



(51) International Patent Classification:

H04N 19/15 (2014.01) H04N 19/174 (2014.01)
H04N 19/132 (2014.01)

(21) International Application Number:

PCT/IB2015/000225

(22) International Filing Date:

16 January 2015 (16.01.2015)

(25) Filing Language:

English

(26) Publication Language:

English

(71) Applicant: INTEL CORPORATION [US/US]; 2200
Mission College Boulevard, Santa Clara, California 95054
(US).

(72) Inventors: GOLIKOVA, Olga; 30 Turgenev Str., Nizhny
Novgorod, 603024 (RU). FROLOV, Pavel; 17 Krylat-
skaya Str., Bldg 4, Moscow, 121614 (RU). FADEEV,
Alexey; 30 Turgeneva Str., Nizhny Novgorod, 603163
(RU).

AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY,
BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM,
DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT,
HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR,
KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG,
MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM,
PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC,
SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN,
TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

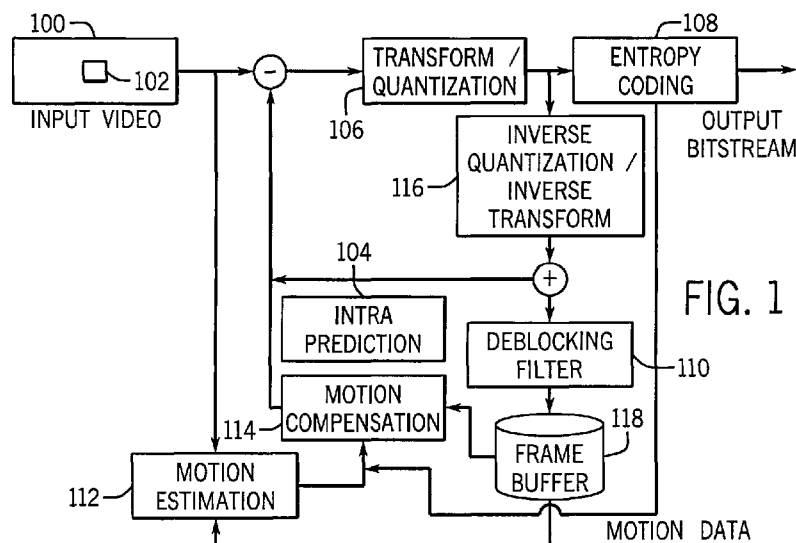
(84) Designated States (unless otherwise indicated, for every
kind of regional protection available): ARIPO (BW, GH,
GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ,
TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU,
TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE,
DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU,
LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK,
SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ,
GW, KM, ML, MR, NE, SN, TD, TG).

Published:

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

(81) Designated States (unless otherwise indicated, for every
kind of national protection available): AE, AG, AL, AM,

(54) Title: ENCODER SLICE SIZE CONTROL WITH COST ESTIMATION



(57) Abstract: The video motion estimation (VME) unit of a graphics processor performs fast preprocessing prior to frame encoding to distribute macroblocks among slices based on weights of the macroblocks. The result of the preprocessing stage is iteratively improved based on the encoding stage. This allows real-time slice size limited encoding with minimum video quality reduction in some embodiments. Hardware preprocessing can be replaced with a software or hardware encoding step that gives macroblock weights (for example macroblock sizes after real encoding).

ENCODER SLICE SIZE CONTROL WITH COST ESTIMATION

Background

[0001] This is related to video distribution over a network including video conferencing. Video transmission uses equally sized data packets, each packet containing a complete video decodable unit. A minimal decodable unit for modern compression formats is a slice. A slice is a spatially distinct region of a frame that is encoded separately from any other region of the same frame. But hardware encoders usually operate at the granularity of a frame for performance reasons.

[0002] If the hardware encoder is not designed to support a slice size limitation then it is not possible to break the frame into slices, because macroblock (MB) distribution in slices is predefined in advance for the frame. Without slice size control (when hardware encoder is a black-box for software user), some compressed video slices may exceed the slice size limit.

[0003] The encoded slice size must not exceed a defined level, but splitting a frame into slices reduces encoding efficiency and requires a higher bit rate for the same visual quality. This visual quality can be achieved by iteratively re-encoding the frame with different slice partitioning until the required slice size reached.

Brief Description Of The Drawings

[0004] Some embodiments are described with respect to the following figures:

Figure 1 is a block diagram for one embodiment;

Figure 2 is a flow chart for one embodiment;

Figure 3 is a system depiction for one embodiment; and

Figure 4 is a front elevational view for one embodiment.

Detailed Description

[0005] Referring to Figure 1, an exemplary encoding algorithm pursuant to the H.264/AVC standard is depicted. See ISO/IEC International Standard 14496-10 (MPEG-4 part 10 2/14) Advanced Video Coding available from ITU Telecommunication Standardization Section (ITU-T). The input video frame 100 is

partitioned into blocks 102. Each block undergoes intra prediction 104 using neighboring reconstructed pixels. A prediction residue from intra prediction undergoes transformation 106 to decorrelate the data. Context adaptive variable length coding (CAVLC) may be employed to entropy code 108 scanned transformed coefficients. Alternatively, the transformed coefficients may be coded by context adaptive binary arithmetic coding (CABAC). To mitigate blocking artifacts, an adaptive in-loop deblocking filter 110 may be applied to the reconstruction from the feedback loop. The video motion estimation unit 112 determines the positions of motion vectors of moving objects within the picture 100. That motion data is then used by the motion compensation stage 114. The frequency transform block 106 transforms a macroblock into the spatial frequency domain. Then the frequency domain data is quantized. The quantization takes into account the range of data values for the current macroblock. An inverse frequency transform block 116 transforms the frequency domain data from the inverse quantization stage back to the spatial domain. The quantized data is encoded by entropy coding block 108 that employs fewer bits to encode more frequently used symbols and more bits to encode less frequently used symbols.

