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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
16 January 2025 (16.01.2025)

WIPO | PCT

(10) International Publication Number
WO 2025/015282 A1

(51) International Patent Classification:
H04N 19/597 (2014.01) *G06T 9/00* (2006.01)

(21) International Application Number:
PCT/US2024/037831

(22) International Filing Date:
12 July 2024 (12.07.2024)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
63/526,537 13 July 2023 (13.07.2023) US
63/526,761 14 July 2023 (14.07.2023) US

(71) Applicant: **COMCAST CABLE COMMUNICATIONS, LLC** [US/US]; 1701 JFK Boulevard, Philadelphia, Pennsylvania 19103 (US).

(72) Inventors: **LASSERRE, Sébastien**; c/o Comcast Cable Communications, LLC, 1701 JFK Boulevard, Philadelphia, Pennsylvania 19103 (US). **TAQUET, Jonathan**; c/o Comcast Cable Communications, LLC, 1701 JFK Boulevard, Philadelphia, Pennsylvania 19103 (US).

(74) Agent: **DIKE, Azuka C.**; 71 South Wacker Drive, Suite 3600, Chicago, Illinois 60606 (US).

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

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ,

(54) **Title:** DUAL MOTION FIELDS FOR CODING GEOMETRY AND ATTRIBUTES OF A POINT CLOUD

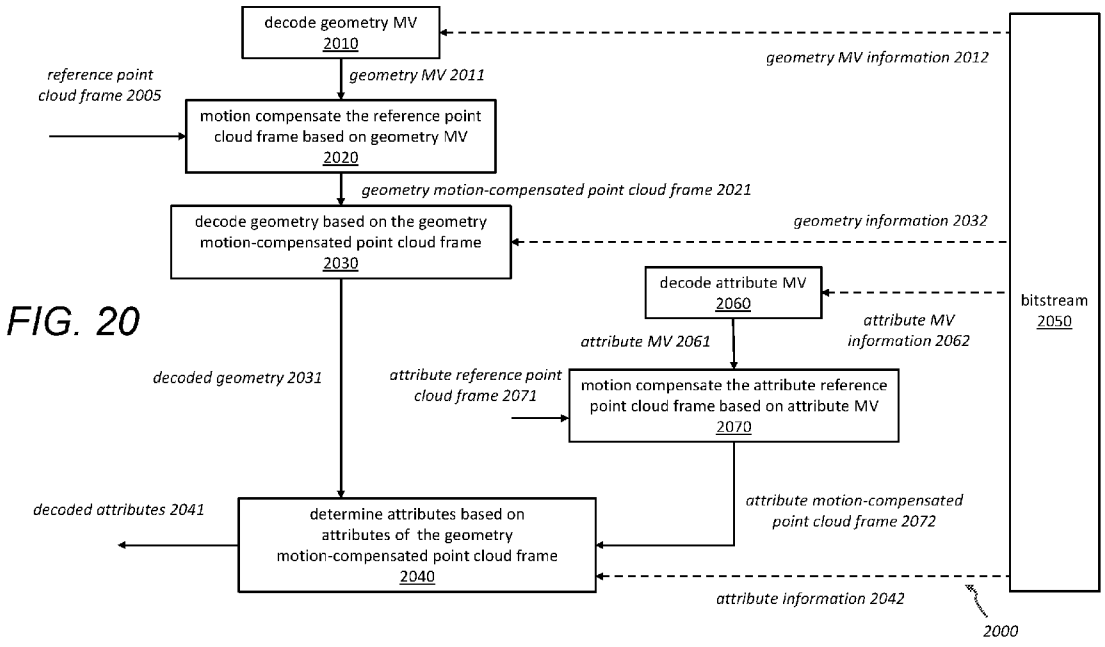

FIG. 20

(57) **Abstract:** Systems, apparatuses, methods, and computer-readable media are described for determining and/or coding geometry and attribute information of a point cloud frame associated with content. The attribute information associated with a reconstructed geometry of a point cloud frame may be coded (e.g., encoded and/or decoded) based on attribute motion vectors associated with the point cloud frame. The attribute motion vectors may be used to determine an attribute motion-compensated point cloud frame.

[Continued on next page]

WO 2025/015282 A1

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))*

Dual Motion Fields for Coding Geometry and Attributes of a Point Cloud

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Provisional Application No. 63/526,761 filed on July 14, 2023, and U.S. Provisional Application No. 63/526,537 filed on July 13, 2023. The above referenced applications are hereby incorporated by reference in their entirety.

BACKGROUND

[0002] An object or scene may be described using volumetric visual data consisting of a series of points. The points may be stored as a point cloud format that includes a collection of points in three-dimensional space. As point clouds can get quite large in data size, transmitting and processing point cloud data may need a data compression scheme that is specifically designed with respect to the unique characteristics of point cloud data.

SUMMARY

[0003] The following summary presents a simplified summary of certain features. The summary is not an extensive overview and is not intended to identify key or critical elements.

[0004] Point cloud information associated with content may comprise geometry information and attribute information (e.g., color or texture of the geometry). Attribute information may be coded separately from geometry information. A reference point cloud frame may be selected for predicting attributes of a current point cloud frame. Geometry information associated with a point cloud frame and attributes associated with a reconstructed geometry of the point cloud frame may be encoded. The reconstructed geometry of the point cloud frame may be determined by decoding a geometry associated with a point cloud frame. Attributes associated with the reconstructed geometry may be decoded based on attribute motion vectors. By encoding and/or decoding the attributes, coding costs (e.g., bitrate) and/or distortion for inter-frame prediction may be reduced.

[0005] These and other features and advantages are described in greater detail below.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] Examples of several of the various embodiments of the present disclosure are described herein with reference to the drawings.

[0007] FIG. 1 shows an example point cloud coding system.

[0008] FIG. 2 shows an example Morton order.

[0009] FIG. 3 shows an example scanning order.

[0010] FIG. 4 shows an example neighborhood of cuboids for entropy coding the occupancy of a child cuboid.

[0011] FIG. 5 shows an example of a dynamic reduction function DR that may be used in dynamic OBUF.

[0012] FIG. 6 shows an example method for coding occupancy of a cuboid using dynamic OBUF.

[0013] FIG. 7 shows an example of an occupied cuboid.

[0014] FIG. 8A shows an example cuboid corresponding to a TriSoup node.

[0015] FIG. 8B shows an example refinement to the TriSoup model.

[0016] FIG. 9 illustrates an example of voxelization.

[0017] FIG. 10 shows an example encoding method using inter frame prediction.

[0018] FIG. 11 shows an example method for encoding point cloud attributes based on prediction transform.

[0019] FIG. 12 shows an example method for decoding point cloud attributes based on prediction transform.

[0020] FIG. 13 shows an example method for encoding point cloud attributes based on prediction with lifting transform.

[0021] FIG. 14 shows an example method for decoding point cloud attributes based on prediction with lifting transform.

[0022] FIG. 15 shows an example region adaptive hierarchical transform (RAHT)transformation used for an octree.

[0023] FIG. 16 shows another example of the RAHT transformation used for an octree.

[0024] FIG. 17 shows an example method for encoding a current point cloud frame.

[0025] FIG. 18 shows an example method for decoding a current point cloud frame.

[0026] FIG. 19 shows an example method for encoding a current point cloud frame.

[0027] FIG. 20 shows an example method for decoding a current point cloud frame.

[0028] FIG. 21 shows an example method for encoding attributes of a current point cloud frame.

[0029] FIG. 22 shows an example method for decoding attributes of a current point cloud frame.

[0030] FIG. 23 shows an example method for encoding attributes of a point cloud frame.

[0031] FIG. 24 shows an example method for decoding attribute information.

[0032] FIG. 25 shows an example method for encoding attributes of a point cloud frame.

[0033] FIG. 26 shows an example method for decoding attribute information.

[0034] FIG. 27 shows an example method for encoding residual attributes.

[0035] FIG. 28 shows an example method for decoding residual attributes.

[0036] FIG. 29 shows an example method for encoding a current point cloud frame.

[0037] FIG. 30 shows an example method for decoding a current point cloud frame.

[0038] FIG. 31 shows an example method for encoding an attribute motion vector.

[0039] FIG. 32 shows an example method for decoding an attribute motion vector.

[0040] FIG. 33 shows an example method for encoding a point cloud frame.

[0041] FIG. 34 shows an example method for decoding a point cloud frame.

[0042] FIG. 35 shows an example computer system in which examples of the present disclosure may be implemented.

[0043] FIG. 36 shows example elements of a computing device that may be used to implement any of the various devices described herein.

DETAILED DESCRIPTION

[0044] The accompanying drawings and descriptions provide examples. It is to be understood that the examples shown in the drawings and/or described are non-exclusive, and that features shown and described may be practiced in other examples. Examples are provided for operation of point cloud or point cloud sequence encoding or decoding systems. More particularly, the technology disclosed herein may relate to point cloud compression as used in encoding and/or decoding devices and/or systems.

[0045] At least some visual data may describe an object or scene in content and/or media using a series of points. Each point may comprise a position in two dimensions (x and y) and one or more optional attributes like color. Volumetric visual data may add another positional dimension to these visual data. For example, volumetric visual data may describe an object or scene in content and/or media using a series of points that each may comprise a position in three dimensions (x, y, and z) and one or more optional attributes like color, reflectance, time stamp, etc. Volumetric visual data may provide a more immersive way to experience visual data, for example, compared to the at least some visual data. For example, an object or scene described by volumetric visual data may be viewed from any (or multiple) angles, whereas the at least some visual data may generally only be viewed from the angle in which it was captured or rendered. As a format for the representation of visual data (e.g., volumetric visual data, three-dimensional video data, etc.) point clouds are versatile in their capability in representing all types of three-dimensional (3D) objects, scenes, and visual content. Point clouds are well suited for use in various applications including, among others: movie post-production, real-time 3D immersive media or telepresence, extended reality, free viewpoint video, geographical information systems, autonomous driving, 3D mapping, visualization, medicine, multi-view replay, and real-time Light Detection and Ranging (LiDAR) data acquisition.

[0046] As explained herein, volumetric visual data may be used in many applications, including extended reality (XR). XR encompasses various types of immersive technologies,

including augmented reality (AR), virtual reality (VR), and mixed reality (MR). Sparse volumetric visual data may be used in the automotive industry for the representation of three-dimensional (3D) maps (e.g., cartography) or as input to assisted driving systems. In the case of assisted driving systems, volumetric visual data may be typically input to driving decision algorithms. Volumetric visual data may be used to store valuable objects in digital form. In applications for preserving cultural heritage, a goal may be to keep a representation of objects that may be threatened by natural disasters. For example, statues, vases, and temples may be entirely scanned and stored as volumetric visual data having several billions of samples. This use-case for volumetric visual data may be particularly relevant for valuable objects in locations where earthquakes, tsunamis and typhoons are frequent. Volumetric visual data may take the form of a volumetric frame. The volumetric frame may describe an object or scene captured at a particular time instance. Volumetric visual data may take the form of a sequence of volumetric frames (referred to as a volumetric sequence or volumetric video). The sequence of volumetric frames may describe an object or scene captured at multiple different time instances.

[0047] Volumetric visual data may be stored in various formats. A point cloud may comprise a collection of points in a 3D space. Such points may be used create a mesh comprising vertices and polygons, or other forms of visual content. As described herein, point cloud data may take the form of a point cloud frame, which describes an object or scene in content that is captured at a particular time instance. Point cloud data may take the form of a sequence of point cloud frames (e.g., point cloud video). As further described herein, point cloud data may be encoded by a source device (e.g., source device 102 as described herein with respect to FIG. 1) that outputs a bitstream containing the encoded point cloud data. The source device may encode the point cloud data based on point cloud compression coding, for example, geometry-based point cloud compression (G-PCC) coding and/or video-based point cloud compression (V-PCC) coding, or next generation coding. A destination device (e.g., destination device 106 as described herein with respect to FIG. 1) receives the bitstream containing the point cloud data and decodes the bitstream containing the point cloud data. The destination device may decode the point cloud data by performing point cloud decompression coding. The decompression coding may be an inverse process of the point cloud compression coding. The point cloud decompression coding may include, for example, G-PCC coding. Decoding may be used to decompress the point cloud data for display and/or other forms of consumption (e.g., further analysis, storage, etc.). The destination device (or a different device) may include, for example, a

renderer for rendering the decoded point cloud data. The renderer may output content, for example, by rendering the point cloud data. The renderer may output content, for example, by rendering the point cloud data along with other data (e.g., audio data).

[0048] One format for storing volumetric visual data may be point clouds. A point cloud may comprise a collection of points in 3D space. Each point in a point cloud may comprise geometry information that may indicate the point's position in 3D space. For example, the geometry information may indicate the point's position in 3D space, for example, using three Cartesian coordinates (x, y, and z) and/or using spherical coordinates (r, phi, theta) (e.g., if acquired by a rotating sensor). The positions of points in a point cloud may be quantized according to a space precision. The space precision may be the same or different in each dimension. The quantization process may create a grid in 3D space. One or more points residing within each sub-grid volume may be mapped to the sub-grid center coordinates, referred to as voxels. A voxel may be considered as a 3D extension of pixels corresponding to the 2D image grid coordinates. For example, similar to a pixel being the smallest unit in the example of dividing the 2D space (or 2D image) into discrete, uniform (e.g., equally sized) regions, a voxel may be the smallest unit of volume in the example of dividing 3D space into discrete, uniform regions. A point in a point cloud may comprise one or more types of attribute information. Attribute information may indicate a property of a point's visual appearance. For example, attribute information may indicate a texture (e.g., color) of the point, a material type of the point, transparency information of the point, reflectance information of the point, a normal vector to a surface of the point, a velocity at the point, an acceleration at the point, a time stamp indicating when the point was captured, or a modality indicating how the point was captured (e.g., running, walking, or flying). A point in a point cloud may comprise light field data in the form of multiple view-dependent texture information. Light field data may be another type of optional attribute information.

[0049] The points in a point cloud may describe an object or a scene. For example, the points in a point cloud may describe the external surface and/or the internal structure of an object or scene. The object or scene may be synthetically generated by a computer. The object or scene may be generated from the capture of a real-world object or scene. The geometry information of a real-world object or a scene may be obtained by 3D scanning and/or photogrammetry. 3D scanning may include different types of scanning, for example, laser scanning, structured light scanning, and/or modulated light scanning. 3D scanning may obtain geometry information. 3D scanning may obtain geometry information, for

example, by moving one or more laser heads, structured light cameras, and/or modulated light cameras relative to an object or scene being scanned. Photogrammetry may obtain geometry information. Photogrammetry may obtain geometry information, for example, by triangulating the same feature or point in different spatially shifted 2D photographs. Point cloud data may take the form of a point cloud frame. The point cloud frame may describe an object or scene captured at a particular time instance. Point cloud data may take the form of a sequence of point cloud frames. The sequence of point cloud frames may be referred to as a point cloud sequence or point cloud video. The sequence of point cloud frames may describe an object or scene captured at multiple different time instances.

[0050] The data size of a point cloud frame or point cloud sequence may be excessive (e.g., too large) for storage and/or transmission in many applications. For example, a single point cloud may comprise over a million points or even billions of points. Each point may comprise geometry information and one or more optional types of attribute information. The geometry information of each point may comprise three Cartesian coordinates (x, y, and z) and/or spherical coordinates (r, phi, theta) that may be each represented, for example, using at least 10 bits per component or 30 bits in total. The attribute information of each point may comprise a texture corresponding to a plurality of (e.g., three) color components (e.g., R, G, and B color components). Each color component may be represented, for example, using 8-10 bits per component or 24-30 bits in total. For example, a single point may comprise at least 54 bits of information, with at least 30 bits of geometry information and at least 24 bits of texture. If a point cloud frame includes a million such points, each point cloud frame may require 54 million bits or 54 megabits to represent. For dynamic point clouds that change over time, at a frame rate of 30 frames per second, a data rate of 1.32 gigabits per second may be required to send (e.g., transmit) the points of the point cloud sequence. Raw representations of point clouds may require a large amount of data, and the practical deployment of point-cloud-based technologies may need compression technologies that enable the storage and distribution of point clouds with a reasonable cost.

[0051] Encoding may be used to compress and/or reduce the data size of a point cloud frame or point cloud sequence to provide for more efficient storage and/or transmission. Decoding may be used to decompress a compressed point cloud frame or point cloud sequence for display and/or other forms of consumption (e.g., by a machine learning based device, neural network-based device, artificial intelligence-based device, or other forms of consumption by other types of machine-based processing algorithms and/or devices).

Compression of point clouds may be lossy (introducing differences relative to the original data) for the distribution to and visualization by an end-user, for example, on AR or VR glasses or any other 3D-capable device. Lossy compression may allow for a high ratio of compression but may imply a trade-off between compression and visual quality perceived by an end-user. Other frameworks, for example, frameworks for medical applications or autonomous driving, may require lossless compression to avoid altering the results of a decision obtained, for example, based on the analysis of the sent (e.g., transmitted) and decompressed point cloud frame.

[0052] FIG. 1 shows an example point cloud coding (e.g., encoding and/or decoding) system 100. Point cloud coding system 100 may comprise a source device 102, a transmission medium 104, and a destination device 106. Source device 102 may encode a point cloud sequence 108 into a bitstream 110 for more efficient storage and/or transmission. Source device 102 may store and/or send (e.g., transmit) bitstream 110 to destination device 106 via transmission medium 104. Destination device 106 may decode bitstream 110 to display point cloud sequence 108 or for other forms of consumption (e.g., further analysis, storage, etc.). Destination device 106 may receive bitstream 110 from source device 102 via a storage medium or transmission medium 104. Source device 102 and destination device 106 may include any number of different devices. Source device 102 and destination device 106 may include, for example, a cluster of interconnected computer systems acting as a pool of seamless resources (also referred to as a cloud of computers or cloud computer), a server, a desktop computer, a laptop computer, a tablet computer, a smart phone, a wearable device, a television, a camera, a video gaming console, a set-top box, a video streaming device, a vehicle (e.g., an autonomous vehicle), or a head-mounted display. A head-mounted display may allow a user to view a VR, AR, or MR scene and adjust the view of the scene, for example, based on movement of the user's head. A head-mounted display may be connected (e.g., tethered) to a processing device (e.g., a server, a desktop computer, a set-top box, or a video gaming console) or may be fully self-contained.

[0053] A source device 102 may comprise a point cloud source 112, an encoder 114, and an output interface 116. A source device 102 may comprise a point cloud source 112, an encoder 114, and an output interface 116, for example, to encode point cloud sequence 108 into a bitstream 110. Point cloud source 112 may provide (e.g., generate) point cloud sequence 108, for example, from a capture of a natural scene and/or a synthetically generated scene. A synthetically generated scene may be a scene comprising computer

generated graphics. Point cloud source 112 may comprise one or more point cloud capture devices, a point cloud archive comprising previously captured natural scenes and/or synthetically generated scenes, a point cloud feed interface to receive captured natural scenes and/or synthetically generated scenes from a point cloud content provider, and/or a processor(s) to generate synthetic point cloud scenes. The point cloud capture devices may include, for example, one or more laser scanning devices, structured light scanning devices, modulated light scanning devices, and/or passive scanning devices.

[0054] Point cloud sequence 108 may comprise a series of point cloud frames 124 (e.g., an example shown in FIG. 1). A point cloud frame may describe an object or scene captured at a particular time instance. Point cloud sequence 108 may achieve the impression of motion by using a constant or variable time to successively present point cloud frames 124 of point cloud sequence 108. A point cloud frame may comprise a collection of points (e.g., voxels) 126 in 3D space. Each point 126 may comprise geometry information that may indicate the point's position in 3D space. The geometry information may indicate, for example, the point's position in 3D space using three Cartesian coordinates (x, y, and z). One or more of points 126 may comprise one or more types of attribute information. Attribute information may indicate a property of a point's visual appearance. For example, attribute information may indicate, for example, a texture (e.g., color) of a point, a material type of a point, transparency information of a point, reflectance information of a point, a normal vector to a surface of a point, a velocity at a point, an acceleration at a point, a time stamp indicating when a point was captured, a modality indicating how a point was captured (e.g., running, walking, or flying), etc. One or more of points 126 may comprise, for example, light field data in the form of multiple view-dependent texture information. Light field data may be another type of optional attribute information. Color attribute information of one or more of points 126 may comprise a luminance value and two chrominance values. The luminance value may represent the brightness (e.g., luma component, Y) of the point. The chrominance values may respectively represent the blue and red components of the point (e.g., chroma components, Cb and Cr) separate from the brightness. Other color attribute values may be represented, for example, based on different color schemes (e.g., an RGB or monochrome color scheme).

[0055] Encoder 114 may encode point cloud sequence 108 into a bitstream 110. To encode point cloud sequence 108, encoder 114 may use one or more lossless or lossy compression techniques to reduce redundant information in point cloud sequence 108. To encode point cloud sequence 108, encoder 114 may use one or more prediction techniques to reduce

redundant information in point cloud sequence 108. Redundant information is information that may be predicted at a decoder 120 and may not be needed to be sent (e.g., transmitted) to decoder 120 for accurate decoding of point cloud sequence 108. For example, Motion Picture Expert Group (MPEG) introduced a geometry-based point cloud compression (G-PCC) standard (ISO/IEC standard 23090-9: Geometry-based point cloud compression). G-PCC specifies the encoded bitstream syntax and semantics for transmission and/or storage of a compressed point cloud frame and the decoder operation for reconstructing the compressed point cloud frame from the bitstream. During standardization of G-PCC, a reference software (ISO/IEC standard 23090-21: Reference Software for G-PCC) was developed to encode the geometry and attribute information of a point cloud frame. To encode geometry information of a point cloud frame, the G-PCC reference software encoder may perform voxelization. The G-PCC reference software encoder may perform voxelization, for example, by quantizing positions of points in a point cloud. Quantizing positions of points in a point cloud may create a grid in 3D space. The G-PCC reference software encoder may map the points to the center coordinates of the sub-grid volume (e.g., voxel) that their quantized locations reside in. The G-PCC reference software encoder may perform geometry analysis using an occupancy tree to compress the geometry information. The G-PCC reference software encoder may entropy encode the result of the geometry analysis to further compress the geometry information. To encode attribute information of a point cloud, the G-PCC reference software encoder may use a transform tool, such as Region Adaptive Hierarchical Transform (RAHT), the Predicting Transform, and/or the Lifting Transform. The Lifting Transform may be built on top of the Predicting Transform. The Lifting Transform may include an extra update/lifting step. The Lifting Transform and the Predicting Transform may be referred to as Predicting/Lifting Transform or pred lift. Encoder 114 may operate in a same or similar manner to an encoder provided by the G-PCC reference software.

[0056] Output interface 116 may be configured to write and/or store bitstream 110 onto transmission medium 104. The bitstream 110 may be sent (e.g., transmitted) to destination device 106. In addition or alternatively, output interface 116 may be configured to send (e.g., transmit), upload, and/or stream bitstream 110 to destination device 106 via transmission medium 104. Output interface 116 may comprise a wired and/or wireless transmitter configured to send (e.g., transmit), upload, and/or stream bitstream 110 according to one or more proprietary, open-source, and/or standardized communication protocols. The one or more proprietary, open-source, and/or standardized communication

protocols may include, for example, Digital Video Broadcasting (DVB) standards, Advanced Television Systems Committee (ATSC) standards, Integrated Services Digital Broadcasting (ISDB) standards, Data Over Cable Service Interface Specification (DOCSIS) standards, 3rd Generation Partnership Project (3GPP) standards, Institute of Electrical and Electronics Engineers (IEEE) standards, Internet Protocol (IP) standards, Wireless Application Protocol (WAP) standards, and/or any other communication protocol.

[0057] Transmission medium 104 may comprise a wireless, wired, and/or computer readable medium. For example, transmission medium 104 may comprise one or more wires, cables, air interfaces, optical discs, flash memory, and/or magnetic memory. In addition or alternatively, transmission medium 104 may comprise one or more networks (e.g., the Internet) or file server(s) configured to store and/or send (e.g., transmit) encoded video data.

[0058] Destination device 106 may decode bitstream 110 into point cloud sequence 108 for display or other forms of consumption. Destination device 106 may comprise one or more of an input interface 118, a decoder 120, and/or a point cloud display 122. Input interface 118 may be configured to read bitstream 110 stored on transmission medium 104. Bitstream 110 may be stored on transmission medium 104 by source device 102. In addition or alternatively, input interface 118 may be configured to receive, download, and/or stream bitstream 110 from source device 102 via transmission medium 104. Input interface 118 may comprise a wired and/or wireless receiver configured to receive, download, and/or stream bitstream 110 according to one or more proprietary, open-source, standardized communication protocols, and/or any other communication protocol. Examples of the protocols include Digital Video Broadcasting (DVB) standards, Advanced Television Systems Committee (ATSC) standards, Integrated Services Digital Broadcasting (ISDB) standards, Data Over Cable Service Interface Specification (DOCSIS) standards, 3rd Generation Partnership Project (3GPP) standards, Institute of Electrical and Electronics Engineers (IEEE) standards, Internet Protocol (IP) standards, and Wireless Application Protocol (WAP) standards.

[0059] Decoder 120 may decode point cloud sequence 108 from encoded bitstream 110. For example, decoder 120 may operate in a same or similar manner as a decoder provided by G-PCC reference software. Decoder 120 may decode a point cloud sequence that approximates a point cloud sequence 108. Decoder 120 may decode a point cloud sequence that approximates a point cloud sequence 108 due to, for example, lossy

compression of the point cloud sequence 108 by encoder 114 and/or errors introduced into encoded bitstream 110, for example, if transmission to destination device 106 occurs.

[0060] Point cloud display 122 may display a point cloud sequence 108 to a user. The point cloud display 122 may comprise, for example, a cathode rate tube (CRT) display, a liquid crystal display (LCD), a plasma display, a light emitting diode (LED) display, a 3D display, a holographic display, a head-mounted display, or any other display device suitable for displaying point cloud sequence 108.

[0061] Point cloud coding (e.g., encoding/decoding) system 100 is presented by way of example and not limitation. Point cloud coding systems different from the point cloud coding system 100 and/or modified versions of the point cloud coding system 100 may perform the methods and processes as described herein. For example, the point cloud coding system 100 may comprise other components and/or arrangements. Point cloud source 112 may, for example, be external to source device 102. Point cloud display device 122 may, for example, be external to destination device 106 or omitted altogether (e.g., if point cloud sequence 108 is intended for consumption by a machine and/or storage device). Source device 102 may further comprise, for example, a point cloud decoder. Destination device 106 may comprise, for example, a point cloud encoder. For example, source device 102 may be configured to further receive an encoded bit stream from destination device 106. Receiving an encoded bit stream from destination device 106 may support two-way point cloud transmission between the devices.

[0062] As described herein, an encoder may quantize the positions of points in a point cloud according to a space precision, which may be the same or different in each dimension of the points. The quantization process may create a grid in 3D space. The encoder may map any points residing within each sub-grid volume to the sub-grid center coordinates, referred to as a voxel or a volumetric pixel. A voxel may be considered as a 3D extension of pixels corresponding to 2D image grid coordinates.

[0063] An encoder may represent or code a point cloud (e.g., a voxelized). An encoder may represent or code a point cloud, for example, using an occupancy tree. For example, the encoder may split the initial volume or cuboid containing the point cloud into sub-cuboids. The initial volume or cuboid may be referred to as a bounding box. A cuboid may be, for example, a cube. The encoder may recursively split each sub-cuboid that contains at least one point of the point cloud. The encoder may not further split sub-cuboids that do not contain at least one point of the point cloud. A sub-cuboid that contains at least one point of the point cloud may be referred to as an occupied sub-cuboid. A sub-cuboid that does

not contain at least one point of the point cloud may be referred to as an unoccupied sub-cuboid. The encoder may split an occupied sub-cuboid into, for example, two sub-cuboids (to form a binary tree), four sub-cuboids (to form a quadtree), or eight sub-cuboids (to form an octree). The encoder may split an occupied sub-cuboid to obtain further sub-cuboids. The sub-cuboids may have the same size and shape at a given depth level of the occupancy tree. The sub-cuboids may have the same size and shape at a given depth level of the occupancy tree, for example, if the encoder splits the occupied sub-cuboid along a plane passing through the middle of edges of the sub-cuboid.

[0064] The initial volume or cuboid containing the point cloud may correspond to the root node of the occupancy tree. Each occupied sub-cuboid, split from the initial volume, may correspond to a node (of the root node) in a second level of the occupancy tree. Each occupied sub-cuboid, split from an occupied sub-cuboid in the second level, may correspond to a node (off the occupied sub-cuboid in the second level from which it was split) in a third level of the occupancy tree. The occupancy tree structure may continue to form in this manner for each recursive split iteration until, for example, some maximum depth level of the occupancy tree is reached or each occupied sub-cuboid has a volume corresponding to one voxel.

[0065] Each non-leaf node of the occupancy tree may comprise or be associated with an occupancy word representing the occupancy state of the cuboid corresponding to the node. For example, a node of the occupancy tree corresponding to a cuboid that is split into 8 sub-cuboids may comprise or be associated with a 1-byte occupancy word. Each bit (referred to as an occupancy bit) of the 1-byte occupancy word may represent or indicate the occupancy of a different one of the eight sub-cuboids. Occupied sub-cuboids may be each represented or indicated by a binary "1" in the 1-byte occupancy word. Unoccupied sub-cuboids may be each represented or indicated by a binary "0" in the 1-byte occupancy word. Occupied and un-occupied sub-cuboids may be represented or indicated by opposite 1-bit binary values (e.g., a binary "0" representing or indicating an occupied sub-cuboid and a binary "1" representing or indicating an unoccupied sub-cuboid) in the 1-byte occupancy word.

[0066] Each bit of an occupancy word may represent or indicate the occupancy of a different one of the eight sub-cuboids. Each bit of an occupancy word may represent or indicate the occupancy of a different one of the eight sub-cuboids, for example, following the so-called Morton order. For example, the least significant bit of an occupancy word may represent or indicate, for example, the occupancy of a first one of the eight sub-cuboids following

the Morton order. The second least significant bit of an occupancy word may represent or indicate, for example, the occupancy of a second one of the eight sub-cuboids following the Morton order, etc.

[0067] FIG. 2 shows an example Morton order. More specifically, FIG. 2 shows a Morton order of eight sub-cuboids 202-216 split from a cuboid 200. Sub-cuboids 202-216 may be labeled, for example, based on their Morton order, with child node 202 being the first in Morton order and child node 216 being the last in Morton order. The Morton order for sub-cuboids 202-216 may be a local lexicographic order in xyz.

[0068] The geometry of a point cloud may be represented by, and may be determined from, the initial volume and the occupancy words of the nodes in an occupancy tree. An encoder may send (e.g., transmit) the initial volume and the occupancy words of the nodes in the occupancy tree in a bitstream to a decoder for reconstructing the point cloud. The encoder may entropy encode the occupancy words. The encoder may entropy encode the occupancy words, for example, before sending (e.g., transmitting) the initial volume and the occupancy words of the nodes in the occupancy tree. The encoder may encode an occupancy bit of an occupancy word of a node corresponding to a cuboid. The encoder may encode an occupancy bit of an occupancy word of a node corresponding to a cuboid, for example, based on one or more occupancy bits of occupancy words of other nodes corresponding to cuboids that are adjacent or spatially close to the cuboid of the occupancy bit being encoded.

[0069] An encoder and/or a decoder may code (e.g., encode and/or decode) occupancy bits of occupancy words in sequence of a scan order. The scan order may also be referred to as a scanning order. For example, an encoder and/or a decoder may scan an occupancy tree in breadth-first order. All the occupancy words of the nodes of a given depth (e.g., level) within the occupancy tree may be scanned. All the occupancy words of the nodes of a given depth (e.g., level) within the occupancy tree may be scanned, for example, before scanning the occupancy words of the nodes of the next depth (e.g., level). Within a given depth, the encoder and/or decoder may scan the occupancy words of nodes in the Morton order. Within a given node, the encoder and/or decoder may scan the occupancy bits of the occupancy word of the node further in the Morton order.

[0070] FIG. 3 shows an example scanning order. FIG. 3 shows an example scanning order (e.g., breadth-first order as described herein) for an occupancy tree 300. More specifically, FIG. 3 shows a scanning order for the first three example levels of an occupancy tree 300. In FIG. 3, a cuboid (e.g., cube) 302 corresponding to a root node of the occupancy tree 300

may be divided into eight sub-cuboids (e.g., sub-cubes). Two sub-cuboids 304 and 306 of the eight sub-cuboids may be occupied. The other six sub-cuboids of the eight sub-cuboids may be unoccupied. Following the Morton order, a first eight-bit occupancy word (e.g., $occW_{1,1}$) may be constructed to represent the occupancy word of the root node. An (e.g., each) occupancy bit of the first eight-bit occupancy word (e.g., $occW_{1,1}$) may represent or indicate the occupancy of a sub-cube of the eight sub-cuboids in the Morton order. For example, the least significant occupancy bit of the first eight-bit occupancy word $occW_{1,1}$ may represent or indicate the occupancy of the first sub-cuboid of the eight sub-cuboids in the Morton order. The second least significant occupancy bit of the first eight-bit occupancy word $occW_{1,1}$ may represent or indicate the occupancy of the second sub-cuboid of the eight sub-cuboids in the Morton order, etc.

[0071] Each of occupied sub-cuboids (e.g., two occupied sub-cuboids 304 and 306) may correspond to a node off the root node in a second level of an occupancy tree 300. The occupied sub-cuboids (e.g., two occupied sub-cuboids 304 and 306) may be each further split into eight sub-cuboids. For example, one of the sub-cuboids 308 of the eight sub-cuboids split from the sub-cube 304 may be occupied, and the other seven sub-cuboids may be unoccupied. Three of the sub-cuboids 310, 312, and 314 of the eight sub-cuboids split from the sub-cube 306 may be occupied, and the other five sub-cuboids of the eight sub-cuboids split from the sub-cube 306 may be unoccupied. Two second eight-bit occupancy words $occW_{2,1}$ and $occW_{2,2}$ may be constructed in this order to respectively represent the occupancy word of the node corresponding to the sub-cuboid 304 and the occupancy word of the node corresponding to the sub-cuboid 306.

[0072] Each of occupied sub-cuboids (e.g., four occupied sub-cuboids 308, 310, 312, and 314) may correspond to a node in a third level of an occupancy tree 300. The occupied sub-cuboids (e.g., four occupied sub-cuboids 308, 310, 312, and 314) may be each further split into eight sub-cuboids or 32 sub-cuboids in total. For example, four third level eight-bit occupancy words $occW_{3,1}$, $occW_{3,2}$, $occW_{3,3}$ and $occW_{3,4}$ may be constructed in this order to respectively represent the occupancy word of the node corresponding to the sub-cuboid 308, the occupancy word of the node corresponding to the sub-cuboid 310, the occupancy word of the node corresponding to the sub-cuboid 312, and the occupancy word of the node corresponding to the sub-cuboid 314.

[0073] Occupancy words of an example occupancy tree 300 may be entropy coded (e.g., entropy encoded by an encoder and/or entropy decoded by a decoder), for example, following the scanning order discussed herein (e.g., Morton order). The occupancy words of the

example occupancy tree 300 may be entropy coded (e.g., entropy encoded by an encoder and/or entropy decoded by a decoder) as the succession of the seven occupancy words $occW_{1,1}$ to $occW_{3,4}$, for example, following the scanning order discussed herein. The scanning order discussed herein may be a breadth-first scanning order. The occupancy word(s) of all node(s) having the same depth (or level) as a current parent node may have already been entropy coded, for example, if the occupancy word of a current child node belonging to the current parent node is being entropy coded. For example, the occupancy word(s) of all node(s) having the same depth (e.g., level) as the current child node and having a lower Morton order than the current child node may have also already been entropy coded. Part of the already coded occupancy word(s) may be used to entropy code the occupancy word of the current child node. The already coded occupancy word(s) of neighboring parent and child node(s) may be used, for example, to entropy code the occupancy word of the current child node. The occupancy bit(s) of the occupancy word having a lower Morton order than a particular occupancy bit may have also already been entropy coded and may be used to code the occupancy bit of the occupancy word of the current child node, for example, if the particular occupancy bit of the occupancy word of the current child node is being coded (e.g., entropy coded).

[0074] FIG. 4 shows an example neighborhood of cuboids for entropy coding the occupancy of a child cuboid. More specifically, FIG. 4 shows an example neighborhood of cuboids with already-coded occupancy bits. The neighborhood of cuboids with already-coded occupancy bits may be used to entropy code the occupancy bit of a current child cuboid 400. The neighborhood of cuboids with already-coded occupancy bits may be determined, for example, based on the scanning order of an occupancy tree representing the geometry of the cuboids in FIG. 4 as discussed herein. The neighborhood of cuboids, of a current child cuboid, may include one or more of: a cuboid adjacent to the current child cuboid, a cuboid sharing a vertex with the current child cuboid, a cuboid sharing an edge with the current child cuboid, a cuboid sharing a face with the current child cuboid, a parent cuboid adjacent to the current child cuboid, a parent cuboid sharing a vertex with the current child cuboid, a parent cuboid sharing an edge with the current child cuboid, a parent cuboid sharing a face with the current child cuboid, a parent cuboid adjacent to the current parent cuboid, a parent cuboid sharing a vertex with the current parent cuboid, a parent cuboid sharing an edge with the current parent cuboid, a parent cuboid sharing a face with the current parent cuboid, etc. As shown in FIG. 4, current child cuboid 400 may belong to a current parent cuboid 402. Following the scanning order of the occupancy words and

occupancy bits of nodes of the occupancy tree, the occupancy bits of four child cuboids 404, 406, 408, and 410, belonging to the same current parent cuboid 402, may have already been coded. The occupancy bit of child cuboids 412 of preceding parent cuboids may have already been coded. The occupancy bits of parent cuboids 414, for which the occupancy bits of child cuboids have not already been coded, may have already been coded. The already-coded occupancy bits of cuboids 404, 406, 408, 410, 412, and 414 may be used to code the occupancy bit of the current child cuboid 400.

[0075] The number (e.g., quantity) of possible occupancy configurations (e.g., sets of one or more occupancy words and/or occupancy bits) for a neighborhood of a current child cuboid may be 2^N, where N is the number (e.g., quantity) of cuboids in the neighborhood of the current child cuboid with already-coded occupancy bits. The neighborhood of the current child cuboid may comprise several dozens of cuboids. The neighborhood of the current child cuboid (e.g., several dozens of cuboids) may comprise 26 adjacent parent cuboids sharing a face, an, edge, and/or a vertex with the parent cuboid of the current child cuboid and also several adjacent child cuboids having occupancy bits already coded sharing a face, an edge, or a vertex with the current child cuboid. The occupancy configuration for a neighborhood of the current child cuboid may have billions of possible occupancy configurations, even limited to a subset of the adjacent cuboids, making its direct use impractical. An encoder and/or decoder may use the occupancy configuration for a neighborhood of the current child cuboid to select the context (e.g., a probability model), among a set of contexts, of a binary entropy coder (e.g., binary arithmetic coder) that may code the occupancy bit of the current child cuboid. The context-based binary entropy coding may be similar to the Context Adaptive Binary Arithmetic Coder (CABAC) used in MPEG-H Part 2 (also known as High Efficiency Video Coding (HEVC)).

[0076] An encoder and/or a decoder may use several methods to reduce the occupancy configurations for a neighborhood of a current child cuboid being coded to a practical number (e.g., quantity) of reduced occupancy configurations. The 2^6 or 64 occupancy configurations of the six adjacent parent cuboids sharing a face with the parent cuboid of the current child cuboid may be reduced to 9 occupancy configurations. The occupancy configurations may be reduced by using geometry invariance. An occupancy score for the current child cuboid may be obtained from the 2^{26} occupancy configurations of the 26 adjacent parent cuboids. The score may be further reduced into a ternary occupancy prediction (e.g., "predicted occupied," "unsure", or "predicted unoccupied") by using score thresholds. The number (e.g., quantity) of occupied adjacent child cuboids and the

number (e.g., quantity) of unoccupied adjacent child cuboids may be used instead of the individual occupancies of these child cuboids.

[0077] An encoder and/or a decoder using/employing one or more of the methods described herein may reduce the number (e.g., quantity) of possible occupancy configurations for a neighborhood of a current child cuboid to a more manageable number (e.g., a few thousands). It has been observed that instead of associating a reduced number (e.g., quantity) of contexts (e.g., probability models) directly to the reduced occupancy configurations, another mechanism may be used, namely Optimal Binary Coders with Update on the Fly (OBUF). An encoder and/or a decoder may implement OBUF to limit the number (e.g., quantity) of contexts to a lower number (e.g., 32 contexts).

[0078] OBUF may use a limited number (e.g., 32) of contexts (e.g., probability models). The number (e.g., quantity) of contexts in OBUF may be a fixed number (e.g., fixed quantity). The contexts used by OBUF may be ordered, referred to by a context index (e.g., a context index in the range of 0 to 31), and associated from a lowest virtual probability to a highest virtual probability to code a "1". A Look-Up Table (LUT) of context indices may be initialized at the beginning of a point cloud coding process. For example, the LUT may initially point to a context (e.g., with a context index 15) with the median virtual probability to code a "1" for all input. The LUT may initially point to a context with the median virtual probability to code a "1", among the limited number (e.g., quantity) of contexts, for all input. This LUT may take an occupancy configuration for a neighborhood of current child cuboid as input and output the context index associated with the occupancy configuration. The LUT may have as many entries as reduced occupancy configurations (e.g., around a few thousand entries). The coding of the occupancy bit of a current child cuboid may comprise steps including determining the reduced occupancy configuration of the current child node, obtaining a context index by using the reduced occupancy configuration as an entry to the LUT, coding the occupancy bit of the current child cuboid by using the context pointed to (or indicated) by the context index, and updating the LUT entry corresponding to the reduced occupancy configuration, for example, based on the value of the coded occupancy bit of the current child cuboid. The LUT entry may be decreased to a lower context index value, for example, if a binary "0" (e.g., indicating the current child cuboid is unoccupied) is coded. The LUT entry may be increased to a higher context index value, for example, if a binary "1" (e.g., indicating the current child cuboid is occupied) is coded. The update process of the context index may be, for example, based on a theoretical model of optimal distribution for virtual

probabilities associated with the limited number (e.g., quantity) of contexts. This virtual probability may be fixed by a model and may be different from the internal probability of the context that may evolve, for example, if the coding of bits of data occurs. The evolution of the internal context may follow a well-known process similar to the process in CABAC.

[0079] An encoder and/or a decoder may implement a "dynamic OBUF" scheme. The "dynamic OBUF" scheme may enable an encoder and/or a decoder to handle a much larger number (e.g., quantity) of occupancy configurations for a neighborhood of a current child cuboid, for example, than general OBUF. The use of a larger number (e.g., quantity) of occupancy configurations for a neighborhood of a current child cuboid may lead to improved compression capabilities, and may maintain complexity within reasonable bounds. By using an occupancy tree compressed by OBUF, an encoder and/or a decoder may reach a lossless compression performance as good as 1 bit per point (bpp) for coding the geometry of dense point clouds. An encoder and/or a decoder may implement dynamic OBUF to potentially further reduce the bit rate by more than 25% to 0.7 bpp.

[0080] OBUF may not take as input a large variety of reduced occupancy configurations for a neighborhood of a current child cuboid, and may potentially cause a loss of useful correlation. With OBUF, the size of the LUT of context indices may be increased to handle more various occupancy configurations for a neighborhood of a current child cuboid as input. Due to such increase, statistics may be diluted, and compression performance may be worsened. For example, if the LUT has millions of entries and the point cloud has a hundred thousand points, then most of the entries may be never visited (e.g., looked up, accessed, etc.). Many entries may be visited only a few times and their associated context index may not be updated enough times to reflect any meaningful correlation between the occupancy configuration value and the probability of occupancy of the current child cuboid. Dynamic OBUF may be implemented to mitigate the dilution of statistics due to the increase of the number (e.g., quantity) of occupancy configurations for a neighborhood of a current child cuboid. This mitigation may be performed by a "dynamic reduction" of occupancy configurations in dynamic OBUF.

