INTERPOLATING THE HISTORY

IMPROVED EXCITATION CODING FOR HIGH QUALITY CELP CODING

Per Hedelin and Thomas Eriksson

Department of Information Theory
Chalmers University of Technology
S-41296 Göteborg, Sweden.

ABSTRACT

A procedure is presented where the conventional innovation codebook approach of CELP coding is replaced by an interpolative scheme. A generalized LTP-codebook constitutes the basis for the interpolation. A pre-requisite to efficient interpolation concerns establishing the neighboring vectors of a given LTP entry. We discuss several working approximations for establishing a suitable neighborhood concept.

By simulations we have found that the interpolative scheme leads to 17-20 bits for the excitation coding of one block. In addition to this, some 3-4 bits are required for block gain. A standard CELP typically spends 25 bits or more for one block of excitation coding. The subjective quality of our proposed coding scheme compares favorably with standard CELP. In particular, the interpolation improves the pitch-related properties giving a less noisy subjective impression.

1. INTRODUCTION

The conventional arrangement of excitation coding in CELP is based on a compromise whereby an attractive balance between performance and complexity is achieved [2]. In VQ terminology, the cascade of an adaptive LTP codebook and a fixed innovation codebook constitutes a multi-stage encoder [1]. In most applications the two stages have been searched one at a time (OAT), where, for each block, the innovation coding is investigated for a single LTP-entry only. In comparison with full search, the OAT principle constitutes a major ease in computational effort.

Future applications of speech transmission will require a wide variety of coding methods, capable of working at different bit-rates. In this work we are concerned with procedures that operate in the range 4.8 to 6.4 kbit/s. Our goal is to improve the coding efficiency by relaxing constraints on coding complexity. In other words we are looking for higher subjective quality for applications where an increase in computational burden can be tolerated.

Random, sparse, multi-pulse, vector-sum and algebraic coders are examples of innovation coding schemes that have been utilized in conjunction with CELP-coders. To a surprising degree these different architectures perform equally well, albeit they have different properties in terms of complexity. To some extent, the perceptual character of the schemes relates to the structure of the innovation codebook; a random codebook often gives a somewhat noisy impression whereas a pulse-like codebook may be perceived as "hard" or "machine-like". Common to all innovation principles mentioned above is the property that inter-harmonic noise may be a problem – in particular when the pitch period is comparable to the length of the blocks. For this reason some researchers have suggested utilizing quasi-periodic innovations.

While we essentially motivate our alternative principle for innovation coding on VQ theory (as elaborated on below) we would like to comment that an interpolative innovation scheme avoids enforcing a synthetic law for signal generation. Moreover, we can utilize quasi-periodic "innovations" in a natural way.

One ultimate arrangement of excitation coding is a single stage encoder. In comparison with a dual-stage encoder, at least three advantages apply to a strict single stage coding of CELP excitation, namely: *i)* an optimized set of vectors; *ii)* a possible search gain (if all entries are evaluated) and *iii)* a bit-reduction since the encoding of separate LTP- and innovation gains is eliminated. Achieving all three advantages is in practice difficult. Exhaustive search of codebooks in excess of say 15 bits is not feasible for many applications and, hence, fast search procedures become a necessity. However, the major problem with a strict single stage encoder concerns an optimal frame-to-frame design of an adaptive codebook.

The approach to excitation coding proposed in this paper is based on an enlarged LTP-type of codebook cascaded with a simple interpolation scheme to provide corrections to the basic vectors. The architecture can be seen as a dual stage encoder where both stages are adaptive.

An interpolation scheme yields a second-stage codebook such that each first stage vector has a unique set of correction vectors. In VQ terminology this principle is referred to as cell-conditioned coding, c.f. [6]. Cell-conditioned dual-stage coding can, in theory, operate arbitrary close to the performance of (optimal) single stage coding. While the theoretical advantage of cell-conditioned coding is indisputable, it is in practice difficult to benefit from the full potential of this principle; in particular for cases where the first stage is adaptive and, thus, time-varying. However, in comparison to a conventional dual-stage arrangement – where correction vectors are duplicated in an identical fashion across first-stage vectors – a suitable cell-conditioned approach normally outperforms conventional dual-stage coding.

