NOKIA
Connecting People

FlexCode

Speech and audio coding with flexible geometric structures

Adriana Vasilache
Flexcode Seminar
March 12, 2009

Flexcode project

- European project (2006-2009)
  - Partners:
    - Academia
      - KTH Royal Institute of Technology
      - RWTH Aachen University
    - Industry
      - Ericsson
      - Orange group
      - Nokia

FlexCode

Flexcode project

NOKIA Connecting People

- Objectives of the project
  - develop a practical, flexible, parameterized, generic speech and audio coding system with the following properties:
    - instantaneous adaptation to rate, quality, and robustness requirements;
    - usage of advanced perceptual error criteria;
    - adaptation to channel quality: packet loss rate and bit error rate;
    - embedded bit rate scalability
    - reasonable computational and memory complexity.
  - demonstrate practical implementations of the proposed system for two relevant service applications, using speech and audio coding

Flexible geometric structures for coding

- Flexible
  - Definition: adaptable or variable
- Geometric structures
  - Lattices
- Coding
  - Lossy compression
    - Quantization $f : R^n \to C \in R^n$
    - Indexing function $I : C \to N$

Lattice definition

- Algebraically, an *n*-dimensional lattice Λ is a set of real vectors whose coordinates are integers in a given basis $\{b_i \in R^n\}_{i=\overline{1,n}}$.

$$\Lambda = \left\{ v \in R^n \mid v = \sum_{i=1}^{n} \alpha_i b_i, \alpha_i \in Z \right\}$$

- Geometrically, a lattice is an infinite regular array of points which uniformly fills the *n*-dimensional space.

Lattice advantages

- Parameters: scale and support region.
- There are efficient algorithms for finding nearest neighbors, indexing, and lossless coding lattice quantizers, which makes them attractive in some applications such as audio coding.
- Good theoretical approximations for performance – special case of high rate theory
- Approximately optimal if used with entropy coding: constant quantizer point density functions over finite volume regions.

Lattice terminology

- Lattice truncation

$$\overline{\Lambda} = \{x \in \Lambda \mid N(x) \le K\}$$

- Lattice shell

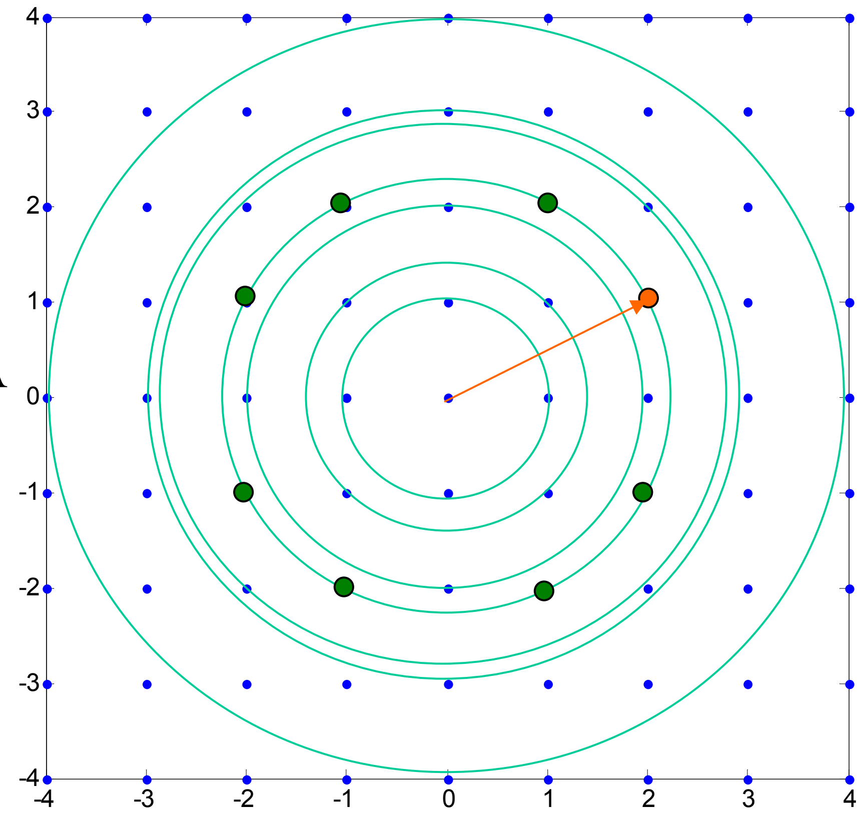
$$\overline{S_K} = \{x \in \Lambda \mid N(x) = K\}$$

- Lattice leader

$$v = (v_m, \ldots, v_m, \ldots, v_i, \ldots v_i, \ldots v_1, \ldots, v_1) \in \Lambda$$

