Innovation, Sciences et Développement économique Canada
Office de la Propriété Intellectuelle du Canada

Innovation, Science and Economic Development Canada
Canadian Intellectual Property Office

CA 2925909 C 2022/03/29

(11)(21) 2 925 909

(12) BREVET CANADIEN
CANADIAN PATENT

(13) C

(86) Date de dépôt PCT/PCT Filing Date: 2014/10/14
(87) Date publication PCT/PCT Publication Date: 2015/04/23
(45) Date de délivrance/Issue Date: 2022/03/29
(85) Entrée phase nationale/National Entry: 2016/03/29
(86) N° demande PCT/PCT Application No.: US 2014/060413
(87) N° publication PCT/PCT Publication No.: 2015/057656
(30) Priorités/Priorities: 2013/10/14 (US61/890,843); 2014/10/10 (US14/512,177)

(51) Cl.Int./Int.Cl. *H04N 19/30* (2014.01), *H04N 19/42* (2014.01)

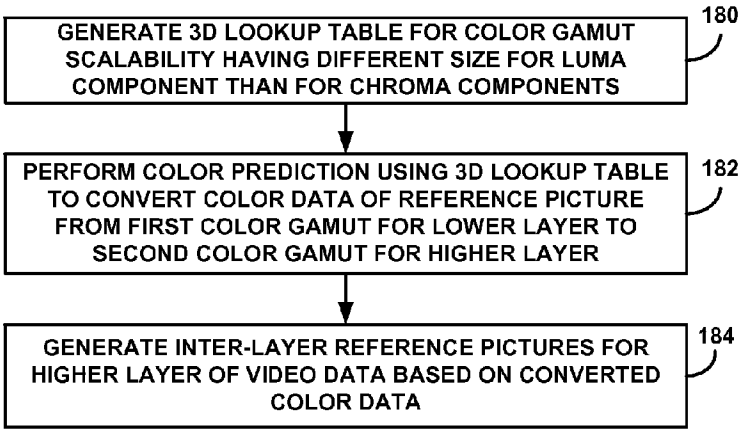
(72) Inventeurs/Inventors:
LI, XIANG, US;
RAPAKA, KRISHNAKANTH, US;
CHEN, JIANLE, US;
KARCZEWICZ, MARTA, US

(73) Propriétaire/Owner:
QUALCOMM INCORPORATED, US

(74) Agent: SMART & BIGGAR LLP

(54) Titre : EXTENSIBILITE DE GAMME DE COULEURS BASEE SUR UNE TABLE DE CONVERSION TRIDIMENSIONNELLE EN CODAGE VIDEO MULTICOUCHE

(54) Title: THREE-DIMENSIONAL LOOKUP TABLE BASED COLOR GAMUT SCALABILITY IN MULTI-LAYER VIDEO CODING



(57) Abrégé/Abstract:
Techniques are described for 3D lookup table based color gamut scalability in multi-layer video coding. A lower layer of video data may include color data in a first color gamut and a higher layer of the video data may include color data in a second color gamut. To generate inter-layer reference pictures, a video coder performs color prediction to convert the color data of a reference picture in the first color gamut to the second color gamut. The video coder may perform color prediction using a 3D lookup table. According to the techniques, the video coder generates at least one 3D lookup table having a size that is different for a luma component than for a first chroma component and a second chroma component. In this way, the total size of the table may be reduced while maintaining good coding performance.

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
23 April 2015 (23.04.2015)

WIPO | PCT

(10) International Publication Number
WO 2015/057656 A1

(51) International Patent Classification:
H04N 19/30 (2014.01) *H04N 19/42* (2014.01)

(21) International Application Number:
PCT/US2014/060413

(22) International Filing Date:
14 October 2014 (14.10.2014)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/890,843 14 October 2013 (14.10.2013) US
14/512,177 10 October 2014 (10.10.2014) US

(71) Applicant: **QUALCOMM INCORPORATED** [US/US]; ATTN: International IP Administration, 5775 Morehouse Drive, San Diego, California 92121-1714 (US).

(72) Inventors: **LI, Xiang**; 5775 Morehouse Drive, San Diego, California 92121-1714 (US). **RAPAKA, Krishnakanth**; 5775 Morehouse Drive, San Diego, California 92121-1714 (US). **CHEN, Jianle**; 5775 Morehouse Drive, San Diego, California 92121-1714 (US). **KARCZEWICZ, Marta**; 5775 Morehouse Drive, San Diego, California 92121-1714 (US).

(74) Agent: **SHUKLA, Darcy L.G.**; Shumaker & Sieffert, P.A., 1625 Radio Drive, Suite 300, Woodbury, Minnesota 55125 (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, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

Published:

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

(54) Title: THREE-DIMENSIONAL LOOKUP TABLE BASED COLOR GAMUT SCALABILITY IN MULTI-LAYER VIDEO CODING

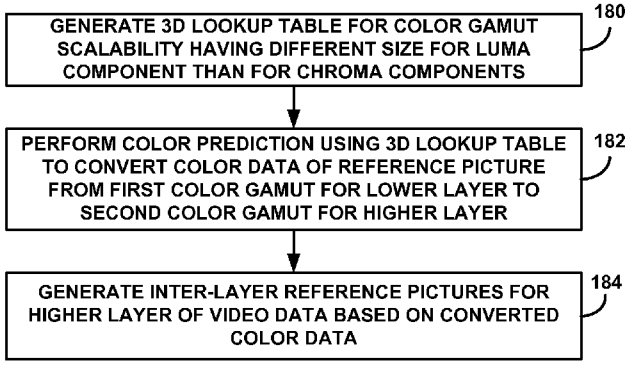


FIG. 13

(57) Abstract: Techniques are described for 3D look-up table based color gamut scalability in multi-layer video coding. A lower layer of video data may include color data in a first color gamut and a higher layer of the video data may include color data in a second color gamut. To generate inter-layer reference pictures, a video coder performs color prediction to convert the color data of a reference picture in the first color gamut to the second color gamut. The video coder may perform color prediction using a 3D lookup table. According to the techniques, the video coder generates at least one 3D lookup table having a size that is different for a luma component than for a first chroma component and a second chroma component. In this way, the total size of the table may be reduced while maintaining good coding performance.

WO 2015/057656 A1

THREE-DIMENSIONAL LOOKUP TABLE BASED COLOR GAMUT SCALABILITY IN MULTI-LAYER VIDEO CODING

[0001]

TECHNICAL FIELD

[0002] This disclosure relates to video coding.

BACKGROUND

[0003] Digital video capabilities can be incorporated into a wide range of devices, including digital televisions, digital direct broadcast systems, wireless broadcast systems, personal digital assistants (PDAs), laptop or desktop computers, tablet computers, e-book readers, digital cameras, digital recording devices, digital media players, video gaming devices, video game consoles, cellular or satellite radio telephones, so-called "smart phones," video teleconferencing devices, video streaming devices, and the like. Digital video devices implement video coding techniques, such as those described in the standards defined by MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Part 10, Advanced Video Coding (AVC), the High Efficiency Video Coding (HEVC) standard, and extensions of such standards. The video devices may transmit, receive, encode, decode, and/or store digital video information more efficiently by implementing such video coding techniques.

[0004] Video coding techniques include spatial (intra-picture) prediction and/or temporal (inter-picture) prediction to reduce or remove redundancy inherent in video sequences. For block-based video coding, a video slice (e.g., a video frame or a portion of a video frame) may be partitioned into video blocks, which may also be referred to as treeblocks, coding units (CUs) and/or coding nodes. Video blocks in an intra-coded (I) slice of a picture are encoded using spatial prediction with respect to reference samples in neighboring blocks in the same picture. Video blocks in an inter-coded (P or B) slice of a picture may use spatial prediction with respect to reference samples in neighboring blocks in the same picture or temporal prediction with respect to reference samples in

Date Reçue/Date Received 2021-03-03

other reference pictures. Pictures may be referred to as frames, and reference pictures may be referred to as reference frames.

[0005] Spatial or temporal prediction results in a predictive block for a block to be coded. Residual data represents pixel differences between the original block to be coded and the predictive block. An inter-coded block is encoded according to a motion vector that points to a block of reference samples forming the predictive block, and the residual data indicating the difference between the coded block and the predictive block. An intra-coded block is encoded according to an intra-coding mode and the residual data. For further compression, the residual data may be transformed from the pixel domain to a transform domain, resulting in residual transform coefficients, which then may be quantized. The quantized transform coefficients, initially arranged in a two-dimensional array, may be scanned in order to produce a one-dimensional vector of transform coefficients, and entropy coding may be applied to achieve even more compression.

SUMMARY

[0006] In general, this disclosure describes techniques for three-dimensional (3D) lookup table based color gamut scalability in multi-layer video coding. As an example, the multi-layer video coding may be in accordance the High Efficiency Video Coding (HEVC) standard, including any of a scalable video coding extension, a multiview video coding extension, and a 3D video coding (i.e., multiview video coding plus depth) extension, or other multi-layer video coding standards. The techniques for color gamut scalability may be used by video encoders and/or video decoders to generate inter-layer reference pictures when a color gamut for a lower layer of video data is different than a color gamut for a higher layer of the video data.

[0007] For example, a video encoder and/or video decoder may first perform color prediction using a 3D lookup table for color gamut scalability to convert the color data of a reference picture for the lower layer to the color gamut for the higher layer, and then generate inter-layer reference pictures based on the converted color data. According to the techniques described in this disclosure, the video encoder and/or video decoder generates at least one 3D lookup table having a size that is different for the luma component than for the first chroma component and the second chroma component. The video encoder and/or video decoder may generate this asymmetric 3D lookup table using a different number of segments for the luma dimension of the 3D

lookup table. In this way, the total size of the table may be reduced while maintaining good coding performance.

[0008] In one example, this disclosure is directed to a method of processing multi-layer video data, the method comprising generating at least one 3D lookup table for color gamut scalability, wherein the at least one 3D lookup table has a size that is different for a luma component than for each of a first chroma component and a second chroma component; performing color prediction using the at least one 3D lookup table to convert color data of a reference picture in a first color gamut for a lower layer of the video data to a second color gamut for a higher layer of the video data; and generating at least one inter-layer reference picture for the higher layer of the video data based on the converted color data.

[0009] In another example, this disclosure is directed to a video processing device for processing multi-layer video data, the device comprising a memory configured to store the multi-layer video data, and one or more processors in communication with the memory. The one or more processors are configured to generate at least one 3D lookup table for color gamut scalability, wherein the at least one 3D lookup table has a size that is different for a luma component than for a first chroma component and a second chroma component, perform color prediction using the at least one 3D lookup table to convert color data of a reference picture in a first color gamut for a lower layer of the video data to a second color gamut for a higher layer of the video data, and generate at least one inter-layer reference picture for the higher layer of the video data based on the converted color data.

[0010] In a further example, this disclosure is directed to a video processing device for processing multi-layer video data, the device comprising means for generating at least one 3D lookup table for color gamut scalability, wherein the at least one 3D lookup table has a size that is different for a luma component than for a first chroma component and a second chroma component; means for performing color prediction using the at least one 3D lookup table to convert color data of a reference picture in a first color gamut for a lower layer of the video data to a second color gamut for a higher layer of the video data; and means for generating at least one inter-layer reference picture for the higher layer of the video data based on the converted color data.

[0011] In another example, this disclosure is directed to a computer-readable storage medium storing instructions for processing multi-layer video data that, when executed, cause one or more processors to generate at least one 3D lookup table for color gamut

scalability, wherein the at least one 3D lookup table has a size that is different for a luma component than for a first chroma component and a second chroma component; perform color prediction using the at least one 3D lookup table to convert color data of a reference picture in a first color gamut for a lower layer of the video data to a second color gamut for a higher layer of the video data; and generate at least one inter-layer reference picture for the higher layer of the video data based on the converted color data.

[0011a] According to one aspect of the present invention, there is provided a method of processing multi-layer video data, the method comprising: generating at least one three-dimensional (3D) lookup table for color gamut scalability, wherein the at least one 3D lookup table has a size that is different for a luma component than for each of a first chroma component and a second chroma component, wherein generating the at least one 3D lookup table includes partitioning the luma component to have a different number of segments than each of the first chroma component and the second chroma component, and wherein the number of segments for the luma component is indicated by one or more syntax elements included in a bitstream; performing color prediction using the at least one 3D lookup table to convert color data of a reference picture in a first color gamut for a lower layer of the video data to a second color gamut for a higher layer of the video data, wherein the first color gamut and the second color gamut are within the same color space for the video data; and generating at least one inter-layer reference picture for the higher layer of the video data based on the converted color data.

[0011b] According to another aspect of the present invention, there is provided a video processing device for processing multi-layer video data, the device comprising: a memory configured to store the multi-layer video data; and one or more processors in communication with the memory and configured to: generate at least one three-dimensional (3D) lookup table for color gamut scalability, wherein the at least one 3D lookup table has a size that is different for a luma component than for a first chroma component and a second chroma component, wherein the one or more processors are configured to partition the luma component to have a different number of segments than each of the first chroma component and the second chroma

component, and wherein the number of segments for the luma component is indicated by one or more syntax elements included in a bitstream, perform color prediction using the at least one 3D lookup table to convert color data of a reference picture in a first color gamut for a lower layer of the video data to a second color gamut for a higher layer of the video data, wherein the first color gamut and the second color gamut are within the same color space for the video data, and generate at least one inter-layer reference picture for the higher layer of the video data based on the converted color data.

[0011c] According to still another aspect of the present invention, there is provided a video processing device for processing multi-layer video data, the device comprising: means for generating at least one three-dimensional (3D) lookup table for color gamut scalability, wherein the at least one 3D lookup table has a size that is different for a luma component than for a first chroma component and a second chroma component, wherein the means for generating the at least one 3D lookup table includes means for partitioning the luma component to have a different number of segments than each of the first chroma component and the second chroma component, and wherein the number of segments for the luma component is indicated by one or more syntax elements included in a bitstream; means for performing color prediction using the at least one 3D lookup table to convert color data of a reference picture in a first color gamut for a lower layer of the video data to a second color gamut for a higher layer of the video data, wherein the first color gamut and the second color gamut are within the same color space for the video data; and means for generating at least one inter-layer reference picture for the higher layer of the video data based on the converted color data.

[0011d] According to yet another aspect of the present invention, there is provided a non-transitory computer-readable storage medium storing instructions for processing multi-layer video data that, when executed, cause one or more processors to: generate at least one three-dimensional (3D) lookup table for color gamut scalability, wherein the at least one 3D lookup table has a size that is different for a luma component than for a first chroma component and a second chroma component, wherein the instructions cause the one or more processors to

partition the luma component to have a different number of segments than each of the first chroma component and the second chroma component, and wherein the number of segments for the luma component is indicated by one or more syntax elements included in a bitstream; perform color prediction using the at least one 3D lookup table to convert color data of a reference picture in a first color gamut for a lower layer of the video data to a second color gamut for a higher layer of the video data, wherein the first color gamut and the second color gamut are within the same color space for the video data; and generate at least one inter-layer reference picture for the higher layer of the video data based on the converted color data.

[0012] The details of one or more examples are set forth in the accompanying drawings and the description below. Other features, objects, and advantages will be apparent from the description and drawings, and from the claims.

BRIEF DESCRIPTION OF DRAWINGS

[0013] FIG. 1 is a block diagram illustrating an example video encoding and decoding system that may utilize techniques for 3D lookup table based color gamut scalability.

[0014] FIG. 2 is a conceptual illustration showing an example of scalability in three different dimensions.

[0015] FIG. 3 is a conceptual illustration showing an example structure of a scalable video coding bitstream.

[0016] FIG. 4 is a conceptual illustration showing example scalable video coding access units in bitstream order.

[0017] FIG. 5 is a block diagram illustrating an example scalable video coding extension to HEVC (SHVC) encoder.

[0018] FIG. 6 is a graph illustrating an example color gamut of a sample video sequence.

[0019] FIG. 7 is a block diagram illustrating conversion from high definition (HD) color gamut BT.709 to ultra-high definition (UHD) color gamut BT.2020.

Date Reçue/Date Received 2021-03-03

[0020] FIG. 8 is a block diagram illustrating a color gamut scalable coder including a color prediction processing unit that may generate an inter-layer reference picture when a base layer color gamut and an enhancement layer color gamut are different.

[0021] FIG. 9 is a conceptual illustration showing an example 3D lookup table for color gamut scalability.

[0022] FIG. 10 is a conceptual illustration showing tri-linear interpolation with the 3D lookup table for color gamut scalability.

[0023] FIG. 11 is a block diagram illustrating an example of a video encoder that may implement techniques for using 3D lookup table based color gamut scalability in multi-layer video coding.

[0024] FIG. 12 is a block diagram illustrating an example of a video decoder that may implement techniques for using 3D lookup table based color gamut scalability in multi-layer video coding.

[0025] FIG. 13 is a flowchart illustrating an example operation of generating inter-layer reference pictures using 3D lookup table based color gamut scalability.

DETAILED DESCRIPTION

[0026] This disclosure describes techniques for three-dimensional (3D) lookup table based color gamut scalability in multi-layer video coding. The multi-layer video coding may be in accordance the High Efficiency Video Coding (HEVC) standard, including any of a scalable video coding extension, a multiview video coding extension, a 3D video coding (i.e., multiview video coding plus depth) extension, or other multi-layer video coding extensions to HEVC. The techniques may be used by video encoders and/or video decoders to generate inter-layer reference pictures when a color gamut for a lower layer of video data is different than a color gamut for a higher layer of the video data.

[0027] A color gamut comprises a complete range of colors that can be reproduced for an image, e.g., in a picture, slice, block or layer of video data. Conventionally, in multi-layer video coding, a lower layer of video data (e.g., a base layer) and a higher layer of the video data (e.g., an enhancement layer) include color data in the same color gamut, e.g., high definition (HD) color gamut BT.709. In this case, a video encoder and/or video decoder may generate inter-layer reference pictures for the higher layer of the video data as up-sampled versions of co-located reference pictures for the lower layer of the video data.

[0028] In some examples, however, a lower layer of video data may include color data in a first color gamut, e.g., BT.709, and a higher layer of the video data may include color data in a different, second color gamut, e.g., ultra-high definition (UHD) color gamut BT.2020. In this example, in order to generate inter-layer reference pictures for the higher layer of the video data, a video encoder and/or video decoder must first perform color prediction to convert the color data of a reference picture in the first color

gamut for the lower layer of the video data to the second color gamut for the higher layer of the video data.

[0029] The video encoder and/or video decoder may perform color prediction using a 3D lookup table for color gamut scalability. In some examples, a separate 3D lookup table may be generated for each of the color components, i.e., a luma (Y) component, a first chroma (U) component and a second chroma (V) component. Each of the 3D lookup tables includes a luma (Y) dimension, a first chroma (U) dimension and a second chroma (V) dimension, and is indexed using the three independent color components (Y, U, V).

[0030] Conventionally, the 3D lookup tables are always symmetric such that the 3D lookup tables have a same size for the luma component, the first chroma component and the second chroma component. In addition, conventionally, the 3D lookup tables are always balanced such that a size of each dimension of the 3D lookup tables is always the same. This results in large table sizes with high computational complexity and high signaling costs. For example, table sizes may be up to 9x9x9 or 17x17x17.

[0031] According to the techniques described in this disclosure, the video encoder and/or video decoder generates at least one 3D lookup table having a size that is different for the luma component than for each of the first chroma component and the second chroma component. The video encoder and/or video decoder may generate this asymmetric 3D lookup table using a different number of segments for the luma dimension of the 3D lookup table. For example, the video encoder and/or video decoder may generate the 3D lookup table to have a larger size, i.e., more segments, for the luma component than for each of the first and second chroma components. In one example, the 3D lookup table may have a size up to 8x2x2. In this way, the total size of the table may be reduced while maintaining good coding performance with higher resolution for the luma component than for the first and second chroma components.

[0032] In some examples, each of the 3D lookup tables may have the same size MxNxK, in which the size (M) of the luma dimension of the 3D lookup table is different than each of the size (N) of the first chroma dimension of the 3D lookup table and the size (K) of the second chroma dimension of the 3D lookup table. For example, the size of the luma dimension may be larger than the sizes of the chroma dimensions, $M > N$ and $M > K$, and the sizes of the chroma dimensions of the 3D lookup table may be the same, $N=K$. In one example, each of the 3D lookup tables may have the same size of 8x2x2.

