



- (51) **International Patent Classification:**
Not classified

(21) **International Application Number:**
PCT/US2024/037639

(22) **International Filing Date:**
11 July 2024 (11.07.2024)

(25) **Filing Language:**
English

(26) **Publication Language:**
English

(30) **Priority Data:**
63/513,268 12 July 2023 (12.07.2023) US

(71) **Applicant:** BYTEDANCE INC. [US/US]; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US).

(72) **Inventors:** VISHWANATH, Bharath; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US). ZHANG, Kai; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US). ZHANG, Li; 12655 West Jefferson
- Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US).

(74) **Agent:** HOWELL, Brandt D. et al.; Conley Rose, P.C., 4965 Preston Park Boulevard, Suite 195E, PLANO, Texas 75093 (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, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, 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, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST,

(54) **Title:** SKIP PROCESSING A NODE IN REGION-ADAPTIVE HIERARCHICAL TRANSFORM CODING

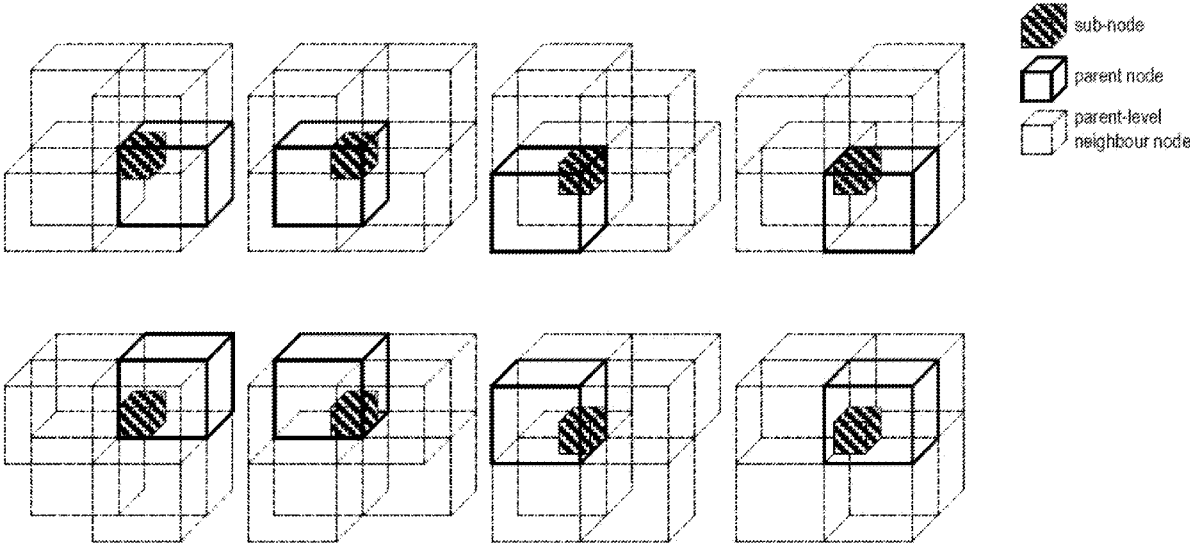


FIG. 1

(57) **Abstract:** A mechanism for processing video data is disclosed. The mechanism includes determining to use an available prediction in a sum of attribute space as a reconstruction instead of performing an inverse transform of a residue to obtain a prediction when all alternating current (AC) coefficients in a 2x2x2 Region-Adaptive Hierarchical Transform (RAHT) node are quantized to zero. A conversion is then performed between a visual media data and a bitstream based on the available prediction.

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, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

- *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

Skip Processing a Node In Region-Adaptive Hierarchical Transform Coding

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This patent application claims the benefit of U.S. Patent Application No. International 62/513,268 filed on July 12, 2023, which is hereby incorporated by reference.

TECHNICAL FIELD

[0002] The present disclosure relates to generation, storage, and consumption of digital audio video media information in a file format.

BACKGROUND

[0003] Digital video accounts for the largest bandwidth used on the Internet and other digital communication networks. As the number of connected user devices capable of receiving and displaying video increases, the bandwidth demand for digital video usage is likely to continue to grow.

SUMMARY

[0004] A first aspect relates to a method for processing video data, comprising: determining to use an available prediction in a sum of attribute space as a reconstruction instead of performing an inverse transform of a residue to obtain a prediction when all alternating current (AC) coefficients in a 2x2x2 Region-Adaptive Hierarchical Transform (RAHT) node are quantized to zero; and performing a conversion between a visual media data and a bitstream based on the available prediction.

[0005] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the available prediction and the reconstruction are identical.

[0006] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the inverse transform is not performed to obtain the reconstruction.

[0007] Optionally, in any of the preceding aspects, another implementation of the aspect provides determining to disable one or more RAHT operations and to inherit direct current (DC) values when there are no AC coefficients in the 2x2x2 RAHT node.

[0008] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the RAHT operations comprise a forward transform.

[0009] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the RAHT operations comprise an inverse transform.

[0010] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the RAHT operations comprise quantization.

[0011] Optionally, in any of the preceding aspects, another implementation of the aspect provides that all of the RAHT operations are disabled when there are no AC coefficients in the 2x2x2 RAHT node.

[0012] Optionally, in any of the preceding aspects, another implementation of the aspect provides that a subset of the RAHT operations are disabled.

[0013] Optionally, in any of the preceding aspects, another implementation of the aspect provides that whether to and/or how to apply the one or more of the methods is signalled in a frame, tile, slice, octree, or the bitstream.

[0014] Optionally, in any of the preceding aspects, another implementation of the aspect provides that whether to and/or how to apply the one or more of the methods is dependent on coded information, and wherein the coded information comprises one or more dimensions, a color format, a color component, a slice type, or a picture type.

[0015] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the conversion includes encoding the media data into the bitstream.

[0016] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the conversion includes decoding the media data from the bitstream.

[0017] A second aspect relates to an apparatus for processing video data comprising: a processor; and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to perform any of the disclosed methods.

[0018] A third aspect relates to a non-transitory computer readable medium comprising a computer program product for use by a video coding device, the computer program product comprising computer executable instructions stored on the non-transitory computer readable medium such that when executed by a processor cause the video coding device to perform any of the disclosed methods.

[0019] A fourth aspect relates to a non-transitory computer-readable recording medium storing a bitstream of a video which is generated by a method performed by a video processing apparatus, wherein the method comprises: determining to use an available prediction in a sum of attribute space

as a reconstruction instead of performing an inverse transform of a residue to obtain a prediction when all alternating current (AC) coefficients in a 2x2x2 Region-Adaptive Hierarchical Transform (RAHT) node are quantized to zero; and generating the bitstream based on the available prediction.

[0020] A fifth aspect relates to a method for storing bitstream of a video comprising: determining to use an available prediction in a sum of attribute space as a reconstruction instead of performing an inverse transform of a residue to obtain a prediction when all alternating current (AC) coefficients in a 2x2x2 Region-Adaptive Hierarchical Transform (RAHT) node are quantized to zero; generating the bitstream based on the available prediction; and storing the bitstream in a non-transitory computer-readable recording medium.

[0021] A sixth aspect relates to a method, apparatus, or system described in the present disclosure.

[0022] For the purpose of clarity, any one of the foregoing embodiments may be combined with any one or more of the other foregoing embodiments to create a new embodiment within the scope of the present disclosure.

[0023] These and other features will be more clearly understood from the following detailed description taken in conjunction with the accompanying drawings and claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0024] For a more complete understanding of this disclosure, reference is now made to the following brief description, taken in connection with the accompanying drawings and detailed description, wherein like reference numerals represent like parts.

[0025] FIG. 1 is an example of parent-level nodes for each sub-node of transform unit node.

[0026] FIG. 2 is a block diagram showing an example video processing system.

[0027] FIG. 3 is a block diagram of an example video processing apparatus.

[0028] FIG. 4 is a flowchart for an example method of video processing.

[0029] FIG. 5 is a block diagram that illustrates an example video coding system.

[0030] FIG. 6 is a block diagram that illustrates an example encoder.

[0031] FIG. 7 is a block diagram that illustrates an example decoder.

[0032] FIG. 8 is a schematic diagram of an example encoder.

DETAILED DESCRIPTION

[0033] It should be understood at the outset that although an illustrative implementation of one or more embodiments are provided below, the disclosed systems and/or methods may be

implemented using any number of techniques, whether currently known or yet to be developed. The disclosure should in no way be limited to the illustrative implementations, drawings, and techniques illustrated below, including the exemplary designs and implementations illustrated and described herein, but may be modified within the scope of the appended claims along with their full scope of equivalents.

1. Initial discussion

[0034] This disclosure is related to media file format. Specifically, it is related to point cloud attribute inter prediction in region-adaptive hierarchical transform. The ideas may be applied individually or in various combination, to any point cloud coding standard or non-standard point cloud codec, e.g., the being-developed Geometry based Point Cloud Compression (G-PCC).

2. Abbreviations

G-PCC	Geometry based Point Cloud Compression
MPEG	Moving Picture Experts Group
3DG	Three Dimensional Graphics Coding Group
CFP	Call For Proposal
V-PCC	Video-based Point Cloud Compression
RAHT	Region-Adaptive Hierarchical Transform

3. Further discussion

[0035] MPEG, short for Moving Picture Experts Group, is one of the main standardization groups dealing with multimedia. In 2017, the MPEG 3D Graphics Coding group (3DG) published a call for proposals (CFP) document to start to develop point cloud coding standard[1]. The final standard will consist in two classes of solutions. Video-based Point Cloud Compression (V-PCC) is appropriate for point sets with a relatively uniform distribution of points[2]. Geometry-based Point Cloud Compression (G-PCC) is appropriate for more sparse distributions[3]. Both V-PCC and G-PCC support the coding and decoding for single point cloud and point cloud sequence.