[0006] The video motion estimation (VME) unit of a graphics processor performs fast preprocessing prior to frame encoding to distribute macroblocks among slices based on weights of the macroblocks. The result of the preprocessing stage is iteratively improved based on the encoding stage. This allows real-time slice size limited encoding with minimum video quality reduction in some embodiments. Hardware preprocessing can be replaced with a software or hardware encoding step that gives macroblock weights (for example macroblock sizes after real encoding).

[0007] An application estimates a maximum number of iterations to guarantee real time encoding. For example, if encoding one frame takes 4 ms., and one frame cost estimation with decoding (or VME) takes less than 2 ms., then no more than 7 recoding iterations can be used and still achieve real time encoding.

[0008] The application can recode a frame using stages as shown in Figure 2, until all encoded frame sizes are less than a defined level.

[0009] The application uses at least 3 stages: “trial run”, “recoding 0” and “skip frame” to provide the slice size limitation. Increasing the number of stages can improve quality, but decreases performance.

[0010] The application can have more than six stages. In that case some stages can be repeated. For example: “trial run”, “recoding 0”, “recoding 1”, “forced mode 0”, “forced mode 0”, “skip frame”.

[0011] Referring to Figure 2, preprocessing 10 to estimate slice sizes based on macroblock statistics can be accomplished with a hardware (VME) unit, a hardware PAK unit or by software encoding, to mention three examples. The macroblock statistics may include a macroblock complexity measure or cost distribution over the frame, plus the same data from previously encoded frames. Macroblock complexity may be estimated using cost functions such as sum of absolute differences (SAD), sum of squared differences (SSD), rate distortion (RD) cost, sum of absolute transformed differences (SATD), Haar cost function (used in VME) or encoded macroblock sizes.

[0012] As an alternative, the frame can be encoded and decoded. Coded macroblock lengths can be used as macroblock cost.

[0013] Preprocessing takes original and reference frames with required coding parameters as inputs and outputs estimated macroblock assignments to the equally sized slices. The parameters may be encoder initialization parameters such as bit rate, frame rate, or group of pictures structure to mention examples. The number of slices in the frame (**numSlices**) is predicted from previous frame statistics such as macroblock sizes and macroblock costs from a previous frame. For example the initial number of slices may be set to be the same as in the previous frame. In one embodiment, when using real time encoding for preprocessing:

- 1) Encode the frame with an equal number of macroblocks in each slice.
- 2) Calculate slice sizes and macroblock sizes by using a decoder
(slice_size[i], $i \in [0, \text{numSlice})$; MBSIZE[j], $j \in [0, \text{numMBInFrame})$)

When using a VME unit call for the preprocessing stage:

- a. Call "VME" with equal number of macroblocks in each slice.
 - b. Set the macroblock size equal to VME macroblock distortions or costs. The costs result from a VME function run. Then the slice size is set to equal to the sum of the macroblock sizes for all blocks in the slice: $MBSize[j] = MB_dist[j]$, $j \in [0, numMBInFrame)$
- 3) Make equally sized slices according to the following condition (which means pushing macroblocks to the slice until the slice does not exceed average slice size):

$$\sum_{k=0}^{LastMBInSlice[i]} MBSize[k] < avg < \sum_{k=0}^{LastMBInSlice[i]+1} MBSize[k]$$

$$\text{where } avg \text{ is } \frac{\sum_{i=0}^{LastMBInFrame} MBSize[i]}{numSlices}; \quad i \in [0; numSlices - 1),$$

so then the sum of macroblock costs in a slice is equal to the average cost.

- 4) Encode the frame with this slice partitioning and calculate real slice sizes ($slice_size[i]$, $i \in [0, numSlices]$)
- 5) If any slice size exceeds a defined level or if slices are too short (diamond 12, NO path), go to next stage. Criteria for short slices:

$$numSlice > \frac{FrameSize * 2}{MaxSliceSize} + 1.$$

Otherwise (diamond 12, YES path) exit at 15.

[0014] Recoding 0 (block 14) is intended to redistribute macroblocks between slices to form slices with equal summary macroblock weight (as indicated at (3) below). The macroblock's weight controls its importance to the subjective appearance of the image. It gets slice sizes from the previous stage and makes a new macroblock per slice distribution:

- 1) Set **MaxSliceSize** as desired/set by user and **MaxSliceWeight=90.0%** of MaxSizeSize as defined by an algorithm to reduce the probability of recoding. The weight is determined to balance between speed (in terms of number of iterations) and quality (in terms of slice size).
- 2) Calculate weight in percent for each slice:

$$SliceWeight[i] = \frac{sliceSize[i] * 100.0}{MaxSliceSize}$$

- 3) Calculate weight in percent for each macroblock:

$$MBWeight[j] = \frac{MBSIZE[j] * SliceWeight[i]}{\sum_{k=FirstMBInSlice[i]}^{LastMBInSlice[i]} MBSIZE[k]}$$

- 4) Make equally weighted slices according to the following condition (which means pushing macroblocks to the slice till it is not exceed **MaxSliceWeight**):

$$\sum_{k=FirstMBInSlice[i]}^{LastMBInSlice[i]} MBWeight[k] < MaxSliceWeight; \quad i \in [0; numSlices1 - 1]$$

- 5) Encode frame and calculate new slice sizes. If the size of any slice exceeds defined level (diamond 16, NO path), go to the next stage. Otherwise (diamond 16, YES path) go to exit 15.

[0015] Recoding 1 (block 18) corrects size of slices ("big slices") that still exceed the required limit:

- 1) After "Recoding0", most of slices meet the maximum size limitation. "Big slices" are divided into two parts using macroblock weight.

