Re-Synchronizing audio watermarking after non-linear time stretching

Martin Steinebach[a], Sascha Zmudzinski[a], Stefan Nürnberger[b]

[a]Fraunhofer SIT, Rheinstr. 75, 64295 Darmstadt, Germany;
[b]TU Darmstadt, Dept. of Computer Science, Hochschulstr. 10, 64289 Darmstadt, Germany

ABSTRACT

Digital audio watermarking today is robust to many common attacks, including lossy compression and digital-to-analogue conversion. One robustness and security challenge, still, is time-stretching. This operation speeds up or slows down the playback speed while preserving the tone pitch. Although inaudible for an uninformed listener if smoothly applied, time-stretching can be confusing for a blind watermark detection algorithm. We introduce a non-blind approach for reconstructing the original timing based on dynamic time warping. Our experiments show that the approach is successful even if non-linear stretching was applied. Our solution can significantly increase the robustness and security of every audio watermarking scheme that is dependent on precise timing conditions at detection time.

Keywords: audio watermarking, security, robustness, time stretching, pitch shifting, attacks,

1. INTRODUCTION

Digital watermarking is an accepted measure to protect intellectual property rights and to reduce illegal distribution of music and video files in the Internet [Di00] [CMB02]. One well-known weakness of watermarking algorithms are de-synchronization attacks [SPR+01]. Here the embedded watermark information is not removed from a media file but only slightly moved to a position where the watermark detection algorithm will not try to retrieve the watermark. Time stretching, the slight increase or decrease of audio playing time without pitch modification or significant quality loss, is one known example for de-synchronization attacks on audio material.
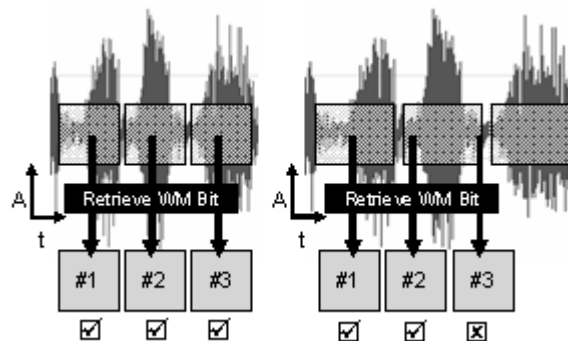

Figure 1: Effect of time stretching on watermark retrieval; left: before time stretching, right: after time stretching

Figure 1 shows how de-synchronization by time stretching works: On the left side a typical example of an audio watermark is given. Individual watermark message bits #1 to #3 are embedded in consecutive frames of a defined length. A detection algorithm will synchronize with the help of a sync pattern then will try to retrieve bits #1 to #3.

On the right side the effects of a time-stretching attack are shown. The audio material and the embedded watermarking frame are now longer than before. After synchronization the retrieval process will detect bit #1 and also may be able to correctly retrieve bit #2. But the original frame length will make the retriever try to detect #3 at a position where now still bit #2 is in effect as the frames are now longer. Retrieval errors are the consequence.

This is obviously a challenge where a repetitive re-synchronization would help. But synchronization in audio watermarking often requires much of the watermarking capacity. Therefore synchronization after each bit would render an algorithm robust but useless due to minimal capacity.

2. TECHNICAL BACKGROUND

This chapter introduces the spread-spectrum Patchwork audio watermarking embedding and detection strategy we were analyzing and time-stretching and pitch-shifting in general.

2.1 Digital Audio Watermarking Algorithm

Digital watermarking schemes have been under research and development for various types of multimedia data for many years, including audio formats like PCM, mp3 or MIDI. In this work we focus on digital PCM audio data. Several approaches for PCM audio watermarking have been introduced in the literature, like in [BTH96], in [CMB02] or also [Stei04]. The latter algorithm is the base of this work.

Our watermarking technique works in the spectrum of uncompressed PCM audio data and follows a spread spectrum Patchwork approach [BGML96]. To provide better understanding of the procedure adopted in this work the basic features of the technique are explained.

1. As a first step, the algorithm for incorporating the watermark transforms the PCM source signal into the frequency spectrum using a Fast Fourier Transform (FFT). Segments of the source signal are created, also known as windows or frames, with the energies contained in the frequency bands.

