StirMark Benchmark: Audio watermarking attacks

Martin Steinebach[1], Fabien A. P. Petitcolas[2], Frédéric Raynal[4], Jana Dittmann[1],
Caroline Fontaine[3], Christian Seibel[1], Nazim Fatès[2], Lucilla Croce Ferri[1]
[1] GMD-IPSI, {martin.steinebach, dittmann, seibel, ferri}@darmstadt.gmd.de
[2] Microsoft Research, fabienpe@microsoft.com, n_fates@scientist.com
[3] USTL-LIFL, caroline.fontaine@lifl.fr
[4] INRIA Rocquencourt, frederic.raynal@inria.fr

Abstract

In this paper we will briefly present the architecture of a public automated evaluation service we are developing for still images, sound and video.

We will also detail new tests that will be included in this platform. The set of tests is related to audio data and addresses the usual equalisation and normalisation but also time stretching, pitch shifting and specially designed audio attack algorithms. These attacks are discussed and results on watermark attacks and perceived quality after applying the attacks are provided.

1. Need for evaluation

The growing number of attacks against watermarking systems (e.g., [1, 2, 3]) has shown that far more research is required to improve the quality of existing watermarking methods so that, for instance, the coming JPEG 2000 (and new multimedia standards) can be more widely used within electronic commerce applications.

With a well-defined benchmark, researchers and watermarking software manufacturers would just need to provide a table of results, which would give a good and reliable summary of the performances of the proposed scheme. So end users can check whether their basic requirements are satisfied. Researchers can compare different algorithms and see how a method can be improved or whether a newly added feature actually improves the reliability of the whole method. As far as the industry is concerned, risks can be properly associated with the use of a particular solution by knowing which level of reliability each contender can achieve.

2. Evaluation tool

As a first step towards a widely accepted way to evaluate watermarking schemes we started to implement an automated benchmark server. The idea is to allow users to send a binary library of their scheme to the server which in turns runs a series of tests on this library and keeps the results in a database accessible to the scheme owner or to all ‘watermarkers’ through the Web.

2.1. Methodology – Need for third party

To gain trust in the reliability of a watermarking scheme, its qualities must be rated. This can be done by:

- trusting the provider of the scheme and his quality assurance (or claims);
- testing the scheme sufficiently oneself;
- having the scheme evaluated by a trusted third party.

Only the third option provides an objective solution to the problem but the general acceptance of the evaluation methodology implies that the evaluation itself is as transparent as possible. This was the aim of StirMark and this remains the aim of the project to build a next generation of StirMark Benchmark. This is why the source code and methodology must be public so one can reproduce the results easily.

2.2. Criteria

- *Simplicity*: In order to be widely accepted this service has a simple interface with existing watermarking libraries (only three functions must be provided by the user).
- *Customisation*: For each type of watermarking scheme, we want to use a different evaluation profile without having to recompile the application tool.
- *Modularity and choice of tests*: Watermarking algorithms are often used in larger system designed to achieve certain goals (e.g., prevention of illegal copying, trading of images).
- *Perceptibility* characterises the amount of distortion introduced by the watermarking scheme itself. The problem here is very similar to the evaluation of compression algorithms. We allow the addition and use of different quality metrics; the simplest and most widely used one being the P.S.N.R.
- The *capacity* of a scheme is the amount of information one can hide. In most applications it will be a fixed constraint of the system so robustness tests will be done with a random payload of given size.
- The *robustness* can be assessed by measuring the detection probability of the mark and the bit error rate for a set of criteria that are relevant to the appli-

cation, which is considered. Part of these evaluation profiles can be defined using a finite and precise set of robustness criteria (e.g., S.D.M.I., IFPI or E.B.U. requirements) and one just needs to check them. Many of the possible tests can be deduced from previous works presenting attacks on watermarking schemes.

- Finally, *speed* is very dependent on the type of implementation: software or hardware. Here we are only concerned with software implementation and our test just computes an average of the time required on a particular given platform to watermark and image depending on its size.

