# AUDIO DATA HIDING WITH APPLICATION TO SURROUND SOUND

*Jim Chou, Kannan Ramchandran**

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

*Dan Sachs and Doug Jones*

UIUC
Department of ECE
Champaign, IL

## ABSTRACT

There has been a lot of interest recently in applying channel coding with side information (CCSI) concepts [2] to data hiding. Part of the interest stems from the fact that information theoretic bounds can be derived for such systems. In this paper, we model the audio data embedding problem as a parallel CCSI problem. This is done by dividing the audio spectrum into frequency bins and then treating each bin as a separate CCSI channel. A perceptual mask is derived from the audio signal to determine the amount of power to use in each channel. It acts as a "water-filling" formula by determining the amount of distortion that can be introduced into each frequency bin without introducing audible distortion. As a result, our data embedding scheme will be imperceptible to the human ear. An exciting application for our audio data embedding solution is to embed data within the audio signal that will enable surround sound at the receiver. The resulting surround sound system will be better than existing surround sound systems that are based on manipulating two stereo channels to derive the other surround channels.

## 1. INTRODUCTION

An advantage that digital systems have over analog systems is that there exist numerous methods for compressing a digital signal prior to transmission, but there are no suitable methods for compressing an analog signal. As a result, analog systems are often much more inefficient than digital systems. For example, it is well known in the audio compression community that the human ear will be insensitive to noise patterns that are dependent on the tones that are present in an audio signal. A digital system can take advantage of this fact by computing a perceptual mask and then quantizing the audio samples according to the perceptual mask so that the quantization distortion is inaudible. An analog system, however, will transmit the audio signal without using the fact that there exist redundancies in the audio frequency components. In this paper, we propose to exploit the inherent inefficiency of analog systems by imperceptibly embedding additional digital data within the analog data. In particular, we will propose a method of audio data embedding that exploits the fact that the audio data is known prior to embedding. Though audio data embedding has been around for a long time, the majority of prior work has focused on embedding a small amount of data with the goal being that the embedded data should be resilient to hostile removal attempts. In our framework, there will be no hostile attacks but instead there will be a channel with a known probability distribution and the goal of the embedding system is to embed as much data as possible for the given channel. One possible application of our embedding system is to use existing analog communication spectrum for sending extra digital data. For example, additional data can be imperceptibly embedded into frequency modulated (FM) audio that can be recovered at the receiver *even if there is noise in the communications medium*.

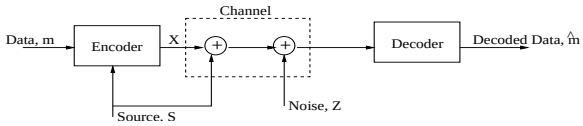


**Fig. 1**. A general model for data embedding. Digital data is encoded into $X$ and is directly added to the source, $S$. The channel noise, $Z$, has power, $N$ and $X$ has power $D$.

There are many challenges in designing a system that can allow high-throughput, reliable delivery of digital data over an audio waveform. The challenges that we address in this paper include (1) constructing a trellis-based code that is flexible in its coding rate for data embedding, (2) using a perceptual mask to adaptively determine the embedding rate and (3) seamlessly conveying the perceptual mask to the decoder. In addition, we will show that there are exciting applications for embedding digital data within audio signals, one of which is, enabling or improving surround sound at the receiver.

*This work was funded in part by an NSF grant: NSF/CCR-0208883.

## 2. PERCEPTUAL DATA EMBEDDING

It has recently been proposed [1] that the data embedding problem can be modeled as in Fig. 1. For this model, the source, $S$, is assumed to be additive noise that is available to the encoder. The output of the encoder, $X$, is constrained to have power $D$, and is sent through the channel consisting of the concatenation of the additive noise $S$, and some additive noise $Z$. If we assume that both $S$ and $Z$ are $i.i.d.$ Gaussian, then it has been shown by Costa [2], that the information-theoretic capacity of such a system is

$$C = \frac{1}{2} log \left(1 + \frac{D}{N}\right) \quad (1)$$

where $N$ is the variance of the noise. When the data is to be embedded over a channel that is bandlimited to frequency, $W$, it can be shown that the capacity of such a system becomes:

