



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2017년05월17일
 (11) 등록번호 10-1737158
 (24) 등록일자 2017년05월11일

- (51) 국제특허분류(Int. Cl.)
 H04N 19/42 (2014.01) H04N 19/124 (2014.01)
 H04N 19/44 (2014.01) H04N 19/46 (2014.01)
 H04N 21/236 (2011.01) H04N 21/434 (2011.01)
- (21) 출원번호 10-2013-0153317
- (22) 출원일자 2013년12월10일
 심사청구일자 2015년05월26일
- (65) 공개번호 10-2015-0053870
- (43) 공개일자 2015년05월19일
- (30) 우선권주장
 1222240.2 2012년12월11일 영국(GB)
 13005263.2 2013년11월08일
 유럽특허청(EPO)(EP)
- (56) 선행기술조사문헌
 KR1020120003909 A*
 GB2362055 A*
 JP2009111649 A*
 WO2010146239 A1
 *는 심사관에 의하여 인용된 문헌

- (73) 특허권자
 구루로직 마이크로시스템스 오이
 핀란드 투르쿠 20100 린난카투 34
- (72) 발명자
 칼레보 오씨
 핀란드 37800 토이자라 케툰헨테 1
 켈케이넨 투오마스
 핀란드 20230 툴쿠 라우타란카투 2 비 17
- (74) 대리인
 김태홍

전체 청구항 수 : 총 33 항

심사관 : 이상래

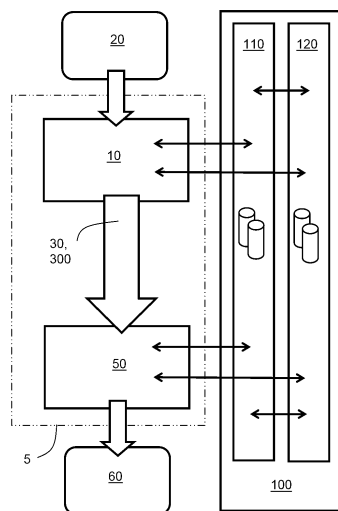
(54) 발명의 명칭 인코더, 디코더 및 방법

(57) 요약

소스 데이터(20)를 인코딩하여 전송 또는 저장을 위한 대응하는 인코딩된 데이터(30)를 발생시키는 방법이 제공되어 있고, 이 방법은

(a) 소스 데이터(20)의 하나 이상의 부분들을 하나 이상의 데이터베이스들(100) 내의 하나 이상의 요소들(E)에 (뒷면에 계속)

대표도 - 도1



정합시키고 - 여기서 하나 이상의 요소들(E)은 대응하는 하나 이상의 데이터 블록들을 나타냄 -, 소스 데이터 (20)의 하나 이상의 부분들을 하나 이상의 정합된 요소들(E)에 관련시키는 참조 값들(R)(300)을 기록하는 단계; 및

(b) 하나 이상의 데이터베이스들(100) 및/또는 하나 이상의 데이터베이스들(100)을 식별해주는 정보와 함께, 참조 값들(R)(300)을 인코딩된 데이터(30)에 포함시키는 단계를 포함한다.

인코딩된 데이터(30)를 디코딩하여 대응하는 디코딩된 출력 데이터(60)를 발생하는 방법도 역시 제공되어 있고, 이 방법은

(a) 참조 값들(R)(300) 및 영역 식별자들(U)에 관한 정보 및 하나 이상의 데이터베이스들(100)에 관한 정보를 포함하는 인코딩된 데이터(30)를 수신하는 단계;

(b) 인코딩된 데이터(30)로부터 참조 값들(R)(300)을 디코딩하는 단계;

(c) 참조 값들(R)에 의해 지시된 대로 하나 이상의 데이터베이스들(100)로부터의 하나 이상의 요소들(E)에 액세스하는 단계 - 여기서 하나 이상의 요소들(E)은 하나 이상의 대응하는 데이터 블록들을 나타냄 -; 및

(d) 출력을 위한 대응하는 디코딩된 데이터(60)를 조립하기 위한 하나 이상의 데이터 블록들을 발생하는 단계를 포함한다.

이들 방법은 유익하게도 인코더(10)에서, 디코더(50)에서 그리고 코덱(5)에서 구현된다. 코덱(5)에서, 적어도 하나의 인코더(10) 및 적어도 하나의 디코더(50)는 인코딩된 데이터(30)에 포함되어 있는 참조 값들(R)에 의해 참조되는 하나 이상의 데이터베이스들(100)을 공유한다.

명세서

청구범위

청구항 1

전송 또는 저장을 위한 대응하는 인코딩된 데이터(30)를 발생하도록 소스 데이터(20)를 인코딩하는 방법에 있어서,

상기 소스 데이터는 오디오 데이터, 영상 데이터, 비디오 데이터, 그래픽 데이터, 다차원 데이터, 측정 데이터 중 적어도 하나이고, 상기 방법은,

(a) 상기 소스 데이터(20)의 하나 이상의 부분들을 하나 이상의 데이터베이스들(100) 내의 하나 이상의 요소들(E) - 상기 하나 이상의 요소들(E)은 대응하는 하나 이상의 데이터 블록들을 나타냄 - 에 정합시키고 (matching), 상기 소스 데이터(20)의 상기 하나 이상의 부분들을 상기 하나 이상의 정합된 요소들(E)에 관련시키는 참조 값들(R)을 기록하는 단계;

(b) 상기 하나 이상의 데이터베이스들(100) 및/또는 상기 하나 이상의 데이터베이스들(100)을 식별해주는 정보와 함께, 상기 참조 값들(R)을 상기 인코딩된 데이터(30)에 포함시키는 단계;

(c) 상기 소스 데이터(20)를 고유의 영역 식별자(U)를 가지는 영역들로 나누는 단계;

(d) 상기 영역들로부터 영역-특유의(area-specific) 샘플들을 취하고 상기 샘플들에 기초하여 대응하는 참조 값들(R)을 계산하는 단계;

(e') 사용한 인코딩 방법을 저장하고/저장하거나 전송하는 단계;

(f1) 소스 데이터(20)에 적용된 계산들로부터 획득된 상기 참조 값들(R)에 대응하는 상기 소스 데이터(20)의 영역들 중 하나의 영역이 상기 하나 이상의 데이터베이스들(100)에 이미 저장되었거나 전송된 경우, 요소들(E)을 일의적으로(uniquely) 식별하기 위해, 소스 데이터(20)의 영역들 중 상기 하나의 영역을 저장하거나 전송하지 않고, 상기 참조 값들(R) 또는 압축된 상태의 상기 참조 값들(R)을 상기 인코딩된 데이터(30)에 저장하고/저장하거나 상기 인코딩된 데이터(30)를 통해 전송하거나,

(f2) 소스 데이터(20)에 적용된 계산들로부터 획득된 상기 참조 값들(R)에 대응하는 상기 소스 데이터(20)의 영역들 중 상기 하나의 영역이 상기 하나 이상의 데이터베이스들(100)에 저장되었거나 전송된 적이 없는 경우, 상기 참조 값들(R)을 저장하거나 전송하지 않고, 상기 인코딩된 및/또는 원래의 소스 데이터(20)의 상기 영역들 중 하나의 영역을 상기 인코딩된 데이터(30)로 저장하고/저장하거나 상기 인코딩된 데이터(30)를 통해 전송하는 단계를 포함하는 것인 인코딩 방법.

청구항 2

제1항에 있어서, 상기 방법은 상기 하나 이상의 데이터베이스들(100)을 구현하기 위해 하나 이상의 제로 평균(zero-mean) 데이터베이스들을 이용하는 단계를 포함하는 것인 인코딩 방법.

청구항 3

제1항에 있어서, 상기 방법은,

(i) 하나 이상의 동적 데이터베이스들(120)로부터의 요소들을 선택함으로써 하나 이상의 정적 데이터베이스들(110)을 발생하는 단계;

(ii) 상기 하나 이상의 동적 데이터베이스들(120)에 포함시키기 위한 하나 이상의 요소들(E)을 발생하기 위해 상기 소스 데이터(20)의 상기 하나 이상의 부분들에 존재하는 정보를 처리하는 단계; 및

(iii) 하나 이상의 이전의 데이터베이스들(100)로부터 요소들(E)을 선택하고 이들을 상기 인코딩된 데이터(30)와 함께 사용하기 위해 상기 하나 이상의 동적 데이터베이스들(120)에 포함시키는 단계

중 적어도 하나를 포함하는 것인 인코딩 방법.

청구항 4

제1항에 있어서, 상기 하나 이상의 데이터베이스들(100)은 하나 이상의 정적 데이터베이스들(110) 및/또는 하나 이상의 동적 데이터베이스들(120)을 포함하고, 상기 하나 이상의 데이터베이스들(100)은 하나 이상의 데이터 서버들에서 호스팅되는 것인 인코딩 방법.

청구항 5

제4항에 있어서, 상기 방법은 상기 하나 이상의 데이터 서버들을 원격 제어하는 단계를 포함하는 것인 인코딩 방법.

청구항 6

제4항에 있어서, 상기 방법은 상기 인코딩된 데이터(30)를 발생하는 동작을 하는 적어도 하나의 인코더(10)와 상기 인코딩된 데이터(30)를 수신하도록 되어 있는 적어도 하나의 디코더(50) 사이에서 상기 하나 이상의 데이터베이스들(100)을 공유하는 단계를 포함하는 것인 인코딩 방법.

청구항 7

제1항에 있어서, 상기 방법은 통신 네트워크를 통해 상기 하나 이상의 데이터베이스들(100)에 결합되어 있는 인코더(10)에서 구현되고, 상기 하나 이상의 데이터베이스들(100) 중 적어도 하나는 상기 인코더(10)의 근거리 통신망과 상호 유사한 통신 네트워크의 근거리 통신망(LAN, 블루투스, WLAN, GSM, 3G, 4G, 5G, LTE)에 있는 것인 인코딩 방법.

청구항 8

제1항에 있어서, 상기 참조 값들(R)(300)은 상기 인코딩된 데이터(30)에 포함시키기 위해 개별적으로 인코딩되는 복수의 부분들(310A, 310B, 310C)을 포함하고 있는 것인 인코딩 방법.

청구항 9

제1항에 있어서, 하나 이상의 동적 데이터베이스들(120)의 요소들은 상기 소스 데이터(20)의 하나 이상의 부분들과 하나 이상의 정적 데이터베이스들(110)에서의 하나 이상의 요소들 간의 일치하는 것들이 발견되지 않을 때 발생하는 것인 인코딩 방법.

청구항 10

제1항에 있어서, 상기 소스 데이터(20)의 하나 이상의 부분들과 상기 하나 이상의 데이터베이스들(100)의 하나 이상의 요소들(E)을 정합시키는 것이 품질 임계값 내에서 행해지고, 상기 품질 임계값은 상기 참조 값들(R)의 발생 동안 동적으로 변경되는 것인 인코딩 방법.

청구항 11

제1항에 있어서, 상기 참조 값들(R)은 상기 참조 값들(R)에 의해 정의된 하나 이상의 요소들(E)을 사용하여 상기 소스 데이터(20)의 하나 이상의 대응하는 부분들을 재구성하는 데 사용되고, 상기 재구성된 하나 이상의 부분들과 상기 소스 데이터에서의 대응하는 원래의 하나 이상의 부분들 간의 오차들이 결정되고, 상기 오차들이 인코딩되어 상기 인코딩된 데이터(30)에 포함되는 것인 인코딩 방법.

청구항 12

제1항에 있어서, 제1 데이터베이스(100)의 하나 이상의 요소들(E)이 상기 하나 이상의 데이터베이스들(100) 중 하나 이상의 다른 데이터베이스에 존재하는 하나 이상의 요소들(E)을 참조하는 동작을 하는 것인 인코딩 방법.

청구항 13

제1항에 있어서, 상기 하나 이상의 데이터베이스들(100)은 상기 소스 데이터(20)에 존재하는 콘텐츠의 성질에 따라 선택되고 및/또는 크기가 변화되는 것인 인코딩 방법.

청구항 14

대응하는 인코딩된 데이터(30)를 발생하도록 소스 데이터(20)를 인코딩하는 인코더(10)에 있어서,

상기 소스 데이터는 오디오 데이터, 영상 데이터, 비디오 데이터, 그래픽 데이터, 다차원 데이터, 측정 데이터 중 적어도 하나이고, 상기 인코더(10)는,

(a) 상기 소스 데이터(20)의 하나 이상의 부분들을 하나 이상의 데이터베이스들(100) 내의 하나 이상의 요소들(E) - 상기 하나 이상의 요소들(E)은 대응하는 하나 이상의 데이터 블록들을 나타냄 - 에 정합시키고, 상기 소스 데이터(20)의 상기 하나 이상의 부분들을 상기 하나 이상의 정합된 요소들(E)에 관련시키는 참조 값들(R)을 기록하는 제1 데이터 처리 하드웨어; 및

(b) 상기 하나 이상의 데이터베이스들(100) 및/또는 상기 하나 이상의 데이터베이스들(100)을 식별해주는 정보와 함께, 상기 참조 값들(R)을 상기 인코딩된 데이터(30)에 포함시키는 제2 데이터 처리 하드웨어를 포함하고, 상기 인코더(10)는,

(c) 상기 소스 데이터(20)를 수신하고, 상기 소스 데이터(20)를 고유의 영역 식별자(U)를 가지는 영역들로 나누고;

(d) 상기 영역들로부터 영역-특유의 샘플들을 취하고 상기 샘플들에 기초하여 대응하는 참조 값들(R)을 계산하며;

(e') 사용한 인코딩 방법을 저장하고/저장하거나 전송하고;

(f1) 소스 데이터(20)에 적용된 계산들로부터 획득된 상기 참조 값들(R)에 대응하는 상기 소스 데이터(20)의 영역들 중 하나의 영역이 상기 하나 이상의 데이터베이스들(100)에 이미 저장되었거나 전송된 경우, 요소들(E)을 일의적으로(uniquely) 식별하기 위해, 소스 데이터(20)의 영역들 중 상기 하나의 영역을 저장하거나 전송하지 않고, 상기 참조 값들(R) 또는 압축된 상태의 상기 참조 값들(R)을 상기 인코딩된 데이터(30)에 저장하고/저장하거나 상기 인코딩된 데이터(30)를 통해 전송하거나,

(f2) 소스 데이터(20)에 적용된 계산들로부터 획득된 상기 참조 값들(R)에 대응하는 상기 소스 데이터(20)의 영역들 중 상기 하나의 영역이 상기 하나 이상의 데이터베이스들(100)에 저장되었거나 전송된 적이 없는 경우, 상기 참조 값들(R)을 저장하거나 전송하지 않고, 상기 인코딩된 및/또는 원래의 소스 데이터(20)의 상기 영역들 중 하나의 영역을 상기 인코딩된 데이터(30)로 저장하고/저장하거나 상기 인코딩된 데이터(30)를 통해 전송하는 동작을 하는 것인 인코더.

청구항 15

제14항에 있어서, 상기 인코더(10)는 제1항 내지 제13항 중 어느 한 항에 청구된 인코딩 방법을 실행하는 동작을 하는 것인 인코더.

청구항 16

대응하는 디코딩된 출력 데이터(60)를 발생하도록 제1항에 청구된 방법에 의해 제공되는 인코딩된 데이터(30)를 디코딩하는 방법에 있어서,

(a) 참조 값들(R)(300) 및 영역 식별자(U)에 관한 정보 및 하나 이상의 데이터베이스들(100)에 관한 정보를 포함하는 인코딩된 데이터(30)를 수신하는 단계;

(b) 상기 인코딩된 데이터(30)로부터 상기 참조 값들(R)(300)을 디코딩하는 단계;

(c) 상기 참조 값들(R)에 의해 지시된 대로 상기 하나 이상의 데이터베이스들(100)로부터의 하나 이상의 요소들(E) - 상기 하나 이상의 요소들(E)은 하나 이상의 대응하는 데이터 블록들을 나타냄 - 에 액세스하는 단계; 및

(d) 대응하는 전체 디코딩된 출력 데이터(60)를 조립하기 위한 상기 하나 이상의 데이터 블록들을 발생하는 단계를 포함하는 디코딩 방법.

청구항 17

제16항에 있어서, 상기 방법은,

(e) 데이터 - 상기 데이터는 상기 참조 값들(R)에 대응하는 오디오 데이터, 영상 데이터, 비디오 데이터, 그래

픽 데이터, 다차원 데이터, 측정 데이터 중 적어도 하나를 포함함 - 를 데이터베이스에 유지하는 단계;

(f) 상기 인코딩된 데이터(30)에 포함되어 있는 영역 식별자들에 관한 정보에 기초하여 영역 식별자들(U)을 발생하는 단계;

(g) 하나 이상의 영역 식별자들(U)에 대응하는 참조 값(R)을 상기 인코딩된 데이터로부터 수신하거나 검색하는 단계;

(h) 상기 참조 값(R)에 대응하는 데이터를, 대응하는 영역 식별자(U)에 대한 상기 참조 값(R)에 기초하여 상기 데이터베이스로부터 검색하는 단계; 및

(i) 상기 영역 식별자(U)에 기초하여 전체 디코딩된 출력 데이터(60)로 조립되는 데이터 블록을, 상기 데이터베이스로부터 검색된 상기 데이터에 기초하여, 발생하는 단계를 포함하는 것인 디코딩 방법.

청구항 18

제16항에 있어서, 상기 하나 이상의 데이터베이스들(100)은 하나 이상의 정적 데이터베이스들(110) 및/또는 하나 이상의 동적 데이터베이스들(120)을 포함하는 것인 디코딩 방법.

청구항 19

제16항에 있어서, 상기 방법은 통신 네트워크를 통해 상기 하나 이상의 데이터베이스들(100)에 결합되어 있는 디코더(50)에서 실행되는 것인 디코딩 방법.

청구항 20

제16항에 있어서, 상기 하나 이상의 데이터베이스들(100)은 공간적으로 상기 디코딩 방법을 실행하도록 구성되어 있는 데이터 처리 하드웨어(50)에 로컬적으로 배치되어 있는 것인 디코딩 방법.

청구항 21

제20항에 있어서, 상기 하나 이상의 데이터베이스들(100)은 상기 데이터 처리 하드웨어(50)도 또한 포함하는 근거리 통신망(LAN, 블루투스, WLAN, GSM, 3G, 4G, 5G, LTE)에서 호스팅되고, 상기 데이터 처리 하드웨어(50)는 상기 데이터 처리 하드웨어(50)의 근거리 통신망과 상호 유사한 통신 네트워크의 근거리 통신망(LAN, 블루투스, WLAN, GSM, 3G, 4G, 5G, LTE)에 있는 상기 하나 이상의 데이터베이스들(100)에 결합되어 있는 것인 디코딩 방법.

청구항 22

제16항에 있어서, 상기 하나 이상의 데이터베이스들(100)은 하나 이상의 데이터 서버들에 의해 제공되는 것인 디코딩 방법.

청구항 23

제22항에 있어서, 상기 하나 이상의 데이터 서버들은 원격 제어되는 것인 디코딩 방법.

청구항 24

제22항에 있어서, 상기 하나 이상의 데이터베이스들(100)은 상기 인코딩된 데이터(30)를 발생하는 동작을 하는 적어도 하나의 인코더(10)와 상기 인코딩된 데이터(30)를 수신하도록 되어 있는 데이터 처리 하드웨어(50)를 포함하는 적어도 하나의 디코더 사이에서 공유되는 것인 디코딩 방법.

청구항 25

제16항에 있어서, 상기 방법은 하나 이상의 정적 데이터베이스들(110)의 하나 이상의 요소들(E)로부터 및/또는 상기 인코딩된 데이터(30)에 제공되는 정보로부터 하나 이상의 동적 데이터베이스들(120)을 발생하는 단계를 포함하고, 상기 발생된 하나 이상의 동적 데이터베이스들(120)은 상기 인코딩된 데이터(30)를 디코딩하는 데 이용되는 것인 디코딩 방법.

청구항 26

대응하는 디코딩된 데이터(60)를 발생하도록 제1항에 청구된 방법에 의해 제공되는 인코딩된 데이터(30)를 디코딩하는 디코더(50)에 있어서,

- (a) 참조 값들(R)(300) 및 영역 식별자들(U)에 관한 정보 및 하나 이상의 데이터베이스들(100)에 관한 정보를 포함하는 인코딩된 데이터(30)를 수신하는 제1 데이터 처리 하드웨어;
- (b) 상기 인코딩된 데이터(30)로부터 상기 참조 값들(R)(300)을 디코딩하는 제2 데이터 처리 하드웨어;
- (c) 상기 참조 값들(R)(300)에 의해 지시된 대로 상기 하나 이상의 데이터베이스들(100)로부터의 하나 이상의 요소들(E) - 상기 하나 이상의 요소들(E)은 하나 이상의 대응하는 데이터 블록들을 나타냄 - 에 액세스하는 제3 데이터 처리 하드웨어; 및
- (d) 대응하는 전체 디코딩된 출력 데이터(60)를 조립하기 위한 상기 하나 이상의 데이터 블록들을 발생하는 제4 데이터 처리 하드웨어를 포함하는 디코더.

청구항 27

제26항에 있어서, 상기 디코더(50)는 제16항 내지 제25항 중 어느 한 항에 청구된 방법을 실행하는 동작을 하는 것인 디코더.

청구항 28

코텍(5)에 있어서,
 소스 데이터(20)를 인코딩하여 대응하는 인코딩된 데이터(30)를 발생하는 제14항에 청구된 적어도 하나의 인코더(10), 및 상기 인코딩된 데이터(30)를 수신하고 상기 인코딩된 데이터(30)를 디코딩하여 대응하는 디코딩된 데이터(60)를 발생하는 제26항에 청구된 적어도 하나의 디코더를 포함하는 코텍.

청구항 29

소프트웨어 제품이 기록되어 있는 비밀시적 기계 판독가능 데이터 저장 매체에 있어서,
 상기 소프트웨어 제품은 제1항에 청구된 방법을 구현하기 위해 인코더(10)의 컴퓨팅 하드웨어 상에서 실행가능한 것인 비밀시적 기계 판독가능 데이터 저장 매체.

청구항 30

소프트웨어 제품이 기록되어 있는 비밀시적 기계 판독가능 데이터 저장 매체에 있어서,
 상기 소프트웨어 제품은 제16항에 청구된 방법을 구현하기 위해 디코더(50)의 컴퓨팅 하드웨어 상에서 실행가능한 것인 비밀시적 기계 판독가능 데이터 저장 매체.

청구항 31

프로세서 및 프로그램 명령어들을 저장하는 저장소(storage)를 포함하는 디바이스에 있어서, 상기 프로그램 명령어들은, 상기 프로세서에 의해 실행될 때, 상기 프로세서로 하여금 제1항에 청구된 인코딩 방법을 수행하도록 구성된 것인, 디바이스.

청구항 32

프로세서 및 프로그램 명령어들을 저장하는 저장소를 포함하는 디바이스에 있어서, 상기 프로그램 명령어들은, 상기 프로세서에 의해 실행될 때, 상기 프로세서로 하여금 제16항에 청구된 디코딩 방법을 수행하도록 구성된 것인, 디바이스.

