



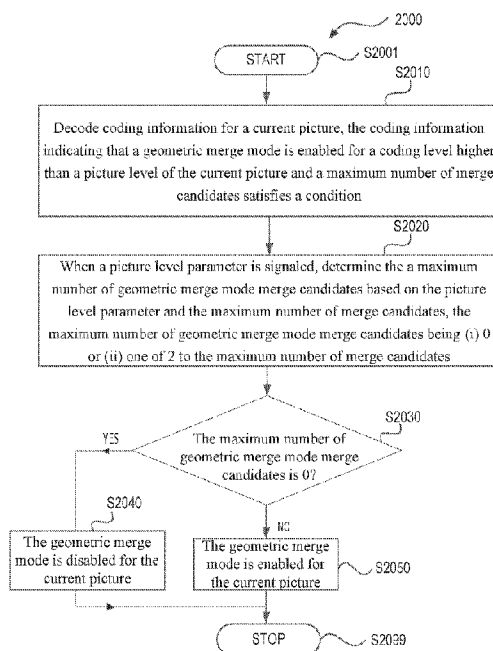
(12) **DEMANDE DE BREVET CANADIEN**  
**CANADIAN PATENT APPLICATION**

(13) **A1**

(86) Date de dépôt PCT/PCT Filing Date: 2020/12/09  
(87) Date publication PCT/PCT Publication Date: 2021/07/01  
(85) Entrée phase nationale/National Entry: 2021/10/14  
(86) N° demande PCT/PCT Application No.: US 2020/063997  
(87) N° publication PCT/PCT Publication No.: 2021/133552  
(30) Priorités/Priorities: 2019/12/28 (US62/954,473);  
2020/11/02 (US17/087,224)

(51) Cl.Int./Int.Cl. *H04N 19/105* (2014.01),  
*H04N 19/46* (2014.01)  
(71) Demandeur/Applicant:  
TENCENT AMERICA LLC, US  
(72) Inventeurs/Inventors:  
LI, LING, US;  
LI, XIANG, US;  
LI, GUICHUN, US;  
LIU, SHAN, US  
(74) Agent: BORDEN LADNER GERVAIS LLP

(54) Titre : PROCÉDE ET APPAREIL DE CODAGE VIDEO  
(54) Title: METHOD AND APPARATUS FOR VIDEO CODING



**FIG. 20**

(57) **Abrégé/Abstract:**

Aspects of the disclosure provide a method and an apparatus including processing circuitry for video decoding. The processing circuitry decodes, from a coded video bitstream, coding information for a current picture. The coding information indicates that a geometric merge mode is enabled for a coding level higher than a picture level of the current picture and a maximum number of merge candidates satisfies a condition. The processing circuitry determines, based on a picture level parameter signaled for the current picture in the coded video bitstream, a maximum number of geometric merge mode merge candidates based on the picture level parameter and the maximum number of merge candidates. The maximum number of geometric merge mode merge candidates can be (i) 0 or (ii) one of 2 to the maximum number of merge candidates. The picture level parameter indicates the maximum number of geometric merge mode merge candidates.



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

(19) World Intellectual Property  
Organization  
International Bureau

(43) International Publication Date  
01 July 2021 (01.07.2021)



(10) International Publication Number  
**WO 2021/133552 A1**

## (51) International Patent Classification:

H04N 19/105 (2014.01) H04N 19/46 (2014.01)

## (21) International Application Number:

PCT/US2020/063997

## (22) International Filing Date:

09 December 2020 (09.12.2020)

## (25) Filing Language:

English

## (26) Publication Language:

English

## (30) Priority Data:

62/954,473 28 December 2019 (28.12.2019) US  
17/087,224 02 November 2020 (02.11.2020) US

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

(72) Inventors: **LI, Ling**; c/o TENCENT AMERICA LLC, 2747 Park Boulevard, Palo Alto, California 94306 (US). **LI, Xiang**; c/o TENCENT AMERICA LLC, 2747 Park Boulevard, Palo Alto, California 94306 (US). **LI, Guichun**; c/o TENCENT AMERICA LLC, 2747 Park Boulevard, Palo Alto, California 94306 (US). **LIU, Shan**; c/o TENCENT AMERICA LLC, 2747 Park Boulevard, Palo Alto, California 94306 (US).

(74) Agent: **MA, Johnny** et al.; Oblon, McClelland, Maier & Neustadt, L.L.P., 1940 Duke Street, Alexandria, Virginia 22314 (US).

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

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

## (54) Title: METHOD AND APPARATUS FOR VIDEO CODING

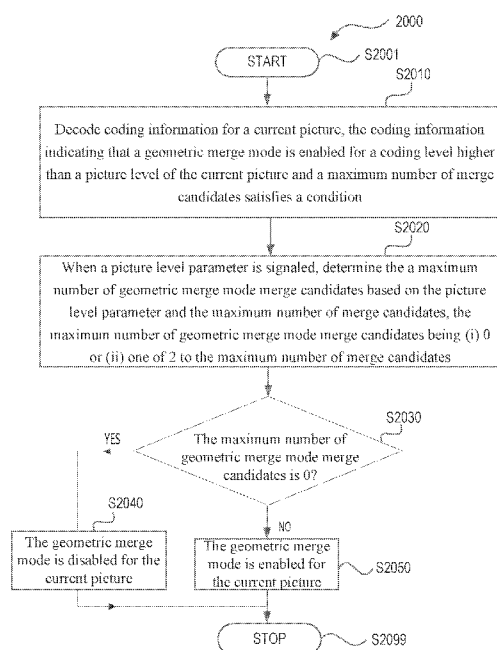


FIG. 20

(57) Abstract: Aspects of the disclosure provide a method and an apparatus including processing circuitry for video decoding. The processing circuitry decodes, from a coded video bitstream, coding information for a current picture. The coding information indicates that a geometric merge mode is enabled for a coding level higher than a picture level of the current picture and a maximum number of merge candidates satisfies a condition. The processing circuitry determines, based on a picture level parameter signaled for the current picture in the coded video bitstream, a maximum number of geometric merge mode merge candidates based on the picture level parameter and the maximum number of merge candidates. The maximum number of geometric merge mode merge candidates can be (i) 0 or (ii) one of 2 to the maximum number of merge candidates. The picture level parameter indicates the maximum number of geometric merge mode merge candidates.

# WO 2021/133552 A1

---

**Published:**

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

## METHOD AND APPARATUS FOR VIDEO CODING

INCORPORATION BY REFERENCE

[0001] This present application claims the benefit of priority to U.S. Patent Application No. 17/087,224, "METHOD AND APPARATUS FOR VIDEO CODING" filed on November 2, 2020, which claims the benefit of priority to U.S. Provisional Application No. 62/954,473, "SIGNALING OF MAXIMUM NUMBER OF TRIANGLE MERGE CANDIDATES" filed on December 28, 2019. The entire disclosures of the prior applications are hereby incorporated by reference in their entirety.

TECHNICAL FIELD

[0002] The present disclosure describes embodiments generally related to video coding.

BACKGROUND

[0003] The background description provided herein is for the purpose of generally presenting the context of the disclosure. Work of the presently named inventors, to the extent the work is described in this background section, as well as aspects of the description that may not otherwise qualify as prior art at the time of filing, are neither expressly nor impliedly admitted as prior art against the present disclosure.

[0004] Video coding and decoding can be performed using inter-picture prediction with motion compensation. Uncompressed digital video can include a series of pictures, each picture having a spatial dimension of, for example, 1920 x 1080 luminance samples and associated chrominance samples. The series of pictures can have a fixed or variable picture rate (informally also known as frame rate), of, for example 60 pictures per second or 60 Hz. Uncompressed video has significant bitrate requirements. For example, 1080p60 4:2:0 video at 8 bit per sample (1920x1080 luminance sample resolution at 60 Hz frame rate) requires close to 1.5 Gbit/s bandwidth. An hour of such video requires more than 600 GBytes of storage space.

[0005] One purpose of video coding and decoding can be the reduction of redundancy in the input video signal, through compression. Compression can help reduce the aforementioned bandwidth or storage space requirements, in some cases by two orders of magnitude or more. Both lossless and lossy compression, as well as a combination thereof can be employed. Lossless compression refers to techniques where an exact copy of the original signal can be reconstructed from the compressed original signal. When using lossy compression, the reconstructed signal may not be identical to the original signal, but the distortion between

original and reconstructed signals is small enough to make the reconstructed signal useful for the intended application. In the case of video, lossy compression is widely employed. The amount of distortion tolerated depends on the application; for example, users of certain consumer streaming applications may tolerate higher distortion than users of television distribution applications. The compression ratio achievable can reflect that: higher allowable/tolerable distortion can yield higher compression ratios.

**[0006]** Motion compensation can be a lossy compression technique and can relate to techniques where a block of sample data from a previously reconstructed picture or part thereof (reference picture), after being spatially shifted in a direction indicated by a motion vector (MV henceforth), is used for the prediction of a newly reconstructed picture or picture part. In some cases, the reference picture can be the same as the picture currently under reconstruction. MVs can have two dimensions X and Y, or three dimensions, the third being an indication of the reference picture in use (the latter, indirectly, can be a time dimension).

**[0007]** In some video compression techniques, an MV applicable to a certain area of sample data can be predicted from other MVs, for example from those related to another area of sample data spatially adjacent to the area under reconstruction, and preceding that MV in decoding order. Doing so can substantially reduce the amount of data required for coding the MV, thereby removing redundancy and increasing compression. MV prediction can work effectively, for example, because when coding an input video signal derived from a camera (known as natural video) there is a statistical likelihood that areas larger than the area to which a single MV is applicable move in a similar direction and, therefore, can in some cases be predicted using a similar motion vector derived from MVs of neighboring area. That results in the MV found for a given area to be similar or the same as the MV predicted from the surrounding MVs, and that in turn can be represented, after entropy coding, in a smaller number of bits than what would be used if coding the MV directly. In some cases, MV prediction can be an example of lossless compression of a signal (namely: the MVs) derived from the original signal (namely: the sample stream). In other cases, MV prediction itself can be lossy, for example because of rounding errors when calculating a predictor from several surrounding MVs.

**[0008]** Various MV prediction mechanisms are described in H.265/HEVC (ITU-T Rec. H.265, "High Efficiency Video Coding", December 2016). Out of the many MV prediction mechanisms that H.265 offers, described here is a technique henceforth referred to as "spatial merge".

**[0009]** Referring to FIG. 1, a current block (101) comprises samples that have been found by the encoder during the motion search process to be predictable from a previous block of the same size that has been spatially shifted. Instead of coding that MV directly, the MV can be derived from metadata associated with one or more reference pictures, for example from the most recent (in decoding order) reference picture, using the MV associated with either one of five surrounding samples, denoted A0, A1, and B0, B1, B2 (102 through 106, respectively). In H.265, the MV prediction can use predictors from the same reference picture that the neighboring block is using.

### SUMMARY

**[0010]** Aspects of the disclosure provide methods and apparatuses for video encoding and/or decoding. In some examples, an apparatus for video decoding includes processing circuitry. The processing circuitry can decode, from a coded video bitstream, coding information for a current picture. The coding information can indicate that a geometric merge mode is enabled for a coding level higher than a picture level of the current picture and a maximum number of merge candidates satisfies a condition. The processing circuitry can determine, based on a picture level parameter signaled for the current picture in the coded video bitstream, a maximum number of geometric merge mode merge candidates based on the picture level parameter and the maximum number of merge candidates. The maximum number of geometric merge mode merge candidates can be (i) 0 or (ii) one of 2 to the maximum number of merge candidates. The picture level parameter can indicate the maximum number of geometric merge mode merge candidates. The geometric merge mode is disabled for the current picture based on the maximum number of geometric merge mode merge candidates being 0, and the geometric merge mode is enabled for the current picture based on the maximum number of geometric merge mode merge candidates not being 0.

**[0011]** In an embodiment, the geometric merge mode is a triangle partition mode (TPM) and the maximum number of geometric merge mode merge candidates is a maximum number of TPM merge candidates.

**[0012]** In an embodiment, the coding level is a sequence level.

**[0013]** In an embodiment, the condition is the maximum number of merge candidates is larger than or equal to 2.

**[0014]** In an embodiment, the condition is the maximum number of merge candidates is larger than or equal to 2. The processing circuitry can determine the maximum number of TPM

merge candidates by subtracting the picture level parameter from the maximum number of merge candidates.

**[0015]** In an embodiment, a picture parameter set (PPS) level parameter that indicates the maximum number of TPM merge candidates is signaled in the coded video bitstream for a PPS associated with the current picture. The PPS level parameter is (i) one of 0 to (the maximum number of merge candidates - 1) or (ii) (the maximum number of merge candidates + 1).

**[0016]** In an embodiment, a picture parameter set (PPS) level parameter indicating the maximum number of TPM merge candidates is not signaled in the coded video bitstream for a PPS associated with the current picture.

**[0017]** In an embodiment, the coded video bitstream includes a picture header for the current picture. The picture level parameter is signaled in the picture header based on the TPM being enabled for the sequence level and the maximum number of merge candidates being larger than or equal to 2, the signaling of the picture level parameter being independent from the PPS level parameter.

**[0018]** In an embodiment, the coded video bitstream includes a picture parameter set (PPS) associated with the current picture. A PPS level parameter indicating the maximum number of TPM merge candidates is signaled in the PPS based at least on a PPS level flag indicating that the PPS level parameter is to be signaled.

**[0019]** In an embodiment, the coded video bitstream includes a picture header for the current picture. The picture level parameter is signaled in the picture header based on the TPM being enabled for the sequence level, the maximum number of merge candidates being larger than or equal to 2, and the PPS level flag indicating that the PPS level parameter is not to be signaled.

**[0020]** In an embodiment, the coded video bitstream includes a picture parameter set (PPS) associated with the current picture. A PPS level parameter indicating the maximum number of TPM merge candidates is signaled in the PPS based at least on the TPM being enabled for the sequence level.

**[0021]** In some examples, an apparatus for video decoding includes processing circuitry. The processing circuitry can decode, from a coded video bitstream, coding information for a current picture. The coding information can indicate a geometric merge mode being enabled at a sequence level, a picture parameter set (PPS) level parameter in a PPS being 0, and a maximum number of merge candidates. The PPS level parameter can indicate a maximum number of

geometric merge mode merge candidates. The processing circuitry can decode a picture level parameter signaled for the current picture in the coded video bitstream based on the maximum number of merge candidates satisfying a condition, the picture level parameter indicating the maximum number of geometric merge mode merge candidates.

**[0022]** In an embodiment, the geometric merge mode is a triangle partition mode (TPM) and the maximum number of geometric merge mode merge candidates is a maximum number of TPM merge candidates.

**[0023]** In an embodiment, the condition is one of (i) the maximum number of merge candidates is larger than 2 and (ii) the maximum number of merge candidates is larger than or equal to 3.

**[0024]** In an embodiment, the maximum number of merge candidates is 2 and does not satisfy the condition, and the picture level parameter is not signaled in the coded video bitstream. The processing circuitry can determine that the maximum number of TPM merge candidates is 2.

**[0025]** Aspects of the disclosure also provide a non-transitory computer-readable medium storing instructions which when executed by a computer for video decoding cause the computer to perform the method for video decoding and/or encoding.

#### BRIEF DESCRIPTION OF THE DRAWINGS

**[0026]** Further features, the nature, and various advantages of the disclosed subject matter will be more apparent from the following detailed description and the accompanying drawings in which:

**[0027]** FIG. 1 is a schematic illustration of a current block and its surrounding spatial merge candidates in one example.

**[0028]** FIG. 2 is a schematic illustration of a simplified block diagram of a communication system (200) in accordance with an embodiment.

**[0029]** FIG. 3 is a schematic illustration of a simplified block diagram of a communication system (300) in accordance with an embodiment.

**[0030]** FIG. 4 is a schematic illustration of a simplified block diagram of a decoder in accordance with an embodiment.

**[0031]** FIG. 5 is a schematic illustration of a simplified block diagram of an encoder in accordance with an embodiment.

**[0032]** FIG. 6 shows a block diagram of an encoder in accordance with another embodiment.



[0033] FIG. 7 shows a block diagram of a decoder in accordance with another embodiment.

[0034] FIGs. 8A-8B show examples of triangle partition based inter prediction according to an embodiment of the disclosure.

[0035] FIG. 9 shows an exemplary geometric merge mode.

[0036] FIG. 10 shows an exemplary sequence level control of the TPM.

[0037] FIG. 11 shows exemplary picture parameter set (PPS) syntax.

[0038] FIG. 12 shows exemplary picture header syntax.

[0039] FIG. 13 shows exemplary picture header syntax.

[0040] FIG. 14 shows exemplary picture header syntax.

[0041] FIG. 15 shows exemplary PPS syntax.

[0042] FIG. 16A shows exemplary picture header syntax.

[0043] FIG. 16B shows exemplary syntax that is not applicable when signaling of a picture level parameter is independent from a PPS level parameter.

[0044] FIG. 17 shows exemplary PPS syntax.

[0045] FIG. 18 shows exemplary picture header syntax.

[0046] FIG. 19 shows exemplary PPS syntax.

[0047] FIG. 20 shows a flow chart outlining a process (2000) according to an embodiment of the disclosure.

[0048] FIG. 21 shows a flow chart outlining a process (2100) according to an embodiment of the disclosure.

[0049] FIG. 22 is a schematic illustration of a computer system in accordance with an embodiment.

#### DETAILED DESCRIPTION OF EMBODIMENTS

[0050] FIG. 2 illustrates a simplified block diagram of a communication system (200) according to an embodiment of the present disclosure. The communication system (200) includes a plurality of terminal devices that can communicate with each other, via, for example, a network (250). For example, the communication system (200) includes a first pair of terminal devices (210) and (220) interconnected via the network (250). In the FIG. 2 example, the first pair of terminal devices (210) and (220) performs unidirectional transmission of data. For example, the terminal device (210) may code video data (e.g., a stream of video pictures that are captured by the terminal device (210)) for transmission to the other terminal device (220) via the

network (250). The encoded video data can be transmitted in the form of one or more coded video bitstreams. The terminal device (220) may receive the coded video data from the network (250), decode the coded video data to recover the video pictures and display video pictures according to the recovered video data. Unidirectional data transmission may be common in media serving applications and the like.

