DATA HIDING FOR SPEECH BANDWIDTH EXTENSION AND ITS HARDWARE IMPLEMENTATION

Fan Wu, Siyue Chen and Henry Leung

University of Calgary
Department of Electrical and Computer Engineering
2500 University Drive NW, Calgary, AB, Canada, T2N 1N4

ABSTRACT

Most of the current speech transmission systems are only able to deliver speech signals in a narrow frequency band. This narrowband speech is characterized by a thin and muffled sound. In this paper, we propose a data hiding scheme to artificially extend the speech bandwidth so that the speech quality can be improved. The hardware perspectives of implementing the proposed scheme are also discussed. A Xilinx MicroBlaze soft processor is used in this study. To balance the trade off between logic resource consumption and processing speed, the data hiding scheme are mainly implemented in application software while fast Fourier transform (FFT) is designed into a hardware acceleration model. Experimental results demonstrate the effectiveness of the proposed data hiding scheme.

1. INTRODUCTION

Speech transmitted in communication networks is mostly in a relatively narrow band (NB), around 300-3400 Hz. Both pleasantness and intelligibility suffer from such a limited bandwidth. Listening experiments [1] have shown that the acoustic bandwidth contributes significantly to the perceived quality and intelligibility of speech. However, it is difficult to change the current speech transmission infrastructure to provide a wider bandwidth due to economic considerations. An alternate method is to artificially, instead of virtually, extend the speech bandwidth [2, 3]. More specifically, the missing high band (HB) components are estimated from the NB speech by exploiting their mutual dependencies. The estimated HB components are then used along with the NB speech to reconstruct a wideband (WB) speech. An accurate estimation of HB components usually requires a complicated speaker-dependent training of statistical models, which is computation-costive and thus not feasible for real-time processing. Although the training process can also be carried out off-line, i.e., speaker-independently, the performance of WB speech reconstruction degrades significantly.

Recently, data hiding [4, 5] has been proposed to transmit extra payload through multimedia without degrading multimedia's perceptual quality. In this paper, we employ the idea of data hiding to perform speech bandwidth extension. That is, the encoded HB information is imperceptibly embedded into the NB speech by modulating the frequency components below the perceptual masking thresholds. While the hidden information can be retrieved from the received NB speech, it is used to reconstruct a WB speech with better quality and intelligibility. Compared to the conventional estimation-based ABE methods, the proposed data hiding scheme has an advantage of using the real HB information instead of the estimated one, thus is more accurate in reconstructing WB speech.

The proposed scheme is also implemented in the hardware level. Generally, there are two ways in hardware implementations. The first is to implement in application specific logic or programmable logic. By this way, a fast processing speed can be achieved, however, with a significant increase of logic resource consumptions. The second way is to implement in application software. Compared to the first way, application software is relatively easier to develop. Thus, its design period is relatively shorter. In addition, application software has a high flexibility and compatibility since application software can be conveniently tested and verified in other environments, although initially, it may be developed for a specific kind of processors. Based on these considerations, we implement the data hiding scheme mainly in application software, however, with fast Fourier transform (FFT) implemented in hardware acceleration model.

The paper is organized as follows. Section II describes the proposed data hiding scheme. Section III presents hardware implementation of the proposed scheme. Experimental results are reported in section IV. Finally, conclusion remarks are given in section V.

2. DATA HIDING SCHEME

The flowchart of the proposed data hiding scheme is plotted in Figure 1. As shown, WB speech with the sampling rate of 16 kHz first undergoes band split by a low-pass and a high-pass filter respectively. The output of the low-pass filter is then down-sampled to provide the NB speech $x_{NB}(k)$, $0 \leq$

1-4244-0367-7/06/$20.00 ©2006 IEEE

$k \leq N-1$, where N is the number of speech samples. The output of the high-pass filter is shifted to the NB frequency range, and also decimated to provide an NB version of the HB speech, i.e., $x_{HB}(k)$, $0 \leq k \leq N-1$.

