A SPEECH CODER POST-PROCESSOR CONTROLLED BY SIDE-INFORMATION

Guillaume Fuchs and Roch Lefebvre

University of Sherbrooke, Dept. of Electrical Eng., Sherbrooke,Québec, J1K 2R1 Canada
{guillaume.fuchs,roch.lefebvre}@usherbrooke.ca

ABSTRACT

Speech coders provide high speech quality at low rates. However they perform poorly when encoding non-speech signals. This paper proposes a new enhancement algorithm requiring minimum side information to reduce the effect of this shortcoming. The enhancement algorithm consists of post-processing the speech decoder output in the spectral domain. Specifically, some frequency components are reduced or forced to zero when the corresponding frequency content is poorly described by the speech coder. The choice of modifying spectral components is determined at the encoder, thus requiring to transmit the decision information. Experiments combining the AMR-WB speech codec and the proposed audio enhancement show that the quality for music signals is improved significantly while not affecting the quality for speech inputs.

1. INTRODUCTION

Although source models allow speech coders to maintain good quality at low rates for speech signals, the specific structure of speech coders penalizes the quality reproduction of more general audio inputs. Speech coders working at low or medium bit rates (from 2 to 20 kbit/s) are indeed based on a source model of speech production. The model assumptions make them well-suited for speech materials, but they are too restrictive to achieve good quality for most non-speech materials.

Yet, several applications such as audio-conferencing and voice communications, which handle mainly speech signals, may have to deal with music or background noise. Using a general audio coder instead of a speech coder is a costly solution in terms of bit rate and may not be practical in terms of encoding delay. It would also degrade the speech quality at low rates. For those reasons, recent approaches have considered hybrid solutions. A classical approach is multi-mode coding which applies signal type discrimination to the input audio. Depending on the classification result, the system switches between coding schemes exploiting different paradigms [1] or adapts the parameters of a coding algorithm (window length, pitch predictor, etc.) [2].

Another approach which can have the advantage of preserving interpretability with existing coders, is embedded coding. A core layer provides basic quality, and additional layers provide quality increments. The core layer is typically a speech coder, whereas the upper layers use more general audio coding techniques, i.e. quantization of transform coefficients ([3], [4]). To be efficient, embedded coding requires a significant increase in bit rate compared to the core layer. One advantage of embedded coding is the ability to adapt the bit-stream rate to the varying network conditions by dropping upper layers.

In this paper, we present a post-processing algorithm which falls in the category of embedded approaches, but without requiring a signifiant increase in bit rate. Specifically, the output of a speech decoder is processed in the frequency domain, using side information sent by the encoder. This produces an enhanced synthesis signal, with significant quality increase in the case of music signals. The frequency domain is used for the post-processing layer since it allows efficient analysis and modification of the perceptually relevant features of the audio signal, as in general audio coding. The post-processing consists essentially in removing frequency content that was badly encoded by the speech encoder, i.e. by reducing or forcing to zero selected frequency components. Compromises can be made between quality increase, and additional bit rate and delay. The receiver has thus both the synthesis of the core speech decoder and the enhanced audio samples at the output of the post-processor. The process is independent from the choice of the core layer. In this work, we consider the AMR-WB speech coding standard [5].

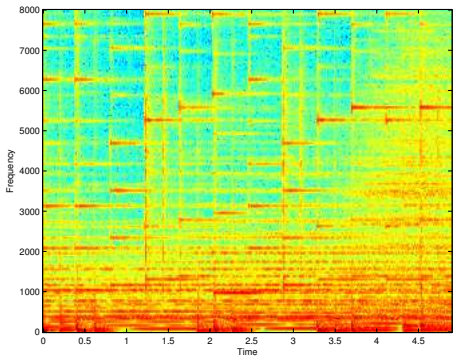
The paper is organized as follows. Section 2 outlines the main artifacts of a speech coder, such as AMR-WB, when the input audio is music. Section 3 presents the proposed post-processing algorithm, along with methods to minimize the bit rate of the related side information. Results of subjective tests are shown in Section 4. Finally, Section 5 gives some conclusions and presents directions for future work.

