On Digital Watermarking for Audio Signals

Hristo Kostadinov and Nikolai L. Manev

Abstract We investigate the possibility of embedding watermarks robust against compression in musical audio files. The process of embedding and retrieving a watermark can be regarded as a binary communication channel. We investigate its statistic to give recommendation how to choose the embedding parameters and what error correcting codes to be used. The investigation covers the case of AAC and MP3 compression. The described method of embedding is based on a combination of key-dependable dither modulation and Haar wavelet transform. We analyze the whole process for its embedding capacitate and robustness. A practical method for choosing the embedding parameters is proposed.

1 Introduction

The goal of data hiding techniques is to provide a way to embed extra information within the original digital contents, without serious degradation of its quality. Data hiding techniques are formally divided into two classes: watermarking and steganography.

Digital watermarking is a data hiding technique of imperceptibly altering a digital object to embed a message about that object. The goal of watermarking is to prevent piracy or to prove the ownership, of course, with a trade-off between the embedding capacity, robustness and quality.

In many practical situations we do not need to provide strong security against removing or modification of the hidden message but it is very important to conceal its existence. *Steganography* is a technique of altering the object that assures

H. Kostadinov (✉)
Institute of Mathematics and Informatics, Bulgarian Academy of Sciences (IMI-BAS),
Acad. G. Bonchev Str., Bl. 8, 1113 Sofia, Bulgaria
e-mail: hristo@math.bas.bg

N. L. Manev
IMI-BAS, Acad. G. Bonchev Str., Bl. 8, 1113 Sofia, Bulgaria
e-mail: nlmanev@math.bas.bg

K. Georgiev et al. (eds.), *Advanced Computing in Industrial Mathematics*,
Studies in Computational Intelligence 793,
https://doi.org/10.1007/978-3-319-97277-0_19

this desired undetectable communication between partners, that is, no one but the intended recipient to be able to detect this altering.

Many audio file formats are used recently but compressed forms dominate in the way of distributing and storing audio content. **MP3** was the first commonly accepted format of compressed digital audio but modern ISO-MPEG codecs such as the **Advanced Audio Coding (AAC)** family began to dominate in recent years. Both are considered to be common audio signal manipulations. That is the reason to address both formats in our investigation.

The general structure of watermark embedding in audio files is given in Fig. 1. One can recognize three nested communication channels:

- outer binary channel - includes error control codes
- medium binary channel - dither modulation and wavelet transformation
- inner real number channel (in Fig. 2).

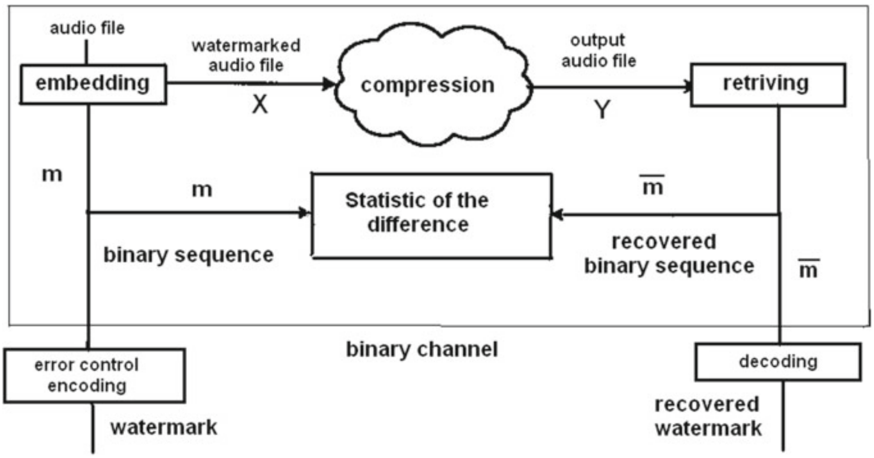
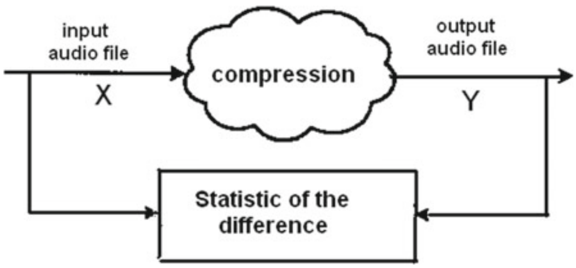


