

MEASUREMENT AND ADJUSTMENT OF EXTENDED REALITY ROUNDTRIP DELAYS

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of European Patent Application No. EP23305767.8, filed May 15, 2023, the content of which is hereby incorporated by reference herein.

BACKGROUND

[0002] Video coding systems may be used to compress digital video signals, for example, to reduce the storage and/or transmission bandwidth needed for such signals. Video coding systems may include, for example, wavelet-based systems, object-based systems, and/or block-based systems, such as a block-based hybrid video coding system. Extended reality (XR) devices may be provided to support various experiences. However, media capabilities of such XR devices may not be efficient.

SUMMARY

[0003] Systems, methods, and instrumentalities are disclosed for measuring and adjusting extended reality (XR) roundtrip delays. An example device may be configured to perform one or more actions. In an example, a device may determine a first time associated with a start of a scene graph update. The device may determine a second time associated with an end of the scene graph update and a start of a rendering process. The device may determine a third time associated with an end of the rendering process. The device may calculate a first latency associated with the scene graph update based on the first time and the second time. The device may calculate a second latency associated with the rendering process based on the second time and the third time. The device may adjust a first parameter associated with user experience quality based, at least in part, on the first latency.

[0004] The device may adjust a second parameter associated with user experience quality based, at least in part, on the second latency. The device may determine that the first latency is below a first delay threshold. The device may increase the quality and resource consumption of the scene graph update based on the determination that the first latency is below the first delay threshold. The device may determine that the first latency is above a second delay threshold. The device may decrease the quality

and resource consumption of the scene graph update based on the determination that the first latency is above the second delay threshold.

[0005] The device may determine that the second latency is below a third delay threshold. The device may increase the quality and resource consumption of the rendering process based on the determination that the second latency is below the third delay threshold. The device may determine that the second latency is above a fourth delay threshold. The device may decrease the quality and resource consumption of the rendering process based on the determination that the second latency is above the fourth delay threshold.

[0006] The device may calculate the first latency associated with the scene graph update based on the first time and the second time by determining a first time difference between the first time and the second time. The first latency may be equal to the first time difference. The device may calculate the second latency associated with the rendering process based on the second time and the third time by determining a second time difference between the second time and the third time. The second latency may be equal to the second time difference.

[0007] The first parameter associated with user experience quality may be associated with at least one of: interpolation between keyframes; a number of keyframes; a mesh associated with collision management; or physics computation.

[0008] The second parameter associated with user experience quality may be associated with at least one of: shadow complexity; ambient light complexity; a level of detail of a mesh; a level of detail of a material; or a level of detail of a texture.

[0009] The device may receive first information associated with an image. The device may perform the scene graph update based on the information associated with the image. The device may perform the scene graph update based on the first information associated with the image by outputting second information associated with the image. The device may perform the rendering process based on the second information associated with the image. The device may perform the rendering process based on the second information associated with the image by outputting a rendering of the image. The device may send the rendering of the image to a display.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] FIG. 1A is a system diagram illustrating an example communications system in which one or more disclosed embodiments may be implemented.

[0011] FIG. 1B is a system diagram illustrating an example wireless transmit/receive unit (WTRU) that may be used within the communications system illustrated in FIG. 1A according to an embodiment.

[0012] FIG. 1C is a system diagram illustrating an example radio access network (RAN) and an example core network (CN) that may be used within the communications system illustrated in FIG. 1A according to an embodiment.

[0013] FIG. 1D is a system diagram illustrating a further example RAN and a further example CN that may be used within the communications system illustrated in FIG. 1A according to an embodiment.

[0014] FIG. 2 is a diagram showing an example video encoder.

[0015] FIG. 3 is a diagram showing an example of a video decoder.

[0016] FIG. 4 is a diagram showing an example of a system in which various aspects and examples may be implemented.

[0017] FIG. 5 illustrates an example of a rendering loop for visual data.

[0018] FIG. 6 illustrates a flowchart of the observation points.

[0019] FIG. 7 illustrates an example rendering loop with observation points.

[0020] FIG. 8 illustrates an example call flow of the rendering loop with the observation points.

[0021] FIG. 9 is an example flowchart of processing step adjustment.

[0022] FIG. 10 illustrates an example of delay adjustment by switching between levels (e.g., image processing modes).

DETAILED DESCRIPTION

[0023] A detailed description of illustrative embodiments will now be described with reference to the various Figures. Although this description provides a detailed example of possible implementations, it should be noted that the details are intended to be exemplary and in no way limit the scope of the application.

[0024] FIG. 1A is a diagram illustrating an example communications system 100 in which one or more disclosed embodiments may be implemented. The communications system 100 may be a multiple access system that provides content, such as voice, data, video, messaging, broadcast, etc., to multiple wireless users. The communications system 100 may enable multiple wireless users to access such content through the sharing of system resources, including wireless bandwidth. For example, the communications systems 100 may employ one or more channel access methods, such as code division multiple access (CDMA), time division multiple access (TDMA), frequency division multiple access (FDMA), orthogonal FDMA (OFDMA), single-carrier FDMA (SC-FDMA), zero-tail unique-word DFT-Spread OFDM (ZT UW DTS-s OFDM), unique word OFDM (UW-OFDM), resource block-filtered OFDM, filter bank multicarrier (FBMC), and the like.

[0025] As shown in FIG. 1A, the communications system 100 may include wireless transmit/receive units (WTRUs) 102a, 102b, 102c, 102d, a RAN 104/113, a CN 106/115, a public switched telephone network (PSTN) 108, the Internet 110, and other networks 112, though it will be appreciated that the disclosed embodiments contemplate any number of WTRUs, base stations, networks, and/or network elements. Each of the WTRUs 102a, 102b, 102c, 102d may be any type of device configured to operate and/or communicate in a wireless environment. By way of example, the WTRUs 102a, 102b, 102c, 102d, any of which may be referred to as a "station" and/or a "STA", may be configured to transmit and/or receive wireless signals and may include a user equipment (UE), a mobile station, a fixed or mobile subscriber unit, a subscription-based unit, a pager, a cellular telephone, a personal digital assistant (PDA), a smartphone, a laptop, a netbook, a personal computer, a wireless sensor, a hotspot or Mi-Fi device, an Internet of Things (IoT) device, a watch or other wearable, a head-mounted display (HMD), a vehicle, a drone, a medical device and applications (e.g., remote surgery), an industrial device and applications (e.g., a robot and/or other wireless devices operating in an industrial and/or an automated processing chain contexts), a consumer electronics device, a device operating on commercial and/or industrial wireless networks, and the like. Any of the WTRUs 102a, 102b, 102c and 102d may be interchangeably referred to as a UE.

[0026] The communications systems 100 may also include a base station 114a and/or a base station 114b. Each of the base stations 114a, 114b may be any type of device configured to wirelessly interface with at least one of the WTRUs 102a, 102b, 102c, 102d to facilitate access to one or more communication networks, such as the CN 106/115, the Internet 110, and/or the other networks 112. By way of example, the base stations 114a, 114b may be a base transceiver station (BTS), a Node-B, an eNode B, a Home Node B, a Home eNode B, a gNB, a NR NodeB, a site controller, an access point (AP), a wireless router, and the like. While the base stations 114a, 114b are each depicted as a single element, it will be appreciated that the base stations 114a, 114b may include any number of interconnected base stations and/or network elements.

[0027] The base station 114a may be part of the RAN 104/113, which may also include other base stations and/or network elements (not shown), such as a base station controller (BSC), a radio network controller (RNC), relay nodes, etc. The base station 114a and/or the base station 114b may be configured to transmit and/or receive wireless signals on one or more carrier frequencies, which may be referred to as a cell (not shown). These frequencies may be in licensed spectrum, unlicensed spectrum, or a combination of licensed and unlicensed spectrum. A cell may provide coverage for a wireless service to a specific geographical area that may be relatively fixed or that may change over time. The cell may further be divided into cell sectors. For example, the cell associated with the base station 114a may be divided into three sectors. Thus, in one embodiment, the base station 114a may include three transceivers, i.e., one for

each sector of the cell. In an embodiment, the base station 114a may employ multiple-input multiple output (MIMO) technology and may utilize multiple transceivers for each sector of the cell. For example, beamforming may be used to transmit and/or receive signals in desired spatial directions.

[0028] The base stations 114a, 114b may communicate with one or more of the WTRUs 102a, 102b, 102c, 102d over an air interface 116, which may be any suitable wireless communication link (e.g., radio frequency (RF), microwave, centimeter wave, micrometer wave, infrared (IR), ultraviolet (UV), visible light, etc.). The air interface 116 may be established using any suitable radio access technology (RAT).

[0029] More specifically, as noted above, the communications system 100 may be a multiple access system and may employ one or more channel access schemes, such as CDMA, TDMA, FDMA, OFDMA, SC-FDMA, and the like. For example, the base station 114a in the RAN 104/113 and the WTRUs 102a, 102b, 102c may implement a radio technology such as Universal Mobile Telecommunications System (UMTS) Terrestrial Radio Access (UTRA), which may establish the air interface 115/116/117 using wideband CDMA (WCDMA). WCDMA may include communication protocols such as High-Speed Packet Access (HSPA) and/or Evolved HSPA (HSPA+). HSPA may include High-Speed Downlink (DL) Packet Access (HSDPA) and/or High-Speed UL Packet Access (HSUPA).

[0030] In an embodiment, the base station 114a and the WTRUs 102a, 102b, 102c may implement a radio technology such as Evolved UMTS Terrestrial Radio Access (E-UTRA), which may establish the air interface 116 using Long Term Evolution (LTE) and/or LTE-Advanced (LTE-A) and/or LTE-Advanced Pro (LTE-A Pro).

[0031] In an embodiment, the base station 114a and the WTRUs 102a, 102b, 102c may implement a radio technology such as NR Radio Access, which may establish the air interface 116 using New Radio (NR).

[0032] In an embodiment, the base station 114a and the WTRUs 102a, 102b, 102c may implement multiple radio access technologies. For example, the base station 114a and the WTRUs 102a, 102b, 102c may implement LTE radio access and NR radio access together, for instance using dual connectivity (DC) principles. Thus, the air interface utilized by WTRUs 102a, 102b, 102c may be characterized by multiple types of radio access technologies and/or transmissions sent to/from multiple types of base stations (e.g., an eNB and a gNB).

