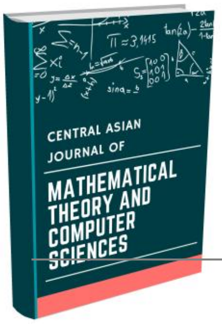


CENTRAL ASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES

https://cajmtcs.centralasianstudies.org

Volume: 03 Issue: 12 | Dec 2022 **ISSN: 2660-5309**

ANALYSIS OF SOUND SIGNALS IN WAV FORMAT IN TELECOMMUNICATION SYSTEMS AND ALGORITHMS FOR THEM PROCESSING

M. A. Kuchkorov

Abstract

In this article, the study of the structure of audio files in WAV format, the development of methods and algorithms for opening audio files in WAV format to computer memory, reading the file title and sound information in it, processing sound data using Haar wavelet, and writing the resulting data into new audio files are carried out.

ARTICLEINFO

Article history:
Received 6 Oct 2022
Revised form 5 Nov 2022
Accepted 10 Dec 2022

Key words: *format, WAV, audio file, WinHex program, file header, audio data, byte, time interval, discretization, amplitude, compressed audio file, uncompressed audio file, Haar wavelet.*

---***---

Today's existing modern computer technologies require the development and application of convenient and suitable methods for high-speed transmission, reception, processing and storage of information. For this reason, lossless transmission and reception of data in telecommunication networks is one of the most urgent issues.

As a result of the rapid development of telecommunication networks at various levels, the amount of data in the network is increasing. In order to find a solution to such problems, a number of researches are being carried out to create new methods and systems that take less volume and less loss in data transmission in networks.

In this article, the issues of opening sound files in WAV format, which is an important format of sound signals, reading and processing information in its structure, transmission and reception over the network, and saving to a new file are considered.

Usually, all types of audio files are divided into two categories:

1. Uncompressed audio files;

2. Compressed audio files.

Uncompressed audio files are recorded in wav, aief, and other formats, while compressed audio files are recorded in .mp3, .aac, .wma, .flas, and other formats. The simplest and most structured audio format of these audio formats is the WAV format. This format is the earliest uncompressed audio format [3].

If we consider a simple WAV file in theory, it is made up of clearly divided parts. One of them is the file header, and the other is the data part (Figure 1).

Файл сарлавҳасида қуйидаги маълумотлардан таркиб топади:

- File size;
- Number of channels;
- Discretization frequency;
- Number of bits in a sample.

Figure 1. WAV file overview

In order to better understand the meaning of the concepts and values in the file header, it is necessary to clarify the issues of the data part of the file and the digitization of sound. Usually, the sound consists of vibrations, and during digitization, it acquires a stair-like appearance. This appearance can be interpreted in such a way that the computer reads and displays a sound of a certain amplitude (height) in an arbitrary short time interval. Thus, the amplitude of the sound changes over short time intervals, and thus the signal takes on a stair-stepped appearance. A short time interval, in turn, determines the signal's discretization frequency. For example, a file with a sampling frequency of 44.1 kHz will have a short time interval of 1/44100 seconds. Today's modern soundboards

It can accept a discretization frequency of up to 192 kHz.

Usually, the signal amplitude determines the volume of the sound in a short time interval and is directly related to the accuracy of the sound. Amplitude is expressed in the form of a numerical value that takes up space from memory or a file, that is, 8, 16, 24, 32 bits. From this, taking into account that 8 bits = 1 byte, it can be understood that some amplitude in a short time interval occupies 1, 2, 3, 4 bytes of memory.

Thus, the larger the number occupies in memory, the greater the range of values of this number, that is, the amplitude.

From this:

- 1 byte – 0..255;
- 2 bytes – 0..65 535;
- 3 bytes – 0..16 777 216;
- 4 bytes – 0..4 294 967 296.

In the mono version of the sound, the amplitude values of the signal are sequentially placed in the file. In the stereo version, the left channel amplitude values are placed first, then the right channel amplitude values, and thus the left and right channel values are alternately placed sequentially. The combined representation of the amplitude and the short time interval is called a sample. [4]

Analyzing the structure of the WAV file "test.wav" used in the research, its complete structure is described in Table 1 below [6][7].

Table 1. WAV file structure

Size	The field	Brief description
4 byte	chunkId	"RIFF" characters in ASCII encoding (0x52494646)
4 byte	chunkSize	The rest of the fragment from here is the audio file size. Header ie chunkId and chunkSize fields size not included (0x65340000)
4 byte	format	stands for "WAVE" (0x57415645)
4 byte	subchunk1Id	stands for "fmt". (0x666d7420)
4 byte	subchunk1Size	For PCM format it is equal to 16 (0x10000000). This is the size of the rest of the audio file fragment from here
2 byte	audioFormat	Audio format is 1 (0x0100) for PCM. A value other than 1 means the file is compressed.
2 byte	numChannels	Number of channels. Mono = 1, Stereo = 2 (0x0100)
4 byte	sampleRate	Sampling frequency: 8000 Hz, 44100 Hz, etc. (0x112b0000)
4 byte	byteRate	Number of bytes transmitted for 1 second display (0x112b0000)
2 byte	blockAlign	Number of bytes in 1 sample for all channels (0x0100)
2 byte	bitsPerSample	Number of bits in a sample (0x0800). Means sound clarity. 8 bit, 16 bit, etc.
4 byte	subchunk2Id	stands for "data" character (0x64617461)
4 byte	subchunk2Size	The number of bytes in the data field, i.e. the size of the data (0x41340000)
45..	data	WAV-data

When the file is opened and analyzed using the WinHex program, the sequence of hexadecimal codes given in Table 1 can be seen realistically as shown in Figure 2.