Fig. 1 The general structure of watermark embedding

Fig. 2 The inner (real number) channel



input: $\mathbf{X} = x_1, x_2, \ldots, x_i, \ldots$, values of the samples;
output: $\mathbf{Y} = y_1, y_2, \ldots, y_i, \ldots$, values after compression.

We aim after a comprehensive investigation of the embedding technique presented in Fig. 1 to propose a method of choosing its parameters so that to achieve acceptable probability of errors (according to characteristics of the watermark) without perceptible loss of quality of the audio file. The proper choice of error correcting code (ECC) is important both to achieve the aforesaid goals and to assure the maximum possible capacity for embedding. To make the choice we need to know the probability of errors, that is, the error characteristics of the medium channel presented by a rectangular in Fig. 1. Obviously, its error characteristics are function of parameters of embedding techniques and distortion due to compression. On the other hand the proper parameters of embedding techniques themselves are function of characteristic of compression. Hence, we need a good knowledge about characteristics of the inner channel presented in Fig. 2.

Our investigation consists of two parts:

- Collecting knowledge about statistical characteristics of the inner and medium channels presented in the above figures;
- Making conclusions about the most proper embedding parameters and how powerful error correcting codes to be used. (For a detail information about ECC we refer to [4].)

In this paper we describe a part of our study, namely, in the case when key-dependable dither modulation and Haar wavelet transformation are used in the embedding process. There are two reasons to study that case: first, the good properties of aforesaid transformations, and the second, the same embedding process is used in [2], thus, we have a base for comparison of the obtained results. Unfortunately, due to page limitations we have no possibility to present even a short review of numerous significant papers in the area and refer the reader to [2].

The remaining part of the paper is organized as follows. The necessary knowledge about wavelets and dither modulation are given in Sect. 2. In Sect. 3 we describe our investigations and present some results. Section 4 illustrates with examples some of our observations. The last section summarizes our conclusions and recommendations.

2 Preliminaries

2.1 Audio Compression Formats

There is much information about audio compression techniques in the Internet. Herein we give only several basic facts about MP3 and AAC that concern our study.

MP3 format. MP3 is commonly used name for MPEG-1 or MPEG-2 Audio Layer III lossy audio compression. It commonly achieves 75–95% reduction in size compared to CD quality digital audio. The CD digital audio represents music by 2 channels with 44,100 samples per second, each sample presented by 16 bits, that is, by $2 \times 16 \times 44,100 = 1\,411.2$ Kbits/s. MPEG-1 Audio Layer III standard supports

3 sampling frequencies of 32, 44.1 and 48 kHz and 14 bit-rates: 32, 40, 48, 56, 64, 80, 96, 112, 128, 160, 192, 224, 256 and 320 kbit/s. The most used bit-rate is 128 Kbits/s.

Although MP3 is still very popular amongst consumers most state-of-the-art media services use modern ISO-MPEG codecs such as the Advanced Audio Coding family which can deliver more features and a higher audio quality at much lower bit-rates compared to mp3. In a result the developer of MP3 in May 2017 terminated its licensing program and announces the “end” of MP3.

AAC format. Advanced Audio Coding is designed to be the successor of the MP3 format and it generally achieves better sound quality than MP3 at the same bit rate. AAC is the default audio format for the most modern devices and platforms. It supports sampling frequencies from 8 to 96 kHz AAC uses the modified discrete cosine transform together with the window length of 1024 (or 960 points). As a result the length of a file after decompression is a integer multiple of 1024. **In order to compare it with the original file the first and the last block of 1024 samples have to be removed. The original file has to be cut to the largest multiple of 1024 less than its length**.

