



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

(45) 공고일자 2012년10월15일  
 (11) 등록번호 10-1190875  
 (24) 등록일자 2012년10월08일

- |  |  |
|--|--|
| (51) 국제특허분류(Int. Cl.)<br>H03M 7/30 (2006.01)<br>(21) 출원번호 10-2006-7017487<br>(22) 출원일자(국제) 2004년01월30일<br>심사청구일자 2008년12월30일<br>(85) 번역문제출일자 2006년08월29일<br>(65) 공개번호 10-2006-0129417<br>(43) 공개일자 2006년12월15일<br>(86) 국제출원번호 PCT/FR2004/000219<br>(87) 국제공개번호 WO 2005/083889<br>국제공개일자 2005년09월09일<br>(56) 선행기술조사문헌<br>P. Combescure et al., "A 16,24,32 kbit/s<br>wideband speech codec based on ATCELP," 1999<br>ICASSP, pp. 5-8, 15 March 1999.<br>KR1020030087373 A | (73) 특허권자<br>프랑스 텔레콤<br>프랑스 파리 75015 뽀라스 달르래 6<br>(72) 발명자<br>랑블랭 끌로드<br>프랑스 에프-22700 뽀로 귀이렉 레지당스 뽀우엔<br>텍<br>비레떼 다비드<br>프랑스 22560 플뢰외르 보두 레지당스 케르녹 비<br>스 슈맹드 브레넬로 1<br>(뒷면에 계속)<br>(74) 대리인<br>박장원 |
|--|--|

전체 청구항 수 : 총 33 항

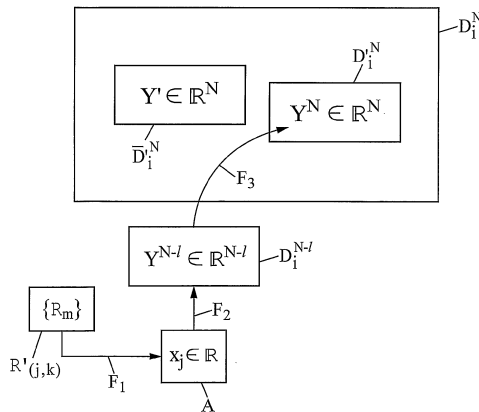
심사관 : 권성락

**(54) 발명의 명칭 차원 벡터 및 가변 분해능 양자화**

**(57) 요약**

본 발명은 특히, 가변 분해능을 정의하는 벡터 가변-속도 양자화에 의해 디지털 신호들의 압축 코딩 및/또는 디코딩에 관한 것이다. 이러한 목적을 위해 동기적인 디셔너리는 주어진 차원에 대해, 서로 겹쳐지는 증가하는 분해능 디셔너리들 및 소정의 삽입 규칙들 (F1)의 최종 세트에 따라 더 작은 차원 코드벡터들로 최종 세트 (A)에서 취해진 요소들을 삽입함으로써 생성된 코드벡터들의 총체성 및 상기 삽입 규칙들의 세트에 따라 더 작은 차원 코드벡터들로의 삽입에 의해 획득되지 않는 코드벡터들(Y')의 제 2 총체성의 조합을 포함한다.

**대표도** - 도2b



(72) 발명자

**코베시 발라즈**

프랑스 22300 라니옹 슈맹 뒤 물랭 아 벵 16

**마살루 도미니끄**

프랑스 에프-22700 베로-귀이렉 뒤 뒤 프레 드 생  
-모르53

---

## 특허청구의 범위

### 청구항 1

가변 차원의 코드벡터들을 포함하고 가변 분해능을 정의하는 가변 속도에서의 벡터 양자화에 의해 디지털 신호들의 압축 코딩 또는 디코딩을 위한 디바이스에서 사용되는 디셔너리를 저장하고 있는 컴퓨터 판독가능한 매체로서,

상기 디셔너리는,

- 한편으로, 주어진 차원에 대해, 증가하는 분해능을 가지며, 상호 간에 내장되는 디셔너리들(inter-embedded dictionaries)과; 그리고

- 다른 한편으로, 주어진 차원에 대해:

● 소정의 삽입 규칙들의 유한 집합에 따라 실수들(real numbers)의 유한 세트로부터 취해진 요소들을 더 낮은 차원의 디셔너리들의 코드벡터들 내에 삽입함으로써 구성되는 코드벡터들로 이루어진 제 1 세트와,

● 상기 삽입 규칙들의 유한 집합에 따라 상기 유한 세트의 요소들을 더 낮은 차원의 코드벡터들 내에 삽입함으로써 얻어질 수 없는 코드벡터들로 이루어지는 제 2 세트의 조합을 포함하는 것을 특징으로 하는 컴퓨터 판독가능한 매체.

### 청구항 2

제 1 항에 있어서,

상기 삽입 규칙들의 유한 집합은 주어진 벡터 위치에서의 구성요소를 가장하는 실수들의 유한 세트의 단일 요소를 삽입하는 것으로 이루어지는 기본 규칙들에 기초하여 공식화되는 것을 특징으로 하는 컴퓨터 판독가능한 매체.

### 청구항 3

제 2 항에 있어서,

- 상기 유한 세트의 요소의 랭크; 및

- 삽입 위치를 나타내는 한 쌍의 2개의 양의 정수들에 의해 각각의 기본 규칙이 정의되는 것을 특징으로 하는 컴퓨터 판독가능한 매체.

### 청구항 4

제 1 항 내지 제 3 항 중 어느 한 항에 따른 가변 차원의 코드벡터들을 포함하고 가변 분해능을 정의하는 가변 속도에서의 벡터 양자화에 의해 디지털 신호들의 압축 코딩 또는 디코딩을 위한 디바이스에서 사용되는 디셔너리를 형성하기 위한 방법으로서,

주어진 차원에 대해:

a) 삽입 규칙 및 삭제 규칙으로 이루어지는 그룹으로부터 선택되는 소정의 동작 규칙들의 유한 집합에 따라 실수들의 유한 세트로부터 취해지는 저차원 및 고차원 요소들중 적어도 하나를 포함하는 차원 요소들의 디셔너리들의 코드벡터들로의 삽입과 이러한 코드벡터들로부터의 삭제로 이루어지는 그룹으로부터 선택되는 동작을 수행함으로써 코드벡터들로 이루어진 제 1 세트를 형성하는 단계와;

b) 상기 주어진 차원에 대해, 적어도 상기 제 1 세트를 포함하는 제 1 중간 디셔너리를 구성하는 단계와; 그리고

c) 적어도 하나의 주어진 분해능에 대해 이용할 수 있도록 상기 디셔너리를 적용시키는 단계를 포함하며,

여기서, 상기 중간 디셔너리에 기초하여, 증가하는 분해능과 감소하는 분해능 중 적어도 하나의 디셔너리들의 내장과 간략화중 적어도 하나를 수행함으로써 제 2 최종 디셔너리가 구성되며,

상기 증가하는 분해능의 디셔너리들은 최소 분해능의 디셔너리로부터 최대 분해능의 디셔너리까지 상호 간에 내장되는 것을 특징으로 하는 디셔너리를 형성하기 위한 방법.

**청구항 5**

제 4 항에 있어서,

주어진 차원  $N$ 에 대해, 상기 방법은,

a0) 상기 주어진 차원  $N$  보다 낮은 초기 차원  $n$ 의 초기 디셔너리를 획득하는 단계와;

a1) 소정의 삽입 규칙들의 유한 집합에 따라 실수들의 유한 세트로부터 취해지는 초기 디셔너리 요소들을 코드 벡터들 내에 삽입함으로써 차원  $n+i$ 의 코드벡터들로 이루어진 제 1 세트를 형성하는 단계와;

a2) 상기 삽입 규칙들의 유한 집합으로 상기 유한 세트의 요소들의 초기 디셔너리의 코드벡터들로의 삽입에 의해 획득되지 않는 차원  $n+i$ 의 코드벡터들로 이루어진 제 2 세트를 제공하는 단계와; 그리고

a3) 상기 제 1 세트와 상기 제 2 세트의 조합을 포함하는 차원  $n+i$ 의 중간 디셔너리를 구성하는 단계를 더 포함하며,

여기서, 상기 단계 a1) 내지 a3)은, 상기 주어진 차원  $N$ 까지 초기 디셔너리를 가장하는 상기 중간 디셔너리로 많아야  $N-n-1$ 회로 반복되는 것을 특징으로 하는 디셔너리를 형성하기 위한 방법.

**청구항 6**

제 4 항에 있어서,

주어진 차원  $N$ 에 대해, 상기 방법은,

a'0) 상기 주어진 차원  $N$  보다 높은 초기 차원  $n$ 의 초기 디셔너리를 획득하는 단계와;

a'1) 소정의 삭제 규칙들의 유한 집합에 따라, 차원  $n$ 의 디셔너리로부터 차원  $n-i$ 의 가능한 코드벡터들의 선택 및 추출에 의해 차원  $n-i$ 의 제 1 세트를 구성하는 단계와;

a'2) 상기 초기 디셔너리의 코드벡터들로부터, 상기 삭제 규칙들의 유한 집합으로 상기 유한 세트의 요소들의 삭제에 의해 획득될 수 없는 차원  $n-i$ 의 코드벡터들로 이루어진 제 2 세트를 제공하는 단계와; 그리고

a'3) 상기 제 1 세트 및 상기 제 2 세트의 조합을 포함하는 차원  $n-i$ 의 중간 디셔너리를 구성하는 단계를 더 포함하며,

여기서, 상기 단계들 a'1) 내지 a'3)은, 상기 주어진 차원  $N$ 까지 초기 디셔너리를 가장하는 상기 중간 디셔너리로 많아야  $n-N-1$ 회로 반복되는 것을 특징으로 하는 디셔너리를 형성하기 위한 방법.

**청구항 7**

제 5 항에 있어서,

차원  $n+1$  내지  $N$ 에 대한 상기 단계들 a1) 내지 a3)의 반복 구현을 통해, 차원  $n$ 의 초기 디셔너리에 기초하여 각각의 차원들 1 내지  $N$ 의  $N$ 개의 연속적인 디셔너리들이 획득되는 것을 특징으로 하는 디셔너리를 형성하기 위한 방법.

**청구항 8**

제 4 항에 있어서,

상기 동작 규칙들의 유한 집합은 주어진 벡터 위치에서의 구성요소를 가장하여 실수들의 유한 세트의 단일 요소 상에서 삽입과 삭제중 적어도 하나의 동작을 수행하는 것으로 이루어지는 기본 규칙들에 기초하여 공식화되는 것을 특징으로 하는 디셔너리를 형성하기 위한 방법.

**청구항 9**

제 8 항에 있어서,

각각의 기본 규칙은:

- 상기 유한 세트의 요소의 랭크; 및

- 상기 동작의 위치를 나타내는 한 쌍의 2개의 양의 정수들에 의해 정의되는 것을 특징으로 하는 디셔너리를 형성하기 위한 방법.

**청구항 10**

제 4 항에 있어서,

상기 동작 규칙들의 유한 집합 및 상기 유한 세트는 양자화될 소스의 분석에 의해 상기 디셔너리를 구성하기 전에 미리 정의되는 것을 특징으로 하는 디셔너리를 형성하기 위한 방법.

**청구항 11**

제 10 항에 있어서,

상기 소스는 러닝 시퀀스에 의해 모델링되며, 상기 유한 세트 및 상기 동작 규칙들의 유한 집합의 정의는 상기 소스의 통계적 분석에 의해 이루어지는 것을 특징으로 하는 디셔너리를 형성하기 위한 방법.

**청구항 12**

제 10 항에 있어서,

상기 유한 세트는 상기 소스의 단일차원 확률 밀도의 추정에 의해 선택되는 것을 특징으로 하는 디셔너리를 형성하기 위한 방법.

**청구항 13**

제 4 항에 있어서,

상기 유한 세트 및 상기 동작 규칙들의 유한 집합은, 구성된 이들 디셔너리들의 통계적 분석을 수반하는, 연속하는 분해능들의 디셔너리들의 내장과 간략화중 적어도 하나를 수행함으로써 디셔너리들의 구성 후에 사후에 정의되는 것을 특징으로 하는 디셔너리를 형성하기 위한 방법.

**청구항 14**

제 10 항에 있어서,

- 제 1 세트 및 동작 규칙들의 제 1 집합이 러닝 시퀀스의 분석에 의해 사전에 선택되어, 하나 이상의 중간 디셔너리들을 형성하고,
- 상기 동작 규칙들의 제 1 집합 및 상기 제 1 세트의 적어도 일부분이 상기 하나 이상의 중간 디셔너리들의 사후 분석에 의해 업데이트되는 것을 특징으로 하는 디셔너리를 형성하기 위한 방법.

**청구항 15**

제 4 항에 있어서,

상기 단계 c)는 다음의 단계들 c0) 내지 c2):

- c0) 상기 주어진 분해능  $r_N$  보다 낮은 초기 분해능  $r_n$ 의 초기 디셔너리를 획득하는 단계와;
- c1) 상기 초기 디셔너리에 기초하여, 초기 분해능  $r_n$  보다 높은 분해능  $r_{n+1}$ 의 중간 디셔너리를 구성하는 단계와; 그리고
- c2) 상기 주어진 분해능  $r_N$ 이 달성될 때까지 상기 단계 c1)을 반복하는 단계를 포함하는 것을 특징으로 하는 디셔너리를 형성하기 위한 방법.

**청구항 16**

제 15 항에 있어서,

상기 단계 c1)의 각 반복에 대해, 등급들 및 중심(centroid)들의 구성이 제공되며, 적어도 현재 분해능  $r_i$  보다 높은 분해능의 디셔너리들에 속하는 중심들이 재계산되고 업데이트되는 것을 특징으로 하는 디셔너리를 형성하

기 위한 방법.

**청구항 17**

제 16 항에 있어서,

현재 분해능  $r_i$  보다 낮은 분해능의 디셔너리들에 속하는 중심들은 더 낮은 분해능의 모든 디셔너리들의 총 왜곡률이 하나의 업데이트로부터 다음으로 감소하는 경우에만 업데이트되는 것을 특징으로 하는 디셔너리를 형성하기 위한 방법.

**청구항 18**

제 4 항에 있어서,

상기 단계 c)는 다음의 단계들:

c'0) 상기 주어진 분해능  $r_N$  보다 높은 초기 분해능  $r_n$ 의 초기 디셔너리를 획득하는 단계와;

c'1) 상기 초기 디셔너리에 기초하여, 상기 초기 분해능  $r_n$  보다 낮은 분해능  $r_{n-1}$ 의 중간 디셔너리를 소정의 기준에 따라 정렬된 여러 서브세트들로의 상기 초기 디셔너리의 분할에 의해 구성하는 단계와; 그리고

c'2) 상기 주어진 분해능  $r_N$ 이 획득될 때까지 상기 단계 c'1)을 반복하는 단계를 포함하는 것을 특징으로 하는 디셔너리를 형성하기 위한 방법.

**청구항 19**

제 18 항에 있어서,

상기 소정의 기준은 상기 서브세트들의 기수, 러닝 시퀀스의 서브세트들의 인보킹, 총 왜곡에 대한 서브세트들의 기여, 및 상기 총 왜곡의 감소에 대한 서브세트들의 기여 중에서 선택되는 것을 특징으로 하는 디셔너리를 형성하기 위한 방법.

**청구항 20**

제 18 항에 있어서,

상기 분할은 상기 동작 규칙들 중 적어도 일부분을 이용하는 것을 특징으로 하는 디셔너리를 형성하기 위한 방법.

**청구항 21**

제 15 항에 있어서,

증가하는 분해능들  $r_{n+1}$  내지  $r_N$ 에 대한 단계 c1)의 반복 구현 및 감소하는 분해능들  $r_{n-1}$  내지  $r_1$ 에 대한 단계 c'1)의 반복 구현을 통해, 각각의 분해능들  $r_1$  내지  $r_N$ 의 N개의 연속 디셔너리들이 중간 분해능  $r_n$ 의 초기 디셔너리에 기초하여 획득되는 것을 특징으로 하는 디셔너리를 형성하기 위한 방법.

**청구항 22**

제 4 항에 있어서,

상기 디셔너리를 코드벡터들의 주어진 차원 N과의 이용에 적응시키기 위해, 한편으로 단계들 a) 및 b), 그리고 다른 한편으로 단계 c)가 실질적으로 반전되어:

- 단계 c)에서, 여전히 차원 N'이지만 더 높은 분해능과 더 낮은 분해능중 적어도 하나의 분해능  $r_n$ 의 제 1 중간 디셔너리가, 실질적으로 상기 제 1 디셔너리의 분해능  $r_n$ 을 획득하도록, 증가하는 분해능과 감소하는 분해능중 적어도 하나의 디셔너리들의 내장 및 간략화중 적어도 하나를 수행함으로써 분해능  $r_n$  및 차원 N'의 초기 디셔너리에 기초하여 구성되며,
- 단계 a)에서, 주어진 차원 N을 획득하기 위해, 소정의 삽입 규칙과 삭제 규칙중 적어도 하나의 유한 집합에

따라 실수들의 유한 세트로부터 취해진 상기 주어진 차원  $N$  요소들 보다 더 낮은 차원과 더 높은 차원중 적어도 하나의 차원  $N'$ 의 제 1 디서너리를 코드벡터들에 삽입하는 것과 이러한 코드벡터들로부터 삭제하는 것으로 이루어지는 그룹으로부터 선택되는 동작을 수행함으로써 형성되는 코드벡터들로 이루어지는 제 1 세트가 구성되며, 그리고

- 단계 b)에서, 상기 분해능  $r_N$ 에 대한 최종의 적응의 가능한 단계에 후속하여, 적어도 상기 제 1 세트를 포함하는 제 2 최종 디서너리가 상기 주어진 차원  $N$ 에 대해 구성되는 것을 특징으로 하는 디서너리를 형성하기 위한 방법.

**청구항 23**

제 4 항에 있어서,

각각이 인덱스 ( $1_r$ )에 의해 식별되는 상기 동작 규칙들의 유한 집합을 메모리에 저장하는 단계를 더 포함하며, 주어진 차원에 대해:

- 상기 제 2 세트는 상기 동작 규칙들의 유한 집합에 따라 상기 주어진 차원 보다 낮은 차원과 높은 차원 중 적어도 하나의 코드벡터들로의 상기 동작의 적용에 의해 획득되지 않는 코드벡터들로 이루어지며,
- 적어도 하나의 대응 테이블이 상기 동작 규칙들의 인덱스들 및 상기 제 2 세트의 요소들을 식별하는 인덱스들을 이용하여, 주어진 차원의 디서너리의 임의의 코드벡터를 재구성할 수 있게 함으로써,

이들 요소들 및 관련된 동작 규칙들에 대한 액세스를 위해 상기 대응 테이블의 링크들 및 상기 제 2 세트의 요소들을 단순히 저장함으로써 상기 주어진 차원에 대한 상기 디서너리의 완전한 저장을 회피할 수 있게 하는 것을 특징으로 하는 디서너리를 형성하기 위한 방법.

**청구항 24**

제 23 항에 있어서,

상기 대응 테이블은:

- 상기 제 2 세트의 현재 차원 ( $j'$ ),
- 상기 제 2 세트의 요소의 현재 인덱스 ( $m'$ ), 및
- 상기 현재 인덱스 ( $m'$ ) 및 상기 현재 차원 ( $j'$ )의 요소에 삽입과 삭제중 적어도 하나의 동작을 적용함으로써, 적어도 주어진 차원 ( $j$ )의 디서너리 ( $D_{N_j}^j$ )의 상기 코드벡터 ( $x_j$ )를 재구성하는데 기여하는 동작 규칙의 인덱스 ( $1_r$ )를 나타내는 3개의 정수 스칼라 값들의 도표화를 통해, 현재 차원 ( $j'$ )의 제 2 세트에 현재 인덱스들 ( $m'$ )의 요소들에 기초하여 재구성될 수 있는 주어진 차원 ( $j$ )의 디서너리 ( $D_{N_j}^j$ )의 코드벡터 ( $x_j$ )의 각 인덱스 ( $m^j$ )에 대해 미리 공식화되는 것을 특징으로 하는 디서너리를 형성하기 위한 방법.

**청구항 25**

가변 분해능을 정의하는 가변 속도에서의 벡터 양자화에 의해 디지털 신호들을 압축 코딩 또는 디코딩하는 방법으로서,

주어진 차원 ( $j$ )의 디서너리 ( $D_j^j$ )에서 입력 벡터  $y=(y_0, \dots, y_k, \dots, y_{j-1})$ 의 최근접 이웃인 코드벡터 ( $x^j$ )를 탐색하는 단계를 포함하며,

여기서, 상기 코드벡터들은, 삽입 규칙과 삭제 규칙으로 이루어지는 그룹으로부터 선택되는 동작 규칙들의 집합의 인덱스들 및 상기 동작 규칙들의 집합에 따라 상기 주어진 차원 보다 더 낮은 차원과 더 높은 차원중 적어도 하나의 코드벡터들에 대해 삽입과 삭제중 적어도 하나의 동작을 적용함으로써 얻을 수 없는 코드벡터들의 세트의 요소들을 식별하는 인덱스들을 이용하여, 상기 주어진 차원의 디서너리의 임의의 코드벡터를 재구성할 수 있게 하는 적어도 하나의 대응 테이블을 이용하여 재구성되며,

상기 방법은,

C01) 필요한 상기 코드벡터 ( $x^j$ )의 현재 인덱스 ( $m^j$ )에 대해, 디셔너리를 작성할 수 있게 하는 상기 대응 테이블에 나타나는 인덱스들 ( $j'$ ,  $m'$ ,  $l_r$ )의 적어도 사전 관독을 통해 상기 현재 인덱스 ( $m^j$ )에 대응하는 인덱스 ( $m^j$ )의 코드벡터의 적어도 일부분을 재구성하는 단계와;

C02) 적어도 코딩 중에, 상기 단계 C01)에서 재구성된 코드벡터와 입력 벡터 사이의 거리를 계산하는 단계와;

C03) 적어도 코딩 중에, 상기 디셔너리의 모든 현재 인덱스들에 대해 상기 단계들 C01) 및 C02)을 반복하는 단계와;

C04) 적어도 코딩 중에, 상기 단계 C02)의 반복들 중 하나의 과정에서 계산되는, 상기 입력 벡터와의 거리( $d_{m_i}$ )가 최소인, 적어도 부분적으로 재구성된 코드벡터의 인덱스( $m_{min}$ )를 식별하는 단계와; 그리고

C05) 적어도 디코딩 중에, 그의 인덱스( $m_{min}$ )가 상기 단계 C04)에서 식별된 코드벡터( $x^j$ )를 가장한 입력 벡터 ( $y$ )의 최근접 이웃을 결정하는 단계를 포함하는 것을 특징으로 하는 벡터 양자화에 의해 디지털 신호들을 압축 코딩 또는 디코딩하는 방법.

