

Audio Processing Method and Apparatus

The present disclosure claims priority to Chinese patent application 202311099735.X, filed on August 29, 2023, the entire contents of which are incorporated herein by reference.

Technical Field

The present disclosure relates to the technical field of data processing, in particular to an audio processing method and apparatus.

Background Art

In work, a user side needs to automatically collect an audio or introduce the audio through a plugin or an SDK (Software Development Kit) to perform audio compression and visualization.

Currently, a method for audio compression and visualization is an FFmpeg method or instance method for audio encoding and compression, which achieves processing through an Linux platform or server side and achieves visual display in a text form, wherein the FFmpeg method refers to an open-source computer program that is developed under the Linux platform and can be configured to record, convert and transform a digital audio; and the instance method is an object method from JavaScript methods and is configured to modify data parameters for audio encoding and compression.

However, both of the above-mentioned processing methods adopt C/C++ for processing and achieve native processing through Android/iOS, and therefore, they cannot directly run on user sides such as browsers, H5 and mini programs; and they both achieve visual display in the text form and do not achieve visualization into audio tracks, which makes it impossible to perform audio visualization analysis.

Summary of the Invention

In view of the above-mentioned problems, the present disclosure provides an audio processing method and apparatus which have a main purpose of achieving audio visualization analysis by direct running on different user sides and visualization into audio tracks.

To solve the above-mentioned technical problems, the present disclosure proposes the following solutions:

In a first aspect, the present disclosure provides an audio processing method, wherein the method includes:

acquiring an execution command;

acquiring, based on the execution command, configuration data corresponding to the execution command from a preset configuration file by using a preset instance method, wherein the preset configuration file at least includes an audio output mode, an audio encoding method, and an audio file format;

acquiring, based on the execution command, audio data by using a preset acquisition rule;

performing, based on the configuration data corresponding to the execution command, audio compression and format processing on the audio data by using a preset processing rule to obtain target audio data; and

performing, based on the target audio data, visualization processing by using a preset visualization rule to obtain visual display corresponding to the target audio data, wherein the visual display includes audio tracks and text annotations.

In a second aspect, the present disclosure provides an audio processing apparatus, wherein the apparatus includes:

a first acquisition unit configured to acquire an execution command;

a second acquisition unit configured to acquire, based on the execution command, configuration data corresponding to the execution command from a preset configuration file by using a preset instance method, wherein the preset configuration file at least includes an audio output mode, an audio encoding method, and an audio file format;

a third acquisition unit configured to acquire, based on the execution command, audio data by using a preset acquisition rule;

a processing unit configured to perform, based on the configuration data corresponding to the execution command, audio compression and format processing on the audio data by using a preset processing rule to obtain target audio data; and

a visualization unit configured to perform, based on the target audio data, visualization processing by using a preset visualization rule to obtain visual display corresponding to the target audio data, wherein the visual display includes audio tracks and text annotations.

In order to achieve the above-mentioned object, according to a third aspect of the present disclosure, provided storage medium includes a stored program, wherein the program, when running, controls a device in which the storage medium is located to perform the audio processing method in the above-mentioned first aspect.

In order to achieve the above-mentioned object, according to a fourth aspect of the present disclosure, provided an electronic device which including a memory, a processor, and a computer program stored in the memory and capable of running on the processor, wherein the processor, when executing the computer program, implements all or parts of steps performed by the audio processing apparatus in the above-mentioned second aspect.

According to the above-mentioned technical solution, the present disclosure provides the audio processing method and apparatus to solve the problems that an FFmpeg method or instance method for audio encoding and compression, which achieves processing through an Linux platform or server side and achieves visual display in a text form, cannot directly run on user sides such as browsers, H5 and mini programs; and they both achieve visual display in the text form and do not achieve visualization into audio tracks, which makes it impossible to perform audio visualization analysis. Therefore, in the present disclosure, an execution command is acquired; based on the execution command, configuration data corresponding to the execution command is acquired from a preset configuration file by using a preset instance method, wherein the preset configuration file at least includes an audio output mode, an audio encoding method, and an audio file format; based on the execution command, audio data is acquired by using a preset acquisition rule; based on the configuration data corresponding to the execution command, audio compression and format processing are performed on the audio data by using a preset processing rule to obtain target audio data; and based on the target audio data, visualization processing is performed by using a preset visualization rule to obtain visual display corresponding to the target audio data, wherein the visual display includes audio tracks and text annotations. According to the present disclosure, the threshold for audio visualization can be lowered, and audio visualization analysis performed by visualization into the audio tracks can be achieved; streams and storage costs can be reduced for users and enterprises through audio compression; different terminals including PCs, H5, mini programs and the like are compatible; and other customized configuration options can be selected for facilitating application in different scenarios.

