



(51) International Patent Classification:

H04N 21/8547 (2011.01) H04N 21/242 (2011.01)
H04N 7/56 (2006.01) H04N 21/2368 (2011.01)
H04N 21/434 (2011.01)

(21) International Application Number:

PCT/FI2014/050138

(22) International Filing Date:

21 February 2014 (21.02.2014)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

20135153 21 February 2013 (21.02.2013) FI

(71) Applicant: LINKOTEC OY [FI/FI]; Lars Sonckin kaari
14, FI-02600 Espoo (FI).

(72) Inventor: OJANEN, Jukka; c/o LinkoTec Oy, Lars Son-
ckin kaari 14, FI-02600 Espoo (FI).

(74) Agent: KOLSTER OY AB; (Iso Roobertinkatu 23), P.O.
Box 148, FI-00121 Helsinki (FI).

(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, BN, 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, IR, IS, JP, KE, KG, 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, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA,
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, RW, SD, SL, SZ, TZ,
UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, 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, RS, SE, SI, SK, SM,
TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW,
KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

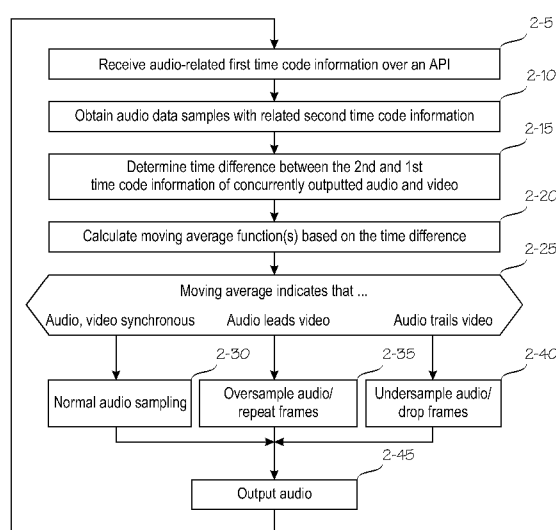
— as to applicant's entitlement to apply for and be granted a
patent (Rule 4.17(ii))

Published:

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

(54) Title: SYNCHRONIZATION OF AUDIO AND VIDEO CONTENT

Fig. 2



(57) Abstract: Synchronizing audio and video comprising: processing video content by a HTTP client in a data processing device. The video content comprises video data frames with a first timecode. A HTTP Server Application in the device (CT) offers an Application Programming Interface ["API"] to the HTTP client and receives (2-10) audio content, which comprises audio data samples associated with a second timecode. The HTTP Server produces pulse-code modulated ["PCM"] audio frames from the audio data samples; receives (2-5) the first timecode from the HTTP client over the API; repeatedly determines (2-15) a time difference between the second and first timecode; and calculates (2-20) a moving average of the repeatedly determined time difference. If the moving average function indicates that the PCM audio data frames lead or trail the outputted video data frames, the PCM audio data frames are oversampled (2-30) or undersampled (2-40) respectively, and otherwise they are sampled normally (2-45).

SYNCHRONIZATION OF AUDIO AND VIDEO CONTENT**FIELD OF THE INVENTION**

[0001] The present invention generally relates to multimedia playback (reproduction) and particularly to methods, apparatuses and software for synchronization between audio and video content obtained from different sources.

BACKGROUND OF THE INVENTION

[0002] A dedicated DVD or Blu-Ray player, or a computer program decoding and reproducing video and audio content from a DVD or Blu-Ray disc, can play back discrete multi-channel audio and video content. In the present context, multi-channel audio content means more than the left and right channels commonly associated with stereo audio. For instance, notions like m.n. (eg 5.0, 5.1, 7.1, etc.) are used to denote the number of audio channels, the integer part denoting the number of audio channels with full frequency range and the decimal part (usually zero or one) denoting the number subwoofer channels with the frequency range limited to frequencies whose contribution to spatial sensation is small and which are not easily reproducible by small speakers. Discrete multi-channel audio means a coding/decoding scheme wherein the m or m+n channels are decoded into separate information streams, as opposed to, say, decoding schemes wherein the front left and right channels are amplitude-encoded into separate information streams, while the rear left and right channels are phase-encoded into the information streams whose amplitude modulation carries the front audio channels.

[0003] Instead of playing back video and audio from a physical DVD or Blu-Ray disc, consumers of multimedia content are increasingly obtaining content over wide-area networks, such as the internet. Unfortunately, multimedia content delivery over the internet is limited to two discrete channels at best. In other words, current technologies for delivering multimedia content over the internet does not support more than two discrete channels.

[0004] ITU (International Telecommunication Union) terminology database defines lip synchronization ("lip-sync") as follows: "Operation to provide the feeling that the speaking motion of the displayed person is synchronized with that person's voice. Minimization of the relative delay between the visual display of a person speaking and the audio of the voice of the person speaking. The objective is to achieve a natural relationship between the visual image and the aural message for the viewer/listener."

[0005] Standards organizations, including ITU, ATSC, SMPTE, EBU, IEC, AES, and

SCTE, have been studying the problem for many years, and there are several standards and recommended practices for tolerances that should be adopted for different parts of the broadcast chain.

5 [0006] The characterization of sensitivity to the alignment of sound and picture includes early work at Bell Laboratories. For film, ITU-R Recommendation BR.265-9 (and its earlier versions) states that the accuracy of the location of the sound record and corresponding picture should be within ± 0.5 film frames, or about ± 22 ms.

10 [0007] In 1998, ITU-R published BT.1359 recommendation, which states that the threshold of detection ability of lip sync errors is about +45ms to -125ms (+: audio early, -: audio late) and that the threshold of acceptability is about +90ms to -185ms, on the average. However, because of the uncertainty of synchronization that may exist in the source material, the ITU recommends that the timing difference in the path from the output of the final program source selection element to the input to the transmitter for emission should be kept within the values +22.5ms and -30ms.

15 [0008] The ATSC Implementation Subcommittee IS-191 in 2003 found that under all operational situations, the sound program should never lead the video program by more than 15ms and should never lag the video program by more than 45ms. According to the IS-191, BT.1359 "was carefully considered and found inadequate for purposes of audio and video synchronization for DTV broadcasting."

[0009] EBU Technical Recommendation R37-2007 gives a range of +40ms and -60ms at the output intended to feed broadcasting transmitters.

25 [0010] Other research shows similar but not identical results, and being a function of human perception, one should expect the results to vary. In 2009 BBC Research White Paper WHP 176 notes that lip-sync errors are dependent upon the distance to a display, the nature of a scene (close-up of a talking head compared to a medium or long shot), image resolution (SD or HD), whether the image is 2D or 3D, among other factors.

30 [0011] The present disclosure acknowledges the fact that there are several sources of uncertainty to gain accurate synchronization of audio to video. Operation of synchronization is dependent on the devices environments in which the devices are used.

35 [0012] One of the problems associated with the above arrangement is that in connection with video or multimedia playback from an external source, current internet technologies do not support discrete multichannel audio, wherein multi-

channel means more than the two channels of a stereo system. A related problem is that the external source may not offer audio in a language desired by the user of the client terminal. If the original audio source is replaced by another audio source, the need for accurate synchronization in a data processing device emerges as another problem. Accurate synchronization is problematic in general-purpose data processing devices wherein multiple processes compete for processing resources, which may lead to variations in execution times.