We have made experiments only with sampling frequency 44.1 KHz and bit-rates 128 Kbits/s.

2.2 Haar Wavelet Transform

Haar wavelet transform is a linear transformation based on multiplication of pairs of coordinates of a target vector (block of samples in our case) by the orthogonal matrix

$$\mathbf{H}_2 = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}, \qquad \mathbf{H}_2^{-1} = \mathbf{H}_2.$$

The target vector with length 2^n is separated in consecutive pairs and each one is multiplied by $\mathbf{H}_2$. The first coordinate of the resultant pair is put at the corresponding place among the first 2^{n-1} positions of the output vector. The second coordinate is put at the same place in the last 2^{n-1} positions. The described transformation is referred to as ***Haar wavelet decomposition (HWD) level 1***.

Haar wavelet decomposition ***level 2*** means applying the transformation twice but the second time it is applied only to the first 2^{n-1} coordinates of the output vector of the level 1. And so on, the last transformation in the case of Haar wavelet decomposition ***level k*** is applied only to the first 2^{n-k+1} coordinates of the output of level $k-1$. The output has the structure

$$(*) \qquad [\, cA_k,\ cD_k,\ \ldots,\ cD_3,\ cD_2,\ cD_1\,],$$

where the length of $|cD_i| = 2^{n-i},\ i = 1, \ldots, k$, and $|cA_k| = 2^{n-k}$. In analogy to the Fourier transformation the positions in the output vector are often called *sub-bands*. Haar wavelet decomposition is realized by function *wavedec* in Matlab (see [3]).

For our investigation we need to know how an "error" is propagated after applying of inverse transformation: the so called *Haar wavelet reconstruction (HWR)* (function *waverec* in Matlab). More precisely, if a level k Haar reconstruction is applied to a vector with 1 only in one position and zeros anywhere else how many positions are nonzero in the output and what their magnitudes are. Knowing the described above structure (*) of Haar wavelet decomposition level k output and the fact that it preserves energy it is not difficult to conclude that if the input vector is $\mathbf{e}_i = [\underbrace{0, \ldots, 0}_{i-1}, 1, 0, \ldots, 0]$, then

- if $1 \leq i \leq 2^{n-k}$, i.e., i belongs to the subset cA_k then the output vector has 2^k consecutive nonzero coordinates (the i-th such group) of values $(1/\sqrt{2})^k$ and zeros anywhere else.
- if $i \in cD_j$, $j = k, k-1, \ldots, 1$, then 2^j consecutive coordinates are nonzero, first half of them are equal to $(1/\sqrt{2})^j$, the second half are equal to $-(1/\sqrt{2})^j$.

2.3 Quantization-Based Embedding Algorithms and Dither Modulation

Quantization-based embedding algorithms are very popular because they are easy to be implemented, assure computational flexibility, high embedding rate and amenability to theoretical analysis. In [1] Chen and Wornell introduced a new quantization-based data hiding algorithm called *Quantization Indexed Modulation (QIM)*. Since then, several variations have been developed. We use a variant of QIM known as *Dither Modulation (DM)* introduced in the same paper [1]. DM's main characteristics are low-complexity by using scalar and uniform quantizers and the possibility of incorporating a private key in the embedding process.

The parameters of the used in our study DM are

- ***Quantizer*** $q(.)$ which is characterized by
  - set Q of points at distance Δ, which is called ***step*** of the quantizer;
  - $q : x \to q(x)$, where $q(x)$ is the closest to x point of Q.
- Dither (pseudo-random key-dependable) parameters:
  - $v(0)$ uniformly distributed in $[-\Delta/2, \Delta/2]$ pseudo-random sequence;
  - $v(1) = \begin{cases} v(0) + \Delta/2, & \text{if } v(0) < 0 \\ v(0) - \Delta/2, & \text{if } v(0) \geq 0 \end{cases}$.