The above description is only an overview of the technical solutions of the present disclosure. In order to have a clearer understanding of the technical means of the present disclosure, it can be implemented according to the content of the specification. Moreover, in order to make the above-mentioned and other objects, features and advantages of the present disclosure more obvious and understandable, specific implementations of the present disclosure are listed below.

Brief Description of the Drawings

By reading the detailed description of the following preferred implementations, various other advantages and benefits will become clear to those of ordinarily skilled in the art. The accompanying drawings are only for the purpose of illustrating the preferred implementations, and are not considered as limitations on the present disclosure. Moreover, the same reference numerals represent the same components in the entire accompanying drawings. In the accompanying drawings:

Fig. 1 shows a flow diagram of an audio processing method according to the embodiment of the

present disclosure;

Fig. 2 shows a flow diagram of another audio processing method according to the embodiment of the present disclosure;

Fig. 3 shows a block diagram of the configuration of an audio processing apparatus according to the embodiment of the present disclosure; and

Fig. 4 shows a block diagram of the configuration of another audio processing apparatus according to the embodiment of the present disclosure.

Detailed Description of the Invention

Exemplary embodiments of the present disclosure will be described in more detail below with reference to the accompanying drawings. Although the exemplary embodiments of the present disclosure are shown in the accompanying drawings, it should be understood that the present disclosure may be implemented in various forms and should not be limited by the embodiments described herein. On the contrary, these embodiments are provided to enable a more thorough understanding of the present disclosure and to fully convey the scope of the present disclosure to those skilled in the art.

Term explanation:

FFmpeg: It is a set of open-source computer programs that can be configured to record and convert digital audios and videos and transform the same into streams. Licensed under LGPL or GPL license. It provides a complete solution for recording, converting and streaming audios and videos.

Linux: full name GNU/Linux, and is a free-to-use and freely-distributed UNIX-like operating system, its kernel was released by Linus Benedict Torvalds, and it is a POSIX-based multi-user and multitasking operating system that supports a plurality of threads and CPUs. It supports 32-bit and 64-bit hardware and can run major Unix utility software, applications, and network protocols.

Webassembly: abbreviated as wasm, it is a virtual instruction set architecture (Virtual ISA), and the overall architecture includes core ISA definition, binary encoding, definition and execution of program semantics, and application programming interfaces (WebAssembly API) for different embedding environments (such as Web).

H5: HTML5 is a language description method for building web contents. HTML5 is a next-generation standard of the Internet and a language method for building and presenting Internet contents. It is considered as one of core technologies of the Internet.

PCM: PCM (Pulse Code Modulation) is one of encoding methods for digital communication. Its main process is to sample analog signals such as voices and images at regular intervals to make them discretized, quantize sample values by rounding them to the nearest level unit, and also

represent the sample values with a set of binary codes to indicate an amplitude of a sampling pulse.

WAV: It is one of the most common sound file formats and is a standard digital audio file specifically developed for Windows by Microsoft, and the file can record various mono or stereo sound information and can also ensure that the sound is not distorted.

MP3: It is an audio compression technology, and its full name is Moving Picture Experts Group Audio Layer III that is abbreviated as MP3. It is designed to significantly reduce the volume of audio data. By using the technology of the MPEG Audio Layer 3, a music is compressed into a smaller file in a compression ratio of 1:10 or even 1:12, and for most users, the playback sound quality is not significantly reduced as comparison with that of an original uncompressed audio.

Ogg: Its full name is OGGVobis which is an audio compression format similar to a music format such as MP3. Ogg is completely free, open, and not restricted by patents. The extension of an OggVorbis file is “.ogg”. The Ogg file format allows for continuous improvement in size and sound quality without affecting existing encoders or players.

Canvas: Canvas API is a newly added tag in HTML5, is configured to generate images on web pages in real time, and can manipulate image contents, and it is essentially a bitmap that can be manipulated with JavaScript.

JavaScript; It is abbreviated as “JS” and is a function-first lightweight interpreted or just-in-time compiled programming language. Although it is known as a scripting language for developing web pages, it has also been used in a plurality of non-browser environments, and JavaScript is a multi-paradigm dynamic scripting language based on prototype programming and supports object-oriented, imperative, declarative and functional programming paradigms.