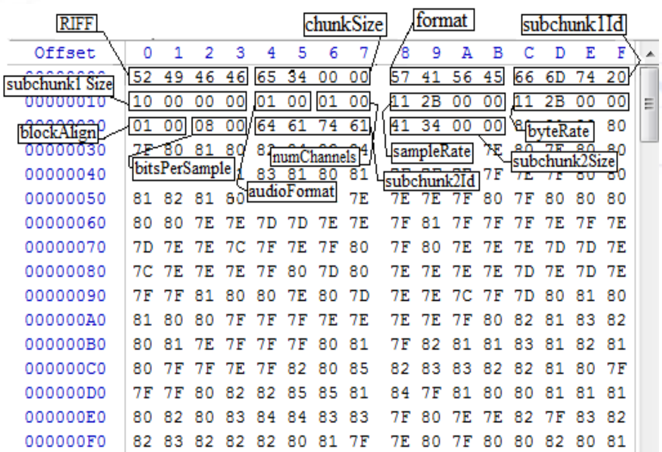


Figure 2. WAV file structure

The WAV file header, in turn, is divided into 3 parts [8]:

- ➢ RIFF part;
- ➢ Fmt part;
- ➢ Data part.

RIFF part: contains the fields "chunkId", "chunkSize" and "format" and serves to identify the format of the multimedia file.

The fmt part: specifies the basic parameters of the sound data in the audio file, including the fields "subchunk1Id", "subchunk1Size", "audioFormat", "numChannels", "sampleRate", "byteRate", "blockAlign", "bitsPerSample".

Data part: consists of "subchunk2Id", "subchunk2Size" and "data" fields and defines the values of sound data in the audio file.

In general, the RIFF and Fmt parts of a WAV file make up the header of the file, and the rest of the data is useful information. As a result of changing the type and size of the audio file, its parameters also change.

The useful data part is the main part of the audio file, on which various operations and calculations are performed by applying various algorithms, and it becomes possible to transfer data in a small volume, quickly and without loss.

As part of the research, a method, algorithm and software were developed for processing sound data of selected audio files in WAV format using Haar wavelet and compressing the data. The structure of the WAVaudio file sound data processing software is shown in Figure 3

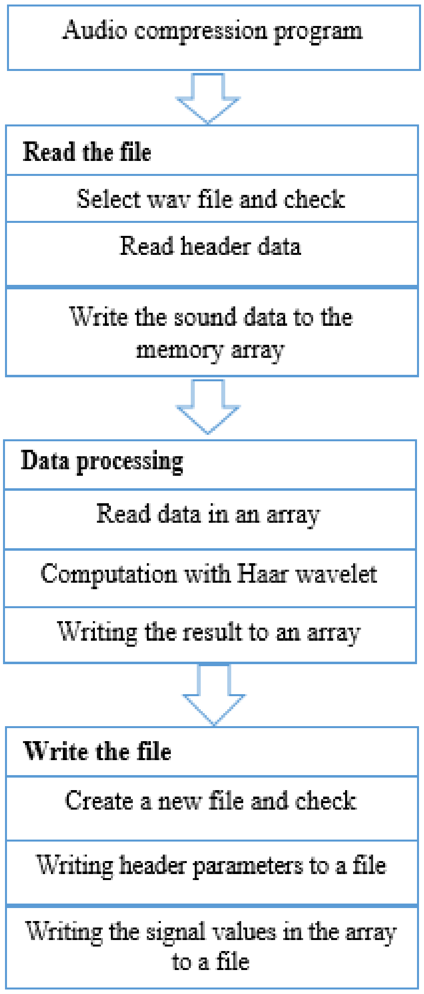


Figure 3. Haar wavelet audio data processing program structure

In the "Read file" block of the program, a file is first selected, the existence of the file and its format (wav format) are checked, and then the values of the sound signal from the data part of the file are written to the memory array.

In the "Information processing" block of the program, the values of the sound signal are read from the memory array as arguments to the variable in pairs, calculated and processed using the Haar wavelet function, high and low frequency values are filtered, and the final results are rewritten to the memory array.

In the program's "Write file" block, a new sound file is created to store the sound signal values, the corresponding header information is placed in the file, and the sound signal values are written from the memory array to the data part of the file, and the file is saved.