**[0051]** In another example, the communication system (200) includes a second pair of terminal devices (230) and (240) that performs bidirectional transmission of coded video data that may occur, for example, during videoconferencing. For bidirectional transmission of data, in an example, each terminal device of the terminal devices (230) and (240) may code video data (e.g., a stream of video pictures that are captured by the terminal device) for transmission to the other terminal device of the terminal devices (230) and (240) via the network (250). Each terminal device of the terminal devices (230) and (240) also may receive the coded video data transmitted by the other terminal device of the terminal devices (230) and (240), and may decode the coded video data to recover the video pictures and may display video pictures at an accessible display device according to the recovered video data.

**[0052]** In the FIG. 2 example, the terminal devices (210), (220), (230) and (240) may be illustrated as servers, personal computers and smart phones but the principles of the present disclosure may be not so limited. Embodiments of the present disclosure find application with laptop computers, tablet computers, media players and/or dedicated video conferencing equipment. The network (250) represents any number of networks that convey coded video data among the terminal devices (210), (220), (230) and (240), including for example wireline (wired) and/or wireless communication networks. The communication network (250) may exchange data in circuit-switched and/or packet-switched channels. Representative networks include telecommunications networks, local area networks, wide area networks and/or the Internet. For the purposes of the present discussion, the architecture and topology of the network (250) may be immaterial to the operation of the present disclosure unless explained herein below.

**[0053]** FIG. 3 illustrates, as an example for an application for the disclosed subject matter, the placement of a video encoder and a video decoder in a streaming environment. The disclosed subject matter can be equally applicable to other video enabled applications, including, for example, video conferencing, digital TV, storing of compressed video on digital media including CD, DVD, memory stick and the like, and so on.

**[0054]** A streaming system may include a capture subsystem (313), that can include a video source (301), for example a digital camera, creating for example a stream of video pictures (302) that are uncompressed. In an example, the stream of video pictures (302) includes samples that are taken by the digital camera. The stream of video pictures (302), depicted as a bold line to emphasize a high data volume when compared to encoded video data (304) (or coded video bitstreams), can be processed by an electronic device (320) that includes a video encoder (303) coupled to the video source (301). The video encoder (303) can include hardware, software, or a combination thereof to enable or implement aspects of the disclosed subject matter as described in more detail below. The encoded video data (304) (or encoded video bitstream (304)), depicted as a thin line to emphasize the lower data volume when compared to the stream of video pictures (302), can be stored on a streaming server (305) for future use. One or more streaming client subsystems, such as client subsystems (306) and (308) in FIG. 3 can access the streaming server (305) to retrieve copies (307) and (309) of the encoded video data (304). A client subsystem (306) can include a video decoder (310), for example, in an electronic device (330). The video decoder (310) decodes the incoming copy (307) of the encoded video data and creates an outgoing stream of video pictures (311) that can be rendered on a display (312) (e.g., display screen) or other rendering device (not depicted). In some streaming systems, the encoded video data (304), (307), and (309) (e.g., video bitstreams) can be encoded according to certain video coding/compression standards. Examples of those standards include ITU-T Recommendation H.265. In an example, a video coding standard under development is informally known as Versatile Video Coding (VVC). The disclosed subject matter may be used in the context of VVC.

**[0055]** It is noted that the electronic devices (320) and (330) can include other components (not shown). For example, the electronic device (320) can include a video decoder (not shown) and the electronic device (330) can include a video encoder (not shown) as well.

**[0056]** FIG. 4 shows a block diagram of a video decoder (410) according to an embodiment of the present disclosure. The video decoder (410) can be included in an electronic device (430). The electronic device (430) can include a receiver (431) (e.g., receiving circuitry). The video decoder (410) can be used in the place of the video decoder (310) in the FIG. 3 example.

**[0057]** The receiver (431) may receive one or more coded video sequences to be decoded by the video decoder (410); in the same or another embodiment, one coded video sequence at a

time, where the decoding of each coded video sequence is independent from other coded video sequences. The coded video sequence may be received from a channel (401), which may be a hardware/software link to a storage device which stores the encoded video data. The receiver (431) may receive the encoded video data with other data, for example, coded audio data and/or ancillary data streams, that may be forwarded to their respective using entities (not depicted). The receiver (431) may separate the coded video sequence from the other data. To combat network jitter, a buffer memory (415) may be coupled in between the receiver (431) and an entropy decoder / parser (420) ("parser (420)" henceforth). In certain applications, the buffer memory (415) is part of the video decoder (410). In others, it can be outside of the video decoder (410) (not depicted). In still others, there can be a buffer memory (not depicted) outside of the video decoder (410), for example to combat network jitter, and in addition another buffer memory (415) inside the video decoder (410), for example to handle playout timing. When the receiver (431) is receiving data from a store/forward device of sufficient bandwidth and controllability, or from an isosynchronous network, the buffer memory (415) may not be needed, or can be small. For use on best effort packet networks such as the Internet, the buffer memory (415) may be required, can be comparatively large and can be advantageously of adaptive size, and may at least partially be implemented in an operating system or similar elements (not depicted) outside of the video decoder (410).

**[0058]** The video decoder (410) may include the parser (420) to reconstruct symbols (421) from the coded video sequence. Categories of those symbols include information used to manage operation of the video decoder (410), and potentially information to control a rendering device such as a render device (412) (e.g., a display screen) that is not an integral part of the electronic device (430) but can be coupled to the electronic device (430), as was shown in FIG. 4. The control information for the rendering device(s) may be in the form of Supplemental Enhancement Information (SEI messages) or Video Usability Information (VUI) parameter set fragments (not depicted). The parser (420) may parse / entropy-decode the coded video sequence that is received. The coding of the coded video sequence can be in accordance with a video coding technology or standard, and can follow various principles, including variable length coding, Huffman coding, arithmetic coding with or without context sensitivity, and so forth. The parser (420) may extract from the coded video sequence, a set of subgroup parameters for at least one of the subgroups of pixels in the video decoder, based upon at least one parameter corresponding to the group. Subgroups can include Groups of Pictures (GOPs), pictures, tiles,

slices, macroblocks, Coding Units (CUs), blocks, Transform Units (TUs), Prediction Units (PUs) and so forth. The parser (420) may also extract from the coded video sequence information such as transform coefficients, quantizer parameter values, motion vectors, and so forth.

**[0059]** The parser (420) may perform an entropy decoding / parsing operation on the video sequence received from the buffer memory (415), so as to create symbols (421).

**[0060]** Reconstruction of the symbols (421) can involve multiple different units depending on the type of the coded video picture or parts thereof (such as: inter and intra picture, inter and intra block), and other factors. Which units are involved, and how, can be controlled by the subgroup control information that was parsed from the coded video sequence by the parser (420). The flow of such subgroup control information between the parser (420) and the multiple units below is not depicted for clarity.

**[0061]** Beyond the functional blocks already mentioned, the video decoder (410) can be conceptually subdivided into a number of functional units as described below. In a practical implementation operating under commercial constraints, many of these units interact closely with each other and can, at least partly, be integrated into each other. However, for the purpose of describing the disclosed subject matter, the conceptual subdivision into the functional units below is appropriate.

**[0062]** A first unit is the scaler / inverse transform unit (451). The scaler / inverse transform unit (451) receives a quantized transform coefficient as well as control information, including which transform to use, block size, quantization factor, quantization scaling matrices, etc. as symbol(s) (421) from the parser (420). The scaler / inverse transform unit (451) can output blocks comprising sample values, that can be input into aggregator (455).

**[0063]** In some cases, the output samples of the scaler / inverse transform (451) can pertain to an intra coded block; that is: a block that is not using predictive information from previously reconstructed pictures, but can use predictive information from previously reconstructed parts of the current picture. Such predictive information can be provided by an intra picture prediction unit (452). In some cases, the intra picture prediction unit (452) generates a block of the same size and shape of the block under reconstruction, using surrounding already reconstructed information fetched from the current picture buffer (458). The current picture buffer (458) buffers, for example, partly reconstructed current picture and/or fully reconstructed current picture. The aggregator (455), in some cases, adds, on a per sample

basis, the prediction information the intra prediction unit (452) has generated to the output sample information as provided by the scaler / inverse transform unit (451).

**[0064]** In other cases, the output samples of the scaler / inverse transform unit (451) can pertain to an inter coded, and potentially motion compensated block. In such a case, a motion compensation prediction unit (453) can access reference picture memory (457) to fetch samples used for prediction. After motion compensating the fetched samples in accordance with the symbols (421) pertaining to the block, these samples can be added by the aggregator (455) to the output of the scaler / inverse transform unit (451) (in this case called the residual samples or residual signal) so as to generate output sample information. The addresses within the reference picture memory (457) from where the motion compensation prediction unit (453) fetches prediction samples can be controlled by motion vectors, available to the motion compensation prediction unit (453) in the form of symbols (421) that can have, for example X, Y, and reference picture components. Motion compensation also can include interpolation of sample values as fetched from the reference picture memory (457) when sub-sample exact motion vectors are in use, motion vector prediction mechanisms, and so forth.

**[0065]** The output samples of the aggregator (455) can be subject to various loop filtering techniques in the loop filter unit (456). Video compression technologies can include in-loop filter technologies that are controlled by parameters included in the coded video sequence (also referred to as coded video bitstream) and made available to the loop filter unit (456) as symbols (421) from the parser (420), but can also be responsive to meta-information obtained during the decoding of previous (in decoding order) parts of the coded picture or coded video sequence, as well as responsive to previously reconstructed and loop-filtered sample values.

**[0066]** The output of the loop filter unit (456) can be a sample stream that can be output to the render device (412) as well as stored in the reference picture memory (457) for use in future inter-picture prediction.

**[0067]** Certain coded pictures, once fully reconstructed, can be used as reference pictures for future prediction. For example, once a coded picture corresponding to a current picture is fully reconstructed and the coded picture has been identified as a reference picture (by, for example, the parser (420)), the current picture buffer (458) can become a part of the reference picture memory (457), and a fresh current picture buffer can be reallocated before commencing the reconstruction of the following coded picture.

**[0068]** The video decoder (410) may perform decoding operations according to a predetermined video compression technology in a standard, such as ITU-T Rec. H.265. The coded video sequence may conform to a syntax specified by the video compression technology or standard being used, in the sense that the coded video sequence adheres to both the syntax of the video compression technology or standard and the profiles as documented in the video compression technology or standard. Specifically, a profile can select certain tools as the only tools available for use under that profile from all the tools available in the video compression technology or standard. Also necessary for compliance can be that the complexity of the coded video sequence is within bounds as defined by the level of the video compression technology or standard. In some cases, levels restrict the maximum picture size, maximum frame rate, maximum reconstruction sample rate (measured in, for example megasamples per second), maximum reference picture size, and so on. Limits set by levels can, in some cases, be further restricted through Hypothetical Reference Decoder (HRD) specifications and metadata for HRD buffer management signaled in the coded video sequence.

**[0069]** In an embodiment, the receiver (431) may receive additional (redundant) data with the encoded video. The additional data may be included as part of the coded video sequence(s). The additional data may be used by the video decoder (410) to properly decode the data and/or to more accurately reconstruct the original video data. Additional data can be in the form of, for example, temporal, spatial, or signal noise ratio (SNR) enhancement layers, redundant slices, redundant pictures, forward error correction codes, and so on.

**[0070]** FIG. 5 shows a block diagram of a video encoder (503) according to an embodiment of the present disclosure. The video encoder (503) is included in an electronic device (520). The electronic device (520) includes a transmitter (540) (e.g., transmitting circuitry). The video encoder (503) can be used in the place of the video encoder (303) in the FIG. 3 example.

**[0071]** The video encoder (503) may receive video samples from a video source (501) (that is not part of the electronic device (520) in the FIG. 5 example) that may capture video image(s) to be coded by the video encoder (503). In another example, the video source (501) is a part of the electronic device (520).

**[0072]** The video source (501) may provide the source video sequence to be coded by the video encoder (503) in the form of a digital video sample stream that can be of any suitable bit depth (for example: 8 bit, 10 bit, 12 bit, ...), any colorspace (for example, BT.601 Y CrCb,

RGB, ...), and any suitable sampling structure (for example Y CrCb 4:2:0, Y CrCb 4:4:4). In a media serving system, the video source (501) may be a storage device storing previously prepared video. In a videoconferencing system, the video source (501) may be a camera that captures local image information as a video sequence. Video data may be provided as a plurality of individual pictures that impart motion when viewed in sequence. The pictures themselves may be organized as a spatial array of pixels, wherein each pixel can comprise one or more samples depending on the sampling structure, color space, etc. in use. A person skilled in the art can readily understand the relationship between pixels and samples. The description below focuses on samples.

**[0073]** According to an embodiment, the video encoder (503) may code and compress the pictures of the source video sequence into a coded video sequence (543) in real time or under any other time constraints as required by the application. Enforcing appropriate coding speed is one function of a controller (550). In some embodiments, the controller (550) controls other functional units as described below and is functionally coupled to the other functional units. The coupling is not depicted for clarity. Parameters set by the controller (550) can include rate control related parameters (picture skip, quantizer, lambda value of rate-distortion optimization techniques, ...), picture size, group of pictures (GOP) layout, maximum motion vector search range, and so forth. The controller (550) can be configured to have other suitable functions that pertain to the video encoder (503) optimized for a certain system design.

**[0074]** In some embodiments, the video encoder (503) is configured to operate in a coding loop. As an oversimplified description, in an example, the coding loop can include a source coder (530) (e.g., responsible for creating symbols, such as a symbol stream, based on an input picture to be coded, and a reference picture(s)), and a (local) decoder (533) embedded in the video encoder (503). The decoder (533) reconstructs the symbols to create the sample data in a similar manner as a (remote) decoder also would create (as any compression between symbols and coded video bitstream is lossless in the video compression technologies considered in the disclosed subject matter). The reconstructed sample stream (sample data) is input to the reference picture memory (534). As the decoding of a symbol stream leads to bit-exact results independent of decoder location (local or remote), the content in the reference picture memory (534) is also bit exact between the local encoder and remote encoder. In other words, the prediction part of an encoder "sees" as reference picture samples exactly the same sample values as a decoder would "see" when using prediction during decoding. This fundamental principle of



reference picture synchronicity (and resulting drift, if synchronicity cannot be maintained, for example because of channel errors) is used in some related arts as well.

**[0075]** The operation of the "local" decoder (533) can be the same as of a "remote" decoder, such as the video decoder (410), which has already been described in detail above in conjunction with FIG. 4. Briefly referring also to FIG. 4, however, as symbols are available and encoding/decoding of symbols to a coded video sequence by an entropy coder (545) and the parser (420) can be lossless, the entropy decoding parts of the video decoder (410), including the buffer memory (415), and parser (420) may not be fully implemented in the local decoder (533).

**[0076]** An observation that can be made at this point is that any decoder technology except the parsing/entropy decoding that is present in a decoder also necessarily needs to be present, in substantially identical functional form, in a corresponding encoder. For this reason, the disclosed subject matter focuses on decoder operation. The description of encoder technologies can be abbreviated as they are the inverse of the comprehensively described decoder technologies. Only in certain areas a more detail description is required and provided below.

**[0077]** During operation, in some examples, the source coder (530) may perform motion compensated predictive coding, which codes an input picture predictively with reference to one or more previously coded picture from the video sequence that were designated as "reference pictures." In this manner, the coding engine (532) codes differences between pixel blocks of an input picture and pixel blocks of reference picture(s) that may be selected as prediction reference(s) to the input picture.

**[0078]** The local video decoder (533) may decode coded video data of pictures that may be designated as reference pictures, based on symbols created by the source coder (530). Operations of the coding engine (532) may advantageously be lossy processes. When the coded video data may be decoded at a video decoder (not shown in FIG. 5), the reconstructed video sequence typically may be a replica of the source video sequence with some errors. The local video decoder (533) replicates decoding processes that may be performed by the video decoder on reference pictures and may cause reconstructed reference pictures to be stored in the reference picture cache (534). In this manner, the video encoder (503) may store copies of reconstructed reference pictures locally that have common content as the reconstructed reference pictures that will be obtained by a far-end video decoder (absent transmission errors).

**[0079]** The predictor (535) may perform prediction searches for the coding engine (532). That is, for a new picture to be coded, the predictor (535) may search the reference picture memory (534) for sample data (as candidate reference pixel blocks) or certain metadata such as reference picture motion vectors, block shapes, and so on, that may serve as an appropriate prediction reference for the new pictures. The predictor (535) may operate on a sample block-by-pixel block basis to find appropriate prediction references. In some cases, as determined by search results obtained by the predictor (535), an input picture may have prediction references drawn from multiple reference pictures stored in the reference picture memory (534).

**[0080]** The controller (550) may manage coding operations of the source coder (530), including, for example, setting of parameters and subgroup parameters used for encoding the video data.

**[0081]** Output of all aforementioned functional units may be subjected to entropy coding in the entropy coder (545). The entropy coder (545) translates the symbols as generated by the various functional units into a coded video sequence, by lossless compressing the symbols according to technologies such as Huffman coding, variable length coding, arithmetic coding, and so forth.

**[0082]** The transmitter (540) may buffer the coded video sequence(s) as created by the entropy coder (545) to prepare for transmission via a communication channel (560), which may be a hardware/software link to a storage device which would store the encoded video data. The transmitter (540) may merge coded video data from the video coder (503) with other data to be transmitted, for example, coded audio data and/or ancillary data streams (sources not shown).

**[0083]** The controller (550) may manage operation of the video encoder (503). During coding, the controller (550) may assign to each coded picture a certain coded picture type, which may affect the coding techniques that may be applied to the respective picture. For example, pictures often may be assigned as one of the following picture types:

**[0084]** An Intra Picture (I picture) may be one that may be coded and decoded without using any other picture in the sequence as a source of prediction. Some video codecs allow for different types of intra pictures, including, for example Independent Decoder Refresh (“IDR”) Pictures. A person skilled in the art is aware of those variants of I pictures and their respective applications and features.