[0033] In other examples, a luma component 3D lookup table (LUT_Y) may have a first size, e.g., MxMxM or MxNxK, that is different than a second size of a first chroma component 3D lookup table (LUT_U), e.g., NxNxN, and a third size of a second chroma component 3D lookup table (LUT_V), e.g., KxKxK. For example, the size of the luma component 3D lookup table may be larger than the sizes of the chroma component 3D lookup tables. The first and second chroma component 3D lookup tables may be the same size or different sizes. In one example, LUT_Y may have the size 8x2x2, and each of LUT_U and LUT_V may have the size 2x2x2.

[0034] In a further example, the precision of the luma component 3D lookup table may also be different than the precision of each of the first chroma component 3D lookup table and the second chroma component 3D lookup table. For example, the precision of the luma component 3D lookup table may be higher than the precision of the chroma component 3D lookup tables. In some cases, to further reduce complexity, the video encoder and/or video decoder may only generate a luma component 3D lookup table, perform luma component prediction using the luma component 3D lookup table, and perform first and second chroma component prediction using one-dimensional (1D) linear mapping or piecewise linear mapping.

[0035] As another example, according to the techniques of this disclosure, the video encoder and/or video decoder may generate an unbalanced 3D lookup table using more segments for the dimension of the 3D lookup table associated with the color component used as a table index for the 3D lookup table. For example, the luma component 3D lookup table may have a larger luma dimension than each of a first chroma dimension and a second chroma dimension based on the luma component being used as a table index for the luma component 3D lookup table. In this example, the LUT_Y may have the size MxNxN, where $M > N$. The sizes of the first and second chroma component 3D lookup tables may be similarly determined. For example, the LUT_U may have the size NxMxN based on the first chroma component being used at the table index, and the LUT_V may have the size NxNxM based on the second chroma component being used at the table index. In this way, the total size of each of the tables may be reduced while maintaining good coding performance with higher resolution for the color component used as the table index.

[0036] Video coding standards include ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 or ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual and ITU-T

H.264 (also known as ISO/IEC MPEG-4 AVC), including its Scalable Video Coding (SVC) and Multi-view Video Coding (MVC) extensions.

[0037] The design of a new video coding standard, namely HEVC, has been finalized by the Joint Collaboration Team on Video Coding (JCT-VC) of ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Motion Picture Experts Group (MPEG). An HEVC draft specification referred to as HEVC Working Draft 10 (WD10), Bross et al., "High efficiency video coding (HEVC) text specification draft 10 (for FDIS & Last Call)," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 12th Meeting: Geneva, CH, 14–23 January 2013, JCTVC-L1003v34, is available from http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v34.zip. The finalized HEVC standard is referred to as HEVC version 1.

[0038] A defect report, Wang et al., "High efficiency video coding (HEVC) Defect Report," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 14th Meeting: Vienna, AT, 25 July–2 August 2013, JCTVC-N1003v1, is available from http://phenix.int-evry.fr/jct/doc_end_user/documents/14_Vienna/wg11/JCTVC-N1003-v1.zip. The finalized standard document is published as ITU-T H.265, Series H: Audiovisual and Multimedia Systems, Infrastructure of audiovisual services – Coding of moving video, High efficiency video coding, Telecommunication Standardization Sector of International Telecommunication Union (ITU), April 2013.

[0039] The multi-view extension to HEVC (MV-HEVC) and another HEVC extension for more advanced 3D video coding (3D-HEVC) are being developed by the JCT-3V. A draft specification of MV-HEVC, referred to as MV-HEVC Working Draft 5 (WD5), Tech et al., "MV-HEVC Draft Text 5," Joint Collaborative Team on 3D Video Coding Extension Development (JCT-3V) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 5th Meeting: Vienna, AT, 27 July–2 August 2013, JCT3V-E1004v6, is available from http://phenix.int-evry.fr/jct/doc_end_user/documents/5_Vienna/wg11/JCT3V-E1004-v6.zip. A draft specification of 3D-HEVC, referred to as 3D-HEVC Working Draft 1 (WD1) and described in Tech et al., "3D-HEVC Draft Text 1," Joint Collaborative Team on 3D Video Coding Extension Development (JCT-3V) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 5th Meeting: Vienna, AT, 27 July–2 August 2013, JCT3V-

E1001v3, is available from http://phenix.it-sudparis.eu/jct2/doc_end_user/documents/5_Vienna/wg11/JCT3V-E1001-v3.zip.

[0040] The scalable extension to HEVC (SHVC) is being developed by the JCT-VC. A draft specification of SHVC, referred to as SHVC Working Draft 3 (WD3), Chen et al., "SHVC Draft 3," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 14th Meeting: Vienna, AT, 25 July--2 August 2013, JCTVC-N1008v3, is available from http://phenix.int-evry.fr/jct/doc_end_user/documents/14_Vienna/wg11/JCTVC-N1008-v3.zip.

[0041] FIG. 1 is a block diagram illustrating an example video encoding and decoding system 10 that may utilize techniques for 3D lookup table based color gamut scalability. As shown in FIG. 1, system 10 includes a source device 12 that provides encoded video data to be decoded at a later time by a destination device 14. In particular, source device 12 provides the video data to destination device 14 via a computer-readable medium 16. Source device 12 and destination device 14 may comprise any of a wide range of devices, including desktop computers, notebook (i.e., laptop) computers, tablet computers, set-top boxes, telephone handsets such as so-called "smart" phones, so-called "smart" pads, televisions, cameras, display devices, digital media players, video gaming consoles, video streaming device, or the like. In some cases, source device 12 and destination device 14 may be equipped for wireless communication.

[0042] Destination device 14 may receive the encoded video data to be decoded via computer-readable medium 16. Computer-readable medium 16 may comprise any type of medium or device capable of moving the encoded video data from source device 12 to destination device 14. In one example, computer-readable medium 16 may comprise a communication medium to enable source device 12 to transmit encoded video data directly to destination device 14 in real-time. The encoded video data may be modulated according to a communication standard, such as a wireless communication protocol, and transmitted to destination device 14. The communication medium may comprise any wireless or wired communication medium, such as a radio frequency (RF) spectrum or one or more physical transmission lines. The communication medium may form part of a packet-based network, such as a local area network, a wide-area network, or a global network such as the Internet. The communication medium may include routers, switches, base stations, or any other equipment that may be useful to facilitate communication from source device 12 to destination device 14.

[0043] In some examples, encoded data may be output from output interface 22 to a storage device. Similarly, encoded data may be accessed from the storage device by input interface. The storage device may include any of a variety of distributed or locally accessed data storage media such as a hard drive, Blu-ray discs, DVDs, CD-ROMs, flash memory, volatile or non-volatile memory, or any other suitable digital storage media for storing encoded video data. In a further example, the storage device may correspond to a file server or another intermediate storage device that may store the encoded video generated by source device 12. Destination device 14 may access stored video data from the storage device via streaming or download. The file server may be any type of server capable of storing encoded video data and transmitting that encoded video data to the destination device 14. Example file servers include a web server (e.g., for a website), an FTP server, network attached storage (NAS) devices, or a local disk drive. Destination device 14 may access the encoded video data through any standard data connection, including an Internet connection. This may include a wireless channel (e.g., a Wi-Fi connection), a wired connection (e.g., DSL, cable modem, etc.), or a combination of both that is suitable for accessing encoded video data stored on a file server. The transmission of encoded video data from the storage device may be a streaming transmission, a download transmission, or a combination thereof.

[0044] The techniques of this disclosure are not necessarily limited to wireless applications or settings. The techniques may be applied to video coding in support of any of a variety of multimedia applications, such as over-the-air television broadcasts, cable television transmissions, satellite television transmissions, Internet streaming video transmissions, such as dynamic adaptive streaming over HTTP (DASH), digital video that is encoded onto a data storage medium, decoding of digital video stored on a data storage medium, or other applications. In some examples, system 10 may be configured to support one-way or two-way video transmission to support applications such as video streaming, video playback, video broadcasting, and/or video telephony.

[0045] In the example of FIG. 1, source device 12 includes video source 18, video encoder 20, and output interface 22. Destination device 14 includes input interface 28, video decoder 30, and display device 32. In accordance with this disclosure, video encoder 20 of source device 12 may be configured to apply the techniques for processing video data in parallel. In other examples, a source device and a destination device may include other components or arrangements. For example, source device 12 may receive video data from an external video source 18, such as an external camera.

Likewise, destination device 14 may interface with an external display device, rather than including an integrated display device.

[0046] The illustrated system 10 of FIG. 1 is merely one example. Techniques for processing video data in parallel may be performed by any digital video encoding and/or decoding device. Although generally the techniques of this disclosure are performed by a video encoding device, the techniques may also be performed by a video encoder/decoder, typically referred to as a "CODEC." Moreover, the techniques of this disclosure may also be performed by a video preprocessor. Source device 12 and destination device 14 are merely examples of such coding devices in which source device 12 generates coded video data for transmission to destination device 14. In some examples, devices 12, 14 may operate in a substantially symmetrical manner such that each of devices 12, 14 include video encoding and decoding components. Hence, system 10 may support one-way or two-way video transmission between video devices 12, 14, e.g., for video streaming, video playback, video broadcasting, or video telephony.

[0047] Video source 18 of source device 12 may include a video capture device, such as a video camera, a video archive containing previously captured video, and/or a video feed interface to receive video from a video content provider. As a further alternative, video source 18 may generate computer graphics-based data as the source video, or a combination of live video, archived video, and computer-generated video. In some cases, if video source 18 is a video camera, source device 12 and destination device 14 may form so-called camera phones or video phones. As mentioned above, however, the techniques described in this disclosure may be applicable to video coding in general, and may be applied to wireless and/or wired applications. In each case, the captured, pre-captured, or computer-generated video may be encoded by video encoder 20. The encoded video information may then be output by output interface 22 onto a computer-readable medium 16.

[0048] Computer-readable medium 16 may include transient media, such as a wireless broadcast or wired network transmission, or storage media (that is, non-transitory storage media), such as a hard disk, flash drive, compact disc, digital video disc, Blu-ray disc, or other computer-readable media. In some examples, a network server (not shown) may receive encoded video data from source device 12 and provide the encoded video data to destination device 14, e.g., via network transmission. Similarly, a computing device of a medium production facility, such as a disc stamping facility, may

receive encoded video data from source device 12 and produce a disc containing the encoded video data. Therefore, computer-readable medium 16 may be understood to include one or more computer-readable media of various forms, in various examples.

[0049] Input interface 28 of destination device 14 receives information from computer-readable medium 16. The information of computer-readable medium 16 may include syntax information defined by video encoder 20, which is also used by video decoder 30, that includes syntax elements that describe characteristics and/or processing of blocks and other coded units, e.g., groups of pictures (GOPs). Display device 32 displays the decoded video data to a user, and may comprise any of a variety of display devices such as a cathode ray tube (CRT), a liquid crystal display (LCD), a plasma display, an organic light emitting diode (OLED) display, or another type of display device.

[0050] Video encoder 20 and video decoder 30 each may be implemented as any of a variety of suitable encoder circuitry, such as one or more microprocessors, digital signal processors (DSPs), application specific integrated circuits (ASICs), field programmable gate arrays (FPGAs), discrete logic, software, hardware, firmware or any combinations thereof. When the techniques are implemented partially in software, a device may store instructions for the software in a suitable, non-transitory computer-readable medium and execute the instructions in hardware using one or more processors to perform the techniques of this disclosure. Each of video encoder 20 and video decoder 30 may be included in one or more encoders or decoders, either of which may be integrated as part of a combined encoder/decoder (CODEC) in a respective device.

[0051] In some examples, video encoder 20 and video decoder 30 operate according to a video compression standard, such as ISO/IEC MPEG-4 Visual and ITU-T H.264 (also known as ISO/IEC MPEG-4 AVC), including its Scalable Video Coding (SVC) extension, Multi-view Video Coding (MVC) extension, and MVC-based three-dimensional video (3DV) extension. In some instances, any bitstream conforming to MVC-based 3DV always contains a sub-bitstream that is compliant to a MVC profile, e.g., stereo high profile. Furthermore, there is an ongoing effort to generate a 3DV coding extension to H.264/AVC, namely AVC-based 3DV. In other examples, video encoder 20 and video decoder 30 may operate according to ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 or ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual, and ITU-T H.264, ISO/IEC Visual.

[0052] In the example of FIG. 1, video encoder 20 and video decoder 30 may operate according to the High Efficiency Video Coding (HEVC) standard finalized by the Joint Collaboration Team on Video Coding (JCT-VC) of ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Motion Picture Experts Group (MPEG). The HEVC draft specification, referenced above, is referred to as HEVC Working Draft 10 (WD10), and the finalized version of the HEVC standard is referred to as HEVC version 1. The MV-HEVC and 3D-HEVC are being developed by the JCT-3V. A recent draft specification of MV-HEVC is referred to as MV-HEVC WD5, and a recent draft specification of 3D-HEVC is referred to as 3D-HEVC WD1. The SHVC is being developed by the JCT-VC. A recent draft specification of SHVC is referred to as SHVC WD3.

[0053] In HEVC and other video coding standards, a video sequence typically includes a series of pictures. Pictures may also be referred to as "frames." A picture may include three sample arrays, denoted S_L, S_{Cb}, and S_{Cr}. S_L is a two-dimensional array (i.e., a block) of luma samples. S_{Cb} is a two-dimensional array of Cb chrominance samples. S_{Cr} is a two-dimensional array of Cr chrominance samples. Chrominance samples may also be referred to herein as "chroma" samples. In other instances, a picture may be monochrome and may only include an array of luma samples.

[0054] Video encoder 20 may generate a set of coding tree units (CTUs). Each of the CTUs may comprise a coding tree block of luma samples, two corresponding coding tree blocks of chroma samples, and syntax structures used to code the samples of the coding tree blocks. In a monochrome picture or a picture that has three separate color planes, a CTU may comprise a single coding tree block and syntax structures used to code the samples of the coding tree block. A coding tree block may be an NxN block of samples. A CTU may also be referred to as a "tree block" or a "largest coding unit" (LCU). The CTUs of HEVC may be broadly analogous to the macroblocks of other video coding standards, such as H.264/AVC. However, a CTU is not necessarily limited to a particular size and may include one or more coding units (CUs). A slice may include an integer number of CTUs ordered consecutively in the raster scan.

[0055] This disclosure may use the term "video unit" or "video block" to refer to one or more blocks of samples and syntax structures used to code samples of the one or more blocks of samples. Example types of video units may include CTUs, CUs, PUs, transform units (TUs) in HEVC, or macroblocks, macroblock partitions, and so on in other video coding standards.

[0056] To generate a coded CTU, video encoder 20 may recursively perform quad-tree partitioning on the coding tree blocks of a CTU to divide the coding tree blocks into coding blocks, hence the name "coding tree units." A coding block is an NxN block of samples. A CU may comprise a coding block of luma samples and two corresponding coding blocks of chroma samples of a picture that has a luma sample array, a Cb sample array and a Cr sample array, and syntax structures used to code the samples of the coding blocks. In a monochrome picture or a picture that has three separate color planes, a CU may comprise a single coding block and syntax structures used to code the samples of the coding block.

[0057] Video encoder 20 may partition a coding block of a CU into one or more prediction blocks. A prediction block may be a rectangular (i.e., square or non-square) block of samples on which the same prediction is applied. A prediction unit (PU) of a CU may comprise a prediction block of luma samples, two corresponding prediction blocks of chroma samples of a picture, and syntax structures used to predict the prediction block samples. In a monochrome picture or a picture that have three separate color planes, a PU may comprise a single prediction block and syntax structures used to predict the prediction block samples. Video encoder 20 may generate predictive luma, Cb and Cr blocks for luma, Cb and Cr prediction blocks of each PU of the CU.

[0058] Video encoder 20 may use intra prediction or inter prediction to generate the predictive blocks for a PU. If video encoder 20 uses intra prediction to generate the predictive blocks of a PU, video encoder 20 may generate the predictive blocks of the PU based on decoded samples of the picture associated with the PU.

[0059] If video encoder 20 uses inter prediction to generate the predictive blocks of a PU, video encoder 20 may generate the predictive blocks of the PU based on decoded samples of one or more pictures other than the picture associated with the PU. Inter prediction may be uni-directional inter prediction (i.e., uni-prediction) or bi-directional inter prediction (i.e., bi-prediction). To perform uni-prediction or bi-prediction, video encoder 20 may generate a first reference picture list (RefPicList0) and a second reference picture list (RefPicList1) for a current slice.

[0060] Each of the reference picture lists may include one or more reference pictures. When using uni-prediction, video encoder 20 may search the reference pictures in either or both RefPicList0 and RefPicList1 to determine a reference location within a reference picture. Furthermore, when using uni-prediction, video encoder 20 may generate, based at least in part on samples corresponding to the reference location, the

predictive sample blocks for the PU. Moreover, when using uni-prediction, video encoder 20 may generate a single motion vector that indicates a spatial displacement between a prediction block of the PU and the reference location. To indicate the spatial displacement between a prediction block of the PU and the reference location, a motion vector may include a horizontal component specifying a horizontal displacement between the prediction block of the PU and the reference location and may include a vertical component specifying a vertical displacement between the prediction block of the PU and the reference location.

[0061] When using bi-prediction to encode a PU, video encoder 20 may determine a first reference location in a reference picture in RefPicList0 and a second reference location in a reference picture in RefPicList1. Video encoder 20 may then generate, based at least in part on samples corresponding to the first and second reference locations, the predictive blocks for the PU. Moreover, when using bi-prediction to encode the PU, video encoder 20 may generate a first motion indicating a spatial displacement between a sample block of the PU and the first reference location and a second motion indicating a spatial displacement between the prediction block of the PU and the second reference location.

[0062] After video encoder 20 generates predictive luma, Cb, and Cr blocks for one or more PUs of a CU, video encoder 20 may generate a luma residual block for the CU. Each sample in the CU's luma residual block indicates a difference between a luma sample in one of the CU's predictive luma blocks and a corresponding sample in the CU's original luma coding block. In addition, video encoder 20 may generate a Cb residual block for the CU. Each sample in the CU's Cb residual block may indicate a difference between a Cb sample in one of the CU's predictive Cb blocks and a corresponding sample in the CU's original Cb coding block. Video encoder 20 may also generate a Cr residual block for the CU. Each sample in the CU's Cr residual block may indicate a difference between a Cr sample in one of the CU's predictive Cr blocks and a corresponding sample in the CU's original Cr coding block.

[0063] Furthermore, video encoder 20 may use quad-tree partitioning to decompose the luma, Cb and, Cr residual blocks of a CU into one or more luma, Cb, and Cr transform blocks. A transform block may be a rectangular block of samples on which the same transform is applied. A transform unit (TU) of a CU may comprise a transform block of luma samples, two corresponding transform blocks of chroma samples, and syntax structures used to transform the transform block samples. In a monochrome picture or a

picture that has three separate color planes, a TU may comprise a single transform block and syntax structures used to transform the transform block samples. Thus, each TU of a CU may be associated with a luma transform block, a Cb transform block, and a Cr transform block. The luma transform block associated with the TU may be a sub-block of the CU's luma residual block. The Cb transform block may be a sub-block of the CU's Cb residual block. The Cr transform block may be a sub-block of the CU's Cr residual block.

[0064] Video encoder 20 may apply one or more transforms to a luma transform block of a TU to generate a luma coefficient block for the TU. A coefficient block may be a two-dimensional array of transform coefficients. A transform coefficient may be a scalar quantity. Video encoder 20 may apply one or more transforms to a Cb transform block of a TU to generate a Cb coefficient block for the TU. Video encoder 20 may apply one or more transforms to a Cr transform block of a TU to generate a Cr coefficient block for the TU.

[0065] After generating a coefficient block (e.g., a luma coefficient block, a Cb coefficient block or a Cr coefficient block), video encoder 20 may quantize the coefficient block. Quantization generally refers to a process in which transform coefficients are quantized to possibly reduce the amount of data used to represent the transform coefficients, providing further compression. Furthermore, video encoder 20 may inverse quantize transform coefficients and apply an inverse transform to the transform coefficients in order to reconstruct transform blocks of TUs of CUs of a picture. Video encoder 20 may use the reconstructed transform blocks of TUs of a CU and the predictive blocks of PUs of the CU to reconstruct coding blocks of the CU. By reconstructing the coding blocks of each CU of a picture, video encoder 20 may reconstruct the picture. Video encoder 20 may store reconstructed pictures in a decoded picture buffer (DPB). Video encoder 20 may use reconstructed pictures in the DPB for inter prediction and intra prediction.

