



(51) International Patent Classification:

G06T 9/00 (2006.01) H04N 19/96 (2014.01)
G06T 9/40 (2006.01)

(21) International Application Number:

PCT/EP2024/085612

(22) International Filing Date:

11 December 2024 (11.12.2024)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

23307326.1 21 December 2023 (21.12.2023) EP

(71) Applicant: **INTERDIGITAL CE PATENT HOLDINGS, SAS** [FR/FR]; 3 rue du Colonel Moll, 75017 Paris (FR).

(72) Inventors: **MARVIE, Jean-Eudes**; c/o InterDigital, Immeuble ZEN 2, 845A Avenue des Champs Blancs, 35510 CESSON-SEVIGNE (FR). **MOCQUARD, Olivier**; c/o InterDigital, Immeuble ZEN 2, 845A Avenue des Champs Blancs, 35510 CESSON-SEVIGNE (FR).

(74) Agent: **INTERDIGITAL**; Immeuble Zen 2, 845A Avenue des Champs Blancs, 35510 CESSON-SEVIGNE (FR).

(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).

Published:

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

(54) Title: ADAPTIVE SUBDIVISION IN MESH CODING

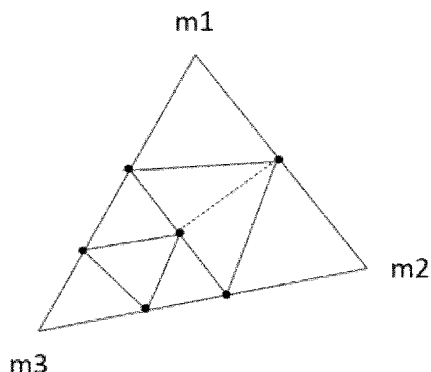


FIG. 11A

(57) Abstract: Encoding and decoding of mesh data. In an example mesh decoding method, a first mesh is obtained comprising a plurality of initial triangles. For each initial triangle, first information is obtained from a bitstream indicating whether to perform a first subdivision of the respective initial triangle, and at least one of the initial triangles is subdivided into a plurality of first-level triangles according to the first information. For each of the first-level triangles, second information is obtained from the bitstream indicating whether to perform a second subdivision of the respective first-level triangle, and at least one of the first-level triangles is subdivided into a plurality of second-level triangles according to the second information. Additional information may be provided to allow for iterative subdivision of triangles in the mesh. The information may be encoded in a tree structure, with a tree being provided for each initial triangle.



ADAPTIVE SUBDIVISION IN MESH CODING

CROSS-REFERENCE

- 5 [0001] This application claims the priority of European Patent Application No. 23307326.1, filed 21 December 2023, entitled "Adaptive Subdivision in Mesh Coding," which is incorporated herein by reference in its entirety.

BACKGROUND

- 10 [0002] The present disclosure relates to systems and methods for encoding and decoding a mesh.

- [0003] Following the MPEG V-Mesh (now renamed V-DMC) call for proposals, the solution proposed by Apple was selected to become the foundation of the MPEG V-Mesh Test Model (TM). The proposal is described in K. Mammou, J. Kim, A. Tourapis and D. Podborski, "m59281 - [V-CG] Apple's Dynamic Mesh Coding CfP Response," Apple Inc, 2022. In the
15 current MPEG V-DMC TM v5 (Specification WD 5.0) the mesh of each frame is subdivided into a low resolution base mesh, the base mesh is then tessellated uniformly, as shown in FIGs. 7A-7C and FIGs. 8A-8C. Wavelet coefficients are used to encode some displacement values which moves the vertices resulting from the tessellation to fit the original mesh.

- [0004] However, a mesh that has been uniformly tessellated is not necessarily the most
20 efficient representation of a surface. The quality of a mesh representation may be improved when some regions are represented with a greater density of triangles, and the efficiency of the mesh coding may be improved when other regions are represented with a lower density of triangles.

SUMMARY

- 25 [0005] A mesh encoding method according to some embodiments comprises: obtaining a first mesh comprising a plurality of initial triangles; for each of a plurality of the triangles in the first mesh, making a first determination of whether to subdivide the respective initial triangle into a plurality of first-level triangles; and signaling in a bitstream first information indicating an outcome of the first determination for each of the initial triangles.
- 30 [0006] In some embodiments, information is signaled that allows for iterative subdivision of the triangles. For example, in some embodiments, for each of a plurality of the first-level triangles, a second determination is made of whether to subdivide the respective first-level triangle into a plurality of second-level triangles; and information is signaled in the bitstream indicating an outcome of the second determination for each of the first-level triangles.

[0007] Some embodiments further include, for each of a plurality of the second-level triangles, making a third determination of whether to subdivide the respective second-level triangle into a plurality of third-level triangles; and signaling in the bitstream third information indicating an outcome of the third determination for each of the second-level triangles.

5 **[0008]** In some embodiments, signaling the first information comprises signaling a tree structure for each of the initial triangles, wherein a value at a root node of the tree structure indicates the outcome of the first determination. In some such embodiments, for at least one of the tree structures, the root node has a plurality of first child nodes, each first child node being associated with a respective one of the first-level triangles, a value of each of the
10 respective first child nodes indicating the outcome of the second determination. In some embodiments, for at least one of the tree structures, at least one of the first child nodes has a plurality of second child nodes, each second child node being associated with a respective one of the second-level triangles, a value of each of the respective second child nodes indicating the outcome of the third determination.

15 **[0009]** Some embodiments further include, in response to a determination that a subdivision of a first triangle creates a T-intersection at a first vertex along a first edge between the first triangle and a second triangle: subdividing the second triangle by creating a second edge between the first vertex and a second vertex, the second vertex being a vertex of the second triangle opposite the first edge.

20 **[0010]** In some embodiments, the determination of whether to subdivide a triangle is based at least in part on an error metric. For example, in some embodiments, the first determination is made based on an error metric between points on the respective initial triangle and corresponding points on an input mesh. Similarly, in some embodiments, the second determination is made based on an error metric between points on the respective first-level
25 triangle and corresponding points on the input mesh; and the third determination is made based on an error metric between points on the respective second-level triangle and corresponding points on the input mesh.

[0011] Some embodiments further include, within a first predetermined region of a video frame, encoding displacements of vertices of the first-level triangles in a first contiguous area and encoding padded values in a second contiguous area; and within a second predetermined
30 region of a video frame, encoding displacements of vertices of the second-level triangles in a third contiguous area and encoding padded values in a fourth contiguous area. In such embodiments, where a plurality of video frames are coded with such displacements, the sizes and positions of the first and second predetermined regions remain consistent (e.g.
35 unchanged) across a plurality of the video frames. Similar predetermined regions with

consistent sizes and positions may be used for the vertices of higher-level (e.g. third-level, fourth-level etc.) triangles.

5 **[0012]** Some embodiments include obtaining a source mesh model. In such embodiments, the first mesh may be obtained by a method that includes decimating at least a portion of the source mesh model.

[0013] Some embodiments include performing uniform subdivision of a mesh to a selected level, followed by adaptive subdivision of triangles to one or more further levels. Information identifying the selected level may be signaled in the bitstream.

10 **[0014]** A mesh decoding method according to some embodiments comprises: obtaining a first mesh comprising a plurality of initial triangles; for each of a plurality of the initial triangles, obtaining from a bitstream first information indicating whether to perform a first subdivision of the respective initial triangle; and subdividing at least one of the initial triangles into a plurality of first-level triangles according to the first information.

15 **[0015]** Some methods further include: for each of a plurality of the first-level triangles, obtaining from the bitstream second information indicating whether to perform a second subdivision of the respective first-level triangle; and subdividing at least one of the first-level triangles into a plurality of second-level triangles according to the second information.

20 **[0016]** Some methods further include: for each of a plurality of the second-level triangles, obtaining from the bitstream third information indicating whether to perform a third subdivision of the respective second-level triangle; and subdividing at least one of the second-level triangles into a plurality of third-level triangles according to the third information.

[0017] In some embodiments, obtaining the first information comprises obtaining a tree structure for each of the initial triangles, wherein a value at a root node of the tree structure indicates whether to perform the first subdivision of the respective initial triangle.

25 **[0018]** In some embodiments, for at least one of the tree structures, the root node has a plurality of first child nodes, each first child node being associated with a respective one of the first-level triangles, a value of each of the respective first child nodes indicating whether to perform the second subdivision of the respective first-level triangle.

30 **[0019]** In some embodiments, for at least one of the tree structures, at least one of the first child nodes has a plurality of second child nodes, each second child node being associated with a respective one of the second-level triangles, a value of each of the respective second child nodes indicating whether to perform the third subdivision of the respective second-level triangle.

[0020] Some embodiments further comprise, in response to a determination that that a subdivision of a first triangle creates a T-intersection at a first vertex along a first edge between the first triangle and a second triangle: subdividing the second triangle by creating a second edge between the first vertex and a second vertex, the second vertex being a vertex of the second triangle opposite the first edge.

[0021] A mesh encoding method according to some embodiments comprises: obtaining a first mesh comprising a plurality of initial triangles; for each initial triangle, and for each triangle that results from subdividing an initial triangle one or more times, making a respective determination of whether to perform a subdivision; subdividing one or more of the initial triangles and the triangles that result from subdividing the initial triangles according to the respective determinations; and signaling in a bitstream information indicating the respective determinations.

[0022] A mesh encoding method according to some embodiments comprises: obtaining a first mesh comprising a plurality of initial triangles; selecting a number N of levels for uniform mesh subdivision; performing N levels of uniform subdivision of the initial triangles to obtain a uniformly-subdivided mesh comprising a plurality of level-N triangles; for each of the level-N triangles, and for each triangle that results from subdividing an N-level triangle one or more times, making a respective determination of whether to perform a subdivision; subdividing one or more of the level-N triangles and the triangles that result from subdividing the level-N triangles according to the respective determinations; and signaling in a bitstream indicating the number N of levels and information indicating the respective determinations.

[0023] A mesh decoding method according to some embodiments comprises: obtaining a first mesh comprising a plurality of initial triangles; obtaining information indicating, for each of the initial triangles, and for each triangle that results from subdividing an initial triangle one or more times, whether to perform a subdivision of the respective triangle; and subdividing one or more of the initial triangles and the triangles that result from subdividing the initial triangles according to the respective determinations.

[0024] A mesh decoding method according to some embodiments comprises: obtaining a first mesh comprising a plurality of initial triangles; obtaining information indicating a number N of levels for uniform mesh subdivision; performing N levels of uniform subdivision of the initial triangles to obtain a uniformly-subdivided mesh comprising a plurality of level-N triangles; obtaining information indicating, for each of the level-N triangles, and for each triangle that results from subdividing a level-N triangle one or more times, whether to perform a subdivision of the respective triangle; and subdividing one or more of the level-N triangles and the triangles that result from subdividing the N-level triangles according to the respective determinations.

[0025] Some embodiments include at least one processor and a computer-readable medium storing instructions for performing any of the methods described herein.

[0026] Some embodiments include a computer-readable medium (which may be non-transitory) storing instructions for performing any of the methods described herein.

5 [0027] Some embodiments include a computer-readable medium storing a mesh encoded according to any of the encoding methods described herein.

[0028] Some embodiments include a signal conveying a mesh encoded according to any of the encoding methods described herein.

BRIEF DESCRIPTION OF THE DRAWINGS

10 [0029] FIG. 1 is a functional block diagram of an example mesh encoding system using intra frame encoding.

[0030] FIG. 2 is a functional block diagram of an example mesh decoding system using intra frame decoding.

15 [0031] FIG. 3 illustrates an example of an edgebreaker mesh codec according to some embodiments.

[0032] FIG. 4 is a block diagram of a block-based hybrid video encoding system.

[0033] FIG. 5 is a block diagram of a block-based video decoder.

[0034] FIG. 6 is a flow diagram of a mesh encoding and decoding process using homogeneous tessellation.

20 [0035] FIGs. 7A-C illustrate uniform tessellation of a triangle in a mesh.

[0036] FIGs. 8A-C illustrate uniform tessellation of a mesh.

[0037] FIGs. 9A-9C illustrate adaptive subdivision of a single triangle of the base mesh according to some embodiments.

25 [0038] FIGs. 10A-10C illustrate an example of tree coding (using 1 bit / node or varying bits / node) of the multi-level adaptive subdivision of one triangle.

[0039] FIGs. 11A-11C illustrate an example of tessellations to prevent T vertices on three different non-regular subdivisions, without adding non-mid-points. The dotted edges are added to remove T vertices.

30 [0040] FIG. 12 is a flow diagram illustrating a mesh encoding and decoding method according to some embodiments.

[0041] FIG. 13 is a 2D side view illustration of mean square error processing between two levels of subdivision.

[0042] FIG. 14A illustrates a packing of displacements, with black points illustrating the distribution of zeroes. FIG. 14B illustrates packing of displacements according to example
5 embodiments, with the zeroes being regrouped in the black areas.

[0043] FIG. 15 is a block diagram of an example of a system in which various aspects and embodiments are implemented.

DETAILED DESCRIPTION

10 Overview of dynamic mesh coding

[0044] FIG. 1 is a schematic block diagram of a mesh encoding process that may be employed in some embodiments. A source mesh model 302 is provided as an input mesh $M(i)$ to the mesh encoding process. The source mesh model 302 is associated with a source texture map 304 that is provided as an input texture map $A(i)$ to the encoding process. The input mesh is
15 decimated at 306 to generate a base mesh $m(i)$ with a reduced number of vertices, and a UV atlas is generated for the base mesh AT 308. The base mesh is quantized at 310 and encoded at 312, with the compressed base mesh data being multiplexed at 314 into a dynamic mesh bitstream. The compressed base mesh data is reconstructed at the encoder to generate reconstructed base mesh $m'(i)$ with a static mesh decoder 316. The reconstructed base mesh
20 is subdivided at 318 by adding new vertices. A subdivision surface fitting process is performed at 320 by comparing the subdivided base mesh with the input mesh $M(i)$ to determine a set of displacements $d(i)$ that deform the vertices of the subdivided base mesh to correspond more closely to the surfaces defined by the input mesh $M(i)$. These displacements may be updated at 322 into updated displacements $d'(i)$ based on difference between the original base mesh
25 $m(i)$ and the reconstructed base mesh $m'(i)$. These updated displacements are encoded using a wavelet transform 324 that generates wavelet coefficients $e'(i)$, which are quantized at 326 and packed at 328 into an image format. A time-varying series of images representing the wavelet coefficients may be encoded at 330 using conventional video encoding techniques, and the encoded video may be multiplexed at 314 with the data representing the compressed
30 base mesh. At the encoder, the displacements are reconstructed from the encoded video through image unpacking 329, inverse quantization 331, and inverse wavelet transform 332 to generate a reconstructed set of displacements $d''(i)$. A reconstructed base mesh $M''(i)$ is obtained through inverse quantization at 334 of the reconstructed quantized base mesh $m'(i)$, and the reconstructed base mesh $M''(i)$ is subdivided at 336. A reconstructed deformed mesh
35 $DM(i)$ is generated at 338 by applying the reconstructed set of displacements $d''(i)$ to the

reconstructed base mesh $m'(i)$. The reconstructed deformed mesh $DM(i)$ is used as a destination mesh model 340 for the purpose of attribute transfer.

[0045] Using the reconstructed deformed mesh $DM(i)$ (destination mesh model 340), the input mesh $M(i)$ (source mesh model 302), and the input texture map $A(i)$ (source texture map 304), an attribute transfer process 341 is performed to provide attribute values for a destination texture map $A'(i)$ that is associated with the reconstructed deformed mesh $DM(i)$. Pixels in the texture map $A'(i)$ that are not associated with any triangle of the reconstructed deformed mesh $DM(i)$ may be filled using a padding process 342. A color space conversion 344 may be performed, a time-varying series of texture maps $A'(i)$ may be encoded using conventional video encoding techniques 346, and the encoded video may be multiplexed at 314 into a bitstream 350 with the data representing the displacements and the compressed base mesh. Patch information 348 may also be multiplexed in the bitstream.