청구항 33

제1항에 있어서, 상기 단계 (f2)가 수행될 때, 상기 참조 값들(R)에 대응하는 상기 소스 데이터(20)의 영역들 중 상기 하나의 영역이 이전에 상기 하나 이상의 데이터베이스들(100)에 저장된 적 없는 경우, 상기 소스 데이터(20)의 영역들 중 상기 하나의 영역을 새로운 요소(E)로서 상기 하나 이상의 데이터베이스들(100)에 저장하는

단계를 포함하는, 인코딩 방법.

발명의 설명

기술 분야

[0001] 본 개시 내용은 소스 데이터(예를 들어, 포착된 영상 데이터 및/또는 오디오 데이터)를 인코딩하여 대응하는 인코딩된 데이터를 발생하는 인코더에 관한 것이고; 더욱이, 본 개시 내용은 인코딩된 데이터를 디코딩하여 대응하는 디코딩된 데이터(예를 들어, 소스 데이터를 나타내는 디코딩된 데이터)를 발생하는 디코더에 관한 것이며; 인코더 및 디코더는, 예를 들어, 디코딩된 데이터를 발생하기 위해 디코더에서 인코딩된 데이터와 함께 처리되는 정보를 포함하는 하나 이상의 데이터베이스들을 이용하는 동작을 하고, 하나 이상의 데이터베이스들은, 예를 들어, 하나 이상의 데이터 서버들에 위치해 있다. 그에 부가하여, 본 개시 내용은 소스 데이터를 인코딩하여 대응하는 인코딩된 데이터를 발생하는 방법에 관한 것이고; 또한 그에 부가하여, 본 개시 내용은 인코딩된 데이터를 디코딩하여 대응하는 디코딩된 데이터(예를 들어, 소스 데이터를 나타내는 디코딩된 데이터)를 발생하는 방법에 관한 것이며; 이 방법은, 예를 들어, 디코딩된 데이터를 발생하기 위해 인코딩된 데이터와 함께 처리되는 정보를 포함하는 하나 이상의 데이터베이스들을 이용하는 것을 포함한다. 게다가, 본 개시 내용은 비일시적(non-transitory, 또는 non-transient) 기계 판독가능 데이터 저장 매체 상에 기록되는 소프트웨어 제품을 관한 것이며, 여기서 소프트웨어 제품은 상기 방법을 구현하는 컴퓨팅 하드웨어 상에서 실행가능하다.

배경 기술

[0002] 종래에는, 예를 들어, 포착된 영상 데이터의 최근의 MPEG 인코딩을 통해 소스 데이터를 인코딩하는 것은 소스 데이터에 하나 이상의 변환들을 적용함으로써 소스 데이터를 인코딩된 데이터로 변환하기 위해 소스 데이터를 처리하는 것을 수반한다. 더욱이, 인코딩된 데이터를 디코딩하는 것은 인코딩된 데이터에 하나 이상의 역변환들을 적용함으로써 인코딩된 데이터를 대응하는 디코딩된 데이터(예를 들어, 소스 데이터를 나타내는 디코딩된 데이터)로 변환하기 위해 인코딩된 데이터를 처리하는 것을 수반한다. 이러한 인코딩은 소스 데이터를 압축하는 데, 예를 들어, 전송을 위해 또는 데이터 캐리어(data carrier) 상에 저장하기 위해 그의 데이터 크기를 감소시키는 데 이용될 수 있고; 다른 대안으로서 또는 그에 부가하여, 이러한 인코딩은 전송 매체를 통해, 예를 들어, 인터넷을 통해 전송될 때 소스 데이터의 보안을 증가시키기 위해 이용될 수 있다. 이러한 인코더 및 디코더를, 예를 들어, 디지털 카메라, DVD 플레이어 및 유사한 소비자 제품에 내장되어 있는, 자체 완비된(self-contained) 처리 유닛으로서 구현하는 것이 관례이다. 그렇지만, 데이터 처리 장치들의 상호연결성이 최근에 더 커짐에 따라, 예를 들어, 최근에 발전되고 업그레이드된 인코딩 변환을 사용하여 인코딩되어 있는 인코딩된 데이터에 대처하도록 디코더를 적응시키기 위해, 인코더 및 디코더가 재구성될 수 있도록 이들을 소프트웨어 제품을 사용하여 구현되도록 설계하는 것이 보다 최근의 관례가 되었다.

[0003] 공개된 미국 특허 US 4553171에는, 원래의 영상의 픽셀 블록들을 나타내는 일련의 코드 워드들을 참조하여 디지털 영상을 디지털적으로 인쇄하는 방법이 기술되어 있다. 원래의 영상은 그 영상을 균일한 크기의 픽셀 블록들로 세분함으로써 인코딩된다. 선택적으로, 각각의 픽셀 블록은, 이전에 스캔된 블록들 중 임의의 것과 동일하지 않는 한, 고유 식별자를 부여받는다. 동일한 픽셀 블록들은 동일한 식별자를 부여받는다. 상기한 픽셀 블록들에 의해 표현되는 원래의 영상을 코드 워드들로서 표현되는 식별자들에 의해 표현되는 것으로서 인쇄할 때, 식별자들이 순차적으로 스캔되고, 원래의 영상의 인쇄물을 발생하는 프린터를 조절하기 위해 대응하는 픽셀 블록들이 버퍼 메모리에 연속적으로 저장된다. 이와 같이, 픽셀 블록들의 표현들을 저장하는 일종의 데이터베이스를 참조하는 블록 인덱스 번호를 포함하는 식별자들을 포함하는 데이터로부터 원래의 영상이 인쇄된다.

[0004] 미국 특허 US 4013828호에는, 영상을 처리하는 방법이 기술되어 있으며, 여기서 영상이 스캔되고 소정의 크기의 화소 그룹으로 디더링 처리된다. 각각의 화소 그룹은 원격지에 위치한 디스플레이 패널의 대응하는 셀 그룹에 대응한다. 각각의 화소 그룹이 스캔될 때, 디더링된 영상 비트를 화소 그룹에 대응시킴으로써 표현되는 패턴이 제1 메모리에 저장되어 있는 패턴들의 사전과 비교된다. 그 패턴이 제1 메모리에 저장되어 있는 패턴들 중에 있지 않은 경우, 그 패턴은 연관된 코드를 할당받고 메모리에 입력된다. 더욱이, 패턴 및 그의 코드 워드 둘 다는 원격지에 위치한 디스플레이 패널로 전송되고, 그곳에서 제2 메모리에 저장되며; 이어서 제2 메모리가 액세스되고, 제2 메모리에 저장되어 있는 패턴에 따라, 스캔된 픽처에 대응하는 셀 그룹의 개별 셀에 전원이 공급된다. 스캔된 패턴이 제1 메모리에 이미 저장되어 있는 것과 동일한 경우, 즉 제2 메모리로 이미 전송된 데이터를 재사용함으로써, 제1 메모리와 제2 메모리 사이의 데이터 흐름의 감소를 달성하기 위해, 그 패턴과 연관되어 있는 그 코드 워드만이 원격 위치로 전송된다. 그렇지만, 이 방법은 종래의 의미에서 데이터 흐름을 직접

압축하지는 않는다.

- [0005] 공개된 영국 특허 GB 2362055에는, 영상을 코딩하는 방법이 기술되어 있으며, 여기서 이 방법은
- [0006] (a) 영상을 영상 블록들로 나누는 단계;
- [0007] (b) 영상 블록들이 데이터베이스에 이미 존재하는 블록들과 비교되는 방식으로 영상 블록들을 인코딩하고, 영상 블록들 중에서의 대응하는 것과 충분히 양호하게 일치하는 것인 기존의 블록을 데이터베이스로부터 선택하는 단계; 및
- [0008] (c) 이미 존재하는 블록들을 나타내는 코드들을 참조하여 영상 블록들을 코딩하는 단계를 포함한다.
- [0009] 이 방법은 코드 라이브러리, 즉 데이터베이스를 이용한다. 그렇지만, 서로 아무런 관련성도 갖지 않는 데이터베이스 요소들과 연관된 데이터로 인해, 일치하는 것을 찾기 위해 데이터베이스에서 검색을 수행하는 것이 어려우며, 데이터베이스의 크기가 증가됨에 따라 상당한 계산 자원을 필요로 한다. 이와 달리, 데이터베이스가 작을 때는, 영상 블록들을 데이터베이스에 이미 존재하는 블록들과 정합시키는 것이 열화되고, 그 결과 이 방법에 의해 발생하는 코딩된 데이터로부터 충분한 정도의 품질로 영상이 구성될 수 없다. 그렇지만, 이 방법은 충분히 양호하게 일치하는 것이 발견되지 않는 경우 새로운 요소들을 생성하는 것을 포함하고, 여기서 새로운 요소는 그 요소를 식별해준 참조 값과 함께 전송된다.
- [0010] 디지털 데이터, 예를 들어, 비디오, 영상, 그래픽 및 오디오의 사용이 해마다 급속히 증가하고 있다. 이러한 사용으로 인해, 저장 및 전송되는 데이터의 양도 역시 경과 시간의 함수로서 급속히 증가하였다. 더욱이, 저장 및 전송되는 데이터의 이러한 증가는 하드웨어 디바이스에 대해 점점 더 많은 자원, 예를 들어, 더 많은 처리 용량을 제공하기 위해 소모되는 더 많은 전력 및 더 큰 통신 전송 대역폭을 필요로 한다. 공개된 미국 특허 출원 US 2010/322,301(출원인: Gurulogic Microsystems Oy)에 기술되어 있는 것과 같은 영상 발생기는 데이터베이스를 사용함으로써 어떻게 바이트를 절감하고 다양한 영상을 발생할지를 해결하는 기술적 해결책을 정의하고 있다. 그렇지만, 모든 종류의 디지털 데이터, 예를 들어, 영상, 비디오, 그래픽 및 오디오 콘텐츠에 대해 사용하기 위해, 보다 효율적인 방식으로 많은 상이한 유형의 데이터베이스들을 전달하는 동작을 하는, 진보된 데이터베이스의 사용에 기초한 인코더 및 대응하는 디코더가 필요하다.

발명의 내용

- [0011] 본 발명은 소스 데이터(예를 들어, 포착된 영상 데이터, 오디오 데이터 등)를 인코딩하여 대응하는 인코딩된 출력 데이터를 발생하는 개선된 방법, 그리고 또한 소스 데이터를 인코딩하는 방법을 구현하는 동작을 하는 인코더를 제공하기 위한 것이다.
- [0012] 더욱이, 본 발명은 인코딩된 데이터를 디코딩하여 대응하는 디코딩된 출력 데이터를 발생하는 개선된 방법을 제공하기 위한 것이다.
- [0013] 제1 측면에 따르면, 첨부된 청구항 제1항에 청구된 방법이 제공되고: 소스 데이터를 인코딩하여 전송 또는 저장을 위한 대응하는 인코딩된 데이터를 발생하는 방법이 제공되고, 여기서 소스 데이터는 오디오 데이터, 영상 데이터, 비디오 데이터, 그래픽 데이터, 다차원 데이터, 측정 데이터 중 적어도 하나를 포함하며, 이 방법은,
- [0014] (a) 소스 데이터의 하나 이상의 부분들을 하나 이상의 데이터베이스들 내의 하나 이상의 요소들(E)에 정합시키고 - 여기서 하나 이상의 요소들(E)은 대응하는 하나 이상의 데이터 블록들을 나타냄 -, 소스 데이터의 하나 이상의 부분들을 하나 이상의 정합된 요소들(E)에 관련시키는 참조 값들(R)을 기록하는 단계; 및
- [0015] (b) 하나 이상의 데이터베이스들 및/또는 하나 이상의 데이터베이스들을 식별해주는 정보와 함께, 참조 값들(R)을 인코딩된 데이터에 포함시키는 단계를 포함하고,
- [0016] 이 방법은
- [0017] (c) 소스 데이터를 고유의 영역 식별자(U)를 가지는 영역들로 나누는 단계;
- [0018] (d) 영역들로부터 영역-관련 샘플들을 취하고 샘플들에 기초하여 대응하는 참조 값들(R)을 계산하는 단계; 및
- [0019] (e) 소스 데이터에 적용된 계산들로부터 획득된 참조 값들(R)이 하나 이상의 데이터베이스들에 이미 저장되었거나 전송되었는지를 검사하고 요소들(E)을 일의적으로 식별하기 위한 참조 값들(R) 또는 압축된 상태로 있는 참조 값들(R)을 인코딩된 데이터에 저장 및/또는 인코딩된 데이터를 통해 전송하거나, 인코딩된 및/또는 원래의 소스 데이터를 인코딩된 데이터로 저장 및/또는 인코딩된 데이터를 통해 전송하고, 참조 값들(R)이 이전에 하나

이상의 데이터베이스들에 저장되어 있지 않은 경우, 선택적으로 새로운 요소(E)로서 하나 이상의 데이터베이스들에 저장하는 단계를 추가로 포함하고 있다.

- [0020] 본 발명은, 공지된 최근의 코텍과 비교하여 전달될 보다 적은 데이터 비트들을 사용하면서, 우수한 디코딩된 비디오, 영상, 그래픽 및 오디오 콘텐츠 품질을 달성할 수 있는 코텍을 제공한다는 점에서 유익하다.
- [0021] (c)에서, 고유의 영역 식별자(U)는 데이터 블록의 크기, 형상 및 위치를 일의적으로 나타낸다. 크기는 선택적으로 고정된 크기(예를 들어, 32개의 데이터 값, 8x8 데이터 값)이거나, 크기는 선택적으로 데이터의 콘텐츠 유형(예를 들어, 데이터 블록 x 데이터 블록)에 의존한다. 영역 식별자(U) 또는 영역 식별자(U)의 정보는 선택적으로 전달되거나 선택적으로 사전 설정되어 있다.
- [0022] (d)에서 언급된 샘플링과 관련하여, 참조 값(R) 또는 그의 일부만을 계산할 때 데이터의 모든 샘플을 사용하는 것이 선택적으로 가능하다. 더욱이, 샘플의 일부만을 사용하는 것이 선택적으로 가능하고, 그에 의해 선택된 샘플들의 대응하는 영역에서 보다 빠른 성능 및 보다 높은 정확도가 얻어진다. 또한 똑같은 가중치와 비교하여 하나 이상의 원하는 이점을 제공하기 위해 참조 값(R)을 계산할 때 샘플들을 가중하는 것을 사용하는 것이 선택적으로 가능하다. 이용되는 샘플링 방법은, 예를 들어, 강건성을 향상시키기 위해 또는 데이터의 특성을 향상시키기 위해 주어진 개개의 경우에서의 요구에 기초하여 선택될 수 있다.
- [0023] 참조 값들(R)은 선택적으로 다음과 같이 해석된다:
- [0024] (i) 블록/패킷에 대해 몇개의 데이터 값을 계산하는 것이 가능하고; 그들 중 하나가, 예를 들어, 전체 블록/패킷의 데이터 값들의 계산된 참조 값 및 그에 부가하여 서브블록/패킷의 데이터 값들의 계산된 하나 이상의 참조 값들일 수 있으며;
- [0025] (ii) 요소의 참조 값은 진폭, 표준 편차, 평균, 최소, 최대, 해쉬, 인덱스 등과 같은 다수의 부분들로 이루어져 있을 수 있고;
- [0026] (iii) 참조 값들은, 예를 들어, 인코딩된 데이터에 존재하는 공간적 디테일(spatial detail)의 함수로서 상이한 정확도로 계산될 수 있다.
- [0027] 선택적으로, 이 방법은 하나 이상의 데이터베이스들을 구현하기 위해 하나 이상의 제로 평균 데이터베이스들을 이용하는 것을 포함한다. 제로 평균 데이터베이스는 하나 이상의 변환된(translated 또는 transformed) 데이터 값을 이용하고, 여기서 하나 이상의 변환된 데이터 값은 변환되기 전의 그의 대응하는 데이터 값과 비교하여 보다 적은 데이터 메모리 용량을 이용한다. 예를 들어, 제로 평균 데이터베이스는 데이터베이스 내의 주어진 데이터베이스 요소에서의 모든 데이터 값들의 평균을 계산하는 것, 및 이어서 변환된 데이터 값이 제로 값에 대해 변하도록 데이터 값들로부터 평균을 차감하는 것에 의해 발생된다. 이러한 하나 이상의 제로 평균 데이터베이스는 어쩌면 크기가 더 작고, 통신 네트워크를 통해 이들을 전달할 필요가 있을 때 보다 효율적으로 압축된다. 선택적으로, 제로 평균 데이터베이스의 데이터베이스 요소는 극히 드물게 제로 평균이다. 그렇지만, 이는 여전히 데이터베이스 요소이고, 여기서 이를 1 만큼 증가시키는 것에 의해 또는 이를 1 만큼 감소시키는 것에 의해, 그 데이터베이스 요소의 평균보다 제로로부터 더 멀리 떨어져 있는 또는 적어도 그 데이터베이스 요소의 평균 정도로 제로로부터 떨어져 있는 새로운 평균을 획득하는 것이 가능하다. 이러한 이유는 다음과 같다: 주어진 원래의 데이터 블록의 평균이 종종 정수가 아니고 그 대신에 실수(즉, 부동 소수점 수)이며, 따라서 정수 부분이 평균으로부터(즉, 요소로부터의 모든 데이터 값들로부터) 차감될 때, 소수 부분은 범위 (-0.5 ... 0.5)에 또는 범위 [-0.5...0.5)에 있을 것이다.
- [0028] 제로 평균의 목적은 데이터 전송에서가 아니라 하나 이상의 데이터베이스들에서의 공간(즉, 메모리 사용)을 절감하는 데 있다. 실제로, 원래의 동적 범위 대신에, 이어서 각각의 요소 값에 대해 부호 비트가 필요한 것으로 인해 제로 평균 값을 표현하기 위해 하나의 추가 비트가 필요하기 때문에, 제로 평균이 사용될 때 개개의 블록에 대한 데이터 크기가 증가한다. 제로 평균 데이터베이스 요소와 함께 사전 설정된 페데스탈 값(pedestal value)을 사용함으로써 실제의 부호 비트 사용의 필요성이 데이터베이스 저장으로부터 제거될 수 있다. 데이터베이스 요소 값들의 동적 범위도 역시 양자화에 의해 감소될 수 있고, 이는 어떤 정보를 상실할 것이다. 더욱이, 탐색 테이블이 동적 압축을 위해 사용될 수 있지만, 이러한 방식도 역시 정보의 손실을 가져온다. 그렇지만, 제로 평균 요소에서는, 평균이 변할 때 새로운 요소들이 데이터베이스에 삽입될 필요가 없고, 그 대신에 동일한 요소가 재사용될 수 있고, 따라서 문제의 동적 범위에서의 모든 평균들에 대해 하나의 요소로 적당하다. 데이터베이스 자체가 전송되는 경우, 예를 들어, 256개 요소(8 비트 동적 범위의 경우) 대신에 단지 하나의 요소만 전송되면 되고; 즉 참조의 평균의 변동이 새로운 요소들을 데이터베이스에 삽입하지 않고 그 대신에 최종

적인 데이터 블록을 구성하는 데 데이터베이스 내의 요소와 함께 평균이 사용된다.

- [0029] 선택적으로, 하나 이상의 데이터베이스들이 하나 이상의 서버들, 예를 들어, 제3자(예를 들어, 하나 이상의 서비스 제공자들)에 속하는 하나 이상의 서버들에 위치해 있다. 보다 선택적으로, 하나 이상의 데이터베이스들의 사용은 요금의 지불(예를 들어, 하나 이상의 데이터베이스들에 액세스하기 위한 가입비 및/또는 사용 요금)을 수반한다.
- [0030] 선택적으로, 이 방법을 구현할 때, 하나 이상의 데이터베이스들은 하나 이상의 정적 데이터베이스들 및/또는 하나 이상의 동적 데이터베이스들을 포함한다.
- [0031] 참조 값(R)은 데이터 샘플들에 기초하여 계산된 값들로 이루어져 있지만, 다른 부분들도 역시 포함할 수 있다. 게다가, 참조 값은 실제로는 부정확한 개념이다. 환언하면, 참조 값들은 고속 검색을 위해 계산되며, 따라서, 그들 중 다수가 있을 수 있고, 그들의 정밀도도 역시 매우 크거나, 다른 대안으로서, 작을 수 있다.
- [0032] 그렇지만, 전송된 참조 값(R)과 관련하여, 이것은 인코더와 디코더 사이에서 문제의 블록을 개별적으로 식별해주는 값을 말하며, 이 전송된 참조 값(R)에 기초하여 요소들이 데이터베이스에 저장된다. 정적 데이터베이스의 경우, 전송된 참조 값(R)은 전형적으로 검색에서 사용되었던 몇개의 양자화된 또는 보다 정확한 참조 값(R), 그리고 종종 서수를 말하는 어떤 값도 포함한다. 동적 데이터베이스의 경우, 전송된 참조 값(R)은 보통 이 요소의 구성 순서를 나타내는 값 및 그의 발생 확률에 불과하다.
- [0033] 정적 데이터베이스는 보통 수정되지 않으며, 따라서 전송된 참조 값(R)은, 인코더에서 그리고 디코더에서 모두, 각각의 요소(E)에 대해 항상 동일하다. 다른 한편, 동적 데이터베이스는 연속적으로 변경될 수 있고; 어떤 요소(E)가 가능성이 더 많게 될 때, 그의 참조 값(R)은 선택적으로 계속적으로 감소되고 따라서 보다 효율적으로 압축될 수 있다. 그렇지만, 심지어 전송된 동적 참조 값(R)조차도 블록의 샘플들과 관계가 있는 부분들을 선택적으로 포함하지만, 그들이 필수적인 것은 아니라는 것을 잘 알 것이다.
- [0034] 그렇지만, 동적 요소(E)를 검색하는 것과 관련하여, 이것은 또한 선택적으로, 각각의 요소(E)에 대해, 양호한 정밀도로 저장된 블록의 샘플들로부터 구성된 대응하는 참조 값들(R)을 가짐으로써 추가적으로 가속화될 수 있다. 동적 데이터베이스에서의 검색은 또한 선택적으로 요소들(E)을 어떠한 확률 순서로 저장함으로써 가속화되고, 따라서 발생하는 요소(E)가 종종 보다 빨리 발견된다. 또한 유의할 점은, 정적 데이터베이스의 전송된 참조 값(R)이 서수를 말하는 상기한 부분을 포함하기 때문에, 이 부분이, 특히 몇 비트 길이인 경우, 또한 선택적으로 (그렇게 요망되는 경우) 동적 데이터베이스의 상기한 전송된 참조 값(R)처럼 동적으로 이용된다는 것이다.
- [0035] 그렇지만, 주어진 정적 데이터베이스에서의 이런 종류의 동적 특성은 종종 원하는 특징이 아닌데, 그 이유는, 이러한 경우에, 임의의 주어진 경우에 어느 요소가 참조되는지가 보장될 수 있을 때까지 전체 데이터 스트림이 그의 시작부터 이용되는 비트 위치까지 인코딩될 필요가 있을 것이기 때문이다. 따라서, 정적 데이터베이스의 한가지 아주 중대한 이점(즉, 분명하고 명확하다는 것)이 상실될 것이다. 약간 더 나은 압축비가 어찌면 달성되지만, 이와 동시에, 참조 값(R)의 동적 부분에 의해 획득되었던, 데이터를 디코딩할 때의 연관된 자유 호핑(free hopping)이 상실될 것이다.
- [0036] 이것을 고려하여, 동적 데이터베이스를 사용하는 것은 선택적으로 유익하게도 시간 제한되어 있으며, 이 경우에 그 상기한 데이터에서의 점프가, 동적 데이터베이스의 재설정 지점에서 디코딩이 복원되는 방식으로, 실행될 수 있다. 정적 데이터베이스의 두번째 중대한 이점은 코딩이 개시되기 전에 항상 알려져 있고, 따라서 그의 요소들(E)이 종종 디코더로도 인코더로도 전송될 필요가 없다는 것이다. 환언하면, 참조 값들(R), 양자화된 참조 값들(R), 서수들 및 전송된 참조 값들(R)(즉, 인덱스들) - 전자들의 조합으로 이루어져 있으며 데이터베이스 내의 하나의 고유의 요소(E)를 말하는 것임 - 이 있다.
- [0037] 이와 같이, 인코더에서 검색이 수행될 때, 인코더로부터 파일로 또는 디코더로 전송되는 것보다 더욱 더 정확한 참조 값들(R)이 이용될 수 있다. 또한 검색이 빠르고 정확하도록 하기 위해[즉, 단지 몇번의 검사가 필요하지만 여전히 최상의 요소(E)를 놓치는 일이 거의 없음], 인코더가 각각의 데이터베이스에 대해 가능한 한 크고 정확한 일련의 참조 값들(R)을 저장하는 것이 유익하다.
- [0038] 선택적으로, 이 방법은 이하의 것들 중 적어도 하나를 이용한다:
- [0039] (a) 하나 이상의 동적 데이터베이스들로부터의 요소들을 선택함으로써 하나 이상의 정적 데이터베이스들이 발생되고;
- [0040] (b) 하나 이상의 동적 데이터베이스들에 포함시키기 위한 하나 이상의 요소들(E)을 발생하기 위해 소스 데이터

