Innovation, Sciences et Développement économique Canada
Office de la Propriété Intellectuelle du Canada

Innovation, Science and Economic Development Canada
Canadian Intellectual Property Office

CA 3135872 A1 2021/04/01

(21) 3 135 872

(12) DEMANDE DE BREVET CANADIEN
CANADIAN PATENT APPLICATION

(13) A1

(86) Date de dépôt PCT/PCT Filing Date: 2020/09/11
(87) Date publication PCT/PCT Publication Date: 2021/04/01
(85) Entrée phase nationale/National Entry: 2021/09/30
(86) N° demande PCT/PCT Application No.: US 2020/050329
(87) N° publication PCT/PCT Publication No.: 2021/061418
(30) Priorités/Priorities: 2019/09/27 (US62/907,352); 2020/09/02 (US17/010,028)

(51) Cl.Int./Int.Cl. *H04N 19/172* (2014.01), *H04N 19/33* (2014.01), *H04N 19/37* (2014.01)

(71) Demandeur/Applicant:
TENCENT AMERICA LLC, US

(72) Inventeurs/Inventors:
CHOI, BYEONGDOO, US;
WENGER, STEPHAN, US;
LIU, SHAN, US

(74) Agent: BORDEN LADNER GERVAIS LLP

(54) Titre : PROCEDE DE SIGNALISATION D'UN ENSEMBLE DE COUCHES DE SOUS-IMAGES DE SORTIE
(54) Title: METHOD FOR SIGNALING OUTPUT SUBPICTURE LAYER SET

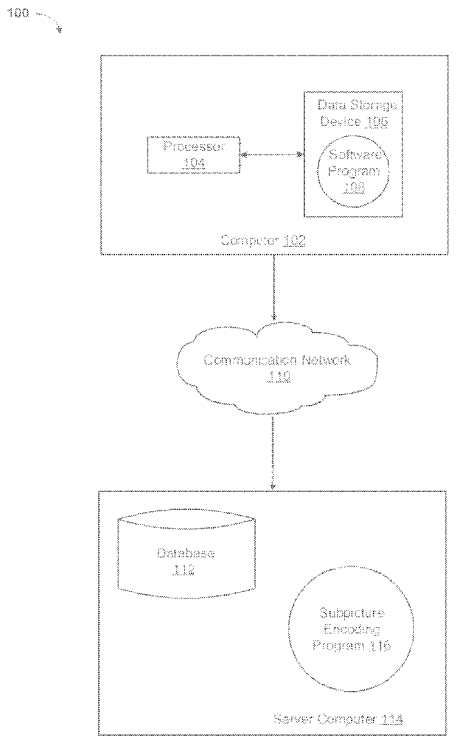


FIG. 1

(57) Abrégé/Abstract:
A method, computer program, and computer system is provided for coding a video sequence. Video information corresponding to one or subpictures within a picture is received. A first subpicture is identified from among the one or more subpictures as a region of interest. The first subpicture corresponding to the region of interest is encoded in a high quality mode. One or more other subpictures from among the one or more subpictures is encoded in a low quality mode. The first encoded subpicture and the encoded one or more other subpictures are output with one or more output layer sets.

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
01 April 2021 (01.04.2021)

WIPO | PCT

(10) International Publication Number
WO 2021/061418 A1

(51) International Patent Classification:
H04N 9/75 (2006.01)

(21) International Application Number:
PCT/US2020/050329

(22) International Filing Date:
11 September 2020 (11.09.2020)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/907,352 27 September 2019 (27.09.2019) US
17/010,028 02 September 2020 (02.09.2020) US

(71) Applicant: **TENCENT AMERICA LLC** [US/US]; 2747 Park Boulevard, Palo Alto, California 94306 (US).

(72) Inventors: **CHOI, Byeongdoo**; c/o TENCENT AMERICA LLC, 2747 Park Boulevard, Palo Alto, California 94306 (US). **WENGER, Stephan**; c/o TENCENT AMERICA LLC, 2747 Park Boulevard, Palo Alto, California 94306 (US). **LIU, Shan**; c/o TENCENT AMERICA LLC, 2747 Park Boulevard, Palo Alto, California 94306 (US).

(74) Agent: **RABENA, John F.** et al.; SUGHRUE MION, PLLC, 2000 Pennsylvania Ave., N.W., Suite 900, Washington, District of Columbia 20006 (US).

(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, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, IT, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, 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, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, 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, ST, 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).

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

(54) **Title:** METHOD FOR SIGNALING OUTPUT SUBPICTURE LAYER SET

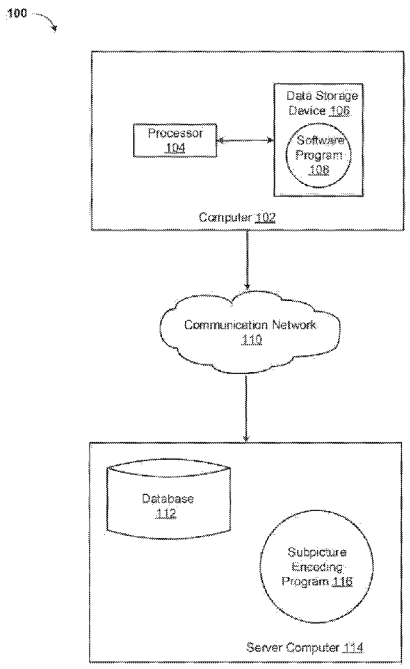


FIG. 1

(57) **Abstract:** A method, computer program, and computer system is provided for coding a video sequence. Video information corresponding to one or subpictures within a picture is received. A first subpicture is identified from among the one or more subpictures as a region of interest. The first subpicture corresponding to the region of interest is encoded in a high quality mode. One or more other subpictures from among the one or more subpictures is encoded in a low quality mode. The first encoded subpicture and the encoded one or more other subpictures are output with one or more output layer sets.

WO 2021/061418 A1

METHOD FOR SIGNALING OUTPUT SUBPICTURE LAYER SET

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority from U.S. Provisional Patent Application No. 62/907,352 filed September 27, 2019, and U.S. Patent Application No. 17/010,028, filed September 2, 2020, the entirety of which are incorporated herein.

BACKGROUND

[0002] This disclosure relates generally to field of computing, and more particularly to video encoding.

[0003] Recent proposed contributions to JVET-P0225 include signaling output layer sets and PTL information. The output layer set with profile-tier-level (PTL) information provides the operation points of a multi-layered bitstream.

SUMMARY

[0004] Embodiments relate to a method, system, and computer readable medium for coding a video sequence. According to one aspect, a method for coding a video sequence is provided. The method may include receiving video information corresponding to one or subpictures within a picture. A first subpicture is identified from among the one or more subpictures as a region of interest. The first subpicture corresponding to the region of interest is encoded in a high quality mode. One or more other subpictures from among the one or more subpictures is encoded in a low quality mode. The first encoded subpicture and the encoded one or more other subpictures are output with one or more output layer sets.

[0005] According to another aspect, a computer system for coding a video sequence is provided. The computer system may include one or more processors, one or more computer-readable memories, one or more computer-readable tangible storage devices, and program instructions stored on at least one of the one or more storage devices for execution by at least one of the one or more processors via at least one of the one or more memories, whereby the computer system is capable of performing a method. The method may include receiving video information corresponding to one or subpictures within a picture. A first subpicture is identified from among the one or more subpictures as a region of interest. The first subpicture

corresponding to the region of interest is encoded in a high quality mode. One or more other subpictures from among the one or more subpictures is encoded in a low quality mode. The first encoded subpicture and the encoded one or more other subpictures are output with one or more output layer sets.

