

DECODER CONFIGURATION METHOD AND APPARATUS, AND MEDIUM AND ELECTRONIC DEVICE

RELATED APPLICATION

[0001] This application claims priority to Chinese Patent Application No. 2021102082858, entitled "DECODER CONFIGURATION METHOD, APPARATUS, MEDIUM AND ELECTRONIC DEVICE" and filed with the National Intellectual Property Administration, PRC on February 24, 2021, and claims priority to Chinese Patent Application No. 202111647541X, entitled "DECODER CONFIGURATION METHOD, APPARATUS, MEDIUM AND ELECTRONIC DEVICE" and filed with the National Intellectual Property Administration, PRC on December 29, 2021, which are incorporated herein by reference in its entirety.

FIELD OF THE TECHNOLOGY

[0002] The present disclosure relates to the field of computer technologies, and in particular to a decoder configuration method, a decoder configuration apparatus, a computer-readable medium and an electronic device.

BACKGROUND OF THE DISCLOSURE

[0003] A phenomenon of decoding frame hoarding may occur to a decoding chip of a terminal device due to problems about properties and power consumption. Decoding frame hoarding means that video frames are cached in an internal cache area of a decoder, and video images cannot be outputted until the internal cache area is fully filled. In a low-latency service scenario (such as cloud gaming), a video image needs to be outputted as soon as possible after one video frame is decoded. Generally, parameters of the decoding chip are configured into a real-time communication mode or low-latency mode to solve the problem of decoding frame hoarding.

[0004] The information disclosed in the above background part is used only for enhancing the understanding of the background of this application, and therefore may include information that does not constitute the related art known to a person of ordinary skill in the art.

SUMMARY

[0005] The present disclosure aims to provide a decoder configuration method, a decoder configuration apparatus, a computer-readable medium and an electronic device to solve the

technical problems about not high compatibility and low decoding efficiency at least to a certain degree.

[0006] Other characteristics and advantages of the present disclosure become apparent from the following detailed description, or may be learned in part through the practice of the present disclosure.

[0007] In one aspect, the embodiment of the present disclosure provides a decoder configuration method, including: acquiring at least two compatibility grade configurations used for configuring decoding parameters of a decoder, the different compatibility grade configurations being different in priority level, decoding property and decoder compatibility;

[0008] selecting a compatibility grade configuration from the at least two compatibility grade configurations according to the priority levels from high to low; and for the current selected compatibility grade configuration:

[0009] configuring the decoder according to the compatibility grade configuration;

[0010] starting the decoder, and performing decoding abnormality detection processing on the decoder to obtain an abnormality detection result; and

[0011] performing hardware decoding processing on a to-be-decoded video according to the compatibility grade configuration to obtain a decoding processing result in response to the abnormality detection result indicating that the decoder is normal, and determining that the current compatibility grade configuration is applicable to the decoder in response to the decoding processing result indicating that decoding is successful.

[0012] In another aspect, the embodiment of the present disclosure provides a decoder configuration apparatus, including: a decoder configuration module, an abnormality detection module and a decoding processing module.

[0013] The decoder configuration module is configured to acquire at least two compatibility grade configurations used for configuring decoding parameters of a decoder, the different compatibility grade configurations being different in priority level, decoding property and decoder compatibility; and select a compatibility grade configuration from the at least two compatibility grade configurations according to the priority levels from high to low.

[0014] For the current selected compatibility grade configuration: the decoder configuration module is configured to configure the decoder according to the compatibility grade configuration and start the decoder; the abnormality detection module is configured to perform decoding abnormality detection processing on the decoder to obtain an abnormality detection result; and the

decoding processing module is configured to perform hardware decoding processing on a to-be-decoded video according to the current selected compatibility grade configuration to obtain a decoding processing result in response to the abnormality detection result indicating the decoder being normal, and determine that the current compatibility grade configuration is applicable to the decoder in response to the decoding processing result indicating that decoding is successful.

[0015] In yet another aspect, the embodiment of the present disclosure provides a computer-readable medium, storing a computer program, the computer program, when executed by a processor, implementing the decoder configuration method in the foregoing technical solutions.

[0016] In yet another aspect, the embodiment of the present disclosure provides an electronic device, including: a processor; and a memory configured to store executable instructions of the processor. The processor is configured to execute the executable instructions to perform the decoder configuration method in the foregoing technical solutions.

[0017] It is to be understood that the foregoing general descriptions and the following detailed descriptions are merely for illustration and explanation purposes and are not intended to limit the present disclosure.

BRIEF DESCRIPTION OF THE DRAWINGS

[0018] FIG. 1 schematically illustrates a schematic architectural diagram of an exemplary system applying a technical solution of the present disclosure.

[0019] FIG. 2A-2B schematically illustrate a step flowchart of a decoder configuration method in some embodiments of the present disclosure.

[0020] FIG. 3 schematically illustrates a step flowchart of a rule configuration processing method in some embodiments of the present disclosure.

[0021] FIG. 4 schematically illustrates a step flowchart of a compatibility logic configuration method in some embodiments of the present disclosure.

[0022] FIG. 5 schematically illustrates a step flowchart of a grade logic configuration method in some embodiments of the present disclosure.

[0023] FIG. 6 schematically illustrates a step flowchart of a parameter grade matched processing method in some embodiments of the present disclosure.

[0024] FIG. 7 schematically illustrates a step flowchart of a method for determining a compatibility grade configuration applicable to a decoder in some embodiments of the present disclosure.

[0025] FIG. 8 schematically illustrates a step flowchart of a decoder configuration method for a terminal device not hoarding frames in some embodiments of the present disclosure.

[0026] FIG. 9 schematically illustrates a step flowchart of a decoder configuration method for a terminal device hoarding frames in some embodiments of the present disclosure.

[0027] FIG. 10A-10B schematically illustrate a structure block diagram of a decoder configuration apparatus in some embodiments of the present disclosure.

[0028] FIG. 11 schematically illustrates a schematic structural diagram of a computer system of an electronic device applicable to implementing an embodiment of the present disclosure.

DESCRIPTION OF EMBODIMENTS

[0029] Exemplary implementations are now described more comprehensively with reference to the accompanying drawings. However, the examples of implementations may be implemented in multiple forms, and it should not be understood as being limited to the examples of implementations described herein. Conversely, the implementations are provided to make the present disclosure more comprehensive and complete, and comprehensively convey the idea of the examples of the implementations to a person skilled in the art.

[0030] In addition, the described features, structures, or characteristics may be combined in one or more embodiments in any appropriate manner. In the following descriptions, more specific details are provided to provide a comprehensive understanding of the embodiments of the present disclosure. However, a person skilled in the art is to be aware of that, the technical solutions in the present disclosure may be implemented without one or more of the particular details, or other methods, unit, apparatus, or step may be adopted. In other cases, well-known methods, apparatuses, implementations, or operations are not shown or described in detail, to avoid obscuring the aspects of the present disclosure.

[0031] The block diagrams shown in the accompanying drawings is merely a functional entity and does not necessarily correspond to a physically independent entity. To be specific, such functional entities may be implemented in the form of software, or implemented in one or more hardware modules or integrated circuits, or implemented in different networks and/or processor apparatuses and/or microcontroller apparatuses.

[0032] The flowcharts shown in the accompanying drawings are merely examples for descriptions, may not include all content and operations/steps, and may not be performed in the described orders either. For example, some operations/steps may be further divided, while some

operations/steps may be combined or partially combined. Therefore, an actual execution order may change according to an actual case.

[0033] In related technologies in the art, factors such as properties and power consumption will be integrated for designing decoders of most of current terminal devices such as mobile phones, and thus decoding chips are uncertain to immediately output a video image after a video frame is decoded in the decoders. For example, part of the decoding chips will have a limitation in decoding frame hoarding.

[0034] Decoding frame hoarding means that video frames are cached in internal cache areas of the decoders, and the video images cannot be outputted until the internal cache areas are fully filled.

[0035] But, in a low-latency video service scenario, a video image needs to be outputted as soon as possible after one video frame is decoded.

[0036] Generally, a low-latency scheme satisfies requirements of the low-latency service scenario basically by configuring corresponding parameters of an Android system, namely by setting the parameters into a real-time communication mode or a low-latency mode.

[0037] But, compatibility of the Android version is not high. For example, a priority parameter is introduced by Android 6.0, and a low-latency parameter is introduced by Android 11.0. The two parameters are realized by part of chip vendors in a private interface manner before being officially listed into an Android official brochure, but the problem about compatibility possibly still exists for other most of chips due to direct configuration. Taking a chip MSM8916 as an example, under the configuration in the real-time communication mode, the configuration will fail when a target frame rate is set to 60.

[0038] Moreover, the situation that part of chip vendors has private interfaces is not considered. In addition, chip decoding efficiency may also be higher through configuration of other parameters according to official documents. Configuring other parameters also needs to consider the problem about compatibility.

[0039] Based on the problems in the above scheme, the present disclosure newly provides a decoder configuration method, a decoder configuration apparatus, a computer-readable medium and an electronic device based on a cloud technology.

[0040] The cloud technology is a hosting technology that unifies a series of resources such as hardware, software, and networks in a wide area network or a local area network to implement computing, storage, processing, and sharing of data.

[0041] A cloud technology is a collective name of a network technology, an information technology, an integration technology, a management platform technology, an application technology, and the like based on an application of a cloud computing service mode, and may form a resource pool. Usage is based on a requirement and is flexible and convenient. A cloud computing technology becomes an important support. A background service of a technical network system requires a large amount of computing and storage resources, such as a video website, an image website, and more portal websites. As the Internet industry is highly developed and applied, each article may have its own identifier in the future and needs to be transmitted to a background system for logical processing. Data at different levels is separately processed, and data in various industries requires strong system support and this can only be implemented through cloud computing.

[0042] Cloud gaming may also be referred to as gaming on demand, which is an online game technology based on a cloud computing technology. High-quality games can run at a thin client with relatively-limited graphic processing and data operation capabilities through the cloud gaming technology. In a cloud gaming scenario, a game runs in a cloud server instead of a player game terminal, and a game scene is rendered into video and audio stream through the cloud server and transmitted to the player game terminal by a network. The player game terminal only needs to have a basic capability of playing streaming media and a capability of acquiring and sending player input instructions to the cloud server without powerful graphic operating and data processing capabilities.

[0043] Cloud Computing Education (CCEDU) refers to an education platform service applied on the basis of a cloud computing service model. On a cloud platform, all educational institutions, training institutions, recruit service institutions, propaganda institutions, industry associations, management institutions, industry media, legal institutions and the like are centralized into a resource pool through cloud integration, and the resources are mutually displayed, interact with one another and communicate according to needs to reach an agreement, thereby reducing educational costs and improving efficiency.

[0044] A cloud conference is an efficient, convenient and low-cost conference form based on the cloud computing technology. A user can rapidly, efficiently and synchronously share audios, data files and videos only by performing simple and usable operation via an Internet interface, while the user is assisted by a cloud conference service provider to operate data transmission, processing and other complex technologies in the conference.

