(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) **EP 3 609 189 B1**

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention
of the grant of the patent:
06.10.2021 Bulletin 2021/40

(51) Int Cl.:
H04N 21/43 (2011.01)   *H04N 21/488* (2011.01)
H04N 21/84 (2011.01)   *H04N 21/434* (2011.01)
H04N 21/4425 (2011.01)   *H04N 21/2187* (2011.01)
H04N 21/44 (2011.01)

(21) Application number: **18188576.5**

(22) Date of filing: **10.08.2018**

(54) **TESTING RENDERING OF SCREEN OBJECTS**

TESTEN DES RENDERING VON BILDSCHIRMOBJEKTEN

TESTS DU RENDU D'OBJETS D'ÉCRAN

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR

(43) Date of publication of application:
12.02.2020 Bulletin 2020/07

(73) Proprietor: **Nagravision SA**
1033 Cheseaux-sur-Lausanne (CH)

(72) Inventors:
• **GORE, Douglas**
Cwmbran, Gwent NP44 3UW (GB)
• **ZOU, Ping**
Cwmbran, Gwent NP44 3UW (GB)

(74) Representative: **Schwegman Lundberg Woessner Limited**
Hillington Park Innovation Centre
1 Ainslie Road
Glasgow G52 4RU (GB)

(56) References cited:
KR-A- 20170 047 547

• **Alberto Sabater: "Automatic Subtitle Synchronization - Machine Learnings", , 14 September 2017 (2017-09-14), XP055520600, Retrieved from the Internet: URL:https://machinelearnings.co/automatic-subtitle-synchronization-e188a9275617 [retrieved on 2018-10-31]**

Description

FIELD

[0001]   The present disclosure relates to methods and devices for testing rendering of one or more screen objects in a video stream

BACKGROUND

[0002]   Automated testing is part of software development life cycles, which serves to continually ensure that software solutions are maintained in a functionally correct state. However, in areas like digital television (DTV) related development, an existing issue is that testing the correctness of the software or an instruction set is very difficult when the output is visual. Solutions like binary comparison of a visual output against a reference image are very brittle, time consuming to implement, and not suited for dynamic contents, e.g. live video streams. Particularly, existing DTV playback visual quality checks, e.g. subtitle rendering, is being done manually, or by a simple fixed image comparison. The existing processes for checking the quality of subtitle rendering are inefficient, hard to repeat, error prone and not feasible to apply to dynamic contents that are rendered for playback. Furthermore, testing of visual output of a video stream of a live event is difficult to implement and almost impossible to guarantee accuracy when done using existing image comparison techniques such as OCR or manual testing methods. It is also very difficult to continually and consistently test a live visual output using known testing techniques. Accordingly, there is a need for an efficient, automated, scalable and accurate process for testing the rendering, i.e. output playback of dynamic content, i.e. a video stream comprising a plurality of video frames. KR20170047547 discloses a method of automatically adjusting the synchronization of subtitles in video. An article by A. Sabater entitled "Automatic Subtitle Synchronization - Machine Learnings (14 Sept 2017, https://machinelearnings.co/automatic-subtitle-synchronization-e188a9275617) discloses the use of a trained neural network to detect speech in videos and the use of that to synchronize the video subtitles.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003]   Aspects and embodiments of this disclosure are now described by way of example for the purpose of illustration and with reference to the accompanying drawings, in which:

Figure 1 is a schematic depicting a method of obtaining training data to train a data model to test rendering of one or more video frames, according to a first aspect of the disclosure.

Figure 1a is an example of a data model for use with the aspects and embodiments of the present disclosure.

Figure 2 is a schematic depicting a second aspect of testing video data that is rendered, where the testing is automatically carried out using a trained data model.

Figure 3 shows an example of original or source metadata of video frames prior to rendering, and an example of extracted metadata corresponding to the video frames after rendering, relating to the second aspect.

Figure 4 is a schematic depicting a method of providing a feedback loop to a media device relating to the outcome of the testing of figure 3.

Figure 5 is an example implementation of a media device for use in accordance with aspects of the present disclosure.

DETAILED DESCRIPTION

[0004]   In overview, methods and devices in accordance with the disclosure relate to obtaining metadata from an output video stream, which is then used to test if the output video stream was correctly rendered. Data models are trained to generate metadata from a video/visual output, i.e. a render of a video data of a media device or application, such as a set top box, to capture more information from the visual output for testing, diagnostics, as well as data for an automated quality check. The term rendering is understood to include playback at or using a media device. In some implementations, the media device may include an application or processing module for a player capable of preparing and providing a video output on a display, such as a visual display unit, graphical interface unit associated with the media device. The rendering also includes processing the video stream at media device for projection on a screen that is not directly associated with the media device, i.e. when the media device is a projector.

[0005]   In a first aspect a method of obtaining training data to train a data model is disclosed. In some implementations, the data model is trained for testing a rendering of one or more video frames at or using a media device. The method of the first aspect comprises obtaining as training data for a data model a set of rendered video frames, the method comprising rendering at a media device, a plurality of video frames of a source video stream, each frame comprising one or more primary screen objects. The primary screen objects include dynamic video content comprising images and optical characters that make up a visual scene represented in a video frame. The source video stream further comprises source or original metadata for describing a further screen object to be rendered in at least one of the plurality of video

frames, where the further screen object is configured to be superimposed on the one or more primary screen objects of a given frame during rendering. The method comprises rendering each frame of the plurality of frames at least once with the further screen object and at least once without the further screen object. The media device includes a setting allowing or not allowing the further screen object to be rendered. The primary screen objects are components that make up a dynamic image in a video frame, for example in-video images of individuals or buildings, vehicles etc. that make up a scene represented in the video frame. Further screen objects in a video frame include a timed text screen object, which for example represents subtitles, otherwise known as closed captions, for the video frame. Thus, the plurality of video frames is rendered at least once with the subtitles turned on, and at least once with the subtitled turned off. Rendering relates to preparing the video frames for video/visual output at or using a media player. This includes preparing the video frames for playback and/or display on a screen or graphical user interface associated with the media device. The method further includes creating a training data set including a set of rendered frames, each frame in the training data set corresponding to one of the plurality of video frames rendered with and without the further screen object, and a corresponding data item for each frame indicating the presence or absence of a further screen object in a given frame

[0006] The source metadata includes information to indicate whether a given frame includes a further screen object. In some implementations this may be a time stamp associated with the given video frame.

[0007] The data item for a given frame is a descriptive label. The label states if there is a further screen object, i.e. subtitle present or not. In some implementations the label may also state the location of the subtitle, dimensions, the text is etc. In other implementations, the data item may be a flag to state if a subtitle is present or not.

[0008] Thus, the training data to be provided to a data model is a set of frames or images, along with a corresponding data item or a flag or label for each.

[0009] In some embodiments, the data item obtained may be descriptive data that is provided or applied by an observer, for example by crowd sourcing or other means of data gathering, or provided by an existing database of associated labels, or any other known means. Crowd sourcing for the data items may typically be done by humans analysing the rendered frames and describing the contents of the frame. In some implementations many different kinds of video frames from video streams are analysed so that a number of various kinds of videos are provided.

[0010] Therefore, the training data is generated by providing videos with and without source metadata for further screen objects such as subtitles and rendering each video. For each video frame, a descriptive label applied by an observer is also provided.

[0011] The above described method proposes two ren-

dered video outputs of video frames that include the same primary screen objects. The difference in the renderings being that a further screen object is overlaid on the primary screen objects of each video frame when rendered with a further screen object setting turned on. The rendered outputs are read and analysed by an artificial neural network to distinguish further screen objects from primary screen objects that may have one or more features that are the same or appear similar to the further screen object. Primary screen objects in a video frame include in-video text labels, e.g. a shop brands or text printed on a moving vehicle in the video background, and the further screen object are timed text.

[0012] As the in-video content is the same in both rendered visual outputs, the method of the first aspect can advantageously distinguish text in the in-video labels from timed text to facilitate an automatic and accurate detection of a timed text object in a rendered visual output.

[0013] In some embodiments, the original metadata for the plurality video frames of the input video stream comprises one or more properties associated with each primary and/or further screen object in each video frame. For example, the metadata may include a position of the screen object in a given video frame, or a time-stamp associated with the instance that the screen object is rendered, or dimensions such as the height and width of the screen object, or a duration of time for which the screen object is rendered etc. This advantageously ensures that the machine/computer readable metadata includes enough information defining how and when each screen object is rendered, thereby making detection of and/or validation of a render of the further screen object easier and accurate in a data model or processor.

[0014] A media device for testing the rendering of one or more video frames is disclosed. The media device may be a set top box or the like comprising a receiver for obtaining a video stream that comprises a plurality of video frames, a player comprising one or more processors for preparing and/or rendering the obtained video stream and a display communicatively coupled with the player for displaying the rendered video stream as a visual output. The media device is configured to implement the method of the first aspect set out above.

