

# MDCT AUDIO CODING WITH PULSE VECTOR QUANTIZERS

Jonas Svedberg, Volodya Grancharov, Sigurdur Sverrisson, Erik Norvell, Tomas Toftgård, Harald Pobloth, and Stefan Bruhn

SMN, Ericsson Research, Ericsson AB  
164 80, Stockholm, Sweden

## ABSTRACT

This paper describes a novel audio coding algorithm that is a building block in the recently standardized 3GPP EVS codec [1]. The presented scheme operates in the Modified Discrete Cosine Transform (MDCT) domain and deploys a Split-PVQ pulse coding quantizer, a noise-fill, and a gain control optimized for the quantizer's properties. A complexity analysis in terms of WMOPS is presented to illustrate that the proposed Split-PVQ concept and dynamic range optimized MPVQ-indexing are suitable for real-time audio coding. Test results from formal MOS subjective evaluations and objective performance figures are presented to illustrate the competitiveness of the proposed algorithm.

*Index Terms*— Audio Coding, MDCT, PVQ, Noise-Fill, EVS

## 1. INTRODUCTION

In the art of high quality speech and audio coding there is a need to efficiently quantize the residual signals after an initial dynamic compression (or prediction) step. At high rates a scalar quantizer (SQ) followed by entropy coding has been in widespread use [2], while at medium rates trained codebooks [3] or lattice vector quantizers (LVQs) [4] have been in use. The main disadvantage with trained codebooks is that they do not scale well to higher rates in terms of storage and search complexity. Scalar quantization with entropy coding has the drawback that the entropy statistics needs to be stored and also trained on the correct material, and the output is by nature variable rate (requiring a rate control loop). For low rate speech coding fixed rate sparse, multi-pulse and phase structured pulse vector quantizers have been in successful use since the early 90's [5]. At low rates they have provided a good trade-off between performance and implementation complexity in terms of both storage and cycles, however without an extendable structure, they have been difficult to apply to higher bit rates. An alternative to pulse vector quantizers at higher rates are lattice quantizers, such as  $D_8$  [6] and rotated  $E_8$  [4]. Due to their regular structure they have very good search properties. However, the handling of lattice outliers and the task of indexing a point in an extended lattice and reconstructing the point from the received index may be cumbersome. Further lattice quantizers require entropy coding to provide good performance [7], and they have a dimensional rigidity, using fixed vector lengths such as 8 or 24 [8].

The Pyramid Vector Quantizer (PVQ) is a structured pulse vector quantizer introduced in the 80's by Fischer [9] [10] (for Laplacian sources). The pyramid structure enables an efficient search, and by limiting the indexing problem to smaller controlled

size sub-sections in [11], [12] and [13], it has been shown that it may give a good tradeoff between performance and complexity.

When employed in a transform domain codec, the energy focusing property of a pulse based quantizer may at lower rates result in uncovering perceptual artifacts, such as local energy losses, which need to be addressed in a post-processing step. In this paper we first give an overview of the MDCT based codec, followed by a presentation of novel modules used to quantize and post-process the band normalized transform coefficients using a PVQ. Finally, to demonstrate the performance of the introduced Split-PVQ concept and the pulse coding optimized noise-fill, subjective and objective measurements are presented.

## 2. CODEC OVERVIEW

The proposed MDCT audio codec is operating along the lines of the ITU-T G.719 framework [6]. As a coding mode in the EVS codec, it handles sample rates of 16/32/48 kHz (WB/SWB/FB) and is used at bit rates of 24.4, 32 and 64 kbps. An overview of the system is depicted in Figure 1. Similar to the G.719 scheme it operates on overlapping input audio frames  $x(t)$  but here a reduced algorithmic delay is achieved by using an alternative asymmetric window [14] which has a smaller overlap compared to the standard sinusoid window. The windowed, time-aliased input  $\tilde{x}(t)$  is transformed to MDCT[15] coefficients  $y(k)$  using  $DCT_{IV}$ :

$$y(k) = \sum_{n=0}^{L-1} \tilde{x}(t) \cos \left[ (t+1/2)(k+1/2) \frac{\pi}{L} \right], \quad k = 0, \dots, L-1. \quad (1)$$