의 하나 이상의 부분들에 존재하는 정보가 처리되며;

- [0041] (c) 하나 이상의 이전의 데이터베이스들로부터 요소들(E)을 선택하고 이들을 인코딩된 데이터와 함께 사용하기 위해 하나 이상의 동적 데이터베이스들에 포함시키는 것.
- [0042] 더욱이, 본 발명은 데이터 저장 장치에서의(예를 들어, 데이터 메모리, 데이터 캐리어 등에서의) 데이터 크기의 감소라는 이점들을 제공하고, 이는 계산 자원, 전기 전력 소모 및 데이터 로드 시간을 절감시킨다.
- [0043] 보다 선택적으로, 이 방법에서, 전 세계에 걸쳐 있는 인터넷 서비스 제공자(Internet Service Provider, ISP)는 클라이언트 근거리 통신망(Local Area Network, LAN)에서 보다 빠른 정적 데이터베이스를 제공할 수 있고, 그에 의해 보다 효율적인 데이터 전송이 달성될 수 있게 되는데, 그 이유는, 동작 중에, 단지 비교적 소량의 새로운 데이터 블록이 송신되고 기존의 데이터베이스 참조 데이터 블록이 주어진 LAN에 있는 정적 데이터베이스 서버로부터 수신되기 때문이다. 선택적으로, 이 방법은 통신 네트워크를 통해 하나 이상의 데이터베이스들에 결합되어 있는 인코더에서 구현된다. 보다 선택적으로, 하나 이상의 데이터베이스들 중 적어도 하나는 인코더의 근거리 통신망과 상호 유사한 통신 네트워크의 근거리 통신망(LAN)에 있다. 선택적으로, 하나 이상의 데이터베이스들은, 예를 들어, 하나 이상의 데이터 서버들에서 호스팅된다. 예를 들어, 데이터베이스 서버는 선택적으로 이 방법을 구현하는 코덱에 공간적으로 인접해 있는 방식으로(예를 들어, 그 코덱에 공간적으로 인접해 있는 메모리 상에) 구현된다.
- [0044] 선택적으로, 이 방법에서, 참조 값들(R)은 인코딩된 데이터에 포함시키기 위해 개별적으로 인코딩되는 다수의 부분들을 포함하고 있다. 보다 선택적으로, 이 방법에서, 다수의 부분들 중 하나 이상은 인코딩된 데이터에 포함시키기 위해 인코딩되기 이전에 결합된다. 선택적으로, 다수의 부분들은 참조 값(R)을 사용하여 액세스되는 하나 이상의 데이터 블록들의 분산 V, 평균 M 및 진폭 A 특성들에 관련되어 있다.
- [0045] 선택적으로, 분산 V 대신에 표준 편차가 사용될 수 있고, 보다 선택적으로, 최소, 최대, 모드 및 메디안이 새로운 참조 값(R)으로서 사용된다. 최소, 최대 및 진폭과 관련하여, 단지 2개만이 선택적으로 필요하고, 제3의 것은 유익하게도 다른 2개에 기초하여 계산될 수 있다. 더욱이, 다양한 다른 해쉬 값들 및 CRC 값들도 역시, 완벽한 히트가 검색될 때 또는 블록들의 조합에서 그렇지 않았으면 유사한 참조 값들(R)을 가지는 블록들을 서수를 사용하는 것 이외의 다른 방식으로 서로 구분하고자 할 때, 유효한 참조 값들(R)이다. 이러한 방식은 해쉬 및 CRC 값들이 주어진 데이터 블록의 데이터로부터 계산될 수 있고 따라서 이들이 전송 순서에 의존하지 않는다는 점에서 유익하다.
- [0046] 선택적으로, 모든 값들이 검색에서 그리고 전체 정밀도(full precision)로 이용될 수 있지만, 참조 값들(R)이 전송되거나 요소들(E)이 데이터베이스에 저장될 때, 특정의 참조 값들이 특정의 정밀도로 선택되고(0 내지 n개의 항목들), 선택적으로 또한 서수와 관계가 있는 섹션들 또는 부분들 모두는 저장된 데이터베이스 요소들 및 그들의 데이터 값을 명확하고 분명하게 식별해준다.
- [0047] 전송된 참조 값(R)의 파라미터들의 정밀도 및 양도 역시 품질 파라미터(Q)에 의존할 수 있다. 예를 들어, 양호한 품질을 갖는 정적 데이터베이스로부터 전송된 참조 값(R)은 8-비트 평균 값, 8-비트 표준 편차 값, 6-비트 진폭 값, 6-비트 최소 값, 및 8-비트 서수를 포함할 것인 반면, 저급 품질을 갖는 정적 데이터베이스로부터 전송된 참조 값(R)에 대해서는, 선택적으로 7-비트 평균 값, 7-비트 표준 편차 값, 5-비트 진폭 값, 및 4-비트 서수로 충분할 것이다.
- [0048] 동적 데이터베이스와 관련하여, 양호한 품질을 갖는 동적 데이터베이스로부터 전송된 참조 값(R)은 선택적으로, 예를 들어, 16 비트로 표현되는 서수를 포함하고, 저급 품질의 경우, 예를 들어, 10 비트로 충분할 것이다. 저장된 참조 값들(R) 및 검색에서 사용된 참조 값들(R)은, 예를 들어, 선택적으로 그들의 평균에 대해 그리고 표준 편차에 대해 16 비트로 표현되고, 데이터의 동적 특성을 나타내는 정보와 함께, 예를 들어, 진폭, 최소, 최대, 모드 및 메디안에 대해 8 비트로 표현된다. 더욱이, 또한 데이터에 의존하는 모든 참조 값들(R)이 선택적으로, 예를 들어, 인코더에서도 디코더에서도, 디코딩된 데이터 블록의 샘플들로부터 계산되고, 따라서 인코더 및 디코더 둘 다가 자체적으로 항상 이들을 동일한 정밀도로 그리고 원하는 정밀도로 계산할 수 있다는 것을 잘 알 것이다.
- [0049] 인코더는 고속 검색을 위해 전체 정밀도를 필요로 하고, 게다가, 인코더 및 디코더는 심지어, 적어도 전송을 위해 필요한 정밀도로, 이미 계산된 그 값들을 가질 필요가 있고, 따라서 데이터베이스 요소들(E)이, 전송된 참조 값(R)에 의해 나타낸 바와 같이, 나중의 사용을 위해 올바른 주소에 저장될 수 있다. 또한, 참조 값(R)의 양자화 및 압축이 상이한 것이지만, 이들 둘 다 인코딩된 데이터의 양을 감소시킬 것임을 잘 알 것이다. 그렇지

만, 압축은 복원가능한 반면, 양자화는 데이터를 사용하기 더 쉽도록 만들기 위해 원하는 방식으로 정보를 상실한다(예를 들어, 명확히 말하면 데이터베이스의 크기 및 전송될 데이터의 양을 감소시킨다).

- [0050] 선택적으로, 이 방법에서, 참조 값들(R)은 하나 이상의 데이터베이스들 내의 대응하는 하나 이상의 요소들(E)의 검색을 안내하기 위한 정보를 포함하고 있다.
- [0051] 선택적으로, 이 방법에서, 하나 이상의 요소들(E)은 하나 이상의 파라미터들을 포함하고 있으며, 이들 파라미터로부터 하나 이상의 대응하는 데이터 블록들이 보간(interpolation)에 의해 계산될 수 있다. 보간에 추가하여 또는 그에 대한 다른 대안으로서, 테시메이션(decimation), 외삽(extrapolation) 및 크로핑(cropping) 등의 다른 계산들이 선택적으로 이용된다. 이들은, 확대하는 것 또는 축소하는 것에 의해, 또는 값들을 더 큰 블록으로 확장하는 것에 의해, 또는 요소의 보다 작은 섹션만을 사용하는 것에 의해, 사용될 수 있는 모든 이러한 데이터베이스 요소들(E)에 관련되어 있다. 외삽 및 크로핑은, 물론, 어디에서 외삽이 행해질 것인지 및 블록의 어느 부분에서 크로핑이 실행되어야 하는지를 나타내는 정보 비트를 요구하지만, 보간 및 테시메이션은 디코딩될 블록의 크기와 데이터베이스 요소(E)의 블록의 크기 사이의 관계에 따라 자동으로 동작할 수 있다. 선택적으로, 보간 및 테시메이션 방법들은 인코딩된 데이터에 저장되고 및/또는 전송될 수 있거나, 사전 선택될 수 있을 것이다.
- [0052] 선택적으로, 이 방법에서, 하나 이상의 동적 데이터베이스들이 제한된 시간 지속기간 동안 생성되고, 그 후에 소거된다.
- [0053] 선택적으로, 이 방법은 참조 값들(R)을 사용하여 액세스하기 위해 보다 빈번히 액세스되는 요소들(E)을 보다 빠르게 렌더링하도록 데이터베이스들 내의 요소들(E)에 액세스하는 빈도수의 함수로서 데이터베이스들 중 하나 이상을 재구성하는 단계를 포함한다.
- [0054] 선택적으로, 이 방법에서, 소스 데이터의 하나 이상의 부분들과 하나 이상의 정적 데이터베이스들 내의 하나 이상의 요소들(E)의 일치가 발견될 수 없거나 요소들(E)이 그들의 하나 이상의 전송된 참조 값들(R)에서 너무 많은 비트를 사용할 때, 하나 이상의 동적 데이터베이스들의 요소들(E)이 발생된다.
- [0055] 선택적으로, 이 방법에서, 소스 데이터의 하나 이상의 부분들과 하나 이상의 데이터베이스들의 하나 이상의 요소들(E)을 정합시키는 것이 품질 임계값 내에서 행해지고, 여기서 품질 임계값은 참조 값들(R)의 발생 동안 동적으로 변경된다.
- [0056] 선택적으로, 이 방법에서, 하나 이상의 참조 값들(R)이 하나 이상의 참조 값들(R)에 의해 정의된 하나 이상의 요소들(E)을 사용하여 소스 데이터의 하나 이상의 대응하는 부분들을 재구성하는 데 사용되고, 여기서 재구성된 하나 이상의 부분들과 소스 데이터에서의 대응하는 원래의 하나 이상의 부분들 간의 오차들이 결정되고, 이 오차들이 인코딩되어 인코딩된 데이터에 포함된다.
- [0057] 선택적으로, 이 방법에서, 하나 이상의 요소들(E)에 대응하는 하나 이상의 데이터 블록들은 1-D, 2-D 또는 3-D, 시각적으로 디스플레이될 때 다각형, 시각적으로 디스플레이될 때 직사각형, 시각적으로 디스플레이될 때 타원형, 시각적으로 디스플레이될 때 원형형, 시각적으로 디스플레이될 때 장방형(elongate), 시각적으로 디스플레이될 때 삼각형 중 적어도 하나이다.
- [0058] 선택적으로, 이 방법에서, 제1 데이터베이스의 하나 이상의 요소들(E)은 하나 이상의 데이터베이스들 중 하나 이상의 다른 데이터베이스에 존재하는 하나 이상의 요소들(E)을 참조하는 동작을 한다. 환언하면, 제1 데이터베이스 내의 요소(E)는 대응하는 데이터 블록이 도출되는 제2 데이터베이스 내의 요소(E)를 참조할 수 있다.
- [0059] 선택적으로, 이 방법에서, 인코딩된 데이터에 포함되어 있는 하나 이상의 참조 값들(R)은 압축된 형태로 되어 있다.
- [0060] 선택적으로, 이 방법에서, 하나 이상의 데이터베이스들은 선택되고 및/또는 소스 데이터에 존재하는 콘텐츠의 성질에 따라 크기가 변화된다.
- [0061] 제2 측면에 따르면, 소스 데이터를 인코딩하여 대응하는 인코딩된 데이터(30)를 발생하는 인코더가 제공되고, 이 인코더는
- [0062] (a) 소스 데이터의 하나 이상의 부분들을 하나 이상의 데이터베이스들 내의 하나 이상의 요소들(E)에 정합시키고 - 여기서 하나 이상의 요소들(E)은 대응하는 하나 이상의 데이터 블록들을 나타냄 -, 소스 데이터의 하나 이상의 부분들을 하나 이상의 정합된 요소들(E)에 관련시키는 참조 값들(R)을 기록하는 제1 데이터 처리

하드웨어; 및

- [0063] (b) 하나 이상의 데이터베이스들 및/또는 하나 이상의 데이터베이스들을 식별해주는 정보와 함께, 참조 값들(R)을 인코딩된 데이터에 포함시키는 제2 데이터 처리 하드웨어를 포함하고,
- [0064] 여기서 소스 데이터는 오디오 데이터, 영상 데이터, 비디오 데이터, 그래픽 데이터, 다차원 데이터, 측정 데이터 중 적어도 하나를 포함하며;
- [0065] 이 인코더는
- [0066] (c) 하나 이상의 데이터 블록들의 형태로 되어 있는 소스 데이터를 수신하고, 하나 이상의 데이터 블록들을 고유의 영역 식별자(U)를 가지는 영역들로 나누며;
- [0067] (d) 하나 이상의 데이터 블록들의 영역들로부터 영역-관련 샘플들을 취하고 샘플들에 기초하여 대응하는 참조 값들(R)을 계산하며;
- [0068] (e) 하나 이상의 데이터 블록들에 적용된 계산들의 결과로서 획득된 참조 값들(R)이 하나 이상의 데이터베이스들에 이미 저장되었거나 전송되었는지를 검사하고, 요소들(E)을 일의적으로 식별하기 위한 계산된 참조 값들(R) 또는 압축된 참조 값들을 인코딩된 데이터에 저장 및/또는 인코딩된 데이터를 통해 전송하거나, 인코딩된 및/또는 원래의 소스 데이터를 인코딩된 데이터로 저장 및/또는 인코딩된 데이터를 통해 전송하고, 참조 값들(R)이 이전에 하나 이상의 데이터베이스들에 저장되어 있지 않은 경우, 선택적으로 새로운 요소(E)로서 하나 이상의 데이터베이스들에 저장하는 동작을 한다.
- [0069] 선택적으로, 이 인코더에서, 하나 이상의 데이터베이스들은 하나 이상의 정적 데이터베이스들 및/또는 하나 이상의 동적 데이터베이스들을 포함한다.
- [0070] 선택적으로, 이 인코더는 이하의 것들 중 적어도 하나를 이용한다:
- [0071] (a) 하나 이상의 동적 데이터베이스들로부터의 요소들을 선택함으로써 하나 이상의 정적 데이터베이스들이 발생되고;
- [0072] (b) 하나 이상의 동적 데이터베이스들에 포함시키기 위한 하나 이상의 요소들(E)을 발생하기 위해 소스 데이터의 하나 이상의 부분들에 존재하는 정보가 처리되며;
- [0073] (c) 하나 이상의 이전의 데이터베이스들로부터 요소들(E)을 선택하고 이들을 인코딩된 데이터와 함께 사용하기 위해 하나 이상의 동적 데이터베이스들에 포함시키는 것.
- [0074] 선택적으로, 이 인코더는 통신 네트워크를 통해 하나 이상의 데이터베이스들에 결합되어 있고, 여기서 하나 이상의 데이터베이스들 중 적어도 하나는 인코더의 근거리 통신망과 상호 유사한 통신 네트워크의 근거리 통신망(LAN)에 있다. 선택적으로, 하나 이상의 데이터베이스들은 하나 이상의 데이터 서버들을 이용하여 구현된다.
- [0075] 보다 선택적으로, 이 인코더에서, 제1 데이터 처리 하드웨어는 인코딩된 데이터에 포함시키기 위해 개별적으로 인코딩되는 다수의 부분들을 포함하도록 참조 값들(R)을 발생하는 동작을 한다. 보다 선택적으로, 이 인코더는 다수의 부분들 중 하나 이상을 인코딩된 데이터에 포함시키기 위해 인코딩하기 이전에 이들을 결합시키는 동작을 한다.
- [0076] 선택적으로, 이 인코더에서, 참조 값들(R)은 하나 이상의 데이터베이스들 내의 대응하는 하나 이상의 요소들(E)의 검색을 안내하기 위한 정보를 포함하고 있다.
- [0077] 선택적으로, 이 인코더에서, 참조 값들(R)은 파라미터들을 포함하고 있으며, 이들 파라미터로부터 하나 이상의 대응하는 데이터 블록들이 보간에 의해 계산될 수 있다. 선택적으로 또는 다른 대안으로서, 이 인코더에서, 테시메이션, 외삽 및 크로핑이 이용된다. 이러한 계산들은 유익하게도, 확대하는 것 또는 축소하는 것에 의해, 또는 값들을 더 큰 블록으로 확장하는 것에 의해, 또는 요소의 보다 작은 섹션만을 사용하는 것에 의해, 사용될 수 있는 모든 이러한 데이터베이스 요소들에 관련되어 있다. 외삽 및 크로핑은, 그에 부가하여, 어디에서 외삽이 행해질 것인지 및 블록의 어느 부분에서 크로핑이 실행될 것인지에 관한 정보를 요구하지만, 보간 및 테시메이션은 디코딩될 블록의 크기와 데이터베이스 요소의 블록의 크기 사이의 관계에 따라 자동으로 동작할 수 있다. 선택적으로, 보간 및 테시메이션 방법들은 인코딩된 데이터에 저장되고 및/또는 전송될 수 있거나, 사전 선택될 수 있을 것이다.
- [0078] 선택적으로, 이 인코더에서, 하나 이상의 동적 데이터베이스들이 제한된 시간 지속기간 동안 생성되고, 그 후에

소거된다.

- [0079] 선택적으로, 이 인코더는 참조 값들(R)을 사용하여 액세스하기 위해 보다 빈번히 액세스되는 요소들(E)을 보다 빠르게 렌더링하도록 데이터베이스들 내의 요소들(E)에 액세스하는 빈도수의 함수로서 데이터베이스들 중 하나 이상을 재구성하는 동작을 한다.
- [0080] 선택적으로, 이 인코더에서, 소스 데이터의 하나 이상의 부분들과 하나 이상의 정적 데이터베이스들 내의 하나 이상의 요소들(E)의 일치가 발견될 수 없거나 요소들(E)이 그들의 전송된 참조 값(들)(R)에서 너무 많은 비트를 사용할 때, 하나 이상의 동적 데이터베이스들의 요소들(E)이 발생된다.
- [0081] 선택적으로, 이 인코더에서, 소스 데이터의 하나 이상의 부분들과 하나 이상의 데이터베이스들의 하나 이상의 요소들(E)을 정합시키는 것이 품질 임계값 내에서 제1 데이터 처리 하드웨어에 의해 행해지고, 여기서 품질 임계값은 참조 값들(R)의 발생 동안 동적으로 변경된다.
- [0082] 선택적으로, 이 인코더에서, 제1 데이터 처리 하드웨어는 하나 이상의 참조 값들(R)에 의해 정의된 하나 이상의 요소들(E)을 사용하여 소스 데이터의 하나 이상의 대응하는 부분들을 재구성하는 데 하나 이상의 참조 값들(R)을 사용하는 동작을 하고, 여기서 제1 데이터 처리 하드웨어는 재구성된 하나 이상의 부분들과 소스 데이터에서의 대응하는 원래의 하나 이상의 부분들 간의 오차들을 식별하는 동작을 하고, 제2 데이터 처리 하드웨어는 오차들을 인코딩하고 이 오차들을 인코딩된 데이터에 포함시키는 동작을 한다.
- [0083] 선택적으로, 이 인코더에서, 하나 이상의 요소들(E)에 대응하는 하나 이상의 데이터 블록들은 1D, 2D 또는 3D, 시각적으로 디스플레이될 때 다각형, 시각적으로 디스플레이될 때 직사각형, 시각적으로 디스플레이될 때 타원형, 시각적으로 디스플레이될 때 원형형, 시각적으로 디스플레이될 때 장방형, 시각적으로 디스플레이될 때 삼각형 중 적어도 하나이다.
- [0084] 선택적으로, 이 인코더에서, 제1 데이터베이스의 하나 이상의 요소들(E)은 하나 이상의 데이터베이스들 중 하나 이상의 다른 데이터베이스에 존재하는 하나 이상의 요소들(E)을 참조하는 동작을 한다.
- [0085] 선택적으로, 이 인코더에서, 인코딩된 데이터에 포함되어 있는 하나 이상의 참조 값들(R)은 압축된 형태로 되어 있다.
- [0086] 선택적으로, 이 인코더에서, 제1 데이터 처리 하드웨어는 소스 데이터에 존재하는 콘텐츠의 성질에 따라 하나 이상의 데이터베이스들을 선택하는 동작을 한다.
- [0087] 제3 측면에 따르면, 인코딩된 데이터를 디코딩하여 대응하는 디코딩된 출력 데이터를 발생하는 방법이 제공되고, 이 방법은
- [0088] (a) 참조 값들(R) 및 영역 식별자들에 관한 정보 및 하나 이상의 데이터베이스들에 관한 정보를 포함하는 인코딩된 데이터를 수신하는 단계;
- [0089] (b) 인코딩된 데이터로부터 참조 값들(R)을 디코딩하는 단계;
- [0090] (c) 참조 값들(R)에 의해 지시된 대로 하나 이상의 데이터베이스들로부터의 하나 이상의 요소들(E)에 액세스하는 단계 - 여기서 하나 이상의 요소들(E)은 하나 이상의 대응하는 데이터 블록들을 나타냄 -; 및
- [0091] (d) 대응하는 전체 디코딩된 출력 데이터를 조립하기 위한 하나 이상의 데이터 블록들을 발생하는 단계를 포함한다.
- [0092] 선택적으로, 이 방법은
- [0093] (e) 데이터를 데이터베이스에 유지하는 단계 - 여기서 데이터는 참조 값들(R)에 대응하는 오디오 데이터, 영상 데이터, 비디오 데이터, 그래픽 데이터, 다차원 데이터, 측정 데이터 중 적어도 하나를 포함함 -;
- [0094] (f) 인코딩된 데이터에 포함되어 있는 영역 식별자들에 관한 정보에 기초하여 영역 식별자들(U)을 발생하는 단계;
- [0095] (g) 하나 이상의 영역 식별자들(U)에 대응하는 참조 값(R)을 인코딩된 데이터로부터 수신하거나 검색하는 단계;
- [0096] (h) 참조 값(R)에 대응하는 데이터를 대응하는 하나 이상의 영역 식별자들(U)에 대한 참조 값(R)에 기초하여 데이터베이스로부터 검색하는 단계; 및
- [0097] (i) 영역 식별자(U)에 기초하여 전체 디코딩된 출력 데이터로 조립되는 데이터 블록을, 데이터베이스로부터 검

색된 데이터에 기초하여, 발생하는 단계를 포함한다.