[0033] In other embodiments, the base station 114a and the WTRUs 102a, 102b, 102c may implement radio technologies such as IEEE 802.11 (i.e., Wireless Fidelity (WiFi)), IEEE 802.16 (i.e., Worldwide Interoperability for Microwave Access (WiMAX)), CDMA2000, CDMA2000 1X, CDMA2000 EV-DO, Interim Standard 2000 (IS-2000), Interim Standard 95 (IS-95), Interim Standard 856 (IS-856), Global System for

Mobile communications (GSM), Enhanced Data rates for GSM Evolution (EDGE), GSM EDGE (GERAN), and the like.

[0034] The base station 114b in FIG. 1A may be a wireless router, Home Node B, Home eNode B, or access point, for example, and may utilize any suitable RAT for facilitating wireless connectivity in a localized area, such as a place of business, a home, a vehicle, a campus, an industrial facility, an air corridor (e.g., for use by drones), a roadway, and the like. In one embodiment, the base station 114b and the WTRUs 102c, 102d may implement a radio technology such as IEEE 802.11 to establish a wireless local area network (WLAN). In an embodiment, the base station 114b and the WTRUs 102c, 102d may implement a radio technology such as IEEE 802.15 to establish a wireless personal area network (WPAN). In yet another embodiment, the base station 114b and the WTRUs 102c, 102d may utilize a cellular-based RAT (e.g., WCDMA, CDMA2000, GSM, LTE, LTE-A, LTE-A Pro, NR etc.) to establish a picocell or femtocell. As shown in FIG. 1A, the base station 114b may have a direct connection to the Internet 110. Thus, the base station 114b may not be required to access the Internet 110 via the CN 106/115.

[0035] The RAN 104/113 may be in communication with the CN 106/115, which may be any type of network configured to provide voice, data, applications, and/or voice over internet protocol (VoIP) services to one or more of the WTRUs 102a, 102b, 102c, 102d. The data may have varying quality of service (QoS) requirements, such as differing throughput requirements, latency requirements, error tolerance requirements, reliability requirements, data throughput requirements, mobility requirements, and the like. The CN 106/115 may provide call control, billing services, mobile location-based services, pre-paid calling, Internet connectivity, video distribution, etc., and/or perform high-level security functions, such as user authentication. Although not shown in FIG. 1A, it will be appreciated that the RAN 104/113 and/or the CN 106/115 may be in direct or indirect communication with other RANs that employ the same RAT as the RAN 104/113 or a different RAT. For example, in addition to being connected to the RAN 104/113, which may be utilizing a NR radio technology, the CN 106/115 may also be in communication with another RAN (not shown) employing a GSM, UMTS, CDMA 2000, WiMAX, E-UTRA, or WiFi radio technology.

[0036] The CN 106/115 may also serve as a gateway for the WTRUs 102a, 102b, 102c, 102d to access the PSTN 108, the Internet 110, and/or the other networks 112. The PSTN 108 may include circuit-switched telephone networks that provide plain old telephone service (POTS). The Internet 110 may include a global system of interconnected computer networks and devices that use common communication protocols, such as the transmission control protocol (TCP), user datagram protocol (UDP) and/or the internet protocol (IP) in the TCP/IP internet protocol suite. The networks 112 may include wired and/or wireless communications networks owned and/or operated by other service providers. For example,

the networks 112 may include another CN connected to one or more RANs, which may employ the same RAT as the RAN 104/113 or a different RAT.

[0037] Some or all of the WTRUs 102a, 102b, 102c, 102d in the communications system 100 may include multi-mode capabilities (e.g., the WTRUs 102a, 102b, 102c, 102d may include multiple transceivers for communicating with different wireless networks over different wireless links). For example, the WTRU 102c shown in FIG. 1A may be configured to communicate with the base station 114a, which may employ a cellular-based radio technology, and with the base station 114b, which may employ an IEEE 802 radio technology.

[0038] FIG. 1B is a system diagram illustrating an example WTRU 102. As shown in FIG. 1B, the WTRU 102 may include a processor 118, a transceiver 120, a transmit/receive element 122, a speaker/microphone 124, a keypad 126, a display/touchpad 128, non-removable memory 130, removable memory 132, a power source 134, a global positioning system (GPS) chipset 136, and/or other peripherals 138, among others. It will be appreciated that the WTRU 102 may include any sub-combination of the foregoing elements while remaining consistent with an embodiment.

[0039] The processor 118 may be a general purpose processor, a special purpose processor, a conventional processor, a digital signal processor (DSP), a plurality of microprocessors, one or more microprocessors in association with a DSP core, a controller, a microcontroller, Application Specific Integrated Circuits (ASICs), Field Programmable Gate Arrays (FPGAs) circuits, any other type of integrated circuit (IC), a state machine, and the like. The processor 118 may perform signal coding, data processing, power control, input/output processing, and/or any other functionality that enables the WTRU 102 to operate in a wireless environment. The processor 118 may be coupled to the transceiver 120, which may be coupled to the transmit/receive element 122. While FIG. 1B depicts the processor 118 and the transceiver 120 as separate components, it will be appreciated that the processor 118 and the transceiver 120 may be integrated together in an electronic package or chip.

[0040] The transmit/receive element 122 may be configured to transmit signals to, or receive signals from, a base station (e.g., the base station 114a) over the air interface 116. For example, in one embodiment, the transmit/receive element 122 may be an antenna configured to transmit and/or receive RF signals. In an embodiment, the transmit/receive element 122 may be an emitter/detector configured to transmit and/or receive IR, UV, or visible light signals, for example. In yet another embodiment, the transmit/receive element 122 may be configured to transmit and/or receive both RF and light signals. It will be appreciated that the transmit/receive element 122 may be configured to transmit and/or receive any combination of wireless signals.

[0041] Although the transmit/receive element 122 is depicted in FIG. 1B as a single element, the WTRU 102 may include any number of transmit/receive elements 122. More specifically, the WTRU 102 may employ MIMO technology. Thus, in one embodiment, the WTRU 102 may include two or more transmit/receive elements 122 (e.g., multiple antennas) for transmitting and receiving wireless signals over the air interface 116.

[0042] The transceiver 120 may be configured to modulate the signals that are to be transmitted by the transmit/receive element 122 and to demodulate the signals that are received by the transmit/receive element 122. As noted above, the WTRU 102 may have multi-mode capabilities. Thus, the transceiver 120 may include multiple transceivers for enabling the WTRU 102 to communicate via multiple RATs, such as NR and IEEE 802.11, for example.

[0043] The processor 118 of the WTRU 102 may be coupled to, and may receive user input data from, the speaker/microphone 124, the keypad 126, and/or the display/touchpad 128 (e.g., a liquid crystal display (LCD) display unit or organic light-emitting diode (OLED) display unit). The processor 118 may also output user data to the speaker/microphone 124, the keypad 126, and/or the display/touchpad 128. In addition, the processor 118 may access information from, and store data in, any type of suitable memory, such as the non-removable memory 130 and/or the removable memory 132. The non-removable memory 130 may include random-access memory (RAM), read-only memory (ROM), a hard disk, or any other type of memory storage device. The removable memory 132 may include a subscriber identity module (SIM) card, a memory stick, a secure digital (SD) memory card, and the like. In other embodiments, the processor 118 may access information from, and store data in, memory that is not physically located on the WTRU 102, such as on a server or a home computer (not shown).

[0044] The processor 118 may receive power from the power source 134, and may be configured to distribute and/or control the power to the other components in the WTRU 102. The power source 134 may be any suitable device for powering the WTRU 102. For example, the power source 134 may include one or more dry cell batteries (e.g., nickel-cadmium (NiCd), nickel-zinc (NiZn), nickel metal hydride (NiMH), lithium-ion (Li-ion), etc.), solar cells, fuel cells, and the like.

[0045] The processor 118 may also be coupled to the GPS chipset 136, which may be configured to provide location information (e.g., longitude and latitude) regarding the current location of the WTRU 102. In addition to, or in lieu of, the information from the GPS chipset 136, the WTRU 102 may receive location information over the air interface 116 from a base station (e.g., base stations 114a, 114b) and/or determine its location based on the timing of the signals being received from two or more nearby base stations. It will be appreciated that the WTRU 102 may acquire location information by way of any suitable location-determination method while remaining consistent with an embodiment.

[0046] The processor 118 may further be coupled to other peripherals 138, which may include one or more software and/or hardware modules that provide additional features, functionality and/or wired or wireless connectivity. For example, the peripherals 138 may include an accelerometer, an e-compass, a satellite transceiver, a digital camera (for photographs and/or video), a universal serial bus (USB) port, a vibration device, a television transceiver, a hands free headset, a Bluetooth® module, a frequency modulated (FM) radio unit, a digital music player, a media player, a video game player module, an Internet browser, a Virtual Reality and/or Augmented Reality (VR/AR) device, an activity tracker, and the like. The peripherals 138 may include one or more sensors, the sensors may be one or more of a gyroscope, an accelerometer, a hall effect sensor, a magnetometer, an orientation sensor, a proximity sensor, a temperature sensor, a time sensor; a geolocation sensor; an altimeter, a light sensor, a touch sensor, a magnetometer, a barometer, a gesture sensor, a biometric sensor, and/or a humidity sensor.

[0047] The WTRU 102 may include a full duplex radio for which transmission and reception of some or all of the signals (e.g., associated with particular subframes for both the UL (e.g., for transmission) and downlink (e.g., for reception) may be concurrent and/or simultaneous. The full duplex radio may include an interference management unit to reduce and or substantially eliminate self-interference via either hardware (e.g., a choke) or signal processing via a processor (e.g., a separate processor (not shown) or via processor 118). In an embodiment, the WTRU 102 may include a half-duplex radio for which transmission and reception of some or all of the signals (e.g., associated with particular subframes for either the UL (e.g., for transmission) or the downlink (e.g., for reception)).

[0048] FIG. 1C is a system diagram illustrating the RAN 104 and the CN 106 according to an embodiment. As noted above, the RAN 104 may employ an E-UTRA radio technology to communicate with the WTRUs 102a, 102b, 102c over the air interface 116. The RAN 104 may also be in communication with the CN 106.