[0036] In one-point cloud, there may be geometry information and attribute information. Geometry information is used to describe the geometry locations of the data points. Attribute information is used to record some details of the data points, such as textures, normal vectors, reflections and so on.

3.1 Region-Adaptive Hierarchical Transform

[0037] In G-PCC, one of important point cloud attribute coding tools is RAHT. It is a transform that uses the attributes associated with a node in a lower level of the octree to predict the attributes of the nodes in the next level[4]. It assumes that the positions of the points are given at both the encoder and decoder. RAHT follows the octree scan backwards, from leaf nodes to root node, at each step recombining nodes into larger ones until reaching the root node. At each level of octree, the nodes are processed in the Morton order. At each decomposition, instead of grouping eight nodes at a time, RAHT does it in three steps along each dimension, (e.g., along z , then y then x). If there are L levels in octree, RAHT takes $3L$ levels to traverse the tree backwards.

[0038] Let the nodes at level l be $g_{l,x,y,z}$, for x, y, z integers. $g_{l,x,y,z}$ was obtained by grouping $g_{l+1,2x,y,z}$ and $g_{l+1,2x+1,y,z}$, where the grouping along the first dimension was an example. RAHT only process occupied nodes. If one of the nodes in the pair is unoccupied, the other one is promoted to the next level, unprocessed, i.e., $g_{l-1,x,y,z} = g_{l,2x,y,z}$ if the latter is the occupied node of the pair. The grouping process is repeated until getting to the root. Note that the grouping process generates nodes at lower levels that are the result of grouping different numbers of voxels along the way. The number of nodes grouped to generate node $g_{l,x,y,z}$ is the weight $\omega_{l,x,y,z}$ of that node.

[0039] At every grouping of two nodes, say $g_{l,2x,y,z}$ and $g_{l,2x+1,y,z}$, with their respective weights, $\omega_{l,2x,y,z}$ and $\omega_{l,2x+1,y,z}$, RAHT apply the following transform:

$$\begin{bmatrix} g_{l-1,x,y,z} \\ h_{l-1,x,y,z} \end{bmatrix} = \mathbf{T}_{\omega_1\omega_2} \begin{bmatrix} g_{l,2x,y,z} \\ h_{l,2x+1,y,z} \end{bmatrix},$$

[0040] Where $\omega_1 = \omega_{l,2x,y,z}$ and $\omega_2 = \omega_{l,2x+1,y,z}$ and

$$\mathbf{T}_{\omega_1\omega_2} = \frac{1}{\sqrt{\omega_1+\omega_2}} \begin{bmatrix} \sqrt{\omega_1} & \sqrt{\omega_2} \\ -\sqrt{\omega_2} & \sqrt{\omega_1} \end{bmatrix}.$$

[0041] Note that the transform matrix changes at all times, adapting to the weights, i.e., adapting to the number of leaf nodes that each $g_{l,x,y,z}$ actually represents. The quantities $g_{l,x,y,z}$ are used to group and compose further nodes at a lower level. $h_{l,x,y,z}$ are the actual high-pass coefficients generated by the transform to be encoded and transmitted. Furthermore, weights accumulate for the level above. In the above example,

$$\omega_{l-1,2x,y,z} = \omega_{l,2x,y,z} + \omega_{l,2x+1,y,z}$$

[0042] In the last stage, the tree root, the remaining two voxels $g_{1,0,0,0}$ and $g_{1,1,0,0}$ are transformed into the final two coefficients as:

$$\begin{bmatrix} g_{DC} \\ h_{0,0,0,0} \end{bmatrix} = T_{\omega_{1,0,0,0}\omega_{1,1,0,0}} \begin{bmatrix} g_{1,0,0,0} \\ g_{1,1,0,0} \end{bmatrix}$$

[0043] Where $g_{DC} = g_{0,0,0,0}$.

3.2 Upsampled transform domain prediction in RAHT

[0044] FIG. 1 is an example of parent-level nodes for each sub-node of transform unit node.

[0045] The transform domain prediction is introduced to improve coding efficiency on RAHT[5]. It is formed of two parts.

[0046] Firstly, the RAHT tree traversal is changed to be descent based from the previous ascent approach, i.e., a tree of attribute and weight sums is constructed and then RAHT is performed from the root of the tree to the leaves for both the encoder and the decoder. The transform is also performed in octree node transform unit that has $2 \times 2 \times 2$ sub-nodes. Within the node, the encoder transform order is from leaves to the root.

[0047] Secondly, for each sub-node of transform unit, a corresponding predicted sub-node is produced by upsampling the previous transform level. Actually, only sub-node that contains at last one point will produce a corresponding predicted sub-node. The transform unit that contains $2 \times 2 \times 2$ predicted sub-nodes is transformed and subtracted from the transformed attributes at the encoder side.

[0048] Each sub-node of transform unit node is predicted by 7 parent-level nodes where 3 coline parent-level neighbour nodes, 3 coplane parent-level neighbour nodes and 1 parent node. Coplane and coline neighbours are the neighbours that share a face and an edge with current transform unit node, respectively. FIG. 1 shows 7 parent-level nodes for each sub-node of transform unit node.

[0049] The attribute a_{up} of each sub-node is predicted depending on the distance between it and its parent-level node as follows.

$$[0050] \quad a_{up} = \sum \omega_k a_k / \sum \omega_k$$

[0051] a_k is the attribute of its one parent-level node and ω_k is weight depending on the distance. In G-PCC, $\omega_{parent} : \omega_{coplane} : \omega_{coline} = 4 : 2 : 1$.

[0052] For alternating current (AC) coefficient, the prediction residual will be signalled.

[0053] For direct current (DC) coefficient, the coefficients are inherited from the previous level, which means that the DC coefficient is signalled without prediction.

3.3 Attribute inter prediction in RAHT

[0054] The attribute inter prediction in RAHT is discussed in [6]. It is proposed to apply inter-prediction to DC and AC coefficients in RAHT. The same octree decomposition is performed on the current frame and the reference frame.

[0055] For the first 5 layers, the same scan of the octree is performed on the two frames. Before performing the octree scan backwards, a point-to-point matching process is performed to ensure that the node of the reference frame can establish a corresponding one-to-one relationship with the node of the current frame. For each point in the reference frame, it will be matched to one point in the current frame in a “upper matching” method. The Morton value of the matched point is the smallest Morton value greater than the Morton value of the current point.

[0056] For DC coefficients, the residual between the DC coefficient for the root node of the current frame and the DC coefficient for the root node of the reference frame is calculated as:

$$DC_{residual} = DC_{current} - DC_{reference}$$

[0057] The $DC_{residual}$ is signaled to the decoder in place of $DC_{current}$.

[0058] For each node in the first N layers, the average attribute of the node in the same octree location in the reference frame is calculated as $Attr_{predicted_inter}$ and the corresponding AC coefficients are calculated as $AC_{predicted_inter}$.

[0059] For AC coefficients, the prediction residual is signalled as:

$$AC_{residual} = AC_{current} - AC_{predicted}$$

$$AC_{predicted} = AC_{predicted_inter} ? AC_{predicted_inter} : AC_{predicted_intra}$$

[0060] If the $AC_{predicted_inter}$ is equal to zero, the $AC_{predicted_intra}$ is applied as the original transform domain prediction.

[0061] Another method in G-PCC is used to perform prediction in RAHT domain instead of sum of attributes space. Accordingly, in G-PCC there are two types: type 0 performs inter-prediction in RAHT domain and type 1 performs prediction in sum of attributes space.

4. Technical problems solved by disclosed technical solutions

[0062] An example design for point cloud attribute inter-prediction in region-adaptive hierarchical transform (RAHT) has the following problems:

[0063] First, in an example design with DC inheritance, when the parent has only one child, i.e., there are no AC coefficients for the current 2x2x2 RAHT node, the encoder and decoder only skip prediction but still perform transform and other operations accounting for unnecessary complexity.

[0064] Second, in the example design, when *all the AC coefficients* for the current 2x2x2 RAHT node are quantized to zero, the prediction is same as reconstruction. In such a case, it is not required to perform inverse transform to obtain the reconstruction.

5. A listing of solutions and embodiments

[0065] To solve the above problems and some other problems not mentioned, methods as summarized below are disclosed. The items should be considered as examples to explain the general concepts and should not be interpreted in a narrow way. Furthermore, these inventions can be applied individually or combined in any manner.

[0066] In the following description, the point cloud (PC) sample may refer to frame/sub-frame/picture/sub-picture/slice/sub-slice/tile and so on.

- 1) For problem 1, it is proposed to disable transform (both forward and inverse), quantization and other operations when there are no AC coefficients in the 2x2x2 RAHT node and DC is simply inherited.
 - a. In one example, only a subset of these operations could be disabled.
- 2) For problem 2, it is proposed to directly use the already available prediction in the sum of attribute space as reconstruction, instead of performing inverse transform of the residue added prediction. They yield identical results since all residuals are quantized to zero implying prediction and reconstruction are identical.
- 3) Whether to and/or how to apply a method disclosed above may be signaled from encoder to decoder in a bitstream/frame/tile/slice/octree/etc.
- 4) Whether to and/or how to apply the disclosed methods above may be dependent on coded information, such as dimensions, colour format, colour component, slice/picture type.