[0066] After video encoder 20 quantizes a coefficient block, video encoder 20 may entropy encode syntax elements that indicate the quantized transform coefficients. For example, video encoder 20 may perform Context-Adaptive Binary Arithmetic Coding (CABAC) on the syntax elements indicating the quantized transform coefficients. Video encoder 20 may output the entropy-encoded syntax elements in a bitstream.

[0067] Video encoder 20 may output a bitstream that includes a sequence of bits that forms a representation of coded pictures and associated data. The bitstream may

comprise a sequence of network abstraction layer (NAL) units. Each of the NAL units includes a NAL unit header and encapsulates a raw byte sequence payload (RBSP). The NAL unit header may include a syntax element that indicates a NAL unit type code. The NAL unit type code specified by the NAL unit header of a NAL unit indicates the type of the NAL unit. A RBSP may be a syntax structure containing an integer number of bytes that is encapsulated within a NAL unit. In some instances, an RBSP includes zero bits.

[0068] Different types of NAL units may encapsulate different types of RBSPs. For example, a first type of NAL unit may encapsulate a RBSP for a picture parameter set (PPS), a second type of NAL unit may encapsulate a RBSP for a coded slice, a third type of NAL unit may encapsulate a RBSP for Supplemental Enhancement Information (SEI), and so on. A PPS is a syntax structure that may contain syntax elements that apply to zero or more entire coded pictures. NAL units that encapsulate RBSPs for video coding data (as opposed to RBSPs for parameter sets and SEI messages) may be referred to as video coding layer (VCL) NAL units. A NAL unit that encapsulates a coded slice may be referred to herein as a coded slice NAL unit. A RBSP for a coded slice may include a slice header and slice data.

[0069] Video decoder 30 may receive a bitstream. In addition, video decoder 30 may parse the bitstream to decode syntax elements from the bitstream. Video decoder 30 may reconstruct the pictures of the video data based at least in part on the syntax elements decoded from the bitstream. The process to reconstruct the video data may be generally reciprocal to the process performed by video encoder 20. For instance, video decoder 30 may use motion vectors of PUs to determine predictive blocks for the PUs of a current CU. Video decoder 30 may use a motion vector or motion vectors of PUs to generate predictive blocks for the PUs.

[0070] In addition, video decoder 30 may inverse quantize coefficient blocks associated with TUs of the current CU. Video decoder 30 may perform inverse transforms on the coefficient blocks to reconstruct transform blocks associated with the TUs of the current CU. Video decoder 30 may reconstruct the coding blocks of the current CU by adding the samples of the predictive sample blocks for PUs of the current CU to corresponding samples of the transform blocks of the TUs of the current CU. By reconstructing the coding blocks for each CU of a picture, video decoder 30 may reconstruct the picture. Video decoder 30 may store decoded pictures in a decoded picture buffer for output and/or for use in decoding other pictures.

[0071] In MV-HEVC, 3D-HEVC, and SHVC, a video encoder may generate a multi-layer bitstream that comprises a series of network abstraction layer (NAL) units. Different NAL units of the bitstream may be associated with different layers of the bitstream. A layer may be defined as a set of video coding layer (VCL) NAL units and associated non-VCL NAL units that have the same layer identifier. A layer may be equivalent to a view in multi-view video coding. In multi-view video coding, a layer can contain all view components of the same layer with different time instances. Each view component may be a coded picture of the video scene belonging to a specific view at a specific time instance. In 3D video coding, a layer may contain either all coded depth pictures of a specific view or coded texture pictures of a specific view. Similarly, in the context of scalable video coding, a layer typically corresponds to coded pictures having video characteristics different from coded pictures in other layers. Such video characteristics typically include spatial resolution and quality level (Signal-to-Noise Ratio). In HEVC and its extensions, temporal scalability may be achieved within one layer by defining a group of pictures with a particular temporal level as a sub-layer.

[0072] For each respective layer of the bitstream, data in a lower layer may be decoded without reference to data in any higher layer. In scalable video coding, for example, data in a base layer may be decoded without reference to data in an enhancement layer. NAL units only encapsulate data of a single layer. In SHVC, a view may be referred to as a "base layer" if a video decoder can decode pictures in the view without reference to data of any other layer. The base layer may conform to the HEVC base specification. Thus, NAL units encapsulating data of the highest remaining layer of the bitstream may be removed from the bitstream without affecting the decodability of data in the remaining layers of the bitstream. In MV-HEVC and 3D-HEVC, higher layers may include additional view components. In SHVC, higher layers may include signal to noise ratio (SNR) enhancement data, spatial enhancement data, and/or temporal enhancement data.

[0073] In some examples, data in a higher layer may be decoded with reference to data in one or more lower layers. The lower layers may be used as reference pictures to compress the higher layer using inter-layer prediction. The data of the lower layers may be up-sampled to have the same resolution as the higher layers. In general, video encoder 20 and video decoder 30 may perform inter-layer prediction in a similar manner as inter prediction described above, except one or more up-sampled lower layers may be used as reference pictures as opposed to one or more neighboring pictures.

[0074] FIG. 2 is a conceptual illustration showing an example of scalability in three different dimensions. In a scalable video coding structure, scalabilities are enabled in three dimensions. In the example of FIG. 2, the scalabilities are enabled in a spatial (S) dimension 100, a temporal (T) dimension 102, and a signal-to-noise ratio (SNR) or quality (Q) dimension 104. In the temporal dimension 102, frame rates with 7.5 Hz (T0), 15 Hz (T1) or 30 Hz (T2), for example, may be supported by temporal scalability. When spatial scalability is supported, different resolutions such as QCIF (S0), CIF (S1) and 4CIF (S2), for example, are enabled in the spatial dimension 100. For each specific spatial resolution and frame rate, SNR layers (Q1) can be added in the SNR dimension 104 to improve the picture quality.

[0075] Once video content has been encoded in such a scalable way, an extractor tool may be used to adapt the actual delivered content according to application requirements, which are dependent e.g., on the clients or the transmission channel. In the example shown in FIG. 2, each cubic contains pictures with the same frame rate (temporal level), spatial resolution, and SNR layers. Better representation may be achieved by adding cubes (i.e., pictures) in any of dimensions 100, 102 or 104. Combined scalability is supported when there are two, three or even more scalabilities enabled.

[0076] In scalable video coding standards, such as the SVC extension to H.264 or SHVC, the pictures with the lowest spatial and SNR layer are compatible with the single layer video codec, and the pictures at the lowest temporal level form the temporal base layer, which may be enhanced with pictures at higher temporal levels. In addition to the base layer, several spatial and/or SNR enhancement layers may be added to provide spatial and/or quality scalabilities. Each spatial or SNR enhancement layer itself may be temporally scalable, with the same temporal scalability structure as the base layer. For one spatial or SNR enhancement layer, the lower layer it depends on may be referred as the base layer of that specific spatial or SNR enhancement layer.

[0077] FIG. 3 is a conceptual illustration showing an example structure 110 of a scalable video coding bitstream. The bitstream structure 110 includes a layer 0 112 that includes pictures or slices I0, P4 and P8, and a layer 1 114 that includes pictures or slices B2, B6 and B10. In addition, bitstream structure 110 includes a layer 2 116 and a layer 3 117 that each includes pictures 0, 2, 4, 6, 8 and 10, and a layer 4 118 that includes pictures 0 through 11.

[0078] A base layer has the lowest spatial and quality layer (i.e., pictures in layer 0 112 and layer 1 114 with QCIF resolution). Among them, those pictures of the lowest

temporal level form the temporal base layer, as shown in layer 0 112 of FIG. 3. The temporal base layer (layer 0) 112 can be enhanced with pictures of a higher temporal level, e.g., layer 1 114 with frame rate of 15 Hz or layer 4 118 with frame rate of 30 Hz.

[0079] In addition to the base layer 112, 114, several spatial and/or SNR enhancement layers may be added to provide spatial and/or quality scalabilities. For example, layer 2 116 with CIF resolution may be a spatial enhancement layer to base layer 112, 114. In another example, layer 3 117 may be an SNR enhancement layer to base layer 112, 114 and layer 2 116. As shown in FIG. 3, each spatial or SNR enhancement layer itself may be temporally scalable, with the same temporal scalability structure as the base layer 112, 114. In addition, an enhancement layer may enhance both spatial resolution and frame rate. For example, layer 4 118 provides a 4CIF resolution enhancement layer, which further increases the frame rate from 15 Hz to 30 Hz.

[0080] FIG. 4 is a conceptual illustration showing example scalable video coding access units 120A-120E ("access units 120") in bitstream order. As shown in FIG. 4, the coded pictures or slices in the same time instance are successive in the bitstream order and form one access unit in the context of a scalable video coding standard, such as the SVC extension to H.264 or SHVC. The access units 120 then follow the decoding order, which could be different from the display order and determined, for example, by the temporal prediction relationship between access units 120.

[0081] For example, access unit 120A includes picture I0 from layer 0 112, picture 0 from layer 2 116, picture 0 from layer 3 117, and picture 0 from layer 4 118. Access unit 120B includes picture P4 from layer 0 112, picture 4 from layer 2 116, picture 4 from layer 3 117, and picture 4 from layer 4 118. Access unit 120C includes picture B2 from layer 1 114, picture 2 from layer 2 116, picture 2 from layer 3 117, and picture 2 from layer 4 118. Access unit 120D includes picture 1 from layer 4 118, and access unit 120E includes picture 3 from layer 4 118.

[0082] FIG. 5 is a block diagram illustrating an example 3-layer SHVC encoder 122. As illustrated in FIG. 5, SHVC encoder 122 includes a base layer encoder 124, a first enhancement layer encoder 125 and a second enhancement layer encoder 126. In high-level syntax only SHVC, there are no new block level coding tools when compared to HEVC single layer coding. In SHVC, only slice and above level syntax changes and picture level operation, such as picture filtering or up-sampling, are allowed.

[0083] To reduce the redundancy between layers, up-sampled co-located reference layer pictures for a lower/base layer may generated and stored in a reference buffer for a

higher/enhancement layer so that inter-layer prediction may be achieved in the same way as inter-frame prediction within a single layer. As illustrated in FIG. 5, a resampled inter-layer reference (ILR) picture 128 is generated from a reference picture in base layer encoder 124 and stored in first enhancement layer encoder 125. Similarly, a resampled ILR picture 129 is generated from a reference picture in first enhancement layer encoder 125 and stored in second enhancement layer encoder 126. In SHVC WD3, the ILR picture is marked as a long term reference picture for the enhancement layer. The motion vector difference associated with an inter-layer reference picture is constrained to zero.

[0084] The upcoming deployment of ultra-high definition television (UHDTV) devices and content will use a different color gamut than legacy devices. Specifically, HD uses the BT.709 recommendation, ITU-R Recommendation BT.709 "Parameter values for the HDTV standards for production and international programme exchange" Dec. 2010, while UHDTV will use the BT.2020 recommendation, ITU-R Recommendation BT.2020 "Parameter values for UHDTV systems for production and international programme exchange" April 2012. A color gamut comprises a complete range of colors that can be reproduced for an image, e.g., in a picture, slice, block or layer of video data. A key difference between these systems is that the color gamut of UHDTV is significantly larger than HD. It is asserted that UHDTV will provide a more life-like or realistic viewing experience, which is consistent with other UHDTV characteristics, such as high resolution.

[0085] FIG. 6 is a graph illustrating an example color gamut of a sample video sequence 130. As illustrated in FIG. 6, the SWG1 sample video sequence 130 is indicated as a cluster of dots within a line outline of the UHD color gamut BT.2020 132. For comparison purposes, an outline of the HD color gamut BT.709 134 and an outline of the International Commission on Illumination (CIE)-XYZ linear color space 136 overlays the SWG1 sample video sequence 130. It is easily observed from FIG. 6 that the UHD color gamut BT.2020 132 is much larger than the HD color gamut BT.709 134. Note the number of pixels in the SWG1 sample video sequence 130 that fall outside of the BT.709 color gamut 134.

[0086] FIG. 7 is a block diagram illustrating conversion from HD color gamut BT.709 134 to UHD color gamut BT.2020 132. Both the HD color gamut BT.709 134 and the UHD color gamut BT.2020 132 define representations of color pixels in luma and chroma components (e.g., YCbCr or YUV). Each color gamut defines conversion to

and from the CIE-XYZ linear color space 136. This common intermediate color space may be used to define the conversion of the luma and chroma values in the HD color gamut BT.709 134 to corresponding luma and chroma values in the UHD color gamut BT.2020 132.

[0087] More details regarding the color gamut of the sample sequence illustrated in FIG. 6 and the color gamut conversion illustrated in FIG. 7 may be found in L. Kerofsky, A. Segall, S.-H. Kim, K. Misra, "Color Gamut Scalable Video Coding: New Results," JCTVC-L0334, Geneva, CH, 14–23 Jan. 2013 (hereinafter referred to as "JCTVC-L0334").

[0088] FIG. 8 is a block diagram illustrating a color gamut scalable coder 140 including a color prediction processing unit 144 that may generate an inter-layer reference picture when a base layer color gamut and an enhancement layer color gamut are different. Color prediction processing unit 144 may be used by a video coder, such as video encoder 20 or video decoder 30 from FIG. 1, to perform color gamut scalable video coding, in which the color gamut of the base and enhancement layer is different.

[0089] In the example illustrated in FIG. 8, a base layer coding loop 142 performs video coding of pictures that include color data in a first color gamut, e.g., BT.709, and an enhancement layer coding loop 146 performs video coding of pictures that include color data in a second color gamut, e.g., BT.2020. Color prediction processing unit 144 performs color prediction to map or convert color data of a base layer reference picture in the first color gamut to the second color gamut, and generates an inter-layer reference picture for the enhancement layer based on the mapped color data of the base layer reference picture.

[0090] To achieve high coding efficiency, color prediction processing unit 144 is configured to perform specific color prediction when generating inter-layer reference pictures. As described in more detail below, color prediction processing unit 144 may be configured to perform color prediction according to any of a linear prediction model, a piecewise linear prediction model, or a 3D lookup table based color prediction model.

[0091] A linear prediction model is proposed in JCTVC-L0334, referenced above. Generally, the color prediction process of the linear prediction model may be described as a gain and offset model. The linear prediction model operates on individual color planes. To facilitate integer calculation, a parameter describes the number of fractional bits used in the calculation using the parameter numFractionBits. For each channel, a gain[c] and offset[c] are specified. The linear prediction model is defined as follows:

Pred[c][x][y] = (gain[c]*In[x][y] + (1<<(numFractionBits-1)) >> numFractionBits + offset[c]

[0092] A piecewise linear prediction model is proposed in C. Auyeung, K. Sato, "AHG14: Color gamut scalable video coding with piecewise linear predictions and shift-offset model," JCTVC-N0271, Vienna, Austria, July 2013, based on JCTVC-L0334, referenced above. The color prediction process of the piecewise linear prediction model may also be described as a gain and offset model. The piecewise linear prediction model is defined as follows:

Let d[c][x][y] = In[c][x][y] - knot[c]. If d[c][x][y] <= 0

Pred[c][x][y] = (gain1[c]*d[c][x][y] + offset[c] + (1<<(numFractionBits-1))) >> numFractionBits

else

Pred[c][x][y] = (gain2[c]*d[c][x][y] + offset[c] + (1<<(numFractionBits-1))) >> numFractionBits

The prediction parameters knot[c], offset[c], gain1[c], and gain2[c] may be encoded in the bitstream.

[0093] FIG. 9 is a conceptual illustration showing an example 3D lookup table 150 for color gamut scalability. A 3D lookup table based color prediction model is proposed in P. Bordes, P. Andrivon, F. Hiron, "AHG14: Color Gamut Scalable Video Coding using 3D LUT: New Results," JCTVC-N0168, Vienna, Austria, Jul. 2013 (hereinafter referred to as "JCTVC-N0168"). The principle of the 3D lookup table for color gamut scalability is depicted in FIG. 9. The 3D lookup table 150 can be considered as a sub-sampling of a first 3D color space, e.g., HD color gamut BT.709, where each vertex is associated with a color triplet (y, u, v) corresponding to second 3D color space (i.e., predicted) values, e.g., UHD color gamut BT.2020).

[0094] In general, the first color gamut may be partitioned into octants or cuboids in each color dimension (i.e., Y, U, and V), and the vertices of the octants are associated with the color triplet corresponding to the second color gamut and used to populate 3D lookup table 150. The number of vertices or segments in each color dimension indicates the size of 3D lookup table. FIG. 9(a) illustrates the vertices or intersecting lattice points of the octants in each color dimension. FIG. 9(b) illustrates the different color values associated with each of the vertices. As illustrated, in FIG. 9(a) each color dimension has four vertices and in FIG. 9(b) each color dimension includes four color values.

[0095] FIG. 10 is a conceptual illustration showing tri-linear interpolation with the 3D lookup table 150 for color gamut scalability. For a given base layer color sample in the first color gamut, the computation of its prediction in the second color gamut for an enhancement layer is made using tri-linear interpolation according to the following equation.

$$\overline{value_y} = K \times \sum_{i=0,1} \sum_{j=0,1} \sum_{k=0,1} s_i(y) \times s_j(u) \times s_k(v) \times \mathrm{LUT}[y_i][u_j][v_k].y$$

Where: $K = \frac{1}{(y_1 - y_0) \times (u_1 - u_0) \times (v_1 - v_0)}$

$S_0(y) = y_1 - y$ and $S_1(y) = y - y_0$

y_0 is the index of the nearest sub-sampled vertex inferior to y,

y_1 is the index of the nearest sub-sampled vertex superior to y.

[0096] In some examples, a separate 3D lookup table may be generated for each of the color components, i.e., a luma (Y) component, a first chroma (U) component and a second chroma (V) component. Each of the 3D lookup tables includes a luma (Y) dimension, a first chroma (U) dimension and a second chroma (V) dimension, and is indexed using the three independent color components (Y, U, V).

[0097] In one example, a mapping function may be defined for each color component based on the 3D lookup table. An example mapping function for a luma (Y) pixel value is presented in the following equation:

$$Y_E = \mathrm{LUT}_Y(Y_B, U_B, V_B)*Y_B + \mathrm{LUT}_U(Y_B, U_B, V_B)*U_B + \mathrm{LUT}_V(Y_B, U_B, V_B)*V_B + \mathrm{LUT}_C(Y_B, U_B, V_B)$$

In the above equation, Y_E represents the luma pixel value in the enhancement layer, (Y_B, U_B, V_B) represents a base layer pixel value, and LUT_Y, LUT_U, LUT_V and LUT_C represent the 3D lookup table for each color component Y, U, V, and a constant. Similar mapping functions may be defined for a first chroma (U) pixel value and a second chroma (V) pixel value in the enhancement layer.

[0098] More details of the 3D lookup table illustrated in FIG. 9 and the tri-linear interpolation with the 3D lookup table illustrated in FIG. 10 may be found in JCTVC-N0168, referenced above.

[0099] In general, 3D lookup table based color gamut scalability results in good coding performance. The size of the 3D lookup table may be concern, however, since the 3D lookup table is typically stored in cache memory in a hardware implementation. A 3D lookup table with a large table size may result in high computational complexity and a

high signaling cost. For example, conventionally, the 3D lookup tables are always symmetric such that the 3D lookup tables have a same size for the luma component, the first chroma component and the second chroma component. In addition, conventionally, the 3D lookup tables are always balanced such that a size of each dimension of the 3D lookup tables is always the same. This results in large table sizes with high computational complexity and high signaling costs. For example, table sizes may be up to 9x9x9 or 17x17x17.

[0100] The following methods are proposed so that both signaling cost and computational cost for the 3D lookup table based color gamut scalability may be reduced.

[0101] The first method includes generating an asymmetric 3D lookup table such that the luma (Y) and chroma (U and V) components have different sizes. In some cases, the 3D lookup table may have a larger size, i.e., more segments, for the luma component than for each of the first and second chroma components. In this case, the chroma components may use a coarser lookup table and the luma component may use a more refined lookup table. In other cases, the 3D lookup table may have a larger size for one or both of the chroma components than for the luma component.