[0081] Dynamic OBUF may add an extra step of reduction of occupancy configurations for a neighborhood of a current child cuboid, for example, before using the LUT of context indices. This step may be called a dynamic reduction because it evolves, for example, based on the progress of the coding of the point cloud or, more precisely, based on already visited (e.g., looked up in the LUT) occupancy configurations.

[0082] As discussed herein, many possible occupancy configurations for a neighborhood of a current child cuboid may be potentially involved but only a subset may be visited if the coding of a point cloud occurs. This subset may characterize the type of the point cloud. For example, most of the visited occupancy configurations may exhibit occupied adjacent cuboids of a current child cuboid, for example, if AR or VR dense point clouds are being coded. On the other hand, most of the visited occupancy configurations may exhibit only a few occupied adjacent cuboids of a current child cuboid, for example, if sensor-acquired sparse point clouds are being coded. The role of the dynamic reduction may be to obtain a more precise correlation, for example, based on the most visited occupancy configuration while putting aside (e.g., reducing aggressively) other occupancy configurations that are much less visited. The dynamic reduction may be updated on-the-fly. The dynamic reduction may be updated on-the-fly, for example, after each visit (e.g., a lookup in the LUT) of an occupancy configuration, for example, if the coding of occupancy data occurs.

[0083] FIG. 5 shows an example of a dynamic reduction function DR that may be used in dynamic OBUF. The dynamic reduction function DR may be obtained by masking bits β_j of occupancy configurations 500

$$\beta = \beta_1 \dots \beta_K$$

made of K bits. The size of the mask may decrease, for example, if occupancy configurations are visited (e.g., looked up in the LUT) a certain number (e.g., quantity) of times. The initial dynamic reduction function DR^0 may mask all bits for all occupancy configurations such that it is a constant function $DR^0(\beta) = 0$ for all occupancy configurations β. The dynamic reduction function may evolve from a function DR^n to an updated function DR^{n+1}. The dynamic reduction function may evolve from a function DR^n to an updated function DR^{n+1}, for example, after each coding of an occupancy bit. The function may be defined by

$$\beta' = DR^n(\beta) = \beta_1 \dots \beta_{kn(\beta)}$$

where $k_n(\beta)$ 510 is the number (e.g., quantity) of non-masked bits. The initialization of DR^0 may correspond to $k_0(\beta)=0$, and the natural evolution of the reduction function toward finer statistics may lead to an increasing number (e.g., quantity) of non-masked bits $k_n(\beta) \leq k_{n+1}(\beta)$. The dynamic reduction function may be entirely determined by the values of k_n for all occupancy configurations β.

[0084] The visits (e.g., instances of a lookup in the LUT) to occupancy configurations may be tracked by a variable NV(β') for all dynamically reduced occupancy configurations β'= $DR^n(\beta)$. The corresponding number (e.g., quantity) of visits $NV(\beta^{V'})$ may be increased by one, for example, after each instance of coding of an occupancy bit based on an occupancy configuration β^V. If this number (e.g., quantity) of visits $NV(\beta^{V'})$ is greater than a threshold th_V,

$$NV(\beta^{V'}) > th_V$$

then the number (e.g., quantity) of unmasked bits $k_n(\beta)$ may be increased by one for all occupancy configurations β being dynamically reduced to $\beta^{V'}$. This corresponds to replacing the dynamically reduced occupancy configuration $\beta^{V'}$ by the two new dynamically reduced occupancy configurations $\beta^{0'}$ and $\beta^{1'}$ defined by

$$\beta^{0'} = \beta^{V'}0 = \beta^{V}_1 \dots \beta^{V}_{kn(\beta)}0 \text{ and } \beta^{1'} = \beta^{V'}1 = \beta^{V}_1 \dots \beta^{V}_{kn(\beta)}1.$$

In other words, the number (e.g., quantity) of unmasked bits has been increased by one $k_{n+1}(\beta) = k_n(\beta) + 1$ for all occupancy configurations β such that $DR^n(\beta) = \beta^{V'}$. The number (e.g., quantity) of visits of the two new dynamically reduced occupancy configurations may be initialized to zero

$$NV(\beta^{0'}) = NV(\beta^{1'}) = 0. \qquad (I)$$

At the start of the coding, the initial number (e.g., quantity) of visits for the initial dynamic reduction function DR^0 may be set to

$$NV(DR^0(\beta)) = NV(0) = 0,$$

and the evolution of NV on dynamically reduced occupancy configurations may be entirely defined.

[0085] The corresponding LUT entry $LUT[\beta^{V'}]$ may be replaced by the two new entries $LUT[\beta^{0'}]$ and $LUT[\beta^{1'}]$ that are initialized by the coder index associated with $\beta^{V'}$. The corresponding LUT entry $LUT[\beta^{V'}]$ may be replaced by the two new entries $LUT[\beta^{0'}]$ and $LUT[\beta^{1'}]$ that are initialized by the coder index associated with $\beta^{V'}$, for example, if a dynamically reduced occupancy configuration $\beta^{V'}$ is replaced by the two new dynamically reduced occupancy configurations $\beta^{0'}$ and $\beta^{1'}$,

$$LUT[\beta^{0'}] = LUT[\beta^{1'}] = LUT[\beta^{V'}], \qquad (II)$$

and then evolve separately. The evolution of the LUT of coder indices on dynamically reduced occupancy configurations may be entirely defined.

[0086] The reduction function DR^n may be modeled by a series of growing binary trees T^n 520 whose leaf nodes 530 are the reduced occupancy configurations $\beta' = DR^n(\beta)$. The initial tree may be the single root node associated with $0 = DR^0(\beta)$. The replacement of the dynamically reduced to $\beta^{V'}$ by $\beta^{0'}$ and $\beta^{1'}$ may correspond to growing the tree T^n from the leaf node associated with $\beta^{V'}$, for example, by attaching to it two new nodes associated with $\beta^{0'}$ and $\beta^{1'}$. The tree T^{n+1} may be obtained by this growth. The number (e.g., quantity) of visits NV and the LUT of context indices may be defined on the leaf nodes and evolve with the growth of the tree through equations (I) and (II).

[0087] The practical implementation of dynamic OBUF may be made by the storage of the array $NV[\beta']$ and the $LUT[\beta']$ of context indices, as well as the trees T^n 520. An alternative to the storage of the trees may be to store the array $k_n[\beta]$ 510 of the number (e.g., quantity) of non-masked bits.

[0088] A limitation for implementing dynamic OBUF may be its memory footprint. In some applications, a few million occupancy configurations may be practically handled, leading to about 20 bits β_i constituting an entry configuration β to the reduction function DR. Each bit β_i may correspond to the occupancy status of a neighboring cuboid of a current child cuboid or a set of neighboring cuboids of a current child cuboid.

[0089] Higher (e.g., more significant) bits β_i (e.g., β_0, β_1, etc.) may be the first bits to be unmasked. Higher (e.g., more significant) bits β_i (e.g., β_0, β_1, etc.) may be the first bits to be unmasked, for example, during the evolution of the dynamic reduction function DR. The order of neighbor-based information put in the bits β_i may impact the compression performance. Neighboring information may be ordered from higher (e.g., highest) priority to lower priority and put in this order into the bits β_i, from higher to lower weight. The priority may be, from the most important to the least important, occupancy of sets of adjacent neighboring child cuboids, then occupancy of adjacent neighboring child cuboids, then occupancy of adjacent neighboring parent cuboids, then occupancy of non-adjacent neighboring child nodes, and finally occupancy of non-adjacent neighboring parent nodes. Adjacent nodes sharing a face with the current child node may also have higher priority than adjacent nodes sharing an edge (but not sharing a face) with the current child node. Adjacent nodes sharing an edge with the current child node may have higher priority than adjacent nodes sharing only a vertex with the current child node.

[0090] FIG. 6 shows an example method for coding occupancy of a cuboid using dynamic OBUF. More specifically, FIG. 6 shows an example method for coding occupancy bit of a current child cuboid using dynamic OBUF. One or more steps of FIG. 6 may be performed by an

encoder and/or a decoder (e.g., the encoder 114 and/or decoder 120 in FIG. 1). All or portions of the flowchart may be implemented by a coder (e.g., the encoder 114 and/or decoder 120 in FIG. 1), an example computer system 2700 in FIG. 27, and/or an example computing device 2830 in FIG. 28.

[0091] At step 602, an occupancy configuration (e.g., occupancy configuration β) of the current child cuboid may be determined. The occupancy configuration (e.g., occupancy configuration β) of the current child cuboid may be determined, for example, based on occupancy bits of already-coded cuboids in a neighborhood of the current child cuboid. At step 604, the occupancy configuration (e.g., occupancy configuration β) may be dynamically reduced. The occupancy configuration may be dynamically reduced, for example, using a dynamic reduction function DR^n. For example, the occupancy configuration β may be dynamically reduced into a reduced occupancy configuration $\beta' = DR^n(\beta)$. At step 606, context index may be looked up, for example, in a look-up table (LUT). For example, the encoder and/or decoder may look up context index LUT[β'] in the LUT of the dynamic OBUF. At step 608, context (e.g., probability model) may be selected. For example, the context (e.g., probability model) pointed to by the context index may be selected. At step 610, occupancy of the current child cuboid may be entropy coded. For example, the occupancy bit of the current child cuboid may be entropy coded (e.g., arithmetic coded), for example, based on the context. The occupancy bit of the current child cuboid may be coded based on the occupancy bits of the already-coded cuboids neighboring the current child cuboid.

[0092] Although not shown in FIG. 6, the encoder and/or decoder may update the reduction function and/or update the context index. For example, the encoder and/or decoder may update the reduction function DR^n into DR^{n+1} and/or update the context index LUT[β'], for example, based on the occupancy bit of the current child cuboid. The method of FIG. 6 may be repeated for additional or all child cuboids of parent cuboids corresponding to nodes of the occupancy tree in a scan order, such as the scan order discussed herein with respect to FIG. 3.

[0093] In general, the occupancy tree is a lossless compression technique. The occupancy tree may be adapted to provide lossy compression, for example, by modifying the point cloud on the encoder side (e.g., down-sampling, removing points, moving points, etc.). The performance of the lossy compression may be weak. The lossy compression may be a useful lossless compression technique for dense point clouds.

[0094] One approach to lossy compression for point cloud geometry may be to set the maximum depth of the occupancy tree to not reach the smallest volume size of one voxel but instead to stop at a bigger volume size (e.g., NxNxN cuboids (e.g., cubes), where N > 1). The geometry of the points belonging to each occupied leaf node associated with the bigger volumes may then be modeled. This approach may be particularly suited for dense and smooth point clouds that may be locally modeled by smooth functions such as planes or polynomials. The coding cost may become the cost of the occupancy tree plus the cost of the local model in each of the occupied leaf nodes.

[0095] A scheme for modeling the geometry of the points belonging to each occupied leaf node associated with a volume size larger than one voxel may use sets of triangles as local models. The scheme may be referred to as the "TriSoup" scheme. TriSoup is short for "Triangle Soup" because the connectivity between triangles may not be part of the models. An occupied leaf node of an occupancy tree that corresponds to a cuboid with a volume greater than one voxel may be referred to as a TriSoup node. An edge belonging to at least one cuboid corresponding to a TriSoup node may be referred to as a TriSoup edge. A TriSoup node may comprise a presence flag (s_k) for each TriSoup edge of its corresponding occupied cuboid. A presence flag (s_k) of a TriSoup edge may indicate whether a TriSoup vertex (V_k) is present or not on the TriSoup edge. At most one TriSoup vertex (V_k) may be present on a TriSoup edge. For each vertex (V_k) present on a TriSoup edge of an occupied cuboid, the TriSoup node corresponding to the occupied cuboid may comprise a position (p_k) of the vertex (V_k) along the TriSoup edge.

[0096] In addition to the occupancy words of an occupancy tree, an encoder may entropy encode, for each TriSoup node of the occupancy tree, the TriSoup vertex presence flags and positions of each TriSoup edge belonging to TriSoup nodes of the occupancy tree. A decoder may similarly entropy decode the TriSoup vertex presence flags and positions of each TriSoup edge and vertex along a respective TriSoup edge belonging to a TriSoup node of the occupancy tree, in addition to the occupancy words of the occupancy tree.

[0097] FIG. 7 shows an example of an occupied cuboid (e.g., cube) 700. More specifically, FIG. 7 shows an example of an occupied cuboid (e.g., cube) 700 of size NxNxN (where N > 1) that corresponds to a TriSoup node of an occupancy tree. An occupied cuboid 700 may comprise edges (e.g., TriSoup edges 710 – 721). The TriSoup node, corresponding to the occupied cuboid 700, may comprise a presence flag (s_k) for each edge (e.g., each TriSoup edge of the TriSoup edges 710-721). For example, the presence flag of a TriSoup edge 714 may indicate that a TriSoup vertex V_1 is present on the TriSoup edge 714. The

presence flag of a TriSoup edge 715 may indicate that a TriSoup vertex V_2 is present on the TriSoup edge 715. The presence flag of a TriSoup edge 716 may indicate that a TriSoup vertex V_3 is present on the TriSoup edge 716. The presence flag of a TriSoup edge 717 may indicate that a TriSoup vertex V_4 is present on the TriSoup edge 717. The presence flags of the remaining TriSoup edges each may indicate that a TriSoup vertex is not present on their corresponding TriSoup edge. The TriSoup node, corresponding to the occupied cuboid 700, may comprise a position for each TriSoup vertex present along one of its TriSoup edges 710-721. More specifically, the TriSoup node, corresponding to the occupied cuboid 700, may comprise a position p_1 for TriSoup vertex V_1, a position p_2 for TriSoup vertex V_2, a position p_3 for TriSoup vertex V_3, and a position p_4 for TriSoup vertex V_4. The TriSoup vertices may be shared among TriSoup nodes along common TriSoup edge(s).

[0098] A presence flag (s_k) and, if the presence flag (s_k) may indicate the presence of a vertex, a position (p_k) of a current TriSoup edge may be entropy coded. The presence flag (s_k) and position (p_k) may be individually or collectively referred to as vertex information or TriSoup vertex information. A presence flag (s_k) and, if the presence flag (s_k) indicates the presence of a vertex, a position (p_k) of a current TriSoup edge may be entropy coded, for example, based on already-coded presence flags and positions, of present TriSoup vertices, of TriSoup edges that neighbor the current TriSoup edge. A presence flag (s_k) and, if the presence flag (s_k) may indicate the presence of a vertex, a position (p_k) of a current TriSoup edge (e.g., indicating a position of the vertex the edge is along) may be additionally or alternatively entropy coded. The presence flag (s_k) and the position (p_k) of a current TriSoup edge may be additionally or alternatively entropy coded, for example, based on occupancies of cuboids that neighbor the current TriSoup edge. Similar to the entropy coding of the occupancy bits of the occupancy tree, a configuration β_{TS} for a neighborhood (also referred to as a neighborhood configuration β_{TS}) of a current TriSoup edge may be obtained and dynamically reduced into a reduced configuration $\beta_{TS}' = DR^n(\beta_{TS})$, for example, by using a dynamic OBUF scheme for TriSoup. A context index $LUT[\beta_{TS}']$ may be obtained from the OBUF LUT. At least a part of the vertex information of the current TriSoup edge may be entropy coded using the context (e.g., probability model) pointed to by the context index.

[0099] The TriSoup vertex position (p_k) (if present) along its TriSoup edge may be binarized. The TriSoup vertex position (p_k) (if present) along its TriSoup edge may be binarized, for example, to use a binary entropy coder to entropy code at least part of the vertex

information of the current TriSoup edge. A number (e.g., quantity) of bits N_b may be set for the quantization of the TriSoup vertex position (p_k) along the TriSoup edge of length N. The TriSoup edge of length N may be uniformly divided into 2^{Nb} quantization intervals. By doing so, the TriSoup vertex position (p_k) may be represented by N_b bits (p_k^j, j=1, …, N_b) that may be individually coded by the dynamic OBUF scheme as well as the bit corresponding to the presence flag (s_k). The neighborhood configuration β_{TS}, the OBUF reduction function DR^n, and the context index may depend on the nature, characteristic, and/or property of the coded bit (e.g., a presence flag (s_k), a highest position bit (p_{k1}), a second highest position bit (p_{k2}), etc.) of the coded bit (e.g., presence flag (s_k), highest position bit (p_k^1), second highest position bit (p_k^2), etc.). There may practically be several dynamic OBUF schemes, each dedicated to a specific bit of information (e.g., presence flag (s_k) or position bit (p_k^j)) of the vertex information.

[0100] FIG. 8A shows an example cuboid 800 (e.g., a cube) corresponding to a TriSoup node. A cuboid 800 may correspond to a TriSoup node with a number K of TriSoup vertices V_k. Within cuboid 800, TriSoup triangles may be constructed from the TriSoup vertices V_k. TriSoup triangles may be constructed from the TriSoup vertices V_k, for example, if at least three ($K \geq 3$) TriSoup vertices are present on the TriSoup edges of cuboid 800. For example, with respect to FIG. 8A, four TriSoup vertices may be present and TriSoup triangles may be constructed. The TriSoup triangles may be constructed around the centroid vertex C defined as the mean of the TriSoup vertices V_k. A dominant direction may be determined, then vertices V_k may be ordered by turning around this direction, and the following K TriSoup triangles (listed as triples of vertices) may be constructed: V_1V_2C, V_2V_3C, …, V_KV_1C. The dominant direction may be chosen among the three directions respectively parallel to the axes of the 3D space to increase or maximize the 2D surface of the triangles, for example, if the triangles are projected along the dominant direction. By doing so, the dominant direction may be somewhat perpendicular to a local surface defined by the points of the point cloud belonging to the TriSoup node.

[0101] FIG. 8B shows an example refinement to the TriSoup model. The TriSoup model may be refined by coding a centroid residual value. A centroid residual value C_{res} may be coded into the bitstream. A centroid residual value C_{res} may be coded into the bitstream, for example, to use $C+C_{res}$ instead of C as a pivoting vertex for the triangles. By using $C+C_{res}$ as the pivoting vertex for the triangles, the vertex $C+C_{res}$ may be closer to the points of the point cloud than the centroid C, the reconstruction error may be lowered, leading to lower distortion at the cost of a small increase in bitrate needed for coding C_{res}.

[0102] The reconstruction of a decoded point cloud from a set of TriSoup triangles may be referred to as "voxelization" and may be performed, for example, by ray tracing or rasterization, for each triangle individually before duplicate voxels from the voxelized triangles are removed.

[0103] FIG. 9 shows an example of voxelization. Rays 900 may be launched parallel to one of the three axes of the 3D space, starting from integer coordinates P_{start}. The intersection P_{int}, if any, with a TriSoup triangle 901 belonging to a cube 902may be rounded to determine a decoded point. The cube 902 may correspond to a TriSoup node. The intersection may be found (e.g., determined), for example, using the Möller-Trumbore algorithm.

[0104] A presence flag (s_k) and, if the presence flag (s_k) indicates the presence of a vertex, a position (p_k) of the vertex along a current TriSoup edge may be entropy coded, for example, based on already-coded presence flags and positions (of present TriSoup vertices) of TriSoup edges that neighbor the current TriSoup edge. The presence flag (s_k) and position (p_k), individually or collectively, may be referred to as vertex information. A presence flag (s_k) and, if the presence flag (s_k) indicates the presence of a vertex, a position (p_k) on (e.g., indicating a position of the vertex along) a current TriSoup edge may be additionally or alternatively entropy coded, for example, based on occupancies of cuboids that neighbor the current TriSoup edge. Similar to the entropy coding of the occupancy bits of the occupancy tree, a configuration β_{TS} for a neighborhood (also referred to as a neighborhood configuration β_{TS}) of a current TriSoup edge may be determined and/or dynamically reduced into a reduced configuration $\beta_{TS}' = DR^n(\beta_{TS})$, for example, by using a dynamic OBUF scheme for TriSoup. A context index $LUT[\beta_{TS}']$ may be determined from the OBUF LUT. At least a part of the vertex information of the current TriSoup edge may be entropy coded, for example, using the context (or probability model) pointed to by the context index.

[0105] The TriSoup vertex position (p_k) (if present) along its TriSoup edge may be binarized. A binary entropy coder may entropy code at least part of the vertex information of the current TriSoup edge. A number of bits N_b may be set for the quantization of the TriSoup vertex position (p_k) along the TriSoup edge of length N that is uniformly divided into 2^{N_b} quantization intervals. The TriSoup vertex position (p_k) may be represented by N_b bits (p_k^j, $j=1,\ldots,N_b$), for example, if the number of bits N_b is set for the quantization of the TriSoup vertex position (p_k) along the TriSoup edge of length N that is uniformly divided into 2^{N_b} quantization intervals. The N_b bits, as well as the bit corresponding to the presence

flag (s_k), may be individually coded by the dynamic OBUF scheme. The neighborhood configuration β_{TS}, the OBUF reduction function DR^n, and/or the context index may depend on the nature/characteristic/property of the coded bit (e.g., presence flag (s_k), highest position bit (p_k^1), or second highest position bit (p_k^2)). Several dynamic OBUF schemes may be implemented, with each dedicated to a specific bit of information (e.g., presence flag (s_k) or position bit (p_k^j)) of the vertex information.

[0106] In video compression, performance may be improved by using inter frame prediction. Bitrates for compressing interframes may be one to two orders of magnitude lower than bitrates of intra frames, which may not use inter frame prediction. Point cloud data may behave differently from, for example, 2D video data. The 3D geometry, for point cloud data, may be coded by 3D point positions. Each point position of the 3D point positions may be associated with attributes (e.g., colors). Geometry and/or attributes may change between frames. Different 3D point positions and/or attributes associated with the corresponding 3D point positions may be coded, for example, for each frame. 2D video data may obtained, for example, by the projection of the 3D geometry and/or attributes onto a 2D plane having a fixed geometry (e.g., a camera sensor). For video coding, the attributes may be coded but the geometry may not be coded (and may not need to be coded). It may be expected that inter frame prediction between 3D point clouds may provide improved compression capability as compared to intra frame prediction (e.g., intra frame prediction alone) within a point cloud, even if, for example, 2D-projected attributes is expected to temporally have a higher correlation than the underlying 3D geometry. The octree may benefit from inter frame prediction and/or geometry compression gains. The general framework of inter frame prediction for 3D point clouds may be similar to the one of video compression.

[0107] FIG. 10 shows an example encoding method 1000. One or more steps of FIG. 10 may be performed by an encoder (e.g., the encoder 114). Encoding method 1000 may use inter frame prediction between different point cloud frames. A current frame 1001 (e.g., an image or a point cloud) may be coded relative to an already-coded reference frame 1010 (e.g., an image or a point cloud). At step 1020, a motion search may be performed from the already-coded reference frame 1010 toward the current frame 1001 to determine motion vectors 1021 that represents a motion flow between the two frames 1010 and 1001. In video compression, motion vectors may be 2-component (or 2D) vectors representing the motion from reference blocks of pixels to current blocks of pixels. In point cloud compression, motion vectors may be 3-component (or 3D) vectors representing the

motion from reference sets of 3D points (e.g., in a reference point cloud) to current sets of 3D points (e.g., in a current point cloud). At step 1025, motion vectors 1021 may be entropy coded into bitstream 1050. At step 1030, the reference frame 1010 may be motion compensated to determine a motion compensated frame 1031. Motion compensation may involve moving the pixels of the reference image based on the 2D (motion vectors, and/or moving the points of the reference point cloud based on the 3D motion vectors. The determined motion compensated frame may be "closer" to the current frame than the reference frame, for example, the color difference (or point distance) between the motion compensated frame 1031 and the current frame 1001 may be, on average, smaller than the color difference (or point distance) between the reference frame 1010 and the current frame 1001. At step 1040, inter frame prediction may be performed to determine inter residuals 1041. At step 1045, the inter residuals 1040 may be entropy coded into bitstream 1050. The inter residuals may carry more compressible information than the current frame that may or may not have undergone an intra prediction process. Therefore, the entropy coding performed at step 1045 may be more efficient for determining a bitstream 1050 with reduced size, compared to a bitstream determined by coding the current frame 1001 that has not benefited from inter frame prediction.

[0108] In video coding, inter residuals may be constructed as the difference of colors, pixel per pixel, between a current block of pixels belonging to the current frame (e.g., image) and a co-located compensated block of pixels belonging to the motion compensated frame (e.g., image). Inter residuals may be arrays of color differences that have typically small magnitude and may be efficiently compressed.

[0109] In point cloud compression, there may not be a "difference" between two sets of points, because there may not be a one-to-one mapping of the two sets of points. The concept of an inter residual may not be straightforwardly used (e.g., generalized) with respect to point clouds. For prediction of an octree representing a point cloud, the concept of inter residual may be replaced by conditional entropy coding, where conditional information for performing conditional entropy coding may be constructed based on a motion compensated point cloud. This may be extended to the framework of dynamic OBUF.

[0110] As described herein, the occupancy of a current volume (e.g., the current volume associated with a current node of an octree) may be provided by a quantity of occupancy bits, e.g., 8 occupancy bits. A current occupancy bit of the octree may be coded by an entropy coder selected by the output of a dynamic OBUF LUT of coder indices that takes a neighborhood configuration β as input. The neighborhood configuration β may be

constructed based on already-coded occupancy bits. The already-coded occupancy bits may be associated with neighboring volumes (e.g., associated with neighboring nodes of the current node). The construction of the neighborhood configuration β may be extended using inter frame information. An inter predictor occupancy bit may be indicated by (e.g., defined for) a current occupancy bit as a bit representative of the presence of at least one point of a motion compensated point cloud within the current volume. A strong correlation between the current occupancy bit and the inter predictor occupancy bit may exist, for example, if that motion compensation is efficient. This may be because the current and motion compensated point clouds are likely close to each other. Using the inter predictor occupancy bit as a bit of the neighborhood configuration β may lead to better compression performance of the octree (e.g., dividing the size of the octree bitstream by a factor two).

[0111] The motion field between octrees may comprise 3D motion vectors associated with 3D prediction units (PU). The PUs may have volumes that may include at least a part of one or several volumes (e.g., cuboids) associated with nodes of the octree. The motion compensation may be performed volume per volume (e.g., cuboid per cuboid) based on the 3D motion vectors for determining a motion compensated point cloud in one or more current volumes. The inter predictor occupancy bit may be determined, for example, based on the presence of at least one point of this motion compensated point cloud.

[0112] The TriSoup scheme may benefit from the motion compensated frame determined while the octree coding is performed, for example, before the TriSoup coding. Predictors of the presence and/or position of TriSoup vertices may be determined based on the motion compensated point cloud. The predictors may be determined, for example, based on the intersection of the compensated point cloud with the edges of the TriSoup nodes. Predictors of the centroid residual values may be determined.

[0113] The entropy coding of TriSoup vertices and/or centroid residual values may be performed, for example, by using these inter predictors. Inter predictors may constitute a part of a contextual information β_{inter} input of a dynamic OBUF instance that codes a TriSoup syntax element. Alternatively or additionally, a context may be selected based on inter predictors. The selected context may be used by an entropy coder (e.g., CABAC) to determine a probability for arithmetically entropy coding of a TriSoup syntax element.

[0114] Attributes associated with points of a point cloud may be coded, for example, after the coding of the underlying geometry (e.g., the positions of the points in the 3D space) has been performed. If the geometry coding is a lossless coding (e.g., by using an octree scheme), the encoder may have direct access to the attribute values associated with the

coded points. The coded geometry may differ from the original geometry, for example, if the geometry coding is a lossy coding (e.g., by using a TriSoup scheme). The original attributes may be mapped by the encoder from the original geometry to the coded geometry for determining mapped attributes on the coded geometry.

[0115] As described herein, attributes may indicate a property of a point's visual appearance (e.g., texture, color, material, transparency, reflectance, time stamp, or velocity). For attributes that are colors, the attribute mapping performed by the encoder may be referred to as a recoloring process. This may be because the colors of the original geometry may be used to color (e.g., recolor) the coded geometry (e.g., a reconstructed geometry).

[0116] The coded geometry associated with the mapped attributes for the coded geometry may comprise a point cloud representative of the original point cloud in geometry and attributes. There may be more than one (e.g., two) attribute coding schemes that may be used and/or selected for coding attributes associated with the coded geometry. The attribute coding schemes may comprise, for example, the prediction with lifting transform ("pred-lift") scheme, and/or the region-adaptive hierarchical transform ("RAHT") scheme. The attribute coding schemes may be used, for example, in G-PCC.

[0117] The pred-lift scheme first may perform a decomposition of the coded geometry into Levels of Details (also known as LoD). For a set (S) of points (e.g., all points) of the coded geometry, the set may be decomposed into disjoint subsets S^i such that $S = \bigcup_{i=0}^{L-1} S^i$. L levels of details may be defined as a tower of point cloud geometries

$$S^0 \subset S^0 \cup S^1 \subset \cdots \subset S^0 \cup \ldots \cup S^{L-1} = S$$

where the set S^0 of points may be the first (e.g., coarsest) level of details, and the set $S^0 \cup \ldots \cup S^{L-1}$ of points may be the L^{th} (e.g., finest) level of details, for example, if the set may be decomposed into disjoint subsets S^i such that $S = \bigcup_{i=0}^{L-1} S^i$.

[0118] Attributes a_j may be associated with the points s_j of the set S of all points of the coded geometry. Considering subsets $a^0, \ldots, a^{L-1}$ of attributes, attributes a^i_j of a subset a^i may be associated with the points s^i_j of the subset S^i. The i^{th} level of details ($S^0 \cup \ldots \cup S^{i-1}$) may have associated attributes comprising the concatenation of attributes of subsets $a^0, \ldots, a^{i-1}$. The set 'a' of attributes (e.g., all attributes) may be partitioned into subsets $a^0, \ldots, a^{L-1}$.

[0119] FIGs. 11, 12, 13, and 14 show examples for coding (e.g., encoding or decoding) point cloud attributes. The coding may be based on intra transform schemes such as a prediction transform scheme or a pred-lift scheme. The prediction transform scheme may be a variation of pred-lift scheme without update operations. The intra transform schemes may be examples of wavelet transforms that may convert (e.g., transform) attribute values into

wavelet coefficients, which may be more efficiently compressed than the original attribute values. The wavelet coefficients may represent values of residual attributes, which may be smaller and more efficiently compressed than the values of the original attributes. The residual attributes may be referred to as and/or comprise wavelet coefficients (or transform/transformed coefficients). The wavelet coefficients may result from application of the prediction transform scheme and/or pred-lift scheme.

[0120] As described herein, the prediction transform scheme and/or pred-lift scheme may operate using prediction for and/or between levels of details of attributes. At the encoder, attributes at a higher (e.g., finer) level of detail may be predicted based on attributes at a lower (e.g., coarser) level of detail. Each level of detail starting from the highest level may be successively predicted, for example, based on lower level(s) of detail. The decoder may perform inverse operations such that attributes at a lower level of detail may be predicted and/or reconstructed, for example, based on residual attributes of higher level(s) of detail. Each level of detail starting from the lowest level may be successively predicted and reconstructed, for example, based on higher level(s) of detail. Although the examples shown in FIGS. 11-14 show three levels of details (LoDs), it is appreciated that a different number of LoDs may be used, for example, by extending (e.g., iteratively) the decomposition scheme.

[0121] FIG. 11 shows an example method for encoding point cloud attributes. The encoding may be based on a prediction transform scheme. One or more steps of FIG. 11 may be performed by an encoder (e.g., the encoder 114 in FIG. 1).

[0122] A set 'a' of attributes may be coded, for example, using prediction for and between one or more (e.g., three) levels of details, from a highest (e.g., first predicted) level of details (e.g., associated with highest frequencies such as res a^2) to a lowest (e.g., third) level of details (e.g., associated with lowest frequencies such as a^0). At step 1110, an encoder may split a set 'a' of attributes into a first set of attributes comprising the attributes of the subset a^2 and a second set of attributes comprising the attributes of the two subsets a^0 and a^1. At step 1120, the encoder may determine predictive values of the attributes of the first set of attributes (a^2) from the attributes of the second set of attributes (a^0 and a^1). At step 1130, the encoder may determine first residual values 'res a^2'. The encoder may determine first residual values 'res a^2', for example, by subtracting the predictive values from the attributes of the first set of attributes (a^2). At step 1170, the encoder may encode the first residual values 'res a^2' into the bitstream. The operations at steps 1120-1130 may be iteratively used for (e.g., applied to) each successively lower (e.g., coarser) LoD.

[0123] At step 1140, the encoder may split the second set of attributes (a^0 and a^1) into a third set of attributes and a fourth set of attributes. The third set of attributes may comprise the attributes of the subsets a^1. The fourth set of attributes may comprise the attributes of the subset a^0. At step 1150, the encoder may determine predictive values of the attributes of the third set of attributes (a^1) from the attributes of the fourth set of attributes (a^0). At step 1160, the encoder may determine second residual values 'res a^1'. The encoder may determine second residual values 'res a^1', for example, by subtracting predictive values from the attributes of the third set (a^1). At step 1170, the encoder may encode the second residual values 'res a^1' into the bitstream. The encoder may encode the attributes of the fourth set of attributes (a^0) into the bitstream.

[0124] The bitstream may comprise data representative of first residual values 'res a^2', second residual values 'res a^1', and/or the attributes of the subset a^0 (e.g., the fourth set of attributes). At step 1170, the residual values may be entropy coded. The encoder may encode the attributes of the subset a^0 into the bitstream. Additionally or alternatively, the encoder may perform intra prediction of a current attribute a^0_j of the subset a^0 to be coded. The encoder may perform intra prediction of a current attribute a^0_j of the subset a^0 to be coded, for example, based on already-coded attributes of the subset a^0. This may improve the compression efficiency of the attributes of the subset a^0.

[0125] The encoder may quantize the attributes of the subset a^0, the first residual value 'res a^2', and/or the second residual value 'res a^1', for example, if lossy attribute coding is allowed. The encoder may encode (e.g., entropy encode), into the bitstream, the attributes of the subset a^0 (or the quantized attributes of the subset a^0), the first residual value 'res a^2' (or the quantized first residual values 'res a^2'), and/or the second residual value 'res a^1' (or the quantized second residual value 'res a^1').

[0126] FIG. 12 shows an example method for decoding point cloud attributes. The decoding may be based on a prediction transform scheme. One or more steps of FIG. 12 may be performed by a decoder (e.g., the decoder 120 in FIG. 1). A set 'a' of attributes may be decoded. The set of 'a' attributes may be encoded, for example, as described herein with respect to FIG. 11. The decoding may use prediction between one or more (e.g., three) levels of details, from a lowest (e.g., third) level of details to a highest (e.g., first) level of details (e.g., in reverse order of the encoder described herein with respect to FIG. 11).

[0127] At step 1210, the decoder may decode the first residual values 'res a^2', the second residual values 'res a^1', and/or attributes of a fourth set of attributes (a^0) from the bitstream. The decoder may use (e.g., apply) dequantization (e.g., for lossy compression). At step 1220,

the decoder may determine predictive values of the attributes of a third set of attributes (a^1) from the decoded attributes of the fourth set of attributes (a^0). The decoder may determine the predictive values, for example, in a way similar to that described with respect to step 1150 of FIG. 11. At step 1230, the decoder may determine decoded attributes of the third set of attributes (a^1), for example, by adding the predictive values to the decoded first residual values 'res a^1'. The operations at steps 1120-1230 may be iteratively used for (e.g., applied to) each successively higher (e.g., finer) LoD.

[0128] At step 1240, the decoder may determine a second set of attributes (a^0 and a^1). The decoder may determine a second set of attributes (a^0 and a^1), for example, by merging the third set of attributes (a^1) and the fourth set of attributes (a^0). Step 1240 may be an inverse of step 1140. At step 1250, the decoder may determine predictive values of the attributes of a first set of attributes (a^2) from the attributes of the second set of attributes (a^0 and a^1). The decoder may determine predictive values in a way similar to that described with respect to step 1120. At step 1260, the decoder may determine decoded attributes of the first set of attributes (a^2), for example, by adding the predictive values to the decoded second residual values 'res a^2'. At step 1270, the decoder may determine the set 'a' of decoded attributes for the coded geometry S (e.g., the whole coded geometry S), for example, by merging the first set of attributes (a^2) and the second set of attributes (a^0 and a^1). Step 1270 may be an inverse of step 1110.

[0129] FIG. 13 shows an example method for encoding point cloud attributes. The encoding may be based on a pred-lift transform scheme with prediction and update. One or more steps of FIG. 13 may be performed by an encoder (e.g., the encoder 114 in FIG. 1). A set 'a' of attributes may be encoded using prediction between one or more (e.g., three) levels of details, from a highest (e.g., first) level of details (e.g., associated with highest frequencies such as res a^2) to a lowest (e.g., third) level of details (e.g., associated with lowest frequencies such as up up a^0).

[0130] At step 1310, an encoder may split a set 'a' of attributes into a first set of attributes comprising the attributes of the subset a^2 and a second set of attributes comprising the attributes of the two subsets a^0 and a^1. Step 1310 may be performed similar to that described herein with respect to step 1110 of FIG. 11. At step 1320, the encoder may determine predictive values of the attributes of the first set of attributes (a^2) from the attributes of the second set of attributes (a^0 and a^1). Step 1320 may be performed similar to that described herein with respect to step 1120. At step 1330, the encoder may determine first residual values 'res a^2', for example, by subtracting predictive values from

the attributes of the first set of attributes (a^2). Step 1320 may be performed similar to that described herein with respect to step 1130. At step 1370, the encoder may encode the first residual values 'res a^2' into the bitstream. Step 1370 may be performed similar to that described herein with respect to step 1170.

[0131] At step 1375, the encoder may determine update attribute values from the first residual values 'res a^2'. The encoder may determine an update attribute value, for example, based on a first residual value. The update attribute value may be determined as the first residual value multiplied by a scaling factor (e.g., ½, ¼, 1/8) that may be predetermined or signaled in the bitstream. At step 1380, the encoder may update attribute values 'up a^0' and 'up a^1' of the second set of attributes (a^0 and a^1) by adding the update attribute values to the attribute values of the second set of attributes (a^0 and a^1). The operations at steps 1320, 1330, 1375, and 1380 may be iteratively used for (e.g., applied to) each successively lower (e.g., coarser) LoD.

[0132] At step 1340, the encoder may split the second set of attributes (a^0 and a^1) into a third set of attributes and a fourth set of attributes. The third set of attributes may comprise the updated attribute values 'up a^1' of the subset a^1. The fourth set of attributes may comprise the updated attribute values 'up a^0' of the subset a^0. At step 1350, the encoder may determine predictive values of the updated attribute values 'up a^1' of the third set of attributes (a^1) from the updated attribute values 'up a^0' of the fourth set of attributes (a^0). At step 1360, the encoder may determine third residual values 'res up a^1', for example, by subtracting predictive values from the updated attribute values 'up a^1' of the third set of attributes (a^1). At step 1370, the encoder may encode the third residual values 'res up a^1' into the bitstream. At step 1385, the encoder may determine update attribute values from the third residual values 'res up a^1'. At step 1390, the encoder may determine further updated attribute values 'up up a^0' of the fourth set of attributes (a^0), for example, by adding the update attribute values to the updated attribute values 'up a^0'of the fourth set of attributes (a^0). At step 1370, the encoder may encode the further updated attribute values 'up up a^0' of the fourth set of attributes (a^0) into the bitstream.

[0133] The bitstream may comprise data representative of transformed attributes (e.g., data representative of the first residual values 'res a^2', third residual values 'res up a^1' and/or further updated attribute values 'up up a^0' of the fourth set of attributes). The encoder may encode the further updated attribute values 'up up a^0' of the fourth set of attributes into the bitstream. The encoder may perform intra prediction of a current further updated attribute values 'up up a^0_j' to be coded, for example, based on already-coded attributes of

the fourth set of attributes. This may improve the compression efficiency of the further updated attribute values 'up up a^0' of the fourth set of attributes.

[0134] The encoder may quantize the further updated attribute values 'up up a^0' of the fourth set of attributes, the first residual value 'res a^2' and/or the third residual value 'res up a^1', for example, if lossy attribute coding is allowed. The encoder may entropy encode into the bitstream the further updated attribute values 'up up a^0' of the fourth set of attributes, the first residual value 'res a^2' (or the quantized first residual values 'res a^2'), and/or the third residual value 'res up a^1' (or the quantized third residual value 'res up a^1').

[0135] FIG. 14 shows an example method for decoding point cloud attributes. The decoding may be based on a pred-lift transform scheme with prediction and update. One or more steps of FIG. 14 may be performed by a decoder (e.g., the decoder 120 in FIG. 1). A set of 'a' attributes may be encoded as described herein with respect to Fig. 13. The set 'a' of attributes (e.g., encoded set 'a' of attributes) may be decoded. The decoding may use prediction between one or more (e.g., three) levels of details, from a lowest (e.g., third) level of details to a highest (e.g., first) level of details (in reverse order of the encoder described herein with respect to FIG. 13).

[0136] At step 1410, the decoder may decode the first residual values 'res a^2', the third residual values 'res up a^1', and/or further updated attribute values 'up up a^0' of a fourth set of attributes (a^0) from the bitstream. The decoder may use (e.g., apply) an optional dequantization for lossy compression. At step 1475, the decoder may determine update attribute values from the decoded third residual values 'res up a^1'. Step 1475 may be performed in a manner similar to that described herein with respect to step 1385. The decoder may determine an update attribute value based on the third residual values. The update attribute value may be determined as the third residual values multiplied by a scaling factor (e.g., ½, ¼, 1/8) that may be predetermined or signaled in the bitstream.

[0137] At step 1480, the decoder may determine updated attribute values 'up a^0' of the fourth set of attributes (a^0), for example, by subtracting the update attribute values from the decoded further updated attribute values 'up up a^0' of the fourth set of attributes (a^0). At step 1420, the decoder may determine predictive values of the updated attribute values 'up a^1' of a third set of attributes (a^1) from the updated attribute values 'up a^0' of the fourth set of attributes (a^0). Step 1420 may be performed in a similar manner as described herein with respect to step 1350. At step 1430, the decoder may determine updated attribute values 'up a^1' of the third set of attributes (a^1), for example, by adding the predictive values to the decoded third residual values 'res up a^1'. At step 1440, the decoder may determine a

second set of attributes (a^0 and a^1), for example, by merging the third set of attributes (a^1) and the fourth set of attributes (a^0). Step 1440 may be an inverse of step 1340. The second set of attributes (a^0 and a^1) may comprise updated attribute values 'up a^0' and updated attribute values 'up a^1'. At step 1485, the decoder may determine update attribute values from the decoded first residual values 'res a^2'. Step 1485 may be performed in a manner similar to that as described herein with respect to step 1375. The operations at steps 1420, 1430, 1440, 1475, and 1480 may be iteratively used for (e.g., applied to) each successively higher (e.g., finer) LoD.

[0138] At step 1490, the decoder may determine attribute values 'a^0' and attribute values 'a^1' of the second set of attributes (a^0 and a^1), for example, by subtracting the update attribute values from the updated attribute values 'up a^0' and from updated attribute values 'up a1' of the second set of attributes (a^0 and a^1). At step 1450, the decoder may determine predictive values of the attributes of a first set of attributes (a^2) from the attributes of the second set of attributes (a^0 and a^1). Step 1450 may be performed in a manner similar to that described herein with respect to step 1320. At step 1460, the decoder may determine decoded attributes of the first set of attributes (a^2), for example, by adding the predictive values to the decoded second residual values 'res a^2'. At step 1470, the decoder may determine the set 'a' of decoded attributes for the coded geometry S (e.g., the whole coded geometry), for example, by merging the first set of attributes (a^2) and the second set of attributes (a^0 and a^1). Step 1470 may be an inverse of step 1310.

[0139] Pred-lift schemes may be similar to the lifting scheme used for (e.g., applied to) wavelets in image coding. A pre-lift scheme may comprise update steps that are not in prediction transform scheme (e.g., in addition to the prediction steps in the prediction transform scheme). The update steps may provide better compression performance (e.g., in combination with the prediction steps). This may compact the energy in the lowest level of details, which may reduce distortion with lossy coding.