6

$$weight = \frac{\sum_{k=FirstMBInSlice[i]}^{LastMBInSlice[i]} MBSize[k]}{2}$$

$$\sum_{k=FirstMBInSlice[i]}^{LastMBInSlice[new]} MBSize[k] \leq weight$$

- 2) Encode frame and calculate slice sizes. If any slice size exceeds defined level (diamond 20, NO path), go to next stage. Otherwise (diamond 20, YES path) go to exit 15.

[0016] Forced mode0 (block 22) is used if there are still “big slices” without the bandwidth or possibility to divide them further. The quantization parameter (QP) (that defines the quantization level that in turn affects visual quality and bit rate of each frame) is increased in combination with the Recoding 1 mode to reduce the size of such slices. If all sizes are “OK” (diamond 24, YES path) go to exit 15. Otherwise (diamond 24, NO path) proceed to forced mode 1.

[0017] Forced mode1 (block 26) is similar to Recoding0 stage with a QP increase. This mode can be used after a few “Forced mode0” calls, especially when number of slices is limited. It rearranges macroblocks in slices that do not belong to “big slices” (reduced by quantization increasing in “forced mode”) to reduce number on macroblocks in a “big slice.” If all sizes are “OK” (diamond 28, YES path) go to exit 15. Otherwise (diamond 28, NO path) proceed to skip frame (block 30).

[0018] In case the required slice size is still not achieved, skip frames (block 30) can be used in a skip mode. One can also skip a slice if hardware supported. This mode guarantees meeting the slice size limitation. Encoded skip frame takes a number of bits (near zero) that is less than MaxSliceSize, because MaxSliceSize must have a reasonable value (e.g. no less than 100 bytes).

[0019] In some embodiments, the sequence shown in Figure 2 may be implemented in software, firmware and/or hardware. In software and firmware embodiments it may be implemented by computer executed instructions stored in one or more non-transitory

computer readable media such as magnetic, optical or semiconductor storages. In some embodiments the sequence may be implemented by instructions stored within a graphics processor or within hardware that is part of a graphics processor. For example, the instructions or the hardware may be part of a video motion estimation unit of a graphics processor and particularly with respect to Figure 1 may be part of the motion estimation block 112 in some embodiments.

[0020] Figure 3 illustrates an embodiment of a system 700. In embodiments, system 700 may be a media system although system 700 is not limited to this context. For example, system 700 may be incorporated into a personal computer (PC), laptop computer, ultra-laptop computer, tablet, touch pad, portable computer, handheld computer, palmtop computer, personal digital assistant (PDA), cellular telephone, combination cellular telephone/PDA, television, smart device (e.g., smart phone, smart tablet or smart television), mobile internet device (MID), messaging device, data communication device, and so forth.

[0021] In embodiments, system 700 comprises a platform 702 coupled to a display 720. Platform 702 may receive content from a content device such as content services device(s) 730 or content delivery device(s) 740 or other similar content sources. A navigation controller 750 comprising one or more navigation features may be used to interact with, for example, platform 702 and/or display 720. Each of these components is described in more detail below.

[0022] In embodiments, platform 702 may comprise any combination of a chipset 705, processor 710, memory 712, storage 714, graphics subsystem 715, applications 716 and/or radio 718. Chipset 705 may provide intercommunication among processor 710, memory 712, storage 714, graphics subsystem 715, applications 716 and/or radio 718. For example, chipset 705 may include a storage adapter (not depicted) capable of providing intercommunication with storage 714.

[0023] Processor 710 may be implemented as Complex Instruction Set Computer (CISC) or Reduced Instruction Set Computer (RISC) processors, x86 instruction set compatible processors, multi-core, or any other microprocessor or central processing unit (CPU). In embodiments, processor 710 may comprise dual-core processor(s),

dual-core mobile processor(s), and so forth. The processor may implement the sequence of Figure 2 together with memory 712.

[0024] Memory 712 may be implemented as a volatile memory device such as, but not limited to, a Random Access Memory (RAM), Dynamic Random Access Memory (DRAM), or Static RAM (SRAM).

[0025] Storage 714 may be implemented as a non-volatile storage device such as, but not limited to, a magnetic disk drive, optical disk drive, tape drive, an internal storage device, an attached storage device, flash memory, battery backed-up SDRAM (synchronous DRAM), and/or a network accessible storage device. In embodiments, storage 714 may comprise technology to increase the storage performance enhanced protection for valuable digital media when multiple hard drives are included, for example.

[0026] Graphics subsystem 715 may perform processing of images such as still or video for display. Graphics subsystem 715 may be a graphics processing unit (GPU) or a visual processing unit (VPU), for example. An analog or digital interface may be used to communicatively couple graphics subsystem 715 and display 720. For example, the interface may be any of a High-Definition Multimedia Interface, DisplayPort, wireless HDMI, and/or wireless HD compliant techniques. Graphics subsystem 715 could be integrated into processor 710 or chipset 705. Graphics subsystem 715 could be a stand-alone card communicatively coupled to chipset 705.

[0027] The graphics and/or video processing techniques described herein may be implemented in various hardware architectures. For example, graphics and/or video functionality may be integrated within a chipset. Alternatively, a discrete graphics and/or video processor may be used. As still another embodiment, the graphics and/or video functions may be implemented by a general purpose processor, including a multi-core processor. In a further embodiment, the functions may be implemented in a consumer electronics device.

[0028] Radio 718 may include one or more radios capable of transmitting and receiving signals using various suitable wireless communications techniques. Such techniques may involve communications across one or more wireless networks.

Exemplary wireless networks include (but are not limited to) wireless local area networks (WLANs), wireless personal area networks (WPANs), wireless metropolitan area network (WMANs), cellular networks, and satellite networks. In communicating across such networks, radio 718 may operate in accordance with one or more applicable standards in any version.

