(19) Europäisches Patentamt / European Patent Office / Office européen des brevets

(11) EP 1 514 355 B1

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent: **19.08.2009 Bulletin 2009/34**

(21) Application number: **03727093.1**

(22) Date of filing: **30.05.2003**

(51) Int Cl.:
H03M 7/30 (2006.01) *G10L 19/10* (2006.01)
G10L 19/06 (2006.01)

(86) International application number: **PCT/CA2003/000829**

(87) International publication number: **WO 2003/103151 (11.12.2003 Gazette 2003/50)**

(54) **METHOD AND SYSTEM FOR MULTI-RATE LATTICE VECTOR QUANTIZATION OF A SIGNAL**

VERFAHREN UND SYSTEM FÜR DIE MULTI-RATE GITTER BASIERTE VEKTOR-QUANTISIERUNG EINES SIGNALS

PROCEDE ET SYSTEME DE QUANTIFICATION VECTORIELLE MULTI-DEBIT EN TREILLIS D'UN SIGNAL

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LI LU MC NL PT RO SE SI SK TR

(30) Priority: **31.05.2002 CA 2388358**

(43) Date of publication of application: **16.03.2005 Bulletin 2005/11**

(73) Proprietor: **Voiceage Corporation**
Ville Mont-Royal,
Quebec H3R 2H6 (CA)

(72) Inventors:
- **BESSETTE, Bruno**
  Rock Forest, Québec J1N 1L2 (CA)
- **RAGOT, Stéphane**
  Sherbrooke, Quebec J1K 2R5 (CA)
- **ADOUL, Jean-Pierre**
  F-34170 Castelnau Le Lez (FR)

(74) Representative: **Schmit, Christian Norbert Marie**
SCHMIT CHRETIEN
111, cours du Médoc
33300 Bordeaux (FR)

(56) References cited:
- **MINJIE XIE ET AL: "Embedded algebraic vector quantizers (EAVQ) with application to wideband speech coding" 1996 IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING CONFERENCE PROCEEDINGS (CAT. NO.96CH35903), 1996 IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING CONFERENCE PROCEEDINGS, ATLANTA, GA, USA, 7-10 M, pages 240-243 vol. 1, XP002252600 1996, New York, NY, USA, IEEE, USA ISBN: 0-7803-3192-3 cited in the application**
- **HAHM CHEUL-HEE ET AL: "Overload vector quantisation by selective shell projection in lattice VQ" ELECTRONICS LETTERS, IEE STEVENAGE, GB, vol. 32, no. 24, 21 November 1996 (1996-11-21), pages 2207-2209, XP006005996 ISSN: 0013-5194**
- **YING ZHANG ET AL: "An improvement image vector quantization based on affine transformation" SYSTEMS, MAN, AND CYBERNETICS, 1997. COMPUTATIONAL CYBERNETICS AND SIMULATION., 1997 IEEE INTERNATIONAL CONFERENCE ON ORLANDO, FL, USA 12-15 OCT. 1997, NEW YORK, NY, USA, IEEE, US, 12 October 1997 (1997-10-12), pages 1094-1099, XP010249571 ISBN: 0-7803-4053-1**

Printed by Jouve, 75001 PARIS (FR)

Description

FIELD OF THE INVENTION

[0001] The present invention relates to encoding and decoding of signals. More specifically, the present invention is concerned with a method and system for multi-rate lattice vector quantization of a signal to be used, for example, in digital transmission and storage systems.

BACKGROUND OF THE INVENTION

[0002] A classical prior-art technique for the coding of digital speech and audio signals is transform coding, whereby the signal to be encoded is divided in blocks of samples called frames, and where each frame is processed by a linear orthogonal transform, e.g. the discrete Fourier transform or the discrete cosine transform, to yield transform coefficients, which are then quantized.

[0003] Figure 1 of the appended drawings shows a high-level framework for transform coding. In this framework, a transform T is applied in an encoder to an input frame giving transform coefficients. The transform coefficients are quantized with a quantizer Q to obtain an index or a set of indices for characterizing the quantized transform coefficients of the frame. The indices are in general encoded into binary codes which can be either stored in a binary form in a storage medium or transmitted over a communication channel. In a decoder, the binary codes received from the communication channel or retrieved from the storage medium are used to reconstruct the quantized transform coefficients with a decoder of the quantizer Q^{-1}. The inverse transform T^{-1} is then applied to these quantized transform coefficients for reconstructing the synthesized frame.

[0004] In vector quantization (VQ), several samples or coefficients are blocked together in vectors, and each vector is approximated (quantized) with one entry of a codebook. The entry selected to quantize the input vector is typically the nearest neighbor in the codebook according to a distance criterion. Adding more entries in a codebook increases the bit rate and complexity but reduces the average distortion. The codebook entries are referred to as codevectors.

[0005] To adapt to the changing characteristics of a source, adaptive bit allocation is normally used. With adaptive bit allocation, different codebook sizes may be used to quantize a source vector. In transform coding, the number of bits allocated to a source vector typically depends on the energy of the vector relative to other vectors within the same frame, subject to a maximum number of available bits to quantize all the coefficients. Figures 2a and 2b detail the quantization blocks of the Figure 1 in the general context of a multi-rate quantizer. This multi-rate quantizer uses several codebooks typically having different bit rates to quantize a source vector x. This source vector is typically obtained by applying a transform to the signal and taking all or a subset of the transform coefficients

[0006] Figure 2(a) depicts an encoder of the multi-rate quantizer, denoted by Q, that selects a codebook number n and a codevector index i to characterize a quantized representation y for the source vector x. The codebook number n specifies the codebook selected by the encoder while the index i identifies the selected codevector in this particular codebook. In general, an appropriate lossless coding technique can be applied to n and i in blocks E_n and E_i, respectively, to reduce the average bit rate of the coded codebook number n_E and index i_E prior to multiplexing (MUX) them for storage or transmission over a communication channel.

[0007] Figure 2(b) shows decoding operations of the multi-rate quantizer. First, the binary codes n_E and i_E are demultiplexed (DEMUX) and their lossless codes are decoded in blocks D_n and D_i, respectively. The retrieved codebook number n and index i are conducted to the decoder of the multi-rate quantizer, denoted by Q-1, that uses them to recover the quantized representation y of the source vector x. Different values of n usually result in different bit allocations, and equivalently different bit rates, for the index i. The codebook bit rate given in bits per dimension is defined as the ratio between the number of bits allocated to a source vector and the dimension of the source vector.

[0008] The codebook can be constructed using several approaches. A popular approach is to apply a training algorithm (e.g. the k-means algorithm) to optimize the codebook entries according to the source distribution. This approach yields an unstructured codebook, which typically has to be stored and searched exhaustively for each source vector to quantize. The limitations of this approach are thus its memory requirements and computational complexity, which increase exponentially with the codebook bit rate. These limitations are even amplified if a multi-rate quantization scheme is based on unstructured codebooks, because in general a specific codebook is used for each possible bit allocation.

[0009] An alternative is to use constrained or structured codebooks, which reduce the search complexity and in many cases the storage requirements.

[0010] Two instances of structured vector quantization will now be discussed in more detail: multi-stage and lattice vector quantization.

[0011] In multi-stage vector quantization, a source vector x is quantized with a first-stage codebook C_1 into a codevector y_1. To reduce the quantization error, the residual error $e_1 = x - y_1$ of the first stage, which is the difference between the input vector x and the selected first-stage codevector y_1, is then quantized with a second-stage codebook C_2 into a codevector y_2. This process may be iterated with subsequent stages up to the final stage, where the residual error $e_{n-1} = x - y_{n-1}$ of the $(n - 1)$th stage is quantized with an nth stage codebook C_n into a codevector y_n.

[0012] When n stages are used ($n \geq 2$), the reconstruction can then be written as a sum of the codevectors $y = y_1 + ... + y_n$, where y_l is an entry of the lth stage codebook C_l for $l = 1, ..., n$. The overall bit rate is the sum of the bit rates of all n codebooks.

[0013] In lattice vector quantization, also termed lattice VQ or algebraic VQ for short, the codebook is formed by selecting a subset of lattice points in a given lattice.

[0014] A lattice is a linear structure in N dimensions where all points or vectors can be obtained by integer combinations of N basis vectors, that is, as a weighted sum of basis vectors with signed integer weights. Figure 3 shows an example in two dimensions, where the basis vectors are v_1 and v_2. The lattice used in this example is well-known as the hexagonal lattice denoted by A_2. All points marked with crosses in this figure can be obtained as

$$y = k_1 v_1 + k_2 v_2 \qquad \text{(Eq. 1)}$$

where y is a lattice point, and k_1 and k_2 can be any integers. Note that Figure 3 shows only a subset of the lattice, since the lattice itself extends to infinity. We can also write Eq. 1 in matrix form

$$y = [y_1 \quad y_2] = [k_1 \quad k_2]\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = [k_1 \quad k_2]\begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix} \qquad \text{(Eq. 2)}$$

where the basis vectors $v_1 = [v_{11}\ v_{12}]$ and $v_2 = [v_{21}\ v_{22}]$ form the rows of the generator matrix. A lattice vector is then obtained by taking an integer combination of these row vectors.

[0015] When a lattice is chosen to construct the quantization codebook, a subset of points is selected to obtain a codebook with a given (finite) number of bits. This is usually done by employing a technique called shaping. Shaping is performed by truncating the lattice according to a shaping boundary. The shaping boundary is typically centered at the origin but this does not have to be the case, and may be for instance rectangular, spherical, or pyramidal. Figure 3 shows an example with a spherical shaping boundary.

[0016] The advantage of using a lattice is the existence of fast codebook search algorithms which can significantly reduce the complexity compared to unstructured codebooks in determining the nearest neighbor of a source vector x among all lattice points inside the codebook. There is also virtually no need to store the lattice points since they can be obtained from the generator matrix. The fast search algorithm generally involve rounding off to the nearest integer the elements of x subject to certain constraints such that the sum of all the rounded elements is even or odd, or equal to some integer in modulo arithmetic. Once the vector is quantized, that is, once the nearest lattice point inside the codebook is determined, usually a more complex operation consists of indexing the selected lattice point.

[0017] A particular class of fast lattice codebook search and indexing algorithms involves the concept of leaders, which is described in detail in the following references:

- C. Lamblin and J.-P. Adoul. Algorithme de quantification vectorielle sphérique à partir du réseau de Gosset d'ordre 8. Ann. Télécommun., vol. 43, no. 3-4, pp. 172-186, 1988 (Lamblin, 1988);
- J.-M. Moureaux, P. Loyer, and M. Antonini. Low-complexity indexing method for Zn and Dn lattice quantizers. IEEE Trans. Communications, vol. 46, no. 12, Dec. 1998 (Moureaux, 1998); and in
- P. Rault and C. Guillemot. Indexing algorithms for Zn, An, Dn, and Dn++ lattice vector quantizers. IEEE Transactions on Multimedia, vol. 3, no. 4, pp. 395-404, Dec. 2001 (Rault, 2001).

[0018] A leader is a lattice point with components sorted, by convention, in descending order. An absolute leader is a leader with all non-negative components. A signed leader is a leader with signs on each component. Usually the lattice structure imposes constraints on the signs of a lattice point, and thus on the signs of a leader. The concept of leaders will be explained in more details hereinbelow.

[0019] A lattice often used in vector quantization is the Gosset lattice in dimension 8, denoted by RE_8. Any 8-dimensional lattice point y in RE_8 can be generated by

$$y = [k_1 \ k_2 \ \dots \ k_8] G_{RE_8} \qquad \text{(Eq. 3)}$$

where $k_1, k_2, \ldots, k_8$ are signed integers and G_{RE8} is the generator matrix, defined as

$$G_{RE_8} = \begin{bmatrix} 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \end{bmatrix} \qquad \text{(Eq. 4)}$$

[0020] The row vectors $v_1, v_2, \ldots, v_8$ are the basis vectors of the lattice. It can be readily checked that the inverse of the generator matrix G_{RE_8} is

$$G_{RE_8}^{-1} = \frac{1}{4} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 5 & -2 & -2 & -2 & -2 & -2 & -2 & 4 \end{bmatrix} \qquad \text{(Eq. 5)}$$

[0021] This inverse matrix is useful to retrieve the basis expansion of y:

$$[k_1 \ k_2 \ \dots \ k_8] = y G_{RE_8}^{-1} \qquad \text{(Eq. 6)}$$

[0022] It is well-known that lattices consist of an infinite set of embedded spheres on which lie all lattice points. These spheres are often referred to as shells. Lattice points on a sphere in RE_8 can be generated from one or several leaders by permutation of their signed components. All permutations of a leader's components are lattice points with the same norm, and thus they fall on the same lattice shell. Leaders are therefore useful to enumerate concisely the shells of a lattice. Indeed, lattice points located on shells close to the origin can be obtained from a very small number of leaders. Only absolute leaders and sign constraints are required to generate all lattice points on a shell.

[0023] To design a RE_8 codebook, a finite subset of lattice points may be selected by exploiting the intrinsic geometry of the lattice, especially its shell structure. As described in (Lamblin, 1988), the lth shell of RE_8 has a radius $\sqrt{8l}$ where l is a non-negative integer. High radius shells comprise more lattice points than lower radius shells. It is possible to enumerate all points on a given shell using absolute and signed leaders, noting that there is a fixed number of leaders on a shell and that all other lattice points on the shell are obtained by permutations of the signed leader components, with some restrictions on the signs.

[0024] In spherical lattice VQ, it is sufficient to reorder in decreasing order the components of x and then perform a nearest-neighbor search among the leaders defining the codebook to determine the nearest neighbor of a source vector x among all lattice points in the codebook. The index of the closest leader and the permutation index obtained indirectly from the ordering operation on x are then sent to the decoder, which can reconstruct the quantized analog of x from this information. Consequently, the concept of leaders allows a convenient indexing strategy, where a lattice point can be described by a cardinality offset referring to a signed leader and a permutation index referring to the relative index of a permutation of the signed leader.

[0025] Based on the shell structure of a lattice, and on the enumeration of the lattice in terms of absolute and signed leaders, it is possible to construct a codebook by retaining only the lower radius shells, and possibly completing the codebook with a few additional leaders of higher radius shells. We refer to this kind of lattice codebook generation as near-spherical lattice shaping. This approach is used in M. Xie and J.-P. Adoul, Embedded algebraic vector quantization (EAVQ) with application to wideband audio coding, IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Atlanta, GA, U.S.A, vol. 1, pp. 240-243, 1996 (Xie, 1996).

[0026] For RE_8, the absolute leaders in shells of radius 0 and $\sqrt{8}$ are shown below.

Absolute leader for the shell of radius 0

[0 0 0 0 0 0 0 0]

Absolute leaders for the shell of radius $\sqrt{8}$

[22000000] and [1 1 1 1 1 1 1 1]

[0027] A more complete listing for low-radius shells, for the specific case of RE_8, can be found in Lamblin (1988).

[0028] For lattice quantization to be used in transform coding with adaptive bit allocation, it is desirable to construct multi-rate lattice codebooks. A possible solution consists of exploiting the enumeration of a lattice in terms of leaders in a similar way as in Xie (1996). As explained in Xie, a multi-rate leader-based lattice quantizer may be designed with for instance:

- embedded algebraic codebooks, whereby lower-rate codebooks are subsets of higher-rate codebooks, or
- nested algebraic codebooks, whereby the multi-rate codebooks do not overlap but are complementary in a similar fashion as a nest of Russian dolls.

[0029] In the specific case of Xie, multi-rate lattice quantization uses six of codebooks named Q_0, Q_1, Q_2, ..., Q_5, where the last five codebooks are embedded, i.e. $Q_1 \subset Q_2 \subset ... \subset Q_5$. These codebooks are essentially derived from an 8-dimensional lattice RE_8. Following the notations of Xie, Q_n refers to the nth RE_8 codebook. The bit allocation of codebook Q_n is $4n$ bits corresponding to 2^{4n} entries. The codebook bit rate being defined as the ratio between the number of bits allocated to a source vector and the dimension of the source vector, and in RE_8 quantization, the dimension of the source vector being 8, the codebook bit rate of Q_n is $4n/8 = n/2$ bits per dimension.

[0030] With the technique of Xie, the codebook bit rate cannot exceed 5/2 bits per dimension. Due to this limitation, a procedure must be applied to saturate outliers. An outlier is defined as a point x in space that has the nearest neighbor y in the lattice RE8 which is not in one of the multi-rate codebooks Q_n. In Xie, such points are scaled down by a factor $g > 1$ until x/g is no more an outlier. Apparently the use of g may result in large quantization errors. This problem is fixed in Xie (1996) by normalizing the source vector prior to multi-rate lattice quantization.

[0031] There are disadvantages and limitations in the multi-rate quantization technique of Xie, including:

1. Outlier saturation is usually a computation burden. Further, saturation may degrade significantly the quantization performance (hence quality) in the case of large outliers.
2. The technique handles outliers with saturation and does not allow to allocate more than 20 bits per 8-dimensional vector. This may be a disadvantage in transform coding, since high-energy vectors (which are more likely to be outliers) shall be normally quantized with a small distortion to maximize quality, implying it shall be possible to use a codebook with enough bits allocated to a specific vector.
3. The codebooks Q_2, Q_3, Q_4 and Q_5 of 8, 12, 16 and 20 bits are specified with 3, 8, 23 and 73 absolute leaders, respectively. Since storage requirements and search complexity are closely related to the number of absolute leaders, the complexity of these lattice codebooks explodes with increasing codebook bit rate.
4. The performance of embedded codebooks is slightly worse than that of non-overlapping (i.e. nested) codebooks.

[0032] Another kind of lattice shaping, as opposed to near-spherical shaping, is Voronoi shaping, which is described in J.H. Conway and N.J.A. Sloane, A fast encoding method for lattice codes and quantizers, IEEE Trans. Inform. Theory, vol. IT-29, no. 6, pp. 820-824, Nov. 1983 (Conway, 1983). It relies on the concept of Voronoi region described for instance in A. Gersho and R.M. Gray, Vector Quantization and Signal Compression, KluwerAcademic Publishers, 1992 (Gersho, 1992). In the specific case of a lattice codebook, a Voronoi region is the region of space where all points in *N*-dimensional space are closer to a given lattice point than any other point in the lattice. Each lattice point has an associated closed Voronoi region that includes also the border points equidistant to neighboring lattice points. In a given lattice, all Voronoi regions have the same shape, that is, they are congruent. This is not the case for an unstructured codebook.

[0033] A Voronoi codebook is a subset of a lattice such that all points of the codebook fall into a region of space with same shape as the Voronoi region of the lattice, appropriately scaled up and translated. To be more precise, a Voronoi codebook $V^{(r)}$ derived from the lattice Λ in dimension N is defined as

$$V^{(r)} = \Lambda \cap \left(2^r V_\Lambda(0) + a\right) \qquad \text{(Eq. 7)}$$

where r is is a non-negative integer parameter defined later in more detail, $V_\Lambda(0)$ is the Voronoi region of Λ around the origin, and a an appropriate *N*-dimensional offset vector. Equation 7 is interpreted as follows: "the Voronoi codebook $V^{(r)}$ is defined as all points of the lattice Λ included in the region of *N*-dimensional space inside a scaled-up and translated Voronoi region $V_\Lambda(0)$, with the scaling factor $m = 2^r$ and the offset vector a". With such a definition, the codebook bit rate of $V^{(r)}$ is r bits per dimension. The role of a is to fix ties, that is, to prevent any lattice point to fall on the shaping region $2^r V_\Lambda(0) + a$.

[0034] Figure 4 illustrates Voronoi coding, Voronoi regions, and tiling of Voronoi regions in the two-dimensional hexagonal lattice A_2. The point o refers to the origin. Both points o and z fall inside the same boundary marked with dashed lines. This boundary is actually a Voronoi region of A_2 scaled by $m = 2$ and slightly translated to the right to avoid lattice points on the region boundary. There are in total 4 lattice points marked with three dots (·) and a plus (+) sign within the boundary comprising o and z. More generally each such a region contains m^N points. It can be seen in Figure 4 that the same pattern, a Voronoi region of A_2 scaled by $m = 2$, is duplicated several times. This process is called tiling. For instance, the points o' and z' can be seen as equivalent to o and z, respectively, with respect to tiling. The point z' may be written as $z' = o' + z$ where o' is a point of $2A_2$. The points of $2A_2$ are shown with plus signs in Figure 4. More generally, the whole lattice can be generated by tiling all possible translations of a Voronoi codebook by points of the lattice scaled by m.

[0035] As described in D. Mukherjee and S.K. Mitra, Vector set-partitioning with successive refinement Voronoi lattice VQ for embedded wavelet image coding, Proc. ICIP, Part I, Chicago, IL, Oct. 1998, pp. 107-111 (Mukherjee, 1998), Voronoi coding can be used to extend lattice quantization by successive refinements. The multi-stage technique of Mukherjee produces multi-rate quantization with finer granular descriptions after each refinement. This technique, which could be used for multi-rate quantization in transform coding, has several limitations:

1. The quantization step is decreased after each successive refinement, and therefore it cannot deal efficiently with large outliers. Indeed, if a large outlier occurs in the first stage, the successive stages cannot reduce efficiently the resulting error, because they are designed to reduce granular noise only. The performance of the first stage is therefore critical.

2. The property of successive refinements implies constraints on the successive quantization steps. This limits the quantization performance.

[0036] It is further known according to Y. Zhang et al. "An Improvement Image Vector Quantization Based on Affine Transformation", SYSTEMS, MAN, AND CYBERNETICS, 1997. COMPUTATIONAL CYBERNETICS AND SIMULATION., 1997 IEEE INTERNATIONAL CONFERENCE ON ORLANDO, FL, USA 12-15 OCT. 1997, NEW YORK, NY, USA,IEEE, US, 12 October 1997 (1997-10-12), pages 1094-1099, a technique for codebook expansion based on affine transformation. Vectors are first quantized with an initial codebook and the quantization distortion is determined. If the distortion is higher than a given threshold, an expanded codebook is generated to quantize the input vector. The technique described is applied to vector quantization using code vectors gained by learning, not to lattice quantizers.

OBJECTS OF THE INVENTION

[0037] An object of the present invention is to provide improved method and system to construct, search and index a multi-rate lattice vector quantizer.

[0038] Another object of the present invention is to provide an improved search and indexing method for lattice codebooks.

SUMMARY OF THE INVENTION

[0039] The above objects are achieved by a multi-rate quantizer using a set of lattice codebooks, called base codebooks, and extension which makes it possible to obtain higher bit rate codebooks from the base codebooks compared to quantizers from the prior art.

[0040] More specifically, in accordance with an aspect of the present invention, there is provided a multi-rate lattice quantization encoding method according to claim 1.

[0041] According to another aspect of the present invention, there is provided a multi-rate lattice quantization decoding method according to claim 20.

[0042] According to a further aspect of the present invention, there is provided a multi-rate lattice quantization encoder according to claim 36.

[0043] Finally, according to a still further aspect of the present invention, there is provided a multi-rate lattice quantization decoder according to claim 38.

[0044] According to methods and systems from the present invention, a two-dimensional base codebook C is extended by scaling it by successive powers of 2 and tiling a Voronoi codebook V(r) around each point of the scaled base codebook. Therefore, the extension method is referred to as Voronoi extension. The extension order r is the number of times the extension is applied. The extended codebooks C(r) comprise more points and extend higher up in the vector space capturing outliers, while keeping the same granularity as the base codebook. This is attained at a cost of increased bit rate required to index not only the base codebook but also the Voronoi codebook, and to transmit side information on the extension order. The bit rate of the multi-rate quantizer obtained by the disclosed means is source-dependent. The number of bits used for indexing the Voronoi codebook is referred to as overhead.

