은 수의 리프 노드를 갖는 트리에 대한 최상의 결과를 갖는 차선의 검색을 사용했다.

[0271] 도 17에서, 코드 설계 예에 대한 엔트로피 한계($\Delta R(p)=R(p)-H(p)$)에 대한 예상 율 증가를 LPB 확률(p)에 대해 도시한다. 비교로서, 우리는 또한 이론적으로 최적인 확률 간격 이산화(3절에서 개발됨)와 추가 제한(p_{Ik-1}=0.5)을 갖는 이론적으로 최적인 확률 간격 이산화에 대한 예상 율 증가(ΔR)를 이 도면 내에 도시하였다. 공동의 확률 간격 이산화 및 V2V 코드 설계가 간격 경계의 시프트를 야기함을 알 수 있다(간격 경계(p_k)(k=1,...,K-1)가 $\Delta R(p)$ 곡선의 국지적 최대치로 주어진다). 실제 V2V 코드를 갖는 설계 예에 대한 엔트로피 한계에 대한 상대적인 예상 전체 율 증가는, 균일한 확률 분포(f(p))를 가정할 때, $\overline{\rho}$=0.24%이다. 이론적으로 최적인 확률 간격 이산화 및 추가 제한(p_{Ik-1})=0.5)을 갖는 이론적으로 최적인 확률 이산화에 대한 대응하는 상대 율 증가는 각각 $\overline{\rho}$=0.12% 및 $\overline{\rho}$=0.13%이다.

[0272] 코드워드 종료는 다음과 같이 행해질 수 있다. 유한 심벌 시퀀스($\{s_0,...,s_{N-1}\}$)를 코딩할 때 K개의 이진수 코더 각각은 유한 코딩 이진수 시퀀스($\mathbf{b}^c_k=\{b^c_0,...,b^c_{Bk-1}\}_k$)(k=0,...,K-1)를 처리한다. 또한, K개의 이진 인코더 각각에 대해, 모든 코딩 이진수 시퀀스($\mathbf{b}^c_k=\{b^c_0,...,b^c_{Bk-1}\}_k$)를 코드워드나 코드워드 시퀀스($\mathbf{c}_k(\mathbf{b}^c_k)$)가 주어질 때 재구성할 수 있음이 보장되게 되었다.

[0273] 산술 코딩을 사용할 때, 코딩 이진수의 시퀀스에 대한 산술 코드워드는, 모든 코딩 이진수가 코드워드에 따라 디코딩될 수 있는 방식으로 종료되어야 한다. 전술한 V2V 코드의 경우, 시퀀스($\mathbf{b}^c_k$)의 끝에서의 이진수는 코드워드와 관련된 이진수 시퀀스를 나타낼 수 없다. 그런 경우, 프레픽스로서 나머지 이진수 시퀀스를 포함하는 임의의 코드워드는 기록될 수 있다. 오버헤드는, 최소 길이를 갖는 대응하는 코드워드(또는 이들 코드워드 중 하나)가 선택되는 경우, 최소화될 수 있다. 디코더 측에서, 비트스트림 신택스 및 이진화 방식에 의해 식별될 수 있는 이진수 시퀀스의 끝에서의 추가 판독 이진수는 버린다.

[0274] 간단한 코드 설계 예를 이후에 제공한다. 예시용으로 우리는 세 개의 문자와 고정된 관련 확률($p_s(a_0)$=0.7, $p_s(a_1)$=0.18 및 $p_s(a_2)$=0.12)의 소스($\{s\}$)의 간단한 예를 고려한다. 대응하는 삼진 선택 트리를 도 18에 도시된 바와 같이 완전 이진수 트리로 전환할 수 있다.

[0275] 도 18의 완전 이진수 트리에 대한 이진화를 표 5에 제공한다. 삼진 심벌 $pmf(p_s)$는 두 개의 이진수 $pmf(p_{b0}=(0.7, 0.3)$와 $p_{b1}=(0.6, 0.4))$로 전환한다. 비트스트림에서의 각 심벌(s)의 경우, 이진수(b_0)가 존재한다. b_0가 0일 때, b_1이 또한 존재한다. 표 2에 제공된 이진화는 소스(s)에 대한 최적 단일-문자 허프만 코드와 동일함을 주목해야 한다.

표 5

[0276]

심벌(a_i)	확률($p(a_i)$)	이진수(b_0)	이진수(b_1)
a_0	0.7	1	
a_1	0.18	0	1

a_2	0.12	0	0
LPB 확률	$p_{LPB}=p(b_j=0)$	0.3	0.4

[0277] 표 5: 세 개의 문자 소스의 이진화. LPB 확률(p_{LPB})은 제1 이진수에 대해서는 0.3이고, 제2 이진수에 대해서는 0.4이다.

[0278] 소스(s)에 대한 엔트로피는

[0279] $$H(0.7,0.18,0.12) = H(0.7,0.3) + 0.3\,H(0.6,0.4)$$
$$= 1.1726\ bit/symbol \qquad \text{(B22)}$$

[0280] 이다. 단일-문자 허프만 코드의 평균 코드워드 길이는 ρ_{HC}=0.1274비트/심벌의 중복도 또는 10.87%의 예상 율 오버헤드에 대응하는 다음과 같이 제공된다.

[0281] $$\bar{\ell}_{HC} = \sum_{i=0}^{M-1} p_i \ell_i^{HC} = 1.3\ bit/symbol \qquad \text{(B23)}$$

[0282] 고정된 pmf를 갖는 특정 이진화 예의 경우, 이진수(b_0 및 b_1)는 이미 코딩 이진수를 나타내며, 이는 두 이진수에 대해 LPB 값(b^j_{LPB})은 0이기 때문이다. LPB 확률의 분포(f(s))는 이산적이며, p=0.3 및 p=0.4를 제외하고는 f(p)=0이다. 결국, 최적의 확률 이산화는 대표값(p_{I0}=0.3 및 p_{I1}=0.4)을 갖는 K=2개의 간격을 야기한다. 이들 간격 사이의 간격 경계(p_1)는 [0.3, 0.4)에서 임의로 선택할 수 있다.

[0283] 소스를 인코딩할 경우, 소스 심벌의 시퀀스는 이진수 시퀀스로 이진화된다. 이진수(b_0)는 매 소스 심벌마다 송신되다. 이진수(b_1)는 b_0=0일 때만 송신된다. 이진수(b_0 및 b_1)는 각각 일정한 LPB 확률(P_{I0}=0.3 및 P_{I1}=0.4)로 개별적으로 코딩된다.

[0284] 고정된 확률을 갖는 이진수 알파벳의 효율적인 코딩은 간단한 V2V 매핑에 의해 달성할 수 있다. LPB 확률(p_{LPB}=0.3 및 p_{LPB}=0.4)에 대한 작은 코딩 표를 갖는 V2V 매핑에 대한 예를 각각 표 6 및 표 7에 제공한다. p_{LPB}=0.3에 대한 V2V 매핑은 0.0069bit/bin, 즉 0.788%의 중복도를 산출한다. p_{LPB}=0.4의 LPB 확률의 경우, 중복도는 0.0053bit/bin, 즉 0.548%이다.

표 6

[0285]

이진수 트리	확률	코드
'11'	0.7^2=0.49	'1'
'01'	0.7 · 0.3=0.21	'01'
'0'	0.3	'00'

[0286] 표 6: p_{LPB}=0.3인 LPB 확률에 대한 이진수 트리 및 코드. 이 코드의 중복도는 0.788%이다.

표 7

[0287]

이진수 트리	확률	코드 트리
'111'	0.6^3=0.216	'11'
'110'	0.6^2 · 0.4=0.144	'001'
'10'	0.6 · 0.4=0.24	'11'
'01'	0.4 · 0.6=0.24	'01'
'00'	0.4^2=0.16	'000'

[0288] 표 7: p_{LPB}=0.4인 LPB 확률에 대한 이진수 트리 및 코드. 이 코드의 중복도는 0.548%이다.

[0289] 새 코딩 방법에 의해 초래되는 전체 예상 율은

[0290] $$\bar{\ell}_{NC} = \bar{\ell}_{b_0} + 0.3 \cdot \bar{\ell}_{b_1} = 1.181 \; bit/symbol \qquad \text{(B24)}$$

[0291] 이다. 전체 중복도는 엔트로피 한계에 비해 0.73%이며, 이것은 단일-문자 허프만 코드와 비교할 때 상당한 개선을 보인다.

[0292] 유사한 코딩 효율 개선을, 런-렝쓰 코드를 만듦으로서 얻을 수 있다고 주장할 수 있다. 상기 예의 경우, 우리는 최대 2개의 심벌의 런을 고려함으로써 가장 큰 확률 심벌에 대한 런-렝쓰 코드를 만들 수 있었다. 이벤트({a_0a_0, a_0a_1, a_0a_2, a_1, a_2}) 각각은 별도의 코드워드와 관련될 것이다. 그러한 코드는 엔트로피 한계에 비해 1.34%의 중복도를 산출한다. 실제로, V2V 코드는 이진수 심벌에 대한 런-렝쓰 코드의 일반화로서 고려될 수 있다(표 3의 V2V 코드는 런-렝쓰 코드를 효과적으로 나타낸다). 고정된 확률의 단일 심벌 알파벳의 경우, 제공했던 접근법에 대해서와 유사한 코딩 효율을 또한, 가변 수의 소스 심벌을 가변 길이 코드워드에 매핑하는 코드를 만듦으로써 달성할 수 있다. 제공했던 접근법의 주된 장점은, 고정된 또는 적응적 확률 추정을 갖는 임의의 소스 심벌 시퀀스를 고정된 LPB 확률로 동작하는 작은 수의 간단한 이진수 코더에 매핑할 때의 그 융통성이다.

[0294] *고유한 디코딩 가능성을 달성하는 방법을 다음에 고려한다.

[0295] 제공했던 엔트로피 코딩 방식으로, 소스 심벌 시퀀스($s=\{s_0,\ldots,s_{N-1}\}$)의 코딩은 다음의 세 개의 기본 단계로 구성된다:

[0296] ● 이진수 시퀀스($\mathbf{b}=\{b_0,\ldots,b_{B-1}\}$)를 산출하는 심벌 이진화($\mathbf{b}=\{b_0,\ldots,b_{B-1}\}=\gamma_b(s)$)

[0297] ● 코딩 이진수 시퀀스($\mathbf{b}^c=\{b^c_0,\ldots,b^c_{B-1}\}=\gamma_c(\mathbf{b})$)로의 이진수 시퀀스의 전환

[0298] ● 확률 간격 이산화 및 K개의 고정된 이진 코더를 사용한 코딩 이진수 시퀀스($\mathbf{b}^c=\{b^c_0,\ldots,b^c_{B-1}\}$)의 이진수 엔트로피 코딩.

[0299] 심벌 시퀀스(($\mathbf{s}=\{s_0,\ldots,s_{N-1}\}$))는, 코딩 이진수 시퀀스($\mathbf{b}^c=\{b^c_0,\ldots,b^c_{B-1}\}$)가 고유하게 디코딩 가능하며 매핑($\gamma_b$ 및 γ_c)을 바꿀 수 있다면, 고유하게 디코딩 가능하다.

[0300] γ_e는 하나 이상의 코딩 이진수 시퀀스($\mathbf{b}^c=\{b^c_0,\ldots\}$)의 하나 이상의 코드워드 시퀀스($\mathbf{c}(\mathbf{b}^c)=\{c_0,\ldots\}$)에의 매핑을 인코더에게 알려준다.

[0301] $$c(\boldsymbol{b}^c) = \gamma_e(\mathbf{b}^c) \qquad \text{(B25)}$$

[0302] 코드워드 시퀀스($\mathbf{c}(\mathbf{b}^c)$)에 의한 코딩 이진수 시퀀스($\mathbf{b}^c$)의 고유한 디코딩 가능성을 위해, 인코더 매핑(γ_e)은 고유한 코드워드($\mathbf{c}(\mathbf{b}^c)$)가 코딩 이진수의 각 가능 시퀀스($\mathbf{b}^c$)에 할당되는 속성을 가져야 한다:

[0303] $$\forall \boldsymbol{b}_i^c, \boldsymbol{b}_j^c \quad \boldsymbol{b}_i^c \neq \boldsymbol{b}_j^c \;\rightarrow\; c(\boldsymbol{b}_i^c) \neq c(\boldsymbol{b}_j^c) \qquad \text{(B26)}$$

[0304] 이 속성은, 산술 코드나 프레픽스 코드가 사용될 때 항상 충족된다. 이것은 특히 (4.3절에 기재한 코드워드 종료를 포함함) 4.1절에 기재된 V2V 코드에 대해 충족되며, 이는 V2V 코드가 가변 수의 이진수에 대한 프레픽스 코드를 나타내기 때문이다.

[0305] 그러나 제공했던 엔트로피 코딩 접근법에서, 코딩 이진수 시퀀스($\mathbf{b}^c$)는 K개의 서브-시퀀스(^c_k)(k=0,...,K-1),

[0306] $$\{b_0^c, \ldots, b_{K-1}^c\} = \gamma_p(\mathbf{b}^c) \qquad \text{(B27)}$$

[0307] 와 서브-시퀀스($\mathbf{b}^c_k$)의 각각에 분할되고, 코드워드 시퀀스($\mathbf{c}_k(\mathbf{b}_k^c)$는 특정한 인코더 매핑($\gamma^k_e$)을 사용하여 할당된다. 결국, 고유한 디코딩 가능성에 관한 조건은 확장되어야 한다. 코딩 이진수의 각 서브-시퀀스($\mathbf{b}_k^c$)가 대응하는 코드워드($\mathbf{c}_k(\mathbf{b}_k^c)$에 의해 디코딩 가능하고 분할 규칙($\gamma_p$)이 디코더에 알려져 있다면, 코딩 이진수 시퀀스($\mathbf{b}^c$)는 K개의 코드워드 시퀀스($\mathbf{c}_k(\mathbf{b}_k^c)$)(k=0,...,K-1)에 의해 고유하게 디코딩 가능하다. 분할 규칙(γ_p)은 코딩 이진수(b^c_j)(j=0,...,B-1)와 관련되는 LPB 확률(p^j_{LPB})과 LPB 확률 간격 이산화($\{I_k\}$)에 의해 주어진다. 그러므로 LPB 확률 간격 이산화($\{I_k\}$)는 디코더 측에서 알려져야 하고, 각 코딩 이진수(b^c_j)(j=0,...,B-1)에 대한 LPB 확률(p^j_{LPB})은 인코더 및 디코더 측에서 동일하게 유도되어야 한다.

[0308] 이진수 시퀀스의 코딩 이진수 시퀀스(^{c})로의 매핑(γ_c)을 위해, 각 단일 b_j(j=0,...,B-1)는 이진수 매핑($b^c_j = \gamma^j_c(b_j) = b_j \oplus b^j_{LPB}$)으로 전환된다. 디코더 측에서, 이진수 시퀀스는 이진수 매핑에 의해 유도될 수 있다.

[0309] $$b_j = (\gamma^j_c)^{-1}(b^c_j) = b^c_j \oplus b^j_{LPB} \qquad \text{(B28)}$$

[0310] 여기서 j=0,...,B-1이다. 각 이진수(b_j)에 대한 LPB 값(b^j_{LPB})이 인코더와 디코더 측에서 동일하게 유도된다면, 이들 매핑($\gamma^j_c)^{-1}$은 대응하는 인코더 매핑(γ^j_c)의 역을 나타내며, 이는

[0311] $$b^c_j \oplus b^j_{LPB} = b_j \oplus b^j_{LPB} \oplus b^j_{LPB} = b_j \oplus 0 = b_j \qquad \text{(B29)}$$

[0312] 이며, 그에 따라 이진수 시퀀스($\mathbf{b}$)의 코딩 이진수 시퀀스($\mathbf{b}^c$)로의 전환(γ_b)이 가역적일 수 있기 때문이다.

[0313] 끝으로, 우리는 이진화($\mathbf{b}=\gamma_b(\mathbf{s})$)의 가역성을 조사하며, 이 이진화에 의해 각 심벌(s_i)(i=0,...,N-1)은 이진수 시퀀스($\mathbf{b}^i=\gamma^i_b(s_i)$)에 매핑된다. 이진화 매핑($\gamma^i_b$)이 상이한 이진수 시퀀스($\mathbf{b}^j_m$)를 심벌($s_i$)에 대해 알파벳($A_i$)의 각 문자($a^i_m$)에 할당한다면, 심벌($s_i$)은 대응하는 이진수 시퀀스($\mathbf{b}^i$)에 의해 고유하게 디코딩될 수 있다. 그러나 이 조건은 충분한 것이 아니며, 이는 이진수 시퀀스($\mathbf{b}$={b0,...,b_{B-1})의 심벌(s_i)(i=0,...,N-1)에 대응하는 이진수 시퀀스($\mathbf{b}^i$)로의 분할이 디코더에 알려져 있지 않기 때문이다. 각 심벌(s_i)에 대해 대응하는 알파벳(A_i)의 문자(a^i_m)와 관련된 이진수 시퀀스($\mathbf{b}^j_m$)가 프레픽스 코드를 형성하고 각 심벌(s_i)(i=0,...,N-1)에 대한 이진화 매핑(γ^i_b)이 디코더 측에 알려져 있을 때 충분한 조건이 주어진다.

[0314] 제공된 엔트로피 코딩 접근법에 대한 고유한 디코딩 가능성의 조건은 다음과 같이 요약할 수 있다:

[0315] ● 이진화 매핑(γ^i_b)이 프레픽스 코드를 나타내며 (심벌 코딩 순서로) 디코더에 알려진다.

[0316] ● 모든 이진수(b_j)에 대한 확률 모델($\{b^j_{LPB}, p^j_{LPB}\}$)이 인코더와 디코더 측에서 동일하게 유도된다.

[0317] ● LPB 확률 간격((0,0.5])의 K개의 간격(I_k)(k=0,...,K-1)의 분할이 디코더에 알려져 있다.

[0318] ● 각 확률 간격(I_k)(k=0,...,K-1)에 대한 매핑(γ^k_e)은 고유하게 디코딩 가능한 코드를 나타낸다.

[0319] 다음의 설명에서, 우리는 전체 인코더 및 디코더 설계에 대한 예를 더 상세하게 기술할 것이다. 우리는, 이진수에 대한 확률 모델($\{b_{LPB}, p_{LPB}\}$)을 인코더 및 디코더 측에서 직접 추정하고 K개의 이진수 코더가 전술한 V2V 매

핑을 사용하는 코딩 방식을 주로 설명할 것이다. 각 소스 심벌(s)은 심벌 카테고리(c_s)와 관련되어야 하며, 이러한 카테고리(c_s)는 심벌의 값의 범위를 포함하는 심벌의 타입을 결정한다. 심벌 순서 및 관련 심벌 카테고리는, 인코더 및 디코더 측에 알려져 있는 것으로 가정되는 신택스에 의해 주어져야 한다.

[0320] 예시적인 PIPE 인코더와 PIPE 디코더 설계의 블록도를 도 19에 예시한다. 인코더 측에서, 관련 심벌 카테고리(c_s)를 갖는 심벌(s)이 *이진화기*에 공급되며, 이진화기는 각 심벌(s)을 이진수 시퀀스($b_s = \gamma_b^{c_s}(s)$)로 전환한다.

[0321] 사용된 이진화 방식($\gamma_b^{c_s}$)은 심벌 카테고리(c_s)를 기반으로 하여 결정한다. 게다가, 이진화기는 이진수 시퀀스($_s$)의 각 이진수(b)를, 이진수(b)를 코딩하는데 사용하는 확률 모델을 명시하는 확률 모델 지시(c_b)와 관련시킨다. 확률 모델 지시(c_b)는 심벌 카테고리(c_s), 이진수 시퀀스($_s$) 내의 현재의 이진수의 이진수 개수 및/또는 이미 코딩된 이진수 및 심벌의 값을 기반으로 하여 유도할 수 있다.

[0322] *확률 추정기 및 할당기*가 값의 쌍($\{b_{LPB}, p_{LPB}\}$)을 특징으로 하는 복수의 확률 모델을 유지한다. 이것은 이진수(b) 및 관련 확률 모델 지시(c_b)를 이진화기로부터 수신해, 지시된 확률 모델의 LPB 값(b_{LPB})과 LPB 확률(p_{LPB})을 코딩 이진수 유도기 및 확률 양자화기에 각각 전송한다. 그 이후, 코딩 확률 모델($\{b_{LPB}, p_{LPB}\}$)은 수신된 이진수(b)을 사용하여 업데이트된다.

[0323] *코딩 이진수 유도기*는 이진수(b) 및 관련 LPB 값(b_{LPB})을 이진화기 및 확률 추정기 및 할당기 각각으로부터 수신하여, ($b^c = b \oplus b_{LPB}$)에 의해 유도되는 코딩 이진수(b^c)를 확률 양자화기에 전송한다. *확률 양자화기*는 각 코딩 이진수(b^c)를 K개의 이진수 인코더 중 하나에 전송한다. 양자화기는 LPB 확률 간격 양자화($\{I_k\}$)에 관한 정보를 포함한다. 코딩 이진수(b_c)와 관련되며 확률 추정기 및 할당기로부터 수신된 LPB 확률(p_{LPB})은 ($p_{LPB} \in I_k$)를 유도한 간격 경계($\{p_k\}$)와 확률 간격 인덱스(k)에 비교한다.

[0324] K개의 *이진수 인코더* 각각은 *이진수 버퍼*와 *이진수 인코더*로 구성된다. 이진수 버퍼는 코딩 이진수(b^c)를 확률 양자화기로부터 수신하여 이들을 코딩 순서로 저장한다. 이진수 인코더는 특정한 V2V 매핑을 구현하여, 이진수 버퍼의 이진수 시퀀스를 코드워드와 관련되는 이진수 시퀀스와 비교한다. 이진수 버퍼의 이진수 시퀀스가 이들 이진수 시퀀스 중 하나와 같다면, 이진수 인코더는 이진수 시퀀스($\{b^c\}$)를 이진수 버퍼로부터 제거하여 관련 코드워드($(\{b^c\})$)를 대응하는 코드워드 스트림에 기록한다. 심벌 시퀀스에 대한 인코딩 처리의 끝에서, 이진수 버퍼가 비어있지 않은 모든 이진수 인코더에 대해, 종료하는 코드워드를 4.3절에 기재한 대로 기록한다.

[0325] K개의 결과적인 코드워드 스트림은 별도로 송신되고, 패킷화되거나 저장될 수 있거나, 송신 또는 저장을 위해 인터리빙될 수 있다(6.2절과 비교).

[0326] 디코더 측에서, 이진수 디코더와 이진수 버퍼로 구성된 K개의 이진 디코더 각각은 하나의 코드워드 스트림을 수신한다. 이진수 디코더는 코드워드($\{b^c\}$)를 코드워드 스트림으로부터 판독하여 관련된 이진수 시퀀스($\{b^c\}$)를 코딩 순서로 이진수 버퍼에 삽입한다.

