

US Patent & Trademark Office

Patent Public Search | Text View

United States Patent Application Publication

20250260785

Kind Code

A1

Publication Date

August 14, 2025

Inventor(s)

Nicholson; John W. et al.

VIDEO PROCESSING ADJUSTMENT BASED ON USER LOOKING/NOT LOOKING

Abstract

In one aspect, a first device includes a processor assembly, a display, and storage. The storage includes instructions executable by the processor assembly to receive video from a second device different from the first device, determine whether a user is looking at the display, and process the video a first way for presentation on the display based on a determination that the user is not looking at the display. The instructions are also executable to process the video a second way for presentation on the display based on a determination that the user is looking at the display. The first way may include using a lower resolution or slower frame rate than the second way. The first way may also include using a CPU little core rather than a GPU.

Inventors: Nicholson; John W. (Morrisville, NC), Cromer; Daryl C. (Morrisville, NC), Locker; Howard (Morrisville, NC)

Applicant: LENOVO (SINGAPORE) PTE. LTD. (SINGAPORE, SG)

Family ID: 94536380

Appl. No.: 18/436501

Filed: February 08, 2024

Publication Classification

Int. Cl.: H04N7/01 (20060101); G06F3/01 (20060101)

U.S. Cl.:

CPC H04N7/0117 (20130101); G06F3/013 (20130101); H04N7/013 (20130101);

Background/Summary

FIELD

[0001] The disclosure below relates to technically inventive, non-routine solutions that are necessarily rooted in computer technology and that produce concrete technical improvements. In particular, the disclosure below relates to techniques for dynamic video decoding based on whether the user is looking at the video or not.

BACKGROUND

[0002] As recognized herein, video decoding is typically done by a computing device's graphics processing unit (GPU). However, this often causes significant battery drain and/or undue power consumption from a wall outlet. In fact, high-quality video streaming applications often require a dedicated GPU for decoding, as that might be the only processing unit capable of maintaining an optimal frame rate and resolution for the video stream. But this accelerates battery drain/power consumption even further.

[0003] Additionally, video decoding on the GPU over long periods of time can result in undue heat buildup within the device, adversely affecting the overall life of other hardware components that absorb the heat that gets generated. Heat buildup can also adversely affect system performance on a short-term basis as well, sometimes even causing the other components to become inoperable until the heat dissipates.

[0004] There are currently no adequate solutions to the foregoing computer-related, technological problems.

SUMMARY

[0005] Accordingly, in one aspect a first device includes a processor assembly, a display accessible to the processor assembly, and storage accessible to the processor assembly. The storage includes instructions executable by the processor assembly to receive video from a second device different from the first device, determine whether a user is looking at the display, and process the video a first way for presentation on the display based on a determination that the user is not looking at the display. The instructions are also executable to process the video a second way for presentation on the display based on a determination that the user is looking at the display, where the second way is different from the first way.

[0006] In some example implementations, the first way may include processing the video for presentation on the display at a first resolution, and the second way may include processing the video for presentation on the display at a second resolution. Here, the second resolution may be higher than the first resolution.

[0007] Additionally or alternatively, the first way may include processing the video for presentation on the display at a first frame rate, and the second way may include processing the video for presentation on the display at a second frame rate. The second frame rate may be faster than the first frame rate.

[0008] As another example, the first way may include processing the video for presentation on the display using a central processing unit (CPU), and the second way may include processing the video for presentation on the display using a graphics processing unit (GPU). The GPU may be different from the CPU. Additionally, in various specific example implementations the first way may include processing the video for presentation on the display using a larger core of two cores on the CPU, while in other example implementations the first way may include processing the video for presentation on the display using a smaller core of at least two cores on the CPU.

[0009] As another specific example, the first way may include each of processing the video for presentation on the display at the first resolution, processing the video for presentation on the display at the first frame rate, and processing the video for presentation on the display using the CPU. In this case, the second way may include each of processing the video for presentation on the display at the second resolution, processing the video for presentation on the display at the second frame rate, and processing the video for presentation on the display using the GPU.

[0010] Still further, in some cases the instructions may be executable to receive input from a camera. The instructions may then be executable to execute eye tracking using the input from the camera to determine whether the user is looking at the display. The first device might even include the camera in some instances.

[0011] In addition to or in lieu of that, the instructions may also be executable to receive input from a human presence detection (HPD) sensor and then determine whether the user is looking at the display using the input from the HPD sensor. Here the first device might even include the HPD sensor itself.

[0012] Furthermore, in various example implementations, the instructions may also be executable to maintain a buffer of a threshold amount of past video frames while processing the video the first way. In one specific example, the instructions may even be executable to maintain a buffer of a threshold amount of past video frames in multiple resolutions for each respective past video frame while processing the video the first way.

[0013] In another aspect, a method includes receiving, at a first device, video from a second device different from the first device. The method also includes determining whether a user is looking at a display on which the video is to be presented. Based on determining that the user is not looking at the display, the method includes processing the video a first way at the first device for presentation on the display. Additionally, based on determining that the user is looking at the display, the method includes processing the video a second way at the first device for presentation on the display. The second way is different from the first way.

[0014] In a specific example implementation, the first way may include declining to process first non-I-frames of the video for presentation on the display, and the second way may include processing the first non-I-frames of the video for presentation on the display. The first non-I-frames may include P-frames of the video and/or B-frames of the video.

[0015] In still another aspect, at least one computer readable storage medium (CRSM) that is not a transitory signal includes instructions executable by a processor assembly to receive, at a first device, video from a second device different from the first device. The instructions are also executable to determine whether a user is looking at a display and, based on a determination that the user is not looking at the display, process the video a first way for presentation on the display. Based on a determination that the user is looking at the display, the instructions are executable to process the video a second way for presentation on the display. The second way is different from the first way.