ASR: It is a speech recognition technology which is also known as Automatic Speech Recognition which aims to convert lexical contents of a human speech into computer-readable inputs such as keys, binary codes, or character sequences. Unlike speaker recognition and speaker confirmation, the latter attempts to recognize or confirm a speaker who makes the speech rather than the lexical contents contained therein.

Opus: It is a lossy sound encoding format developed by the Xiph.Org Foundation and then standardized by the IETF (Internet Engineering Task Force), and its goal is to replace Speex and Vorbis with a single format containing a sound and a speech, it is suitable for low-latency real-time sound transmission on a network, and its standard format is defined in an RFC 6716 file. The Opus format is an open format with no patents or restrictions on its use.

CDN: Its full name is Content Delivery Network. A CDN system can redirect a request from a user to the nearest service node in real time according to a network stream as well as a connection, a load condition, a distance to the user, response time and other comprehensive information of

each node through an intelligent virtual network based on the existing Internet, which is formed by placing node servers on all places of a network. Its purpose is to enable users to access required contents nearby, thereby solving the problem of Internet congestion and improving response speeds when the users access websites.

Raw: It is commonly known as “source code”. Its meaning is to output an audio format on a compact disc from a coaxial and optical fiber without processing.

For the current adoption of an FFmpeg method or instance method for audio encoding and compression, which achieves processing through an Linux platform or server side and achieves visual display in a text form; however, they both adopt C/C++ for processing and achieve native processing through Android/iOS, and therefore, they cannot directly run on user sides such as browsers, H5 and mini programs; and they both achieve visual display in the text form and do not achieve visualization into audio tracks, which makes it impossible to perform audio visualization analysis. In response to this problem, the inventor came up with the idea that an audio is processed by cooperating a preset instance method and a preset configuration file with secondary audio compression, and thus, visualization analysis is achieved by combining audio tracks with text annotations.

To this end, an embodiment of the present disclosure provides one audio processing method by which audio visualization analysis performed by direct running on different user sides and visualization into audio tracks is achieved, and specific steps thereof are shown in Fig. 1 and include:

101. Acquiring an execution command.

When operating an audio processing item to be completed, a user side needs to firstly call a preset instance method through a preset call address, and then selects, by using the instance method, the audio processing item to be completed, such as audio compression or audio visualization, which is not specifically limited in this embodiment; and when the audio processing item to be processed is selected, the execution command is generated.

102. Based on the execution command, configuration data corresponding to the execution command is acquired from a preset configuration file by using a preset instance method.

The preset configuration file at least includes an audio output mode, an audio encoding method, and an audio file format. The audio output mode can be “Raw” and “PCM”, the audio encoding method can be “waveform encoding” and “hybrid encoding”, and the audio file format can be WAV, MP3, ogg, and other formats, which is not specifically limited in this embodiment.

The execution command can be obtained from step 101, and the execution command is parsed to obtain a configuration item required for completing the selected audio processing item, such as an audio output mode and an audio encoding method, which is not specifically limited in this

embodiment; and the corresponding configuration data is acquired from the preset configuration file according to the required configuration item so as to be used for subsequent processing for the collected audio data.

103. Based on the execution command, audio data is acquired by using a preset acquisition rule. The acquisition of the generated execution command from step 101 requires an authorization from the user side. The audio data can only be obtained when the user side agrees to the collection of the audio data. Therefore, the preset acquisition rule can be set, and the preset acquisition rule can be that it is determined whether the user side authorizes the acquisition of the audio data. When the user side authorizes the acquisition, the audio data can be collected to complete the audio processing item.

104. Based on the configuration data corresponding to the execution command, audio compression and format processing are performed on the audio data by using a preset processing rule to obtain target audio data.

The configuration data corresponding to the execution command can be obtained from step 103, and the configuration data includes data required for completing audio compression and format processing, such as a sampling rate or an audio file format, which is not specifically limited in this embodiment.

According to the configuration data, the target audio data can be obtained by directly compressing the audio data by using a method for encapsulating FFmpeg with Webassembly, or the target audio data can be obtained by performing, based on a compression ratio, primary compression on the audio data and then performing secondary compression by using a method for encapsulating FFmpeg with Webassembly, which is not specifically limited in this embodiment. The target audio data is audio data that has been compressed and subjected to format conversion and can be used for visualization operations.