[0045] Other objects, advantages and features of the present invention will become more apparent upon reading the following non restrictive description of illustrative embodiments thereof, given by way of example only with reference to the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0046] In the appended drawings:

Figure 1, which is labeled "prior art", is a block diagram illustrating a transform coder according to the prior art;

Figures 2(a) and 2(b), which are labeled "prior art", are block diagram respectively of the encoder and decoder of a multi-rate quantizer according to a method from the prior art;

Figure 3, which is labeled "prior art", is a schematic view illustrating spherical shaping on a two-dimensional hexagonal lattice A_2, according to a method from the prior art;

Figure 4, which is labeled "prior art", is a schematic view illustrating Voronoi coding, Voronoi regions, and tiling of Voronoi regions in a two-dimensional hexagonal lattice A_2, according to a method from the prior art;

Figure 5 is a graph illustrating points from the hexagonal lattice A_2;

Figure 6 is the graph of Figure 5, including a shaping boundary for defining a base codebook;

Figure 7 is a graph illustrating a base codebook C obtained by retaining only the lattice points that fall into the shaping boundary shown in Figure 6;

Figure 8 is a graph illustrating the base codebook C from Figure 7, with Voronoi regions around each codevector;

Figure 9 is the graph of Figure 8 illustrating the position of a source vector;

Figure 10 is a graph illustrating the base codebook C from Figure 8 scaled by the factor $m = 2$;

Figure 11 is a graph illustrating the scaled base codebook from Figure 10, with the shifted, scaled Voronoi regions, and a Voronoi codebook comprising 4 points;

Figure 12 is the graph from Figure 11, illustrating an extended codebook of order $r = 1$;

Figure 13 is a graph illustrating the extended codebook from Figure 12 with the related Voronoi regions;

Figure 14 is the graph from Figure 13, illustrating the quantized vector y, reconstructed as a sum of the scaled codevector mc and the codevector v of the Voronoi codebook;

Figure 15 is a flow chart illustrating a multi-rate lattice quantization encoding method according a first illustrative embodiment of a second aspect of the present invention;

Figure 16 is a flow chart illustrating a multi-rate lattice quantization decoding method according a first illustrative embodiment of a third aspect of the present invention;

Figure 17 is a flowchart illustrating the generation of the extended codebooks Q_5, Q_6, and of other high-rate codebooks according to an aspect of the present invention;

Figure 18 is a flow chart illustrating a multi-rate lattice quantization encoding method according a second illustrative embodiment of the second aspect of the present invention;

Figures 19A-19B are schematic views illustrating the structure of the codevector index i as produced by the encoding method of Figure 18 respectively in the case where no extension is used, and when extension is used; and

Figure 20 is a flow chart illustrating a multi-rate lattice quantization decoding method according a second illustrative embodiment of the third aspect of the present invention.

DESCRIPTION OF ILLUSTRATIVE EMBODIMENTS

[0047] Turning first to Figures 5 to 14 a method for multi-rate lattice codebooks extension according to a first illustrative embodiment of a first aspect of the present invention will be described. The extension methods according to the present invention will be referred to herein as Voronoi extension methods.

[0048] This first illustrative embodiment is described by way of a two-dimensional example based on an hexagonal lattice A_2.

[0049] For the sake of clarity, key symbols related to the first illustrative embodiment are gathered in Table 1.

Table 1. List of symbols related to Voronoi extension method in accordance with a first illustrative embodiment of the present invention.

Symbol	Definition	Note
A_2	Hexagonal lattice in dimension 2.	
N	Source dimension.	
Λ	Lattice in dimension N.	E.g., $\Lambda = A2$ with $N = 2$.
x	Source vector in dimension N.	
y	Closest lattice point to x in Λ.	
n	Codebook number.	
i	Codevector index. When the extension is not applied, i is an index to C represented with NR bits. With the extension, i is an index to the extended codebook $C^{(r)}$ comprising a multiplex of j and k, where j is an index to C and k is a Voronoi index corresponding to v. In this case, i is represented with $N(R + r)$ bits.	
a	Offset for Voronoi coding, a vector in dimension N.	

(continued)

Symbol	Definition	Note
r	Extension order, a non-negative integer.	
m	Scaling factor of the extension.	$m = 2^r$
c	Base codevector in C.	
v	Voronoi codevector in Λ.	Computed such that v is congruent to y.
C	Base codebook in Λ.	Comprises 2^{NR} entries.
R	Bit rate of the base codebook C in bits per dimension.	
$C^{(r)}$	Extended codebook of order r.	See Eq. 9 for definition.
k	Voronoi index in $V^{(r)}$, represented with Nr bits.	See Eq. 10 for the computation of k.
G_Λ	Generator matrix of Λ.	
w	Difference vector $w = y - v$.	
$V^{(r)}$	Voronoi codebook of order r.	See Eq. 7 for definition.
$V_\Lambda(0)$	Voronoi region of Λ around the origin.	
mod_m	Modulo operation on a vector. If $y = [y_1 \ldots y_8]$, then $\mathrm{mod}_m(y) = [y_1 \bmod m \ldots y_8 \bmod m]$ where mod is the scalar modulo.	

[0050] Figure 5 shows a part of the hexagonal lattice A_2 that extends to infinity. A base codebook is obtained by appropriately shaping this lattice to get a finite set of lattice points. This is illustrated in Figure 6, where a spherical shaping boundary is shown with solid line, and in Figure 7, where only the lattice points inside the shaping boundary are retained. The points inside the shaping boundary comprise the base codebook C. Even though spherical shaping is used in the present illustrative embodiment, other boundaries can alternatively be used, such as a square, a pyramid, a rectangle, etc..

[0051] In this particular example, the base codebook C comprises 31 lattice points, and for the sake of simplicity, we will assume that an index i of 5 bits is used to label this codebook. The Voronoi regions of the base codebook are the hexagonal areas centered around each lattice point shown with dots (·) in Figure 8.

[0052] Figure 9 shows a source vector x in a two-dimensional plane. One can see in this illustrative example that the nearest neighbor y (not shown) of x in the lattice is not an entry of the base codebook C. It is to be noted that the nearest neighbor search is not limited to the base codebook C; the nearest neighbor y being defined as the closest point to x in the whole lattice A_2. In the specific case of Figure 9, y is an *outlier*. It is reminded that a prior art method for dealing with such an outlier y is to scale the codebook by a given factor, for example a power of 2, resulting in a scaled codebook illustrated in Figure 10. However, this would increase the Voronoi region, and thus the granular distortion.

[0053] To keep the same Voronoi region for maintaining the granularity while extending the codebook to include the outlier, the base codebook is scaled by 2, and a Voronoi codebook is inserted around each scaled codevector as shown in Figures 11 and 12. This scaling procedure yields a Voronoi codebook $V^{(1)}$ in two dimensions comprising 4 lattice points and requiring 2 additional bits as an overhead to index it. The resulting extended codebook $C^{(1)}$ is depicted in Figure 13. As can be seen in Figure 13, the nearest neighbor y of x is no more an outlier, since it belongs to the extended codebook. However, 5 + 2 = 7 bits are now required to describe y in the extended codebook compared to the 5 bits required by the base codebook without any extension. As shown in Figure 14, the quantized vector y can be represented as

$$y = m\,c + v \qquad \text{(Eq. 8)}$$

where m is the extension scaling factor (here, $m = 2$), c is a codevector of the base codebook C, and v belongs to the Voronoi codebook used to extend C.

[0054] Following this last two-dimensional example illustrating a method to extend lattice codebooks to prevent saturation, a lattice codebook extension method according to the first aspect of the present invention will now be presented with reference to a second illustrative embodiment.

[0055] It is now assumed that a base codebook C is derived from a lattice Λ in dimension N having a bit rate of R bits per dimension. In other words, C contains 2^{NR} N-dimensional codevectors and requires NR bits for indexing.

[0056] The extension includes scaling the base codebook by successive powers of 2 (2, 4, 8, etc.), and tiling a Voronoi codebook around each point of the scaled base codebook. For this reason, the extension method is referred to herein as *Voronoi extension*. The extension of the base codebook C of order r is the codebook $C^{(r)}$ defined as

$$C^{(r)} = \bigcup_{\substack{c \in C \\ v \in V^{(r)}}} mc + v \qquad \text{(Eq. 9)}$$

where $m = 2^r$ and $V^{(r)}$ is a Voronoi codebook of size $m^N = 2^{rN}$ derived from the same lattice Λ as C. The extension order r defines the number of times the extension has been applied. The extended codebooks comprise more codevectors and consequently use more bits than the base codebook C. The definition in Eq. 9 implies that the extension codebook $C^{(r)}$ requires NR bits for indexing first the base codebook and then Nr bits for the Voronoi codebook, resulting in a total of $N(R + r)$ bits plus side information on the extension order r.

[0057] The scaling of the base codebook by successive powers of 2 allows to have Voronoi indices represented on an exact number of bits (not fractional). However in general, m may be any integer superior or equal to 2.

[0058] Note that the *granularity* of this basic form of Voronoi extension is 1 bit per dimension, since the increment in codebook bit rate is 1 bit per dimension from the rth to the $(r + 1)$th extension.

[0059] It is to be noted that the previous two-dimensional example used a specific base codebook C derived from the lattice A_2. In the example case of Figure 7, $\Lambda = A_2$, $N=2$, and the bit rate of the base codebook $R = 5/2$ bits per dimension.

[0060] A multi-rate lattice quantization encoding method 100 according to the first illustrative embodiment of the second aspect of the present invention will now be described with reference to Figure 15.

[0061] Let x be an N-imensional source vector to be quantized. Let C denote the base codebook derived from a lattice Λ, and define $m\Lambda$ as the lattice Λ scaled by an integer factor $m > 0$. Then, the steps to encode a vector x using C or one of its extensions according to the method 100 are as follows:

[0062] In step 102, the nearest neighbor y of x is determined in the infinite lattice Λ. Step 102 yields a quantized vector y.

[0063] Then, in step 104, it is determined if y is an entry of the base codebook C. If y is in C (step 106), the number of bits used to quantize x is thus NR, which corresponds to the number of bits used by the base codebook. The codebook number n is set to 0 and the encoding method terminates. If y is not in the base codebook C, y is considered an outlier and the method 100 proceeds with step 108, which, with steps 110-118, form a Voronoi extension method according to a third embodiment of the first aspect of the.present invention.

[0064] As discussed hereinbelow, since y is an outlier, more bits are required to quantize x with y compared with the case where y is part of the base codebook. The extension procedure, which is iterative, generates an extended codebook, eventually including a lattice vector y, which can then be indexed properly.

[0065] Step 108 is an initialization step, where the extension order r is set to 1 and the scaling factor m to $2^r = 2$.

[0066] The Voronoi index k is then computed of the lattice point y (step 110) that was the nearest neighbor of vector x in lattice Λ obtained in step 102. The Voronoi index k depends on the extension order r and the scaling factor m. The Voronoi index k is computed via the following modulo operations such that it depends only on the relative position of y in a scaled and translated Voronoi region:

$$k = \text{mod}_m(y G_\Lambda^{-1}) \qquad \text{(Eq. 10)}$$

where G_Λ is the generator matrix of Λ and $\text{mod}_m(\cdot)$ is the componentwise modulo-m operation. Hence, the Voronoi index k is a vector of integers with each component in the interval 0 to $m - 1$.

[0067] In step 112, the Voronoi codevector v is computed from the Voronoi index k given m. This can be implemented, for example, using an algorithm described in Conway (1983)

[0068] The computation of v can be done as follows:

1. computing z=k * G (RE8);
2. finding the nearest neighbour w of 1/m.(z-a) in RE8;
3. computing v=z-m*w.

[0069] In step 114, the difference vector $w = y - v$ is first computed. This difference vector w always belongs to the scaled lattice $m\Lambda$. Then, $c = w/m$ is computed by applying the inverse scaling to the difference vector w. The codevector c belongs to the lattice Λ, since w belongs to $m\,\Lambda$.

[0070] It is then verified if c is in the base codebook C (step 116). If c is not in the base codebook C, the extension order r is incremented by 1, the scaling factor m is multiplied by 2 (step 118), and the Voronoi extension proceed with a new iteration (step 110). However, if c is in C, then an extension order r and a scaling factor $m = 2^r$ sufficiently large to quantize the source vector x with y without saturation has been found. y is then indexed as a base codevector into j (step 120) as in Lamblin (1988). j and k are multiplexed into an index i (step 122) and the codebook number n is set to the extension order ($n = r$) in step 124, which terminates the encoding method 100. As it is well known in the art, the multiplexing includes a concatenation of j and k, which means that the bits of j are followed by the bits of k.

[0071] The output of the quantization method consists of the codebook number n and the index i of the codevector y. If the Voronoi extension is used, $n > 0$. Otherwise $n = 0$. The index i is:

- the index of $y = c$ in the base codebook, if the Voronoi extension is not used,
- the multiplex of j and k, where j is the index of c in the base codebook C and k is the Voronoi index corresponding to the vector v.

[0072] It is to be noted that in Eq. 10 the Voronoi index k is defined as $k = \mathrm{mod}_m(yG_\Lambda^{-1})$, where $m = 2^r$. Since y is a lattice point in Λ, yG_Λ^{-1} actually corresponds to the basis expansion of y in Λ and consequently is an N-dimensional vector of integers. Therefore k is also a vector of N integers, and due to the component-wise modulo operation mod_m each component of k is an integer between 0 and m - 1. Since $m = 2^r$, by construction k requires a total of Nr bits to index all of its N components.

[0073] The quantization method 100 is completed by defining the lossless encoding of the codebook number n and the index i to obtain n_E and i_E to be multiplexed, and stored or transmitted over a communications channel as was illustrated in Figure 2.

[0074] In general, the output of a multi-rate vector quantizer consists of a codebook number n and an index i that may both exhibit statistical redundancy. Without limiting the scope or generality of the present invention, we address here only the entropy coding of the codebook number n to reduce the average bit rate of the quantizer, while no coding is applied to the index i giving $i_E = i$. Any appropriate prior-art lossless coding technique such as arithmetic coding or Huffman coding (Gersho, 1992) may be employed for n. A simple coding method is the unary code, in which a positive integer n is represented in binary form by n- 1 ones, followed by zero. This coding scheme will be described hereinbelow in more detail.

[0075] Turning now to Figure 16 of the appended drawings, a multi-rate lattice quantization decoding method 200 in accordance with the first illustrative embodiment of the third aspect of the present invention will now be described. The encoded codebook number n_E is first read from the channel and the lossless coding technique used in the method 100 is inverted to get the codebook number n (step 202). It is important to note that n indicates the bit allocation of the multi-rate quantizer and is required to demultiplex the quantization index i in step 204.

[0076] If $n = 0$ (step 206), the Voronoi extension is not used. In this case, the index i is decoded to form the codevector c of the base codebook C (step 208) using a prior-art technique such as described in (Lamblin, 1988), (Moureaux, 1998) or (Rault, 2001). The quantized vector is then simply reconstructed as $y = c$.

[0077] If $n > 0$ (step 206), the Voronoi extension is used. The extension order and the scale factor are set to $r = n$ and $m = 2^r$ (step 210), respectively. The indices j and k are demultiplexed (step 212). The index j is decoded into c in the base codebook C (step 214), while k is decoded into v in the Voronoi codebook $V^{(r)}$ (step 216). The quantized vector is reconstructed in step 218 as

$$y = m\,c + v \qquad \text{(Eq. 11)}$$

[0078] It is to be noted that the extension method used in the illustrative embodiment of the present invention is required only if the nearest lattice point y to the vector x to be quantized lies outside the base codebook. Consequently this extension prevents saturation provided the memory (number of available bits) is sufficient. It is important to note that the extended codebook reaches further out in N-dimensional space, while having the same lattice granularity as the base codebook (see for instance Figure 5). However, more bits are required when the extension is used.

[0079] In some instances, the quantizer may run out of bits without being able to capture the source vector x. In other words, the number of bits available to quantize the source vector x may be smaller than the number of bits required for the codevector index i and the codebook number n. In this case, the quantization error is not constrained by the granular structure of the base codebook, but a large error may occur. This typically happens with a very large outlier.

[0080] Several strategies can be implemented to handle outliers, such as down scaling the source vector x prior to multi-rate quantization. The scaling factor applied on x can be varied in such a way that there is no bit budget overflow.

[0081] For arbitrary outliers x, the complexity of the extension as described previously is unbounded, because the extension always starts with $r = 0$ and increments r by 1 at each iteration, independently of x. However, in practice, the extension order r is limited because of the size allocated to integers on the implementation platform, e.g., 16 for 16-bit unsigned integers. This limitation relates to the maximum value of the components of the Voronoi index k.

[0082] It has been found preferable that all lattice points be entries of either the base codebook C or one of its extensions $C^{(r)}$ for $r = 1, 2, \ldots$ Otherwise, some of the lattice points are impossible to index. For example, a base codebook C designed by near-spherical shaping centered around the origin meets this condition. Also, most of the codebooks obtained by shaping (truncating) a lattice with a centered (convex) region will meet this condition.

[0083] Multi-rate lattice quantization encoding and decoding methods according to third embodiments of respectively the second and the third aspects of the present invention will now be described.

[0084] These third embodiments of the present invention are based on the RE_8 lattice discussed hereinabove.

[0085] The previous illustrative embodiments of multi-rate quantization encoding and decoding methods according to the invention were based on a single base codebook derived from a lattice Λ that was extended with a bit-rate granularity of 1 bit per dimension. In particular, the extension method used is adapted to extend several near-spherical base codebooks so as to obtain a rate granularity of 1/2 bit per dimension, 4 bits in dimension 8.

[0086] For clarity purposes, the key symbols related to the 8-dimensional multi-rate lattice quantization methods are gathered in Table 2.

Table 2. List of symbols related to the 8-dimensional multi-rate lattice quantization method in accordance with the third illustrative embodiment of the invention.

Symbol	Definition	Note
RE_8	Gosset lattice in dimension 8.	
x	Source vector in dimension 8.	
y	Closest lattice point to x in RE_8.	The index of y is i. The quantization device outputs y as the quantized vector provided the bit butdget is sufficient.
n	Codebook number, restricted to the set {0, 2, 3, 4, 5, ...}.	
Q_n	Lattice codebook in RE_8 of index n. There are two categories of codebooks Q_n: 1) the base codebooks Q_0, Q_2, Q_3 and Q_4, and 2) the extended codebooks Q_n for $n > 4$.	Q_n is indexed with $4n$ bits.
i	Index of the lattice pointy in a codebook Q_n. If the Voronoi extension is used ($n > 4$), the index i is amultiplex of the indices j and k. The index j corresponds to a lattice point c in Q_3 or Q_4, while the index k is a Voronoi index. For $n \leq 4$, i is an index in a base codebook.	The index i is represented with $4n$ bits.
a	Offset for Voronoi coding, an 8-dimensional vector.	a = [2 0 0 0 0 0 0 0]
r	Extension order, a non-negative integer.	
m	Scaling factor of the extension.	$m = 2^r$
c	Base codevector in RE_8.	
j	Index of the base codevector c.	

(continued)

Symbol	Definition	Note
v	Voronoi codevector in RE_8.	The index of v is k.
y_a	Absolute leader, $y_a = [y_{a1} \ldots y_{a8}]$.	
k_a	Identifier of the absolute leader y_a. The base codebooks are specified in terms of absolute leaders (see Table 3). For the absolute leader y_a, the identifier k_a can be computed as follows: • Compute $s = (y_{a1}^4 + \ldots + y_{a8}^4)/8$. • Find the entry s in Table 4 giving k_a.	$0 \leq k_a \leq$ the number of absolute leaders
k	Index of the Voronoi codevector v or Voronoi index.	See Eq. 10 for a general definition, where $\Lambda = RE_8$ and G_Λ as in Eq. 4.
s	Key used to compute the identifier k_a, $s = (y_{a1}^4 + \ldots + y_{a8}^4)/8$.	See Table 4.

[0087] As will be explained hereinbelow in more detail, the encoding method according to the third illustrative embodiment of the second aspect of the present invention includes taking an 8-dimensional source vector x as an input and outputting an index i and a codebook number n. The codebook number n identifies a specific RE_8 codebook denoted by Q_n, that is, each Q_n is a subset of the RE_8 lattice. The codebook bit rate of Q_n is $4n/8$ bits per dimension. The number of bits in the index i is thus $4n$. The decoder uses the same multi-rate codebooks Q_n as the encoder, and simply reconstructs the lattice point y from the index i and the codebook number n.

[0088] According to the third illustrative embodiment, n is allowed to be any non-negative integer except unity, taking its value in the set {0, 2, 3, 4, 5, 6, ...}. The case $n = 1$ is not advantageous since it corresponds to a bit allocation of 4 bits in dimension 8. Indeed, at such a low bit rate, lattice quantization is not very efficient, and it is usually better in the context of transform coding to use a noise fill technique instead.

[0089] According to this third illustrative embodiment, the multi-rate codebooks are divided into two categories:

[0090] Low-rate base codebooks Q_0, Q_2, Q_3 and Q_4, which are classical near-spherical lattice codebooks. In the case where the method is implemented in a device, these codebooks are made available using tables stored in a memory or hardcoded in the device. Note that in the illustrative embodiment, the codebooks Q_2 and Q_3 are embedded, that is, Q_2 is a subset of Q_3.

[0091] High-rate extended codebooks Q_n for $n > 4$ which are virtually constructed in the device by applying a Voronoi extension method according to the present invention alternately to Q_3 and Q_4 such that Q_5 is generated as the first-order extension of Q_3, Q_6 is generated as the first-order extension of Q_4, Q_7 is generated as the second-order extension of Q_3, etc. More generally as illustrated in Figure 17, the extended codebook $Q_{n'}$ for $n' = n + 2r > 4$ is generated as the rth order extension of Q_n such that $n = 3$ for odd n' and $n = 4$ for even n'.

[0092] The separation between low and high-rate extended codebooks at Q4 allows a compromise between quality (performance) and complexity. For example, setting, for example, the separation at Q5 would yield bigger indexing tables, while setting the separation at Q3 would cause degradation of the quality but reduction of the complexity.

[0093] Table 3 defines the mapping of absolute leaders for Q_0, Q_2, Q_3 and Q_4. This mapping allows specifying the base codebooks unambiguously. Note that the sign constraints associated to these leaders do not appear in the table, but it is believed to be within the reach of a person having ordinary skills in the art to find them from the properties of RE_8 (Lamblin, 1988).

Table 3. List of the absolute leaders in RE_8 defining the base codebooks Q_0, Q_2, Q_3 and Q_4 in accordance with the third illustrative embodiment of the second aspect of the present invention.

Absolute leader	Q_0	Q_2	Q_3	Q_4
0, 0, 0, 0, 0, 0, 0, 0	X			
1, 1, 1, 1, 1, 1, 1, 1		X	X	

(continued)

Absolute leader	Q_0	Q_2	Q_3	Q_4
2, 2, 0, 0, 0, 0, 0, 0		X	X	
2, 2, 2, 2, 0, 0, 0, 0			X	
3, 1, 1, 1, 1, 1, 1, 1			X	
4, 0, 0, 0, 0, 0, 0, 0		X	X	
2, 2, 2, 2, 2, 2, 0, 0				X
3, 3, 1, 1, 1, 1, 1, 1				X
4, 2, 2, 0, 0, 0, 0, 0			X	
2, 2, 2, 2, 2, 2, 2, 2				X
3, 3, 3, 1, 1, 1, 1, 1				X
4, 2, 2, 2, 2, 0, 0, 0				X
4, 4, 0, 0, 0, 0, 0, 0			X	
5, 1, 1, 1, 1, 1, 1, 1				X
3, 3, 3, 3, 1, 1, 1, 1				X
4, 2, 2, 2, 2, 2, 2, 0				X
4, 4, 2, 2, 0, 0, 0, 0				X
5, 3, 1, 1, 1, 1, 1, 1				X
6, 2, 0, 0, 0, 0, 0, 0			X	
4, 4, 4, 0, 0, 0, 0, 0				X
6, 2, 2, 2, 0, 0, 0, 0				X
6, 4, 2, 0, 0, 0, 0, 0				X
7, 1, 1, 1, 1, 1, 1, 1				X
8, 0, 0, 0, 0, 0, 0, 0				X
6, 6, 0, 0, 0, 0, 0, 0				X
8, 2, 2, 0, 0, 0, 0, 0				X
8, 4, 0, 0, 0, 0, 0, 0				X
9, 1, 1, 1, 1, 1, 1, 1				X
10, 2, 0, 0, 0, 0, 0, 0				X
8, 8, 0, 0, 0, 0, 0, 0				X
10, 6, 0, 0, 0, 0, 0, 0				X
12, 0, 0, 0, 0, 0, 0, 0				X
12, 4, 0, 0, 0, 0, 0, 0				X
10, 10, 0, 0, 0, 0, 0, 0				X
14, 2, 0, 0, 0, 0, 0, 0				X
12, 8, 0, 0, 0, 0, 0, 0				X
16, 0, 0, 0, 0, 0, 0, 0				X

[0094] Furthermore, the 8-dimensional offset a defining the Voronoi shaping is set as a = [2 0 0 0 0 0 0 0].

[0095] The multi-rate lattice encoding method 300 in accordance with the second embodiment of the second aspect of the invention will now be described in more detail in reference to Figure 18.

[0096] Providing x be a source vector to be quantized in dimension 8. With step 302, the method first proceeds by finding the nearest neighbor y of the 8-dimensional input x in the infinite lattice RE_8. Then, in step 304, verification is made as to if y is an entry of a base codebook Q_0, Q_2, Q_3, or Q_4. This verification outputs a codebook number n and an identified k_a if $n > 0$. The details of this verification will be provided hereinbelow. The codebook number n at this stage is taken in the set {0, 2, 3, 4, *out*}. The value *out* is an integer set arbitrarily to *out* = 100. The value n = *out* is used to indicate that an outlier has been detected, meaning that the lattice point y is not an entry in any of the base codebooks. There are then two cases:

- If $n \leq 4$ (step 304), the encoding is completed in step 306. If $n = 0$ (step 306), y is the zero vector and encoding is terminated. If n = 2, 3, or 4, then extra Information k_a identifies one of the absolute leaders defining Q_n. The vector y is indexed in Q_n given k_a as will be explained hereinbelow in more detail.