- [0098] 이 방법의 단계 (c)와 관련하여, 유의할 점은, 디코더에서의 요소들(E)에 대한 참조 값들(R)이 전송에서 사용된 것(즉, 양자화되어 있고 몇개의 섹션들로 이루어져 있음)이라는 것이다. 동일한 요소(E)에 대해, 몇개의 참조 값들(R)이 존재할 수 있는데, 그 이유는 참조 값(R)이 하나 이상의 동적 데이터베이스들에서 및/또는 하나 이상의 정적 데이터베이스들에서 동시에 사용 중일 수 있기 때문이다. 더욱이, 어떤 양호한 품질의 데이터 블록들이 하위 품질의 데이터 블록에서도 사용될 수 있고, 이어서 상이한 양자화 레벨들에서의 참조 값(R)의 일부가 블록의 상이한 참조 값(R)을 생성할 것이다. 환언하면, 주어진 요소(E)는 선택적으로 몇개의 참조 값들(R)을 가질 수 있지만, 몇개의 요소들(E)이 상이한 데이터베이스들에 존재하지 않는 한, 또는 몇개의 요소들(E)이 상이한 품질들에 대해 사용되지 않는 한, 또는 몇개의 요소들(E)이 몇개의 대안들 중에서 명확하고 분명하게 선택될 수 있는 요소(E)에 대한 이러한 속성을 포함하지 않지 않는 한, 하나의 참조 값(R)이 몇개의 요소들(E)을 참조할 수는 없다.
- [0099] 예를 들어, 평균 값이 생략되어 있는 이러한 요소들(E)이 사용되면, 큰 평균 값을 사용하여 발생한 큰 진폭을 갖는 블록이 작은 평균 값을 가지는 블록에 대한 부정(negation)으로서 사용될 수 있는 반면, 이는 그 자체로서 큰 평균 값을 갖는 블록에 대해 여전히 사용될 수 있다. 이들을 서로 바꾸려고 시도하는 경우, 이것은 데이터 값들의 동적 범위 내에 속하지 않게 될 값들(예컨대, 평균 값을 갖지 않는 ± 8 -비트 데이터 값의 데이터베이스 요소들이 사용되는 8-비트 데이터)을 생성할 것이다. 예를 들어, 진폭이 200이고 저장할 시의 평균 값 value1이 220이고 사용할 시의 평균 값 value2가 30인 경우, 부정을 사용하는 것 또는 부정을 사용하지 않는 것이 명확하게 추론가능할 것이다. 이 경우에, 평균 값을 갖지 않는 데이터베이스 요소의 데이터 값은, 예를 들어, -170일 수 있고, 이는 평균 값 value2와 결합되어 사용되지 않을 수 있지만, 부정으로서, 문제의 데이터 값에 대한 +170이 유효할 것이다. 이와 같이, 전송된 참조 값은 따라서 참조 값들 또는 서수들의 섹션들 이외에 다른 정보를 포함할 수 있지만, 실제로는 이 다른 정보는 종종 유리하게도 별도의 코딩 정보로서 전송된다. 예를 들어, 평균 값을 갖지 않는 데이터베이스의 경우, 이미 평균 값만이 항상 필요한 정보이고, 참조 값(R) 또는 별도의 인코딩 정보 삽입의 일부일 수 있다.
- [0100] 그에 대응하여, 동일한 요소(E)가 회전된 또는 반전된(flipped) 버전을 포함할 것이 요망되는 경우, 상이한 변동들에 대해, 참조 값(R)에 별개의 수치 정보 섹션이 존재할 수 있거나, 다른 대안으로서, 동일한 정보를 표시하기 위해 인코딩 정보의 개별적인 전달이 수행된다. 이들 모두에서의 공통점은, 데이터베이스에 저장되어 있는 하나의 요소(E)에 대해, 평균 값, 회전 및 반전(flipping) 등의 그 파라미터들이 어떻게 설정되어 있는지에 따라, 스케일링, 크로핑, 외삽 등에 대해 유사하게, 상이한 디코딩 블록들이 생성될 수 있다는 사실이다.
- [0101] 이들 정보의 일부는, 이미 수신된 다른 정보에 기초하여 또는 이전의 디코딩된 블록들에 기초하여, 올바른 변동이 추론될 수 있도록 되어 있지만, 그에 대응하여, 대안들의 사용이 디코더에서 허용되어 있는 경우, 디코딩 방법을 명백하게 하기 위해 그 정보의 일부가 전송되어야만 할 것이다. 예를 들어, 부정에 의한 진폭과 평균 값 간의 관계가 큰 진폭에 의해 그리고 작은 또는 큰 평균 값에 의해 추론가능하지만, 그렇지 않은 경우, 추론가능하지 않다. 그에 대응하여, 이전의 디코딩된 블록들에 기초하여, 어느 회전 또는 반전이 올바른 것인지가 종종 연결성에 기초하여 추론될 수 있다. 그렇지만, 추론이 분명하게 명확하지 않은 모든 경우에, 참조 값(R)의 일부로서 또는 별도의 정보로서 부가의 정보를 전송할 필요가 있다.
- [0102] 선택적으로, 이 방법에서, 하나 이상의 데이터베이스들은 하나 이상의 정적 데이터베이스들 및/또는 하나 이상의 동적 데이터베이스들을 포함한다.
- [0103] 선택적으로, 이 방법에서, 하나 이상의 데이터베이스들은 공간적으로 이 방법을 실행하도록 구성되어 있는 데이터 처리 하드웨어에 로컬적으로 배치되어 있다. 보다 선택적으로, 이 방법에서, 하나 이상의 데이터베이스들이 데이터 처리 하드웨어도 포함하고 있는 LAN에서 호스팅된다.
- [0104] 선택적으로, 이 방법은 하나 이상의 정적 데이터베이스들의 하나 이상의 요소들(E)로부터 및/또는 인코딩된 데이터에 제공되는 정보로부터 하나 이상의 동적 데이터베이스들을 발생하는 단계를 포함하고, 여기서 발생한 하나 이상의 동적 데이터베이스들은 인코딩된 데이터를 디코딩하는 데 이용된다.
- [0105] 선택적으로, 이 방법은 1D, 2D 또는 3D, 시각적으로 디스플레이될 때 다각형, 시각적으로 디스플레이될 때 직사각형, 시각적으로 디스플레이될 때 타원형, 시각적으로 디스플레이될 때 원형형, 시각적으로 디스플레이될 때 장방형, 시각적으로 디스플레이될 때 삼각형 중 적어도 하나인 하나 이상의 요소들(E)로부터 대응하는 하나 이상의 데이터 블록들을 발생하는 단계를 포함한다.

- [0106] 제4 측면에 따르면, 인코딩된 데이터를 디코딩하여 대응하는 디코딩된 데이터를 발생하는 디코더가 제공되고, 이 디코더는
- [0107] (a) 참조 값들(R) 및 영역 식별자들(U)에 관한 정보 및 하나 이상의 데이터베이스들에 관한 정보를 포함하는 인코딩된 데이터를 수신하는 제1 데이터 처리 하드웨어;
- [0108] (b) 인코딩된 데이터로부터 참조 값들(R)을 디코딩하는 제2 데이터 처리 하드웨어;
- [0109] (c) 참조 값들(R)에 의해 지시된 대로 하나 이상의 데이터베이스들로부터의 하나 이상의 요소들(E)에 액세스하는 제3 데이터 처리 하드웨어 - 여기서 하나 이상의 요소들(E)은 하나 이상의 대응하는 데이터 블록들을 나타냄 -; 및
- [0110] (d) 대응하는 전체 디코딩된 출력 데이터를 조립하기 위한 하나 이상의 데이터 블록들을 발생하는 제4 데이터 처리 하드웨어를 포함한다.
- [0111] 제1 및 제2 데이터 처리 하드웨어가 동일한 처리 유닛이거나 개별적인 처리 유닛들일 수 있다는 것을 잘 알 것이다. 이와 유사하게, 제1 및/또는 제2 처리 하드웨어의 기능들을 수행하는 많은 처리 유닛들이 있을 수 있다. 가능한 구현예들이 수없이 많으며, 본 명세서에 기술되어 있는 예들로 제한되지 않는다.
- [0112] 선택적으로, 이 디코더는
- [0113] (e) 참조 값들(R) 및 데이터를 데이터베이스에 유지하고 - 여기서 데이터는 참조 값들(R)에 대응하는 오디오 데이터, 영상 데이터, 비디오 데이터, 그래픽 데이터, 다차원 데이터, 측정 데이터 중 적어도 하나를 포함함 -;
- [0114] (f) 인코딩된 데이터에 포함되어 있는 영역 식별자들에 관한 정보에 기초하여 영역 식별자들(U)을 발생하며;
- [0115] (g) 하나 이상의 영역 식별자들(U)에 대응하는 참조 값(R)을 인코딩된 데이터로부터 수신하거나 검색하고;
- [0116] (h) 참조 값(R)에 대응하는 데이터를 하나 이상의 영역 식별자들(U)에 대한 참조 값(R)에 기초하여 메모리로부터 검색하며;
- [0117] (i) 영역 식별자(U)에 기초하여 전체 디코딩된 출력 데이터로 조립되는 데이터 블록을, 데이터베이스로부터 검색된 데이터에 기초하여, 발생하도록 구성되어 있다.
- [0118] 선택적으로, 이 디코더에서, 하나 이상의 데이터베이스들은 하나 이상의 정적 데이터베이스들 및/또는 하나 이상의 동적 데이터베이스들을 포함한다.
- [0119] 선택적으로, 이 디코더에서, 하나 이상의 데이터베이스들은 공간적으로 이 디코더의 데이터 처리 하드웨어에 로컬적으로 배치되어 있다. 보다 선택적으로, 이 디코더에서, 하나 이상의 데이터베이스들이 디코더의 데이터 처리 하드웨어도 포함하고 있는 LAN에서 호스팅된다. 다른 대안으로서 또는 그에 부가하여, 하나 이상의 데이터베이스들이 디코더에, 예를 들어, 그의 데이터 메모리(RAM, ROM)에 위치해 있다.
- [0120] 선택적으로, 이 디코더는 하나 이상의 요소들(E) 및/또는 하나 이상의 정적 데이터베이스들로부터 및/또는 인코딩된 데이터에 제공되는 정보로부터 하나 이상의 동적 데이터베이스들을 발생하는 동작을 하고, 여기서 하나 이상의 발생된 동적 데이터베이스들은 인코딩된 데이터를 디코딩하여 대응하는 디코딩된 데이터를 발생하는 데 이용된다.
- [0121] 선택적으로, 이 디코더는 하나 이상의 요소들(E)로부터 하나 이상의 데이터 블록들을 발생하는 동작을 하고, 여기서 하나 이상의 데이터 블록들은 1D, 2D 또는 3D, 시각적으로 디스플레이될 때 다각형, 시각적으로 디스플레이될 때 직사각형, 시각적으로 디스플레이될 때 타원형, 시각적으로 디스플레이될 때 원형형, 시각적으로 디스플레이될 때 장방형, 시각적으로 디스플레이될 때 삼각형 중 적어도 하나이다.
- [0122] 제5 측면에 따르면, 소스 데이터를 인코딩하여 대응하는 인코딩된 데이터를 발생하는 제2 측면에 따른 적어도 하나의 인코더, 및 인코딩된 데이터를 수신하고 인코딩된 데이터를 디코딩하여 대응하는 디코딩된 데이터를 발생하는 제4 측면에 따른 적어도 하나의 디코더를 포함하는 코덱이 제공된다.
- [0123] 선택적으로, 이 코덱은 하나 이상의 소비자 전자 제품, 예를 들어, 개인용 컴퓨터(PC), 비디오 레코더, 비디오 플레이어, 스마트폰, 게임 디바이스, 과학 장비, 의료 장비, 감시 장치, 보안 장치, 센서 및 기타 측정 장치 및 디지털 카메라에 포함되어 있다.
- [0124] 선택적으로, 이 코덱은 적어도 하나의 인코더 및 적어도 하나의 디코더가 인코딩된 데이터에 포함되어 있는 참

조 값들(R)에 의해 참조되는 하나 이상의 데이터베이스들을 공유하도록 구현된다.

[0125] 제6 측면에 따르면, 비밀시적 기계 관독가능 데이터 저장 매체 상에 기록되어 있는 소프트웨어 제품이 제공되고, 이 소프트웨어 제품은 제1 측면에 따른 방법을 구현하는 인코더의 컴퓨팅 하드웨어 상에서 실행가능하다.

[0126] 제7 측면에 따르면, 비밀시적 기계 관독가능 데이터 저장 매체 상에 기록되어 있는 소프트웨어 제품이 제공되고, 이 소프트웨어 제품은 제3 측면에 따른 방법을 구현하는 디코더의 컴퓨팅 하드웨어 상에서 실행가능하다.

[0127] 본 발명의 특징들이 첨부된 특허청구범위에 의해 한정된 본 발명의 범위를 벗어나지 않고 다양한 조합으로 결합될 수 있다는 것을 잘 알 것이다.

도면의 간단한 설명

[0128] 이제부터, 단지 예로서, 이하의 도면을 참조하여 본 개시 내용의 실시예들에 대해 기술할 것이다.

도 1은 본 개시 내용에 따른 인코더 및 디코더 - 즉, 결합하여 코덱을 구성함 - 의 개략도.

도 2는 도 1의 코덱에서 새로운 요소가 송신되는 것을 나타낸 개략도.

도 3은 도 1의 코덱 내에서 전달되는 참조 값의 다수의 개별적으로 정의된 부분들의 개략도.

첨부 도면에서, 밑줄친 번호는 밑줄친 번호가 위치되어 있는 항목 또는 밑줄친 번호가 인접해 있는 항목을 나타내기 위해 이용된다. 밑줄이 없는 번호는 밑줄이 없는 번호를 항목에 연결시키는 라인에 의해 식별되는 항목에 관한 것이다.

발명을 실시하기 위한 구체적인 내용

[0129] 이하에서 본 개시 내용의 실시예들을 기술할 때, 표 1에 제공되어 있는 바와 같은 약어들이 이용된다:

표 1

실시예들을 기술하는 데 이용되는 약어들의 상세

[0130]

약어	상세
1D	1 차원, 예를 들어, 신호 또는 데이터 패킷을 말함
2D	2 차원, 예를 들어, 신호 또는 데이터 패킷을 말함
3D	3 차원, 예를 들어, 신호 또는 데이터 패킷을 말함
블록	디지털 데이터로부터의 다수의 데이터 요소들, 즉 디지털 데이터의 일부
CRC	순환 중복 검사(cyclic redundancy check)
코덱	디지털 데이터에 대한 인코더 및 디코더
DB	RAM-기반 또는 ROM-기반 메모리 내의 데이터베이스
DC	평균 밝기에 대응하고 영상에 존재하는 최저 공간 주파수를 나타내는 영상의 DC 성분, 즉 영상 평균
델타 코딩	델타 코딩은 전체 데이터 파일보다는 순차적 데이터 간의 차의 형태로 데이터를 저장 또는 전송하는 한 방식이다.
DCT	이산 코사인 변환(Discrete Cosine Transform)
ISP	인터넷 서비스 제공자(Internet Service Provider)
LAN	근거리 통신망(Local Area Network)
RAM	랜덤 액세스 메모리(Random Access Memory)
RD	레이트 왜곡(Rate-Distortion)
RLE	런 길이 인코딩(Run-Length Encoding)
ROI	관심 영역(Region of Interest)
ROM	관독 전용 메모리(Read Only Memory)
VLC	가변 길이 코드(Variable-Length Code)
XOR	배타적 OR(Exclusive OR)(논리 합수)

[0131] 개요에서, 도 1에 예시된 바와 같이, 본 발명은 소스 데이터(20)를 인코딩하여 대응하는 인코딩된 데이터(30)를

발생하는 인코더(10), 및 인코딩된 데이터(30)를 수신하고 디코딩하여 대응하는 디코딩된 데이터(60)를 발생시키는 디코더(50)에 관한 것이며; 인코더(10) 및 디코더(50)는, 결합하여, 코덱(5)을 구성한다. 디코딩된 데이터(60)는 선택적으로 소스 데이터(20) - 예를 들어, 오디오, 1D 영상, 2D 영상, 3D 영상, 비디오 콘텐츠, 그래픽 데이터, 포착된 오디오, 포착된 영상, 포착된 비디오, ASCII 및 이진 데이터는 물론 측정 데이터 및 발생된 데이터 중 적어도 하나임 - 의 표현이다. 인코더(10) 및 디코더(50)는 소스 데이터(20)를 인코딩하기 위해 그리고 인코딩된 데이터(30)를 디코딩하기 위해 하나 이상의 데이터베이스들(100)을 이용하는 동작을 한다. 더욱이, 하나 이상의 데이터베이스들(100)은 하나 이상의 정적 데이터베이스들(110) 및/또는 하나 이상의 동적 데이터베이스들(120)을 포함하며, 이에 대해서는 나중에 더 상세히 기술할 것이다. 선택적으로, 하나 이상의 데이터베이스들(110, 120)은, 예를 들어, 인코더(10) 및 디코더(50)로부터 공간적으로 멀리 떨어져 있는 하나 이상의 데이터 서버들을 통해 제공된다. 선택적으로, 인코더(10) 및 디코더(50) 둘 다에 대해 적어도 하나의 동일한 데이터베이스(100)가 제공되어 있고; 이러한 구성은, 데이터 통신 네트워크 내에서, 예를 들어, 복수의 공간적으로 떨어져 있는 데이터 서버들(예를 들어, 데이터 서버들의 피어-투-피어 네트워크로서 구현됨)에서 데이터의 다수의 사본이 이용가능하게 될 필요가 없기 때문에, 더 높은 데이터 보안을 제공할 수 있다. 다른 대안으로서, 선택적으로, 하나 이상의 데이터베이스들(100)은 인코더(10) 및 디코더(50) 각각이, 예를 들어, 하나 이상의 데이터 서버들로부터 호스팅되는 그 자신의 전용 인코더 데이터베이스 및 디코더 데이터베이스들을 각각 가지도록 구성되어 있다.

[0132] 하나 이상의 데이터베이스들(100)이 많은 상이한 방식으로 구현될 수 있다는 것을 잘 알 것이다. 예를 들어, 인코더(10) 및 디코더(50) 중 하나 이상이, 예를 들어, 하나 이상의 데이터 서버들로서 구현되어 있는 하나 이상의 데이터베이스들(100)로부터의 정보를 필요로 할 때, 그 정보는 직접적인 방식으로, 다른 대안으로서, 간접적인 방식으로 제공될 수 있다. 직접적인 방식에서, 인코더(10) 및/또는 디코더(50)는 정보를 제공하라는 요청을 데이터 서버들 중 하나 이상으로 송신하고; 하나 이상의 데이터 서버들은 이어서 정보(예를 들어, 암호화 정보, 인코딩 방법 정보, 디코딩 방법 정보, 영상 요소들의 라이브러리 등)를 인코더(10) 및/또는 디코더(50)에 제공한다. 간접적인 방식에서, 인코더(10) 및/또는 디코더(50)는 제1 요청을 서버들 중 하나 이상의 서버들로 송신하고, 그에 응답하여, 그 서버들은 그 정보가 이용가능한 하나 이상의 다른 서버들에 관한 정보를 제공하며; 인코더(10) 및/또는 디코더(50)는 이어서 인코더(10) 및/또는 디코더(50)에 정보를 제공하라는 제2 요청을 하나 이상의 다른 데이터 서버들로 송신한다. 선택적으로, 데이터 서버들과 관련하여, 3개 이상의 레벨의 요청이 이용된다. 선택적으로, 불법 인코딩 및/또는 디코딩을 방지하기 위해, 예를 들어, 최근의 음악 산업에 대한 중대한 문제인 불법 파일 전달을 방지하기 위해, 제1 요청 및 제2 요청 중 적어도 하나가 암호화된다. 선택적으로, 제1 요청은 공간적으로 멀리 떨어져 있는 데이터 서버로 송신되고, 제2 요청은 공간적으로 가까운 데이터 서버로 송신되며; 제1 요청과 연관되어 있는 정보 트래픽은 선택적으로 적절하고, 그에 의해 통신 네트워크에 의해 용이하게 대처되는 반면, 제2 요청에 응답하는 것과 연관되어 있는 정보 트래픽은 상당할 수 있고 공간적으로 인코더(10) 및/또는 디코더(50)에 로컬적으로 대처되는 것이 가장 편리하다. 선택적으로, 제1 요청은 인코더(10) 및/또는 디코더(50)가 공간적으로 또는 통신 네트워크 내에서 어디에 배치되어 있는지를 나타내는 위치 정보를 포함하고, 제1 요청에 대한 응답은 관련 인코딩 및/또는 디코딩 정보를 인코더(10) 및/또는 디코더(50)에 제공할 수 있는, 인코더(10) 및/또는 디코더(50)에 공간적으로 가장 가까운 적당한 데이터베이스 또는 가장 가까운 적당히 연결된 데이터베이스(예를 들어, 데이터 서버)에 관한 정보를 포함하고 있다. 선택적으로, 제1 요청에 대한 응답은 적당한 정보를 제공할 수 있는 복수의 가능한 데이터베이스들 중의 선택을 인코더(10) 및/또는 디코더(50)에 제공하고; 선택적으로, 이 선택은 하나 이상의 기술적 기준[예를 들어, 가장 빠른 데이터 연결, 가장 적은 에너지를 소모하는 데이터 연결, 가장 저렴한 데이터 연결(이들로 제한되지 않음)]에 기초하고 있다. 제1 요청 및 제2 요청의 상황이 여기에 설명되어 있지만, 예를 들어, 인코딩된 콘텐츠의 불법 디코딩이 불법 당사자에 의해 수행되기 훨씬 더 어렵게 만들기 위해, 선택적으로, 필요한 경우, 3개 이상의 요청(예를 들어, 다중 요청)이 이용될 수 있다는 것을 잘 알 것이다.