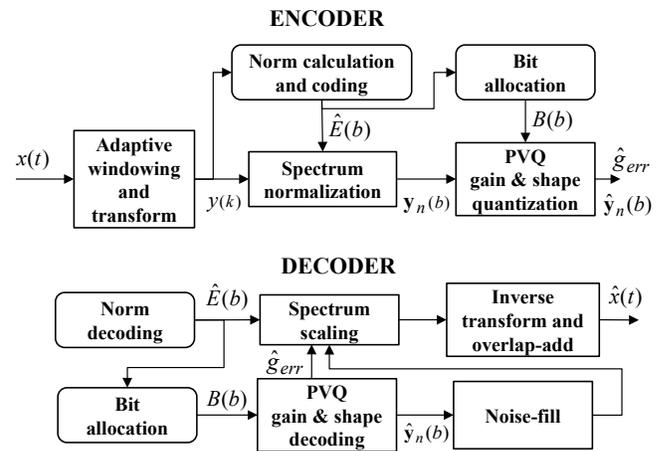


Figure 1: Overview of the MDCT based codec system

The MDCT spectrum is partitioned into bands, where each band  $b$  is associated with a norm factor  $E(b)$ , a quantized norm  $\hat{E}(b)$  and a norm index  $I(b)$ . The set of  $I(b)$  are input to a bit distribution algorithm to produce a bit allocation  $B(b)$ . Each band is normalized using the quantized norm  $\mathbf{y}_n(b) = \mathbf{y}(b) / \hat{E}(b)$ , with  $\mathbf{y}(b) = [y(s_b), y(s_b + 1), \dots, y(e_b)]^T$ , where  $s_b$  and  $e_b$  denote the start and end indices of band  $b$ . The band structure for 48 kHz is the same as in [6], but a complete list of band structures can be found in [14]. The normalized bands are then encoded using the PVQ described in section 3. Additional fine gain adjustments are done for each band as outlined in section 4. The decoder may reconstruct the encoded spectrum from the encoded parameters by reversing the steps of the encoder. The bands with zero allocated bits  $B(b) = 0$  are filled using methods described in section 5.

### 3. QUANTIZER

The normalized MDCT-coefficients in each band are quantized using a Split-PVQ concept. Bands where the allocated bits per band (excluding bits for fine gain coding)  $R(b)$  are less than a threshold are directly quantized into an energy normalized PVQ vector  $\hat{\mathbf{y}}_n$ . Bands with a higher number of allocated bits are first split into smaller target segments using an adaptive split methodology and then each smaller segment is coded. In both cases the resulting codewords are subsequently encoded by a range encoder [16] supporting both uniform and tailor made probability density functions (PDFs).

#### 3.1. Split-PVQ Methodology

Pulse position coding often results in large codeword indices, especially for long input vectors, due to the rapidly increasing number of combinations for large dimensions. For low complexity implementations a process of distributing the available bits between shorter sub segments can be used. Earlier examples of binary recursive splitting approaches can be found in [11] and in the indexing of [12]. An issue with those approaches is that the segmentation may result in a split vector with very different segment sizes. This may result in very inefficient positional coding. To demonstrate the inefficiency of non-uniform segmentation, consider a split of a 16 dimensional vector in two ways: A) symmetric (8+8) and B) asymmetric (2+14). Assume there are 2 pulses to code in each segment, then not considering overlap and sign, the number of codewords in each segment is  $N! / K!(N - K)!$ , where  $N$  is the segment dimension and  $K$  is the number of unit pulses. This results in A)  $28+28 = 56$  combinations, and B)  $1+91 = 92$  combinations, which motivates the usage of more symmetric segmentations. In the presented scheme, an algorithm for band splitting is activated when the bit budget for a particular band  $R(b)$  exceeds a pre-determined threshold. The input vector  $\mathbf{y}_n(b)$  is split into uniform (or close to uniform) segments  $\mathbf{y}_{seg}$  in a non-recursive way. The number of segments is calculated as:

