(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
19 January 2012 (19.01.2012)

PCT

(10) International Publication Number
WO 2012/006784 A1

(51) **International Patent Classification:**
H04N 7/26 (2006.01)

(21) **International Application Number:**
PCT/CN2010/075184

(22) **International Filing Date:**
15 July 2010 (15.07.2010)

(25) **Filing Language:** English

(26) **Publication Language:** English

(71) **Applicant** *(for all designated States except US):* **MEDIATEK SINGAPORE PTE. LTD.** [SG/SG]; 20, Ayer Rajah Crescent #04-01, Singapore 139964 (SG).

(72) **Inventor; and**

(75) **Inventor/Applicant** *(for US only):* **CUI, Jun** [CN/CN]; A7F, Building No.W2, Middle District, High-tech Industrial Park, Nanshan District, Shenzhen, Guangdong 518057 (CN).

(74) **Agent: BEIJING SANYOU INTELLECTUAL PROPERTY AGENCY LTD.;** 16th FL, Block A, Corporate Square, No. 35 Jinrong Street, Beijing 100033 (CN).

(81) **Designated States** *(unless otherwise indicated, for every kind of national protection available):* AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available):* ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: METHOD FOR SEARCHING FOR FLASH VIDEO TAG IN BITSTREAM AND SEARCHING APPARATUS THEREOF

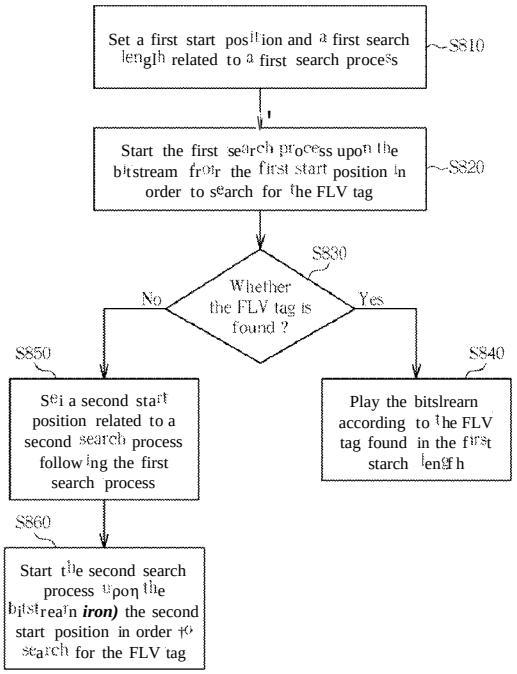

FIG. 8

(57) Abstract: A method for searching for a flash video (FLV) tag in a bitstream includes the following steps: setting a first start position and a first search length related to a first search process, wherein the first search length indicates a bitstream length of the first search process performed upon the bitstream; starting the first search process upon the bitstream from the first start position to search for the FLV tag; when the FLV tag is not found in the first search length, setting a second start position related to a second search process immediately following the first search process, wherein the first start position and the second start position are separated by a time period equaling a sum of the first search length and a first skip length corresponding to the first search process; and starting the second search process upon the bitstream from the second start position to search for the FLV tag.

WO 2012/006784 A1

Published:

— *with international search report (Art. 21(3))*

METHOD FOR SEARCHING FOR FLASH VIDEO TAG IN BITSTREAM AND SEARCHING APPARATUS THEREOF

TECHNICAL FIELD

The present disclosure relates to searching for a video tag in a bitstream, and more particularly, to a searching apparatus and a related method for searching for a flash video (FLV) tag in a bitstream.

BACKGROUND

An FLV format is usually adopted in a media player, such as the Adobe Flash Player. In addition, the FLV format has been widely applied to many video websites, for example, You Tube, tudou, 56.com, ku6.com, Sina Video, and so on. Therefore, the support of the FLV format has become an indispensable part of multimedia functions in consumer electronic products, such as digital TVs, mobiles, or PDAs.

During a playback process of the FLV format, the video data and the audio data need to be extracted by using a "tag" as the basic unit to be decoded by a video decoder and an audio decoder, respectively. For this reason, the correct start position of each FLV tag must be found before the extraction of the video data and the audio data. However, since there is no start code available for locating the start position (or the relative position) of each FLV tag, the playback process of the data complying with the FLV format will be interrupted when an error occurs in the FLV tag.

Hence, how to improve the performance of searching the FLV tag has become an important issue to be solved by designers in this field.

SUMMARY

It is one of the objectives of the present disclosure to provide a searching apparatus and a related method for searching a bitstream for an FLV tag to solve the above-mentioned problems.

According to an embodiment of the present disclosure, a method for searching for a flash video (FLV) tag in a bitstream is provided. The method includes the following steps: setting a first start position and a first search length related to a first search process, wherein the first search length indicates a bitstream length of the first search process performed upon the bitstream; starting the first search process upon the bitstream from the first start position in order to search for the FLV tag; when the FLV tag is not found in the first search length, setting a second start position related to a second search process immediately following the first search process, wherein the first start position and the second start position are separated by a time period equaling a sum of the first search length and a first skip length corresponding to the first search process; and starting the second search process upon the bitstream from the second start position in order to search for the FLV tag.

According to another embodiment of the present disclosure, a searching apparatus for searching for a flash video (FLV) tag in a bitstream is provided. The searching apparatus includes a setting module and a searching module. The setting module is used for setting a first start position and a first search length related to a first search process, wherein the first search length indicates a bitstream length of the first search process performed upon the bitstream. The searching module is used for starting the first search process upon the bitstream from the first start position in order to search for the FLV tag. When the FLV tag is not found in the first search length, the setting module is further used for setting a second start position related to a second search process immediately following the first search process, where the first start position and the second start position are separated by a time period equaling a sum of the first search length and a first skip length corresponding to the first search process; and the searching module is further used for starting the second search process upon the bitstream from the second start position in order to search for the FLV tag.

These and other objectives of the present disclosure will no doubt become obvious to those of ordinary skill in the art after reading the following detailed description of the preferred embodiment that is illustrated in the various figures and drawings.

BRIEF DESCRIPTION OF DRAWINGS

FIG. 1 is a diagram illustrating a data structure of an FLV tag.

FIG.2 is a diagram illustrating a data structure of an audio data in the FLV tag shown in FIG. 1.

FIG. 3 is a diagram illustrating a data structure of a video data in the FLV tag shown in FIG. 1.

FIG.4 is a block diagram illustrating a searching apparatus for searching a bitstream for an FLV tag according to a first embodiment of the present disclosure.

FIG. 5 is a diagram illustrating how to search a bitstream for an FLV tag.

FIG. 6 is a block diagram illustrating a searching apparatus for searching a bitstream for an FLV tag according to a second embodiment of the present disclosure.

FIG. 7 is a diagram showing an exemplary implementation of the searching module shown in FIG.4 or FIG.6.

FIG. 8 is a flowchart illustrating a method for searching a bitstream for an FLV tag according to an exemplary embodiment of the present disclosure.

FIG. 9 is a flowchart illustrating a method for searching a bitstream for an FLV tag according to another exemplary embodiment of the present disclosure.

FIG. 10 is a flowchart illustrating a method for searching a bitstream for an FLV tag according to another exemplary embodiment of the present disclosure.

FIG.1 1A is a flowchart illustrating the step of sequentially searching the bitstream byte by byte for the FLV tag by reference to the specific characteristic of the FLV tag according to an exemplary embodiment of the present disclosure.

FIG. 1IB is a flowchart illustrating the detailed steps of the step S1210 shown in FIG. 11A.

FIG. l lC is a flowchart illustrating the detailed steps of the step S1220 shown in FIG. 11A.

FIG. 1ID is a flowchart illustrating the detailed steps of the step S1230 shown in FIG. 11A.

DETAILED DESCRIPTION

Certain terms are used throughout the description and following claims to refer to particular components. As one skilled in the art will appreciate, manufacturers may refer to a component by different names. This document does not intend to distinguish between components that differ in name but not function. In the following description and in the claims, the terms "include" and "comprise" are used in an open-ended fashion, and thus should be interpreted to mean "include, but not limited to ...". Also, the term "couple" is intended to mean either an indirect or direct electrical connection. Accordingly, if one device is coupled to another device, that connection may be through a direct electrical connection, or through an indirect electrical connection via other devices and connections.

First, in order to make the specification of the present disclosure easy to understand, a brief description of a data structure of an FLV tag is given as below.