DISCLOSURE OF THE INVENTION

[0013] An object of the present invention is thus to eliminate or mitigate at least one of the problems identified above. This object is attained by aspects of the inventions as defined in the attached independent claims. The dependent claims and the following detailed description and drawings relate to specific embodiments which solve additional problems and/or provide additional benefits.

[0014] An aspect of the invention is a method for synchronizing audio and video in a data processing device. The method comprises a number of acts performed by the data processing device. These acts are labeled here for the purpose of discussion only, and such labeling does not restrict execution of the steps, unless execution of a step clearly requires an outcome of an earlier step:

a) receiving and processing video content and a first audio content by at least one HTTP client in the data processing device:

b) wherein the processing of the received video content and the first audio content comprises a first synchronization of the first audio content with the video content, and based on the first synchronization, outputting video data frames associated with a first time code information.

[0015] The method further comprises the following acts performed by a HTTP Server Application in the data processing device:

c) offering an Application Programming Interface ["API"] to the at least one HTTP client;

d) receiving audio content from an audio source, wherein the received audio content comprises audio data samples associated with a second time code information;

e) producing pulse-code modulated ["PCM"] audio frames based on the received audio data samples.

[0016] The acts performed by the HTTP Server Application in the data processing device further comprise a second synchronization of the PCM audio frames with the video data frames, wherein the second synchronization comprises:

- 5 f) receiving the first time code information or its derivative from the at least one HTTP client over the API;
- g) repeatedly determining a time difference between the second time code information and the first time code information respectively associated with concurrently outputted PCM audio data frames and video data frames;
- 10 h) calculating at least one moving average function based on the repeatedly determined time difference;
- i) in response to the at least one moving average function indicating that the PCM audio data frames lead or trail the outputted video data frames, oversampling or undersampling, respectively, the PCM audio data frames,
- 15 and otherwise sampling the PCM audio data frames normally.

[0017] An illustrative but non-exhaustive list of representative data processing devices includes a personal computer, a laptop computer, a palmtop computer or a tablet computer, a smart phone, a media player with online connections, or the like. The data processing device comprises a video or multimedia player application which is capable of acting as a HTTP client. Apart from the ability of acting as a HTTP client, steps a) and b) represent conventional functionality of a virtually any video or multimedia player application. The video content may be obtained from a source internal to the data processing device or external to it. Examples of internal sources include optical disks residing in an optical disk drive and locally stored multimedia files, such as those stored in MP4, DivX, or comparable formats. Examples of external sources include online repositories accessed via content servers.

[0018] According to the invention, an HTTP server application in the data processing device offers an application programming interface ["API"] to the video or multimedia player application, which acts as the HTTP client. The HTTP server application in the data processing device performs steps labeled c) through i).

[0019] Step d) comprises receiving audio content from an audio source, wherein the received audio content comprises audio data samples which are associated with audio-related time code information, which in the present context will be called a second time code information. For instance, the audio data samples may comprise a file or stream of PCM frames or other time-related samples, and the

second time code information comprises a frame or sample number and/or a time stamp from the beginning of the file or stream or a portion of it, such as a track.

[0020] Step e) comprises producing pulse-code modulated ["PCM"] audio frames based on the received audio data samples. In other words, the HTTP server application acts as an audio player application. In itself, outputting the audio samples is normal functionality of an audio player. A problem underlying the invention is that without the additional steps described later, the outputted video and audio are likely to fall out of synchronization with each other. Accordingly, the acts performed by the HTTP Server Application in the data processing device further comprise a second synchronization of the PCM audio frames with the video data frames, wherein the second synchronization comprises steps f) through i).

[0021] The HTTP server application offers an Application Programming Interface ["API"] to the multimedia or video player acting as a HTTP client and receives the first time code information or its derivative from the HTTP client over the API. Although the HTTP server application acting as an audio player application doesn't need or use the video content processed by the video player / HTTP client, the HTTP server application needs the audio-related first time code information or a derivative of it at least until the second synchronization has been accomplished, after which the audio content outputted by the video player or the audio-related first time code may optionally be muted. For instance, the video-related first time code information may be a frame number from the start of the file or stream, whereas a derivative may comprise a modulo function (a number of least significant bits) of the frame number or a time stamp derived from the frame number.

[0022] Herein, the act of sending the first time code information by the video or multimedia player application on one hand, and the act of receiving the first time code information by the HTTP server application acting as the audio player application on the other hand, may be accomplished by techniques used in client/server architectures. For instance, a software company providing the HTTP server application / audio player application may also publish an applet, such as a length of JavaScript code whose execution by the web-enabled video or multimedia player application causes the latter to output the video-related first time code information over the API to the HTTP server application / audio player application.

[0023] Step g) comprises repeatedly determining a time difference between the

second time code information and the first time code information respectively associated with concurrently outputted audio data samples and video data samples. In other words, the HTTP server application compares 1) the second time code information associated with outputted PCM audio data frames with 2) the first time code information associated with video data samples outputted concurrently with the audio data samples. In principle, such comparison indicates whether video or audio or neither is leading or trailing the other. But it is not very useful to jump to conclusions based on individual time differences. For instance, the time differences may be spurious and disappear spontaneously. Furthermore, overly rapid corrective measures may be more disturbing than periods of failed synchronization between the audio and video outputs. Accordingly, step h) comprises calculating at least one moving average function based on the repeatedly determined time difference. In the present context, a "moving average function" means a function whose result has at least 0.5 correlation with a mathematically precise definition of moving average. Reasons for deviating from a mathematically precise moving average may include a desire to reduce processing load by substituting coarse table lookup values for accurate multiplication and division results, for example.

[0024] In step i) the moving average function(s) is/are used to indicate if the outputted audio data samples lead or trail the outputted video data samples, or are synchronous with them. If the moving average function(s) indicate that the outputted audio data samples lead the outputted video data samples, a corrective measure is taken by oversampling the received audio data samples. Oversampling means that a set of audio samples is processed to yield a set of processed audio samples whose combined length is longer than the nominal lengths of the audio samples. Conversely, if the moving average function(s) indicate that the outputted audio data samples trail the outputted video data samples, a corrective measure is taken by undersampling the received audio data samples. Undersampling means that a set of audio samples is processed to yield a set of processed audio samples whose combined length is shorter than the nominal lengths of the audio samples.

[0025] If the outputted audio is in synchronization with the outputted video, no corrective measure needs to be taken and the audio data is sampled normally, that is, without oversampling or undersampling.

[0026] A simple implementation of the invention may utilize one moving average, while more ambitious implementations may involve more, such as three,

moving averages, each moving average calculated over a respective plurality of audio or video samples. For instance, assuming that a “short” moving average, ie, a moving average calculated over a small number of samples, indicates that audio output is leading or trailing video output, a relative small corrective measure may be taken. In contrast, if a “long” moving average, ie, a moving average calculated over a large number of samples, indicates that audio output is leading or trailing video output, a larger corrective measure may be taken.

[0027] There are several alternative technologies for implementing the inventive audio replacement and synchronization scheme, such as independent applications or “apps”, browser extensions (plugins), or the like. The inventor has discovered, however, that independent apps are not supported or allowed on all platforms. Many platforms allow web browsers to support plugins that extend the functionality of the web browser. Depending on the browser and the version, the term may be add-on (addon), extension or plugin. Web browsers provide programming interfaces (APIs) allowing third parties to create plugins that interact with the browser. These APIs are vendor-specific, and usually incompatible across browsers. For instance, Microsoft Internet Explorer supports ActiveX extensions, Mozilla Firefox supports NPAPI plug-ins, and Google Chrome supports PPAPI plug-ins. There is no single API that is supported by all web browsers. An additional problem is that on closed platforms, such as most mobile platforms, the web browsers don’t support any plugins at all.

