LIVE STREAMING VOICE PLAYING METHOD AND APPARATUS, DEVICE, AND STORAGE MEDIUM

[0001] The present application claims priority to Chinese Patent Application No. 202510166685.5 filed with the China National Intellectual Property Administration (CNIPA) on Feb. 14, 2025, the disclosure of which is incorporated herein by reference in its entirety.

TECHNICAL FIELD

[0002] The present application relates to the technical field of audio processing, for example, a live streaming voice playing method and apparatus, a device and a storage medium.

BACKGROUND

[0003] With the booming development of the live streaming industry and the continuous maturity of the artificial intelligence (AI) technology, a new type of live streaming using a digital person instead of a live streamer has emerged. In the traditional live streaming, audio may be obtained by collecting a live streamer by using a recording device. However, when a digital person performs live streaming, audio is MP3 generated through the text-to-speech (TTS) technology. The audio required by a live streaming platform must be a continuous voice stream. If the discrete MP3 is directly pushed for streaming, the live streaming sound may be interrupted, causing poor user experience in hearing.

SUMMARY

[0004] The present application provides a live streaming voice playing method and apparatus, a device and a storage medium, implementing the interruption and insertion of audio content in real time, meeting the requirements of flexibly adjusting audio content in a live streaming scenario, solving the problem of discontinuous audio in the live streaming of a digital person, and making live streaming sound smoother and more natural.

[0005] According to an aspect of the present application, a live streaming voice playing method is provided. The method includes the steps below.

[0006] To-be-played audio data corresponding to a digital live streaming object is acquired in a live streaming process.

[0007] The to-be-played audio data is split to obtain a plurality of pieces of audio fragment data and a fragment index file corresponding to the plurality of pieces of audio fragment data.

[0008] The plurality of pieces of audio fragment data are pushed to a live streaming audio channel based on the fragment index file, and live streaming voice playing of the digital live streaming object is implemented based on an arrangement sequence of the plurality of pieces of audio fragment data.

[0009] According to an aspect of the present application, a live streaming voice playing apparatus is provided. The apparatus includes an audio data acquisition module, a fragment data obtaining module, and a live streaming voice playing module.

[0010] The audio data acquisition module is configured to acquire to-be-played audio data corresponding to a digital live streaming object in a live streaming process.

[0011] The fragment data obtaining module is configured to split the to-be-played audio data to obtain a plurality of pieces of audio fragment data and a fragment index file corresponding to the plurality of pieces of audio fragment data.

[0012] The live streaming voice playing module is configured to push the plurality of pieces of audio fragment data to a live streaming audio channel based on the fragment index file and implement live streaming voice playing of the digital live streaming object based on an arrangement sequence of the plurality of pieces of audio fragment data.

[0013] According to another aspect of the present application, an electronic device is provided. The electronic device includes at least one processor and a memory communicatively connected to the at least one processor.

[0014] The memory stores a computer program executable by the at least one processor to enable the at least one processor to perform the live streaming voice playing method of any

embodiment of the present application.

[0015] According to another aspect of the present application, a computer-readable storage medium is provided. The computer-readable storage medium has computer instructions which, when executed by a processor, cause the processor to perform the live streaming voice playing method of any embodiment of the present application.

BRIEF DESCRIPTION OF DRAWINGS

[0016] FIG. 1 is a flowchart of a live streaming voice playing method according to embodiment one of the present application.

[0017] FIG. 2 is a flowchart of a live streaming voice playing method according to embodiment two of the present application.

[0018] FIG. 3 is a structural view of a live streaming voice playing apparatus according to embodiment three of the present application.

[0019] FIG. 4 is a structural diagram of an electronic device for implementing a live streaming voice playing method according to an embodiment of the present application.