[0140] Attributes may be coded using the RAHT scheme. The RAHT scheme may be based on the iterative use of a two-point transform. In point cloud attribute coding, the two-point RAHT transform may be used for (e.g., applied to) two sets (A_1 and A_2) of attributes. Each of A_1 and A_2 may have respectively w_1 and w_2 number of attributes. Each of A_1 and A_2 may have respective associated coefficients c_{A1} and c_{A2}. Each of c_{A1} and c_{A2} is representative of the sum of attribute values over the corresponding set divided by the square root of the number of attributes.

$$c_{Ai} = \frac{1}{\sqrt{w_i}} \sum_{a \in A_i} a. \qquad (w_i = \#A_i) \qquad (*)$$

[0141] The two-point RAHT transform may depend on the weights w_1 and w_2 . The two-point RAHT transform may be defined by a 2x2 matrix as follows

$$\text{RAHT}(w_1, w_2) = \frac{1}{\sqrt{w_1+w_2}}\begin{bmatrix}\sqrt{w_1} & \sqrt{w_2}\\ -\sqrt{w_2} & \sqrt{w_1}\end{bmatrix}.$$

[0142] Two new coefficients DC and AC may be determined, for example, if used for (e.g., applied to) the two coefficients c_{A1} and c_{A2}.

$$\begin{bmatrix}DC\\ AC\end{bmatrix} = \text{RAHT}(w_1, w_2)\begin{bmatrix}c_{A1}\\ c_{A2}\end{bmatrix}$$

[0143] As described herein, the above property (*) on coefficients may hold for the DC coefficient.

$$DC = \frac{1}{\sqrt{w_1+w_2}}\left(\sqrt{w_1}c_{A1} + \sqrt{w_2}c_{A2}\right) = \frac{1}{\sqrt{w_1+w_2}}\left(\sum_{a\in A_1} a + \sum_{a\in A_2} a\right) = \frac{1}{\sqrt{w_1+w_2}}\sum_{a\in A_1\cup A_2} a$$

$$= c_{A1\cup A2}$$

[0144] The two point RAHT transform may be iteratively used for (e.g., applied to) DC coefficients. This may be referred to as the RAHT iterative method. AC coefficients may not undergo further transformation, for example, after being determined. At the start of the RAHT iterative method, there may be as many initial sets A_i of attributes as there are points in the coded geometry S. Each initial set A_i of attributes may contain one attribute (w_i=1).The coefficient c_{Ai} may be equal to the value of the one attribute, and/or may fulfill the property (*). By induction, the property (*) may hold for subsequent DC coefficients (e.g., all subsequent DC coefficients) determined, for example, after iterative application of the two point RAHT transform.

[0145] At a particular stage of the RAHT iterative method, determined coefficients may be the union of a set of DC coefficients fulfilling the property (*) and a set of AC coefficients. The RAHT iterative method may continue until DC coefficients are depleted and only one DC coefficient may be left. The one DC coefficient may be equal to C_A where A may be the set of attributes (e.g., all attributes) associated with the coded geometry S (e.g., the complete coded geometry S). The RAHT iterative method may follow an order among pairs of DC coefficients.

[0146] The two-point inverse RAHT transform may be defined by a 2x2 matrix as follows

$$\text{iRAHT}(w_1, w_2) = \frac{1}{\sqrt{w_1+w_2}}\begin{bmatrix}\sqrt{w_1} & -\sqrt{w_2}\\ \sqrt{w_2} & \sqrt{w_1}\end{bmatrix}$$

The two-point inverse RAHT transform may be used with respect to (e.g., applied to) DC and AC coefficients for obtaining back the two coefficients c_{A1} and c_{A2}.

$$\begin{bmatrix} c_{A1} \\ c_{A2} \end{bmatrix} = \text{iRAHT}(w_1, w_2) \begin{bmatrix} DC \\ AC \end{bmatrix}$$

[0147] The inverse iterative RAHT method may use (e.g., apply) the inverse two-point RAHT to DC and AC coefficient in reverse order relative to their obtention by the iterative RAHT method. At the end of the inverse iterative RAHT method, coefficients c_{Ai} associated with the initial sets A_i of attributes may be obtained. These coefficients c_{Ai} may be equal to the values of the one attribute associated with the initial sets A_i.

[0148] For lossy RAHT compression of attributes, coefficients may be further compressed based on using (e.g., applying) a quantization to the DC and AC coefficients, for example, before encoding in the bitstream. The decoder may use (e.g., apply) a dequantization, for example, after decoding of the quantized DC and AC coefficients from the bitstream.

[0149] The RAHT iterative method may follow an octree as a specific iterative order, for example, in G-PCC. One or more (e.g., up to eight) DC coefficients, associated with one or more (e.g., up to eight) occupied child nodes of a parent node in the octree, may undergo a cascade of two-point RAHT transformations until one DC coefficient remains, together with the remaining (e.g., up to seven) AC coefficients. This one DC coefficient may be pushed at parent node level. The method may be repeated at upper octree depth, for example, until the root node is reached.

[0150] FIG. 15 shows an example RAHT transformation. The RAHT transformation may be applied on child nodes of an octree parent node along three successive directions. The parent node 1500 may have multiple (e.g., five) occupied child nodes each having a respective associated coefficient c_i and weight w_i. A first RAHT transformation 1510 may be performed along a first direction 1511. If there are two adjacent occupied child nodes 1513 along the first direction, the two adjacent occupied child nodes 1513 may undergo a two-point RAHT transform. A new DC coefficient 1514 and an AC coefficient 1515 may be determined and pushed to a set 1550 of AC coefficients. The node may be left as is and its DC coefficient may be kept 1517, for example, if there is only one occupied child node 1516 along the first direction. In this example, the child nodes may be collapsed along the first direction to determine a new set 1519 of nodes (e.g., a set of three nodes), with associated new DC coefficients, may be determined.

[0151] A second RAHT transformation 1520 may be performed, for example, after the first RAHT transformation. The second RAHT transformation may be along a second direction 1521. The second RAHT transformation may be performed similarly to the first RAHT transformation. The child nodes 1522 may be determined. The child nodes 1522 may have been collapsed along the first two directions 1511 and 1521. The AC coefficients 1523

may be pushed to the set 1550 of AC coefficients. A third RAHT transformation 1530 may be performed, for example, after the second RAHT transformation. The third RAHT transformation may be along a third direction 1531. The third RAHT transformation may be performed similarly to the first and/or second RAHT transformation. A (e.g., unique) child node 1532 may be determined. The child node 1532 may result from the collapse along all three directions. The AC coefficients 1533 may be pushed to the set 1550 of AC coefficients. The child node 1532 may have an associated DC coefficient that is pushed to the parent node (e.g., as shown for example in FIG. 16).

[0152] FIG. 16 shows an example RAHT transformation. The RAHT transformation may be used for (e.g., applied to) octree nodes (e.g., all octree nodes) at depth 'd' to determine DC coefficients at depth d-1 and AC coefficients. Occupied nodes 1600 of an octree may be at depth d. Occupied nodes 1600 may undergo a RAHT transformation along the three directions. DC coefficients for each node may be pushed up to the corresponding occupied parent nodes 1610 that belong to the octree at depth d-1. The three DC coefficients of the child nodes 1601 may, for example, undergo a RAHT transformation along the three directions. A unique DC coefficient associated with the parent node 1611 may be determined and two AC coefficients 1621 may be pushed to a set 1620 of AC coefficients. By performing this method for (e.g., all) occupied nodes 1600 of the octree at depth d, the DC coefficients associated with occupied nodes of the octree at depth d may be transformed into DC coefficients associated with occupied nodes 1610 of the octree at depth d-1 and a set 1620 of AC coefficients.

[0153] This bottom-up method may be repeated depth per depth, for example, until reaching the minimum depth (the root node). The result of the RAHT transformation over the octree (e.g., complete octree) may be a set of coefficients comprising a unique DC coefficient and a set of (many) AC coefficients. The RAHT transformation method may start from the highest depth where occupied child nodes correspond to a unique point (voxel) of the coded point cloud S. The unique point may be associated with a unique attribute among the set 'a' of attributes. The DC coefficient at highest depth may be set as the value of the unique attribute associated with each occupied node. The weights 'w' may be set to 1.

[0154] The inverse RAHT method on an octree may be a top-down method, from the root node down to the last depth made of leaf nodes that each contain one point of the point cloud, and one associated attribute. The DC coefficients of occupied nodes 1610 of the octree at depth d-1 may be inverse transformed into DC coefficients of occupied nodes 1600 of the octree at depth d, for example, by using (e.g., applying) the inverse two-point RAHT

transform to the DC coefficient of each of the occupied node of the octree at depth d-1 and to the related AC coefficients from set 1620 of AC coefficients. The inverse two-point RAHT transform may be applied along the three directions, in reverse order to invert the node transformation as described herein with respect to FIG. 15. DC coefficients of the leaf nodes may be obtained, and the values of the DC coefficients may correspond to the attributes associated with the unique point of each of the leaf nodes. Like geometry coding of a point cloud, coding of attributes associated with the points of a current point cloud may benefit from inter frame prediction using a motion compensated point cloud. The motion compensated point cloud may inherit attributes from a reference point cloud that has been motion compensated. If, for example, motion occurs, points may keep their associated attributes. The motion compensated attributes (e.g., the attributes associated with the points of the motion compensated point cloud) may be used. This may improve compression of the attributes of the coded geometry of the current point cloud.

[0155] Inter pred-lift scheme may use motion compensated attributes or residual attributes based on differences between attributes and motion-compensated attributes. Using residual attributes may increase compression. Inter pred-lift scheme may be used in the pred-lift scheme, for example, by plugging the inter pred-lift scheme to the prediction steps 1120, 1150, 1220, 1250, 1320, 1350, 1420 and/or 1450. Attributes (or residual attributes) a^i of the set S^i of points may be predicted, for example, by attributes (or residual attributes) of the subsets a^0, ..., a^{i-1} of the lower level of details $S^0 \cup ... \cup S^{i-1}$. Additionally or alternatively, attributes (or residual attributes) a^i of the set S^i of points may be predicted by motion compensated attributes (or residual attributes) of a set a^{inter} associated with points of a motion compensated point cloud S^{inter}. The prediction step may be performed, for example, based on attributes (or residual attributes) of subsets a^0, ..., a^{i-1} and of a set a^{inter} of an augmented lower level of details $S^0 \cup ... \cup S^{i-1} \cup S^{inter}$.

[0156] The encoder and/or decoder may determine predictive values of the attributes (or residual attributes) of the fourth set of attributes (or residual attributes) (subset a^0 of the coarsest level of details S^0) from the motion compensated attributes (or residual attributes) of the set a^{inter} associated with the points of a motion compensated point cloud S^{inter}. The encoder and/or decoder may subtract the predictive values from the attributes (or residual attributes) of the fourth set of attributes (or residual attributes) to determine residual values 'res a^0'. The encoder may encode the residual values 'res a^0' (or residual of residual attributes) into the bitstream instead of the attributes (or residual attributes) of the fourth set of attributes.

[0157] Inter RAHT scheme may use inter prediction for predicting the values of the DC and the AC coefficients determined by the RAHT iterative method. It may be beneficial to maintain a common attribute octree structure for both a current point cloud S^{coded} to be coded and a motion compensated point cloud S^{inter}, for example, because the generation of DC and AC coefficients follows an octree. A common bounding box encompassing both point clouds may be determined. An octree partitioning may be performed, from a root node associated with the common bounding box, for both point clouds. This may lead to two octree partitioning that may be differently, for example, if the point clouds are not equal. The two octrees may have a common subtree starting from the root node. On the common subtree, occupied node topology may be the same, and/or a common set of DC and AC coefficients may be determined for both point clouds. The subset of DC and AC coefficients, associated with nodes of the common subtree and determined from the attributes of the current point cloud S^{coded}, may be predicted from DC and AC coefficients determined from the attributes of the motion compensated point cloud S^{inter}. The encoder and/or decoder may determine coefficient residual values, for example, by subtracting the DC and AC coefficients, determined from the attributes of the motion compensated point cloud S^{inter}, from the DC and AC coefficients associated with nodes of the common subtree and determined from the attributes of the current point cloud S^{coded}. The encoder may encode, into the bitstream, the coefficient residual values and/or may not encode, into the bitstream, the DC and AC coefficients associated with nodes of the common subtree and determined from the attributes of the current point cloud S^{coded}.

[0158] The DC and AC coefficients that are not associated with nodes of the common subtree may not be predicted, and/or may be coded directly in a similar way as performed without inter prediction. Additionally or alternatively, predicted DC coefficients of the current point cloud S^{coded} may be determined at some depth (e.g., without predicting AC coefficients at the same depth). This may be based on the assumption that both the octree of the current point cloud S^{coded} and the octree of the motion compensated point cloud S^{inter} have a same occupancy of a node at this depth. The predicted DC coefficients may be determined from the corresponding co-located DC coefficients of the motion compensated point cloud S^{inter}. DC residual values may be determined, for example, by subtracting the predicted DC coefficients from the DC coefficients of the current point cloud S^{coded}. The RAHT transformation may go up in the octree starting from DC residual values replacing the DC coefficients of the coded current point cloud, for example, after the predicted DC coefficients be determined.

[0159] A RAHT scheme that does not use information from a reference frame different from the current frame may be called an intra RAHT scheme. Intra prediction may be performed between DC and AC coefficients of an intra RAHT scheme. Inter-depth prediction within a current frame may have been integrated into, for example, the RAHT scheme of GPCC. The inter-depth prediction mechanism may predict the DC coefficients associated with nodes of the octree at depth d, for example, by using interpolation of DC coefficients associated with nodes of the octree at lower depth d-1.

[0160] FIG. 17 shows an example method for encoding a current point cloud frame. More specifically, FIG. 17 shows an example method for encoding geometry and attributes of a current point cloud frame. The encoding may use a motion-compensated point cloud frame determined from a motion field selected (e.g., to optimize or reduce geometry differences) for geometry coding. One or more steps of the example method of FIG. 17 (e.g., method 1700) may be performed and/or implemented by an encoder (e.g., encoder 114 of FIG. 1).

[0161] A current point cloud frame 1710 of a sequence of point cloud frames of a dynamic point cloud may be obtained, for example, by an encoder. An already-coded reference point cloud frame 1705 may be also obtained, for example, by the encoder. The already-coded reference point cloud frame may be obtained, for example, from among a plurality of already-decoded reference point cloud frames. At step 1720, an encoder may determine geometry motion vectors (MV) 1721. The geometry motion vectors (MV) 1721 may be determined, for example, by performing a geometry motion search. For example, geometry MV 1721 may approximate the 3D motion field from the reference point cloud frame 1705 to the geometry of the current point cloud frame 1710. The geometry MV 1721 may be selected to best approximate the 3D motion field. The geometry MV 1721 may be selected to best approximate the 3D motion field, for example, based on a cost minimization function. The cost minimization function may minimize differences between a geometry of the reference point cloud frame and a geometry of the current point cloud frame. The geometry of the reference point cloud frame may be adjusted by a geometry MV of the motion-compensated point cloud frame (e.g., using geometry MV 1721). The encoder may encode geometry MV information 1722 into bitstream 1770, for example, as a representation of the geometry MV 1721. The geometry MV information may indicate the already-coded reference point cloud frame 1705 among a plurality of already-coded reference point cloud frames.

[0162] At step 1730, the encoder may determine a geometry motion-compensated point cloud frame 1731. The geometry motion-compensated point cloud frame 1731 may be determined, for example, by performing a motion compensation of the reference point cloud frame 1705 based on the geometry MV 1721. At step 1740, the encoder may encode the geometry of the current point cloud frame 1710. The geometry of the current point cloud frame 1710 may be encoded, for example, based on the geometry MV 1721. The geometry of the current point cloud frame 1710 may be encoded, for example, based on the geometry motion-compensated point cloud frame 1731. The geometry of the current point cloud frame 1710 may be encoded, based on the geometry motion-compensated point cloud frame 1731, as geometry information 1742 into the bitstream 1770. The geometry of the current point cloud frame 1710 may be encoded as geometry information 1742 into the bitstream 1770.

[0163] The encoder may provide a decoded geometry 1741 (e.g., a reconstructed geometry) of the current point cloud frame 1710. The encoder may code (e.g., decode) the encoded geometry, for example, to determine a reconstructed geometry corresponding to a decoded geometry at the decoder. The decoded geometry 1741 and the geometry of the current point cloud frame 1710 may differ significantly, for example, if the geometry compression is lossy. At step 1750, the encoder may determine mapped attributes 1751 associated with the decoded geometry 1741. The mapped attributes may be determined, for example, by mapping the attributes associated with the geometry of the current point cloud frame 1710 to the decoded geometry 1741. Attributes may comprise colors. The mapped attributes may be determined, for example, based on recoloring the attributes.

[0164] Mapped attributes may be determined, for example, based on a k nearest neighbor (KNN) search algorithm to determine nearest points from the geometry of the current point cloud frame 1710 to the decoded geometry 1741. The k nearest neighbor (KNN) search algorithm may include, for example, using a space partitioning algorithm such as a KD Tree search, a Ball/metric Tree search, a Brute force search, etc.). A mapped attribute of a point of the decoded geometry 1741 may be, for example, the average attribute values. The average attribute values may be associated with the nearest points of the current point cloud frame 1710 relative to the point of the decoded geometry.

[0165] The decoded geometry 1741 (e.g., reconstructed geometry) and the geometry of the current point cloud frame 1710 may be the same, for example, if the geometry compression is lossless. The attribute mapping may associate attributes of each point of the geometry of the current point cloud frame 1710 to each point of the decoded geometry

1741. The mapped attributes may be the attributes of the current point cloud frame 1710. At step 1760, the encoder may encode the mapped attributes 1751. The mapped attributes 1751 may be encoded, as attribute information 1762 into bitstream 1770. The mapped attributes 1751 may be encoded as attribute information 1762, for example, based on the attributes of the geometry motion-compensated point cloud frame 1731.

[0166] FIG. 18 shows an example method for decoding a current point cloud frame. More specifically, FIG. 18 shows an example method for decoding geometry and attributes of a current point cloud frame. The geometry and attributes may be encoded as described herein with respect to FIG. 17. The decoding may use a motion-compensated point cloud frame determined from a motion field selected (e.g., optimized to reduce geometry differences) for geometry coding. Method 1800 shows corresponding, but inverse, operations as those in method 1700, as described herein. Method 1800 may be an inverse of method 1700. One or more steps of the example method of FIG. 18 (e.g., method 1800) may be performed and/or implemented by a decoder (e.g., decoder 120 of FIG. 1).

[0167] An already-decoded reference point cloud frame 1805 may be obtained by a decoder. An already-decoded reference point cloud frame 1805 may be obtained by the decoder, for example, among a plurality of already-decoded reference point cloud frames. At step 1810, the decoder may determine geometry MV 1811. The geometry MV 1811 may be determined, for example, by decoding geometry MV information 1812 from bitstream 1850. Geometry MV 1721, as described herein with respect to FIG. 17, may be the geometry MV 1811. At step 1820, the decoder may determine a geometry motion-compensated point cloud frame 1821. The geometry motion-compensated point cloud frame 1821 may be determined, for example, by performing a motion compensation of the reference point cloud frame 1805. The motion compensation performed on the reference point cloud frame 1805, to determine the geometry motion-compensated point cloud frame 1821, may be based on the geometry MV 1811. At step 1830, the decoder may determine a decoded geometry 1831 of the current point cloud frame. The decoded geometry 1831 of the current point cloud frame may be determined, for example, by decoding geometry information 1832 from bitstream 1850 based on the geometry motion-compensated point cloud frame 1821. At step 1840, the decoder may decode attribute information 1842 from bitstream 1850. The decoder may determine decoded attributes 1841 associated with the decoded geometry 1831. The decoder may determine decoded attributes 1841, associated with the decoded geometry 1831, for example, based on the

decoded attribute information 1842 and the attributes associated with the geometry motion-compensated point cloud frame 1821.

[0168] In at least some systems, inter attribute coding (e.g., as described herein with respect to step 1760 of FIG. 17) and decoding (e.g., as described herein with respect to step 1840) may be based on a geometry motion-compensated point cloud frame (e.g., geometry motion-compensated point cloud frames 1731, 1821) determined from geometry MV (e.g., geometry MVs 1721, 1811) optimized for geometry coding. However, selection of geometry-based MV may not be optimal for attribute coding.

[0169] A geometry motion search (e.g., a geometry motion search as described herein with respect to step 1720 of FIG. 17) may be an iterative method that tests multiple candidate geometry motion vectors. The iterative method may further select the candidate geometry motion vector, among the candidate geometry motion vectors, that minimizes a geometry (e.g., spatial) distortion between the geometry of the current point cloud frame 1710 and a geometry determined by motion compensation of the reference point cloud frame 1705 using the candidate geometry motion vector. The selected geometry motion vectors may not provide a geometry motion-compensated point cloud frame (e.g., geometry motion-compensated point cloud frames 1731, 1821) that is efficient at compressing mapped attributes (e.g., compressing mapped attributes 1751 as described herein with respect to step 1760).

[0170] It may be impossible to determine a motion-compensated point cloud frame that is good for predicting both geometry and attributes of a current point cloud frame, for example, for some types of point clouds. For example, a static object (e.g., a house) may have a static geometry but moving (e.g., non-static or changing) attributes over time (e.g., the projection on a wall of the house of the shadow of a moving object). The optimal motion vector field for coding the geometry of the object (e.g., the house) may be determined as the zero motion field (i.e., motion vectors are equal to zero). However, the optimal motion vector field for coding the attributes (e.g., colors of the wall) of the object may be determined as a non-zero motion field (e.g., that approximates the motion of the shadow on the wall). As another example, a moving object (e.g., a moving semi-trailer truck) may have static local colors (e.g., the semi-trailer may be uniformly white with the projected shadow of a static object). The optimal motion field for geometry coding may be determined to be non-zero. The optimal motion field for attribute coding may be determined to be zero. Non-optimal compression may be obtained, for example, if the

motion field is either not good for geometry coding or not good for attribute coding because the size of the bitstream is increased.

[0171] As described herein, an optimal geometry motion field may not always be optimal for attribute coding. Improvements described herein include advantages such as determining an attribute motion field made of attribute motion vectors selected (e.g., to optimize) for attribute coding. Attribute motion vectors may be used to determine an attribute motion-compensated point cloud frame used for coding (e.g., encoding and/or decoding) attributes of a current point cloud frame.

[0172] Dual motion fields may be used to encode and/or decode geometry and attributes of a current point cloud frame. A geometry motion field may be used for geometry encoding and/or decoding. An attribute motion field may be used for attribute encoding and/or decoding. The attribute motion field may comprise attribute motion vectors. The attribute motion vectors may motion compensate an attribute reference point cloud frame, for example, to determine an attribute motion-compensated point cloud frame used for the encoding and/or decoding of attributes of the current point cloud frame.

[0173] FIG. 19 shows an example method for encoding a current point cloud frame. More specifically, FIG. 19 shows an example method for encoding geometry and attributes of a current point cloud frame. One or more steps of the example method of FIG. 19 (e.g., method 1900) may be performed and/or implemented by an encoder (e.g., encoder 114 of FIG. 1), an example computer system 3500 in FIG. 35, and/or an example computing device 3630 in FIG. 36. Steps (e.g., blocks) of the example method of FIG. 19 may be omitted, performed in other orders, and/or otherwise modified and/or one or more additional steps may be added.

[0174] Steps 1720, 1730, 1740 and 1750 for encoding the geometry of the current point cloud frame 1710, as described herein with respect to FIG. 17, may respectively be the same as steps 1920, 1930, 1940 and 1950 for encoding geometry of the current point cloud frame 1910 as described herein with respect to FIG. 19. Geometry MV 1721, geometry motion-compensated point cloud frame 1731, decoded geometry 1741, and mapped attributes 1751 as described herein with respect to FIG. 17 may respectively be the same as geometry MV 1921, geometry motion-compensated point cloud frame 1931, decoded geometry 1941, and mapped attributes 1951 as described herein with respect to FIG. 19.

[0175] A current point cloud frame 1910 of a sequence of point cloud frames of a dynamic point cloud may be obtained, for example, by an encoder. An already-coded reference point cloud frame 1905 and an already-coded attribute reference point cloud frame 1981 may

be obtained by the encoder, for example, among a plurality of already-decoded reference point cloud frames. Already-coded reference point cloud frame 1905 may be selected from a first plurality of already-coded reference point cloud frames. Attribute reference point cloud frame 1981 may be selected from a second plurality of already-coded point cloud frames. The first and second plurality of already-coded reference point cloud frames may be the same. The first and second plurality of already-coded reference point cloud frames may be different. At step 1980, the encoder may determine attribute motion vectors (MVs) 1983, for example, by performing an attribute motion search. The encoder may perform the attribute motion search such that attribute MV 1983 best approximates the 3D motion field of attributes from the reference point cloud frame 1981 to the current point cloud frame 1910. The encoder may encode attribute MV information 1982 into bitstream 1970 as a representation of the attribute MV 1983. The attribute MV information may indicate the already-coded reference point cloud frame 1981 among a plurality of already coded reference point cloud frames.

[0176] The attribute motion search (e.g., the attribute motion search as described herein with respect to step 1980) may be an iterative method that tests locally multiple candidate attribute motion vectors. The attribute motion search may select the candidate attribute motion vector, among the candidate attribute motion vectors, that minimizes an attribute distortion. The attribute distortion may occur between the attributes of the current point cloud frame 1910 and attributes determined by motion compensation (e.g., adjustment) of the attributes of the attribute reference point cloud frame 1981. The attributes may be determined, for example, by using the candidate attribute motion vector to perform the motion compensation of the attributes of the attribute reference point cloud frame 1981. Attribute distortion may be based on differences between the attributes of the current point cloud frame 1910 and attributes determined by motion compensation of the attributes of the attribute reference point cloud frame 1981. The attribute distortion may be based, for example, on a Sum of Absolute Differences (SAD), a Sum of Absolute Transformed Differences (SATD), a Sum of Squared Error (SSE), etc., between the attributes of the current point cloud frame 1910 and attributes determined by motion compensation of the attributes of the attribute reference point cloud frame 1981. The attributes may be determined, for example, by using the candidate attribute motion vector to perform the motion compensation of the attributes of the attribute reference point cloud frame 1981.

[0177] At step 1990, the encoder may determine an attribute motion-compensated point cloud frame 1991. The encoder may determine an attribute motion-compensated point cloud

frame 1991, for example, by performing a motion compensation of the attribute reference point cloud frame 1981 based on the attribute MV 1983. The attribute distortion, used by the attribute motion search, for a point of the current point cloud frame 1910 may be determined, for example, by comparing the attribute value of this point and the attribute value of (one of) its closest neighbor in the attribute motion-compensated point cloud frame 1991.

[0178] At step 1960, the encoder may encode the mapped attributes 1951, as attribute information 1962, into bitstream 1970. The encoder may encode the mapped attributes 1951 as attribute information 1962 into bitstream 1970, for example, based on the attributes of the attribute motion-compensated point cloud frame 1991. The encoder may encode the mapped attributes 1951 as attribute information 1962 based on the attributes of the attribute motion-compensated point cloud frame 1991, instead of the geometry motion-compensated point cloud frame 1931. The encoder may encode the mapped attributes 1951 as attribute information 1962 based on the attribute motion vector 1983.

[0179] A motion field may be determined, for example, if a geometry motion search is performed. This determined motion field may minimize geometry distortion. The motion field determined, for example, if the geometry motion search as described herein with respect to step 1920) is performed, may minimize geometry distortion. A motion field may be determined, for example, if an attribute motion search is performed. This determined motion filed may minimize attribute distortion. The motion field determined, for example, if the attribute motion search as described herein with respect to step 1980 is performed, may minimize attribute distortion. Geometry and attribute motion-compensated point cloud frames (e.g., geometry motion-compensated point cloud frame 1931 and attribute motion-compensated point cloud frame 1991) may respectively be efficient at predicting geometry and attributes of the current point cloud frame 1910.The compression capability of the point cloud encoder may be improved, for example, by using geometry and attribute motion-compensated point cloud frames to respectively predict geometry and attributes of the current point cloud frame 1910.

[0180] FIG. 20 shows an example method for decoding a current point cloud frame. More specifically, FIG. 20 shows an example method for decoding geometry and attributes of a current point cloud frame. One or more steps of the example method of FIG. 20 (e.g., method 2000) may be performed and/or implemented by a decoder (e.g., decoder 120 of FIG. 1), an example computer system 3500 in FIG. 35, and/or an example computing device 3630 in FIG. 36. Steps (e.g., blocks) of the example method of FIG. 20 may be

omitted, performed in other orders, and/or otherwise modified and/or one or more additional steps may be added.

[0181] The decoding method may decode a point cloud frame encoded by the encoding method as described herein with respect to FIG. 19. Steps 1810, 1820 and 1830 of the decoding method as described herein with respect to FIG. 18 may respectively be the same as steps 2010, 2020 and 2030 of FIG. 20. Geometry MV 1811, geometry motion-compensated point cloud frame 1821, decoded geometry 1831 as described herein with respect to FIG. 18 may respectively be the same as geometry MV 2011, geometry motion-compensated point cloud frame 2021, decoded geometry 2031. An already-coded reference point cloud frame 2005 and an already-decoded attribute reference point cloud frame 2071 may be obtained by the decoder, for example, among a plurality of already-decoded reference point cloud frames. The already-coded reference point cloud frame 1905 and the already-coded reference point cloud frame 2005 may be the same. The already-decoded attribute reference point cloud frame 1981 and the already-decoded attribute reference point cloud frame 2071 may be the same.

[0182] At step 2060, the decoder may determine attribute motion vectors (MVs) 2061. The decoder may determine attribute motion vectors (MVs) 2061, for example, by decoding attribute motion vector information 2062 from bitstream 2050. Attribute MV information 2062 and attribute motion vector (MV) information 1982 may be the same. Attribute MV 1983 and attribute MV 2061 may be the same. At step 2070, the decoder may determine attribute motion-compensated point cloud frame 2072. The decoder may determine attribute motion-compensated point cloud frame 2072, for example, by performing motion compensation of the attribute reference point cloud frame 2071. The motion compensation of the attribute reference point cloud frame 2071 may be based on the attribute MV 2061. At step 2040, the decoder may decode attribute information 2042 from bitstream 2050. The decoder may determine decoded attributes 2041 for the current point cloud frame, for example, based on the attributes associated with the attribute motion-compensated point cloud frame 2072.

[0183] The decoding method as described herein with respect to FIG. 20 may generate decoded attributes 2041 for the current point cloud frame. The decoding method as described herein with respect to FIG. 20 may generate decoded attributes 2041 for the current point cloud frame, for example, by decoding the attribute information 2042. The attribute information 2042 may be representative of the attributes associated with the decoded geometry 2031. The decoding the attribute information 2042 may be based on an attribute motion-

compensated point cloud frame 2072, instead of the geometry motion compensated point cloud frame 2021.

[0184] The inter attribute encoding (e.g., as described herein with respect to step 1960) and decoding (e.g., as described herein with respect to step 2040) may be performed by any inter attribute coder. The inter attribute encoding (e.g., as described herein with respect to step 1960) and decoding (e.g., as described herein with respect to step 2040) may be a pred-lift scheme. The pred-lift scheme may use points of the attribute motion-compensated point cloud frame (e.g., 1991, 2072) in its prediction step. Attributes a^i of a set S^i of points may be predicted by attributes of the subsets a^0, ..., a^{i-1} of the lower level of details $S^0 \cup ... \cup S^{i-1}$, and by attributes associated with points of the attribute motion-compensated point cloud frame (e.g., attribute motion-compensated point cloud frame 1991, attribute motion-compensated point cloud frame 2072).

[0185] The inter attribute encoding (e.g., as described herein with respect to 1960) and decoding (e.g., as described herein with respect to 2040) may be an inter RAHT scheme. The inter RAHT scheme may predict the RAHT coefficients (e.g., AC and DC coefficients) of a current point cloud frame. The inter RAHT scheme may predict the RAHT coefficients (e.g., AC and DC coefficients) of a current point cloud frame, for example, based on the RAHT coefficients (e.g., AC and DC coefficients) of the attribute motion-compensated point cloud frame (e.g., attribute motion-compensated point cloud frame 1991, attribute motion-compensated point cloud frame 2072).

[0186] FIG. 21 shows an example method for encoding attributes. More specifically, FIG. 21 shows an example method for encoding attributes of a current point cloud frame. One or more steps of the example method of FIG. 21 (e.g., method 2100) may be performed and/or implemented by an encoder (e.g., encoder 114 of FIG. 1), an example computer system 3500 in FIG. 35, and/or an example computing device 3630 in FIG. 36. Steps (e.g., blocks) of the example method of FIG. 21 may be omitted, performed in other orders, and/or otherwise modified and/or one or more additional steps may be added.

[0187] At step 2110, the encoder may determine projected attributes 2111. The encoder may perform a projection of attributes of the attribute motion-compensated point cloud frame 1991 (as described herein with respect to FIG. 19) onto the decoded geometry 1941 (as described herein with respect to FIG. 19). The encoder may determine projected attributes 2111, for example, based on attribute motion vectors (e.g., attribute motion vector 1983).

[0188] At step 2120, the encoder may encode, in bitstream 1970 (as described herein with respect to FIG. 19), attribute information 2121. The attribute information 2121 may represent the

mapped attributes 1951 as described herein with respect to (FIG. 19) associated with the decoded geometry 1941 (as described herein with respect to FIG. 19). The encoder may encode the attribute information 2121 based on, for example, attribute prediction (e.g., attribute predictors) determined from the projected attributes 2111.

[0189] The attribute predictors may comprise a respective attribute predictor, for each respective point of points of the decoded geometry 1941. The respective attribute predictor may be based on a projected attribute, of the projected attributes, corresponding to the point. Projected attributes and mapped attributes may belong to the same geometry of the point cloud frame (e.g., the decoded geometry 1941). The prediction of the mapped attributes 1951 based on the projected attributes 2111 may be more efficient, for example, if geometry discrepancy has been removed.

[0190] FIG. 22 shows an example method for decoding attributes of a current point cloud frame. One or more steps of the example method of FIG. 22 (e.g., method 2200) may be performed and/or implemented by a decoder (e.g., decoder 120 of FIG. 1), an example computer system 3500 in FIG. 35, and/or an example computing device 3630 in FIG. 36. Steps (e.g., blocks) of the example method of FIG. 22 may be omitted, performed in other orders, and/or otherwise modified and/or one or more additional steps may be added.

[0191] At step 2210, the decoder may determine projected attributes 2211. The projected attributes 2211 may be determined, for example, by performing a projection of attributes of the attribute motion-compensated point cloud frame 2072 onto the decoded geometry 2031 (FIG. 20). The decoder may determine projected attributes 2211, for example, based on attribute motion vectors (e.g., attribute motion vector 2061). At step 2220, the decoder may determine decoded attributes 2041 for the current point cloud frame. The decoded attributes 2041 may be determined, for example, based on attribute prediction (e.g., attribute predictors) determined from the projected attributes 2211. The attribute prediction (e.g., attribute predictors) may be determined from the projected attributes 2211, for example, by decoding attribute information 2042 (FIG. 20). The attribute predictors may comprise a respective attribute predictor for each respective point of points of the decoded geometry 2031. The respective attribute predictor, for each respective point of points of the decoded geometry 2031, may be based on a projected attribute, of the projected attributes, corresponding to the point.

[0192] FIG. 23 shows an example method for encoding mapped attributes. More specifically, FIG. 23 shows an example method for encoding mapped attributes based on attributes prediction (e.g., attribute predictors). The attributes prediction (e.g., attribute predictors)

may be determined, for example, from projected attributes. One or more steps of the example method of FIG. 23 (e.g., method 2300) may be performed and/or implemented by an encoder (e.g., encoder 114 of FIG. 1), an example computer system 3500 in FIG. 35, and/or an example computing device 3630 in FIG. 36. Steps (e.g., blocks) of the example method of FIG. 23 may be omitted, performed in other orders, and/or otherwise modified and/or one or more additional steps may be added. Method 2300 may correspond to step 2120, as described herein with respect to FIG. 21.

[0193] At step 2310, the encoder may determine predicted attributes. The encoder may determine predicted attributes, for example, from the projected attributes 2111 (as described herein with respect to FIG. 21). The encoder may determine residual attributes 2311, for example, based on differences between the predicted attributes (e.g., attribute predictors) and the mapped attribute 1951 (as described herein with respect to FIG. 19). At step 2320, the encoder may determine quantized residual attributes 2321. The encoder may determine quantized residual attributes 2321, for example, by quantizing the residual attributes 2311. The encoder may determine quantized residual attributes 2321 by quantizing the residual attributes 2311, for example, if lossy compression is allowed and/or performed. At step 2330, the encoder may encode the residual attributes or the quantized residual attributes 2321, as attribute information 2121 (as described herein with respect to FIG. 21).

[0194] FIG. 24 shows an example method for decoding attribute information. More specifically, FIG. 24 shows an example method for decoding attribute information based on attributes prediction determined from projected attributes. One or more steps of the example method of FIG. 24 (e.g., method 2400) may be performed and/or implemented by a decoder (e.g., decoder 120 of FIG. 1), an example computer system 3500 in FIG. 35, and/or an example computing device 3630 in FIG. 36. Steps (e.g., blocks) of the example method of FIG. 24 may be omitted, performed in other orders, and/or otherwise modified and/or one or more additional steps may be added. Method 2400 may correspond to step 2220 as described herein with respect to FIG. 22.

[0195] At step 2410, the decoder may determine quantized residual attributes 2411 (or residual attributes). The decoder may determine quantized residual attributes 2411 (or residual attributes), for example, by decoding attribute information 2042 (as described herein with respect to FIG. 20). At step 2420, the decoder may determine a decoded residual attribute 2421. The decoder may determine a decoded residual attribute 2421, for example, by dequantizing the quantized residual attributes 2411. At step 2430, the decoder may

determine predicted attributes (e.g., attribute predictors) from the projected attributes 2211 (as described herein with respect to FIG. 22). The decoder may determine decoded attributes 2041 (as described herein with respect to FIG. 20). The decoder may determine decoded attributes 2041, for example, based on adding the decoded residual attributes 2421 to the predicted attributes 221 (e.g., attribute predictors).

[0196] FIG. 25 shows an example method for encoding mapped attributes. More specifically, FIG. 25 shows an example method for encoding mapped attributes based on attributes prediction determined from projected attributes. One or more steps of the example method of FIG. 25 (e.g., method 2500) may be performed and/or implemented by an encoder (e.g., encoder 114 of FIG. 1), an example computer system 3500 in FIG. 35, and/or an example computing device 3630 in FIG. 36. Steps (e.g., blocks) of the example method of FIG. 25 may be omitted, performed in other orders, and/or otherwise modified and/or one or more additional steps may be added.

[0197] At step 2510, the encoder may determine smoothed projected attributes 2511. The encoder may determine smoothed projected attributes 2511, for example, by smoothing the projected attributes 2111 (as described herein with respect to FIG. 21). Smoothing the projected attributes 2111 may remove spurious high frequencies that may be damaging to compression performance.

[0198] At step 2520, the encoder may determine predicted attributes (e.g., attribute predictors) from the smoothed projected attributes 2511. The encoder may determine residual attributes 2311 (as described herein with respect to FIG. 23). The encoder may determine residual attributes 2311, for example, based on differences between the predicted attributes (e.g., attribute predictors) and the mapped attributes 1951 (as described herein with respect to FIG. 19).

[0199] FIG. 26 shows an example method for decoding attribute information. More specifically, FIG. 26 shows an example method for decoding attribute information based on attributes prediction determined from projected attributes. One or more steps of the example method of FIG. 26 (e.g., method 1700) may be performed and/or implemented by a decoder (e.g., decoder 120 of FIG. 1), an example computer system 3500 in FIG. 35, and/or an example computing device 3630 in FIG. 36. Steps (e.g., blocks) of the example method of FIG. 26 may be omitted, performed in other orders, and/or otherwise modified and/or one or more additional steps may be added.

[0200] At step 2610, the decoder may determine smoothed projected attributes 2611 (e.g., smoothed values). The decoder may determine smoothed projected attributes 2611, for

example, by smoothing the projected attributes 2211 (as described herein with respect to FIG. 22). Smoothing the projected attributes 2221 may remove spurious high frequencies that may be damaging to compression performance. At step 2620, the decoder may determine predicted attributes from the smoothed projected attributes 2611. The decoder may determine decoded attributes 2041 (as described herein with respect to FIG. 20), for example, by adding the decoded residual attributes 2421 (as described herein with respect to FIG. 24) to the predicted attributes.

[0201] Inter correlation (e.g., temporal correlation) may be used (e.g., exploited) by constructing residual attributes. The residual attributes may be encoded and/or decoded by an intra attribute coding scheme to reduce intra correlation and to improve compression performance. The residual attributes may be encoded and/or decoded based on a prediction with lifting (e.g., pred-lift) scheme. The pred-lift scheme may include S, a set of all residual attributes (or quantized residual attributes) to be encoded and/or decoded.

[0202] FIG. 27 shows an example method for encoding residual attributes. One or more steps of the example method of FIG. 27 (e.g., method 2700) may be performed and/or implemented by an encoder (e.g., encoder 114 of FIG. 1), an example computer system 3500 in FIG. 35, and/or an example computing device 3630 in FIG. 36. Steps (e.g., blocks) of the example method of FIG. 27 may be omitted, performed in other orders, and/or otherwise modified and/or one or more additional steps may be added. Method 2700 may correspond to step 2330 as described herein with respect to FIG. 23.

[0203] At step 2710, the encoder may determine transformed coefficients 2711. The encoder may determine transformed coefficients 2711, for example, based on using (e.g., applying) an intra transform to the (quantized) residual attributes 2321 (as described herein with respect to FIG. 23). At step 2720, the encoder may determine quantized transformed coefficients 2721. The encoder may determine quantized transformed coefficients 2721, for example, by quantizing the transformed coefficients 2711. Quantizing may be used, for example, if lossy compression is allowed and/or performed. At step 2730, the encoder may encode (e.g., entropy encode) the transformed coefficients or the quantized transformed coefficients 2721 as attribute information 2121 (as described herein with respect to FIG. 21).

[0204] FIG. 28 shows an example method for decoding residual attributes. One or more steps of the example method of FIG. 28 (e.g., method 2800) may be performed and/or implemented by a decoder (e.g., decoder 114 of FIG. 1), an example computer system 3500 in FIG. 35, and/or an example computing device 3630 in FIG. 36. Steps (e.g., blocks)

of the example method of FIG. 28 may be omitted, performed in other orders, and/or otherwise modified and/or one or more additional steps may be added. Method 2800 may correspond to step 2410 as described herein with respect to FIG. 24.

[0205] At step 2810, the decoder may determine quantized transformed coefficients 2811 (or transformed coefficients). The decoder may determine quantized transformed coefficients 2811 (or transformed coefficients), for example, by decoding (e.g., entropy decoding) attribute information 2042 (as described herein with respect to FIG. 20). At step 2820, the decoder may determine transformed coefficients 2821. The decoder may determine transformed coefficients 2821, for example, by dequantizing the quantized transformed coefficients 2811. At step 2830, the decoder may determine (quantized) residual attributes 2411. The decoder may determine (quantized) residual attributes 2411, for example, based on using (e.g., applying) an inverse intra transform to the transformed coefficients 2821. The operations of the inverse intra transform may correspond to the inverse operations of the intra transform used (e.g., applied) in step 2710 as described herein with respect to FIG. 27.

[0206] The intra transform may comprise an Adaptive-DCT and the inverse intra transform may comprise an inverse Adaptive-DCT (A-DCT). The intra transform may comprise a RAHT transform and the inverse intra transform may comprise an inverse RAHT transform of the RAHT scheme. Quantization may not be performed, and residual attributes may be encoded (e.g., entropy encoded) without transformation (e.g., without applying intra transform as described herein with respect to step 2710 and inverse intra transform as described herein with respect to step 2830). Quantization may not be performed, and residual attributes may be encoded without transformation, for example, if lossless attribute coding is performed. The intra transform may comprise a Haar transform and the inverse intra transform may comprise an inverse Haar transform. The intra transform may comprise a Haar transform and the inverse intra transform may comprise an inverse Haar transform, for example, if lossless attribute coding is performed.