In summary we exploit points *ii)* and *iii)* above, while our attempt at reaching the advantage of *i)* is suboptimal.

2. THEORY

2.1 Framework

Let the coding of the speech signal be arranged in frames of N_f samples; each frame further sub-divided in blocks of N_b samples. We accept the conventional perceptual weighting filter $W(z) = A(z)/A(z/\rho)$ in evaluating an appropriate distortion D between a speech block $\{x(n)\}$ and its encoded correspondent $\{\hat{x}(n)\}$. The filter $A(z)$ is an M:th order polynomial determined by a conventional LPC-analysis. Appropriate encoding of the LPC parameters can be achieved with 20-25 bits/frame [3, 4, 5]. In this contribution all simulations reported on below have been performed with unquantized filters $A(z)$. Moreover, for the simulations discussed below we have selected the familiar values $N_f = 160$, $N_b = 40$ and $M = 10$, c.f. [2].

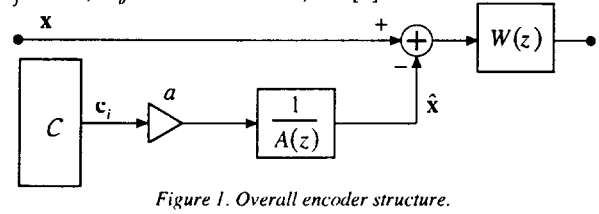

Figure 1. Overall encoder structure.

We denote the codebook of excitation vectors $\mathcal{C} = \{\mathbf{c}_i\}_{i=1}^{K}$. For each block, the synthesized signal is achieved by feeding the production filter $1/A(z)$ with a gain-scaled excitation vector $a\mathbf{c}_i$.

0-7803-2431-5/95 $4.00

The index i is selected for each block to minimize the perceptual distortion D. In our further discussion we refer to the speech block in the weighted domain by $\mathbf{x}'$ and, likewise, by $\mathbf{c}_i'$ we denote the excitation signal in the corresponding domain. In other words, $\mathbf{c}_i'$ is $\mathbf{c}_i$ filtered by $1/A(z/\rho)$. In this notation we can write the distortion D as $D = \|\mathbf{x}' - a\mathbf{c}_i'\|^2$.

2.2 Principles of the Excitation Codebook

The success of CELP-coding relies to a great extent on the efficient usage of an adaptive LTP codebook. An LTP ensures that the quasi-periodic nature of speech is exploited in the coding process. Consequently, we employ the history of coded excitation as the basis for the generation of the codebook $\mathcal{C}$. By $\mathcal{D}$ we denote a set of vectors $\{\mathbf{d}_j\}_{j=1}^{L}$ drawn by sliding a N_b samples wide window over the immediate past of the quantized excitation signal.

2.2.1 Fractions and Delay Range

In conventional CELP, L=128 vectors are drawn corresponding to a delay of 2.5 to 18.5 ms to generate $\mathcal{D}$. In several improved CELP schemes fractional delays are utilized in order to gather L=256 vectors from the same region of delay, cf. [7]. We performed several preliminary experiments concerning fractions and an extended delay range. Both methods increase the rate, and thus the performance, of the codebook $\mathcal{D}$.

Fractional delays The delay resolution was varied from 1 to 1/8 sample. SNR values after the first-stage codebook are presented in table 1 below.

Increasing the range of delay We increased the delay range to 256, 512 and 1024 samples, while keeping the lowest delay at 20 samples. The results are presented below in table 1.

In order to evaluate the performance of the first stage only – without the influence of a particular innovation mechanism –we, in the experiment listed in table 1, utilized unquantized vectors in the codebook $\mathcal{D}$.

delay resolution	Delay range (in samples)			
	128	256	512	1024
1	0	+0.33	+0.56	0.77
0.5	+0.55	+0.86	+1.07	1.27
0.25	+0.72	+1.01	+1.22	1.43
0.125	+0.79	+1.08	+1.29	1.51

Table 1. Improvement in SNR relative a standard L=128, 1-sample-resolution LTP codebook. Values are listed for different resolutions and ranges of delay. The results have been compiled for an unquantized codebook D.

