



**(19) 대한민국특허청(KR)**  
**(12) 등록특허공보(B1)**

(45) 공고일자 2017년05월17일  
 (11) 등록번호 10-1737294  
 (24) 등록일자 2017년05월11일

(51) 국제특허분류(Int. Cl.)  
*H03M 7/30* (2006.01) *G06F 7/24* (2006.01)  
*G06K 9/38* (2006.01) *G06T 1/20* (2006.01)  
*G11B 20/14* (2006.01) *H03M 7/00* (2006.01)  
*H03M 7/14* (2006.01) *H03M 7/40* (2006.01)

(52) CPC특허분류  
*H03M 7/30* (2013.01)  
*G06F 7/24* (2013.01)

(21) 출원번호 10-2016-7022772  
 (22) 출원일자(국제) 2015년02월20일  
 심사청구일자 2016년11월17일  
 (85) 번역문제출일자 2016년08월19일  
 (65) 공개번호 10-2016-0123306  
 (43) 공개일자 2016년10월25일  
 (86) 국제출원번호 PCT/EP2015/025008  
 (87) 국제공개번호 WO 2015/124324  
 국제공개일자 2015년08월27일

(30) 우선권주장  
 1403039.9 2014년02월20일 영국(GB)

(56) 선행기술조사문헌  
 KR1020120109753 A  
 US5838266 A

(73) 특허권자  
 구루로직 마이크로시스템스 오이  
 핀란드 투르쿠 20100 린난카투 34

(72) 발명자  
 칼레보 오시  
 핀란드 에프아이엔-37800 아카 케툰한타 1  
 카르크카이넨 투오마스  
 핀란드 에프아이엔-20320 투르쿠 비17 라우탈란카  
 투 2  
 후흐타니에미 아서  
 핀란드 에프아이엔-20100 투르쿠 린난카투 34

(74) 대리인  
 김태홍, 김진희

전체 청구항 수 : 총 25 항

심사관 : 조춘근

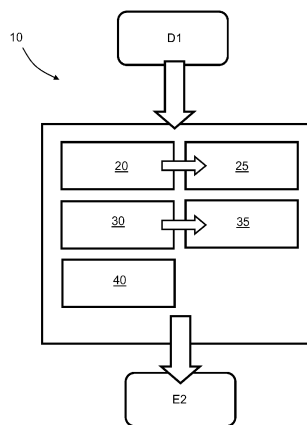
(54) 발명의 명칭 **심볼 압축을 수반하는 데이터의 소스 코딩 및 디코딩을 위한 방법들 및 디바이스들**

**(57) 요약**

인코더(50)에서 입력 데이터(D1)를 인코딩하여 대응하는 인코딩된 데이터(E2)를 생성하는 방법이 제공된다. 그 방법은, (a) 입력 데이터(D1)를 하나 이상의 데이터 청크들로 분리 및/또는 변환하고, 입력 데이터(D1)에 존재하는 심볼들을 분석하고 하나 이상의 데이터 청크들에서의 심볼들의 발생의 함수로서 심볼들을 압축하는 단계; (b)

(뒷면에 계속)

**대표도 - 도2**



하나 이상의 데이터 청크들에 존재하는 심볼들에 대한 하나 이상의 코드 테이블들, 및/또는 하나 이상의 빈도 테이블들, 및/또는 하나 이상의 코드 워드 길이 테이블들을 생성하는 단계; (c) 각각의 데이터 청크에서의 심볼들 및/또는 압축된 심볼들을 하나 이상의 코드 테이블들에서의 엔트리들, 및/또는 하나 이상의 빈도 테이블들, 및/또는 하나 이상의 코드 워드 길이 테이블들에 관련시키는 하나 이상의 인덱스 세트들을 컴퓨팅하는 단계; 및 (d) 인코딩된 데이터(E2)를 생성하기 위해, 하나 이상의 빈도 테이블들, 및/또는 하나 이상의 코드 테이블들, 및/또는 이러한 하나 이상의 테이블들을 나타내는 정보와 함께, 하나 이상의 인덱스 세트들을 어셈블링하는 단계를 포함한다. 대응하는 디코더(60)와 함께 그 방법을 이용하는 인코더(50)가 또한 제공되어 있는데, 인코더(50)와 디코더(60)는 조합하여 코덱(100)을 형성한다.

(52) CPC특허분류

- G06K 9/38* (2013.01)
  - G06T 1/20* (2013.01)
  - G11B 20/1419* (2013.01)
  - H03M 7/00* (2013.01)
  - H03M 7/14* (2013.01)
  - H03M 7/4006* (2013.01)
-

**명세서**

**청구범위**

**청구항 1**

대응하는 인코딩된 데이터(E2)를 생성하기 위하여 인코더(50)에서 입력 데이터(D1)를 인코딩하는 방법에 있어서,

(a) 상기 입력 데이터(D1)에 존재하는 심볼들을 분석하고, 상기 입력 데이터(D1)를 하나 이상의 데이터 청크(chunk)로 분리하는 동작과 상기 입력 데이터(D1)를 하나 이상의 데이터 청크로 변환하는 동작 중 적어도 하나의 동작을 수행하는 단계;

(b) 상기 하나 이상의 데이터 청크에 존재하는 상기 심볼들에 대한 하나 이상의 코드 테이블, 하나 이상의 빈도 테이블, 하나 이상의 코드 워드 길이 테이블, 하나 이상의 확률 테이블 중 적어도 하나를, 상기 심볼들의 발생의 함수로서 생성하는 단계;

(c) 각 데이터 청크 내의 상기 심볼들을, 상기 하나 이상의 코드 테이블, 상기 하나 이상의 빈도 테이블, 상기 하나 이상의 코드 워드 길이 테이블, 상기 하나 이상의 확률 테이블 중 적어도 하나에서의 엔트리들에 관련시키는 하나 이상의 인덱스 세트를 컴퓨팅(computing)하는 단계;

(d) 각 데이터 청크 내의 상기 심볼들을 관련시키는 상기 하나 이상의 인덱스 세트의 정보를, 상기 하나 이상의 빈도 테이블, 상기 하나 이상의 코드 테이블, 상기 하나 이상의 코드 워드 길이 테이블, 상기 하나 이상의 확률 테이블, 상기 하나 이상의 테이블을 나타내는 정보 중 적어도 하나와 함께 상기 인코딩된 데이터(E2)에 어셈블링(assembling)하는 단계; 및

(e) 상기 하나 이상의 테이블을 이용하여 상기 하나 이상의 데이터 청크의 상기 심볼들을 상기 인코딩된 데이터(E2)에 압축하는 단계

를 포함하는 것을 특징으로 하는, 입력 데이터(D1)의 인코딩 방법.

**청구항 2**

제1항에 있어서,

상기 하나 이상의 테이블 중 적어도 하나의 테이블이 미리 정의되는 것을 특징으로 하는, 입력 데이터(D1)의 인코딩 방법.

**청구항 3**

제1항 또는 제2항에 있어서,

상기 방법은, 상기 인코딩된 데이터(E2) - 상기 인코딩된 데이터(E2)는, 상기 압축된 심볼들 뿐 아니라, 상기 하나 이상의 빈도 테이블, 상기 하나 이상의 코드 테이블, 상기 하나 이상의 코드 워드 길이 테이블, 상기 하나 이상의 확률 테이블, 상기 하나 이상의 테이블을 나타내는 정보 중 적어도 하나와 함께, 각 데이터 청크 내의 상기 심볼들을 관련시키는 상기 하나 이상의 인덱스 세트의 정보를 포함함 - 를 디코더(60)로 전달하는 단계를 포함하는 것을 특징으로 하는, 입력 데이터(D1)의 인코딩 방법.

**청구항 4**

삭제

**청구항 5**

삭제

**청구항 6**

삭제

청구항 7

삭제

청구항 8

삭제

청구항 9

삭제

청구항 10

삭제

청구항 11

삭제

청구항 12

삭제

청구항 13

삭제

청구항 14

삭제

청구항 15

삭제

청구항 16

삭제

청구항 17

삭제

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

청구항 21

대응하는 인코딩된 데이터(E2)를 생성하기 위하여 입력 데이터(D1)를 인코딩하는 인코더(50)에 있어서,

(a) 상기 입력 데이터(D1)에 존재하는 심볼들을 분석하고, 상기 입력 데이터(D1)를 하나 이상의 데이터 청크로 분리하는 동작과 상기 입력 데이터(D1)를 하나 이상의 데이터 청크로 변환하는 동작 중 적어도 하나의 동작을 수행하는 분석기;

(b) 상기 하나 이상의 데이터 청크에 존재하는 상기 심볼들에 대한 하나 이상의 코드 테이블, 하나 이상의 빈도 테이블, 하나 이상의 코드 워드 길이 테이블, 하나 이상의 확률 테이블 중 적어도 하나를, 상기 심볼들의 발생의 함수로서 생성하는 생성기;

(c) 각 데이터 청크 내의 상기 심볼들을, 상기 하나 이상의 코드 테이블, 상기 하나 이상의 빈도 테이블, 상기 하나 이상의 코드 워드 길이 테이블, 상기 하나 이상의 확률 테이블 중 적어도 하나에서의 엔트리들에 관련시키는 하나 이상의 인덱스 세트를 컴퓨팅하는 컴퓨팅 엔진;

(d) 각 데이터 청크 내의 상기 심볼들을 관련시키는 상기 하나 이상의 인덱스 세트의 정보를, 상기 하나 이상의 빈도 테이블, 상기 하나 이상의 코드 테이블, 상기 하나 이상의 코드 워드 길이 테이블, 상기 하나 이상의 확률 테이블, 상기 하나 이상의 테이블을 나타내는 정보 중 적어도 하나와 함께 상기 인코딩된 데이터(E2)에 어셈블링하는 데이터 어셈블러; 및

(e) 상기 하나 이상의 테이블을 이용하여 상기 하나 이상의 데이터 청크의 상기 심볼들을 상기 인코딩된 데이터(E2)에 압축하는 압축기

를 포함하는 것을 특징으로 하는, 입력 데이터(D1)를 인코딩하는 인코더(50).

**청구항 22**

제21항에 있어서,

상기 인코더(50)는, 상기 인코딩된 데이터(E2) - 상기 인코딩된 데이터(E2)는, 상기 압축된 심볼들 뿐 아니라, 상기 하나 이상의 빈도 테이블, 상기 하나 이상의 코드 테이블, 상기 하나 이상의 코드 워드 길이 테이블, 상기 하나 이상의 확률 테이블, 상기 하나 이상의 테이블을 나타내는 정보 중 적어도 하나와 함께, 각 데이터 청크 내의 상기 심볼들을 관련시키는 상기 하나 이상의 인덱스 세트의 정보를 포함함 - 를 디코더(60)로 전달하도록 동작 가능한 것을 특징으로 하는, 입력 데이터(D1)를 인코딩하는 인코더(50).

**청구항 23**

제22항에 있어서,

상기 인코더(50)는, 상기 하나 이상의 테이블 중 적어도 하나의 테이블이 후속하는 재사용을 위해 저장 가능한 방식으로, 상기 하나 이상의 테이블 중 상기 적어도 하나의 테이블을 디코더(60)로 전달하도록 동작 가능한 것을 특징으로 하는, 입력 데이터(D1)를 인코딩하는 인코더(50).

**청구항 24**

삭제

**청구항 25**

삭제

**청구항 26**

제21항 내지 제23항 중 어느 한 항에 있어서,

상기 인코더(50)는, 상기 입력 데이터(D1)를 복수의 데이터 청크들로 분리하고, 상기 복수의 데이터 청크들을 병행하는 방식으로(in a concurrent manner) 프로세싱하는 프로세서들의 병렬 아키텍처를 채용하도록 동작 가능한 것을 특징으로 하는, 입력 데이터(D1)를 인코딩하는 인코더(50).

**청구항 27**

삭제

**청구항 28**

삭제

**청구항 29**

제21항에 있어서,

상기 인코더(50)는, 상기 하나 이상의 데이터 청크 중 데이터 청크의 상기 심볼들을 상기 인코딩된 데이터(E2)에 압축할 때, 상기 데이터 청크에 대한 달성가능한 데이터 압축률(achievable data compression ratio)에 기초하여, 상기 하나 이상의 데이터 청크 중 상기 데이터 청크 내의 상기 심볼들을 상기 인코딩된 데이터(E2)에 압축하는 것과 상기 하나 이상의 데이터 청크 중 상기 데이터 청크를 압축되지 않은 채로 상기 인코딩된 데이터(E2)에 어셈블링하는 것 사이에서 동적으로 전환(switch)하도록 동작 가능한 것을 특징으로 하는, 입력 데이터(D1)를 인코딩하는 인코더(50).

**청구항 30**

삭제

**청구항 31**

삭제

**청구항 32**

컴퓨터 판독가능 명령어들을 저장하고 있는 비일시적(non-transitory) 컴퓨터 판독가능 저장 매체에 있어서, 상기 컴퓨터 판독가능 명령어들은 제1항 또는 제2항에 기재된 방법을 실행하기 위하여 프로세싱 하드웨어를 포함하는 컴퓨터화된 디바이스에 의해 실행 가능한 것인, 비일시적 컴퓨터 판독가능 저장 매체.

**청구항 33**

대응하는 디코딩된 데이터(D3)를 생성하기 위하여 제21항 내지 제23항 중 어느 한 항에 기재된 인코더(50)에 의해 생성된 인코딩된 데이터(E2)를 디코딩하는 디코더(60)에서의 방법에 있어서,

(i) 상기 인코딩된 데이터(E2)를 수신하고, 수신된 상기 인코딩된 데이터(E2)로부터, 압축된 상기 하나 이상의 데이터 청크의 심볼들 뿐 아니라, 하나 이상의 빈도 테이블, 하나 이상의 코드 테이블, 하나 이상의 코드 워드 길이 테이블, 하나 이상의 확률 테이블, 상기 하나 이상의 테이블을 나타내는 정보 중 적어도 하나와 함께, 하나 이상의 인덱스 세트의 정보를 추출하는 단계;

(ii) 상기 하나 이상의 인덱스 세트의 정보로부터, 압축된 상기 하나 이상의 데이터 청크의 심볼들에 관련된, 상기 하나 이상의 코드 테이블, 상기 하나 이상의 빈도 테이블, 상기 하나 이상의 코드 워드 길이 테이블, 상기 하나 이상의 확률 테이블 중 적어도 하나를 컴퓨팅하는 단계;

(iii) 상기 하나 이상의 코드 테이블, 상기 하나 이상의 빈도 테이블, 상기 하나 이상의 코드 워드 길이 테이블, 상기 하나 이상의 확률 테이블 중 적어도 하나로부터의 정보를 이용하여, 상기 압축된 심볼들로부터 상기 하나 이상의 데이터 청크를 재생성하는 단계; 및

(iv) 상기 디코딩된 데이터(D3)를 생성하기 위하여, 상기 하나 이상의 데이터 청크를 결합하는 동작과 상기 하나 이상의 데이터 청크를 변환하는 동작 중 적어도 하나의 동작을 수행하는 단계

를 포함하는 것을 특징으로 하는, 인코딩된 데이터(E2)를 디코딩하는 디코더(60)에서의 방법.

**청구항 34**

제33항에 있어서,

상기 하나 이상의 테이블 중 적어도 하나의 테이블이 미리 정의되는 것을 특징으로 하는, 인코딩된 데이터(E2)를 디코딩하는 디코더(60)에서의 방법.

**청구항 35**

제33항에 있어서,

상기 방법은, 대응하는 트랜스코딩된 데이터(D4)를 생성하기 위하여 상기 디코딩된 데이터(D3)를 트랜스코딩하는 단계와 상기 인코딩된 데이터(E2)로부터 상기 대응하는 트랜스코딩된 데이터(D4)를 생성하는 단계 중 적어도 하나의 단계를 포함하는 것을 특징으로 하는, 인코딩된 데이터(E2)를 디코딩하는 디코더(60)에서의 방법.

- 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

**청구항 52**

제33항에 있어서,

상기 방법은, 복수의 데이터 소스들로부터 상기 인코딩된 데이터(E2)를 수신하는 단계와, 상기 인코딩된 데이터(E2)를 재생성하기 위하여 상기 소스들로부터 제공된 데이터를 결합하는 단계를 포함하는 것을 특징으로 하는, 인코딩된 데이터(E2)를 디코딩하는 디코더(60)에서의 방법.

**청구항 53**

대응하는 디코딩된 데이터(D3)를 생성하기 위하여 제21항에 기재된 인코더(50)에 의해 생성된 인코딩된 데이터(E2)를 디코딩하는 디코더(60)에 있어서,

(i) 상기 인코딩된 데이터(E2)를 수신하고, 수신된 상기 인코딩된 데이터(E2)로부터, 압축된 상기 하나 이상의 데이터 청크의 심볼들 뿐 아니라, 하나 이상의 빈도 테이블, 하나 이상의 코드 테이블, 하나 이상의 코드 워드 길이 테이블, 하나 이상의 확률 테이블, 상기 하나 이상의 테이블을 나타내는 정보 중 적어도 하나와 함께, 하나 이상의 인덱스 세트의 정보를 추출하고;

(ii) 상기 하나 이상의 인덱스 세트의 정보로부터, 압축된 상기 하나 이상의 데이터 청크의 심볼들에 관련된, 상기 하나 이상의 코드 테이블, 상기 하나 이상의 빈도 테이블, 상기 하나 이상의 코드 워드 길이 테이블, 상기 하나 이상의 확률 테이블 중 적어도 하나를 컴퓨팅하고;

(iii) 상기 하나 이상의 코드 테이블, 상기 하나 이상의 빈도 테이블, 상기 하나 이상의 코드 워드 길이 테이블, 상기 하나 이상의 확률 테이블 중 적어도 하나로부터의 정보를 이용하여, 상기 압축된 심볼들로부터 상기 하나 이상의 데이터 청크를 재생성하며;

(iv) 상기 디코딩된 데이터(D3)를 생성하기 위하여, 상기 하나 이상의 데이터 청크를 결합하는 동작과 상기 하나 이상의 데이터 청크를 변환하는 동작 중 적어도 하나의 동작을 수행하도록

동작 가능한 것을 특징으로 하는, 인코딩된 데이터(E2)를 디코딩하는 디코더(60).

**청구항 54**

제53항에 있어서,

상기 디코더(60)는, 대응하는 트랜스코딩된 데이터(D4)를 생성하기 위하여 상기 디코딩된 데이터(D3)를 트랜스코딩하는 동작과 상기 인코딩된 데이터(E2)로부터 상기 대응하는 트랜스코딩된 데이터(D4)를 생성하는 동작 중 적어도 하나의 동작을 수행하는 트랜스코더(70)를 더 포함하는 것을 특징으로 하는, 인코딩된 데이터(E2)를 디코딩하는 디코더(60).

**청구항 55**

삭제

**청구항 56**

제53항에 있어서,

상기 디코더(60)는, 상기 하나 이상의 테이블 중 적어도 하나의 테이블이 후속하는 재사용을 위해 저장 가능한 방식으로, 상기 하나 이상의 테이블 중 상기 적어도 하나의 테이블을 수신하도록 동작 가능한 것을 특징으로 하는, 인코딩된 데이터(E2)를 디코딩하는 디코더(60).

**청구항 57**

삭제

**청구항 58**

삭제

**청구항 59**



제53항, 제54항, 및 제56항 중 어느 한 항에 있어서,

상기 디코더(60)는, 상기 디코딩된 데이터(D3)를 생성하기 위하여, 상기 하나 이상의 데이터 청크 중 복수의 데이터 청크들을 병행하는 방식으로 프로세싱하는 프로세서들의 병렬 아키텍처를 채용함으로써 상기 복수의 데이터 청크들을 결합하도록 동작 가능한 것을 특징으로 하는, 인코딩된 데이터(E2)를 디코딩하는 디코더(60).

**청구항 60**

삭제

**청구항 61**

삭제

**청구항 62**

제53항에 있어서,

상기 디코더(60)는, 상기 하나 이상의 데이터 청크 중 데이터 청크를 재생성할 때, 상기 데이터 청크에 대한 달성가능한 데이터 압축해제율(achievable data decompression ratio)에 기초하여, 상기 하나 이상의 데이터 청크 중 상기 데이터 청크를 압축된 상태로 재생성하는 것과 상기 데이터 청크를 압축되지 않은 상태로 재생성하는 것 사이에서 동적으로 전환하도록 동작 가능한 것을 특징으로 하는, 인코딩된 데이터(E2)를 디코딩하는 디코더(60).

**청구항 63**

삭제

**청구항 64**

삭제

**청구항 65**

제53항, 제54항, 및 제56항 중 어느 한 항에 있어서,

상기 인코더(50)의 데이터 어셈블러는 상기 인코딩된 데이터(E2)에 어셈블링된 상기 하나 이상의 코드 테이블에 하나 이상의 데이터 압축 알고리즘을 적용하고, 상기 디코더(60)는 상기 인코딩된 데이터(E2)에 포함된 상기 하나 이상의 코드 테이블을 압축해제하도록 동작 가능한 것을 특징으로 하는, 인코딩된 데이터(E2)를 디코딩하는 디코더(60).