**청구항 26**

제 25 항에 있어서,

적어도 디코딩 중에, 상기 단계 C01)은,

C011) 상기 동작 규칙들 및 상기 제 2 세트에 대한 링크들을 나타내며,

- 상기 제 2 세트의 서브세트의 현재 차원의 인덱스,
- 상기 서브세트의 요소의 현재 인덱스, 및
- 상기 요소에 기초하여 주어진 차원의 디셔너리의 코드벡터의 구성을 위해 동작 규칙의 인덱스를 포함하는 인덱스들을 상기 대응 테이블에서 관독하는 단계와;

C012) 그의 현재 인덱스에 의해 식별된 상기 요소를, 그의 현재 차원에 의해 식별된 서브세트에서 관독하는 단계와;

C013) 상기 단계 C012)에서 관독된 상기 요소에 상기 단계 C011)에서 관독된 인덱스에 의해 식별되는 동작 규칙을 적용함으로써 상기 주어진 차원에 대한 상기 코드벡터를 완전히 재구성하는 단계를 포함하는 것을 특징으로 하는 벡터 양자화에 의해 디지털 신호들을 압축 코딩 또는 디코딩하는 방법.

**청구항 27**

제 25 항에 있어서,

코딩 중에,

\* 상기 단계 C01)은:

C011) 상기 동작 규칙들 및 상기 제 2 세트에 대한 링크들을 나타내며:

- 상기 제 2 세트의 서브세트의 현재 차원의 인덱스,
- 상기 서브세트의 요소의 현재 인덱스, 및
- 주어진 차원의 디셔너리의 코드벡터의 구성을 위한 동작 규칙의 인덱스를 포함하는 인덱스들을 상기 대응 테이블에서 관독하는 단계와;

C012) 그의 현재 차원에 의해 식별된 서브세트에서, 그의 현재 인덱스에 의해 식별된 상기 요소를 관독하는 단계를 포함하며,

\* 상기 단계 C02)에서, 상기 거리는:

- 상기 동작 규칙의 인덱스, 및

- 현재 인덱스에 의해 식별된 서브세트의 요소의 함수로서 추정된 왜곡 기준의 함수로서 계산되며,  
 그에 의해, 단순히 디코딩을 위한 완전한 재구성을 예약함으로써, 상기 단계 C01)에서의 상기 주어진 차원을 갖는 상기 코드벡터의 부분적인 재구성만을 가능하게 할 수 있는 것을 특징으로 하는 벡터 양자화에 의해 디지털 신호들을 압축 코딩 또는 디코딩하는 방법.

**청구항 28**

제 25 항에 있어서,

치환 코드들의 조합에 따른 보조 구조화 특성이 제공되며, 상기 치환 코드들의 조합의 인덱스를 이용하여:

CP1) 입력 신호에 기초하여, 그의 절대 벡터  $|y| = (|y_0|, \dots, |y_k|, \dots, |y_{j-1}|)$ 에 의해 그리고  $\varepsilon_k = \pm 1$ 을 갖는 부호 벡터  $\varepsilon = (\varepsilon_0, \dots, \varepsilon_k, \dots, \varepsilon_{j-1})$ 에 의해 정의되는 입력 벡터  $y = (y_0, \dots, y_k, \dots, y_{j-1})$ 을 형성하는 단계와;

CP2) 리더 벡터  $|\tilde{y}^i|$ 을 획득하기 위해, 치환에 의해, 감소하는 값들에 의해 상기 벡터  $|y|$ 의 구성요소들을 랭크하는 단계와;

CP3) 상기 리더 벡터  $|\tilde{y}^i|$ 의 최근접 이웃  $x^{j'}$ 을 차원  $j$ 의 디셔너리  $D_i^j$ 의 리더 벡터들로부터 결정하는 단계와;

CP4) 상기 디셔너리  $D_i^j$ 의 상기 최근접 이웃  $x^{j'}$ 의 랭크의 인덱스를 결정하는 단계와; 그리고

CP5) 상기 단계 CP4)에서 결정된 상기 인덱스, 상기 단계 CP2)에서 결정된 상기 치환 및 상기 단계 CP1)에서 결정된 상기 부호 벡터에 의존하는 입력 벡터에 코딩과 디코딩중 적어도 하나의 유효 값을 적용하는 단계를 포함하는 것을 특징으로 하는 벡터 양자화에 의해 디지털 신호들을 압축 코딩 또는 디코딩하는 방법.

**청구항 29**

제 25 항에 있어서,

적어도 상기 대응 테이블은 코딩 디바이스와 디코딩 디바이스중 적어도 하나의 메모리에 저장되는 것을 특징으로 하는 벡터 양자화에 의해 디지털 신호들을 압축 코딩 또는 디코딩하는 방법.

**청구항 30**

프로세싱 유닛에 의해 실행될 때, 제 4 항에 따른 방법을 실행하는 명령어들이 수록되어 있는 컴퓨터 판독가능한 매체.

**청구항 31**

프로세싱 유닛에 의해 실행될 때, 제 25 항에 따른 방법을 실행하는 명령어들이 수록되어 있는 컴퓨터 판독가능한 매체.

**청구항 32**

제 6 항에 있어서,

차원  $n-1$  내지 1에 대한 상기 단계들 a'1) 내지 a'3)의 반복 구현을 통해, 차원  $n$ 의 초기 디셔너리에 기초하여 각각의 차원들 1 내지  $N$ 의  $N$ 개의 연속적인 디셔너리들이 획득되는 것을 특징으로 하는 디셔너리를 형성하기 위한 방법.

**청구항 33**

제 14 항에 있어서,

상기 하나 이상의 중간 디셔너리들을 형성하는 코드벡터들의 세트 중 적어도 일부가 업데이트되는 것을 특징으로 하는 디셔너리를 형성하기 위한 방법.

명세서

기술분야

[0001] 본 발명은 오디오, 비디오 신호들 및 더 일반적으로는 멀티미디어 신호들과 같은 디지털 신호들을 저장 및/또는 전송하기 위한 압축 코딩 및/또는 디코딩에 관한 것이다.

배경기술

[0002] 디지털 신호들의 압축에 있어서 매우 일반적인 솔루션은 벡터 양자화이다. 벡터 양자화를 사용하는 제 1 벡터는 코딩될 벡터들의 차원을 부스팅함으로써 더 우수한 성능이 달성될 수 있는, 섀넌(Shannon)에 의해 개발된 블록 코딩 이론에서 발견될 수 있다. 벡터 양자화는 유한 세트로부터 선택된 유사 차원의 벡터에 의해 입력 벡터를 나타내는 것으로 이루어진다. 따라서, M 레벨들(또는 코드벡터들)을 양자화기에 제공함으로써 입력 벡터들의 세트(일반적으로 n 차원들  $R^n$  또는  $R^n$ 의 서브세트를 갖는 유클리드(Euclidian) 실제 공간)로부터  $R^n$ 의 유한 서브세트(Y)로의 비-전단사(non-bijective) 사상을 형성하게 된다. 상기 서브세트(Y)는 M개의 구별되는 요소들을 포함한다:

[0003]  $Y = \{y_1, y_2, \dots, y_M\}$ .

[0004] Y는 재생 알파벳 또는 사전 또는 디렉토리라 칭해진다. Y의 요소들은 "코드벡터들", "코드 워드들", "엑시트 포인트(exit point)" 또는 "대표들"이라 칭해진다.

[0005] 양자화기의 차원당 속도(r)(또는 "분해능")는 다음에 의해 정의된다:

[0006]  $r = \frac{1}{n} \log_2 M$

[0007] 벡터 양자화에서, n개 샘플들의 블록이 차원 n의 벡터로서 처리된다. 상기 벡터는 M개 코드 벡터들로부터, 가장 "답은" 벡터인 코드 벡터를 선택함으로써 코딩된다. 일반적으로, 상기 코드 벡터와 입력 벡터 사이의 거리의 측정을 최소화하는 디셔너리의 요소를 선택하기 위해 상기 디셔너리의 모든 요소들에 대한 소모적인 탐색이 이루어진다.

[0008] 소스 코딩 이론에 따르면, 상기 차원이 너무 크면, 상기 벡터 양자화의 성능은 "소스의 속도-왜곡 경계"란 명칭의 한계에 이르게 된다. 상기 공간의 차원화 외에도, 벡터 양자화는 코딩될 소스의 속성들, 예를 들어 비선형 및/또는 선형 의존성 또는 확률 분포의 형태를 이용할 수 있다. 일반적으로, 벡터 양자화의 디셔너리들은 일반화된 로이드(Lloyd) 알고리즘(GLA라 칭해짐)과 같은 통계적 절차들에 기초하여 설계된다. 잘 알려진 상기 알고리즘은 벡터 양자화의 최적화의 필요 조건들에 기초한다. 코딩될 소스 및 초기 디셔너리의 트레이닝 시퀀스 대표에 기초하여, 상기 디셔너리는 반복적으로 구성된다. 각 반복은 2 단계들을 포함한다:

[0009] - 최근접 이웃의 규칙에 따라 상기 트레이닝 시퀀스의 양자화에 의한 양자화 영역들의 구성, 및

[0010] - (무계중심의 규칙에 따라) 상기 영역들의 무계중심들로 예전 코드벡터들을 교체함에 의한 상기 디셔너리의 개선.

[0011] 이러한 결정성 반복 알고리즘의 로컬 최소치로의 수렴을 피하기 위해, 무계중심을 구성하는 단계 및/또는 등급들을 구성하는 단계에 랜덤성(randomness)을 도입함으로써 시뮬레이트된 어닐링(annealing)의 기술에 의해 주입된 "확률 이완(stochastic relaxation)"이란 명칭의 변형들("확률 K-평균 알고리즘"을 나타내는 SKA를 나타냄)이 제안되었다. 이에 의해 얻어진 통계적 벡터 양자화기들은 임의의 구조를 소유하지 않으며, 그에 의해 계산들 및 메모리 욕심(greedy)의 관점에서 비싼 탐색이 이루어지게 된다. 구체적으로, 상기 코딩 및 저장 둘 다의 복잡성은  $n \cdot 2^{nr}$ 에 비례한다. 상기 벡터들 및 속도의 차원의 함수로서 이러한 지수 증가는 실시간으로 차원들 및 낮은 속도들을 주입할 수 있도록 비구조화 벡터 양자화기들의 이용을 작은 차원들 및/또는 낮은 속도들로 제한한다.

[0012] 개별적으로 샘플들을 양자화하는 스칼라 양자화는 상기 소스 및 선형 의존성의 확률 분포의 형태만을 이용할 수 있기 때문에, 벡터 양자화만큼 효율적이지 않다. 그러나, 스칼라 양자화는 벡터 양자화보다 계산들 및 메모리 관점에서 비용이 덜 비싸다. 또한, 엔트로피 코딩과 관련된 스칼라 양자화는 중간 분해능들에서도 우수한 성능을 달성할 수 있다.

- [0013] 크기 및 차원의 제약들을 회피하기 위해, 상기 기본 벡터 양자화의 여러 변형들이 연구되었으며, 상기 디서너리의 구조 부재를 개선하려 하여 복잡도를 품질의 손상까지 감소시키게 한다. 그러나, 상기 성능/복잡도 절충안이 개선되며, 그에 의해 분해능들 및/또는 차원들의 길이를 상기 벡터 양자화가 계산 또는 메모리 비용 관점에서 효율적으로 적용될 수 있는 정도까지 증가시킬 수 있게 된다.
- [0014] 구조화 벡터 양자화기들의 여러 방식들은 문헌에서 제안되었다. 주요 방식들은 다음과 같다:
- [0015] - 상기 디서너리 상의 계층적 트리 구조를 부과하는 트리 벡터 양자화기: 상기 탐색 절차는 간략화되지만 상기 양자화기는 더 큰 저장 메모리를 필요로 함,
- [0016] - 더 작은 레벨들의 벡터 양자화기들을 캐스캐이드하는 멀티 스테이지 벡터 양자화기: 상기 디서너리들은 크기가 감소되며, 계산 시간 및 메모리 비용에 관하여도 동일함,
- [0017] - 더 작은 차원들 및 크기들의 N 등급의 벡터 양자화기들의 "데카르트 곱(Cartesian product)" 명칭의 벡터 양자화기: 상기 입력 벡터는 N 서브벡터들로 분해되며, 각 서브벡터는 다른 서브벡터들과 독립적으로 양자화됨,
- [0018] - "이득/방위" 벡터 양자화기는 "데카르트 곱"의 특정 경우를 구성한다: 2개의 양자화기들이 제공되는데, 하나는 스칼라 양자화기이고 다른 하나는 벡터 양자화기이며, 이들 양자화기들은 개별적으로, 독립적으로 또는 그렇지 않으면 (정규화 입력 벡터를 고려함으로써) 상기 벡터의 이득(또는 표준) 및 그의 방위를 코딩한다. 벡터 양자화의 상기 타입은 "구형" 벡터 양자화 또는 "극성" 벡터 양자화라 칭해진다,
- [0019] - 상기 "치환 코드" 벡터 양자화기의 코드 벡터들은 리더 벡터의 구성요소들의 치환 및 치환 코드들의 합성(또는 조합)으로의 일반화에 의해 획득된다.
- [0020] 상술한 기술들은 모두 통계적 방법 범주 내에 있다.
- [0021] 또 하나의 근본적으로 다른 방법이 또한 제안되었다. 상기 방법은 포인트들 또는 에러 정정기 코드들의 규칙적인 격자들로부터 발생하는 고도로 구조화된 디서너리들을 이용하는 대수 벡터 양자화이다. 디서너리들의 대수 특성들에 의해, 대수 벡터 양자화기들은 구현하기 간단하며 메모리에 저장될 필요가 없다. 이들 디서너리들의 규칙적 구조를 이용함으로써, 최적 및 고속 탐색 알고리즘들 및 특히 대응하는 코드벡터와 인덱스를 관련시키기 위한 메커니즘의 개발이 가능해진다(예를 들어, 공식을 통해). 대수 벡터 양자화기들은 구현하기에 덜 복잡하며 더 적은 메모리를 필요로 한다. 그러나, 상기 양자화기들은 소스(공간 또는 하이퍼스피어(hypersphere)의 표면)의 균일한 분배만을 위해 최적이다. 상기 균일한 스칼라 양자화기의 일반화에서, 상기 대수 벡터 양자화기는 소위 "압선(companding)" 기술을 통해 상기 소스의 분배를 맞추기 더 어렵다. 또한, 상기 코드벡터들의 지수화 방식(indexation)(또는 넘버링) 및 역 동작(디코딩)은 이들 동작들이 테이블로부터의 간단한 관독에 의해 수행되는 통계적 벡터 양자화기들의 경우보다 더 많은 계산들을 필요로 한다.
- [0022] 가변-차원 양자화의 소정 양상들 및 조우하는 문제점들이 이하에 제시된다.
- [0023] 먼저, 벡터 양자화는 고정 길이의 샘플들의 블록들을 코딩하기 위한 잘 알려진 효율적인 기술이다. 그러나, 디지털 신호 압축의 수많은 애플리케이션들에서, 코딩될 신호는 가변 길이 파라미터들의 시퀀스로 모델링된다. 음성 또는 오디오 코더들("MBE" 코더, 고조파 코더, 사인파 코더, 변환 기반 코더, 원형 파형의 보간에 기초한 코더)과 같은 많은 멀티미디어 코더들의 설계를 위해 가변 차원의 이들 벡터들의 효율적인 압축이 중요하다.
- [0024] 사인파 코더들에서, 추출된 사인 곡선들의 수는 상기 신호에서 검출된 사인파 스파이크들의 수에 의존하며, 그 수는 오디오 신호의 본질적인 기능에 따라 시간의 경과에 의해 변동한다.
- [0025] 또한, 수많은 음성 압축 기술들이 상기 신호의 장기 주기성을 이용한다. 대화자의 기본 주기의 고조파인 주파수들의 세트의 스펙트럼 구성요소들이 코딩되는 고조파 코더들의 경우가 그와 같은 경우이다. 상기 기본 주기가 대화자에 따라(전형적으로, 남성보다 더 높은 성대의 진동 주파수를 갖는 아이들), 그리고 시간의 경과에 따라 변동하며, 양자화된 구성요소들의 수가 프레임 마다 시간에 따라 변화하기 때문에, 스펙트럼 고조파 스파이크들의 수가 상기 기본 주파수에 역으로 비례한다.
- [0026] 원형 파형들이 상기 피치의 주기와 동일한 길이의 세그먼트들에 걸쳐 추출되며, 그에 따라 일시적으로 가변하는 PWI 코더들("원형 파형 보간"이라 나타냄)에 대한 경우가 그와 같은 경우이다. PWI 코더들에서, 가변 길이의 파형들의 양자화는 동일한 가변 길이의 2개 파형들: REW 파형("급속 전개 파형") 및 SEW 파형("저속 전개 파형")으로 분해되는 표준 파형 및 이득(또는 "자승 평균 평방근(Root-Mean-Square)"을 나타내는 "RMS")을 개별적으로 코딩함으로써 이루어진다. 고정 길이의 프레임에 대해, 원형의 수가 가변이며, REW 및 SEW 파형들의 차원과 같

이 이득들, REW 및 SEW의 수도 마찬가지로 가변이다.

- [0027] 변환-기반 오디오 코더들과 같은 다른 타입들의 코더들에서, 고정-길이 프레임 길이들에 걸쳐 획득된 변환 계수들의 수가 부과되지만 대개 이들 계수들을 그들의 양자화를 위해 주파수 대역들로 그룹화하는 것이 일반적이다. 전형적으로, 이러한 분할은 귀의 임계 대역들에 후속하여 인간의 청취의 사이코-음향 (psychoacoustic) 특성들을 이용하도록 동일하지 않은 폭들의 대역들로 행해진다. 변환 계수들의 이들 벡터들의 차원의 변동 길이는 광대역 코더에서(50 Hz - 7000 Hz) 3내지(저주파수 대역들에 대해) 15까지(고주파수 대역들에 대해) 변동하며, 심지어 FM 대역 코더(20 Hz - 16000 Hz 가청 범위를 커버함)에서 최대 24까지 변동한다.
- [0028] 이론적으로, 가변 차원의 최적 벡터 양자화기는 각각이 입력 벡터의 가능한 차원에 대한, 고정 차원의 디셔너리들의 세트를 이용한다. 예를 들어, 고조파 코더들에서, 60 내지 450 Hz의 피치 주기에 대해, 전화 대역에서의 고조파 스파이크들의 수는 높은-피치 음성들(아이들)에 대한 7로부터 낮은-피치 음성들(남성)에 대한 52까지 변동하며, 이는 46(46=52-7) 벡터 양자화기들을 구성하고, 메모리에 두며 구현할 필요가 있다. 각 디셔너리의 설계는 상기 입력 벡터들의 통계들을 정확하게 표현하는데 충분히 긴 학습 시퀀스(learning sequence)를 필요로 한다. 또한, 모든 디셔너리들의 저장은 실행 불가능한 것으로 또는 메모리 비용이 매우 비싼 것으로 판단된다. 따라서, 가변 차원의 경우에 메모리 저장 및 트레이닝 시퀀스들의 제약들에 따르면 벡터 양자화의 장점들을 활용하기가 어렵다는 것을 알 수 있다.
- [0029] 이하에는 가변 분해능 및 조우하는 문제점들을 갖는 양자화의 소정의 양상들이 제시되어 있다.
- [0030] 먼저 지적되는 바와 같이, 상기 입력 신호의 가변성은 전적으로 코딩될 파라미터들의 수의 변동을 통해서 나타나는 것이 아니라, 주어진 품질에 대해 전송될 이진 정보량의 변동을 통해 나타난다. 예를 들어, 음성, 개시(onset), 음성 사운드 및 무음성 사운드는 하나에 대해 동일한 속도 및 동일한 품질을 요구하지 않는다. 비교적 예측 불가능한 개시는 더 안정적이며 상기 속도를 감소시킬 수 있는 "예측기들"에 의해 그의 정지성이 이용될 수 있는 음성 사운드들보다 높은 속도를 필요로 한다. 마지막으로, 상기 무음성 사운드들은 높은 코딩 정밀도를 필요로 하지 않으며, 따라서 아주 작은 속도만을 필요로 한다.
- [0031] 음성 또는 비디오와 같은 멀티미디어 신호들의 특성들의 일시적 변동을 이용하기 위해, 가변 속도 코더들을 설계하는 것이 현명하다. 이들 가변 속도 코더들은 특히 인터넷, ATM 또는 다른 것들과 같은 격자들, 패킷와이즈(packetwise)를 통한 통신에 적합하다.
- [0032] 구체적으로, 패킷 스위칭은 더 유연한 방식으로 정보 비트들을 취급 및 처리할 수 있게 하며, 평균 속도를 감소 시킴으로써 상기 채널의 용량을 증가시킬 수 있게 한다. 가변 속도 코더들의 이용은 시스템의 혼잡을 제거하며 및/또는 액세스 조건들의 다이버시티를 수용하는 효율적인 수단이다.
- [0033] 멀티미디어 통신에서, 가변 속도 양자화는 또한 다음과 같은 것들 사이에서 속도의 분배를 최적화할 수 있게 한다:
- [0034] - 상기 소스 및 채널 코딩들: AMR("적응형 멀티 속도(Adaptive Multi Rate)")의 개념에서와 같이, 상기 속도는 상기 트래픽 및 채널 에러 조건들에 동적으로 적응되도록 각 20-ms 프레임마다 스위칭될 수 있다. 상기 음성의 전체 품질은 상기 채널이 품질 저하되면 상기 소스의 코딩에 대한 속도를 감소시키면서 에러들에 대한 우수한 보호를 보증함으로써 개선된다;
- [0035] - 다양한 타입들의 매체 신호들(비디오 회의 애플리케이션들에서의 음성 및 비디오와 같음);
- [0036] - 하나 및 동일한 신호의 여러 파라미터들: 변환 기반 오디오 코더들에서, 예를 들어, 스펙트럼 포락선과 계수들의 다양한 대역들 사이에서 동적으로 비트들을 분배하는 것이 일반적이다. 종종, 상기 포락선의 엔트로피 코딩이 먼저 수행되며 그의 목적은 가변 길이 코드들을 상기 코드 워드들에 할당함으로써 상기 코드 워드들의 불균일한 분배를 이용하는 것이며, 가장 확률이 높은 코드 워드들은 가장 확률이 낮은 코드워드들보다 더 짧은 길이를 갖으며, 그에 의해 상기 코드 워드들의 평균 길이의 최소화를 발생시킨다. 또한, 인간 귀의 사이코-음향 특성들을 이용하기 위해, 잔여 (가변) 속도는 그들의 지각적 중요도의 기능으로서 계수들의 주파수 대역들에 동적으로 할당된다.
- [0037] 멀티미디어 코딩(오디오 및 비디오 등)의 새로운 애플리케이션들은 차원 및 속도 둘 다를 고려한 고도로 유연한 양자화를 필요로 한다. 고품질을 달성할 수 있게 하기 위해 더한 속도들의 범위를 갖고, 이들 멀티차원 및 멀티분해능 양자화기들은 고분해능을 목표로 해야 한다. 새로운 기술들의 프로세싱 능력들 및 메모리 용량들의 증가에도 불구하고, 달성되는 성능 그 자체에는 이들 벡터 양자화기들에 의해 발생하는 복잡도 장애가 남아있다.