One of the difficulties to set up a base of tests relates to the diversity of the algorithms and the audio-visual signals. Certain algorithms are created relative with a particular type of image when others want to be more general. Our base of audio-visual signals must be very wide. Thus, each algorithm will be tested on a randomly generated subset of signals. In a second time, it will be possible for a user to specify one or more sets of signals, either with the sight of the results obtained at the time of the first series of tests, or because the algorithm is dedicated to precise signals in order to refine the results obtained.

3. Architecture

The evaluation service only requires three functions to be exported from the watermarking library supplied by the user. The first one, *GetSchemeInfo* provides information about the marking scheme such as its type and purpose, its operational environment, its author, version, release date, etc. The two other functions are the complementary *Embed* and *Extract* functions.

We tried to capture all possible cases and ended up with a solution where several parameters are provided but not all of them are mandatory. They include the original audio-visual signal, the watermarked signal, the embedding key, the 'strength' of the embedding, the payload, the maximum distortion tolerated and the certainty of extraction. This very simple technique allows interoperability with schemes of various types and only requires having a common unique source code header to maintain.

The *strength* parameter – which can be used to evaluate the compromise between imperceptibility, capacity and robustness – is assumed to have the following properties:

- strength is a floating point number between 0 and 100;
- the higher the value of strength, the lower the quality of the output image and the higher (hopefully) the 'robustness';
- strength = 0 corresponds to no watermarking (P.S.N.R. → ∞);
- strength = 100 should correspond to a watermarked picture with P.S.N.R. around 20 dB;
- the distribution of strength should be 'harmonious'.

The project is being written using the C++ language to take full advantage of the inheritance and polymorphism features of an object-oriented language. Support for various watermarking application is achieved by the use of an initialisation file per evaluation profile in which each test has its own parameters stored.

4. Audio attacks

Many of the possible tests can be deduced from previous works presenting attacks on watermarking schemes. In this section we present some attacks, which are dedicated to audio as past research has mainly focused on still images. We show the impact and the audibility of the attacks depending on various audio materials. The influence of the attacks on different audio watermarking schemes is also discussed to prove their importance.

4.1. Audio attack classification

Any manipulation of an audio file can result in an attack on the embedded watermarks. Depending on the way the audio information will be used, some attacks are more likely than others. Based on this, we set up postproduction models for different environments as shown in Figure 1. An example for such a model could be the preparation of audio material to be transmitted at a radio station: the material will be normalised and compressed to fit the loudness level of the transmission. Equalisation will be used to optimise the perceived quality. A denoiser or dehisser reduces unwanted parts of the audio information and filters will cut off any frequency which can not be transmitted.

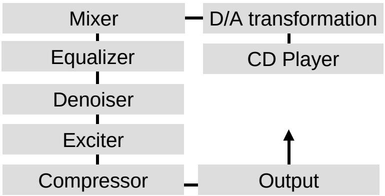

Figure 1 – Attacks on a CD signal played over a set of audio modules, for example for radio transmission.

If a watermark is used to detect radio transmission of commercials, it has to be robust against all the attacks described above, or the detection will not be possible as it is destroyed.

Another example is the Internet: if a company wants to embed watermarks as copyright protection, the watermark has to be robust against all operations usually applied to the material. In this case the main attack will be lossy compression like mp3, sometimes at high compression rates.

To evaluate weaknesses of watermarking algorithms, we also build groups of attacks based on the way manipulation works. Based on the attack models we have identified the following groups of attacks:

- Dynamics – These change the loudness profile of an audio file. Increasing or decreasing are the most basic attacks. Limiting, expansion and compression are more complicated, as they consist of non-linear changes depending on the material. There are even frequency dependent compression algorithms which only affect a part of the frequency range.
- Filter – Filters cut off or increase a selected part of the spectrum. The most basic filters are high-pass and low-pass filters but equalizers can also be seen as filters, they are usually used to increase or decrease certain parts of a spectrum.
- Ambience – This group consists of audio effects simulating the presence of a room. The most common effects are reverb and delay, which offer a lot of parameters depending on the quality of the effect.
- Conversion – Audio material is often subject to format changes. Mono data has to be doubled to be used in stereo environments. Sampling frequencies differ from 32 kHz to 48 kHz and now even 96 kHz. Sample size changes from 16 to 24 bit and back.
- Lossy compression – Audio compression algorithms based on psycho acoustic effects are used to reduce the amount of audio data by factor 10 or better.
- Noise – Noise can be the result of most of the attacks described above. Most hardware components in an audio chain also induce noise into the signal. A very common attack also is to try to add noise to destroy the watermark.
- Modulation – Modulation effects like vibrato, chorus, amplitude modulation or flanging are usually not used in postproduction. As most audio processing software includes such effects, they can be used as attacks to watermarks.
- Time stretch and pitch shift – These either change the length of an audio event without changing its pitch or change the pitch without changing the length. They are used for fine tuning or fitting audio parts into time windows.
- Sample permutations – This group consists of algorithms not used for audio manipulation in usual environments. Theses are specialised ways to attack watermarks embedded in audio files. Examples are sample permutation, dropping samples and similar approaches.