[0049] The RAN 104 may include eNode-Bs 160a, 160b, 160c, though it will be appreciated that the RAN 104 may include any number of eNode-Bs while remaining consistent with an embodiment. The eNode-Bs 160a, 160b, 160c may each include one or more transceivers for communicating with the WTRUs 102a, 102b, 102c over the air interface 116. In one embodiment, the eNode-Bs 160a, 160b, 160c may implement MIMO technology. Thus, the eNode-B 160a, for example, may use multiple antennas to transmit wireless signals to, and/or receive wireless signals from, the WTRU 102a.

[0050] Each of the eNode-Bs 160a, 160b, 160c may be associated with a particular cell (not shown) and may be configured to handle radio resource management decisions, handover decisions, scheduling of users in the UL and/or DL, and the like. As shown in FIG. 1C, the eNode-Bs 160a, 160b, 160c may communicate with one another over an X2 interface.

[0051] The CN 106 shown in FIG. 1C may include a mobility management entity (MME) 162, a serving gateway (SGW) 164, and a packet data network (PDN) gateway (or PGW) 166. While each of the foregoing elements is depicted as part of the CN 106, it will be appreciated that any of these elements may be owned and/or operated by an entity other than the CN operator.

[0052] The MME 162 may be connected to each of the eNode-Bs 162a, 162b, 162c in the RAN 104 via an S1 interface and may serve as a control node. For example, the MME 162 may be responsible for authenticating users of the WTRUs 102a, 102b, 102c, bearer activation/deactivation, selecting a particular serving gateway during an initial attach of the WTRUs 102a, 102b, 102c, and the like. The MME 162 may provide a control plane function for switching between the RAN 104 and other RANs (not shown) that employ other radio technologies, such as GSM and/or WCDMA.

[0053] The SGW 164 may be connected to each of the eNode Bs 160a, 160b, 160c in the RAN 104 via the S1 interface. The SGW 164 may generally route and forward user data packets to/from the WTRUs 102a, 102b, 102c. The SGW 164 may perform other functions, such as anchoring user planes during inter-eNode B handovers, triggering paging when DL data is available for the WTRUs 102a, 102b, 102c, managing and storing contexts of the WTRUs 102a, 102b, 102c, and the like.

[0054] The SGW 164 may be connected to the PGW 166, which may provide the WTRUs 102a, 102b, 102c with access to packet-switched networks, such as the Internet 110, to facilitate communications between the WTRUs 102a, 102b, 102c and IP-enabled devices.

[0055] The CN 106 may facilitate communications with other networks. For example, the CN 106 may provide the WTRUs 102a, 102b, 102c with access to circuit-switched networks, such as the PSTN 108, to facilitate communications between the WTRUs 102a, 102b, 102c and traditional land-line communications devices. For example, the CN 106 may include, or may communicate with, an IP gateway (e.g., an IP multimedia subsystem (IMS) server) that serves as an interface between the CN 106 and the PSTN 108. In addition, the CN 106 may provide the WTRUs 102a, 102b, 102c with access to the other networks 112, which may include other wired and/or wireless networks that are owned and/or operated by other service providers.

[0056] Although the WTRU is described in FIGS. 1A-1D as a wireless terminal, it is contemplated that in certain representative embodiments that such a terminal may use (e.g., temporarily or permanently) wired communication interfaces with the communication network.

[0057] In representative embodiments, the other network 112 may be a WLAN.

[0058] A WLAN in Infrastructure Basic Service Set (BSS) mode may have an Access Point (AP) for the BSS and one or more stations (STAs) associated with the AP. The AP may have an access or an interface to a Distribution System (DS) or another type of wired/wireless network that carries traffic in to and/or out of

the BSS. Traffic to STAs that originates from outside the BSS may arrive through the AP and may be delivered to the STAs. Traffic originating from STAs to destinations outside the BSS may be sent to the AP to be delivered to respective destinations. Traffic between STAs within the BSS may be sent through the AP, for example, where the source STA may send traffic to the AP and the AP may deliver the traffic to the destination STA. The traffic between STAs within a BSS may be considered and/or referred to as peer-to-peer traffic. The peer-to-peer traffic may be sent between (e.g., directly between) the source and destination STAs with a direct link setup (DLS). In certain representative embodiments, the DLS may use an 802.11e DLS or an 802.11z tunneled DLS (TDLS). A WLAN using an Independent BSS (IBSS) mode may not have an AP, and the STAs (e.g., all of the STAs) within or using the IBSS may communicate directly with each other. The IBSS mode of communication may sometimes be referred to herein as an "ad-hoc" mode of communication.

[0059] When using the 802.11ac infrastructure mode of operation or a similar mode of operations, the AP may transmit a beacon on a fixed channel, such as a primary channel. The primary channel may be a fixed width (e.g., 20 MHz wide bandwidth) or a dynamically set width via signaling. The primary channel may be the operating channel of the BSS and may be used by the STAs to establish a connection with the AP. In certain representative embodiments, Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) may be implemented, for example in 802.11 systems. For CSMA/CA, the STAs (e.g., every STA), including the AP, may sense the primary channel. If the primary channel is sensed/detected and/or determined to be busy by a particular STA, the particular STA may back off. One STA (e.g., only one station) may transmit at any given time in a given BSS.

[0060] High Throughput (HT) STAs may use a 40 MHz wide channel for communication, for example, via a combination of the primary 20 MHz channel with an adjacent or nonadjacent 20 MHz channel to form a 40 MHz wide channel.

[0061] Very High Throughput (VHT) STAs may support 20MHz, 40 MHz, 80 MHz, and/or 160 MHz wide channels. The 40 MHz, and/or 80 MHz, channels may be formed by combining contiguous 20 MHz channels. A 160 MHz channel may be formed by combining 8 contiguous 20 MHz channels, or by combining two non-contiguous 80 MHz channels, which may be referred to as an 80+80 configuration. For the 80+80 configuration, the data, after channel encoding, may be passed through a segment parser that may divide the data into two streams. Inverse Fast Fourier Transform (IFFT) processing, and time domain processing, may be done on each stream separately. The streams may be mapped on to the two 80 MHz channels, and the data may be transmitted by a transmitting STA. At the receiver of the receiving STA, the above described operation for the 80+80 configuration may be reversed, and the combined data may be sent to the Medium Access Control (MAC).

[0062] Sub 1 GHz modes of operation are supported by 802.11af and 802.11ah. The channel operating bandwidths, and carriers, are reduced in 802.11af and 802.11ah relative to those used in 802.11n, and 802.11ac. 802.11af supports 5 MHz, 10 MHz and 20 MHz bandwidths in the TV White Space (TVWS) spectrum, and 802.11ah supports 1 MHz, 2 MHz, 4 MHz, 8 MHz, and 16 MHz bandwidths using non-TVWS spectrum. According to a representative embodiment, 802.11ah may support Meter Type Control/Machine-Type Communications, such as MTC devices in a macro coverage area. MTC devices may have certain capabilities, for example, limited capabilities including support for (e.g., only support for) certain and/or limited bandwidths. The MTC devices may include a battery with a battery life above a threshold (e.g., to maintain a very long battery life).

[0063] WLAN systems, which may support multiple channels, and channel bandwidths, such as 802.11n, 802.11ac, 802.11af, and 802.11ah, include a channel which may be designated as the primary channel. The primary channel may have a bandwidth equal to the largest common operating bandwidth supported by all STAs in the BSS. The bandwidth of the primary channel may be set and/or limited by a STA, from among all STAs in operating in a BSS, which supports the smallest bandwidth operating mode. In the example of 802.11ah, the primary channel may be 1 MHz wide for STAs (e.g., MTC type devices) that support (e.g., only support) a 1 MHz mode, even if the AP, and other STAs in the BSS support 2 MHz, 4 MHz, 8 MHz, 16 MHz, and/or other channel bandwidth operating modes. Carrier sensing and/or Network Allocation Vector (NAV) settings may depend on the status of the primary channel. If the primary channel is busy, for example, due to a STA (which supports only a 1 MHz operating mode), transmitting to the AP, the entire available frequency bands may be considered busy even though a majority of the frequency bands remains idle and may be available.

[0064] In the United States, the available frequency bands, which may be used by 802.11ah, are from 902 MHz to 928 MHz. In Korea, the available frequency bands are from 917.5 MHz to 923.5 MHz. In Japan, the available frequency bands are from 916.5 MHz to 927.5 MHz. The total bandwidth available for 802.11ah is 6 MHz to 26 MHz depending on the country code.

[0065] FIG. 1D is a system diagram illustrating the RAN 113 and the CN 115 according to an embodiment. As noted above, the RAN 113 may employ an NR radio technology to communicate with the WTRUs 102a, 102b, 102c over the air interface 116. The RAN 113 may also be in communication with the CN 115.

[0066] The RAN 113 may include gNBs 180a, 180b, 180c, though it will be appreciated that the RAN 113 may include any number of gNBs while remaining consistent with an embodiment. The gNBs 180a, 180b, 180c may each include one or more transceivers for communicating with the WTRUs 102a, 102b, 102c over the air interface 116. In one embodiment, the gNBs 180a, 180b, 180c may implement MIMO

technology. For example, gNBs 180a, 180b may utilize beamforming to transmit signals to and/or receive signals from the gNBs 180a, 180b, 180c. Thus, the gNB 180a, for example, may use multiple antennas to transmit wireless signals to, and/or receive wireless signals from, the WTRU 102a. In an embodiment, the gNBs 180a, 180b, 180c may implement carrier aggregation technology. For example, the gNB 180a may transmit multiple component carriers to the WTRU 102a (not shown). A subset of these component carriers may be on unlicensed spectrum while the remaining component carriers may be on licensed spectrum. In an embodiment, the gNBs 180a, 180b, 180c may implement Coordinated Multi-Point (CoMP) technology. For example, WTRU 102a may receive coordinated transmissions from gNB 180a and gNB 180b (and/or gNB 180c).

[0067] The WTRUs 102a, 102b, 102c may communicate with gNBs 180a, 180b, 180c using transmissions associated with a scalable numerology. For example, the OFDM symbol spacing and/or OFDM subcarrier spacing may vary for different transmissions, different cells, and/or different portions of the wireless transmission spectrum. The WTRUs 102a, 102b, 102c may communicate with gNBs 180a, 180b, 180c using subframe or transmission time intervals (TTIs) of various or scalable lengths (e.g., including varying number of OFDM symbols and/or lasting varying lengths of absolute time).