Embedding Algorithm
Data:

- $\mathbf{x} = x_1, x_2, \ldots, x_i, \ldots$ signal points;
- $\mathbf{m} = m_1, m_2, \ldots, m_i, \ldots$ binary sequence (watermark);

- $\hat{\mathbf{x}} = \hat{x}_1, \hat{x}_2, \ldots, \hat{x}_i, \ldots$ output signal points.

Embedding: $\hat{x}_i = q(x_i + v(m_i)) - v(m_i)$.

Retrieving Algorithm
Data:

- $\mathbf{y} = y_1, y_2, \ldots, y_i, \ldots,$ received signal points, which are the noisy $\mathbf{x}$ due to attack or other manipulations
- $\hat{\mathbf{m}} = \hat{m}_1, \hat{m}_2, \ldots, \hat{m}_i, \ldots$ - retrieved watermark.

Retrieving:

$$\hat{m}_i = b \in \{0, 1\} \text{ such that } y_i + v(b) - q(y_i + v(b)) \text{ is minimal.}$$

Remark. To perform the retrieving step it is not necessary to keep the sequences $v(0)$ and $v(1)$, but only the length of embedded sequence and the initial state (the "key" of embedding) of the pseudo-random generator.

3 Description of Investigations and Experiments

In accordance with our intentions given in the Introduction our study past through the following main research stages:

1. Choosing 30 musical files each of duration ≈ 4 minutes from five genres: classical, instrumental, pop, hard rock, soft rock. From our point of view the chosen files represent well the variety in each genre.
2. Determining statistical characteristics of the inner channel, that is, the distortion effect of compression.
3. Making conclusion about the value of Δ, the level of HWD, and in which subbands to embed.
4. Determining statistical characteristics of the middle (binary) channel.
5. Taking a decision about parameters of interleaving and error correcting codes.

3.1 Stage 1 and Stage 2

All files are .wav stereo format with sampling rate 44.1 KHz and resolution 16 bits. They are chosen randomly aiming only the files in each genre to differ as much as possible. To process file we use Matlab. Audio files are presented by $N \times 2$ matrices with entries in $[-1, 1]$, where the columns correspond to the left and right stereo channels. In Table 1 are given the mean *mean(abs)*, standard deviation *std(abs)*, and maximum *max(abs)* of absolute values of the elements of corresponding matrices

Table 1 Characteristics of different musical genres

Musical genres	Mean(abs)		Std(abs)		Max(abs)	
	1	2	1	2	1	2
Classic1	0.0626	0.0535	0.0668	0.0557	0.6184	0.5844
Classic2	0.0564	0.0535	0.0546	0.0524	0.6261	0.7410
Instrumental1	0.1004	0.1101	0.1019	0.1074	0.9199	0.9214
Instrumental2	0.1644	0.1519	0.1425	0.1330	0.9989	0.9989
HardRock1	0.0888	0.0890	0.0794	0.0801	0.9756	0.9938
HardRock2	0.1291	0.1194	0.1187	0.1107	0.9039	0.9039
SoftRock1	0.1319	0.1361	0.1266	0.1238	0.9204	0.9206
SoftRock2	0.1610	0.1666	0.1412	0.1458	0.9888	0.9888
Pop1	0.1360	0.1361	0.1295	0.1287	0.9989	0.9989
Pop2	0.1730	0.1782	0.1583	0.1620	0.9989	0.9989

for two files from each genre. The representative files of the genres are chosen to be maximum different from one another.

One can observe the differences in each genre as well as between genres. We would like to attract attention of the reader to the files with classical music. The larger difference with the files of the other genres can be also observed in next results given below.