**청구항 66**

삭제

**청구항 67**

삭제

**청구항 68**

제53항, 제54항, 및 제56항 중 어느 한 항에 있어서,

상기 디코더(60)는, 후속하여 전송된 데이터를 디코딩하기 위해 상기 하나 이상의 코드 테이블이 상기 디코더(60)에서 채용되는 것을 가능하게 하는 방식으로, 상기 하나 이상의 코드 테이블을 수신하도록 동작 가능한 것을 특징으로 하는, 인코딩된 데이터(E2)를 디코딩하는 디코더(60).

**청구항 69**

삭제

**청구항 70**

삭제

**청구항 71**

제53항에 있어서,

상기 디코더(60)는, 복수의 데이터 소스들로부터 상기 인코딩된 데이터(E2)를 수신하고, 상기 인코딩된 데이터(E2)를 재생성하기 위하여 상기 소스들로부터 제공된 데이터를 결합하도록 동작 가능한 것을 특징으로 하는, 인코딩된 데이터(E2)를 디코딩하는 디코더(60).

**청구항 72**

대응하는 인코딩된 데이터(E2)를 생성하기 위하여 입력 데이터(D1)를 인코딩하는 적어도 하나의 제21항에 기재된 인코더(50)와, 디코딩된 출력 데이터(D3)를 생성하기 위하여 상기 인코딩된 데이터(E2)를 디코딩하는 적어도 하나의 제53항에 기재된 디코더(60)를 포함하는 것을 특징으로 하는, 코덱(100).

**청구항 73**

제72항에 있어서,

상기 적어도 하나의 인코더(50)와 상기 적어도 하나의 디코더(60)는 서로 공간적으로 떨어져 있고 데이터 통신 네트워크를 통해 서로 상호 결합되는 것을 특징으로 하는, 코덱(100).

**청구항 74**

제73항에 있어서,

상기 데이터 통신 네트워크는 피어 투 피어(peer-to-peer) 통신 네트워크의 방식으로 구성되는 것을 특징으로 하는, 코덱(100).

**청구항 75**

제72항에 있어서,

상기 인코더(50)에서 실행되는 프로세싱 기능들은, 상기 디코더(60)에서와 대응하는 역 기능들로서 구현되고 상기 디코더(60)에서와 역 순으로 실행되는 것을 특징으로 하는, 코덱(100).

**발명의 설명**

**기술 분야**

[0001] 본 개시물은 입력 데이터를 인코딩하여 대응하는 인코딩된 데이터를 생성하는 방법들에 관한 것이다. 더구나, 본 개시물은 또한, 전술한 인코딩된 데이터를 디코딩하여 대응하는 디코딩된 출력 데이터를 생성하는 방법들에 관한 것이다. 더욱이, 본 개시물은 또한, 전술한 방법들을 구현하도록 동작 가능한 인코더들 및 디코더들에 관한 것이다. 덧붙여, 본 개시물은, 컴퓨터 판독가능 명령어들 - 컴퓨터 판독가능 명령어들은 전술한 방법들을 실행하기 위해 프로세싱 하드웨어를 포함하는 컴퓨터화 디바이스에 의해 실행 가능함 - 을 저장하고 있는 비일시적 컴퓨터 판독가능 저장 매체를 포함하는 컴퓨터 프로그램 제품들에 관한 것이다.

**배경 기술**

[0002] 개관하여, 도 1에 예시된 바와 같이, 입력 데이터(D1)를 인코딩하여 대응하는 인코딩된 출력 데이터(E2)를 생성하는 알려진 인코딩 방법들이, 하나 이상의 변환들(T)을 입력 데이터(D1)에 적용하여 대응하는 변환된 인코딩된 출력 데이터(E2)를 생성하는 것을 수반하는데, 변환된 인코딩된 출력 데이터(E2)는 채용된 하나 이상의 변환들(T)을 규정하는 하나 이상의 코딩 테이블들을 나타내는 코딩 테이블 데이터(C) 정보와 연관된다. 인코딩된 변환된 데이터(E2)와 코딩 테이블 데이터(C) 정보, 즉 총칭하여 인코딩된 출력 데이터(E2)는 인코딩된 출력 데이터(E2)를 디코딩하여 대응하는 디코딩된 데이터(D3)를 생성하기 위해 하나 이상의 역 변환들(T<sup>-1</sup>)을 적용하도록 동작 가능한 하나 이상의 디코더들에게 데이터 반송파를 통해 그리고/또는 데이터 통신 네트워크를 통해 종종 통신된다. 예를 들어 인코딩된 출력 데이터(E2)를 통신할 때 통신 네트워크 용량 부하를 줄이기 위해, 인코딩된 출력 데이터(E2)는 입력 데이터(D1)에 비하여 압축되는 것이 종종 바람직하다. 더구나, 인코딩된 출력 데이

터(E2)는 실질적으로 무손실 방식으로 압축되어서, 디코딩된 데이터(D3)는 입력 데이터(D1)에 포함된 정보의 정확한 재현물(reproduction)이 되는 것이 또한 바람직하다. 입력 데이터(D1)에 비하여 인코딩된 출력 데이터(E2)에서 달성 가능한 데이터 압축은, 코딩 테이블 데이터(C) 정보가 인코딩된 변환된 데이터(E2)에 비하여 사이즈가 상당할 때, 즉 코딩 테이블 데이터(C) 정보가 변환된 인코딩된 데이터(E2)에서의 상당한 데이터 오버헤드에 해당할 때 잠재적으로 비효율적이다.

[0003] 입력 데이터(D1)를 인코딩하여 인코딩된 출력 데이터(E2)를 생성하는 여러 알려진 방법들이 있다. 예를 들어, 알려진 허프만(Huffman) 인코딩 또는 다른 VLC 인코딩 방법들은 다양한 유형들의 데이터를 압축하기 위해 종종 채용된다. 더구나, 산술 코딩(Arithmetic coding), 또는 레인지 코딩(Range coding)이, 입력 데이터를 압축하기 위해 점점 더 인기가 많아지고 있지만, 다음의 상황들에서는 꽤 비효율적이다:

[0004] (i) 입력 데이터(D1)에 대한 빈도 테이블이, 입력 데이터(D1)를 인코딩하여 대응하는 인코딩된 출력 데이터(E2)를 생성하도록 동작 가능한 인코더에 의해, 그리고 인코딩된 출력 데이터(E2)를 디코딩하도록 동작 가능한 디코더에 의해 미리 알려져 있지 않는 상황; 및

[0005] (ii) 예를 들어 입력 데이터(D1)가 작은 데이터 세그먼트들 또는 데이터 청크(chunk)들로 통신되고 각각의 데이터 세그먼트 또는 데이터 청크에는 대응하는 빈도 테이블이 동반되고 있는 상황에서, 입력 데이터의 양이 비교적 작은 상황.

[0006] 전술한 바와 같이, 이러한 비효율성은 상당한 데이터 공간을 소비하는 하나 이상의 빈도 테이블들의 전달로 인해, 예를 들어 디코더가 그 때 국부적으로 저장하고 있는 가능한 빈도 테이블들의 리스트로부터 상대적으로 적은 수의 식별 파라미터들을 사용하여 빈도 테이블을 선택할 수 없다면 발생한다. 더구나, 이러한 리스트로부터 적합한 코드 테이블들보다는 적합한 빈도 테이블들을 찾는 것이 또한 가능성이 더 적다. 종종, 인코딩될 입력 데이터(D1)는 또한 매우 국부적이며, 예를 들어 통신 네트워크에 대한 공간적으로 국부적인 데이터 표준들에 순응하도록 통신 네트워크를 통한 송신 동안 변환된다.

[0007] 심볼들로부터 도출된 인코딩된 데이터 콘텐츠를 통신하는 것에 관련하여 코드 테이블들 또는 빈도 테이블들을 전달하기 위해 이용 가능한 알려진 방법들이 있다. 대부분의 알려진 방법들은 심볼들의 허프만 트리 또는 빈도들의 직접 전달을 채용한다. 이러한 알려진 방법들은 그렇게 만족스럽지 못하는데, 그 방법들이 상당한 정보 인코더로부터 대응하는 디코더로 전달되는 것을 요구하기 때문이다. 더구나, 예를 들어 현대적으로 사용되지 않는 알려진 인텔 IPP-라이브러리에서 채용된 바와 같이, 코드 테이블 심볼들의 길이들을 전달하는 알려진 방법들이 또한 있으며; 코드 테이블을, 즉 "HuffLenCodeTablePack"에 의하여 압축하고, 그것을, 즉 "HuffLenCodeTableUnpack"에 의하여 다시 역으로 디코딩하는 방법이 채용되어 있지만; 이 방법은 만족스럽지 않고 때때로 인코딩 프로세스들 동안 데이터의 사이즈를 심지어 증가시키기도 한다. 더구나, 그 방법은 256 개 심볼들이 있을 것과 0부터 255까지의 모든 심볼들이 그것들의 코드 워드들을 위해 영이 아닌 길이를 가질 것을 또한 요구한다. 코드 테이블들을 전달하는 방법들은, 예를 들어 허프만 인코딩 기법들에 의해 생성되는 접두부 코드(prefix code)들에 대해 현재 이용 가능한 가장 효율적인 전달 메커니즘들 중에 여전히 분명히 있다. 허프만 트리가 인코더로부터 대응하는 디코더로 전달될 때, 인코더로부터의 생성된 코드 심볼들은 인코더에서와 디코더에서 항상 유사하다. 빈도 테이블만이 전달될 때, 디코더에서 적절한 방식으로 심볼들의 디코딩을 가능하게 하기 위해, 허프만 트리가 필요하다면, 빈도 테이블로부터 실제 허프만 트리 생성을 위해 인코더에서 그리고 디코더에서 사용되는 유사한 알고리즘들이 달성되어 있어야만 한다. 코드 테이블 심볼들의 길이들이 전달된다면, 심볼 길이들-대-빈도 테이블 변환으로부터의 유사한 방법들이 적절한 방식으로 심볼들의 디코딩을 가능하게 하기 위해 인코더에서 그리고 디코더에서 또한 필요하다. 인코더에서부터 디코더로 심볼들의 길이들을 통신하는 것은 산술 코딩 및 레인지 코딩에 대해 빈도들을 전달하는 실용적인 방법이 아닌데, 산술 코딩 및 레인지 코딩이 코드 심볼들의 길이들을 통신하는 것에 의해 단지 가능하게 된 것보다 더욱 정확한 빈도 테이블들을 지원하도록 설계되기 때문이다. 코드 심볼들의 길이들은 산술 코딩 및 레인지 코딩에서 또한 사용될 수 있다. 그러나, 이들 방법들은, 테이블들의 적응적 업데이트가 장래의 데이터에 대해 나중에 실행되지 않는다면, 예를 들어 허프만 코딩에 비하여 이점을 제공하지 않는다. 확률들을 나타내는 정보의 전달은 허프만 코딩과 대비하여 레인지 코딩 또는 산술 코딩과 통상 더욱 최적의 코딩 결과들을 제공한다. 심볼들의 확률들은 심볼들의 발생 빈도들을 심볼 발생 빈도들의 합계, 즉 심볼들의 수에 의해 나눔으로써 계산될 수 있다. 이러한 확률들의 전달이 스케일링된 확률 값들을 사용함으로써 유익하게 이루어진다. 스케일링된 확률 값들은 원래의 심볼 확률 값들과 유리하게는 2 값의 거듭제곱, 즉  $2^n - n$ 은 정수-인 정수 수를 곱한 다음, 그것을 가장 가까운 정수 값으로 반올림함으로써 계산될 수 있다. 정수 수들인 이들 스케일링된 확률들의 합계는 승수 값과 동일한 것으로

등화된다. 그렇지 않고 자신들 소유의 영이 아닌 스케일링된 확률 값에 할당되지 않는 심볼들에 대해 이스케이프 코드(escape code) 심볼이 또한 유익하게 생성된다. 이는 이스케이프 코드를 필요로 하는 그들 심볼들이 선택된 승수 값으로 제시될 수 있는 것보다 더 작은 확률을 가짐을 의미한다. 두 개의 상이한 메커니즘들로 이스케이프 코드를 채용하는 일 없이 스케일링된 확률들을 생성하는 것이 또한 가능하다. 승수 값은 증가될 수 있고 그러면 새로운 확률 값들이 계산될 수 있다. 영과 동일한 이용 가능한 심볼들에 대한 그들 스케일링된 확률 값들을 일과 동일한 것이 되게 업그레이드하는 것이 또한 가능하다. 이 확률 값 업그레이드는 확률 값들의 증가가 다른 심볼들의 확률 값들을 감소시킴으로써 보상되는 것을 요구한다. 이는 확률들의 합계를 승수 값과 정확히 동일하게 하기 위해서 행해진다. 이 절차는 확률 값들을 가능할 수 있는 것만큼 잘 최적화되게 하지는 않지만, 이스케이프 심볼들이 필요하지 않고, 몇몇 경우들에서 그것은 여전히 최적의 코딩 해법일 수 있다. 심볼들의 길이들 또는 확률 값들은 가변 장 코딩(variable length coding) 심볼들을 채용하는 방법들, 예를 들어 허프만 코딩, 레인지 코딩, 산술 코딩 및 임의의 다른 가변 장 코딩 방법들을 위해 사용될 수 있는 빈도 테이블의 대강 추정치(rough estimate)를 규정한다. 스케일링된 확률 테이블은 심볼 빈도들이 필요할 때 그 심볼 빈도들의 대강 추정치로서 직접적으로 사용될 수 있고, 심볼들의 길이들은 이용 전에 심볼 빈도들의 대강 추정치로 먼저 변환되는 것이 필요하다는 것이 이해될 것이다. 심볼들의 길이로부터 빈도 테이블로의 변환은 나중에 데이터 인코딩 및 테이블 전달 동안 보여질 것이다.

[0008] 데이터를 인코딩하는 많은 알려진 실용적인 방법들은 최적화된 코드 테이블들을 전혀 이용하지 않으며, 즉 그 방법들은 데이터를 인코딩하여 대응하는 인코딩된 데이터를 생성하기 위한 고정된 코드 테이블들과, 그 뒤에 인코딩된 데이터를 디코딩하기 위한 고정된 코드 테이블들을 이용한다. 때때로 테이블들은 전달된 심볼들에 기초하여 적응적 방법들로 업데이트된다. 특정한 알려진 방법들에서는, 인코더에서 데이터를 인코딩하고 대응하여 디코더에서 인코딩된 데이터를 디코딩하기 위해 두 개의 상이한 코드 테이블들, 대안적으로 빈도 테이블들이 때때로 이용되고 있는데, 선택된 코드 테이블, 확률 테이블, 또는 빈도 테이블을 규정하는 인덱스가 인코더로부터 디코더로 정보로서 전달된다. 특정한 방법들에서는, 휘도(luminance) 및 컬러 채널들에 대해, 인트라 블록들 및 인트라 블록들에 대해, 또는 데이터의 상이한 종류들에 대해 별개의 테이블들이 채용되고 있지만; 그 별개의 테이블들은 비효율적인 방식으로 통신되며; 예를 들어, 여기서는 다음의 인터넷 웹-사이트(Wikipedia)가 참조된다: [http://en.wikipedia.org/wiki/Huffman\\_coding](http://en.wikipedia.org/wiki/Huffman_coding). 압축해제 동안, 허프만 기반 방법들을 사용하여, 허프만 트리가 재구축되어야만 한다. 캐릭터 빈도들이 비교적 예측 가능한 가장 간단한 경우들에서, 그 트리는 재구축될 가능성이 있고, 심지어 각각의 압축 사이클 상에서 통계적으로 조정되고, 따라서 적어도 어느 정도의 압축 효율을 희생하여 가능하면 언제든지 재사용되고; 대안으로는, 허프만 트리 정보가 *사전에*, 즉, 미리 전송되어야만 한다.

[0009] 압축된 데이터의 출력 스트림으로 코딩되는 심볼들에 관련한 빈도 카운트들을 덧붙이는 간단한 접근법이, 실제로 적어도 수 킬로바이트(kB)만큼 압축된 데이터에서의 데이터 볼륨을 증가시키는 주요 단점을 갖고, 그래서 이러한 간단한 접근법은 그다지 실제로 사용되지 않는다. 그 데이터가 정규화 인코딩(canonical encoding)을 사용하여 압축된다면, 압축 모델은 정보의  $B^B$  비트들 - B는 심볼 당 비트 수 - 만으로, 예를 들어, 2 KB를 요구하는 8 비트들로 정확히 복원될 수 있다.

[0010] 다른 방법이 단순히 허프만 트리를, 비트 단위로, 압축된 출력 스트림에 덧붙이는 것이다. 예를 들어, 0의 값이 부모 노드와 1의 리프 노드를 나타낸다고 가정하면, 후자가 마주하게 될 때마다, 트리 구축 루틴이 그 특정 리프의 캐릭터 값을 결정하기 위해 다음의 8 비트들을 단순히 읽는다. 이러한 프로세스는 마지막 리프 노드에 도달되기까지 재귀적으로 계속되며; 그 포인트에서, 허프만 트리는 그에 따라, 예를 들어 디코더에서 충실하게 복원된다. 이러한 방법을 사용하는 것으로부터 발생하는 데이터 오버헤드가, 8-비트 알파벳을 가정하면, 대략 2부터 320까지의 범위이다.

[0011] 데이터를 인코딩하는 추가의 알려진 방법들과 인코딩된 데이터를 디코딩하는 대응하는 방법들을 설명하기 위하여, 허프만 디코딩이 다음에서 개략적으로 설명될 것이다. 임의의 다른 방법들, 예를 들어 레인지 디코딩 또는 산술 디코딩이, 허프만 디코딩 대신 또한 이용될 수 있다는 것이 이해될 것이다. 데이터 파일의 압축을 시작하기 전에, 압축기가, 인코더에서, 압축을 실행할 때 채용될 코드들을 결정해야 한다.

[0012] 허프만 디코딩이 채용될 때, 심볼들을 포함하는 주어진 데이터 파일의 압축을 시작하여 대응하는 인코딩된 출력 데이터를 생성하기 전에, 인코더가 주어진 데이터를 나타내는데 사용될 코드들을 결정해야 한다. 편리하게는, 그 코드들은 주어진 데이터 파일에서의 심볼들의 확률들, 즉 발생 빈도들에 기초한다. 그러나, 심볼들의 빈도들, 확률들, 또는 길이들이 인코딩된 출력 데이터에서, 예를 들어 사이드 정보로서, 즉 보충 정보로서 기록되어

야 하고, 그래서 임의의 허프만 디코더가 인코딩된 출력 데이터를 디코딩하여 대응하는 디코딩된 데이터를 생성할 수 있을 것이다. 편리하게는, 심볼들의 발생 빈도들 또는 길이들은 정수들, 또는 스케일(scales) 정수들로서 표현될 수 있는 확률들이며; 보충 정보에 포함된 그러한 정수들은 종종 인코딩된 출력 데이터에 단지 수백 바이트들만을 추가한다. 옵션으로, 가변 장 코드들 자체를 인코딩된 출력 데이터에 기록하는 것이 또한 가능하지만, 이는 특정한 환경들에서는 곤란한데, 그 코드들이 상호 상이한 사이즈들을 가질 수도 있기 때문이다. 대안으로, 허프만 트리를 인코딩된 출력 데이터에 기록하는 것이 실현 가능하지만, 이는 주어진 데이터에서 심볼들의 발생 빈도들을 단순히 통신하는 것보다는 더 많은 데이터가 통신되는 것을 요구한다.

- [0013] 동작 동안, 디코더에는 디코딩되기 위해 디코더에서 수신되는 인코딩된 압축된 파일의 처음에 무엇이 있는지에 관한 정보가 제공되어야만 한다. 인코딩된 압축된 파일로부터, 예를 들어 그 압축된 파일의 처음으로부터 추출된 데이터로부터, 디코더는 허프만 트리의 알파벳을 구축하도록 동작 가능하다. 허프만 트리가 디코더에서 구축된 후, 디코더는 허프만 트리를 디코딩 도구로서 사용하여 파일의 나머지를 디코딩할 수 있다. 디코더는 다음의 단계들을 포함하는 비교적 간단한 디코딩 알고리즘을 채용한다:
- [0014] (a) 허프만 트리의 루트에서 시작한 다음, 허프만 트리를 사용하여 디코딩될 인코딩된 출력 데이터의 첫 번째 비트를 읽으며;
- [0015] (b) 그 첫 번째 비트가 "1"이면, 허프만 트리의 상단 에지를 따라가며; 그 첫 번째 비트가 "0"이면, 허프만 트리의 하단 에지를 따라가며;
- [0016] (c) 인코딩된 출력 데이터의 두 번째 비트를 읽은 다음, 허프만 트리의 "리프"에 마침내 도달하기까지 허프만 트리 등의 "리프들"을 향해 단계 (b)와 유사한 방식으로 두 번째 비트를 채용하며, 그래서 원래의 비압축 심볼, 종종 연관된 ASCII 코드가 발견되며; 그 코드는 그 다음에 디코더로부터 출력되며; 그리고
- [0017] (d) 단계 (b)와 단계 (c)는 인코딩된 출력 데이터가 디코딩되기까지 반복된다.
- [0018] 알려진 현대적 허프만 인코딩은 인코딩된 스트링이 스트링을 생성하기 위해 채용된 코드 테이블에 비하여 사이즈가 더 클 때 채용하는데 유익하다. 더구나, 이러한 현대적 허프만 코딩은 코드 테이블이 인코더 및 대응하는 디코더 양쪽 모두에 대해 선형적으로 규정될 때 채용하는데 유익하다. 따라서 데이터를 인코딩 및 디코딩하는 것에 대한 알려진 접근법들, 예를 들어 전술한 허프만 인코딩 및 디코딩 방법들에 연관된 앞서 언급한 제한들을 해결하는 대안적 인코딩 방법들이 대한 요구가 있다.

