

CERTIFICATE OF TRANSLATION ACCURACY

May 14, 2026

I, the undersigned, Steve Ni, hereby certify:

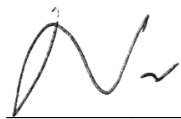
I am a professional translator. I work as an independent contractor. I was contracted by Divergent Language Solutions, LLC to provide document translation from Chinese into English for this matter.

I am fluent in both Chinese and English. My qualifications include 20 years of full-time professional translation experience from Chinese into English.

I am familiar and competent with both the Chinese and English languages. I have received and translated from Chinese into English the document *Chinese Patent Application No. CN106060628 A* and hereby attest that the English translation is a complete and accurate rendition of the Chinese original to the best of my knowledge and belief.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on the information and belief are believed to be true; I declare under penalty of perjury that the foregoing is true and correct, and understand that willful false statements and the like are punishable by fine or imprisonment, or both pursuant to 18 U.S.C. 1001.

Executed on May 14, 2026.



Steve Ni



Divergent Language Solutions, LLC

London | New York | San Francisco | Honolulu

600 California Street, 11th Floor | San Francisco, CA 94108 | p 415.400.4538 | f 415.525.4313

divergent@divergentls.com | www.divergentls.com



(21) Application number: 201610491470.1
(22) Date of application: 06.28.2016.
(71) Applicant: Baofeng Group Co., Ltd.
Address: 6th Floor, Shouxiang Technology Building,
51 Xueyuan Road, Haidian District,
Beijing City, China; Postal code: 100083
(72) Inventors: Wu Chao, Zhu Chunming, Zhu Yandong,
Zhang Pengyu
(74) Patent agency: Beijing Shengrui Zhijie Intellectual
Property Agency (Special General
Partnership); 11603

H04N 21/4402 (2011.01)

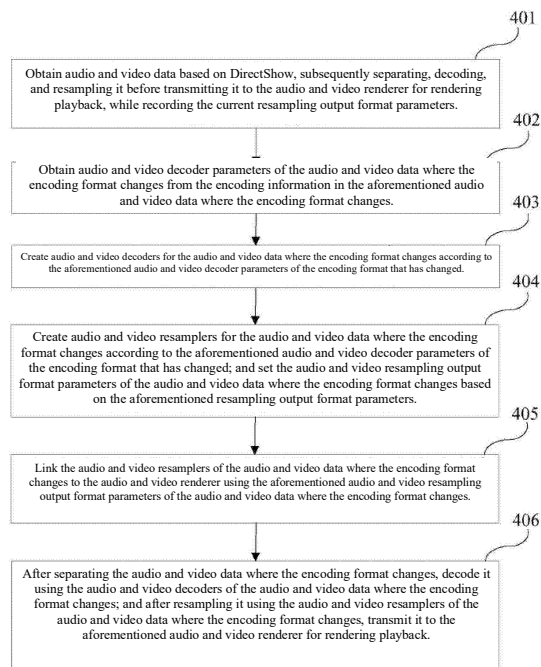
Agent: Yu Miao
(51) Int. Cl.
H04N 21/426 (2011.01)
H04N 21/434 (2011.01)
H04N 21/438 (2011.01)
H04N 21/4385 (2011.01)

Scope of Patent Claims: 3 pages. Detailed Explanation: 11 pages. Figures attached: 10 pages.

(54) Title of Invention
Method and System Supporting Variable Encoding
Based on DirectShow

(57) Abstract

This invention discloses a method supporting variable encoding based on DirectShow, comprising the following steps: obtaining audio and video data based on DirectShow, subsequently separating, decoding, and resampling it for rendering playback, while recording the current resampling output format parameters; obtaining the audio and video decoder parameters of the audio and video data where the encoding format changes; creating audio and video decoders and audio and video resamplers for the audio and video data where the encoding format changes; setting the audio and video resampling output format parameters for the audio and video data where the encoding format changes according to the resampling output format parameters and linking them to the audio and video renderer; separating, decoding, and resampling the audio and video data where the encoding format changes and transmitting it to the audio and video renderer for rendering playback. This invention does not require disconnecting and updating the existing renderer, ensuring the continuity of audio data playback, and will not cause interruptions while the user is watching audio and video.



1. A method supporting variable encoding based on DirectShow, comprising the following steps:
 - obtaining audio and video data based on DirectShow, subsequently separating, decoding, and resampling it before transmitting it to the audio and video renderer for rendering playback, while recording the current resampling output format parameters; wherein the aforementioned audio and video data includes: encoding information, video data, and audio data;
 - obtaining the audio and video decoder parameters of the audio and video data where the encoding format changes from the encoding information in the aforementioned audio and video data where the encoding format changes;
 - creating audio and video decoders for the audio and video data where the encoding format changes according to the aforementioned audio and video decoder parameters of the encoding format that has changed;
 - creating audio and video resamplers for the audio and video data where the encoding format changes according to the aforementioned audio and video decoder parameters of the encoding format that has changed;
 - setting the audio and video resampling output format parameters of the audio and video data where the encoding format changes based on the aforementioned resampling output format parameters;
 - linking the audio and video resamplers of the audio and video data where the encoding format changes to the audio and video renderer using the aforementioned audio and video resampling output format parameters of the audio and video data where the encoding format changes;
 - after separating the audio and video data where the encoding format changes, decoding it using the audio and video decoders of the audio and video data where the encoding format changes; and after resampling it using the audio and video resamplers of the audio and video data where the encoding format changes, transmitting it to the aforementioned audio and video renderer for rendering playback.
2. A method supporting variable encoding based on DirectShow, according to Claim 1, characterized by that the aforementioned method of obtaining audio and video data based on DirectShow comprises sequentially separating, decoding, resampling, and rendering playback while recording the current output format parameters after resampling, and comprises the following steps:
 - after separating the audio and video data obtained based on DirectShow, obtaining audio and video track information from the encoding information of the aforementioned audio and video data, and retrieving the audio and video decoder parameters related to the aforementioned audio and video data from the aforementioned audio and video track information;
 - creating audio and video decoders according to the aforementioned audio and video decoder parameters, and setting the output format parameters of the aforementioned audio and video decoders;
 - creating resamplers according to the aforementioned audio and video decoder parameters, and recording the current resampling output format parameters.
3. A method supporting variable encoding based on DirectShow, according to Claim 1, characterized by further comprising the following steps: obtaining the audio and video decoder parameters of the audio and video data where the encoding format has not changed from the encoding information in the aforementioned audio and video data where the encoding format has not changed;
 - after separating the audio and video data where the encoding format has not changed, transmitting it for decoding using the audio and video decoder parameters linked to the audio and video data where the encoding format has not changed;
 - and resampling with the resamplers of the audio and video data where the encoding format has not changed before transmitting it to the aforementioned audio and video renderer for rendering playback.
4. A method supporting variable encoding based on DirectShow, according to Claim 1, characterized by that the aforementioned method sets the audio and video resampling output format parameters of the audio and video data where the encoding format has changed according to the aforementioned resampling output format parameters, and further comprises the following step:
 - setting the audio and video resampling output format parameters of the audio and video data where the encoding format has changed to be consistent with the aforementioned resampling output format parameters.

5. A method supporting variable encoding based on DirectShow, according to Claim 1, characterized by that the decoders for the audio and video data where the encoding format has changed and the resamplers for the audio and video data where the encoding format has changed are provided by third-party decoders that are not based on DirectShow.

6. A system supporting variable encoding based on DirectShow, characterized by comprising: a player, a detection module, a decoder creation module, a resampling creation module, and a linking module, wherein:

the aforementioned player is used to obtain audio and video data based on DirectShow, sequentially separating, decoding, and resampling it before transmitting it to the audio and video renderer for rendering playback, while recording the current resampling output format parameters; wherein the aforementioned audio and video data includes: encoding information, video data, and audio data;

the aforementioned obtaining module is used to retrieve the audio and video decoder parameters of the audio and video data where the encoding format has changed from the encoding information in the aforementioned audio and video data where the encoding format has changed;

the aforementioned decoder creation module is used to create audio and video decoders for the audio and video data where the encoding format has changed according to the aforementioned audio and video decoder parameters;

the aforementioned resampling creation module is used to create audio and video resamplers for the audio and video data where the encoding format has changed according to the aforementioned audio and video decoder parameters after the encoding format has changed;

setting the audio and video resampling output format parameters of the audio and video data where the encoding format changes based on the aforementioned resampling output format parameters;

the aforementioned linking module is used to link the audio and video resamplers of the audio and video data where the encoding format has changed to the aforementioned audio and video renderer using the aforementioned audio and video resampling output format parameters;

the aforementioned player is also used to decode the audio and video data where the encoding format has changed using the audio and video decoders of the audio and video data after separating it; and after resampling it using the audio and video resamplers for the audio and video data where the encoding format has changed, transmitting it to the aforementioned audio and video renderer for rendering playback.

7. A system supporting variable encoding based on DirectShow, according to Claim 6, characterized by that the aforementioned player is also used to:

after separating the audio and video data obtained based on DirectShow, obtaining audio and video track information from the encoding information of the aforementioned audio and video data, and retrieving the audio and video decoder parameters related to the aforementioned audio and video data from the aforementioned audio and video track information;

creating audio and video decoders according to the aforementioned audio and video decoder parameters, and setting the output format parameters of the aforementioned audio and video decoders;

create resamplers according to the aforementioned audio and video decoder parameters, and record the current output format parameters of the resampling.

8. A system supporting variable encoding based on DirectShow, according to Claim 6, characterized by that the aforementioned obtaining module is also used to: retrieve the audio and video decoder parameters of the audio and video data where the encoding format has not changed from the encoding information in the aforementioned audio and video data where the encoding information format has not changed;

the aforementioned decoder creation module is also used to decode the audio and video data where the encoding format has not changed after separating it and transmitting it for decoding using the audio and video decoder parameters linked to the audio and video data where the encoding format has not changed;

the aforementioned resampling creation module is also used to perform resampling on the decoded audio and video data where the encoding format has not changed using the resamplers of the audio and video data where the encoding format has not changed;

the aforementioned player is also used to transmit the audio and video data of the resampled audio and video data where the encoding format has not changed to the aforementioned audio and video renderer for rendering playback.

9. A system supporting variable encoding based on DirectShow, according to Claim 6, characterized by that the aforementioned resampling creation module is also used for:

setting the audio and video resampling output format parameters of the audio and video data where the encoding format has changed to be consistent with the aforementioned resampling output format parameters.

10. A system supporting variable encoding based on DirectShow, according to Claim 6, characterized by that the decoders for the audio and video data where the encoding format has changed and the resamplers for the audio and video data where the encoding format has changed are provided by third-party decoders that are not based on DirectShow.

Method and System for Supporting Variable Encoding Based on DirectShow

Technical Field

[0001] This invention relates to the technical field of video playback, more specifically, it relates to a method and system for supporting variable encoding based on DirectShow.

Background Technology

[0002] When playing existing format video files based on DirectShow, once a change occurs (such as the addition of video encoding formats; for example, after MP4 adds support for H.265, it will introduce the "hevc" structure), it is necessary to upgrade the demultiplexer; otherwise, it cannot recognize or play the video.

[0003] DirectShow is a new generation development package for streaming media processing based on COM (Component Object Model), launched by Microsoft on the basis of ActiveMovie and Video for Windows, released together with the DirectX development package. DirectShow uses a Filter Graph model to manage the entire data stream processing process. When DirectShow operates, it links the filters (the functional module units of DirectShow) via MediaType and transmits data through MediaSample.]

[0004] The typical operation mode of DirectShow is: developers create a Filter Graph and can selectively add corresponding conversion schemes to the Filter Graph before playing a file. Using DirectShow, it is very convenient to capture data from capture cards that support the WDM driver model, perform corresponding post-processing, and even store it in a file. This makes access to multimedia data in multimedia database management systems (MDBMS) much easier. DirectShow widely supports various media formats, including Asf, Mpeg, Avi, Dv, Mp3, Wave, etc., providing strong support for the capture and playback of multimedia streams. Therefore, DirectShow, as a widely adopted audio and video playback solution on the Windows operating system, has found widespread application in the field of video playback. DirectShow defines a series of standard interfaces that facilitate functional expansion, such as the required demultiplexers, decoders, etc.

[0005] Currently, when performing online video playback based on DirectShow, especially during live broadcasts, changes in video types may occur, leading to changes in the parameters encoded by DirectShow. For example, during a live broadcast, when inserting advertisements, there may be changes in audio and video formats, changes in encoder sampling rates, and changes in audio and video resolutions. For instance, when playing audio and video based on DirectShow, if encoding parameters change (such as the audio format changing from mp3 to aac, the sampling rate changing from 48k to 44.1k, or the video resolution changing from 1920*1080 to 720*576, etc.), it is clearly not feasible to decode and play using the original decoder. Fig. 1 shows a schematic diagram of the system structure for audio and video decoding and playback based on DirectShow in traditional solutions. The method for audio and video playback based on DirectShow in traditional solutions is: first, separating audio and video in the DirectShow demultiplexer 101; then, decoding video and audio respectively in video decoder 102 and audio decoder 103; after decoding, the processed video and audio are rendered and played back in video renderer 104 and audio renderer 105, respectively. The steps of the traditional solution for variable encoding playback based on DirectShow are shown in Fig. 2:

[0006] Step 201: When a change in encoding parameters is detected during audio and video playback based on DirectShow, stop the current audio and video playback based on DirectShow;

[0007] Step 202: Disconnect the original audio and video decoder links, and disconnect the original audio and video renderer links;

[0008] Step 203: Release the original audio and video decoders and the original audio and video renderers;

[0009] Step 204: Create new audio and video decoders and new renderers according to the changed audio and video formats.

[0010] Step 205: After linking the new audio and video decoders and new audio and video renderers, begin decoding and playback.

[0011] As can be seen from Fig. 2, the traditional solution for variable encoding audio and video decoding and playback based on DirectShow requires re-linking of decoders and renderers before continuing to play the new audio and video. During the process of re-linking the decoders and renderers, there may be pauses, black screens, or jumps, which affect the user's viewing experience; furthermore, the time required to disconnect the original decoders and renderers and link the new decoders and renderers increases the waiting time for users, which is detrimental to their viewing experience.

[0012] Therefore, providing a method for fast and smooth variable encoding playback of audio and video based on DirectShow is an urgent problem to be solved in this field.

Content of Invention

[0013] In view of this, this invention provides a method and system for supporting variable encoding based on DirectShow, addressing the issues in the existing technologies where variable encoding audio and video decoding and playback based on DirectShow may cause pauses, black screens, or jumps.

[0014] In order to solve the above technical problems, this invention proposes a method for supporting variable encoding based on DirectShow, including:

[0015] Acquiring audio and video data based on DirectShow, sequentially separating, decoding, and resampling it before transmitting to audio and video renderers for playback, while recording the current resampling output format parameters; wherein, the aforementioned audio and video data includes: encoding information, video data, and audio data;

[0016] Obtaining the audio and video decoder parameters of the audio and video data from the encoding information where changes in encoding formats occur;

[0017] Creating audio and video decoders for audio and video data with changed encoding formats based on the aforementioned audio and video decoder parameters;

[0018] Creating audio and video resamplers for audio and video data with changed encoding formats based on the aforementioned audio and video decoder parameters;

[0019] Setting the output format parameters of the audio and video resampler for the audio and video data with changed encoding formats according to the aforementioned resampling output format parameters;

[0020] Linking the audio and video resamplers for the audio and video data with changed encoding formats to the audio and video renderers using the aforementioned resampling output format parameters;

[0021] After separating the audio and video data with changed encoding formats, decoding them using the audio and video decoders for the audio and video data with changed encoding formats; and after resampling using the audio and video resamplers for the audio and video data with changed encoding formats, transmitting to the audio and video renderers for rendering playback.

[0022] Furthermore, wherein, acquiring audio and video data based on DirectShow involves sequentially separating, decoding, resampling, and rendering for playback, while recording the current output format parameters after resampling, including:

[0023] After separating the audio and video data based on DirectShow, obtaining the audio and video track information from the encoding information of the aforementioned audio and video data, and acquiring the relevant audio and video decoder parameters from the aforementioned audio and video track information;

[0024] Creating audio and video decoders based on the aforementioned audio and video decoder parameters, and setting the output format parameters for the aforementioned audio and video decoders;

[0025] Creating resamplers based on the aforementioned audio and video decoder parameters, and recording the current resampling output format parameters.

[0026] Furthermore, wherein, the method also includes:

[0027] Obtaining the audio and video decoder parameters of the audio and video data where encoding formats have not changed from the encoding information in the aforementioned audio and video data;

[0028] After separating the audio and video data where encoding formats have not changed, transmitting it to decode using the audio and video decoder parameters linked from the aforementioned audio and video data where encoding formats have not changed;

[0029] And after resampling using the resamplers for the audio and video data where encoding formats have not changed, transmitting to the aforementioned audio and video renderer for rendering playback.

[0030] Furthermore, wherein, the aforementioned setting of the output format parameters for the audio and video resampler for the audio and video data with changed encoding formats according to the aforementioned resampling output format parameters also includes:

[0031] Setting the output format parameters of the audio and video resampler for the audio and video data with changed encoding formats to be consistent with the aforementioned resampling output format parameters.

[0032] Furthermore, wherein, the decoders for the audio and video data with changed encoding formats and the resamplers for the audio and video data with changed encoding formats are provided by third-party decoders that are not DirectShow.