[0207] FIG. 29 shows an example method for encoding a current point cloud frame. More specifically, FIG. 29 shows an example method for encoding geometry and attributes of a current point cloud frame. One or more steps of the example method of FIG. 29 (e.g., method 2900) may be performed and/or implemented by an encoder (e.g., encoder 114 of FIG. 1), an example computer system 3500 in FIG. 35, and/or an example computing device 3630 in FIG. 36. Steps (e.g., blocks) of the example method of FIG. 29 may be

omitted, performed in other orders, and/or otherwise modified and/or one or more additional steps may be added.

[0208] At step 2910, the encoder may select an attribute reference point cloud frame (e.g., the attribute reference point cloud frame 1981 as described herein with respect to FIG. 19) among a plurality of already-coded reference point cloud frames. The encoder may encode, into bitstream 1970 (FIG. 19), attribute reference point cloud frame information 2911 representative of the selected attribute reference point cloud frame (e.g., the attribute reference point cloud frame 1981). The attribute reference point cloud frame 1981 may be selected independently of the reference point cloud frame 1905 used for the encoding the mapped attributes 1951.

[0209] The attribute reference point cloud frame information 2911 may indicate a global selection of the attribute reference point cloud frame. A single attribute reference point cloud frame 1981 may be selected for encoding the mapped attributes 1951. The single attribute reference point cloud frame 1981 and the reference point cloud frame 1905 may not be the same. The single attribute reference point cloud frame 1981 and the reference point cloud frame 1905 may be the same. The attribute reference point cloud frame information 2911 may indicate whether the attribute reference point cloud frame 1981 and the reference point cloud frame 1905 are the same or not. The attribute reference point cloud frame 1981 and the geometry motion-compensated point cloud frame 1931 may not be the same. The attribute reference point cloud frame 1981 and the geometry motion-compensated point cloud frame 1931 may be the same.

[0210] The geometry MV 1921 may provide a good approximation of the motion field between the current point cloud frame 1910 and the attribute reference point cloud frame 1981. The geometry MV 1921 may provide a good approximation of the motion field between the current point cloud frame 1910 and the attribute reference point cloud frame 1981, for example, if the geometry MV 1921 is a good first approximation of the attribute MV 1983.

[0211] The attribute reference point cloud frame information 2911 may indicate whether the attribute reference point cloud frame 1981 and the geometry motion-compensated cloud frame 1931 are the same or not. The attribute reference point cloud frame information 2911 may indicate an index to a table of indices referencing the attribute reference point cloud frame 1981. Each index, of the table of indices, may reference a specific reference point cloud frame among a plurality of already-coded reference point cloud frames.

[0212] The encoder may determine spatial regions from the decoded geometry 1941. The encoder may determine spatial regions from the decoded geometry 1941 such as bricks.

The bricks may be partitions of the 3D space encompassing the point cloud, and each partition may have their own coding parameters that may be independently coded relative to each other or a set of nodes of the tree associated with the decoded geometry 1941. The encoder may select an attribute reference point cloud frame 1981. The encoder may select an attribute reference point cloud frame 1981, for each spatial region, for encoding the attributes of the points of the point cloud frame 1910 belonging to the spatial region. The attribute reference point cloud frame information 2911 may indicate a local selection of the attribute reference point cloud frame 1981. The attribute reference point cloud frame information 2911 may indicate a definition of the spatial regions. The definition of the spatial region may include boundary information, for instance a bounding box.

[0213] The attribute reference point cloud frame information 2911 may indicate whether an attribute reference point cloud frame 1981 is selected (e.g., global selection) or whether an attribute reference point cloud frame 1981 is selected per spatial region (e.g., local selection). The attribute reference point cloud frame information 2911 may indicate definition of the spatial regions. The attribute reference point cloud frame information 2911 may indicate a reference point cloud frame 1905 for each spatial region. The definition of the spatial region may include boundaries information, for instance a bounding box. The attribute reference point cloud frame (e.g., as described herein with respect to step 2910), selected for a spatial region, may not be equal to the reference point cloud frame 1905 used for encoding the geometry of the points belonging to the spatial region. The attribute reference point cloud frame (e.g., as described herein with respect to step 2910), selected for a spatial region, may be equal to the reference point cloud frame 1905 used for encoding the geometry of the points belonging to the spatial region.

[0214] The attribute reference point cloud frame information 2911 may indicate whether an attribute reference point cloud frame 1981, selected for a spatial region, may be equal to the reference point cloud frame 1905 used for encoding the geometry of the points belonging to the spatial region or not. An attribute reference point cloud frame 1981, selected for a spatial region, may be equal to the geometry motion-compensated point cloud frame 1931 used for encoding the geometry of the points belonging to the spatial region. An attribute reference point cloud frame 1981, selected for a spatial region, may not equal to the geometry motion-compensated point cloud frame 1931 used for encoding the geometry of the points belonging to the spatial region.

[0215] The attribute reference point cloud frame information 2911 may indicate whether an attribute reference point cloud frame 1981, selected for a spatial region, may equal the

geometry motion-compensated point cloud frame 1931 used for encoding the geometry of the points belonging to the spatial region or not. The attribute reference point cloud frame information 2911 may indicate an index of a table of indices referencing an attribute reference point cloud frame 1981 for a spatial region. Each index, of the table of indices, may reference a specific reference point cloud frame among a plurality of already- coded reference point cloud frames.

[0216] FIG. 30 shows an example method for decoding a current point cloud frame. More specifically, FIG. 30 shows an example method for decoding geometry and attributes of a current point cloud frame. One or more steps of the example method of FIG. 30 (e.g., method 3000) may be performed and/or implemented by a decoder (e.g., decoder 120 of FIG. 1), an example computer system 3500 in FIG. 35, and/or an example computing device 3630 in FIG. 36. Steps (e.g., blocks) of the example method of FIG. 30 may be omitted, performed in other orders, and/or otherwise modified and/or one or more additional steps may be added.

[0217] At step 3010, the decoder may decode an attribute reference point cloud frame information 3011 from bitstream 2050 (as described herein with respect to FIG. 20). The decoder may select the attribute reference point cloud frame 2071 (as described herein with respect to FIG. 20) among a plurality of already-decoded reference point cloud frames. The decoder may select the attribute reference point cloud frame 2071 among a plurality of already-decoded reference point cloud frames, for example, based on the decoded attribute reference point cloud frame information 3011.

[0218] The attribute reference point cloud frame 2071 may be selected independently of the reference point cloud frame 2005 (as described herein with respect to FIG. 20) used for determining the decoded attributes 2041. The attribute reference point cloud frame information 3011 may indicate a global selection of the attribute reference point cloud frame. An attribute reference point cloud frame 2071 may be selected for determining the decoded attributes 2041. The attribute reference point cloud frame 2071 and the reference point cloud frame 2005 may not the same. The attribute reference point cloud frame 2071 and the reference point cloud frame 2005 may be the same. Attribute reference point cloud frame information (e.g., the attribute reference point cloud frame information 3011) may indicate whether the attribute reference point cloud frame 2071 and the reference point cloud frame 2005 are the same or not.

[0219] The attribute reference point cloud frame 2071 and the geometry motion-compensated point cloud frame 2021 may not be the same. The single attribute reference point cloud

frame 2071 and the geometry motion-compensated point cloud frame 2021 may be the same.

[0220] The geometry MV 2011 may provide a good approximation of the motion field between the current point cloud frame 1910 and the attribute reference point cloud frame 1981. The geometry MV 2011 may provide a good approximation of the motion field between the current point cloud frame 1910 and the attribute reference point cloud frame 1981, for example, if the geometry MV 2011 is a good first approximation of the attribute MV 2061. The attribute reference point cloud frame information 3011 may indicate whether the attribute reference point cloud frame 2071 and the geometry motion-compensated cloud frame 2021 are the same or not. The attribute reference point cloud frame information 3011 may indicate an index to a table of indices referencing the single attribute reference point cloud frame 2071. Each index, of the table of indices, may reference a specific reference point cloud frame among a plurality of already-decoded reference point cloud frames.

[0221] The decoder may determine spatial regions from the decoded geometry 2031. The decoder may determine spatial regions from the decoded geometry 2031 such as bricks. The bricks may be partitions of the 3D space encompassing the point cloud, and each partition may have each their own coding parameters that may be independently coded relative to each other) or a sets of nodes of the tree associated with the decoded geometry 2031. The decoder may select an attribute reference point cloud frame 2071. The decoder may select an attribute reference point cloud frame 2071, for each spatial region, for determining the decoded attributes 2041 belonging to the spatial region. The attribute reference point cloud frame information 3011 may indicate a local selection of the attribute reference point cloud frame 2071. The attribute reference point cloud frame information 3011 may indicate a definition of the spatial regions. The definition of the spatial region may include boundaries information, for instance a bounding box.

[0222] The attribute reference point cloud frame information 3011 may indicate whether an attribute reference point cloud frame 2071 is selected (e.g., global selection) or whether an attribute reference point cloud frame 2071 is selected per spatial region (e.g., local selection). The attribute reference point cloud frame information 3011 may indicate a definition of the spatial regions. The definition of the spatial region may include boundaries information, for instance a bounding box. The attribute reference point cloud frame information 3011 may indicate a reference point cloud frame 2005 for each spatial region. The definition of the spatial region may include boundaries information, for

instance a bounding box. The attribute reference point cloud frame (e.g., as described herein with respect to step 3010), selected for a spatial region, may not be equal to the reference point cloud frame 2005 used for decoding the geometry of the points belonging to the spatial region. The attribute reference point cloud frame (e.g., as described herein with respect to step 3010), selected for a spatial region, may be equal to the reference point cloud frame 2005 used for decoding the geometry of the points belonging to the spatial region. The attribute reference point cloud frame information 3011 may indicate whether an attribute reference point cloud frame 2071 selected for a spatial region may be equal to the reference point cloud frame 2005 used for decoding the geometry of the points belonging to the spatial region or not.

[0223] An attribute reference point cloud frame 2071, selected for a spatial region, may be equal to the geometry motion-compensated point cloud frame 2021 used for decoding the geometry of the points belonging to the spatial region. An attribute reference point cloud frame 2071, selected for a spatial region, may not be equal to the geometry motion-compensated point cloud frame 2021 used for decoding the geometry of the points belonging to the spatial region. The attribute reference point cloud frame information 3011 may indicate whether an attribute reference point cloud frame 2071, selected for a spatial region, may equal the geometry motion-compensated point cloud frame 2021 used for decoding the geometry of the points belonging to the spatial region or not. The attribute reference point cloud frame information 3011 may indicate an index of a table of indices referencing an attribute reference point cloud frame 2071 for a spatial region. Each index, of the table of indices, may reference a specific reference point cloud frame among a plurality of already-decoded reference point cloud frames.

[0224] The attribute MV information 1982 (as described herein with respect to FIG. 19) and the attribute MV information 2062 (as described herein with respect to FIG. 20) may indicate that the attribute motion vectors (e.g., attribute motion vector 1983, attribute motion vector 2061) and the geometry motion vectors (e.g., geometry motion vector 1921, geometry motion vector 2011) are associated with a same motion field structure.

[0225] A motion field structure may be made of a partitioning of a 3D space encompassing the decoded geometry (e.g., decoded geometry 1941, decoded geometry 2031) into motion units (MUs). A geometry motion vector may be associated with (e.g., correspond to) each of the motion units. The motion compensation of the reference point cloud frame 1905 (as described herein with respect to step 1920) and the motion compensation of the reference point cloud frame 2005 (as described herein with respect to step 2020) may be performed,

for example, within each motion unit. The motion compensation of the reference point cloud frame 1905 and the motion compensation of the reference point cloud frame 2005 may be performed within each motion unit, for example, based on the associated geometry motion vector. An attribute motion vector may be associated with (e.g., correspond to) each of the motion units. The motion compensation of the attribute reference point cloud frame 1981 (as described herein with respect to step 1990) and the motion compensation of the attribute reference point cloud frame 2071 (as described herein with respect to step 2070) may be performed, for example, within each motion unit. The motion compensation of the attribute reference point cloud frame 1981 and the motion compensation of the attribute reference point cloud frame 2071 may be performed within each motion unit, for example, based on the associated attribute motion vector.

[0226] The motion units (MUs) may be non-intersecting cuboids. The attribute MV information (e.g., attribute MV information 1982, attribute MV information 2062) may define and/or indicate the motion field structure. The motion field structure may be associated with the attribute MV (e.g., attribute MV 1983, attribute MV 2061) and the geometry MV (e.g., geometry MV 1921, geometry MV 2011). The attribute MV information (e.g., attribute MV information 1982, attribute MV information 2062) may define and/or indicate the motion filed structure, for example, via a partitioning of the 3D space, associated with the attribute MV (e.g., attribute MV 1983, attribute MV 2061) and the geometry MV (e.g., geometry MV 1921, geometry MV 2011). The attribute MV information (e.g., attribute MV information 1982, attribute MV information 2062) may indicate that the attribute motion vectors (e.g., attribute motion vector1983, attribute motion vector 2061) and the geometry motion vectors (e.g., geometry motion vector1921, geometry motion vector 2011) are associated with different motion field structures.

[0227] An attribute motion field structure, associated with the attribute MV (e.g., attribute MV 1983, attribute MV 2061), may be determined. The attribute motion field structure, associated with the attribute MV (e.g., attribute MV 1983, attribute MV 2061), may be determined, for example, from a first partitioning of the 3D space encompassing the decoded geometry (e.g., 1941, 2031) into attribute motion units. A geometry motion filed structure associated with the geometry motion vectors (e.g., geometry motion vector1921, geometry motion vector 2011) may be determined. The geometry motion filed structure associated with the geometry motion vectors (e.g., geometry motion vector 1921, geometry motion vector 2011) may be determined, for example, from a second partitioning of the 3D space encompassing the decoded geometry (e.g., decoded geometry

1941, decoded geometry 2031) into geometry motion units. A motion field structure may be defined by a shape, size and/or location of motion units of a partition of the 3D space encompassing the decoded geometry (e.g., decoded geometry 1941, decoded geometry 2031).

[0228] Attribute MV information (e.g., attribute MV information 1982, attribute MV information 2062) may indicate a bound (e.g. threshold, range limit, etc.) on a maximum magnitude of the geometry motion vectors (e.g., geometry motion vector 1921, geometry motion vector 2011). Additionally or alternatively, attribute MV information (e.g., attribute MV information 1982, attribute MV information 2062) may indicate a bound (e.g. threshold, range limit, etc.) on the attribute motion vectors (e.g., attribute motion vector 1983, attribute motion vector 2061).

[0229] FIG. 31 shows an example method for encoding an attribute motion vector into a bitstream. One or more steps of the example method of FIG. 31 (e.g., method 3100) may be performed and/or implemented by an encoder (e.g., encoder 114 of FIG. 1), an example computer system 3500 in FIG. 35, and/or an example computing device 3630 in FIG. 36. Steps (e.g., blocks) of the example method of FIG. 31 may be omitted, performed in other orders, and/or otherwise modified and/or one or more additional steps may be added.

[0230] At step 3110, the encoder may determine an attribute MV (e.g., attribute MV 1983). The encoder may determine the attribute MV 1983, for example, by performing an attribute motion search. The attribute motion search may be performed such that attribute MV 1983 best approximates the 3D motion field of the attributes from the attribute reference point cloud frame 1981 to the current point cloud frame 1910. At step 3120, the encoder may determine an attribute motion field structure associated with the attribute MV 1983. The encoder may determine an attribute motion field structure associated with the attribute MV 1983, for example, from a geometry motion field structure associated with the geometry MV 1921.

[0231] At step 3130, the encoder may determine a motion vector residual (e.g., motion vector residual 3131). The encoder may determine the motion vector residual 3131, for example, based on differences between geometry MV 1921 of the geometry motion field structure and attribute MV 1983 of the attribute motion field structure. Geometry MV 1921 may be subtracted from attribute MV 1983. At step 3140, the encoder may determine quantized motion vector residual 3141. The encoder may determine quantized motion vector residual 3141, for example, by quantizing the motion vector residual 3131. At step 3150, the encoder may encode (e.g., entropy encode) the (quantized) motion vector residual

3141. The encoder may determine the attribute MV information 1982 as being representative of the encoded motion residual vector. The number of bits required to encode the attribute MV information 1982 in bitstream 1970 may be reduced, for example, by encoding the (quantized) motion vector residual 3142 and determining the attribute MV information 1982 as being representative of the encoded motion residual vector.

[0232] The geometry motion field structure may be determined (e.g., defined) by partitioning the 3D space encompassing the decoded geometry 1941 into geometry MUs. A geometry MV 1921 may be associated with (e.g., correspond to) each of the geometry MUs. The attribute motion field structure may be determined (e.g., defined), for example, by partitioning the geometry motion field structure into attribute MUs. Attribute MUs may be determined, for example, by partitioning geometry MUs. The encoder may determine the motion vector residual, for example, by subtracting geometry MV 1921 associated with geometry MUs from attribute MV 1983 and attribute MV 2061 associated with attribute MUs. The attribute MV 1983 and attribute MV 2061 may be associated with attribute MUs that partitioned the geometry MUs.

[0233] FIG. 32 shows an example method for decoding the attribute MV information from a bitstream. One or more steps of the example method of FIG. 32 (e.g., method 3200) may be performed and/or implemented by a decoder (e.g., decoder 120 of FIG. 1), an example computer system 3500 in FIG. 35, and/or an example computing device 3630 in FIG. 36. Steps (e.g., blocks) of the example method of FIG. 32 may be omitted, performed in other orders, and/or otherwise modified and/or one or more additional steps may be added.

[0234] At step 3210, the decoder may determine a (quantized) motion vector residual 3211. The decoder may determine a (quantized) motion vector residual 3211, for example, by decoding (e.g., entropy decoding) the attribute MV information 2062. At step 3220, the decoder may determine motion vector residual 3221. The decoder may determine motion vector residual 3221, for example, by inverse quantizing the quantized motion vector residual 3211. At step 3230, the decoder may determine attribute MV 2061 of an attribute motion field structure. The decoder may determine attribute MV 2061 of an attribute motion field structure, for example, based on adding the motion vector residual 3211 and geometry MV 2011 of a geometry motion field structure.

[0235] The geometry motion field structure may be determined (e.g., defined), for example, by partitioning the 3D space encompassing the decoded geometry 2031 into geometry MUs. The attribute motion filed structure may be determined (e.g., defined), for example by partitioning the geometry motion field structure into attribute MUs. Attribute MUs may

be determined, for example, by partitioning geometry MUs. The decoder may determine the attribute MV 2061 associated with the attribute MUs (e.g., one attribute MV per attribute MU), for example, by adding the motion vector residual 3221 to the geometry MV 2011 associated with geometry MUs containing the attribute MUs.

[0236] References in the specification to encoding information (e.g., geometry information, attribute information, geometry MV information, attribute MV information) may indicate encoding information as at least one single bit (e.g., a flag) or as at least one word comprising each more than one bit or as a combination of at least one flag and at least one word. Encoding information into a bitstream may indicate writing into the bitstream at least one single bit (e.g., a flag) or at least one word comprising each more than one bit or a combination of at least one flag and at least one word representing the information according to a specific syntax.

[0237] References in the specification to decoding information (e.g., geometry information, attribute information, geometry MV information, attribute MV information) may indicate decoding information from at least one single bit (e.g., a flag) or from at least one word comprising each more than one bit or from a combination of at least one flag and at least one word. Decoding information from a bitstream may indicate parsing the bitstream according to a specific syntax and reading from the bitstream at least one single bit (e.g., a flag) or at least one word comprising each more than one bit or a combination of at least one flag and at least one word representing the information.

[0238] FIG. 33 shows an example method for encoding a point cloud frame. More specifically, FIG. 33 shows a flowchart 3300 of an example method for encoding geometry and attributes of a point cloud frame. The point cloud frame may be a current point cloud frame. The method of flowchart 3300 may be performed and/or implemented by an encoder (e.g., encoder 114 in FIG. 1), an example computer system 3500 in FIG. 35, and/or an example computing device 3630 in FIG. 36. Steps of the example method of FIG. 33 may be omitted, performed in other orders, and/or otherwise modified, and/or one or more additional steps may be added.

[0239] At step 3305, the encoder may determine geometry motion vectors (MVs). The encoder may determine geometry MVs, for example, based on performing a geometry motion search. The geometry MVs may approximate the 3D motion field from a reference point cloud frame to a geometry of a current point cloud frame. The reference point cloud frame may be determined (e.g., selected) from a first plurality of already-coded point cloud frames. The encoder may encode geometry MV information into the bitstream. The

encoder may encode geometry MV information into the bitstream, for example, as a representation of one or more geometry MVs.

[0240] At step 3310, the encoder may motion compensate a reference point cloud frame. The encoder may motion compensate the reference point cloud frame discussed herein with respect to step 3305. The encoder may motion compensate a reference point cloud frame, for a current point cloud frame, for example, based on the geometry motion vectors. The encoder may motion compensate a reference point cloud frame, for a current point cloud frame, based on the geometry motion vectors, for example, to determine a geometry motion-compensated point cloud frame.

[0241] At step 3315, the encoder may determine attribute motion vectors (MVs). The encoder may determine attribute MVs, for example, based on performing an attribute motion search. The encoder may perform the attribute motion search such that attribute MVs best approximates a 3D motion field of attributes from a reference point cloud frame (e.g., attribute reference point cloud frame) to a current point cloud frame. The reference point cloud frame (e.g., attribute reference point cloud frame) may be determined (e.g., selected) from a first plurality of already-coded point cloud frames. The encoder may encode attribute MV information into the bitstream. The encoder may encode attribute MV information into the bitstream, for example, as a representation of one or more attribute MVs.

[0242] At step 3320, the encoder may motion compensate an attribute reference point cloud frame. The encoder may motion compensate the attribute reference point cloud frame discussed herein with respect to step 3315. The encoder may motion compensate an attribute reference point cloud frame, for the point cloud frame, for example, based on attribute motion vectors. The encoder may motion compensate an attribute reference point cloud frame, for the point cloud frame, based on attribute motion vectors, for example, to determine an attribute motion-compensated point cloud frame. The attribute reference point cloud frame may be determined (e.g., selected) from a second plurality of already-coded point cloud frames. The first and second plurality of already-coded point cloud frames may be the same.

[0243] At step 3330, the encoder may encode a geometry associated with the point cloud frame. The encoder may encode a geometry associated with the point cloud frame, for example, based on the geometry MVs. The encoder may encode a geometry associated with the point cloud frame, for example, based on the geometry motion-compensated point cloud frame. At step 3340, the encoder may encode attributes associated with a reconstructed

geometry of the point cloud frame. The encoder may encode attributes associated with a reconstructed geometry of the point cloud frame, for example, based on the attribute MVs. The encoder may encode attributes associated with a reconstructed geometry of the point cloud frame, for example, based on attributes of the attribute motion-compensated point cloud frame.

[0244] FIG. 34 shows an example method for decoding a point cloud frame. More specifically, FIG. 34 shows a flowchart 3400 of an example method for decoding geometry and attributes of a point cloud frame. The point cloud frame may be a current point cloud frame. The method of flowchart 3400 may be performed and/or implemented by a decoder (e.g., decoder 120 in FIG. 1). Steps of the example method of FIG. 24 may be omitted, performed in other orders, and/or otherwise modified, and/or one or more additional steps may be added.

[0245] At step 3405, the decoder may determine geometry motion vectors (MVs). The decoder may determine geometry MVs, for example, by decoding geometry MV information from a bitstream. At step 3410, the decoder may motion compensate a reference point cloud frame. The decoder may motion compensate a reference point cloud frame, for a current point cloud frame, for example, based on the geometry motion vectors. The decoder may motion compensate a reference point cloud frame, for a current point cloud frame, based on the geometry motion vectors, for example, to determine a geometry motion-compensated point cloud frame. The reference point cloud frame may be determined (e.g., selected), for example, from a first plurality of already-coded point cloud frames.

[0246] At step 3415, the decoder may determine attribute motion vectors (MVs). The decoder may determine attribute MVs, for example, by attribute geometry MV information from a bitstream. At step 3420, the decoder may motion compensate an attribute reference point cloud frame. The decoder may motion compensate an attribute reference point cloud frame, for the point cloud frame, for example, based on the attribute MVs. The decoder may motion compensate an attribute reference point cloud frame, for the point cloud frame, based on attribute MVs, for example, to determine an attribute motion-compensated point cloud frame. The attribute reference point cloud frame may be determined (e.g., selected), for example, from a second plurality of already-coded point cloud frames. The first and second plurality of already-coded point cloud frames may be the same.

[0247] At step 3430, the decoder may decode a geometry associated with the point cloud frame. The decoder may decode the geometry associated with the point cloud frame, for example,

based on the geometry MVs. The decoder may decode the geometry associated with the point cloud frame based on the geometry MVs, for example, to determine a reconstructed geometry of the point cloud frame. The decoder may decode the geometry associated with the point cloud frame, for example, based on the geometry motion-compensated point cloud frame. The decoder may decode the geometry associated with the point cloud frame based on the geometry motion-compensated point cloud frame, for example, to determine a reconstructed geometry of the point cloud frame. At step 3440, the decoder may decode attributes associated with the reconstructed geometry. The decoder may decode attributes associated with the reconstructed geometry, for example, based on the attribute MVs. The decoder may decode attributes associated with the reconstructed geometry, for example, based on attributes of the attribute motion-compensated point cloud frame.

[0248] FIG. 35 shows an example computer system in which examples of the present disclosure may be implemented. The example computer system 3500 shown in FIG. 35 may implement one or more of the methods described herein. Various devices and/or systems described herein (e.g., in FIGS. 1, 2, and 3) may be implemented in the form of one or more computer systems 3500. Furthermore, each of the steps of the flowcharts depicted in this disclosure may be implemented on one or more computer systems 3500.

[0249] The computer system 3500 may comprise one or more processors, such as a processor 3504. The processor 3504 may be a special purpose processor, a general purpose processor, a microprocessor, and/or a digital signal processor. The processor 3504 may be connected to a communication infrastructure 3502 (for example, a bus or network). The computer system 3500 may also comprise a main memory 3506 (e.g., a random access memory (RAM)), and/or a secondary memory 3508.

[0250] The secondary memory 3508 may comprise a hard disk drive 3510 and/or a removable storage drive 3512 (e.g., a magnetic tape drive, an optical disk drive, and/or the like). The removable storage drive 3512 may read from and/or write to a removable storage unit 3516. The removable storage unit 3516 may comprise a magnetic tape, optical disk, and/or the like. The removable storage unit 3516 may be read by and/or may be written to the removable storage drive 3512. The removable storage unit 3516 may comprise a computer usable storage medium having stored therein computer software and/or data.

[0251] The secondary memory 3508 may comprise other similar means for allowing computer programs or other instructions to be loaded into the computer system 3500. Such means may include a removable storage unit 3518 and/or an interface 3514. Examples of such means may comprise a program cartridge and/or cartridge interface (such as in video game

devices), a removable memory chip (such as an erasable programmable read-only memory (EPROM) or a programmable read-only memory (PROM)) and associated socket, a thumb drive and USB port, and/or other removable storage units 3518 and interfaces 3514 which may allow software and/or data to be transferred from the removable storage unit 3518 to the computer system 3500.

[0252] The computer system 3500 may also comprise a communications interface 3520. The communications interface 3520 may allow software and data to be transferred between the computer system 3500 and external devices. Examples of the communications interface 3520 may include a modem, a network interface (e.g., an Ethernet card), a communications port, etc. Software and/or data transferred via the communications interface 3520 may be in the form of signals which may be electronic, electromagnetic, optical, and/or other signals capable of being received by the communications interface 3520. The signals may be provided to the communications interface 3520 via a communications path 3522. The communications path 3522 may carry signals and may be implemented using wire or cable, fiber optics, a phone line, a cellular phone link, an RF link, and/or any other communications channel(s).

[0253] A computer program medium and/or a computer readable medium may be used to refer to tangible storage media, such as removable storage units 3516 and 3518 or a hard disk installed in the hard disk drive 3510. The computer program products may be means for providing software to the computer system 3500. The computer programs (which may also be called computer control logic) may be stored in the main memory 3506 and/or the secondary memory 3508. The computer programs may be received via the communications interface 3520. Such computer programs, when executed, may enable the computer system 3500 to implement the present disclosure as discussed herein. In particular, the computer programs, when executed, may enable the processor 3504 to implement the processes of the present disclosure, such as any of the methods described herein. Accordingly, such computer programs may represent controllers of the computer system 3500.

[0254] Features of the disclosure may be implemented in hardware using, for example, hardware components such as application-specific integrated circuits (ASICs) and gate arrays. Implementation of a hardware state machine to perform the functions described herein will also be apparent to persons skilled in the relevant art(s).

[0255] FIG. 36 shows example elements of a computing device that may be used to implement any of the various devices described herein, including, for example, a source device (e.g.,

102), an encoder (e.g., 114), a destination device (e.g., 106), a decoder (e.g., 120), and/or any computing device described herein. The computing device 3630 may include one or more processors 3631, which may execute instructions stored in the random-access memory (RAM) 3633, the removable media 3634 (such as a Universal Serial Bus (USB) drive, compact disk (CD) or digital versatile disk (DVD), or floppy disk drive), or any other desired storage medium. Instructions may also be stored in an attached (or internal) hard drive 3635. The computing device 3630 may also include a security processor (not shown), which may execute instructions of one or more computer programs to monitor the processes executing on the processor 3631 and any process that requests access to any hardware and/or software components of the computing device 3630 (e.g., ROM 3632, RAM 3633, the removable media 3634, the hard drive 3635, the device controller 3637, a network interface 3639, a GPS 3641, a Bluetooth interface 3642, a WiFi interface 3643, etc.). The computing device 3630 may include one or more output devices, such as the display 3636 (e.g., a screen, a display device, a monitor, a television, etc.), and may include one or more output device controllers 3637, such as a video processor. There may also be one or more user input devices 3638, such as a remote control, keyboard, mouse, touch screen, microphone, etc. The computing device 3630 may also include one or more network interfaces, such as a network interface 3639, which may be a wired interface, a wireless interface, or a combination of the two. The network interface 3639 may provide an interface for the computing device 3630 to communicate with a network 3640 (e.g., a RAN, or any other network). The network interface 3639 may include a modem (e.g., a cable modem), and the external network 3640 may include communication links, an external network, an in-home network, a provider's wireless, coaxial, fiber, or hybrid fiber/coaxial distribution system (e.g., a DOCSIS network), or any other desired network. Additionally, the computing device 3630 may include a location-detecting device, such as a global positioning system (GPS) microprocessor 3641, which may be configured to receive and process global positioning signals and determine, with possible assistance from an external server and antenna, a geographic position of the computing device 3630.

[0256] The example in FIG. 36 may be a hardware configuration, although the components shown may be implemented as software as well. Modifications may be made to add, remove, combine, divide, etc. components of the computing device 3630 as desired. Additionally, the components may be implemented using basic computing devices and components, and the same components (e.g., processor 3631, ROM storage 3632, display 3636, etc.) may be used to implement any of the other computing devices and components

described herein. For example, the various components described herein may be implemented using computing devices having components such as a processor executing computer-executable instructions stored on a computer-readable medium, as shown in FIG. 36. Some or all of the entities described herein may be software based, and may co-exist in a common physical platform (e.g., a requesting entity may be a separate software process and program from a dependent entity, both of which may be executed as software on a common computing device).

[0257] A computing device may perform a method comprising multiple operations. The computing device may comprise a decoder. The computing device may determine geometry motion vectors from a bit stream. The computing device may determine attribute motion vectors from a bit stream. The computing device may motion compensate a reference point cloud frame, for a current point cloud frame, based on the geometry motion vectors to determine a geometry motion-compensated point cloud frame. The computing device may motion compensate an attribute reference point cloud frame, for the point cloud frame, based on the attribute motion vectors to determine an attribute motion-compensated point cloud frame. The computing device may determine a reconstructed geometry of a point cloud frame. The point cloud frame may be associated with content. The computing device may determine the reconstructed geometry of the point cloud frame by decoding a geometry associated with the point cloud frame. A reference point cloud frame, for the point cloud from, may be motion compensated based on the geometry motion vectors to determine the reconstructed geometry of the point cloud frame. The geometry associated with the point cloud frame may be decoded based on the geometry motion vectors. The computing device may decode geometry motion vector information indicating the geometry motion vectors, wherein the geometry motion vectors are associated with a same motion field structure. The computing device may decode attribute motion vector information indicating the attribute motion vectors. The computing device may decode attribute motion vector information by determining, based on decoding the attribute motion vector information, a motion vector residual. The computing device may decode attribute motion vector information by determining, based on adding the motion vector residual and geometry motion vectors of a geometry motion field structure, attribute motion vectors of an attribute motion field structure. The computing device may decode attributes associated with the reconstructed geometry. The computing device may decode the attributes associated with the reconstructed geometry by decoding residual attributes indicating differences between the attributes of the reconstructed geometry and

attribute predictors associated with the reconstructed geometry. The computing device may decode the attributes associated with the reconstructed geometry by determining based on adding the attribute predictors and the residual attributes, the attributes of the reconstructed geometry. The attributes associated with the reconstructed geometry maybe decoded based on the attribute motion vectors. The computing device may determine, based on the attribute motion vectors, projected attributes. The computing device may determine the projected attributes based on motion compensating an already-coded reference point cloud frame, wherein the already-coded reference point cloud frame is motion compensated by the attribute motion vectors. The computing device may determine the projected attributes by determining an attribute motion compensated point cloud frame. The computing device may determine the projected attributes by projecting attributes of the attribute motion compensated point cloud frame onto the reconstructed geometry. The computing device may determine, based on the projected attributes, attribute predictors of the attributes associated with the reconstructed geometry. The computing device may decode, based on the attribute predictors, the attributes associated with the reconstructed geometry. The computing device may decode attribute reference point cloud frame information indicating an attribute reference point cloud frame. The computing device may select, based on the attribute reference point cloud frame information, the attribute reference point cloud frame from a plurality of already-decoded reference point cloud frames. The computing device may comprise one or more processors and memory, storing instructions that, when executed by the one or more processors, perform the method described herein. A system may comprise the computing device configured to perform the described method, additional operations, and/or include additional elements; and a second computing device configured to encode the point cloud frame. A computer-readable medium may store instructions that, when executed, cause performance of the described method, additional operations, and/or include additional elements.

[0258] A computing device may perform a method comprising multiple operations. The computing device may comprise a encoder. The computing device may determine, based on a point cloud frame associated with content, geometry motion vectors. The computing device may determine, based on the point cloud frame, attribute motion vectors. The computing device may motion compensate a reference point cloud frame, for a current point cloud frame, based on the geometry motion vectors to determine a geometry motion-compensated point cloud frame. The computing device may motion compensate an

attribute reference point cloud frame, for the point cloud frame, based on the attribute motion vectors to determine an attribute motion-compensated point cloud frame. The computing device may encode, based on the geometry motion vectors, a geometry associated with the point cloud frame. The computing device may encode, based on the attribute motion vectors, attributes associated with a reconstructed geometry of the point cloud frame. The computing device may encode the attributes associated with the reconstructed geometry by: determining, based on the attribute motion vectors, projected attributes; determining, based on the projected attributes, attribute predictors of the attributes associated with the reconstructed geometry; and encoding, based on the attribute predictors, the attributes associated with the reconstructed geometry. The computing device may determine projected attributes based on motion compensating an already-coded reference point cloud frame, wherein the already-coded reference point cloud frame is motion compensated by the attribute motion vectors. The computing device may determine the attribute motion vectors based on differences between: the attributes associated with the reconstructed geometry of the point cloud frame; and attributes of a geometry, of the already-coded reference point cloud frame, adjusted by the attribute motion vectors. The computing device may determine, based on mapping attributes of the geometry associated with the point cloud frame to the reconstructed geometry, the attributes associated with the reconstructed geometry of the point cloud frame. The attributes associated with the reconstructed geometry of the point cloud frame comprise colors. The computing device may determine, based on recoloring, the mapped attributes of the geometry associated with the point cloud frame. The computing device may motion compensate a reference point cloud frame, for the point cloud frame, based on geometry motion vectors. The computing device may motion compensate the reference point cloud frame, based on the geometry motion vectors, to determine the reconstructed geometry of the point cloud frame. The computing device may encode geometry motion vector information indicating the geometry motion vectors, wherein the geometry motion vectors are associated with a same motion field structure. The computing device may encode attribute motion vector information indicating the attribute motion vectors. The computing device may encode the attribute motion vector information by: determining a motion vector residual based on differences between the attribute motion vectors and the geometry motion vectors; and encoding the motion vector residual as the attribute motion vector information. The computing device may comprise one or more processors and memory, storing instructions that, when executed by the one or more processors, perform

the method described herein. A system may comprise the computing device configured to perform the described method, additional operations, and/or include additional elements; and a second computing device configured to decode the point cloud frame. A computer-readable medium may store instructions that, when executed, cause performance of the described method, additional operations, and/or include additional elements.

[0259] A computing device may perform a method comprising multiple operations. The computing device may comprise a decoder. The computing device may determine, from a bitstream, attribute motion vectors and geometry motion vectors. The computing device may determine a reconstructed geometry of a point cloud frame associated with content, by decoding, based on the geometry motion vectors, a geometry associated with the point cloud frame. The computing device may determine, based on the attribute motion vectors, projected attributes. The computing device may determine, based on the projected attributes, attribute predictors of attributes associated with the reconstructed geometry. The computing device may decode, from the bitstream, residual attributes indicating differences between the attributes associated with the reconstructed geometry and the attribute predictors. The computing device may decode the residual attributes by: decoding, from the bitstream, transformed coefficients indicating the residual attributes; and determining the residual attributes based on applying an inverse intra transform to the decoded transformed coefficients. The residual attributes may be decoded based on a prediction with lifting transform scheme. The computing device may determine the attribute predictors by: smoothing the projected attributes; and determining, based on the smoothed projected attributes, the attribute predictors. The computing device may comprise one or more processors and memory, storing instructions that, when executed by the one or more processors, perform the method described herein. A system may comprise the computing device configured to perform the described method, additional operations, and/or include additional elements; and a second computing device configured to encode the point cloud frame. A computer-readable medium may store instructions that, when executed, cause performance of the described method, additional operations, and/or include additional elements.

[0260] A computing device may perform a method comprising multiple operations. The computing device may comprise a encoder. The computing device may motion compensate a reference point cloud frame, for a current point cloud frame, based on geometry motion vectors to determine a geometry motion-compensated point cloud frame. The computing device may motion compensate an attribute reference point cloud frame,

for the point cloud frame, based on attribute motion vectors to determine an attribute motion-compensated point cloud frame. The computing device may encode a geometry associated with the point cloud frame based on the geometry motion-compensated point cloud frame. The computing device may encode attributes associated with a reconstructed geometry of the point cloud frame based on attributes of the attribute motion-compensated point cloud frame. The geometry motion compensated point cloud frame may be determined from an already-coded reference point cloud frame. The geometry motion compensated point cloud frame may be determined based on motion compensating the already-coded reference point cloud frame. The already-coded reference point cloud frame may be motion compensated by a geometry motion vector. The attribute motion compensated point cloud frame may be determined from an already-coded reference point cloud frame. The attribute motion compensated point cloud frame may be determined based on motion compensating the already-coded reference point cloud frame. The already-coded reference point cloud frame may be motion compensated by an attribute motion vector. The computing device may encode the attribute motion vector and/or an indication (e.g., index or ID) of the already-coded reference point cloud frame. The computing device may determine the attribute motion vector based on differences between the attributes associated with the reconstructed geometry and the attributes of a geometry, of the already-coded reference point cloud frame, adjusted by the attribute motion vector. The computing device may encode the geometry motion vector and/or an indication (e.g., index or ID) of the already-coded reference point cloud frame. The computing device may determine the geometry motion vector based on differences between the reconstructed geometry and a geometry, of the already-coded reference point cloud frame, adjusted by the geometry motion vector. The computing device may encode the attributes by: determining residual attributes based on differences between the attributes of the reconstructed geometry and the attribute predictors; and encoding the residual attributes. The computing device may determine the attribute predictors by smoothing the projected attributes, the attribute predictors being determined from the smoothed projected attributes. The attribute predictors may comprise a respective attribute predictor, for each respective vertex of vertices of the reconstructed geometry, that is based on a projected attribute, of the projected attributes, corresponding to the vertex. The computing device may encode the residual attributes by: determining transformed coefficients based on applying an intra transform to the residual attributes; and entropy encoding, in the bitstream, the transformed coefficients corresponding to (e.g., representing or indicating)

the residual attributes. The computing device may quantize the transformed coefficients, the quantized transformed coefficients being entropy encoded. The residual attributes may be encoded based on a prediction with lifting (pred-lift) transform scheme. The intra transform may comprise an Adaptive-DCT, a RAHT transform, or a Haar transform. The inverse intra transform comprises an inverse Adaptive-DCT (A-DCT), an inverse RAHT transform of a RAHT scheme, or an inverse Haar transform. The computing device may select the attribute reference point cloud frame from a plurality of already-coded reference point cloud frames. The computing device may encode attribute reference point cloud frame information representative of the selected attribute reference point cloud frame. The attribute reference point cloud frame is selected independently of the reference point cloud frame. The attribute reference point cloud frame information may indicate a selection of a single attribute reference point cloud frame. The single attribute reference point cloud frame and the reference point cloud frame may be the same or may not be the same. The attribute reference point cloud frame information may indicate whether the single attribute reference point cloud frame and the reference point cloud frame are the same or not. The single attribute reference point cloud frame and the geometry motion-compensated point cloud frame may be the same or may not be the same. The attribute reference point cloud frame information may indicate whether the single attribute reference point cloud frame and the geometry motion-compensated cloud frame are the same or not. The attribute reference point cloud frame information may indicate an index to a table of indices referencing the single attribute reference point cloud frame, each index referencing a specific reference point cloud frame among a plurality of already-decoded reference point cloud frames. The computing device may determine spatial regions from the decoded geometry; and select an attribute reference point cloud frame per spatial region. The attribute reference point cloud frame information may indicate whether a single attribute reference point cloud frame is selected or whether an attribute reference point cloud frame is selected per spatial region. The attribute reference point cloud frame information may indicate definition of the spatial regions. The attribute reference point cloud frame information may indicate a reference point cloud frame per spatial region. The attribute reference point cloud frame selected for a spatial region may be or may not be equal to the reference point cloud frame used for decoding the geometry of the points belonging to said spatial region. The attribute reference point cloud frame information may indicate whether an attribute reference point cloud frame selected for a spatial region equals to the reference point cloud frame used for decoding the geometry of the points belonging to the

spatial region or not. An attribute reference point cloud frame may be selected for a spatial region may be equal or may not be equal to the geometry motion-compensated point cloud frame used for decoding the geometry of the points belonging to the spatial region. The attribute reference point cloud frame information may indicate whether an attribute reference point cloud frame selected for a spatial region equals to the geometry motion-compensated point cloud frame used for decoding the geometry of the points belonging to the spatial region or not. The attribute reference point cloud frame information may indicate an index to a table of indices referencing an attribute reference point cloud frame for a spatial region, each index referencing a specific reference point cloud frame from a plurality of already decoded reference point cloud frames. The computing device may encode geometry motion vector information representative of the geometry motion vectors; and encode attribute motion vector information representative of the attribute motion vectors. The attribute motion vector information may indicate the attribute motion vectors and geometry motion vectors are associated with a same motion field structure. A motion field structure may comprise a partitioning of a 3D space encompassing the reconstructed geometry into motion units, wherein a geometry motion vector is associated with each of the motion units, and wherein the motion compensation of the reference point cloud frame is performed within each motion unit based on the associated geometry motion vector, wherein an attribute motion vector is associated with each of the motion units, and wherein the motion compensation of the attribute reference point cloud frame is performed within each motion unit based on the associated attribute motion vector. The motion units may comprise non-intersecting cuboids. The attribute motion vector information may indicate the motion field structure. The attribute motion information may indicate the attribute motion vectors and the geometry motion vectors are associated with different motion field structures. An attribute motion field structure associated with the attribute motion vectors is determined from a first partitioning of the 3D space encompassing the decoded geometry associated with a point cloud frame into attribute motion units, and wherein a geometry motion field structure associated with the geometry motion vectors is determined from a second partitioning of the 3D space encompassing the decoded geometry associated with a point cloud frame into geometry motion units. The computing device may encode the attribute motion vector information by: determining an attribute motion field structure associated with the attribute motion vectors from a geometry motion field structure associated with the geometry motion vectors; determining a motion vector residual based on differences between: attribute motion

vectors of the attribute motion field structure; and geometry motion vectors of the geometry motion field structure ; and encoding the motion vector residual as the attribute motion information. The attribute motion information may indicate a bound (e.g., threshold, range, limit, etc.) on the maximal magnitude of the geometry motion vectors and/or a bound on the attribute motion vectors. The computing device may comprise one or more processors and memory, storing instructions that, when executed by the one or more processors, perform the method described herein. A system may comprise the computing device configured to perform the described method, additional operations, and/or include additional elements; and a second computing device configured to decode the current point cloud frame. A computer-readable medium may store instructions that, when executed, cause performance of the described method, additional operations, and/or include additional elements.