The next step in our study is to collect some information about the distortion effect of compression, that is, to study the inner channel given in Fig. 2. For that purpose we compress and then decompress each files:

$$\text{file.wav} \Longrightarrow \text{file.aac} \Longrightarrow \text{file-modified.wav}$$

Then we compute and analyze the difference between matrices corresponding to file.wav and file-modified.wav. The obtained results are illustrated with Table 2. The file used for making Table 2 are the same as ones used for Table 1 and it demonstrates the following two facts observed during our experiments:

- The mean M and the standard deviation σ are approximately equal;
- Mutual relations among characteristics of a file, among files and genres in both tables are very similar. Hence, by determining only simple characteristics of the musical file we can make conclusions about distortions of compression (both for AAC and MP3).

One can achieve the aforesaid conclusions by carefully analyzing the methods used for AAC and MP3 compressions but such considerations are out of the range of this paper.

Table 2 The absolute value of differences for 128 Kbits/s AAC; $\alpha = M + \sigma$

Musical genre	Mean M	Std σ	Max	% diff > α	% diff > 2α
Classic1	0.0032	0.0034	0.1074	11.50	1.58
Classic2	0.0022	0.0024	0.0910	12.32	2.00
Instrumental1	0.0168	0.0179	0.3976	13.44	2.25
Instrumental2	0.0195	0.0215	0.5318	12.96	2.12
HardRock1	0.0120	0.0120	0.3629	13.62	1.40
HardRock2	0.0117	0.0117	0.3200	12.78	1.20
SoftRock1	0.0201	0.0198	0.3074	14.01	1.65
SoftRock2	0.0247	0.0245	0.4341	14.64	1.88
Pop1	0.0203	0.0216	0.6197	13.96	2.25
Pop2	0.0250	0.0297	1.1321	11.39	2.20

3.2 Stage 3

Process of embedding carried out in the middle channel consists of the following steps

Step E1. The sequence of samples of both stereo channels (or only of one) is divided into blocks of length $1024 = 2^{10}$ (or 512). If the length of the audio file is not multiple of 1024 we cut it or add zeros at the end.

Step E2. Applying Haar wavelet decomposition level k to each block of 1024 samples. Usually $k = 5$ or 6.

Step E3. Embedding 2^{10-k} bits from the watermark $\mathbf{m}$ into each block using dither modulation method with step Δ as it is described in Sect. 2.3. The embedding is into sub-bands cA_k or cD_k.

Step E4. Applying Haar wavelet reconstruction level k to each modified block of 1024 samples.

Step E5. Reassembling blocks as a file and compressing it into .aac or .mp3 file with bit rate 128 Kbits/s.

The parameters of the embedding process are step Δ and level k. After Stage 2 we have sufficient information to decide how to choose Δ. Depending on the watermark the acceptable probability is less than 10^{-3}, in many cases the upper bound is 10^{-4} or less. The values of the changes in Step 3 belong to $[-\Delta/2, \Delta/2]$ but after Step 4 they result in modifications with values in

$$\left[-\frac{\Delta}{2(\sqrt{2})^k}, \frac{\Delta}{2(\sqrt{2})^k}\right].$$

If the above interval is very small the compression will distort modifications. In such a case the bit error probability (BER) for medium channel is about 1/2 and no error correcting codes can help to recover the watermark. On the other hand if the interval is large it worsens the quality of the musical file and makes embedding too perceptible. Hence the choice of Δ has to assure the balance between two opposing goals: good quality of the modified file and lower BER.

