WATERMARK SYNCHRONIZATION FOR FEATURE-BASED EMBEDDING: APPLICATION TO SPEECH

David J. Coumou[*], *Gaurav Sharma*[*]
[*] Electrical and Computer Engineering, Department, University of Rochester, Rochester, NY 14627
Email: DavidCoumou@ieee.org, gaurav.sharma@rochester.edu

ABSTRACT

We propose a novel framework for synchronization in feature-based data embedding systems. The framework is tolerant to de-synchronizing errors in feature estimates, which have hitherto crippled feature-based embedding methods. The method uses a concatenated coding system comprising of an outer q-ary LDPC code and an inner insertion-deletion code to recover from both de-synchronization caused by feature estimation discrepancies between the transmitter and receiver; and errors in estimated symbols arising from other channel perturbations. We illustrate the framework in a speech watermarking application employing pitch modification for data-embedding. We show that the method indeed allows recovery of watermark data even in the presence of de-synchronization errors in the underlying pitch-based embedding. The resilience of the method is also demonstrated over channels employing low bit rate speech encoders.

1. INTRODUCTION

Synchronization of oblivious watermark data channels is an extremely challenging problem. While several approaches have been proposed for this problem, each has limitations on the types of channels which can be handled, and therefore synchronization remains the Achilles heel of a majority of current watermarking systems.

In this paper, we propose a new framework for watermark synchronization based on embedding in multi-media features and recovery of synchronization using practical insertion-deletion codes that have recently been developed by Davey and MacKay [1]. Watermarking methods that use semantically meaningful signal features, either for embedding [2] or for partitioning the signal space into regions for embedding [3] are inherently attractive since large perturbations to these features typically also cause undesirable perceptible distortions in the content. Unfortunately, thus far, feature-based data embedding methods have also been among the most challenging from a synchronization perspective [4]. This is because robust and repeatable extraction of semantically meaningful image features continues to be a challenging research problem in itself and even benign processing or the process of data embedding itself can alter estimated features leading to de-synchronization of the watermark channel.

To remedy this problem, we propose a new framework that combines the feature-based embedding with special error correction codes for channels with insertions, deletions and substitutions (IDS)[1][5] to allow recovery of synchronization when feature mismatches occur between the transmitting and receiving entities. We use a speech watermarking system based on pitch modification previously developed within our group [2] for a sample implementation of the framework.

2. FEATURE-BASED MULTIMEDIA DATA EMBEDDING WITH SYNCHRONIZATION

Our proposed framework for multi-media watermarking incorporating synchronization is shown in Figure 1. The basic data embedding and extraction technique is indicated as the block outlined with the solid border. At the transmitting end, the method embeds data t in the signal through modifications of semantic features of the multimedia signal. The receiver attempts to recover this data (by estimating the semantic features), yielding an estimate $\hat{t}$. Though several methods of this type are known, when de-synchronization occurs due to differences in feature estimates between transmitting and receiving ends, the methods fail catastrophically [4]. To recover from these failures, we propose the addition of an encoder/decoder for synchronization and error recovery, shown as the dotted block in Figure 1. The resulting combination yields a novel watermarking framework that is able to achieve overall watermark synchronization despite loss of synchronization in the underlying data embedding method. In the rest of this paper, we illustrate this framework using a speech-specific pitch-modification based technique for the data-embedding [2] and a concatenated coding system [1] for the synchronization. A more general perspective and taxonomy of synchronization methods may be found in [6].

3. SPEECH DATA EMBEDDING BY PITCH MODIFICATION WITH SYNCHRONIZATION

Figure 2 illustrates the system for watermark embedding in speech based on our proposed framework. Space constraints limit us to a high level overview of the presented system. Details shall be available in a companion paper, currently under preparation [7]. The two main elements are the pitch-based data embedding and the concatenated coding system for handling insertion, deletion, substitution errors.