Please refer to FIG.l, which is a diagram illustrating a data structure of an FLV tag. As FIG.l depicts, an FLV tag includes a field named as "Tag Type", a field named as "Data Size", a field named as "Time Stamp", a field named as "Time Stamp Extended", a field named as "Stream ID", as well as a field named as "Data". Herein the field named as "Tag Type" indicates the tag type. Be noted that, when a byte value corresponding to the field "Tag Type" is equal to a first value (e.g., 8), it is indicative of an audio type; when the byte value corresponding to the field "Tag Type" is equal to a second value (e.g., 9), it is indicative of a video type. In addition, the field "Data Size" indicates a data length of the field "Data" in the FLV tag. The fields "Time Stamp" and "Time Stamp Extended" are related to the playback time. The byte value corresponding to the field "Stream ID" is usually 0. The field "Data" indicates the data type in the FLV tag, such as audio data, video data, or scrip data object.

FIG.2 is a diagram illustrating a data structure of an audio data in the FLV tag shown in FIG.l, and FIG. 3 is a diagram illustrating a data structure of a video data in the FLV tag shown in FIG.l. As FIG.2 depicts, the audio data includes a sub-field

named "Sound Format", a sub-field named "Sound rate", a sub-field named "Sound size", a sub-field named "Sound type", and a sub-field named " Sound Data". As FIG. 3 depicts, the video data includes a sub-field named "Frame Type", a sub-field named "Codec ID", and a sub-field named " Video Data". As the type and the descriptions corresponding to each sub-field have already been detailed in FIG. 2 and FIG. 3, and further description is omitted here for brevity.

Please refer to FIG.4 together with FIG. 5. FIG.4 is a block diagram illustrating a searching apparatus for searching for an FLV tag in a bitstream according to a first embodiment of the present disclosure, and FIG. 5 is a diagram illustrating how to search for an FLV tag in a bitstream BS. As shown in FIG.4, the searching apparatus 400 includes a setting module 410 and a searching module 420. The setting module 410 is used for setting a first start position PI and a first search length LI related to a first search process SRI, wherein the first search length LI indicates a bitstream length of the first search process SRI performed upon the bitstream BS. The searching module 420 is used for starting the first search process SRI upon the bitstream BS from the first start position PI determined by the setting module 410 in order to search for the FLV tag. When the FLV tag is not found in the first search length LI, the setting module 410 is further used for setting a second start position P2 related to a second search process SR2 following the first search process SRI, where the first start position PI and the second start position P2 are separated by a time period Tl equaling a sum of the first search length LI and a first skip length LI' corresponding to the first search process SRI; and the searching module 420 is further used for starting the second search process SR2 upon the bitstream BS from the second start position P2 in order to search for the FLV tag.

Furthermore, when the FLV tag is not found in the first search length LI, the setting module 410 is further used for setting a second search length L2 related to the second search process SR2, wherein the second search length L2 indicates a bitstream length of the second search process SR2 performed upon the bitstream BS. Thus, when the FLV tag is not found in the second search length L2, the setting module 410 is further used for setting a third start position P3 related to a third search process SR3 following the second search process SR2, wherein the third start position P3 and the second start position P2 are separated by a time period T2

equaling a sum of the second search length L2 and a second skip length L2' corresponding to the second search process SR2. The following search processes may be easily deduced by analogy. For example, when the FLV tag is not found in the n^{th} search length Ln, the setting module 410 is further used for setting a $(n+l)^{th}$ search length L(n+1) related to the $(n+l)^{th}$ search process SR(n+l), wherein the $(n+l)^{th}$ search length L(n+1) indicates a bitstream length of the $(n+l)^{th}$ search process SR(n+l) performed upon the bitstream BS.

Please note that, in this embodiment as shown in FIG. 5, the second skip length L2' is set to be larger than the first skip length LI', the third skip length L3' is set to be larger than the second skip length L2', and so on. But this in no way should be considered to be limitations of the scope of the present disclosure. In other embodiments, for example, the second skip length L2' may be set to be substantially equal to the first skip length LI', and so on. Certainly, people skilled in the art will readily appreciate that various modifications of these skip length Ll'~Ln' are feasible without departing from the scope of the present disclosure.

Additionally, in this embodiment, the first search length LI (or the second search length L2) may be set to be equal to a sum of a video tag size related to a video format and an audio tag size related to an audio format. In other embodiments, the search length Ll~Ln may be set to be greater than or less than the sum of a video tag size related to a video format and an audio tag size related to an audio format, which also belongs to the scope of the present disclosure. Moreover, in this embodiment, the first start position PI may be set as an error position where a bitstream error occurs in the bitstream BS, but this should not be considered to be limitations of the present disclosure.

FIG. 6 is a block diagram illustrating a searching apparatus 600 for searching for an FLV tag in a bitstream according to a second embodiment of the present disclosure. The architecture of the searching apparatus 600 is similar to that of the searching apparatus 400 shown in FIG.4, and the major difference between them is that the searching apparatus 600 further includes a data buffer 630 positioned between a setting module 610 and a searching module 620 for buffering a portion of the bitstream BS as indicated by the first search length LI. Therefore, the searching

module 620 starts the first search process SRI upon the buffered portion of the bitstream BS' from the first start position PI in order to search for the FLV tag.

Please note that the searching module 420/620 may sequentially searches for the FLV tag in the bitstream (or the buffered portion of the bitstream BS) byte by byte according to a specific characteristic of the FLV tag. Further description is detailed as follows.

FIG. 7 is a diagram showing an exemplary implementation of the searching module 420/620 shown in FIG. 4 or FIG. 6. In this embodiment, the searching module 420/620 includes, but is not limited to, a first detecting unit 710, a second detecting unit 720, a third detecting unit 730, and a fourth detecting unit 740. Please also refer to FIG.l and FIG.2. The first detecting unit 710 is used for detecting whether a byte value corresponding to a tag type in the bitstream is equal to a first value (e.g., 8) indicative of an audio type or a second value (e.g., 9) indicative of a video type. The second detecting unit 720 is used for detecting whether a byte value corresponding to a time stamp is greater than a threshold TH and determining whether a time stamp extended includes at least one byte value equal to a first predetermined value (e.g., 0) according to a detection result when the byte value corresponding to the tag type is equal to the first value (e.g., 8) or the second value (e.g., 9). Be noted that: the second detecting unit 720 may be implemented by a first determining sub-unit 722 and a second determining sub-unit 724. The first determining sub-unit 722 is used for determining whether a byte value corresponding to the time stamp extended is equal to the first predetermined value (e.g., 0) when the byte value corresponding to the time stamp is not greater than the threshold TH. The second determining sub-unit 724 is used for determining whether one byte value corresponding to a stream identification (ID) is equal to a second predetermined value (e.g., 0) when the byte value corresponding to the time stamp extended is equal to the first predetermined value (e.g., 0), and for determining whether one byte value corresponding to the stream ID is equal to the second predetermined value (e.g., 0) when the byte value corresponding to the time stamp is greater than the threshold TH.

In addition, the third detecting unit 730 is used for determining whether the

byte value corresponding to the tag type in the bitstream is equal to the first value when the byte value corresponding to the stream ID is equal to the second predetermined value (e.g., 0). The fourth detecting unit 740 is used for determining whether an audio information is received when the byte value corresponding to the tag type is equal to the first value (e.g., 8), for determining whether a byte value corresponding to the audio information is equal to the received audio information when the audio information is received, and for indicating that the FLV tag related to an audio format is found when the byte value corresponding to the audio information is equal to the audio information.

Please note that the third detecting unit 730 shown in FIG. 7 may be configured for determining whether the byte value corresponding to the tag type in the bitstream is equal to the second value (e.g., 9) when the byte value corresponding to the stream ID is equal to the second predetermined value (e.g., 0); and the fourth detecting unit 740 shown in FIG. 7 may be configured for determining whether a codec ID is received when the byte value corresponding to the tag type is equal to the second value (e.g., 9), for determining whether a byte value corresponding to the codec ID is equal to the received codec ID, and for indicating that the FLV tag related to a video format is found when the byte value corresponding to the codec ID is equal to the received codec ID.

Please note that, in some embodiments, the threshold TH can be set as Oxffffff, which indicates the maximum byte value corresponding to the time stamp (3 bytes, or 24 bits). However, this is not meant to be a limitation of the present disclosure. In other embodiments, the threshold TH can be set to be smaller than Oxffffff, and the first determining sub-unit 722 can be used for determining whether a byte value corresponding to a combination of the remaining non-used part of the time stamp as well as the time stamp extended is equal to the first predetermined value (e.g., 0) when the byte value corresponding to the time stamp is not greater than the threshold TH.

Please refer to FIG. 8. FIG. 8 is a flowchart illustrating a method for searching for an FLV tag in a bitstream according to an exemplary embodiment of the present disclosure. Please note that the following steps are not limited to be performed

according to the exact sequence shown in FIG. 8 if a roughly identical result can be obtained. The method includes, but is not limited to, the following steps:

Step S810: Set a first start position and a first search length related to a first search process, wherein the first search length indicates a bitstream length of the first search process performed upon the bitstream.