2. AMR-WB CHARACTERISTICS

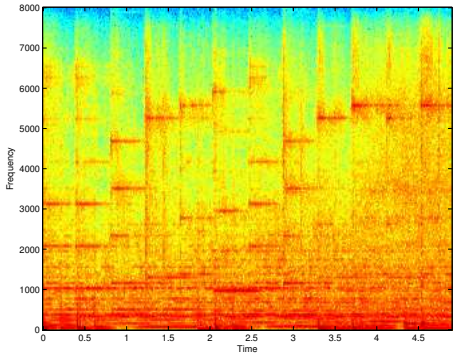
The AMR-WB speech coding standard [5] is based on the Algebraic Code-Excited Linear Prediction (ACELP) model. ACELP coders belong to the class of analysis-by-synthesis linear predictive coders. They combine the features of model-based vocoders, by representing the formant and the pitch structure of speech, with the properties of waveform coders by matching the output and input waveforms (in a weighted domain). AMR-WB handles wideband signals (50-7000 Hz) by coding separately the lower band (50-6400 Hz) and the higher band (above 6400 Hz). The lower band is encoded using ACELP, while the higher band is reconstructed at the decoder using the parameters of the lower band and a partially random excitation. AMR-WB operates at multiple bit rates from 6.6 to 23.85 kbit/s. The post-processing described in Section 3 is applied only to the lower band.

The AMR-WB performs very well on speech signals, but the quality for music inputs suffers from different artifacts due to the speech-specific source model in ACELP. One of the problems is that the residual of the short-term and long-term predictors in ACELP still exhibits large correlations when handling music. The fixed codebook in ACELP fails to encode adequately all this information, especially at lower rates.

As shown in Fig. 1, taking violin music with long pseudo-stationary periods as an example, the spectrogram of the synthesis signal for such non-speech material has a high noise floor between the sinusoidal components (Fig. 1(b)). We note also that the tones at high frequencies are not as well represented as the tones in the lower frequencies. However, the main differences are located in the spectrum valleys rather than in the spectrum peaks. This is one of the motivations for the post-processing described in the next section.



(a) Spectrogram of the original signal



(b) Spectrogram of the AMR-WB synthesis

Fig. 1. AMR-WB fails to reproduce properly music sound (taking violin as an example)

3. ZERO FORCING IN THE FREQUENCY DOMAIN

3.1. Principle

Considering the previous observations, we propose to apply a post-processing to the speech decoder output to obtain an enhanced synthesis signal. The high level block diagram of the proposed method is shown in Fig. 2, where bitstream 1 is the encoded output of the speech encoder, and bitstream 2 is the side information that controls the post-processing. The goal of the post-processing, which operates in the frequency domain, is to lower the energy of the spectral valleys where the noise introduced by the speech encoder is too high. For this purpose, the simplest approach consists in setting to zero some of the frequency bins in the FFT of the decoded speech, according to a mask determined at the encoder and sent as side information. This is illustrated in Fig. 3. A Short-Term Fourier Transform (STFT) is applied to the speech decoder output. The received and decoded mask M is formed by a series of 0's and 1's, where 0 indicates that the amplitude of the corresponding frequency bin is set to zero, and 1 indicates that the frequency bin is left unchanged. Other modifications, such as gain scaling, can also be applied. The phases are left unchanged. Note that in Fig. 3, the STFT and STFT^{-1} boxes include proper time windowing for overlap-and-add.

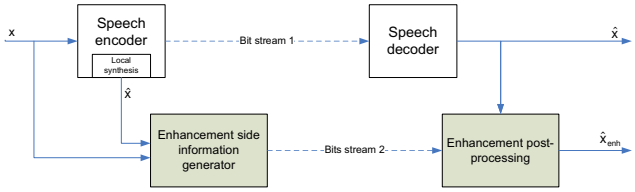


Fig. 2. High level block diagram of the proposed system

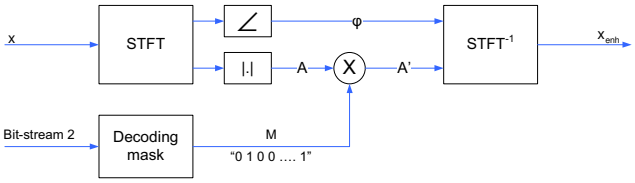


Fig. 3. Post-processing is applied to the speech decoder output $\hat{x}$

The mask M is generated at the encoder depending on the original and local synthesis signals (Fig. 4). The component of the mask at a given frequency bin will be set to 0 if the distortion due to the speech coder is above the original signal at the given frequency. The power spectrum of $x - \hat{x}_{enh}$ is thus forced to be below that of the original signal, x.