[0327] 심벌 시퀀스의 디코딩은 기저의 신택스에 의해 운영된다. 심벌(s)에 대한 요청은 심벌 카테고리(c_s)와 함께 이진화기로 전송된다. *이진화기*는 이들 심벌 요청을 이진수에 대한 요청으로 전환한다. 이진수에 대한 요청은 인코더에서와 동일하게 유도되는 확률 모델 지시(c_b)와 관련되며, 확률 추정기 및 할당기에 전송된다. *확률 추정기 및 할당기*는 인코더 측에서의 그 상대 장치와 유사하게 동작한다. 확률 모델 지시(c_b)를 기반으로 하여, 이 장치는 확률 모델을 식별하여 그 LPB 값(b_{LPB})과 LPB 확률(p_{LPB})을 이진수 유도기 및 확률 양자화기에 각각 전송한다.

[0328] *확률 양자화기*는, 이진 인코더가 인코더 측에서 결정되는 것과 동일하게 LPB 확률(p_{LPB})을 기반으로 K개의 이진

디코더 중 하나를 결정하고, 제1 코딩 이진수(b^c)를 코딩 순서로 대응하는 이진수 버퍼로부터 제거하여, 이것을 이진수 유도기에 전송한다. *이진수 유도기*는 코딩 이진수(b_c) 및 관련 LPB 값(b_{LPB})을 확률 양자화기 및 확률 추정기 및 할당기 각각으로부터 수신하여, 이진수 값($b = b^c \oplus b_{LPB}$)을 결정한다. 이진화기에 의해 전송된 이진수 요청에 대한 마지막 응답으로서, 이진수 유도기는 디코딩된 이진수 값(b)을 이진화기 및 확률 추정기 및 할당기에 전송한다.

[0329] 확률 추정기 및 할당기에서, 디코딩된 이진수(b)의 값은, 인코더 측에서와 동일하게 관련 값(c_b)에 의해 선택되었던 확률 모델($\{b_{LPB}, p_{LPB}\}$)을 업데이트하는데 사용된다. 마지막으로, 이진화기는 수신된 이진수(b)를 심벌 요청을 위해 이미 수신되었던 이진수 시퀀스($_s$)에 더하고 이 이진수 시퀀스($_s$)를 이진화 방식($\gamma^{c_s}_{b}$)에 의해 심벌 값과 관련되는 이진수 시퀀스와 비교한다. 이진수 시퀀스($_s$)가 이들 이진수 시퀀스 중 하나와 매칭한다면, 대응하는 디코딩된 심벌(s)은 심벌 요청에 대한 마지막 응답으로서 출력된다. 그렇지 않다면, 이진화기는 심벌(s)이 디코딩될 때까지 이진수 요청을 추가로 전송한다.

[0330] 심벌 시퀀스의 디코딩은, 신택스에 의해 운영되는 추가 심벌 요청이 수신되지 않는다면 종료된다. (종료 코드워드의 결과로서) 엔트로피 디코딩 처리의 끝에서 이진수 버퍼에 포함될 수 있는 코딩 이진수(b^c)는 버린다.

[0331] 도 3 내지 도 13에 관한 도 1 및 도 2에서의 PIPE 인코더 및 PIPE 디코더에 대한 특정 실시예를 기술하고 일반적으로 도 14 내지 도 19에 대한 PIPE 코딩에 관한 수학적 배경을 제공한 이후, 도 20 내지 도 24에 있어서, 엔트로피 인코딩 및 디코딩 장치에 대한 더 상세한 실시예를 기재한다. 도 22 내지 도 24의 다음의 실시예는 PIPE 인코딩된 비트스트림을 서로 간에 인터리빙할뿐만 아니라 VLC 비트스트림과 PIPE 인코딩된 비트스트림을 함께 인터리빙한다. 이에 비교하여, PIPE 인코딩된 비트스트림에 대한 인터리빙을 갖는 PIPE 인코더 및 PIPE 디코더에 대한 실시예, 즉 도 7 내지 도 13의 실시예는 단지 PIPE 인코딩된 비트스트림의 별도의 인터리빙을 제공했다. 이미 전술한 바와 같이, 이들 실시예도, 도 5 및 도 6에 도시한 바와 같은 비트스트림 인터리빙을 사용하는 것과 같이 다른 인터리버/디인터리버 쌍(134 및 228) 각각의 사용에 의해(도 1 및 도 2 참조) 완전히 인터리빙된 비트스트림을 달성하기 위한 기초로서 역할을 할 수 있다. 그러나 다음에 기재한 실시예는, 달리 말하면 1-스테이지 인터리버/디인터리버(128 및 230) 각각을 사용한 PIPE 인코디된 비트스트림뿐만 아니라 VLC 비트스트림 둘 모두에서 한 번의 인터리빙을 실행한다.

[0332] VLC 및 PIPE 코딩된 심벌이 비트스트림 내에서 인터리빙되는 실시예를 상세하게 설명하기 전에, 복잡도와 코딩 효율 사이의 더 적절한 절충을 달성하기 위해, 인터리빙이 없는 것의 기본적인 구조를 도 20 및 도 21에 관해 기술한다.

[0333] 도 1a의 엔트로피 인코딩 장치의 구조를 도 20에 도시한다. 엔트로피 인코딩 장치는 도 1 및 도 2의 VLC 인코딩된 및 PIPE 인코딩된 소스 심벌, 즉 각각 106 및 218의 조합에 대응하는 소스 심벌 스트림(1a)을 둘 이상의 부분 비트스트림(12 및 12a) 세트로 전환하며, 비트스트림(12a)은 도 1 및 도 2의 비트스트림(112 및 206)에 대응한다.

[0334] 앞서 이미 주목한 바와 같이, 각 소스 심벌(1a)은, 소스 심벌이 도 1의 VLC 인코더(102)에 대응하는 VLC 인코더(22a) 내의 표준 VLC 코드를 사용하여 코딩되는지나, 소스 심벌이 PIPE 코딩 컨셉으로 코딩될지에 대해 명시하는 지시와 관련될 수 있다. 도 1 및 도 2에 관해 이미 전술한 바와 같이, 이 지시는 디코딩 측에 명백하게 송신되지 않을 수 있다. 오히려, 관련된 지시는 소스 심벌 자체의 타입이나 심벌로부터 기인할 수 있다.

[0335] VLC 코딩된 심벌(1b)은 표준 VLC 코드로 코딩되며, 이 코드는 이제 VLC 인코더(22a)를 사용하여 바로 앞에서 언급한 심벌 카테고리나 심벌 타입에 의존할 수 있다. 대응하는 코드워드(11a)는 구별된 부분 비트스트림(12a)에 기록된다. 비-VLC 코드 심벌(1)은 예컨대 도 1 및 도 3에 대해 전술한 바와 같이 PIPE 코딩을 사용하여 코딩되며, 그에 따라 복수의 부분 비트스트림(12)을 얻는다. 소스 심벌(1a) 중 일부는 이미, 이진화가 도 1a에 관련하여 앞서 이미 언급한 바와 같이 두 부분으로 구성되는 방식으로 이진화되었을 수 도 있다. 이들 부분 중 하나는 PIPE 접근법으로 코딩될 수 있어서 대응하는 부분 비트스트림(12)에 기록될 수 있다. 이진수 시퀀스의 다른 부분은 표준 VLC 코드로 코딩될 수 있어서 대응하는 부분 비트스트림(12a)에 기록될 수 있다.

[0336] 도 20의 실시예에 맞는 기본적인 엔트로피 디코딩 장치를 도 21에 도시한다.

[0337] 디코더는 기본적으로 도 20의 인코더의 역동작을 실행하여, 소스 심벌의 이전이 인코딩된 시퀀스(27 및 27a)가 둘 이상의 부분 비트스트림(24 및 24a)의 세트로부터 디코딩된다. 디코더는 두 개의 상이한 처리 흐름을 포함한다: 인코더의 데이터 흐름을 복제하는 데이터 요청에 대한 흐름과, 인코더 데이터 흐름의 역을 나타내는 데이터 흐름. 도 21에서의 예시에서, 점선 화살표는 데이터 요청 흐름을 나타내는 반면, 실선은 데이터 흐름을 나타낸다. 디코더의 빌딩 블록은 기본적으로 인코더의 빌딩 블록을 복제하지만, 역 동작을 구현한다.

[0338] 본 발명의 바람직한 실시예에서, 각 심벌 요청(13a)은, 소스 심벌이 표준 VLC 코드를 사용하거나 PIPE 코딩 컨셉을 통해 코딩되는지를 명시하는 지시와 관련된다. 도 20과 관련하여 앞서 이미 구현된 바와 같이, 이 지시는, 소스 심벌 자체로 나타낸 신택스 요소의 신택스나 파싱 규칙으로부터 기인할 수 있다. 예컨대, 도 1 및 도 2를 참조하면, 상이한 신택스 요소 타입은 상이한 코딩 방식, 즉 VLC 코딩 또는 PIPE 코딩과 관련될 수 있음이 기재되어 있다. 동일한 사실이 상이한 이진화 부분, 또는 더 일반적으로는 신택스 요소의 다른 심벌화에 적용할 수 있다. 심벌이 VLC 코딩된다면, 요청은 VLC 디코더(22a)에 전달되며, VLC 코드워드(23a)가 구별된 부분 비트스트림(24a)으로부터 판독된다. 대응하는 디코딩 심벌(27a)이 출력된다. 심벌이 PIPE로 코딩된다면, 심벌(27)은 예컨대 도 4와 관련하여 전술한 바와 같이 부분 비트스트림(24)의 세트로부터 디코딩된다.

[0339] 본 발명의 다른 바람직한 실시예에서, 소스 심벌 중 일부는 이진화가 두 부분으로 구성되는 방식으로 이진화된다. 이들 부분 중 하나는 PIPE 접근법으로 코딩되며 관련 부분 비트스트림(24)으로부터 그에 따라 디코딩된다. 이진수 시퀀스 중 다른 부분은 표준 VLC 코드로 코딩되어, 구별된 부분 비트스트림(24a)으로부터 대응하는 코드워드(23a)를 판독하는 VLC 디코더(22a)로 디코딩된다.

[0340] **(VLC 코딩되고 PIPE 코딩된) 부분 비트스트림의 송신 및 다중화**

[0341] 인코더에 의해 만들어진 부분 비트스트림(12 및 12a)은 개별적으로 송신될 수 있거나, 단일 비트스트림으로 다중화될 수 있거나, 부분 비트스트림의 코드워드가 단일 비트스트림에 인터리빙될 수 있다.

[0342] 본 발명의 바람직한 실시예에서, 다량의 데이터에 대한 각 부분 비트스트림은 하나의 데이터 패킷에 기록된다. 다량의 데이터는 정지 화상, 비디오 시퀀스의 필드 또는 프레임, 정지 화상의 슬라이스, 비디오 시퀀스의 필드나 프레임의 슬라이스, 또는 오디오 샘플의 프레임 등과 같은 소스 심벌의 임의의 세트일 수 있다.

[0343] 본 발명의 다른 바람직한 실시예에서, 다량의 데이터에 대한 부분 비트스트림(12 및 12a) 중 둘 이상 또는 다량의 데이터에 대한 모든 부분 비트스트림은 하나의 데이터 패킷으로 다중화된다. 다중화된 부분 비트스트림을 포함하는 데이터 패킷의 구조는 도 5에 예시한 대로일 수 있다.

[0344] 데이터 패킷(300)은 (고려한 다량의 데이터에 대한) 각 부분 비트스트림의 데이터에 대한 하나의 분할과 헤더로 구성된다. 데이터 패킷의 헤더(301)는 데이터 패킷(그 나머지)의 비트스트림 데이터(302)의 세그먼트로의 분할에 대한 지시를 포함한다. 분할에 대한 지시 외에, 헤더는 추가 정보를 포함할 수 있다. 본 발명의 바람직한 실시예에서, 데이터 패킷의 분할에 대한 지시는 비트 또는 바이트 또는 복수의 비트 또는 복수의 바이트 단위인 데이터 세그먼트의 시작의 위치이다. 본 발명의 바람직한 실시예에서, 데이터 세그먼트의 시작의 위치는, 데이터 패킷의 시작에 대한 또는 헤드의 끝에 대한 또는 이전 데이터 패킷의 시작에 대한 데이터 패킷의 헤더에서 절대값으로 코딩된다. 본 발명의 다른 바람직한 실시예에서, 데이터 세그먼트의 시작의 위치는 차별적으로 코딩된다. 즉 데이터 세그먼트의 실제 시작과 데이터 세그먼트의 시작에 대한 예측 사이의 차이만이 코딩된다. 예측은 데이터 패킷의 전체 크기, 헤더의 크기, 데이터 패킷에서의 데이터 세그먼트의 개수, 이전 데이터 세그먼트의 시작의 위치와 같은 이미 알려져 있거나 송신된 정보를 기반으로 하여 유도할 수 있다. 본 발명의 바람직한 실시예에서, 제1 데이터 패킷의 시작의 위치는 코딩되는 것이 아니라 데이터 패킷 헤더의 크기를 기반으로 추론된다. 디코더 측에서, 송신된 분할 지시는 데이터 세그먼트의 시작을 유도하는데 사용된다. 데이터 세그먼트는 그 후 부분 비트스트림(12 및 12a)으로서 사용되며, 데이터 세그먼트에 포함된 데이터가 순차적인 순서로 대응하는 이진수 디코더 및 VLC 디코더에 공급된다.

[0345] **코드워드(VLC 및 PIPE 코드워드)의 인터리빙**

[0346] 일부 응용의 경우, 하나의 데이터 패킷에서의 (다량의 소스 심벌에 대한) 부분 비트스트림의 전술한 다중화는 다음의 단점을 가질 수 있다: 한편으론, 작은 데이터 패킷의 경우, 분할을 신호화하는데 필요한 부차적인 정보에 대한 비트 수가 부분 비트스트림에서 실제 데이터에 대해 상당하게 될 수 있으며, 이점은 결국 코딩 효율을 감소시킨다. 다른 한편으론, 다중화는 (예컨대, 비디오 회의 응용에 대해서와 같이) 작은 지연을 필요로 하는

응용에 대해 적절하지 않을 수 있다. 기술한 다중화로, 인코더는 부분 비트스트림이 완전히 만들어졌기 이전에 데이터 패킷의 송신을 시작할 수 없으며, 이는 분할의 시작의 위치는 이전에 알려져 있기 대문이다. 더 나아가, 일반적으로, 디코더는, 데이터 패킷의 디코딩을 시작할 수 있기 전에 마지막 데이터 세그먼트의 시작을 수신할 때까지 대기해야 한다. 비디오 회의 시스템과 같은 응용의 경우, 이들 지연은 (화상을 인코딩/디코딩하기 위해 두 화상 사이의 시간 간격을 거의 필요로 하는 인코더/디코더에 대해서 및 송신 비트율에 가까운 비트율에 대해) 여러 비디오 화상의 시스템의 추가 전체 지연에 합해질 수 있으며, 이점은 그러한 응용에서 중요하다. 특정 응용에 대한 단점을 극복하기 위해, 본 발명의 바람직한 실시예의 인코더는, 둘 이상의 이진수 인코더 및 VLC 인코더에 의해 생성되는 코드워드가 단일 비트스트림으로 인터리빙되는 방식으로 구성될 수 있다. 인터리빙된 코드워드를 갖는 비트스트림은 (작은 버퍼 지연을 무시할 경우, 아래 참조) 직접 디코더에 전송될 수 있다. 디코더 측에서, 둘 이상의 이진수 디코더 및 VLC 디코더는 디코딩 순서로 비트스트림으로부터 직접 코드워드를 판독하며, 디코딩은 수신된 제1 비트로 시작할 수 있다. 게다가, 부차적인 정보는 부분 비트스트림의 다중화(또는 인터리빙)를 신호화하는데 필요치 않다.

[0347] 코드워드 인터리빙을 갖는 인코더의 기본 구조를 도 22에 도시한다. 이진수 인코더(10) 및 VLC 인코더(10a)는 코드워드를 부분 비트스트림에 직접 기록하는 것이 아니라, 단일 코드워드 버퍼(29)와 연결되며, 이 버퍼로부터 코드워드가 코딩 순서로 비트스트림(34)에 기록된다. 이진수 인코더(10)는 하나 이상의 새 코드워드 버퍼 엔트리에 대한 요청(28)을 코드워드 버퍼(29)에 전송하며, 후에 코드워드(30)를 코드워드 버퍼(29)에 전송하며, 코드워드는 예비 버퍼 엔트리에 저장된다. VLC 인코더(10a)는 VLC 코드워드(30a)를 직접 코드워드 버퍼(29)에 기록한다. 코드워드 버퍼(29)의 (일반적으로 가변 길이) 코드워드(31)가, 대응하는 비트(33)를 생성된 비트스트림(34)에 기록하는 코드워드 기록기(32)에 의해 접근된다. 코드워드 버퍼(29)는 선입선출 버퍼로서 동작하여, 더 일찍 예비된 코드워드 엔트리가 비트스트림에 더 일찍 기록된다.

[0348] 본 발명의 바람직한 실시예에서, 코드워드 버퍼는 다음과 같이 동작한다. 새로운 이진수(7)가 특정한 이진수 버퍼(8)에 전송되고, 이진수 버퍼에 이미 저장된 이진수의 개수가 0이며, 특정한 이진수 버퍼와 연결된 이진수 인코더에 대한 코드워드 버퍼에 예비된 코드워드가 현재 없다면, 연결된 이진수 인코더(10)는 코드워드 버퍼에 요청을 전송하며, 이에 의해, 하나 이상의 코드워드 엔트리가 특정한 이진수 인코더(10)에 대한 코드워드 버퍼(29)에 예비된다. 코드워드 엔트리는 가변 개수의 비트를 가질 수 있고, 버퍼 엔트리에서 비트 개수에 대한 상한 임계치는 보통 대응하는 이진수 인코더에 대한 최대 코드워드 크기에 의해 주어진다. (코드워드 엔트리나 코드워드 엔트리들이 예비되어진) 이진수 인코더에 의해 생성된 그 다음 코드워드나 그 다음 코드워드들은 코드워드 버퍼의 예비된 엔트리 또는 엔트리들에 저장된다. 특정한 이진수 인코더에 대한 코드워드 버퍼에서의 모든 예비된 버퍼 엔트리가 코드워드로 채워지며, 그 다음 이진수가 특정한 이진수 인코더와 연결된 이진수 버퍼에 전송된다면, 하나 이상의 새 코드워드가 특정한 이진수 버퍼 등에 대한 코드워드 버퍼에 예비된다. VLC 인코더(10a)는 VLC 코드워드(30a)를 코드워드 버퍼(29)의 그 다음 사용 중이지 않은 엔트리에 직접 기록한다. 즉, VLC 인코더의 경우, 코드워드 예비 및 코드워드의 기록은 한 번에 행해진다. 코드워드 버퍼(29)는 특정한 방식으로 선입선출 버퍼를 나타낸다. 버퍼 엔트리는 순차적인 순서로 예비된다. 대응하는 버퍼 엔트리가 더 일찍 예비되어진 코드워드가 더 일찍 비트스트림에 기록된다. 코드워드 기록기(32)는, 연속해서나 코드워드(30)가 코드워드 버퍼(29)에 기록된 이후 코드워드 버퍼(29)의 상태를 체크한다. 제1 버퍼 엔트리가 전체 코드워드를 포함한다면 (즉, 버퍼 엔트리가 예비되는 것이 아니라, 코드워드를 포함한다면), 대응하는 코드워드(31)와 대응하는 버퍼 엔트리를 코드워드 버퍼(20)로부터 제거하고, 코드워드의 비트(33)를 비트스트림에 기록한다. 이러한 처리를, 제1 버퍼 엔트리가 코드워드를 포함하지 않을 때까지(즉, 예비되거나 사용 중이지 않을 때까지) 반복된다. 디코딩 처리의 끝에서, 즉 고려된 다량의 데이터의 모든 소스 심벌이 처리되었다면, 코드워드 버퍼를 플러싱해야 한다. 그러한 플러싱 처리의 경우, 다음의 절차가 각각의 이진수 버퍼/이진수 인코더에 제1 단계로서 적용된다. 이진수 버퍼가 이진수를 포함한다면, 특정한 또는 임의의 값을 갖는 이진수를, 결과적인 이진수 시퀀스가 코드워드와 관련된 이진수 시퀀스를 나타낼 때까지 더하고(앞서 주목한 바와 같이, 이진수를 더하는 하나의 바람직한 방법은, 이진수 버퍼의 원래의 내용을 프레픽스로서 포함하는 이진수 시퀀스와 관련되는 최단 가능 코드워드를 생성하는 이진수 값-또는 이들 값 중 하나-을 더하는 것이다), 그 후, 코드워드가 대응하는 이진수 인코더에 대한 그 다음 예비된 버퍼 엔트리에 기록되며, (대응하는) 이진수 버퍼가 비워진다. 하나 보다 많은 버퍼 엔트리가 하나 이상의 이진수 인코더에 예비된다면, 코드워드 버퍼는 여전히 예비된 코드워드 엔트리를 포함할 수 있다. 그 경우, 이들 코드워드 엔트리는 대응하는 이진수 인코더에 대한 임의의 그러나 유효한 코드워드로 채워진다. 본 발명의 바람직한 실시예에서, 최단 유효 코드워드 또는(복수 개가 있다면) 최단 유효 코드워드 중 하나를 삽입한다. VLC 인코더는 임의의 종료를 필요로 하지 않는다. 끝으로, 코드워드 버퍼에서의 모든 남은 코드워드를 비트스트림에 기록한다.

[0349] 코드워드 버퍼의 상태에 대한 두 개의 예를 도 23에 예시한다. 예(a)에서, 코드워드 버퍼는, 코드워드로 채워진 4개의 엔트리(이들 중 두 개는 VLC 엔트리임)와 3개의 예비 엔트리를 포함한다. 게다가, 그 다음 사용 중이지 않은 버퍼 엔트리를 표기한다. 제1 엔트리는 코드워드로 채워진다(즉, 이진수 인코더(2)는 코드워드를 이전에 예비된 엔트리에 막 기록했다). 다음 단계로, 이 코드워드는 코드워드 버퍼로부터 제거되고 비트스트림에 기록될 것이다. 그 후, 이진수 인코더(3)에 대한 제1 예비된 코드워드가 제1 버퍼 엔트리가 되지만, 이 엔트리는 코드워드 버퍼로부터 제거될 수 없으며, 이는 이 엔트리는 단지 예비된 것이며, 코드워드가 이 엔트리에 기록되지 않았기 때문이다. 예(b)에서, 코드워드 버퍼는 코드워드로 채워지는 4개의 엔트리(이들 중 하나는 VLC 버퍼 엔트리임)와 4개의 예비 엔트리를 포함한다. 제1 엔트리는 예비로서 표기되며, 따라서 코드워드 기록기는 코드워드를 비트스트림에 기록할 수 없다. 4개의 코드워드가 코드워드 버퍼에 포함될지라도, 코드워드 기록기는, 코드워드가 이진수 인코더(3)에 대한 제1 예비된 버퍼 엔트리에 기록될 때까지 대기해야 한다. 코드워드는, 디코더 측에서 처리를 반전시킬 수 있기 위해서(하기 부분 참조), 이들 코드워드가 예비된 순서로 기록되어야 함을 주목해야 한다. 또한, VLC 버퍼 엔트리가 항상 완료되며, 이는 코드워드의 예비 및 기록이 한 번에 행해지기 때문임을 주목해야 한다.