$$N_p = \left\lceil R(b) / (R_{segmax} + R_{splitcost}) \right\rceil, \quad (2)$$

where  $R_{segmax} = 32$  is the maximum allowed number of bits for a segment and  $R_{splitcost}$  is an experimentally determined split

overhead cost. If the number of allocated bits for each segment is close to  $R_{segmax}$  and the variance of the segment energies is high, the number of splits is increased by one, allowing for higher flexibility in the segment bit allocation.

Each segment's shape is individually quantized and energy ratios describe the relation between the segment energies. A difficulty in this approach comes from the fact that energy variations between different segments might be large, which is unfavorable for the coding of relative energies. This is managed by calculating angles representing energy ratios between a left and right group of segments, with as equal number of segments as possible, using a recursive top-down approach. At level  $l$  of the recursion the angle is determined as  $\alpha^l = \arctan\left(\sqrt{E_R^l / E_L^l}\right)$ ,

where  $E_L^l$  and  $E_R^l$  are the energies of the left and right groups. In case of even  $N_p$ , the first level angle will represent the energy ratio between a left and right group with an equal number of segments. At each step of the recursion additional angles within each group are determined until the left and the right groups consist of one segment each. In case of an odd number of splits, the angle calculated in the first iteration will have a larger number of segments in the right group than in the left.

The angles are additionally used to recursively distribute bits to the already determined segments. At each level of recursion the bits  $R_L^l$  and  $R_R^l$  allocated to the left and the right groups are determined from the available number of bits  $R^l$  (after angle coding), the lengths of the groups  $L_L^l$  and  $L_R^l$  (number of coefficients), and the angle  $\alpha^l$  by:

$$R_L^l = \frac{R^l - \log_2(\tan(\alpha^l)) \cdot L_R^l}{1 + L_R^l / L_L^l}, \quad R_R^l = R^l - R_L^l. \quad (3)$$

E.g. if the allocated bits for the PVQ  $R(b)$  require a split into three segments, the angle calculation and bit distribution is done as follows:

**Step 0:** The angle  $\alpha^0$  is used to distribute the bits  $R^0$ , i.e.  $R(b)$  minus the bits used to code  $\alpha^0$ , to the left and the right groups of segments ( $R_L^0$  and  $R_R^0$ ). The left group includes just one segment while the right group includes two segments.

**Step 1:** The right group from **Step 0** is split in two groups where the angle  $\alpha_R^1$  is used to distribute the remaining bits of  $R_R^0$  (after coding of  $\alpha_R^1$ ) to  $R_{RL}^1$  (the second segment),  $R_{RR}^1$  (the third segment). The bits  $R_L^0$  are directly used to code the first segment.

#### 3.2. Normalization and Shape Search

For quantization of each split segment  $\mathbf{y}_{seg}$  we employ a unit hyper-sphere normalization approach similar to the concept in [8], however now applied in the PVQ-context (as in [12]). The quantized split segment  $\hat{\mathbf{y}}_{seg}(g_{seg}, N, K)$  is defined as:

$$\hat{\mathbf{y}}_{seg}(g_{seg}, N, K) = g_{seg} \frac{\mathbf{z}_{N,K}}{\sqrt{\mathbf{z}_{N,K}^T \mathbf{z}_{N,K}}}, \quad (4)$$

$$\mathbf{z}_{N,K} = \left\{ e : \sum_{i=0}^{N-1} |e_i| = K \right\}, \quad N = 1, \dots, 64, \quad K = 1, \dots, 512, \quad (5)$$

where  $\mathbf{z}_{N,K}$  is an integer point on the surface of an  $N$ -dimensional hyper-pyramid such that the L1 norm of  $\mathbf{z}_{N,K}$  is  $K$ , and  $g_{seg}$  is the segment gain ( $\cos(\alpha)$  or  $\sin(\alpha)$  at the lowest split level). The goal of the PVQ search is to minimize the shape error between the split segments  $\mathbf{y}_{seg}$  and  $\hat{\mathbf{y}}_{seg}$ . The search is performed by first using a projection [9] to a lower sub-pyramid, followed by an iterative approach (as described in [12]) of evaluating the addition of a single unit pulse to each dimension until the target pyramid is reached. Further, the search keeps track of the previous pyramid's maximum accumulated amplitude, to speed up the innermost loop in a fixed point implementation.