[0102] In general, the 3D lookup table based color prediction is a kind of 3D piecewise linear prediction. Theoretically, the more segments used in each color dimension (i.e., Y, U, and V) of the 3D lookup table, the higher the color prediction accuracy. A large number of segments, however, may lead to a high signaling cost and high computational complexity (i.e., larger table size). In practice, the chroma signal may be less important and easier to predict. In this case, it may be better to have a high-resolution lookup table for the luma (Y) component and have low-resolution lookup tables for each of the first chroma (U) component and the second chroma (V) component.

[0103] According to the techniques of this disclosure, a video coder, such as video encoder 20 and/or video decoder 30, may generate the asymmetric 3D lookup table using a different number of segments for the luma dimension of the 3D lookup table than for the first chroma dimension and the second chroma dimension of the 3D lookup table. For example, video encoder 20 and/or video decoder 30 may generate the 3D lookup table to have a larger size, i.e., more segments, for the luma component than for each of the first and second chroma components. In one example, the 3D lookup table may have a size up to 8x2x2. In this way, the total size of the table may be reduced and the signaling cost may also be reduced while maintaining good coding performance

with higher resolution for the luma component than for the first and second chroma components. The table sizes for the 3D lookup table may be signaled in a bitstream such as in a video parameter set (VPS), sequence parameter set (SPS), picture parameter set (PPS), slice header or related extensions.

[0104] In some examples, each of the 3D lookup tables may have the same size MxNxK, in which the size (M) of the luma dimension of the 3D lookup table is different than the size (N) of the first chroma dimension of the 3D lookup table and the size (K) of the second chroma dimension of the 3D lookup table. For example, the size of the luma dimension may be larger than the sizes of the chroma dimensions, M > N and M > K, and the sizes of the chroma dimensions of the 3D lookup table may be the same, N=K. In one example, each of the 3D lookup tables may have the same size of 8x2x2. In another example, each of the 3D lookup tables may have the same size of 9x6x6.

[0105] In other examples, a luma component 3D lookup table (LUT_Y) may have a first size, e.g., MxMxM or MxNxK, that is different than a second size of a first chroma component 3D lookup table (LUT_U), e.g., NxNxN, and a third size of a second chroma component 3D lookup table (LUT_V), e.g., KxKxK. For example, the size of the LUT_Y may be larger than the sizes of LUT_U and LUT_V. The sizes of LUT_U and LUT_V may be the same or different from each other. In one example, LUT_Y may have the size 8x2x2, and each of LUT_U and LUT_V may have the size 2x2x2. In another example, LUT_Y may have the size 9x9x9 or 9x6x6, and each of LUT_U and LUT_V may have the size 9x3x3 or 3x3x3.

[0106] In a further example, the precision of the 3D lookup table may be dependent on the associated color component. For example, a first precision value of the luma component 3D lookup table may be different than a second precision value of both the first and second chroma component 3D lookup tables. For example, the first precision value of the luma component 3D lookup table may be higher than the second precision value of the chroma component 3D lookup tables. In one example, the first precision value may be 8-bit for the luma component and the second precision value may be 6-bit for the chroma components. In some cases, when generating the 3D lookup tables, an additional shift may be applied in order to meet a target bit depth of the enhancement layer. The use of different precision values from a default precision value for 3D lookup tables based on the associated color component may be indicated in the VPS, SPS, PPS, slice header or related extensions.

[0107] The second method includes generating an unbalanced 3D lookup table, i.e. table[M][N][K], such that the size of each dimension is different depending on which color component is being used as a table index for the 3D lookup table. The 3D lookup table may have a larger size for the dimension associated with the color component used as the table index. In this case, the color mapping may be more accurate for the color component used as the table index, while being less accurate for the other color components.

[0108] The 3D lookup table proposed in JCTVC-N0168 is always balanced such that the size of each dimension of the 3D lookup table is always same. As discussed above, the more segments used in each color dimension (i.e. Y, U, and V) of the 3D lookup table, the better the color prediction efficiency. A large number of segments, however, may lead to a high signaling cost and high computational complexity (i.e., larger table size). Considering that each color component in a first color gamut usually has higher correlation with the same color component in a second color gamut, it may be more helpful to improve the prediction efficiency by using more segments, i.e. using a larger size, for the table dimension when the associated color component is used as a table index for the 3D lookup table. The size of the table dimension may be smaller when a different color component is used as a table index for the 3D lookup table.

[0109] According to the techniques of this disclosure, a video coder, such as video encoder 20 and/or video decoder 30, may generate the unbalanced 3D lookup table using more segments for the dimension of the 3D lookup table associated with the color component used as a table index for the 3D lookup table. For example, the luma component 3D lookup table may have a larger luma dimension than a first chroma dimension and a second chroma dimension based on the luma component being used as a table index for the luma component 3D lookup table. In one example, the LUT_Y may have the size MxNxN, where M > N. The sizes of the first and second chroma component 3D lookup tables may be similarly determined. For example, the LUT_U may have the size NxMxN based on the first chroma component being used at the table index, and the LUT_V may have the size NxNxM based on the second chroma component being used at the table index. In this way, the total size of each of the tables may be reduced while maintaining good coding performance with higher resolution for the color component used as the table index.

[0110] In one example, when the Y, U, V component is used as input to the 3D lookup table, an 8x2x2 table may be used for the Y component, a 2x8x2 table may be used for

the U component, and a 2x2x8 table may be used for the V component. In another example, when the Y, U, V component is used as input to the 3D lookup table, a 9x3x3 table may be used for the Y component, a 3x9x3 table may be used for the U component, and a 3x3x9 table may be used for the V component.

[0111] The table sizes for the 3D lookup table may be signaled in a bitstream such as in the VPS, SPS, PPS, slice header or related extensions. In the above examples, only the values of M and N may be signaled to indicate the table sizes. In some cases, instead of signaling the values of M and N, default values may be set for M and N so that no signaling of the table size is needed. For example, N may be set to a value of 3 or a value of 2 by default.

[0112] The third method includes generating only a luma component 3D lookup table, and only using the 3D lookup table to perform luma component prediction, while using one-dimensional (1D) linear mapping or piecewise linear mapping for the chroma components. In this example, the 3D lookup table is simplified such that it is only applied for luma component prediction. For chroma component prediction, the piecewise linear prediction model may be used. Since the piecewise linear prediction model shows similar performance to the 3D lookup table based color prediction model for chroma components, this simplification may maintain the prediction accuracy while reducing the computational complexity and the signaling cost. For example, only using a luma component 3D lookup table may reduce the computational complexity by avoiding tri-linear interpolation for the chroma components, and may reduce the signaling cost by not signaling 3D lookup tables for the chroma components.

[0113] When the 3D lookup table is only used to perform luma component prediction, the 3D mapping function described above with respect to FIG. 9 and FIG. 10 may be simplified to a 1D mapping function for the chroma components. Example mapping functions for a first chroma (U) pixel value and a second chroma (V) pixel value are presented in the following equation. The 3D mapping function for the luma (Y) pixel value may be the same as presented above.

$$U_E = LUT_U(U_B)*U_B + LUT_C(U_B)$$

$$V_E = LUT_V(V_B)*V_B + LUT_C(V_B)$$

In the above equations, U_E represents the first chroma pixel value in the enhancement layer, V_E represents the second chroma pixel value in the enhancement layer, U_B and V_B represent a base layer chroma pixel values, and LUT_U, LUT_V and LUT_C represent the 1D lookup table for each chroma component U, V, and a constant.

[0114] The use of any of the methods described above may be indicated in the VPS, SPS, PPS, slice header or related extension. In some examples, one or more of the methods described above for performing 3D lookup table based color gamut scalable may be combined.

[0115] FIG. 11 is a block diagram illustrating an example of video encoder 20 that may implement techniques for using 3D lookup table based color gamut scalability in multi-layer video coding. Video encoder 20 may perform intra- and inter-coding of video blocks within video slices. Intra-coding relies on spatial prediction to reduce or remove spatial redundancy in video within a given video frame or picture. Inter-coding relies on temporal prediction to reduce or remove temporal redundancy in video within adjacent frames or pictures of a video sequence. Intra-mode (I mode) may refer to any of several spatial based coding modes. Inter-modes, such as uni-directional prediction (P mode) or bi-prediction (B mode), may refer to any of several temporal-based coding modes.

[0116] As shown in FIG. 11, video encoder 20 receives a current video block within a video frame to be encoded. In the example of FIG. 11, video encoder 20 includes mode select unit 40, a video data memory 41, decoded picture buffer 64, summer 50, transform processing unit 52, quantization unit 54, and entropy encoding unit 56. Mode select unit 40, in turn, includes motion compensation unit 44, motion estimation unit 42, intra-prediction unit 46, partition unit 48, and color prediction processing unit 66. For video block reconstruction, video encoder 20 also includes inverse quantization unit 58, inverse transform processing unit 60, and summer 62. A deblocking filter (not shown in FIG. 11) may also be included to filter block boundaries to remove blockiness artifacts from reconstructed video. If desired, the deblocking filter would typically filter the output of summer 62. Additional filters (in loop or post loop) may also be used in addition to the deblocking filter. Such filters are not shown for brevity, but if desired, may filter the output of summer 50 (as an in-loop filter).

[0117] Video data memory 41 may store video data to be encoded by the components of video encoder 20. The video data stored in video data memory 41 may be obtained, for example, from video source 18. Decoded picture buffer 64 may be a reference picture memory that stores reference video data for use in encoding video data by video encoder 20, e.g., in intra- or inter-coding modes. Video data memory 41 and decoded picture buffer 64 may be formed by any of a variety of memory devices, such as dynamic random access memory (DRAM), including synchronous DRAM (SDRAM),

magnetoresistive RAM (MRAM), resistive RAM (RRAM), or other types of memory devices. Video data memory 41 and decoded picture buffer 64 may be provided by the same memory device or separate memory devices. In various examples, video data memory 41 may be on-chip with other components of video encoder 20, or off-chip relative to those components.

[0118] During the encoding process, video encoder 20 receives a video frame or slice to be coded. The frame or slice may be divided into multiple video blocks. Motion estimation unit 42 and motion compensation unit 44 perform inter-predictive coding of the received video block relative to one or more blocks in one or more reference frames to provide temporal prediction. Intra-prediction unit 46 may alternatively perform intra-predictive coding of the received video block relative to one or more neighboring blocks in the same frame or slice as the block to be coded to provide spatial prediction. Video encoder 20 may perform multiple coding passes, e.g., to select an appropriate coding mode for each block of video data.

[0119] Moreover, partition unit 48 may partition blocks of video data into sub-blocks, based on evaluation of previous partitioning schemes in previous coding passes. For example, partition unit 48 may initially partition a frame or slice into LCUs, and partition each of the LCUs into sub-CUs based on rate-distortion analysis (e.g., rate-distortion optimization). Mode select unit 40 may further produce a quadtree data structure indicative of partitioning of an LCU into sub-CUs. Leaf-node CUs of the quadtree may include one or more PUs and one or more TUs.

[0120] Mode select unit 40 may select one of the coding modes, intra or inter, e.g., based on error results, and provides the resulting intra- or inter-coded block to summer 50 to generate residual block data and to summer 62 to reconstruct the encoded block for use as a reference frame. Mode select unit 40 also provides syntax elements, such as motion vectors, intra-mode indicators, partition information, and other such syntax information, to entropy encoding unit 56.

[0121] Motion estimation unit 42 and motion compensation unit 44 may be highly integrated, but are illustrated separately for conceptual purposes. Motion estimation, performed by motion estimation unit 42, is the process of generating motion vectors, which estimate motion for video blocks. A motion vector, for example, may indicate the displacement of a PU of a video block within a current video frame or picture relative to a predictive block within a reference picture (or other coded unit) relative to the current block being coded within the current picture (or other coded unit). A

predictive block is a block that is found to closely match the block to be coded, in terms of pixel difference, which may be determined by sum of absolute difference (SAD), sum of square difference (SSD), or other difference metrics. In some examples, video encoder 20 may calculate values for sub-integer pixel positions of reference pictures stored in decoded picture buffer 64. For example, video encoder 20 may interpolate values of one-quarter pixel positions, one-eighth pixel positions, or other fractional pixel positions of the reference picture. Therefore, motion estimation unit 42 may perform a motion search relative to the full pixel positions and fractional pixel positions and output a motion vector with fractional pixel precision.

[0122] Motion estimation unit 42 calculates a motion vector for a PU of a video block in an inter-coded slice by comparing the position of the PU to the position of a predictive block of a reference picture. The reference picture may be selected from a first reference picture list (List 0) or a second reference picture list (List 1), each of which identify one or more reference pictures stored in decoded picture buffer 64. Motion estimation unit 42 sends the calculated motion vector to entropy encoding unit 56 and motion compensation unit 44.

[0123] Motion compensation, performed by motion compensation unit 44, may involve fetching or generating the predictive block based on the motion vector determined by motion estimation unit 42. Again, motion estimation unit 42 and motion compensation unit 44 may be functionally integrated, in some examples. Upon receiving the motion vector for the PU of the current video block, motion compensation unit 44 may locate the predictive block to which the motion vector points in one of the reference picture lists. Summer 50 forms a residual video block by subtracting pixel values of the predictive block from the pixel values of the current video block being coded, forming pixel difference values, as discussed below. In general, motion estimation unit 42 performs motion estimation relative to luma components, and motion compensation unit 44 uses motion vectors calculated based on the luma components for both chroma components and luma components. Mode select unit 40 may also generate syntax elements associated with the video blocks and the video slice for use by video decoder 30 in decoding the video blocks of the video slice.

[0124] Intra-prediction unit 46 may intra-predict a current block, as an alternative to the inter-prediction performed by motion estimation unit 42 and motion compensation unit 44, as described above. In particular, intra-prediction unit 46 may determine an intra-prediction mode to use to encode a current block. In some examples, intra-

prediction unit 46 may encode a current block using various intra-prediction modes, e.g., during separate encoding passes, and intra-prediction unit 46 (or mode select unit 40, in some examples) may select an appropriate intra-prediction mode to use from the tested modes.

[0125] For example, intra-prediction unit 46 may calculate rate-distortion values using a rate-distortion analysis for the various tested intra-prediction modes, and select the intra-prediction mode having the best rate-distortion characteristics among the tested modes. Rate-distortion analysis generally determines an amount of distortion (or error) between an encoded block and an original, unencoded block that was encoded to produce the encoded block, as well as a bit rate (that is, a number of bits) used to produce the encoded block. Intra-prediction unit 46 may calculate ratios from the distortions and rates for the various encoded blocks to determine which intra-prediction mode exhibits the best rate-distortion value for the block.

[0126] After selecting an intra-prediction mode for a block, intra-prediction unit 46 may provide information indicative of the selected intra-prediction mode for the block to entropy encoding unit 56. Entropy encoding unit 56 may encode the information indicating the selected intra-prediction mode. Video encoder 20 may include in the transmitted bitstream configuration data, which may include a plurality of intra-prediction mode index tables and a plurality of modified intra-prediction mode index tables (also referred to as codeword mapping tables), definitions of encoding contexts for various blocks, and indications of a most probable intra-prediction mode, an intra-prediction mode index table, and a modified intra-prediction mode index table to use for each of the contexts.

[0127] Video encoder 20 forms a residual video block by subtracting the prediction data from mode select unit 40 from the original video block being coded. Summer 50 represents the component or components that perform this subtraction operation. Transform processing unit 52 applies a transform, such as a discrete cosine transform (DCT) or a conceptually similar transform, to the residual block, producing a video block comprising residual transform coefficient values. Transform processing unit 52 may perform other transforms which are conceptually similar to DCT. Wavelet transforms, integer transforms, sub-band transforms or other types of transforms could also be used. In any case, transform processing unit 52 applies the transform to the residual block, producing a block of residual transform coefficients. The transform may convert the residual information from a pixel value domain to a transform domain, such

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

[0128] Following quantization, entropy encoding unit 56 entropy codes the quantized transform coefficients. For example, entropy encoding unit 56 may perform context adaptive variable length coding (CAVLC), context adaptive binary arithmetic coding (CABAC), syntax-based context-adaptive binary arithmetic coding (SBAC), probability interval partitioning entropy (PIPE) coding or another entropy coding technique. In the case of context-based entropy coding, context may be based on neighboring blocks. Following the entropy coding by entropy encoding unit 56, the encoded bitstream may be transmitted to another device (e.g., video decoder 30) or archived for later transmission or retrieval.

[0129] Inverse quantization unit 58 and inverse transform processing unit 60 apply inverse quantization and inverse transformation, respectively, to reconstruct the residual block in the pixel domain, e.g., for later use as a reference block. Motion compensation unit 44 may calculate a reference block by adding the residual block to a predictive block of one of the frames of decoded picture buffer 64. Motion compensation unit 44 may also apply one or more interpolation filters to the reconstructed residual block to calculate sub-integer pixel values for use in motion estimation. Summer 62 adds the reconstructed residual block to the motion compensated prediction block produced by motion compensation unit 44 to produce a reconstructed video block for storage in decoded picture buffer 64. The reconstructed video block may be used by motion estimation unit 42 and motion compensation unit 44 as a reference block to inter-code a block in a subsequent video frame.

[0130] According to the techniques described in this disclosure, video encoder 20 is configured to perform 3D lookup table based color gamut scalability when encoding multi-layer video data. Video encoder 20 may predict and encode multi-layer video data in accordance any of the SHVC extension, the MV-HEVC extension, and the 3D-HEVC extension, or other multi-layer video coding extensions. Specifically, color prediction processing unit 66 of video encoder 20 may generate inter-layer reference

pictures used to predict video blocks in a picture of a higher layer of the video data when a color gamut for the higher layer of the video data is different than a color gamut for a lower layer of video data.

[0131] Color prediction processing unit 66 of video encoder 20 may perform color prediction using a 3D lookup table for color gamut scalability to convert color data of a reference picture in a first color gamut for the lower layer of the video data to a second color gamut for the higher layer of the video data. In some examples, color prediction processing unit 66 may generate a separate 3D lookup table may be generated for each of the color components, i.e., a luma component, a first chroma component and a second chroma component. Each of the 3D lookup tables includes a luma dimension, a first chroma dimension and a second chroma dimension, and is indexed using the three independent color components.

[0132] According to the techniques described in this disclosure, color prediction processing unit 66 of video encoder 20 generates at least one 3D lookup table having a size that is different for the luma component than for each of the first chroma component and the second chroma component. Color prediction processing unit 66 may generate this asymmetric 3D lookup table using a different number segments for the luma dimension of the 3D lookup table. In one example, color prediction processing unit 66 may generate the 3D lookup table to have a larger size, i.e., more segments, for the luma component than for each of the first and second chroma components. In this way, the total size of the table may be reduced while maintaining good coding performance with higher resolution for the luma component than for the first and second chroma components.

[0133] In some examples, color prediction processing unit 66 may generate each of the 3D lookup tables to have the same size, in which a size of the luma dimension of the 3D lookup table is different than each of a size of the first chroma dimension of the 3D lookup table and a size of the second chroma dimension of the 3D lookup table. In other examples, color prediction processing unit 66 may generate a luma component 3D lookup table to have a first size that is different than each of a second size of a first chroma component 3D lookup table and a third size of a second chroma component 3D lookup table. In a further example, color prediction processing unit 66 may generate a luma component 3D lookup table to have a different precision value than each of the first chroma component 3D lookup table and the second chroma component 3D lookup table. In some cases, to further reduce complexity, color prediction processing unit 66

may only generate a luma component 3D lookup table, perform luma component prediction using the luma component 3D lookup table, and perform first and second chroma component prediction using 1D linear mapping or piecewise linear mapping.

[0134] As another example, according to the techniques of this disclosure, color prediction processing unit 66 may generate an unbalanced 3D lookup table using more segments for the dimension of the 3D lookup table associated with the color component used as a table index for the 3D lookup table. For example, the luma component 3D lookup table may have a larger luma dimension than each of a first chroma dimension and a second chroma dimension based on the luma component being used as a table index for the luma component 3D lookup table. The sizes of the first and second chroma component 3D lookup tables may be similarly determined based on the respective one of the first or second chroma component being used at the table index. In this way, the total size of each of the tables may be reduced while maintaining good coding performance with higher resolution for the color component used as the table index.