[0068] The gNBs 180a, 180b, 180c may be configured to communicate with the WTRUs 102a, 102b, 102c in a standalone configuration and/or a non-standalone configuration. In the standalone configuration, WTRUs 102a, 102b, 102c may communicate with gNBs 180a, 180b, 180c without also accessing other RANs (e.g., such as eNode-Bs 160a, 160b, 160c). In the standalone configuration, WTRUs 102a, 102b, 102c may utilize one or more of gNBs 180a, 180b, 180c as a mobility anchor point. In the standalone configuration, WTRUs 102a, 102b, 102c may communicate with gNBs 180a, 180b, 180c using signals in an unlicensed band. In a non-standalone configuration WTRUs 102a, 102b, 102c may communicate with/connect to gNBs 180a, 180b, 180c while also communicating with/connecting to another RAN such as eNode-Bs 160a, 160b, 160c. For example, WTRUs 102a, 102b, 102c may implement DC principles to communicate with one or more gNBs 180a, 180b, 180c and one or more eNode-Bs 160a, 160b, 160c substantially simultaneously. In the non-standalone configuration, eNode-Bs 160a, 160b, 160c may serve as a mobility anchor for WTRUs 102a, 102b, 102c and gNBs 180a, 180b, 180c may provide additional coverage and/or throughput for servicing WTRUs 102a, 102b, 102c.

[0069] Each of the gNBs 180a, 180b, 180c may be associated with a particular cell (not shown) and may be configured to handle radio resource management decisions, handover decisions, scheduling of users in the UL and/or DL, support of network slicing, dual connectivity, interworking between NR and E-UTRA, routing of user plane data towards User Plane Function (UPF) 184a, 184b, routing of control plane

information towards Access and Mobility Management Function (AMF) 182a, 182b and the like. As shown in FIG. 1D, the gNBs 180a, 180b, 180c may communicate with one another over an Xn interface.

[0070] The CN 115 shown in FIG. 1D may include at least one AMF 182a, 182b, at least one UPF 184a, 184b, at least one Session Management Function (SMF) 183a, 183b, and possibly a Data Network (DN) 185a, 185b. While each of the foregoing elements are depicted as part of the CN 115, it will be appreciated that any of these elements may be owned and/or operated by an entity other than the CN operator.

[0071] The AMF 182a, 182b may be connected to one or more of the gNBs 180a, 180b, 180c in the RAN 113 via an N2 interface and may serve as a control node. For example, the AMF 182a, 182b may be responsible for authenticating users of the WTRUs 102a, 102b, 102c, support for network slicing (e.g., handling of different PDU sessions with different requirements), selecting a particular SMF 183a, 183b, management of the registration area, termination of NAS signaling, mobility management, and the like. Network slicing may be used by the AMF 182a, 182b in order to customize CN support for WTRUs 102a, 102b, 102c based on the types of services being utilized WTRUs 102a, 102b, 102c. For example, different network slices may be established for different use cases such as services relying on ultra-reliable low latency (URLLC) access, services relying on enhanced massive mobile broadband (eMBB) access, services for machine type communication (MTC) access, and/or the like. The AMF 162 may provide a control plane function for switching between the RAN 113 and other RANs (not shown) that employ other radio technologies, such as LTE, LTE-A, LTE-A Pro, and/or non-3GPP access technologies such as WiFi.

[0072] The SMF 183a, 183b may be connected to an AMF 182a, 182b in the CN 115 via an N11 interface. The SMF 183a, 183b may also be connected to a UPF 184a, 184b in the CN 115 via an N4 interface. The SMF 183a, 183b may select and control the UPF 184a, 184b and configure the routing of traffic through the UPF 184a, 184b. The SMF 183a, 183b may perform other functions, such as managing and allocating UE IP address, managing PDU sessions, controlling policy enforcement and QoS, providing downlink data notifications, and the like. A PDU session type may be IP-based, non-IP based, Ethernet-based, and the like.

[0073] The UPF 184a, 184b may be connected to one or more of the gNBs 180a, 180b, 180c in the RAN 113 via an N3 interface, which may provide the WTRUs 102a, 102b, 102c with access to packet-switched networks, such as the Internet 110, to facilitate communications between the WTRUs 102a, 102b, 102c and IP-enabled devices. The UPF 184a, 184b may perform other functions, such as routing and forwarding packets, enforcing user plane policies, supporting multi-homed PDU sessions, handling user plane QoS, buffering downlink packets, providing mobility anchoring, and the like.

[0074] The CN 115 may facilitate communications with other networks. For example, the CN 115 may include, or may communicate with, an IP gateway (e.g., an IP multimedia subsystem (IMS) server) that serves as an interface between the CN 115 and the PSTN 108. In addition, the CN 115 may provide the WTRUs 102a, 102b, 102c with access to the other networks 112, which may include other wired and/or wireless networks that are owned and/or operated by other service providers. In one embodiment, the WTRUs 102a, 102b, 102c may be connected to a local Data Network (DN) 185a, 185b through the UPF 184a, 184b via the N3 interface to the UPF 184a, 184b, and an N6 interface between the UPF 184a, 184b and the DN 185a, 185b.

[0075] In view of Figures 1A-1D, and the corresponding description of Figures 1A-1D, one or more, or all, of the functions described herein with regard to one or more of: WTRU 102a-d, Base Station 114a-b, eNode-B 160a-c, MME 162, SGW 164, PGW 166, gNB 180a-c, AMF 182a-b, UPF 184a-b, SMF 183a-b, DN 185a-b, and/or any other device(s) described herein, may be performed by one or more emulation devices (not shown). The emulation devices may be one or more devices configured to emulate one or more, or all, of the functions described herein. For example, the emulation devices may be used to test other devices and/or to simulate network and/or WTRU functions.

[0076] The emulation devices may be designed to implement one or more tests of other devices in a lab environment and/or in an operator network environment. For example, the one or more emulation devices may perform the one or more, or all, functions while being fully or partially implemented and/or deployed as part of a wired and/or wireless communication network in order to test other devices within the communication network. The one or more emulation devices may perform the one or more, or all, functions while being temporarily implemented/deployed as part of a wired and/or wireless communication network. The emulation device may be directly coupled to another device for purposes of testing and/or may performing testing using over-the-air wireless communications.

[0077] The one or more emulation devices may perform the one or more, including all, functions while not being implemented/deployed as part of a wired and/or wireless communication network. For example, the emulation devices may be utilized in a testing scenario in a testing laboratory and/or a non-deployed (e.g., testing) wired and/or wireless communication network in order to implement testing of one or more components. The one or more emulation devices may be test equipment. Direct RF coupling and/or wireless communications via RF circuitry (e.g., which may include one or more antennas) may be used by the emulation devices to transmit and/or receive data.

[0078] This application describes a variety of aspects, including tools, features, examples or embodiments, models, approaches, etc. Many of these aspects are described with specificity and, at least to show the individual characteristics, are often described in a manner that may sound limiting. However,

this is for purposes of clarity in description, and does not limit the application or scope of those aspects. Indeed, all of the different aspects may be combined and interchanged to provide further aspects. Moreover, the aspects may be combined and interchanged with aspects described in earlier filings as well.

[0079] The aspects described and contemplated in this application may be implemented in many different forms. FIGS. 5-10 described herein may provide some embodiments, but other embodiments are contemplated. The discussion of FIGS. 5-10 does not limit the breadth of the implementations. At least one of the aspects generally relates to video encoding and decoding, and at least one other aspect generally relates to transmitting a bitstream generated or encoded. These and other aspects may be implemented as a method, an apparatus, a computer readable storage medium having stored thereon instructions for encoding or decoding video data according to any of the methods described, and/or a computer readable storage medium having stored thereon a bitstream generated according to any of the methods described.

[0080] In the present application, the terms "reconstructed" and "decoded" may be used interchangeably, the terms "pixel" and "sample" may be used interchangeably, the terms "image," "picture" and "frame" may be used interchangeably.

[0081] Various methods are described herein, and each of the methods comprises one or more steps or actions for achieving the described method. Unless a specific order of steps or actions is required for proper operation of the method, the order and/or use of specific steps and/or actions may be modified or combined. Additionally, terms such as "first", "second", etc. may be used in various embodiments to modify an element, component, step, operation, etc., such as, for example, a "first decoding" and a "second decoding". Use of such terms does not imply an ordering to the modified operations unless specifically required. So, in this example, the first decoding need not be performed before the second decoding, and may occur, for example, before, during, or in an overlapping time period with the second decoding.

[0082] Various methods and other aspects described in this application may (for example, be used to) modify modules, for example, pre-encoding processing 201, intra prediction 260, entropy coding 245 and/or entropy decoding modules 330, intra prediction 360, post-decoding processing 385, of a video encoder 200 and a video decoder 300 as shown in FIG. 2 and FIG. 3 respectively. Moreover, the subject matter disclosed herein presents aspects that are not limited to VVC or HEVC, and may be applied, for example, to any type, format or version of video coding, whether described in a standard or a recommendation, whether pre-existing or future-developed, and extensions of any such standards and recommendations (e.g., including VVC and HEVC). Unless indicated otherwise, or technically precluded, the aspects described in this application may be used individually or in combination.

[0083] Various numeric values are used in examples described the present application, such as minimum and maximum value ranges (for example, 0 to 1, 0 to N or 0 to 255), bit values for indications or

determinations, default values, ID numbers (for example, for adaptation IDs), etc. These and other specific values are for purposes of describing examples and the aspects described are not limited to these specific values.

[0084] FIG. 2 is a diagram showing an example video encoder. Variations of example encoder 200 are contemplated, but the encoder 200 is described below for purposes of clarity without describing all expected variations.

[0085] Before being encoded, the video sequence may go through pre-encoding processing (201), for example, applying a color transform to the input color picture (e.g., conversion from RGB 4:4:4 to YCbCr 4:2:0), or performing a remapping of the input picture components in order to get a signal distribution more resilient to compression (for instance using a histogram equalization of one of the color components). Metadata may be associated with the pre-processing, and attached to the bitstream.

[0086] In the encoder 200, a picture is encoded by the encoder elements as described below. The picture to be encoded is partitioned (202) and processed in units of, for example, coding units (CUs). Each unit is encoded using, for example, either an intra or inter mode. When a unit is encoded in an intra mode, it performs intra prediction (260). In an inter mode, motion estimation (275) and compensation (270) are performed. The encoder decides (205) which one of the intra mode or inter mode to use for encoding the unit, and indicates the intra/inter decision by, for example, a prediction mode flag. Prediction residuals are calculated, for example, by subtracting (210) the predicted block from the original image block.