[0029] In embodiments, display 720 may comprise any television type monitor or display. Display 720 may comprise, for example, a computer display screen, touch screen display, video monitor, television-like device, and/or a television. Display 720 may be digital and/or analog. In embodiments, display 720 may be a holographic display. Also, display 720 may be a transparent surface that may receive a visual projection. Such projections may convey various forms of information, images, and/or objects. For example, such projections may be a visual overlay for a mobile augmented reality (MAR) application. Under the control of one or more software applications 716, platform 702 may display user interface 722 on display 720.

[0030] In embodiments, content services device(s) 730 may be hosted by any national, international and/or independent service and thus accessible to platform 702 via the Internet, for example. Content services device(s) 730 may be coupled to platform 702 and/or to display 720. Platform 702 and/or content services device(s) 730 may be coupled to a network 760 to communicate (e.g., send and/or receive) media information to and from network 760. Content delivery device(s) 740 also may be coupled to platform 702 and/or to display 720.

[0031] In embodiments, content services device(s) 730 may comprise a cable television box, personal computer, network, telephone, Internet enabled devices or appliance capable of delivering digital information and/or content, and any other similar device capable of unidirectionally or bidirectionally communicating content between content providers and platform 702 and/display 720, via network 760 or directly. It will be appreciated that the content may be communicated unidirectionally and/or bidirectionally to and from any one of the components in system 700 and a content provider via network 760. Examples of content may include any media information including, for example, video, music, medical and gaming information, and so forth.

[0032] Content services device(s) 730 receives content such as cable television programming including media information, digital information, and/or other content. Examples of content providers may include any cable or satellite television or radio or Internet content providers. The provided examples are not meant to limit the applicable embodiments.

[0033] In embodiments, platform 702 may receive control signals from navigation controller 750 having one or more navigation features. The navigation features of controller 750 may be used to interact with user interface 722, for example. In embodiments, navigation controller 750 may be a pointing device that may be a computer hardware component (specifically human interface device) that allows a user to input spatial (e.g., continuous and multi-dimensional) data into a computer. Many systems such as graphical user interfaces (GUI), and televisions and monitors allow the user to control and provide data to the computer or television using physical gestures.

[0034] Movements of the navigation features of controller 750 may be echoed on a display (e.g., display 720) by movements of a pointer, cursor, focus ring, or other visual indicators displayed on the display. For example, under the control of software applications 716, the navigation features located on navigation controller 750 may be mapped to virtual navigation features displayed on user interface 722, for example. In embodiments, controller 750 may not be a separate component but integrated into platform 702 and/or display 720. Embodiments, however, are not limited to the elements or in the context shown or described herein.

[0035] In embodiments, drivers (not shown) may comprise technology to enable users to instantly turn on and off platform 702 like a television with the touch of a button after initial boot-up, when enabled, for example. Program logic may allow platform 702 to stream content to media adaptors or other content services device(s) 730 or content delivery device(s) 740 when the platform is turned "off." In addition, chip set 705 may comprise hardware and/or software support for 5.1 surround sound audio and/or high definition 7.1 surround sound audio, for example. Drivers may include a graphics driver for integrated graphics platforms. In embodiments, the graphics driver may comprise a peripheral component interconnect (PCI) Express graphics card.

[0036] In various embodiments, any one or more of the components shown in system 700 may be integrated. For example, platform 702 and content services device(s) 730 may be integrated, or platform 702 and content delivery device(s) 740 may be integrated, or platform 702, content services device(s) 730, and content delivery device(s) 740 may be integrated, for example. In various embodiments, platform 702 and display 720 may be an integrated unit. Display 720 and content service device(s) 730 may be integrated, or display 720 and content delivery device(s) 740 may be integrated, for example. These examples are not meant to be scope limiting.

[0037] In various embodiments, system 700 may be implemented as a wireless system, a wired system, or a combination of both. When implemented as a wireless system, system 700 may include components and interfaces suitable for communicating over a wireless shared media, such as one or more antennas, transmitters, receivers, transceivers, amplifiers, filters, control logic, and so forth. An example of wireless shared media may include portions of a wireless spectrum, such as the RF spectrum and so forth. When implemented as a wired system, system 700 may include components and interfaces suitable for communicating over wired communications media, such as input/output (I/O) adapters, physical connectors to connect the I/O adapter with a corresponding wired communications medium, a network interface card (NIC), disc controller, video controller, audio controller, and so forth. Examples of wired communications media may include a wire, cable, metal leads, printed circuit board (PCB), backplane, switch fabric, semiconductor material, twisted-pair wire, co-axial cable, fiber optics, and so forth.

[0038] Platform 702 may establish one or more logical or physical channels to communicate information. The information may include media information and control information. Media information may refer to any data representing content meant for a user. Examples of content may include, for example, data from a voice conversation, videoconference, streaming video, electronic mail ("email") message, voice mail message, alphanumeric symbols, graphics, image, video, text and so forth. Data from a voice conversation may be, for example, speech information, silence periods, background noise, comfort noise, tones and so forth. Control information may refer to any data representing commands, instructions or control words meant for an automated

system. For example, control information may be used to route media information through a system, or instruct a node to process the media information in a predetermined manner. The embodiments, however, are not limited to the elements or in the context shown or described in Figure 3.

[0039] As described above, system 700 may be embodied in varying physical styles or form factors. Figure 4 illustrates embodiments of a small form factor device 800 in which system 700 may be embodied. In embodiments, for example, device 800 may be implemented as a mobile computing device having wireless capabilities. A mobile computing device may refer to any device having a processing system and a mobile power source or supply, such as one or more batteries, for example.