[0015] In a second aspect of the present disclosure, a method of training a data model for detecting screen objects in one or more video frames rendered at or using a media device is disclosed. In an example implementation, the data model comprises a data structure that may be provided in a memory area of a computing device. The data structure may in some implementations be based on an artificial neural network configured to receive, analyse, characterise and store data such as video frames i.e. pixels and images contents, or metadata associated with a plurality of screen objects from a plurality of video frames. The method comprises initialising a data model for obtaining and storing data associated with a plurality of video frames and obtaining as a first input, a

plurality of rendered video frames. The rendered frames of the first input may be video frames rendered at the media player of the first aspect. As a second input, data items corresponding to each of the rendered frames of the first input is provided. In some implementations, the data items may be descriptive labels, each for a given rendered frame. For example, as with the first aspect the data items may be obtained by crowd sourcing by human observers or from a database. The corresponding data items for each frame is indicative of the presence or absence of a further screen object in the corresponding frame. The second aspect of training the data model is provided by applying each rendered frame to an input of the data model and reducing an error between an output of the data model for a given frame and the corresponding data item for the given frame.

[0016]   Thus, once we have the training data set, for instance, as set out in the first aspect, including a set of rendered frames, and corresponding data items for each frame labelled with descriptive labels, this set ca be used to train an ANN, such as CNN like Caffe, which is a known existing image based neural network.

[0017]   In some implementations, the method of training the data model is by applying each frame to an input layer of the CNN. In the simplest case, it may be considered that for example the CNN has a neuron for each pixel, for generating the output. The output is the extracted metadata from the rendered frames in the first input generated by the data model. In some implementations, this extracted metadata will include details on the further screen object rendered. Next, the method of training the data model includes adjusting connection weights of the neurons within the data model to minimise the error or difference between the recorded output metadata and the data items, i.e. descriptive metadata in the labels in the second input.

[0018]   As noted above, in the simplest case this can be an indication if a further screen object such as a subtitle is present or not. The output of the data model may a flag or setting or metadata that indicates the presence or absence of a further screen object. This output is referred to as extracted metadata, as it related to computer/machine readable data relating to one or more screen objects that are rendered in a video output. In some embodiments, the exacted metadata maybe in a binary output form, i.e. a flag or setting, where "1" or similar indicates the presence of a further screen object such as a subtitle, and a "0" or similar indicates the absence of a further screen object.

[0019]   The data model is an artificial neural network (ANN), such as a convolutional neural network (CNN), that receives and analyses images i.e. the screen objects on in a video frame, and outputs a flag or a result indicating the presence of a further screen object such as a subtitle. This is extracted metadata explaining what the further screen object is, i.e. a subtitle superimposed on other primary screen objects, or an extension in a frame on which there is zero primary screen object information,

etc. The extracted metadata is therefore the output of the data model.

[0020]   In some implementations, a classification and or clustering algorithm may be used to train the data model for identifying one or more specific characteristics of the further screen objects in the video frames of the first input.

[0021]   Advantageously, training the data model with a training data set as set out in the first aspect, that includes properties of the primary screen objects in the input video frame with and without the further screen objects of the same video frames before rendering enables the data model to accurately detect such properties subsequent or other video streams when generating the extracted metadata for such subsequent or other streams.

[0022]   The method of training the data model comprises detecting the presence of one or more specific or defined characteristics in the video frames of the first input. Advantageously, this enables accurate identification of rendered screen objects that include the one or more specific characteristics to better differentiate from rendered screen objects that do not include them. In some implementations, the method comprises providing a binary output identifier to indicate the presence or absence of a further screen object in the rendered video frame, based on the outcome of detecting one or more specific characteristics.

[0023]   In a third aspect of the present disclosure, a method of testing video data being rendered at or using a media device is disclosed. The method of testing according to the third aspect comprises receiving at a media device a plurality of video frames to be rendered, each frame comprising one or more primary screen objects and at least one further screen object. The method of testing comprises rendering the received frames at or using the media device wherein the at least one further screen object is superimposed on the one or more primary screen objects of a given frame during rendering. The plurality of frames is rendered in accordance with source or original metadata, including machine readable instructions for a media player or the like to render primary and/or further screen objects from a source of the plurality of video frames.

[0024]   The rendered frames are then provided to a data model, said data model being trained to identify a further screen object rendered in the video frame. Metadata is then extracted from the rendered video frames by the data model. As mentioned in relation to the second aspect, the extracted metadata comprises one or more properties associated with at least a further screen object rendered for a given video frame. The data model is one that is trained in accordance with the second aspect set out above. The data model is trained with training data as set out above in relation to the first aspect.

[0025]   The original metadata is provided to the data model from a source of the video stream that comprises the video frames provided to the media device to be rendered. The original metadata that is provided includes

metadata relating to only the further screen object for each video frame, and the original metadata relate to the primary screen objects and the further screen objects. The method of testing according to the third aspect further comprises detecting the presence or absence of a further screen object by the trained data model, wherein the output of the data model is extracted metadata indicating such presence or absence for each frame. The method of testing further includes validating a detected further screen object using the original or source metadata relating to at least the further screen object before rendering. In some implementations, the validation is carried out if the source metadata indicates that a subtitle should be generated for the rendered frame and the output or extracted metadata from the data model indicates that a subtitle has been rendered, or if the source metadata indicates that a subtitle should not be generated for the rendered frame and the output or extracted indicates that a subtitle has not been rendered

[0026]   Advantageously, the method according to the third aspect enables the use of computer readable data, such as metadata from an output video stream, to test if the output video stream was correctly rendered. This enables testing of an output rendering to be done automatically, rather than by manually checking if the dynamic media components for each frame are correctly rendered. As metadata is generated by a data model to identify several types of screen objects and detect the presence one or more further screen object, this process is more accurate and effective when compared to a manual or a fixed image comparison process. As the testing process is automatic, i.e. carried out by computing devices implementing the data model that can characterise, classify and/or analyse the further screen objects in a rendered video frame, the process of testing a render can be repeated as often as needed for testing dynamic contents of a video stream that is rendered for playback. Furthermore, the method can be used to accurately test a live stream of video data in real time from metadata extracted from the render of the live video frames.

[0027]   In some embodiments, the method of testing comprises identifying and comparing one or more specific characteristics of a detected further screen object from a rendered video frame to one or more corresponding or related specific characteristics of the original metadata pertaining to the detected further screen object. This is for identifying the presence or absence of a variance in each specific characteristic for the detected further screen object.

[0028]   Advantageously, the automatic detecting of a variance in specific characteristics enables an identification of whether the result of the automatic testing is positive or negative. The presence of a variance indicates that further screen object has not been rendered correctly. In some implementations, the further screen object is a timed text instruction and the specific characteristic is a time stamp relative to one or more primary screen objects in a given frame. Therefore, a variance indicates

that the timed text is out of sync for the given video frame.

[0029]   In some implementations an offset is calculated based on a detected variance. The offset advantageously indicates the extent to which the further screen object is incorrectly rendered. In some embodiments, the offset is provided as feedback to the media device for subsequent adjustments. This advantageously enables the media device to adjust a subsequent rendering of the plurality of video frames based on the offset, so that the subsequent rendering no longer includes the variance. This enables application of a correction factor to automatically ensure that the same error(s) do not reoccur when rendering the video stream.

[0030]   Some specific components and embodiments are now described by way of illustration with reference to the accompanying drawings, in which like reference numerals refer to like features.

[0031]   With reference to Figure 1, a schematic implementation of a method of obtaining training data for training a data model is shown, according to a first aspect.

[0032]   A video stream 102 comprising a plurality of video frames is shown, along with source or original metadata 104 associated the video stream 102. In some implementations of the first aspect, the video stream is considered to be a test video stream. The original metadata is a machine-readable representation of one or more dynamic components of the video frames in the video stream 102. The original metadata 104 provided at a source of the video stream 102 is for specifying properties dictating the output or render of the video stream 102. In some implementations, this original metadata 104 may at least include metadata for generating a subtitle for one or more video frames in the video stream 102. The video stream 102 in Figure 1 may be considered as a test video stream, rendered samples of which are to be provided for training a data model 114.

[0033]   Each video frame in the video stream 102 comprises a plurality of primary screen objects, i.e. in-video images and dynamic optical characters. A plurality of such video frames in the video stream 102 also have a further screen object that is to be superimposed on the primary screen objects, during rendering. The further screen object is timed text, i.e. a subtitle, for a given frame.

[0034]   In step 106, the video stream 102 is provided to a media device 108 in two input stages. In the first input stage, the media device 108 is provided with the video stream 102 without any machine-readable instructions for defining the subtitles. In a second input stage the media device is provided with the same video stream 102, but this time along with machine-readable instructions that defines the subtitles to be rendered or generated for each video frame. The machine-readable instructions provided to the media device 108 are considered to be source or original metadata 104. The original metadata is obtained from a source of the media stream 102 to the media device 108. The media device 108 is configured to render the video stream 102 with the sub-

titles added for each frame received in the second input stage separately, or after rendering the same video stream 102 without the subtitles of the first input stage.

[0035] In step 110a the media 108 renders video stream 102 without any subtitle instructions, i.e. no subtitle instructions are present in the original metadata 104, to create a first sample 112a comprising a set of rendered video frames. This can be considered in implementation as a playback of the video stream with subtitles turned off. Therefore, no subtitle instructions are processed for this render and the video frames with in-video dynamic content, i.e. primary screen objects, are rendered to create the first sample of video frames 112a.

[0036] In step 110b the media device 108 renders video stream 102 with the subtitle instructions to create a second sample 112b comprising a set of rendered video frames with their associated subtitles. For example, in implementation the video stream 102 can be considered as being rendered by a media device with the subtitle turned on to create a second sample 112b. In an example implementation, one video frame in the second sample is rendered per second by the media device 108.