105. Based on the target audio data, visualization processing is performed by using a preset visualization rule to obtain visual display corresponding to the target audio data.

The visual display includes audio tracks and text annotations. The preset visualization rule may be that visual display of the target audio data in the audio tracks is formed by using Canvas and JavaScript.

The target audio data can be obtained from step 104, the visual display of the target audio data in the audio tracks is formed by using Canvas and JavaScript, and meanwhile, text annotations such as keywords and explanations which are not specifically limited in this embodiment are performed based on a corresponding relationship between an audio and a text. Moreover, on the visual display, audio track analysis and other operations can be performed by highlighting texts or dragging the audio tracks.

Based on the implementation of the above-mentioned embodiment shown in Fig. 1, it can be known that the present disclosure provides an audio processing method. In the present disclosure, an execution command is acquired; based on the execution command, configuration data corresponding to the execution command is acquired from a preset configuration file by using a preset instance method, wherein the preset configuration file at least includes an audio output mode, an audio encoding method, and an audio file format; based on the execution command, audio data is acquired by using a preset acquisition rule; based on the configuration data corresponding to the execution command, audio compression and format processing are performed on the audio data by using a preset processing rule to obtain target audio data; and based on the target audio data, visualization processing is performed by using a preset visualization rule to obtain visual display corresponding to the target audio data, wherein the visual display includes audio tracks and text annotations. According to the present disclosure, the threshold for audio visualization can be lowered, and audio visualization analysis performed by visualization into the audio tracks can be achieved; streams and storage costs can be reduced for users and enterprises through audio compression; different terminals including PC, H5, mini programs and the like are compatible; and other customized configuration options can be selected for facilitating application in different scenarios.

Further, as a refinement and extension of the embodiment shown in Fig. 1, an embodiment of the present disclosure further provides the other audio processing method, as shown in Fig. 2, specific steps thereof are described as follows:

201. A preset instance method is called by using a preset calling rule.

A developer will design the preset instance method before the audio processing item and save the same in a preset storage address. The user side can call the preset instance method through the preset storage address, or upload the same to the user side for use through a mobile storage device. For example, the preset instance method is called by setting a command for downloading or performing introduction through a CDN.

202. A preset configuration file is created by using the preset instance method.

After the preset instance method is called in step 201, the preset configuration file can be created by using the preset instance method. The creation process is to a configuration item and corresponding data by using the preset instance method. For example, the configuration item can select an audio file format, and then, corresponding data is set as "WAV", which is not specifically limited in this embodiment.

203. Acquiring an execution command.

This step is described in conjunction with step 101 in the above-mentioned method, and the same content will not be repeated herein.

204. Based on the execution command, configuration data corresponding to the execution command is acquired from the preset configuration file by using the preset instance method.

This step is described in conjunction with step 102 in the above-mentioned method, and the same content will not be repeated herein.

The execution command can be obtained from step 203, and then, the execution command is parsed to obtain a configuration item required for completing the execution command, such as an audio output mode, an audio encoding method, and an audio file format, which is not specifically limited in this embodiment; and based on the configuration item, the configuration data corresponding to the configuration item is acquired from the preset configuration file by using the preset instance method to obtain the configuration data corresponding to the execution command, for example, the audio output mode is “PCM”, the audio encoding method is “waveform encoding”, and the audio file format is “MP3”.

205. Based on the execution command, audio data is acquired by using a preset acquisition rule.

This step is described in conjunction with step 103 in the above-mentioned method, and the same content will not be repeated herein.

The execution command can be obtained from step 203, and then, it is determined, based on the execution command, whether a user side authorizes the acquisition of the audio data; if so, the audio data is acquired; and if not, it returns to a previous step to perform an action.

206. Based on the configuration data corresponding to the execution command, audio compression and format processing are performed on the audio data by using a preset processing rule to obtain target audio data.

This step is described in conjunction with step 104 in the above-mentioned method, and the same content will not be repeated herein.

The configuration data may include a collection sampling rate and an output sampling rate, which is not specifically limited in this embodiment; after the configuration data corresponding to the execution command is obtained from step 205, firstly, a ratio of the collection sampling rate to the output sampling rate is calculated to obtain a compression ratio, that is, the ratio is the compression ratio; based on the compression ratio, the audio data is compressed by using a preset compression rule to obtain primary compressed audio data; and secondary compression (i.e., opus compression) is performed on the primary compressed audio data by using a method for encapsulating FFmpeg with Webassembly to obtain the target audio data, and thus, the compression capability is further improved.