In order to store the values of the sound signal, the header of the sound file in WAV format is formed in the S++ programming language in the form of the following sequence of program code and written to the file [5]. After that, that is, after this header, the signal values are placed in the file.

```
struct WAVHEADER
{
   char chunkId[4];
   unsigned long chunkSize;
   char format[4];
   char subchunk1Id[4];
   unsigned long subchunk1Size;
   unsigned short audioFormat;
   unsigned short numChannels;
   unsigned long sampleRate;
   unsigned long byteRate;
   unsigned short blockAlign;
   unsigned short bitsPerSample;
   char subchunk2Id[4];
   unsigned long subchunk2Size;
}
```

Wavelet functions are widely used in real-time compression of one-dimensional (sound) and two-dimensional (image) signals. The Haar wavelet function was also used in the compression of sound signals transmitted through telecommunication systems and networks within the framework of this research.

The Haar wavelet is the simplest and most commonly used type of wavelet. The Haar wavelet is associated with various mathematical operations called Haar transforms. Haar transforms serve as a model for all other wavelet transforms [2].

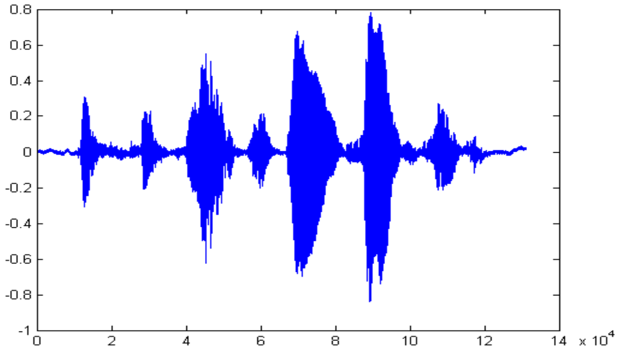
Haar wavelet transforms are expressed by the following formulas:

$$a_m = \frac{f_{2m-1} + f_{2m}}{\sqrt{2}} \qquad (1)$$

$$b_m = \frac{f_{2m-1} - f_{2m}}{\sqrt{2}} \qquad (2)$$

Based on the given expressions (1) and (2), it can be said that when the first and second pairs of data coming in a row are calculated by the expression (1), low frequency values of the signal are formed, when calculated by the expression (2) the high frequency values of the signal are formed will be In this way, all the paired values are calculated and then filtered to separate the low and high frequency values [1].

The initial (before processing) and the resulting (after processing) image of the WAV file of the sample whose values were processed using the Haar wavelet is presented in Figure 4.



a)

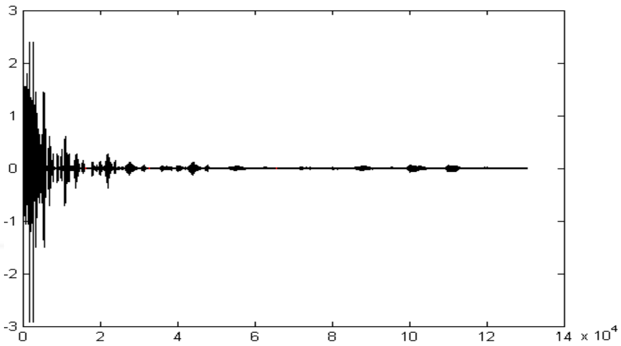


Figure 4. WAV file image a) raw audio file b) Haar wavelet processed audio file

As you can see from the figure, after processing and filtering the audio signal values using Haar wavelet, some of the audio values are equal to "0" value or very close to "0" value. It is also possible not to transmit values equal to or close to this "0" in order to reduce traffic during signal transmission.

In conclusion, as a result of the conducted research, algorithms and software were developed for the analysis and processing of sound signals in WAV format in telecommunication systems. By developing and applying Haar wavelet processing, compression and filtering algorithms to sound signals, the number of signal values in the 1st-stage processing is up to 2 times, and the number of signal values in the 2nd-stage processing

A 4-fold reduction was achieved.

References:

1. “Haar Wavelets” Richard W. Hamming, 1999 CRC Press LLC
2. Khamdamov Utkir. Algorithms for parallel Bitmap image processing based on the Haar Wavelet. *International Conference on Information Science and Communications Technologies*. 2017, 978-1-5386-2168-4 IEEE
3. Chun-Lin, Liu. A Tutorial of the Wavelet Transform. February 23, 2010
4. Oliver R, Martin V. Wavelets and signal processing. IEEE SP MAGAZINE October 1991
5. Юнусова М. Новый этап введения искусственного интеллекта в обществе //Review of law sciences. – 2020. – №. 3. – С. 207-210.
6. Tech 3285 v2. Specification of the Broadcast Wave Format (BWF) Geneva May 2011
7. Andreas Spanias, Ted Painter, Venkatraman Atti. Audio Signal Processing and Coding. John Wiley & Sons, Inc., 2007, ISBN:9780471791478

8. Lili Cui, Shu-Xun Wang, Tanfeng Sun. The application of wavelet analysis and audio compression technology in digital audio watermarking. Proceedings ща International Conference on Neural Networks and Signal Processing, 2003. 1533–1537, Vol.2

9. Deqin Chen, Wenhui Zhang, Zhibo Zhang, Wei Huang, Jia Ao. Audio retrieval based on wavelet transform. IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS). 2017, pages: 531-534