6. References

- [1] MPEG 3DG and Requirements, “Call for Proposals for Point Cloud Compression V2”, ISO/IEC JTC1/SC29 WG11 N16763.
- [2] ISO/IEC JTC 1/SC 29/WG 07, “Information technology — Coded Representation of Immersive Media — Part 5: Visual Volumetric Video-based Coding (V3C) and Video-based Point Cloud Compression (V-PCC)”, ISO/IEC 23090-5.

- [3] ISO/IEC JTC 1/SC 29/WG 11, “Information technology — MPEG-I (Coded Representation of Immersive Media) — Part 9: Geometry-based Point Cloud Compression”, ISO/IEC 23090-9:2020(E).
- [4] Ricardo L. De Queiroz and Philip A. Chou, “Compression of 3D Point Clouds Using a Region-Adaptive Hierarchical Transform”, IEEE Transactions on Image Processing.
- [5] S. Lasserre, D. Flynn, “On an improvement of RAHT to exploit attribute correlation”, ISO/IEC JTC1/SC29/WG11 M47378.
- [6] Y.-Z. Xu, W. Wang, K. Zhang, L. Zhang, [G-PCC][EE13.2 related][New proposal] Inter-Prediction for RAHT Attribute Coding, ISO/IEC JTC1/SC29/WG7 m61083, October 2022.

[0067] FIG. 2 is a block diagram showing an example video processing system 4000 in which various techniques disclosed herein may be implemented. Various implementations may include some or all of the components of the system 4000. The system 4000 may include input 4002 for receiving video content. The video content may be received in a raw or uncompressed format, e.g., 8 or 10 bit multi-component pixel values, or may be in a compressed or encoded format. The input 4002 may represent a network interface, a peripheral bus interface, or a storage interface. Examples of network interface include wired interfaces such as Ethernet, passive optical network (PON), etc. and wireless interfaces such as wireless fidelity (Wi-Fi) or cellular interfaces.

[0068] The system 4000 may include a coding component 4004 that may implement the various coding or encoding methods described in the present disclosure. The coding component 4004 may reduce the average bitrate of video from the input 4002 to the output of the coding component 4004 to produce a coded representation of the video. The coding techniques are therefore sometimes called video compression or video transcoding techniques. The output of the coding component 4004 may be either stored, or transmitted via a communication connected, as represented by the component 4006. The stored or communicated bitstream (or coded) representation of the video received at the input 4002 may be used by a component 4008 for generating pixel values or displayable video that is sent to a display interface 4010. The process of generating user-viewable video from the bitstream representation is sometimes called video decompression. Furthermore, while certain video processing operations are referred to as “coding” operations or tools, it will be appreciated that the coding tools or operations are used at an encoder and corresponding decoding tools or operations that reverse the results of the coding will be performed by a decoder.

[0069] Examples of a peripheral bus interface or a display interface may include universal serial bus (USB) or high definition multimedia interface (HDMI) or Displayport, and so on. Examples of storage interfaces include SATA (serial advanced technology attachment), PCI, IDE interface, and the like. The techniques described in the present disclosure may be embodied in various electronic devices such as mobile phones, laptops, smartphones or other devices that are capable of performing digital data processing and/or video display.

[0070] FIG. 3 is a block diagram of an example video processing apparatus 4100. The apparatus 4100 may be used to implement one or more of the methods described herein. The apparatus 4100 may be embodied in a smartphone, tablet, computer, Internet of Things (IoT) receiver, and so on. The apparatus 4100 may include one or more processors 4102, one or more memories 4104 and video processing circuitry 4106. The processor(s) 4102 may be configured to implement one or more methods described in the present disclosure. The memory (memories) 4104 may be used for storing data and code used for implementing the methods and techniques described herein. The video processing circuitry 4106 may be used to implement, in hardware circuitry, some techniques described in the present disclosure. In some embodiments, the video processing circuitry 4106 may be at least partly included in the processor 4102, e.g., a graphics co-processor.

[0071] FIG. 4 is a flowchart for an example method 4200 of video processing. In block 4242, the method 4200 includes determining to use an available prediction in a sum of attribute space as a reconstruction instead of performing an inverse transform of a residue to obtain a prediction when all alternating current (AC) coefficients in a 2x2x2 Region-Adaptive Hierarchical Transform (RAHT) node are quantized to zero. In block 4204, a conversion is performed between a visual media data and a bitstream based on the available prediction. The conversion of step 4204 may include encoding at an encoder or decoding at a decoder, depending on the example.

[0072] It should be noted that the method 4200 can be implemented in an apparatus for processing video data comprising a processor and a non-transitory memory with instructions thereon, such as video encoder 4400, video decoder 4500, and/or encoder 4600. In such a case, the instructions upon execution by the processor, cause the processor to perform the method 4200. Further, the method 4200 can be performed by a non-transitory computer readable medium comprising a computer program product for use by a video coding device. The computer program product comprises computer executable instructions stored on the non-transitory computer readable

medium such that when executed by a processor cause the video coding device to perform the method 4200.

[0073] FIG. 5 is a block diagram that illustrates an example video coding system 4300 that may utilize the techniques of this disclosure. The video coding system 4300 may include a source device 4310 and a destination device 4320. Source device 4310 generates encoded video data which may be referred to as a video encoding device. Destination device 4320 may decode the encoded video data generated by source device 4310 which may be referred to as a video decoding device.

[0074] Source device 4310 may include a video source 4312, a video encoder 4314, and an input/output (I/O) interface 4316. Video source 4312 may include a source such as a video capture device, an interface to receive video data from a video content provider, and/or a computer graphics system for generating video data, or a combination of such sources. The video data may comprise one or more pictures. Video encoder 4314 encodes the video data from video source 4312 to generate a bitstream. The bitstream may include a sequence of bits that form a coded representation of the video data. The bitstream may include coded pictures and associated data. The coded picture is a coded representation of a picture. The associated data may include sequence parameter sets, picture parameter sets, and other syntax structures. I/O interface 4316 may include a modulator/demodulator (modem) and/or a transmitter. The encoded video data may be transmitted directly to destination device 4320 via I/O interface 4316 through network 4330. The encoded video data may also be stored onto a storage medium/server 4340 for access by destination device 4320.

[0075] Destination device 4320 may include an I/O interface 4326, a video decoder 4324, and a display device 4322. I/O interface 4326 may include a receiver and/or a modem. I/O interface 4326 may acquire encoded video data from the source device 4310 or the storage medium/ server 4340. Video decoder 4324 may decode the encoded video data. Display device 4322 may display the decoded video data to a user. Display device 4322 may be integrated with the destination device 4320, or may be external to destination device 4320, which can be configured to interface with an external display device.

[0076] Video encoder 4314 and video decoder 4324 may operate according to a video compression standard, such as the High Efficiency Video Coding (HEVC) standard, Versatile Video Coding (VVC) standard and other current and/or further standards.

[0077] FIG. 6 is a block diagram illustrating an example of video encoder 4400, which may be video encoder 4314 in the system 4300 illustrated in FIG. 5. Video encoder 4400 may be configured

to perform any or all of the techniques of this disclosure. The video encoder 4400 includes a plurality of functional components. The techniques described in this disclosure may be shared among the various components of video encoder 4400. In some examples, a processor may be configured to perform any or all of the techniques described in this disclosure.

[0078] The functional components of video encoder 4400 may include a partition unit 4401, a prediction unit 4402 which may include a mode select unit 4403, a motion estimation unit 4404, a motion compensation unit 4405, an intra prediction unit 4406, a residual generation unit 4407, a transform processing unit 4408, a quantization unit 4409, an inverse quantization unit 4410, an inverse transform unit 4411, a reconstruction unit 4412, a buffer 4413, and an entropy encoding unit 4414.

[0079] In other examples, video encoder 4400 may include more, fewer, or different functional components. In an example, prediction unit 4402 may include an intra block copy (IBC) unit. The IBC unit may perform prediction in an IBC mode in which at least one reference picture is a picture where the current video block is located.

[0080] Furthermore, some components, such as motion estimation unit 4404 and motion compensation unit 4405 may be highly integrated, but are represented in the example of video encoder 4400 separately for purposes of explanation.

[0081] Partition unit 4401 may partition a picture into one or more video blocks. Video encoder 4400 and video decoder 4500 may support various video block sizes.

[0082] Mode select unit 4403 may select one of the coding modes, intra or inter, e.g., based on error results, and provide the resulting intra or inter coded block to a residual generation unit 4407 to generate residual block data and to a reconstruction unit 4412 to reconstruct the encoded block for use as a reference picture. In some examples, mode select unit 4403 may select a combination of intra and inter prediction (CIIP) mode in which the prediction is based on an inter prediction signal and an intra prediction signal. Mode select unit 4403 may also select a resolution for a motion vector (e.g., a sub-pixel or integer pixel precision) for the block in the case of inter prediction.

[0083] To perform inter prediction on a current video block, motion estimation unit 4404 may generate motion information for the current video block by comparing one or more reference frames from buffer 4413 to the current video block. Motion compensation unit 4405 may determine a predicted video block for the current video block based on the motion information and decoded samples of pictures from buffer 4413 other than the picture associated with the current video block.

[0084] Motion estimation unit 4404 and motion compensation unit 4405 may perform different operations for a current video block, for example, depending on whether the current video block is in an I slice, a P slice, or a B slice.

[0085] In some examples, motion estimation unit 4404 may perform uni-directional prediction for the current video block, and motion estimation unit 4404 may search reference pictures of list 0 or list 1 for a reference video block for the current video block. Motion estimation unit 4404 may then generate a reference index that indicates the reference picture in list 0 or list 1 that contains the reference video block and a motion vector that indicates a spatial displacement between the current video block and the reference video block. Motion estimation unit 4404 may output the reference index, a prediction direction indicator, and the motion vector as the motion information of the current video block. Motion compensation unit 4405 may generate the predicted video block of the current block based on the reference video block indicated by the motion information of the current video block.