[0006] According to yet another aspect, a computer readable medium for coding a video sequence is provided. The computer readable medium may include one or more computer-readable storage devices and program instructions stored on at least one of the one or more tangible storage devices, the program instructions executable by a processor. The program instructions are executable by a processor for performing a method that may accordingly include receiving video information corresponding to one or subpictures within a picture. A first subpicture is identified from among the one or more subpictures as a region of interest. The first subpicture corresponding to the region of interest is encoded in a high quality mode. One or more other subpictures from among the one or more subpictures is encoded in a low quality mode. The first encoded subpicture and the encoded one or more other subpictures are output with one or more output layer sets.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] These and other objects, features and advantages will become apparent from the following detailed description of illustrative embodiments, which is to be read in connection with the accompanying drawings. The various features of the drawings are not to scale as the illustrations are for clarity in facilitating the understanding of one skilled in the art in conjunction with the detailed description. In the drawings:

FIG. 1 illustrates a networked computer environment according to at least one embodiment;

FIG. 2 is a set of exemplary syntax elements according to at least one embodiment;

FIG. 3 is an operational flowchart illustrating the steps carried out by a program for encoding video with subpictures in multiple layer sets, according to at least one embodiment;

FIG. 4 is a block diagram of internal and external components of computers and servers depicted in FIG. 1 according to at least one embodiment;

FIG. 5 is a block diagram of an illustrative cloud computing environment including the computer system depicted in FIG. 1, according to at least one embodiment; and

FIG. 6 is a block diagram of functional layers of the illustrative cloud computing environment of FIG. 5, according to at least one embodiment.

DETAILED DESCRIPTION

[0008] Detailed embodiments of the claimed structures and methods are disclosed herein; however, it can be understood that the disclosed embodiments are merely illustrative of the claimed structures and methods that may be embodied in various forms. Those structures and methods may, however, be embodied in many different forms and should not be construed as limited to the exemplary embodiments set forth herein. Rather, these exemplary embodiments are provided so that this disclosure will be thorough and complete and will fully convey the scope to those skilled in the art. In the description, details of well-known features and techniques may be omitted to avoid unnecessarily obscuring the presented embodiments.

[0009] Embodiments relate generally to the field of computing, and more particularly to video encoding. The following described exemplary embodiments provide a system, method and computer program to, among other things, encode video with subpictures in multiple layer sets. Therefore, some embodiments have the capacity to improve the field of computing by allowing for different encoding schemes of varying quality to be used in encoding subpictures within the video data.

[0010] As previously described, recent proposed contributions to JVET-P0225 include signaling output layer sets and PTL information. The output layer set with profile-tier-level (PTL) information provides the operation points of a multi-layered bitstream. However, the PTL information cannot define operation points combined with subpictures, since each subpicture in each layer may or may not be present in some applications. For example, in a picture-in-picture (PIP) use case, a region of interest (ROI) may be enlarged and encoded with a high quality as a subpicture in an enhancement layer, while all areas (i.e., all subpictures) in the base layer may be encoded with a low quality. In an output mode, the subpicture in the enhancement layer may be decoded and outputted, while all areas of the base layer may decoded and outputted in another output mode. In a specific output mode, the output layer set can be composed of one or more

subpictures from the base layer and one or more subpictures from enhancement layers, in order to show an enhanced visual quality at an ROI (e.g. 360 viewport dependent processing). It may be advantageous, therefore, to include syntax elements for signaling the encoding of multiple subpictures contained across multiple layers within video data.

[0011] Aspects are described herein with reference to flowchart illustrations and/or block diagrams of methods, apparatus (systems), and computer readable media according to the various embodiments. It will be understood that each block of the flowchart illustrations and/or block diagrams, and combinations of blocks in the flowchart illustrations and/or block diagrams, can be implemented by computer readable program instructions.

[0012] The following described exemplary embodiments provide a system, method and computer program for encoding video with subpictures in multiple layer sets. Referring now to FIG. 1, a functional block diagram of a networked computer environment illustrating a system 100 (hereinafter "system") for encoding video with subpictures in multiple layer sets. It should be appreciated that FIG. 1 provides only an illustration of one implementation and does not imply any limitations with regard to the environments in which different embodiments may be implemented. Many modifications to the depicted environments may be made based on design and implementation requirements.

[0013] The system 100 may include a computer 102 and a server computer 114. The computer 102 may communicate with the server computer 114 via a communication network 110 (hereinafter "network"). The computer 102 may include a processor 104 and a software program 108 that is stored on a data storage device 106 and is enabled to interface with a user and communicate with the server computer 114. As will be discussed below with reference to FIG. 4 the computer 102 may include internal components 800A and external components 900A, respectively, and the server computer 114 may include internal components 800B and external components 900B, respectively. The computer 102 may be, for example, a mobile device, a telephone, a personal digital assistant, a netbook, a laptop computer, a tablet computer, a desktop computer, or any type of computing devices capable of running a program, accessing a network, and accessing a database.

[0014] The server computer 114 may also operate in a cloud computing service model, such

as Software as a Service (SaaS), Platform as a Service (PaaS), or Infrastructure as a Service (laaS), as discussed below with respect to FIGS. 5 and 6. The server computer 114 may also be located in a cloud computing deployment model, such as a private cloud, community cloud, public cloud, or hybrid cloud.

[0015] The server computer 114, which may be used for encoding video with subpictures in multiple layer sets is enabled to run a Subpicture Encoding Program 116 (hereinafter "program") that may interact with a database 112. The Subpicture Encoding Program method is explained in more detail below with respect to FIG. 3. In one embodiment, the computer 102 may operate as an input device including a user interface while the program 116 may run primarily on server computer 114. In an alternative embodiment, the program 116 may run primarily on one or more computers 102 while the server computer 114 may be used for processing and storage of data used by the program 116. It should be noted that the program 116 may be a standalone program or may be integrated into a larger subpicture encoding program.

[0016] It should be noted, however, that processing for the program 116 may, in some instances be shared amongst the computers 102 and the server computers 114 in any ratio. In another embodiment, the program 116 may operate on more than one computer, server computer, or some combination of computers and server computers, for example, a plurality of computers 102 communicating across the network 110 with a single server computer 114. In another embodiment, for example, the program 116 may operate on a plurality of server computers 114 communicating across the network 110 with a plurality of client computers. Alternatively, the program may operate on a network server communicating across the network with a server and a plurality of client computers.

[0017] The network 110 may include wired connections, wireless connections, fiber optic connections, or some combination thereof. In general, the network 110 can be any combination of connections and protocols that will support communications between the computer 102 and the server computer 114. The network 110 may include various types of networks, such as, for example, a local area network (LAN), a wide area network (WAN) such as the Internet, a telecommunication network such as the Public Switched Telephone Network (PSTN), a wireless network, a public switched network, a satellite network, a cellular network (e.g., a fifth

generation (5G) network, a long-term evolution (LTE) network, a third generation (3G) network, a code division multiple access (CDMA) network, etc.), a public land mobile network (PLMN), a metropolitan area network (MAN), a private network, an ad hoc network, an intranet, a fiber optic-based network, or the like, and/or a combination of these or other types of networks.

[0018] The number and arrangement of devices and networks shown in FIG. 1 are provided as an example. In practice, there may be additional devices and/or networks, fewer devices and/or networks, different devices and/or networks, or differently arranged devices and/or networks than those shown in FIG. 1. Furthermore, two or more devices shown in FIG. 1 may be implemented within a single device, or a single device shown in FIG. 1 may be implemented as multiple, distributed devices. Additionally, or alternatively, a set of devices (e.g., one or more devices) of system 100 may perform one or more functions described as being performed by another set of devices of system 100.

[0019] Referring now to FIG. 2, exemplary syntax elements 200 is depicted according to one or more embodiments. The syntax elements 200 may include, among other things, a flag that may indicate whether the subpicture partitions may be aligned across layers, a flag that may indicate whether one or more subpicture identification value present in each layer, a function to output subpicture IDs that may be present in each output layer of each output layer set, and PTL information for each output layer set combined with output subpictures. It may be assumed that the syntax elements mapping a subpicture ID to a specific region of a picture may be presented in a parameter set or elsewhere. More specifically, the syntax elements 200 may include:

vps_subpics_present_flag equal to 1 may specify that the value of subpics_present_flag of one or more SPSs referring to this VPS may be equal to 1. vps_subpics_present_flag equal to 0 may specify that the value of subpics_present_flag of any SPS referring to this VPS may be equal to 0.

vps_subpic_aligned_across_layers_flag equal to 1 may specify that that vps_max_subpics_minus1 and vps_sub_pic_id_layer[j] may be present in VPS. vps_max_subpics_minus1[i] may be inferred to be equal to vps_max_subpics_minus1, and vps_sub_pic_id_layer[i][j] may be inferred to be equal to vps_sub_pic_id_layer[j], when i is greater than 0. vps_subpic_aligned_across_layers_flag equal to 0 may specify that vps_max_subpics_minus1[i] and vps_sub_pic_id_layer[i][j] may be present, when i is in the

range of 0 to vps_max_layers_minus1, inclusive.