Step S820: Start the first search process upon the bitstream from the first start position in order to search for the FLV tag.

Step S830: Whether the FLV tag is found. When the FLV tag is found, go to step S840; otherwise, go to step S850.

Step S840: Play the bitstream according to the FLV tag found in the first search length.

Step S850: Set a second start position related to a second search process following the first search process, wherein the first start position and the second start position are separated by a time period equaling a sum of the first search length and a first skip length corresponding to the first search process.

Step S860: Start the second search process upon the bitstream from the second start position in order to search for the FLV tag.

As a person skilled in the art can readily understand the details of the steps in FIG. 8 after reading above paragraphs directed to the searching apparatus 400, further description is omitted here for brevity. Herein the steps S810 and S850 may be executed by the setting module 410, and the steps S820 and S860 may be executed by the searching module 420.

FIG. 9 is a flowchart illustrating a method for searching for an FLV tag in a bitstream according to another exemplary embodiment of the present disclosure. Please note that the following steps are not limited to be performed according to the exact sequence shown in FIG. 9 if a roughly identical result can be obtained. The method includes, but is not limited to, the following steps:

Step S810: Set a first start position and a first search length related to a first search process, wherein the first search length indicates a bitstream length of the first search process performed upon the bitstream.

Step S820: Start the first search process upon the bitstream from the first start position in order to search for the FLV tag.

Step S830: Whether the FLV tag is found. When the FLV tag is found, go to step S840; otherwise, go to step S850.

Step S840: Play the bitstream according to the found FLV tag.

Step S850: Set a second start position related to a second search process following the first search process, wherein the first start position and the second start position are separated by a time period equaling a sum of the first search length and a first skip length corresponding to the first search process.

Step S860: Start the second search process upon the bitstream from the second start position in order to search for the FLV tag.

Step S910: Whether the FLV tag is found. When the FLV tag is found, go to step S920; otherwise, go to step S840.

Step S920: Set a second search length related to the second search process, wherein the second search length indicates a bitstream length of the second search process performed upon the bitstream.

Step S930: Set a third start position related to a third search process following the second search process, wherein the third start position and the second start position are separated by a time period equaling a sum of the second search length and a second skip length corresponding to the second search process.

The steps shown in FIG. 9 are similar to the steps shown in FIG. 8, and the difference between them is that the flowchart shown in FIG. 9 further includes two steps S920 and S930 employed for setting the second search length related to the second search process when the FLV tag is not found in the first search length and for setting a third start position related to a third search process following the second search process when the FLV tag is not found in the second search length, respectively.

FIG. 10 is a flowchart illustrating a method for searching for an FLV tag in a bitstream according to yet another exemplary embodiment of the present disclosure. Please note that the following steps are not limited to be performed according to the exact sequence shown in FIG. 10 if a roughly identical result can be obtained. The method includes, but is not limited to, the following steps:

Step S1010: Buffer a portion of the bitstream as indicated by the first search length.

Step S810: Set a first start position and a first search length related to a first search process, wherein the first search length indicates a bitstream length of the first search process performed upon the bitstream.

Step SI020: Start the first search process upon the buffered portion of the bitstream from the first start position in order to search for the FLV tag.

Step S830: Whether the FLV tag is found. When the FLV tag is found, go to step S840; otherwise, go to step S850.

Step S840: Play the bitstream according to the found FLV tag.

Step S850: Set a second start position related to a second search process following the first search process, wherein the first start position and the second start position are separated by a time period equaling a sum of the first search length and a first skip length corresponding to the first search process.

Step S860: Start the second search process upon the bitstream from the second start position in order to search for the FLV tag.

The steps shown in FIG. 10 are similar to the steps shown in FIG.8, and the difference between them is that the flowchart shown in FIG. 10 further includes a step for buffering a portion of the bitstream as indicated by the first search length (i.e., the step S1010), and the step S820 originally shown in FIG.8 is replaced by the step SI020. Please note that operations of the steps S1010 and SI020 are also detailed in the abovementioned embodiment shown in FIG. 6.

Please note that, the steps of the abovementioned flowchart are presented merely for describing the present invention, and in no way should be considered to be limitations of the scope of the present invention. Those skilled in the art should observe that the methods shown in FIG.8, FIG.9, and FIG. 10 can include other intermediate steps or several steps can be merged into a single step without departing from the spirit of the present invention. For example, the step S820 shown in FIG.8 and FIG.9 or the step SI020 shown in FIG. 10 may be performed by sequentially searching for the FLV tag in the bitstream byte by byte according to a specific characteristic of the FLV tag, and may further include several detailed steps.

Please refer to FIG. 11A, which is a flowchart illustrating the detailed steps of sequentially searching for the FLV tag in the bitstream byte by byte by reference to the specific characteristic of the FLV tag. The method includes, but is not limited to, the following steps:

Step SI 110: Detect whether a byte value corresponding to a tag type in the bitstream is equal to a first value indicative of an audio type or a second value indicative of a video type.

Step SI 120: Detect a byte value corresponding to a time stamp, and detect a byte value corresponding to a time stamp extended.

Step SI 130: Determine whether the FLV tag related to an audio format is found when the byte value corresponding to the tag type is equal to the first value.

Step SI 140: Determine whether the FLV tag related to a video format is found when the byte value corresponding to the tag type is equal to the second value.

FIG. 1IB is a flowchart illustrating the detailed steps of the step SI 120 shown in FIG. 11A, which includes the following detailed steps:

Step SI 121 : When the byte value corresponding to the time stamp is not greater than the threshold, determine whether a byte value corresponding to the time stamp extended is equal to the first predetermined value.

Step SI 122: When the byte value corresponding to the time stamp extended is equal to the first predetermined value, determine whether one byte value corresponding to a stream identification (ID) is equal to a second predetermined value.

Step SI 123: When the byte value corresponding to the time stamp is greater than the threshold, determine whether one byte value corresponding to the stream ID is equal to the second predetermined value.

FIG. l lC is a flowchart illustrating the detailed steps of the step SI 130 shown in FIG. 11A, which includes the following detailed steps:

Step S1131: When the byte value corresponding to the stream ID is equal to the second predetermined value, determine whether the byte value corresponding to the

tag type in the bitstream is equal to the first value.

Step SI 132: When the byte value corresponding to the tag type is equal to the first value, determine whether an audio information is received.

Step SI 133: When the audio information is received, determine whether a byte value corresponding to the audio information is equal to the received audio information.

Step SI 134: When the byte value corresponding to the audio information is equal to the audio information, indicate that the FLV tag related to an audio format is found.

FIG. 1ID is a flowchart illustrating the detailed steps of the step SI 140 shown in FIG. 11A, which includes the following steps:

Step SI 141: When the byte value corresponding to the stream ID is equal to the second predetermined value, determine whether the byte value corresponding to the tag type in the bitstream is equal to the second value.

Step SI 142: When the byte value corresponding to the tag type is equal to the second value, determine whether a codec ID is received.

Step SI 143: When the codec ID is received, determine whether a byte value corresponding to the codec ID is equal to the received codec ID.

Step SI 144: When the byte value corresponding to the codec ID is equal to the received codec ID, indicate that the FLV tag related to a video format is found.

Please note that, the steps of the abovementioned flowchart are presented merely for describing the present invention, and in no way should be considered to be limitations of the scope of the present invention. Those skilled in the art should observe that the methods shown in FIG.8, FIG.9, FIG. 10, and FIG.l lA-FIG.l lD may include other intermediate steps or several steps can be merged into a single step without departing from the spirit of the present invention.

The above-mentioned embodiments are presented merely for describing features of the present disclosure, and in no way should be considered to be limitations of the scope of the present disclosure. In summary, the present disclosure provides a searching apparatus and a related method for searching for an FLV tag in

a bitstream. By inserting the skip length disclosed in the present disclosure into each search process, the FLV tag can be found more easily and more quickly for the extraction of the video data and the audio data. Therefore, the playback process of the data complying with the FLV format can be played unceasingly when an error occurs in the FLV tag.

Those skilled in the art will readily observe that numerous modifications and alterations of the device and method may be made while retaining the teachings of the invention.

CLAIMS

1. A method for searching for a flash video (FLV) tag in a bitstream, comprising:

setting a first start position and a first search length related to a first search process, wherein the first search length indicates a bitstream length of the first search process performed upon the bitstream;

starting the first search process upon the bitstream from the first start position in order to search for the FLV tag;

when the FLV tag is not found in the first search length, setting a second start position related to a second search process following the first search process, wherein the first start position and the second start position are separated by a time period equaling a sum of the first search length and a first skip length corresponding to the first search process; and

starting the second search process upon the bitstream from the second start position in order to search for the FLV tag.