The last two columns of Table 2 gives an idea how to choose Δ. They show for how many percents of the samples the distortion values are greater than α and 2α, respectively. In order the embedded bit to be retrieved the altering of the sample value has to be greater than the distortion value. Therefore to achieve BER about $10^{-2} \div 10^{-1}$ we need

$$\alpha \leq \frac{\Delta}{2(\sqrt{2})^k} \leq 2\alpha \qquad \Longrightarrow \qquad \Delta \in [2\alpha(\sqrt{2})^k,\ 4\alpha(\sqrt{2})^k]$$

For "noisy" musical files and genres Δ can be chosen about the right end of the interval or even greater in order to decrease BER. On the other hand for some files, mainly classical music and some instrumental, Δ must be close or even less than $2\alpha(\sqrt{2})^k$ to assure acceptable quality. But in this case very powerful error correcting codes has to be used which results in lower capacity of embedding. Note that with exception of classical music genre $\alpha \approx 0.035 \div 0.045$.

The above observations confirm the conclusion about classical music in [2].

Where to embed?

Let us first note that embedding into $cD_j,\ j < k$, sub-bands is equivalent to use of HWD level j. Hence the use of HWD level k means that we embed into cA_k or cD_k sub-bands. There is an widespread understanding that modifications to low-frequencies produce audible distortion. In [2] the authors recommend to embed in several sub-bands with highest correlation between amount of noise and average energy of sub-band. It is logically argued but leads to "buzz" effect. The embedding only into several sub-bands modifies only several samples in each block that introduces a periodical signal in the modified file heard as a buzz.

On the contrary, if we embed into all sub-bands of cA_k or cD_k all positions of the time-domain block are altered. The embedded signal correlates with the pseudo random sequences, it is very similar to white Gaussian noise, and it is perceived as a noise of the speaker or computer ventilator by the listener if the noise is enough strong.

Therefore we **recommend embedding into all sub-bands of cA_k or cD_k**.

Also we recommend the Haar wavelet transformation to be with level k equals at least 5, but there is no noticeable improvement for larger k.

At the end of this stage we carry out the steps E4 and E5 and produce *file-modified.aac*.

3.3 Stage 4 and Stage 5

To compute BER of the middle channel for the studied file we convert *file-modified.aac* to *file-aftercompression.wav*, retrieve watermark (with errors) following the procedure given below, and compare it with the original.

Table 3 The probability of error (BER)

Musical genre	E(Δ)	Δ = 0.15	Δ = 0.7	Δ = 0.9	Δ = 1.1
Classic1	0.1584	0.0522	0.0000	0.0000	0.0000
Classic2	0.0996	0.0235	0.0000	0.0000	0.0000
Instrumental1	0.7580	0.4232	0.0386	0.0170	0.0083
Instrumental2	0.9322	≈ 0.5	0.0883	0.0430	0.0210
HardRock1	0.5453	≈ 0.5	0.0105	0.0035	0.0013
HardRock2	0.5295	≈ 0.5	0.0105	0.0034	0.0012
SoftRock1	0.9028	≈ 0.5	0.0685	0.0310	0.0150
SoftRock2	1.0881	≈ 0.5	0.0892	0.0441	0.0220
Pop1	0.9481	≈ 0.5	0.0412	0.0182	0.0092
Pop2	1.2488	≈ 0.5	0.1034	0.0646	0.0432

Retrieving the watermark:

Step D1. Delete the first 1024 samples for both stereo channels of *file-aftercompression.wav* and then carry out step E1.

Step D2. Do step E2.

Step D3. For each block retrieve 2^{10-k} bits from the positions cA_k (or cD_k if they are target for embedding) using retrieving algorithm described in Sect. 2.3.

Step D4. Reassemble retrieved pieces of bits into a binary sequence in order to compare with the original watermark.

Table 3 presents the probability of error per bit for the same audio files used for Tables 1 and 2. The results are obtained under the following conditions

- HWD level 5 over blocks of 1024 bits;
- Embedding into $1 \div 32$ sub-bands, i.e., into cA_5;
- 128 Kbits/s ACC compression;
- $\mathrm{E}(\Delta) = 4\alpha(\sqrt{2})^5$ is given for comparison with the used values of Δ (given in the top of the columns).

