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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
18 July 2024 (18.07.2024)

WIPO | PCT

(10) International Publication Number
WO 2024/149348 A1

(51) International Patent Classification:
H04N 19/136 (2014.01) *G06T 9/00* (2006.01)
G06T 17/20 (2006.01)

(21) International Application Number:
PCT/CN2024/071911

(22) International Filing Date:
12 January 2024 (12.01.2024)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
PCT/CN2023/071918
12 January 2023 (12.01.2023) CN
PCT/CN2023/089469
20 April 2023 (20.04.2023) CN
PCT/CN2023/106331
07 July 2023 (07.07.2023) CN
PCT/CN2023/123207
07 October 2023 (07.10.2023) CN

(71) Applicants: **DOUYIN VISION CO., LTD.** [CN/CN]; Room B-0035, 2/F, No. 3 Building, No. 30, Shixing Road, Shijingshan District, Beijing 100041 (CN). **BYTEDANCE INC.** [US/US]; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US).

(72) Inventors: **XU, Jizheng**; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US). **WANG, Danying**; Jinritoutiao Post Office, China Satellite Communications Tower No.63, Zhichun Road, Haidian District, Beijing 100080 (CN).

(74) Agent: **LIU , SHEN & ASSOCIATES**; 10th Floor, Building 1, 10 Caihefang Road, Haidian District, Beijing 100080 (CN).

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

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

Declarations under Rule 4.17:
— *of inventorship (Rule 4.17(iv))*

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

(54) Title: JOINTLY CODING OF TEXTURE AND DISPLACEMENT DATA IN DYNAMIC MESH CODING

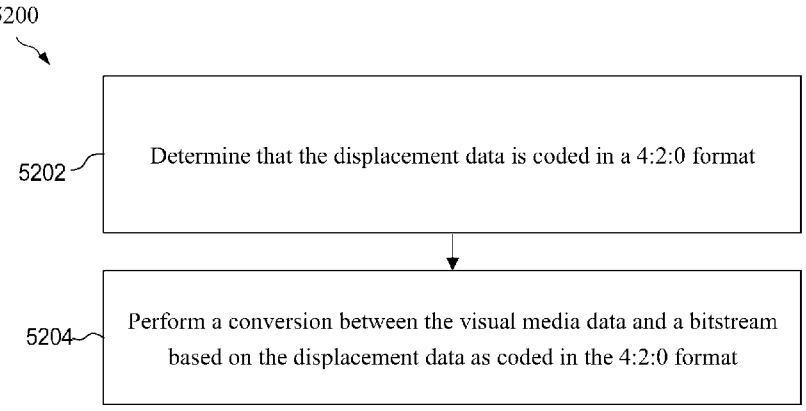


FIG. 12

(57) Abstract: A mechanism for processing video data is disclosed. The mechanism includes determining that the displacement data is coded in a 4:2:0 format. A conversion is performed between the visual media data and a bitstream based on the displacement data as coded in the 4:2:0 format.

WO 2024/149348 A1

Jointly Coding of Texture and Displacement Data in Dynamic Mesh Coding

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This patent application claims the benefit of International Patent Application No. PCT/CN2023/089469 filed on April 20, 2023, International Patent Application No. PCT/CN2023/071918 filed on January 12, 2023, International Patent Application No. PCT/CN2023/106331 filed on July 7, 2023, and International Patent Application No. PCT/CN2023/123207 filed on October 7, 2023, each of which is hereby incorporated by reference.

TECHNICAL FIELD

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

BACKGROUND

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

SUMMARY

[0004] A first aspect relates to a method for processing visual media data including displacement data, comprising: determining that the displacement data is coded in a 4:2:0 format; and performing a conversion between the visual media data and a bitstream based on the displacement data as coded in the 4:2:0 format.

[0005] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the displacement data is converted to the 4:2:0 format prior to encoding, and converted to a 4:4:4 format after decoding.

[0006] Optionally, in any of the preceding aspects, another implementation of the aspect provides that one or more of an sps_chroma_format_idc syntax element and a ChromaFormatIdc variable is set to 1 for coding the displacement data.

[0007] Optionally, in any of the preceding aspects, another implementation of the aspect provides coding of the displacement data uses a main profile or a main10 profile.

[0008] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the displacement data is packed into luma components and chroma components in the 4:2:0 format when the displacement data has only one non-zero component.

[0009] Optionally, in any of the preceding aspects, another implementation of the aspect provides that a subblock size of a subblock of the visual media data is represented by 2S, where S is a non-negative integer.

[0010] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the bitstream includes a value equal to S-K, where K is a non-negative integer.

[0011] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the value in the bitstream is represented as an unsigned integer Exp-Golomb-coded syntax element (ue(v)) or as an unsigned integer using n bits (u(n)).

[0012] Optionally, in any of the preceding aspects, another implementation of the aspect provides that K is equal to 0.

[0013] Optionally, in any of the preceding aspects, another implementation of the aspect provides that K is equal to 6.

[0014] Optionally, in any of the preceding aspects, another implementation of the aspect provides that K is equal to 7.

[0015] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the value of S-K is in a range of 0 to 3, inclusive.

[0016] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the video media data includes the displacement data and texture data, and wherein the texture data and the displacement data are included in a single bitstream.

[0017] Optionally, in any of the preceding aspects, another implementation of the aspect provides converting the displacement data to a 4:2:0 format, and concatenating the displacement data as converted with the texture data in the 4:2:0 format.

[0018] Optionally, in any of the preceding aspects, another implementation of the aspect provides converting the displacement data to N-bit, wherein the N-bit is a bitdepth of the texture data, and where N is an integer.

[0019] Optionally, in any of the preceding aspects, another implementation of the aspect provides that N is 10.

[0020] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the texture data and the displacement data are coded in different slices.

[0021] Optionally, in any of the preceding aspects, another implementation of the aspect provides that one or more of a position and a size of the texture data are included in the single bitstream.

[0022] Optionally, in any of the preceding aspects, another implementation of the aspect provides that one or more of a position and a size of the displacement data are included in the single bitstream.

[0023] Optionally, in any of the preceding aspects, another implementation of the aspect provides that one or more of a position and a size of the texture data are inferred based on information in the single bitstream.

[0024] Optionally, in any of the preceding aspects, another implementation of the aspect provides that one or more of a position and a size of the displacement data are inferred based on information in the single bitstream.

[0025] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the texture data and the displacement data are included in a single bitstream and use different coding methods.

[0026] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the different coding methods comprise a first quantization parameter and a second quantization parameter different from the first quantization parameter, and wherein the texture data uses the first quantization parameter and the displacement data uses the second quantization parameter.

[0027] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the different coding methods comprise lossless coding, and wherein all video units of the displacement data use the lossless coding.

[0028] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the different coding methods comprise a transquant bypass mode from the high efficiency video coding (HEVC) standard, and wherein all video units of the displacement data use the transquant bypass mode.

[0029] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the different coding methods comprise a transform skip mode and a quantization

parameter, and wherein all video units of the displacement data use the transform skip mode and the quantization parameter.

[0030] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the quantization parameter is equal to 4+6*K, where K is an integer.

[0031] Optionally, in any of the preceding aspects, another implementation of the aspect provides that K has a value of zero.

[0032] Optionally, in any of the preceding aspects, another implementation of the aspect provides padding a picture in the single bitstream using data other than the texture data and the displacement data.

[0033] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the picture is padded with a fixed value.

[0034] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the picture is padded with a middle pixel value, and wherein the middle pixel value is 128 for an 8-bit video or 512 for a 10-bit video.

[0035] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the picture is padded with a value of a nearest pixel in the texture data or with a value of a nearest pixel in the displacement data.

[0036] Optionally, in any of the preceding aspects, another implementation of the aspect provides inserting N rows of luma samples and N/2 rows of chroma samples between the texture data and the displacement data when the picture is padded, where N is an integer.

[0037] Optionally, in any of the preceding aspects, another implementation of the aspect provides that a value of N is 16.

[0038] Optionally, in any of the preceding aspects, another implementation of the aspect provides that a value of N is 0.

[0039] Optionally, in any of the preceding aspects, another implementation of the aspect provides that all samples in the N rows of luma samples and the N/2 rows of chroma samples have a same value.

[0040] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the same value comprises a middle pixel value.

[0041] Optionally, in any of the preceding aspects, another implementation of the aspect provides applying a smoothing process to a picture in the single bitstream.

[0042] Optionally, in any of the preceding aspects, another implementation of the aspect provides deriving, at a decoder, whether the displacement data has only one non-zero component or has three non-zero components.

[0043] Optionally, in any of the preceding aspects, another implementation of the aspect provides determining, at a decoder, whether the displacement data has only one non-zero component or has three non-zero components based on a video resolution of the displacement data and a number of base mesh points.

[0044] Optionally, in any of the preceding aspects, another implementation of the aspect provides determining, at a decoder, whether the displacement data has only one non-zero component or has three non-zero components based on a video resolution of the displacement data and a number of vertexes.

[0045] Optionally, in any of the preceding aspects, another implementation of the aspect provides converting the texture data to a color space prior to encoding and converting the texture data back to the color space after decoding, wherein the color space is not a blue green red (BGR) color space or a YUV color space, where Y represents a luma component and U and V represent blue and red chroma components.

[0046] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the texture data in the bitstream is coded in a green blue red (GBR) color space.

[0047] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the texture data in the bitstream is coded in a green red blue (GRB) color space.

[0048] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the texture data in the bitstream is coded in a YCgCo color space, where Y represents a luma component, Cg represents a green chroma component, and Co represents an orange chroma component.

[0049] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the texture data in the bitstream is coded in a YCoCg color space, where Y represents a luma component, Co represents an orange chroma component, and Cg represents a green chroma component.

[0050] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the color space is used for losslessly coding the texture data.

[0051] Optionally, in any of the preceding aspects, another implementation of the aspect provides an index of color primaries representing a green red blue (GRB) color space is used for coding according to the International Telecommunication Union - Telecommunication Standardization Sector (ITU-T) standard.

[0052] Optionally, in any of the preceding aspects, another implementation of the aspect provides that a first subblock size (SB1) used to code a first color component of the visual media data is different from a second subblock size (SB2) used to code a second color component of the visual media data.

[0053] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the second subblock size is greater than the first subblock size.

[0054] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the first subblock size is 100 and the second subblock size is 200.

[0055] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the first subblock size is 128 and the second subblock size is 256.

[0056] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the displacement data comprises inter displacement data and intra displacement data, and wherein the inter displacement data uses a different block size than the intra displacement data.

[0057] Optionally, in any of the preceding aspects, another implementation of the aspect provides that a pin occupancy present flag is equal to 0 when a packing information scheme is used to support jointly coding of the displacement data and texture data in dynamic mesh coding.

[0058] Optionally, in any of the preceding aspects, another implementation of the aspect provides that a region of the visual media data corresponding to the pin occupancy present flag is ignored by a dynamic mesh decoder when the pin occupancy present flag is not equal to 0.

[0059] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the pin occupancy present flag is designated pin_occupancy_present_flag[j].

[0060] Optionally, in any of the preceding aspects, another implementation of the aspect provides that a pin region type syntax element corresponding to the visual media data is not 0.

[0061] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the pin region type syntax element is designated pin_region_type_id_minus2[j][i].

[0062] Optionally, in any of the preceding aspects, another implementation of the aspect provides that no region in the packing information scheme overlaps another region in the packing information scheme.

[0063] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the conversion includes encoding the media data into a bitstream.

[0064] Optionally, in any of the preceding aspects, another implementation of the aspect provides that the conversion includes decoding the media data from a bitstream.

[0065] A second aspect relates to an apparatus for processing media data comprising: a processor; and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to perform the method of any of the disclosed embodiments.

[0066] A third aspect relates to a non-transitory computer readable medium, comprising a computer program product for use by a video coding device, the computer program product comprising computer executable instructions stored on the non-transitory computer readable medium such that when executed by a processor cause the video coding device to perform the method of any of the disclosed embodiments.

[0067] A fourth aspect relates to a non-transitory computer-readable recording medium storing a bitstream of a video which is generated by a method performed by a video processing apparatus, wherein the method comprises the method of any of the disclosed embodiments.

[0068] A fifth aspect relates to a method for storing a bitstream of a video comprising the method of any of the disclosed embodiments.

[0069] A sixth aspect relates to a method, apparatus, or system described in the present disclosure.

[0070] For the purpose of clarity, any one of the foregoing embodiments may be combined with any one or more of the other foregoing embodiments to create a new embodiment within the scope of the present disclosure.

[0071] These and other features will be more clearly understood from the following detailed description taken in conjunction with the accompanying drawings and claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0072] For a more complete understanding of this disclosure, reference is now made to the following brief description, taken in connection with the accompanying drawings and detailed description, wherein like reference numerals represent like parts.

[0073] FIG. 1 illustrates an example decoder design for dynamic mesh coding.

[0074] FIG. 2 illustrates an example structure of a dynamic mesh coding test model.

[0075] FIG. 3 illustrates an example combination of texture and displacement data into one picture.

[0076] FIG. 4 illustrates another example combination of texture and displacement data into one picture.

[0077] FIG. 5 is a block diagram showing an example video processing system.

[0078] FIG. 6 is a block diagram of an example video processing apparatus.

[0079] FIG. 7 is a flowchart for an example method of video processing.

[0080] FIG. 8 is a block diagram that illustrates an example video coding system.

[0081] FIG. 9 is a block diagram that illustrates an example encoder.

[0082] FIG. 10 is a block diagram that illustrates an example decoder.

[0083] FIG. 11 is a schematic diagram of an example encoder.

[0084] FIG. 12 is a flowchart for an example method of video processing.

DETAILED DESCRIPTION

[0085] It should be understood at the outset that although an illustrative implementation of one or more embodiments are provided below, the disclosed systems and/or methods may be implemented using any number of techniques, whether currently known or yet to be developed. The disclosure should in no way be limited to the illustrative implementations, drawings, and techniques illustrated below, including the exemplary designs and implementations illustrated and described herein, but may be modified within the scope of the appended claims along with their full scope of equivalents.

[0086] Section headings are used in the present disclosure for ease of understanding and do not limit the applicability of techniques and embodiments disclosed in each section only to that section. Furthermore, the techniques described herein are applicable to other video codec protocols and designs.

1. Initial discussion

[0087] This disclosure is related to Moving Picture Experts Group (MPEG)-I video-based dynamic mesh coding. Specifically, it is related to how to jointly code the displacement and texture data. It may be also applicable to other immersive video coding standards or codecs.

2. Further discussion

[0088] In computer graphics, a three dimensional (3D)/immersive content can usually be represented by a 3D mesh and a texture map. Those mesh and texture data can be generated by a machine or can be converted from images captured by multiple cameras from different angles. Similar to two dimensional (2D) video, when those 3D contents change with time, the mesh and texture data also change and consist a sequence of dynamic mesh. The data volume of dynamic mesh are usually huge and make it difficult to store and transmit. To meet the requirement of applications that use dynamic mesh, MPEG, issued a call for proposal. To efficiently use the 2D codecs, one of the requirements is to use a 2D video coding standard to compress most data and keep other parts simple and of low complexity. Such a requirement can guarantee that the representation can take advantages of the 2D video hardware/software systems, without much efforts to redesign a specific system just for dynamic mesh.