[0133] 선택적으로, 디코더(50)가 그곳에서 콘텐츠 데이터를 디코딩하기 위한 데이터를 획득할 수 있게 해주는 하나 이상의 액세스 코드가 중앙 위치로부터 배포되고, 따라서, 예를 들어, 새로운 블록 버스터 영화가 개봉되어 통신 네트워크를 통해 배포되는 경우, 복수의 이러한 디코더들(50)은 실질적으로 시간상 동시에 인에이블되며, 여기서 관람하기 위한 영화의 개봉이 바람직하게도 조절되고 제어된다. 이러한 제어는 또한, 예를 들어, 법원 명령 또는 저작권 침해가 시행되고 있는 경우, 비디오 콘텐츠의 개봉후 검열이 달성될 수 있게 해준다.

[0134] 본 발명의 실시예들은 유익하게도 미국 특허 출원 US2010/322301("Image Processor, Image Generator and Computer Program"; 발명자 - Tuomas Karkkainen; 출원인: Gurulogic Microsystems Oy, 2009)에 기술되어 있는 것과 같은 영상 발생기에 대해 기술되어 있는 것과 같은 기법을 이용하며, 이 출원의 내용은 참조 문헌으로서

본 명세서에 포함된다. 그렇지만, 본 발명의 실시예들은 이러한 기법들 및 다른 기법들을 보다 효율적인 방식으로 그리고 일반적인 방식으로 이용한다.

[0135] 본 개시 내용의 실시예들에서, 하나 이상의 동적 데이터베이스들(120) 및 그들의 연관된 요소들(E)은, 인코딩된 데이터(30)가 인코더(10)로부터 디코더(50)로 전달될 때, 생성된다. 더욱이, 하나 이상의 정적 데이터베이스들(110)이 유익하게도 하나 이상의 동적 데이터베이스들(120)로부터 생성되거나, 인코딩된 데이터(30)가 디코더(50)로 전달되는 것보다 앞서 전달되거나, 디코더(50)에 사전 설치되어 있다.

[0136] 새로운 데이터베이스 요소들(E)이 디코더(50)로 전달될 때, 이들은, 이용되는 코딩 알고리즘에 따라, 선택적으로 하나 이상의 참조 값들(R)과 함께, 부분적으로 참조 값(R)과 함께, 또는 참조 값(R) 없이 송신된다. 유익하게도, 인코더(10) 및 디코더(50)는 전체 참조 값(R), 또는 주어진 데이터베이스 요소(E)의 참조 값(R)의 일부를, 주어진 데이터베이스 요소(E)에 저장되어야 하는 데이터로부터 직접, 즉 재구성된 데이터 블록의 방식으로 계산하는 동작을 한다. 더욱이, 주어진 데이터베이스 요소(E)와 관련하여, 주어진 데이터베이스 요소(E)의 계산된 참조 값(R)의 적어도 일부가 인코더(10)와 디코더(50) 사이에서 전달될 때, 오차 검출 메커니즘이 구현된다.

[0137] 인코더(10) 및 디코더(50)는 유익하게도, 각각, 표 2에 제공되어 있는 것과 같은 공지의 인코딩 표준들에 따라 적어도 부분적으로 인코딩된 데이터를 인코딩 및 디코딩하는 동작을 한다.

표 2

[0138] 본 개시 내용의 실시예들과 관련하여 이용가능한 인코딩 표준들

JPEG	JPEG MVC	Lucid	AAC
JPEG2000	MP3	GIF	FLAC
JPEG XR	H.261	PNG	Ogg Vorbis
MPEG-1	H.263	TIFF	Speex
MPEG-2	H.264	BMP	Opus
MPEG-4	WebP	VC-1	7Zip
MPEG-4 AVC	VP8	Theora	HVEC
VP9	RAR	ZIP	

[0139] 도 1에서, 하나 이상의 데이터베이스들(100) 내의 요소들(E)에 대한 데이터는 인코더(10)로부터 디코더(50)로의 전송을 위해 인코딩되고, 이는, 인코딩된 영상 데이터 또는 기타 유형의 데이터(오디오 데이터, 비디오 데이터, 그래픽 데이터, 다차원 데이터, 측정 데이터, 텍스트 데이터, 이진 데이터 등)가 전달되고 있을 때, 데이터베이스 요소들(E)을 동시에 전달하는 것을 가능하게 해준다. 요소들(E)은 종종 인코딩된 데이터로부터, 즉 디코딩 후에 생성된다. 더욱이, 하나 이상의 데이터베이스들(100)로부터의 요소들의 인코딩은 인코더(10)와 디코더(50) 사이의 효율적인 데이터베이스 전달을 가능하게 해준다 - 즉, 예를 들어, 디코더(50)를 구현하는 데 이용되는 디바이스들에 데이터베이스(100)를 효율적으로 저장하는 것을 가능하게 해준다 -. 하나 이상의 정적 데이터베이스들(110)이 이용되는 경우에, 인코더(10) 및 디코더(50)는 이용가능한 데이터베이스들을 알게 되고; 다른 대안으로서, 인코더(10) 및 디코더(50)는 어느 데이터베이스들이 사용가능한지를 판정하기 위해 상호 통신한다. 선택적으로 인코더(10) 및 디코더(50) 둘 다는, 인코딩된 데이터가 인코더(10)로부터 디코더(50)로 전달되기 전에, 상호 대화를 통해, 어느 데이터베이스들이 사용되어야 하는지(예를 들어, 하나 이상의 데이터 서버들)를 정의하는 동작을 하고, 여기서 데이터베이스들은 인코딩된 데이터(30)를 디코더(50)에서 디코딩하여 디코딩된 데이터(60)를 발생하기 위해 이용된다.

[0140] 인코더(10) 및 디코더(50)는 유익하게도 표 3에 제공되는 것과 같은 넓은 범위의 실제 응용들에서, 예를 들어, 다양한 산업들에서 이용된다:

표 3

[0141] 인코더(10) 및 디코더(50)의 실제 응용 분야들

화상 통신	디스플레이	위성 시스템	데스크톱 컴퓨터
보안 시스템	텔레비전	프린터	랩톱 컴퓨터
오디오/영상/비디오 스트리밍 서비스	프로젝터	스캐너, 예를 들어, 문서 스캐너	사고가 발생하기 직전을 기록하는 차량내 사고 기록 장치

의료 및 군사 응용 분야	홀로그래픽 프로젝터	복사기, 예를 들어, 사진 복사기	과학 기기, 예를 들어, 천체 망원경, 광학 현미경
패드 및 휴대폰 등의 모바일 디바이스(무선 통신 디바이스)	디지털 카메라	센서 노드	

- [0142] 유익하게도, 표 3에 열거되어 있는 것과 같은 이러한 실제 응용은 유익하게도 영국 특허 출원 GB1214414.3 및 상응하는 미국 특허 출원 US 13/584,005은 물론, 대응하는 유럽 특허 출원 EP13002521(이들 모두는 참조 문헌으로서 본 명세서에 포함됨)에 기술되어 있는 것과 같은 인코더, 및 영국 특허 출원 GB1214400.2 및 상응하는 미국 특허 출원 US 13/584,047은 물론, 대응하는 유럽 특허 출원 EP13002520(이들 모두는 참조 문헌으로서 본 명세서에 포함됨)에 기술되어 있는 것과 같은 디코더를 이용한다. GB1218942.9(역시 대응하는 유럽 특허 출원 EP13003859을 가짐), GB1303658.7, GB1303661.1, GB1303660.3, GB1312815.2, 및 GB1312818.6 등의 추가의 계류 중인 특허 출원은 그 모두가 참조 문헌으로서 본 명세서에 포함된다.
- [0143] 개요에서, 도 1의 코텍(5)은 인코더(10)로부터 디코더(50)로 전달될 디지털 데이터, 예를 들어, 비디오, 영상, 그래픽 및 오디오 콘텐츠가 종종 대체로 또는 전체적으로 하나 이상의 데이터베이스들(100)로부터 도출되는 데이터베이스 요소들(E)로부터 구성되도록 동작한다. 사용되는 하나 이상의 데이터베이스들(100)은
- [0144] (i) 사용되는 하나 이상의 데이터베이스들(100)이 크기가 변할 수 있음[예를 들어, 인코더(10)로부터 디코더(50)로 전송되고 있는 데이터 콘텐츠의 유형에 기초함];
- [0145] (ii) 사용되는 하나 이상의 데이터베이스들(100)이 디코더(50)에서 요구되는 데이터 재구성 품질에 의존할 수 있음;
- [0146] (iii) 사용되는 하나 이상의 데이터베이스들(100)이 인코더(10)로부터 디코더(50)로 전송될 데이터 크기에 의존할 수 있음;
- [0147] (iv) 사용되는 하나 이상의 데이터베이스들(100)이 인코딩된 데이터(30)를 인코더(10)로부터 디코더(50)로 전송하는 데 이용가능한 대역폭에 의존할 수 있음;
- [0148] (v) 사용되는 하나 이상의 데이터베이스들(100)이 하나 이상의 데이터베이스들(100)과 디코더(50) 및/또는 인코더(10) 사이에서 이용가능한 대역폭에 의존할 수 있음;
- [0149] (vi) 사용되는 하나 이상의 데이터베이스들(100)이 하나 이상의 데이터베이스들(100)과 디코더(50) 및/또는 인코더(10) 사이의 응답 시간에 의존할 수 있음;
- [0150] (vii) 사용되는 하나 이상의 데이터베이스들(100)이 그로부터의 데이터에 액세스하는 비용(즉, 금융 가격)에 의존할 수 있음;
- [0151] (viii) 사용되는 하나 이상의 데이터베이스들(100)이 그로부터 제공되는 데이터에서 발생하는 데이터 오류의 함수에 의존할 수 있음;
- [0152] (ix) 사용되는 하나 이상의 데이터베이스들(100)이 또한 데이터베이스들의 공간적(지리적) 위치에 의존할 수 있고 선택적으로 또한 고려되는 것과 같은 하나 이상의 인자들에 의존하지만,
- [0153] 이러한 선택으로 제한되지 않는다.
- [0154] 더욱이, 인코딩된 데이터(30)에서, 전달되는 하나 이상의 데이터베이스 참조들(R)이 인코딩된 데이터 블록들 대신에 저장되고 송신된다. 인코더(10)로부터 디코더(50)로의 이러한 데이터 전달 방식을 달성하기 위해, 하나 이상의 정적 및 동적 데이터베이스들(110, 120)이 가능한 한 커야만 한다. 보다 큰 데이터베이스는 그의 저장을 위해 보다 많은 메모리 용량을 필요로 하지만, 이러한 보다 큰 데이터베이스는 코텍(5)이 디코딩된 데이터(60)에서 보다 나은 재구성 품질을 달성할 수 있게 해준다. 유익하게도, 코텍(5)은 인코딩된 데이터(30)에서 전달될 주어진 유형의 데이터(예를 들어, 비디오, 영상, 그래픽 및 오디오 콘텐츠)에 대해 가장 적절한 데이터베이스들의 조합을 발견하기 위해, 즉 데이터베이스들(100)의 효율적인 사용 및 대응하는 증가된 압축률을 달성하기 위해, 데이터베이스들(100) 중에서 선택하는 동작을 한다. 그렇지만, 데이터베이스(100)가 보다 크고 보다 많은 요소들(E)을 포함할 때, 데이터베이스 참조 값(R)이 보다 많은 데이터 비트를 필요로 한다는 것을 잘 알 것이다.
- [0155] 하나 이상의 데이터베이스들(100)의 효율적인 사용을 위해, 유익하게도 통상적인 디지털 데이터(예를 들어, 비

디오, 영상, 그래픽 및 오디오 콘텐츠)에 대한 몇개의 정적 데이터베이스들(110)이 발생된다. 인코딩된 데이터(30)에 대해 데이터 압축을 제공하기 위해, 하나 이상의 정적 데이터베이스들(100)이 인코더(10)와 관련하여 사용될 수 있다. 예를 들어, 인코딩될 디지털 데이터 콘텐츠의 유형에 따라 정적 데이터베이스(110)가 선택되고, 정적 데이터베이스들(110)은 그들의 크기에 따라, 사용될 메모리 용량에 따라, 그리고 또한 디코더(50)에서 달 성될 요구된 재구성 품질에 따라 아주 다양할 수 있고; 정적 데이터베이스들(110)은 이용되는 데이터 블록들의 크기, 이용되는 데이터베이스 요소들(E)의 수, 기타와 관련하여 변할 수 있다.

[0156] 인코더(10) 및 디코더(50)는 상기한 바와 같이 결합하여 코덱(5)을 구성하고, 여기서 인코더(10)로부터 디코더(50)로 데이터를 전달할 때 하나 이상의 동적 데이터베이스들(120)이 유익하게도 사용된다. 동적 데이터베이스(120)는 정적 데이터베이스(110)와 비교하여 종종 상이한 요소들(E)을 포함하는데, 그 이유는 동적 데이터베이스(120)의 요소들(E)이 통상적으로, 소스 데이터(20)를 인코딩할 때 정적 데이터베이스(110)에 충분히 적당한 요소들이 없을 때, 인코더(10)에 의해 생성되기 때문이다. 그렇지만, 인코더(10) 및 디코더(50) 둘 다, 상기한 바와 같이, 선택적으로 소스 데이터(20)의 인코딩 및 대응하는 인코딩된 데이터(30)의 디코딩 동안 하나 이상의 동적 데이터베이스들(120)을 생성할 필요가 있을 수 있고, 따라서 인코더(10)와 디코더(50) 사이의 호환성이 보장될 수 있고, 재구성된 디코딩된 데이터(60)가 실질적으로, 예를 들어, 소스 데이터(20)와 일치할 것이다. 하나 이상의 동적 데이터베이스들(120)은 선택적으로 일시적 사용을 위해, 예를 들어, 인코딩된 데이터(30)가 비디오 콘텐츠를 포함할 때 비디오 프레임들의 정의된 구간 동안 생성되고, 더욱이, 하나 이상의 동적 데이터베이스들(120)이 선택적으로, 예를 들어, 매초마다 재생성되고, 그 후에 일정 기간 동안, 즉 소거되기 전 30초의 기간 동안 사용될 수 있다. 이 기간은 선택적으로, 예를 들어, 몇 분짜리의 인코딩된 비디오 콘텐츠에 대해, 수초 내지 전체 비디오 장면 및 영화 챕터의 범위 내에서 변한다. 선택적으로, 본 발명에 따르면, 이전에 생성된 하나 이상의 동적 데이터베이스들에 대해 발생된 요소들(E)로부터 새로운 정적 데이터베이스(110)를 생성하는 것이 가능하고; 다른 형태들에서, 생성된 동적 데이터베이스들(120)은 그의 내용이 일정하게 될 수 있고, 그 후에 정적 데이터베이스들(110)로서 기능하도록 동작할 수 있다.

[0157] 본 개시 내용에 따른, 즉 인코더(10)를 디코더(50)와 함께 포함하는 코덱(5)은, 어느 유형의 데이터베이스(100)가 이용되는지(정적 데이터베이스인지 또는 동적 데이터베이스인지)에 관계없이, 인코딩된 데이터(30)(예를 들어, 비디오, 영상, 그래픽 및 오디오 콘텐츠)에 대응하는 데이터를 저장하고 전송할 때, 처리 시간(즉 요구되는 처리 자원 및 데이터 메모리 용량)을 감소시키는 동작을 할 수 있다. 이러한 이점은 인코딩된 데이터(30)가 데이터 통신 네트워크(예를 들어, 무선 통신 네트워크 및 인터넷)를 통해 전달되어야 할 때 아주 중요한데, 그 이유는 통신 네트워크가 인코딩된 데이터(30)를, 예를 들어, 실시간 스트리밍 방식으로 대처하는 것이 덜 곤란하기 때문이다.

[0158] 공개된 미국 특허 출원 US2010/322,301에 기술된 바와 같은 영상 발생기와 비교하여, 도 1의 코덱(5)은 인코딩된 데이터(30)에서 더 높은 정도의 데이터 압축을 제공할 수 있다. 개선된 압축은 참조 값들(R)을 송신하는 것은 물론 영역 식별자들(U)에 관계된 정보 둘 다에 관련되어 있다. 본 개시 내용과 관련하여 영역 식별자들(U)은 주어진 블록이 대응하는 데이터 중의(예를 들어, 원래의 영상 중의, 원래의 센서 데이터 중의, 기타 중의) 어디에서 도출되는지의 공간적 정보를 전달하는 데 이용된다. 유익하게도, 영역 식별자(U)는 데이터에서의 데이터 블록의 크기, 형상 및 위치를 나타낸다. 선택적으로, 영역 식별자(U)가 대응하는 블록과 함께 송신되지만, 이것은 종종 너무 비효율적인 방법이다. 유익하게도, 본 개시 내용의 실시예들을 구현할 때, 단지 영역 식별자들(U) 자체를 전달하는 것에 대한 대안으로서, 영역 식별자들(U)에 관한 정보가 선택적으로 전달된다. 예를 들어, 영역 식별자들(U)은 선택적으로 영역 식별자들(U)에 관한 정보(예를 들어, 분할/결합 정보)에 기초하여 디코딩 동안 인코딩된 데이터로부터 획득된다. 보다 선택적으로, 영역 식별자들(U)은 사전 설정될 수 있고, 어떤 다른 정보도 전달될 필요가 없다. 유익하게도, 데이터베이스는 영역 식별자들(U)을 결코 편집하거나 폐치하지 않고, 그 대신에 참조 값(R)에 기초하여 데이터베이스로부터 검색된 요소들(E)이 취해지고, 그들의 재구성된 데이터 블록들이 재구성된 결과 데이터 내에서 영역 식별자(U)가 가리키는 위치에 배치된다.

[0159] 때때로, 영역 식별자들(U)에 관한 정보가 데이터에 사전 설정되어 있고(예를 들어, 블록들이 똑같은 크기로 되어 있음), 이들이 특정의 순서로 코딩되어 있으며; 이 경우에 어떤 실제의 새로운 정보가 전송될 필요가 없다. 들어오는 인코딩된 데이터 블록에 대한 영역 식별자들(U)을 그들의 순서에 따라 발생하는 사전 설정된 방식을 변경하기 위한 추가의 정보가 영역 식별자들(U)에 관한 데이터에 없기 때문에, 사전 설정된 방식이 이용되는 것은 자명하다.

[0160] 가끔, 선택적으로 하나 이상의 원하는 블록 크기들 및 처리 순서를 나타내는 정보가 전송되거나, 몇가지 대안들 중에서 선택된 하나 이상의 블록 크기들 및 처리 순서가 전송된다. 때때로, 인코더는 또한 데이터 블록들의 크

기, 형상 및 위치를 정의하는 이러한 분할/결합 정보 그리고 또한 그들의 처리 순서를 디코더로 전송할 수 있고, 따라서 수신된 그리고 디코딩된 데이터 블록(예를 들어, 데이터베이스로부터 검색된 데이터 블록)은 항상 재구성된 결과 데이터에서의 올바른 위치에 분명하고 일의적으로 배치될 수 있다.

[0161] 영역 식별자들(U)을 정의하는 이런 종류의 분할/결합 정보는, 예를 들어, 미국 특허 출원 US 13/584,005 및 US 13/584,047에 제시되어 있는 코딩 해결 방안을 사용하여 전송될 수 있다.

[0162] 영역 식별자들(U)은 선택적으로, 이용되는 코딩 방법에 따라, 많은 방식들로 발생된다. 선택적으로, 참조 문헌으로서 본 명세서에 포함되어 있는, 상기한 바와 같은 블록 인코더/디코더에서, 영역 식별자 정보(U)는 블록들을 분할/결합시킬 때 분할/결합 정보로부터 도출되고, 때때로 영역 식별자 정보(U)는 이미 이용가능하거나 알려져 있으며, 이 경우에 영역 식별자들(U)에 관련된 정보는 단순히 이미 이용가능하고 및/또는 알려져 있는 영역 식별자(U)의 정보이다.

[0163] 때때로, 전송 및 저장에서 사용되는 방법으로 인해, 인코딩된 데이터의 순서가 보장되지 않을 수 있다. 그 때문에, 선택적으로 실제의 데이터 인코딩 및 데이터 디코딩에 필요한 것보다 더 많은 영역 식별자들(U)이 데이터에서 이용된다. 더욱이, 하나의 영역 식별자(U)가 선택적으로, 예를 들어, 각각의 데이터 청크에 삽입되어, 그 데이터 청크에 대응하는 제1 데이터 블록이 어디에 위치해 있는지를 표현한다. 이와 유사하게, 때때로 데이터 청크의 순서를 표시하기 위해 선택적으로 각각의 데이터 청크에 서수가 부가된다. 이 정보는 이어서 부가의 영역 식별자들(U)을 필요 없게 해준다. 이 발명은 선택적으로, 예를 들어, 미국 특허 US 8,169,547에 기술된 것과 같은 변경된 및 변경되지 않은 데이터 블록들을 사용하는 해결 방안과 함께 사용된다. 변경된 데이터 블록들만을 전달하는 해결 방안이 있는 경우, 그 변경된 블록들의 위치를 나타내기 위해 영역 식별자들(U)이 사용될 필요가 있으며, 변경된 데이터 블록 위치들을 일의적으로 나타내는 다른 정보가 없는 경우에 특히 그렇다.

[0164] 도 2에 예시된 바와 같이, 인코더(10)로부터 디코더(50)로 동적 데이터베이스(120)의 새로운 요소(E)(200)를, 그의 대응하는 참조 값(R)(210)을 새로운 요소(200)와 함께 송신할 필요 없이, 전달함으로써 이러한 향상된 데이터 압축이 달성된다. 이러한 특성은, 디코더(50) 및 인코더(10) 둘 다에서, 디지털 데이터 블록들(220)의 요소들(E)(200)에 대한 모든 참조 값들(R)이 재구성된 데이터 블록(220')으로부터 컴퓨팅 하드웨어(230, 230R)를 사용하여 재계산될 가능성이 있는 것에 의해 달성된다. 그에 따라, 요소(E)(200)가 인코더(10)로부터 디코더(50)로 송신되는 인코딩된 데이터(30)의 일부에 불과하다는 것을 잘 알 것이다. 선택적으로, 참조 값(R)의 일부는 유익하게도 그렇지 않았으면 인코더(10) 및 디코더(50) 둘 다에 존재할[예를 들어, 몇개의 데이터베이스 요소들(E)이 하나 이상의 데이터베이스들(100)로부터 이미 이용가능한지로부터 도출되는] 또는 그렇지 않았으면(디코딩 후에) 인코딩된 데이터 블록에 기초하여(재)계산할 수 있는 정보를 포함하고 있다. 양/개수에 부가하여, 예를 들어, 요소들(E)의 확률 순서가 참조 값(R)의 수치 부분에 영향을 줄 수 있다. 더욱이, 상기한 양/개수 또는 확률 순서가 항상 전체 데이터베이스를 참조할 필요는 없고, 그 대신에, 예를 들어, 전송된 참조 값들(R)의 선택된 조합을 참조할 수 있고, 이 조합에 대해 따라서 이 값의 도움을 받아 2개 이상의 상이한 변동이 주어질 수 있다.