**[0085]** A predictive picture (P picture) may be one that may be coded and decoded using intra prediction or inter prediction using at most one motion vector and reference index to predict the sample values of each block.

**[0086]** A bi-directionally predictive picture (B Picture) may be one that may be coded and decoded using intra prediction or inter prediction using at most two motion vectors and reference indices to predict the sample values of each block. Similarly, multiple-predictive pictures can use more than two reference pictures and associated metadata for the reconstruction of a single block.

**[0087]** Source pictures commonly may be subdivided spatially into a plurality of sample blocks (for example, blocks of 4x4, 8x8, 4x8, or 16x16 samples each) and coded on a block-by-block basis. Blocks may be coded predictively with reference to other (already coded) blocks as determined by the coding assignment applied to the blocks' respective pictures. For example, blocks of I pictures may be coded non-predictively or they may be coded predictively with reference to already coded blocks of the same picture (spatial prediction or intra prediction). Pixel blocks of P pictures may be coded predictively, via spatial prediction or via temporal prediction with reference to one previously coded reference picture. Blocks of B pictures may be coded predictively, via spatial prediction or via temporal prediction with reference to one or two previously coded reference pictures.

**[0088]** The video encoder (503) may perform coding operations according to a predetermined video coding technology or standard, such as ITU-T Rec. H.265. In its operation, the video encoder (503) may perform various compression operations, including predictive coding operations that exploit temporal and spatial redundancies in the input video sequence. The coded video data, therefore, may conform to a syntax specified by the video coding technology or standard being used.

**[0089]** In an embodiment, the transmitter (540) may transmit additional data with the encoded video. The source coder (530) may include such data as part of the coded video sequence. Additional data may comprise temporal/spatial/SNR enhancement layers, other forms of redundant data such as redundant pictures and slices, SEI messages, VUI parameter set fragments, and so on.

**[0090]** A video may be captured as a plurality of source pictures (video pictures) in a temporal sequence. Intra-picture prediction (often abbreviated to intra prediction) makes use of spatial correlation in a given picture, and inter-picture prediction makes uses of the (temporal or

other) correlation between the pictures. In an example, a specific picture under encoding/decoding, which is referred to as a current picture, is partitioned into blocks. When a block in the current picture is similar to a reference block in a previously coded and still buffered reference picture in the video, the block in the current picture can be coded by a vector that is referred to as a motion vector. The motion vector points to the reference block in the reference picture, and can have a third dimension identifying the reference picture, in case multiple reference pictures are in use.

**[0091]** In some embodiments, a bi-prediction technique can be used in the inter-picture prediction. According to the bi-prediction technique, two reference pictures, such as a first reference picture and a second reference picture that are both prior in decoding order to the current picture in the video (but may be in the past and future, respectively, in display order) are used. A block in the current picture can be coded by a first motion vector that points to a first reference block in the first reference picture, and a second motion vector that points to a second reference block in the second reference picture. The block can be predicted by a combination of the first reference block and the second reference block.

**[0092]** Further, a merge mode technique can be used in the inter-picture prediction to improve coding efficiency.

**[0093]** According to some embodiments of the disclosure, predictions, such as inter-picture predictions and intra-picture predictions are performed in the unit of blocks. For example, according to the HEVC standard, a picture in a sequence of video pictures is partitioned into coding tree units (CTU) for compression, the CTUs in a picture have the same size, such as 64x64 pixels, 32x32 pixels, or 16x16 pixels. In general, a CTU includes three coding tree blocks (CTBs), which are one luma CTB and two chroma CTBs. Each CTU can be recursively quadtree split into one or multiple coding units (CUs). For example, a CTU of 64x64 pixels can be split into one CU of 64x64 pixels, or 4 CUs of 32x32 pixels, or 16 CUs of 16x16 pixels. In an example, each CU is analyzed to determine a prediction type for the CU, such as an inter prediction type or an intra prediction type. The CU is split into one or more prediction units (PUs) depending on the temporal and/or spatial predictability. Generally, each PU includes a luma prediction block (PB), and two chroma PBs. In an embodiment, a prediction operation in coding (encoding/decoding) is performed in the unit of a prediction block. Using a luma prediction block as an example of a prediction block, the prediction block includes a matrix of

values (e.g., luma values) for pixels, such as 8x8 pixels, 16x16 pixels, 8x16 pixels, 16x8 pixels, and the like.

**[0094]** FIG. 6 shows a diagram of a video encoder (603) according to another embodiment of the disclosure. The video encoder (603) is configured to receive a processing block (e.g., a prediction block) of sample values within a current video picture in a sequence of video pictures, and encode the processing block into a coded picture that is part of a coded video sequence. In an example, the video encoder (603) is used in the place of the video encoder (303) in the FIG. 3 example.

**[0095]** In an HEVC example, the video encoder (603) receives a matrix of sample values for a processing block, such as a prediction block of 8x8 samples, and the like. The video encoder (603) determines whether the processing block is best coded using intra mode, inter mode, or bi-prediction mode using, for example, rate-distortion optimization. When the processing block is to be coded in intra mode, the video encoder (603) may use an intra prediction technique to encode the processing block into the coded picture; and when the processing block is to be coded in inter mode or bi-prediction mode, the video encoder (603) may use an inter prediction or bi-prediction technique, respectively, to encode the processing block into the coded picture. In certain video coding technologies, merge mode can be an inter picture prediction submode where the motion vector is derived from one or more motion vector predictors without the benefit of a coded motion vector component outside the predictors. In certain other video coding technologies, a motion vector component applicable to the subject block may be present. In an example, the video encoder (603) includes other components, such as a mode decision module (not shown) to determine the mode of the processing blocks.

**[0096]** In the FIG. 6 example, the video encoder (603) includes the inter encoder (630), an intra encoder (622), a residue calculator (623), a switch (626), a residue encoder (624), a general controller (621), and an entropy encoder (625) coupled together as shown in FIG. 6.

**[0097]** The inter encoder (630) is configured to receive the samples of the current block (e.g., a processing block), compare the block to one or more reference blocks in reference pictures (e.g., blocks in previous pictures and later pictures), generate inter prediction information (e.g., description of redundant information according to inter encoding technique, motion vectors, merge mode information), and calculate inter prediction results (e.g., predicted block) based on the inter prediction information using any suitable technique. In some examples,

the reference pictures are decoded reference pictures that are decoded based on the encoded video information.

**[0098]** The intra encoder (622) is configured to receive the samples of the current block (e.g., a processing block), in some cases compare the block to blocks already coded in the same picture, generate quantized coefficients after transform, and in some cases also intra prediction information (e.g., an intra prediction direction information according to one or more intra encoding techniques). In an example, the intra encoder (622) also calculates intra prediction results (e.g., predicted block) based on the intra prediction information and reference blocks in the same picture.

**[0099]** The general controller (621) is configured to determine general control data and control other components of the video encoder (603) based on the general control data. In an example, the general controller (621) determines the mode of the block, and provides a control signal to the switch (626) based on the mode. For example, when the mode is the intra mode, the general controller (621) controls the switch (626) to select the intra mode result for use by the residue calculator (623), and controls the entropy encoder (625) to select the intra prediction information and include the intra prediction information in the bitstream; and when the mode is the inter mode, the general controller (621) controls the switch (626) to select the inter prediction result for use by the residue calculator (623), and controls the entropy encoder (625) to select the inter prediction information and include the inter prediction information in the bitstream.

**[0100]** The residue calculator (623) is configured to calculate a difference (residue data) between the received block and prediction results selected from the intra encoder (622) or the inter encoder (630). The residue encoder (624) is configured to operate based on the residue data to encode the residue data to generate the transform coefficients. In an example, the residue encoder (624) is configured to convert the residue data from a spatial domain to a frequency domain, and generate the transform coefficients. The transform coefficients are then subject to quantization processing to obtain quantized transform coefficients. In various embodiments, the video encoder (603) also includes a residue decoder (628). The residue decoder (628) is configured to perform inverse-transform, and generate the decoded residue data. The decoded residue data can be suitably used by the intra encoder (622) and the inter encoder (630). For example, the inter encoder (630) can generate decoded blocks based on the decoded residue data and inter prediction information, and the intra encoder (622) can generate decoded blocks based on the decoded residue data and the intra prediction information. The decoded blocks are

suitably processed to generate decoded pictures and the decoded pictures can be buffered in a memory circuit (not shown) and used as reference pictures in some examples.

**[0101]** The entropy encoder (625) is configured to format the bitstream to include the encoded block. The entropy encoder (625) is configured to include various information according to a suitable standard, such as the HEVC standard. In an example, the entropy encoder (625) is configured to include the general control data, the selected prediction information (e.g., intra prediction information or inter prediction information), the residue information, and other suitable information in the bitstream. Note that, according to the disclosed subject matter, when coding a block in the merge submode of either inter mode or bi-prediction mode, there is no residue information.

**[0102]** FIG. 7 shows a diagram of a video decoder (710) according to another embodiment of the disclosure. The video decoder (710) is configured to receive coded pictures that are part of a coded video sequence, and decode the coded pictures to generate reconstructed pictures. In an example, the video decoder (710) is used in the place of the video decoder (310) in the FIG. 3 example.

**[0103]** In the FIG. 7 example, the video decoder (710) includes an entropy decoder (771), an inter decoder (780), a residue decoder (773), a reconstruction module (774), and an intra decoder (772) coupled together as shown in FIG. 7.

**[0104]** The entropy decoder (771) can be configured to reconstruct, from the coded picture, certain symbols that represent the syntax elements of which the coded picture is made up. Such symbols can include, for example, the mode in which a block is coded (such as, for example, intra mode, inter mode, bi-predicted mode, the latter two in merge submode or another submode), prediction information (such as, for example, intra prediction information or inter prediction information) that can identify certain sample or metadata that is used for prediction by the intra decoder (772) or the inter decoder (780), respectively, residual information in the form of, for example, quantized transform coefficients, and the like. In an example, when the prediction mode is inter or bi-predicted mode, the inter prediction information is provided to the inter decoder (780); and when the prediction type is the intra prediction type, the intra prediction information is provided to the intra decoder (772). The residual information can be subject to inverse quantization and is provided to the residue decoder (773).

**[0105]** The inter decoder (780) is configured to receive the inter prediction information, and generate inter prediction results based on the inter prediction information.

**[0106]** The intra decoder (772) is configured to receive the intra prediction information, and generate prediction results based on the intra prediction information.

**[0107]** The residue decoder (773) is configured to perform inverse quantization to extract de-quantized transform coefficients, and process the de-quantized transform coefficients to convert the residual from the frequency domain to the spatial domain. The residue decoder (773) may also require certain control information (to include the Quantizer Parameter (QP)), and that information may be provided by the entropy decoder (771) (data path not depicted as this may be low volume control information only).

**[0108]** The reconstruction module (774) is configured to combine, in the spatial domain, the residual as output by the residue decoder (773) and the prediction results (as output by the inter or intra prediction modules as the case may be) to form a reconstructed block, that may be part of the reconstructed picture, which in turn may be part of the reconstructed video. It is noted that other suitable operations, such as a deblocking operation and the like, can be performed to improve the visual quality.

**[0109]** It is noted that the video encoders (303), (503), and (603), and the video decoders (310), (410), and (710) can be implemented using any suitable technique. In an embodiment, the video encoders (303), (503), and (603), and the video decoders (310), (410), and (710) can be implemented using one or more integrated circuits. In another embodiment, the video encoders (303), (503), and (503), and the video decoders (310), (410), and (710) can be implemented using one or more processors that execute software instructions.

**[0110]** Aspects of the disclosure are related to video coding technologies, such as signaling of a maximum number of triangle merge candidates for a triangle partition mode (TPM) or a Geometric Merge Mode (GEO) that is, for example, beyond HEVC and used in VVC.

**[0111]** A triangle partition for inter prediction can be described below. The TPM can be supported for inter prediction, for example, in VVC. In an example, the TPM is only applied to CUs that are 8x8 or larger. The TPM can be signaled using a CU-level flag as a kind of merge mode, with other merge modes including a regular merge mode, a merge mode with motion vector difference (MMVD) mode, a combined inter and intra prediction (CIIP) mode, a subblock merge mode, and/or the like.

**[0112]** FIGs. 8A-8B show examples of triangle partition based inter prediction according to an embodiment of the disclosure. When the TPM is used, a CU (800) can be split evenly into



two triangle-shaped partitions (also referred to as triangle partitions or partitions) (e.g., a partition 1 (811) and a partition 2 (812) in FIG. 8A and a partition 1 (821) and a partition 2 (822) in FIG. 8B), using either a diagonal split or an anti-diagonal split. The partitions (811)-(812) are divided by a line (810). The partitions (821)-(822) are divided by a line (820). Each triangle partition in the CU (800) is inter-predicted using motion information of the respective triangle partition. In an example, only uni-prediction is allowed for each triangle partition, and thus each triangle partition has one MV and one reference index. The uni-prediction motion constraint can be applied to ensure that only two motion compensated predictions are used for each CU, which is the same as a bi-prediction applied to the CU.

**[0113]** If the TPM is used for a current CU, a flag indicating a direction of the triangle partition (e.g., a diagonal split or an anti-diagonal split) and two merge indices (one for each partition) can be further signaled. A parameter indicating a maximum number of TPM merge candidates can be signaled explicitly at a picture parameter set (PPS) level, a picture header (PH) level, or the like. After predicting samples in each of the triangle partitions (e.g., the partitions (811)-(812) in FIG. 8A or the partitions (821)-(822) in FIG. 8B), values of samples along the diagonal or the anti-diagonal edge can be adjusted using a blending processing with adaptive weights. After deriving the prediction signal for the CU (800), a transform process and a quantization process can be further applied to the CU (800) as in other prediction modes.

**[0114]** The geometric merge mode (also referred to as a geometric partitioning mode) can support a plurality of different partitioning manners. FIG. 9 shows an exemplary geometric merge mode. In the geometric merge mode, the CU (900) can be partitioned into two partitions, partitions 1-2 divided by a line or an edge (910). Each of the two partitions can have any suitable shape, such as a triangle, a trapezoid, a pentagon, or the like.

**[0115]** When the line (910) is the line (810) or the line (820), the geometric merge mode is the TPM. In an example, the TPM is an example of the geometric merge mode and the geometric merge mode includes the TPM. The descriptions (embodiments, examples, and the like) for the TPM in the disclosure can be suitably adapted to the geometric merge mode, for example, by replacing the TPM with the geometric merge mode. The descriptions (embodiments, examples, and the like) for the geometric merge mode in the disclosure can be suitably adapted to the TPM, for example, by replacing the geometric merge mode with the TPM.

[0116] The TPM can be controlled at a high level. The high level can refer to a picture level, a PPS level, a sequence level, or a video level that is associated with a picture header (PH), a PPS, a sequence parameter set (SPS), or a video parameter set (VPS). In an example, the high level does not refer to a sub-picture level (e.g., a slice level).

[0117] The TPM can be controlled (e.g., enabled or disabled) in the sequence level using an SPS syntax element. FIG. 10 shows an exemplary sequence level control of the TPM. A sequence-level triangle flag (e.g., a `sps_triangle_enabled_flag`) can specify whether a triangular shape based motion compensation can be used for inter prediction. The sequence-level triangle flag (e.g., the `sps_triangle_enabled_flag`) that is equal to 0 can specify that the syntax is constrained such that no triangular shape based motion compensation is used in a coded layer video sequence (CLVS), and parameters or syntax elements (e.g., `merge_triangle_split_dir`, `merge_triangle_idx0`, and `merge_triangle_idx1`) related to the TPM are not present in coding unit syntax of the CLVS. The sequence-level triangle flag (e.g., the `sps_triangle_enabled_flag`) that is equal to 1 can specify that the triangular shape based motion compensation can be used in the CLVS.

[0118] A parameter indicating the maximum number of TPM merge candidates can be explicitly signaled at a high level (e.g., the PPS level in the PPS, the picture level in the PH, or another high level). FIG. 11 shows an exemplary PPS raw byte sequence payload (RBSP) in which a maximum number of TPM merge candidates can be signaled in a PPS. In an example, a PPS-level parameter or a PPS level parameter (e.g., `pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1`) is explicitly signaled in the PPS to indicate the maximum number of TPM merge candidates. FIG. 12 shows an exemplary picture header RBSP in which a maximum number of TPM merge candidates can be signaled in a picture header. In an example, a picture header level parameter or a picture level parameter (e.g., `pic_max_num_merge_cand_minus_max_num_triangle_cand`) is explicitly signaled in the picture header to indicate the maximum number of TPM merge candidates.

[0119] A PPS-level parameter (e.g., `pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1`) that is equal to 0 can specify that the picture level parameter (e.g., `pic_max_num_merge_cand_minus_max_num_triangle_cand`) is present (e.g., signaled) in PHs of slices referring to the PPS. A PPS-level parameter (e.g., `pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1`) that is greater than 0 can

specify that the picture level parameter (e.g., `pic_max_num_merge_cand_minus_max_num_triangle_cand`) is not present (e.g., not signaled) in the PHs referring to the PPS. A value of the PPS-level parameter (e.g., `pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1`) can be in a range of 0 to a `MaxNumMergeCand - 1`, where `MaxNumMergeCand` is a maximum number of merge candidates in a merge candidate list for a regular merge mode. When the PPS-level parameter (e.g., `pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1`) is not present, the PPS-level parameter can be inferred to be 0.

**[0120]** The picture level parameter (e.g., `pic_max_num_merge_cand_minus_max_num_triangle_cand`) can specify a maximum number of triangular merge mode candidates (`MaxNumTriangleMergeCand`) supported in slices associated with the picture header subtracted from the maximum number of merge candidates (`MaxNumMergeCand`).

**[0121]** In an example, when the picture level parameter (e.g., `pic_max_num_merge_cand_minus_max_num_triangle_cand`) is not present (e.g., not signaled), the sequence-level triangle flag (e.g., the `sps_triangle_enabled_flag`) is equal to 1, and `MaxNumMergeCand` is greater than or equal to 2, the picture level parameter (e.g., `pic_max_num_merge_cand_minus_max_num_triangle_cand`) is inferred to be equal to (`pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1 - 1`).