[0089] MPEG received several responses to the call for proposal. Among them, one scheme showed better performance compared with others. A test model was built for the development of the planned dynamic mesh coding standard.

[0090] A test model of dynamic mesh coding can be found via this link http://mpegx.int-evry.fr/software/MPEG/dmc/mpeg-vmesh-tm/-/tags/v2.0; and the latest working draft document is WD 1.0.

2.1 Data representation in dynamic mesh coding

[0091] FIG. 1 illustrates an example decoder design for dynamic mesh coding. FIG. 1 shows am example decoder design. It can be seen that a dynamic mesh decoder receives 3 bitstreams and performs decoding to reconstruct the dynamic mesh plus texture signals. The first bitstream is to represent the base mesh, a decimated version of the original mesh. The second bitstream is to represent displacement vectors between the reconstructed base mesh and the original mesh. The displacement vectors are arranged as a 2D video and compressed with an 2D video coding standard compliant codec. The third bitstream is to represent the texture (or attribute map). The attribute map is also arranged as a 2D video and compressed with an 2D video coding standard compliant codec.

The design philosophy is to make the base mesh part small enough so that the module to process base mesh can be implemented simply. On the other hand, the displacement vectors and the attribute map accounts for most of the volume of the whole dynamic mesh data, which can be processed with dedicated high efficient 2D video coding systems. Such a design can reduce the extra efforts to implement the dynamic mesh coding system and guarantee the high throughout and coding efficiency for the dynamic mesh data.

2.2 Test model of dynamic mesh coding

[0092] FIG. 2 illustrates an example structure of a dynamic mesh coding test model. FIG. 2 shows the structure of an example dynamic mesh coding model. In this model, Draco is used to compress base mesh and the High efficiency video coding (HEVC) test model (HM) is used to compress displacement vectors and attribute map. However, it should be noted that other mesh or video coding systems can also be used in dynamic mesh coding.

[0093] The base mesh m is generated from the original mesh with a down-sampling scheme. Its quantized version m' is then coded using Draco. The reconstrused base mesh m'' can be obtained by inverse quantization of m'. Displacement vectors d' are generated by making the difference between the original mesh and the subdivided version of m'' using a subdivision scheme.

2.3 Coding of displacement vectors

[0094] After obtaining displacement vectors d', the difference between the original mesh and the subdivied base mesh, a lifting-based wavelet transform is applied to further make the energy compact. Then the wavelet transform coefficients are traversed from low to high frequency using a Morton order to form 2D coefficient blocks. Various 2D coefficient blocks comprise a picture to be processed by a 2D codec.

2.4 Motion field coding

[0095] In an example test model, motion fields between base meshes are directly coded using arithmetic coding. An example design investigates coding of motion fields also with a standard compliant 2D coding system and showed that the coding efficiency loss is marginal. Thus, it may make sense to further shift the coding process of motion field to a 2D video codec.

2.5 Chroma formats

[0096] In H.264/advanced video coding (AVC), H.265/HEVC and H.266/versatile video coding (VVC), different chroma formats are supported. The format may be signalled by the syntax element

sps_chroma_format_idc and represented by the variable ChromaFormatIdc. The following table illustrates the chroma formats corresponding to different sps_chroma_format_idc:

sps_chroma_format_idc	Chroma format
0	Monochrome
1	4:2:0
2	4:2:2
3	4:4:4

2.6 Lossless coding

[0097] In H.264/AVC and H.266/VVC, lossless coding can be achieved by setting QP to be 4 and apply a invertible spatial transform or transform skipping to the block. In H.265/HEVC, in addition to the above method, lossless coding can also be achieve by setting the cu_transquant_bypass_flag of a coding unit to be 1.

2.7 Example designs

[0098] In an example design, ideas are presented to combine multiple attributes, including texture, displacement data, occupancy data into one video for encoding/decoding without requiring multiple encoding/decoding capabilities on a device. This disclosure describes details on how texture and displacement data are jointly coded for dynamic mesh coding.

2.8 Arithmetic coded of displacement data

[0099] In an example design of dynamic mesh coding [5], displacement data can be coded using arithmetic coding. The following syntax table and semantics.

J.7.1.3.7 Displacement intra data unit syntax

displ_intra_unit(unitSize, lodCount, subblock_size, vertexCount) {	**Descriptor**
for(k = 0; k < 3; k++) {	
diu_last_sig_coeff[k]	ae(v)
for(b = 0; b < lodCount; b++) {	
diu_coded_block_flag[k][b]	u(v)
if(diu_coded_block_flag[i][j]) {	
for(s = 0; s < vertexCount[b] % subblock_size; s++) {	
diu_coded_subblock_flag[k][b][s]	u(v)
if(diu_coded_subblock_flag[k][b][s]) {	

for(v = vStart; v < subblock_size; v++) {	
diu_coeff_abs_level_gt0[k][b][s][v]	u(v)
if(diu_coeff_abs_level_gt0[k][b][s][v]) {	
diu_coeff_abs_level_gt1[k][b][s][v]	u(v)
diu_coeff_sign[k][b][s][v]	u(1)
if(diu_coeff_abs_level_gt1[k][b][s][v]) {	
diu_coeff_abs_level_rem[k][b][s][v]	ue(v)
}	
}	
}	
}	
}	
}	
if (dsps_single_dimension_flag) {	
break;	
}	
}	
}	

[0100] In the above table, subblock_size is signalled in the bitstream as u(16).

J.7.3.5 Displacement intra data unit semantics

[0101] The arithmetic decoding engine is a context-separated, binary arithmetic decoder, performing binary renormalization and producing binary outputs.

[0102] The displacement values are derived from the arithmetic decoding.

[0103] diu_last_sig_coeff[k] indicates the index of the last position of the nonzero displacement coefficient level in the k-th components.

[0104] diu_coded_block_flag[k][b] indicates whether the block with index b has any nonzero displacement coefficient levels in the k-th components (when 1), or not (when 0).

[0105] diu_coded_subblock_flag[k][b][s] indicates whether the subblock with index s of the block with index b has any nonzero displacement coefficient levels in the k-th components (when 1), or not (when 0).

[0106] diu_coeff_abs_level_gt0[k][b][s][v] indicates whether the k-th component of the displacement coefficient level associated with the vertex with index v of the subblock with index s of the block with index b has an absolute value higher than zero (when 1), or not (when 0).

[0107] diu_coeff_abs_level_gt1[k][b][s][v] indicates whether the k-th component of the displacement coefficient level associated with the vertex with index v of the subblock with index s of the block with index b has an absolute value higher than one (when 1), or not (when 0). If diu_coeff_abs_level_gt1[k][b][s][v] is not present it shall be inferred to be equal to 0.

[0108] diu_coeff_sign[k][b][s][v] indicates whether the k-th component of the displacement coefficient level associated with the vertex with index v of the subblock with index s of the block with index b has a positive sign (when 1), or not (when 0). If diu_coeff_sign[k][b][s][v] is not present it shall be inferred to be equal to 1.

[0109] diu_coeff_abs_level_rem[k][b][s][v] indicates the absolute value of the k-th component of the displacement coefficient level associated with the vertex with index v of the block with index b minus 2. If diu_coeff_abs_level_rem[k][b][s][v] is not present it shall be inferred to be equal to 0.

2.9 Packing information in the V3C design

[0110] In the latest V3C design [7], a packing information scheme is supported to have multiple information coded in one frame. The V3C unit type is V3C_PVD to use such a scheme, The corresponding syntax and semantics are showed below:

8.3.4.7 Packing information syntax

packing_information(j) {	**Descriptor**
pin_codec_id[j]	u(8)
pin_occupancy_present_flag[j]	u(1)
pin_geometry_present_flag[j]	u(1)
pin_attribute_present_flag[j]	u(1)
if(pin_occupancy_present_flag[j]) {	
pin_occupancy_2d_bit_depth_minus1[j]	u(5)

pin_occupancy_msb_align_flag[j]	u(1)
pin_lossy_occupancy_compression_threshold[j]	u(8)
}	
if(pin_geometry_present_flag[j]) {	
pin_geometry_2d_bit_depth_minus1[j]	u(5)
pin_geometry_msb_align_flag[j]	u(1)
pin_geometry_3d_coordinates_bit_depth_minus1[j]	u(5)
}	
if(pin_attribute_present_flag[j]) {	
pin_attribute_count[j]	u(7)
for(i = 0; i < **pin_attribute_count**[j]; i++) {	
pin_attribute_type_id[j][i]	u(4)
pin_attribute_2d_bit_depth_minus1[j][i]	u(5)
pin_attribute_msb_align_flag[j][i]	u(1)
pin_attribute_map_absolute_coding_persistence_flag[j][i]	u(1)
d = **pin_attribute_dimension_minus1**[j][i]	u(6)
if(d == 0) {	
pin_attribute_dimension_partitions_minus1[j][i] = 0	
m = 0	
} else	
m = **pin_attribute_dimension_partitions_minus1**[j][i]	u(6)
for(k = 0; k < m; k++) {	
if(k + d == m) {	
pin_attribute_partition_channels_minus1[j][i][k] = 0	
n = 0	
} else	

n = **pin_attribute_partition_channels_minus1**[j][i][k]	ue(v)
d –= n + 1	
}	
pin_attribute_partition_channels_minus1[j][i][m] = d	
}	
}	
pin_regions_count_minus1[j]	ue(v)
for(i = 0; i <= pin_regions_count_minus1[j]; i++) {	
pin_region_tile_id[j][i]	u(8)
pin_region_type_id_minus2[j][i]	u(2)
pin_region_top_left_x[j][i]	u(16)
pin_region_top_left_y[j][i]	u(16)
pin_region_width_minus1[j][i]	u(16)
pin_region_height_minus1[j][i]	u(16)
pin_region_unpack_top_left_x[j][i]	u(16)
pin_region_unpack_top_left_y[j][i]	u(16)
pin_region_rotation_flag[j][i]	u(1)
if(pin_region_type_id_minus2[j][i] + 2 == V3C_AVD || pin_region_type_id_minus2[j][i] + 2 == V3C_GVD) {	
pin_region_map_index[j][i]	u(4)
pin_region_auxiliary_data_flag[j][i]	u(1)
}	
if(pin_region_type_id_minus2[j][i] + 2 == V3C_AVD) {	
pin_region_attr_index[j][i]	u(7)
k = pin_region_attr_index[j][i]	
if(pin_attribute_dimension_minus1[j][k] > 0)	
pin_region_attr_partition_index[j][i]	u(5)
}	
}	

}	

8.4.4.7 Packing information semantics

[0111] Packed video frames can be divided into one or more rectangular regions. One region shall be mapped to exactly one atlas tile. The rectangular regions of a packed video frame are not allowed to overlap.

[0112] pin_codec_id[j] indicates a mapping index of a codec identifier of the video decoder used to decode the packed video sub-bitstream, if present, for the atlas with ID j, as specified in subclause 9.6. pin_codec_id[j] shall be in the range of 0 to 255, inclusive.

[0113] pin_occupancy_present_flag[j] equal to 0 indicates that packed video frames of the atlas with atlas ID j do not contain regions with occupancy data. pin_occupancy_present_flag[j] equal to 1 indicates that packed video frames of the atlas with atlas ID j do contain regions with occupancy data. When pin_occupancy_present_flag[j] is not present, its value is inferred to be equal to 0.

[0114] It is a requirement of bitstream conformance that if pin_occupancy_present_flag[j] is equal to 1 for an atlas with atlas ID j, vps_occupancy_video_present_flag[j] shall be equal to 0 for an atlas with the same atlas ID j.

[0115] pin_geometry_present_flag[j] equal to 0 indicates that packed video frames of the atlas with atlas ID j do not contain regions with geometry data. pin_geometry_present_flag[j] equal to 1 indicates that packed video frames of the atlas with atlas ID j do contain regions with geometry data. When pin_geometry_present_flag[j] is not present, its value is inferred to be equal to 0.

[0116] It is a requirement of bitstream conformance that if pin_geometry_present_flag[j] is equal to 1 for an atlas with atlas ID j, vps_geometry_video_present_flag[j] shall be equal to 0 for an atlas with the same atlas ID j.

[0117] pin_attribute_present_flag[j] equal to 0 indicates that packed video frames of the atlas with atlas ID j do not contain regions with attribute data. pin_attribute_present_flag[j] equal to 1 indicates that packed video frames of the atlas with atlas ID j do contain regions with attribute data. When pin_attribute_present_flag[j] is not present, its value is inferred to be equal to 0.

[0118] It is a requirement of bitstream conformance that if pin_attribute_present_flag[j] is equal to 1 for an atlas with atlas ID j, vps_attribute_video_present_flag[j] shall be equal to 0 for an atlas with the same atlas ID j.

[0119] pin_occupancy_2d_bit_depth_minus1[j] plus 1 indicates the nominal 2D bit depth to which the decoded regions containing occupancy data for the atlas with atlas ID j shall be converted. pin_occupancy_msb_align_flag[j] shall be in the range of 0 to 31, inclusive.

[0120] pin_occupancy_msb_align_flag[j] indicates how the decoded regions containing occupancy samples associated with an atlas with atlas ID j are converted to samples at the nominal occupancy bit depth, as specified in Annex B.

[0121] pin_lossy_occupancy_compression_threshold[j] indicates the threshold to be used to derive the binary occupancy from the decoded regions containing occupancy data for the atlas with atlas ID j, as specified in Annex B. pin_lossy_occupancy_compression_threshold[j] shall be in the range of 0 to 255, inclusive.

[0122] pin_geometry_2d_bit_depth_minus1[j] plus 1 indicates the nominal 2D bit depth to which the decoded regions containing geometry data for the atlas with atlas ID j shall be converted. pin_geometry_2d_bit_depth_minus1[j] shall be in the range of 0 to 31, inclusive.

[0123] pin_geometry_msb_align_flag[j] indicates how the decoded regions containing geometry samples associated with an atlas with atlas ID j are converted to samples at the nominal occupancy bit depth, as specified in Annex B.

[0124] pin_geometry_3d_coordinates_bit_depth_minus1[j] plus 1 indicates the bit depth of the geometry coordinates of the reconstructed volumetric content for the atlas with atlas ID j. pin_geometry_3d_coordinates_bit_depth_minus1[j] shall be in the range of 0 to 31, inclusive.

[0125] pin_attribute_count[j] indicates the number of attributes with unique attribute types present in packed video frames for the atlas with atlas ID j. pin_attribute_count[j] shall be in the range of 1 to 127, inclusive.

[0126] pin_attribute_type_id[j][i] indicates the attribute type of the attribute with attribute with index i, for the atlas with atlas ID j. Table 4 describes the list of supported attributes types.

[0127] pin_attribute_2d_bit_depth_minus1[j][i] plus 1 indicates the nominal 2D bit depth to which the regions containing an attribute with attribute index i, for the atlas with atlas ID j, shall be converted. pin_attribute_2d_bit_depth_minus1[j][i] shall be in the range of 0 to 31, inclusive.