[0016] Thus, in one example the first way may include declining to process first (e.g., any/all) non-I-frames of the video for presentation on the display, and the second way may include processing the first non-I-frames of the video for presentation on the display.

[0017] Additionally or alternatively, the first way may include processing the video for presentation on the display using a central processing unit (CPU), and the second way may include processing the video for presentation on the display using a graphics processing unit (GPU). The GPU may be different from the CPU.

[0018] The details of present principles, both as to their structure and operation, can best be understood in reference to the accompanying drawings, in which like reference numerals refer to like parts, and in which:

Description

BRIEF DESCRIPTION OF THE DRAWINGS

[0019] FIG. 1 is a block diagram of an example system consistent with present principles;

[0020] FIG. 2 is a block diagram of an example network of devices consistent with present principles;

[0021] FIG. 3 shows an example notification that may be presented while video has been

downgraded as a user looks offscreen consistent with present principles;

[0022] FIG. 4 shows an example notification that may be presented while the video is played out at optimal/maximum quality based on the user looking at the video consistent with present principles;

[0023] FIG. 5 illustrates example logic in example flow chart format that may be executed by a device consistent with present principles; and

[0024] FIG. 6 shows an example settings GUI that may be presented on a display to configure one or more settings of a device to operate consistent with present principles.

DETAILED DESCRIPTION

[0025] To save power, reduce heat buildup, and improve the functioning and life cycle of a computing device, present principles recognize that if a target user is not looking at a video streaming window during video playout, the device need not present the video in the highest quality resolution and fastest frame rate available since the user would not be watching the video anyway. But when the user looks back at the video stream again, the video stream may seamlessly switch back to the higher quality resolution and faster frame rate. Furthermore, processors that consume less power overall (such as a CPU big core or, for even greater power savings, a CPU little core) may also be used for video decoding when the user is not looking. Note that a given processor chip may include multiple big cores and multiple little cores.

[0026] Thus, incoming video streams such as online prerecorded videos and live video conferencing streams may be decoded/processed using a combination of CPU and GPU processing consistent with present principles. Indeed, many ARM (advanced RISC machines) processors and x86 processors have both big and little cores. Big cores may be performance-focused and can handle heavier tasks, while little cores are even more efficiency-oriented and can run background tasks while consuming considerably less energy than the CPU's larger core. Thus, present principles discuss, among other things, dynamically switching between which processor component handles video decoding to save system power and reduce heat generation where possible (e.g., while also reducing power consumption due to system fans not being needed or needed as much to cool system components, which in turn may also help reduce the amount of noise produced by the fans).

[0027] Specifically, systems and methods are disclosed below for detecting when a user is not looking at the video streaming window so that the device can adjust playback to conserve the device's battery or overall power consumption (e.g., from an AC wall outlet). There are several adjustments to playback that the device may make, which may be used alone or in combination.

[0028] For example, for streaming video, there are often multiple resolutions that are available for receipt. This is done for network bandwidth purposes, but a device operating consistent with present principles can make use of this availability to reduce power consumption by processing a lower resolution/higher compression video stream when the user is looking away.

[0029] Additionally, a device operating consistent with present principles might only decode/process the periodic I-frames of the video and skip over or not process the P-frames and B-frames of the video while the user is not looking. Implementation of this option may vary per video based on the framerate of the I-frames for the respective video. For example, if an I-frame is decoded only every 5 seconds, this might be too slow and result in what essentially would be a series of still images being presented on the display rather than a video. So in that case, P-frames and/or B-frames might also be decoded. As another example, if the intraframe (I-frame) rate is 2 frames per second (fps), a device operating consistent with present principles may skip every other intraframe or every Nth intraframe to further reduce power consumption.

[0030] What's more, as alluded to above, the processing/decoding of the periodic frames may be moved from the GPU to another processor, such as to the energy efficient little core of the CPU. Whether this is done and which CPU core might be assigned to handle the decoding may be determined on a video-by-video basis based on the resolution and framerate of whatever video is

being received as well as the capabilities of the respective CPU core itself. So if the downgraded video decoding is still a relatively high-load task but some power savings is desired, the CPU's larger core may be used, whereas for relatively lighter tasks and even greater power savings the CPU's smaller core may be used instead. In one specific but non-limiting example implementation, the device might begin by always moving decoding to the CPU little core to save the most power, then work backwards to figure out resolution, frame rate, etc. that the little core can handle and possibly move decoding to the CPU big core if appropriate. So, e.g., the device might want to do 640×480 at one frame per second and therefore determine if the little core can handle that processing. If it can, the little core may be used, but if not then decoding may be moved to the CPU big core. Thus, this may be a function of how long it takes the little core to decode an I frame, and whether the little core was doing anything else that might adversely affect decoding capability.

[0031] Or as another specific non-limiting example, the device may calculate what is needed in a given instance for decoding with the big and little CPU cores and then determine whether the little core could do the decoding based on the calculation and, if not, use the big core.

[0032] Then when the user returns or looks back at the screen, the device may switch video decoding back to the GPU for normal/optimal framerate and resolution processing and presentation. Given this, in some instances the video may be processed faster-than-real time from the last intraframe through all the prediction frames (P-frames and B-frames) to get to the current frame and resume normal playback (e.g., resume playback from a P-frame or B-frame in particular). As such, a buffer of past frames may be maintained, possibly at multiple video resolutions.

[0033] For example, while the user looks at the screen, the video may be GPU hardware-assisted decoded at 60 fps and ultra high definition (HD) resolution (3840×2160 resolution). When the user looks away, the video may be decoded on the CPU little core (e.g., using a software codec) at one fps and VGA resolution (640×480 resolution).