- [0038] 이하에서 알 수 있는 바와 같이, 제안된 소스 코딩 기술들의 대부분은 가변 차원에 관련된 문제점들 또는 가변 분해능에 관련된 문제점들을 해결하는데 목적을 둔다. 현재 제안된 기술들은 이들 2가지 문제점들을 거의 공통으로 해결할 수 없다.
- [0039] 알려진 가변 차원을 갖는 벡터 양자화와 관련하여, 코딩될 파라미터들의 차원의 가변성은 그 자체로 벡터 양자화의 이용에 대한 장애를 구성한다. 따라서, 상기 변환 기반 코더의 제 1 버전들은 로이드-맥스(Lloyd-Max) 스칼라 양자화기들을 사용한다. 본 출원인에 의해 개발된 "TDAC"란 명칭의 이러한 타입의 코더는 특히 다음에 설명되어 있다:
- [0040] - 1994년 11월 IEEE Trans. Commun, Vol. 42, No 11, pp. 3010-3019에 Y. Mahieux, J.P. Petit에 의해 "64 kbit/s의 고품질 오디오 변환 코딩".
- [0041] 가변 차원 벡터 양자화의 문제점을 해결하기 위해 다른 솔루션들이 제안되었다. 상기 "IMBE" 코더는 가변 이진 할당들 및 스칼라/벡터 하이브리드 양자화를 갖는 복잡한 코딩 방식을 이용한다.
- [0042] 가변 차원의 벡터들을 양자화하는데 매우 공통적으로 사용되는 방법은 가변 차원의 벡터를 사전-프로세싱하여 양자화전에 고정 차원의 또 하나의 벡터로 변환하는 것에 있다. 차원 변환과 관련된 상기 벡터 양자화 기술의 여러 변형들이 존재한다(이러한 타입의 벡터 양자화는 "차원 변환 벡터 양자화(Dimension Conversion Vector Quantization)"를 나타내는 DCVQ라 칭해진다).
- [0043] 제안된 다양한 차원 변환 절차들 중에, 특히 언급되는 것은 절사(truncation), 서브샘플링, 보간, "길이 워핑(length warping)"이다.
- [0044] 사인과 음성 코더들 또는 MBE들에 대해, 제안된 바와 같이 스펙트럼 계수들은 고정 차수의 모든-극성 모델에 의해 근사화되며, 상기 모델의 파라미터들의 고정 차원의 벡터 양자화가 수행된다. 비평방 매트릭스 변환에 의해 벡터 양자화의 또 하나의 기술은 고정 차원(K)(K<L)의 벡터 양자화와 비평방 매트릭스 선형 변환(LxK)을 결합함으로써 가변 차원(L)의 벡터 양자화의 문제점을 해결한다.
- [0045] 또한, 고정 차원 K의 벡터 양자화기를 여전히 사용하는 차원 변환과 관련된 또 하나의 타입의 벡터 양자화가 존재하지만, 상기 차원 변환은 입력 벡터로서 동일한 차원을 갖는 코드벡터들을 획득하도록 상기 코드벡터들에 적용된다.
- [0046] 차원 변환과 관련된 벡터 양자화의 결합은 총 왜곡이 2가지 구성요소들: 양자화로 인한 것과 차원 변환으로 인한 것을 갖는다는 것이다. 차원 변환으로 인한 상기 왜곡을 피하기 위해, 가변 차원의 또 하나의 벡터 양자화 방법은 차원 L(K<L)의 "하부" 벡터의 구성요소들의 서브셋으로 형성된 가변 차원 L의 각 입력 벡터를 고려하며, 그럼에도 불구하고 상기 입력 벡터들의 차원들의 전체 길이를 커버하는 고정 차원 K의 단지 단일의 "범용" 디서너리를 설계 및 이용하는데 있으며, 상기 입력 벡터 사이의 일치성은 선택기에 의해 영향받는다. 그러나, 더 낮은 차원들의 다른 모든 디서너리들을 포함하는 상기 "범용" 디서너리는 최저 차원들에 대해서는 최적으로 나타나지 않는다. 특히, 차원당 최대 분해능( $r_{max}$ )은 저장 제약 및 파라미터들의 벡터당 속도에 의해 제한된다. 크기  $2^{Kr_{max}}$ 의 디서너리에 대해, 상기 디서너리를 저장하는데 필요한 메모리 양은  $K2^{Kr_{max}}$  값들이며 그의 파라미터들의 벡터당 속도는  $Kr_{max}$ 이다. 따라서, 하나 및 동일한 크기의 디서너리에 대해(그리고 파라미터들의 벡터당 및 프레임당 하나 및 동일한 속도에 대해), 차원 L(L<K)의 벡터는 K/L 배만큼 더 큰 분해능(또는 차원당 속도)을 가질 수 있으며, K/L배만큼 저장될 정보량이 더 작다.
- [0047] 알려진 가변 분해능을 갖는 벡터 양자화와 관련하여, 가변 차원을 갖는 벡터 양자화의 경우에 관한 한 간단한 솔루션은 예를 들어 TDAC 변환 기반 코더의 제 1 버전들에서와 같이, 스칼라 양자화를 이용하는 것이다.
- [0048] 그러나, 샘플당 정수 분해능의 이용은 동적 이진 할당 절차의 효율성을 저해하는 계수들의 대역당 분해능의 조악한 입도를 수반한다. 따라서 공동 이진 트레인으로서 코딩된 인덱스들을 배열하기 위한 절차와 조합하여, 홀수 정수개의 재구성 레벨들을 갖는 스칼라 양자화기들의 이용이 제안되었다. 상기 이진 할당 절차에 대한 더 유리한 가능한 분해능의 정교한 입도에 의해, 상기 인덱스들을 조합하는 알고리즘의 복잡도를 희생하여 품질을 개선할 수 있으며, 상기 알고리즘은 속도의 관점에서 효율적인 이진 트레인에의 배치를 위해 필요하다. 그럼에도 불구하고, 더 큰 수의 계수들을 갖는 상승 주파수 대역들에 대해, 스칼라 양자화로 인하여 샘플당 정수개의 레벨들의 제약이 대역 당 분해능의 너무 조악한 입도를 통해 여전히 나타난다.
- [0049] 벡터 양자화는 샘플당 정수 레벨들의 수의 이러한 제약을 회피할 수 있게 하며, 이용가능한 분해능들의 미세한

입도를 허용한다. 다른 한편, 상기 벡터 양자화의 복잡도는 종종 이용가능한 속도들의 수를 제한한다. 예를 들어, 잘 알려진 ACELP 기술에 기반한 AMR-NB 다중속도 음성 코더는 12.2 kbit/s 내지 4.75 kbit/s의 범위에 있는 8개의 고정 속도들을 포함하며, 각각은 소스 코딩과 채널 코딩 사이의 속도의 서로 다른 분배에 의해 에러들에 대한 서로 다른 보호 레벨을 갖는다. 상기 ACELP 코더의 파라미터들의 각각에 대해(LSP, LTP 지연기들, 여기 이득들, 고정 여기), 서로 다른 분해능의 디서너리들이 구성되었다. 그러나, 이들 파라미터들의 각각에 대해 이용가능한 속도들의 수는 비대수 벡터 양자화기들의 저장 복잡도에 의해 제한된다. 또한, 6.60 내지 23.85 kbit/s의 범위에 있는 9개 속도들을 포함하는 AMR-WB 다중속도 코더에서, 상기 속도들의 변동은 본질적으로 저장을 필요로 하지 않는 대수 여기 디서너리들에 의해 보증된다. 8개의 디서너리들이 존재하며, 따라서 고정 여기에 대한 8개 속도들이 있는 한편, 확률 디서너리들을 이용하는 다른 파라미터들(LSP, 이득들, 절대 및 차동 지연들)은 단지 2개의 가능한 속도들을 갖는다.

[0050] 나타난 바와 같이, AMR 다중속도 코더들에서 사용되는 확률 벡터 양자화기들은 제약된 구조를 갖는 벡터 양자화기들(데카르트 곱 및 다중 스테이지들)이다. 가변 속도 양자화기의 큰 계열(family)은 사실 다중 스테이지들, 데카르트 곱들을 갖는 이미 언급된 양자화기들과 같은 제약된 구조 벡터 양자화기들에 기초할 뿐 아니라 트리-기반 벡터 양자화기들에 기초할 수 있다. 가변 속도 코딩에 대한 이들 트리-기반 벡터 양자화기들의 이용은 수많은 연구 과제를 형성하였다. 상기 이진 트리-기반 벡터 양자화기가 처음으로 도입되었다. 이것은 상기 트레이닝 시퀀스의 중심인, "루트" 노드에 기초하여 상기 무게중심들의 연속적인 분배들에 의해 벡터 양자화기를 설계하는 LBG 알고리즘으로부터 도출된다. 프루닝(pruning)에 기초하여 또는 반대로 왜곡과 같은 속성들에 따라 상기 트리의 어떤 노드들을 분기(ramify)하는데 기초한 변형 트리-타입 벡터 양자화기들이 제안되었으며, 그들의 모집단은 비이진 및/또는 비평형 트리-기반 벡터 양자화기들을 발생시킨다.

[0051] 도 1a 및 1b는 트리-구조화 벡터 양자화기들을 나타낸다. 특히, 도 1a는 평형 이진 트리를 나타내는 반면, 도 1b는 비이진 및 비평형 트리를 나타낸다.

[0052] 다중-분해능 벡터 양자화기들은 원하는 다양한 분해능들에 대응하는 노드들의 수를 선택함으로써, 트리-타입 벡터 양자화기에 기초하여 용이하게 구성된다. 상기 트리-타입 계층적 구조는 매력적이며 상기 탐색 절차를 간략화한다. 다른 한편, 상기 중간 레벨들의 모든 노드들을 통해 상기 루트-노드로부터 상기 단말 노드들로의 상기 트리의 모든 노드들이 저장되어야 하기 때문에, 이것은 차선의(suboptimal) 탐색 및 필요한 메모리들에서의 상당한 증가와 관련된다. 또한, 더 낮은 분해능의 디서너리의 노드들의 세트가 더 높은 분해능의 디서너리들에 포함되지 않기 때문에, 상기 벡터 양자화기의 속도의 증가 함수로서 양자화 에러의 감소가 국부적으로 보장되지 않는다.

[0053] 또한, 대수 코드들에 기초하여 가변 분해능 양자화기들, 특히 차원 8의 규칙적인 고셋(Gosset) 격자의 구면 코드들의 서브세트들을 이용하는 EAVQ 내장 대수 벡터 양자화기들을 구성하는 방법이 알려져 있다.

[0054] 상기 문서에서:

[0055] - "A 16, 24, 32 kbit/s wideband speech codec based on ACELP" by P. Combescure, J. Schnitzler, K. Fischer, R. Kircherr, C. Lamblin, A. Le Guyader, D. Massaloux, C. Quinquis, J. Stegmann, P. Vary, in Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 1, pp 5-8, 1999,

[0056] 이와 같은 내장된 대수 벡터 양자화 방법은 다양한 차원들의 대수 코드들을 이용하여 가변 차원 양자화로 확장되었다. EAVQ 양자화의 이러한 일반화는 가변 분해능들에서의 가변 차원의 벡터들을 양자화할 수 있게 하더라도, 상기 방법은 결함들을 갖는다.

[0057] 상기 입력 벡터들의 분배는 균일해야 한다. 그러나, 이러한 제약에 대해 상기 소스의 분배를 적응시키는 것은 매우 어려운 작업이다. 규칙적 격자들에 기초한 대수 양자화기들의 설계는 또한 원하는 다양한 분해능들을 획득하도록 다양한 규칙적 격자들의 영역들을 절사 및 적응시키며, 상기 다양한 차원들에 대해서도 그렇게 해야 하는 문제점을 발생시킨다.

**발명의 상세한 설명**

[0058] 본 발명의 목적은 이와 같은 상황을 개선하는 것이다.

[0059] 본 발명의 목적들 중 하나는 일반적으로 가변 차원의 벡터들의 가변 속도 양자화의 문제점에 대한(특히 저장 메모리의 관점에서) 효율적이고 경제적인 솔루션을 제안하는 것이다.

- [0060] 본 발명의 또 다른 목적은 비제한 방식으로, 고조파 코더들의 스펙트럼 진폭들 및/또는 주파수 코더들의 변환 계수들의 양자화를 이용하여 디지털 신호들, 특히 음성 및/또는 음성 신호들의 코딩 및 디코딩을 유용하게 수용하는 벡터 양자화를 제안하는 것이다.
- [0061] 이를 위해, 가변 차원의 코드벡터들을 포함하며, 가변 분해능을 정의하는 가변 속도에서의 벡터 양자화에 의해 디지털 신호들의 압축 코딩 및/또는 디코딩을 위한 장치에 사용되는 디셔너리를 제안하며, 상기 디셔너리는:
- [0062] - 한편으로, 주어진 차원에 대해, 증가하는 분해능을 가지며, 상호 간에 내장되는 디셔너리들(inter-embedded dictionaries)과,
- [0063] - 및, 다른 한편으로 주어진 차원에 대해:
- [0064] ● 소정의 삽입 규칙들의 유한 집합에 따라 실수의 유한 세트로부터 취해진 요소들을 더 낮은 차원의 디셔너리들의 코드벡터들로 삽입함으로써 구성되는 코드벡터들로 이루어진 제 1 세트와,
- [0065] ● 상기 삽입 규칙들의 집합에 따라 상기 유한 세트의 요소들의 더 낮은 차원의 코드벡터들로의 삽입에 의해 얻어질 수 없는 코드벡터들로 이루어지는 제 2 세트의 조합을 포함한다.
- [0066] 바람직하게는, 상기 삽입 규칙들의 집합은 주어진 벡터 위치에서의 구성요소를 가장하는 실수들의 유한 세트의 단일 요소를 삽입하는 것으로 이루어진 기본 규칙들에 기초하여 공식화된다.
- [0067] 각 기본 규칙은 바람직하게는:
- [0068] - 상기 유한 세트의 요소의 랭크,
- [0069] - 및 삽입 위치를 나타내는 2개의 양의 정수들의 쌍에 의해 정의된다.
- [0070] 이해되는 바와 같이, 특성화된 삽입 규칙들이 판독되며, 본 발명의 의미 내의 디셔너리의 실제 구조로부터 직접 추론된다.
- [0071] 물론, 순수하게 가역 방식으로, 더 낮은 차원  $N(N < N')$ 을 달성하도록 주어진 차원  $N'$ 의 유한 세트의 하나 이상의 요소들을 삭제하는 것으로 이루어진 삭제 규칙들을 정의할 수 있다.
- [0072] 본 발명은 또한 주어진 차원에 대해, 본 발명에 따른 디셔너리를 형성하는 방법에 관한 것이며, 상기 디셔너리는:
- [0073] a) 소정의 삽입/삭제 규칙들의 유한 집합에 따라 실수들의 유한 세트로부터 취해진 더 낮은/더 높은 차원 요소들의 디셔너리들의 코드벡터들에/로부터 삽입/삭제에 의해 형성된 코드벡터들로 이루어진 제 1 세트와,
- [0074] b) 상기 주어진 차원에 대해, 적어도 상기 제 1 세트가 구성되어 있는 제 1 중간 디셔너리와, 그리고
- [0075] c) 적어도 하나의 주어진 분해능의 이용에 상기 디셔너리를 적응시키기 위해, 상기 중간 디셔너리에 기초하여, 증가/감소 분해능의 디셔너리들의 내장/간략화에 의해 구성되는 제 2 최종 디셔너리를 포함하며, 상기 증가하는 분해능들의 디셔너리들은 최소 분해능의 디셔너리로부터 최대 분해능의 디셔너리까지 상호 간에 내장된다.
- [0076] 물론, 상기 용어 "세트 A를 세트 B에 내장"은 상기 세트 A가 상기 세트 B에 포함되는 사실을 의미하는 것이다. 또한, 상기 용어 "세트 B를 획득하기 위한 세트 A의 간략화"는 상기 세트 A가 상기 세트 B를 포함하는 사실을 의미하는 것이다.
- [0077] 변형 또는 보충에 의해, 이해되는 바와 같이 한편으로 단계들 a) 및 b)와, 다른 한편으로 단계 c)는 상기 디셔너리를 코드벡터들의 주어진 차원  $N$ 으로의 이용에 적응시키기 위해 가역될 수 있다.
- [0078] 이 경우에:
- [0079] - 단계 c)에서, 여전히 차원  $N'$ 이지만 더 높은/더 낮은 분해능  $r_N$ 의 제 1 중간 디셔너리는 상기 제 1 디셔너리의 분해능  $r_N$ 을 실질적으로 획득하도록 증가/감소 분해능들의 디셔너리들을 내장/간략화함으로써 분해능  $r_N$  및 차원  $N'$ 의 초기 디셔너리에 기초하여 구성되고,
- [0080] - 단계 a)에서, 주어진 차원  $N$ 을 획득하기 위해, 소정의 삽입/삭제 규칙들의 유한 집합에 따라 실수들의 유한 세트들로부터 취해진 상기 주어진 차원  $N$  요소들보다 낮은/높은 차원  $N'$ 의 제 1 디셔너리의 코드벡터들로/로부터 삽입/삭제함으로써 형성된 코드벡터들로 이루어진 제 1 세트가 구성되며, 그리고

- [0081] - 단계 b)에서, 상기 분해능  $r_N$ 에 대한 최종 적응의 가능한 단계에 후속하여 상기 주어진 차원  $N$ 에 대해 적어도 상기 제 1 세트를 포함하는 제 2의 최종의, 디셔너리가 구성된다.
- [0082] 연속적인 차원들을 증가함으로써 단계 a)를 구현할 수 있다. 이 경우에, 주어진 차원  $N$ 에 대해:
- [0083] a0) 상기 주어진 차원  $N$ 보다 낮은 초기 차원  $n$ 의 초기 디셔너리가 획득되고,
- [0084] a1) 소정의 삽입 규칙들의 유한 집합에 따라 실수들의 유한 세트로부터 취해진 초기 디셔너리 요소들의 코드벡터들에 삽입함으로써 형성된 차원  $n+i$ 의 코드벡터들로 이루어진 제 1 세트가 구성되고,
- [0085] a2) 상기 삽입 규칙들의 집합으로 상기 유한 세트의 요소들의 초기 디셔너리의 코드벡터들로의 삽입에 의해 획득되지 않을 수 있는 차원  $n+i$ 의 코드벡터들로 이루어진 제 2 세트가 제공되고,
- [0086] a3) 상기 제 1 세트와 상기 제 2 세트의 조합을 포함하는 차원  $n+i$ 의 중간 디셔너리가 구성되며,
- [0087] 그리고, 단계 a1) 내지 a3)이 ( $i=1$ )인 경우에, 상기 주어진 차원  $N$ 까지 초기 디셔너리를 가장하는 상기 중간 디셔너리로 많아야  $N-n-1$ 회로 반복된다.
- [0088] 또한, 연속하는 차원들을 감소시킴으로써 단계 a)를 구현할 수 있다. 이 경우에, 주어진 차원  $N$ 에 대해:
- [0089] a'0) 상기 주어진 차원  $N$ 보다 높은 초기 차원  $n$ 의 초기 디셔너리가 획득되고,
- [0090] a'1) 소정의 삭제 규칙들의 유한 집합에 따라, 차원  $n-i$ 의 제 1 세트가 차원  $n$ 의 디셔너리로부터 차원  $n-i$ 의 가능한 코드벡터들의 선택 및 추출에 의해 구성되고,
- [0091] a'2) 상기 초기 디셔너리의 코드벡터들로부터, 상기 삭제 규칙들의 집합으로 상기 유한 세트의 요소들의 삭제에 의해 단순하게 획득될 수 없는 차원  $n-i$ 의 코드벡터들로 이루어진 제 2 세트가 제공되고,
- [0092] a'3) 상기 제 1 세트 및 상기 제 2 세트의 조합을 포함하는 차원  $n-i$ 의 중간 디셔너리가 구성되며,
- [0093] 그리고 단계들 a'1) 내지 a'3)은 ( $i=1$ )인 경우에, 상기 주어진 차원  $N$ 까지 초기 디셔너리를 가장하는 상기 중간 디셔너리로 많아야  $n-N-1$ 회로 반복된다.
- [0094] 연속하는 차원들 1 내지  $N$ 의 복수의  $N$  디셔너리들을 획득하기 위해, 바람직하게는 차원  $n(n < N)$ 의 초기 디셔너리에 기초하여, 차원  $n+1$  내지  $N$ 에 대한 단계들 a1) 내지 a3)의 반복 구현을 통해, 그리고 차원  $n-1$  내지 1에 대한 단계들 a'1) 내지 a'3)의 반복 구현을 통해 단계들 a1) 내지 a3) 및 단계들 a'1) 내지 a'3)를 조합할 수 있다.
- [0095] 따라서, 최대 차원의 디셔너리가 차원  $N$ 을 갖는  $N$  디셔너리들의 전부 또는 일부가 얻어진다.
- [0096] 연속적인 차원들의 디셔너리들의 구성을 위해 이용되는 삽입/삭제 규칙들의 집합 및 유한 세트가 다음과 같이 정의될 수 있다:
- [0097] - 사전에(a priori), 즉 상기 디셔너리를 구성하기 전에 양자화될 소스의 분석에 의해,
- [0098] - 또는 사후에(a posteriori), 즉 디셔너리들의 구성 후에, 바람직하게는 연속적인 분해능들의 디셔너리들의 내장/간략화에 의해, 상기 구성은 그에 따라 구성된 이들 디셔너리들의 통계적 분석 후에 이루어진다.
- [0099] 지시되는 바와 같이, 양자화될 소스는 바람직하게는 러닝(learning) 시퀀스에 의해 모델링되고, 상기 유한 세트 및 삽입/삭제 규칙들의 집합의 "사전(a priori)" 정의는 바람직하게는 상기 소스의 통계적 분석에 의해 이루어진다. 상기 유한 세트는 바람직하게는 양자화될 소스의 단일차원 확률 밀도의 추정에 의해 선택된다.
- [0100] 상기 유한 세트 및 상기 삽입 규칙들의 사전 및 사후 정의들을 조합함으로써:
- [0101] - 삽입/삭제 규칙들의 제 1 집합 및 제 1 세트가 러닝 시퀀스의 분석에 의해 사전에 유용하게 선택될 수 있어, 하나 이상의 중간 디셔너리들을 형성하고,
- [0102] - 삽입/삭제 규칙들의 제 1 집합 및/또는 제 1 세트의 적어도 일부분이 상기 하나 이상의 중간 디셔너리들의 사후 분석에 의해 업데이트되고,
- [0103] - 적절하게, 상기 하나 이상의 중간 디셔너리들을 형성하는 코드벡터들의 세트 중 적어도 일부분이 업데이트된다.
- [0104] 바람직하게는, 주어진 분해능으로의 적응 단계 c)는 증가하는 분해능을 획득하기 위해, 다음의 동작들을 포함한

