HARMONIC VECTOR QUANTIZATION

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

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

ABSTRACT

Audio coding of harmonic signals is a challenging task for conventional MDCT coding schemes. In this paper we introduce a novel algorithm for improved transform coding of harmonic audio. The algorithm does not deploy the conventional scheme of splitting the input signal into a spectrum envelope and a residual, but models the spectral peak regions. Test results indicate that the presented algorithm outperforms the conventional coding concept.

Index Terms— *Audio coding, MDCT, VQ*

1. INTRODUCTION

Transform coding is one main technology used to compress and transmit audio signals. Typically the Modified Discrete Cosine Transform (MDCT) [3] is used and the vector of MDCT coefficients from one frame of audio is split into multiple bands of pre-defined width. Then the energy in each band is calculated and quantized. These energies are used to produce a residual vector by scaling down the MDCT vector. Unfortunately, this commonly used concept does not work well for transform coding of harmonic audio signals, e.g. single instruments. The reason is that normalization with band energies does not result in sufficiently "flat" residual vectors, and the residual coding scheme cannot represent the dynamics in the residual vector. Moreover, modeling the pitch in the MDCT domain is a cumbersome task due to the mix of time- and frequency domain information.

The presented algorithm provides an alternative audio coding model that can efficiently process harmonic audio signals. The main concept is that the MDCT vector of low-frequencies (LF) is not split in envelope and residual, but instead spectral peaks are directly extracted and quantized together with neighboring MDCT bins. Low energy regions, outside the spectral peaks neighborhood, are not coded but noise-filled at the decoder. In this way, the conventional coding model: "spectrum envelope and residual" is replaced with the concept of: "spectral peaks and noise-floor".

2. MDCT FRAMEWORK

The presented Harmonic Vector Quantization (HVQ) algorithm is integrated into the harmonic part of the EVS codec [1] to deal with stationary harmonic content. It is targeted for the sample rates of 32 and 48 kHz, and bit rates of 24.4 and 32 kb/s. Similar to the framework in [2], it operates on overlapping input audio frames $x(n)$ but here it uses an alternative asymmetric window [1], which has a smaller overlap compared to the standard sinusoid window and hence a reduced algorithmic delay. The windowed, time-aliased input $\tilde{x}(n)$ is transformed to MDCT coefficients $X(k)$ using $\mathrm{DCT}_{\mathrm{IV}}$:

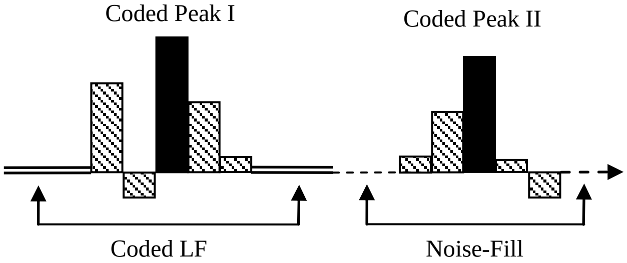

Figure 1. *Example of a MDCT vector according to the HVQ model. The two spectral peaks and the corresponding surrounding coefficients are accurately coded. The remaining bits are allocated to LF content, and the rest of the vector is noise-filled.*

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

Here K is the size of the MDCT vector ($K=640$ for 32 kHz and $K=960$ for 48 kHz sampled input) and k is the transform coefficient index. The lower 224 coefficients (up to 5.6 kHz), for 24.4 kb/s, and 320 coefficients (up to 8 kHz), for 32 kb/s, are coded by the introduced HVQ scheme. This lower spectrum range is encoded without a band structure, but relies on identification of spectral peaks to be encoded. The high-frequency (HF) range of the MDCT spectrum is partitioned into bands, where each band b is associated with a norm factor $N(b)$. These norm factors are quantized and transmitted to the decoder [1]. The decoding method includes spectral filling which populates the non-coded LF parts of the spectrum, as well as the entire HF part, which is scaled up with the set of norm factors.

3. HVQ

The signal model used in HVQ, to code the LF transform coefficients, is illustrated in Figure 1. The MDCT vector corresponding to a particular audio signal frame is assumed to be composed of prominent spectral peaks. These peaks and their surroundings are accurately coded, as described in sections 3.2 and 3.3. The number of peaks and consequently the number of bits used to code the peak regions vary with time. The remaining bits not used for peak coding are used to directly code non-peak LF MDCT coefficients, as low frequencies are of higher perceptual importance. This is described in section 3.4. The remaining parts of the MDCT vector are noise-filled, as illustrated in section 3.5. Thus, the major algorithmic steps at the HVQ encoder are: detect and code spectral peaks regions, code LF spectral coefficients (the size of the coded region depends on the number of remaining bits after peak coding), code noise-floor gains for spectral coefficients outside the peaks regions and code HF norm factors to be used with the HF noise-fill.