A selection of these attack types is used as attack algorithms for the StirMark Benchmark environment. These selected attacks are introduced in Section 4.3.

4.2. Audio attack tests

The two major goals of our tests were to find attacks that can destroy watermarks embedded in audio data and to find out if the perceived quality of the audio files is changed by the attacks. Therefore testing took place in three phases:

1. *Attack identification*: We used audio editing software[1] effects and own attack algorithms to manipulate audio data. As all attacks provide parameters to adjust the strength of the manipulation, we tried to find the strongest inaudible parameter settings based on our own perception. In StirMark Benchmark, the parameters will be changed by algorithms to find the setting where the watermark is destroyed.
2. *Watermark attacks*: The identified attacks from phase 1 were used to attack different audio watermarking algorithms. All audio examples were marked and then attacked.
3. *Subjective tests*: The most promising attacks from phase 1 and 2 were tested by a number of test subjects using a double blind triple stimulus test.

To evaluate our attacks, we have chosen six sound files with variant characteristics:

- ‘Serenade des Abschieds’: spoken text, a poem recorded in high quality;
- ‘Menuetto’: classical music example by Mozart with violins, rather quiet;
- ‘I’m ready’: Pop music by Brian Addams, life recording with audience noise;
- ‘Time in’, jazz music by Joe Pass, guitar solo;
- ‘Endorphinmachine’ by Prince, very loud recording of rock-pop music;
- ‘City’: Urban ambience recoding, sounds of a truck starting.

4.3. Applied attacks

The selection of attacks cannot be exhaustive as there are many different audio effect algorithms. We have chosen a number of different attacks, trying to provide a wide range of different attack classes. There are musical effects and algorithmic attacks.

4.3.1. Dynamics

- *Compressor*: A compressor is used to decrease the range of signal strengths in audio signals, thereby making it possible to receive a louder overall signal as peaks are reduced to a limit and do not cause distortions. We have used the following settings: Attack time 1 ms, release time 500 ms, output gain 0 dB, threshold –50 dB and ratio 1:1.1. This is a very fast and almost inaudible setting changing all signals louder then –50 dB by a small amount.

[1] Sonic Foundery Sound Forge 4.5 and CoolEdit 2000 by Syntrillium

- *Denoiser*: Denoisers are used to remove noise from audio signals. A parameter is used to set the loudness of signals interpreted as noise. We have used setting of –80 dB and –60 dB. This is similar to a gate. There are more sophisticated denoisers using declickers and other tools to provide better quality.

4.3.2. *Filters*

- *High pass*: A filter removing all frequencies lower than a chosen threshold, 50 Hz in our case.
- *Low pass*: A filter removing all frequencies higher than a chosen threshold, 15 kHz in our case.
- *Equalizer*: An equalizer is used to reduce frequency channels by 48 dB. The used bandwidth was frequency divided by 10,000. Three versions of this attack have been tested using a range from 31 Hz to 16 kHz: 10 frequencies with the distance of 1 octave, 20 frequencies with the distance of 1/2 octave and 30 frequencies with the distance of 1/3 octave.
- *L/R-Splitting:* An equalizer effect is used to increase the perceived stereo image. Working on a stereo channel, frequency shares are reduced in one channel and increased in the other. The spectrum is divided into 20 frequency bands, and every second frequency band is reduced by 6 dB on the left audio channel and increased on the right audio channel. To hide the resulting volume change, the overall volume of the channel has to be normalized.