2. Then, the watermark is incorporated by the deliberate alternation of energy coefficients according to the Patchwork embedding rule: two non-overlapping subsets A and B are selected from all energy coefficients. If the message bit is '0' (or '1', respectively) all coefficients in A are increased in energy while all in B are decreased (and vice versa, respectively). This introduces a significant difference in the mean energies in the two subsets. The selection of the coefficients is done pseudo-randomly dependent on a secret key. The strength of the changes in the frequency domain is controlled by a psycho-acoustic model such that audible distortions are avoided as good as possible.

3. In the final step, the algorithm converts the data back into PCM format. In order to prevent noticeable gaps between the segments, special mechanisms are used to fade between marked and unmarked parts of the audio.

Using the method described above, up to 21 bits of information can be embedded in one second of audio data. In order to achieve better reliability in the technique, only 7 bits per second are used in the version used for this work. The watermark message is generally embedded several times within a media file.

In order to retrieve the watermark, the algorithm again requires the secret key: Without knowledge of the key it cannot be determined which frequencies were changed without the original being present for comparison. Thus, the hidden message cannot be reconstructed by an unauthorized person.

The retrieval process consists of the following steps:

1. Since the watermark is embedded in the frequency band as described above, a conversion of the PCM information to the Fourier spectrum has to be carried out first of all. The FFT is applied again for this reason. Appropriate synchronizations mechanisms to detect the correct file positions are not discussed here and can be looked up in the references given before.

2. The watermark message bit is then retrieved. Using the correct key, the same subsets A and B of FFT coefficients are selected. Now the minimal difference in the mean energies between the subsets can be analyzed. The message bits of the watermark originally incorporated can be retrieved by evaluating this energy difference. If ist sign positive, the message bit is '0' (and vice versa, respectively). The absolute value can be considered as the goodness of the detection or the degree of confidence. We will refer to this detection as detection score in the following.

2.2 Time Stretching and pitch shifting

Time-stretching is a signal processing operation that moderately changes the playback speed of an audio recording. Unlike slowing down or speeding up the playback speed of, for example, a vinyl record disc, the tone pitch usually is not modified by digital time-stretching algorithms. That is, the fundamental frequency of a recorded musical tone or voice is not changed. Time stretching attacks in digital audio watermarking can be seen analog to attacks on image watermarking caused by geometric distortions, e.g. scaling or any other affine transformation.

On the contrary, pitch shifting is an operation that changes the fundamental frequency and the perceived tone pitch without changing the playback speed. Time-stretching and pitch shifting are common processing operation in audio post-production, playback and broadcasting. Usually this is achieved by means of scaling and interpolation in the spectral domain or granular synthesis methods.

For example, time-stretching is used to adapt and align music songs with respect to their measure and their "beats per minute" (BPM) rate. This is highly desired when two music songs are mixed and cross-faded in a discotheque or in a radio show. Here, one of the songs is increased / decreased in playback speed by a constant stretching factor such that its BPM rate matches the BPM rate of the other song and both beats can be synchronized. Another example is the DAISY standard for audio books. Here, many DAISY player devices provide time-stretching to speed up or slow down the play back speed of the reader for better comprehension of the "digital talking book".

Without further ado, common spread-spectrum audio watermarking algorithms usually are not robust to time stretching and pitch shifting in the first place, if a blind watermark detector is involved. The cover data is a time-dependent sequence of digital audio samples and the watermark message symbols usually are only present at particular positions in time and in the spectrum. Thus, significant time-stretching or pitch shifting can cause a correspondent blind detector to fail because it attempts (in vain) to detect the watermark message symbol at wrong positions in time or in the spectrum.
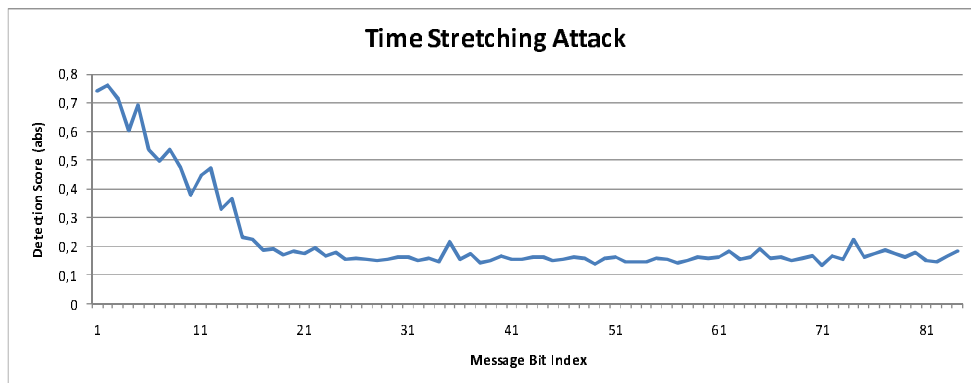