[0040] As shown in Figure 4, device 800 may comprise a housing 802, a display 804 and 810, an input/output (I/O) device 806, and an antenna 808. Device 800 also may comprise navigation features 812. Display 804 may comprise any suitable display unit for displaying information appropriate for a mobile computing device. I/O device 806 may comprise any suitable I/O device for entering information into a mobile computing device. Examples for I/O device 806 may include an alphanumeric keyboard, a numeric keypad, a touch pad, input keys, buttons, switches, rocker switches, microphones, speakers, voice recognition device and software, and so forth. Information also may be entered into device 800 by way of microphone. Such information may be digitized by a voice recognition device. The embodiments are not limited in this context.

[0041] As described above, examples of a mobile computing device may include a personal computer (PC), laptop computer, ultra-laptop computer, tablet, touch pad, portable computer, handheld computer, palmtop computer, personal digital assistant (PDA), cellular telephone, combination cellular telephone/PDA, television, smart device (e.g., smart phone, smart tablet or smart television), mobile internet device (MID), messaging device, data communication device, and so forth.

[0042] Examples of a mobile computing device also may include computers that are arranged to be worn by a person, such as a wrist computer, finger computer, ring computer, eyeglass computer, belt-clip computer, arm-band computer, shoe

computers, clothing computers, and other wearable computers. In embodiments, for example, a mobile computing device may be implemented as a smart phone capable of executing computer applications, as well as voice communications and/or data communications. Although some embodiments may be described with a mobile computing device implemented as a smart phone by way of example, it may be appreciated that other embodiments may be implemented using other wireless mobile computing devices as well. The embodiments are not limited in this context.

[0043] The following clauses and/or examples pertain to further embodiments:

One example embodiment may be a computer-implemented method comprising estimating slice sizes based on macroblock statistics, assigning macroblocks to equally sized slices until each slice does not exceed a pre-identified size, redistributing macroblocks between slices to reduce slice size, and if a slice size is still greater than the predefined size, inserting skip frames. The method may also include predicting a number of slices for a frame using statistics from previous frames. The method may also include correcting sizes of slices that still exceed said predefined size by dividing said slices into parts. The method may also include calculating a weight in percent of each macroblock based on macroblock size, sum of macroblock size and slice weight. The method may also include reducing slice size by increasing a quantization parameter. The method may also include rearranging macroblock assignments to a slice based on the increased quantization parameter. The method may also include iteratively reducing slice sizes. The method may also include estimating a number of iterations that guarantee real time encoding. The method may also include determining whether slices are too short. The method may also include making the sum of macroblock weights in a slice less than a maximum slice weight.

[0044] Another example may be one or more non-transitory computer readable media storing instructions for execution by a processor to perform a sequence comprising estimating slice sizes based on macroblock statistics, assigning macroblocks to equally sized slices until each slice does not exceed a pre-identified size, redistributing macroblocks between slices to reduce slice size, and if a slice size is still greater than the predefined size, inserting skip frames. The media may include said sequence including predicting a number of slices for a frame using statistics from previous frames.

The media may include said sequence including correcting sizes of slices that still exceed said predefined size by dividing said slices into parts. The media may include said sequence including calculating a weight in percent of each macroblock based on macroblock size, sum of macroblock size and slice weight. The media may include said sequence including reducing slice size by increasing a quantization parameter. The media may include said sequence including rearranging macroblock assignments to a slice based on the increased quantization parameter. The media may include said sequence including iteratively reducing slice sizes. The media may include said sequence including estimating a number of iterations that guarantee real time encoding. The media may include said sequence including determining whether slices are too short. The media may include said sequence including making the sum of macroblock weights in a slice less than a maximum slice weight.

[0045] In another example embodiment may be an apparatus comprising a processor to estimate slice sizes based on macroblock statistics, assign macroblocks to equally sized slices until each slice does not exceed a pre-identified size, redistribute macroblocks between slices to reduce slice size, and if a slice size is still greater than the predefined size, insert skip frames, and a storage coupled to said processor. The apparatus may include said processor to predict a number of slices for a frame using statistics from previous frames. The apparatus may include said processor to correct sizes of slices that still exceed said predefined size by dividing said slices into parts. The apparatus may include said processor to calculate a weight in percent of each macroblock based on macroblock size, sum of macroblock size and slice weight. The apparatus may include said processor to reduce slice size by increasing a quantization parameter. The apparatus may include said processor to rearrange macroblock assignments to a slice based on the increased quantization parameter. The apparatus may include said processor to iteratively reduce slice sizes. The apparatus may include a display communicatively coupled to the circuit. The apparatus may include a battery coupled to the circuit. The apparatus may include firmware and a module to update said firmware.

[0046] The graphics processing techniques described herein may be implemented in various hardware architectures. For example, graphics functionality may be integrated

within a chipset. Alternatively, a discrete graphics processor may be used. As still another embodiment, the graphics functions may be implemented by a general purpose processor, including a multicore processor.

[0047] References throughout this specification to “one embodiment” or “an embodiment” mean that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one implementation encompassed within the present disclosure. Thus, appearances of the phrase “one embodiment” or “in an embodiment” are not necessarily referring to the same embodiment. Furthermore, the particular features, structures, or characteristics may be instituted in other suitable forms other than the particular embodiment illustrated and all such forms may be encompassed within the claims of the present application.

[0048] While a limited number of embodiments have been described, those skilled in the art will appreciate numerous modifications and variations therefrom. It is intended that the appended claims cover all such modifications and variations as fall within the true spirit and scope of this disclosure.