[0045] Current domestic cloud conferences mainly focus on service contents with Software as a Service (SaaS) model as a subject, including service forms via telephones, network, videos and the like, and a video conference based on cloud computing is referred to as the cloud conference.

[0046] In the age of cloud conference, data transmission, processing and storage are all processed by computer resources of video conference vendors so that the user can perform the efficient teleconference only by opening a browser and logging in a corresponding interface without purchasing expensive hardware and installing complex software.

[0047] A cloud conference system supports multi-server dynamic cluster deployment and provides a plurality of high-performance servers, thereby greatly improving conference stability, security and usability. In recent years, the video conference is popular with many users because of significantly improving communication efficiency, reducing communication costs and upgrading an inner management level, and is widely applied to various fields such as government, army, traffic, transportation, finance, operators, education and enterprises. It is no doubt that the video conference has stronger attraction in convenience, rapidness and usability after applying cloud computing, which surely will motivate a new climax of video conference application.

[0048] In cloud gaming, cloud education, cloud conference and other low-latency video service scenarios based on the cloud technology, the decoder configuration method can solve the problem about decoding frame hoarding of the decoder and the problem about time delay in a decoder hardware decoding processing process, a decoding frame rate is higher, hardware rapidly performs decoding and exports frames, the requirements of the low-latency video service scenarios are satisfied, the problem about compatibility of different decoders is solved as well, and an optimal decoding property of the decoders is brought into play.

[0049] FIG. 1 illustrates a schematic diagram of an exemplary system architecture applying a technical solution of the present disclosure.

[0050] As shown in FIG. 1, the system architecture 100 may include a terminal 110, a network 120, and a server side 130. The terminal 110 and the server side 130 are connected by the network 120.

[0051] The terminal 110 may be a smartphone, a tablet computer, a notebook computer, a desktop computer, a smart speaker, a smartwatch, or the like, but is not limited thereto. The network 120 may be a communication medium capable of providing various connection types for communication links between the terminal 110 and the server side 130, such as a wire communication link, a wireless communication link or an optical cable, which is not limited by this

application. The server 130 may be an independent physical server, or may be a server cluster including a plurality of physical servers or a distributed system, or may be a cloud server providing basic cloud computing services, such as a cloud service, a cloud database, cloud computing, a cloud function, cloud storage, a network service, cloud communication, a middleware service, a domain name service, a security service, a content delivery network (CDN), big data, and an artificial intelligence platform.

[0052] Specifically, the terminal 110 acquires at least two compatibility grade configurations used for configuring decoding parameters of a decoder, the different compatibility grade configurations being different in priority level, decoding property and decoder compatibility; sequentially selects a compatibility grade configuration from the compatibility grade configurations according to the priority levels from high to low; and for the current selected compatibility grade configuration: configures the decoder according to the compatibility grade configuration; starts the decoder, and performs decoding abnormality detection processing on the decoder to obtain an abnormality detection result; and performs hardware decoding processing on a to-be-decoded video to obtain a decoding processing result according to the compatibility grade configuration based on the abnormality detection result indicating that the decoder is normal, and determines that the current compatibility grade configuration is applicable to the decoder according to the decoding processing result if the decoding processing result indicates that decoding is successful.

[0053] In addition, the decoder configuration method in the embodiment of the present disclosure may be applied to the terminal and may also be applied to the server side, which is not specially limited by the present disclosure. The embodiment of the present disclosure exemplarily introduces applying the decoder configuration method to the terminal 110.

[0054] The decoder configuration method, a decoder configuration apparatus, a computer-readable medium and an electronic device provided by the present disclosure are described in detail by combining specific implementations below.

[0055] FIG. 2A illustrates a flowchart of a decoder configuration method in some embodiments of the present disclosure. As shown in FIG. 2A, the method may include the following steps.

[0056] S21: Acquiring at least two compatibility grade configurations used for configuring decoding parameters of a decoder, the different compatibility grade configurations being different in priority level, decoding property and decoder compatibility.

[0057] S22: Sequentially selecting a compatibility grade configuration from the compatibility grade configurations according to the priority levels from high to low, and for the current selected compatibility grade configuration, performing step S23-step S25.

[0058] S23: Configuring the decoder according to the compatibility grade configuration.

[0059] S24: Starting the decoder, and performing decoding abnormality detection processing on the decoder to obtain an abnormality detection result.

[0060] S25: Performing hardware decoding processing on a to-be-decoded video to obtain a decoding processing result according to the compatibility grade configuration if the abnormality detection result indicates that the decoder is normal, and determining that the current compatibility grade configuration is applicable to the decoder if the decoding processing result indicates that decoding is successful.

[0061] In some embodiments, before step S21, following operation may be firstly performed so as to determine at least two compatibility grade configurations according to a practical decoding demand. Following operation is specifically included:

[0062] Acquiring decoding parameters of the decoder and determining a parameter configuration rule corresponding to the decoding demand.

[0063] Configuring the decoding parameters according to the parameter configuration rule to obtain a plurality of configuration results of the decoding parameters, different configuration results corresponding to different decoding properties.

[0064] Setting at least two compatibility grades according to the decoding parameters, the different compatibility grades being different in decoder compatibility.

[0065] Obtaining at least two compatibility grade configurations according to parameter grade matched processing of the at least two compatibility grades and the plurality of configuration results.

[0066] FIG. 2B schematically illustrates a step flowchart of a decoder configuration method in some embodiments of the present disclosure. As shown in FIG. 2B, the decoder configuration method may include the following steps:

[0067] S210: Acquiring decoding parameters of a decoder and determining a parameter configuration rule corresponding to a decoding demand.

[0068] S220: Configuring the decoding parameters according to the parameter configuration rule to obtain a plurality of configuration results of the decoding parameters and setting at least two

compatibility grades, the different configuration results corresponding to different decoding properties, and the different compatibility grades being different in decoder compatibility.

[0069] S230: Obtaining at least two compatibility grade configurations according to parameter grade matched processing of the at least two compatibility grades and the plurality of configuration results, and selecting one compatibility grade configuration from the at least two compatibility grade configurations.

[0070] S240: Configuring and starting the decoder according to the compatibility grade configuration, and performing decoding abnormality detection processing on the decoder to obtain an abnormality detection result.

[0071] S250: Performing hardware decoding processing on a to-be-decoded video to obtain a decoding processing result according to the compatibility grade configuration if the abnormality detection result indicates that the decoder is normal, and determining that the current compatibility grade configuration is applicable to the decoder if the decoding processing result indicates that decoding is successful.

[0072] In the exemplary embodiment of the present disclosure, on one hand, the decoding parameters are configured, so that the problem about decoding frame hoarding of the decoder and the problem about time delay in a decoder hardware decoding processing process are solved, a decoding frame rate is higher, hardware rapidly performs decoding and exports frames, and the requirements of a low-latency video service scenario are satisfied; and on the other hand, compatibility logic configuration and parameter grade matched processing are performed on the decoding parameters to obtain the compatibility grade configurations, thereby solving the problem about compatibility of different decoders, and bringing the optimal decoding property of the decoders into play.

[0073] Various steps in the decoder configuration method are described in detail below.

[0074] In step S210, the decoding parameters of the decoder are acquired, and the parameter configuration rule corresponding to the decoding demand is determined.

[0075] In the exemplary embodiments of the present disclosure, decoding is a process of restoring a coded bitstream into original represented contents, and corresponds to coding. Decoding may include software decoding and hardware decoding.

[0076] Software decoding refers to compiling a decoding algorithm on a software level and achieving a decoding purpose by performing computing and coded bitstream processing via a central processing unit. Hardware decoding refers to decoding the coded bitstream by a hardware

decoding apparatus included by the device. The hardware decoding apparatus may include a decoder such as decoding chip.

[0077] For example, when the decoder is a decoding chip of an Android system, decoding parameters may include parameters configurable by interfaces provided by the Android system, and the interfaces may include a setInteger interface and a setFeatureEnabled interface. The setInteger interface is configured to bind parameters with a mapping type being int (integer), an application method is setInteger (int position, int val), position is used for assigning an nth parameter, and val is used for assigning a corresponding parameter value; and the setFeatureEnabled interface is configured to add any to-be-added specific function, etc.

[0078] Corresponding to the setInteger interface and the setFeatureEnabled interface, the decoding parameters may be “priority”, “frame-rate”, “low-latency” and “operating-rate” and may also include corresponding parameters of private interfaces provided by chip vendors, such as “vendor.qti-...-low-latency.enable”, “vendor.hisi-...-low-latency-req” and “vendor. ...-low-latency.enable”. Low-latency may correspond to two parameters of the setInteger interface and the setFeatureEnabled interface respectively.

[0079] The to-be-decoded video corresponding to the decoder is capable of being subjected to hardware decoding processing by the decoder.

[0080] Furthermore, the parameter configuration rule corresponding to the decoding demand is determined. According to the parameter configuration rule, the decoding parameters may be correspondingly configured according to a practical need. For example, when the practical need is about solving the problem of real time of the to-be-decoded video, the problem may be solved from two aspects including a target frame rate in a real-time communication mode and low-latency removal of a frame hoarding behavior of the decoder, and thus, the parameter configuration rule may consider the decoding parameters from the two aspects, thereby enabling the decoding parameters to conform to the practical need.

[0081] Specifically, decoding parameters corresponding to the target frame rate in the real-time communication mode may include priority, frame-rate and operating-rate, and parameters corresponding to low-latency removal of the frame hoarding behavior of the decoder may include low-latency corresponding to the setInteger interface, low-latency corresponding to the setFeatureEnabled interface, and the corresponding parameters of the chip vendor private interfaces, such as “vendor.qti-...-low-latency.enable”, “vendor.hisi-...-low-latency-req” and “vendor. ...-low-latency.enable”.

[0082] In step S220, rule configuration processing is performed on the decoding parameters according to the parameter configuration rule to obtain a plurality of configuration results of the decoding parameters, and at least two compatibility grades are set.

[0083] In the exemplary embodiments of the present disclosure, after the parameter configuration rule is determined, rule configuration processing may be performed on the decoding parameters according to the parameter configuration rule.

[0084] In some embodiments, decoding parameters include frame rate setting parameters and decoding hoarded frame parameters, and parameter configuration results include frame rate configuration results and hoarded frame configuration results, and besides, may include a default configuration result in which no configuration is performed on the decoding parameters. FIG. 3 illustrates a step flowchart of a rule configuration processing method. As shown in FIG. 3, the method at least may include the following steps: S310: Configuring frame rate setting parameters according to a parameter configuration rule to obtain a plurality of frame rate configuration results.

[0085] The frame rate setting parameters include priority, frame-rate and operating-rate.

[0086] To make the frame rate setting parameters satisfy the requirements of a target frame rate in a real-time communication mode, priority may be set as 0, frame-rate is set to be 60-120, and operating-rate is set to be 120-240, thereby obtaining the plurality of frame rate configuration results corresponding to the frame rate setting parameters.