**[0122]** The maximum number of triangular merge mode candidates, (`MaxNumTriangleMergeCand`) can be determined based on `MaxNumMergeCand` and the picture level parameter (e.g., `pic_max_num_merge_cand_minus_max_num_triangle_cand`). In an example, `MaxNumTriangleMergeCand` is determined using Eq. 1.

$$\text{MaxNumTriangleMergeCand} = \text{MaxNumMergeCand} - \text{pic\_max\_num\_merge\_cand\_minus\_max\_num\_triangle\_cand} \quad (\text{Eq. 1})$$

**[0123]** In an example, when the picture level parameter (e.g., `pic_max_num_merge_cand_minus_max_num_triangle_cand`) is present, the value of `MaxNumTriangleMergeCand` is in a range of 2 to `MaxNumMergeCand`, inclusive. For example, if `MaxNumMergeCand` is 4, the range is [2, 3, 4], and `MaxNumTriangleMergeCand` is one of 2, 3, and 4.

**[0124]** When the picture level parameter (e.g., `pic_max_num_merge_cand_minus_max_num_triangle_cand`) is not present and one of two

conditions is true, MaxNumTriangleMergeCand is set to be equal to 0. The two conditions include (i) the sequence-level triangle flag (e.g., a `sps_triangle_enabled_flag`) is equal to 0 and (ii) MaxNumMergeCand is less than 2.

[0125] In an example, when MaxNumTriangleMergeCand is equal to 0, the TPM is not allowed for the slices associated with the PH.

[0126] The PPS-level parameter (e.g., `pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1`), the picture level parameter (e.g., `pic_max_num_merge_cand_minus_max_num_triangle_cand`), and MaxNumMergeCand can be used to determine MaxNumTriangleMergeCand. A value of MaxNumTriangleMergeCand can be designated not to exceed a value of MaxNumMergeCand. The value of MaxNumTriangleMergeCand can vary, for example, depending on specific applications. PPS signaling can be effective when the value of MaxNumTriangleMergeCand does not vary frequently, and thus does not need to be signaled for each picture. On the other hand, picture header signaling can be effective when MaxNumTriangleMergeCand changes more frequently, for example, from one picture to another picture.

[0127] The PPS-level parameter (e.g., `pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1`), the picture level parameter (e.g., `pic_max_num_merge_cand_minus_max_num_triangle_cand`), a value of MaxNumMergeCand, and the sequence-level triangle flag (e.g., a `sps_triangle_enabled_flag`) can be used to determine a value of MaxNumTriangleMergeCand. The value of the PPS-level parameter (e.g., `pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1`) can be in a range of 0 to (the value of MaxNumMergeCand – 1).

[0128] In an example, the PPS-level parameter is parsed or inferred to be 0, the sequence-level triangle flag is 0 indicating that the TPM is disabled, the picture level parameter is not present, and a value of MaxNumTriangleMergeCand is 0 regardless of the value of MaxNumMergeCand.

[0129] In an example, the PPS-level parameter is parsed or inferred to be 0, the sequence-level triangle flag is 1 indicating that the TPM is enabled, the picture level parameter is not present, and a value of MaxNumTriangleMergeCand is 0 when the value of MaxNumMergeCand is less than 2.

[0130] In an example, the PPS-level parameter is parsed or inferred to be 0, the sequence-level triangle flag is 1 indicating that the TPM is enabled, the picture level parameter is

present and parsed, and a value of `MaxNumTriangleMergeCand` is  $(\text{MaxNumMergeCand} - \text{a value of the picture level parameter})$  (e.g.,  $\text{MaxNumMergeCand} - \text{pic\_max\_num\_merge\_cand\_minus\_max\_num\_triangle\_cand}$ ) when the value of `MaxNumMergeCand` is larger than or equal to 2. The value of `MaxNumTriangleMergeCand` can be in a range of 2 to `MaxNumMergeCand`, inclusive.

**[0131]** In an example, the PPS-level parameter is parsed and is not 0, the sequence-level triangle flag is 0 indicating that the TPM is disabled, the picture level parameter is not present, and a value of `MaxNumTriangleMergeCand` is 0 regardless of the value of `MaxNumMergeCand`.

**[0132]** In an example, the PPS-level parameter is parsed and is not 0, the sequence-level triangle flag is 1 indicating that the TPM is enabled, the picture level parameter is not present, and a value of `MaxNumTriangleMergeCand` is 0 when the value of `MaxNumMergeCand` is less than 2.

**[0133]** In an example, the PPS-level parameter is parsed and is not 0, the sequence-level triangle flag is 1 indicating that the TPM is enabled, the picture level parameter is not present and is inferred to be  $(\text{the value of the PPS-level parameter} - 1)$  (e.g.,  $\text{pps\_max\_num\_merge\_cand\_minus\_max\_num\_triangle\_cand\_plus1} - 1$ ), and a value of `MaxNumTriangleMergeCand` is  $(\text{MaxNumMergeCand} - (\text{the value of the PPS-level parameter} - 1))$  (e.g.,  $\text{MaxNumMergeCand} - (\text{pps\_max\_num\_merge\_cand\_minus\_max\_num\_triangle\_cand\_plus1} - 1)$ ) when the value of `MaxNumMergeCand` is larger than or equal to 2.

**[0134]** Design of signaling the maximum number of triangle merge candidates can be challenging. When the value of `MaxNumMergeCand` is 2, the sequence-level triangle flag (e.g., a `sps_triangle_enabled_flag`) is 1, and the PPS-level parameter (e.g.,  $\text{pps\_max\_num\_merge\_cand\_minus\_max\_num\_triangle\_cand\_plus1}$ ) is 0, the picture level parameter (e.g.,  $\text{pic\_max\_num\_merge\_cand\_minus\_max\_num\_triangle\_cand}$ ) can be signaled in the picture header (see FIG. 12) and can be subsequently parsed. However, as described above, when  $\text{pic\_max\_num\_merge\_cand\_minus\_max\_num\_triangle\_cand}$  is present, the value of `MaxNumTriangleMergeCand` is in the range of 2 to `MaxNumMergeCand`, inclusive. Thus, `MaxNumTriangleMergeCand` can be inferred to be 2 since the value of `MaxNumMergeCand` is also 2. Accordingly, parsing  $\text{pic\_max\_num\_merge\_cand\_minus\_max\_num\_triangle\_cand}$  can be redundant.

**[0135]** According to aspects of the disclosure, the picture level parameter (e.g., `pic_max_num_merge_cand_minus_max_num_triangle_cand`) is signaled in the picture header and is decoded only when the sequence-level triangle flag (e.g., the `sps_triangle_enabled_flag`) is 1, `MaxNumMergeCand` is larger than 2, and the PPS-level parameter (e.g., `pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1`) is 0, as shown in FIG. 13. Alternatively, the picture level parameter (e.g., `pic_max_num_merge_cand_minus_max_num_triangle_cand`) is signaled in the picture header and is decoded only when the sequence-level triangle flag (e.g., the `sps_triangle_enabled_flag`) is 1, `MaxNumMergeCand` is larger than or equal to 3, and the PPS-level parameter (e.g., `pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1`) is 0, as shown in FIG. 14.

**[0136]** FIG. 13 shows an exemplary picture header syntax where the maximum number of TPM merge candidates is signaled in the picture header using the picture header level parameter (e.g., `pic_max_num_merge_cand_minus_max_num_triangle_cand`). A box (1310) in FIG. 13 indicates a difference between the syntax (e.g., `MaxNumMergeCand` is larger than or equal to 2) shown in FIG. 12 and the syntax (e.g., `MaxNumMergeCand` is larger than 2) shown in FIG. 13.

**[0137]** The picture header level parameter (e.g., `pic_max_num_merge_cand_minus_max_num_triangle_cand`) can specify a maximum number of triangular merge mode candidates (`MaxNumTriangleMergeCand`) supported in slices associated with the picture header subtracted from the maximum number of merge candidates (`MaxNumMergeCand`).

**[0138]** In an example, when the picture level parameter (e.g., `pic_max_num_merge_cand_minus_max_num_triangle_cand`) is not present (e.g., not signaled), the sequence-level triangle flag (e.g., the `sps_triangle_enabled_flag`) is equal to 1, and `MaxNumMergeCand` is greater than or equal to 2, the picture level parameter (e.g., `pic_max_num_merge_cand_minus_max_num_triangle_cand`) is inferred to be equal to  $(pps\_max\_num\_merge\_cand\_minus\_max\_num\_triangle\_cand\_plus1 - 1)$ .

**[0139]** The maximum number of triangular merge mode candidates, (`MaxNumTriangleMergeCand`) can be determined based on `MaxNumMergeCand` and the picture level parameter (e.g., `pic_max_num_merge_cand_minus_max_num_triangle_cand`). In an example, `MaxNumTriangleMergeCand` is determined using Eq. 1.

[0140] In an example, when the picture level parameter (e.g., `pic_max_num_merge_cand_minus_max_num_triangle_cand`) is present, the value of `MaxNumTriangleMergeCand` is in a range of 2 to `MaxNumMergeCand`, inclusive. According to aspects of the disclosure, when the picture level parameter (e.g., `pic_max_num_merge_cand_minus_max_num_triangle_cand`) is not present and the `MaxNumMergeCand` is 2, the `MaxNumTriangleMergeCand` is 2.

[0141] When the picture level parameter (e.g., `pic_max_num_merge_cand_minus_max_num_triangle_cand`) is not present and one of the two conditions is true, `MaxNumTriangleMergeCand` is set to be equal to 0. As described above, the two conditions include (i) the sequence-level triangle flag (e.g., a `sps_triangle_enabled_flag`) is equal to 0 and (ii) `MaxNumMergeCand` is less than 2.

[0142] In an example, when `MaxNumTriangleMergeCand` is equal to 0, the TPM is not allowed for the slices associated with the picture header.

[0143] FIG. 14 shows an exemplary picture header syntax where the maximum number of TPM merge candidates is signaled in the picture header, for example using the picture header level parameter (e.g., `pic_max_num_merge_cand_minus_max_num_triangle_cand`). A box (1410) in FIG. 14 indicates a difference between the syntax (e.g., `MaxNumMergeCand` is larger than or equal to 2) shown in FIG. 12 and the syntax (e.g., `MaxNumMergeCand` is larger than or equal to 3) shown in FIG. 14.

[0144] In general, coding information for a current picture can be decoded from a coded video bitstream. The coding information can indicate that the geometric merge mode is enabled at the sequence level and the PPS level parameter in the PPS indicating the maximum number of geometric merge mode merge candidates is 0. In an example, the geometric merge mode is the TPM. Further, the coding information can indicate the maximum number of merge candidates. When the maximum number of merge candidates satisfies a condition, the picture level parameter signaled for the current picture in the coded video bitstream can be decoded. The picture level parameter can indicate the maximum number of geometric merge mode merge candidates. In an example, the maximum number of geometric merge mode merge candidates is the maximum number of TPM merge candidates.

[0145] In an example, when the maximum number of merge candidates does not satisfy the condition, the picture level parameter that is not signaled in the coded video bitstream is not decoded. In an example, the picture level parameter is not signaled in the coded video bitstream

and the maximum number of merge candidates is 2 and does not satisfy the condition, thus the maximum number of TPM merge candidates is determined to be 2.

**[0146]** In an example, the condition is one of (i) the maximum number of merge candidates is larger than 2 and (ii) the maximum number of merge candidates is larger than or equal to 3.

**[0147]** In some examples, when the sequence-level triangle flag (e.g., the `sps_triangle_enabled_flag`) is 1 and `MaxNumMergeCand` is larger than or equal to 2, the TPM cannot be disabled for certain pictures. Therefore, the picture level lacks flexibility, for example, in the TPM control (e.g., disabling the TPM for certain pictures).

**[0148]** In an embodiment, when the TPM is disabled in the sequence level, the PPS-level parameter (e.g., `pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1`) should not be decoded in order to reduce a signaling overhead. In some examples (e.g., FIG. 11), regardless of a sequence level control of the TPM, the PPS-level parameter (e.g., `pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1`) is decoded when a flag (e.g., a `constant_slice_header_params_enabled_flag`) equals to 1, and thus signaling efficiency, for example, at the sequence level can be relatively low.

**[0149]** According to aspects of the disclosure, coding information for a current picture can be decoded from a coded video bitstream. The coding information can indicate that the geometric merge mode is enabled for a coding level higher than a picture level of the current picture and a maximum number of merge candidates (e.g., for a regular merge mode) satisfies a condition. The coding level that is higher than the picture level can be a sequence level. The condition can include that the maximum number of merge candidates is larger than or equal to 2.

**[0150]** When the picture level parameter indicating the maximum number of geometric merge mode merge candidates (e.g., `MaxNumTriangleMergeCand`) is signaled for the current picture in the coded video bitstream, the maximum number of geometric merge mode merge candidates for the current picture can be determined based on the picture level parameter and the maximum number of merge candidates (`MaxNumMergeCand`). According to aspects of the disclosure, the maximum number of geometric merge mode merge candidates can be in a range including 0 and a subrange of 2 to the maximum number of merge candidates, inclusive. In an example, if `MaxNumMergeCand` is 4, the subrange is 2 to 4. The range includes 0 and 2 to 4 or [0, 2, 3, 4]. Alternatively, the maximum number of geometric merge mode merge candidates can be (i) 0 or (ii) one of 2 to the maximum number of merge candidates. In an example, if



MaxNumMergeCand is 4, the maximum number of geometric merge mode merge candidates can be 0, 2, 3, or 4.

**[0151]** In an example, the geometric merge mode is the TPM, and the maximum number of geometric merge mode merge candidates is the maximum number of TPM merge candidates. The picture level parameter can be the `pic_max_num_merge_cand_minus_max_num_triangle_cand`.

**[0152]** When the maximum number of geometric merge mode merge candidates (e.g., the maximum number of TPM merge candidates) is 0, the geometric merge mode (e.g., the TPM) is disabled for the current picture. When the maximum number of geometric merge mode merge candidates (e.g., the maximum number of TPM merge candidates) is not 0 (e.g., larger than 0), the geometric merge mode (e.g., the TPM) is enabled for the current picture.

**[0153]** The maximum number of TPM merge candidates can be determined by subtracting the picture level parameter from the maximum number of merge candidates, for example, using Eq. (1).

**[0154]** The above description can support functionality to disable the TPM for certain pictures when the TPM is enabled in the sequence level and MaxNumMergeCand is larger than or equal to 2. Thus, picture level flexibility is supported.

**[0155]** In an embodiment, when `pic_max_num_merge_cand_minus_max_num_triangle_cand` is present (e.g., signaled), the value of MaxNumTriangleMergeCand can be in the range of 0, 2 to MaxNumMergeCand, inclusive.

**[0156]** Therefore, when `pic_max_num_merge_cand_minus_max_num_triangle_cand` is present, the value of `pic_max_num_merge_cand_minus_max_num_triangle_cand` can be equal to the value of MaxNumMergeCand. Thus, the value of MaxNumTriangleMergeCand can be 0 which is a picture level control to disable the TPM at the picture level.

**[0157]** The PPS-level parameter or the PPS level parameter (e.g., `pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1`) that indicates the maximum number of TPM merge candidates can be signaled in the coded video bitstream for the PPS associated with the current picture. The PPS level parameter can be (i) one of 0 to (the maximum number of merge candidates - 1) or (ii) (the maximum number of merge candidates + 1). For example, when MaxNumMergeCand is 4, the PPS level parameter can be 0, 1, 2, 3, or 5.

**[0158]** The PPS level parameter can be in a range including a subrange of 0 to (the maximum number of merge candidates - 1) and (the maximum number of merge candidates + 1),

inclusive. In an example, referring to syntax elements in FIG. 11, the PPS level parameter (e.g., `pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1`) is to be in the range of 0 to  $(\text{MaxNumMergeCand} - 1)$ , and  $(\text{MaxNumMergeCand} + 1)$ , inclusive. For example, when `MaxNumMergeCand` is 4, the subrange for the PPS level parameter is 0 to 3, and the range for the PPS level parameter is 0 to 3, and 5. Alternatively, the range for the PPS level parameter includes [0, 1, 2, 3, 5].

**[0159]** In an embodiment, the PPS level parameter (e.g., `pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1`) that indicates the maximum number of TPM merge candidates (e.g., for the PPS associated with the current picture) is not signaled in the coded video bitstream. For example, referring to FIG. 15, a box (1510) indicates that `pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1` is removed from the PPS RBSP in FIG. 11 to have a simple control of the TPM when the TPM is to be disabled in the picture level.

**[0160]** Referring to FIG. 16A, the coded video bitstream can include the picture header for the current picture. The picture level parameter can be signaled in the picture header when the TPM is enabled for the sequence level (e.g., the sequence-level triangle flag, such as the `sps_triangle_enabled_flag`, is 1) and the maximum number of merge candidates is larger than or equal to 2. The signaling of the picture level parameter can be independent from the PPS level parameter.

**[0161]** A box (1610) indicates that the signaling of the picture level parameter can be independent from the PPS level parameter (e.g., `pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1`). Further, as indicated by a box (1620) in FIG. 16B, the following description: “when the picture level parameter (e.g., `pic_max_num_merge_cand_minus_max_num_triangle_cand`) is not present (e.g., not signaled), the sequence-level triangle flag (e.g., the `sps_triangle_enabled_flag`) is equal to 1, and `MaxNumMergeCand` is greater than or equal to 2, the picture level parameter (e.g., `pic_max_num_merge_cand_minus_max_num_triangle_cand`) is inferred to be equal to  $(\text{pps\_max\_num\_merge\_cand\_minus\_max\_num\_triangle\_cand\_plus1} - 1)$ ” does not apply when the signaling of the picture level parameter is independent from the PPS level parameter (e.g., `pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1`).

**[0162]** The picture level parameter (e.g., `pic_max_num_merge_cand_minus_max_num_triangle_cand`) can specify the maximum number

of triangular merge mode candidates (MaxNumTriangleMergeCand) supported in the slices associated with the picture header subtracted from the maximum number of merge candidates (MaxNumMergeCand).

**[0163]** The maximum number of triangular merge mode candidates, (MaxNumTriangleMergeCand) can be determined based on MaxNumMergeCand and the picture level parameter (e.g., `pic_max_num_merge_cand_minus_max_num_triangle_cand`), for example, using Eq. 1.