DETAILED DESCRIPTION

[0020] Terms such as "first" and "second" in the description, claims and drawings of the present application are used to distinguish between similar objects and are not necessarily used to describe a particular order or sequence. It is to be understood that data used in this manner are interchangeable where appropriate so that the embodiments of the present application described herein can be implemented in an order not illustrated or described herein. Additionally, terms "include", "have", and any variations thereof are intended to encompass a non-exclusive inclusion. For example, a process, method, system, product, or device that includes a series of steps or units not only includes the expressly listed steps or units but may also include other steps or units that are not expressly listed or are inherent to such process, method, product, or device.

[0021] Embodiment one

[0022] FIG. 1 is a flowchart of a live streaming voice playing method according to embodiment one of the present application. This embodiment can be applied to the case where live streaming voice configuration is performed for a digital live streamer. The method can be performed by a live streaming voice playing apparatus. The live streaming voice playing apparatus can be implemented in the form of hardware and/or software and can be configured in an electronic device. As shown in FIG. 1, the method includes the steps below.

[0023] In S101, to-be-played audio data corresponding to a digital live streaming object is acquired in a live streaming process.

[0024] The digital live streaming object may be a virtual object created by using the digital technology in the live streaming process, for example, a virtual character and a virtual doll. The digital live streaming object can simulate human expressions, actions, and sounds, thereby providing a realistic interactive experience for a user. The to-be-played audio data may be audio data for generating live streaming voice, for example, discrete MP3 audio segments.

[0025] In a live streaming scenario, a digital person may serve as a streamer or a guest and interact with the user in real time. In order to implement this function, the technical support behind the digital person is critical, including the audio stream generation service. This generation service can convert the discrete MP3 audio segments into a continuous voice stream and support real-time content insertion, thereby meeting audio requirements in the live streaming of the digital person.

[0026] The present application is applied to a server. The server provides the live streaming audio generation service for a caller (such as a live streaming platform or a digital person system). Before the to-be-played audio data corresponding to the digital live streaming object is acquired in the live streaming process, the server establishes a long connection with the caller so as to transmit and process the to-be-played audio data in real time. Exemplarily, the server may establish a long connection with the caller based on the WebSocket protocol. The server sends

an audio acquisition request to the caller. The caller sends to-be-played audio data to the server to implement the real-time transmission and processing of audio content, thereby improving the efficiency and real-time performance of data transmission.

[0027] In S102, the to-be-played audio data is split to obtain a plurality of pieces of audio fragment data and a fragment index file corresponding to the plurality of pieces of audio fragment data.

[0028] The pieces of audio fragment data may be audio fragments obtained by splitting to-be-played audio data. The fragment index file may refer to a text index file corresponding to the pieces of audio fragment data and may be used to arrange the plurality of pieces of audio fragment data in sequence so that a player can perform download and continuous play as required so as to form a complete audio stream.

[0029] Exemplarily, the pieces of audio fragment data are transport stream (TS) fragments. The fragment index file is a moving picture experts group audio layer 3 uniform resource locator, Unicode transformation format – 8-bit (M3U8) index file. It is to be noted that the TS fragments are multiple .ts files, each only a few seconds in length, into which the audio stream divided in the hypertext transfer protocol (HTTP) based live streaming (HLS) protocol. These files contain part of the content in the audio/video stream and are arranged in sequence so that the player can perform download and continuous play as required. The M3U8 file is an extended version of an M3U file and is specially optimized for a Unicode character set. It adds the support for UTF-8 character coding so as to better process multilingual content.

[0030] Exemplarily, with the adoption of the HLS technology, the server splits the to-be-played audio data into the pieces of audio fragment data with finer granularity and generates a fragment index file corresponding to the pieces of audio fragment data.

[0031] In S103, the plurality of pieces of audio fragment data are pushed to a live streaming audio channel based on the fragment index file, and the live streaming voice playing of the digital live streaming object is implemented based on an arrangement sequence of the plurality

of pieces of audio fragment data.

[0032] Exemplarily, the server pushes the pieces of audio fragment data to the live streaming audio channel of the caller through an FFmpeg tool according to the sequence in the fragment index file so that the caller implements the live streaming voice playing of the digital live streaming object according to the arrangement sequence of the plurality of pieces of audio fragment data.