[0034] In terms of gaze detection, the device can detect if the user is not looking at the video streaming window by using human presence detection methods, including eye tracking using cameras as well as other methods. Note that the delay time/latency in gaze tracking may be short enough that the user's experience can be changed according to the foregoing frame rate and resolution adjustments in an almost imperceptible manner. Likewise, moving a decoding task between cores and/or codecs and/or GPUs can be done imperceptibly fast as well.

[0035] As for decoding itself, it is to be understood consistent with present principles that video decoding may include converting a compressed digital video file (encoded video) into a decompressed, playable format that can be presented by a device. During decoding, the compressed video data may therefore be decompressed and transformed back into its original format (and hence larger overall size), allowing the video to be presented in a playable form. As such, a video codec may be used to decode and decompress the received video data.

[0036] Prior to delving further into the details of the instant techniques, note with respect to any computer systems discussed herein that a system may include server and client components, connected over a network such that data may be exchanged between the client and server components. The client components may include one or more computing devices including televisions (e.g., smart TVs, Internet-enabled TVs), computers such as desktops, laptops and tablet computers, so-called convertible devices (e.g., having a tablet configuration and laptop configuration), and other mobile devices including smart phones. These client devices may employ, as non-limiting examples, operating systems from Apple Inc. of Cupertino CA, Google Inc. of Mountain View, CA, or Microsoft Corp. of Redmond, WA. A Unix® or similar such as Linux® operating system may be used, as may a Chrome or Android or Windows or macOS operating system. These operating systems can execute one or more browsers such as a browser made by Microsoft or Google or Mozilla or another browser program that can access web pages and applications hosted by Internet servers over a network such as the Internet, a local intranet, or a

virtual private network.

[0037] As used herein, instructions refer to computer-implemented steps for processing information in the system. Instructions can be implemented in software, firmware or hardware, or combinations thereof and include any type of programmed step undertaken by components of the system; hence, illustrative components, blocks, modules, circuits, and steps are sometimes set forth in terms of their functionality.

[0038] A processor may be any single- or multi-chip processor that can execute logic by means of various lines such as address lines, data lines, and control lines and registers and shift registers. Moreover, any logical blocks, modules, and circuits described herein can be implemented or performed with a system processor, a digital signal processor (DSP), a field programmable gate array (FPGA) or other programmable logic device such as an application specific integrated circuit (ASIC), discrete gate or transistor logic, discrete hardware components, or any combination thereof designed to perform the functions described herein. A processor can also be implemented by a controller or state machine or a combination of computing devices. Thus, the methods herein may be implemented as software instructions executed by a processor, suitably configured application specific integrated circuits (ASIC) or field programmable gate array (FPGA) modules, or any other convenient manner as would be appreciated by those skilled in the art. Where employed, the software instructions may also be embodied in a non-transitory device that is being vended and/or provided, and that is not a transitory, propagating signal and/or a signal per se. For instance, the non-transitory device may be or include a hard disk drive, solid state drive, or CD ROM. Flash drives may also be used for storing the instructions. Additionally, the software code instructions may also be downloaded over the Internet (e.g., as part of an application (“app”) or software file). Accordingly, it is to be understood that although a software application for undertaking present principles may be vended with a device such as the system **100** described below, such an application may also be downloaded from a server to a device over a network such as the Internet. An application can also run on a server and associated presentations may be displayed through a browser (and/or through a dedicated companion app) on a client device in communication with the server.

[0039] Software modules and/or applications described by way of flow charts and/or user interfaces herein can include various sub-routines, procedures, etc. Without limiting the disclosure, logic stated to be executed by a particular module can be redistributed to other software modules and/or combined together in a single module and/or made available in a shareable library. Also, the user interfaces (UI)/graphical UIs described herein may be consolidated and/or expanded, and UI elements may be mixed and matched between UIs.

[0040] Logic when implemented in software, can be written in an appropriate language such as but not limited to hypertext markup language (HTML)-5, Java®/JavaScript, C # or C++, and can be stored on or transmitted from a computer-readable storage medium such as a hard disk drive (HDD) or solid state drive (SSD), a random access memory (RAM), read-only memory (ROM), electrically erasable programmable read-only memory (EEPROM), a hard disk drive or solid state drive, compact disk read-only memory (CD-ROM) or other optical disk storage such as digital versatile disc (DVD), magnetic disk storage or other magnetic storage devices including removable thumb drives, etc.

[0041] In an example, a processor can access information over its input lines from data storage, such as the computer readable storage medium, and/or the processor can access information wirelessly from an Internet server by activating a wireless transceiver to send and receive data. Data typically is converted from analog signals to digital by circuitry between the antenna and the registers of the processor when being received and from digital to analog when being transmitted. The processor then processes the data through its shift registers to output calculated data on output lines, for presentation of the calculated data on the device.

[0042] Components included in one embodiment can be used in other embodiments in any

appropriate combination. For example, any of the various components described herein and/or depicted in the Figures may be combined, interchanged or excluded from other embodiments.

[0043] “A system having at least one of A, B, and C” (likewise “a system having at least one of A, B, or C” and “a system having at least one of A, B, C”) includes systems that have A alone, B alone, C alone, A and B together, A and C together, B and C together, and/or A, B, and C together, etc.

[0044] The term “circuit” or “circuitry” may be used in the summary, description, and/or claims. As is well known in the art, the term “circuitry” includes all levels of available integration, e.g., from discrete logic circuits to the highest level of circuit integration such as VLSI, and includes programmable logic components programmed to perform the functions of an embodiment as well as processors (e.g., special-purpose processors) programmed with instructions to perform those functions.