**[0164]** In an example, when the picture level parameter (e.g., `pic_max_num_merge_cand_minus_max_num_triangle_cand`) is present, the value of MaxNumTriangleMergeCand is in the range of 2 to MaxNumMergeCand, inclusive.

**[0165]** When the picture level parameter (e.g., `pic_max_num_merge_cand_minus_max_num_triangle_cand`) is not present and one of the two conditions is true, MaxNumTriangleMergeCand is set to be equal to 0. The two conditions include (i) the sequence-level triangle flag (e.g., `sps_triangle_enabled_flag`) is equal to 0 and (ii) MaxNumMergeCand is less than 2.

**[0166]** In an example, when MaxNumTriangleMergeCand is equal to 0, the TPM is not allowed for the slices associated with the PH.

**[0167]** The coded video bitstream can include the PPS associated with the current picture. The PPS level parameter (e.g., `pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1`) indicating the maximum number of TPM merge candidates can be signaled in the PPS (e.g., PPS RBSP in FIG. 17) based at least on a PPS level flag (e.g., `pps_max_num_triangle_merge_cand_present_flag`) indicating that the PPS level parameter is to be signaled. Referring to FIG. 17, a box (1610) indicates that the PPS level flag (e.g., `pps_max_num_triangle_merge_cand_present_flag`) is signaled in the PPS RBSP. As indicated by a box (1620), when the PPS level flag (e.g., `pps_max_num_triangle_merge_cand_present_flag`) is true (e.g., having a value of 1), the PPS level parameter (e.g., `pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1`) can be signaled in the PPS RBSP. Referring to FIG. 17, in an example, the PPS level parameter is signaled when the PPS level flag (e.g., `pps_max_num_triangle_merge_cand_present_flag`) is true and a flag (e.g., `constant_slice_header_params_enabled_flag`) is true.

**[0168]** The PPS level flag (e.g., `pps_max_num_triangle_merge_cand_present_flag`) can specify the presence of a syntax element (e.g.,

pps\_max\_num\_merge\_cand\_minus\_max\_num\_triangle\_cand) in the bitstream (e.g., the coded video bitstream). When pps\_max\_num\_triangle\_merge\_cand\_present\_flag is equal to 1, pps\_max\_num\_merge\_cand\_minus\_max\_num\_triangle\_cand can be present. When pps\_max\_num\_triangle\_merge\_cand\_present\_flag is equal to 0, pps\_max\_num\_merge\_cand\_minus\_max\_num\_triangle\_cand is not present. When pps\_max\_num\_triangle\_merge\_cand\_present\_flag is not present, pps\_max\_num\_triangle\_merge\_cand\_present\_flag can be inferred to be 0. When the value of sps\_triangle\_enabled\_flag is 0, it is a requirement of bitstream conformance that the value of pps\_max\_num\_triangle\_merge\_cand\_present\_flag is equal to 0.

[0169] pps\_max\_num\_merge\_cand\_minus\_max\_num\_triangle\_cand can specify the maximum number of TPM merge candidates supported in the pictures referred to in the PPS. For example, the maximum number of TPM merge candidates is obtained using Eq. 1.

[0170] When pps\_max\_num\_merge\_cand\_minus\_max\_num\_triangle\_cand is not present, pps\_max\_num\_merge\_cand\_minus\_max\_num\_triangle\_cand can be inferred to be equal to 0.

[0171] Referring to FIG. 18, the coded video bitstream can include the picture header for the current picture. The picture level parameter (e.g., pic\_max\_num\_merge\_cand\_minus\_max\_num\_triangle\_cand) can be signaled in the picture header (e.g., the picture header RBSP in FIG. 18) when the TPM is enabled for the sequence level (the sequence-level triangle flag, e.g., the sps\_triangle\_enabled\_flag is equal to 1), the maximum number of merge candidates is larger than or equal to 2 ( $\text{MaxNumMergeCand} \geq 2$ ), and the PPS level flag (e.g., pps\_max\_num\_triangle\_merge\_cand\_present\_flag) indicates that the PPS level parameter is not to be signaled (e.g., pps\_max\_num\_triangle\_merge\_cand\_present\_flag is 0) as shown by a box (1810).

[0172] The picture level parameter (e.g., pic\_max\_num\_merge\_cand\_minus\_max\_num\_triangle\_cand) can specify the maximum number of TPM merge candidates supported in the slices associated with the PH. In an example, the maximum number of TPM merge candidates is obtained using Eq. 1.

[0173] When the picture level parameter (e.g., pic\_max\_num\_merge\_cand\_minus\_max\_num\_triangle\_cand) is not present, the picture level parameter can be inferred to be equal to the value of the PPS parameter (e.g., pps\_max\_num\_merge\_cand\_minus\_max\_num\_triangle\_cand).

[0174] The maximum number of TPM merge candidates (MaxNumTriangleMergeCand) can be derived as follows: (i) if `sps_triangle_enabled_flag` is equal to 1 and `MaxNumMergeCand` is larger than or equal to 2, `MaxNumTriangleMergeCand` can be determined using Eq. 1; (ii) otherwise, `MaxNumTriangleMergeCand` can be set to 0.

[0175] When `MaxNumTriangleMergeCand` is equal to 0, the TPM is not allowed for the slices associated with the PH.

[0176] Referring to FIG. 19, the coded video bitstream can include the PPS associated with the current picture. The PPS level parameter (e.g., `pps_max_num_merge_cand_minus_max_num_triangle_cand`) indicating the maximum number of TPM merge candidates can be signaled in the PPS (e.g., PPS RBSP in FIG. 19) based at least on the TPM being enabled for the sequence level.

[0177] A box (1910) indicates that a sequence level TPM enabling/disabling flag can be used to determine whether to decode a syntax element (e.g., `pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1`). The sequence level TPM enabling/disabling flag can be the `sps_triangle_enabled_flag`.

[0178] FIG. 20 shows a flow chart outlining a process (2000) according to an embodiment of the disclosure. The process (2000) can be used in signaling parameter(s) indicating the maximum number of geometric merge mode merge candidates for the geometric merge mode. In various embodiments, the process (2000) are executed by processing circuitry, such as the processing circuitry in the terminal devices (210), (220), (230) and (240), the processing circuitry that performs functions of the video encoder (303), the processing circuitry that performs functions of the video decoder (310), the processing circuitry that performs functions of the video decoder (410), the processing circuitry that performs functions of the video encoder (503), and the like. In some embodiments, the process (2000) is implemented in software instructions, thus when the processing circuitry executes the software instructions, the processing circuitry performs the process (2000). The process starts at (S2001) and proceeds to (S2010).

[0179] At (S2010), coding information for a current picture can be decoded from a coded video bitstream. The coding information can indicate that a geometric merge mode is enabled for a coding level higher than a picture level of the current picture and a maximum number of merge candidates satisfies a condition.

**[0180]** At (S2020), when a picture level parameter is signaled for the current picture in the coded video bitstream, a maximum number of geometric merge mode merge candidates can be determined based on the picture level parameter and the maximum number of merge candidates. The maximum number of geometric merge mode merge candidates can be (i) 0 or (ii) one of 2 to the maximum number of merge candidates. In an example, the geometric merge mode is the TPM and the maximum number of geometric merge mode merge candidates is the maximum number of TPM merge candidates. The picture level parameter can indicate the maximum number of geometric merge mode merge candidates. The geometric merge mode can be disabled for the current picture based on the maximum number of geometric merge mode merge candidates being 0. The geometric merge mode can be enabled for the current picture based on the maximum number of geometric merge mode merge candidates not being 0.

**[0181]** In an example, at (S2030), whether the maximum number of geometric merge mode merge candidates is 0 can be determined. If the maximum number of geometric merge mode merge candidates is determined to be 0, the process (2000) proceeds to (S2040). Otherwise, the process (2000) proceeds to (S2050).

**[0182]** At (S2040), the geometric merge mode can be disabled for the current picture. The process (2000) proceeds to (S2099) and terminates.

**[0183]** At (S2050), the geometric merge mode can be enabled for the current picture. The process (2000) proceeds to (S2099) and terminates.

**[0184]** The process (2000) can be suitably adapted. Step(s) in the process (2000) can be modified and/or omitted. Additional step(s) can be added. Any suitable order of implementation can be used.

**[0185]** FIG. 21 shows a flow chart outlining a process (2100) according to an embodiment of the disclosure. The process (2100) can be used in signaling parameter(s) indicating the maximum number of geometric merge mode merge candidates for the geometric merge mode. In various embodiments, the process (2100) are executed by processing circuitry, such as the processing circuitry in the terminal devices (210), (220), (230) and (240), the processing circuitry that performs functions of the video encoder (303), the processing circuitry that performs functions of the video decoder (310), the processing circuitry that performs functions of the video decoder (410), the processing circuitry that performs functions of the video encoder (503), and the like. In some embodiments, the process (2100) is implemented in software instructions, thus when the processing circuitry executes the software instructions, the

processing circuitry performs the process (2100). The process starts at (S2101) and proceeds to (S2110).

**[0186]** At (S2110), coding information for a current picture can be decoded from a coded video bitstream. The coding information can indicate a geometric merge mode being enabled at a sequence level, a picture parameter set (PPS) level parameter in a PPS being 0, and a maximum number of merge candidates. The PPS level parameter can indicate a maximum number of geometric merge mode merge candidates. In an example, the geometric merge mode is the TPM and the maximum number of geometric merge mode merge candidates is the maximum number of TPM merge candidates.

**[0187]** At (S2120), a picture level parameter signaled for the current picture in the coded video bitstream can be decoded when the maximum number of merge candidates satisfies a condition. The picture level parameter can indicate the maximum number of geometric merge mode merge candidates. The geometric merge mode can be a triangle partition mode (TPM) and the maximum number of geometric merge mode merge candidates is a maximum number of TPM merge candidates. The condition can be one of (i) the maximum number of merge candidates is larger than 2 and (ii) the maximum number of merge candidates is larger than or equal to 3. The process (2100) proceeds to (S2199), and terminates.

**[0188]** The process (2100) can be suitably adapted. Step(s) in the process (2100) can be modified and/or omitted. Additional step(s) can be added. Any suitable order of implementation can be used. For example, when the maximum number of merge candidates does not satisfy the condition, the picture level parameter is not decoded.

**[0189]** Embodiments in the disclosure may be used separately or combined in any order. Further, each of the methods (or embodiments), an encoder, and a decoder may be implemented by processing circuitry (e.g., one or more processors or one or more integrated circuits). In one example, the one or more processors execute a program that is stored in a non-transitory computer-readable medium.

**[0190]** The techniques described above, can be implemented as computer software using computer-readable instructions and physically stored in one or more computer-readable media. For example, FIG. 22 shows a computer system (2200) suitable for implementing certain embodiments of the disclosed subject matter.

**[0191]** The computer software can be coded using any suitable machine code or computer language, that may be subject to assembly, compilation, linking, or like mechanisms to

create code comprising instructions that can be executed directly, or through interpretation, micro-code execution, and the like, by one or more computer central processing units (CPUs), Graphics Processing Units (GPUs), and the like.

**[0192]** The instructions can be executed on various types of computers or components thereof, including, for example, personal computers, tablet computers, servers, smartphones, gaming devices, internet of things devices, and the like.

**[0193]** The components shown in FIG. 22 for computer system (2200) are exemplary in nature and are not intended to suggest any limitation as to the scope of use or functionality of the computer software implementing embodiments of the present disclosure. Neither should the configuration of components be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in the exemplary embodiment of a computer system (2200).

**[0194]** Computer system (2200) may include certain human interface input devices. Such a human interface input device may be responsive to input by one or more human users through, for example, tactile input (such as: keystrokes, swipes, data glove movements), audio input (such as: voice, clapping), visual input (such as: gestures), olfactory input (not depicted). The human interface devices can also be used to capture certain media not necessarily directly related to conscious input by a human, such as audio (such as: speech, music, ambient sound), images (such as: scanned images, photographic images obtain from a still image camera), video (such as two-dimensional video, three-dimensional video including stereoscopic video).

**[0195]** Input human interface devices may include one or more of (only one of each depicted): keyboard (2201), mouse (2202), trackpad (2203), touch screen (2210), data-glove (not shown), joystick (2205), microphone (2206), scanner (2207), camera (2208).

**[0196]** Computer system (2200) may also include certain human interface output devices. Such human interface output devices may be stimulating the senses of one or more human users through, for example, tactile output, sound, light, and smell/taste. Such human interface output devices may include tactile output devices (for example tactile feedback by the touch-screen (2210), data-glove (not shown), or joystick (2205), but there can also be tactile feedback devices that do not serve as input devices), audio output devices (such as: speakers (2209), headphones (not depicted)), visual output devices (such as screens (2210) to include CRT screens, LCD screens, plasma screens, OLED screens, each with or without touch-screen input capability, each with or without tactile feedback capability—some of which may be capable to output two



dimensional visual output or more than three dimensional output through means such as stereographic output; virtual-reality glasses (not depicted), holographic displays and smoke tanks (not depicted)), and printers (not depicted).

**[0197]** Computer system (2200) can also include human accessible storage devices and their associated media such as optical media including CD/DVD ROM/RW (2220) with CD/DVD or the like media (2221), thumb-drive (2222), removable hard drive or solid state drive (2223), legacy magnetic media such as tape and floppy disc (not depicted), specialized ROM/ASIC/PLD based devices such as security dongles (not depicted), and the like.

**[0198]** Those skilled in the art should also understand that term “computer readable media” as used in connection with the presently disclosed subject matter does not encompass transmission media, carrier waves, or other transitory signals.

**[0199]** Computer system (2200) can also include an interface to one or more communication networks. Networks can for example be wireless, wireline, optical. Networks can further be local, wide-area, metropolitan, vehicular and industrial, real-time, delay-tolerant, and so on. Examples of networks include local area networks such as Ethernet, wireless LANs, cellular networks to include GSM, 3G, 4G, 5G, LTE and the like, TV wireline or wireless wide area digital networks to include cable TV, satellite TV, and terrestrial broadcast TV, vehicular and industrial to include CANBus, and so forth. Certain networks commonly require external network interface adapters that attached to certain general purpose data ports or peripheral buses (2249) (such as, for example USB ports of the computer system (2200)); others are commonly integrated into the core of the computer system (2200) by attachment to a system bus as described below (for example Ethernet interface into a PC computer system or cellular network interface into a smartphone computer system). Using any of these networks, computer system (2200) can communicate with other entities. Such communication can be uni-directional, receive only (for example, broadcast TV), uni-directional send-only (for example CANbus to certain CANbus devices), or bi-directional, for example to other computer systems using local or wide area digital networks. Certain protocols and protocol stacks can be used on each of those networks and network interfaces as described above.

**[0200]** Aforementioned human interface devices, human-accessible storage devices, and network interfaces can be attached to a core (2240) of the computer system (2200).

**[0201]** The core (2240) can include one or more Central Processing Units (CPU) (2241), Graphics Processing Units (GPU) (2242), specialized programmable processing units in the form

of Field Programmable Gate Areas (FPGA) (2243), hardware accelerators for certain tasks (2244), and so forth. These devices, along with Read-only memory (ROM) (2245), Random-access memory (2246), internal mass storage such as internal non-user accessible hard drives, SSDs, and the like (2247), may be connected through a system bus (2248). In some computer systems, the system bus (2248) can be accessible in the form of one or more physical plugs to enable extensions by additional CPUs, GPU, and the like. The peripheral devices can be attached either directly to the core's system bus (2248), or through a peripheral bus (2249). Architectures for a peripheral bus include PCI, USB, and the like.

[0202] CPUs (2241), GPUs (2242), FPGAs (2243), and accelerators (2244) can execute certain instructions that, in combination, can make up the aforementioned computer code. That computer code can be stored in ROM (2245) or RAM (2246). Transitional data can be also be stored in RAM (2246), whereas permanent data can be stored for example, in the internal mass storage (2247). Fast storage and retrieve to any of the memory devices can be enabled through the use of cache memory, that can be closely associated with one or more CPU (2241), GPU (2242), mass storage (2247), ROM (2245), RAM (2246), and the like.

[0203] The computer readable media can have computer code thereon for performing various computer-implemented operations. The media and computer code can be those specially designed and constructed for the purposes of the present disclosure, or they can be of the kind well known and available to those having skill in the computer software arts.

[0204] As an example and not by way of limitation, the computer system having architecture (2200), and specifically the core (2240) can provide functionality as a result of processor(s) (including CPUs, GPUs, FPGA, accelerators, and the like) executing software embodied in one or more tangible, computer-readable media. Such computer-readable media can be media associated with user-accessible mass storage as introduced above, as well as certain storage of the core (2240) that are of non-transitory nature, such as core-internal mass storage (2247) or ROM (2245). The software implementing various embodiments of the present disclosure can be stored in such devices and executed by core (2240). A computer-readable medium can include one or more memory devices or chips, according to particular needs. The software can cause the core (2240) and specifically the processors therein (including CPU, GPU, FPGA, and the like) to execute particular processes or particular parts of particular processes described herein, including defining data structures stored in RAM (2246) and modifying such data structures according to the processes defined by the software. In addition or as an

alternative, the computer system can provide functionality as a result of logic hardwired or otherwise embodied in a circuit (for example: accelerator (2244)), which can operate in place of or together with software to execute particular processes or particular parts of particular processes described herein. Reference to software can encompass logic, and vice versa, where appropriate. Reference to a computer-readable media can encompass a circuit (such as an integrated circuit (IC)) storing software for execution, a circuit embodying logic for execution, or both, where appropriate. The present disclosure encompasses any suitable combination of hardware and software.