[0033] Exemplarily, after the plurality of pieces of audio fragment data are pushed to the live streaming audio channel, the method further includes updating a total index file according to the fragment index file.

[0034] After a calling end is provided with the total index file, and after the plurality of pieces of audio fragment data are pushed to the live streaming audio channel, the fragment index file needs to be additionally written into the total index file so as to conveniently perform unified management and maintenance on each index file. For example, content in the index file can be easily added, deleted, or updated with no separate operation on each piece of audio fragment data, helping reduce management costs and improve maintenance efficiency. Moreover, the updating and upgrading of the audio stream become simpler and more convenient.

[0035] In some embodiments, the method further includes in response to the unplayed quantity of the pieces of audio fragment data being less than a preset quantity threshold, the to-be-played audio data continues to be acquired, and the live streaming voice playing of the digital live streaming object is implemented.

[0036] The preset quantity threshold may be set according to an actual situation or historical experience. For example, the preset quantity threshold may be 1.

[0037] That is, when the unplayed quantity of the pieces of audio fragment data is less than the preset quantity threshold, or when one sentence is left for playing the live streaming voice, the server continues to acquire the to-be-played audio data of the next stage from the caller in advance and repeats the preceding processing steps to implement the continuous playing of the

live streaming voice of the digital live streaming object.

[0038] In some embodiments, the method further includes in response to not acquiring the to-be-played audio data, blank fragment data is pushed to the live streaming audio channel.

[0039] That is, when the caller fails to provide new to-be-played audio data in time, the server pushes a blank audio fragment to the audio channel, guaranteeing uninterrupted audio streaming and improving the stability and user experience of live streaming.

[0040] In this embodiment of the present application, the to-be-played audio data corresponding to the digital live streaming object is acquired in the live streaming process; the to-be-played audio data is split to obtain the plurality of pieces of audio fragment data and the fragment index file corresponding to the plurality of pieces of audio fragment data; and the plurality of pieces of audio fragment data are pushed to the live streaming audio channel based on the fragment index file, and the live streaming voice playing of the digital live streaming object is implemented based on the arrangement sequence of the plurality of pieces of audio fragment data. This method solves the problem of discontinuous audio in the live streaming of the digital person, meets the requirements of flexibly adjusting audio content in the live streaming scenario, and makes the live streaming sound of the living streaming of the digital person smoother and more natural.

[0041] On the basis of the preceding embodiments, the method further includes the following: an audio insertion instruction is received, and to-be-inserted audio data is acquired according to the received audio insertion instruction; and insertion fragment data, which is obtained after the to-be-inserted audio data is split, is inserted into the live streaming audio channel to implement the insertion voice playing of the digital live streaming object.

[0042] The audio insertion instruction may refer to an instruction that requires audio to be inserted during a preset live streaming audio playing process. The to-be-inserted audio data may refer to the audio data of which the to-be-inserted audio is to be generated. The insertion fragment data may refer to audio fragments after the to-be-inserted audio data is split.

[0043] Exemplarily, after the audio insertion instruction is received, the server requests, from the caller, the acquisition of the to-be-inserted audio data corresponding to the audio insertion instruction. The server splits the to-be-inserted audio data into the insertion fragment data according to the requirements of the audio insertion instruction and generates an insertion index file. The insertion fragment data is inserted into the live streaming audio channel of the caller by referring to the arrangement sequence of the insertion fragment data in the insertion index file, implementing the insertion voice playing of the digital live object. Moreover, the total index file in the caller may also be updated by using the insertion index file.

[0044] In some embodiments, before the insertion fragment data is inserted into the live streaming audio channel, the method further includes the following: an insertion type corresponding to the audio insertion instruction is determined, where the insertion type includes interruptive insertion and temporary insertion; in response to determining that the insertion type is the interruptive insertion, the live streaming audio channel is cleared, and the insertion fragment data is inserted into the cleared live streaming audio channel; and in response to determining that the insertion type is the temporary insertion, the insertion fragment data is inserted into the live streaming audio channel based on an insertion time point corresponding to the audio insertion instruction.