$$0 < v_1 < \ldots < v_i < \ldots < v_m$$

- Leader class

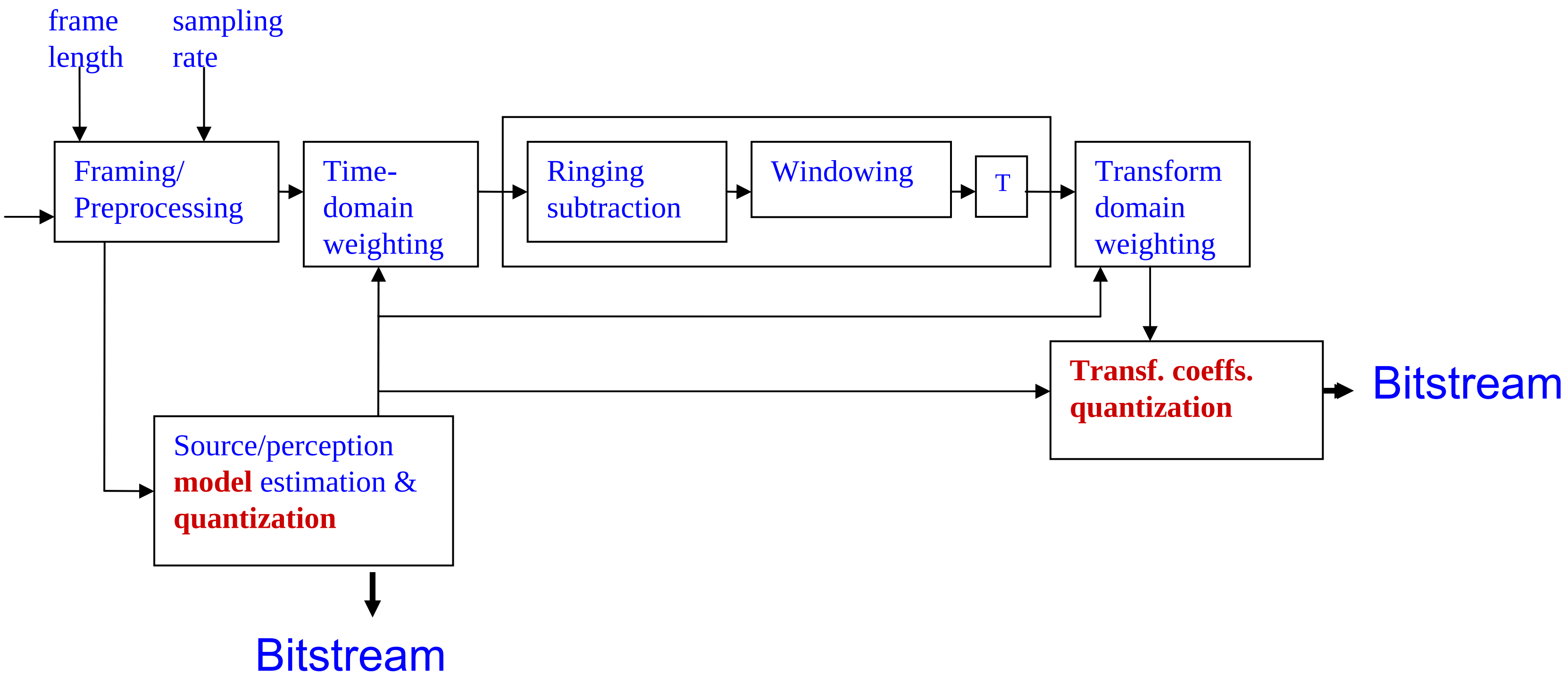


(2 1)
(-2 1)
(2 -1)
(-2 -1)
(1 2)
(-1 2)
(1 -2)
(-1 -2)

Presentation topics

- Speech and audio coding with geometric structures
  - Bitrate domain for lattice quantizers
  - Bit errors resilient indexing function
  - Lattice entropy encoding (continuous bit-rate adaptation)

High level overview of the coder



FlexCode

Flexibility?

NOKIA Connecting People

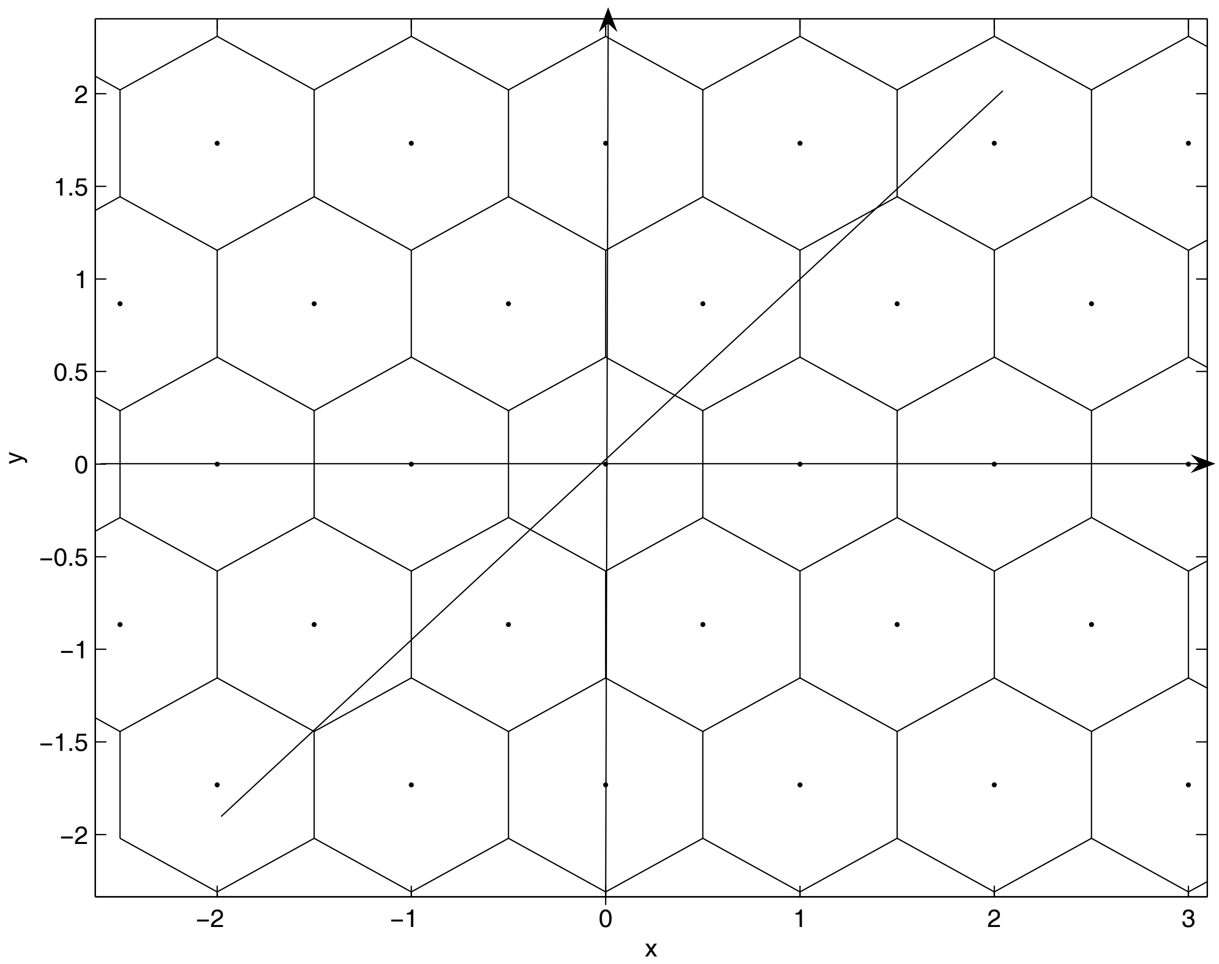
- Lattice quantizers are not efficient for very low bitrates. → Lattice rotations
- The indexing functions for lattices are generally sensitive to bit errors. → Bit error resilient indexing function
- Existing entropy coding methods for lattices are not practical in high dimensions. → Product code based entropy coding