The step that based on the compression ratio, the audio data is compressed by using a preset compression rule to obtain primary compressed audio data includes: noisy data and null data in the audio data are recognized, and deletion processing is performed to obtain to-be-compressed audio

data, wherein the processing can improve the compression capability; and the to-be-compressed audio data is compressed according to the compression ratio to obtain the primary compressed audio data.

For example:

A user side collects an audio for sampling rate processing, i.e., conversion from 48000/44100Hz to 8000/16000Hz, then performs encoding and compression to compress a pcm stream into an opus data stream, and supports the conversion from the pcm stream into WAV, mp3, ogg and other audio formats for audio stream transmission.

207. Based on the target audio data, visualization processing is performed by using a preset visualization rule to obtain visual display corresponding to the target audio data.

This step is described in conjunction with step 105 in the above-mentioned method, and the same content will not be repeated herein.

The target audio data can be obtained from step 206, and then, the target audio data is recognized by ASR to acquire a corresponding relationship between an audio and a text, which can be stored on a server side; based on the corresponding relationship between the audio and the text as well as the target audio data (that can be acquired from the server side), visual display in audio tracks is formed by using Canvas and JavaScript, that is, JavaScript controls Canvas to draw the audio tracks, wherein the visual display in the audio tracks includes the text, for example, keywords are placed in the audio tracks; and based on the visual display in the audio tracks, audio analysis and annotation are performed by using a preset movement command. The preset movement command can be highlighting texts or dragging audio tracks, that is, audio analysis and annotation can be achieved by highlighting texts and dragging audio tracks.

Based on the above-mentioned implementation shown in Fig. 2, it can be known that the present disclosure provides an audio processing method. By adopting the secondary compression processing method in the present disclosure, the compressed audio can occupy less space in a memory, the problems of excessive space occupation in the memory on the user side due to excessively large data packets as well as large packets and high memory consumption during C/C++ runtime due to the adoption of C/C++ for encapsulation in the prior art are solved, and meanwhile, audios can be collected and transmitted more conveniently in different terminals; due to the use of the preset instance method and the preset configuration file which are suitable for audio compression on different user sides, the present disclosure overcomes the problem in the prior art that it cannot be directly applied to the user sides during use; the present disclosure can achieve the visual display in the audio tracks by combining the audio tracks with the texts, thereby solving the problem of visual content analysis during enterprise data analysis and expanding value mining methods and efficiencies of audio assets.

Further, as an implementation of the above-mentioned method shown in Fig. 1, an embodiment of the present disclosure further provides one audio processing apparatus configured to implement the above-mentioned method shown in Fig. 1. The apparatus embodiment corresponds to the aforementioned method embodiment. For ease of reading, the details of the aforementioned method embodiment will not be repeated one by one in the apparatus embodiment. However, it should be clarified that the apparatus in this embodiment can correspondingly implement all the contents of the aforementioned method embodiment. As shown in Fig. 3, the apparatus includes:

- a first acquisition unit 31 configured to acquire an execution command;
- a second acquisition unit 32 configured to acquire, based on the execution command obtained from the first acquisition unit 31, configuration data corresponding to the execution command by using a preset instance method, wherein the execution command is obtained from a preset configuration file, and the preset configuration file at least includes an audio output mode, an audio encoding method, and an audio file format;
- a third acquisition unit 33 configured to acquire, based on the execution command obtained from the first acquisition unit 31, audio data by using a preset acquisition rule;
- a processing unit 34 configured to perform, based on the configuration data corresponding to the execution command, audio compression and format processing on the audio data by using a preset processing rule to obtain target audio data, wherein the configuration data corresponding to the execution command is obtained from the second acquisition unit 32, and the audio data is obtained from the third acquisition unit 33; and
- a visualization unit 35 configured to perform, based on the target audio data obtained from the processing unit 34, visualization processing by using a preset visualization rule to obtain visual display corresponding to the target audio data, wherein the visual display includes audio tracks and text annotations.