[0028] Accordingly the inventor has discovered that an implementation of the inventive audio replacement and synchronization scheme as an HTTP server has the broadest support across different platforms. Somewhat counter-intuitively, the HTTP server typically resides in the same data processing device, such as a mobile terminal, that also hosts the client component.

[0029] Other aspects of the invention include an apparatus and a computer program product for carrying out the inventive method.

BRIEF DESCRIPTION OF THE DRAWINGS

[0030] In the following the invention will be described in greater detail by means of specific embodiments with reference to the attached drawings, in which Figure 1 shows an exemplary network architecture in which the invention can be used;

Figure 2 is a flow chart illustrating an operating principle according to an embodiment of the invention;

Figure 3 is a block diagram illustrating an exemplary hardware implementation; and

Figure 4 is another block diagram showing major hardware blocks in a data processing device.

5 DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS

[0031] Figure 1 shows an exemplary network environment in which the invention and its embodiments can be used. In Figure 1, a client terminal CT accesses a video repository VR. The term video repository VR does not exclude audio, and a typical video repository VR indeed stores audio content as well. Instead, the term
10 video repository VR implies that the invention and its embodiments enable use of video content from the video repository VR and audio from an audio repository AR, which is or may be a repository entirely distinct from the video repository.

[0032] In the network environment shown in Figure 1 the video repository VR is a remote repository, typically implemented as a data base server, which the client
15 terminal CT accesses over a data network DN and, assuming that the client terminal CT is a mobile terminal, over an access network AN. In a typical scenario, the data network DN is the internet or its closed subnetworks, commonly called intranets or extranets. The access network AN may be a cellular mobile network or a wireless local-area network (WLAN), to name just a few of the available options.
20 In some cases the video repository VR is a local memory resource in the client terminal CT, such as a local DVD or Blu-Ray disc or a multimedia file, such as an MP4 or DivX file, stored locally in the client terminal. The client terminal CT comprises means for video or multimedia playback. Formerly, when data processing devices, and particularly mobile ones, suffered from inferior processing power compared with bigger devices, video or multimedia playback was typically im-
25 plemented via dedicated playback circuitry. Now that even hand-held devices offer adequate processing power, video or multimedia playback is typically implemented in software, such as via applications, applets or browser plug-ins. It should be understood, however, that the list of possible transmission, storage or
30 processing technologies is merely illustrative and by no means exhaustive.

[0033] A key issue in the scenario shown in Figure 1 is that when the video repository VR is external to the client terminal CT, and the client terminal accesses the video repository over internet protocols, current technologies do not support discrete multichannel audio, wherein multichannel means more than the two
35 channels of a stereo system. A related problem is that the video repository VR

may not offer audio in a language desired by the user of the client terminal. In order to support audio content in a format or language desired by the user of the client terminal CT, such as discrete multichannel audio or audio in the user's preferred language, the network environment shown in Figure 1 comprises an audio repository AR, from which audio content is retrieved to the client terminal CT.

[0034] A residual problem is how to synchronize the video content obtained from the video repository VR and audio content obtained from the audio repository AR. The synchronization problems are most acute over poor remote connections, but they are not totally absent in cases where the video repository VR and/or the audio repository AR are stored locally in the client terminal. For instance, the client terminal's processor may be multitasking, and another process requires processing power to the extent that playback may be interrupted for short intervals.

[0035] Figure 2 is a flow chart illustrating an operating principle according to an embodiment of the invention. Figure 2 shows the processing steps carried out by the audio playback device. More particularly, Figure 2 shows the processing steps that relate to maintaining synchronization, and the steps of converting a digital audio stream to analog audio signals is presumed known to the skilled reader.

[0036] In step 2-5 an audio player 3-200, acting as an HTTP server, receives video-related time-code information, called first time-code information in the following, over an application programming interface (API, see item 3-150 in Figure 3). In step 2-10 the audio player obtains audio-related time-code information, called second time-code information. In step 2-15 the audio player determines a time difference between the second and first time-code information of concurrently outputted audio and video streams. In order to minimize effects of spurious events, steps 2-20 comprises calculating a time-filtered version, such as one or more moving average functions, from the time difference determined in the preceding step. In step 2-25 the time-filtered version(s), such as the moving average function(s) is/are analyzed.

[0037] If the analysis in step 2-25 shows that audio and video are synchronized with each other, the process proceeds to step 2-30, wherein the incoming audio samples are processed (sampled) normally. This is the normal processing of audio samples that is presumed known to the skilled reader. In other words, step 2-30 performs what a conventional audio player does.

[0038] If the analysis in step 2-25 shows that audio leads (is ahead of) video, the process proceeds to step 2-35, wherein the incoming audio samples are over-

sampled. Oversampling of incoming audio samples has the effect that processing of a given set of audio samples lasts longer than by normal processing, and the audio leads video by a progressively smaller amount of time. Over time, the time difference determined in step 2-15 changes sign and the moving average(s) calculated in step 2-20 approach zero. If the audio signal leads video by a significant margin, the process may repeat entire frames. Each repeated frame doubles the amount of time that frame is outputted, and reduces the time that audio is ahead of video by the duration of the frame.

[0039] On the other hand, if the analysis in step 2-25 shows that audio trails (is behind) video, the process proceeds to step 2-40, wherein the incoming audio samples are undersampled. Undersampling of incoming audio samples has the effect that processing of a given set of audio samples lasts less time than by normal processing, and the audio trails video by a progressively smaller amount of time. Over time, the time difference determined in step 2-15 changes sign and the moving average(s) calculated in step 2-20 approach zero. If the audio signal trails video by a significant margin, the process may drop entire frames. Each dropped frame advances audio (reduces lag) in relation to video by the duration of the frame.

[0040] Whichever sampling (normal, over- or undersampling) mode is selected in steps 2-25 to 2-40, the sampled audio signal is outputted in step 2-45 and the process returns to step 2-15.

[0041] Figure 3 is a block diagram illustrating an exemplary hardware implementation of a data processing device, such as the client terminal CT shown in Figure 1. Reference numeral 3-100 denotes a video player, which may be implemented as an external application (a HTTP client) to the audio player acting as an HTTP Server 3-200. Outputs of a typical video player 3-100 include a video output 3-130 and an audio output 3-132, which the video player 3-100 synchronizes with each other by an internal audio-to-video synchronization mechanism. For the purposes of the present context, the video output 3-130 of the video player 3-100 is utilized normally, that is, typically outputted to an output device. While the video player 3-100 also generates an audio output 3-132, it may be replaced by a different audio output, denoted by reference number 3-230, which is based on a different audio stream, denoted by reference number 3-202.

[0042] Processing of the incoming audio stream 3-202 in the audio player 3-200 involves the following blocks. Dashed outline around some of the blocks indicates that the block is involved in audio-video synchronization and not merely to con-

ventional audio processing.