[0261] A computing device may perform a method comprising multiple operations. The computing device may comprise a decoder. The computing device may motion compensate a reference point cloud frame, for a current point cloud frame, based on geometry motion vectors to determine a geometry motion-compensated point cloud frame. The computing device may motion compensate an attribute reference point cloud frame, for the point cloud frame, based on attribute motion vectors to determine an attribute motion-compensated point cloud frame. The computing device may decode, based on the geometry motion-compensated point cloud frame, a geometry associated with the point cloud frame to determine a reconstructed geometry of the point cloud frame. The computing device may decode attributes associated with the reconstructed geometry based on attributes of the attribute motion-compensated point cloud frame. The computing device may decode attributes associated with the reconstructed geometry by: determining attribute predictors of the attributes associated with the reconstructed geometry based on projecting attributes of the attribute motion-compensated point cloud frame onto the reconstructed geometry; and decoding the attributes of the reconstructed geometry based on the attribute predictors. The computing device may decode attributes associated with the reconstructed geometry by: decoding, from a bitstream, residual attributes indicating differences between the attributes of the reconstructed geometry and the attribute predictors; and determining the decoded attributes based on adding the attribute predictors and the decoded residual attributes. The computing device may decode of the residual attributes by decoding, from the bitstream, transformed coefficients corresponding to (e.g., representing or indicating) the residual attributes; and determining the residual

attributes based on applying an inverse intra transform to the decoded transformed coefficients. The computing device may dequantize the transform coefficients, the residual attributes being determined by applying the inverse intra transform to the dequantized transformed coefficients. The residual attributes may be decoded based on a prediction with lifting (pred-lift) transform scheme. The intra transform may comprise an Adaptive-DCT, a RAHT transform, or a Haar transform. The inverse intra transform comprises an inverse Adaptive-DCT (A-DCT), an inverse RAHT transform of a RAHT scheme, or an inverse Haar transform. The computing device may select the attribute reference point cloud frame from a plurality of already-coded reference point cloud frames. The computing device may decode attribute reference point cloud frame information representative of the selected attribute reference point cloud frame; and select the attribute reference point cloud frame from a plurality of already decoded reference point cloud frames according to the decoded attribute reference point cloud frame information. The attribute reference point cloud frame is selected independently of the reference point cloud frame. The attribute reference point cloud frame information may indicate a selection of a single attribute reference point cloud frame. The single attribute reference point cloud frame and the reference point cloud frame may be the same or may not be the same. The attribute reference point cloud frame information may indicate whether the single attribute reference point cloud frame and the reference point cloud frame are the same or not. The single attribute reference point cloud frame and the geometry motion-compensated point cloud frame may be the same or may not be the same. The attribute reference point cloud frame information may indicate whether the single attribute reference point cloud frame and the geometry motion-compensated cloud frame are the same or not. The attribute reference point cloud frame information may indicate an index to a table of indices referencing the single attribute reference point cloud frame, each index referencing a specific reference point cloud frame among a plurality of already-decoded reference point cloud frames. The computing device may determine spatial regions from the decoded geometry; and select an attribute reference point cloud frame per spatial region. The attribute reference point cloud frame information may indicate whether a single attribute reference point cloud frame is selected or whether an attribute reference point cloud frame is selected per spatial region. The attribute reference point cloud frame information may indicate definition of the spatial regions. The attribute reference point cloud frame information may indicate a reference point cloud frame per spatial region. The attribute reference point cloud frame selected for a spatial region may

be or may not be equal to the reference point cloud frame used for decoding the geometry of the points belonging to said spatial region. The attribute reference point cloud frame information may indicate whether an attribute reference point cloud frame selected for a spatial region equals to the reference point cloud frame used for decoding the geometry of the points belonging to the spatial region or not. An attribute reference point cloud frame may be selected for a spatial region may be equal or may not be equal to the geometry motion-compensated point cloud frame used for decoding the geometry of the points belonging to the spatial region. The attribute reference point cloud frame information may indicate whether an attribute reference point cloud frame selected for a spatial region equals to the geometry motion-compensated point cloud frame used for decoding the geometry of the points belonging to the spatial region or not. The attribute reference point cloud frame information may indicate an index to a table of indices referencing an attribute reference point cloud frame for a spatial region, each index referencing a specific reference point cloud frame from a plurality of already decoded reference point cloud frames. The computing device may decode geometry motion vector information representative of the geometry motion vectors; and decode attribute motion vector information representative of the attribute motion vectors. The attribute motion vector information may indicate the attribute motion vectors and geometry motion vectors are associated with a same motion field structure. A motion field structure may comprise a partitioning of a 3D space encompassing the reconstructed geometry into motion units, wherein a geometry motion vector is associated with each of the motion units, and wherein the motion compensation of the reference point cloud frame is performed within each motion unit based on the associated geometry motion vector, wherein an attribute motion vector is associated with each of the motion units, and wherein the motion compensation of the attribute reference point cloud frame is performed within each motion unit based on the associated attribute motion vector. The motion units may comprise non-intersecting cuboids. The attribute motion vector information may indicate the motion field structure. The attribute motion information may indicate the attribute motion vectors and the geometry motion vectors are associated with different motion field structures. An attribute motion field structure associated with the attribute motion vectors is determined from a first partitioning of the 3D space encompassing the decoded geometry associated with a point cloud frame into attribute motion units, and wherein a geometry motion field structure associated with the geometry motion vectors is determined from a second partitioning of the 3D space encompassing the decoded geometry associated with a point

cloud frame into geometry motion units. The computing device may decode the attribute motion vector information by: determining an attribute motion field structure associated with the attribute motion vectors from a geometry motion field structure associated with the geometry motion vectors; determining a motion vector residual based on differences between: attribute motion vectors of the attribute motion field structure; and geometry motion vectors of the geometry motion field structure ; and encoding the motion vector residual as the attribute motion information. The attribute motion information may indicate a bound (e.g., threshold, range, limit, etc.) on the maximal magnitude of the geometry motion vectors and/or a bound on the attribute motion vectors. The computing device may comprise one or more processors and memory, storing instructions that, when executed by the one or more processors, perform the method described herein. A system may comprise the computing device configured to perform the described method, additional operations, and/or include additional elements; and a second computing device configured to encode the current point cloud frame. A computer-readable medium may store instructions that, when executed, cause performance of the described method, additional operations, and/or include additional elements.

[0262] Hereinafter, various characteristics will be highlighted in a set of numbered clauses or paragraphs. These characteristics are not to be interpreted as being limiting on the invention or inventive concept, but are provided merely as a highlighting of some characteristics as described herein, without suggesting a particular order of importance or relevancy of such characteristics.

[0263] Clause 1. A method comprising determining, by a decoder and from a bitstream, geometry motion vectors.

[0264] Clause 2. The method of clause 1, further comprising determining, by a decoder and from a bitstream, geometry motion vectors.

[0265] Clause 3. The method of any one of clauses 1-2, further comprising determining a reconstructed geometry of a point cloud frame associated with content by decoding, based on the geometry motion vectors, a geometry associated with the point cloud frame.

[0266] Clause 4. The method of any one of clauses 1-3, further comprising decoding, based on the attribute motion vectors, attributes associated with the reconstructed geometry.

[0267] Clause 5. The method of clause 4, wherein the decoding the attributes associated with the reconstructed geometry comprises: determining, based on the attribute motion vectors, projected attributes; determining, based on the projected attributes, attribute predictors of

the attributes associated with the reconstructed geometry; and decoding, based on the attribute predictors, the attributes associated with the reconstructed geometry.

[0268] Clause 6. The method of clause 5, wherein determining the projected attributes further comprises: determining an attribute motion compensated point cloud frame; and determining the projected attributes by projecting attributes of the attribute motion compensated point cloud frame onto the reconstructed geometry.

[0269] Clause 7. The method of any one of clauses 4-6, wherein the decoding the attributes associated with the reconstructed geometry comprises: decoding residual attributes indicating differences between the attributes of the reconstructed geometry and attribute predictors associated with the reconstructed geometry; and determining based on adding the attribute predictors and the residual attributes, the attributes of the reconstructed geometry.

[0270] Clause 8. The method of any one of clauses 3-7, wherein a reference point cloud frame, for the point cloud from, is motion compensated based on the geometry motion vectors to determine the reconstructed geometry of the point cloud frame.

[0271] Clause 9. The method of clause 8, further comprising: decoding geometry motion vector information indicating the geometry motion vectors, wherein the geometry motion vectors are associated with a same motion field structure; and decoding attribute motion vector information indicating the attribute motion vectors.

[0272] Clause 10. The method of clause 9, wherein the decoding the attribute motion vector information comprise: determining, based on decoding the attribute motion vector information, a motion vector residual; and determining, based on adding the motion vector residual and geometry motion vectors of a geometry motion field structure, attribute motion vectors of an attribute motion field structure.

[0273] Clause 11. The method of any one of clauses 3-11, further comprising: decoding attribute reference point cloud frame information indicating an attribute reference point cloud frame; and selecting, based on the attribute reference point cloud frame information, the attribute reference point cloud frame from a plurality of already-decoded reference point cloud frames.

[0274] Clause 12. A computing device comprising one or more processors and memory storing instructions that, when executed by the one or more processors, cause the computing device to perform the method of any one of clauses 1-11.

[0275] Clause 13. A system comprising: a computing device configured to perform the method of any one of clauses 1-11, and a second computing device configured to encode the point cloud frame.

[0276] Clause 14. A computer-readable medium storing instructions that, when executed, cause performance of the method of any one of clauses 1-11.

[0277] Clause 15. A method comprising determining, by an encoder and based on a point cloud frame associated with content, geometry motion vectors.

[0278] Clause 16. The method of clause 15, further comprising: determining, based on the point cloud frame, attribute motion vectors.

[0279] Clause 17. The method of any one of clauses 15-16, further comprising: encoding, based on the geometry motion vectors, a geometry associated with the point cloud frame.

[0280] Clause 18. The method of any one of clauses 15-17, further comprising: encoding, based on the attribute motion vectors, attributes associated with a reconstructed geometry of the point cloud frame.

[0281] Clause 19. The method of clause 18, wherein the encoding the attributes associated with the reconstructed geometry of the point cloud frame comprises: determining, based on the attribute motion vectors, projected attributes; determining, based on the projected attributes, attribute predictors of the attributes associated with the reconstructed geometry; and encoding, based on the attribute predictors, the attributes associated with the reconstructed geometry.

[0282] Clause 20. The method of any one of clauses 18-19, further comprising: determining projected attributes based on motion compensating an already-coded reference point cloud frame, wherein the already-coded reference point cloud frame is motion compensated by the attribute motion vectors.

[0283] Clause 21. The method of clause 20, further comprising: determining the attribute motion vectors based on differences between: the attributes associated with the reconstructed geometry of the point cloud frame; and attributes of a geometry, of the already-coded reference point cloud frame, adjusted by the attribute motion vectors.

[0284] Clause 22. The method of any one of clauses 18-21, further comprising: determining, based on mapping attributes of the geometry associated with the point cloud frame to the reconstructed geometry, the attributes associated with the reconstructed geometry of the point cloud frame.

[0285] Clause 23. The method of clause 22, wherein the attributes associated with the reconstructed geometry of the point cloud frame comprise colors.

[0286] Clause 24. The method of clause 23, further comprising: determining, based on recoloring, the mapped attributes of the geometry associated with the point cloud frame.

[0287] Clause 25. The method of any one of clauses 18-24, wherein a reference point cloud frame, for the point cloud frame, is motion compensated based on geometry motion vectors to determine the reconstructed geometry of the point cloud frame.

[0288] Clause 26. The method of clause 25, further comprising: encoding geometry motion vector information indicating the geometry motion vectors, wherein the geometry motion vectors are associated with a same motion field structure; and encoding attribute motion vector information indicating the attribute motion vectors.

[0289] Clause 27. The method of clause 26, wherein the encoding the attribute motion vector information comprises: determining a motion vector residual based on differences between the attribute motion vectors and the geometry motion vectors; and encoding the motion vector residual as the attribute motion vector information.

[0290] Clause 28. A computing device comprising one or more processors and memory storing instructions that, when executed by the one or more processors, cause the computing device to perform the method of any one of clauses 15-27.

[0291] Clause 29. A system comprising: a computing device configured to perform the method of any one of clauses 15-27, and a second computing device configured to decode the point cloud frame.

[0292] Clause 30. A computer-readable medium storing instructions that, when executed, cause performance of the method of any one of clauses 15-27.

[0293] Clause 31. A method comprising determining, by a decoder and from a bitstream, attribute motion vectors and geometry motion vectors.

[0294] Clause 32. The method of clause 31, further comprising: determining a reconstructed geometry of a point cloud frame associated with content, by decoding, based on the geometry motion vectors, a geometry associated with the point cloud frame.

[0295] Clause 33. The method of any one of clauses 31-32, further comprising: determining, based on the attribute motion vectors, projected attributes.

[0296] Clause 34. The method of clause 33, further comprising: determining, based on the projected attributes, attribute predictors of attributes associated with the reconstructed geometry.

[0297] Clause 35. The method of clause 34, further comprising: decoding, from the bitstream, residual attributes indicating differences between the attributes associated with the reconstructed geometry and the attribute predictors.

[0298] Clause 36. The method of clause 35, wherein the decoding the residual attributes comprises: decoding, from the bitstream, transformed coefficients indicating the residual attributes; and determining the residual attributes based on applying an inverse intra transform to the decoded transformed coefficients.

[0299] Clause 37. The method of any one of clauses 35-36, wherein the residual attributes are decoded based on a prediction with lifting transform scheme.

[0300] Clause 38. The method of any one of clauses 35-37, wherein the determining the attribute predictors further comprises: smoothing the projected attributes; and determining, based on the smoothed projected attributes, the attribute predictors.

[0301] Clause 39. A computing device comprising one or more processors and memory storing instructions that, when executed by the one or more processors, cause the computing device to perform the method of any one of clauses 31-38.

[0302] Clause 40. A system comprising: a computing device configured to perform the method of any one of clauses 31-38, and a second computing device configured to encode the point cloud frame.

[0303] Clause 41. A computer-readable medium storing instructions that, when executed, cause performance of the method of any one of clauses 31-38.

[0304] Clause 42. A method comprising motion compensating a reference point cloud frame, for a current point cloud frame, based on geometry motion vectors to determine a geometry motion-compensated point cloud frame.

[0305] Clause 43. The method of clause 42, further comprising: motion compensating an attribute reference point cloud frame, for the point cloud frame, based on attribute motion vectors to determine an attribute motion-compensated point cloud frame.

[0306] Clause 44. The method of any one of clauses 42-43, further comprising encoding a geometry associated with the point cloud frame based on the geometry motion-compensated point cloud frame.

[0307] Clause 45. The method of any one of clauses 43-44, further comprising encoding attributes associated with a reconstructed geometry of the point cloud frame based on attributes of the attribute motion-compensated point cloud frame.

[0308] Clause 46. The method of clause 45, wherein the encoding the attributes associated with the reconstructed geometry of the point cloud frame comprises: determining attribute predictors of the attributes associated with the reconstructed geometry based on projecting attributes of the attribute motion-compensated point cloud frame onto the reconstructed

geometry; and encoding the attributes of the reconstructed geometry based on the attribute predictors.

[0309] Clause 47. The method of any one of clauses 44-46, further comprising: determining the attributes associated with the reconstructed geometry of the point cloud frame based on mapping attributes of a geometry of the point cloud frame to the reconstructed geometry.

[0310] Clause 48. The method of clause 47, further comprising: encoding the geometry of the point cloud frame; and determining the reconstructed geometry based on decoding the encoded geometry.

[0311] Clause 47. The method of any one of clauses 47-48, wherein the attributes are colors and the mapped attributes are determined based on recoloring.

[0312] Clause 48. The method of any one of clauses 47-48, wherein the mapped attributes for each point of the reconstructed geometry is determined based on a nearest neighbor search of nearest points, from the geometry of the point cloud frame, to the point in the reconstructed geometry.

[0313] Clause 49. The method of any one of clauses 46-48, wherein the encoding of attributes comprises: determining residual attributes based on differences between the attributes of the reconstructed geometry and the attribute predictors; and encoding the residual attributes.

[0314] Clause 50. The method of clause 49, wherein the encoding of the residual attributes comprises: determining transformed coefficients based on applying an intra transform to the residual attributes; and entropy encoding, in the bitstream, the transformed coefficients corresponding to (e.g., representing or indicating) the residual attributes.

[0315] Clause 51. The method of clause 50, further comprising quantizing the transformed coefficients, the quantized transformed coefficients being entropy encoded.

[0316] Clause 52. A computing device comprising one or more processors and memory storing instructions that, when executed by the one or more processors, cause the computing device to perform the method of any one of clauses 42-51.

[0317] Clause 53. A system comprising: a computing device configured to perform the method of any one of clauses 42-51, and a second computing device configured to decode the current point cloud frame.

[0318] Clause 54. A computer-readable medium storing instructions that, when executed, cause performance of the method of any one of clauses 42-51.

[0319] Clause 55. A method comprising motion compensating a reference point cloud frame, for a current point cloud frame, based on geometry motion vectors to determine a geometry motion-compensated point cloud frame.

[0320] Clause 56. The method of clause 55, further comprising: motion compensating an attribute reference point cloud frame, for the point cloud frame, based on attribute motion vectors to determine an attribute motion-compensated point cloud frame.

[0321] Clause 57. The method of any one of clauses 55-56, further comprising decoding, based on the geometry motion-compensated point cloud frame, a geometry associated with the point cloud frame to determine a reconstructed geometry of the point cloud frame.

[0322] Clause 58. The method of any one of clauses 56-57, further comprising decoding attributes associated with the reconstructed geometry based on attributes of the attribute motion-compensated point cloud frame.

[0323] Clause 59. The method of clause 58, wherein the decoding the attributes associated with the reconstructed geometry comprises: determining attribute predictors of the attributes associated with the reconstructed geometry based on projecting attributes of the attribute motion-compensated point cloud frame onto the reconstructed geometry; and decoding the attributes of the reconstructed geometry based on the attribute predictors.

[0324] Clause 60. The method of clause 59, wherein the decoding the attributes associated with the reconstructed geometry comprises: decoding, from a bitstream, residual attributes indicating differences between the attributes of the reconstructed geometry and the attribute predictors; and determining the decoded attributes based on adding the attribute predictors and the decoded residual attributes.

[0325] Clause 61. The method of clause 60, wherein the decoding of the residual attributes comprises: entropy decoding, from the bitstream, transformed coefficients corresponding to (e.g., representing or indicating) the residual attributes; and determining the residual attributes based on applying an inverse intra transform to the decoded transformed coefficients.

[0326] Clause 62. The method of clause 61, further comprising: dequantizing the transform coefficients, the residual attributes being determined by applying the inverse intra transform to the dequantized transformed coefficients.

[0327] Clause 63. The method of any one of clauses 60-62, wherein the residual attributes are decoded based on a prediction with lifting (pred-lift) transform scheme.

[0328] Clause 64. The method of any one of clauses 60-63, wherein the intra transform comprises an Adaptive-DCT, a RAHT transform, or a Haar transform.

[0329] Clause 65. The method of any one of clauses 60-63, wherein the inverse intra transform comprises an inverse Adaptive-DCT (A-DCT), an inverse RAHT transform of a RAHT scheme, or an inverse Haar transform.

[0330] Clause 66. The method of any one of clauses 56-65, wherein the geometry motion compensated point cloud frame is determined from an already-coded reference point cloud frame.

[0331] Clause 67. The method of clause 66, wherein the geometry motion compensated point cloud frame is determined based on motion compensating the already-coded reference point cloud frame.

[0332] Clause 68. The method of any one of clauses 66-67, wherein the geometry motion compensated point cloud frame is determined based on motion compensating the already-coded reference point cloud frame.

[0333] Clause 69. The method of any one of clauses 66-68, further comprising encoding the geometry motion vector and/or an indication (e.g., index or ID) of the already-coded reference point cloud frame.

[0334] Clause 70. The method of any one of clauses 66-69, further comprising determining the geometry motion vector based on differences between the reconstructed geometry and a geometry, of the already-coded reference point cloud frame, adjusted by the geometry motion vector.

[0335] Clause 71. The method of any one of clauses 56-70, wherein the attribute motion compensated point cloud frame is determined from an already-coded reference point cloud frame.

[0336] Clause 72. The method of clause 71, wherein the attribute motion compensated point cloud frame is determined based on motion compensating the already-coded reference point cloud frame.

[0337] Clause 73. The method of any one of clauses 71-72, wherein the already-coded reference point cloud frame is motion compensated by an attribute motion vector.

[0338] Clause 74. The method of any one of clauses 71-73, further comprising encoding the attribute motion vector and/or an indication (e.g., index or ID) of the already-coded reference point cloud frame.

[0339] Clause 75. The method of any one of clauses 71-74, further comprising determining the attribute motion vector based on differences between the attributes associated with the reconstructed geometry and the attributes of a geometry, of the already-coded reference point cloud frame, adjusted by the attribute motion vector.

[0340] Clause 76. The method of any one of clauses 59-75, wherein the determining the attribute predictors comprises smoothing the projected attributes, the attribute predictors being determined from the smoothed projected attributes.

[0341] Clause 77. The method of any one of clauses 59-76, wherein the attribute predictors comprises a respective attribute predictor, for each respective vertex of vertices of the reconstructed geometry, that is based on a projected attribute, of the projected attributes, corresponding to the vertex.

[0342] Clause 78. The method of any one of clauses 58-77, further comprising: decoding attribute reference point cloud frame information representative of the selected attribute reference point cloud frame; and selecting the attribute reference point cloud frame from a plurality of already decoded reference point cloud frames according to the decoded attribute reference point cloud frame information.

[0343] Clause 79. The method of clause 78, wherein the attribute reference point cloud frame is selected independently of the reference point cloud frame.

[0344] Clause 80. The method of any one of clauses 78-79, wherein the attribute reference point cloud frame information indicates a selection of a single attribute reference point cloud frame.

[0345] Clause 81. The method of clause 80, wherein the single attribute reference point cloud frame and the reference point cloud frame are not the same.

[0346] Clause 82. The method of clause 80, wherein the single attribute reference point cloud frame and the reference point cloud frame are the same.

[0347] Clause 83. The method of any one of clauses 80-82, wherein the attribute reference point cloud frame information indicates whether the single attribute reference point cloud frame and the reference point cloud frame are the same or not.

[0348] Clause 84. The method of clause 80, wherein the single attribute reference point cloud frame and the geometry motion-compensated point cloud frame are not the same.

[0349] Clause 85. The method of clause 80, wherein the single attribute reference point cloud frame and the geometry motion-compensated point cloud frame are the same.

[0350] Clause 86. The method of clause 80, wherein the attribute reference point cloud frame information indicates whether the single attribute reference point cloud frame and the geometry motion-compensated cloud frame are the same or not.

[0351] Clause 87. The method of any one of clauses 80-86, wherein the attribute reference point cloud frame information indicates an index to a table of indices referencing the single

attribute reference point cloud frame, each index referencing a specific reference point cloud frame among a plurality of already-decoded reference point cloud frames.

[0352] Clause 88. The method of any one of clauses 80-87, further comprising: determining spatial regions from the decoded geometry; and selecting an attribute reference point cloud frame per spatial region.

[0353] Clause 89. The method of clause 88, wherein the attribute reference point cloud frame information indicates whether a single attribute reference point cloud frame is selected or whether an attribute reference point cloud frame is selected per spatial region.

[0354] Clause 90. The method of any one of clauses 88-89, wherein the attribute reference point cloud frame information indicates definition of the spatial regions.

[0355] Clause 91. The method of any one of clauses 88-90, wherein the attribute reference point cloud frame information indicates a reference point cloud frame per spatial region.

[0356] Clause 92. The method of clause 91, wherein the attribute reference point cloud frame selected for a spatial region is not equal to the reference point cloud frame used for decoding the geometry of the points belonging to said spatial region.

[0357] Clause 93. The method of any one of clauses 88-92, wherein the attribute reference point cloud frame selected for a spatial region is equal to the reference point cloud frame used for decoding the geometry of the points belonging to said spatial region.

[0358] Clause 94. The method of any one of clauses 88-93, wherein the attribute reference point cloud frame information indicates whether an attribute reference point cloud frame selected for a spatial region equals to the reference point cloud frame used for decoding the geometry of the points belonging to the spatial region or not.

[0359] Clause 95. The method of any one of clauses 88-94, wherein an attribute reference point cloud frame selected for a spatial region equals to the geometry motion-compensated point cloud frame used for decoding the geometry of the points belonging to the spatial region.

[0360] Clause 96. The method of any one of clauses 88-94, wherein an attribute reference point cloud frame selected for a spatial region is not equal to the geometry motion-compensated point cloud frame used for decoding the geometry of the points belonging to the spatial region.

[0361] Clause 97. The method of any one of clauses 88-96, wherein the attribute reference point cloud frame information indicates whether an attribute reference point cloud frame selected for a spatial region equals to the geometry motion-compensated point cloud frame used for decoding the geometry of the points belonging to the spatial region or not.

[0362] Clause 98. The method of any one of clauses 88-97, wherein the attribute reference point cloud frame information indicates an index to a table of indices referencing an attribute reference point cloud frame for a spatial region, each index referencing a specific reference point cloud frame from a plurality of already decoded reference point cloud frames.

[0363] Clause 99. The method of any one of clauses 58-98, further comprising: decoding geometry motion vector information representative of the geometry motion vectors; and decoding attribute motion vector information representative of the attribute motion vectors.

[0364] Clause 100. The method of clause 99, wherein the attribute motion vector information indicates the attribute motion vectors and geometry motion vectors are associated with a same motion field structure.

[0365] Clause 101. The method of clause 100, wherein a motion field structure comprises a partitioning of a 3D space encompassing the reconstructed geometry into motion units, wherein a geometry motion vector is associated with each of the motion units, and wherein the motion compensation of the reference point cloud frame is performed within each motion unit based on the associated geometry motion vector, wherein an attribute motion vector is associated with each of the motion units, and wherein the motion compensation of the attribute reference point cloud frame is performed within each motion unit based on the associated attribute motion vector.

[0366] Clause 102. The method of clause 101, wherein the motion units comprise non-intersecting cuboids.

[0367] Clause 103. The method of any one of clauses 101-102, wherein the attribute motion vector information indicates the motion field structure.

[0368] Clause 104. The method of any one of clauses 99-103, wherein the attribute motion information indicates the attribute motion vectors and the geometry motion vectors are associated with different motion field structures.

[0369] Clause 105. The method of clause 104, wherein an attribute motion field structure associated with the attribute motion vectors is determined from a first partitioning of the 3D space encompassing the decoded geometry associated with a point cloud frame into attribute motion units, and wherein a geometry motion field structure associated with the geometry motion vectors is determined from a second partitioning of the 3D space encompassing the decoded geometry associated with a point cloud frame into geometry motion units.

[0370] Clause 106. The method of clause 105, wherein the decoding the attribute motion vector information comprises: ermining a motion vector residual based on decoding the attribute motion information; and determining attribute motion vectors of an attribute motion field structure based on adding the motion vector residual and geometry motion vectors of a geometry motion field structure.

[0371] Clause 107. The method of any one of clauses 99-106, wherein the attribute motion information indicates a bound (e.g., threshold, range, limit, etc.) on the maximal magnitude of the geometry motion vectors and/or a bound on the attribute motion vectors.

[0372] Clause 108. A computing device comprising one or more processors and memory storing instructions that, when executed by the one or more processors, cause the computing device to perform the method of any one of clauses 55-107.

[0373] Clause 109. A system comprising: a computing device configured to perform the method of any one of clauses 55-107, and a second computing device configured to encode the current point cloud frame.

[0374] Clause 110. A computer-readable medium storing instructions that, when executed, cause performance of the method of any one of clauses 55-107.

[0375] One or more examples herein may be described as a process which may be depicted as a flowchart, a flow diagram, a data flow diagram, a structure diagram, and/or a block diagram. Although a flowchart may describe operations as a sequential process, one or more of the operations may be performed in parallel or concurrently. The order of the operations shown may be re-arranged. A process may be terminated when its operations are completed, but could have additional steps not shown in a figure. A process may correspond to a method, a function, a procedure, a subroutine, a subprogram, etc. If a process corresponds to a function, its termination may correspond to a return of the function to the calling function or the main function.

[0376] Operations described herein may be implemented by hardware, software, firmware, middleware, microcode, hardware description languages, or any combination thereof. When implemented in software, firmware, middleware or microcode, the program code or code segments to perform the necessary tasks (e.g., a computer-program product) may be stored in a computer-readable or machine-readable medium. A processor(s) may perform the necessary tasks. Features of the disclosure may be implemented in hardware using, for example, hardware components such as application-specific integrated circuits (ASICs) and gate arrays. Implementation of a hardware state machine to perform the functions described herein will also be apparent to persons skilled in the art.

[0377] One or more features described herein may be implemented in a computer-usable data and/or computer-executable instructions, such as in one or more program modules, executed by one or more computers or other devices. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types when executed by a processor in a computer or other data processing device. The computer executable instructions may be stored on one or more computer readable media such as a hard disk, optical disk, removable storage media, solid state memory, RAM, etc. The functionality of the program modules may be combined or distributed as desired. The functionality may be implemented in whole or in part in firmware or hardware equivalents such as integrated circuits, field programmable gate arrays (FPGA), and the like. Particular data structures may be used to more effectively implement one or more features described herein, and such data structures are contemplated within the scope of computer executable instructions and computer-usable data described herein. Computer-readable medium may comprise, but is not limited to, portable or non-portable storage devices, optical storage devices, and various other mediums capable of storing, containing, or carrying instruction(s) and/or data. A computer-readable medium may include a non-transitory medium in which data can be stored and that does not include carrier waves and/or transitory electronic signals propagating wirelessly or over wired connections. Examples of a non-transitory medium may include, but are not limited to, a magnetic disk or tape, optical storage media such as compact disk (CD) or digital versatile disk (DVD), flash memory, memory or memory devices. A computer-readable medium may have stored thereon code and/or machine-executable instructions that may represent a procedure, a function, a subprogram, a program, a routine, a subroutine, a module, a software package, a class, or any combination of instructions, data structures, or program statements. A code segment may be coupled to another code segment or a hardware circuit by passing and/or receiving information, data, arguments, parameters, or memory contents. Information, arguments, parameters, data, etc. may be passed, forwarded, or transmitted via any suitable means including memory sharing, message passing, token passing, network transmission, or the like.

[0378] A non-transitory tangible computer readable media may comprise instructions executable by one or more processors configured to cause operations described herein. An article of manufacture may comprise a non-transitory tangible computer readable machine-accessible medium having instructions encoded thereon for enabling programmable

hardware to cause a device (e.g., an encoder, a decoder, a transmitter, a receiver, and the like) to allow operations described herein. The device, or one or more devices such as in a system, may include one or more processors, memory, interfaces, and/or the like.

[0379] Communications described herein may be determined, generated, sent, and/or received using any quantity of messages, information elements, fields, parameters, values, indications, information, bits, and/or the like. While one or more examples may be described herein using any of the terms/phrases message, information element, field, parameter, value, indication, information, bit(s), and/or the like, one skilled in the art understands that such communications may be performed using any one or more of these terms, including other such terms. For example, one or more parameters, fields, and/or information elements (IEs), may comprise one or more information objects, values, and/or any other information. An information object may comprise one or more other objects. At least some (or all) parameters, fields, IEs, and/or the like may be used and can be interchangeable depending on the context. If a meaning or definition is given, such meaning or definition controls.

[0380] One or more elements in examples described herein may be implemented as modules. A module may be an element that performs a defined function and/or that has a defined interface to other elements. The modules may be implemented in hardware, software in combination with hardware, firmware, wetware (e.g., hardware with a biological element) or a combination thereof, all of which may be behaviorally equivalent. For example, modules may be implemented as a software routine written in a computer language configured to be executed by a hardware machine (such as C, C++, Fortran, Java, Basic, Matlab or the like) or a modeling/simulation program such as Simulink, Stateflow, GNU Octave, or LabVIEWMathScript. Additionally or alternatively, it may be possible to implement modules using physical hardware that incorporates discrete or programmable analog, digital and/or quantum hardware. Examples of programmable hardware may comprise: computers, microcontrollers, microprocessors, application-specific integrated circuits (ASICs); field programmable gate arrays (FPGAs); and/or complex programmable logic devices (CPLDs). Computers, microcontrollers and/or microprocessors may be programmed using languages such as assembly, C, C++ or the like. FPGAs, ASICs and CPLDs are often programmed using hardware description languages (HDL), such as VHSIC hardware description language (VHDL) or Verilog, which may configure connections between internal hardware modules with lesser

functionality on a programmable device. The above-mentioned technologies may be used in combination to achieve the result of a functional module.

[0381] One or more of the operations described herein may be conditional. For example, one or more operations may be performed if certain criteria are met, such as in computing device, a communication device, an encoder, a decoder, a network, a combination of the above, and/or the like. Example criteria may be based on one or more conditions such as device configurations, traffic load, initial system set up, packet sizes, traffic characteristics, a combination of the above, and/or the like. If the one or more criteria are met, various examples may be used. It may be possible to implement any portion of the examples described herein in any order and based on any condition.

[0382] Although examples are described above, features and/or steps of those examples may be combined, divided, omitted, rearranged, revised, and/or augmented in any desired manner. Various alterations, modifications, and improvements will readily occur to those skilled in the art. Such alterations, modifications, and improvements are intended to be part of this description, though not expressly stated herein, and are intended to be within the spirit and scope of the descriptions herein. Accordingly, the foregoing description is by way of example only, and is not limiting.

PCT CLAIMS

1. A method comprising:

determining, by a decoder and from a bitstream, geometry motion vectors;

determining, from the bitstream, attribute motion vectors;

determining a reconstructed geometry of a point cloud frame associated with content by decoding, based on the geometry motion vectors, a geometry associated with the point cloud frame; and

decoding, based on the attribute motion vectors, attributes associated with the reconstructed geometry.

2. The method of claim 1, wherein the decoding the attributes associated with the reconstructed geometry comprises:

determining, based on the attribute motion vectors, projected attributes;

determining, based on the projected attributes, attribute predictors of the attributes associated with the reconstructed geometry; and

decoding, based on the attribute predictors, the attributes associated with the reconstructed geometry.

3. The method of claim 2, wherein determining the projected attributes comprises:

determining the projected attributes based on motion compensating an already-coded reference point cloud frame, wherein the already-coded reference point cloud frame is motion compensated by the attribute motion vectors.

4. The method of claim 3, wherein determining the projected attributes further comprises:

determining an attribute motion compensated point cloud frame; and

determining the projected attributes by projecting attributes of the attribute motion compensated point cloud frame onto the reconstructed geometry.

5. The method of any one of claims 1 to 4, wherein the decoding the attributes associated with the reconstructed geometry comprises:

decoding residual attributes indicating differences between the attributes of the reconstructed geometry and attribute predictors associated with the reconstructed geometry; and

determining based on adding the attribute predictors and the residual attributes, the attributes of the reconstructed geometry.

6. The method of any one of claims 1 to 5, wherein a reference point cloud frame, for the point cloud from, is motion compensated based on the geometry motion vectors to determine the reconstructed geometry of the point cloud frame, the method further comprising:

decoding geometry motion vector information indicating the geometry motion vectors, wherein the geometry motion vectors are associated with a same motion field structure; and

decoding attribute motion vector information indicating the attribute motion vectors.

7. The method of claim 6, wherein the decoding the attribute motion vector information comprises:

determining, based on decoding the attribute motion vector information, a motion vector residual; and

determining, based on adding the motion vector residual and geometry motion vectors of a geometry motion field structure, attribute motion vectors of an attribute motion field structure.

8. The method of any one of claims 1 to 7, further comprising:

decoding attribute reference point cloud frame information indicating an attribute reference point cloud frame; and

selecting, based on the attribute reference point cloud frame information, the attribute reference point cloud frame from a plurality of already-decoded reference point cloud frames.

9. A method comprising:

determining, by a decoder and from a bitstream, attribute motion vectors and geometry motion vectors;

determining a reconstructed geometry of a point cloud frame associated with content, by decoding, based on the geometry motion vectors, a geometry associated with the point cloud frame;

determining, based on the attribute motion vectors, projected attributes;

determining, based on the projected attributes, attribute predictors of attributes associated with the reconstructed geometry; and

decoding, from the bitstream, residual attributes indicating differences between the attributes associated with the reconstructed geometry and the attribute predictors.

10. The method of claim 9, wherein the decoding the residual attributes comprises:

decoding, from the bitstream, transformed coefficients indicating the residual attributes; and

determining the residual attributes based on applying an inverse intra transform to the decoded transformed coefficients.

11. The method of any one of claims 9 to 10, wherein the residual attributes are decoded based on a prediction with lifting transform scheme.

12. The method of any one of claims 9 to 11, wherein the determining the attribute predictors further comprises:

smoothing the projected attributes; and

determining, based on the smoothed projected attributes, the attribute predictors.

13. A computing device comprising: one or more processors; and memory storing instructions that, when executed by the one or more processors, cause the computing device to perform:

the method of any one of claims 1 to 8; or

the method of any one of claims 9 to 12.

14. A system comprising:

a computing device configured to perform:

the method of any one of claims 1 to 8; or

the method of any one of claims 9 to 12; and

a second computing device configured to encode the point cloud frame.

15. A computer-readable medium storing instructions that, when executed, cause performance of:

the method of any one of claims 1 to 8; or

the method of any one of claims 9 to 12.