Further, as an implementation of the above-mentioned method shown in Fig. 2, an embodiment of the present disclosure further provides another audio processing apparatus configured to implement the above-mentioned method shown in Fig. 2. The apparatus embodiment corresponds to the aforementioned method embodiment. For ease of reading, the details of the aforementioned method embodiment will not be repeated one by one in this apparatus embodiment. However, it should be clarified that the apparatus in this embodiment can correspondingly implement all the contents of the aforementioned method embodiment. As shown in Fig. 4, the apparatus includes:

- a first acquisition unit 31 configured to acquire an execution command;
- a second acquisition unit 32 configured to acquire, based on the execution command obtained from the first acquisition unit 31, configuration data corresponding to the execution command from a preset configuration file by using a preset instance method, wherein the preset

configuration file is obtained from a creating unit 37, and the preset configuration file at least includes an audio output mode, an audio encoding method, and an audio file format;

a third acquisition unit 33 configured to acquire, based on the execution command obtained from the first acquisition unit 31, audio data by using a preset acquisition rule;

a processing unit 34 configured to perform, based on the configuration data corresponding to the execution command, audio compression and format processing on the audio data by using a preset processing rule to obtain target audio data, wherein the configuration data corresponding to the execution command is obtained from the second acquisition unit 32, and the audio data is obtained from the third acquisition unit 33; and

a visualization unit 35 configured to perform, based on the target audio data obtained from the processing unit 34, visualization processing by using a preset visualization rule to obtain visual display corresponding to the target audio data, wherein the visual display includes audio tracks and text annotations;

a calling unit 36 configured to call the preset instance method by using a preset call rule; and

a creating unit 37 configured to create the preset configuration file by using the preset instance method obtained from the calling unit 36.

Further, the configuration data at least includes a collection sampling rate and an output sampling rate; and the processing unit 34 includes:

a calculation module 341 configured to calculate a ratio of the collection sampling rate to the output sampling rate to obtain a compression ratio;

a first compression module 342 configured to compress, based on the compression ratio obtained from the calculation module 341, the audio data by using a preset compression rule to obtain primary compressed audio data; and

a second compression module 343 configured to perform secondary compression on the primary compressed audio data by using a method for encapsulating FFmpeg with Webassembly to obtain the target audio data, wherein the primary compressed audio data is obtained from the first compression module 342.

Further, the first compression module 342 includes:

a recognition submodule 3421 configured to recognize noisy data and null data in the audio data, and perform deletion processing to obtain to-be-compressed audio data; and

a compression submodule 3422 configured to compress, according to the compression ratio, the to-be-compressed audio data obtained from the recognition submodule 3421 to obtain the primary compressed audio data.

Further, the visualization unit 35 includes:

a recognition module 351 configured to recognize the target audio data by ASR to acquire a

corresponding relationship between an audio and a text;

a visualization module 352 configured to form, based on the corresponding relationship between the audio and the text as well as the target audio data, visual display in audio tracks by using Canvas and JavaScript, wherein the corresponding relationship between the audio and the text is obtained from the recognition module 351, and the visual display in the audio tracks includes the text; and

an analysis module 353 configured to perform, based on the visual display in the audio tracks obtained from the visualization module 352, audio analysis and annotation by using a preset movement command.

Further, the second acquisition unit 32 includes:

a parsing module 321 configured to parse the execution command to obtain a configuration item required for completing the execution command; and

an acquisition module 322 configured to acquire, based on the configuration item obtained from the parsing module 321, configuration data corresponding to the configuration item from the preset configuration file by using the preset instance method to obtain the configuration data corresponding to the execution command.

Further, the third acquisition unit 33 includes:

a determination module 331 configured to determine, based on the execution command, whether a user side authorizes the acquisition of the audio data;

an acquisition module 332 configured to acquire the audio data if the determination module 331 determines that the user side authorizes the acquisition of the audio data; and

a returning module 333 configured to return to a previous step to perform an action if the determination module 331 determines that the user side does not authorize the acquisition of the audio data.

Further, an embodiment of the present disclosure further provides a processor configured to run a program, wherein the program, when running, performs the above-mentioned audio processing method shown in Figs. 1-2.

Further, an embodiment of the present disclosure further provides a storage medium configured to store a computer program, wherein the computer program, when running, controls a device in which the storage medium is located to perform the above-mentioned audio processing method shown in Figs. 1-2.

In the above-mentioned embodiments, the description of each embodiment has its own emphasis. Parts that are not detailed in a certain embodiment refer to relevant descriptions in other embodiments.

It can be understood that relevant features in the above-mentioned methods and apparatuses can be

mutually referenced. In addition, terms such as “first” and “second” in the above-mentioned embodiments are intended to distinguish each embodiment, and do not represent the advantages or disadvantages of each embodiment.