$$C = W log \left(1 + \frac{D}{N}\right) \quad (2)$$

To effectively use the above results, we choose to model data embedding for analog audio sources as a set of parallel AWGN channels (see Fig. 2). There are several benefits to doing this. For one, audio signals allow for varying amounts of distortion in different frequency bands and thus, one would expect that different frequency components will allow for different embedding rates (see (2)). Furthermore, the noise is not necessarily white. For example, in an FM communication system, the noise can be modeled as additive colored noise, where the noise spectrum assumes the following shape [3]:

$$S_n(f) = \begin{cases} \frac{N_0 f^2}{A_c^2} & |f| \leq W \\ 0 & |f| > W \end{cases}$$

Thus, we propose to transform the audio signal into its frequency components by taking a wavelet transform of the audio signal, then dividing the set of all frequencies into many smaller frequency bins, and treating each frequency bin as a separate channel (see Fig. 2). The amount of power (i.e., signal distortion) used in each frequency bin for digital transmission is determined by a perceptual mask that can be derived from the audio signal. The perceptual mask represents the maximum amount of distortion that is allowed in each frequency bin without it being audible to the human ear. It is typically calculated by using the fact that tones in the audio signal will mask out noise that occurs in surrounding frequencies [4] if the tone is much greater than the surrounding noise. On the other hand, if the surrounding noise is much greater than the tone, then the noise may mask the tone. Thus, the perceptual mask provides us with a prescription of the amount of power that should be used

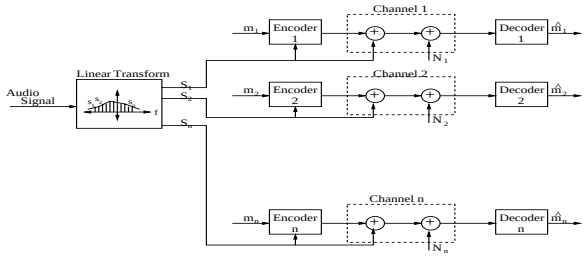


**Fig. 2**. Modeling the audio data embedding problem as a parallel AWGN channel. The audio is transformed into the frequency domain, and the set of all frequencies is partitioned into frequency bins, with each frequency bin corresponding to a separate channel.

for digital transmission in each frequency band. The total capacity of this system will be:

$$C = \sum_{i=1}^{n} \frac{W}{n} log \left(1 + \frac{D_i}{N_i}\right) \quad (3)$$

where we assume that there are $n$ equal-sized frequency bins and $D_i$ and $N_i$ represent the distortion and noise variance in bin $i$ respectively.

In the next section, we will look into practical code designs that attempt to achieve the bound provided in (3).

## 3. CODE CONSTRUCTIONS

There has been alot of interest recently in designing codes to try and achieve the capacity of (3) [5, 6, 7]. All of these codes are designed for a fixed encoding rate, however, and are therefore not practical for our problem, which demands a flexible encoding rate that is determined by the perceptual mask. Furthermore, in order for the decoder to be able to extract the embedded data, it must also have knowledge of the perceptual mask that the encoder used to encode the data. In this section, we propose a trellis-based construction that allows for embedding of both the data and a description of the perceptual mask.

### *3.0.1. Trellis construction*

To design a good data embedding code requires the observation that the encoder resembles a source encoding operation and the decoder resembles a channel decoding operation. More specifically, in Fig. 1, we can see that the goal of the encoder is to embed the data by perturbing the source by some distortion, $X$, and the goal of the decoder is to recover the data by correcting for the channel noise, $Z$. One may therefore obtain a code construction by partitioning a channel code into $2^{LR}$ different source codes, where $L$ is the number of samples to be encoded. The data to be embedded, will index one of the $2^{LR}$ different source codes to encode

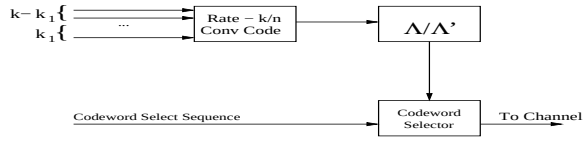


**Fig. 3**. Trellis construction of a data embedding code. The channel code is given by the trellis code based on the rate-$\frac{k}{n}$ convolutional code. The source code partitions are given by the trellis codes based on the rate-$\frac{k-k_1}{n}$ convolutional codes.

the source, and each of the $2^{LR}$ source codes is designed to produce distortion that is less than or equal to $D$. The decoder may recover the data by decoding the channel output in the composite channel codebook and then declaring the index of the partition that the decoded codeword belongs to as the decoded data. The total embedding rate for this code will be $R$.