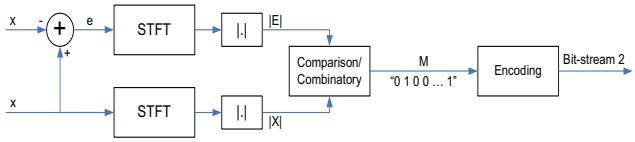


Fig. 4. The frequency zero forcing decision is taken at the encoder

The entire process can be summarized by the following formula:

$$\hat{X}(k)_{enh} = \begin{cases} 0 & \text{if} \quad |E(k)| > |X(k)| \\ \hat{X}(k) & \text{otherwise} \end{cases} \qquad (1)$$

where $X(k)$ and $E(k)$ are the k-th STFT coefficients of the original signal and the speech coder error, respectively, and $\hat{X}_{enh}$ the spectrum of the post-processing output.

The time-frequency analysis is an important issue for the efficiency of the process. The accuracy of the frequency representation will dictate the sharpness of the post-processing. Thus, it is essential to design a prototype filter with a small transition width and a strong stopband attenuation. Furthermore, block effects must be avoided. This leads us to adopt a STFT with a sine window overlapping of 50%. This is an appropriate compromise between accurate frequency representation and overcomplete decomposition due to overlap. Because the processing is applied only on the spectrum amplitudes, the mask M is output at critical rate, i.e. 1

bit/sample. We will see in the next sections how to reduce the side-information rate by exploiting the redundancies of the mask M.

The transform length dictates the tradeoff between time and frequency resolution and should be adjusted based upon the stationary of a given processed frame. Therefore, an adaptive window size is adopted in order to avoid annoying effects like pre-echoes. Large windows are used in stationary parts and short windows in transients. The window size is determined based on the speech decoder output, known by both the encoder and the decoder, by computing the time-segment energy-entropy [6]. The size N of the larger window is constrained to an integer multiple of the AMR-WB frame size (256 samples). The shorter window size is set to 64 samples (5 ms), which is in the order of backward temporal masking duration. Thus, the post-processing introduces a total delay of N samples. In this work we have used either a 512 (40 ms) or a 1024 (80 ms) point STFT for the larger window.

The spectrogram of the synthesis obtained after post-processing is shown in Fig. 5 for the violin example of Fig. 1(a). It can be seen that a large part of the high energy noise surrounding tonal components in the AMR-WB synthesis (see Fig. 1(b)) is removed in the lower band (0-6400 Hz). Note that the upper band remains unchanged since post-processing is only applied below 6400 Hz.

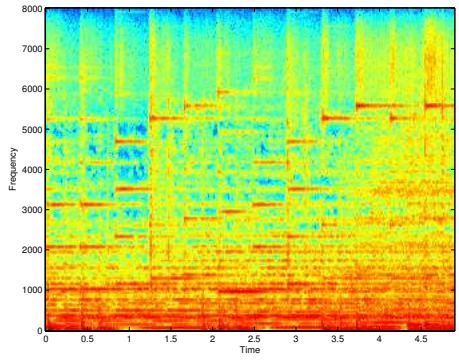


Fig. 5. Post-processing cleans the synthesis from the noise floor introduced by speech coder

3.2. Lossless coding of the mask

The mask M made of binary values needs considerable amount of data to be sent without any redundancy removal procedures. As seen in section 3.1, the uncompressed side information rate is 1 bit/sample, i.e 12.8 kbit/s for the 0-6400 Hz frequency band.

However, the mask M exhibits a great potential for redundancy reduction by means of run-length coding. Because the mask is only set to 0 in specific parts of the spectrum, specifically the valleys, long runs of 0s or 1s occur. This property could be exploited in run-length coding associated with a variable length coding. The two-valued process (mask M) is then transformed to a many-valued process before applying Huffman codes. Two separate run-length Huffman codes are used for 0-runs and 1-runs, which exhibit different distributions.