2. The method of claim 1, further comprising:

when the FLV tag is not found in the first search length, setting a second search length related to the second search process, wherein the second search length indicates a bitstream length of the second search process performed upon the bitstream; and

when the FLV tag is not found in the second search length, setting a third start position related to a third search process following the second search process, wherein the third start position and the second start position are separated by a time period equaling a sum of the second search length and a second skip length corresponding to the second search process.

3. The method of claim 2, wherein the second skip length is substantially equal to the first skip length.

4. The method of claim 2, wherein the second skip length is larger than the first skip length.

5. The method of claim 1, wherein the first search length is equal to a sum of a video tag size related to a video format and an audio tag size related to an audio format.

6. The method of claim 1, wherein the first start position is set as an error position where a bitstream error occurs in the bitstream.

7. The method of claim 1, further comprising:

buffering a portion of the bitstream as indicated by the first search length;

wherein the step of starting the first search process upon the bitstream from the first start position comprises:

starting the first search process upon the buffered portion of the bitstream from the first start position in order to search for the FLV tag.

8. The method of claim 1, wherein the step of starting the first search process upon the bitstream from the first start position in order to search for the FLV tag comprises:

sequentially searching for the FLV tag in the bitstream byte by byte according to a specific characteristic of the FLV tag.

9. The method of claim 8, wherein the step of sequentially searching for the FLV tag in the bitstream byte by byte according to the specific characteristic of the FLV tag comprises:

detecting whether a byte value corresponding to a tag type in the bitstream is equal to a first value indicative of an audio type or a second value indicative of a video type.

10. The method of claim 9, wherein the step of sequentially searching for the FLV tag in the bitstream byte by byte according to the specific characteristic of the FLV tag further comprises:

when the byte value corresponding to the tag type is equal to the first value or the second value, detecting whether a byte value corresponding to a time stamp is greater than a threshold and determining whether a time stamp extended includes at

least one byte value equal to a first predetermined value according to a detection result.

11. The method of claim 10, wherein the step of determining whether the time stamp extended include at least one byte value equal to the first predetermined value according to the detection result comprises:

when the byte value corresponding to the time stamp is not greater than the threshold, determining whether a byte value corresponding to the time stamp extended is equal to the first predetermined value;

when the byte value corresponding to the time stamp extended is equal to the first predetermined value, determining whether one byte value corresponding to a stream identification (ID) is equal to a second predetermined value; and

when the byte value corresponding to the time stamp is greater than the threshold, determining whether one byte value corresponding to the stream ID is equal to the second predetermined value.

12. The method of claim 11, wherein the step of sequentially searching for the FLV tag in the bitstream byte by byte according to the specific characteristic of the FLV tag further comprises:

when the byte value corresponding to the stream ID is equal to the second predetermined value, determining whether the byte value corresponding to the tag type in the bitstream is equal to the first value;

when the byte value corresponding to the tag type is equal to the first value, determining whether an audio information is received;

when the audio information is received, determining whether a byte value corresponding to the audio information is equal to the received audio information; and

when the byte value corresponding to the audio information is equal to the audio information, indicating that the FLV tag related to an audio format is found.

13. The method of claim 11, wherein the step of sequentially searching t for the FLV tag he bitstream byte by byte by according to the specific characteristic of the FLV tag further comprises:

when the byte value corresponding to the stream ID is equal to the second

predetermined value, determining whether the byte value corresponding to the tag type in the bitstream is equal to the second value;

when the byte value corresponding to the tag type is equal to the second value, determining whether a codec ID is received;

when the codec ID is received, determining whether a byte value corresponding to the codec ID is equal to the received codec ID; and

when the byte value corresponding to the codec ID is equal to the received codec ID, indicating that the FLV tag related to a video format is found.

14. A searching apparatus for searching for a flash video (FLV) tag in a bitstream, comprising:

a setting module, for setting a first start position and a first search length related to a first search process, wherein the first search length indicates a bitstream length of the first search process performed upon the bitstream; and

a searching module, for starting the first search process upon the bitstream from the first start position in order to search for the FLV tag;

wherein when the FLV tag is not found in the first search length, the setting module is further used for setting a second start position related to a second search process immediately following the first search process, where the first start position and the second start position are separated by a time period equaling a sum of the first search length and a first skip length corresponding to the first search process; and the searching module is further used for starting the second search process upon the bitstream from the second start position in order to search for the FLV tag.

15. The searching apparatus of claim 14, wherein the setting module is further used for:

setting a second search length related to the second search process when the FLV tag is not found in the first search length, wherein the second search length indicates a bitstream length of the second search process performed upon the bitstream; and

setting a third start position related to a third search process following the second search process when the FLV tag is not found in the second search length, wherein the third start position and the second start position are separated by a time period equaling a sum of the second search length and a second skip length

corresponding to the second search process.

16. The searching apparatus of claim 15, wherein the second skip length is substantially equal to the first skip length.

17. The searching apparatus of claim 15, wherein the second skip length is larger than the first skip length.

18. The searching apparatus of claim 14, wherein the first search length is equal to a sum of a video tag size related to a video format and an audio tag size related to an audio format.

19. The searching apparatus of claim 14, wherein the setting module sets the first start position as an error position where a bitstream error occurs in the bitstream.

20. The searching apparatus of claim 14, further comprising:

a data buffer, for buffering a portion of the bitstream as indicated by the first search length;

wherein the searching module starts the first search process upon the buffered portion of the bitstream from the first start position in order to search for the FLV tag.

21. The searching apparatus of claim 14, wherein the searching module sequentially searches for the FLV tag in the bitstream byte by byte f according to a specific characteristic of the FLV tag.

22. The searching apparatus of claim 21, wherein the searching module comprises:

a first detecting unit, for detecting whether a byte value corresponding to a tag type in the bitstream is equal to a first value indicative of an audio type or a second value indicative of a video type.

23. The searching apparatus of claim 22, wherein the searching module further comprises:

a second detecting unit, for detecting whether a byte value corresponding to a time stamp is greater than a threshold and determining whether a time stamp extended includes at least one byte value equal to a first predetermined value according to a detection result when the byte value corresponding to the tag type is equal to the first value or the second value.

24. The searching apparatus of claim 23, wherein the second detecting unit further comprises:

a first determining sub-unit, for determining whether a byte value corresponding to the time stamp extended is equal to the first predetermined value when the byte value corresponding to the time stamp is not greater than the threshold; and

a second determining sub-unit, for determining whether one byte value corresponding to a stream identification (ID) is equal to a second predetermined value when the byte value corresponding to the time stamp extended is equal to the first predetermined value, and for determining whether one byte value corresponding to the stream ID is equal to the second predetermined value when the byte value corresponding to the time stamp is greater than the threshold.

25. The searching apparatus of claim 24, wherein the searching module further comprises:

a third detecting unit, for determining whether the byte value corresponding to the tag type in the bitstream is equal to the first value when the byte value corresponding to the stream ID is equal to the second predetermined value; and

a fourth detecting unit, for determining whether an audio information is received when the byte value corresponding to the tag type is equal to the first value, for determining whether a byte value corresponding to the audio information is equal to the received audio information when the audio information is received, and for indicating that the FLV tag related to an audio format is found when the byte value corresponding to the audio information is equal to the audio information.

26. The searching apparatus of claim 24, wherein the searching module further comprises:

a third detecting unit, for determining whether the byte value corresponding to

the tag type in the bitstream is equal to the second value when the byte value corresponding to the stream ID is equal to the second predetermined value;

a fourth detecting unit, for determining whether a codec ID is received when the byte value corresponding to the tag type is equal to the second value, for determining whether a byte value corresponding to the codec ID is equal to the received codec ID, and for indicating that the FLV tag related to a video format is found when the byte value corresponding to the codec ID is equal to the received codec ID.