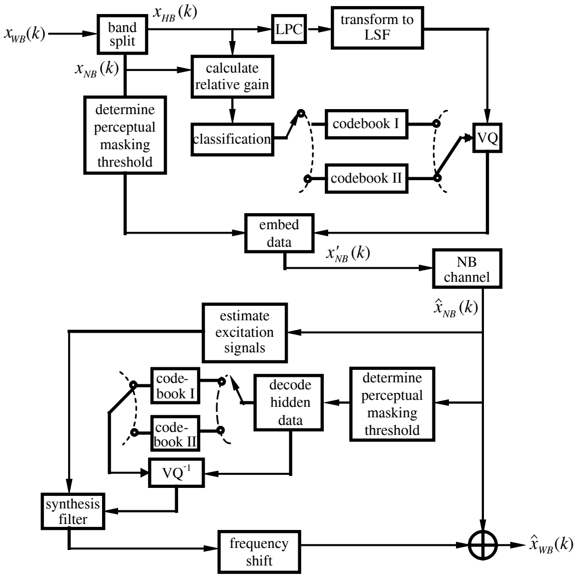

Fig. 1. The flowchart of the data hiding scheme.

In order to imperceptibly embed HB information into $x_{NB}(k)$, it would be desirable to minimize the number of the digital bits that represent $x_{HB}(k)$. In this study, linear prediction coding (LPC) [6] and vector quantization (VQ) are employed to achieve this purpose. LPC is based on the assumption that human voice production can be modelled by a process of passing excitation signals through an all-pole synthesis filter. The filter coefficients are the reciprocal of the coefficients of an auto-regressive (AR) filter. Assume that the AR coefficients of $x_{HB}(k)$ are denoted as $\{a_{HB}(i), i = 1, \ldots, N_a$, where N_a is the filter order. They are obtained by using the Levinson-Durbin algorithm [6] to solve the set of equations $\sum_{i=1}^{N_a} a_{HB}(i)u'(|i-j|) = -u'(j)$, $j = 1, \ldots, N_a$, where $u'(i)$ is the modified autocorrelation coefficients [6]. Before applying VQ on the AR coefficients, they should be transformed to line spectral frequencies (LSF). This is because AR coefficients are sensitive to quantization errors. A slight change in AR coefficients will result in significant distortions when reconstructing HB speech [6]. Instead, LSF are relatively less sensitive to quantization errors. Moreover, the AR coefficients can be precisely transformed back from the corresponding LSF.

Besides LSF, the gain of $x_{HB}(k)$ should also be embedded since the synthesized HB speech has to be scaled to an appropriate energy to avoid over-estimation [7]. Therefore, the relative gain of $x_{HB}(k)$ against $x_{NB}(k)$, i.e., $G_{rel} = \frac{G_{HB}}{G_{NB}}$, is calculated, and combined with N_a LSF to provide a representation vector of $x_{HB}(k)$, i.e., $\mathbf{a} = [LSF_1, \ldots, LSF_{N_a}, G_{rel}]$. $\mathbf{a}$ is then quantized to the closes entry of a VQ codebook that is generated by the fuzzy c-means (FCM) algorithm [8]. By doing so, only the entry index, instead of $\mathbf{a}$, is to be embedded into NB speech. It is noted that two codebooks are used in VQ. One is for the situation of $G_{rel} < -15$ dB, and the other is for $G_{rel} \geq -15$ dB. This is because that the situation of $G_{rel} \geq -15$ dB has a low probability to occur. If only one codebook is used, the representation vectors belonging to this category may all quantized to one entry, which will result in significant information loss.