### 3.3. Split-PVQ Codeword Encoding

For each split angle  $\alpha$ , angle bits are allocated based on the total number of bits allocated for the two groups of segments and the length  $L_L + L_R$ . The angles are uniformly quantized and then encoded by the range encoder. Efficient coding is achieved by modelling the angle statistics using an asymmetric triangular PDF, where the asymmetry is given by the ratio  $L_L / L_R$ .

The  $\mathbf{z}_{N,K}$  vector is then indexed using a method that has similarities to both the product code enumeration in [11] and the optimized Fischer recursion in [12]. Two shape codewords are composed as follows: a first codeword representing the first sign encountered in the vector independent of its position; a second codeword representing (in a recursive fashion) all the remaining pulses in the remaining vector which is now guaranteed to have a leading positive pulse. The second codeword is enumerated using the structure displayed in Table 1. The recursive structure defines the  $U(N,K)$  offset matrix and enables the recursion computations to stay within the  $B-1$  dynamics of a  $B$  bits signed integer.

Lead value	Section size	Section definition
$K$	1	The all pulses consumed case; zeroes in remaining dimensions
$K-1$	$2 \cdot U(N,K)$	All initial pulse amplitude cases with a subsequent new leading sign, (positive or negative).
$\vdots$		
1		
0	$N_{MPVQ}(N-1,K)$	The no initial pulse consumed cases; the current leading sign is kept for the next dimension.
$\vdots$		
0		

**Table 1: Modular-PVQ (MPVQ) indexing structure**

From Table 1 we find that the total number of entries (with the very first leading sign information removed) can be expressed as:

$$N_{MPVQ}(N,K) = 1 + 2 \cdot U(N,K) + N_{MPVQ}(N-1,K) \quad (6)$$

Combining (6) with Fischer's PVQ-recursion in [9], we find that the total number of entries can be expressed as:

$$N_{MPVQ}(N,K) = 1 + U(N,K) + U(N,K+1) \quad (7)$$

Runtime computed values of the  $U(N,K)$  matrix may now be used as the basis for the MPVQ-indexing and the update of the symmetric  $U$  matrix from row  $N-1$  to row  $N$  can be performed as:

$$U(N,K+1) = 1 + U(N-1,K) + U(N-1,K+1) + U(N,K), \quad (8)$$

with initial conditions,  $U(N,0) = U(N,1) = U(0,K) = U(1,K) = 0$ .

The two short codewords of size 1 and size  $B-1$  are finally sent to the range encoder using a uniform PDF.

## 4. GAIN CONTROL

The Split-PVQ is a pure shape quantizer and a fine gain is needed to scale the quantized shape to the correct level. To determine the fine gain the allocated bits  $B(b)$  are split into bits for the PVQ quantizer  $R(b)$  and bits for a fine gain quantizer  $R_{FG}(b)$  according to:

$$\begin{cases} R_{FG}(b) &= \text{bits}_{FG}(R_{samp}(b)) \\ R_{samp}(b) &= m \times (\lfloor B(b) / L_b \rfloor, 7) \\ R(b) &= B(b) - R_{FG}(b) \end{cases}, \quad (9)$$

where  $\text{bits}_{FG}(R_{samp})$  is a lookup-table for fine gain bits. After quantizing the shape vector, a fine gain prediction is made using an accuracy measure  $a(b)$  based on the bandwidth  $L_b$ , the total number of pulses used in the Split-PVQ encoding  $N_{pulses}(b)$  and the highest identified integer pulse amplitude  $P_{max}(b)$  in the set of encoded shape vectors  $\mathbf{z}_{N,K}$ , forming  $\hat{\mathbf{y}}_n(b)$ , according to:

$$\begin{cases} g_{pred}(b) = 1 - 0.05 / a(b) \\ a(b) = N_{pulses}(b) / L_b \cdot P_{max}(b). \end{cases} \quad (10)$$