**발명의 내용**

**해결하려는 과제**

- [0019] 본 발명은 데이터(D1)를 인코딩하여 대응하는 인코딩된 데이터(E2)를 생성하는 개선된 방법을 제공하는 것을 추구한다.
- [0020] 본 발명은 데이터를 인코딩하는 전술한 개선된 방법을 채용하도록 동작 가능한 개선된 인코더를 제공하는 것을 추구한다. 본 발명은 인코딩된 데이터(E2)를 디코딩하여 대응하는 디코딩된 데이터(D3)를 생성하는 개선된 방법을 제공하는 것을 추구한다.
- [0021] 본 발명은 전술한 인코딩된 데이터(E2)를 디코딩하여 대응하는 디코딩된 데이터(D3)를 생성하는 개선된 디코더를 제공하는 것을 추구한다.

**과제의 해결 수단**

- [0022] 제 1 양태에 따르면, 대응하는 인코딩된 데이터(E2)를 생성하기 위하여 인코더에서 입력 데이터(D1)를 인코딩하는 방법이 제공되는데, 그 방법은,
- [0023] (a) 입력 데이터(D1)에 존재하는 심볼들을 분석하고, 입력 데이터(D1)를 하나 이상의 데이터 청크들로 분리 및/또는 변환하는 단계;
- [0024] (b) 하나 이상의 데이터 청크들에 존재하는 심볼들에 대한 하나 이상의 코드 테이블들, 하나 이상의 빈도 테이블들, 하나 이상의 코드 워드 길이 테이블들, 하나 이상의 확률 테이블들 중 적어도 하나를, 심볼들의 발생의 함수로서 생성하는 단계;
- [0025] (c) 각각의 데이터 청크에서의 심볼들을, 하나 이상의 코드 테이블들, 하나 이상의 빈도 테이블들, 하나 이상의

코드 워드 길이 테이블들, 하나 이상의 확률 테이블들 중 적어도 하나에서의 엔트리들에 관련시키는 하나 이상의 인덱스 세트들을 컴퓨팅하는 단계;

[0026] (d) 하나 이상의 빈도 테이블들, 하나 이상의 코드 테이블들, 하나 이상의 코드 워드 길이 테이블들, 하나 이상의 확률 테이블들, 하나 이상의 테이블들을 나타내는 정보와 함께, 각각의 데이터 청크에서의 심볼들을 인코딩된 데이터(E2)에 관련시키는 하나 이상의 인덱스 세트들의 정보를 어셈블링하는 단계; 및

(e) 인코딩된 데이터(E2)에 하나 이상의 테이블들 내의 심볼들을 압축하는 단계를 포함하는 것을 특징으로 한다.

[0027] 본 발명은, 그 방법이 입력 데이터(D1)를 하나 이상의 데이터 청크들로 분리, 즉 분할하는 단계, 및/또는 입력 데이터(D1)에서의 심볼들을 압축하는 단계를 수반하여서, 입력 데이터(D1)는, 예를 들어, 인덱스들("indexes")과 그 인덱스들에 의해 참조되는 연관된 하나 이상의 테이블들을 사용하여 각각의 데이터 청크 또는 압축된 심볼에 가장 적합한 방식으로, 효율적으로 인코딩될 수 있다는 점이 장점이다.

옵션으로, 본 방법에서, 하나 이상의 테이블들 중 적어도 하나의 테이블이 미리 정의된다.

옵션으로, 본 방법은, 하나 이상의 빈도 테이블들, 하나 이상의 코드 테이블들, 하나 이상의 코드 워드 길이 테이블들, 하나 이상의 확률 테이블들, 하나 이상의 테이블들을 나타내는 정보 중 적어도 하나와 함께, 각각의 데이터 청크에서의 상기 심볼들을 관련시키는 하나 이상의 인덱스 세트들의 정보뿐만 아니라, 압축된 심볼들을 포함하는 인코딩된 데이터(E2)를 전달하는 단계를 포함한다.

옵션으로, 본 방법은, 대칭성 및 대응하는 대칭성 표시 정보에 기초하여 디코딩 동안 재생성되도록 하나 이상의 테이블들을 생성하는 단계를 포함한다.

[0028] 옵션으로, 그 방법은 하나 이상의 테이블들 중 적어도 하나의 테이블이 후속하는 재사용을 위해 저장 가능한 방식으로 하나 이상의 테이블들 중 적어도 하나의 테이블을 전달하는 단계를 포함한다.

옵션으로, 본 방법에서, 하나 이상의 인덱스 세트들의 정보는,

(i) 청크에서 그렇게 발생하며 그것들의 확률 값들이 데이터 청크에서 인덱스들의 총 카운트가 앞에 나오게 인코딩된 데이터(E2) 속에 삽입될 것들인 심볼들의 인덱스들; 또는

(ii) 모든 가능한 심볼 값들 중에서 어느 심볼들이 청크에서 발생하는지와, 이 심볼들에 대해서 확률 값들이 인코딩된 데이터(E2) 속에 삽입될 것을 나타내는 비트들; 또는

(iii) 그것 자체가 또한 상기 인덱스들의 카운트를 나타내는 상기 인코딩된 데이터(E2) 속에 삽입될 모든 심볼 값 확률들이 시작 부분에서 시작하는 것을 명시하는 정보를 포함한다.

[0029] 이 전술한 분리, 즉 분할하는 단계는, 옵션으로 입력 데이터(D1)의 세분을 포함한다.

[0030] 옵션으로, (a)에서, 분리하는 단계가 보통 행해지지만, 때때로 데이터를 새로운 데이터 청크들로 분리하는 일 없이 최적의 코드 테이블로 이용 가능한 데이터 청크를 압축하는 것이 또한 필요하다. 더구나, 때때로, 원래의 데이터는 분리되지 않지만, 대신, 새로운 데이터가, 예를 들어 가장 효율적으로 압축되는 것이 필요한 하나 이상의 데이터 청크들에 대해 하나 이상의 변환들에 의해 생성된다. 본 개시물에 의하여 구현되는 인코더들은 상이한 데이터 청크들을 생성하는데 사용될 수 있다. 따라서, 이 방법이 상이한 시간 슬롯들로부터의 청크들을 갖는 데이터를 코딩하도록 동작 가능한 비디오 코덱들 및 오디오 코덱들에 특히 아주 적합하다는 것이 이해될 것이다. 상이한 프레임들 또는 섹션들은 상이한 데이터 청크들이고 그것들은 본 개시물에 의한 하나 이상의 방법들을 구현하는 인코더들을 사용함으로써 하나 이상의 청크들로 여전히 분리될 수 있다. 모든 이들 데이터 청크들은 예를 들어 동일한 프레임에서 또는 이전의 프레임에서 더 일찍 전달되는 임의의 테이블을 재사용할 수 있다.

[0031] 본 발명의 실시형태들은 코드 테이블들 또는 빈도 테이블들의 효율적인 전달을 가능하게 한다. 이는 테이블 전달 및/또는 저장에 필요한 데이터 통신 및/또는 데이터 저장 오버헤드를 감소시킨다. 각각의 개개의 데이터 청크에 대해 더 낮게 최적화된 코딩 테이블들을 이용함으로써 더 적은 데이터 청크들이 코딩되는 것이 또한 가능하다. 그리하여, 증가된 더 큰 압축 효율이 달성될 수 있는데, 이는 데이터 저장 용량, 송신 대역폭 및 에너지 소비가 감소될 수 있다는 것을 의미한다.

[0032] 데이터의 다양한 부분들의 빈도들은 종종 상호 상이하고, 종종 그것들의 상대적 데이터 엔트로피는 또한 상호

상이고, 이런 이유로 데이터를 다수의 부분들, 즉 데이터 청크들로 분리하는 것이 유익하다. 유익하게는, 상이한 코드 테이블들이, 그 부분들의 데이터의 성질 및/또는 데이터의 유형 및/또는 데이터의 내용에 의존하여, 그 부분들에 대해 사용되며; "성질"에 의하면 데이터의 하나 이상의 특성들 및/또는 파라미터들을 의미한다. 본 발명은 코드 테이블 또는 빈도 테이블의 전달이 이러한 데이터 청크들에 대해 최적화될 수 있다는 연관된 유익과 함께, 주어진 큰 데이터 파일이 더 작은 부분들, 즉 데이터 청크들로 더욱 효율적으로 분리되는 것을 가능하게 하는 방법들을 제공한다. 이 빅 데이터(big data) 파일 분리는 수반되는 데이터의 엔트로피를 수정하는 것에 관해서 실질적인 유익을 가능하게 하고, 그래서 통신될 인코딩된 데이터의 양을 고도로 감소시키는 것이 가능하다. 하나 이상의 데이터 청크들에서의 데이터 값들은 앞서 언급한 바와 같이 또한 분리될 수 있다. 이 데이터 값 분리, 즉 분할은, 예를 들어, MSB(most significant bits) 및 LSB(least significant bits)를 상호 분리함으로써 구현될 수 있다. 데이터 값들은 또한 옵션으로 두 개를 초과하는 별개의 데이터 값 청크들로 분리된다.

- [0033] 옵션으로, 본 방법은 인코딩된 데이터(E2)를 생성하기 위해 단계 (d)에서 하나 이상의 데이터 압축 알고리즘들을 적용하는 단계를 포함한다. 추가적인 옵션으로, 그 방법에서, 하나 이상의 데이터 압축 알고리즘들은, 허프만 인코딩, VLC, 엔트로피 인코딩, 산술 인코딩, 레인지 인코딩 중 적어도 하나를 포함하지만, 그것들로 제한되지는 않는다.
- [0034] 옵션으로, 그 방법은, 입력 데이터(D1)를 복수의 데이터 청크들로 분리하는 단계와 복수의 데이터 청크들을 실질적으로 병행하는 방식으로 프로세싱하기 위해 프로세서들의 병렬 아키텍처를 채용하는 단계를 포함한다.
- [0035] 옵션으로, 그 방법은, 함께 결합되는 다수의 데이터 값들에 기초하여 하나 이상의 인덱스 세트들을 생성하는 단계를 포함한다. 더한 옵션으로, 그 방법에서, 인덱스들(즉"indexes")은 R, G 및 B 화소 값들 또는 Y, U 및 V 화소 값들을 포함하는 하나 이상의 RGB 화소들, 또는 YUV 화소들로부터 도출된다.
- [0036] 더한 옵션으로, 그 방법은, 인코딩된 데이터(E2)에 포함될 때 데이터 청크들에 대한 달성가능 데이터 압축 비율의 함수로서, 인코딩된 데이터(E2)가 되게 데이터 청크들을 인코딩식 어셈블링하는 것 또는 비인코딩식 어셈블링하는 것 간을 동적으로 스위칭하는 단계를 포함한다.
- [0037] 옵션으로, 그 방법은, 심볼이 "코드 테이블의 변경"에 관련되는지 또는 "데이터의 말단"에 관련되는지를 나타내는 적어도 하나의 트레일링 비트를 인코딩된 데이터(E2) 속에 통합하는 단계를 포함한다.
- [0038] 옵션으로, 그 방법은, 주어진 데이터 청크에 대해, 주어진 데이터 청크에 존재하는 하나 이상의 심볼들을 참조하기 위해 요구된 인덱스들만을 실질적으로 생성하는 단계를 포함한다.
- [0039] 옵션으로, 전달되는 코드 테이블들은, 예를 들어 허프만 코딩을 채용함으로써 또한 압축되고, 그러한 코드 테이블 압축 방법에는 자신 소유의 연관된 하나 이상의 코드 테이블들이 또한 옵션으로 제공된다. 추가적인 옵션으로, 본 방법에서, 하나 이상의 코드 테이블들의 압축은 하나 이상의 보조 코드 테이블들을 채용한다.  
 옵션으로, 본 방법은, 하나 이상의 코드 테이블들이 후속하여 전송된 데이터를 디코딩하기 위해 디코더에서 채용되는 것을 가능하게 하는 방식으로 하나 이상의 코드 테이블들을 전달하는 단계를 포함한다.  
 옵션으로, 본 방법은, 하나 이상의 데이터베이스들, 및/또는 하나 이상의 프록시 데이터베이스들을 통해, 상기 하나 이상의 코드 테이블이 액세스될 가능성이 있는 경우를 나타내는 하나 이상의 식별 코드들을, 인코딩된 데이터(E2)에 포함시키는 단계를 포함한다.  
 옵션으로, 본 방법은, 다음의 유형들, 즉, 캡처된 오디오 신호들, 캡처된 비디오 신호들, 캡처된 이미지들, 텍스트 데이터, 지진계 데이터, 센서 신호들, 아날로그-디지털(ADC) 변환된 데이터, 생체의학 신호 데이터, 캘린더 데이터, 경제 데이터, 수학 데이터, 이진수 데이터 중 하나 이상의 유형의 데이터를 인코딩하도록 배열된다.
- [0040] 제 2 양태에 따르면, 대응하는 인코딩된 데이터(E2)를 생성하기 위하여 입력 데이터(D1)를 인코딩하는 인코더가 제공되는데, 그 인코더는,
- [0041] (a) 입력 데이터(D1)에 존재하는 심볼들을 분석하고, 입력 데이터(D1)를 하나 이상의 데이터 청크들로 분리 및/또는 변환하는 분석기;
- [0042] (b) 하나 이상의 코드 테이블들, 하나 이상의 빈도 테이블들, 하나 이상의 코드 워드 길이 테이블들, 하나 이상의 확률 테이블들 중 적어도 하나를, 심볼들의 발생의 함수로서 생성하는 생성기;
- [0043] (c) 각각의 데이터 청크에서의 심볼들을, 하나 이상의 코드 테이블들, 하나 이상의 빈도 테이블들, 하나 이상의

코드 워드 길이 테이블들, 하나 이상의 확률 테이블들 중 적어도 하나에서의 엔트리들에 관련시키는 하나 이상의 인덱스 세트들(즉, "인덱스들")을 컴퓨팅하는 컴퓨팅 엔진;

[0044] (d) 하나 이상의 빈도 테이블들, 하나 이상의 코드 테이블들, 하나 이상의 코드 워드 길이 테이블들, 하나 이상의 확률 테이블들, 하나 이상의 테이블들을 나타내는 정보와 함께, 각각의 데이터 청크 내의 심볼들을 인코딩된 데이터(E2)에 관련시키는 하나 이상의 인덱스 세트들의 정보를 어셈블링하는 데이터 어셈블러; 및

(e) 인코딩된 데이터(E2)에 하나 이상의 테이블들 내의 심볼들을 압축하는 압축기를 포함하는 것을 특징으로 한다.

[0045] 이러한 분리, 즉 분할은, 옵션으로 입력 데이터(D1)의 세분을 포함한다.

[0046] 옵션으로, (a)에서, 분리가 보통 행해지지만, 때때로 데이터를 새로운 데이터 청크들로 분리하는 일 없이 최적의 코드 테이블로 이용 가능한 데이터 청크를 압축하는 것이 또한 필요하다. 더구나, 때때로, 원래의 데이터는 분리되지 않지만, 대신, 새로운 데이터가, 예를 들어 가장 효율적으로 압축되는 것이 필요한 하나 이상의 데이터 청크들에 대해 하나 이상의 변환들에 의해 생성된다. 본 개시물에 의하여 구현되는 인코더들은 상이한 데이터 청크들을 생성하는데 사용될 수 있다. 따라서, 이 방법이 상이한 시간 슬롯들로부터의 청크들을 갖는 데이터를 코딩하는 비디오 코덱들 및 오디오 코덱들에 특히 아주 적합하다는 것이 이해될 것이다. 상이한 프레임들 또는 섹션들은 상이한 데이터 청크들이고 그것들은 본 개시물에 의한 방법을 구현하는 인코더들을 사용함으로써 하나 이상의 청크들로 여전히 분리될 수 있다. 모든 이들 데이터 청크들은 예를 들어 동일한 프레임에서 또는 이전의 프레임에서 더 일찍 전달되는 임의의 테이블을 재사용할 수 있다.

옵션으로, 인코더는, 하나 이상의 빈도 테이블들, 하나 이상의 코드 테이블들, 하나 이상의 코드 워드 길이 테이블들, 하나 이상의 확률 테이블들, 하나 이상의 테이블들을 나타내는 정보 중 적어도 하나와 함께, 각각의 데이터 청크에서의 심볼들을 관련시키는 하나 이상의 인덱스 세트들의 정보뿐만 아니라, 압축된 상기 심볼들을 포함하는 상기 인코딩된 데이터(E2)를 전달하도록 동작 가능하다.

옵션으로, 인코더는, 하나 이상의 테이블들 중 적어도 하나의 테이블이 후속하는 재사용을 위해 저장 가능한 방식으로 하나 이상의 테이블들 중 적어도 하나의 테이블을 전달하도록 동작 가능하다.

옵션으로, 인코더는 인코딩된 데이터(E2)를 생성하기 위해 데이터 어셈블러에서 하나 이상의 데이터 압축 알고리즘들을 적용하도록 동작 가능하다. 더한 옵션으로, 그 인코더에서, 하나 이상의 데이터 압축 알고리즘들은, 허프만 인코딩, VLC, 엔트로피 인코딩, 산술 인코딩, 레인지 인코딩 중 적어도 하나를 포함한다.

[0047] 삭제

[0048] 옵션으로, 그 인코더는 입력 데이터(D1)를 복수의 데이터 청크들로 분리하도록 그리고 복수의 데이터 청크들을 실질적으로 병행하는 방식으로 프로세싱하는 프로세서들의 병렬 아키텍처를 채용하도록 동작 가능하다.

[0049] 옵션으로, 그 인코더에서, 생성기는 함께 결합되는 다수의 데이터 값들에 기초하여 하나 이상의 인덱스 세트들을 생성하도록 동작 가능하다. 더한 옵션으로, 그 인코더에서, 인덱스들은 R, G 및 B 화소 값들 또는 Y, U, 및 V 화소 값들을 포함하는 하나 이상의 RGB-화소들 또는 YUV-화소들로부터 도출된다. 더한 옵션으로, 그 인코더는 인코딩된 데이터(E2)에 포함될 때 데이터 청크들에 대한 (달성 가능한) 달성가능 데이터 압축 비율의 함수로서 인코딩된 데이터(E2)가 되게 데이터 청크들을 인코딩식 어셈블링하는 것 또는 비인코딩식 어셈블링하는 것 간을 동적으로 스위칭하도록 동작 가능하다.

[0050] 옵션으로, 그 인코더는, 심볼이 "코드 테이블의 변경"에 관련되는지 또는 "데이터의 말단"에 관련되는지를 나타내는 적어도 하나의 트레일링 비트를 인코딩된 데이터(E2) 속에 통합하도록 동작 가능하다.

[0051] 옵션으로, 그 인코더에서, 생성기는, 주어진 데이터 청크에 대해, 주어진 데이터 청크에 존재하는 하나 이상의 심볼들을 참조하기 위해 요구된 인덱스들만을 실질적으로 생성하도록 동작 가능하다.

[0052] 옵션으로, 전달되는 코드 테이블들은, 예를 들어, 허프만 코딩을 채용함으로써 압축되고, 이 압축 방법은 옵션으로 때때로 자신 소유의 하나 이상의 연관된 코드 테이블들을 필요로 한다.

[0053] 옵션으로, 전달되는 코드 테이블들은 동일한 데이터 프레임에서 또는 다음의 데이터 프레임들에서 재사용될 수 있으며, 즉 데이터 청크 인코딩은 이 데이터 프레임에서의 또는 이전의 데이터 프레임들에서의 다른 데이터 청



크에 대해 그 전에 전달되는 임의의 테이블을 재사용할 수 있다.

