Real-Time Implementation for Digital Watermarking in Audio Signals Using Perceptual Masking

Tal Mizrahi, Eran Borenstein, George Leifman, Yuval Cassuto, Michael Lustig, Shay Mizrachi, and Nimrod Peleg.
Signal and Image Processing Lab., Dept. of EE, Technion, Haifa, Israel.
www-sipl.technion.ac.il

Abstract

The advancement of digital audio enables the reproduction of copyright-protected media without any degradation in audio quality. This causes severe problems of copyright violations and emphasizes the need for copyright protection mechanisms.
Digital watermarking is one of the most popular copyright protection mechanisms.
In this paper we present a novel real-time signature embedding system for embedding of digital watermark in audio signals. The signature detection is done using off-line Windows 9x/NT application (this application enables embedding and detection of digital signature). The embedding mechanism enables an owner to insert into its own media a digital watermark in such a way, that the audio quality is not reduced. This is done, by using the human auditory system's masking characteristics. The detection mechanism enables the owner to check for the existence of the watermark in a tested media. The presented system solves the deadlock problem (multiple ownership claims) by keeping the original media or parts of it for future ownership claims.

1. Introduction

Recent rapid development in the field of digital media has raised the issue of copyright protection. Digital watermarking offers a solution to copyright violation problems [3].
The watermark is a signature, embedded within the data of the original signal, which in addition to being inaudible to the human ear, should also be statistically undetectable and resistant to any malicious attempts to remove it. In addition, the watermark should be able to resolve multiple ownership claims (known as the deadlock problem), by using the original signal (i.e., the unmarked signal) in the signature detection process.
In order to meet the above demands, perceptual masking is used [1,2], both in the frequency domain (using a psycho-acoustic model) and in the time domain. The added signature is signal dependent, and thus is inaudible as well as robust enough to survive attempts to destroy it.

The audio signal is divided into segments. For each segment a local key is calculated and summed up with a general key (independent of the segment) to initiate a pseudo-random noise sequence for the segment. The noise is colored by a filter, whose coefficients are calculated according to the psycho-acoustic model.
After applying a temporal mask (in order to reduce the pre-echo effect), the colored noise becomes a watermark.

The work done includes a Matlab™ Simulation, a watermark insertion-verification 'C' code and a real-time implementation of the insertion mechanism, running on TI TMS320c54/55 DSP.

The paper is organized as follows: after this short introduction, the algorithm steps are presented, followed by special consideration regarding the real-time implementation on TI's TMS320c54/55 DSP.

2. Signature embedding algorithm

The embedding mechanism enables the owner to insert a signature into his original media. Figure 1 presents the embedding scheme. The resulting signature is inaudible because it is using the HAS – Human Auditory System masking characteristics [4], both in time and frequency domains. The embedding algorithm steps are as follows:

a. The audio file is divided into segments, and the watermark is calculated for each segment individually.
b. For each segment a local key is calculated, using a hash function on the sum of the samples in the segment. The key is unique for each segment, and is highly sensitive to any slight change in the signal.

c. The local key is combined with the owner's key, which is a serial number, to create a combined key. This key is used to initialize a PN generator, which produces a pseudo-random noise sequence of samples of the length of the segment.

d. A masking threshold is calculated in the frequency domain, using a psychoacoustic model:
   - The spectrum of the signal is calculated, using FFT.
   - The hearing threshold in silence is used to initialize the masking threshold.
   - Tonal and non-tonal components are calculated for the segment.
   - For each component, an individual masking threshold is calculated.
   - The individual thresholds are summed up and added to the threshold in silence to create the global masking threshold.

e. The masking threshold is used as a filter to color the pseudo-random noise sequence. The colored sequence is then scaled to be below the threshold. The outcome is an inaudible noise sequence.

f. In order to avoid the pre-echo effect, the temporal post-making effect is taken advantage of. The noise is compared to a masking threshold in the time domain, and the watermark level is set to the minimum between the two.

g. The watermark is then added to the original signal, to create the watermarked signal.