[0033] On the other hand, this invention also provides a system for supporting variable encoding based on DirectShow, the aforementioned system includes: a player, detection module, decoder creation module, resampler creation module, and linking module, wherein,

[0034] the aforementioned player is used to acquire audio and video data based on DirectShow, sequentially separating, decoding, and resampling it before transmitting to the audio and video renderers for rendering playback, while recording the current resampling output format parameters; wherein, the aforementioned audio and video data includes: encoding information, video data, and audio data;

[0035] the aforementioned acquisition module is used to obtain the audio and video decoder parameters of the audio and video data with changed encoding formats from the encoding information;

[0036] the aforementioned decoder creation module is used to create the audio and video decoders for the audio and video data with changed encoding formats based on the aforementioned audio and video decoder parameters;

[0037] the aforementioned resampler creation module is used to create the audio and video resamplers for the audio and video data with changed encoding formats based on the aforementioned audio and video decoder parameters;

[0038] And set the output format parameters for the audio and video resamplers for the audio and video data with changed encoding formats according to the aforementioned resampling output format parameters;

[0039] the aforementioned linking module is used to link the audio and video resamplers for the audio and video data with changed encoding formats to the aforementioned audio and video renderers using the aforementioned output format parameters;

[0040] the aforementioned player is also used to separate the audio and video data with changed encoding formats and decode using the audio and video decoders for the audio and video data with changed encoding formats; and after resampling using the audio and video resamplers for the audio and video data with changed encoding formats, transmit to the aforementioned audio and video renderer for rendering playback.

[0041] Furthermore, wherein, the aforementioned player is also used for:

[0042] After acquiring audio and video data based on DirectShow and separating it, obtaining the audio and video track information from the encoding information of the aforementioned audio and video data, and acquiring the relevant audio and video decoder parameters from the aforementioned audio and video track information;

[0043] Creating audio and video decoders based on the aforementioned audio and video decoder parameters, and

setting the output format parameters for the aforementioned audio and video decoders.

[0044] Creating a resampler according to the aforementioned audio and video decoder parameters, and recording the current output format parameters of the resampling.

[0045] Furthermore, wherein, the aforementioned acquisition module is also used to: obtain the audio and video decoder parameters of the audio and video data where encoding formats have not changed from the unchanged encoding information format in the aforementioned obtained audio and video data;

[0046] The aforementioned decoder creation module is also used to: after separating the audio and video data where encoding formats have not changed, transmit it to decode the audio and video data where encoding formats have not changed using the audio and video decoder parameters linked from the aforementioned audio and video data where encoding formats have not changed;

[0047] The aforementioned resampler creation module is also used to: use the resamplers for the audio and video data where encoding formats have not changed to resample the decoded audio and video data where encoding formats have not changed;

[0048] The aforementioned player is also used to: transmit the audio and video data of the resampled audio and video data where encoding formats have not changed to the aforementioned audio and video renderer for rendering playback.

[0049] Furthermore, wherein, the aforementioned resampler creation module is also used to:

[0050] Set the output format parameters for the audio and video resampling of the audio and video data with changed encoding formats to be consistent with the aforementioned resampling output format parameters.

[0051] Furthermore, wherein, the decoders for the audio and video data with changed encoding formats and the resamplers for the audio and video data with changed encoding formats are provided by third-party decoders that are not DirectShow.

[0052] Compared with existing technologies, the method and system based on DirectShow for supporting variable encoding of this invention achieve the following beneficial effects:

[0053] (1) The aforementioned method and system based on DirectShow for supporting variable encoding of this invention ensures that when the encoding information format in the audio and video data changes, the new audio and video data's decoding and resampling output format remains consistent with the previously set output format by setting the output format after resampling consistently, eliminating the need to disconnect and update existing renderers, thus ensuring continuity in audio data playback without affecting user experience.

[0054] (2) The aforementioned method and system based on DirectShow for supporting variable encoding of this invention saves linking time for new decoders and renderers when the encoding information format in the audio and video data changes, allowing the new audio and video data's decoding and resampling output format to remain consistent with the previously set output format, reducing user wait time and enhancing user experience.

[0055] Of course, implementing any product of this invention does not necessarily require achieving all of the aforementioned technical effects simultaneously.

[0056] Through the detailed description of an exemplary embodiment of this invention with reference to the accompanying drawings, other features and advantages of this invention will become clear.

Explanation of the attached figures

[0057] The drawings, which are included in the Detailed Explanation and constitute a part of the Detailed Explanation, illustrate embodiments of this invention and are used along with their descriptions to explain the principles of this invention.

[0058] Fig. 1 is a schematic diagram of the system structure for audio and video decoding and playback based on DirectShow in existing technologies;

[0059] Fig. 2 is a flowchart of the method for variable audio and video encoding playback based on DirectShow in existing technologies;

[0060] Fig. 3 is a schematic diagram of the system structure for audio and video decoding and playback based on DirectShow in this invention;

[0061] Fig. 4 is a flowchart of the method supporting variable encoding based on DirectShow in this invention;

[0062] Fig. 5 is a schematic diagram of the format structure of the audio and video data in this invention;

[0063] Fig. 6 is a flowchart of an optional embodiment of the method supporting variable encoding based on DirectShow in this invention embodiment 2;

[0064] Fig. 7 is a flowchart of the process for decoding and resampling audio and video data using FFmpeg;

[0065] Fig. 8 is a flowchart of the process for decoding and resampling using FFmpeg during audio and video playback based on DirectShow;

[0066] Fig. 9 is a flowchart of the initialization process of a specific example of the method supporting variable encoding based on DirectShow in the aforementioned embodiment of this invention;

[0067] Fig. 10 is a flowchart of the process when no change in encoding format is detected in a specific example of the method supporting variable encoding based on DirectShow in this invention embodiment 3;

[0068] Fig. 11 is a flowchart of the process when a change in encoding format is detected in a specific example of the method supporting variable encoding based on DirectShow in this invention embodiment 3;

[0069] Fig. 12 is a schematic diagram of the system structure supporting variable encoding based on DirectShow in this invention.

Specific Implementation Methods

[0070] Now, various exemplary embodiments of this invention will be described in detail with reference to the attached drawings. It should be noted that unless otherwise specifically indicated, the relative arrangements, numerical expressions, and values of components and steps described in these embodiments do not limit the scope of this invention.

[0071] The following description of at least one exemplary embodiment is intended solely for illustrative purposes and does not serve as any limitation on this invention or its applications or uses.

[0072] Known technologies, methods, and devices that are common in the relevant field may not be discussed in detail; however, where appropriate, the aforementioned technologies, methods, and devices should be considered part of the Detailed Explanation.

[0073] In all examples shown and discussed here, any specific values should be interpreted as merely illustrative rather than as limitations. Therefore, other examples of the exemplary embodiment may have different values.

[0074] It should be noted that similar numerals and letters in the following figures denote similar items; therefore, once an item is defined in one figure, further discussion of it is not required in subsequent figures.

[0075] Embodiment 1

[0076] As shown in Fig. 3 and Fig. 4, Fig. 3 is a schematic diagram of the system structure for audio and video decoding and playback based on DirectShow in this invention; wherein, the separator 301 integrates a separation unit 311, video decoder 302, audio decoder 303, video resampler 304, and audio resampler 305. After the audio and video data undergoes separation processing, it is transmitted to the video renderer 306 and audio renderer 307 for rendering playback. Fig. 4 is a flowchart of the method supporting variable encoding based on DirectShow in this invention. The method supporting variable encoding based on DirectShow in this invention avoids issues of audio and video playback discontinuity caused by disconnection and relinking between the decoder and renderer. The method supporting variable encoding based on DirectShow in this embodiment comprises the following steps:

[0077] Step 401: Obtain audio and video data based on DirectShow, sequentially separate, decode, and resample it before transmitting it to the audio and video renderers for rendering playback, and record the current resampling output format parameters; wherein, the aforementioned audio and video data includes: encoding information, video data, and audio data. As shown in Fig. 5, it is a schematic diagram of the format structure of the audio and video data, including: encoding information 501, video data 502, and audio data 503.

[0078] Step 402: Obtain audio and video decoder parameters from the encoding information of the audio and video data in which the encoding format has changed.

[0079] Step 403: Disconnect the current audio and video decoders, and create audio and video decoders for the audio and video data with changed encoding format based on the aforementioned audio and video decoder parameters.

[0080] Step 404: Disconnect the current audio and video resamplers, and create audio and video resamplers for the audio and video data with changed encoding format based on the aforementioned audio and video decoder parameters; and set the audio and video resampling output format parameters of the audio and video data with changed encoding format according to the aforementioned resampling output format parameters. Here, resampling refers to converting formats or parameters of uncompressed data, such as converting video from RGB format to YUV format, or changing the sample rate of audio.

[0081] Step 405: Link the audio and video resamplers of the audio and video data with changed encoding format to the aforementioned audio and video renderers using the audio and video resampling output format parameters of the audio and video data with changed encoding format.

[0082] Step 406: After separating the audio and video data with changed encoding format, decode it using the audio and video decoders of the audio and video data with changed encoding format; and transmit the data to the aforementioned audio and video renderers for rendering playback after resampling using the audio and video resamplers of the audio and video data with changed encoding format.

[0083] The method supporting variable encoding based on DirectShow in this embodiment integrates decoding and resampling functions within the separator, directly sending uncompressed audio and video data to the renderers. This ensures that the output format of the separator remains consistent and unchanged, thereby allowing normal operation without needing to relink between the separator and renderer. It guarantees continuity of playback data when the encoding format or parameters change, ensuring no impact on user experience. Furthermore, the internal decoders and resamplers in the separator can also be provided by non-DirectShow sources (such as using FFmpeg or other third-party decoders). All operations during this process occur within the separator, thus not affecting the user experience.

[0084] Embodiment 2

[0085] Fig. 6 is a flowchart of an optional embodiment of the method supporting variable encoding based on DirectShow as described in Embodiment 2 of this invention. This embodiment describes the specific details for re-decoding and resampling audio and video data when there is a change in encoding information, building upon Embodiment 1. The steps of the method supporting variable encoding based on DirectShow in this embodiment are as follows:

[0086] Step 601: After separating audio and video data obtained based on DirectShow, acquire audio and video track information based on the encoding information of the aforementioned audio and video data (the audio and video information includes multiple audio and video track information), and retrieve the audio and video decoder parameters related to the aforementioned audio and video data from the aforementioned audio and video track information; wherein, the aforementioned audio and video data includes: encoding information, video data, and audio data.

[0087] Step 602: Create audio and video decoders based on the aforementioned audio and video decoder parameters, and set the output format parameters of the aforementioned audio and video decoders.

[0088] Step 603: Create resamplers based on the aforementioned audio and video decoder parameters, and record the current resampling output format parameters.

[0089] Step 604: Transmit the resampled audio and video data to the audio and video renderers for rendering playback, and record the current resampling output format parameters.

[0090] Step 605: Retrieve the audio and video decoder parameters of the audio and video data with changed encoding format from the encoding information where the encoding format has changed in the aforementioned audio and video data.

[0091] Step 606: Disconnect the current audio and video decoders, and create audio and video decoders for the audio and video data with changed encoding format based on the aforementioned audio and video decoder parameters.

[0092] Step 607: Disconnect the current audio and video resamplers, and create audio and video resamplers for the audio and video data with changed encoding format based on the aforementioned audio and video decoder

parameters, and set the audio and video resampling output format parameters of the audio and video data with changed encoding format according to the aforementioned resampling output format parameters; preferably, set the audio and video resampling output format parameters of the audio and video data with changed encoding format to be consistent with the aforementioned resampling output format parameters.

[0093] Step 608: Use the audio and video resampling output format parameters of the audio and video data with changed encoding format to link the audio and video resamplers of the audio and video data with changed encoding format to the aforementioned audio and video renderers.

[0094] Step 609: Retrieve the audio and video decoder parameters of the audio and video data where the encoding format has not changed from the encoding information of the aforementioned audio and video data.

[0095] Step 610: After separating the audio and video data where the encoding format has not changed, transmit it to decode using the audio and video decoders linked to the audio and video data where the encoding format has not changed based on the aforementioned audio and video decoder parameters; and perform resampling using the resamplers of the audio and video data where the encoding format has not changed before transmitting it to the aforementioned audio and video renderers for rendering playback.

[0096] Preferably, in this embodiment, the aforementioned decoding and resampling are provided by third-party decoders that are not DirectShow-based, such as using FFmpeg or other third-party decoders. FFmpeg is an open-source library that has functions for audio and video decoding and resampling. This embodiment uses FFmpeg as an example, and as shown in Fig. 7, it illustrates the flow of audio and video data decoding and resampling using FFmpeg, including the following steps:

[0097] Step 701: Input audio and video data;

[0098] Step 702: Determine whether there are audio and video tracks;

[0099] Step 703: If audio and video tracks exist, retrieve the encoding information of the audio and video tracks;

[0100] Step 704: Create an FFmpeg decoder based on the retrieved encoding information;

[0101] Step 705: Set the output parameters of the FFmpeg decoder;

[0102] Step 706: Create FFmpeg resampling;

[0103] Step 707: Record resampling format parameters;

[0104] Step 708: Create an output transmission function unit (pin), and continue to determine if there is a next audio and video track;

[0105] Step 709: If no audio and video tracks exist, end audio and video data decoding and resampling.

[0106] As shown in Fig. 8, when performing audio and video playback based on DirectShow, the flow of using FFmpeg for decoding and resampling is illustrated, with the specific steps as follows:

[0107] Step 801: Start inputting audio and video data;

[0108] Step 802: Sequentially read audio and video data;

[0109] Step 803: Determine whether audio and video data has been read;

[0110] Step 804: If audio and video data has not been read, then end FFmpeg decoding and resampling;

[0111] Step 805: If audio and video data has been read, compare whether the encoding of the audio and video data has changed;

[0112] Step 806: If the encoding of the audio and video data has not changed, perform FFmpeg decoding;

[0113] Step 807: Perform FFmpeg resampling;

[0114] Step 808: Output the data after FFmpeg resampling for rendering and proceed to the next data reading;

[0115] Step 809: If the encoding of the audio and video data has changed, close the FFmpeg decoder;

[0116] Step 810: Close FFmpeg resampling;

[0117] Step 811: Create a new FFmpeg decoder;

[0118] Step 812: Create a new FFmpeg resampling;

[0119] Step 813: Record FFmpeg resampling format parameters and perform FFmpeg decoding.

[0120] Embodiment 3

[0121] As shown in Fig. 9, this embodiment illustrates an initialization flow diagram of the aforementioned method based on DirectShow that supports variable encoding. The method comprises the following steps:

[0122] Initialization steps for audio and video data:

[0123] Step 901: Obtain the video decoder type as H.264 with parameters of 1920*1080. Here, the currently played video format is H.264, 1920*1080, and the audio format is mp3, 2 channels, 16 bit, 44.1k.

[0124] Step 902: Create an H.264 decoder, setting the decoder output format to YV12, 1920*1080.

[0125] Step 903: Create video resampling, setting the output format to RGB32, 1920*1080, and recording the current format as H.264, 1920*1080, with the resampling output format as RGB32, 1920*1080; create a video output transport function unit (Pin), specifying the format as RGB32, 1920*1080, which will henceforth be used as the standard format for the video output transport function unit.

[0126] Step 904: Create an MP3 decoder, setting the decoder output format to PCM 2 channels, 16 bit, 44.1k.

[0127] Step 905: Create audio resampling, setting the output format to PCM 2 channels, 16 bit, 44.1k, and recording the current format as mp3, 2 channels, 16 bit, 44.1k, with the resampling output format as PCM, 2 channels, 16 bit, 44.1k; create an audio output Pin, specifying the format as PCM 2 channels, 16 bit, 44.1k. This will henceforth be the standard format.

[0128] As shown in Fig. 10, this embodiment illustrates the flow when no changes in encoding format are detected in the specific instance of the aforementioned method based on DirectShow supporting variable encoding. The process steps when the encoding format of the audio and video data remains unchanged during playback based on DirectShow are as follows:

[0129] Step 1001: Read video data, format: H.264, 1920*1080, detecting that the audio and video data encoding format has not changed.

[0130] Step 1002: Decode video data, outputting as YV12, 1920*1080.

[0131] Step 1003: Resample video data, outputting as RGB32, 1920*1080, and output the video data through the Pin in the format of RGB32, 1920*1080.

[0132] Step 1004: Read audio data, format: mp3, 2 channels, 16 bit, 44.1k, detecting that the audio data encoding format and parameters have not changed.

[0133] Step 1005: Decode audio data, outputting as PCM, 2 channels, 16 bit, 44.1k.

[0134] Step 1006: Resample audio data, outputting as PCM, 2 channels, 16 bit, 44.1k, and output the audio data through the Pin in the format of PCM, 2 channels, 16 bit, 44.1k.

[0135] As shown in Fig. 11, this embodiment illustrates the flow when a change in encoding format is detected in the specific instance of the aforementioned method based on DirectShow supporting variable encoding. After playing for a while, it detects that the audio and video data encoding format has changed, with the video format changing to Divx, 720*576, and the audio format changing to AAC, 6 channels, 16 bit, 48k. At this point, the steps for playback based on the method of this invention are as follows:

[0136] Step 1101: Read the encoding information from the audio and video data, detecting that the video encoding format and parameters have changed to Divx, 720*576, differing from the original H.264, 1920*1080.

[0137] Step 1102: Release the original H.264 decoder, create a Divx decoder, setting the video decoder output format to RGB 720*576.

[0138] Step 1103: Release the original video resampler and create a new video resampler, setting the output format to RGB32, 1920*1080, and recording the current video format as Divx 720*576.

[0139] Step 1104: Detect that the audio encoding format and parameters have changed to AAC, 6 channels, 16

bit, 48k, differing from the original mp3, 2 channels, 16 bit, 44.1k.

[0140] Step 1105: Release the original mp3 decoder, create an AAC decoder, and set the audio decoder output format to PCM 6 channels, 16 bit, 48k.

[0141] Step 1106: Release the original audio resampler, create a new audio resampler, setting the output format to PCM 2 channels, 16 bit, 44.1k, and recording the current audio format as AAC 6 channels, 16 bit, 48k.

[0142] Step 1107: Read video data in the format of Divx 720*576, decoding it to output format RGB 720*576.

[0143] Step 1108: Resample the decoded video data, outputting in the format of RGB32 1920*1080, and output the video data through the Pin in the format of RGB32 1920*1080.

[0144] Step 1109: Read audio data in the format of AAC, 6 channels, 16 bit, 48k, decoding it to output format PCM, 6 channels, 16 bit, 48k.