[0086] In other examples, motion estimation unit 4404 may perform bi-directional prediction for the current video block, motion estimation unit 4404 may search the reference pictures in list 0 for a reference video block for the current video block and may also search the reference pictures in list 1 for another reference video block for the current video block. Motion estimation unit 4404 may then generate reference indexes that indicate the reference pictures in list 0 and list 1 containing the reference video blocks and motion vectors that indicate spatial displacements between the reference video blocks and the current video block. Motion estimation unit 4404 may output the reference indexes and the motion vectors of the current video block as the motion information of the current video block. Motion compensation unit 4405 may generate the predicted video block of the current video block based on the reference video blocks indicated by the motion information of the current video block.

[0087] In some examples, motion estimation unit 4404 may output a full set of motion information for decoding processing of a decoder. In some examples, motion estimation unit 4404 may not output a full set of motion information for the current video. Rather, motion estimation unit 4404 may signal the motion information of the current video block with reference to the motion information of another video block. For example, motion estimation unit 4404 may determine that the motion information of the current video block is sufficiently similar to the motion information of a neighboring video block.

[0088] In one example, motion estimation unit 4404 may indicate, in a syntax structure associated with the current video block, a value that indicates to the video decoder 4500 that the current video block has the same motion information as another video block.

[0089] In another example, motion estimation unit 4404 may identify, in a syntax structure associated with the current video block, another video block and a motion vector difference (MVD). The motion vector difference indicates a difference between the motion vector of the current video block and the motion vector of the indicated video block. The video decoder 4500 may use the motion vector of the indicated video block and the motion vector difference to determine the motion vector of the current video block.

[0090] As discussed above, video encoder 4400 may predictively signal the motion vector. Two examples of predictive signaling techniques that may be implemented by video encoder 4400 include advanced motion vector prediction (AMVP) and merge mode signaling.

[0091] Intra prediction unit 4406 may perform intra prediction on the current video block. When intra prediction unit 4406 performs intra prediction on the current video block, intra prediction unit 4406 may generate prediction data for the current video block based on decoded samples of other video blocks in the same picture. The prediction data for the current video block may include a predicted video block and various syntax elements.

[0092] Residual generation unit 4407 may generate residual data for the current video block by subtracting the predicted video block(s) of the current video block from the current video block. The residual data of the current video block may include residual video blocks that correspond to different sample components of the samples in the current video block.

[0093] In other examples, there may be no residual data for the current video block for the current video block, for example in a skip mode, and residual generation unit 4407 may not perform the subtracting operation.

[0094] Transform processing unit 4408 may generate one or more transform coefficient video blocks for the current video block by applying one or more transforms to a residual video block associated with the current video block.

[0095] After transform processing unit 4408 generates a transform coefficient video block associated with the current video block, quantization unit 4409 may quantize the transform coefficient video block associated with the current video block based on one or more quantization parameter (QP) values associated with the current video block.

[0096] Inverse quantization unit 4410 and inverse transform unit 4411 may apply inverse quantization and inverse transforms to the transform coefficient video block, respectively, to reconstruct a residual video block from the transform coefficient video block. Reconstruction unit 4412 may add the reconstructed residual video block to corresponding samples from one or more predicted video blocks generated by the prediction unit 4402 to produce a reconstructed video block associated with the current block for storage in the buffer 4413.

[0097] After reconstruction unit 4412 reconstructs the video block, the loop filtering operation may be performed to reduce video blocking artifacts in the video block.

[0098] Entropy encoding unit 4414 may receive data from other functional components of the video encoder 4400. When entropy encoding unit 4414 receives the data, entropy encoding unit 4414 may perform one or more entropy encoding operations to generate entropy encoded data and output a bitstream that includes the entropy encoded data.

[0099] FIG. 7 is a block diagram illustrating an example of video decoder 4500 which may be video decoder 4324 in the system 4300 illustrated in FIG. 5. The video decoder 4500 may be configured to perform any or all of the techniques of this disclosure. In the example shown, the video decoder 4500 includes a plurality of functional components. The techniques described in this disclosure may be shared among the various components of the video decoder 4500. In some examples, a processor may be configured to perform any or all of the techniques described in this disclosure.

[0100] In the example shown, video decoder 4500 includes an entropy decoding unit 4501, a motion compensation unit 4502, an intra prediction unit 4503, an inverse quantization unit 4504, an inverse transformation unit 4505, a reconstruction unit 4506, and a buffer 4507. Video decoder 4500 may, in some examples, perform a decoding pass generally reciprocal to the encoding pass described with respect to video encoder 4400.

[0101] Entropy decoding unit 4501 may retrieve an encoded bitstream. The encoded bitstream may include entropy coded video data (e.g., encoded blocks of video data). Entropy decoding unit 4501 may decode the entropy coded video data, and from the entropy decoded video data, motion compensation unit 4502 may determine motion information including motion vectors, motion vector precision, reference picture list indexes, and other motion information. Motion compensation unit 4502 may, for example, determine such information by performing the AMVP and merge mode.

[0102] Motion compensation unit 4502 may produce motion compensated blocks, possibly performing interpolation based on interpolation filters. Identifiers for interpolation filters to be used with sub-pixel precision may be included in the syntax elements.

[0103] Motion compensation unit 4502 may use interpolation filters as used by video encoder 4400 during encoding of the video block to calculate interpolated values for sub-integer pixels of a reference block. Motion compensation unit 4502 may determine the interpolation filters used by video encoder 4400 according to received syntax information and use the interpolation filters to produce predictive blocks.

[0104] Motion compensation unit 4502 may use some of the syntax information to determine sizes of blocks used to encode frame(s) and/or slice(s) of the encoded video sequence, partition information that describes how each macroblock of a picture of the encoded video sequence is partitioned, modes indicating how each partition is encoded, one or more reference frames (and reference frame lists) for each inter coded block, and other information to decode the encoded video sequence.

[0105] Intra prediction unit 4503 may use intra prediction modes for example received in the bitstream to form a prediction block from spatially adjacent blocks. Inverse quantization unit 4504 inverse quantizes, i.e., de-quantizes, the quantized video block coefficients provided in the bitstream and decoded by entropy decoding unit 4501. Inverse transform unit 4505 applies an inverse transform.

[0106] Reconstruction unit 4506 may sum the residual blocks with the corresponding prediction blocks generated by motion compensation unit 4502 or intra prediction unit 4503 to form decoded blocks. If desired, a deblocking filter may also be applied to filter the decoded blocks in order to remove blockiness artifacts. The decoded video blocks are then stored in buffer 4507, which provides reference blocks for subsequent motion compensation/intra prediction and also produces decoded video for presentation on a display device.

[0107] FIG. 8 is a schematic diagram of an example encoder 4600. The encoder 4600 is suitable for implementing the techniques of VVC. The encoder 4600 includes three in-loop filters, namely a deblocking filter (DF) 4602, a sample adaptive offset (SAO) 4604, and an adaptive loop filter (ALF) 4606. Unlike the DF 4602, which uses predefined filters, the SAO 4604 and the ALF 4606 utilize the original samples of the current picture to reduce the mean square errors between the original samples and the reconstructed samples by adding an offset and by applying a finite impulse

response (FIR) filter, respectively, with coded side information signaling the offsets and filter coefficients. The ALF 4606 is located at the last processing stage of each picture and can be regarded as a tool trying to catch and fix artifacts created by the previous stages.

[0108] The encoder 4600 further includes an intra prediction component 4608 and a motion estimation/compensation (ME/MC) component 4610 configured to receive input video. The intra prediction component 4608 is configured to perform intra prediction, while the ME/MC component 4610 is configured to utilize reference pictures obtained from a reference picture buffer 4612 to perform inter prediction. Residual blocks from inter prediction or intra prediction are fed into a transform (T) component 4614 and a quantization (Q) component 4616 to generate quantized residual transform coefficients, which are fed into an entropy coding component 4618. The entropy coding component 4618 entropy codes the prediction results and the quantized transform coefficients and transmits the same toward a video decoder (not shown). Quantization components output from the quantization component 4616 may be fed into an inverse quantization (IQ) components 4620, an inverse transform component 4622, and a reconstruction (REC) component 4624. The REC component 4624 is able to output images to the DF 4602, the SAO 4604, and the ALF 4606 for filtering prior to those images being stored in the reference picture buffer 4612.

[0109] A listing of solutions preferred by some examples is provided next.

[0110] 1. A method for processing media data comprising: determining to disable at least one Region-Adaptive Hierarchical Transform (RAHT) operation when there are no alternating current (AC) coefficients in a 2x2x2 RAHT node and direct current (DC) values are inherited; and performing a conversion between a visual media data and a bitstream based on RAHT.

[0111] 2. The method of solution 1, wherein a forward transform, an inverse transform, and all other RAHT operations are disabled.

[0112] 3. The method of any of solutions 1-2, wherein a subset of RAHT operations are disabled.

[0113] 4. The method of any of solutions 1-3, wherein available prediction is directly used in a sum of attribute space as reconstruction.

[0114] 5. The method of any of solutions 1-4, wherein inverse transform of the residue added prediction is not performed.

[0115] 6. The method of any of solutions 1-5, wherein all residuals are quantized to zero and prediction is identical to reconstruction.

[0116] 7. The method of any of solutions 1-6, wherein usage is signaled from encoder to decoder in a bitstream/frame/tile/slice/octree/etc.