[0045] Exemplarily, when the audio insertion instruction requires that the content currently played is inserted interruptedly, the caller discards the audio fragment data that is not completely played in the live streaming audio channel to clear the live streaming audio channel and adds the insertion index file to total indexes. When the audio insertion instruction requires that the content currently played is inserted temporarily, the insertion fragment data is inserted into the live streaming audio channel according to the insertion time point required by the audio insertion instruction, and the insertion index file is added to the total indexes to implement uninterrupted insertion. In embodiments of the present application, insertion is performed by supporting both the interruption of the content currently played and no interruption of the current content, meeting diversified content insertion requirements in the live streaming scenario and improving user experience.

[0046] Embodiment two

[0047] FIG. 2 is a flowchart of a live streaming voice playing method according to embodiment two of the present application. This embodiment is an optional embodiment on the basis of each preceding embodiment. As shown in FIG. 2, the method includes the steps below.

[0048] In a sequential streaming stage, a server establishes a WebSocket long connection with a caller.

[0049] The server sends an audio acquisition request to the caller, and the caller sends to-be-played audio data to the server.

[0050] The server splits the to-be-played audio data into audio fragment data and a fragment index file by using the HLS technology.

[0051] The server stores the audio fragment data into a to-be-played list and additionally writes the fragment index file into total indexes.

[0052] The server pushes the audio fragment data to an audio channel for playing by using an FFmpeg tool according to the sequence in the total index file.

[0053] When only one sentence remains unplayed in the current audio channel, the server requests the next to-be-played audio data from the caller in advance and repeats the preceding steps.

[0054] In an insertion stage, the caller sends an audio insertion instruction to the server.

[0055] The server splits to-be-inserted audio data into insertion fragment data and an insertion index file according to the requirements of the audio insertion instruction.

[0056] When the audio insertion instruction requires that the content currently played is interrupted, the caller discards the audio fragment data not completely played, inserts the insertion fragment data into the audio channel, and adds the insertion index file to the total indexes.

[0057] When an insertion message does not need the current content to be interrupted, the server adds the insertion index file to the total indexes. The server pushes the insertion fragment data to the audio channel according to the sequence in the updated total index file.

[0058] In an exception processing stage, when the caller fails to provide new to-be-played audio data in time, the server pushes a blank audio fragment to the audio channel to prevent the streaming from being interrupted.

[0059] Embodiment three

[0060] FIG. 3 is a structural view of a live streaming voice playing apparatus according to embodiment three of the present application. As shown in FIG. 3, the apparatus includes an audio data acquisition module 301, a fragment data obtaining module 302, and a live streaming voice playing module 303.

[0061] The audio data acquisition module 301 is configured to acquire to-be-played audio data corresponding to a digital live streaming object in a live streaming process.

[0062] The fragment data obtaining module 302 is configured to split the to-be-played audio data to obtain a plurality of pieces of audio fragment data and a fragment index file corresponding to the plurality of pieces of audio fragment data.

[0063] The live streaming voice playing module 303 is configured to push the plurality of pieces of audio fragment data to a live streaming audio channel based on the fragment index file and implement the live streaming voice playing of the digital live streaming object based on an arrangement sequence of the plurality of pieces of audio fragment data.

[0064] In this embodiment of the present application, the to-be-played audio data corresponding to the digital live streaming object is acquired in the live streaming process; the to-be-played audio data is split to obtain the plurality of pieces of audio fragment data and the fragment index file corresponding to the plurality of pieces of audio fragment data; and the plurality of pieces of audio fragment data are pushed to the live streaming audio channel based on the fragment index file, and the live streaming voice playing of the digital live streaming

object is implemented based on the arrangement sequence of the plurality of pieces of audio fragment data. This method solves the problem of discontinuous audio in the live streaming of the digital person, meets the requirements of flexibly adjusting audio content in the live streaming scenario, and makes the live streaming sound of the living streaming of the digital person smoother and more natural.