Figure 2: Average detection score for all bits in a 84 bit watermark message;
Detection from time stretched audio content

This can easily be seen from the following example: We embedded an arbitrary binary message of length 84 bit in an audio file using the spread-spectrum Patchwork approach as described in the previous section. Then, the file was subject to time-stretching by 1% during the duration of the file. In Fig. 2 we can see the detection score for each of the retrieved message bits, i.e. the degree of confidence of the detection. We can clearly see that beyond bit position 15, the score significantly drops to low absolute values. This can be explained by the increasing delay introduced by the time-stretching. After this position the detector is completely off the correct positions in time and the remaining message bits are not found which results in extremely low detection scores.

3. STATE OF THE ART

In the previous chapter we addressed certain aspects of vulnerability of audio watermarking with respect to time-stretching and pitch shifting. This chapter discusses measures that enable audio watermarking detector to detect successfully even if non-linear time-stretching is applied.

In the following, we focus on the aspect of time-stretching only. Fortunately, common audio watermarking approaches are able to detect and retrieve the watermark message successfully if the first time-stretching operation is reversed by re-stretching the audio again with the inverted stretching factor(s). This is given by the fact that common time-stretching filters usually do not introduce significant artifacts that interfere with the embedded message even when double time-stretching is applied.

Reversing the time-stretching requires to have the knowledge about the original stretching factor(s), be it constant or linearly increasing (like in the example in the previous section). Then, reversing is rather trivial to handle in practice as the related re-stretching functionality is provided by common audio editor software.

But the robustness to time-stretching is much more challenging, when stretching factors are arbitrarily varying over time. Such time-variant time-stretching can even be seen as a serious security attack on digital audio watermarking. The challenge here is to estimate the stretching factors or the correspondent delay for all points in time in the audio file. Several approaches for successful watermark detection after time-stretching shall be introduced in the following.

3.1 Manual approach

One obvious approach can be seen by reversing the previously applied time-stretching manually with time-stretching filter in any standard audio editor software. For example, if the exact duration of the file or its original BPM rate of a music song is known a priori (e.g. taken from a CD cover) and a constant stretching factor can be assumed, the original playback speed can be restored with minimal effort and consecutive watermark detection will be sufficiently precise and finally successful.

3.2 Brute-force approach

A brute-force reverse stretching is done by testing a set of stretching factors from a predefined range followed by watermark detection. The search range can be limited to a feasible interval, for example from +10% to -10%, to reduce the computational effort. This becomes necessary if the characteristics of the previous time-stretching are ambiguous, i.e. if the stretching factor(s) are unknown and if it is obscure if constant or arbitrarily time-variant stretching was applied. Usually both, time-stretching and watermark detection, are computationally demanding operations. Even worse, very small step sizes are required (for example 0.1%) because the reconstruction and re-alignment must be very precise.

An alternative to this approach can be seen by modifying the detector algorithm itself such that it anticipates time-stretching attacks and ab initio modifies the detection time positions internally accordingly. Although actual re-stretching becomes obsolete, this approach is still computationally demanding as the internal step size of the brute force search must be very small again.

Although all excessive strategies explained before are (in the end) successful for constant or linearly increasing stretching factors, time-variant time-stretching can hardly be reversed with sufficient precision.

3.3 Forensic approach

Forensic approaches are based on the assumption that many common, even seamless, post-processing or filtering operations, introduce statistical artifacts or anomalies that can be detected and analyzed by specially designed detection algorithms, mainly for image data [GK+07] [MS07]. For example, decreasing the playback speed by time-stretching usually is based on dividing the audio in small audio sections (granules) and repeating and fading those granules appropriately. This granular synthesis causes the audio material to show certain time-invariant similarity and correlation properties between consecutive sections that un-stretched audio data usually does not show. Even the value of the stretching factors can be estimated from the characteristics of statistical anomalies by forensic means. No access to the original and un-stretched material is required here, i.e. using forensic is a completely blind approach.