[0350] 코드워드 인터리빙을 갖는 디코더의 기본 구조를 도 24에 도시한다. 이진수 디코더(22) 및 VLC 디코더(2a)는 별도의 부분 비트스트림으로부터 직접 코드워드를 판독하지 않지만, 비트 버퍼(38)에 연결되며, 이 버퍼(38)로부터 코드워드(37 및 37a)가 코딩 순서로 판독된다. 비트 버퍼(38)가 반드시 필요한 것은 아니며, 이는 코드워드가 직접 비트스트림으로부터 판독될 수 도 있기 때문임을 주목해야 한다. 비트 버퍼(38)에는 주로 처리 체인의 명백하게 분리된 상이한 양상을 위해서 예시적으로 포함된다. 따라서, 인터리빙된 코드워드를 갖는 비트스트림(40)의 비트(39)는 선입선출 버퍼를 나타내는 비트 버퍼(38)에 순차적으로 삽입된다. 특정한 이진수 디코더(22)가 하나 이상의 이진수 시퀀스에 대한 요청(35)을 수신한다면, 이진수 디코더(22)는 비트에 대한 요청(36)을 통해 비트 버퍼(38)로부터 하나 이상의 코드워드(37)를 판독한다. 디코더는 즉시 소스 심벌을 디코딩할 수 있다. 유사하게, VLC 디코더(22a)가 새로운 심벌에 대한 요청(19a)을 수신한다면, 이것은 대응하는 VLC 코드워드(37a)를 비트 버퍼(38)로부터 판독하여 디코딩된 심벌(27a)을 복귀시킨다. (전술한 바와 같이) 인코더가, 코드워드 버퍼를 적절히 동작함으로써 코드워드가 이들 코드워드가 이진수 디코더에 의해 요청된 비트스트림에 동일한 순서로 기록됨을 보장해야 함을 주목해야 한다. 디코더에서, 전체 디코딩 처리는 소스 심벌에 대한 요청에 의해 트리거된다. 특정한 이진수 인코더에 의해 인코더 측에서 수신된 코드워드의 개수와 대응하는 이진수 디코더에 의해 판독된 코드워드의 개수로서 파라미터는 동일해야 한다.

[0351] **작은 지연 제약을 갖는 가변 길이 코드워드의 인터리빙**

[0352] 기술한 코드워드 인터리빙은, 임의의 분할 정보가 부차적인 정보로서 전송됨을 필요로 하지 않는다. 또한, 코드워드가 비트스트림에 인터리빙되므로, 지연은 일반적으로 작다. 그러나 (예컨대, 코드워드 버퍼에서 저장된 최대 비트 수에 의해 명시되는) 특정한 지연 제약을 따름을 보장하지는 않는다. 더 나아가, 코드워드 버퍼에 대한 필요한 버퍼 크기는 이론적으로는 매우 크게 될 수 있다. 도 23(b)에서의 예를 고려할 때, 추가 이진수가 이진수 버퍼(3)에 전송되지 않으며, 따라서 이진수 인코더(3)는, 데이터 패킷의 끝에서의 플러싱 처리가 적용될 때까지 임의의 새로운 코드워드를 코드워드 버퍼에 전송하지 않는 것이 가능할 수 있다. 그 후 이진수 인코더(1 및 2)에 대한 모든 코드워드는, 이들이 비트스트림에 기록될 수 있기 전에 데이터 패킷의 끝까지 대기해야 할 것이다. 이러한 단점은 추가 메커니즘을 인코딩 처리에 (및 또한 후술될 디코딩 처리에) 더함으로써 회피할 수 있다. 그 추가 메커니즘의 기본 개념은, 지연 또는 지연의 상한에 관련된 측정치(이후 참조)가 명시한 임계치를 초과한다면, 제1 예비된 버퍼 엔트리가 (데이터 패킷의 끝에서와 유사한 메커니즘을 사용하여) 대응하는 이진수 버퍼를 플러싱함으로써 채워진다는 점이다. 그러한 메커니즘에 의해, 대기 중인 버퍼 엔트리의 개수는, 관련된 지연 측정치가 명시한 임계치 미만일 때까지 감소한다. 디코더 측에서, 지연 제약에 따르기 위해 인코더 측에 삽입되었던 이진수는 버려져야 한다. 이렇게 이진수를 버리기 위해서, 기본적으로 인코더 측에서와 동일한 메커니즘이 사용될 수 있다.

[0353] VLC 및 PIPE 코딩의 비트스트림을 인터리빙하기 위한 가능성을 상세하게 기재한 후, 다음의 설명에서, 도 1b, 도 1c 및 도 2b에 대해 언급한 바와 같이 이미 앞서 언급한 소스 심벌로 분해된 신택스 요소에 대해 다시 주로 다룬다. 예시용으로, 다음의 설명은, 그에 따라 분해된 신택스 요소가 절대 변환 계수 레벨이고 가정한다. 그러나 이것은 단지 예이며, 다른 타입의 신택스 요소를 또한 다룰 수 있다. *특히, 다음의 설명에서, 블록-기반 영상 및 비디오 코더에서 상이한 엔트로피 코드를 분할하고 사용함으로써 절대 레벨의 코딩을 기술한다.*

[0354] 예컨대, 비디오 시퀀스의 화상은 보통 블록으로 분해된다. 블록 또는 블록의 컬러 성분은 움직임 보상 예측이나 인트라 예측에 의해 예측된다. 블록은 상이한 크기를 가질 수 있으며, 쿼드러틱(quadratic) 또는 렉탱귤러

(rectangualr)일 수 있다. 블록 또는 블록의 컬러 성분의 모든 샘플은 참조 인덱스(이미 코딩된 화상 세트에서의 참조 화상을 식별함), 움직임 파라미터(참조 화상과 현재의 화상 사이의 블록의 움직임에 대한 측정치를 명시함), 보간 필터를 명시하는 파라미터, 인트라 예측 모드 등과 같은 동일한 예측 파라미터 세트를 사용하여 예측된다. 움직임 파라미터는 수평 및 수직 성분을 갖는 변위 벡터에 의해서나 6개의 성분으로 구성된 아핀(affine) 움직임 파라미터와 같은 더 고차의 움직임 파라미터에 의해 표현될 수 있다. (참조 인덱스와 움직임 파라미터와 같은) 하나 보다 많은 예측 파라미터 세트가 단일 블록과 관련되는 것도 가능하다. 그 경우, 각 예측 파라미터 세트의 경우, 블록이나 블록의 컬러 성분에 대한 단일 중간 예측 신호를 생성하여, 중간 예측 신호의 가중된 합에 의해 최종 예측 신호를 만든다. 가중 파라미터 및 잠재적으로는 상수 오프셋(가중된 합에 더해짐)은 화상, 또는 참조 화상, 또는 참조 화상 세트마다 정해질 수 있거나, 대응하는 블록에 대한 예측 파라미터 세트에 포함될 수 있다. 유사하게, 정지 영상은 또한 종종 블록으로 분해되며, 블록은 (블록의 DC 성분을 예측하는 공간 인트라 예측 방법이나 간단한 인트라 예측 방법일 수 있는) 인트라-예측 방법에 의해 예측된다. 코너인 경우, 예측 신호는 또한 0일 수 있다.

[0355] 원래의 블록이나 원래의 블록의 컬러 성분과 잔여 신호로도 지칭되는 대응하는 예측 신호 사이의 차이는 보통 변환되고 양자화된다. 2차원 변환이 잔여 신호에 적용되며, 최종적인 변환 계수를 양자화한다. 이러한 변환 코딩의 경우, 특정한 예측 파라미터 세트가 사용되었던 블록이나 블록의 컬러 성분은 변환을 적용하기 전에 더 분할할 수 있다. 변환 블록은 예측에 사용되는 블록 이하일 수 있다. 정지 영상이나 비디오 시퀀스의 화상에서의 상이한 변환 블록은 상이한 크기를 가질 수 있으며, 변환 블록은 쿼드롤릭 또는 렉탱귤러 블록을 나타낼 수 있다.

[0356] 이들 모든 예측 및 잔여 파라미터는 각각 신택스 요소(138 및 226)의 스트림을 형성할 수 있다.

[0357] 변환 계수 레벨로도 지칭되는 최종적인 양자화된 변환 계수는 이제 상기 코딩 방식 중 임의의 방식에 의한 엔트로피 코딩을 사용하여 송신될 수 있다. 이를 위해, 변환 계수 레벨 블록이 스캔을 사용하여 변환 계수 값의 벡터(즉, 정해진 세트)에 매핑될 수 있으며, 여기서 상이한 스캔이 상이한 블록마다 사용될 수 있다. 종종 지그재그 스캔을 사용한다. 인터레이스된 프레임의 한 필드의 샘플만을 포함하는 블록(이들 블록은 코딩된 필드의 블록이거나 코딩된 프레임의 필드 블록일 수 있음)의 경우, 필드 블록을 위해 특히 설계된 상이한 스캔을 사용하는 것이 또한 일반적이다. 변환 계수의 최종적인 순서가 정해진 시퀀스를 인코딩하기 위한 가능한 코딩 방식은 런-렝쓰 코딩이다. 보통, 매우 많은 수의 변환 계수 레벨은 0이며, 0인 연속적인 변환 계수 레벨 세트는, 각각의 신택스 요소에 의해 0(런)인 연속 변환 계수 레벨의 수를 코딩함으로써 효율적으로 표현할 수 있다. 나머지 (비-0) 변환 계수의 경우, 실제 레벨은 각각의 신택스 요소의 형태로 코딩된다. 런-렝쓰 코드의 여러 대안이 있다. 비-영 계수 이전의 런 및 비-0 변환 계수의 레벨은 단일 신택스 요소를 사용하여 함께 코딩될 수 있다. 종종, 마지막 비-0 변환 계수 이후 전송된 블록 끝에 대한 특별한 신택스 요소를 포함시킨다. 또는, 비-0 변환 계수 레벨의 수를 먼저 인코딩하고, 이 수에 따라, 레벨 및 런이 코딩된다.

[0358] 다소 상이한 접근법이 H.264/AVC에서 매우 효율적인 CABAC 엔트로피 코딩에서 사용된다. 여기서, 변환 계수 레벨의 코딩은 세 단계로 분할된다. 제1 단계에서, 이진수 신택스 요소 coded_block_flag가 각 변환 블록에 대해 송신되어, 변환 블록이 상당한 변환 계수 레벨(즉, 비-0인 변환 계수)을 포함하는지를 신호화한다. 이 신택스 요소가 상당한 변환 계수 레벨이 존재함을 지시한다면, 이진-값 중요성(significance) 맵이 코딩되어, 변환 계수 레벨 중 어떤 것이 비-0 값을 갖는지를 명시한다. 또한, 역 스캔 순서로, 비-0 변환 계수 레벨의 값이 코딩된다. 중요성 맵은 다음과 같이 신택스 요소 스트림(138)으로 코딩된다. 스캔 순서의 각 계수에 대해, 이진수 신택스 요소 significant_coeff_flag은 코딩되어, 대응하는 변환 계수 레벨이 0이 아닌지를 명시한다. significant_coeff_flag 이진수가 0이라면, 즉 비-0 변환 계수 레벨이 이 스캐닝 위치에서 존재한다면, 추가 이진 신택스 요소 last_significant_coeff_flag가 코딩된다. 이 이진수는, 현재의 중요한 변환 계수 레벨이 블록 내부의 마지막 중요한 변환 계수 레벨인지나 추가적인 중요한 변환 계수 레벨이 스캐닝 순서에서 뒤따르는지를 지시한다. last_significant_coeff_flag가, 추가적인 중요한 변환 계수가 뒤따르지 않음을 지시한다면, 추가적인 신택스 요소가 블록에 대한 중요성 맵을 명시하기 위해 코딩되지 않는다. 다음 단계로, 중요한 변환 계수 레벨의 값이 코딩되며, 블록 내의 그 위치는 이미 중요성 맵에 의해 결정된다. 중요한 변환 계수 레벨의 값은 다음의 세 개의 신택스 요소를 사용하여 역 스캐닝 순서로 코딩된다. 이진 신택스 요소 coeff_abs_greater_one은, 중요한 변환 계수 레벨의 절대값이 1보다 큰지를 지시한다. 이진수 신택스 요소 coeff_abs_greater_one가, 절대값이 1보다 큼을 지시한다면, 추가적인 신택스 요소 coeff_abs_level_minus_two가 전송되어, 변환 계수 레벨의 절대값-2를 명시한다. 이것이 다음의 실시예에서 도 1b, 도 1c 및 도 2b에 따라 그 처리가 실행되는 신택스 요소의 종류이다. 끝으로, 변환 계수 값의 부호를 명시하는 이진 신택스 요소 coeff_sign_flag가 각 중요한 변환

계수 레벨마다 코딩된다. 다시, 중요성 맵에 관련된 신택스 요소는 스캐닝 순서로 코딩되는 반면, 변환 계수 레벨의 실제값에 관련된 신택스 요소는 역 스캐닝 순서로 코딩되어 더 적절한 컨텍스트 모델의 사용을 허용함을 주목해야 한다. 적응성 스캔 패턴이 H.265/HEVC의 제1 테스트-모델에서처럼 중요성 맵에 사용되는 것이 또한 가능하다. 다른 개념이, H.265/HEVC의 제1 테스트-모델에서의 4x4보다 큰 변환 블록에 대한 절대 변환 계수 레벨의 코딩에 사용된다. 4x4보다 큰 변환 블록의 경우, 더 큰 변환 블록이 4x4 블록으로 분할되며, 4x4 블록은 스캔 순서로 코딩되는데 반해, 각 4x4 블록에 대해 역 스캔 순서가 사용된다.

[0359] H.264/AVC의 CABAC 엔트로피 코딩에서, 변환 계수 레벨에 대한 모든 신택스 요소가 이진 확률 모델링을 사용하여 코딩된다. 비-이진 신택스 요소 coeff_abs_level_minus_two는, 예컨대 먼저 이진화되며, 즉 이진수 결정 시퀀스(이진수)에 매핑하고, 이들 이진수가 순차적으로 코딩된다. 이진수 신택스 요소 significant_coeff_flag, last_significant_coeff_flag, coeff_abs_greater_one 및 coeff_sign_flag는 직접 코딩된다. (이진 신택스 요소를 포함한) 각 코딩된 이진수는 컨텍스트와 관련된다. 컨텍스트는 코딩된 이진수의 등급에 대한 확률 모델을 나타낸다. 두 개의 가능한 이진수 값 중 하나에 대한 확률에 관련된 측정치는 대응하는 컨텍스트와 이미 코딩되었던 이진수의 값을 기반으로 하여 각 컨텍스트에 대해 추정된다. 변환 코딩에 관련된 여러 이진수에 대해, 코딩에 사용된 컨텍스트는 이미 송신된 신택스 요소를 기반으로 하거나 블록 내부의 위치를 기반으로 하여 선택된다.

[0360] 중요성 맵을 코딩한 후, 블록은 역 스캔 순서로 처리된다. 전술한 바와 같이, 다른 컨셉이 H.265/HEVC의 제1 테스트-모델에 사용된다. 4x4보다 큰 변환 블록은 4x4 블록으로 분할되며, 최종적인 4x4 블록은 스캔 순서로 처리되는데 반해, 4x4 블록의 계수는 역 스캔 순서로 코딩된다. 다음의 설명은 H.265/HEVC 및 H.264/AVC의 제1 테스트-모델에서의 모든 4x4 블록에 대해 그리고 또한 H.264/AVC에서의 8x8 블록에 대해 유효하며, 이 설명은 또한 신택스 요소(138 및 226) 각각의 스트림 구성에 적용될 수 있다.

[0361] 스캔 위치가 중요하다면, 즉 계수가 0과는 상이하다면, 이진 신택스 요소 coeff_abs_greater_one은 스트림(138) 내에서 송신된다. 초기에, (블록 내에서), 대응하는 컨텍스트 모델 세트의 제2 컨텍스트 모델이 coeff_abs_greater_one 신택스 요소에 대해 선택된다. 블록 내부의 임의의 coeff_abs_greater_one 신택스 요소의 코딩된 값이 1보다 크다면(즉, 절대 계수가 2보다 크다면), 컨텍스트 모델링이 세트의 제1 컨텍스트 모델로 다시 전환되어, 이 컨텍스트 모델을 블록의 끝까지 사용한다. 그렇지 않다면, (블록 내의 coeff_abs_greater_one의 모든 코딩된 값이 0이며, 대응하는 절대 계수 레벨이 1이면), 컨텍스트 모델은, 고려된 블록의 역 스캔에서 이미 처리되었던 0인 coeff_abs_greater_one 신택스 요소의 수에 따라 선택된다. 신택스 요소 coeff_abs_greater_one에 대한 컨텍스트 모델 선택은 다음의 수학식에 의해 요약할 수 있으며, 여기서 현재의 컨텍스 모델 인덱스(C_{t+1})는 이전의 컨텍스트 모델 인덱스(C_t) 및 수학식에서 bint로 표현된 이전에 코딩된 신택스 요소 coeff_abs_greater_one의 값을 기반으로 하여 선택된다. 블록 내부의 제1 신택스 요소 coeff_abs_greater_one의 경우, 컨텍스트 모델 인덱스는 C_1=1이 되게 설정된다.

$$C_{t+1}(C_t, bin_t) = \begin{cases} 0 & bin_t = 1 \\ \min(C_t + 1, 4) & bin_t = 0 \end{cases}$$

[0362]

[0363] 동일한 스캔 위치에 대한 coeff_abs_greater_one 신택스트 요소가 1일 때, 절대 변환 계수 레벨을 코딩하기 위한 제2 신택스 요소 coeff_abs_level_minus_two만이 코딩된다. 비-이진수 신택스 요소 coeff_abs_level_minus_two는 이진수 시퀀스로 이진화되며, 이 이진화의 제1 이진수에 대해, 컨텍스트 모델 인덱스는 다음에서 기재된 대로 선택된다. 이진화의 나머지 이진수는 고정된 컨텍스트로 코딩된다. 이진화의 제1 이진수에 대한 컨텍스트는 다음과 같이 선택된다. 제1 coef f_abs_level_minus_two 신택스 요소에 대해, coeff_abs_level_minus_two 신택스 요소의 제1 이진수에 대한 컨텍스트 모델의 세트 중 제1 컨텍스트 모델이 선택되고, 대응하는 컨텍스트 모델 인덱스는 C_1=0이 되게 설정된다. coeff_abs_level_minus_two 신택스 요소의 각각의 추가적인 제1 이진수에 대해, 컨텍스트 모델링은 세트에서 그 다음 컨텍스트 모델로 전환하며, 여기서 세트에서의 컨텍스트 모델의 수는 5로 제한된다. 컨텍스트 모델 선택은 다음의 공식에 의해 표현할 수 있으며, 여기서 현재의 컨텍스트 모델 인덱스(C_{t+1})는 이전 컨텍스트 모델 인덱스(C_t)를 기반으로 하여 선택된다.

$$C_{t+1}(C_t) = \min(C_t + 1, 4)$$

[0364]

[0365] 전술한 바와 같이, 블록 내의 제1 신택스 요소 coeff_abs_remain_minus_two에 대해, 컨텍스트 모델 인덱스는

C_t=0이 되게 설정된다. 그 상이한 컨텍스트 모델의 세트는 신택스 요소 coeff_abs_greater_one 및 coeff_abs_remain_minus_two에 대해 정의될 수 있음을 주목해야 한다. 또한, H.265/HEVC의 제1 테스트-모델에 대해, 4x4보다 큰 변환 블록이 4x4 블록으로 분할될 수 있음을 주목해야 한다. 분할된 4x4 블록은 스캐닝 순서로 처리될 수 있고, 각 분할된 4x4 블록에 대해 컨텍스트 세트는 이전의 4x4 블록에서 1보다 큰 계수의 수를 기반으로 하여 유도할 수 있다. 4x4보다 큰 변환 블록의 제1 4x4 블록에 대해 및 원래의 4x4 변환 블록에 대해, 개별적인 컨텍스트 세트를 사용할 수 있다.

[0366] 즉, 다음의 설명에서, 컨텍스트를 기반으로 한 코딩이 coeff_abs_greater_one 및 coeff_abs_remain_minus_two가 다음의 실시예에 따라 분해되는 소스 심벌 중 임의의 것에 대해 사용될 때, 이 컨텍스트 유도는 예컨대 할당기(114 및 212) 및 VLC 인/디코더(102 및 202)에 의해 사용될 수 있다.

[0367] 종래기술에 비교하여 CABAC 또는 PIPE에 의해 처리된 이진수의 수에서 및 또한 계산 복잡도 면에서의 복잡도를 감소시키거나 또한 코딩 효율을 증가시키기 위해서, 후술할 실시예는 영상 및 비디오 인코더 및 디코더에서 상이한 분할(140_1 내지 140_3)마다 상이한 가변 길이 코드를 사용함으로써 절대 레벨을 코딩하기 위한 접근법을 기재한다. 후술될 실시예는 그러나 적응형 루프 필터의 움직임 벡터차나 계수처럼 영상 및 비디오 코더에 대한 절대 레벨의 모든 종류에 적용될 수 있다. 변환 계수 레벨의 코딩이 후술될 바와 같이 실행되는 동안, 중요성 맵의 코딩은 H.265/HEVC의 제1 시험-모델에서처럼 또는 전술한 바와 같이 남아 있을 수 있거나, 중요성 맵의 코딩이 또한 H.264/AVC 등에서처럼 행해질 수 도 있다.

[0368] 전술한 바와 같이, 절대 변환 레벨의 코딩은 복수의 분할(140_{1-3})에서 실행된다. 코딩 방식은 세 개의 분할(140_{1-3})을 갖고 도 1b에서 예시적으로 예시되었다. 방식의 경계(142 및 144)는 가변적이며, 결국 가변 분할 크기를 갖는다. 코딩 방식은 다음과 같이 행해진다.