[0045] Now specifically in reference to FIG. 1, an example block diagram of an information handling system and/or computer system **100** is shown that is understood to have a housing for the components described below. Note that in some embodiments the system **100** may be a desktop computer system, such as one of the ThinkCentre® or ThinkPad® series of personal computers sold by Lenovo (US) Inc. of Morrisville, NC, or a workstation computer, such as the ThinkStation®, which are sold by Lenovo (US) Inc. of Morrisville, NC; however, as apparent from the description herein, a client device, a server or other machine in accordance with present principles may include other features or only some of the features of the system **100**. Also, the system **100** may be, e.g., a game console such as XBOX®, and/or the system **100** may include a mobile communication device such as a mobile telephone, notebook computer, and/or other portable computerized device.

[0046] As shown in FIG. 1, the system **100** may include a so-called chipset **110**. A chipset refers to a group of integrated circuits, or chips, that are designed to work together. Chipsets are usually marketed as a single product (e.g., consider chipsets marketed under the brands INTEL®, AMD®, etc.).

[0047] In the example of FIG. 1, the chipset **110** has a particular architecture, which may vary to some extent depending on brand or manufacturer. The architecture of the chipset **110** includes a core and memory control group **120** and an I/O controller hub **150** that exchange information (e.g., data, signals, commands, etc.) via, for example, a direct management interface or direct media interface (DMI) **142** or a link controller **144**. In the example of FIG. 1, the DMI **142** is a chip-to-chip interface (sometimes referred to as being a link between a “northbridge” and a “southbridge”).

[0048] The core and memory control group **120** includes a processor assembly **122** (e.g., one or more single core or multi-core processors, etc.) and a memory controller hub **126** that exchange information via a front side bus (FSB) **124**. A processor assembly such as the assembly **122** may therefore include one or more processors acting independently or in concert with each other to execute an algorithm, whether those processors are in one device or more than one device. Additionally, as described herein, various components of the core and memory control group **120** may be integrated onto a single processor die, for example, to make a chip that supplants the “northbridge” style architecture.

[0049] The memory controller hub **126** interfaces with memory **140**. For example, the memory controller hub **126** may provide support for DDR SDRAM memory (e.g., DDR, DDR2, DDR3, etc.). In general, the memory **140** is a type of random-access memory (RAM). It is often referred to as “system memory.”

[0050] The memory controller hub **126** can further include a low-voltage differential signaling interface (LVDS) **132**. The LVDS **132** may be a so-called LVDS Display Interface (LDI) for support of a display device **192** (e.g., a CRT, a flat panel, a projector, a touch-enabled light emitting diode (LED) display or other video display, etc.). A block **138** includes some examples of technologies that may be supported via the LVDS interface **132** (e.g., serial digital video,

HDMI/DVI, display port). The memory controller hub **126** also includes one or more PCI-express interfaces (PCI-E) **134**, for example, for support of discrete graphics **136**. Discrete graphics using a PCI-E interface has become an alternative approach to an accelerated graphics port (AGP). For example, the memory controller hub **126** may include a 16-lane (x16) PCI-E port for an external PCI-E-based graphics card (including, e.g., one or more GPUs). An example system may include AGP or PCI-E for support of graphics.

[0051] In examples in which it is used, the I/O hub controller **150** can include a variety of interfaces. The example of FIG. **1** includes a SATA interface **151**, one or more PCI-E interfaces **152** (optionally one or more legacy PCI interfaces), one or more universal serial bus (USB) interfaces **153**, a local area network (LAN) interface **154** (more generally a network interface for communication over at least one network such as the Internet, a WAN, a LAN, a Bluetooth network using Bluetooth 5.0 communication, etc. under direction of the processor(s) **122**), a general purpose I/O interface (GPIO) **155**, a low-pin count (LPC) interface **170**, a power management interface **161**, a clock generator interface **162**, an audio interface **163** (e.g., for speakers **194** to output audio), a total cost of operation (TCO) interface **164**, a system management bus interface (e.g., a multi-master serial computer bus interface) **165**, and a serial peripheral flash memory/controller interface (SPI Flash) **166**, which, in the example of FIG. **1**, includes basic input/output system (BIOS) **168** and boot code **190**. With respect to network connections, the I/O hub controller **150** may include integrated gigabit Ethernet controller lines multiplexed with a PCI-E interface port. Other network features may operate independent of a PCI-E interface. Example network connections include Wi-Fi as well as wide-area networks (WANs) such as 4G and 5G cellular networks.

[0052] The interfaces of the I/O hub controller **150** may provide for communication with various devices, networks, etc. For example, where used, the SATA interface **151** and/or PCI-E interface **152** provide for reading, writing or reading and writing information on one or more drives **180** such as HDDs, SSDs or a combination thereof, but in any case the drives **180** are understood to be, e.g., tangible computer readable storage mediums that are not transitory, propagating signals. The I/O hub controller **150** may also include an advanced host controller interface (AHCI) to support one or more drives **180**. The PCI-E interface **152** allows for wireless connections **182** to devices, networks, etc. The USB interface **153** provides for input devices **184** such as keyboards (KB), mice and various other devices (e.g., cameras, phones, storage, media players, etc.).

[0053] In the example of FIG. **1**, the LPC interface **170** provides for use of one or more ASICs **171**, a trusted platform module (TPM) **172**, a super I/O **173**, a firmware hub **174**, BIOS support **175** as well as various types of memory **176** such as ROM **177**, Flash **178**, and non-volatile RAM (NVRAM) **179**. With respect to the TPM **172**, this module may be in the form of a chip that can be used to authenticate software and hardware devices. For example, a TPM may be capable of performing platform authentication and may be used to verify that a system seeking access is the expected system.

[0054] The system **100**, upon power on, may be configured to execute boot code **190** for the BIOS **168**, as stored within the SPI Flash **166**, and thereafter processes data under the control of one or more operating systems and application software (e.g., stored in system memory **140**). An operating system may be stored in any of a variety of locations and accessed, for example, according to instructions of the BIOS **168**.