[0145] Step 1110: Resample the decoded audio data, outputting in the format of PCM 2 channels, 16 bit, 44.1k, and output the audio data through the Pin in the format of PCM, 2 channels, 16 bit, 44.1k.

[0146] Embodiment 4

[0147] As shown in Fig. 12, this embodiment illustrates the system architecture based on DirectShow that supports variable encoding. The system for implementing the aforementioned method supporting variable encoding based on DirectShow includes: Player 1201, Acquisition Module 1202, Decoder Creation Module 1203, Resampling Creation Module 1204, and Linking Module 1205.

[0148] Among them, the aforementioned Player 1201 is connected to the aforementioned Acquisition Module 1202, used for obtaining audio and video data based on DirectShow, sequentially separating, decoding, and resampling before transmitting to the audio and video renderer for playback, and recording the current resampling output format parameters; where the aforementioned audio and video data includes: encoding information, video data, and audio data;

[0149] The aforementioned Acquisition Module 1202 is connected to the aforementioned Player 1201 and Decoder Creation Module 1203, used for obtaining the audio and video decoder parameters of audio and video data with changed encoding formats from the encoding information acquired from the aforementioned audio and video data;

[0150] The aforementioned Decoder Creation Module 1203 is connected to the aforementioned Acquisition Module 1202 and Resampling Creation Module 1204, used for creating audio and video decoders for the audio and video data with changed encoding formats according to the aforementioned audio and video decoder parameters;

[0151] The aforementioned Resampling Creation Module 1204 is connected to the aforementioned Decoder Creation Module 1203 and Linking Module 1205, used for creating audio and video resamplers for the audio and video data with changed encoding formats according to the aforementioned parameters;

[0152] And setting the audio and video resampling output format parameters for the audio and video data with changed encoding formats based on the aforementioned resampling output format parameters.

[0153] The aforementioned Linking Module 1205 is connected to the aforementioned Resampling Creation Module 1204 and Player 1201, used for linking the audio and video resamplers of the audio and video data with changed encoding formats to the aforementioned audio and video renderer using the audio and video resampling output format parameters.

[0154] The aforementioned Player 1201 is also connected to the aforementioned Linking Module 1205, used for decoding the audio and video data with changed encoding formats after separation using the audio and video decoders of the audio and video data with changed encoding formats; and transmitting the resampled audio and video data to the aforementioned audio and video renderer for rendering and playback.

[0155] Furthermore, the aforementioned Player 1201 is also used for:

[0156] Obtaining audio and video data based on DirectShow, separating it, and acquiring audio and video track information according to the encoding information of the aforementioned audio and video data, and obtaining the relevant audio and video decoder parameters from the aforementioned audio and video track information;

[0157] Creating audio and video decoders according to the aforementioned audio and video decoder parameters, and setting the output format parameters of the aforementioned audio and video decoders;

[0158] Creating resamplers according to the aforementioned audio and video decoder parameters, and recording the current output format parameters of the resampling.

[0159] The aforementioned Acquisition Module 1202 is also used for:

[0160] Obtaining the audio and video decoder parameters of audio and video data whose encoding format has not changed from the encoding information format obtained from the aforementioned audio and video data;

[0161] The aforementioned Decoder Creation Module 1203 is also used for: After separating the audio and video data whose encoding format has not changed, transmitting it to decode the audio and video data linked by the audio and video decoder parameters using the aforementioned unchanged encoding format;

[0162] The aforementioned Resampling Creation Module 1204 is also used for: Using resamplers for audio and video data of the unchanged encoding format to resample the decoded audio and video data of the unchanged encoding format;

[0163] The aforementioned Player 1201 is also used for: Transmitting the resampled audio and video data of the unchanged encoding format to the aforementioned audio and video renderer for rendering and playback.

[0164] The aforementioned Resampling Creation Module 1203 is also used for:

[0165] Setting the output format parameters of the audio and video resampling of the audio and video data with changed encoding format to be consistent with the aforementioned resampling output format parameters.

[0166] Preferably, in this embodiment, the decoder for audio and video data with changed encoding format and the resampler for audio and video data with changed encoding format are provided by a third-party decoder that is not DirectShow, for example, decoding and resampling audio and video data with changed encoding formats through FFmpeg.

[0167] From the above embodiments, it can be seen that the method and system based on DirectShow supporting variable encoding have beneficial effects:

[0168] (1) The method and system of this invention based on DirectShow support variable encoding. By ensuring consistent output formats after resampling, when the encoding information format changes in the audio and video data, the new audio and video data's decoded resampling output format remains consistent with the previously set output format, eliminating the need to disconnect and update existing decoders and renderers, ensuring continuity in audio playback without affecting user experience.

[0169] (2) The method and system of this invention based on DirectShow support variable encoding. When the encoding information format changes in the audio and video data, the new audio and video data's decoded resampling output format remains consistent with the previously set output format, saving the linking time for new decoders and renderers, reducing user wait times, and enhancing user experience.

[0170] Technicians in the field should understand that the embodiments of this invention can be provided as methods, devices, or computer program products. Therefore, this invention can take the form of a completely hardware embodiment, a completely software embodiment, or a combination of software and hardware aspects. Moreover, this invention can be implemented in the form of computer program products embodied on one or more computer-readable storage media containing computer-readable program codes (including but not limited to magnetic disk storage, CD-ROMs, optical storage, etc.).

[0171] Although some specific embodiments of this invention have been described in detail through examples,

those skilled in the art should understand that the above examples are provided for illustration purposes only and are not intended to limit the scope of this invention. Those skilled in the art should understand that modifications can be made to the above embodiments without departing from the scope and spirit of this invention. The scope of this invention is defined by the attached Claims.