What is claimed is:

- 1 1. A computer-implemented method comprising:
2 estimating slice sizes based on macroblock statistics;
3 assigning macroblocks to equally sized slices until each slice does not exceed
4 a pre-identified size;
5 redistributing macroblocks between slices to reduce slice size; and
6 if a slice size is still greater than the predefined size, inserting skip frames.
- 1 2. The method of claim 1 including predicting a number of slices for a frame
2 using statistics from previous frames.
- 1 3. The method of claim 1 including correcting sizes of slices that still exceed said
2 predefined size by dividing said slices into parts.
- 1 4. The method of claim 1 including calculating a weight in percent of each
2 macroblock based on macroblock size, sum of macroblock size and slice weight.
- 1 5. The method of claim 1 including reducing slice size by increasing a
2 quantization parameter.
- 1 6. The method of claim 5 including rearranging macroblock assignments to a
2 slice based on the increased quantization parameter.
- 1 7. The method of claim 1 including iteratively reducing slice sizes.
- 1 8. The method of claim 7 including estimating a number of iterations that
2 guarantee real time encoding.
- 1 9. The method of claim 1 including determining whether slices are too short.

1 10. The method of claim 1 including making the sum of macroblock weights in a
2 slice less than a maximum slice weight.

1 11. One or more non-transitory computer readable media storing instructions for
2 execution by a processor to perform a sequence comprising:
3 estimating slice sizes based on macroblock statistics;
4 assigning macroblocks to equally sized slices until each slice does not exceed
5 a pre-identified size;
6 redistributing macroblocks between slices to reduce slice size; and
7 if a slice size is still greater than the predefined size, inserting skip frames.

1 12. The media of claim 11, said sequence including predicting a number of slices
2 for a frame using statistics from previous frames.

1 13. The media of claim 11, said sequence including correcting sizes of slices that
2 still exceed said predefined size by dividing said slices into parts.

1 14. The media of claim 11, said sequence including calculating a weight in
2 percent of each macroblock based on macroblock size, sum of macroblock size and
3 slice weight.

1 15. The media of claim 11, said sequence including reducing slice size by
2 increasing a quantization parameter.

1 16. The media of claim 15, said sequence including rearranging macroblock
2 assignments to a slice based on the increased quantization parameter.

1 17. The media of claim 11, said sequence including iteratively reducing slice
2 sizes.

1 18. The media of claim 17, said sequence including estimating a number of
2 iterations that guarantee real time encoding.

- 1 19. An apparatus comprising:
2 a processor to estimate slice sizes based on macroblock statistics, assign
3 macroblocks to equally sized slices until each slice does not exceed a pre-identified
4 size, redistribute macroblocks between slices to reduce slice size, and if a slice size
5 is still greater than the predefined size, insert skip frames; and
6 a storage coupled to said processor.
- 1 20. The apparatus of claim 19, said processor to predict a number of slices for a
2 frame using statistics from previous frames.
- 1 21. The apparatus of claim 19, said processor to correct sizes of slices that still
2 exceed said predefined size by dividing said slices into parts.
- 1 22. The apparatus of claim 19, said processor to calculate a weight in percent of
2 each macroblock based on macroblock size, sum of macroblock size and slice
3 weight.
- 1 23. The apparatus of claim 19, said processor to reduce slice size by increasing a
2 quantization parameter.
- 1 24. The apparatus of claim 23, said processor to rearrange macroblock
2 assignments to a slice based on the increased quantization parameter.
- 1 25. The apparatus of claim 19, said processor to iteratively reduce slice sizes.