- If n = *out* (step 304), a Voronoi extension method according to the present invention is applied. The first step (308) of the extension method is to initialize the parameters of the extention, and then to iterate the Voronoi extension algorithm, as described hereinbelow, until y is included in an extended codebook.

[0097] In the second embodiment of the multi-rate lattice vector encoding and decoding methods, the Voronoi extension was iterated by incrementing the extension order r until y is reached. This may yield an unbounded complexity. In order to constrain the worst-case complexity, it has been found advantageous to use a maximum of two iterations for the Voronoi extension.

[0098] In the initialization step of the extension method, the extension order r and the extension scaling factor m are set to some initial values given the lattice vector $y = [y_1 ... y_8]$ so as to minimize the number of iterations over r. The iteration counter *iter* is set to zero.

[0099] The pre-selection of r and m is implemented as follows. First, the floating-point value $\sigma = (y_1^2 + ... + y_8^2)/32$ is computed, $r = 1$ is set and $m = 2^r = 2$. Then while $\sigma > 11$, iterate σ, r, and m by updating $\sigma := \sigma/4$, $r := r + 1$ and $m := 2m$. The rationale of this procedure is justified by the following two observations:

- When moving from the rth order extension to the $(r + 1)$th extension, the base codebooks Q_3 and Q_4 are multiplied by $m = 2^{r+1}$ instead of $m = 2^r$. The squared norm of the scaled base codevectors is multiplied by 4 after the $(r + 1)$ th extension compared to the rth extension. This explains the factor 1/4 applied to σ after each iteration over r.

- The union of the base codebooks $Q_0 \cup Q_2 \cup Q_3 \cup Q_4$ comprises lattice points on the RE_8 shells 0 to 32. It can be verified that the outmost complete shell in $Q_0 \cup Q_2 \cup Q_3 \cup Q_4$ is the shell 5. The constant 11 in the termination condition $\sigma > 11$ has been selected experimentally between 5 and 32.

[0100] An alternative to this initialization procedure consists of computing directly how many times σ goes into 11 and then set r and m accordingly. The result will be naturally identical to the iterative procedure described previously.

[0101] If *iter* = 2 (step 310), the method exits the loop comprising steps 310-326.

[0102] In step 312, the Voronoi index k of the lattice point y is computed given m using modular arithmetic. The Voronoi index k depends on the extension order r and the scaling factor m. In the particular case of the RE_8 lattice, the Voronoi index is calculated as follows:

$$k = \mathrm{mod}_m(y G_{RE_8}^{-1})$$

where G_{RE_8} is the generator matrix defined in Eq. 4, and $\mathrm{mod}_m(\cdot)$ is the component wise modulo-m operation. Hence, the Voronoi index k is a vector of 8 integers, each component being between 0 and m - 1. By construction k requires thus a total of $8r$ bits for indexing all components.

[0103] The Voronoi codevector v is computed from the Voronoi index k given m (step 314). The algorithm described in (Conway, 1983) can be used for this purpose.

[0104] The difference vector $w = y - v$ is computed (step 316). This difference vector w is a point in the scaled lattice mRE_8. $c = w/m$ is then computed (step 316), that is, the inverse scaling is applied to the difference vector w. The point c belongs to the lattice RE_8 since w belongs to mRE_8.

[0105] The extension method proceeds with the verification as to if c is an entry of the base codebooks Q_2, Q_3 or Q_4 (step 318). This verification outputs a codebook number n and also an identifier k_a. The details of the verification will be explained hereinbelow in more detail. With the disclosed multi-rate base codebooks, c cannot be in Q_0 at this stage. As a result, the actual value of n may be either 2, 3, 4, or *out*.

[0106] If $n = out$, the extension order r is incremented by one and the extension scale m is multiplied by 2 (step 320).

[0107] If c is an entry of Q_2, it is also an entry of Q_3 because Q_2 is embedded in Q_3 ($Q_2 \subset Q_3$). Therefore if $n = 2$, n is set to $n = 3$ (step 322). An extension order rand scaling factor $m = 2^r$ sufficiently large to quantize the source vector x without saturation occurs when $n = 3$ or $n = 4$. The codebook number n is updated to incorporate the extension order r. This is achieved by adding $2r$ to n (step 322). c, k_a, n and k are then saved into a memory (step 324) and the extension order r is decremented by one and the scaling factor m is divided by 2 (step 326).

[0108] The incrementation of the iteration counter *iter* is incremented by one (step 328), before the beginning of a new iteration (step 310).

[0109] When the loop is terminated after two iterations, the parameters characterizing the extension are retrieved from the memory (step 330) that contains the following values:

- The vector c computed in step 316 which is an entry of Q_3 or Q_4;
- The Voronoi index k computed in step 312. It is to be noted that the Voronoi index k is an 8-dimensional vector, where each component is an integer between 0 and m - 1 and can be indexed by r bits;
- The identifier k_a computed as a side product of step 318; and
- The codebook number n incorporating the extension order r as computed in step 320.

[0110] The index j of c is computed in step 332. This index is multiplexed with the Voronoi index k to form the index i (step 334). To complete the encoding, the codebook number n is subjected to lossless encoding as will be described later and multiplexed with the index i for storage or transmission over a channel.

[0111] It is to be noted that the encoding method 300 assumes that sufficient memory is available for characterizing the source vector x with a attice point y. Therefore, the multi-rate quantization encoding method 300 is preferably applied in a gain-shape fashion by down-scaling the source vector x as x/g prior to applying the method 300. The scalar parameter $g > 1$ is determined to avoid an overrun in memory, and quantized separately using prior-art means. However, if the overrun of the memory occurs when failing to choose g properly, by default n is set to zero and the reconstruction y becomes a zero vector. The selection and quantization technique of g is not detailed here since it depends on the actual application.

[0112] Furthermore, assuming the method 300 is implemented on a software platform where the components of Voronoi index k are represented with 16-bit integers, the maximum value of r is $r = 16$ resulting in the maximum value $n = 4 + 2{\times}16 = 36$ for the codebook number.

[0113] The methods for finding the identifier of an absolute leader in base codebooks, and for verifying if a lattice point is in a base codebook (steps 304 and 318) will now be explained in more detail.

[0114] According to such a method from the prior art, to verify if a lattice point y is an entry in a near-spherical lattice codebook Q_2, Q_3 or Q_4, the absolute values of the components of y are reordered in descending order and compared directly component by component with the absolute leaders defining the codebooks (see for example Lamblin, 1988). The method according to the present invention is based on the use of the identifier k_a. This method is described below in three steps:

1) Computing the value s from y. Writing y component wise as $y = [y_1 \ldots y_8]$, this value is computed as

$$s = (y_1^4 + \ldots + y_8^4)/8 \qquad \text{(Eq. 12)}$$

2) The values of s computed for the absolute leaders y defining the base codebooks differ all from each other. Moreover, all valid signed permutations of y will result in the same value s. As a consequence, the value s is referred to here as a *key*, because it identifies uniquely an absolute leader and all the related signed permutations.

3) Setting k_a to value 36 if $s = 0$, y is a zero vector. Otherwise looking-up for the key s in a mapping table translating s into the identifier k_a that may attain integer values between 0 and 37. Table 4 gives this mapping that can readily be computed from Table 3. If the key s is an entry of Table 4, the identifier k_a is an integer between 0 and 36 (see Table 4). Otherwise, y is declared an outlier by setting k_a to 37.

Table 4. List of identifiers k_a for the absolute leaders in RE_8 defining the base codebooks Q_2, Q_3 and Q_4.

Identifier k_a	Key in hexadecimal S	Absolute leader
0	0001	1, 1, 1, 1, 1, 1, 1, 1
1	0004	2, 2, 0, 0, 0, 0, 0, 0

(continued)

Identifier k_a	Key in hexadecimal S	Absolute leader
2	0008	2, 2, 2, 2, 0, 0, 0, 0
3	000B	3, 1, 1, 1, 1, 1, 1, 1
4	0020	4, 0, 0, 0, 0, 0, 0, 0
5	000C	2, 2, 2, 2, 2, 2, 0, 0
6	0015	3, 3, 1, 1, 1, 1, 1, 1
7	0024	4, 2, 2, 0, 0, 0, 0, 0
8	0010	2, 2, 2, 2, 2, 2, 2, 2
9	001F	3, 3, 3, 1, 1, 1, 1, 1
10	0028	4, 2, 2, 2, 2, 0, 0, 0
11	0040	4, 4, 0, 0, 0, 0, 0, 0
12	004F	5, 1, 1, 1, 1, 1, 1, 1
13	0029	3, 3, 3, 3, 1, 1, 1, 1
14	002C	4, 2, 2, 2, 2, 2, 2, 0
15	0044	4, 4, 2, 2, 0, 0, 0, 0
16	0059	5, 3, 1, 1, 1, 1, 1, 1
17	00A4	6, 2, 0, 0, 0, 0, 0, 0
18	0060	4, 4, 4, 0, 0, 0, 0, 0
19	00A8	6, 2, 2, 2, 0, 0, 0, 0
20	00C4	6, 4, 2, 0, 0, 0, 0, 0
21	012D	7, 1, 1, 1, 1, 1, 1, 1
22	0200	8, 0, 0, 0, 0, 0, 0, 0
23	0144	6, 6, 0, 0, 0, 0, 0, 0
24	0204	8, 2, 2, 0, 0, 0, 0, 0
25	0220	8, 4, 0, 0, 0, 0, 0, 0
26	0335	9, 1, 1, 1, 1, 1, 1, 1
27	04E4	10, 2, 0, 0, 0, 0, 0, 0
28	0400	8, 8, 0, 0, 0, 0, 0, 0
29	0584	10, 6, 0, 0, 0, 0, 0, 0
30	0A20	12, 0, 0, 0, 0, 0, 0, 0
31	0A40	12, 4, 0, 0, 0, 0, 0, 0
32	09C4	10, 10, 0, 0, 0, 0, 0 ,0
33	12C4	14, 2, 0, 0, 0, 0, 0, 0
34	0C20	12, 8, 0, 0, 0, 0, 0, 0
35	2000	16, 0, 0, 0, 0, 0, 0, 0

[0115] It is to be noted that the last column on the right in table 4 is only informative and defines without ambiguity the mapping between the identifier k_a and the related absolute leader.

[0116] At this stage, the identifier verifies $0 \leq k_a \leq 37$. Indeed, if $0 \leq k_a < 36$, y is either in Q_2, Q_3 or Q_4. If $k_a = 36$, y is in Q_0. If $k_a = 37$, y is not in any of the base codebooks. The identifier k_a is then mapped to the codebook number n with Table 5.

Table 5. Mapping table to translate the identifier k_a of absolute leaders into a base codebook number n in accordance with the present invention.

Identifier k_a	**Codebook number n**	**Identifier k_a**	**Codebook number n**
0	2	19	4
1	2	20	4
2	3	21	4
3	3	22	3
4	2	23	4
5	4	24	4
6	4	25	4
7	3	26	4
8	4	27	4
9	4	28	4
10	4	29	4
11	3	30	4
12	4	31	4
13	4	32	4
14	4	33	4
15	4	34	4
16	4	35	4
17	3	36	0
18	4	37	*out* = 100

[0117] The method for indexing in base codebooks Q_2, Q_3, and Q_4 based on the identifier of the absolute leader (steps 306 and 332) will now be described in more detail.

[0118] The index of an entry y in the base codebooks Q_2, Q_3, and Q_4 is computed using a prior-art indexing technique for near-spherical codebooks as described in Lamblin (1988). To be more precise, the index of y, for example j, is computed as follows: j = *cardinality offset* + *rank* of *permutation*. The *rank* of *permutation* is computed according to Schalkwijk's formula, which is well known in the art and described in (Lamblin, 1988). The *cardinality offset* is found by table look-up after computing the signed leader for y. This table look-up is based on the identifier k_a of the related absolute leader.

[0119] According to the illustrative embodiments of the invention, the codebook number n is encoded using a variable-length binary code, well-known in the prior art as a *unary code*, based on the following mapping:

$Q_0 \rightarrow 0$
$Q_2 \rightarrow 10$
$Q_3 \rightarrow 110$
$Q_4 \rightarrow 1110$
$Q_5 \rightarrow 11110$
...

[0120] The right hand side of the above mapping gives n_E in binary representation to be multiplexed with the codevector index i. Further, when the Voronoi extension is used (when $n > 4$), the codevector index i comprises two sub-indices multiplexed together:

- the base codebook index j of 12 bits for odd n and 16 bits for even n; and
- the Voronoi index k of $8r$ bits, comprising eight integers of r bits as its components.

[0121] The structure of i is illustrated in Figure 19(a) for $2 \leq n \leq 4$, while Figure 19(b) considers the case $n > 4$. It is to be note that the multiplexed bits may be permuted in any fashion within the whole block of 4n bits.

[0122] The multi-rate lattice decoding method 400 in accordance with the second embodiment of the third aspect of the invention will now be described in more detail in reference to Figure 20.

[0123] The coded codebook number n_E is first read from the channel and decoded to get the codebook number n (step 402). The codebook number n is retrieved from n_E by reversing the mapping described in the method 300. Then the decoder interprets the codevector index i differently depending on the codebook number n.

[0124] If $n = 0$ (step 404), y is reconstructed as a zero vector that is the only entry of the base codebook Q_0 (step 406).

[0125] If $0 < n \leq 4$ (step 404), the codevector index i of size $4n$ bits from the channel is demultiplexed (step 408). Then i is decoded as an index of a base codebook Q_2, Q_3, or Q_4 and y is reconstructed using prior-art techniques (step 410) such as those described in Lamblin (1988), Moureaux (1998) and Rault (2001).

[0126] A value $n > 4$ in step 404 indicates that the Voronoi extension is employed. The codevector index i of size $4n$ bits from the channel (step 408) is demultiplexed as the base codebook index j and the Voronoi index k from i (step 412). The extension order r is also retrieved and so is scaling factor m from n (step 414). The value of r is obtained as a quotient of $(n - 3)/2$, and $m = 2^r$. Then $2r$ is substracted from n to identify either Q_3 or Q_4, and the index j is decoded into c using the subtracted value of n (step 416). The decoding of j is based on a prior-art method comprising of rank decoding and table look-ups as described in Lamblin (1988), Moureaux (1998) and Rault (2001). The Voronoi index k is decoded into v (step 418) based on m using the prior-art algorithm described in (Conway, 1983). Finally, in step 420, the decoder computes the reconstruction as $y = m\,c + v$, where $m = 2^r$ is the extension scaling factor, c is a codevector of the base codebook and v is a codevector of the Voronoi codebook. When $r = 0$, no extension is used, and v is a zero vector and m becomes one.

[0127] Although the present invention has been described hereinabove by way of illustrative embodiments thereof, it can be modified without departing from the scope of the subject invention, as defined in the appended claims.

Claims

1. A method for encoding a source signal, for transmission or storage using multi-rate lattice quantization, the method comprising the steps of:

   i) providing a source vector x representing a frame of examples of the source signal;
   ii) providing a base codebook C derived from a lattice Λ of points;
   iii) associating to the source vector x a lattice point y in said lattice Λ; **characterized by**

   - if the lattice point y is included in the base codebook C, performing the following step iv):

   iv) indexing the lattice point y in the base codebook C yielding quantization indices, and ending the method; and

   - if the lattice point y is not included in the base codebook C, performing the following steps v), vi) and vii):

   v) extending the base codebook C, yielding an extended codebook;
   vi) associating to the lattice point y a codevector c from the extended codebook; and
   vii) indexing the lattice point y in the extended codebook yielding quantization indices;

   wherein the quantization indices form a quantized representation of the source vector x.

2. A method as recited in claim 1, wherein the extended codebook is represented by the expression $mC + V$, wherein m is a scaling factor, C is the base codebook and V is a set of points in the lattice Λ.

3. A method as recited in claim 1, wherein step iii) comprises selecting the lattice point y in the lattice Λ as a nearest neighbour of the source vector x in the lattice Λ.

4. A method as recited in claim 1, wherein step v) includes providing an integer seating factor $m \geq 2$; step vi) includes computing a Voronoi codevector v corresponding to the lattice point y using the scaling factor m; and step vi) further includes computing the codevector c using the Voronoi codevector v and the scaling factor m.

5. A method as recited in claim 4, wherein step v) comprises setting the scaling factor m to 2^r, with r being an extension order; step v) further includes computing a Voronoi index k; and step vi) includes computing the Voronoi codevector

v corresponding to the lattice point y using the Voronoi index k and the scaling factor m.

6. A method as recited in claim 4, wherein step vi) comprises computing the codevector c as $c = (y - v) / m$.

7. A method as recited in claim 4, wherein step vi) further includes verifying if the codevector c is in the base codebook C,

- if the codevector c is in the base codebook C, performing the following step a):

  a) in step vii) indexing the lattice point y as a base codevector and multiplexing j and k yielding quantization indices, where j is an index of the codevector c in the base codebook C and k is a Voronoi index corresponding to the Voronoi codevector v;

- if the codevector c is not in the base codebook C, performing the following step b):

  b) increasing the scaling factor m and an order of a Voronoi extension and repeating steps v) to vi).

8. A method as recited in claim 4, further comprising in step vii) defining a lossless encoding of a codebook number corresponding to an extension order r and an index i of the lattice point y in the base codebook C, yielding an encoded codebook number n_E and an encoded index i_E; and multiplexing the encoded codebook number n_E and the encoded index i_E.

9. A method as recited in claim 1, further comprising a step viii) of storing the quantization indices in storing means.

10. A method as recited in claim 1, further comprising a step viii) of transmitting the quantization indices over a communication channel.

11. A method for encoding a source signal using multi-rate lattice quantization as defined in claim 1, wherein:

step v) comprises providing a subset V of points of the lattice Λ;
step vii) comprises indexing the lattice point y into an integer codebook number n and an index i as $y = mc+v$, wherein c is an element of the base codebook C, v is an element of the subset V, and m is an integer greater than or equal to two;

wherein the base codebook C and subset V of the lattice Λ, the integer m and the index i are defined from the codebook number n; and
wherein n and i are the quantization indices forming a quantized representation of the source vector x.

12. A method as recited in claim 11, wherein the codebook number n is represented by a unary code.

13. A method as recited in claim 11, wherein the subset V of the lattice Λ is a Voronoi codebook and an index of the element v is a Voronoi index.

14. A method as recited in claim 11, wherein the index i is a concatenation of an index of the element c and an index of the element v.

15. A method as recited in claim 11, wherein the lattice point $y = [0 \dots 0]$ and the integer codebook number n is set to a predetermined value when a number of allocated bits available in implementing the method is not sufficient to represent the source vector x in the lattice Λ.

16. A method as recited in claim 11, wherein the integer $m = 2^r$, with r being an integer greater than or equal to 1; the base codebook C being predetermined; and the codebook number n being equal to r plus a predetermined integer.

17. A method for encoding a source signal using multi-rate lattice quantization as defined in claim 1, wherein:

step i) comprises providing an 8-dimension source vector x representing a frame of the source signal;
step ii) comprises providing low-rate lattice base codebooks Q_0, Q_2, Q_3 and Q_4 derived from a lattice of points RE_8;
step iii) comprises determining a lattice point y in the lattice RE_8 which is a nearest neighbour of the source

vector x in the lattice RE_8;

- if the lattice point y is included in the low-rate lattice base codebook Q_n, where n equals 0, 2, 3 or 4, step iv) comprises:

a) memorizing the number n and an identifier k_a identifying one of absolute leaders defining the base codebook Q_n,
b) if $n > 0$, performing the following step b1:

b1) indexing the lattice point y in the base codebook Q_n yielding quantization indices,
if a scaling factor $m = 0$, performing the following step b2:
b2) the lattice point y is outputted as a zero vector, and

c) ending the method;

- if the lattice point y is not included in the low-rate lattice base codebook Q_n, where n equals 0, 2, 3 or 4:

step v) comprises the steps of viii) providing an extension order r; ix) setting the scaling factor m to 2^r; x) setting an iteration number $iter = 0$; xi) computing a Voronoi index k of the lattice point y; xii) computing a Voronoi codevector v corresponding to the lattice point y using the Voronoi index k and the scaling factor m;
step vi) comprises the step of xiii) computing the codevector c as $c = (y - v) / m$; and
step vii) comprises the steps of:

xiv) - if the codevector c is included in the base codebook Q_n, wherein n equals 2, 3 or 4, performing the following step aa):

aa) providing the number n and the identifier k_a; if a codebook number n is equal to 2 ($n = 2$), setting $n = 3$; incrementing the codebook number n by $2r$; storing the values of the Voronoi index k, the codevector c, the number n and the identifier k_a; dividing the scaling factor m by 2; decreasing the extension order r by 1;

- if the codevector c is not included in the base codebook Q_n, wherein n equals 2, 3 or 4, performing the following step bb):

bb) multiplying the scaling factor m by 2; increasing r by 1;

xv) increasing the iteration number $iter$ by 1;
xvi) - if said iteration number $iter = 2$ then performing the following step aaa):

aaa) retrieving the values of the Voronoi index k, the codevector c, the number n and the identifier k_a; indexing the codevector c in the base codebook Q_3 or Q_4, given k_a; multiplexing j and k to form index i, where i is the index of the lattice point y and j is the index of the codevector c;

- if said iteration number $iter \neq 2$ then performing the following step bbb):

bbb) repeating steps xi) to xvi);

wherein steps iv) and xvi) yield the quantization indices forming a quantized representation of the source vector x.

18. A method as recited in claim 17, wherein in xi) the Voronoi index k is computed as being:

$$k = \mathrm{mod}_m(y G_{RE_8}^{-1})$$

where G_{RE_8} is a generator matrix defined as

$$G_{RE_8} = \begin{bmatrix} 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \end{bmatrix}$$

and $mod_m(\cdot)$ is a component wise modulo-m operation.

19. A method as recited in claim 17, further comprising down-scaling the source vector x as x/g prior to step iii); g being chosen so as to be greater than 1 and as to avoid an overrun in a memory used in implementing the method.

20. A method for decoding a transmitted or stored source signal using multi-rate lattice quantization, the method comprising the steps of:

i) providing a base codebook C derived from a lattice Λ;
ii) providing a codebook number n and a quantization index i; **characterized by**
iii) demultiplexing the quantization index i using the codebook number n;
iv) if $n = 0$ then decoding the quantization index i using the base codebook C, yielding a quantized codevector y, and ending the method;
v) if $n > 0$ then

a) providing a preselected Voronoi codebook $V^{(r)}$;
b) setting an extension order to $r = n$ and a scaling factor $m = 2^r$;
c) demultiplexing indices j and k from the quantization index i;
d) decoding j into a codevector c in the base codebook C;
e) decoding k into a codevector v in the Voronoi codebook $V^{(r)}$; and
f) reconstructing a quantized codevector as

$$y = m\,c + v$$

wherein the reconstructed codevector y represents a quantized representation of a frame of samples the source signal.

21. A method as recited in claim 20, wherein in step ii) an encoded codebook number n_E and an encoded index i_E are first provided; then a predetermined lossless coding technique is applied on the encoded codebook number n_E and the encoded index i_E to get the codebook number n and the quantization index i, respectively.

22. A method as recited in claim 20, wherein the codebook number n and the quantization index i are read from a communication channel.

23. A method as recited in claim 20, wherein the codebook number n and the quantization index i are read from storing means.

24. A method for decoding a source signal using multi-rate lattice quantization as defined in claim 20, wherein:

the lattice Λ is a lattice of points
the base codebook C is a first subset of the lattice Λ of points;
the Voronoi codebook $V^{(r)}$ is a second subset of the lattice Λ of points;

reconstructing the codevector y as $y = mc + v$ comprises reconstructing the point c as an element of the subset C of the lattice Λ, lattice L, reconstructing the codevector v as an element of the subset V of the lattice Λ; wherein the source signal decoding method comprises defining the subsets C and V of the lattice Λ, the value of the integer m and the size of the index i from the codebook number n.

25. A method as recited in claim 24, wherein the codebook number n is reconstructed from a unary code.

26. A method as recited in claim 24, wherein the subset V of the lattice Λ is a Voronoi code and the index of the codevector v is a Voronoi index.

27. A method for decoding a source signal using multi-rate lattice quantization as defined in claim 20, wherein:

step i) comprises providing low-rate lattice base codebooks Q_0, Q_2, Q_3, and Q_4 derived from a lattice RE_8;

if $n = 0$, step iv) is replaced by a step of vi) reconstructing a codevector y as a zero vector;

step v) comprises the steps of:

if $0 < n \leq 4$, decoding the codevector index i as an index of the base codebook Q_2, Q_3 or Q_4 and reconstructing the codevector y;
if $n > 4$, using the codebook number n to identify either base codebook Q_3 or Q_4, and performing steps a)-f) using the selected base codebook Q_3 or Q_4 as base codebook C.