Lattice rotations

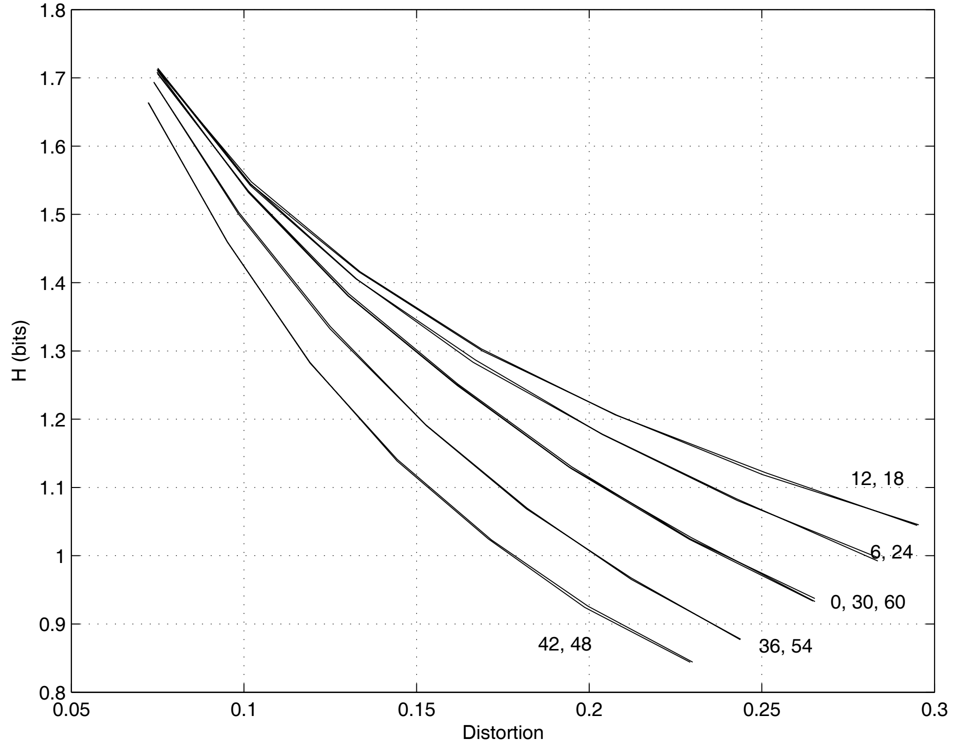
- Data has vanishing directions
- The bitrate is relatively low
  - i.e. high rate theory does not apply
  - lattice border effects are significant

➔ Rotate the lattice such that the ”denser” direction corresponds to the “denser” direction in the data.

Lattice rotations - Directions in a lattice (A_2)

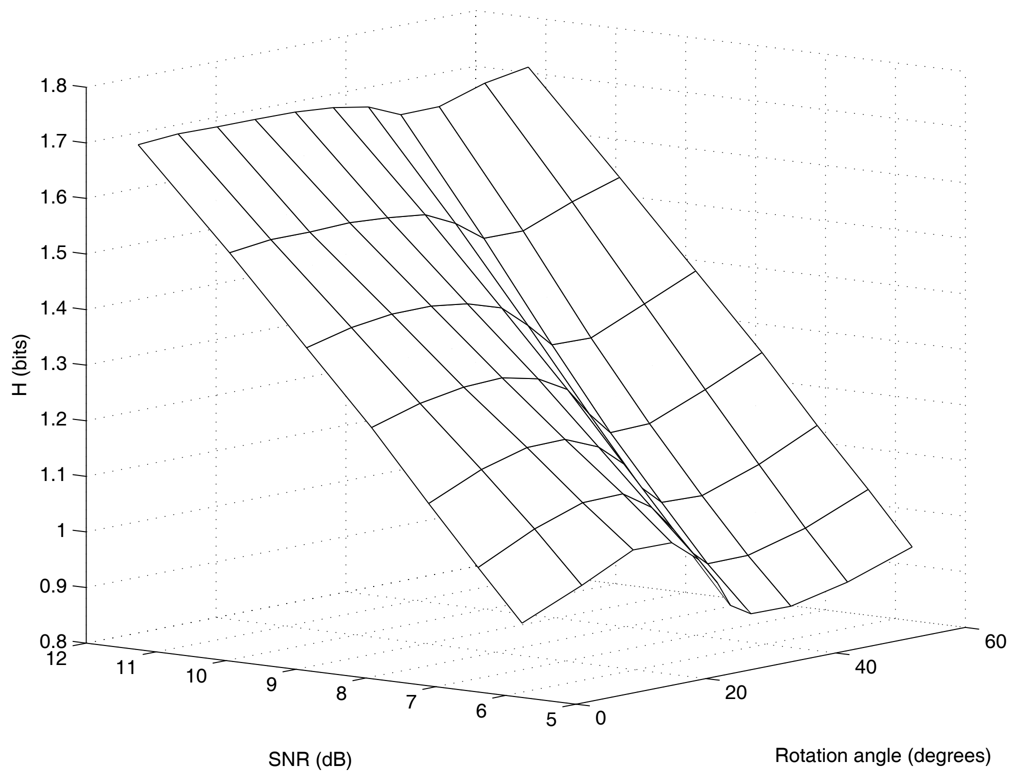


Lattice rotations for correlated data



Experimental rate distortion functions for rotated A2 lattices for Gaussian correlated sources with correlation coefficient 0.9