[0369] 제1 엔트로피 코드가 제1 성분 또는 소스 심벌, 즉 절대 변환 계수 레벨(z)이 limit1보다 작은 경우 이것 또는 그렇지 않은 경우 limit1을 코딩하는데 사용된다. 절대 변환 계수 레벨이 제1 분할(140_1)의 경계 limit1 이상이라면, 제1 분할(140_1)의 경계 limit1(142)은 절대 변환 계수 레벨로부터 감산되며, 결과적인 값(z')이 제2 엔트로피 코드로 코딩된다. 남은 절대 변환 계수 레벨(z')이 제2 분할(140_2)에 대한 경계 limit2-limit1 이상이라면, 제2 분할의 경계 limit2-limit1은 절대 변환 계수 레벨(z')로부터 다시 감산되며, 결과적인 값이 제3 엔트로피 코드로 코딩된다. 일반적으로 말해, 분할의 경계에 도달할 때, 경계로의 그 다음 분할에 대한 엔트로피 코드가 절대 변환 계수 레벨-대응하는 분할의 경계로부터 얻은 값을 코딩하는데 사용된다.

[0370] 엔트로피 코드는 런-렝쓰 코드와 같은 간단한 가변 길이 코드나 룩업 표(예컨대, 허프만 코드)나 CABAC 또는 PIPE와 같은 확률 모델을 사용하는 더 복잡한 엔트로피 코드일 수 있다. 분할의 수와 분할의 경계는 가변적일 수 있거나, 실제 신택스 요소에 의존할 수 있다. 도 1b에 도시한 분할 컨셉은 다음의 장점을 갖는다. 다음의 설명에서, 우리는 예컨대 절대 변환 계수를 사용하지만, 이들은 어떤 다른 신택스 요소로 대체할 수 있음을 이해해야 한다. 예로서, 절대 변환 계수 레벨의 확률 분포는 대략 기하학적 분포일 수 있다. 그러므로 기하학적 분포에 대해 최적화된 엔트로피 코드는 절대 변환 계수 레벨을 코딩하는데 사용될 수 있다. 그러나 컨텍스트 모델링 및 확률 모델 선택을 사용하더라도, 그러한 모델은 항상 국지적으로 최적이지는 않다. 예컨대, 변환 블록의 경우, 국지적 절대 변환 계수 레벨은, 큰 수의 법칙(the law of large numbers)으로 인해 영상이나 비디오에서 특정한 양의 블록에 (특정한 정확도로) 모델이 유효한 동안 동일한 양의 매우 작은 및 중간 범위의 변환 계수 레벨을 포함한다면, 기하학적이지 않은 분포를 따른다. 그러한 경우, 기하학적인 분포는 적절한 모델이 아니다. 또한, 큰 값을 갖는 절대 변환 계수 레벨을 고려할 때, 이들 큰 값의 분포는 종종 균일하다. 분할 컨셉은 상이한 절대 변환 계수 레벨마다 상이한 확률 모델을 허용한다. 더 작은 절대값에 대해, 더 복잡한 엔트로피 코드가 더 큰 효율을 위해 적용될 수 있는 반면, 더 큰 절대값에 대해, 덜 복잡한 엔트로피 코드가 사용되어 복잡도를 줄일 수 있다.

[0371] 전술한 바와 같이, 상이한 분할에 적절한 엔트로피 코드가 사용된다. 본 발명의 바람직한 실시예에서, 세 가지 타입의 엔트로피 코드를 사용한다. 제1 엔트로피 코드는 PIPE를 사용한다. 그러나 대안에 따라서는 CABAC와 같은 엔트로피 코딩 방법이나 임의의 다른 산술 코더가 택일적으로 사용될 수 있음을 주목해야 한다. 즉, 제1 심벌(s_1)(도 2b 참조)은 PIPE 코딩 경로를 통해 코딩될 수 있다. 서브디바이더(100) 및 재결합기(200)가 적절히 동작한다.

[0372] 제2 타입의 경우, 골롬 코드 및 서브세트를 포함하는 일부 트렁케이트된 변형(예컨대, 골롬-라이스 코드)이 사

용될 수 있다. 즉, 제2 심벌(s_2)(도 2b 참조)이 VLC 인/디코더(102/202)에서 그러한 VLC 코드를 통해 코딩될 수 있다. 서브디바이더(100) 및 재결합기(220)가 적절히 동작한다.

[0373] Exp-골롬 코드가 제3 타입으로서 사용된다. 즉, 제3 심벌(s_3)(도 2b 참조)이 VLC 인/디코더(102/202)에서 그러한 VLC 코드를 통해 코딩될 수 있다. 서브디바이더(100) 및 재결합기(220)가 적절히 동작한다. 상이한 VLC 코드와 VLC 및 PIPE 또는 VLC 및 산술 코드의 상이한 조합이 가능하다.

[0374] 제1 코드가 더 복잡하지만 더 양호한 압축 성능을 산출하는 반면, 제2 엔트로피 코드는 복잡도와 성능 사이의 합리적인 절충을 나타낸다. 예컨대, Exp-골롬 코드와 같은 마지막 엔트로피 코드는 복잡도가 매우 낮다. 다음의 설명에서, 상이한 분할의 코딩을 기술한다.

[0375] 심벌(s_1)처럼 분할(140_1)과 같은 분할이 PIPE 코더(104)와 같은 확률 모델을 사용하는 엔트로피 코더를 사용하여 엔트로피 코딩된다면(다시, CABAC 또는 임의의 다른 산술 코더가 더 이상 기재되지 않은 대안에서 사용될 수 있음), 서브디바이더(120)는 이 분할을 PIPE 인코더(104)에 보낸다. 먼저, 비-이진수 값의 절대 변환 계수 레벨은 이진화 방법을 사용하여 심벌라이저(122)에서 이진화될 수 있다. 이진화 방법은 비-이진수 값의 절대 변환 계수 레벨을 이진 이진수 시퀀스에 매핑한다. 이진수 스트링의 각 이진수는 할당기(114)에 의해 선택된 컨텍스트로 코딩된다. 컨텍스트 모델링은 제1 이진수에 대해 행해질 수 있고 이진수 시퀀스의 다음의 이진수에 대해 H.264/AVC에서 coeff_abs_level_minus_two로서 고정될 수 있거나, 상이한 컨텍스트 모델링이 이진수 스트링의 각 이진수에 대해 사용될 수 있다. 이진화는 골롬 코드 또는 Exp-골롬 코드 또는 다른 가변 길이 코드와 같은 가변 길이 코드일 수 있다.

[0376] 다음으로, 분할(140_2) 또는 심벌(s_2)과 같은 분할이 골롬 코드로 여기서는 VLC 인코더(102)에서 코딩될 수 있으며, 각각이 VLC 디코더에서 디코딩된다. 골롬 코드는 기하학적 분포 소스에 대해 설계된 엔트로피 코드 세트이다. 골롬 코드의 순서가 0이라면, 골롬 코드는 일진 코드로서도 알려져 있다. 일진 코드는 H.264/AVC에서의 coeff_abs_level_minus_two의 이진화에 관련된다. 골롬 코드는 다음과 같이 구성된다. 특정한 골롬 파라미터(k)에 대해, 값(n)은 정수 나눗셈을 사용하여 골롬 파라미터(k)로 나눠져 나머지(r)를 계산한다.

[0377] $$p = \left\lfloor \frac{n}{k} \right\rfloor$$

[0378] r=n-pk

[0379] 상기 공식에 의해 명시된 파라미터를 유도한 후, 값(n)은 두 부분으로 코딩될 수 있다. 제1 부분은, 프레픽스 부분으로도 지칭되며, 일진 코드이다. 결과 값(p+1)은 1의 수 및 종료 0을 명시하거나 그 역의 관계가 성립한다. 나머지 값은, 나머지 부분으로도 지칭되고 r로 표시되며, 트렁케이트된 이진수 코드로 표현된다. 특정한 실시예에 따라, 골롬-라이스 코드가 소스 심벌(s_2)과 같은 소스 심벌을 코딩하는데 사용되며, 골롬-라이스 코드는 골롬 코드의 서브세트이다. 또한, 분할(140_2)과 같은 경계를 포함하는 분할(140_{1-3})에 대해 그러한 엔트로피 코드를 사용할 때, (소스 심벌(s_2)과 같은) 각각의 소스 심벌의 알파벳은 제한되며, 골롬-라이스 코드는 변경될 수 있어서, 코드 효율을 개선할 수 있다. 골롬-라이스 코드의 파라미터는 고정되거나 가변적일 수 있다. 파라미터가 가변적이라면, 파라미터는 컨텍스트 모델링 스테이지의 부분으로서 추정될 수 있다. 예컨대, 소스 심벌(s_2)이 VLC 인코더(102)에 들어간다면, 이 인코더는 s_2의 컨텍스트로부터 골롬-라이스 코드의 파라미터를 결정한다. 골롬-라이스 코드는 2의 파워로의 파라미터를 갖는 골롬 코드이다. 따라서, 이들 코드는 2에 의한 나눗셈 및 곱셈을 기반으로 하며, 그러므로 이들은 시프트 및 덧셈 동작을 갖는 이진 아키텍처에서 효율적으로 구현할 수 있다. 골롬-라이스 파라미터와 골롬 파라미터 사이의 관계는 그러므로 $k_{GOLOMB}=2^{kRICE}$이다. 골롬-라이스 코드의 경우에, 나머지 부분은 정확히 나머지 값의 이진 표현이다. 0의 골롬-라이스 파라미터에 대해, 결과 코드는 일진 코드와 동일하며 나머지 부분을 갖지 않는다. 1인 파라미터에 대해, 나머지 부분은 동일한 일진 프레픽스를 공유하는 2개의 입력 심벌을 갖는 하나의 이진수로 구성된다. 다음의 설명에서, 선택된 골롬-라이스 파라미터에 대한 일부 예시적인 표를 예시한다.

표 8

[0380]

값	프레픽스	나머지	프레픽스	나머지	프레픽스	나머지	프레픽스	나머지
	k=0		k=1		k=2		k=3	
0	0		0	0	0	00	0	000
1	10		0	1	0	01	0	001
2	110		10	0	0	10	0	010
3	1110		10	1	0	11	0	011
4	11110		110	0	10	00	0	100
5	111110		110	1	10	01	0	101
6	1111110		1110	0	10	10	0	110
7	11111110		1110	1	10	11	0	111
8	111111110		11110	0	110	00	10	000
9	1111111110		11110	1	110	01	10	001
10	11111111110		111110	0	110	10	10	010
11	111111111110		111110	1	110	11	10	011
12	1111111111110		1111110	0	1110	00	10	100
13	11111111111110		1111110	1	1110	01	10	101

[0381] 분할(140_2)의 경우 예컨대 limit2-limit1와 같은 분할의 범위와 골롬-라이스 코드의 파라미터에 의하면, 트렁케이션은 다음과 같이 행해질 수 있다. 골롬-라이스 파라미터는 나머지 부분을 나타내는데 필요한 이진수의 개수를 기술하며, 파라미터 값의 파워로의 2(the two to the power of the parameter value)는 동일한 프레픽스로 나타낼 수 있는 값의 수를 기술한다. 이들 값은 프레픽스 그룹을 형성한다. 예컨대, 파라미터 0에 대해, 단 하나의 프레픽스가 하나의 특정한 값을 나타낼 수 있는 반면, 파라미터 3에 대해, 8개의 입력 값이 동일한 프레픽스를 공유하며 따라서 프레픽스 그룹은 파라미터 3에 대해 8개의 값을 포함한다. 제한된 소스 알파벳 및 주어진 골롬-라이스 코드에 대해, 프레픽스의 마지막 이진수는 마지막 프레픽스 그룹에서의 값에 대해 생략될 수 있어서, 고정된 길이의 프레픽스를 얻을 수 있다.

[0383] *예컨대, 이 범위는 9일 수 있으며, 골롬-라이스 파라미터는 2이다. 이 예의 경우에, 동일한 프레픽스로 표현될 수 있는 값의 수는 4이다. 최대 값은 9이며, 이것은 또한 경계가 초과됨과 그 다음 분할의 그 다음 엔트로피 코드가 사용되어야 함을 지시한다. 이 예의 경우에, 0-3의 값은 프레픽스 0을 가지며, 4-7의 값은 프레픽스 10을 가지며, 8-9의 값은 프레픽스 110을 갖는다. 값 8-9가 마지막 프레픽스 그룹을 형성했기 때문에, 이들의 뒤이어 오는 0은 생략할 수 있고 값 8-9는 11로 표현할 수 있다. 다시 말해, VLC 인코더(102)에 입력된 소스 심벌(s_2)로서, s_2의 가능 값의 수가 9(=limit2-limit1+1)이고 이 소스 심벌에 대한 골롬-라이스 파라미터가 2인 소스 심벌(s_2)를 가정해 본다. 그러면, 각각의 골롬-라이스 코드워드는 방금 기재한 바와 같이 프레픽스를 갖는 그 소스 심벌에 대해 VLC 인코더(102)에 의해 출력될 것이다. 코드워드의 나머지 부분에 대해, 트렁케이트된 코드는 VLC 인코더(102)에 의해 다음과 같이 유도할 수 있다. 보통, 골롬-라이스 코드의 파라미터는 나머지 부분의 이진수의 수를 지시한다. 트렁케이트된 경우에, 나머지의 모든 이진수가 코딩될 필요는 없다. 트렁케이트된 경우에 대해, 고정된 프레픽스를 갖는 모든 값(예컨대, 프레픽스의 이진수는 생략함)을 계수한다. 계수한 값은 항상 프레픽스에 대한 값의 최대수 이하이며, 이는 코드가 트렁케이트되기 때문이다. 나머지 부분의 트렁케이션이 가능하다면, 마지막 프레픽스 그룹에 대한 트렁케이트된 나머지 부분의 유도를 다음의 단계로 진행할 수 있다. 먼저, 계수된 수 이하인 2의 파워로의 최대수(l)를 유도한다. 그 후, 제2 단계로서, 계수된 수보다 큰 2의 파워로의 최저수(h)를 유도한다. 제1 값(l)은 h개의 이진수의 나머지로 프레픽스 그룹의 값의 수를 기술한다. 이들 값의 나머지 모두는 0과 그 다음에 나머지 그룹에서 값의 수로 제한되는 나머지의 이진수 표현으로 시작한다. 새로운 프레픽스 그룹으로서 다뤄지는 마지막 프레픽스 그룹의 나머지 값에 대해, 제1 나머지 그룹을 형성한 결과 값을 제외하고는 동일한 절차가 행해지며, 나머지는 1로 시작한다. 이 절차는, 모든 나머지가 유도될 때까지 행해진다. 예컨대, 범위가 14이고, 파라미터는 3이다. 제1 프레픽스 그룹은 0-7의 값을 포함하며, 제2 프레픽스 그룹은 8-13을 갖는다. 제2 프레픽스 그룹은 6개의 값을 포함한다. 파라미터는 l=2 및 h=3이다. 따라서, 프레픽스 그룹의 제1 네 개의 값은 세 개의 이진수를 갖는 나머지로 표현된다(이어지는 0과 네 개의 값을 구별하기 위한 이진 표현). 마지막 2개의 값에 대해, 동일한 절차가 다시 행해진다. 파라미터는 l=1 및 h=2이다. 마지막 2개의 값의 나머지가 이제 10 및 11로 표현될 수 있다. 방법을 입증하는 다른 예는 4의 골롬-라이스 파라미터와 10의

범위이다. 이 예에 대해, 파라미터는 l=3 및 h=4이다. 이들 파라미터로, 제1 8개의 값에 대한 트렁케이트된 나머지 부분은 네 개의 이진수로 표현된다. 나머지 두 개의 값은 앞선 이전 예에서처럼 동일한 나머지 부분을 갖는다. 범위가 이전 예에 대해 9라면, 제2 런에 대한 파라미터는 l=0 및 h=1이다. 남은 단 하나의 값의 나머지 부분은 1이다.

[0384] 엔트로피 코드의 제3 타입은 Exp-골롬 코드일 수 있다. 이들 코드는 소스 심벌(s_3)과 같은 동일한 확률 분포(예컨대 파라미터 0을 가짐)에 대해 사용할 수 있다. 즉, VLC 인코더/디코더 쌍은 그 코딩에 책임을 질 수 있다. 이전처럼, 더 큰 절대 변환 계수 레벨은 종종 균일하게 분포된다. 더욱 정밀하게, 0-차 Exp-골롬 코드가 마지막 분할(140_3)을 코딩하는데 사용할 수 있다. 이전 분할(140_2)에 대한 시작 및 그에 따른 경계(144)는 가변적일 수 있다. 경계(144)의 위치는 이전 인코딩된/디코딩된 소스 심벌(106, 108 및/또는 110) 또는 신택스 요소(138)(또는 218, 204 및/또는 208 또는 신택스 요소(226))에 의존하여 VLC 인/디코더(102/200)에 의해 제어될 수 있다.

[0385] 바람직한 실시예에서, 분할의 수는 도 1b에 도시된 대로 3이며, 경계(142 및 144)는 가변적일 수 있다. 제1 분할(140_1)에 대해, PIPE 코딩은 전술한 바와 같이 사용할 수 있다. 그러나 택일적으로는 CABAC를 또한 사용할 수 있다. 이 경우, PIPE 인/디코더 쌍은 이진 산술 코딩 인/디코더 쌍에 의해 대체될 것이다. 트렁케이트된 골롬-라이스 코드는 제2 분할(140_2)에 대해 사용될 수 있으며, 0-차 Exp-골롬 코드가 마지막 분할(140_3)에 대해 사용될 수 있다.

[0386] 다른 바람직한 실시예에서, 분할(140_{1-3})의 수는 3이며, 제1 경계(142)는 고정되는 반면, 제2 경계(144)는 가변적이다. 제1 분할(140_1)에 대해, CABAC 또는 PIPE가 사용된다. 트렁케이트된 골롬-라이스 코드가 제2 분할(140_2)에 대해 사용될수 있고, 0-차 Exp-골롬 코드가 마지막 분할(140_3)에 대해 사용될 수 있다.

[0387] 다른 바람직한 실시예에서, 분할의 수는 2이다. 제1 분할(140_1)은 CABAC 또는 PIPE를 사용할 수 있다. 제2 분할은 140_2와 같은 골롬-라이스 코드를 사용할 수 있다.

[0388] 다른 바람직한 실시예에서, 분할의 수는 3인 반면, 두 경계(142 및 144)는 가변적이다. 예컨대, 제1 분할(140_1)에 대해, CABAC 또는 PIPE이 사용되는 반면, 제2 분할(140_2)은 트렁케이트된 골롬-라이스 코드를 사용할 수 있으며, 제2 분할(140_3)은 0-차 Exp-골롬 코드를 사용한다.

[0389] 바람직한 실시예에서, 적응형 확률 모델을 사용하는 CABAC 또는 PIPE를 사용하는 제1 분할(140_1)의 경계(142)는 2이다. 이 바람직한 실시예에서, 제1 이진수에 대한 컨텍스트 모델링은 전술한 바와 같이 coeff_abs_greater_one에 대해 기재된 대로 행해질 수 있고, 제2 이진수에 대한 컨텍스트 모델링은 또한 전술했던 대로 H.264/AVC에서 coeff_abs_level_minus_two에 대해 기술된 대로 행해질 수 있다. 후자의 컨텍스트 결정은 할당기(114 및 212)에 의해 각각 결정될 것이다. 다른 바람직한 실시예에서, 확률 모델링(예컨대, PIPE 또는 CABAC)을 사용하는 엔트로피 코딩을 사용한 제1 분할(140_1)의 경계(142)는 2이다. 이 바람직한 실시예에 대해, 제1 및 제2 이진수 모두에 대한 컨텍스트 모델링은 전술한 바와 같이 H.264/AVC에서 coeff_abs_greater_one에 대해 기술된 대로 행할 수 있다. coeff_abs_greater_one에 대해 기술된 바와 같은 컨텍스트 세트 평가는 제2 이진수에 대해 별도로 행할 수 있다.

[0390] 바람직한 실시예에서, 확률 모델링(예컨대, PIPE 또는 CABAC)을 사용하는 엔트로피 코딩을 사용한 제1 분할(140_1)의 경계(142)는 1일 수 있다. 각각의 소스 심벌의 이진수 스트링의 이진수(또는 알파벳 심벌)에 대해서만, 컨텍스트 모델링은 전술한 바와 같이 H.264/AVC에서 coeff_abs_greater_one에 대해 기술된 대로 행할 수 있다.

[0391] 바람직한 실시예에서, 확률 모델링(예컨대, PIPE 또는 CABAC)을 사용하는 엔트로피 코딩을 사용한 제1 분할(140_1)의 경계(142)는 3일 수 있다. 각각의 소스 심벌의 이진수 스트링의 제1 및 제2 이진수의 컨텍스트 모델링은 H.264/AVC에서 coeff_abs_greater_one처럼 행할 수 있다. 제3 이진수에 대한 컨텍스트 모델링은 H.264/AVC에서 coeff_abs_level_minus_two처럼 행해질 수 있다. coeff_abs_greater_one과 같은 컨텍스트 세트 평가는 제2 이진수에 대해 별도로 행할 수 있다.

[0392] 바람직한 실시예에서, 트렁케이트된 골롬-라이스 코드 세트는 제2 분할(140_2)의 엔트로피 코드로서 사용될 수 있

다. 엔트로피 코드 파라미터에 따라 제3 분할(140_3)의 시작을 명시하는 제2 분할의 경계(144)는 가변적일 수 있다. 또한, 이 바람직한 실시예에서, 골롬-라이스 파라미터는 3으로 제한될 수 있으며 파라미터 선택은 H.264/AVC에서 coeff_abs_level_minus_two에 대한 컨텍스트 모델링처럼 행할 수 있다. 범위 limit-limit2는 가변적일 수 있으며 골롬-라이스 파라미터에 의존할 수 있다. 파라미터가 0이라면, 범위는 8이다. 파라미터 1에 대해, 범위는 10이다. 파라미터 2인 경우, 범위는 12이며, 파라미터 3인 경우, 범위는 16이다. 이 바람직한 실시예에서, 골롬-라이스 파라미터는 변환 계수 블록의 시작에서 0으로 설정된다. 제1 경계 이상인 블록에서의 각 코딩된 변환 계수 레벨에 대해, 대응하는 골롬-라이스 코드를 사용한다. 레벨을 코딩(또는 디코딩)한 후, 제1 경계이상인 그 다음 레벨의 코딩(또는 디코딩)에 대한 골롬-라이스 파라미터를 업데이트하기 위해 다음의 평가가 이뤄진다. 골롬-라이스 파라미터는 이러한 적응 형식을 사용하여 감소할 수는 없음을 주목해야 한다.