vps_max_subpics_minus1[i] plus 1 may specify the maximum number of subpictures of the i-th layer in the CVS referring to the VPS. vps_max_subpics_minus1[i] may be equal to max_subpics_minus1 of the SPS with nuh_layer_id equal to vps_layer_id[i].

vps_sub_pic_id_layer[i][j] may specify the subpicture ID of the j-th subpicture of the layer with nuh_layer_id equal to vps_layer_id[i]. For example,

```
if(vps_subpic_aligned_across_layers_flag)
    for(i=1; i< vps_max_layers_minus1; i++) {
            vps_max_subpics_minus1[i] = vps_max_subpics_minus1
            for(j = 0; j <= vps_max_subpics_minus1[i]; j++ ) {
                    vps_sub_pic_id_layer[i][j] = vps_sub_pic_id_layer[j]
            }
    }
```

num_output_layer_sets_minus1 plus 1 may specify the number of output layer set in the coded vide sequence referring to the VPS. When not present, the value of num_output_layer_sets_minus1 may be inferred to be equal to 0.

num_profile_tile_levels_minus1 plus 1 may specify the number of profile/tier/level information in the coded vide sequence referring to the VPS. When not present, the value of num_profile_tile_levels_minus1 may be inferred to be equal to 0.

vps_output_layers_mode[i] equal to 0 may specify that only the highest layer may be output in the i-th output layer set. vps_output_layer_mode[i] equal to 1 may specify that all layers may be output in the i-th output layer set. vps_output_layer_mode[i] equal to 2 may specify that the layers that are output may be the layers with vps_output_layer_flag[i][j] equal to 1 in the i-th output layer set. The value of vps_output_layers_mode[i] may be in the range of 0 to 2, inclusive.

vps_output_layer_flag[i][j] equal to 1 may specify that the j-th layer of the i-th output layer set may be output. vps_output_layer_flag[i] [j] equal to 0 may specify that the j-th layer of the i-th output layer set may be not output.

profile_tier_level_idx[i][j] may specify the index, into the list of profile_tier_level( ) syntax structures in the VPS, of the profile_tier_level( ) syntax structure that applies to the j-th layer of the i-th output layer set.

all_subpic_output_flag[i][j] equal to 1 may specify that all subpictures of the j-th layer of the i-th output layer set may be outputted. all_subpic_output_flag[i][j] equal to 0 may

specify that one or more subpictures of the j-th layer of the i-th output layer set may be outputted.

num_output_subpic_layer_minus1[i][j] may specify the number of output subpictures of the j-th layer of the i-th output layer set.

output_sub_pic_id_layer[i][j][k] may specify the subpicture ID of the k-th subpicture of the j-th layer of the i-th output layer set.

[0020] Referring now to FIG. 3, an operational flowchart 300 illustrating the steps carried out by a program for encoding video with subpictures in multiple layer sets is depicted. FIG. 3 may be described with the aid of FIGS. 1 and 2. As previously described, the Subpicture Encoding Program 116 (FIG. 1) may quickly and effectively signal the encoding of multiple subpictures contained across multiple layers within video data.

[0021] At 302, video information corresponding to one or subpictures within a picture is received. The video information may include multiple layer sets each containing multiple layers and the one or more subpictures. The subpictures may be aligned within each of the layers and across multiple layer sets. In operation, the Subpicture Encoding Program 116 (FIG. 1) on the server computer 114 (FIG. 1) may receive video data. The video data may be received from the computer 102 (FIG. 1) over the communication network 110 (FIG. 1) or may be retrieved from the database 112 (FIG. 1).

[0022] At 304, a first subpicture is identified from among the one or more subpictures as a region of interest. The first subpicture may be a picture-in-picture region in the video data that may be extracted from a base layer for export at a higher quality. In operation, the Subpicture Encoding Program 116 (FIG. 1) identify a subpicture as a region of interest using one or more of the syntax elements 200 (FIG. 2).

[0023] At 306, the first subpicture corresponding to the region of interest is encoded in a high quality mode. The region of interest may be enlarged and encoded in a high quality mode as a subpicture in an enhancement layer, while the remaining subpictures in the base layer may be encoded with a low quality to save on bandwidth and processing power. In operation, the Subpicture Encoding Program 116 (FIG. 1) may encode the identified region of interest in a high quality and may output the encoded video information via the communication network 110 (FIG. 1).

[0024] It may be appreciated that FIG. 3 provides only an illustration of one implementation and does not imply any limitations with regard to how different embodiments may be implemented. Many modifications to the depicted environments may be made based on design and implementation requirements.

[0025] FIG. 4 is a block diagram 400 of internal and external components of computers depicted in FIG. 1 in accordance with an illustrative embodiment. It should be appreciated that FIG. 4 provides only an illustration of one implementation and does not imply any limitations with regard to the environments in which different embodiments may be implemented. Many modifications to the depicted environments may be made based on design and implementation requirements.

[0026] Computer 102 (FIG. 1) and server computer 114 (FIG. 1) may include respective sets of internal components 800A,B and external components 900A,B illustrated in FIG 4. Each of the sets of internal components 800 include one or more processors 820, one or more computer-readable RAMs 822 and one or more computer-readable ROMs 824 on one or more buses 826, one or more operating systems 828, and one or more computer-readable tangible storage devices 830.

[0027] Processor 820 is implemented in hardware, firmware, or a combination of hardware and software. Processor 820 is a central processing unit (CPU), a graphics processing unit (GPU), an accelerated processing unit (APU), a microprocessor, a microcontroller, a digital signal processor (DSP), a field-programmable gate array (FPGA), an application-specific integrated circuit (ASIC), or another type of processing component. In some implementations, processor 820 includes one or more processors capable of being programmed to perform a function. Bus 826 includes a component that permits communication among the internal components 800A,B.

[0028] The one or more operating systems 828, the software program 108 (FIG. 1) and the Subpicture Encoding Program 116 (FIG. 1) on server computer 114 (FIG. 1) are stored on one or more of the respective computer-readable tangible storage devices 830 for execution by one or more of the respective processors 820 via one or more of the respective RAMs 822 (which typically include cache memory). In the embodiment illustrated in FIG. 4, each of the computer-

readable tangible storage devices 830 is a magnetic disk storage device of an internal hard drive. Alternatively, each of the computer-readable tangible storage devices 830 is a semiconductor storage device such as ROM 824, EPROM, flash memory, an optical disk, a magneto-optic disk, a solid state disk, a compact disc (CD), a digital versatile disc (DVD), a floppy disk, a cartridge, a magnetic tape, and/or another type of non-transitory computer-readable tangible storage device that can store a computer program and digital information.

[0029] Each set of internal components 800A,B also includes a R/W drive or interface 832 to read from and write to one or more portable computer-readable tangible storage devices 936 such as a CD-ROM, DVD, memory stick, magnetic tape, magnetic disk, optical disk or semiconductor storage device. A software program, such as the software program 108 (FIG. 1) and the Subpicture Encoding Program 116 (FIG. 1) can be stored on one or more of the respective portable computer-readable tangible storage devices 936, read via the respective R/W drive or interface 832 and loaded into the respective hard drive 830.

[0030] Each set of internal components 800A,B also includes network adapters or interfaces 836 such as a TCP/IP adapter cards; wireless Wi-Fi interface cards; or 3G, 4G, or 5G wireless interface cards or other wired or wireless communication links. The software program 108 (FIG. 1) and the Subpicture Encoding Program 116 (FIG. 1) on the server computer 114 (FIG. 1) can be downloaded to the computer 102 (FIG. 1) and server computer 114 from an external computer via a network (for example, the Internet, a local area network or other, wide area network) and respective network adapters or interfaces 836. From the network adapters or interfaces 836, the software program 108 and the Subpicture Encoding Program 116 on the server computer 114 are loaded into the respective hard drive 830. The network may comprise copper wires, optical fibers, wireless transmission, routers, firewalls, switches, gateway computers and/or edge servers.