In Table 1 the performance of run length coding is shown. The two first columns show the entropy rates of the 0-run symbols and 1-run symbols calculated for each file. The total entropy rate gives the minimum average number of bits required for each input's sample. It is used as the benchmark for comparison with the performance of the implemented adaptive Huffman coding (coding rate column). It can be observed that the coding rate gets close to the entropy limit and that the achieved compression gains depend on the audio nature. However, the rate rarely comes down below 0.7 bit/sample.

Signal	0-run entropy rate	1-run entropy rate	Total entropy rate	Coding rate
Female voice	0.35	0.48	0.82	0.86
Male voice	0.24	0.4	0.64	0.68
Organ	0.48	0.34	0.82	0.85
Pop-music	0.47	0.47	0.94	0.98

Table 1. Rates obtained with the lossless coding approach (bit/sample)

3.3. Lossy coding of the mask

Lossy coding allows higher compression gains. Indeed, by simplifying the mask M, statistical proprieties of the 0-runs and 1-runs can be improved while minimizing the impact on the post-processing efficiency. Specifically, selected values of the mask are inverted to obtain longer runs and then avoid isolated occurrences. To minimize the loss of the post-processing efficiency, we define a candidature criterion for the mask component inversion:

$$C(k) = \begin{cases} \text{true} & \text{if } \frac{|X(k)|}{g_1} < |E(k)| < |X(k)|.g_0 \\ \text{false} & \text{otherwise} \end{cases} \quad (2)$$

where $g_1 \geq 1$ and $g_0 \geq 1$ are two user defined constants. The higher g_1 (resp. g_0) is, the more 1's (resp. 0's) are prone to be inverted. Values of $g_1 = 5$ and $g_0 = 1.3$ show good results.

Mask components which are candidates of inversion are processed by run. A run of candidates will be set to the same value. Knowing that a run of candidates is always bounded by two non-candidates values, we process as follows: if the run is bounded by two non-candidates of same value, the candidates will take this value; otherwise, the sum $W = \sum_i (|X(i)| - |E(i)|)$ is computed over the run. Then if $W \geq 0$, the candidates are set to one. Otherwise, they are set to zero.

Table 2 shows the significant compression gains achieved by the lossy coding approach. The gains are higher for signals already well handled by the speech coder, due to the long runs of 1's in M. On the other hand, the coding gain is not as high for music because of the higher entropy of runs occurring in M even after the mask simplification proposed above.

4. SUBJECTIVE EVALUATION

We conducted subjective tests using the MUSHRA [7] method to evaluate the performance of the post-processing. The test material consisted of four sequences sampled at 16 kHz : two speech sequences and two music sequences. Eight trained listeners participated in the test. For each sequence, the signal was encoded by AMR-WB at 12.65 kbit/s, G722.1 at 24 kbit/s, and AMR-WB at

Signal	0-run entropy rate	1-run entropy rate	Total entropy rate	Coding rate
Female voice	0.06	0.12	0.18	0.20
Male voice	0.04	0.09	0.13	0.15
Organ	0.24	0.23	0.48	0.49
Pop-music	0.14	0.21	0.34	0.36

Table 2. Rates obtained with the lossy coding approach (bit/sample)

12.65 kbit/s followed by the proposed post-processing using both lossless and lossy coding of mask M and two different window sizes of 40 and 80 ms. In total, with the hidden reference incorporated, 7 different versions of the sequences had to be graded for each audio sample. The results are reported in the Fig. 6.

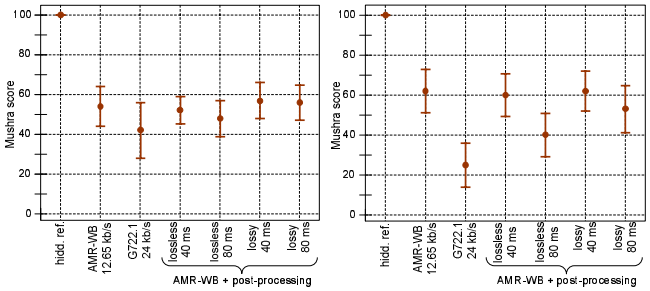
In the case of speech, the AMR-WB at 12.65 kbit/s outperforms G722.1 at 24 kbit/s, especially for the german male sequence. The proposed post-processing doesn't affect and even slightly improves the AMR-WB quality when using 40 ms windows. The larger window size (80 ms) leads to a poorer quality because of the short-time stationary of speech signals. The choice of parameters g_0 and g_1 makes lossy approach more conservative than lossless coding and limits the amount of frequency bins being set to zero. It explains the good behavior of lossy coding for speech samples.