3.4 Non-Blind approach

Another obvious approach is to make use of the original un-stretched cover audio – if it is available at detection time – which is referred to as non-blind or informed detection in the literature. As we will introduce in the next chapter, the

original audio can be used to estimate the characteristics of the time-stretching attack and to match and re-align the time-stretched audio to the cover.

Another approach in the literature is discussed by combining robust hashing and digital watermarking. In the video domain, in [HKM2005] the authors use robust hashes extracted at the watermark embedding position and stored in a database to later re-synchronize the watermark. The marked video is scanned for the hash stored in the database and the watermark is retrieved at the position the hash is found. For audio, in [BVL2004] a method also is proposed to use extracted and stored hashes to re-synchronize the watermarks. While this method may help to retrieve the embedded watermarks, the obvious drawback is the need to have the stored hashes available at watermark retrieval.

4. CONCEPT AND IMPLEMENTATION

This chapter introduces our approach for successful reversing of previously applied time-stretching or pitch-shifting using dynamic time warping.

Given two versions of the same audio content, one version being the original O and the second version being the manipulated one M. The task is to match the manipulated version M as close as possible to the original version O, concerning its frequency and timing characteristics, without damaging the embedded watermark that is hidden in M. The original file O may not even necessarily have a watermark embedded, so that the detection processes discussed herein should be able to successfully re-align the manipulated content to the original, of which the watermark is obviously unknown and therefore needs to be reconstructed.

The time stretching applied to the audio data can be non-linear. Assume the audio content is divided in small consecutive chunks $c_1, c_2, ..., c_n$ which, concatenated together, form the manipulated content $M = c_1||c_2||c_3|| ... ||c_n$. An approximation to the actual non-linear curve that represents the applied time stretch would be to stretch each chunk c_i linearly by a factor s_i' so that the time stretched chunks $c_i' = c_i * s_i'$ concatenated would form the closest match to the original version $O \cong c_1'||c_2'|| ... ||c_n'$.

4.1 Re-Stretching

Let us now assume the factors s_i' (corresponding to the chunks c_i) are found in ascending order .. That means, when finding the best match for factor s_i' we can assume that $\forall k < i : s_k'$ are already the closest matches. This implies that the starting position of chunk c_i is identical to its corresponding position in the original file O, but their stretching may differ.

To find the corresponding stretch factor s' either a brute force match to minimize the difference can be used or dynamic time warping (DTW), as introduced in [CB94], can be applied. DTW is a technique that assesses the similarity between two given time-series, i.e. our given audio signals, and allows an optimal re-alignment if the audio signals are arbitrarily delayed ("warped") compared to each other. The principle is, at first, to measure the distance between the audio signal using an appropriate distance measure, e.g. Euclidean distance. Then, a dynamic programming approach is used to align the audio signals such that this distance is minimized. DTW is commonly used for example in single word speech recognition or biometry, where biometric data needs to be matched to a given set of enrolment data.

The naive way to solve the issue would be, at first, to represent the audio content in a two-dimensional representation of time and frequency. The chunks must then overlap to a certain degree as the frequency spectrum gathered always represents a period of time and underlies the usual frequency/time resolution tradeoff. Then, the two versions must be matched applying a two-dimensional DTW. This is computationally very demanding, because the two-dimensional DTW turns out to be NP-complete. If it can be assumed that no pitch shifting was applied, the matching can be done along the time axis only. That is, the problem can be reduced to one-dimensional DTW which can be solved in polynomial time. For that purpose, the distance function for the DTW between two spectra has to be expressed in a real value, which is smaller, when the spectra a more similar, and greater, when they differ more.

4.2 Pitch and Loudness Invariant Space

Both versions (O and M) are transformed to a pitch shift invariant space representation under the assumption that time stretching only was applied (possibly in a non-linear way).

No matter how a frequency spectrum is stretched, the relative distances between the peaks always stay the same. The algorithm introduced herein uses this observation to create a representation that is invariant to pitch shifting. The algorithm works as follows:

1. Get all peaks of the given spectrum and sort them in descending order by their absolute magnitude: $E = \{e_1, e_2, \dots, e_{NUMPEAKS} \mid i < j \Rightarrow e_i \geq e_j\}$.