3. Detection Algorithm

The detection mechanism enables the owner to check for the existance of his signatue in a tested media. Figure 2 presents the detecion scheme. The deadlock problem, i.e., multiple ownership claims, is solved by using the original media in the detection process. The main idea is that the signature is created using both the owner key and the original media. The main steps of this mechanism are:

a. The original signal (i.e. before it was watermarked) and the owner's key are used to recalculate the signature using the above algorithm.

b. Using the original signal, the tested signal is processed to match its energy with the original signal energy in order to deal with gain modifications of the signal.

c. The original signal is subtracted from the tested signal, to create the tested watermark.

d. The correlation between the tested watermark and the recalculated watermark is calculated, and compared to a threshold, to determine if the watermark was detected or not.

4. Real-Time Implementation

4.1. The need for real-time

A real time implementation is needed in the following applications:

1. Selling music files on the Internet, and while sending the file, embedding it with a signature containing the new owner's key, transparently to the customer. Thus not compelling additional waiting time.
2. Another usage may be embedding the signature into live broadcasts where it can not be done offline. For example, concerts, Internet radio, conferences and media news scoops.

4.2. Why a DSP ?

The algorithm is computationally very heavy, consuming much CPU time. The Windows application processing time is 8 times the actual signal time (at 44.1kHz sampling rate). This ratio makes this application inappropriate for the above usage. Being aware of this, we examined the algorithm and found that the blocks consuming most computations are the FFT, noise filtering and creating the PN sequence. A suitable platform for these operations is a DSP.

4.3. Selected platform characteristic

The TMS320C54x [5] is a 16 bits processor. This does not mean that calculations are limited to 16 bits. The adder unit and accumulators are 40 bits wide and the multiplier gives a 32 bit result which can be stored either as a 16 MSB or, for extended precision, in 2 words of 16 bits.

4.4. Fixed point arithmetic

One of the main difficulties we have encountered is that the TMS320C54x is a fixed-point arithmetic processor. This enforces limited and varying precision. Every calculation must be examined and fit to the appropriate representation range in order to achieve maximum precision. Finding the best range and than normalizing numbers to this range is a single cycle operation using the EXP encoder unit of the C54x.

4.5. Division operation optimization

Implementation of division is very complicated, because contrary to adding, shifting and multiplying the DSP doesn't have a special unit for division. A division operation requires repeating conditional substractions with carry. Changes in the algorithm were made to optimize divisions, such as using powers of 2 divisions and modulo operations. That way we are able to substitute a bulk number of divisions by binary shifts which requires no additional cycles (the shifter unit can work in parallel to any other execution).

For example, the Windows application process averages 10 samples of the PN-26 sequence to get one sample of the pseudo-noise. The real-time application uses 8 samples instead of 10 and replaces the division by 10 operation with a shift right operation with an argument of 3.

4.6. Parallel execution

Noise filtering (coloring the white noise), which is an important and cycle consuming phase of the algorithm, is done very efficiently using the multiply-accumulate (MAC) instruction. This instruction (when done in a repeat mode) can generate addresses of a coefficient and a the sample, multiply them and add the result to the accumulator – all in a single cycle, using multiple busses and separate arithmetic units.

The DSP architecture enables several memory accesses in a single cycle.

Whenever possible, multiple access addressing mode was used.

4.7. Memory management

Fast dual access memory is limited. Data buffer recycling was used to optimize the strict storage space. Moreover a cyclic addressing mode is used whenever large buffers are accessed, to minimize the need of copying and saving precious dual-access RAM storage space.

4.8. Matching the psycho-acoustic Model to DSP architecture

The psycho-acoustic model used in the algorithm represents energy levels in decibels. The fact that logarithm calculation is done by Taylor polynomial approximation of the function ln(1+x) suggests that using the natural base logarithm, instead of base-10, will save a multiplication by a constant for every sample. We adjusted therefore all model tables accordingly.

5. Results

In preliminary run-time tests of the main blocks of the application, we got a rough estimation of 20 million cycles, which means that using a C54x 's 100 MIPS real time is signature embedding achievable.

6. Conclusions

The possibility of real-time was verified, nevertheless we should further consider the following:

1. Stereo channel embedding or concurrent multiple signature embedding for various customers, each one with a different owner's key, may require a faster DSP. The code compatible TMS320C55x can then be used.
2. The connection channel between the HOST-PC and the EVM (evaluation module) is based on the PC parallel port. This connection reduces the data-transfer rate and may be the bottleneck of the complete system.