[0043] A source buffer or cache 3-210 permits the audio player to store incoming audio packets, by virtue of which the audio player can survive brief communication delays without audible gaps in outputted audio. An AV demultiplexer extracts a selected audio stream from an incoming AV bitstream, which may contain multiple bitstreams, such as audio streams in several languages. Reference number 3-214 denotes a frame dropper, which is used in situations wherein audio significantly trails video. By dropping audio for some frames, the outputted audio can be made to quickly catch up with the video. Operation of the frame dropper 3-214 is controlled by the synchronization controller 3-240. The audio frames are conveyed to an audio decoder 3-216 which in this example outputs pulse-code modulated (PCM) frames to a multichannel mixer 3-218 for downmixing or upmixing of audio. Output from the multichannel mixer 3-218 is conveyed to a frame repeater 3-220, whose function is opposite to that of the frame dropper 3-214. In cases where audio significantly leads video, the frame repeater 3-220 repeats some of the frames, thus spending more than a normal amount of time to process a series of audio frames. As a result the outputted audio can be made to slow down to match the time code of the currently played video. Reference number 3-222 denotes a sample rate converter, which the synchronization controller 3-240 activates in cases where the outputted audio leads or trails video by an amount which is not large enough to activate the frame dropper 3-214 or the frame repeater 3-220. The sample rate converter has the effect that, instead of dropping or repeating whole frames, individual frames are played for a longer or shorter time, depending on whether the outputted audio leads or trails the video. Finally, the PCM frames processed by the audio player 3-200 are conveyed to an audio playback device generally denoted by reference number 3-224 and outputted as audio output 3-230.

[0044] The audio player / HTTP Server 3-200 offers an Application Programming Interface ["API"] 3-150 to the video player / HTTP client 3-100. In the present illustrative but non-restrictive example, the notion "RESTful" signifies that the software architecture conforms to REST constraints. REST, on the other hand, refers to "REpresentational State Transfer", which is a style of software architecture for distributed systems such as the World Wide Web. Although REST is by no means the only design model, it has emerged as a predominant model for Web service design. The term "representational state transfer" was introduced in the doctoral dissertation of Roy Fielding, who is one of the principal authors of the

Hypertext Transfer Protocol (HTTP) specification versions 1.0 and 1.1. By means of the API 3-150, the video player provides the audio player 3-200 with audio-related time code 3-160, which may be the audio output 3-132 itself or some related or derivative timing signal from the audio output 3-132.

5 **[0045]** The synchronization controller 3-240 obtains the audio-related time code 3-160 via the API 3-150, and calculates a time-delay estimation (TDE). An exemplary technique for the TDE estimation is based on Generalized Cross-Correlation (GCC). This technique is presented in reference document #1 (identified at the end of this description).

10 **[0046]** To further improve stability and accuracy of the synchronization between Video Player 3-100 and Audio Player 3-200, some implementations of the invention may deliver the audio output signal 3-132 to the synchronization controller 3-240. The synchronization controller 3-240 is then able to calculate a time-delay estimate (TDE) between the audio output 3-132 of the video player 3-100 and the
15 audio output 3-230 of the audio player 3-200, or some internally processed signal, such as the output of the audio decoder 3-216.

[0047] For example World Wide Web Consortium (W3C) has specified a Web Audio API which describes a high-level JavaScript API for processing and synthesizing audio in web applications. Internet Engineering Task Force (IETF) in RFC
20 6455 specifies WebSocket protocol. The WebSocket Protocol enables two-way communication between a client and a server. The WebSocket Protocol is an independent TCP-based protocol, but its handshake is interpreted by HTTP servers as an Upgrade request. Some implementations of the video player might use these two protocols to route the audio output 3-132 by using WebSocket to the
25 synchronization controller 3-240.

[0048] In order to compute a delay between two audio signals, both audio signals may be downmixed to a single channel (mono). A stereo signal can be downmixed to mono by summing the left and right channel, and attenuating the signal by -6 dB, or by any of a number of equivalent processes.

30 **[0049]** Some implementations of the invention are based on the assumption that if a multichannel audio content is available, that multichannel audio content was used to produce the downmixed stereo. The downmixing scheme (downmixing matrix) used to produce the stereo may also be known and this knowledge may be used to correctly downmix the multichannel audio to stereo.

35 **[0050]** In some other implementations the downmixing scheme is not known. In such implementations the downmixing scheme may be assumed to be one of Dol-

by® Surround, Dolby® Pro Logic or Dolby® Pro Logic II scheme, which are the ones most widely used in this field, all being registered trademarks of Dolby Laboratories, Inc. The use of these schemes may be detected using a decoder of the downmixing scheme. Furthermore, the number of channels in multichannel content may be used to determine the downmixing scheme for multichannel audio.

[0051] An incorrect choice of downmixing and other imperfections can be interpreted as noise. Thus to improve robustness and stability of correlation, phase transformation (PHAT) may be used as the frequency weighting function of GCC. Such a method is known as GCC-PHAT. This technique is presented in reference document #2.

[0052] Given two signals, $\chi_i(n)$ and $\chi_j(n)$, the GCC-PHAT is defined as:

$$G_{PHAT}(f) = \frac{X_i(f)[X_j(f)]^*}{|X_i(f)[X_j(f)]^*|}$$

[0053] Herein, $X_i(f)$ and $X_j(f)$ are the Fourier transforms of the two signals, while $[]^*$ denotes the complex conjugate. The delay for these signals can be estimated as:

$$\hat{d}_{PHAT}(i, j) = \underset{d}{argmax} \left(\hat{R}_{PHAT}(d) \right)$$

Herein, $\hat{R}_{PHAT}(d)$ is the inverse Fourier transform of $G_{PHAT}(f)$.

[0054] To make sure that a low correlation between signals does not lead to poor delay estimation, a cross correlation threshold may be employed. The cross correlation threshold is the minimum correlation value at which the correlation is expected to return feasible results. Efficient computation of GCC-PHAT may be accomplished by using zero padded Fast Fourier Transform (FFT), scaling and inverse FFT. An FFT function is known to have a complexity of $O(N \log N)$. As the aim of this technique synchronize playback, it is to run in real time. If delay estimation of longer delays is required, a longer window is required, which increases the required amount of processing. Thus the maximum delay estimation using GCC-PHAT is limited by the processing capabilities of the device. But as this synchronization method is secondary to time-code based synchronization, the maximum window size can be adapted to the processing capabilities, while switching between synchronization methods can be done accordingly.

[0055] Reference numeral 3-250 denotes a system hardware clock of the data processing device CT. Although the system hardware clock may be used for other purposes, it is an unsuitable clock for the purpose of synchronizing audio and video. Instead, a synchronization controller denoted by reference numeral 3-240

performs the audio-to-video synchronization.

[0056] The presently described embodiment utilizes a clock synchronization algorithm based on an internal clock model, where a goal is to minimize the maximum difference between two clocks. Clock synchronization is asymmetric, which is also known as a master-slave synchronization scheme. In the embodiment shown in Figure 3, the video player is the master while the audio player is the slave. The presently described embodiment requires minimal changes to the video player 3-100, compared with a typical prior art video player. The clock synchronization process uses a technique commonly known as a slave-controlled scheme, where the master (the video player 3-100) provides a reference time to which the slave (the audio player 3-200) synchronizes.