[0055] Additionally, the system **100** may also include one or more cameras **191** that gather one or more images and provide the images and related input to the processor assembly **122** (e.g., for eye tracking consistent with present principles). The camera may be a thermal imaging camera, an infrared (IR) camera, a digital camera such as a webcam, a three-dimensional (3D) camera, and/or a camera otherwise integrated into the system **100** and controllable by the processor assembly **122** to gather still images and/or video of a user's eyes during video consumption.

[0056] The system **100** may further include one or more human presence detection (HPD sensors) **193**, such as but not limited to a microphone (for detecting a user speaking and hence user

presence), an infrared (IR) sensor (for detecting heat and/or motion and hence user presence), keyboards and track pads and mice and touch-enabled displays (for detecting user input and hence user presence via input device use), and Bluetooth transceivers and other wireless transceivers (for detecting wireless signals from another device associated with the user and hence user presence). [0057] Additionally, though not shown for simplicity, in some embodiments the system **100** may include a gyroscope that senses and/or measures the orientation of the system **100** and provides related input to the processor assembly **122**, an accelerometer that senses acceleration and/or movement of the system **100** and provides related input to the processor assembly **122**, and/or a magnetometer that senses and/or measures directional movement of the system **100** and provides related input to the processor assembly **122**. Still further, the system **100** may include an audio receiver/microphone that provides input from the microphone to the processor assembly **122** based on audio that is detected, such as via a user providing audible input to the microphone.

[0058] Also, the system **100** may include a global positioning system (GPS) transceiver that is configured to communicate with satellites to receive/identify geographic position information and provide the geographic position information to the processor assembly **122**. However, it is to be understood that another suitable position receiver other than a GPS receiver may be used in accordance with present principles to determine the location of the system **100**.

[0059] It is to be understood that an example client device or other machine/computer may include fewer or more features than shown on the system **100** of FIG. 1. In any case, it is to be understood at least based on the foregoing that the system **100** is configured to undertake present principles.

[0060] Turning now to FIG. 2, example devices are shown communicating over a network **200** such as the Internet consistent with present principles (e.g., for video streaming from one device to another). It is to be understood that each of the devices described in reference to FIG. 2 may include at least some of the features, components, and/or elements of the system **100** described above. Indeed, any of the devices disclosed herein may include at least some of the features, components, and/or elements of the system **100** described above.

[0061] FIG. 2 shows a notebook computer and/or convertible computer **202**, a desktop computer **204**, a wearable device **206** such as a smart watch, a smart television (TV) **208**, a smart phone **210**, a tablet computer **212**, and a server **214** such as an Internet server that may provide cloud storage accessible to the devices **202-212**. It is to be understood that the devices **202-214** may be configured to communicate with each other over the network **200** to undertake present principles.

[0062] Referring now to FIG. 3, suppose an end-user is streaming video to his or her device, such as a smartphone, laptop computer, wearable device, computerized smart glasses, or even an augmented reality (AR) or virtual reality (VR) headset. The video might be from a video-sharing website or social media platform, for example. The video might also be streamed as part of a live video conference in which the end-user is participating, and as such the video may be real time video showing the faces of other conference participants. Other types of video may also be received and processed consistent with present principles.

[0063] In any case, FIG. 3 shows that a graphical user interface (GUI) **300** may be presented on the device's display. The GUI **300** itself may present the received and decoded video (designated as video **310** in FIG. 3) in real time as it is received and processed. However, note that in the present instance the video presentation has been downgraded to a lower resolution and slower frame rate owing to the device determining that the user is not looking at the at the device's display generally or even a particular display area that is actually presenting the video (e.g., where the video is presented in a non-maximized or non-full screen window). Additionally or alternatively, the device may offload video decoding from the GPU to a larger or smaller core of a CPU when the user is not looking, also resulting in a downgraded video presentation but one that consumes less power to present.

[0064] In addition to performing the foregoing actions in response to determining that the user is not looking at the display/video window, the device may present a graphical notification **320** over

top of the video **310**. The notification **320** may include a text indication **330** that the video has been downgraded. The notification **320** may also include a selector **340** that is selectable to command the device to stop downgrading the video presentation and to present the video **310** at an optimal/maximum available resolution and frame rate using the device's GPU. Therefore, the user might choose to select the selector **340** if the device's eye tracking results in a false positive where the user is not actually looking away but the device has determined as much. Or the user might choose to select the selector **340** in other situations as well, such as where more than one person is viewing the video **320** on the device's display and the device might only be tracking the gaze of one of the two users that happens to be looking offscreen (while the other user continues to view the video and hence would not wish it to be downgraded).

[0065] Turning to FIG. **4**, the GUI **300** is again presented on the display, which again includes presenting the video **310** itself as part of the GUI **300**. However, the notification **320** may be replaced by the notification **400** as shown at the top of the GUI **300** in FIG. **4**. The notification **400** may be presented in response to determining that the user is looking back at the device's display again. Accordingly, the notification **400** may include a text indication **410** that the video has been upgraded and/or is being presented at an optimal or best-available video quality again. The notification **400** may also include a selector **420** that is selectable to command the device to downgrade the video presentation consistent with present principles (e.g., process the video **310** using a slower frame rate and lower resolution on a CPU core rather than the GPU) even though the user might still be determined to be looking at the display/video window. The user might choose to select the selector **420** if, for example, the user is not paying particularly close attention to the video **310** and wishes to save power, if the user keeps glancing back and forth between the display and something else and hence does not desire optimal video quality, or if the user simply wants to reduce power consumption and heat generation during video playout regardless.

[0066] With the foregoing in mind, reference is now made to FIG. **5**. This figure shows example logic that may be executed by a first device such as the system **100**, processor assembly **122**, and/or client device of FIGS. **3** and **4** consistent with present principles. Note that while the logic of FIG. **5** is shown in flow chart format, other suitable logic may also be used. Also note that the first device may be a client device such as a smartphone, laptop computer, wearable device, headset, etc.