7. Acknowledgment

The authors would like to thank the head of the lab, Prof. David Malah, for his help and support. The authors are also thankful to the support provided by TI through the Elite program.

8. References

[1] M.D. Swanson, M. Kobayashi and A.H. Tewfik, "Multimedia Data-Embedding and Watermarking Technologies", *Proc. of IEEE, Vol. 86, NO. 6, June 1998.*

[2] M. D. Swanson et al., "Robust Audio Watermarking using Perceptual Masking", Signal Processing 66, 1998

[3] W. Bender, D.Gruhl and N. Morimoto, "Techniques for Data Hiding", *IBM System Journal, Vol.35, NOS 3&$, 1996.*

[4] E. Zwicker, G. Flottorp and S.S. Stevens, "Critical Band Width in Loudness Summation", *The Journal of the Acoustical Society of America, Vol.29, No.5, 1957*

[5] TMS320c54x DSP volume I: CPU and Peripherals Reference Set, Texas Instruments, 1997

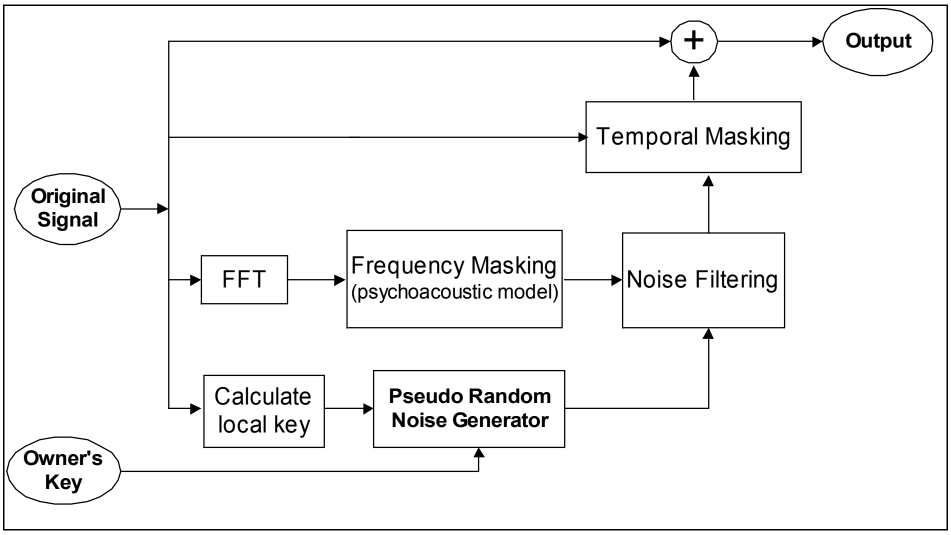

Figure 1: Embedding scheme

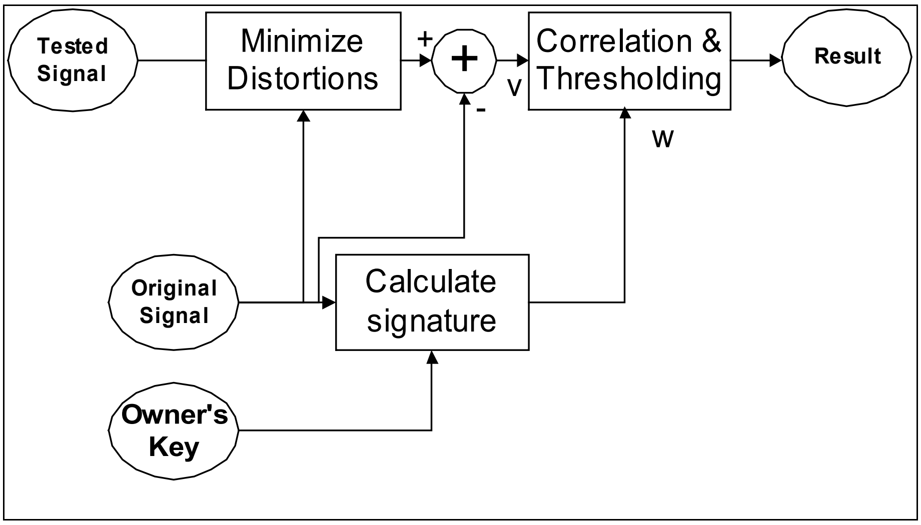

Figure 2: Detection scheme