AUDIO WATERMARKING BY TIME-SCALE MODIFICATION

Mohamed F. Mansour and Ahmed H. Tewfik
Department of Electrical and Computer Engineering, University of Minnesota,
Minneapolis, MN 55455
{mmansour, tewfik} @ ece.umn.edu

ABSTRACT
A new algorithm for audio watermarking is proposed. The basic idea of the algorithm is to change the length of the intervals between salient points of the audio signal to embed data. We propose several novel ideas for practical implementations that can be used by other watermarking schemes as well. The algorithm is robust to common audio processing operations e.g. mp3 lossy compression, low pass filtering, and time-scale modification. The watermarked signal has very high perceptual quality and is indistinguishable from the original signal.

1. INTRODUCTION

Watermarking techniques provide tools for copyright protection of digital media. It has become a hot research area because of the increasingly demand on digital media distribution through the internet.
Most audio watermarking techniques are based on either spread spectrum methods or changing the least significant bits of selected coefficients of a certain signal transform. To ensure watermark imperceptibility, the audio masking phenomena is exploited by these techniques either explicitly or implicitly [1].
A watermarking system should be robust to common signal processing operations that the signal may undergo. Some of which are lossy compression, low pass filtering, and time scale modification. In addition the system should be transparent, i.e. the watermarked signal should be indistinguishable from the original one.
In this work we propose a new idea for audio watermarking different from common embedding approaches. The intervals between salient points of the audio signal are modified to embed data. In this work, we choose the non-orthogonal wavelet extrema of the signal envelope as our salient points.
The paper is organized as follows. Section 2 describes the embedding and extraction algorithms. Section 3 discusses several practical issues related to the implementation. Section 4 gives the experimental results of the algorithm.

2. ALGORITHM

2.1 Redundant wavelet decomposition
Unlike the common orthogonal wavelet transform, the redundant wavelet transform does not involve subsampling after filtering at each scale. In our system, we use the derivative of the *cubic spline* as our wavelet basis [2]. This basis is the first derivative of the cubic spline smoothing function. Hence a local extrema in the wavelet coefficients represents a maxima or minima of the derivative of the original signal. So the extrema locations are actually the locations of the signal edges. An edge in the audio signal represents either a transition from silence to voice activity or a change in the dominant tone. The redundant wavelet extrema have many other interesting features ([2], chapter 6).

2.2 Embedding Algorithm
The basic idea of the data embedding algorithm is to quantize the intervals between *selected* wavelet extrema and force the quantization index to be either odd or even according to the input data. The overall system is shown in figure 1.

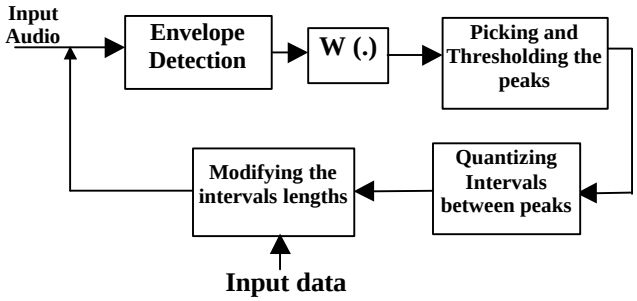

Figure 1. Embedding Algorithm

The embedding algorithm has the following broad steps:
1. The envelope of the input signal is calculated by low pass filtering the full-wave rectified signal.
2. The non-orthogonal dyadic wavelet decomposition of the envelope is calculated using the derivative of cubic spline wavelet basis (typically 10-11 dyadic scales are needed).
3. The coefficients of the coarsest scale are used for subsequent operations. We will denote these coefficients as *C(t)*.
4. The power of *C(t)* is normalized to a predefined value and the extrema are picked.
5. The extrema are then refined so that only *strong* ones are preserved. The practicalities of thresholding are described in the next section.
6. The lengths of the intervals between successive extrema are modified to embed data. One bit is embedded per interval. First, we define a quantization step Δ. Each interval is quantized with this step, and the quantization index is forced to be either odd or even to embed one or zero (respectively). The procedure for changing the interval length is described in the next section.
7. Steps 1 through 5 are repeated until convergence (typically 3 iterations are needed).