[0135] Upon generating the 3D lookup table, color prediction processing unit 66 performs color prediction of a reference picture for the lower layer of the video data using the 3D lookup table, and generates an inter-layer reference picture for the higher layer of the video data based on the color predicted reference picture. Upon generating the inter-layer reference pictures, motion compensation unit 44 of video encoder 20 may operate as described above to predict video blocks in a picture of the higher layer of the video data based on the inter-layer reference pictures generated using the 3D lookup table. Video encoder 20 may then encode residual data of the predicted video blocks in a bitstream for transmission to video decoder 30. In some examples, video encoder 20 may also encode one or more syntax elements indicating the size of the 3D lookup table in the bitstream, where the size is different for the luma component than for the first and second chroma components.

[0136] FIG. 12 is a block diagram illustrating an example of video decoder 30 that may implement techniques for determining using 3D lookup table based color gamut scalability in multi-layer video coding. In the example of FIG. 12, video decoder 30 includes an entropy decoding unit 70, a video data memory 71, motion compensation unit 72, intra prediction processing unit 74, color prediction processing unit 86, inverse quantization unit 76, inverse transform processing unit 78, decoded picture buffer 82 and summer 80. Video decoder 30 may, in some examples, perform a decoding pass

generally reciprocal to the encoding pass described with respect to video encoder 20 (FIG. 11). Motion compensation unit 72 may generate prediction data based on motion vectors received from entropy decoding unit 70, while intra-prediction unit 74 may generate prediction data based on intra-prediction mode indicators received from entropy decoding unit 70.

[0137] Video data memory 71 may store video data, such as an encoded video bitstream, to be decoded by the components of video decoder 30. The video data stored in video data memory 71 may be obtained, for example, from computer-readable medium 16, e.g., from a local video source, such as a camera, via wired or wireless network communication of video data, or by accessing physical data storage media. Video data memory 71 may form a coded picture buffer (CPB) that stores encoded video data from an encoded video bitstream. Decoded picture buffer 82 may be a reference picture memory that stores reference video data for use in decoding video data by video decoder 30, e.g., in intra- or inter-coding modes. Video data memory 71 and decoded picture buffer 82 may be formed by any of a variety of memory devices, such as dynamic random access memory (DRAM), including synchronous DRAM (SDRAM), magnetoresistive RAM (MRAM), resistive RAM (RRAM), or other types of memory devices. Video data memory 71 and decoded picture buffer 82 may be provided by the same memory device or separate memory devices. In various examples, video data memory 71 may be on-chip with other components of video decoder 30, or off-chip relative to those components.

[0138] During the decoding process, video decoder 30 receives an encoded video bitstream that represents video blocks of an encoded video slice and associated syntax elements from video encoder 20. Entropy decoding unit 70 of video decoder 30 entropy decodes the bitstream to generate quantized coefficients, motion vectors or intra-prediction mode indicators, and other syntax elements. Entropy decoding unit 70 forwards the motion vectors to and other syntax elements to motion compensation unit 72. Video decoder 30 may receive the syntax elements at the video slice level and/or the video block level.

[0139] When the video slice is coded as an intra-coded (I) slice, intra prediction processing unit 74 may generate prediction data for a video block of the current video slice based on a signaled intra prediction mode and data from previously decoded blocks of the current frame or picture. When the video frame is coded as an inter-coded (i.e., B or P) slice, motion compensation unit 72 produces predictive blocks for a video block of

the current video slice based on the motion vectors and other syntax elements received from entropy decoding unit 70. The predictive blocks may be produced from one of the reference pictures within one of the reference picture lists. Video decoder 30 may construct the reference picture lists, List 0 and List 1, using default construction techniques based on reference pictures stored in decoded picture buffer 82. Motion compensation unit 72 determines prediction information for a video block of the current video slice by parsing the motion vectors and other syntax elements, and uses the prediction information to produce the predictive blocks for the current video block being decoded. For example, motion compensation unit 72 uses some of the received syntax elements to determine a prediction mode (e.g., intra- or inter-prediction) used to code the video blocks of the video slice, an inter-prediction slice type (e.g., B slice or P slice), construction information for one or more of the reference picture lists for the slice, motion vectors for each inter-encoded video block of the slice, inter-prediction status for each inter-coded video block of the slice, and other information to decode the video blocks in the current video slice.

[0140] Motion compensation unit 72 may also perform interpolation based on interpolation filters. Motion compensation unit 72 may use interpolation filters as used by video encoder 20 during encoding of the video blocks to calculate interpolated values for sub-integer pixels of reference blocks. In this case, motion compensation unit 72 may determine the interpolation filters used by video encoder 20 from the received syntax elements and use the interpolation filters to produce predictive blocks.

[0141] Inverse quantization unit 76 inverse quantizes, i.e., de-quantizes, the quantized transform coefficients provided in the bitstream and decoded by entropy decoding unit 70. The inverse quantization process may include use of a quantization parameter QP_Y calculated by video decoder 30 for each video block in the video slice to determine a degree of quantization and, likewise, a degree of inverse quantization that should be applied. Inverse transform processing unit 78 applies an inverse transform, e.g., an inverse DCT, an inverse integer transform, or a conceptually similar inverse transform process, to the transform coefficients in order to produce residual blocks in the pixel domain.

[0142] After motion compensation unit 72 generates the predictive block for the current video block based on the motion vectors and other syntax elements, video decoder 30 forms a decoded video block by summing the residual blocks from inverse transform processing unit 78 with the corresponding predictive blocks generated by motion

compensation unit 72. Summer 80 represents the component or components that perform this summation operation. If desired, a deblocking filter may also be applied to filter the decoded blocks in order to remove blockiness artifacts. Other loop filters (either in the coding loop or after the coding loop) may also be used to smooth pixel transitions, or otherwise improve the video quality. The decoded video blocks in a given frame or picture are then stored in decoded picture buffer 82, which stores reference pictures used for subsequent motion compensation. Decoded picture buffer 82 also stores decoded video for later presentation on a display device, such as display device 32 of FIG. 1.

[0143] According to the techniques described in this disclosure, video decoder 30 is configured to perform 3D lookup table based color gamut scalability when decoding multi-layer video data. Video decoder 30 may decode and reconstruct predicted multi-layer video data in accordance any of the SHVC extension, the MV-HEVC extension, the 3D-HEVC extension, or other multi-layer video coding extensions to HEVC. Specifically, color prediction processing unit 86 of video decoder 30 may generate inter-layer reference pictures used to predict video blocks in a picture of a higher layer of the video data when a color gamut for the higher layer of the video data is different than a color gamut for a lower layer of video data.

[0144] Color prediction processing unit 86 of video decoder 30 may perform color prediction using a 3D lookup table for color gamut scalability to convert color data of a reference picture in a first color gamut for the lower layer of the video data to a second color gamut for the higher layer of the video data. In some examples, color prediction processing unit 86 may generate a separate 3D lookup table may be generated for each of the color components, i.e., a luma component, a first chroma component and a second chroma component. Each of the 3D lookup tables includes a luma dimension, a first chroma dimension and a second chroma dimension, and is indexed using the three independent color components.

[0145] According to the techniques described in this disclosure, color prediction processing unit 86 of video decoder 30 generates at least one 3D lookup table having a size that is different for the luma component than for each of the first chroma component and the second chroma component. In some examples, video decoder 30 may decode one or more syntax elements indicating the size of the 3D lookup table in the bitstream, where the size is different for the luma component than for the first and second chroma components. Color prediction processing unit 86 may generate this

asymmetric 3D lookup table according to the indicated size using a different number of segments for the luma dimension of the 3D lookup table. In one example, color prediction processing unit 86 may generate the 3D lookup table to have a larger size, i.e., more segments, for the luma component than for each of the first and second chroma components. In this way, the total size of the table may be reduced while maintaining good coding performance with higher resolution for the luma component than for the first and second chroma components.

[0146] In some examples, color prediction processing unit 86 may generate each of the 3D lookup tables to have the same size, in which a size of the luma dimension of the 3D lookup table is different than each of a size of the first chroma dimension of the 3D lookup table and a size of the second chroma dimension of the 3D lookup table. In other examples, color prediction processing unit 86 may generate a luma component 3D lookup table to have a first size that is different than each of a second size of a first chroma component 3D lookup table and a third size of a second chroma component 3D lookup table. In a further example, color prediction processing unit 86 may generate a luma component 3D lookup table to have a different precision value than each of the first chroma component 3D lookup table and the second chroma component 3D lookup table. In some cases, to further reduce complexity, color prediction processing unit 86 may only generate a luma component 3D lookup table, perform luma component prediction using the luma component 3D lookup table, and perform first and second chroma component prediction using 1D linear mapping or piecewise linear mapping.

[0147] As another example, according to the techniques of this disclosure, color prediction processing unit 86 may generate an unbalanced 3D lookup table using more segments for the dimension of the 3D lookup table associated with the color component used as a table index for the 3D lookup table. For example, the luma component 3D lookup table may have a larger luma dimension than each of a first chroma dimension and a second chroma dimension based on the luma component being used as a table index for the luma component 3D lookup table. The sizes of the first and second chroma component 3D lookup tables may be similarly determined based on the respective one of the first or second chroma component being used at the table index. In this way, the total size of each of the tables may be reduced while maintaining good coding performance with higher resolution for the color component used as the table index.

[0148] Upon generating the 3D lookup table, color prediction processing unit 86 performs color prediction of a reference picture for the lower layer of the video data using the 3D lookup table, and generates an inter-layer reference picture for the higher layer of the video data based on the color predicted reference picture. Upon generating the inter-layer reference pictures, motion compensation unit 72 of video decoder 30 may operate as described above to reconstruct predicted video blocks in a picture of the higher layer of the video data based on decoded residual data and the inter-layer reference pictures generated using the 3D lookup table.

[0149] FIG. 13 is a flowchart illustrating an example operation of generating inter-layer reference pictures using 3D lookup table based color gamut scalability. The example operation of FIG. 13 is described herein as being performed by color prediction processing unit 86 of video decoder 30 of FIG. 12. In other examples, the operation may be performed by color prediction processing unit 66 of video encoder 20 of FIG. 11 or color prediction processing unit 144 of FIG. 8.

[0150] Conventionally, in multi-layer video coding, a lower layer of video data (e.g., a base layer) and a higher layer of the video data (e.g., an enhancement layer) include color data in the same color gamut, e.g., HD color gamut BT.709. In this case, a video decoder may generate inter-layer reference pictures for the higher layer of the video data as up-sampled versions of co-located reference pictures for the lower layer of the video data. In the examples described in this disclosure, a lower layer of video data may include color data in a first color gamut, e.g., BT.709, and a higher layer of the video data may include color data in a different, second color gamut, e.g., UHD color gamut BT.2020. In this example, in order to generate inter-layer reference pictures for the higher layer of the video data, video decoder 30 uses color prediction processing unit 86 to perform color prediction to convert the color data of a reference picture in the first color gamut for the lower layer of the video data to the second color gamut for the higher layer of the video data. Color prediction processing unit 86 of video decoder 30 may perform color prediction using a 3D lookup table for color gamut scalability.

[0151] According to the techniques of this disclosure, color prediction processing unit 86 generates at least one 3D lookup table for color gamut scalability having a different size for a luma component than for each of a first chroma component and a second chroma component (180). Entropy decoding unit 70 may decode one or more syntax elements indicating the size of the at least one 3D lookup table, where the size is different for the luma component than for each of the first and second chroma

components. Color prediction processing unit 86 may generate this asymmetric 3D lookup table according to the indicated size by using a different number of segments for the luma dimension of the 3D lookup table than for the first and second chroma components of the 3D lookup table.

[0152] For example, color prediction processing unit 86 may generate the 3D lookup table to have a larger size, i.e., more segments, for the luma component than for each of the first and second chroma components. In one example, the 3D lookup table may have a size up to 8x2x2. In this way, the total size of the table may be reduced while maintaining good coding performance with higher resolution for the luma component than for the first and second chroma components.

[0153] Color prediction processing unit 86 may generate a separate 3D lookup table for each of the color components, i.e., a luma (Y) component, a first chroma (U) component and a second chroma (V) component. Each of the 3D lookup tables includes a luma (Y) dimension, a first chroma (U) dimension and a second chroma (V) dimension, and is indexed using the three independent color components (Y, U, V). In some examples, color prediction processing unit 86 may generate each of the 3D lookup tables to have the same size MxNxK, in which the size (M) of the luma dimension of the 3D lookup table is different than each of the size (N) of the first chroma dimension of the 3D lookup table and the size (K) of the second chroma dimension of the 3D lookup table. For example, the size of the luma dimension may be larger than the sizes of the chroma dimensions, M > N and M > K, and the sizes of the chroma dimensions may be the same, N=K. In one example, each of the 3D lookup tables may have the same size of 8x2x2.

[0154] In other examples, color prediction processing unit 86 may generate a luma component 3D lookup table (LUT_Y) to have a first size, e.g., MxMxM or MxNxK, that is different than each of a second size of a first chroma component 3D lookup table (LUT_U), e.g., NxNxN, and a third size of a second chroma component 3D lookup table (LUT_V), e.g., KxKxK. For example, the size of the luma component 3D lookup table may be larger than the sizes of the chroma component 3D lookup tables. The first and second chroma component 3D lookup tables may be the same size or different sizes. In one example, LUT_Y may have the size 8x2x2, and each of LUT_U and LUT_V may have the size 2x2x2.

[0155] In a further example, color prediction processing unit 86 may generate the luma component 3D lookup table with a different precision value than the precision value of

each of the first chroma component 3D lookup table and the second chroma component 3D lookup table. For example, the precision value of the luma component 3D lookup table may be higher than the precision value of the chroma component 3D lookup tables. In some cases, to further reduce complexity, color prediction processing unit 86 may only generate a luma component 3D lookup table.

[0156] As another example, according to the techniques of this disclosure, color prediction processing unit 86 of video decoder 30 may generate the at least one 3D lookup table as an unbalanced 3D lookup table using more segments for the dimension of the 3D lookup table associated with the color component used as a table index for the 3D lookup table. For example, the luma component 3D lookup table may have a larger luma dimension than a first chroma dimension and a second chroma dimension based on the luma component being used as a table index for the luma component 3D lookup table. In this example, the LUT_Y may have the size MxNxN, where M > N. The sizes of the first and second chroma component 3D lookup tables may be similarly determined. For example, the LUT_U may have the size NxMxN based on the first chroma component being used at the table index, and the LUT_V may have the size NxNxM based on the second chroma component being used at the table index. In this way, the total size of each of the tables may be reduced while maintaining good coding performance with higher resolution for the color component used as the table index.

[0157] After generating the at least one 3D lookup table, color prediction processing unit 86 performs color prediction using the at least one 3D lookup table to convert the color data of the reference picture in the first color gamut for the lower layer of the video data to the second color gamut for the higher layer of the video data (182). Color prediction processing unit 86 then generates at least one inter-layer reference picture for the higher layer of the video data based on the converted color data (184). In cases where only a luma component 3D lookup table is generated, color prediction processing unit 86 may perform luma component prediction using the luma component 3D lookup table, and perform first and second chroma component prediction using 1D linear mapping or piecewise linear mapping to generate an inter-layer reference picture. Video decoder 30 may then reconstruct predicted video blocks in a picture of the higher layer of video data based on decoded residual data and the at least one inter-layer reference picture generated using the at least one 3D lookup table.

[0158] Certain aspects of this disclosure have been described with respect to extensions of the HEVC standard for purposes of illustration. However, the techniques described

in this disclosure may be useful for other video coding processes, including other standard or proprietary video coding processes not yet developed.

[0159] A video coder, as described in this disclosure, may refer to a video encoder or a video decoder. Similarly, a video coding unit may refer to a video encoder or a video decoder. Likewise, video coding may refer to video encoding or video decoding, as applicable.

[0160] It is to be recognized that depending on the example, certain acts or events of any of the techniques described herein can be performed in a different sequence, may be added, merged, or left out altogether (e.g., not all described acts or events are necessary for the practice of the techniques). Moreover, in certain examples, acts or events may be performed concurrently, e.g., through multi-threaded processing, interrupt processing, or multiple processors, rather than sequentially.

[0161] In one or more examples, the functions described may be implemented in hardware, software, firmware, or any combination thereof. If implemented in software, the functions may be stored on or transmitted over as one or more instructions or code on a computer-readable medium and executed by a hardware-based processing unit. Computer-readable media may include computer-readable storage media, which corresponds to a tangible medium such as data storage media, or communication media including any medium that facilitates transfer of a computer program from one place to another, e.g., according to a communication protocol. In this manner, computer-readable media generally may correspond to (1) tangible computer-readable storage media which is non-transitory or (2) a communication medium such as a signal or carrier wave. Data storage media may be any available media that can be accessed by one or more computers or one or more processors to retrieve instructions, code and/or data structures for implementation of the techniques described in this disclosure. A computer program product may include a computer-readable medium.

[0162] By way of example, and not limitation, such computer-readable storage media can comprise RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage, or other magnetic storage devices, flash memory, or any other medium that can be used to store desired program code in the form of instructions or data structures and that can be accessed by a computer. Also, any connection is properly termed a computer-readable medium. For example, if instructions are transmitted from a website, server, or other remote source using a coaxial cable, fiber optic cable, twisted pair, digital subscriber line (DSL), or wireless technologies such as infrared, radio, and

microwave, then the coaxial cable, fiber optic cable, twisted pair, DSL, or wireless technologies such as infrared, radio, and microwave are included in the definition of medium. It should be understood, however, that computer-readable storage media and data storage media do not include connections, carrier waves, signals, or other transitory media, but are instead directed to non-transitory, tangible storage media. Disk and disc, as used herein, includes compact disc (CD), laser disc, optical disc, digital versatile disc (DVD), floppy disk and Blu-ray disc, where disks usually reproduce data magnetically, while discs reproduce data optically with lasers. Combinations of the above should also be included within the scope of computer-readable media.

[0163] Instructions may be executed by one or more processors, such as one or more digital signal processors (DSPs), general purpose microprocessors, application specific integrated circuits (ASICs), field programmable logic arrays (FPGAs), or other equivalent integrated or discrete logic circuitry. Accordingly, the term "processor," as used herein may refer to any of the foregoing structure or any other structure suitable for implementation of the techniques described herein. In addition, in some aspects, the functionality described herein may be provided within dedicated hardware and/or software modules configured for encoding and decoding, or incorporated in a combined codec. Also, the techniques could be fully implemented in one or more circuits or logic elements.

[0164] The techniques of this disclosure may be implemented in a wide variety of devices or apparatuses, including a wireless handset, an integrated circuit (IC) or a set of ICs (e.g., a chip set). Various components, modules, or units are described in this disclosure to emphasize functional aspects of devices configured to perform the disclosed techniques, but do not necessarily require realization by different hardware units. Rather, as described above, various units may be combined in a codec hardware unit or provided by a collection of interoperative hardware units, including one or more processors as described above, in conjunction with suitable software and/or firmware.

[0165] Various examples have been described. These and other examples are within the scope of the following claims.

CLAIMS:

1. A method of processing multi-layer video data, the method comprising:

generating at least one three-dimensional (3D) lookup table for color gamut scalability, wherein the at least one 3D lookup table has a size that is different for a luma component than for each of a first chroma component and a second chroma component, wherein generating the at least one 3D lookup table includes partitioning the luma component to have a different number of segments than each of the first chroma component and the second chroma component, and wherein the number of segments for the luma component is indicated by one or more syntax elements included in a bitstream;

performing color prediction using the at least one 3D lookup table to convert color data of a reference picture in a first color gamut for a lower layer of the video data to a second color gamut for a higher layer of the video data, wherein the first color gamut and the second color gamut are within the same color space for the video data; and

generating at least one inter-layer reference picture for the higher layer of the video data based on the converted color data.

2. The method of claim 1, wherein the size of the at least one 3D lookup table is larger for the luma component than for each of the first chroma component and the second chroma component.

3. The method of claim 1, wherein the at least one 3D lookup table has more segments for the luma component than for each of the first chroma component and the second chroma component.

4. The method of claim 1, wherein generating the at least one 3D lookup table comprises generating a luma component 3D lookup table having a size M×N×K, wherein M represents a size for a luma dimension of the luma component 3D lookup table, N represents a size for a first chroma dimension of the luma component 3D lookup table, and K represents a size for a

Date Reçue/Date Received 2021-03-03

second chroma dimension of the luma component 3D lookup table, and wherein M is different than each of N and K.

5. The method of claim 4, wherein generating the at least one 3D lookup table further comprises generating a first chroma component 3D lookup table having the size M×N×K, and generating a second chroma component 3D lookup table having the size M×N×K.

