Egecan Serbester 2019400231
Meriç Keskin 2019400024

CMPE362 Project

We picked watermarking methods for audio in signal processing for this project. Let's first explain why we chose it in detail.

1. Overview of Audio Watermarking:

When imperceptible information is inserted into audio signals for purposes like copyright protection, authentication, and content identification, it is referred to as audio watermarking. This report gives a general overview of audio watermarking, as well as an introduction to its purpose, uses, and methods for implementing it in MATLAB.

Audio watermarking is the process of embedding hidden information, referred to as a watermark, into an audio signal. The watermark is typically a sequence of bits representing data, ownership information, or authentication details. The primary goal is to ensure the watermark's robustness, imperceptibility, and security.

2. Importance of Audio Watermarking: Audio watermarking plays a crucial role in various applications, including:

a) Copyright Protection: Audio watermarking helps protect intellectual property rights by embedding unique identifiers or ownership information within audio files. This enables copyright holders to identify and claim ownership of their content, even if it is distributed or shared without authorization.[1]

b) Content Authentication: Audio watermarking provides a means to verify the authenticity and integrity of audio content. Watermarks can be used to detect any unauthorized modifications, tampering, or unauthorized distribution of the audio material.[2]

c) **Content Identification**: Watermarks enable the identification and tracking of audio content across different platforms, facilitating monitoring, content recognition, and usage analytics in broadcasting, streaming, and digital media distribution.[3]

[1] According to Barni and Bartolini (2004)

[2] Kundur and Hatzinakos (1999)

[3] According to Cox et al. (2002)

Egecan Serbester 2019400231
Meriç Keskin 2019400024

3. Methods and Techniques for Audio Watermarking in MATLAB:

a) **Time-Domain Audio Watermarking**: This method involves manipulating the audio samples directly in the time domain to embed the watermark. Techniques like Least Significant Bit (LSB) modification or echo hiding can be employed to modify specific sample values. We used this method in homework 2. Thus we didn't mention it once we know this a little bit.

b) Frequency-Domain Audio Watermarking: This method uses methods like the Short-Time Fourier Transform (STFT) or Discrete Fourier Transform, it converts the audio signal to the frequency domain (DFT). By changing the magnitude or phase of the frequency co mponents, the watermark is embedded.

c) Spread Spectrum Audio Watermarking: Spread spectrum techniques spread the watermark signal across multiple audio samples using pseudo-random noise sequences. This method provides robustness against various attacks and offers imperceptible embedding.

d) Echo Hiding Audio Watermarking: Echo hiding watermarking is a technique used to embed watermarks in audio signals by exploiting the properties of human auditory perception. It takes advantage of the fact that the human ear is less sensitive to echoes in the audio signal compared to the original sound.

We chose c) and d) for implementing watermarking audio signals in MatLAB. Let's start with Spread Spectrum Audio Watermarking.

Egecan Serbester 2019400231
Meriç Keskin 2019400024

Spread Spectrum Audio Watermarking

This method derives its name from spread spectrum communication techniques used in wireless communications. The basic idea of spread spectrum audio watermarking is to spread the watermark signal over multiple audio samples using a pseudorandom noise sequence. In this way, the watermark becomes more robust to signal processing operations, attacks, and noise, while being imperceptible to human listeners.

2. Advantages of Spread Spectrum Audio Watermarking: Spread Spectrum Audio Watermarking offers the following advantages:

a) **Robustness**: The spread spectrum method spreads the watermark information over a wide frequency band, making it robust against various attacks such as compression, filtering, and noise addition. The watermark remains recoverable even after signal distortion or intentional manipulation attempts.

b) **Imperceptibility**: Spread spectrum-based embedding guarantees that the watermark is undetectable and does not significantly alter the host audio signal's perceptual propertiesThe human auditory system is less sensitive to the spread spectrum modulated components, maintaining the audio quality.

c) **Capacity**: Spread Spectrum Audio Watermarking provides a high data embedding capacity due to the wide frequency band used for spreading the watermark. This allows for the insertion of larger amounts of watermark data without significant degradation in the audio signal quality.

d) **Security**: The pseudo-random sequences used for spreading the watermark provide a level of security. Without knowledge of the spreading sequence, it becomes difficult for unauthorized parties to remove or tamper with the embedded watermark.

Barni, M., & Bartolini, F. (2004). A survey of watermarking techniques for copyright protection of images. EURASIP Journal on Advances in Signal Processing, 2004(2), 1-28.

Egecan Serbester 2019400231
Meriç Keskin 2019400024

Thanks to these advantages, we picked the Spread Spectrum Audio Watermarking to implement in MatLAB. Let's explain **implementation** part: The corresponding code in SSW.m file.