다:

- [0105] c0) 상기 주어진 분해능  $r_N$ 보다 낮은 초기 분해능  $r_n$ 의 초기 디셔너리가 얻어지고,
- [0106] c1) 상기 초기 디셔너리에 기초하여, 초기 분해능  $r_n$ 보다 높은 분해능  $r_{n+1}$ 의 중간 디셔너리가 구성되며,
- [0107] c2) 동작 c1)은 상기 주어진 분해능  $r_N$ 이 달성될 때까지 반복된다.
- [0108] 유용하게는, 동작 c1)의 각 반복에 대해, 등급들 및 중심들의 구조가 제공되며, 적어도 현재 분해능  $r_i$ 보다 높은 분해능의 디셔너리들에 속하는 중심들이 재계산되고 업데이트된다. 또한, 현재 분해능  $r_i$ 보다 낮은 분해능의 디셔너리들에 속하는 중심들은 바람직하게는 더 낮은 분해능의 모든 디셔너리들의 총 왜곡들이 하나의 업데이트로부터 다음으로 감소하는 경우에만 바람직하게 업데이트된다.
- [0109] 보조 또는 변형에 의해, 단계 c)는 감소하는 분해능들을 달성하기 위해 다음의 동작들을 포함한다:
- [0110] c'0) 상기 주어진 분해능  $r_N$ 보다 높은 초기 분해능  $r_n$ 의 초기 디셔너리가 획득되고,
- [0111] c'1) 상기 초기 디셔너리에 기초하여, 상기 초기 분해능  $r_n$ 보다 낮은 분해능  $r_{n-1}$ 의 중간 디셔너리가 소정의 기준에 따라 정렬된 여러 서브세트들로의 상기 초기 디셔너리의 분할에 의해 구성되며,
- [0112] c'2) 동작 c'1)은 상기 주어진 분해능  $r_N$ 이 획득될 때까지 반복된다.
- [0113] 유용하게는, 상기 분할은 구현된 삽입/삭제 규칙들 중 적어도 일부분을 이용하여, 단계들 a) 및 b)의 의미 내의 제어된 확장에 의해 상기 부분 합성을 이용할 수 있다.
- [0114] 상기 분해능들  $r_1$  및  $r_N$  사이의 중간 분해능  $r_n$ 의 초기 디셔너리에 기초하여 각각의 분해능들  $r_1$  내지  $r_N$ 의 복수의  $N$  연속 디셔너리들을 획득하기 위해, 유용하게는 상기 증가하는 분해능들  $r_{n+1}$  내지  $r_N$ 에 대해 단계 c1)의 반복을 구현할 수 있으며, 상기 감소하는 분해능들  $r_{n-1}$  내지  $r_1$ 에 대해 단계 c'1)의 반복을 구현할 수 있다.
- [0115] 이해되는 바와 같이, 삽입/삭제 규칙들의 집합 및 유한 세트는 원하는 차원들 및 분해능들의 본 발명의 의미 내의 디셔너리를 형성하기 위해, 여러 분해능들 및 차원들의 획득된 디셔너리들의 통계의 사후 연구를 통해 유용하게 선택될 수 있다.
- [0116] 본 발명에 의해 산출되는 장점들 중 하나에 따르면, 상기 코딩/디코딩의 구현을 위해 요구되는 메모리 저장량이 상당히 감소할 수 있다. 구체적으로, 유용하게는 각각 인덱스에 의해 식별되는 삽입/삭제 규칙들의 상기 집합이 메모리에 저장되며, 주어진 차원에 대해:
- [0117] - 상기 제 2 세트는 상기 주어진 차원보다 더 낮은/더 높은 차원의 코드벡터들에 상기 삽입/삭제 규칙들을 적용함으로써 획득될 수 없는 코드벡터들로 이루어지며,
- [0118] - 상기 삽입/삭제 규칙들의 인덱스들 및 상기 제 2 세트의 요소들을 식별하는 인덱스들을 이용하여, 적어도 하나의 대응 테이블이 주어진 차원의 디셔너리의 임의의 코드벡터를 재구성하게 할 수 있다.
- [0119] 따라서, 상기 제 2 세트의 요소들 및, 이들 요소들과 상기 관련 삽입/삭제 규칙들에의 액세스를 위한 대응 테이블의 링크들을 단순히 저장함으로써, 상기 주어진 차원에 대한 디셔너리의 완전한 저장이 회피된다.
- [0120] 따라서, 이해되는 바와 같이, 주어진 차원에 대해 상기 제 2 세트는 유용하게는 상기 주어진 차원보다 낮은 차원들의 제 2 서브세트들로 이루어질 수 있다.
- [0121] 일 실시예에서, 상기 삽입/삭제 메커니즘 자체는 프로그램 루틴을 가장하여 저장될 수 있는 한편, 주어진 삽입/삭제 규칙에 대해 상기 삽입/삭제 파라미터들은 상기 주어진 삽입/삭제 규칙과 조합하여, 일반적인 대응 테이블(원칙적으로 상술한 대응 테이블과 다름)에 저장될 수 있다.
- [0122] 바람직하게는, 상기 대응 테이블들은,
- [0123] - 상기 제 2 세트의 현재 차원,
- [0124] - 상기 제 2 세트의 요소의 현재 인덱스,
- [0125] - 삽입/삭제 규칙 인덱스를 나타내는 3개의 정수 스칼라 값들의 도표 작성을 통해, 현재 차원의 제 2 세트에서

의 현재 인덱스들의 요소들에 기초하여 재구성될 수 있는 주어진 차원의 디셔너리의 코드벡터의 각 인덱스에 대해, 미리 작성된다. 이 삽입/삭제 규칙은 적어도 상기 현재 인덱스 및 상기 현재 차원에 대응하는 요소에 삽입/삭제를 적용함으로써, 소정 차원의 디셔너리의 상기 코드 벡터를 재구성하는 것에 적어도 기여한다.

- [0126] 이들 후자의 특성들은 유용하게는 이하에 설명되는 바와 같은 압축 코딩/디코딩 방법으로 구현될 수 있다.
- [0127] 이러한 관점에서, 본 발명은 또한 가변 분해능을 정의하는 가변 속도로의 벡터 양자화에 의해, 디지털 신호들의 압축 코딩/디코딩에 대한 상술한 단계들의 구현을 통해 얻어지는, 본 발명에 따른 디셔너리를 이용하기 위한 것이다. 특히, 주어진 차원  $j$ 에서의 입력 벡터  $y=(y_0, \dots, y_k, \dots, y_{j-1})$ 의 최근접한 코드벡터에 대한 탐색이 이루어진다. 이러한 이용은 다음의 단계들을 구현한다:
- [0128] C01) 필요한 코드벡터의 현재 인덱스에 대해, 상기 대응 테이블들에 나타나는 인덱스들, 그리고 적절한 경우 상기 디셔너리를 작성할 수 있게 하는 상기 제 2 세트의 요소의 사전 판독을 통해 상기 현재 인덱스에 대응하는 인덱스의 코드벡터의 적어도 일부분을 재구성하는 단계,
- [0129] 적절한 코딩/디코딩 단계들로 계속되는 방법은:
- [0130] C02) 적어도 코딩 중에, 단계 C01)에서 재구성된 코드벡터와 입력 벡터 사이의 거리를 계산하는 단계,
- [0131] C03) 적어도 코딩 중에, 상기 디셔너리의 모든 현재 인덱스들에 대해 상기 단계들 C01) 및 C02)을 반복하는 단계,
- [0132] C04) 적어도 코딩 중에, 단계 C02)의 반복들 중 하나의 과정에서 계산되는, 상기 입력 벡터와의 거리가 최소인, 적어도 부분적으로 재구성된 코드벡터의 인덱스를 식별하는 단계, 그리고
- [0133] C05) 적어도 디코딩 중에, 그의 인덱스가 단계 C04)에서 식별된 코드벡터를 가장한 입력 벡터( $y$ )의 최근접 이웃을 결정하는 단계를 포함한다.
- [0134] 상기에 나타난 바와 같이, 상술한 "제2" 세트는 바람직하게는 상기 제 2 세트의 주어진 차원보다 낮은 차원들의 "제2" 서브세트들로 이루어진다.
- [0135] 특정 실시예에서, 적어도 디코딩에 대해 단계 C01)은:
- [0136] C011) 상기 삽입 규칙들 및 상기 제 2 세트에 대한 링크들을 나타내며:
- [0137] - 상기 제 2 세트의 서브세트의 현재 차원의 인덱스,
- [0138] - 상기 서브세트의 요소의 현재 인덱스, 및
- [0139] - 상기 요소에 기초하여 주어진 차원의 디셔너리의 코드벡터의 구성을 위해 적절한 삽입 규칙의 인덱스를 포함하는 인덱스들을 대응 테이블들에서 판독하는 단계,
- [0140] C012) 그의 현재 인덱스에 의해 식별된 상기 요소를, 그의 현재 차원에 의해 식별된 서브세트에서 판독하는 단계,
- [0141] C013) 단계 C012)에서 판독된 상기 요소에 단계 C011)에서 판독된 인덱스에 의해 식별된 적절한 삽입 규칙을 적용함으로써 상기 주어진 차원에 대한 상기 코드벡터를 완전히 재구성하는 단계를 포함한다.
- [0142] 특정 실시예에서, 코딩에 대해,
- [0143] 단계 C01)은:
- [0144] C011) 상기 삽입 규칙들 및 상기 제 2 세트에 대한 링크들을 나타내며:
- [0145] - 상기 제 2 세트의 서브세트의 현재 차원의 인덱스,
- [0146] - 상기 서브세트의 요소의 현재 인덱스, 및
- [0147] - 상기 요소에 기초하여, 주어진 차원의 디셔너리의 코드벡터의 구성을 위해 적절한 삽입 규칙의 인덱스를 포함하는 인덱스들을 상기 대응 테이블들에서 판독하는 단계,
- [0148] C012) 그의 현재 차원에 의해 식별된 서브세트에서, 그의 현재 인덱스에 의해 식별된 상기 요소를 판독하는 단계를 포함하며,

- [0149] 단계 C02)에서, 상기 거리는:
- [0150] - 상기 삽입 규칙의 함수(function), 및
- [0151] - 상기 요소의 함수(function)로서 추정된 왜곡 기준의 함수(function)로서 계산된다.
- [0152] 따라서, 단순히 디코딩을 위한 완전한 재구성을 예약함으로써, 단계 C01)에서의 상기 주어진 차원을 갖는 상기 코드벡터의 부분적인 재구성만을 제공할 수 있다.
- [0153] 유용한 실시예에서, 치환 코드들의 조합에 따른 보조 구조화 특성이 제공되며, 상기 치환 코드들의 조합의 인덱스가 다음의 단계들의 구현에서 이용된다:
- [0154] CP1) 입력 신호에 기초하여, 그의 절대 벡터  $|y| = (|y_0|, \dots, |y_k|, \dots, |y_{j-1}|)$ 에 의해 그리고  $\epsilon_k = \pm 1$ 을 갖는 부호 벡터  $\epsilon = (\epsilon_0, \dots, \epsilon_k, \dots, \epsilon_{j-1})$ 에 의해 정의되는 입력 벡터  $y = (y_0, \dots, y_k, \dots, y_{j-1})$ 를 형성하는 단계,
- [0155] CP2) 리더 벡터  $|\tilde{y}^i|$ 을 획득하기 위해 치환에 의해 감소하는 값들에 의해 상기 벡터  $|y|$ 의 구성요소들이 랭크되는 단계,
- [0156] CP3) 상기 리더 벡터  $|\tilde{y}^i|$ 의 최근접 이웃  $x^{j'}$ 이 차원  $j$ 의 디셔너리  $D_i^j$ 의 리더 벡터들로부터 결정되는 단계,
- [0157] CP4) 상기 디셔너리  $D_i^j$ 의 상기 최근접 이웃  $x^{j'}$ 의 랭크의 인덱스가 결정되는 단계, 그리고
- [0158] CP5) 단계 CP4)에서 결정된 상기 인덱스, 단계 CP2)에서 결정된 상기 치환 및 단계 CP1)에서 결정된 상기 부호 벡터에 의존하는 입력 벡터에 코딩/디코딩의 유효 값이 적용되는 단계.
- [0159] 본 발명의 또 하나의 유용한 양상에 따르면, 코딩/디코딩을 위해 그리고 가능하게는 상기 디셔너리 또는 디셔너리들의 구성을 위해, 특히 압축 코딩/디코딩 디바이스의 메모리에 상술한 제 2 세트의 요소들 및 대응 테이블들을 저장하기 위한 설비가 제공된다.
- [0160] 이러한 관점에서, 본 발명은 그와 같은 코딩/디코딩 디바이스에 관한 것이다.
- [0161] 본 발명은 특히 컴퓨터 또는 이동 단말기의 프로세싱 유닛의 메모리 또는 착탈식 메모리 매체에 저장되며, 상기 프로세싱 유닛의 판독기와 협력하기 위한 컴퓨터 프로그램 제품에 관한 것으로, 상기 프로그램은 상기의 디셔너리들을 구성하는 방법의 구현을 위한 명령들을 포함한다.
- [0162] 본 발명은 또한, 이러한 타입의 프로그램, 특히 코딩/디코딩 디바이스를 통합하는 컴퓨터 또는 이동 단말기의 프로세싱 유닛의 메모리에, 또는 착탈식 메모리 매체 상에 저장되며 상기 프로세싱 유닛의 판독기와 협력하기 위한 컴퓨터 프로그램 제품에 관한 것으로, 상기 프로그램은 상기의 압축 코딩/디코딩에의 상기 애플리케이션의 구현을 위한 명령들을 포함한다.
- [0163] 본 발명의 다른 특성들 및 장점들은 상기에 설명한 도 1a 및 1b에서 벗어나, 첨부한 도면들과 상세한 설명을 검사하면 명백해질 것이다:

**실시예**

- [0182] 먼저 도 2a 및 도 2b를 참조하면, 본 발명의 의미 내의 디셔너리  $D_i^N$ 의 2가지 메인 특성들이 예시된다.
- [0183] 도 2a에서, 주어진 차원  $N$ 에 대해, 각각의 증가하는 분해능들  $r_1, r_2, \dots, r_i$ 의 디셔너리들  $D_1^N, D_2^N, \dots, D_i^N$ 이 서로의 내부에 내장된다. 따라서, 최대 분해능  $r_i$ 의 디셔너리  $D_i^N$ 은 이후에 나타나는 바와 같이, 더 낮은 분해능  $r_j(j < i)$ 의 디셔너리  $D_j^N$ 을 결정하게 할 수 있다. PR로 표시된 상기 제 1 특성은 이하에 "내장 특성"이라 칭한다.
- [0184] 이제 도 2b를 참조하면, 주어진 차원  $N$  및 분해능  $r_i$ 의 임의의 디셔너리  $D_i^N$ 은 2개의 서로 소인(disjoint) 세트들의 조합이다:

- [0185] o 삽입 규칙들의 유한 집합 {Rm}에 따라 실수의 유한 세트 A로부터 취해진 요소들  $x_j$ (화살표 F2)의 더 낮은 차원 N-1의 디셔너리들  $D_i^{N-1}$ 을 삽입될 요소들  $x_j$ 을 결정하는 삽입 규칙  $R'(j,k)$ (화살표 F1) 및 (예를 들어, 구성 중인 벡터  $Y^N$ 의 위치 k에서의) 요소들을 삽입하는 방법으로 코드벡터들  $Y^{N-1}$ 로 삽입함으로써 구성된 코드벡터들  $Y^N$ (화살표 F3)으로 이루어진 제 1 세트  $D_i^N$ , 그리고
- [0186] o 상기 삽입 규칙들의 집합에 따라 상기 유한 세트의 더 낮은 차원 요소들의 코드벡터들에 삽입함으로써 얻어질 수 없는 벡터들  $Y'$ 으로 이루어지는 제 2 세트  $\bar{D}'_i^N$ .
- [0187] PD로 표시된 제 2 특성은 이하에 "제어 확장에 의한 부분적 합성의 특성"이라 칭해진다.
- [0188] 도 2a 및 도 2b 및 상기 본 발명의 요약에서, 분해능 및/또는 차원에 대한 인덱스들은 예를 들어, 정수 1로부터 시작하여 주어진 정수로(경우에 따라  $i$ ,  $n$  또는  $N$ 일 수 있음) 이루어진다. 프로그래밍 분야, 특히 C++ 언어 분야의 당업자는 이들 인덱스들이 컨텍스트에 따라 0으로부터 시작하여  $i-1$ ,  $n-1$  또는  $N-1$ 에 도달함을 이해할 것이다.
- [0189] 이하에는 특히, 구조화된 이들 디셔너리들을 구성하기 위한 알고리즘들에서, 2개의 구조화 특성들 PR 및 PD를 소유하는 디셔너리를 구성하는 방법이 설명된다. 상기 2개의 구조화 특성들에 의해 유도되는 링크들은 "GLA" 또는 "SKA"와 같은 이하에 공통으로 사용되고 설명된 반복 구성 알고리즘들을 적용시킴으로써 상기 디셔너리들을 구성하기 위한 알고리즘들을 개발하도록 유용하게 이용된다.
- [0190] 일반적인 방식으로, 지시된 바와 같이:
- [0191] - 동일한 차원 및 서로 다른 분해능들의 상호관계 디셔너리들이 상기 내장 특성 PR을 이용하여 연속적으로 구성되고,
- [0192] - 보충 또는 변형으로서, 제어된 확장에 의한 부분적 합성의 특성 PD에 의해 상호관계되는 서로 다른 차원들의 디셔너리들이 구성되며,
- [0193] - 상기 2개의 구조화 특성들 PD 및 PR을 소유하는 여러 차원들 및 분해능들의 디셔너리들이 획득된다.
- [0194] 일반적인 방식으로, 주어진 차원 (PR)에 대한 분해능을 증가시킴으로써 내장 디셔너리들을 구성하기 위해, 3개의 구성 방법들이 제안된다.
- [0195] 제 1 방법은 (최소 분해능으로부터 최대 분해능까지 오름차순) 증가하는 분해능들에 따라 상기 디셔너리들을 구성하는 것으로 이루어진다.
- [0196] 제 2 방법은 (최대 분해능로부터 최소 분해능으로 내림차순) 감소하는 분해능들에 따라 상기 디셔너리들을 구성하는 것으로 이루어진다.
- [0197] 제 3 방법은 상기 분해능들을 상기 최소 분해능으로 내림차순으로 감소시키고, 상기 분해능들을 상기 최대 분해능까지 오름차순으로 증가시킴으로써 중간 분해능의 디셔너리에 기초하여 상기 디셔너리들을 구성하는 것으로 이루어진다. 상기 프로세스는 가변 분해능의 벡터 양자화기의 공칭 분해능이 상기 중간 분해능일 때 특히 유의하다.
- [0198] 차원 j에 대한 상기 디셔너리들의 내장 특성 PR이 마지막으로 다음에 의해 전달된다:
- [0199]  $D_0^j \subset D_1^j \subset \dots \subset D_i^j \subset D_{i+1}^j \dots \subset D_{N_j-1}^j$
- [0200] 기록되기는:
- [0201] -  $N_j$ : 차원 j에 대한 분해능들(또는 가변 속도 코더에서 가능한 속도들의)의 수,
- [0202] - 차원 j에 대한 분해능들의 세트
- [0203]  $R_j = \{r_0^j, r_1^j, \dots, r_i^j, r_{i+1}^j, \dots, r_{N_j-1}^j\}$
- [0204] 여기서  $r_0^j < r_1^j < \dots < r_i^j < r_{i+1}^j < \dots < r_{N_j-1}^j$

- [0205] -  $D_i^j$ : 분해능  $r_i^j$ 의 차원  $j$ 의 디셔너리
- [0206] -  $T_i^j$ : 분해능의 디셔너리의 크기
- [0207]  $r_i^j (T_i^j = 2^{ir_{ij}}, \text{ 즉 } r_i^j = 1/j \log_2 T_i^j)$
- [0208] 도 3은 증가하는 분해능들의 기능으로서 상기 디셔너리들의 내장을 도시한다.
- [0209] 더 낮은 분해능의 디셔너리들의 재업데이팅 없이 증가하는 분해능들에 따른 구성을 위한 알고리즘의 흐름도가 도 5에 주어진다.
- [0210] 도 5를 참조하면, 먼저  $i=0$  및 루프 반복 인덱스  $iter=0$ 으로 고정하는 초기화 단계들(51 및 52)에 후속하여, 최저 분해능의 디셔너리  $D_0^j$ 를 구성한다. 그 후에, 최저 분해능의 디셔너리  $D_0^j$ 가 고정되며, 즉시 더 높은 분해능  $D_1^j$ 의 디셔너리가 이하에 설명된 종래의 구성 알고리즘의 변형에 의해 구성된다. 그 후에, 상기 방법은 최대 분해능  $D_{N_j-1}^j$ 의 디셔너리가 구성될 때까지 반복된다.
- [0211] 따라서, 단계(53)에서, 반복 프로세스에 의해,  $(T_i^j - T_{i-1}^j)$  벡터들을 더 낮은 분해능  $r_{i-1}^j$ 의 디셔너리  $D_{i-1}^j$ 에 첨부함으로써 형성되는, 초기 디셔너리  $D_i^j(0)$ 에 기초하여 디셔너리  $D_i^j$ 를 구성하려 한다.
- [0212] 등급들(54)을 구성하기 위한 알고리즘은 종래의 알고리즘과 동일하지만 상기  $T_i^j$  중심들(55)을 구성하기 위한 알고리즘이 수정된다. 구체적으로, 더 낮은 분해능의 디셔너리들에 속하지 않는  $(T_i^j - T_{i-1}^j)$  중심들은 재계산되고 업데이트되는 한편, 더 낮은 분해능의 디셔너리들의  $(T_{i-1}^j)$  중심들이 재업데이트되지 않는다. 더 낮은 분해능의 모든 디셔너리들의 총 왜곡들이 감소하거나 일정하게 남아있는 경우에, 일 변형이 상기 더 낮은 분해능들의 디셔너리들의 중심들의 재업데이팅을 허용한다. 이 경우에, 더 낮은 분해능들의 디셔너리들이 그에 따라 수정된다.
- [0213] 루프 인덱스  $iter$ 은  $i^{\text{th}}$  분해능 및 상기 차원  $j$ 에 의존하여(테스트 57) 최대 번호  $Niter(i, j)$ 까지 증분된다(단계 56). 원하는 분해능  $N_j$ 에 일단 도달되면(테스트 58), 상기 분해능  $N_j$ 에서의 디셔너리(최종 단계 59) 및 1 내지  $N_j$ 의 범위에 있는  $i$ 에 대해, 분해능  $r_i$ 의 디셔너리들  $D_i^j$ 의 세트를 획득한다.
- [0214] 감소하는 분해능들에 따른 디셔너리들을 구성하기 위해, 먼저 최고 분해능의 디셔너리를 구성한다. 그 후에, 최고 분해능의 디셔너리가 고정되고, 어떤 기준에 따라 정렬되는 여러 서브세트들로 상기 최고 분해능의 디셔너리를 분할한다. 상기 분할을 정렬하기 위해 여러 기준이 이용될 수 있다. 예를 들어, 상기 서브세트들의 기수(cardinal), 러닝 시퀀스의 인보크(invok)(즉, 상기 양자화 영역들의 기수), 총 왜곡 또는 더 정확하게는 상기 왜곡의 감소에 대한 기여에 따라 상기 서브세트들을 정렬할 수 있다. 명백하게, 여러 기준을 조합하고 그들의 각각의 중요성에 가중치를 둘 수 있다. 유사하게, 상기 디셔너리의 분할은 여러 방식으로: 기본 분할로부터(각 서브세트의 일 요소) 더 정교한 분할 방식으로 수행될 수 있다. 상기 정렬된 분할은 그의 정렬된 등급들의 진보적인 조합에 의해 상기 내장 디셔너리들의 구성에 기초한다.
- [0215] 바람직하게는, 상기 분할은 삽입 규칙들의 집합의 서브세트에 기초하여(상기 세트 자체와 동일함) 동일한 코드 벡터의 확장에 기초한 요소들과 함께 그룹핑함으로써 제어 확장에 의해 부분적 합성의 특성 PD에 기초할 수 있다.
- [0216] 주목할 사항으로서, 여러 절차들을 교번함으로써 여러 반복들을 행할 수 있다. 예를 들어, 내장 디셔너리들은 증가하는 분해능 절차에 따라 구성된 후, 상기 감소하는 분해능 절차가 적용된다. 상기 2가지 방법들을 조합함으로써, 분해능에 의해 내장된 디셔너리들은 중간 분해능  $r_i$ 의 디셔너리에 기초하여 구성된다. 상기  $i^{\text{th}}$  디셔너리는 먼저 구성된다. 그 후에, 상기 디셔너리에 기초하여, 더 낮은 분해능의 디셔너리들은 감소하는 분해능들에 의한 제 2 방법에 의해 구성되며, 더 높은 분해능들의 디셔너리들은 증가하는 분해능들의 제 1 방법에 의해 구

성된다.

