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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
12 January 2023 (12.01.2023)

WIPO | PCT

(10) International Publication Number
WO 2023/280147 A1

(51) International Patent Classification:
G06T 7/246 (2017.01) *H04N 19/119* (2014.01)
G06T 9/00 (2006.01)

(21) International Application Number:
PCT/CN2022/103834

(22) International Filing Date:
05 July 2022 (05.07.2022)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
PCT/CN2021/104573
05 July 2021 (05.07.2021) CN
PCT/CN2021/119488
21 September 2021 (21.09.2021) CN

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

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

(74) Agent: **SHIHUI PARTNERS**; 42/F, Tower C, Beijing Yintai Centre, No.2 Jianguomenwai Avenue, Chaoyang District, Beijing 100022 (CN).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, 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, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU,

(54) Title: METHOD, APPARATUS, AND MEDIUM FOR POINT CLOUD CODING

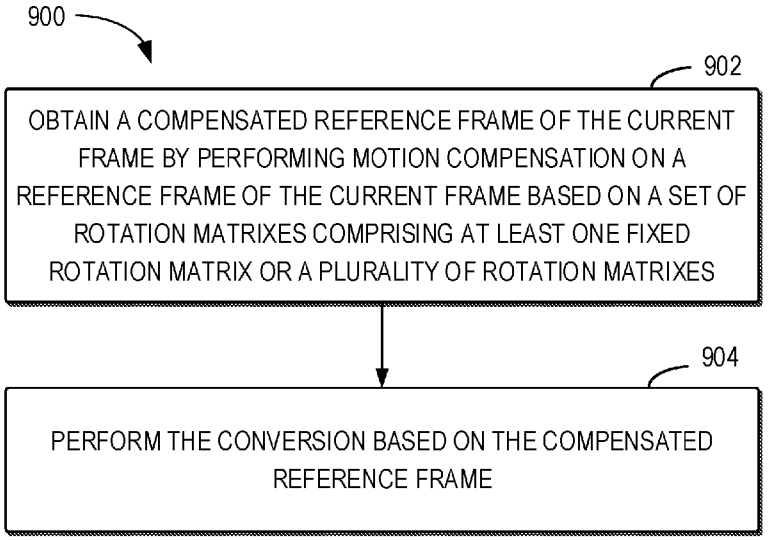


Fig. 9

(57) Abstract: Embodiments of the present disclosure provide a method for point cloud coding. The method comprises: obtaining, during a conversion between a current frame of a point cloud sequence and a bitstream of the point cloud sequence, a compensated reference frame of the current frame by performing motion compensation on a reference frame of the current frame based on a set of rotation matrixes comprising at least one fixed rotation matrix or a plurality of rotation matrixes; and performing the conversion based on the compensated reference frame. Compared with the conventional solution, the proposed method can advantageously improve coding efficiency.

WO 2023/280147 A1

[Continued on next page]

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

Published:

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

METHOD, APPARATUS, AND MEDIUM FOR POINT CLOUD CODING

FIELD

[0001] Embodiments of the present disclosure relates generally to point cloud coding techniques, and more particularly, to classification-based global motion estimation methods for point cloud coding.

BACKGROUND

[0002] A point cloud is a collection of individual data points in a three-dimensional (3D) plane with each point having a set coordinate on the X, Y, and Z axes. Thus, a point cloud may be used to represent the physical content of the three-dimensional space. Point clouds have shown to be a promising way to represent 3D visual data for a wide range of immersive applications, from augmented reality to autonomous cars.

[0003] Point cloud coding standards have evolved primarily through the development of the well-known MPEG organization. MPEG, short for Moving Picture Experts Group, is one of the main standardization groups dealing with multimedia. In 2017, the MPEG 3D Graphics Coding group (3DG) published a call for proposals (CFP) document to start to develop point cloud coding standard. The final standard will consist in two classes of solutions. Video-based Point Cloud Compression (V-PCC or VPCC) is appropriate for point sets with a relatively uniform distribution of points. Geometry-based Point Cloud Compression (G-PCC or GPCC) is appropriate for more sparse distributions. However, coding efficiency of conventional point cloud coding techniques is generally expected to be further improved.