1-4244-0367-7/06/$20.00 ©2006 IEEE

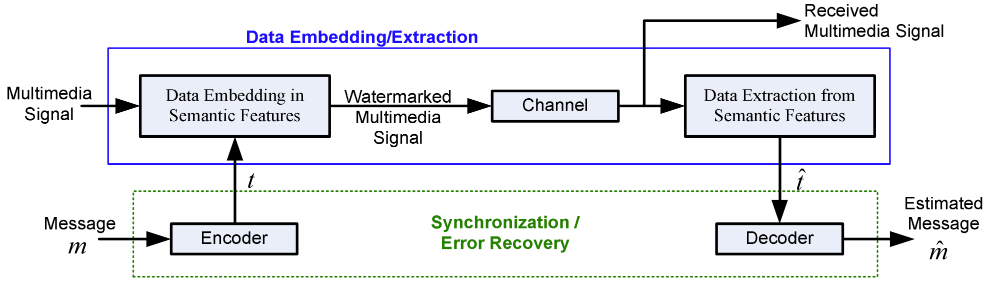

Figure 1: Feature-based data embedding with synchronization

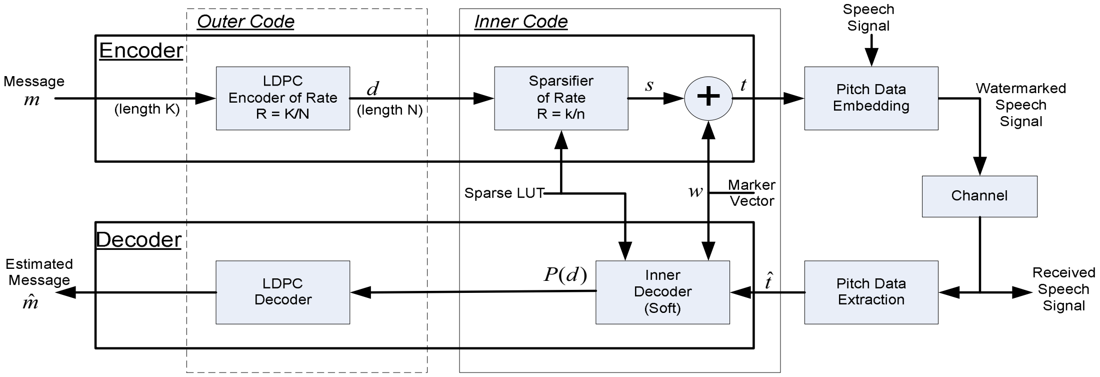

Figure 2: Pitch-based speech watermark with synchronization

Data-embedding by pitch-modification

As illustrated in the right most blocks in Figure 2, we use pitch of voiced regions of a speech signal as the "semantic" feature for data embedding [2]. The choice is motivated by the structure of most low bit-rate speech encoders [8][9] that ensures pitch information is preserved.

Data is embedded by altering the pitch period of voiced segments that have at least M contiguous windows. M is experimentally selected to avoid small isolated regions that may erroneously be classified as voiced. Within each selected voice segment one or more bits are embedded. A single bit is embedded by QIM of the average pitch value. This corresponds to the method presented in [2]. For multi-bit embedding, the voiced segment is partitioned into blocks of J contiguous analysis windows ($J \leq M$) and a bit is embedded by scalar QIM of the average pitch of the corresponding block. Specifically, the average pitch for a block is computed as $p_{avg} = \frac{1}{J}\sum_{i=1}^{J} p_i$, where $\{p_i\}_{i=1}^{J}$ are pitch values corresponding to the analysis windows in the block. Scalar QIM [10] is applied to the average pitch for the block: $p'_{avg} = Q_b(p_{avg})$ where b is the embedded bit and $Q_b()$ denotes the corresponding quantizer. The stream of embedded bits forms the embedded message t. Modified pitch intervals for the analysis windows in the block are computed as: $p'_i = p_i + (p'_{avg} - p_{avg})$. The corresponding pitch modifications are then incorporated in the speech waveform using the pitch synchronous overlap add (PSOLA) [11] algorithm. Embedding in average pitch over blocks of analysis windows enables embedding even when the pitch period exceeds the duration of a single window and also reduces perceptibility of the changes introduced. The use of multiple embedding blocks within a voiced segment (of J analysis windows each) ameliorates data capacity as compared to the single bit embedding in each voice segment.