[0217] 일반적인 방식으로, 제어 확장에 의한 부분적 합성에 의해 다양한 차원들의 디서너리를 구성하기 위한 3가지 방법들이 제안된다(특성 PD).

[0218] 제 1 방법은 상기 차원들을 증가시키는 것으로 이루어진다. 또 하나의 방법이 상기 차원들을 감소시키는 것으로 이루어진다. 마지막으로, 최종 방법은 중간 차원의 디서너리를 구성하기 시작하여, 차원을 계속 증가 및 감소시킴으로써 더 높은 그리고 더 낮은 차원들의 디서너리들을 구성하는 것으로 이루어진다. 제어된 확장에 의한 부분적 합성은 이하에 설명되는 실수들 및 삽입 규칙들의 집합의 유한 세트를 결정하기 위한 절차들의 미세-조정을 발생시킨다. 간단하게 나타나는 바와 같이, 바람직하게는 "확장" 요소들의 비율(proportion)(상기 디서너리의 기수에 관한 제 1 세트의 요소들의 수)는 차원에 따라 증가함으로써 차원에 따라 증가하는 상기 제 2 세트의 저장 비용을 감소시킬 수 있게 한다. 상기 비율은 상기 애플리케이션의 복잡한 제약들(메모리/계산 능력)에 의해 사전에 고정될 수 있거나, 또는 "자유로운" 상태로 있다. 이후의 경우에, 유용하게 상기 구성 알고리즘은 이하에 나타나는 바와 같이 제어된 확장에 의해 획득된 요소들을 포함하는 상기 제 1 세트의 요소들을 제공한다.

[0219] 따라서, 제어된 확장에 의한 부분적 합성의 제 2 특성 PD는 마지막으로 다음에 의해 전달된다:

[0220] 
$$D^j_i = D'^j_i \cup \overline{D''^j_i}$$

[0221] 여기서,

[0222] -  $D'^j_i$ : 삽입 규칙들의 유한 집합  $\{R_m\}$ 에 따라  $\mathbf{R}$ 의 유한 세트 A로부터 취해진 더 낮은 차원들 요소들의 디서너리들의 코드벡터들로 삽입함으로써 획득될 수 있는  $D_i^j$ 의 코드벡터들의 세트,

[0223] -  $\overline{D''^j_i}$ :  $D_i^j$ 의 컴플리먼트(complement), 삽입 규칙들의 집합  $\{R_m\}$ 에 따라 A의 더 낮은 차원 요소들의 코드벡터들로 삽입함으로써 얻어질 수 없는  $D_i^j$ 의 코드벡터들의 세트.

[0224] 이하에는 상기 제 2 특성 PD를 확인하기 위한 삽입 규칙들의 예가 설명된다.

[0225] 먼저, 기본 삽입 규칙들의 집합이 정의된다: 각 기본 규칙은 주어진 벡터 위치에서의 구성요소로서 실수 A의 유한 세트의 일 요소만을 삽입하는 것에 이루어진다. 한 쌍의 2개의 양의 정수들에 의해 각각의 기본 규칙이 제공되며, 하나의 정수는 상기 유한 세트의 요소의 랭크를 제공하며, 다른 하나의 정수는 삽입 위치의 랭크를 제공한다. 상기 기본 규칙들의 집합에 기초하여, 구성요소들을 삽입하기 위한 임의의 더 정교한 규칙을 구성할 수 있다.

[0226] 물론, 순수하게 가역 방식으로 더 낮은 차원 N-n에 도달하도록 주어진 차원 N의 유한 세트의 하나 이상의 요소들을 삭제하는 것으로 이루어진 삭제 규칙들을 정의할 수 있다.

[0227] 삽입 규칙을 정의하기 위해:

[0228] -  $N_a$ 는 A의 기수이고  $a_i$ 는 그의  $i^{th}$  요소:

[0229] 
$$A = \{a_0, a_1, \dots, a_i, \dots, a_{N_a-1}\},$$

[0230] -  $R' (i_m, p_m)$ 은 위치  $p_m$ 에서  $a_{i_m}$ 을 삽입하는 것으로 이루어진 기본 삽입 규칙.

[0231] 따라서, 최대 차원이  $j_{max}$ 이면, 가능한 기본 규칙들의 수는  $N_a * j_{max}$ 이다. 예를 들어,  $N_a=2$  및  $j_{max}=3$ 에 대해, 6개의 가능한 기본 규칙들이 모두 카운트된다:

[0232]  $R'(0,0)$ : 위치 0에서  $a_0$ 를 삽입,

[0233]  $R'(1,0)$ : 위치 0에서  $a_1$ 를 삽입,

[0234]  $R'(0,1)$ : 위치 1에서  $a_0$ 를 삽입,

[0235]  $R'(1,1)$ : 위치 1에서  $a_1$ 를 삽입,

- [0236] R'(0,2): 위치 2에서  $a_0$ 를 삽입,
- [0237] R'(1,2): 위치 2에서  $a_1$ 를 삽입,
- [0238] 규칙들 R'(0,0) 및 R'(0,1)의 합성은 상기 규칙을 제공한다: 위치 0 및 1에서  $a_0$ 을 삽입. 이에 의해, 차원 j의 코드벡터에 기초하여 차원 j+2의 코드벡터를 획득할 수 있게 된다.
- [0239] 규칙들 R'(1,0) 및 R'(0,2)의 합성은 상기 규칙을 제공한다: 위치 0에서의  $a_1$  삽입 및 위치 2에서의  $a_0$  삽입. 이에 의해, 차원 j의 코드벡터에 기초하여 차원 j+2의 코드벡터를 획득할 수 있다.
- [0240] 더 일반적으로, n 기본 규칙들 R'(i<sub>m</sub>,p<sub>m</sub>)(m=0부터 n-1까지)의 합성에 대해 R(n,{(i<sub>m</sub>,p<sub>m</sub>)<sub>m=0, n=1</sub>)로 쓸 수 있으며, 이에 의해 차원 j의 코드벡터에 기초하여 차원 j+n의 코드벡터를 획득할 수 있다. 주목할 사항으로서, 상기 i<sub>m</sub>은 p<sub>m</sub>이 구별되는 n 위치들만큼 다를 필요는 없다. 바람직하게는, 상기 위치들 p<sub>m</sub>은 증가하는 순서로 배치되는데, 즉:
  - [0241]  $p_0 < p_1 < \dots < p_m < \dots < p_{n-1}$ .
- [0242] 도 4는 삽입 규칙들 및 더 낮은 차원들의 디셔너리들의 코드벡터들에 기초하여 디셔너리의 코드벡터들의 구성을 도시한다.
- [0243] 다양한 차원들의 디셔너리들, 2개의 서로 소인 세트들의 조합들을 구성하는 여러 실시예들이 제공되며, 제 1 세트는 삽입 규칙들의 집합에 따라 실수들의 유한 세트로부터 취해진 요소들을 더 낮은 차원들의 디셔너리들의 코드벡터들에 삽입함으로써 구성된 코드벡터들로 이루어지며, 제 2 세트는 상기 삽입 규칙들의 집합에 따라 실수들의 상기 유한 세트의 더 낮은 차원 요소들의 코드벡터들에 삽입함으로써 얻어질 수 없는 벡터들로 이루어진다.
- [0244] 상기 제 1 세트는 실수들의 유한 세트뿐 아니라(즉 그의 기수 및 값들) 삽입 규칙들의 집합의 결정을 필요로 한다.
- [0245] 상기 유한 세트의 구성 및 삽입 규칙들의 집합 형성이 수행된다:
  - [0246] - "사전": 상기 유한 세트 및 삽입 규칙들의 집합이 상기 디셔너리들을 구성하기 전에 결정된다. 이러한 선택은 바람직하게는 양자화될, 예를 들어 러닝 시퀀스에 의해 모델링되는 소스의 통계적 분석에 기초한다. 예를 들어, 상기 유한 세트의 선택은 상기 소스의 단일차원 확률 밀도(또는 그의 히스토그램)에 기초할 수 있다;
  - [0247] - 또는 "사후": 먼저 제어된 확장에 의해 부분적 합성의 규칙을 수반할 필요없이, 모든 차원들에 대한 분해능에 의해 내장된 디셔너리들을 구성한다. 상기 유한 세트 및 삽입 규칙들의 집합의 선택은 이들 "초기" 디셔너리들의 통계 연구에 의해 이루어진다.
- [0248] 상기 2가지 솔루션들 "사전" 또는 "사후"는 연속적으로 사용되고, 및/또는 조합될 수 있다. 예를 들어, 제 1 세트 및 삽입 규칙들의 제 1 집합은 상기 러닝 시퀀스의 분석에 의해 선택될 수 있으며, 상기 디셔너리들의 제 1 구성 후에, 이들 디셔너리들의 분석이 상기 세트 A 및/또는 삽입 규칙들의 집합의 총 또는 부분적 업데이트를 야기할 수 있다.
- [0249] 주목할 사항으로서, 상기 유한 세트 및/또는 삽입 규칙들의 집합은 상기 차원들에 의존할 수도 있으며 또는 의존하지 않을 수도 있다. 그 후에, 각 쌍의 차원들 (j, j')에 대해 특정한 집합 및/또는 세트, 또는 차원의 차이만큼 특정한 집합 및/또는 세트를 결정하거나, 전체 세트를 결정할 수 있다. 다시 한번, 상기 선택은 사전에 또는 러닝 시퀀스 및/또는 상기 디셔너리들의 통계적 분석 후에 이루어진다.
- [0250] 증가하는 차원들에 따라 상기 디셔너리들을 구성하기 위해, 먼저 상술한 바와 같이 벡터 양자화 설계의 종래의 절차에 의해 최저 차원의 디셔너리를 구성한다. 그 후에, 상기 디셔너리가 구성되고, 즉시 더 높은 차원의 디셔너리가 종래의 구성 알고리즘의 변형에 의해 구성된다. 더 낮은 차원의 디셔너리에 기초하여, 모든 가능한 초기 코드벡터들은 삽입 규칙들을 적용함으로써 구성되고, 상기 디셔너리는 "자유로운" 코드벡터들(즉, 확장에 의해 획득될 수 없는 코드벡터들)로 보충될 수 있다. 주목할 사항으로서, 상기 초기 디셔너리의 크기는 원하는 크기보다 더 클 수 있다. 상기 초기 디셔너리에 기초하여, 벡터 양자화기를 구성하기 위한 반복 알고리즘의 변형이 적용된다. 등급들은 상기 러닝 시퀀스의 양자화에 의해 구성되며, 중심들이 상기 제 1 세트의 코드벡터들에 대한 제어된 확장 제약에 순응함으로써 업데이트된다. 상기 제 1 세트의 이들 코드벡터들에 대해, 삽입에 의해 얻

어진 구성요소들을 재계산하지 않거나, 또는 모든 구성요소들을 재계산하고, 상기 삽입 규칙들에 의해 획득된 구성요소들을 되돌리기 위해 획득된 코드벡터들을 수정할 수 있다. 따라서, 상기 디서너리의 크기가 상기 원하는 크기보다 큰 경우에 빈 등급들을 제거한다. 상기 알고리즘의 종료에서, 상기 디서너리의 크기가 상기 원하는 분해능보다 큰 경우에, 첫 번째 코드벡터들만을 유지하도록 상기 디서너리의 요소들을 분류하기 위한 절차가 적용된다. 상기 반복 알고리즘은 재실행된다. 그 후에, 더 높은 차원의 디서너리의 구성으로 넘어가고, 초기의 디서너리는 2개의 최소 차원들의 2개의 디서너리들에 기초하여 제어된 확장에 의해 구성되고 "자유로운" 코드벡터들로 보충되며, 그 후에 벡터 양자화를 구성하기 위한 반복 알고리즘의 변형이 적용된다. 상기 방법은 그 후에, 최대 크기의 디서너리가 구성될 때까지 반복된다.

[0251] 일 변형으로서, 감소하는 차원들에 따라 디서너리들을 구성하기 위해, 먼저 최대 차원의 디서너리를 구성한다. 그 후에, 상기 디서너리가 고정되며, 더 낮은 차원의 가능한 코드벡터들이 추출된다. 유용하게는, 이들 코드벡터들의 구성요소들로서 A의 요소들을 나타내도록 더 높은 차원들의 코드벡터들을 수정함으로써 상기 추출 절차가 촉진된다.

[0252] 보충적인 변형에서, 한편으로 증가하는 차원들에 따라, 그리고 다른 한편으로 감소하는 차원들에 따라 상기 2개의 구성들을 교대함으로써 여러 반복들이 유용하게 수행된다.

[0253] 상기 제어된 확장 절차를 촉진하기 위해, 본 발명은 또한, 상기 코드벡터들의 구성요소들의 변환을 수행할 수 있다. 예시적인 변환은 높은 분해능에서의 스칼라 양자화이다. 이들 차원들이 상기 벡터 양자화에 의해 직접 사용되지 않는 경우에도 더 낮은 차원들의 "디서너리들"을 구성하는 것이 유익하다. 예를 들어, 상기 스칼라 양자화가 사용되지 않는 경우에도 상기 차원 1로 시작할 수 있다. 유사하게, 또한 중간 차원들의 디서너리들을 구성하는 것이 유익할 수 있다. 이들 "디서너리들"은 저장 및 계산들의 복잡도를 감소시키기 위해 제어된 확장 절차에 의해 더 유용하게 사용된다.

[0254] 더 나타난 바와 같이, 분해능(PR)에 의한 내장에 의해 디서너리들을 구성하는 알고리즘들을 제어된 확장(PD)에 의한 부분적 합성에 의한 구성용 알고리즘과 공정하게 조합함으로써, 여러 구성 방법들이 개발될 수 있다. 주목할 사항으로서, 상기 알고리즘들은 반복적이며, 다양한 기술들이 변형되어 이용될 수 있다. 예를 들어, 최소 차원에 대한 최대 분해능의 디서너리를 구성하는 것으로 시작하여, 이로부터 감소하는 분해능들(특성 PR)에 의해 내장된 디서너리들을 추론하며, 그 후에 상기 특성 PD에 의해 즉시 더 높은 차원에 대한 최대 분해능의 디서너리를 구성하며, 이러한 차원에 대해 분해능에 의해 내장된 디서너리들을 구성하며, 최대 차원의 디서너리들(분해능에 의해 내장됨)이 획득될 때까지 반복된다.

[0255] 디서너리들의 세트  $\{D_i^j\}_{i=0, \dots, N_j-1, j=j_{\min}, \dots, j_{\max}}$ 를 구성하기 위해 증가하는 차원들 및 감소하는 분해능들에 따라 디서너리 구성의 기술들을 조합하는 바람직한 구성이 이하에 설명되는 실시예에서 사용된다.

[0256] 이하에는 본 발명의 의미 내의 디서너리들을 이용하여 디지털 신호들(오디오, 비디오, 등)의 압축 코딩/디코딩, 특히 상기 디서너리들의 구조(제어된 확장에 의한 내장 및 부분적 합성)를 이용하는 코딩 및 디코딩 알고리즘들이 설명된다. 일반적으로, 이해되는 바와 같이 상기 코더 및/또는 디코더에서의 메모리/계산들 사이의 절충의 최적화가 상기 애플리케이션의 제약들에 따라 시도된다.

[0257] 예를 들어, 이하에는 16 kHz(광대역)에서 샘플링된 디지털 오디오 신호들을 코딩하도록 사용되는 "TDAC 코더"라 지칭되는 오디오 코더이다. 상기 코더는 다양한 속도들에서 동작할 수 있는 변환 기반 코더이다. 특히, 상기 속도는 상기 통신의 확립 전에 고정될 수 있거나, 또는 통신 중에 프레임 간에 변동할 수 있다.

[0258] 도 6은 상기 TDAC 코더의 블록도를 제공한다. 7 kHz로 제한되고 16 kHz로 샘플링된 오디오 신호  $x(n)$  대역은 320 샘플들의 프레임들(20 ms)로 분할된다. 수정된 이산 코사인 변환(61)이 50%의 중첩(즉, 매 20 ms마다 MDCT 분석의 리프레싱)을 갖는 640 샘플들의 입력 신호의 블록들에 적용된다.  $y(k)$ 를 획득한 스펙트럼은 최종 31 계수들을 제로화함으로써(첫 번째 289 계수들만이 0과 다름) 7225 Hz로 제한된다. 마스킹 곡선은 상기 마스킹된 계수들의 제로화를 수행하는 마스킹 모듈(62)에 의해 결정된다. 상기 스펙트럼은 동일하지 않은 폭들의 32 대역들로 분할된다. 상기 마스킹된 대역들은, 존재할 경우 상기 신호  $x(n)$ 의 변환된 계수들의 함수로서 결정된다. 상기 스펙트럼의 각 대역에 대해, 상기 MDCT 계수들의 에너지가 계산된다(스케일 인자들을 말함). 상기 32 스케일 인자들은 상기 프레임에서 양자화되고, 코딩되며 전송되는 신호의 스펙트럼 포락선을 구성한다(블록 63). 상기 양자화 및 상기 코딩은 호프만(Huffman) 코딩을 이용한다. 가변 속도에서의 스펙트럼 포락선의 양자화 후에 남아있는 가변하는 비트들의 수가 계산된다. 이들 비트들은 상기 스펙트럼의 MDCT 계수들의 벡터 양자화(65)를 위해 분배된다. 상기 역양자화 스펙트럼 포락선은 대역방향 마스킹 임계값들의 세트를 계산하는데 이용되며, 상

기 마스크 곡선은 상기 비트들의 동적 할당(64)을 결정한다. 대역 단위, 그리고 상기 양자화된 스펙트럼 포락선에 기초한 상기 마스크 곡선의 계산은 상기 이진 할당에 관련된 보조 정보의 전송을 회피한다. 구체적으로, 상기 디코더는 상기 코더와 동일한 방식으로 비트들의 동적 할당을 계산한다. 상기 MDCT 계수들은 그들 대역의 역양자화 스케일 인자들에 의해 정규화되고, 그 후에 가변 차원 및 가변 속도의 벡터 양자화기들에 의해 양자화된다. 마지막으로, 상기 이진 트레인은 상기 스펙트럼 포락선 및 프레임에서 코딩 및 전송되는 이들 대역방향 정규화 계수들에 관한 정보의 멀티플렉싱(66)에 의해 구성된다. 나타난 바와 같이, 도 6에서의 기준들(67 및 68)은 음성 또는 무음성 신호  $x(n)$ 의 검출, 및 톤의 검출(톤 주파수들의 결정)로서 각각 알려진 단계들에 대응한다.

[0259] 이하에는 상기 TDAC 코더에 상기 MDCT 계수들의 동일하지 않은 폭들의 대역들에 기반한 가변 속도를 갖는 벡터 양자화기들이 설명된다. 상기 대역방향 정규화 MDCT 계수들의 양자화는 특히 본 발명에 따라 구성된 디서너리들을 이용한다. 동일하지 않은 폭들의 대역들로의 분배는 실제로 다양한 차원들의 벡터들을 발생시킨다. 상기 사용된 대역 분배를 제공하는 도 7a의 어레이는 또한, 계수들의 벡터의 최종적인 차원, 즉 제 3 열에 의해 표시되는 계수들의 수를 나타낸다.

[0260] 상기 스펙트럼 포락선의 호프만 코딩 후에 남아있는 가변적인 비트들의 수는 상기 다양한 대역들에 동적으로 할당된다. 도 7b의 어레이는 1 내지 15의 범위에 있는 차원  $j$ 에 대해, 상기 차원들  $j$ 에 대한 밴드 당 레이트  $j \cdot R_j$ 의 세트들(즉 대역 당 분해능들의 값들) 및 분해능들  $N_j$ 의 수들을 제공한다. 주목할 사항으로서, 제어된 확장에 의한 부분적 합성의 구조화 특성을 유용하게 이용하기 위해, 임의의 대역폭에 대응하지 않고, 그의 요소들이 더 큰 차원의 코드벡터들을 합성하는데 이용되는 차원들 1, 2, 6, 11에서 벡터 양자화기들이 구성된다. 또한 주목할 사항은 큰 차원들에 대해서도 상기 분해능들의 입도의 미세도이다.

[0261] 상기 모듈(62)에서의 마스크된 계수들의 제로화는 상기 정규화 MDCT 계수들의 분석 동안, 상기 기본 삽입 규칙들의 모든 가능한 합성물 중 시작 세트  $A=\{0\}$  및 삽입 규칙들의 집합으로서의 선택을 발생시킨다. 이에 의해, 임의의 위치에서 제로들을 삽입하게 된다.