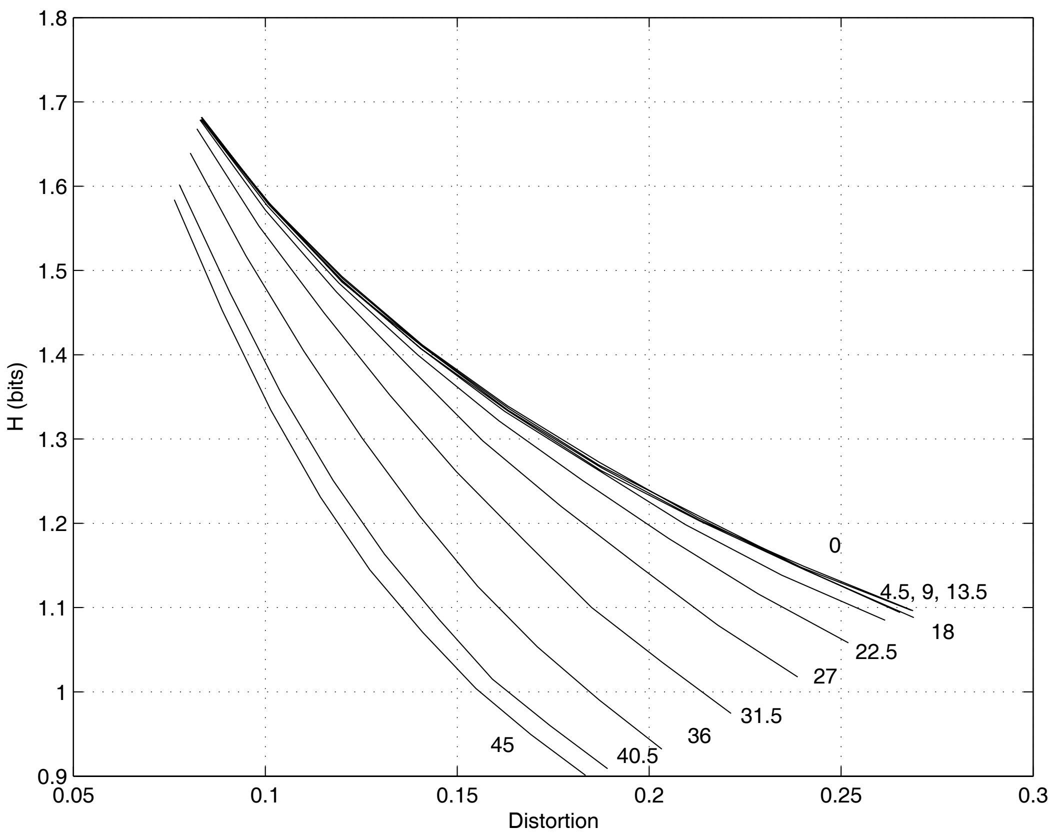
Lattice rotations for correlated data



Experimental rate distortion functions for rotated A2 lattices for Gaussian correlated sources with correlation coefficient

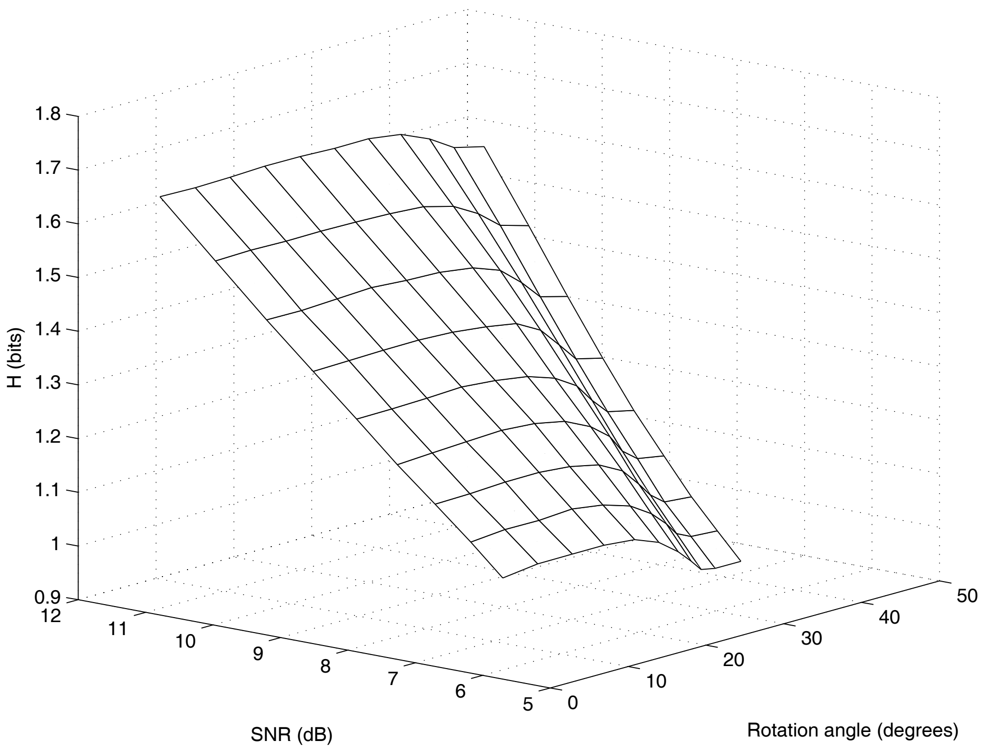
0.9 (3D view).