[0065] In some embodiments, the audio data acquisition module 301 is further configured to perform the step below.

[0066] In response to the unplayed quantity of the pieces of audio fragment data being less than a preset quantity threshold, the to-be-played audio data continues to be acquired, and the live streaming voice playing of the digital live streaming object is implemented.

[0067] In some embodiments, the live streaming voice playing module 303 is further configured to perform the step below.

[0068] In response to not acquiring the to-be-played audio data, blank fragment data is pushed to the live streaming audio channel.

[0069] In some embodiments, the apparatus further includes a live streaming voice insertion module.

[0070] The live streaming voice insertion module is configured to perform the steps below.

[0071] An audio insertion instruction is received, and to-be-inserted audio data is acquired according to the received audio insertion instruction.

[0072] Insertion fragment data, which is obtained after the to-be-inserted audio data is split, is inserted into the live streaming audio channel to implement the insertion voice playing of the digital live streaming object.

[0073] In some embodiments, the apparatus further includes an insertion type determination module.

[0074] The insertion type determination module is configured to perform the steps below.

[0075] Before the insertion fragment data is inserted into the live streaming audio channel, an insertion type corresponding to the audio insertion instruction is determined, where the insertion type includes interruptive insertion and temporary insertion.

[0076] In response to determining that the insertion type is the interruptive insertion, the live streaming audio channel is cleared, and the insertion fragment data is inserted into the cleared live streaming audio channel.

[0077] In response to determining that the insertion type is the temporary insertion, the insertion fragment data is inserted into the live streaming audio channel based on an insertion time point corresponding to the audio insertion instruction.

[0078] In some embodiments, the apparatus further includes an index file updating module.

[0079] The index file updating module is configured to perform the step below.

[0080] After the plurality of pieces of audio fragment data are pushed to the live streaming audio channel, a total index file is updated according to the fragment index file.

[0081] In some embodiments, a piece of audio fragment data is a TS fragment, and the fragment index file is an M3U8 file.

[0082] The live streaming voice playing apparatus provided in embodiments of the present application may execute the live streaming voice playing method of any one of the embodiments of the present application and has functional modules and beneficial effects corresponding to the executed method.

[0083] Embodiment four

[0084] FIG. 4 is a diagram illustrating the structure of an electronic device 10 for implementing embodiments of this application. The electronic device is intended to represent various forms of digital computers, for example, a laptop computer, a desktop computer, a worktable, a personal digital assistant, a server, a blade server, a mainframe computer, and another applicable computer. The electronic device may also represent various forms of mobile

apparatuses, for example, a personal digital assistant, a cellphone, a smartphone, a wearable device (such as a helmet, glasses, or a watch), and another similar computing apparatus. Herein the shown components, the connections and relationships between these components, and the functions of these components are illustrative.

[0085] As shown in FIG. 4, the electronic device 10 includes at least one processor 11 and a memory communicatively connected to the at least one processor 11, such as a read-only memory (ROM) 12 and a random-access memory (RAM) 13. The memory stores a computer program executable by the at least one processor. The at least one processor 11 can perform various appropriate actions and processing according to a computer program stored in the read-only memory (ROM) 12 or a computer program loaded into the random-access memory (RAM) 13 from a storage unit 18. Various programs and data required for the operation of the electronic device 10 may also be stored in the RAM 13. The at least one processor 11, the ROM 12, and the RAM 13 are connected to each other through a bus 14. An input/output (I/O) interface 15 is also connected to the bus 14.

[0086] Multiple components in the electronic device 10 are connected to the I/O interface 15. The multiple components include an input unit 16 such as a keyboard or a mouse, an output unit 17 such as various types of display or speaker, the storage unit 18 such as a magnetic disk or an optical disk, and a communication unit 19 such as a network card, a modem, or a wireless communication transceiver. The communication unit 19 allows the electronic device 10 to exchange information/data with other devices over a computer network such as the Internet and/or various telecommunications networks.