[0031] Each of the sets of external components 900A,B can include a computer display monitor 920, a keyboard 930, and a computer mouse 934. External components 900A,B can also include touch screens, virtual keyboards, touch pads, pointing devices, and other human interface devices. Each of the sets of internal components 800A,B also includes device drivers 840 to interface to computer display monitor 920, keyboard 930 and computer mouse 934. The device drivers 840, R/W drive or interface 832 and network adapter or interface 836 comprise hardware

and software (stored in storage device 830 and/or ROM 824).

[0032] It is understood in advance that although this disclosure includes a detailed description on cloud computing, implementation of the teachings recited herein are not limited to a cloud computing environment. Rather, some embodiments are capable of being implemented in conjunction with any other type of computing environment now known or later developed.

[0033] Cloud computing is a model of service delivery for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, network bandwidth, servers, processing, memory, storage, applications, virtual machines, and services) that can be rapidly provisioned and released with minimal management effort or interaction with a provider of the service. This cloud model may include at least five characteristics, at least three service models, and at least four deployment models.

[0034] Characteristics are as follows:

On-demand self-service: a cloud consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with the service's provider.

Broad network access: capabilities are available over a network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, laptops, and PDAs).

Resource pooling: the provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to demand. There is a sense of location independence in that the consumer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter).

Rapid elasticity: capabilities can be rapidly and elastically provisioned, in some cases automatically, to quickly scale out and rapidly released to quickly scale in. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time.

Measured service: cloud systems automatically control and optimize resource use by

leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported providing transparency for both the provider and consumer of the utilized service.

[0035] Service Models are as follows:

Software as a Service (SaaS): the capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through a thin client interface such as a web browser (e.g., web-based e-mail). The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.

Platform as a Service (PaaS): the capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including networks, servers, operating systems, or storage, but has control over the deployed applications and possibly application hosting environment configurations.

Infrastructure as a Service (laaS): the capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, deployed applications, and possibly limited control of select networking components (e.g., host firewalls).

[0036] Deployment Models are as follows:

Private cloud: the cloud infrastructure is operated solely for an organization. It may be managed by the organization or a third party and may exist on-premises or off-premises.

Community cloud: the cloud infrastructure is shared by several organizations and supports a specific community that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be managed by the organizations or a third party and may exist on-premises or off-premises.

Public cloud: the cloud infrastructure is made available to the general public or a large industry group and is owned by an organization selling cloud services.

Hybrid cloud: the cloud infrastructure is a composition of two or more clouds (private, community, or public) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load-balancing between clouds).

[0037] A cloud computing environment is service oriented with a focus on statelessness, low coupling, modularity, and semantic interoperability. At the heart of cloud computing is an infrastructure comprising a network of interconnected nodes.

[0038] Referring to FIG. 5, illustrative cloud computing environment 500 is depicted. As shown, cloud computing environment 500 comprises one or more cloud computing nodes 10 with which local computing devices used by cloud consumers, such as, for example, personal digital assistant (PDA) or cellular telephone 54A, desktop computer 54B, laptop computer 54C, and/or automobile computer system 54N may communicate. Cloud computing nodes 10 may communicate with one another. They may be grouped (not shown) physically or virtually, in one or more networks, such as Private, Community, Public, or Hybrid clouds as described hereinabove, or a combination thereof. This allows cloud computing environment 500 to offer infrastructure, platforms and/or software as services for which a cloud consumer does not need to maintain resources on a local computing device. It is understood that the types of computing devices 54A-N shown in FIG. 5 are intended to be illustrative only and that cloud computing nodes 10 and cloud computing environment 500 can communicate with any type of computerized device over any type of network and/or network addressable connection (e.g., using a web browser).

[0039] Referring to FIG. 6, a set of functional abstraction layers 600 provided by cloud computing environment 500 (FIG. 5) is shown. It should be understood in advance that the components, layers, and functions shown in FIG. 6 are intended to be illustrative only and embodiments are not limited thereto. As depicted, the following layers and corresponding functions are provided:

[0040] Hardware and software layer 60 includes hardware and software components.

Examples of hardware components include: mainframes 61; RISC (Reduced Instruction Set Computer) architecture based servers 62; servers 63; blade servers 64; storage devices 65; and networks and networking components 66. In some embodiments, software components include network application server software 67 and database software 68.

[0041] Virtualization layer 70 provides an abstraction layer from which the following examples of virtual entities may be provided: virtual servers 71; virtual storage 72; virtual networks 73, including virtual private networks; virtual applications and operating systems 74; and virtual clients 75.

[0042] In one example, management layer 80 may provide the functions described below. Resource provisioning 81 provides dynamic procurement of computing resources and other resources that are utilized to perform tasks within the cloud computing environment. Metering and Pricing 82 provide cost tracking as resources are utilized within the cloud computing environment, and billing or invoicing for consumption of these resources. In one example, these resources may comprise application software licenses. Security provides identity verification for cloud consumers and tasks, as well as protection for data and other resources. User portal 83 provides access to the cloud computing environment for consumers and system administrators. Service level management 84 provides cloud computing resource allocation and management such that required service levels are met. Service Level Agreement (SLA) planning and fulfillment 85 provide pre-arrangement for, and procurement of, cloud computing resources for which a future requirement is anticipated in accordance with an SLA.

[0043] Workloads layer 90 provides examples of functionality for which the cloud computing environment may be utilized. Examples of workloads and functions which may be provided from this layer include: mapping and navigation 91; software development and lifecycle management 92; virtual classroom education delivery 93; data analytics processing 94; transaction processing 95; and Subpicture Encoding 96. Subpicture Encoding 96 may encode video with subpictures in multiple layer sets.

[0044] Some embodiments may relate to a system, a method, and/or a computer readable medium at any possible technical detail level of integration. The computer readable medium may include a computer-readable non-transitory storage medium (or media) having computer readable

program instructions thereon for causing a processor to carry out operations.

[0045] The computer readable storage medium can be a tangible device that can retain and store instructions for use by an instruction execution device. The computer readable storage medium may be, for example, but is not limited to, an electronic storage device, a magnetic storage device, an optical storage device, an electromagnetic storage device, a semiconductor storage device, or any suitable combination of the foregoing. A non-exhaustive list of more specific examples of the computer readable storage medium includes the following: a portable computer diskette, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM or Flash memory), a static random access memory (SRAM), a portable compact disc read-only memory (CD-ROM), a digital versatile disk (DVD), a memory stick, a floppy disk, a mechanically encoded device such as punch-cards or raised structures in a groove having instructions recorded thereon, and any suitable combination of the foregoing. A computer readable storage medium, as used herein, is not to be construed as being transitory signals per se, such as radio waves or other freely propagating electromagnetic waves, electromagnetic waves propagating through a waveguide or other transmission media (e.g., light pulses passing through a fiber-optic cable), or electrical signals transmitted through a wire.

[0046] Computer readable program instructions described herein can be downloaded to respective computing/processing devices from a computer readable storage medium or to an external computer or external storage device via a network, for example, the Internet, a local area network, a wide area network and/or a wireless network. The network may comprise copper transmission cables, optical transmission fibers, wireless transmission, routers, firewalls, switches, gateway computers and/or edge servers. A network adapter card or network interface in each computing/processing device receives computer readable program instructions from the network and forwards the computer readable program instructions for storage in a computer readable storage medium within the respective computing/processing device.

[0047] Computer readable program code/instructions for carrying out operations may be assembler instructions, instruction-set-architecture (ISA) instructions, machine instructions, machine dependent instructions, microcode, firmware instructions, state-setting data,

configuration data for integrated circuitry, or either source code or object code written in any combination of one or more programming languages, including an object oriented programming language such as Smalltalk, C++, or the like, and procedural programming languages, such as the "C" programming language or similar programming languages. The computer readable program instructions may execute entirely on the user's computer, partly on the user's computer, as a stand-alone software package, partly on the user's computer and partly on a remote computer or entirely on the remote computer or server. In the latter scenario, the remote computer may be connected to the user's computer through any type of network, including a local area network (LAN) or a wide area network (WAN), or the connection may be made to an external computer (for example, through the Internet using an Internet Service Provider). In some embodiments, electronic circuitry including, for example, programmable logic circuitry, field-programmable gate arrays (FPGA), or programmable logic arrays (PLA) may execute the computer readable program instructions by utilizing state information of the computer readable program instructions to personalize the electronic circuitry, in order to perform aspects or operations.