For bands where  $R_{FG}(b) > 0$ , a fine gain prediction error is computed as  $g_{err}(b) = g_{fg}(b) / g_{pred}(b)$ , where the fine gain is  $g_{fg}(b) = \mathbf{y}_n(b)^T \hat{\mathbf{y}}_n(b) \cdot (\hat{\mathbf{y}}_n(b)^T \hat{\mathbf{y}}_n(b))^{-1}$ , and  $\hat{\mathbf{y}}_n(b)$  has been normalized to unit root mean square (RMS). The gain prediction error  $g_{err}(b)$  is encoded with a non-uniform scalar quantizer in logarithmic domain using  $R_{FG}(b)$  bits. The decoder reconstructs the fine gain adjustment through:

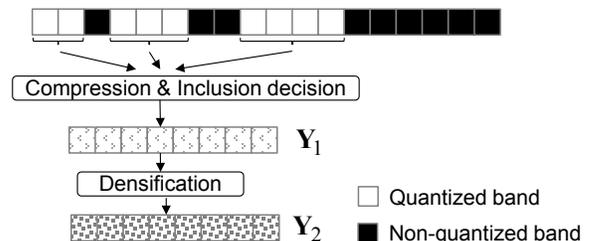
$$\hat{g}_{fg}(b) = \hat{g}_{err}(b) \cdot g_{pred}(b), \quad (11)$$

with  $\hat{g}_{err}(b) = 1$  when  $R_{FG}(b) = 0$ . The final scaling of the reconstructed band is obtained as:

$$\hat{\mathbf{y}}(b) = \hat{\mathbf{y}}_n(b) \cdot \hat{E}(b) \cdot \hat{g}_{fg}(b). \quad (12)$$

## 5. NOISE-FILL

The use of a pulse based VQ may give a peaky and sparse fine structure which may be less suitable for noise-like signals. The main spectral filling concept is to use the encoded fine structure to fill the unencoded bands. This way a similarity to the surrounding fine structure is preserved. In case of low spectral stability at lower SWB rates, a two-stage densification procedure is used, as illustrated in Figure 2.



**Figure 2: Dual spectral fill codebook generation**

A compression of the coded residual vectors is performed as:

$$\hat{y}_{n,comp}(k) = \begin{cases} \text{sign}(\hat{y}_n(k)), & \hat{y}_n \neq 0 \\ 0, & \hat{y}_n = 0, \end{cases} \quad (13)$$

where  $\hat{y}_n(k)$  refers to elements in  $\hat{y}_n(b)$ . The virtual codebook which constitutes the spectral codebook is built from only the non-sparse sub-vectors of  $\hat{y}_{n,comp}(k)$  where each sub-vector has a length of 8. Compressed sub-vectors that do not fulfill a sparsity criterion:

$$\sum_{k=s_b+8i}^{s_b+8i+7} |\hat{y}_{n,comp}(k)| \geq 2, \quad i = 0, \dots, \frac{L_b}{8}, \quad (14)$$

are rejected. The remaining compressed sub-vectors are concatenated into codebook  $\mathbf{Y}_1(k)$ , of length  $L_Y$ . For a second codebook  $\mathbf{Y}_2(k)$ ,  $k = 0, \dots, L_Y$  a densification process is used:

$$\mathbf{Y}_2(k) = \begin{cases} \text{sign}(\mathbf{Y}_1(k)) \times (|\mathbf{Y}_1(k)| + |\mathbf{Y}_1(L_Y - k)|), & \mathbf{Y}_1(k) \neq 0 \\ |\mathbf{Y}_1(L_Y - k)|, & \mathbf{Y}_1(k) = 0. \end{cases} \quad (15)$$

Figure 3 depicts the filling procedure, where  $\mathbf{Y}_1(k)$  is used below the crossover frequency  $f_{cb} = 4.8$  kHz and  $\mathbf{Y}_2(k)$  is used above. The spectral fill above the last encoded band is handled the same way as in [6] for 64 kbps, while for 24.4 and 32 kbps there are complementary spectral fill methods in use, as described in [14].