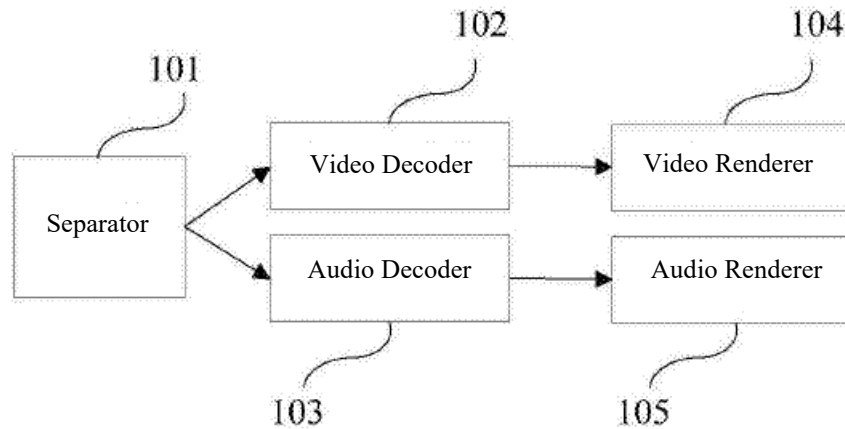


Fig. 1

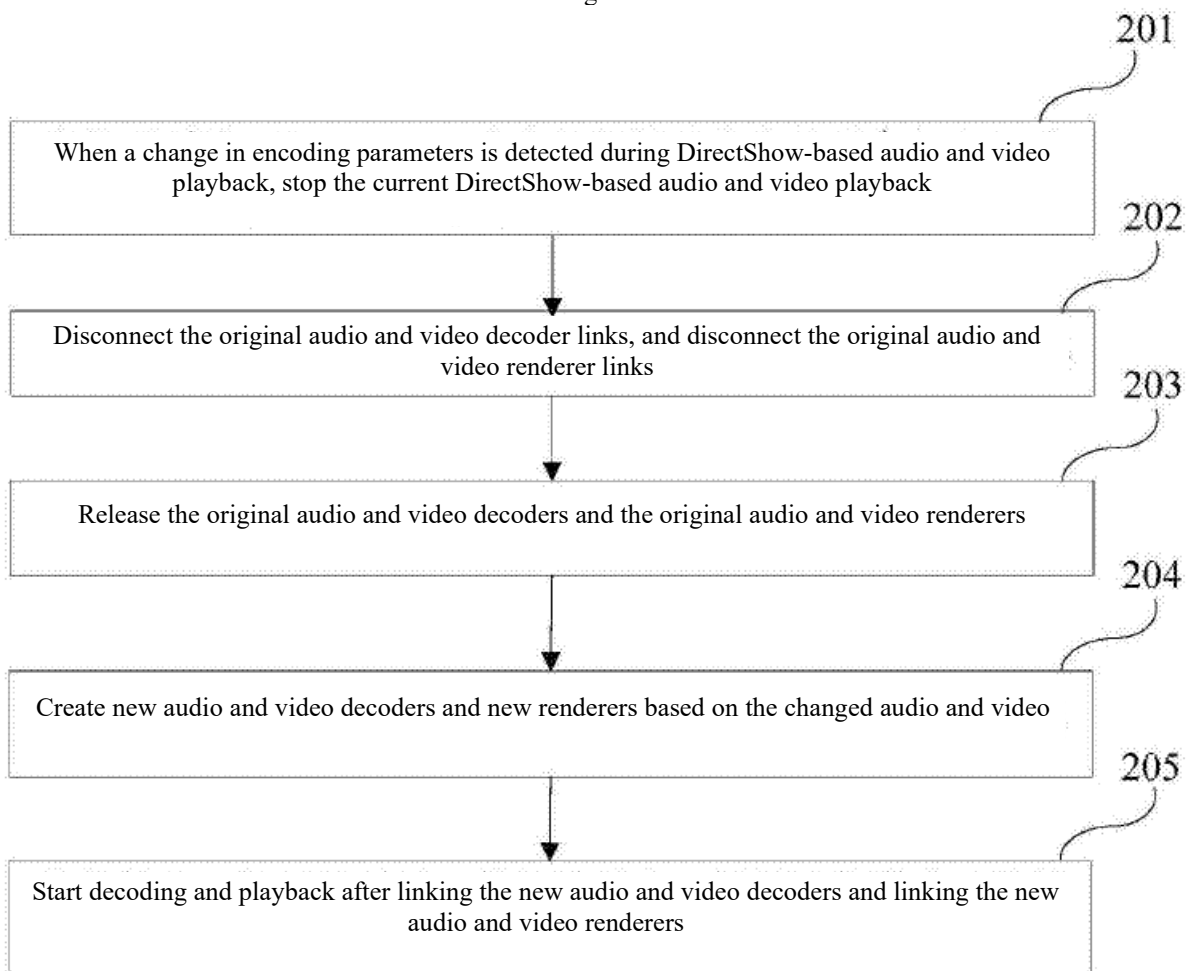


Fig. 2

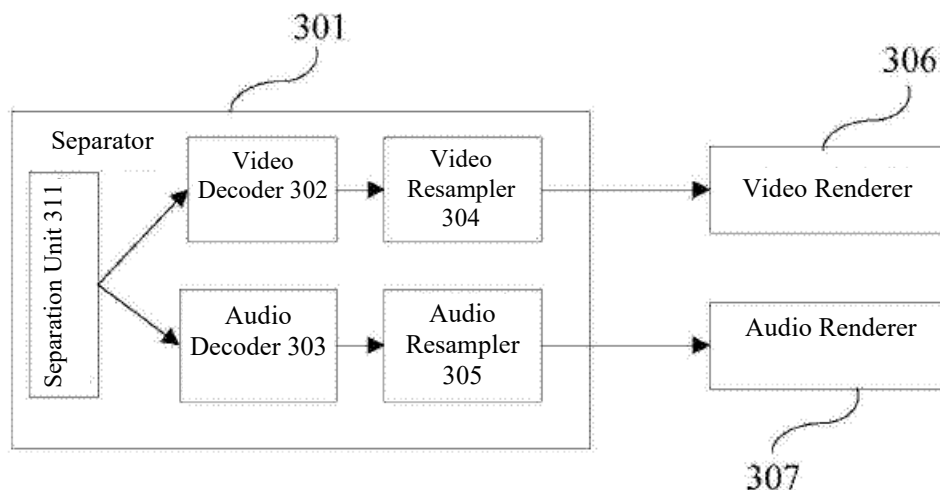


Fig. 3

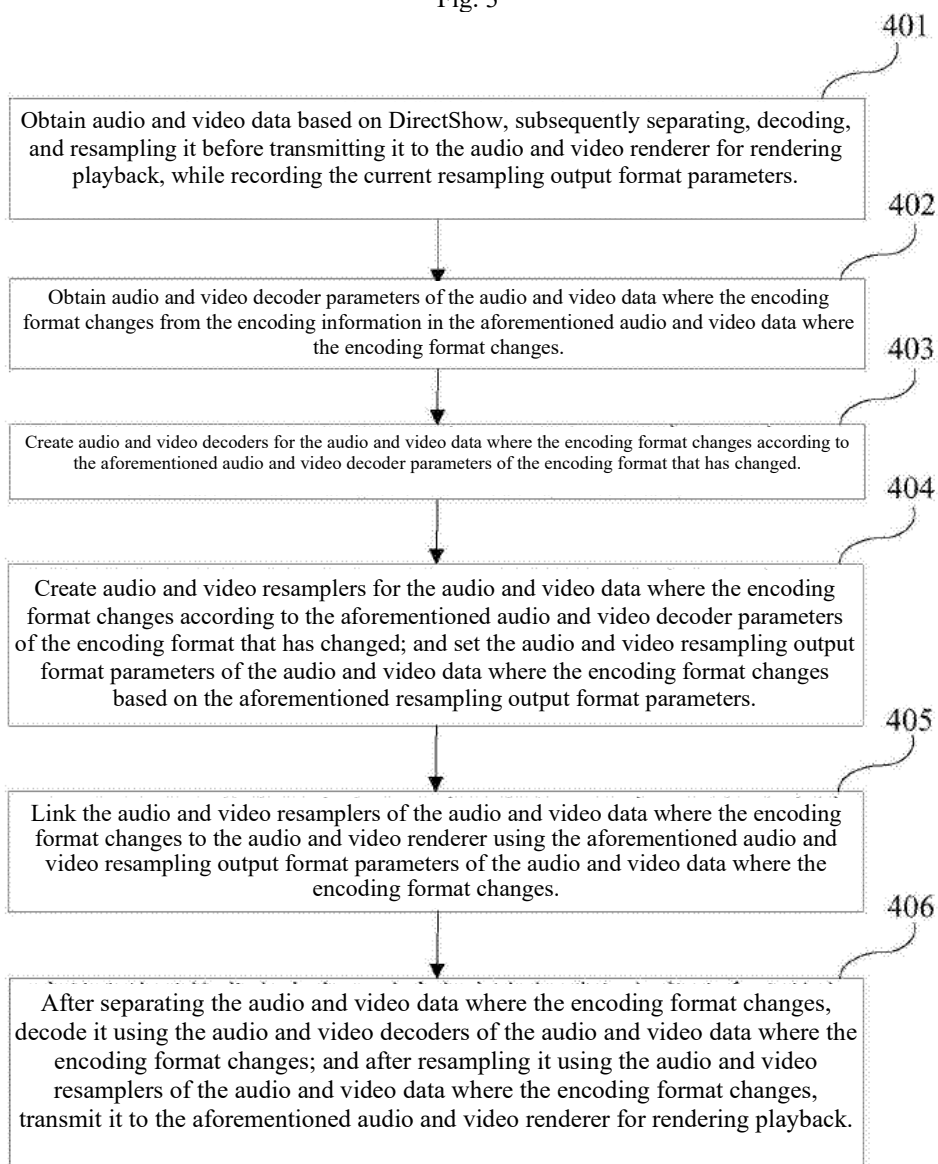


Fig. 4

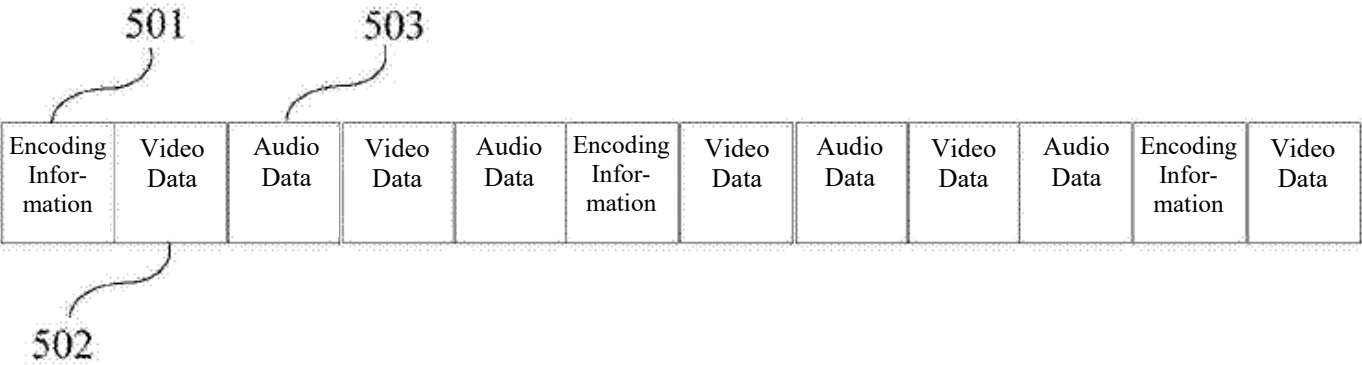


Fig. 5

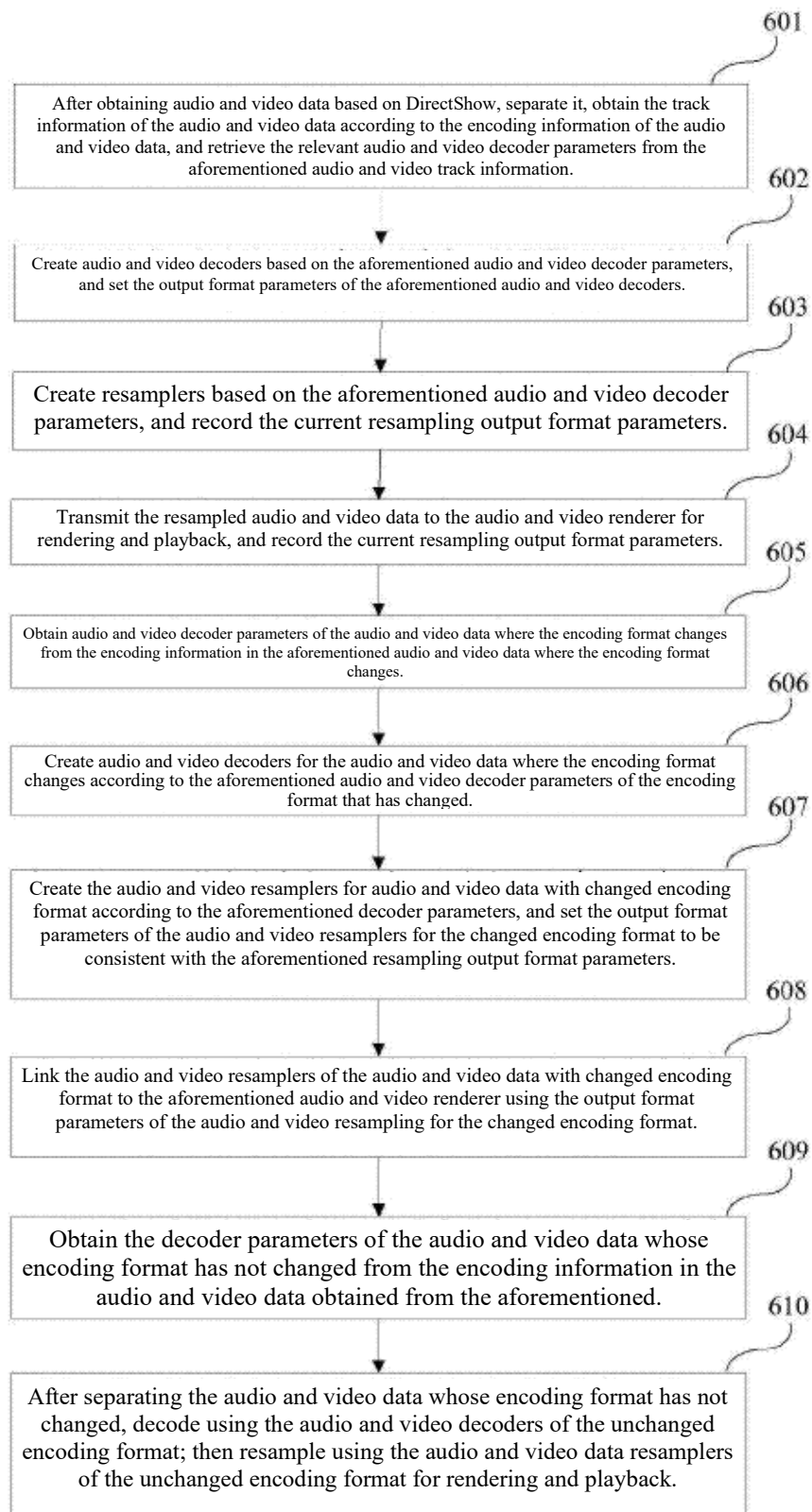


Fig. 6

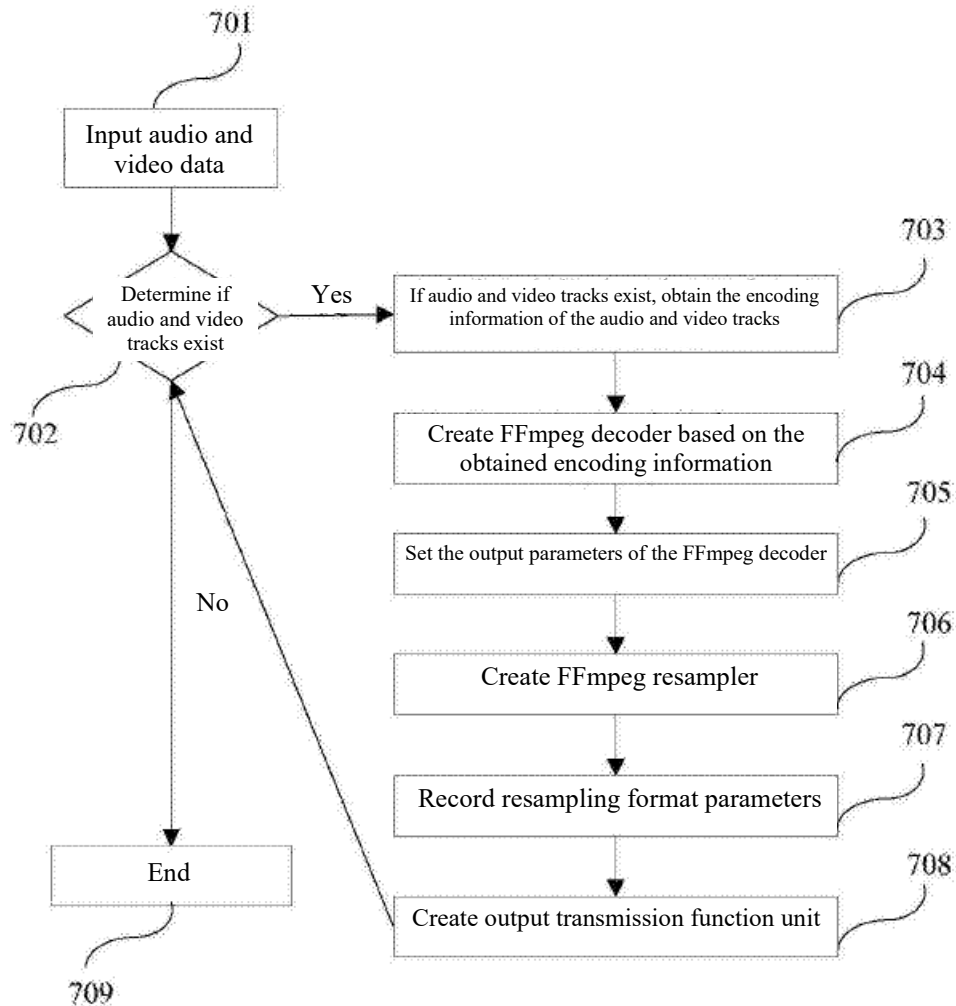


Fig. 7

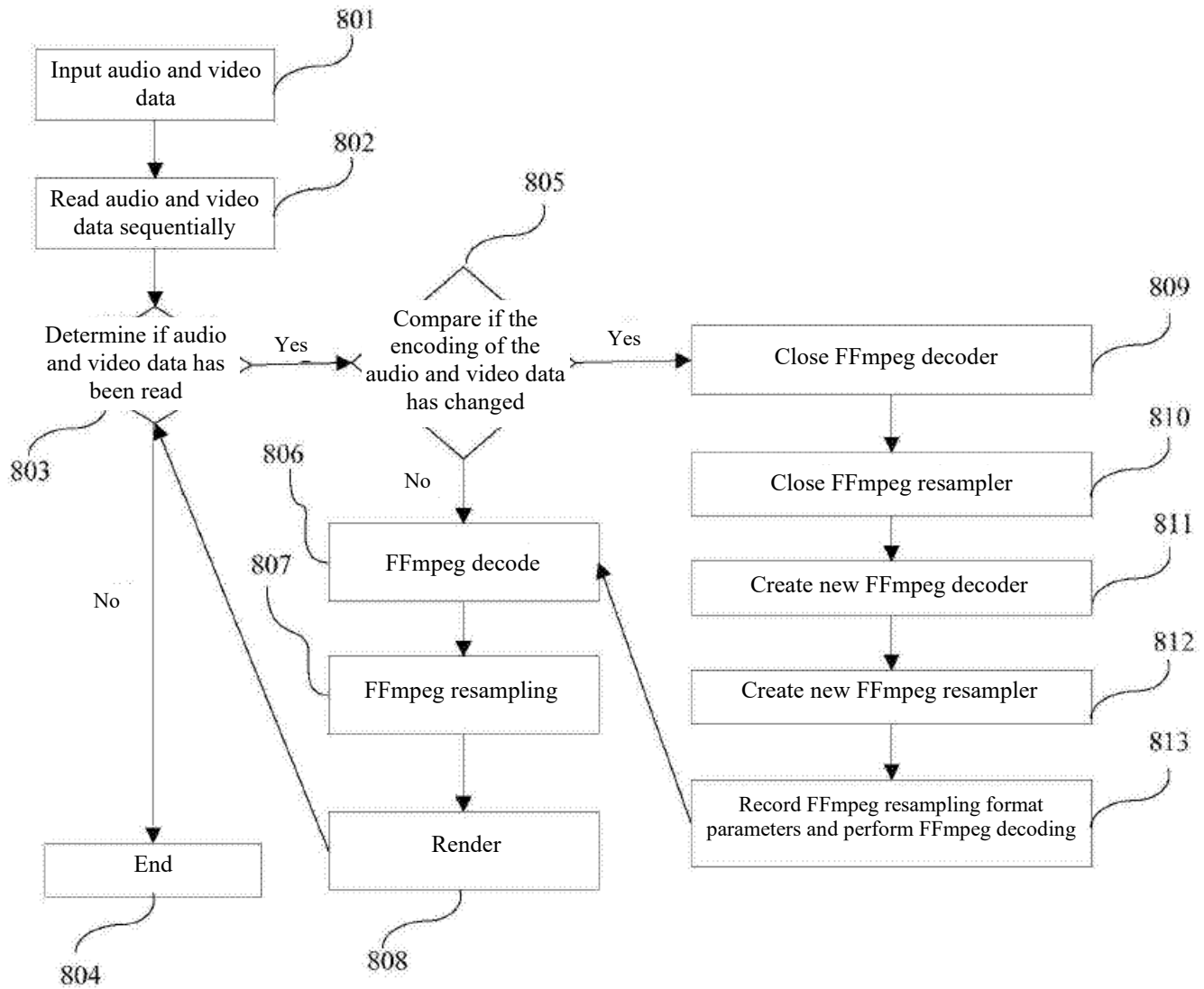


Fig. 8

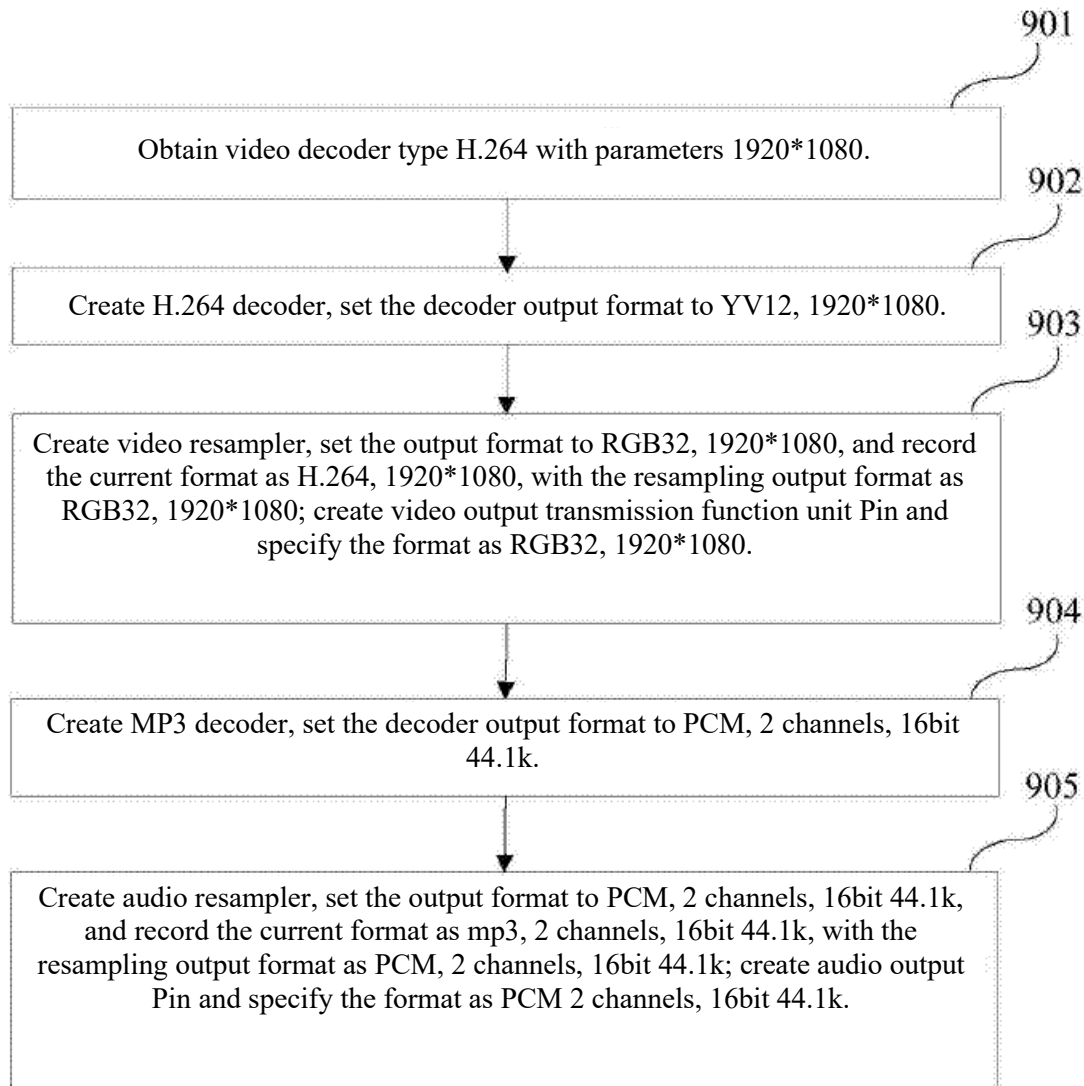


Fig. 9

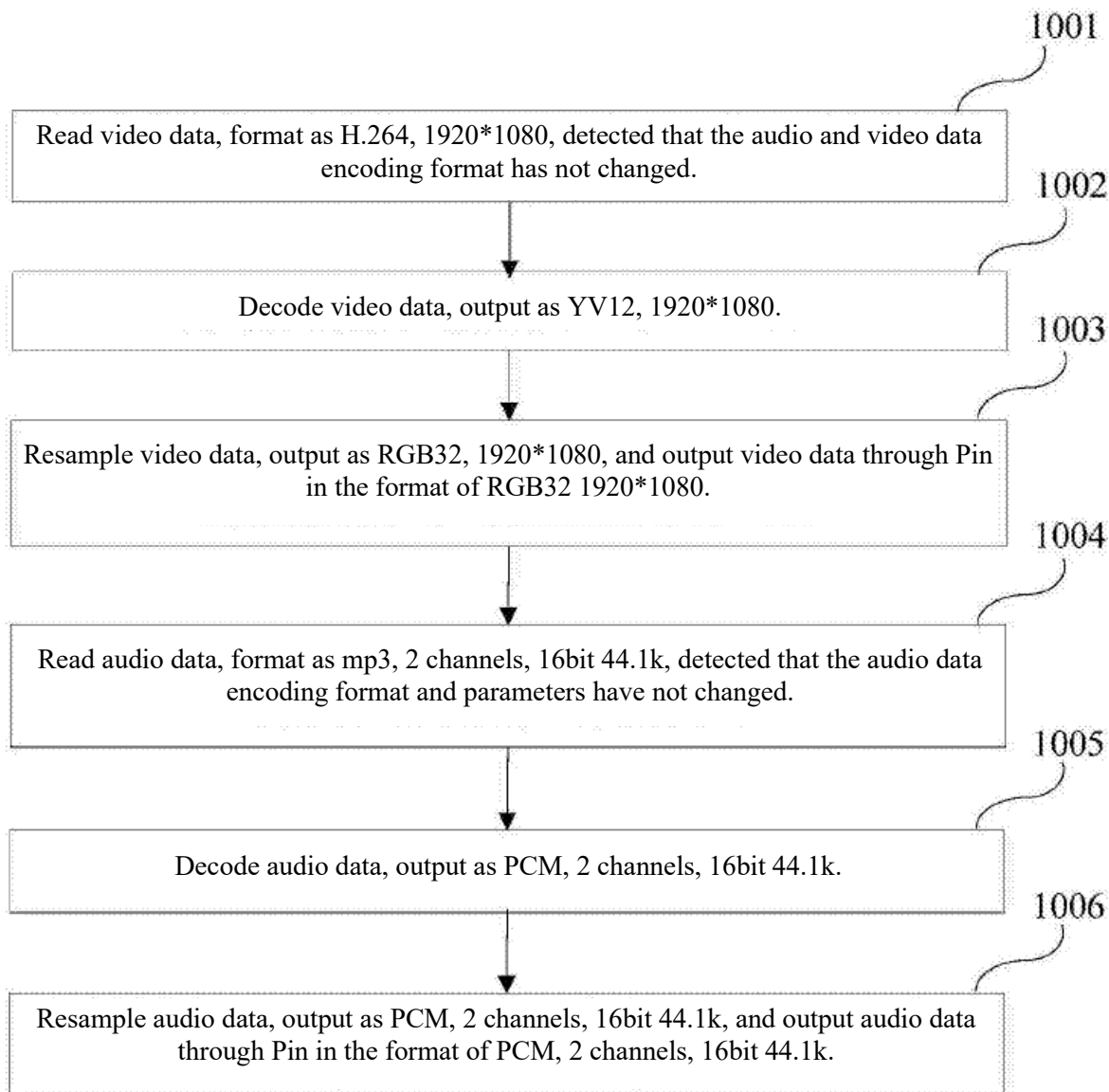


Fig. 10

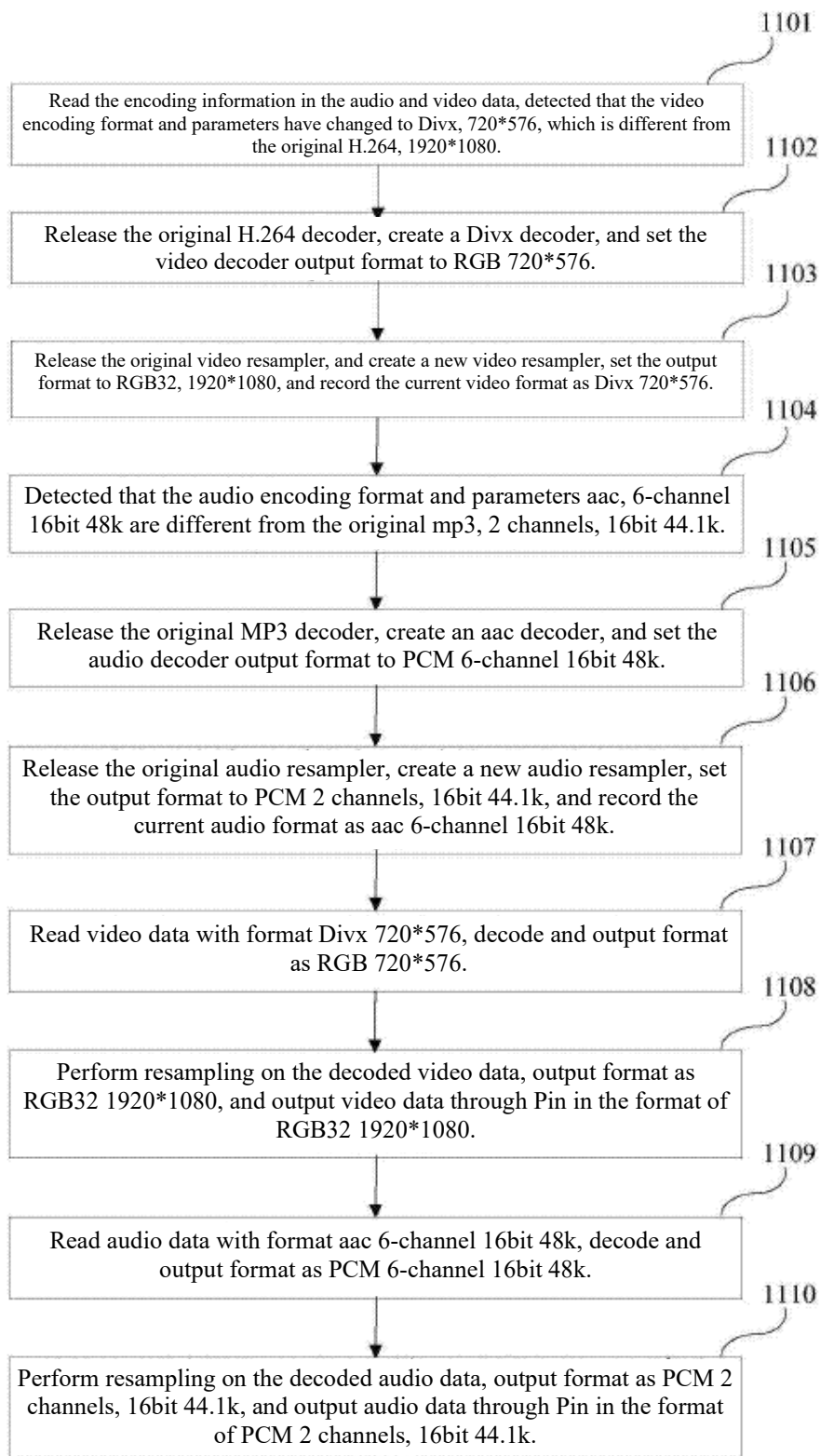


Fig. 11

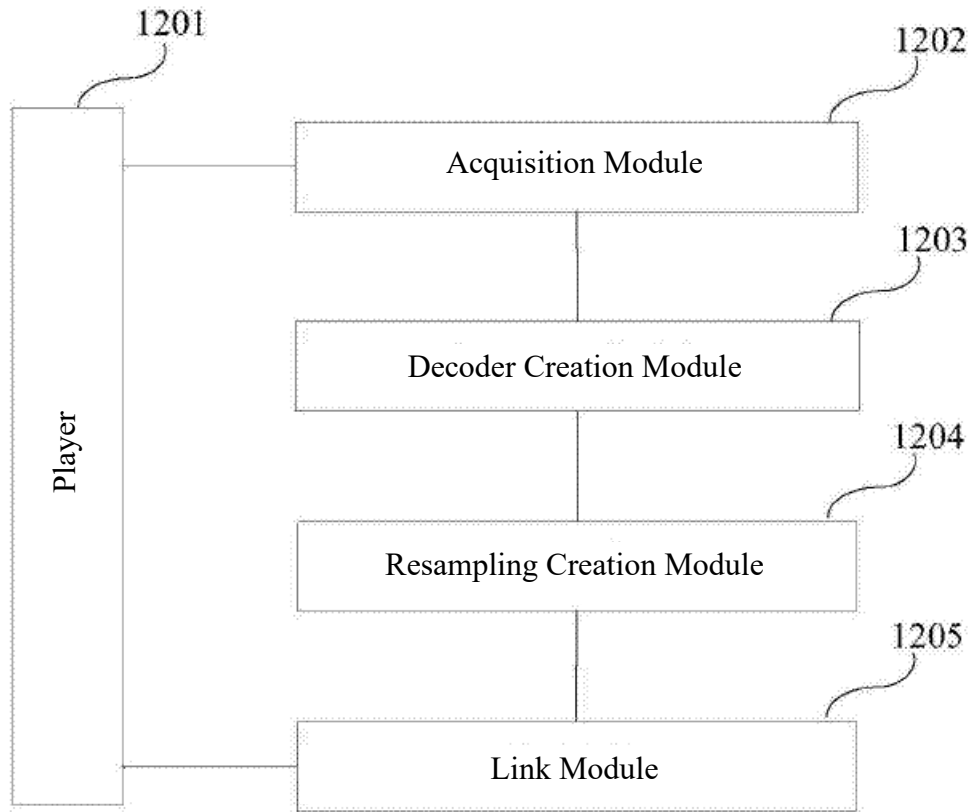


Fig. 12



(12)发明专利申请

(10)申请公布号 CN 106060628 A

(43)申请公布日 2016. 10. 26

(21)申请号 201610491470.1

H04N 21/4402(2011.01)

(22)申请日 2016.06.28

(71)申请人 暴风集团股份有限公司

地址 100083 北京市海淀区学院路51号首
享科技大厦6层

(72)发明人 吴超 朱春明 朱彦东 张鹏宇

(74)专利代理机构 北京晟睿智杰知识产权代理
事务所(特殊普通合伙)
11603

代理人 于淼

(51)Int.Cl.

H04N 21/426(2011.01)

H04N 21/434(2011.01)

H04N 21/438(2011.01)

H04N 21/4385(2011.01)

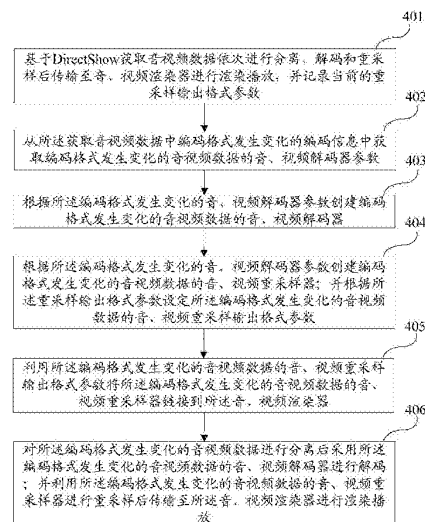
权利要求书3页 说明书11页 附图10页

(54)发明名称

基于DirectShow支持可变编码的方法及系
统

(57)摘要

本发明公开基于DirectShow支持可变编码的方法,包括:基于DirectShow获取音视频数据依次进行分离、解码和重采样后渲染播放,并记录当前的重采样输出格式参数;获取编码格式发生变化的音视频数据的音、视频解码器参数;创建编码格式发生变化的音视频数据的音、视频解码器及音、视频重采样器;并根据重采样输出格式参数设定编码格式发生变化的音视频数据的音、视频重采样输出格式参数并链接到音、视频渲染器;对编码格式发生变化的音视频数据进行分离、解码、重采样后传输至音、视频渲染器进行渲染播放。本发明不需要断开并更新现有的渲染器,保证了音频数据播放的连续性,不会在用户观看音视频时造成停顿。



1.一种基于DirectShow支持可变编码的方法,其特征在于,包括:

基于DirectShow获取音视频数据依次进行分离、解码和重采样后传输至音、视频渲染器进行渲染播放,并记录当前的重采样输出格式参数;其中,所述音视频数据包括:编码信息、视频数据及音频数据;

从所述获取音视频数据中编码格式发生变化的编码信息中获取编码格式发生变化的音视频数据的音、视频解码器参数;

根据所述编码格式发生变化的音、视频解码器参数创建编码格式发生变化的音视频数据的音、视频解码器;

根据所述编码格式发生变化的音、视频解码器参数创建编码格式发生变化的音视频数据的音、视频重采样器;

并根据所述重采样输出格式参数设定所述编码格式发生变化的音视频数据的音、视频重采样输出格式参数;

利用所述编码格式发生变化的音视频数据的音、视频重采样输出格式参数将所述编码格式发生变化的音视频数据的音、视频重采样器链接到所述音、视频渲染器;

对所述编码格式发生变化的音视频数据进行分离后采用所述编码格式发生变化的音视频数据的音、视频解码器进行解码;并利用所述编码格式发生变化的音视频数据的音、视频重采样器进行重采样后传输至所述音、视频渲染器进行渲染播放。

2.根据权利要求1所述的基于DirectShow支持可变编码的方法,其特征在于,所述基于DirectShow获取音视频数据依次进行分离、解码、重采样及渲染后播放,并记录当前的重采样后的输出格式参数,包括:

基于DirectShow获取音视频数据进行分离后,根据所述音视频数据的编码信息获取音、视频轨信息,并从所述音、视频轨信息中获取所述音视频数据相关的音、视频解码器参数;

根据所述音、视频解码器参数创建音、视频解码器,并设定所述音、视频解码器的输出格式参数;

根据所述音、视频解码器参数创建重采样器,并记录当前的重采样输出格式参数。

3.根据权利要求1所述的基于DirectShow支持可变编码的方法,其特征在于,还包括:

从所述获取音视频数据中编码格式未发生变化的编码信息中获取编码格式未发生变化的音视频数据的音、视频解码器参数;

对所述编码格式未发生变化的音视频数据进行分离后传输至利用所述编码格式未发生变化的音视频数据的音、视频解码器参数链接的编码格式未发生变化的音视频数据的音、视频解码进行解码;

并采用编码格式未发生变化的音视频数据的音、视频数据的重采样器进行重采样后传输至所述音、视频渲染器进行渲染播放。

4.根据权利要求1所述的基于DirectShow支持可变编码的方法,其特征在于,所述根据所述重采样输出格式参数设定所述码格式发生变化的音视频数据的音、视频重采样输出格式参数,还包括:

设定所述编码格式发生变化的音视频数据的音、视频重采样输出格式参数与所述重采样输出格式参数一致。

5. 根据权利要求1所述的基于DirectShow支持可变编码的方法,其特征在于,所述编码格式发生变化的音视频数据的解码器和编码格式发生变化的音视频数据的重采样器由非DirectShow的第三方解码器提供。

6. 一种基于DirectShow支持可变编码的系统,其特征在于,包括:播放器、检测模块、解码器创建模块、重采样创建模块及链接模块,其中,

所述播放器,用于基于DirectShow获取音视频数据依次进行分离、解码和重采样后传输至音、视频渲染器进行渲染播放,并记录当前的重采样输出格式参数;其中,所述音视频数据包括:编码信息、视频数据及音频数据;