[0087] S320: Configuring decoding hoarded frame parameters according to the parameter configuration rule to obtain a plurality of hoarded frame configuration results.

[0088] The decoding hoarded frame parameters include low-latency corresponding to a setInteger interface, low-latency corresponding to a setFeatureEnabled interface, and corresponding parameters of chip vendor private interfaces, such as “vendor.qti-...-low-latency.enable”, “vendor.hisi-...-low-latency-req” and “vendor. ...-low-latency.enable”.

[0089] To make the decoding hoarded frame parameters solve the problem about low-latency removal of decoder frame hoarding, low-latency corresponding to the setInteger interface may be set as 1, low-latency corresponding to the setFeatureEnabled interface is set to be true, and the corresponding parameters of the chip vendor private interfaces, such as “vendor.qti-...-low-latency.enable”, “vendor.hisi-...-low-latency-req” and “vendor. ...-Low-latency.enable” are set to be 1, thereby obtaining the hoarded frame configuration results corresponding to the decoding hoarded frame parameters.

[0090] In the exemplary embodiment, a data basis may be provided for solving the problem about real time of hardware decoding processing by configuring the frame rate setting parameters and the decoding hoarded frame parameters according to the parameter configuration rule to obtain the corresponding plurality of frame rate configuration results and hoarded frame configuration results.

[0091] After the plurality of configuration results of the decoding parameters are obtained, compatibility logic configuration may be performed on the decoding parameters, namely, at least two compatibility grades are set, and the different compatibility grades are different in decoder compatibility. In other words, the different compatibility grades are different in compatibility for a decoder (such as a decoding chip).

[0092] In some embodiments, FIG. 4 illustrates a step flowchart of a compatibility grade configuration method. As shown in FIG. 4, the method at least may include the following steps: S410: Performing compatibility grade division processing to obtain at least two compatibility grades.

[0093] Specifically, 5 compatibility grades may be divided and include Ultimate, UltraFast, Fast, LowLatency and Normal. Decoding parameters may be configured according to the 5 compatibility grades in a subsequent process. In addition, compatibility grades different in number may be divided or the compatibility grades may be divided in other manners according to practical situations and demands, which is not specially limited by the exemplary embodiment.

[0094] S420: Performing grade logic configuration on at least two compatibility grades to obtain a compatibility detection rule.

[0095] In some embodiments, FIG. 5 illustrates a step flowchart of a grade logic configuration method. As shown in FIG. 5, the method at least may include the following steps: S510: Performing priority level configuration on at least two compatibility grades to obtain priority levels of the at least two compatibility grades.

[0096] An Ultimate level is set to have the best decoding capability and can bring the extreme decoding capability of decoders such as a decoding chip into play, and thus, the priority level is the highest; an UltraFast level, a Fast level and a LowLatency level are set to make choices for decoding parameters to adapt to the compatibility problem of a terminal with different decoders, thus, the UltraFast level, the Fast level and the LowLatency level may be set to the same in priority level but be lower than the Ultimate level in priority level, the priority level of the UltraFast level may be set to be lower than that of the Ultimate level, the priority level of the Fast level is lower

than that of the UltraFast level, and the priority level of the LowLatency level is lower than that of the Fast level; and a Normal level is weakest in decoding capability but best in compatibility and may serve as a minimum guaranteed compatibility grade, and thus, the priority level of the Normal level is the lowest.

[0097] In addition, priority levels of other compatibility grades may be set according to practical situations and demands, which is not specially limited by the exemplary embodiment.

[0098] S520: Performing grade logic configuration on the at least two compatibility grades according to the priority levels to obtain a compatibility detection rule.

[0099] Grade logic configuration may be performed on the at least two compatibility grades after the compatibility grades and the corresponding priority levels are determined.

[0100] Specifically, the compatibility detection rule obtained by grade logic configuration may firstly select the Ultimate level to determine whether the decoder is applicable or not. When the decoder is not applicable to the Ultimate level, the UltraFast level is secondly selected to determine whether the decoder is applicable or not. When the decoder is not applicable to the Ultimate level and the UltraFast level, the Fast level is thirdly selected to determine whether the decoder is applicable or not. When the decoder is not applicable to the Ultimate level, the UltraFast level and the Fast level, the LowLatency level is fourthly selected to determine whether the decoder is applicable or not. When the decoder is not applicable to the Ultimate level, the UltraFast level, the Fast level and the LowLatency level, the Normal level is finally selected to be applicable to the decoder.

[0101] In the exemplary embodiment, the compatibility detection rule is obtained by performing compatibility logic configuration on the decoding parameters to provide a decoding logic for later adopting the decoders for hardware decoding processing, and compatibility of different decoders is brought into play under the situation that the optimal decoding property of the decoders is brought into full play.

[0102] S230: Obtaining at least two compatibility grade configurations according to parameter grade matched processing of the at least two compatibility grades and a plurality of configuration results, and selecting one compatibility grade configuration from the at least two compatibility grade configurations.

[0103] In the exemplary embodiments of the present disclosure, since the at least two compatibility grades are determined, and to further determine a corresponding relation between the different compatibility grades and the parameter configuration results, parameter grade matched

processing may be performed on the at least two compatibility grades and the plurality of configuration results.

[0104] In some embodiments, FIG. 6 illustrates a step flowchart of a parameter grade matched processing method. As shown in FIG. 6, the method at least may include the following steps: S610: Performing parameter grade matched processing on at least two compatibility grades and a plurality of frame rate configuration results to obtain a first compatibility grade configuration.

[0105] The compatibility grades performing parameter grade matched processing with the plurality of frame rate configuration results may be an Ultimate level and an UltraFast level.

[0106] In some embodiments, a plurality of frame rate configuration results may include: a first frame rate configuration result in which frame rate setting parameters are set according to the maximum value; and a second frame rate configuration result in which frame rate setting parameters are set according to the minimum value. Specifically, in the Ultimate level, to-be-configured frame rate setting parameters include priority, frame-rate and operating-rate. The first frame rate configuration result is matched with the Ultimate level, namely, frame-rate and operating-rate are configured according to the maximum value. Frame-rate is set as 120, and operating-rate is set as 240.

[0107] In the UltraFast level, to-be-configured frame rate setting parameters may also include priority, frame-rate and operating-rate. The second frame rate configuration result is matched with the UltraFast grade, namely, frame-rate and operating-rate are configured according to the minimum value. Frame-rate is set as 60, and operating-rate is set as 120.

[0108] S620: Performing parameter grade matched processing on the at least two compatibility grades and a plurality of hoarded frame configuration results to obtain a second compatibility grade configuration.

[0109] The compatibility grades performing parameter grade matched processing with the hoarded frame configuration results may be a Fast level and a LowLatency level.

[0110] Specifically, in the Fast level, a matched hoarded frame configuration result includes: low-latency corresponding to a setInteger interface is set as 1, low-latency corresponding to a setFeatureEnabled interface is set to be true, and corresponding parameters of chip vendor private interfaces, such as “vendor.qti-...-low-latency.enable”, “vendor.hisi-...-low-latency-req” and “vendor. ...-low-latency.enable” are set as 1.

[0111] In the LowLatency level, a matched hoarded frame configuration result includes: low-latency corresponding to the setInteger interface is set as 1, and low-latency corresponding to the setFeatureEnabled interface is set to be true.

[0112] S630: Performing parameter grade matched processing on the at least two compatibility grades and a default configuration result to obtain a third compatibility grade configuration.

[0113] A third compatibility grade performing parameter grade matched processing with the default configuration result instead of the frame rate configuration results and the hoarded frame configuration results may be a Normal level. Namely, the Normal level does not need to configure any decoding parameter and only needs to guarantee that the decoders can normally output decoding results.

[0114] In the exemplary embodiment, the corresponding compatibility grade configurations may be obtained by performing parameter grade matched processing on the compatibility grades and the parameter configuration results, thereby providing a parameter basis for the decoders for performing hardware decoding processing according to the compatibility grades and guaranteeing the low-latency decoding capability of the decoders.

[0115] Furthermore, grade configuration selecting processing may be performed on the at least two compatibility grade configurations to determine one compatibility grade configuration from the at least two compatibility grade configurations.

[0116] In some embodiments, grade configuration selecting processing is performed on at least two compatibility grade configurations according to the priority levels to determine one compatibility grade configuration.

[0117] A compatibility detection rule is set according to the priority levels corresponding to the compatibility grades, and thus, when the compatibility grade is selected for configuration for the first time, a corresponding configuration of an Ultimate level may be firstly selected as the determined compatibility configuration. When the compatibility grade is not selected for configuration for the first time, a next compatibility grade configuration of the current compatibility grade configuration may be selected according to the compatibility grade. Whether the next compatibility grade configuration needs to be selected or not may be decided according to whether a decoder being abnormal is displayed in the current compatibility grade configuration or not.

[0118] S240: Configuring and start a decoder according to the compatibility grade configuration, and performing decoding abnormality detection processing on the decoder to obtain an abnormality detection result.

[0119] In the exemplary embodiment of the present disclosure, decoding abnormality detection processing may be performed on the decoder under the current compatibility grade configuration to obtain the abnormality detection result. Decoding abnormality detection processing on the decoder may be determined in a process of performing hardware decoding processing on a to-be-decoded video and may also be determined by a front detection means of hardware decoding processing, which is not specially limited by the exemplary embodiment. The abnormality detection result may indicate that a decoder is normal and a decoder is abnormal.

[0120] S250: Performing hardware decoding processing on the to-be-decoded video to obtain a decoding processing result according to the compatibility grade configuration if the abnormality detection result indicates that the decoder is normal, and determining that the current compatibility grade configuration is applicable to the decoder if the decoding processing result indicates that decoding is successful.

[0121] In the exemplary embodiment of the present disclosure, the abnormality detection result may indicate that a decoder is normal and a decoder is abnormal, and thus, further hardware decoding processing may be performed on the to-be-decoded video according to different abnormality detection results.

[0122] In some embodiments, hardware decoding processing is performed on a to-be-decoded video according to a compatibility grade configuration to obtain a decoding processing result if an abnormality detection result indicates that a decoder is normal.

[0123] After decoding abnormality detection processing is performed on the decoder according to the compatibility grade configuration, the abnormality detection result indicates that the decoder is normal if a driver layer dose not send any abnormal information. Namely, the decoder is applicable to the current compatibility grade configuration and may further perform hardware decoding processing on the to-be-decoded video.

[0124] In addition, an abnormality detection result may also indicate that a decoder is abnormal, and under the situation, the compatibility grade configuration needs to be replaced with a compatibility grade configuration capable of performing hardware decoding processing.

[0125] In some embodiments, if an abnormality detection result indicates that a decoder is abnormal, a next compatibility grade configuration of the compatibility grade configuration is

determined from at least two compatibility grade configurations according to the priority levels, and step S240 is repeatedly performed.