- [0054] 옵션으로, 테이블을 전달하기 위한 최적의 구현예들이, 인코더를 구현할 때, 예를 들어 인코딩된 데이터에서, 또는, 예를 들어 하나 이상의 데이터베이스들, 하나 이상의 프록시 데이터베이스들 등으로부터 테이블이 액세스될 수 있는 경우를 나타내는 하나 이상의 식별 코드들을 인코딩된 데이터 내에 포함시킴으로써 유익하게 채용된다.
- [0055] 옵션으로, 데이터의 더욱 효율적인 인코딩, 인코딩된 데이터의 전달 및 인코딩된 데이터의 디코딩을 제공하기 위해, 전달된 그리고/또는 참조된 테이블들은, 앞서 언급한 바와 같이, 예를 들어 저장된 테이블의 인덱스가 전달된 때에 예를 들어 나중에 재사용되도록 저장되는 것이 유익하다. 이러한 접근법은 본 개시물에 의하여, 예를 들어 인코더로부터 대응하는 디코더로 통신될 필요가 있는 데이터의 볼륨을 감소시킬 수 있다.
- [0056] 제 3 양태에 따르면, 컴퓨터 관독가능 명령어들을 저장하고 있는 비밀시적 컴퓨터 관독가능 저장 매체를 포함하는 컴퓨터 프로그램 제품이 제공되며, 그 컴퓨터 관독가능 명령어들은 제 1 양태에 의하는 방법을 실행하기 위해 프로세싱 하드웨어를 포함하는 컴퓨터화 디바이스에 의해 실행 가능하다.
- [0057] 제 4 양태에 따르면, 제 2 양태에 의하는 인코더에 의해 생성된 인코딩된 데이터(E2)를 디코딩하는 방법이 제공되는데, 즉, 대응하는 디코딩된 데이터(D3)를 생성하기 위하여, 인코더에 의해 생성된 인코딩된 데이터(E2)를 디코딩하는 디코더에서의 방법이 제공되는데, 그 방법은,
  - [0058] (i) 인코딩된 데이터(E2)를 수신하고, 수신된 인코딩된 데이터로부터, 하나 이상의 빈도 테이블들, 하나 이상의 코드 테이블들, 하나 이상의 코드 워드 길이 테이블들, 하나 이상의 확률 테이블들, 하나 이상의 테이블들을 나타내는 정보와 함께, 하나 이상의 인덱스 세트들의 정보를 추출하는 단계;
  - [0059] (ii) 하나 이상의 데이터 체크들, 하나 이상의 코드 테이블들에서의 엔트리들의 압축된 심볼들, 하나 이상의 빈도 테이블들, 하나 이상의 코드 워드 길이 테이블들, 하나 이상의 확률 테이블들을 중 적어도 하나에서의 대응하는 심볼들을, 하나 이상의 인덱스 세트들로부터 컴퓨팅하는 단계;
  - [0060] (iii) 하나 이상의 코드 테이블들, 하나 이상의 빈도 테이블들, 하나 이상의 코드 워드 길이 테이블들, 하나 이상의 확률 테이블들 중 적어도 하나로부터의 정보를 사용하여, 하나 이상의 데이터 체크들을 심볼들로부터 재생성하는 단계; 및
  - [0061] (iv) 디코딩된 데이터(D3)를 생성하기 위해 하나 이상의 데이터 체크들을 결합 및/또는 변환하는 단계를 포함하는 것을 특징으로 한다.

옵션으로, 본 방법에서, 하나 이상의 테이블들 중 적어도 하나의 테이블이 미리 정의된다.
- [0062] 옵션으로, 그 방법은, 디코딩된 데이터(D3)를 트랜스코딩하여 대응하는 트랜스코딩된 데이터(D4)를 생성하는 단계 및/또는 인코딩된 데이터(E2)로부터 대응하는 트랜스코딩된 데이터(D4)를 생성하는 단계를 포함한다.
 

옵션으로, 본 방법은, 하나 이상의 테이블들을 그것들의 대칭성 및 대응하는 대칭성 표시 정보에 기초하여 재생성하는 단계를 포함한다.
- [0063] 옵션으로, 그 방법은, 하나 이상의 테이블들 중 적어도 하나의 테이블이 후속하는 재사용을 위해 저장 가능한 방식으로 하나 이상의 테이블들 중 적어도 하나의 테이블을 수신하는 단계를 포함한다.
- [0064] 옵션으로, 그 방법은, 디코딩된 데이터(D3)를 생성하기 위해 단계 (iv)에서 하나 이상의 데이터 압축해제 알고리즘들을 적용하는 단계를 포함한다. 더한 옵션으로, 그 방법에서, 하나 이상의 데이터 압축해제 알고리즘들은, 허프만 디코딩, VLC 디코딩, 엘트로퍼 디코딩, 산술 디코딩, 레인지 디코딩 중 적어도 하나를 포함한다.
- [0065] 옵션으로, 그 방법은, 복수의 데이터 체크들을 실질적으로 병행하는 방식으로 프로세싱하는 프로세서들의 병렬 아키텍처를 채용함으로써, 디코딩된 데이터(D3)를 생성하기 위해 복수의 하나 이상의 데이터 체크들을 결합하는 단계를 포함한다.
- [0066] 옵션으로, 그 방법은, 함께 결합되는 다수의 데이터 값들에 기초하여 하나 이상의 인덱스 세트들을 생성하는 단계를 포함한다. 더한 옵션으로, 그 방법에서, 인덱스들은 R, G 및 B 화소 값들 또는 Y, U, 및 V 화소 값들을 포함하는 하나 이상의 RGB-화소들 또는 YUV-화소들로부터 도출된다. 더한 옵션으로, 그 방법은, 인코딩된 데이터(E2)에 포함될 때 데이터 체크들에 대한 달성가능 데이터 압축해제 비율의 함수로서, 인코딩된 데이터(E2)가

되게 하나 이상의 데이터 청크들을 인코딩식으로 생성하는 것 또는 비인코딩식으로 생성하는 것 간을 동적으로 스위칭하는 단계를 포함한다.

- [0067] 옵션으로, 그 방법에서, 디코더는 심볼이 "코드 테이블의 변경"에 관련되는지 또는 "데이터의 말단"에 관련되는지를 나타내는 적어도 하나의 트레일링 비트를 인코딩된 데이터(E2)로부터 추출하도록 동작 가능하다.
- [0068] 옵션으로, 그 방법은, 주어진 데이터 청크에 존재하는 하나 이상의 심볼들을 참조하기 위해 요구된 실질적으로 그 인덱스들만으로부터 주어진 데이터 청크를 생성하는 단계를 포함한다.
- [0069] 옵션으로, 그 방법은 인코딩된 데이터(E2)에 포함된 하나 이상의 코드 테이블들을 압축해제하는 단계를 포함한다. 더한 옵션으로, 그 방법은, 허프만 디코딩을 채용함으로써 하나 이상의 코드 테이블들을 압축해제하는 단계를 포함한다. 더한 옵션으로, 그 방법에서, 하나 이상의 코드 테이블들의 압축해제는 하나 이상의 보조 코드 테이블들을 채용한다.
- [0070] 옵션으로, 그 방법은, 하나 이상의 코드 테이블들이 후속하여 전송된 데이터를 디코딩하기 위해 디코더에서 채용되는 것을 가능하게 하는 방식으로 하나 이상의 코드 테이블들을 수신하는 단계를 포함한다.
- [0071] 옵션으로, 그 방법은, 하나 이상의 데이터베이스들, 및/또는 하나 이상의 프록시 데이터베이스들을 통해, 하나 이상의 코드 테이블이 액세스될 가능성이 있는 경우를 나타내는 하나 이상의 식별 코드들을 인코딩된 데이터(E2)로부터 추출하는 단계를 포함한다.
- [0072] 옵션으로, 그 방법은 다음의 유형들, 즉, 캡처된 오디오 신호들, 캡처된 비디오 신호들, 캡처된 이미지들, 텍스트 데이터, 지진계 데이터, 센서 신호들, 아날로그-디지털(ADC) 변환된 데이터, 생체의학 신호 데이터, 캘린더 데이터, 경제 데이터, 수학 데이터, 이진수 데이터 중 하나 이상의 유형의 데이터를 디코딩하는 단계를 포함한다.
- [0073] 옵션으로, 그 방법은, 복수의 데이터 소스들로부터 인코딩된 데이터(E2)를 수신하는 단계와, 소스들로부터 제공된 데이터를 결합하여 인코딩된 데이터(E2)를 재생성하는 단계를 포함한다.
- [0074] 제 5 양태에 따르면, 컴퓨터 판독가능 명령어들을 저장하고 있는 비일시적 컴퓨터 판독가능 저장 매체를 포함하는 컴퓨터 프로그램 제품이 제공되며, 컴퓨터 판독가능 명령어들은 제 4 양태에 의하는 방법을 실행하기 위해 프로세싱 하드웨어를 포함하는 컴퓨터화 디바이스에 의해 실행 가능하다.
- [0075] 제 6 양태에 따르면, 제 2 양태에 의하는 인코더에 의해 생성된 인코딩된 데이터(E2)를 디코딩하는 디코더가 제공되는데, 즉, 대응하는 디코딩된 데이터(D3)를 생성하기 위하여, 인코더에 의해 생성된 인코딩된 데이터(E2)를 디코딩하는 디코더가 제공되는데, 그 디코더는,
  - [0076] (i) 인코딩된 데이터(E2)를 수신하고 수신된 인코딩된 데이터로부터, 하나 이상의 빈도 테이블들, 하나 이상의 코드 테이블들, 하나 이상의 코드 워드 길이 테이블들, 하나 이상의 확률 테이블들, 하나 이상의 테이블들을 나타내는 정보 중 적어도 하나와 함께, 하나 이상의 인덱스 세트들의 정보를 추출하도록;
  - [0077] (ii) 하나 이상의 데이터 청크들, 하나 이상의 코드 테이블들에서의 엔트리들의 압축된 심볼들, 하나 이상의 빈도 테이블들, 하나 이상의 코드 워드 길이 테이블들, 하나 이상의 확률 테이블들 중 적어도 하나에서의 대응하는 심볼들을, 하나 이상의 인덱스 세트들로부터 컴퓨팅하도록;
  - [0078] (iii) 하나 이상의 코드 테이블들, 하나 이상의 빈도 테이블들, 하나 이상의 코드 워드 길이 테이블들, 하나 이상의 확률 테이블들 중 적어도 하나로부터의 정보를 사용하여, 하나 이상의 데이터 청크들을 심볼들로부터 재생성하도록; 그리고
  - [0079] (iv) 디코딩된 데이터(D3)를 생성하기 위해 하나 이상의 데이터 청크들을 결합 및/또는 변환하도록 동작 가능하다는 것을 특징으로 한다.
- [0080] 옵션으로, 디코더는, 디코딩된 데이터(D3)를 트랜스코딩하여 대응하는 트랜스코딩된 데이터(D4)를 생성하고 그리고/또는 인코딩된 데이터(E2)로부터 대응하는 트랜스코딩된 데이터(D4)를 생성하는 트랜스코더를 더 포함한다.  
 옵션으로, 디코더(60)는, 하나 이상의 테이블들을 그것들의 대칭성 및 상기 디코더에 제공된 대응하는 대칭성 표시 정보에 기초하여 재생성하도록 동작 가능하다.
- [0081] 옵션으로, 디코더는, 하나 이상의 테이블들 중 적어도 하나의 테이블이 후속하는 재사용을 위해 저장 가능한 방

식으로 하나 이상의 테이블들 중 적어도 하나의 테이블을 수신하도록 동작 가능하다.

- [0082] 옵션으로, 디코더는 디코딩된 데이터(D3)를 생성하기 위해 (iv)에서 하나 이상의 데이터 압축해제 알고리즘들을 적용하도록 동작 가능하다. 더한 옵션으로, 디코더에서, 하나 이상의 데이터 압축해제 알고리즘들은, 허프만 디코딩, VLC 디코딩, 엔트로피 디코딩, 산술 디코딩, 레인지 디코딩 중 적어도 하나를 포함한다.
- [0083] 옵션으로, 디코더는, 복수의 데이터 청크들을 실질적으로 병행하는 방식으로 프로세싱하는 프로세서들의 병렬 아키텍처를 채용함으로써, 디코딩된 데이터(D3)를 생성하기 위해 복수의 하나 이상의 데이터 청크들을 결합하도록 동작 가능하다.
- [0084] 옵션으로, 디코더는, 함께 결합되는 다수의 데이터 값들에 기초하여 하나 이상의 인덱스 세트들을 생성하도록 동작 가능하다. 더한 옵션으로, 그 디코더에서, 인덱스들은 R, G 및 B 화소 값들 또는 Y, U, 및 V 화소 값들을 포함하는 하나 이상의 RGB-화소들 또는 YUV-화소들로부터 도출된다. 더한 옵션으로, 디코더는, 인코딩된 데이터(E2)에 포함될 때 데이터 청크들에 대한 달성가능 데이터 압축해제 비율의 함수로서, 인코딩된 데이터(E2)가 되게 하나 이상의 데이터 청크들을 인코딩식으로 생성하는 것 또는 비인코딩식으로 생성하는 것 간을 동적으로 스위칭하도록 동작 가능하다.
- [0085] 옵션으로, 그 디코더는 심볼이 "코드 테이블의 변경"에 관련되는지 또는 "데이터의 말단"에 관련되는지를 나타내는 적어도 하나의 트레일링 비트를 인코딩된 데이터(E2)로부터 추출하도록 동작 가능하다.
- [0086] 옵션으로, 그 디코더는, 주어진 데이터 청크에 존재하는 하나 이상의 심볼들을 참조하기 위해 요구된 실질적으로 그 인덱스들만으로부터 주어진 데이터 청크를 생성하도록 동작 가능하다.
- [0087] 옵션으로, 그 디코더는 인코딩된 데이터(E2)에 포함된 하나 이상의 코드 테이블들을 압축해제하도록 동작 가능하다. 더한 옵션으로, 디코더는 허프만 디코딩을 채용함으로써 하나 이상의 코드 테이블들을 압축해제하도록 동작 가능하다. 더한 옵션으로, 그 디코더에서, 하나 이상의 코드 테이블들의 압축해제는 하나 이상의 보조 코드 테이블들을 채용한다.
- [0088] 옵션으로, 디코더는, 하나 이상의 코드 테이블들이 후속하여 전송된 데이터를 디코딩하기 위해 디코더(60)에서 채용되는 것을 가능하게 하는 방식으로 하나 이상의 코드 테이블들을 수신하도록 동작 가능하다.
- [0089] 옵션으로, 디코더는, 하나 이상의 데이터베이스들, 및/또는 하나 이상의 프록시 데이터베이스들을 통해, 하나 이상의 코드 테이블이 액세스될 가능성이 있는 경우를 나타내는 하나 이상의 식별 코드들을 인코딩된 데이터(E2)로부터 추출하도록 동작 가능하다.
- [0090] 옵션으로, 디코더는, 다음의 유형들, 즉, 캡처된 오디오 신호들, 캡처된 비디오 신호들, 캡처된 이미지들, 텍스트 데이터, 지진계 데이터, 센서 신호들, 아날로그-디지털(ADC) 변환된 데이터, 생체의학 신호 데이터, 캘린더 데이터, 경제 데이터, 수학 데이터, 이진수 데이터 중 하나 이상의 유형의 데이터를 디코딩하도록 동작 가능하다.
- [0091] 옵션으로, 디코더는, 복수의 데이터 소스들로부터 인코딩된 데이터(E2)를 수신하도록, 그리고 소스들로부터 제공된 데이터를 결합하여 인코딩된 데이터(E2)를 재생성하도록 동작 가능하다.
- [0092] 제 7 양태에 따르면, 대응하는 인코딩된 데이터(E2)를 생성하기 위하여 입력 데이터(D1)를 인코딩하는 제 2 양태에 의하는 적어도 하나의 인코더와, 디코딩된 출력 데이터(D3)를 생성하기 위하여 인코딩된 데이터(E2)를 디코딩하는 제 6 양태에 의하는 적어도 하나의 디코더를 포함하는 코덱이 제공된다.
- [0093] 옵션으로, 코덱은 적어도 하나의 인코더와 적어도 하나의 디코더가 상호 공간적으로 원격이고 데이터 통신 네트워크를 통해 서로 상호 커플링되도록 구현된다. 더한 옵션으로, 코덱은, 데이터 통신 네트워크가 피어 투 피어 통신 네트워크의 방식으로 구성되도록 구현된다. 옵션으로, 코덱은 그것의 인코더 및 그것의 디코더가 그것들을 통한 그것들의 데이터 프로세싱에서 대칭적이도록 구현되며; 다르게 말하면, 인코더에서 실행되는 프로세싱 기능들은 대응하는 역 기능들로서 구현되고 디코더에서 역 순으로 실행된다.
- [0094] 본 발명의 특징들이 첨부 청구항들에 의해 정해진 발명의 범위로부터 벗어나는 일없이 다양한 조합들로 결합되는 것이 가능할 수 있다는 것이 이해될 것이다.

**도면의 간단한 설명**

- [0095] 본 게시물의 실시형태들은 다음의 도면들을 참조하여 예로서만 이제 설명될 것이다.

도 1은 데이터를 인코딩하고 디코딩하는 알려진 인코더 및 알려진 디코더의 예시도이며;

도 2는 본 개시물의 일 실시형태에 의하는 데이터를 인코딩하는 방법의 예시도이며;

도 3a는 본 개시물에 의하는 인코더 및 디코더, 총칭하여 코덱의 실시형태의 예시도이며; 그리고

도 3b는 본 개시물에 의하는 인코더 및 디코더, 총칭하여 코덱의 대체 실시형태의 예시도인데, 디코더에서의 디코딩된 데이터(D3)는 트랜스코딩된 데이터(D4)를 생성하기 위해 트랜스코딩된다.

첨부의 도면들에서, 밀줄 친 번호는 밀줄 친 번호가 위치되는 아이템 또는 밀줄 친 번호가 인접하는 아이템을 나타내기 위해 채용된다. 번호가 비 밀줄 쳐지고 연관된 화살표가 동반되는 경우, 비 밀줄친 번호는 화살표가 가리키는 일반 아이템을 식별하는데 사용된다.

**발명을 실시하기 위한 구체적인 내용**

[0096] 개괄하여, 본 개시물은, 예를 들어, 인코더들, 디코더들, 코덱들 및 연관된 동작 방법들에 관련된다. 더구나, 본 개시물의 실시형태들은 알려진 방법들과 비교되는 코드 테이블들, 빈도 테이블들, 코드 워드의 길이(length of code word) 테이블들, 또는 확률 테이블들의 전달을 개선시키는 동작을 할 수 있다. 더욱이, 본 개시물의 실시형태들은 더 적은 비트들이 하나 이상의 테이블들의 전달을 위해 사용되는 것을 가능하게 하는 방식으로 하나 이상의 허프만 트리들을 또한 전달할 수 있으며; 이로써, 특히 하나 이상의 테이블들을 동반하는 코딩된 데이터의 양이 비교적 작을 때 인코딩 동안 이용 가능한 데이터 압축 비율에서 증가가 제공된다. 코드 테이블들, 빈도 테이블들, 코드 워드 길이 테이블들 또는 확률 테이블들은 많은 상이한 엔트로피 코딩 방법들에 대해, 예를 들어 허프만 코딩, 산술 코딩, 레인지 코딩과 같지만 그것들로 제한되지 않는 가변 장 코딩(VLC) 방법들에 대해 필요하다. 인코더들, 예를 들어 송신기들, 및 디코더들, 예를 들어 수신기들 양쪽 모두는, 아래에서 설명될 바와 같은 방법들을 유익하게 채용한다.

[0097] 아래에서 설명되는 본 개시물의 실시형태들은 저장 및 통신되는 데이터의 볼륨이 시간의 경과와 함께 급속히 증가하는 세상에 관계가 있다. 이러한 데이터의 저장 및 송신은 상당한 저장 용량, 송신 대역폭 및 에너지를 소비한다. 세상에서의 데이터의 대부분은 캡처된 오디오 신호들, 캡처된 비디오 신호들, 캡처된 이미지들, 텍스트 데이터, 지진계 데이터, 센서 신호 데이터, 아날로그-디지털(ADC) 변환된 데이터, 생체의학 신호 데이터, 캘린더 데이터, 경제 데이터, 수학 데이터, 이진수 데이터이지만 그것들로 제한되지 않는다. 본 개시물의 실시형태들은 모든 앞서 언급된 데이터 유형들과, 또한 데이터의 다른 유형들에 대한 인코딩된 데이터의 양을 감소시키도록 동작 가능하며; 이로써 코드 테이블들, 빈도 테이블들, 코드 워드 길이 테이블들 또는 확률 테이블들의 효율적인 전달이 가능하게 되며, 따라서 데이터의 엔트로피를 효율적으로, 즉 데이터의 사이즈를 효율적으로 감소시키는 더 작은 데이터 청크들의 사용을 가능하게 한다. 더구나, 더 작은 데이터 청크들은 결과들을 더 빠르게 출력하기 위해 병렬 프로세스들을 통해 효과적으로 핸들링될 수 있고, 이러한 병렬처리(parallelism)는 현대의 마이크로프로세서 아키텍처에서, 특히 마이크로프로세서들의 장래의 구성들, 예를 들어 데이터 프로세서 어레이들 및 RISC(reduced instruction set computer) 프로세서들의 고속 구성들에서 일반적이다.