28. A method as defined in claim 1, wherein the base codebook C defines a subset of codevectors, and wherein step v) comprises the steps of:

scaling the codevectors of the base codebook C by a predetermined scale factor, yielding scaled codevectors; and
iii) inserting a Voronoi codebook around each scaled codevector, yielding the extended codebook.

29. A method as recited in claim 28, comprising the step of selecting the subset of codevectors by retaining codevectors within a boundary.

30. A method as recited in claim 28, wherein the step of scaling the codevectors comprises providing an integer scaling factor $m \geq 2$.

31. A method as recited in claim 28, wherein, in the step of scaling the codevectors, the scale factor m is set to 2^r, where r is an extension order.

32. A method as recited in claim 29, wherein the boundary is provided by a shape selected from the group consisting of a sphere, a square, a pyramid and a rectangle.

33. A method for encoding a source signal using multi-rate lattice quantization according to claim 1, wherein:

step ii) comprises providing the base codebook C formed of a subset of codevectors from the lattice Λ having a dimension N, wherein the base codebook C requires NR bits for indexing, where R is a bit rate per dimension of said base codebook C;
step iii) comprises determining a codevector from the lattice Λ nearest to the source vector x;

- if the nearest codevector is comprised within the base codebook C, step iv) comprises:

indexing the nearest codevector in the base codebook yielding quantization indices, outputting the quantization indices, and ending the method;

- if the nearest codevector is not comprised within the base codebook C, step v) comprises the following steps viii) to x):

viii) providing a predetermined scale factor;

ix) scaling the base codebook C by the scale factor, yielding a scaled codebook including scaled codevectors;
x) inserting a Voronoi codebook around each scaled codevector, yielding an extended codebook requiring $N(R+r)$ bits for indexing, where r is the order of the Voronoi codebook:
if the nearest codevector is comprised within the extended codebook then step vii) comprises indexing the nearest codevector in the extended codebook yielding quantization indices, outputting the quantization indices, and ending the method;
if the nearest codevector is not comprised within the extended codebook and if the number of bits required for indexing the extended codebook has not exceeded a predetermined threshold then repeating steps viii) to x) for increasing the scale factor and the order of the Voronoi codebook;
if the number of bits required for indexing the extended codebook has exceeded a predetermined threshold, performing a step of xi) considering the nearest codevector as a remote outlier and outputting predetermined corresponding quantization indices.

34. A method as recited in claim 33, wherein the predetermined scale factor is 2 and the scale factor is increased by 2.

35. A method as recited in claim 33, wherein the order of the Voronoi codebook is increased by 1.

36. A signal encoder for encoding a source signal for transmission or storage using multi-rate lattice quantization, the encoder comprising:

receiving means for providing a source vector x representing a frame of samples of the source signal;
memory means including a base codebook C derived from a lattice Λ of points;
means for associating to the source vector x a lattice point y in the lattice Λ; **characterized by**;
means for verifying if the lattice point y is included in the base codebook C;
if the lattice point y is included in the base codebook C, means for indexing the lattice point y in the base codebook C yielding quantization indices;
if the lattice point y is not included in the base codebook C, means for extending the base codebook and for yielding an extended codebook; means for associating to the lattice point y a codevector c from the extended codebook; and means for indexing the lattice point y in the extended codebook and for yielding quantization indices;
wherein the quantization indices form a quantized representation of the source vector x.

37. A signal encoder for encoding a source signal using multi-rate lattice quantization as defined in claim 36, wherein:

the means for extending the base codebook C comprises means for providing a subset V of points of the lattice Λ;
the means for indexing the lattice point y in the extended codebook comprises means for indexing the lattice point y into an integer codebook number n and an index i as $y = mc + v$, wherein c is an element of the base codebook C, v is an element of the subset V, and m is an integer greater than or equal to 2;
the signal encoder comprises means for defining the base codebook C, the subset V of the lattice Λ, the integer m and the index i from the codebook number n.

38. A signal decoder for decoding a transmitted or stored encoded source signal using multi-rate lattice quantization, the decoder comprising:

memory means for providing a base codebook C derived from a lattice Λ;
receiving means for providing an encoded codebook number n and an encoded quantization index i; **characterized by**
means for demultiplexing the quantization index i using the codebook number n;
means for verifying if the codebook number $n = 0$;
if $n = 0$, means for decoding the quantization index i using the base codebook C, yielding a quantized codevector y;
means for verifying if $n > 0$;
if $n > 0$, means for providing a preselected Voronoi codebook $V^{(r)}$; means for setting an extension order to $r = n$ and a scaling factor to $m = 2^r$; means for demultiplexing indices j and k from the quantization index i; means for decoding j so as to yield a codevector c from the base codebook C; means for decoding the index k so as to yield a codevector v from the Voronoi codebook $V^{(r)}$; and means for reconstructing a quantized codevector as $y = m\,c + v$;
wherein the reconstructed codevector y represents a quantized representation of a frame of samples of the

source signal.

39. A signal decoder using multi-rate lattice quantization as defined in claim 38, wherein:

the means for providing a preselected Voronoi codebook $V^{(r)}$ comprises means for providing a subset V of the lattice Λ;
the means for reconstructing the quantized codevector y as $y = mc + v$ comprises means for reconstructing the lattice point c as an element of the base codebook C and means for reconstructing the lattice point v as an element of the subset V of the lattice Λ;
wherein the signal decoder comprises means for defining the base codebook C and the subset V of the lattice Λ, the integer m and the index i from the codebook number n.

Patentansprüche

1. Verfahren zum Codieren eines Quellsignals zur Übertragung oder Speicherung unter Verwendung einer Mehrfachraten-Gitterquantisierung, wobei das Verfahren die folgenden Schritte umfasst:

i) Vorsehen eines Quellvektors x, der einen Rahmen von Abtastwerten des Quellsignals repräsentiert;
ii) Vorsehen eines Basiscodebuchs C, das aus einem Gitter Λ von Punkten abgeleitet ist;
iii) Zuordnen eines Gitterpunkts y in dem Gitter Λ zu dem Quellvektor x;
gekennzeichnet durch

- falls der Gitterpunkt y in dem Basiscodebuch C enthalten ist, Ausführen des folgenden Schrittes iv):

iv) Indexieren des Gitterpunkts y in dem Basiscodebuch C, was Quantisierungsindizes ergibt, und Beenden des Verfahrens; und

- falls der Gitterpunkt y nicht in dem Basiscodebuch C enthalten ist, Ausführen der folgenden Schritte v), vi) und vii);

v) Erweitern des Basiscodebuchs C, was ein erweitertes Codebuch ergibt;
vi) Zuordnen eines Codevektors c aus dem erweiterten Codebuch zu dem Gitterpunkt y; und
vii) Indexieren des Gitterpunkts y in dem erweiterten Codebuch, was Quantisierungsindizes ergibt;

wobei die Quantisierungsindizes eine quantisierte Darstellung des Quellvektors x bilden.

2. Verfahren nach Anspruch 1, wobei das erweiterte Codebuch durch den Ausdruck mC + V repräsentiert wird, wobei m ein Skalierungsfaktor ist, C das Basiscodebuch ist und V eine Menge von Punkten in dem Gitter Λ ist.

3. Verfahren nach Anspruch 1, wobei der Schritt iii) das Auswählen des Gitterpunkts y in dem Gitter Λ als einen nächsten Nachbarn des Quellvektors x in dem Gitter Λ umfasst.

4. Verfahren nach Anspruch 1, wobei der Schritt v) das Vorsehen eines ganzzahligen Skalierungsfaktors $m \geq 2$ umfasst; der Schritt vi) das Berechnen eines Voronoi-Codevektors v, der dem Gitterpunkt y entspricht, unter Verwendung des Skalierungsfaktors m umfasst; und der Schritt vi) ferner das Berechnen des Codevektors c unter Verwendung des Voronoi-Codevektors v und des Skalierungsfaktors m umfasst.

5. Verfahren nach Anspruch 4, wobei der Schritt v) das Setzen des Skalierungsfaktors m auf 2^r umfasst, wobei r eine Erweiterungsordnung ist; der Schritt v) ferner das Berechnen eines Voronoi-Indexes k umfasst; und der Schritt vi) das Berechnen des Voronoi-Codevektors v, der dem Gitterpunkt y entspricht, unter Verwendung des Voronoi-Indexes k und des Skalierungsfaktors m umfasst.

6. Verfahren nach Anspruch 4, wobei der Schritt vi) das Berechnen des Codevektors c gemäß $c = (y - v)/m$ umfasst.

7. Verfahren nach Anspruch 4, wobei der Schritt vi) ferner das Verifizieren, ob der Codevektor c im Basiscodebuch C enthalten ist, umfasst,

- falls der Codevektor c im Basiscodebuch C enthalten ist, Ausführen des folgenden Schrittes a):

  a) im Schritt vii) Indexieren des Gitterpunkts y als einen Basiscodevektor und Multiplexieren von j und k, was Quantisierungsindizes ergibt, wobei j ein Index des Codevektors c im Basiscodebuch C ist und k ein Voronoi-Index ist, der dem Voronoi-Codevektor v entspricht;

- falls der Codevektor c nicht im Basiscodebuch C enthalten ist, Ausführen des folgenden Schrittes b):

  b) Erhöhen des Skalierungsfaktors m und einer Ordnung einer Voronoi-Erweiterung und Wiederholen der Schritte v) bis vi).

8. Verfahren nach Anspruch 4, das ferner im Schritt vii) das Definieren einer verlustlosen Codierung einer Codebuchnummer, die einer Erweiterungsordnung r entspricht, und eines Indexes i des Gitterpunkts y im Basiscodebuch C, was eine codierte Codebuchnummer n_E bzw. einen codierten Index i_E ergibt, und das Multiplexieren der codierten Codebuchnummer n_E und des codierten Indexes i_E umfasst.

9. Verfahren nach Anspruch 1, das ferner einen Schritt viii) des Speicherns der Quantisierungsindizes in Speichermitteln umfasst.

10. Verfahren nach Anspruch 1, das ferner einen Schritt viii) des Übertragens der Quantisierungsindizes über einen Kommunikationskanal umfasst.

11. Verfahren zum Codieren eines Quellsignals unter Verwendung einer Mehrfachraten-Gitterquantisierung nach Anspruch 1, wobei:

der Schritt v) das Vorsehen einer Untermenge V von Punkten des Gitters Λ umfasst;
der Schritt vii) das Indexieren des Gitterpunkts y in eine ganzzahlige Codebuchnummer n und einen Index i gemäß y = mc + v umfasst, wobei c ein Element des Basiscodebuchs C ist, v ein Element der Untermenge V ist und m eine ganze Zahl größer oder gleich Zwei ist;
wobei das Basiscodebuch C und die Untermenge V des Gitters A, die ganze Zahl m und der Index i aus der Codebuchnummer n definiert sind; und
wobei n und i die Quantisierungsindizes sind, die eine quantisierte Darstellung des Quellvektors x bilden.

12. Verfahren nach Anspruch 11, wobei die Codebuchnummer n durch einen unären Code repräsentiert wird.

13. Verfahren nach Anspruch 11, wobei die Untermenge V des Gitters Λ ein Voronoi-Codebuch ist und ein Index des Elements v ein Voronoi-Index ist.

14. Verfahren nach Anspruch 11, wobei der Index i eine Verknüpfung aus einem Index des Elements c und aus einem Index des Elements v ist.

15. Verfahren nach Anspruch 11, wobei der Gitterpunkt y = [0 ... 0] und die ganzzahlige Codebuchnummer n auf einen vorgegebenen Wert gesetzt werden, wenn eine Anzahl zugewiesener Bits, die bei der Implementierung des Verfahrens verfügbar sind, nicht ausreicht, um den Quellvektor x im Gitter Λ zu repräsentieren.

16. Verfahren nach Anspruch 11, wobei die ganze Zahl $m = 2^r$ ist, wobei r eine ganze Zahl größer oder gleich 1 ist; das Basiscodebuch C vorgegeben ist; und die Codebuchnummer n gleich r plus einer vorgegebenen ganzen Zahl ist.

17. Verfahren zum Codieren eines Quellsignals unter Verwendung einer Mehrfachraten-Gitterquantisierung nach Anspruch 1, wobei:

der Schritt i) das Vorsehen eines achtdimensionalen Quellvektors x, der einen Rahmen des Quellsignals repräsentiert, umfasst;
der Schritt ii) das Vorsehen von Niedrigraten-Gitter-Basiscodebüchern Q_0, Q_2, Q_3 und Q_4, die aus einem Gitter von Punkten RE_8 abgeleitet werden, umfasst;
der Schritt iii) das Bestimmen eines Gitterpunkts y im Gitter RE_8, der ein nächster Nachbar des Quellvektors x im Gitter RE_8 ist, umfasst;

- falls der Gitterpunkt y in dem Niedrigraten-Gitter-Basiscodebuch Q_n enthalten ist, wobei n gleich 0, 2, 3 oder 4 ist, der Schritt iv) umfasst:

a) Speichern der Anzahl n und eines Identifizierers k_a, der einen der absoluten Führer identifiziert, die das Basiscodebuch Q_n definieren,
b) falls n > 0, Ausführen des folgenden Schrittes b1:

b1) Indexieren des Gitterpunkts y in dem Basiscodebuch Q_n, was Quantisierungsindizes ergibt,

- falls ein Skalierungsfaktor m = 0, Ausführen des folgenden Schrittes b2:

b2) der Gitterpunkt y wird als ein Nullvektor ausgegeben, und
c) Beenden des Verfahrens;

- falls der Gitterpunkt y nicht in dem Niedrigraten-Gitter-Basiscodebuch Q_n enthalten ist, wobei n gleich 0, 2, 3 oder 4:

der Schritt v) die folgenden Schritte umfasst: viii) Vorsehe n einer Erweiterungsordnung r; ix) Setzen des Skalierungsfaktors m auf 2^r; x) Setzen einer Iterationsanzahl iter = 0; xi) Berechnen eines Voronoi-Indexes k des Gitterpunkts y; und xii) Berechnen eines Voronoi-Codevektors v, der dem Gitterpunkt y entspricht, unter Verwendung des Voronoi-Indexes k und des Skalierungsfaktors m;
der Schritt vi) den Schritt xiii) des Berechnens des Codevektors c als c = (y - v)/m umfasst; und
der Schritt vii) die folgenden Schritte umfasst:

xiv) - falls der Codevektor c im Basiscodebuch Q_n enthalten ist, wobei n gleich 2, 3 oder 4 ist, Ausführen des folgenden Schrittes aa):

aa) Vorsehen der Anzahl n und des Identifizierers k_a; falls eine Codebuchnummer n gleich 2 ist (n = 2), Setzen von n = 3; Inkrementieren der Codebuchnummer n um 2r; Speichern der Werte des Voronoi-Indexes k, des Codevektors c, der Anzahl n und des Identifizierers k_a; Dividieren des Skalierungsfaktors m durch 2; Erniedrigen der Erweiterungsordnung r um 1;

- falls der Codevektor c nicht in dem Basiscodebuch Q_n enthalten ist, wobei n gleich 2, 3 oder 4 ist, Ausführen des folgenden Schrittes bb):

bb) Multiplizieren des Skalierungsfaktors m mit 2; Erhöhen von r um 1;

xv) Erhöhen der Iterationsanzahl iter um 1;
xvi) - falls die Iterationsanzahl iter = 2, Ausführen des folgenden Schrittes aaa):

aaa) Wiedergewinnen der Werte des Voronoi-Indexes k, des Codevektors c, der Anzahl n und des Identifizierers k_a; Indexieren des Codevektors c in dem Basiscodebuch Q_3 oder Q_4, wenn k_a gegeben ist; Multiplizieren von j und k, um einen Index i zu bilden, wobei i der Index des Gitterpunkts y ist und j der Index des Codevektors c ist;

- falls die Iterationsanzahl iter ≠ 2, Ausführen des folgenden Schrittes bbb):

bbb) Wiederholen der Schritte xi) bis xvi);

wobei die Schritte iv) und xvi) die Quantisierungsindizes ergeben, die eine quantisierte Darstellung des Quellvektors x bilden.

18. Verfahren nach Anspruch 17, wobei in xi) der Voronoi-Index k folgendermaßen berechnet wird:

$$k = \mathrm{mod}_m\left(yG_{RE_8}^{-1}\right)$$

wobei G_{RE_8} eine Generatormatrix ist, die folgendermaßen definiert ist:

$$G_{RE_8} = \begin{bmatrix} 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \end{bmatrix}$$

und $mod_m(\cdot)$ eine komponentenartige modulo-m-Operation ist.

19. Verfahren nach Anspruch 17, das ferner vor dem Schritt iii) das Abwärtsskalieren des Quellvektors x gemäß x/g umfasst; wobei g so gewählt ist, dass es größer als 1 ist und ein Überlauf in einem Speicher, der bei der Implementierung des Verfahrens verwendet wird, vermieden wird.

20. Verfahren zum Decodieren eines übertragenen oder gespeicherten Quellsignals unter Verwendung einer Mehrfachraten-Gitterquantisierung, wobei das Verfahren die folgenden Schritte umfasst:

i) Vorsehen eines Basiscodebuchs C, das aus einem Gitter Λ abgeleitet wird;
ii) Vorsehen einer Codebuchnummer n und eines Quantisierungsindexes i;
gekennzeichnet durch
iii) Demultiplexieren des Quantisierungsindexes i unter Verwendung der Codebuchnummer n;
iv) falls n = 0, Decodieren des Quantisierungsindexes i unter Verwendung des Basiscodebuchs C, was einen quantisierten Codevektor y ergibt, und Beenden des Verfahrens;
v) falls n > 0, dann

a) Vorsehen eines im Voraus gewählten Voronoi-Codebuchs $V^{(r)}$;
b) Setzen einer Erweiterungsordnung auf r = n und eines Skalierungsfaktors $m = 2^r$;
c) Demultiplexieren von Indizes j und k aus dem Quantisierungsindex i;
d) Decodieren von j in einen Codevektor c im Basiscodebuch C;
e) Decodieren von k in einen Codevektor v in dem Voronoi-Codebuch $V^{(r)}$; und
f) Rekonstruieren eines quantisierten Codevektors gemäß

$$y = mc + v$$

wobei der rekonstruierte Codevektor y eine quantisierte Darstellung eines Rahmens von Abtastwerten des Quellsignals repräsentiert.

21. Verfahren nach Anspruch 20, wobei im Schritt ii) als Erstes eine codierte Codebuchnummer n_E und ein codierter Index i_E vorgesehen werden; dann eine vorgegebene verlustfreie Codierungstechnik auf die codierte Codebuchnummer n_E und auf den codierten Index i_E angewendet wird, um die Codebuchnummer n bzw. den Quantisierungsindex i zu erhalten.

22. Verfahren nach Anspruch 20, wobei die Codebuchnummer n und der Quantisierungsindex i aus einem Kommunikationskanal gelesen werden.

23. Verfahren nach Anspruch 20, wobei die Codebuchnummer n und der Quantisierungsindex i aus Speichermitteln gelesen werden.

24. Verfahren zum Decodieren eines Quellsignals unter Verwendung einer Mehrfachraten-Gitterquantisierung nach Anspruch 20, wobei:

das Gitter Λ ein Gitter von Punkten ist,
das Basiscodebuch C eine erste Untermenge des Gitters Λ von Punkten ist;
das Voronoi-Codebuch $V^{(r)}$ eine zweite Untermenge des Gitters Λ von Punkten ist;
das Rekonstruieren des Codevektors y gemäß y = mc + v das Rekonstruieren des Punkts c als ein Element der Untermenge C des Gitters Λ und das Rekonstruieren des Codevektors v als ein Element der Untermenge V des Gitters Λ umfasst;
wobei das Quellsignal-Decodierungsverfahren das Definieren der Untermengen C und V des Gitters Λ, des Wertes der ganzen Zahl m und der Größe des Indexes i aus der Codebuchnummer n umfasst.

25. Verfahren nach Anspruch 24, wobei die Codebuchnummer n aus einem unären Code rekonstruiert wird.

26. Verfahren nach Anspruch 24, wobei die Untermenge V des Gitters Λ ein Voronoi-Code ist und der Index des Codevektors v ein Voronoi-Index ist.

27. Verfahren zum Decodieren eines Quellsignals unter Verwendung einer Mehrfachraten-Gitterquantisierung nach Anspruch 20, wobei:

der Schritt i) das Vorsehen von Niedrigraten-Gitter-Basiscodebüchern Q_0, Q_2, Q_3 und Q_4, die aus einem Gitter RE_8 abgeleitet werden, umfasst;
falls n = 0, der Schritt iv) durch einen Schritt vi) ersetzt wird, indem ein Codevektor y als ein Nullvektor rekonstruiert wird;
der Schritt v) die folgenden Schritte umfasst:
falls $0 < n \leq 4$, der Codevektorindex i als ein Index des Basiscodebuchs Q_2, Q_3 oder Q_4 decodiert wird und der Codevektor y rekonstruiert wird;
falls n > 4, die Codebuchnummer n verwendet wird, um entweder das Basiscodebuch Q_3 oder Q_4 zu identifizieren, und die Schritte a)-f) unter Verwendung des ausgewählten Basiscodebuchs Q_3 oder Q_4 als Basiscodebuch C ausgeführt werden.

28. Verfahren nach Anspruch 1, wobei das Basiscodebuch C eine Untermenge von Codevektoren definiert und wobei der Schritt v) die folgenden Schritte umfasst:

Skalieren der Codevektoren des Basiscodebuchs C mit einem vorgegebenen Skalierungsfaktor, was skalierte Codevektoren ergibt; und
iii) Einsetzen eines Voronoi-Codebuchs um jeden skalierten Codevektor, was das erweiterte Codebuch ergibt.

29. Verfahren nach Anspruch 28, das den Schritt des Auswählens der Untermenge von Codevektoren durch Halten von Codevektoren innerhalb einer Grenze umfasst.

30. Verfahren nach Anspruch 28, wobei der Schritt des Skalierens der Codevektoren das Vorsehen eines ganzzahligen Skalierungsfaktors $m \geq 2$ umfasst.

31. Verfahren nach Anspruch 28, wobei im Schritt des Skalierens der Codevektoren der Skalierungsfaktor m auf 2^r gesetzt wird, wobei r eine Erweiterungsordnung ist.

32. Verfahren nach Anspruch 29, wobei die Grenze durch eine Form geschaffen wird, die aus der Gruppe gewählt ist, die aus einer Kugel, einem Quadrat, einer Pyramide und einem Rechteck besteht.

33. Verfahren zum Codieren eines Quellsignals unter Verwendung einer Mehrfachraten-Gitterquantisierung nach Anspruch 1, wobei:

der Schritt ii) das Vorsehen des Basiscodebuchs C, das aus einer Untermenge von Codevektoren aus dem Gitter Λ mit Dimension N gebildet ist, umfasst,
wobei das Basiscodebuch C zum Indexieren NR Bits erfordert, wobei R eine Bitrate pro Dimension des Basis-

codebuchs C ist;
der Schritt iii) das Bestimmen eines Codevektors aus dem Gitter Λ, der sich am nächsten bei dem Quellvektor x befindet, umfasst;

- falls der nächste Codevektor in dem Basiscodebuch C enthalten ist, der Schritt iv) umfasst:

  Indexieren des nächsten Codevektors in dem Basiscodebuch, was Quantisierungsindizes ergibt, Ausgeben der Quantisierungsindizes und Beenden des Verfahrens;

- falls der nächste Codevektor nicht in dem Basiscodebuch C enthalten ist, der Schritt v) die folgenden Schritte viii) bis x) umfasst:

  viii) Vorsehen eines vorgegebenen Skalierungsfaktors;
  ix) Skalieren des Basiscodebuchs C mit dem Skalierungsfaktor, was ein skaliertes Codebuch ergibt, das skalierte Codevektoren enthält;
  x) Einsetzen eines Voronoi-Codebuchs um jeden skalierten Codevektor, was ein erweitertes Codebuch ergibt, wobei zum Indexieren N(R + r) Bits erforderlich sind, wobei r die Ordnung des Voronoi-Codebuchs ist;

falls der nächste Codevektor in dem erweiterten Codebuch enthalten ist, der Schritt vii) das Indexieren des nächsten Codevektors in dem erweiterten Codebuch, was Quantisierungsindizes ergibt, das Ausgeben der Quantisierungsindizes und das Beenden des Verfahrens umfasst;
falls der nächste Codevektor nicht in dem erweiterten Codebuch enthalten ist und falls die Anzahl von Bits, die zum Indexieren des erweiterten Codebuchs erforderlich ist, einen vorgegebenen Schwellenwert nicht überschritten hat, Wiederholen der Schritte viii) bis x), um den Skalierungsfaktor und die Ordnung des Voronoi-Codebuchs zu erhöhen;
falls die Anzahl von Bits, die zum Indexieren des erweiterten Codebuchs erforderlich ist, einen vorgegebenen Schwellenwert überschritten hat, Ausführen eines Schrittes xi), in dem der nächste Codevektor als ein entfernter Ausreißer angesehen wird und vorgegebene entsprechende Quantisierungsindizes ausgegeben werden.

