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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
17 October 2019 (17.10.2019)

(10) International Publication Number
WO 2019/199838 A1

(51) International Patent Classification:
H04N 19/115 (2014.01) *H04N 19/48* (2014.01)
H04N 19/12 (2014.01) *H04N 19/60* (2014.01)
H04N 19/18 (2014.01) *H04N 19/88* (2014.01)

(21) International Application Number: PCT/US2019/026596

(22) International Filing Date: 09 April 2019 (09.04.2019)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/656,488 12 April 2018 (12.04.2018) US

(71) Applicant: FUTUREWEI TECHNOLOGIES, INC. [US/US]; 5700 Tennyson Parkway, Suite 600, Plano, Texas 75024 (US).

(72) Inventors: AUYEUNG, Cheung; 1459 Flamingo Way, Sunnyvale, California 94087 (US). CHEN, Jianle; 10756 Corte De Tiburon, San Diego, California 92130 (US).

(74) Agent: DIETRICH, William H. et al.; Conley Rose, P.C., 5601 Granite Parkway, Suite 500, Plano, Texas 75024 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, 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).

(54) Title: REDUCING CONTEXT SWITCHING FOR CODING TRANSFORM COEFFICIENTS

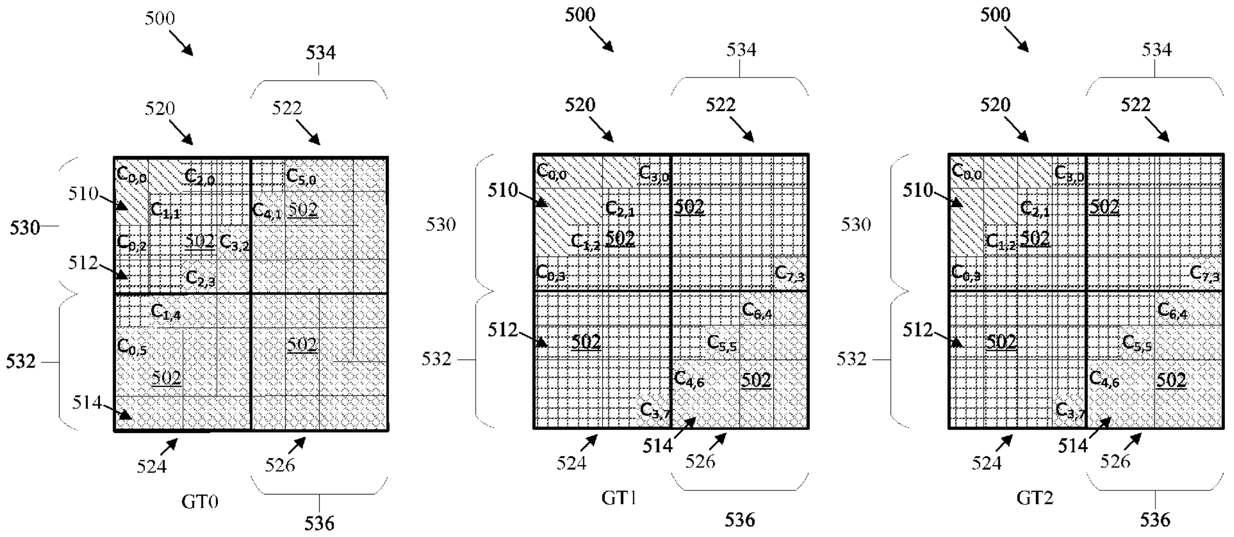

FIG. 5

(57) Abstract: A method of context modeling implemented by a coding device. The method includes selecting a context index group from a plurality of context index groups based on a location of a transform coefficient when a coefficient group contains an origin transform coefficient, and using a single available context index group, which is independent of the location of the transform coefficient, when the coefficient group does not contain the origin transform coefficient. The method further includes selecting a context index from the context group that was selected or from the single available context group being used, and coding using the context index that was selected.

WO 2019/199838 A1

[Continued on next page]

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

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

Reducing Context Switching for Coding Transform Coefficients

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This patent application claims the benefit of U.S. Provisional Patent Application No. 62/656,488, filed April 12, 2018, by Cheung Auyeung, et al., and titled "Reducing Context Switching for Coding Transform Coefficients," which is hereby incorporated by reference.

TECHNICAL FIELD

[0002] The present disclosure is generally related to coding transform coefficients, and is specifically related to coding transform coefficients using context switching.

BACKGROUND

[0003] The amount of video data needed to depict even a relatively short video can be substantial, which may result in difficulties when the data is to be streamed or otherwise communicated across a communications network with limited bandwidth capacity. Thus, video data is generally compressed before being communicated across modern day telecommunications networks. The size of a video could also be an issue when the video is stored on a storage device because memory resources may be limited. Video compression devices often use software and/or hardware at the source to code the video data prior to transmission or storage, thereby decreasing the quantity of data needed to represent digital video images. The compressed data is then received at the destination by a video decompression device that decodes the video data. With limited network resources and ever increasing demands of higher video quality, improved compression and decompression techniques that improve compression ratio with little to no sacrifice in image quality are desirable.

SUMMARY

[0004] A first aspect relates to a method of context modeling implemented by a coding device. The method includes selecting, by the coding device, a context index group from a plurality of context index groups based on a location of a transform coefficient when a coefficient group contains the transform coefficient contains an origin transform coefficient; using, by the coding device, a single available context index group when the coefficient group contains the transform coefficient does not contain the origin transform coefficient;

selecting, by the coding device, a context index from the context group that was selected or from the single available context group being used; and coding, by the coding device, using the context index that was selected.

[0005] A second aspect relates to a method of context modeling implemented by a coding device. The method includes selecting, by the coding device, a context index group from a plurality of context index groups based on a location of a transform coefficient when a coefficient group containing the plurality of the transform coefficients intersects with a transform sub-block containing an origin transform coefficient; and using, by the coding device, a single available context index group when the coefficient group contains the transform coefficient does not contain the origin transform coefficient; selecting, by the coding device, a context index from the context group that was selected or from the single available context group being used; and coding, by the coding device, using the context index that was selected.

[0006] The techniques and/or methods limit the number/amount of position-based context index switching (e.g., between contest index groups) while maintaining or improving coding efficiency. As will be more fully explained below, the present disclosure restricts or limits context switching to the coefficient group that intersects with the sub-block that contains an origin transform coefficient. Because there is no position-based context index switching in other coefficient groups, computational complexity and power consumption of a coding device are reduced. This improves the overall performance of the coding device relative to other coding devices without the benefit of the coding techniques and/or methods disclosed herein.

[0007] In a first implementation form of the method according to the first or second aspect as such, the origin transform coefficient is disposed at position $C_{0,0}$ in a transform block.

[0008] In a second implementation form of the method according to the first or second aspect as such or any preceding implementation form of the first or second aspect, the coefficient group contains transform coefficients of a sub-block of a transform block.

[0009] In a third implementation form of the method according to the first or second aspect as such or any preceding implementation form of the first or second aspect, the transform coefficients in the sub-block of the transform block that contains the origin transform coefficient have been divided into regions.

[0010] In a fourth implementation form of the method according to the first or second aspect as such or any preceding implementation form of the first or second aspect, the regions for a *significant_coeff_flag (GT0)*, a *coeff_abs_greater1 (GT1)*, and a *coeff_abs_greater2 (GT2)* are the same.

[0011] In a fifth implementation form of the method according to the first or second aspect as such or any preceding implementation form of the first or second aspect, the *significant_coeff_flag (GT0)* is encoded in a first pass, and the *coeff_abs_greater1 (GT1)*, the *coeff_abs_greater2 (GT2)*, and a *coeff_abs_level_remaining* are coded in a second pass.

[0012] In a sixth implementation form of the method according to the first or second aspect as such or any preceding implementation form of the first or second aspect, the *significant_coeff_flag (GT0)*, the *coeff_abs_greater1 (GT1)*, and the *coeff_abs_greater2 (GT2)* are encoded in different passes.

[0013] In a seventh implementation form of the method according to the first or second aspect as such or any preceding implementation form of the first or second aspect, the the transform coefficients for the coefficient group are separated into multiple scan passes for encoding the *significant_coeff_flag (GT0)*, the *coeff_abs_greater1 (GT1)*, the *coeff_abs_greater2 (GT2), ...*, and a *coeff_abs_greaterN (GTN)* for a pre-determined integer *N*.

[0014] In an eighth implementation form of the method according to the first or second aspect as such or any preceding implementation form of the first or second aspect, each of the regions corresponds to one of the plurality of context index groups.

[0015] In a ninth implementation form of the method according to the first or second aspect as such or any preceding implementation form of the first or second aspect, the regions are separated by diagonal lines.

[0016] In a tenth implementation form of the method according to the first or second aspect as such or any preceding implementation form of the first or second aspect, the diagonal lines comprise a first diagonal line (d_0) defined by the formula $d_0 = x + y$ and a second diagonal line (d_1) defined by the formula $d_1 = x + y$, where x and y are coordinates of the transform block.

[0017] In an eleventh implementation form of the method according to the first or second aspect as such or any preceding implementation form of the first or second aspect, wherein sub-blocks of the transform block corresponding to coefficient groups that do not contain the origin transform coefficient corresponding to the single available context index group.

[0018] In a twelfth implementation form of the method according to the first or second aspect as such or any preceding implementation form of the first or second aspect, the plurality of context index groups contain a first context index group, a second context index group, and a third context index group.

[0019] In a thirteenth implementation form of the method according to the first or second aspect as such or any preceding implementation form of the first or second aspect, the first context index group comprises context indexes 12-17, the second context index group comprises context indexes 6-11, and the third context index group comprises context indexes 0-5.

[0020] In a fourteenth implementation form of the method according to the first or second aspect as such or any preceding implementation form of the first or second aspect, the transform block is an 4nx4m block, n>1 or m>1, and the sub-block is a 4x4 sub-block.

[0021] In a fifteenth implementation form of the method according to the first or second aspect as such or any preceding implementation form of the first or second aspect, the coefficient group includes at least one non-zero transform coefficient.

[0022] In a sixteenth implementation form of the method according to the first or second aspect as such or any preceding implementation form of the first or second aspect, the coding device is an encoder.

[0023] In a seventeenth implementation form of the method according to the first or second aspect as such or any preceding implementation form of the first or second aspect, the coding device is a decoder.

[0024] A third aspect relates to a coding apparatus that includes a receiver configured to receive a picture to encode or to receive a bitstream to decode, a transmitter coupled to the receiver, the transmitter configured to transmit the bitstream to a decoder or to transmit a decoded image to a display, a memory coupled to at least one of the receiver or the transmitter, the memory configured to store instructions, and a processor coupled to the memory, the processor configured to execute the instructions stored in the memory to perform the method of any of the preceding aspects or implementations.

[0025] The coding apparatus limits the number/amount of position-based context index switching (e.g., between contest index groups) while maintaining or improving coding efficiency. As will be more fully explained below, the present disclosure restricts or limits context switching to the coefficient group that intersects with the sub-block that contains an origin transform coefficient. Because there is no position-based context index switching in other coefficient groups, computational complexity and power consumption of a coding device are reduced. This improves the overall performance of the coding device relative to other coding devices without the benefit of the coding techniques and/or methods disclosed herein.

[0026] In a first implementation form of the apparatus according to the third aspect as such, the apparatus further includes a display configured to display an image.

[0027] A fourth aspect relates to a coding apparatus including a transform unit configured to select a context index group from a plurality of context index groups based on a location of a transform coefficient when a coefficient group containing a plurality of the transform coefficients contains an origin transform coefficient; and select a context index from the context group that was selected or from the context group being used; and a coding unit coupled to the transform unit, the coding unit configured to code using the context index that was selected.

[0028] The coding apparatus limits the number/amount of position-based context index switching (e.g., between contest index groups) while maintaining or improving coding efficiency. As will be more fully explained below, the present disclosure restricts or limits context switching to the coefficient group that intersects with the sub-block that contains an origin transform coefficient. Because there is no position-based context index switching in other coefficient groups, computational complexity and power consumption of a coding device are reduced. This improves the overall performance of the coding device relative to other coding devices without the benefit of the coding techniques and/or methods disclosed herein.

[0029] In a first implementation form of the apparatus according to the fourth aspect as such, the transform unit comprises a transform processing unit or an inverse transform unit.

[0030] In a second implementation form of the apparatus according to the fourth aspect as such, the coding unit comprises an entropy coding unit or an entropy decoding unit.

[0031] A fifth aspect relates to a system that includes an encoder and a decoder in communication with the encoder. The encoder or the decoder includes the coding apparatus of any of the preceding aspects or implementations.

[0032] The system limits the number/amount of position-based context index switching (e.g., between contest index groups) while maintaining or improving coding efficiency. As will be more fully explained below, the present disclosure restricts or limits context switching to the coefficient group that intersects with the sub-block that contains an origin transform coefficient. Because there is no position-based context index switching in other coefficient groups, computational complexity and power consumption are reduced. This improves the overall performance of the system relative to other systems without the benefit of the coding techniques and/or methods disclosed herein.

[0033] A sixth aspect relates to a means for coding that includes receiving means configured to receive a picture to encode or to receive a bitstream to decode, transmission means coupled to the receiving means, the transmission means configured to transmit the bitstream to a decoder or to transmit a decoded image to a display means, storage means coupled to at least one of the receiving means or the transmission means, the storage means configured to store instructions, and processing means coupled to the storage means, the processing means configured to execute the instructions stored in the storage means to perform the methods in any of the preceding aspects or implementations.

[0034] The means for coding limits the number/amount of position-based context index switching (e.g., between contest index groups) while maintaining or improving coding efficiency. As will be more fully explained below, the present disclosure restricts or limits context switching to the coefficient group that intersects with the sub-block that contains an origin transform coefficient. Because there is no position-based context index switching in other coefficient groups, computational complexity and power consumption of the means for coding are reduced. This improves the overall performance of the means for coding relative to other means for coding without the benefit of the coding techniques and/or methods disclosed herein.

[0035] For the purpose of clarity, any one of the foregoing embodiments may be combined with any one or more of the other foregoing embodiments to create a new embodiment within the scope of the present disclosure.