#### Appendix A: Acronyms

JEM: joint exploration model

VVC: versatile video coding

BMS: benchmark set

MV: Motion Vector

HEVC: High Efficiency Video Coding

SEI: Supplementary Enhancement Information

VUI: Video Usability Information

GOPs: Groups of Pictures

TUs: Transform Units,

PU: Prediction Units

CTUs: Coding Tree Units

CTBs: Coding Tree Blocks

PBs: Prediction Blocks

HRD: Hypothetical Reference Decoder

SNR: Signal Noise Ratio

CPUs: Central Processing Units

GPUs: Graphics Processing Units

CRT: Cathode Ray Tube

LCD: Liquid-Crystal Display

OLED: Organic Light-Emitting Diode

CD: Compact Disc

DVD: Digital Video Disc

ROM: Read-Only Memory

RAM: Random Access Memory

ASIC: Application-Specific Integrated Circuit

PLD: Programmable Logic Device

LAN: Local Area Network

GSM: Global System for Mobile communications

LTE: Long-Term Evolution

CANBus: Controller Area Network Bus

USB: Universal Serial Bus

PCI: Peripheral Component Interconnect

FPGA: Field Programmable Gate Areas

SSD: solid-state drive

IC: Integrated Circuit

CU: Coding Unit

**[0205]** While this disclosure has described several exemplary embodiments, there are alterations, permutations, and various substitute equivalents, which fall within the scope of the disclosure. It will thus be appreciated that those skilled in the art will be able to devise numerous systems and methods which, although not explicitly shown or described herein, embody the principles of the disclosure and are thus within the spirit and scope thereof.

## WHAT IS CLAIMED IS:

1. A method for video decoding in a decoder, comprising:  
decoding, from a coded video bitstream, coding information for a current picture, the coding information indicating that a geometric merge mode is enabled for a coding level higher than a picture level of the current picture and a maximum number of merge candidates satisfies a condition; and  
determining, based on a picture level parameter signaled for the current picture in the coded video bitstream, a maximum number of geometric merge mode merge candidates based on the picture level parameter and the maximum number of merge candidates, the maximum number of geometric merge mode merge candidates being (i) 0 or (ii) one of 2 to the maximum number of merge candidates, the picture level parameter indicating the maximum number of geometric merge mode merge candidates, wherein  
the geometric merge mode is disabled for the current picture based on the maximum number of geometric merge mode merge candidates being 0, and  
the geometric merge mode is enabled for the current picture based on the maximum number of geometric merge mode merge candidates not being 0.
2. The method of claim 1, wherein the geometric merge mode is a triangle partition mode (TPM) and the maximum number of geometric merge mode merge candidates is a maximum number of TPM merge candidates.
3. The method of claim 2, wherein the coding level is a sequence level.
4. The method of claim 2, wherein the condition is the maximum number of merge candidates is larger than or equal to 2.
5. The method of claim 3, wherein  
the condition is the maximum number of merge candidates is larger than or equal to 2;  
and  
the determining the maximum number of geometric merge mode merge candidates includes determining the maximum number of TPM merge candidates by subtracting the picture level parameter from the maximum number of merge candidates.

6. The method of claim 5, wherein  
a picture parameter set (PPS) level parameter that indicates the maximum number of TPM merge candidates is signaled in the coded video bitstream for a PPS associated with the current picture; and  
the PPS level parameter is (i) one of 0 to (the maximum number of merge candidates - 1) or (ii) (the maximum number of merge candidates + 1).

7. The method of claim 5, wherein a picture parameter set (PPS) level parameter indicating the maximum number of TPM merge candidates is not signaled in the coded video bitstream for a PPS associated with the current picture.

8. The method of claim 7, wherein  
the coded video bitstream includes a picture header for the current picture; and  
the picture level parameter is signaled in the picture header based on the TPM being enabled for the sequence level and the maximum number of merge candidates being larger than or equal to 2, the signaling of the picture level parameter being independent from the PPS level parameter.

9. The method of claim 5, wherein  
the coded video bitstream includes a picture parameter set (PPS) associated with the current picture; and  
a PPS level parameter indicating the maximum number of TPM merge candidates is signaled in the PPS based at least on a PPS level flag indicating that the PPS level parameter is to be signaled.

10. The method of claim 9, wherein  
the coded video bitstream includes a picture header for the current picture; and  
the picture level parameter is signaled in the picture header based on the TPM being enabled for the sequence level, the maximum number of merge candidates being larger than or equal to 2, and the PPS level flag indicating that the PPS level parameter is not to be signaled.

11. The method of claim 5, wherein

the coded video bitstream includes a picture parameter set (PPS) associated with the current picture; and

a PPS level parameter indicating the maximum number of TPM merge candidates is signaled in the PPS based at least on the TPM being enabled for the sequence level.

12. A method for video decoding in a decoder, comprising:

decoding, from a coded video bitstream, coding information for a current picture, the coding information indicating a geometric merge mode being enabled at a sequence level, a picture parameter set (PPS) level parameter in a PPS being 0, and a maximum number of merge candidates, the PPS level parameter indicating a maximum number of geometric merge mode merge candidates; and

decoding a picture level parameter signaled for the current picture in the coded video bitstream based on the maximum number of merge candidates satisfying a condition, the picture level parameter indicating the maximum number of geometric merge mode merge candidates.

13. The method of claim 12, wherein the geometric merge mode is a triangle partition mode (TPM) and the maximum number of geometric merge mode merge candidates is a maximum number of TPM merge candidates.

14. The method of claim 13, wherein the condition is one of (i) the maximum number of merge candidates is larger than 2 and (ii) the maximum number of merge candidates is larger than or equal to 3.

15. The method of claim 14, wherein,

the maximum number of merge candidates is 2 and does not satisfy the condition, and the picture level parameter is not signaled in the coded video bitstream; and

the method further includes determining that the maximum number of TPM merge candidates is 2.

16. An apparatus for video decoding, comprising processing circuitry configured to:

decode, from a coded video bitstream, coding information for a current picture, the coding information indicating that a geometric merge mode is enabled for a coding level higher

than a picture level of the current picture and a maximum number of merge candidates satisfies a condition; and

determine, based on a picture level parameter signaled for the current picture in the coded video bitstream, a maximum number of geometric merge mode merge candidates based on the picture level parameter and the maximum number of merge candidates, the maximum number of geometric merge mode merge candidates being (i) 0 or (ii) one of 2 to the maximum number of merge candidates, the picture level parameter indicating the maximum number of geometric merge mode merge candidates, wherein

the geometric merge mode is disabled for the current picture based on the maximum number of geometric merge mode merge candidates being 0, and

the geometric merge mode is enabled for the current picture based on the maximum number of geometric merge mode merge candidates not being 0.

17. The apparatus of claim 16, wherein the geometric merge mode is a triangle partition mode (TPM) and the maximum number of geometric merge mode merge candidates is a maximum number of TPM merge candidates.

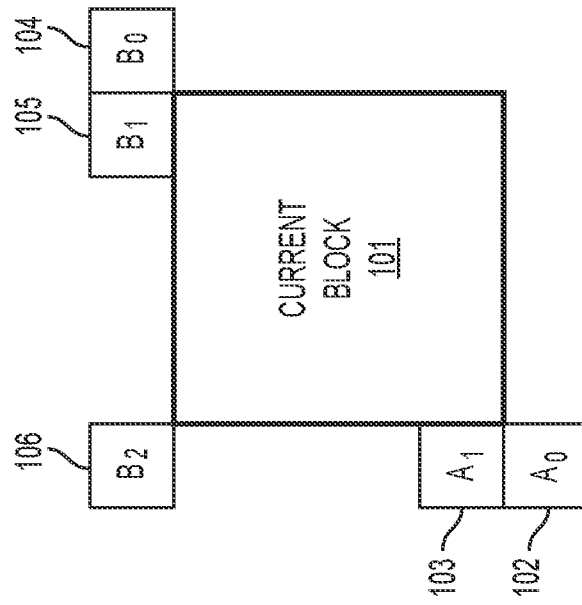
18. The apparatus of claim 17, wherein the coding level is a sequence level.

19. The apparatus of claim 17, wherein the condition is the maximum number of merge candidates is larger than or equal to 2.

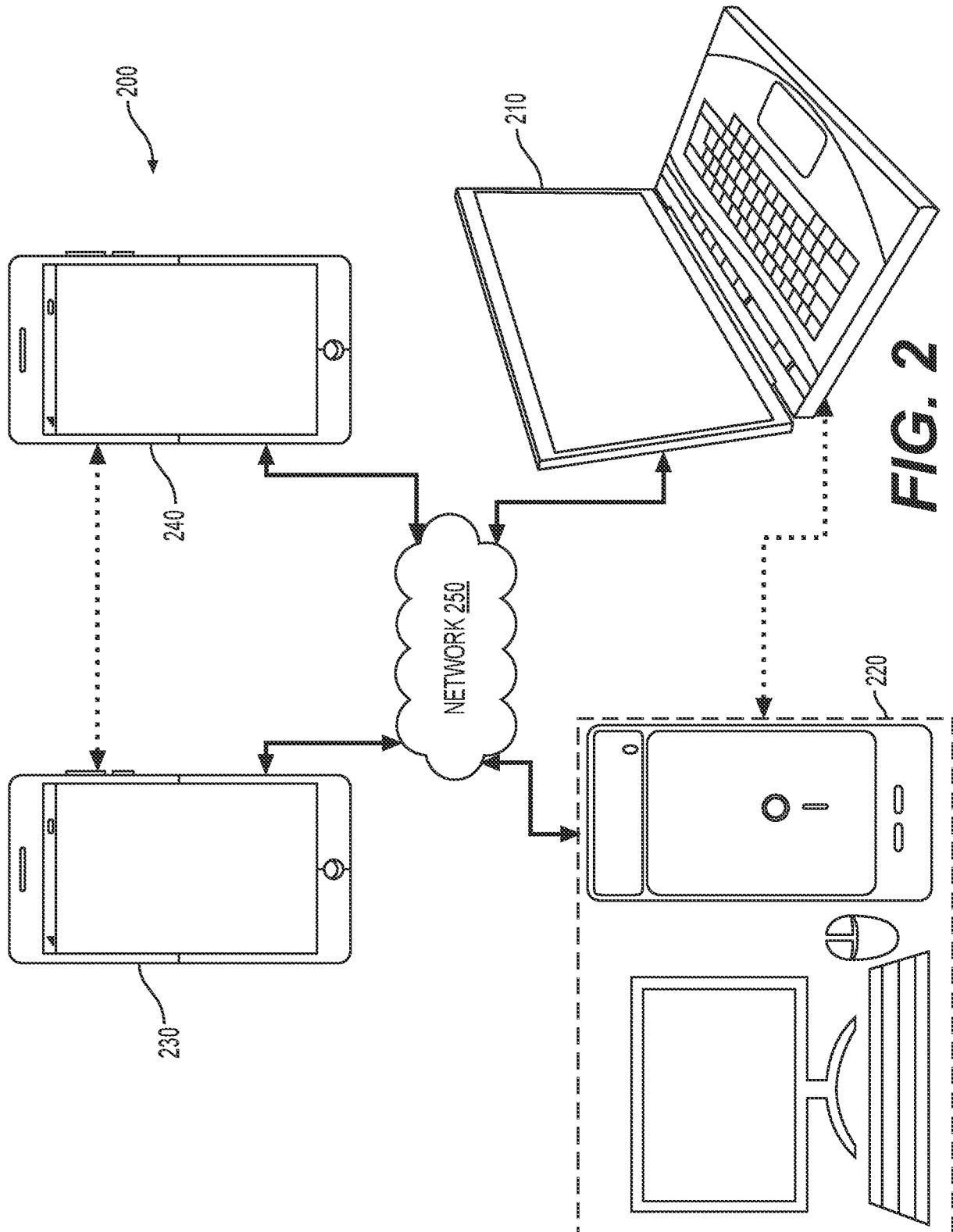
20. The apparatus of claim 18, wherein  
the condition is the maximum number of merge candidates is larger than or equal to 2;  
and

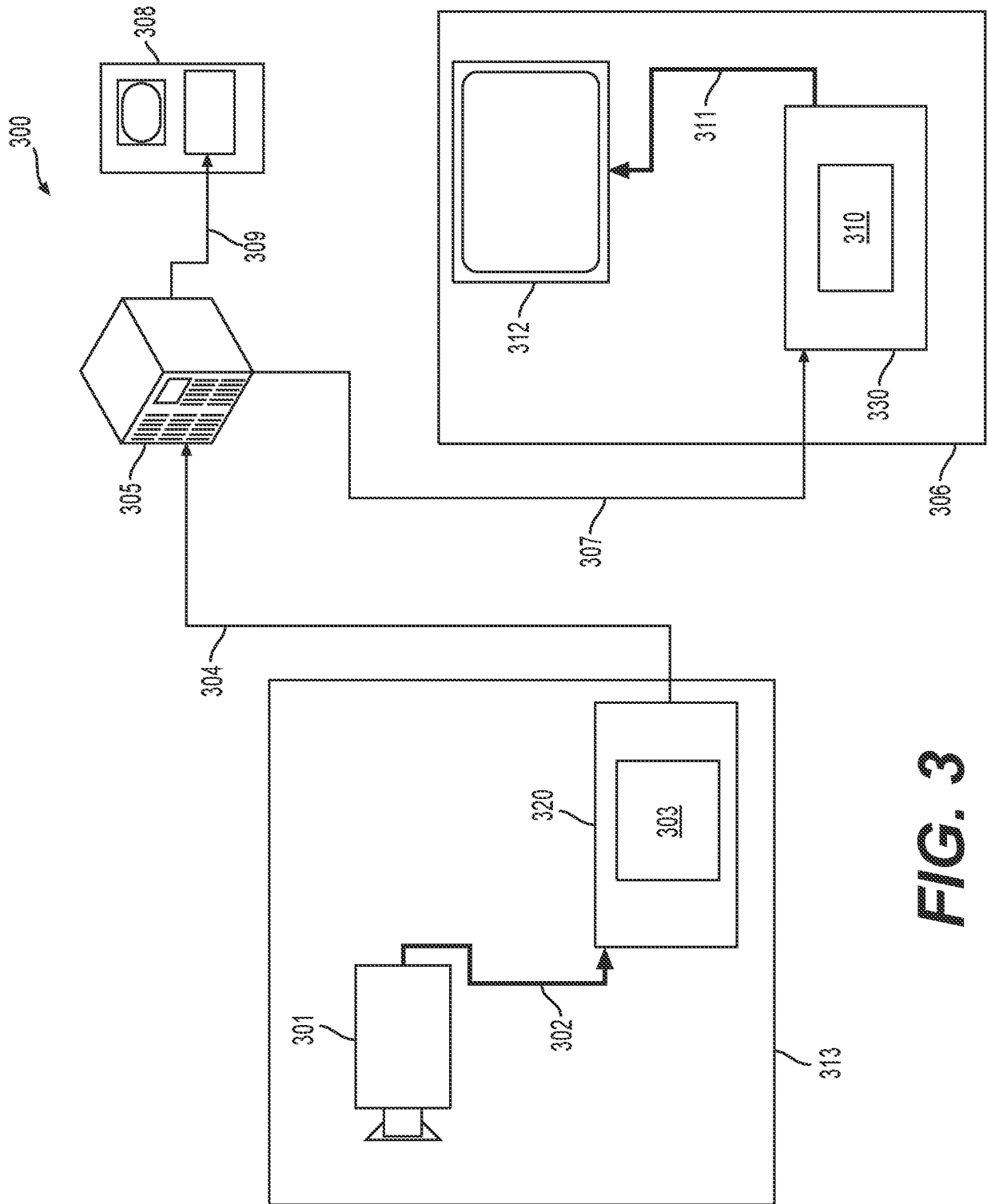
the processing circuitry is configured to determine the maximum number of TPM merge candidates by subtracting the picture level parameter from the maximum number of merge candidates.