[0393] 파라미터 적응 규칙은 다음과 같이 요약할 수 있으며, 여기서 k_{t+1}은 그 다음 레벨 값의 코딩에 사용될 골롬-라이스 파라미터를 표시하며, $value_t$는 대응하는 골롬-라이스 파라미터(k_t)를 갖는 이전 코딩된 값을 표시한다.

$$k_{t+1} = \begin{cases} 0 & value_t \in [0,1] \wedge k_t < 1 \\ 1 & value_t \in [2,3] \wedge k_t < 2 \\ 2 & value_t \in [4,5] \wedge k_t < 3 \\ 3 & value_t > 5 \wedge k_t < 4 \\ k_t & otherwise \end{cases} \qquad (QQ)$$

[0394]

[0395] 바람직한 실시예에서, 트렁케이트된 골롬-라이스 코드의 세트는 제2 분할(140_2)의 엔트로피 세트로서 사용될 수 있다. 엔트로피 코드 파라미터에 따라 제3 분할(140_3)의 시작을 명시하는 제2 분할(140_2)의 경계(144)는 가변적일 수 있다. 또한, 이 바람직한 실시예에서, 골롬-라이스 파라미터는 3으로 제한될 수 있으며, 파라미터 선택은 H.264/AVC에서 coeff_abs_level_minus_two에 대한 컨텍스트 모델링처럼 행할 수 있다. 범위는 가변적일 수 있으며, 골롬-라이스 파라미터에 의존할 수 있다. 파라미터가 0이면, 범위는 8이다. 파라미터 1의 경우, 범위는 10이다. 파라미터 2의 경우, 범위는 12이며 파라미터 3의 경우 범위는 16이다. 이 바람직한 실시예에서, 골롬-라이스 파라미터는 블록의 시작에서 0으로 설정된다. 골롬-라이스 파라미터 적응은 수학식(QQ)에 의해 기술된 대로 실행된다. 파라미터가 이 적응 형태를 사용함으로써 감소할 수 없음을 주목해야 한다.

[0396] 다른 바람직한 실시예에서, 트렁케이트된 골롬-라이스 코드의 세트는 제2 분할(140_2)의 엔트로피 코드로서 사용될 수 있다. 엔트로피 코드 파라미터에 따라 제3 분할(140_3)의 시작을 명시하는 제2 분할(140_2)의 경계(144)는 고정될 수 있다. 또한, 이 바람직한 실시예에서, 골롬-라이스 파라미터는 3으로 제한될 수 있으며, 파라미터 선택은 H.264/AVC에서 coeff_abs_level_minus_two에 대한 컨텍스트 모델링처럼 행할 수 있다. 제2 분할(140_2)의 범위는 14로 고정될 수 있다. 이 바람직한 실시예에서, 골롬-라이스 파라미터는 블록의 시작에서 0으로 설정될 수 있다. 골롬-라이스 파라미터 적응은 수학식(QQ)에 의해 기술된 대로 실행된다. 파라미터가 이 적응 형태를 사용함으로써 감소할 수 없음을 주목해야 한다.

[0397] 다른 바람직한 실시예에서, 트렁케이트된 골롬-라이스 코드의 세트는 제2 분할(140_2)의 엔트로피 세트로서 사용될 수 있다. 엔트로피 코드 파라미터에 따라 제3 분할(140_3)의 시작을 명시하는 제2 분할(140_2)의 경계(144)는 가변적일 수 있다. 또한, 이 바람직한 실시예에서, 골롬-라이스 파라미터는 3으로 제한될 수 있으며, 파라미터 선택은 H.264/AVC에서 coeff_abs_level_minus_two에 대한 컨텍스트 모델링처럼 행할 수 있다. 범위는 가변적일 수 있으며, 골롬-라이스 파라미터에 의존할 수 있다. 파라미터가 0이면, 범위는 8이다. 파라미터 1의 경우, 범위는 10이다. 파라미터 2의 경우, 범위는 12이며 파라미터 3의 경우 범위는 16이다. 이 바람직한 실시예에서, 골롬-라이스 파라미터는 블록의 시작에서 0으로 설정된다. 골롬-라이스 파라미터 적응은 수학식(QQ)에 의해 기술된 대로 실행된다. 파라미터가 이 적응 형태를 사용함으로써 감소할 수 없음을 주목해야 한다. 또한, 예컨대 0으로부터 3으로의 직접 전환도 가능함을 주목해야 한다. 이 바람직한 실시예에서, 골롬-라이스 코드의 프레픽스 부분은 확률 모델을 사용하는 엔트로피 코드로 코딩된다. 컨텍스트 모델링은 H.264/AVC에서 coeff_abs_level_minus_two에 대해서처럼 행할 수 있다.

[0398] 다른 바람직한 실시예에서, 제1 골롬-라이스 파라미터는 현재의 변환 블록에서의 모든 변환 계수 레벨을 코딩하는데 사용될 수 있다. 이 실시예에서, 이전 블록의 최상의 파라미터를 계산할 수 있으며 현재의 변환 블록에 대

해 사용할 수 있다. 이 실시예에 대해, 범위는 14로 고정될 수 있다.

[0399] 다른 바람직한 실시예에서, 고정된 골롬-라이스 파라미터는 현재의 변환 블록에서의 모든 변환 계수 레벨을 코딩하는데 사용될 수 있다. 이 실시예에서, 이전 블록의 최상의 파라미터를 계산할 수 있으며 현재의 변환 블록에 대해 사용할 수 있다. 이 실시예에 대해, 범위는 이전에 기술한 바와 같이 가변적일 수 있다.

[0400] 다른 바람직한 실시예에서, 현재의 스캔 인덱스의 이미 코딩된(또는 디코딩된) 이웃이 이전 경계보다 큰 절대 변환 계수 레벨을 포함하는지를 평가한다. 이 바람직한 실시예에 대해, 최상의 파라미터는 국지적 코즐 템플릿(local causal template)에서 이웃을 사용하여 유도할 수 있다.

[0401] 따라서 상술한 실시예는 특히 엔트로피 인코딩 장치로서, 신택스 요소의 적어도 서브그룹을 각각의 개수(n)의 소스 심벌(s_i)(i=1...n)로 개별적으로 분해함으로써 신택스 요소의 시퀀스(138)를 소스 심벌(106)의 시퀀스(106)로 전환하도록 구성된 분해기(136)-각각의 개수(n)의 소스 심벌은, n개의 분할(140_{1-3}) 중 어떤 시퀀스에, 각각의 신택스 요소의 값 범위가 세분되며 각각의 신택스 요소의 값(z)이 속하는지에 의존하여, 각각의 개수의 소스 심벌(s_i)의 값의 합이 z를 산출하며, n>1인 경우, 모든 i=1...n-1에 대해, s_i의 값이 i번째 분할의 범위에 대응하게 됨-;소스 심벌의 시퀀스(106)를 소스 심벌의 제1 서브시퀀스(108)와 소스 심벌의 제2 서브시퀀스(110)로 세분하여, 모든 소스 심벌(s_x)(x는 {1...n}의 제1 서브세트의 일원임)이 제1 서브시퀀스(108) 내에 포함되고 모든 소스 심벌(s_y)(y는 제1 서브세트와 공통되지 않는 {1...n}의 제2 서브세트의 일원임)이 제2 서브시퀀스(110) 내에 포함되게 하는 서브디바이더(100); 제1 서브시퀀스(108)의 소스 심벌을 심벌-방식으로 인코딩하도록 구성된 VLC 인코더(102); 및 소스 심벌의 제2 서브시퀀스(110)를 인코딩하도록 구성된 PIPE 또는 산술 인코더(104)를 포함하는 엔트로피 인코딩 장치를 기재했다.

[0402] 신택스 요소의 서브그룹의 값(z)은 절대값일 수 있다. 제2 서브세트는 {1}일 수 있고, p번째 분할이 모든 p,q∈{1..n}(p>q)에 대해 q번째 분할보다 더 큰 값의 값의 범위를 커버하도록 배치된 n개의 분할의 시퀀스를 갖는다. 제1 서브세트는 {2, 3}일 수 있으며, VLC 인코더(102)는 소스 심벌(s_2)을 심벌-방식으로 인코딩하도록 골롬-라이스 코드를 사용하고 소스 심벌(s_3)을 심벌-방식으로 인코딩하도록 Exp-골롬 코드를 사용하도록 구성된다. 더 일반적으로, 2는 제1 서브세트의 요소일 수 있으며, VLC 인코더(102)는 이전에 인코딩된 소스 심벌에 따라 골롬-라이스 파라미터, 즉 골롬-라이스 코드의 k를 적응시키고 소스 심벌(s_2)을 심벌-방식으로 인코딩하도록 골롬-라이스 코드를 사용하도록 구성된다. 분해기는 이전에 인코딩된 소스 심벌에 따라 분할 사이의 한계 중 하나 이상을 적응시키도록 구성될 수 있다. 두 적응을 결합할 수 있다. 즉, 제2 분할을 제한하는 한계의 위치는, 골롬-라이스 코드의 길이, 즉 그 코드워드의 수는 제2 분할의 폭의 길이에 대응하도록(또는 규정하도록) 서로로부터 떨어져 있도록 적응될 수 있다. 제1 분할과 제2 분할 사이를 분리하는 한계는 그 밖에 정의에 의해 고정되는 것이나 다른 컨텍스트 의존성에 따라 적응된 것과 같이 정의될 수 있으며, 그 경우, k의 적응은 각각 제2 분할의 폭과 골롬-라이스 코드의 길이를 통한 제2 및 제3 분할을 분리하는 한계의 위치를 정의할 수 있다. 제2 분할의 폭이 골롬-라이스 코드의 길이에 대응하도록 k의 적응을 결합함으로써, 코드 효율은 최적으로 사용된다. k를 신택스 요소의 통계에 적응시키면, 제3 분할이 전체 코딩 복잡도를 감소시키기에 가능한 만큼 커버할 수 있도록 제2 분할의 폭을 적응시킬 수 있으며, 이는 더 낮은 복잡도 코드가 Exp-골롬 코드와 같은 제3 분할에 사용되어야 할 수 있기 때문이다. 또한, 제1 분할의 길이는 최저 세 개의 레벨과 같이 j∈{1, 2, 3}으로 제한될 수 있다. 고려중인 신택스 요소는 차별적으로 코딩될 수 있거나, 예측 잔여를 나타내는 상기 예시적인 변환 계수 레벨을 갖는 경우에서처럼 예측 잔여를 나타낸다. 제1 소스 심벌(s_1)은 트렁케이트된 일진 코드의 사용에 의해 심벌화/디심벌화될 수 있으며, 최종 j개의 이진수는- 부분적으로 또는 이들 모두는- 앞서 언급한 바와 같이 컨텍스트-적응적으로 또는 그렇지 않게 코딩된다.

[0403] 신택스트 요소의 서브그룹은 화상의 변환 블록의 절대 변환 계수의 절대 변환 계수 레벨을 포함하며, 각각의 변환 블록의 절대 변환 계수 레벨은 각각의 변환 블록의 절대 변환 계수를 통해 이어져 있는 스캔 경로에 따라 신택스 요소의 시퀀스(138) 내에 배치되며, 분해기는 스캔 순서로 선행하는 각각의 변환 블록의 절대 변환 계수의 이미 인코딩된 절대 변환 계수 레벨에 의존하거나, 현재 스캔 순서로 분해될 절대 변환 계수 레벨의 위치에 의존하거나 현재 분해될 절대 변환 계수 레벨의 위치에-공간적으로나 스캔 순서로-이웃하는 변환 계수의 이미 재구성된 절대 변환 계수 레벨의 평가를 기반으로 각각의 변환 블록의 절대 변환 계수의 절대 변환 계수 레벨을 분해하는 동안 분할 사이의 한계 중 하나 이상을 적응시키도록 구성될 수 있다.

[0404] 나아가, 전술한 실시예는 특히 엔트로피 디코딩 장치로서, 소스 심벌의 제1 서브시퀀스(204)의 소스 심벌을 제1

비트스트림(206)의 코드워드로부터 코드워드-방식으로 재구성하도록 구성되는 VLC 디코더(200); 소스 심벌의 제2 서브시퀀스(208)를 재구성하도록 구성되는 PIPE 또는 산술 디코더(202); 각각의 수의 소스 심벌로부터 각 신택스 요소를 개별적으로 구성함으로써 소스 심벌의 제1 서브시퀀스(204)와 소스 심벌의 제2 서브시퀀스(208)로부터 신택스 요소의 시퀀스(226)를 구성하도록 구성된 구성기(224)로서, 각 구성기는, s_i의 값이 신택스 요소(z)의 값을 얻기 위해 i번째 분할의 범위에 대응하는 한, 소스 심벌(s_i)의 각각의 개수의 값을 1부터 n까지 합함으로써, 신택스 요소의 적어도 서브그룹에 대해, n개의 분할(140_{1-3})의 시퀀스 중 어떤 시퀀스에, 각각의 신택스 요소의 값 범위가 세분되고 각각의 신택스 요소의 값(z)이 속하는지에 따라, 소스 심벌(s_i)의 각각의 개수(n)(i=1...n)를 결정하도록 구성되며, 구성기(224)는 제1 서브시퀀스(204)로부터 모든 소스 심벌(s_x)(x는 {1...n}의 제1 서브세트 중 일원임)과 제2 서브시퀀스(208)로부터 모든 소스 심벌(s_y)(y는 제1 서브세트와 공통되지 않는 {1...n}의 제2 서브세트 중 일원임)을 회수하도록 구성되는, 구성기(224)를 포함하는 엔트로피 디코딩 장치를 기재한다. 신택스 요소의 서브그룹의 값(z)은 절대값일 수 있다. 제2 서브세트는 {1}일 수 있으며, n개의 분할의 시퀀스는 p번째 분할이 모든 p,q∈{1..n}(p>q)에 대해 q번째 분할보다 더 큰 값의 값의 범위를 커버하도록 배치된다. 제1 서브세트는 {2, 3}일 수 있으며, VLC 디코더(200)는 소스 심벌(s_2)을 코드워드-방식으로 인코딩하도록 골롬-라이스 코드를 사용하고 소스 심벌(s_3)을 코드워드-방식으로 인코딩하도록 Exp-골롬 코드를 사용하도록 구성된다. 더 일반적으로, 2는 제1 서브세트의 요소일 수 있으며, VLC 디코더(102)는 이전에 재구성된 소스 심벌에 따라 골롬-라이스 코드의 골롬-라이드 파라미터를 적응시키고 소스 심벌(s_2)을 코드워드-방식으로 재구성하도록 골롬-라이스 코드를 사용하도록 구성된다. 엔트로피 디코딩 장치는 소스 심벌의 제1 서브시퀀스(204)와 소스 심벌의 제2 서브시퀀스를 재결합하여 소스 심벌의 시퀀스(218)를 얻도록 구성된 재결합기(220)를 더 포함할 수 있다. 신택스 요소는 상이한 타입일 수 있으며, 구성기는 신택스 요소의 타입에 따라 개별 구성을 실행하도록 구성될 수 있다. 신택스 요소의 서브그룹은 화상의 변환 블록의 절대 변환 계수의 절대 변환 계수 레벨을 포함할 수 있으며, 각각의 변환 블록의 절대 변환 계수 레벨은 각각의 변환 블록의 절대 변환 계수를 통해 이어지는 스캔 경로에 따라 신택스 요소의 시퀀스(138) 내에 배치되며, 구성기는 스캔 순서에서 선행하는 각각의 변환 블록의 절대 변환 계수의 이미 재구성된 절대 변환 계수 레벨에 의존하거나, 스캔 순서로 현재 구성될 절대 변환 계수 레벨의 위치에 의존하거나, 현재 구성될 절대 변환 계수 레벨의 위치에-공간적으로나 스캔 순서로-이웃하는 변환 계수의 이미 재구성된 절대 변환 계수 레벨의 평가를 기반으로 각각의 변환 블록의 절대 변환 계수의 절대 변환 계수 레벨을 구성하는 동안 분할 사이의 한계 중 하나 이상을 적응시키도록 구성될 수 있다.

[0405] 도 1b에 따른 분해를 사용하여 PIPE 코딩을 VLC 코딩과 결합하는 실시예에 관해, 달리 말해 그 일부 양상을 반복하기 위해 다음의 설명을 주목해야 한다.

[0406] 심벌 시퀀스의 비트스트림으로의 매핑 및 역 매핑을 기술하였다. 각 심벌은 인코더 및 디코더에 즉시 알려지는 관련 파라미터(들)를 반송한다. 엔트로피 코덱은 심벌과 관련된 파라미터(들) 서브세트에 각각 할당되는 복수의 선입선출(FIFO) 버퍼를 포함한다. 심벌의 소정의 파라미터(들)에 대해, 인코더는 심벌을 대응하는 FIFO 버퍼에 할당한다. 인코더 할당 규칙이 디코더 측에 알려져 있으므로, 디코더는, 인코더가 심벌을 할당했던 FIFO 버퍼로부터 판독한다.

[0407] 일부 신택스 요소는 표준 가변 길이 코드를 사용하여 코딩되어 특정한 버퍼에 기록된다. 다른 신택스 요소는 확률 간격 분할 엔트로피(PIPE: Probability Interval Partitioning Entropy) 코딩 컨셉을 이용하여 코딩된다. 이때, 심벌이 먼저 이진화되고 결과적인 이진수가 관련 확률 추정을 기반으로 하여 분류된다. 확률 추정은 인코더 및 디코더에서 즉시 진행될 수 있는 측정으로부터 주어지거나 유도할 수 있다. 특정한 FIFO 버퍼는 선택된 확률 서브세트에 속한 추정된 확률 값을 갖는 심벌을 포함하여, 엔트로피 코딩은 개선될 수 있다. PIPE 컨셉의 VLC와의 조합에 의해 달성한 개선은 높은 코딩 효율을 여전히 제공하면서도 복잡성을 감소시킨다는 점이다. 표준 VLC 코드가 적절한 심벌은 간단하고 낮은 복잡도의 VLC 접근법으로 코딩되는 반면, VLC 코드로 코딩하면 비트율이 상당히 증가하게 될 다른 심벌은 더 정교한 PIPE 컨셉으로 코딩된다.

[0408] 따라서, 엔트로피 코딩의 복잡도를 더 감소시키기 위해, 심벌은 두 개의 카테고리로 분할되었다. 제1 카테고리의 심벌은 VLC 코드로 충분히 표현될 수 있으며, 더 복잡한 PIPE 코딩을 필요치 않는 반면, 제2 카테고리의 심벌은 VLC 코드로 효율적으로 표현될 수 없으며 이들 심벌에 대한 PIPE 코딩은 필요한 비트율을 상당히 감소시켰다.

[0409] 상기 실시예는 하나의 VLC 인코더와 디코더로만 기재되었지만, 이 컨셉은 둘 이상이 VLC 인코더 및 디코더의 사용으로 PIPE 코더와 인터리빙되도록 일반화될 수 있음이 명백함을 주목해야 한다.

[0410] 일부 양상이 장치의 환경에서 기재되었을지라도, 이들 양상은 또한 대응하는 방법의 기재를 나타내며, 여기서 블록이나 장치는 방법 단계나 방법 단계의 특성에 대응한다. 유사하게, 방법 단계의 환경에서 기재된 양상도 대응하는 장치의 대응하는 블록이나 구성장치나 특성의 기재를 나타낸다. 방법 단계 중 일부나 모두는, 예컨대 마이크로프로세서, 프로그램 가능한 컴퓨터, 또는 전자 회로와 같은 하드웨어 장치에 의해(또는 이것을 이용하여) 수행될 수 있다. 일부 실시예에서, 가장 중요한 방법 단계 중 일부 하나 이상은 그러한 장치에 의해 수행될 수 있다.

[0411] 본 발명의 인코딩된/압축된 신호는 디지털 저장 매체 상에 저장될 수 있거나, 인터넷 등의 무선 송신 매체나 유선 송신 매체와 같은 송신 매체 상에서 송신될 수 있다.

[0412] 특정 구현 요건에 따라, 본 발명의 실시예는 하드웨어나 소프트웨어로 구현될 수 있다. 그러한 구현은 예컨대 그 내부에 전자적으로 판독 가능한 제어 신호를 갖는 플로피 디스크, DVD, 블루레이, CD, ROM, PROM, EPROM, EEPROM 또는 FLASH 메모리와 같은 디지털 저장 매체를 사용하여 실행될 수 있으며, 그러한 매체는 각각의 방법이 실행되는 프로그램 가능한 컴퓨터 시스템과 협력한다(또는 그러한 시스템과 협력할 수 있다). 그러므로 디지털 저장 매체는 컴퓨터로 판독 가능할 수 있다.

[0413] 본 발명에 따른 일부 실시예는 전자적으로 판독 가능한 제어 신호를 갖는 데이터 캐리어를 포함하며, 그러한 캐리어는 본 명세서에서 기재한 방법 중 하나가 실행되도록 프로그램 가능한 컴퓨터 시스템과 협력할 수 있다.

[0414] 일반적으로, 본 발명의 실시예는 프로그램 코드를 갖는 컴퓨터 프로그램 제품으로서 구현될 수 있으며, 프로그램 코드는, 컴퓨터 프로그램 제품이 컴퓨터 상에 운영될 때 방법 중 하나를 실행하도록 동작한다. 프로그램 코드는 예컨대 기계로 판독 가능한 캐리어 상에 저장될 수 있다.

[0415] 다른 실시예는, 기계로 판독 가능한 캐리어 상에 저장된, 본 명세서 기재한 방법 중 하나를 실행하는 컴퓨터 프로그램을 포함한다.

[0416] 다시 말해, 본 발명의 방법의 실시예는, 그러므로, 컴퓨터 프로그램이 컴퓨터 상에 실행될 때, 본 명세서에서 기재된 방법 중 하나를 실행하기 위한 프로그램 코드를 갖는 컴퓨터 프로그램이다.

[0417] 본 발명의 방법의 다른 실시예는, 그러므로, 본 명세서에 기재한 방법 중 하나를 실행하는 컴퓨터 프로그램을 그 내부에 기록하여 포함하는 데이터 캐리어(또는 디지털 저장 매체, 또는 컴퓨터로 판독 가능한 매체)이다.

[0418] 본 발명의 방법의 다른 실시예는 그러므로 본 명세서에 기재된 방법 중 하나를 실행하는 컴퓨터 프로그램을 나타내는 데이터 스트림이나 신호 시퀀스이다. 데이터스트림이나 신호 시퀀스는 예컨대 인터넷 등을 통해서와 같은 데이터 통신 연결을 통해서 전송되도록 구성될 수 있다.

[0419] 다른 실시예는 예컨대 본 명세서에서 기재한 방법 중 하나를 실행하도록 구성되거나 적응된 컴퓨터나 프로그램 가능한 로직 장치와 같은 처리 수단을 포함한다.