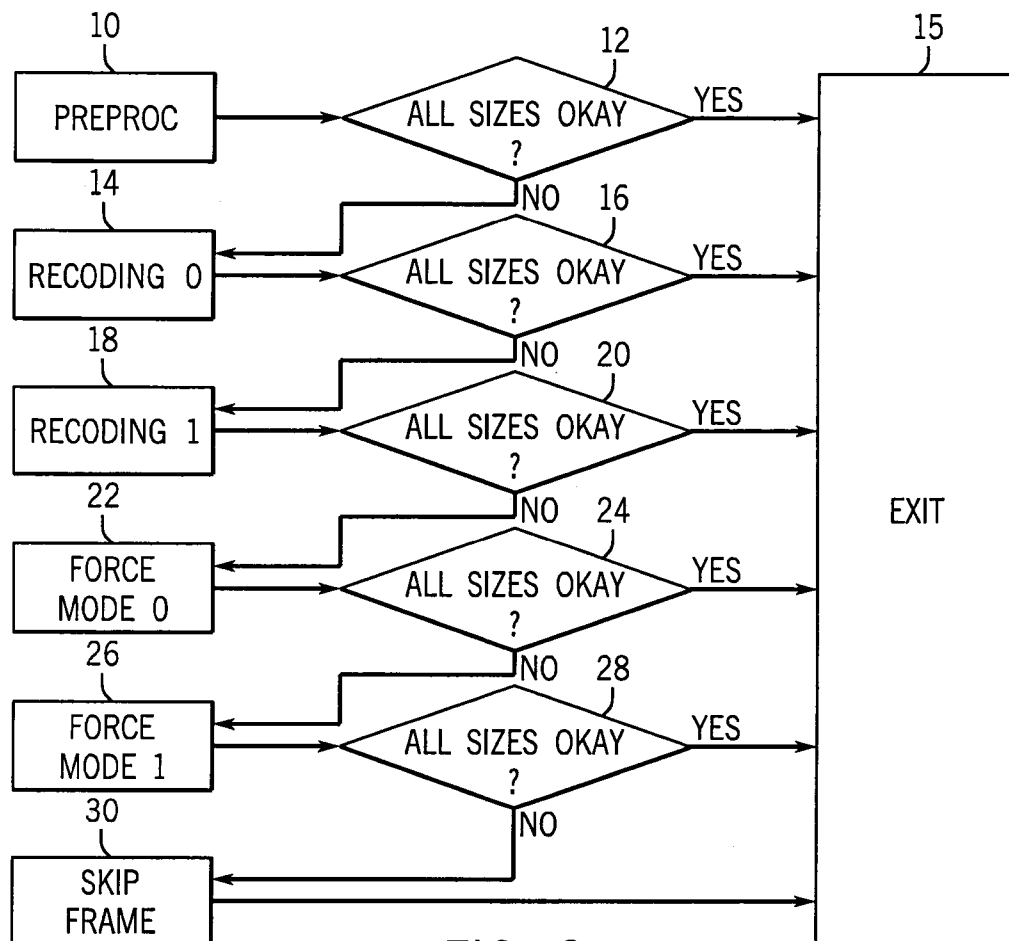
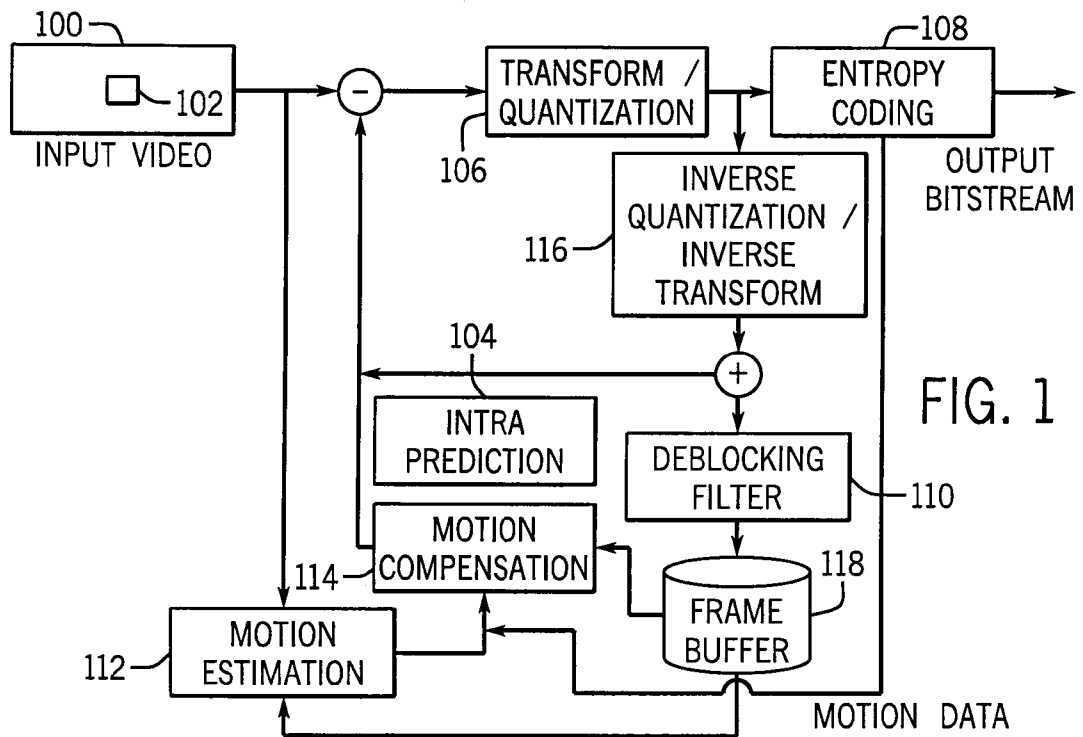