Assuming that the codebook size is N_c, $\log_2 N_c$ binary digits are needed to represent a specific entry index. Considering one more bit is required to indicate which codebook is used, we then have to embed $\log_2 N_C + 1$ data bits in total, i.e., $\{b_m\}, m = 0, 1, \ldots, \log_2 N_c + 1$. To determine the location of data embedding, the perceptual masking thresholds of NB speech, i.e., $T(k)$, are estimated using the method in [9]. Since the frequency components with their magnitudes below $T(k)$ are imperceptible to human ears, they can be employed to carry b_m without degrading the perceptual quality of NB speech. The process of embedding b_m into the imperceptible components are formulated as

$$\begin{array}{ll} X'(k) = |\mathrm{Re}[X(k)]| + |\mathrm{Im}[X(k)]| & \text{if } b_m = 0, \\ X'(k) = -|\mathrm{Re}[X(k)]| - |\mathrm{Im}[X(k)]| & \text{if } b_m = 1, \end{array} \tag{1}$$

where $X(k)$ is the original FFT coefficients of $x_{NB}(k)$, $X'(k)$ is the modified coefficients, Re[·] and Im[·] are the operators to return the real and the imaginary part respectively. The frequency spectrum now becomes

$$X(k) = \left\{ \begin{array}{ll} X(k), & \text{if } |X(k)|^2 \geq T(k), \\ X'(k), & \text{if } |X(k)|^2 < T(k), \end{array} \right. \quad k = 0, \ldots, N-1. \tag{2}$$

$X(k)$ is then transformed back to the time domain, providing the composite NB speech $x'_{NB}(k)$ to be sent through the PSTN channel.

At the receiver side, conventional phone sets treat the received signal as an ordinary speech signal. Since the modified frequency coefficients are under the perceptual masking thresholds, they will not be perceptible to human listeners. Meanwhile, a pre-designed decoding mechanism is able to retrieve $\{b_m\}$. In more details, the perceptual masking thresholds are re-estimated. Since the only modification applied on the original NB speech is changing the phases of the imperceptible components, the re-estimated perceptual masking thresholds should be the same as those obtained at the transmitter. For the components below the thresholds, a decoding process is applied as

$$b_m = \left\{ \begin{array}{ll} 0, & \text{if } \mathrm{Re}[X(k)] > 0 \text{ and } \mathrm{Im}[X(k)] > 0, \\ 1, & \text{if } \mathrm{Re}[X(k)] < 0 \text{ and } \mathrm{Im}[X(k)] < 0. \end{array} \right. \tag{3}$$

Given b_m, the quantized LSF and the relative gain can be properly retrieved from the VQ codebook. The LSF are transformed back to the AR coefficients. Meanwhile, the excitation signal is obtained as the residual of an LPC analysis on the received NB signal, i.e., $r(k) = \hat{x}_{NB}(k) - \sum_{i=1}^{N_a} a_{NB}(i)$ $\hat{x}_{NB}(k-i)$, where $r(k)$ is the residual and $a_{NB}(i)$ denotes the AR coefficients of the received NB speech. $r(k)$ is then used to excite the all-pole synthesis filter described by the coefficients obtained from the quantized LSF, generating $\hat{x}_{HB}(k)$, the reconstructed $x_{HB}(k)$. The gain of $\hat{x}_{HB}(k)$ is adjusted to $\hat{G}_{HB}$, which is obtained by $\hat{G}_{HB} = \hat{G}_{NB} \cdot \bar{G}_{rel}$, where $\hat{G}_{NB}$ is the gain of the received NB speech, $\bar{G}_{rel}$ is the one retrieved from the VQ codebook. At this point, $\hat{x}_{NB}(k)$ and $\hat{x}_{HB}(k)$ are still the signals sampled at 8 kHz. They should be up-sampled to 16 kHz, the sampling rate of WB speech. In addition, the up-sampled $\hat{x}_{HB}(k)$ should be shifted to its destination frequency band by a high-pass filter, providing the restored HB speech. Finally, a WB speech is artificially generated by adding the restored HB speech to the up-sampled NB speech.

3. HARDWARE IMPLEMENTATION

Hardware implementation of the proposed data hiding scheme is presented in this section. The processor used in our design is a Xilinx MicroBlaze soft processor. It is a reduced instruction set computer (RISC) CPU, which allows users to select any combination of peripherals and controllers. Furthermore, it can be easily connected with user-defined hardware acceleration intelligent property (IP) core through its on-chip peripheral bus (OPB) or its peripheral local bus (PLB). This feature helps to maximally meet the requirements of a non-standard design. The FPGA employed in this study is Virtex II XCV2000FF896-4, a ball grid arrays (BGA) packed 624-pin large memory resource chip. When synthesizing the processor into the FPGA, the clock rate is determined at 87 MHz.