[0036] These and other features will be more clearly understood from the following detailed description taken in conjunction with the accompanying drawings and claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0037] For a more complete understanding of this disclosure, reference is now made to the following brief description, taken in connection with the accompanying drawings and detailed description, wherein like reference numerals represent like parts.

[0038] FIG. 1 is a block diagram illustrating an example coding system that may utilize context modeling techniques.

[0039] FIG. 2 a block diagram illustrating an example video encoder that may implement context modeling techniques.

[0040] FIG. 3 a block diagram illustrating an example video decoder that may implement context modeling techniques.

[0041] FIG. 4 is a schematic diagram of a coding block containing a plurality of transform coefficients.

[0042] FIG. 5 is a schematic diagram of three coding blocks containing a plurality of transform coefficients.

[0043] FIG. 6 is a schematic diagram of an embodiment of three coding blocks containing a plurality of transform coefficients.

[0044] FIG. 7 is an embodiment of a method of context modeling implemented by a coding device.

[0045] FIG. 8 is an embodiment of a method of context modeling implemented by a coding device.

[0046] FIG. 9 is a schematic diagram of an embodiment of three coding blocks containing a plurality of transform coefficients.

[0047] FIG. 10 is a schematic diagram of an example video coding device.

[0048] FIG. 11 is a schematic diagram of an embodiment of a means for coding.

DETAILED DESCRIPTION

[0049] It should be understood at the outset that although an illustrative implementation of one or more embodiments are provided below, the disclosed systems and/or methods may be implemented using any number of techniques, whether currently known or in existence. The disclosure should in no way be limited to the illustrative implementations, drawings, and techniques illustrated below, including the exemplary designs and implementations illustrated and described herein, but may be modified within the scope of the appended claims along with their full scope of equivalents.

[0050] FIG. 1 is a block diagram illustrating an example coding system 10 that may utilize video coding techniques for applying transforms to residual blocks to generate transform coefficients. As shown in FIG. 1, the coding system 10 includes a source device 12 that provides encoded video data to be decoded at a later time by a destination device 14. In particular, the source device 12 may provide the video data to destination device 14 via a computer-readable medium 16. Source device 12 and destination device 14 may comprise any of a wide range of devices, including desktop computers, notebook (e.g., laptop) computers, tablet computers, set-top boxes, telephone handsets such as so-called “smart” phones, so-called “smart” pads, televisions, cameras, display devices, digital media players, video gaming consoles, video streaming device, or the like. In some cases, source device 12 and destination device 14 may be equipped for wireless communication.

[0051] Destination device 14 may receive the encoded video data to be decoded via computer-readable medium 16. Computer-readable medium 16 may comprise any type of medium or device capable of moving the encoded video data from source device 12 to destination device 14. In one example, computer-readable medium 16 may comprise a communication medium to enable source device 12 to transmit encoded video data directly to destination device 14 in real-time. The encoded video data may be modulated according to a communication standard, such as a wireless communication protocol, and transmitted to destination device 14. The communication medium may comprise any wireless or wired communication medium, such as a radio frequency (RF) spectrum or one or more physical transmission lines. The communication medium may form part of a packet-based network, such as a local area network, a wide-area network, or a global network such as the Internet. The communication medium may include routers, switches, base stations, or any other equipment that may be useful to facilitate communication from source device 12 to destination device 14.

[0052] In some examples, encoded data may be output from output interface 22 to a storage device. Similarly, encoded data may be accessed from the storage device by input interface. The storage device may include any of a variety of distributed or locally accessed data storage media such as a hard drive, Blu-ray discs, digital video disks (DVD)s, Compact Disc Read-Only Memories (CD-ROMs), flash memory, volatile or non-volatile memory, or any other suitable digital storage media for storing encoded video data. In a further example, the storage device may correspond to a file server or another intermediate storage device that may store the encoded video generated by source device 12. Destination device 14 may access stored video data from the storage device via streaming or download. The file server may be any type of server capable of storing encoded video data and transmitting that encoded video data to the destination device 14. Example file servers include a web server (e.g., for a website), a file transfer protocol (FTP) server, network attached storage (NAS) devices, or a local disk drive. Destination device 14 may access the encoded video data through any standard data connection, including an Internet connection. This may include a wireless channel (e.g., a Wi-Fi connection), a wired connection (e.g., digital subscriber line (DSL), cable modem, etc.), or a combination of both that is suitable for accessing encoded video data stored on a file server. The transmission of encoded video data from the storage device may be a streaming transmission, a download transmission, or a combination thereof.

[0053] The techniques of this disclosure are not necessarily limited to wireless applications or settings. The techniques may be applied to video coding in support of any of a variety of multimedia applications, such as over-the-air television broadcasts, cable television transmissions, satellite television transmissions, Internet streaming video transmissions, such as dynamic adaptive streaming over HTTP (DASH), digital video that is encoded onto a data storage medium, decoding of digital video stored on a data storage medium, or other applications. In some examples, coding system 10 may be configured to support one-way or two-way video transmission to support applications such as video streaming, video playback, video broadcasting, and/or video telephony.

[0054] In the example of FIG. 1, source device 12 includes video source 18, video encoder 20, and output interface 22. Destination device 14 includes input interface 28, video decoder 30, and display device 32. In accordance with this disclosure, video encoder 20 of the source device 12 and/or the video decoder 30 of the destination device 14 may be configured to apply the techniques for video coding. In other examples, a source device and a destination device may include other components or arrangements. For example, source device 12 may receive video data from an external video source, such as an external camera. Likewise, destination device 14 may interface with an external display device, rather than including an integrated display device.

[0055] The illustrated coding system 10 of FIG. 1 is merely one example. Techniques for video coding may be performed by any digital video encoding and/or decoding device. Although the techniques of this disclosure generally are performed by a video coding device, the techniques may also be performed by a video encoder/decoder, typically referred to as a "CODEC." Moreover, the techniques of this disclosure may also be performed by a video preprocessor. The video encoder and/or the decoder may be a graphics processing unit (GPU) or a similar device.

[0056] Source device 12 and destination device 14 are merely examples of such coding devices in which source device 12 generates coded video data for transmission to destination device 14. In some examples, source device 12 and destination device 14 may operate in a substantially symmetrical manner such that each of the source and destination devices 12, 14 includes video encoding and decoding components. Hence, coding system 10 may support one-way or two-way video transmission between video devices 12, 14, e.g., for video streaming, video playback, video broadcasting, or video telephony.

[0057] Video source 18 of source device 12 may include a video capture device, such as a video camera, a video archive containing previously captured video, and/or a video feed interface to receive video from a video content provider. As a further alternative, video source 18 may generate computer graphics-based data as the source video, or a combination of live video, archived video, and computer-generated video.

[0058] In some cases, when video source 18 is a video camera, source device 12 and destination device 14 may form so-called camera phones or video phones. As mentioned above, however, the techniques described in this disclosure may be applicable to video coding in general, and may be applied to wireless and/or wired applications. In each case, the captured, pre-captured, or computer-generated video may be encoded by video encoder 20. The encoded video information may then be output by output interface 22 onto a computer-readable medium 16.

[0059] Computer-readable medium 16 may include transient media, such as a wireless broadcast or wired network transmission, or storage media (that is, non-transitory storage media), such as a hard disk, flash drive, compact disc, digital video disc, Blu-ray disc, or other computer-readable media. In some examples, a network server (not shown) may receive encoded video data from source device 12 and provide the encoded video data to destination device 14, e.g., via network transmission. Similarly, a computing device of a medium production facility, such as a disc stamping facility, may receive encoded video data from source device 12 and produce a disc containing the encoded video data. Therefore, computer-readable medium 16 may be understood to include one or more computer-readable media of various forms, in various examples.

[0060] Input interface 28 of destination device 14 receives information from computer-readable medium 16. The information of computer-readable medium 16 may include syntax information defined by video encoder 20, which is also used by video decoder 30, that includes syntax elements that describe characteristics and/or processing of blocks and other coded units, e.g., group of pictures (GOPs). Display device 32 displays the decoded video data to a user, and may comprise any of a variety of display devices such as a cathode ray tube (CRT), a liquid crystal display (LCD), a plasma display, an organic light emitting diode (OLED) display, or another type of display device.

[0061] Video encoder 20 and video decoder 30 may operate according to a video coding standard, such as the High Efficiency Video Coding (HEVC) standard presently under development, and may conform to the HEVC Test Model (HM). Alternatively, video encoder 20 and video decoder 30 may operate according to other proprietary or industry standards, such as the International Telecommunications Union Telecommunication Standardization Sector (ITU-T) H.264 standard, alternatively referred to as Motion Picture Expert Group (MPEG)-4, Part 10, Advanced Video Coding (AVC), H.265/HEVC, or extensions of such standards. The techniques of this disclosure, however, are not limited to any particular coding standard. Other examples of video coding standards include MPEG-2 and ITU-T H.263. Although not shown in FIG. 1, in some aspects, video encoder 20 and video decoder 30 may each be integrated with an audio encoder and decoder, and may include appropriate multiplexer-demultiplexer (MUX-DEMUX) units, or other hardware and software, to handle encoding of both audio and video in a common data stream or separate data streams. If applicable, MUX-DEMUX units may conform to the ITU H.223 multiplexer protocol, or other protocols such as the user datagram protocol (UDP).

[0062] Video encoder 20 and video decoder 30 each may be implemented as any of a variety of suitable encoder circuitry, such as one or more microprocessors, digital signal processors (DSPs), application specific integrated circuits (ASICs), field programmable gate arrays (FPGAs), discrete logic, software, hardware, firmware or any combinations thereof. When the techniques are implemented partially in software, a device may store instructions for the software in a suitable, non-transitory computer-readable medium and execute the instructions in hardware using one or more processors to perform the techniques of this disclosure. Each of video encoder 20 and video decoder 30 may be included in one or more encoders or decoders, either of which may be integrated as part of a combined encoder/decoder (CODEC) in a respective device. A device including video encoder 20 and/or video decoder 30 may comprise an integrated circuit, a microprocessor, and/or a wireless communication device, such as a cellular telephone.

[0063] FIG. 2 is a block diagram illustrating an example of video encoder 20 that may implement video coding techniques. Video encoder 20 may perform intra- and inter-coding of video blocks within video slices. Intra-coding relies on spatial prediction to reduce or remove spatial redundancy in video within a given video frame or picture. Inter-coding relies on temporal prediction to reduce or remove temporal redundancy in video within adjacent frames or pictures of a video sequence. Intra-mode (I mode) may refer to any of several spatial based coding modes. Inter-modes, such as uni-directional (a.k.a., uni prediction) prediction (P mode) or bi-prediction (a.k.a., bi prediction) (B mode), may refer to any of several temporal-based coding modes.

[0064] As shown in FIG. 2, video encoder 20 receives a current video block within a video frame to be encoded. In the example of FIG. 2, video encoder 20 includes mode select unit 40, reference frame memory 64, summer 50, transform processing unit 52, quantization unit 54, and entropy coding unit 56. Mode select unit 40, in turn, includes motion compensation unit 44, motion estimation unit 42, intra-prediction (a.k.a., intra prediction) unit 46, and partition unit 48. For video block reconstruction, video encoder 20 also includes inverse quantization unit 58, inverse transform unit 60, and summer 62. A deblocking filter (not shown in FIG. 2) may also be included to filter block boundaries to remove blockiness artifacts from reconstructed video. If desired, the deblocking filter would typically filter the output of summer 62. Additional filters (in loop or post loop) may also be used in addition to the deblocking filter. Such filters are not shown for brevity, but if desired, may filter the output of summer 50 (as an in-loop filter).

[0065] During the encoding process, video encoder 20 receives a video frame or slice to be coded. The frame or slice may be divided into multiple video blocks. Motion estimation unit 42 and motion compensation unit 44 perform inter-predictive coding of the received video block relative to one or more blocks in one or more reference frames to provide temporal prediction. Intra-prediction unit 46 may alternatively perform intra-predictive coding of the received video block relative to one or more neighboring blocks in the same frame or slice as the block to be coded to provide spatial prediction. Video encoder 20 may perform multiple coding passes, e.g., to select an appropriate coding mode for each block of video data.

[0066] Moreover, partition unit 48 may partition blocks of video data into sub-blocks, based on evaluation of previous partitioning schemes in previous coding passes. For example, partition unit 48 may initially partition a frame or slice into largest coding units (LCUs), and partition each of the LCUs into sub-coding units (sub-CUs) based on rate-distortion analysis (e.g., rate-distortion optimization). Mode select unit 40 may further produce a quad-tree data structure indicative of partitioning of a LCU into sub-CUs. Leaf-node CUs of the quad-tree may include one or more prediction units (PUs) and one or more transform units (TUs).

[0067] The present disclosure uses the term "block" to refer to any of a CU, PU, or TU, in the context of HEVC, or similar data structures in the context of other standards (e.g., macroblocks and sub-blocks thereof in H.264/AVC). In HEVC, a CU includes a coding node, PUs, and TUs associated with the coding node. A size of the CU corresponds to a size of the coding node and is square or rectangular in shape. The size of the CU may range from 8×8 pixels up to the size of the treeblock with a maximum of 64×64 pixels or greater. Each CU may contain one or more PUs and one or more TUs. Syntax data associated with a CU may describe, for example, partitioning of the CU into one or more PUs. Partitioning modes may differ between whether the CU is skip or direct mode encoded, intra-prediction mode encoded, or inter-prediction (a.k.a., inter prediction) mode encoded. PUs may be partitioned to be non-square in shape. Syntax data associated with a CU may also describe, for example, partitioning of the CU into one or more TUs according to a quad-tree. A TU can be square or non-square (e.g., rectangular) in shape.

[0068] Mode select unit 40 may select one of the coding modes, intra- or inter-, e.g., based on error results, and provides the resulting intra- or inter-coded block to summer 50 to generate residual block data and to summer 62 to reconstruct the encoded block for use as a reference frame. Mode select unit 40 also provides syntax elements, such as motion vectors, intra-mode indicators, partition information, and other such syntax information, to entropy coding unit 56.