First, we loaded the host and watermark audio signals with audioread method.
Next, we normalized host and watermark signals to ensure their amplitudes are within a suitable range. We normalized it with dividing its signal to max value of its signal in this way, all signals are in range between -1 and 1.
Then we define the segments for watermark embedding. We pick start and end index for that. Then extract the segment from the host signal where the watermark will be embedded.
After we prapare everything for embedding, we transformed the selected segment of hostSignal into frequency domain using Discrete Fourier Transform (DFT) by applying **fft** function of MatLAB. We do the same thing for watermark signal
Then we define a spreadFactor which controls the strength or intensity of the spreading effect. After that we modulate the frequency components of the segment DFT with the watermark DFT. We did this operations in frequency domain, but we need to place it in a time domain therefore we perform inverse DFT to obtain the watermarked segment with **ifft** function which converts the frequency domain representation back to the time domain. In this way our watermarkedSegment became in time-domain. Lastly, we replaced the watermarked segment in the host signal to put it. As a result, the watermark information is spread across the frequency components of the host audio signal. The spreading effect helps distribute the watermark energy over a wide frequency band, making it more robust against signal processing operations and attacks, while still remaining imperceptible to human listeners.
After that we wrote the watermarked host audio as a "watermarked_host_ssw.wav" with audiowrite function.
Finally, to show that we can extract the watermarked audio from the watermarked_host audio, we extract it. Again to make operations on that audio we first converted its domain to frequency domain, make operations on it and then again converted it to time domain and write it with audiowrite function named "extracted_watermark_ssw.wav". If you have questions about implementation, you can also check comments on the code.
Additionally, you can play the audios in the code from commented out parts.

Egecan Serbester 2019400231
Meriç Keskin 2019400024

Echo Hiding Watermarking

Echo Hiding Watermarking is a technique used for embedding and extracting watermarks in audio signals. It is based on the principle of exploiting the properties of the human auditory system to hide the watermark in the echo region of the audio signal.

The basic idea behind Echo Hiding Watermarking is to introduce a small delay or echo to the original audio signal and then embed the watermark within this echo. The watermark is typically a low-amplitude signal containing information or identification data.
The embedding process involves the following steps:

Echo Hiding Watermarking offers several advantages, such as imperceptibility, robustness against common audio signal processing operations, and resistance to removal attacks. It takes advantage of the psychoacoustic properties of the human auditory system to hide the watermark in a perceptually transparent manner.

Echo Hiding Watermarking combines both the time and frequency domains to achieve effective watermark embedding and extraction. Therefore, we don't have to convert frequency domain to time domain or vice a versa operations like the Spread Spectrum WM.

Thanks to these advantages, we picked the Echo Hiding Watermarking to implement in MatLAB. Let's explain **implementation** part: The corresponding code in EHW.m file. We use the same host and watermarked audios again.

First we read their values then normalized it as we did in SSW.
After that, we first calculate how much watermarked signal repetition needs to cover whole host audio. Then we use **repmat** function which replicates and tiles arrays in order to create larger arrays with repeated patterns. Then we had a repeated watermarkSignals has the same length with host signal.

Then we defined echo parameters which are delay and gain. Delay is a delay between original signal and its echoed version. Gain on the other hand determine the echo signal's amplitude. We made it 0.05 to hear it a little bit, if you don't want to hear it you can decrease this value.
Then we calculated the echoLength for watermarking.
Then create zero-filled column vector for echo then the corresponding portion of the **repeatSignal** (the repeated watermark signal) is multiplied by the **gain** value and assigned to the echo vector.
After that we obtained the watermarked by **watermarkedSignal = hostSignal + echo(1:length(hostSignal));** which seems very easy to implement.
Then we write it with name "`watermarked_host_ehw.wav`"

After that we extracted the watermarked audio from host which includes the watermark. We did it by simply subtract watermarkedSignal by original host signal and divide the result the gain to get the original amplitude of the watermark audio. Lastly, we removed the repetitions to obtain one watermark audio instead of repetitions. Finally we write this audio named with 'extracted_watermarked_ehw.wav'.

Conclusion:

We learned a lot of things from this project. Although we knew some practices for watermarking from the lessons, with this project we widen our horizon on this issue. We learned new methods and their advantages as well. Therefore, we think that it is very useful for our knowledge about signal processing. We met an unpredictable difficulty when we develop this project. We couldn't any suitable audio file to watermark on the net easily, we still don't know why. The exciting point for us is, the implementation of Echo Hiding is very easier than the other methods. Including spread spectrum and Time-Domain WaterMarking when we did in PS'es. Therefore maybe in the next years, you can develop a watermarking with this in PSes or maybe introduce this method as well to watermarking.