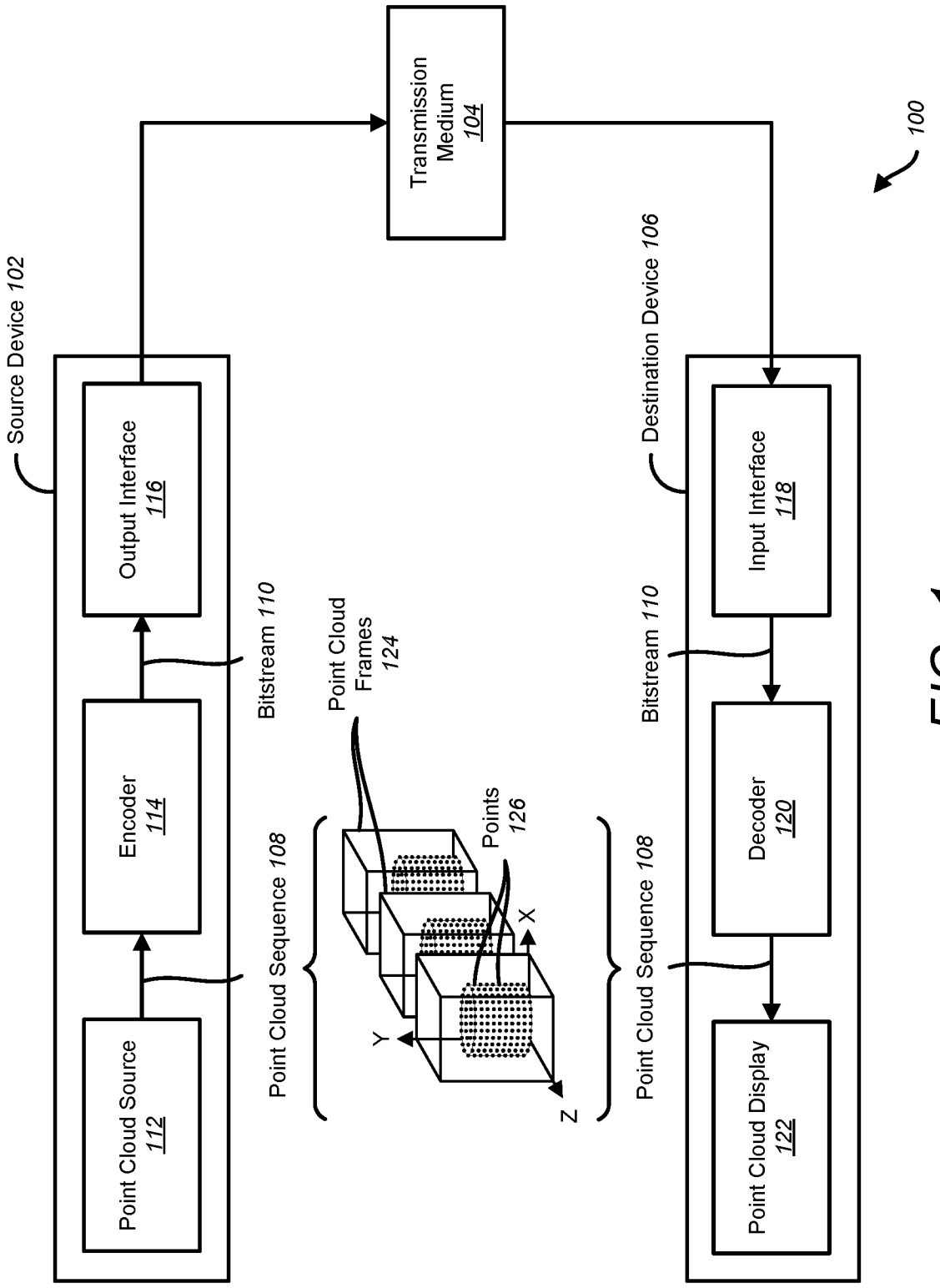
Source Device 102
Point Cloud Source 112
Encoder 114
Output Interface 116
Bitstream 110
Point Cloud Sequence 108
Point Cloud Frames 124
Points 126
X
Y
Z
Transmission Medium 104
Destination Device 106
Input Interface 118
Decoder 120
Point Cloud Display 122
100

FIG. 1

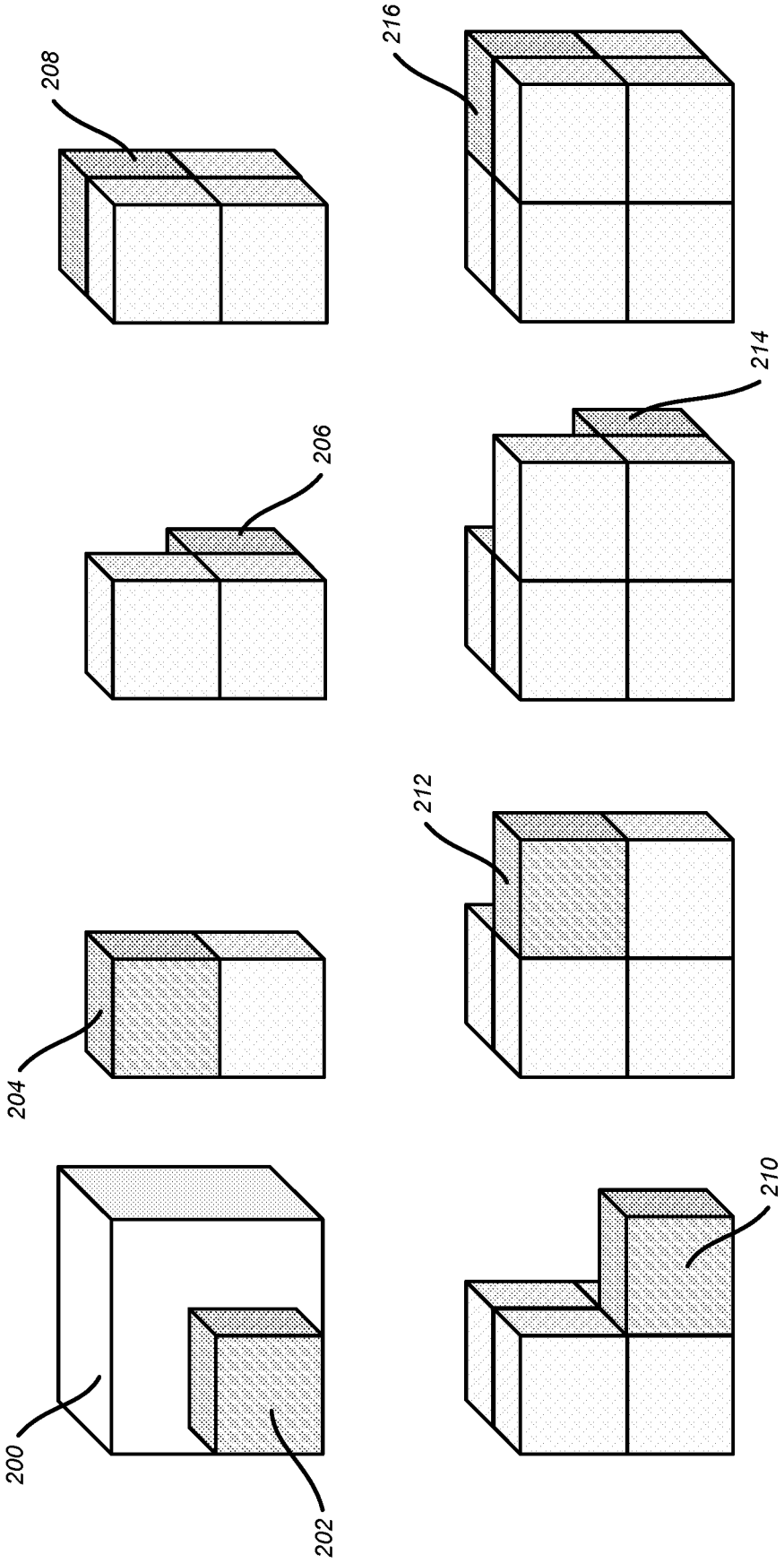

FIG. 2

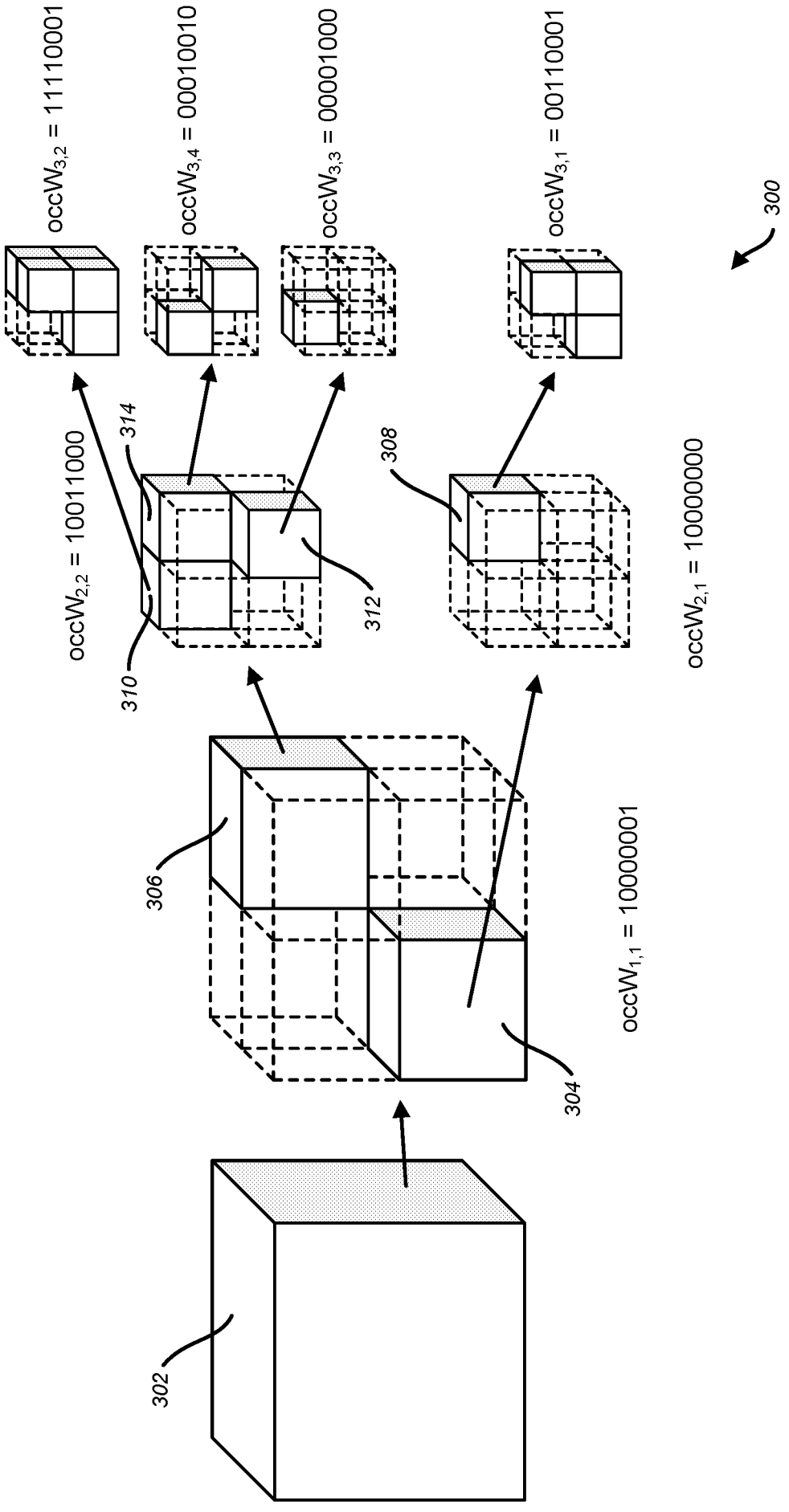
occW3,2 = 11110001
occW3,4 = 00010010
occW3,3 = 00001000
occW3,1 = 00110001
300
occW2,2 = 10011000
314
310
312
308
occW2,1 = 10000000
306
occW1,1 = 10000001
304
302

FIG. 3

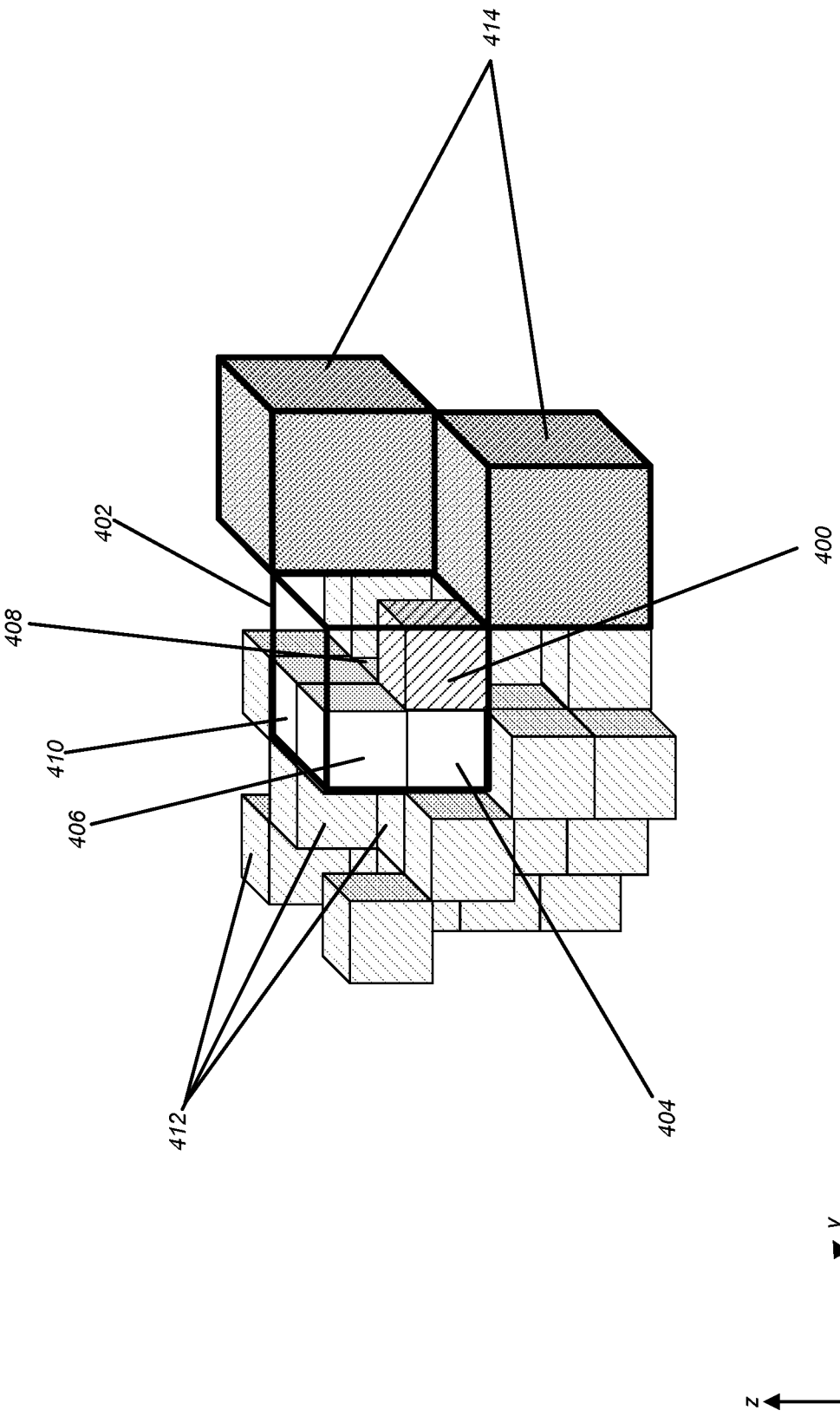

FIG. 4

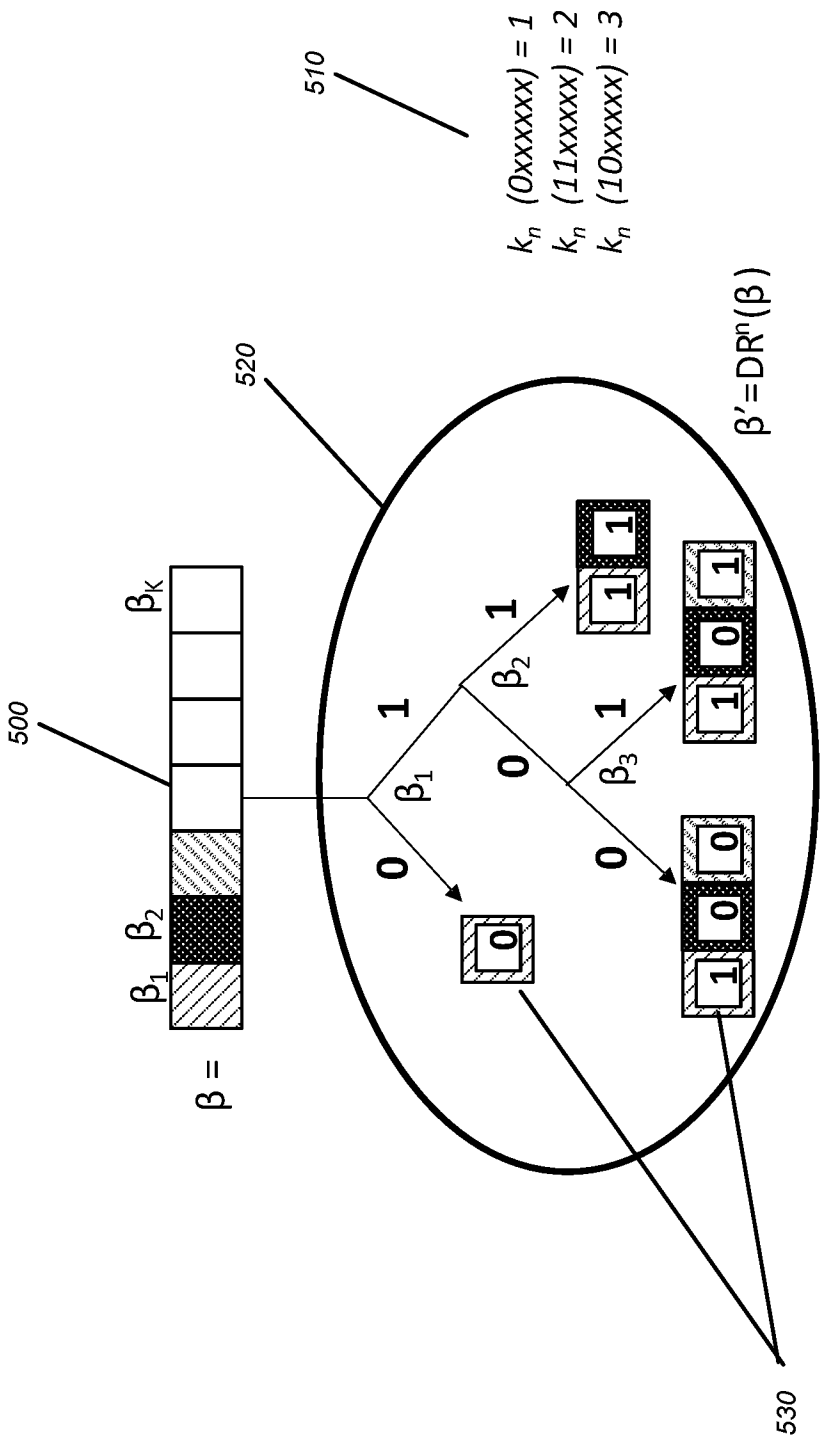
510
k_n (0xxxxxx) = 1
k_n (11xxxxx) = 2
k_n (10xxxxx) = 3
520
β'=DR^n(β)
500
β =
β_K
β_2
β_1
β_1
β_2
β_3
0
1
530

FIG. 5

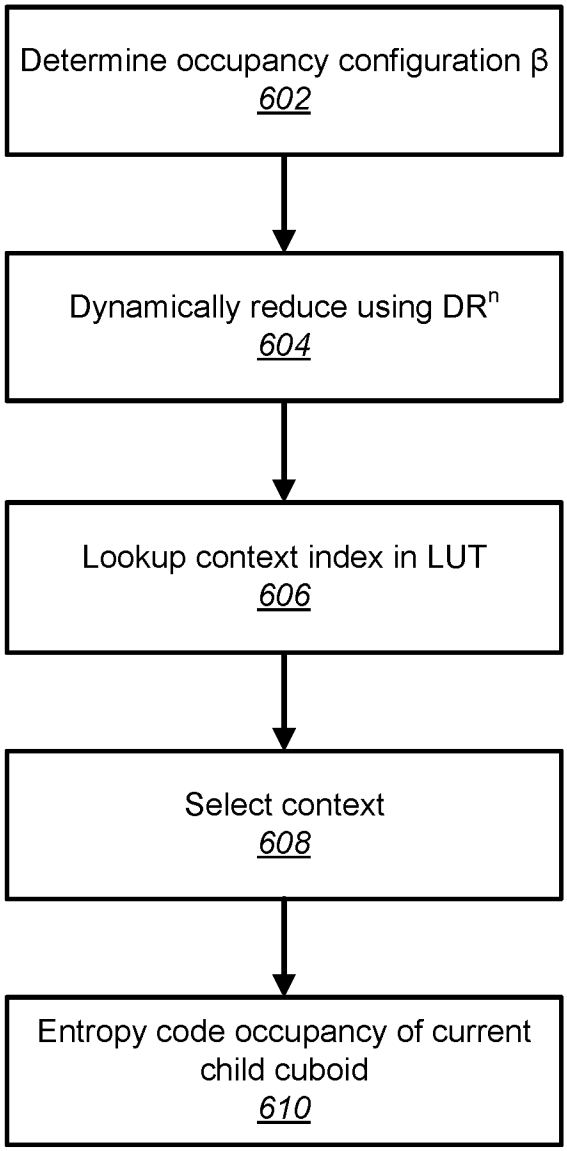
Determine occupancy configuration β
602
Dynamically reduce using DR^n
604
Lookup context index in LUT
606
Select context
608
Entropy code occupancy of current child cuboid
610

FIG. 6

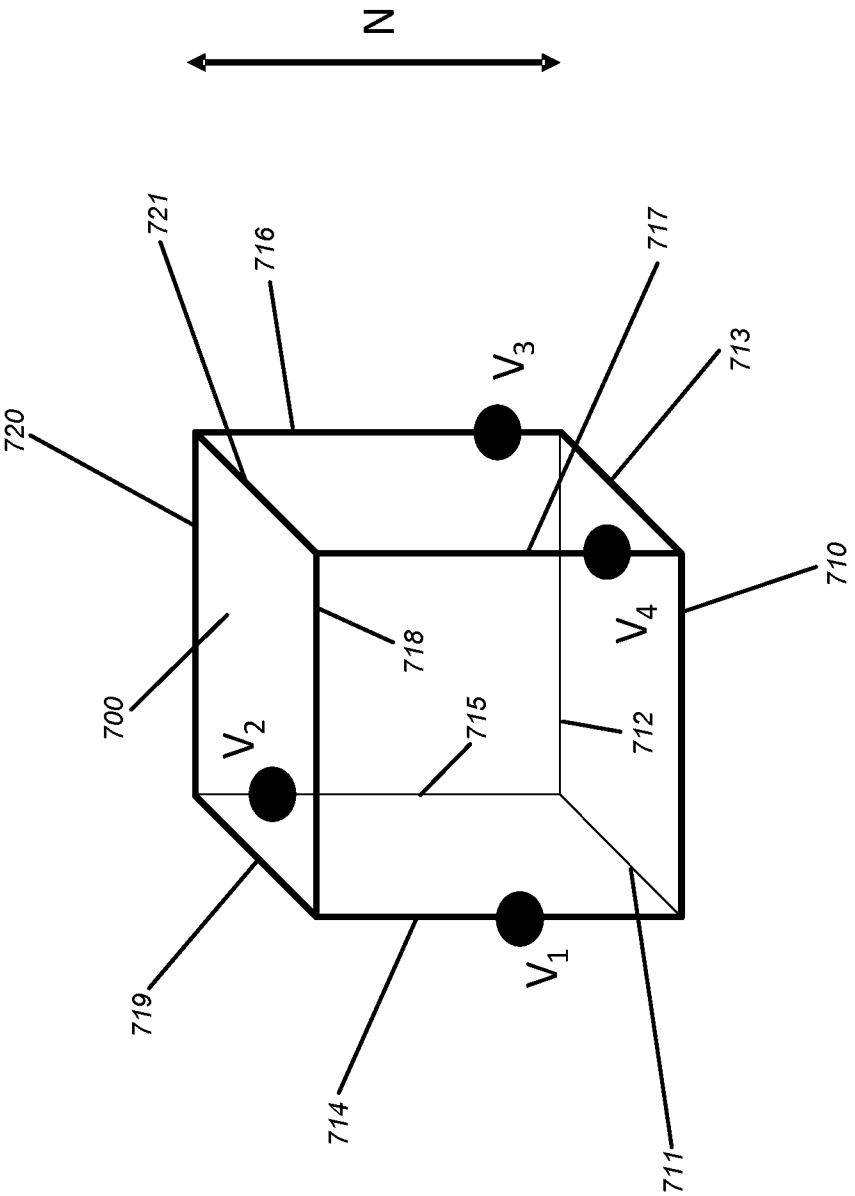

FIG. 7

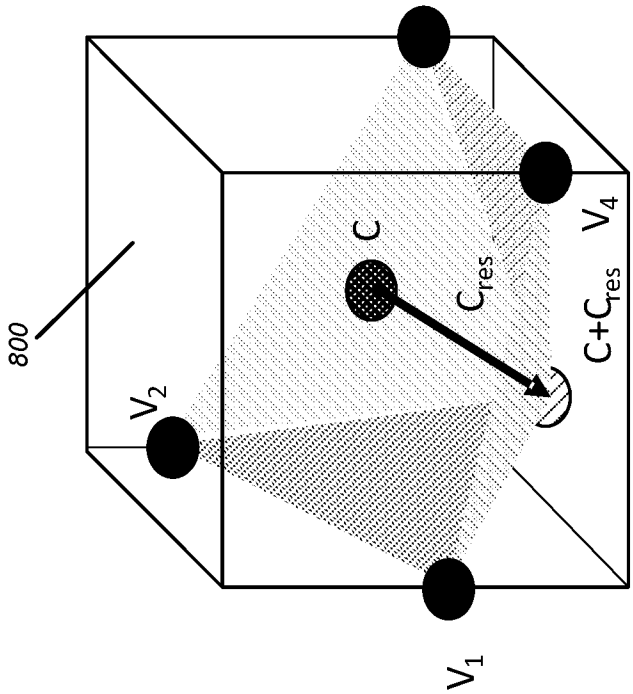

FIG. 8B

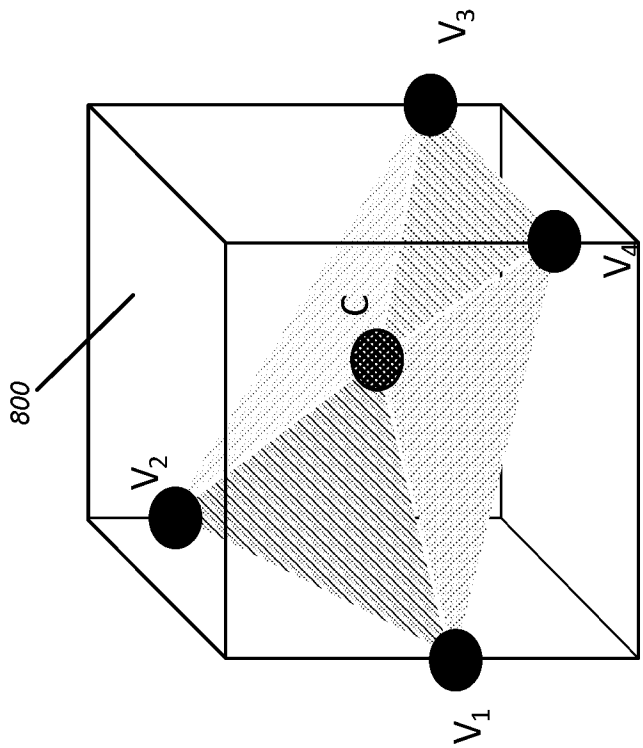

FIG. 8A

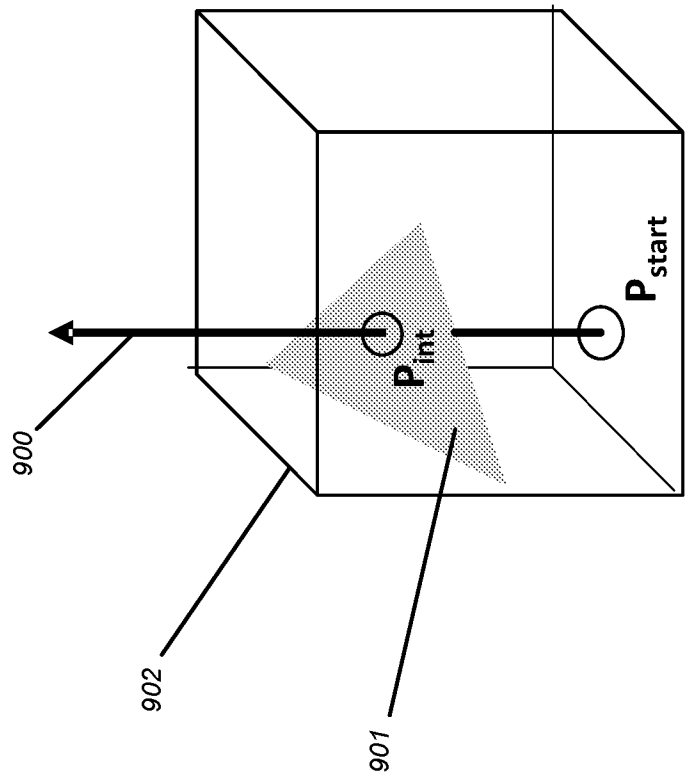

FIG. 9

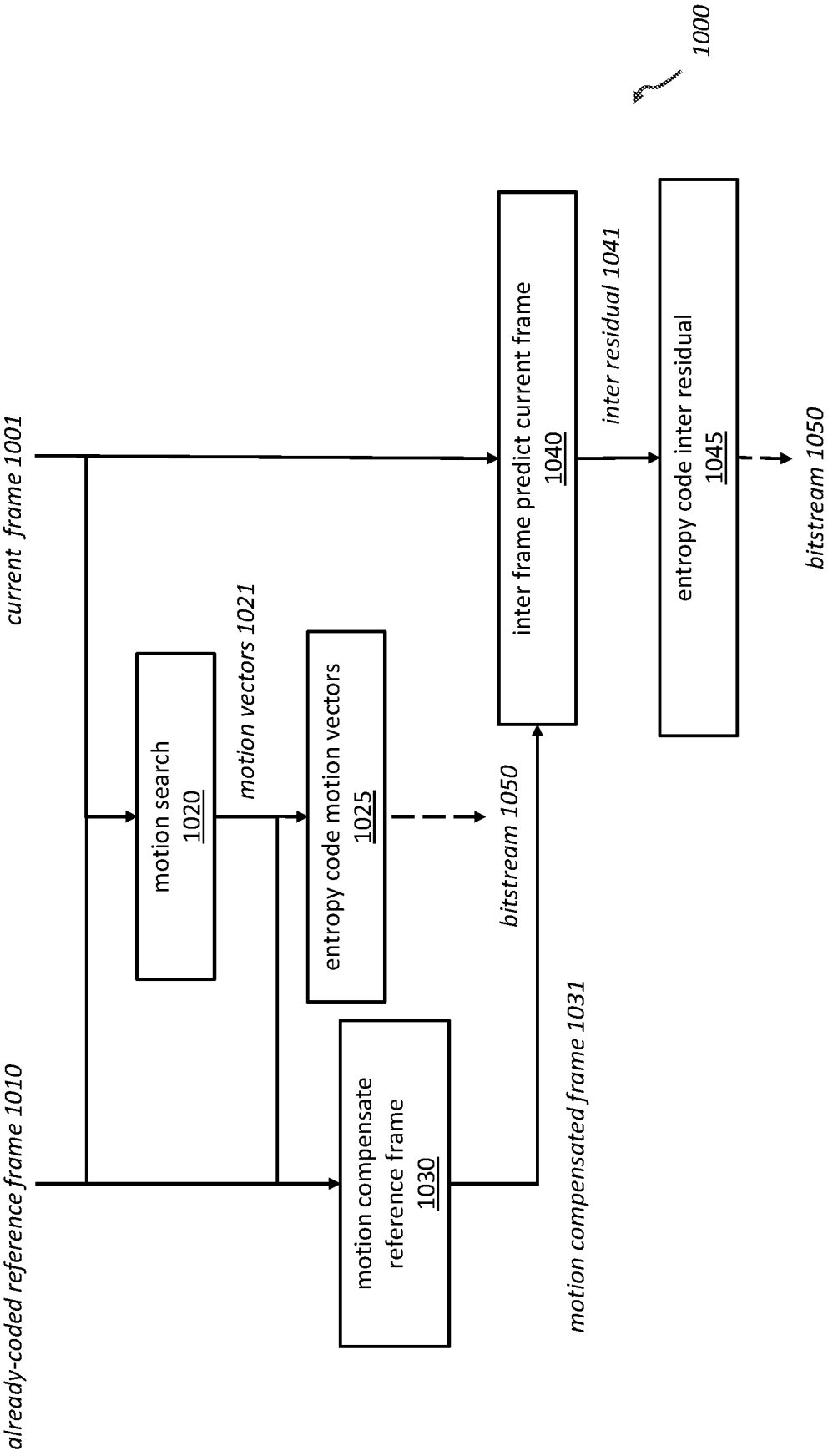
1000
current frame 1001
already-coded reference frame 1010
motion search 1020
motion vectors 1021
entropy code motion vectors 1025
bitstream 1050
motion compensate reference frame 1030
motion compensated frame 1031
inter frame predict current frame 1040
inter residual 1041
entropy code inter residual 1045
bitstream 1050

FIG. 10

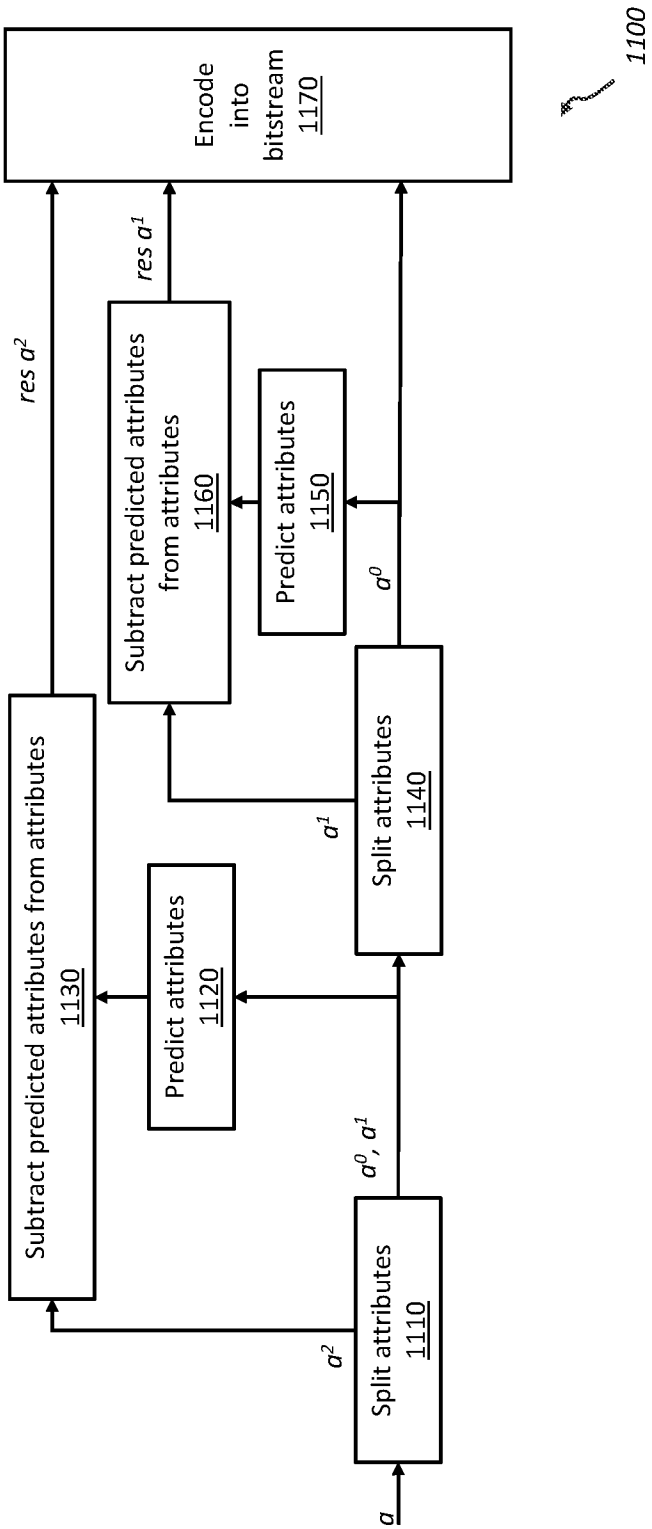
1100
Encode into bitstream 1170
res a1
res a2
Subtract predicted attributes from attributes 1160
Predict attributes 1150
a0
Subtract predicted attributes from attributes 1130
Predict attributes 1120
a1
Split attributes 1140
a0, a1
a2
Split attributes 1110
a

FIG. 11

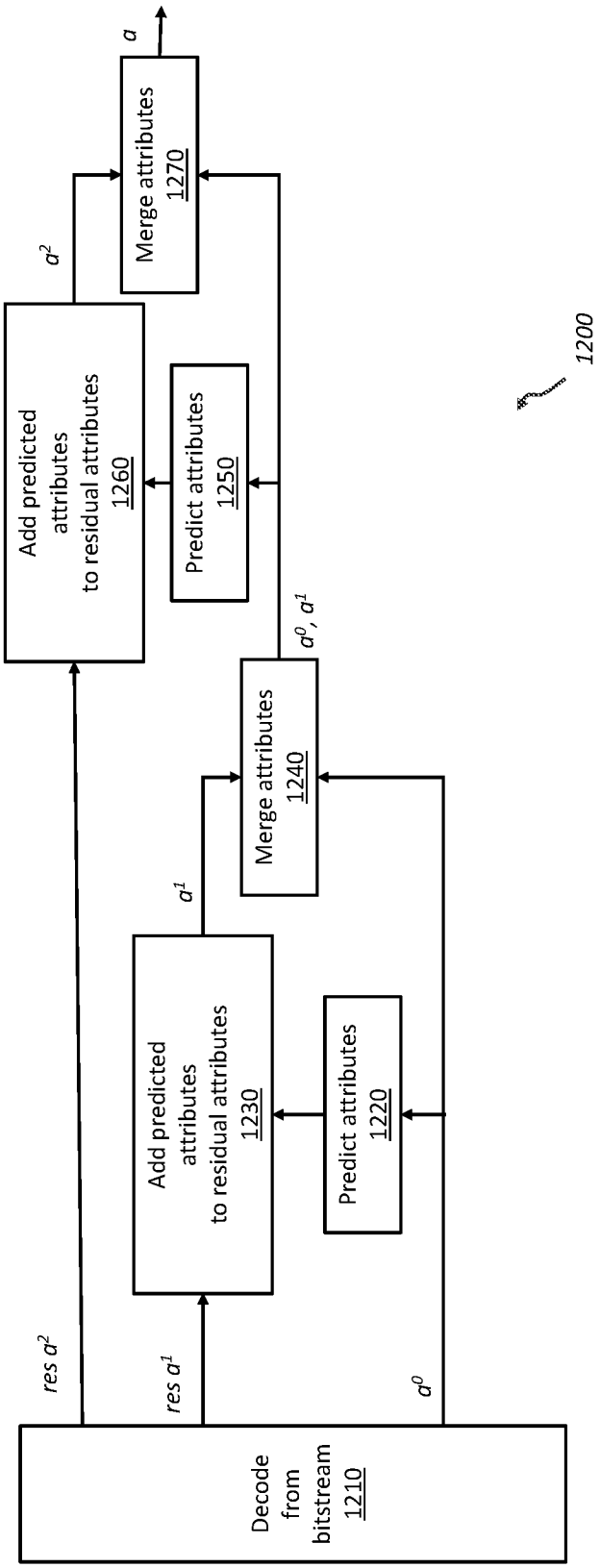
Decode from bitstream 1210
res a2
res a1
a0
Predict attributes 1220
Add predicted attributes to residual attributes 1230
a1
Merge attributes 1240
a0, a1
Predict attributes 1250
Add predicted attributes to residual attributes 1260
a2
Merge attributes 1270
a
1200

FIG. 12

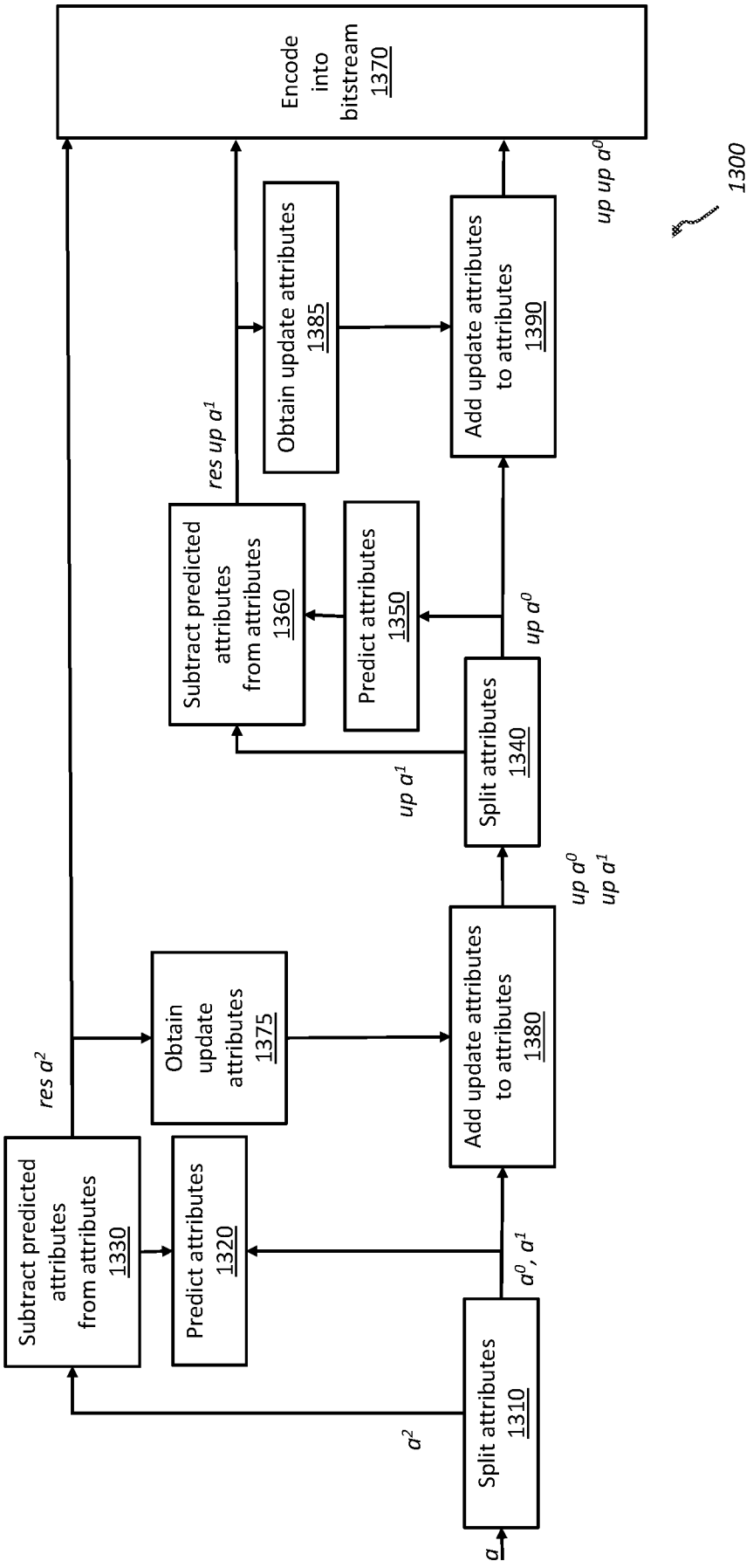
Encode into bitstream 1370
1300
up up a^0
Add update attributes to attributes 1390
Obtain update attributes 1385
res up a^1
Subtract predicted attributes from attributes 1360
Predict attributes 1350
up a^0
up a^1
Split attributes 1340
up a^0
up a^1
Add update attributes to attributes 1380
Obtain update attributes 1375
res a^2
Subtract predicted attributes from attributes 1330
Predict attributes 1320
a^0, a^1
a^2
Split attributes 1310
a

FIG. 13

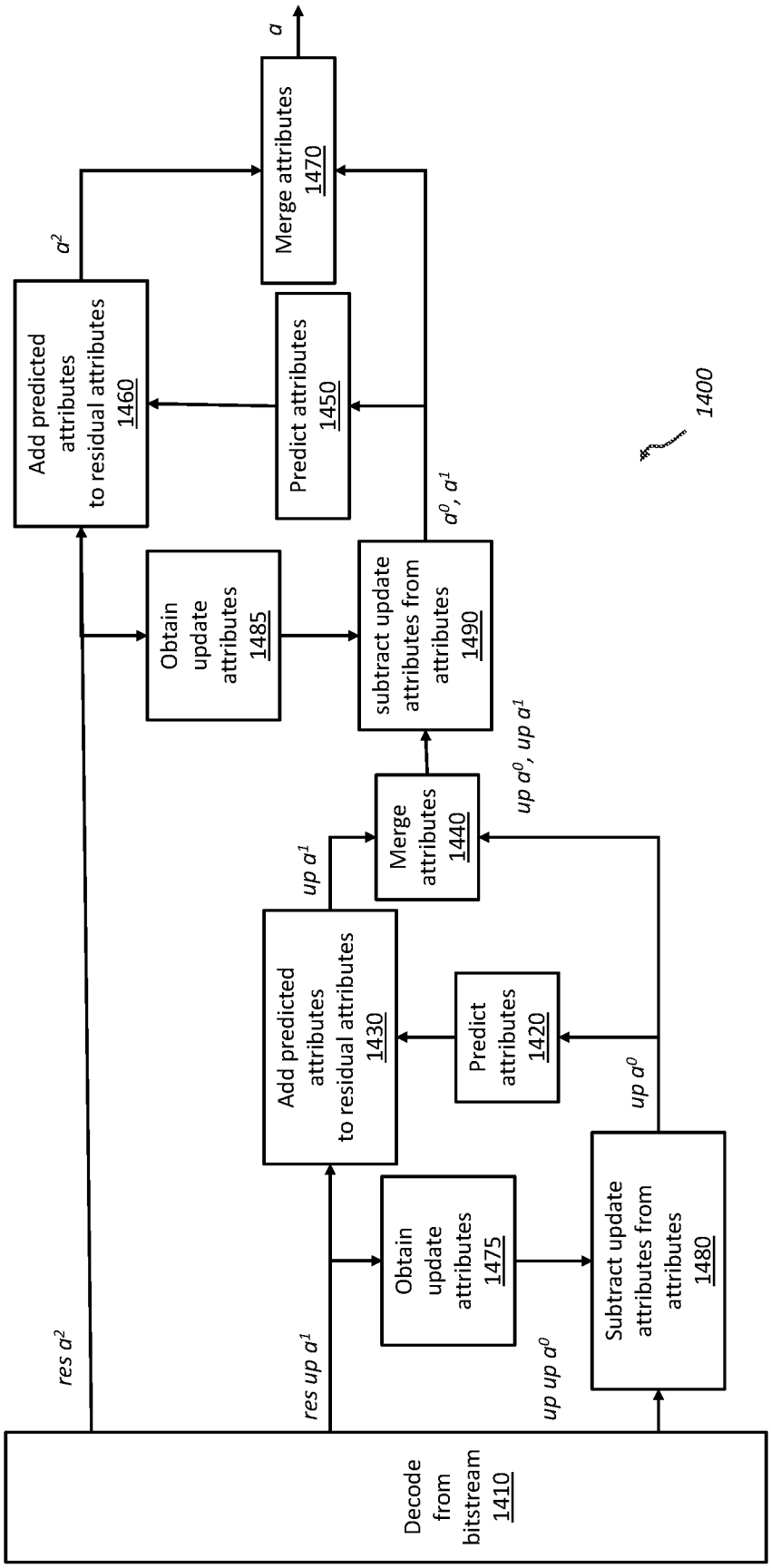
Decode from bitstream 1410
res a2
res up a1
up up a0
Obtain update attributes 1475
Subtract update attributes from attributes 1480
Add predicted attributes to residual attributes 1430
Predict attributes 1420
up a0
up a1
Merge attributes 1440
up a0, up a1
Obtain update attributes 1485
subtract update attributes from attributes 1490
a0, a1
Add predicted attributes to residual attributes 1460
Predict attributes 1450
a2
Merge attributes 1470
a
1400