[0098] 주어진 인코딩 방법을 위해, 대응하는 코드 테이블이, 예를 들어 비트들로서 표현되는 코드 워드들의 길이들, 코드 워드들을 나타내는 코드들, 및 코드 워드들의 인덱스들(즉 "indexes")을 나타내는 정보를 포함한다. 코드 테이블은 코드 워드들의 길이들로부터 또한 생성될 수 있다. 코드 워드들의 인덱스들(즉 "indexes")은 코드 워드들로 코딩되는 원래의 심볼들에 대응하는 값을 나타낸다. 마찬가지로, 빈도 테이블은 심볼들의 발생 빈도들 및 심볼의 인덱스들을 포함한다. 심볼들의 인덱스들은 각각 인덱스들에 의해 코딩된 원래의 심볼들의 값들을 나타낸다. 빈도 테이블은 확률 테이블로 변환될 수 있고 확률 테이블은 빈도 테이블의 대강 추정치로서 사용될 수 있다. 빈도 테이블 및 코드 워드들의 길이 간의 그리고 그 반대의 변환이 또한 행해질 수 있다. 앞서 언급된 테이블들 중의 임의의 것이 전달될 때, 이러한 전달에서의 하나의 매우 중요한 파라미터가 테이블의 최대 인덱스이다. 테이블의 최대 인덱스는 얼마나 많은 상이한 심볼들, 또는 얼마나 많은 가능한 상이한 심볼들이, 전달된 테이블에서, 그리고 또한 입력 데이터에서 이용 가능한지를 나타낸다. 예를 들어, 주어진 데이터가 다음 이면:

[0099] 4, 3, 0, 1, 0, 4, 3, 4,

[0100] 실제 최대 인덱스는 4이고, 최소 인덱스는 0인데, 이는 데이터에 존재하는 상이한 심볼들(0, 1, 2, 3, 4)이 잠재적으로  $5(\max - \min + 1 = 4 - 0 + 1)$  개가 있다는 것을 의미한다. 데이터에 존재하는 상이한 심볼들(0, 1, 3, 4)이 단지 4 개만 있기 때문에, 테이블은 '4', 즉 가능한 상이한 심볼들 대신, '3'을 최대 인덱스, 즉 이용

가능한 상이한 심볼들의 카운트로서 사용하는 것에 의해 옵션으로 또한 전달된다. 그 값 3이 테이블의 최대 인덱스 값을 위해 사용될 때, 어떤 심볼들이 각각의 테이블 인덱스들을 위해 사용되는지에 관한 정보를 전달하기 위해 어떤 다른 메커니즘이 요구된다.

[0101] 심볼들이 순서대로 있을 때, 최대 인덱스(4)와 가용성 비트들, 예를 들어, 이 예의 경우 11011가 옵션으로 전달되고 있다. 이러한 전달은 테이블 인덱스 0 = 인덱스로서 마킹되는 심볼 0이도록 매핑되고, 연관된 심볼 쌍은 그러면 (0, 0)이 될 것이다. 마찬가지로, 인덱스 및 심볼 쌍들의 나머지는 (1, 1), (2, 3) 및 (3, 4)이다. 상이한 테이블 인덱스들에 대해 사용되는 인덱스들을 정의하기 위해 인덱스 및 심볼 쌍들을 직접 사용하는 것과 그 다음에 테이블의 최대 인덱스를 '3'으로 하여 전달하는 것이 또한 옵션으로 가능하다. 이는, 예를 들어 전달된 테이블에서의 심볼들이 그것들의 빈도들에 기초하여 저장될 때 매우 가치 있는 방법이다.

[0102] 예를 들어, 위에 경우에서, 인덱스 및 심볼 쌍들은, 예를 들어, (0, 4), (1, 0), (2, 3), 및 (3, 1)이다. 때때로, 사용된 인덱스 및 심볼 쌍들은 미리 정의되는 반면, 다른 때에는 사용되는 심볼의 인덱스와 인덱스 쌍 테이블이 전달되며, 다른 상이한 상황에서는, 인덱스 및 심볼 쌍들이 인코딩된 데이터(E2)와 함께 전달된다. 또 다른 상이한 상황에서, 디코더(60)는 사용되는 인덱스 및 심볼 쌍들을 알려진 로케이션으로부터 추출하도록 동작 가능하며; 또 다른 상이한 상황에서, 디코더(60)는 대응하는 로케이션 정보가 인코딩된 데이터(E2)와 함께 전달되었던 로케이션으로부터 사용되는 인덱스 및 심볼 쌍들을 추출하도록 동작 가능하다.

[0103] 몇몇 상황들에서, 대칭성의 도움으로 전달될 테이블을 근사화하는 것이 유리하다. 대칭성을 이용하는 것은, 테이블이 코드 길이들에 기초하는지, 확률들에 기초하는지 또는 발생 빈도들에 기초하는지에 관계 없이, 요구될 전체 테이블을 전달하는 것보다는 더 작은 데이터 사이즈로 사용되는 코딩 테이블을 전달하는 것을 가능하게 한다. 더구나, 인코더(50) 및 디코더(60) 둘 다에서 대칭적 테이블을 생성하는 것이 더 빠르는데, 그 테이블이 적은 엘리먼트들, 즉 더 적은 심볼들을 포함하기 때문이다. 그러나, 대칭성을 사용하는 것은 테이블을 약간 최적화되지 않게 하지만, 전달된 테이블 데이터 사이즈에서 얻어진 절약은, 특히 전송된 데이터의 양이 상당히 적다면 그 손실을 전적으로 또는 그 이상으로 보상한다.

[0104] 위의 최신의 예에서, 대칭성은 테이블을 전달함에 있어서 옵션으로 이용되는데, 심볼 값 '0'이 '1'보다 더욱 가능성이 있고 대응하여, 심볼 값 '4'가 '3'보다 더욱 가능성이 있기 때문이다. 더구나, 심볼 값들 '0' 및 '4'의 확률들은 서로 근접하고, 대응하여, 심볼 값들 '1' 및 '3'의 확률들이 또한 서로 근접한다. 그러나, 값 '2'는 데이터에서 전혀 발생하지 않으며, 그러므로 그 값은 그것이 검사되는 방향에 무관하게 가장 비사실적 값이다.

[0105] 대칭성이 이용되는 경우들에서, 코딩 테이블이 대칭적으로 대응하는 엘리먼트들의 발생들의 합들에 기초하여 그렇게 항상 생성될 수 있다. 이러한 경우에, 심볼 값 '0' 및 '4'의 전부 합해서 5 회의 발생들이 있고, 대응하여, 심볼 값 '1' 및 '3'의 전부 합해서 3 회의 발생들이 있다. 엘리먼트 '2'는 데이터에서 전혀 발생하지 않고, 그러므로 그것은 자신 소유의 심볼이 반드시 주어질 필요는 없다. 그러나, 몇몇 상황들에서, 심볼이 엘리먼트 '2'에 대해서도 생성될 수 있고, 그 경우에, 대칭성이 이용될 때, 그것은 우측 테이블 및 좌측 테이블 둘 다에 포함될 것이다.

[0106] 따라서, 범위들에 대한 발생들의 카운트들이 5, 3 및 (0)이고, 따라서, 심지어 최적의 발생빈도 기반 레인지 테이블이 당연히 3, 2, 0, 1, 2더라도, 사용되는 범위 테이블이 심볼들(0 내지 4)에 대해 5, 3, 0, 3, 5일 그런 방식으로 범위 코딩 테이블이 옵션으로 생성되며, 이는, 발생 빈도에 대한 두 개의 값들만이 전송, 즉 전달될 필요가 있는, 대칭성에 기초하는 위에서 설명된 옵션의 구현예와 비교하여, 발생 빈도를 위해 네 개의 값들의 전달을 요구할 것이다.

[0107] 이 대칭성에 기초한 동일한 아이디어는 옵션으로 허프만 코딩과 같은 다른 코딩 방법들과 함께 사용되고, 그 경우에, 대칭성 기반 테이블이, 예를 들어, 좌측 값들이 코드를 수용하고 우측 테이블 값들이 코드 값 '1'을 수용하는 테이블일 것이다. 따라서, 허프만 코드 워드들은, 예를 들어, 01, 00일 것이고, 10, 11이 존재하지 않는다. 대안으로, 장래에 존재할 값 '2'에 대한 옵션을 유보하고자 한다면, 그 코드들은 01, 001, 000/100, 101, 11일 것이다. 이 구현예에서, 원칙적으로, 두 개의 코드 길이들만이 (1 및 1)을 전송/전달하기 위해 필요할 것이거나, 또는 아니면 세 개의 코드 길이들이 (1, 2, 2)를 전송/전달하기 위해 필요할 것이고, 사용되는 테이블은 그 테이블이 대칭성을 이용한다고 알려진 경우들의 디코더(60)에서 완전히 복원 가능할 것이다. 더 긴 테이블들에서, 장점들이 더욱 뚜렷이 관찰 가능할 것이다.

[0108] 테이블이 대칭성을 이용하는지의 여부에 관한 정보 조각(piece of information)이, 옵션으로 이전에 이미 알려지거나, 또는 아니면 최적의 테이블이 전송되는지 또는 미리 정의된 테이블이 전송되는지, 또는 이전의 테이블

들로부터 동적으로 생성되었던 테이블이 전송되는지에 관한 정보 조각과 동일한 방식으로 송신/전달된다는 것이 이해될 것이다. 테이블이 대칭성을 이용하는지의 여부에 관한 정보 조각의 전달은 사용되고 있는 테이블의 인덱스를 디코더(60)로 전송함으로써 실행된다.

- [0109] 예를 들면 다음과 같다:
- [0110] (i) 테이블 인덱스 '0'은 테이블이 그 전부가 전송/전달됨을 의미하며;
- [0111] (ii) 테이블 인덱스 '1'은 테이블이 대칭적이라는 것과 그 테이블의 절반만이 전달될 것이라는 것을 의미하며;
- [0112] (iii) 테이블 인덱스들(2 내지 63)은 미리 정의된 테이블이 사용되고 있다는 것을 의미하고; 및
- [0113] (iv) 테이블 인덱스들(64 내지 127)은 이전에 전달 및 저장된 동적 테이블이 사용되고 있다는 것을 의미한다.
- [0114] 대칭적 테이블들이 미리 정의된 바와 같이 또는 동적으로 저장된 테이블들로서 또한 이용될 수 있다는 것이 이해될 것이다.
- [0115] 옵션으로, 다양한 코딩 방법들이, 예를 들어, ODelta 코딩과 함께 또는 그것 없이 사용되는데, ODelta 코딩은 데이터를 0 및 1 값들의 시퀀스가 되게 상이하게 코딩하는 것과 카운팅 워드랩(counting wraparound)을 채용하는 것을 수반한다. 더구나, 이들 다양한 코딩 방법들을 이용할 때, 그 방법들을 사용되고 있는 테이블의 테이블 인덱스와 조합하여 채용하는 것과, 또한 ODelta 방법이 코딩 전에 데이터에 대해 실행되었는지의 여부와, 또한 디코더(60)가 디코딩 후에 역의 동작을 대응하여 실행해야만 하는지의 여부를 옵션으로 표현하는 것이 유리하다.
- [0116] 이러한 경우에, 예를 들어 테이블 인덱스들은 그렇지 않으면 동일하지만, 값(128<sub>dec</sub>)은 ODelta가 사용되고 있음을 나타내기 위해 추가된다. 이 삽입이 수행되지 않았다면, ODelta는 코딩 전에 데이터에 대해 실행되지 않았다. 물론, 옵션으로, 다른 값들이 테이블 인덱스 값에 또한 추가될 수 있지만; 테이블 인덱스가 어떤 종류의 테이블이 사용되고 있는지와, 인코딩된 데이터와 함께, 어떤 종류의 부가적인 데이터가 테이블 인덱스로 송신/전달되는지를 표현한다는 것이 본질적으로 이해될 것이다.
- [0117] 도 2와 도 3a를 참조하면, 입력 데이터(D1)를 인코딩하여 대응하는 인코딩된 출력 데이터(E2)를 생성하는 방법의 단계들의 예시도가 제공되어 있는데, 그 방법의 단계들은 전반적으로 10에 의해 나타내어지고, 옵션으로는 하나 이상의 빈도 테이블들(25)을 생성하는 단계 20, 하나 이상의 코딩 테이블들(35)을 생성하는 단계 30, 입력 데이터(D1)를 분석하여 가장 적합한 인코딩 접근법을 선택하는 단계 40 중 하나 이상을 채용하지만 그것들로 제한되지는 않는다. 입력 데이터(D1)를 인코딩하는 이러한 방법(10)을 구현할 때, 입력 데이터(D1)에 존재하는 하나 이상의 심볼들을 대응하는 인덱스들(즉 "*indexes*")로 변경하는 어떤 메커니즘이 이용 가능하게 되어야 하며; 예를 들어, 인덱스들에 포함된 주어진 인덱스가, 예를 들어 화소 어레이 이미지에서의 화소 값과 동일하다. 그 인덱스는 화소 값 빼기 화소 어레이 이미지에 존재하는 최소 화소 값과 또한 동일할 수 있다. 이러한 상황에서, 그 방법은 인코딩된 출력 데이터(E2)에서 최소 화소 값을 전달하는 것, 즉 방법(10)을 채용함으로써, 또는 그것의 역을 채용함으로써 인코더(50)로부터 대응하는 디코더(60)로 어떻게든 전달되는 것이 또한 필요한데, 그렇지 않으면 디코더(60)는 대응하는 주어진 심볼을 그것의 주어진 인덱스의 사용을 통해 그것의 원래의 값으로 디코딩할 수 없기 때문이다. 그 인덱스는 다수의 정보로부터, 예를 들어 이산 코사인 변환들(DCT들) 중 하나 이상, 절대 AC 계수 값들, AC 계수의 부호(sign)들, 자신과 이전의 영이 아닌 AC/DC 계수 간의 영의 AC 계수들의 런(run), 및 현재의 AC 계수에 관계된 정보를 나타내는 표시 플래그를 포함하는 개개의 AC 계수들, 그리고 마지막 영이 아닌 AC 계수를 통해 또한 생성될 수 있다. 그 인덱스는 또한, 예를 들어 8-비트 R, G 및 B 화소 값들을 포함하는 24-비트 RGB-화소 또는 두 개의 5-비트 Y 화소 값들을 포함하는 10-비트 값을 함께 결합하는 다수의 화소 값들에 기초하여 생성된 것일 가능성이 또한 있다.
- [0118] 도 3b를 참조하면, 대응하는 트랜스코딩된 데이터(D4)를 생성하기 위한 디코더(60)에서의 데이터(D3)의 트랜스코딩은 트랜스코더(70)를 통해, 예를 들어 다음의 멀티캐스팅 상황에서 옵션으로 구현될 수 있다;
- [0119] (i) 각각의 디바이스가 인코딩된 데이터(E2)를 수신하는 디코더(60)를 호스팅하는 다수의 디바이스들이 있는 상황;
- [0120] (ii) 다수의 디바이스들 중 적어도 일부가 상호 상이하고, 연관된 상호 상이한 출력 포맷들, 예를 들어 디스플레이 스크린 레이아웃, 해상도, 앰플리튜드 비, 디스플레이 스크린 드라이버 버퍼 용량 등을 갖는 상황.
- [0121] 데이터(D3)의 트랜스코딩은 하드웨어에 의해 호환 가능한 방식으로 핸들링될 수 있고 그리고/또는 디바이스들의

소프트웨어 계층들을 지원하는 대응하는 데이터(D4)를 생성하는 것이 요구되며; 그 디바이스들은 옵션으로는 컴퓨팅 하드웨어, 예를 들어 스마트폰들, 전문 과학 장비, 텔레비전들, 하이파이 장치, 화상회의 장치 등에 기초한다. 트랜스코더(70)는 소프트웨어에서 그리고/또는 전용 데이터 프로세싱 하드웨어, 예를 들어 ASIC에서 구현된다.

[0122] 아래에서 언급되는 바와 같이, 방법(10)을 구현할 때, 하나 이상의 심볼 값들을 하나 이상의 대응하는 인덱스들로 변경하는 단계가 항상 포함되어 있어야 하고, 이 단계, 또는 그 역은, 디코더(60)에게 통신되어야 하거나 또는 그렇지 않으면 그것들은 인코더(50) 및 디코더(60) 둘 다를 위해 사전 설정되어야 한다. 이러한 통신을 달성하는 가장 쉬운 접근법이 주어진 심볼 값 및 그것의 대응하는 인덱스 값 간의 직접적인 관계를 채용하는 것이며, 예를 들어 인덱스 값이 대응하는 화소 값과 동일하거나, 또는 그 인덱스 값은, 예를 들어 다음에 의해 나타내어지는 바와 같이  $S =$  부호 플래그,  $V = 10$ -비트 계수 값,  $R = 6$ -비트의 0이 아닌 런 값, 및  $L =$  마지막 플래그를 나타내는 비트들에 기초하는 숫자이다:

[0123]  $S V V V V V V V V V V R R R R R R L$

[0124] 예를 들어, 직접적인 관계가 채용될 때 다음의 다수의 이유들로 직접적인 관계를 사용하는 것이 항상 가능하지는 않다:

[0125] (a) 심볼들의 데이터, 인덱스들(즉 "*indexes*"), 빈도들, 확률들 또는 길이들을 인코딩 또는 디코딩하는 것이 불가능하거나 또는 비효율적이거나;

[0126] (b) 상이한 인덱스들의 양이 엄청나게; 또는

[0127] (c) 모든 심볼들이 이용 가능한 빈도 정보를 갖고 있지는 않고 그런 이유로 몇몇 알고리즘들은 이들 심볼들에 대한 코드들을 생성할 수 없다.

[0128] 이러한 문제 (a) 내지 문제 (c) 중 일부는, 적어도 부분적으로, 이스케이프 코드들을 사용하여 또는 모든 심볼들에 대한 빈도 정보를 생성하는 어떤 로직을 사용하여 해소될 수 있다. 꽤 자주, 심볼들을 인덱스들(또한 "*indexes*"이라고 알려짐)로 변환하는 어떤 다른 접근법을 사용하는 것이 여전히 유익하다. 하나의 접근법은 이용 가능한 심볼들을 위해 사용되는 인덱스들을 특징하는 어떤 룩업 테이블(look-up-table, LUT)이 있는 것을 항상 보장하는 것을 수반하며; 여기서, 이스케이프 코드는 테이블들을 인코딩하는 사이즈를 감소시키는데 매우 유익하다. 이 LUT는 인코더(50)에서 그리고 디코더(60)에서 이용 가능해야 하거나, 또는 인코더(50)로부터 디코더(60)로, 또는 반대의 경우로 전달되어야 한다. 더욱 최적의 코딩이 더 나은 압축이 달성되는 것을 가능하게 하기 위해 필요할 때, 이용 가능한 LUT들의 인덱스에 기초하여 선택될 수 있는 유익하게 채용된 다수의 테이블들이 있다. 그러나, 이는 때때로 실용적이지 않은데, 빈도, 또는 코드 워드의 길이, 그 조합들이 너무 엄청나서 모든 상이한 테이블들을 데이터 메모리에 저장할 수 없거나, 또는 이러한 LUT들의 전달이 너무 많은 데이터가 인코더(50) 및 디코더(60) 간에 통신되는 것을 요구하기 때문이다. 따라서, 본 개시물에 의하는 방법들(10)은 인덱스 하나 이상의 변환들에 적합한 심볼들을 사용하여 달성될 수 있는 인코더(50)로부터 디코더(60)로의 빈도들, 코드 워드들의 길이, 또는 확률들의 효율적인 송신을 가능하게 한다. 코딩 방법이 빈도 정보에 의하여 제공된 더욱 정확한 정보를 이용할 수 없다면, 예를 들어 VLC 코딩 방법들이 빈도 정보를 이용할 수 없지만 역으로 산술 코딩 및 레인지 코딩이 빈도 또는 확률 정보를 이용할 수 있다면, 빈도들 대신 코드 워드들의 길이를 사용하는 것이 항상 유익하다.

[0129] 본 개시물의 실시형태의 일 예 - 인코딩 목적을 위해 채용된 코드 워드들의 길이가 있음 - 가 이제 더욱 상세히 설명될 것이지만; 빈도 정보 또는 확률 정보를 채용하는 대응 실시형태가 또한 실현 가능하다.

[0130] 인코딩된 데이터(E2, 도 2 참조), 예를 들어 인코딩된 데이터 스트림에서는, 20 개 심볼들을 포함할 수 있는 데이터 값들, 즉 0부터 19까지의 값들이 포함되어 있지만, 그것들 중 8 개의 데이터 값들, 즉 최소 값 = 2 및 최대 값 = 19이 현재 데이터 스트림에서 실제로 이용 가능하다. 이들 최소 및 최대 값들은 테이블 전달에서 더 많은 절약이 달성되는 것을 가능하게 하기 위해 나중에 설명될 바와 같이 옵션으로 또한 따로따로 전달된다. 합계 = 148, 코드 워드들의 길이가 최소 길이 = 1 및 최대 길이 = 6을 갖고, 심볼들에 대한 인덱스들 ("*indexes*")이, 예를 들어, 아래의 표 1에 기초하는 빈도들에 대응하여, 이들 심볼들을 압축하는 일 없이, 비트 스트림을 전달하기 위해  $148 * 5$  비트 = 740 비트가 요구된다고 결정될 수 있다.