[0128] pin_attribute_msb_align_flag[j][i] indicates how the decoded regions containing attribute with attribute index i, for the atlas with atlas ID j, are converted to samples at the nominal attribute bit depth, as specified in Annex B.

[0129] pin_attribute_map_absolute_coding_persistence_flag[j][i] equal to 1 indicates that the decoded regions containing attribute maps, for the attribute with index i, that correspond to the atlas with atlas ID j, are coded without any form of map prediction. pin_attribute_map_absolute_coding_persistence_flag[j][i] equal to 0 indicates that the decoded regions containing attribute maps of the attribute with index i, that correspond to the atlas with atlas ID j, shall use the same map prediction method as used for the geometry component of the atlas with atlas ID j. If pin_attribute_map_absolute_coding_persistence_flag[j][i] is not present, its value shall be inferred to be equal to 1.

[0130] The 3D array AttributeMapAbsoluteCodingEnabledFlag, which indicates if a particular map of an attribute is to be coded with or without prediction, is obtained as follows:

```
if( pin_attribute_map_absolute_coding_persistance_flag[ j ][ i ] == 1) {
    for( k = 0; k < vps_map_count_minus1[ j ]; k++ )
        PackingAttributeMapAbsoluteCodingEnabledFlag[ j ][ i ][ k ] = 1
}
else{
    for( k = 0; k < vps_map_count_minus1[ j ]; k++ )
        PackingAttributeMapAbsoluteCodingEnabledFlag[ j ][ i ][ k ] =
            vps_map_absolute_coding_enabled_flag[ j ][ i ]
}
```

[0131] pin_attribute_dimension_minus1[j][i] plus 1 indicates the total number of dimensions (i.e., number of channels) of the regions containing an attribute with index i, for the atlas with atlas ID j. pin_attribute_dimension_minus1[j][i] shall be in the range of 0 to 63, inclusive.

[0132] pin_attribute_dimension_partitions_minus1[j][i] plus 1 indicates the number of partition groups in which the attribute channels of the regions containing an attribute with index i, for the atlas with atlas ID j, should be grouped in. pin_attribute_dimension_partitions_minus1[j][i] shall be in the range of 0 to 63, inclusive.

[0133] pin_attribute_partition_channels_minus1[j][i][k] plus 1 indicates the number of channels assigned to the dimension partition group with index k, of the regions containing an attribute with index i, for the atlas with atlas ID j. ai_attribute_partition_channels_minus1[j][i][k] shall be in the range of 0 to pin_attribute_dimension_minus1[j][i], inclusive, for all dimension partition groups.

[0134] pin_regions_count_minus1[j] plus 1 indicates the number of regions packed in one video frame for the atlas with ID j. pin_regions_count_minus1 shall be in the range of 0 to 255, inclusive.

[0135] pin_region_tile_id[j][i] indicates the atlas tile ID, of the atlas with ID j, corresponding to the region with index i.

[0136] pin_region_type_id_minus2[j][i] plus 2 indicates the ID of the region with index i for the atlas with ID j. The value of pin_region_type_id_minus2[j][i] shall be in the range of 0 to 2, inclusive.

[0137] pin_region_top_left_x[j][i] specifies the horizontal position of the top left sample of the region with index i for the atlas with ID j in unit of luma samples in the packed video component frame.

[0138] pin_region_top_left_y[j][i] specifies the vertical position of top left sample of the region with index i for the atlas with ID j in unit of luma samples in in the packed video component frame.

[0139] pin_region_width_minus1[j][i] plus 1 specifies the width of the region with index i for the atlas with ID j in units of luma samples.

[0140] pin_region_height_minus1[j][i] plus 1 specifies the height of the region with index i for the atlas with ID j in units of luma samples.

[0141] pin_region_unpack_top_left_x[j][i] specifies the horizontal position of the top left sample of the region with index i for the atlas with ID j in unit of luma samples in the unpacked video component frame.

[0142] pin_region_unpack_top_left_y[j][i] specifies the vertical position of the top left sample of the region with index i for the atlas with ID j in unit of luma samples in the unpacked video component frame.

[0143] pin_region_rotation_flag[j][i] equal to 0 indicates that no rotation on the region with index i for the atlas with ID j is performed. pin_region_rotation_flag[j][i] equal to 1 indicates that the region with index i for the atlas with ID j is rotated 90 degrees.

[0144] pin_region_map_index[j][i] specifies the map index of the region with index i of the atlas with ID j.

[0145] pin_region_auxiliary_data_flag[j][i] equal to 1 indicates that the region with index i of the atlas with ID j contains RAW and/or EOM coded points only. pin_region_auxiliary_data_flag

equal to 0 indicates that the region with index i of the atlas with ID j may contains RAW and/or EOM coded points.

[0146] pin_region_attr_index[j][i] indicates the attribute index of the region with index i of the atlas with ID j. The value of pin_region_attr_index[j][i] shall be in the range of 0 to pin_attribute_count[j] – 1, inclusive.

[0147] pin_region_attr_partition_index[j][i] indicates the attribute partition index of the region with index i of the atlas with ID j. When not present, the value of pin_region_attr_partition_index[j][i] is inferred to be equal to 0.

3. Technical problems solved by disclosed technical solutions

[0148] In an example design of dynamic mesh coding, there are the following problems:

[0149] First, texture and displacement data are coded into two separate bitstreams, which needs two encoders or decoders to support the implementation. A single bitstream thus is more preferred.

[0150] Second, displacement data are coded in 4:4:4 or 4:0:0 format, which needs more encoding/decoding resources in HEVC.

[0151] Third, the coding efficiency for texture data can be improved when the data are in green red blue (GRB)/green blue red (GBR)/YCgCo-R domain, where Y represents a luma component, Cg represents a green chroma component, and Co represents an orange chroma component.

[0152] Fourth, when arithmetic coding is used to code displacement data, subblock size is signalled in u(16), which has a too wide range and cannot be implemented using bit-shifting.

[0153] Fifth, when arithmetic coding is used to code displacement data, using different subblock sizes for different colour components or different type of frame may be better.

[0154] Sixth, when using packing_information() structure to jointly code texture and displacement, there should be some modifications/constraints.

4. A listing of solutions and embodiments

[0155] The detailed list below should be considered as examples to explain general concepts. These examples should not be interpreted in a narrow way. Furthermore, these examples can be combined in any manner.

1. To solve problem 1, texture and displacement data are combined into one video for encoding/decoding.
    a. In one example, displacement data may be converted to 4:2:0 format and concatenated with texture in 4:2:0 format.

b. In one example, displacement data may be converted to N-bit that is the bitdepth of the texture.
    i. In one example, N is 10.
c. In one example, texture and displacement data are coded in different slices.
d. In one example, the positions and/or sizes of the texture area and/or the displacement data area may be signalled in the bitstream.
e. In one example, the positions and/or sizes of the texture area and/or the displacement data area may be inferred from the bitstream.

2. To better solve problem 1, texture part and displacement part coded in one video may use different encoding strategies/methods.
    a. In one example, texture may use one QP and displacement data may use a different QP.
    b. In one example, all video units of displacement data use lossless coding.
        i. In one example, all video units of displacement data apply transquant_bypass mode in HEVC.
        ii. In one example, all video units of displacement data apply transform skip mode and QP equal to 4+6*K.
            1. In one example, K is 0.
3. To better solve problem 1, an encoder may pad data to area in the combined picture but not belongs to either texture or displacement data.
    a. In one example, the area is padded with a fixed value.
        i. In one example, the area is padded with the middle pixel value, i.e. 1<<(bitdepth-1), for example, 128 for 8-bit video and 512 for 10-bit video.
    b. In one example, the area is padded with the value of the nearest pixel either in texture area or displacement data area.
    c. In one example, N rows of luma samples and N/2 rows of chroma samples are inserted between texture and displacement data.
        i. In one example, N is 16.
        ii. In one example, N is 0.
        iii. In one example, all inserted samples have the same value, e.g. middle pixel value.

4. To better solve problem 1, a smoothing process may be applied to the combined picture.
    a. In one example, only the padded area may be smoothed.
5. To solve problem 2, displacement data may be coded in 4:2:0 format.
    a. In one example, displacement data may be converted to 4:2:0 before encoding and converted back to 4:4:4 after decoding.
    b. In one example, sps_chroma_format_idc and/or ChromaFormatIdc may be set to 1 for displacement data coding.
    c. In one example, in HEVC, coding of displacement data may use main or main10 profile.
    d. In one example, when displacement data have only one non-zero component, the displacement data may be packed into luma and chroma components in 4:2:0 format.
6. To better solve problem 2, whether displacement data have only one non-zero component or three non-zero components may be derived at the decoder.
    a. In one example, based on the displacement video resolution and number of base mesh points, whether displacement data have only one non-zero component or three non-zero components may be derived at the decoder.
    b. In one example, based on the displacement video resolution and number of vertexes, whether displacement data have only one non-zero component or three non-zero components may be derived at the decoder.
7. To solve problem 3, it is proposed that texture data may be converted to a colour space other than YUV or BGR before encoding and converted back after decoding.
    a. In one example, texture data may be coded in GBR colour space.
    b. In one example, texture data may be coded in GRB colour space.
    c. In one example, texture data may be coded in YCgCo colour space.
    d. In one example, texture data may be coded in YCoCg colour space.
    e. In the above sub-items, the colour space may be used for losslessly coding of texture data.
8. To better solve problem 3, it is proposed to have a new index of colour primaries to represent GRB colour space in the International Telecommunication Union - Telecommunication Standardization Sector (ITU-T) H.273 recommendation.

9. To solve problem 4, it is proposed to represent subblock size to be 2^S and signalled (S-K) in the bitstream in ue(v) or u(n), where where S is a non-negative integer, where ue(v) is an unsigned integer Exp-Golomb-coded syntax element (ue(v)), and wherein u(n) is an unsigned integer using n bits (u(n)), and where K is a non-negative integer.
    a. In one example, K is equal to 0.
    b. In one example, K is equal to 6.
    c. In one example, K is equal to 7.
    d. In one example, S-K shall be in the range of 0 to 3, inclusive.
10. To solve problem 5, the subblock size (denoted as SB2) used in coding the 2nd and 3rd colour components may be different from the subblock size (denoted as SB1) used in coding the 1st colour component.
    a. In one example, SB2 is greater than SB1.
        i. In one example, SB1 is equal to 100 and SB2 is equal to 200.
        ii. In one example, SB1 is equal to 128 and SB2 is equal to 256.
11. To solve problem 4, alternatively, a different subblock size may be used for inter displacement data than the intra displacement data.
12. To solve problem 6, when the packing information scheme is used to support jointly texture and displacement coding in dynamic mesh coding, one or more following constraints or their combinations are presented:
    a. In one example, pin_occupancy_present_flag[j] shall be equal 0.
        i. Alternatively, when pin_occupancy_present_flag[j] is not equal to 0, a dynamic mesh decoder should ignore the region corresponding to the flag.
    b. In one example, pin_region_type_id_minus2[j][i] shall not be equal to 0.
    c. In one example, each region described in the packing information shall not overlap with each other.

5. Embodiments

5.1. Embodiment 1

[0156] FIG. 3 illustrates an example combination of texture and displacement data into one picture. Assume that the size of texture is texture width x texture height (WTxHT) and the size of displacement data is displacement width x displacement height (WDxHD) in luma sample unit. FIG. 3 shows one exemplary combination of texture and displacement data. The texture is placed

from (0, 0) to (WT-1, HT-1) in the combined picture. The displacement will be placed from (0, floor(HT/N)*N) to (WD, floor(HT/N)*N+HT-1) after being converted to 4:2:0 format, N is the length of the minimum coding unit. The picture has the minimal size to cover both the texture and the displacement data and can be processed directly by a standard compliant coding system. A new slice begins from the upper-left corner of the displacement data region in the combined picture. All the others are of padding area, in which an encoder pad data according some methods.

5.2. Embodiment 2

[0157] FIG. 4 illustrates another example combination of texture and displacement data into one picture. Assume that the size of texture is WTxHT and the size of displacement data is WDxHD in luma sample unit. The following figure shows another exemplary combination of texture and displacement data. The texture is placed from (0, 0) to (WT-1, HT-1) in the combined picture. The displacement data are firstly converted to 4:2:0 format and then stretched into rectangular area with height being H, where H is the length of the basic displacement block (currently H is equal to 16 in luma sample unit). The stretched displacement data will be placed starting from (0, floor(HT/N)*N) coding unit (usually the value is 8), top to bottom, left to right. The picture has the minimal size to cover both the texture and the displacement data and can be processed directly by a standard compliant coding system. All the others are of padding area, in which an encoder pad data according some methods.

5.3 Embodiment 3

[0158] The following modifications are based on [7].

[0159] Most relevant parts that have been added or modified are indicated in bold typeface, and some of the deleted parts are indicated in bold italic typeface. There may be some other changes that are editorial in nature and thus not indicated. There may also be other changes that are not indicated.

8.4.2.2 V3C unit header semantics

Table 2 - V3C unit types

vuh_unit_type	Identifier	V3C unit type	Description
0	V3C_VPS	V3C parameter set	V3C level parameters
1	V3C_AD	Atlas data	Atlas information
2	V3C_OVD	Occupancy video data	Occupancy information
3	V3C_GVD	Geometry video data	Geometry information

4	V3C_AVD	Attribute video data	Attribute information
5	V3C_PVD	Packed video data	Packing information
6	V3C_CAD	Common atlas data	Information that is common for atlases in a CVS. Specified in ISO/IEC 23090-12
7	**V3C_BMD**	**Base mesh data**	**Base mesh information**
78…31	V3C_RSVD	Reserved	-

6. References

[1] MPEG technical requirements, "CfP for. Dynamic Mesh Coding," ISO/IEC JTC 1/SC 29/WG 2 doc. no. N145, in Oct. 2021.

[2] K. Mammou, J. Kim, A. Tourapis and D. Podborski, "[V-CG] Apple's Dynamic Mesh Coding CfP Response," ISO/IEC JTC 1/SC 29/WG 7 doc. no. m59281, in Apr. 2022.

[3] MPEG output document, "WD 1.0 of V-DMC," ISO/IEC JTC 1/SC 29/WG 7 doc. no. N0486, in Nov. 2022.

[4] C. Huang, X. Xu, X. Zhang, J. Tian and S. Liu, "Investigation of video coding of motion fields," ISO/IEC JTC 1/SC 29/WG 7 doc. no. m61005, in Jul. 2022.

[5] "WD 3.0 of V-DMC," ISO/IEC JTC 1/SC 29/WG 7 doc. no. N611, in Apr. 2023.

[6] ITU-T recommendation H.273 : Coding-independent code points for video signal type identification, can be accessed via https://www.itu.int/rec/T-REC-H.273.

[7] "Text of ISO/IEC FDIS 23090-5 2nd Edition Visual volumetric video-based coding (V3C) and video-based point cloud compression (V-PCC)," ISO/IEC JTC 1/SC 29/WG 7 doc. no. N553, in Jan. 2023.