[0037] Training data to be provided to a train a data model includes the first sample 112a and the second sample 112b. In other words, the training data set includes the test video frames rendered once with the further screen object and once without. In addition to the pairs of rendered frames, the training data set also includes a data item associated with each of the rendered test video frames. The data item includes descriptive labels for at least the further screen objects of the video frames. The descriptive label may include information that is somewhat similar to the source metadata, i.e. it provides information on how, where and when a certain screen object is rendered for the video frame. As mentioned above, the descriptive labels may be obtained from crowdsourcing, where one or more human observers assign labels to describe the screen object of a certain video frame. Accordingly, the training data set includes video frames rendered with and without a further screen object, as well as a data item as explained above for each frame. This advantageously enables, configures or trains the data model 114 to detect the presence of one or more given screen objects in a render of any video stream based on the properties of the rendered frame and/or validate if the render of the detected screen object has been performed correctly.

[0038] The data model 114 is an artificial neural network (ANN), such as a convolutional neural network (CNN). For example, Caffe™ is example of a CNN computational model may be used for detecting objects from a dynamic visual output or image.

[0039] ANNs, otherwise known as connectionist systems, are computing systems vaguely inspired by biological neural networks. It is known that such networks "learn" tasks by considering data samples, generally without task-specific pre-programming and without any prior knowledge about a task, and instead, evolve their own set of relevant characteristics from learning material or samples that they process. ANNs can be hardware-based (where neurons are represented by physical components) or software-based (computer models) and can use a variety of topologies, learning algorithms or functions.

[0040] To initialise such a data model 114, which is an ANN, to be able to detect one or more screen objects from a rendered output- one or more functions may be defined for the data model 114. For example, the defined functions can be one or more of: a function that returns a structure describing labels for list of images, and/or a function that returns a region of interest (ROI) structure that describes bounding box annotations, for subtitles for instance; and/or a function that provides a test evaluation, i.e. comparison function. The text evaluation function in some embodiments can be a binary function, where a 0 indicates no subtitles (which will be the case for the first sample 112a) and 1 indicates that a subtitle exists (which will be the case for second sample 112b).

[0041] An example implementation of an ANN in its simplest form that can be used as the data model 114 for aspects and embodiments of the present invention is seen in Figure 1a. ANNs usually have at least three layers that are interconnected. The first layer, which is an input layer consists of input neurons. Those neurons are configured to send data on to the second layer, referred to as a hidden layer which implements one or more functions on data from the input layer, and which in turn is configured to send the output neurons to the third, or output layer. There may be a plurality of hidden layers in the ANN, but for simplicity a model with one such layer is shown.

[0042] The second or hidden layer in a neural network implements one or more functions. For example, the function or functions may each compute a linear transformation of the output from a previous layer or compute logical functions. For instance, considering an implementation where an input vector can be represented as x, the second or hidden layer functions as h, and the third or output layer as y, then the ANN may be understood as implementing a function $f(x)$ using the first layer and the second layer that maps from x to h, and another function $g(x)$ using the second layer and the third layer that maps from h to y.

[0043] Thus, the hidden layer is activated by $f(x)$ and the output of the network is $g(f(x))$. In some implementations of the present described embodiments, it may be understood that $f(x)$ can represent subtitle or further object detection, and $g(x)$ can represent validation of the render of the detected subtitle or further object at $f(x)$, for instance. Subtitle detection and validation is further explained below in relation to Figure 2 below.

[0044] The data model 114 detects the presence of absence of a subtitle and provides extracted metadata indicating this in an output, so that a further validation can take place to establish if the model produces the same output as the source metadata for each frame. For

instance, the data model output can be represented as

Extracted Metadata => subtitle present? (Yes/no = 0/1)

[0045] The training data of the first aspect, for example, is applied to the input and the weights of the neurons are adjusted so that the output 0 or 1 (or <0.5 or >0.5,) depending on whether a subtitle was in the training frame or not.

[0046] Finally, further to validation, the data model may be represented as: {extracted metadata; source metadata} => validated (yes or no).

[0047] With reference to Figure 2, a method of automatically testing a rendering of a video stream 202 is described, in accordance with a third aspect of the present disclosure. For the purposes of illustration, the video stream 202 is a live or real-time stream of an event such as a game or concert taking place, or a news channel broadcast live from a television studio. Like the video stream 102 in Figure 1, The live video stream 202 comprises a plurality of video frames, each frame having a plurality of primary screen objects and at least one further screen object, which is timed text, i.e. a subtitle. Original or source metadata 204 for at least the subtitle is provided at the source of the video stream 202. The original metadata 204 comprises, for example, a subtitle render instruction that includes properties defining how and when the subtitle is supposed to be displayed for a given video frame in the video stream 202, when the video stream 202 is rendered at a media device. An example of the original metadata 204 for a subtitle is seen for example in Figure 3 and will be explained in further detailed below.

[0048] In step 206 the live video stream 202 is provided to a media device 208. The media device 208 is configured to render the input video frames from the video stream 202 for playback and display with in-video primary screen objects, along with the superimposed or overlaid further screen object, i.e. the subtitle, that is rendered using the instructions in the original metadata 204. Upon rendering in step 210, rendered video frames 212 are provided as a visual output. The media device 208 may be arranged in step 210 to render the video frames 212 by sampling one video frame of the input live video stream 202 per second.

[0049] In step 214, the rendered frames are provided to a further screen object detection module Given that the further screen object is a subtitle in the illustrated embodiments, Figure 2 shows a subtitle detection unit 216. The subtitle detection unit 216 is included in or associated with a data model 114 as described above in the second aspect, that has been trained based on training data illustrated and explained in relation to Figure 1, for accurately detecting and testing subtitles in a rendered video frame by generating metadata, i.e. this is extracted metadata obtained from a render of a visual output. Thus, in some implementations, as with the data model 114 in the second aspect, a binary output of '1' is produced to indicate if a subtitle is detected. Otherwise,

a binary output of '0' is produced.

[0050] In some examples, when rendered video frame 212 is provided to the data model 114, which includes or is associated with the subtitle detection unit 216, the output in some embodiments may include the following, for each video frame, referred to as extracted metadata from the data model 114.

Timestamp ...
Is subtitle input provided (Y/N or 0/1)?
(and optional dimensions such as)
subtitle-top: ...
subtitle-left: ...
subtitle-width: ...
subtitle-height: ...

[0051] The data model 114 detects the presence of a subtitle in the rendered frame 212. This is possible as the data model 114 has been trained using the training data set as described in relation to the first aspect including a pairs of video frames rendered with and without a further screen object, and a data item with a descriptive label for at least the further screen objects (see the first aspect in figure 1).

[0052] In step 218, if a subtitle is detected in a rendered video frame by the subtitle detection unit 216, then the extracted metadata output from the subtitle detection unit 216 of a detected subtitle 220, or in some implementations the extracted metadata of the video frame including the detected subtitle 220, is provided to a subtitle validation unit 224. The subtitle validation unit 224 may in some embodiments be implemented using or associated with the trained data module 114 mentioned earlier in relation to Figure 1.

[0053] In step 222, original subtitle metadata 204 in the live video stream 202 corresponding to the rendered video frame(s) with detected subtitle metadata 220 from the subtitle detection unit 216 is then provided as a further input to the subtitle validation unit 224.

[0054] Thus, the subtitle validation unit 224 is provided with extracted metadata 220 of the detected subtitle, as well as the corresponding original metadata 204, i.e. source subtitle instruction for the same input subtitle prior to rendering. Based on a comparative analysis of both source 204 and detected metadata 220, the subtitle validation unit 224 is configured to provide an output result 226 to indicate whether the detected subtitle has been rendered correctly by the media device 208

[0055] An example implementation indicating the original metadata 204 and the extracted metadata for the detected subtitle 220 is shown in Figure 3, to illustrate how the data model 114 implementing the subtitle validating unit 224 can be used to test if one or more subtitles were correctly rendered. The original subtitle metadata 204 in Figure 3 indicates an instruction to render a subtitle ('First subtitle') beginning at the 2nd second and finishing at the 5th second. Also, the metadata 204 includes an instruction to render another one ('Second subtitle') be-

ginning at the 10th second and finishing at the 15th second.

[0056] The extracted subtitle metadata for the detected subtitle 220 or the video frame including the detected subtitle 220 indicates that a subtitle is rendered between the 2nd second and the 5th second. So, for the First subtitle in this illustration, the validation result will be a pass, i.e. output result 226 will indicate that the subtitle was rendered correctly and as defined by the original metadata 204.

[0057] The extracted subtitle metadata for the detected subtitle 220 also indicates that another subtitle is rendered between the 13th second and the 17th second during the render of the video stream 202. In this case the validation results 226 will be a fail in that, the Second subtitle starts to be rendered 2 seconds later than instructed in the original metadata 204, and the Second subtitle lasts 2 seconds longer than the instruction in the original metadata 204. The validation result 226 thus also in some embodiments, indicates that the detected rendered subtitle 220 is 2 seconds out of sync during rendering by the media device 208.