At the receiver, the speech waveform is analyzed to detect voiced segments and pitch values are estimated for non-overlapping analysis windows of L samples each. In a process mirroring the embedding, average pitch values are computed over blocks of J contiguous analysis windows. For each block, an estimated value of the embedded bit is computed as the index 0/1 of the quantizer $\{Q_b(\ )\}_{b=0}^{1}$ that a reconstruction value closest to the average pitch. This provides an estimate $\hat{t}$ of the embedded data.

One challenge for the data embedding by pitch modification is that estimates of voiced segments at the receiver may differ from those at the embedder [2]. Multiple voiced segments at the embedder may coalesce into a single voiced segment at the receiver, or vice versa. In addition, relatively small voiced segments may be detected at one end and not the other. In general, these types of mis-matches result in IDS errors in the estimates of the embedded data[1] (See Figure 3). Insertion/deletion events are particularly insidious since they cause a loss of synchronization and

[1] As remarked earlier, these types of errors are encountered in almost all feature-based data embedding methods.

cannot be corrected using conventional error correction codes.

IDS Codes for Synchronization

To address synchronization over IDS channels, we next incorporate a concatenated code from [1]. This is shown as the inner and outer codes in Figure 2. The first step in the concatenated coding scheme encodes the q-ary message m of length K with a q-ary low-density parity check (LDPC) [12] code to produce a codeword d of length N. In the next step, each of the q-ary symbols in d is mapped via a look-up-table (LUT) to a sparse binary vector s of length n ($n > k = \log_2(q)$). Information is communicated to the receiver via this sparse vector by "piggy-backing" it as deliberate bit-inversions in a pseudo-random marker vector w that is known at the receiver (through knowledge of the generating key). The data t to be embedded is computed as the modulo-2 sum of s and w. In the absence of any data (e.g. $s=0$), the marker vector w forms the bits t that are embedded in the speech signal. In this scenario of no embedding, from the received vector $\hat{t}$, IDS events in the channel may be estimated (with some uncertainty) by "aligning" the vector against the known marker code w. When data is embedded, this "alignment" is still effective because only sparse changes are made in w. Using the alignment, bit inversions may be readily located and, using the redundancy introduced by the LDPC code, the embedded data may be recovered. Note that in the preceding description we have adopted a slightly imprecise description in the interest of conveying the primary intuition of the technique concisely.

The "alignment" alluded to in our preceding discussion is actually accomplished by the inner decoder using a hidden Markov model (HMM) [13] to represent the IDS channel[1], whose parameters $\bar{H}$ are the probabilities of insertions, deletions and substitutions of the channel, the mean density of the sparse binary vectors and the watermark code, w. The HMM estimates the (q-ary) symbol-by-symbol likelihood probabilities $P(\hat{t}\,|d_i,\bar{H})$ for each of the N symbols in the LDPC codeword. These are utilized by the outer q-ary LDPC decoder as soft-inputs which are used in the iterative sum-product belief-propagation algorithm [12]. The iterative probabilistic procedure produces an estimated message $\hat{d}$ at the end of each iteration. Iterations are terminated once the LDPC parity check condition is satisfied, i.e. $H\hat{d}$ = 0, where H denotes the LDPC parity check matrix. If a predetermined number of iterations is exceeded a decoder failure is declared.

4. EXPERIMENTAL RESULTS

We implemented the proposed system using the PRAAT toolbox [14] for the pitch manipulation operations for analysis and embedding and MATLAB^{TM} for the inner and outer decoding processes. The channel operations corresponding to various compressors were performed using separately available speech codecs. For the sparse LUT we generated $q = 2^k$ vectors of length n with the lowest possible density of 1's and ordered them sequentially to represent the $q = 2^k$ possible values for a codeword symbol. For computational efficiency in the message passing for the q-ary code we utilized the FFT method suggested by Richardson et al [15].