6. The method of claim 1, wherein generating the at least one 3D lookup table comprises generating a luma component 3D lookup table having a first size, generating a first chroma component 3D lookup table having a second size, and generating a second chroma component 3D lookup table having a third size, wherein the first size is different than each of the second size and the third size.

7. The method of claim 1, wherein generating the at least one 3D lookup table comprises generating a luma component 3D lookup table having a first precision value, generating a first chroma component 3D lookup table having a second precision value, and generating a second chroma component 3D lookup table having the second precision value, wherein the first precision value is different than the second precision value.

8. The method of claim 1, wherein generating the at least one 3D lookup table comprises generating an unbalanced 3D lookup table using more segments for a luma dimension of the 3D lookup table than for each of a first chroma dimension and a second chroma dimension of the 3D lookup table based on the luma component being used as a table index for the 3D lookup table.

9. The method of claim 1, wherein generating the at least one 3D lookup table comprises:

generating a luma component 3D lookup table having a luma dimension that is larger than each of a first chroma dimension and a second chroma dimension based on the luma component being used as a table index for the luma component 3D lookup table;

generating a first chroma component 3D lookup table having a first chroma dimension that is larger than each of a luma dimension and a second chroma dimension based on the first

chroma component being used as a table index for the first chroma component 3D lookup table; and

generating a second chroma component 3D lookup table having a second chroma dimension that is larger than each of a first chroma dimension and a luma dimension based on the second chroma component being used as a table index for the second chroma component 3D lookup table.

10. The method of claim 1, wherein generating the at least one 3D lookup table comprises only generating a luma component 3D lookup table.

11. The method of claim 1, wherein performing color prediction comprises performing luma component prediction using the at least one 3D lookup table, the method further comprising performing first and second chroma component predictions using one of linear mapping or piecewise linear mapping.

12. The method of claim 1, further comprising:

predicting video blocks in a picture of the higher layer of the video data based on the at least one inter-layer reference picture generated using the at least one 3D lookup table;

encoding residual data of the predicted video blocks in the bitstream; and

encoding the one or more syntax elements indicating the number of segments for the luma component of the 3D lookup table in the bitstream.

13. The method of claim 1, further comprising:

decoding, from the bitstream, the one or more syntax elements indicating the number of segments for the luma component of the 3D lookup table;

decoding, from the bitstream, residual data of predicted video blocks; and

reconstructing the predicted video blocks in a picture of the higher layer of the video data based on the decoded residual data and the at least one inter-layer reference picture generated using the at least one 3D lookup table.

14. A video processing device for processing multi-layer video data, the device comprising:

a memory configured to store the multi-layer video data; and

one or more processors in communication with the memory and configured to:

generate at least one three-dimensional (3D) lookup table for color gamut scalability, wherein the at least one 3D lookup table has a size that is different for a luma component than for a first chroma component and a second chroma component, wherein the one or more processors are configured to partition the luma component to have a different number of segments than each of the first chroma component and the second chroma component, and wherein the number of segments for the luma component is indicated by one or more syntax elements included in a bitstream,

perform color prediction using the at least one 3D lookup table to convert color data of a reference picture in a first color gamut for a lower layer of the video data to a second color gamut for a higher layer of the video data, wherein the first color gamut and the second color gamut are within the same color space for the video data, and

generate at least one inter-layer reference picture for the higher layer of the video data based on the converted color data.

15. The device of claim 14, wherein the size of the at least one 3D lookup table is larger for the luma component than for each of the first chroma component and the second chroma component.

16. The device of claim 14, wherein the at least one 3D lookup table has more segments for the luma component than for each of the first chroma component and the second chroma component.

17. The device of claim 14, wherein, to generate the at least one 3D lookup table, the one or more processors are configured to generate a luma component 3D lookup table having a size M×N×K, wherein M represents a size for a luma dimension of the luma component 3D lookup table, N represents a size for a first chroma dimension of the luma component 3D lookup table, and K represents a size for a second chroma dimension of the luma component 3D lookup table, and wherein M is different than each of N and K.

18. The device of claim 17, wherein the one or more processors are configured to generate a first chroma component 3D lookup table having the size M×N×K, and generate a second chroma component 3D lookup table having the size M×N×K.

19. The device of claim 14, wherein, to generate the at least one 3D lookup table, the one or more processors are configured to generate a luma component 3D lookup table having a first size, generate a first chroma component 3D lookup table having a second size, and generate a second chroma component 3D lookup table having a third size, wherein the first size is different than each of the second size and the third size.

20. The device of claim 14, wherein, to generate the at least one 3D lookup table, the one or more processors are configured to generate a luma component 3D lookup table having a first precision value, generate a first chroma component 3D lookup table having a second precision value, and generate a second chroma component 3D lookup table having the second precision value, wherein the first precision value is different than the second precision value.

21. The device of claim 14, wherein, to generate the at least one 3D lookup table, the one or more processors are configured to generate an unbalanced 3D lookup table using more segments for a luma dimension of the 3D lookup table than for each of a first chroma dimension and a second chroma dimension of the 3D lookup table based on the luma component being used as a table index for the 3D lookup table.

22. The device of claim 14, wherein, to generate the at least one 3D lookup table, the one or more processors are configured to:

generate a luma component 3D lookup table having a luma dimension that is larger than each of a first chroma dimension and a second chroma dimension based on the luma component being used as a table index for the luma component 3D lookup table;

generate a first chroma component 3D lookup table having a first chroma dimension that is larger than each of a luma dimension and a second chroma dimension based on the first chroma component being used as a table index for the first chroma component 3D lookup table; and

generate a second chroma component 3D lookup table having a second chroma dimension that is larger than each of a first chroma dimension and a luma dimension based on the second chroma component being used as a table index for the second chroma component 3D lookup table.

23. The device of claim 14, wherein, to generate the at least one 3D lookup table, the one or more processors are configured to only generate a luma component 3D lookup table.

24. The device of claim 14, wherein the one or more processors are configured to perform luma component prediction using the at least one 3D lookup table, and perform first and second chroma component predictions using one of linear mapping or piecewise linear mapping.

25. The device of claim 14, wherein the video processing device comprises a video encoding device, and wherein the one or more processors are configured to:

predict video blocks in a picture of the higher layer of the video data based on the at least one inter-layer reference picture generated using the at least one 3D lookup table;

encode residual data of the predicted video blocks in the bitstream; and

encode the one or more syntax elements indicating the number of segments for the luma component of the 3D lookup table in the bitstream.

26. The device of claim 14, wherein the video processing device comprises a video decoding device, and wherein the one or more processors are configured to:

decode, from the bitstream, the one or more syntax elements indicating the number of segments for the luma component of the 3D lookup table;

decode, from the bitstream, residual data of predicted video blocks; and

reconstruct the predicted video blocks in a picture of the higher layer of the video data based on the decoded residual data and the at least one inter-layer reference picture generated using the at least one 3D lookup table.

27. A video processing device for processing multi-layer video data, the device comprising:

means for generating at least one three-dimensional (3D) lookup table for color gamut scalability, wherein the at least one 3D lookup table has a size that is different for a luma component than for a first chroma component and a second chroma component, wherein the means for generating the at least one 3D lookup table includes means for partitioning the luma component to have a different number of segments than each of the first chroma component and the second chroma component, and wherein the number of segments for the luma component is indicated by one or more syntax elements included in a bitstream;

means for performing color prediction using the at least one 3D lookup table to convert color data of a reference picture in a first color gamut for a lower layer of the video data to a second color gamut for a higher layer of the video data, wherein the first color gamut and the second color gamut are within the same color space for the video data; and

means for generating at least one inter-layer reference picture for the higher layer of the video data based on the converted color data.

28. A non-transitory computer-readable storage medium storing instructions for processing multi-layer video data that, when executed, cause one or more processors to:

generate at least one three-dimensional (3D) lookup table for color gamut scalability, wherein the at least one 3D lookup table has a size that is different for a luma component than for a first chroma component and a second chroma component, wherein the instructions cause the one or more processors to partition the luma component to have a different number of segments than each of the first chroma component and the second chroma component, and wherein the number of segments for the luma component is indicated by one or more syntax elements included in a bitstream;

perform color prediction using the at least one 3D lookup table to convert color data of a reference picture in a first color gamut for a lower layer of the video data to a second color gamut for a higher layer of the video data, wherein the first color gamut and the second color gamut are within the same color space for the video data; and

generate at least one inter-layer reference picture for the higher layer of the video data based on the converted color data.

Date Reçue/Date Received 2021-03-03

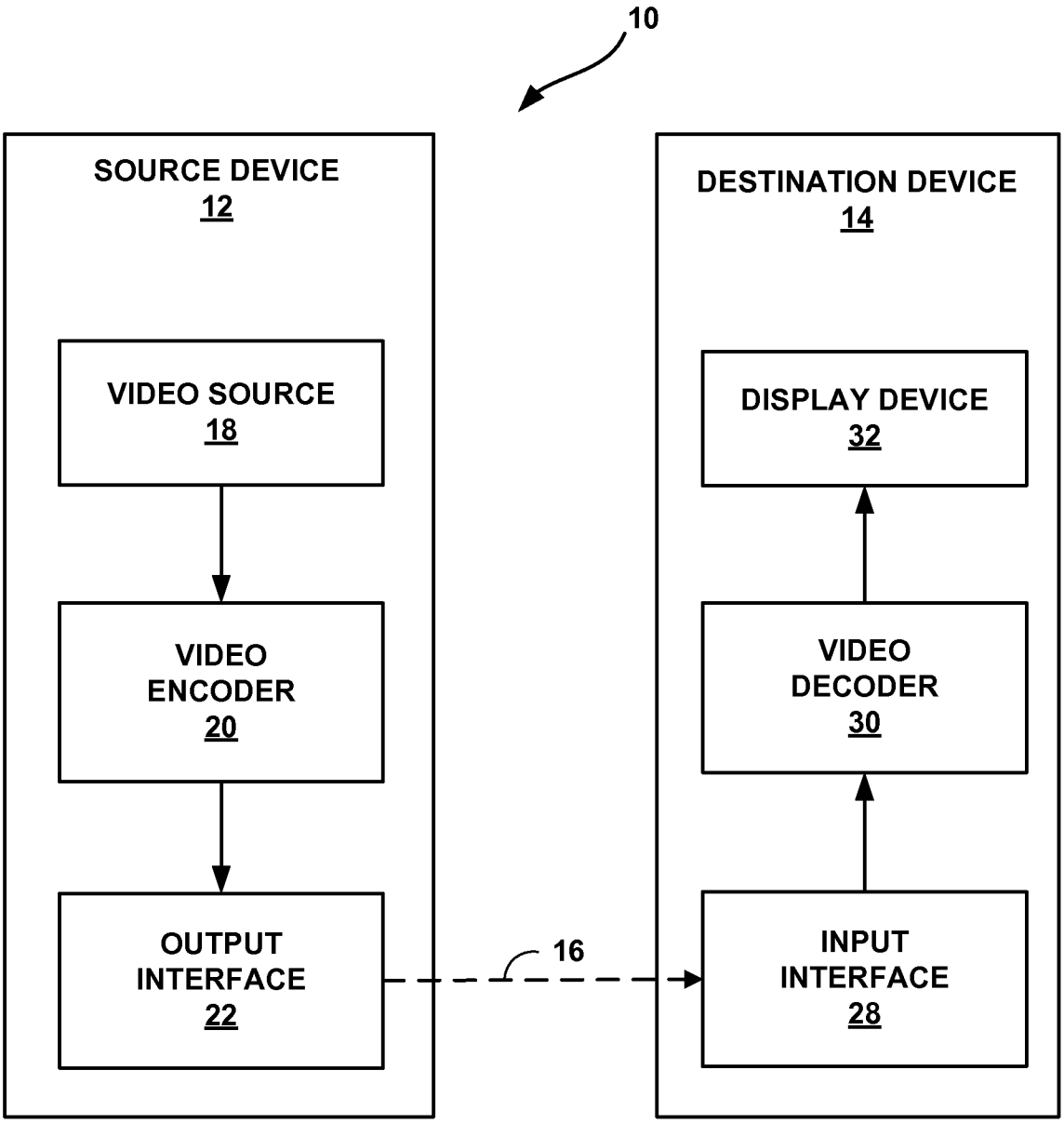

10
SOURCE DEVICE
12
VIDEO SOURCE
18
VIDEO
ENCODER
20
OUTPUT
INTERFACE
22
16
DESTINATION DEVICE
14
DISPLAY DEVICE
32
VIDEO
DECODER
30
INPUT
INTERFACE
28

FIG. 1

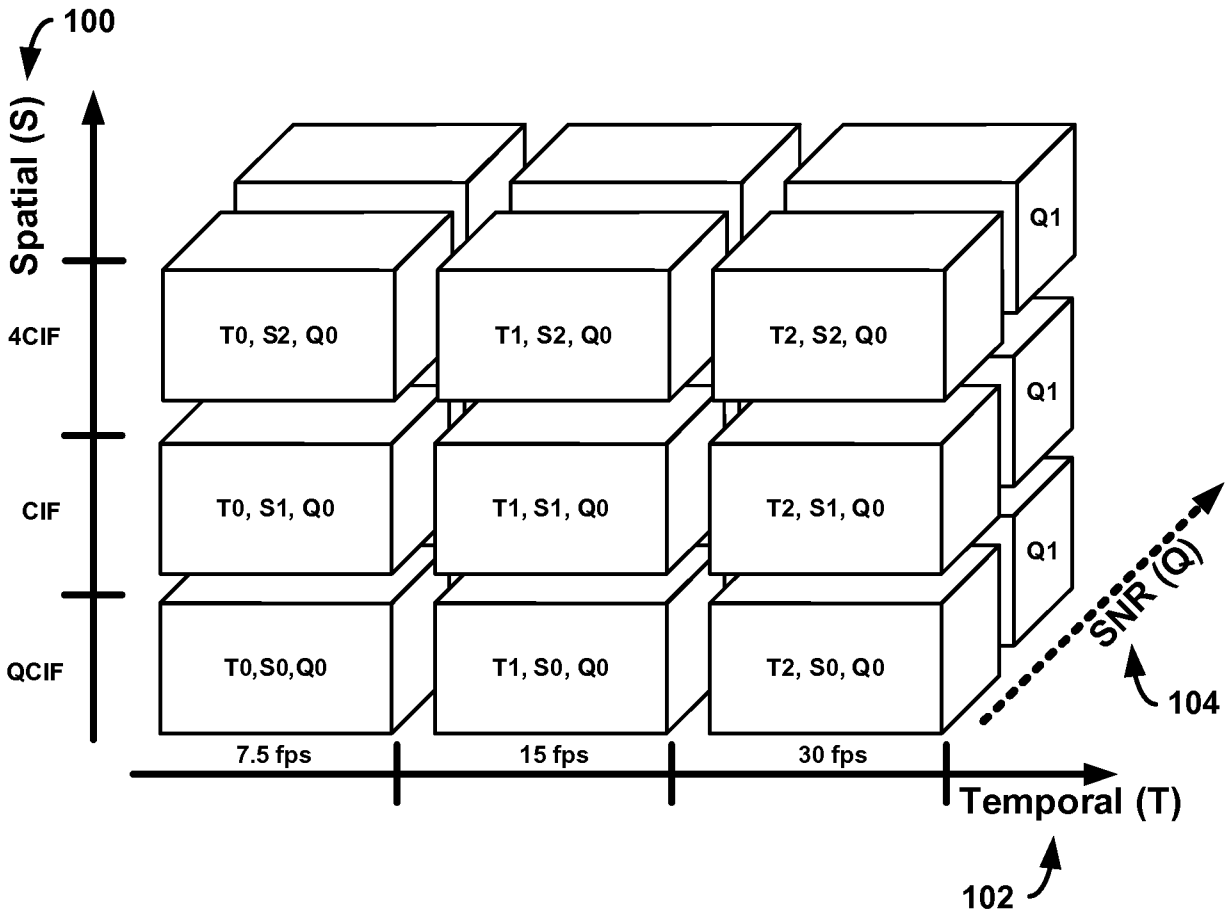

100
Spatial (S)
4CIF
CIF
QCIF
T0, S2, Q0
T1, S2, Q0
T2, S2, Q0
T0, S1, Q0
T1, S1, Q0
T2, S1, Q0
T0,S0,Q0
T1, S0, Q0
T2, S0, Q0
Q1
Q1
Q1
SNR (Q)
104
7.5 fps
15 fps
30 fps
Temporal (T)
102

FIG. 2

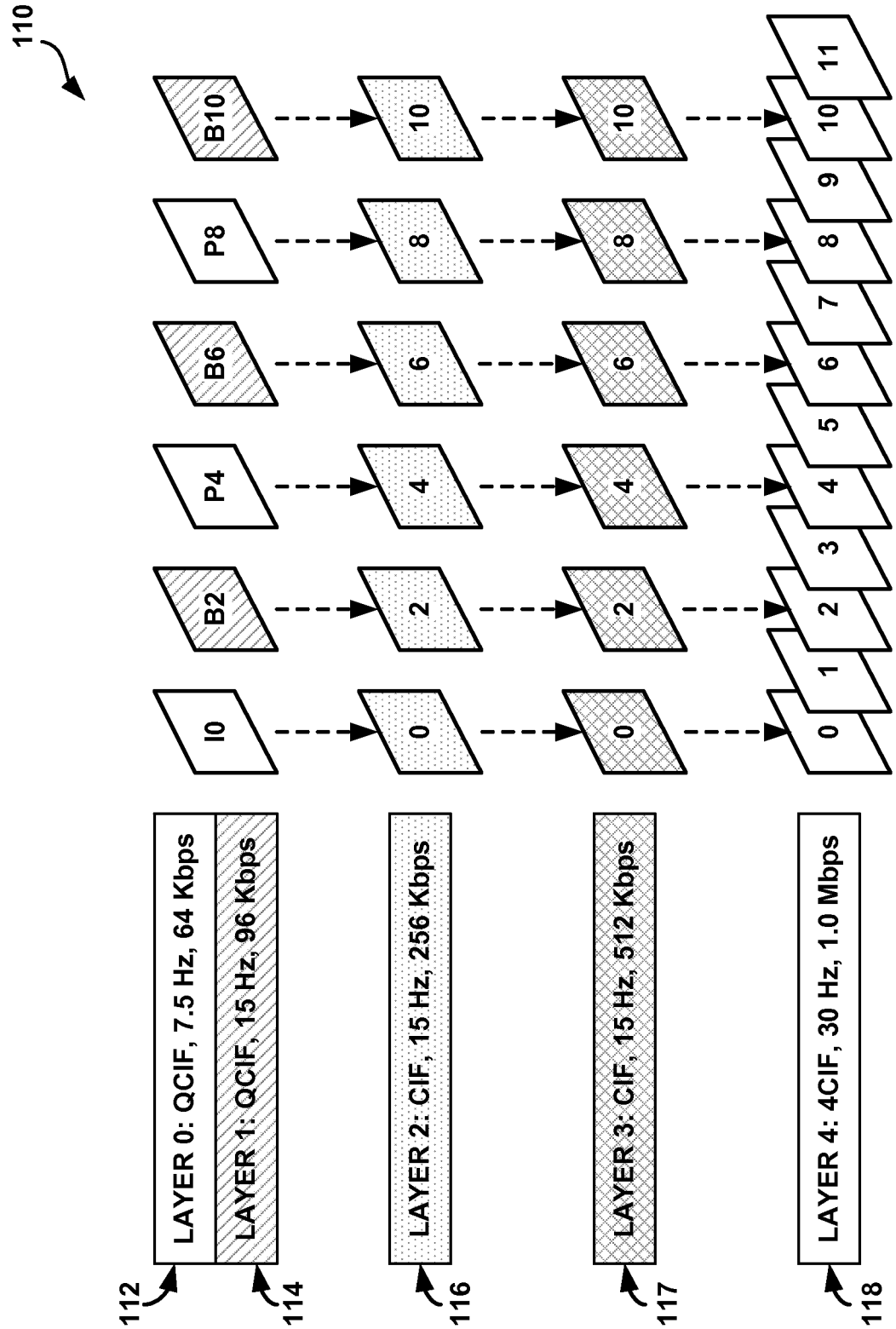

110
I0
B2
P4
B6
P8
B10
0
2
4
6
8
10
0
2
4
6
8
10
0
1
2
3
4
5
6
7
8
9
10
11
LAYER 0: QCIF, 7.5 Hz, 64 Kbps
LAYER 1: QCIF, 15 Hz, 96 Kbps
LAYER 2: CIF, 15 Hz, 256 Kbps
LAYER 3: CIF, 15 Hz, 512 Kbps
LAYER 4: 4CIF, 30 Hz, 1.0 Mbps
112
114
116
117
118