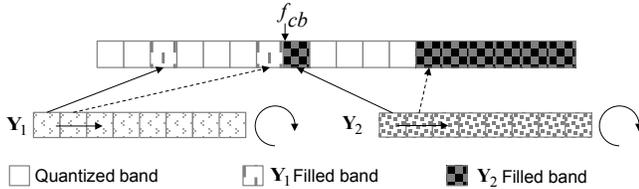


Figure 3: Spectral filling from two codebooks

## 6. PERFORMANCE

### 6.1. Objective Performance

Figure 4 displays a SNR-comparison for mixed and music content in relation to an  $\text{RE}_8 + \text{D}_8$  based lattice quantizer [6]. The evaluated Split-PVQ includes both the shape and the fine gain as the lattice quantizer captures both shape and gain of the quantized vectors.

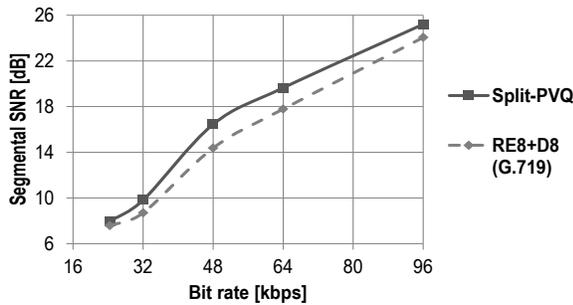


Figure 4: Segmental SNR (per frame) vs. bit rate

In Figure 5, the complexity in terms of fixed point WMOPS [17] for the described encoder side quantization algorithms is shown.

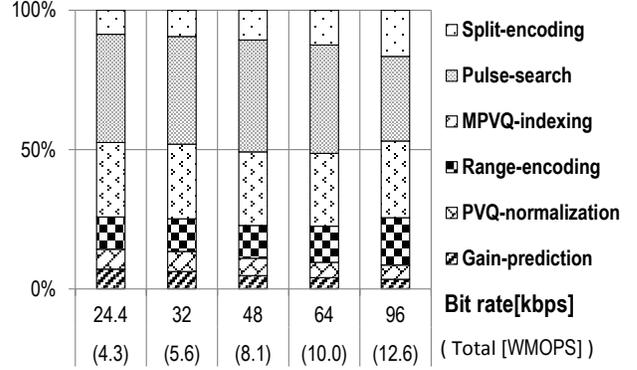


Figure 5: Encoding WMOPS vs. bit rate

At the decoder side the MPVQ de-indexing complexity is on average 25% heavier than the indexing, and the range decoder complexity is roughly 150% of the range encoding complexity.

### 6.2. Subjective Evaluation

In addition to the EVS selection tests [18], numerous tests by the proponent companies have been performed. The listening environment, levels, etc. in the proponents' test are described in [19]. The performance of the technologies described in this paper is exemplified by the performance of the EVS codec for WB mixed and music content. Table 2 below gives average results for two P.800 DCR [20] subjective tests of the EVS codec performed by the proponents for WB mixed and music content with 20 listeners in each test. In addition, for SWB mixed and music content, the proponents observed results that are not worse than the original for 64 kbps. For 32 kbps some tests revealed statistical significant improvements compared to G.719@32kbps while other tests only concluded an increase of the average MOS score within the confidence region. It should be noted that for SWB at rates 24.4 and 32 kbps a larger portion of the band is subject to spectral fill using methods which are outside the scope of this paper. Hence, the WB results are deemed more relevant for the techniques described here.

