NEXT GENERATION TECHNIQUES FOR ROBUST AND IMPERCEPTIBLE AUDIO DATA HIDING

Jim Chou, Kannan Ramchandran[*]

University of California - Berkeley
Department of EECS
Berkeley, CA 94708

Antonio Ortega

USC
Department of EE
Los Angeles, CA

ABSTRACT

In this work, we combine recent theoertical and algorithmic advances in the area of information-hiding with the current mature knowledge-base in the human audio perception system to propose a novel audio data-hiding technique that significantly pushes the state-of-the-art in the field. Our work is based on a combination of advances in two disjoint fields: information-hiding and human auditory masking. The field of information-hiding has recently seen a resurgence due to advances in the understanding of fundamental bounds from information theory. By integrating this with the human perceptual system knowledge that has been successfully exploited for several years in the audio compression community, we derive a new and improved audio data-hiding technique that finds application in a number of exciting scenarios like music enhancement and digital communications over analog data channels. Our preliminary results show that we can embed data at an order of magnitude higher rate than existing audio data hiding systems, while being robust to channel noise.

1. INTRODUCTION

It is a well known fact in the audio compression community that only a few bits per sample are needed to represent compact disk (CD) quality music. In fact, [1] pointed out that two to three bits per sample are usually sufficient for representing most genres of music. This implies that for uncompressed music, noise can be injected into the signal without it being perceptible to the end user. We utilize this fact, not for compression, but instead for hiding data in music. In particular, we will leverage recent promising work [2] in the field of data hiding to show how large amounts of data can be hidden in uncompressed audio signals. The method of data hiding is a constructive attempt at bridging the gap between what is currently available in data hiding technology and what is theoretically possible [3]. In the work of [3], bounds were given on the amount of data that can be hidden in a signal when the signal is $i.i.d.$ Gaussian and the noise that the data is subjected to is a concatenation of the known signal and unknown $i.i.d.$ Gaussian noise. In this work, we will formulate a method in which the audio signal can be modeled as a set of parallel Gaussian channels and show how data can be hidden in an imperceptible and robust fashion into the audio signal.

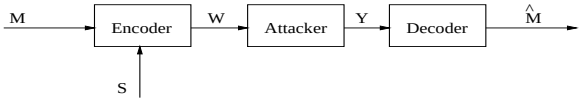

Fig. 1. The data hiding problem

We will then show how our method of audio data hiding can be extended to exciting applications ranging from embedding extra information onto CDs to increasing the throughput of existing analog communication channels. Current methods of audio data hiding [**?**] can embed a significant amount of information into audio signals but is typically not robust to channel noise, and is hence not applicable to the above applications. From our simulations, we will show that our method of audio data hiding is an order of magnitude above existing audio data hiding techniques and is also robust to channel noise.

2. GENERAL DATA HIDING

In general, the data hiding problem is formulated as follows (see [2]). The encoder has access to two signals; the information (an index set), M, to be embedded, and the signal that the information is to be embedded in. The output of the encoder, W, will then be subjected to random noise. The decoder will receive the corrupted encoded signal and will attempt to recover the embedded data. The goal, then, is to embed as much data as possible into the signal without altering the fidelity of the original signal. The fidelity constraint can be posed as a distortion constraint between the original signal and the encoded signal where the distortion

[*] Phillips and Microsoft Research.

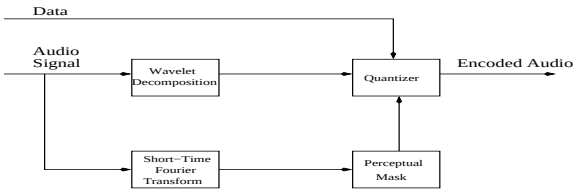

Fig. 2. Audio data hiding system diagram.

measure can range from a Euclidean-based measure to some perceptual measure. Mathematically, the goal is to solve the following constrained minimization problem:

$$min_{||W-S||^2 \leq D_1, ||Y-W||^2 \leq D_2} P_e(\hat{M}), \tag{1}$$

where $P_e(\hat{M})$ represents the probability of decoding error.

The above problem can be generalized into the problem of channel coding with side information, for which the capacity can be calculated (see Fig. 1). The capacity [4, 5, 3] of such systems is given by

$$C = max_{p(U,S|X)}[I(U;Y) - I(U;S)], \tag{2}$$

where the maximization is over all conditional probability density functions $p(U, X|S)$. The signal S is the side information about the channel and U represents the codeword space.