3.1. Classification

As the HVQ concept is applied on harmonic signals, the first essential algorithmic step is signal classification, i.e. to decide on the activation of the harmonic mode. Since that decision is dependent on the spectral peak structure, the peak selection is performed at the same step. First, the instantaneous noise-level $E_{ne}(k)$ and peak-level $E_{pe}(k)$ are estimated from the absolute values of the transform coefficients $|X(k)|$. The noise-level is calculated as:

$$E_{ne}(k) = \alpha E_{ne}(k-1) + (1-\alpha)|X(k)|, \qquad k = 0,\cdots,L-1, \quad (2)$$

with $L = 224$ at 24.4 kb/s and $L = 320$at 32 kb/s and where

$$\alpha = \begin{cases} 0.9578 & \text{if } |X(k)| > E_{ne}(k-1) \\ 0.6472 & \text{otherwise} \end{cases}. \qquad (3)$$

Similarly, the peak-level is calculated as:

$$E_{pe}(k) = \beta E_{pe}(k-1) + (1-\beta)|X(k)|, \qquad k = 0,\cdots,L-1, \quad (4)$$

where

$$\beta = \begin{cases} 0.4223 & \text{if } |X(k)| > E_{pe}(k-1) \\ 0.8029 & \text{otherwise} \end{cases}, \qquad (5)$$

and both $E_{ne}(-1)$ and $E_{pe}(-1)$ are initialized to 800. Per-band averages of noise-level $\overline{E}_{ne}(b)$ and peak-level $\overline{E}_{pe}(b)$ are calculated by averaging the corresponding instantaneous level in bands of 32 bins. The number of bands is $B = 7$ (5.6 kHz) at 24.4 kb/s and $B = 10$ (8 kHz) at 32 kb/s.

These noise and peak-level averages are used to derive three variables, used by the decision logic. The first variable is the number of detected peaks N_{peaks}. A threshold for selecting peak candidates is calculated as:

$$\Theta(b) = \left(\frac{\overline{E}_{pe}(b)}{\overline{E}_{ne}(b)} \right)^{0.88} \overline{E}_{pe}(b). \qquad (6)$$

Absolute values of the transform coefficients $|X(k)|$ are compared to the threshold $\Theta(b)$, the $|X(k)| > \Theta(b)$ form a vector of peak candidates. The threshold is adjusted on both sides of the peaks in the previous frame, with $\{1/\sqrt{2},\ \ 0.5,\ \ 0.25,\ \ 0.5,\ \ 1/\sqrt{2}\}$ around each peak position from the previous frame. This stabilizes the peak selection over frames.

Elements from the peaks candidate vector are extracted in decreasing order of peak amplitude, until the maximum number of peaks (17 at 24.4 kb/s and 23 for 32 kb/s) is reached. This procedure results in a set of N_{peaks} spectral peaks.

The second variable used in the classification logic is N_{sharp}, calculated as the number of bands for which $Sharp_{ne}(b) > 9$, where the measure of frequency sharpness per-band $Sharp_{ne}(b)$ is the peak to noise-floor ratio in each band, defined as:

$$Sharp_{ne}(b) = \frac{\max|X(b)|}{\overline{E}_{ne}(b)}, \qquad (7)$$

where $X(b)$ is the set of all coefficients in the band b.

The third variable D_{sharp} is calculated as:

$$D_{sharp} = \sum_{b=0}^{B-1} (Sharp_{ne}(b) - 9). \qquad (8)$$

The decision to switch to HVQ mode is based on comparing the above defined variables to the thresholds in Table 1. If $N_{sharp} > T_s$, $N_{peaks} \le T_p$, and $D_{sharp} > T_d$ the EVS codec will process the current frame in HVQ mode.

Rate	T_s	T_p	T_d
24.4 kb/s	4	20	22
32 kb/s	7	23	22

Table 1. *Thresholds for HVQ mode decision.*

3.2. Peak gains and positions

The essence of the HVQ algorithm is in explicit coding of spectral peak positions and amplitudes. The peaks amplitudes $G_p(m)$, where m is the peak index, are scalar quantized in a logarithmic domain to form the quantized peak gains $\hat{G}_p(m)$ and differentially coded by 5 bits. The codewords are additionally Huffman coded [4]. Since the HVQ is used to code harmonic signals the peak positions p_m will typically be equally spaced. However, some frames have an irregular harmonic structure or the peak picker does not select all the peaks in the harmonic structure. This leads to irregular peak locations. Therefore, two methods are used to code the peak positions efficiently.