[0067] Beginning at block **500**, the first device may receive video from a second device different from the first device. The second device might be, for example, an Internet-based server or cloud platform hosting prerecorded videos uploaded by other end-users, such as a YouTube server. Or the second device may be a server of a video streaming service such as Netflix or Amazon Prime. The second device might also be a video conferencing server that routes audio video streams between participating client devices, or even another one of the conferencing client devices themselves. The second device might also be another client device outside the video conferencing context, such as another client device from which prerecorded video or a live, real time video stream is being received. Other implementations for the second device are encompassed by present principles as well.

[0068] From block **500** the logic may then proceed to block **510**. At block **510** the first device may receive input from one or more sensors such as a camera (e.g., the camera **191**) or other HPD sensor (including other HPD sensors **193** mentioned above). The logic may then proceed to block **520** where, if camera input is received, an eye tracking algorithm may be executed using the camera input to determine whether the user is looking at the first device's display (at decision diamond **530**).

[0069] Additionally or alternatively, at block **520** the first device may process other types of HPD sensor input that might be received at block **510** to determine (at diamond **530**) whether the user is looking at the first device's display. For example, if the HPD sensor is a microphone, audible input to the microphone may be processed to identify a person as speaking, in which case the person may

be assumed to be looking at the display. Or the microphone input may be processed using natural language understanding (NLU) to determine that the person that is speaking is speaking specifically about content associated with the video itself (e.g., as identified from video metadata or NLU being executed on the video's associated audio content), in which case the person may be assumed to be looking at the display. Additionally or alternatively, voice identification may be executed using the microphone input to identify a user that is speaking as being a particular user associated with the first device, in which case the person may be assumed to be looking at the display since the person is already a registered/recognized user.

[0070] As another example, input from an IR sensor may be used to detect heat and/or motion via the IR sensor input, in which case a user that is the source of the heat or motion may be assumed to be looking at the display. As yet another example, input received at connected input devices such as a keyboard, a touch-enabled laptop track pad, a mouse, and/or a touch-enabled display may be used to detect user presence owing to someone instigating the input in the first place, in which case the person may be assumed to be looking at the display.

[0071] As but one more non-limiting example, other HPD techniques also being encompassed by present principles, input from a Bluetooth transceiver and other wireless transceiver (e.g., ultrawideband transceiver, Wi-Fi transceiver, and/or near-field communication transceiver) may also be used to determine that another client device is present owing to wireless signals being received from it, in which case a nearby person may be assumed to be looking at the display owing to the presence of the other client device. In some specific examples, the wireless signals received from the other client device may include identifying information for the other client device itself, which may then be correlated to a particular user that is registered with the first device or that is otherwise recognizable as a past user of the first device (and hence the first device may assume that the recognized/registered user is looking at the display). Additionally or alternatively, a unique user identifier (UUID) from which the user might be similarly identified may be provided via the wireless signals and then used to assume the user is looking at the display based on the presence of the UUID in the signals.

[0072] Still further, in some examples a received signal strength indicator (RSSI) algorithm may be executed using the received wireless signals to not just identify a given user that might be registered or otherwise associated with the first device as being present somewhere around the first device, but to determine that the other client device itself is within a threshold non-zero distance of the first device. The threshold distance may be less than the maximum range that the respective wireless signals are capable of being transmitted, and may be an amount sufficient to infer not just user presence in a general area around the first device but actual user proximity to the first device to assume the user is close enough to potentially be looking at the first device's display. As such, the threshold distance may be ten feet or less when the first device is a laptop computer, smartphone, wearable device, or headset. If the first device is a television instead, the threshold distance might be more based on the recognition that people generally sit farther from a television than they might a laptop or smartphone display, and as such the threshold distance for a television implementation may be twenty feet.

[0073] Still in reference to FIG. 3, any of the eye tracking/HPD techniques mentioned above may therefore be used to determine whether the user is in fact looking at the display or not. Responsive to the first device determining that the user is not looking at the display, one or more of blocks 550-556 may then be executed to process the video a first way for presentation on the display. However, responsive to the first device determining that the user is actually looking at the display, one or more of blocks 560-564 may instead be executed to process the video a second way for presentation on the display.

[0074] Beginning with blocks 550 and 560, the first device may process the video for presentation on the display at a first resolution at block 550, whereas at block 560 the first device may instead process the video for presentation on the display at a second resolution that is higher than the first

resolution. For example, while the user looks at the display, the video may be decoded at ultra-high definition (UHD) resolution (3840×2160 resolution) at block **560**, whereas while the user looks away from the display (or is not present at all) the video may be decoded at VGA resolution (640×480 resolution) at block **550**. Accordingly, for example, in streaming video there are often multiple resolutions that are available for streaming/receipt. The first device may therefore make use of this availability to reduce power consumption by requesting and then processing a lower resolution/higher compression video stream that is available while the user is looking away from the display, and then switching to requesting and processing a higher resolution/lower compression video stream that is available while the user is looking back at the display again.

[0075] Turning to blocks **552** and **562**, the first device may process the video for presentation on the display at a first frame rate at block **552**, whereas at block **562** the first device may process the video for presentation on the display at a second frame rate that is faster than the first frame rate. For example, while the user looks at the display, the video may be decoded at 60 fps, whereas while the user looks away from the display the video may be decoded at one fps at block **562**.

[0076] Still in terms of frame rate and providing another example, at block **552** the device may decline to process first non-I-frames of the video for presentation on the display (e.g., decline to process P-frames and B-frames) since the user is not looking at the display at that point. However, at block **562** based on the user being determined as looking at the display, the first device may instead process the first non-I-frames of the video for presentation on the display. The first non-I-frames may include all available/received non-I-frames.