FIG. 14

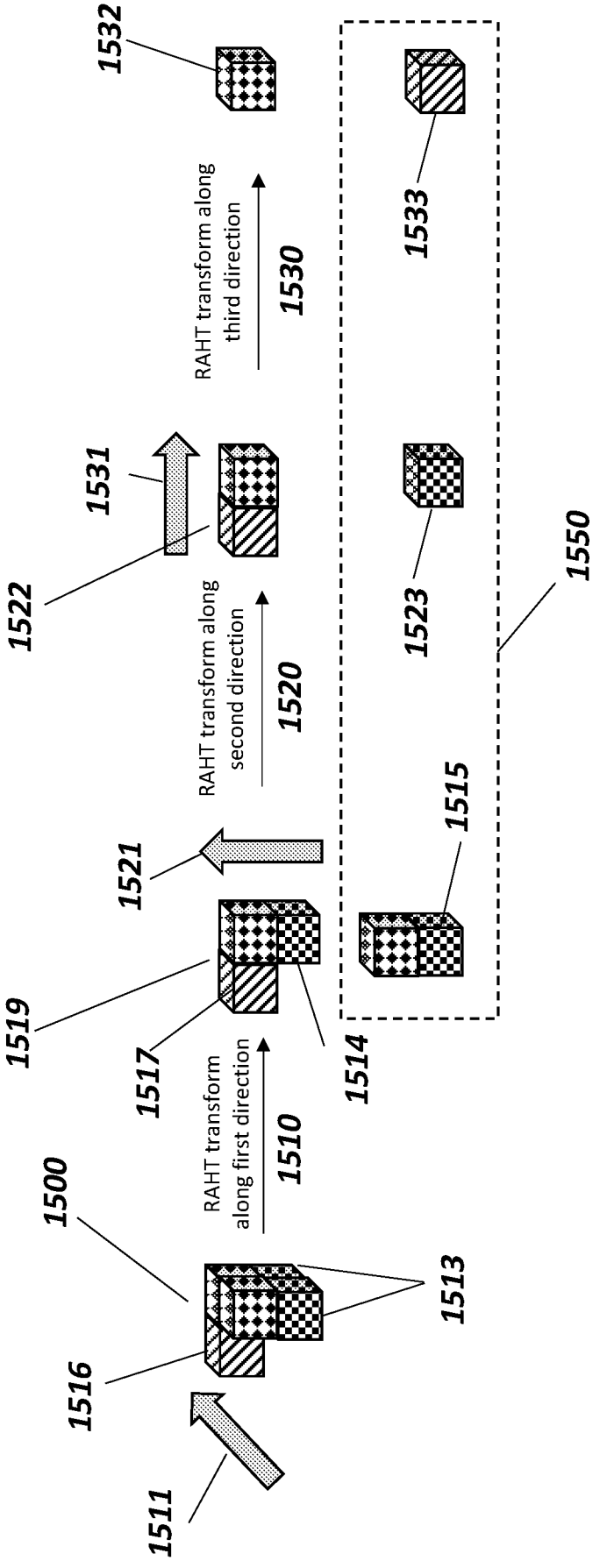
1532
1533
RAHT transform along third direction
1530
1531
1522
1523
1550
RAHT transform along second direction
1520
1521
1515
1519
1517
1514
RAHT transform along first direction
1510
1500
1513
1516
1511

FIG. 15

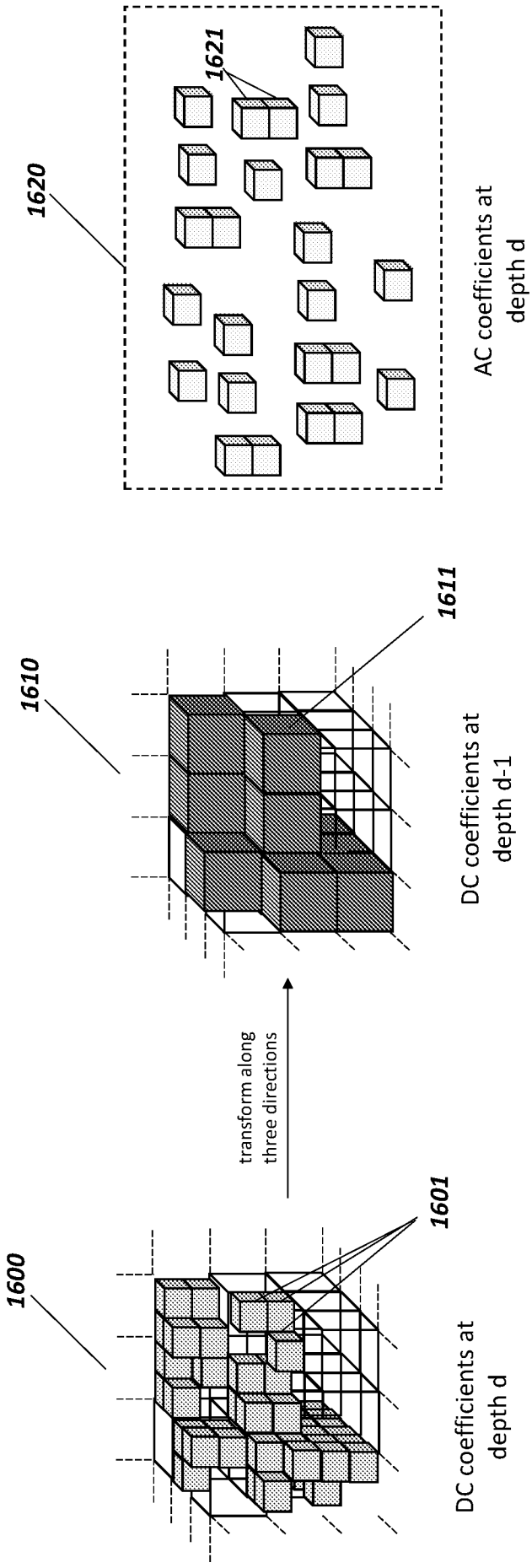
1620
1621
AC coefficients at depth d
1610
1611
DC coefficients at depth d-1
transform along three directions
1600
1601
DC coefficients at depth d

FIG. 16

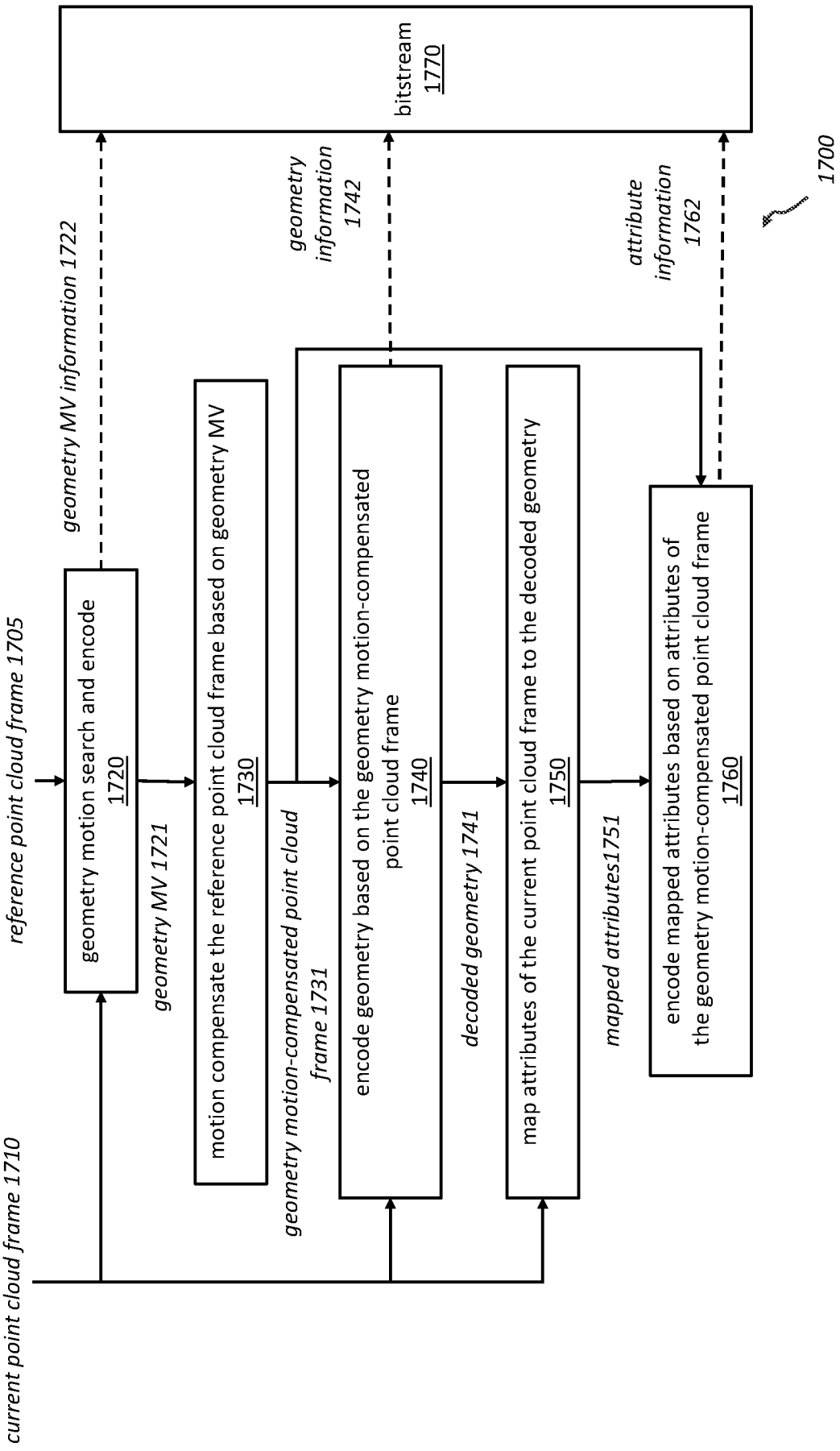
bitstream 1770
1700
geometry MV information 1722
geometry information 1742
attribute information 1762
reference point cloud frame 1705
geometry motion search and encode 1720
geometry MV 1721
motion compensate the reference point cloud frame based on geometry MV 1730
geometry motion-compensated point cloud frame 1731
encode geometry based on the geometry motion-compensated point cloud frame 1740
decoded geometry 1741
map attributes of the current point cloud frame to the decoded geometry 1750
mapped attributes1751
encode mapped attributes based on attributes of the geometry motion-compensated point cloud frame 1760
current point cloud frame 1710

FIG. 17

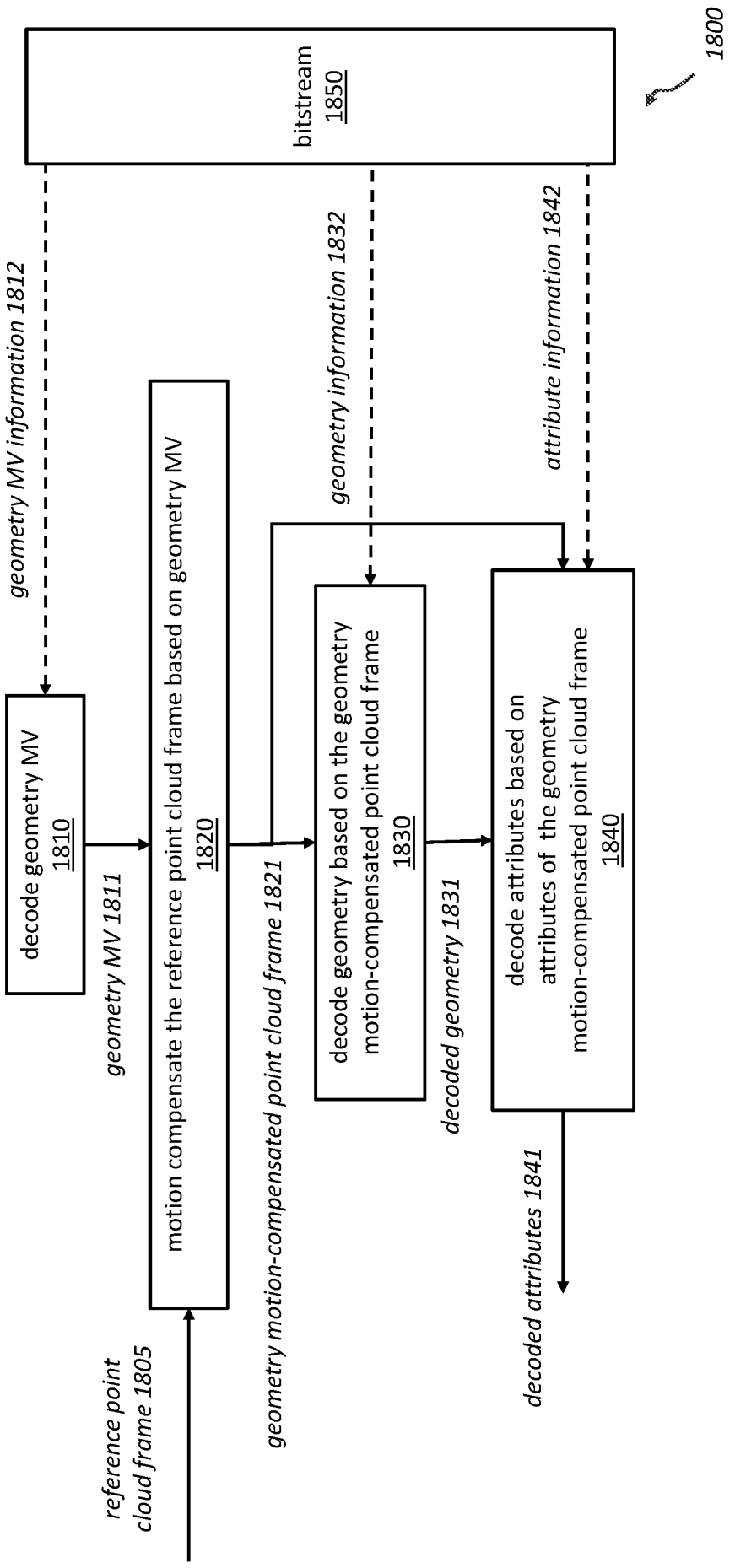
1800
bitstream
1850
geometry MV information 1812
geometry information 1832
attribute information 1842
decode geometry MV
1810
geometry MV 1811
motion compensate the reference point cloud frame based on geometry MV
1820
geometry motion-compensated point cloud frame 1821
decode geometry based on the geometry motion-compensated point cloud frame
1830
decoded geometry 1831
decode attributes based on attributes of the geometry motion-compensated point cloud frame
1840
decoded attributes 1841
reference point cloud frame 1805

FIG. 18

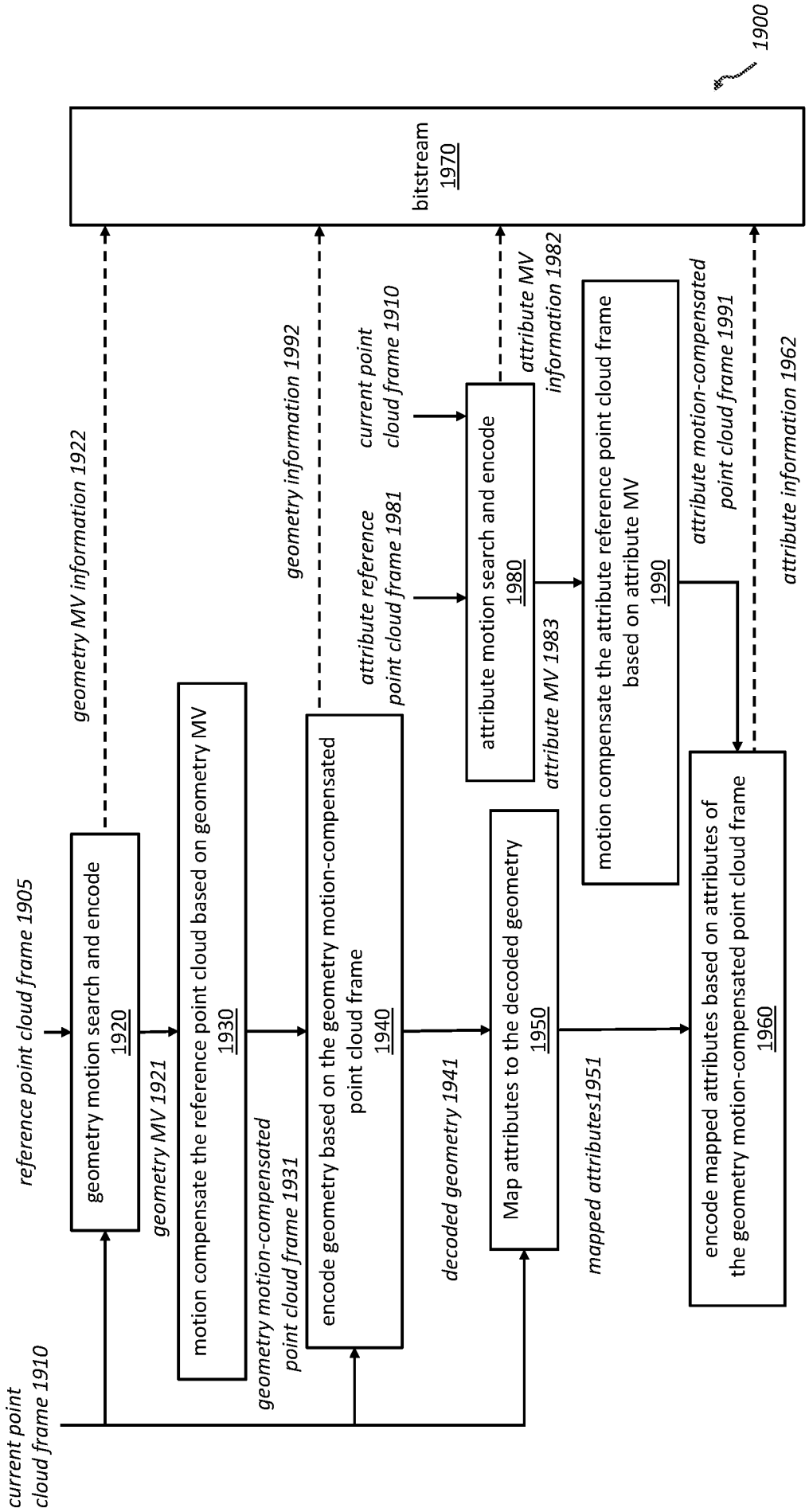
1900
bitstream 1970
geometry MV information 1922
geometry information 1992
attribute MV information 1982
attribute information 1962
current point cloud frame 1910
attribute reference point cloud frame 1981
attribute MV 1983
reference point cloud frame 1905
geometry motion search and encode 1920
geometry MV 1921
motion compensate the reference point cloud based on geometry MV 1930
geometry motion-compensated point cloud frame 1931
encode geometry based on the geometry motion-compensated point cloud frame 1940
decoded geometry 1941
Map attributes to the decoded geometry 1950
mapped attributes1951
attribute motion search and encode 1980
motion compensate the attribute reference point cloud frame based on attribute MV 1990
attribute motion-compensated point cloud frame 1991
encode mapped attributes based on attributes of the geometry motion-compensated point cloud frame 1960
current point cloud frame 1910

FIG. 19

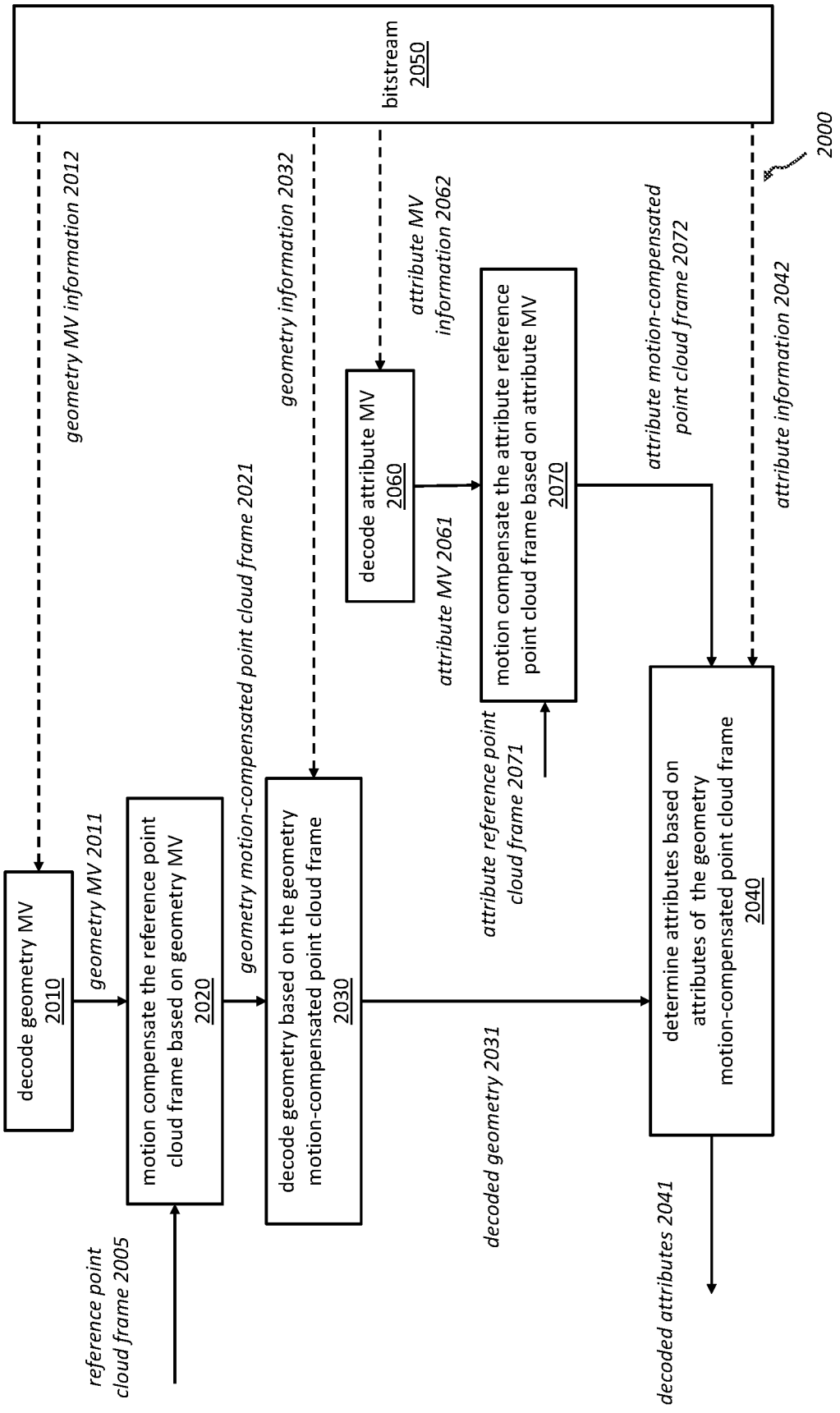
bitstream 2050
2000
geometry MV information 2012
geometry information 2032
attribute MV information 2062
attribute information 2042
decode geometry MV 2010
geometry MV 2011
motion compensate the reference point cloud frame based on geometry MV 2020
geometry motion-compensated point cloud frame 2021
decode geometry based on the geometry motion-compensated point cloud frame 2030
decode attribute MV 2060
attribute MV 2061
motion compensate the attribute reference point cloud frame based on attribute MV 2070
attribute motion-compensated point cloud frame 2072
attribute reference point cloud frame 2071
determine attributes based on attributes of the geometry motion-compensated point cloud frame 2040
reference point cloud frame 2005
decoded geometry 2031
decoded attributes 2041

FIG. 20

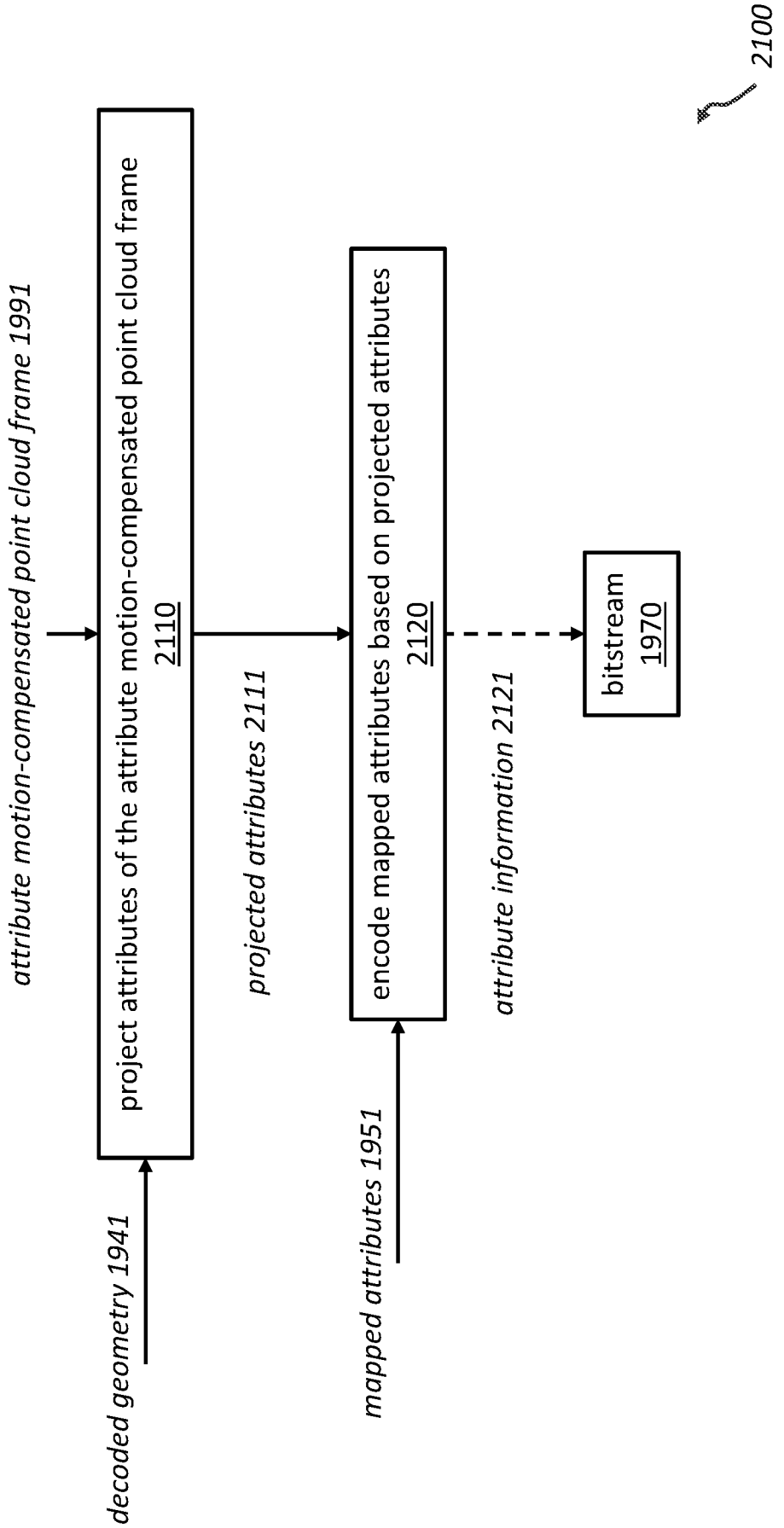
2100
attribute motion-compensated point cloud frame 1991
project attributes of the attribute motion-compensated point cloud frame
2110
projected attributes 2111
encode mapped attributes based on projected attributes
2120
attribute information 2121
bitstream
1970
decoded geometry 1941
mapped attributes 1951

FIG. 21

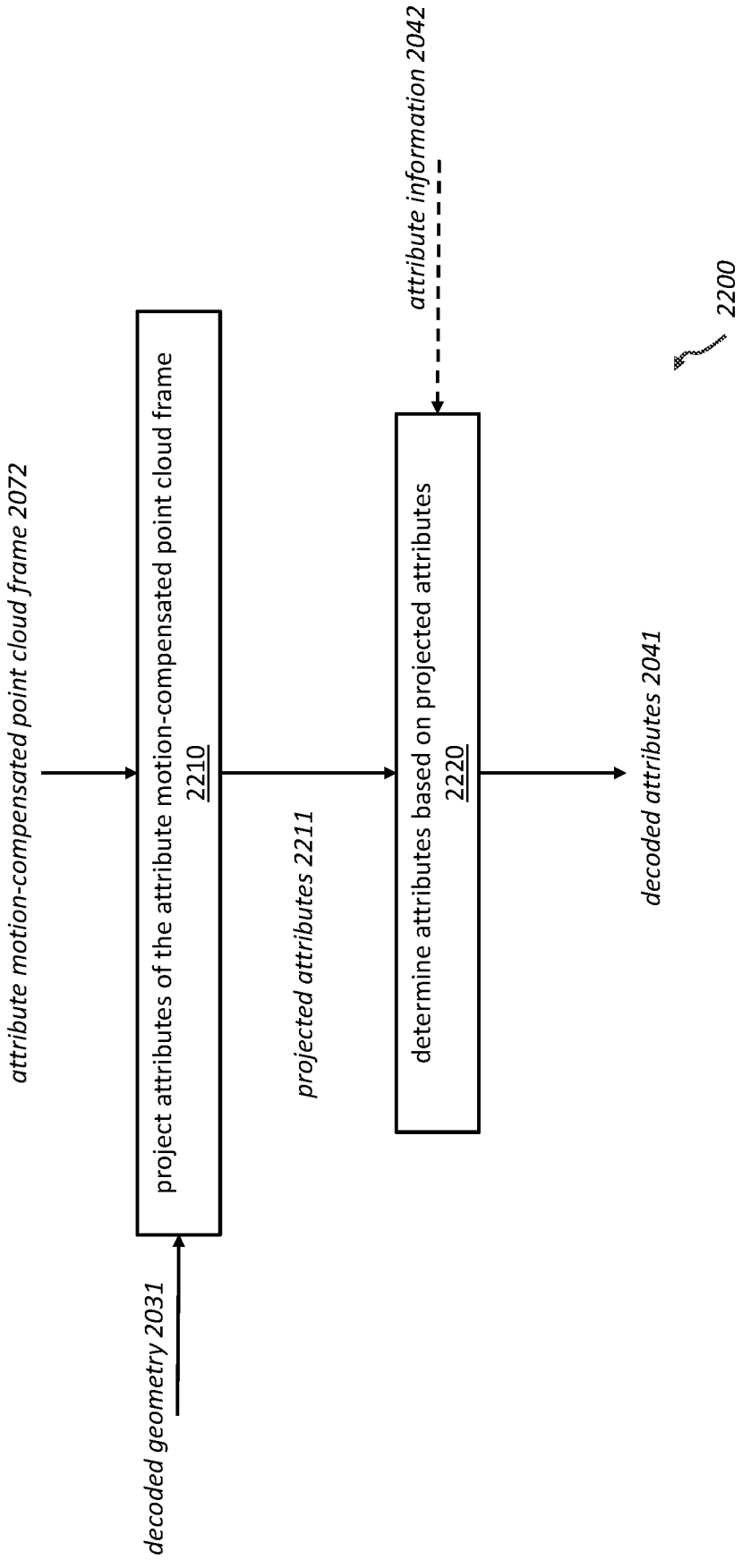
attribute motion-compensated point cloud frame 2072
project attributes of the attribute motion-compensated point cloud frame
2210
projected attributes 2211
attribute information 2042
determine attributes based on projected attributes
2220
decoded attributes 2041
2200
decoded geometry 2031

FIG. 22

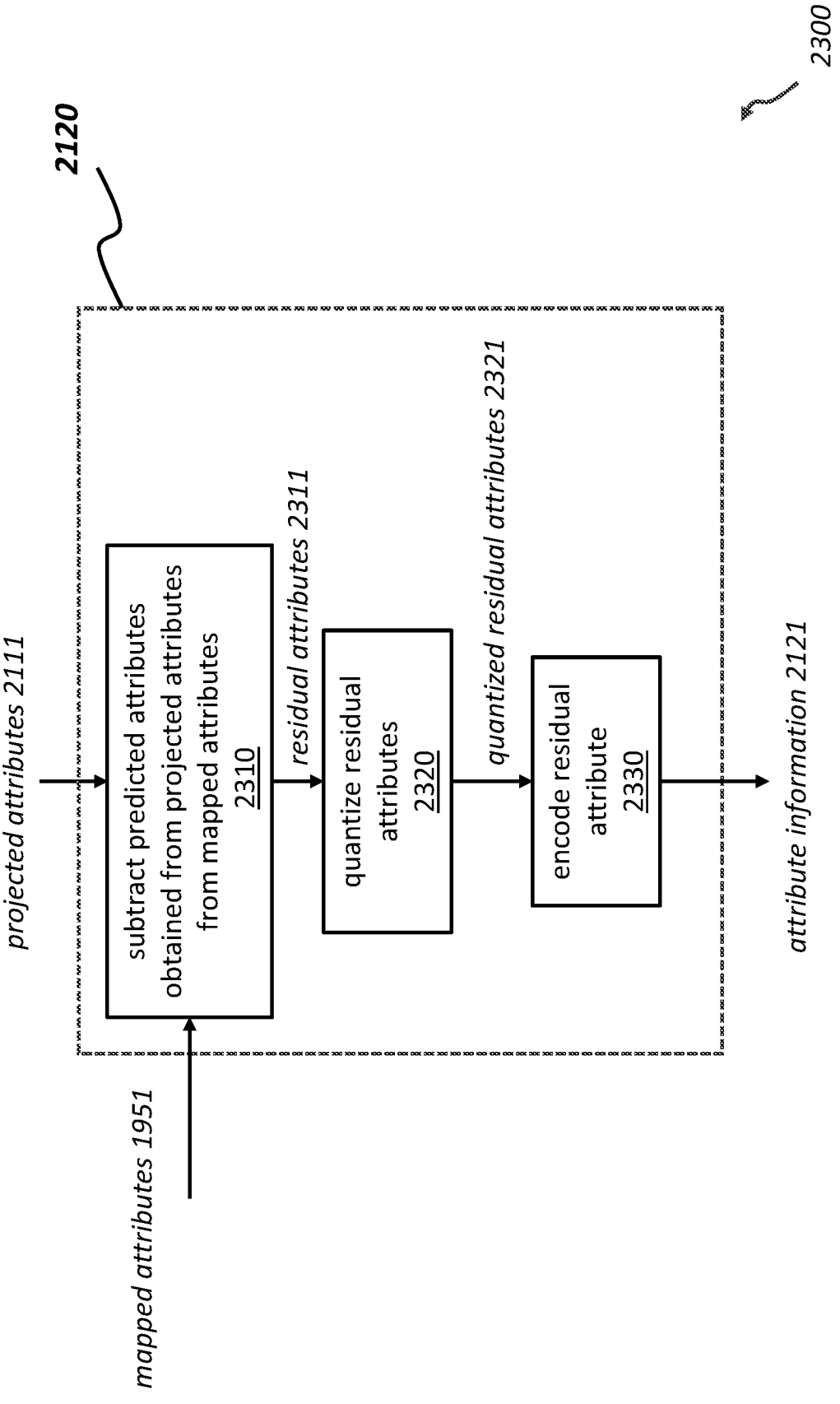
2300
2120
projected attributes 2111
mapped attributes 1951
subtract predicted attributes obtained from projected attributes from mapped attributes 2310
residual attributes 2311
quantize residual attributes 2320
quantized residual attributes 2321
encode residual attribute 2330
attribute information 2121

FIG. 23

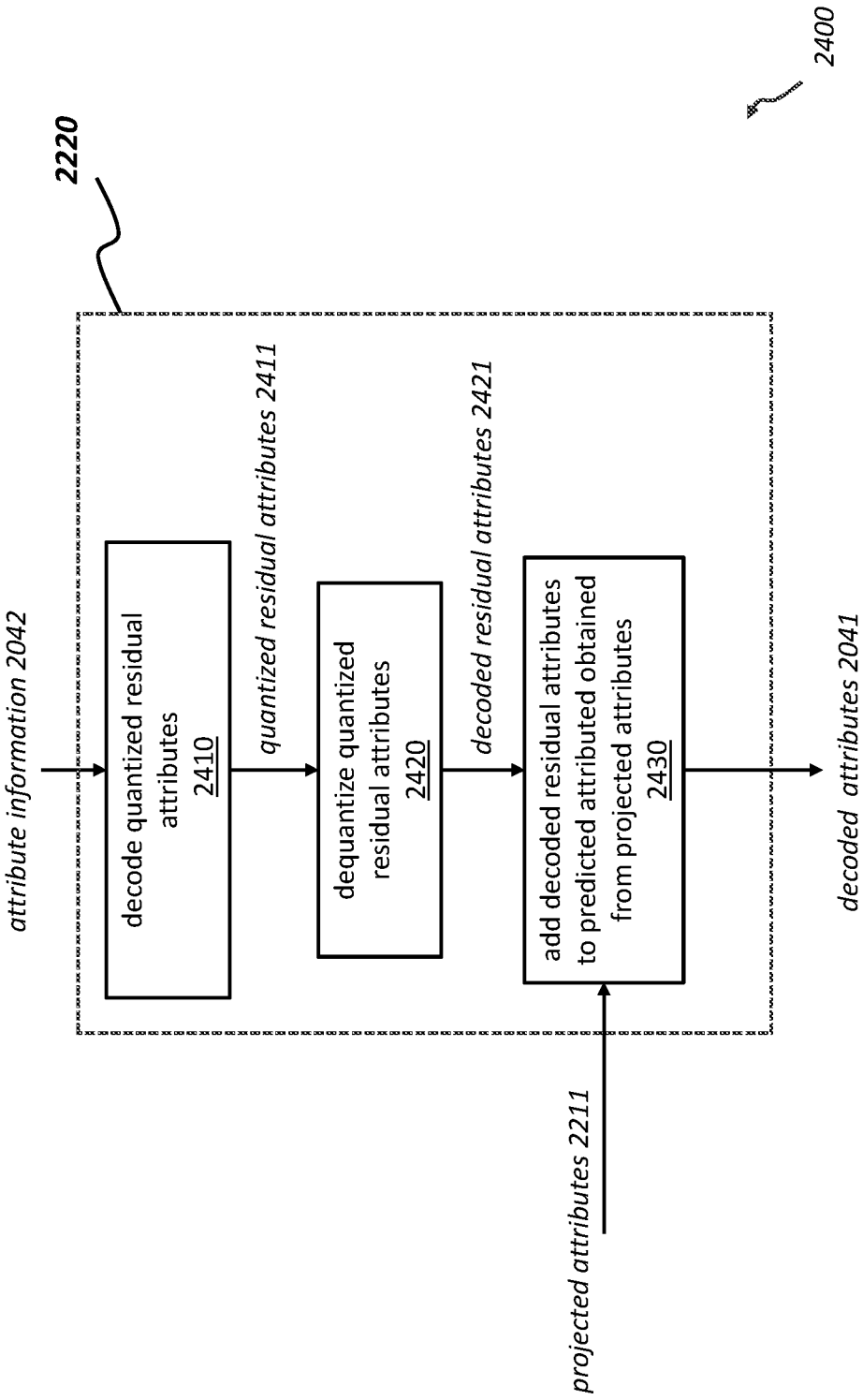
2400
2220
attribute information 2042
decode quantized residual attributes 2410
quantized residual attributes 2411
dequantize quantized residual attributes 2420
decoded residual attributes 2421
add decoded residual attributes to predicted attributed obtained from projected attributes 2430
decoded attributes 2041
projected attributes 2211

FIG. 24

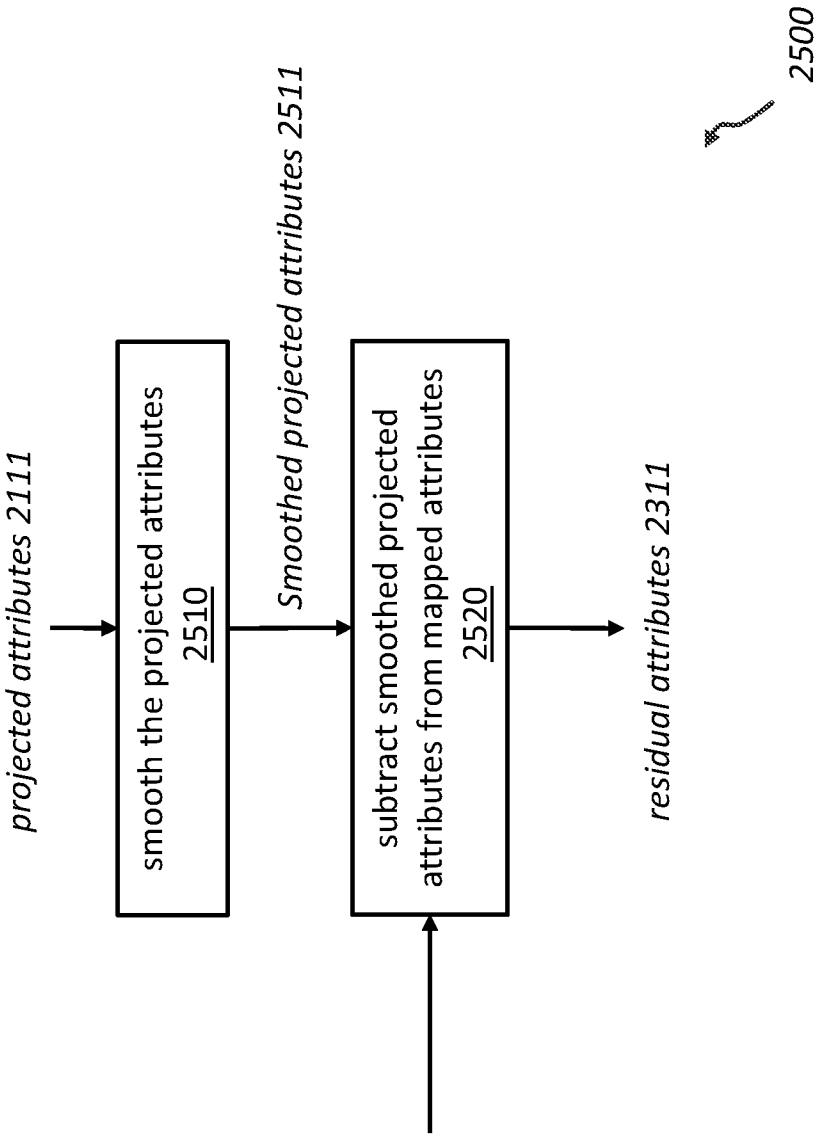
2500
projected attributes 2111
smooth the projected attributes
2510
Smoothed projected attributes 2511
subtract smoothed projected attributes from mapped attributes
2520
residual attributes 2311