The first method is Delta and Huffman coding. Due to the constraint that peaks cannot be closer than 2 MDCT bins (otherwise they are considered as one peak), Deltas are defined as:

$$\Delta_{m-1} = p_m - p_{m-1} - M, \qquad (9)$$

where $M = 3$. The lack of small deltas reduces the size of the Huffman table. Additionally, the largest delta in the table is $\Delta^{\max} = 51$. An alternative sparse coding scheme is used if $\Delta > \Delta^{\max}$. The reduced table size make the Huffman coding efficient.

The second method is based on the sparse coding algorithm described in [5]. The entire vector of the peak locations is coded, forming another vector, with 1 indicating peak presence and 0 no peak. This vector is then in a first layer divided into sub-vectors of length 5. The elements of the sub-vectors are OR-ed and the concatenation of the results of the OR operation for each first layer vector form another vector (the second layer). Each bit in this second layer indicates presence or absence of peaks in the 5-dim sub-vector from the first layer. In this way only the 5-dim sub-vectors from the first layer that are not indicated as all-zero by the

Sub-vector	Index
10000	000
01000	001
00100	010
00010	011
00001	100
10010	101
10001	110
01001	111

Table 2. *The set of 5-dim vectors corresponding to the possible peaks positions are indexed with 3-bit codewords.*

second layer have to be transmitted. The second layer is always transmitted. Prior to transmission the non-zero sub-vectors from the first layer are mapped to exploit the fact that peaks cannot be closer than 2 positions, and not all 5-dim vector combinations are therefore possible.

For example, the positional vector {01000, 00000, 00000, 00100}, with commas added to increase readability, is compressed to {**1001**, 001, 010}. The decoder reads from the bitstream the layer 2 vector **1001**. These 4 bits indicate that what will follow in the bitstream is a description of the 1st and the 4th group, while the 2nd and the 3rd have to be filled-in with zeroes. The peak positions for the 1st and the 4th group are indicated by 3bit indices, and extracted from Table 2.

The decision logic for selection of the peak position coding scheme operates as follows: if $\max(\Delta) > \Delta^{\max}$, then the sparse coding is selected, otherwise the coding scheme that uses fewer bits is chosen.

3.3. Vector quantization of peak regions

After the peak gain and position quantization, a neighborhood of 4 MDCT coefficients around each peak is quantized. The peak region coefficients (the peak itself and two neighboring bins on each side) are all scaled down by the quantized peak gain $\hat{G}_p(m)$. In this way the central bin is scaled to unit amplitude, while the surrounding 4 bins are normalized in relation to the central one. The shape vector $\mathbf{S}_m$ of the peak region, centered at bin k is defined as:

$$\mathbf{S}_m(k)=\frac{1}{\hat{G}_p(m)}\left(X(k-2) \quad X(k-1) \quad X(k+1) \quad X(k+2)\right). \qquad (10)$$

These shape vectors are quantized by means of a classified and structured vector quantizer (VQ) [6], with a trained codebook (CB) [7] with the 4-dim codevectors:

$$\mathbf{u}^i = \left(u^i(0) \quad u^i(1) \quad u^i(2) \quad u^i(3)\right) \qquad (11)$$

The numbers of peak regions vary over frames, which leads to large variation in complexity due to different number of CB searches. For audio codecs there are generally constraints on the computational complexity and the memory usage. A large number of CB searches would then limit the CB size and that is not desirable as the quantization error will increase.

To keep the complexity nearly constant, while achieving low quantization error, the search for each $\mathbf{S}_m$ is performed in a structured CB, with dynamically selected offset and size of the search region. The starting point for the search is determined by an initial classification of the input shape vector, while the length of the search region depends on the number of shape vectors to quantize. The CB is classified into two classes, with the centroids $\mathbf{C}^0$ and $\mathbf{C}^1$, and ordered such that the codewords $\mathbf{u}_i$ closest to $\mathbf{C}^0$ and most distant to $\mathbf{C}^1$ are in one side of the CB, while the codewords closest to $\mathbf{C}^1$ and most distant to $\mathbf{C}^0$ are clustered in the other side of the CB. Additionally it is noted that the shape vectors exhibit certain symmetries (the MDCT coefficients on both sides of the spectral peak have similar statistics). Therefore half of the codevectors can be represented by a flipped version of the other half, which reduces the memory usage as the flipped part does not have to be pre-stored. The search is performed both among the codevectors $\mathbf{u}^i$ as well as the flipped codevectors:

$$\mathbf{u}_f^i = \left(u^i(3) \quad u^i(2) \quad u^i(1) \quad u^i(0)\right) \qquad (12)$$