[0087] The at least one processor 11 may be various general-purpose and/or special-purpose processing components having processing and computing capabilities. Examples of the processor 11 include, but are not limited to, a central processing unit (CPU), a graphics processing unit (GPU), a special-purpose AI computing chip, a processor executing machine learning models and algorithms, a digital signal processor (DSP), and any appropriate processor, controller, or microcontroller. The processor 11 performs the various methods and processing

described above, such as the live streaming voice playing method.

[0088] In some examples, the live streaming voice playing method may be implemented as computer programs tangibly contained in a computer-readable storage medium such as the storage unit 18. In some embodiments, part or all of the computer programs may be loaded and/or installed onto the electronic device 10 via the ROM 12 and/or the communication unit 19. When the computer programs are loaded to the RAM 13 and executed by the processor 11, one or more steps of the preceding live streaming voice playing method may be performed. In other embodiments, the processor 11 may be configured, in any other suitable manner (for example, by means of firmware), to perform the live streaming voice playing method.

[0089] The various embodiments of the systems and techniques described herein may be implemented in digital electronic circuitry, integrated circuitry, a field-programmable gate array (FPGA), an application-specific integrated circuit (ASIC), an application specific standard product (ASSP), a system on a chip (SoC), a complex programmable logic device (CPLD), computer hardware, firmware, software and/or a combination thereof. The various embodiments may include implementations in one or more computer programs. The one or more computer programs are executable and/or interpretable on a programmable system including at least one programmable processor. A programmable processor may be a special-purpose or general-purpose programmable processor for receiving data and instructions from a memory system, at least one input apparatus, and at least one output apparatus and transmitting data and instructions to the memory system, the at least one input apparatus, and the at least one output apparatus.

[0090] Computer programs for implementation of the methods of the present application may be written in one programming language or any combination of multiple programming languages. The computer programs may be provided for a processor of a general-purpose computer, a special-purpose computer, or another programmable data processing apparatus to enable functions/operations specified in a flowchart and/or a block diagram to be implemented when the computer programs are executed by a processor. The computer programs may be

executed entirely on a machine, partly on a machine, as a stand-alone software package, partly on a machine and partly on a remote machine, or entirely on a remote machine or a server.

[0091] In the context of the present application, the computer-readable storage medium may be a tangible medium that may include or store a computer program used by or in conjunction with an instruction execution system, apparatus, or device. The computer-readable storage medium may include an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, or device or any appropriate combination thereof. Alternatively, the computer-readable storage medium may be a machine-readable signal medium. The machine-readable storage medium may include an electrical connection based on one or more wires, a portable computer disk, a hard disk, a RAM, a ROM, an erasable programmable read-only memory (EPROM), a flash memory, an optical fiber, a portable compact disc read-only memory (CD-ROM), an optical storage device, a magnetic storage device, or any appropriate combination thereof.

[0092] In order that interaction with a user is provided, the systems and techniques described herein may be implemented in the electronic device. The electronic device has a display apparatus (for example, a cathode-ray tube (CRT) or a liquid-crystal display (LCD) monitor) for displaying information to the user; and a keyboard and a pointing apparatus (for example, a mouse or a trackball) through which the user can provide input for the electronic device. Other types of apparatuses may also be used for providing interaction with the user. For example, feedback provided for the user may be sensory feedback in any form (for example, visual feedback, auditory feedback, or tactile feedback); and input from the user may be received in any form (including acoustic input, voice input, or tactile input).

[0093] The systems and techniques described herein may be implemented in a computing system including a back-end component (for example, a data server), a computing system including a middleware component (for example, an application server), a computing system including a front-end component (for example, a client computer having a graphical user interface or a web browser through which the user can interact with embodiments of the

systems and techniques described herein), or a computing system including any combination of such back-end, middleware, or front-end components. Components of a system may be interconnected by any form or medium of digital data communication (for example, a communication network). Examples of the communication network include a local area network (LAN), a wide area network (WAN), a blockchain network, and the Internet.