[0131] 표 1: 예시적인 비트 스트림 인코딩

값	빈도	CWLen	Freq1 (비트수)	Frequency2	Freq3 (비트수)	Index1	Index2
2	7	4	4 (4)	84	128 (4)	2	0
4	2	6	1 (6)	24	32 (6)	4	1
7	81	1	32 (1)	972	1024 (1)	7	2
9	1	6	1 (6)	12	16 (7)	9	3
12	35	2	16 (2)	420	512 (2)	12	4
13	9	4	4 (4)	108	128 (4)	13	5
14	5	5	2 (5)	60	64 (5)	14	6
19	8	4	4 (4)	96	128 (4)	19	7

[0132]

[0133]

표 1에서 제시된 바와 같은 코딩 체계에 대해, 예를 들어 인코더(50)에서 그리고 디코더(60)에서 참조 인덱스에 의해 특정 가능한 또는 미리 정의된, 이용 가능한 적합한 빈도 테이블 또는 코드 테이블이 없다면, 이들 필요한 코드 워드들, 즉 길이들 및 코드들을 인코더(50)로부터 디코더(60)로 전송하기 위해 잠재적으로 이용 가능한 여러 방법들이 있다.

[0134]

첫 번째 예의 방법이 데이터의 빈도들을 수정한 다음 대응하는 코드 테이블을 생성하는 것인데, 그 방법에서는, 최소 가능성 심볼, 즉 가장 긴 코드 워드에는 하나의 추가적인 비트가 할당되고, 그 데이터에서 이용 가능하지 않은 모든 심볼들에는 코딩에 영향을 미치지 않지만 인코더(50) 및 디코더(60)가 상호 유사한 빈도 테이블 및 상호 유사한 허프만 트리를 생성하는 것을 가능하게 하는 그러한 긴 길이들의 코드 워드들이 할당된다. 12 개의 누락 심볼들이 있다는 이유로, 그러한 빈도 수정은, 예를 들어, 원래의 빈도들에 12를 곱하고 임의의 실제 빈도 값, 즉 빈도 값 = 0을 갖지 않는 모든 심볼들에 대해 1의 빈도 값을 설정함으로써 구현될 수 있다. 수정된 빈도 값들은, 예를 들어, 이용 가능한 빈도들을 갖는 모든 심볼들에 대해 표 1의 Frequency2 열로부터 알 수 있다. 이들 새로운 빈도들에 기초하여, 모든 20 개 심볼들에 대한 코드 워드들의 길이들은 다음과 같이 생성될 수 있다:

[0135]

11, 11, 4, 11, 6, 11, 11, 1, 11, 7, 11, 11, 2, 4, 5, 10, 10, 10, 10, 4

[0136]

이 종류의 코드 테이블을 사용하면, 인코더(50)로부터 디코더(60)로 코딩된 심볼들을 전달하는데 필요한 비트들이  $(7*4 + 2*6 + 81*1 + 1*7 + 35*2 + 9*4 + 5*5 + 8*4) = 291$  개이다. 올바른 데이터 인코딩 및 후속하는 데이터 디코딩을 위해, 인코더(50) 및 디코더(60)는 유사한 빈도들을 사용해야 하는데, 그러한 빈도들은 코드 워드들의 이들 새로운 길이들에 기초하여 생성될 수 있고 그 결과는 빈도를 갖는 심볼들에 대해 열  $Freq3(= 2^{(maxbitlen - bitlen)})$ 으로부터 알 수 있으며; "maxbitlen"는 "최대 비트 길이"에 대한 약어이고, "bitlen"는 "비트 길이"에 대한 약어이다. 다른 심볼들에는 frequency 2(bitlen = 10) 또는 frequency 1(bitlen = 11)이 할당된다.

[0137]

이 첫 번째 방법은 모든 가능한 심볼들, 즉 앞서 언급된 예에서의 20 개 심볼들을 포함하는 코드 테이블의 생성을 가능하게 하며; 그것은 동일한 코드 테이블이 데이터의 다른 유사한 유형들을 위해 또한 사용될 때 유익하다. 이러한 종류의 길이들은 또한 옵션으로, 임의의 부가적인 정보 없이 임의의 압축 방법으로 압축될 수 있고, 그래서 코드 테이블은 잠재적으로 모든 상황에서 인코더(50) 및 디코더(60) 간에 쉽사리 전달된다. 예를 들어, 압축 없이, 이 코드 테이블은 코드 워드들의 모든 길이를 나타내기 위해 4 개의 비트들을 요구하여  $\Rightarrow 20 \text{ 길이} * 4 \text{ 비트들} / \text{길이} = 80 \text{ 개 비트들}$ 이다.

[0138]

이 첫 번째 방법은, 최적의 허프만 코드들에 비하여, 모든 최소 가능성 심볼에 대해 단지 1 비트의, 즉 이 첫 번째 예의 방법에서 단지 1 비트의 비효율성을 갖는다. 덧붙여, 코드 워드들의 길이들, 즉 코드 테이블을 인코더(50)로부터 디코더(60)로 압축 및 전달하기 위해 필요한 그들 비트들이 또한 있다. 숫자 20, 즉 모든 가능한 심볼들의 수는, 옵션으로 전달되거나, 또는 디코더(60)에 의해 또한 알려질 수도 있다.

[0139]

다음으로, 두 번째 예의 방법이 본 발명의 대체 실시형태를 예시하기 위해 설명될 것이다. 두 번째 방법은 이용 가능한 심볼들에 대해서만, 즉 빈도 값 > 0을 갖는 그들 심볼들에 대해서만 코드 워드들의 길이들을 생성한다. 허프만 코드 테이블 생성을 위해 사용되는 인덱스가 index2(표 1 참조)이지만, 인코더(50)로부터 디코더(60)로 전달되어야 하는 인덱스는 index1, 즉 = 심볼 값이다(표 1 참조). 이렇게 생성된 코드 워드들의 길이들은 열 CWLen(표 1 참조)으로부터 알 수 있다. 이 종류의 코드 테이블을 사용하면, 코딩된 심볼들을 전달하는데



필요한 것이  $(7*4 + 2*6 + 81*1 + 1*6 + 35*2 + 9*4 + 5*5 + 8*4) = 290$  개 비트들이다. 이들 코드 워드들의 길이에 기초하여, 인코더(50) 및 디코더(60)는 표 1의 열 "Freq1"에서 보인 빈도 테이블을 생성하도록 동작 가능하다.

[0140] 이 종류의 코드 테이블에 대한 제 1 전달 방법은 코드 워드의 길이 및 코드 워드의 인덱스를 다음과 같은 숫자들의 쌍들으로서 전달하며:

[0141] (2, 4), (4, 6), (7, 1), (9, 6), (12, 2), (13, 4), (14, 5), (19, 4)

[0142] 여기서 그 쌍들은 괄호들에 의해 표시된다.

[0143] 이러한 전달 방법은 모든 인덱스마다 5 개 비트들과 코드 워드의 모든 길이마다 3 개 비트들을 요구하여 => 모든 쌍마다 8 개 비트들과 총  $8 * 8$  개 비트들 = 64 개 비트들을 요구한다.

[0144] 이들 인덱스들(즉 "*indexes*")은 또한 델타 인코딩되고 그러면 그 쌍들은 다음과 같다:

[0145] (2, 4), (2, 6), (3, 1), (2, 6), (3, 2), (1, 4), (1, 5), (5, 4)

[0146] 이제, 단지 모든 인덱스마다 3 개의 비트들과 모든 코드 워드의 길이마다 3 개의 비트들이 필요하여 => 모든 쌍마다 6 개 비트들과  $8 * 6$  개 비트들 = 48 개 비트들이 코드 테이블을 전송하기 위해 필요하다는 것이 이해될 것이다.

[0147] 코드 워드 값들의 이들 인덱스 및 길이들은 결합된 8-비트 또는 6-비트 값들과 비교하여 개선된 압축이 달성되는 것을 종종 가능하게 하는 대응하는 데이터 스트림들을 소유하도록 유익하게 분리된다. 그 스트림들은 이제 다음이며:

[0148] 2, 4, 7, 9, 12, 13, 14, 19

[0149] 및

[0150] 4, 6, 1, 6, 2, 4, 5, 4

[0151] =>  $8 * 5$  개 비트들 +  $8 * 3$  개 비트들 = 총 64 개 비트들이다.

[0152] 제 1 스트림의 인덱스들("*indexes*")이 델타 코딩될 때, 다음이 발생되며:

[0153] 2, 2, 3, 2, 3, 1, 1, 5

[0154] 및

[0155] 4, 6, 1, 6, 2, 4, 5, 4

[0156] =>  $8 * 3$  개 비트들 +  $8 * 3$  개 비트들 = 총 48 개 비트들이다.

[0157] 이는, 가능한 심볼들의 양이 많을 때이지만, 데이터가 단지 몇 개의 상호 상이한 심볼들만을 포함할 때, 최상의, 즉 최적의, 전달 방법이다.

[0158] 모든 이들 앞서 언급한 데이터 스트림들은 동작 중에 있을 때 압축되고 인코더(50)로부터 디코더(60)로 전달될 수 있다. 최적의 허프만 코드들에 비하여 이 방법에 연관된 비효율이 없지만, 본 개시물의 이 방법은 코드 워드들의 인덱스들(즉 "*indexes*") 및 길이들의 정보를 포함하는 이들 스트림들을 전달하기 위해 상당한 비트들을 여전히 소비한다. 값 8, 즉 이용 가능한 심볼들의 수는 또한 전달되어야 하는데, 그렇지 않으면 얼마나 많은 값들 또는 쌍들이 코드 테이블에 대해 코딩되어야 하는지를 디코더(60)가 확인할 수 없기 때문이다. 이 경우, 숫자 20, 즉 모든 가능한 심볼들의 수는, 인코더(50) 및 디코더(60) 간에 전달될 필요가 없다.

[0159] 앞서 설명된 방법들의 일부는, 인코더(50)에서 입력 데이터(D1)의 특히 유리한 인코딩을 달성하기 위한 조합에서는, 디코더(60)에서의 그것의 역과 마찬가지로, 대응하는 인코딩된 데이터(E2)를 생성하기 위해 채용될 가능성이 있다. 옵션으로, 모든 이들 해법들은, 예를 들어 두 번째 방법과 유사한 방식으로, 이용 가능한 심볼들에 대해서만 허프만 코드들을 생성한다. 더구나, 이들 데이터 길이들로부터 생성된 빈도 테이블은 열 Freq1(표 1 참조)에서 알 수 있다.

[0160] 첫 번째 방법과 비교하여, 두 번째 방법은 코드 워드의 길이를 그것이 이용 가능하지 않을 때 영으로 설정하는 것이 또한 가능하지만, 이러한 정보를 인코더(50)로부터 디코더(60)로 전달하는 것이 여전히 필요하며; 이는 코드 워드의 실제 길이가 결코 영일 수 없다는 점 때문에 구현하는 것이 가능하다. 이는 다음 길이들의 스트림을

발생시키며:

[0161] 0, 0, 4, 0, 6, 0, 0, 1, 0, 6, 0, 0, 2, 4, 5, 0, 0, 0, 0, 4

[0162] 이는 코드 워드들의 모든 길이에 대해 원래의 3 개의 비트들만을 요구하여 => 20 \* 3 개 비트들 = 60 개 비트들만을 요구한다.

[0163] 이들 유형들의 길이들은 임의의 부가적인 정보 없이 임의의 압축 방법으로 또한 압축될 수 있고 그래서 코드 테이블은 모든 상황들에서 인코더(50) 및 디코더(60) 간에 전달되는 것이 용이하다. 더구나, 이 데이터는 또한 수많은 영의 값들을 종종 포함하고 그래서, 예를 들어 VLC 또는 RLE를 사용함으로써 쉽게 압축될 수 있다. 숫자 20, 즉 모든 가능한 심볼들의 수가, 옵션으로 인코더(50)로부터 디코더(60)로 전달되거나, 또는 디코더(60)에 의해 선형적으로 알려진다.

[0164] 다른 실시형태가 어떤 데이터 심볼들이 이용 가능한지와 어떤 것이 그렇지 않은지를 특정하기 위해 비트들을 사용한다. 다음과 같이 제 1 스트림이 코드 워드들의 비트들을 포함하고 제 2 스트림이 코드 워드들의 길이들을 포함하는 생성된 두 개의 데이터 스트림들이 있는 상황에서:

[0165] 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1

[0166] 및

[0167] 4, 6, 1, 6, 2, 4, 5, 4

[0168] 모든 심볼에 대해 1 비트가 그리고 이용 가능한 코드 워드의 모든 길이에 대해 3 개 비트들이 요구되고 있어 => 20 \* 1 개 비트들 + 8 \* 3 개 비트들 = 총 44 개 비트들이 요구되고 있다. 이는 보통 최상의, 즉 최적의, 전달 방법이다.

[0169] 이러한 스트림들은 엔트로피 수정자(entropy modifier, EM) 및/또는 VLC를 이용함으로써 추가로 압축 가능하다. 앞서 언급한 바와 같이, 숫자 20, 즉 모든 가능한 심볼들의 수는, 옵션으로 인코더(50) 및 디코더(60)로부터 전달되거나, 또는 디코더(60)에게 선형적으로 공급된다. 더구나, 숫자 8, 즉 이용 가능한 심볼들의 수는, 인코더(50) 및 디코더(60) 간에 전달될 필요가 없는데, 그 수가 값 1을 갖는 비트들로부터 계산될 수 있기 때문이다.

[0170] 본 개시물에 의하는 모든 이전의 설명된 방법들의 경우, 인코더(50) 및 디코더(60)는 얼마나 많은 상이한 심볼들이 사용될 수 있는지, 예를 들어 전술한 예에서는 20 개 심볼들이 사용될 수 있다는 것, 또는 얼마나 많은 상이한 심볼들이 이용 가능한지, 예를 들어 전술한 예에서는 8 개가 이용 가능하다는 것에 관한 정보를 보유하고 있어야 한다. 그 값, 즉 상이한 심볼들의 수가 디코더(60)에서 이용 가능하지 않다면, 그 값은 디코더(60)로 전달되어야 한다. 옵션으로, 얼마간의 데이터는 데이터 값들이 이용 가능한 범위를 특정하는 적은 양들의 부가적인 정보를 전송함으로써 전달되는 테이블에 저장될 수 있고, 예를 들어, 값들이 포함될 범위를 특정하기 위해 값 2(최소) 및 값 19(최대)를 전달하는 것이 실현 가능하다. 이 예에서, 이러한 전달은 절약하는 것보다는 더 많은 비트들을 종종 사용하지만, 예를 들어, 8-비트 화소들이 60부터 200까지의 값들만을 포함하는 상황에서, 인코더(50)로부터 디코더(60)로 통신될 비트들에서의 절약은 현저하다. 이러한 범위의 전달은 그렇지 않으면 최소 값보다 더 작은 것과 최대 값보다 더 큰 것을 위해 사용되는 모든 비트들 또는 값들이 인코더(50)로부터 디코더(60)로 전달되는 것이 필요하지 않는 것을 가능하게 한다. 더구나, 그 범위가 인코더(50)로부터 디코더(60)로 전달될 때, 인덱스 값들이 그렇지 않으면 델타 인코딩으로 또는 델타 인코딩 없이 전송되었을 상황에서, 첫 번째 인덱스 값 및 마지막 인덱스 값을 전달할 필요가 없다는 것이 이해될 것이다. 동일한 것이 마지막 예에서의 첫 번째 및 마지막 값 1 비트들에 대해 또한 적용된다. 최소 및 최대 값들의 전달은, 다른 방법들, 예를 들어 ODelta 코딩, Gurulogic Microsystems Oy에 의해 2013년 3월 1일자로 출원된 특허 출원 GB1303661.1호에서 개시되며 참조로 본원에 통합되는 방법을 채용할 때, 옵션으로 또한 이용된다. 엔트로피를 수정하고 엔트로피 코딩을 구현하는 모든 방법들이 동일한 정보를 사용하고 그 정보가 한번만 전달될 때, 최소 및 최대 값들을 전달하는 최상의 장점이 달성된다.

[0171] 앞서 설명된 바와 같은 방법들은, 예를 들어 얼마나 많은 심볼들이 주어진 데이터 청크에서 인코딩되는 것인지에, 예를 들어 인코더(50)로부터 디코더(60)로 전송될 전체 덩어리의 데이터(total body of data)로부터 분리되는지에 응답하여 옵션으로 선택적으로 채용된다. 따라서, 이전에 밝힌 방법들의 선택은, 얼마나 많은 상이한 심볼들이 이용 가능한지, 그 심볼들 중에서 얼마나 많이 실제로 사용되는지, 최소 가능성 심볼의 빈도가 무엇인지, 그리고 이용 가능한 심볼들의 인덱스들(즉 "indexes")이 가능한 심볼들을 통해 분산되는 방법에 의존한다.

[0172] 스케일링된 확률들은 빈도 값들에 기초하여 표 1에서 도시된 심볼들에 대해 또한 계산될 수 있다. 심볼들의 수

는, 예를 들어, 148이다. 이 예에서 스케일링된 확률들은 두 개의 상이한 확률 승수들, 즉 256 및 32를 사용함으로써 유익하게 계산된다. 첫 번째 심볼에 대해 확률 승수 256을 사용하면, 그로부터 스케일링된 확률 값이  $\text{Round}(256 * 7/148) = 12$ 로서 계산되는데, "Round"는 정수 올림(rounding-up) 함수이다. 승수 256을 이용한 모든 계산된 스케일링된 확률 값들은 따라서 다음과 같다:

[0173] 12, 3, 140, 2, 61, 16, 9, 14

[0174] 스케일링된 확률 값들의 합은 257이며, 즉 > 256이고, 어떤 값을 1만큼 감소시키는 것이 유익하다. 이러한 감소는 실제 코딩에 대해 가능한 한 적은 효과를 초래하기 위하여 유익하게 구현된다. 예를 들어, 이 경우, 값 2, 즉 최소 값은, 값 1로 감소될 수 있거나, 또는 값 9, 즉 최소 올림된 값은 대응하여 값 8로 감소될 수 있어서, 레인지 코딩을 위해 또는 산술 코딩을 위해 승수 256을 이용한 스케일링된 확률 값들은 다음과 같다:

[0175] 12, 3, 140, 1, 61, 16, 9, 14(합계 = 256 = 확률 승수).

[0176] 스케일링된 확률들(승수 256을 이용함)의 전달은 다음과 같이 이루어질 수 있다: 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1

[0177] 및

[0178] 12, 3, 140, 1, 61, 16, 9, 14

[0179] 확률 승수 값이 32일 때, 스케일링된 확률 값들은 다음과 같고:

[0180] 2, 0, 18, 0, 8, 2, 1, 2

[0181] 합의 등화 후에는 다음과 같다:

[0182] 1, 0, 18, 0, 8, 2, 1, 2

[0183] 이제, 영의 값이 스케일링된 확률 값들의 일부에 대해 계산된다는 것이 이해될 것이다. 이는 그들 값들이, 예를 들어, 이스케이프 심볼을 사용함으로써 전달되어야만 한다는 것을 의미한다. 이스케이프 심볼에 대한 스케일링된 확률은 계산될 필요가 있고, 값 1보다 작지 않을 수도 있다. 이 경우, 그것에는 값 1이 할당되는데,  $\text{Round}(32 * (2 + 1) / 148) = 1$ 이기 때문이다. 이제, 이 이스케이프 심볼은 다른 심볼들에 추가되어야 하고 심볼들의 새로운 세트는 그러면 다음이 된다: "이스케이프", 2, 7, 12, 13, 14, 및 19. 그들 새로운 심볼들에는 0부터 6까지의 범위에서의 인덱스들이 유익하게 할당된다. 새로운 심볼들에 대한 스케일링된 확률들은, 이스케이프 심볼의 확률이 하나 이상의 다른 심볼들로부터 감소될 때, 레인지 코딩 또는 산술 코딩에 대해 다음과 같다:

[0184] 1 및 1, 18, 8, 2, 1, 1(합계 = 32 = 확률 승수).

[0185] 스케일링된 확률들(승수 32 및 이스케이프 코드를 이용함)의 전달은 다음과 같이 이루어질 수 있다:

[0186] 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1

[0187] 및

[0188] 1, 1, 18, 8, 2, 1, 1

[0189] 첫 번째 심볼은 이스케이프 심볼이고 비트들은 다른 심볼 값들을 특정한다는 것이 이해될 것이다.

[0190] 이스케이프 심볼이 정의될 때, 이 테이블은 앞서 언급한 바와 같이, 장래에 또한 더 낮게 사용될 수 있다. 이제, 이 데이터에 존재하지 않았던 심볼들 또한, 그 심볼들이 장래의 데이터에 존재할 것이라면, 이스케이프 심볼을 사용하여 전달될 수 있다.

[0191] 레인지 인코딩과 함께 이스케이프 코드들의 이용은, 본원의 출원인인 Gurulogic Microsystems Oy에 의해 출원 번호 GB1403038.1호로 2014년 2월 20일자로 명칭 "Encoder, decoder and method(인코더, 디코더 및 방법)"으로 출원되고 본원에 참조로 통합된 다른 특허 출원에서 제시된다.

[0192] 위의 표 1의 설명에서 언급했듯이, 데이터(D1)가 인코더(50)에서 인코딩되고, 디코더(60)에서 후속하여 디코딩될 때, 미리 정의된 테이블 또는 전달되지 않고 인덱스에 의해 기술되는 테이블을 사용하는 것이 옵션으로 또한 가능하다. 이는, 사용된 코드, 빈도, 확률 또는 코드 길이 테이블이 인코더(50)에 의해 그리고 디코더(60)에 의해 미리 알려지거나, 또는 테이블이 대안적 테이블들의 제한된 세트로부터 선택되고, 인코더(50)가 그 선택을

디코더(60)로 전달한다는 것을 의미한다. 미리 정의된 테이블은 옵션으로 디코더(60)에게는 국부적으로 이용 가능하다.