[0117] 8. The method of any of solutions 1-7, wherein usage is dependent on coded information including dimensions, color format, color component, or slice/picture type.

[0118] 9. An apparatus for processing video data comprising: a processor; and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to perform the method of any of solutions 1-8.

[0119] 10. A non-transitory computer readable medium comprising a computer program product for use by a video coding device, the computer program product comprising computer executable instructions stored on the non-transitory computer readable medium such that when executed by a processor cause the video coding device to perform the method of any of solutions 1-8.

[0120] 11. A non-transitory computer-readable recording medium storing a bitstream of a video which is generated by a method performed by a video processing apparatus, wherein the method comprises: determining to disable at least one Region-Adaptive Hierarchical Transform (RAHT) operation when there are no alternating current (AC) coefficients in a 2x2x2 RAHT node and direct current (DC) values are inherited; and generating a bitstream based on the determining.

[0121] 12. A method for storing bitstream of a video comprising: determining to disable at least one Region-Adaptive Hierarchical Transform (RAHT) operation when there are no alternating current (AC) coefficients in a 2x2x2 RAHT node and direct current (DC) values are inherited; generating a bitstream based on the determining; and storing the bitstream in a non-transitory computer-readable recording medium.

[0122] 13. A method, apparatus or system described in the present disclosure.

[0123] A listing of further example solutions is provided next.

[0124] 1. A method for processing media data, comprising: determining to use an available prediction in a sum of attribute space as a reconstruction instead of performing an inverse transform of a residue to obtain a prediction when all alternating current (AC) coefficients in a 2x2x2 Region-Adaptive Hierarchical Transform (RAHT) node are quantized to zero; performing a conversion between a visual media data and a bitstream based on the available prediction.

[0125] 2. The method of solution 1, wherein the available prediction and the reconstruction are identical.

- [0126] 3. The method of any of solutions 1-2, wherein the inverse transform is not performed to obtain the reconstruction.
- [0127] 4. The method of any of solutions 1-3, further comprising determining to disable one or more RAHT operations and to inherit direct current (DC) values when there are no AC coefficients in the 2x2x2 RAHT node.
- [0128] 5. The method of solution 4, wherein the RAHT operations comprise a forward transform.
- [0129] 6. The method of solution 4, wherein the RAHT operations comprise an inverse transform.
- [0130] 7. The method of solution 4, wherein the RAHT operations comprise quantization.
- [0131] 8. The method of solution 4, wherein all of the RAHT operations are disabled when there are no AC coefficients in the 2x2x2 RAHT node.
- [0132] 9. The method of solution 4, wherein a subset of the RAHT operations are disabled.
- [0133] 10. The method of any of solutions 1-9, wherein whether to and/or how to apply the one or more of the methods is signalled in a frame, tile, slice, octree, or the bitstream.
- [0134] 11. The method of any of solutions 1-10, wherein whether to and/or how to apply the one or more of the methods is dependent on coded information, and wherein the coded information comprises one or more dimensions, a color format, a color component, a slice type, or a picture type.
- [0135] 12. The method of any of solutions 1-11, wherein the conversion includes encoding the media data into the bitstream.
- [0136] 13. The method of any of solutions 1-11, wherein the conversion includes decoding the media data from the bitstream.
- [0137] 14. An apparatus for processing video data comprising: a processor; and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to perform the method of any of solutions 1-13.
- [0138] 15. A non-transitory computer readable medium comprising a computer program product for use by a video coding device, the computer program product comprising computer executable instructions stored on the non-transitory computer readable medium such that when executed by a processor cause the video coding device to perform the method of any of solutions 1-13.

[0139] 16. A non-transitory computer-readable recording medium storing a bitstream of a video which is generated by a method performed by a video processing apparatus, wherein the method comprises: determining to use an available prediction in a sum of attribute space as a reconstruction instead of performing an inverse transform of a residue to obtain a prediction when all alternating current (AC) coefficients in a 2x2x2 Region-Adaptive Hierarchical Transform (RAHT) node are quantized to zero; and generating the bitstream based on the available prediction.

[0140] 17. A method for storing bitstream of a video comprising: determining to use an available prediction in a sum of attribute space as a reconstruction instead of performing an inverse transform of a residue to obtain a prediction when all alternating current (AC) coefficients in a 2x2x2 Region-Adaptive Hierarchical Transform (RAHT) node are quantized to zero; generating the bitstream based on the available prediction; and storing the bitstream in a non-transitory computer-readable recording medium.

[0141] 18. A method, apparatus, or system described in the present disclosure.

[0142] In the solutions described herein, an encoder may conform to the format rule by producing a coded representation according to the format rule. In the solutions described herein, a decoder may use the format rule to parse syntax elements in the coded representation with the knowledge of presence and absence of syntax elements according to the format rule to produce decoded video.

[0143] In the present disclosure, the term “video processing” may refer to video encoding, video decoding, video compression or video decompression. For example, video compression algorithms may be applied during conversion from pixel representation of a video to a corresponding bitstream representation or vice versa. The bitstream representation of a current video block may, for example, correspond to bits that are either co-located or spread in different places within the bitstream, as is defined by the syntax. For example, a macroblock may be encoded in terms of transformed and coded error residual values and also using bits in headers and other fields in the bitstream. Furthermore, during conversion, a decoder may parse a bitstream with the knowledge that some fields may be present, or absent, based on the determination, as is described in the above solutions. Similarly, an encoder may determine that certain syntax fields are or are not to be included and generate the coded representation accordingly by including or excluding the syntax fields from the coded representation.

[0144] The disclosed and other solutions, examples, embodiments, modules and the functional operations described in this disclosure can be implemented in digital electronic circuitry, or in computer software, firmware, or hardware, including the structures disclosed in this disclosure and their structural equivalents, or in combinations of one or more of them. The disclosed and other embodiments can be implemented as one or more computer program products, i.e., one or more modules of computer program instructions encoded on a computer readable medium for execution by, or to control the operation of, data processing apparatus. The computer readable medium can be a machine-readable storage device, a machine-readable storage substrate, a memory device, a composition of matter effecting a machine-readable propagated signal, or a combination of one or more them. The term “data processing apparatus” encompasses all apparatus, devices, and machines for processing data, including by way of example a programmable processor, a computer, or multiple processors or computers. The apparatus can include, in addition to hardware, code that creates an execution environment for the computer program in question, e.g., code that constitutes processor firmware, a protocol stack, a database management system, an operating system, or a combination of one or more of them. A propagated signal is an artificially generated signal, e.g., a machine-generated electrical, optical, or electromagnetic signal, that is generated to encode information for transmission to suitable receiver apparatus.

[0145] A computer program (also known as a program, software, software application, script, or code) can be written in any form of programming language, including compiled or interpreted languages, and it can be deployed in any form, including as a stand-alone program or as a module, component, subroutine, or other unit suitable for use in a computing environment. A computer program does not necessarily correspond to a file in a file system. A program can be stored in a portion of a file that holds other programs or data (e.g., one or more scripts stored in a markup language document), in a single file dedicated to the program in question, or in multiple coordinated files (e.g., files that store one or more modules, sub programs, or portions of code). A computer program can be deployed to be executed on one computer or on multiple computers that are located at one site or distributed across multiple sites and interconnected by a communication network.

[0146] The processes and logic flows described in this disclosure can be performed by one or more programmable processors executing one or more computer programs to perform functions by operating on input data and generating output. The processes and logic flows can also be performed

by, and apparatus can also be implemented as, special purpose logic circuitry, e.g., a field programmable gate array (FPGA) or an application specific integrated circuit (ASIC).

[0147] Processors suitable for the execution of a computer program include, by way of example, both general and special purpose microprocessors, and any one or more processors of any kind of digital computer. Generally, a processor will receive instructions and data from a read only memory or a random-access memory or both. The essential elements of a computer are a processor for performing instructions and one or more memory devices for storing instructions and data. Generally, a computer will also include, or be operatively coupled to receive data from or transfer data to, or both, one or more mass storage devices for storing data, e.g., magnetic, magneto optical disks, or optical disks. However, a computer need not have such devices. Computer readable media suitable for storing computer program instructions and data include all forms of non-volatile memory, media and memory devices, including by way of example semiconductor memory devices, e.g., erasable programmable read-only memory (EPROM), electrically erasable programmable read-only memory (EEPROM), and flash memory devices; magnetic disks, e.g., internal hard disks or removable disks; magneto optical disks; and compact disc read-only memory (CD ROM) and Digital versatile disc-read only memory (DVD-ROM) disks. The processor and the memory can be supplemented by, or incorporated in, special purpose logic circuitry.

[0148] While the present disclosure contains many specifics, these should not be construed as limitations on the scope of any subject matter or of what may be claimed, but rather as descriptions of features that may be specific to particular embodiments of particular techniques. Certain features that are described in the present disclosure in the context of separate embodiments can also be implemented in combination in a single embodiment. Conversely, various features that are described in the context of a single embodiment can also be implemented in multiple embodiments separately or in any suitable subcombination. Moreover, although features may be described above as acting in certain combinations and even initially claimed as such, one or more features from a claimed combination can in some cases be excised from the combination, and the claimed combination may be directed to a subcombination or variation of a subcombination.

[0149] Similarly, while operations are depicted in the drawings in a particular order, this should not be understood as requiring that such operations be performed in the particular order shown or in sequential order, or that all illustrated operations be performed, to achieve desirable results.

Moreover, the separation of various system components in the embodiments described in the present disclosure should not be understood as requiring such separation in all embodiments.