[0094] The computing system may include clients and servers. A client and a server are generally remote from each other and typically interact through a communication network. The relationship between the client and the server arises by virtue of computer programs running on respective computers and having a client-server relationship to each other. The server may be a cloud server, also referred to as a cloud computing server or a cloud host. As a host product in a cloud computing service system, the server solves the defects of difficult management and weak service scalability in conventional physical host and virtual private server (VPS) services.

What is claimed is:

1. A live streaming voice playing method, comprising:

acquiring to-be-played audio data corresponding to a digital live streaming object in a live streaming process;

splitting the to-be-played audio data to obtain a plurality of pieces of audio fragment data and a fragment index file corresponding to the plurality of pieces of audio fragment data; and

pushing the plurality of pieces of audio fragment data to a live streaming audio channel based on the fragment index file and implementing live streaming voice playing of the digital live streaming object based on an arrangement sequence of the plurality of pieces of audio fragment data.

2. The live streaming voice playing method according to claim 1, further comprising:

in response to an unplayed quantity of the plurality of pieces of audio fragment data being less than a preset quantity threshold, continuing to acquire the to-be-played audio data and implementing the live streaming voice playing of the digital live streaming object.

3. The live streaming voice playing method according to claim 2, further comprising:

in response to not acquiring the to-be-played audio data, pushing blank fragment data to the live streaming audio channel.

4. The live streaming voice playing method according to claim 1, further comprising:

receiving an audio insertion instruction and acquiring to-be-inserted audio data according to the received audio insertion instruction; and

inserting insertion fragment data, which is obtained after the to-be-inserted audio data is split, into the live streaming audio channel to implement insertion voice playing of the digital live streaming object.

5. The live streaming voice playing method according to claim 4, before the insertion fragment data is inserted into the live streaming audio channel, further comprising:

determining an insertion type corresponding to the audio insertion instruction, wherein the insertion type comprises interruptive insertion and temporary insertion;

in response to determining that the insertion type is the interruptive insertion, clearing the live streaming audio channel and inserting the insertion fragment data into the cleared live streaming audio channel; and

in response to determining that the insertion type is the temporary insertion, inserting the

insertion fragment data into the live streaming audio channel based on an insertion time point corresponding to the audio insertion instruction.

6. The live streaming voice playing method according to claim 1, after the plurality of pieces of audio fragment data are pushed to the live streaming audio channel, further comprising:

updating a total index file according to the fragment index file.

7. The live streaming voice playing method according to any one of claims 1 to 6, wherein one of the plurality of pieces of audio fragment data is a transport stream (TS) fragment, and the fragment index file is a moving picture experts group audio layer 3 uniform resource locator, Unicode transformation format – 8-bit (M3U8)file.

8. A live streaming voice playing apparatus, comprising:

an audio data acquisition module configured to acquire to-be-played audio data corresponding to a digital live streaming object in a live streaming process;

a fragment data obtaining module configured to split the to-be-played audio data to obtain a plurality of pieces of audio fragment data and a fragment index file corresponding to the plurality of pieces of audio fragment data; and

a live streaming voice playing module configured to push the plurality of pieces of audio fragment data to a live streaming audio channel based on the fragment index file and implement live streaming voice playing of the digital live streaming object based on an arrangement sequence of the plurality of pieces of audio fragment data.

9. An electronic device, comprising:

at least one processor; and

a memory communicatively connected to the at least one processor,

wherein the memory is configured to store a computer program executable by the at least one processor to cause the at least one processor to perform the live streaming voice playing method according to any one of claims 1 to 7.

10. A computer-readable storage medium for storing computer instructions configured to, when executed by a processor, cause the processor to perform the live streaming voice playing method according to any one of claims 1 to 7.