[0087] The prediction residuals are then transformed (225) and quantized (230). The quantized transform coefficients, as well as motion vectors and other syntax elements, are entropy coded (245) to output a bitstream. The encoder can skip the transform and apply quantization directly to the non-transformed residual signal. The encoder can bypass both transform and quantization, i.e., the residual is coded directly without the application of the transform or quantization processes.

[0088] The encoder decodes an encoded block to provide a reference for further predictions. The quantized transform coefficients are de-quantized (240) and inverse transformed (250) to decode prediction residuals. Combining (255) the decoded prediction residuals and the predicted block, an image block is reconstructed. In-loop filters (265) are applied to the reconstructed picture to perform, for example, deblocking/SAO (Sample Adaptive Offset) filtering to reduce encoding artifacts. The filtered image is stored at a reference picture buffer (280).

[0089] FIG. 3 is a diagram showing an example of a video decoder. In example decoder 300, a bitstream is decoded by the decoder elements as described below. Video decoder 300 generally performs a decoding pass reciprocal to the encoding pass as described in FIG. 2. The encoder 200 may also generally perform video decoding as part of encoding video data. For example, the encoder 200 may

perform one or more of the video decoding steps presented herein. The encoder reconstructs the decoded images, for example, to maintain synchronization with the decoder with respect to one or more of the following: reference pictures, entropy coding contexts, and other decoder-relevant state variables.

[0090] In particular, the input of the decoder includes a video bitstream, which may be generated by video encoder 200. The bitstream is first entropy decoded (330) to obtain transform coefficients, motion vectors, and other coded information. The picture partition information indicates how the picture is partitioned. The decoder may therefore divide (335) the picture according to the decoded picture partitioning information. The transform coefficients are de-quantized (340) and inverse transformed (350) to decode the prediction residuals. Combining (355) the decoded prediction residuals and the predicted block, an image block is reconstructed. The predicted block may be obtained (370) from intra prediction (360) or motion-compensated prediction (i.e., inter prediction) (375). In-loop filters (365) are applied to the reconstructed image. The filtered image is stored at a reference picture buffer (380).

[0091] The decoded picture can further go through post-decoding processing (385), for example, an inverse color transform (e.g., conversion from YCbCr 4:2:0 to RGB 4:4:4) or an inverse remapping performing the inverse of the remapping process performed in the pre-encoding processing (201). The post-decoding processing can use metadata derived in the pre-encoding processing and signaled in the bitstream.

[0092] FIG. 4 is a diagram showing an example of a system in which various aspects and embodiments described herein may be implemented. System 400 may be embodied as a device including the various components described below and is configured to perform one or more of the aspects described in this document. Examples of such devices include, but are not limited to, various electronic devices such as personal computers, laptop computers, smartphones, tablet computers, digital multimedia set top boxes, digital television receivers, personal video recording systems, connected home appliances, and servers. Elements of system 400, singly or in combination, may be embodied in a single integrated circuit (IC), multiple ICs, and/or discrete components. For example, in at least one example, the processing and encoder/decoder elements of system 400 are distributed across multiple ICs and/or discrete components. In various embodiments, the system 400 is communicatively coupled to one or more other systems, or other electronic devices, via, for example, a communications bus or through dedicated input and/or output ports. In various embodiments, the system 400 is configured to implement one or more of the aspects described in this document.

[0093] The system 400 includes at least one processor 410 configured to execute instructions loaded therein for implementing, for example, the various aspects described in this document. Processor 410 can include embedded memory, input output interface, and various other circuitries as known in the art. The

system 400 includes at least one memory 420 (e.g., a volatile memory device, and/or a non-volatile memory device). System 400 includes a storage device 440, which can include non-volatile memory and/or volatile memory, including, but not limited to, Electrically Erasable Programmable Read-Only Memory (EEPROM), Read-Only Memory (ROM), Programmable Read-Only Memory (PROM), Random Access Memory (RAM), Dynamic Random Access Memory (DRAM), Static Random Access Memory (SRAM), flash, magnetic disk drive, and/or optical disk drive. The storage device 440 can include an internal storage device, an attached storage device (including detachable and non-detachable storage devices), and/or a network accessible storage device, as non-limiting examples.

[0094] System 400 includes an encoder/decoder module 430 configured, for example, to process data to provide an encoded video or decoded video, and the encoder/decoder module 430 can include its own processor and memory. The encoder/decoder module 430 represents module(s) that may be included in a device to perform the encoding and/or decoding functions. As is known, a device can include one or both of the encoding and decoding modules. Additionally, encoder/decoder module 430 may be implemented as a separate element of system 400 or may be incorporated within processor 410 as a combination of hardware and software as known to those skilled in the art.

[0095] Program code to be loaded onto processor 410 or encoder/decoder 430 to perform the various aspects described in this document may be stored in storage device 440 and subsequently loaded onto memory 420 for execution by processor 410. In accordance with various embodiments, one or more of processor 410, memory 420, storage device 440, and encoder/decoder module 430 can store one or more of various items during the performance of the processes described in this document. Such stored items can include, but are not limited to, the input video, the decoded video or portions of the decoded video, the bitstream, matrices, variables, and intermediate or final results from the processing of equations, formulas, operations, and operational logic.

[0096] In some embodiments, memory inside of the processor 410 and/or the encoder/decoder module 430 is used to store instructions and to provide working memory for processing that is needed during encoding or decoding. In other embodiments, however, a memory external to the processing device (for example, the processing device may be either the processor 410 or the encoder/decoder module 430) is used for one or more of these functions. The external memory may be the memory 420 and/or the storage device 440, for example, a dynamic volatile memory and/or a non-volatile flash memory. In several embodiments, an external non-volatile flash memory is used to store the operating system of, for example, a television. In at least one embodiment, a fast external dynamic volatile memory such as a RAM is used as working memory for video coding and decoding operations, such as, for example, MPEG-2 (MPEG refers to the Moving Picture Experts Group, MPEG-2 is also referred to as ISO/IEC 13818, and 13818-1 is

also known as H.222, and 13818-2 is also known as H.262), HEVC (HEVC refers to High Efficiency Video Coding, also known as H.265 and MPEG-H Part 2), or VVC (Versatile Video Coding, a new standard being developed by JVET, the Joint Video Experts Team).

[0097] The input to the elements of system 400 may be provided through various input devices as indicated in block 445. Such input devices include, but are not limited to, (i) a radio frequency (RF) portion that receives an RF signal transmitted, for example, over the air by a broadcaster, (ii) a Component (COMP) input terminal (or a set of COMP input terminals), (iii) a Universal Serial Bus (USB) input terminal, and/or (iv) a High Definition Multimedia Interface (HDMI) input terminal. Other examples, not shown in FIG. 4, include composite video.

[0098] In various embodiments, the input devices of block 445 have associated respective input processing elements as known in the art. For example, the RF portion may be associated with elements suitable for (i) selecting a desired frequency (also referred to as selecting a signal, or band-limiting a signal to a band of frequencies), (ii) down-converting the selected signal, (iii) band-limiting again to a narrower band of frequencies to select (for example) a signal frequency band which may be referred to as a channel in certain embodiments, (iv) demodulating the down-converted and band-limited signal, (v) performing error correction, and (vi) demultiplexing to select the desired stream of data packets. The RF portion of various embodiments includes one or more elements to perform these functions, for example, frequency selectors, signal selectors, band-limiters, channel selectors, filters, downconverters, demodulators, error correctors, and demultiplexers. The RF portion can include a tuner that performs various of these functions, including, for example, down-converting the received signal to a lower frequency (for example, an intermediate frequency or a near-baseband frequency) or to baseband. In one set-top box embodiment, the RF portion and its associated input processing element receives an RF signal transmitted over a wired (for example, cable) medium, and performs frequency selection by filtering, down-converting, and filtering again to a desired frequency band. Various embodiments rearrange the order of the above-described (and other) elements, remove some of these elements, and/or add other elements performing similar or different functions. Adding elements can include inserting elements in between existing elements, such as, for example, inserting amplifiers and an analog-to-digital converter. In various embodiments, the RF portion includes an antenna.

[0099] Additionally, the USB and/or HDMI terminals can include respective interface processors for connecting system 400 to other electronic devices across USB and/or HDMI connections. It is to be understood that various aspects of input processing, for example, Reed-Solomon error correction, may be implemented, for example, within a separate input processing IC or within processor 410 as necessary. Similarly, aspects of USB or HDMI interface processing may be implemented within separate interface ICs

or within processor 410 as necessary. The demodulated, error corrected, and demultiplexed stream is provided to various processing elements, including, for example, processor 410, and encoder/decoder 430 operating in combination with the memory and storage elements to process the data stream as necessary for presentation on an output device.

[0100] Various elements of system 400 may be provided within an integrated housing. Within the integrated housing, the various elements may be interconnected and transmit data therebetween using suitable connection arrangement 425, for example, an internal bus as known in the art, including the Inter-IC (I2C) bus, wiring, and printed circuit boards.

[0101] The system 400 includes communication interface 450 that enables communication with other devices via communication channel 460. The communication interface 450 can include, but is not limited to, a transceiver configured to transmit and to receive data over communication channel 460. The communication interface 450 can include, but is not limited to, a modem or network card and the communication channel 460 may be implemented, for example, within a wired and/or a wireless medium.

[0102] Data is streamed, or otherwise provided, to the system 400, in various embodiments, using a wireless network such as a Wi-Fi network, for example IEEE 802.11 (IEEE refers to the Institute of Electrical and Electronics Engineers). The Wi-Fi signal of these examples is received over the communications channel 460 and the communications interface 450 which are adapted for Wi-Fi communications. The communications channel 460 of these embodiments is typically connected to an access point or router that provides access to external networks including the Internet for allowing streaming applications and other over-the-top communications. Other embodiments provide streamed data to the system 400 using a set-top box that delivers the data over the HDMI connection of the input block 445. Still other embodiments provide streamed data to the system 400 using the RF connection of the input block 445. As indicated above, various embodiments provide data in a non-streaming manner. Additionally, various embodiments use wireless networks other than Wi-Fi, for example a cellular network or a Bluetooth network.