[0150] Only a few implementations and examples are described and other implementations, enhancements and variations can be made based on what is described and illustrated in the present disclosure.

[0151] A first component is directly coupled to a second component when there are no intervening components, except for a line, a trace, or another medium between the first component and the second component. The first component is indirectly coupled to the second component when there are intervening components other than a line, a trace, or another medium between the first component and the second component. The term “coupled” and its variants include both directly coupled and indirectly coupled. The use of the term “about” means a range including $\pm 10\%$ of the subsequent number unless otherwise stated.

[0152] While several embodiments have been provided in the present disclosure, it should be understood that the disclosed systems and methods might be embodied in many other specific forms without departing from the spirit or scope of the present disclosure. The present examples are to be considered as illustrative and not restrictive, and the intention is not to be limited to the details given herein. For example, the various elements or components may be combined or integrated in another system or certain features may be omitted, or not implemented.

[0153] In addition, techniques, systems, subsystems, and methods described and illustrated in the various embodiments as discrete or separate may be combined or integrated with other systems, modules, techniques, or methods without departing from the scope of the present disclosure. Other items shown or discussed as coupled may be directly connected or may be indirectly coupled or communicating through some interface, device, or intermediate component whether electrically, mechanically, or otherwise. Other examples of changes, substitutions, and alterations are ascertainable by one skilled in the art and could be made without departing from the spirit and scope disclosed herein.

CLAIMS

What is claimed is:

1. A method for processing media data, comprising:
determining to use an available prediction in a sum of attribute space as a reconstruction instead of performing an inverse transform of a residue to obtain a prediction when all alternating current (AC) coefficients in a 2x2x2 Region-Adaptive Hierarchical Transform (RAHT) node are quantized to zero;
performing a conversion between a visual media data and a bitstream based on the available prediction.
2. The method of claim 1, wherein the available prediction and the reconstruction are identical.
3. The method of any of claims 1-2, wherein the inverse transform is not performed to obtain the reconstruction.
4. The method of any of claims 1-3, further comprising determining to disable one or more RAHT operations and to inherit direct current (DC) values when there are no AC coefficients in the 2x2x2 RAHT node.
5. The method of claim 4, wherein the RAHT operations comprise a forward transform.
6. The method of claim 4, wherein the RAHT operations comprise an inverse transform.
7. The method of claim 4, wherein the RAHT operations comprise quantization.
8. The method of claim 4, wherein all of the RAHT operations are disabled when there are no AC coefficients in the 2x2x2 RAHT node.
9. The method of claim 4, wherein a subset of the RAHT operations are disabled.

10. The method of any of claims 1-9, wherein whether to and/or how to apply the one or more of the methods is signalled in a frame, tile, slice, octree, or the bitstream.
11. The method of any of claims 1-10, wherein whether to and/or how to apply the one or more of the methods is dependent on coded information, and wherein the coded information comprises one or more dimensions, a color format, a color component, a slice type, or a picture type.
12. The method of any of claims 1-11, wherein the conversion includes encoding the media data into the bitstream.
13. The method of any of claims 1-11, wherein the conversion includes decoding the media data from the bitstream.
14. An apparatus for processing video data comprising: a processor; and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to perform the method of any of claims 1-13.
15. A non-transitory computer readable medium comprising a computer program product for use by a video coding device, the computer program product comprising computer executable instructions stored on the non-transitory computer readable medium such that when executed by a processor cause the video coding device to perform the method of any of claims 1-13.
16. A non-transitory computer-readable recording medium storing a bitstream of a video which is generated by a method performed by a video processing apparatus, wherein the method comprises:
 - determining to use an available prediction in a sum of attribute space as a reconstruction instead of performing an inverse transform of a residue to obtain a prediction when all alternating current (AC) coefficients in a 2x2x2 Region-Adaptive Hierarchical Transform (RAHT) node are quantized to zero; and
 - generating the bitstream based on the available prediction.

17. A method for storing bitstream of a video comprising:
 - determining to use an available prediction in a sum of attribute space as a reconstruction instead of performing an inverse transform of a residue to obtain a prediction when all alternating current (AC) coefficients in a 2x2x2 Region-Adaptive Hierarchical Transform (RAHT) node are quantized to zero;
 - generating the bitstream based on the available prediction; and
 - storing the bitstream in a non-transitory computer-readable recording medium.
18. A method, apparatus, or system described in the present disclosure.