[0058] Figure 4 illustrates an example embodiment for adjusting a rendering using the result of the validation unit 226 obtained in Figure 2. Figure 4 includes all the steps and computations explained above for Figure 2. In addition, in step 228 of Figure 4, an offset to indicate a variance of an incorrect or a failed subtitle test is obtained. This can be the 2 second out of sync result that is seen in Figure 3, for example. The offset is provided as a feedback to the media device 208. One or more processors or computing devices in or associated with the media device 208 may be configured to apply this as a correction factor. The correction factor may in some implementations be provided to ensure that a future rendering or playback of the detected subtitle 220 or subtitles subsequently rendered in the video stream is no longer out of sync by 2 seconds.

[0059] Accordingly, from Figures 1 to 4, techniques for automatically testing rendering of visual outputs based on metadata extracted from a render of a live or real time stream of video data have been explained and illustrated. Advantageously the illustrated embodiments in Figures 1 to 4 enable automatic testing of visual output of a video stream of a live event and provides an efficient, automated, scalable and accurate process for testing the rendering of dynamic content.

[0060] Figure 5 illustrates a block diagram of one implementation of a computing device 500 within which a set of instructions, for causing the computing device to perform any one or more of the methodologies discussed herein, may be executed. The computing device 500 may be the media device or a device implementing the data model in the present disclosure. Similarly, the computing device 500 may also related to an example implementation of the source of the video stream. In alternative implementations, the computing device 500 may be connected (e.g., networked) to other machines in a Local Area Network (LAN), an intranet, an extranet, or the Internet. The computing device may operate in the capacity of a server or a client machine in a client-server network environment, or as a peer machine in a peer-to-peer (or distributed) network environment. The computing device may be a personal computer (PC), a tablet computer, a set-top box (STB), a Personal Digital Assistant (PDA), a cellular telephone, a web appliance, a server, a network router, switch or bridge, or any machine capable of executing a set of instructions (sequential or otherwise) that specify actions to be taken by that machine. Further, while only a single computing device is illustrated, the term "computing device" shall also be taken to include any collection of machines (e.g., computers) that individually or jointly execute a set (or multiple sets) of instructions to perform any one or more of the methodologies discussed herein.

[0061] The example computing device 500 includes a processing device 502, a main memory 504 (e.g., read-only memory (ROM), flash memory, dynamic random-access memory (DRAM) such as synchronous DRAM (SDRAM) or Rambus DRAM (RDRAM), etc.), a static memory 506 (e.g., flash memory, static random-access memory (SRAM), etc.), and a secondary memory (e.g., a data storage device 518), which communicate with each other via a bus 530.

[0062] Processing device 502 represents one or more general-purpose processors such as a microprocessor, central processing unit, or the like. More particularly, the processing device 502 may be a complex instruction set computing (CISC) microprocessor, reduced instruction set computing (RISC) microprocessor, very long instruction word (VLIW) microprocessor, processor implementing other instruction sets, or processors implementing a combination of instruction sets. Processing device 502 may also be one or more special-purpose processing devices such as an application specific integrated circuit (ASIC), a field programmable gate array (FPGA), a digital signal processor (DSP), network processor, or the like. Processing device 502 is configured to execute the processing logic (instructions 522) for performing the operations and steps discussed herein.

[0063] The computing device 500 may further include a network interface device 508. The computing device 500 also may include a video display unit 510 (e.g., a liquid crystal display (LCD) or a cathode ray tube (CRT)), an alphanumeric input device 512 (e.g., a keyboard or touchscreen), a cursor control device 514 (e.g., a mouse or touchscreen), and an audio device 516 (e.g., a speaker).

[0064] The data storage device 518 may include one or more machine-readable storage media (or more specifically one or more non-transitory computer-readable storage media) 528 on which is stored one or more sets of instructions 522 embodying any one or more of the methodologies or functions described herein. The instructions 522 may also reside, completely or at least partially, within the main memory 504 and/or within the

processing device 502 during execution thereof by the computer system 500, the main memory 504 and the processing device 502 also constituting computer-readable storage media.

[0065] The various methods described above may be implemented by a computer program. The computer program may include computer code arranged to instruct a computer to perform the functions of one or more of the various methods described above. The computer program and/or the code for performing such methods may be provided to an apparatus, such as a computer, on one or more computer readable media or, more generally, a computer program product. The computer readable media may be transitory or non-transitory. The one or more computer readable media could be, for example, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, or a propagation medium for data transmission, for example for downloading the code over the Internet. Alternatively, the one or more computer readable media could take the form of one or more physical computer readable media such as semiconductor or solid-state memory, magnetic tape, a removable computer diskette, a random-access memory (RAM), a read-only memory (ROM), a rigid magnetic disc, and an optical disk, such as a CD-ROM, CD-R/W or DVD.

[0066] In an implementation, the modules, components and other features described herein can be implemented as discrete components or integrated in the functionality of hardware components such as ASICS, FPGAs, DSPs or similar devices.

[0067] A "hardware component" is a tangible (e.g., non-transitory) physical component (e.g., a set of one or more processors) capable of performing certain operations and may be configured or arranged in a certain physical manner. A hardware component may include dedicated circuitry or logic that is permanently configured to perform certain operations. A hardware component may be or include a special-purpose processor, such as a field programmable gate array (FPGA) or an ASIC. A hardware component may also include programmable logic or circuitry that is temporarily configured by software to perform certain operations.

[0068] Accordingly, the phrase "hardware component" should be understood to encompass a tangible entity that may be physically constructed, permanently configured (e.g., hardwired), or temporarily configured (e.g., programmed) to operate in a certain manner or to perform certain operations described herein.

[0069] In addition, the modules and components can be implemented as firmware or functional circuitry within hardware devices. Further, the modules and components can be implemented in any combination of hardware devices and software components, or only in software (e.g., code stored or otherwise embodied in a machine-readable medium or in a transmission medium).

[0070] Unless specifically stated otherwise, as apparent from the following discussion, it is appreciated that throughout the description, discussions utilizing terms such as "receiving", "determining", "obtaining", "sending," "implementing," , "detecting", "extracting", "establishing" , "providing" , "rendering" , " identifying" or the like, refer to the actions and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices.

[0071] It is to be understood that the above description is intended to be illustrative, and not restrictive. Many other implementations will be apparent to those of skill in the art upon reading and understanding the above description. Although the present disclosure has been described with reference to specific example implementations, it will be recognized that the disclosure is not limited to the implementations described but can be practiced with modification and alteration within the scope of the appended claims. Accordingly, the specification and drawings are to be regarded in an illustrative sense rather than a restrictive sense. The scope of the disclosure should, therefore, be determined with reference to the appended claims.

Claims

1. A method of creating training data for training an artificial neural network for detecting screen objects in one or more video frames rendered at or using a media device (108, 208), the method comprising:

   rendering at a media device (108, 208), a plurality of video frames of a source video stream (102, 202), each frame comprising one or more primary screen objects, the source video stream (102, 202) further comprising original metadata (104, 204) for describing a further screen object to be rendered in at least one of the plurality of video frames, wherein the further screen object is configured to be superimposed on the one or more primary screen objects of a given frame during rendering;
   rendering each frame of the plurality of frames at least once with the further screen object and at least once without the further screen object; and
   creating a training data set including a set of rendered video frames (112, 212), each frame in the training data set corresponding to one of the plurality of video frames rendered with and without the further screen object, and further including a corresponding data item for each frame indicating the presence or absence of a further screen object in each frame;
   wherein the one or more primary screen objects

comprise dynamic images and/or optical characters and the further screen object is a timed text screen object.

2. The method as claimed in claim 1, further including obtaining the corresponding data item for each frame, wherein each data item includes a descriptive label for agiven frame, wherein said obtaining includes analysing the given frame by an observer to identify the presence or absence of a further screen object in the frame.

3. The method as claimed in any one of the preceding claims, wherein the original metadata comprises one or more properties associated with each primary and/or further screen object in each video frame.

4. The method as claimed in claim 3, wherein the one or more properties comprise at least one of:

   a position of the screen object in a video frame;
   a time-stamp associated with the instance that the screen object is rendered;
   dimensions of the screen object when rendered relative to a display associated with the media device; and
   a duration of time for which the screen object is rendered.

5. A media device (108, 208) for obtaining training data for testing rendering of one or more video frames, the media device comprising:

   a receiver for obtaining a video stream (102, 202) that comprises a plurality of dynamic video frames;
   a player comprising one or more processors for rendering the obtained video stream (102, 202); and
   a display communicatively coupled with the player for displaying the rendered video stream as a visual output,
   wherein the media device (108, 208) is configured to implement the method steps of any one of claims 1 to 4.

6. A computer implemented method of training an artificial neural network for detecting screen objects in one or more video frames rendered at or using a media device, the method implemented using one or more processors that are associated with a memory, the method comprising:

   initialising an artificial neural network in the memory for obtaining and/or storing data associated with a plurality of video frames;
   obtaining as a first input training data including a plurality of rendered frames of a video stream

rendered at or using the media player;
obtaining as a second input training data including data items associated with each of the rendered frames of the first input, each data item being indicative of the presence or absence of a further screen object in a given frame;
obtaining as an output of the artificial neural network, extracted metadata indicating the presence or absence of a further screen object in a given frame;
applying one or more functions to reduce an error between the output extracted metadata of the artificial neural network and the corresponding data item for the given frame;
wherein the first and second inputs are provided by the method of obtaining training data according to any one of claims 1 to 4.

7. The method as claimed in claim 6, wherein training the artificial neural network further comprises:

   detecting the presence of the one or more specific characteristics in the rendered video frames; and
   providing a binary output identifier to indicate the presence or absence of a further screen object in the rendered video frames, based on the outcome of the detecting step.