[0103] The system 400 can provide an output signal to various output devices, including a display 475, speakers 485, and other peripheral devices 495. The display 475 of various embodiments includes one or more of, for example, a touchscreen display, an organic light-emitting diode (OLED) display, a curved display, and/or a foldable display. The display 475 may be for a television, a tablet, a laptop, a cell phone (mobile phone), or other device. The display 475 can also be integrated with other components (for example, as in a smart phone), or separate (for example, an external monitor for a laptop). The other peripheral devices 495 include, in various examples of embodiments, one or more of a stand-alone digital video disc (or digital versatile disc) (DVR, for both terms), a disk player, a stereo system, and/or a lighting

system. Various embodiments use one or more peripheral devices 495 that provide a function based on the output of the system 400. For example, a disk player performs the function of playing the output of the system 400.

[0104] In various embodiments, control signals are communicated between the system 400 and the display 475, speakers 485, or other peripheral devices 495 using signaling such as AV.Link, Consumer Electronics Control (CEC), or other communications protocols that enable device-to-device control with or without user intervention. The output devices may be communicatively coupled to system 400 via dedicated connections through respective interfaces 470, 480, and 490. Alternatively, the output devices may be connected to system 400 using the communications channel 460 via the communications interface 450. The display 475 and speakers 485 may be integrated in a single unit with the other components of system 400 in an electronic device such as, for example, a television. In various embodiments, the display interface 470 includes a display driver, such as, for example, a timing controller (T Con) chip.

[0105] The display 475 and speakers 485 can alternatively be separate from one or more of the other components, for example, if the RF portion of input 445 is part of a separate set-top box. In various embodiments in which the display 475 and speakers 485 are external components, the output signal may be provided via dedicated output connections, including, for example, HDMI ports, USB ports, or COMP outputs.

[0106] The embodiments may be carried out by computer software implemented by the processor 410 or by hardware, or by a combination of hardware and software. As a non-limiting example, the embodiments may be implemented by one or more integrated circuits. The memory 420 may be of any type appropriate to the technical environment and may be implemented using any appropriate data storage technology, such as optical memory devices, magnetic memory devices, semiconductor-based memory devices, fixed memory, and removable memory, as non-limiting examples. The processor 410 may be of any type appropriate to the technical environment, and can encompass one or more of microprocessors, general purpose computers, special purpose computers, and processors based on a multi-core architecture, as non-limiting examples.

[0107] Various implementations involve decoding. "Decoding", as used in this application, can encompass all or part of the processes performed, for example, on a received encoded sequence in order to produce a final output suitable for display. In various embodiments, such processes include one or more of the processes typically performed by a decoder, for example, entropy decoding, inverse quantization, inverse transformation, and differential decoding. In various embodiments, such processes also, or alternatively, include processes performed by a decoder of various implementations described in this application, for example, operating in a first image processing mode; calculating, based on a plurality of

timestamps, a latency associated with an image processing action of the first image processing mode; comparing the latency to a delay threshold; determining whether to switch to a second image processing mode based on the comparison; etc.

[0108] As further embodiments, in one example “decoding” refers only to entropy decoding, in another embodiment “decoding” refers only to differential decoding, and in another embodiment “decoding” refers to a combination of entropy decoding and differential decoding. Whether the phrase “decoding process” is intended to refer specifically to a subset of operations or generally to the broader decoding process will be clear based on the context of the specific descriptions and is believed to be well understood by those skilled in the art.

[0109] Various implementations involve encoding. In an analogous way to the above discussion about “decoding”, “encoding” as used in this application can encompass all or part of the processes performed, for example, on an input video sequence in order to produce an encoded bitstream. In various embodiments, such processes include one or more of the processes typically performed by an encoder, for example, partitioning, differential encoding, transformation, quantization, and entropy encoding. In various embodiments, such processes also, or alternatively, include processes performed by an encoder of various implementations described in this application, for example, operating in a first image processing mode; calculating, based on a plurality of timestamps, a latency associated with an image processing action of the first image processing mode; comparing the latency to a delay threshold; determining whether to switch to a second image processing mode based on the comparison; etc.

[0110] As further examples, in one embodiment “encoding” refers only to entropy encoding, in another embodiment “encoding” refers only to differential encoding, and in another embodiment “encoding” refers to a combination of differential encoding and entropy encoding. Whether the phrase “encoding process” is intended to refer specifically to a subset of operations or generally to the broader encoding process will be clear based on the context of the specific descriptions and is believed to be well understood by those skilled in the art.

[0111] Note that syntax elements as used herein, such as syntax elements that may be indicated in Tables 1-2 and otherwise indicated in discussion or figures presented herein, are descriptive terms. As such, they do not preclude the use of other syntax element names.

[0112] When a figure is presented as a flow diagram, it should be understood that it also provides a block diagram of a corresponding apparatus. Similarly, when a figure is presented as a block diagram, it should be understood that it also provides a flow diagram of a corresponding method/process.

[0113] During the encoding process, the balance or trade-off between the rate and distortion is usually considered, often given the constraints of computational complexity. The rate distortion optimization is

usually formulated as minimizing a rate distortion function, which is a weighted sum of the rate and of the distortion. There are different approaches to solve the rate distortion optimization problem. For example, the approaches may be based on an extensive testing of all encoding options, including all considered modes or coding parameters values, with a complete evaluation of their coding cost and related distortion of the reconstructed signal after coding and decoding. Faster approaches may also be used, to save encoding complexity, in particular with computation of an approximated distortion based on the prediction or the prediction residual signal, not the reconstructed one. Mix of these two approaches can also be used, such as by using an approximated distortion for only some of the possible encoding options, and a complete distortion for other encoding options. Other approaches only evaluate a subset of the possible encoding options. More generally, many approaches employ any of a variety of techniques to perform the optimization, but the optimization is not necessarily a complete evaluation of both the coding cost and related distortion.

[0114] The implementations and aspects described herein may be implemented in, for example, a method or a process, an apparatus, a software program, a data stream, or a signal. Even if only discussed in the context of a single form of implementation (for example, discussed only as a method), the implementation of features discussed can also be implemented in other forms (for example, an apparatus or program). An apparatus may be implemented in, for example, appropriate hardware, software, and firmware. The methods may be implemented in, for example, a processor, which refers to processing devices in general, including, for example, a computer, a microprocessor, an integrated circuit, or a programmable logic device. Processors also include communication devices, such as, for example, computers, cell phones, portable/personal digital assistants ("PDAs"), and other devices that facilitate communication of information between end-users.

[0115] Reference to "one embodiment," "an embodiment," "an example," "one implementation" or "an implementation," as well as other variations thereof, means that a particular feature, structure, characteristic, and so forth described in connection with the embodiment is included in at least one embodiment. Thus, the appearances of the phrase "in one embodiment," "in an embodiment," "in an example," "in one implementation," or "in an implementation", as well any other variations, appearing in various places throughout this application are not necessarily all referring to the same embodiment or example.

[0116] Additionally, this application may refer to "determining" various pieces of information. Determining the information can include one or more of, for example, estimating the information, calculating the information, predicting the information, or retrieving the information from memory. Obtaining may include receiving, retrieving, constructing, generating, and/or determining.

[0117] Further, this application may refer to “accessing” various pieces of information. Accessing the information can include one or more of, for example, receiving the information, retrieving the information (for example, from memory), storing the information, moving the information, copying the information, calculating the information, determining the information, predicting the information, or estimating the information.

[0118] Additionally, this application may refer to “receiving” various pieces of information. Receiving is, as with “accessing”, intended to be a broad term. Receiving the information can include one or more of, for example, accessing the information, or retrieving the information (for example, from memory). Further, “receiving” is typically involved, in one way or another, during operations such as, for example, storing the information, processing the information, transmitting the information, moving the information, copying the information, erasing the information, calculating the information, determining the information, predicting the information, or estimating the information.

[0119] It is to be appreciated that the use of any of the following “/”, “and/or”, and “at least one of”, for example, in the cases of “A/B”, “A and/or B” and “at least one of A and B”, is intended to encompass the selection of the first listed option (A) only, or the selection of the second listed option (B) only, or the selection of both options (A and B). As a further example, in the cases of “A, B, and/or C” and “at least one of A, B, and C”, such phrasing is intended to encompass the selection of the first listed option (A) only, or the selection of the second listed option (B) only, or the selection of the third listed option (C) only, or the selection of the first and the second listed options (A and B) only, or the selection of the first and third listed options (A and C) only, or the selection of the second and third listed options (B and C) only, or the selection of all three options (A and B and C). This may be extended, as is clear to one of ordinary skill in this and related arts, for as many items as are listed.

[0120] Also, as used herein, the word “signal” refers to, among other things, indicating something to a corresponding decoder. In this way, in an embodiment the same parameter is used at both the encoder side and the decoder side. Thus, for example, an encoder can transmit (explicit signaling) a particular parameter to the decoder so that the decoder can use the same particular parameter. Conversely, if the decoder already has the particular parameter as well as others, signaling may be used without transmitting (implicit signaling) to simply allow the decoder to know and select the particular parameter. By avoiding transmission of any actual functions, a bit savings is realized in various embodiments. It is to be appreciated that signaling may be accomplished in a variety of ways. For example, one or more syntax elements, flags, and so forth are used to signal information to a corresponding decoder in various embodiments. While the preceding relates to the verb form of the word “signal”, the word “signal” can also be used herein as a noun.

[0121] As will be evident to one of ordinary skill in the art, implementations may produce a variety of signals formatted to carry information that may be, for example, stored or transmitted. The information can include, for example, instructions for performing a method, or data produced by one of the described implementations. For example, a signal may be formatted to carry the bitstream of a described embodiment. Such a signal may be formatted, for example, as an electromagnetic wave (for example, using a radio frequency portion of spectrum) or as a baseband signal. The formatting may include, for example, encoding a data stream and modulating a carrier with the encoded data stream. The information that the signal carries may be, for example, analog or digital information. The signal may be transmitted over a variety of different wired or wireless links, as is known. The signal may be stored on a processor-readable medium.

[0122] Device architectures for an Extended Reality (XR) application may be defined.

[0123] The delays and latencies in XR application may impact the quality of experience (QoE) of the XR application (e.g., ensure a high QoE). Different XR roundtrip delays may be involved in an XR application. For example, a motion-to-photon latency may be less than 20ms. As another example, a roundtrip interaction delay may be as small as 50-60ms.