[0046] FIG. 2 illustrates a mesh decoding method that may be performed in some embodiments. A compressed bitstream $b(i)$ is demultiplexed into data representing patch information, data representing a static mesh, video data representing mesh displacements, and video data representing attributes. The data representing a static mesh is decoded by a static mesh decoder into a reconstructed quantized base mesh $m'(i)$ and inverse quantized, resulting in a decoded base mesh $m''(i)$. The video data representing mesh displacements is decoded. The decoded image is unpacked and inverse quantized. An inverse wavelet transform is applied, resulting in decoded displacements $d''(i)$. The deformed mesh is reconstructed using the decoded base mesh $m''(i)$ and the decoded displacements $d''(i)$, resulting in a decoded mesh $M''(i)$. The video data representing attributes is decoded, and color format/space conversion is applied, resulting in a decoded attribute map $A''(i)$.

[0047] FIGs. 1 and 2 illustrate examples of intra mesh encoding and decoding. It should be noted that example embodiments described herein may also be implemented in the case of inter mesh encoding and decoding.

Overview of an example edgebreaker codec

[0048] Some embodiments are based on edgebreaker technology. Edgebreaker is a technology that is capable of efficiently coding the connectivity of a triangular mesh. In its most straightforward implementation, a mesh encoded using edgebreaker is represented by an ordered series made up of the symbols, C, L, E, R, and S, called the "CLERS" sequence. Generally speaking, beginning with a starting triangle, these symbols describe different ways to attach a new triangle, providing information on whether or how different edges of the new triangle are connected to one or more of the existing triangles.

[0049] Edgebreaker (EB) is a technology for encoding and decoding (collectively “coding”) the topology (connectivity and handles) of the mesh. FIG. 3 illustrates an example of an edgebreaker mesh codec that may be used in some embodiments. The top row is the encoding line, and the bottom row is the decoding line. As illustrated in FIG. 3, the coding of the connectivity may include some or all of the following. During encoding, pre-processing 102 may be used to clean potential connectivity issues (non-manifolds edges and vertices) that can exist on the input mesh. This cleanup is performed because the edgebreaker algorithm by itself does not work on meshes having such connectivity issues. In some embodiments, cleaning non manifold edges and vertices involves duplicating a number of points. Some embodiments keep track of those duplicated vertices to merge those when decoding. This enables a reduction in the number of points in the decoded mesh but calls for sending some additional information in the bitstream. In some embodiments, this pre-processing 102 further includes adding some dummy points to fill potential holes on the surface because the edgebreaker algorithm alone does not handle holes. Example embodiments operate to fill the holes before encoding and recreate the holes after decoding. Example embodiments use “virtual” dummy points and generate and encode dummy triangles attached to these dummy points, but the 3D positions of those points are not encoded or decoded. In some embodiments, the vertex attributes are quantized if needed. Those attributes can be provided to the coder already quantized.

[0050] Example embodiments encode the connectivity of the mesh at 104 using a modified version of the edgebreaker algorithm from which generates a CLERS table (a table made of ‘C’, ‘L’, ‘E’, ‘R’, and ‘S’ symbols). This stage also generates some tables in memory that are used for the attribute prediction stage. The vertex attributes are then predicted at 106, starting with the position attributes. Then, other attributes are predicted, eventually relying on the position predictions, which is the case for the texture UV coordinates. Configuration and metadata are also provided in the bitstream, the CLERS table, some other connectivity and all the attribute prediction residuals are entropy coded at 110 and added to the bitstream.

[0051] In an example decoding method, all of the entropy coded sub-bitstreams are entropy decoded at 112, and the mesh connectivity is reconstructed at 114 using the CLERS table and the edgebreaker algorithm. Extra information may be used to manage the handles, which describe the topology. Example embodiments use the mesh connectivity as well as a minimal set of vertex positions expressed in 3D coordinates to predict all the other per-vertex positions at 116. The attribute residuals are then applied to correct the predictions and obtain the reconstructed vertex positions. The other attributes are also decoded at 116, potentially relying on decoded positions, as for UV coordinates. The connectivity of attributes using separate

index tables is reconstructed using per a per edge binary seam information that is entropy coded.

[0052] While the example of edgebreaker encoding and decoding is given here, it should be noted that the example embodiments are not limited to the use of an edgebreaker codec and may also be implemented using other techniques for the coding of a static mesh (including, but not limited to, a reverse-edgebreaker implementation).

[0053] In a post-processing stage, at 120, dummy triangles are removed. Optionally, the non-manifold issues are re-created in case the coder is configured to perform lossless coding, and dequantization of the vertex attributes may be performed if the model was quantized by the encoder.

Overview of block-based video coding

[0054] As noted above, the systems and methods disclosed herein may be used in the coding of textured meshes, which may be dynamic textured meshes. In some embodiments, information representing the displacements of a dynamic mesh and/or information representing attributes of the mesh (e.g. texture information) may be coded using known video coding techniques. An overview of block-based video coding techniques that may be used in some embodiments is provided below.

[0055] The video coding standards HEVC and VVC, among others, are built upon the block-based hybrid video coding framework. FIG. 4 is a block diagram of a block-based hybrid video encoding system 200. Variations of this encoder 200 are contemplated, but the encoder 200 is described below for purposes of clarity without describing all expected variations.

[0056] Before being encoded, a video sequence may go through pre-encoding processing (204), for example, applying a color transform to an input color picture (e.g., conversion from RGB 4:4:4 to YCbCr 4:2:0), or performing a remapping of the input picture components in order to get a signal distribution more resilient to compression (for instance using a histogram equalization of one of the color components). Metadata can be associated with the pre-processing and attached to the bitstream.

[0057] The input video signal 202 including a picture to be encoded is partitioned (206) and processed block by block in units of, for example, CUs. Different CUs may have different sizes. In VTM-1.0, a CU can be up to 128x128 pixels. However, different from the HEVC which partitions blocks only based on quad-trees, in the VTM-1.0, a coding tree unit (CTU) is split into CUs to adapt to varying local characteristics based on quad/binary/ternary-tree. Additionally, the concept of multiple partition unit type in the HEVC is removed, such that the separation of CU, prediction unit (PU) and transform unit (TU) does not exist in the VVC-1.0 anymore; instead, each CU is always used as the basic unit for both prediction and transform

without further partitions. In the multi-type tree structure, a CTU is firstly partitioned by a quad-tree structure. Then, each quad-tree leaf node can be further partitioned by a binary and ternary tree structure. Different splitting types may be used, such as quaternary partitioning, vertical binary partitioning, horizontal binary partitioning, vertical ternary partitioning, and horizontal ternary partitioning.

[0058] In the encoder of FIG. 4, spatial prediction (208) and/or temporal prediction (210) may be performed. Spatial prediction (or “intra prediction”) uses pixels from the samples of already coded neighboring blocks (which are called reference samples) in the same video picture/slice to predict the current video block. Spatial prediction reduces spatial redundancy inherent in the video signal. Temporal prediction (also referred to as “inter prediction” or “motion compensated prediction”) uses reconstructed pixels from the already coded video pictures to predict the current video block. Temporal prediction reduces temporal redundancy inherent in the video signal. A temporal prediction signal for a given CU may be signaled by one or more motion vectors (MVs) which indicate the amount and the direction of motion between the current CU and its temporal reference. Also, if multiple reference pictures are supported, a reference picture index may additionally be sent, which is used to identify from which reference picture in the reference picture store (212) the temporal prediction signal comes.

[0059] The mode decision block (214) in the encoder chooses the best prediction mode, for example based on a rate-distortion optimization method. This selection may be made after spatial and/or temporal prediction is performed. The intra/inter decision may be indicated by, for example, a prediction mode flag. The prediction block is subtracted from the current video block (216) to generate a prediction residual. The prediction residual is de-correlated using transform (218) and quantized (220). (For some blocks, the encoder may bypass both transform and quantization, in which case the residual may be coded directly without the application of the transform or quantization processes.) The quantized residual coefficients are inverse quantized (222) and inverse transformed (224) to form the reconstructed residual, which is then added back to the prediction block (226) to form the reconstructed signal of the CU. Further in-loop filtering, such as deblocking/SAO (Sample Adaptive Offset) filtering, may be applied (228) on the reconstructed CU to reduce encoding artifacts before it is put in the reference picture store (212) and used to code future video blocks. To form the output video bit-stream 230, coding mode (inter or intra), prediction mode information, motion information, and quantized residual coefficients are all sent to the entropy coding unit (108) to be further compressed and packed to form the bit-stream.

[0060] FIG. 5 is a block diagram of a block-based video decoder 250. In the decoder 250, a bitstream is decoded by the decoder elements as described below. Video decoder 250

generally performs a decoding pass reciprocal to the encoding pass as described in FIG. 4. The encoder 200 also generally performs video decoding as part of encoding video data.

[0061] In particular, the input of the decoder includes a video bitstream 252, which can be generated by video encoder 200. The video bit-stream 252 is first unpacked and entropy
5 decoded at entropy decoding unit 254 to obtain transform coefficients, motion vectors, and other coded information. Picture partition information indicates how the picture is partitioned. The decoder may therefore divide (256) the picture according to the decoded picture partitioning information. The coding mode and prediction information are sent to either the spatial prediction unit 258 (if intra coded) or the temporal prediction unit 260 (if inter coded) to
10 form the prediction block. The residual transform coefficients are sent to inverse quantization unit 262 and inverse transform unit 264 to reconstruct the residual block. The prediction block and the residual block are then added together at 266 to generate the reconstructed block. The reconstructed block may further go through in-loop filtering 268 before it is stored in reference picture store 270 for use in predicting future video blocks.

[0062] The decoded picture 272 may further go through post-decoding processing (274), for example, an inverse color transform (e.g. conversion from YCbCr 4:2:0 to RGB 4:4:4) or an inverse remapping performing the inverse of the remapping process performed in the pre-encoding processing (204). The post-decoding processing can use metadata derived in the pre-encoding processing and signaled in the bitstream. The decoded, processed video may
20 be sent to a display device 276. The display device 276 may be a separate device from the decoder 250, or the decoder 250 and the display device 276 may be components of the same device.

[0063] Various methods and other aspects described in this disclosure can be used to modify modules of a video encoder 200 or decoder 250. Moreover, the systems and methods
25 disclosed herein are not limited to VVC or HEVC, and can be applied, for example, to other standards and recommendations, whether pre-existing or future-developed, and extensions of any such standards and recommendations (including VVC and HEVC). Unless indicated otherwise, or technically precluded, the aspects described in this disclosure can be used individually or in combination.

30 Overview of example embodiments

[0064] FIG. 6 is a flow diagram of a known mesh encoding and decoding process using homogeneous tessellation, such as in V-DMC TM (test model) v4. FIGs. 7A-7C illustrate a homogeneous tessellation of a single triangle in a base mesh. FIG. 7A illustrates the original triangle. FIG. 7B illustrates a first subdivision of the triangle, created using the midpoints of

the sides of the original triangle. FIG. 7C illustrates a second subdivision of the triangle, created by subdividing the triangles of FIG. 7B.

[0065] One potential problem with uniform tessellation as shown in FIGs. 7A-7C is that some areas of the original mesh might not need as many generated triangles as others to fit the original model. Uniform tessellation can thus result in the generation of many triangles that do not substantially contribute to the accuracy of the mesh model. For instance, V-DMC might generate a huge number of triangles after decoding for an input mesh of around 40K triangles. For the sake of illustration, an example input mesh is illustrated schematically in FIG. 8A. As seen in FIG. 8A, some regions are represented with smaller triangles, which may be used to convey finer details, e.g. near the center of the mesh in this example, while other, flatter regions are represented with larger triangles. FIG. 8B schematically illustrates an example of a base mesh generated from the input mesh of FIG. 8A. The base mesh of FIG. 8B has fewer, larger triangles than the input mesh. FIG. 8C schematically illustrates an example of a mesh resulting from uniform tessellation of the mesh of FIG. 8B. The resulting mesh has smaller triangles, but the small triangles are distributed relatively evenly across the mesh and are not localized in the regions of fine detail found in the input mesh.

[0066] To handle non uniform (adaptive) tessellation, it is possible in some embodiments to code information for each face of the base mesh to define how many subdivision levels the face should have during the tessellation. To do so, techniques may be used such as those described in patent application no. EP 23305941.9, filed 14 June 2023, entitled "Efficient End-to-End Edge Breaker Implementation." The information may be encoded using a face Id value such as 1, 2, 3 etc. to represent the number of subdivision levels per face of the base mesh.

[0067] However, the encoding of a single per-face Id value for each triangle to provide non-uniform tessellation does not provide fine grain adaptive subdivision. For example, if one triangle of the base mesh is set to be subdivided 2 times, using a faceId=2, it will generate all the sub triangles uniformly. Hence, only the first level is properly considered adaptive.

[0068] Example embodiments allow for the possibility of subdividing one of the three triangles of level 1 down to level 2 but not the others, as shown in FIG. 9A-9C.

[0069] When performing an irregular subdivision as in FIGs. 9A-9C, some T vertices may appear. T-vertices may introduce artefacts during the rendering due to the fact that there is no more coherence on each side of the edge where the vertex is. Indeed, there is a vertex on one side that may have approximations whereas on the other side it is the edge that is interpolated. In addition, if a quantization is applied after the subdivision, the same issues may appear since the T vertex will be moved off the edge due to position alteration. Some example embodiments further prevent the creation of such T vertices.

[0070] Example embodiments provide techniques for encoding and decoding such multi-level adaptive subdivision. Further embodiments also include tools to prevent T vertex generation. Some embodiments use a partial tree to encode the structure of the multi-level adaptive subdivisions. Example embodiments further implement tools exploiting this tree to effectively
5 generate subdivided meshes without T vertices.

Coding of adaptive subdivisions

[0071] In example embodiments, the subdivision process operates, at each new subdivision of a triangle, at each depth of the subdivision, to make use of a metric that indicates whether or not the triangle is refined for at least one additional level. The metric may be used to
10 construct a tree in which each node of the tree corresponds to a triangle, and each node contains a one or a zero. In some embodiments, a zero expresses that the triangle is to be further subdivided, and a one is used otherwise to stop subdivision. FIG. 10B illustrates an example of one such tree, which may be used to encode a triangle subdivision as shown in FIG. 10A. In embodiments where a tree is provided for each of a plurality of triangles in the
15 base mesh, the collection of trees may be referred to as a forest. In some embodiments, each original triangle in a base mesh is referred to as a root. In some embodiments, the coding of the subdivision uses one bit per node. In other embodiments, the coding of the subdivision uses a varying number of bits per node. FIG. 10C illustrates an example of a tree using a
varying number of bits per node.

20 [0072] The information provided in such tree structures may be used for additional processing such as actual tessellation, displacements packing or other operations. The trees may also be encoded using AC coding and used at decoding to regenerate the same subdivision and other unpacking.

[0073] As noted above, example embodiments may use one bit per tree node (FIG. 10B) or a
25 varying number of bits per node (FIG. 10C). In the case of a varying bits, the following semantics may be used:

- (0) means stop subdivision.
- (10) means subdivide to four sub triangles and continue for each child.
- (11) means subdivide to four sub triangles and stop all.

30 [0074] Statistics may be used to find which of the three choices is the most current one and use 0 for this one. The 2 bit version is a hypothesis, which may be generated from the 1-bit representation prior to AC coding in some cases.