Figure 2 plots the hardware design of the proposed data hiding-based ABE scheme . As shown, a single soft processor is used for both encoding and decoding due to the limitation of hardware resources. The encoder output $x'_{NB}(k)$ is thus directly connected to the decoder input. Since the sampling rates of $x_{NB}(k)$ and $x_{HB}(k)$ are 8 kHz and the clock rate is 87 MHz, three FIFOs (First in, First out) are used in both the input and the output port so that the fast processing speed are able to match the low sampling rate. Each FIFO has its word width as 8 bits and its depth as 512. In total, three FIFOs consume 1.5 kilobytes of the FPGA resource. The FIFOs are connected with the processor through PLB, which can read an 8-bit data within one clock period.

It is noticed that FFT should be carried out for every speech frame to estimate the perceptual masking thresholds. If FFT can be implemented with a fast speed, the processing time of our scheme would be shortened significantly. When FFT is implemented in software application program and run by the soft processor, it takes 0.425 millisecond (ms) to encode and decode one frame. Meanwhile, if FFT is implemented in hardware acceleration IP core with 32-bit precision, as shown in Figure 2, the encoding and decoding of each frame costs 0.212 ms. Therefore, implementation of FFT in hardware acceleration IP core are able to increase the processing speed by 50.12%.

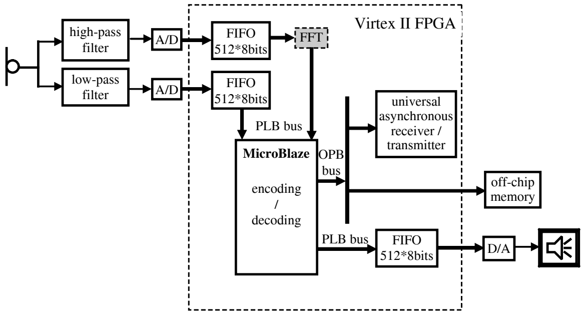

Fig. 2. Hardware implementation of the data hiding scheme.

4. EXPERIMENTAL RESULTS

Experiments are designed to evaluate the performances of the proposed scheme. The WB speech is sampled at 16 kHz. It is segmented into frames for processing. Each frame contains 512 samples, overlapping with the previous frame by 384 samples. After processing, the segmented frames are concatenated by the overlapping-add algorithm [10]. The order of the AR filter is set as 14, i.e., $N_a = 14$.

To evaluate the perceptual similarity of the reconstructed and the original HB speech, we employ log spectral distortion (LSD) [2], defined as

$$\begin{aligned} LSD \;=\; & \frac{1}{2\pi}\int_{-\pi}^{\pi}\Bigg(20\log_{10}\frac{G_{rel}}{|A_{HB}(e^{j\omega})|} \\ & -20\log_{10}\frac{\hat{G}_{rel}}{|\hat{A}_{HB}(e^{j\omega})|}\Bigg)^2 d\omega, \end{aligned} \tag{4}$$

where $A_{HB}(e^{j\omega}) = \sum_{i=1}^{N_a} a_{HB}(i)e^{-ji\omega}$, due to its reasonable correlation with the subjective speech quality. A smaller value of LSD indicates a better perceptual quality of the reconstructed HB speech. For the comparison purpose, the ABE method based on codebook mapping using hidden Markov Model (HMM) [2] is also implemented. This method estimates the AR coefficients of HB speech by averaging the weighted entries in a codebook. The weights are determined by the training results of HMM. In our experiment, the HMM training is performed offline to meet the need of real-time processing. Figure 3 plots the LSD results versus codebook sizes. It is seen that for the both schemes, the LSD value becomes smaller when we increase the codebook size. This is because the more entries in a codebook, the less the quantization errors. In addition, we also find that when the codebook size is larger than 8, the data hiding scheme always outperforms the HMM codebook mapping scheme, no matter whether one codebook or two codebooks are used. It should be noted that when two codebooks are used, the codebook size is the total number of the entries in both codebooks. From Figure 3, we observe that the data hiding scheme with two codebooks is always better than that with one codebook. Therefore, classifying WB speech into two categories is necessary to improve the performance of our proposed scheme.