The skilled in the relevant art can clearly understand that, for the sake of convenience and simplicity in description, specific working processes of the systems, apparatuses and units described above can be known with reference to the corresponding processes in the aforementioned method embodiments so as to be no longer repeated herein.

The algorithms and display provided herein are not inherently related to any specific computer, virtual system, or other device. Various general-purpose systems can also be used together with demonstrations based on this. According to the above description, a structure required for constructing such a system is obvious. Moreover, the present disclosure is not directed to any specific programming language. It should be understood that the contents of the present disclosure described herein can be achieved by adopting various programming languages, and the above description of specific languages is intended to disclose the best implementation of the present disclosure.

Moreover, the memory may include a volatile memory, a random access memory (RAM), and/or a non-volatile memory and other forms such as a read-only memory (ROM) or a flash RAM in computer-readable media, and the memory includes at least one storage chip.

The skilled in the art should understand that the embodiments of the present disclosure can be provided as methods, systems, or computer program products. Therefore, the present disclosure may take the form of a fully hardware embodiment, a fully software embodiment, or a software and hardware combined embodiment. Moreover, the present disclosure may take the form of a computer program product implemented on one or more computer-usable storage media (including, but not limited to a disk memory, a CD-ROM, and an optical memory) containing computer-usable program codes.

The present disclosure is described with reference to the flow diagrams and/or block diagrams of the methods, devices (systems) and computer program products according to the embodiments of the present disclosure. It should be understood that each process and/or block in the flow diagrams and/or block diagrams as well as combinations of processes and/or blocks in the flow diagrams and/or block diagrams can be implemented according to computer program instructions. These computer program instructions can be provided to a processor of a general-purpose computer, a special-purpose computer, an embedded processor or other programmable data processing devices to produce a machine, such that the instructions executed by the processor of the computer or other programmable data processing devices produce an apparatus for achieving functions specified in one or more processes in the flow diagrams and/or one or more blocks in the block

diagrams.

These computer program instructions can also be stored in a computer-readable memory that can guide a computer or other programmable data processing devices to operate in a specific manner, such that the instructions stored in the computer-readable memory produce a manufactured product including an instruction apparatus that achieves the functions specified in one or more processes in the flow diagrams and/or one or more blocks in the block diagrams.

These computer program instructions can also be loaded onto a computer or other programmable data processing devices, such that a series of operation steps are performed on the computer or other programmable devices to generate processing implemented by the computer. Thus, the instructions executed on the computer or other programmable devices provide steps for achieving the functions specified in one or more processes in the flow diagrams and/or one or more blocks in the block diagrams.

In a typical configuration, a computing device includes one or more processors (CPUs), an input/output interface, a network interface, and a memory.

The memory may include a volatile memory, a random access memory (RAM), and/or a non-volatile memory and other forms such as a read-only memory (ROM) or a flash RAM in computer-readable media. The memory is an example of the computer-readable media.

The computer-readable media include volatile, non-volatile, removable and non-removable media and can achieve information storage by using any method or technology. Information may be computer-readable instructions, data structures, modules of programs, or other data. Examples of storage media of a computer include, but are not limited to, a phase-change memory (PRAM), a static random access memory (SRAM), a dynamic random access memory (DRAM), other types of random access memories (RAM), a read-only memory (ROM), an electrically erasable programmable read-only memory (EEPROM), a flash memory or other memory technologies, a compact disc read-only memory (CD-ROM), a digital versatile disc (DVD) or other optical storage, a magnetic cartridge tape, magnetic tape and magnetic disk storage or other magnetic storage devices, or any other non-transmission media, and can be configured to store information that can be accessed by the computing device. According to the definition in this article, the computer-readable media do not include temporary computer-readable media (transitory media) such as modulated data signals and carriers.

It should be further noted that terms “include”, “including”, or any other variations thereof are intended to encompass non-exclusive inclusion, such that a process, method, commodity or device including a series of elements not only includes those elements, but also includes other elements not listed explicitly, or further includes inherent elements of such a process, method, commodity, or device. Without further limitations, the element limited by the statement “including a...” does

not exclude the existence of other identical elements in the process, method, commodity, or device that includes the element.

The skilled in the art should understand that the embodiments of the present disclosure can be provided as methods, systems, or computer program products. Therefore, the present disclosure may take the form of a fully hardware embodiment, a fully software embodiment, or a software and hardware combined embodiment. Moreover, the present disclosure may take the form of a computer program product implemented on one or more computer-usable storage media (including, but not limited to a disk memory, a CD-ROM, and an optical memory) containing computer-usable program codes.