[0075] During an encoding process, some embodiments may use a depth-first traversal, and other embodiments may use a breadth-first traversal. Statistics on datasets may be used to choose one or the other traversal technique.

[0076] In many cases, the first level is always subdivided. To take advantage of this property, in some embodiments, a level N is signaled in a bitstream, where N indicates a level to which subdivision is performed uniformly. After the uniform subdivision to the N th level, a forest of tree structures as described herein may be used to indicate any further subdivision. The root of each such tree structure corresponds to one of the triangles resulting from the uniform subdivision to the N th level. The signaling of uniform subdivision up to a designated level may reduce the number of bits needed in the coding of the tree structures.

[0077] In addition to the use of a tree structure to guide the subdivision of triangles, some embodiments make use of information regarding which of the generated triangles are connected to other triangles to be able to generate an appropriate tessellation that does not introduce T vertices. FIGs. 11A-11C illustrate an example of tessellations to prevent T vertices on three different non-regular subdivisions, without adding non-mid-points. In these examples, the dotted edges are added to remove T vertices. For example, in response to a determination that a subdivision of a first triangle creates a T -intersection at a first vertex along a first edge between the first triangle and a second triangle, the second triangle is subdivided by creating a second edge between the first vertex and a second vertex, the second vertex being a vertex of the second triangle opposite the first edge.

[0078] In an example embodiment, during construction of the tree, additional data is stored in a separate data structure to track adjacency information of faces. This information may be re-processed on the fly at decoding and does not need to be encoded into the bitstream.

[0079] FIG. 12 is a flow diagram illustrating an overview of an encoding and decoding process using adaptive tessellation according to embodiments described herein.

[0080] In some embodiments, a determination of whether or not to subdivide a particular triangle may be based on a mean distance md between current level displaced and next level displaced. FIG. 13 is a schematic side view illustration of mean square error processing between two levels of subdivision. For example, in some embodiments, if the md is under a certain threshold, then there is no further subdivision. The value of md may be calculated by perpendicularly projecting every new point p_i of child triangles level $n + 1$ to the supporting parent triangle of level n to obtain p'_i . Then, md may be calculated as the average of the distances d_i between p_i and p'_i . Thus, md may be calculated using the following equation, where np be the number of new points, usually 3 per subdivision:

$$md = \frac{1}{np} \sum_{i=0}^{np} \| p_i p'_i \|$$

[0081] In some embodiments, more intermediate p_i^+ sampling points are included, beyond only the new points in the subdivision. These points may be generated using, for example, a regular sampling process, to improve the metric precision.

5 [0082] In some embodiments, if the three vertices from level n are not displaced in the same manner as the extreme points of the subdivided triangle, then those are also considered in the calculus of md . This depends on the displacement processing brick.

[0083] In general, a greater number of subdivisions leads to a smaller value of md_n with n being the subdivision level. In example embodiments, a threshold md_{min} is used to stop the
10 subdivision on metric. In addition, some embodiments make use of a maximum subdivision depth max_{div} in case subdivision is not enough to reach metric threshold md_{min} .

[0084] In some embodiments, a flag is provided in the bitstream to indicate whether adaptive tessellation is used.

[0085] In some embodiments, a value N is signaled in the bitstream to indicate a starting depth
15 of adaptive tessellation, such that uniform tessellation is performed for levels up to N , and adaptive tessellation is performed for level N (and possibly for subsequent levels) based on information in the tree structures.

[0086] In some embodiments, the tree structures are entropy coded, e.g. using arithmetic coding such as Dirac, RANS, or other techniques. The entropy-coded tree structures may be
20 provided in a bitstream by an encoder and read by a decoder.

Coding of displacements

[0087] In the current test model, the displacements are coded using wavelet coefficients (that are quantized using a pred lift scheme).

25 [0088] All the displacements of level 1 are packed in a rectangular region, then all those of level 2 and so on, and those are stored in a video frame. Then when some displacements coefficients are null, those are distributed all over the different rectangles and might not be contiguous, resulting in a noise-like signal that is not very efficient for the lossless (or lossy) video coder.

30 [0089] In example embodiments, the information provided in the tree structures makes it possible to determine which triangles will not necessitate the coding of displacements. The triangles that are not in the tree do not have displacements. In some embodiments, the displacements are encoded only for the triangles that require the displacements. The

rectangular regions of video frames that are used to code the displacements may become smaller if many of the triangles have no displacement, and those regions may be more spatially coherent when they have fewer cases of zero displacement. In some embodiments, padding with the same amount of removed null displacements may be used at the end of each rectangle to keep aspect ratio of the frames constant over time.

[0090] FIG. 14A schematically illustrates a video frame that encodes displacement information for triangles at different levels without regard to the positioning of zeros within the frame. Positions of zeroes within the frame are schematically illustrated with black dots.

[0091] FIG. 14B schematically illustrates a video frame that encodes displacement information for triangles at different levels according to an example embodiment. In an embodiment according to FIG. 14B, displacement information for triangles with zero displacement is grouped into a contiguous area of the frame, illustrated schematically in FIG. 14B as a black rectangle. The regions of zero displacement for triangles at different levels may be grouped into different contiguous regions. In some embodiments, the rectangle sizes for each level of displacements is left unchanged.

[0092] For the coding of displacements, vertices may be categorized in different levels. Vertices of initial triangles in a mesh before any subdivision may be referred to as initial vertices or level-zero vertices. Vertices that are created by subdividing these initial triangles into first-level triangles may be referred to as first-level vertices, vertices that are created by subdividing the first-level triangles into second-level triangles may be referred to as second-level vertices, and so on. Displacements of these vertices may be encoded in a video frame, with displacement information for vertices at different levels being encoded in different regions of the same video frame.

[0093] In some embodiments, the encoding of displacements at a particular level proceeds as follows. The per-vertex displacements of the vertices v in each level are expressed as a one-dimensional ($d=0$) or a three-dimensional ($d=0,1,2$) quantity in an array such as `dispArray[v][d]`. These displacements may be converted using a wavelet transform into wavelet transform coefficients that are quantized and packed into a portion of the video frame corresponding to the appropriate level. In cases where the portion of the video frame corresponding to a particular level is larger than needed to encode the quantized wavelet transform coefficients for that level, a region of that portion of the video frame (e.g. a rectangular region) in which no non-zero wavelet transform coefficients are encoded may be padded with zeroes or other values. Such padding may be used to allow for the different regions to have a consistent size and consistent position from frame to frame, allowing for greater compression efficiency using inter-frame prediction.

Example system hardware

[0094] Example embodiments of encoders and/or decoders (collectively coders) configured to implement embodiments described herein may be implemented using systems such as the system of FIG. 15. FIG. 15 is a block diagram of an example of a system in which various aspects and embodiments are implemented. System 1000 can be embodied as a device including the various components described below and is configured to perform one or more of the aspects described in this document. Examples of such devices, include, but are not limited to, various electronic devices such as personal computers, laptop computers, smartphones, tablet computers, digital multimedia set top boxes, digital television receivers, personal video recording systems, connected home appliances, and servers. Elements of system 1000, singly or in combination, can be embodied in a single integrated circuit (IC), multiple ICs, and/or discrete components. For example, in at least one embodiment, the processing and encoder/decoder elements of system 1000 are distributed across multiple ICs and/or discrete components. In various embodiments, the system 1000 is communicatively coupled to one or more other systems, or other electronic devices, via, for example, a communications bus or through dedicated input and/or output ports. In various embodiments, the system 1000 is configured to implement one or more of the aspects described in this document.

[0095] The system 1000 includes at least one processor 1010 configured to execute instructions loaded therein for implementing, for example, the various aspects described in this document. Processor 1010 can include embedded memory, input output interface, and various other circuitries as known in the art. The system 1000 includes at least one memory 1020 (e.g., a volatile memory device, and/or a non-volatile memory device). System 1000 includes a storage device 1040, which can include non-volatile memory and/or volatile memory, including, but not limited to, Electrically Erasable Programmable Read-Only Memory (EEPROM), Read-Only Memory (ROM), Programmable Read-Only Memory (PROM), Random Access Memory (RAM), Dynamic Random Access Memory (DRAM), Static Random Access Memory (SRAM), flash, magnetic disk drive, and/or optical disk drive. The storage device 1040 can include an internal storage device, an attached storage device (including detachable and non-detachable storage devices), and/or a network accessible storage device, as non-limiting examples.

[0096] System 1000 includes an encoder/decoder module 1030 configured, for example, to process data to provide an encoded video or decoded video, and the encoder/decoder module 1030 can include its own processor and memory. The encoder/decoder module 1030 represents module(s) that can be included in a device to perform the encoding and/or

decoding functions. As is known, a device can include one or both of the encoding and decoding modules. Additionally, encoder/decoder module 1030 can be implemented as a separate element of system 1000 or can be incorporated within processor 1010 as a combination of hardware and software as known to those skilled in the art.

5 **[0097]** Program code to be loaded onto processor 1010 or encoder/decoder 1030 to perform the various aspects described in this document can be stored in storage device 1040 and subsequently loaded onto memory 1020 for execution by processor 1010. In accordance with various embodiments, one or more of processor 1010, memory 1020, storage device 1040, and encoder/decoder module 1030 can store one or more of various items during the
10 performance of the processes described in this document. Such stored items can include, but are not limited to, the input video, the decoded video or portions of the decoded video, the bitstream, matrices, variables, and intermediate or final results from the processing of equations, formulas, operations, and operational logic.

[0098] In some embodiments, memory inside of the processor 1010 and/or the
15 encoder/decoder module 1030 is used to store instructions and to provide working memory for processing that is needed during encoding or decoding. In other embodiments, however, a memory external to the processing device (for example, the processing device can be either the processor 1010 or the encoder/decoder module 1030) is used for one or more of these functions. The external memory can be the memory 1020 and/or the storage device 1040, for
20 example, a dynamic volatile memory and/or a non-volatile flash memory. In several embodiments, an external non-volatile flash memory is used to store the operating system of, for example, a television. In at least one embodiment, a fast external dynamic volatile memory such as a RAM is used as working memory for video coding and decoding operations, such as for MPEG-2 (MPEG refers to the Moving Picture Experts Group, MPEG-2 is also referred
25 to as ISO/IEC 13818, and 13818-1 is also known as H.222, and 13818-2 is also known as H.262), HEVC (HEVC refers to High Efficiency Video Coding, also known as H.265 and MPEG-H Part 2), or VVC (Versatile Video Coding, a new standard being developed by JVET, the Joint Video Experts Team).

[0099] The input to the elements of system 1000 can be provided through various input
30 devices as indicated in block 1130. Such input devices include, but are not limited to, (i) a radio frequency (RF) portion that receives an RF signal transmitted, for example, over the air by a broadcaster, (ii) a Component (COMP) input terminal (or a set of COMP input terminals), (iii) a Universal Serial Bus (USB) input terminal, and/or (iv) a High Definition Multimedia Interface (HDMI) input terminal. Other examples, not shown in FIG. 1C, include composite
35 video.

[0100] In various embodiments, the input devices of block 1130 have associated respective input processing elements as known in the art. For example, the RF portion can be associated with elements suitable for (i) selecting a desired frequency (also referred to as selecting a signal, or band-limiting a signal to a band of frequencies), (ii) downconverting the selected
5 signal, (iii) band-limiting again to a narrower band of frequencies to select (for example) a signal frequency band which can be referred to as a channel in certain embodiments, (iv) demodulating the downconverted and band-limited signal, (v) performing error correction, and (vi) demultiplexing to select the desired stream of data packets. The RF portion of various embodiments includes one or more elements to perform these functions, for example,
10 frequency selectors, signal selectors, band-limiters, channel selectors, filters, downconverters, demodulators, error correctors, and demultiplexers. The RF portion can include a tuner that performs various of these functions, including, for example, downconverting the received signal to a lower frequency (for example, an intermediate frequency or a near-baseband frequency) or to baseband. In one set-top box embodiment, the RF portion and its associated
15 input processing element receives an RF signal transmitted over a wired (for example, cable) medium, and performs frequency selection by filtering, downconverting, and filtering again to a desired frequency band. Various embodiments rearrange the order of the above-described (and other) elements, remove some of these elements, and/or add other elements performing similar or different functions. Adding elements can include inserting elements in between
20 existing elements, such as, for example, inserting amplifiers and an analog-to-digital converter. In various embodiments, the RF portion includes an antenna.

[0101] Additionally, the USB and/or HDMI terminals can include respective interface processors for connecting system 1000 to other electronic devices across USB and/or HDMI connections. It is to be understood that various aspects of input processing, for example,
25 Reed-Solomon error correction, can be implemented, for example, within a separate input processing IC or within processor 1010 as necessary. Similarly, aspects of USB or HDMI interface processing can be implemented within separate interface ICs or within processor 1010 as necessary. The demodulated, error corrected, and demultiplexed stream is provided to various processing elements, including, for example, processor 1010, and encoder/decoder
30 1030 operating in combination with the memory and storage elements to process the datastream as necessary for presentation on an output device.

[0102] Various elements of system 1000 can be provided within an integrated housing. Within the integrated housing, the various elements can be interconnected and transmit data therebetween using suitable connection arrangement 1140, for example, an internal bus as
35 known in the art, including the Inter-IC (I2C) bus, wiring, and printed circuit boards.

[0103] The system 1000 includes communication interface 1050 that enables communication with other devices via communication channel 1060. The communication interface 1050 can include, but is not limited to, a transceiver configured to transmit and to receive data over communication channel 1060. The communication interface 1050 can include, but is not limited to, a modem or network card and the communication channel 1060 can be implemented, for example, within a wired and/or a wireless medium.

[0104] Data is streamed, or otherwise provided, to the system 1000, in various embodiments, using a wireless network such as a Wi-Fi network, for example IEEE 802.11 (IEEE refers to the Institute of Electrical and Electronics Engineers). The Wi-Fi signal of these embodiments is received over the communications channel 1060 and the communications interface 1050 which are adapted for Wi-Fi communications. The communications channel 1060 of these embodiments is typically connected to an access point or router that provides access to external networks including the Internet for allowing streaming applications and other over-the-top communications. Other embodiments provide streamed data to the system 1000 using a set-top box that delivers the data over the HDMI connection of the input block 1130. Still other embodiments provide streamed data to the system 1000 using the RF connection of the input block 1130. As indicated above, various embodiments provide data in a non-streaming manner. Additionally, various embodiments use wireless networks other than Wi-Fi, for example a cellular network or a Bluetooth network.

[0105] The system 1000 can provide an output signal to various output devices, including a display 1100, speakers 1110, and other peripheral devices 1120. The display 1100 of various embodiments includes one or more of, for example, a touchscreen display, an organic light-emitting diode (OLED) display, a curved display, and/or a foldable display. The display 1100 can be for a television, a tablet, a laptop, a cell phone (mobile phone), or other device. The display 1100 can also be integrated with other components (for example, as in a smart phone), or separate (for example, an external monitor for a laptop). The other peripheral devices 1120 include, in various examples of embodiments, one or more of a stand-alone digital video disc (or digital versatile disc) (DVR, for both terms), a disk player, a stereo system, and/or a lighting system. Various embodiments use one or more peripheral devices 1120 that provide a function based on the output of the system 1000. For example, a disk player performs the function of playing the output of the system 1000.

[0106] In various embodiments, control signals are communicated between the system 1000 and the display 1100, speakers 1110, or other peripheral devices 1120 using signaling such as AV.Link, Consumer Electronics Control (CEC), or other communications protocols that enable device-to-device control with or without user intervention. The output devices can be communicatively coupled to system 1000 via dedicated connections through respective

interfaces 1070, 1080, and 1090. Alternatively, the output devices can be connected to system 1000 using the communications channel 1060 via the communications interface 1050. The display 1100 and speakers 1110 can be integrated in a single unit with the other components of system 1000 in an electronic device such as, for example, a television. In various
5 embodiments, the display interface 1070 includes a display driver, such as, for example, a timing controller (T Con) chip.