[0126] After decoding abnormality detection processing is performed on the decoder according to the compatibility grade configuration, a driver layer sends abnormal information indicating that the abnormality detection result indicates the decoder being abnormal, and thus, the decoder is not applicable to the current compatibility grade configuration.

[0127] Furthermore, the next compatibility grade configuration of the current compatibility grade configuration is determined according to the priority levels to continuously perform decoding abnormality detection processing till it is determined that an abnormality detection result corresponding to a compatibility grade configuration indicates the decoder being normal, and hardware decoding processing is performed on the to-be-decoded video according to the compatibility grade configuration.

[0128] Decoding processing results obtained after hardware decoding processing on the to-be-decoded video may indicate successful decoding and failed decoding, and thus, determination needs to be performed according to different decoding processing results so that the compatibility grade configuration applicable to the decoder can be determined.

[0129] In some embodiments, if a decoding processing result indicates that a first-frame image of a to-be-decoded video is successfully decoded, it is determined that a compatibility grade configuration is applicable to a decoder.

[0130] For a terminal device not hoarding frames, when an abnormality detection result indicates that a decoder is normal and the decoding processing result indicates that the first-frame image of the to-be-decoded video is successfully decoded, it indicates that the compatibility configuration is applicable to the decoder nominally, in addition, decoding may succeed in a practical usage process, and thus, the compatibility grade configuration is applicable to the decoder.

[0131] In addition, a decoding processing result may indicate that the first-frame image of the to-be-decoded video fails in decoding, and thus, decoding overtime detection processing may be performed on the decoder to determine a compatibility grade configuration applicable to the decoder.

[0132] In some embodiments, FIG. 7 illustrates a step flowchart of a method for determining a compatibility grade configuration applicable to a decoder. As shown in FIG. 7, the method at least may include the following steps: S710: Performing decoding overtime detection

processing on the decoder to obtain overtime detection results if a decoding processing result indicates that a first-frame image of a to-be-decoded video fails in decoding.

[0133] For example, a MediaCodec being a decoder adopts a hardware decoding processing manner to perform hardware processing on the to-be-decoded video, the decoding processing result indicates that the first-frame image of the to-be-decoded video fails in decoding, and it indicates that the decoding failure result is possibly caused by a compatibility grade configuration. Thus, a decoding overtime detection processing process may be added to determine an applicable compatibility grade configuration.

[0134] Specifically, during overtime detection processing, a waiting duration threshold may be set, and accordingly, whether a duration of decoding the first-frame image of the to-be-decoded video meets a decoding requirement or not is determined according to the waiting duration threshold. For example, the waiting duration threshold may be 10ms, 100ms or 1s, and may also be set into other values according to practical situations, which is not specially limited by the exemplary embodiment. Thus, the overtime detection results may include waiting overtime and waiting no overtime.

[0135] S720: Determining a compatibility grade configuration applicable to the decoder according to the overtime detection result.

[0136] In some embodiments, if an overtime detection result indicates waiting overtime, hardware decoding processing is performed on other video frame images except a first-frame image in a to-be-decoded video to determine a compatibility grade configuration applicable to a decoder.

[0137] A duration of decoding the first-frame image of the to-be-decoded video being greater than the waiting duration threshold, or being equal to or greater than the waiting duration threshold indicates that the overtime detection result is waiting overtime.

[0138] Under the situation, to avoid jamming in a later decoding process of other video frame images, hardware decoding processing may be continuously performed on other video frame images, thereby determining whether the current compatibility grade configuration is applicable to the decoder or not in a later hardware decoding process of other video frame images. Moreover, decoding duration detection processing may be performed in the later process of performing hardware decoding processing on other video frame images, and the compatibility grade configuration may be replaced when the abnormality detection result indicates that the decoder is abnormal. In other words, replacement of the compatibility grade configuration is not limited to the decoding abnormality detection processing process of the first-frame image.

[0139] In the process of performing hardware decoding processing on other video frame images, if the first-frame image is successfully decoded, it is determined that the current compatibility grade configuration is applicable to the decoder; and in the process of performing hardware decoding processing on other video frame images, if the first-frame image fails in decoding, a next compatibility grade configuration may be continuously selected for performing processes including decoding abnormality detection processing and hardware decoding processing, thereby determining a compatibility grade configuration applicable to the decoder from the remaining compatibility grade configurations.

[0140] In addition, the overtime detection result may indicate waiting no overtime, and thus, secondary decoding abnormality detection processing needs to be performed to determine a compatibility grade configuration applicable to the decoder.

[0141] In some embodiments, if an overtime detection result indicates waiting no overtime, decoding abnormality detection processing is performed on a decoder again according to a compatibility grade configuration to determine the compatibility grade configuration applicable to the decoder.

[0142] A duration of decoding a first-frame image of a to-be-decoded video being less than or equal to a waiting duration threshold, or being less than the waiting duration threshold indicates that the overtime detection result is waiting overtime.

[0143] Under the situation, it is possible that a frame hoarding terminal device caches a first-frame image hardware decoding processing result in an inner cache area of the decoder, and consequently, the first-frame image hardware decoding processing result indicates successful decoding but cannot be sensed. Thus, to eliminate the frame hoarding situation, a secondary decoding abnormality detection processing process may be performed according to the compatibility grade to determine that the first-frame image cannot be successfully decoded when the decoder is normal.

[0144] Specifically, when the abnormality detection result of secondary decoding abnormality detection processing indicates that the decoder is normal, it indicates that the terminal device has a frame hoarding behavior at the time, and thus it may be determined that the current compatibility grade configuration is applicable to the decoder. When the abnormality detection result of secondary decoding abnormality detection processing indicates that the decoder is abnormal, it indicates that unsuccessful decoding of the first-frame image is caused by abnormality of the decoder, and thus a next compatibility grade may be adopted through replacement to perform

later decoding abnormality detection processing and hardware decoding processing to determine a compatibility grade configuration applicable to the decoder.

[0145] In the two exemplary embodiments, the compatibility grade configurations applicable to the decoder may be determined according to the different overtime detection results, which cannot jam the later hardware decoding processing process of other video frame images and can determine the compatibility grade configuration under the situation that the frame hoarding behavior is eliminated, accuracy is better, and practicability is higher.

[0146] The decoder configuration method provided in the embodiment of the present disclosure is described in detail by combining a specific application scenario below.

[0147] FIG. 8 illustrates a step flowchart of a decoder configuration method for a terminal device not hoarding frames. As shown in FIG. 8, S810: Configuring low-latency parameters.

[0148] Low-latency parameter configuration may mean that rule configuration processing is performed on decoding parameters of a decoder according to a parameter configuration rule.

[0149] When the terminal device not hoarding frames uses an Android system, the decoding parameters are mainly configured, by a setInteger interface and a setFeatureEnabled interface, in a MediaFormat according to inputted key values. Common decoding parameters, namely key names may be provided by an Android official document, and in addition, when the decoder is a decoding chip, a chip vendor will expose private key names serving as the decoding parameters.

[0150] Specifically, refer to Table 1:

Table 1

Parameter set	Interface	Key name	Key value
Real-time communication	setInteger	“priority”	0
	setInteger	“frame-rate”	60/120
Low-latency	setInteger	“low-latency”	1
	setFeatureEnabled	“low-latency”	true
Operating frequency	setInteger	“operating-rate”	120/240
Chip vendor private	setInteger	“vendor.qti-...-low-latency.enable”	1
	setInteger	“vendor.hisi-...-low-latency-req”	1

interface	setInteger	“vendor. ...-low-latency.enable”	1
	setInteger	...	...

[0151] Decoding parameters corresponding to “real-time communication” are set as priority and frame-rate, configure the decoding chip in a real-time communication mode, and set a target frame rate in the real-time communication mode. Decoding parameters corresponding to “low-latency” configure low-latency to remove hoarded frames of the decoding chip to reach a purpose of rapid frame exporting. Decoding parameters corresponding to “operating frequency” configure operating-rate to adjust the operating frequency in the decoding chip to adapt to a decoding target frame rate. Decoding parameters corresponding to “chip vendor private interface” include private parameters of vendors with current mainstream decoding chips, and through configuration of the private parameters, the decoding chip can run with higher efficiency, the frame hoarding behavior in the decoding chip is closed, and decoding time delay is shortened.

[0152] There are many kinds of decoding chips of the Android system, the decoding chips designed by part of chip vendors may only support one or several of decoding parameters in Table 1 and possibly have a compatibility problem for the decoding parameters in Table 1, and thus compatibility detection may be further performed on the decoding parameters.

[0153] The decoding parameters in Table 1 may have a certain compatibility problem for the decoding chips produced by the different chip vendors, different Android system versions and video resolutions of different to-be-decoded videos.

[0154] For example, when a decoding chip such as MSM8916 sets “frame-rate” as 60, the decoding chip will fail in initialization, and when a video resolution of a to-be-decoded video is 1080p, a decoding chip such as SDM670 sets operating-rate as 240, which causes failed initialization of the decoding chip. Thus, it is necessary to perform compatibility detection on the decoding parameters.

[0155] Firstly, compatibility grade division processing is performed on the decoding parameters to obtain 5 compatibility grades including Ultimate, UltraFast, Fast, LowLatency and Normal.

[0156] Specifically, the Ultimate level belongs to an ultimate configuration, and all the parameters in Table 1 are configured according to the maximum value in the level. Frame-rate is set as 120, and operating-rate is set as 240.

[0157] The UltraFast level belongs to an ultrafast configuration, and all the parameters in Table 1 are configured according to the minimum value in the level, at the time, frame-rate is set as 60, and operating-rate is set as 120.

[0158] The Fast level belongs to a fast configuration, decoding parameters corresponding to “real-time communication” and “operating frequency” in the level are not configured, and only decoding parameters corresponding to “low-latency” and “chip vendor private interface” are configured.

[0159] The LowLatency level belongs to a low-latency configuration, and only decoding parameters corresponding to “low-latency” are configured in the level.

[0160] The Normal level belongs to a normal configuration, and none of the decoding parameters in Table 1 is configured in the level.

[0161] In the five compatibility grades, the Ultimate level has the best decoding capability and can bring the extreme decoding capability of decoders such as decoding chips into play, a decoding frame rate of part of the decoding chips is improved by near 5 times compared with the Normal level, and thus the priority level is the highest; the UltraFast level, the Fast level and the LowLatency level are set to make choices for the decoding parameters to adapt to the compatibility problem of a terminal with different decoders, thus, the UltraFast level, the Fast level and the LowLatency level may be set to the same in priority level but be lower than the Ultimate level in priority level, the priority level of the UltraFast level may be set to be lower than that of the Ultimate level, the priority level of the Fast level is lower than that of the UltraFast level, and the priority level of the LowLatency level is lower than that of the Fast level; and the Normal level is weakest in decoding capability but best in compatibility and may serve as a minimum guaranteed compatibility grade, and thus, it is guaranteed that the decoder may normally perform decoding and output video images.