[0069] Motion estimation unit 42 and motion compensation unit 44 may be highly integrated, but are illustrated separately for conceptual purposes. Motion estimation, performed by motion estimation unit 42, is the process of generating motion vectors, which estimate motion for video blocks. A motion vector, for example, may indicate the displacement of a PU of a video block within a current video frame or picture relative to a predictive block within a reference frame (or other coded unit) relative to the current block being coded within the current frame (or other coded unit). A predictive block is a block that is found to closely match the block to be coded, in terms of pixel difference, which may be determined by sum of absolute difference (SAD), sum of square difference (SSD), or other difference metrics. In some examples, video encoder 20 may calculate values for sub-integer pixel positions of reference pictures stored in reference frame memory 64. For example, video encoder 20 may interpolate values of one-quarter pixel positions, one-eighth pixel positions, or other fractional pixel positions of the reference picture. Therefore, motion estimation unit 42 may perform a motion search relative to the full pixel positions and fractional pixel positions and output a motion vector with fractional pixel precision.

[0070] Motion estimation unit 42 calculates a motion vector for a PU of a video block in an inter-coded slice by comparing the position of the PU to the position of a predictive block of a reference picture. The reference picture may be selected from a first reference picture list (List 0) or a second reference picture list (List 1), each of which identify one or more reference pictures stored in reference frame memory 64. Motion estimation unit 42 sends the calculated motion vector to entropy encoding unit 56 and motion compensation unit 44.

[0071] Motion compensation, performed by motion compensation unit 44, may involve fetching or generating the predictive block based on the motion vector determined by motion estimation unit 42. Again, motion estimation unit 42 and motion compensation unit 44 may be functionally integrated, in some examples. Upon receiving the motion vector for the PU of the current video block, motion compensation unit 44 may locate the predictive block to which the motion vector points in one of the reference picture lists. Summer 50 forms a residual video block by subtracting pixel values of the predictive block from the pixel values of the current video block being coded, forming pixel difference values, as discussed below. In general, motion estimation unit 42 performs motion estimation relative to luma components, and motion compensation unit 44 uses motion vectors calculated based on the luma components for both chroma components and luma components. Mode select unit 40 may also generate syntax elements associated with the video blocks and the video slice for use by video decoder 30 in decoding the video blocks of the video slice.

[0072] Intra-prediction unit 46 may intra-predict a current block, as an alternative to the inter-prediction performed by motion estimation unit 42 and motion compensation unit 44, as described above. In particular, intra-prediction unit 46 may determine an intra-prediction mode to use to encode a current block. In some examples, intra-prediction unit 46 may encode a current block using various intra-prediction modes, e.g., during separate encoding passes, and intra-prediction unit 46 (or mode select unit 40, in some examples) may select an appropriate intra-prediction mode to use from the tested modes.

[0073] For example, intra-prediction unit 46 may calculate rate-distortion values using a rate-distortion analysis for the various tested intra-prediction modes, and select the intra-prediction mode having the best rate-distortion characteristics among the tested modes. Rate-distortion analysis generally determines an amount of distortion (or error) between an encoded block and an original, unencoded block that was encoded to produce the encoded block, as well as a bitrate (that is, a number of bits) used to produce the encoded block. Intra-prediction unit 46 may calculate cost values from the distortions and rates for the various encoded blocks to determine which intra-prediction mode exhibits the best rate-distortion cost value for the block.

[0074] In addition, intra-prediction unit 46 may be configured to code depth blocks of a depth map using a depth modeling mode (DMM). Mode select unit 40 may determine whether an available DMM mode produces better coding results than an intra-prediction mode and the other DMM modes, e.g., using rate-distortion optimization (RDO). Data for a texture image corresponding to a depth map may be stored in reference frame memory 64. Motion estimation unit 42 and motion compensation unit 44 may also be configured to inter-predict depth blocks of a depth map.

[0075] After selecting an intra-prediction mode for a block (e.g., a conventional intra-prediction mode or one of the DMM modes), intra-prediction unit 46 may provide information indicative of the selected intra-prediction mode for the block to entropy coding unit 56. Entropy coding unit 56 may encode the information indicating the selected intra-prediction mode. Video encoder 20 may include in the transmitted bitstream configuration data, which may include a plurality of intra-prediction mode index tables and a plurality of modified intra-prediction mode index tables (also referred to as codeword mapping tables), definitions of encoding contexts for various blocks, and indications of a most probable intra-prediction mode, an intra-prediction mode index table, and a modified intra-prediction mode index table to use for each of the contexts.

[0076] Video encoder 20 forms a residual video block by subtracting the prediction data from mode select unit 40 from the original video block being coded. Summer 50 represents the component or components that perform this subtraction operation.

[0077] Transform processing unit 52 applies a transform, such as a discrete cosine transform (DCT) or a conceptually similar transform, to the residual block, producing a video block comprising residual transform coefficient values. Transform processing unit 52 may perform other transforms which are conceptually similar to DCT. Wavelet transforms, integer transforms, sub-band transforms or other types of transforms could also be used.

[0078] Transform processing unit 52 applies the transform to the residual block, producing a block of residual transform coefficients. The transform may convert the residual information from a pixel value domain to a transform domain, such as a frequency domain. Transform processing unit 52 may send the resulting transform coefficients to quantization unit 54. Quantization unit 54 quantizes the transform coefficients to further reduce bit rate. The quantization process may reduce the bit depth associated with some or all of the coefficients. The degree of quantization may be modified by adjusting a quantization parameter. In some examples, quantization unit 54 may then perform a scan of the matrix including the quantized transform coefficients. Alternatively, entropy encoding unit 56 may perform the scan.

[0079] Following quantization, entropy coding unit 56 entropy codes the quantized transform coefficients. For example, entropy coding unit 56 may perform context adaptive variable length coding (CAVLC), context adaptive binary arithmetic coding (CABAC), syntax-based context-adaptive binary arithmetic coding (SBAC), probability interval partitioning entropy (PIPE) coding or another entropy coding technique. In the case of context-based entropy coding, context may be based on neighboring blocks. Following the entropy coding by entropy coding unit 56, the encoded bitstream may be transmitted to another device (e.g., video decoder 30) or archived for later transmission or retrieval.

[0080] Inverse quantization unit 58 and inverse transform unit 60 apply inverse quantization and inverse transformation, respectively, to reconstruct the residual block in the pixel domain, e.g., for later use as a reference block. Motion compensation unit 44 may calculate a reference block by adding the residual block to a predictive block of one of the frames of reference frame memory 64. Motion compensation unit 44 may also apply one or more interpolation filters to the reconstructed residual block to calculate sub-integer pixel values for use in motion estimation. Summer 62 adds the reconstructed residual block to the motion compensated prediction block produced by motion compensation unit 44 to produce a reconstructed video block for storage in reference frame memory 64. The reconstructed video block may be used by motion estimation unit 42 and motion compensation unit 44 as a reference block to inter-code a block in a subsequent video frame.

[0081] FIG. 3 is a block diagram illustrating an example of video decoder 30 that may implement video coding techniques. In the example of FIG. 3, video decoder 30 includes an entropy decoding unit 70, motion compensation unit 72, intra-prediction unit 74, inverse quantization unit 76, inverse transformation unit 78, reference frame memory 82, and summer 80. Video decoder 30 may, in some examples, perform a decoding pass generally reciprocal to the encoding pass described with respect to video encoder 20 (FIG. 2). Motion compensation unit 72 may generate prediction data based on motion vectors received from entropy decoding unit 70, while intra-prediction unit 74 may generate prediction data based on intra-prediction mode indicators received from entropy decoding unit 70.

[0082] During the decoding process, video decoder 30 receives an encoded video bitstream that represents video blocks of an encoded video slice and associated syntax elements from video encoder 20. Entropy decoding unit 70 of the video decoder 30 entropy decodes the bitstream to generate quantized coefficients, motion vectors or intra-prediction mode indicators, and other syntax elements. Entropy decoding unit 70 forwards the motion vectors and other syntax elements to motion compensation unit 72. Video decoder 30 may receive the syntax elements at the video slice level and/or the video block level.

[0083] When the video slice is coded as an intra-coded (I) slice, intra-prediction unit 74 may generate prediction data for a video block of the current video slice based on a signaled intra-prediction mode and data from previously decoded blocks of the current frame or picture. When the video frame is coded as an inter-coded (e.g., B, P, or GPB) slice, motion compensation unit 72 produces predictive blocks for a video block of the current video slice based on the motion vectors and other syntax elements received from entropy decoding unit 70. The predictive blocks may be produced from one of the reference pictures within one of the reference picture lists. Video decoder 30 may construct the reference frame lists, List 0 and List 1, using default construction techniques based on reference pictures stored in reference frame memory 82.

[0084] Motion compensation unit 72 determines prediction information for a video block of the current video slice by parsing the motion vectors and other syntax elements, and uses the prediction information to produce the predictive blocks for the current video block being decoded. For example, motion compensation unit 72 uses some of the received syntax elements to determine a prediction mode (e.g., intra- or inter-prediction) used to code the video blocks of the video slice, an inter-prediction slice type (e.g., B slice, P slice, or GPB slice), construction information for one or more of the reference picture lists for the slice, motion vectors for each inter-encoded video block of the slice, inter-prediction status for each inter-coded video block of the slice, and other information to decode the video blocks in the current video slice.

[0085] Motion compensation unit 72 may also perform interpolation based on interpolation filters. Motion compensation unit 72 may use interpolation filters as used by video encoder 20 during encoding of the video blocks to calculate interpolated values for sub-integer pixels of reference blocks. In this case, motion compensation unit 72 may determine the interpolation filters used by video encoder 20 from the received syntax elements and use the interpolation filters to produce predictive blocks.

[0086] Data for a texture image corresponding to a depth map may be stored in reference frame memory 82. Motion compensation unit 72 may also be configured to inter-predict depth blocks of a depth map.

[0087] Thus, when performing video coding, an encoder partitions images into image blocks. The image blocks are then compressed into prediction blocks by either intra-prediction or inter-prediction. Prediction matches a current image block with a reference sample and then encodes the relationship instead of the pixels in the image block. When the image block does not exactly match the reference sample, the differences are maintained as a residual block (also known simply as residual). The residual block can be further compressed by applying transforms to convert the residual block into a group of transform coefficients. A decoder can recover the original image or an approximation of the original image by applying the encoding process in reverse. The following disclosure relates to mechanisms corresponding to applying transforms to residual blocks to generate transform coefficients. Such disclosure may be applied at the encoder and/or at the decoder, depending on the example.

[0088] In HEVC, the transform coefficients of a coding block are coded using non-overlapped coefficient groups (CGs). Each CG contains the transform coefficients of a 4×4 block of a coding block. The CGs inside a coding block, and the transform coefficients within a CG, are coded according to pre-defined scan orders. The coding of transform coefficient levels of a CG with at least one non-zero transform coefficient may be separated into multiple scan passes. In a first pass, a first bin (denoted by bin0, also referred to as the *significant_coeff_flag*, which indicates to a coding device that the magnitude of the transform coefficient is larger than zero) is coded. Next, two scan passes are applied for context coding the second/third bins (denoted by bin1 and bin2, respectively, also referred as the *coeff_abs_greater1_flag* and the *coeff_abs_greater2_flag*). Finally, two more scan passes for coding the sign information and the remaining values (also referred as *coeff_abs_level_remaining*) of coefficient levels are invoked, if desired. Only bins in the first three scan passes are coded in a regular mode, and those bins are termed regular bins in the following description.

[0089] For the development of video compression with capability beyond High Efficiency Video Coding (HEVC), the Joint Video Exploration Team (JVET) developed the Joint Exploration Test Model 7 (JEM7) as documented in J. Chen, E. Alshina, G. J. Sullivan, J.-R. Ohm, J. Boyce (JEM editors) "Algorithm description of Joint Exploration Test Model 7", JVET-G1001, Torino, Jul. 2017. JVET implemented the JEM7 in the JEM 7.0 software package.

[0090] JEM7 changed the context modeling for the arithmetic coding of transform coefficients from that used in HEVC. In particular, the context modeling for regular bins is changed. A context index is, for example, a listing of the probability models used in CABAC. When coding bin *i* in the *i*-th scan pass (*i* being 0, 1, 2), the context index is dependent on the values of the *i*-th bins of previously coded coefficients in the neighborhood covered by a local template. More specifically, the context index is determined based on the sum of the *i*-th bins of neighboring coefficients.

[0091] FIG. 4 illustrates a coding block 400 containing a plurality of transform coefficients 402. The transform coefficients 402 are, for example, values obtained when a transform (e.g., a discrete cosine transform (DST) or other conceptually similar transform) is applied to a residual block. Each of the transform coefficients 402 may be identified or referred to by the location of the transform coefficient 402 within the coding block 400. To identify the location of the transform coefficients 402, the coding block 400 has been labeled with a vertical axis 404 and a horizontal axis 406. The vertical axis 404 and the horizontal axis 406 each include numbering that begins at the top, left corner of the coding block 400 as oriented in FIG. 4. Because the coding block 400 depicted in FIG. 4 is an 8x8 block, the vertical axis 404 and the horizontal axis 406 each include numbering that increases from 0 to 7. In practical applications, the coding block 400 may have other dimensions (e.g., 16x16, 32x32, etc.). In such cases, the vertical axis 404 and the horizontal axis 406 are numbered accordingly.

[0092] The transform coefficient 402 located at the top, left of the coding block 400 in FIG. 4 is referred to as an origin transform coefficient. For the purpose of identification, the origin transform coefficient is labeled $C_{0,0}$. The origin transform coefficient may also be referred to as the direct current (DC) component. The transform coefficient 402 located in the opposite corner at the bottom, right of the coding block 400 is labeled $C_{7,7}$. Using the vertical axis 404 and the horizontal axis 406, the other transform coefficients 402 in the coding block 400 can be similarly identified.

[0093] Still referring to FIG. 4, the coding block 400 contains a local template 408. The local template 408 contains a current transform coefficient, which is labeled X, and up to five spatial neighboring transform coefficients 402, which are labeled X_0, X_1, X_2, X_3, and X_4. While the local template 408 in FIG. 4 contains five of the spatial neighboring transform coefficients 402, any number of spatial neighboring transform coefficients may be used in practical applications.

[0094] To capture the characteristics of transform coefficients 402 at different frequencies, the coding block 400 may be split into up to three regions 410, 412, 414 (also identified by shading in the figure). In an embodiment, the splitting methodology is fixed regardless of the size of the coding block 400. For example, when coding bin0 of luma transform coefficients 402 as depicted in FIG. 4, the coding block 400 is split into three regions 410, 412, and 414. Luma and chroma transform components are treated in a similar way, but with separate sets of context models (a.k.a., context index groups). Moreover, the context model selection for bin0 (e.g., the significant flags) of the luma component is further dependent on transform size.