[0057] The embodiment of Figure 3 imposes few if any restrictions regarding the source from which the video player receives the video content or the manner in which the video content is retrieved. In order to synchronize the audio player 3-200 with the video player 3-100 (and the video content), the audio player 3-200 and the video player 3-100 should use the same time reference. In an illustrative but non-restrictive example, the video player 3-100 receives a packetized stream of multiplexed audio and video in an MPEG-2 or MPEG-4 container. These container formats may define several different time stamps to control playback rate. Normally the video player 3-100 merely needs to read the time stamps and wait until the right time to output each video frame on the display (inherent feature in a video player, not shown separately). But abnormal situations may occur that cause one or more video frames being displayed at an incorrect time or for an incorrect duration, and this results in slower or faster playback of the video. Examples of such abnormal situations include system overload, transmission delays, reception errors, or the like. Precise reasons and natures of failed synchronization are irrelevant for the purposes of the present invention, however. A video player may compensate for such abnormal situations by adjusting the presentation time of each frame. A residual problem in this kind of compensation technique is creation of deviation (jitter) in the master clock.

[0058] The embodiment of Figure 3 begins a clock synchronization process in such a manner that the video player 3-100 sends its current presentation timestamp (a timestamp of a currently displayed video frame) to the audio player 3-200 using the HTTP Control API 3-150. For the purposes of the present embodiment, such video-related time stamp information is called first time code information. The video player 3-100 is expected to continue sending the first time code

information as long as the video is being played. It is not necessary to send the first time code information for each video frame being played, but it is beneficial to define an upper limit for the interval between transmissions of the first time code information. For instance, the upper limit may range from one to a few seconds, such as two seconds. In an illustrative but non-restrictive example, the video player 3-100 may use watchdog timers to send the first time code information.

[0059] In the embodiment shown in Figure 3, the clock synchronization controller block 3-240 receives the video-related first time-code information and employs a clock synchronization algorithm, which will be described in the following.

On receiving the first time code information (the video-related timestamp), the clock synchronization algorithm employed by the clock synchronization controller 3-240 compares the first time code information $V(i)$ and the second time code information (the audio-related timestamp) $A(i)$ with corresponding previous timestamps $V(i-1)$ and $A(i-1)$ to calculate an estimate of the clock deviation:

$$D(i) = (V(i) - V(i-1)) / (A(i) - A(i-1))$$

[0060] The video player 3-100 is known or expected to have jitter on its timestamps, and since network latencies and operating system schedulers add to this jitter, the estimate of the clock deviation $D(i)$ is smoothed by using moving average filter(s). What follows is a mathematical description of moving average filters, and those skilled in the art realize that approximate quantities may be substituted for mathematically precise quantities in cases where faster processing and/or shorter program routines are desired.

[0061] An exponential moving average (EMA), also known as an exponentially weighted moving average (EWMA) is a type of infinite impulse response filter that applies exponentially decreasing weighting factors. Although the weighting for each older datum point decreases exponentially, it never reaches true zero in a mathematically exact description.

[0062] The EMA for a series D may be calculated recursively

$$E(1) = D(1)$$

$$\text{for } t > 1, E(t) = \alpha * D(t) + (1 - \alpha) * E(t-1),$$

[0063] Here the coefficient α represents the degree of weight decrease, a constant smoothing factor between 0 and 1.

[0064] A more ambitious implementation of the clock synchronization algorithm employs several different weighting factors to estimate clock deviation in different time periods. The system may also simplify EMA calculation by using integer numbers instead of floating-point numbers:

$$E(1) = D(1)$$

for $t > 1$, $E(t) = (\alpha * D(t) + \beta * E(t-1)) / (\alpha + \beta)$,

[0065] Here the coefficients α and β are positive integers. In a typical but non-restrictive implementation, α may have a value of 1, while β has values 3, 15, 31 for three different time periods.

[0066] The resulting filtered value gives an estimation of clock skew. Based on this estimated clock skew, the synchronization controller 3-240 decides to drop or repeat audio frames in order to compensate for the clock skew. If the clock skew is too large, the synchronization controller 3-240 instructs the A/V demultiplexer 3-212 to seek to a new position in the audio content. When the new position has been achieved, and the clock skew has been reduced to a few hundred milliseconds at most, the sample rate converter 3-222 is used to compensate for the clock skew. As long as the sample rate variations is low enough, the change in pitch produced by the sample rate conversion will not be audible. In the field of psycho-acoustics the notion of "just noticeable difference", or JND, is used to describe how large a pitch change can be until it is detected by the human ear. The precise value of the JND depends on the intensity (volume) and frequency of the signal, but a rule of thumb says that changes of up to 0.25% are unnoticeable. Accordingly, a sample rate change of 0.25% means that the system can compensate for clock skew at a rate of 2.5 milliseconds per second. The synchronization controller 3-240 may stop correcting for clock skew when the clock skew estimation changes its sign. In a representative but non-restrictive implementation, as long as the absolute value of the clock skew remains below 20 milliseconds, the synchronization controller 3-240 lets the audio play normally without making any time-related corrections.

[0067] Figure 4 is another block diagram showing major hardware blocks in a data processing device. Figure 4 schematically shows an exemplary block diagram for a data processing device 4-100, such as the client terminal CT, which includes the HTTP server acting as the audio player. The data processing device comprises one or more central processing units CP1 ... CPn, generally denoted by reference numeral 4-110. Embodiments comprising multiple processing units 4-110 are preferably provided with a load balancing unit 4-115 that balances processing load among the multiple processing units 4-110. The multiple processing units 4-110 may be implemented as separate processor components or as physical processor cores or virtual processors within a single component case. The data processing device 4-100 further comprises a network interface 4-120 for communi-

cating with various data networks, which are generally denoted by reference sign DN. The data networks DN may include local-area networks, such as an Ethernet network, and/or wide-area networks, such as the internet. The data processing device may connect to online video and/or audio repositories via data networks DN.

[0068] The data processing device 4-100 of the present embodiment also comprises a user interface 4-125. For video playback the user interface comprises a display. The nature of the user interface depends on which kind of computer is used to implement the data processing device 4-100. For example the user interface 4-125 may comprise a traditional keyboard, mouse and/or touchpad, and/or a touch-sensitive display as used in tablet computers.

[0069] The data processing device 4-100 also comprises memory 4-150 for storing program instructions, operating parameters and variables. Reference numeral 4-160 denotes a program suite for the data processing device 4-100.

[0070] The data processing device 4-100 also comprises circuitry for various clocks, interrupts and the like, and these are generally depicted by reference numeral 4-130. The data processing device 4-100 further comprises a disk interface to the disk system 4-190. The various elements 4-110 through 4-150 intercommunicate via a bus 4-105, which carries address signals, data signals and control signals, as is well known to those skilled in the art.

[0071] The inventive method may be implemented in the data processing device as follows. The program suite 4-160 comprises program code instructions for instructing the set of processors 4-110 to execute the functions of the inventive method, wherein the functions include performing common operating system functions, normal data processor functions, including data entry, presentation and communication. For the purposes of the present invention, the functions include video player functionality, audio player functionality and clock synchronization. Specifically, the functions of the inventive method include the acts defined in claim 1.

[0072] It will be apparent to a person skilled in the art that, as the technology advances, the inventive concept can be implemented in various ways. The invention and its embodiments are not limited to the examples described above but may vary within the scope of the claims.

[0073] Reference documents:

1. C. H. Knapp and G. C. Carter: *"The generalized correlation method for estimation of time delay"* by IEEE Transactions on Acoustics, Speech and

Signal Processing vol. ASSP-24, pp. 320-327, Aug. 1976.

2. G. C. Carter, A. H. Nuttall, and P. G. Cable: *"The smoothed coherence transform"*, Tech. Rep., Naval Underwater Systems Center, New London Lab., 1972.