[0162] According to the five compatibility grades, grade logic configuration is sequentially performed according to the priority levels from the Ultimate level to the Normal level to obtain a compatibility detection rule.

[0163] Specifically, the compatibility detection rule obtained by grade logic configuration may firstly select the Ultimate level to determine whether the decoder is applicable or not. When the decoder is not applicable to the Ultimate level, the UltraFast level is secondly selected to determine whether the decoder is applicable or not. When the decoder is not applicable to the Ultimate level and the UltraFast level, the Fast level is thirdly selected to determine whether the decoder is

applicable or not. When the decoder is not applicable to the Ultimate level, the UltraFast level and the Fast level, the LowLatency level is fourthly selected to determine whether the decoder is applicable or not. When the decoder is not applicable to the Ultimate level, the UltraFast level, the Fast level and the LowLatency level, the Normal level is finally selected to be applicable to the decoder.

[0164] Furthermore, parameter grade matched processing is performed on the compatibility grade and parameter configuration results in Table 1 to obtain compatibility grade configurations.

[0165] In the Ultimate level, to-be-configured frame rate setting parameters include priority, frame-rate and operating-rate. Configuration on frame-rate and operating-rate is performed according to the maximum value. Frame-rate is set as 120, and operating-rate is set as 240.

[0166] In the UltraFast level, to-be-configured frame rate setting parameters may also include priority, frame-rate and operating-rate. Configuration on frame-rate and operating-rate is performed according to the minimum value. Frame-rate is set as 60, and operating-rate is set as 120.

[0167] In the Fast level, low-latency corresponding to a setInteger interface is set as 1, low-latency corresponding to a setFeatureEnabled interface is set to be true, and corresponding parameters of chip vendor private interfaces, such as “vendor.qti-...-low-latency.enable”, “vendor.hisi-...-low-latency-req” and “vendor. ...-low-latency.enable” are set as 1.

[0168] In the LowLatency level, low-latency corresponding to the setInteger interface is set as 1, and low-latency corresponding to the setFeatureEnabled interface is set to be true.

[0169] The Normal level does not need to configure any decoding parameter and only needs to guarantee that the decoders can normally output decoding results.

[0170] When one selected compatibility grade succeeds in configuration, the compatibility grade configuration may be recorded as the optimal level of the decoding chip in a low-latency scenario, and later same-resolution to-be-decoded videos are subjected to adaption according to the compatibility grade configuration in the level. However, when the compatibility grade fails in configuration, switching to a next compatibility grade of the compatibility grade is performed for configuration.

[0171] A compatibility detection rule is set according to the priority levels corresponding to the compatibility grades, and thus, when the compatibility grade is selected for configuration for the first time, a corresponding configuration of an Ultimate level may be firstly selected as the determined compatibility configuration. When the compatibility grade is not selected for configuration for the first time, a next compatibility grade configuration of the current compatibility

grade configuration may be selected according to the compatibility grade. Whether the next compatibility grade configuration needs to be selected or not may be decided according to whether the decoder being abnormal is displayed in the current compatibility grade configuration or not.

[0172] S820: Starting the decoder.

[0173] To determine whether the current compatibility grade configuration is adaptive to the decoder or not, the decoder may be started for performing later decoding abnormality detection processing.

[0174] S830: Determining whether the decoder is abnormal or not.

[0175] A decoder configuration process and a decoder starting process are asynchronous, thus it is undesirable to simply return to determine whether configuration succeeds or not, namely whether the compatibility grade configuration is applicable to the decoder or not, and in other words, the decoder may be determined by performing decoding abnormality detection processing according to the current compatibility grade configuration.

[0176] S840: Replacing a configuration level if the decoder is abnormal.

[0177] If an abnormality detection result of decoding abnormality detection processing indicates that the decoder is abnormal, a next compatibility grade configuration of the compatibility grade configuration is determined from at least two compatibility grade configurations according to the priority levels, thereby performing hardware decoding processing on a to-be-decoded video to obtain a decoding processing result.

[0178] After decoding abnormality detection processing is performed on the decoder according to the compatibility grade configuration, a driver layer sends abnormal information indicating that the abnormality detection result indicates the decoder being abnormal, and thus, the decoder is not applicable to the current compatibility grade configuration.

[0179] Furthermore, the next compatibility grade configuration of the current compatibility grade configuration is determined according to the priority levels to continuously perform decoding abnormality detection processing till it is determined that an abnormality detection result corresponding to a compatibility grade configuration indicates the decoder being normal, and hardware decoding processing is performed on the to-be-decoded video according to the compatibility grade configuration.

[0180] Decoding processing results obtained after hardware decoding processing on the to-be-decoded video may indicate successful decoding and failed decoding, and thus, determination

needs to be performed according to different decoding processing results so that the compatibility grade configuration applicable to the decoder can be determined.

[0181] S850: Determining whether a first frame of the to-be-decoded video is successfully decoded or not.

[0182] If the abnormality detection result indicates that the decoder is normal, hardware decoding processing is performed on the to-be-decoded video according to the compatibility grade configuration to obtain a decoding processing result.

[0183] After decoding abnormality detection processing is performed on the decoder according to the compatibility grade configuration, the abnormality detection result indicates that the decoder is normal if a driver layer does not send any abnormal information. Namely, the decoder is applicable to the current compatibility grade configuration and may further perform hardware decoding processing on the to-be-decoded video.

[0184] For a machine type not hoarding frames, the decoding processing result can be determined after a first-frame image is outputted. When the first-frame image fails in output, namely, the decoding processing result indicates that the first-frame image of the to-be-decoded video fails in decoding, secondary decoding abnormality detection processing may be performed to determine whether the reason for failed decoding of the first-frame image is the failed compatibility grade configuration or not.

[0185] S860: Decoding following video frames.

[0186] For the machine type not hoarding frames, when the decoding processing result indicates that the first-frame image is outputted, it indicates that the first-frame image succeeds in decoding, and thus the compatibility grade configuration is adaptive to the decoder and may be utilized for performing hardware decoding processing on other video frame images except the first-frame image of the to-be-decoded video.

[0187] FIG. 9 illustrates a step flowchart of a decoder configuration method for a terminal device hoarding frames. As shown in FIG. 9, S910: Configuring low-latency parameters.

[0188] Low-latency parameter configuration may mean that rule configuration processing is performed on decoding parameters of a decoder according to a parameter configuration rule.

[0189] When a terminal device not hoarding frames uses an Android system, the decoding parameters are mainly configured, by a setInteger interface and a setFeatureEnabled interface, in a MediaFormat according to inputted key values. Common decoding parameters, namely key names

may be provided by an Android official document, and in addition, when the decoder is a decoding chip, a chip vendor will expose private key names serving as the decoding parameters.

[0190] Specifically, refer to decoding parameters shown in Table 1. Decoding parameters corresponding to “real-time communication” configure the decoding chip in a real-time communication mode by setting priority and frame-rate, and set a target frame rate in the real-time communication mode. Decoding parameters corresponding to “low-latency” configure low-latency to remove hoarded frames of the decoding chip to reach a purpose of rapid frame exporting. Decoding parameters corresponding to “operating frequency” configure operating-rate to adjust the operating frequency in the decoding chip to adapt to a decoding target frame rate. Decoding parameters corresponding to “chip vendor private interface” include private parameters of vendors with current mainstream decoding chips, and through configuration of the private parameters, the decoding chip can run with higher efficiency, the frame hoarding behavior in the decoding chip is closed, and decoding time delay is shortened.

[0191] There are many kinds of decoding chips of the Android system, the decoding chips designed by part of chip vendors may only support one or several of decoding parameters in Table 1 and possibly have a compatibility problem for the decoding parameters in Table 1, and thus compatibility detection may be further performed on the decoding parameters.

[0192] The decoding parameters in Table 1 may have a certain compatibility problem for the decoding chips produced by the different chip vendors, different Android system versions and video resolutions of different to-be-decoded videos.

[0193] For example, when a decoding chip such as MSM8916 sets “frame-rate” as 60, the decoding chip will fail in initialization, and when a video resolution of a to-be-decoded video is 1080p, a decoding chip such as SDM670 sets operating-rate as 240, which causes failed initialization of the decoding chip. Thus, it is necessary to perform compatibility detection on the decoding parameters.

[0194] Firstly, compatibility grade division processing is performed on the decoding parameters to obtain 5 compatibility grades including Ultimate, UltraFast, Fast, LowLatency and Normal.

[0195] Specifically, the Ultimate level belongs to an ultimate configuration, and all the parameters in Table 1 are configured according to the maximum value in the level. Frame-rate is set as 120, and operating-rate is set as 240.

[0196] The UltraFast level belongs to an ultrafast configuration, and all the parameters in Table 1 are configured according to the minimum value in the level, at the time, frame-rate is set as 60, and operating-rate is set as 120.

[0197] The Fast level belongs to a fast configuration, decoding parameters corresponding to “real-time communication” and “operating frequency” in the level are not configured, and only decoding parameters corresponding to “low-latency” and “chip vendor private interface” are configured.

[0198] The LowLatency level belongs to a low-latency configuration, and only decoding parameters corresponding to “low-latency” are configured in the level.

[0199] The Normal level belongs to a normal configuration, and none of the decoding parameters in Table 1 is configured in the level.

[0200] In the five compatibility grades, the Ultimate level has the best decoding capability and can bring the extreme decoding capability of decoders such as decoding chips into play, a decoding frame rate of part of the decoding chips is improved by near 5 times compared with the Normal level, and thus the priority level is the highest; the UltraFast level, the Fast level and the LowLatency level are set to make choices for the decoding parameters to adapt to the compatibility problem of a terminal with different decoders, thus, the UltraFast level, the Fast level and the LowLatency level may be set to the same in priority level but be lower than the Ultimate level in priority level, the priority level of the UltraFast level may be set to be lower than that of the Ultimate level, the priority level of the Fast level is lower than that of the UltraFast level, and the priority level of the LowLatency level is lower than that of the Fast level; and the Normal level is weakest in decoding capability but best in compatibility and may serve as a minimum guaranteed compatibility grade, and thus, it is guaranteed that the decoder may normally perform decoding and output video images.

[0201] According to the five compatibility grades, grade logic configuration is sequentially performed according to the priority levels from the Ultimate level to the Normal level to obtain a compatibility detection rule.

[0202] Specifically, the compatibility detection rule obtained by grade logic configuration may firstly select the Ultimate level to determine whether the decoder is applicable or not. When the decoder is not applicable to the Ultimate level, the UltraFast level is secondly selected to determine whether the decoder is applicable or not. When the decoder is not applicable to the Ultimate level and the UltraFast level, the Fast level is thirdly selected to determine whether the decoder is

applicable or not. When the decoder is not applicable to the Ultimate level, the UltraFast level and the Fast level, the LowLatency level is fourthly selected to determine whether the decoder is applicable or not. When the decoder is not applicable to the Ultimate level, the UltraFast level, the Fast level and the LowLatency level, the Normal level is finally selected to be applicable to the decoder.