[0095] The boundary between the regions 410, 412, 414 is depicted by diagonal lines, d_0 and d_1. The location of the diagonal lines d_0, d_1 relative to the coding block 400 is based on the formula $x+y=d_0$ and $x+y=d_1$, where x and y represent the coordinates of the transform coefficients 402. As shown in FIG. 4, the diagonal line d_0 passes through the transform coefficients 402 located at positions 2,0, 1,1, and 0,2. Likewise, the diagonal line d_1 passes through the transform coefficients 402 located at positions 5,0, 4,1, 3,2, 2,3, 1,4, and 0,5. The diagonal lines d_0, d_1 may pass through other transform coefficients 402 in practical applications.

[0096] Each of the regions 410, 412, and 414 has a corresponding context index group (e.g., group of available contexts). For example, the region 410 corresponds to context index group g_2 containing context indexes 0 to 5 (e.g., $g_2 = \{0, 1, 2, 3, 4, 5\}$). The region 412 corresponds to context index group g_1 containing context indexes 6 to 11 (e.g., $g_1 = \{6, 7, 8, 9, 10, 11\}$) and the region 414 corresponds to context index group g_0 containing context indexes 12 to 17 (e.g., $g_0 = \{12, 13, 14, 15, 16, 17\}$). The context index groups g_2, g_1, and g_0

may have any number of context indexes in practical applications. In some circumstances, the context indexes are for context-adaptive binary arithmetic coding (CABAC).

[0097] FIG. 5 illustrates three coding blocks 500. The coding block 500 labeled GT0 corresponds to the *significant_coeff_flag*, the coding block 500 labeled GT1 corresponds to the *coeff_abs_greater1_flag*, and the coding block 500 labeled GT2 and corresponds to the *coeff_abs_greater2_flag*. Each of these flags provides information to the coding device, as described above. For the purpose of identification, the coding blocks 500 may be individually referred to as coding block 500 (GT0), coding block 500 (GT1), and coding block 500 (GT2).

[0098] The coding blocks 500 in FIG. 5 are similar to the coding block 400 of FIG. 4. In that regard, the coding blocks 500 in FIG. 5 include a plurality of transform coefficients 502. Moreover, the transform coefficients 502 (e.g., luminance transform coefficients) in the coding blocks 500 have been divided into three regions 510, 512, and 514 (also identified by shading in the figure). For ease of illustration, diagonal lines similar to those used in FIG. 4 that depict a boundary between the regions have been intentionally omitted in FIG. 5. However, the boundaries are still indicated using the various coordinate designations for the transform coefficients 502. For example, one boundary exists in the coding block 500 along the diagonal line created by transform coefficients 502 at $C_{0,2}$, $C_{1,1}$, and $C_{2,0}$ and another boundary exists along the diagonal line created by transform coefficients 502 at $C_{0,5}$, $C_{1,4}$, $C_{2,3}$, $C_{3,2}$, $C_{4,1}$, and $C_{5,0}$.

[0099] Notably, the regions 510, 512, and 514 in the coding block 500 (GT0), the coding block 500 (GT1), and the coding block 500 (GT2) are not the same size. For example, region 510 in coding block 500 (GT0) includes three transform coefficients 502 and region 510 in coding block 500 (GT1) and coding block 500 (GT2) includes six transform coefficients 502.

[00100] As shown in FIG. 5, the coding blocks 500 have been divided into transform sub-blocks 520, 522, 524, and 526. The transform sub-blocks 520, 522, 524, and 526 may also be referred to herein as sub-blocks. When the coding block 500 is, for example, an 8x8 block, each of the transform sub-blocks 520, 522, 524, 526 comprises a 4x4 sub-block. The transform sub-blocks 520, 522, 524, 526 may have different sizes in practical applications.

[00101] JEM inherited the concept of a coefficient group from HEVC. Therefore, the transform coefficients 502 in the coding blocks 500 of FIG. 5 are grouped into a first coefficient group 530, a second coefficient group 532, a third coefficient group 534, and a fourth coefficient group 536. Each of the first, second, third, and fourth coefficient groups 530, 532, 534, 536 corresponds to one of the transform sub-blocks 520, 522, 524, 526. For instance, the first coefficient group 530 corresponds to the transform sub-block 520, the second coefficient group 532 corresponds to the transform sub-block 524, the third coefficient group 534 corresponds to the transform sub-block 522, and the fourth coefficient group 536 corresponds to the transform sub-block 526. For a diagonal scan of the transform coefficients 502, each coefficient group 530, 532, 534, 536 comprises a 4x4 block.

[00102] For a transform coefficient $\mathrm{C}_{\mathrm{x,y}}$ located at the coordinate (x, y) in a $\mathrm{M} \times \mathrm{N}$ transform unit with $0 \leq \mathrm{x} \leq (\mathrm{M}-1)$ and $0 \leq \mathrm{y} \leq (\mathrm{N}-1)$, the three regions 510, 512, 514 are separated by two diagonal lines $\mathrm{x}+\mathrm{y}=\mathrm{d}_0$ and $\mathrm{x}+\mathrm{y}=\mathrm{d}_1$ where $(\mathrm{d}_0, \mathrm{d}_1)=(2,5)$, and the context group is assigned as in the following pseudo Python code:

```
For each coefficient group with at least one non-zero coefficient:
    For each coefficient $C_{x,y}$ in the coefficient group:
        If $(x+y) < d_0$ the context group $g_0$ is assigned to $C_{x,y}$
        Else if $(x+y) < d_1$ the context group $g_1$ is assigned to $C_{x,y}$
        Else the context group $g_2$ is assigned to $C_{x,y}$
```

[00103] In order to determine the context index groups (a.k.a., the context model indexes) for the coding block 500 (GT0), the coding block 500 (GT1), and the coding block 500 (GT2), the software from the JEM 7.0 software package repeatedly tests where a transform coefficient is located within the coding block.

[00104] Unfortunately, in JEM, regions are permitted to cross over into different coefficient groups for transform coefficients with diagonal scanning. For example, as shown in FIG. 5, region 512 extends into the first coefficient group 530, the second coefficient group 532, and the third coefficient group 534 and intersects with the first transform sub-block 520, the second transform sub-block 522, and the third transform sub-block 524. Further, each region is associated with a different context index group. Because of this, the software from the JEM 7.0 software package has to repeatedly test where each transform coefficient 502 is located within the coding block 500 (GT0), the coding block 500 (GT1), and the coding block 500 (GT2) to determine the context index group (a.k.a., the context model index). This process is made more difficult due to the fact that the regions 510, 512, and 514 within the coding block 500 (GT0), the coding block 500 (GT1), and the coding block 500 (GT2) do not necessarily have the same size and shape.

[00105] Disclosed herein are coding techniques and/or methods that limit the number/amount of position-based context index switching (e.g., between contest index groups) while maintaining or improving coding efficiency. As will be more fully explained below, the present disclosure restricts or limits context switching to the coefficient group that intersects with the sub-block that contains an origin transform coefficient. Because there is no position-based context index switching in other coefficient groups, computational complexity and power consumption of a coding device are reduced. This improves the overall performance of the coding device relative to other coding devices without the benefit of the coding techniques and/or methods disclosed herein.

[00106] FIG. 6 illustrates three coding blocks 600. The coding block 600 labeled GT0 corresponds to the *significant_coeff_flag*, the coding block 600 labeled GT1 corresponds to the *coeff_abs_greater1_flag*, and the coding block 600 labeled GT2 corresponds to the *coeff_abs_greater2_flag*. For the purpose of identification, the coding blocks 600 may be individually referred to as coding block 600 (GT0), coding block 600 (GT1), and coding block 600 (GT2).

[00107] The coding blocks 600 in FIG. 6 are similar to the coding blocks 500 of FIG. 5. In that regard, the coding blocks 600 in FIG. 6 include a plurality of transform coefficients 602. Moreover, the transform coefficients 602 (e.g., luminance transform coefficients) in the coding blocks 600 have been divided into three regions 610, 612, and 614 (also identified by shading in the figure). For ease of illustration, diagonal lines similar to those used in FIG. 4 that depict a boundary between the regions have been intentionally omitted in FIG. 6. However, the boundaries are still indicated using the various coordinate designations for the transform coefficients 602. For example, one boundary exists in the coding block 600 along the diagonal line created by transform coefficients 602 at $C_{0,2}$, $C_{1,1}$, and $C_{2,0}$ and another boundary exists along the diagonal line created by transform coefficients 602 at $C_{0,4}$, $C_{1,3}$, $C_{2,2}$, $C_{3,1}$, and $C_{4,0}$.

[00108] Notably, the regions 610, 612, and 614 in the coding block 600 (GT0), the coding block 600 (GT1), and the coding block 600 (GT2) are the same size and shape. For example, region 610 in coding block 600 (GT0) includes three transform coefficients 602 and region 610 in coding block 600 (GT1) and coding block 600 (GT2) also includes three transform coefficients 602. Therefore, the present disclosure supports more reuse between the modules for coding block 600 (GT0), coding block 600 (GT1), and coding block 600 (GT2) in hardware and/or software implementation relative to the current methodology of JEM. Despite the regions 610, 612, and 614 having the same size and shape, in an embodiment context index groups for similar regions in different coding blocks 600 may contain a different number of context indexes. For example, the region 610 in coding block 600 (GT0) corresponds to context index group g_2 containing context indexes 0 to 5 (e.g., $g_2 = \{0, 1, 2, 3, 4, 5\}$) while the region 610 in coding block 600 (GT1) corresponds to context index group g_2 containing context indexes 0 to 4 (e.g., $g_2 = \{0, 1, 2, 3, 4\}$).

[00109] As shown in FIG. 6, the coding blocks 600 have been divided into transform sub-blocks 620, 622, 624, and 626. The transform sub-blocks 620, 622, 624, and 626 may also be referred to herein as sub-blocks. When the coding block 600 is, for example, an 8x8 block, each of the transform sub-blocks 620, 622, 624, 626 comprises a 4x4 sub-block. The transform sub-blocks 620, 622, 624, 626 may have different sizes in practical applications.

[00110] In contrast to the coding block 500 of FIG. 5 where, for example, region 512 is permitted to cross into several coefficient groups, namely first coefficient group 530, second coefficient group 532, and third coefficient group 534, the regions 610 and 612 in FIG. 6 are restricted to the first coefficient group 630 containing the origin coefficient 602 (labeled $C_{0,0}$) for each of coding block 600 (GT0), coding block 600 (GT1), and coding block 600 (GT2). Because the regions 610 and 612 are only found within the first coefficient group 630, positional context index switching (e.g., making a choice between different context index groups g_2, g_1, and g_0) is limited to the first coefficient group 630. That is, in each scan pass of the first, second, third, and fourth coefficient groups 630-636, the position context index switching is limited to the coefficient group (e.g., the first coefficient group 630) that intersects with the sub-block (e.g., transform sub-block 620) containing the origin transform coefficient. In addition, positional context index switching is no longer needed for the second coefficient group 632, the third coefficient group 634, and the fourth coefficient group 636 because those coefficient groups have only one context index group to choose from (e.g., g_0). That is, for any transform coefficient 602 located in the second coefficient group 632, the third coefficient group 634, and the fourth coefficient group 636, the software from the JEM 7.0 software package (as modified by the teachings of the present disclosure) would no longer have to repeatedly test where each transform coefficient 602 is located within the coding block 600 (GT0), the coding block 600 (GT1), and the coding block 600 (GT2) to determine the context index group (a.k.a., the context model index). Consequently, the computation and power consumption needed for positional context switching is removed from all but one coefficient group (e.g., the first coefficient group 630) in each diagonal scanning pass of the coding block 600.

[00111] FIG. 7 is an embodiment of a method 700 of context modeling implemented by a coding device. In an embodiment, the method 700 is performed by an encoder or a decoder (e.g., video encoder 20 or video decoder 30). The method 700 may be performed when encoding or decoding transform coefficients (e.g., transform coefficients 602). The method 700 may be performed to reduce the computation and power consumption needed for positional context switching relative to the current software and/or hardware implementations provided by JEM.

[00112] In block 702, a context index group (e.g., either g_0, g_1, or g_2) is selected from a plurality of context index groups (e.g., g_0, g_1, and g_2) based on a location of a transform coefficient (e.g., transform coefficient 602) when a coefficient group (e.g., first coefficient group 630) contains an origin transform coefficient (e.g., the transform coefficient 602 labeled $C_{0,0}$). The selection may be performed by, for example, a processor of the coding device. In an embodiment, the selection may be performed using firmware, hardware, software, or some combination thereof.

[00113] In block 704, the single available context index group (e.g., g2), which is independent of the location of the transform coefficient (e.g., transform coefficient 602), is used when the coefficient group (e.g., second coefficient group 632, third coefficient group 634, and fourth coefficient group 636) does not contain the origin transform coefficient (e.g., the transform coefficient 602 labeled $C_{0,0}$). The single available context index group may be used by, for example, a processor of the coding device. In an embodiment, the selection or use of the context index group may be performed using firmware, hardware, software, or some combination thereof.

[00114] It should be noted that, blocks 702 and 704 do not limit the process order, actually, for each transform coefficient, either block 702 or block 704 will be involved in encoding or decoding. That is, blocks 702 and 704 are two options in encoding or decoding.

[00115] Once the context index group selection is made, the coding process may continue as described above in FIGS. 1-3 until an encoded bitstream is generated and transmitted toward a decoder, or an image is generated and displayed for a user on a user device. For example, a context index can be selected from the context group that was selected or from the context group being used. Thereafter, the context index that was selected can be used for coding (e.g., to generate the encoded bitstream transmitted toward the decoder, or to generate the image displayed for the user on the user device).

[00116] FIG. 8 is an embodiment of a method 800 of context modeling implemented by a coding device. In an embodiment, the method 800 is performed by an encoder or a decoder (e.g., video encoder 20 or video decoder 30). The method 800 may be performed when encoding or decoding transform coefficients (e.g., transform coefficients 602). The method

800 may be performed to reduce the computation and power consumption needed for positional context switching relative to the current software and/or hardware implementations provided by JEM.