34. Verfahren nach Anspruch 33, wobei der vorgegebene Skalierungsfaktor gleich 2 ist und der Skalierungsfaktor um 2 erhöht wird.

35. Verfahren nach Anspruch 33, wobei die Ordnung des Voronoi-Codebuchs um 1 erhöht wird.

36. Signalcodierer zum Codieren eines Quellsignals zur Übertragung oder Speicherung unter Verwendung einer Mehrfachraten-Gitterquantisierung, wobei der Codierer umfasst:

Empfangsmittel, um einen Quellvektor x vorzusehen, der einen Rahmen von Abtastwerten des Quellsignals repräsentiert;
Speichermittel, die ein Basiscodebuch C enthalten, das aus einem Gitter Λ von Punkten abgeleitet ist;
Mittel, um dem Quellvektor x einen Gitterpunkt y in dem Gitter Λ zuzuordnen;
gekennzeichnet durch
Mittel, um zu verifizieren, ob der Gitterpunkt y in dem Basiscodebuch C enthalten ist;
falls der Gitterpunkt y in dem Basiscodebuch C enthalten ist, Mittel, um den Gitterpunkt y in dem Basiscodebuch C zu indexieren, was Quantisierungsindizes ergibt;
falls der Gitterpunkt y nicht in dem Basiscodebuch C enthalten ist, Mittel, um das Basiscodebuch zu erweitern und um ein erweitertes Codebuch zu liefern; Mittel, um dem Gitterpunkt y einen Codevektor c aus dem erweiterten Codebuch zuzuordnen; und Mittel, um den Gitterpunkt y in dem erweiterten Codebuch zu indexieren und um Quantisierungsindizes zu liefern;
wobei die Quantisierungsindizes eine quantisierte Darstellung des Quellvektors x bilden.

37. Signalcodierer zum Codieren eines Quellsignals unter Verwendung einer Mehrfachraten-Gitterquantisierung nach Anspruch 36, wobei:

die Mittel zum Erweitern des Basiscodebuchs C Mittel umfassen, um eine Untermenge V von Punkten des Gitters Λ vorzusehen;

die Mittel zum Indexieren des Gitterpunkts y in dem erweiterten Codebuch Mittel umfassen, um den Gitterpunkt y in eine ganzzahlige Codebuchnummer n und in einen Index i gemäß $y = mc + v$ zu indexieren, wobei c ein Element des Basiscodebuchs C ist, v ein Element der Untermenge V ist und m eine ganze Zahl größer oder gleich 2 ist;
der Signalcodierer Mittel umfasst, um das Basiscodebuch C, die Untermenge V des Gitters A, die ganze Zahl m und den Index i aus der Codebuchnummer n zu definieren.

38. Signaldecodierer zum Decodieren eines übertragenen oder gespeicherten codierten Quellsignals unter Verwendung einer Mehrfachraten-Gitterquantisierung, wobei der Decodierer umfasst:

Speichermittel, um ein Basiscodebuch C vorzusehen, das aus einem Gitter Λ abgeleitet ist;
Empfangsmittel, um eine codierte Codebuchnummer n und einen codierten Quantisierungsindex i vorzusehen;
gekennzeichnet durch
Mittel, um den Quantisierungsindex i unter Verwendung der Codebuchnummer n zu demultiplexieren;
Mittel, um zu verifizieren, ob die Codebuchnummer $n = 0$;
falls $n = 0$, Mittel, um den Quantisierungsindex i unter Verwendung des Basiscodebuchs C zu decodieren, was einen quantisierten Codevektor y ergibt;
Mittel, um zu verifizieren, ob $n > 0$;
falls $n > 0$, Mittel, um ein im Voraus gewähltes Voronoi-Codebuch $V^{(r)}$ vorzusehen; Mittel, um eine Erweiterungsordnung auf $r = n$ zu setzen und um einen Skalierungsfaktor auf $m = 2^r$ zu setzen; Mittel, um Indizes j und k aus dem Quantisierungsindex i zu demultiplexieren; Mittel, um j zu decodieren, um einen Codevektor c aus dem Basiscodebuch C zu liefern; Mittel, um den Index k zu decodieren, um einen Codevektor v aus dem Voronoi-Codebuch $V^{(r)}$ zu liefern; und Mittel, um einen quantisierten Codevektor gemäß $y = mc + v$ zu rekonstruieren;
wobei der rekonstruierte Codevektor y eine quantisierte Darstellung eines Rahmens von Abtastwerten des Quellsignals repräsentiert.

39. Signaldecodierer, der eine Mehrfachraten-Gitterquantisierung verwendet, nach Anspruch 38, wobei:

die Mittel zum Vorsehen eines im Voraus gewählten Voronoi-Codebuchs $V^{(r)}$ Mittel umfassen, um eine Untermenge V des Gitters Λ zu schaffen;
die Mittel zum Rekonstruieren des quantisierten Codevektors y gemäß $y = mc + v$ Mittel, um den Gitterpunkt c als ein Element des Basiscodebuchs C zu rekonstruieren, und Mittel, um den Gitterpunkt v als ein Element der Untermenge V des Gitters Λ zu rekonstruieren, umfassen;
wobei der Signaldecodierer Mittel umfasst, um das Basiscodebuch C und die Untermenge V des Gitters Λ, die ganze Zahl m und den Index i aus der Codebuchnummer n zu definieren.

Revendications

1. Procédé de codage d'un signal source destiné à la transmission ou au stockage en utilisant une quantification multidébit en treillis, le procédé comprenant les étapes consistant en :

i) la fourniture d'un vecteur source x représentant une trame d'échantillons du signal source ;
ii) la fourniture d'un guide de codification de base C dérivé d'un treillis Λ de points ;
iii) l'association au vecteur source x d'un point de treillis y dans ledit treillis Λ ; **caractérisé en ce que :**

- si le point de treillis y est inclus dans le guide de codification de base C, la réalisation de l'étape suivante iv) :

iv) l'indexation du point de treillis y dans le guide de codification de base C donnant des indices de quantification et l'arrêt du procédé ; et

- si le point de treillis y n'est pas inclus dans le guide de codification de base C, la réalisation des étapes suivantes v), vi) et vii) :

v) l'extension du guide de codification de base C, donnant un guide de codification étendu ;
vi) l'association au point de treillis y d'un vecteur code c à partir du guide de codification étendu ; et
vii) l'indexation du point de treillis y dans le guide de codification étendu donnant des indices de quantification ;

dans lequel les indices de quantification forment une représentation quantifiée du vecteur source x.

2. Procédé tel que revendiqué dans la revendication 1, dans lequel le guide de codification étendu est représenté par l'expression mC + V dans laquelle m est un facteur d'échelle, C est le guide de codification de base et V est un ensemble de points dans le treillis Λ.

3. Procédé tel que revendiqué dans la revendication 1, dans lequel l'étape iii) comprend la sélection d'un point de treillis y dans le treillis Λ comme voisin le plus proche du vecteur source x dans le treillis Λ.

4. Procédé tel que revendiqué dans la revendication 1, dans lequel l'étape v) inclut la fourniture d'un facteur d'échelle en nombre entier $m \geq 2$; l'étape vi) inclut le calcul d'un vecteur code v de Voronoï correspondant au point de treillis y en utilisant le facteur d'échelle m ; et l'étape vi) inclut en outre le calcul du vecteur code c en utilisant le vecteur code v de Voronoï et le facteur d'échelle m.

5. Procédé tel que revendiqué dans la revendication 4, dans laquelle l'étape v) comprend la définition du facteur d'échelle m sur 2^r, r étant un ordre d'extension ; l'étape v) inclut en outre le calcul d'un indice de Voronoï k ; et l'étape vi) inclut le calcul du vecteur code de Voronoï *v* correspondant au point de treillis *y* en utilisant l'indice de Voronoï *k* et le facteur d'échelle *m*.

6. Procédé tel que revendiqué dans la revendication 4, dans lequel l'étape vi) comprend le calcul du vecteur code *c* sous la forme de $c = (y - v)/m$.

7. Procédé tel que revendiqué dans la revendication 4, dans lequel l'étape vi) inclut en outre la vérification que le vecteur code *c* se trouve dans le guide de codification de base *C*,

- si le vecteur code *c* se trouve dans le guide de codification de base *C*, l'exécution de l'étape suivante a) :

  a) dans l'étape vii), l'indexation du point de treillis *y* en tant que vecteur code de base et le multiplexage de *j* et *k* donnant des indices de quantification, où *j* est un indice du vecteur code *c* dans le guide de codification de base *C*, et *k* est un indice de Voronoï correspondant au vecteur code de Voronoï *v* ;

- si le vecteur code *c* ne se trouve pas dans le guide de codification de base *C*, l'exécution de l'étape suivante b) :

  b) l'augmentation du facteur d'échelle *m* et d'un ordre de l'extension de Voronoï et la répétition des étapes v) à vi).

8. Procédé tel que revendiqué dans la revendication 4, comprenant en outre à l'étape vii) la définition d'un codage sans perte d'un nombre de guide de codification correspondant à un ordre d'extension *r* et un indice *i* du point de treillis *y* dans le guide de codification de base *C*, donnant un nombre de guide de codification codé n_E et un indice codé i_E ; et le multiplexage du nombre de guide de codification codé n_E et de l'indice codé i_E.

9. Procédé tel que revendiqué dans la revendication 1, comprenant en outre une étape viii) consistant à stocker les indices de quantification dans les moyens de stockage.

10. Procédé tel que revendiqué dans la revendication 1, comprenant en outre l'étape viii) de transmission des indices de quantification sur un canal de communication.

11. Procédé de codage d'un signal source en utilisant une quantification multi-débit en treillis, tel que défini dans la revendication 1, dans lequel :

   l'étape v) comprend la fourniture d'un sous-ensemble *V* de points du treillis Λ ;
   l'étape vii) comprend l'indexation du point de treillis *y* dans un nombre entier de guide de codification *n* et un indice *i* tel que $y = mc + v$, dans lequel *c* est un élément du guide de codification de base *C*, *v* est un élément du sous-ensemble *V* et *m* est un nombre entier supérieur ou égal à deux ;
   dans lequel le guide de codification de base *C* et le sous-ensemble *V* du treillis Λ, le nombre entier *m* et l'indice *i* sont définis à partir du nombre de guide de codification *n* ; et
   dans lequel *n* et *i* sont des indices de quantification formant une représentation quantifiée du vecteur source *x*.

12. Procédé tel que revendiqué dans la revendication 11, dans lequel le nombre du guide de codification dans lequel le nombre du guide de codification n est représenté par un code monadique.

13. Procédé tel que revendiqué dans la revendication 11, dans lequel le sous-ensemble V du treillis Λ est un guide de codification de Voronoï et un indice de l'élément v est un indice de Voronoï.

14. Procédé tel que revendiqué dans la revendication 11, dans lequel l'indice i est une concaténation d'un indice de l'élément c et un indice de l'élément v.

15. Procédé tel que revendiqué dans la revendication 11, dans lequel le point de treillis $y = [0...0]$ et le nombre entier du guide de codification n est défini selon une valeur prédéterminée lorsqu'un nombre de bits alloués disponibles dans la mise en oeuvre du procédé n'est pas suffisant pour représenter le vecteur source x dans le treillis Λ.

16. Procédé tel que revendiqué dans la revendication 11, dans lequel le nombre entier $m = 2^r$, avec r qui est un nombre entier supérieur ou égal à 1 ; le guide de codification de base C étant prédéterminé ; et le nombre de guide de codification n étant égal à r plus un nombre entier prédéterminé.

17. Procédé de codage d'un signal source utilisant une codification multi-débit en treillis, tel que défini dans la revendication 1, dans lequel :

l'étape i) comprend la fourniture d'un vecteur source à 8 dimensions x représentant une trame du signal source ;
l'étape ii) comprend la fourniture de guides de codification de base à bas débit en treillis Q_0, Q_2, Q_3 et Q_4 dérivés d'un treillis de points RE_8 ;
l'étape iii) comprend la détermination d'un point de treillis y dans le treillis RE_8 qui est un voisin le plus proche du vecteur source x dans le treillis RE_8 ;

- si le point de treillis y est inclus dans le guide de codification de base à bas débit en treillis Q_n, où n est égal à 0, 2, 3 ou 4, l'étape iv) comprend :

a) la mémorisation du nombre n et d'un identifiant k_a identifiant l'un des leaders absolus définissant le guide de codification de base Q_n
b) si $n > 0$, l'exécution de l'étape suivante b1 :

b1) l'indexation du point de treillis y dans le guide de codification de base Q_n donnant des indices de quantification,
si un facteur d'échelle $m = 0$, l'exécution de l'étape suivante b2 :
b2) le point de treillis y est sorti comme vecteur zéro, et

c) l'arrêt du procédé ;

- si le point de treillis y n'est pas inclus dans le guide de codification de base à faible débit en treillis Q_n, où n est égal à 0, 2, 3 ou 4 :

l'étape v) comprend les étapes de viii) la fourniture d'un ordre d'extension r, ix) la définition du facteur d'échelle m sur 2^r ; x) la définition d'un nombre d'itération $iter = 0$; xi) le calcul d'un indice de Voronoï k du point de treillis y ; xii) le calcul d'un vecteur code de Voronoï v correspondant au point de treillis y en utilisant l'indice de Voronoï k et le facteur d'échelle m ;
l'étape vi) comprend l'étape du xiii) calcul d'un vecteur code c tel que $c = (y - v) / m$; et
l'étape vii) comprend les étapes consistant en :

xiv)- si le vecteur code c est inclus dans le guide de codification de base Q_n, dans lequel n est égal à 2, 3 ou 4, l'exécution de l'étape suivante aa) :

aa) la fourniture du nombre n et de l'identifiant k_a; si un nombre n du guide de codification est égal à 2 ($n = 2$), la définition de $n = 3$; l'incrémentation du nombre du guide de codification n de $2r$, le stockage des valeurs de l'indice de Voronoï k, du vecteur code c, du nombre n et de l'identifiant k_a ; la division du facteur d'échelle m par 2 ; la diminution de l'ordre d'extension r de 1 ;.

- si le vecteur code c n'est pas inclus dans le guide de codification de base Q_n, dans lequel n est égal à 2, 3 ou 4, l'exécution de l'étape suivante bb):

bb) la multiplication du facteur d'échelle m par 2 ; l'augmentation de r de 1 ;

xv) l'augmentation du nombre d'itération iter de 1 ;
xvi)- si ledit nombre d'itération $iter = 2$, alors l'exécution de l'étape suivante aaa) :

aaa) l'extraction des valeurs de l'indice de Voronoï k, du vecteur code c, du nombre n et de l'identifiant k_a ; l'indexation du vecteur code c dans le guide de codification de base Q_3 ou Q_4, étant donné k_a ;

le multiplexage de j et k pour former l'indice i, où i est l'indice du point de treillis y et j est l'indice du vecteur code c ;
- si ledit nombre d'itération $iter \neq 2$, alors la réalisation de l'étape suivante bbb) :

bbb) la répétition des étapes xi) à xvi) ;

dans lesquelles les étapes iv) et xvi) donnent les indices de quantification formant une représentation quantifiée du vecteur source x.

18. Procédé tel que revendiqué dans la revendication 17, dans lequel dans xi) l'indice de Voronoï k est calculé comme étant :

$$k = \mathrm{mod}_m(y G_{RE_8}^{-1})$$

où G_{RE} est une matrice de générateur définie sous la forme de

$$G_{RE_8} = \begin{bmatrix} 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \end{bmatrix}$$

et $mod_m(\cdot)$ est une opération modulo-m composant par composant.

19. Procédé tel que revendiqué dans la revendication 17, comprenant en outre la rétrogradation du vecteur source x en tant que x/g avant l'étape iii) ; g étant choisi de façon à être supérieur à 1 et de façon à éviter un écrasement dans une mémoire utilisée lors de la mise en oeuvre du procédé.

20. Procédé de décodage d'un signal source transmis ou stocké en utilisant une quantification multi-débit en treillis, le procédé comprenant les étapes consistant en :

i) la fourniture d'un guide de codification de base C dérivé d'un treillis Λ ;
ii) la fourniture d'un nombre de guide de codification n et d'un indice de quantification i ; **caractérisé par**
iii) le démultiplexage de l'indice de quantification i en utilisant le nombre du guide de codification n ;

iv) si $n = 0$ alors le décodage de l'indice de quantification i en utilisant le guide de codification de base C, donnant un vecteur code quantifié y, et l'arrêt du procédé ;
v) si $n > 0$ alors

a) la fourniture d'un guide de codification de Voronoï présélectionné $V^{(r)}$;
b) la définition d'un ordre d'extension sur $r = n$ et d'un facteur d'échelle $m = 2^r$;
c) le démultiplexage des indices j et k à partir de l'indice de quantification i ;
d) le décodage de j en un vecteur code c dans le guide de codification de base C ;
e) le décodage de k en un vecteur code v dans le guide de codification de Voronoï $V^{(r)}$; et
f) la reconstruction d'un vecteur code quantifié tel que

$$y = mc + v$$

dans lequel le vecteur code reconstruit y représente une représentation quantifiée d'une trame d'échantillons du signal source.

21. Procédé tel que revendiqué dans la revendication 20, dans lequel à l'étape ii) un nombre de guide de codification codé n_E et un indice codé i_E sont tout d'abord fournis ; puis une technique de codage sans perte prédéterminée est appliquée sur le nombre du guide de codification codé n_E et l'indice codé i_E pour obtenir le nombre du guide de codification n et l'indice de quantification i, respectivement.

22. Procédé tel que revendiqué dans la revendication 20, dans lequel le nombre du guide de codification n et l'indice de quantification i sont lus dans un canal de communication.

23. Procédé tel que revendiqué dans la revendication 20, dans lequel le nombre du guide de codification n et l'indice de quantification i sont lus dans les moyens de stockage.

24. Procédé de décodage d'un signal source utilisant une quantification multi-débit en treillis, tel que défini dans la revendication 20, dans lequel :

le treillis Λ est un treillis de points
le guide de codification de base C est un premier sous-ensemble du treillis Λ de points ;
le guide de codification de Voronoï $V^{(r)}$ est un second sous-ensemble du treillis Λ de points ;
la reconstruction du vecteur code y, tel que $y = mc + v$ comprend la reconstruction du point c comme élément du sous-ensemble C du treillis Λ, le treillis L, la reconstruction du vecteur code v comme élément du sous-ensemble V du treillis Λ ;
dans lequel le procédé de décodage du signal source comprend la définition des sous-ensembles C et V du treillis Λ, la valeur du nombre entier m et la taille de l'indice i provenant du nombre du guide de codification n.

25. Procédé tel que revendiqué dans la revendication 24, dans lequel le nombre du guide de codification n est reconstitué à partir d'un code monadique.

26. Procédé tel que revendiqué dans la revendication 24, dans lequel le sous-ensemble V du treillis Λ est un code de Voronoï et l'indice du vecteur code v est un indice Voronoï.

27. Procédé de décodage d'un signal source utilisant une quantification multi-débit en treillis, tel que défini dans la revendication 20, dans lequel :

l'étape i) comprend la fourniture de codes de quantification de base à faible débit en treillis Q_0, Q_2, Q_3, et Q_4 dérivés d'un treillis RE_8 ;
si $n = 0$, l'étape iv) est remplacée par une étape de vi) reconstruction d'un vecteur code y comme vecteur zéro ;
l'étape v) comprend les étapes consistant en :

si $0 < n \leq 4$, le décodage de l'indice du vecteur code i en tant qu'indice du guide de codification de base Q_2, Q_3, ou Q_4 et la reconstitution du vecteur code y ;
si $n > 4$, l'utilisation du nombre du guide de codification n pour identifier soit le guide de codification de

base Q_3 ou Q_4, et la réalisation des étapes a) à f) en utilisant le guide de codification de base sélectionné Q_3 ou Q_4 comme guide de codification de base C.

28. Procédé tel que défini dans la revendication 1, dans lequel le guide de codification de base C définit un sous-ensemble de vecteurs codes, et dans lequel l'étape v) comprend les étapes consistant en :

La mise à l'échelle des vecteurs codes du guide de codification de base C grâce à un facteur d'échelle prédéterminé, donnant des vecteurs codes à l'échelle ; et

iii) l'insertion d'un guide de codification de Voronoï autour de chaque vecteur code mis à l'échelle, donnant le guide de codification étendu.

29. Procédé tel que revendiqué dans la revendication 28, comprenant l'étape de sélection du sous-ensemble de vecteurs codes en retenant les vecteurs codes à l'intérieur d'une limite.

30. Procédé tel que revendiqué dans la revendication 28, dans lequel l'étape de mise à l'échelle des vecteurs codes comprend la fourniture d'un facteur d'échelle de nombre entier $m \geq 2$.

31. Procédé tel que revendiqué dans la revendication 28, dans lequel, dans l'étape de mise à l'échelle des vecteurs codes, le facteur d'échelle m est défini sur 2^r, où r est un ordre d'extension.

32. Procédé tel que revendiqué dans la revendication 29, dans lequel la limite est fournie par une forme sélectionnée dans le groupe consistant en une sphère, un carré, une pyramide et un rectangle.

33. Procédé de codage d'un signal source utilisant une quantification multi-débit en treillis selon la revendication 1, dans lequel :

l'étape ii) comprend la fourniture du guide de codification de base C formé d'un sous-ensemble de vecteurs codes issus du treillis Λ ayant une dimension N, dans lequel le guide de codification de base C nécessite NR bits pour l'indexation, où R est un débit binaire par dimension dudit guide de codification de base C ;
l'étape iii) comprend la détermination d'un vecteur code à partir du treillis Λ le plus proche du vecteur source x ;

- si le vecteur code le plus proche est compris dans les limites du guide de codification de base C, l'étape iv) comprend :

l'indexation du vecteur code le plus proche dans le guide de codification de base donnant des indices de quantification, la sortie des indices de quantification et l'arrêt du procédé ;

- si le vecteur code le plus proche n'est pas compris dans les limites du guide de codification de base C, l'étape v) comprend les étapes suivantes viii) consistant en x) :

viii) la fourniture d'un facteur d'échelle prédéterminé ;
ix) la mise à l'échelle du guide de codification C par le facteur d'échelle, donnant un guide de codification mis à l'échelle incluant des vecteurs codes mis à l'échelle ;
x) l'insertion d'un guide de codification de Voronoï autour de chaque vecteur code mis à l'échelle, donnant un guide de codification étendu nécessitant $N(R+r)$ bits pour l'indexation, où r est l'ordre du guide de codification de Voronoï ;

si le vecteur code le plus proche est compris dans les limites du guide de codification étendu, alors l'étape vii) comprend l'indexation du vecteur code le plus proche dans le guide de codification étendu, donnant des indices de quantification, la sortie des indices de quantification, et l'arrêt du procédé ;
si le vecteur code le plus proche n'est pas compris dans les limites du guide de production étendu et si le nombre de bits nécessaires pour l'indexation du guide de codification n'a pas dépassé un certain seuil, alors la répétition des étapes viii) à x) afin d'augmenter le facteur d'échelle et l'ordre du guide de codification de Voronoï ;
si le nombre de bits nécessaires pour l'indexation du guide de codification étendu a dépassé un seuil prédéterminé, la réalisation de l'étape de ix) la prise en compte du vecteur code le plus proche comme étant un déviant distant et la sortie des indices de quantification correspondants prédéterminés.

34. Procédé tel que revendiqué dans la revendication 33, dans lequel le facteur d'échelle prédéterminé est 2 et le facteur d'échelle est augmenté de 2.

35. Procédé tel que revendiqué dans la revendication 33, dans lequel l'ordre du guide de codification de Voronoï est augmenté de 1.

36. Codeur de signal permettant de coder un signal source destiné à la transmission ou au stockage en utilisant une quantification multi-débit en treillis, le codeur comprenant :

des moyens récepteurs permettant de fournir un vecteur source x représentant une trame d'échantillons de signal source ;
des moyens de mémoire incluant un guide de codification de base C dérivé d'un treillis Λ de points ;
des moyens d'association au vecteur source x d'un point de treillis y dans le treillis Λ; **caractérisé par**
des moyens permettant de vérifier si le point de treillis est inclus dans le guide de codification de base C ;
si le point de treillis y est inclus dans le guide de codification de base C, des moyens d'indexation du point de treillis y dans le guide de codification de base C donnant des indices de quantification ;
si le point de treillis y n'est pas inclus dans le guide de codification de base C, des moyens d'extension du guide de codification de base et permettant d'obtenir un guide de codification étendu ; des moyens d'association au point de treillis y comme vecteur code c à partir du guide de codification étendu ; et des moyens d'indexation du point de treillis y dans le guide de codification étendu et d'obtention des indices de quantification ;
dans lequel les indices de quantification forment une représentation quantifiée du vecteur source x.