The main conclusions from the experiments are:

I) Increasing the delay resolution to better than 0.5 sample gives low gains.
II) Increasing the range of delay is a competitive method.

We have also experimented with non-uniform resolutions, and found that a good compromise is to employ an increased resolution for the lower delay values only.

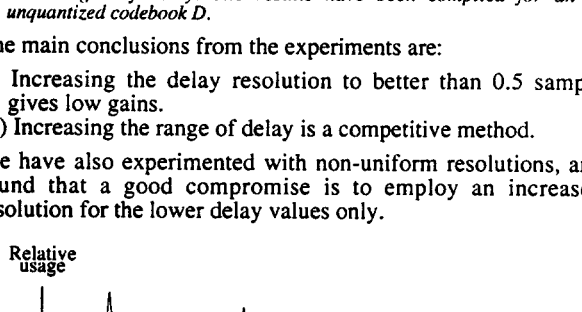

Figure 2. Relative usage of various delays for an LTP-type of codebook $\mathcal{D}$. Based on 20.000 sub-blocks of male and female speech.

Figure 2 above depicts the relative usage of various delays when the operative range is 2.5 to 80 ms divided between 737 indices employing unequally large fractions. For short delays fractions of 4 were utilized while at long delays a decimation of 4 was selected. For intermediate delays we selected intermediate values for the fractions. We stress that the figure illustrates that long delays are utilized in reasonable proportion. By a suitable fractional profile we can make all the first-stage codewords to be of comparable probability. Listening tests confirmed the objective advantage of an enlarged delay range; we did not, for instance, perceive unwanted effects due to "pitch halving".

2.2.2 Building the Basic Codebook

With the idea of interpolating in a basic codebook $\mathcal{D}$ it is not wise to build $\mathcal{D}$ on past signals only. Even with an extended delay range, the immediate past captured in $\mathcal{D}$ may during transients such as voicing transitions provide a far from optimal model of the current signal block. Hence, we have further extended the set $\mathcal{D}$ with a small amount of random as well as impulse based vectors. Our experiments indicate that on the order of 10-25% of random and impulse signals is appropriate. In several of our experiments we have employed a grand total of 1024 entries in the basic codebook $\mathcal{D}$. A typical mixture has been 700 delayed vectors, 100 impulses and 200 random waveforms. We have also successfully replaced N_b of the random vectors in $\mathcal{D}$ with the eigenvectors of the impulse response of the filter $1/A(z/\rho)$.

2.3 Establishing a Neighborhood

For the interpolation in a basic codebook elaborated on below, we are interested in establishing the neighborhood $\mathcal{N}_i$ of each entry in $\mathcal{D}$. In our scheme we interpolate from one entry to its neighbors only.

The neighborhood $\mathcal{N}_i$ is a list of indices of entries $\mathbf{d}_k$ that are close to the vector $\mathbf{d}_i$ in some sense. Since the selection of the appropriate excitation vector is performed in the weighted domain we, in fact, should look for properties in the weighted domain. The relevant measure concerns the properties of vectors $\mathbf{d}_k'$ and $\mathbf{d}_i'$; the weighted domain correspondents to $\mathbf{d}_k$ and $\mathbf{d}_i$.

We have investigated several procedures for establishing $\mathcal{N}_i$. One natural principle is to utilize the Voronoi neighbors, i.e. declaring two first stage vectors as neighbors if they share a common face in the Voronoi partition of $\mathcal{D}'$. This idea is however not really practicable since establishing the Voronoi partitioning is a computationally expensive task; requiring a computational effort several magnitudes larger than acceptable.

The concept of Gabriel neighbors is closely related to the one of Voronoi neighborhood. Two first-stage vectors are declared Gabriel neighbors if their mid-point lies on the hyperplane of the Voronoi partition of $\mathcal{D}'$ that separates the two vectors. Formally, if

$$\|(\mathbf{d}_k' + \mathbf{d}_i')/2 - \mathbf{d}_i'\|^2 \le \|(\mathbf{d}_k' + \mathbf{d}_i')/2 - \mathbf{d}_j'\|^2 \quad \forall j$$

then $\mathbf{d}_i$ and $\mathbf{d}_k$ are declared Gabriel neighbors. Consequently, a list of Gabriel neighbors always constitutes a subset of the corresponding Voronoi list. Finding a list of Gabriel neighbors is computationally expensive, but the effort is tolerable.