[00117] In block 802, a context index group (e.g., either g_0, g_1, or g_2) is selected from a plurality of context index groups (e.g., g_0, g_1, and g_2) based on a location of a transform coefficient (e.g., transform coefficient 602) when a coefficient group (e.g., first coefficient group 630) intersects with a transform sub-block (e.g., sub-block 620) containing an origin transform coefficient (e.g., the transform coefficient 602 labeled $C_{0,0}$). The selection may be performed by, for example, a processor of the coding device.

[00118] In block 804, the single available context index group (e.g., g2), which is independent of the location of the transform coefficient (e.g., transform coefficient 602), is used when the coefficient group (e.g., second coefficient group 632, third coefficient group 634, and fourth coefficient group 636) does not intersect with a transform sub-block (e.g., sub-block 622, 624, or 626) containing the origin transform coefficient (e.g., the transform coefficient 602 labeled $C_{0,0}$). The single available context index group may be used by, for example, a processor of the coding device. In an embodiment, the selection or use of the context index group may be performed using firmware, hardware, software, or some combination thereof.

[00119] It should be noted that, blocks 802 and 804 do not limit the process order, actually, for each transform coefficient, either block 802 or block 804 will be involved in encoding or decoding. That is, blocks 802 and 804 are two options in encoding or decoding.

[00120] Once the context index group selection is made, the coding process may continue as described above in FIGS. 1-3 until an encoded bitstream is generated and transmitted toward a decoder, or an image is generated and displayed for a user on a user device. For example, a context index can be selected from the context group that was selected or from the context group being used. Thereafter, the context index that was selected can be used for coding (e.g., to generate the encoded bitstream transmitted toward the decoder, or to generate the image displayed for the user on the user device).

[00121] Keeping the above in mind, the coding of transform coefficient levels of a coefficient group (CG) with at least one non-zero transform coefficient in JEM 7.0 is separated into multiple scan passes for encoding *significant_coeff_flag (GT0)*, *coeff_abs_greater1_flag (GT1)*, *coeff_abs_greater2_flag (GT2),* and *coeff_abs_level_remaining*. In an embodiment, the maximum transform coefficient level is 2 before the *coeff_abs_level_remaining* pass for encoding *coeff_abs_greater2_flag (GT2)*.

[00122] The present disclosure is also applicable to the cases that the transform coefficient levels of a CG are encoded into multiple scan passes for encoding *coeff_abs_greaterX_flag (GTX)* for *X* equal to 0, 1, 2, ..., *K* for some predetermined integer *K*. Based on the foregoing, the positional context index switching is limited to the CG intersected with the predetermined $\mathrm{m} \times \mathrm{n}$ blocks (e.g., transform sub-block 620) containing the $\mathrm{C}_{0,0}$ component for encoding *coeff_abs_greaterX_flag (GTX)* for *X* equals to 0, 1, 2, ..., *K*. When the CGs are 4×4 blocks and $(\mathrm{m}, \mathrm{n}) = (4,4)$, the positional context index switching is limited to the CG containing the $\mathrm{C}_{0,0}$ component. In this case, only one CG has positional context index switching. All other CGs do not have positional context switching. Therefore, the computation and power consumption needed for positional context switching is removed from all but one CG.

[00123] The present disclosure is also applicable to various methods on how the transform coefficient levels of a CG are separated into scan passes for encoding transform coefficient levels of a CG. In one embodiment, *the significant_coeff_flag (GT0)* is encoded in one pass and the *coeff_abs_greater1_flag (GT1)*, *coeff_abs_greater2_flag (GT2)*, ..., *coeff_abs_greaterN_flag (GTN)*, and the *coeff_abs_level_remaining* are encoded in a second pass. In each scan pass of a CG, the positional context index switching is limited to the CG intersected with the $\mathrm{m} \times \mathrm{n}$ blocks containing the $\mathrm{C}_{0,0}$ component. When the CGs are 4×4 blocks and $(\mathrm{m}, \mathrm{n}) = (4,4)$, the positional context index switching is limited to the CG containing the $\mathrm{C}_{0,0}$ component. In this case, for each pass, only one CG has positional context index switching, and all other CGs in the same pass do not have positional context switching. Consequently, the computation and power consumption needed for positional context switching is removed from all but one CG in each pass of the CG.

[00124] An example of coding that may be utilized to implement the techniques of the present disclosure is provided. In general, let the *coeff_flag (GTX), X=0,..., K* have R regions and R groups of context index $[\mathrm{g}_0, \mathrm{g}_1, \ldots, \mathrm{g}_{\mathrm{R}-1}]$ for some predetermined integer K. For a transform coefficient $\mathrm{C}_{\mathrm{x,y}}$ located at the coordinate (x, y) in a $\mathrm{M} \times \mathrm{N}$ transform unit with $0 \leq \mathrm{x} \leq (\mathrm{M}-1)$ and $0 \leq \mathrm{y} \leq (\mathrm{N}-1)$, the R regions are separated by $\mathrm{R}-1$ distinct diagonal lines $\mathrm{x}+\mathrm{y} = \mathrm{d}_{\mathrm{i}}$, $\mathrm{i} = 0, \ldots, \mathrm{R}-2$, and d_{i} is monotonic increasing with respect to i. For predetermined integer tuple (m, n), the *coeff_flag (GTX)* of the transform coefficients is assigned as in the following pseudo Python code:

```
For each coefficient group with at least one non-zero coefficient:
    If the coefficient group intersects with $\{\mathrm{C}_{\mathrm{x,y}} \mid 0 \leq \mathrm{x} < \mathrm{m}, 0 \leq \mathrm{y} < \mathrm{n}\}$:
        For each coefficient $C_{x,y}$ in the coefficient group:
        Assign the context group $g_{R-1}$ to $C_{x,y}$
            For i in $[0, \ldots, R-1]$:
                If $(x + y) < d_i$:
                    Assign the context group $g_i$ to $C_{x,y}$
                    Break
    Else:
        For each coefficient $C_{x,y}$ in the coefficient group:
            Assign the context group $g_{R-1}$ to $C_{x,y}$
```

[00125] Another example of coding that may be utilized to implement the techniques of the present disclosure is provided. When there are $R = 3$ regions for encoding the *coeff_flag (GTX)* with $(m, n) = (4,4)$ and the size of the coefficient group is also 4×4, the pseudo Python code for context index group assignment is as follows:

```
For each coefficient group with at least one non-zero coefficient:
    If the coefficient group contains $C_{0,0}$:
        For each coefficient $C_{x,y}$ in the coefficient group:
            If $(x + y) < d_0$ assign the context group $g_0$ to $C_{x,y}$
            Else if $(x + y) < d_1$ assign the context group $g_1$ to $C_{x,y}$
            Else assign the context group $g_2$ to $C_{x,y}$
        Else:
            For each coefficient $C_{x,y}$ in the coefficient group:
                Assign the context group $g_2$ to $C_{x,y}$
```

[00126] When there are $R = 2$ regions for encoding the *coeff_flag (GTX)* with $(m, n) = (4,4)$ and the size of the coefficient group is also 4×4, the pseudo Python code for context index group assignment is as follows:

```
For each coefficient group with at least one non-zero coefficient:
    If the coefficient group contains $C_{0,0}$:
        For each coefficient $C_{x,y}$ in the coefficient group:
            If $(x + y) < d_0$ assign the context group $g_0$ to $C_{x,y}$
            Else assign the context group $g_1$ to $C_{x,y}$
    Else:
        For each coefficient $C_{x,y}$ in the coefficient group:
```

Assign the context group g_1 to $C_{x,y}$

[00127] For the encoding of the *coeff_flag (GTX)* with $(m, n) = (4,4)$ and the size of the coefficient group is also 4×4, the decision for checking whether the coefficient group contains $C_{0,0}$ can be implemented as checking if the coefficient group is the first coefficient group according to the forward scan order starting from the coefficient $C_{0,0}$.

[00128] Examples of positional based index group assignment in accord with this disclosure are depicted in FIG. 9 when the size of the group is 4x4 and $(m, n) = (4,4)$.

[00129] FIG. 9 illustrates three coding blocks 900. The coding blocks 900 are similar to the coding blocks 600 of FIG. 6. However, in the embodiment of FIG. 9, each of the coding blocks 900 is labeled GTX and the organization of regions 910-914 is somewhat different. For example, the coding block 900 in the middle of the figure contains only two regions, namely region 910 and 914. In practical applications, the regions could also be regions 910 and 912 or something similar. Moreover, any of the coding blocks may contain only two regions. In addition, sub-block 920 could contain any number of regions.

[00130] As before, the coding blocks 900 include a plurality of transform coefficients 902. The transform coefficients 902 (e.g., luminance transform coefficients) in the coding blocks 900 have been divided into two 910, 914 or three regions 910-914 (also identified by shading in the figure). As shown in FIG. 9, the coding blocks 900 have been divided into transform sub-blocks 920, 922, 924, and 926. The regions 910 and 912 in FIG. 9 are restricted to the first coefficient group 930 containing the origin coefficient 902 (labeled $C_{0,0}$) for each coding block 900 (GTX).

[00131] Because the regions 910 and 912 are only found within the first coefficient group 930, positional context index switching (e.g., making a choice between different context index groups g_2, g_1, and g_0) is limited to the first coefficient group 930. That is, in each scan pass of the first, second, third, and fourth coefficient groups 930-936, the position context index switching is limited to the coefficient group (e.g., the first coefficient group 930) that intersects with the sub-block (e.g., transform sub-block 920) containing the origin transform coefficient. In addition, positional context index switching is no longer needed for the second coefficient group 932, the third coefficient group 934, and the fourth coefficient group 936 because those coefficient groups have only one context index group to choose from (e.g., g_0). That is, for any transform coefficient 902 located in the second coefficient group 932, the third coefficient group 934, and the fourth coefficient group 936, the software from the JEM 7.0 software package (as modified by the teachings of the present disclosure) would no longer have to repeatedly test where each transform coefficient 902 is located within the coding block 900 (GTX) to determine the context index group (a.k.a., the context model index). Consequently, the computation and power consumption needed for positional context switching is removed from all but one coefficient group (e.g., the first coefficient group 930) in each diagonal scanning pass of the coding block 900.

[00132] FIG. 10 is a schematic diagram of a video coding device 1000 (e.g., a client device, a content server, etc.) according to an embodiment of the disclosure. The video coding device 1000 is suitable for implementing the methods and processes disclosed herein. The video coding device 1000 comprises ingress ports 1010 and receiver units (Rx) 1020 for receiving data; a processor, logic unit, or central processing unit (CPU) 1030 to process the data; transmitter units (Tx) 1040 and egress ports 1050 for transmitting the data; and a memory 1060 for storing the data. The video coding device 1000 may also comprise optical-to-electrical (OE) components and electrical-to-optical (EO) components coupled to the ingress ports 1010, the receiver units 1020, the transmitter units 1040, and the egress ports 1050 for egress or ingress of optical or electrical signals.

[00133] The processor 1030 is implemented by hardware and software. The processor 1030 may be implemented as one or more CPU chips, cores (e.g., as a multi-core processor), field-programmable gate arrays (FPGAs), application specific integrated circuits (ASICs), and digital signal processors (DSPs). The processor 1030 is in communication with the ingress ports 1010, receiver units 1020, transmitter units 1040, egress ports 1050, and memory 1060. The processor 1030 comprises a coding module 1070. The coding module 1070 implements the disclosed embodiments described above. The inclusion of the coding module 1070 therefore provides a substantial improvement to the functionality of the coding device 1000 and effects a transformation of the video coding device 1000 to a different state. Alternatively, the coding module 1070 is implemented as instructions stored in the memory 1060 and executed by the processor 1030.

[00134] The video coding device 1000 may also include input and/or output (I/O) devices 1080 for communicating data to and from a user. The I/O devices 1080 may include output devices such as a display for displaying video data, speakers for outputting audio data, etc. The I/O devices 1080 may also include input devices, such as a keyboard, mouse, trackball, etc., and/or corresponding interfaces for interacting with such output devices.

[00135] The memory 1060 comprises one or more disks, tape drives, and solid-state drives and may be used as an over-flow data storage device, to store programs when such programs are selected for execution, and to store instructions and data that are read during program execution. The memory 1060 may be volatile and non-volatile and may be read-only memory (ROM), random-access memory (RAM), ternary content-addressable memory (TCAM), and static random-access memory (SRAM).

[00136] FIG. 11 is a schematic diagram of an embodiment of a means for coding 1100. In embodiment, the means for coding 1100 is implemented in a video coding device 1102 (e.g., a video encoder 20 or a video decoder 30). The video coding device 1102 includes receiving means 1101. The receiving means 1101 is configured to receive a picture to encode or to receive a bitstream to decode. The video coding device 1102 includes transmission means 1107 coupled to the receiving means 1101. The transmission means 1107 is configured to transmit the bitstream to a decoder or to transmit a decoded image to a display means (e.g., one of the I/O devices 1080).

[00137] The video coding device 1102 includes a storage means 1103. The storage means 1103 is coupled to at least one of the receiving means 1101 or the transmission means 1107. The storage means 1103 is configured to store instructions. The video coding device 1102 also includes processing means 1105. The processing means 1105 is coupled to the storage means 1103. The processing means 1105 is configured to execute the instructions stored in the storage means 1103 to perform the methods disclosed herein.

[00138] From the foregoing, the present disclosure limits the context switching to only in the coefficient group that intersects with the $m \times n$ block containing the $C_{0,0}$ component. In addition, the present disclosure supports the same positional context index group assignment for encoding GTX for all X.

[00139] While several embodiments have been provided in the present disclosure, it may be understood that the disclosed systems and methods might be embodied in many other specific forms without departing from the spirit or scope of the present disclosure. The present examples are to be considered as illustrative and not restrictive, and the intention is not to be limited to the details given herein. For example, the various elements or components may be combined or integrated in another system or certain features may be omitted, or not implemented.

[00140] In addition, techniques, systems, subsystems, and methods described and illustrated in the various embodiments as discrete or separate may be combined or integrated with other systems, components, techniques, or methods without departing from the scope of the present disclosure. Other examples of changes, substitutions, and alterations are ascertainable by one skilled in the art and may be made without departing from the spirit and scope disclosed herein.