FIG. 2

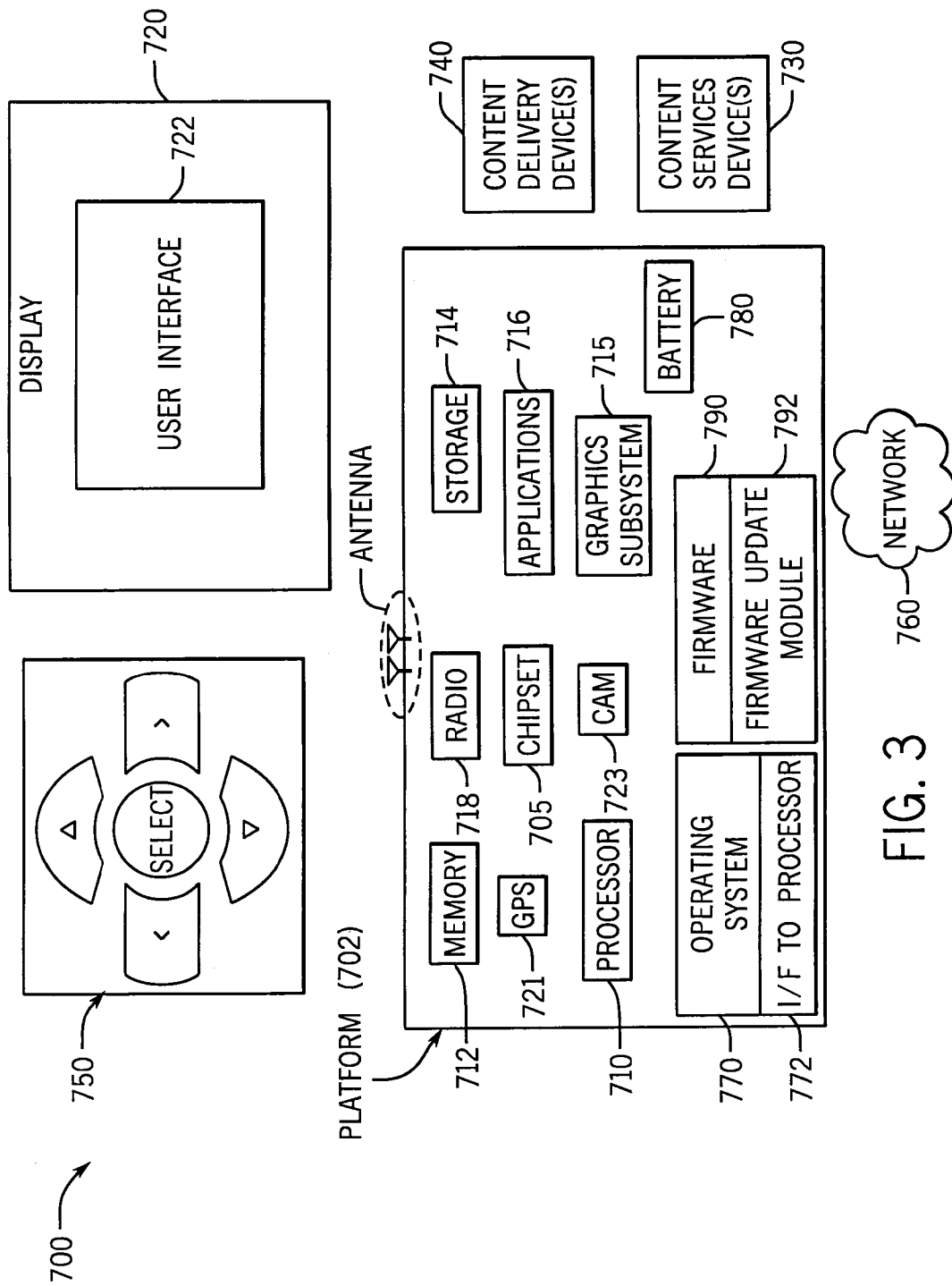


FIG. 3

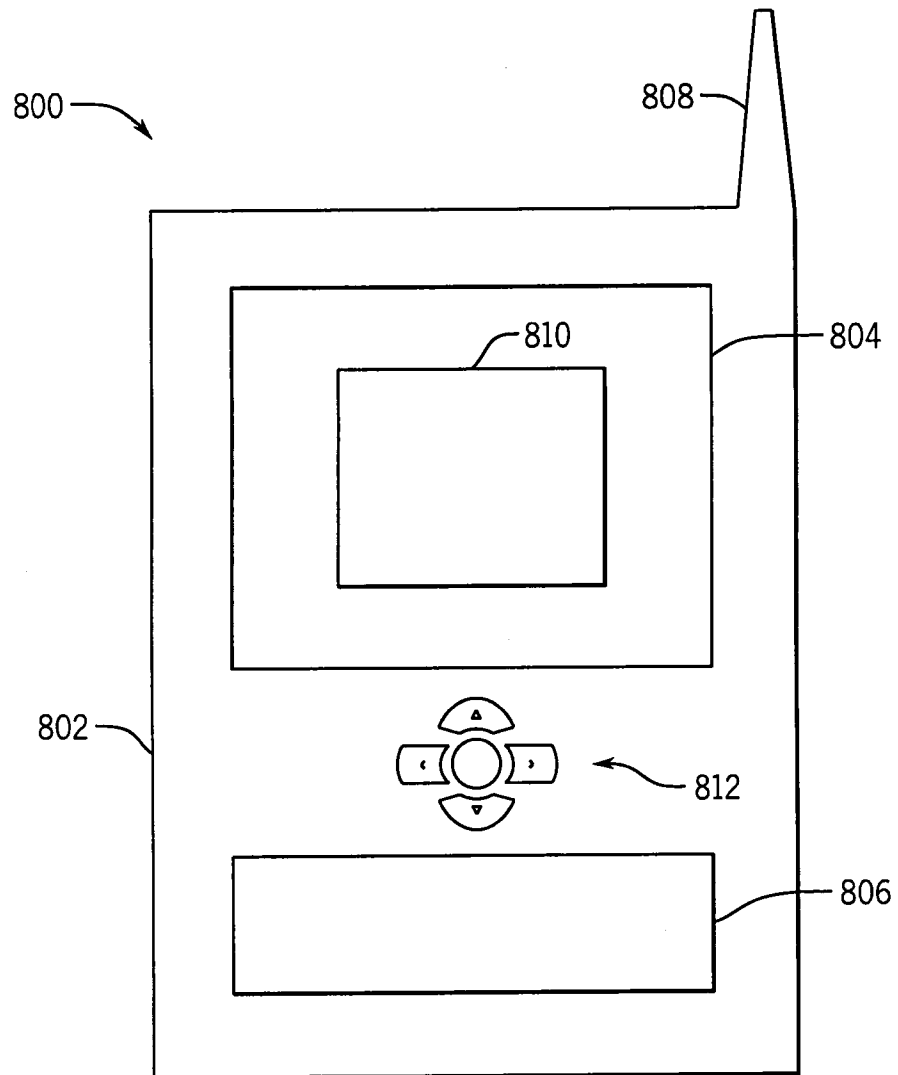


FIG. 4

INTERNATIONAL SEARCH REPORT

International application No
PCT/IB2015/000225

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04N19/15 H04N19/132 H04N19/174
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

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

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2012/281768 A1 (MATSUBA YASUTOMO [US] ET AL) 8 November 2012 (2012-11-08) paragraphs [0068] - [0127] -----	1-25
Y	US 2012/300836 A1 (SHIMIZU MINAKO [JP] ET AL) 29 November 2012 (2012-11-29) paragraph [0029]; figure 5 -----	1-25



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

11 May 2015

Date of mailing of the international search report

19/05/2015

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Di Cagno, Gianluca

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/IB2015/000225

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2012281768	A1	08-11-2012	NONE

US 2012300836	A1	29-11-2012	JP 5396302 B2 22-01-2014
			JP 2011166370 A 25-08-2011
			US 2012300836 A1 29-11-2012
			WO 2011096173 A1 11-08-2011