[0203] Furthermore, parameter grade matched processing is performed on the compatibility grade and parameter configuration results in Table 1 to obtain compatibility grade configurations.

[0204] In the Ultimate level, to-be-configured frame rate setting parameters include priority, frame-rate and operating-rate. Configuration on frame-rate and operating-rate is performed according to the maximum value. Frame-rate is set as 120, and operating-rate is set as 240.

[0205] In the UltraFast level, to-be-configured frame rate setting parameters may also include priority, frame-rate and operating-rate. Configuration on frame-rate and operating-rate is performed according to the minimum value. Frame-rate is set as 60, and operating-rate is set as 120.

[0206] In the Fast level, low-latency corresponding to a setInteger interface is set as 1, low-latency corresponding to a setFeatureEnabled interface is set to be true, and corresponding parameters of chip vendor private interfaces, such as “vendor.qti-...-low-latency.enable”, “vendor.hisi-...-low-latency-req” and “vendor. ...-low-latency.enable” are set as 1.

[0207] In the LowLatency level, low-latency corresponding to the setInteger interface is set as 1, and low-latency corresponding to the setFeatureEnabled interface is set to be true.

[0208] The Normal level does not need to configure any decoding parameter and only needs to guarantee that the decoders can normally output decoding results.

[0209] When one selected compatibility grade succeeds in configuration, the compatibility grade configuration may be recorded as the optimal level of the decoding chip in a low-latency scenario, and later same-resolution to-be-decoded videos are subjected to adaption according to the compatibility grade configuration in the level. However, when the compatibility grade fails in configuration, switching to a next compatibility grade of the compatibility grade is performed for configuration.

[0210] A compatibility detection rule is set according to the priority levels corresponding to the compatibility grades, and thus, when the compatibility grade is selected for configuration for the first time, a corresponding configuration of an Ultimate level may be firstly selected as the determined compatibility configuration. When the compatibility grade is not selected for configuration for the first time, a next compatibility grade configuration of the current compatibility

grade configuration may be selected according to the compatibility grade. Whether the next compatibility grade configuration needs to be selected or not may be decided according to whether a decoder being abnormal is displayed in the current compatibility grade configuration or not.

[0211] S920: Starting the decoder.

[0212] To determine whether the current compatibility grade configuration is adaptive to the decoder or not, the decoder may be started for performing later decoding abnormality detection processing.

[0213] S930: Determining whether the decoder is abnormal or not.

[0214] A decoder configuration process and a decoder starting process are asynchronous, thus it is undesirable to simply return to determine whether configuration succeeds or not, namely whether the compatibility grade configuration is applicable to the decoder or not, and in other words, the decoder may be determined by performing decoding abnormality detection processing according to the current compatibility grade configuration.

[0215] S940: Replacing a configuration level.

[0216] If an abnormality detection result of decoding abnormality detection processing indicates that the decoder is abnormal, a next compatibility grade configuration of the compatibility grade configuration is determined from at least two compatibility grade configurations according to the priority levels, thereby performing hardware decoding processing on the to-be-decoded video to obtain a decoding processing result.

[0217] After decoding abnormality detection processing is performed on the decoder according to the compatibility grade configuration, a driver layer sends an abnormality information indicating that the abnormality detection result indicates the decoder being abnormal, and thus, the decoder is not applicable to the current compatibility grade configuration.

[0218] Furthermore, the next compatibility grade configuration of the current compatibility grade configuration is determined according to the priority levels to continuously perform decoding abnormality detection processing till it is determined that an abnormality detection result corresponding to a compatibility grade configuration indicates the decoder being normal, and hardware decoding processing is performed on the to-be-decoded video according to the compatibility grade configuration.

[0219] Decoding processing results obtained after hardware decoding processing on the to-be-decoded video may indicate successful decoding and failed decoding, and thus, determination

needs to be performed according to different decoding processing results so that the compatibility grade configuration applicable to the decoder can be determined.

[0220] S950: Determining whether a first frame of the to-be-decoded video is successfully decoded or not.

[0221] For the terminal device hoarding frames, decoded images can be outputted as long as a specific number of video frames are inputted. For example, for a terminal device hoarding 3 frames, a 1st-frame video frame can be outputted after a 4th-frame video frame is inputted. The problem of frame hoarding can be solved according to the decoder configuration method shown in FIG. 8, namely, the frame hoarding behavior in the decoding chip is closed.

[0222] Thus, determination on whether hardware decoding processing on the first-frame image of the frame-hoarding machine type succeeds or not may be determined according to a decoding processing result about whether the first-frame image is outputted or not.

[0223] S960: Determining whether waiting is overtime or not if decoding fails.

[0224] For decoding chips produced by some small vendors and old-generation decoding chips, the problem of frame hoarding cannot be solved by the decoder configuration method shown in FIG. 8. Thus, a waiting duration threshold may be set to perform decoding overtime detection processing on the decoder to prevent jamming of the following hardware decoding process of other video frame images in the detection process.

[0225] If an overtime detection result indicates waiting no overtime, decoding abnormality detection processing is performed on the decoder again according to a compatibility grade configuration to determine the compatibility grade configuration applicable to the decoder.

[0226] A duration of decoding the first-frame image of the to-be-decoded video being less than or equal to the waiting duration threshold, or being less than the waiting duration threshold indicates that the overtime detection result is waiting overtime.

[0227] Under the situation, it is possible that the frame hoarding terminal device caches a first-frame image hardware decoding processing result in an inner cache area of the decoder, and consequently, the first-frame image hardware decoding processing result indicates successful decoding but cannot be sensed. Thus, to eliminate the frame hoarding situation, a secondary decoding abnormality detection processing process may be performed according to the compatibility grade to determine that the first-frame image cannot be successfully decoded when the decoder is normal.

[0228] Specifically, when the abnormality detection result of secondary decoding abnormality detection processing indicates that the decoder is normal, it indicates that the terminal device has a frame hoarding behavior at the time, and thus it may be determined that the current compatibility grade configuration is applicable to the decoder. When the abnormality detection result of secondary decoding abnormality detection processing indicates that the decoder is abnormal, it indicates that unsuccessful decoding of the first-frame image is caused by abnormality of the decoder, and thus a next compatibility grade may be adopted through replacement to perform later decoding abnormality detection processing and hardware decoding processing to determine a compatibility grade configuration applicable to the decoder.

[0229] S970: Decoding following video frames.

[0230] If the overtime detection result indicates waiting overtime, hardware decoding processing is performed on other video frame images except the first-frame image in the to-be-decoded video to determine a compatibility grade configuration applicable to the decoder.

[0231] A duration of decoding the first-frame image of the to-be-decoded video being greater than the waiting duration threshold, or being equal to or greater than the waiting duration threshold indicates that the overtime detection result is waiting overtime.

[0232] Under the situation, to avoid jamming in a later decoding process of other video frame images, hardware decoding processing may be continuously performed on other video frame images, thereby determining whether the current compatibility grade configuration is applicable to the decoder or not in a later hardware decoding process of other video frame images. Moreover, decoding duration detection processing may be performed in the later process of performing hardware decoding processing on other video frame images, and the compatibility grade configuration may be replaced when the abnormality detection result indicates that the decoder is abnormal. In other words, replacement of the compatibility grade configuration is not limited to the decoding abnormality detection processing process of the first-frame image.

[0233] In the process of performing hardware decoding processing on other video frame images, if the first-frame image is successfully decoded, it is determined that the current compatibility grade configuration is applicable to the decoder; and in the process of performing hardware decoding processing on other video frame images, if the first-frame image fails in decoding, a next compatibility grade configuration may be continuously selected for performing processes including decoding abnormality detection processing and hardware decoding processing, thereby determining a compatibility grade configuration applicable to the decoder from the remaining compatibility grade configurations.

[0234] It can be known from the above application scenario that the decoder configuration method provided by the embodiments of this application, on one hand, performs rule configuration processing on the decoding parameters, solves the problem about decoding frame hoarding of the decoder and the problem about time delay in the decoder hardware decoding processing process, achieves a higher decoding frame rate and an effect of rapid hardware decoding and frame exporting, and satisfies the requirements of the low-latency video service scenario; and on the other hand, compatibility logic configuration and parameter grade matched processing are performed on the decoding parameters to obtain the compatibility grade configurations, thereby solving the problem about compatibility of different decoders, and bringing the optimal decoding property of the decoders into play.

[0235] Although the steps of the method in the present disclosure are described in a specific order in the accompanying drawings, this does not require or imply that the steps have to be performed in the specific order, or all the steps shown have to be performed to achieve an expected result. Additionally or alternatively, some steps may be omitted, a plurality of steps may be combined into one step, and/or one step may be decomposed into a plurality of steps for execution, and the like.

[0236] The following describes apparatus embodiments of the present disclosure, and the apparatus embodiments may be used for performing the decoder configuration method in the foregoing embodiment of the present disclosure. For details not disclosed in the apparatus embodiments of the present disclosure, reference may be made to the decoder configuration method in the foregoing embodiments of the present disclosure.

[0237] FIG. 10A illustrates a schematic structural diagram of a decoder configuration apparatus in some embodiments of the present disclosure. As shown in FIG. 10A, the apparatus 10 includes: a decoder configuration module 11, an abnormality detection module 12 and a decoding processing module 13.

[0238] The decoder configuration module 11 is configured to acquire at least two compatibility grade configurations used for configuring decoding parameters of a decoder, where the different compatibility grade configurations being different in priority level, decoding property and decoder compatibility; and sequentially select a compatibility grade configuration from the compatibility grade configurations according to the priority levels from high to low.

[0239] For the current selected compatibility grade configuration: the decoder configuration module 11 is configured to configure the decoder according to the compatibility grade configuration and start the decoder; the abnormality detection module 12 is configured to perform decoding

abnormality detection processing on the decoder to obtain an abnormality detection result; and the decoding processing module 13 is configured to perform hardware decoding processing on a to-be-decoded video according to the current selected compatibility grade configuration to obtain a decoding processing result if the abnormality detection result indicates that the decoder is normal, and determine that the current compatibility grade configuration is applicable to the decoder if the decoding processing result indicates that decoding is successful.

[0240] In some embodiments, the apparatus 10 further includes:

[0241] a rule configuration module 14 configured to acquire decoding parameters of a decoder and determine a parameter configuration rule corresponding to a decoding demand; and configure the decoding parameters according to the parameter configuration rule to obtain a plurality of configuration results of the decoding parameters, different configuration results corresponding to different decoding properties; and

[0242] a compatibility grade configuration module 15 configured to set at least two compatibility grades according to the decoding parameters, the different compatibility grades being different in decoder compatibility; and obtain at least two compatibility grade configurations according to parameter grade matched processing of the at least two compatibility grades and the plurality of preliminary configuration results.