[0124] A rendering loop of an XR application may be used to measure latency for the QoE metric. The rendering loop may be a component of an XR application. The rendering loop may generate the images that are displayed to the user.

[0125] Information related to latencies may be collected. The rendering loop of an XR Runtime may need to be understood (e.g., to collect the information related to latencies).

[0126] FIG. 5 illustrates an example of a rendering loop for visual data. The rendering loop may be used if a session is running and is in a focused state.

[0127] An XR application may wait on a swapchain image (e.g., before the application begins writing to the swapchain image) to avoid writing to the swapchain image before the compositor has finished reading from the swapchain image. The XR application may synchronize its rendering loop to the XR Runtime. If the XR application has pipelined frame submissions, the XR application may compute a target display time. For example, the XR application may compute the target display time using a predicted display time and a predicted display interval (e.g., both the predicted display time and the predicted display interval).

[0128] The XR application may initiate the rendering process (e.g., after the wait time is complete). The XR application may render different views. The XR Runtime may provide access to viewer pose and projection parameters (e.g., that may be used to render the different views). The view and projection information may be provided for a display time within a specified XR space. The target/predicted display time may be for a given frame.

[0129] The XR application may perform rendering work. Rendering work may involve copying (e.g., directly copying) data from the application into the swapchain. The rendering work may involve iterating over the scene graph nodes and rendering complex objects. If the views/layers (e.g., all views/layers) are rendered, the application may send the views/layers to the XR Runtime for compositing (e.g., final compositing). The application may include (e.g., also send) the expected display time and the associated render pose.

[0130] The XR application may offload the composition of the final image to an XR Runtime-supplied compositor. The compositor may blend and flatten the layers (e.g., all layers). The compositor may present this image (e.g., with the blended and flattened layers) to the system's display at a specific display time.

[0131] Feature(s) associated with metrics computation are provided herein. The time difference between sampling of the view and projection information from the Runtime and providing the corresponding rendered information to the Runtime for display with the associated render pose may be computed. The difference between the associated render pose and the actual pose at the time when the data is rendered may be computed.

[0132] The rendering loop may include a (e.g., single) rendering processing module. The rendering processing module may take as inputs the actions and the viewer pose to render a three-dimensional (3D) scene.

[0133] The XR application may consider a change/modification of the virtual objects by updating the related scene graph (e.g., before rendering the 3D scene from the viewer pose). The update operation may include a potential physics simulation. The update operation may handle interactivity events (e.g., trigger activation, launching of actions, etc.). The update operation may be time-consuming. The duration of the update operation may vary from one rendering frame to another.

[0134] The scene update step and the rendering processing step for the delay measurement may be distinguished. The distinction may improve the QoE of the XR application. The XR application may be able to perform a (e.g., proper) latency adjustment strategy on the steps (e.g., each of the two steps) to improve the XR roundtrip delays (e.g., such as the motion-to-photon or the roundtrip interaction delay).

[0135] Measuring the latency of a processing step (e.g., each processing step) of the rendering loop may involve adding observation points to collect measurements. The configuration of a stage (e.g., each of these stages) may be adjusted (e.g., to ensure that the XR application provides a high quality of user experience).

[0136] The adjustment may be based on different configuration levels (e.g., that modify quality mode and resource consumption). If the delay of the processing step has been measured, the application may switch between levels (e.g., to stay in a target range of the processing delay).

[0137] A separation between the scene update and the rendering may be introduced. The steps of the rendering loop may be modified. For example, observation points (OPs) may be used to collect measurements. Feature(s) associated with proper delay adjustment are provided herein.

[0138] Feature(s) associated with modifying the rendering loop are provided herein.

[0139] The scene update may be introduced (e.g., added) to the rendering loop.

[0140] Scene update may involve updating the 3D scene graph based on the current state of the application. The scene update may involve performing one or more of the following tasks (e.g., to maintain an up-to-date representation of the scene): updating the background texture with an image captured by a camera for video see-through displays; adding/removing objects to/from the scene; updating the pose of the virtual viewpoint according to the viewer pose estimation; updating the pose of a virtual objects according to physics simulation, defined animations, runtime interactions, and/or the pose estimation of an augmented reality (AR) anchor; updating the occlusion mask with the 3D reconstruction of the real space for AR application; updating the properties of an object (e.g., the object's color, size, etc.) based on runtime interactivity; and/or provisioning the scene for rendering or updating the complexity of the scene according to 3D rendering performances.

[0141] Scene rendering may involve the (e.g., actual) 3D rendering of the scene. The system may generate images of the visible objects in the scene. The system may apply various visual effects (e.g., lighting and shadows). The system may generate a final image that is ready to be displayed to the user.

[0142] A display stage of the rendering loop may involve displaying the rendered image on the device's display. The system may perform additional post-processing steps.

[0143] The update of the scene graph may occur at the beginning of the rendering loop (e.g., before the rendering stages take place).

[0144] FIG. 6 illustrates a flowchart of the observation points.

[0145] Observation points may be added at an interface (e.g., each interface) of the processing steps. The observation points may be used to collect relevant information in order to measure the latency of a step (e.g., each step).

[0146] FIG. 7 illustrates an example rendering loop with observation points.

[0147] The information related to the viewer pose and the user interaction (actions) may be collected at a first observation point (OP1). The time when the scene manager retrieves the viewer pose and the actions may be recorded at OP1. This time may be associated with a start of a scene graph update (e.g., correspond to a start-scene-update-time timestamp). For example, the time at OP1 may be the time at which the scene manager starts to update the scene graph.

[0148] The device may determine a time associated with an end of the scene graph update and a start of a rendering process. The time when the scene graph update ends may be recorded at a second observation point (OP2). This time may correspond to an end-scene-update-time timestamp (e.g., the time at which the scene manager has updated the scene graph). The scene graph update may involve updating the pose of the virtual camera in the scene, the pose of objects in the scene, applying physics simulations, and/or processing user input. If the scene graph update is complete, the scene representation may be ready to be rendered.

[0149] The end-scene-update-time timestamp or start-render-time timestamp may be captured at the start of the rendering at OP2.

[0150] The device may determine a time associated with an end of the rendering process. For example, the time when the rendering process ends may be recorded at a third observation point (OP3). This time may correspond to an end-render-time timestamp (e.g., the end of the rendering process). One or more (e.g., all) views/layers may be rendered. The XR application may send the rendered views/layers to the XR Runtime for final compositing and display.

[0151] The end-render-time timestamp may be captured at the end of the rendering at OP3.

[0152] FIG. 8 illustrates an example call flow of the rendering loop with the observation points. The roundtrip interaction delay and the motion-to-photon may be impacted (e.g., directly impacted) by the scene update and the rendering.

[0153] In an example (e.g., with a Unity 3D framework), the observation points may be defined based on an order of execution.

[0154] For example, OP1 may be at the start of a new frame cycle (e.g., when the user interactions and the viewer pose are read from the input buffer). OP1 may be inserted at the beginning of an "EarlyUpdate" event.

[0155] OP2 may be at the end of the game logic (e.g., on a "LateUpdate" event, before the scene rendering, at the beginning of a "PostLateUpdate" event).

[0156] OP3 may be at the end of a frame (e.g., on an "EndOfFrame" event or at the end of the "PostLateUpdate" event, after the rendering).

[0157] Latencies may be measured and adjusted.

[0158] Using the timestamps collected at the observation points, the latency of the scene update and the rendering may be calculated by the XR application. The device may calculate a first latency associated with the scene graph update based on the time at OP1 and the time at OP2. The device may calculate a second

latency associated with the rendering process based on the time at OP2 and the time at OP3. For example, the latency of the scene update may be calculated as:

$$\text{scene-update-delay} = \text{start-render-time} - \text{start-scene-update-time}$$

and the latency of the rendering may be calculated as:

$$\text{rendering-delay} = \text{end-render-time} - \text{start-render-time}.$$

[0159] The XR application may adjust one or more parameters associated with user experience quality based on the first and/or second latency. For example, the XR application may adjust the processing step configuration based on the delays (e.g., to ensure that the QoE for motion-to-photon and/or the roundtrip interaction delay is achieved).

[0160] Feature(s) described herein may be implemented in a standard.

[0161] The observation points may be used to estimate the times taken by the following two processes: update of the 3D scene, and rendering.

[0162] For a process (e.g., each process), one or more (e.g., two) thresholds may be defined. For example, a low delay threshold (e.g., low time budget), and a high delay threshold (e.g., high time budget) may be used.

[0163] If the measured delay is under (e.g., below) the "low delay threshold," the complexity of the process may be increased to meet the XR round trip delay. If the measured delay is over the "high delay threshold," the complexity of the process may be decreased to meet the XR round trip delay.

[0164] Thresholds values may be expressed in milliseconds.

[0165] A set of parameters may be used to modify the process time. Different values of these parameters may be sorted and arranged by level. The levels may be representative of different image processing modes. For example, a profile at level 0 may correspond to a best quality mode (e.g., with more resource consumption). The subsequent levels may correspond to increasingly degraded modes.

[0166] Thresholds may be different for the two processes (updating the scene and rendering). One or more (e.g., two) sets of thresholds may be defined (e.g., when the scene is created by the content creator).

[0167] FIG. 9 is an example flowchart of processing step adjustment. FIG. 10 illustrates an example of delay adjustment. FIG. 10 illustrates an example of switching between levels. The measurement and the processing step adjustments are illustrated in FIG. 10.

[0168] As illustrated in FIG. 10, level 0 may be used at initialization. If the first significant measured value is over the "High delay threshold," the application may switch to level 1. If the measured value falls under the "Low delay threshold," the application may switch to level 0.

[0169] If the measured value is under the "Low delay threshold" at level 0, the system may not change levels (e.g., the system may do nothing).

[0170] If the measured value is between the "Low delay threshold" and the "High delay threshold" at level 0, the system may not change levels (e.g., the system may do nothing).

[0171] The 3D scene may be update. Parameters may be used to modify processing time. To improve update cycle, some elements of the scene graph may be adjusted. For example, animations (e.g., interpolation between keyframes, number of keyframes, etc.) may be adjusted. Colliders (e.g., meshes used to manage collisions) may be adjusted. For example, a collision may have a bounding box (e.g., and may therefore use less computation power). Physics (e.g., materials used to compute physics) may be adjusted. Other parameters may be adjusted.