Field	Format	Description
Tag Type	UI18(1 byte)	8：audio type，9：video type
Data Size	UI24(3 byte)	The byte length of data
Time Stamp	UI24(3 byte)	The playback time
Time Stamp Extended	UI18(1 byte)	The playback time
Stream ID	UI24(3 byte)	Usually 0
Data	Audio data, video data or script data object	The data type in the FLV tag, such as audio data, video data, or script data object

FIG. 1

Field	Format	Description
Sound Format	UB[4](4 bit)	0：Linear PCM，1：ADPCM， 2：MP3，10：AAC
Sound Rate	UB[2](2 bit)	0：5.5KHz，1：11KHz， 2：22KHz，3：44KHz
Sound Size	UB[1](1 bit)	0：8 bit，1：16 bit
Sound Type	UB[1](1 bit)	0：mono，1：stereo
Sound Data	UI8[N]	Sound Data

FIG. 2

Field	Format	Description
Frame Type	UB[4](4 bit)	1：Key Frame，2：Inter Frame
Code cid	UB[4](4 bit)	1：JPEG，2：Sorenson H263， 3：Screen Video，4：VP6，7：AVC
Video Data	UI8[N]	Video Data

FIG. 3

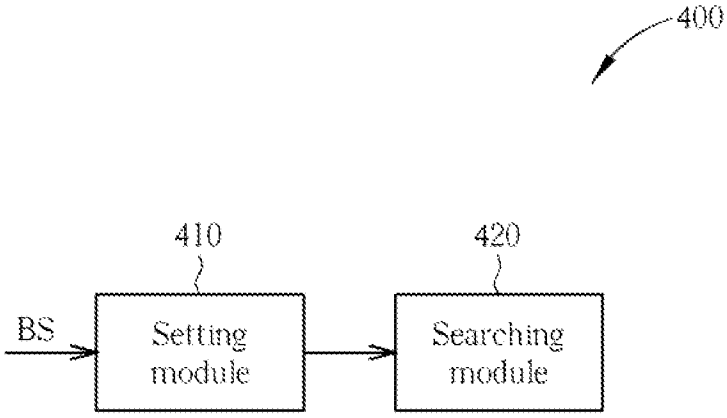

FIG. 4

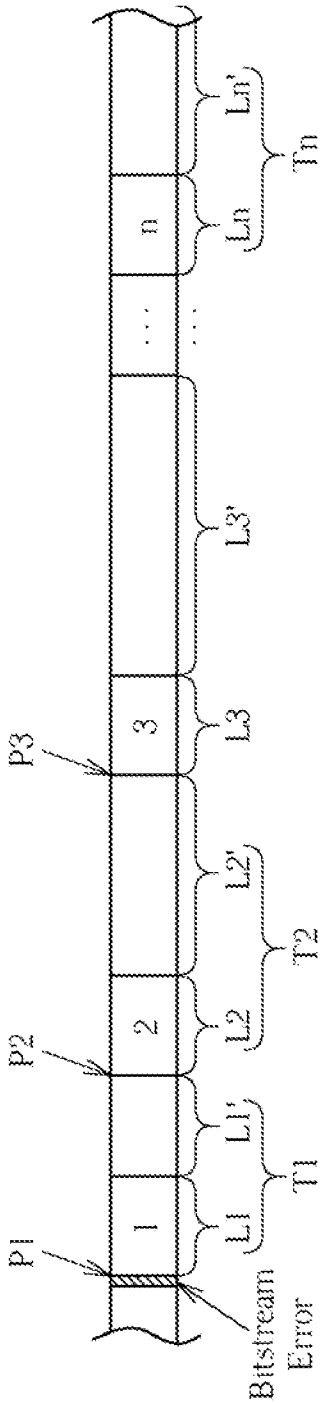

FIG. 5

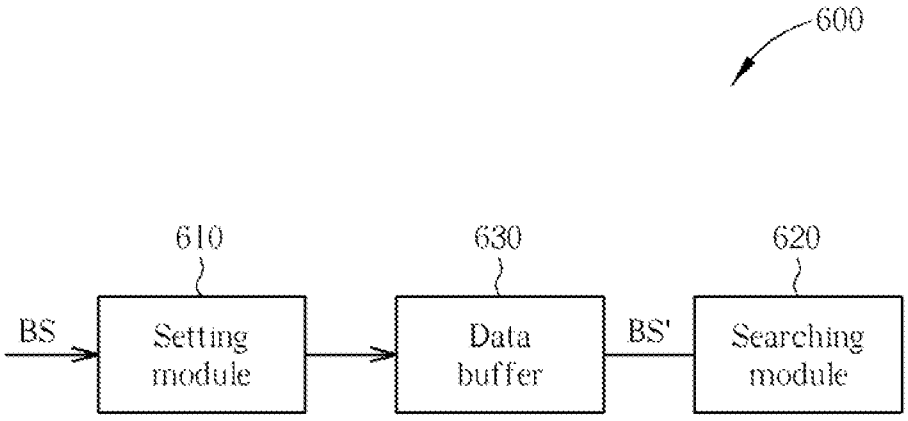
600
610
630
620
BS
Setting module
Data buffer
BS'
Searching module

FIG. 6

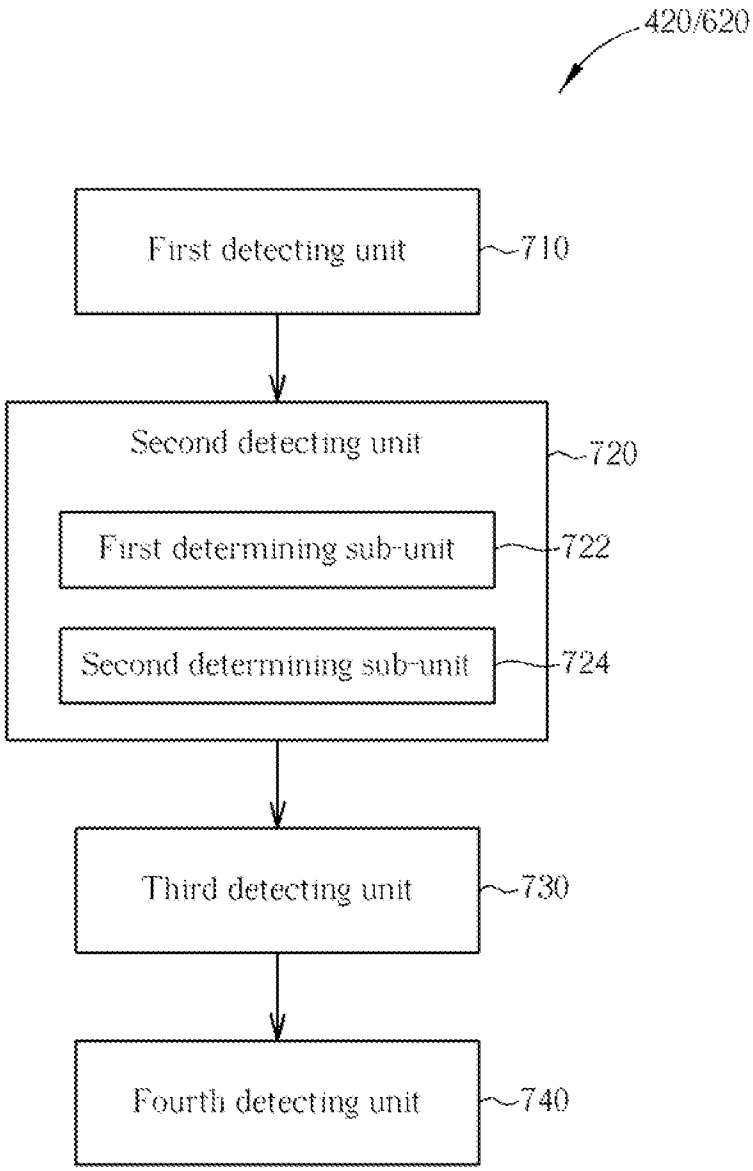
420/620
First detecting unit
710
Second detecting unit
720
First determining sub-unit
722
Second determining sub-unit
724
Third detecting unit
730
Fourth detecting unit
740