4.3.3. *Ambience*

- *Delay*: A delayed copy of the original is added to it. This is used to simulate wide spaces. We use a delay time of 400 ms, the volume of the delayed signal is 10 % of the original.
- *Reverb*: This effect is used to simulate rooms or buildings. It is similar to delay, but uses shorter delay time and reflections.

4.3.4. *Conversion*

- *Resampling*: The sampling frequency of a signal is changed. Typical applications down-sample from 48 kHz to 44.1 kHz in CD production. We changed a 44.1 kHz signal to 29.4 kHz. Thereby the highest possible frequency in the signal is reduced, the result is similar to a low pass filter.
- *Inversion*: This is an inaudible attack changing the sign of the samples. It was used for completeness as the watermarking algorithms we tested had been defeated by this in previous tests.

4.3.5. *Noise*

- *Random noise*: Addition of random numbers to the samples, constrained by a parameter giving the relative amount of the number compared with the original signal. Up the 0.91 % of the original sample value could be added as noise without degrading the perceived quality.

4.3.6. *Modulation*

- *Chorus*: A modulated echo is added to the signal with various delay time, modulation strength and number of voices. We have used the following settings: 5 voices, maximum delay 30 ms, delay rate 1.2 Hz, feedback 10 %, voice spread 60 ms, vibrato depth 5 dB, vibrato rate 2 Hz, dry out (unchanged signal) 100 % and wet out (effect signal) 5%.
- *Flanger*: Flanging is usually created by mixing a signal with a slightly delayed copy of itself, where the length of the delay is constantly changing.
- *Enhance*: An enhancer is used to increase the amount of high frequencies in a signal, thereby increasing its perceived brilliance. This effect is also known as ‘exciter’. We used Sound Forge to apply the effect with a medium setting – detailed information about the parameters is not provided by the program.

4.3.7. *Time stretch and pitch shift*

- *Pitch shifter*: This effect is used to change the base frequency without changing the speed of the audio signal. This is one of the most sophisticated algorithms in audio editing today and there are many different specialized algorithms providing varying quality depending on the characteristics of the original signal. We use Sound Forge to increase the pitch by 5 cents, this is a 480th of an octave.
- *Time stretch*: A similar effect to the pitch shifter. It is used to increase or decrease the duration of an audio signal without changing its pitch. We have used Sound Forge to produce signals with a length of 98 % of the original duration.

4.3.8. *Sample permutations*

- *Zero-cross-inserts*: We search for samples with the value 0 and add 20 zeros at this position, creating a small pause in the signal. The minimal distance between pauses is one second.
- *Copy samples*: Samples are randomly chosen and repeated in the signal, thereby increasing its duration. Our tests used 20 signal repetitions in 0.5 seconds.
- *Flip samples*: The positions of randomly chosen samples are exchanged. Again we used 20 samples in 0.5 seconds.
- *Cut samples*: A sequence of randomly chosen samples is deleted from the signal. To make this attack inaudible, we had to use a maximum sequence length of 50 and a maximum value difference between start and end sample. We deleted 20 sequences in 0.5 seconds.

4.4. Watermark attack tests

We tested our attack algorithms with four audio watermarking algorithms, three of them provided by companies[2] and one of them was designed by GMD. The algorithms will be called A to D, where D is GMD's algorithm. It is the prototypic implementation of a PCM watermarking algorithm still under development. Two of the companies provide algorithms with parameters to regulate the strength of the embedded mark. Here we used a standard and a strong mark. D was tested with medium embedding strength and medium watermark audibility.

Figure 2 shows the test results for D. The percentage of the destroyed watermarks is given. Delay and low pass are the weakest attacks while resampling and zero-cross-inserts change are remove all watermarks. Resampling even destroys the marks completely.