For the applications we will be considering, we can limit ourselves to the case where the channel is AWGN (see Fig. 1) and the signal (side information, S) is $i.i.d.$ gaussian. Our constructions can of course be generalized to accomodate other channels and input distributions. In the AWGN case, it was shown [3] that the capacity (2) is given as

$$C = \frac{1}{2} log(\frac{P}{N} + 1), \tag{3}$$

where P and N represent the transmitter power constraint and the variance of the channel noise respectively. It is interesting to note that as the variance of the channel noise approaches 0, the capacity approaches infinity. This implies that an infinite amount of data can be hidden within a signal given that the channel does not introduce any random noise! Of course, it must be recognized that no existing systems have come close to achieving the attainable bound of (3). Promising work on practical constructions for attaining (3) can be found in [6, 7].

3. AUDIO DATA HIDING

The method with which we hide data in audio signals is similar to that of state-of-the-art audio compression codecs [8]; the interpretation, however is different. A block diagram of our audio data hiding system is given in Fig. 2. From Fig. 2 we can see that the audio signal is first divided into short time frames and fed in parallel paths to a wavelet decomposition and a short-time Fourier transform. The wavelet decomposition serves the purpose of decomposing the audio signal into critical frequency bands that closely model the human auditory response. The short-time Fourier transform decomposes the audio signal into frequency coefficients that can be used to estimate the various tones that are present in the audio signal. These tones can then be used to estimate a perceptual mask that will dictate the amount of noise that can be added to the wavelet coefficients and be imperceptible to the ear. There are many ways to compute a perceptual mask; we chose to use the generic methods that are commonly employed in MPEG audio encoders [8]. The data that is to be embedded in the signal will index a quantizer from a set of quantizers to use for quantizing the wavelet coefficients. The choice of quantizer is based on the methodology used in [2] and the constraints on the quantization noise imposed by the perceptual mask. To effectively utilize the methodology of [2], recall that the coefficients in which the data is to be hidden must be $i.i.d.$ Gaussian random variables. Furthermore, the quantizer must be continuously varied to accomodate the quantization noise constraints. We will address the previous two points in the following subsections.

3.1. Data Modelling

We can divide the wavelet coefficients into groups, with group i representing a realization of a Gaussian random variable with mean μ_i and variance σ_i. The groups of coefficients will correspond to coefficients within the same band of frequencies. From empirical evidence, modelling each group of coefficients as Gaussian random variables is in general fairly accurate. We can then apply the methods of data hiding used in [2], by using a separate encoder for each group of coefficients. The encoder is designed so that the noise which is added to the wavelet coefficients as a result of embedding data will fall below the perceptual mask. This can also be viewed as a power constraint, P_i, on the total amount of noise that can be added to the coefficients of group i. From (3) we know that the maximum amount of data that can be hidden in group i is then:

$$C_i = \frac{1}{2} log(\frac{P_i}{N_i} + 1) \tag{4}$$

where N_i represents the variance of the $Gaussian$ noise that can be added to group i. In general, this can be treated as a water-filling problem, where each channel is independent of the other, and has capacity (4). The total amount of data that can be hidden in the audio signal is then given as:

$$C_{total} = \sum_i C_i \tag{5}$$

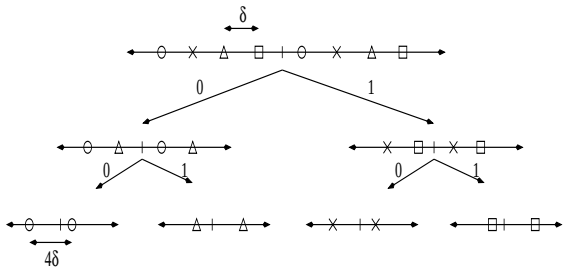

Fig. 3. Codebook represented as a lattice partition: the root codebook specifies the channel code. The leaves of the lattice partition specify the source code. The bits specifying a path to a leaf are the bits to be hidden in the data.

where the sum ranges over all groups of coefficients and the Gaussian noise that corrupts each group of coefficients has variance N_i.