[0165] 인코더(10) 및 디코더(50)는 유익하게도 지금까지 모르고 있던 참조 값들(R)을 계산하는 방법을 이용한다. 유익하게도, 인코더(10)는 대응하는 참조 값들(R)을 계산하기 전에 데이터 블록 값들을 재구성(즉, 양자화 및 역양자화)하는 동작을 한다. 이 때문에, 각각의 데이터 블록(220)은, 예를 들어, 표 2에 제공된 것과 같은 임의의 공지된 손실 또는 무손실 압축 알고리즘을 사용하여, 또는 이러한 손실 또는 무손실 압축 알고리즘들의 조합에 의해 인코더(10)에서 인코딩될 수 있고; 도 1의 코덱(5)은 그가 이용하는, 주요 데이터 압축 이점들을 제공하는 압축 알고리즘들과 관련하여 이와 같이 다양할 수 있다. 이들 방법은 DC, 슬라이드(slide), 라인(line), 다중 레벨(multilevel), DCT, 보간 및 외삽을 포함하지만, 이들로 제한되지 않는다. 데이터베이스 요소들(200)은 또한, 예를 들어, 임의의 블록 크기에 대해 그리고 디코딩된 영상의 임의의 위치로부터 생성될 수 있다. 유익하게도, 참조 값들(R)은 독립적으로 계산될 수 있는 복수의 성분 부분들을 포함하고 있다 - 예를 들어, 하나의 성분은 블록 값들의 평균을 나타내고, 다른 성분은 블록 값들의 분산을 나타내며, 또 다른 성분은 블록 값들의 진폭을 나타내고, 또 다른 성분은 블록 값들에 대한 체크섬 또는 주어진 데이터베이스(100)에 포함되어 있는 요소들(E)을 나타낸다 -. 더욱이, 전체 참조 값(R) 또는 참조 값(R)의 일부는 또한 선택적으로 인코딩된 데이터(30)와 함께 디코더(50)로 전달되고; 이러한 정보는 유익하게도 전달의 정확성 - 예를 들어, 인코딩된 데이터(30)를 인코더(10)로부터 디코더(50)로 전송하는 것으로부터 발생하는 임의의 품질 열화 - 을 평가하기 위해 이용된다. 선택적으로, 디코더(50)는, 예를 들어, 유효성 확인을 위해 및/또는 품질 관리를 위해, 전체 참조 값(R) 또는 그의 일부를 인코더(10)로 다시 반환하는 동작을 한다.

- [0166] 주어진 데이터베이스(100)가 보다 클 때, 상기한 데이터베이스 참조 값들(R)은 보다 많은 데이터 비트들을 필요로 한다. 하나의 고유의 데이터베이스 참조 값(R)을 정의하는 데 필요한 압축되지 않은 데이터 비트들의 수는 데이터베이스(100) 내의 요소들(E)(200)의 수의 \log_2 값을 취함으로써 계산될 수 있다. 이와 같이, 보다 큰 데이터베이스 참조 값(R)은 보다 많은 비트들을 필요로 할 것이고, 각각의 기존의 데이터베이스 데이터 블록에 대해 인코딩된 데이터(30)에서 달성가능한 압축비가 감소된다. 보통, 데이터베이스 참조 값(R)에서, 유익하게도 미사용 데이터 요소들(E)에 대해 예약되어 있는 하나 이상의 비트들이 있다. 그렇지만, 큰 정적 데이터베이스(110) 또는 동적 데이터베이스(120)에서 과도한 미사용 데이터베이스 요소들(E)을 갖는 것을 피하기 위해, 새로운 보다 작은 동적 데이터베이스(120)의 요소들(E)이 사용된 정적 데이터베이스(110) 또는 사용된 동적 데이터베이스(120)의 요소들(E)로부터 생성될 수 있다. 새로운 보다 작은 동적 데이터베이스(120)가 이어서 디코더(50)에서 디코딩을 위해 사용될 수 있고, 여기서 새로운 동적 데이터베이스는, 보다 큰 원래의 데이터베이스와 비교하여, 새로운 작은 동적 데이터베이스에 존재하는 요소들(E)을 일의적으로 식별하기 위한 그의 참조 값들(R)에서 보다 적은 비트들을 필요로 한다.
- [0167] 유의할 점은, 여기서 언급된 참조 값(R)이 엄밀히 말하면 전송된 참조 값(데이터베이스에서의 요소(E)를 일의적으로 명시하는 인덱스)이고, 따라서 원래의 전송된 참조 값이 이전의 보다 큰 데이터베이스에서 사용하는 것보다 새로운 보다 작은 데이터베이스에서 보다 작은 비트 카운트를 받는다는 것이다. 이것은 물론 검색에서 사용되는 실제의 정확한 참조 값(R)에 어떤 영향도 미치지 않는데, 그 이유는 그것이 여기서 결코 변경되지 않는 요소(E)의 실제의 데이터 값에 의존하기 때문이다.
- [0168] 새로운 동적 데이터베이스에 대한 참조 값(R)은, 예를 들어, 새로운 데이터베이스에 존재하는 요소들(E)의 총수일 수 있다. 선택적으로, 새로운 동적 데이터베이스에서의 데이터베이스 요소(E)에 대응하는 이 참조 값(R)은 실제의 데이터를 포함하거나, 다른 보다 큰 데이터베이스에 링크되어 있을 수 있다. 새로운 동적 데이터베이스가 참조 값들(R)을 코딩하는 데 사용될 때, 참조 값들(R)이 공지된 코딩 알고리즘, 예를 들어, 공지된 델타 코딩을 이용함으로써 효율적으로 압축될 수 있다는 것을 알았다. 이와 같이, 새로운 동적 데이터베이스를 사용하여, 디코더(50)로부터 출력되는 디코딩된 데이터(60)의 품질 또는 성능이 실질적으로 감소하는 일 없이, 상당한 미압축된 참조 값(R) 비트들을 절감하는 것이 가능하다.
- [0169] 한 예로서,
- [0170] (a) 주어진 정적 데이터베이스(110)가 대응하는 데이터 값들(R)에 대해 24 비트의 데이터를 사용하여 표현가능한 1600만개의 요소들(E)을 포함하는 경우;
- [0171] (b) 동적 데이터베이스(120)가 대응하는 데이터 값들(R)에 대해 10 비트의 데이터를 사용하여 표현가능한 단지 1024개의 요소들(E)을 포함하는 경우; 및
- [0172] (c) 특정의 데이터 블록 인덱스를 갖는 데이터 블록이 동적 데이터베이스(120)에 존재하는 경우,
- [0173] 참조 값(R)은, 데이터 블록이 정적 데이터베이스(110)에 포함되어 있는 경우 그 데이터 블록을 일의적으로 정의하는 24 비트와 비교하여, 동적 데이터베이스(120)를 통해 데이터 블록을 일의적으로 정의하기 위해 단지 10 비트를 필요로 할 것이다. 이 예에서 주어진 데이터 블록이 정적 데이터베이스(110)로부터 동적 데이터베이스(120)로 전송되는 경우, 그 데이터 블록을 일의적으로 식별하기 위해 보다 적은 비트들이 필요할 것이고; 유익하게도 인코더(10)는 정적 데이터베이스(110)로부터 동적 데이터베이스(120)로의 하나 이상의 데이터 블록들의 이동 또는 복사에 관해 디코더(50)에 통보한다. 도 1의 코덱(5)은 유익하게도 또한 선택적으로 하나 이상의 데이터 블록들에 대해 일어나는 데이터베이스 사용의 양에 따라 하나 이상의 데이터 블록들이 정적 데이터베이스(110)로부터 동적 데이터베이스(120)로 이동 또는 복사되는 방식으로 동작할 수 있다. 환언하면, 인코딩된 참조 값들(R)이 주어진 데이터 블록을 빈번히 참조할 때, 디코더(50)에서의 데이터 블록은 유익하게도, 디코더(50)로 전송된 또는 디코더(50)에 의해 액세스가능하게 되어 있는 정적 데이터베이스(110)로부터보다는, 디코더(50)로 전송된 또는 디코더(50)에 의해 액세스가능하게 되어 있는 주어진 동적 데이터베이스(120)로부터 획득된다. 상기한 바와 같이, 데이터베이스들(110, 120)은 선택적으로, 예를 들어, 하나 이상의 제3자들에 속하는 하나 이상의 데이터 서버들에 의해 호스팅된다.
- [0174] 도 1의 코덱(5)은 보다 큰 정적 또는 동적 데이터베이스들(100)의 가장 많이 사용되는 데이터베이스 요소들(E)로부터 작은 크기의 정적 또는 동적 데이터베이스를 생성함으로써 데이터베이스 참조 값들(R)에 대한 비트들을 절감할 수 있게 해준다. 코덱(5)에서, 보다 큰 크기의 동적 데이터베이스들(120) 및/또는 정적 데이터베이스들(110)은 모든 기존의 데이터 블록들에 대한 사용 카운터를 유지하고, 이러한 사용 카운터 정보의 기초로서 요소

들(E)을 선택함으로써 새로운 데이터베이스들이 생성된다. 이러한 보다 작은 데이터베이스는 유익하게도 인코더(10)에서의 인코딩 동안 생성되고, 이어서 인코딩된 데이터(30)를 디코딩하기 위해 디코더(50)로 전달되거나 다른 방식으로 디코더(50)에 의해 이용가능하게 되며; 다른 방식으로 이용가능하게 되는 이러한 보다 작은 데이터베이스는, 예를 들어, 원격 인터넷-기반 서버로부터 및/또는, 예를 들어, 클라우드 컴퓨팅 환경에서 호스팅되고 많은 이러한 디코더들(50)에 의해 액세스가능한, 다른 대안으로서, 디코더(50)의 LAN과 유사한 LAN에서 호스팅되는, 클라이언트 LAN에 있는 데이터베이스 서버들로부터 디코더(50)로 다운로드하기 위해 이용가능하게 되는 보다 작은 데이터베이스에 대한 URL 링크를 포함하는 인코딩된 데이터(30)를 포함할 수 있다.

[0175] 소스 데이터(20)를 인코딩할 때, 인코더(10)는, 대응하는 인코딩된 데이터(30)가 디코더(50)에 수신될 때 이를 디코딩하기 위해, 매 순간마다 몇개의 상이한 데이터베이스들(100), 어떤 종류의 데이터베이스들(100) 및 어느 특성의 데이터베이스들(100)이 사용되어야 하는지를 선택하는 동작을 하고, 여기서 인코더(10)에 의해 행해지는 데이터베이스(100)의 선택을 나타내는 정보가 디코더(50)로 전달된다. 이 정보는 로컬적으로(즉, 응용 프로그램 의존적 데이터베이스 정보임) 또는 전역적으로(즉 항상 동일한 데이터베이스 정보임) 제공될 수 있다. 특성의 상황에서, 선택된 데이터베이스는 디코더(50)가 이미 알고 있는 것이고, 이러한 경우에, 특성의 정보가 전달, 저장 및/또는 전송될 필요가 없다 - 즉, 그 정보가 또한 디코더(50)가 이미 알고 있는 어떤 것일 수 있다 - . 편리하게도, 데이터베이스(100)의 선택은 물론 사용되는 인코딩 방법의 선택을 나타내는 정보가 인코딩된 데이터(30)의 비디오 헤더 또는 컨테이너에 저장된다. 더욱이, 디코더(50)는 선택적으로, 인코딩된 데이터(30)를 디코딩하는 작업을 실행하기 전에, 하나 이상의 정적 데이터베이스들(110) 모두를 갖추고 있다. 디코더(50)가 인코딩된 데이터(30)에 정의되어 있는 주어진 데이터베이스를 갖고 있지 않다는 것을 디코더(50)가 발견한 경우, 디코더(50)는, 예를 들어, 원격 네트워크-연결된 데이터 서버로부터 디코더(50)로 송신되어야 하는 또는 다른 방식으로 이용가능하게 되어야 하는 누락된 주어진 데이터베이스에 대한 요청을 인코더(10)로 송신한다. 선택적으로, 원격 서버는 디코더(50)에 비교적 가까이에, 예를 들어, 상호 유사한 LAN 네트워크 내에 위치해 있고, 그로써 디코더(50)로의 누락된 데이터베이스(100)의 고속 전달을 제공한다. 선택적으로, 누락된 데이터베이스는 압축된 형식으로 디코더(50)에 제공되고, 따라서 디코더(50)는 디코더(50)의 데이터 메모리에 저장하기 위해 누락된 데이터베이스를 압축 해제하는 동작을 한다. 유익하게도, 디코더(50)에 의해 가장 빈번히 이용되는 데이터 블록들은 유익하게도 작은 정적 데이터베이스(110)에 저장된다.

[0176] RAM으로부터의 자원들을 해제시키기 위해, 인코더(10) 또는 디코더(50) 또는 둘 다에서, 미사용 데이터베이스 요소들이 유익하게도 제거된다. 데이터베이스 요소들(E)의 사용이 유익하게도 데이터베이스에서의 사용된 요소들(E)의 수를 카운트하는 것에 의해 또는 데이터베이스 요소들(E)이 마지막으로 사용된 때에 대한 통계를 유지하는 것에 의해 측정된다. 도 1의 코텍(5)에서, 미사용 요소들(E)을 제거하기 위한 소거 목적을 위해 모든 동적 데이터베이스들(120)이 처리된다. 선택적으로, 정적 데이터베이스들(110)도 역시 소거 처리되고; 다른 대안으로서, 정적 데이터베이스들(110) 중 하나 이상이 소거 목적을 위해 주기적으로 새로운 정적 데이터베이스들로 대체된다. 선택적으로, 이전의 데이터베이스들(110)이 또한 백업 메모리로 전송될 수 있고, 따라서 이들은 인코더(10) 및/또는 디코더(50)로 반환하기 위해 필요한 경우 나중에 액세스될 수 있지만, 정적 데이터베이스들(110)에서의 메모리 용량을 차지하지는 않으며; 유익하게도, 백업 메모리는 하드 드라이브, 플래시 메모리, 외부 메모리, 광 디스크 데이터 ROM, 클라우드-기반 데이터 저장소 등을 통해 구현된다. 동작을 설명하면, 디코더(50)는 유익하게도, 예를 들어, 디코더(50)가 임의의 이러한 특이 사항(aberration)을 해결하기 위해 그의 데이터베이스들(100)의 업데이트를 요청할 수 있게 해주기 위해, 데이터베이스들(100) 중 하나에서의 임의의 특이한 변화를 통지받는다. 유익하게도, 데이터베이스들(100)은 또한 고유 식별자(예를 들어, 버전 번호)에 의해 식별된다. 데이터베이스들(100) 중 하나 이상인, 예를 들어, 인코더(10)에서 변경되는 경우, 전체 데이터베이스들(100)을 디코더(50)로 전송하는 것보다는 데이터베이스들(100)에서의 변화만이 디코더(50)로 전달되는 것이 바람직하다. 선택적으로, 하나 이상의 데이터베이스들(100)이 인코더, 예를 들어, 공지된 유형의 인코더에 의해 업데이트될 수 있고; 더욱이, 인코더(10)는 선택적으로, 예를 들어, 고상 ROM에 저장되어 있는, 연관된 사전 로드된 정적 데이터베이스(110)를 구비하고 있다. 게다가, 하나 이상의 동적 데이터베이스들(120)이 디코더(50)에서 업데이트될 때, 상이한 업데이트 모드들이 이용될 수 있다. 예를 들어,

[0177] (i) 제1 업데이트 모드에서, 이전의 요소(E)는 새로운 요소(E)에 의해 덮어쓰기되고;

[0178] (ii) 제2 업데이트 모드에서, 이전의 요소(E)는 보유되고, 하나 이상의 동적 데이터베이스들(120)의 크기를 증가시키기 위해 새로운 요소(E)가 추가된다.

[0179] 선택적으로, 작은 동적 데이터베이스(120)가 다 채워져 있을 때[즉, 그의 모든 요소들(E)이 할당되어 있을 때] 작은 동적 데이터베이스(120)는, 작은 동적 데이터베이스(120)의 어느 요소들(E)이 가장 빈번히 사용되는지를

관정하기 위해, 컴퓨팅 하드웨어 상에서 실행 중인 하나 이상의 자동화된 소프트웨어 도구들을 사용하여 검사되고, 이어서 데이터베이스(120) 내의 요소들(E)이 정렬되며, 따라서 가장 빈번히 사용되는 요소들(E)은, 예를 들어, 데이터베이스(120)를 보다 빈번히 사용되는 요소들(E)에 대해 액세스하기 보다 빠르도록 하기 위해 및/또는, 예를 들어, 데이터 통신 네트워크(예를 들어, 인터넷 또는 무선 데이터 통신 네트워크)를 통해 인코더(10)로부터 하나 이상의 디코더들(50)로의 전달을 위해, 데이터베이스(120)를 보다 효율적으로 압축가능하도록 하기 위해, 데이터베이스(120)의 처음으로 이동되고; 유익하게도, 작은 동적 데이터베이스(120)의 다른 덜 빈번히 사용되는 요소들(E)은 해제된다. 인코더(10) 및 디코더(50) 둘 다는 유익하게도, 하나 이상의 데이터베이스들(120)이 채워져 있을 때, 이러한 검사 및 정렬을 수행한다. 선택적으로, 이러한 검사 및 정렬은 인코더(10)로부터 송출되는 명령어에 의해 개시된다.

[0180] 데이터 저장 또는 데이터 전송이 초기화될 때, 인코더(10)는 데이터베이스들(100) 중 어느 것이 디코더(50)에 의해 차후에 사용되는지[예를 들어, 인코딩된 데이터(30)를 디코딩하기 시작할 시에 데이터베이스들(100)이 비어 있는지 여부 또는 인코딩된 데이터(30)를 디코딩하기 시작할 시에 데이터베이스들(100)이 이미 어떤 데이터베이스 요소들(E)을 포함하고 있는지]를 선택하는 동작을 한다. 예를 들어, 디코더(50)는 선택적으로, 작은 일반 정적 데이터베이스(110)를 이용하여, 인코딩된 데이터(30)를 디코딩하기 시작하고, 여기서 디코더(50)는 인코더(50)가 인코딩된 데이터(30)에 포함되어 있는 데이터 블록들로부터 관련된 누락된 요소들(E)로 동적 데이터베이스(120)를 계속하여 채우고; 동적 데이터베이스(120)는 이어서 유익하게도 이상에서 정의된 바와 같은 규칙을 사용하여 필요 시에 소거된다. 선택적으로, 발생한 동적 데이터베이스(120)는 디코더(50)에 수신되는 장래의 인코딩된 데이터(30)를 디코딩할 때의 장래의 사용을 위해 디코더(50)에 의해 새로운 정적 데이터베이스(110)로서 저장된다.

[0181] 소스 데이터(20)(예를 들어, 비디오, 영상, 그래픽 및 오디오 콘텐츠 중 대응하는 적어도 하나)를 인코딩할 때, 데이터 블록을 포함하는 데이터베이스가 디코더(50)에 아직 존재하지 않는 경우라도, 예를 들어, 디코더(50)가 주어진 데이터 블록을 나타내는 요소(E)를 포함하는 적당한 정적 데이터베이스(110)를 디코더(50)에 로컬적으로 이용가능한[예를 들어, 디코더(50)와 유사한 LAN, ISP 또는 다른 서버 상에 위치해 있는] 어떤 정적 데이터베이스 저장소로부터 폐지할 수 있다는 것을 인코더(10)가 알고 있는 경우에, 인코더(10)는 선택적으로 주어진 데이터 블록의 데이터베이스 요소 참조 값들(R)을 송신한다. 이러한 방식은 하나의 인코더(10)가 동일한 인코딩된 데이터(30)를 약간 다른 데이터베이스들을 이용할 수 있는 많은 디코더들(50)로 전달할 수 있게 해준다. 이러한 방법이 이용될 때, 디코더들(50)은 큰 정적 데이터베이스들 또는 관련 리소스들을 호스팅할 필요가 없다. 이와 같이, 이러한 방법이 이용될 수 있는 경우, 인코더(10)는 유익하게도 디코더(50)와 협상한다. 이 방법이 성공하기 위해, 인코더(10)는, 인코더(10)로부터 송신된 인코딩된 데이터(30)의 디코딩을 실행하기 위해 디코더(50)가 필요로 하는 하나 이상의 데이터베이스들(100)을 디코더(50)에 제공하기 위해, 디코더(50)가 액세스할 수 있는 공개 및/또는 비밀 데이터베이스를 알고 있을 필요가 있다. 이 방법은 인코딩된 데이터 블록들을 전혀 송신할 필요가 없을 때(즉, 다른 인코더로부터 디코더(50)로의 데이터 블록들의 이러한 공급이 가능할 때) 아주 적합하고, 여기서 다른 인코더는 인코딩된 데이터를 앞서 인코딩하였다. 선택적으로, 데이터 블록들은 피어-투-피어 통신 방식으로 다른 인코더들로부터 전달된다. 더욱이, 이 방법은 인터넷을 통한 인코딩된 비디오 데이터의 라이브 스트리밍에 적합하고, 그에 의해 인코딩된 비디오 데이터가 디코더(50)로부터 멀리 떨어진 곳으로부터 스트리밍될 수 있게 해주며, 여기서 데이터 블록들은 디코더(50)에 로컬적으로 공급받을 수 있고; 이러한 경우에, 실질적으로 참조 값들(R)만이 디코더(50)로부터 멀리 떨어져 있는 위치로부터 전달된다.

[0182] 상기한 바와 같이, 하나 이상의 동적 데이터베이스들(120)은 유익하게도 인코더(10)에서 또는 디코더(50)에서 재구성된 데이터 블록들로부터 발생되거나 복사된 데이터베이스 요소들(E)로부터 구성된다. 데이터 블록들은 1D, 2D 또는 3D, 또는 이들의 임의의 조합일 수 있다. 선택적으로, 데이터 블록들은, 예를 들어, 디스플레이 상에서 볼 때, 몇가지 예를 들자면, 예를 들어, 다각형, 직사각형, 타원형, 원형, 장방형, 삼각형 형태의 크기 및 형상일 수 있다.

[0183] 선택적으로, 1D 데이터 블록과 관련하여, 그의 형상은 디코딩된 데이터(60)에서 제시될 때 1차원 데이터 항목들의 집합일 수 있다. 더욱이, 선택적으로, 2D 데이터 블록과 관련하여, 그의 형상은 디코딩된 데이터(60)에서 제시될 때 정사각형, 직사각형, 삼각형, 평행사변형 또는 원일 수 있다. 게다가, 3D 데이터 블록과 관련하여, 그의 형상은 정육면체, 피라미드, 원통, 볼 등과 같이 보다 다형적(polymorphous)일 수 있고; 이러한 3D 데이터 블록은, 예를 들어, 디코딩된 데이터(60)가 3D 그래픽 디스플레이를 위한 것일 때, 예를 들어, 사용자가 보기 위한 3D 영상을 전달하는 데 유익하다. 통상적으로 이용되는 2D 데이터 블록은 유익하게도 본 개시 내용의 실시예들에서 이용되는 8x8 픽셀들 또는 유사한 디스플레이 요소들에 대응하고; 인코딩된 데이터(30)를 발생할 때

인코더(10)에 의해 이용될 수 있는 8x8 픽셀 블록 크기에 대한 다수의 상이하고 아주 효율적인 공지된 코딩 방법들이 있다.

[0184] 본 개시 내용을 더 상세히 설명하기 위해, 인코더(10)에서 데이터 블록들을 인코딩하는 것에 대해 이제부터 설명할 것이다.