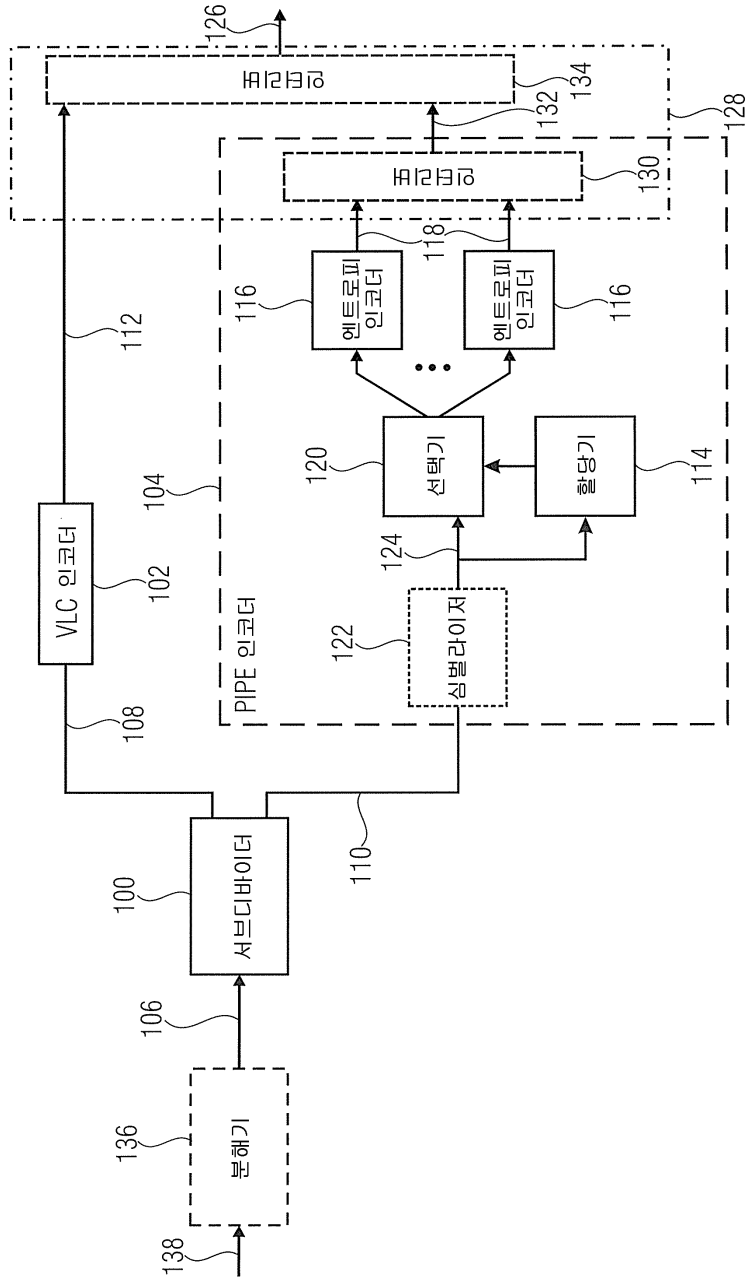
[0420] 다른 실시예는 본 명세서에 기재한 방법 중 하나를 실행하는 컴퓨터 프로그램을 그 내부에 설치한 컴퓨터를 포함한다.

[0421] 일부 실시예에서, 프로그램 가능한 로직 장치(예컨대, 필드 프로그램 가능한 게이트 어레이)는 본 명세서에 기재한 방법의 기능 중 일부나 모두를 실행하는데 사용될 수 있다. 일부 실시예에서, 필드 프로그램 가능한 게이트 어레이는 마이크로프로세서와 협력하여 본 명세서에 기재된 방법 중 하나를 실행할 수 있다. 일반적으로, 방법은 임의의 하드웨어 장치에 의해 실행된다.

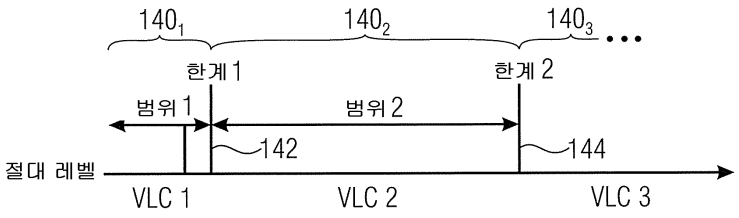
[0422] 상술한 실시예는 본 발명의 원리에 대한 단순한 예시이다. 본 명세서에 기재한 장치 및 세부 내용의 변경 및 수정은 당업자에게 명백할 것임을 이해해야 한다. 그러므로, 본 명세서의 실시예의 기재 및 설명에 의해 제공된 특정한 세부내용에 의해서가 아니라 수반한 특허청구범위의 범위에 의해서만 제한될 것이다.

도면

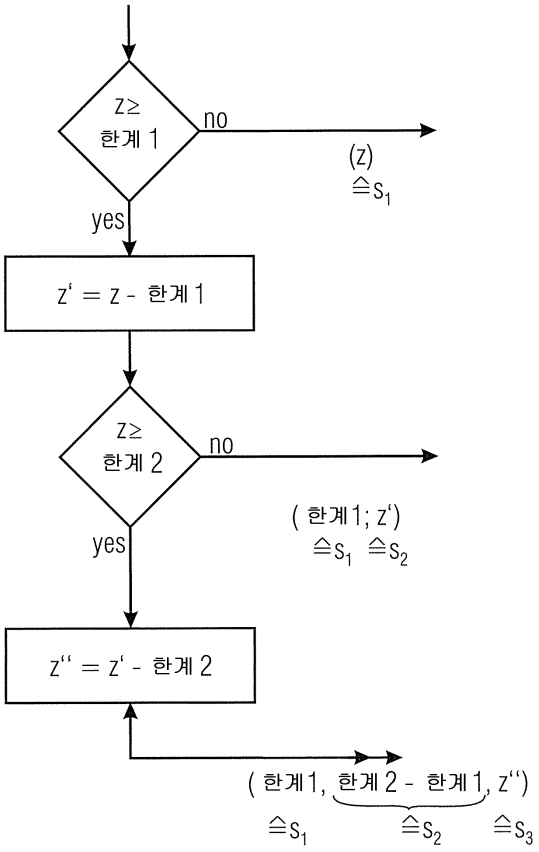
도면1a


126
버퍼타입
134
132
128
버퍼타입
130
118
엔트로피 인코더
116
엔트로피 인코더
116
112
104
120
선택기
할당기
114
124
VLC 인코더
102
PIPE 인코더
122
심벌라이저
108
110
100
서브디바이더
106
136
분해기
138

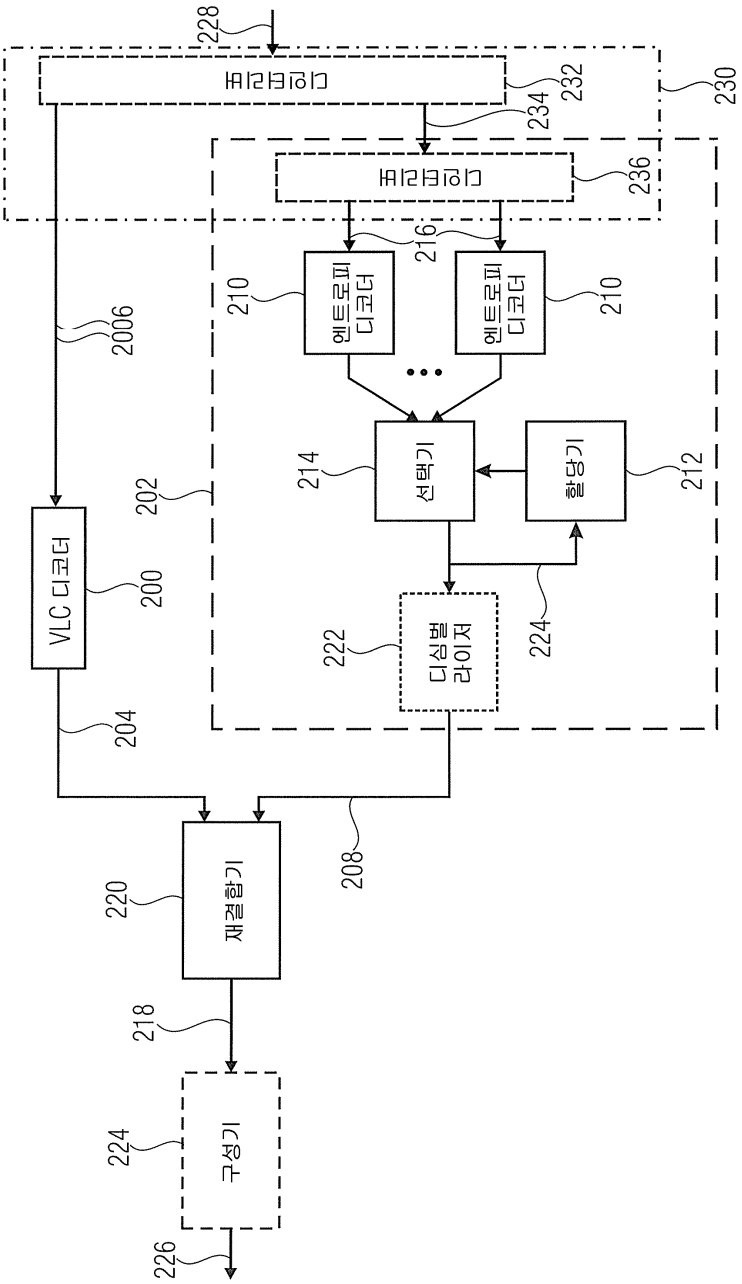
도면1b


140_1
140_2
140_3
한계 1
한계 2
범위 1
범위 2
142
144
절대 레벨
VLC 1
VLC 2
VLC 3

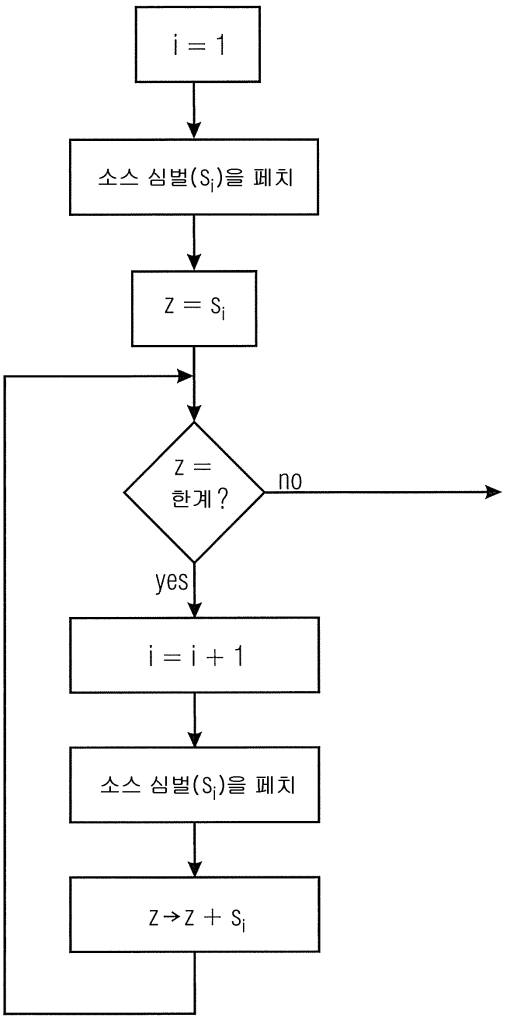
도면1c


z≥
한계 1
no
(z)
≙s_1
yes
z' = z - 한계 1
z≥
한계 2
no
(한계1; z')
≙s_1 ≙s_2
yes
z'' = z' - 한계 2
(한계1, 한계 2 - 한계 1, z'')
≙s_1
≙s_2
≙s_3

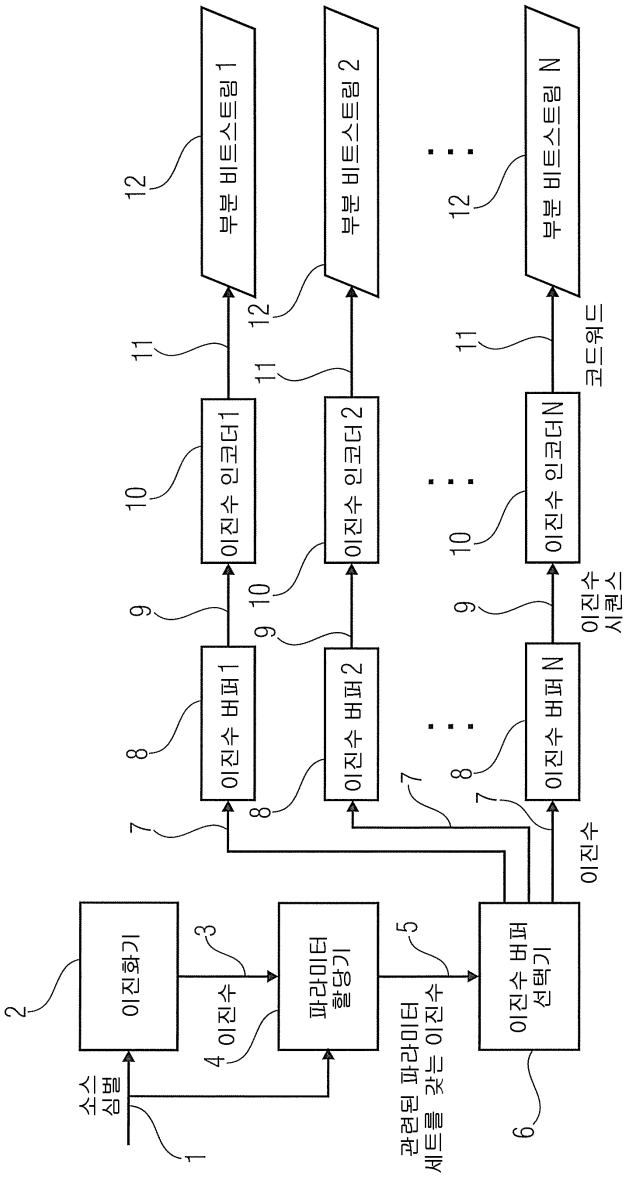
도면2a


228
버퍼관리기
232
234
230
버퍼관리기
236
216
엔트로피 디코더
210
엔트로피 디코더
210
2006
214
선택기
할당기
212
202
VLC 디코더
200
222
디심볼 라이저
224
204
208
220
재결합기
218
224
구성기
226

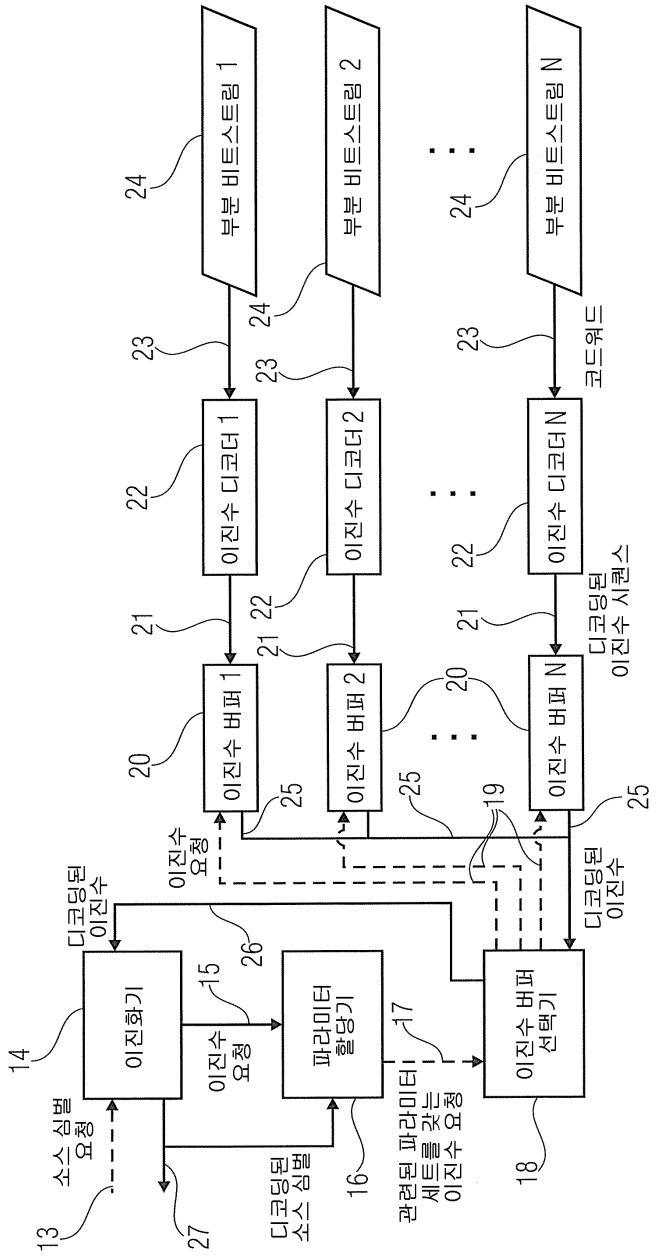
도면2b



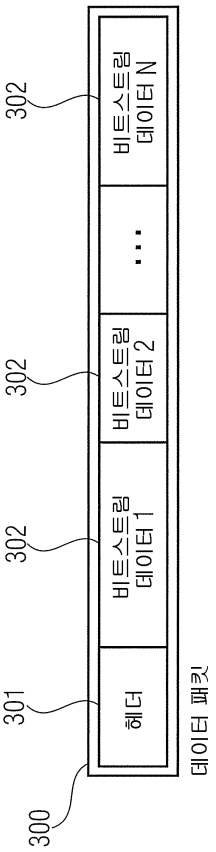
도면3


소스 심벌
1
이진화기
2
이진수
3
4
파라미터 할당기
관련된 파라미터 세트를 갖는 이진수
5
이진수 버퍼 선택기
6
이진수
7
이진수 버퍼 1
8
이진수 버퍼 2
이진수 버퍼 N
9
이진수 시퀀스
10
이진수 인코더1
이진수 인코더 2
이진수 인코더N
11
코드워드
12
부분 비트스트림 1
부분 비트스트림 2
부분 비트스트림 N

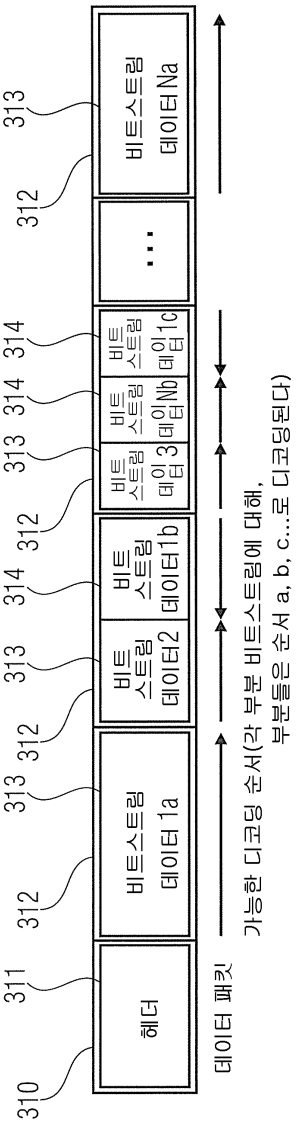
도면4


부분 비트스트림 1
부분 비트스트림 2
부분 비트스트림 N
코드워드
이진수 디코더 1
이진수 디코더 2
이진수 디코더 N
디코딩된 이진수 시퀀스
이진수 버퍼 1
이진수 버퍼 2
이진수 버퍼 N
이진수 요청
디코딩된 이진수
이진화기
이진수 요청
파라미터 할당기
이진수 버퍼 선택기
소스 심볼 요청
디코딩된 소스 심볼
관련된 파라미터 세트를 갖는 이진수 요청
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27

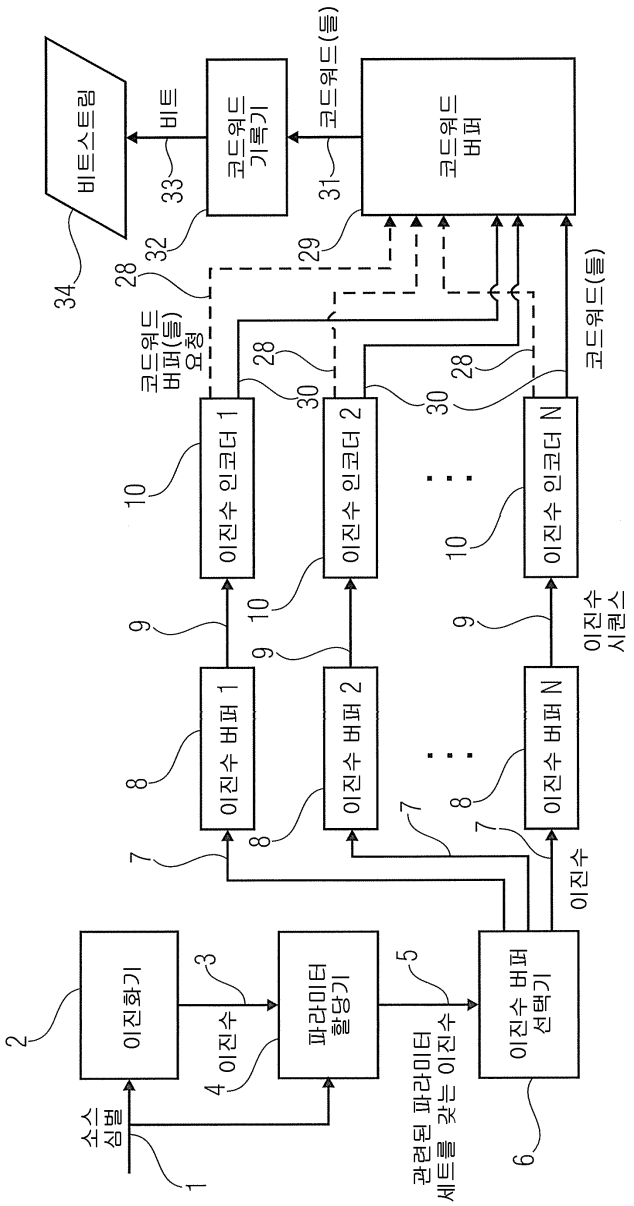
도면5



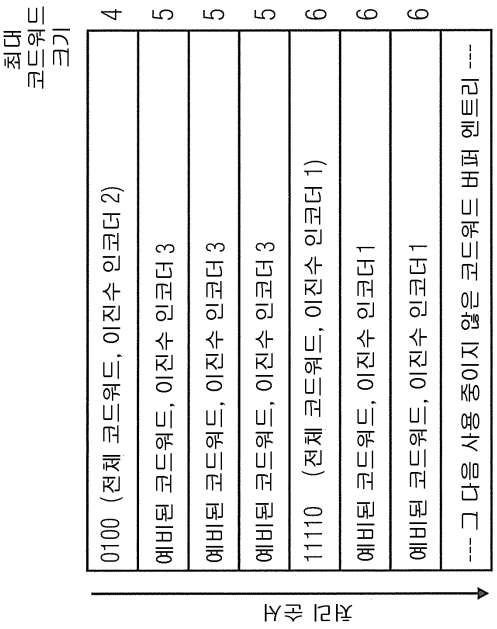
도면6


310
311
312
313
314
헤더
비트스트림 데이터 1a
비트 스트림 데이터2
비트 스트림 데이터1b
비트 스트림 데이터 3
비트 스트림 데이터Nb
비트 스트림 데이터1c
...
비트스트림 데이터 Na
데이터 패킷
가능한 디코딩 순서(각 부분 비트스트림에 대해,
부분들은 순서 a, b, c...로 디코딩된다)