In the case of music, the scores obtained by AMR-WB at 12.65 kbit/s are very low compared to the G722.1 at 24 kbit/s performance. The subjective tests show that the post-processing greatly narrows the performance gap between the two codecs. For the organ sound, G722.1 scored around 80 while AMR-WB scored as low as 14. The post-processing synthesis using lossless coding and 80 ms windows increases the score to 60. It is still lower than G722.1 but significatively improves the AMR-WB synthesis quality. Unlike speech, the 80 ms windows are more adapted to the music case.

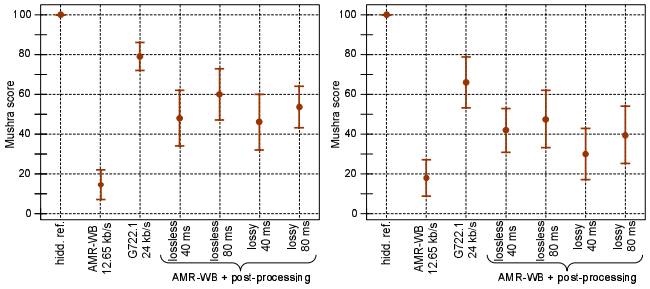
5. DISCUSSION

This paper presents a new speech coder post-processing developed mainly to enhance the quality of general audio. Subjective tests showed that the post-processing provides significant improvements for music signals while maintaining advantages of speech coders over audio coders when coding speech at low bit-rates. Even when limiting the delay to 40 ms, the post-processing is still efficient. Furthermore, the side-information rate can be reduced below 0.5 bit/sample with the help of a statistic enhancement algorithm and entropy coding. The side-information rate drops when handling speech materials.

In future investigations, studying inter-frame correlations could further minimize the bit rate. In addition, it would be interesting to try others time-frequency representations, such as wavelets or wavelet packets. Finally, the time and frequency resolutions are adaptive because of the window switching. However, the choice between three or more window sizes would be more flexible. It is hoped that futur advancements will make the solution appealing for scalable and universal speech and audio coding at low bit-rates.



(a) French female voice (b) German male voice



(c) Organ music piece (d) Lenox's pop-music song

Fig. 6. MUSHRA results: mean scores and the 95% confident intervals

6. ACKNOWLEDGEMENT

The authors wish to acknowledge the contribution of C. Laflamme, retired from the University of Sherbrooke, from whom the original idea comes.

7. REFERENCES

[1] L. Tancerel, S. Ragot, V.T. Ruoppila, and R. Lefebvre, "Combined speech and audio coding by discrimination," *IEEE Speech Coding Workshop*, pp. 158–160, Sept. 2000.

[2] S.A. Ramprashad, "The Multimode Transform Predictive Coding Paradigm," *IEEE Trans. on Speech and Audio Proc.*, vol. 11, no. 2, pp. 117–129, March 2003.

[3] S.A. Ramprashad, "Embedded coding using a mixed speech and audio coding paradigm," *International Journal of Speech Technology*, vol. 2, pp. 359–372, 1999.

[4] B. Kovesi, D. Massaloux, and A. Sollaud, "A scalable speech and audio coding scheme with continuous bitrate flexibility," *Proc. of International Conference on Acoustics, Speech, and Signal Processing*, pp. 273–276, May 2004.

[5] B. Bessette et al., "The Adaptive Multirate Wideband Speech Codec (AMR-WB)," *IEEE Trans. on Speech and Audio Processing*, vol. 10, no. 8, pp. 620–637, Nov. 2002.

[6] D. Sinha and A.H. Tewfik, "Low bit rate transparent audio compression using adapted wavelets," *IEEE Trans. on Speech and Audio Processing*, vol. 41, no. 12, pp. 3463–3479, Dec. 1993.

[7] Recommendation ITU-R BS.1534-1, "Method for the subjective assessment of intermediate quality level of coding systems," .