[0160] FIG. 5 is a block diagram showing an example video processing system 4000 in which various techniques disclosed herein may be implemented. Various implementations may include some or all of the components of the system 4000. The system 4000 may include input 4002 for receiving video content. The video content may be received in a raw or uncompressed format, e.g., 8 or 10 bit multi-component pixel values, or may be in a compressed or encoded format. The input 4002 may represent a network interface, a peripheral bus interface, or a storage interface. Examples

of network interface include wired interfaces such as Ethernet, passive optical network (PON), etc. and wireless interfaces such as wireless fidelity (Wi-Fi) or cellular interfaces.

[0161] The system 4000 may include a coding component 4004 that may implement the various coding or encoding methods described in the present disclosure. The coding component 4004 may reduce the average bitrate of video from the input 4002 to the output of the coding component 4004 to produce a coded representation of the video. The coding techniques are therefore sometimes called video compression or video transcoding techniques. The output of the coding component 4004 may be either stored, or transmitted via a communication connected, as represented by the component 4006. The stored or communicated bitstream (or coded) representation of the video received at the input 4002 may be used by a component 4008 for generating pixel values or displayable video that is sent to a display interface 4010. The process of generating user-viewable video from the bitstream representation is sometimes called video decompression. Furthermore, while certain video processing operations are referred to as "coding" operations or tools, it will be appreciated that the coding tools or operations are used at an encoder and corresponding decoding tools or operations that reverse the results of the coding will be performed by a decoder.

[0162] Examples of a peripheral bus interface or a display interface may include universal serial bus (USB) or high definition multimedia interface (HDMI) or Displayport, and so on. Examples of storage interfaces include serial advanced technology attachment (SATA), peripheral component interconnect (PCI), integrated drive electronics (IDE) interface, and the like. The techniques described in the present disclosure may be embodied in various electronic devices such as mobile phones, laptops, smartphones or other devices that are capable of performing digital data processing and/or video display.

[0163] FIG. 6 is a block diagram of an example video processing apparatus 4100. The apparatus 4100 may be used to implement one or more of the methods described herein. The apparatus 4100 may be embodied in a smartphone, tablet, computer, Internet of Things (IoT) receiver, and so on. The apparatus 4100 may include one or more processors 4102, one or more memories 4104 and video processing circuitry 4106. The processor(s) 4102 may be configured to implement one or more methods described in the present disclosure. The memory (memories) 4104 may be used for storing data and code used for implementing the methods and techniques described herein. The video processing circuitry 4106 may be used to implement, in hardware circuitry, some techniques

described in the present disclosure. In some embodiments, the video processing circuitry 4106 may be at least partly included in the processor 4102, e.g., a graphics co-processor.

[0164] FIG. 7 is a flowchart for an example method 4200 of video processing. The method 4200 includes determining texture and displacement data are combined into one set of video media data at step 4202. A conversion is performed between a visual media data and a bitstream based on the combined texture and displacement data at step 4204. The conversion of step 4204 may include encoding at an encoder or decoding at a decoder, depending on the example.

[0165] It should be noted that the method 4200 can be implemented in an apparatus for processing video data comprising a processor and a non-transitory memory with instructions thereon, such as video encoder 4400, video decoder 4500, and/or encoder 4600. In such a case, the instructions upon execution by the processor, cause the processor to perform the method 4200. Further, the method 4200 can be performed by a non-transitory computer readable medium comprising a computer program product for use by a video coding device. The computer program product comprises computer executable instructions stored on the non-transitory computer readable medium such that when executed by a processor cause the video coding device to perform the method 4200.

[0166] FIG. 8 is a block diagram that illustrates an example video coding system 4300 that may utilize the techniques of this disclosure. The video coding system 4300 may include a source device 4310 and a destination device 4320. Source device 4310 generates encoded video data which may be referred to as a video encoding device. Destination device 4320 may decode the encoded video data generated by source device 4310 which may be referred to as a video decoding device.

[0167] Source device 4310 may include a video source 4312, a video encoder 4314, and an input/output (I/O) interface 4316. Video source 4312 may include a source such as a video capture device, an interface to receive video data from a video content provider, and/or a computer graphics system for generating video data, or a combination of such sources. The video data may comprise one or more pictures. Video encoder 4314 encodes the video data from video source 4312 to generate a bitstream. The bitstream may include a sequence of bits that form a coded representation of the video data. The bitstream may include coded pictures and associated data. The coded picture is a coded representation of a picture. The associated data may include sequence parameter sets, picture parameter sets, and other syntax structures. I/O interface 4316 may include a modulator/demodulator (modem) and/or a transmitter. The encoded video data may be transmitted directly to destination

device 4320 via I/O interface 4316 through network 4330. The encoded video data may also be stored onto a storage medium/server 4340 for access by destination device 4320.

[0168] Destination device 4320 may include an I/O interface 4326, a video decoder 4324, and a display device 4322. I/O interface 4326 may include a receiver and/or a modem. I/O interface 4326 may acquire encoded video data from the source device 4310 or the storage medium/ server 4340. Video decoder 4324 may decode the encoded video data. Display device 4322 may display the decoded video data to a user. Display device 4322 may be integrated with the destination device 4320, or may be external to destination device 4320, which can be configured to interface with an external display device.

[0169] Video encoder 4314 and video decoder 4324 may operate according to a video compression standard, such as the High Efficiency Video Coding (HEVC) standard, Versatile Video Coding (VVC) standard and other current and/or further standards.

[0170] FIG. 9 is a block diagram illustrating an example of video encoder 4400, which may be video encoder 4314 in the system 4300 illustrated in FIG. 8. Video encoder 4400 may be configured to perform any or all of the techniques of this disclosure. The video encoder 4400 includes a plurality of functional components. The techniques described in this disclosure may be shared among the various components of video encoder 4400. In some examples, a processor may be configured to perform any or all of the techniques described in this disclosure.

[0171] The functional components of video encoder 4400 may include a partition unit 4401, a prediction unit 4402 which may include a mode select unit 4403, a motion estimation unit 4404, a motion compensation unit 4405, an intra prediction unit 4406, a residual generation unit 4407, a transform processing unit 4408, a quantization unit 4409, an inverse quantization unit 4410, an inverse transform unit 4411, a reconstruction unit 4412, a buffer 4413, and an entropy encoding unit 4414.

[0172] In other examples, video encoder 4400 may include more, fewer, or different functional components. In an example, prediction unit 4402 may include an intra block copy (IBC) unit. The IBC unit may perform prediction in an IBC mode in which at least one reference picture is a picture where the current video block is located.

[0173] Furthermore, some components, such as motion estimation unit 4404 and motion compensation unit 4405 may be highly integrated, but are represented in the example of video encoder 4400 separately for purposes of explanation.

[0174] Partition unit 4401 may partition a picture into one or more video blocks. Video encoder 4400 and video decoder 4500 may support various video block sizes.

[0175] Mode select unit 4403 may select one of the coding modes, intra or inter, e.g., based on error results, and provide the resulting intra or inter coded block to a residual generation unit 4407 to generate residual block data and to a reconstruction unit 4412 to reconstruct the encoded block for use as a reference picture. In some examples, mode select unit 4403 may select a combination of intra and inter prediction (CIIP) mode in which the prediction is based on an inter prediction signal and an intra prediction signal. Mode select unit 4403 may also select a resolution for a motion vector (e.g., a sub-pixel or integer pixel precision) for the block in the case of inter prediction.

[0176] To perform inter prediction on a current video block, motion estimation unit 4404 may generate motion information for the current video block by comparing one or more reference frames from buffer 4413 to the current video block. Motion compensation unit 4405 may determine a predicted video block for the current video block based on the motion information and decoded samples of pictures from buffer 4413 other than the picture associated with the current video block.

[0177] Motion estimation unit 4404 and motion compensation unit 4405 may perform different operations for a current video block, for example, depending on whether the current video block is in an I slice, a P slice, or a B slice.

[0178] In some examples, motion estimation unit 4404 may perform uni-directional prediction for the current video block, and motion estimation unit 4404 may search reference pictures of list 0 or list 1 for a reference video block for the current video block. Motion estimation unit 4404 may then generate a reference index that indicates the reference picture in list 0 or list 1 that contains the reference video block and a motion vector that indicates a spatial displacement between the current video block and the reference video block. Motion estimation unit 4404 may output the reference index, a prediction direction indicator, and the motion vector as the motion information of the current video block. Motion compensation unit 4405 may generate the predicted video block of the current block based on the reference video block indicated by the motion information of the current video block.

[0179] In other examples, motion estimation unit 4404 may perform bi-directional prediction for the current video block, motion estimation unit 4404 may search the reference pictures in list 0 for a reference video block for the current video block and may also search the reference pictures in list 1 for another reference video block for the current video block. Motion estimation unit 4404 may

then generate reference indexes that indicate the reference pictures in list 0 and list 1 containing the reference video blocks and motion vectors that indicate spatial displacements between the reference video blocks and the current video block. Motion estimation unit 4404 may output the reference indexes and the motion vectors of the current video block as the motion information of the current video block. Motion compensation unit 4405 may generate the predicted video block of the current video block based on the reference video blocks indicated by the motion information of the current video block.

[0180] In some examples, motion estimation unit 4404 may output a full set of motion information for decoding processing of a decoder. In some examples, motion estimation unit 4404 may not output a full set of motion information for the current video. Rather, motion estimation unit 4404 may signal the motion information of the current video block with reference to the motion information of another video block. For example, motion estimation unit 4404 may determine that the motion information of the current video block is sufficiently similar to the motion information of a neighboring video block.

[0181] In one example, motion estimation unit 4404 may indicate, in a syntax structure associated with the current video block, a value that indicates to the video decoder 4500 that the current video block has the same motion information as another video block.

[0182] In another example, motion estimation unit 4404 may identify, in a syntax structure associated with the current video block, another video block and a motion vector difference (MVD). The motion vector difference indicates a difference between the motion vector of the current video block and the motion vector of the indicated video block. The video decoder 4500 may use the motion vector of the indicated video block and the motion vector difference to determine the motion vector of the current video block.

[0183] As discussed above, video encoder 4400 may predictively signal the motion vector. Two examples of predictive signaling techniques that may be implemented by video encoder 4400 include advanced motion vector prediction (AMVP) and merge mode signaling.

[0184] Intra prediction unit 4406 may perform intra prediction on the current video block. When intra prediction unit 4406 performs intra prediction on the current video block, intra prediction unit 4406 may generate prediction data for the current video block based on decoded samples of other video blocks in the same picture. The prediction data for the current video block may include a predicted video block and various syntax elements.

[0185] Residual generation unit 4407 may generate residual data for the current video block by subtracting the predicted video block(s) of the current video block from the current video block. The residual data of the current video block may include residual video blocks that correspond to different sample components of the samples in the current video block.

[0186] In other examples, there may be no residual data for the current video block for the current video block, for example in a skip mode, and residual generation unit 4407 may not perform the subtracting operation.

[0187] Transform processing unit 4408 may generate one or more transform coefficient video blocks for the current video block by applying one or more transforms to a residual video block associated with the current video block.

[0188] After transform processing unit 4408 generates a transform coefficient video block associated with the current video block, quantization unit 4409 may quantize the transform coefficient video block associated with the current video block based on one or more quantization parameter (QP) values associated with the current video block.

[0189] Inverse quantization unit 4410 and inverse transform unit 4411 may apply inverse quantization and inverse transforms to the transform coefficient video block, respectively, to reconstruct a residual video block from the transform coefficient video block. Reconstruction unit 4412 may add the reconstructed residual video block to corresponding samples from one or more predicted video blocks generated by the prediction unit 4402 to produce a reconstructed video block associated with the current block for storage in the buffer 4413.

[0190] After reconstruction unit 4412 reconstructs the video block, the loop filtering operation may be performed to reduce video blocking artifacts in the video block.

[0191] Entropy encoding unit 4414 may receive data from other functional components of the video encoder 4400. When entropy encoding unit 4414 receives the data, entropy encoding unit 4414 may perform one or more entropy encoding operations to generate entropy encoded data and output a bitstream that includes the entropy encoded data.

[0192] FIG. 10 is a block diagram illustrating an example of video decoder 4500 which may be video decoder 4324 in the system 4300 illustrated in FIG. 8. The video decoder 4500 may be configured to perform any or all of the techniques of this disclosure. In the example shown, the video decoder 4500 includes a plurality of functional components. The techniques described in this disclosure may be shared among the various components of the video decoder 4500. In some

examples, a processor may be configured to perform any or all of the techniques described in this disclosure.

[0193] In the example shown, video decoder 4500 includes an entropy decoding unit 4501, a motion compensation unit 4502, an intra prediction unit 4503, an inverse quantization unit 4504, an inverse transformation unit 4505, a reconstruction unit 4506, and a buffer 4507. Video decoder 4500 may, in some examples, perform a decoding pass generally reciprocal to the encoding pass described with respect to video encoder 4400.

[0194] Entropy decoding unit 4501 may retrieve an encoded bitstream. The encoded bitstream may include entropy coded video data (e.g., encoded blocks of video data). Entropy decoding unit 4501 may decode the entropy coded video data, and from the entropy decoded video data, motion compensation unit 4502 may determine motion information including motion vectors, motion vector precision, reference picture list indexes, and other motion information. Motion compensation unit 4502 may, for example, determine such information by performing the AMVP and merge mode.

[0195] Motion compensation unit 4502 may produce motion compensated blocks, possibly performing interpolation based on interpolation filters. Identifiers for interpolation filters to be used with sub-pixel precision may be included in the syntax elements.

[0196] Motion compensation unit 4502 may use interpolation filters as used by video encoder 4400 during encoding of the video block to calculate interpolated values for sub-integer pixels of a reference block. Motion compensation unit 4502 may determine the interpolation filters used by video encoder 4400 according to received syntax information and use the interpolation filters to produce predictive blocks.

[0197] Motion compensation unit 4502 may use some of the syntax information to determine sizes of blocks used to encode frame(s) and/or slice(s) of the encoded video sequence, partition information that describes how each macroblock of a picture of the encoded video sequence is partitioned, modes indicating how each partition is encoded, one or more reference frames (and reference frame lists) for each inter coded block, and other information to decode the encoded video sequence.

[0198] Intra prediction unit 4503 may use intra prediction modes for example received in the bitstream to form a prediction block from spatially adjacent blocks. Inverse quantization unit 4504 inverse quantizes, i.e., de-quantizes, the quantized video block coefficients provided in the bitstream

and decoded by entropy decoding unit 4501. Inverse transform unit 4505 applies an inverse transform.

[0199] Reconstruction unit 4506 may sum the residual blocks with the corresponding prediction blocks generated by motion compensation unit 4502 or intra prediction unit 4503 to form decoded blocks. If desired, a deblocking filter may also be applied to filter the decoded blocks in order to remove blockiness artifacts. The decoded video blocks are then stored in buffer 4507, which provides reference blocks for subsequent motion compensation/intra prediction and also produces decoded video for presentation on a display device.