In addition to the geometrically oriented principles of neighborhood, we have experimented with code-oriented schemes for defining a suitable neighborhood. In particular we have employed the idea of Hamming-1 neighborhood. Two vectors $\mathbf{d}_k$ and $\mathbf{d}_i$ are declared Hamming-1 neighbors if their indices k and i differ in exactly one bit. Independent of how the set of first-stage vectors are indexed this principle leads to a symmetric structure in the sense that all entries have the same number of neighbors, namely $\log_2 L$ Moreover, one particular entry appears exactly $\log_2 L$ times in a list $\mathcal{N}_i$. To be more explicit, with a 10 bit first-stage codebook all entries have exactly 10 Hamming-1 neighbors. This symmetric property is a contrast to that of both the Voronoi and the Gabriel principle where the number of neighbors varies from vector to vector (typically involving a considerably larger number of neighbors).

In working with Hamming-1 neighbors we have normally relied on the vectorial sorting described in [5] in order to bring the Hamming distances in correspondence with the geometrical distances. The particular procedure described in [5] gives a fast and adaptive access to the neighborhood list. The overall computational complexity is proportional to $L(\log_2 L)^2$.

Finally, we have experimented with non-adaptive procedures for establishing a neighborhood. We have been surprised to find these non-adaptive schemes to be only slightly inferior to the adaptive ones described above. One simple non-adaptive technique is to use the Hamming-1 concept for the natural ordering of the first stage vectors; first-stage vectors are indexed according to their LTP-delay.

Encouraged by the nice performance of simple non-adaptive rules for neighborhood we, by trial and error, developed an ad hoc rule for ten neighbors. Here, the neighborhood is set up by two vectors at ±1 samples of delay, another two at ±3 samples of delay, one at twice the delay and so on. We refer to this principle as "best ad hoc rule" below.

2.4 Comparing Different Neighborhood Principles

The choice of neighbor principle is, obviously, of importance for the success of the interpolation. We have briefly commented on the performance of the schemes when introducing them above. Table 2 lists formal scores in terms of innovation gain for the methods in our study. In order to give all principles a fair and equal comparison, we used an unquantized codebook $\mathcal{D}$ in this experiment, i.e. all methods operated on one and the same set of first-stage vectors. We also emphasize that we here only evaluated and report the innovation gain of the winning first-stage entry, i.e. no tree search coding was applied.

Ten vectors from the (unquantized) codebook $\mathcal{D}$ were selected with the neighborhood principles listed. The performance was evaluated for a system with 10 *optimum* weights, to separate the effect of the weight quantization from the vector selection process. Since no quantization occurs, the results constitute upper bounds for the performance of the methods.

Neighbor concept	SNR [dB]
Random	3.7
Sorted Hamming-1	3.9
Natural IA Hamming-1	4.0
Best ad hoc rule	4.2
Gabriel	3.6

Table 2. SNR values of innovation gain for different sets of neighbors. Unquantized weights. Unquantized first-stage codebook.

By random in table 2 we denote a random selection of neighbors. The sorted and natural index assignment Hamming-1 principles are listed next, followed by a non-adaptive ad hoc rule (briefly discussed above). The Gabriel neighborhood is tabulated last and performs worst. Since the number of Gabriel neighbors is time-varying, this principle may suffer from our truncation of the neighborhood list to exactly 10 neighbors in all occasions.

The results of adaptive and non-adaptive schemes are, in general, close. When performing a full search with all first-stage parents, 2 dB or more of tree search gain is observed for all methods except the Gabriel approach. The Gabriel neighborhood leads to a small tree search gain, close to non-significant.

In the next section we address the problem of how to quantize the weights. We do so well aware that an interpolation employing 10 neighbors can give an innovation gain far in excess of the 2-3 dB that is required for high quality speech.

3. INTERPOLATION

The set $\mathcal{D}$ of first-stage (generalized LTP) vectors defines a Voronoi partition of the space spanned by the $\mathbf{x}'$ vectors. In extending a (close to optimal) codebook it is in general desired to place vectors in accordance with the first-stage Voronoi partition. Unless this property is met, the extension will tend to give results close to that of conventional (random) innovation coding.