The red color in the table means that the quality of the modified file is bad and unacceptable, the yellow color - medium quality, and the green color - good and acceptable quality. The table as well as experiments with the other files confirm our conclusion in Sect. 3.2 that choosing Δ with values close to E(Δ) we can achieve BER of order 10^{-2}. With not very complex error correcting codes the BER can be decreased to 10^{-4} or less. In some cases there is no need of codes: 10^{-2} turn out to be sufficient the watermark to fulfill its function.

The capacity

If we embed in both channels of the target audio file by using HWD level k and embedding in cA_k or cD_k the amount of embedded bits is

$$(2 \times \text{sampling rate} \times 2^{10-k})/2^{10} = \text{sampling rate}/2^{k-1} \quad \text{bits per second.}$$

If we use ECC with rate R the capacity is

$$\frac{\text{sampling rate}}{2^{k-1}} \times R \quad \textbf{information bits per second}.$$

Even when $R = 1/5$ with $k = 5$ we achieve 551 information bits/s, or 132 300 information bits in the whole file. This amount is equivalent of a 128×128 pixels gray-scale picture.

The errors in the middle channel tend to group in bursts. For example, $\approx$700 zeros and then an interval of length $\approx 200 \div 400$ with many ones. (Demonstrates behavior of Markov chain.) Hence the use of interleaving and/or burst error correcting codes (e.g., Reed-Solomon codes over a large finite field) is a good decision.

In [2] it is proposed to use a combination of LDPC codes with very long repetition codes. Such an approach decreases the capacity and requires a modification of the algorithm for retrieving watermark (Sect. 2.3). Although there are no limitation in time and computing resources for recovering the watermark it is not necessary to involve very complex ECC design. Its choice is also function of the characteristic of the watermark. In the examples given in the next section the retrieved watermark can be doubtless used for proving ownership although very week codes are used.

4 Examples

In this section we give two examples of embedding a gray-scale picture into a musical audio file. In the both examples we use as a target file "Classical 2" - a piano performance having many silent passages that makes modifications more perceivable. This file is a representative of the most inappropriate for embedding audio files. Although the used error correcting codes are very week the qualities of the both retrieved watermark and marked audio file are very good. These examples demonstrate the importance of proper choice of Δ and embedding place. In both examples $\Delta = 0.08$, which is less than but close to $E(\Delta)$. Example 2 illustrates the influance of interleaving.

Example 1 The embedded watermark is the 102×102 pixels photo of Lenna shown in Fig. 2a. After transformation into a $\{0, 1\}$-sequence of length 83,232, it is repeated 7 times to construct the embedded sequence of 582,624 bits. The repetition is equivalent of use of $[7, 1, 1]$ code with some form of interleaving as far as each bit is repeated after 83,232 bits. The embedding is in $1 \div 32$ sub-bands. The decoding algorithm consists of choosing the symbol (0 or 1) dominated in the codeword. The probability achieved after decoding is **BER = 0.0062** and the retrieved image is shown in Fig. 3b. The quality of the modified audio file is very good - the embedded data is imperceivable.

If the embedding is in $33 \div 64$ sub-bands, i.e. in cD_5, we achieve **BER = 0.0004** (see Fig. 3c), but the quality of the modified audio file is slightly worse. Nonetheless it remains acceptable good.

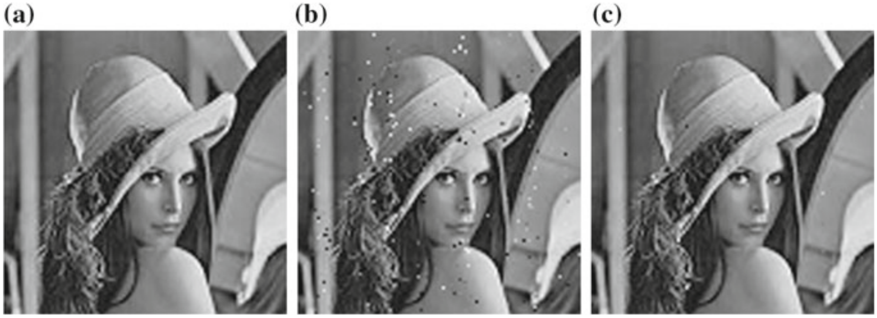


Fig. 3 **a** The original; **b** embedding in 1÷32 sub-bands; **c** in 33÷64 sub-bands