[0107] The display 1100 and speaker 1110 can alternatively be separate from one or more of the other components, for example, if the RF portion of input 1130 is part of a separate set-top box. In various embodiments in which the display 1100 and speakers 1110 are external
10 components, the output signal can be provided via dedicated output connections, including, for example, HDMI ports, USB ports, or COMP outputs.

[0108] The embodiments can be carried out by computer software implemented by the processor 1010 or by hardware, or by a combination of hardware and software. As a non-limiting example, the embodiments can be implemented by one or more integrated circuits.
15 The memory 1020 can be of any type appropriate to the technical environment and can be implemented using any appropriate data storage technology, such as optical memory devices, magnetic memory devices, semiconductor-based memory devices, fixed memory, and removable memory, as non-limiting examples. The processor 1010 can be of any type appropriate to the technical environment, and can encompass one or more of
20 microprocessors, general purpose computers, special purpose computers, and processors based on a multi-core architecture, as non-limiting examples.

Further methods and systems

[0109] A mesh encoding method according to some embodiments comprises: obtaining a first
25 mesh comprising a plurality of initial triangles; for each of a plurality of the triangles in the first mesh, making a first determination of whether to subdivide the respective initial triangle into a plurality of first-level triangles; and signaling in a bitstream first information indicating an outcome of the first determination for each of the initial triangles.

[0110] In some embodiments, information is signaled that allows for iterative subdivision of
30 the triangles. For example, in some embodiments, for each of a plurality of the first-level triangles, a second determination is made of whether to subdivide the respective first-level triangle into a plurality of second-level triangles; and information is signaled in the bitstream indicating an outcome of the second determination for each of the first-level triangles.

[0111] Some embodiments further include, for each of a plurality of the second-level triangles,
35 making a third determination of whether to subdivide the respective second-level triangle into

a plurality of third-level triangles; and signaling in the bitstream third information indicating an outcome of the third determination for each of the second-level triangles.

[0112] In some embodiments, signaling the first information comprises signaling a tree structure for each of the initial triangles, wherein a value at a root node of the tree structure indicates the outcome of the first determination. In some such embodiments, for at least one of the tree structures, the root node has a plurality of first child nodes, each first child node being associated with a respective one of the first-level triangles, a value of each of the respective first child nodes indicating the outcome of the second determination. In some embodiments, for at least one of the tree structures, at least one of the first child nodes has a plurality of second child nodes, each second child node being associated with a respective one of the second-level triangles, a value of each of the respective second child nodes indicating the outcome of the third determination.

[0113] Some embodiments further include, in response to a determination that a subdivision of a first triangle creates a T-intersection at a first vertex along a first edge between the first triangle and a second triangle: subdividing the second triangle by creating a second edge between the first vertex and a second vertex, the second vertex being a vertex of the second triangle opposite the first edge.

[0114] In some embodiments, the determination of whether to subdivide a triangle is based at least in part on an error metric. For example, in some embodiments, the first determination is made based on an error metric between points on the respective initial triangle and corresponding points on an input mesh. Similarly, in some embodiments, the second determination is made based on an error metric between points on the respective first-level triangle and corresponding points on the input mesh; and the third determination is made based on an error metric between points on the respective second-level triangle and corresponding points on the input mesh.

[0115] Some embodiments further include, within a first predetermined region of a video frame, encoding displacements of vertices of the first-level triangles in a first contiguous area and encoding padded values in a second contiguous area; and within a second predetermined region of a video frame, encoding displacements of vertices of the second-level triangles in a third contiguous area and encoding padded values in a fourth contiguous area. In such embodiments, where a plurality of video frames are coded with such displacements, the sizes and positions of the first and second predetermined regions remain consistent (e.g. unchanged) across a plurality of the video frames. Similar predetermined regions with consistent sizes and positions may be used for the vertices of higher-level (e.g. third-level, fourth-level etc.) triangles.

[0116] Some embodiments include obtaining a source mesh model. In such embodiments, the first mesh may be obtained by a method that includes decimating at least a portion of the source mesh model.

5 [0117] Some embodiments include performing uniform subdivision of a mesh to a selected level, followed by adaptive subdivision of triangles to one or more further levels. Information identifying the selected level may be signaled in the bitstream.

[0118] A mesh decoding method according to some embodiments comprises: obtaining a first mesh comprising a plurality of initial triangles; for each of a plurality of the initial triangles, obtaining from a bitstream first information indicating whether to perform a first subdivision of
10 the respective initial triangle; and subdividing at least one of the initial triangles into a plurality of first-level triangles according to the first information.

[0119] Some methods further include: for each of a plurality of the first-level triangles, obtaining from the bitstream second information indicating whether to perform a second subdivision of the respective first-level triangle; and subdividing at least one of the first-level
15 triangles into a plurality of second-level triangles according to the second information.

[0120] Some methods further include: for each of a plurality of the second-level triangles, obtaining from the bitstream third information indicating whether to perform a third subdivision of the respective second-level triangle; and subdividing at least one of the second-level triangles into a plurality of third-level triangles according to the third information.

20 [0121] In some embodiments, obtaining the first information comprises obtaining a tree structure for each of the initial triangles, wherein a value at a root node of the tree structure indicates whether to perform the first subdivision of the respective initial triangle.

[0122] In some embodiments, for at least one of the tree structures, the root node has a plurality of first child nodes, each first child node being associated with a respective one of the
25 first-level triangles, a value of each of the respective first child nodes indicating whether to perform the second subdivision of the respective first-level triangle.

[0123] In some embodiments, for at least one of the tree structures, at least one of the first child nodes has a plurality of second child nodes, each second child node being associated with a respective one of the second-level triangles, a value of each of the respective second
30 child nodes indicating whether to perform the third subdivision of the respective second-level triangle.

[0124] Some embodiments further comprise, in response to a determination that that a subdivision of a first triangle creates a T-intersection at a first vertex along a first edge between the first triangle and a second triangle: subdividing the second triangle by creating a second

edge between the first vertex and a second vertex, the second vertex being a vertex of the second triangle opposite the first edge.

[0125] A mesh encoding method according to some embodiments comprises: obtaining a first mesh comprising a plurality of initial triangles; for each initial triangle, and for each triangle
5 that results from subdividing an initial triangle one or more times, making a respective determination of whether to perform a subdivision; subdividing one or more of the initial triangles and the triangles that result from subdividing the initial triangles according to the respective determinations; and signaling in a bitstream information indicating the respective determinations.

[0126] A mesh encoding method according to some embodiments comprises: obtaining a first mesh comprising a plurality of initial triangles; selecting a number N of levels for uniform mesh
10 subdivision; performing N levels of uniform subdivision of the initial triangles to obtain a uniformly-subdivided mesh comprising a plurality of level-N triangles; for each of the level-N triangles, and for each triangle that results from subdividing an N-level triangle one or more
15 times, making a respective determination of whether to perform a subdivision; subdividing one or more of the level-N triangles and the triangles that result from subdividing the level-N triangles according to the respective determinations; and signaling in a bitstream indicating the number N of levels and information indicating the respective determinations.

[0127] A mesh decoding method according to some embodiments comprises: obtaining a first
20 mesh comprising a plurality of initial triangles; obtaining information indicating, for each of the initial triangles, and for each triangle that results from subdividing an initial triangle one or more times, whether to perform a subdivision of the respective triangle; and subdividing one or more of the initial triangles and the triangles that result from subdividing the initial triangles according to the respective determinations.

[0128] A mesh decoding method according to some embodiments comprises: obtaining a first
25 mesh comprising a plurality of initial triangles; obtaining information indicating a number N of levels for uniform mesh subdivision; performing N levels of uniform subdivision of the initial triangles to obtain a uniformly-subdivided mesh comprising a plurality of level-N triangles; obtaining information indicating, for each of the level-N triangles, and for each triangle that
30 results from subdividing a level-N triangle one or more times, whether to perform a subdivision of the respective triangle; and subdividing one or more of the level-N triangles and the triangles that result from subdividing the N-level triangles according to the respective determinations.

[0129] Some embodiments include at least one processor and a computer-readable medium storing instructions for performing any of the methods described herein.

[0130] Some embodiments include a computer-readable medium (which may be non-transitory) storing instructions for performing any of the methods described herein.

[0131] Some embodiments include a computer-readable medium storing a mesh encoded according to any of the encoding methods described herein.

5 [0132] Some embodiments include a signal conveying a mesh encoded according to any of the encoding methods described herein.

[0133] An apparatus according to some embodiments comprises one or more processors configured to perform any of the methods disclosed herein.

10 [0134] A computer program product according to some embodiments includes instructions which, when the program is executed by one or more processors, cause the one or more processors to carry out any of the methods described herein.

[0135] This disclosure describes a variety of aspects, including tools, features, embodiments, models, approaches, etc. Many of these aspects are described with specificity and, at least to show the individual characteristics, are often described in a manner that may sound limiting.
15 However, this is for purposes of clarity in description, and does not limit the disclosure or scope of those aspects. Indeed, all of the different aspects can be combined and interchanged to provide further aspects. Moreover, the aspects can be combined and interchanged with aspects described in earlier filings as well.

[0136] The aspects described and contemplated in this disclosure can be implemented in
20 many different forms. While some embodiments are illustrated specifically, other embodiments are contemplated, and the discussion of particular embodiments does not limit the breadth of the implementations. At least one of the aspects generally relates to mesh encoding and decoding, and at least one other aspect generally relates to transmitting a bitstream generated or encoded. These and other aspects can be implemented as a method, an apparatus, a
25 computer readable storage medium having stored thereon instructions for encoding or decoding mesh data according to any of the methods described, and/or a computer readable storage medium having stored thereon a bitstream generated according to any of the methods described.

[0137] In the present disclosure, the terms “reconstructed” and “decoded” may be used
30 interchangeably, the terms “pixel” and “sample” may be used interchangeably, the terms “image,” “picture” and “frame” may be used interchangeably. Usually, but not necessarily, the term “reconstructed” is used at the encoder side while “decoded” is used at the decoder side.

[0138] The terms HDR (high dynamic range) and SDR (standard dynamic range) often convey specific values of dynamic range to those of ordinary skill in the art. However, additional

embodiments are also intended in which a reference to HDR is understood to mean “higher dynamic range” and a reference to SDR is understood to mean “lower dynamic range.” Such additional embodiments are not constrained by any specific values of dynamic range that might often be associated with the terms “high dynamic range” and “standard dynamic range.”

5 [0139] Various methods are described herein, and each of the methods comprises one or more steps or actions for achieving the described method. Unless a specific order of steps or actions is required for proper operation of the method, the order and/or use of specific steps and/or actions may be modified or combined. Additionally, terms such as “first”, “second”, etc., may be used in various embodiments to modify an element, component, step, operation, etc.,
10 such as, for example, a “first decoding” and a “second decoding”. Use of such terms does not imply an ordering to the modified operations unless specifically required. So, in this example, the first decoding need not be performed before the second decoding, and may occur, for example, before, during, or in an overlapping time period with the second decoding.

[0140] Various numeric values may be used in the present disclosure, for example. The
15 specific values are for example purposes and the aspects described are not limited to these specific values.

[0141] Embodiments described herein may be carried out by computer software implemented by a processor or other hardware, or by a combination of hardware and software. As a non-limiting example, the embodiments can be implemented by one or more integrated circuits.
20 The processor can be of any type appropriate to the technical environment and can encompass one or more of microprocessors, general purpose computers, special purpose computers, and processors based on a multi-core architecture, as non-limiting examples.

[0142] Various implementations involve decoding. “Decoding”, as used in this disclosure, can encompass all or part of the processes performed, for example, on a received encoded
25 sequence in order to produce a final output suitable for display. In various embodiments, such processes include one or more of the processes typically performed by a decoder, for example, entropy decoding, inverse quantization, inverse transformation, and differential decoding. In various embodiments, such processes also, or alternatively, include processes performed by a decoder of various implementations described in this disclosure, for example,
30 extracting a picture from a tiled (packed) picture, determining an upsampling filter to use and then upsampling a picture, and flipping a picture back to its intended orientation.

[0143] As further examples, in one embodiment “decoding” refers only to entropy decoding, in another embodiment “decoding” refers only to differential decoding, and in another embodiment “decoding” refers to a combination of entropy decoding and differential decoding.
35 Whether the phrase “decoding process” is intended to refer specifically to a subset of

operations or generally to the broader decoding process will be clear based on the context of the specific descriptions.

[0144] Various implementations involve encoding. In an analogous way to the above discussion about “decoding”, “encoding” as used in this disclosure can encompass all or part of the processes performed, for example, on an input mesh sequence in order to produce an encoded bitstream. In various embodiments, such processes include one or more of the processes typically performed by an encoder, for example, partitioning, differential encoding, transformation, quantization, and entropy encoding. In various embodiments, such processes also, or alternatively, include processes performed by an encoder of various implementations described in this disclosure.

[0145] As further examples, in one embodiment “encoding” refers only to entropy encoding, in another embodiment “encoding” refers only to differential encoding, and in another embodiment “encoding” refers to a combination of differential encoding and entropy encoding. Whether the phrase “encoding process” is intended to refer specifically to a subset of operations or generally to the broader encoding process will be clear based on the context of the specific descriptions.

[0146] When a figure is presented as a flow diagram, it should be understood that it also provides a block diagram of a corresponding apparatus. Similarly, when a figure is presented as a block diagram, it should be understood that it also provides a flow diagram of a corresponding method/process.

[0147] Various embodiments refer to rate distortion optimization. In particular, during the encoding process, the balance or trade-off between the rate and distortion is usually considered, often given the constraints of computational complexity. The rate distortion optimization is usually formulated as minimizing a rate distortion function, which is a weighted sum of the rate and of the distortion. There are different approaches to solve the rate distortion optimization problem. For example, the approaches may be based on an extensive testing of all encoding options, including all considered modes or coding parameters values, with a complete evaluation of their coding cost and related distortion of the reconstructed signal after coding and decoding. Faster approaches may also be used, to save encoding complexity, in particular with computation of an approximated distortion based on the prediction or the prediction residual signal, not the reconstructed one. A mix of these two approaches can also be used, such as by using an approximated distortion for only some of the possible encoding options, and a complete distortion for other encoding options. Other approaches only evaluate a subset of the possible encoding options. More generally, many approaches employ any of

a variety of techniques to perform the optimization, but the optimization is not necessarily a complete evaluation of both the coding cost and related distortion.

[0148] The implementations and aspects described herein can be implemented in, for example, a method or a process, an apparatus, a software program, a data stream, or a signal.

5 Even if only discussed in the context of a single form of implementation (for example, discussed only as a method), the implementation of features discussed can also be implemented in other forms (for example, an apparatus or program). An apparatus can be implemented in, for example, appropriate hardware, software, and firmware. The methods can be implemented in, for example, a processor, which refers to processing devices in general,
10 including, for example, a computer, a microprocessor, an integrated circuit, or a programmable logic device. Processors also include communication devices, such as, for example, computers, cell phones, portable/personal digital assistants (“PDAs”), and other devices that facilitate communication of information between end-users.

[0149] Reference to “one embodiment” or “an embodiment” or “one implementation” or “an
15 implementation”, as well as other variations thereof, means that a particular feature, structure, characteristic, and so forth described in connection with the embodiment is included in at least one embodiment. Thus, the appearances of the phrase “in one embodiment” or “in an embodiment” or “in one implementation” or “in an implementation”, as well as any other variations, appearing in various places throughout this disclosure are not necessarily all
20 referring to the same embodiment.