CLAIMS

What is claimed is:

1. A method of context modeling implemented by a coding device, the method comprising:

selecting, by the coding device, a context index group from a plurality of context index groups based on a location of a transform coefficient when a coefficient group contains the transform coefficient contains an origin transform coefficient;

using, by the coding device, a single available context index group when the coefficient group contains the transform coefficient does not contain the origin transform coefficient;

selecting, by the coding device, a context index from the context group that was selected or from the single available context group being used; and

coding, by the coding device, using the context index that was selected.

2. A method of context modeling implemented by a coding device, the method comprising:

selecting, by the coding device, a context index group from a plurality of context index groups based on a location of a transform coefficient when a coefficient group containing the plurality of the transform coefficients intersects with a transform sub-block containing an origin transform coefficient;

using, by the coding device, a single available context index group when the coefficient group contains the transform coefficient does not contain the origin transform coefficient;

selecting, by the coding device, a context index from the context group that was selected or from the single available context group being used; and

coding, by the coding device, using the context index that was selected.

3. The method of claim 1 or 2, wherein the origin transform coefficient is disposed at position $C_{0,0}$ in a transform block.

4. The method of any of claims 1 to 3, wherein the coefficient group contains transform coefficients of a sub-block of a transform block.

5. The method of any of claims 1 to 4, wherein the transform coefficients in the sub-block of the transform block that contains the origin transform coefficient have been divided into regions.

6. The method of claim 5, wherein the regions for a *significant_coeff_flag (GT0)*, a *coeff_abs_greater1 (GT1)*, and a *coeff_abs_greater2 (GT2)* are the same.

7. The method of claim 6, wherein the *significant_coeff_flag (GT0)* is encoded in a first pass, and the *coeff_abs_greater1 (GT1)*, the *coeff_abs_greater2 (GT2)*, and a *coeff_abs_level_remaining* are coded in a second pass.

8. The method of claim 6 to 7, wherein the *significant_coeff_flag (GT0)*, the *coeff_abs_greater1 (GT1)*, and the *coeff_abs_greater2 (GT2)* are encoded in different passes.

9. The method of any of claims 1 to 8, wherein the transform coefficients for the coefficient group are separated into multiple scan passes for encoding the *significant_coeff_flag (GT0)*, the *coeff_abs_greater1 (GT1)*, the *coeff_abs_greater2 (GT2)*, ..., and a *coeff_abs_greaterN (GTN)* for a pre-determined integer *N*.

10. The method of any of claims 5 to 9, wherein each of the regions corresponds to one of the plurality of context index groups.

11. The method of any of claims 5 to 10, wherein the regions are separated by diagonal lines.

12. The method of claim 11, wherein the diagonal lines comprise a first diagonal line (d_0) defined by the formula $d_0 = x + y$ and a second diagonal line (d_1) defined by the formula $d_1 = x + y$, where x and y are coordinates of the transform block.

13. The method of any of claims 1 to 12, wherein sub-blocks of the transform block corresponding to coefficient groups that do not contain the origin transform coefficient corresponding to the single available context index group.

14. The method of any of claims 1 to 13, wherein the plurality of context index groups contain a first context index group, a second context index group, and a third context index group.

15. The method of any of claims 1 to 14, wherein the first context index group comprises context indexes 12-17, the second context index group comprises context indexes 6-11, and the third context index group comprises context indexes 0-5.

16. The method of any of claims 1 to 15, wherein the transform block is an 4nx4m block, n>1 or m>1, and the sub-block is a 4x4 sub-block.

17. The method of any of claims 1 to 16, wherein the coefficient group includes at least one non-zero transform coefficient.

18. The method of any of claims 1 to 17, wherein the coding device is an encoder or a decoder.

19. A coding apparatus, comprising:

a receiver configured to receive a picture to encode or to receive a bitstream to decode;

a transmitter coupled to the receiver, the transmitter configured to transmit the bitstream to a decoder or to transmit a decoded image to a display;

a memory coupled to at least one of the receiver or the transmitter, the memory configured to store instructions; and

a processor coupled to the memory, the processor configured to execute the instructions stored in the memory to perform the method in any of claims 1 to 18.

20. The coding apparatus of claim 19, further comprising a display configured to display an image.

21. A coding apparatus, comprising:

a transform unit configured to:

select a context index group from a plurality of context index groups based on a location of a transform coefficient when a coefficient group containing a plurality of the transform coefficients contains an origin transform coefficient;

use a single available context index group when the coefficient group contains the transform coefficient does not contain the origin transform coefficient; and

select a context index from the context group that was selected or from the single available context group being used; and

a coding unit coupled to the transform unit, the coding unit configured to code using the context index that was selected.

22. The coding apparatus of claim 21, wherein the transform unit comprises a transform processing unit or an inverse transform unit.

23. The coding apparatus of claim 21, wherein the coding unit comprises an entropy coding unit or an entropy decoding unit.

24. A system, comprising:

an encoder; and

a decoder in communication with the encoder, wherein the encoder or the decoder includes the coding apparatus of any of claims 19 to 23.

25. A means for coding, comprising:

receiving means configured to receive a picture to encode or to receive a bitstream to decode;

transmission means coupled to the receiving means, the transmission means configured to transmit the bitstream to a decoder or to transmit a decoded image to a display means;

storage means coupled to at least one of the receiving means or the transmission means, the storage means configured to store instructions; and

processing means coupled to the storage means, the processing means configured to execute the instructions stored in the storage means to perform the method in any of claims 1 to 18.

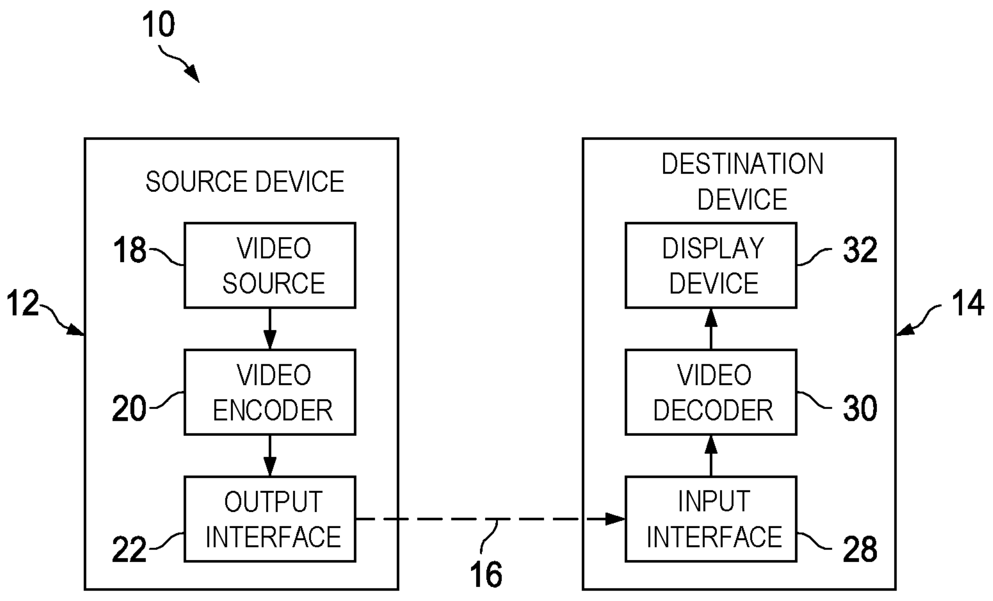
10
SOURCE DEVICE
18
VIDEO SOURCE
12
VIDEO ENCODER
20
OUTPUT INTERFACE
22
16
DESTINATION DEVICE
DISPLAY DEVICE
32
14
VIDEO DECODER
30
INPUT INTERFACE
28

FIG. 1

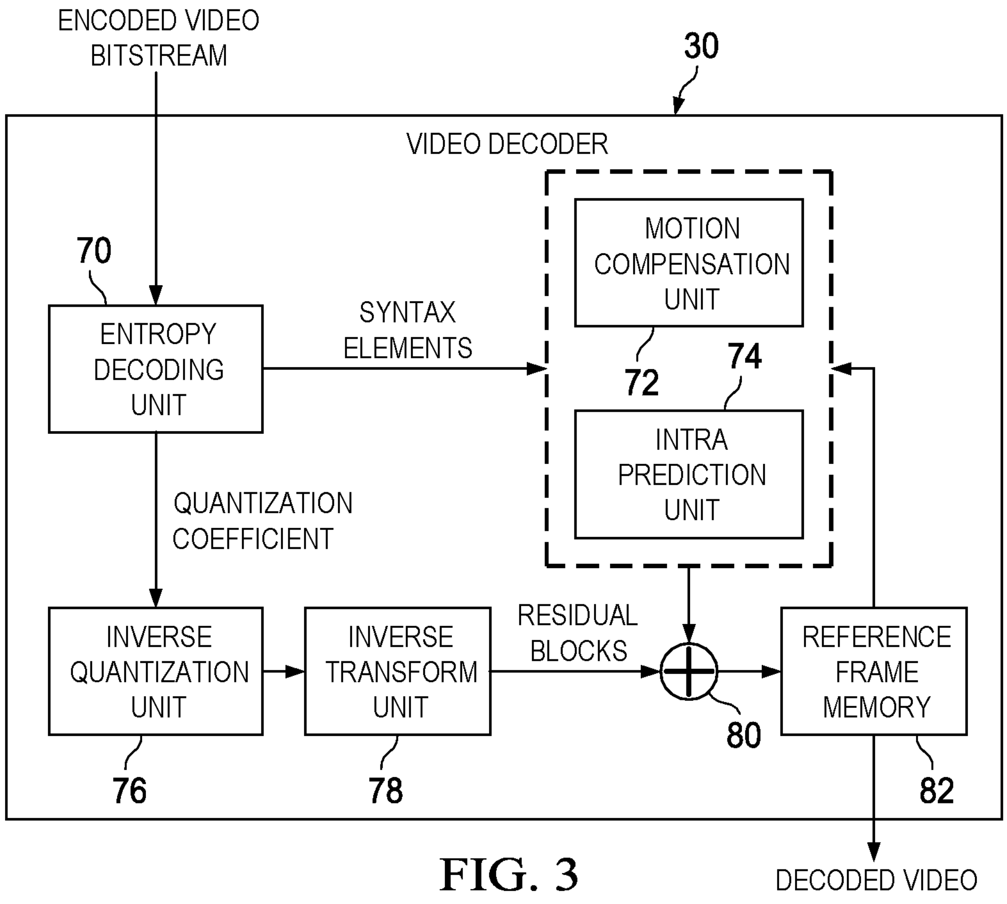
ENCODED VIDEO BITSTREAM
30
VIDEO DECODER
70
ENTROPY DECODING UNIT
SYNTAX ELEMENTS
MOTION COMPENSATION UNIT
72
74
INTRA PREDICTION UNIT
QUANTIZATION COEFFICIENT
INVERSE QUANTIZATION UNIT
76
INVERSE TRANSFORM UNIT
78
RESIDUAL BLOCKS
80
REFERENCE FRAME MEMORY
82
DECODED VIDEO

FIG. 3

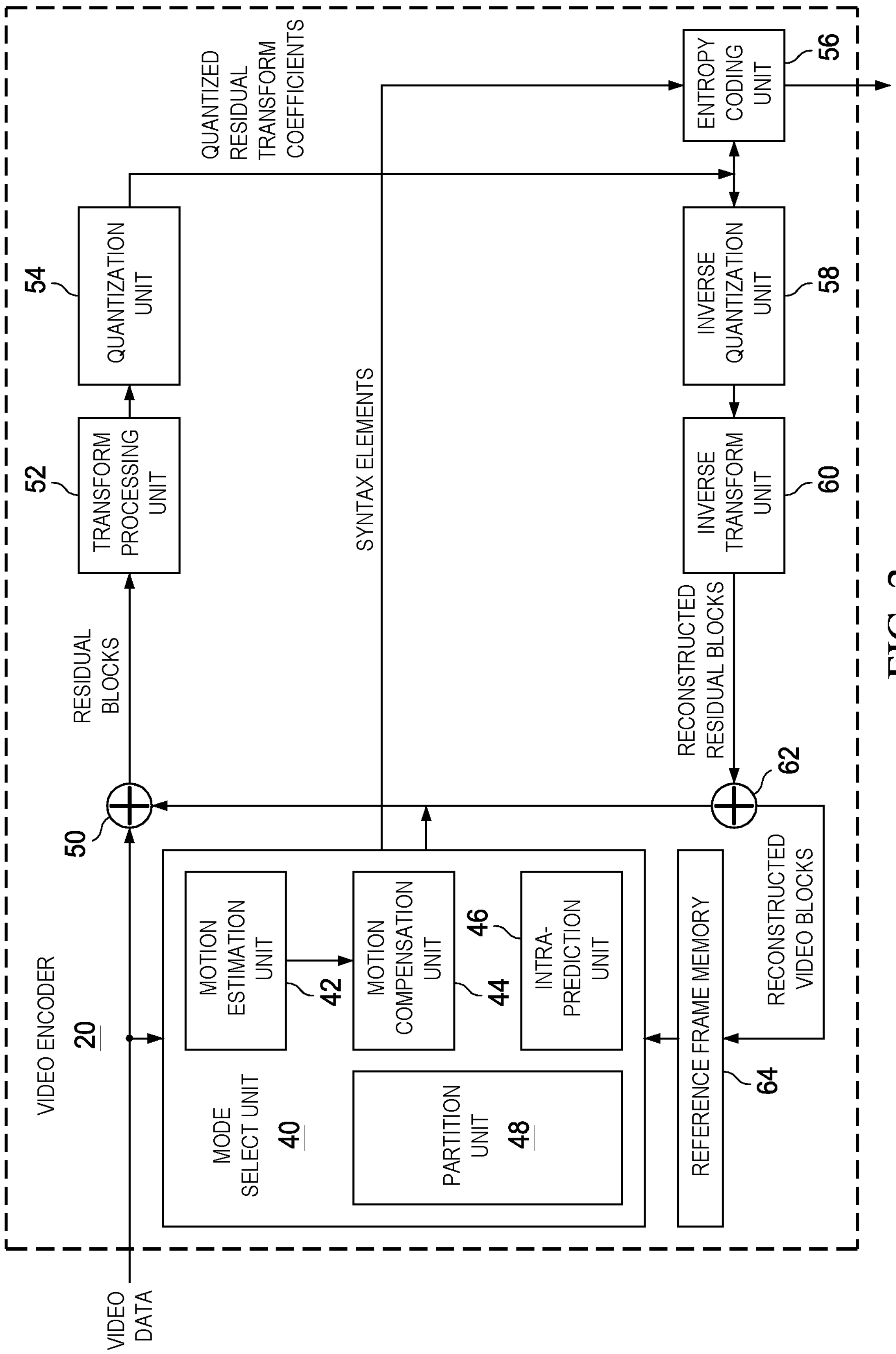
VIDEO ENCODER
20
VIDEO DATA
50
RESIDUAL BLOCKS
TRANSFORM PROCESSING UNIT
52
QUANTIZATION UNIT
54
QUANTIZED RESIDUAL TRANSFORM COEFFICIENTS
SYNTAX ELEMENTS
ENTROPY CODING UNIT
56
INVERSE QUANTIZATION UNIT
58
INVERSE TRANSFORM UNIT
60
RECONSTRUCTED RESIDUAL BLOCKS
62
RECONSTRUCTED VIDEO BLOCKS
REFERENCE FRAME MEMORY
64
MODE SELECT UNIT
40
PARTITION UNIT
48
MOTION ESTIMATION UNIT
42
MOTION COMPENSATION UNIT
44
INTRA-PREDICTION UNIT
46