[0185] 단계 1: 이상적인 상황에서, 소스 데이터(20)에 존재하는 대응하는 주어진 데이터 블록에 대해[예를 들어, 소스 데이터(20)에 존재하는 영상 및/또는 오디오의 일부분에 대해] 주어진 데이터베이스 참조 값(R)이 계산된다. 하나 이상의 데이터베이스들(100)에 적당한 기존의 요소(E)가 있는 경우[즉, 주어진 데이터 블록에 대해 하나 이상의 데이터베이스들(100)에서의 기존의 요소(E)와 일치하는 것이 발견될 수 있는 경우], 하나 이상의 데이터베이스들(100)로부터 기존의 요소(E)를 선택하기 위해 참조 값(R)이 계산된다. 주어진 데이터 블록과 기존의 데이터 블록 간의 오차가 정의된 임계값 미만인 것에 따라 일치가 판정되고, 여기서 정의된 임계값은 인코딩 품질 지수(encoding quality index)에 대응한다.

[0186] 단계 2: 단계 1에서 일치가 이루어진 경우, 선택된 기존의 데이터 블록에 대한 참조 값(R)이, 예를 들어, 인코딩된 데이터(30)에 저장되거나 인코딩된 데이터(30)를 통해 송신된다. 선택된 기존의 데이터 블록은 복수의 데이터 블록들(100)로부터 검색될 수 있고, 여기서 저장되거나 송신되는 참조 값(R)은 또한 데이터베이스(100)의 식별자를 포함하며, 선택된 기존의 데이터 블록은 디코더(50)에 의해 발견될 수 있다. 이와 달리, 단계 1에서의 적합한 일치 발견될 수 없는 경우, 주어진 데이터 블록은, 예를 들어, DC 인코딩, 슬라이드 인코딩, 다중 레벨 인코딩, DCT 인코딩 또는 임의의 다른 적당한 인코딩 알고리즘을 사용하여 인코딩되고, 재구성된 데이터 블록에 대해 데이터베이스 참조 값(R)이 계산되며; 인코딩된 데이터 블록은 유익하게도, 예를 들어, 디코더(50)에서 수신하도록, 인코딩된 데이터(30)에 포함된다.

[0187] 다른 대안으로서 또는 그에 부가하여, 인코더(10)는, 소스 블록에 대해서와 유사하게, 재구성된 데이터 블록에 대해 이용가능한 적당한 요소(E)가 있는지 여부를 검사하는 동작을 하고; 이러한 경우에, 상이한 인코딩 품질 임계값이 이용될 수 있다. 재구성된 데이터 블록을 나타내기 위해 특정의 요소(E)가 선택되는 경우, 그의 참조 값(R)은, 예를 들어, 인코딩된 데이터(30)에 저장되거나 인코딩된 데이터(30)를 통해 송신될 수 있다. 다른 대안으로서, 재구성된 데이터 블록(즉, 재편성된 데이터 블록)을 나타내기 위해 적당한 데이터베이스 요소(E)가 선택될 수 없는 경우, 재구성된 데이터 블록 또는 소스 데이터(20)에서의 주어진 데이터 블록이 재구성된 데이터 블록(즉, 재편성된 데이터 블록)에 가장 적합한 인코딩 알고리즘을 사용하여 송신되어야만 한다. 유익하게도, 인코딩된 데이터(30)에 보다 적은 비트들을 발생하거나, 보다 적은 오류를 생성하거나, 예를 들어, RD 최적화에 대응하는 어떤 람다 계수(lambda coefficient)를 사용함으로써, 생성된 오류에 대해 사용된 비트들을 최적화하는 방법이 선택된다. 이 인코딩된 데이터 블록은 이어서 인코더(10)에서 및/또는 디코더(50)에서 데이터를 인코딩할 때의 장래의 사용을 위해 동적 데이터베이스들(120) 중 하나 이상에 부가될 수 있다.

[0188] 선택적으로, 인코더(10)는 유럽 특허 EP 1787262 및 유럽 특허 출원 EP12173534.4(참조 문헌으로서 본 명세서에 포함됨)에 기술되어 있는 것과 같은 인코딩 기법들을 이용하고; 예를 들어, 비디오 콘텐츠를 인코딩하기 위해, 기술된 바와 같은 이들 기법을 이용할 때, 변하지 않은 데이터 블록을 나타내는 색상이 데이터베이스(100)에서 발견될 수 있거나, 변하지 않은 블록들이 검출되고 상이하게 코딩될 수 있다. 변화된 데이터 블록들만이, 예를 들어, 공지된 코딩 기법들(예를 들어, 표 2에 제공된 것 등)을 사용하여 코딩을 필요로 하고; 표 2에 부가하여, 다른 종류의 코딩 방법들이 선택적으로 이용되는데, 그 이유는 표 2가 전체적인 코딩 해결 방안들만을 기술하고 있고 보간 및 외삽, 다중 레벨, DCT 등과 같은 개별 방법들은 기술하고 있지 않기 때문이다. 유럽 특허 EP 1787262 및 유럽 특허 출원 EP12173534.4(상기한 바와 같이, 참조 문헌으로서 본 명세서에 포함됨)에 기초한 이 예에서의 성능을 최적화하기 위해, 유사한 기능이 인코더(10) 및 디코더(50) 둘 다에 제공될 수 있다. 다른 관련된 특허 출원들(즉, 2013년 6월 17일자로 출원된 유럽 특허 출원 EP13172237.3)의 내용이 또한 참조 문헌으로서 본 명세서에 포함되어 있다.

[0189] 본 개시 내용의 실시예들을 추가로 설명하기 위해, 데이터베이스들(100) 및 상기한 참조 값들(R)에 대해 이제부터 더 상세히 기술할 것이며; 예를 들어, 공개된 미국 특허 출원 US 2010/322,301(참조 문헌으로서 본 명세서에 포함됨)을 참조한다. 상기한 바와 같이, 정적 데이터베이스(110) 및 동적 데이터베이스(120)와 같은 2가지 유형의 데이터베이스가 있다. 정적 데이터베이스(110)는 임의의 유형의 메모리에 저장될 수 있고; 보다 빠른 성능을 위해, 정적 데이터베이스(110)는 선택적으로 ROM에 저장된다[예를 들어, ROM은 실리콘 집적 회로에 기입되거나 현장 프로그래밍될 수 있음(field programmed)]. 다른 대안으로서 또는 그에 부가하여, 정적 데이터베이스(110)는 비일시적(non-transitory 또는 non-transient) 기계 판독가능 데이터 저장 매체에 기록되어 있는 컴파

일된 소프트웨어 제품에 하드 코딩된다. 예를 들어, 본 개시 내용의 실시예들에서, 정적 데이터베이스(110)가, 상기한 바와 같이, 동적 데이터베이스(120)로부터 생성되고, 대응하는 정적 데이터베이스(110)로서 저장될 수 있다. 유익하게도, 정적 데이터베이스들(110) 각각은 그에 관련된 고유 참조 코드에 의해 식별된다. 동적 데이터베이스들(120)은 선택적으로 RAM 또는 임의의 동등한 판독-기입 데이터 메모리(예를 들어, 고속 자기 디스크 데이터 저장 장치 등의 자기 또는 광 데이터 저장 매체)에 기입된다.

[0190] 선택적으로, 본 개시 내용의 실시예들은 "점핑(jumping)"을 이용한다. 점핑은 데이터베이스(100) 내의 누락된 요소들(E)이 다른 곳으로부터 공급될 수 있게 해준다. 더욱이, 점핑은 정적 데이터베이스(110) 및 동적 데이터베이스(120) 둘 다에 대해 각각 이용될 수 있다. 게다가, 점핑은 동적 데이터베이스(120)가 누락된 요소들과 관련하여 신속하게 업데이트될 수 있게 해준다. 유익하게도, 이러한 점핑이 효율적으로 기능하기 위해, 모든 데이터가, 데이터베이스의 생성부터 시작하여 현재의 사용 시점까지, 유사한 순서로 디코딩되는 것이 바람직하다.

[0191] 상기한 참조 값들(R)은 데이터베이스(100)에서의 대응하는 요소들(E)을 일의적으로 식별해주고, 여기서 요소들(E)은 상이한 유형의 데이터 블록들을 나타낸다. 그렇지만, 인코더(10)에 의해 인코딩되어야 하는 소스 데이터(20)에 존재하는 데이터 콘텐츠의 유형에 적합하게 되어 있는 방법을 사용하여 참조 값들(R)이 발생하는 것이 바람직하다. 환언하면, 소스 데이터(20)의 일부분에 존재하는 각각의 대응하는 데이터 블록 또는 대응하는 코딩된 데이터 블록에 대해 참조 값(R)이 인코더에서 계산된다. 인코더(10)에서, 디코더(50)가 하나 이상의 데이터베이스들(100)에 부가되는 코딩된 데이터 블록에 대해 동일한 데이터베이스 참조 값(R)을 발생할 수 있도록 하기 위해, 그의 참조 값(R)에 의해 나타내어지는 인코딩된 데이터 블록이 재구성되고[즉, 인코더(10)에서 인코딩된 다음에 다시 디코딩됨]; 이러한 동작 방식으로 하는 이유는, 문제의 주어진 블록에 대해 인코딩 방법이 선택되었을 때, 그 방법이 생성하는 파라미터에 기초하여, 블록이 재구성되거나 디코딩되기 때문이다. 이 재구성된 블록으로부터 이어서 데이터베이스 요소(E) 및 그의 참조 값(R)이 구성된다. 디코더(50)는 소스 데이터(20)를 인코딩할 때 인코더(10)에서 이용되는 것과 완전히 동일한 하나 이상의 데이터베이스들(100)을 발생한다. 본 개시 내용의 실시예들을 구현할 때, 참조 값들(R)을 계산하기 위해 다양한 방법들이 이용될 수 있지만, 이 참조 값(R)은 유익하게도 인코더(10)에서 인코딩될 소스 데이터(20)에 존재하는 데이터 블록들에 관련된 진폭(또는 최소 및 최대) 값, 분산(또는 표준 편차) 값, 평균(또는 합) 값 및 체크섬(또는 해쉬 또는 인덱스) 값으로부터 계산된다. "진폭"은, 예를 들어, 주어진 데이터 블록 내에서 가장 큰 원래의 데이터 또는 픽셀 값과 가장 작은 원래의 데이터 또는 픽셀 값 사이의 차를 말한다. (2D) 데이터 블록에 대한 합 값은 선택적으로 하기의 수학적 식 1을 사용하여 계산되고:

수학적 식 1

$$SUM = \sum_{i=1}^m \sum_{j=1}^n block(i, j)$$

- [0192] .
- [0193] 여기서
- [0194] SUM은 합 값이고;
- [0195] m x n은 (2D) 데이터 블록에 존재하는 픽셀들 또는 데이터 값들의 수이며(예를 들어, 8x8 픽셀 데이터 블록에 대해 n = 8, m = 8임);
- [0196] i, j는 참조 인덱스(reference index)이다.
- [0197] 평균 값은 합 값을 $1/(m \times n)$ 와 곱함으로써(즉 합 값을 합 값을 발생하기 위해 합산된 데이터 값들의 수로 나눔으로써) 계산된다.
- [0198] 더욱이, (2D) 데이터 블록의 분산은 하기의 수학적 식 2를 사용하여 계산될 수 있고:

수학식 2

$$V = \frac{1}{m * n} \left(\left(\sum_{i=1}^m \sum_{j=1}^n block[i, j]^2 \right) - M^2 \right)$$

[0199]

[0200]

[0201]

[0202]

[0203]

여기서

V는 분산이다.

표준 편차는 분산(V)의 제곱근으로부터 계산될 수 있다.

유익하게도, 데이터 블록에 대한 체크섬은 모든 값[즉, 관심 영역(ROI) 내의 픽셀]에 대해 논리 XOR 함수를 사용하여 계산된다. 예를 들어, 정적 데이터베이스(110) 및 동적 데이터베이스(120)에 대해, 각각, 진폭, 분산(V), 평균(M) 및 체크 섬이 표 4에 제공되어 있다:

표 4

[0204]

예시적인 진폭, 분산, 평균 및 체크 섬

파라미터	예시적인 값
진폭	AMPLITUDE_BITS = 4
분산	VARIANCE_BITS = 3
평균	MEAN_BITS = 8
체크 섬	CHECKSUM_BITS = 5

[0205]

데이터베이스 참조 값(R)의 기지의 요소들(E)에 대해, 의사 코드 계산은, 예를 들어, 다음과 같이 실행될 수 있다:

[0206]

$$DBREF = (AMPLITUDE \text{ SHL } (DBREF_BITS - AMPLITUDE_BITS)) + (VARIANCE \text{ SHL } (DBREF_BITS - (AMPLITUDE_BITS + VARIANCE_BITS))) + (MEAN \text{ SHL } (DBREF_BITS - (AMPLITUDE_BITS + VARIANCE_BITS + MEAN_BITS))) + CHECKSUM$$

[0207]

유의할 점은, 계산될 참조 값(R)이 정적 데이터베이스(110)에 대한 결합된 전송된 참조 값(R)이고, 요소(E)에 대한 실제의 검색에서, 보다 정확하고 및/또는 보다 부정확한 값들이 사용될 수 있는 것이다. 체크 섬은, 완벽하게 일치하는 것이 탐색되지 않는 한, 검색을 제한하는 것에 대해 어떤 영향도 미치지 않는다.

[0208]

정적 데이터베이스(110) 또는 동적 데이터베이스(120)의 할당된 최대 바이트 수는, 각각, 데이터베이스 참조 값(R)에 의해 정의된다. 주어진 데이터베이스 참조 값(R)에 대한 총 비트 카운트를 계산하기 위해, 참조 값(R)이 계산되는 경우 모든 요소들(E)이 부가되어야만 하고, 여기서 요소들(E)은 진폭, 분산(V), 평균(M) 및 체크 섬을 포함한다[즉, MVA 결합(MVA combination)]. 예를 들어, 데이터베이스 참조 값(R)에 대한 비트 카운트는 다음과 같이 계산될 수 있다:

[0209]

$$DBREF_BITS = AMPLITUDE_BITS + VARIANCE_BITS + MEAN_BITS + CHECKSUM_BITS$$

[0210]

데이터베이스는 전송된 참조(R)의 요소 의존적 부분의 길이보다 더 많은 요소들을 포함할 수 없다(즉, 데이터베이스가 평균 값을 사용하지 않는 경우(예를 들어, 제로 평균 데이터베이스), 평균 값은 길이에 영향을 미치지 않지만, 그렇지 않은 경우, 영향을 미친다).

[0211]

예를 들어, 표 4의 예에서와 같이 비트들이 설정되어 있는 경우, 보통의 데이터베이스에 있어서는, 요소들(E)의 수가 20 비트를 따를 것이고, 평균 값을 사용하지 않는 데이터베이스에 있어서는 12 비트를 따를 것이며, 이어서 그에 따라 양은 대략 백만개 요소 대 4000개 요소일 것이다. 더욱이, 각각의 요소(E)는, 물론, 데이터의 제시 형식에 따라 바이트들을 필요로 한다. 예를 들어, 8-비트 값들을 갖는 8x8 블록에 대해, 보통의 요소(즉,

평균 값을 가짐)는 요소마다 64 바이트를 필요로 하는 반면, 평균 값 부분을 갖지 않는 요소는 요소(E)마다 $64 * 9/8 = 72$ 바이트를 필요로 하는데, 그 이유는 각각의 데이터 값에 대해 부호 비트가 저장될 필요가 있기 때문이다. 이것은 평균 값을 사용하는 데이터베이스에 대한 예에서 제시되는 바와 같이 8x8 데이터베이스에 대한 최대 크기(64 MB)를 생성하고, 평균 값을 사용하지 않는 데이터베이스에 대해서는, 단지 288 kB를 생성한다. 이들 수치는 따라서 데이터베이스 요소들의 데이터 값들에 대한 최대 할당만을 나타내고, 크기는 따라서 참조 값(R)의 다양한 부분들에 대한 비트 카운트에, 블록의 크기에, 그리고 데이터 값들의 동적 특성에 의존한다. 이것에 부가하여, 고속 검색을 수행하기 위해, 보다 정확하고 및/또는 보다 부정확한 참조 값들(R)이 인코더(10)에 저장될 필요가 있을 것이다. 디코더(50)에 대해서는, 올바른 참조 데이터의 주소들이 쉽게 발견되는 방식으로 구성되어 있는 이러한 데이터베이스 데이터를 갖는 것으로 충분하다.

[0212] 추가의 예로서, 주어진 데이터베이스(100)의 총 크기는 1,048,576개의 요소(즉, 값 2^{20} - 즉, 대략 1 메가 요소들 -)인 것으로 계산된다. 각각의 요소(E)가 픽셀당 1 바이트 동적 범위를 가지는 64개 픽셀을 포함하는 경우(즉, 8x8 픽셀 데이터 블록으로서 구현되는 경우), 미압축된 주어진 데이터베이스(100)의 총 크기는 64 메가바이트이다.

[0213] 도 3에 예시된 바와 같이, 주어진 참조 값(R)은 300으로 표시되어 있고, 몇개의 상호 상이한 부분들(310A, 310B, 310C)[예를 들어, 각각, 분산(V), 평균(M), 진폭(A)]을 포함하고 있다. 이들 부분은 유익하게도 개별적으로 계산되고, 따라서 인코더(10) 및/또는 디코더(50)에서의 검색이 가능한 한 신속하게 구현될 수 있고, 그에 의해 고속 인코딩 및/또는 디코딩 동작을 가능하게 해준다. 상이한 부분들(310)은 유익하게도, 예를 들어, 310D로 표시되어 있는 바와 같은 순환 중복 검사(CRC)의 부가에 의해 개별적인 데이터베이스 서브참조(sub-reference)(즉, 인덱스)로서 기능한다. 더욱이, 큰 데이터베이스(100)가 생성 순서에만 기초하여 또는, 다른 대안으로서, CRC에 기초하여 구성되는 경우, 코덱이 탐색 테이블도 발생하는 것이 유리하다. 이 탐색 테이블은 유익하게도 평균(M), 분산(V) 및 진폭(A)의 적당한 조합을 빠르게 브라우징하는 데 이용된다.

[0214] 참조 값들(R)(300)을 계산할 때, 상이한 샘플링 방식들이 이용될 수 있고, 참조 값들(R)(300)은 유익하게도 부가의 가중 계수들을 포함하고 있으며, 여기서 부가의 가중 계수들은 보다 적은 수의 대안들 중에서 검색하는 것을 가능하게 해주거나, 부가의 가중 계수들은 코덱이, 예를 들어, 영상의 중앙 부분과 비교하여 디코딩 품질과 관련하여 덜 중요한 영상들의 경계들에 대한 요소(E)를 발견할 수 있게 해주고; 이것은, 예를 들어, 소스 데이터(20)를 인코딩할 때 인코더(10)에서 동적으로 변하는 품질 임계값을 이용함으로써 달성될 수 없다. 코덱(5)에서의 검색은 그에 의해 보다 빠르게 실행될 수 있는데, 그 이유는 참조 값(R)이 그와 관련되어 있는 데이터의 보다 정확한 파라미터들을 포함하기 때문이고; 더욱이, CRC 또는 유사한 인덱스는 하나의 기본적인 조합에 대해 몇개의 데이터베이스 요소들(E)을 가지는 것을 가능하게 해준다.

[0215] 선택적으로, 주어진 참조 값(R)에 기초한 하나 이상의 데이터베이스들(100)에서의 검색이 어느 요소들(E)이 그리고 어느 보다 작은 데이터베이스들(100)에서 동일하거나 유사한 유형의 요소들(E)을 가지는지에 관한 참조를 포함하는 큰 데이터베이스(100)로부터 시작하여 실행된다. 다른 대안으로서, 작은 데이터베이스(100)는 또한 작은 데이터베이스(100)가 보유하는 데이터와 관련하여 MVA 조합에 관한 이러한 정보를 유지하고; 검색되고 있는 주어진 데이터 블록에서의 그 조합이 충분히 상이한 경우, 이 정보는 보다 작은 데이터베이스(100)가 인코더(10) 및/또는 디코더(50)에 의해 신속하게 무시될 수 있게 해준다.