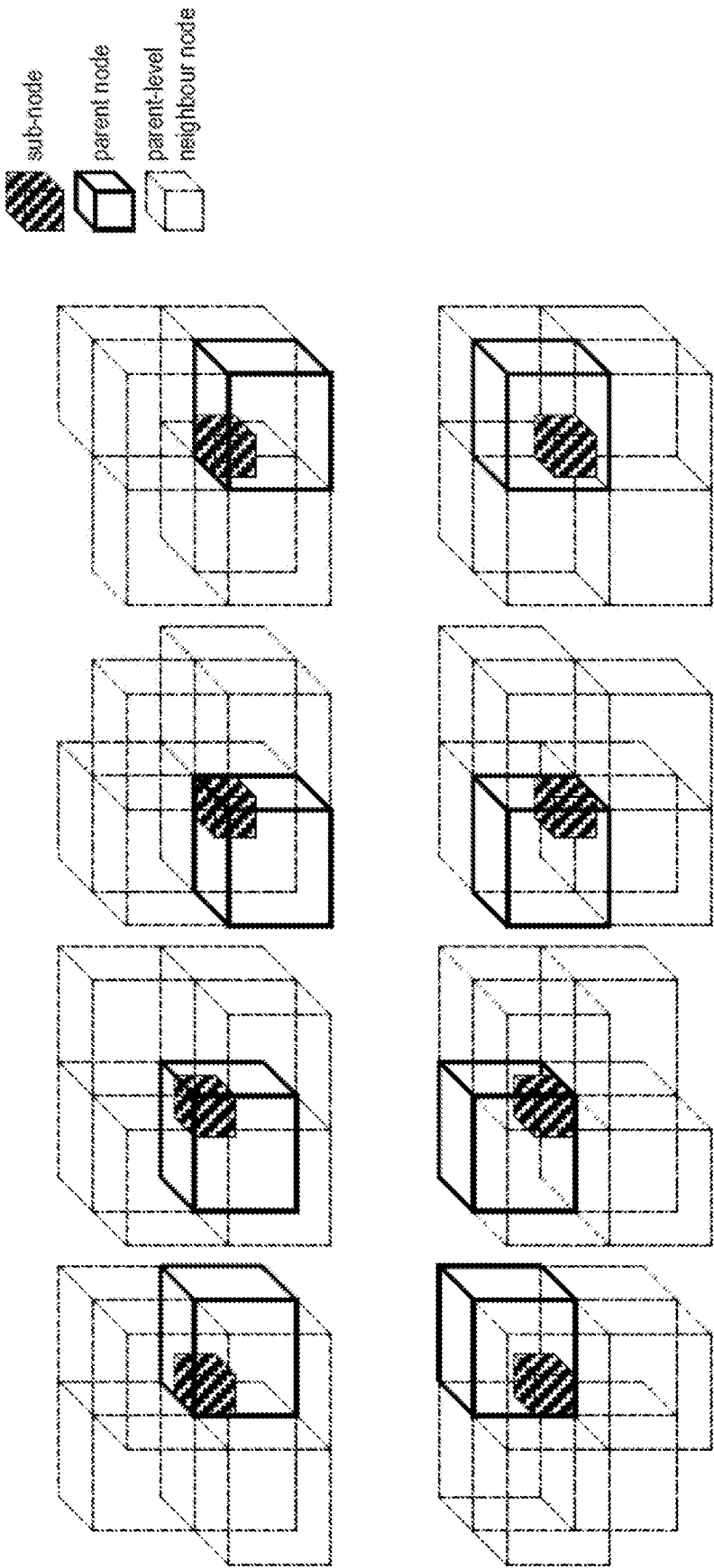


FIG. 1

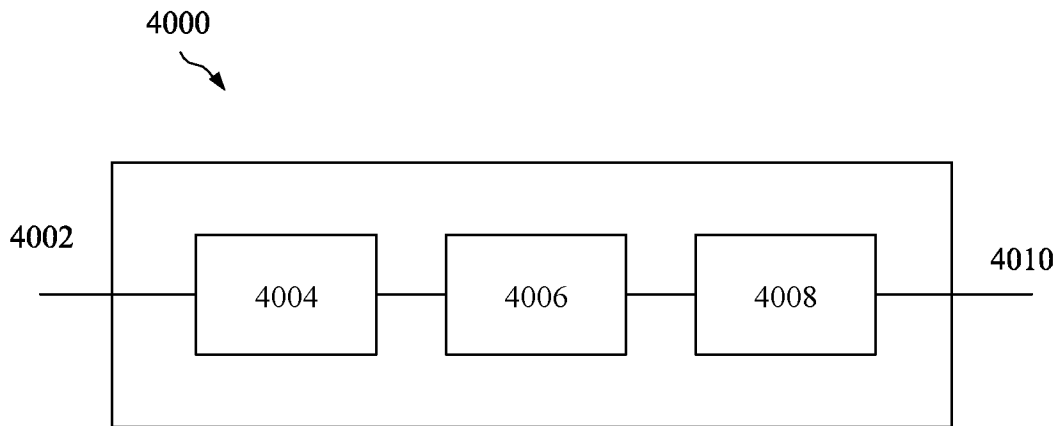


FIG. 2

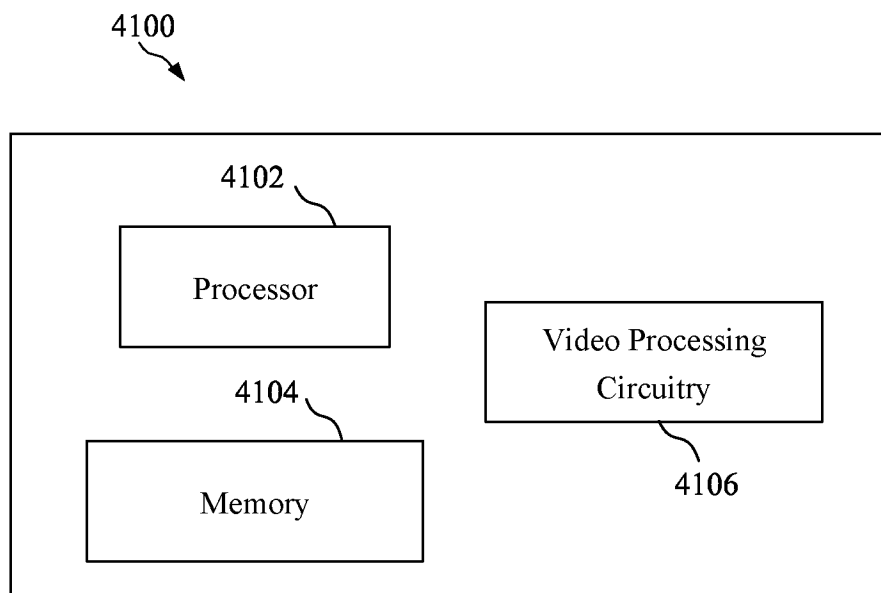


FIG. 3

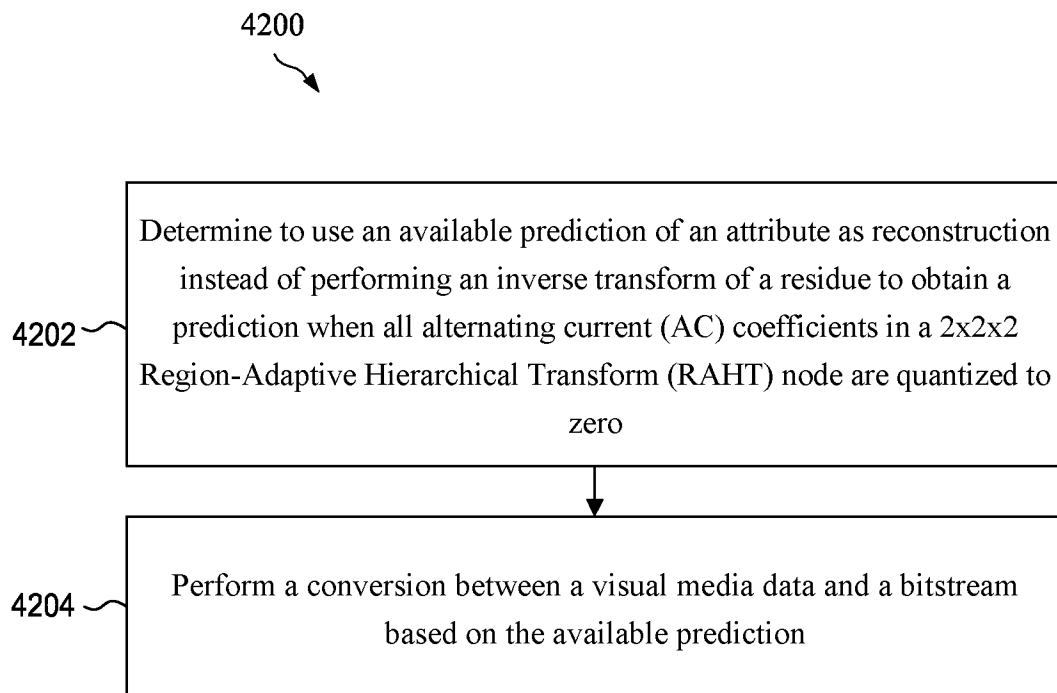


FIG. 4