Bit rate	MOS score	
32 kbps	G.722.1	EVS
	4.18	4.42
64 kbps	G.722	EVS
	4.20	4.54

Table 2: Subjective MOS scores (WB P.800 DCR)

## 7. CONCLUSIONS

The objective and subjective experiments confirm that the proposed novel MDCT-codec design is superior to several legacy coding schemes. The complexity of the introduced Split-PVQ concept, the dynamic range optimized MPVQ indexing, and the efficient fine gain prediction allows their usage for real-time coding applications. The results from formal MOS subjective evaluations indicate that the designed system including the novel noise-fill strategy, tailored for pulse-coding schemes, provides a significant perceptual improvement.

## 8. REFERENCES

- [1] 3GPP TS 26.441, "EVS Codec General Overview", 2014
- [2] 3GPP TS 26.401, "Enhanced aacPlus general audio codec General Description", 2004
- [3] K. Ikeda, T. Moriya, N. Iwakami, S. Miki, "A design of Twin VQ audio-codec for personal communication systems", IEEE ICUPC 1995, pp 803-807, 1995
- [4] J. Makinen, B. Bessette, S. Bruhn, P. Ojala, R. Salami, A. Taleb, "AMR-WB+: a new audio coding standard for 3rd generation mobile audio services", ICASSP 2005, pp. ii/1109- ii/1112 , Vol. 2, 2005
- [5] R. Salami, C. LaFlamme, J-P. Adoul, D. Massaloux, "A Toll Quality 8 Kb/s Speech Codec for the Personal Communications System (PCS)", IEEE Trans. on Vehicular Technology, Vol.43, NO. 3, August 1994
- [6] ITU-T Rec. G.719, "Low-complexity full-band audio coding for high-quality conversational applications", 2008
- [7] A. Gerso, R. Gray, "Vector Quantization and Signal Compression", Kluwer Academic Publishers, 1992
- [8] J.P. Adoul, C. Lamblin, A. Leguader, "Baseband speech Coding at 2400bps using 'Spherical Vector Quantization'", ICASSP 1984, pp. 45-48, Vol. 9, 1984
- [9] T.R. Fischer, "A Vector Laplacian Quantizer", Proceedings 21<sup>st</sup> annual Allerton Conf. on Comm., Control and Computing, October 5-7, pp 501-510, 1983
- [10] T. Fischer, "A Pyramid Vector Quantizer," IEEE Trans. Inf. Theory, vol IT-32, NO. 4, 1986
- [11] A.C. Hung, T.H-Y Meng, "Error Resilient Pyramid Vector Quantization for Image Compression", in Proc. IEEE Int. Conf. Image Processing, 1994, pp 583-587, 1994
- [12] J.-M. Valin, T. B. Terriberry, G. Maxwell, "A Full-Bandwidth Audio Codec with Low Complexity and Very Low Delay", Proc. EUSIPCO, 2009
- [13] J-M. Valin, K. Vos , T. Terriberry, "Definition of the Opus Audio Codec", IETF/RFC6716, September 2012
- [14] 3GPP TS 26.445, "EVS Codec Detailed Algorithmic Description", 2014
- [15] H. S. Malvar, "Signal Processing with Lapped Transforms", Norwood, MA, Artech House, 1992
- [16] G. Nigel, N. Martin, "Range encoding: An algorithm for removing redundancy from a digitized message", Video & Data Recording Conf., 1979
- [17] ITU-T Rec. G.191, "Software tools for speech and audio coding standardization", "BASOP: ITU-T Basic Operators", 2009
- [18] 3GPP Tdoc S4-141065, "Report of the Global Analysis Lab for the EVS Selection Phase", Dynastat Inc., 2014
- [19] 3GPP Tdoc AHEVS-311, "EVS Permanent Document EVS-8b: Test plans for selection phase including lab task specification", 2014
- [20] ITU-T Rec. P.800, "Methods for Subjective Determination of Transmission Quality", 1996