2.3 Extraction Algorithm
The extraction algorithm is straightforward. The first five steps of the embedding algorithm are repeated in exactly the same way. The lengths of the intervals between successive extrema are quantized and the decision rule is:

$$d(i) = \mathrm{rem}(\lfloor d_i/\Delta \rfloor, 2) \quad (1)$$

Where $d(i)$ is the i^{th} extracted bit, $\mathrm{rem}(.)$ is the *remainder* function, d_i is the length of the i^{th} interval, Δ is the quantization step, and $\lfloor . \rfloor$ is the *floor integer* function.

3. PRACTICAL ISSUES

3.1 Envelope Detection

The envelope of the signal is used rather than the signal itself because it is relatively invariant to the change in the instantaneous frequency. The sharp transitions in the envelope are basically due to silence/voice transition. On the other hand, if the original signal is used directly, wavelet extrema exist at points of discontinuities, and these discontinuities may arise either from silence/voice transition or from a change in the instantaneous frequency. After compression, the change in the envelope is very small compared to the signal change. In addition, for stereo signal the difference in the envelope of the two channels is typically very small compared to the signals themselves.

The envelope is calculated by low pass filtering the full-wave rectified signal. The typical bandwidth of the envelope is 5-20 Hz, and because of this low-pass content, only the wavelet coefficients at the coarsest scales are considered in subsequent processing. The wavelet coefficients at fine scales have negligible power.

3.2 Peaks Detection

If we assume that the input signal $f(t) \in L^2(R)$ and is uniformly *Lipchitz* α during its interval, then the amplitude of the wavelet coefficients across scales is bounded by [2]:

$$|Wf(\tau,s)| \leq A\, s^{\alpha+1/2} \qquad (2)$$

where A is a constant greater than zero, and '*s*' increases with coarser scales. i.e. the upper bound of the amplitude of wavelet coefficients increases exponentially as we move to coarser scales. As a result, the peaks at coarser scales are more distinctive. In our system, because the envelope has a very low pass frequency content, the fine scales coefficients have negligible power and the power is much concentrated at coarser scales. For the cubic spline wavelet it was found that, the scale 2^{11} has the most distinctive peaks for an envelope with 10 Hz bandwidth.

3.3 Thresholding

As usual with systems that incorporate thresholding, the selection of the threshold value has a crucial impact on the system performance. For our system, false alarms or missed peaks may occur if the threshold is not chosen carefully. These errors lead to misalignment of the extracted data.

We propose a solution for estimating the threshold so as to minimize the detection and false alarm errors. The threshold is selected from the candidate peaks such that there is a *guard band* above and below the threshold where no peak exists. The size of the guard band is a trade-off between the size of embedded data and the required accuracy. Large guard bands give higher accuracy at the expense of reducing the number of peaks (reducing embedding capacity), and vice versa. Practically, the guard band is selected to be a fraction of the standard deviation of the peaks amplitudes. To accommodate for possible amplitude scaling, the wavelet coefficients are normalized to a reference power value.

The straightforward choice for the peaks is those with highest amplitude. However, It was noticed that such peaks might be concentrated in a small portion of the signal. A possible solution is to rectify any peak if it is close to a stronger (in amplitude) one. However, this approach is not very robust under *mp3* compression, as strong peaks tend to move after filtering.

The ultimate choice for peaks refinement is to select strong peaks at the onset of strong signal activity (following silence or low power activity). Two thresholding phases are incorporated. The first phase is based on the peak amplitude where only the strongest peaks are preserved. In the second phase, only the peaks at the onset of signal activity are preserved. Hence, the refined peaks are the strong ones, which are not preceded by any other peaks within a predefined interval.

3.4 Modifying the interval length

The quality of the output signal depends completely on how the intervals are modified. Modification of the lengths should be accomplished carefully to avoid discontinuities, which may result in clicks in the synthesized output.

The lengths are modified to ensure that the interval length lies in the middle of the correct quantization slot. Odd quantization indices are used to embed "1", and even ones are used to embed "0". For example, if the quantized value of interval length does not match the corresponding data bit, then the interval length is increased to be in the middle of the next quantization slot, and if the quantization index matches the corresponding data bit, then the interval length is moved (either by increasing or decreasing it) to the middle of the current slot.