FIG. 7

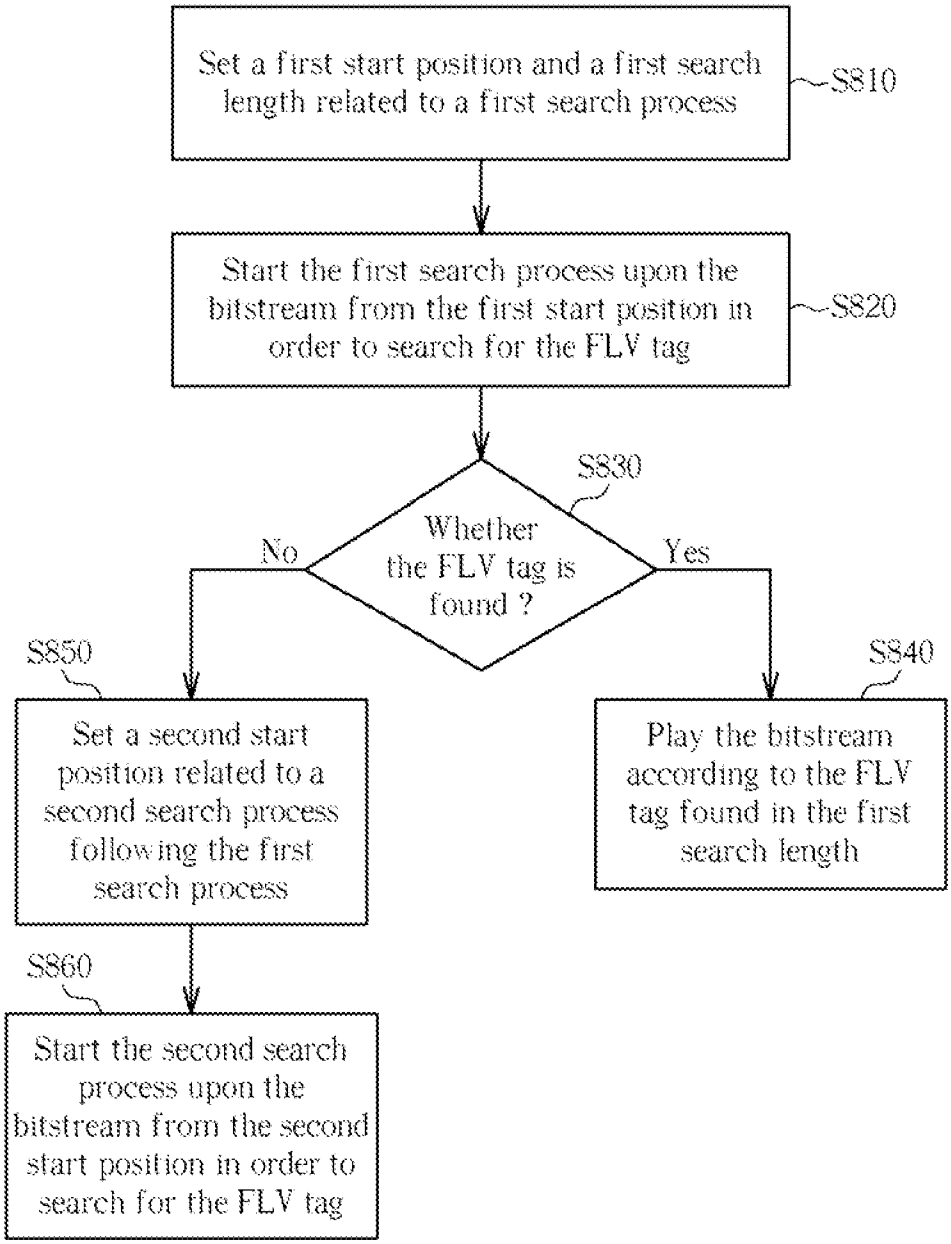
Set a first start position and a first search length related to a first search process
S810
Start the first search process upon the bitstream from the first start position in order to search for the FLV tag
S820
S830
Whether the FLV tag is found ?
No
Yes
S850
Set a second start position related to a second search process following the first search process
S840
Play the bitstream according to the FLV tag found in the first search length
S860
Start the second search process upon the bitstream from the second start position in order to search for the FLV tag

FIG. 8

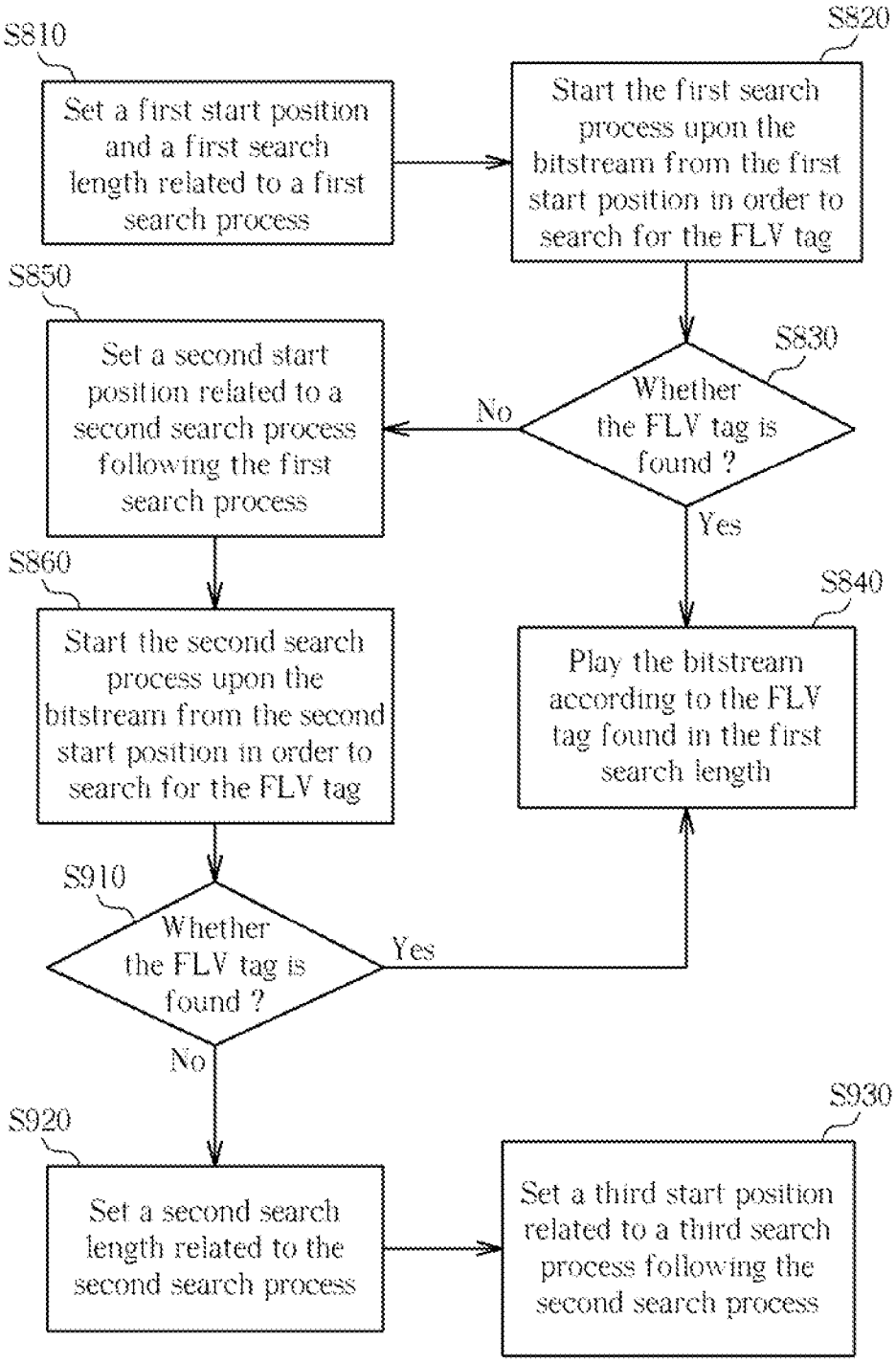
S810
Set a first start position and a first search length related to a first search process
S820
Start the first search process upon the bitstream from the first start position in order to search for the FLV tag
S830
Whether the FLV tag is found ?
No
Yes
S850
Set a second start position related to a second search process following the first search process
S840
Play the bitstream according to the FLV tag found in the first search length
S860
Start the second search process upon the bitstream from the second start position in order to search for the FLV tag
S910
Whether the FLV tag is found ?
Yes
No
S920
Set a second search length related to the second search process
S930
Set a third start position related to a third search process following the second search process