[0150] Additionally, this disclosure may refer to “determining” various pieces of information. Determining the information can include one or more of, for example, estimating the information, calculating the information, predicting the information, or retrieving the information from memory.

25 **[0151]** Further, this disclosure may refer to “accessing” various pieces of information. Accessing the information can include one or more of, for example, receiving the information, retrieving the information (for example, from memory), storing the information, moving the information, copying the information, calculating the information, determining the information, predicting the information, or estimating the information.

30 **[0152]** Additionally, this disclosure may refer to “receiving” various pieces of information. Receiving is, as with “accessing”, intended to be a broad term. Receiving the information can include one or more of, for example, accessing the information, or retrieving the information (for example, from memory). Further, “receiving” is typically involved, in one way or another, during operations such as, for example, storing the information, processing the information,
35 transmitting the information, moving the information, copying the information, erasing the

information, calculating the information, determining the information, predicting the information, or estimating the information.

[0153] It is to be appreciated that the use of any of the following “/”, “and/or”, and “at least one of”, for example, in the cases of “A/B”, “A and/or B” and “at least one of A and B”, is intended to encompass the selection of the first listed option (A) only, or the selection of the second listed option (B) only, or the selection of both options (A and B). As a further example, in the cases of “A, B, and/or C” and “at least one of A, B, and C”, such phrasing is intended to encompass the selection of the first listed option (A) only, or the selection of the second listed option (B) only, or the selection of the third listed option (C) only, or the selection of the first and the second listed options (A and B) only, or the selection of the first and third listed options (A and C) only, or the selection of the second and third listed options (B and C) only, or the selection of all three options (A and B and C). This may be extended for as many items as are listed.

[0154] Also, as used herein, the word “signal” refers to, among other things, indicating something to a corresponding decoder. For example, in certain embodiments the encoder signals a particular one of a plurality of parameters for region-based filter parameter selection for de-artifact filtering. In this way, in an embodiment the same parameter is used at both the encoder side and the decoder side. Thus, for example, an encoder can transmit (explicit signaling) a particular parameter to the decoder so that the decoder can use the same particular parameter. Conversely, if the decoder already has the particular parameter as well as others, then signaling can be used without transmitting (implicit signaling) to simply allow the decoder to know and select the particular parameter. By avoiding transmission of any actual functions, a bit savings is realized in various embodiments. It is to be appreciated that signaling can be accomplished in a variety of ways. For example, one or more syntax elements, flags, and so forth are used to signal information to a corresponding decoder in various embodiments. While the preceding relates to the verb form of the word “signal”, the word “signal” can also be used herein as a noun.

[0155] Implementations can produce a variety of signals formatted to carry information that can be, for example, stored or transmitted. The information can include, for example, instructions for performing a method, or data produced by one of the described implementations. For example, a signal can be formatted to carry the bitstream of a described embodiment. Such a signal can be formatted, for example, as an electromagnetic wave (for example, using a radio frequency portion of spectrum) or as a baseband signal. The formatting can include, for example, encoding a data stream and modulating a carrier with the encoded data stream. The information that the signal carries can be, for example, analog or digital

information. The signal can be transmitted over a variety of different wired or wireless links, as is known. The signal can be stored on a processor-readable medium.

[0156] We describe a number of embodiments. Features of these embodiments can be provided alone or in any combination, across various claim categories and types.

- 5 **[0157]** Although features and elements are described above in particular combinations, each feature or element can be used alone or in any combination with the other features and elements. In addition, the methods described herein may be implemented in a computer program, software, or firmware incorporated in a computer-readable medium for execution by a computer or processor. Examples of computer-readable storage media include, but are not
- 10 limited to, a read only memory (ROM), a random access memory (RAM), a register, cache memory, semiconductor memory devices, magnetic media such as internal hard disks and removable disks, magneto-optical media, and optical media such as CD-ROM disks, and digital versatile disks (DVDs). A processor in association with software may be used to implement a radio frequency transceiver for use in a WTRU, UE, terminal, base station, RNC,
- 15 or any host computer.

CLAIMS

1. A mesh encoding method comprising:

- 5 obtaining a first mesh comprising a plurality of initial triangles;
 for each of a plurality of the initial triangles in the first mesh, making a first determination
of whether to subdivide the respective initial triangle into a plurality of first-level triangles; and
 signaling in a bitstream first information indicating an outcome of the first determination
for each of the initial triangles.

10

2. A mesh encoding apparatus comprising one or more processors configured to perform at least:

- obtaining a first mesh comprising a plurality of initial triangles;
 for each of a plurality of the initial triangles in the first mesh, making a first determination
15 of whether to subdivide the respective initial triangle into a plurality of first-level triangles; and
 signaling in a bitstream first information indicating an outcome of the first determination
for each of the initial triangles.

3. The method of claim 1, or the apparatus of claim 2 further comprising:

- 20 for each of a plurality of the first-level triangles, making a second determination of
whether to subdivide the respective first-level triangle into a plurality of second-level triangles;
and
 signaling in the bitstream second information indicating an outcome of the second
determination for each of the first-level triangles.

25

4. The method of claim 3 as it depends from claim 1, or the apparatus of claim 3 as it depends from claim 2, further comprising:

- for each of a plurality of the second-level triangles, making a third determination of
whether to subdivide the respective second-level triangle into a plurality of third-level triangles;
30 and
 signaling in the bitstream third information indicating an outcome of the third
determination for each of the second-level triangles.

5. The method of claim 1 or claims 3-4 as they depend from claim 1, or the apparatus of claim
35 2 or claims 3-4 as they depend from claim 2, wherein signaling the first information comprises
signaling a tree structure for each of the initial triangles, wherein a value at a root node of the
tree structure indicates the outcome of the first determination.

6. The method of claim 3 as it depends from claim 1, or the apparatus of claim 3 as it depends from claim 2, wherein:

signaling the first information comprises signaling a tree structure for each of the initial triangles, wherein a value at a root node of the tree structure indicates the outcome of the first determination; and

for at least one of the tree structures, the root node has a plurality of first child nodes, each first child node being associated with a respective one of the first-level triangles, a value of each of the respective first child nodes indicating the outcome of the second determination.

7. The method of claim 1 or claims 3-6 as they depend from claim 1, or the apparatus of claim 2 or claims 3-6 as they depend from claim 2, further comprising, in response to a determination that a subdivision of a first triangle creates a T-intersection at a first vertex along a first edge between the first triangle and a second triangle:

subdividing the second triangle by creating a second edge between the first vertex and a second vertex, the second vertex being a vertex of the second triangle opposite the first edge.

8. A mesh decoding method comprising:

obtaining a first mesh comprising a plurality of initial triangles;
for each of a plurality of the initial triangles, obtaining from a bitstream first information indicating whether to perform a first subdivision of the respective initial triangle; and
subdividing at least one of the initial triangles into a plurality of first-level triangles according to the first information.

9. A mesh decoding apparatus comprising one or more processors configured to perform at least:

obtaining a first mesh comprising a plurality of initial triangles;
for each of a plurality of the initial triangles, obtaining from a bitstream first information indicating whether to perform a first subdivision of the respective initial triangle; and
subdividing at least one of the initial triangles into a plurality of first-level triangles according to the first information.

10. The method of claim 8, or the apparatus of claim 9, further comprising:

for each of a plurality of the first-level triangles, obtaining from the bitstream second information indicating whether to perform a second subdivision of the respective first-level triangle; and

subdividing at least one of the first-level triangles into a plurality of second-level triangles according to the second information.

11. The method of claim 10 as it depends from claim 8, or the apparatus of claim 10 as it
5 depends from claim 9, further comprising:

for each of a plurality of the second-level triangles, obtaining from the bitstream third information indicating whether to perform a third subdivision of the respective second-level triangle; and

10 subdividing at least one of the second-level triangles into a plurality of third-level triangles according to the third information.

12. The method of claim 8 or claims 10-11 as they depend from claim 8, or the apparatus of claim 9 or claims 10-11 as they depend from claim 9, wherein obtaining the first information comprises obtaining a tree structure for each of the initial triangles, wherein a value at a root
15 node of the tree structure indicates whether to perform the first subdivision of the respective initial triangle.

13. The method of claim 10 as it depends from claim 8, or the apparatus of claim 10 as it depends from claim 9, wherein:

20 obtaining the first information comprises obtaining a tree structure for each of the initial triangles, wherein a value at a root node of the tree structure indicates whether to perform the first subdivision of the respective initial triangle; and

for at least one of the tree structures, the root node has a plurality of first child nodes, each first child node being associated with a respective one of the first-level triangles, a value
25 of each of the respective first child nodes indicating whether to perform the second subdivision of the respective first-level triangle.

14. The method of claim 11 as it depends from claim 8, or the apparatus of claim 11 as it depends from claim 9, wherein:

30 obtaining the first information comprises obtaining a tree structure for each of the initial triangles, wherein a value at a root node of the tree structure indicates whether to perform the first subdivision of the respective initial triangle;

for at least one of the tree structures, the root node has a plurality of first child nodes, each first child node being associated with a respective one of the first-level triangles, a value
35 of each of the respective first child nodes indicating whether to perform the second subdivision of the respective first-level triangle; and

for at least one of the tree structures, at least one of the first child nodes has a plurality of second child nodes, each second child node being associated with a respective one of the second-level triangles, a value of each of the respective second child nodes indicating whether to perform the third subdivision of the respective second-level triangle.

5

15. The method of claim 8 or claims 10-14 as they depend from claim 8, or the apparatus of claim 9 or claims 10-14 as they depend from claim 9, further comprising, in response to a determination that that a subdivision of a first triangle creates a T-intersection at a first vertex along a first edge between the first triangle and a second triangle:

10

subdividing the second triangle by creating a second edge between the first vertex and a second vertex, the second vertex being a vertex of the second triangle opposite the first edge.