8. A method of testing video data being rendered at or using a media device (108, 208), the method comprising:

   receiving at a media device (108, 208) a plurality of video frames to be rendered in accordance with original metadata, each frame comprising one or more primary screen objects and at least one further screen object;
   rendering the received frames at or using the media device (108, 208) wherein the at least one further screen object is superimposed on the one or more primary screen objects of a given frame during rendering, the received plurality of frames being rendered in accordance with original metadata;
   providing the rendered frames to an artificial neural network, said artificial neural network being trained in accordance with the method as claimed in claim 6 or claim 7;
   obtaining extracted metadata as an output of the artificial neural network, the extracted metadata detecting the presence or absence of a further screen object in a given rendered frame;
   providing the original metadata associated with the plurality of video frames to the artificial neural network; and
   validating the rendering of a detected further screen object based on the original metadata

relating to the given video frame.

9. The method as claimed in claim 8, wherein the step of validating further comprises:

    identifying a specific characteristic of the detected further screen object;
    comparing the identified characteristic to a related specific characteristic in the original metadata; and
    detecting the presence of a variance in the identified specific characteristic based on the outcome of the comparing step.

10. The method as claimed in 9, further comprising:

    responsive to a variance being detected, calculating an offset based on the detected variance;
    providing the offset as feedback to a processor associated with the media device (108, 208); and
    adjusting a subsequent rendering of the video frames based on the offset.

11. A system for testing video data being rendered at or using a media device (108, 208), the system including the media device (108, 208) and at least one processor associated with a memory for storing a plurality of instructions, wherein the at least one processor is configured to execute the plurality of instructions to implement the method as claimed in any one claims 8 to 10.

Patentansprüche

1. Verfahren zum Erzeugen von Trainingsdaten zum Trainieren eines künstlichen neuronalen Netzwerks, um Bildschirmobjekte in einem oder mehreren Videoframes, die auf oder unter Verwendung einer Medienvorrichtung (108, 208) gerendert werden, zu erkennen, wobei das Verfahren umfasst:

    Rendern einer Vielzahl von Videoframes eines Quellen-Videostreams (102, 202) auf einer Medienvorrichtung (108, 208), wobei jeder Frame ein oder mehrere primäre Bildschirmobjekte umfasst, wobei der Quellen-Videostream (102, 202) weiter ursprüngliche Metadaten (104, 204) umfasst, um ein weiteres Bildschirmobjekt, das in mindestens einem aus der Vielzahl von Videoframes gerendert werden soll, zu beschreiben, wobei das weitere Bildschirmobjekt so ausgelegt ist, dass es beim Rendern dem einen oder den mehreren primären Bildschirmobjekten eines gegebenen Frames überlagert wird;
    Rendern jedes Frames aus der Vielzahl von Frames mindestens einmal mit dem weiteren Bild-

schirmobjekt, und mindestens einmal ohne dem weiteren Bildschirmobjekt; und
Erzeugen eines Trainingsdatensatzes, der einen Satz gerenderter Videoframes (112, 212) einschließt, wobei jeder Frame in dem Trainingsdatensatz einem aus der Vielzahl von mit und ohne dem weiteren Bildschirmobjekt gerenderten Videoframes entspricht, und der weiter ein entsprechendes Datenelement für jeden Frame einschließt, das das Vorhandensein oder Fehlen eines weiteren Bildschirmobjekts in jedem Frame angibt;
wobei das eine oder die mehreren primären Bildschirmobjekte dynamische Bilder und/oder optische Zeichen umfassen, und das weitere Bildschirmobjekt ein Timed Text-Bildschirmobjekt ist.

2. Verfahren nach Anspruch 1, das weiter das Erhalten des entsprechenden Datenelements für jeden Frame einschließt, wobei jedes Datenelement eine beschreibende Bezeichnung für einen gegebenen Frame einschließt, wobei das Erhalten das Analysieren des gegebenen Frames durch einen Beobachter einschließt, um das Vorhandensein oder Fehlen eines weiteren Bildschirmobjekts im Frame zu identifizieren.

3. Verfahren nach einem der vorstehenden Ansprüche, wobei die ursprünglichen Metadaten eine oder mehrere Eigenschaften umfassen, die mit jedem primären und/oder weiteren Bildschirmobjekt in jedem Videoframe verknüpft sind.

4. Verfahren nach Anspruch 3, wobei die eine oder die mehreren Eigenschaften mindestens eines umfassen aus:

    einer Position des Bildschirmobjekts in einem Videoframe;
    einem Zeitstempel, der mit dem Zeitpunkt, zu dem das Bildschirmobjekt gerendert wird, verknüpft ist;
    Abmessungen des Bildschirmobjekts, wenn es gerendert wird, in Bezug auf eine Anzeige, die mit der Medienvorrichtung verknüpft ist; und
    einer Zeitdauer, während der das Bildschirmobjekt gerendert wird.

5. Medienvorrichtung (108, 208) zum Erhalten von Trainingsdaten, um das Rendern eines oder mehrer Videoframes zu testen, wobei die Medienvorrichtung umfasst:

    einen Empfänger, um einen Videostream (102, 202) zu erhalten, der eine Vielzahl von dynamischen Videoframes umfasst;
    ein Abspielgerät, das einen oder mehrere Pro-

zessoren umfasst, um den erhaltenen Video-stream (102, 202) zu rendern; und

eine Anzeige, die kommunikationsmäßig mit dem Abspielgerät gekoppelt ist, um den gerenderten Videostream als eine visuelle Ausgabe anzuzeigen,

wobei die Medienvorrichtung (108, 208) so ausgelegt ist, dass sie die Verfahrensschritte nach einem der Ansprüche 1 bis 4 implementiert.

6. Computerimplementiertes Verfahren zum Trainieren eines künstlichen neuronalen Netzwerks, um Bildschirmobjekte in einem oder mehreren Videoframes, die auf oder unter Verwendung einer Medienvorrichtung gerendert werden, zu erkennen, wobei das Verfahren unter Verwendung eines oder mehrerer Prozessoren, die mit einem Speicher verknüpft sind, implementiert wird, wobei das Verfahren umfasst:

Initialisieren eines künstlichen neuronalen Netzwerks im Speicher, um Daten, die mit einer Vielzahl von Videoframes verknüpft sind, zu erhalten und/oder zu speichern;

Erhalten von Trainingsdaten, die eine Vielzahl von gerenderten Frames eines Videostreams, der auf oder unter Verwendung des Medienabspielgeräts gerendert wurde, einschließen, als eine erste Eingabe;

Erhalten von Trainingsdaten, die Datenelemente, welche mit jedem der gerenderten Frames der ersten Eingabe verknüpft sind, einschließen, als eine zweite Eingabe, wobei jedes Datenelement das Vorhandensein oder Fehlen eines weiteren Bildschirmobjekts in einem gegebenen Frame angibt;

Erhalten von extrahierten Metadaten, die das Vorhandensein oder Fehlen eines weiteren Bildschirmobjekts in einem gegebenen Frame angeben, als eine Ausgabe des künstlichen neuronalen Netzwerks;

Anwenden einer oder mehrerer Funktionen, um einen Fehler zwischen den ausgegebenen extrahierten Metadaten des künstlichen neuronalen Netzwerks und dem entsprechenden Datenelement für den gegebenen Frame zu reduzieren;

wobei die erste und die zweite Eingabe über das Verfahren zum Erhalten von Trainingsdaten nach einem der Ansprüche 1 bis 4 bereitgestellt werden.

7. Verfahren nach Anspruch 6, wobei das Trainieren des künstlichen neuronalen Netzwerks weiter umfasst:

Erkennen des Vorhandenseins des einen oder der mehreren spezifischen Merkmale in den ge-

renderten Videoframes; und

Bereitstellen, auf Basis des Ergebnisses des Erkennungsschritts, eines binären Ausgabe-Identifikators, um das Vorhandensein oder Fehlen eines weiteren Bildschirmobjekts in den gerenderten Videoframes anzugeben.

8. Verfahren zum Testen von Videodaten, die auf oder unter Verwendung einer Medienvorrichtung (108, 208) gerendert werden, wobei das Verfahren umfasst:

Empfangen einer Vielzahl von Videoframes, die in Übereinstimmung mit ursprünglichen Metadaten gerendert werden sollen, an einer Medienvorrichtung (108, 208), wobei jeder Frame ein oder mehrere primäre Bildschirmobjekte und mindestens ein weiteres Bildschirmobjekt umfasst;

Rendern der empfangenen Frames auf oder unter Verwendung der Medienvorrichtung (108, 208), wobei das mindestens eine weitere Bildschirmobjekt beim Rendern dem einen oder den mehreren primären Bildschirmobjekten eines gegebenen Frames überlagert wird, wobei die empfangene Vielzahl von Frames in Übereinstimmung mit ursprünglichen Metadaten gerendert wird;

Bereitstellen der gerenderten Frames für ein künstliches neuronales Netzwerk, wobei das künstliche neuronale Netzwerk in Übereinstimmung mit dem Verfahren nach Anspruch 6 oder Anspruch 7 trainiert ist;

Erhalten von extrahierten Metadaten als eine Ausgabe des künstlichen neuronalen Netzwerks, wobei die extrahierten Metadaten das Vorhandensein oder Fehlen eines weiteren Bildschirmobjekts in einem gegebenen gerenderten Frame erkennen;

Bereitstellen der ursprünglichen Metadaten, die mit der Vielzahl von Videoframes verknüpft sind, für das künstliche neuronale Netzwerk; und

Validieren des Renderings eines erkannten weiteren Bildschirmobjekts auf Basis der ursprünglichen Metadaten, die sich auf den gegebenen Videoframe beziehen.