FIG. 3

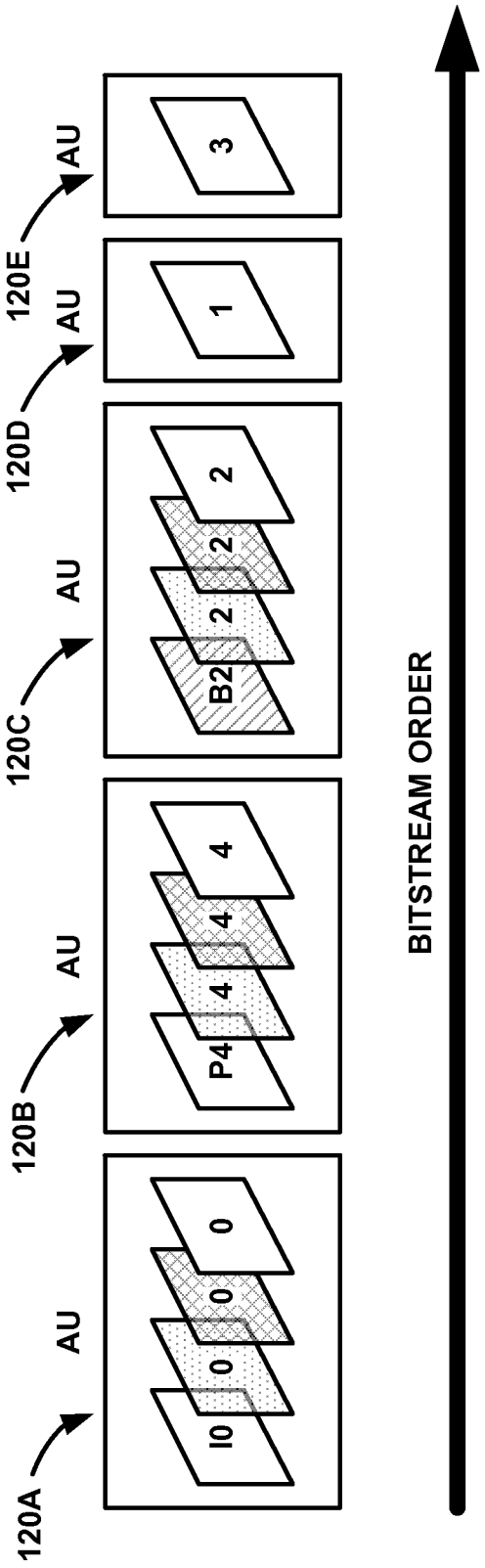

120A
AU
I0
0
0
0
120B
AU
P4
4
4
4
120C
AU
B2
2
2
2
120D
AU
1
120E
AU
3
BITSTREAM ORDER

FIG. 4

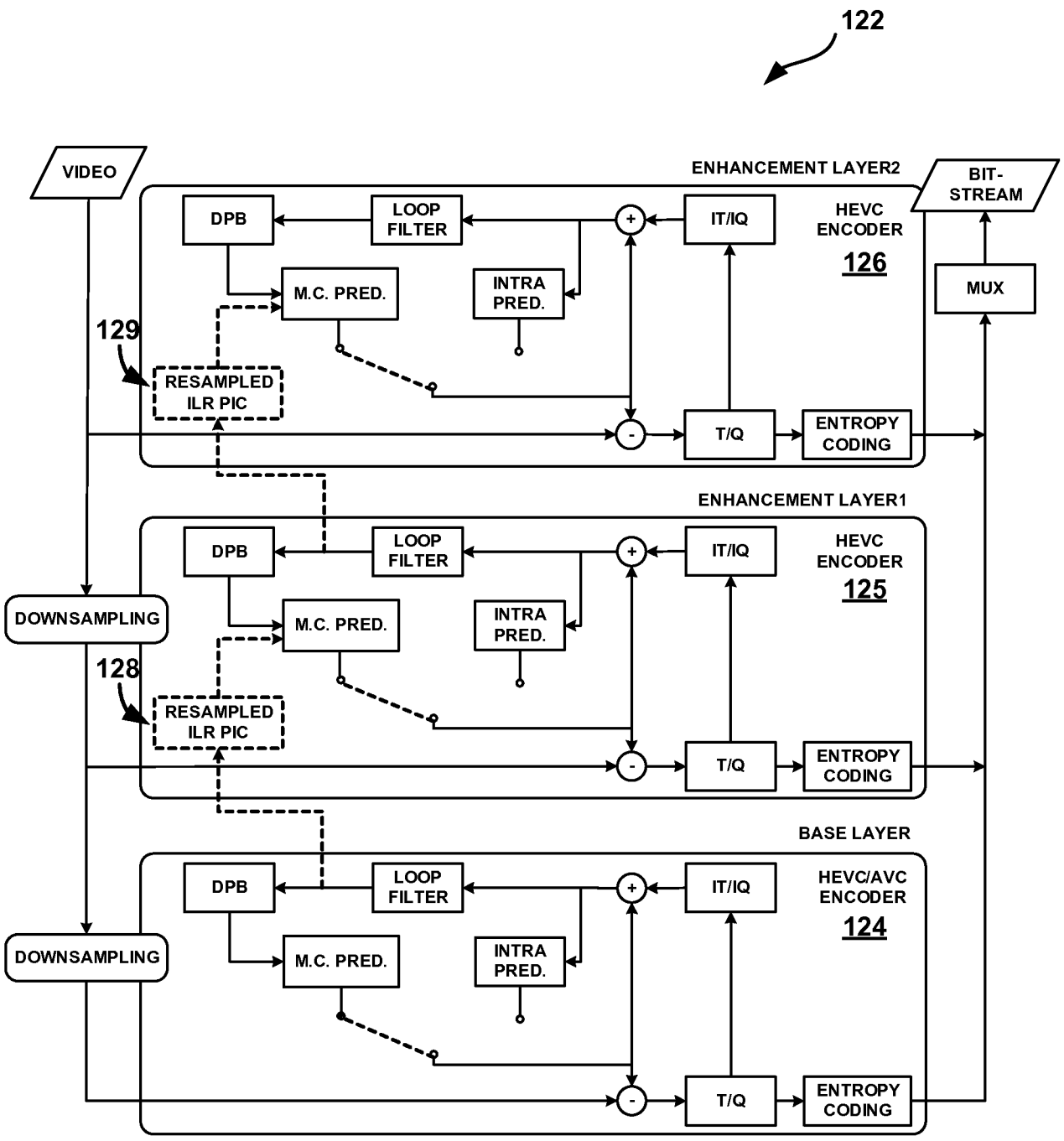

122
VIDEO
ENHANCEMENT LAYER2
BIT-STREAM
DPB
LOOP FILTER
IT/IQ
HEVC ENCODER
126
M.C. PRED.
INTRA PRED.
MUX
129
RESAMPLED ILR PIC
T/Q
ENTROPY CODING
ENHANCEMENT LAYER1
DOWNSAMPLING
128
HEVC ENCODER
125
BASE LAYER
HEVC/AVC ENCODER
124

FIG. 5

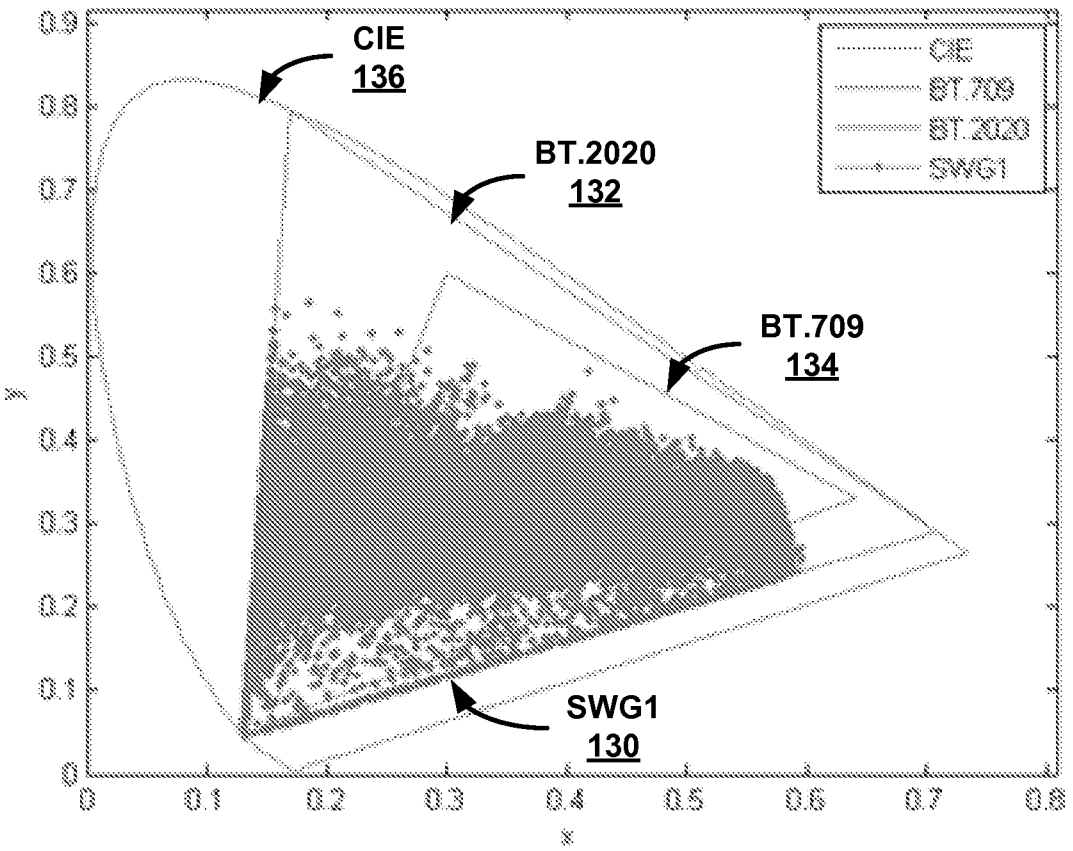

CIE
136
BT.2020
132
BT.709
134
SWG1
130
CIE
BT.709
BT.2020
SWG1
y
x
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

FIG. 6

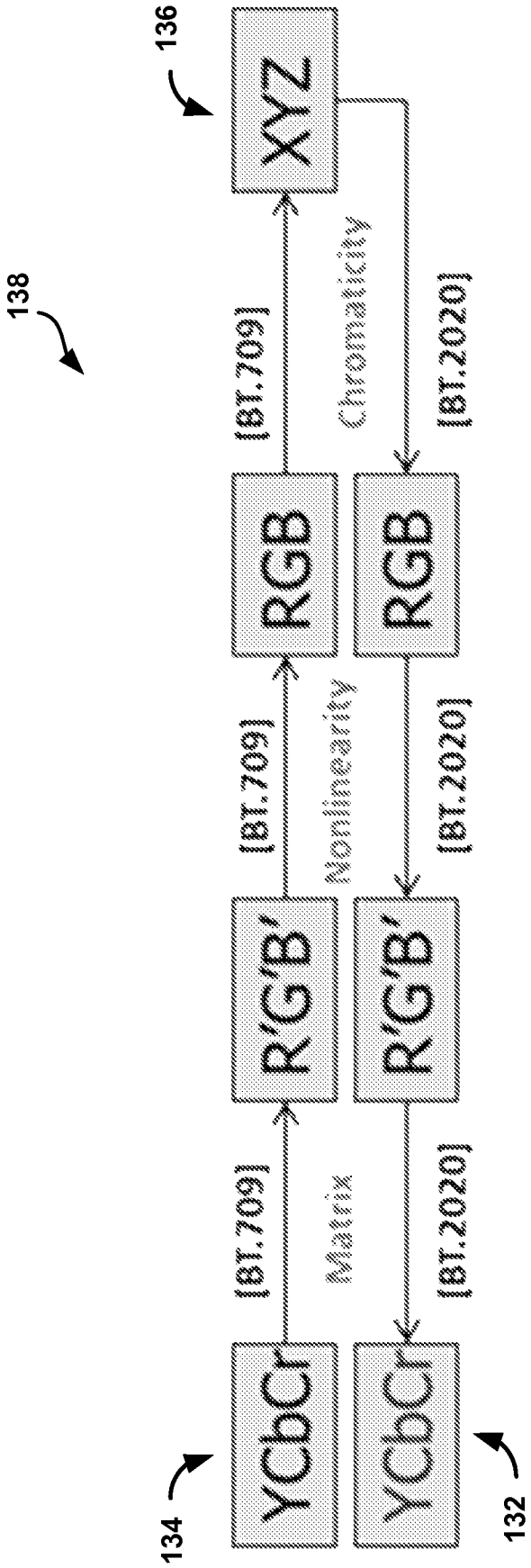

136
XYZ
138
[BT.709]
Chromaticity
[BT.2020]
RGB
RGB
[BT.709]
Nonlinearity
[BT.2020]
R'G'B'
R'G'B'
[BT.709]
Matrix
[BT.2020]
YCbCr
YCbCr
134
132

FIG. 7

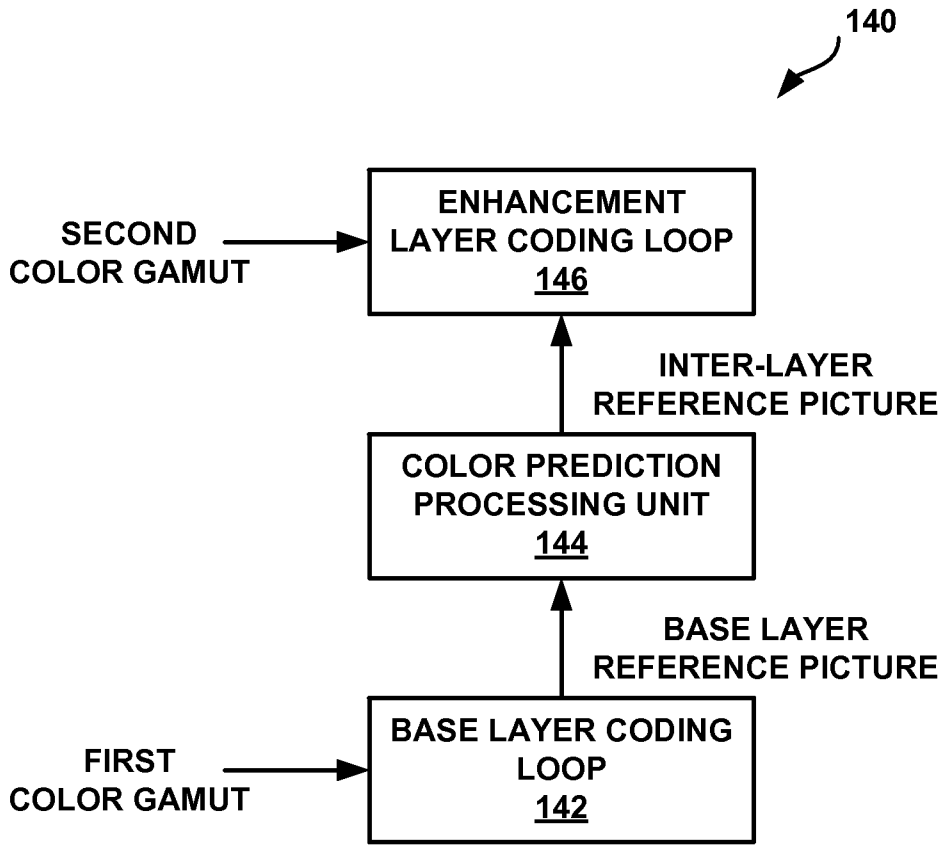

140
SECOND COLOR GAMUT
ENHANCEMENT LAYER CODING LOOP 146
INTER-LAYER REFERENCE PICTURE
COLOR PREDICTION PROCESSING UNIT 144
BASE LAYER REFERENCE PICTURE
FIRST COLOR GAMUT
BASE LAYER CODING LOOP 142

FIG. 8

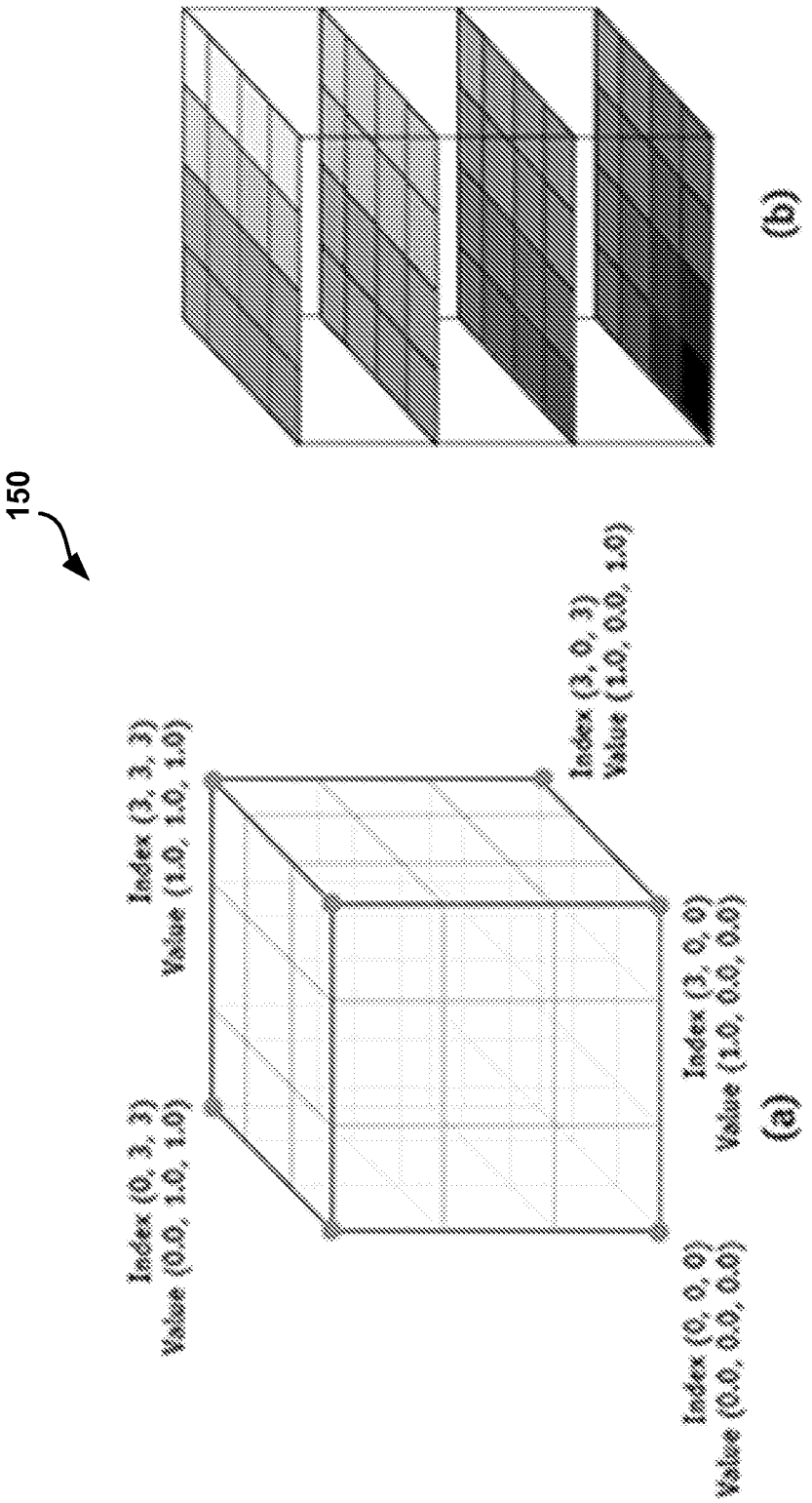

150
(b)
(a)
Index (3, 3, 3)
Value (1.0, 1.0, 1.0)
Index (3, 0, 3)
Value (1.0, 0.0, 1.0)
Index (0, 3, 3)
Value (0.0, 1.0, 1.0)
Index (3, 0, 0)
Value (1.0, 0.0, 0.0)
Index (0, 0, 0)
Value (0.0, 0.0, 0.0)