FIG. 9

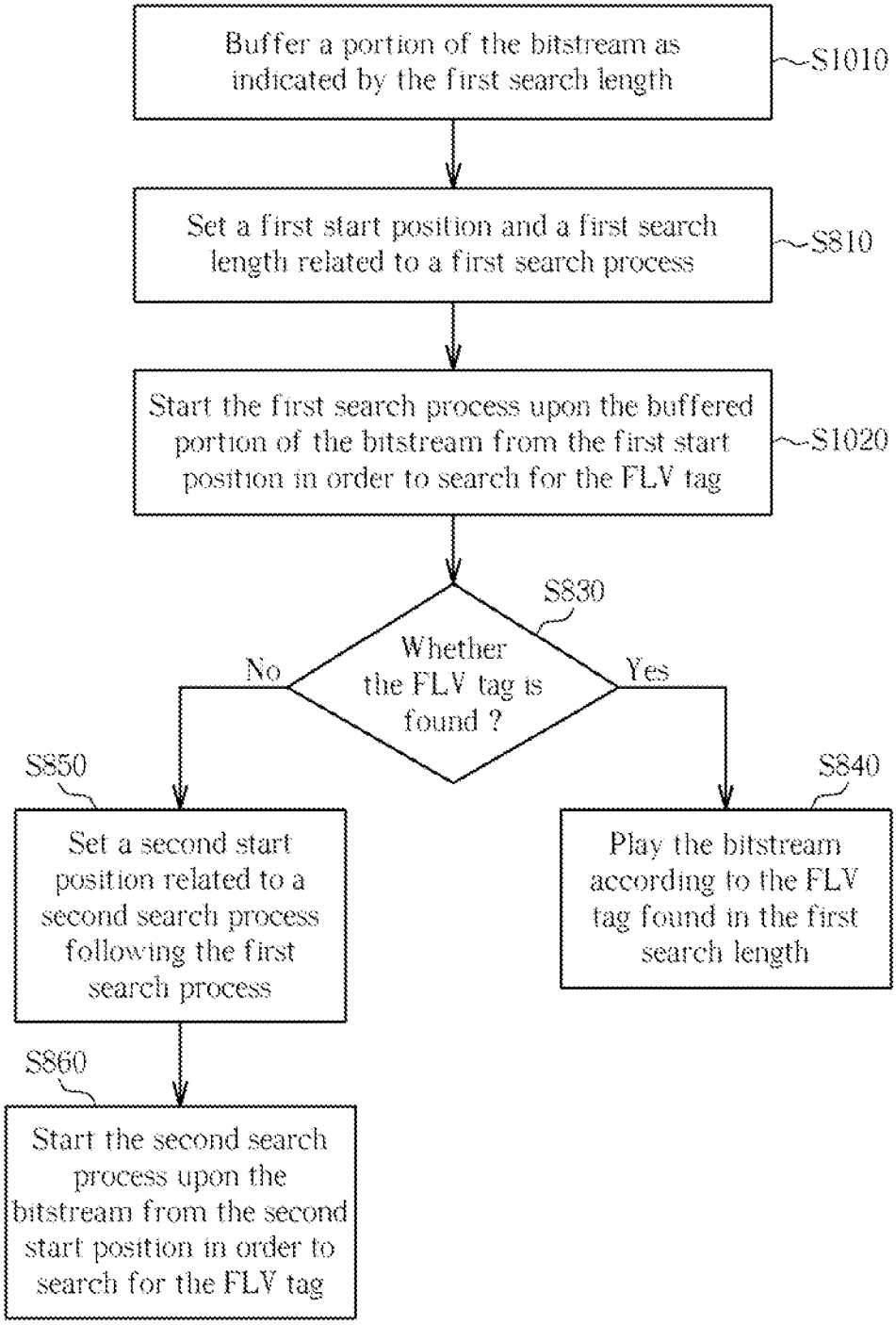
Buffer a portion of the bitstream as indicated by the first search length
S1010
Set a first start position and a first search length related to a first search process
S810
Start the first search process upon the buffered portion of the bitstream from the first start position in order to search for the FLV tag
S1020
S830
Whether the FLV tag is found ?
No
Yes
S850
Set a second start position related to a second search process following the first search process
S840
Play the bitstream according to the FLV tag found in the first search length
S860
Start the second search process upon the bitstream from the second start position in order to search for the FLV tag

FIG. 10

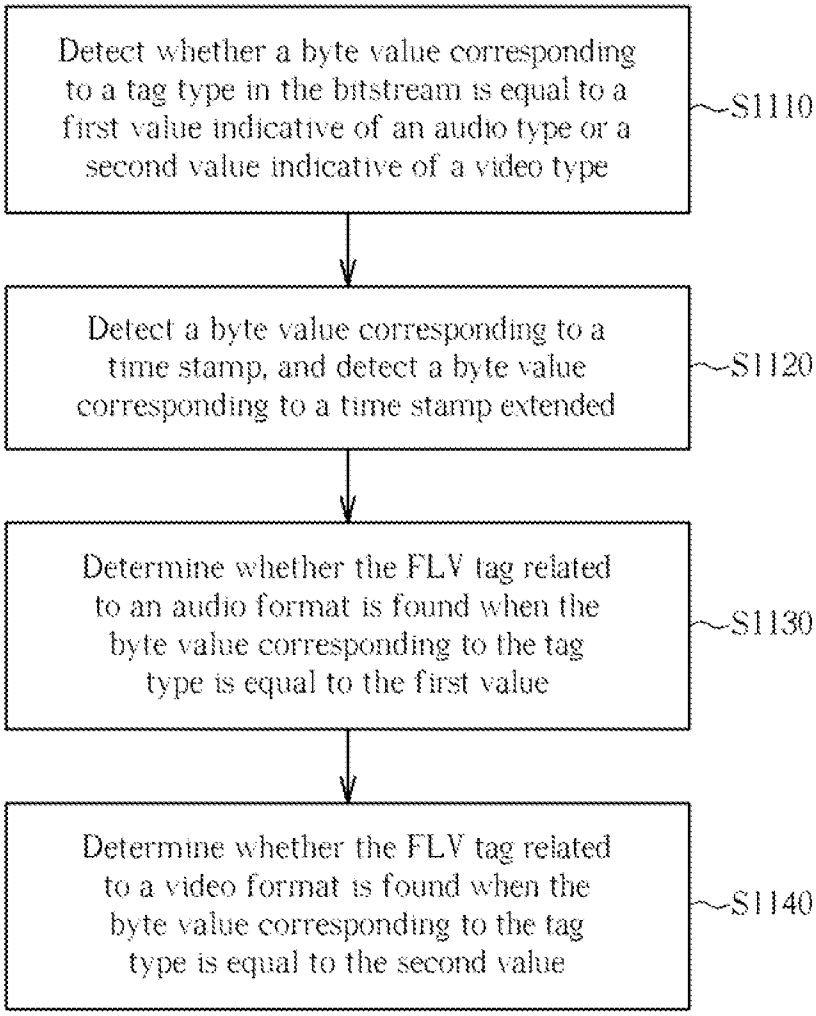
Detect whether a byte value corresponding to a tag type in the bitstream is equal to a first value indicative of an audio type or a second value indicative of a video type
S1110
Detect a byte value corresponding to a time stamp, and detect a byte value corresponding to a time stamp extended
S1120
Determine whether the FLV tag related to an audio format is found when the byte value corresponding to the tag type is equal to the first value
S1130
Determine whether the FLV tag related to a video format is found when the byte value corresponding to the tag type is equal to the second value
S1140

FIG. 11A

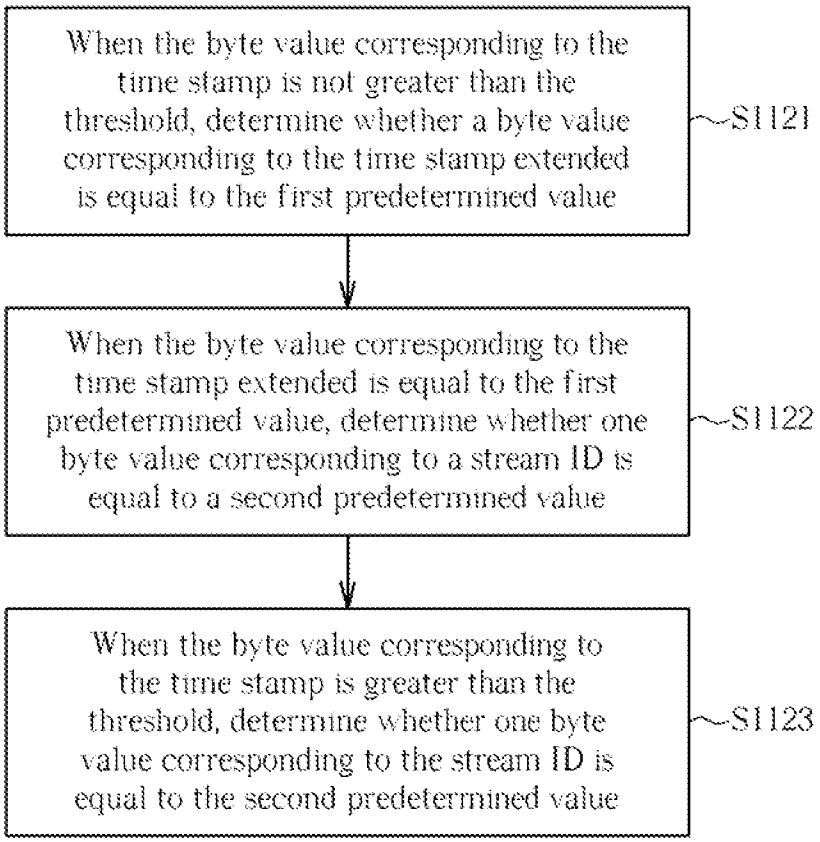
When the byte value corresponding to the time stamp is not greater than the threshold, determine whether a byte value corresponding to the time stamp extended is equal to the first predetermined value
S1121
When the byte value corresponding to the time stamp extended is equal to the first predetermined value, determine whether one byte value corresponding to a stream ID is equal to a second predetermined value
S1122
When the byte value corresponding to the time stamp is greater than the threshold, determine whether one byte value corresponding to the stream ID is equal to the second predetermined value
S1123