Lattice rotation for correlated data



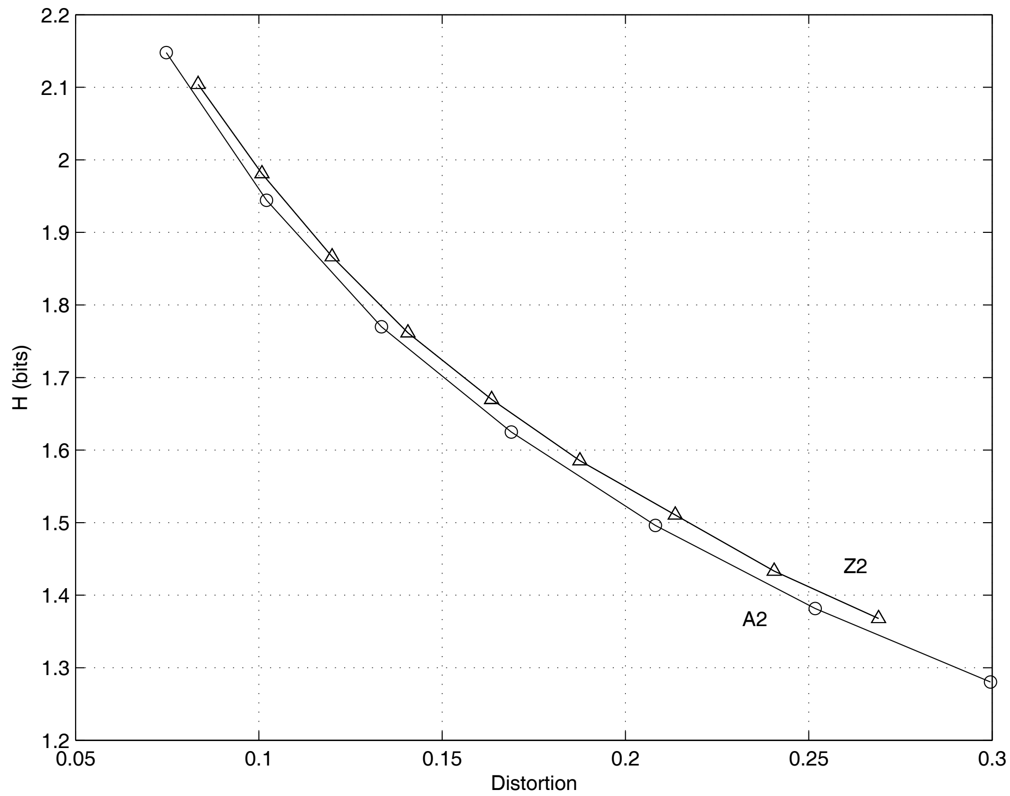
Experimental rate distortion functions for rotated Z2 lattices for Gaussian correlated sources with correlation coefficient 0.9

Lattice rotations for correlated data



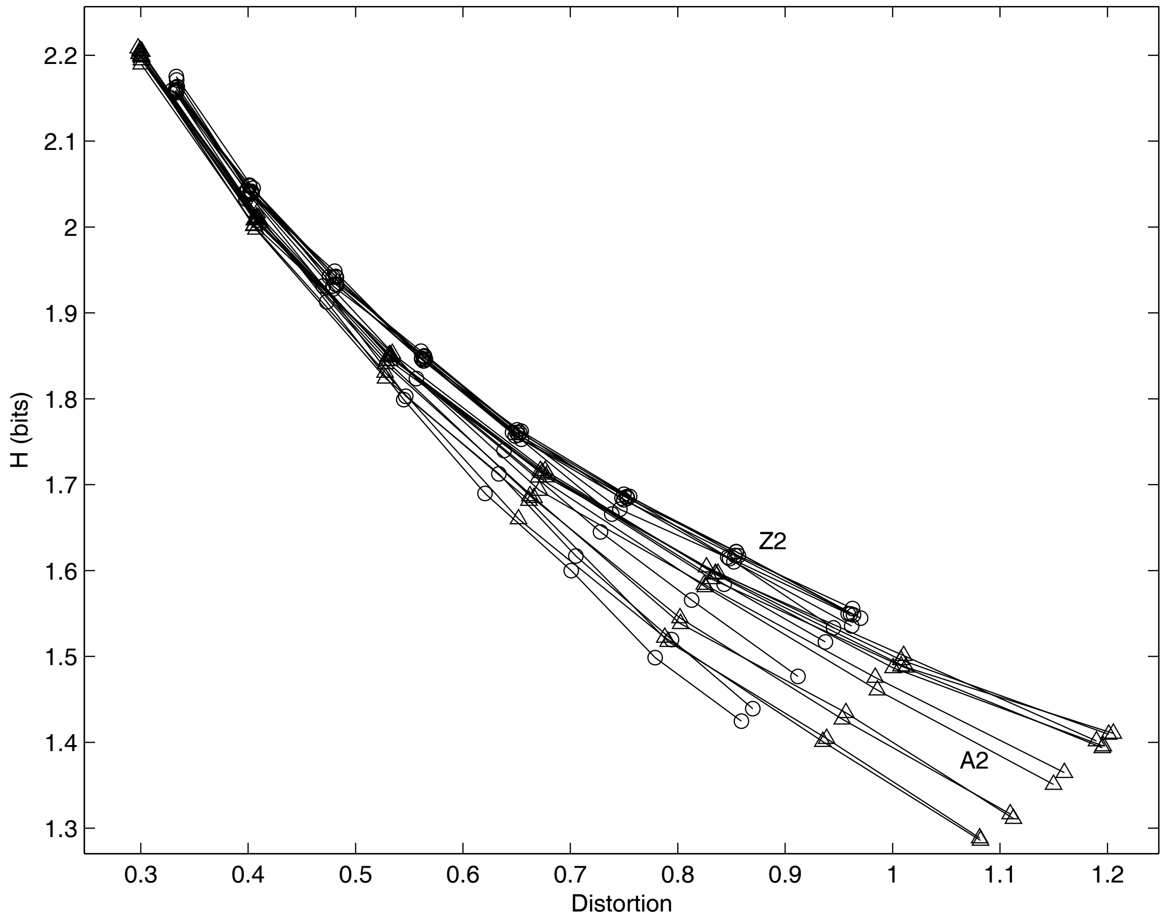
Experimental rate distortion functions for rotated Z2 lattices for Gaussian correlated sources with correlation coefficient 0.9 (3D view)