2. Get the positions p_i of the peaks e_i in the spectrum.

3. Create a d-dimensional vector $v = (p_1, p_2, \dots, p_{NUMPEAKS}, 0, 0, \dots, 0)$ consisting of $NUMPEAKS$ peak positions and filled with zeros up to the dth dimension.

4. Normalize this vector $|v| = 1$ so that every vector built out of the corresponding spectrum has the same norm.

Note: The value of d can either be set to a fixed number (maximum number of expected peaks) or can be calculated, given that a peak always consists of a maximum surrounded by at least two adjacent neighbours of lower amplitude. The worst case would be an alternate maximum/minimum spectrum which implies the smallest upper bound for d is half the number of bins of the FFT spectrum.

An example is given in figure 3. Here, the peaks in order by amplitudes are: (70 dBA, 40 dbA, 30 dbA) which corresponds to peaks index #1, #3 and #2 respectively. Their according frequency indexes are (10, 51, 42).
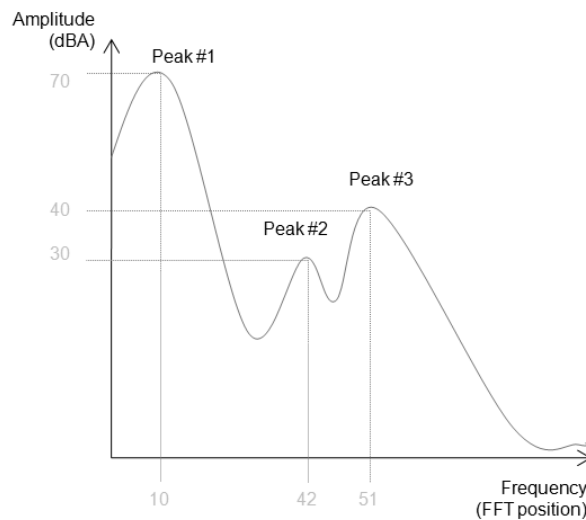

Figure 3: example FFT spectrum

Assume we set the maximum number of peaks to expect to $d = 5$, this leads to a vector $v = (10, 51, 42, 0, 0)$ with an Euclidean length of $\sim 66{,}8$, thus normaliziation gives $v = (0.15, 0.76, 0.63, 0, 0)$.

As the positions of the peaks are saved in descending order of their magnitude (and therefore amplitude), the amplitude information is only implicitly available. For all monotonously increasing functions $f_{mon}: R \rightarrow R$ applied to all amplitudes of the original frequency of the spectrum, the order is preserved. This means this representation is even invariant to changes in volume and to (logarithmic) audio compressor functions.

For the DTW distance to work, similarities $y \in [0,1]$ have to be found between two vector representations. The similarity should be closest ($y = 1$) if the two spectra are identical, but may be at different volume levels. The similarity should remain $y = 1$ for pitch shifted versions and should tend towards $y = 0$ the more the spectra differ.

To achieve this, the angle similarity (Cosine Similarity) is used:

$$similarity(A, B) = \cos(\phi) = \frac{<A, B>}{|A| * |B|}$$

where $<A, B>$ defines the scalar product.

As result of the DTW-transform, we obtain an optimal estimation of the temporal delay and the audio signal with the original time-stretching almost completely reversed. This reconstructed audio signal is then subject to watermark detection as described above.

Due to aliasing effects when performing a Fourier transformation, overlapping FFTs that are only slightly shifted in time domain, feature slight noise in frequency domain when compared to each other. As the comparison process of the spectra is crucial to this algorithm, a simple 3x3 or 5x5 median filter is applied to the consecutive FFT frames, thus to the STFT.
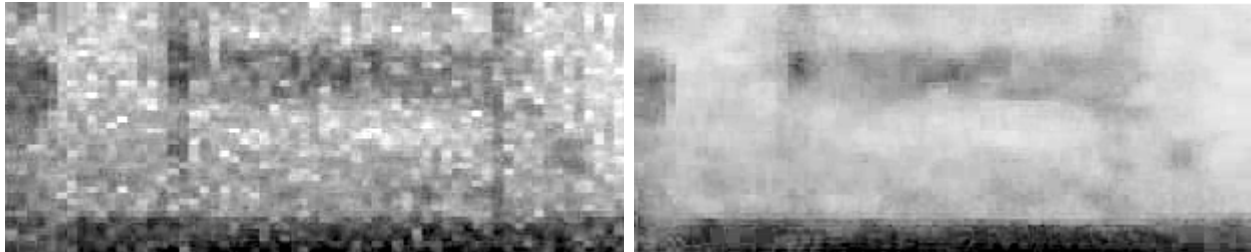