FIG. 2

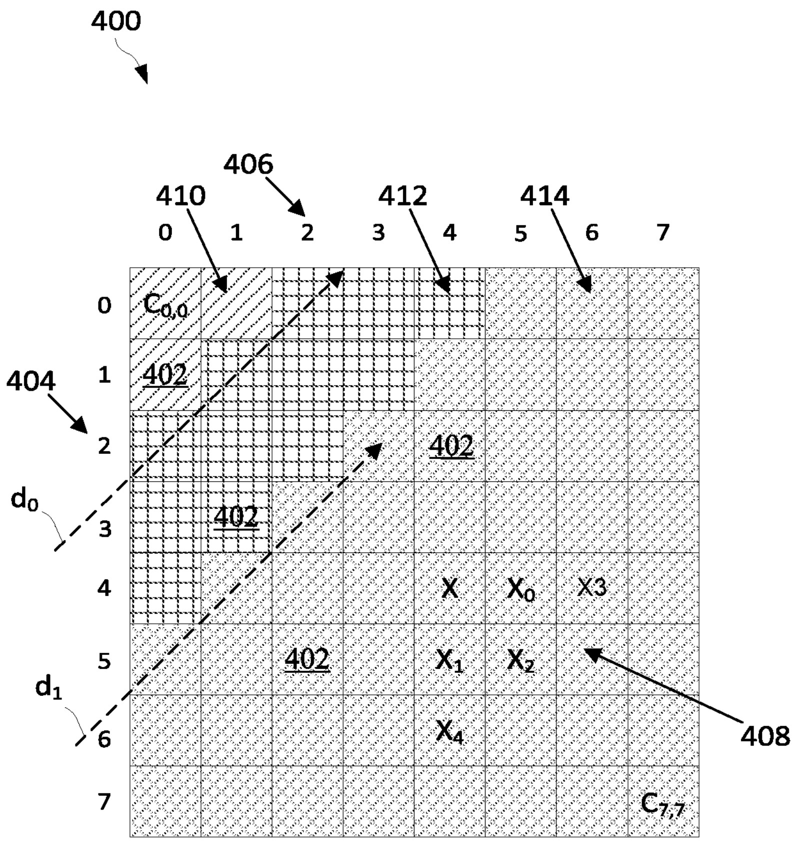
400
406
410
412
414
0
1
2
3
4
5
6
7
$C_{0,0}$
402
404
d_0
d_1
X
X_0
X3
X_1
X_2
X_4
408
$C_{7,7}$

FIG. 4

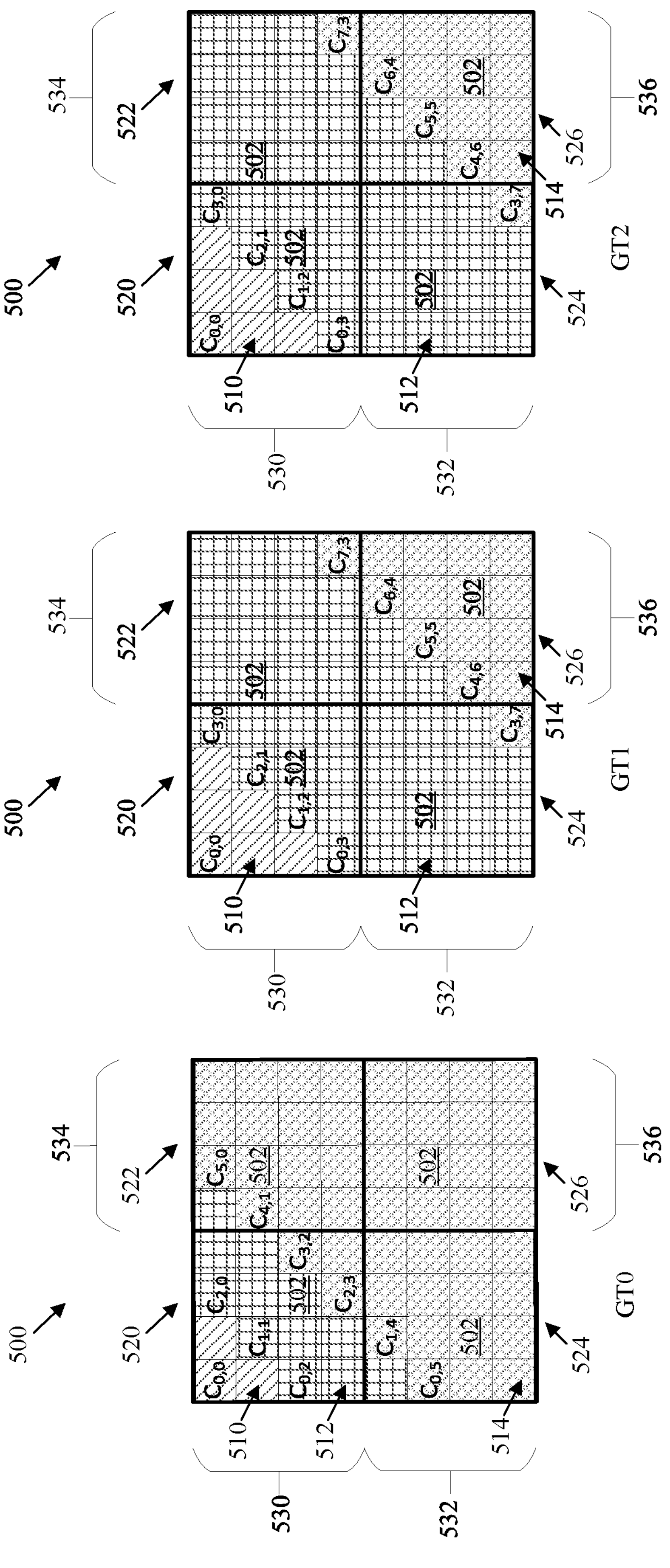
500
520
522
524
526
510
512
514
530
532
534
536
502
GT0
$C_{0,0}$
$C_{1,1}$
$C_{2,0}$
$C_{0,2}$
$C_{3,2}$
$C_{2,3}$
$C_{4,1}$
$C_{5,0}$
$C_{1,4}$
$C_{0,5}$
GT1
GT2
$C_{0,0}$
$C_{1,2}$
$C_{2,1}$
$C_{3,0}$
$C_{0,3}$
$C_{3,7}$
$C_{4,6}$
$C_{5,5}$
$C_{6,4}$
$C_{7,3}$

FIG. 5

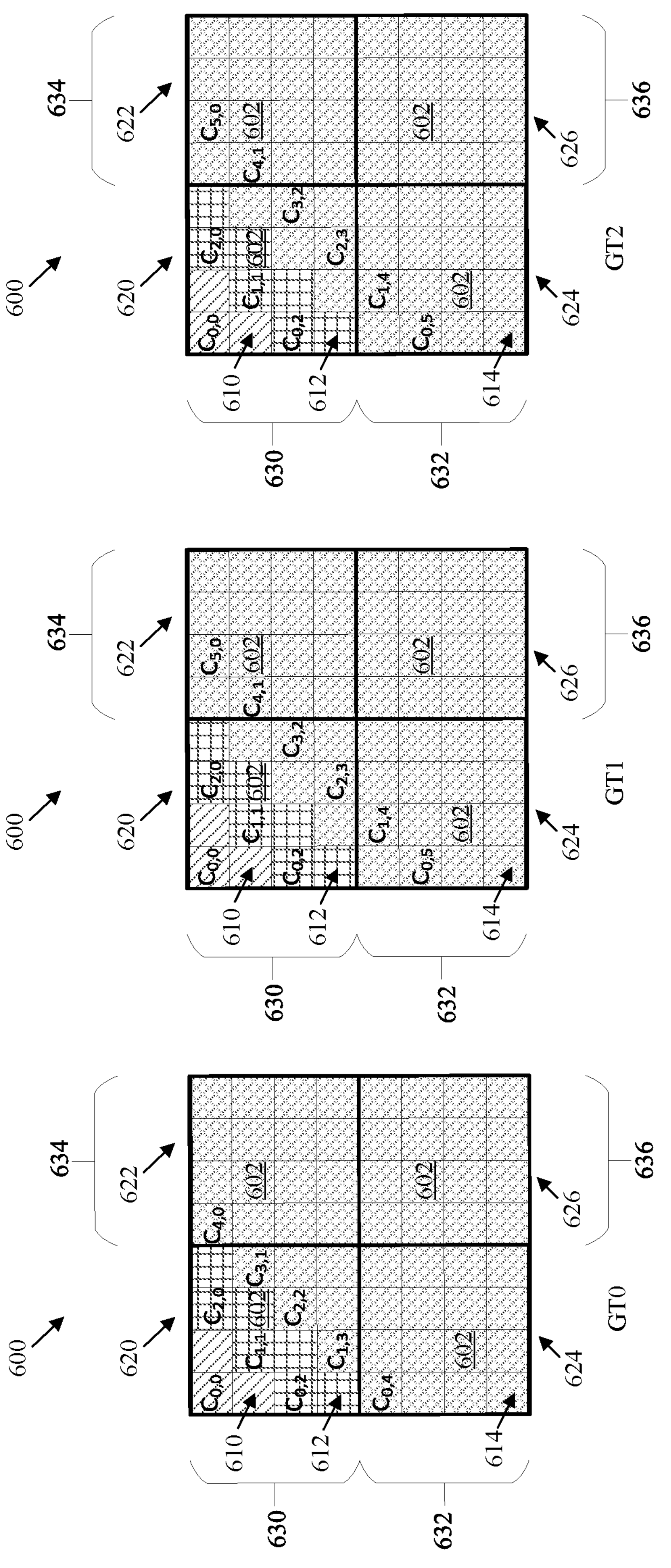
600
620
622
624
626
610
612
614
630
632
634
636
602
GT0
GT1
GT2
$C_{0,0}$
$C_{1,1}$
$C_{2,0}$
$C_{0,2}$
$C_{1,3}$
$C_{2,2}$
$C_{3,1}$
$C_{4,0}$
$C_{0,4}$
$C_{2,3}$
$C_{3,2}$
$C_{4,1}$
$C_{5,0}$
$C_{1,4}$
$C_{0,5}$

FIG. 6

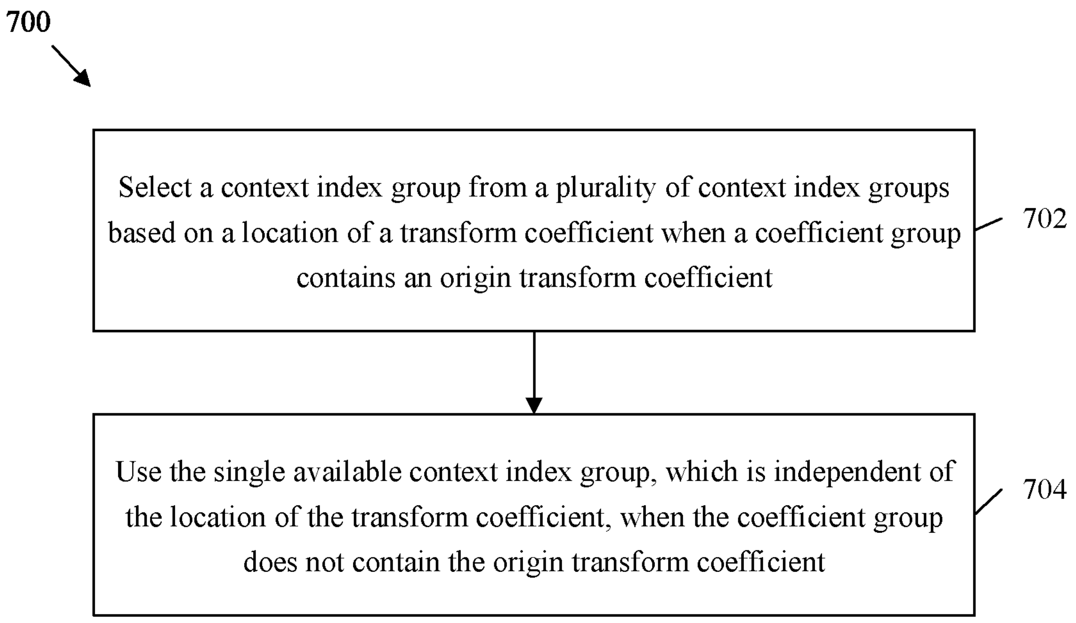
700
Select a context index group from a plurality of context index groups based on a location of a transform coefficient when a coefficient group contains an origin transform coefficient
702
Use the single available context index group, which is independent of the location of the transform coefficient, when the coefficient group does not contain the origin transform coefficient
704