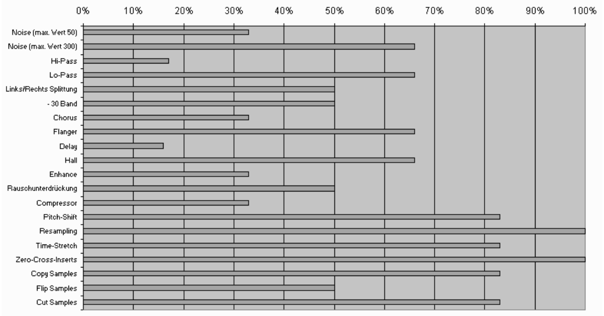

Figure 2 – Attack results for D - percentage of destroyed watermarks.

Table 1 provides an overview of the attack results. Every algorithm has its strengths and weaknesses. C is robust against many attacks but is completely destroyed by resampling and inversion. A and B are robust against resampling, but sample cut and time stretch removes all marks. D is only removed completely by resampling, other attacks change some bits. Error correction codes can be implemented to provide protection against this. Pitch shifting was the most effective attack removing all marks in A to C and changing the mark in five examples in D.

The table does not provide information about which examples were affected how often. B had problems with 'Serenade', the mark in it was more often removed than in any other example. A+ had most problems with 'Menuetto'. D did not show a strength or weakness with one example. For noise and equalizer attacks, only the results of the strongest setting are listed.

4.5. Subjective listening tests

We used samples form the six sound examples with a length of 4 to 6 seconds. Listening equipment was a professional audio recording sound card and studio monitor speakers. The listeners' distance to the two speakers was exactly the same and they were placed on the same height as the listeners' head. This environment provides a more transparent sound than most home stereo sets and a far better quality than common PC-based sound sets.

The test had a duration of about 90 minutes, breaks were allowed to keep up listener concentration. We used a triple stimulus hidden reference double blind test described in [4] where listeners are given three signals. The first is known to be the original, the following two are either original or attacked original. These two signals are given marks from 5 (no changes perceived) to 1 (very low quality). The test was done with ten persons, five of them musicians or experienced listeners.

Attack	A	A+	B	B+	C	D
Compressor						2
Denoiser					**3**	3
High pass		**1**				1
Low pass		**1**			1	6
Equalizer	**6**	**6**	6	4	1	3
L/R split	**6**	**6**	6			2
Delay	**6**	**2**	**1**	**1**		1
Reverb	**6**	**6**	**1**	**1**		3
Resampling					**6**	**6**
Inversion					6	
Noise	**1**	**1**	6	6	**3**	**1**
Flanger	**6**	**6**				3
Chorus	**6**	**2**	6			2
Enhancer						2
Pitch	**6**	**6**	**6**	**6**	**6**	5
Time	**6**	**6**	**6**	**6**		5
Zero-cross	**6**	**6**	**6**	**4**		6
Copy	**6**	**6**	**6**	**4**		5
Flip						3
Cut	**6**	**6**	**6**	**6**		4

Table 1: Attack results for A to D, + means strong mark, numbers are the total number of affected examples, bold = destroyed, normal = changed

The audibility of the attacks is highly dependent on the attacked material: The same attack could be perceived in one example and was inaudible in the next. For 'city' (Figure 3) the cut sample attack produced test results of 4, a good perceived quality while for 'Serenade' (Figure 4) the same attack was rated as a significant distortion by trained listeners. When we compare the results for 'city' and 'serenade', we can see that 'serenade' got more low ratings than 'city'. The human ear is more sensitive to speech quality than to noises, therefore the listeners are more critical. Pitch shifting, the most effective attack against watermarks, was often rated as low quality and therefore can not be used as a standard attack against watermarking algorithms.

[2] None of them are related to the authors.

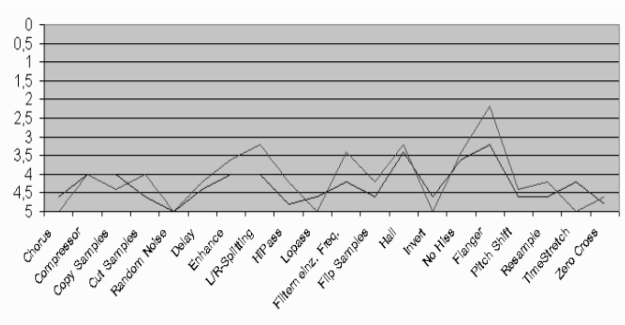

Figure 3 – Subjective test results for 'city'; the darker line stands for the untrained listeners.