Figure 4 - left: Original STFT in time-frequency domain (featuring alias noise), right: median filter applied

5. TEST RESULTS

To evaluate the performance of the approach described in section 4, a set of 110 audio files, each of 120 seconds playtime, is watermarked. The message length is 32 bit, error correction coding is used as well as a CRC checksum. Parameterization with respect to watermarking strength, frame redundancy and similar features is applied as in standard application scenarios for audio books and music.

In this way a maximum of 6 complete watermarks can be embedded in each audio track. As an attack we use a gliding time stretch stating at -5% speed at the beginning of the audio track and +5% speed at the end. We then try to detect the watermark directly after the attack, after a manual reversion and after multiple rounds of automatic repairs.

Figure 5 summarizes the test results at the different stages of the test. The number of audio files is provided in which the correct audio watermark has been retrieved.

- No attack: Detection result directly after embedding. Here in all 110 test files the correct watermark was retrieved.
- Post attack: Detection result after the gliding pitch shift attack without countermeasures. The watermark could only be retrieved in six audio files.
- Manual: We apply a gilding time stretch with inverse parameters, starting at +5% and ending at -5%. In this way the watermark was found in 105 of 110 files.
- Repair1: Detection result after the automatic repair routine run once. In 46 of the 110 attacked files the watermark is retrieved. This is less than half of the success rate of a manual repair, but almost 8 times the success rate without countermeasures
- Repair2: Detection results after a second run, using the result of repair1 as the input for the repair. Only 20 times the watermark was found, showing a decrease in success rate in multiple applications of our algorithm.
- Repair3: The third run of the auto-repair, reducing the success rate even more.
- Repair sum: This is the overall sum of audio files in which the watermark was found in one of the three repair rounds. An audio file is counted if a watermark was retrieved successfully in at least one of the three rounds. The additional two rounds only result in one successful detection more compared to a single run.

The auto-repair is less successful than a manual repair, but still increases the number of successes when detecting watermarks in attacked files significantly. And other than a manual repair it can be part of an automatic process, only requiring the original audio files as a comparison. In this way the efficiency of watermark detection after time stretching attacks can be increased at the cost of detection successes. Still, a detection rate of almost 50% for an individual file should be sufficient to discourage misuse of audio files at medium or large scale.
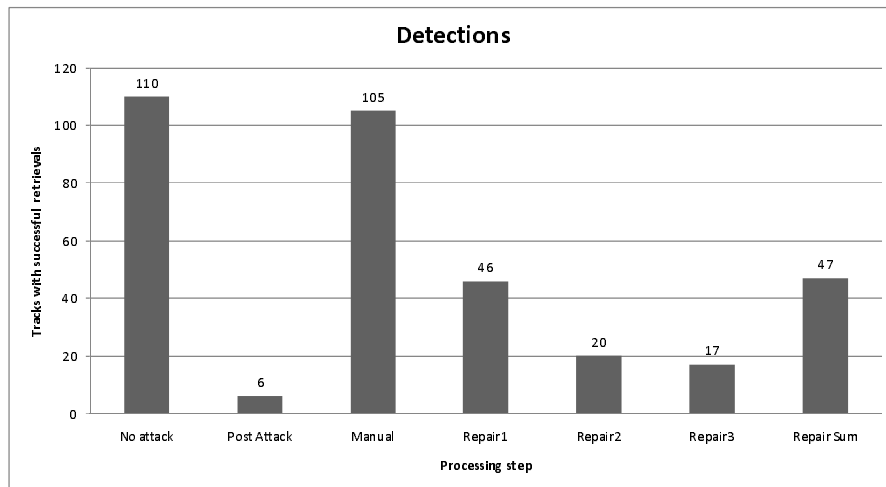

Figure 5: Detection successes at the different test stages

From the results it can be seen that recursive repairs do not improve that detection success rate. Figure 6 shows the average detections scores of the 84 message bits. It can be seen that the average detection score decreases significantly with the second and third repair stage. That is, the first repair step already represents the best match in terms of the chosen distance measure.