[0077] Note that I-frames are the least compressible frames of the three, but generally do not require other video frames to decode them (e.g., they are fully-renderable on the display on their own accord without data from prior or later frames being used to render a complete image). In contrast, P-frames may use data from previous frames for decompression and rendering of the P-frames (and are more compressible than I-frames). B-frames may use both previous and later frames for data reference to get a higher amount of data compression for subsequent decompression and rendering by the first device.

[0078] As for blocks **554** and **564**, the first device may process the video for presentation on the display using a CPU larger or smaller core, whereas at block **564** the first device may process the video for presentation on the display using a relatively higher-performance GPU instead. For example, while the user looks at the display, the video may be decoded using the GPU, whereas while the user looks away from the display the video may be decoded using the CPU's larger “big” core to reduce power consumption and heat generation while still maintaining a relatively high level of performance. Or for even greater power savings and heat reduction, the CPU's smaller “little” core may be used instead of the GPU. Either way, the CPU might still use a software codec for decoding, for example.

[0079] Note that from block **564** the logic may proceed directly to block **570**, as will be described shortly. However, in terms of block **554**, note that the logic may proceed therefrom to block **556** before ultimately proceeding to block **570**.

[0080] At block **556** and still based on determining that the user is not looking at the display, the first device may maintain a buffer of a threshold amount of past video frames while processing the video the first way(s). The buffer may be maintained in random access memory (RAM) of the first device or at another storage location, such as a persistent solid-state drive (SSD) or hard disk drive (HDD) storage location. In any case, the buffering of past images may be performed to process the images faster-than-real time from a last/most-recent past I-frame and through all the ensuing prediction frames (P-frames and B-frames) that might also be past frames to get to the current frame's rendering and then resume normal playback even without any intervening additional I-frames being received and/or processed (e.g., resume playback from a current P-frame or B-frame). Thus, the threshold amount of past video frames that are maintained in the buffer may include all P-frames and B-frames subsequent to a most recently-received I-frame (the most recently-received

I-frame also being maintained in the buffer until a next I-frame in the I-frame sequence is received). Thus, a reference (most-recent) I-frame and intervening P-frames and B-frames may accumulate in the buffer and then removed/deleted from the buffer responsive to another I-frame being received (since the past P-frames and B-frames in reference to the past I-frame would no longer be needed for image rendering owing to a subsequent I-frame has been received).

[0081] Additionally, in one particular example implementation, the first device may maintain the buffer of the threshold amount of past video frames not just in one resolution but in multiple resolutions for each respective past video frame at block **556**. This may be done so that a highest-available resolution for the frames may be dynamically selected and used on the fly notwithstanding any potential network bandwidth issues or other constraints, with each respective I-frame, B-frame, and/or P-frame as buffered at multiple resolutions already being cached in local storage for real time playout without delay.

[0082] Now describing the aforementioned block **570**, note that it may be arrived at via block **556** or block **564**. At block **570** the first device may return to decision diamond **530** to again determine whether the user is looking at the display during playout of a later portion of the same video. Thus, the process of FIG. 5 may continually repeat during video playout to adjust playback back and forth between different resolutions, frame rates, and decoding processors, enabling the first device to be more energy-efficient on the fly where possible.

[0083] Thus, in one specific example implementation consistent with the logic of FIG. 5, when the first device detects that the user is not looking at the display, the first device can move the video decoding from the high power GPU to the lower power CPU little core. Recognizing that the CPU little core might not have as much performance as the GPU might, in this case the first device may reduce resolution, reduce frame rate, and skip B-frames, P frames, and even I frames to conform to the performance factor of the CPU little core when moving the video decoding thereto.

[0084] Also note with respect to the logic of FIG. 5 that should the user override the current handling of video decoding by the GPU, or CPU big or little core, according to user input such as described above with respect to FIGS. 3 and 4, then the user's command would override the logic above and the device would operate in accordance with the user's command.

[0085] What's more, note that in instances where decoding is offloaded to the CPU big or little core, the GPU may be turned off or otherwise deenergized during CPU decoding to save additional power. Then when the user is determined to look back at the display again, the GPU may be powered back on and used for video decoding.

[0086] Turning now to FIG. 6, another example GUI **600** is shown that may be presented on the display of a client device configured to undertake present principles. The GUI **600** may be presented to set or enable one or more settings of the first device (e.g., its guest operating system) or a particular video streaming app stored at the first device (that might itself execute the logic of FIG. 5). As such, the GUI **600** may be navigated to through a device or app menu, for example. Also note that each of the example options discussed below may be selected by directing touch or cursor input to the associated check box adjacent to the respective option.

[0087] As shown in FIG. 6, the GUI **600** may include an option **610** that may be selected to set or configure the device/app to undertake present principles. Thus, option **610** may be selected a single time to set or enable the device/app to, in multiple future instances of video streaming (e.g., for different discrete videos/video files), execute the logic of FIG. 5 and present the notifications of FIGS. 3 and 4. Thus, option **610** may be selected to enable dynamic video decoding/processing using user gaze to save power and reduce heat generation that might otherwise reduce the overall life of different components within the first device.

[0088] The GUI **600** may also include an option **620**. The option **620** may be selected to set or enable the device to specifically not execute dynamic video processing based on user gaze when two or more people are determined to be at least present if not looking directly at the first device's display. This might be done using camera input as well as computer vision and/or facial recognition

to thus identify the number of people shown in the camera input. As another example, microphone input may be processed to identify two different voices as being detected to therefore infer that two or more people are present. IR sensors and other types of HPD sensors may also be used for determining as much. Thus, options **610** and **620** might both be concurrently selected in one specific example so that video presentation may be dynamically downgraded when a single user that is present is determined to be looking away from the display, but also so that video presentation is not downgraded when more than one user is present so that optimal video quality is utilized even if one of the users happens to look away and the device detects as much.