In order to evaluate the performance of our proposed synchronization method for speech data embedding based on pitch modification, we used sample speech files from a database provided by [16]. The files consist of continuous sentences read by male/female speakers. For the q-ary LDPC code we generated an irregular binary parity check matrix with column weight of 3 and coding rate of ¼. The columns of the matrix were then assigned q-ary symbol values from the heuristically optimized sets made available by Mackay [17]. A generator matrix for systematic encoding was obtained using Gaussian elimination. The marker vector w was generated using a pseudo-random number generator whose seed served as a shared key between the transmitter and receiver. Coarse estimates of the channel parameters were found by performing a sample pitch based embedding and extraction that was manually aligned (with help from the timing information) to determine the number of insertion, deletion, and substitution events. The mean density of sparse vectors was obtained from the sparse LUT and made available to the inner decoder for the forward-backward passes. We point out that in all cases the impact of the watermark on the signals was imperceptible (in our limited testing).

To test the system, random message vectors of q=16-ary message symbols were generated. These were arranged in blocks of K=25 and encoded as LDPC code vectors of length N=100. The length of the sparse vectors was chosen as n = 10; resulting in an overall coding rate of 0.10. The binary data obtained from the sparsifier was embedded into the speech signal by QIM of the average pitch using a quantization step of $\Delta = 10$ Hz. Blocks of J = 5 analysis windows were used in the embedding, which (for our speech samples) provided an (uncoded) embedding data rate of approximately 8 bits per second. The communication channel was variously chosen as:

a) None, i.e., no compression was applied.
b) GSM-06.10 (Global System for Mobile Communications coder, Ver 6.10) at 13 kbps [8].
c) AMR (Adaptive Multi-Rate coder) at 5.1 kbps [9].

Figure 3 illustrates the impact of synchronization loss. The plot shows the differences between inserted bits t in the speech waveform and extracted bits $\hat{t}$. The "+" symbols at 0 along the y axis indicate locations where the embedded and extracted bits match and those at 1 indicate locations where they differ. A small initial segment (for small sequence index numbers) shows reasonable agreement after which the agreement is random due to loss of synchronization. Using the proposed system we are able to recover from this loss of synchronization. Table 1 summarizes the results for the proposed system over the different channels listed above. In each case we see that the method allows recovery of embedded data, despite the high number of initial errors caused by synchronization loss. Figure 4 shows the symbol error count for the tentative decoding from the outer LDPC code as a function of iteration count. In the absence of compression and for the GSM codec, the number of symbol errors rapidly decreases with each iteration, achieving correct decoding in less than 10 iterations. On the other hand, for the lower rate AMR codec, a large number of iterations are necessary in order to correct all the errors.

5. CONCLUSION

We introduced a novel paradigm for synchronization in multi-media data embedding that combines feature-based embedding with error correction codes capable of correcting insertion, deletion, substitution (IDS) errors. Experimental results for a speech based watermark implemented in the framework show that it indeed allows recovery of embedded data under common scenarios, where some feature mismatches occur between the transmitting and receiving ends. We anticipate that this paradigm is likely to be very useful and powerful with applications to a variety of other watermarking applications.

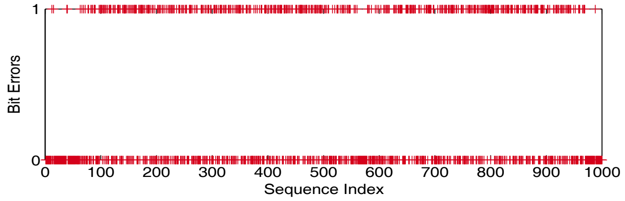

Figure 3: Bit errors in the absence of synchronization

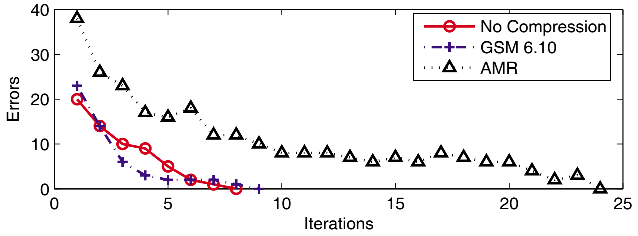

Figure 4: LDPC iterations vs. outer decoder errors

Channel Compression	Bit Errors w/o Synchronization	Errors w/ Proposed Method	# LDPC Iterations
None	464	0	8
AMR	313	0	24
GSM	441	0	9