We tested a simple, yet effective in many cases, approach for this modification. The intervals between successive extrema are segmented into several segments of lengths equal to the required modification length, which can be either positive or negative. Then, the powers of all these segments are calculated, and the minimum power segment is picked. If the interval is to be lengthened, then this segment is copied immediately after its end, and if the interval is to be shortened then this segment is removed. This algorithm works well if the audio input contains large silence periods, where changes in these periods are quite inaudible. However, if the intervals between successive peaks do not include any silence periods, then the above algorithm results in audible glitches due to the discontinuity of the synthesized output.

Another approach for interval modification is to use time-scale modification (*TSM*) using *overlap-add* techniques [3]. We adopt the use of *WSOLA* technique for time-scale modification. The intervals between extrema are slightly expanded or compressed. We used a *Hanning* window with 50% overlapping for synthesizing the output. The reference points (in the input) for the window centers (in the output) are determined using a piecewise linear function between the original locations of the peaks and their modified locations.

It should be mentioned that, any other *TSM* algorithm could be used as well. The better the performance of the modification algorithm the better the output quality. A remarkable *TSM* algorithm was proposed in [4].

3.5 Adaptive Quantization Step

To account for possible time scaling, the quantization step "Δ" should be signal dependent. If the signal undergoes time scaling both the intervals between the peaks and the quantization step will change by the same factor leaving the quantization indices unchanged. The estimation of the quantization step must be robust to any possible signal processing operation because of its crucial effect on subsequent decoding operations. An error in estimating the quantization step will result in decoding random data even if the peaks are extracted correctly.

In our system, instead of using a single quantization step for all slots, we use a quantization step that is equal to a certain fraction of the previous slot length (after modification). For example, if the interval after modification, between the i^{th} and $(i+1)^{st}$ peaks is L, then the quantization step of the interval between $(i+1)^{st}$ and $(i+2)^{nd}$ peaks will be αL where α is a predefined constant (typically 0.05-0.1) , and in general,

$$\Delta_{i+1} = \alpha.(\lfloor d_i/\Delta_i \rfloor + 0.5).\Delta_i \qquad (3)$$

Where D_i is the quantization step for the i^{th} slot, and d_i is the length of the i^{th} interval. *However*, There are two problems with the above algorithm. First, the quantization step of the first segment must be determined accurately. Second, an error in one slot can propogate to the next.

We propose another algorithm for the above problem that works quite well with resampling. At the decoding stage we search for the *best* quantization step among all possible candidates. Exhaustive search is conducted if there is no a priori information about the scaling factors. However, if such information exists, then the problem is reduced to selection rather than estimation. The best quantization step is the one that minimizes the corresponding quantization error, i.e.

$$\tilde{\Delta} = \arg \min_{\Delta} \left(\frac{1}{\Delta} \sum_{i=1}^{B} \left| d_i - Q(d_i) \right| \right) \qquad (4)$$

$$\text{where} \quad Q(d) = \Delta.\left(\left\lfloor \frac{d}{\Delta} \right\rfloor + 0.5 \right) \qquad (5)$$

Where B is the total number of intervals, d_i is the length of the i^{th} interval, and Q(.) is the quantization function. It should be mentioned again that, at the embedding stage the intervals are forced to be in the middle of the corresponding slot. At the extraction stage, the best peak is the one that minimizes the quantization error. This step is used for subsequent decoding of data.

3.6 Analysis of added samples

The average number of added samples depends on the quantization step Δ. New samples are added in two cases: when the quantization index has a wrong value or when it has the correct value but the interval length is less than the middle of the quantization slot. If we assume Δ is fixed, then in the former case, the average added samples is Δ, while in the later case the average added samples is $\Delta/4$. Samples are removed in only one case: when the quantization index has the correct value but the interval length is higher than the middle of the quantization slot. In this case the average removed samples are $\Delta/4$. If we assume equal probabilities of being in the correct or the wrong slot and equal probabilities of being the lower half or the upper half of the correct slot, then the average added samples per bit is $\Delta/2$. Hence the total increment in the signal length is $BD/2$, where B is the total number of embedded bits.