Lattice rotations – non-correlated data



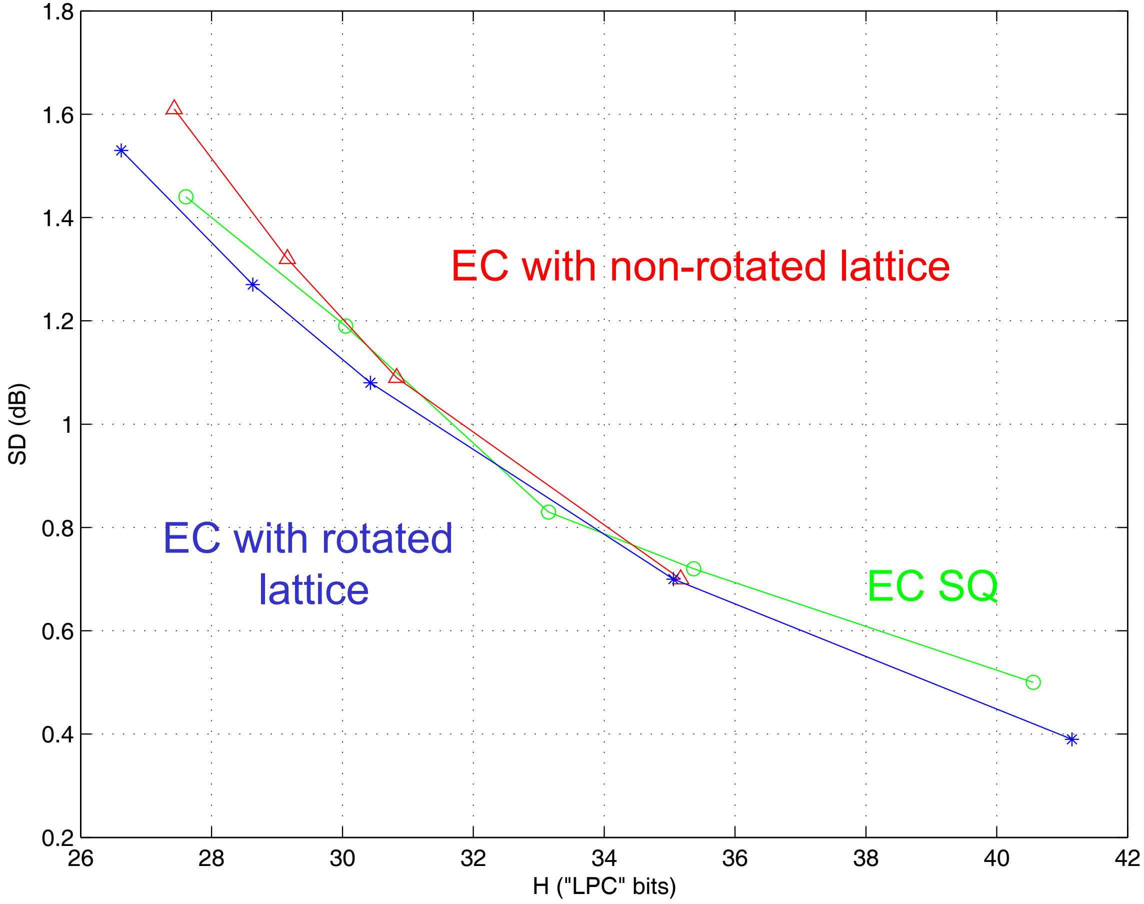
Experimental rate distortion functions for Z2 and A2 lattices for non-correlated Gaussian sources with same std. dev on both directions.

Lattice rotations for data with vanishing directions



Experimental rate distortion functions for rotated Z2 and A2 lattices for non-correlated Gaussian sources with std. dev. ratio 1:4

Comparison of theoretical number of bits needed to encode the LPC coefficients



Lattice indexing functions

- Practical indexing algorithms
  - Usually they require the lattice definition as
    - union of leaders
    - union of shells
    - large memory storage requirements
  - The union of leaders should be large enough to include all codevectors that are likely to appear at the quantization in the infinite lattice.

Indexing on leaders (IL)

- The lattice is defined as a union of leader classes.
- The lattice points are enumerated within the leader class they belong to.
- Each leader class has an offset index.
- Enumeration of the unsigned vectors followed by sign enumeration.

Indexing on shells (IS)

- The lattice is defined as a union of shells (under the given norm).
- The lattice points are enumerated within the shells they belong to.
- Each shell has an offset index.

IS – product code indexing

- Truncations (spherical, pyramidal, rectangular) of Zn lattices and cosets of Zn
- E.g. for a rectangular shell of Zn
  - $\max\{|x_i|\}=K$, $(x_1,\ldots,x_n)$ lattice point
  - $\exists\ 0 < j \leq n$ such that $x_j = K$
  - Product code
    - Number of significant components
    - Number of maximum valued components
    - Position of maximum valued components
    - Values of significant non-maximum components
    - Position of significant non maximum values
    - Sign of significant components

IS – product code indexing –ctnd.

Example Indexing of a 3D vector having maximum norm equal to 2.

n=3, K=2

		Vectors	Indexes
S=3	2 2 2 M=3	±2 ±2 ±2	0,…, 7,
	2 2 1 M=2	±2 ±2 ±1	8,…,15,
		±2 ±1 ±2	16,…,23,
		±1 ±2 ±2	24,…,31,
	2 1 1 M=1	±2 ±1 ±1	32,…,39,
		±1 ±2 ±1	40,…,47,
		±1 ±2 ±2	48,…,55,
S=2	2 2 0 M=2	±2 ±2 0	56,…,59,
		±2 0 ±2	60,…,63,
		0 ±2 ±2	64,…,67,
	2 1 0 M=1	±2 ±1 0	68,…,71,
		…	…
S=1	2 0 0 M=1		