[0262] 그러나, 더 미세한 분석은 모든 치환들 및 모든 부호들이 허용됨에 따라 타입 II의 정규화된 치환 코드들의 조합으로 형성된 디서너리들을 이용함으로써, 상기 디서너리들에 대한 추가의 구조적 제약을 부과한다. 타입 II의 각 치환 코드에 대해, 사전식 의미 내의 최대 벡터는 절대 리더라 칭해지며, 감소하는 순서로 상기 구성요소들의 절대값들을 정렬함으로써 획득된다. 상기 디서너리들의 구성은 그들의 정규화 절대 리더들을 결정하게 한다. 상기 제어된 확장을 이들 절대 리더들에 적용하는 것은 최종 구성요소들로서 이들에 제로들을 삽입하는 것이다.

[0263] 또한, 왜곡 기준이 고정된다. 바람직하게는, 선택된 상기 왜곡 기준은 여기서 유클리드 거리이다. 상기 디서너리는 정규화되고, 양자화된 입력 벡터와의 유클리드 거리를 최소화하는 코드벡터에 대한 탐색은 상기 입력 벡터와의 스칼라 곱을 최대화하는 코드벡터의 탐색으로 된다. 또한, 상기 디서너리는 치환 코드들의 조합이 되며, 입력 벡터와의 스칼라 곱을 최대화하는 코드벡터에 대한 탐색은 상기 입력 벡터의 절대 리더와의 스칼라 곱을 최대화하는 것에 대한 상기 디서너리의 절대 리더들 중의 탐색이 된다(이는 또한, 감소하는 순서로 랭크하도록 그의 구성요소들의 절대값들의 치환에 의해 얻어진다).

[0264] 이하에는 본 발명의 의미 내의 벡터 양자화들의 설계를 위한 러닝 시퀀스가 정의된다. 상술한 바와 같이, 양자화기의 설계를 위한 러닝 시퀀스를 결정하는 것이 바람직하다. 그들 대역의 스케일 인자에 의해 정규화된 289 MDCT 계수들의 프레임으로 이루어진 긴 시퀀스가 먼저 광대역 오디오 신호들의 다양한 샘플들에 기초하여 획득된다. 그 후에, 계수들의 각 정규화 벡터에 대해, 그의 절대 리더가 추론된다. 다양한 차원들의 절대 리더들의 세트에 기초하여, 다중차원 러닝 시퀀스들  $S^0$  및  $S^1$ 의 2개 카테고리들이 생성된다:

[0265] -  $S^0 = \{S_j^0\}_{j \in [1, 15]}$ ,  $S_j^0$ 는  $j$ 개의 비제로 계수들을 갖는 절대 리더들의 첫 번째  $j$  구성요소들에 의해 형성된 모든 벡터들의 세트이다. 따라서,  $S_j^0$ 는 제로 계수를 갖지 않는 차원  $j$ 의 절대 리더들로 이루어지고, 차원  $j+1$ 의 절대 리더들은 단일의 제로 계수를 갖고, 차원  $j+2$ 의 절대 리더들은 2개의 제로 계수들을 갖고, ... 차원 15의 절대 리더들은  $15-j$  제로 계수들을 갖고,

[0266] - 그리고  $S^1 = \{S_j^1\}_{j \in [3, 4, 5, 7, 8, 9, 10, 12, 13, 14, 15]}$ ,  $S_j^1$ 는  $j$  계수들을 갖는 대역들의 모든 절대 리더들의 세트이다.

[0267] 예를 들어, 계수들 (0.;0.6;0.;0.;0.8)의 정규화 벡터에 기초하여, 상기 시퀀스  $S_5^1$ 에 속하는 절대 리더 (0.8;0.6;0.;0.;0.) 및 절대 리더의 첫 번째 2개의 비제로 구성요소들에 의해 형성되는  $S_2^0$ 의 요소 (0.8;0.6)를 추론한다.

[0268] 시퀀스들의 제 1 카테고리는 바람직하게는  $\overline{D'_N}$ 의 리더들의 초기 디셔너리들을 결정하도록 사용된다. 제 2 카테고리는 바람직하게는 2개의 구조화 특성들을 소유하는 다중차원 및 다중분해능 디셔너리들을 구성하도록 사용된다.

[0269] 시퀀스들의 제 1 카테고리  $S^0$ 에 기초하여, 소위 "k-평균 알고리즘(k-means algorithm)"과 같은 종래의 알고리즘을 상기 시퀀스  $S_j^0$ 에 적용함으로써 각 차원 j(j는 1 내지 15 사이의 범위에 있음)에 대한 정규화 절대 리더들의 제 1 디셔너리를 획득한다. 양의 실수 구성요소들을 갖는 이들 리더들은 제 1 구성요소에 관하여(즉 가장 큰 구성요소) 소정의 임계값 아래의 구성요소들을 억제함으로써 수정된다. 상기 소위 "중심-클리핑(clipping)" 절차는 유용하게는 제로들을 발생시키며 더 낮은 차원의 제로 구성요소들이 없는 절대 리더들을 추출할 수 있게 한다. 제어된 확장을 더 제공하기 위해, 이들 추출된 리더들의 구성요소들의 변환이 적용된다. 이러한 목적을 위해, 정수 재구성 레벨들을 갖는 간격 1의 균일한 스칼라 양자화를 수반하는 최소 비제로 구성요소에 의해 각 리더의 정규화를 이용한다(이는 각 리더의 구성요소들을 최근접 정수로 라운드한다). 상기 변환은 메모리에서 상당한 크기의 감소를 발생시키는데, 이는 정확한 정규화 인자를 상기 거리 계산에 도입함으로써 상기 절대 리더들이 정수들의 형태로 저장될 수 있기 때문이다. 주목할 사항으로서, 획득되는, 또는 여러 시퀀스들  $S_j^0$ 에 기초한 서로 다른 실제 리더들은 하나 및 동일한 정수 리더로 변환될 수 있다. 그 후에, 가능한 리더던시들을 제거하고 비제로 정수 구성요소들을 갖는 모든 절대 리더들의 세트  $L^0 = \bigcup_{j \in \{1, \dots, 15\}} L_j^0$ 를 형성하기 위한 절차가 제공되며, 여기서  $L_j^0$ 는 차원 j의 리더들로 이루어지는 서브세트이다.  $L^0$ 를 구성하기 위한 기술은 감소하는 차원들에 따라 제어 확장에 의한 부분적 합성에 의해 디셔너리들을 구성하는 기술에 착안한 것이다. 또한, 주목할 사항으로서  $L^0$ 의 모든 리더들이 최종 구성요소로서 적어도 하나의 "1"을 갖기 때문에, 상기 요소 "1"을 첨부하도록 사전에 이루어진 세트 A의 선택은 사후에 검토될 수 있다.

[0270] 상기 세트  $L^0$ 는 제어된 확장 PD에 의한 부분 합성 및 내장 PR의 2개의 구조화 특성들을 갖는 분해능 및 다중차원들을 갖는 벡터 양자화기들의 설계를 위한 리더들의 초기 디셔너리들의 합성에 기초로서 이용된다. 상기 시퀀스  $S^1$ 에 기초하여, 이들 양자화기들을 구성하기 위한 알고리즘은 증가하는 차원 및 감소하는 분해능을 통해 진행된다.

[0271] 차원 j에 대해, 리더들  $L^1_j$ 의 초기 디셔너리는  $L^0_j$ 의 모든 리더들 및 (j-j') 제로들을 상기 세트들의 리더들  $L^0_j$ 에 삽입함으로써 더 낮은 차원들 j'(j'<j)의 리더들의 제어된 확장에 의해 얻어진 모든 리더들에 의해 형성된다. 예를 들어, 차원 3에서, 리더들의 디셔너리가  $L^0_3$ 의 리더들로 보충되는  $L^1_1 = \{(1)\}$ ,  $L^1_2 = \{(11), (21), (31), (41), (51), (91)\}$ 에 기초한 제어된 확장에 의해 구성된다.

[0272] 각 차원 j에 대해,  $L^1_j$ 에 의해 특성화되는 상기 치환 코드들의 조합은 가능하게는 원하는 최대 분해능보다 더 큰, 높은 분해능의 디셔너리를 구성한다. 따라서, 이들 치환 코드들은 상기 디셔너리의 자연 분할을 수행하며, 상기 분할의 각 등급은 그의 리더에 의해 표현되는 치환 코드이다. 상기 분할의 등급들에 대응하는 최근접 이웃의 영역들의 구성이 상기 시퀀스  $S^1$ 의 양자화에 의해 수행된다. 상기 분할은 상기 치환 코드들의 기수의 증가에 따라 정렬된다. 상기 치환 코드들의 기수들이 동일한 경우에, 제어된 확장에 의해 획득된 리더들의 코드들은 상술한 바와 같이  $L^0_j$ 의 리더들의 코드들에 대해 제공된다. 동일한 세트( $D^1_{N_j}$ , 또는  $\overline{D'_N}$ )에 속하는 2개 등급들의 기수들과 동일한 경우에, 상기 등급들은 양자화 영역의 기수와 상기 총 왜곡의 감소에 대한 기여를 조합하는 기준에 따라 정렬된다. 따라서 정렬된 치환 코드들의 기수들의 집합은 벡터당 대응하는 비율로서 각 치환 코드에

대해 계산된다.  $L_j^1$ 의 정렬된 리더들의 세트를  $L_j^1$ 라 칭한다. 상기 코딩된 인덱서들을 결합한 이진 트레인으로서 배열하는 절차를 피하기 위해, 정수 분해능들만을 사용하는 것을 선택한다.

[0273] 도 7c의 어레이를 참조하면, 상기 기수들의 집합의 속도가 바로 더 높은 정수와 가장 근접한 각각의 다른 분해능의 최종 치환 코드로서 선택함으로써 분해능에 의해 내장된 다중분해능 디셔너리들이 구성된다.  $L_j^1$ 에 의해 특성화된 디셔너리의 분해능이 원하는 최대 분해능보다 더 높은 경우에, 마지막으로 사용되지 않은 치환 코드들이 제거된다.  $D_{Nj-1}^j$ 의 리더들의 최종 정렬 세트를  $L_j(\subseteq L_j^1)$ 라 칭한다. 차원들에 대한 반복들의 종료시에,  $L^0$ 의 어떤 리더들이  $\{L_j\}_{j \in \{3,4,5,7,8,9,10,12,13,14,15\}}$ 의 리더들을 구성하도록 사용되지 않으면, 상기 세트  $L^0$ 는 그들을 제거함으로써 업데이트된다. 이러한 세트를  $L^0 = \bigcup_{j \in \{1, \dots, 15\}} L_j^0$ 로서 기록한다.

[0274] 도 7c 내지 도 7e의 어레이들은 상기 내장 특성 및 제어된 확장에 의한 부분적 합성 특성에 의해 공급되는 메모리 절감을 도모한다. 도 7c의 어레이는 벡터 양자화기들을 여러 차원들에 대한 다중 분해능들과 비교한다: 상기 제 1 양자화기들은 치환 코드들의 조합들로서 간단하게 구조화되고, 제 2 양자화기들은 분해능에 의한 내장 특성을 더 소유한다.

[0275] 도 7c에서:

[0276] -j: 상기 차원,

[0277] - $N_j$ : 차원 j에서의 분해능들의 수,

[0278] - $L_{D_i}^j$ : 상기 디셔너리  $D_i^j$ 의 리더들의 수,

[0279] - $L_D^j$ : 상기 디셔너리  $D_{Nj-1}^j$ 의 리더들의 수,

[0280] -  $\sum_{i=1}^{N_j+1} L_{D_i}^j$ : 상기 내장 특성이 없는 차원 j의 모든 디셔너리들의 리더들을 저장하는데 필요한 메모리(워드들의 수),

[0281] - $jL_D^j$ : 상기 내장 특성을 갖는 차원 j의 모든 디셔너리들의 리더들을 저장하는데 필요한 메모리.

[0282] 도 7d의 어레이는 다중 차원들에 대해 사용되는 이들 후자의 양자화기들과 제어된 확장에 의한 부분적 합성의 구조화 특성을 소유하는 양자화기들을 비교한다.

[0283] 도 7d에서:

[0284] -j: 차원

[0285] - $L_{D_j}$ : 상기 디셔너리  $D_{Nj-1}^j$ 의 리더들의 수,

[0286] -  $\sum_{k=1}^j L_{D_k}$ : 차원들 1 내지 j의 최대 분해능의 디셔너리들의 리더들의 수의 합계,

[0287] -  $\sum_{k=1}^j kL_{D_k}$ : 제어된 확장에 의한 부분적 합성의 특성이 없는 이들 리더들을 저장하는데 필요한 메모리,

[0288] - $L_j$ : 상기 세트  $L_j^0$ 의 리더들의 수,

[0289] -  $\sum_{k=1}^j L_k$ : 상기 차원들 1 내지 j에 대한 합계,

[0290] -  $\sum_{k=1}^j kL_k$ : 제어된 확장에 의한 부분적 합성의 특성을 갖는 차원들 1 내지 j의 모든 디셔너리들의 리더들을 저장하는데 필요한 메모리.

[0291] 도 7e의 어레이는 벡터 양자화기들과 다중 분해능들 및 차원들을 비교한다: 상기 제 1 양자화기들은 치환 코드들의 조합으로서 간단하게 구조화되며, 상기 제 2 양자화기들은 분해능에 의한 내장 및 제어된 확장에 의한 부분적 합성의 구조화 특성들을 더 소유한다.

[0292] 도 7e에서:

[0293] -j: 차원

[0294] -N<sub>j</sub>: 차원 j의 분해능들의 수,

[0295] -  $\sum_{i=1}^{N_j} L_{D_i^j}$  : 내장 특성 또는 제어된 부분적 확장의 특성이 없는 N<sub>j</sub> 분해능들에 대해 저장될 차원 j의 리더들의 수,

[0296] -  $j \sum_{i=1}^{N_j} L_{D_i^j}$  : 이들 2가지 특성들이 없는 차원 j의 모든 디서너리들의 이들 리더들을 저장하는데 필요한 메모리(워드들의 수),

[0297] -  $\sum_{k=1}^j k \sum_{i=1}^{N_k} L_{D_i^k}$  : 이들 2가지 특성들이 없는 차원들 1 내지 j의 모든 디서너리들의 리더들을 저장하는데 필요한 메모리(워드들의 수),

[0298] -  $\sum_{k=1}^j L_k$  : 차원들 1 내지 j에 대한 합계,

[0299] -  $\sum_{k=1}^j k L_k$  : 제어된 확장에 의해 부분적 합성 특성 및 내장 특성을 갖는 차원들 1 내지 j의 모든 디서너리들의 리더들을 저장하는데 필요한 메모리.

[0300] 3가지 어레이들에서, 최종 열(column)은 메모리 감소 인자의 중요성을 도시한다. 상기 내장 특성 자체는 차원 3에서는 3배 이상, 차원 7에서는 5배 이상, 차원 15에서는 7배 이상으로 메모리를 감소시킬 수 있게 한다. 상기 내장 특성에 의해, 차원 j의 분해능들의 세트에 대해 D<sub>i</sub><sup>j</sup>의 모든 리더들을 저장하는 대신에, D<sub>N<sub>j</sub>-1</sub><sup>j</sup>의 리더들만을 (L<sub>j</sub>의 리더들) 저장한다. 제어된 확장에 의한 부분적 합성의 첨부는 도 7d의 어레이의 최종 열(column)에 의해 도시된 바와 같이 상기 메모리를 더 감소시킬 수 있게 한다. 상기 특성에 의해 제공되는 추가의 절감은:

[0301] - 차원 4에서는 1.5배 이상 절감되고,

[0302] - 차원 8에서는 3배 이상 절감되며,

[0303] - 그리고 차원 15에서는 7배 이상으로 절감된다.

[0304] 도 7e에 도시된 바와 같이, 치환 코드들의 조합으로서 간단하게 구조화된 양자화기들에 관하여, 분해능에 의한 내장 특성 및 제어된 확장에 의한 부분적 합성의 특성의 2가지 구조화 특성들을 더 소유하는 양자화기들의 이용은 차원 3에서는 4배, 차원 7에서는 13배, 그리고 11보다 큰 차원들에 대해서는 35배 이상으로 상기 메모리를 감소시킬 수 있다.

[0305] 제어된 확장에 의한 부분적 합성의 특성을 이용하여, L<sup>0</sup>의 리더들만이 저장되어야 하며, 상기 {L<sub>j</sub>}의 리더들은 L<sub>j</sub>의 리더들의 인덱스들로부터 L<sup>0</sup>의 리더들의 인덱스들로의 대응 테이블로부터 검색된다.

[0306] 이제, 벡터 양자화기들을 효율적으로 구현하는 방법을 설명한다.

[0307] 차원 j 및 분해능 r<sub>i</sub>의 벡터 양자화기를 구현하기 위해, 다음의 3가지 문제점들을 해결할 필요가 있다:

[0308] - D<sub>i</sub><sup>j</sup>의 입력 벡터의 최근접 이웃의 탐색,

[0309] - D<sub>i</sub><sup>j</sup>의 코드벡터의 인덱스의 탐색,

- [0310] - 그리고 교호적으로, 그의 인덱스에 기초한  $D_i^j$ 의 코드벡터의 탐색.
- [0311] 인덱세이션(indexation)이 고려되는 한, 디셔너리의 코드벡터들, 타입 II의 치환 코드들의 조합을 인덱싱하는 여러 알려진 방식들이 존재한다. 상기 실시예에서 사용되는 넘버링은 고셋(Gosset) 격자의 구면 코드들을 인덱스하는데 사용된 것에 착안한다.
- [0312] 임의의 차원  $j(j \in \{3,4,5,7,8,9,10,12,13,14,15\})$ 에 대해,  $D_{N_j-1}^j$ 의 각 코드벡터는 그의 치환 코드, 부호들의 조합을 제공하는 이진 인덱스 및 그의 치환 코드의 랭크의 오프셋 특성에 의해 인덱스된다. 치환 코드의 오프셋은  $D_{N_j-1}^j$ 에서 그에 선행하는 치환 코드들의 기수들의 집합이다. 치환들을 넘버링하기 위한 공식들 중에, 소위 Schalkwijk 공식을 선택한다.
- [0313]  $D_{N_j-1}^j$ 의 코드벡터들의 이러한 종래의 넘버링에 더하여,  $L_j$ 의 리더들의 인덱스들로부터  $L^0$ 의 리더들의 인덱스들까지 대응 테이블을 이용한다.  $L^0$ 의 리더들이 저장되며, 따라서  $L^0$ 의 인덱세이션이 자유롭게 이용가능하다. 예를 들어, 증가하는 차원에 의해 비제로 정수 구성요소들을 갖는 이들 리더들을 분류할 수 있다.  $L_j$ 의 리더  $x^j$ 의 각 인덱스  $m^j$ 는  $L^0$ 의 리더  $x^{j'}$ 의 인덱스  $l_m$ 과 관련된다. 상기 인덱스  $l_m$ 에 기초하여, 상기 리더  $x^{j'}$ 의 차원  $j'$  및 리더 자체를 검색한다. 그 후에, 상기 리더  $x^j$ 는  $x^{j'}$ 의 최종 구성요소들로서  $(j-j')$  제로들을 삽입함으로써 검색된다.
- [0314] 도 7f의 어레이는  $L^0$ 의 첫 번째 23개 리더들을 제공한다. 도 7g의 어레이는 각 리더  $x^3$ 에 대해 이를 획득하기 위해 확장된 차원  $j'(j' \leq j)$ 의  $L_{j'}^0$ 의 리더  $x^{j'}$ 을 나타냄으로써 차원 3의 디셔너리들의 치환 코드들의 리더들을 제공한다. 덧붙여서,  $j=j'$ 이면,  $x^{j'}=x^3$ 이다.
- [0315] 도 7f에서:
- [0316] - 1:  $L^0$ 의 리더의 인덱스(516으로부터),
- [0317] - j: 그의 차원,
- [0318] -  $l^j$ :  $L_j^0$ 의 리더들의 인덱스.
- [0319] 도 7g에서:
- [0320] - $m^3$ :  $D_{N_3}^3$ 의 23개 리더들 중 리더  $x^3$ 의 인덱스,
- [0321] -i: 상기 리더가 속하는 최소 분해능의 디셔너리의 인덱스(즉,  $x^3 \notin D_{i-1}^3$  및  $x^3 \in D_i^3$ ),
- [0322] - $jr_i$ : 상기 디셔너리  $D_i^3$ 의 벡터당 속도,
- [0323] - $j'$ :  $L^0$ 의 리더  $x^{j'}$ 의 차원(비제로 구성요소들의 수),
- [0324] - $l_m$ :  $L^0$ 의 516 리더들로부터의  $x^{j'}$ 의 인덱스.
- [0325] 이하에는 일반적인 경우에 적절한 코딩 및 디코딩 알고리즘들이 설명되며, 이후에 추가의 구조적 제약(치환 코드들의 조합)이 추가된 특히 유용한 경우가 설명된다.
- [0326] 먼저 나타난 바와 같이, 바람직하게는 최근접 이웃 탐색 알고리즘의 복잡도를 감소시킬 수 있는 제어된 확장 특성에 의해 특히 유도되는 디셔너리 구조를 이용한다. 특히, 동일한 삽입 규칙을 갖는 코드벡터들은 함께 그룹핑될 수 있다. 예를 들어, 이하에 더 상세하게 다루어질 유클리드 거리 왜곡 기준의 경우에, 디셔너리  $D_i^j$ 의 차원  $j$ 의  $L$  코드벡터들  $\{x_1^j, l=0,1,\dots,L-1\}$ 이 디셔너리  $D_i^{j-n}$ 의 차원  $j-n$ 의  $L$  코드벡터들  $x_1^{j-n}$ 에 기초한 동일한 삽입

규칙  $R(n, \{(i_m, p_m)\}_{m=0, n-1})$ 에 의해 획득되며, 입력 벡터로부터 코드벡터들  $x_i^j$ 의 L 거리들의 계산  $y: \text{Dist}(y, x_i^j) = \sum_{k=0}^{j-1} (y_k - x_{k,j}^j)^2$  은 먼저 항  $\sum_{m=0}^{n-1} (y_{pm} - a_m)^2$  를 계산한 후에, y를 n 구성요소들  $y_{pm}: \text{Dist}(y', x_i^{j-n}) =$

$\sum_{k=0}^{j-n-1} (y'_k - x_{k,j-n}^j)^2$  로 상승시켜 얻어진 차원 (j-n)의 벡터 y'으로부터 상기 코드벡터들  $x_i^{j-n}$ 의 L 거리들을 계산함으로써 가속화될 수 있다.

[0327] 상기에 나타난 바와 같이, 각 차원에 대해 최대 분해능의 디셔너리의 일 부분만이 저장되며, 다른 코드벡터들은 더 낮은 차원의 최대 분해능 및 삽입 규칙들의 디셔너리들로부터 취해진 요소들에 기초하여 추론된다.

[0328] 이하에는 본 발명에 따른 상기 디셔너리 생성 방법을 이용한 압축 코딩/디코딩의 상세한 예시적인 실시예가 주어진다.