Our straightforward way of extending a codebook $\mathcal{D}$ is to consider vectors $\mathbf{c}_i$ formed by a weighted sum of two or several entities in $\mathcal{D}$. In mixing two vectors a reasonable choice is $\mathbf{c}_i = \mathbf{d}_{i_1} \pm (\mathbf{d}_{i_2} - \mathbf{d}_{i_1})/2$. This generates a point half way in between $\mathbf{d}_{i_2}$ and $\mathbf{d}_{i_1}$, and a complementary point at the same distance from $\mathbf{d}_{i_1}$ in the opposite direction. If $\mathbf{d}_{i_2}$ and $\mathbf{d}_{i_1}$ are Gabriel neighbors (and are equally probable) then the half way point is not only a likely point but also locally optimal in a mean square sense. Interpreting $\mathbf{d}_{i_2} - \mathbf{d}_{i_1}$ as an innovation vector there, hence, exists a natural innovation gain of ±0.5. We may also distinguish adding and subtracting $\mathbf{d}_{i_2} - \mathbf{d}_{i_1}$ in the way that the former constitutes an interpolation, while the latter acts as an extrapolation since it can generate points outside a closely scattered set $\mathcal{D}$.

Figure 3 below illustrates the Gabriel extension rule for a 2-dimensional codebook. Here, the points $\mathbf{c}_i = \mathbf{d}_{i_1} + (\mathbf{d}_{i_2} - \mathbf{d}_{i_1})/2$ are used to extend a basic codebook – we thus only obtain interpolation. For ideal cases of optimized basic codebooks, this extension rule gives very good results; close to the one of the global optimum. The Voronoi regions of the extended codebook become congruent with the Voronoi regions of the basic codebook.

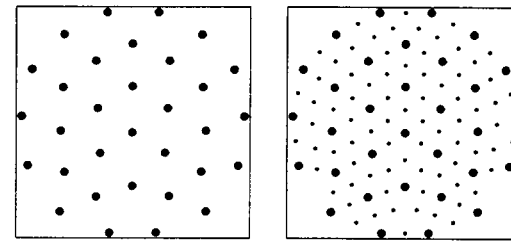

Figure 3. Two-vector interpolation; extending a codebook by the Gabriel points. Left: a basic codebook. Right: the extended codebook.

While the Gabriel points are close to optimal for an optimized first-stage codebook it is an open question if this extension rule has the same nice properties when employed for a generalized LTP codebook as the first-stage codebook. In order to test the rule we used two neighborhood principles, namely *i*) Gabriel neighbors and *ii*) Hamming-1 neighbors given the index assignment of the vectorial sort discussed above.

Depending on the speech material, a generalized LTP codebook $\mathcal{D}$ of 10-bits gives a segmental SNR of 6-7 dB in the weighted domain when cascaded with a conventional (large-sized) second-stage codebook. For both of the Hamming-1 neighborhood principles (and a set $\mathcal{N}_i$ of 10 entries) the two-vector interpolation scheme yields an innovation gain of 0.7-0.9 dB, also in the weighted domain. In terms of bits for the codebook $\mathcal{C}$, this amounts to a total of 13.3 bits. When instead using the Gabriel neighbors we obtained a similar 0.7-0.9 dB of "innovation gain". The number of Gabriel neighbors proved to be highly variable; the mean was approx. 30 neighbors. Occasional frames, however, proved to have several hundred vectors as Gabriel neighbors to the winning first-stage vector. For coding purposes, utilizing the Gabriel neighbors (or the Voronoi neighbors) is not attractive. It is not clear how to design a fixed rate coding scheme that exploits the variable nature of the number of Gabriel neighbors.

In passing we comment that a 0.7-0.9 dB of innovation gain is not sufficient for reasonable quality. In practice, the two-vector interpolation mechanism leads to a complete coder break down when used as the single innovation generator. The history of coded excitation cannot be driven by interpolating between two neighboring vectors only.