Table 1: Error correction performance

6. ACKNOWLEDGEMENT

We would like to thank Mehmet U. Celik and Matthew C. Davey for help with implementing data-embedding in pitch and IDS codes, respectively.

7. REFERENCES

[1] M. C. Davey and D. J. C. Mackay, “Reliable communication over channels with insertions, deletions, and substitutions,” *IEEE Trans. Info. Theory*, pp. 687–698, Feb. 2001.

[2] M. Celik, G. Sharma, and A. M. Tekalp, “Pitch and duration modification for speech watermarking,” *Proc. IEEE Intl. Conf. Acoustics Speech and Sig. Proc.*, Mar. 2005, pp. II, 17–20.

[3] P. Bas, Chassery, J-M, and B. Macq, “Geometrically invariant watermarking using feature points,” *IEEE Trans. Image Proc.*, Vol. 11, No. 9, pp. 1014–1028, Sept. 2002.

[4] M. U. Celik, E. Saber, G. Sharma, and A. M. Tekalp, “Analysis of feature-based geometry invariant watermarking”, *Proc. SPIE: Security and Watermarking of Multimedia Contents III*, vol. 4314, Jan. 2001, pp. 261–268.

[5] L. R. Bahl and F. Jelinek, “Decoding for channels with insertions, deletions, and substitutions with applications to speech recognition,” *IEEE Trans. Info. Theory*, pp. 404–411, Jul. 1975.

[6]G. Sharma and D. J. Coumou, “Watermark synchronization: Perspectives and a new paradigm,” in Proc. 40th Annual Conf. on Info. Sciences and Systems (CISS), Princeton, NJ, 22-24 Mar. 2006, pp. 1182-1187 (invited paper).

[7] D. J. Coumou and G. Sharma, “Insertion, Deletion codes with feature-based embedding: A new paradigm for watermark synchronization with applications to speech watermarking,” In preparation for submission to *IEEE Trans. on Information Forensics and Security*, 2006.

[8] 3GPP TS6.10: "Full Rate Speech Transcoding", http://www.3gpp.org/ftp/Specs/archive/06_series/06.10/

[9]3GPP TS26.071: "AMR speech Codec; General description", www.3gpp.org/ftp/Specs/archive/26_series/26.071

[10] B. Chen and G. W. Wornell, “Quantization index modulation: a class of provably good methods for digital watermarking and information embedding,” *IEEE Trans. Info. Theory*, Vol. 47, No. 4, May 2001, pp. 1423–1443.

[11] E. Molines and F. Charpentier, “Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diaphones,” *Speech Communication*, pp. 453–467, 1990

[12] M. C. Davey and D. J. C. MacKay, “Low density parity check codes over GF(q)”, *IEEE Comm. Letters*, Vol. 2, No. 6, p. 165-167, June 1998.

[13] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proc. IEEE*, Vol. 77, No. 2, pp. 257–286, Feb. 1989.

[14] P. Boersma and D. Weenik, “Praat: doing phonetics by computer”; [Online]. Available: http://www.fon.hum.uva.nl/praat

[15] T. Richardson and R. Urbanke. “The capacity of low-density parity check codes under message-passing decoding”, *IEEE Trans. Info. Theory*, Vol. 47, No 2, p. 638-656, Feb., 2001

[16] Ohio State University Speech Corpus; http://buckeyecorpus.osu.edu

[17] D. J. C. MacKay, “Optimizing Sparse Graph Codes over GF(q)”, http://www.cs.toronto.edu/~mackay/gfqoptimize.pdf