For adaptive quantization step, where the quantization step is a fraction of the previous slot, the same procedure holds. In this case Δ is replaced by the *average* quantization step, which is defined as:

$$\overline{\Delta} = a.\overline{L} = a.\frac{L_{tot}}{B}$$

Where L_{tot} is the total signal length. Hence the total increase in adaptive quantization case is $aL_{tot}/2$.

3.7 False Alarms Detection

The structure of the embedding system allows self-detection of false alarms that may arise. This will help in better aligning the output data stream.

Theoretically speaking, the interval lengths should be in the middle of each quantization slot. Hence for a perfect detection system, the quantization error in the detection stage should drop to zero. For each extracted peak the quantization error is a measure of confidence about the correctness of this peak.

If a false alarm exists, it will lie at an arbitrary location between two correct peaks. It splits the correct slot into two parts with relatively high quantization error in each. Hence for each extracted peak, the ratio between the minimum quantization errors of the two surrounding slots, and the quantization error of the larger slot between the previous and the next peaks is a good measure of confidence about the correctness of this peak. Hence if d_i is the length of the i^{th} interval, then our measure of confidence about the i^{th} peak is:

$$C = \frac{\min\left(\left| d_i - Q(d_i) \right|, \left| d_{i-1} - Q(d_{i-1}) \right| \right)}{\left| d_i + d_{i-1} - Q(d_i + d_{i-1}) \right|} \qquad (6)$$

Where Q(.) is defined in (5). The smaller the value of C the higher the confidence about the correctness of the corresponding peak. If this value is greater than a certain threshold (typically 5), then this peak is most probably a false alarm and should be removed.

In the above analysis we assume that the erroneous peaks are primarily due to false alarms rather than localization error, and this assumption is justified experimentally.

3.8 Stereo Embedding

For stereo signals, the bit rate can be doubled by embedding different streams in the two channels. However it is always desirable to keep a fixed lag between the two channels as time increases, and embedding different bitstreams in the two channels may result in a noticeable lag between them.

The redundancy of the stereo signal can be exploited for error detection. Typically the peaks of both channels exist at the same *relative* locations. Hence, the quantization indices of corresponding intervals in the two channels are usually equal. Consequently, during decoding, if the quantization indices are the same for both channels, then the corresponding bit is decoded without problems. However, if the indices are different for the corresponding intervals due to some error in locating the peaks, we pick the one with smaller quantization error. The correct interval is used for decoding the following bit in the correct channel, while the following interval in the wrong channel is ignored. One interval after, the decoding becomes simultaneous again.

The redundancy can be also used to remove false alarms. It is very unlikely to have two false peaks that appear in both channels simultaneously. If a peak appeared in one channel while there is no corresponding peak in the other, then we have two possibilities. The first possibility is that this peak is a false alarm, and the second possibility is the absence of the correct peak in the other channel. To decide we use the technique described in the previous section for false alarm detection. If the confidence measure of the existing peak is above a certain threshold then it is classified as a false alarm, otherwise it is classified as a correct peak.

4. RESULTS

The algorithm was applied to a set of test audio signals. The lengths of the sequences were around 50 seconds. The test signals include pure music, pure speech, and combined music and speech. The test signals are either mono with sampling rate 22.05 kHz or stereo with sampling rate 44.1 kHz.

The first series of tests aim to check the validity of the basic algorithm (with fixed quantization step) for noiseless channel. The algorithm works perfectly, and no errors are encountered for any sequence.

Next, we test the performance of the system against mp3 compression. The results are given in Table 1, where the comparison is based on the number of correct peaks. Here the test signals are mono with sampling frequency 22.05 kHz, and the bandwidth of the envelope filter is 10 Hz.

Bitrate Kbps	Correct Peaks	False Alarms	Missed Peaks
32	119	3	1
56	126	1	0
96	120	0	1

Table 1. Performance of basic algorithm after mp3 compression

Next, we test the performance after low pass filtering. The original signal (at 22050) is low pass filtered to 4 kHz. As expected the technique is extremely robust to LPF because of the very low pass component in the envelope.