[0193] 그 테이블은 전달된 파라미터, 예를 들어 테이블의 인덱스 및/또는 테이블의 최대 인덱스에 기초하여 미리 저장될 수 있다. 대안으로, 테이블은 초기화 기능에서 구현된 알고리즘으로 또는 이전에 저장된 테이블에서의 알고리즘으로 생성될 수 있으며; 테이블의 이러한 작성, 즉 생성은 전달된 파라미터들에, 예를 들어 테이블의 인덱스에 그리고/또는 테이블의 최대 인덱스에 그리고/또는 테이블의 최소 인덱스에 또한 기초한다. 미리 저장된 테이블 대신, 인코딩된 데이터(E2)를 생성하기 위해 데이터(D1)의 데이터 인코딩 동안 더 일찍 전달되었던 테이블을 사용하는 것이 또한 옵션으로 가능하다.

[0194] 예를 들어, VLC 코딩이 사용될 때, 통상 코드들의 길이는 상이한 테이블 인덱스들에 대해 더 일찍 저장되고, 코드 테이블은 전체 테이블을 사용함으로써 또는 테이블 값들의 부분만을 사용함으로써 그들 길이 값들에 기초하여 생성될 수 있다. 사용된 부분은 전달된 테이블 파라미터들에 기초하여 또는 테이블 인덱스에 기초하여 정의될 수 있다. 마찬가지로, 레인지 코딩이 사용될 때, 통상 확률 테이블은, 예를 들어 테이블 인덱스의 사용을 통해 전달된 테이블의 형태와, 예를 들어 테이블의 최대 인덱스로서 전달된 테이블의 길이에 기초하여 생성된다.

[0195] 현재 주어진 데이터 청크에 대해 그리고 약간 상이한 심볼 빈도들을 갖는 장래의 데이터 청크들에 대해 모든 심볼들이 매우 효율적으로 코딩되는 것을 여전히 가능하게 하면서도, 별도의 이스케이프 심볼을 사용하는 일 없이 효율적인 코드 전달을 가능하게 하는 본 개시물의 다른 예의 방법이 이제 설명될 것이다. 이 다른 방법은 모든 심볼들, 즉 이용 가능한 값들을 갖는 심볼들뿐만 아니라 이용 가능한 값들을 갖지 않는 심볼들에, 적어도 스케일링된 확률로서의 값 1이 할당될 수 있도록 구현될 수 있다. 스케일링된 확률 값이 1이면, 가용성 비트는 0과 동일할 것이고, 다른 스케일링된 확률 값들의 경우 가용성 비트는 1과 동일할 것이다. 스케일링된 확률 값은 그러면 가용성 비트가 1과 동일한 심볼들에 대해서만 전달되는 것이 필요하다. 그 방법의 세부사항들은 본 출원인 Gurulogic Microsystems Oy에 의한 특허 출원, 즉 본원에 통합된 GB1403038.1호에서 발견될 수 있지만, 위의 이전의 예에서의 테이블 전달은 다음과 같다:

[0196] 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0

[0197] 및

[0198] 10, 4(합계 = 18\*1 + 10 + 4 = 32 = 확률 승수).

[0199] 이 예는 데이터의 엔트로피를 인코딩하는 허프만 코딩으로 달성될 수 있는 것보다는 레인지 코딩으로 꽤 유사한 성능을 산출하는 이로운 해법을 이제 보여준다. 이러한 해법은 코드 확률 테이블들을 매우 효율적으로 전달할 수 있다. 다른 유형들의 데이터로, 또는 다른 유형들의 확률 승수 값들로, 이 해법은 분명히, 많은 상황들에서, 채용할 최상의, 즉 최적의, 코딩 방법이다. 그런 이유로, 그것의 코드 테이블의 전달은 여기서 더 상세히 설명된다.

[0200] 옵션으로, 인코더(50) 및 디코더(60)는 인코더(50) 및 디코더(60)에 의해 이용될 것인 모든 테이블들을 정적 방식 또는 동적으로 업데이트 가능한 방식으로, 저장할 수 있다. 그 테이블들이 저장된다면, 그것들은 자신에게 대응하는 저장된 테이블을 고유하게 식별하는 인덱스에 의하여 인코더(50)로부터 디코더(60)로 전송된 데이터에서 유일하게 식별된다. 이러한 테이블들의 인덱싱은, 그렇지 않으면 코드 테이블을 인코더(50)로부터 디코더(60)로 전달하기 위해 전송할 것이 요구된 오버헤드 데이터의 엄청난 절약을 잠재적으로 가능하게 한다.

[0201] 이전에 저장된 테이블의 이용은, 예를 들어, 다음의 예에서 이해될 수 있다. 전달된 테이블에 대해, 이전의 예에서 제시된 마지막 확률 테이블, 즉 1, 1, 1, 1, 1, 1, 1, 10, 1, 1, 1, 1, 4, 1, 1, 1, 1, 1, 1 이 재사용 목적을 위해 추가되도록 선택되며; 이러한 목적을 위해, 테이블에는, 예를 들어, "17"의 인덱스 값이 할당된다. 이제 새로운 데이터 청크가 인코딩될 것이 필요하고 그것은 표 2에서 제공된 바와 같은 심볼 값들 및 빈도들을 갖는다.

[0202] 표 2 테이블 재사용을 위한 제 2 심볼 값들 및 빈도들을 갖는 예

값	빈도
1	3
5	1
7	68
8	1
10	4
12	32
14	3
18	1

[0203]

[0204]

이제 레인지 인코딩을 위한 모든 이용 가능한 확률 테이블들이 표 2로부터 평가될 수 있고, 테이블 17이 레인지 코딩과 함께 인코더(50)로부터 디코더(60)로 이 새로운 데이터를 전달하기 위해 사용될 최상의 확률 테이블이 되도록 선택되는 것은 매우 가능성이 있다. 적어도, 새로운 확률 테이블의 전달이 테이블 17이 이상적인 엔트로피 인코딩 결과에 대해 생성하는 부가적인 데이터의 양보다 아주 많은 데이터를 요구하는 것을 쉽게 알 수 있다. 그런 이유로, 테이블 17, 또는 어떤 다른 확률 테이블이, 새로운 더 최적화된 확률 테이블을 전달하는 것을 필요로 하는 대신 재사용될 수 있다. 테이블이 재사용될 때, 그것의 인덱스 17은 인코더(50)로부터 디코더(60)로 전달되고, 그 후, 레인지 코딩 값들이 인코더(50)로부터 디코더(60)로 전달될 수 있다. 테이블이 재사용될 수 없을 때, 값, 예를 들어 0 또는 새로운 테이블 전달을 정의하는 다음의 자유 테이블 인덱스가 인코더(50)로부터 디코더(60)로 먼저 전달된 다음, 그 테이블이 인코더(50)로부터 디코더(60)로 또한 전달되고, 그 후, 레인지 코딩 값들이 인코더(50)로부터 디코더(60)로 전달될 수 있다. 종종, 코딩된 심볼들의 양은, 예를 들어 통상 레인지 인코딩된 데이터 전에 전달될 필요가 또한 있으며, 이는 디코더(60)에서 데이터의 적절한 디코딩을 가능하게 한다.

[0205]

옵션으로, 이미 사용 중인 코드 테이블을 수정하고 그 안에서의 따라서 새로운 코드 테이블에 대한 변경들만을 전송하는 것이 또한 가능하다. 추가의 옵션으로, 이미 전달된 코드 테이블은 전달/수신 후 적응적 방식으로 사용될 수 있다.

[0206]

옵션으로, 인코더(50) 및 디코더(60)는, 현재 누락 심볼들이 존재할 수도 있을 경우, 유사한 유형들의 데이터의 추가의 사용을 위해 다른 심볼들의 코딩을 또한 가능하게 하는 유사한 전체 테이블들을 생성하도록 동작 가능하다. 이들 테이블들에는 새로운 코드 테이블 인덱스가 저장 및 제공될 수 있다. 양쪽 모두의 유형들의 테이블들, 즉 충족(fulfill)이 없는 하나와 및 충족이 있는 다른 하나를 저장하는 것이 가능하다. 더구나, 원래의 테이블만을 저장하는 것이 또한 가능하고, 충족 속성을 갖는 것이 다음 번에 필요할 때, 그것은 통신 인코더(50)로부터 디코더(60)로 따로따로 나타내어질 수 있다. 충족된 테이블들을 저장하는 해법이 더욱 바람직한데, 그것이 의사결정을 단순화하고 필요한 테이블이 충족되는지의 여부를 나타내는 임의의 부가적인 표시가 전달되는 것을 요구하지 않기 때문이다. 테이블의 충족은 모든 값들을 포함하는 테이블이, 예를 들어, 4에서 1로 감소하는 빈도들로 채워져서 다음의 심볼들에는 비교적 더 짧은 심볼들이 할당되고 마지막 심볼들에는 상대적으로 더 긴 심볼들이 할당되도록 또한 구현될 수 있다. 이러한 접근법을 채용함으로써, 심볼들의 순서는 데이터의 장래의 스트림에서의 이용 가능한 심볼들의 순서에 대응한다.

[0207]

앞서 설명된 예들에서, 코드 워드들의 길이들의 전달 및 스케일링된 확률들의 전달은 유익하게 채용되지만, 유사한 기법들은 허프만 코딩, 산술 코딩 또는 레인지 코딩을 위해 필요한 빈도 값들을 전달하기 위해 또한 사용될 수 있다. 유익하게는, 채용할 최상의 코딩 방법이 인코딩된 데이터에 코드 테이블, 코드 워드들의 길이, 확률 테이블 또는 빈도 테이블을 전달하는데 필요한 오버헤드 정보가 추가될 때 최소 비트들을 사용함으로써, 예를 들어, 데이터의 더 작은 부분들, 즉 데이터 청크들에 포함된 콘텐츠의 성질 및/또는 데이터 유형에 대해 특히 최적화된 인코딩 방법들을 사용하여 데이터의 더 작은 부분들이 인코더(50)로부터 디코더(60)로 전송되는 것을 가능하게 한다. 이런 이유로, 산술 코딩 및 레인지 코딩을 이용한 최상의 결과는 빈도 값들이 적어도 제한된 정도로 양자화될 때 달성될 수 있어서, 빈도 테이블이 정확한(exact) 테이블보다는 분명히 적은 비트량으로 전달될 수 있는 거의 정확한 값들을 제시하고, 그래서, 예를 들어 엔트로피 코딩을 채용할 때, 최적성(optimality)은 심볼들의 코딩으로부터 약간만 감소된다. 더구나, 스케일링된 확률 테이블들의 전달은 매우 효율적이고 거의 최적의 레인지 코딩 및 산술 코딩 구현예들이 실현되는 것을 가능하게 한다.

[0208]

인코더(50)로부터 디코더(60)로 통신될 데이터의 작은 양만이 있는 경우, 임의의 형태의 인코딩을 실질적으로

채용하는 일 없이 데이터를 통신하는 것이 통상 더 좋다. 그러나, 데이터의 양이 증가할 때, 대략 정확한 길이들의 코드 워드들을 갖는 허프만 코딩을 채용하는 것이 유익하다. 인코더(50)로부터 디코더(60)로 통신될 데이터의 볼륨이 증가함에 따라, 점진적으로 더 많은 이점이 더욱 정확한 코드 테이블들을 사용하는 것으로부터 도출된다. 더구나, 산술 코딩될 또는 레인지 코딩될 상당한 양의 데이터가 있을 경우, 최상의 코딩 결과들이 더욱 효율적인 엔트로피 코딩이 채용될 때 획득되며; 따라서 코드 테이블을 전달하기 위해 필요한 비트들에 비하여 빈도 또는 확률을 전달하기 위해 필요한 비트들을 통한 코딩 장점들이 획득된다. 확률 테이블의 인덱스, 빈도 테이블, 코드 워드 길이 테이블들, 또는 코드 테이블을 전달하는 것은 항상 유사하고, 인덱스가 사용되는 경우, 이용 가능한 최상의 테이블을 갖는 방법은 최상의 압축 성능이 달성되는 것을 가능하게 한다. 사용되는 코딩 방법의 선택이 데이터에 기초하여 또는 데이터의 양에 기초하여 고정되지 않는다면, 사용된 코딩 방법의 선택이 인코더(50)로부터 디코더(60)로 또한 전달되어야 한다.

[0209] 인코더(50)와 디코더(60)는 전체적으로 코덱(100)을 형성한다. 실제 상황에서, 예를 들어 인코더(50)가, 예를 들어 무선, 광섬유 통신 네트워크 등을 통해, 수많은 디코더들(60)에게 광범위하게 브로드캐스팅, 즉 "멀티캐스팅"되는 인코딩된 데이터(E2)를 생성하는 상황에서, 하나의 인코더(50) 및 하나 이상의 디코더들(60)이 있을 수도 있다. 더구나, 예를 들어 피어 투 피어 데이터 통신 네트워크에서, 디코더(60)가 피어 투 피어 네트워크 내에서 커플링된 여러 인코더들(50)로부터 자신의 인코딩된 데이터(E2)를 공급받는 상황들이 발생할 수 있으며, 이 상황에서 인코딩된 데이터(E2)는 디코더(60)에서 함께 수집된 따로따로 인코딩된 데이터 청크들로 공급되며; 이러한 구성은 인코딩된 데이터(E2)의 특정한 부분들이 그러한 피어 투 피어 네트워크들을 구현하기 위해 채용된 장거리 데이터 통신 네트워크 링크들 상의 데이터 로드를 용이하게 하는 디코더(60)에게 더욱 국부적으로 공급되기 때문에 유익하다. 인코더(50) 및 디코더(60)는 전용 디지털 하드웨어에서, 비-일시적 데이터 저장 매체들 상에 기록된 하나 이상의 소프트웨어 제품들을 실행하도록 동작 가능한 컴퓨팅 하드웨어에서, 또는 그것들의 조합에서 구현되는 것이 가능하다. 인코더(50) 및 디코더(60)는 오디오 레코딩 및/또는 플레이백 장치에서, 비디오 레코딩 및/또는 플레이백 장치에서, 개인용 컴퓨터들에서, 스마트 폰들에서, 디지털 카메라들에서, 비디오 카메라들에서, 텔레비전들에서, 인터넷 단말들에서, 과학적 장치에서, 감시 및/또는 보안 시스템들에서, 지면 모니터링 기능들을 위해 구성된 위성들에서, 지진 감지 시스템들에서 사용 가능하지만, 그것들로 제한되지는 않는다.

[0210] 본 개시물의 실시형태들은 테이블들, 예를 들어 코딩 테이블들, 빈도 테이블들, 확률 테이블들 또는 코드 워드들의 길이들의 더욱 효율적인 전달을 가능하게 할 수 있음으로써, 예를 들어 최적의 방식으로 개별적으로 인코딩될 수 있는 데이터의 더 작은 청크들로 데이터를 분리하는 것을 매력적이 되게 한다. 더구나, 그 테이블들은 옵션으로는 엔트로피 인코딩 방법을 사용하여 인코딩된다. 이들 더 적은 데이터 청크들은 그들 소유의 대응하는 코드 테이블, 빈도 테이블, 확률 테이블 또는 코드 워드들의 길이들을 옵션으로 필요로 하며; 이는 이용 가능한 상이한 테이블들이 있다면 유익한데, 주어진 테이블들의 인덱스들만이 상이한 데이터 청크들에 대해 전달될 필요가 있어서이다. 그렇지 않으면, 새로운 테이블은 디코더(60)로 전달되는 것이 또한 필요하다. 주어진 테이블이 전달될 때, 그 테이블을 그것 소유의 고유 참조 인덱스와 함께하는 장래의 사용을 위해 데이터 메모리, 예를 들어 디코더(60)의 데이터 메모리에 저장하는 것이 종종 유익하다.

[0211] 본 개시물의 일 예의 실시형태에서, 모든 데이터 블록이 개개의 데이터 블록으로서 전달되며, 얼마나 많은 데이터 블록들이 상호 유사한 데이터 덩어리에 속하는 지를 기술하는 정보가 전송되는 것을 지원하며; 이러한 데이터 블록들의 통신이 종종 꽤 비효율적인데, 모든 데이터 블록들에 대해, 채용된 코딩 방법의 식별 ID(identification), 심볼들의 양, 및 채용되는 코드 테이블, 빈도 테이블 또는 확률 테이블을 전달할 필요가 있기 때문이다. 덧붙여, 상호 유사한 데이터 덩어리에 속한 데이터 블록들의 수는 인코더(50)로부터 디코더(60)로 전달되는 것이 또한 필요하다.

[0212] 옵션으로, 코드 테이블에서는, 자신 소유의 대응하는 의미를 갖는 하나 이상의 부가적인 심볼들의 삽입이 유익하게 채용된다. 보통, 큰 코드 테이블들에서, "이스케이프" 심볼이 유익하게는 자신 소유의 코드 워드를 갖는다. 더구나, JPEG에서 DCT 계수들을 위해 사용되는 코딩 테이블에서의 "계수들의 말단" 심볼에 대해 이용 가능한 소유 코드 워드가 또한 유익하다. 이는 그 방법이 디코더(60)에게 이미 알려져 있어서, 그 방법은, 예를 들어, "코드 테이블의 변경", "데이터의 말단"에 대해 그리고 또한 "이스케이프"에 대해 이용될 수 있는 새로운 코딩 심볼들을 필요하다면 추가함으로써 매우 효율적인 방식으로 이용될 수 있다는 것을 의미한다. 이들 부가적인 심볼들은 그 심볼이 사용될 때마다 그 심볼들의 빈도가 1이도록 생성될 수 있다. 이용 가능한 테이블이 사용된다면, 대응하는 식별ID는 심볼로서 추가되어서 심볼들 중 가장 긴 코드 워드를 갖는 하나의 심볼로 코드를 분리한다. 남아 있는 데이터가 있다면, 예를 들어 현재 데이터 청크 후에 새로운 데이터 청크가 있다면, 인

코더(50)는 "코딩 테이블의 변경" 심볼을 유익하게 사용하지만 "데이터의 말단" 심볼은 그렇지 않다. 이 "코딩 테이블의 변경" 심볼이 전달될 때, 새로운 코딩 테이블의 인덱스가 그 후에 전달된다. 새로운 코딩 테이블을 정의하는 인덱스의 값은, 예를 들어, 이용 가능한 테이블이 없을 때 채용되고 이용 가능한 테이블들이 이미 있는 경우 1부터 테이블들의 수량까지가 채용된다. 옵션으로, 새로운 코딩 테이블에 대한 이 인덱스는 데이터에 대해 모든 이용 가능한 또는 적합한 테이블들을 제시하기 위해 요구되는 만큼 많은 비트들을 사용하는 값을 갖는다. 값 0이 새로운 코딩 테이블에 대한 인덱스로서 전달된다면, 코드 테이블의 전달은 다음의 심볼들이 인코더(50)로부터 디코더(60)로 제공되는 데이터 스트림으로 인코딩되기 전에 필요하다. 그렇지 않으면, 새로운 코딩 테이블을 식별하는 인덱스 뒤에, 새로운 심볼들은 이 새로운 코딩 테이블 내에서 즉시 인코딩될 수 있다. 마지막 데이터 청크가 코딩되고 마지막 데이터 값이 디코더(60)로 전달될 때, 인코더(50)는 "데이터의 말단" 심볼을 전달한다. 이 경우, "데이터의 말단" 심볼만이 유효하고 "코딩 테이블의 변경"은 사용되지 않는다. "데이터의 말단" 심볼이 전달될 때, 임의의 데이터의 전달을 그 후에 계속할 필요는 없다. 이 "데이터의 말단" 심볼은 각각의 데이터 청크에 대한 데이터 값들의 수를 전달할 필요가 없게 하는 것을 가능하게 한다. 더구나, 코딩 방법이 디코더(60)로 전달될 필요가 없는데, 사용된 코드 테이블, 빈도 테이블, 확률 테이블 또는 코드 워드들의 길이들만이, 상이한 데이터 청크들에 대해 변경되기 때문이다. 따라서, 인코더(50)로부터 디코더(60)로 전송될 오버헤드 데이터의 총량은 코드 테이블이 데이터의 인코딩 및 후속하는 디코딩 동안 변경될 때 매우 적다. 심볼이 "코드 테이블의 변경"에 또는 "데이터의 말단"에 관계가 있거나, 또는 코드 테이블에 대해 빈도 1을 갖는 심볼들 양쪽 모두를 생성하는 것이 가능하다면, 하나의 트레일링 비트가 검출될 필요가 있다.