[0243] In some embodiments, decoding parameters include frame rate setting parameters and decoding hoarded frame parameters, and configuration results include frame rate configuration results, hoarded frame configuration results and a default configuration result in which no configuration is performed on the decoding parameters; and

[0244] the rule configuration module 14 is further configured to:

[0245] configure the frame rate setting parameters according to a parameter configuration rule to obtain the plurality of frame rate configuration results; and

[0246] configure the decoding hoarded frame parameters according to the parameter configuration rule to obtain the plurality of hoarded frame configuration results.

[0247] In some embodiments, the compatibility grade configuration module 15 is further configured to:

[0248] perform parameter grade matched processing on at least two compatibility grades and a plurality of frame rate configuration results to obtain a first compatibility grade configuration; and

[0249] perform parameter grade matched processing on the at least two compatibility grades and the plurality of hoarded frame configuration results to obtain a second compatibility grade configuration; and

[0250] perform parameter grade matched processing on the at least two compatibility grades and a default configuration result to obtain a third compatibility grade configuration.

[0251] In some embodiments, the compatibility grade configuration module 15 is further configured to: perform priority level configuration on the at least two compatibility grades to obtain priority levels of the at least two compatibility grades.

[0252] In some embodiments, the decoding processing module 13 is further configured to: trigger a decoder configuration module 11 to determine a next compatibility grade configuration of the current compatibility grade configuration from at least two compatibility grade configurations according to the priority levels if an abnormality detection result indicates that a decoder is abnormal, configure the decoder again according to the compatibility grade configuration, restart the decoder, and trigger an abnormality detection module 12 to perform decoding abnormality detection processing on the decoder again.

[0253] In some embodiments, the decoding processing module 13 is further configured to: determine that a decoding processing result indicates successful decoding if a first-frame image of a to-be-decoded video is successfully decoded and outputted from a decoder.

[0254] In some embodiments, the decoding processing module 13 is further configured to: trigger a decoder configuration module 11 to determine a next compatibility grade configuration of the current compatibility grade configuration from at least two compatibility grade configurations according to the priority levels if a decoding processing result indicates that decoding fails, configure a decoder again according to the compatibility grade configuration, restart the decoder, and trigger an abnormality detection module 12 to perform decoding abnormality detection processing on the decoder again.

[0255] In some embodiments, the decoding processing module 13 is further configured to: perform decoding overtime detection processing on a decoder to obtain an overtime detection result if a decoding processing result indicates that decoding fails; and perform decoding processing on other video frame images except a first-frame image in a to-be-decoded video if the overtime detection result indicates waiting overtime, and determine whether the current compatibility grade configuration is applicable to the decoder or not in the decoding process of other video frame images.

[0256] In some embodiments, the decoding processing module 13 is further configured to: determine whether a decoding duration of a first-frame image of a to-be-decoded video is greater than a duration threshold or not according to the preset duration threshold; and determine that an overtime detection result is waiting overtime if the decoding duration is greater than or equal to or greater than the duration threshold.

[0257] FIG. 10B schematically illustrates a structure block diagram of a decoder configuration apparatus in some embodiments of the present disclosure. As shown in FIG. 10B, the decoder configuration apparatus 1000 may mainly include: a rule configuration module 1010, a detection rule module 1020, a grade matching module 1030, an abnormality detection module 1040 and a decoding processing module 1050.

[0258] The rule configuration module 1010 is configured to acquire decoding parameters of a decoder and a to-be-decoded video corresponding to the decoder, and determine a parameter configuration rule corresponding to the decoding parameters; the detection rule module 1020 is configured to perform rule configuration processing on the decoding parameters according to the parameter configuration rule to obtain parameter configuration results and perform compatibility logic configuration on the decoding parameters to obtain a compatibility detection rule including at least two compatibility grades; the grade matching module 1030 is configured to perform parameter grade matched processing on the at least two compatibility grades and parameter configuration results to obtain at least two compatibility grade configurations, and perform grade configuration selecting processing on the at least two compatibility grade configurations to determine one compatibility grade configuration; the abnormality detection module 1040 is configured to perform decoding abnormality detection processing on the decoder according to the compatibility grade configuration to obtain an abnormality detection result; and the decoding processing module 1050 is configured to perform hardware decoding processing on the to-be-decoded video according to the abnormality detection result, and determine the compatibility grade configuration applicable to the decoder according to the decoding processing result.

[0259] In some embodiments of the present disclosure, the detection rule module 1020 includes:

[0260] a frame rate setting submodule configured to perform rule configuration processing on frame rate setting parameters according to a parameter configuration rule to obtain frame rate configuration results; and

[0261] a frame hoarding setting submodule configured to perform rule configuration processing on decoding hoarded frame parameters according to the parameter configuration rule to obtain hoarded frame configuration results.

[0262] In some embodiments of the present disclosure, the grade matching module 1030 includes:

[0263] a frame rate matching submodule configured to perform parameter grade matched processing on at least two compatibility grades and frame rate configuration results to obtain a compatibility grade configuration; and

[0264] a frame hoarding matching submodule configured to perform parameter grade matched processing on the at least two compatibility grades and hoarded frame configuration results to obtain a compatibility grade configuration; and

[0265] a mismatching submodule configured not to perform parameter grade matched processing on the at least two compatibility grades and the frame rate configuration results and not to perform parameter grade matched processing on the at least two compatibility grades and the hoarded frame configuration results to obtain a compatibility grade configuration.

[0266] In some embodiments of the present disclosure, the detection rule module 1020 includes:

[0267] a grade division submodule configured to perform compatibility grade division processing on decoding parameters to obtain at least two compatibility grades; and

[0268] a logic configuration submodule configured to perform grade logic configuration on the at least two compatibility grades to obtain a compatibility detection rule.

[0269] In some embodiments of the present disclosure, the logic configuration submodule includes:

[0270] a grade configuration unit configured to perform priority level configuration on at least two compatibility grades to obtain priority levels of the at least two compatibility grades; and

[0271] a grade assigning unit configured to perform grade logic configuration on the at least two compatibility grades according to the priority levels to obtain a compatibility detection rule.

[0272] In some embodiments of the present disclosure, the grade matching module 1030 includes: a configuration selecting submodule configured to perform grade configuration selecting processing on at least two compatibility grade configurations according to the priority levels to determine one compatibility grade configuration.

[0273] In some embodiments of the present disclosure, the decoding processing module 1050 includes: a result abnormality submodule configured to determine a next compatibility grade configuration of a compatibility grade configuration from at least two compatibility grade configurations according to the priority levels if an abnormality detection result indicates a decoder being abnormal, thereby performing hardware decoding processing on a to-be-decoded video to obtain a decoding processing result.

[0274] In some embodiments of the present disclosure, the decoding processing module 1050 includes: a normal decoding submodule configured to perform hardware decoding processing on a to-be-decoded video according to a compatibility grade configuration to obtain a decoding processing result if an abnormality detection result indicates a decoder being normal.

[0275] In some embodiments of the present disclosure, the decoding processing module 1050 includes: a successful decoding submodule configured to determine a compatibility grade configuration being applicable to a decoder if a decoding processing result indicates that a first-frame image of a to-be-decoded video is successfully decoded.

[0276] In some embodiments of the present disclosure, the decoding processing module 1050 includes: a failed decoding submodule configured to perform decoding overtime detection processing on a decoder to obtain an overtime detection result if a decoding processing result indicates that a first-frame image of a to-be-decoded video fails in decoding; and

[0277] a detection result submodule configured to determine a compatibility grade configuration applicable to the decoder according to the overtime detection result.

[0278] In some embodiments of the present disclosure, the detection result submodule includes: a waiting overtime unit configured to perform hardware decoding processing on other video frame images except a first-frame image in a to-be-decoded video to determine a compatibility grade configuration applicable to a decoder if an overtime detection result indicates waiting overtime.

[0279] In some embodiments of the present disclosure, the detection result submodule includes: a normal duration unit configured to perform decoding abnormality detection processing on a decoder again according to a compatibility grade configuration to determine the compatibility grade configuration applicable to the decoder if an overtime detection result indicates waiting no overtime.

[0280] Concrete details of the decoder configuration apparatus provided in various embodiments of the present disclosure are described in detail in corresponding method embodiments, which are not repeated herein.

[0281] FIG. 11 is a schematic structural diagram of a computer system adapted to implement an electronic device according to an embodiment of the present disclosure.

[0282] The computer system 1100 of the electronic device shown in FIG. 11 is merely an example, and does not constitute any limitation on functions and use ranges of the embodiments of this application.

[0283] As shown in FIG. 11, the computer system 1100 comprises a central processing unit (CPU) 1101. The CPU 1101 may perform various appropriate actions and processing according to a program stored in a read-only memory (ROM) 1102 or a program loaded from a storage portion 1108 into a random access memory (RAM) 1103. The RAM 1103 further stores various programs and data required for operating the system. The CPU 1101, the ROM 1102, and the RAM 1103 are connected to each other through a bus 1104. An input/output (I/O) interface 1105 is also connected to the bus 1104.

[0284] Following components are connected to the I/O interface 1105: an input part 1106 including a keyboard and a mouse; an output part 1107 including a Cathode Ray Tube (CRT), a Liquid Crystal Display (LCD) and a loudspeaker; a storage part 1108 including a hard disk; and a communication part 1109 including a Local Area Network (LAN) card, a modem card and other network interface cards. The communication part 1109 performs communication processing by using a network such as the Internet. A driver 1110 is also connected to the I/O interface 1105 as required. A removable medium 1111, such as a magnetic disk, an optical disc, a magneto-optical disk, or a semiconductor memory, is installed on the drive 1110 as required, so that a computer program read from the removable medium is installed into the storage part 1108 as required.

[0285] Particularly, according to an embodiment of the present disclosure, the processes described in each method by referring to the flowcharts may be implemented as computer software programs. For example, an embodiment of the present disclosure includes a computer program product. The computer program product includes a computer program stored in a computer-readable medium. The computer program includes a program code used for performing a method shown in the flowchart. In such an embodiment, the computer program may be downloaded and installed through the communication part 1109 from a network, and/or installed from the removable medium 1111. When the computer program is executed by the CPU 1101, the various functions defined in the system of this application are executed.

[0286] The computer-readable medium shown in the embodiments of the present disclosure may be a computer-readable signal medium or a computer-readable storage medium or any combination thereof. The computer-readable storage medium may be, for example, but is not limited to, an electrical, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, or device, or any combination thereof. A more specific example of the computer-readable storage medium may include but is not limited to: An electrical connection having one or more wires, a portable computer magnetic disk, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM), a flash memory, an optical fiber, a compact disk read-only memory (CD-ROM), an optical storage device, a magnetic storage device, or any appropriate combination thereof. In the present disclosure, the computer-readable storage medium may be any tangible medium containing or storing a program, and the program may be used by or used in combination with an instruction execution system, an apparatus, or a device. In the present disclosure, the computer-readable signal medium may be a data signal included in a baseband or propagated as a part of a carrier, in which computer-readable program code is carried. The propagated data signal may be in a plurality of forms, including, but not limited to, an electromagnetic signal, an optical signal, or any appropriate combination thereof. The computer-readable signal medium may be further any computer-readable medium in addition to a computer-readable storage medium. The computer-readable medium may send, propagate, or transmit a program that is used by or used in combination with an instruction execution system, apparatus, or device. The program code included in the computer-readable medium may be transmitted using any suitable medium, including but not limited to: a wireless medium, a wire medium, or the like, or any suitable combination thereof.