[0048] These computer readable program instructions may be provided to a processor of a general purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions, which execute via the processor of the computer or other programmable data processing apparatus, create means for implementing the functions/acts specified in the flowchart and/or block diagram block or blocks. These computer readable program instructions may also be stored in a computer readable storage medium that can direct a computer, a programmable data processing apparatus, and/or other devices to function in a particular manner, such that the computer readable storage medium having instructions stored therein comprises an article of manufacture including instructions which implement aspects of the function/act specified in the flowchart and/or block diagram block or blocks.

[0049] The computer readable program instructions may also be loaded onto a computer, other programmable data processing apparatus, or other device to cause a series of operational steps to be performed on the computer, other programmable apparatus or other device to produce a computer implemented process, such that the instructions which execute on the computer, other programmable apparatus, or other device implement the functions/acts specified in the flowchart and/or block diagram block or blocks.

[0050] The flowchart and block diagrams in the Figures illustrate the architecture, functionality, and operation of possible implementations of systems, methods, and computer readable media according to various embodiments. In this regard, each block in the flowchart or block diagrams may represent a module, segment, or portion of instructions, which comprises one or more executable instructions for implementing the specified logical function(s). The method, computer system, and computer readable medium may include additional blocks, fewer blocks, different blocks, or differently arranged blocks than those depicted in the Figures. In some alternative implementations, the functions noted in the blocks may occur out of the order noted in the Figures. For example, two blocks shown in succession may, in fact, be executed concurrently or substantially concurrently, or the blocks may sometimes be executed in the reverse order, depending upon the functionality involved. It will also be noted that each block of the block diagrams and/or flowchart illustration, and combinations of blocks in the block diagrams and/or flowchart illustration, can be implemented by special purpose hardware-based systems that perform the specified functions or acts or carry out combinations of special purpose hardware and computer instructions.

[0051] It will be apparent that systems and/or methods, described herein, may be implemented in different forms of hardware, firmware, or a combination of hardware and software. The actual specialized control hardware or software code used to implement these systems and/or methods is not limiting of the implementations. Thus, the operation and behavior of the systems and/or methods were described herein without reference to specific software code—it being understood that software and hardware may be designed to implement the systems and/or methods based on the description herein.

[0052] No element, act, or instruction used herein should be construed as critical or essential unless explicitly described as such. Also, as used herein, the articles “a” and “an” are intended to include one or more items, and may be used interchangeably with “one or more.” Furthermore, as used herein, the term “set” is intended to include one or more items (e.g., related items, unrelated items, a combination of related and unrelated items, etc.), and may be used interchangeably with “one or more.” Where only one item is intended, the term “one” or similar language is used. Also, as used herein, the terms “has,” “have,” “having,” or the like are intended to be open-ended terms. Further, the phrase “based on” is intended to mean “based, at least in part, on” unless

explicitly stated otherwise.

[0053] The descriptions of the various aspects and embodiments have been presented for purposes of illustration, but are not intended to be exhaustive or limited to the embodiments disclosed. Even though combinations of features are recited in the claims and/or disclosed in the specification, these combinations are not intended to limit the disclosure of possible implementations. In fact, many of these features may be combined in ways not specifically recited in the claims and/or disclosed in the specification. Although each dependent claim listed below may directly depend on only one claim, the disclosure of possible implementations includes each dependent claim in combination with every other claim in the claim set. Many modifications and variations will be apparent to those of ordinary skill in the art without departing from the scope of the described embodiments. The terminology used herein was chosen to best explain the principles of the embodiments, the practical application or technical improvement over technologies found in the marketplace, or to enable others of ordinary skill in the art to understand the embodiments disclosed herein.

CLAIMS

What is claimed is:

1. A method of coding a video sequence comprising:

receiving video information corresponding to one or subpictures within a picture;

identifying a first subpicture from among the one or more subpictures as a region of interest; and

encoding the first subpicture corresponding to the region of interest in a high quality mode.

2. The method of claim 1, further comprising encoding one or more other subpictures from among the one or more subpictures in a low quality mode.

3. The method of claim 2, further comprising outputting the first encoded subpicture and the encoded one or more other subpictures with one or more output layer sets.

4. The method of claim 3, wherein each output layer set from among the one or more output layer sets comprises: one or more output layers and one or more of the subpictures.

5. The method of claim 4, wherein a flag indicates whether subpicture partitions are aligned across the one or more output layers.

6. The method of claim 4, wherein a flag indicates whether subpicture identification values are present in each of the one or more output layers.

7. The method of claim 6, further comprising outputting the subpicture identification values present in each output layer of each output layer set.

8. A computer system for coding a video sequence, the computer system comprising:

one or more computer-readable non-transitory storage media configured to store computer program code; and

one or more computer processors configured to access said computer program code and operate as instructed by said computer program code, said computer program code including:

receiving code configured to cause the one or more processors to receive video information corresponding to one or subpictures within a picture;

identifying code configured to cause the one or more processors to identify a first subpicture from among the one or more subpictures as a region of interest; and

first encoding code configured to cause the one or more processors to encode the first subpicture corresponding to the region of interest in a high quality mode.

9. The computer system of claim 8, further comprising second encoding code configured to cause the one or more processors to encode one or more other subpictures from among the one or more subpictures in a low quality mode.

10. The computer system of claim 9, further comprising outputting code configured to cause the one or more processors to output the first encoded subpicture and the encoded one or more other subpictures with one or more output layer sets.

11. The computer system of claim 10, wherein each output layer set from among the one or more output layer sets comprises: one or more output layers and one or more of the subpictures.

12. The computer system of claim 11, wherein a flag indicates whether subpicture partitions are aligned across the one or more output layers.

13. The computer system of claim 11, wherein a flag indicates whether subpicture identification values are present in each of the one or more output layers.

14. The computer system of claim 13, further comprising outputting code configured to cause the one or more processors to output the subpicture identification values present in each output layer of each output layer set.

15. A non-transitory computer readable medium having stored thereon a computer program

for coding a video sequence, the computer program configured to cause one or more computer processors to:

receive video information corresponding to one or subpictures within a picture;

identify a first subpicture from among the one or more subpictures as a region of interest; and

encode the first subpicture corresponding to the region of interest in a high quality mode.

16. The computer readable medium of claim 15, wherein the computer program is further configured to cause one or more computer processors to encode one or more other subpictures from among the one or more subpictures in a low quality mode.

17. The computer readable medium of claim 16, wherein the computer program is further configured to cause one or more computer processors to output the first encoded subpicture and the encoded one or more other subpictures with one or more output layer sets.

18. The computer readable medium of claim 17, wherein each output layer set from among the one or more output layer sets comprises: one or more output layers and one or more of the subpictures.

19. The computer readable medium of claim 18, wherein a flag indicates whether subpicture partitions are aligned across the one or more output layers.

20. The computer readable medium of claim 18, wherein a flag indicates whether subpicture identification values are present in each of the one or more output layers.