9. Verfahren nach Anspruch 8, wobei der Schritt des Validierens weiter umfasst:

Identifizieren eines spezifischen Merkmals des erkannten weiteren Bildschirmobjekts;

Vergleichen des identifizierten Merkmals mit einem zugehörigen spezifischen Merkmal in den ursprünglichen Metadaten; und

Erkennen des Vorhandenseins einer Abweichung des identifizierten spezifischen Merkmals auf Basis des Ergebnisses des Vergleichs-

schritts.

10. Verfahren nach 9, weiter umfassend:

in Reaktion darauf, dass eine Abweichung erkannt wird, Berechnen eines Versatzes auf Basis der erkannten Abweichung;
Bereitstellen des Versatzes als Rückmeldung für einen Prozessor, der mit der Medienvorrichtung (108, 208) verknüpft ist; und
Anpassen eines anschließenden Renderings der Videoframes auf Basis des Versatzes.

11. System zum Testen von Videodaten, die auf oder unter Verwendung einer Medienvorrichtung (108, 208) gerendert werden, wobei das System die Medienvorrichtung (108, 208) und mindestens einen Prozessor einschließt, der mit einem Speicher verknüpft ist, um eine Vielzahl von Anweisungen zu speichern, wobei der mindestens eine Prozessor so ausgelegt ist, dass er die Vielzahl von Anweisungen ausführt, um das Verfahren nach einem der Ansprüche 8 bis 10 zu implementieren.

Revendications

1. Procédé de création de données d'apprentissage pour apprendre à un réseau neuronal artificiel la détection d'objets d'écran dans une ou plusieurs trames vidéo rendues au niveau ou en utilisant un dispositif multimédia (108, 208), le procédé comprenant :

le rendu au niveau d'un dispositif multimédia (108, 208), d'une pluralité de trames vidéo d'un flux vidéo source (102, 202), chaque trame comprenant un ou plusieurs objets d'écran primaires, le flux vidéo source (102, 202) comprenant en outre des métadonnées d'origine (104, 204) pour décrire un objet d'écran supplémentaire à rendre dans au moins l'une de la pluralité de trames vidéo, dans lequel l'objet d'écran supplémentaire est configuré pour être superposé sur les un ou plusieurs objets d'écran primaires d'une trame donnée pendant le rendu ;
le rendu de chaque trame de la pluralité de trames au moins une fois avec l'objet d'écran supplémentaire et au moins une fois sans l'objet d'écran supplémentaire ; et
la création d'un ensemble de données d'apprentissage incluant un ensemble de trames vidéo rendues (112, 212), chaque trame dans l'ensemble de données d'apprentissage correspondant à l'une de la pluralité de trames vidéo rendues avec ou sans l'objet d'écran supplémentaire, et incluant en outre un élément de données correspondant pour chaque trame indiquant la présence ou l'absence d'un objet d'écran supplémentaire dans chaque trame ;
dans lequel les un ou plusieurs objets d'écran primaires comprennent des images dynamiques et/ou des caractères optiques et l'objet d'écran supplémentaire est un objet d'écran textuel temporisé.

2. Procédé selon la revendication 1, incluant en outre l'obtention de l'élément de données correspondant pour chaque trame, dans lequel chaque élément de données inclut une étiquette descriptive pour une trame donnée, dans lequel ladite obtention inclut l'analyse de la trame donnée par un observateur pour identifier la présence ou l'absence d'un objet d'écran supplémentaire dans la trame.

3. Procédé selon l'une quelconque des revendications précédentes, dans lequel les métadonnées d'origine comprennent une ou plusieurs propriétés associées à chaque objet d'écran primaire et/ou supplémentaire dans chaque trame vidéo.

4. Procédé selon la revendication 3, dans lequel les une ou plusieurs propriétés comprennent au moins l'un parmi :

une position de l'objet d'écran dans une trame vidéo ;
un horodatage associé au cas où l'objet d'écran est rendu ;
des dimensions de l'objet d'écran lorsqu'il est rendu par rapport à un affichage associé au dispositif multimédia ; et
une durée pendant laquelle l'objet d'écran est rendu.

5. Dispositif multimédia (108, 208) d'obtention de données d'apprentissage pour tester le rendu d'une ou plusieurs trames vidéo, le dispositif multimédia comprenant :

un récepteur pour obtenir un flux vidéo (102, 202) qui comprend une pluralité de trames vidéo dynamiques ;
un lecteur comprenant un ou plusieurs processeurs pour rendre le flux vidéo obtenu (102, 202) ; et
un affichage couplé en communication avec le lecteur pour afficher le flux vidéo rendu en tant que sortie visuelle,
dans lequel le dispositif multimédia (108, 208) est configuré pour mettre en œuvre les étapes du procédé selon l'une quelconque des revendications 1 à 4.

6. Procédé mis en œuvre par ordinateur pour apprendre à un réseau neuronal artificiel la détection d'ob-

jets d'écran dans une ou plusieurs trames vidéo rendues au niveau ou en utilisant un dispositif multimédia, le procédé mis en œuvre utilisant un ou plusieurs processeurs qui sont associés à une mémoire, le procédé comprenant :

> l'initialisation d'un réseau neuronal artificiel dans la mémoire pour obtenir et/ou stocker des données associées à une pluralité de trames vidéo ;
>
> l'obtention, en tant que première entrée, des données d'apprentissage incluant une pluralité de trames rendues d'un flux vidéo rendu au niveau ou en utilisant le dispositif multimédia ;
>
> l'obtention, en tant que seconde entrée, des données d'apprentissage incluant des éléments de données associés à chacune des trames rendues de la première entrée, chaque élément de données indiquant la présence ou l'absence d'un objet d'écran supplémentaire dans une trame donnée ;
>
> l'obtention, en tant que sortie du réseau neuronal artificiel, de métadonnées extraites indiquant la présence ou l'absence d'un objet d'écran supplémentaire dans une trame donnée ;
>
> l'application d'une ou plusieurs fonctions pour réduire une erreur entre les métadonnées extraites de sortie du réseau neuronal artificiel et les éléments de données correspondants pour la trame donnée ;
>
> dans lequel les première et seconde entrées sont fournies dans le procédé d'obtention de données d'apprentissage selon l'une quelconque des revendications 1 à 4.

7. Procédé selon la revendication 6, dans lequel l'apprentissage du réseau neuronal artificiel comprend en outre :

> la détection de la présence des une ou plusieurs caractéristiques spécifiques dans les trames vidéo rendues ; et
>
> la fourniture d'un identifiant de sortie binaire pour indiquer la présence ou l'absence d'un objet d'écran supplémentaire dans les trames vidéo rendues, sur la base du résultat de l'étape de détection.

8. Procédé de test de données vidéo rendues au niveau ou en utilisant un dispositif multimédia (108, 208), le procédé comprenant :

> la réception au niveau d'un dispositif multimédia (108, 208) d'une pluralité de trames vidéo à rendre en fonction de métadonnées d'origine, chaque trame comprenant un ou plusieurs objets d'écran primaires et au moins un objet d'écran

supplémentaire ;

> le rendu des trames reçues au niveau ou en utilisant le dispositif multimédia (108, 208) dans lequel l'au moins un objet d'écran supplémentaire est superposé sur les un ou plusieurs objets d'écran primaires d'une trame donnée pendant le rendu, la pluralité reçue de trames étant rendues en fonction de métadonnées d'origine ;
>
> la fourniture des trames rendues à un réseau neuronal artificiel, ledit réseau neuronal artificiel suivant un apprentissage en fonction du procédé selon la revendication 6 ou la revendication 7 ;
>
> l'obtention de métadonnées extraites en tant que sortie du réseau neuronal artificiel, les métadonnées extraites détectant la présence ou l'absence d'un objet d'écran supplémentaire dans une trame rendue donnée ;
>
> la fourniture des métadonnées d'origine associées à la pluralité de trames vidéo au réseau neuronal artificiel ; et
>
> la validation du rendu d'un objet d'écran supplémentaire détecté sur la base des métadonnées d'origine concernant la trame vidéo donnée.