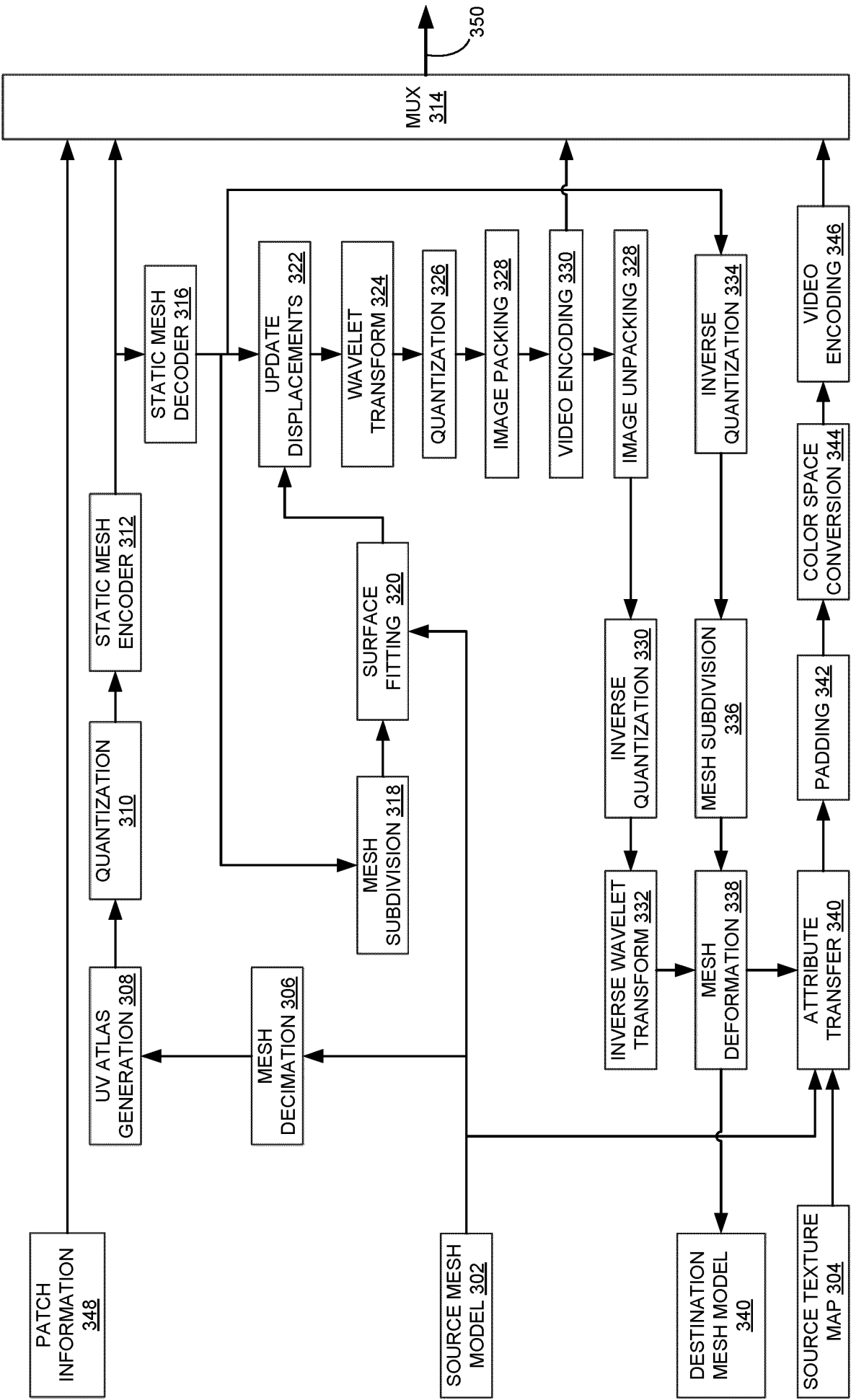


FIG. 1

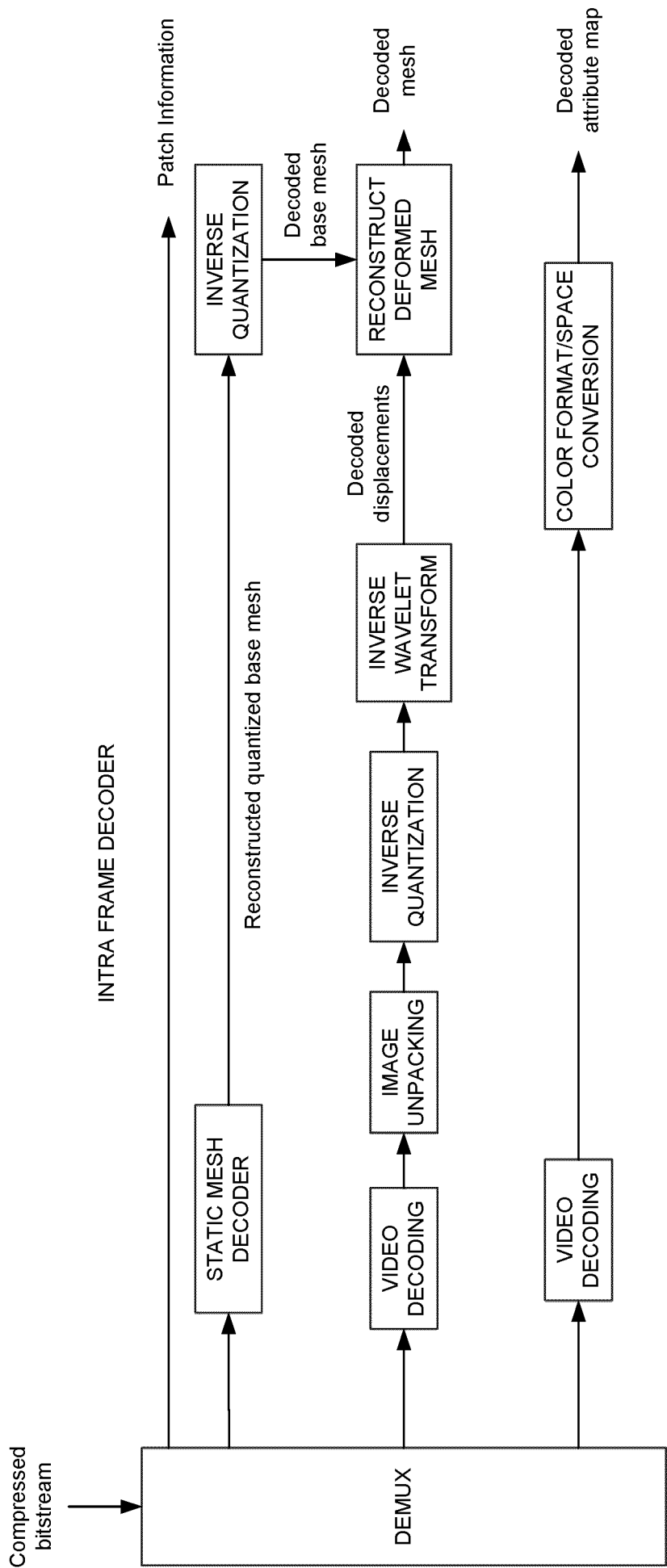


FIG. 2

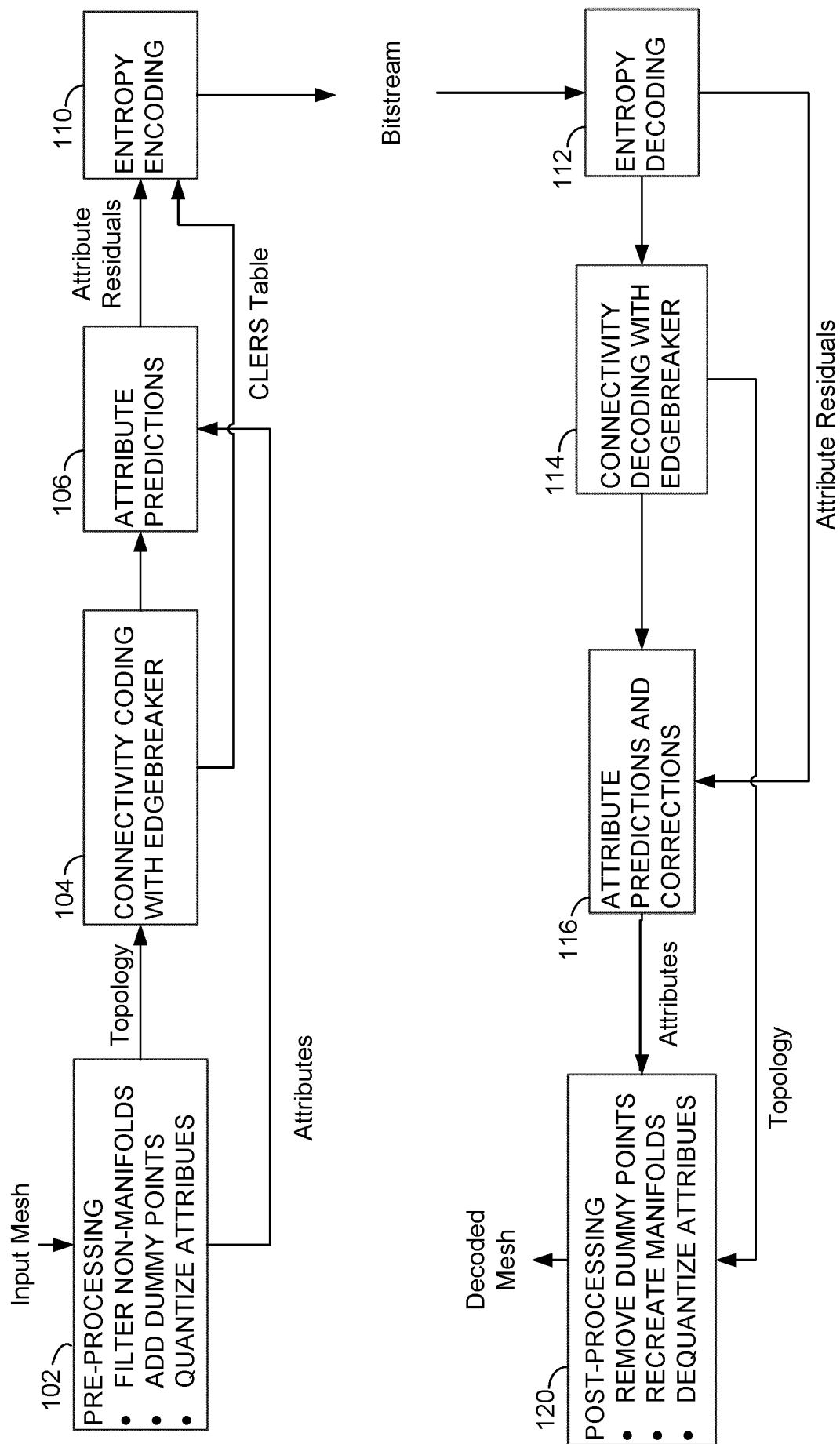


FIG. 3

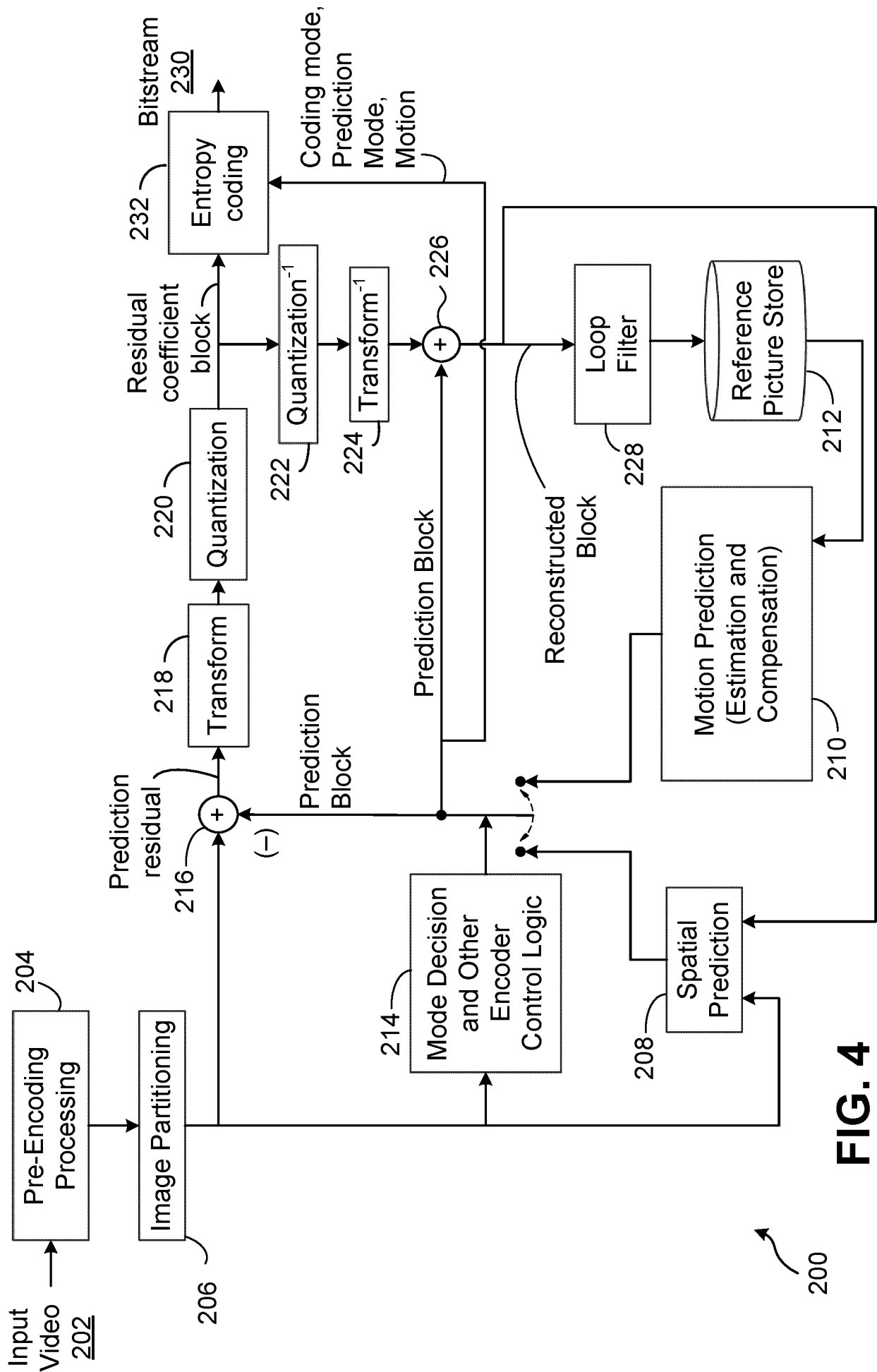


FIG. 4

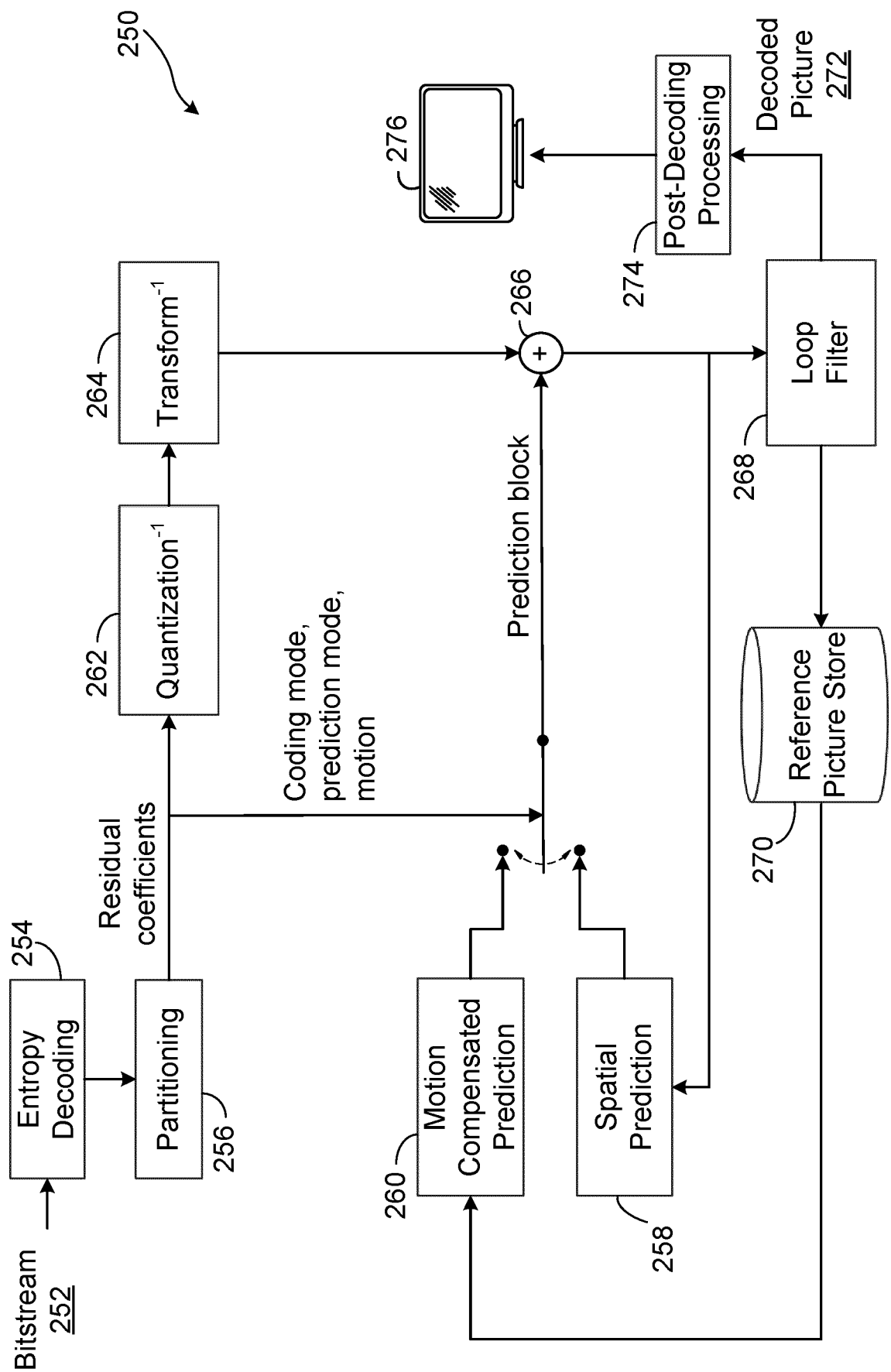


FIG. 5

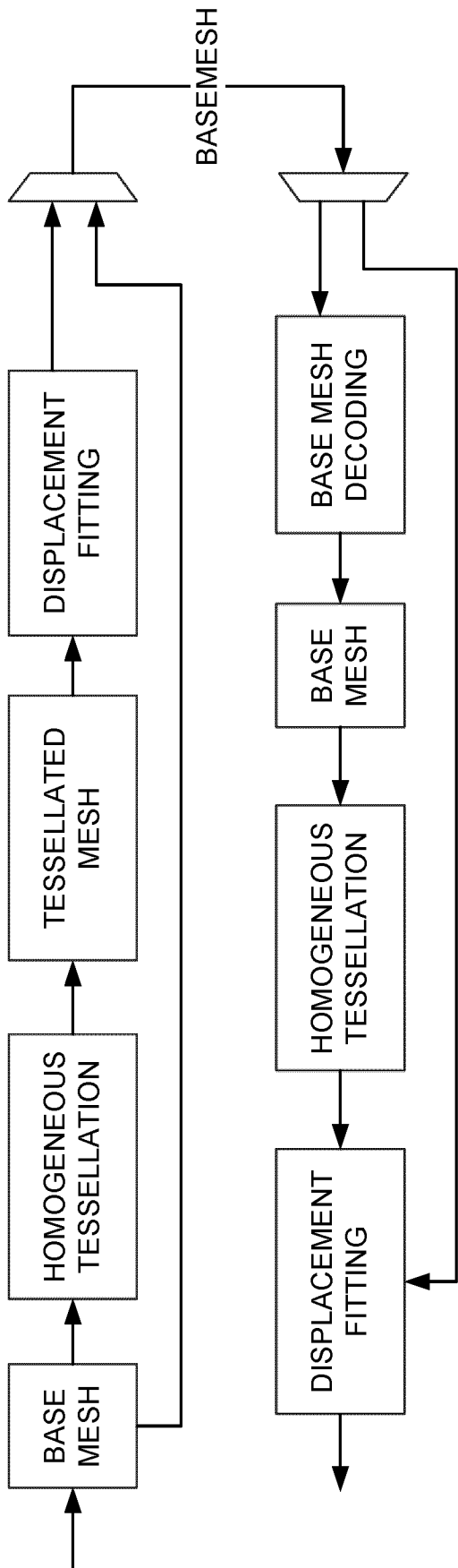


FIG. 6

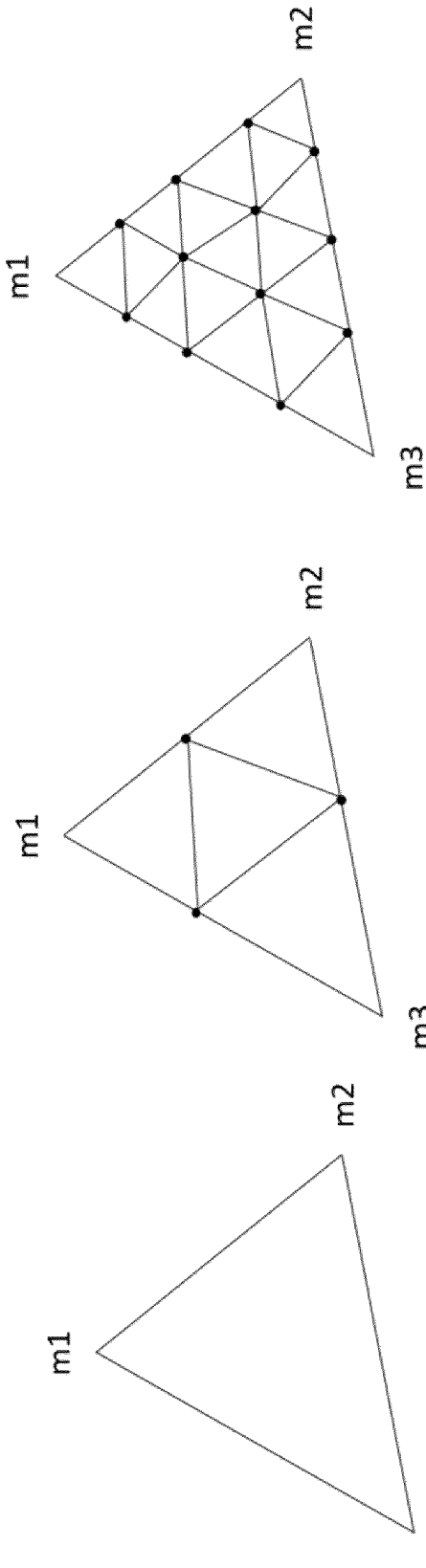


FIG. 7A

FIG. 7B

FIG. 7C

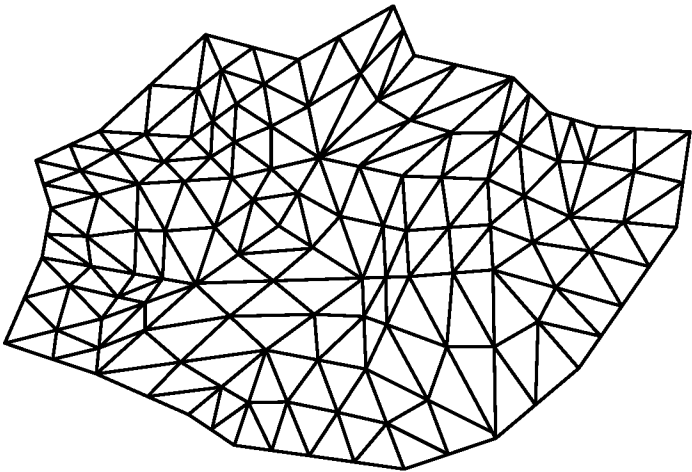


FIG. 8C

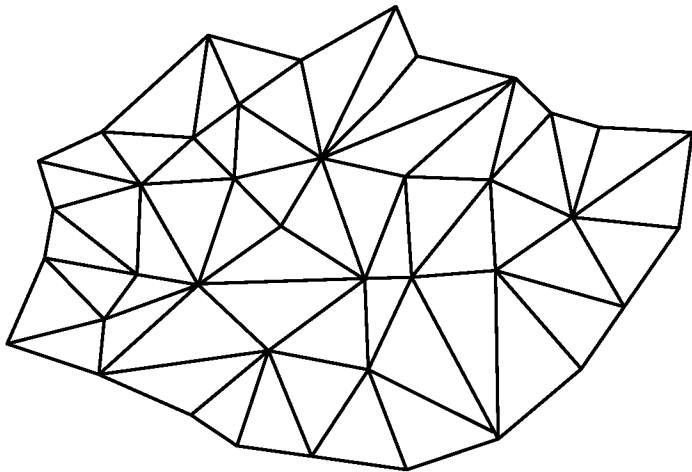


FIG. 8B

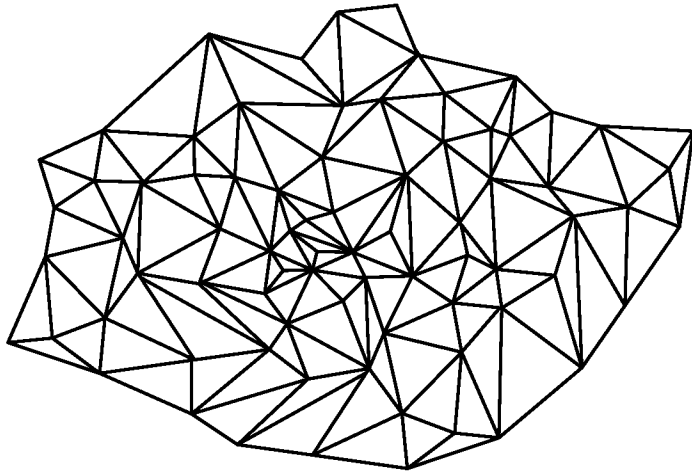


FIG. 8A

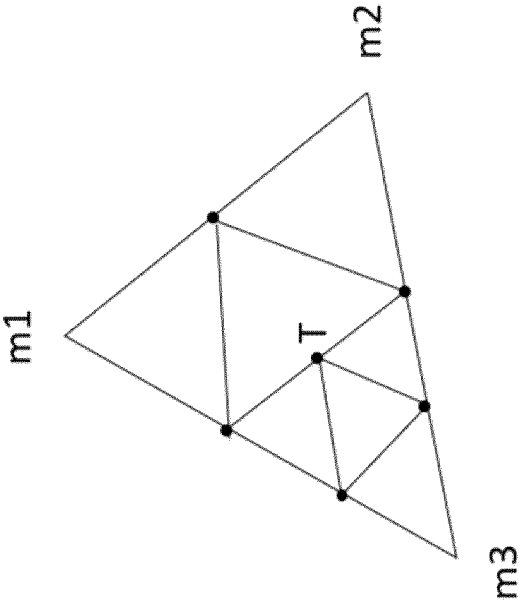


FIG. 9A

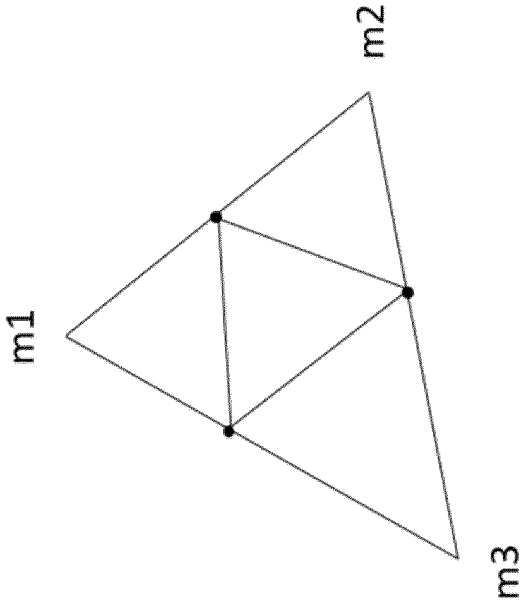


FIG. 9B

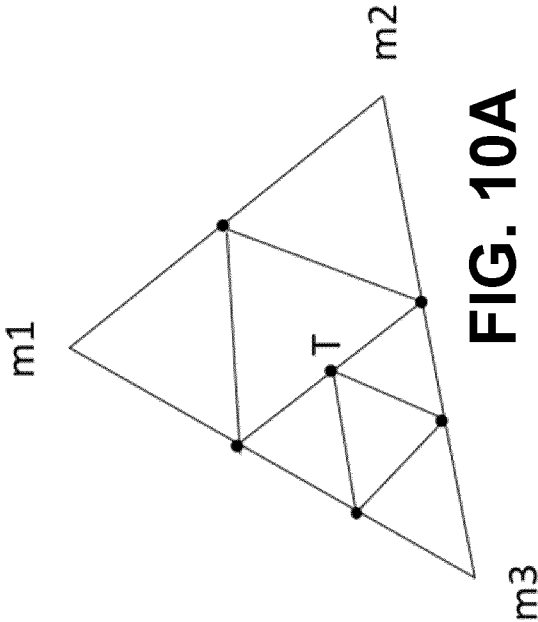


FIG. 10A

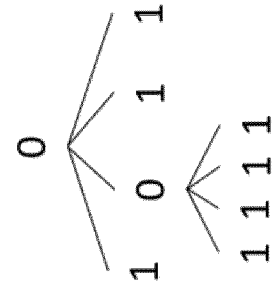


FIG. 10B

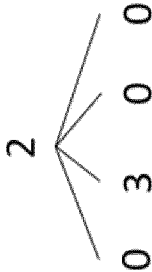


FIG. 10C

FIG. 9C

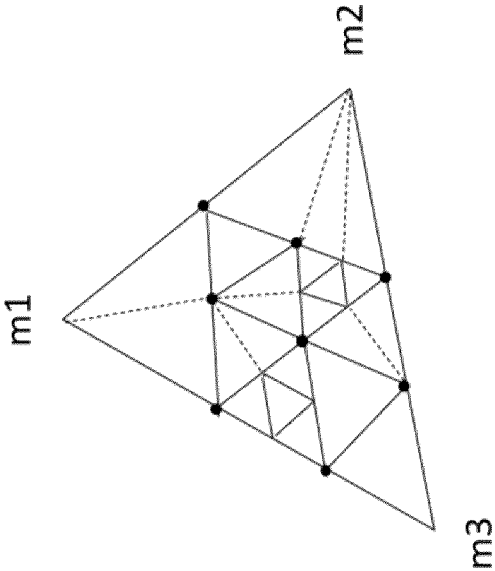


FIG. 11A