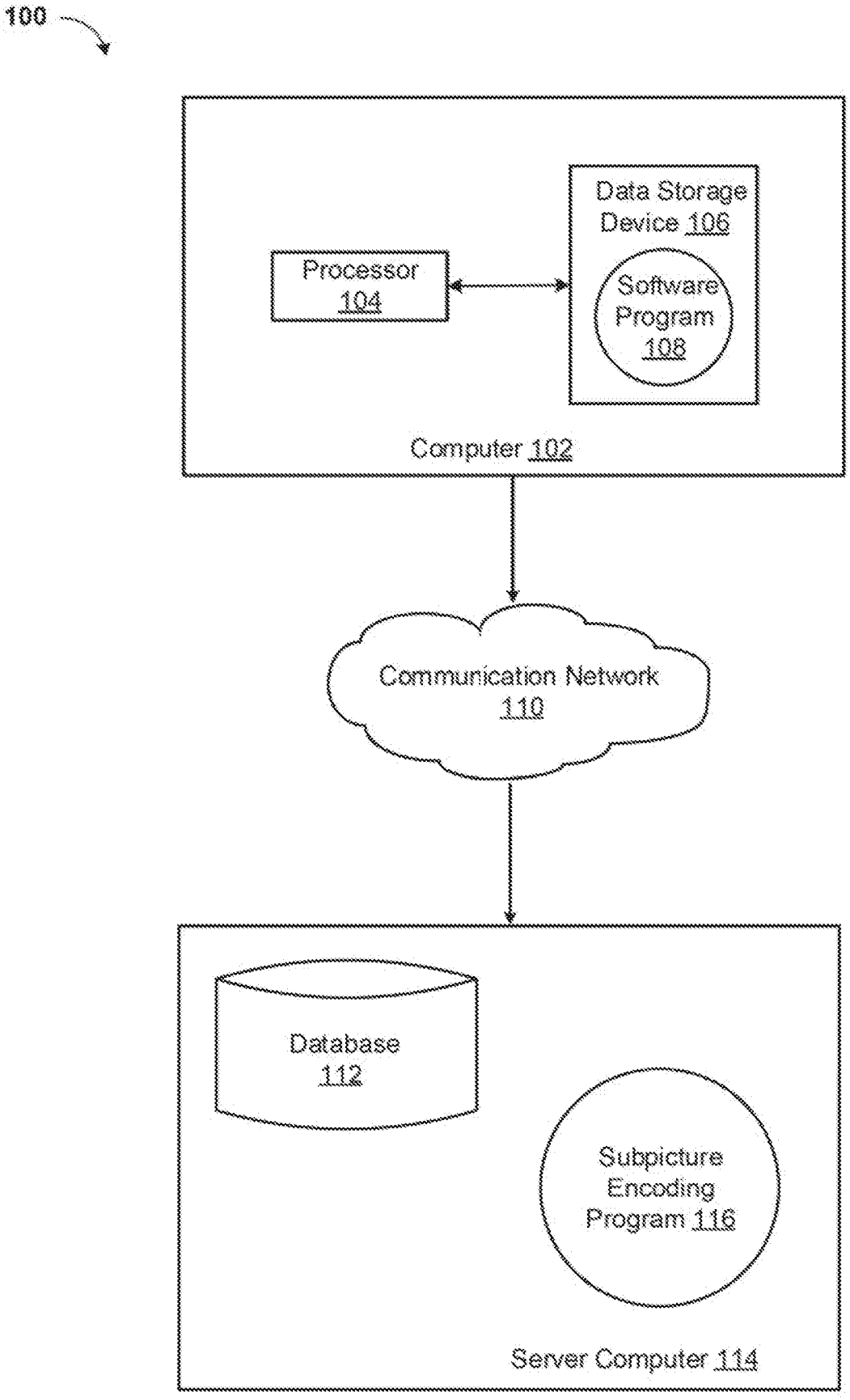
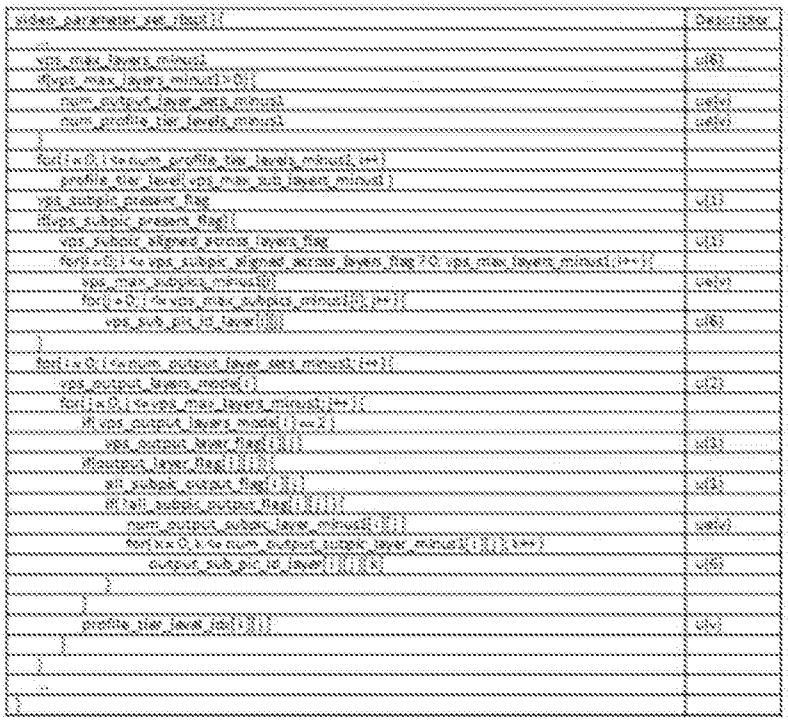

100
Data Storage Device 106
Processor 104
Software Program 108
Computer 102
Communication Network 110
Database 112
Subpicture Encoding Program 116
Server Computer 114

FIG. 1

200



video_parameter_set_rbsp( )	Descriptor
...	
vps_max_layers_minus1	u(6)
if(vps_max_layers_minus1 > 0) {	
num_output_layer_sets_minus1	ue(v)
num_profile_tier_levels_minus1	ue(v)
}	
for(i = 0; i <= num_profile_tier_levels_minus1; i++)	
profile_tier_level(vps_max_sub_layers_minus1)	
vps_subpic_present_flag	u(1)
if(vps_subpic_present_flag) {	
vps_subpic_aligned_across_layers_flag	u(1)
for(i = 0; i <= vps_subpic_aligned_across_layers_flag ? 0: vps_max_layers_minus1; i++) {	
vps_max_subpics_minus1[i]	ue(v)
for(j = 0; j <= vps_max_subpics_minus1[i]; j++)	
vps_sub_pic_id_len[i][j]	ue(v)
}	
for(i = 0; i <= num_output_layer_sets_minus1; i++)	
vps_output_layers_mode[i]	u(2)
for(i = 0; i <= vps_max_layers_minus1; i++)	
if(vps_output_layers_mode[i] == 2)	
vps_output_layer_flag[i][j]	u(1)
if(vps_output_layer_flag[i][j])	
vps_output_subpic_flag[i][j]	u(1)
if(vps_output_subpic_flag[i][j])	
num_output_subpics_minus1[i][j]	ue(v)
for(k = 0; k <= num_output_subpics_minus1[i][j]; k++)	
output_sub_pic_id_idx[i][j][k]	u(v)
}	
}	
profile_tier_level_idx[i][j]	u(v)
}	
}	
...	
}	