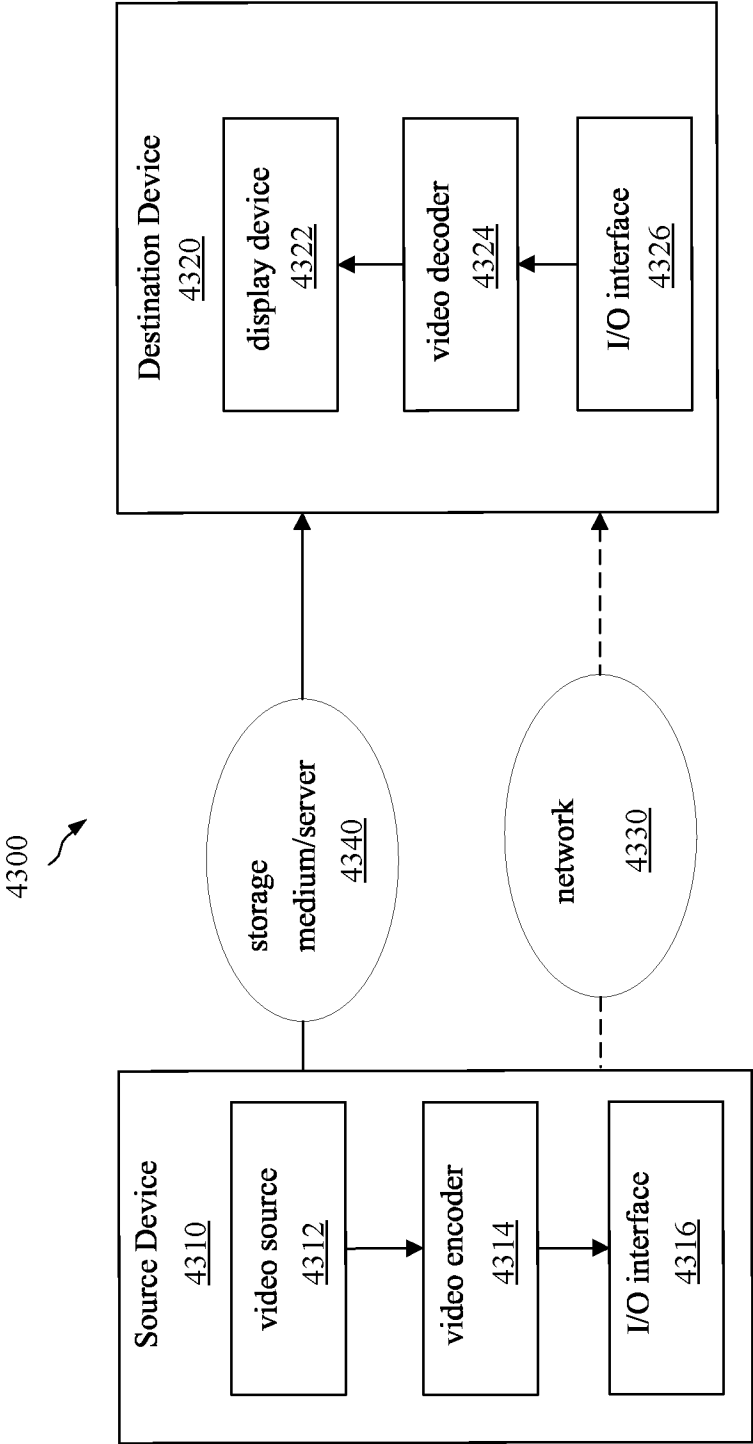


FIG. 5

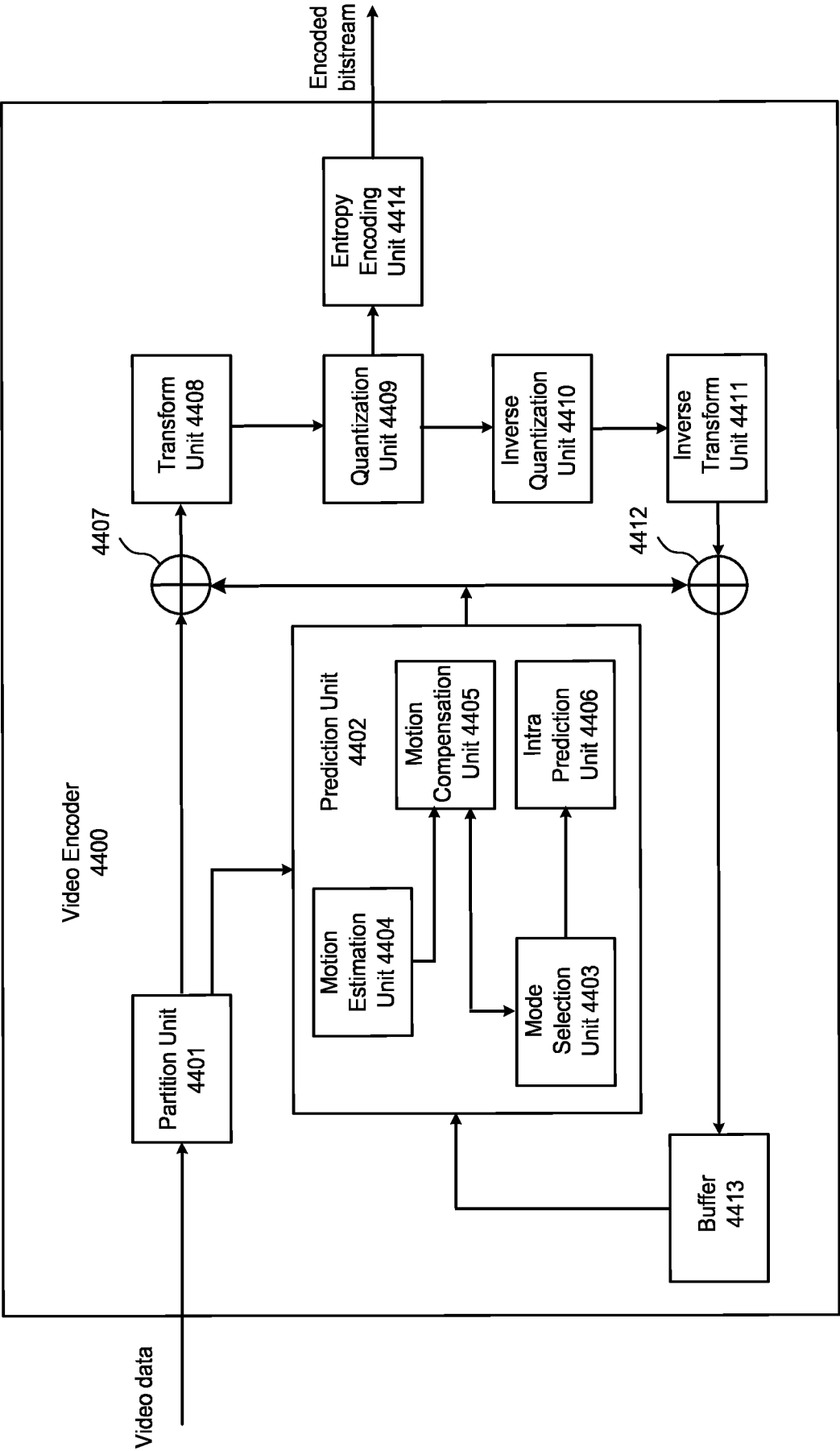


FIG. 6

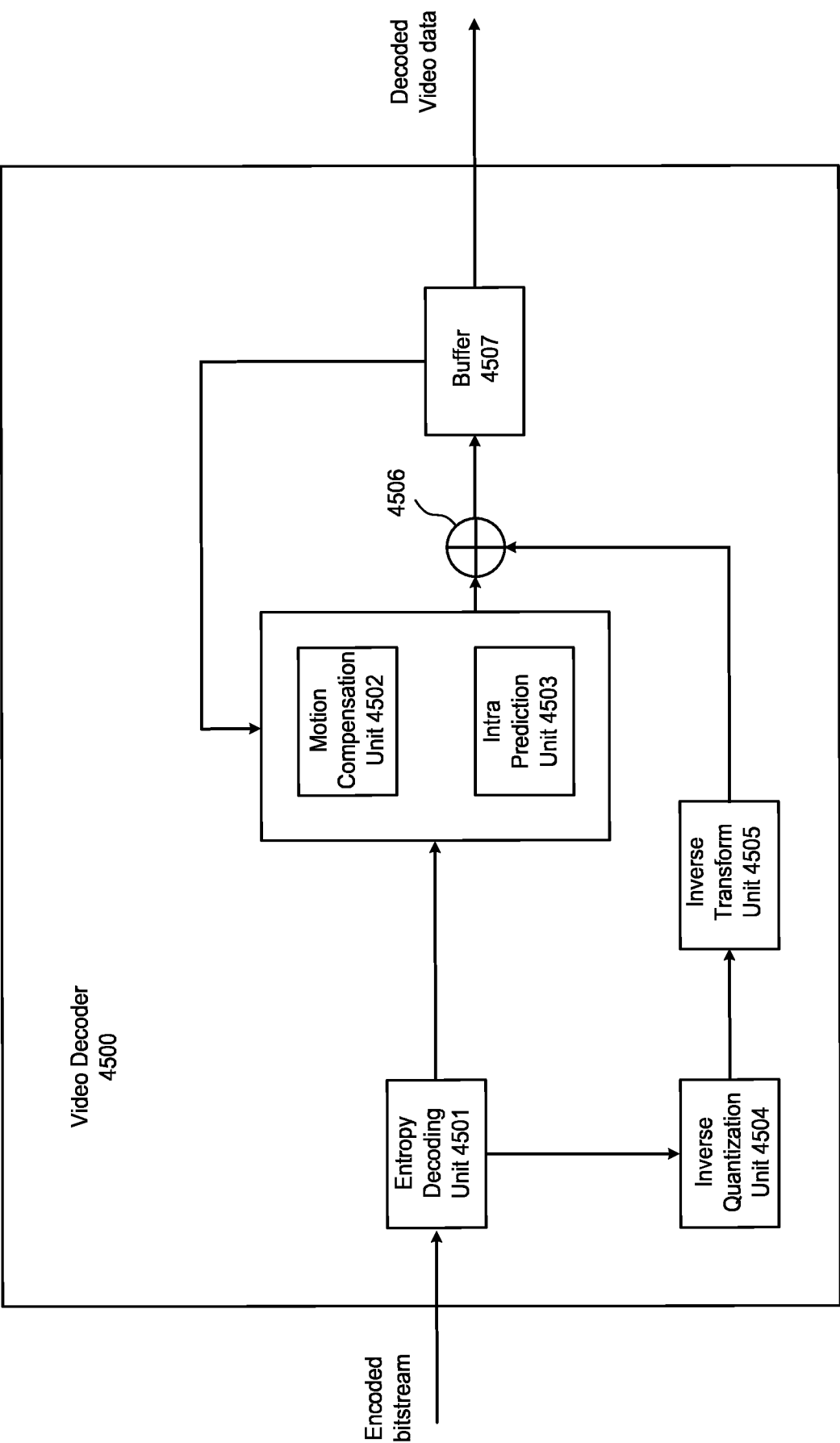


FIG. 7

7/7

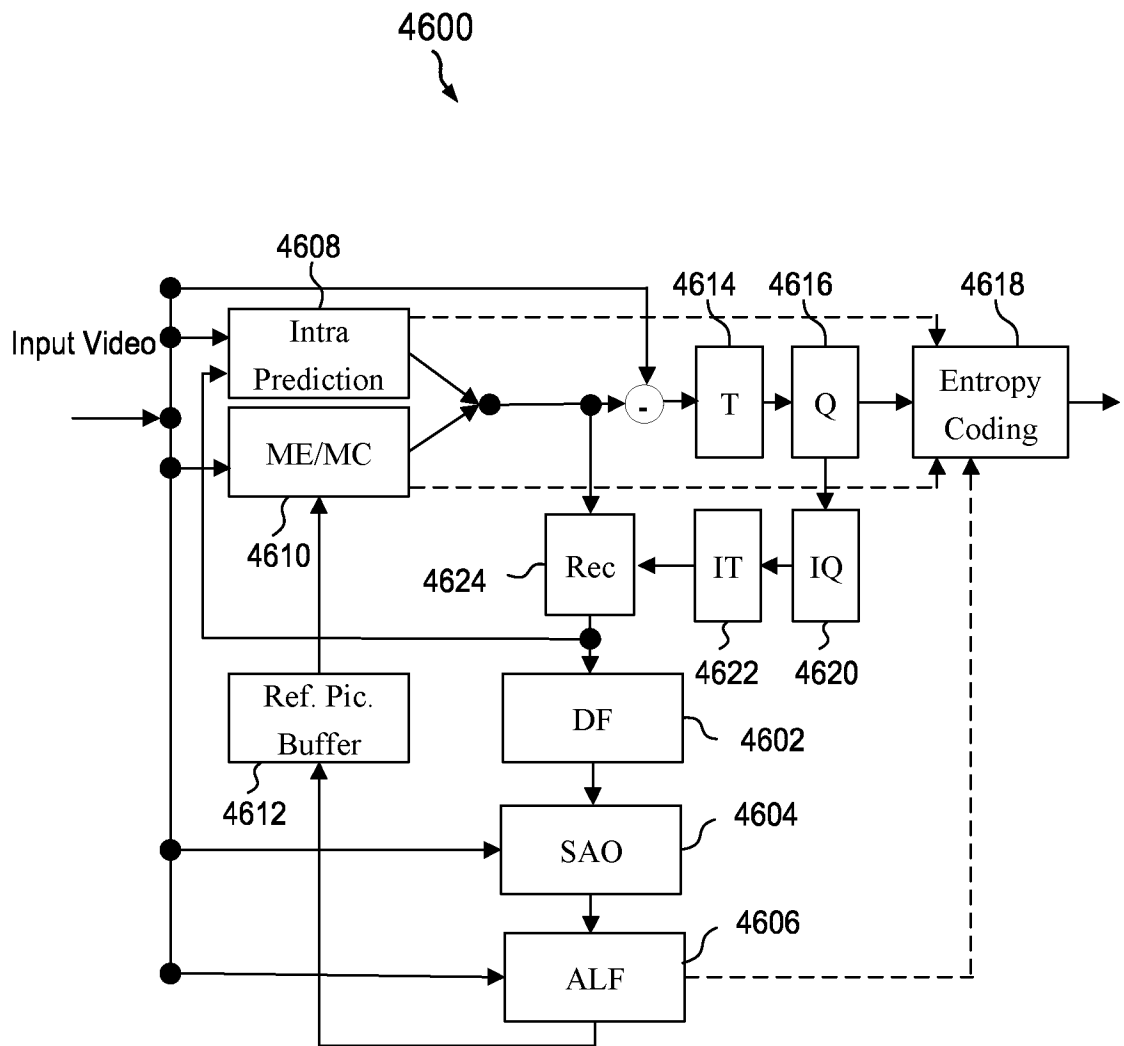


FIG. 8