FIG. 11B

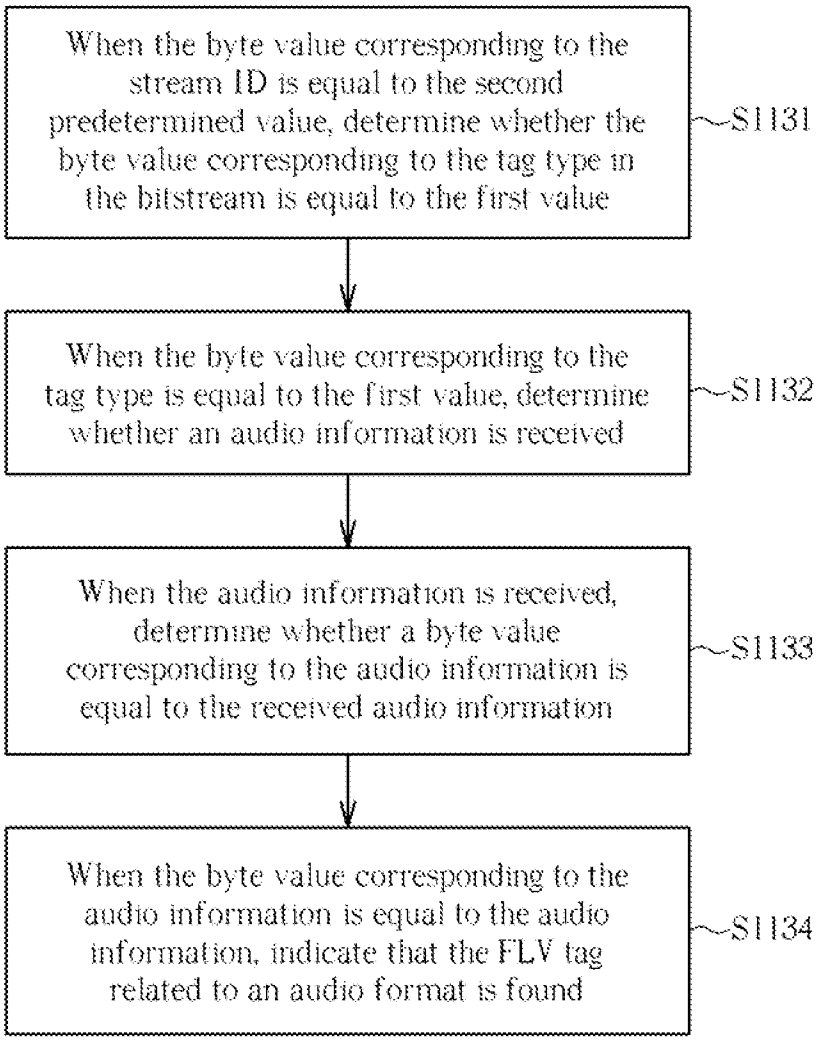
When the byte value corresponding to the stream ID is equal to the second predetermined value, determine whether the byte value corresponding to the tag type in the bitstream is equal to the first value
S1131
When the byte value corresponding to the tag type is equal to the first value, determine whether an audio information is received
S1132
When the audio information is received, determine whether a byte value corresponding to the audio information is equal to the received audio information
S1133
When the byte value corresponding to the audio information is equal to the audio information, indicate that the FLV tag related to an audio format is found
S1134

FIG. 11C

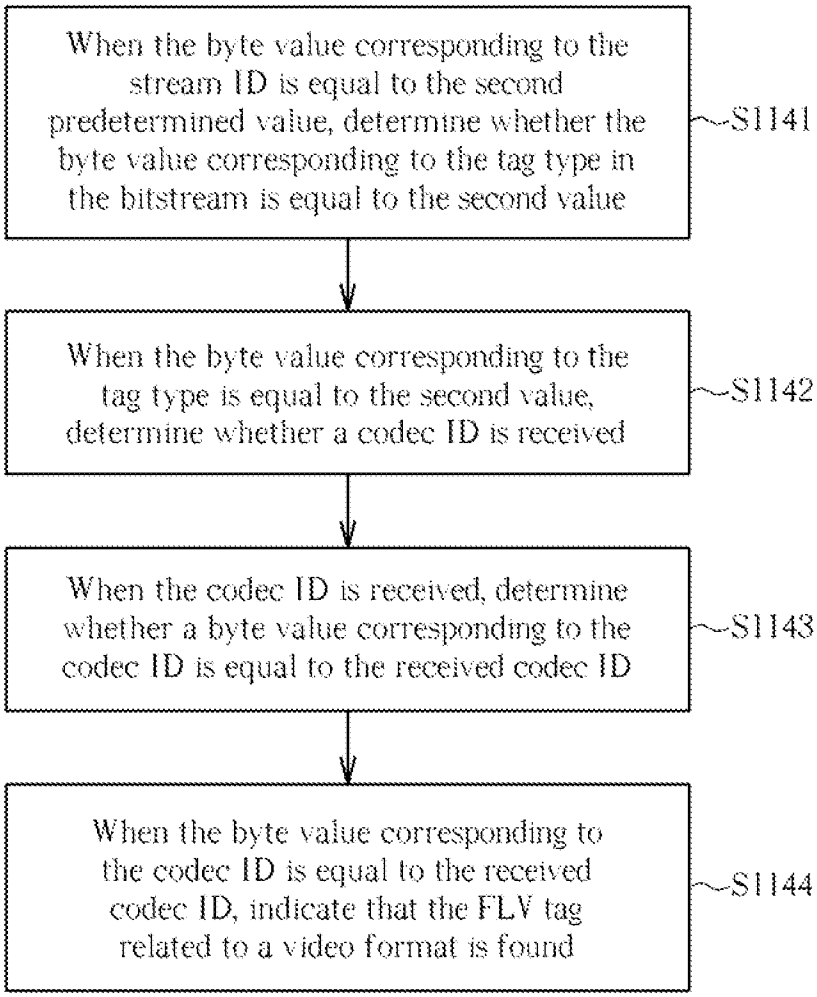
When the byte value corresponding to the stream ID is equal to the second predetermined value, determine whether the byte value corresponding to the tag type in the bitstream is equal to the second value
S1141
When the byte value corresponding to the tag type is equal to the second value, determine whether a codec ID is received
S1142
When the codec ID is received, determine whether a byte value corresponding to the codec ID is equal to the received codec ID
S1143
When the byte value corresponding to the codec ID is equal to the received codec ID, indicate that the FLV tag related to a video format is found
S1144

FIG. 11D

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2010/075184

A. CLASSIFICATION OF SUBJECT MATTER

H04N 7/26 (2006.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC: H04N,H04Q,H04W,H04L,G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNPAT,WPI,EPODOC,IEEE,CNKI:FLV, flash w video, position, locat+,start+, head+,length, size, dimension,search+, hunt+, find +,abstract+,obtain+,tag,identify+,ID, label, playback+,replay+, reconstruct+, reproduc+

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	CN101778289A (UNIV. SHANGHAI JIAOTONG), 14 Jul. 2010 (14.07.2010) the whole document	1-26
A	CN101312531A (INNOFIDEI TECHNOLOGY CO., LTD.), 26 Nov. 2008 (26.11.2008) the whole document	1-26
A	CN101415069A (UNIV. QINGHUA et al.), 22 Apr. 2009 (22.04.2009) the whole document	1-26
A	CN101246491A (MENG, Zhiping), 20 Aug. 2008 (20.08.2008) the whole document	1-26
A	KR20090055544A (DIGITALARIA CO., LTD.), 02 Jun. 2009 (02.06.2009) the whole document	1-26
A	US2009254578A1 (HALL, Michael Andrew), 08 Oct. 2009 (08.10.2009) the whole document	1-26

☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim (S) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
22 Mar.2011 (22.03.2011)	**12 May 2011 (12.05.2011)**
Name and mailing address of the ISA/CN The State Intellectual Property Office, the P.R.China 6 Xitucheng Rd., Jimen Bridge, Haidian District, Beijing, China 100088 Facsimile No. 86-10-62019451	Authorized officer XIA,Kan Telephone No. (86-10)62413461

Form PCT/ISA /210 (second sheet) (July 2009)

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2010/075184

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
CN101778289A	14.07.2010	NONE	
CN101312531A	26.11.2008	NONE	
CN101415069A	22.04.2009	NONE	
CN101246491A	20.08.2008	NONE	
KR20090055544A	02.06.2009	JP20 10268420A	25.11 .2010
US2009254578A1	08.10.2009	NONE	