[0200] FIG. 11 is a schematic diagram of an example encoder 4600. The encoder 4600 is suitable for implementing the techniques of VVC. The encoder 4600 includes three in-loop filters, namely a deblocking filter (DF) 4602, a sample adaptive offset (SAO) 4604, and an adaptive loop filter (ALF) 4606. Unlike the DF 4602, which uses predefined filters, the SAO 4604 and the ALF 4606 utilize the original samples of the current picture to reduce the mean square errors between the original samples and the reconstructed samples by adding an offset and by applying a finite impulse response (FIR) filter, respectively, with coded side information signaling the offsets and filter coefficients. The ALF 4606 is located at the last processing stage of each picture and can be regarded as a tool trying to catch and fix artifacts created by the previous stages.

[0201] The encoder 4600 further includes an intra prediction component 4608 and a motion estimation/compensation (ME/MC) component 4610 configured to receive input video. The intra prediction component 4608 is configured to perform intra prediction, while the ME/MC component 4610 is configured to utilize reference pictures obtained from a reference picture buffer 4612 to perform inter prediction. Residual blocks from inter prediction or intra prediction are fed into a transform (T) component 4614 and a quantization (Q) component 4616 to generate quantized residual transform coefficients, which are fed into an entropy coding component 4618. The entropy coding component 4618 entropy codes the prediction results and the quantized transform coefficients and transmits the same toward a video decoder (not shown). Quantization components output from the quantization component 4616 may be fed into an inverse quantization (IQ) components 4620, an inverse transform component 4622, and a reconstruction (REC) component 4624. The REC component 4624 is able to output images to the DF 4602, the SAO 4604, and the ALF 4606 for filtering prior to those images being stored in the reference picture buffer 4612.

[0202] FIG. 12 is a flowchart 5200 for an example method of video processing. In an embodiment, the method is performed by an encoder (e.g., encoding device) or by a decoder (e.g., a decoding device). In block 5202, the method includes determining that the displacement data is coded in a 4:2:0 format. In block 5204, the method includes performing a conversion between the visual media data and a bitstream based on the displacement data as coded in the 4:2:0 format. The conversion of step 5204 may include encoding at an encoder or decoding at a decoder, depending on the example.

[0203] A listing of solutions preferred by some examples is provided next.

[0204] The following solutions show examples of techniques discussed herein.

[0205] 1. A method for processing video data comprising: determining texture and displacement data are combined into one set of video media data; and performing a conversion between the visual media data and a bitstream based on the combined texture and displacement data.

[0206] 2. The method of solution 1, wherein the displacement data is converted to 4:2:0 format and concatenated with the texture in 4:2:0 format.

[0207] 3. The method of any of solutions 1-2, wherein the displacement data is converted to an N-bit bitdepth that is equal to a bitdepth of the texture.

[0208] 4. The method of any of solutions 1-3, wherein the N-bit bitdepth is a bit depth of 10.

[0209] 5. The method of any of solutions 1-4, wherein texture and displacement data are coded in different slices.

[0210] 6. The method of any of solutions 1-5, wherein positions of a texture area, sizes of the texture area, or a displacement data area are signalled in the bitstream.

[0211] 7. The method of any of solutions 1-6, wherein positions of a texture area, sizes of the texture area, or a displacement data area are inferred from the bitstream.

[0212] 8. The method of any of solutions 1-7, wherein the texture and displacement data are coded using different coding mechanisms.

[0213] 9. The method of any of solutions 1-8, wherein the texture and displacement data are coded using different quantization parameters (QPs).

[0214] 10. The method of any of solutions 1-9, wherein the displacement data are coded using lossless coding.

[0215] 11. The method of any of solutions 1-10, wherein the displacement data are coded using transquant_bypass mode.

[0216] 12. The method of any of solutions 1-11, wherein the displacement data are coded using transform skip mode and QP equal to 4+6*K, wherein k is equal to zero.

[0217] 13. The method of any of solutions 1-12, wherein an area of a picture including the texture and displacement data is padded.

[0218] 14. The method of any of solutions 1-13, wherein the area is padded with a fixed value.

[0219] 15. The method of any of solutions 1-14, wherein the area is padded with a middle pixel value.

[0220] 16. The method of any of solutions 1-15, wherein the area is padded with a nearest pixel value.

[0221] 17. The method of any of solutions 1-16, wherein a smoothing process is applied to the picture including the texture and displacement data.

[0222] 18. The method of any of solutions 1-17, wherein the smoothing process is applied only to a padded area in the picture including the texture and displacement data.

[0223] 19. The method of any of solutions 1-18, wherein displacement data is coded in 4:2:0 format.

[0224] 20. The method of any of solutions 1-19, wherein the displacement data is converted from 4:4:4 to 4:2:0 when the conversion includes encoding, and wherein the displacement data is converted from back 4:2:0 to 4:4:4 when the conversion includes decoding.

[0225] 21. The method of any of solutions 1-20, wherein sps_chroma_format_idc or ChromaFormatIdc are set to 1 for displacement data coding.

[0226] 22. The method of any of solutions 1-21, wherein displacement data is coded using a main or main10 profile.

[0227] 23. An apparatus for processing video data comprising: a processor; and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to perform the method of any of solutions 1-22.

[0228] 24. A non-transitory computer readable medium comprising a computer program product for use by a video coding device, the computer program product comprising computer executable instructions stored on the non-transitory computer readable medium such that when executed by a processor cause the video coding device to perform the method of any of solutions 1-22.

[0229] 25. A non-transitory computer-readable recording medium storing a bitstream of a video which is generated by a method performed by a video processing apparatus, wherein the method comprises: determining texture and displacement data are combined into one set of video media data; and generating the bitstream based on the determining.

[0230] 26. A method for storing bitstream of a video comprising: determining texture and displacement data are combined into one set of video media data; generating the bitstream based on the determining; and storing the bitstream in a non-transitory computer-readable recording medium.

[0231] 27. A method, apparatus, or system described in the present disclosure.

[0232] In the solutions described herein, an encoder may conform to the format rule by producing a coded representation according to the format rule. In the solutions described herein, a decoder may use the format rule to parse syntax elements in the coded representation with the knowledge of presence and absence of syntax elements according to the format rule to produce decoded video.

[0233] In the present disclosure, the term "video processing" may refer to video encoding, video decoding, video compression or video decompression. For example, video compression algorithms may be applied during conversion from pixel representation of a video to a corresponding bitstream representation or vice versa. The bitstream representation of a current video block may, for example, correspond to bits that are either co-located or spread in different places within the bitstream, as is defined by the syntax. For example, a macroblock may be encoded in terms of transformed and coded error residual values and also using bits in headers and other fields in the bitstream. Furthermore, during conversion, a decoder may parse a bitstream with the knowledge that some fields may be present, or absent, based on the determination, as is described in the above solutions. Similarly, an encoder may determine that certain syntax fields are or are not to be included and generate the coded representation accordingly by including or excluding the syntax fields from the coded representation.

[0234] The disclosed and other solutions, examples, embodiments, modules and the functional operations described in this disclosure can be implemented in digital electronic circuitry, or in computer software, firmware, or hardware, including the structures disclosed in this disclosure and their structural equivalents, or in combinations of one or more of them. The disclosed and other embodiments can be implemented as one or more computer program products, i.e., one or more modules of computer program instructions encoded on a computer readable medium for execution

by, or to control the operation of, data processing apparatus. The computer readable medium can be a machine-readable storage device, a machine-readable storage substrate, a memory device, a composition of matter effecting a machine-readable propagated signal, or a combination of one or more them. The term "data processing apparatus" encompasses all apparatus, devices, and machines for processing data, including by way of example a programmable processor, a computer, or multiple processors or computers. The apparatus can include, in addition to hardware, code that creates an execution environment for the computer program in question, e.g., code that constitutes processor firmware, a protocol stack, a database management system, an operating system, or a combination of one or more of them. A propagated signal is an artificially generated signal, e.g., a machine-generated electrical, optical, or electromagnetic signal, that is generated to encode information for transmission to suitable receiver apparatus.

[0235] A computer program (also known as a program, software, software application, script, or code) can be written in any form of programming language, including compiled or interpreted languages, and it can be deployed in any form, including as a stand-alone program or as a module, component, subroutine, or other unit suitable for use in a computing environment. A computer program does not necessarily correspond to a file in a file system. A program can be stored in a portion of a file that holds other programs or data (e.g., one or more scripts stored in a markup language document), in a single file dedicated to the program in question, or in multiple coordinated files (e.g., files that store one or more modules, sub programs, or portions of code). A computer program can be deployed to be executed on one computer or on multiple computers that are located at one site or distributed across multiple sites and interconnected by a communication network.

[0236] The processes and logic flows described in this disclosure can be performed by one or more programmable processors executing one or more computer programs to perform functions by operating on input data and generating output. The processes and logic flows can also be performed by, and apparatus can also be implemented as, special purpose logic circuitry, e.g., a field programmable gate array (FPGA) or an application specific integrated circuit (ASIC).

[0237] Processors suitable for the execution of a computer program include, by way of example, both general and special purpose microprocessors, and any one or more processors of any kind of digital computer. Generally, a processor will receive instructions and data from a read only memory or a random-access memory or both. The essential elements of a computer are a processor for performing instructions and one or more memory devices for storing instructions and data.

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

[0238] While the present disclosure contains many specifics, these should not be construed as limitations on the scope of any subject matter or of what may be claimed, but rather as descriptions of features that may be specific to particular embodiments of particular techniques. Certain features that are described in the present disclosure in the context of separate embodiments can also be implemented in combination in a single embodiment. Conversely, various features that are described in the context of a single embodiment can also be implemented in multiple embodiments separately or in any suitable subcombination. Moreover, although features may be described above as acting in certain combinations and even initially claimed as such, one or more features from a claimed combination can in some cases be excised from the combination, and the claimed combination may be directed to a subcombination or variation of a subcombination.

[0239] Similarly, while operations are depicted in the drawings in a particular order, this should not be understood as requiring that such operations be performed in the particular order shown or in sequential order, or that all illustrated operations be performed, to achieve desirable results. Moreover, the separation of various system components in the embodiments described in the present disclosure should not be understood as requiring such separation in all embodiments.

[0240] Only a few implementations and examples are described and other implementations, enhancements and variations can be made based on what is described and illustrated in the present disclosure.

[0241] A first component is directly coupled to a second component when there are no intervening components, except for a line, a trace, or another medium between the first component and the second component. The first component is indirectly coupled to the second component when

there are intervening components other than a line, a trace, or another medium between the first component and the second component. The term "coupled" and its variants include both directly coupled and indirectly coupled. The use of the term "about" means a range including ±10% of the subsequent number unless otherwise stated.

[0242] While several embodiments have been provided in the present disclosure, it should be understood that the disclosed systems and methods might be embodied in many other specific forms without departing from the spirit or scope of the present disclosure. The present examples are to be considered as illustrative and not restrictive, and the intention is not to be limited to the details given herein. For example, the various elements or components may be combined or integrated in another system or certain features may be omitted, or not implemented.

[0243] In addition, techniques, systems, subsystems, and methods described and illustrated in the various embodiments as discrete or separate may be combined or integrated with other systems, modules, techniques, or methods without departing from the scope of the present disclosure. Other items shown or discussed as coupled may be directly connected or may be indirectly coupled or communicating through some interface, device, or intermediate component whether electrically, mechanically, or otherwise. Other examples of changes, substitutions, and alterations are ascertainable by one skilled in the art and could be made without departing from the spirit and scope disclosed herein.

CLAIMS

What is claimed is:

1. A method for processing visual media data including displacement data, comprising:

determining that the displacement data is coded in a 4:2:0 format; and

performing a conversion between the visual media data and a bitstream based on the displacement data as coded in the 4:2:0 format.

2. The method of claim 1, wherein the displacement data is converted to the 4:2:0 format prior to encoding, and converted to a 4:4:4 format after decoding.

3. The method of claim 1, wherein one or more of an sps_chroma_format_idc syntax element and a ChromaFormatIdc variable is set to 1 for coding the displacement data.

4. The method of claim 1, wherein coding of the displacement data uses a main profile or a main10 profile.

5. The method of claim 1, wherein the displacement data is packed into luma components and chroma components in the 4:2:0 format when the displacement data has only one non-zero component.

6. The method of claim 1, wherein a subblock size of a subblock of the visual media data is represented by 2^S, where S is a non-negative integer.

7. The method of claim 6, wherein the bitstream includes a value equal to S-K, where K is a non-negative integer.

8. The method of claim 7, wherein the value in the bitstream is represented as an unsigned integer Exp-Golomb-coded syntax element (ue(v)) or as an unsigned integer using n bits (u(n)).

9. The method of any of claims 7-8, wherein K is equal to 0.

10. The method of any of claims 7-8, wherein K is equal to 6.

11. The method of any of claims 7-8, wherein K is equal to 7.

12. The method of any of claims 7-8, wherein the value of S-K is in a range of 0 to 3, inclusive.

13. The method of any of claims 1-12, wherein the video media data includes the displacement data and texture data, and wherein the texture data and the displacement data are included in a single bitstream.

14. The method of any of claims 1-13, further comprising converting the displacement data to a 4:2:0 format, and concatenating the displacement data as converted with the texture data in the 4:2:0 format.

15. The method of any of claims 1-13, further comprising converting the displacement data to N-bit, wherein the N-bit is a bitdepth of the texture data, and where N is an integer.

16. The method of claim 15, wherein N is 10.

17. The method of any of claims 1-16, wherein the texture data and the displacement data are coded in different slices.

18. The method of any of claims 1-17, wherein one or more of a position and a size of the texture data are included in the single bitstream.

19. The method of any of claims 1-17, wherein one or more of a position and a size of the displacement data are included in the single bitstream.

20. The method of any of claims 1-17, wherein one or more of a position and a size of the texture data are inferred based on information in the single bitstream.

21. The method of any of claims 1-17, wherein one or more of a position and a size of the displacement data are inferred based on information in the single bitstream.

22. The method of any of claims 1-21, wherein the texture data and the displacement data are included in a single bitstream and use different coding methods.

23. The method of claim 22, wherein the different coding methods comprise a first quantization parameter and a second quantization parameter different from the first quantization parameter, and wherein the texture data uses the first quantization parameter and the displacement data uses the second quantization parameter.

24. The method of any of claims 22-23, wherein the different coding methods comprise lossless coding, and wherein all video units of the displacement data use the lossless coding.

25. The method of any of claims 22-24, wherein the different coding methods comprise a transquant bypass mode from the high efficiency video coding (HEVC) standard, and wherein all video units of the displacement data use the transquant bypass mode.

26. The method of any of claims 22-25, wherein the different coding methods comprise a transform skip mode and a quantization parameter, and wherein all video units of the displacement data use the transform skip mode and the quantization parameter.

27. The method of claim 26, wherein the quantization parameter is equal to 4+6*K, where K is an integer.

28. The method of claim 27, wherein K has a value of zero.

29. The method of any of claims 1-28, further comprising padding a picture in the single bitstream using data other than the texture data and the displacement data.

30. The method of claim 29, wherein the picture is padded with a fixed value.

31. The method of any of claims 29-30, wherein the picture is padded with a middle pixel value, and wherein the middle pixel value is 128 for an 8-bit video or 512 for a 10-bit video.