The code construction that we propose, is a trellis-based channel codebook that can be partitioned into several source code partitions that are also trellis codes (see Fig. 3). To do this, we first represent the trellis code [8] as a rate-$\frac{k}{n}$ convolutional code that indexes a lattice partition, $\Lambda/\Lambda'$. An appropriate codeword can then be selected from a coset of $\Lambda'$. The partition is constructed by reserving $k_1$ of the $k$ input bits to the convolutional code for the data bits. The $k_1$ data bits will effectively select a trellis code based on the resulting rate-$\frac{k-k_1}{n}$ convolutional code to encode the source.

The encoder will therefore embed $k_1$ bits per $M$ samples, where $M$ is the dimension of $\Lambda'$, by selecting a rate-$\frac{k-k_1}{n}$ convolutional code and then Viterbi decoding the source with respect to the resulting trellis code. Note that because $\Lambda'$ contains an infinite set of codewords, *the trellis will cover all of Euclidean space* and there will be an infinite set of parallel transitions in each branch of the trellis. As a result, a modified Viterbi decoder must be used to account for the infinite set of parallel transitions. The decoder will also use this modified Viterbi decoder to decode the channel output to the closest codeword in the trellis derived from the composite rate-$\frac{k}{n}$ convolutional code. The $k_1$ bits that are associated with the decoded codeword are declared the decoded message bits.

### *3.0.2. Perceptual mask embedding*

To embed data in an audio signal, a perceptual mask is first derived from the audio signal. The perceptual mask is derived by first converting the audio signal into the frequency domain, and then determining the significant masking components. These masking components will be convolved with a spreading function to form the perceptual mask. The perceptual mask can be used with the above trellis construction to determine the number of bits to embed in each audio frequency component. The reason for this is because the quantization distortion varies directly as a function of the number of bits that are embedded; when few bits are embedded, the resulting trellis code will give small distortion, and when many bits are embedded, the resulting trellis code will give large distortion. Thus, a variable number of bits are embedded into each frequency band for a given time segment. The decoder must also have access to the perceptual mask in order to determine the number of bits to extract from each frequency band. Naively, one might think that the decoder can calculate its own perceptual mask and use this mask to recover the data. When the bits are embedded, however, the perceptual mask is also altered. Thus, in order for the decoder to be able to recover the data, the encoder must convey the perceptual mask to the decoder. We propose to do this in a seamless manner.

If we denote the perceptual mask of the original audio signal as $p_1(f)$ and the perceptual mask of the audio signal after data has been embedded as $p_2(f)$, then we can leverage the fact that $p_1(f)$ and $p_2(f)$ are correlated. Now, the decoder will be able to calculate $p_2(f)$ on its own, so the problem becomes one of encoding $p_1(f)$ given that there is side information, $p_2(f)$ at the decoder that is correlated to $p_1(f)$. We can, therefore, use results on source coding with side information [9, 10] to encode $p_1(f)$. A codebook for source coding with side information will consist of a composite source codebook that partitions channel codebooks. The encoder will then encode the source (in our case, $p_1(f)$) using the composite source codebook and determine the channel code partition to which the resulting codeword belongs. The index of this partition is transmitted to the decoder and the decoder will use this index to specify the channel codebook for channel decoding the side information. The resulting compression rate of this code will be $R$.

The encoder can now convey the perceptual mask to the decoder by first compressing the perceptual mask, $p_1(f)$ and then embedding the resulting bits by using the data embedding scheme. The encoder and decoder will agree upon the number of bits to be used for encoding the perceptual mask beforehand. The decoder can now recover the perceptual mask by decoding the channel output and extracting the $k$ bits that correspond to the decoded codeword. Of these $k$ bits, a fixed amount, $k_2$ bits, are used to channel decode the perceptual mask that the decoder calculates. This should give the original perceptual mask, from which the decoder can determine the amount of bits from the $k - k_2$ bits that are actual data.