Next, we test the adaptive quantization scheme as described in subsection 3.5. The original signal (at 22.05 kHz) is resampled to higher and lower rates at powers of 2. To search for the optimal quantization step, the bandwidth of the envelope filter is changed by the inverse of the resampling ratio. For example if the signal is downsampled by a factor of 2 then the bandwidth of the envelope filter is doubled and so on. In addition, the number of necessary decomposition scales for the envelope is changed by the same factor. For example if 11 dyadic scales are used for the original signal, then only 10 dyadic scales are needed for the downsampled version (by 2) of the signal, and 12 dyadic scales are needed for the upsampled version (by 2) of the signal. Table 2 gives the results of applying our algorithm. The original signal is quantized with step Δ, then it is resampled by the factors in the first column. The entries of each row are the relative quantization error, as defined in (4), for each resampled signal using different quantization steps.

Resampling Factor	Decoding Quantization Step				
	$\Delta/4$	$\Delta/2$	Δ	2Δ	4Δ
0.25	0.018	0.264	0.223	n/a*	N/a
0.5	0.259	0.011	0.287	0.278	N/a
1	0.248	0.254	.010	0.285	0.285
2	0.228	0.249	0.254	0.010	0.286
4	0.239	0.232	0.249	0.254	0.010

* n/a : no peaks exist after thresholding

Table 2. Relative Quantization Errors of the Adaptive Quantization Step Algorithm

As noticed from the table the relative quantization error is very small when the correct step is tested. When this step is used in the decoding stage, the embedded data is extracted without any errors.

Next we test the same algorithm against slight time scale modifications between 0.9 and 1.1. For such modifications we did not change the bandwidth of the envelope filter nor the coarsest decomposition scale. Exhaustive search is conducted on the interval $[0.9\Delta, 1.1\Delta]$ with step 0.01Δ. The correct quantization step is always identified for scales in [0.92,1.08[. However for higher or lower modification scales, errors in identifying the quantization step occur frequently. The performance of the decoding algorithm using the *estimated* quantization step is shown in table 3.

Scaling Factor	Correct Peaks	False Alarms	Missed Peaks	Localization Errors
0.90	86	2	1	1
0.92	122	2	1	4
0.94	122	2	1	4
0.96	126	1	1	1
0.98	129	0	0	0
1.02	127	0	0	2
1.04	124	2	3	1
1.06	120	2	3	4
1.08	80	1	4	6
1.10	42	1	2	4

Table 3. Adaptive Quantization Performance against Slight Time Scale Modification

The localization errors for the peaks are primarily due to thresholding problems because of the effect of time scaling on the amplitude of the peak envelope. For modification scales lower than 0.92 or higher than 1.08, the bandwidth of the envelope filter as well as the coarsest decomposition scale should be changed accordingly.

5. CONCLUSION

In this paper, we introduced a novel algorithm for embedding data in audio signal by changing the interval lengths between salient points in the signal. We used the extrema of the wavelet coefficients of the envelope as our salient points. The proposed algorithm is flexible to work with any other choice of the salient points. We proposed a set of effective techniques to solve thresholding and quantization problems that can be used in other watermarking schemes as well. The proposed algorithm is robust to mp3 compression, low pass filtering, and it can be made robust to time scaling by using adaptive quantization steps.

The major drawback of the algorithm is the low embedding rate. Typically for acceptable error performance the embedded bit rate should be kept within 1-2 bits/second.

6. REFERENCES

1. M. Swanson, B. Zhu, and A. Tewfik, "*Current state of the art, challenges and future directions for audio watermarking* ", in Proceeding ICMCS 1999, pp. 19-24.
2. S. Mallat, " *A Wavelet Tour of Signal Processing*", Academic Press, Second Edition, 1999.
3. W. Verhelst and Marc Roelands, "*An Overlap-Add Technique based on Waveform Similarity (WSOLA) for High Quality Time-Scale Modification of Speech*", in Proceeding ICASSP93, Vol. II, pp. 554-557.
4. K. Hamdy, A. Tewfik, T. Chen, and S. Takagi, "Time-Scale Modification of Audio Signals with Combined Harmonic and Wavelet Representations", in Proceeding ICASSP97, pp. 439-442.