3.2. Code Constructions

One method of encoding for attempting to achieve capacity (see (4)) entails generating a codebook and partitioning the codebook into subcodebooks. The data to be embedded will index a particular subcodebook, and this subcodebook will be used to encode a group of coefficients. In [9], it was shown that one could design a codebook to achieve capacity by distributing the codewords on a hyper-sphere of radius specified by two parameters; the variance of the Gaussian random variable in which the data is to be hidden, σ_i and the power constraint P_i. In practice, the encoding complexity using such a codebook would be exponential and hence impractical. A more practical construction entails taking some code, $\mathcal{C}_0$ and partitioning it into subcodes, using another code $\mathcal{C}_1$. The union of $\mathcal{C}_1$ with its cosets will constitute the subcodes. The message should then have a one to one correspondence with the quotient group $\mathcal{C}_0/\mathcal{C}_1$. It was shown by Forney [11] that the partition can be done in accordance with error correction codes. In this case, the codewords of the error correction code will correspond to the subcode $\mathcal{C}_1$. The cosets of $\mathcal{C}_1$ will then correspond to the cosets of the codewords of the error correction code. As a result, we can represent the messages that are to be embedded in the data as the syndromes of the error correction code. There are computationally efficient ways to calculate the syndrome; hence our method of embedding data will be easy and cost effective to implement. In the above example, we considered a single partition of $\mathcal{C}_0$ using $\mathcal{C}_1$. This is easily generalized into multiple partitions, by further partitioning $\mathcal{C}_1$ using another code $\mathcal{C}_2$ and partitioning $\mathcal{C}_2$ using $\mathcal{C}_3$. This process can continue indefinitely. The message to be embedded will then correspond to the syndrome associated with the group $\mathcal{C}_0/\mathcal{C}_1/\mathcal{C}_2/\mathcal{C}_3/\ldots$ The advantage of having multiple partitions is that a variable number of bits can be easily embedded into the data by using this method. In terms of choosing $\mathcal{C}_0$ and $\mathcal{C}_1, \mathcal{C}_2, \ldots$ we would like $\mathcal{C}_0$ to have a large shaping gain and a large coding gain and for $\mathcal{C}_1, \mathcal{C}_2, \ldots$ to have a large granular gain and a large boundary gain [7] For the interested reader, we provide general codebook constructions in [7].

As an example we consider the case where $\mathcal{C}_0 = \mathcal{Z}$, $\mathcal{C}_1 = 2\mathcal{Z}$, $\mathcal{C}_1 = 4\mathcal{Z}$, and so on, where $\mathcal{Z}$ is the integer lattice. The n-dimensional code will then simply be the product space of the above one-dimensional codes. A representation of this codebook for each dimension is given in Fig. 3. As can be seen from the figure, the codebook consists of lattice partitions which form a binary tree. The leaves of the tree, will represent the subcodebooks that are used for encoding the audio signal. The bits specifying the path to a particular leaf, will specify the bits to be hidden in the audio signal. And, the root of the tree will represent the composite channel codebook that is used at the decoder to decode the hidden bits. As an example, consider the case of the user wanting to hide two bits corresponding to (1,1) into an audio coefficient; the user would then use the right-most codebook (see Fig. 3) to encode the audio coefficient, and transmit the encoded audio coefficient across the channel. The decoder would receive the encoded audio coefficient in addition to some noise and decode the received coefficient to the closest codeword (relative to some distortion metric) in the root codebook. The subcodebook containing the decoded codeword is then found, and the bits that specify the path leading to the subcodebook is declared to be the decoded data bits. One can observe from Fig. 3 that δ, the distance between codewords in the root codebook, governs the amount of noise that the decoder can tolerate from the channel and still recover the hidden bits successfully. For AWGN channels, the probability of bit error can be found as

$$p = Q(\sqrt{\frac{\delta^2}{2N}}) \tag{6}$$

where N represents the variance of the noise from the channel. For n samples, probability of decoding error becomes:

$$P_e = 1 - (1-p)^n \tag{7}$$

Furthermore, one can deduce an estimate of the distortion that is introduced by calculating the expected distortion using the probability distribution of the quantization noise. One should note, that if the channel does not introduce any noise, then δ can be arbitrarily small and the number of levels in the root codebook can be arbitrarily large. In this case, the lattice tree can be made to be infinite, and hence an infinite number of bits can be hidden within an audio coefficient, while meeting the distortion constraint that is imposed by the perceptual mask! We are now equipped with a general method for hiding data in audio.