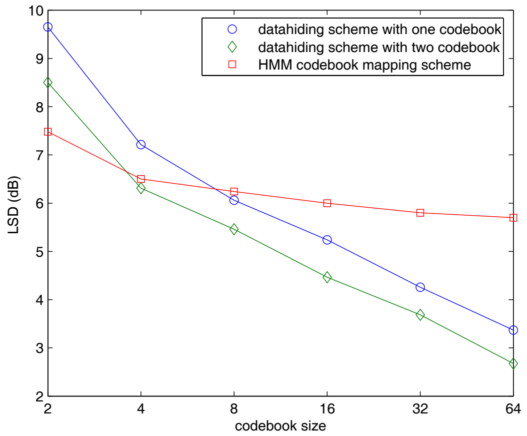

Fig. 3. The LSD performances of the data hiding scheme and the HMM codebook mapping scheme.

The hardware performance results are shown in Table 1. It is observed that due to a high computation load, it takes the HMM codebook mapping scheme around 1 ms to process one speech frame. Meanwhile the data hiding scheme needs only 0.21 ms. Therefore, the data hiding scheme is faster than the HMM codebook mapping scheme in hardware implementation. Because the soft processor consumes most of memory resources, the memory consumption is up to around 50 Megabytes.

5. CONCLUSIONS

This paper proposes a data hiding scheme to extend speech bandwidth. More specifically, HB speech is encoded into digital bits and embedded into NB speech imperceptibly. When the hidden data is decoded at the receiver, HB speech can be reconstructed and combined with NB speech to provide a WB speech. It is shown that the proposed scheme has a good WB reconstruction performance in terms of LSD. Furthermore, it has a processing speed of 2381 frames/second. Compared to the HMM codebook mapping method, the proposed scheme is able to meet the requirement of real-time processing and consumes less hardware resources. It provides a practical and effective solution to speech bandwidth extension.

6. REFERENCES

[1] S. Voran, “Listener ratings of speech passbands,” *Proc. IEEE Workshop on Speech Coding*, pp. 81–82, Pocono Manor, PA, USA, Sept 1997.

[2] P. Jax and P. Vary, “On artificial bandwidth extension of telephone speech,” *Signal Processing*, vol. 83, no. 8, pp. 1707–1719, 2003.

[3] T. Unno and A. McCree, “A robust narrowband to wideband extension system featuring enhanced codebook mapping,” *Proc. IEEE ICASSP*, vol. 1, pp. 805–808, Philadelphia, USA, March 2005.

[4] F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn, “Information hiding - A survey,” *Proc. of IEEE*, vol. 87, no. 7, pp. 1062–1078, 1999.

[5] M. Wu and B. Liu, “Data hiding in image and video-I: Fundamental issues and solutions,” *IEEE Transactions on Image Processing*, vol. 12, no. 6, pp. 685–695, 2003.

[6] L. Hanzo, F. C. A. Somerville and J. P. Woodard, *Voice Compression and Communications: Principles and Applications for Fixed and Wireless Channels*, IEEE Press, 2001.

[7] M. Nilsson and W. B. Kleijn, “Avoiding overestimation in bandwidth extension of telephony speech,” *Proc. IEEE ICASSP*, pp. 869–872, May 2001.

[8] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, New York: Plenum, 1981.

[9] ISO/IEC JTC 1/SC 29/WG 11, ISOIEC 13818-3 “Information technology-Generic coding of moving pictures and associated audio information–Part3: Audio,” April 15, 1998.

[10] A. V. Oppenheim, R. W. Schafer and J. R. Buck, *Discrete-Time Signal Processing*, 2nd edition, NJ: Prentice Hall, 1999.

Table 1. Experimental results on hardware performances

	processing time (ms)	logical consumption (logical units)	memory consumption (kilobytes)
Data hiding	0.212	2192	53760
HMM codebook mapping	1	2168	54784