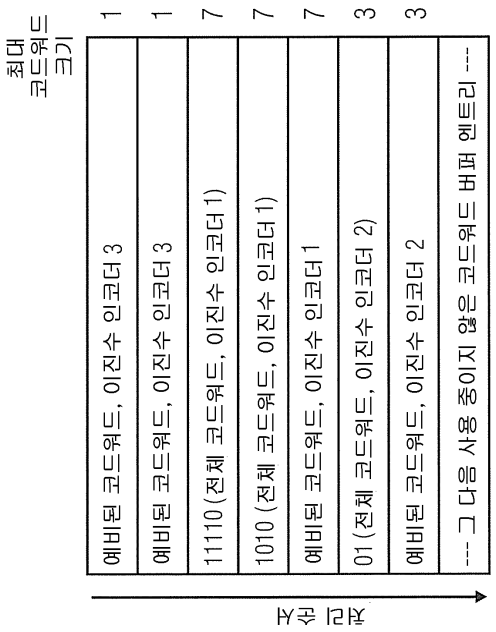
도면7


소스 심벌
이진화기
파라미터 할당기
이진수 버퍼 선택기
관련된 파라미터 세트를 갖는 이진수
이진수
이진수 버퍼 1
이진수 버퍼 2
이진수 버퍼 N
이진수 시퀀스
이진수 인코더 1
이진수 인코더 2
이진수 인코더 N
코드워드 버퍼(들) 요청
코드워드(들)
코드워드 버퍼
코드워드 기록기
비트
비트스트림

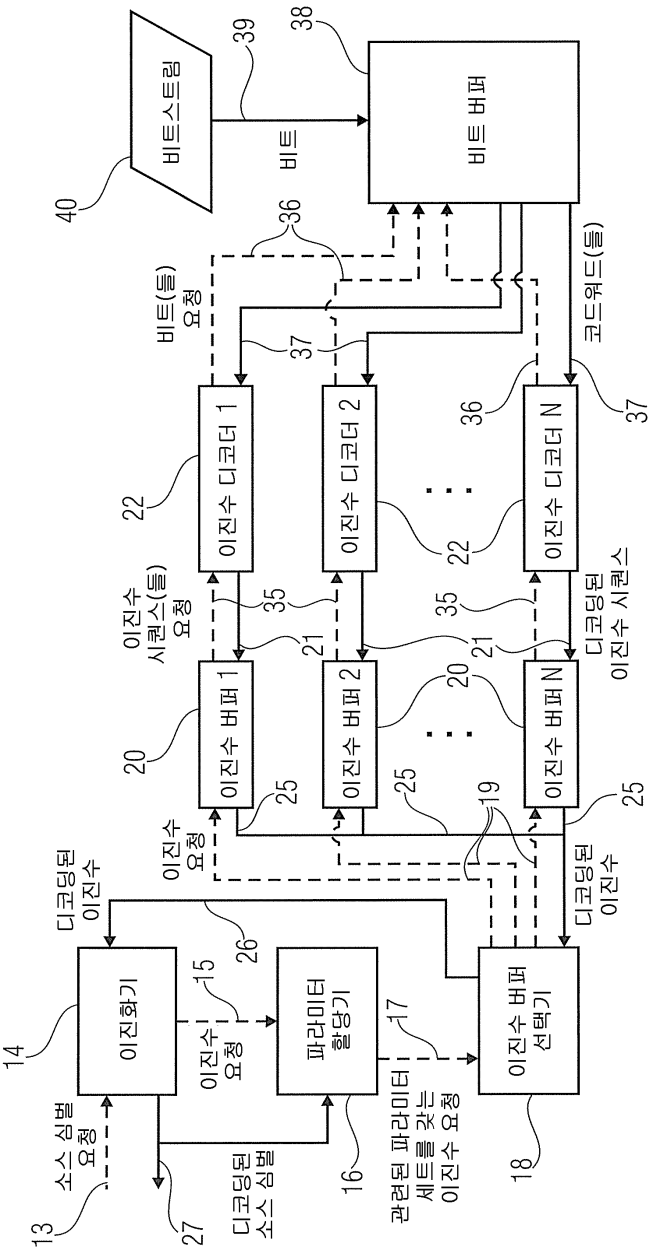
도면8a


최대 코드워드 크기
4
5
5
5
6
6
6
0100 (전체 코드워드, 이진수 인코더 2)
예비된 코드워드, 이진수 인코더 3
예비된 코드워드, 이진수 인코더 3
예비된 코드워드, 이진수 인코더 3
11110 (전체 코드워드, 이진수 인코더 1)
예비된 코드워드, 이진수 인코더 1
예비된 코드워드, 이진수 인코더 1
--- 그 다음 사용 중이지 않은 코드워드 버퍼 엔트리 ---
처리 순서

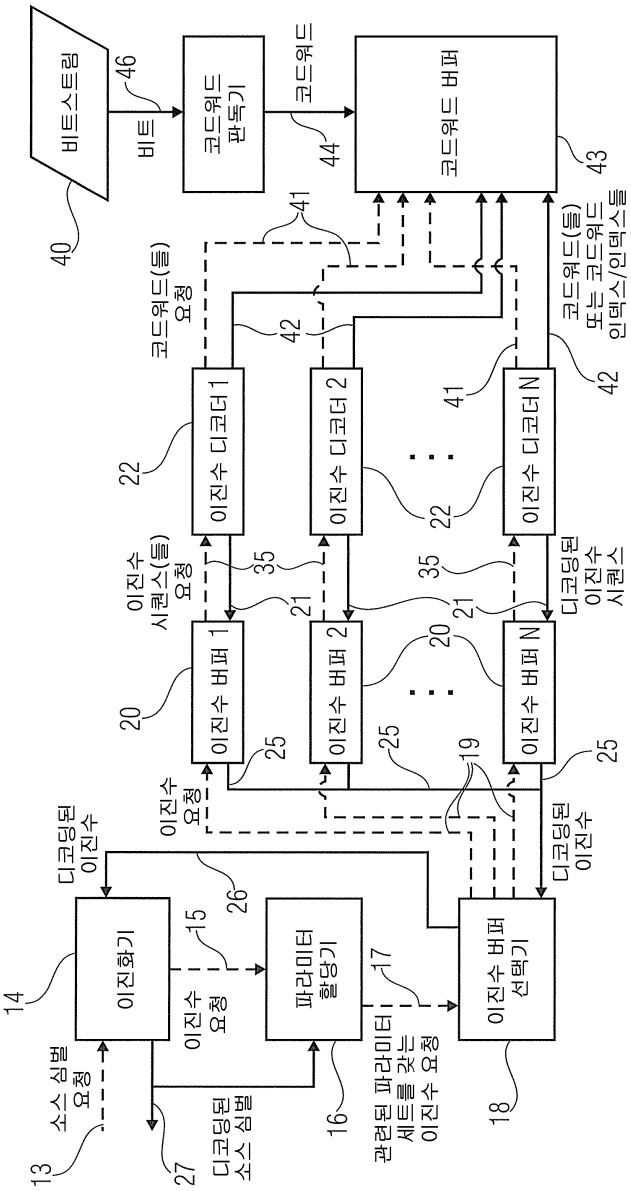
도면8b


최대 코드워드 크기
1
1
7
7
7
3
3
예비된 코드워드, 이진수 인코더 3
예비된 코드워드, 이진수 인코더 3
11110 (전체 코드워드, 이진수 인코더 1)
1010 (전체 코드워드, 이진수 인코더 1)
예비된 코드워드, 이진수 인코더 1
01 (전체 코드워드, 이진수 인코더 2)
예비된 코드워드, 이진수 인코더 2
--- 그 다음 사용 중이지 않은 코드워드 버퍼 엔트리 ---
처리 순서

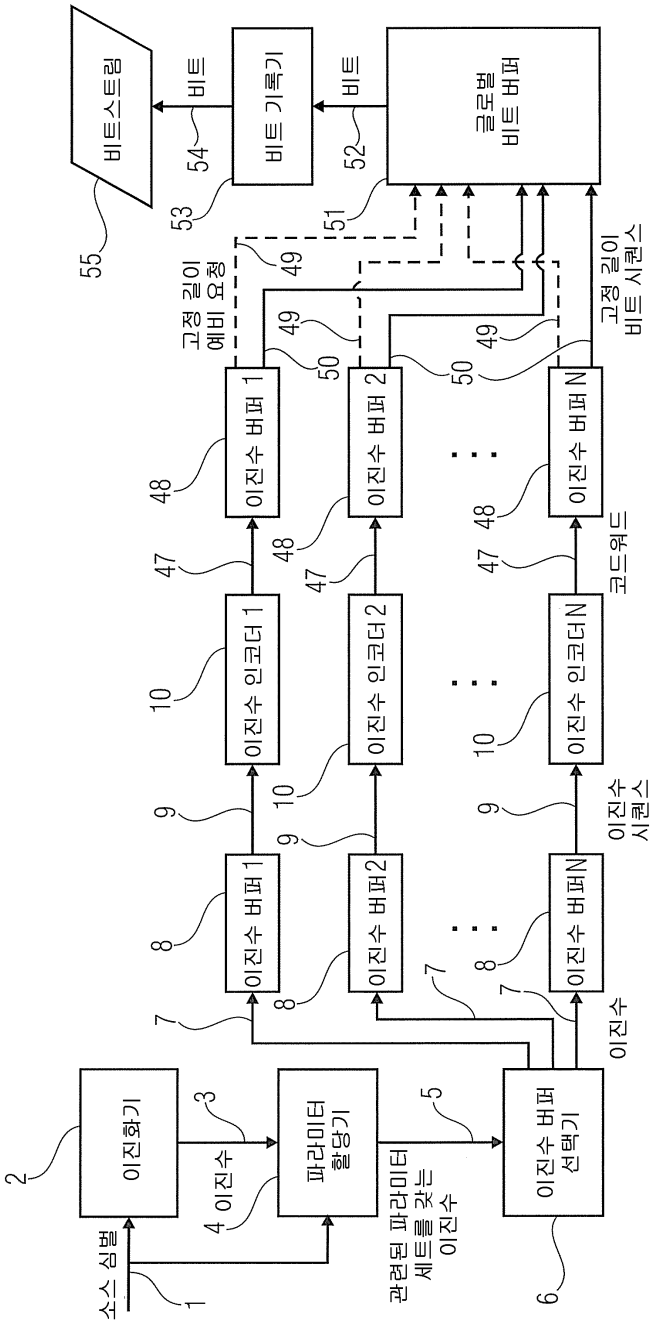
도면9


비트스트림
비트
비트 버퍼
비트(들) 요청
코드워드(들)
이진수 디코더 1
이진수 디코더 2
이진수 디코더 N
이진수 시퀀스(들) 요청
디코딩된 이진수 시퀀스
이진수 버퍼 1
이진수 버퍼 2
이진수 버퍼 N
이진수 요청
디코딩된 이진수
디코딩된 이진수
이진화기
이진수 요청
파라미터 할당기
이진수 버퍼 선택기
소스 심벌 요청
디코딩된 소스 심벌
관련된 파라미터 세트를 갖는 이진수 요청

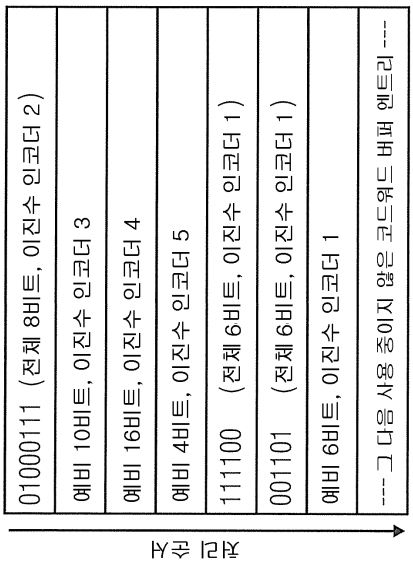
도면10


비트스트림
40
비트
46
코드워드 판독기
코드워드
44
코드워드 버퍼
43
41
코드워드(들) 요청
42
코드워드(들) 또는 코드워드 인덱스/인덱스들
이진수 디코더 1
이진수 디코더 2
이진수 디코더 N
22
22
41
42
이진수 시퀀스(들) 요청
35
35
21
21
디코딩된 이진수 시퀀스
이진수 버퍼 1
이진수 버퍼 2
이진수 버퍼 N
20
20
25
25
25
19
이진수 요청
디코딩된 이진수
디코딩된 이진수
26
이진화기
15
파라미터 할당기
17
이진수 버퍼 선택기
14
소스 심벌 요청
이진수 요청
13
27
디코딩된 소스 심벌
16
관련된 파라미터 세트를 갖는 이진수 요청
18

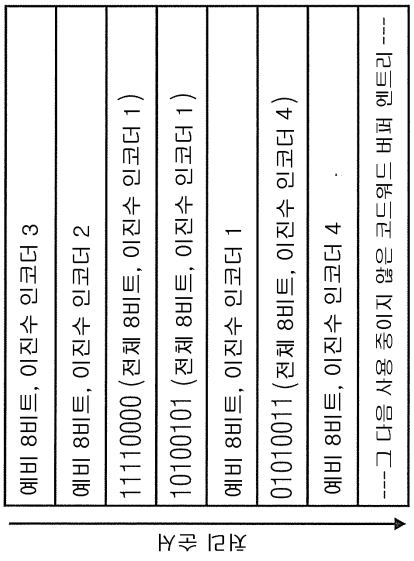
도면11


비트스트림
비트
비트 기록기
비트
글로벌 비트 버퍼
55
54
53
52
51
고정 길이 예비 요청
49
고정 길이 비트 시퀀스
이진수 버퍼 1
이진수 버퍼 2
이진수 버퍼 N
48
47
코드워드
이진수 인코더 1
이진수 인코더 2
이진수 인코더 N
10
9
이진수 시퀀스
이진수 버퍼1
이진수 버퍼2
이진수 버퍼N
8
7
이진수
이진화기
파라미터 할당기
이진수 버퍼 선택기
2
3
4 이진수
5
관련된 파라미터 세트를 갖는 이진수
6
소스 심벌
1

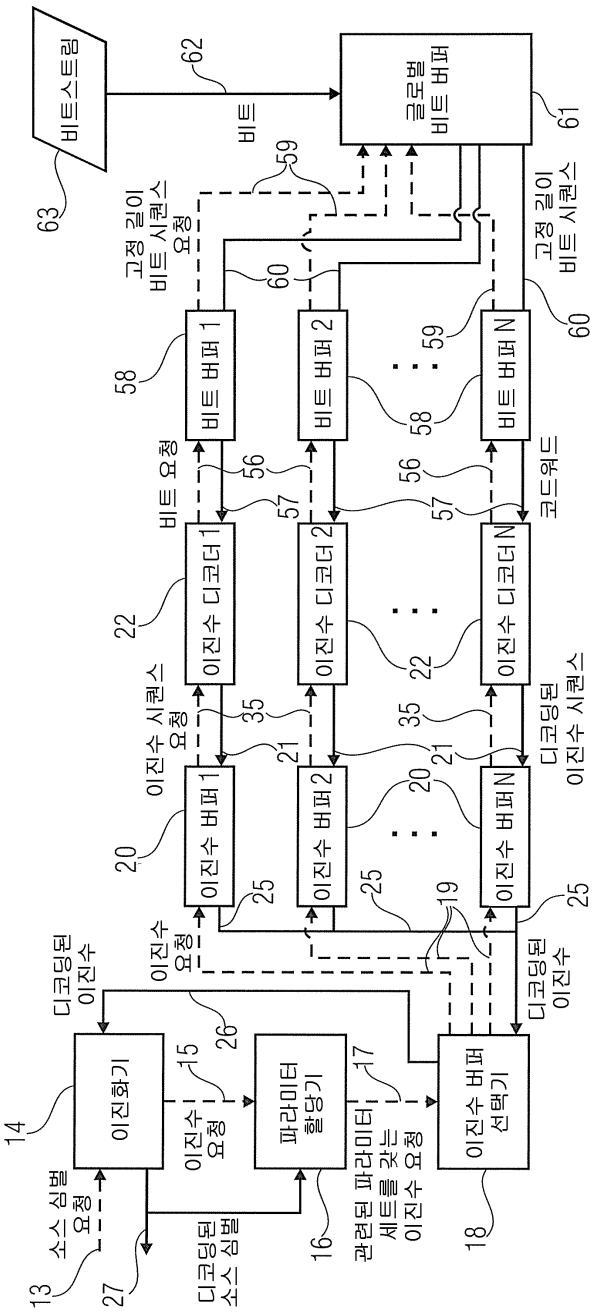
도면12a


01000111 (전체 8비트, 이진수 인코더 2)
예비 10비트, 이진수 인코더 3
예비 16비트, 이진수 인코더 4
예비 4비트, 이진수 인코더 5
111100 (전체 6비트, 이진수 인코더 1)
001101 (전체 6비트, 이진수 인코더 1)
예비 6비트, 이진수 인코더 1
--- 그 다음 사용 종이지 않은 코드워드 버퍼 엔트리 ---
전달 방식

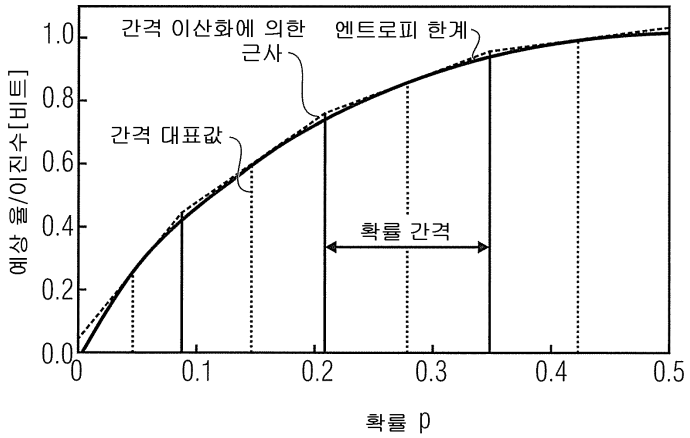
도면12b


예비 8비트, 이진수 인코더 3
예비 8비트, 이진수 인코더 2
11110000 (전체 8비트, 이진수 인코더 1)
10100101 (전체 8비트, 이진수 인코더 1)
예비 8비트, 이진수 인코더 1
01010011 (전체 8비트, 이진수 인코더 4)
예비 8비트, 이진수 인코더 4
--- 그 다음 사용 종이지 않은 코드워드 버퍼 엔트리 ---
전달 순서

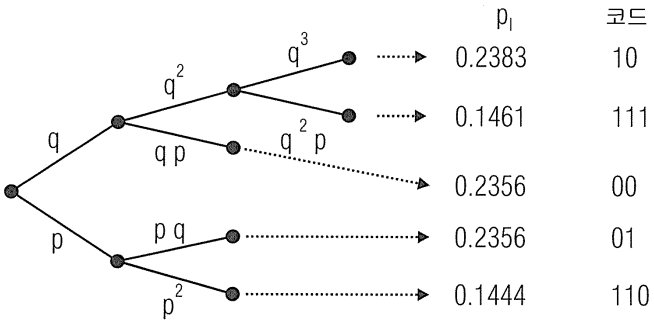
도면13


비트스트림
63
62
비트
글로벌 비트 버퍼
61
고정 길이 비트 시퀀스 요청
59
60
고정 길이 비트 시퀀스
비트 버퍼 1
비트 버퍼 2
비트 버퍼 N
58
비트 요청
56
57
코드워드
이진수 디코더 1
이진수 디코더 2
이진수 디코더 N
22
이진수 시퀀스 요청
35
21
디코딩된 이진수 시퀀스
이진수 버퍼 1
이진수 버퍼 2
이진수 버퍼 N
20
25
19
이진수 요청
디코딩된 이진수
26
이진화기
15
14
파라미터 할당기
17
이진수 버퍼 선택기
18
소스 심벌 요청
13
27
디코딩된 소스 심벌
16
관련된 파라미터 세트를 갖는 이진수 요청