The reason can be seen as follows: After a valid stretch factor for each consecutive chunk was found, this particular chunk is stretched, so that its end matches the beginning of the following chunk in the original file. Hence, the applied stretching forms a piecewise linear function that approximates the non-linear stretching of the attacked file. The re-synthesis of each linear chunk is done by interpolation in frequency space, i.e. the spectrum of each chunk is crushed or stretched to fit in its desired length. Due to the interpolation, the associated phases of each FFT bin are lost (as they are now arbitrary) and must be estimated, which results in poor quality compared to state-of-the-art time-stretching algorithms. This simple approach is however sufficient to recover the watermark from approximately half of the files. When in turn applied repeatedly, the quality degradation introduced is audible and even hinders the successful detection process.

Figure 6: Detection scores of the 84 message bits; upper curve: score from marked audio (no stretching attack);
lower three curves: score after 1, 2 and 3 recursive stages of repairing

Another interesting aspect of the test was the file position at which the most watermarks where retrieved correctly. As stated above, the watermark is embedded 6 times over the 120 seconds of the test files, roughly every 20 seconds. In Figure 7 we see the number of correctly retrieved watermarks at different file positions for the stages "No Attack", "Post Attack", "Manual" and "Repair1". The detection position is the second in which the successfully retrieved watermark starts. At stage "No Attack" the watermark is detected with similar success at all positions. Directly after the attack only in the middle of the audio file between second 60 and 65 watermarks can be retrieved. Here the effect of the gliding time stretch is the weakest, crossing factor 0% in its constant change from -5% to +5%. "Manual" repair is most successful in the area between second 20 and 40.
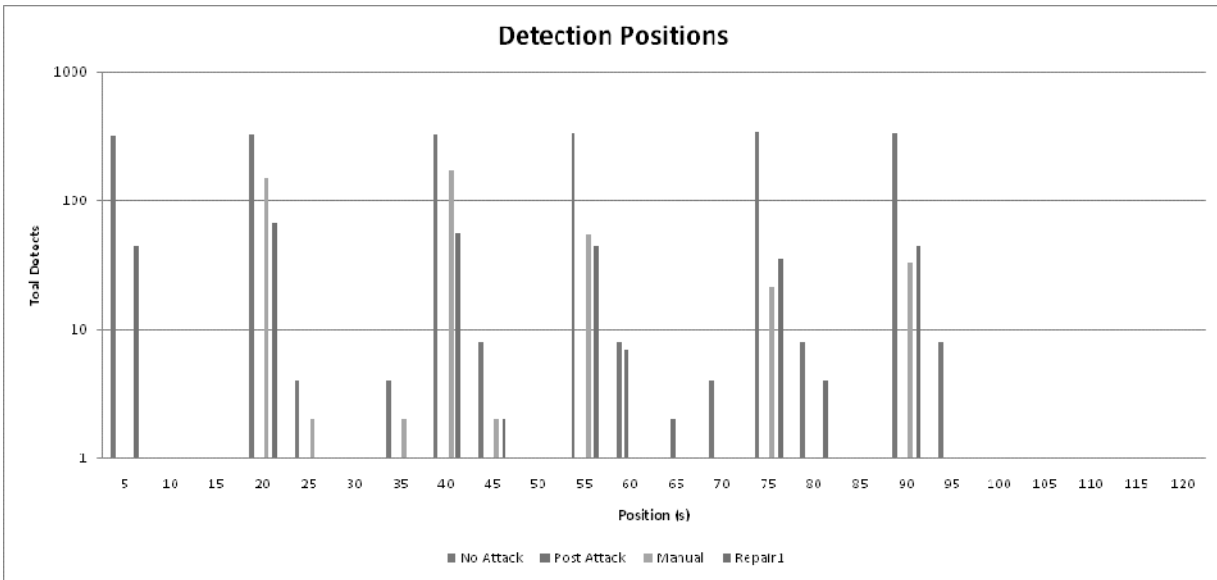

Figure 7: Detection position histogram

Only at the beginning of the file no successes can be gained with the approach, hinting at too many distortions caused by the time stretching effect applied twice with factor 5%. The "auto-repair" shows an equal distribution of successes over the whole duration with its peak also in the area between second 20 and 40. So while manual repair is overall more successful than automatic repair, in some cases, especially at the beginning of the file, our algorithm can allow detections where manual repair fails.