CLAIMS

1. A method for synchronizing audio and video in a data processing device (CT), the method comprising the following acts performed by the data processing device:

- 5 - receiving and processing video content and a first audio content by at least one HTTP client (3-100) in the data processing device;
- wherein the processing of the received video content and the first audio content comprises a first synchronization of the first audio content with the video content, and based on the first synchronization, outputting video
- 10 data frames (3-130) associated with a first time code information (3-160);

the method further comprising the following acts performed by a HTTP Server Application (3-200) in the data processing device (CT):

- offering an Application Programming Interface ["API"] (3-150) to the at least one HTTP client (3-100);
- 15 - receiving (2-10) a second audio content (3-202) from an audio source (AR), wherein the second audio content comprises audio data samples associated with a second time code information;
- producing pulse-code modulated ["PCM"] audio frames (3-230) based on the received audio data samples;

20 wherein the acts performed by the HTTP Server Application (3-200) in the data processing device (CT) further comprise a second synchronization of the PCM audio frames (3-230) with the video data frames (3-130), wherein the second synchronization comprises:

- receiving (2-5) the first time code information (3-160) or its derivative
- 25 from the at least one HTTP client (3-100) over the API (3-150);
- repeatedly determining (2-15) a time difference between the second time code information and the first time code information respectively associated with concurrently outputted PCM audio data frames (3-230) and video data frames (3-130);
- 30 - calculating (2-20) at least one moving average function based on the repeatedly determined time difference;
- in response to the at least one moving average function indicating that the PCM audio data frames lead or trail the outputted video data frames, oversampling (2-30) or undersampling (2-40), respectively, the PCM audio data
- 35 frames, and otherwise sampling the PCM audio data frames normally (2-45).

2. The method according to claim 1, wherein the received audio content comprises audio data samples for at least two discrete audio channels.
3. The method according to claim 2, wherein the received audio content comprises audio data samples for more than two discrete audio channels.
- 5 4. The method according to any one of the preceding claims, wherein each moving average function is calculated for a respective plurality of video data frames and audio data frames.
5. The method according to any one of the preceding claims, further comprising receiving the video content and the audio content from different sources.
- 10 6. The method according to any one of the preceding claims, wherein the over-sampling (2-30) or undersampling (2-40) respectively comprises repeating or dropping audio data frames if the at least one moving average function exceeds a predetermined threshold.
7. The method according to claim 6, wherein the method comprises repeating or
15 dropping no more than one consecutive audio data frame.
8. A data processing device (CT), comprising:
- a memory system (4-150) for storing program code instructions and data;
 - a processing system (4-110) comprising at least one processing unit, wherein the processing system is configured to execute at least some of
20 the program code instructions and to process the data stored in the memory system;
 - wherein the data processing device (CT) comprises or cooperates with a video player (3-100) configured to receive and process video content and a first audio content by at least one HTTP client (3-100) in the data processing device, wherein the processing of the received video content and
25 the first audio content comprises a first synchronization of the first audio content with the video content, and based on the first synchronization, outputting video data frames (3-130) associated with a first time code information (3-160);
- 30 wherein the memory system (4-150) comprises a HTTP Server Application (3-200) executable by the processing system, wherein the HTTP Server Application is configured to:

- offer an Application Programming Interface ["API"] (3-150) to the at least one HTTP client (3-100);
- receive (3-10) a second audio content (3-202) from an audio source (AR), wherein the second audio content comprises audio data samples associated with a second time code information;
- produce pulse-code modulated ["PCM"] audio frames (3-230) based on the received audio data samples;

wherein the HTTP Server Application (3-200) is further configured to perform a second synchronization of the PCM audio frames (3-230) with the video data frames (3-130) wherein the second synchronization comprises:

- receiving (3-5) the first time code information (3-160) or its derivative from the at least one HTTP client (3-100) over the API (3-150);
- repeatedly determining (2-15) a time difference between the second time code information and the first time code information respectively associated with concurrently outputted PCM audio data frames (3-230) and video data frames (3-130);
- calculating (3-20) at least one moving average function based on the repeatedly determined time difference;
- in response to the at least one moving average function indicating that the PCM audio data frames lead or trail the outputted video data frames, oversampling (2-30) or undersampling (2-40), respectively, the PCM audio data frames, and otherwise sampling the PCM audio data frames normally (2-45).

9. A tangible program carrier embodying instructions for a data processing device, which comprises a memory system (4-150) for storing program code instructions and data; and a processing system (4-110) comprising at least one processing unit, wherein the processing system is configured to execute at least some of the program code instructions and to process the data stored in the memory system; wherein the data processing device (CT) comprises or cooperates with a video player (3-100) configured to receive and process video content and a first audio content by at least one HTTP client (3-100) in the data processing device, wherein the processing of the received video content and the first audio content comprises a first synchronization of the first audio content with the video content, and based on the first synchronization, outputting video data frames associated with a first time code information (3-160);

wherein the tangible program carrier (4-150) comprises a HTTP Server Application (3-200) executable by the processing system, wherein the HTTP Server Application comprises program code instructions configured to:

- 5 - offer an Application Programming Interface ["API"] (3-150) to the at least one HTTP client (3-100);
- receive (3-10) a second audio content (3-202) from an audio source (AR), wherein the second audio content comprises audio data samples associated with a second time code information;
- 10 - produce pulse-code modulated ["PCM"] audio frames (3-230) based on the received audio data samples;

wherein the HTTP Server Application (3-200) is further configured to perform a second synchronization of the PCM audio frames (3-230) with the video data frames (3-130) wherein the second synchronization comprises:

- 15 - receiving (3-5) the first time code information (3-160) or its derivative from the at least one HTTP client (3-100) over the API (3-150);
- repeatedly determining (2-15) a time difference between the second time code information and the first time code information respectively associated with concurrently outputted PCM audio data frames (3-230) and video data frames (3-130);
- 20 - calculating (3-20) at least one moving average function based on the repeatedly determined time difference;
- in response to the at least one moving average function indicating that the PCM audio data frames lead or trail the outputted video data frames, oversampling (2-30) or undersampling (2-40), respectively, the PCM audio data frames, and otherwise sampling the PCM audio data frames normally (2-45).
- 25