[0089] Moving on from FIG. 6, as alluded to above, in some examples a device operating consistent with present principles may not just determine whether a user is generally looking at a display or not. Instead, the device may specifically determine whether the user is looking at a particular display area or window at which the video content itself is presented, where that area or window may be less than an entire area of the display on which images are presentable. Thus, if the user was viewing the video in a relatively small window on the display while also working on a word processing document, responding to emails, or viewing other content that is concurrently presented on the display, the device may present the video according to the first way(s) of FIG. 5 while the user is determined to be looking at the other content and then present the video according to the second way(s) of FIG. 5 when the user switches back to looking at the video itself.

[0090] As another example, if the user had another window maximized in the foreground as the active window, and had the video's window minimized or at least in the background as a non-active window (e.g., even if still partially or fully presented on some portion of the display), the device may present the video according to the first way(s) of FIG. 5 while the other window is the active window in the foreground and then present the video according to the second way(s) of FIG. 5 when the user switches back to the video window being the active window in the foreground.

[0091] It may now be appreciated that present principles provide for an improved computer-based user interface that increases the functionality and overall life of the devices disclosed herein while also making the devices more energy-efficient. The disclosed concepts are rooted in computer technology for computers to carry out their functions.

[0092] Components included in one embodiment can be used in other embodiments in any appropriate combination. For example, any of the various components described herein and/or depicted in the Figures may be combined, interchanged or excluded from other embodiments.

[0093] It is to be understood that whilst present principals have been described with reference to some example embodiments, these are not intended to be limiting, and that various alternative arrangements may be used to implement the subject matter claimed herein. Accordingly, while particular techniques and devices are herein shown and described in detail, it is to be understood that the subject matter which is encompassed by the present application is limited only by the claims.

Claims

1. A first device, comprising: a processor assembly; a display accessible to the processor assembly; and storage accessible to the processor assembly and comprising instructions executable by the processor assembly to: receive video from a second device different from the first device; determine whether a user is looking at the display; based on a determination that the user is not looking at the display, process the video a first way for presentation on the display; and based on a determination that the user is looking at the display, process the video a second way for presentation on the display, the second way being different from the first way.
2. The first device of claim 1, wherein the first way comprises processing the video for presentation on the display at a first resolution, and wherein the second way comprises processing the video for presentation on the display at a second resolution, the second resolution being higher than the first

resolution.

3. The first device of claim 1, wherein the first way comprises processing the video for presentation on the display at a first frame rate, and wherein the second way comprises processing the video for presentation on the display at a second frame rate, the second frame rate being faster than the first frame rate.
4. The first device of claim 1, wherein the first way comprises processing the video for presentation on the display using a central processing unit (CPU), and wherein the second way comprises processing the video for presentation on the display using a graphics processing unit (GPU), the GPU being different from the CPU.
5. The first device of claim 4, wherein the first way comprises processing the video for presentation on the display using a larger core of two cores on the CPU.
6. The first device of claim 4, wherein the first way comprises processing the video for presentation on the display using a smaller core of at least two cores on the CPU.
7. The first device of claim 1, wherein: the first way comprises each of: processing the video for presentation on the display at a first resolution, processing the video for presentation on the display at a first frame rate, and processing the video for presentation on the display using a central processing unit (CPU); and the second way comprises each of: processing the video for presentation on the display at a second resolution, processing the video for presentation on the display at a second frame rate, and processing the video for presentation on the display using a graphics processing unit (GPU); wherein the second resolution is higher than the first resolution, the second frame rate is faster than the first frame rate, and the GPU is different from the CPU.
8. The first device of claim 1, wherein the instructions are executable to: receive input from a camera; and execute eye tracking using the input from the camera to determine whether the user is looking at the display.
9. The first device of claim 8, comprising the camera.
10. The first device of claim 1, wherein the instructions are executable to: receive input from a human presence detection (HPD) sensor; and determine whether the user is looking at the display using the input from the HPD sensor.
11. The first device of claim 10, comprising the HPD sensor.
12. The first device of claim 1, wherein the instructions are executable to: while processing the video the first way, maintain a buffer of a threshold amount of past video frames.
13. The first device of claim 1, wherein the instructions are executable to: while processing the video the first way, maintain a buffer of a threshold amount of past video frames in multiple resolutions for each respective past video frame.
14. A method, comprising: receiving, at a first device, video from a second device different from the first device; determining whether a user is looking at a display on which the video is to be presented; based on determining that the user is not looking at the display, processing the video a first way at the first device for presentation on the display; and based on determining that the user is looking at the display, processing the video a second way at the first device for presentation on the display, the second way being different from the first way.
15. The method of claim 14, wherein the first way comprises declining to process first non-I-frames of the video for presentation on the display, and wherein the second way comprises processing the first non-I-frames of the video for presentation on the display.
16. The method of claim 15, wherein the first non-I-frames comprise P-frames of the video.
17. The method of claim 15, wherein the first non-I-frames comprise B-frames of the video.
18. At least one computer readable storage medium (CRSM) that is not a transitory signal, the at least one CRSM comprising instructions executable by a processor assembly to: receive, at a first device, video from a second device different from the first device; determine whether a user is looking at a display; based on a determination that the user is not looking at the display, process the video a first way for presentation on the display; and based on a determination that the user is

looking at the display, process the video a second way for presentation on the display, the second way being different from the first way.

19. The at least one CRSM of claim 18, wherein the first way comprises declining to process first non-I-frames of the video for presentation on the display, and wherein the second way comprises processing the first non-I-frames of the video for presentation on the display.

20. The at least one CRSM of claim 18, wherein the first way comprises processing the video for presentation on the display using a central processing unit (CPU), and wherein the second way comprises processing the video for presentation on the display using a graphics processing unit (GPU), the GPU being different from the CPU.