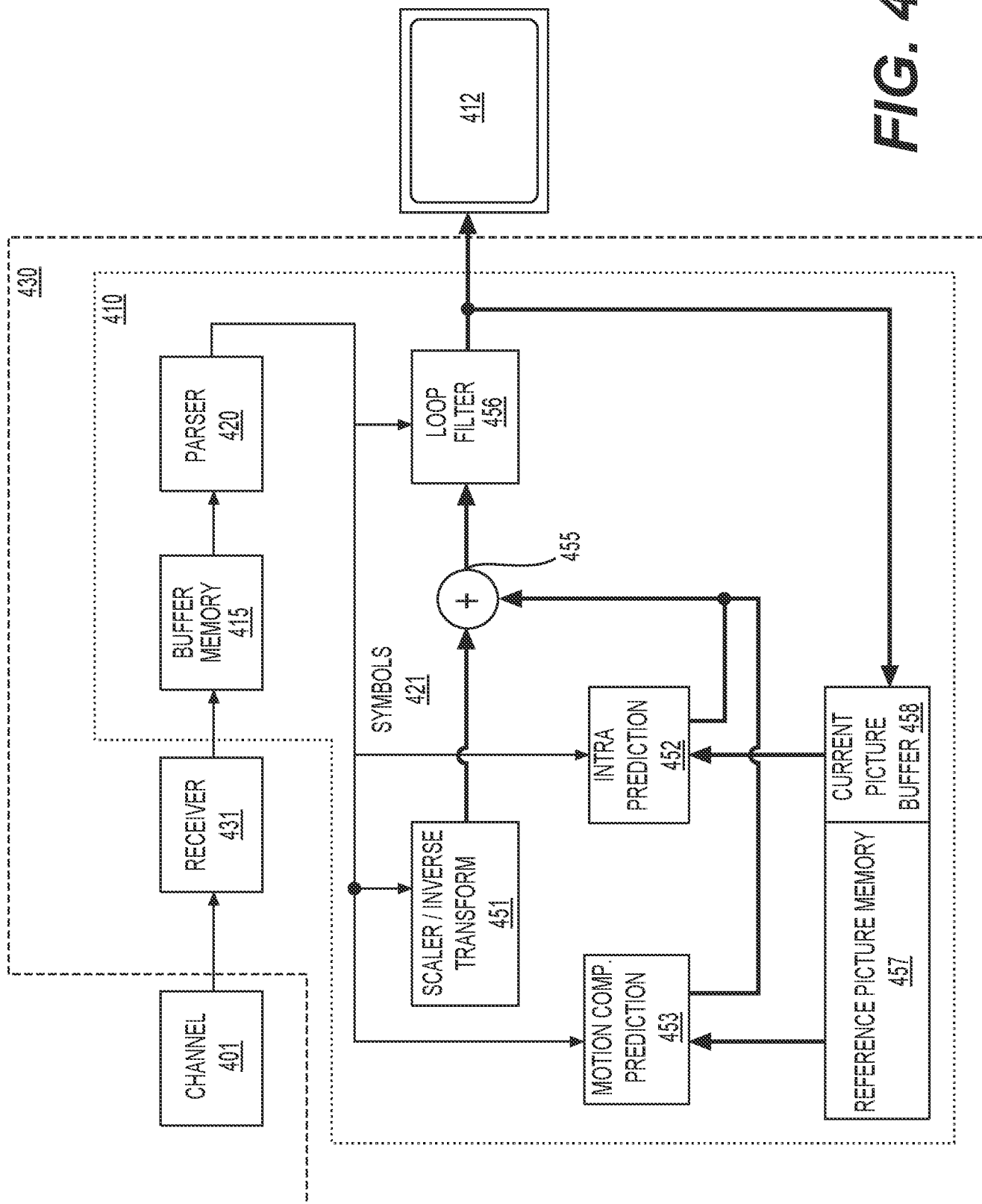


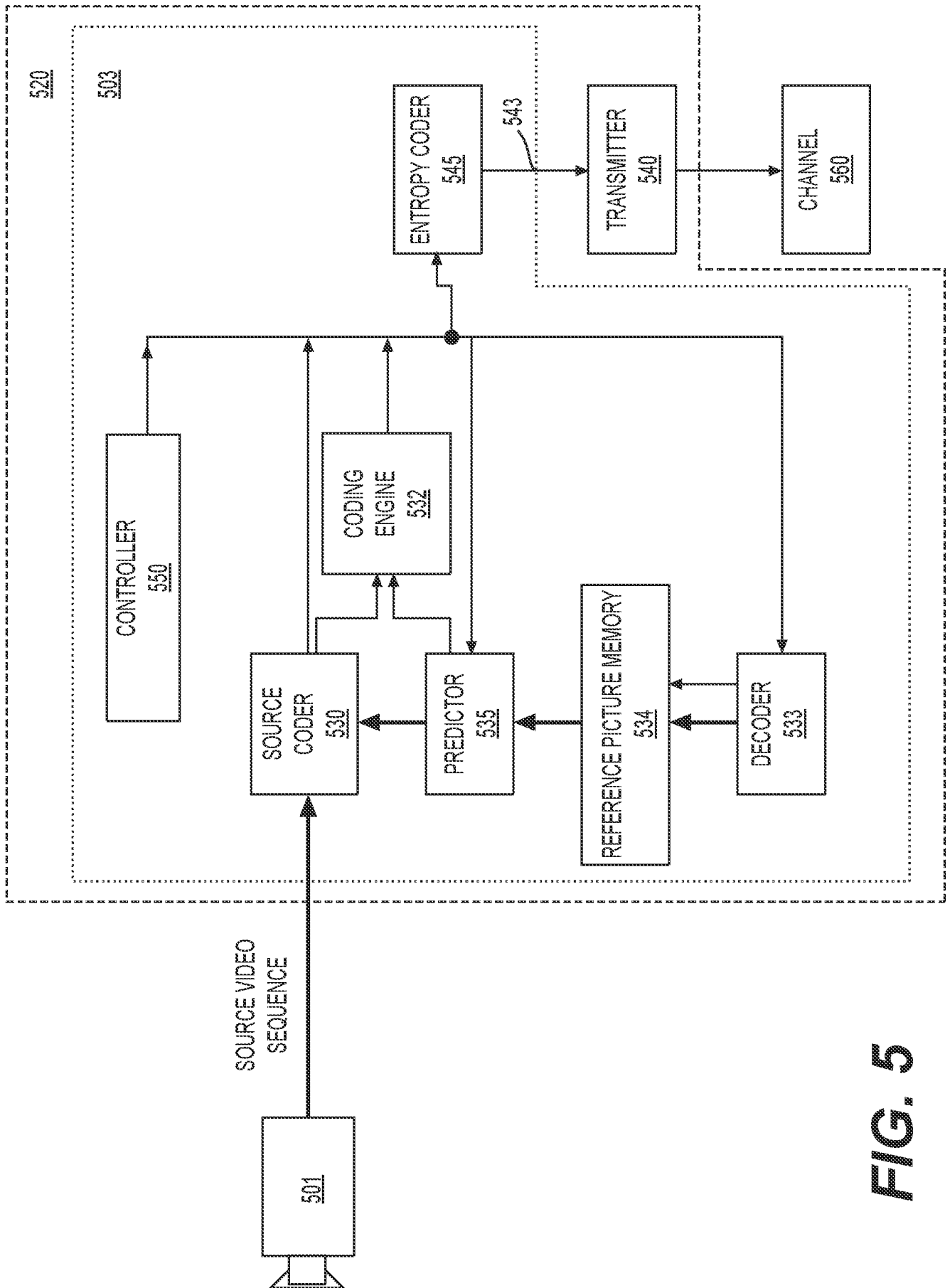


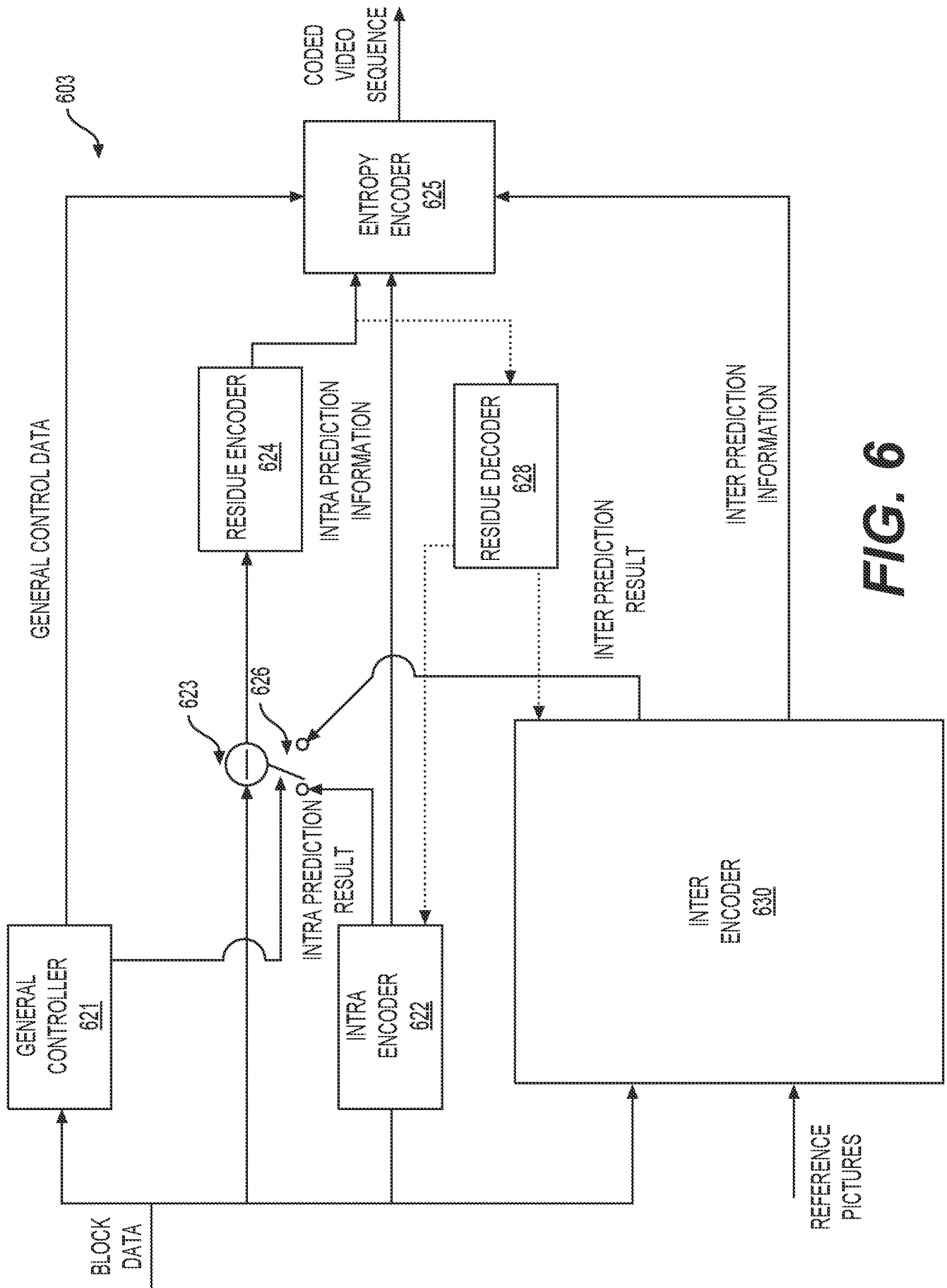
**FIG. 1**  
(Related Art)

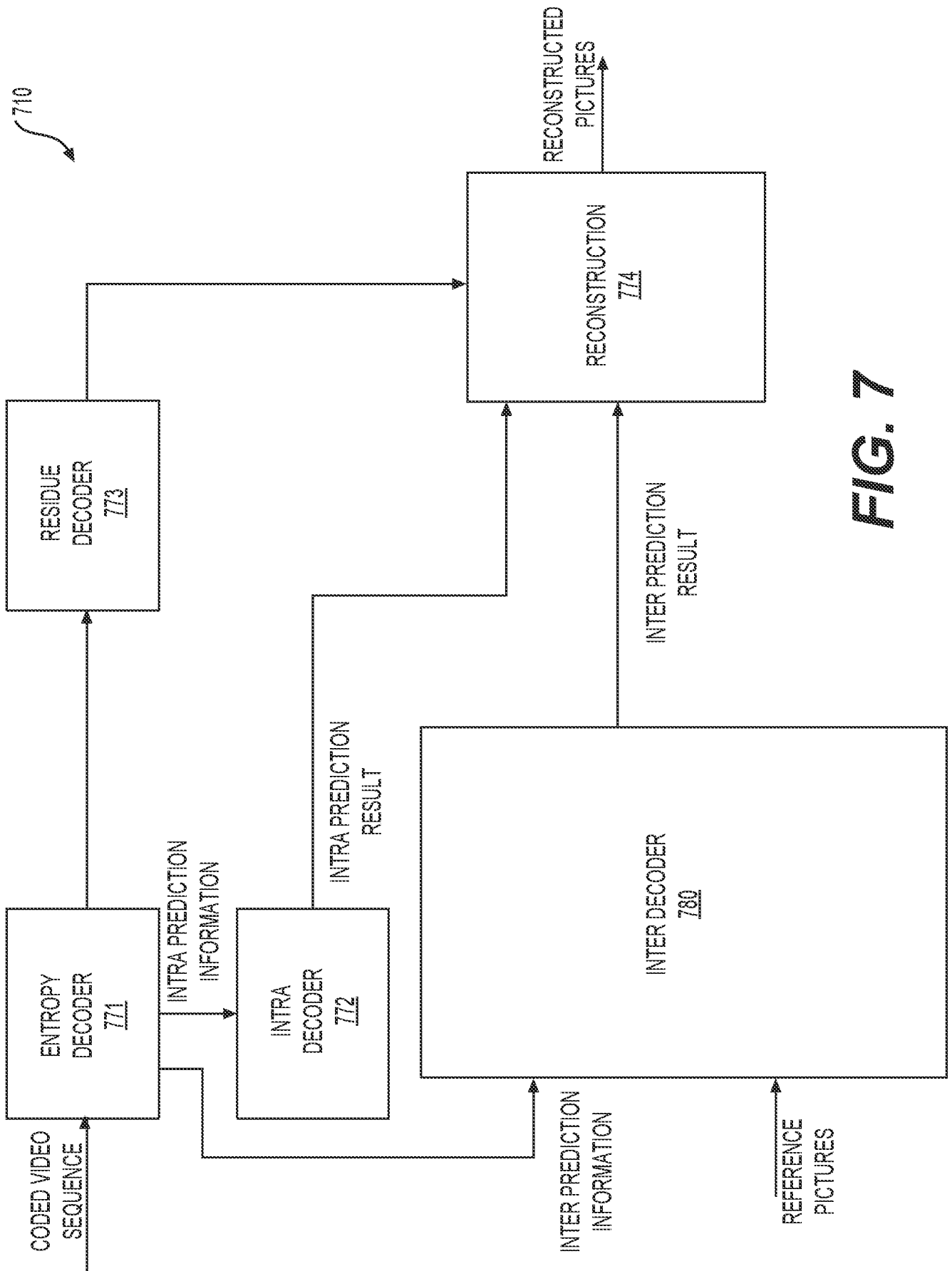


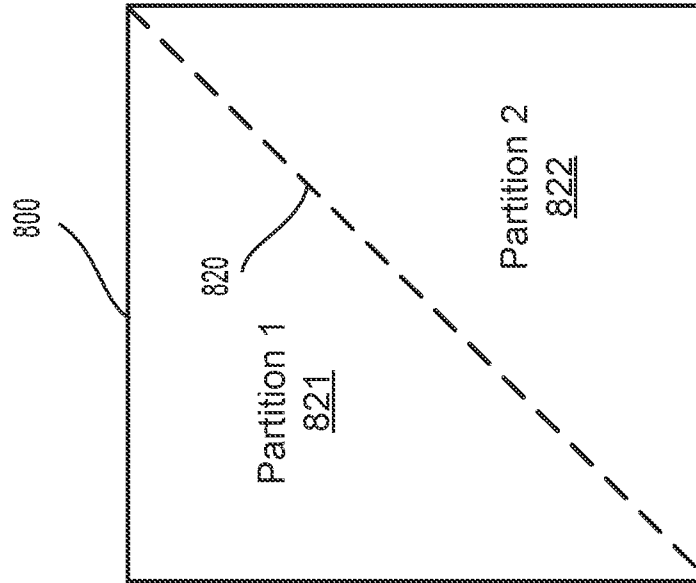
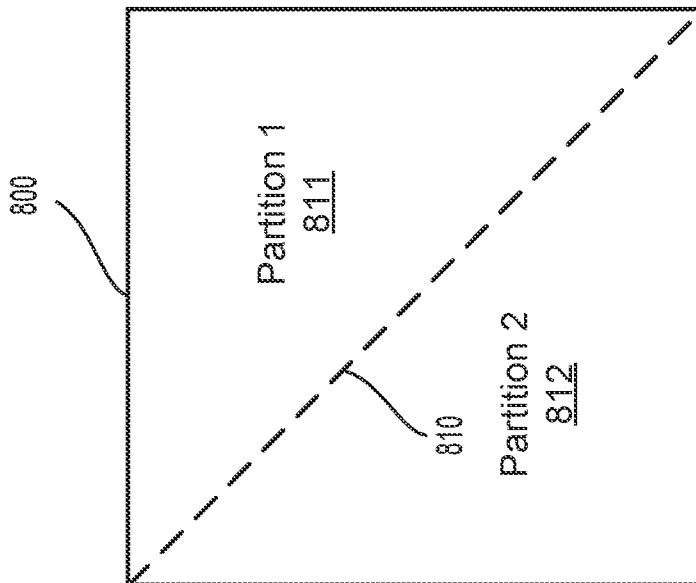
**FIG. 3**



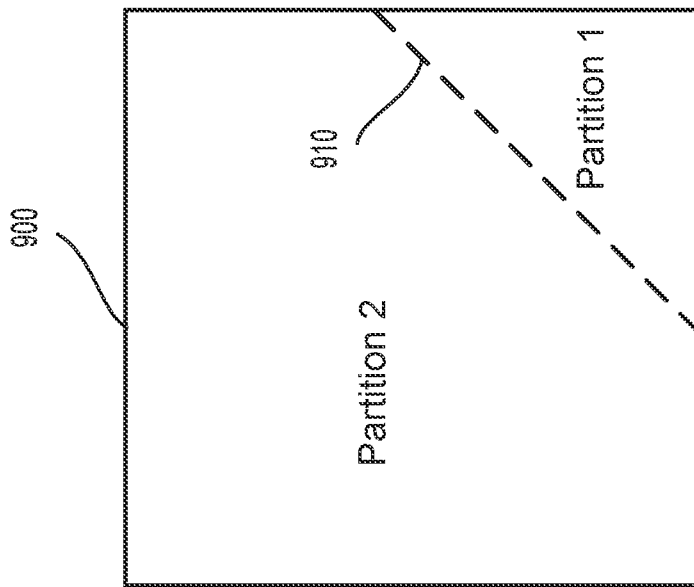
**FIG. 5**

**FIG. 6**

**FIG. 7**

**FIG. 8A****FIG. 8B**



**FIG. 9**

seq_parameter_set_rbsp() {	Descriptor
...	
sps_triangle_enabled_flag	u(1)
...	
}	

FIG. 10

pic_parameter_set_rbsp() {	Descriptor
...	
constant_slice_header_params_enabled_flag	u(1)
if( constant_slice_header_params_enabled_flag ) {	
pps_dep_quant_enabled_idc	u(2)
for( i = 0; i < 2; i++ )	
pps_ref_pic_list_sps_idc[ i ]	u(2)
pps_mvd_11_zero_idc	u(2)
pps_collocated_from_l0_idc	u(2)
pps_six_minus_max_num_merge_cand_plus1	ue(v)
pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1	ue(v)
}	
...	
}	

FIG. 11

picture_header_rbsp() {	Descriptor
...	
if( sps_triangle_enabled_flag && MaxNumMergeCand >= 2 && !pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1 )	
pic_max_num_merge_cand_minus_max_num_triangle_cand	ue(v)
...	
}	

FIG. 12

1310

picture_header_rbsp() {	Descriptor
...	
if( sps_triangle_enabled_flag && MaxNumMergeCand <div>&gt;</div> 2 && lpps_max_num_merge_cand_minus_max_num_triangle_cand_plus1 )	
pic_max_num_merge_cand_minus_max_num_triangle_cand	ue(v)
...	
}	

FIG. 13

1410

picture_header_rbsp() {	Descriptor
...	
if ( sps_triangle_enabled_flag && MaxNumMergeCand	
!pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1 )	
pic_max_num_merge_cand_minus_max_num_triangle_cand	ue(v)
...	
}	

FIG. 14

pic_parameter_set_rbsp() {	Descriptor
...	
constant_slice_header_params_enabled_flag	u(1)
if( constant_slice_header_params_enabled_flag ) {	
pps_dep_quant_enabled_idc	u(2)
for( i = 0; i < 2; i++ )	
pps_ref_pic_list_sps_idc[ i ]	u(2)
pps_mvd_l1_zero_idc	u(2)
pps_collocated_from_l0_idc	u(2)
pps_six_minus_max_num_merge_cand_plus1	ue(v)
pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1	ue(4)
}	
...	
}	

1510

FIG. 15

picture_header_rbsp() {	Descriptor
...	
if( sps_triangle_enabled_flag && MaxNumMergeCand >= 2	
<div>    &amp;&amp;pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1</div>	
pic_max_num_merge_cand_minus_max_num_triangle_cand	ue(v)
...	
}	

1610

FIG. 16A

When pic\_max\_num\_merge\_cand\_minus\_max\_num\_triangle\_cand is not present, and  
sps\_triangle\_enabled\_flag is equal to 1 and MaxNumMergeCand greater than or equal to 2,  
pic\_max\_num\_merge\_cand\_minus\_max\_num\_triangle\_cand is inferred to be equal to  
pps\_max\_num\_merge\_cand\_minus\_max\_num\_triangle\_cand\_plus1.