Thus the searched CB (both physically stored and virtual) can be seen as clustered into four classes with centroids $\mathbf{C}^0$, $\mathbf{C}^1$, $\mathbf{C}_f^0$ and $\mathbf{C}_f^1$.

Number of coded peaks	Number of vectors searched in codebook	
	24.4kb/s	**32kb/s**
23	-	128
22	-	134
21	-	141
20	-	149
19	-	158
18	-	168
17	128	179
16	136	192
15	145	206
14	155	224
13	167	244
12	181	256
11	197	⋮
10	217	⋮
9	241	⋮
8	256	⋮
⋮	⋮	⋮
1	256	256

Table 3. *Search space for 24.4 and 32 kb/s, as a function of the number of spectral peaks for the current audio frame.*

The following algorithmic steps are executed for every input shape vector $\mathbf{S}_m$; First the minimum mean squared distance between the input vector and the four centroids, $\mathbf{C}^0$, $\mathbf{C}^1$, $\mathbf{C}_f^0$ and $\mathbf{C}_f^1$ is used to determine the starting point of the search and the CB orientation. Then the search space is dynamically adjusted such that when larger numbers of peaks are to be quantized in the current frame, the search space is reduced to limit the maximum complexity according to Table 3. The maximum search space (full-search) is used with 8 peaks or less at 24.4 kb/s, and 12 peaks or less at 32 kb/s.

Since the codevectors in the CB are sorted by the distance between each codevector and the centroids, the search procedure goes first over the set of vectors that is likely to contain the best match. This property leads to minimum performance loss even though the search space is dynamically adjusted. This trade-off between complexity and performance is illustrated in section 4.1.

3.4. Allocation of remaining bits

As the number of spectral peaks varies in time, the presented coding scheme will by its nature result in a variable bitrate. Bits available after peak coding are used to directly code LF MDCT coefficients. The target coefficients $X(k)$, outside of the peak regions, are concatenated into a vector $Z(k_c)$ which is partitioned into bands of 24 coefficients. The coding always starts at the beginning of the MDCT vector, e.g. if the available bits allow coding of only one 24-dim band, then only the coefficients $Z(k_c)$, $k_c = 0,\cdots,23$ will be coded. The number of bands N_z in $Z(k_c)$ that will be coded depends on the remaining number of bits R_{avail} after peak coding, and is determined by:

$$N_z = \left\lfloor \frac{R_{avail}}{R_{max}} \right\rfloor + \max\left(\operatorname{sign}\left(\left(R_{avail} \bmod R_{max}\right) - 30\right), 0\right), \quad (13)$$

where $R_{max} = 80$ at 24.4 kb/s and $R_{max} = 95$ at 32 kb/s. The resulting vector $Z(k_c)$ is encoded with a gain-shape VQ with 5 bits per gain and PVQ encoded [8] shape vectors.

3.5. High-frequency regions and noise-fill

The non-coded coefficients below 5.6 kHz for 24.4 kb/s and 8 kHz for 32 kb/s are grouped into two sections and filled with random noise, using noise floor gains derived from the per-band noise-level averages described in section 3.1. The average of the elements of the first half of $\overline{\mathbf{E}}_{\mathbf{ne}}$ gives the noise-floor gain $G_{ne}(0)$, while the second half of average noise levels forms $G_{ne}(1)$. These norm factors are quantized and transmitted to the decoder.

The frequency part above 5.6 kHz for 24.4 kb/s and 8 kHz for 32 kb/s is reconstructed by means of coded HF gain factors $N(b)$ and spectral filling codebook generated from the coded LF part. In the process of creating the noise-fill codebook, the LF peak positions are also recorded. Exploiting that knowledge, the applied envelope gains are modified based on the presence of a peak in the fill codebook according to:

$$N_b = \begin{cases} 0.1N(b-1) + 0.8N(b) + 0.1N(b+1), & \text{peak in band } b \\ \min\left(N(b-1), N(b), N(b+1)\right), & \text{otherwise.} \end{cases} \quad (14)$$

The purpose of the gain modification is to avoid that a peak is significantly attenuated if it happens that the corresponding gain comes from a band without any peaks. Additionally, a noise-like structure in the noise-fill codebook should not be amplified by applying a strong gain that is calculated from an original band that contained one or more peaks.