[0329] 먼저 나타나는 바와 같이, 고려되는 모든 차원들 j에 대해 모든 디셔너리들  $\{D_i^j\}_{i=1, \dots, N_j}$ 의 세트를 저장하는 대신에, 대응하는 테이블들뿐 아니라  $\{\overline{D_{N_j}^j}\}$  만을 저장한다. 이들 테이블들은 그의 인덱스에 기초하여  $D_{N_j}^j$ 의 코드벡터를 재구성할 수 있게 한다. 이하에 설명되는 바와 같이, 이들 테이블들을 작성하고 저장하는 여러 방법이 존재한다. 예를 들어, 고려되는 모든 차원들 j에 대해, ( $D_{N_j}^j$ 의 코드벡터  $x^j$ 의) 각 인덱스  $m_j$ 에 대해 3개의 스칼라 정수 값들:  $j'$ ,  $m'$  및  $l_r$ 을 도표화할 수 있으며, 여기서  $l_r$ 은  $\overline{D_{N_j}^j}$ 의 세트의 인덱스  $m'$ 의 요소에 적용된 제어 확장에 의한 부분적 합성에 의해  $x^j$ 를 재구성할 수 있게 하는 삽입 규칙의 수이다. 대응하는 테이블들은 이제  $3 \sum_{j=1}^N T_{N_j}^j$  워드들의 저장만을 필요로 한다( $T_i^j$ 가 디셔너리  $D_i^j$ 의 크기인 것을 기억하라). 다중 분해능 및 차원을 갖는 벡터 양자화기의 디셔너리들의 적절한 저장이 고려되는 한, 분해능의 내장 및 확장의 부분 합성의 2개의 구조화 특성들을 갖지 않는 벡터 양자화기의 경우에  $\sum_{j=1}^N \sum_{m=1}^{N_j} T_{N_j}^j$  워드들을 필요로 하는 한편, 이들 2개의 구조화 특성들을 갖는 벡터 양자화기의 디셔너리들의 저장은 단지 하나의  $\sum_{j=1}^N j \overline{T_{N_j}^j}$  워드들을 필요로 하며, 상기 세트  $\overline{D_{N_j}^j} (T_{N_j}^j \leq T_{N_j}^j)$ 의 크기에 대해  $\overline{T_{N_j}^j}$ 라 기입한다. 그러나, 일반적인 방식에서,  $\overline{T_{N_j}^j}$ 는  $T_{N_j}^j$ 보다 훨씬 작는데, 그 이유는 상기 세트  $\overline{D_{N_j}^j}$ 에 대해 세트  $D_{N_j}^j$ 를 제공하려 하기 때문이다. 저장용량을 절감하는 약간의 수치적 예들이 이하에 설명되는 실시예에 제공된다.

[0330] 입력 벡터  $y=(y_0, \dots, y_k, \dots, y_{j-1})$ 의  $D_i^j$ 의 최근접 이웃  $x^j$ 를 탐색하는 것으로 이루어지는 코딩 알고리즘은 바람직하게는 다음의 단계들을 포함한다:

[0331] 단계 C00) 매 인덱스  $m^j \in [0, T_i^j[$ 에 대해  $d_{\min} = \text{VALMAX}$ ;  $m_{\min} = -1$ ;  $m^j=0$ 인 초기화 단계,

[0332] 다음의 단계 C01) 인덱스  $m^j$ 의 코드벡터  $x^j$ 의 재구성으로 이루어지며, 바람직하게는 다음과 같이 수행된다:

[0333] a)  $D_{N_j}^j$ 와 관련된 대응 테이블들에서 3개의 인덱스들  $j'$ ,  $m'$  및  $l_r$ 을 판독,

[0334] b) 차원  $j'$  및 인덱스  $m'$ 의 벡터  $x^{j'}$ 의 세트  $\overline{D_{N_{j'}}^{j'}}$ 의 판독,

[0335] c) 인덱스  $l_r$ 의 삽입 규칙에 따른 제어된 확장에 의한 부분적 합성 특성의  $x^{j'}$ 에의 적용에 의한 상기 코드벡터  $x^j$ 의 재구성.

[0336] 단계 C02) 선택된 왜곡 기준에 따른 y와  $x^j$  사이의 거리  $d(y, x^j)$ 의 계산.

[0337] 그 다음의 단계들 C03) 및 C04)는 상기 입력 벡터로부터의 거리가 최소인 벡터의 인덱스를 식별하기 위해 동작

들 C01) 및 C02)를 반복한다.

[0338] 따라서:

[0339] \*  $d(y, x^j) < d_{min}$ 이면,  $d_{min} = d(y, x^j)$  및  $m_{min} = m^j$

[0340] \* 그 후에,  $m^j$ 를 증분:  $m^j = m^j + 1$

[0341] \* 종료 테스트가 제공된다:

[0342]  $m^j < T_i^j$ 이면 단계 C01)으로 진행,

[0343] 그렇지 않으면: 중단.

[0344] 종료 단계 C05)에서, 그의 인덱스  $m_{min}$ 이 입력 벡터  $y$ 를 갖는 최소 거리  $d_{min}$ 과 일치하는 것으로 식별된 코드벡터의 입력벡터  $y$ 의 최근접 이웃인 코드벡터를 결정한다.

[0345] 따라서, 상기 알고리즘은 단계 C05)로 계속된다:

[0346] \* 종료

[0347]  $D_i^j$ 의  $y$ 의 최근접 이웃  $x_j$ 가 인덱스  $m_{min}$ 의 코드벡터임.

[0348] 그의 인덱스에 기초하여  $D_i^j$ 의 코드벡터를 탐색하는 디코딩 알고리즘은 코딩 알고리즘의 단계 C01)에 의해 제공된다. 특히 나타나는 바와 같이, 상기 디코딩은 디코딩되는 인덱스에 관계없이, 단계 C01)의 코드벡터  $x^j$ (동작 c)의 완전한 재구성과 관련된다.

[0349] 다른 한편, 코딩에 관하여, 상기 재구성은 부분적일 수 있다. 구체적으로, 단계 C02)의 거리 계산의 왜곡 기준이 2개 항목으로 분해될 수 있는 경우 상기 재구성은 생략될 수 있다:

[0350] - 상기 삽입 규칙의 인덱스에만 전적으로 의존함,

[0351] - 또 하나의 코드벡터  $x^{j'}$

[0352] 예를 들어, 유클리드 거리 왜곡 기준의 경우에, 초기화 단계 C00)에서,  $D_i^j$ 에 사용되는 인덱스  $l_r$ 의 각 삽입 규칙에 대해, (인덱스  $l_r$ 의 삽입 규칙이 위치들  $p_m$ 에서의  $j-j'$  구성요소들  $a_{im}$ 을 삽입하는 것인 경우, 여기서  $m$ 은

0 내지  $j-j'-1$ 의 범위에 있음) 상기 거리  $d_{lr} = \sum_{m=0}^{j-j'-1} (y_{p_m} - a_{im})^2$  을 미리 계산할 수 있다.  $y$ 와 단계 C02)의 벡터

$x^j(j', m', l_r)$  사이의 거리 계산은 상기 거리:  $d(y', x^{j'}) = \sum_{k=0}^{j'-1} (y'_k - x'_k)^2$  을 계산하게 되며, 여기서:

[0353] -  $x^{j'}$ 은 단계 C01)의 단계 b)에서 획득된 벡터이고,

[0354] -  $y'$ 은  $y$ 를  $j-j'$  구성요소들  $y_{p_m}$ 으로 올림으로써 획득된 차원  $j'$ 의 벡터이고,

[0355] - 상기 거리  $d(y, x^j)$ 는 간단한 합산  $d(y, x^j) = d_{lr} + d(y', x^{j'})$ 에 의해 획득된다.

[0356] 이 때문에, 상기 코딩 프로세스 동안, (완전하게 재구성된 코드벡터  $x^j$ 의 차원이 되는) 차원  $j$ 보다 낮은 차원  $j'$ 의 코드벡터  $x^{j'}$ 의 재구성을 "부분적"이라 정의한다.

[0357] 또한, 벡터  $x^{j'}$ 이 (여러 삽입 규칙들을 갖는)  $D_i^j$ 의 코드벡터들의 구성시에 여러번 간섭하는 경우, 초기화 단계에서 항  $d(y', x^{j'})$ 을 미리 계산할 수 있다. 따라서, 상기 코딩의 저장(일시적)/복잡도 사이의 절충이 상기 애플리케이션의 요건에 따라 맞춰질 수 있다.

- [0358] 유사하게, 인덱세이션의 저장/복잡도 사이의 절충 또한 상기 애플리케이션의 요건에 맞춰질 수 있다.
- [0359] 코딩을 위해, 상기에 언급된 치환 코드들의 조합의 추가 제약의 경우에, 차원 8의 규칙적인 고셋 격자의 구면 코드들에 대한 최근접 이웃 탐색 알고리즘은 타입 II의 치환 코드들의 조합에 의해, 이들 디서너리들에 대한 간략화에 의해 용이하게 개괄한다.
- [0360] 그와 같은 탐색 알고리즘은 특히 다음에 설명되어 있다:
- [0361] - "Algorithme de Quantification Vetorielle Algebrigue Spherique par le Reseau de Gosset  $E_8$ ", C. Lamblin, J.P. Adoul, Annales Des Telecommunications, no 3-4, 1988 ["Spherical algebraic vector quantization algorithm by the  $E_8$  Gosset lattice"].
- [0362] 제 1 간략화는 홀수 구성요소들을 갖는 고셋 격자의 치환 코드들에 의해 소유되지 않는 타입 II의 치환 코드들의 부호들의 "자유도"에 의해 제공된다. 제 2 간략화는 상기 스칼라 곱의 계산에 대한 각 리더의 비제로 구성요소들의 수를 고려하여 제공된다. 이는 상기 코딩 알고리즘에 의한 제어된 확장의 부분적 합성의 특성에 의해 유도된 구조의 이용을 예시한다. 최종 수정은  $L^0$ 의 리더들의 정수 형태로 상기 저장을 고려함으로써, 엄격하게 양의 정수 구성요소들을 갖는 이들 리더들의 유클리드 기준의 역의 값과 동일한 정정 인자의 스칼라 곱의 계산을 도입하게 된다.
- [0363] 이하에는 본 발명의 2개의 구조화 특성들에 대하여, 상기 디서너리  $D_i^j$ 의 차원  $j$ 의 입력벡터  $y$ 의 최근접 이웃의 탐색이 치환 코드들의 조합으로서 상기 구조를 이용하는 실시예가 설명된다.
- [0364] 3가지 추가 단계들이 전체적으로 제공된다:
- [0365] - (상기의 재구성 단계 C01) 이전에) 상기 절대 리더  $|\tilde{y}|$  및 코딩될 벡터의 부호 벡터  $\varepsilon$ 을 결정하기 위한 2개의 예비 단계들(단계들 CP1) 및 CP2)),
- [0366] - 및 상기 디서너리의 최근접 이웃의 랭크를 계산하기 위한 최종 단계(단계 CP5)).
- [0367] 상술한 탐색은 더 이상  $D_i^j$ 의  $T_j^i$  코드벡터들 중에서가 아니라(즉, 더 이상  $m^j \in [0, T_j^i[$ 에 대해서가 아니라), ( $m^j \in [0, L_{D^j}$  [,  $D_i^j$ 의 리더들 또는 치환 코드들의 수에 대해  $L_{D^j}$  라 기입함)  $D_i^j$ 의  $L_{D_j^i}$ 의 세트  $L_j(i)$ 에 걸쳐서만 행해진다.
- [0368] 본 실시예에서,  $D_i^j$ 의  $y$ 의 최근접 이웃에 대한 탐색은 ( $L_j$ 의 상기  $L_{D^j}$  첫 번째 리더들 중으로부터) 상기 세트  $L_j(i)$ 에서의  $|\tilde{y}|$ 의 최근접 이웃에 대한 탐색이 먼저 이루어진다. 상술한 바와 같이, 이들 리더들을 완전하게 재구성할 필요는 없고(단계 C01)의 동작 c)), 상기 왜곡 기준(여기서 수정된 스칼라 곱)은 리더의 비제로 구성요소들에 대해서만 계산된다. 따라서, 이는 각 리더에 대해  $L_j$ 의 리더들의 인덱스들로부터  $L^0$ 의 리더  $x^j$ 의 인덱스  $l_m$ 과  $L_j$ 의 리더  $x^j$ 의 각 인덱스  $m^j$ 와 관련된  $L^0$ 의 리더들의 인덱스들까지 대응 테이블을 이용하여  $L^0$ 의 대응하는 리더를 결정하는데 충분하다.
- [0369] 그 후에, 상기 알고리즘은 바람직하게는 다음의 예에 따라 실행된다:
- [0370] \* 단계 CP1):
- [0371] 절대 벡터  $|y| = (|y_0|, \dots, |y_k|, \dots, |y_{j-1}|)$  및 그의 부호 벡터  $\varepsilon = (\varepsilon_0, \dots, \varepsilon_k, \dots, \varepsilon_{j-1})$ (여기서,  $y_k \geq 0$ 이면  $\varepsilon_k=1$ 이고, 그렇지 않으면  $\varepsilon_k=-1$ )로의 입력 벡터  $y=(y_0, \dots, y_k, \dots, y_{j-1})$ 의 통과.
- [0372] \* 단계 CP2):
- [0373] 구성요소들을 감소하는 순서로 배치하도록 그의 구성요소들의 치환에 의해  $|y|$ 의 리더  $|\tilde{y}|$ 를 탐색
- [0374] \* 단계 CP3):

- [0375] \* 단계 Co0'): 초기화:
- [0376] 모든 인덱스  $m^j \in [0, L_{D_j}]$ 에 대해
- [0377]  $ps_{max} = -1.; m_{max} = -1; m^j = 0$
- [0378] \* 단계 Co1'): 인덱스  $m^j$ 의 리더의 재구성:
- [0379] a)  $L_j$ 의 리더들과  $L^0$ 의 리더들을 관련시킨 대응 테이블에서,  $L_j$ 의 인덱스  $m^j$ 의 리더와 관련된 리더  $x^{j'}$ 의 인덱스  $l_m$ 의 판독, 그 후에 상기 리더  $x^{j'}$ 의 차원  $j'$ 의 결정 및 정정 인자  $\alpha$  ( $\frac{1}{\alpha} = \sqrt{\sum_{k=0}^{j'-1} (x_k^{j'})^2}$ )의 판독
- [0380] b) 차원  $j'$ 의 리더  $x^{j'}$  및 인덱스  $l_m$ 의 세트  $L^0$ 에서의 판독.
- [0381] \* 단계 Co2')  $|\tilde{y}|$ 와  $x^{j'}$  사이의 수정된 스칼라 곱의 계산:  $ps(|\tilde{y}|, x^{j'}) = \alpha \sum_{k=0}^{j'-1} (\tilde{y}_k \cdot x_k^{j'})$
- [0382] 다음의 단계들은 입력벡터의 절대 리더와의 수정된 스칼라 곱이 최대인 코드 리더의 인덱스를 식별하기 위해 동작들 Co1') 및 Co2')을 반복하는 것이다. 따라서:
- [0383] if  $ps(|\tilde{y}|, x^{j'}) > ps_{max}$  then  $ps_{max} = ps(|\tilde{y}|, x^{j'})$  and  $m_{max} = m^j$
- [0384] \* 다음에,  $m^j$ 를 증분:  $m^j = m^j + 1$
- [0385] \* 테스트 종료
- [0386]  $m^j < L_{D_j}$ 이면 단계 Co1')으로 진행하고, 그렇지 않으면 중단,
- [0387] 상기 종료 단계에서, 단계 CP3)에서 구해진 치환 코드  $m_{max}$ 의 수, 단계 CP2)에서 수행된 치환의 랭크 및 단계 CP1)에서 결정된 부호 벡터에 기초한 치환 코드들의 조합의 인덱세이션의 절차에 의해  $D_i^j$ 의  $y$ 의 최근접 이웃의 인덱스를 계산한다.
- [0388] 주목할 사항으로서, 단계 CP2)는 가속될 수 있다. 구체적으로,  $n_i^j$ 가  $L_j(i)$ 의 리더들의 비제로 구성요소들의 최대 수이면,  $|y|$ 의  $n_i^j$  최대 구성요소들을 탐색하는데 충분하다. 원하는 상기 저장/복잡도 절충안에 의존하여 단계 CP3)의 여러 변형들이 존재한다. 계산들의 수를 최소화하고자 하는 경우에,  $L^0$ 의 모든 리더들을 단순히 그들의 차원  $j'$  및 그들의 정정 인자로 도표화할 수 있다. 단계 CP3)에서 언급된 차원  $j'$ 의 결정은 이 경우에 대응 테이블을 판독하는 것으로 이루어진다. 역으로, 상기 메모리를 감소하고자 하는 경우, 이러한 결정은 인덱스  $l_m$ 에 기초하여 실행된다. 유사하게, 상기 정정 인자는 상기 리더  $x^{j'}$ 의 판독 후에 계산될 수 있다.
- [0389] 따라서, 치환 코드들의 조합으로서의 구조를 이용하여, 상기 디셔너리  $D_i^j$ 의 차원  $j$ 의 입력 벡터  $y$ 의 최근접 이웃을 탐색하는 알고리즘은 바람직하게는 다음과 같이 요약될 수 있다:
- [0390] CP1) 절대 벡터  $|y| = (|y_0|, \dots, |y_k|, \dots, |y_{j-1}|)$  및 그의 부호 벡터  $\varepsilon = (\varepsilon_0, \dots, \varepsilon_k, \dots, \varepsilon_{j-1})$ (여기서,  $y_k \geq 0$ 이면  $\varepsilon_k = 1$ 이고, 그렇지 않으면  $\varepsilon_k = -1$ )로의 입력 벡터  $y = (y_0, \dots, y_k, \dots, y_{j-1})$ 의 통과.
- [0391] CP2) 구성요소들을 감소하는 순서로 배치하도록 그의 구성요소들의 치환에 의해  $|y|$ 의 리더  $|\tilde{y}|$ 를 탐색,
- [0392] CP3) ( $D_i^j$ 의 치환 코드들의 수에 대해  $M_i^j$ 를 기입함으로써  $L_j$ 의  $M_i^j$  첫 번째 리더들 중에서)  $D_i^j$ 의 리더들의 세트  $L_j(i)$ 에서  $|\tilde{y}|$ 의 최근접 이웃을 탐색한다. 상기에 나타난 바와 같이, 상기 단계는  $L_j$ 의 리더들의 인덱스들로부

터  $L^0$ 의 리더들의 인덱스들로의 대응 테이블에 의해 표시된  $L^0$ 의  $M_i^j$  리더들의 리스트 중에 수정된 스칼라 곱을 최대화하는  $L^0$ 의 리더를 탐색하게 된다.  $L^0$ 의 리더  $x^{j'}$ 의 차원이  $j'(j' \leq j)$ 이면,  $|\tilde{y}^j|$ 과의 스칼라 곱의 계산은  $|\tilde{y}^j|$ 의 첫 번째  $j'$  구성요소들에 대해서만 수행되며, 그 후에  $x^{j'}$ 의 유클리드 역 기준에 의해 곱해진다.

[0393] CP4) 그리고 이전의 단계에서 구해진 치환 코드의 수, 단계 CP2)에서 수행된 치환의 랭크 및 단계 CP1)에서 결정된 부호 벡터에 기초하여 치환 코드들의 조합의 인덱세이션 절차에 의해  $D_i^j$ 의  $y$ 의 상기 최근접 이웃의 랭크의 인덱스를 계산한다.

[0394] 요약하면, 단계 CP2)는 가속화될 수 있다. 구체적으로,  $n_i^j$ 가  $L_j(i)$ 의 리더들의 비제로 구성요소들의 최대 수이면,  $|y|$ 의  $n_i^j$  최대 구성요소들을 탐색하는데 충분하다.

[0395] 이제, 유용한 실시예로서 상술한 치환 코드들 조합 인덱세이션을 제한적으로 이용할 필요없이, 일반적 의미 내의 디코딩 알고리즘을 설명한다. 상기 디코딩 알고리즘은 바람직하게는 다음의 형태를 취한다.

[0396] 수신한 인덱스  $m_j$ 에 기초하여, 상기 인덱스가  $\overline{D_{N_j-1}^j}$  또는  $D_{N_j-1}^j$ 에 속하는 코드벡터에 대응하는지 여부를 결정한다.

[0397] 제 1 경우에,  $m_j$ 는  $\overline{D_{N_j-1}^j}$ 의 고유한 인덱스와 관련되며, 상기 코드벡터는 대응 테이블의 간단한 판독을 통해 획득된다.

[0398] 제 2 경우에,  $m_j$ 는 요소  $\overline{D_{N_j-1}^j}$  ( $j' < j$ ) 및 삽입 규칙을 지시한다.

[0399]  $x_{m_j}^j$ 가  $D_{N_j-1}^j$  또는 그의 보수에 속하는지 여부를 결정은 다양한 방식으로 수행될 수 있다. 예를 들어, 각 인덱스에 대한 이진 표시를 이용할 수 있다. 또한, 각 분해능  $r_i$ 에 대해,  $\overline{D_i^j}$ 에 속하는 "자유로운" 요소들을 수반하는,  $D_i^j$ 에 속하는 제어된 확장에 의해 획득된 요소들로 시작하여,  $D_i^j$ 의 보수  $D_{i-1}^j$ 의 요소들을 인덱싱할 수 있다.  $D_{N_j-1}^j$  또는  $\overline{D_{N_j-1}^j}$ 의 멤버십은 단순한 테스트들을 통해 이루어진다. 유사하게, 상기 삽입 규칙은 명시적으로 인덱싱될 수 있다.

[0400] 예를 들어, 이하에 설명된 실시예들에서, 상기 삽입 규칙은 상기 인덱스에 기초하여 암시적으로 검색된다. 또한, 이해되는 바와 같이, 상기 저장/인덱세이션 복잡도 절충안은 상기 애플리케이션의 요건들의 기능으로서 맞춰질 수 있다.

[0401] 여기서, 치환 코드들의 조합에 의해 정의된 추가의 제약의 특정 경우로 돌아간다.

[0402] 우선적으로, 상기 디코딩 알고리즘은 상기 문서로부터의 착안점을 취한다:

[0403] - "Algorithme de Quantification Vectorielle Algebrique Spherique par le Reseau de Gosset  $E_8$ ", C. Lamblin, J.P. Adoul, Annales Des Telecommunications, No. 3-4, 1988, using in addition the correspondence table from the indices of leaders of  $L_j$  to those of  $L^0$ .

[0404]  $D_i^j$ 의 코드벡터의 인덱스에 기초하여,  $L_j(i)$ 의 리더의 인덱스, 그의 치환 코드의 랭크 및 비제로 구성요소들의 부호를 결정한다. 그 후에, 대응 테이블은 상기 디코딩된 코드벡터를 정규화할 수 있게 하는 정규화 인자뿐만 아니라 메모리에 저장된 테이블의 간단한 판독을 통해 획득되는  $L^0$ 의 리더의 인덱스를 제공한다.