所述获取模块,用于从所述获取音视频数据中编码格式发生变化的编码信息中获取编码信息发生变化的音视频数据的音、视频解码器参数;

所述解码器创建模块,用于根据所述编码格式发生变化的音、视频解码器参数创建编码格式发生变化的音视频数据的音、视频解码器;

所述重采样创建模块,用于根据所述编码格式发生变化后音、视频解码器参数创建编码格式发生变化的音视频数据的音、视频重采样器;

并根据所述重采样输出格式参数设定所述编码格式发生变化的音视频数据的音、视频重采样输出格式参数;

所述链接模块,用于利用所述编码格式发生变化的音视频数据的音、视频重采样输出格式参数将所述编码格式发生变化的音视频数据的音、视频重采样器链接到所述音、视频渲染器;

所述播放器,还用于对所述编码格式发生变化的音视频数据进行分离后采用所述编码格式发生变化的音视频数据的音、视频解码器进行解码;并利用所述编码格式发生变化的音视频数据的音、视频重采样器进行重采样后传输至所述音、视频渲染器进行渲染播放。

7. 根据权利要求6所述的基于DirectShow支持可变编码的系统,其特征在于,所述播放器,还用于:

基于DirectShow获取音视频数据进行分离后,根据所述音视频数据的编码信息获取音、视频轨信息,并从所述音、视频轨信息中获取所述音视频数据相关的音、视频解码器参数;

根据所述音、视频解码器参数创建音、视频解码器,并设定所述音、视频解码器的输出格式参数;

根据所述音、视频解码器参数创建重采样器,并记录当前的重采样的输出格式参数。

8. 根据权利要求6所述的基于DirectShow支持可变编码的系统,其特征在于,所述获取模块,还用于:从所述获取的音视频数据中编码信息格式未发生变化的编码信息中获取编码格式未发生变化的音视频数据的音、视频解码器参数;

所述解码器创建模块,还用于:对所述编码格式未发生变化的音视频数据进行分离后传输至利用所述编码格式未发生变化的音视频数据的音、视频解码器参数链接的编码格式未发生变化的音视频数据的音、视频解码进行解码;

所述重采样创建模块,还用于:采用编码格式未发生变化的音视频数据的音、视频数据的重采样器对解码后的编码格式未发生变化的音、视频数据进行重采样;

所述播放器,还用于:将重采样后的编码格式未发生变化的音视频数据的音、视频数据

传输至所述音、视频渲染器进行渲染播放。

9. 根据权利要求6所述的基于DirectShow支持可变编码的系统,其特征在于,所述重采样创建模块,还用于:

设定所述编码格式发生变化的音视频数据的音、视频重采样输出格式参数与所述重采样输出格式参数一致。

10. 根据权利要求6所述的基于DirectShow支持可变编码的系统,其特征在于,所述编码格式发生变化的音视频数据的解码器和编码格式发生变化的音视频数据的重采样器由非DirectShow的第三方解码器提供。

基于DirectShow支持可变编码的方法及系统

技术领域

[0001] 本发明涉及视频播放的技术领域,更具体地,涉及一种基于DirectShow支持可变编码的方法及系统。

背景技术

[0002] 基于DirectShow播放现有格式的视频文件时,一旦变化(如增加视频编码格式,mp4增加对H.265支持后会增加“hevc”结构),就需要升级分离器,否则无法识别、播放。

[0003] DirectShow是微软公司在ActiveMovie和Video for Windows的基础上推出的新一代基于COM(Component Object Model)的流媒体处理的开发包,与DirectX开发包一起发布。DirectShow使用Filter Graph(滤镜图)模型来管理整个数据流的处理过程,DirectShow工作的时候,通过MediaType进行Filter(DirectShow的功能模块单元)之间的链接,通过MediaSample进行数据传输。

[0004] DirectShow运行的方式通常是:开发者创建Filter Graph,可订制地把相应的转换方案加入Filter Graph,然后播放文件。运用DirectShow,可以很方便地从支持WDM驱动模型的采集卡上捕获数据,并且进行相应的后期处理乃至存储到文件中。使在多媒体数据库管理系统(MDBMS)中多媒体数据的存取变得更加方便。DirectShow广泛地支持各种媒体格式,包括Asf、Mpeg、Avi、Dv、Mp3、Wave等,为多媒体流的捕捉和回放提供了强有力的支持。因此,DirectShow作为一种windows操作系统上广泛采用的音视频播放方案在视频播放领域得到普遍应用。DirectShow定义了一系列的标准接口,可以方便的进行功能扩展,如播放需要的分离器、解码器等。

[0005] 目前,基于DirectShow进行网络视频播放,尤其是进行网络直播的过程中,可能会伴随着视频类型的改变,造成基于DirectShow进行编码的参数发生改变的情况,例如,在直播过程中插入广告时,会发生音视频格式的变化、编码器采样率的改变以及音视频的分辨率的改变等。举例来说,基于DirectShow在进行音视频播放的时候,如果编码参数发生变化(如音频格式类型由mp3变为aac、采样率由48k变为44.1k或视频分辨率大小由1920*1080变为720*576等)时,还按照原来的解码器进行解码播放显然是不行的,如图1为传统方案中基于DirectShow进行音视频解码播放的系统结构示意图,传统方案中基于DirectShow的音视频播放方法是:先在基于DirectShow的分离器101中进行音视频的分离;再分别在视频解码器102及音频解码器103中进行视频和音频解码;经过解码处理后的视频和音频分别在视频渲染器104和音频渲染器105中进行渲染后播放。传统的基于DirectShow进行音视频可变编码播放的解决方案步骤如图2所示:

[0006] 步骤201、当检测到基于DirectShow的音视频播放发生编码参数变化时,停止当前基于DirectShow的音视频播放;

[0007] 步骤202、断开原音、视频解码器链接,并断开原音、视频渲染器链接;

[0008] 步骤203、释放原音、视频解码器和原音、视频渲染器;

[0009] 步骤204、根据变化后的音视频创建新的音、视频解码器和新的渲染器。

[0010] 步骤205、链接新音、视频解码器,并链接新音、视频渲染器后开始解码播放。

[0011] 从图2中可以看出,传统的基于DirectShow进行可变编码音视频解码播放的解决方案中需要重新链接解码器和渲染器,然后再继续播放新的音视频。在重新链接解码器和渲染器的过程中会产生停顿、画面会变黑或发生跳动等现象,从而影响用户的观看体验;并且在断开原解码器和渲染器,进行新解码器和渲染器的链接过程需要一定的时间,这会增加用户的等待时间,不利于用户的观看体验。