The above embodiments are only embodiments of the present disclosure and are not intended to limit the present disclosure. For those skilled in the art, the present disclosure may have various changes and variations. Any modifications, equivalent substitutions, improvements, etc. made within the spirit and principles of the present disclosure shall fall within the scope of the claims of the present disclosure.

CLAIMS

1. An audio processing method, wherein the method comprises:
acquiring an execution command;
acquiring, based on the execution command, configuration data corresponding to the execution command from a preset configuration file by using a preset instance method, wherein the preset configuration file at least comprises an audio output mode, an audio encoding method, and an audio file format;
acquiring, based on the execution command, audio data by using a preset acquisition rule;
performing, based on the configuration data corresponding to the execution command, audio compression and format processing on the audio data by using a preset processing rule to obtain target audio data; and
performing, based on the target audio data, visualization processing by using a preset visualization rule to obtain visual display corresponding to the target audio data, wherein the visual display comprises audio tracks and text annotations.
2. The method of claim 1, wherein before acquiring an execution command, the method further comprises:
calling the preset instance method by using a preset calling rule; and
creating the preset configuration file by using the preset instance method.
3. The method of claim 1, wherein the configuration data at least comprises collection sampling rate and output sampling rate;
the performing, based on the configuration data corresponding to the execution command, audio compression and format processing on the audio data by using a preset processing rule to obtain target audio data comprises:
calculating a ratio of the collection sampling rate to the output sampling rate to obtain a compression ratio;
compressing, based on the compression ratio, the audio data by using a preset compression rule to obtain primary compressed audio data; and
performing secondary compression on the primary compressed audio data by using a method for encapsulating FFmpeg with Webassembly to obtain the target audio data.
4. The method of claim 3, wherein the compressing, based on the compression ratio, the audio data by using a preset compression rule to obtain primary compressed audio data comprises:
recognizing noisy data and null data in the audio data, and performing deletion processing to obtain to-be-compressed audio data; and
compressing the to-be-compressed audio data according to the compression ratio to obtain the

primary compressed audio data.

5. The method of any one of claims 1-4, wherein the performing, based on the target audio data, visualization processing by using a preset visualization rule to obtain corresponding visual display corresponding to the target audio data comprises:

recognizing the target audio data by ASR to acquire a corresponding relationship between an audio and a text;

forming, based on the corresponding relationship between the audio and the text as well as the target audio data, visual display in audio tracks by using Canvas and JavaScript, wherein the visual display in the audio tracks comprises the text; and

performing, based on the visual display in the audio tracks, audio analysis and annotation by using a preset movement command.

6. The method of claim 5, wherein the acquiring, based on the execution command, configuration data corresponding to the execution command from a preset configuration file by using a preset instance method comprises:

parsing the execution command to obtain a configuration item required for completing the execution command; and

acquiring, based on the configuration item, configuration data corresponding to the configuration item from the preset configuration file by using the preset instance method to obtain the configuration data corresponding to the execution command.

7. The method of claim 5, wherein the acquiring, based on the execution command, audio data by using a preset acquisition rule comprises:

determining, based on the execution command, whether a user side authorizes the acquisition of the audio data;

if so, acquiring the audio data; and

if not, returning to the previous step to perform the action.

8. An audio processing apparatus, comprising:

a first acquisition unit configured to acquire an execution command;

a second acquisition unit configured to acquire, based on the execution command, configuration data corresponding to the execution command from a preset configuration file by using a preset instance method, wherein the preset configuration file at least comprises an audio output mode, an audio encoding method, and an audio file format;

a third acquisition unit configured to acquire, based on the execution command, audio data by using a preset acquisition rule;

a processing unit configured to perform, based on the configuration data corresponding to the execution command, audio compression and format processing on the audio data by using a preset

processing rule to obtain target audio data; and

a visualization unit configured to perform, based on the target audio data, visualization processing by using a preset visualization rule to obtain visual display corresponding to the target audio data, wherein the visual display comprises audio tracks and text annotations.

9. A storage medium, comprising a stored program, wherein the program, when running, controls a device in which the storage medium is located to perform the audio processing method of any one of claims 1 to 7.

10. An electronic device, comprising a memory, a processor, and a computer program stored in the memory and capable of running on the processor, wherein the processor, when executing the program, implements the audio processing method of any one of claims 1 to 7.