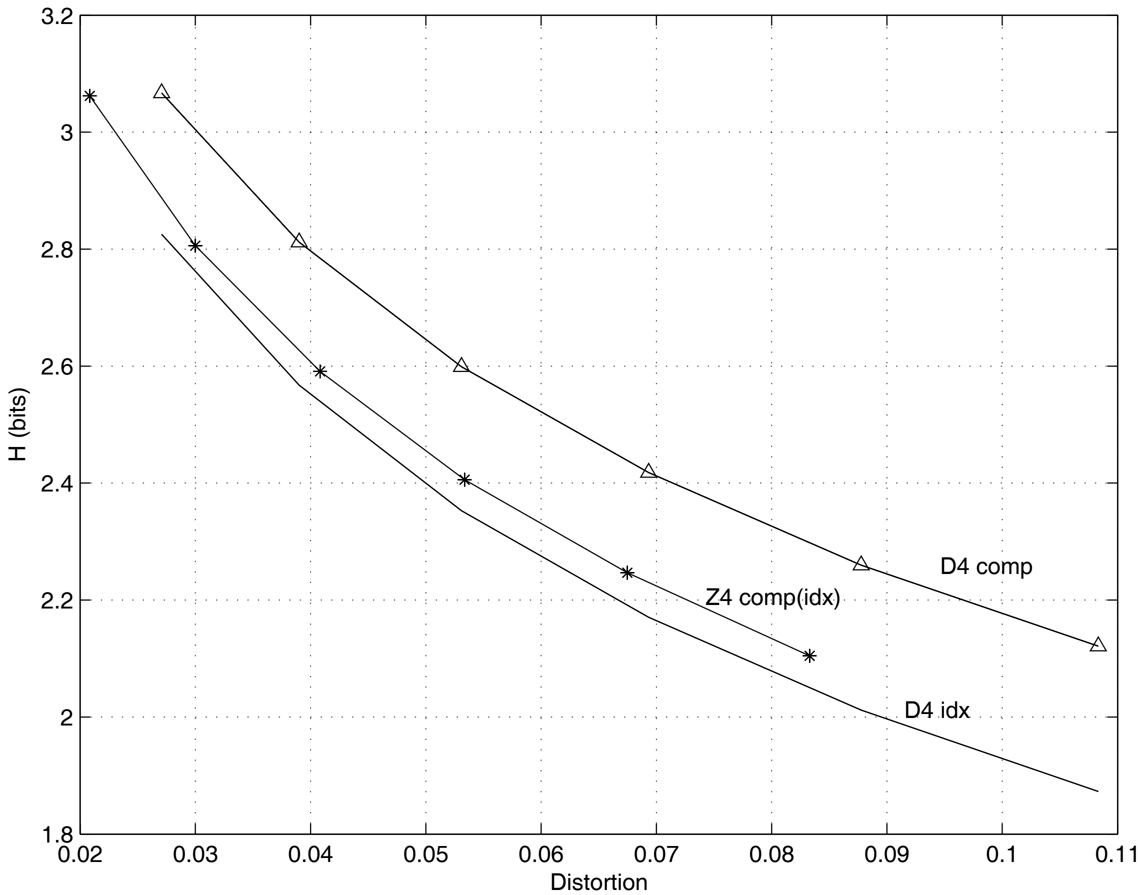
IS - applications

- Index factorization – beneficial for channel error resilience
  - Demo
- Index factorization
  - Enables scalable indexing because only parts of the index can be decoded to an approximation of the initial codevector.

Entropy encoding of lattice codevectors

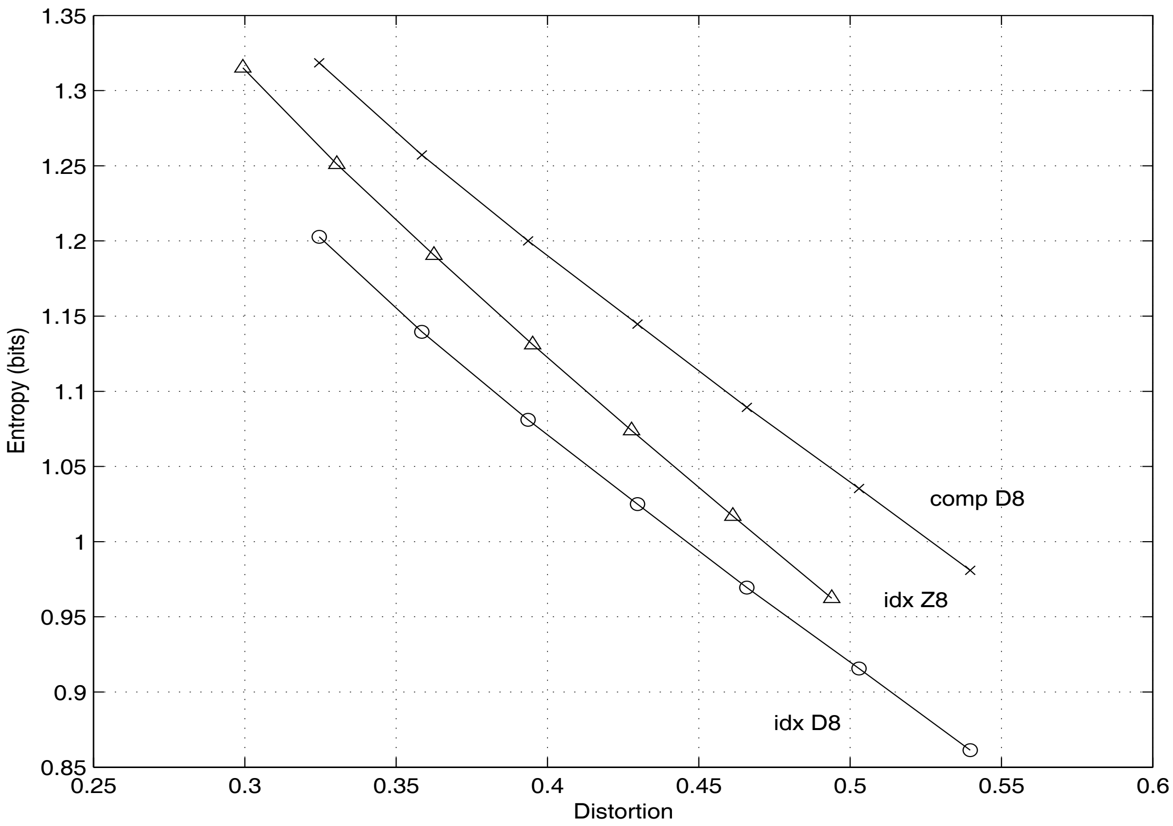
- Ideal case: considering each codevector individually
  - For high dimensions the number of codevectors, even for low bitrates, becomes too large.
- Practical methods:
  - Group the codevectors into sets (shells, leader classes) and entropy encode the index of the set, while the index of the lattice codevector within the set is encoded using enumerative encoding.
  - Entropy encode the lattice codevector components

Lattice entropy coding 4D



Entropy comparison between coding lattice codevector components and lattice codevector indices for Z4 and D4 lattices. Independent Gaussian data is used.