## 4. APPLICATION FOR SURROUND SOUND

Finally, we used the data hiding to create a sample application: using the hidden data to improve channel seperation in

Dolby Surround streams.

Dolby surround attempts to encode four channels of audio data (left, center, right, and surround) into two channels. [11] This is done by encoding the center channel equally into the left and right channels, and the surround channel into the left and right channels with +90 and -90 degree phase shifts, A passive Dolby decoder sends the left channel to the left speaker, right channel to the right speaker, one-half times the sum of the left channel and the right channel to the center, and the difference between the left and right channels, delayed and filtered, to the surround channel.

Because the use of passive decoding permits only 3 dB of seperation between output channels, typically active active decoding ("Dolby ProLogic") techniques are used. Active decodoing modifies the gains of each output channel from the passive decoder using heuristic techniques, attempting to increase seperation between channels.

Data hiding permits us to improve these heuristic techniques with hidden information. Rather than using the analog waveforms to attempt to reconstruct the original gains, we can simply send gain vectors for each of the four original channels in the hidden data. This does not interfere with existing Dolby decoders, but allows improved directionality when the gain information encoded in the hidden data streams are used. It also permits us to create two rear channels with independent gains, allowing for panning surround effects. For our test application, 8-bit gain vectors were sent every 30ms for each of the five output channels. (The left and right surround channels are generated using gains from the passive decoder's Surround output.) To prevent the 30ms sampling period from causing audible noise, the gains are linearly interpolated between sample points. The encoder sets the gains to minimize the energy difference between the original input to each of the five channels and the output from the decoder.

The small amount of hidden data in this application—approximately 1.3 kbits/sec—is imperceptible, and the added data allows for effects, such as rear panning, impossible with Dolby's decoders. In addition to allowing for effects that are otherwise unachievable in two-channel systems, this system also allows us to improve the apparent seperation between channels.

Embedding data to improve surround sound can certainly be used on various mediums such as a compact disc (CD) or an FM channel. At an embedding rate of 1.3 kbps, not only will the embedded data be imperceptible to the human ear, but it will also be robust to channel noise. For a variety of audio signals that we used in our experiments, 3 dB of signal-to-noise ratio (SNR) is sufficient for achieving such embedding rates. For CDs the SNR is typically very high, because there are error correction codes in the CD that provide for a clean channel. An FM channel, however, will typically have much lower SNRs but the SNR is usually above 3 dB. As a result, a tantalizing use of data-embedding for surround sound might be to enable surround sound for FM audio.

## 5. CONCLUSION

In this paper, we have derived a bound on imperceptibly embedding data into an audio signal that will be robust to channel noise. We proposed a variable-rate trellis construction that can be used in conjunction with a perceptual mask to try and achieve this bound. We also showed how source coding with side information can be used to seamlessly convey the perceptual mask from encoder to decoder. To conclude, we showed that audio data embedding can be used to improve or enable surround sound at the receiver. Future work remains in investigating synchronization and modulation issues for embedding digital data over analog communication channels.

## 6. REFERENCES


[1] 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.

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

[3] S Haykin, *Communication Systems*, Wiley, New York, 1994.

[4] 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.

[5] J. Chou, S. Sandeep Pradhan, and K. Ramchandran, "Turbo-coded trellis-based constructions for data embedding: Channel coding with side information," *Proceedings of Asilomar Conference on Signal and Systems, Pacific Grove*, November 2001.

[6] M. Kesal, M. Mihcak, R. Koetter, and P. Moulin, "Iteratively decodable codes for watermarking applications," *Int. Symp. on Turbo Codes, Brest France*, September 2000.

[7] J. Eggers, J. Su, and B. Girod, "Performance of a practical blind watermarking scheme," *Proceedings of SPIE*, January 2001.

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

[9] A. D. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. on Inform. Theory*, vol. IT-22, pp. 1–10, January 1976.

[10] S. S. Pradhan and K Ramchandran, "Distributed source coding using syndromes: Design and construction," *Proceedings of the Data Comression Conference (DCC)*, March 1999.

[11] R. Dressler, "Dolby surround pro logic ii principles of operation," *http://www.dolby.com/tech/l.wh.0007.PLIIops.html*.