The general method for hiding data within audio can now be summarized by the following steps (refer to Fig. 2 and Fig. 3) (1) Choose a codebook for a group of wavelet coefficients. Encode this group of coefficients using the subcodebook that is specified by the data to be hidden in the coefficients. (2) Send encoded coefficients across the channel. The decoder will receive the encoded coefficients and decode each group of coefficients using the composite codebook for that group of coefficients. One problem, that the decoder will encounter is that the decoder will not know which codebook was used for encoding which group of coefficients. If we use an n-dimensional lattice partition tree as our codebook, then the encoder can use different levels of the tree for encoding each group of coefficients and send the level of the tree as side information to the decoder. In general, this method of data hiding will require $log_2(n)$ bits of side information per group of coefficients encoded. The througput representing the number of bits hidden within the audio coefficients can be optimized in a rate-distortion sense similar to the work done in [10]. This throughput optimization, however, will also depend upon the amount of channel noise that the decoder is designed to tolerate.

4. APPLICATIONS

Up to now, we have only described the lattice-tree partition as a possible codebook to use for hiding data within audio. Using the principles of [11], one can design better codebooks using similar principles to the tree partition. For example, a trellis codebook that is partitioned into trellis subcodebooks can be designed in a manner similar to designing the lattice tree partition. Using the lattice-tree partition, we found that we could hide 140 kbps of data within CD quality audio (44.1 kHz) without altering the quality of the audio. Furthermore, the hidden bits could be perfectly decoded given a signal-to-noise ratio (SNR) of 15 dB. These results, however, did not accout for the side-information necessary to specify to the decoder which codebook was used to encode which group of coefficients. Accounting for the side-information, we found that we could successfully hide 100 kbps of data without altering the quality of the audio.

One possible applications of our audio data hiding scheme is to hide data within CDs for quality enhancement. Another more exciting application, is to hide data within analog communication channels. To do so, one would send the analog audio through an Analog-to-Digital (A/D) converter and feed the output of the A/D straight to the data hiding system described by Fig. 2. The output of the hiding system is then fed through a Digital-to-Analog (D/A) converter and the output of the D/A is modulated onto the analog communications channel. This application is useful for users that want to receive extra data but do not have the requisite bandwidth for transmitting the extra data. Because we have targeted high-capacity robust data hiding, our method may be used to transmit significant amounts of extra information for various applications.

5. CONCLUSION

In this paper we have introduced a robust method of imperceptible audio data hiding. We developed a method of data hiding that represents the codebook as a tree structure and varies the height of the tree based on perceptual constraints given by the audio signal. Our method of audio data hiding can embed over 100 kbps of data in CD quality audio and still be robust to noise; this is significantly higher than existing audio data hiding techniques in the literature. As a result, we can employ our audio data hiding system in various applications to significantly improve performance.

6. REFERENCES

[1] J. Johnston, "Transform coding of audio signals using perceptual noise criteria," *IEEE Journal on Selected Areas of Communication*, vol. 6, no. 2, pp. 314–323, Feb 1988.

[2] J. Chou, S. Sandeep Pradhan, L. El Ghaoui, and K. Ramchandran, "Solutions to the data hiding problem using distributed source coding principles," *Proceedings of ICIP, Vancouver*, September 2000.

[3] M Costa, "Writing on dirty paper," *IEEE Trans. on Information Theory*, vol. 29, pp. 439–441, May 1983.

[4] C Heegard and A El Gamal, "On the capacity of computer memory with defects," *IEEE Trans. on Information Theory*, vol. 29, pp. 731–739, September 1983.

[5] S Gel'fand and M Pinsker, "Coding for channel with random parameters," *Problems of Control and Information Theory*, vol. 9, pp. 19–31, 1980.

[6] B. Chen and G. W. Wornell, "Preprocessed and postprocessed quantization index modulation methods for digital watermarking," *Proc. SPIE Security and Watermarking Multimedia Contents*, vol. 3971, Jan 1999.

[7] J. Chou, S. Sandeep Pradhan, and K. Ramchandran, "Methods of code construction for channel coding with side information," *In Preparation for submission to IEEE Trans. on Comm.*, 2000.

[8] International Standard, "Coding of moving pictures and associated audio," *ISO/IEC 11172-3*, Aug 1993.

[9] S. Sandeep Pradhan, J. Chou, and K. Ramchandran, "On the duality between distributed source coding and channel coding with side information," *Submitted to IEEE Trans. on IT*, 2000.

[10] P. Prandoni and M. Vetterli, "R/d optimal data hiding," *Proc. of SPIE*, Jan 1999.

[11] G. Forney, "Coset codes - part 1: Introduction and geometrical classification," *IEEE Trans. on Info Theory*, vol. 34, no. 5, pp. 1123–1151, Sep 1988.