37. Codeur de signal permettant de coder un signal source en utilisant une quantification multi-débit en treillis, tel que défini dans la revendication 36, dans lequel :

les moyens d'extension du guide de codification de base C comprennent des moyens de fourniture d'un sous-ensemble V de points du treillis Λ ;
les moyens d'indexation du point de treillis y dans le guide de codification étendu comprenant des moyens d'indexation du point y dans un nombre entier de guide de codification n et un indice i tel que $y = mc + v$, dans lequel c est un élément du guide de codification de base C, v est un élément du sous-ensemble V, et m est un nombre entier supérieur ou égal à 2 ;
le codeur de signal comprend des moyens de définition du guide de codification de base C, du sous-ensemble V du treillis Λ, du nombre entier m et de l'indice i pour le nombre de guide de codification n.

38. Décodeur de signal permettant de décoder un signal source codé transmis ou stocké en utilisant une quantification multi-débit en treillis, le décodeur comprenant :

des moyens de mémoire permettant de fournir un guide de codification de base C dérivé d'un treillis Λ ;
des moyens de réception permettant de fournir un nombre de guide de codification codé n et un indice de quantification codé i ;
caractérisé par
des moyens de démultiplexage de l'indice de quantification i en utilisant le nombre de guide de codification n ;
des moyens de vérification du nombre de guide de codification $n = 0$,
si $n = 0$, des moyens de décodage de l'indice de quantification i en utilisant le guide de codification de base C, en donnant un vecteur code quantifié y ;
des moyens de vérification si $n > 0$;
si $n > 0$, des moyens de fourniture d'un guide de codification de Voronoï présélectionné $V^{(r)}$; des moyens de définition d'un ordre d'extension sur $r = n$ et d'un facteur d'échelle sur $m = 2^r$; des moyens de démultiplexage des indices j et k de l'indice de quantification i ; des moyens de décodage de j de sorte à obtenir un vecteur code c à partir du guide de codification de base C ; des moyens de décodage de l'indice k de façon à obtenir un vecteur code v à partir du guide de codification de Voronoï $V^{(r)}$; et des moyens de reconstruction d'un vecteur code quantifié tel que $y = mc + v$;
dans lequel le vecteur code reconstruit y représente une représentation quantifiée d'une trame d'échantillons du signal source.

39. Décodeur de signal utilisant une quantification multi-débit en treillis, tel que défini dans la revendication 38, dans lequel :

les moyens de fourniture d'un guide de codification de Voronoï présélectionné $V^{(r)}$ comprennent des moyens de fourniture d'un sous-ensemble V du treillis Λ ;
les moyens de reconstruction du vecteur code quantifié y tel que $y = mc + v$ comprennent des moyens de reconstruction du point de treillis c comme élément du guide de codification de base C et des moyens de reconstruction du point de treillis v comme élément du sous-ensemble V du treillis Λ ;
dans lequel le décodeur de signal comprend des moyens de définition du guide de codification de base C et du sous-ensemble V du treillis Λ, du nombre entier m et de l'indice i à partir du nombre du guide de codification n.

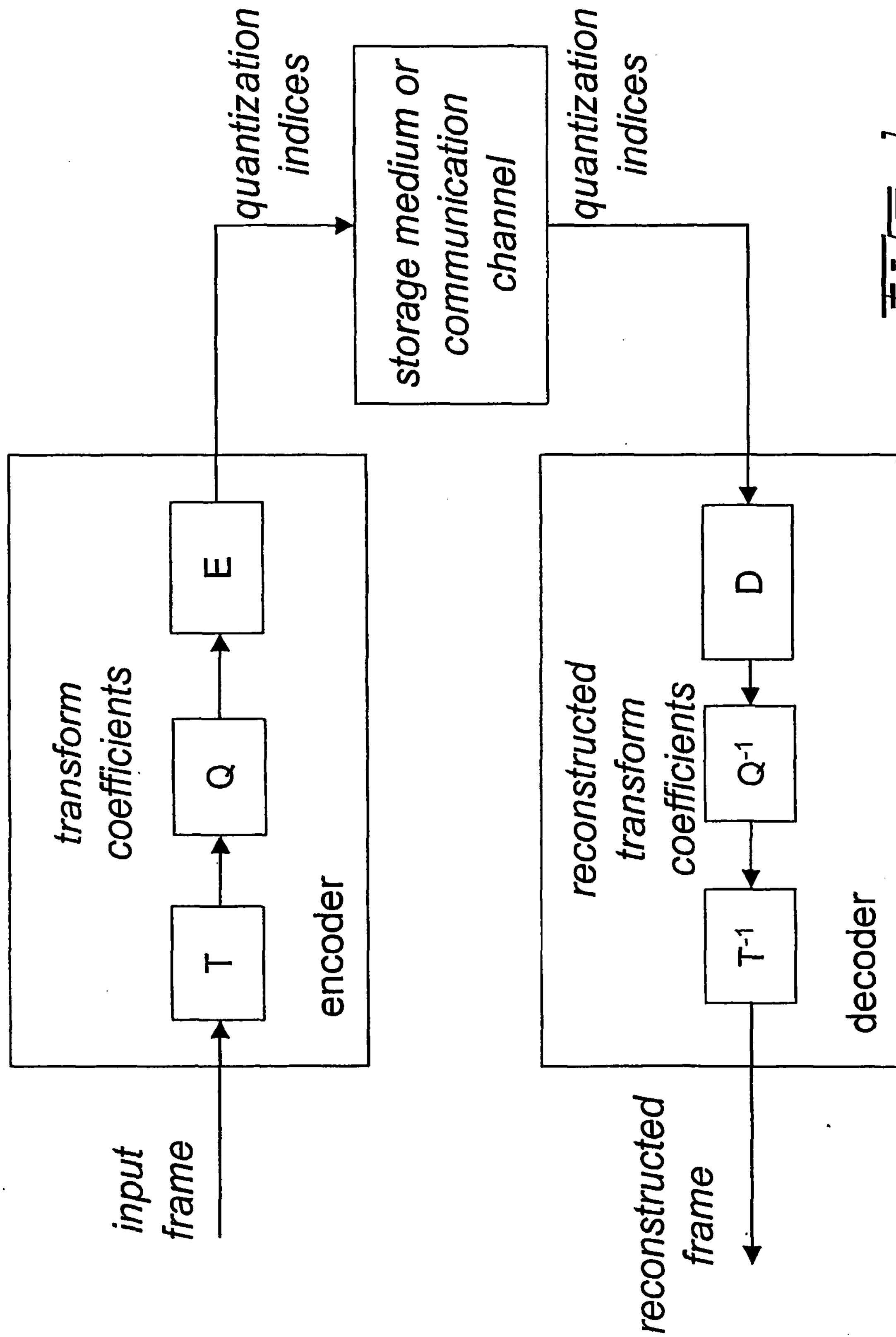

FIG. 1
(PRIOR ART)

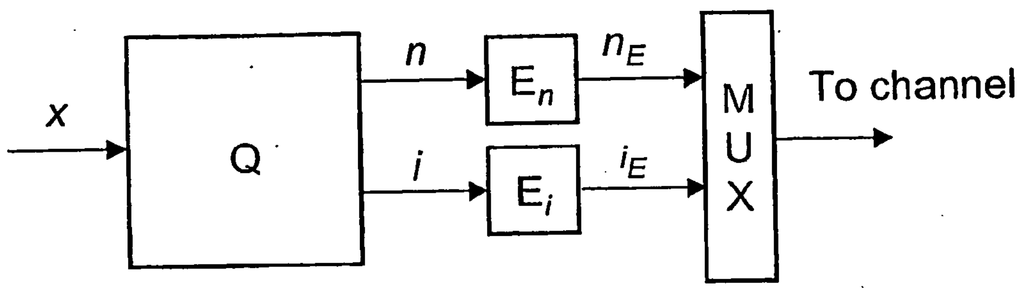

FIG. 2A
(PRIOR ART)

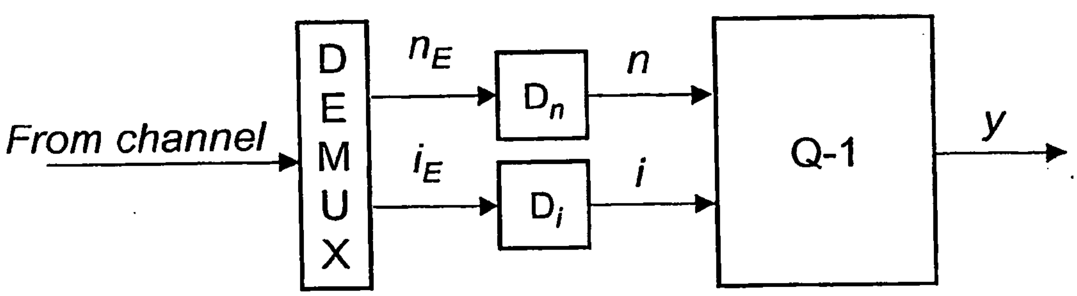

FIG. 2B
(PRIOR ART)

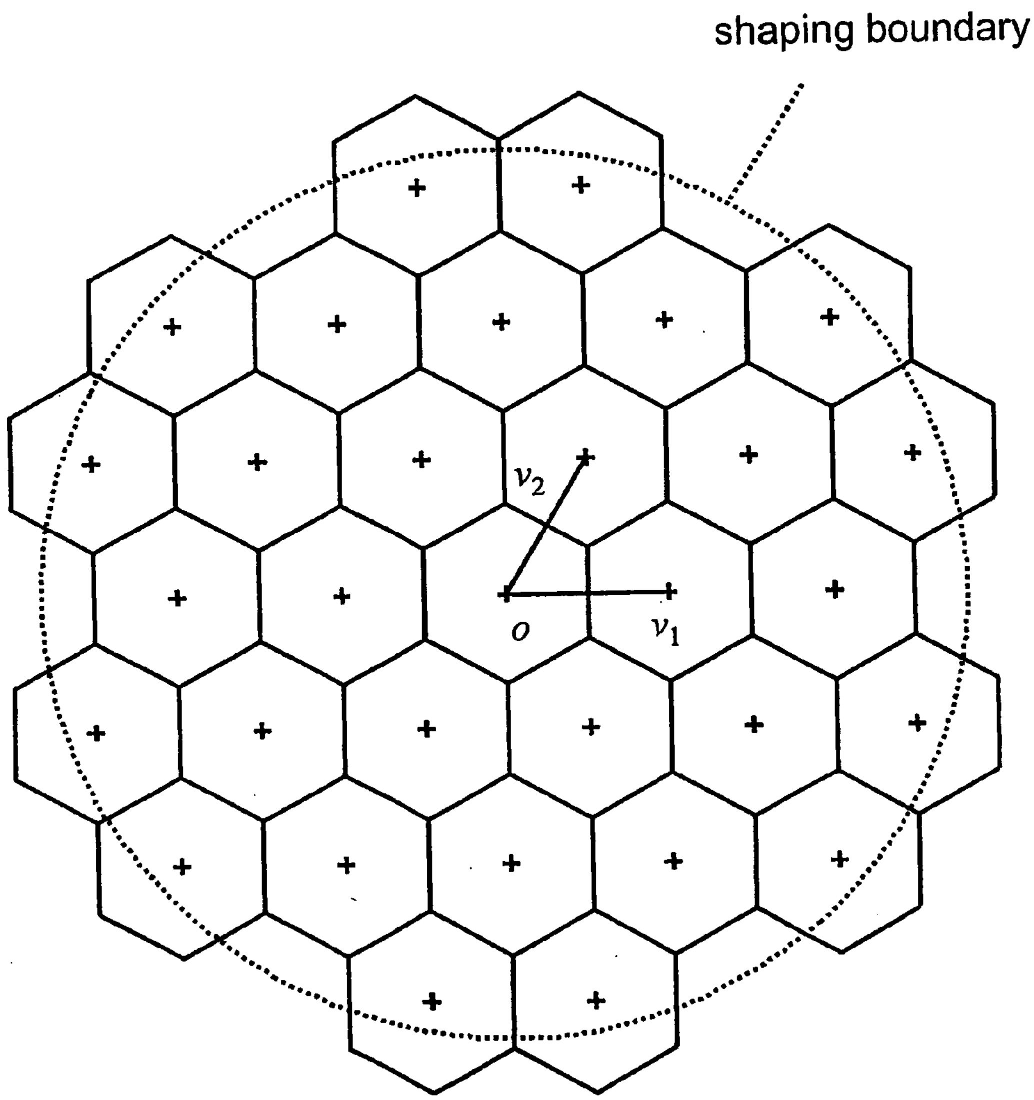

FIG. 3
(PRIOR ART)

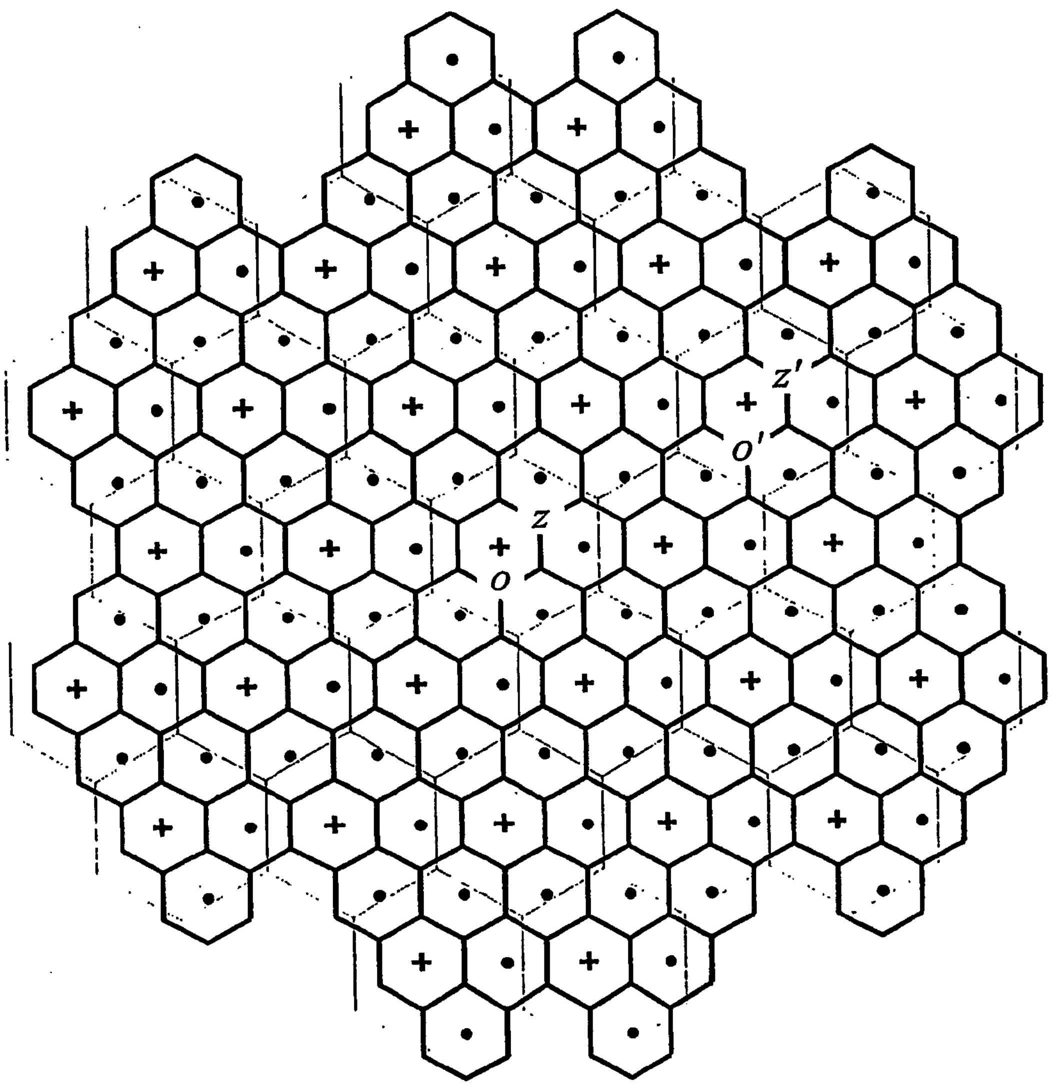

FIG. 4
(PRIOR ART)