[0216] 상기한 참조 값(R)(300)과 관련하여, 선택적으로, 예를 들어, 참조 값(R)(300)을 정의하는 데 필요한 비트들의 총수를 감소시키기 위해 그리고 따라서 도 1에서 인코더(10)로부터 디코더(50)로 전달될 필요가 있는 인코딩된 데이터(30)의 크기를 감소시키기 위해, 분산(V) 및 진폭(A) 모두를 압축하는 것이 유리하고; 이러한 이점은 분산(V) 및 진폭(A)이 강하게 상관되어 있는 것으로부터 도출된다. 이와 달리, 참조 값(R)(300)의 평균(M) 및 CRC/인덱스 부분들은 유익하게도 개별적으로 압축되는데, 왜냐하면 평균(M)은 종종 상대적 델타 특성(relatively delta characteristic)에 부합하지만 CRC는 아주 랜덤하기 때문이다. 보다 적은 비트들을 사용하여 양자화되는 분산(V) 및 진폭(A) 부분들의 조합에 대해, 인코딩된 데이터(30)를 발생하기 위해 런 길이 인코딩(RLE)을 이용하는 것이 유리하고; 다른 대안으로서, 이러한 목적을 위해 델타 인코딩이 이용될 수 있다. 인코딩된 데이터(30)에서 제1 주어진 분산(V) 및 진폭(A) 조합이 비교적 적은 비트들로 양자화되고 제2 주어진 분산(V) 및 진폭(A) 조합이 보다 많은 비트들로 인코딩되는 경우, 이 조합들이 개별적으로 인코딩되는 것이 유리하다(예를 들어, 제1 주어진 조합에 대해 RLE가 이용될 수 있고, 제2 주어진 조합에 대해 허프만 코딩이 이용될 수 있지만, 이들로 제한되지 않음). 다른 유형의 인코딩, 예를 들어,

[0217] Range, SRLE: 2013년 3월 1일자로 Gurulogic Microsystems Oy에 의해 출원된 GB1303660.3에 개시되어 있는 방

법인 "분할 런 길이 인코딩(Split run-length encoding)";

- [0218] EM: 2013년 3월 1일자로 출원된 Gurulogic Microsystems Oy에 의해 출원된 GB1303658.7에 개시되어 있는 방법인 "엔트로피 수정기(Entropy Modifier)"; 및
- [0219] Odelta: 2013년 3월 1일자로 Gurulogic Microsystems Oy에 의해 출원된 GB1303661.1에 개시되어 있는 방법이 유익하게도 이용되지만,
- [0220] 이들로 제한되지 않는다.
- [0221] 인코더(10)로부터 디코더(50)로 인코딩된 데이터(30)를 통해 전달될 참조 값들(R)(300)은 유익하게도 다양한 압축 기법들, 예를 들어,
- [0222] (i) Lempel-Ziv 알고리즘들(예를 들어, ZLIB, LZ0, LZSS, LZ77)을 이용하는 데이터베이스 코딩 방법들을 갖는 허프만 코딩 등의 VLC-기반 방법들; 및
- [0223] (ii) 상기한 값들[예를 들어, 분산(V) 및 진폭(A)]의 조합들에 대한 허프만 코딩 등을 사용하여 압축된다.
- [0224] 선택적으로, 참조 값들(R)(300)은 전체 값으로서 압축될 수 있거나, 예를 들어, 도 3에 예시된 바와 같이, 참조 값들(R)(300)의 일부들이 개별적으로 압축될 수 있다. 더욱이, 선택적으로 인코딩된 데이터(30)에서의 훨씬 더 높은 데이터 압축을 달성하기 위해 데이터 압축 기법들의 조합을 이용하는 것이 가능하고; 예를 들어, 참조 값들(R)(300)의 분산(V) 부분들이 먼저 제1 압축된 데이터를 발생하기 위해 델타 코딩을 사용하여 압축될 수 있고, 이어서 RLE 코딩 기법들이 제2 압축된 데이터를 발생하기 위해 제1 압축된 데이터에 적용될 수 있으며, 이어서 허프만 코딩이 제3 압축된 데이터를 발생하기 위해 제2 압축된 데이터에 적용될 수 있고, 여기서 제3 압축된 데이터는, 예를 들어, 인코딩된 데이터(30)를 통해 인코더(10)로부터 디코더(50)로 전달된다.
- [0225] 데이터베이스들(100) 중 하나 이상은 이전에 알려져 있는 순서로(예를 들어, 점진적으로 증가하는 방식으로) 데이터 블록들의 데이터 값들로 이루어져 있는 요소들(E)을 포함할 수 있다. 이 순서가 도 1의 코덱(5)에 대해 가정된 기본 순서와 상이한 경우, 이러한 상이한 순서는 유익하게도 순서 구성 파라미터에 의해 인코딩된 데이터(30)에 정의되어 있다. 예를 들어, 이러한 종류의 데이터 값을 갖는 데이터베이스 요소(E)가 임의의 올바른 크기의 데이터 블록으로부터(예를 들어, 재구성된 영상으로부터) 생성될 수 있다. 선택적으로, 인코딩된 데이터(30)가 인코더(10)로부터 디코더(50)로 전달되는 동안, 파라미터는 상태들 사이에서 토글할 수 있다.
- [0226] 선택적으로, 이전의 공지된 순서에 따라 업데이트되는 하나 이상의 동적 데이터베이스들(120)은, 예를 들어, 도 1의 코덱(5)이 비교적 보다 적은 정보 콘텐츠를 갖는 영상들에 대해 보다 큰 데이터 압축을 달성할 수 있게 해주기 위해, 프레임당 유연한 수의 요소들(E)을 이용한다. 예를 들어, 상이한 프레임들에 대한 주어진 동적 데이터베이스(120)의 크기들이 126, 193, 252, 303개의 요소인 경우, 각각의 프레임에 대한 각각의 요소를 일의적으로 어드레싱하는 데 필요한 비트들의 수는, 각각, 7, 8, 8 및 9 비트이다. 큰 데이터베이스(100)에서, 요소들(E)은, 요소(E) 어드레싱과 관련하여, ("set number", "get number") 유형의 구조체에 배열되어 있을 수 있고; 예를 들어, 8-비트의 바이트에서, 바이트의 한 비트는 전체 값이 전송되었는지를 수신자에게 통지하기 위해 예약되어 있고, 나머지 7 비트들은 송신되는 값(선택적으로, 그 후에 추가의 비트들을 가짐)을 정의하는 데 사용된다. 이러한 방식에 의해, 7-비트 분량의 정보를 1 바이트로 전송하고 14-비트 분량의 정보를 2 바이트로 전송하는 것이 가능하다. 인코딩된 데이터(30)에 정보 비트들을 포함시킬 필요가 없게 하는 것은 도 1의 코덱(5)에서 향상된 데이터 압축을 제공할 수 있다.
- [0227] 선택적으로, 하나 이상의 데이터베이스들(100) 내에 포함되어 있는 요소들(E)은, 예를 들어, 보다 적은 데이터 메모리를 차지하도록 결합될 수 있거나, 예를 들어, 소스 데이터(20)를 인코딩할 때 및/또는 인코딩된 데이터(30)를 디코딩할 때 적절한 방식으로 요소들(E)을 제공하기 위해 하나 이상의 파라미터들을 사용하여 계산적으로 적합하게 되어 있을 수 있는 스케일링 데이터베이스(100) 또는 제로 평균을 갖는 데이터베이스(100)에 대해, 합 또는 계수 요소들을 포함할 수 있다.
- [0228] 선택적으로, 임의의 하나 이상의 파라미터들 또는 블록이 데이터베이스(100) 내로 삽입되기 전에 블록 자체로부터 삭제된 이러한 하나 이상의 파라미터들을 사용하는 것이 가능하다. 그에 대응하여, 데이터베이스 요소(E)가 사용되는 경우, 얻어진 블록이 구성될 때 이들 파라미터에 의해 생성된 데이터가 부가될 필요가 있다. 제1 구현에서, 평균 색상 파라미터가 이용된다. 슬라이딩 색상 파라미터들 및 다른 파라미터들이 또한 사용될 수 있다. 원칙적으로, 메모리에 있는 데이터베이스 요소(E)에 의해, 예측 또는 임의의 다른 코딩 방법이 사용될 수 있다. 이 평균은 아주 유용한데, 그 이유는 다른 방식들에서도 참조(R)의 일부로서 평균을 사용하는 것이 가능

하고, 따라서 그를 이용하기 위해 새로운 데이터가 전송될 필요가 없으며, 따라서 메모리에 저장되어 있는 실제 데이터베이스(100)의 크기를 상당히 감소시키는 것이 가능하기 때문이다. 물론, 다른 참조들(R)에 비트들을 부가함으로써, 메모리에 저장되어 있는 데이터베이스(100)의 크기를 증가시키고, 따라서 유익하게도 코딩에서 사용될 수 있는 많은 새로운 요소들(E)을 획득하는 것도 가능하다. 평균이 블록으로부터 삭제될 때, 그 요소는 대략 제로 평균으로 된다.

[0229] 한 예가 이하에 주어지 있으며, 여기서 블록 값들로부터 평균 값을 생략하는 것이 이용되고, 그에 의해 상이한 평균 값들과 관련하여 데이터베이스의 보다 효율적인 사용을 제공한다. 상기한 인코딩 방법은 유익하게도 슬라이딩 블록들에 대해, 예측에 대해, 또는 어떤 다른 방식으로 코딩된 블록들에 대해 적용된다. 이와 같이 얻어진 결과는 이어서, 데이터베이스 요소들(E)의 구성 이전에, 데이터베이스로부터 삭제되고, 그에 대응하여, 데이터 블록의 재구성 시에, 디코딩과 관련하여 데이터베이스로부터 폐치된 데이터베이스 요소(E)에 첨부될 수 있다.

[0230] 따라서, 평균 값은 가장 간단한 코딩 방법이고, 유익하게도, 그와 관련해서도, 참조 값(R)이 다수의 부분들로부터 구성될 때 참조 값(R)에서의 하나의 요소 또는 부분에 대한 필요성을 실제로 명확히 제거하며, 그에 의해 데이터베이스의 크기를 감소시킨다. 진폭도 역시 데이터 값들을 상대적으로 취급하는 데 사용될 수 있고, 그에 의해 데이터베이스의 크기를 감소시키는 일 없이 데이터베이스의 적용 범위(coverage)에 대한 이점들을 제공할 것이다.

[0231] 다른 예측들 또는 코딩 방법들은 실제로 예측 오차 블록들만을 생성하고, 그들의 사용은 그만큼 실용적이지 않다. 물론, 평균 값들이 또한 그들로부터 생략될 수 있고, 그에 의해 원래의 블록들에 대한 데이터베이스로서 대응하는 속성들을 갖는 예측 오차 유형 데이터베이스를 제공한다. 물론, 데이터베이스 요소들(E)이 다른 데이터베이스 블록에 대한 예측 오차 데이터로서 사용될 수 있고, 따라서 데이터베이스 요소들(E)의 반복이 또한 본 개시 내용에 따라 지원된다는 것을 잘 알 것이다.

[0232] 그렇지만, 데이터베이스들이 아주 작지 않는 한 또는 정말로 정확한 재구성이 요망되지 않는 한, 많은 수의 반복을 실행하는 것은 유리하지 않은데, 그 이유는 데이터 블록의 인코딩된 데이터 크기가 빠르게 아주 커질 것이기 때문이다.

[0233] 더욱이, 하나 이상의 데이터베이스들(100)은, 예를 들어, 외삽 기법을 사용하여 계산된 요소들(E)을 포함할 수 있고; 환언하면, 코텍(5)은, 유익하게도 데이터 블록의 데이터 값들이 계산될 수 있는 기초를 제공하는 하나 이상의 값들과 함께, 주어진 데이터 블록을 발생하는 데 사용된 계산 방법 및 주어진 데이터 블록의 총 크기를 알고 있다. 이러한 계산은 유익하게도 DC 방법 및 슬라이드 방법을 사용하여 구현된다. DC 방법은 주어진 데이터 블록 내의 모든 데이터 값들에 대해 나중에 적용되는 하나의 값(예를 들어, 오프셋 값)을 이용한다. 이와 달리, 예를 들어, 1차원 슬라이드 방법으로서 구현되는 슬라이드 방법은 2개의 값(예를 들어, 주어진 블록에서의 첫번째 데이터 값 및 마지막 데이터 값)을 이용하고, 이어서 선형 방식으로, 다른 대안으로서, 다항식 방식으로 첫번째 데이터 값 및 마지막 데이터 값에 기초하여 선형 보간에 의해 데이터 블록에 대한 나머지 값들을 계산한다. 더욱이, 데이터 블록에서의 코너 점에 대응하는 4개의 값을 이용하는 2차원 슬라이드 방법이 선택적으로 이용되고, 여기서 데이터 블록 내의 나머지 점들은 이중 선형 보간(bi-linear interpolation)을 사용하여 계산된다. 게다가, 도 1의 코텍(5) 내에서 비교적 적은 초기 값들로부터 데이터 블록들에 대한 값들을 계산하기 위해 다른 유형의 계산 알고리즘들이 선택적으로 이용된다.

[0234] 데이터베이스들(100) 중 하나 이상은 선택적으로 다른 데이터베이스들에 대한 참조들인 요소들(E)을, 예를 들어, 계층적 방식으로, 포함하고 있다. 이러한 경우에, 요소(E)는 다른 데이터베이스(100)를 정의하는 정보 그리고 또한 유익하게도 그 다른 데이터베이스(100) 내의 요소(E)에 대한 참조 값(R)을 포함하고 있다. 이러한 정보에 기초하여, 주어진 원하는 데이터 블록에 대한 데이터 값들이 계산될 수 있다. 제1 데이터베이스의 요소(E)로부터 호출된 제2 데이터베이스로부터의 하나 이상의 요소들(E)에 액세스하는 이러한 방식은 유익하게도, 예를 들어, 제2 데이터베이스를 참조하는 데이터베이스 요소(E)가 제1 데이터베이스의 요소(E)를 호출하는 것에 의해 반환될 원하는 데이터 블록에 대해 상이한 크기(예를 들어, 데이터 블록 크기)를 가지는 경우, 하나 이상의 스케일링 계수들 및/또는 하나 이상의 픽셀 값 오프셋 천이를 이용한다. 유익하게도, 이러한 방식은 URL http://en.wikipedia.org/wiki/Affine_transformation을 통해 2012년 10월 19일자로 액세스된 바와 같은 공개된 위키피디아에 기술되어 있는 바와 같이 Affine 모델을 이용한다. 표 5에는, 도 1의 코텍(5)을 구현하는 데 이용되는 데이터베이스 요소 유형들의 개요가 제공되어 있다.

표 5

데이터베이스 요소 유형들

유형	서브유형	상세한 설명
데이터 값을 갖는 요소들		이들은 데이터 블록의 데이터 값들로 이루어져 있다.
계산된 요소들	DC 방법	이들은 사용되는 방법 및 그 방법에서 사용하기 위한 특성의 주요 값들을 알 필요가 있다.
	슬라이드 방법	
다른 데이터베이스들에 대한 참조들	이들은 선택적으로 스케일링 계수 및/또는 픽셀 천이 또는 변환을 포함하고 있다.	이들은, 제2 데이터베이스에서의 참조 값들과 함께, 사용되는 방법을 알고 있을 필요가 있다.
결합된 요소들	이들은 선택적으로 합 또는 계수 요소들(예를 들어, 스케일링 데이터베이스 또는 제로 평균 값을 갖는 데이터베이스)을 포함하고 있다.	이들은 하나 이상의 부가의 파라미터와 함께 이용될 수 있다.

[0236] 본 개시 내용의 실시예들은 어쩌면 다양한 유형의 콘텐츠(예를 들어, 스트리밍된 비디오 콘텐츠, 인터넷 TV, 광 관독가능 디스크 등의 물리적 데이터 캐리어를 통해 전달되는 콘텐츠 등)를 나타내는 데이터를 송신 및/또는 수신하는 동작을 하는 전자 제품에서 광범위한 응용 분야를 가진다. 더욱이, 예를 들어, 실질적으로 대체로, 참조 값들(R)이 인코딩된 데이터(30)를 통해 전달되도록 데이터베이스들(100)이 특성의 당사자들에게만 이용가능하게 되어 있을 때, 본 개시 내용은 데이터 암호화 서비스를 제공할 수 있고; 여기서의 착상/목적은, 데이터베이스로부터의 참조의 도움을 받아, 인코딩된 데이터의 적어도 대부분을 송신하는 것이 가능하고 작은 부분만이 어떤 다른 코딩 방법들에 의한 블록 값들의 코딩을 필요로 한다는 것이다. 이와 같이, 실시예들은 많은 다양한 기술 산업에서 적용가능하다. 더욱이, 실시예들은 통신 네트워크 부하 또는 요구사항을 감소시킬 수 있고, 데이터 저장 용량 요구사항을 감소시킬 수 있는 것은 물론, 어쩌면 고품질 데이터 인코딩 및 디코딩이 달성될 수 있게 해줄 수 있고; 이들은 유익한 기술적 효과이다.

[0237] 인코더(10) 및 디코더(50)는 유익하게도 하나 이상의 하드웨어 데이터 프로세서들로서 구현된다. 선택적으로, 하드웨어 데이터 프로세서들은, 예를 들어, ASIC 또는 유사한 디지털 회로를 이용함으로써, 하드와이어드 방식으로 구현된다. 선택적으로, 하드웨어 데이터 프로세서들은 컴퓨팅 하드웨어를 사용하여 구현되고, 여기서 컴퓨팅 하드웨어는 비일시적(non-transitory 또는 non-transient) 기계 관독가능 데이터 저장 매체 상에 기록되어 있는 하나 이상의 소프트웨어 제품을 실행하는 동작을 한다.

[0238] 상기한 하나 이상의 데이터베이스들(100)은 선택적으로 ISP에 의해 제공되는 인코더(10)에서, 디코더(50)에서, LAN에서 구현되고, 하드 디스크 메모리 등의 개별적인 메모리에서, 메모리 카드에서, 블루레이 디스크 상에, RAM에, ROM에, 플래시 데이터 메모리에, SSD 상에, DVD 상에, CD 상에 및 그의 데이터에의 액세스를 제공하는 임의의 다른 유형의 데이터 캐리어 상에 구현되어 있다. "블루레이"는 등록 상표이다.

[0239] 하나 이상의 데이터베이스들(100)은, 하나 이상의 데이터 서버들이 공간적으로 어디에 위치하든 간에, 예를 들어, 하나 이상의 메모리 카드들로서 구현되는 하나 이상의 데이터 서버들에 의해 제공될 때, 고정된 연결을 통해, 접촉점을 통해, 케이블을 통해, 무선 링크를 통해, 무선 네트워크(예를 들어, 블루투스 및 WLAN)를 통해, 셀룰러 네트워크(예를 들어, GSM, 3G, 4G, 5G, LTE 등)를 통해, 위성 연결을 통하는 것(이들로 제한되지 않음)과 같은 하나 이상의 방식들로 인코더(10) 및/또는 디코더(50)에 연결될 수 있다. "블루투스"는 등록 상표이다.

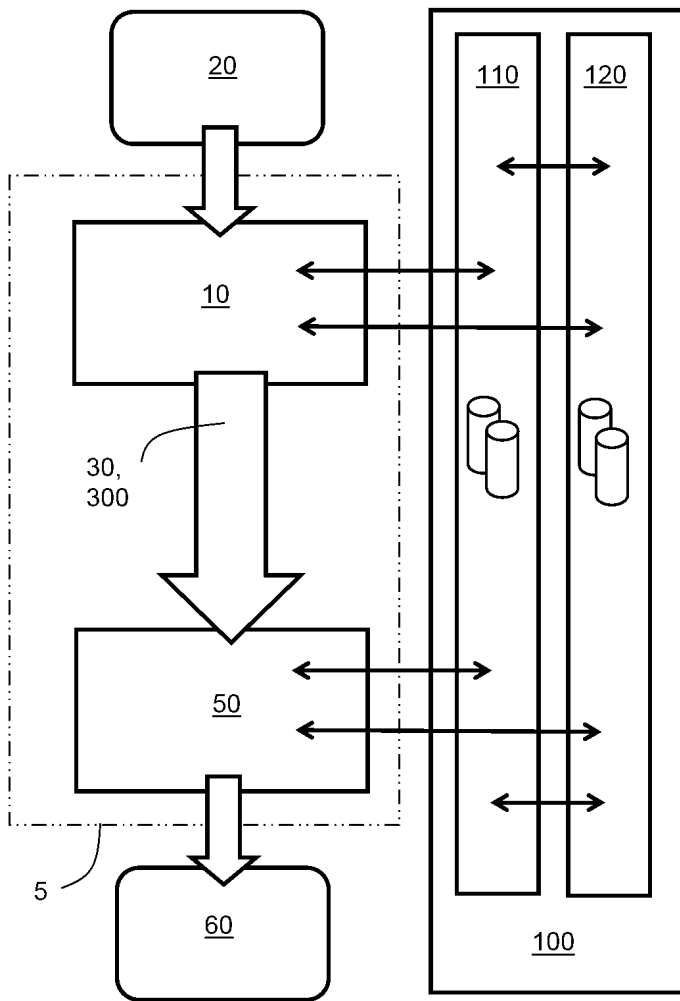
[0240] 선택적으로, 이상에서, 요소들(E)은, 한번에 하나씩, 디코딩된 데이터 블록들 중에서 데이터베이스(100)에 삽입되거나, 결합되거나 또는 부분적으로만 사용될 수 있다. 즉, 원칙적으로, 임의의 디코딩된 데이터 값들의 세트가 주어진 또는 합의된 원칙에 따라 데이터베이스에서의 새로운 요소(E)를 구성할 수 있고, 임의의 코딩 방법 또는 코딩 해결 방안이 디코딩된 데이터를 생성하기 위해 사용될 수 있고, 그 요소들 중에서 데이터베이스에 삽입된다.

[0241] 선택적으로, 인코더(10) 및 디코더(20)는, GB1218942.9 및 동등한 미국 특허 출원 제13/657,382호는 물론, 대응하는 출원들 EP13003859.9 및 US 14/058,793에 개시되어 있는 방법에 따라, 다중 레벨 인코더 및 대응하는 다중 레벨 디코더의 서브구성요소로서 포함되어 있다. 다중 레벨 인코더에서, 예를 들어, 소스 데이터(20)에 존재하는 콘텐츠의 성질에 따라, 소스 데이터(20)가 분석되고 복수의 데이터 스트림들로 분할된다. 복수의 데이터 스트림들 중 하나가 대응하는 인코딩된 데이터를 발생시키는 인코더(10)에 제공되고; 대응하는 인코딩된 데이터는 다중 레벨 인코더로부터 제공되는 최종적인 인코딩된 데이터를 발생시키기 위해 복수의 데이터 스트림들 중의 다른 데이터 스트림들을 처리하는 다른 인코더들로부터의 다른 인코딩된 출력 데이터와 결합된다. 다중 레벨 디코더에서, 최종적인 인코딩된 데이터가 수신되고, 그 후에 복수의 데이터 스트림들로 분할되며, 여기서 데이터 스트림들 중 하나는 대응하는 디코딩된 데이터를 발생시키기 위해 디코더(50)에 제공되고; 대응하는 디코딩된 데이터는 최종적인 디코딩된 출력 데이터를 발생시키기 위해 다중 레벨 디코더의 다른 디코더들로부터 제공되는 디코딩된 데이터와 결합된다. 이러한 다중 레벨 인코더들 및 디코더들이 단지 인코더(10) 및 디코더(50)보다 더 복잡하지만, 이러한 다중 레벨 인코더들 및 디코더들은, 광범위한 유형의 데이터를 처리할 때, 우수한 인코딩, 디코딩 및 데이터 압축 성능을 제공할 수 있다.

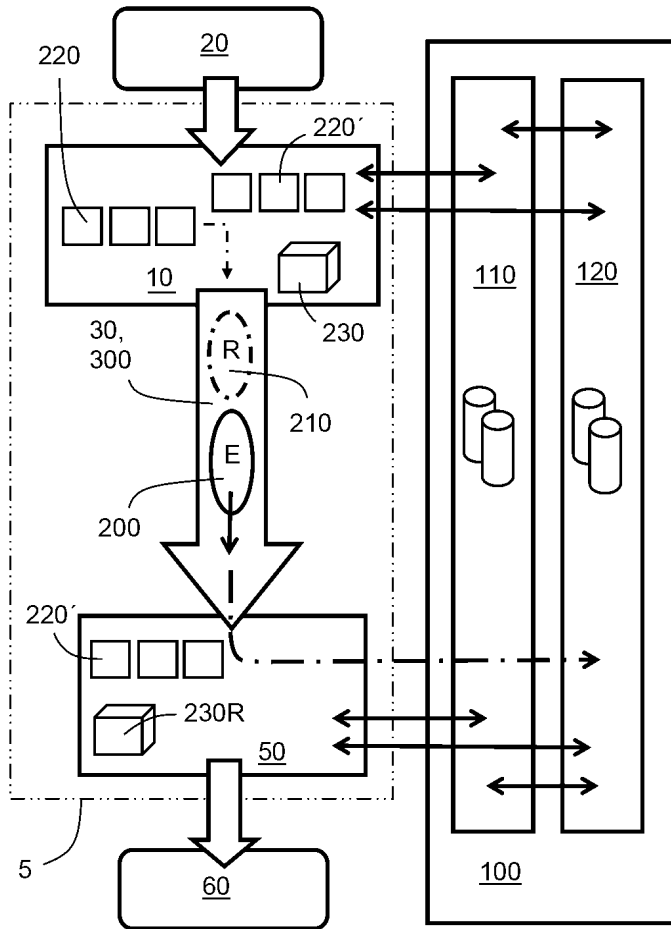
[0242] 첨부된 특허청구범위에 의해 한정된 본 발명의 범위를 벗어나지 않고 이상에서 기술된 본 발명의 실시예에 대한 수정이 가능하다. "포함하는", "구비하는", "내포하는", "~로 이루어져 있는", "가지는", "~인" 등의 표현은 본 발명이 비배타적 방식으로 해석되어야만 한다(즉, 명시적으로 기술되어 있지 않은 항목, 구성요소 또는 요소가 또한 존재할 수 있다)는 것을 기술하고 청구하기 위해 사용된다. 단수를 언급하는 것이 또한 복수에 관련되어 있는 것으로 해석되어야 한다. 첨부된 청구항들에서 괄호 안에 포함된 숫자는 청구항의 이해를 돕기 위한 것이며, 이들 청구항에 의해 청구되는 발명 요지를 결코 제한하는 것으로 해석되어서는 안된다.

도면

도면1



도면2



도면3