FIG. 7

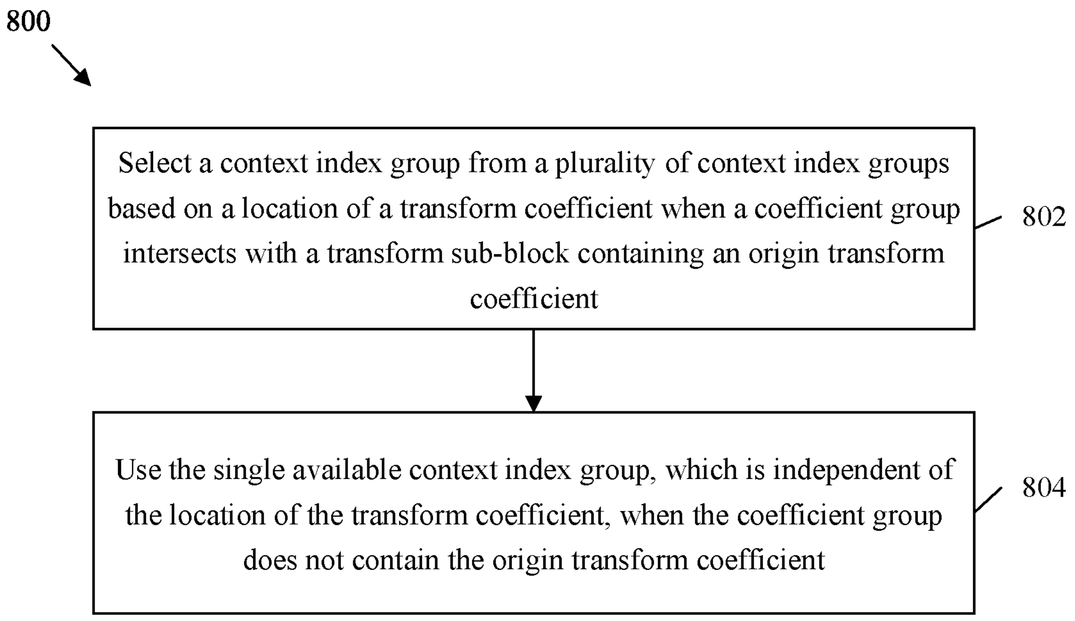
800
Select a context index group from a plurality of context index groups based on a location of a transform coefficient when a coefficient group intersects with a transform sub-block containing an origin transform coefficient
802
Use the single available context index group, which is independent of the location of the transform coefficient, when the coefficient group does not contain the origin transform coefficient
804

FIG. 8

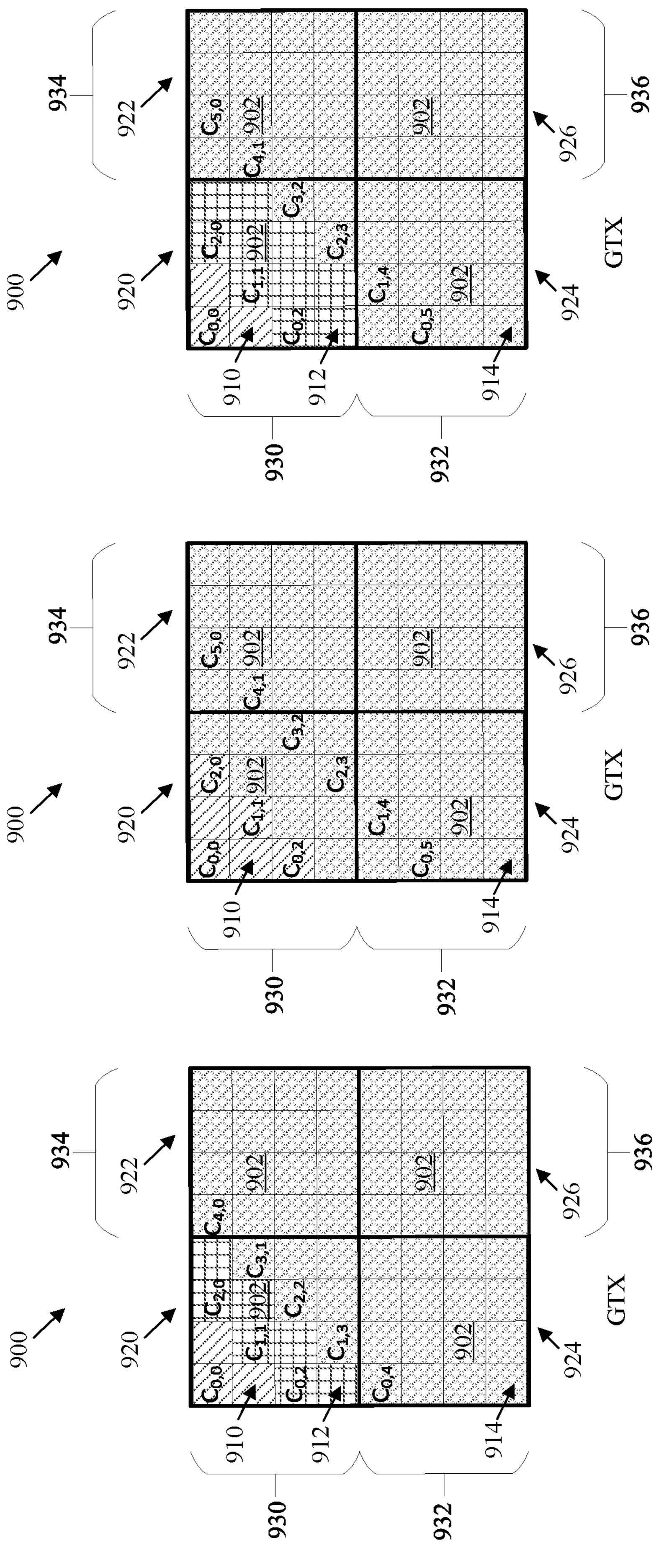
900
920
922
924
926
930
932
934
936
910
912
914
GTX
902
$C_{0,0}$
$C_{1,1}$
$C_{2,0}$
$C_{3,1}$
$C_{2,2}$
$C_{0,2}$
$C_{1,3}$
$C_{4,0}$
$C_{0,4}$
$C_{3,2}$
$C_{2,3}$
$C_{4,1}$
$C_{5,0}$
$C_{1,4}$
$C_{0,5}$

FIG. 9

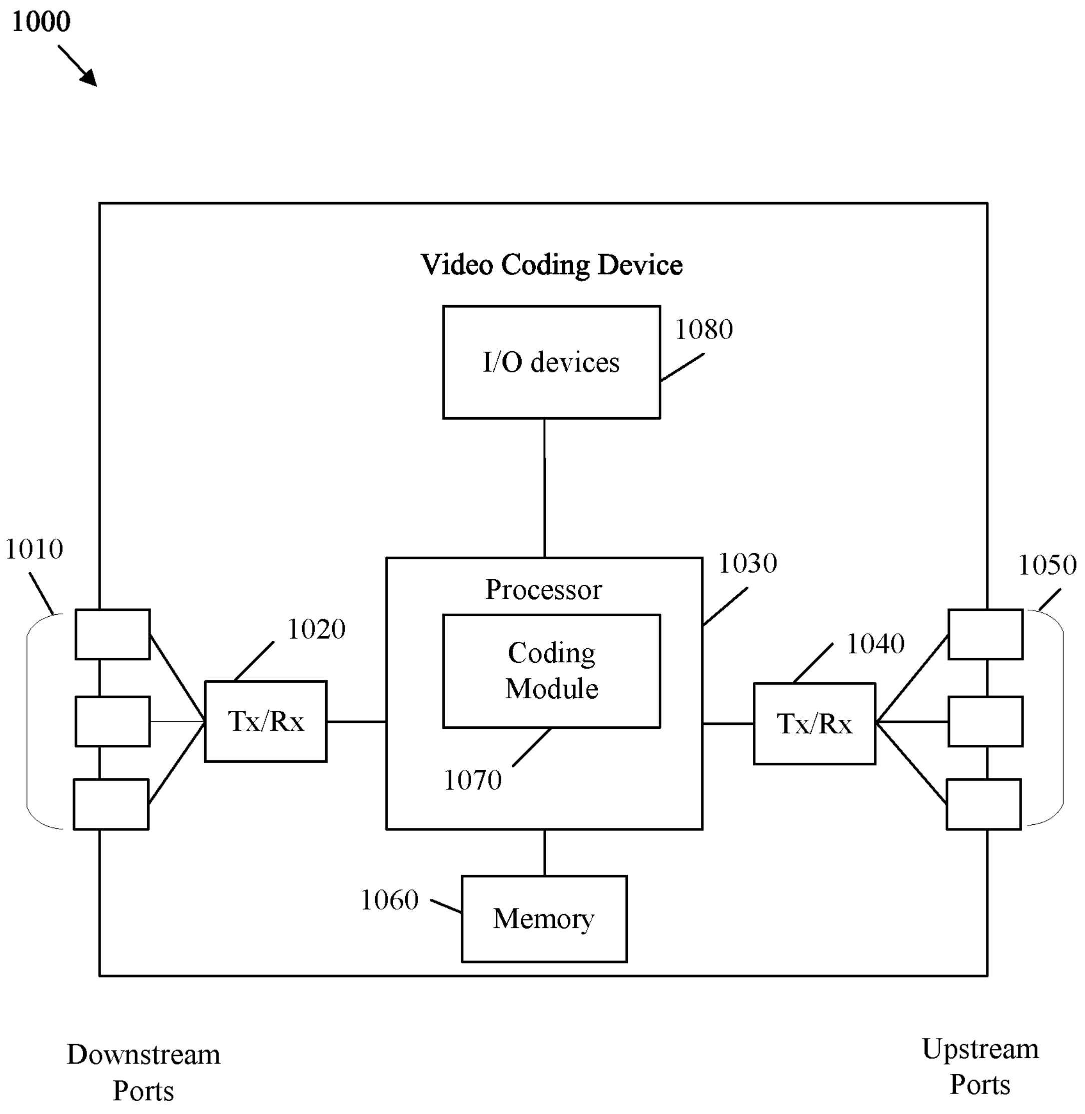
1000
Video Coding Device
I/O devices
1080
1010
Processor
1030
1050
1020
Coding Module
1040
Tx/Rx
Tx/Rx
1070
1060
Memory
Downstream Ports
Upstream Ports

FIG. 10

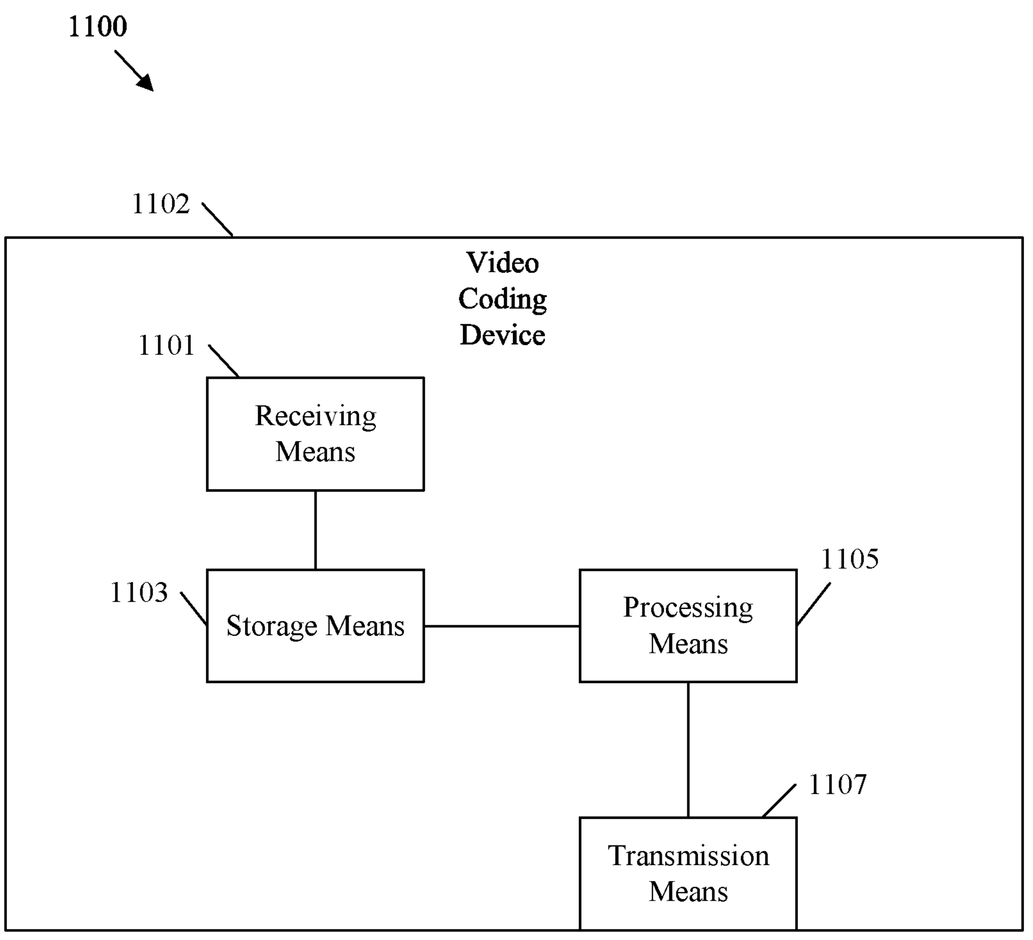
1100
1102
Video Coding Device
1101
Receiving Means
1103
Storage Means
1105
Processing Means
1107
Transmission Means

FIG. 11

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US19/26596

A. CLASSIFICATION OF SUBJECT MATTER
IPC - H04N 19/115, 19/12, 19/18, 19/48, 19/60, 19/88 (2019.01)
CPC - H04N 19/115, 19/12, 19/18, 19/48, 19/60, 19/88

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)
See Search History document

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

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	CA 2854074 A1 (SAMSUNG ELECTRONICS CO., LTD.) 10 May 2013; entire document	1-3, 21-23
A	EP 2647204 B1 (VELOS MEDIA INTERNATIONAL LIMITED) 04 April 2018; entire document	1-3, 21-23
A	US 2013/0272379 A1 (QUALCOMM INCORPORATED) 17 October 2013; entire document	1-3, 21-23

☐ 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 01 June 2019 (01.06.2019)	Date of mailing of the international search report 26 JUN 2019
Name and mailing address of the ISA/ Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-8300	Authorized officer Shane Thomas PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US19/26596

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☒ Claims Nos.: 4-20, 24, 25
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

☐ No protest accompanied the payment of additional search fees.