[0012] 因此,提供一种基于DirectShow快速、流畅地进行可变编码播放音视频的方法是本领域亟待解决的问题。

发明内容

[0013] 有鉴于此,本发明提供了一种基于DirectShow支持可变编码的方法及系统,解决了现有技术中基于DirectShow进行可变编码音视频解码播放会产生停顿、画面会变黑或发生跳动等现象的问题。

[0014] 为了解决上述技术问题,本发明提出一种基于DirectShow支持可变编码的方法,包括:

[0015] 基于DirectShow获取音视频数据依次进行分离、解码和重采样后传输至音、视频渲染器进行渲染播放,并记录当前的重采样输出格式参数;其中,所述音视频数据包括:编码信息、视频数据及音频数据;

[0016] 从所述获取音视频数据中编码格式发生变化的编码信息中获取编码格式发生变化的音视频数据的音、视频解码器参数;

[0017] 根据所述编码格式发生变化的音、视频解码器参数创建编码格式发生变化的音视频数据的音、视频解码器;

[0018] 根据所述编码格式发生变化的音、视频解码器参数创建编码格式发生变化的音视频数据的音、视频重采样器;

[0019] 并根据所述重采样输出格式参数设定所述编码格式发生变化的音视频数据的音、视频重采样输出格式参数;

[0020] 利用所述编码格式发生变化的音视频数据的音、视频重采样输出格式参数将所述编码格式发生变化的音视频数据的音、视频重采样器链接到所述音、视频渲染器;

[0021] 对所述编码格式发生变化的音视频数据进行分离后采用所述编码格式发生变化的音视频数据的音、视频解码器进行解码;并利用所述编码格式发生变化的音视频数据的音、视频重采样器进行重采样后传输至所述音、视频渲染器进行渲染播放。

[0022] 进一步地,其中,所述基于DirectShow获取音视频数据依次进行分离、解码、重采样及渲染后播放,并记录当前的重采样后的输出格式参数,包括:

[0023] 基于DirectShow获取音视频数据进行分离后,根据所述音视频数据的编码信息获取音、视频轨信息,并从所述音、视频轨信息中获取所述音视频数据相关的音、视频解码器参数;

[0024] 根据所述音、视频解码器参数创建音、视频解码器,并设定所述音、视频解码器的输出格式参数;

[0025] 根据所述音、视频解码器参数创建重采样器,并记录当前的重采样输出格式参数。

- [0026] 进一步地,其中,该方法还包括:
- [0027] 从所述获取音视频数据中编码格式未发生变化的编码信息中获取编码格式未发生变化的音视频数据的音、视频解码器参数;
- [0028] 对所述编码格式未发生变化的音视频数据进行分离后传输至利用所述编码格式未发生变化的音视频数据的音、视频解码器参数链接的编码格式未发生变化的音视频数据的音、视频解码进行解码;
- [0029] 并采用编码格式未发生变化的音视频数据的音、视频数据的重采样器进行重采样后传输至所述音、视频渲染器进行渲染播放。
- [0030] 进一步地,其中,所述根据所述重采样输出格式参数设定所述码格式发生变化的音视频数据的音、视频重采样输出格式参数,还包括:
- [0031] 设定所述编码格式发生变化的音视频数据的音、视频重采样输出格式参数与所述重采样输出格式参数一致。
- [0032] 进一步地,其中,所述编码格式发生变化的音视频数据的解码器和编码格式发生变化的音视频数据的重采样器由非DirectShow的第三方解码器提供。
- [0033] 另一方面,本发明还提供一种基于DirectShow支持可变编码的系统,所述系统包括:播放器、检测模块、解码器创建模块、重采样创建模块及链接模块,其中,
- [0034] 所述播放器,用于基于DirectShow获取音视频数据依次进行分离、解码和重采样后传输至音、视频渲染器进行渲染播放,并记录当前的重采样输出格式参数;其中,所述音视频数据包括:编码信息、视频数据及音频数据;
- [0035] 所述获取模块,用于从所述获取音视频数据中编码格式发生变化的编码信息中获取编码信息发生变化的音视频数据的音、视频解码器参数;
- [0036] 所述解码器创建模块,用于根据所述编码格式发生变化的音、视频解码器参数创建编码格式发生变化的音视频数据的音、视频解码器;
- [0037] 所述重采样创建模块,用于根据所述编码格式发生变化后音、视频解码器参数创建编码格式发生变化的音视频数据的音、视频重采样器;
- [0038] 并根据所述重采样输出格式参数设定所述编码格式发生变化的音视频数据的音、视频重采样输出格式参数;
- [0039] 所述链接模块,用于利用所述编码格式发生变化的音视频数据的音、视频重采样输出格式参数将所述编码格式发生变化的音视频数据的音、视频重采样器链接到所述音、视频渲染器;
- [0040] 所述播放器,还用于对所述编码格式发生变化的音视频数据进行分离后采用所述编码格式发生变化的音视频数据的音、视频解码器进行解码;并利用所述编码格式发生变化的音视频数据的音、视频重采样器进行重采样后传输至所述音、视频渲染器进行渲染播放。
- [0041] 进一步地,其中,所述播放器,还用于:
- [0042] 基于DirectShow获取音视频数据进行分离后,根据所述音视频数据的编码信息获取音、视频轨信息,并从所述音、视频轨信息中获取所述音视频数据相关的音、视频解码器参数;
- [0043] 根据所述音、视频解码器参数创建音、视频解码器,并设定所述音、视频解码器的

输出格式参数；

[0044] 根据所述音、视频解码器参数创建重采样器，并记录当前的重采样的输出格式参数。

[0045] 进一步地，其中，所述获取模块，还用于：从所述获取的音视频数据中编码信息格式未发生变化的编码信息中获取编码格式未发生变化的音视频数据的音、视频解码器参数；

[0046] 所述解码器创建模块，还用于：对所述编码格式未发生变化的音视频数据进行分离后传输至利用所述编码格式未发生变化的音视频数据的音、视频解码器参数链接的编码格式未发生变化的音视频数据的音、视频解码进行解码；

[0047] 所述重采样创建模块，还用于：采用编码格式未发生变化的音视频数据的音、视频数据的重采样器对解码后的编码格式未发生变化的音、视频数据进行重采样；

[0048] 所述播放器，还用于：将重采样后的编码格式未发生变化的音视频数据的音、视频数据传输至所述音、视频渲染器进行渲染播放。

[0049] 进一步地，其中，所述重采样创建模块，还用于：

[0050] 设定所述编码格式发生变化的音视频数据的音、视频重采样输出格式参数与所述重采样输出格式参数一致。

[0051] 进一步地，其中，所述编码格式发生变化的音视频数据的解码器和编码格式发生变化的音视频数据的重采样器由非DirectShow的第三方解码器提供。

[0052] 与现有技术相比，本发明的基于DirectShow支持可变编码的方法及系统，实现了如下的有益效果：

[0053] (1)本发明所述的基于DirectShow支持可变编码的方法及系统，通过设置重采样后的输出格式一致，使得音视频数据中编码信息格式发生变化时，新的音视频数据解码后重采样输出格式与先前设定的输出格式一致，不需要断开并更新现有的渲染器，保证了音频数据播放的连续性，不会对用户体验造成影响。

[0054] (2)本发明所述的基于DirectShow支持可变编码的方法及系统，使得音视频数据中编码信息格式发生变化时，新的音视频数据解码后重采样输出格式与先前设定的输出格式一致，节省了新解码器和渲染器的链接时间，减少了用户等待的时间，提升了用户体验。

[0055] 当然，实施本发明的任一产品必不特定需要同时达到以上所述的所有技术效果。

[0056] 通过以下参照附图对本发明的示例性实施例的详细描述，本发明的其它特征及其优点将会变得清楚。

附图说明

[0057] 被结合在说明书中并构成说明书的一部分的附图示出了本发明的实施例，并且连同其说明一起用于解释本发明的原理。

[0058] 图1为现有技术中基于DirectShow进行音视频解码播放的系统结构示意图；

[0059] 图2为现有技术中基于DirectShow进行音视频可变编码播放的方法流程示意图；

[0060] 图3为本发明中基于DirectShow进行音视频解码播放的系统结构示意图；

[0061] 图4为本发明所述的基于DirectShow支持可变编码的方法流程示意图；

[0062] 图5为本发明中所述音视频数据的格式结构示意图；

[0063] 图6为本发明实施例2所述的基于DirectShow支持可变编码的方法的一个可选实施例的流程示意图；

[0064] 图7为采用FFmpeg进行音视频数据解码和重采样的流程示意图；

[0065] 图8为基于DirectShow进行音视频播放时使用FFmpeg进行解码和重采样的流程示意图；

[0066] 图9为本实施例所述的基于DirectShow支持可变编码的方法的一个具体实例的初始化流程示意图；

[0067] 图10为本发明实施例3所述的基于DirectShow支持可变编码的方法的一个具体实例中未检测到编码格式发生变化时的流程示意图；

[0068] 图11为本发明实施例3所述的基于DirectShow支持可变编码的方法的一个具体实例中检测到编码格式发生变化时的流程示意图；

[0069] 图12为本发明所述的基于DirectShow支持可变编码的系统结构示意图。

具体实施方式

[0070] 现在将参照附图来详细描述本发明的各种示例性实施例。应注意到：除非另外具体说明，否则在这些实施例中阐述的部件和步骤的相对布置、数字表达式和数值不限制本发明的范围。

[0071] 以下对至少一个示例性实施例的描述实际上仅仅是说明性的，决不作为对本发明及其应用或使用的任何限制。

[0072] 对于相关领域普通技术人员已知的技术、方法和设备可能不作详细讨论，但在适当情况下，所述技术、方法和设备应当被视为说明书的一部分。

[0073] 在这里示出和讨论的所有例子中，任何具体值应被解释为仅仅是示例性的，而不是作为限制。因此，示例性实施例的其它例子可以具有不同的值。

[0074] 应注意到：相似的标号和字母在下面的附图中表示类似项，因此，一旦某一项在一个附图中被定义，则在随后的附图中不需要对其进行进一步讨论。

[0075] 实施例1

[0076] 如图3及图4所示，图3为本发明中基于DirectShow进行音视频解码播放的系统结构示意图；其中，分离器301中集成了分离单元311、视频解码器302、音频解码器303、视频重采样304及音频重采样305，音视频数据进行分离器处理后传输至视频渲染器306和音频渲染器307进行渲染播放。图4为本发明所述的基于DirectShow支持可变编码的方法流程示意图。本发明所述的基于DirectShow支持可变编码的方法，避免了解码器与渲染器之间的断开及重新链接带来的音视频播放不流畅的问题。本实施例中基于DirectShow支持可变编码的方法包括以下步骤：

[0077] 步骤401、基于DirectShow获取音视频数据依次进行分离、解码和重采样后传输至音、视频渲染器进行渲染播放，并记录当前的重采样输出格式参数；其中，所述音视频数据包括：编码信息、视频数据及音频数据。如图5所示，为音视频数据的格式结构示意图，包括：编码信息501、视频数据502及音频数据503。

[0078] 步骤402、从所述获取音视频数据中编码格式发生变化的编码信息中获取编码格式发生变化的音视频数据的音、视频解码器参数。

[0079] 步骤403、断开当前的音、视频解码器,根据所述编码格式发生变化的音、视频解码器参数创建编码格式发生变化的音视频数据的音、视频解码器。

[0080] 步骤404、断开当前的音、视频重采样器,根据所述编码格式发生变化的音、视频解码器参数创建编码格式发生变化的音视频数据的音、视频重采样器;并根据所述重采样输出格式参数设定所述编码格式发生变化的音视频数据的音、视频重采样输出格式参数。其中,重采样是指对无压缩数据进行格式或参数转换,如将RGB格式的视频转换为YUV格式的视频,或对音频进行采样率转换。

[0081] 步骤405、利用所述编码格式发生变化的音视频数据的音、视频重采样输出格式参数将所述编码格式发生变化的音视频数据的音、视频重采样器链接到所述音、视频渲染器。

[0082] 步骤406、对所述编码格式发生变化的音视频数据进行分离后采用所述编码格式发生变化的音视频数据的音、视频解码器进行解码;并利用所述编码格式发生变化的音视频数据的音、视频重采样器进行重采样后传输至所述音、视频渲染器进行渲染播放。

[0083] 本实施例所述的基于DirectShow支持可变编码的方法,在分离器中集成了解码和重采样功能,直接发送无压缩的音视频数据到渲染器。保证分离器的输出格式始终保持统一,不发生变化,因此分离器与渲染器之间不需要重新链接就能正常工作。保证了当编码格式或参数发生变化时,播放数据的连续性,不会对用户体验造成影响。并且,分离器内部的解码器和重采样还可以由非DirectShow提供(如使用FFmpeg或其他第三方解码器等)。此过程中所有操作都在分离器内部进行,不会对用户体验产生影响。

[0084] 实施例2

[0085] 如图6所示,为本发明实施例2所述的基于DirectShow支持可变编码的方法的一个可选实施例的流程示意图。本实施例在实施例1的基础上,描述了基于DirectShow发生音视频数据编码信息变化时,进行重新解码、重采样的具体内容。本实施例所述的基于DirectShow支持可变编码的方法的步骤如下:

[0086] 步骤601、基于DirectShow获取音视频数据进行分离后,根据所述音视频数据的编码信息获取音、视频轨信息(音、视频信息中包含有多个音、视频轨信息),并从所述音、视频轨信息中获取所述音视频数据相关的音、视频解码器参数;其中,所述音视频数据包括:编码信息、视频数据及音频数据。

[0087] 步骤602、根据所述音、视频解码器参数创建音、视频解码器,并设定所述音、视频解码器的输出格式参数。

[0088] 步骤603、根据所述音、视频解码器参数创建重采样器,并记录当前的重采样输出格式参数。

[0089] 步骤604、将重采样后的音、视频数据传输至音、视频渲染器进行渲染播放,并记录当前的重采样输出格式参数。

[0090] 步骤605、从所述获取音视频数据中编码格式发生变化的编码信息中获取编码格式发生变化的音视频数据的音、视频解码器参数。

[0091] 步骤606、断开当前的音、视频解码器,根据所述编码格式发生变化的音、视频解码器参数创建编码格式发生变化的音视频数据的音、视频解码器。

[0092] 步骤607、断开当前的音、视频重采样器,根据所述编码格式发生变化的音、视频解码器参数创建编码格式发生变化的音视频数据的音、视频重采样器,并根据所述重采样输

出格式参数设定所述编码格式发生变化的音视频数据的音、视频重采样输出格式参数；优选地，设定所述编码格式发生变化的音视频数据的音、视频重采样输出格式参数与所述重采样输出格式参数一致。

[0093] 步骤608、利用所述编码格式发生变化的音视频数据的音、视频重采样输出格式参数将所述编码格式发生变化的音视频数据的音、视频重采样器链接到所述音、视频渲染器。

[0094] 步骤609、从所述获取音视频数据中编码格式未发生变化的编码信息中获取编码格式未发生变化的音视频数据的音、视频解码器参数。

[0095] 步骤610、对所述编码格式未发生变化的音视频数据进行分离后传输至利用所述编码格式未发生变化的音视频数据的音、视频解码器参数链接的编码格式未发生变化的音视频数据的音、视频解码进行解码；并采用编码格式未发生变化的音视频数据的音、视频数据的重采样器进行重采样后传输至所述音、视频渲染器进行渲染播放。

[0096] 优选地，本实施例中所述的解码和重采样由非DirectShow的第三方解码器提供，如使用FFmpeg或其他第三方解码器，FFmpeg是一种具有音视频解码、重采样等功能的开源库。本实施例采用FFmpeg为例进行说明，如图7所示，为采用FFmpeg进行音视频数据解码和重采样的流程示意图，包括如下步骤：

[0097] 步骤701、输入音视频数据；

[0098] 步骤702、判断是否存在音视频轨；

[0099] 步骤703、存在音视频轨时，获取该音视频轨的编码信息；

[0100] 步骤704、根据获取的编码信息创建FFmpeg解码器；

[0101] 步骤705、设定FFmpeg解码器的输出参数；

[0102] 步骤706、创建FFmpeg重采样；

[0103] 步骤707、记录重采样格式参数；

[0104] 步骤708、创建输出传输功能单元(pin)，并继续判断是否存在下一音视频轨；

[0105] 步骤709、不存在音视频轨时，结束音视频数据解码和重采样。

[0106] 如图8为基于DirectShow进行音视频播放时，使用FFmpeg进行解码和重采样的流程示意图，其具体步骤如下：

[0107] 步骤801、开始输入音视频数据；

[0108] 步骤802、依次读取音视频数据；

[0109] 步骤803、判断是否读取到音视频数据；

[0110] 步骤804、若未读取到音视频数据，则结束FFmpeg解码和重采样；

[0111] 步骤805、若读取到音视频数据，比较该音视频数据编码是否发生变化；

[0112] 步骤806、若音视频数据编码未发生变化，则进行FFmpeg解码；

[0113] 步骤807、进行FFmpeg重采样；

[0114] 步骤808、FFmpeg重采样后输出数据进行渲染并进行下一数据读取；

[0115] 步骤809、若音视频数据编码发生变化，则关闭FFmpeg解码器；

[0116] 步骤810、关闭FFmpeg重采样；

[0117] 步骤811、创建新的FFmpeg解码器；

[0118] 步骤812、创建新的FFmpeg重采样；

[0119] 步骤813、记录FFmpeg重采样格式参数，并进行FFmpeg解码。

[0120] 实施例3

[0121] 如图9所示,为本实施例所述的基于DirectShow支持可变编码的方法的一个具体实例的初始化流程示意图。本实施例所述的基于DirectShow支持可变编码的方法包括如下步骤:

[0122] 音视频数据的初始化步骤:

[0123] 步骤901、获取视频解码器类型为H.264参数为1920*1080。其中,当前播放的视频格式为H.264,1920*1080,音频格式为mp3,2声道16bit 44.1k。