FIG. 2

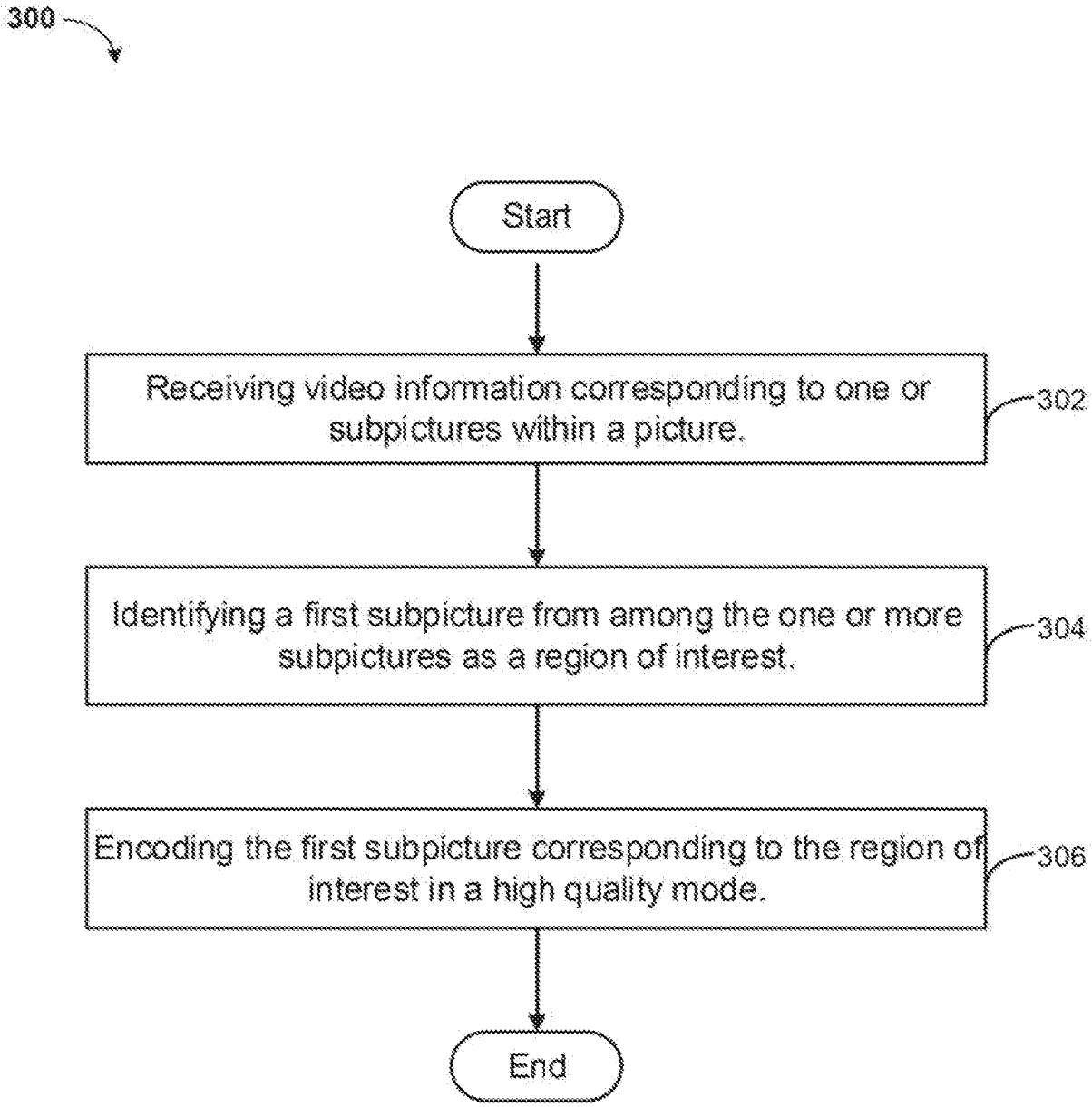

300
Start
Receiving video information corresponding to one or subpictures within a picture.
302
Identifying a first subpicture from among the one or more subpictures as a region of interest.
304
Encoding the first subpicture corresponding to the region of interest in a high quality mode.
306
End

FIG. 3

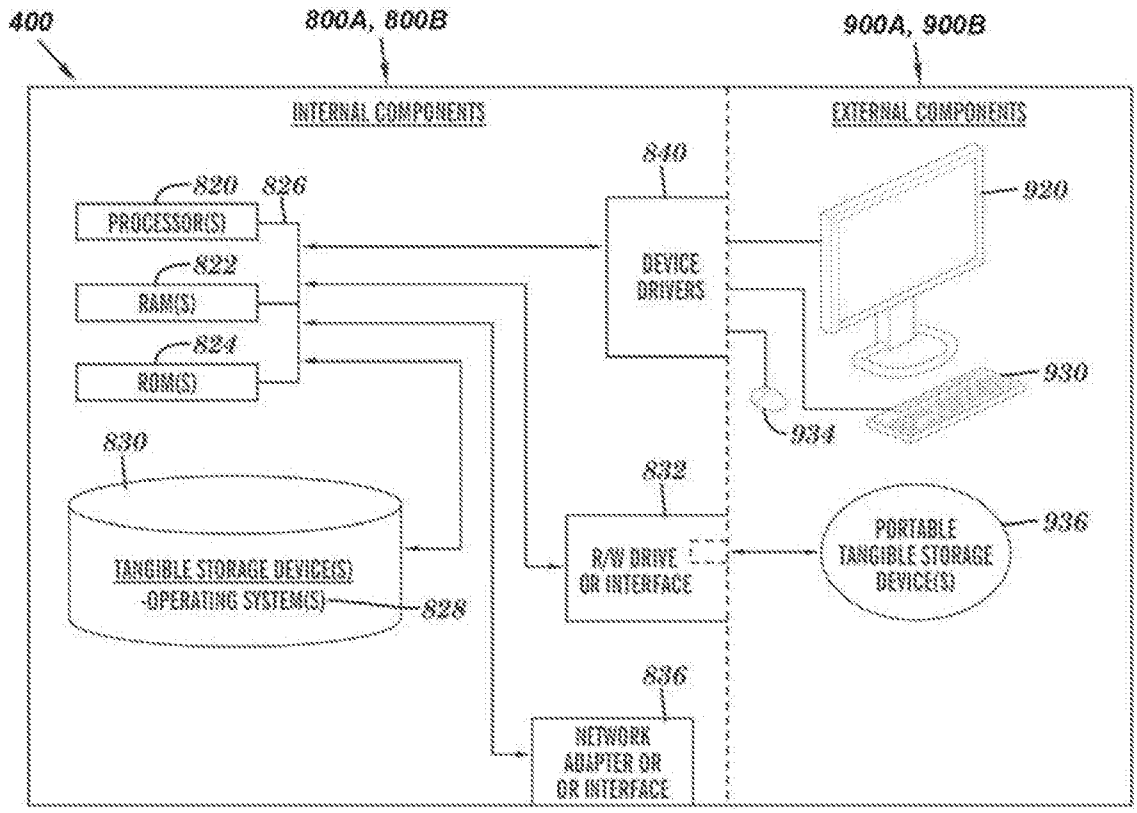

400
800A, 800B
900A, 900B
INTERNAL COMPONENTS
EXTERNAL COMPONENTS
820
826
PROCESSOR(S)
822
RAM(S)
824
ROM(S)
830
TANGIBLE STORAGE DEVICE(S)
OPERATING SYSTEM(S)
828
840
DEVICE DRIVERS
832
R/W DRIVE OR INTERFACE
836
NETWORK ADAPTER OR OR INTERFACE
920
930
934
936
PORTABLE TANGIBLE STORAGE DEVICE(S)