9. Procédé selon la revendication 8, dans lequel l'étape de validation comprend en outre :

> l'identification d'une caractéristique spécifique de l'objet d'écran supplémentaire détecté ;
>
> la comparaison de la caractéristique identifiée à une caractéristique spécifique liée dans les métadonnées d'origine ; et
>
> la détection de la présence d'une variation de la caractéristique spécifique identifiée sur la base du résultat de l'étape de comparaison.
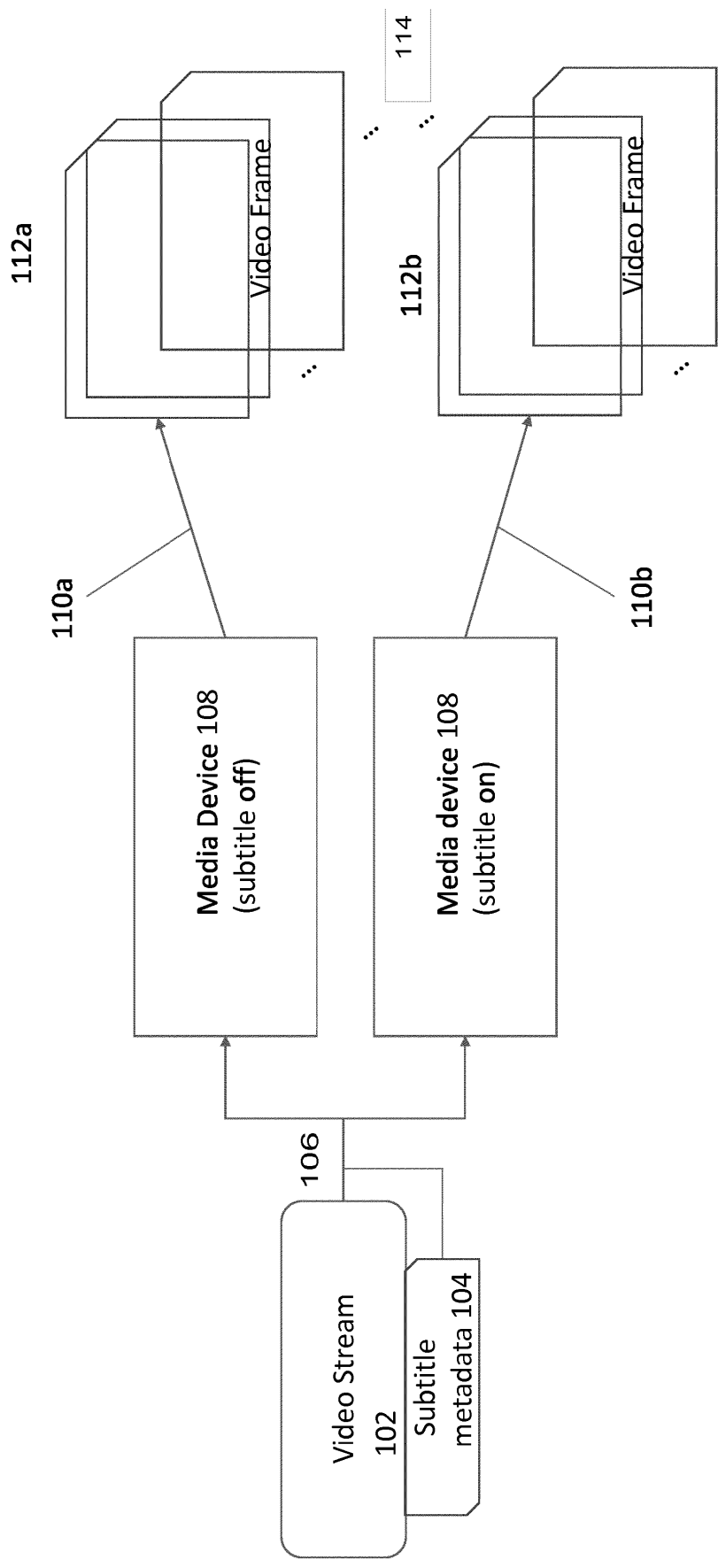
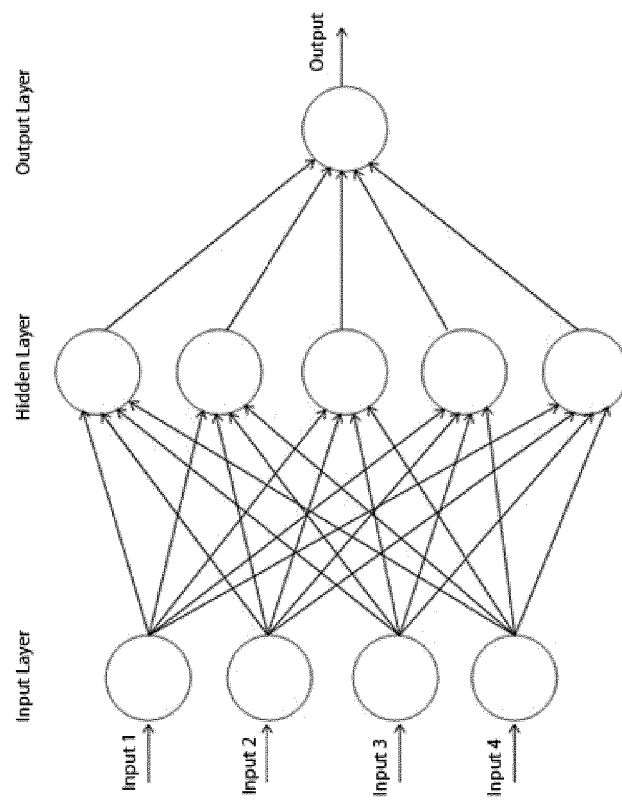10. Procédé selon 9, comprenant en outre :

> en réponse à une variation détectée, le calcul d'un écart sur la base de la variation détectée ;
>
> la fourniture de l'écart en tant que retour à un processeur associé au dispositif multimédia (108, 208) ; et
>
> l'ajustement d'un rendu suivant des trames vidéo sur la base de l'écart.

11. Système de test de données vidéo rendues au niveau ou en utilisant un dispositif multimédia (108, 208), le système incluant le dispositif multimédia (108, 208) et au moins un processeur associé à une mémoire pour stocker une pluralité d'instructions, dans lequel l'au moins un processeur est configuré pour exécuter la pluralité d'instructions pour mettre en œuvre le procédé selon l'une quelconque des revendications 8 à 10.

Figure 1

114

112a

Video Frame

112b

Video Frame

110a

110b

Media Device 108
(subtitle off)

Media device 108
(subtitle on)

106

Video Stream
102

Subtitle
metadata 104

Figure 1a

Data model 114

Figure 2

Video Frame
212

210

Media Device 208
(subtitle on)

206

Video Stream
202

Subtitle
metadata **204**

214

Subtitle
Detection unit
216

218

Subtitle
Validation unit
224

214

detected subtitle metadata220
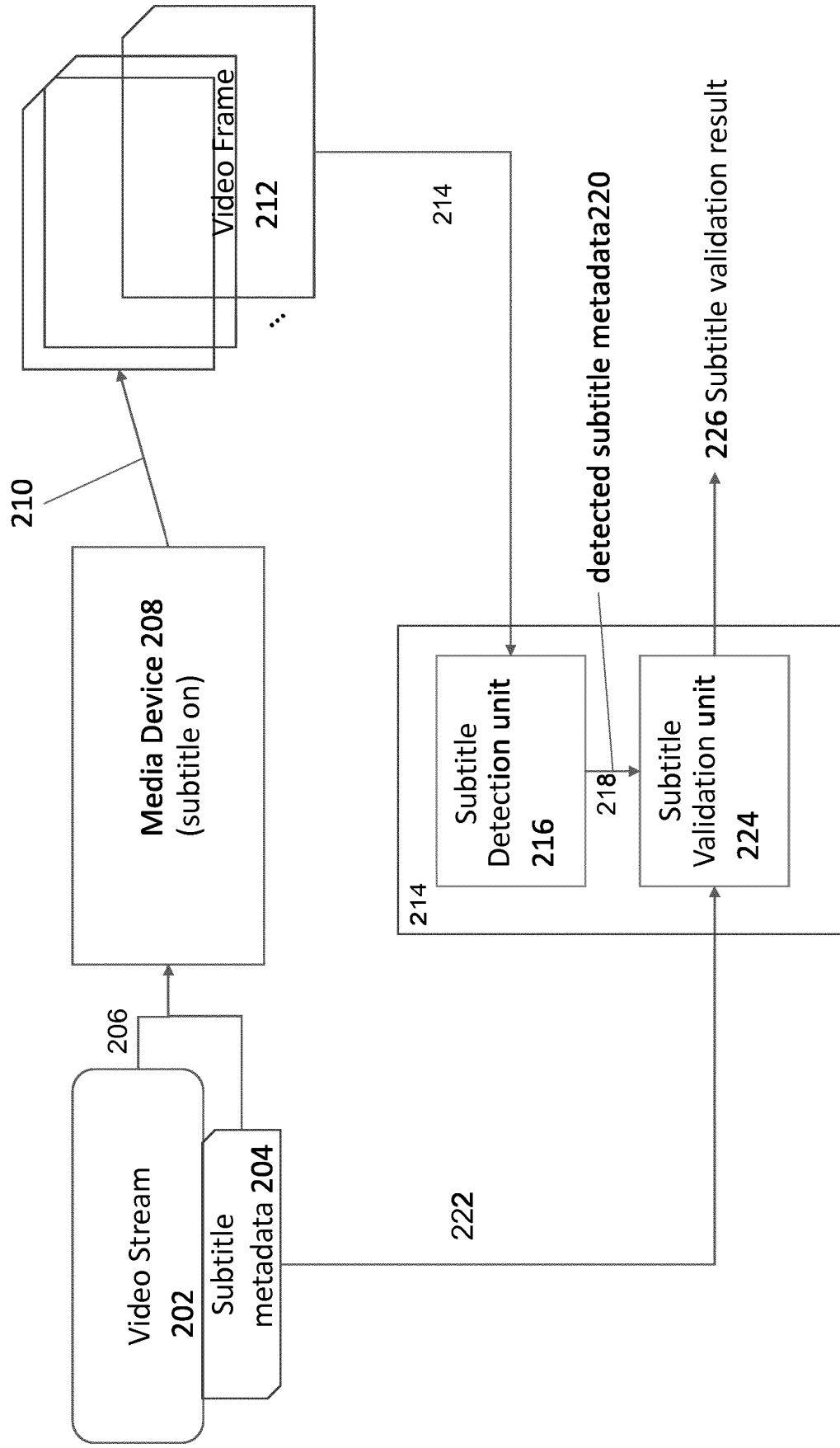
226 Subtitle validation result