[0287] The flowcharts and block diagrams in the accompanying drawings illustrate possible system architectures, functions, and operations that may be implemented by a system, a method, and a computer program product according to various embodiments of the present disclosure. In this regard, each box in the flowchart or the block diagram may represent a module, a program segment, or a part of code. The module, the program segment, or the part of code includes one or more executable instructions used for implementing specified logic functions. In some implementations used as substitutes, functions annotated in boxes may alternatively occur in a sequence different from that annotated in an accompanying drawing. For example, actually two boxes shown in succession may be performed basically in parallel, and sometimes the two boxes may be performed in a reverse sequence. This is determined by a related function. Each box in the block diagram or the flowchart, and a combination of blocks in the block diagram or the flowchart may be implemented by using a dedicated hardware-based system that performs a specified

function or operation, or may be implemented by using a combination of dedicated hardware and computer instructions.

[0288] Although a plurality of modules or units of a device configured to perform actions are discussed in the foregoing detailed description, such division is not mandatory. Actually, according to the implementations of the present disclosure, the features and functions of two or more modules or units described above may be specified in one module or unit. Conversely, features and functions of one module or unit described above may be further divided into a plurality of modules or units for implementation.

[0289] Through the descriptions of the foregoing implementations, a person skilled in the art easily understands that the exemplary implementations described herein may be implemented through software, or may be implemented through software located in combination with necessary hardware. Therefore, the technical solutions of the embodiments of the present disclosure may be implemented in a form of a software product. The software product may be stored in a non-volatile storage medium (which may be a CD-ROM, a USB flash drive, a removable hard disk, or the like) or on the network, including several instructions for instructing a computing device (which may be a personal computer, a server, a touch terminal, a network device, or the like) to perform the methods according to the embodiments of the present disclosure.

[0290] Other embodiments of the present disclosure will be apparent to a person skilled in the art from consideration of the specification and practice of the present disclosure. This application is intended to cover any variation, use, or adaptive change of the present disclosure. These variations, uses, or adaptive changes follow the general principles of the present disclosure and include common general knowledge or common technical means in the art that are not disclosed in the present disclosure.

[0291] It is to be understood that the present disclosure is not limited to the precise structures described above and shown in the accompanying drawings, and various modifications and changes can be made without departing from the scope of the present disclosure. The scope of the present disclosure is subject only to the appended claims.

CLAIMS

What is claimed is:

1. A decoder configuration method, executed by a computing device, the method comprising:

acquiring at least two compatibility grade configurations used for configuring decoding parameters of a decoder, the different compatibility grade configurations being different in priority level, decoding property and decoder compatibility;

selecting a compatibility grade configuration from the at least two compatibility grade configurations according to the priority levels from high to low;

for the current selected compatibility grade configuration:

configuring the decoder according to the compatibility grade configuration;

starting the decoder, and performing decoding abnormality detection processing on the decoder to obtain an abnormality detection result; and

performing hardware decoding processing on a to-be-decoded video according to the compatibility grade configuration to obtain a decoding processing result in response to the abnormality detection result indicating that the decoder is normal, and determining that the current compatibility grade configuration is applicable to the decoder in response to the decoding processing result indicating that decoding is successful.

2. The decoder configuration method according to claim 1, further comprising:

acquiring the decoding parameters of the decoder and determining a parameter configuration rule corresponding to a decoding demand;

configuring the decoding parameters according to the parameter configuration rule to obtain a plurality of configuration results of the decoding parameters, different configuration results corresponding to different decoding properties;

setting at least two compatibility grades according to the decoding parameters, the different compatibility grades being different in decoder compatibility; and

performing parameter grade matched processing on the at least two compatibility grades and the plurality of configuration results to obtain the at least two compatibility grade configurations.

3. The decoder configuration method according to claim 2, wherein the decoding

parameters include frame rate setting parameters and decoding hoarded frame parameters, and the parameter configuration results include frame rate configuration results, hoarded frame configuration results and a default configuration result in which no configuration is performed on the decoding parameters; and

performing rule configuration processing on the decoding parameters according to the parameter configuration rule to obtain the plurality of configuration results of the decoding parameters comprises:

configuring the frame rate setting parameters according to the parameter configuration rule to obtain a plurality of frame rate configuration results; and

configuring the decoding hoarded frame parameters according to the parameter configuration rule to obtain a plurality of hoarded frame configuration results.

4. The decoder configuration method according to claim 3, wherein performing the parameter grade matched processing on the at least two compatibility grades and the plurality of configuration results to obtain the at least two compatibility grade configurations comprises:

performing parameter grade matched processing on the at least two compatibility grades and the plurality of frame rate configuration results to obtain a first compatibility grade configuration;

performing parameter grade matched processing on the at least two compatibility grades and the plurality of hoarded frame configuration results to obtain a second compatibility grade configuration; and

performing parameter grade matched processing on the at least two compatibility grades and the default configuration result to obtain a third compatibility grade configuration.

5. The decoder configuration method according to claim 2, further comprising:

performing priority level configuration on the at least two compatibility grades to obtain priority levels of the at least two compatibility grades.

6. The decoder configuration method according to claim 1, further comprising:

determining a next compatibility grade configuration of the current compatibility grade configuration from the at least two compatibility grade configurations according to the priority levels in response to the abnormality detection result indicating that the decoder is abnormal, and returning to the operation of configuring the decoder according to the compatibility grade configuration.

7. The decoder configuration method according to any one of claims 1-6, further comprising:

determining that the abnormality detection result indicates the decoder being abnormal in response to abnormal information transmitted by the decoder being received; and

determining that the abnormality detection result indicates the decoder being normal in response to the abnormal information transmitted by the decoder being not received.

8. The decoder configuration method according to any one of claims 1-6, further comprising:

determining that the decoding processing result indicates successful decoding in response to a first-frame image of a to-be-decoded video being successfully decoded and outputted from the decoder.

9. The decoder configuration method according to claim 8, further comprising:

determining a next compatibility grade configuration of the current compatibility grade configuration from the at least two compatibility grade configurations according to the priority levels in response to the decoding processing result indicating that decoding fails, and returning to the operation of configuring the decoder according to the compatibility grade configuration.

10. The decoder configuration method according to claim 8, further comprising:

performing decoding overtime detection processing on the decoder to obtain an overtime detection result in response to the decoding processing result indicating that decoding fails; and

performing decoding processing on other video frame images except the first-frame image in the to-be-decoded video in response to the overtime detection result indicating waiting overtime, and determining whether the current compatibility grade configuration is applicable to the decoder or not in the decoding process of other video frame images.

11. The decoder configuration method according to claim 10, wherein performing the decoding overtime detection processing on the decoder to obtain the overtime detection result comprises:

determining whether a decoding duration of the first-frame image of the to-be-decoded video is greater than a duration threshold or not according to a preset duration threshold; and determining that the overtime detection result is waiting overtime in response to the decoding duration being greater than or equal to or greater than the duration threshold.

12. The decoder configuration method according to claim 10, further comprising:

returning to the operation of performing decoding abnormality detection processing on the decoder in response to the overtime detection result indicating waiting no overtime.

13. A decoder configuration apparatus, comprising:

a decoder configuration module configured to acquire at least two compatibility grade configurations used for configuring decoding parameters of a decoder, the different compatibility grade configurations being different in priority level, decoding property and decoder compatibility; select a compatibility grade configuration from the at least two compatibility grade configurations according to the priority levels from high to low; wherein

for the current selected compatibility grade configuration:

the decoder configuration module is configured to configure the decoder according to the compatibility grade configuration and start the decoder;

an abnormality detection module is configured to perform decoding abnormality detection processing on the decoder to obtain an abnormality detection result; and

a decoding processing module is configured to perform hardware decoding processing on a to-be-decoded video according to the current selected compatibility grade configuration to obtain a decoding processing result in response to the abnormality detection result indicating the decoder being normal, and determine that the current compatibility grade configuration is applicable to the decoder in response to the decoding processing result indicating that decoding is successful.

14. The apparatus according to claim 13, further comprising:

a rule configuration module configured to acquire the decoding parameters of the decoder and determine a parameter configuration rule corresponding to a decoding demand, the parameter configuration rule being a decoding parameter configuration rule determined according to the decoding demand of the to-be-decoded video; and perform rule configuration processing on the decoding parameters according to the parameter configuration rule to obtain a plurality of configuration results of the decoding parameters, different configuration results corresponding to different decoding properties; and

a compatibility grade configuration module configured to set at least two compatibility grades according to the decoding parameters, the different compatibility grades being different in decoder compatibility; and perform parameter grade matched processing on the at least two compatibility grades and the plurality of configuration results to obtain the at least two compatibility grade configurations.

15. The apparatus according to claim 14, wherein the decoding parameters include frame rate setting parameters and decoding hoarded frame parameters, and the configuration results include frame rate configuration results, hoarded frame configuration results and a default configuration result in which no configuration is performed on the decoding parameters; and

the rule configuration module is further configured to:

configure the frame rate setting parameters according to the parameter configuration rule to obtain a plurality of frame rate configuration results; and

configure the decoding hoarded frame parameters according to the parameter configuration rule to obtain a plurality of hoarded frame configuration results.

16. The apparatus according to claim 15, wherein the compatibility grade configuration module is further configured to:

perform parameter grade matched processing on at least two compatibility grades and the plurality of frame rate configuration results to obtain a first compatibility grade configuration;

performing parameter grade matched processing on the at least two compatibility grades and the plurality of hoarded frame configuration results to obtain a second compatibility grade configuration; and

perform parameter grade matched processing on the at least two compatibility grades and the default configuration result to obtain a third compatibility grade configuration.

17. The apparatus according to claim 14, wherein the compatibility grade configuration module is further configured to: perform priority level configuration on the at least two compatibility grades to obtain priority levels of the at least two compatibility grades.

18. The apparatus according to claim 13, wherein the decoding processing module is further configured to: trigger a grade configuration module to determine a next compatibility grade configuration of the current compatibility grade configuration from the at least two compatibility grade configurations according to the priority levels in response to the abnormality detection result indicating that the decoder is abnormal, configure the decoder again according to the compatibility grade configuration, restart the decoder, and trigger the abnormality detection module to perform decoding abnormality detection processing on the decoder again.

19. A computer-readable medium, storing a computer program, the computer program, when executed by a processor, implementing the decoder configuration method according to any one of claims 1 to 12.

20. An electronic device, comprising:

- a processor; and
- a memory, configured to store executable instructions of the processor;

the processor being configured to execute the executable instructions to perform the decoder configuration method according to any one of claims 1 to 12.