1/3

Fig. 1

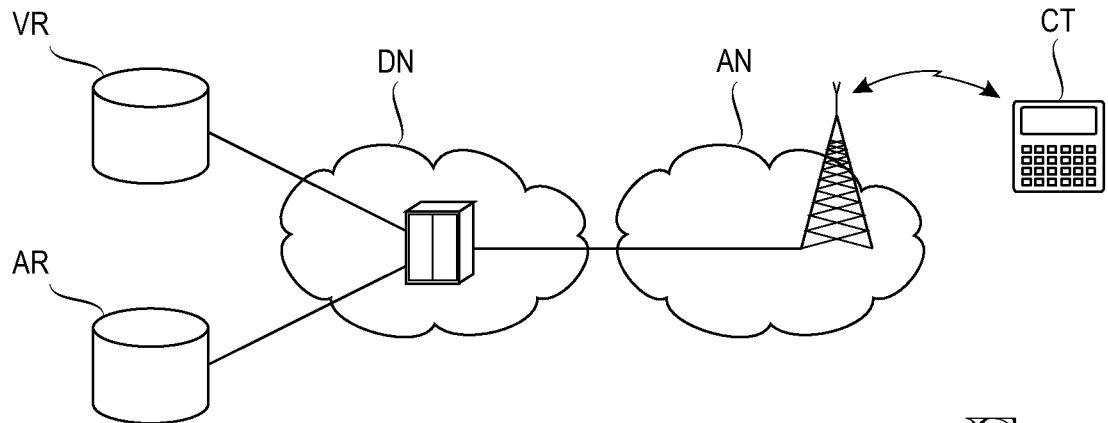
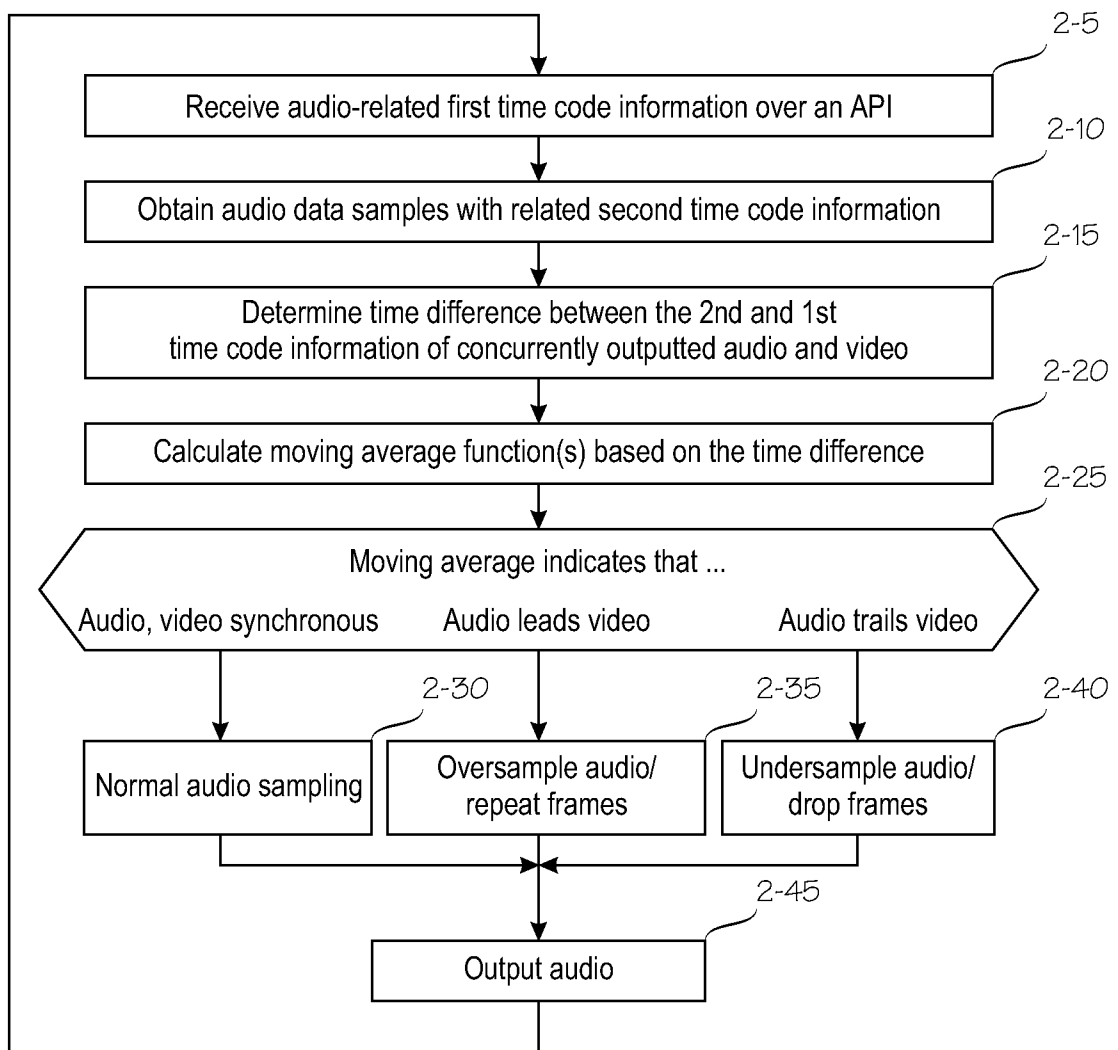
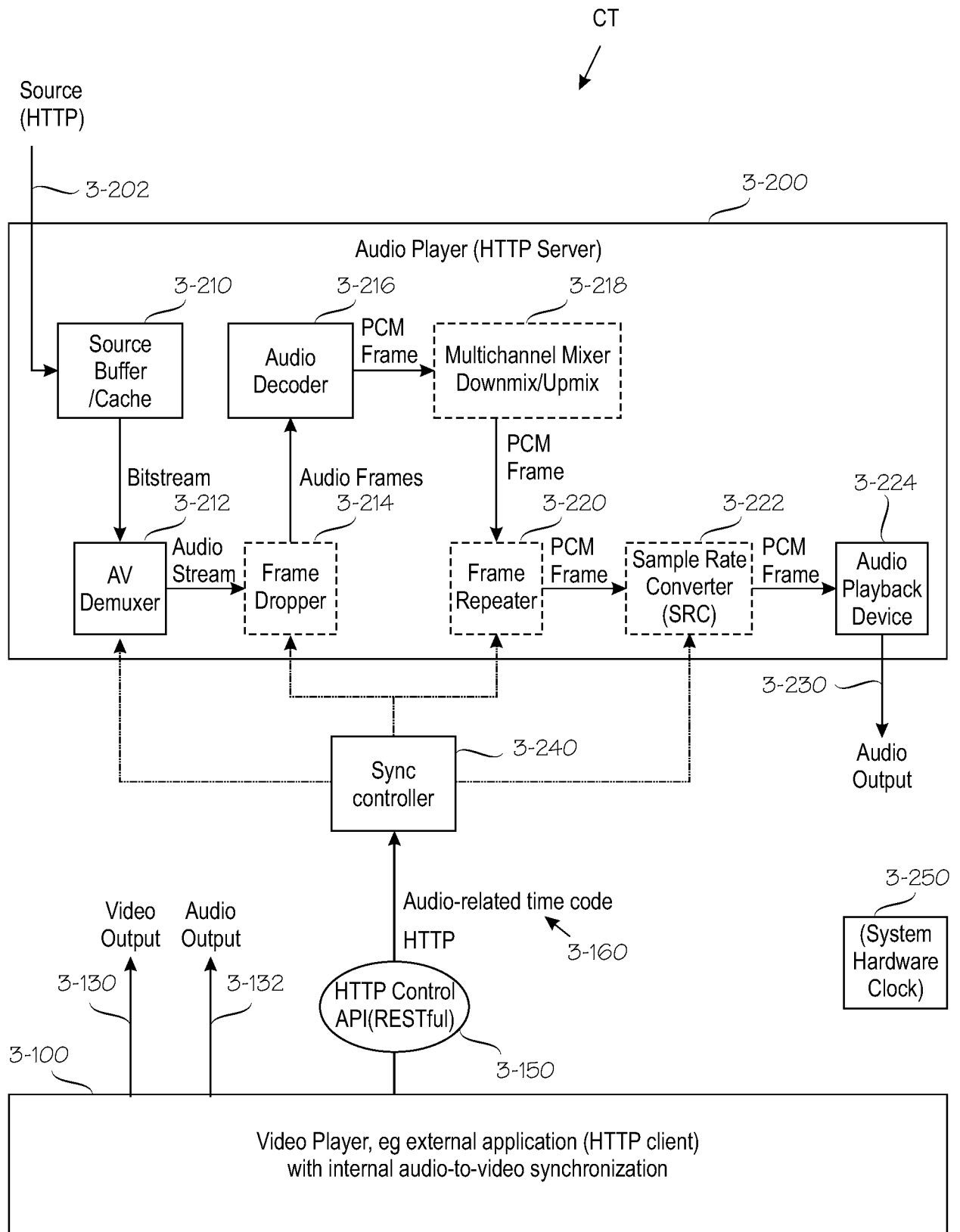