1620

FIG. 16B



	Descriptor
pic_parameter_set_rbsp() {	
...	
constant_slice_header_params_enabled_flag	u(1)
if( constant_slice_header_params_enabled_flag ) {	
pps_dep_quant_enabled_idc	u(2)
for( i = 0; i < 2; i++ )	
pps_ref_pic_list_sps_idc[ i ]	u(2)
pps_mvd_l1_zero_idc	u(2)
pps_collocated_from_l0_idc	u(2)
pps_six_minus_max_num_merge_cand_plus1	ue(v)
pps_max_num_triangle_merge_cand_present_flag	u(1)
if( pps_max_num_triangle_merge_cand_present_flag ) {	
pps_max_num_merge_cand_minus_max_num_triangle_cand	ue(v)
}	
...	
}	

1610

1620

FIG. 17

picture_header_rbsp() {	Descriptor
...	
if( sps_triangle_enabled_flag && MaxNumMergeCand >= 2 &&	
! pps_max_num_triangle_merge_cand_present_flag)	
pic_max_num_merge_cand_minus_max_num_triangle_cand	ue(v)
...	
}	

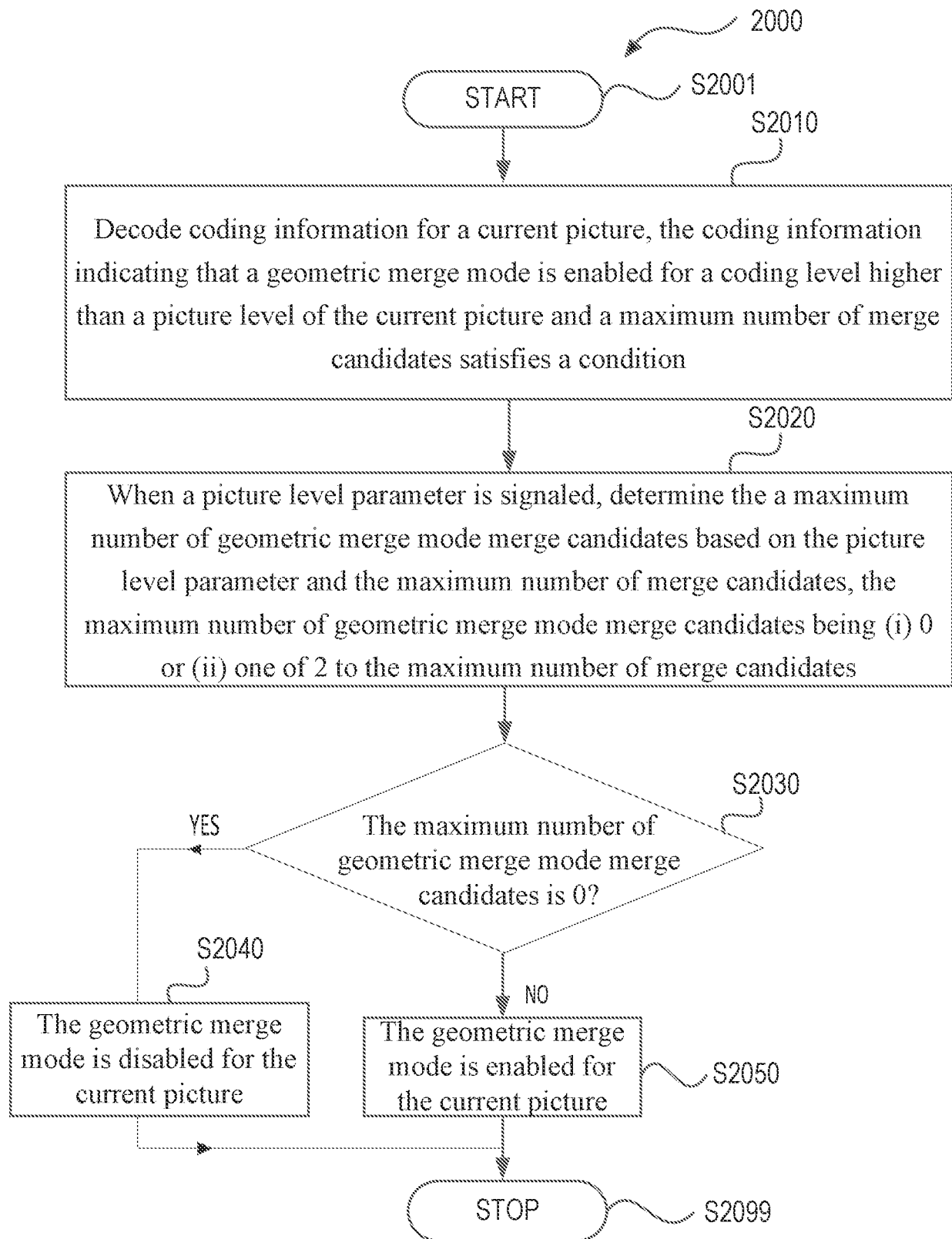
1840

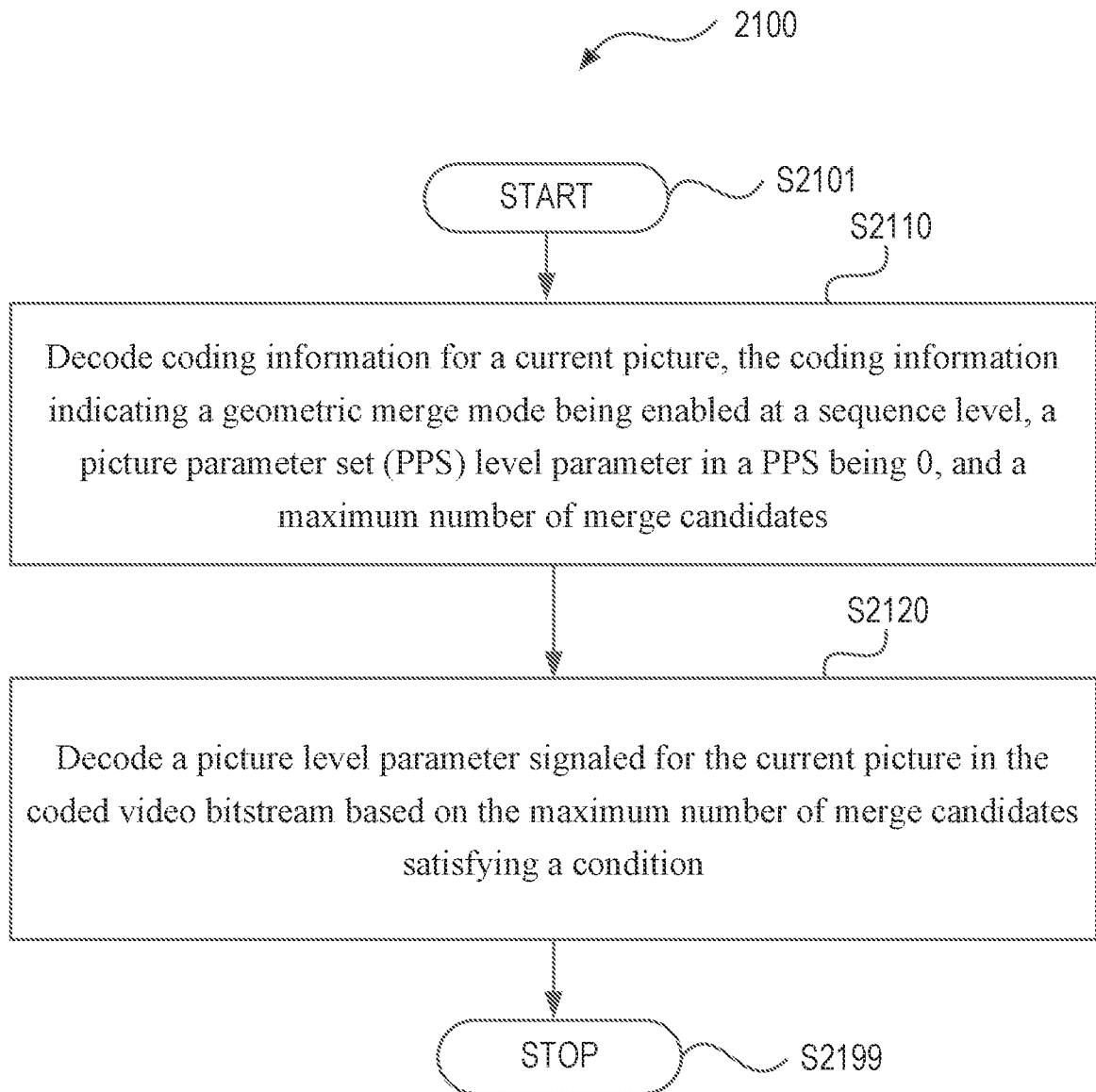
FIG. 18

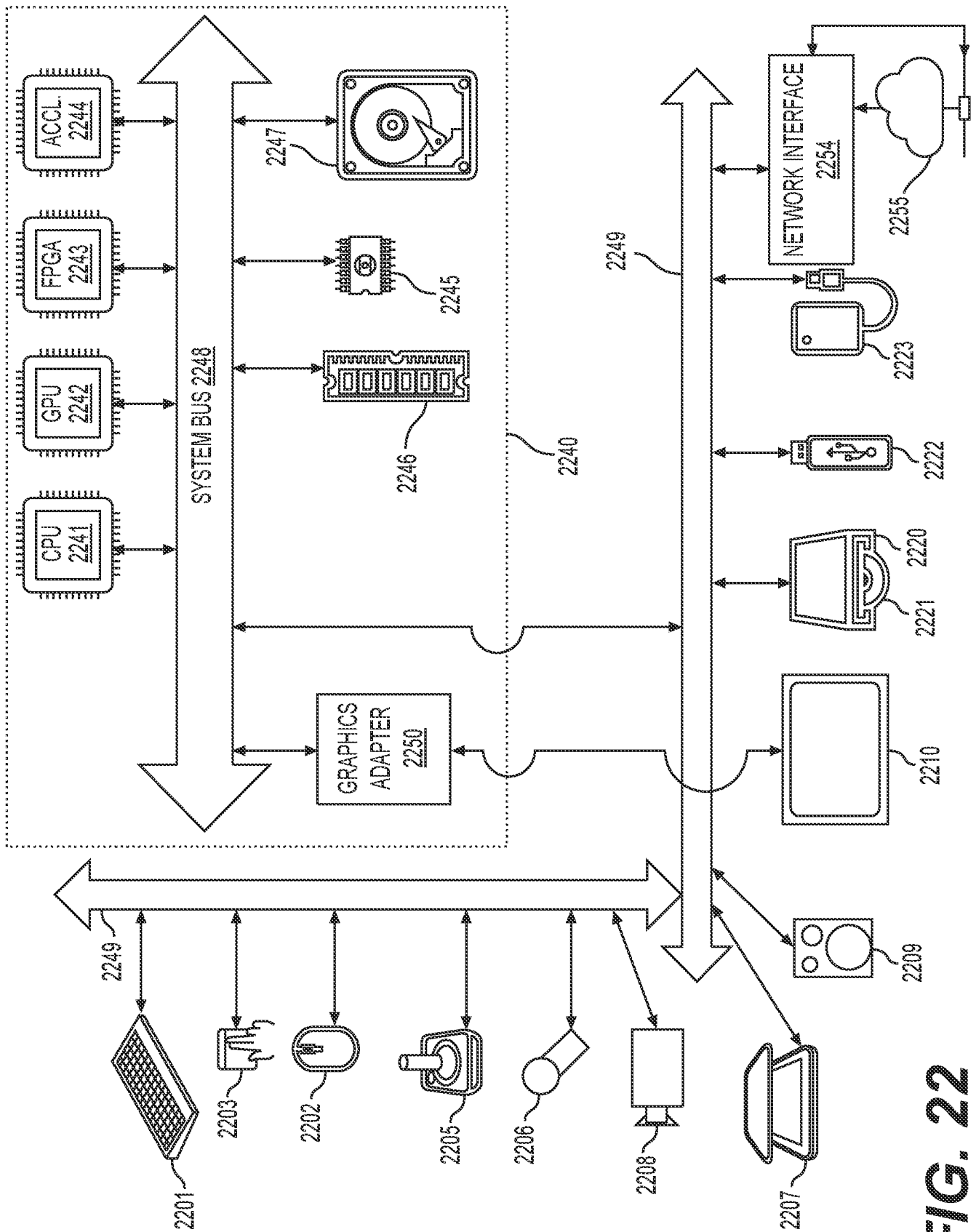
pic parameter set_rbsp() {	Descriptor
...	
constant_slice_header_params_enabled_flag	u(1)
if( constant_slice_header_params_enabled_flag ) {	
pps_dep_quant_enabled_idc	u(2)
for( i = 0; i < 2; i++ )	
pps_ref_pic_list_sps_idc[ i ]	u(2)
pps_mvd_l1_zero_idc	u(2)
pps_collocated_from_l0_idc	u(2)
pps_six_minus_max_num_merge_cand_plus1	ue(v)
if( sps_triangle_enabled_flag )	
pps_max_num_merge_cand_minus_max_num_triangle_cand_plus1	ue(v)
}	
...	
}	

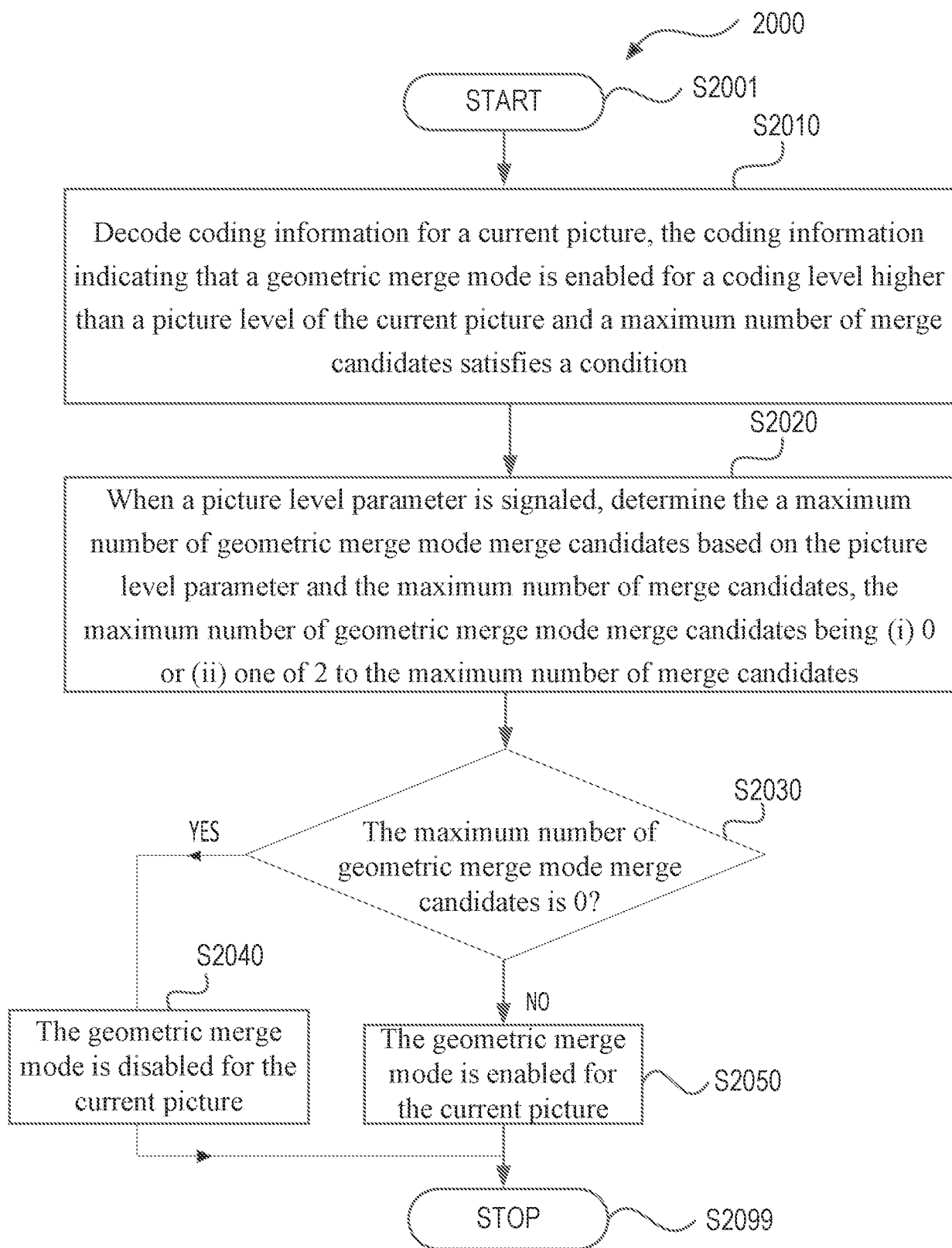
1910

FIG. 19

**FIG. 20**

**FIG. 21**

**FIG. 22**



**FIG. 20**