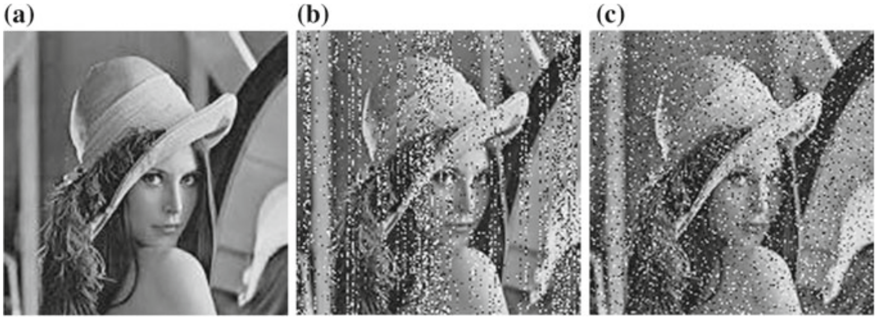


Fig. 4 **a** Original watermark; **b** without interleaving; **c** 177 × 3385 interleaving

Example 2 In this example the target audio file is also Classical2.wav and the watermark is Lenna's photo but with a larger size of 158 × 158 pixels. We use the [15, 5, 7] BCH code with zeros $\beta, \beta^2, \beta^3, \beta^4, \beta^5, \beta^6$, where β is the primitive element of $GF(16)$ with minimal polynomial $x^4 + x + 1$. The generating polynomial of the code is $g(x) = x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1$ and the code can correct up to 3 errors in codeword. Its larger coding rate of 1/3 enables us to increase the amount of embedded data more than two times in comparison with Example 1 but the price we pay is a larger BER and worse quality of the retrieved image.

We performed embedding in two variants: without and with 177 × 3385 interleaving. The values of BER are relatively large 0.1105 and 0.0787, respectively. The original and the retrieved watermarks are shown in Fig. 4.

5 Conclusion

The results of our investigations can be summarized as follows:

- Embedding in all sub-bands of cA_k (or cD_k) improves audio quality.

- The proposed approach to choosing Δ is more practical than the using SNR. The choice of Δ close to $E(\Delta)$ leads to BER in $[10^{-2}, 3.10^{-2}]$. These values are typical for telephone channels and there are many error correcting codes with low complexity designed for such channels.
- The better audio quality requires smaller Δ and more powerful codes to be used. Errors in middle channel demonstrate inclination towards grouping. (More exactly, the probabilistic model of the channel is Markov chain.) Thus, the use of interleaving and burst error correcting codes is recommended.
- The capacity in the case of HWD level k and ECC with rate R is

$$\frac{\text{sampling frequency}}{2^{k-1}} \times R \quad \textbf{information bits per second.}$$

Acknowledgements This work was partially supported by the National Science Fund of Bulgaria under Grant DFNI-I02/8.

References

1. Chen, B., Wornell, G.W.: Quantization index modulation: a class of provably good methods for digital watermarking and information embedding. IEEE Trans. on Inf. Theory **47**(4), 1423–1443 (2001)
2. Martinez-Noriega, R., Nakano, M., Kurkoski, B., Yamaguchi, K.: High payload audio watermarking: toward channel characterization of MP3 compression. J. Inf. Hiding Multimedia Signal Proc **2**(2), 91–107 (2011)
3. MathWorks, https://www.mathworks.com/help/wavelet/index.html
4. Pless, V.S., Huffman, W.C. (eds.): Handbook of Coding Theory. Elsevier, Amsterdam (1998)