222

Figure 3

20 seconds stream example

Original subtitle metadata

1
00:00:02.000 --> 00:00:05.000
First subtitle

2
00:00:10.000 --> 00:00:15.000
Second subtitle

Orginal Subtitle metadata 204

Extracted subtitle metadata

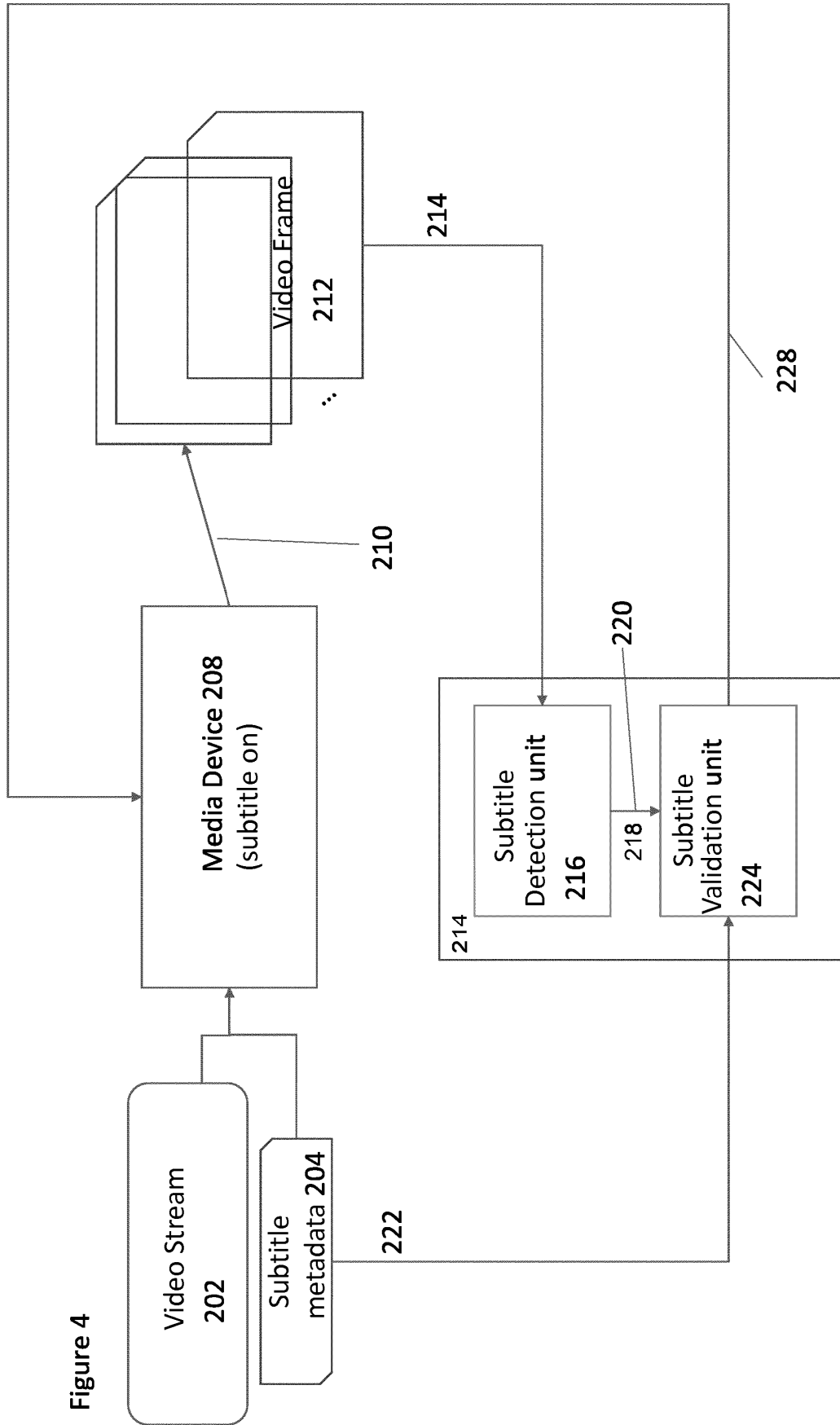|	
00:00:00.045	0
00:00:01.050	0
00:00:02.046	1
00:00:03.053	1
00:00:04.067	1
00:00:05.023	0
00:00:06.045	0
00:00:07.056	0
00:00:08.050	0
00:00:09.034	0
00:00:10.045	0
00:00:11.034	0
00:00:12.042	0
00:00:13.048	1
00:00:14.046	1
00:00:15.051	1
00:00:16.050	1
00:00:17.047	1
00:00:18.044	0
00:00:19.045	0

Detected metadata 220
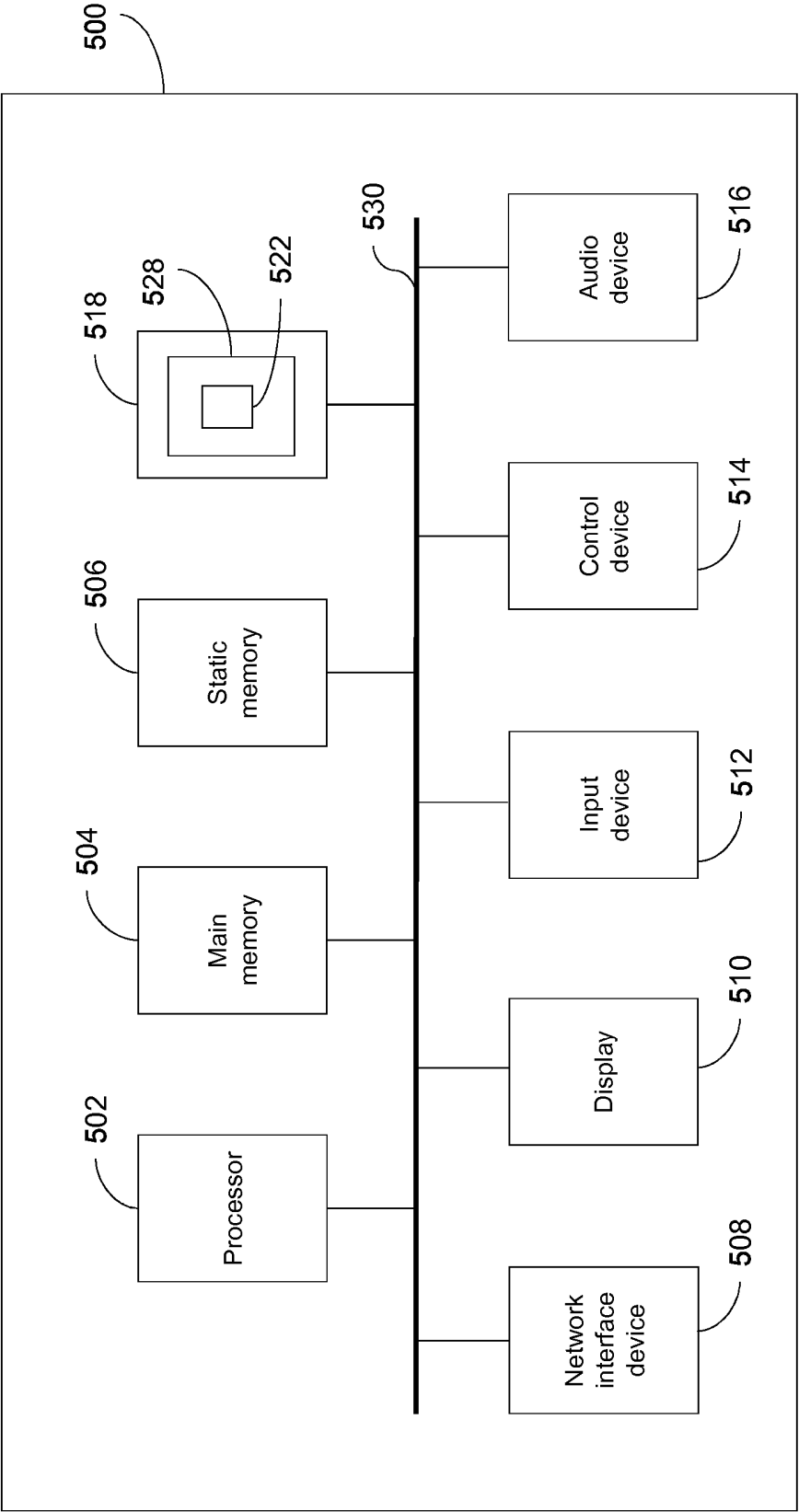
Figure 4

Fig. 5

Figure 5

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- KR 20170047547 **[0002]**

Non-patent literature cited in the description

- **A. SABATER.** *Automatic Subtitle Synchronization - Machine Learnings,* 14 September 2017, https://machinelearnings.co/automatic-subtitle-synchronization-e188a9275617 **[0002]**