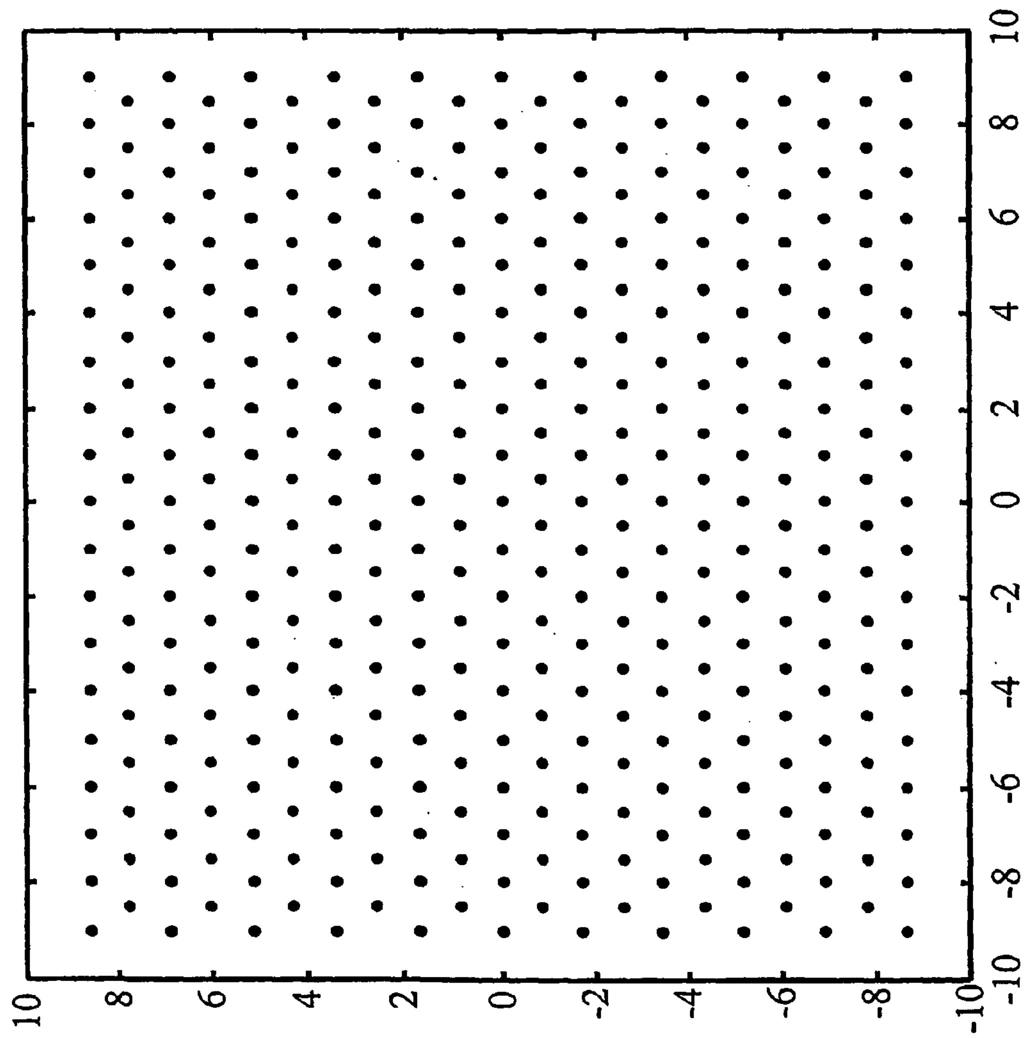

FIG. 5

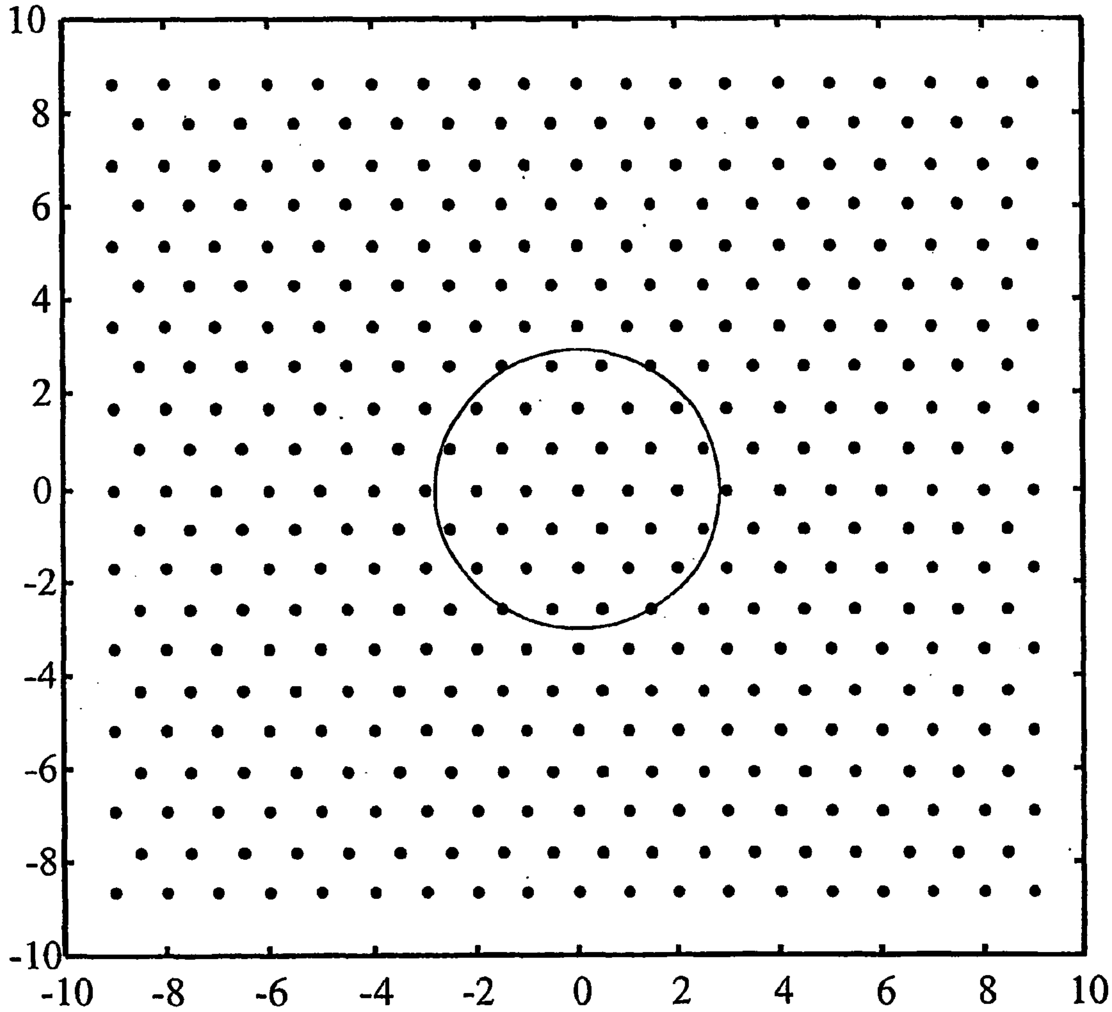

FIG_6

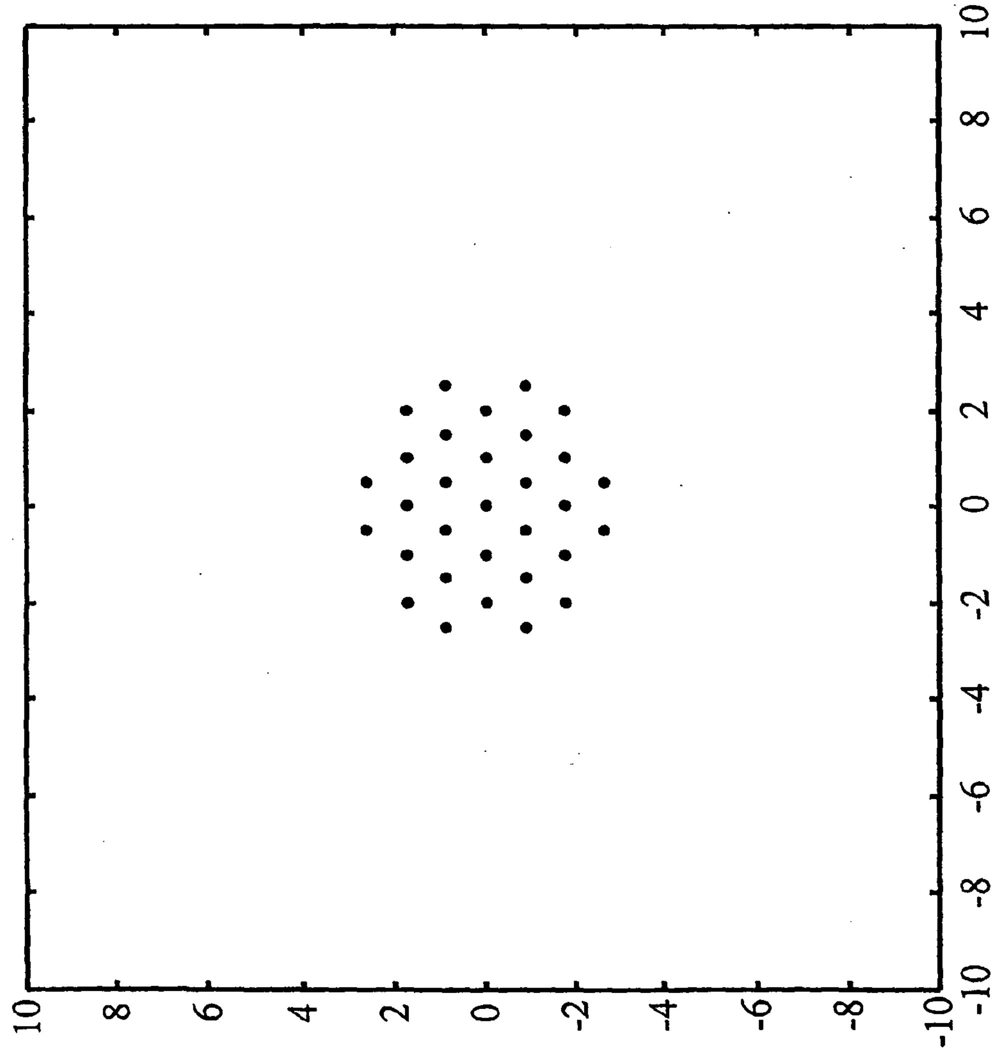

Fig. 7

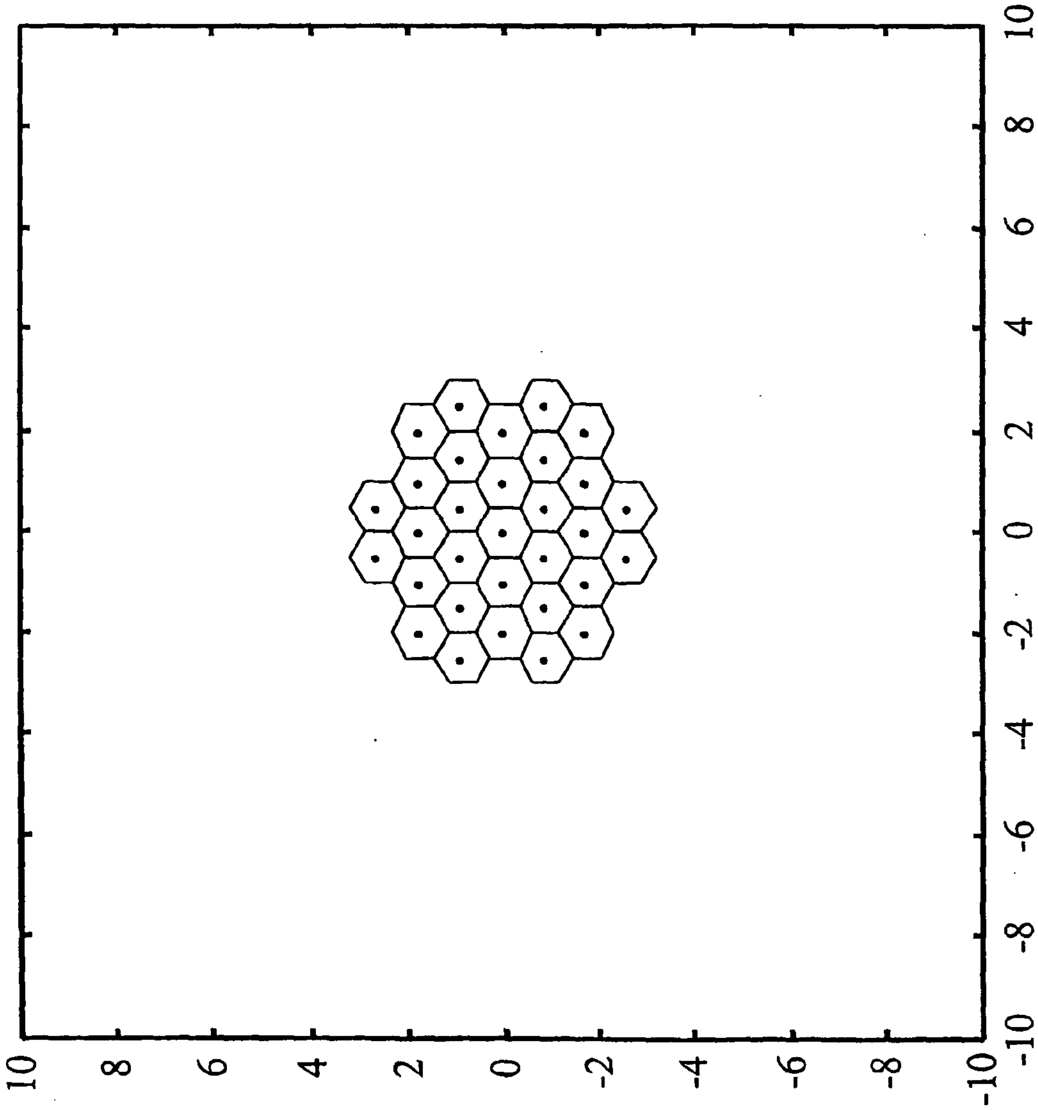

FIG. 8

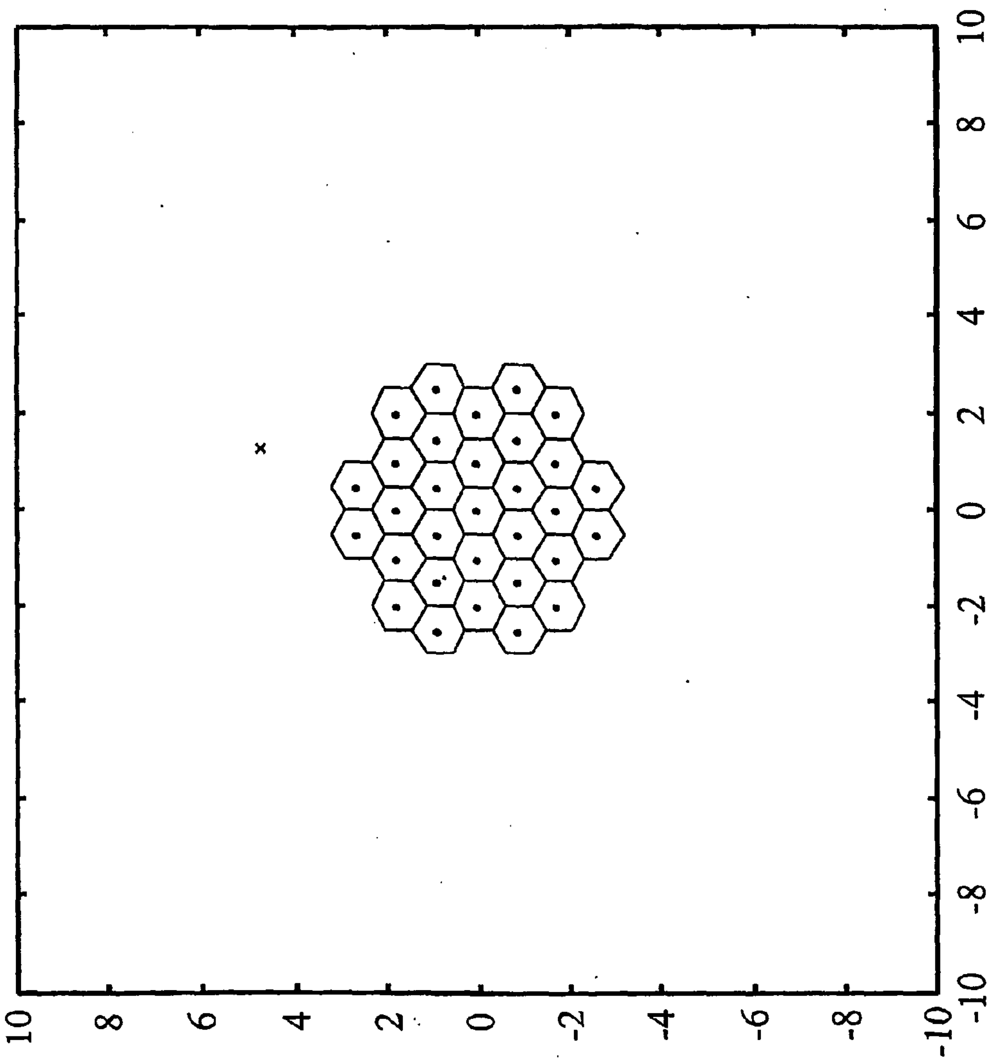

FIG. 9

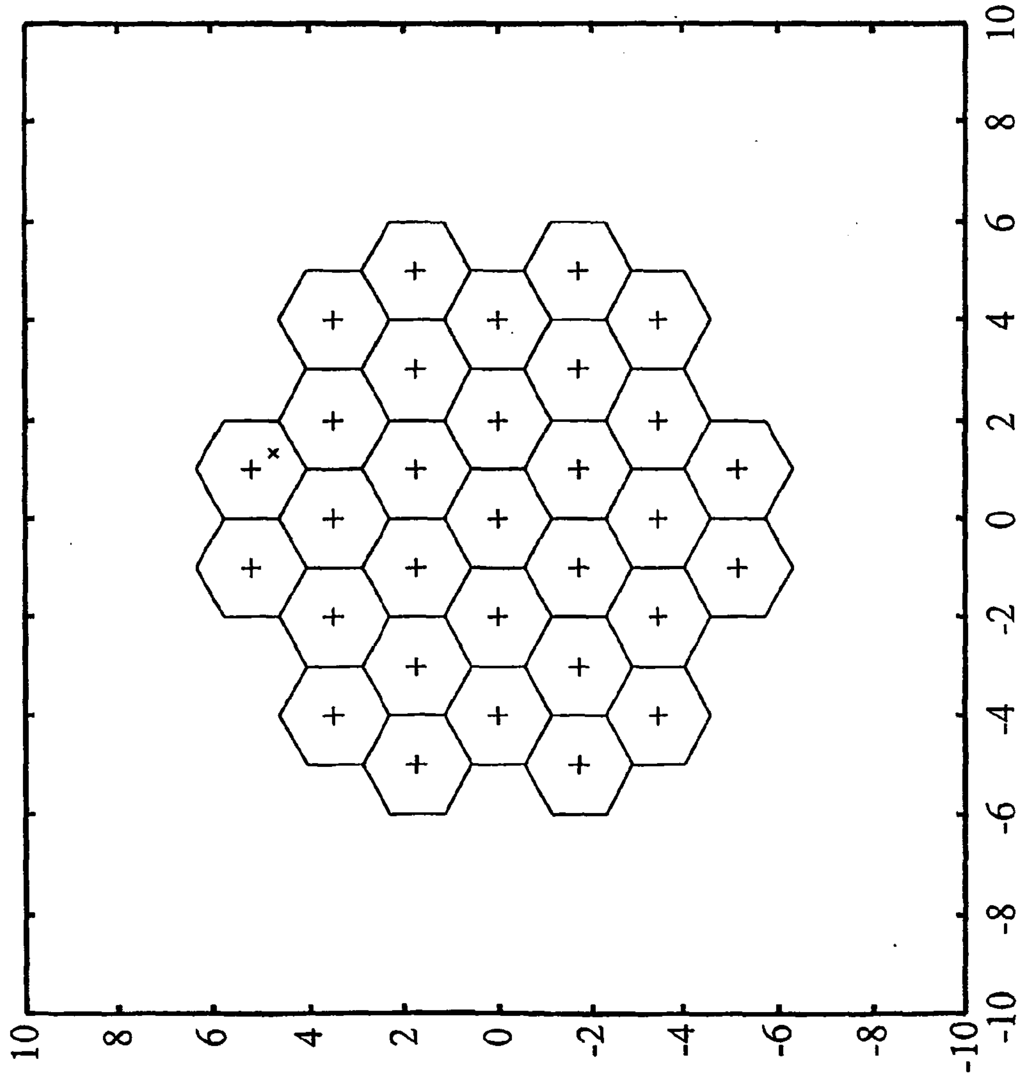

FIG. 10

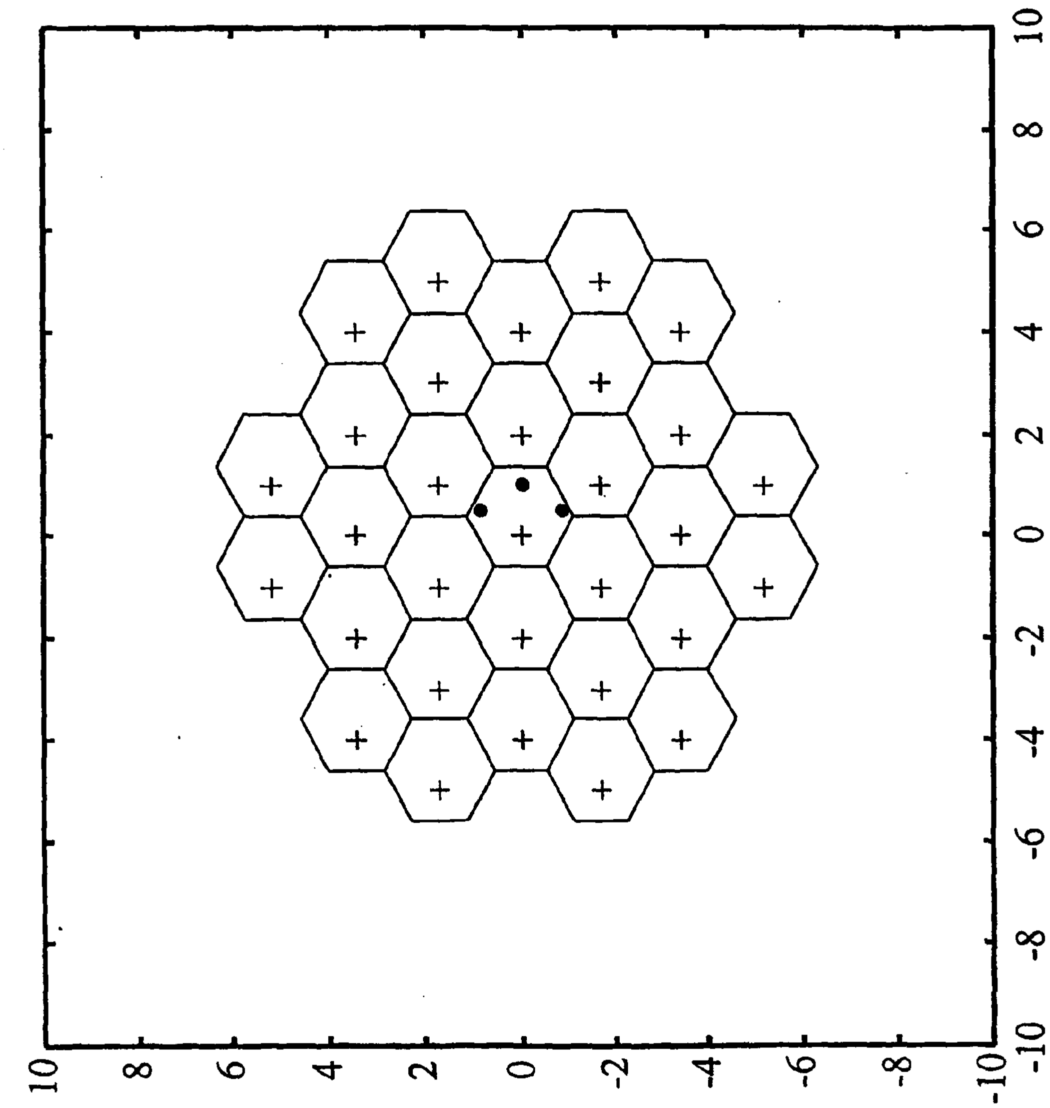

FIG. 11

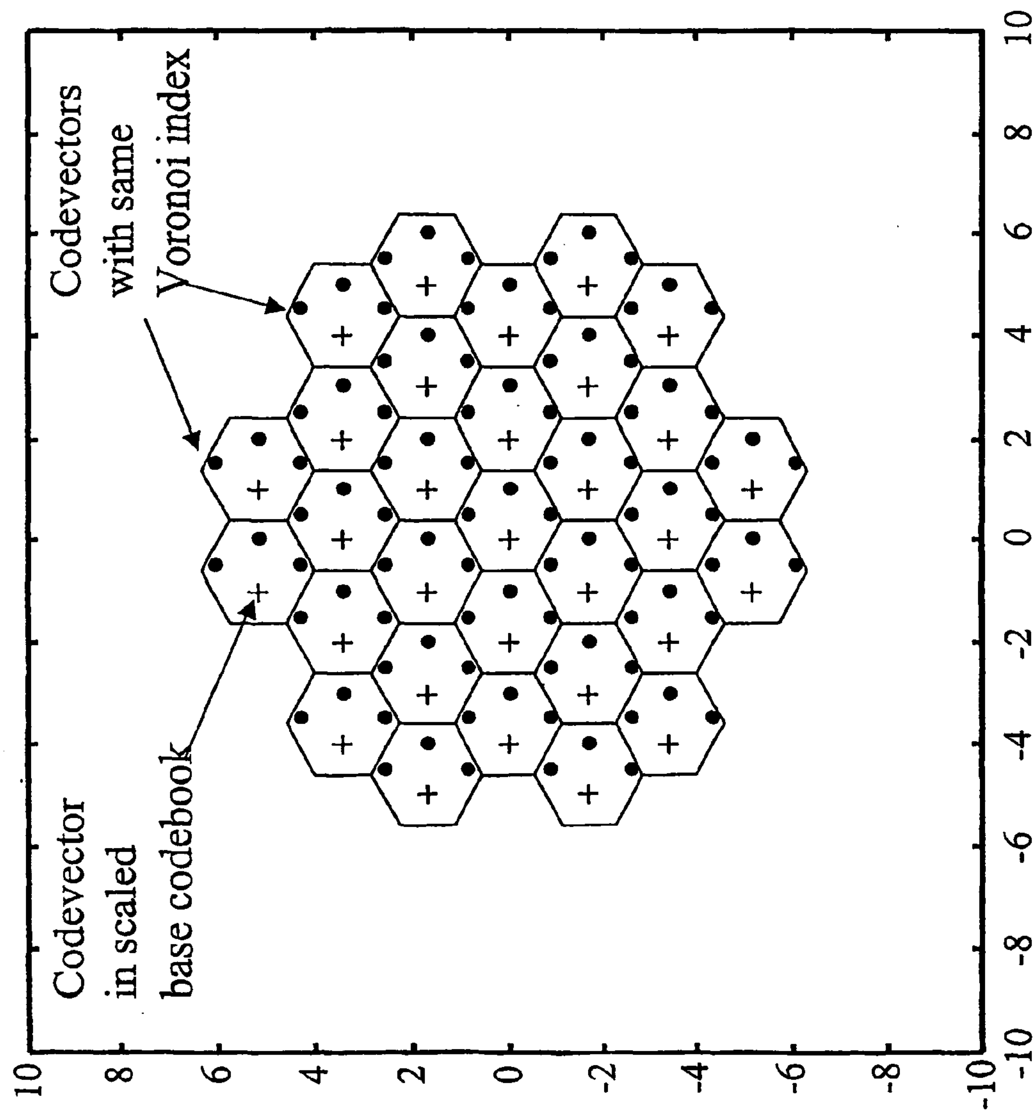
Codevectors
with same
Voronoi index
Codevector
in scaled
base codebook
10
8
6
4
2
0
-2
-4
-6
-8
-10