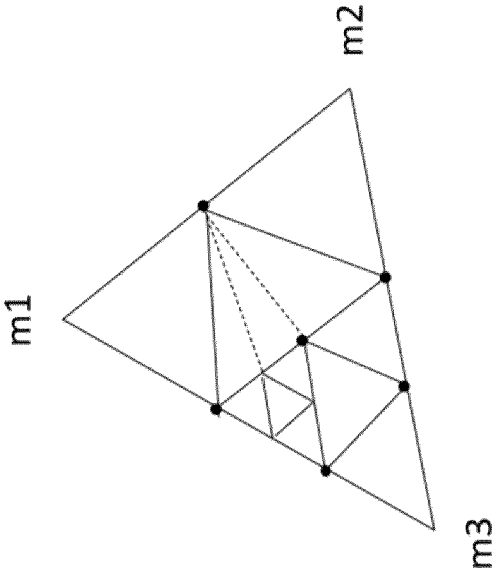


FIG. 11B

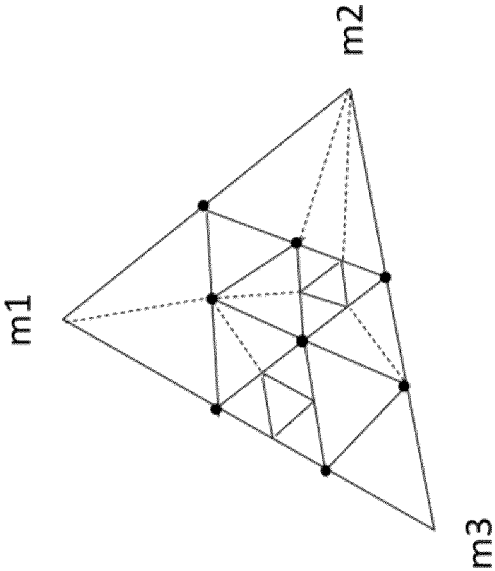


FIG. 11C

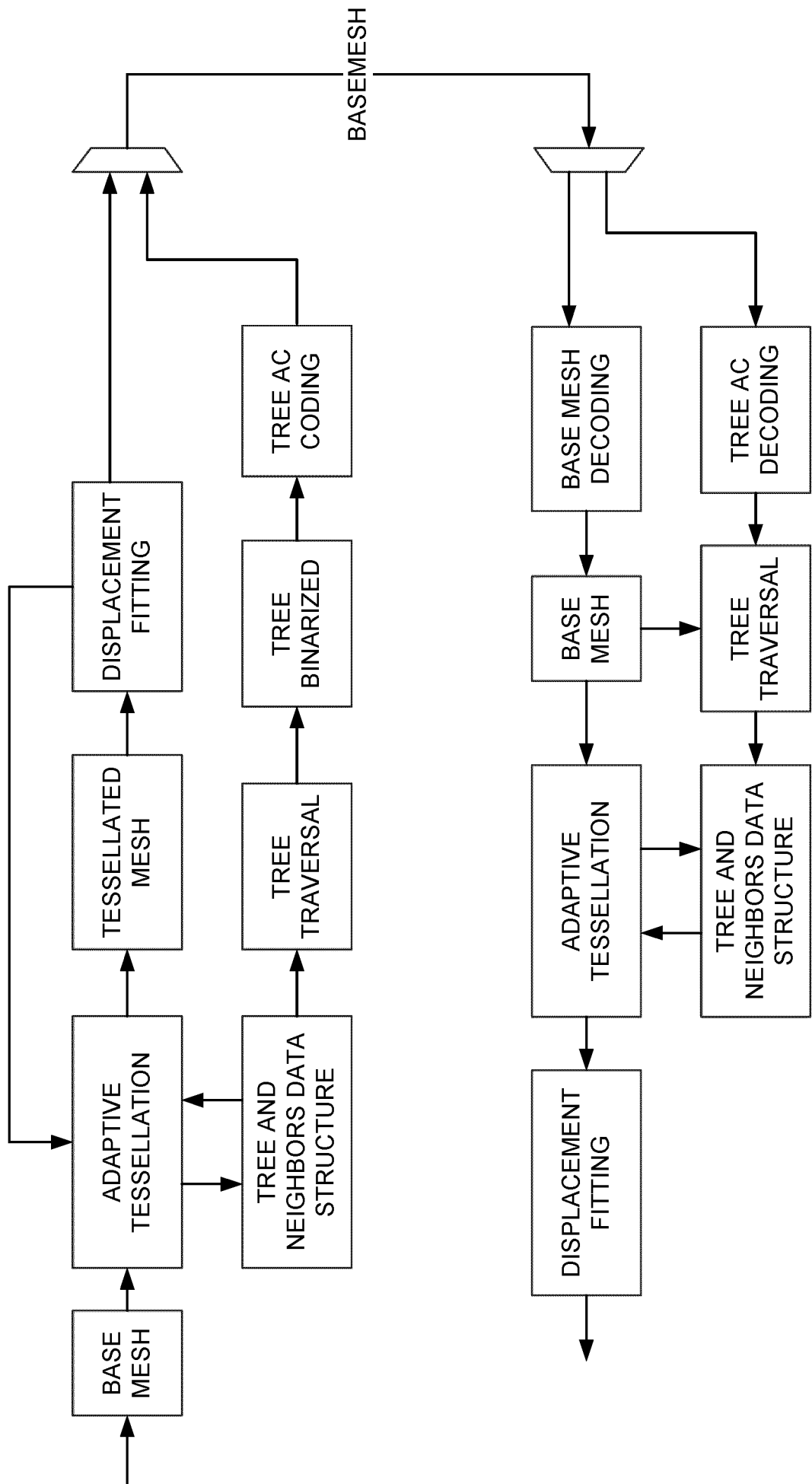


FIG. 12

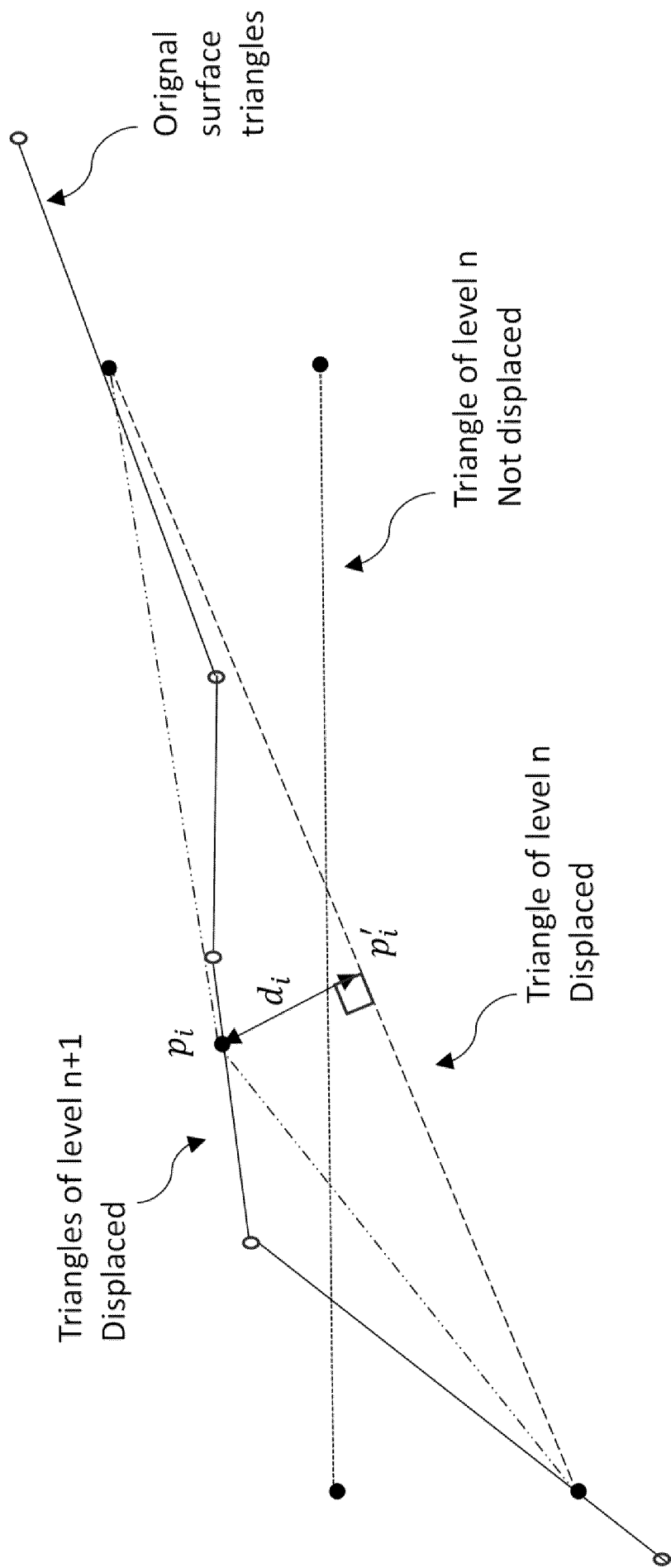


FIG. 13

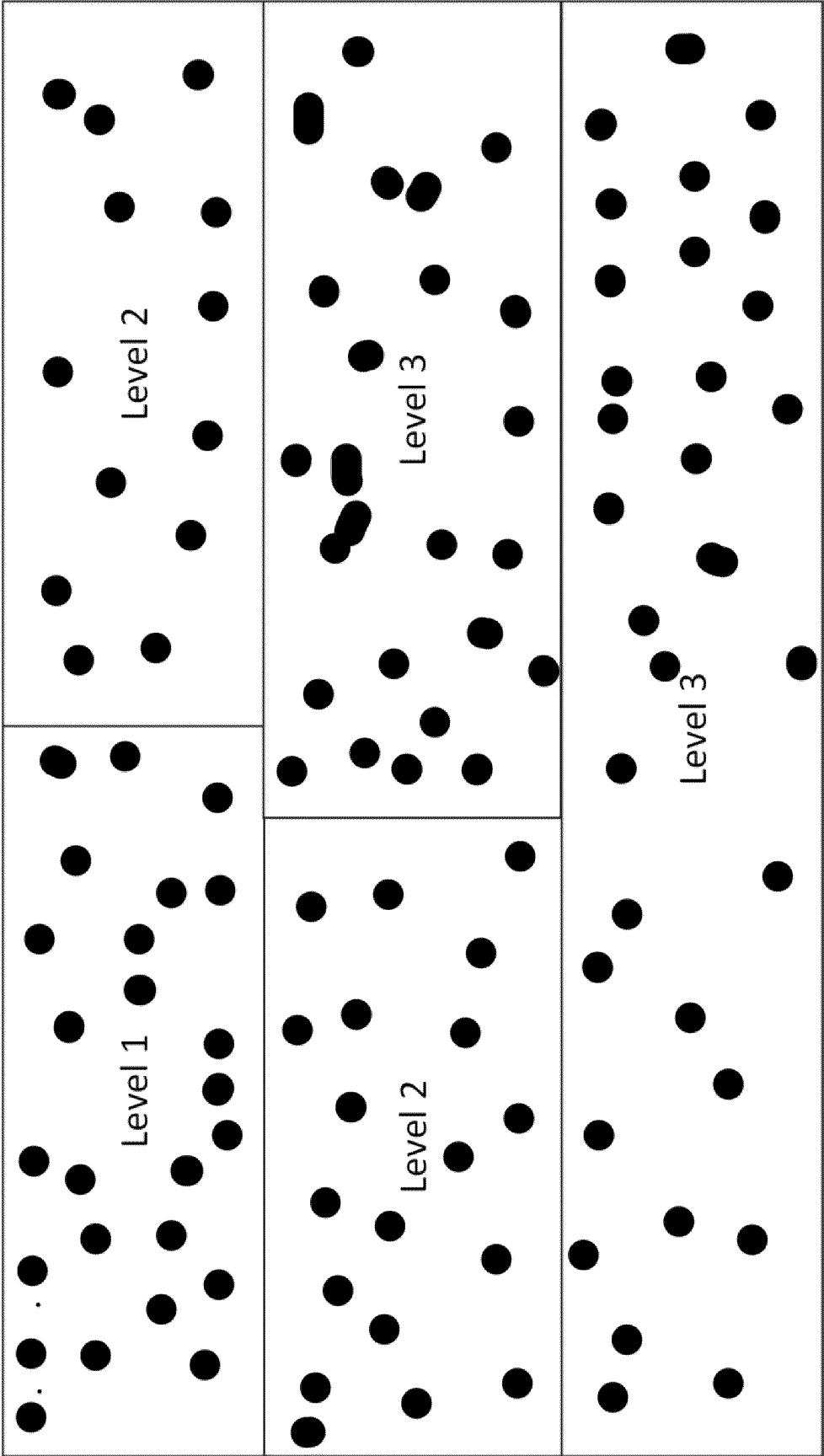


FIG. 14A

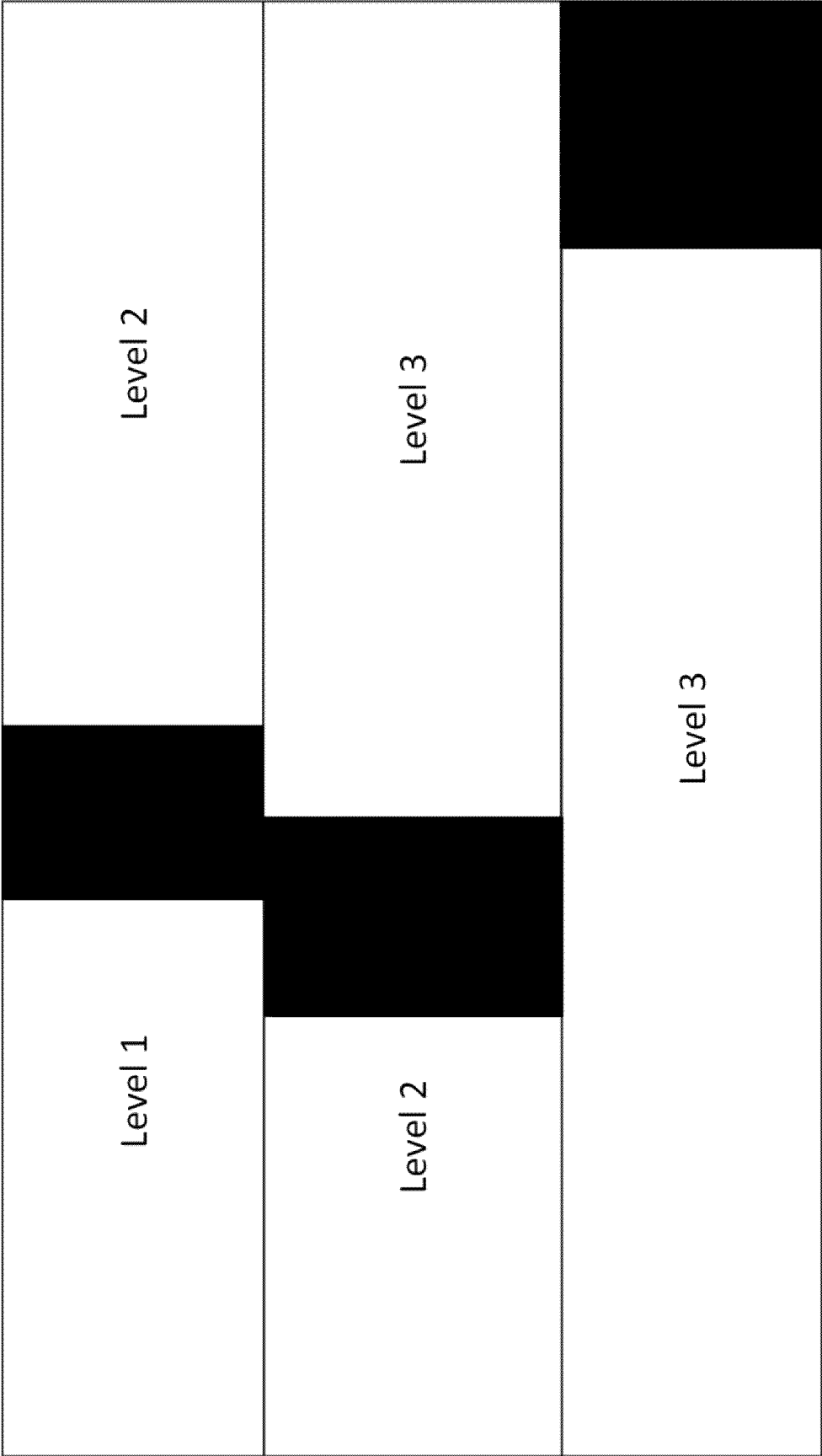


FIG. 14B

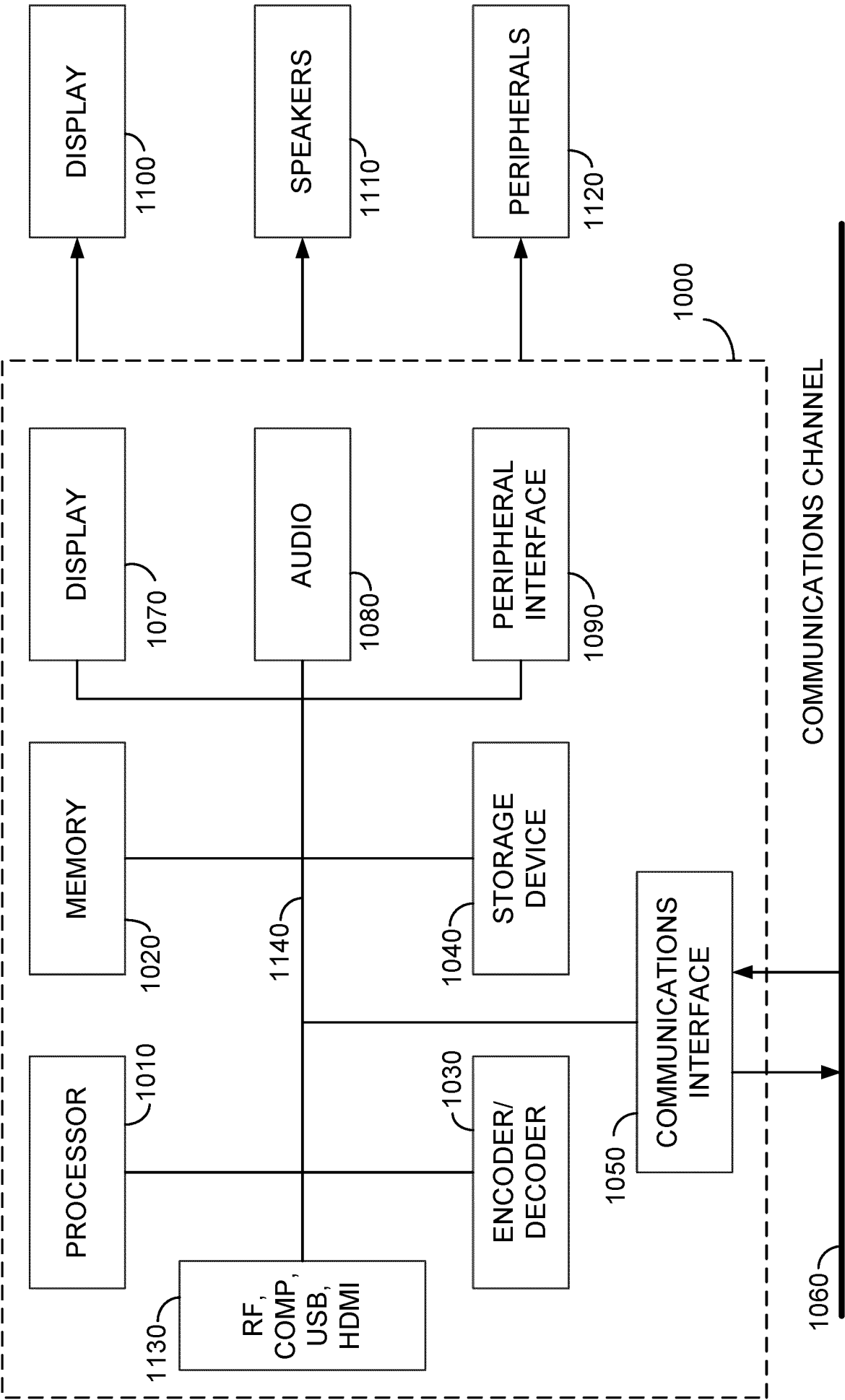


FIG. 15

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2024/085612

A. CLASSIFICATION OF SUBJECT MATTER INV. G06T9/00 G06T9/40 H04N19/96 ADD.		
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) G06T H04N 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) EPO-Internal		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CHAI B-B ET AL: "Depth map compression for real-time view-based rendering", PATTERN RECOGNITION LETTERS, ELSEVIER, AMSTERDAM, NL, vol. 25, no. 7, 1 May 2004 (2004-05-01), pages 755-766, XP004500943, ISSN: 0167-8655, DOI: 10.1016/J.PATREC.2004.01.002 abstract; figures 4,5,6 -----	1-15
X	CHRISTIAN KEIMEL ET AL: "Improving the Visual Quality of AVC/H.264 by Combining It with Content Adaptive Depth Map Compression", PICTURE CODING SYMPOSIUM 2010; 8-12-2010 - 10-12-2010; NAGOYA,, 8 December 2010 (2010-12-08), XP030082037, paragraph [03.3] -----	1-6,8-14
☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.		
* Special categories of cited documents : "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance;; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance;; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search 24 January 2025		Date of mailing of the international search report 03/02/2025
Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016		Authorized officer Oelbaum, Tobias