3.1 Interpolation with Several Basic Vectors

In order to work properly in extending the generalized LTP codebook $\mathcal{D}$ we have found that the interpolation must involve a

subset of at least 3, or, preferably 4 or more vectors from the neighborhood $\mathcal{N}_i$. For the four-vector case, we use

$$\mathbf{c}_i = \gamma_0 \mathbf{d}_{i_0} + \gamma_1(\mathbf{d}_{i_1} - \mathbf{d}_{i_0}) + \gamma_2(\mathbf{d}_{i_2} - \mathbf{d}_{i_0}) + \gamma_3(\mathbf{d}_{i_3} - \mathbf{d}_{i_0})$$

where $\mathbf{d}_{i_1}$, $\mathbf{d}_{i_2}$ and $\mathbf{d}_{i_3}$ are three neighbors to $\mathbf{d}_{i_0}$. Similarly for the three-vector case we use

$$\mathbf{c}_i = \gamma_0 \mathbf{d}_{i_0} + \gamma_1(\mathbf{d}_{i_1} - \mathbf{d}_{i_0}) + \gamma_2(\mathbf{d}_{i_2} - \mathbf{d}_{i_0})$$

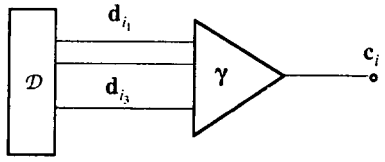

Figure 4. Three-vector interpolation – three vectors from the generalized LTP-codebook $\mathcal{D}$ are weighted to form a vector $\mathbf{c}$.

We have experimented with various designs for the set of interpolation weights $\Gamma = \{\boldsymbol{\gamma}^{(k)}\}$ where $\boldsymbol{\gamma} = (\gamma_0, \gamma_1, \gamma_2, \gamma_3)^T$ (and $\boldsymbol{\gamma} = (\gamma_0, \gamma_1, \gamma_2)^T$ for the three-vector case). We have tested and evaluated several schemes along the lines of the two-vector case. In addition we have compared these with trained weighting codebooks using the familiar LBG-algorithm. We have tested both the Gabriel neighborhood concept and the Hamming-1 concept. In all cases the Hamming-1 neighborhood proved superior in performance.

Specifying 2 neighbors from a neighborhood of 10 vectors requires $\log_2(10 \cdot 9/2) \approx 5.5$ bits whereas a specification of three neighbors from the same list requires $\log_2(10 \cdot 9 \cdot 8/6) \approx 7$ bits.

In interpolating three vectors, a logical choice is the four combinations of $\Gamma_3 = (1/3, \pm 1/3, \pm 1/3)^T$. For four vectors the corresponding 8-set is $\Gamma_4 = (1/4, \pm 1/4, \pm 1/4, \pm 1/4)^T$.

Table 3 below lists the relative usage of each of the interpolation rules. In the tabulated experiment all alternatives were available of selecting either zero, one, two or three vectors from a set of ten neighbors. This amounts to a total of 176=1+10 +45+120 subsets. The most frequently used rule is the perhaps most reasonable one, namely $\mathbf{c}_i = (\mathbf{d}_{i_0} + \mathbf{d}_{i_1} + \mathbf{d}_{i_2} + \mathbf{d}_{i_3})/4$.

Number	sign sequence of weights in γ							
of neigh.	++++	+-++	++-+	+--+	+++-	+-+-	++--	+---
0	12							
1	62	174						
2	415	501	480	144				
3	1690	1111	1056	715	840	687	524	886

Table 3. Relative usage of the interpolation rules. Based on the non-silent parts of 10.000 subblocks.

When comparing the above 176 interpolation rules with LBG-trained weighting codebooks (of equal size) we have found only small differences in performance. Hence, we conclude that the structured interpolation rules are close enough to the optimum to be used in our final design.

4. TREE-SEARCH GAIN

For an arbitrary two-stage codebook arrangement there, in general, exists a tree-search gain. Only by confining the second stage to generate vectors well within the Voronoi region of the parent vector, this tree-search gain becomes negligible.