32. The method of claim 29, wherein the picture is padded with a value of a nearest pixel in the texture data or with a value of a nearest pixel in the displacement data.

33. The method of claim 29, further comprising inserting N rows of luma samples and N/2 rows of chroma samples between the texture data and the displacement data when the picture is padded, where N is an integer.

34. The method of claim 33, wherein a value of N is 16.

35. The method of claim 33, wherein a value of N is 0.

36. The method of claim 33, wherein all samples in the N rows of luma samples and the N/2 rows of chroma samples have a same value.

37. The method of claim 36, wherein the same value comprises a middle pixel value.

38. The method of any of claims 1-37, further comprising applying a smoothing process to a picture in the single bitstream.

39. The method of claim 1, further comprising deriving, at a decoder, whether the displacement data has only one non-zero component or has three non-zero components.

40. The method of claim 1, further comprising determining, at a decoder, whether the displacement data has only one non-zero component or has three non-zero components based on a video resolution of the displacement data and a number of base mesh points.

41. The method of claim 1, further comprising determining, at a decoder, whether the displacement data has only one non-zero component or has three non-zero components based on a video resolution of the displacement data and a number of vertexes.

42. The method of claim 1, further comprising converting the texture data to a color space prior to encoding and converting the texture data back to the color space after decoding, wherein the color space is not a blue green red (BGR) color space or a YUV color space, where Y represents a luma component and U and V represent blue and red chroma components.

43. The method of claim 42, wherein the texture data in the bitstream is coded in a green blue red (GBR) color space.

44. The method of claim 42, wherein the texture data in the bitstream is coded in a green red blue (GRB) color space.

45. The method of claim 42, wherein the texture data in the bitstream is coded in a YCgCo color space, where Y represents a luma component, Cg represents a green chroma component, and Co represents an orange chroma component.

46. The method of claim 42, wherein the texture data in the bitstream is coded in a YCoCg color space, where Y represents a luma component, Co represents an orange chroma component, and Cg represents a green chroma component.

47. The method of claim 42, wherein the color space is used for losslessly coding the texture data.

48. The method of claim 1, further comprising an index of color primaries representing a green red blue (GRB) color space is used for coding according to the International Telecommunication Union - Telecommunication Standardization Sector (ITU-T) standard.

49. The method of claim 1, wherein a first subblock size (SB1) used to code a first color component of the visual media data is different from a second subblock size (SB2) used to code a second color component of the visual media data.

50. The method of claim 49, wherein the second subblock size is greater than the first subblock size.

51. The method of claim 50, wherein the first subblock size is 100 and the second subblock size is 200.

52. The method of claim 50, wherein the first subblock size is 128 and the second subblock size is 256.

53. The method of claim 1, wherein the displacement data comprises inter displacement data and intra displacement data, and wherein the inter displacement data uses a different block size than the intra displacement data.

54. The method of claim 1, wherein a pin occupancy present flag is equal to 0 when a packing information scheme is used to support jointly coding of the displacement data and texture data in dynamic mesh coding.

55. The method of claim 54, wherein a region of the visual media data corresponding to the pin occupancy present flag is ignored by a dynamic mesh decoder when the pin occupancy present flag is not equal to 0.

56. The method of any of claims 54-55, wherein the pin occupancy present flag is designated pin_occupancy_present_flag[j].

57. The method of claim 1, wherein a pin region type syntax element corresponding to the visual media data is not 0.

58. The method of claim 57, wherein the pin region type syntax element is designated pin_region_type_id_minus2[j][i].

59. The method of claim 54, wherein no region in the packing information scheme overlaps another region in the packing information scheme.

60. The method of any of claims 1-59, wherein the conversion includes encoding the media data into a bitstream.

61. The method of any of claims 1-59, wherein the conversion includes decoding the media data from a bitstream.

62. An apparatus for processing media data comprising: a processor; and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to perform the method of any of claims 1-61.

63. A non-transitory computer readable medium, comprising a computer program product for use by a video coding device, the computer program product comprising computer executable instructions stored on the non-transitory computer readable medium such that when executed by a processor cause the video coding device to perform the method of any of claims 1-61.

64. A non-transitory computer-readable recording medium storing a bitstream of a video which is generated by a method performed by a video processing apparatus, wherein the method comprises the method of any of claims 1-61.

65. A method for storing a bitstream of a video comprising the method of any of claims 1-61.

66. A method, apparatus, or system described in the present disclosure.

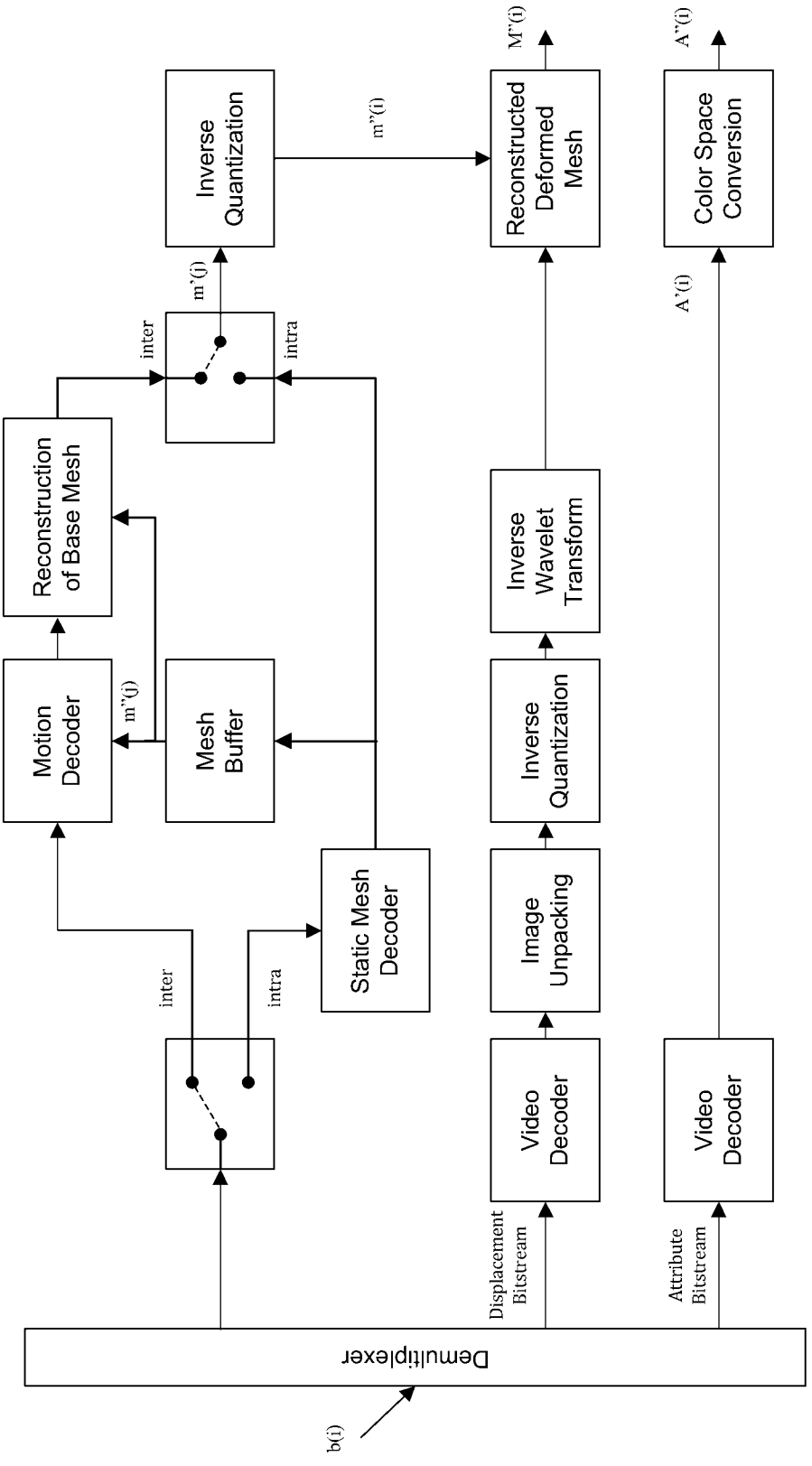
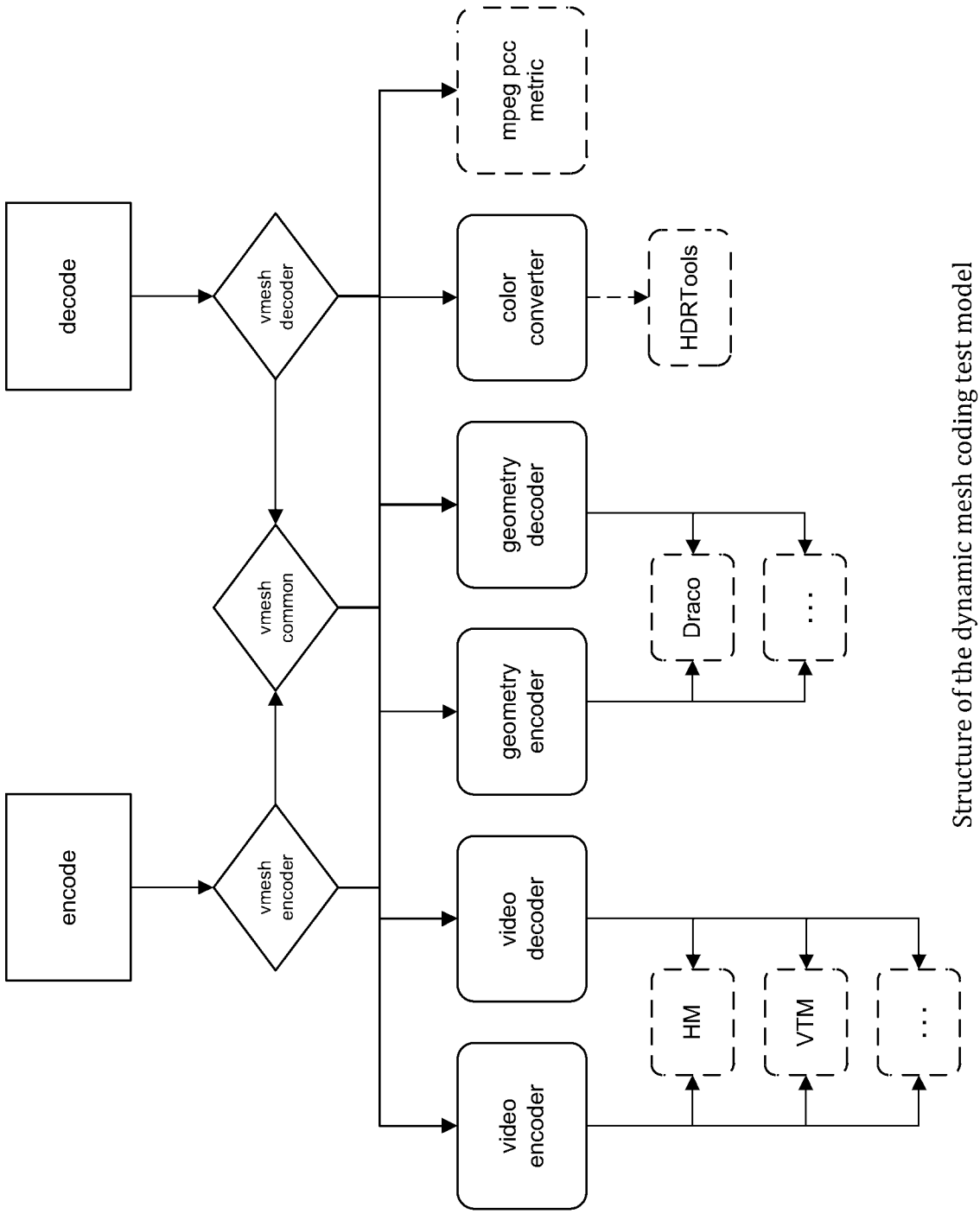


FIG. 1

Current decoder design of dynamic mesh coding



Structure of the dynamic mesh coding test model

FIG. 2

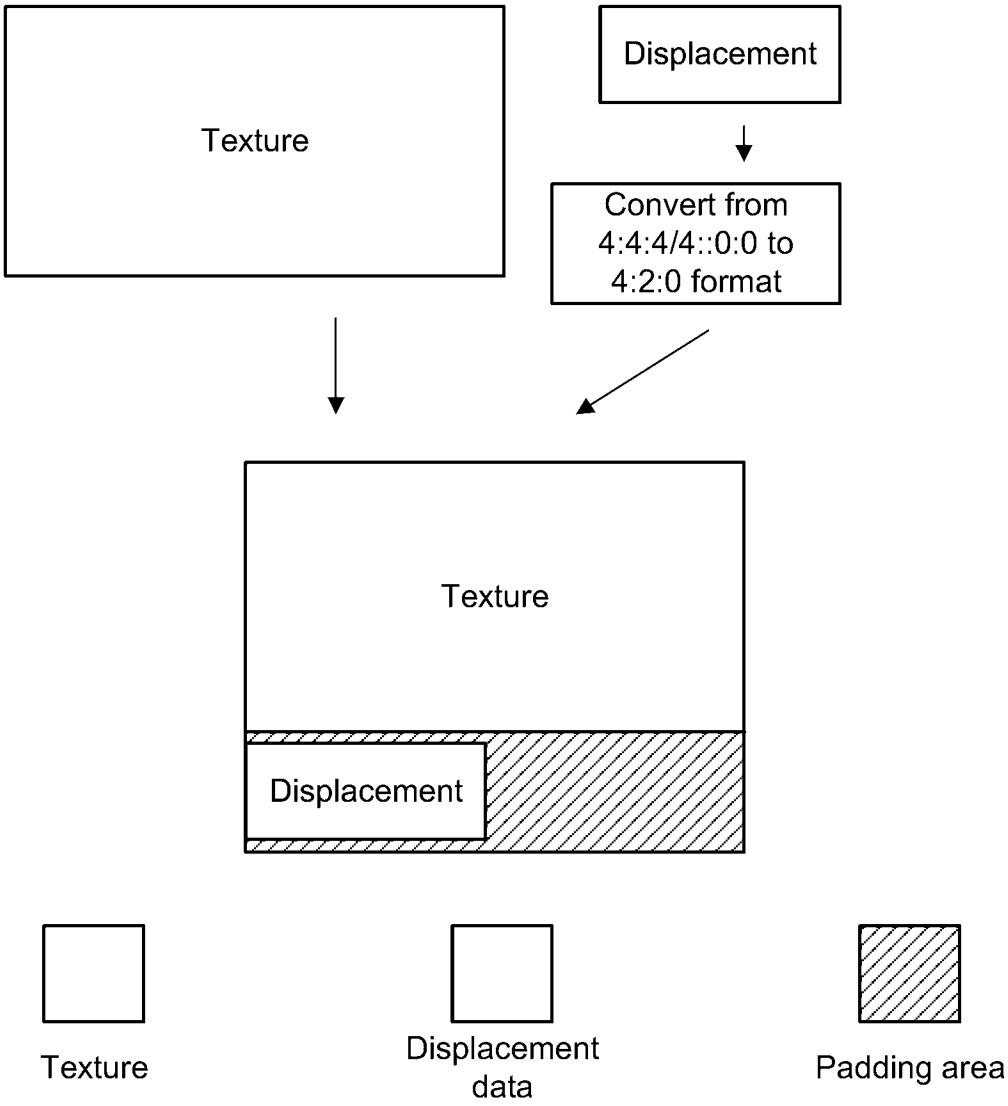

Texture
Displacement
Convert from 4:4:4/4::0:0 to 4:2:0 format
Texture
Displacement
Texture
Displacement data
Padding area