[0172] Parameter(s) may be used to modify rendering processing time. To improve rendering cycle, one or more parameters may be adjusted. For example, lights (e.g., complexity of cast shadows, ambient light, etc.) may be adjusted. For example, for a selected light (e.g., each light selected), a cast shadows operation may be skipped.

[0173] A level of detail (LOD) (e.g., usage of simplified meshes, materials or textures, etc.) may be adjusted. For example, for LOD, meshes or materials may be switched (e.g., instead of nodes). Other parameters may be adjusted.

[0174] The parameters used to update the processing time may be provided in a file (e.g., a manifest-like file or a scene description file) sent to the XR application (e.g., at the beginning of the XR session).

[0175] In examples, the parameters used to update the processing time may be similar to, or the same as, parameters provided in an MPEG-I Scene Description based on a Khronos glTF format.

[0176] For scene update, table 1 illustrates an index of elements that may be used for level 0 and corresponding alternatives for the other levels.

[0177] At level 0, for animation, samplers at indexes 10 and 11 may be used, meshes at indexes 10 and 12 may be used to compute collision, and material at index 0 may be used.

[0178] At level 1, for animation, samplers at indexes 10 and 11 may be replaced by samplers at indexes 20 and 21, meshes at indexes 10 and 12 may be replaced by meshes at indexes 20 and 22 to compute collision, and material at index 0 may be replaced by material at index 5.

Update scene	of	Animations/samplers	Colliders/meshes	Physics/Materials
Level 0		10;11	10;12	0
Level 1		20;21	20;22	5

Level 2	30;31	30;34	9
---------	-------	-------	---

Table 1: index of elements for different levels

- [0179] For rendering, table 2 illustrates an index of elements that may be used for level 0 and corresponding alternatives for the other levels.
- [0180] At level 0, for light at index 0, cast shadow may be set to true, and nodes at indexes 0 and 1 may be used.
- [0181] At level 1, light at index 0 may be replaced by light at index 5, cast shadow may be set to true, and nodes at indexes 0 and 1 may be replaced by nodes at indexes 10 and 11.
- [0182] At level 2, light at index 5 may be replaced by light at index 7, cast shadow may be set to false, and nodes at indexes 10 and 11 may be replaced by nodes at indexes 20 and 21.

Rendering	Lights (light index, cast shadow)	LOD List index of nodes
Level 0	(0, true)	0;1
Level 1	(5, true)	10;11
Level 2	(7, false)	20;21
Level 3	(not set)	30;31

Table 2: example of level adjustment

- [0183] An example of implementation in a glTF file (e.g., with a new extension that may be referred to as "IDCC_roundTripMGT") is provided below.

```
{
  "asset": {
    "generator": "Khronos glTF Blender I/O v1.8.19",
    "version": "2.0"
  },
  "extensions": {
    "IDCC_roundTripMGT": {
      "update": {
        "criteria": [{"low delay threshold":10,"high delay threshold":50}],
        "animations":{
          "samplers":[{"level0":10,"level1":20,"level2":30},{ "level0":11,
"level1":21,"level2":31}]
        },
        "colliders":{
          "meshes":[{"level0":10,"level1":20,"level2":30},{ "level0":12,
```

```

"level1":22,"level2":34}}
    },
    "physics":{
        "materials":[
            {"level0": 0, "level1":5,"level2":9}
        ]
    }
},
"rendering":
{
    "criteria":{"low delay threshold":16,"high delay threshold":34}},
    "lights":{"level0":{"light":0, "castShadow":true} , "level1":{"light":5,
"castShadow":true},"level2":{"light":7, "castShadow":false},"level3":{"light":7, "castShadow":false}},
    "lod":{
        "nodes":{"level0":0,"level1":10,"level2":20,"level3":30},{level0":1,
"level1":11,"level2":21,"level3":31}}
    },
    "materials":{"level0": 0, "level1":5,"level2":9,"level3":19}},
    "meshes":{"level0":1,"level1":8,"level2":15,"level3":22},{level0":2,
"level1":9,"level2":16,"level3":23}}
    }
}

},
"scene" : 0,
"scenes" : [
    {
        "name" : "Scene",
        "nodes" : [
            0,
            1,
            2,
            4
        ]
    }
],

```

[0184] Although features and elements are described above in particular combinations, one of ordinary skill in the art will appreciate that each feature or element can be used alone or in any combination with the other features and elements. In addition, the methods described herein may be implemented in a computer program, software, or firmware incorporated in a computer-readable medium for execution by a computer or processor. Examples of computer-readable media include electronic signals (transmitted over wired or wireless connections) and computer-readable storage media. Examples of computer-readable storage

media include, but are not limited to, a read only memory (ROM), a random access memory (RAM), a register, cache memory, semiconductor memory devices, magnetic media such as internal hard disks and removable disks, magneto-optical media, and optical media such as CD-ROM disks, and digital versatile disks (DVDs). A processor in association with software may be used to implement a radio frequency transceiver for use in a WTRU, UE, terminal, base station, RNC, or any host computer.

CLAIMS

1. A device, comprising:
 - a processor configured to:
 - determine a first time associated with a start of a scene graph update;
 - determine a second time associated with an end of the scene graph update and a start of a rendering process;
 - determine a third time associated with an end of the rendering process;
 - calculate a first latency associated with the scene graph update based on the first time and the second time;
 - calculate a second latency associated with the rendering process based on the second time and the third time; and
 - adjust a first parameter associated with user experience quality based, at least in part, on the first latency.
2. The device of claim 1, wherein the processor is further configured to adjust a second parameter associated with user experience quality based, at least in part, on the second latency.
3. The device of claim 1 or 2, wherein the processor being configured to adjust the first parameter associated with user experience quality based, at least in part, on the first latency comprises the processor being configured to:
 - determine that the first latency is below a first delay threshold, and
 - increase the quality and resource consumption of the scene graph update based on the determination that the first latency is below the first delay threshold; or
 - determine that the first latency is above a second delay threshold, and
 - decrease the quality and resource consumption of the scene graph update based on the determination that the first latency is above the second delay threshold.
4. The device of claim 2 or 3, wherein the processor being configured to adjust the second parameter associated with user experience quality based, at least in part, on the second latency comprises the processor being configured to:
 - determine that the second latency is below a third delay threshold, and

increase the quality and resource consumption of the rendering process based on the determination that the second latency is below the third delay threshold; or
determine that the second latency is above a fourth delay threshold, and
decrease the quality and resource consumption of the rendering process based on the determination that the second latency is above the fourth delay threshold.

5. The device of any one of claims 1-4, wherein the processor being configured to calculate the first latency associated with the scene graph update based on the first time and the second time comprises the processor being configured to determine a first time difference between the first time and the second time, wherein the first latency is equal to the first time difference, and the processor being configured to calculate the second latency associated with the rendering process based on the second time and the third time comprises the processor being configured to determine a second time difference between the second time and the third time, wherein the second latency is equal to the second time difference.

6. The device of any one of claims 1-5, wherein the first parameter associated with user experience quality is associated with at least one of:

- interpolation between keyframes;
- a number of keyframes;
- a mesh associated with collision management; or
- physics computation.

7. The device of any one of claims 2-6, wherein the second parameter associated with user experience quality is associated with at least one of:

- shadow complexity;
- ambient light complexity;
- a level of detail of a mesh;
- a level of detail of a material; or
- a level of detail of a texture.

8. The device of claim 1 or 2, wherein the processor is further configured to:

- receive first information associated with an image;
- perform the scene graph update based on the information associated with the image, wherein the processor being configured to perform the scene graph update based on the first information associated

with the image comprises the processor being configured to output second information associated with the image;

perform the rendering process based on the second information associated with the image,
wherein the processor being configured to perform the rendering process based on the second information associated with the image comprises the processor being configured to output a rendering of the image;
and

send the rendering of the image to a display.

9. A method, comprising:

determining a first time associated with a start of a scene graph update;

determining a second time associated with an end of the scene graph update and a start of a rendering process;

determining a third time associated with an end of the rendering process;

calculating a first latency associated with the scene graph update based on the first time and the second time;

calculating a second latency associated with the rendering process based on the second time and the third time; and

adjusting a first parameter associated with user experience quality based, at least in part, on the first latency.

10. The method of claim 9, wherein the method further comprises adjusting a second parameter associated with user experience quality based, at least in part, on the second latency.

11. The method of claim 9 or 10, wherein adjusting the first parameter associated with user experience quality based, at least in part, on the first latency comprises:

determining that the first latency is below a first delay threshold, and

increasing the quality and resource consumption of the scene graph update based on the determination that the first latency is below the first delay threshold; or

determining that the first latency is above a second delay threshold, and

decreasing the quality and resource consumption of the scene graph update based on the determination that the first latency is above the second delay threshold.

12. The method of claim 10 or 11, wherein adjusting the second parameter associated with user experience quality based, at least in part, on the second latency comprises:

determining that the second latency is below a third delay threshold, and
increasing the quality and resource consumption of the rendering process based on the
determination that the second latency is below the third delay threshold; or
determining that the second latency is above a fourth delay threshold, and
decreasing the quality and resource consumption of the rendering process based on the
determination that the second latency is above the fourth delay threshold.

13. The method of any one of claims 9-12, wherein calculating the first latency associated with the scene graph update based on the first time and the second time comprises determining a first time difference between the first time and the second time, wherein the first latency is equal to the first time difference, and calculating the second latency associated with the rendering process based on the second time and the third time comprises determining a second time difference between the second time and the third time, wherein the second latency is equal to the second time difference.

14. The method of any one of claims 9-13, wherein the first parameter associated with user experience quality is associated with at least one of:

- interpolation between keyframes;
- a number of keyframes;
- a mesh associated with collision management; or
- physics computation.

15. The method of any one of claims 10-14, wherein the second parameter associated with user experience quality is associated with at least one of:

- shadow complexity;
- ambient light complexity;
- a level of detail of a mesh;
- a level of detail of a material; or
- a level of detail of a texture.

16. The method of claim 9 or 10, wherein the method further comprises:

- receiving first information associated with an image;
- performing the scene graph update based on the information associated with the image, wherein performing the scene graph update based on the first information associated with the image comprises outputting second information associated with the image;

performing the rendering process based on the second information associated with the image,
wherein performing the rendering process based on the second information associated with the image
comprises outputting a rendering of the image; and
sending the rendering of the image to a display.