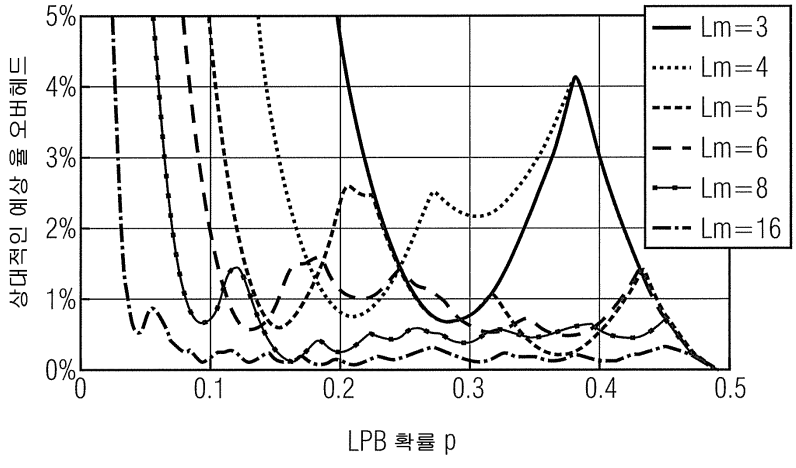
도면14



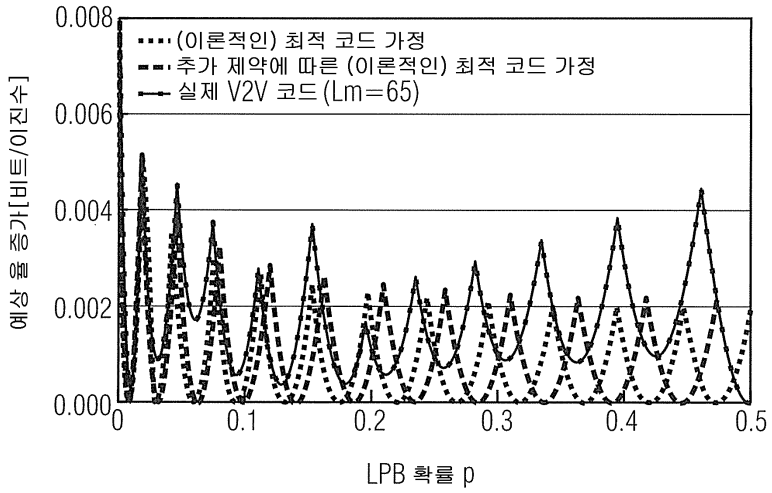
도면15



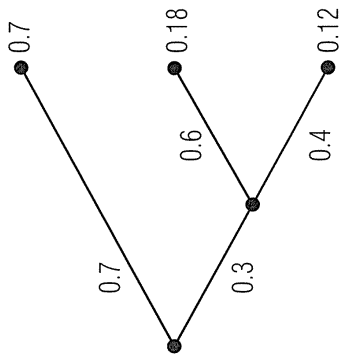
도면16

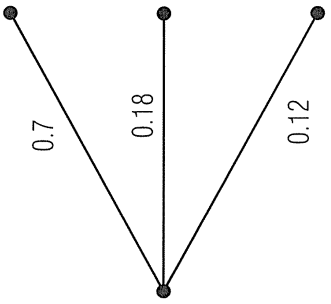


도면17

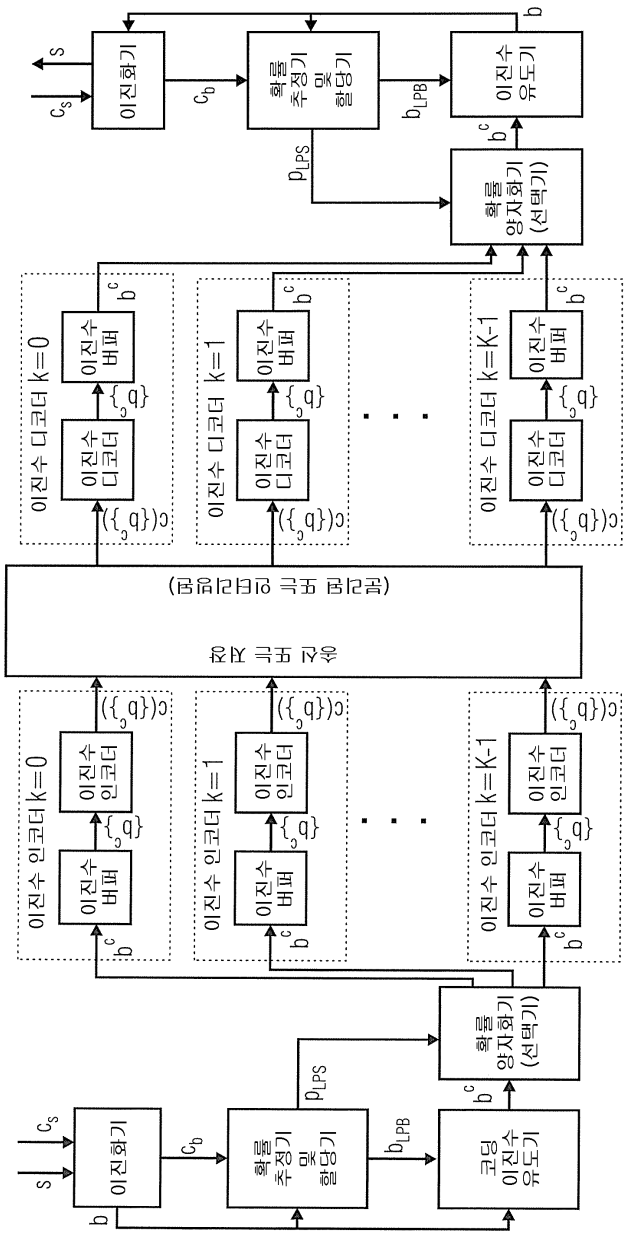

0.008
0.006
0.004
0.002
0.000
0
0.1
0.2
0.3
0.4
0.5
(이론적인) 최적 코드 가정
추가 제약에 따른 (이론적인) 최적 코드 가정
실제 V2V 코드 (Lm=65)
예상 율 증가[비트/이진수]
LPB 확률 p

도면18

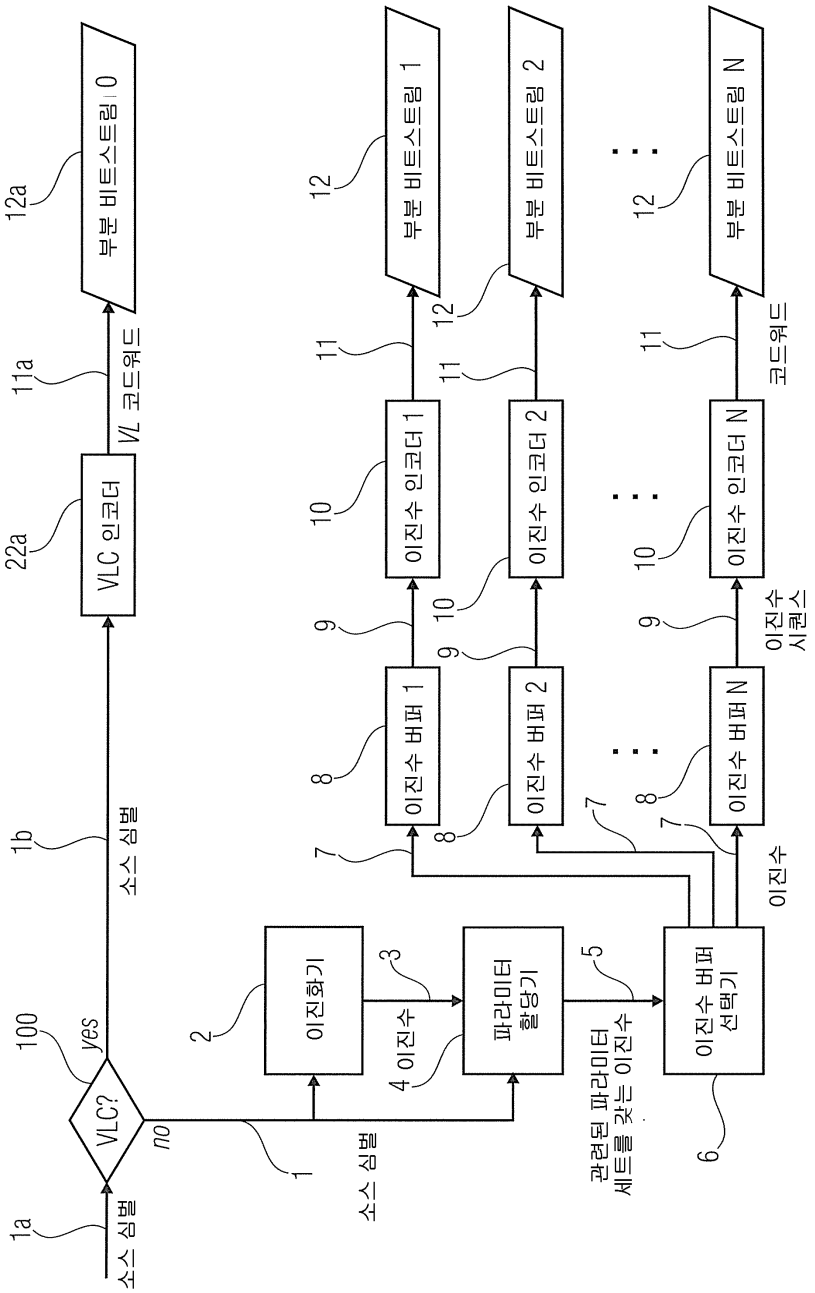

0.7
0.18
0.12
0.6
0.4
0.7
0.3


0.7
0.18
0.12

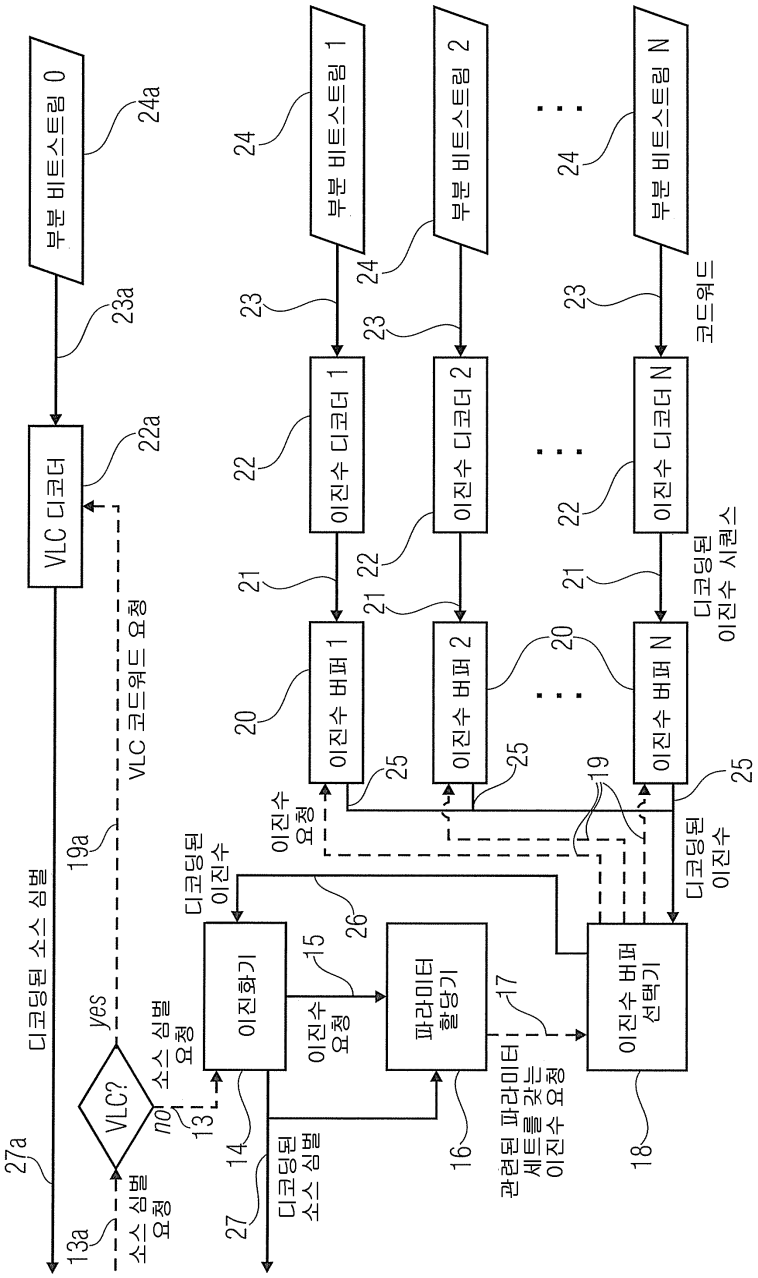
도면19


이진화기
확률 추정기 및 할당기
이진수 유도기
확률 양자화기 (선택기)
이진수 디코더 k=0
이진수 디코더 k=1
이진수 디코더 k=K-1
이진수 버퍼
이진수 디코더
이진수 인코더 k=0
이진수 인코더 k=1
이진수 인코더 k=K-1
이진수 버퍼
이진수 인코더
코딩 이진수 유도기
확률 양자화기 (선택기)
이진화기
확률 추정기 및 할당기
이진수 유도기

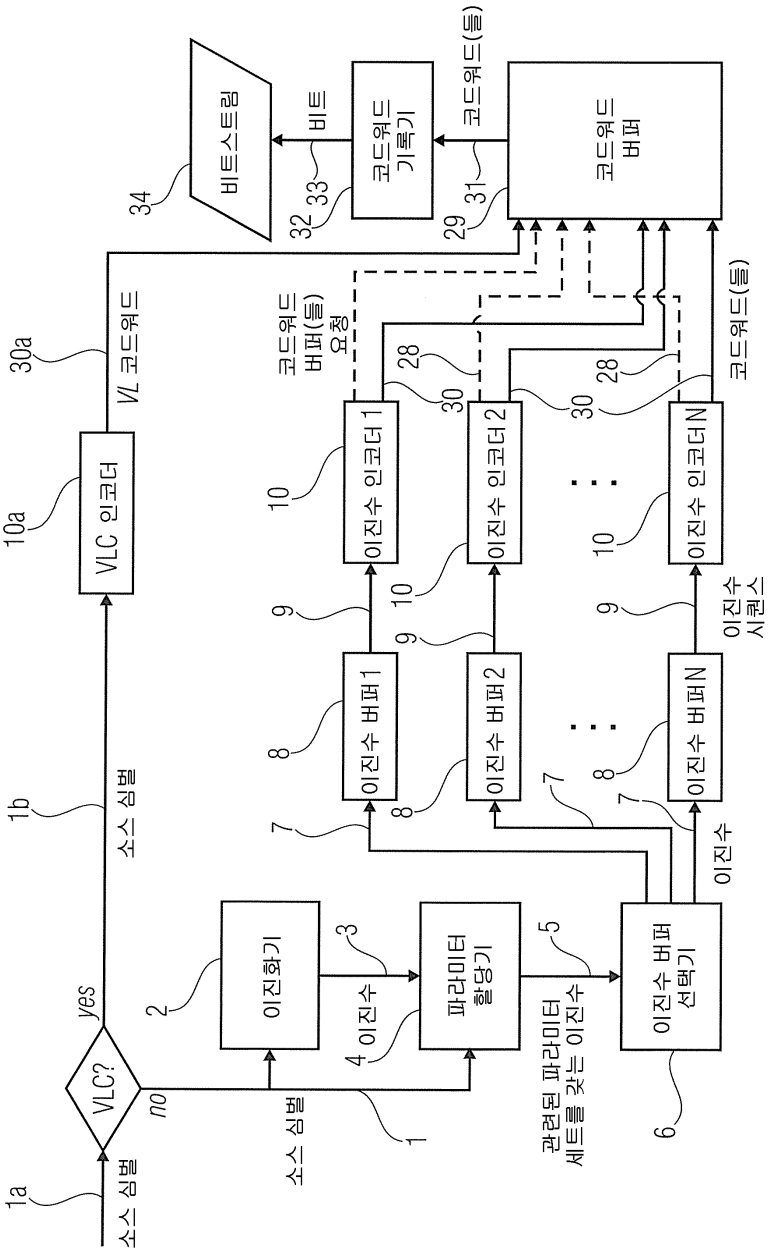
도면20


1a
소스 심벌
100
VLC?
yes
no
1b
소스 심벌
1
소스 심벌
2
이진화기
3
4 이진수
파라미터 할당기
5
관련된 파라미터 세트를 갖는 이진수
6
이진수 버퍼 선택기
7
이진수
8
이진수 버퍼 1
이진수 버퍼 2
이진수 버퍼 N
9
이진수 시퀀스
10
이진수 인코더 1
이진수 인코더 2
이진수 인코더 N
11
코드워드
12
부분 비트스트림 1
부분 비트스트림 2
부분 비트스트림 N
11a
VL 코드워드
12a
부분 비트스트림 0
22a
VLC 인코더

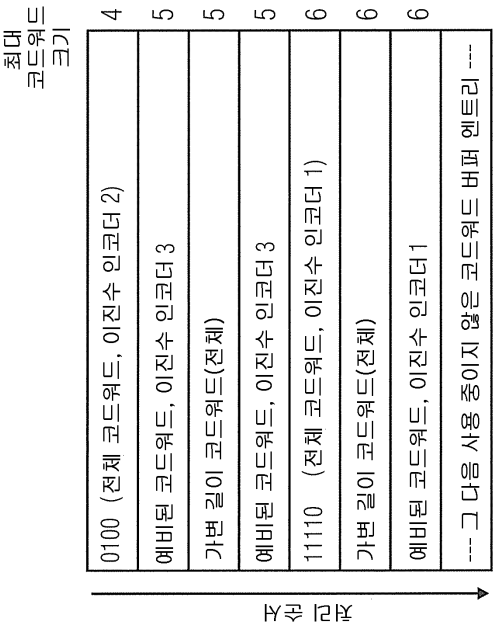
도면21


부분 비트스트림 0
부분 비트스트림 1
부분 비트스트림 2
부분 비트스트림 N
VLC 디코더
VLC 코드워드 요청
디코딩된 소스 심벌
VLC?
yes
no
소스 심벌 요청
소스 심벌 요청
이진화기
이진수 요청
파라미터 할당기
이진수 버퍼 선택기
관련된 파라미터 세트를 갖는 이진수 요청
디코딩된 이진수
디코딩된 이진수
이진수 요청
이진수 버퍼 1
이진수 버퍼 2
이진수 버퍼 N
이진수 디코더 1
이진수 디코더 2
이진수 디코더 N
코드워드
디코딩된 이진수 시퀀스
13a
13
14
15
16
17
18
19
19a
20
21
22
22a
23
23a
24
24a
25
26
27
27a

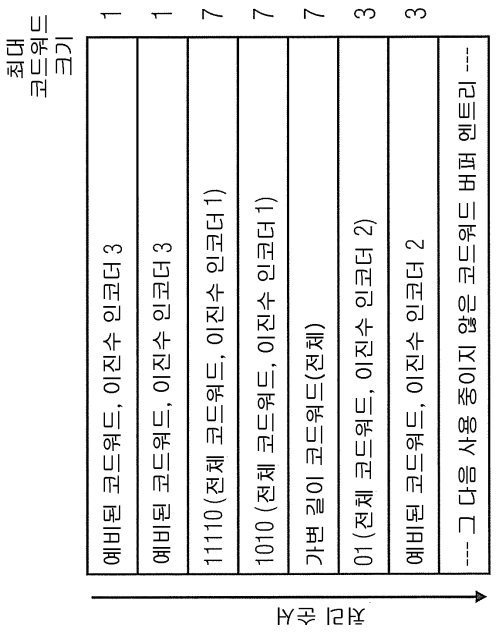
도면22


소스 심벌
1a
VLC?
yes
no
소스 심벌
1
이진화기
2
이진수
3
파라미터 할당기
4
관련된 파라미터 세트를 갖는 이진수
5
이진수 버퍼 선택기
6
이진수
7
이진수 버퍼1
8
이진수 버퍼2
8
이진수 버퍼N
8
9
이진수 인코더1
10
이진수 인코더2
10
이진수 인코더N
10
이진수 시퀀스
30
28
코드워드 버퍼(들) 요청
코드워드(들)
29
코드워드 버퍼
31
코드워드(들)
32
코드워드 기록기
33
비트
34
비트스트림
1b
소스 심벌
VLC 인코더
10a
VL 코드워드
30a

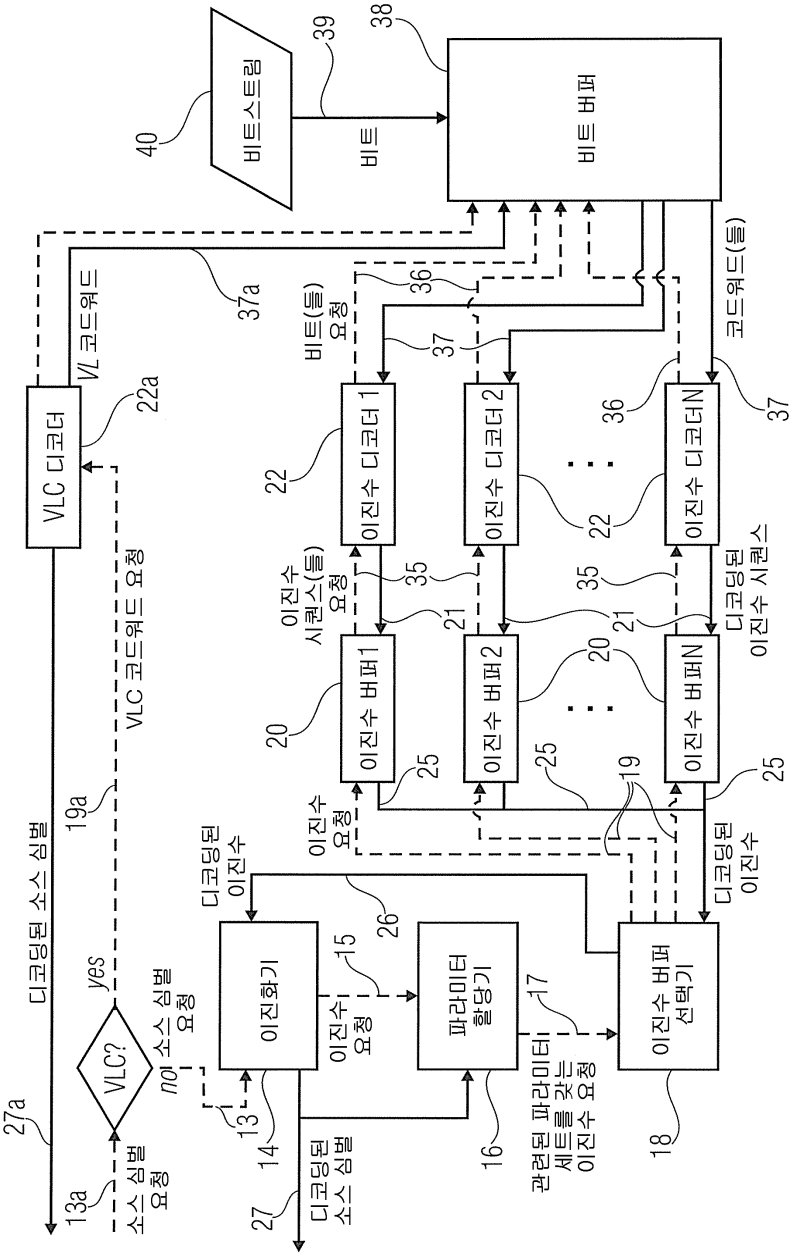
도면23a


최대
코드워드
크기
4
5
5
5
6
6
6
0100 (전체 코드워드, 이진수 인코더 2)
예비된 코드워드, 이진수 인코더 3
가변 길이 코드워드(전체)
예비된 코드워드, 이진수 인코더 3
11110 (전체 코드워드, 이진수 인코더 1)
가변 길이 코드워드(전체)
예비된 코드워드, 이진수 인코더 1
--- 그 다음 사용 중이지 않은 코드워드 버퍼 엔트리 ---
처리 순서

도면23b


최대
코드워드
크기
1
1
7
7
7
3
3
예비된 코드워드, 이진수 인코더 3
예비된 코드워드, 이진수 인코더 3
11110 (전체 코드워드, 이진수 인코더 1)
1010 (전체 코드워드, 이진수 인코더 1)
가변 길이 코드워드(전체)
01 (전체 코드워드, 이진수 인코더 2)
예비된 코드워드, 이진수 인코더 2
--- 그 다음 사용 중이지 않은 코드워드 버퍼 엔트리 ---
처리 순서

도면24


비트스트림
비트
비트 버퍼
VLC 디코더
VL 코드워드
VLC 코드워드 요청
비트(들) 요청
코드워드(들)
이진수 디코더 1
이진수 디코더 2
이진수 디코더N
이진수 시퀀스(들) 요청
디코딩된 이진수 시퀀스
이진수 버퍼1
이진수 버퍼2
이진수 버퍼N
이진수 요청
디코딩된 이진수
디코딩된 소스 심벌
VLC?
yes
no
소스 심벌 요청
이진화기
이진수 요청
파라미터 할당기
관련된 파라미터 세트를 갖는 이진수 요청
이진수 버퍼 선택기
소스 심벌 요청
디코딩된 소스 심벌