FIG. 3

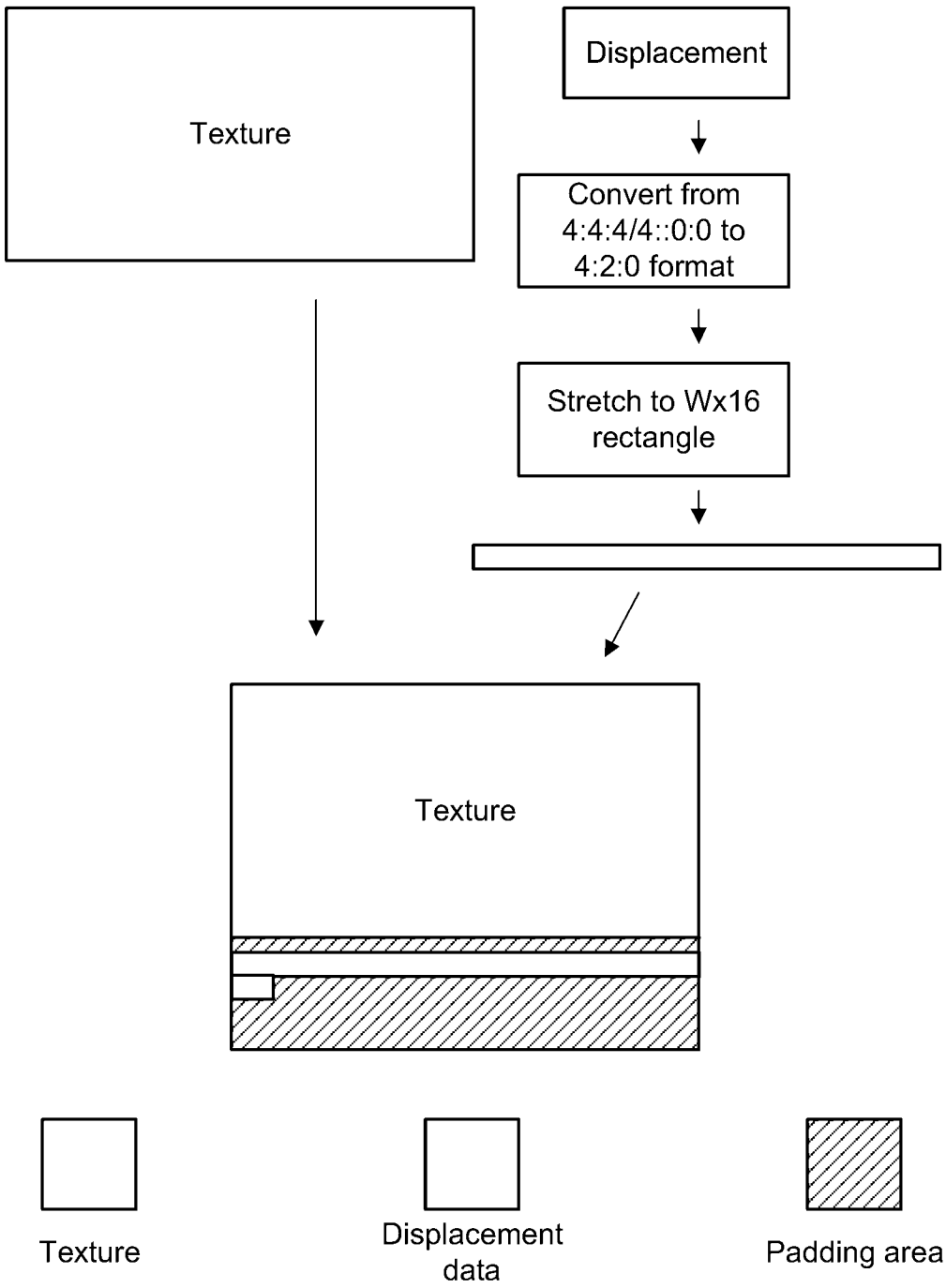

Texture
Displacement
Convert from 4:4:4/4::0:0 to 4:2:0 format
Stretch to Wx16 rectangle
Texture
Texture
Displacement data
Padding area

FIG. 4

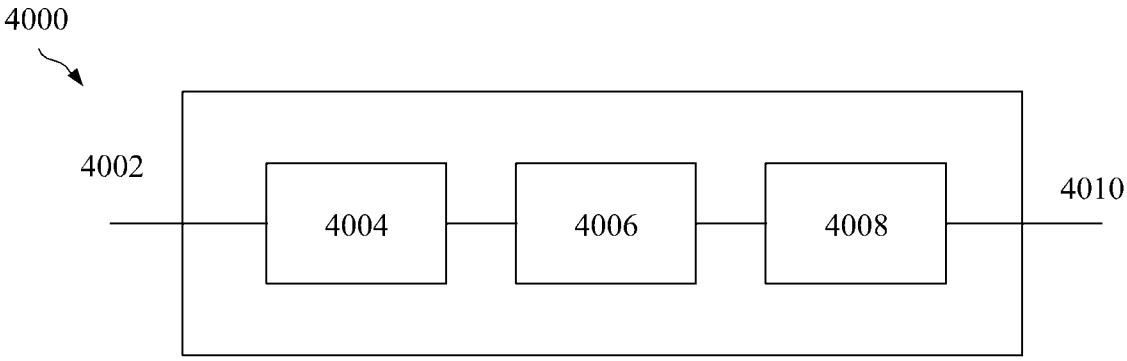


FIG. 5

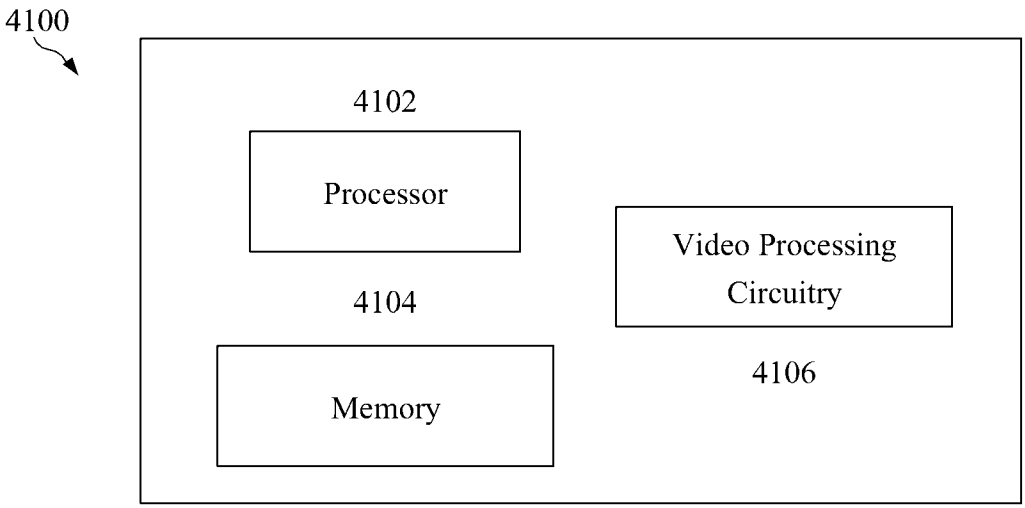


FIG.6

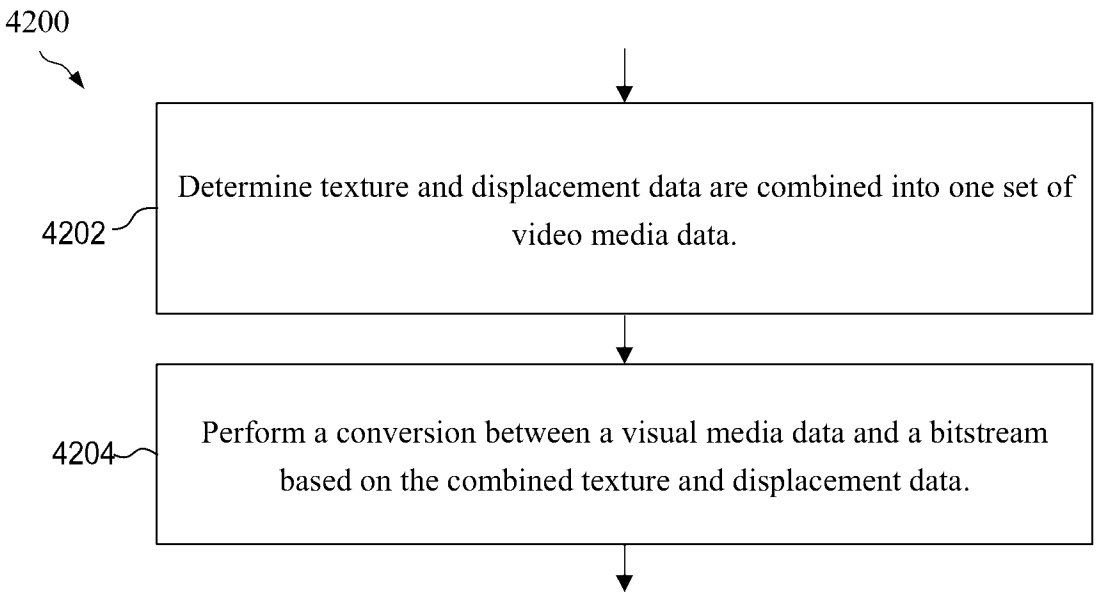


FIG. 7

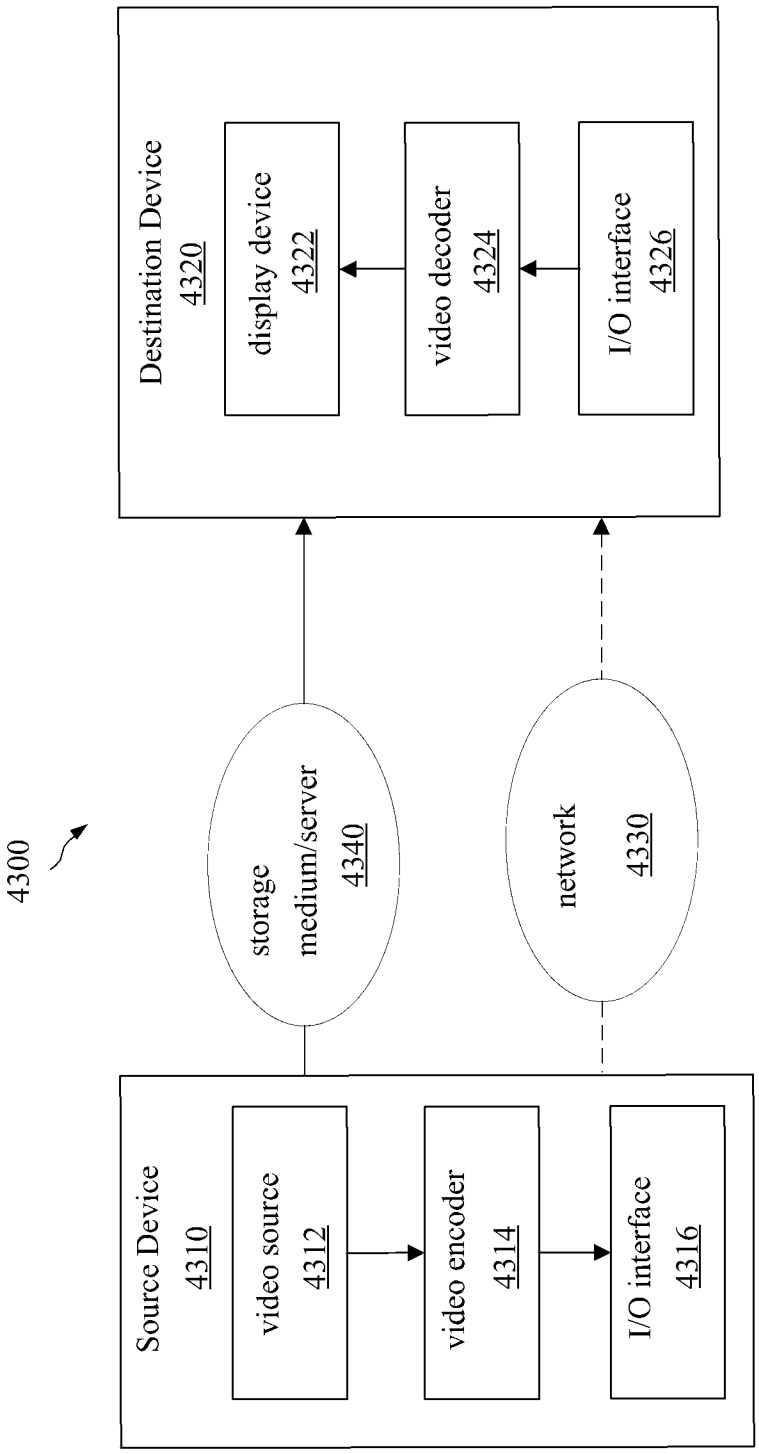

4300
Source Device
4310
video source
4312
video encoder
4314
I/O interface
4316
storage
medium/server
4340
network
4330
Destination Device
4320
display device
4322
video decoder
4324
I/O interface
4326