4. EVALUATION

This section consists of complexity measurements confirming the efficiency of the proposed dynamic search space adjustment, as well as subjective evaluation of the HVQ coding mode by mean of an ABX test [9]. Further details on formal perceptual evaluation by means of MOS [10] tests of the EVS codec can be found in the selection tests, available at [11].

4.1. Complexity measurements

The computational complexity in this section is measured according to [12] and was obtained from a simulation over 1312 harmonic signal frames. The data contained on average of 19.4 peaks per frame with a variance of 7.1 (minimum 12 and maximum 23 peaks). As illustrated in Table 4 the use of adaptive search in the classified structured codebook gives significant complexity reduction with maintained accuracy. The adaptive search reduces the maximum complexity with 45% while the average SNR decreases by only 0.2%.

	Search	
	Full	**Adaptive**
Average SNR [dB]	8.743	8.723
Max complexity [WMOPS]	4.857	2.638

Table 4. *Average SNR and maximum computational complexity for the shape VQ with full and adaptive search.*

4.2. Subjective evaluation

Subjective evaluation of the introduced algorithm has been performed in terms of a 5 level ABX test. After being presented with the reference signal and a pair of **A** and **B** samples in random order, the listeners had to choose among the following options: **B** much better (-2), **B** slightly better (-1), **A** and **B** are the same (0), **A** slightly better (+1), **A** much better (+2). The systems under test were the EVS codec with active HVQ module (option **A)** and the EVS baseline with HVQ module deactivated (option **B)**. The test consisted of 10 harmonic items, and the votes from 7 experienced listeners are presented in Table 5. The average results are presented in Figure 2.

Vote	-2	-1	0	+1	+2
Count	3	24	42	50	21

Table 5. *Results from the subjective ABX test indicate clear preference for the EVS codec with active HVQ module (positive scores), compared to EVS with deactivated HVQ module (negative scores).*

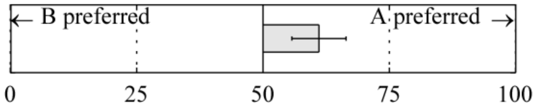

Figure 2. *Mean score and 95% confidence interval indicate statistically significant preference of HVQ module in the ABX test.*

5. CONCLUSIONS

By means of a subjective test we have demonstrated that the proposed concept of a harmonic audio coding model provides superior perceptual quality over the conventional coding scheme. Complexity measurements confirm that the introduced adaptive VQ concept provides significant complexity reduction while maintaining quantization errors at the level of exhaustive search. These subjective and objective measurements show that the HVQ improves coding of harmonic audio signals.

6. REFERENCES

[1] 3GPP TS 26.445, “Codec for Enhanced Voice Services (EVS); Detailed algorithmic description,” 2014.

[2] ITU-T Rec. G.719, “Low-complexity full-band audio coding for high-quality conversational applications,” 2008.

[3] H. S. Malvar, “Signal Processing with Lapped Transforms,” Norwood, MA: Artech House, 1992.

[4] D. Huffman, “A Method for the Construction of Minimum-Redundancy Codes,” Proc. IRE 40 (9), 1952.

[5] D. Salomon, G. Motta, and D. Bryant, “Sparse Strings” in “Handbook of Data Compression,” Springer, 2010.

[6] A. Gersho and R. Gray, “Vector Quantization and Signal Compression,” Springer, 1991.

[7] Y. Linde, A. Burzo, and R. Gray, “An algorithm for vector quantizer design,” IEEE Trans. Commun. COM-28, 1980.

[8] T. Fischer, “A Pyramid Vector quantizer,” IEEE Transact. Inf. Theory, IT-32 (4), 1986.

[9] D. Clark, “High-Resolution Subjective Testing Using a Double-Blind Comparator,” J. Audio Eng. Soc. 30 (5), 1982.

[10] ITU-T Rec. P.800, “Methods for Subjective Determination of Transmission Quality,” 1996.

[11] 3GPP Tdoc S4-141065, “Report of the Global Analysis Lab for the EVS Selection Phase,” Dynastat Inc., 2014.

[12] ITU-T Rec. G.191, “Software Tool Library 2009 User’s Manual,” Geneva, 2009.