FIG. 9

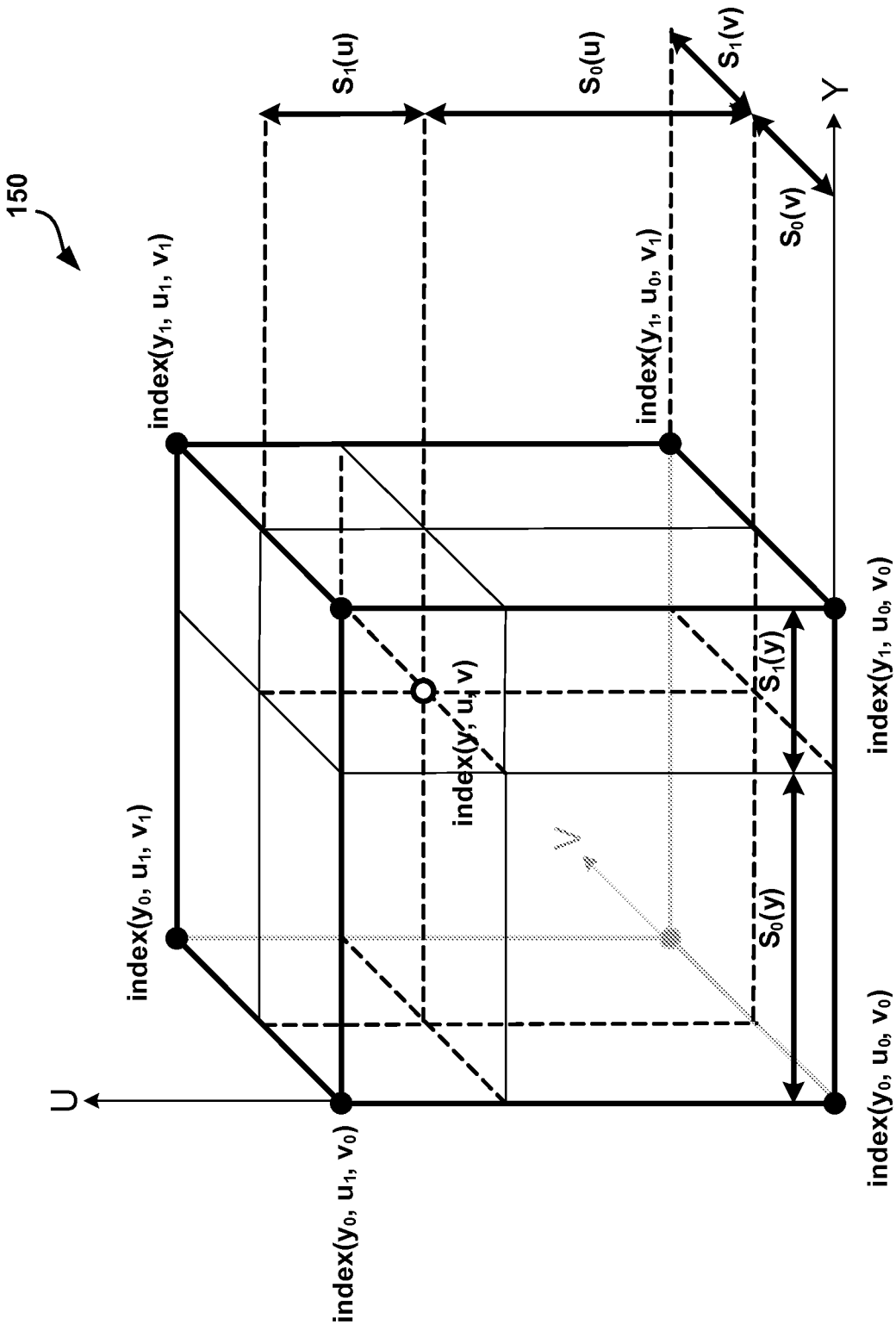

150
$S_1(u)$
$S_0(u)$
$S_1(v)$
$S_0(v)$
Y
U
V
index(y_1, u_1, v_1)
index(y_1, u_0, v_1)
index(y_1, u_0, v_0)
index(y, u, v)
$S_1(y)$
$S_0(y)$
index(y_0, u_1, v_1)
index(y_0, u_1, v_0)
index(y_0, u_0, v_0)

FIG. 10

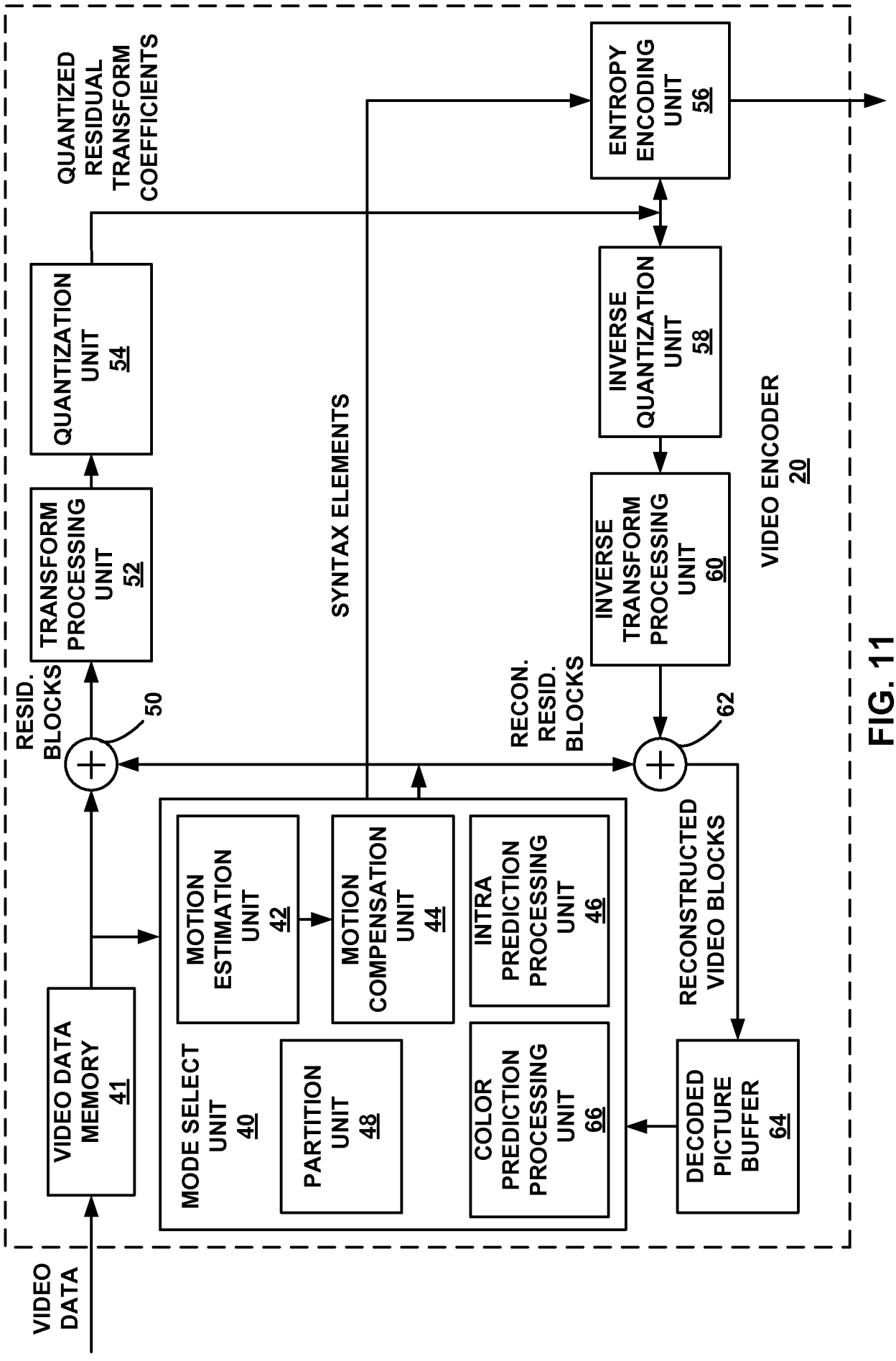

VIDEO DATA
VIDEO DATA MEMORY 41
MODE SELECT UNIT 40
MOTION ESTIMATION UNIT 42
MOTION COMPENSATION UNIT 44
INTRA PREDICTION PROCESSING UNIT 46
PARTITION UNIT 48
COLOR PREDICTION PROCESSING UNIT 66
DECODED PICTURE BUFFER 64
RESID. BLOCKS
50
TRANSFORM PROCESSING UNIT 52
QUANTIZATION UNIT 54
QUANTIZED RESIDUAL TRANSFORM COEFFICIENTS
SYNTAX ELEMENTS
ENTROPY ENCODING UNIT 56
INVERSE QUANTIZATION UNIT 58
INVERSE TRANSFORM PROCESSING UNIT 60
RECON. RESID. BLOCKS
62
RECONSTRUCTED VIDEO BLOCKS
VIDEO ENCODER 20

FIG. 11

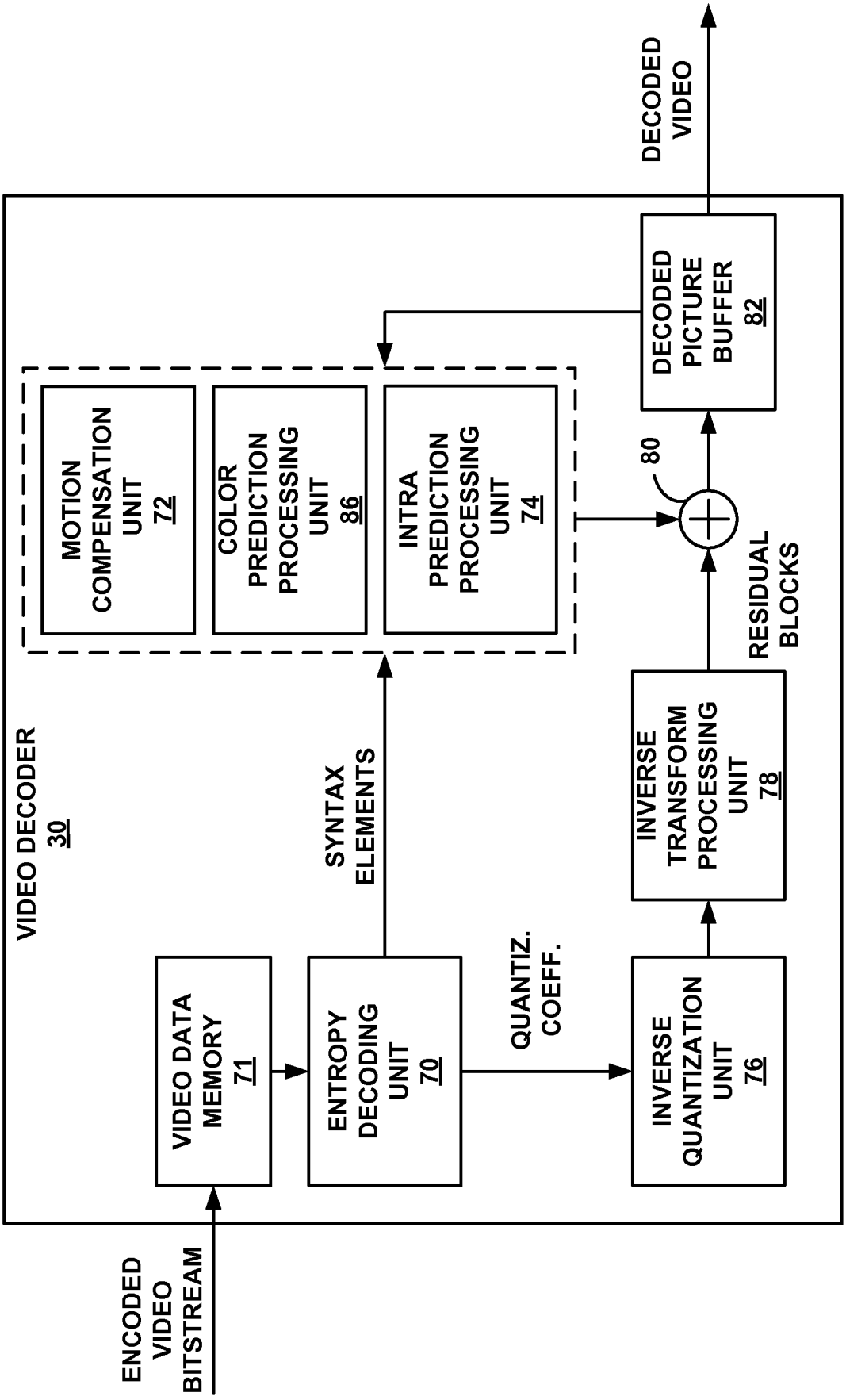

ENCODED VIDEO BITSTREAM
VIDEO DECODER 30
VIDEO DATA MEMORY 71
ENTROPY DECODING UNIT 70
SYNTAX ELEMENTS
QUANTIZ. COEFF.
INVERSE QUANTIZATION UNIT 76
INVERSE TRANSFORM PROCESSING UNIT 78
RESIDUAL BLOCKS
80
MOTION COMPENSATION UNIT 72
COLOR PREDICTION PROCESSING UNIT 86
INTRA PREDICTION PROCESSING UNIT 74
DECODED PICTURE BUFFER 82
DECODED VIDEO

FIG. 12

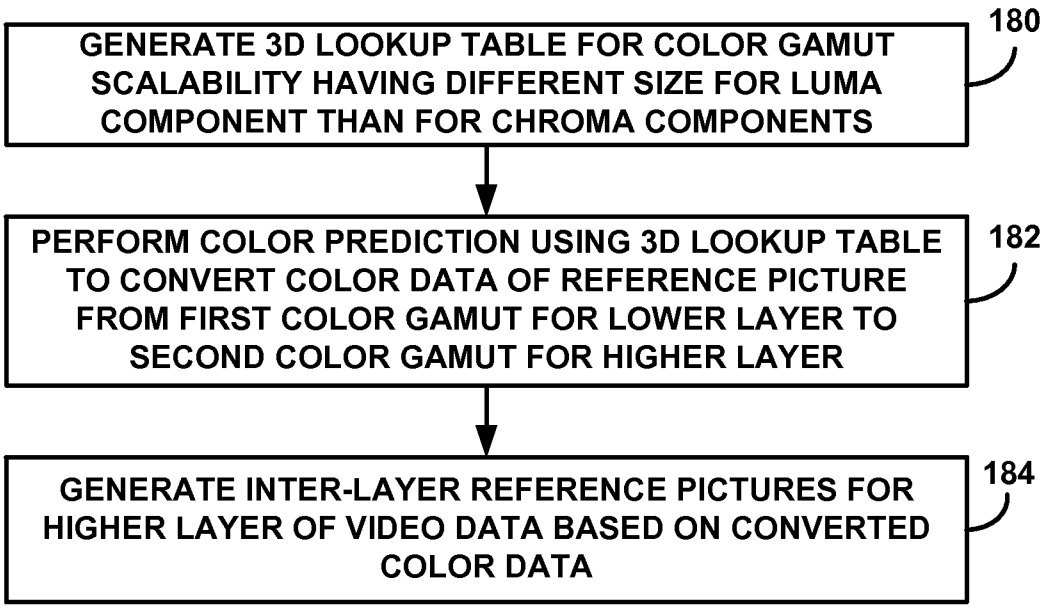
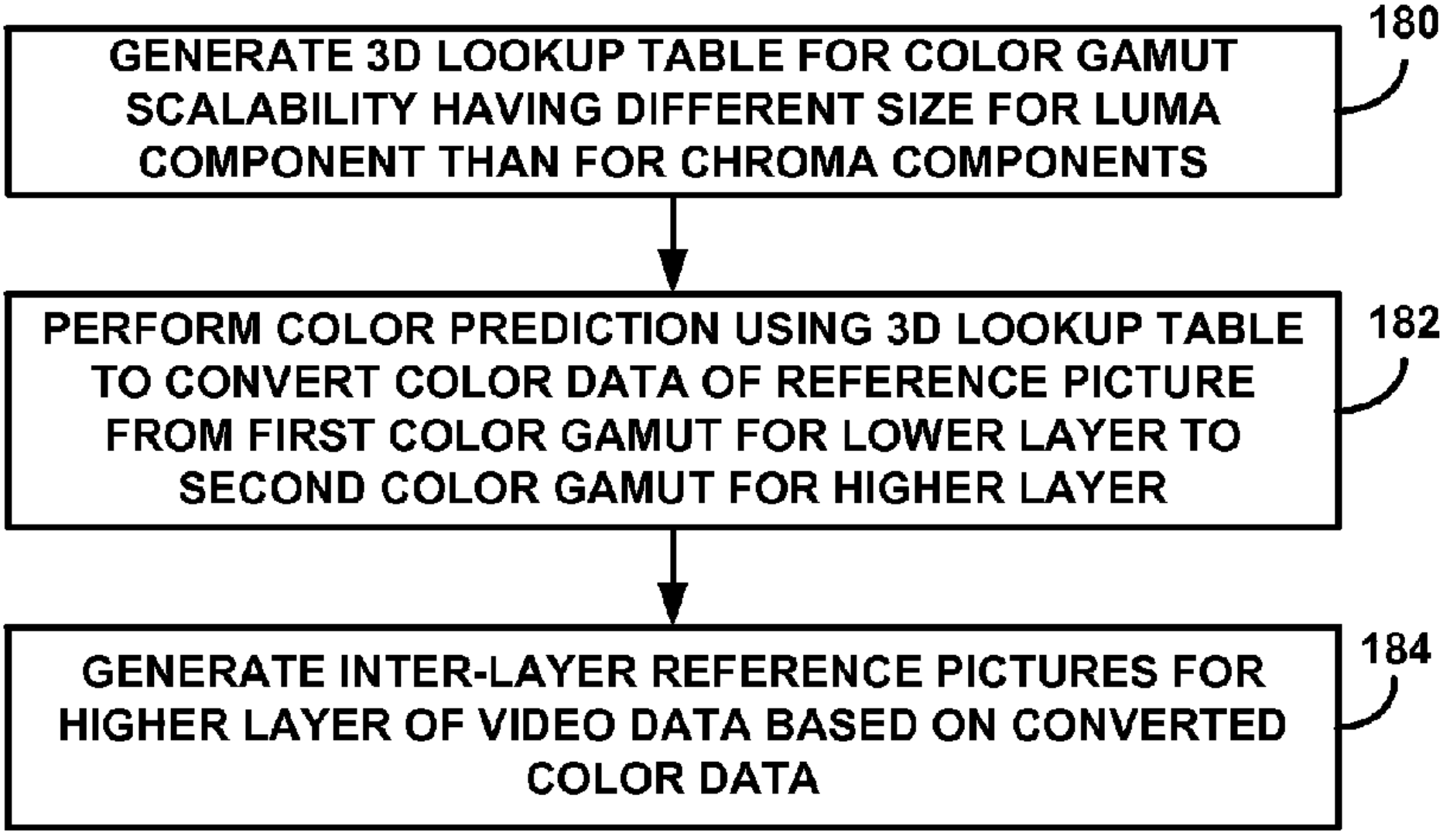

GENERATE 3D LOOKUP TABLE FOR COLOR GAMUT SCALABILITY HAVING DIFFERENT SIZE FOR LUMA COMPONENT THAN FOR CHROMA COMPONENTS
180
PERFORM COLOR PREDICTION USING 3D LOOKUP TABLE TO CONVERT COLOR DATA OF REFERENCE PICTURE FROM FIRST COLOR GAMUT FOR LOWER LAYER TO SECOND COLOR GAMUT FOR HIGHER LAYER
182
GENERATE INTER-LAYER REFERENCE PICTURES FOR HIGHER LAYER OF VIDEO DATA BASED ON CONVERTED COLOR DATA
184

FIG. 13


GENERATE 3D LOOKUP TABLE FOR COLOR GAMUT SCALABILITY HAVING DIFFERENT SIZE FOR LUMA COMPONENT THAN FOR CHROMA COMPONENTS
180
PERFORM COLOR PREDICTION USING 3D LOOKUP TABLE TO CONVERT COLOR DATA OF REFERENCE PICTURE FROM FIRST COLOR GAMUT FOR LOWER LAYER TO SECOND COLOR GAMUT FOR HIGHER LAYER
182
GENERATE INTER-LAYER REFERENCE PICTURES FOR HIGHER LAYER OF VIDEO DATA BASED ON CONVERTED COLOR DATA
184