FIG. 12

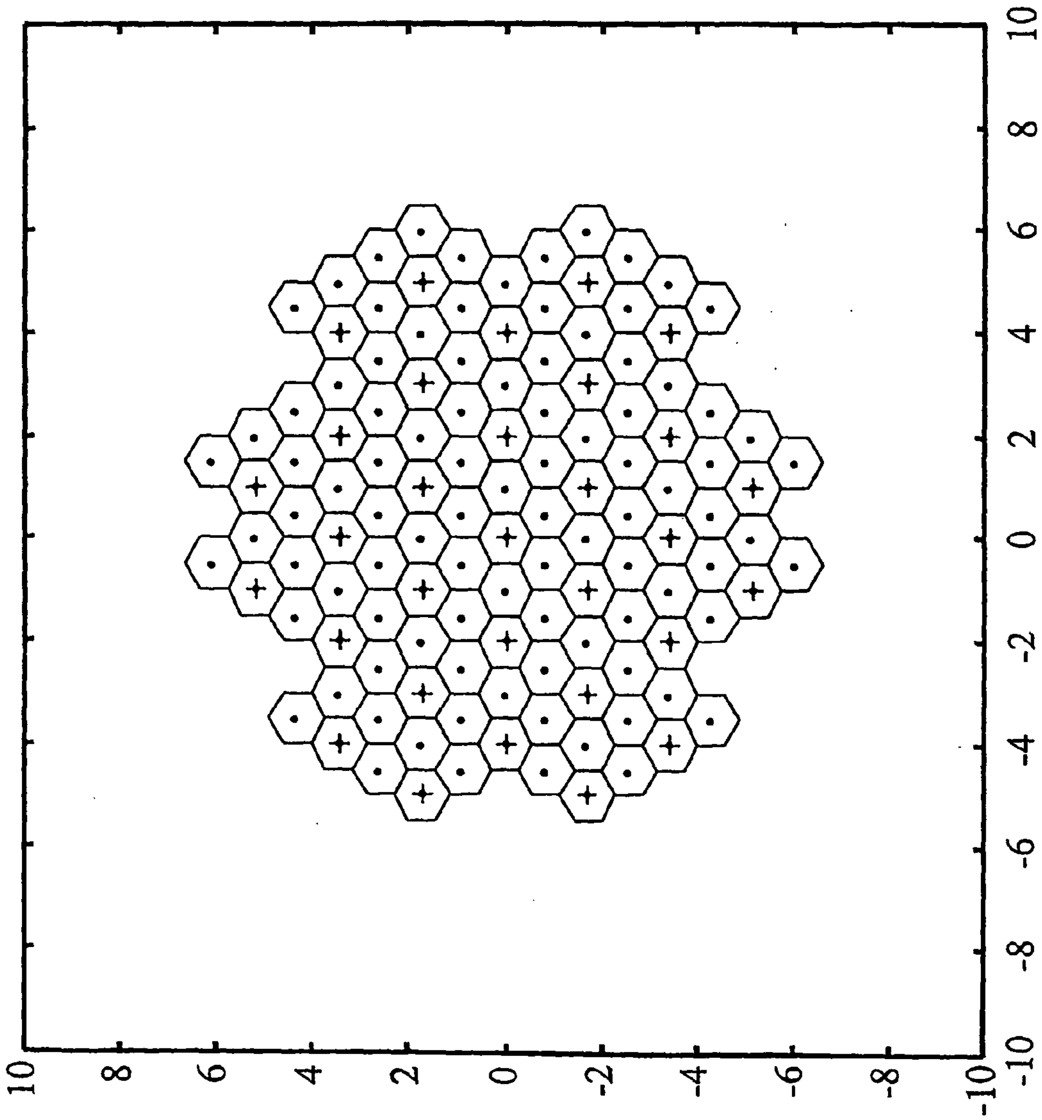

FIG. 13

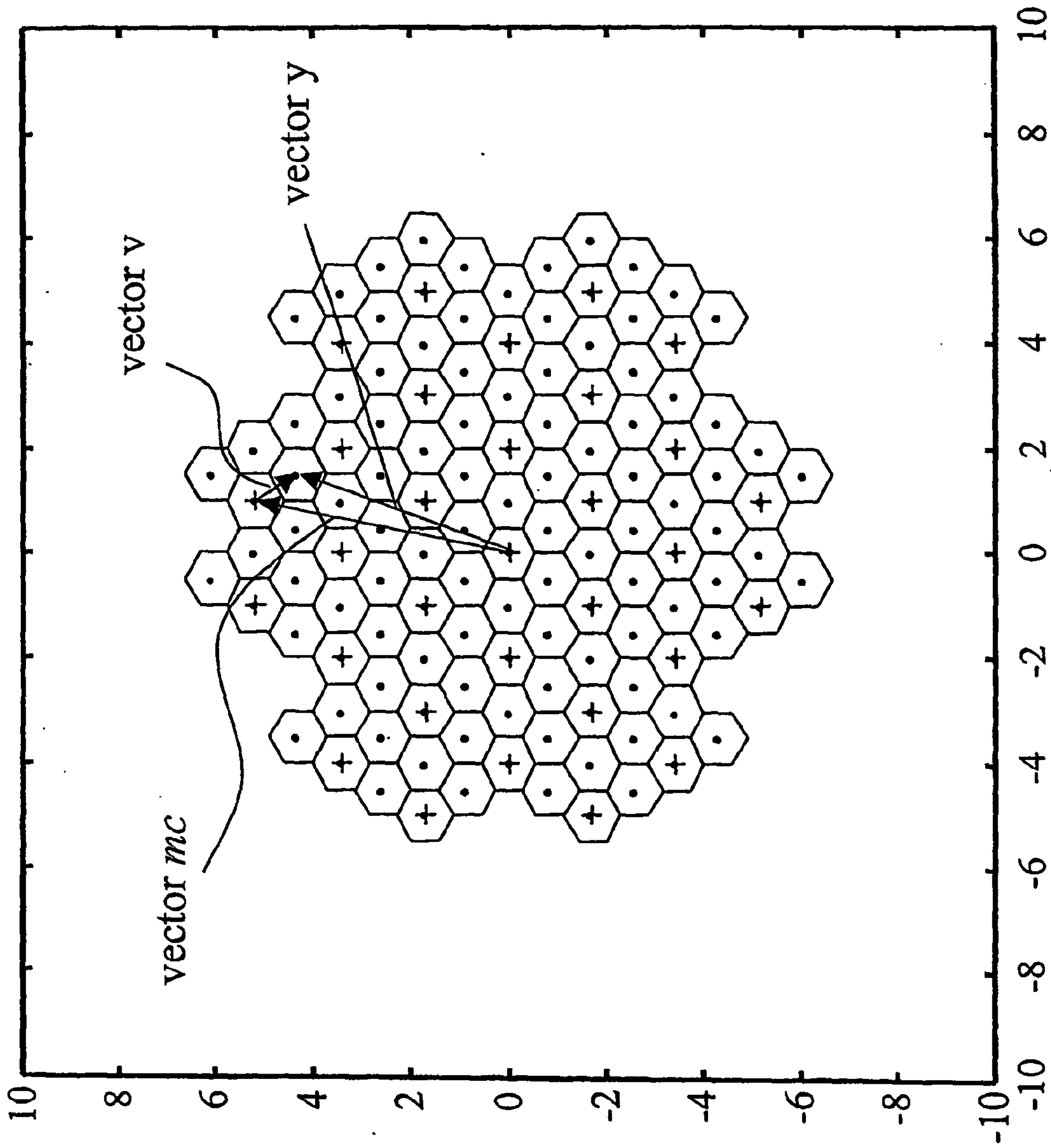

FIG. 14

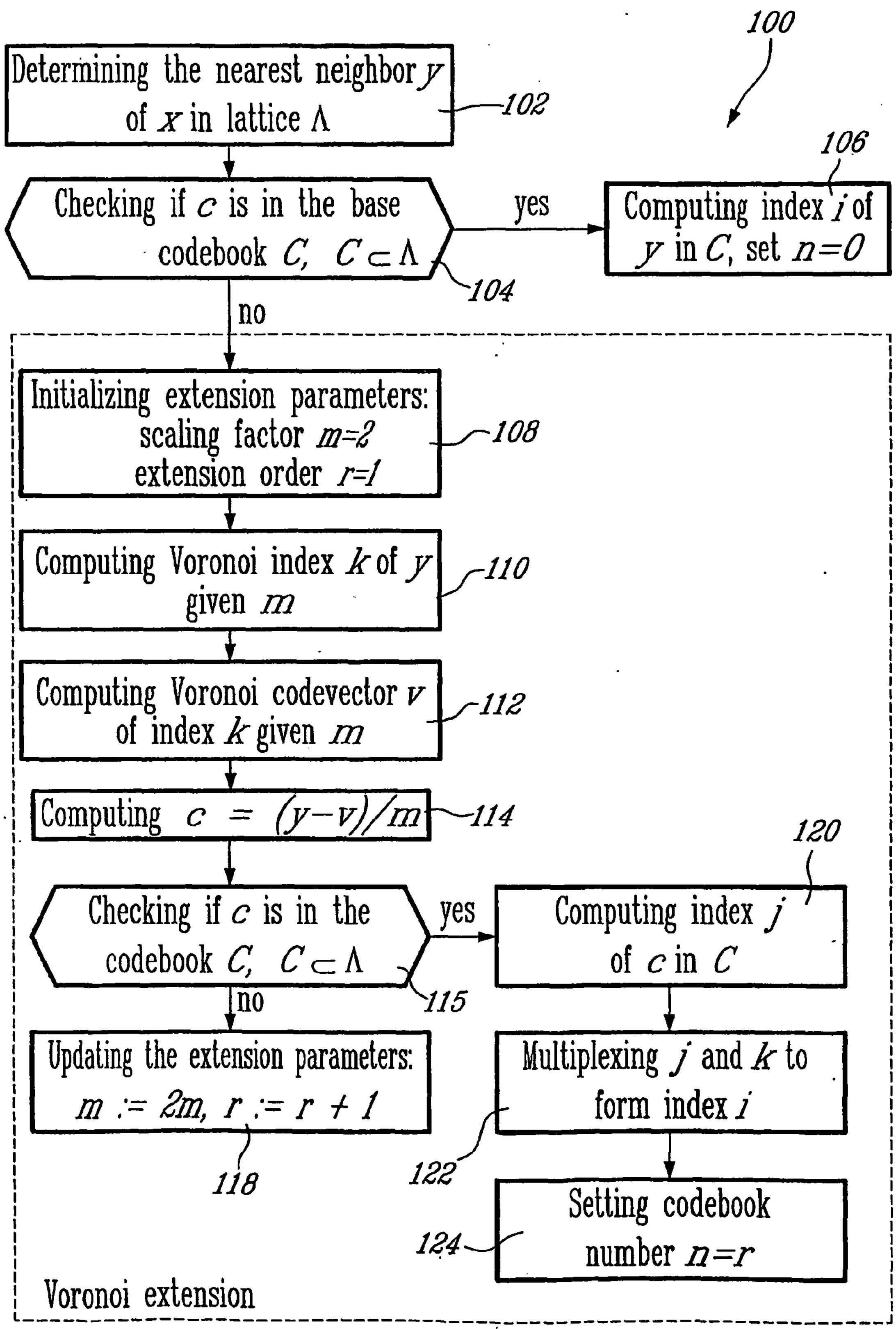
100
Determining the nearest neighbor y of x in lattice Λ
102
Checking if c is in the base codebook C, C⊂Λ
104
yes
Computing index i of y in C, set n=0
106
no
Initializing extension parameters: scaling factor m=2 extension order r=1
108
Computing Voronoi index k of y given m
110
Computing Voronoi codevector v of index k given m
112
Computing c = (y−v)/m
114
Checking if c is in the codebook C, C⊂Λ
115
yes
Computing index j of c in C
120
no
Updating the extension parameters: m := 2m, r := r + 1
118
Multiplexing j and k to form index i
122
Setting codebook number n=r
124
Voronoi extension

FIG. 15

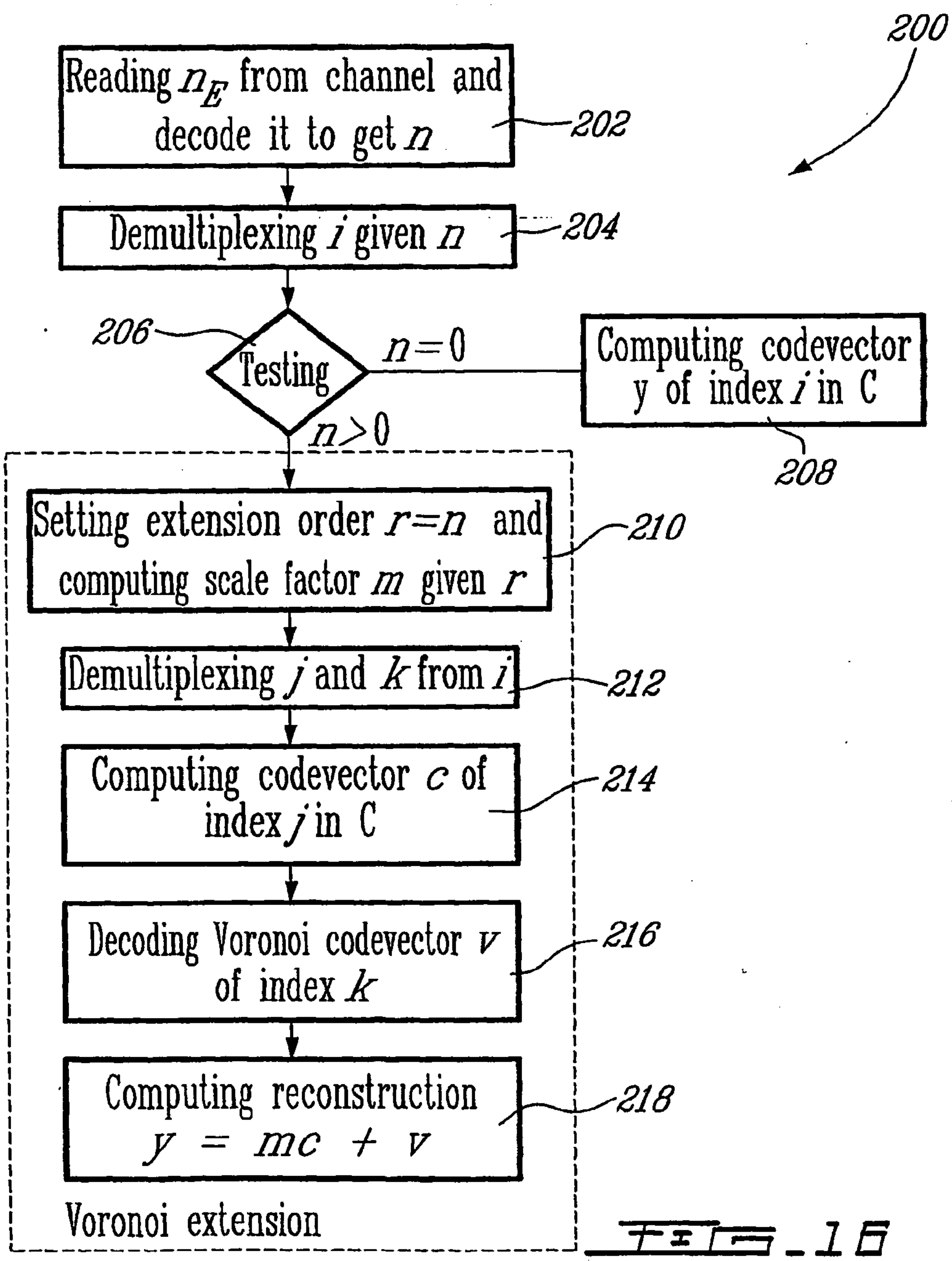
200
Reading n_E from channel and decode it to get n
202
Demultiplexing i given n
204
206
Testing
$n=0$
Computing codevector y of index i in C
208
$n>0$
Setting extension order $r=n$ and computing scale factor m given r
210
Demultiplexing j and k from i
212
Computing codevector c of index j in C
214
Decoding Voronoi codevector v of index k
216
Computing reconstruction $y = mc + v$
218
Voronoi extension

FIG. 16

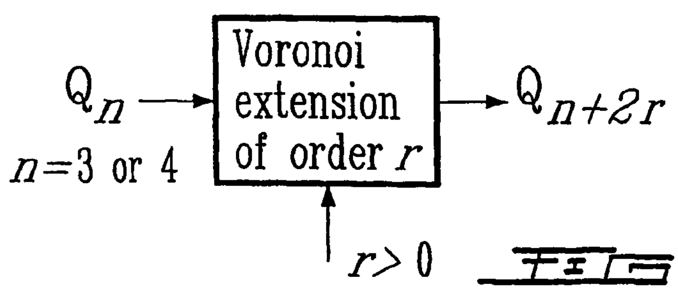
Q_n
$n=3$ or 4
Voronoi extension of order r
Q_{n+2r}
$r>0$

FIG. 17

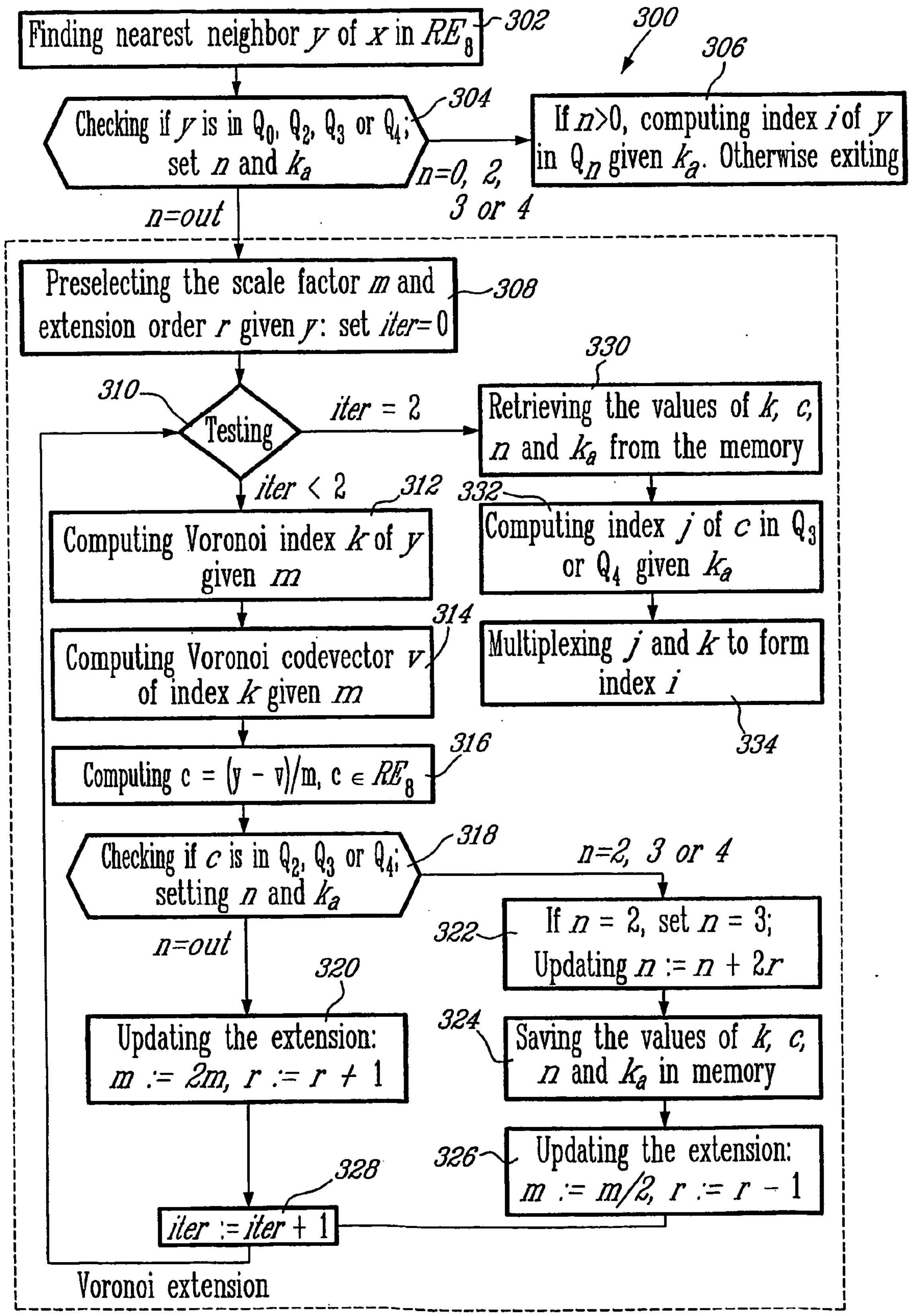
Finding nearest neighbor y of x in RE8
302
300
306
Checking if y is in Q0, Q2, Q3 or Q4; set n and ka
304
If n>0, computing index i of y in Qn given ka. Otherwise exiting
n=0, 2, 3 or 4
n=out
Preselecting the scale factor m and extension order r given y: set iter=0
308
330
310
Testing
iter = 2
Retrieving the values of k, c, n and ka from the memory
iter < 2
312
332
Computing Voronoi index k of y given m
Computing index j of c in Q3 or Q4 given ka
314
Computing Voronoi codevector v of index k given m
Multiplexing j and k to form index i
316
334
Computing c = (y - v)/m, c ∈ RE8
318
Checking if c is in Q2, Q3 or Q4; setting n and ka
n=2, 3 or 4
n=out
322
If n = 2, set n = 3; Updating n := n + 2r
320
Updating the extension: m := 2m, r := r + 1
324
Saving the values of k, c, n and ka in memory
326
Updating the extension: m := m/2, r := r - 1
328
iter := iter + 1
Voronoi extension

FIG. 18

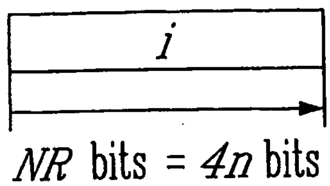

FIG. 19A

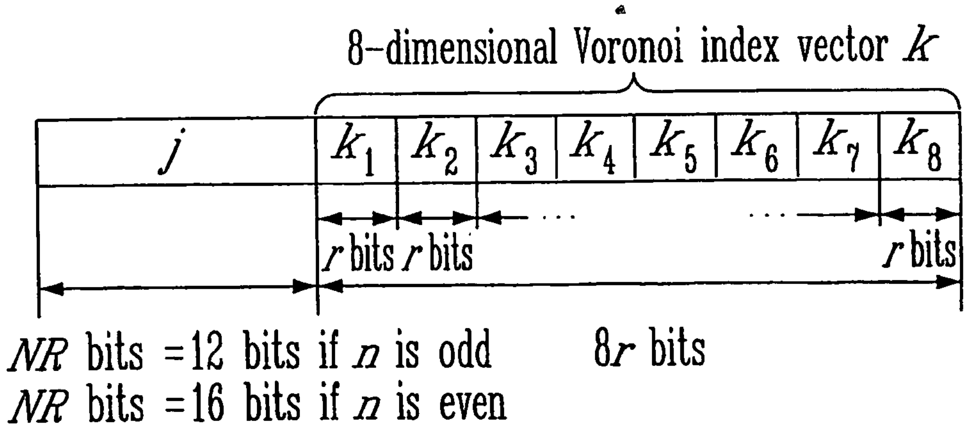

FIG. 19B

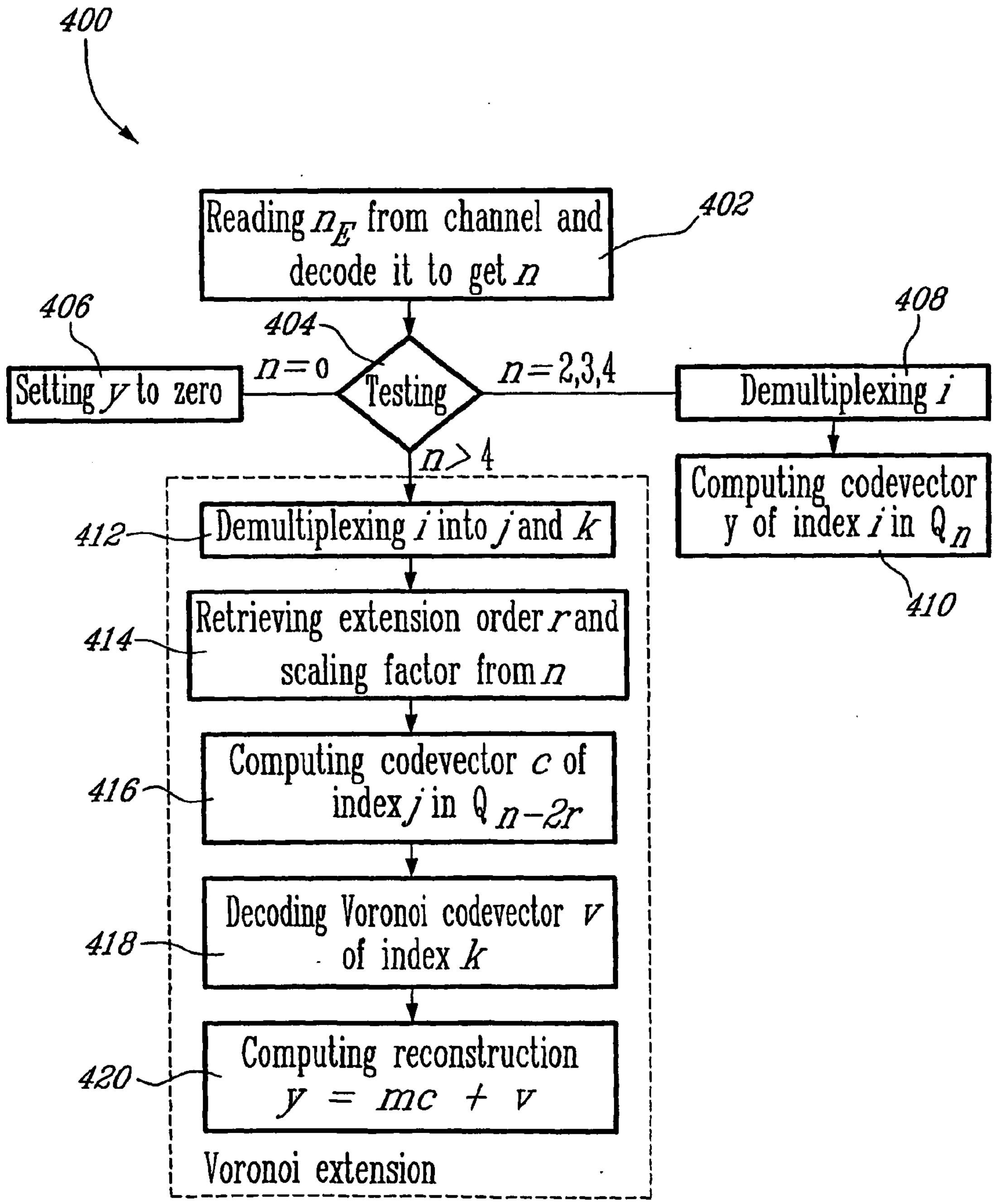
400
Reading n_E from channel and decode it to get n
402
406
404
408
Setting y to zero
$n=0$
Testing
$n=2,3,4$
Demultiplexing i
$n>4$
Computing codevector y of index i in Q_n
412
Demultiplexing i into j and k
410
414
Retrieving extension order r and scaling factor from n
416
Computing codevector c of index j in Q_{n-2r}
418
Decoding Voronoi codevector v of index k
420
Computing reconstruction $y = mc + v$
Voronoi extension

FIG. 20

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Non-patent literature cited in the description

- **C. Lamblin ; J.-P. Adoul.** Algorithme de quantification vectorielle sphérique à partir du réseau de Gosset d'ordre 8. *Ann. Télécommun.,* 1988, vol. 43 (3-4), 172-186 **[0017]**
- **J.-M. Moureaux ; P. Loyer ; M. Antonini.** Low-complexity indexing method for Zn and Dn lattice quantizers. *IEEE Trans. Communications,* December 1998, vol. 46 (12 **[0017]**
- **P. Rault ; C. Guillemot.** Indexing algorithms for Zn, An, Dn, and Dn++ lattice vector quantizers. *IEEE Transactions on Multimedia,* December 2001, vol. 3 (4), 395-404 **[0017]**
- **M. Xie ; J.-P. Adoul.** Embedded algebraic vector quantization (EAVQ) with application to wideband audio coding. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP),* 1996, vol. 1, 240-243 **[0025]**
- **J.H. Conway ; N.J.A. Sloane.** A fast encoding method for lattice codes and quantizers. *IEEE Trans. Inform. Theory,* November 1983, vol. IT-29 (6), 820-824 **[0032]**
- **A. Gersho ; R.M. Gray.** Vector Quantization and Signal Compression. KluwerAcademic Publishers, 1992 **[0032]**
- **D. Mukherjee ; S.K. Mitra.** Vector set-partitioning with successive refinement Voronoi lattice VQ for embedded wavelet image coding. *Proc. ICIP,* October 1998, 107-111 **[0035]**
- **Y. Zhang et al.** An Improvement Image Vector Quantization Based on Affine Transformation. *SYSTEMS, MAN, AND CYBERNETICS, 1997. COMPUTATIONAL CYBERNETICS AND SIMULATION., 1997 IEEE INTERNATIONAL CONFERENCE ON ORLANDO,* 12 October 1997, 1094-1099 **[0036]**