[0213] 가끔, 데이터 값들의 양, 인코딩된 데이터의 양을 전송하거나 또는 디코더(60) 및 인코더(50)의 코딩 방법 및 구현에서 사용된 데이터인, 데이터의 양에 의존하여 "데이터의 말단" 심볼을 사용하는 것이 유익하다. 더구나, 인코더(50) 및/또는 디코더(60)에서 데이터를 프로세싱할 때 병렬처리를 채용하는 것이 옵션으로 유익하며, 즉 코딩된 데이터의 양이 전달되고 그래서 디코더(60)는 상이한 프로세서들, 프로세스들 및 스레드들에 대해 데이터를 쉽게 분리할 수 있다. 통상, 얼마나 많은 데이터 값들이 디코딩될 것이 필요한 지에 관한 정보를 전달하는 것이 가장 적합한 다양한 접근법들이 종종 있고, 이런 경우, 대응하는 선택을 전달할 필요는 없지만; 다수의 최상의 선택들이 이용 가능할 때, 인코더(50)는 그 방법을 선택하고 가장 적합한 선택에 관한 대응하는 결정을 디코더(60)에게 전달한다.

[0214] 예를 들어 도 3a에 도시된 바와 같이, 데이터(D1) 및 데이터(D3)가 상호 실질적으로 유사한 것이 될 때, 디코더(60)는 인코더(50)에서 실행되는 인코딩 기능들의 역을 실질적으로 구현한다는 것이 전술한 바로부터 이해될 것이다. 그러나, 많은 실제 상황들은, 예를 들어 인코딩된 데이터(E2)를 복수의 상호 상이한 디바이스들로 멀티캐스팅할 때, 도 3b에 도시된 바와 같이, 데이터(D3)를 트랜스코딩하여 디코더(60) 및 그것의 연관된 트랜스코더(70)를 호스팅하는 주어진 디바이스와 호환 가능한 대응하는 트랜스코딩된 데이터(D4)를 생성하기 위해 트랜스코더(70)의 사용을 요구한다. 옵션으로, 디코더(60) 및 트랜스코더(70) 둘 다는 컴퓨팅 하드웨어를 사용하여 구현되며; 옵션으로, 트랜스코더(70)는 전용 트랜스코딩 하드웨어, 예를 들어 하드웨어 동글(dongle) 또는 유사물로 구현된다. 본 개시물의 실시형태들은 데이터의 무손실 또는 손실 인코딩 및 디코딩을 제공하도록 구성될 가능성이 있다. 옵션으로, 디코더(60)는 트랜스코딩을 수행하여, 예를 들어 데이터(D1)를 랜더링하도록 요구된 것과는 상이한 디스플레이 디바이스에 데이터를 제공하도록 또한 동작 가능하다. 이러한 경우에, 코덱(100)을 통해 프로세싱된 데이터는 결코 그것의 원본 포맷으로 역으로 디코딩되지 않는다. 대신, 인코딩된 데이터(E2)는, 예를 들어 스크린에 랜더링되거나 또는 파일 속에 저장될 어떤 다른 포맷으로, 예를 들어 직접적으로 변환된다. 이러한 트랜스코딩의 일 예의 실시형태가 데이터(D1)가 원래 YUV 포맷이었고, 그 데이터는 압축되고 수신기로 송신되며; 수신기가 그 데이터를 블록 단위로 복원하고, 그것에 대해 컬러 변환을 수행하고 그것을, 심지어 풀 해상도 YUV 결과 이미지를 복원하는 일 없이, 스크린 상에 랜더링하기에 적합한 RGB 이미지로 스케일링하는 경우이다.

[0215] 디코더(60)는 인코더(50)에 의해 생성된 인코딩된 데이터(E2)를 디코딩하여, 대응하는 디코딩된 데이터(D3)를 생성하는 방법을 채용하도록 동작 가능한데, 그 방법은 다음의 단계들을 포함한다:

[0216] (i) 인코딩된 데이터(E2)를 수신하고 수신된 인코딩된 데이터로부터, 하나 이상의 빈도 테이블들, 및/또는 하나 이상의 코드 테이블들, 및/또는 하나 이상의 코드 워드 길이 테이블들, 및/또는 하나 이상의 확률 테이블들, 및/또는 이러한 하나 이상의 테이블들을 나타내는 정보와 함께, 하나 이상의 인덱스 세트들을 추출하는 단계;

[0217] (ii) 하나 이상의 데이터 청크들에서의 대응하는 심볼들 및/또는 하나 이상의 코드 테이블들에서의 엔트리들의 압축된 심볼들, 및/또는 하나 이상의 빈도 테이블들, 및/또는 하나 이상의 코드 워드 길이 테이블들, 및/또는

하나 이상의 확률 테이블들을 하나 이상의 인덱스 세트들로부터 컴퓨팅하는 단계; 및

- [0218] (iii) 하나 이상의 코드 테이블들, 및/또는 하나 이상의 빈도 테이블들, 및/또는 하나 이상의 코드 워드 길이 테이블들, 및/또는 하나 이상의 확률 테이블들로부터의 정보를 사용하여, 하나 이상의 데이터 청크들을 심볼들로부터 재생성하는 단계; 및
- [0219] (iv) 디코딩된 데이터(D3)를 생성하기 위해 하나 이상의 데이터 청크들을 결합 및/또는 변환하는 단계.
- [0220] 옵션으로, 그 방법은, 하나 이상의 테이블들 중 적어도 하나의 테이블이 후속하는 재사용을 위해 저장 가능한 방식으로 하나 이상의 테이블들 중 적어도 하나의 테이블을 수신하는 단계를 포함한다.
- [0221] 옵션으로, 그 방법은, 디코딩된 데이터(D3)를 생성하기 위해 단계 (iv)에서 하나 이상의 데이터 압축해제 알고리즘들을 적용하는 단계를 포함한다. 더한 옵션으로, 그 방법에서, 하나 이상의 데이터 압축해제 알고리즘들은 다음 중 적어도 하나를 포함한다: 허프만 디코딩, VLC 디코딩, 엔트로피 디코딩, 산술 디코딩, 레인지 디코딩.
- [0222] 옵션으로, 그 방법은, 복수의 데이터 청크들을 실질적으로 병행하는 방식으로 프로세싱하는 프로세서들의 병렬 아키텍처를 채용함으로써, 디코딩된 데이터(D3)를 생성하기 위해 복수의 하나 이상의 데이터 청크들을 결합하는 단계를 포함한다.
- [0223] 옵션으로, 그 방법은, 함께 결합되는 다수의 데이터 값들에 기초하여 하나 이상의 인덱스 세트들을 생성하는 단계를 포함한다. 더한 옵션으로, 그 방법에서, 인덱스들은 R, G 및 B 화소 값들 또는 Y, U, 및 V 화소 값들을 포함하는 하나 이상의 RGB-화소들로부터 도출된다. 더한 옵션으로, 그 방법은, 인코딩된 데이터(E2)에 포함될 때 데이터 청크들에 대한 달성가능 데이터 압축해제 비율의 함수로서, 인코딩된 데이터(E2)가 되게 하나 이상의 데이터 청크들을 인코딩식으로 생성하는 것 또는 비인코딩식으로 생성하는 것 간을 동적으로 스위칭하는 단계를 포함한다.
- [0224] 옵션으로, 그 방법에서, 디코더는 심볼이 "*코드 테이블의 변경*"에 관련되는지 또는 "*데이터의 말단*"에 관련되는지를 나타내는 적어도 하나의 트레일링 비트를 인코딩된 데이터(E2)로부터 추출하도록 동작 가능하다.
- [0225] 옵션으로, 그 방법은, 주어진 데이터 청크에 존재하는 하나 이상의 심볼들을 참조하기 위해 요구된 실질적으로 그 인덱스들만으로부터 주어진 데이터 청크를 생성하는 단계를 포함한다.
- [0226] 옵션으로, 그 방법은 인코딩된 데이터(E2)에 포함된 하나 이상의 코드 테이블들을 압축해제하는 단계를 포함한다. 더한 옵션으로, 그 방법은, 허프만 디코딩을 채용함으로써 하나 이상의 코드 테이블들을 압축해제하는 단계를 포함한다. 더한 옵션으로, 그 방법에서, 하나 이상의 코드 테이블들의 압축해제는 하나 이상의 보조 코드 테이블들을 채용한다.
- [0227] 옵션으로, 그 방법은, 하나 이상의 코드 테이블들이 후속하여 전송된 데이터를 디코딩하기 위해 디코더(60)에서 채용되는 것을 가능하게 하는 방식으로 하나 이상의 코드 테이블들을 수신하는 단계를 포함한다.
- [0228] 옵션으로, 그 방법은, 하나 이상의 데이터베이스들, 및/또는 하나 이상의 프록시 데이터베이스들을 통해, 하나 이상의 코드 테이블이 액세스될 가능성이 있는 경우를 나타내는 하나 이상의 식별 코드들을 인코딩된 데이터(E2)로부터 추출하는 단계를 포함한다.
- [0229] 옵션으로, 그 방법은 다음의 유형들, 즉, 캡처된 오디오 신호들, 캡처된 비디오 신호들, 캡처된 이미지들, 텍스트 데이터, 지진계 데이터, 센서 신호들, 아날로그-디지털(ADC) 변환된 데이터, 생체의학 신호 데이터, 캘린더 데이터, 경제 데이터, 수학 데이터, 이진수 데이터 중 하나 이상의 유형의 데이터를 디코딩하는 단계를 포함한다.
- [0230] 옵션으로, 그 방법은, 복수의 데이터 소스들로부터 인코딩된 데이터(E2)를 수신하는 단계와, 소스들로부터 제공된 데이터를 결합하여 인코딩된 데이터(E2)를 재생성하는 단계를 포함한다.
- [0231] 디코더(60)는 인코딩된 데이터(E2)를 디코딩하여 디코딩된 데이터(D3)를 생성하는 앞서 언급된 방법을 구현하도록 동작 가능하며; 인코더(50)에 의해 생성된 인코딩된 데이터(E2)를 디코딩하는 디코더(60)가 제공되는데, 그 디코더(60)는,
- [0232] (i) 인코딩된 데이터(E2)를 수신하고 수신된 인코딩된 데이터로부터, 하나 이상의 빈도 테이블들, 및/또는 하나 이상의 코드 테이블들, 및/또는 하나 이상의 코드 워드 길이 테이블들, 및/또는 하나 이상의 확률 테이블들, 및/또는 이러한 하나 이상의 테이블들을 나타내는 정보와 함께, 하나 이상의 인덱스 세트들을 추출하도록;



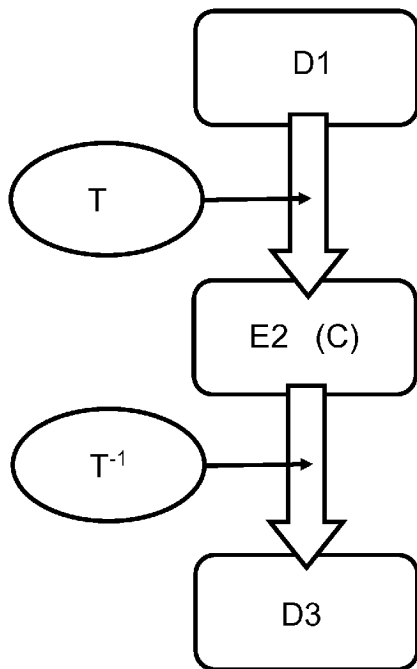
- [0233] (ii) 하나 이상의 데이터 청크들에서의 대응하는 심볼들 및/또는 하나 이상의 코드 테이블들에서의 엔트리들의 압축된 심볼들, 및/또는 하나 이상의 빈도 테이블들, 및/또는 하나 이상의 코드 워드 길이 테이블들, 및/또는 하나 이상의 확률 테이블들을 하나 이상의 인덱스 세트들로부터 컴퓨팅하도록; 및
- [0234] (iii) 하나 이상의 코드 테이블들, 및/또는 하나 이상의 빈도 테이블들, 및/또는 하나 이상의 코드 워드 길이 테이블들, 및/또는 하나 이상의 확률 테이블들로부터의 정보를 사용하여, 하나 이상의 데이터 청크들을 상기 심볼들로부터 재생성하도록; 및
- [0235] (iv) 디코딩된 데이터(D3)를 생성하기 위해 하나 이상의 데이터 청크들을 결합 및/또는 변환하도록 동작 가능하다.
- [0236] 옵션으로, 디코더(60)는, 디코딩된 데이터(D3)를 트랜스코딩하여 대응하는 트랜스코딩된 데이터(D4)를 생성하고 그리고/또는 인코딩된 데이터(E2)로부터 대응하는 트랜스코딩된 데이터(D4)를 생성하는 트랜스코더(70)를 더 포함한다.
- [0237] 옵션으로, 디코더(60)는, 하나 이상의 테이블들 중 적어도 하나의 테이블이 후속하는 재사용을 위해 저장 가능한 방식으로 하나 이상의 테이블들 중 적어도 하나의 테이블을 수신하도록 동작 가능하다.
- [0238] 옵션으로, 디코더(60)는 디코딩된 데이터(D3)를 생성하기 위해 단계 (iv)에서 하나 이상의 데이터 압축해제 알고리즘들을 적용하도록 동작 가능하다. 더한 옵션으로, 그 디코더(60)에서, 하나 이상의 데이터 압축해제 알고리즘들은, 허프만 디코딩, VLC 디코딩, 엔트로피 디코딩, 산술 디코딩, 레인지 디코딩 중 적어도 하나를 포함한다.
- [0239] 옵션으로, 디코더(60)는, 복수의 데이터 청크들을 실질적으로 병행하는 방식으로 프로세싱하는 프로세서들의 병렬 아키텍처를 채용함으로써, 디코딩된 데이터(D3)를 생성하기 위해 복수의 하나 이상의 데이터 청크들을 결합하도록 동작 가능하다.
- [0240] 옵션으로, 디코더(60)는, 함께 결합되는 다수의 데이터 값들에 기초하여 하나 이상의 인덱스 세트들을 생성하도록 동작 가능하다. 더한 옵션으로, 그 디코더(60)에서, 인덱스들은 R, G 및 B 화소 값들 또는 Y, U, 및 V 화소 값들을 포함하는 하나 이상의 RGB-화소들로부터 도출된다. 더한 옵션으로, 디코더(60)는, 인코딩된 데이터(E2)에 포함될 때 데이터 청크들에 대한 달성가능 데이터 압축해제 비율의 함수로서, 인코딩된 데이터(E2)가 되게 하나 이상의 데이터 청크들을 인코딩식으로 생성하는 것 또는 비인코딩식으로 생성하는 것 간을 동적으로 스위칭하도록 동작 가능하다.
- [0241] 옵션으로, 그 디코더(60)는 심볼이 "코드 테이블의 변경"에 관련되는지 또는 "데이터의 말단"에 관련되는지를 나타내는 적어도 하나의 트래일링 비트를 인코딩된 데이터(E2)로부터 추출하도록 동작 가능하다.
- [0242] 옵션으로, 그 디코더(60)는, 주어진 데이터 청크에 존재하는 하나 이상의 심볼들을 참조하기 위해 요구된 실질적으로 그 인덱스들만으로부터 주어진 데이터 청크를 생성하도록 동작 가능하다.
- [0243] 옵션으로, 그 디코더(60)는 인코딩된 데이터(E2)에 포함된 하나 이상의 코드 테이블들을 압축해제하도록 동작 가능하다. 더한 옵션으로, 디코더(60)는 허프만 디코딩을 채용함으로써 하나 이상의 코드 테이블들을 압축해제하도록 동작 가능하다. 더한 옵션으로, 그 디코더(60)에서, 하나 이상의 코드 테이블들의 압축해제는 하나 이상의 보조 코드 테이블들을 채용한다.
- [0244] 옵션으로, 디코더(60)는, 하나 이상의 코드 테이블들이 후속하여 전송된 데이터를 디코딩하기 위해 디코더(60)에서 채용되는 것을 가능하게 하는 방식으로 하나 이상의 코드 테이블들을 수신하도록 동작 가능하다.
- [0245] 옵션으로, 디코더(60)는, 하나 이상의 데이터베이스들, 및/또는 하나 이상의 프록시 데이터베이스들을 통해, 하나 이상의 코드 테이블이 액세스될 가능성이 있는 경우를 나타내는 하나 이상의 식별 코드들을 인코딩된 데이터(E2)로부터 추출하도록 동작 가능하다.
- [0246] 옵션으로, 디코더(60)는, 다음의 유형들, 즉, 캡처된 오디오 신호들, 캡처된 비디오 신호들, 캡처된 이미지들, 텍스트 데이터, 지진계 데이터, 센서 신호들, 아날로그-디지털(ADC) 변환된 데이터, 생체의학 신호 데이터, 캘린더 데이터, 경제 데이터, 수학 데이터, 이진수 데이터 중 하나 이상의 유형의 데이터를 디코딩하도록 동작 가능하다.
- [0247] 옵션으로, 디코더(60)는, 복수의 데이터 소스들로부터 인코딩된 데이터(E2)를 수신하도록, 그리고 소스들로부터 제공된 데이터를 결합하여 인코딩된 데이터(E2)를 재생성하도록 동작 가능하다. 예를 들어, 복수의 소스들은

인코더(50)로부터 디코더(60)로 인코딩된 데이터(E2)를 통신하는 피어 투 피어 데이터 통신 네트워크에 포함된다.

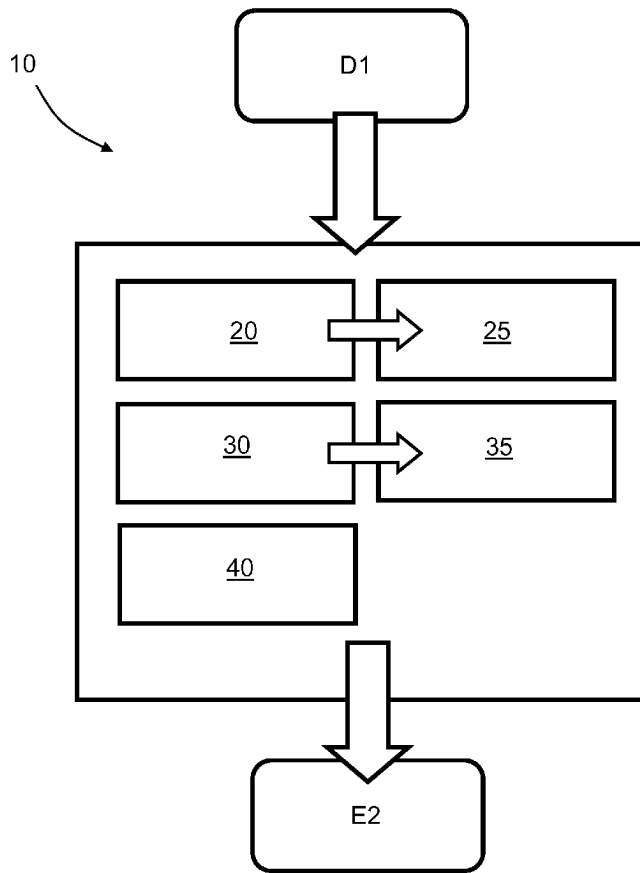
[0248] 앞서 설명된 본 발명의 실시형태들에 대한 변형예들이 첨부된 청구항들에 의해 규정된 바와 같은 발명의 범위로 부터 벗어나는 일 없이 가능하다. 본 발명을 설명하고 청구하기 위해 사용되는 "구비하는", "포함하는", "통합하는", "이루어지는", "갖는", "~인"과 같은 표현들은, 비-배타적 방식으로 즉, 명시적으로 기술 또는 제시되지 않은 아이템들, 컴포넌트들 또는 엘리먼트들을 허용하는 것으로 해석되도록 의도된다. 단수에 대한 언급은 또한 복수에 관련하는 것으로 해석된다. 첨부된 청구항들에서 괄호 속에 포함된 숫자들은 청구항의 이해를 돕기 위해 의도된 것이고 이들 청구항들에 의해 청구된 요지를 제한하는 어떠한 방법으로도 해석되지 않아야 한다.

도면

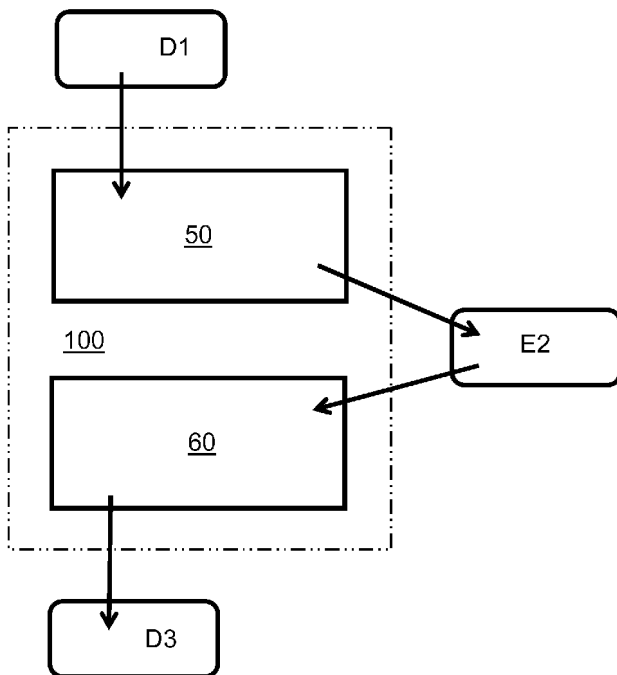
도면1



도면2



도면3a



도면3b