With a cell conditioned approach, as our interpolation scheme, we have a possibility to control the proportion of the second-stage vectors that reside outside the parent Voronoi region. Interpolating with Gabriel neighbors proved to give exactly the advantage we were looking for; an OAT search gives results very close to those of a full search of the entire codebook $\mathcal{C}$. However, all other neighborhood concepts proved to have a significant tree search gain. Unfortunately, utilizing the Gabriel neighbors was inferior in terms of performance (when used in conjunction with a LTP-codebook) to all other neighborhood principles we have tested. We interpret this result in the following way. While the LTP-concept is powerful, the associated Voronoi partition is irregular enough to prevent a successful direct exploitation of its inner structure.

4.1 Search Methods

In our simulations we have employed a true equivalent of full search of all entries in the excitation codebook $\mathcal{C}$. For each first stage (parent) vector we have evaluated an upper bound for the performance of the interpolation with the particular neighbors of the parent. The leafs of the coding tree (i.e. the explicit interpolation rules) have been evaluated only for parents with a bound above the score of the best found fully quantized leaf. This principle reduced the overall computational load to 15 %. For a real-time application the computational effort is still high and a standard pruning of the tree (based on the first-stage scoring) would probably be a necessity.

5. RESULTS

At the point of writing, we have evaluated the subjective performance of the interpolative scheme in informal listening tests only. At a total of 19 bits in the extended codebook $\mathcal{C}$ (plus 3 bits of block gain) we obtain high quality speech, preferable to that of a conventional CELP with up-sampling operating at 25 bits for excitation coding. The somewhat noisy character of standard CELP is distinctly reduced with the interpolative procedure. On the tested speech material, the segmental SNR of the interpolative scheme is 0.3-0.4 dB higher than that of conventional coding.

At a total of 20 bits in the extended codebook $\mathcal{C}$ (plus 4 bits of block gain) we obtain a quality distinctly is preferable to that of a VSELP coder at 7.95 kbit/s. Table 4 below lists segmental SNR values for our interpolative scheme, referred to as cHELP (Coded History Excited Linear Prediction). For comparison we have included SNR values of the VSELP coder (at 7.95 kbit/s). On the average, over a larger speech material, the cHELP coder has a segmental SNR 0.5 dB higher than that of VSELP.

Coder	male	fem	child	male	fem	child	male	fem	child
cHELP	10.4	13.4	11.4	12.2	13.0	13.7	13.4	13.6	14.1
VSELP	10.2	12.9	10.9	11.3	12.6	13.0	12.6	13.5	13.6

Table 4. Segmental SNR values for 9 different voices. The HELP coder operates at 24 bits/subblock utilizing the Hamming-1 neighbors of a sorted LTP.

6. SUMMARY

We have presented a family a methods for replacing the innovation codebook of CELP with interpolation in a generalized adaptive LTP-codebook. Using 26 bits per frame for the spectral parameters and another 6 bits for frame gain our proposed scheme can be coded at 6.4 kbit/s yielding a subjective quality well in excess of the familiar VSELP coder.

7. REFERENCES

[1] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, Boston 1992.

[2] M. Schroeder and B. S. Atal, "Code-Excited Linear Prediction (CELP): High Quality Speech at Low Bit-rates", Proc. ICASSP-85, pp. 937-940, 1985.

[3] K. K. Paliwal and B. S. Atal, "Efficient Vector Quantization of LPC Parameters at 24 Bits/Frame", Proc. ICASSP-91, 1991.

[4] R. Hagen and P. Hedelin, "A Robust Single-Stage VQ for Spectral Coding", 1993 IEEE Speech Coding Workshop, Québec, Canada, Oct. 1993.

[5] P. Hedelin, "Spectral Coding at 20 bits", IEEE ICASSP-94, Adelaide, Australia, April 1994.

[6] P. Hedelin, "A Multi-Stage Perspective on CELP- Speech Coding," *Proc.* IEEE Int. Conf. ICASSP-92 part I, pp 57-60, San Fransisco, March 1992.

[7] P. Kroon and B. S. Atal, "Pitch predictors with high temporal resolution," in *Proc.* IEEE Int. Conf. ICASSP-90, 1990.

[8] J. Makhoul and M. Berouti, "Adaptive noise spectral shaping and entropy coding in predictive coding of speech," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 27, no. 1, pp. 63-73, 1979.

[9] A. Okabe, et. al., *Spatial Tessalations: Concepts and Applications of Voronoi Diagrams*, Chichester, England, UK: John Wiley & Sons, 1992.