Lattice entropy coding 8D



Entropy comparison between coding lattice codevector components and lattice codevector indices for Z8 and D8 lattices. Independent Gaussian data is used.

Product code

- The idea of the product code is to extract different informational entities from the vector to be indexed and concatenate their respective codes.

Product code

- E.g. The information contained in the vector from a rectangular Zn lattice truncation is represented by the following entities:
  - The number of the significant components (S);
  - The number of maximum valued components (in absolute value) (M);
  - The position of the significant values (posS)
  - The position of the maximum valued components (posM);
  - The values of the significant non-maximum components (idx_nonM);
  - The signs of the significant components (Sg).

Product code - Entropy coding

- The different informational entities extracted from the vector, can be also interpreted as means of classifying the vectors into different sets.
- The existence of several entities implies the division of all the vectors into sets, sub-sets and so forth.
- If the indexes corresponding to all or part of the set (sub-set) types are entropy encoded, an entropy code can be obtained for the initial lattice vector.
- The statistics of the data from each of the sets and sub-sets are different and the data is coded accordingly.

E.g. Bitrate calculation for rectangular truncation of Z_n, for the norm K.

- Any vector from this set can be represented on N bits.
- Enumerative encoding

$$N = \left| \log_2\left((2K+1)^n\right) \right|$$

Product code for entropy encoding-1

- The number of significant values, S, is entropy encoded on n1 bits

$$N = n_1 + \left\lceil \log_2 \left(2^S \left(\binom{n}{1}\binom{n-1}{S-1}(K-1)^{S-1} + \binom{n}{2}\binom{n-2}{S-2}(K-1)^{S-2} + \ldots + \binom{n}{S} \right) \right) \right\rceil$$

FlexCode

Product code for entropy encoding-2

NOKIA
Connecting People

- The number of significant values is entropy encoded on n_1 bits;
- The number of maximum valued components, M, is encoded on n_2 bits;
- The index of positions for the maximum valued components is encoded on n_3 bits;
- The index of positions for the significant values is encoded on n_4 bits.

$$N = n_1 + n_2 + n_3 + n_4 + \left| \log_2 \left(2^S (K-1)^{S-M} \right) \right|$$

Lattice entropy coding for transform coefficients within an audio codec

- The MDCT transform coefficients are quantized with lattice vector quantizers.
- The coefficients are grouped in scale factor bands corresponding to a perceptual model.
- Each band is quantized with a lattice vector quantizer (dimension 4, 8, 12, 16, 20, 24).
- The lattice codevectors for each band are product code indexed and entropy coded.

Coding of transform coefficients for audio signals

File	BS32[%]	BS48[%]
es01	6.60	4.75
es02	8.00	5.62
es03	8.80	6.00
sc01	11.20	7.00
sc02	7.20	5.25
sc03	4.80	3.62
si01	5.40	2.75
si02	9.00	7.00
si03	10.60	6.75
sm01	7.40	4.00
sm02	8.40	5.25
sm03	4.80	3.25

Table 1. Bit rate savings (wrt. to enumerative coding) when the number of significant values is entropy encoded.

File	BS32[%]	BS48[%]
es01	70.20	53.12
es02	37.80	30.12
es03	49.20	36.62
sc01	34.80	21.75
sc02	60.20	48.75
sc03	76.60	63.87
si01	-21.80	-6.25
si02	-19.80	-0.75
si03	14.00	9.62
sm01	35.40	28.12
sm02	-0.80	-0.50
sm03	46.20	40.37

Table 2. Bit rate savings (wrt. to enumerative coding) when the number of significant values, the number of maximum valued components, and their position indexes are entropy encoded.

Coding of transform coefficients for audio signals

File	BS32[%]	BS48[%]
es01	6.60	4.75
es02	8.00	5.62
es03	8.80	6.00
sc01	11.20	7.00
sc02	7.20	5.25
sc03	4.80	3.62
si01	5.40	2.75
si02	9.00	7.00
si03	10.60	6.75
sm01	7.40	4.00
sm02	8.40	5.25
sm03	4.80	3.25

Table 1. Bit rate savings (wrt. to enumerative coding) when the number of significant values is entropy encoded.

File	BS32[%]	BS48[%]
es01	70.20	53.12
es02	37.80	30.12
es03	49.20	36.62
sc01	34.80	21.75
sc02	60.20	48.75
sc03	76.60	63.87
si01	-21.80	-6.25
si02	-19.80	[illegible]
si03	14.00	[illegible]
sm01	35.40	[illegible]
sm02	-0.80	[illegible]
sm03	46.20	[illegible]

30% bit rate reduction

Table 2. Bit rate savings (wrt. to enumerative coding) when the number of significant values, the number of maximum valued components, and their position indexes are entropy encoded.

References:

- Adriana Vasilache, "Indexing of lattice codevectors applied to error resilient audio coding," Proceedings of 30th AES International Conference, Saariselkä, Finland, March 15–17, 2007.
- Adriana Vasilache, "Entropic encoding of lattice codevectors based on product code indexing," in *Proceedings of EUSIPCO-2008.*, Lausanne, Switzerland, August 25-29, 2008.
- Stefan Bruhn, Volodya Grancharov, W. Bastiaan Kleijn, Janusz Klejsa, Minyue Li, Jan Plasberg, Harald Pobloth, Stephane Ragot, Adriana Vasilache, "The FlexCode Speech and Audio Coding Approach," ITG Fachtagung Sprachkommunikation, Aachen, October, 2008.