FIG. 4

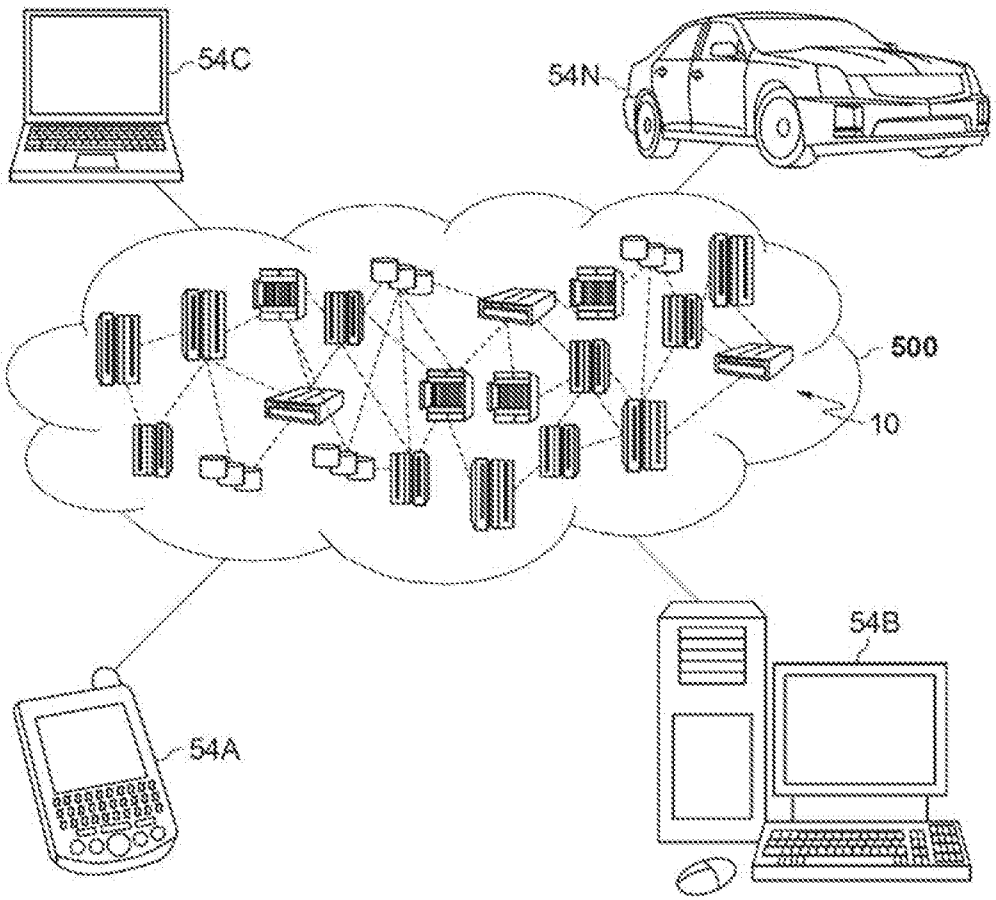


FIG. 5

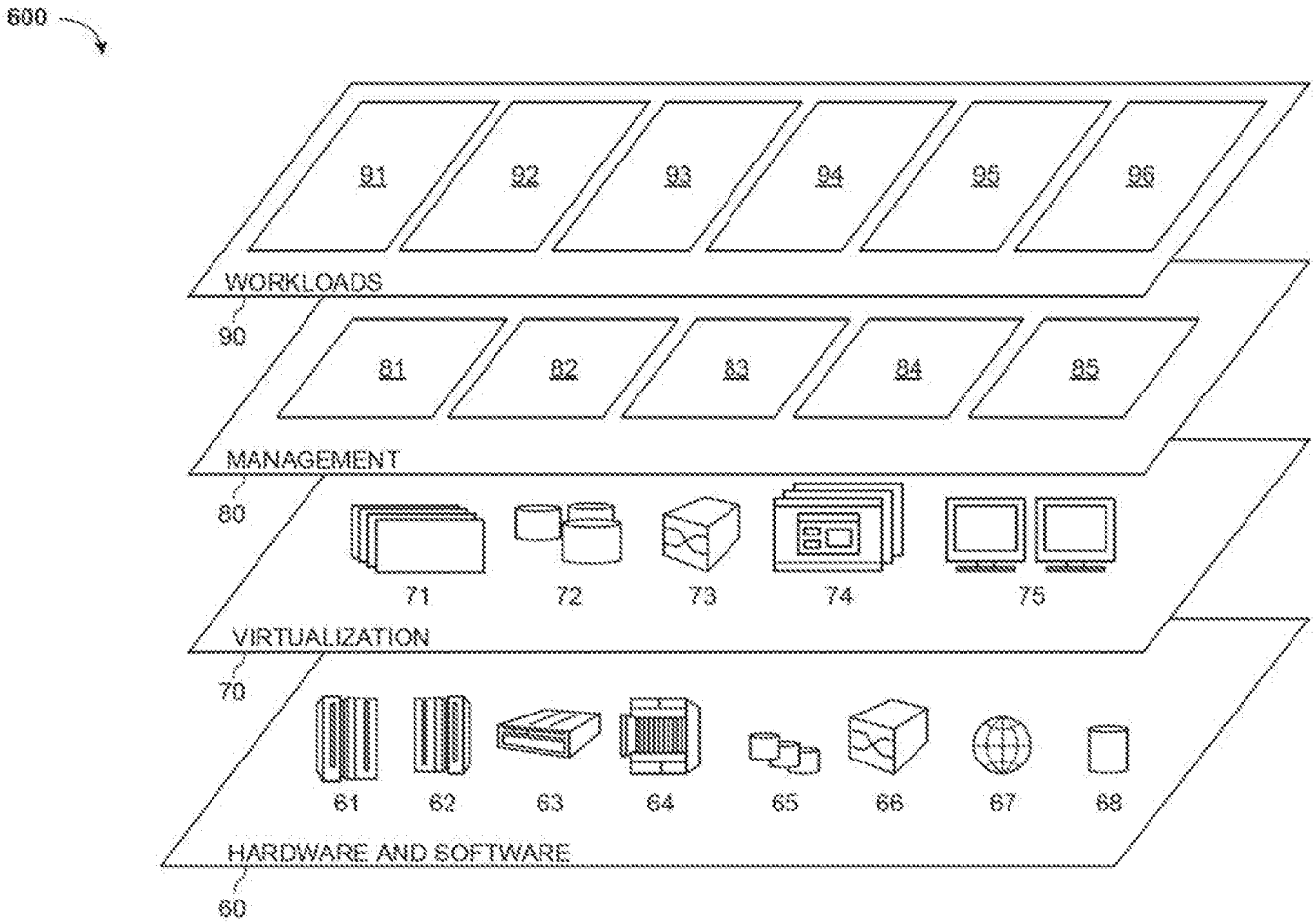

600
91
92
93
94
95
96
WORKLOADS
90
81
82
83
84
85
MANAGEMENT
80
71
72
73
74
75
VIRTUALIZATION
70
61
62
63
64
65
66
67
68
HARDWARE AND SOFTWARE
60

FIG. 6

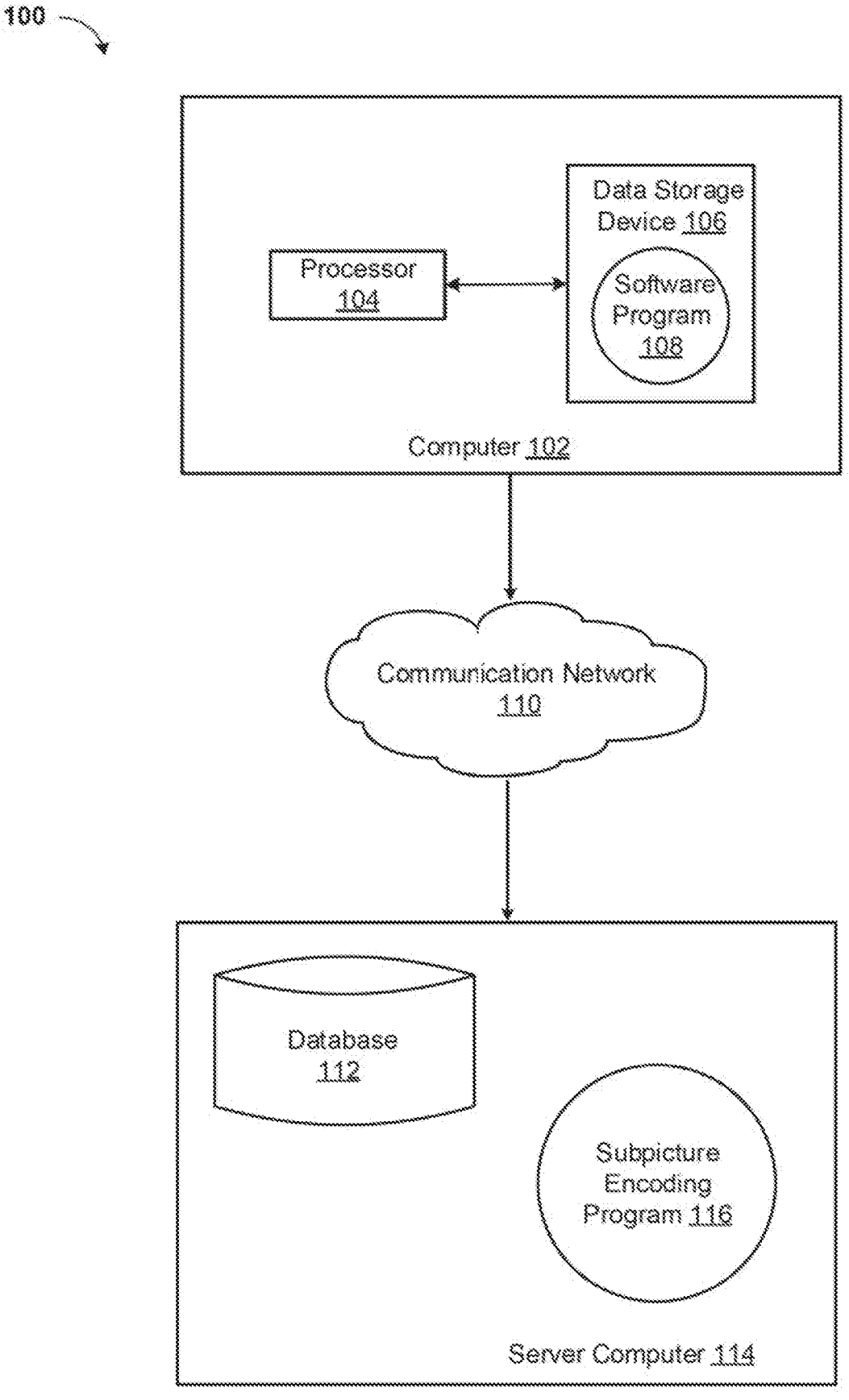

100
Data Storage Device 106
Processor 104
Software Program 108
Computer 102
Communication Network 110
Database 112
Subpicture Encoding Program 116
Server Computer 114

FIG. 1