FIG. 25

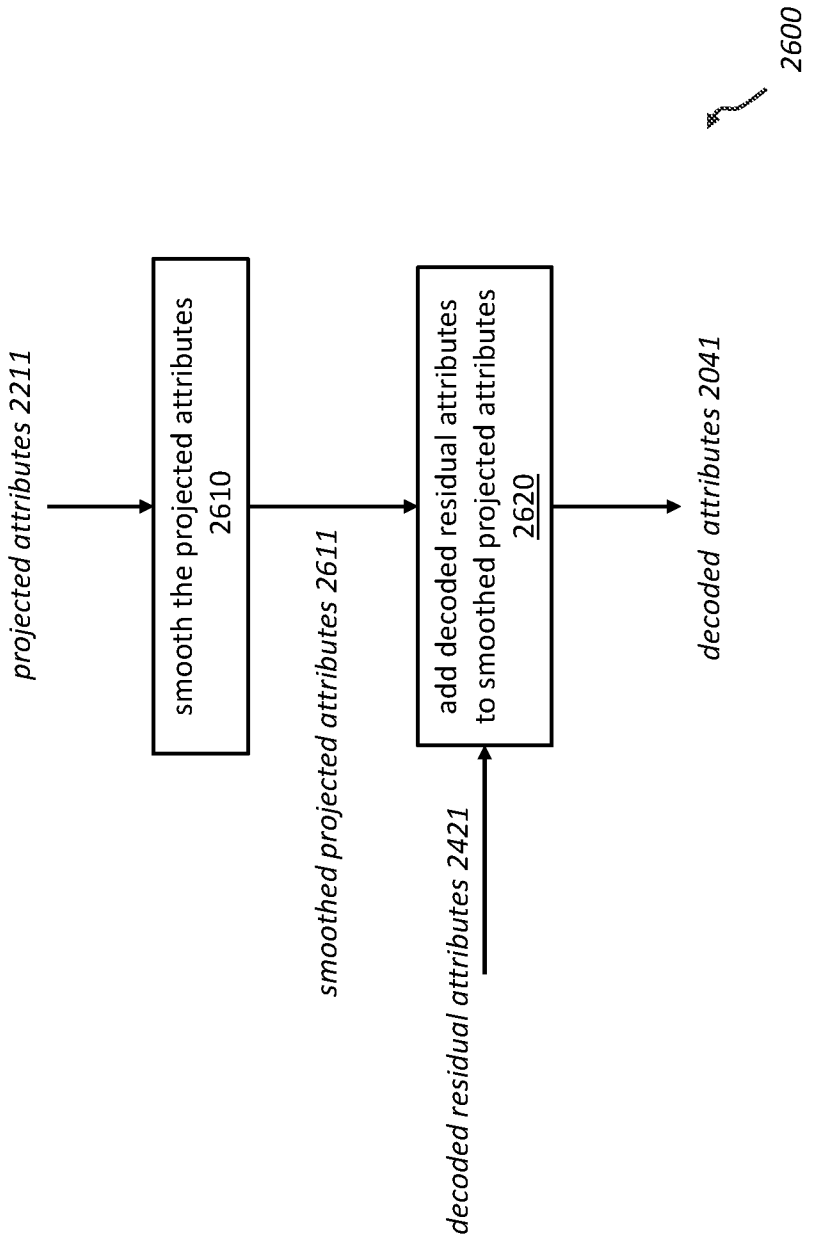
2600
projected attributes 2211
smooth the projected attributes 2610
smoothed projected attributes 2611
add decoded residual attributes to smoothed projected attributes 2620
decoded residual attributes 2421
decoded attributes 2041

FIG. 26

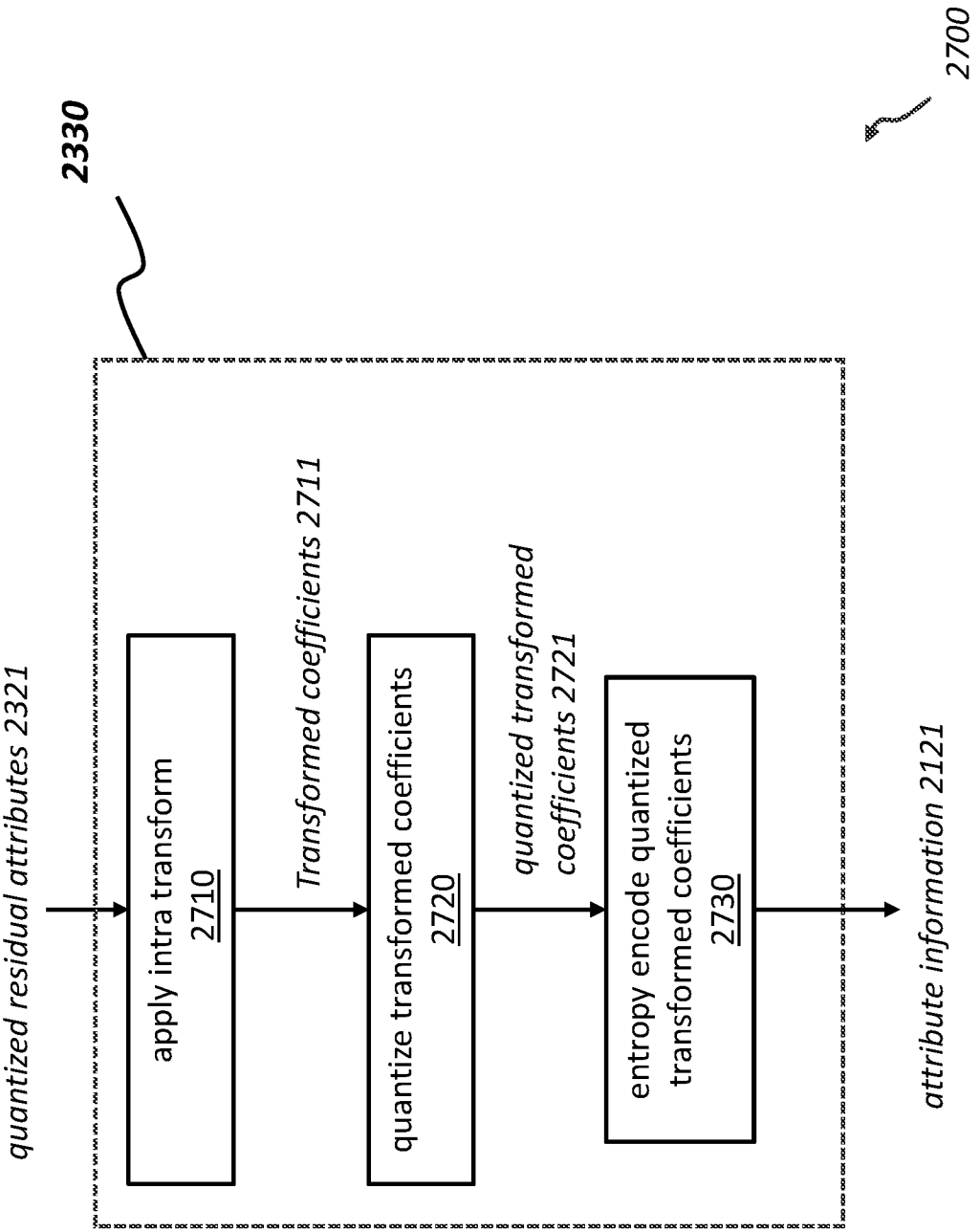
2700
2330
quantized residual attributes 2321
apply intra transform
2710
Transformed coefficients 2711
quantize transformed coefficients
2720
quantized transformed coefficients 2721
entropy encode quantized transformed coefficients
2730
attribute information 2121

FIG. 27

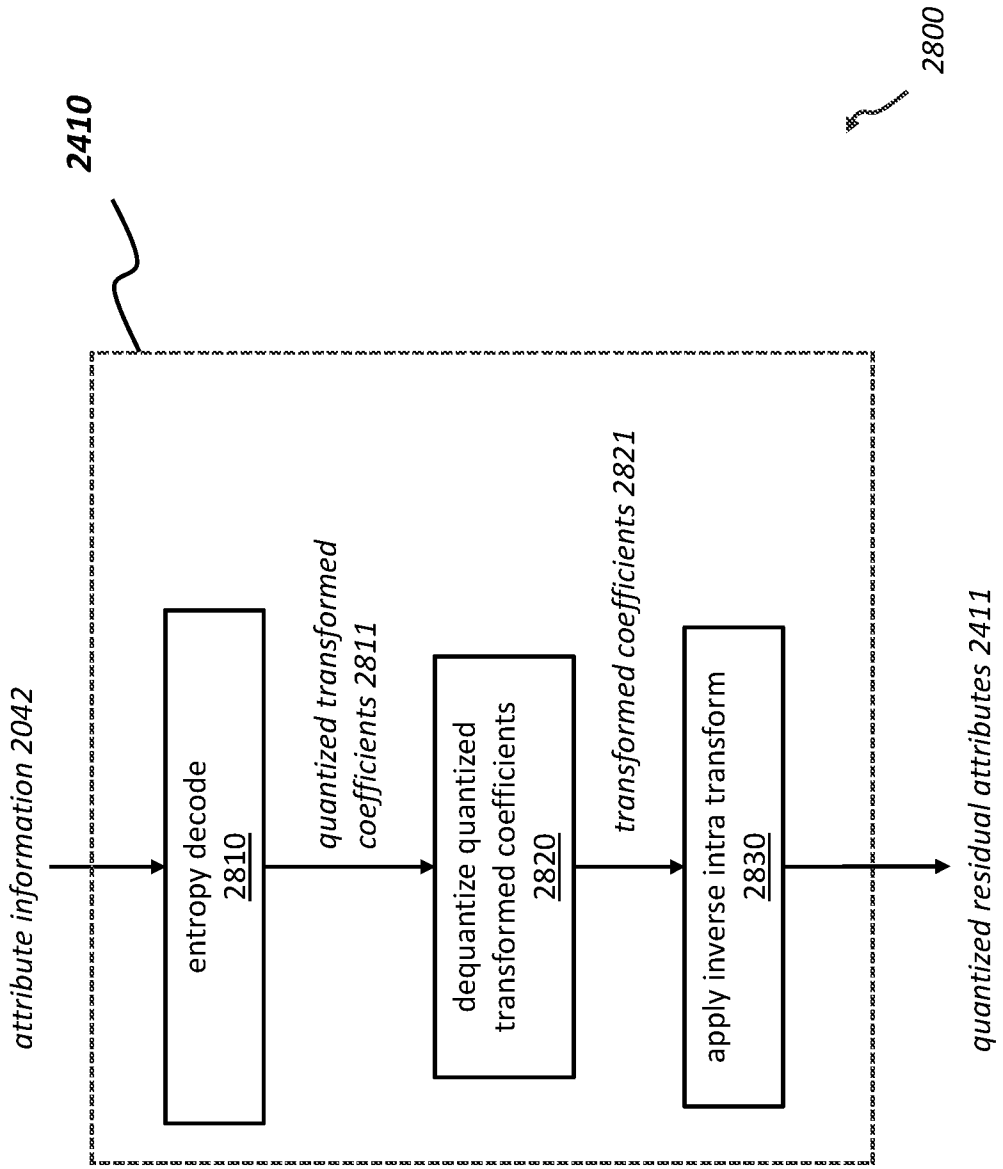
2800
2410
attribute information 2042
entropy decode
2810
quantized transformed coefficients 2811
dequantize quantized transformed coefficients
2820
transformed coefficients 2821
apply inverse intra transform
2830
quantized residual attributes 2411

FIG. 28

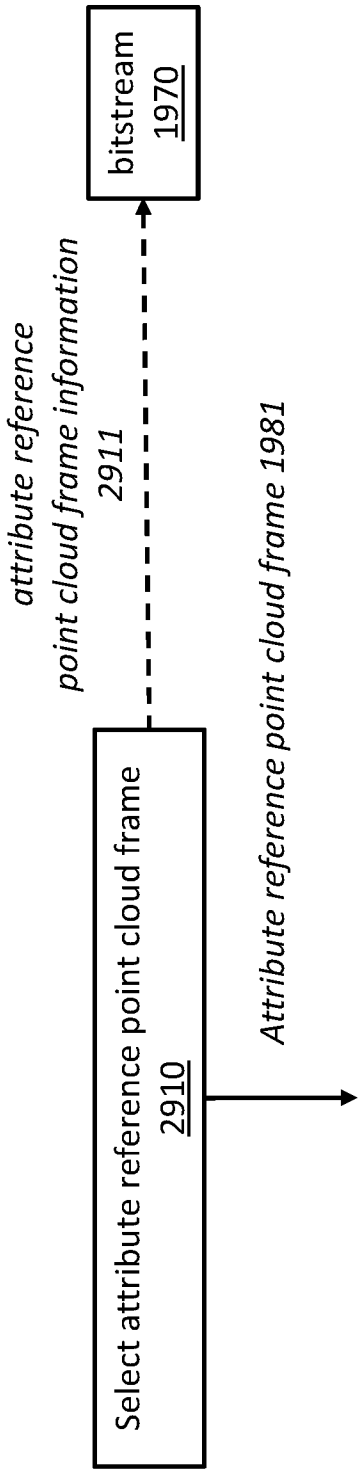
bitstream
1970
attribute reference
point cloud frame information
2911
Select attribute reference point cloud frame
2910
Attribute reference point cloud frame 1981
2900

FIG. 29

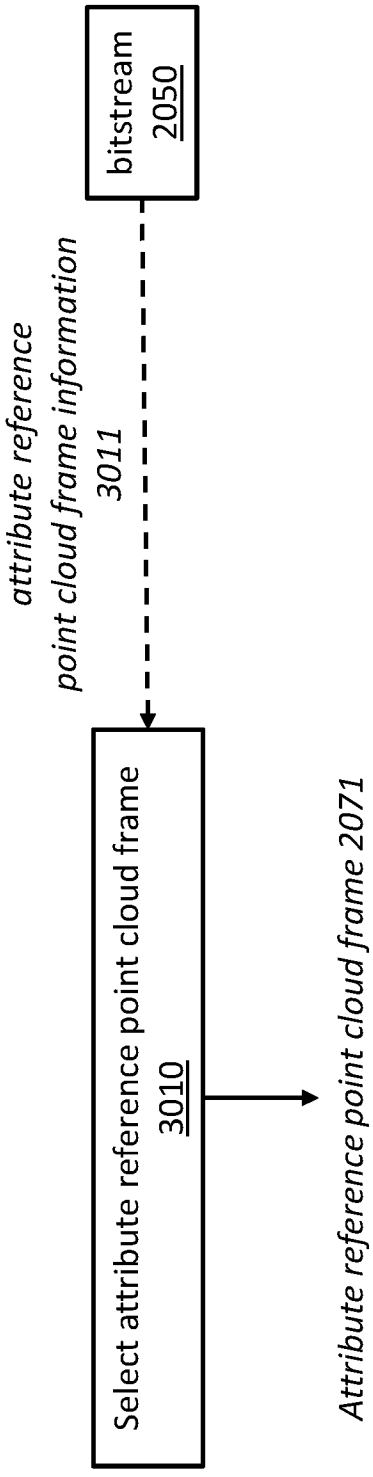
bitstream
2050
attribute reference
point cloud frame information
3011
Select attribute reference point cloud frame
3010
Attribute reference point cloud frame 2071
3000

FIG. 30

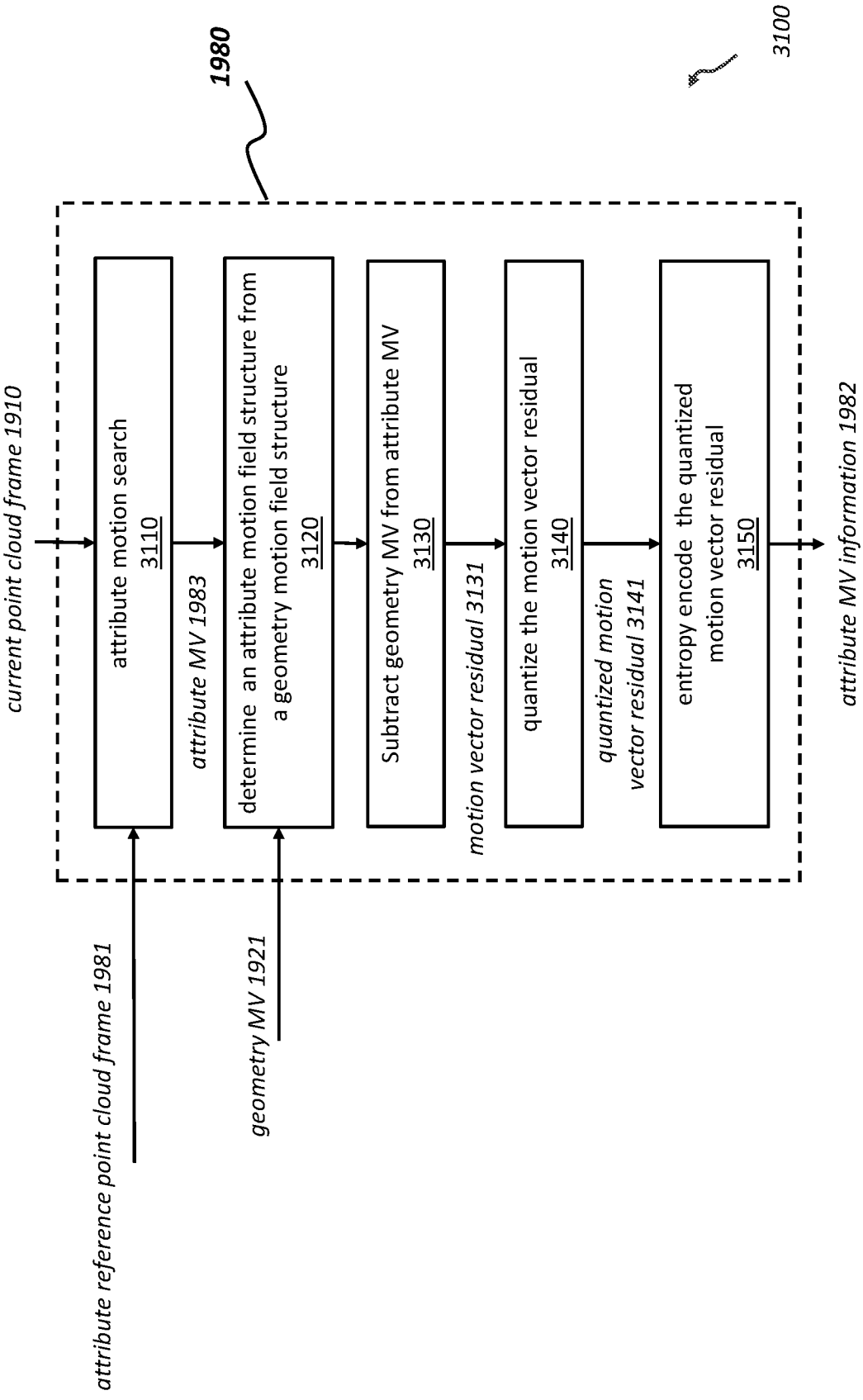
1980
3100
current point cloud frame 1910
attribute motion search
3110
attribute MV 1983
determine an attribute motion field structure from a geometry motion field structure
3120
Subtract geometry MV from attribute MV
3130
motion vector residual 3131
quantize the motion vector residual
3140
quantized motion vector residual 3141
entropy encode the quantized motion vector residual
3150
attribute MV information 1982
attribute reference point cloud frame 1981
geometry MV 1921

FIG. 31

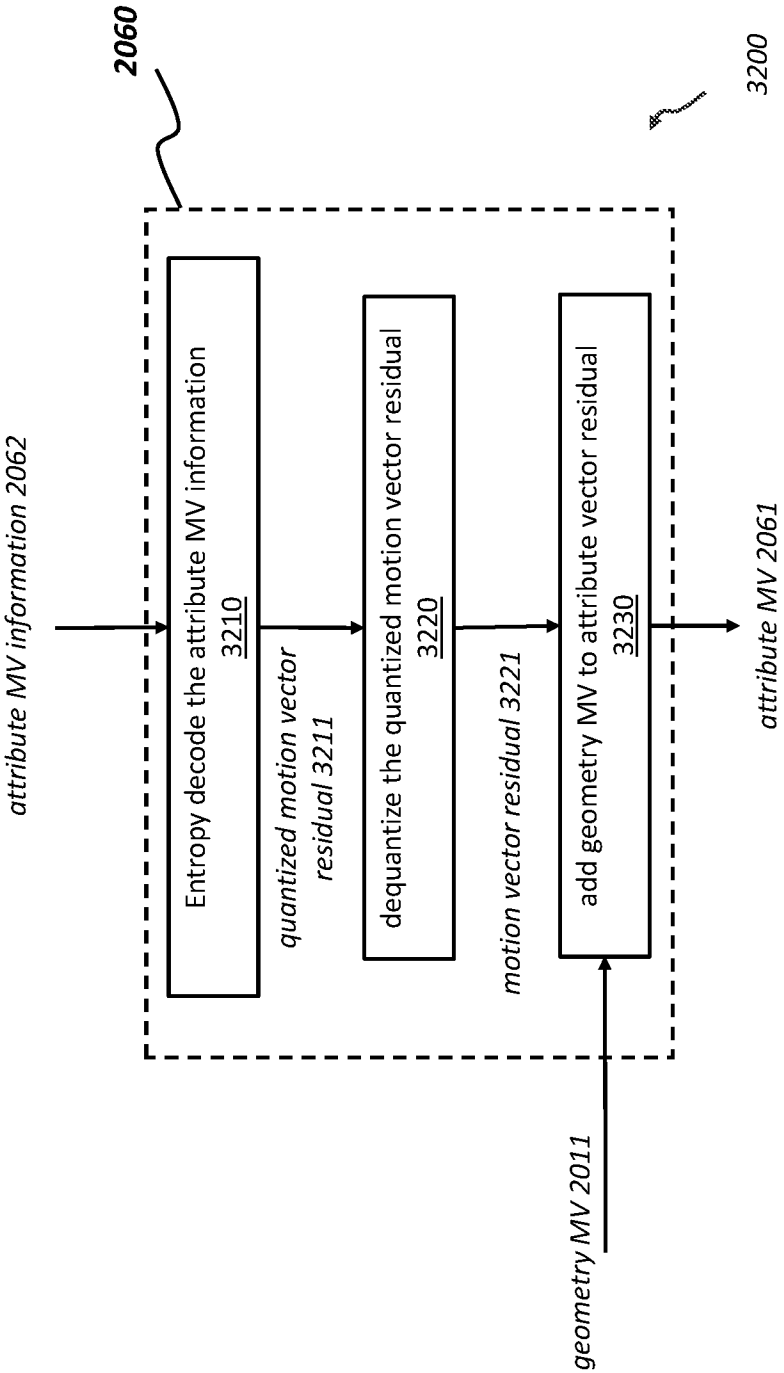
2060
3200
attribute MV information 2062
Entropy decode the attribute MV information 3210
quantized motion vector residual 3211
dequantize the quantized motion vector residual 3220
motion vector residual 3221
add geometry MV to attribute vector residual 3230
attribute MV 2061
geometry MV 2011

FIG. 32

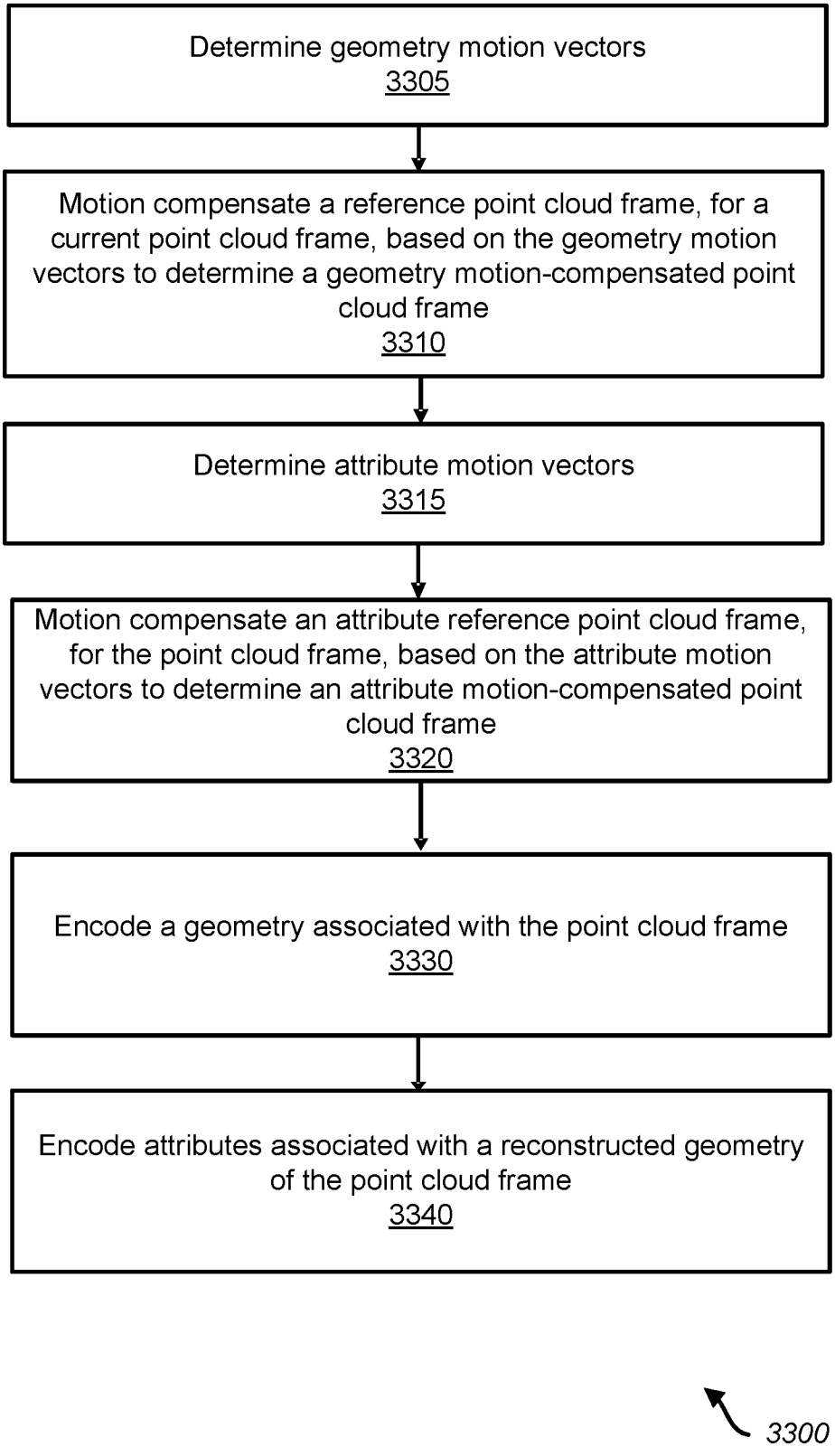
Determine geometry motion vectors
3305
Motion compensate a reference point cloud frame, for a current point cloud frame, based on the geometry motion vectors to determine a geometry motion-compensated point cloud frame
3310
Determine attribute motion vectors
3315
Motion compensate an attribute reference point cloud frame, for the point cloud frame, based on the attribute motion vectors to determine an attribute motion-compensated point cloud frame
3320
Encode a geometry associated with the point cloud frame
3330
Encode attributes associated with a reconstructed geometry of the point cloud frame
3340
3300

FIG. 33

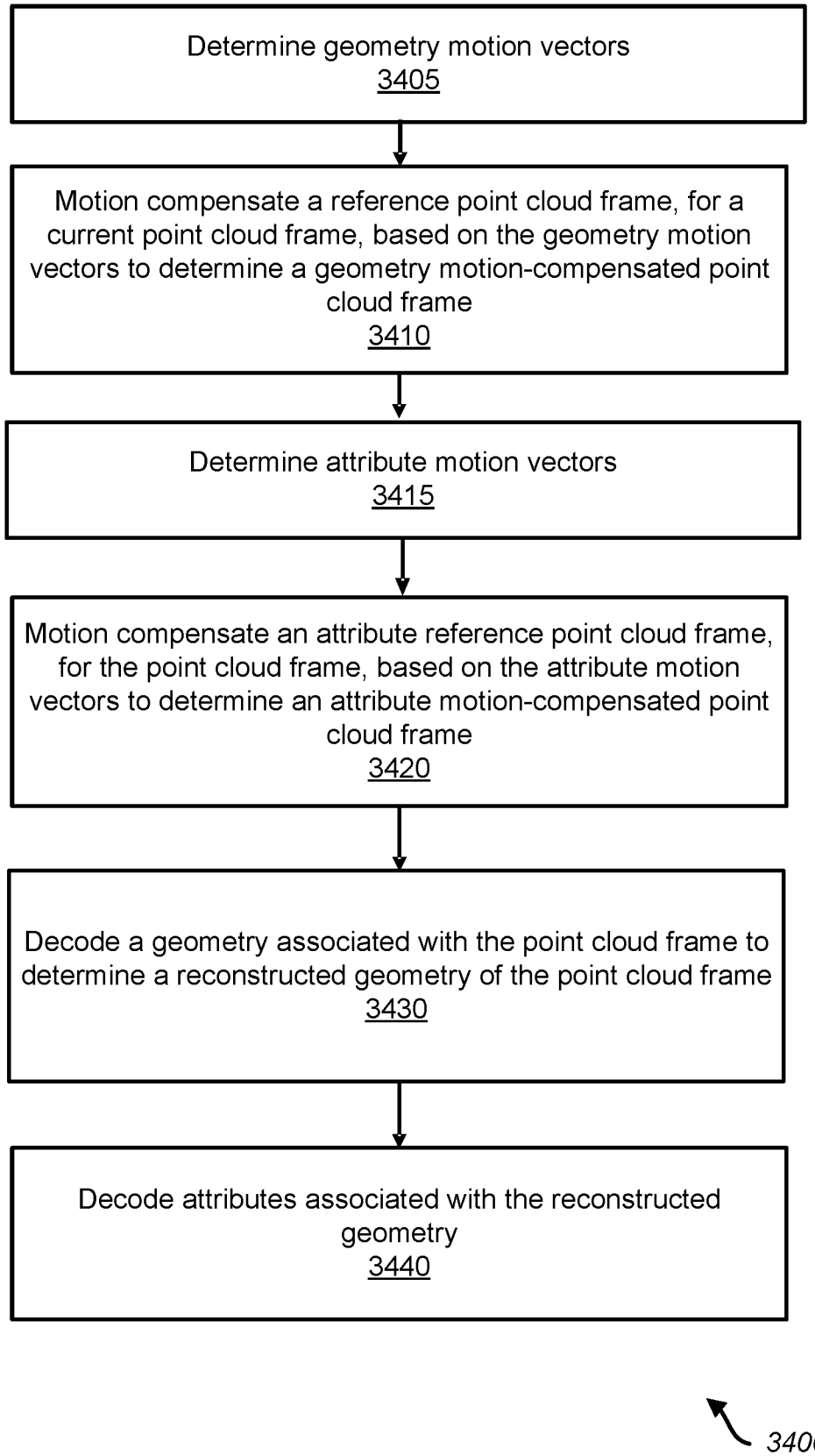
Determine geometry motion vectors
3405
Motion compensate a reference point cloud frame, for a current point cloud frame, based on the geometry motion vectors to determine a geometry motion-compensated point cloud frame
3410
Determine attribute motion vectors
3415
Motion compensate an attribute reference point cloud frame, for the point cloud frame, based on the attribute motion vectors to determine an attribute motion-compensated point cloud frame
3420
Decode a geometry associated with the point cloud frame to determine a reconstructed geometry of the point cloud frame
3430
Decode attributes associated with the reconstructed geometry
3440
3400

FIG. 34

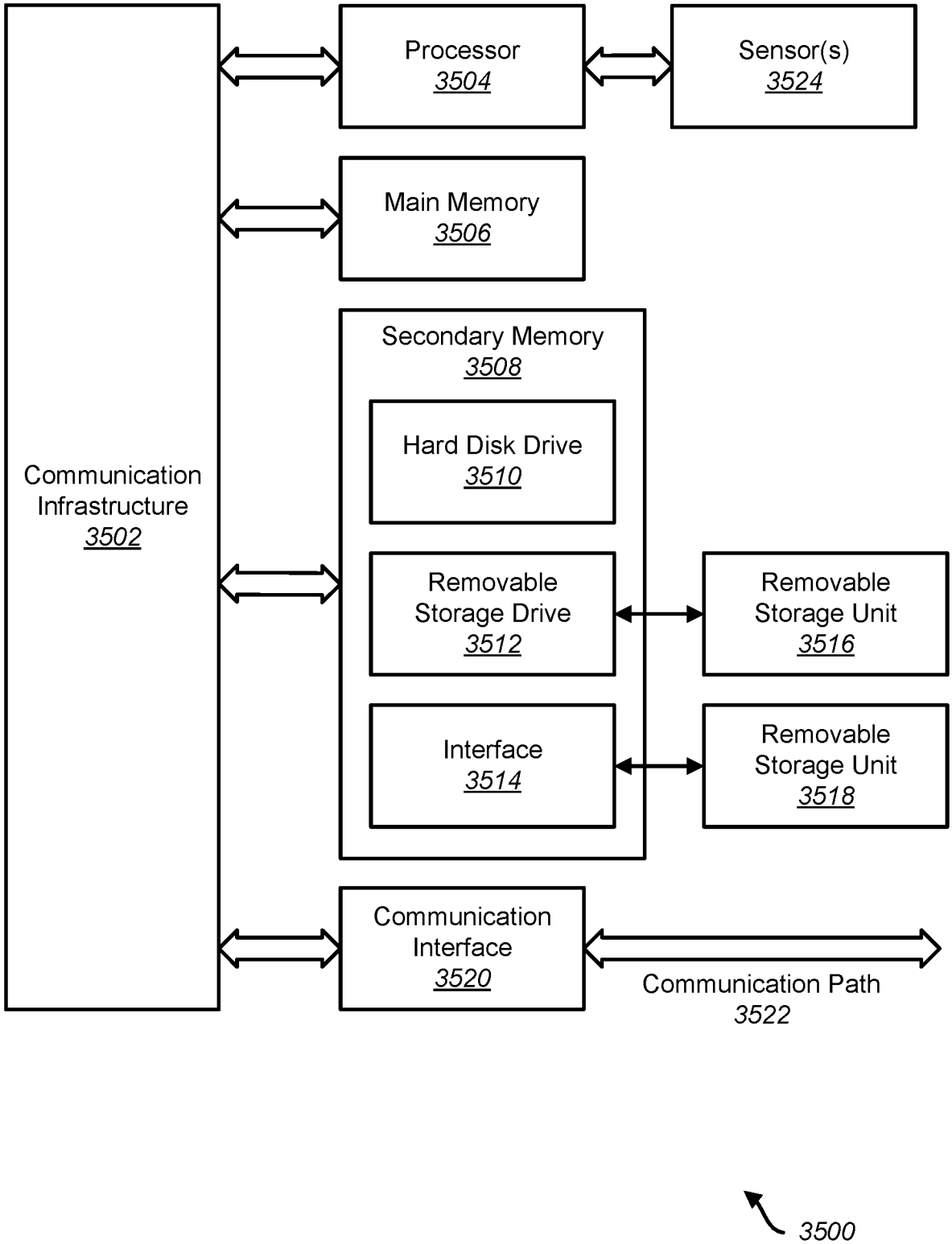
Processor
3504
Sensor(s)
3524
Main Memory
3506
Secondary Memory
3508
Hard Disk Drive
3510
Communication
Infrastructure
3502
Removable
Storage Drive
3512
Removable
Storage Unit
3516
Interface
3514
Removable
Storage Unit
3518
Communication
Interface
3520
Communication Path
3522
3500

FIG. 35

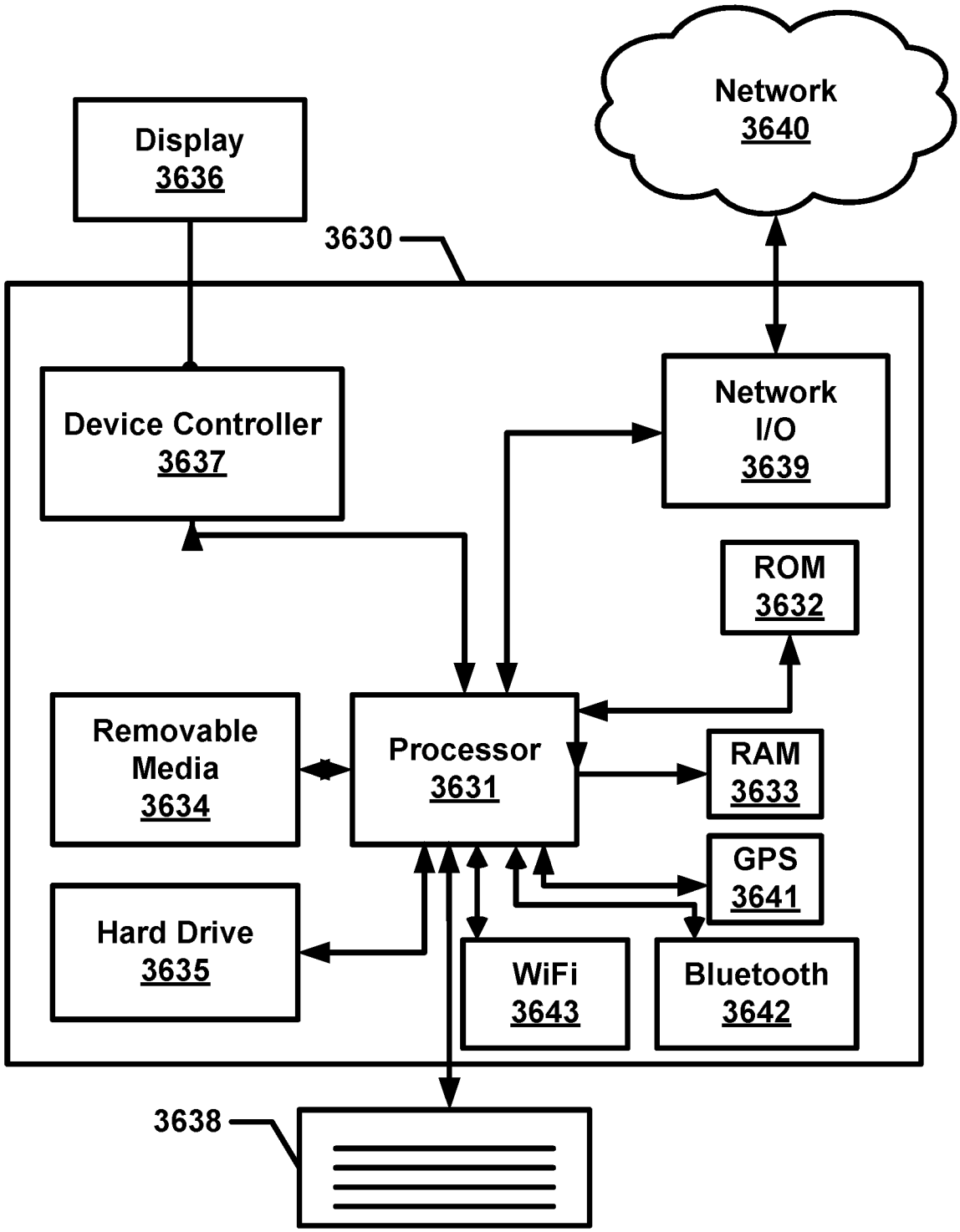
Network
3640
Display
3636
3630
Device Controller
3637
Network
I/O
3639
ROM
3632
Removable
Media
3634
Processor
3631
RAM
3633
GPS
3641
Hard Drive
3635
WiFi
3643
Bluetooth
3642
3638

FIG. 36

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2024/037831

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04N19/597 G06T9/00
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)
H04N G06T

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

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
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 Y	WO 2023/054932 A1 (LG ELECTRONICS INC [KR]) 6 April 2023 (2023-04-06) paragraphs [0205] - [0218] paragraphs [0250] - [0258] paragraphs [0305] - [0309] paragraphs [0457] - [0472] paragraph [0498] & EP 4 412 213 A1 (LG ELECTRONICS INC [KR]) 7 August 2024 (2024-08-07) paragraphs [0138] - [0151] paragraphs [0180] - [0188] paragraphs [0235] - [0239] paragraphs [0338] - [0353] paragraph [0378] -----	1,5-8, 13-15 2-4,9-12

-/--

[x] Further documents are listed in the continuation of Box C.

[x] 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	Date of mailing of the international search report
18 October 2024	28/10/2024

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	André, Thomas

1

Form PCT/ISA/210 (second sheet) (April 2005)

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	GUANGCHI FANG ET AL: "4DAC: Learning Attribute Compression for Dynamic Point Clouds", ARXIV.ORG, CORNELL UNIVERSITY LIBRARY, 201 OLIN LIBRARY CORNELL UNIVERSITY ITHACA, NY 14853, 25 April 2022 (2022-04-25), XP091209396, sections 2.2, 3.2, 3.3, 3.4 and 3.5.1	2-4,9-12
Y	WO 2021/062279 A1 (APPLE INC [US]) 1 April 2021 (2021-04-01) paragraphs [0083] - [0100] paragraphs [0113], [0114]	2-4,9-12
A	ANIQUE AKHTAR (QUALCOMM) ET AL: "[AI-3DGC][EE5.3-related] Update on baseline attribute compression for ML-based PCC", 142. MPEG MEETING; 20230424 - 20230428; ANTALYA; (MOTION PICTURE EXPERT GROUP OR ISO/IEC JTC1/SC29/WG11), , no. m63255 19 April 2023 (2023-04-19), XP030310361, Retrieved from the Internet: URL:https://dms.mpeg.expert/doc_end_user/documents/142_Antalya/wg11/m63255-v1-m63255_v1.zip m63255_v1.docx [retrieved on 2023-04-19] sections 1 and 2	2-4,9-12
A	WO 2023/056860 A1 (BEIJING BYTEDANCE NETWORK TECH CO LTD [CN]; BYTEDANCE INC [US]) 13 April 2023 (2023-04-13) page 12, section 1 pages 16-20, item 2)	8
A	CHUANG MA ET AL: "[G-PCC] CE13.34 report on filter based attribute prediction scheme", 132. MPEG MEETING; 20201012 - 20201016; ONLINE; (MOTION PICTURE EXPERT GROUP OR ISO/IEC JTC1/SC29/WG11), , no. m54951 6 October 2020 (2020-10-06), XP030292452, Retrieved from the Internet: URL:https://dms.mpeg.expert/doc_end_user/documents/132_OnLine/wg11/m54951-v1-m54951-ce13.34.zip m54951-ce13.34/m54951_CE13.34.docx [retrieved on 2020-10-06] the whole document	12

-/--

1

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2024/037831

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X,P	TAQUET (OFINNO) J ET AL: "[G-PCC][EE13.60] Test1 - On Dual Motion Field", 145. MPEG MEETING; 20240122 - 20240126; ONLINE; (MOTION PICTURE EXPERT GROUP OR ISO/IEC JTC1/SC29/WG11), , no. m66238 17 January 2024 (2024-01-17), XP030315537, Retrieved from the Internet: URL:https://dms.mpeg.expert/doc_end_user/documents/145_OnLine/wg11/m66238-v1-m66238_v1.zip m66238 [G-PCC][EE13.60] Test 1 - on dual motion field.docx [retrieved on 2024-01-17] the whole document -----	1-15
A	MA CHUANG ET AL: "Fast Recolor Prediction Scheme in Point Cloud Attribute Compression", 2020 IEEE INTERNATIONAL CONFERENCE ON VISUAL COMMUNICATIONS AND IMAGE PROCESSING (VCIP), 1 December 2020 (2020-12-01), pages 50-53, XP055959257, DOI: 10.1109/VCIP49819.2020.9301768 ISBN: 978-1-7281-8068-7 abstract section IV -----	2-4,9-12
A	"G-PCC 2nd Edition codec description", 139. MPEG MEETING; 20220718 - 20220722; ONLINE; (MOTION PICTURE EXPERT GROUP OR ISO/IEC JTC1/SC29/WG11), , no. n21684 20 October 2022 (2022-10-20), XP030306169, Retrieved from the Internet: URL:https://dms.mpeg.expert/doc_end_user/documents/139_OnLine/wg11/MDS21684_WG07_N00364.zip MDS21684_WG07_N00364_v1.docx [retrieved on 2022-10-20] section 6.4.2 -----	2-4,9-12
A	COHEN ROBERT A ET AL: "Point Cloud Attribute Compression Using 3-D Intra Prediction and Shape-Adaptive Transforms", 2016 DATA COMPRESSION CONFERENCE (DCC), IEEE, 30 March 2016 (2016-03-30), pages 141-150, XP033027696, DOI: 10.1109/DCC.2016.67 [retrieved on 2016-12-15] sections 3, 4.1 and 6 -----	2-4,9-12

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2024/037831

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2023054932 A1	06-04-2023	CN 118020304 A	10-05-2024
		EP 4412213 A1	07-08-2024
		WO 2023054932 A1	06-04-2023
WO 2021062279 A1	01-04-2021	US 2021099711 A1	01-04-2021
		WO 2021062279 A1	01-04-2021
WO 2023056860 A1	13-04-2023	CN 118369913 A	19-07-2024
		US 2024249441 A1	25-07-2024
		WO 2023056860 A1	13-04-2023