FIG. 8

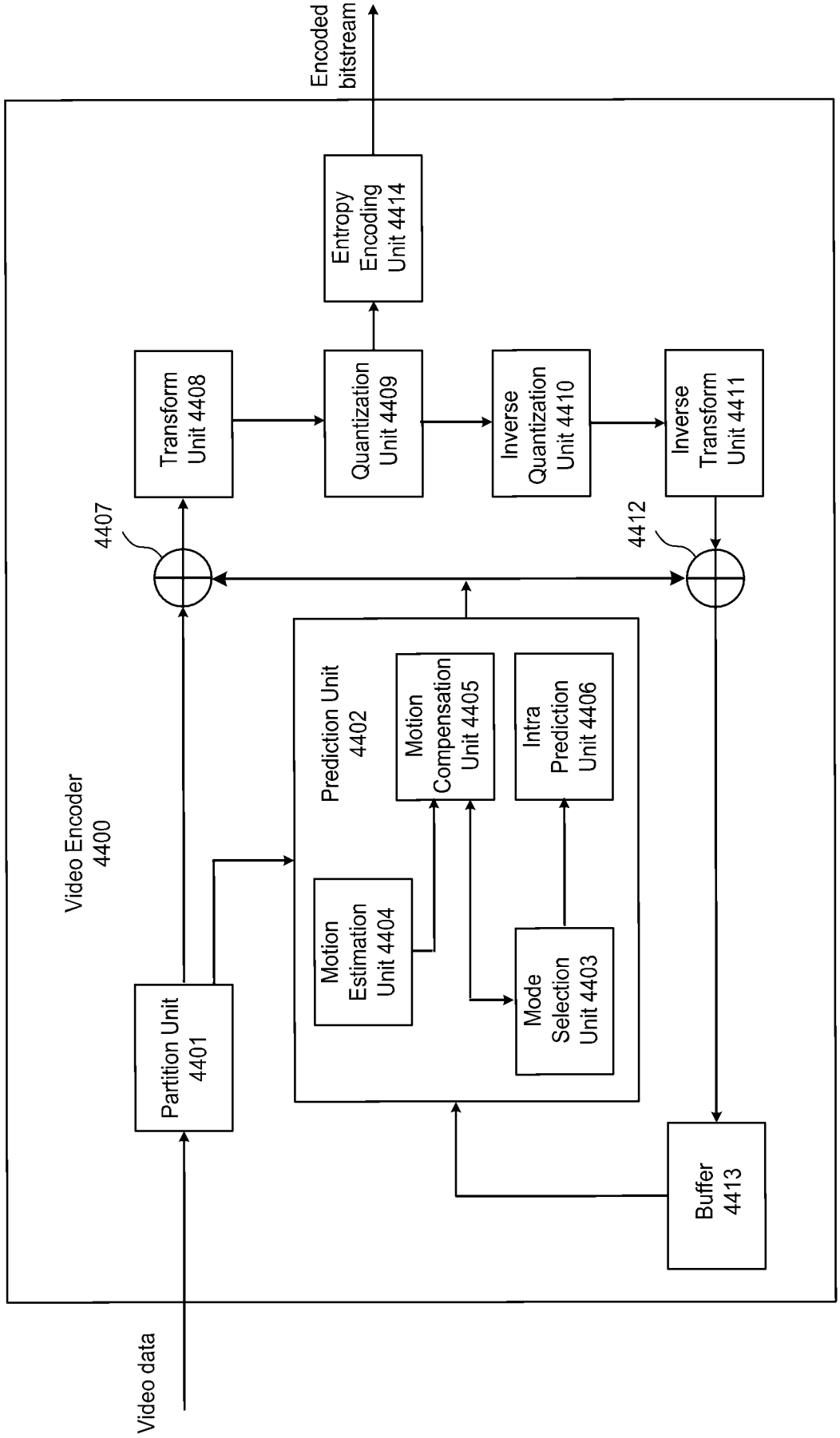

Video data
Video Encoder 4400
Partition Unit 4401
Prediction Unit 4402
Motion Estimation Unit 4404
Motion Compensation Unit 4405
Intra Prediction Unit 4406
Mode Selection Unit 4403
4407
Transform Unit 4408
Quantization Unit 4409
Entropy Encoding Unit 4414
Encoded bitstream
Inverse Quantization Unit 4410
Inverse Transform Unit 4411
4412
Buffer 4413

FIG. 9

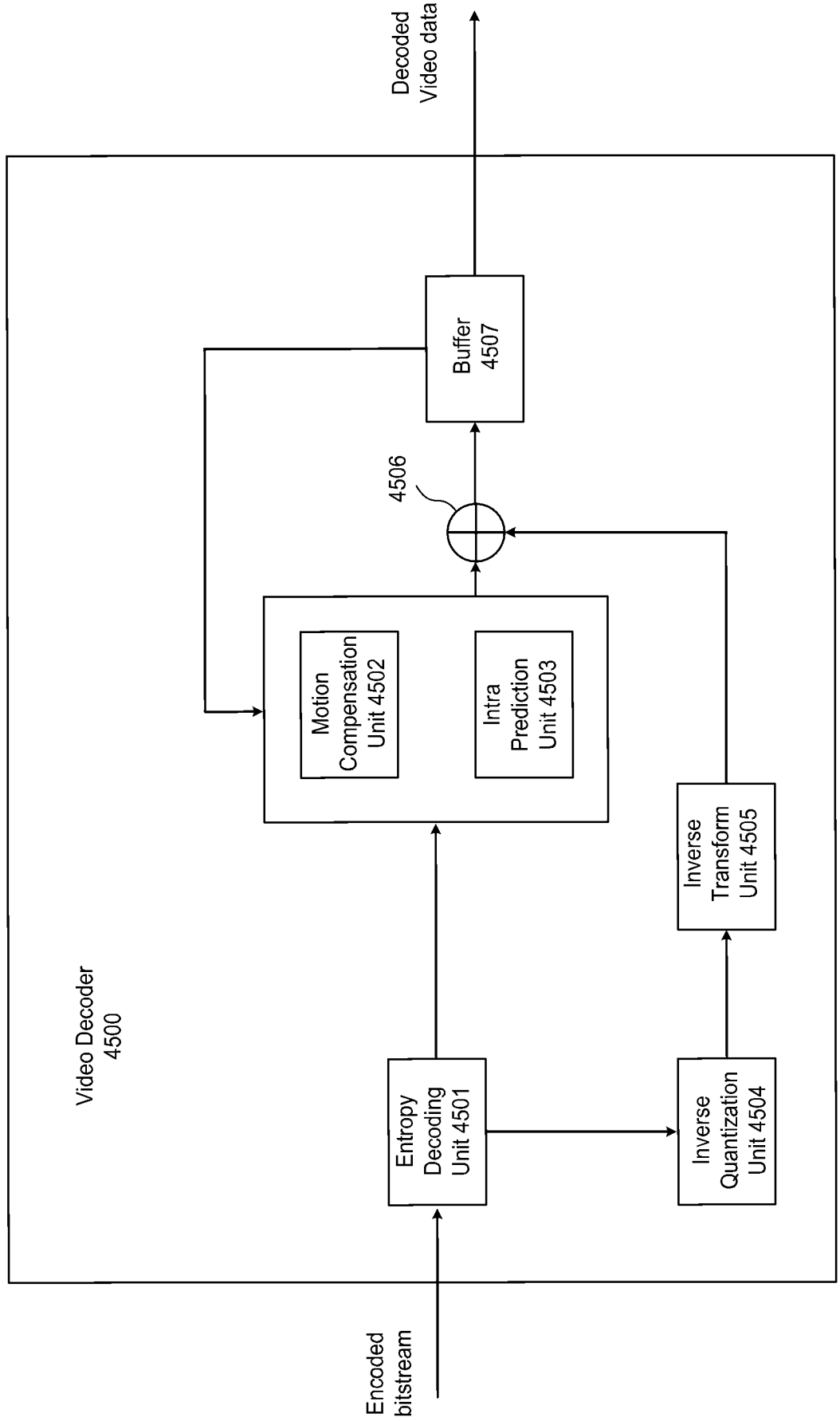

Decoded Video data
Buffer 4507
4506
Motion Compensation Unit 4502
Intra Prediction Unit 4503
Inverse Transform Unit 4505
Video Decoder 4500
Entropy Decoding Unit 4501
Inverse Quantization Unit 4504
Encoded bitstream

FIG. 10

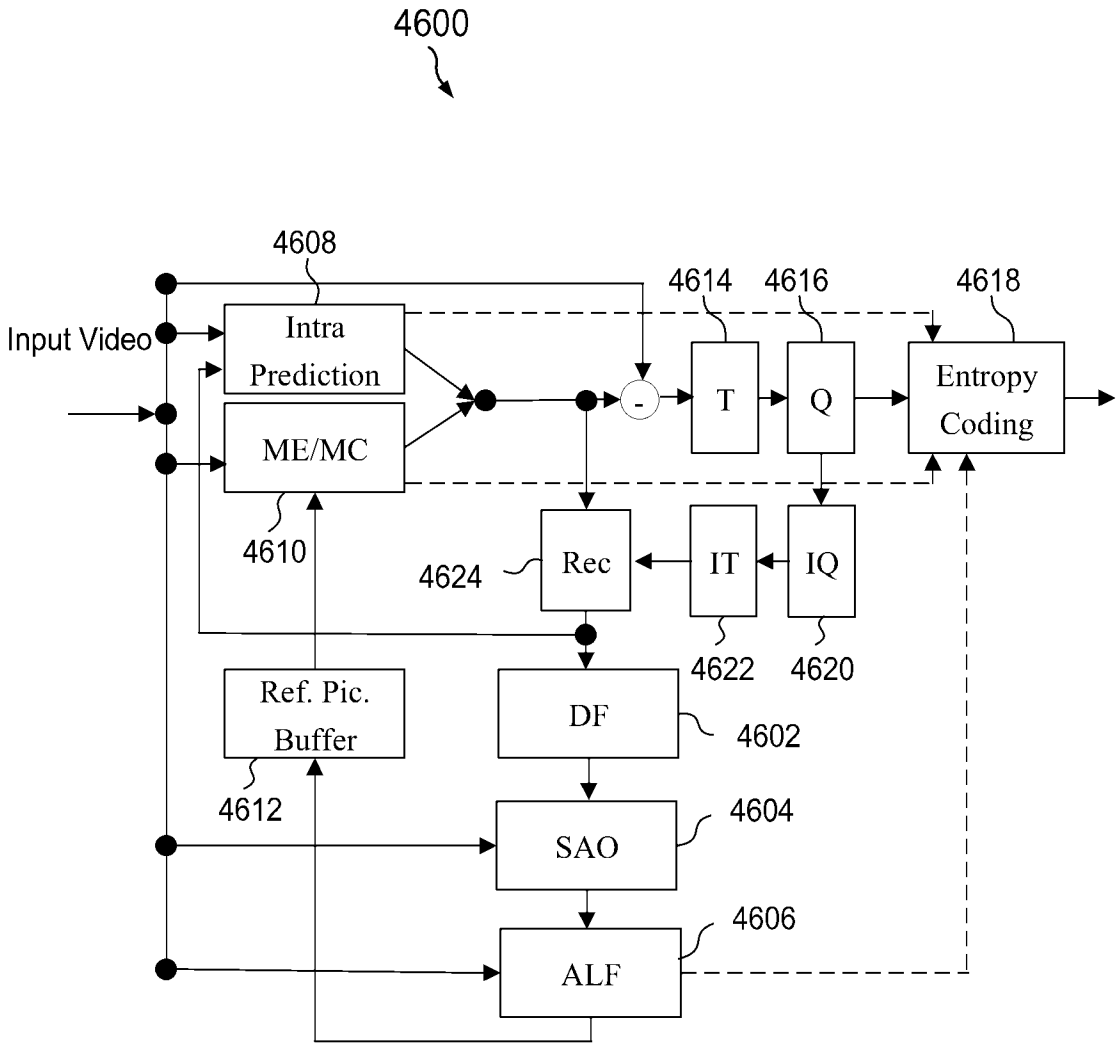

4600
4608
4614
4616
4618
Input Video
Intra Prediction
ME/MC
4610
-
T
Q
Entropy Coding
4624
Rec
IT
IQ
4622
4620
Ref. Pic. Buffer
4612
DF
4602
SAO
4604
ALF
4606

FIG. 11

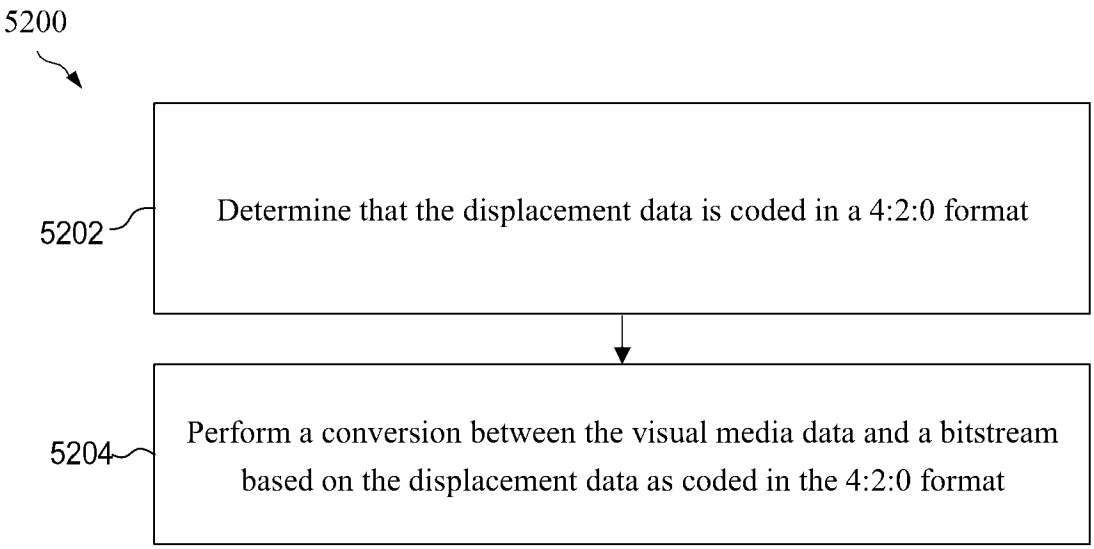

5200
5202
Determine that the displacement data is coded in a 4:2:0 format
5204
Perform a conversion between the visual media data and a bitstream based on the displacement data as coded in the 4:2:0 format

FIG. 12

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2024/071911

A. CLASSIFICATION OF SUBJECT MATTER

H04N19/136(2014.01)i; G06T17/20(2006.01)i; G06T9/00(2006.01)i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC: H04N; G06T

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

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

CNTXT,ENTXT,VEN,DWPI,CNKI,JCTVC,IEEE:bitstream, cod+, difference, displacement, mesh, motion, texture, dynamic mesh, "4:4:4", reconstruct+, transquant_bypass, "4:2:0", "3D"

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2021375005 A1 (KOREA ADVANCED INSTITUTE OF SCIENCE AND TECHNOLOGY) 02 December 2021 (2021-12-02) claims 1,8	1-66
A	WO 2020233514 A1 (BEIJING BYTEDANCE NETWORK TECHNOLOGY CO., LTD. et al.) 26 November 2020 (2020-11-26) The whole document	1-66
A	US 2015264348 A1 (QUALCOMM INCORPORATED) 17 September 2015 (2015-09-17) The whole document	1-66
A	US 2021090301 A1 (APPLE INC.) 25 March 2021 (2021-03-25) The whole document	1-66
A	US 2022165030 A1 (ADOBE INC. et al.) 26 May 2022 (2022-05-26) The whole document	1-66

☐ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"D" document cited by the applicant in the international application
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
12 March 2024	**28 March 2024**

Name and mailing address of the ISA/CN	Authorized officer
CHINA NATIONAL INTELLECTUAL PROPERTY ADMINISTRATION **6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing 100088, China**	**ZHANG,LuWen** Telephone No. **(+86) 010-53961724**

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2024/071911

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
US 2021375005 A1	02 December 2021	KR 102272569 B1	05 July 2021
WO 2020233514 A1	26 November 2020	CN 113853798 A	28 December 2021
US 2015264348 A1	17 September 2015	WO 2015142854 A1	24 September 2015
US 2021090301 A1	25 March 2021	US 2023030913 A1	02 February 2023
		US 2023401751 A1	14 December 2023
		WO 2021062044 A1	01 April 2021
US 2022165030 A1	26 May 2022	None	