6. FUTURE WORK AND CONCLUSION

Digital watermarking is an accepted measure to protect intellectual property rights and to reduce illegal distribution of music and video files in the Internet. One specific requirement on audio watermarking is the robustness with respect to time-stretching, which is a common post-processing operation for music and audio book productions in some cases. Time-stretching, although inaudible if smoothly applied, can even be seen as a security attack for watermarking: Without further ado, many audio watermarking approaches might not be able to detect a watermark if the watermarked content was subject to smoothly time-varying time-stretching.

We introduce an approach using dynamic time warping (DTW) that estimates the stretching factor(s), reverses the initial stretching attack and, thus, reconstructs the original timing conditions in the audio. It is a non-blind approach, i.e. the reconstruction is only possible if the original, un-stretched, audio is available at detection time. Experimental results show that the reconstruction is precise enough such that the detection success rate of the watermark detection can be significantly improved. The experimental evaluation was done using one particular audio watermarking algorithm taken from our previous work. Nevertheless, the security of every audio watermarking, that requires precise timing

information, can be increased. That makes it much more difficult for an adversary to circumvent watermarking-based protection mechanisms.

In our future work we will address the robustness and security with respect to arbitrary pitch-shifting. First results show that it is promising to extend the existing approach to use the pitch-invariant representation to further undo pitch shifting, besides time-stretching only.

ACKNOWLEDGEMENT

REFERENCES

[AM03]   A. M. Alattar, J. Meyer, Watermark Re-synchronization Using Log-polar Mapping of Image Auto-correlation, Proceedings of the 2003 International Symposium on Circuits and Systems, ISCAS '03. vol. 2, pp. 928-931, 2003.

[BTH96]   Laurence Boney, Ahmed H. Tewfik and Khaled N. Hamdy, Digital Watermarks for Audio Signals, 1996 IEEE Int. Conf. on Multimedia Computing and Systems June 17-23, Hiroshima, Japan, p. 473-480, 1996..

[CB94]   James Clifford, Donald J. Berndt. Using Dynamic Time Warping to Find Patterns in Time Series. Working Notes of the Knowledge Discovery in Databases Workshop. 1994, pp. 359-370. 1994.

[CMB02]   Cox, Miller, Bloom; Digital Watermarking, Academic Press, San Diego, USA, ISBN 1-55860-714-5, 2002

[Di00]   Dittmann; Digitale Wasserzeichen, Springer Verlag, ISBN 3 - 540 - 66661 - 3, 2000

[GK+07]   Gloe, T., Kirchner, M., Winkler, A., and Böhme, R.:. Can we trust digital image forensics? In: Proceedings of the 15th international Conference on Multimedia (Augsburg, Germany, September 25-29, 2007). MULTIMEDIA '07. ACM, New York, NY, 78-86, 2007.

[KH+03]   X. Kang, J. Huang, Y. Shi, Y. Lin, A DWT-DFT Composite watermarking Scheme Robust to Both Affine Transform and JPEG Compression", IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no. 8, pp. 776-786, August 2003.

[MS07]   Mahdian, Saic, On periodic properties of interpolation and their application to image authentication. In Third International Symposium on Information Assurance and Security, pages 439-446, 2007.

[PAK98]   Fabien A. P. Petitcolas, Ross J. Anderson, Markus G. Kuhn, Attacks on Copyright Marking Systems, Lecture Notes In Computer Science; Vol. 1525, Proc. of the Second International Workshop on Information, pp. 218-238, ISBN 3-540-65386 4, 1998.

[RP98]   J. J. K. O'Ruanaidh and T. Pun, "Rotation, scale and translation invariant spread spectrum digital image watermarking," Signal Processing, vol. 66, no. 3, pp. 303– 317, 1998.

[SPR+2001]   Steinebach, Martin; Petitcolas, Fabien A. P.; Raynal, Frederic; Dittmann, Jana; Fontaine, Caroline; Seibel, Christian; Fates, Nazim; Croce Ferri, Lucilla (2001). StirMark Benchmark: Audio watermarking attacks. In: Int. Conference on Information Technology: Coding and Computing (ITCC 2001), April 2-4, Las Vegas, Nevada, pp. 49 - 54, ISBN 0-7695-1062-0, 2001.

[Stei2004]   Steinebach, Digitale Wasserzeichen für Audiowasserzeichen, ISBN 3-8322-2507-2, Shaker Verlag, 2004.