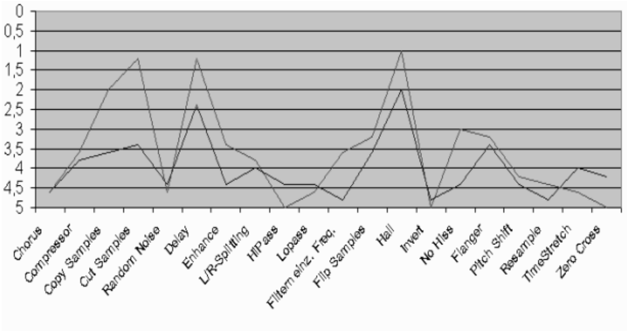

Figure 4 – Subjective test results for 'serenade'; the darker line stands for the untrained listeners.

This makes it necessary for an automatic system like the Stirmark Benchmark to either analyse the audio signal and use only specific allowed attacks or to implement quality checking to provide acceptable test results with any test material. Another way is to select certain test samples and manually adjust the attacks to them. If we choose the six examples used in our tests, we can find the parameter settings where the attacks are still not audible and set these as maximums to the automatic attacks.

The second method is much more practicable at the moment. Another series of testing will be necessary to find the best parameter settings. Problems could occur if the watermarking algorithms change the signal in a certain way so that the following attacks otherwise inaudible become audible. This could happen for example if the amount of high frequencies is increased or noise is added by the watermarking scheme and an equalizer attack then increases again the strength of the high frequency bands.

5. Conclusions and future work

In this paper we have briefly described the architecture of a fully automated evaluation tool for digital watermarking schemes and several new audio tests that we plan to include to this tool. It is the logical continuation of the early benchmark introduced into StirMark.

The results of the audio attacks show how important it is to use different attacks and different audio materials. Every watermarking algorithm has its own weakness against certain types of attack. The audibility of the different attacks varies with the selected audio material. The robustness of embedded watermarks is also dependent on the marked material. There are also more attack types to be tested – a very important one is mp3 or wma [5] compression and similar lossy compression algorithms. We will provide a comparison of the influence of different algorithms on audio material.

Another interesting experiment will be to combine the attacks in a sequence. With the information about audibility and attack success rates, an inaudible but effective group can be set up removing all watermarks while not degrading quality.

Hopefully this new generation of watermarking testing tool will be very useful to the watermarking community as it will provide a standard way of testing and it will allow fair comparison between different watermarking schemes.

6. References

[1] Jonathan K. Su and Bernd Girod. Fundamental performance limits of power-spectrum condition-compliant watermarks. In Ping Wah Wong and Edward J. Delp, editors, proceedings of *electronic imaging, security and watermarking of multimedia contents II*, vol. 3971, pp. 314–325, San Jose, California, U.S.A., 24–26 January 2000. The Society for imaging science and technology (I.S.&T.) and the international Society for optical engineering (SPIE). ISSN 0277-786X. ISBN 0-8194-3589-9.

[2] Martin Kutter. Watermark copy attack. In Ping Wah Wong and Edward J. Delp, editors, proceedings of *electronic imaging, security and watermarking of multimedia contents II*, vol. 3971, pp. 371–380, San Jose, California, U.S.A., 24–26 January 2000. The Society for imaging science and technology (I.S.&T.) and the international Society for optical engineering (SPIE). ISSN 0277-786X. ISBN 0-8194-3589-9.

[3] Fabien A. P. Petitcolas, Ross J. Anderson and Markus G. Kuhn. Attacks on copyright marking systems. In David Aucsmith, editor, *second workshop on information hiding*, in vol. 1525 of *lecture notes in computer science*, pp. 218–238, Portland, Oregon, U.S.A., 14–17 April, 1998. ISBN 3-540-65386-4.

[4] David Kirby, Kaoru Watanabe, *Report on the formal subjective Listening Tests of MPEG-2 NBC multichannel audio coding*, IOFS-Report, 1996.

[5] Henrique Malvar. A Modulated Complex Lapped Transform and its Applications to Audio Processing. ICASSP, 1998.