[0124] 步骤902、创建H.264解码器,设定解码器输出格式为YV12,1920*1080。

[0125] 步骤903、创建视频重采样,设定输出格式为RGB32,1920*1080,并记录当前格式为H.264,1920*1080,重采样输出格式为RGB32,1920*1080;创建视频输出传输功能单元(Pin),并指定格式为RGB32,1920*1080,以后以此格式为视频输出传输功能单元格式标准。

[0126] 步骤904、创建mp3解码器,设定解码器输出格式为PCM 2声道,16bit 44.1k。

[0127] 步骤905、创建音频重采样,设定输出格式为PCM 2声道,16bit 44.1k,并记录当前格式为mp3,2声道16bit 44.1k,重采样输出格式为PCM,2声道16bit 44.1k;创建音频输出Pin,并指定格式为PCM 2声道16bit 44.1k。以后以此格式为标准。

[0128] 如图10所示,为本实施例所述的基于DirectShow支持可变编码的方法的一个具体实例中未检测到编码格式发生变化时的流程示意图。基于DirectShow进行播放时音视频数据编码格式未发生变化的流程步骤如下:

[0129] 步骤1001、读取视频数据,格式为H.264,1920*1080,检测到音视频数据编码格式未发生变化。

[0130] 步骤1002、解码视频数据,输出为YV12,1920*1080。

[0131] 步骤1003、重采样视频数据,输出为RGB32,1920*1080,并通过Pin输出视频数据,格式为RGB32 1920*1080。

[0132] 步骤1004、读取音频数据,格式为mp3,2声道16bit 44.1k,检测到音频数据编码格式及参数未发生变化。

[0133] 步骤1005、解码音频数据,输出为PCM,2声道16bit 44.1k。

[0134] 步骤1006、重采样音频数据,输出为PCM,2声道16bit 44.1k,通过Pin输出音频数据,格式为PCM,2声道16bit 44.1k。

[0135] 如图11所示,为本实施例所述的基于DirectShow支持可变编码的方法的一个具体实例中检测到编码格式发生变化时的流程示意图。播放一段时间后,检测到音视频数据编码格式发生变化,视频格式变为Divx,720*576,音频格式变为aac,6声道16bit 48k,此时按照本发明方法进行基于DirectShow播放的步骤如下:

[0136] 步骤1101、读取音视频数据中的编码信息,检测到视频编码格式及参数变为Divx,720*576,与原始的H.264,1920*1080不同。

[0137] 步骤1102、释放原H.264解码器,创建Divx解码器,设定视频解码器输出格式为RGB 720*576。

[0138] 步骤1103、释放原视频重采样,并创建新视频重采样,设定输出格式为RGB32,1920*1080,并记录当前视频格式为Divx 720*576。

[0139] 步骤1104、检测到音频编码格式及参数aac,6声道16bit 48k,与原始的mp3,2声道

16bit 44.1k不同。

[0140] 步骤1105、释放原mp3解码器,创建aac解码器,并设定音频解码器输出格式为PCM 6声道16bit 48k。

[0141] 步骤1106、释放原音频重采样,创建新音频重采样,设定输出格式为PCM 2声道 16bit 44.1k,并记录当前音频格式为aac 6声道16bit 48k。

[0142] 步骤1107、读取格式为Divx 720*576的视频数据,解码后输出格式为RGB 720*576。

[0143] 步骤1108、对解码后输出的视频数据进行重采样,输出格式为RGB32 1920*1080,并通过Pin输出视频数据,格式为RGB32 1920*1080。

[0144] 步骤1109、读取格式为aac,6声道16bit 48k的音频数据,进行解码后输出格式为PCM,6声道16bit 48k。

[0145] 步骤1110、对解码后的音频数据进行重采样,输出格式为PCM 2声道16bit 44.1k,通过Pin输出音频数据,格式为PCM,2声道16bit 44.1k。

[0146] 实施例4

[0147] 如图12所示,为本发明所述的基于DirectShow支持可变编码的系统结构示意图。本实施例的基于DirectShow支持可变编码的系统用于实现上述实施例中的基于DirectShow支持可变编码的方法,该系统包括:播放器1201、获取模块1202、解码器创建模块1203、重采样创建模块1204及链接模块1205。

[0148] 其中,所述播放器1201,与所述获取模块1202相藕接,用于基于DirectShow获取音视频数据依次进行分离、解码和重采样后传输至音、视频渲染器进行渲染播放,并记录当前的重采样输出格式参数;其中,所述音视频数据包括:编码信息、视频数据及音频数据;

[0149] 所述获取模块1202,与所述播放器1201及解码器创建模块1203相藕接,用于从所述获取音视频数据中编码格式发生变化的编码信息中获取编码信息发生变化的音视频数据的音、视频解码器参数;

[0150] 所述解码器创建模块1203,与所述获取模块1202及重采样创建模块1204相藕接,用于根据所述编码格式发生变化的音、视频解码器参数创建编码格式发生变化的音视频数据的音、视频解码器;

[0151] 所述重采样创建模块1204,与所述解码器创建模块1203及链接模块1205相藕接,用于根据所述编码格式发生变化后音、视频解码器参数创建编码格式发生变化的音视频数据的音、视频重采样器;

[0152] 并根据所述重采样输出格式参数设定所述编码格式发生变化的音视频数据的音、视频重采样输出格式参数。

[0153] 所述链接模块1205,与所述重采样创建模块1204及播放器1201相藕接,用于利用所述编码格式发生变化的音视频数据的音、视频重采样输出格式参数将所述编码格式发生变化的音视频数据的音、视频重采样器链接到所述音、视频渲染器。

[0154] 所述播放器1201,还与所述链接模块1205相藕接,用于对所述编码格式发生变化的音视频数据进行分离后采用所述编码格式发生变化的音视频数据的音、视频解码器进行解码;并利用所述编码格式发生变化的音视频数据的音、视频重采样器进行重采样后传输至所述音、视频渲染器进行渲染播放。

- [0155] 进一步地,所述播放器1201,还用于:
- [0156] 基于DirectShow获取音视频数据进行分离后,根据所述音视频数据的编码信息获取音、视频轨信息,并从所述音、视频轨信息中获取所述音视频数据相关的音、视频解码器参数;
- [0157] 根据所述音、视频解码器参数创建音、视频解码器,并设定所述音、视频解码器的输出格式参数;
- [0158] 根据所述音、视频解码器参数创建重采样器,并记录当前的重采样的输出格式参数。
- [0159] 所述获取模块1202还用于:
- [0160] 从所述获取的音视频数据中编码信息格式未发生变化的编码信息中获取编码格式未发生变化的音视频数据的音、视频解码器参数;
- [0161] 所述解码器创建模块1203还用于:对所述编码格式未发生变化的音视频数据进行分离后传输至利用所述编码格式未发生变化的音视频数据的音、视频解码器参数链接的编码格式未发生变化的音视频数据的音、视频解码进行解码;
- [0162] 所述重采样创建模块1204还用于:采用编码格式未发生变化的音视频数据的音、视频数据的重采样器对解码后的编码格式未发生变化的音、视频数据进行重采样;
- [0163] 所述播放器1201还用于:将重采样后的编码格式未发生变化的音视频数据的音、视频数据传输至所述音、视频渲染器进行渲染播放。
- [0164] 所述重采样创建模块1203还用于:
- [0165] 设定所述编码格式发生变化的音视频数据的音、视频重采样输出格式参数与所述重采样输出格式参数一致。
- [0166] 优选地,本实施中的所述编码格式发生变化的音视频数据的解码器和编码格式发生变化的音视频数据的重采样器由非DirectShow的第三方解码器提供,例如,通过FFmpeg提供编码格式发生变化的音视频数据的解码和重采样。
- [0167] 通过以上各个实施例可知,本发明的基于DirectShow支持可变编码的方法及系统存在的有益效果是:
- [0168] (1)本发明所述的基于DirectShow支持可变编码的方法及系统,通过设置重采样后的输出格式一致,使得音视频数据中编码信息格式发生变化时,新的音视频数据解码后重采样输出格式与先前设定的输出格式一致,不需要断开并更新现有的解码器和渲染器,保证了音频数据播放的连续性,不会对用户体验造成影响。
- [0169] (2)本发明所述的基于DirectShow支持可变编码的方法及系统,使得音视频数据中编码信息格式发生变化时,新的音视频数据解码后重采样输出格式与先前设定的输出格式一致,节省了新解码器和渲染器的链接时间,减少了用户等待的时间,提升了用户体验。
- [0170] 本领域内的技术人员应明白,本发明的实施例可提供为方法、装置、或计算机程序产品。因此,本发明可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且,本发明可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质(包括但不限于磁盘存储器、CD-ROM、光学存储器等)上实施的计算机程序产品的形式。
- [0171] 虽然已经通过例子对本发明的一些特定实施例进行了详细说明,但是本领域的技

术人员应该理解,以上例子仅是为了进行说明,而不是为了限制本发明的范围。本领域的技术人员应该理解,可在不脱离本发明的范围和精神的情况下,对以上实施例进行修改。本发明的范围由所附权利要求来限定。