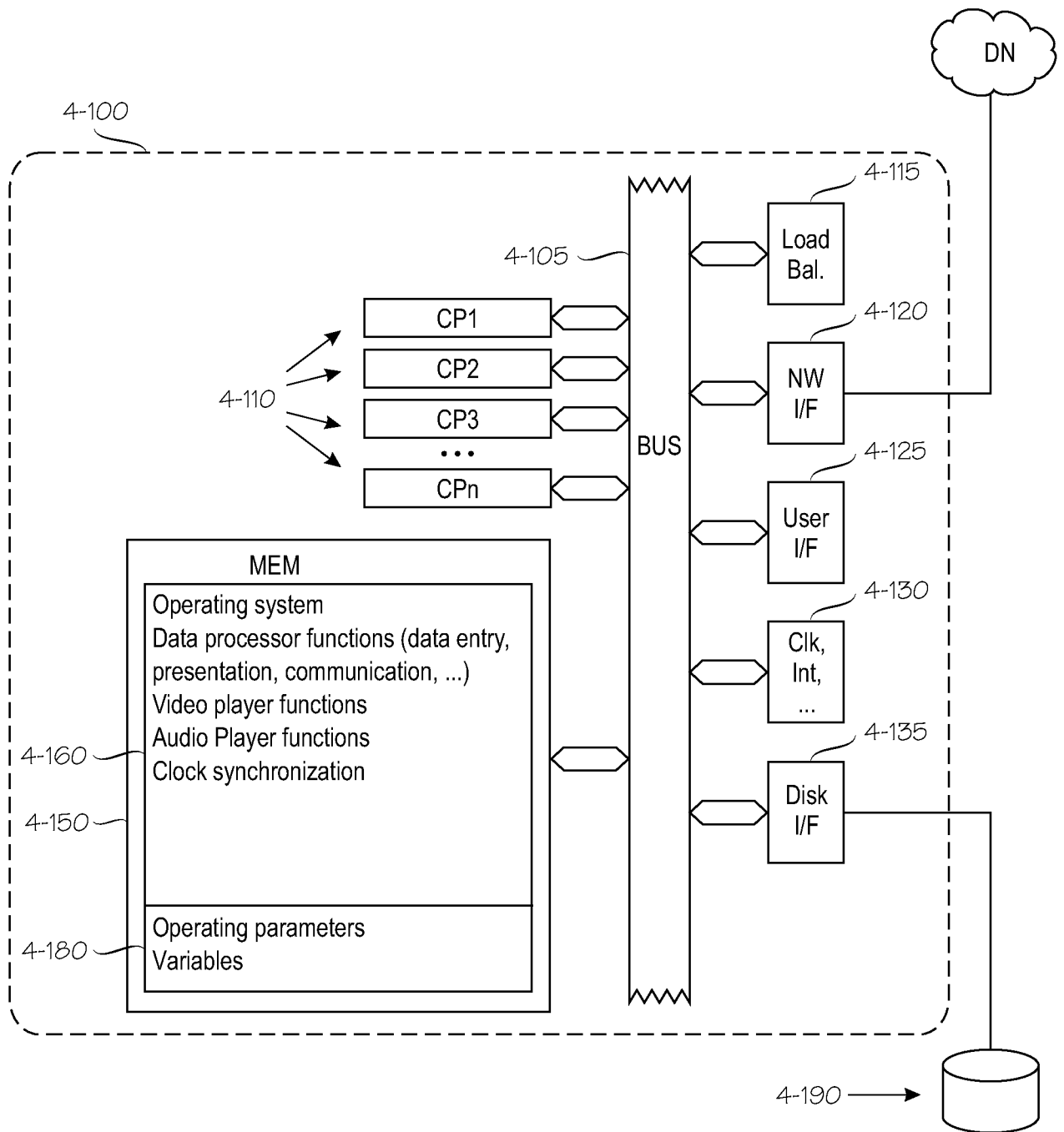
Fig. 2



2/3

Fig. 3





INTERNATIONAL SEARCH REPORT

International application No.

PCT/FI2014/050138

A. CLASSIFICATION OF SUBJECT MATTER

See extra sheet

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

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
FI, SE, NO, DK

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

EPO-Internal, WPI, INSPEC, COMPDX, XPIEE, XPI3E, XPESP, XPESP2, IEEE Xplore, Google Scholar

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	WO 2012049223 A2 (COMPASS INTERACTIVE LTD) 19 April 2012 (19.04.2012) page 3, lines 21-31; page 4, lines 12-17; page 5, lines 30-33; page 6, lines 1-2; page 7, lines 22-27; page 9, lines 22-24; page 11, lines 30-33; page 12, lines 1-3; page 13, lines 24-28; page 18, lines 31-33; page 19, lines 1-12	1-9
Y	US 2008219637 A1 (SANDREW, B. B.) 11 September 2008 (11.09.2008) paragraphs [0026], [0028]-[0030]	1-9
A	US 2012039582 A1 (FOO, T. W. et al.) 16 February 2012 (16.02.2012) the whole document	1-9
A	US 2012307149 A1 (MEEK, D. R. et al.) 06 December 2012 (06.12.2012) the whole document	1-9
A	US 6122668 A (TENG, P. Y. et al.) 19 September 2000 (19.09.2000) the whole document	1-9

☐ 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

21 May 2014 (21.05.2014)

Date of mailing of the international search report

23 May 2014 (23.05.2014)

Name and mailing address of the ISA/FI
Finnish Patent and Registration Office
P.O. Box 1160, FI-00101 HELSINKI, Finland

Facsimile No. +358 9 6939 5328

Authorized officer

Ari Viholainen

Telephone No. +358 9 6939 500

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/FI2014/050138

Patent document cited in search report	Publication date	Patent family members(s)	Publication date
WO 2012049223 A2	19/04/2012	CA 2814573 A1 CN 103460128 A EP 2628047 A2 GB 201017174 D0 US 2013272672 A1	19/04/2012 18/12/2013 21/08/2013 24/11/2010 17/10/2013
US 2008219637 A1	11/09/2008	US 8179475 B2 US 2008219641 A1	15/05/2012 11/09/2008
US 2012039582 A1	16/02/2012	CN 102405639 A EP 2422514 A1 WO 2010122447 A1 JP 2012524446 A RU 2011147112 A	04/04/2012 29/02/2012 28/10/2010 11/10/2012 27/05/2013
US 2012307149 A1	06/12/2012	US 2008209482 A1 US 8272008 B2	28/08/2008 18/09/2012
US 6122668 A	19/09/2000	None	

CLASSIFICATION OF SUBJECT MATTER

Int. Cl.

H04N 21/8547 (2011.01)**H04N 7/56** (2006.01)**H04N 21/434** (2011.01)**H04N 21/242** (2011.01)**H04N 21/2368** (2011.01)