[0405] 본 발명의 또 하나의 예시적인 구현이 이하에 제공된다. 상기 예는 상기 TDAC 타입 변환 기반 코더에 기초하지만, 16 kHz에서 샘플링된 디지털 오디오 신호들을 코딩하기 위해 광대역에서의 TDAC 코더 사용의 상기에 주어진 예와 반대로, 32 kHz 및 대역폭 15 kHz(FM 대역)에서 샘플링된 디지털 오디오 신호들을 코딩하기 위한 이용을

위한 것이다.

- [0406] 상기 코더의 원리는 16 kHz에서의 TDAC 광대역 코더의 원리와 유사하다. 16 kHz로 제한된 대역 및 현재 32 kHz에서 샘플링된 오디오 신호는 20 ms의 프레임들로 분배된다. 이는 MDCT 변환 후에 640 계수들의 획득을 야기한다. 상기 스펙트럼은 동일하지 않은 폭들의 52 대역들로 분배되며, 상기 광대역의 분배는 상기 광대역 TDAC 코더에 의해 수행된 분배와 동일하다.
- [0407] 도 8a의 어레이는 사용된 대역 분배 및 계수들의 벡터의 최종 차원(상기 제 3 열에 표시된 계수들의 수에 대응함)을 제공한다.
- [0408] 상기 스펙트럼 포락선의 양자화는 호프만 코딩을 이용하며, 상기 나머지 가변 속도는 상기 스펙트럼 포락선의 역양자화 버전에 기초한 계수들에 동적으로 할당된다.
- [0409] 상기 MDCT 계수들의 양자화는 본 발명에 따라 구성된 디서너리들을 이용한다. 이전에 설명된 경우와 같이, 상기 디서너리들은 또한 치환 코드들의 조합으로서 구조화된다. 15보다 낮은 차원들에 대해, 벡터 양자화기들은 광대역의 벡터 양자화기들과 동일하다. 따라서, 차원들 16, 17, 18, 19, 20 및 24에 대한 디서너리들을 구성한다. 차원 24에 대해, 상기 구조는 데카르트 곱 구조와 결합되었다. 24 계수들의 최종 상부 대역은 차원 12의 2개 벡터들로 분배된다: 하나는 짝수 계수들로 이루어지고, 다른 하나는 홀수 계수들로 이루어진다. 여기서, 차원 12에 대해 구성된 벡터 양자화기들이 이용되었다.
- [0410] 도 8b의 어레이는 차원들 1 내지 24에 대한 그들의 값들뿐만 아니라 서로 다른 분해능들의 수를 제공한다.
- [0411] 따라서, 본 발명은 가변 속도 및 가변 차원의 벡터 양자화의 문제점에 대한 효율적인 솔루션을 제공한다. 본 발명은 공동으로 여러 차원들 및 분해능들에 대한 디서너리들이 상기의 구조화 특성들 PR 및 PD를 소유하는 벡터 양자화기를 제공함으로써 가변 분해능 및 가변 차원의 2가지 문제점들을 해결한다.
- [0412] 주어진 차원에 대해, 한편으로 디서너리들의 내장은 분해능의 함수로서 상기 왜곡의 국부적 감소를 보장하고, 다른 한편으로 저장을 위해 필요한 메모리량을 현저하게 감소시키는데, 그 이유는 더 낮은 분해능들의 디서너리들이 저장되지 않아도 되기 때문이며, 이들 디서너리들의 모든 요소들은 최대 분해능의 디서너리 내에 있다. 도 1a 및 1b의 트리로서 구조화된 벡터 양자화기와 비교할 때, 상기 디서너리들을 내장하는 선택은 아마 2가지 장점들을 제공한다: 증가하는 분해능들의 함수로서 국부적 왜곡의 감소의 보증 및 감소된 저장용량. 필요한 경우, 비트보다 작은 입도가 2의 제곱과 동일하지 않은 크기들의 디서너리들의 선택을 촉진하는 더 큰 분해능의 정교함을 허용한다. 상기 분해능들의 미세한 입도는 이진 트레인으로서 상기 인덱스들을 배치하기 위한 알고리즘과 이들 비정수 벡터 기반 속도 양자화기들을 관련시킴으로써, 가변 차원 및/또는 가변 분해능의 여러 벡터들이 프레임당 양자화되어야 하는 경우에 특히 유용하다.
- [0413] 상기 디서너리들의 내장 특성 PR은 최대 분해능의 디서너리들을 저장하기만 하면 됨을 암시한다. 상기 제 2 특성 PD에 의해, 저장 메모리량은 훨씬 더 감소된다. 구체적으로, 최대 분해능의 디서너리들의 요소들의 일부는 저장될 필요가 없는데, 그 이유는 소정의 삼입 규칙들  $\{R_m\}$ 을 고려하여, 더 낮은 차원의 최대 분해능의 디서너리들로부터 취해진 요소들로부터 추론되기 때문이다. 구조화된 요소들의 비율은 용이하게 적응가능하며, 저장 메모리량을 미세하게 조절할 수 있다.
- [0414] 따라서, 이들 2가지 특성들 PR 및 PD에 의해 유도된 구조는 필요한 저장 메모리를 유용하게 감소시킬 수 있다. 이것은 상기의 종래 기술을 참조한 도입 부분에서 이미 설명된 것과 같은 디서너리들에 추가의 구조적 제약들을 부과함으로써 더욱 명백해진다. 바람직한 실시예들에서, 예를 들어 구면 벡터 양자화기들의 이용을 위해, 상기에 설명된 데카르트 곱 구조에 적절하게 결합된 치환 코드들의 조합이 제공된다.
- [0415] 대수 벡터 양자화기들과 비교할 때, 2가지 특성들에 의해 유도되는 디서너리들의 상기 구조는 차원들의 선택 및 분해능들의 선택의 관점에서 매우 큰 설계 유연성을 제공한다. 또한, 이들 벡터 양자화기들은 코딩되는 소스의 통계에 적응하여, 코딩될 소스의 분배가 균일하게 되도록 대수 벡터 양자화에 의무적인 "벡터 압신"의 까다로운 설계 문제점을 회피할 수 있다.

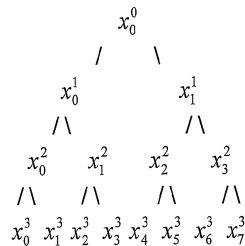
**도면의 간단한 설명**

- [0164] 도 1a 및 1b는 트리-구조화 벡터 양자화기들을 나타낸다.
- [0165] 도 2a는 주어진 차원 N에 대해, 본 발명의 의미 내의 디서너리의 내장 특성을 도시한다.

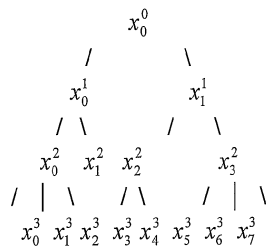
- [0166] 도 2b는 본 발명의 의미 내의 디셔너리의 제어 확장에 의한 부분 합성의 특성을 도시한다.
- [0167] 도 3은 증가하는 분해능들의 기능으로서 상기 디셔너리의 내장을 도시한다.
- [0168] 도 4는 더 낮은 차원들 및 삽입 규칙들의 디셔너리들의 코드벡터들에 기초하여 디셔너리의 코드벡터들의 구성을 도시한다.
- [0169] 도 5는 더 낮은 분해능의 디셔너리들의 재업데이팅없이 내장된 디셔너리들의 증가 분해능들에 따른 구성을 도시한다.
- [0170] 도 6은 상기 "TDAC" 코더의 블록도를 도시한다.
- [0171] 도 7a 내지 도 7g는 본 발명의 의미 내의 벡터 양자화기를 이용하는 광대역 TDAC 코더에 대해:
- [0172] \* 32 대역들로의 분할 (도 7a),
- [0173] \* 차원 당 분해능들 (도 7b),
- [0174] \* 상기 내장 특성에 의해 얻어지는 메모리 이득 (도 7c),
- [0175] \* 내장 및 제어 확장의 2가지 특성들에 의해 얻어지는 메모리 이득 (도 7d),
- [0176] \* 이들 2가지 특성들을 이용하지 않고 디셔너리의 저장을 위해 필요한 메모리 크기에 관하여, 차원 및 속도의 기능으로서 각각 2개의 구조화 특성들에 의해 얻어지는 메모리 이득 (도 7e),
- [0177] \* 차원들(1, 2, 3)의 세트  $L^0$ 의 제 1 리더들 (도 7f), 및
- [0178] \* 차원(3)의 디셔너리들의 치환 코드들의 리더들 (도 7g)를 예시하는 어레이들을 나타낸다.
- [0179] 도 8a 및 8b는 상기 FM 대역 TDAC 코더에 대해, 각각:
- [0180] \* 52 대역들로의 분배(도 8a), 및
- [0181] \* 차원 당 분해능들(도 8b)을 예시하는 어레이들을 나타낸다.

**도면**

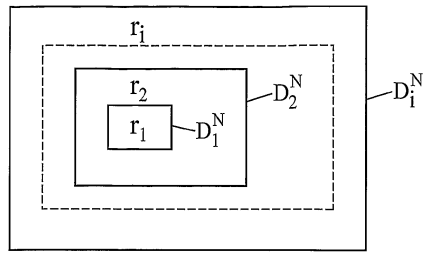
**도면1a**



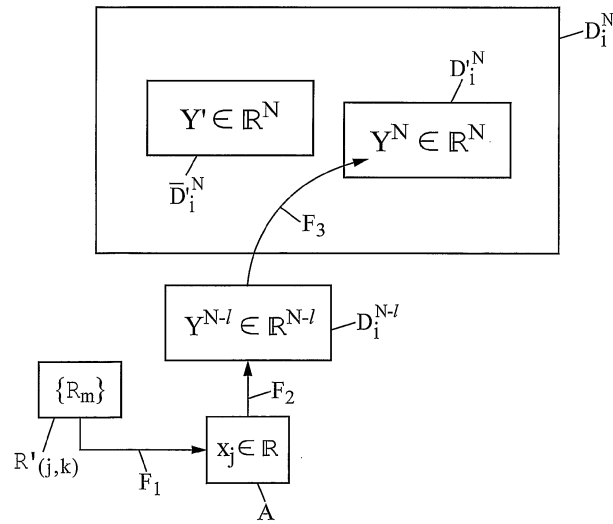
**도면1b**



도면2a



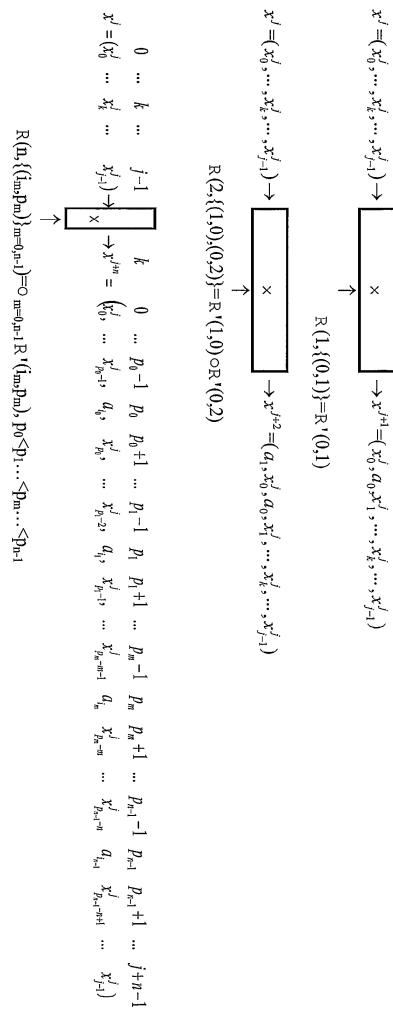
도면2b



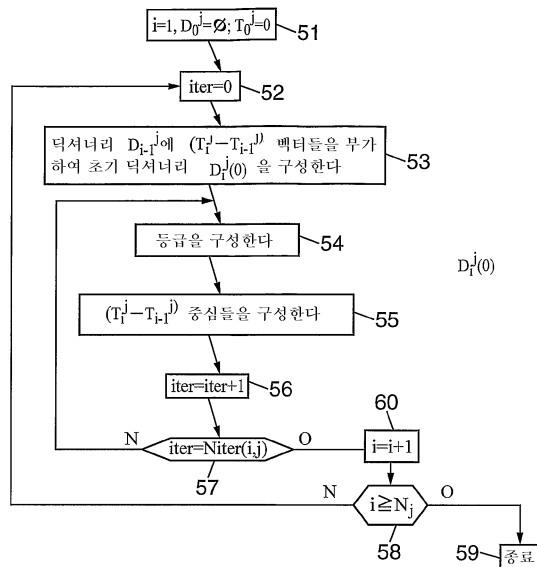
도면3

$$D_{N_j-1}^j = \left( \underbrace{\left( x_0^j, x_1^j, \dots, x_{r_0^j-1}^j \right)}_{D_0^j} \mid \underbrace{\left( x_{r_0^j}^j, \dots, x_{r_1^j-1}^j \right)}_{D_1^j} \mid \dots \mid \underbrace{\left( x_{r_{j-2}^j}^j, \dots, x_{r_{N_j-1}^j-1}^j \right)}_{D_{j-1}^j} \right)$$

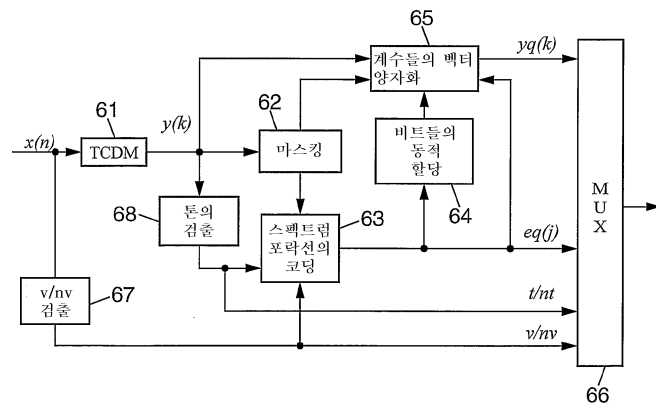
도면4



도면5



도면6



도면7a

대역	상한 (Hz)	계수들의 수	대역	상한 (Hz)	계수들의 수
0	75	3	16	2375	10
1	150	3	17	2625	10
2	225	3	18	2875	10
3	300	3	19	3175	12
4	375	3	20	3475	12
5	475	4	21	3775	12
6	575	4	22	4075	12
7	675	4	23	4400	13
8	800	5	24	4725	13
9	925	5	25	5050	13
10	1050	5	26	5400	14
11	1225	7	27	5750	14
12	1425	8	28	6100	14
13	1650	9	29	6475	15
14	1875	9	30	6850	15
15	2125	10	31	7225	15

도면7b

j	N <sub>j</sub>	멘트당 레이트 {j <sup>r</sup> <sub>0</sub> <sup>j</sup> , j <sup>r</sup> <sub>1</sub> <sup>j</sup> , ..., j <sup>r</sup> <sub>i</sub> <sup>j</sup> , ..., j <sup>r</sup> <sub>N<sub>j</sub>-1</sub> <sup>j</sup> }
1	1	1
2	5	2, 3, 4, 5, 6
3	8	3, 4, 5, 6, 7, 8, 9, 10
4	10	3, 5, 6, 7, 8, 9, 10, 11, 12, 13
5	13	4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17
6	15	4, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20
7	18	4, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23
8	20	4, 7, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26
9	23	5, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29
10	25	5, 8, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32
11	25	5, 8, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32
12	25	5, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32
13	25	5, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32
14	24	5, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32
15	24	5, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32

도면7c

j	N <sub>j</sub>	$\sum_{i=1}^{N_j-1} L_{D_i^j}$	$j \sum_{i=1}^{N_j-1} L_{D_i^j}$	L <sub>D<sup>j</sup></sub>	jL <sub>D<sup>j</sup></sub>	$jL_{D_i^j} / j \sum_{i=1}^{N_j-1} L_{D_i^j}$
1	1	1	1	1	1	1.00
2	5	18	36	7	14	0.39
3	8	71	213	23	69	0.32
4	10	166	664	50	200	0.30
5	13	418	2090	98	490	0.23
6	15	695	4170	146	876	0.21
7	18	1095	7665	209	1463	0.19
8	20	1620	12960	281	2248	0.17
9	23	2251	20259	354	3186	0.16
10	25	3057	30570	437	4370	0.14
11	25	2447	26917	369	4059	0.15
12	25	2101	25212	321	3852	0.15
13	25	1826	23738	268	3484	0.15
14	24	1608	22512	233	3262	0.14
15	24	1447	21705	205	3075	0.14

도면7d

j	L <sub>D<sup>j</sup></sub>	$\sum_{k=1}^j L_{D^k}$	$\sum_{k=1}^j kL_{D^k}$	L <sub>j</sub>	$\sum_{k=1}^j L_k$	$\sum_{k=1}^j kL_k$	$\sum_{k=1}^j kL_k / \sum_{k=1}^j kL_{D^k}$
1	1	1	1	1	1	1	1.00
2	7	8	15	6	7	13	0.87
3	23	31	84	16	23	61	0.73
4	50	81	284	27	50	169	0.60
5	98	179	774	48	98	409	0.53
6	146	325	1650	48	146	697	0.42
7	209	534	3113	63	209	1138	0.37
8	281	815	5361	72	281	1714	0.32
9	354	1169	8547	73	354	2371	0.28
10	437	1606	12917	83	437	3201	0.25
11	369	1975	16976	10	447	3311	0.20
12	321	2296	20828	33	480	3707	0.18
13	268	2564	24312	19	499	3954	0.16
14	233	2797	27574	10	509	4094	0.15
15	205	3002	30649	7	516	4199	0.14

도면7e

j	$N_j$	$\sum_{i=1}^{N_j} L_{D_i^j}$	$j \sum_{i=1}^{N_j} L_{D_i^j}$	$\sum_{k=1}^i k \sum_{i=1}^{N_k} L_{D_i^k}$	$L_j$	$\sum_{k=1}^j k L_k$	$\sum_{k=1}^j k L_k / \sum_{k=1}^j k \sum_{i=1}^{N_k} L_{D_i^k}$
1	1	1	1	1	1	1	1
2	5	18	36	37	6	13	0,351
3	8	71	213	250	16	61	0,244
4	10	166	664	914	27	169	0,185
5	13	418	2090	3004	48	409	0,136
6	15	695	4170	7174	48	697	0,097
7	18	1095	7665	14839	63	1138	0,077
8	20	1620	12960	27799	72	1714	0,062
9	23	2251	20259	48058	73	2371	0,049
10	25	3057	30570	78628	83	3201	0,041
11	25	2447	26917	105545	10	3311	0,031
12	25	2101	25212	130757	33	3707	0,028
13	25	1826	23738	154495	19	3954	0,026
14	24	1608	22512	177007	10	4094	0,023
15	24	1447	21705	198712	7	4199	0,021

도면7f

l	j	$l^j$	리더	l	j	$l^j$	리더
0	1	0	1	12	3	5	4 1 1
1	2	0	1 1	13	3	6	4 4 1
2	2	1	2 1	14	3	7	5 1 1
3	2	2	3 1	15	3	8	6 1 1
4	2	3	4 1	16	3	9	3 2 1
5	2	4	5 1	17	3	10	4 2 1
6	2	5	9 1	18	3	11	4 3 1
7	3	0	1 1 1	19	3	12	5 2 1
8	3	1	2 1 1	20	3	13	5 4 1
9	3	2	2 2 1	21	3	14	7 3 1
10	3	3	3 1 1	22	3	15	8 2 1
11	3	4	3 3 1				

도면7g

$m^3$	i	$j r_i$	리더 $x^3$	$j'$	$x^{j'}$	$l_m$	$m^3$	i	$j r_i$	리더 $x^3$	$j'$	$x^{j'}$	$l_m$
0	0	3	1 0 0	1	1	0	12	6	9	4 1 1	3	4 1 1	12
1	1	4	1 1 1	3	1 1 1	7	13	6	9	4 4 1	3	4 4 1	13
2	2	5	1 1 0	2	1 1	1	14	6	9	5 1 1	3	5 1 1	14
3	3	6	3 1 0	2	3 1	3	15	6	9	6 1 1	3	6 1 1	15
4	4	7	4 1 0	2	4 1	4	16	6	9	3 2 1	3	3 2 1	16
5	4	7	5 1 0	2	5 1	5	17	6	9	4 2 1	3	4 2 1	17
6	4	7	9 1 0	2	9 1	6	18	6	9	4 3 1	3	4 3 1	18
7	5	8	2 1 0	2	2 1	2	19	7	10	5 2 1	3	5 2 1	19
8	5	8	2 1 1	3	2 1 1	8	20	7	10	5 4 1	3	5 4 1	20
9	5	8	2 2 1	3	2 2 1	9	21	7	10	7 3 1	3	7 3 1	21
10	5	8	3 1 1	3	3 1 1	10	22	7	10	8 2 1	3	8 2 1	22
11	5	8	3 3 1	3	3 3 1	11							

도면8a

대역	상한 (Hz)	계수들의 수	대역	상한 (Hz)	계수들의 수	대역	상한 (Hz)	계수들의 수
0	75	3	18	2875	10	35	8725	15
1	150	3	19	3175	12	36	9100	15
2	225	3	20	3475	12	37	9500	16
3	300	3	21	3775	12	38	9900	16
4	375	3	22	4075	12	39	10300	16
5	475	4	23	4400	13	40	10700	16
6	575	4	24	4725	13	41	11125	17
7	675	4	25	5050	13	42	11575	18
8	800	5	26	5400	14	43	12025	18
9	925	5	27	5750	14	44	12475	18
10	1050	5	28	6100	14	45	12925	18
11	1225	7	29	6475	15	46	13400	19
12	1425	8	30	6850	15	47	13900	20
13	1650	9	31	7225	15	48	14400	20
14	1875	9	32	7600	15	49	14900	20
15	2125	10	33	7975	15	50	15400	20
16	2375	10	34	8350	15	51	16000	24
17	2625	10						

도면8b

j	$N_j$	$\{jr'_0, jr'_1, \dots, jr'_i, \dots, jr'_{N_j-1}\}$
1	1	1
2	5	2, 3, 4, 5, 6
3	8	3, 4, 5, 6, 7, 8, 9, 10
4	10	3, 5, 6, 7, 8, 9, 10, 11, 12, 13
5	13	4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17
6	15	4, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20
7	18	4, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23
8	20	4, 7, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26
9	23	5, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29
10	25	5, 8, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32
11	25	5, 8, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32
12	25	5, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32
13	25	5, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32
14	24	5, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32
15	24	5, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32
16	23	9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 31
17	23	10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 31
18	23	10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 31
19	23	10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 31
20	23	10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 31
21	0	$\emptyset$
22	0	$\emptyset$
23	0	$\emptyset$
24	24	9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 31