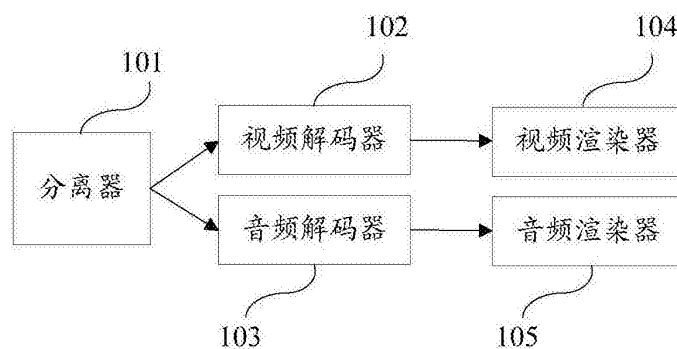


图1

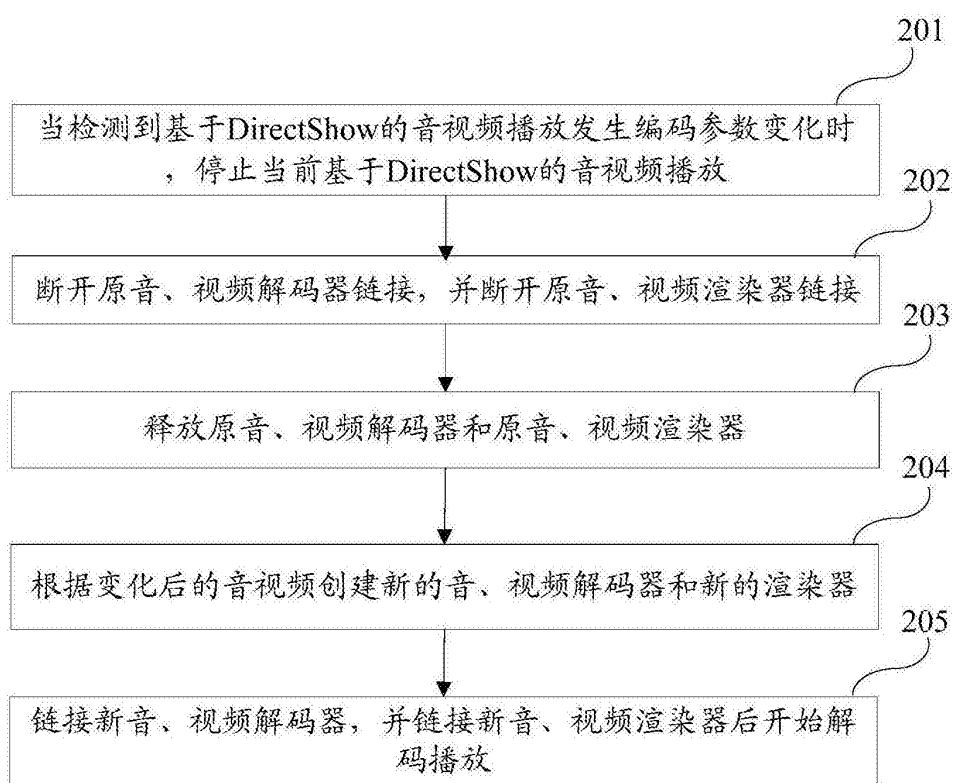


图2

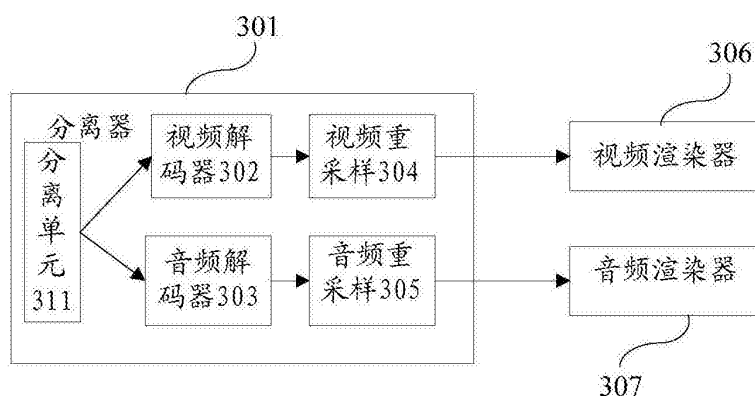


图3

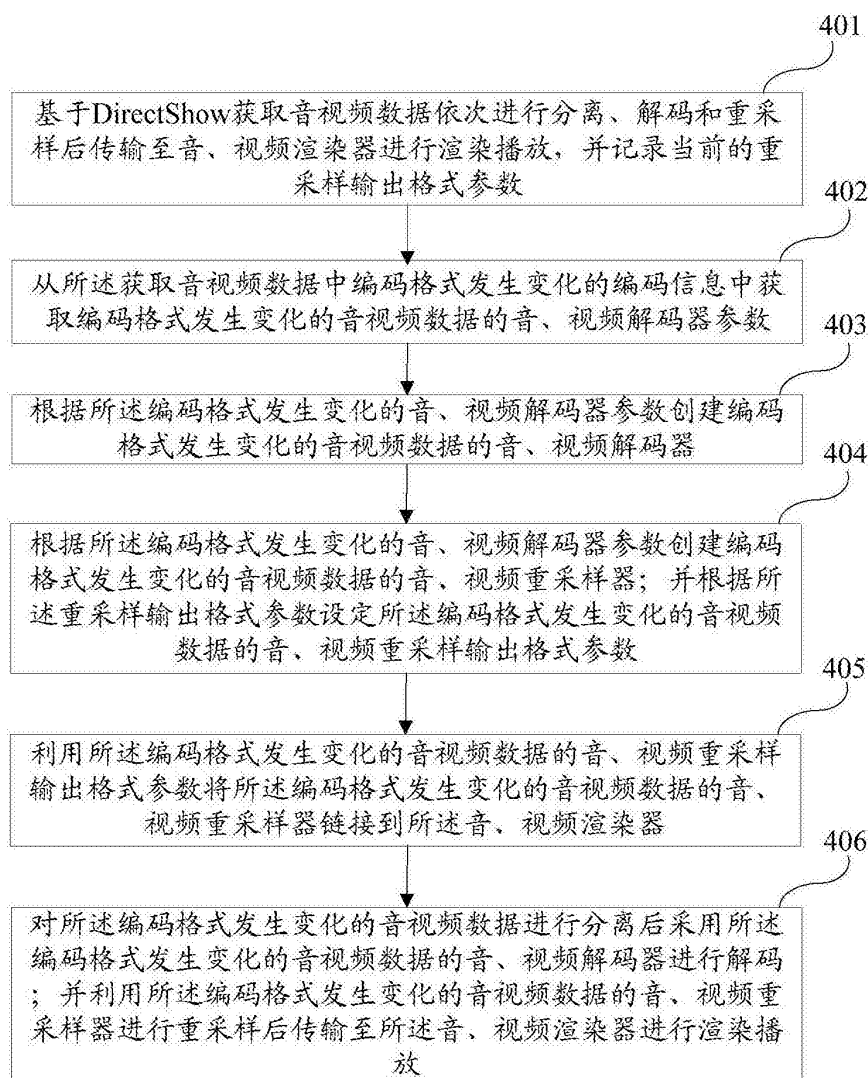


图4



图5

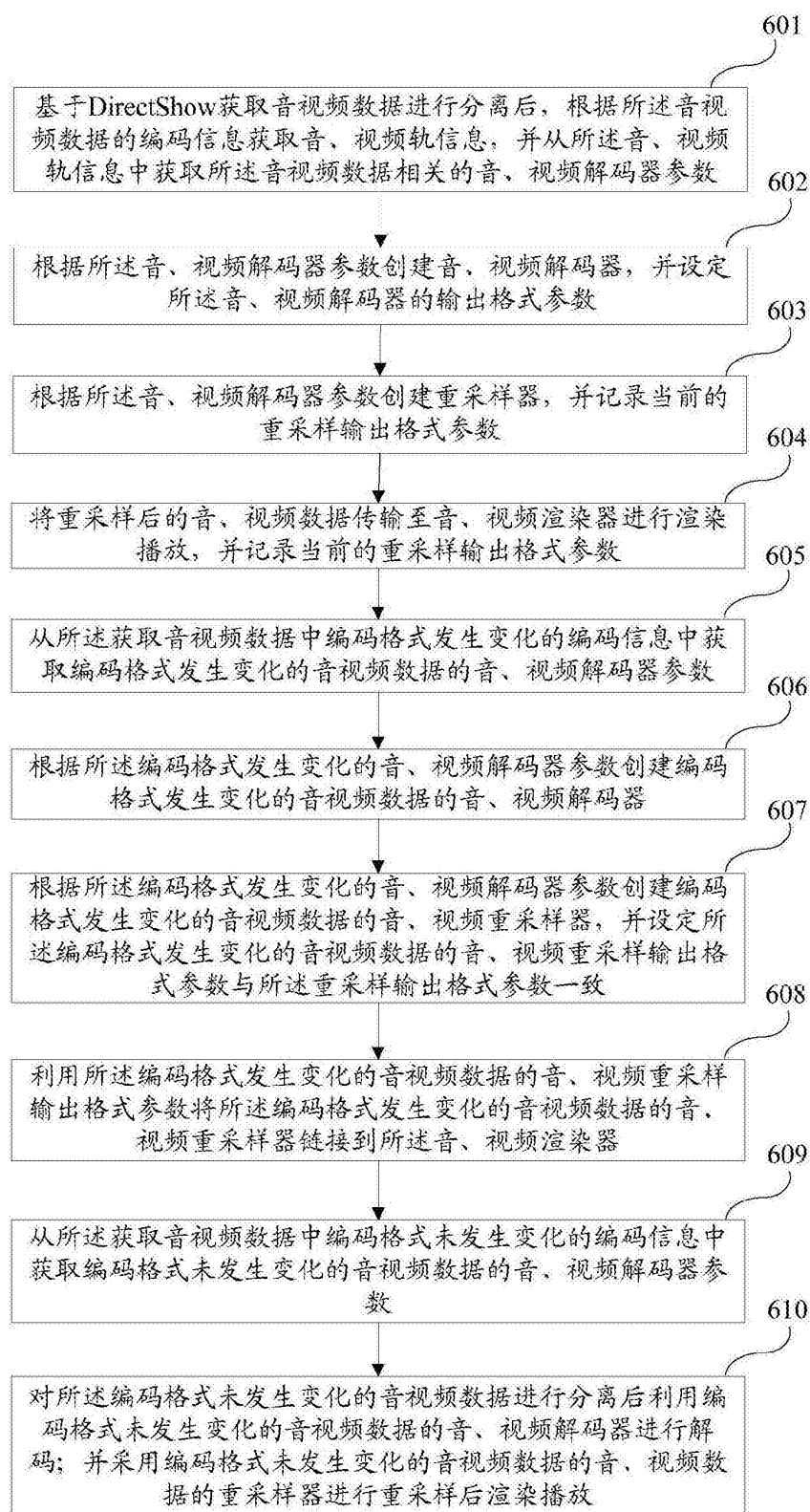


图6

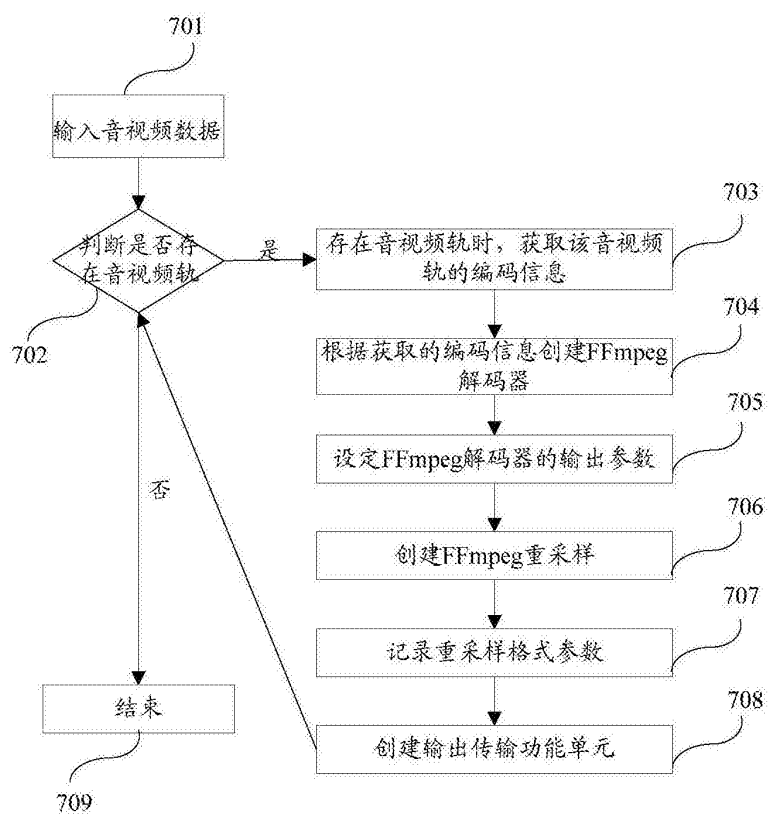


图7

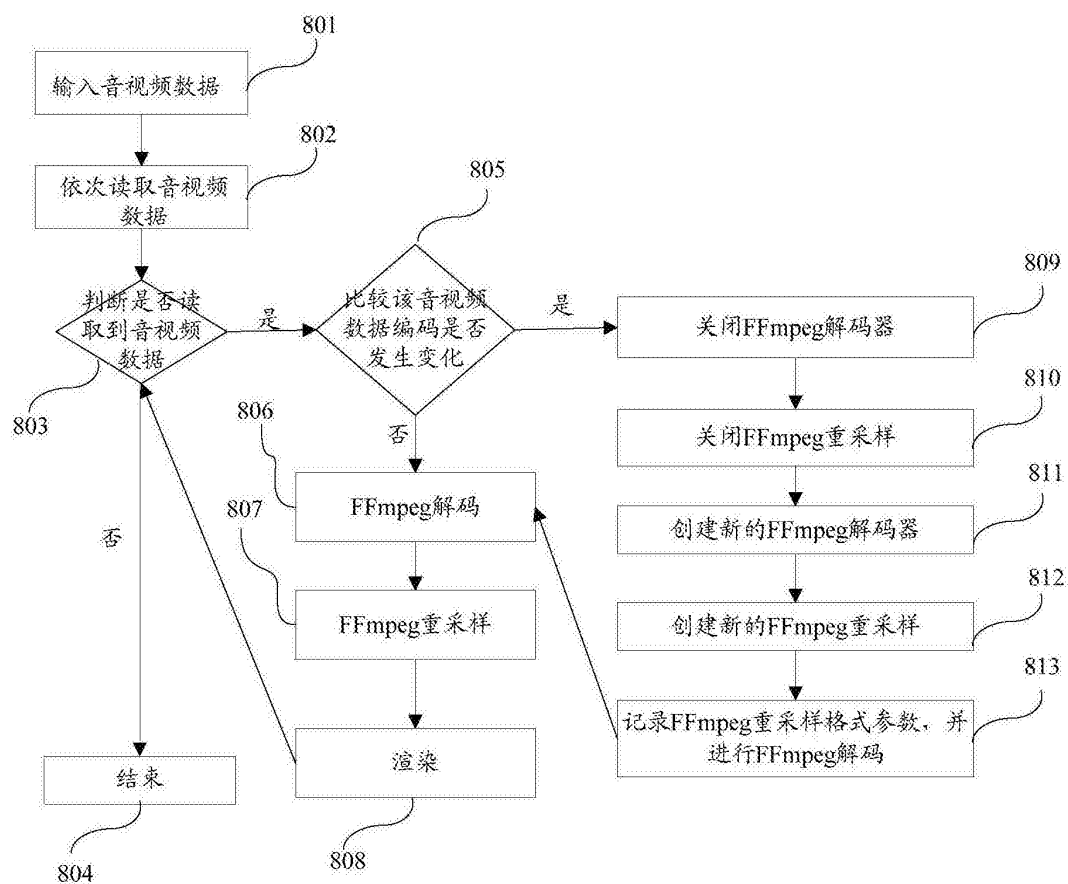


图8

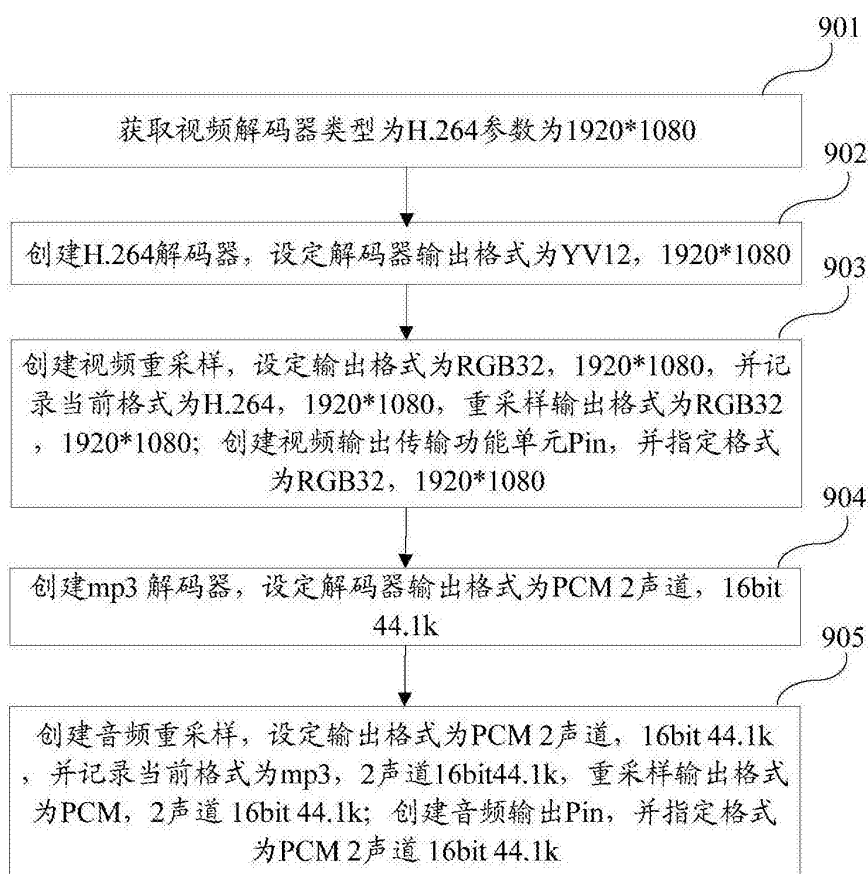


图9

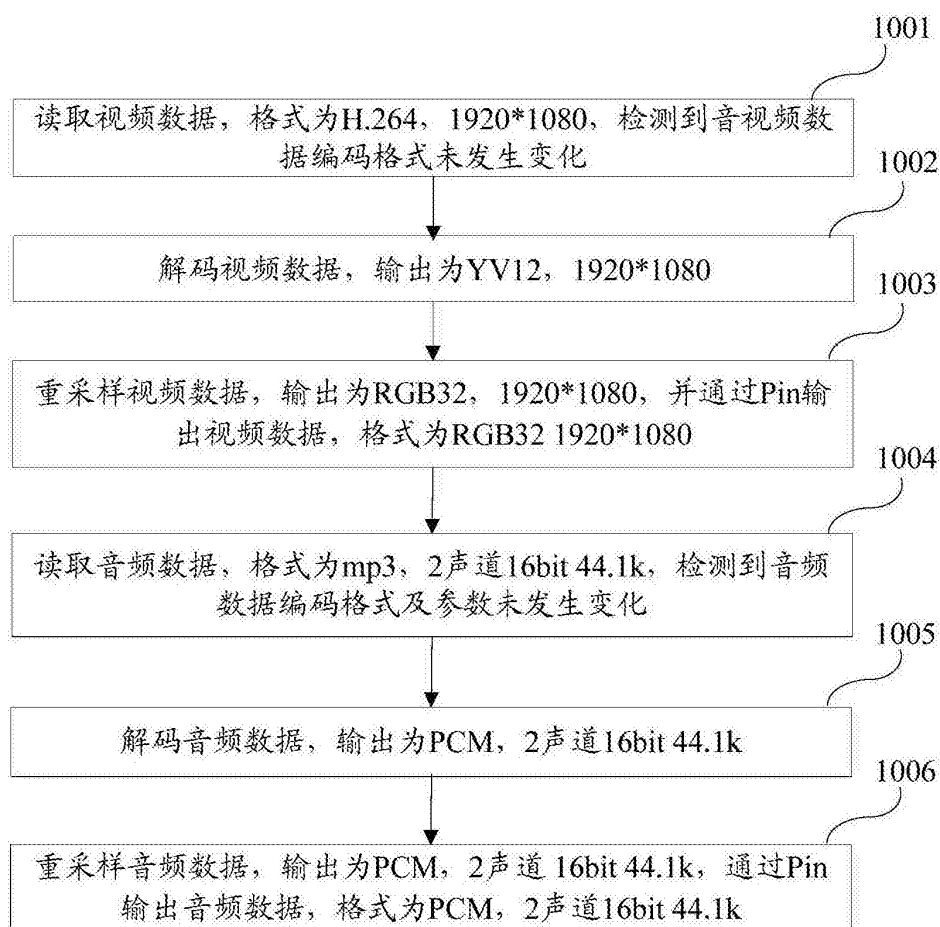


图10

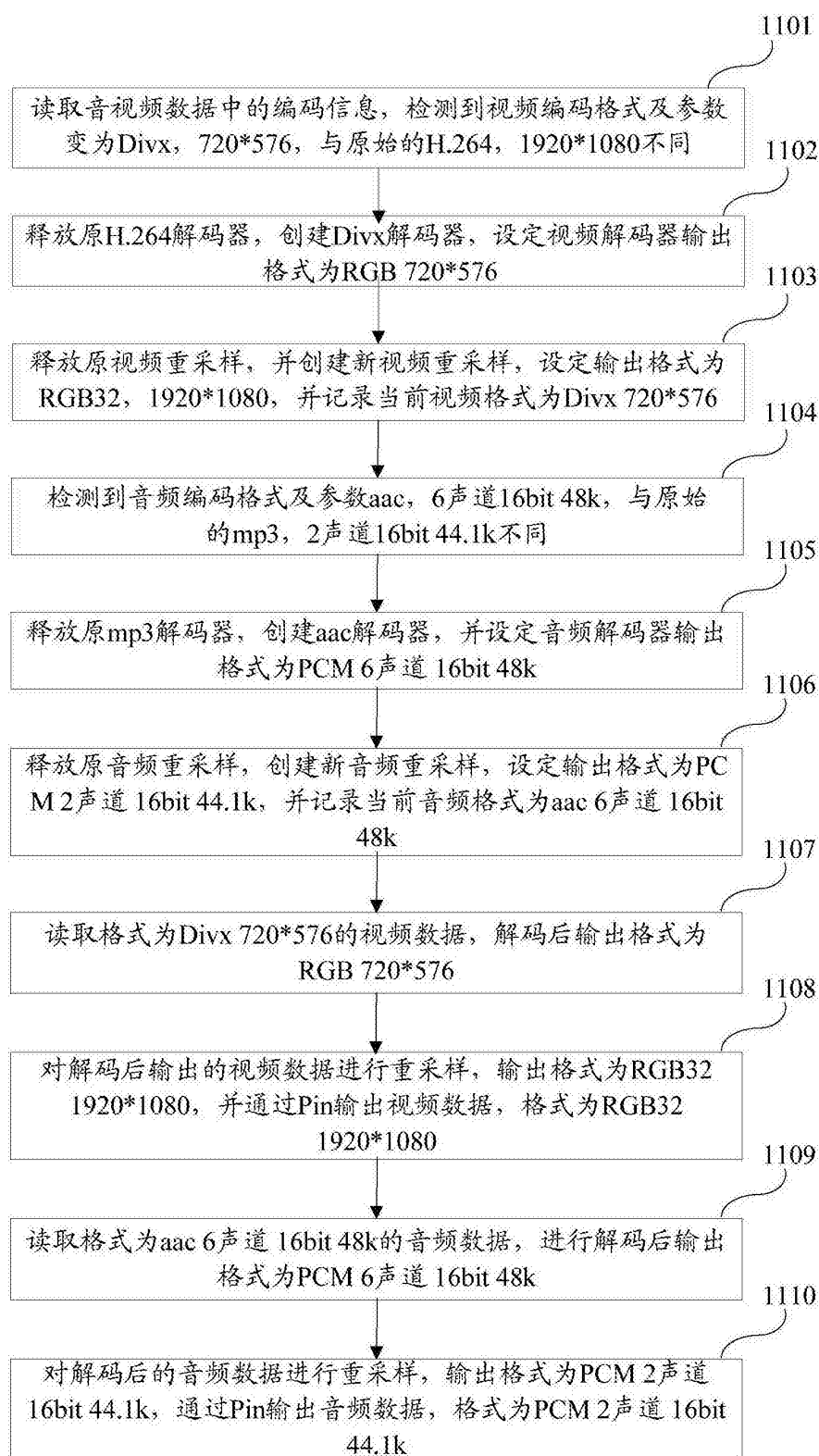


图11

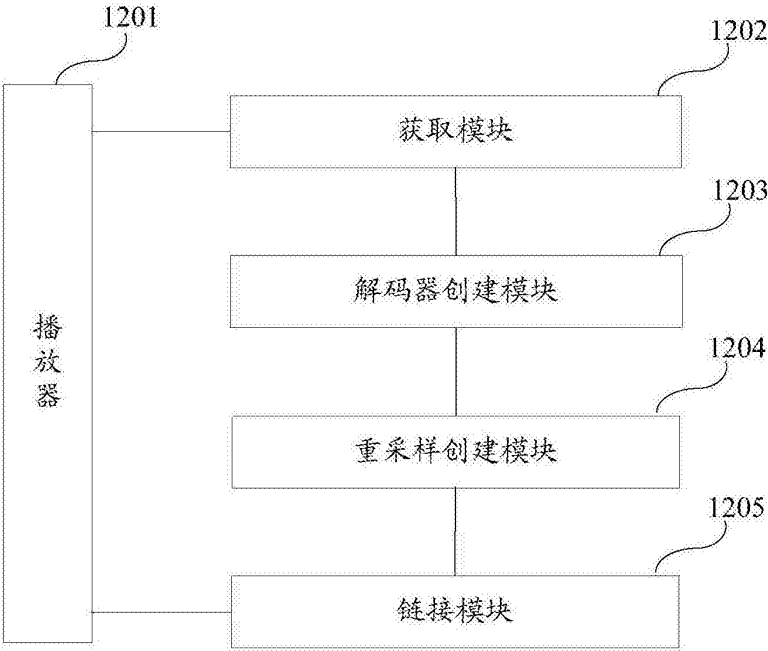


图12