SUMMARY

[0004] In a first aspect, a method for point cloud coding is proposed. The method comprises: obtaining, during a conversion between a current frame of a point cloud sequence and a bitstream of the point cloud sequence, a compensated reference frame of the current frame by performing motion compensation on a reference frame of the current frame based on a set of rotation matrixes comprising at least one fixed rotation matrix or a plurality of rotation matrixes; and performing the convention based on the compensated reference frame.

[0005] According to the method in accordance with the first aspect of the present disclosure, a fixed rotation matrix or a plurality of rotation matrixes may be used for performing motion compensation. If a fixed rotation matrix is used, the proposed method can advantageously reduce bits and resources wasted on processing rotation matrix and improve coding efficiency. If a plurality of rotation matrixes are used, the proposed method can advantageously improve the accuracy of motion compensation and coding efficiency.

[0006] In a second aspect, another method for point cloud coding is proposed. The method comprises: classifying, during a conversion between a current frame of a point cloud sequence and a bitstream of the point cloud sequence, points in the point cloud sequence based on a plurality types of process units for global motion estimation of the current frame; and performing the conversion based on the classification.

[0007] The method in accordance with the second aspect of the present disclosure utilizes various types of process units to perform the classification process. Compared with the conventional solution where only a block is used, the proposed method can advantageously reduce the number of iterations in the classification process and thus reduce the complexity of classification process and improve the coding efficiency.

[0008] In a third aspect, an apparatus for processing point cloud data is proposed. The apparatus for processing point cloud data comprises a processor and a non-transitory memory with instructions thereon. The instructions upon execution by the processor, cause the processor to perform a method in accordance with the first or second aspect of the present disclosure.

[0009] In a fourth aspect, a non-transitory computer-readable storage medium is proposed. The non-transitory computer-readable storage medium stores instructions that cause a processor to perform a method in accordance with the first or second aspect of the present disclosure.

[0010] In a fifth aspect, a non-transitory computer-readable recording medium is proposed. The non-transitory computer-readable recording medium stores a bitstream of a point cloud sequence which is generated by a method performed by a point cloud processing apparatus. The method comprises: obtaining a compensated reference frame of a current frame of the point cloud sequence by performing motion compensation on a reference frame of the current frame based on a set of rotation matrixes comprising at least one fixed rotation matrix or a plurality of rotation matrixes; and generating the bitstream based on the compensated reference frame.

[0011] In a sixth aspect, a method for storing a bitstream of a point cloud sequence is proposed. The method comprises: obtaining a compensated reference frame of a current frame of the point cloud sequence by performing motion compensation on a reference frame of the current frame based on a set of rotation matrixes comprising at least one fixed rotation matrix or a plurality of rotation matrixes; generating the bitstream based on the compensated reference frame; and storing the bitstream in a non-transitory computer-readable recording medium.

[0012] In a seventh aspect, another non-transitory computer-readable recording medium is proposed. The non-transitory computer-readable recording medium stores a bitstream of a point cloud sequence which is generated by a method performed by a point cloud processing apparatus. The method comprises: classifying points in the point cloud sequence based on a plurality types of process units for global motion estimation of a current frame of the point cloud sequence; and generating the bitstream based on the classification.

[0013] In an eighth aspect, another method for storing a bitstream of a point cloud sequence is proposed. The method comprises: classifying points in the point cloud sequence based on a plurality types of process units for global motion estimation of a current frame of the point cloud sequence; generating the bitstream based on the classification; and storing the bitstream in a non-transitory computer-readable recording medium.

[0014] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used to limit the scope of the claimed subject matter.

BRIEF DESCRIPTION OF THE DRAWINGS

[0015] Through the following detailed description with reference to the accompanying drawings, the above and other objectives, features, and advantages of example embodiments of the present disclosure will become more apparent. In the example embodiments of the present disclosure, the same reference numerals usually refer to the same components.

[0016] Fig. 1 is a block diagram that illustrates an example point cloud coding system that may utilize the techniques of the present disclosure;

[0017] Fig. 2 illustrates a block diagram that illustrates an example point cloud encoder, in accordance with some embodiments of the present disclosure;

[0018] Fig. 3 illustrates a block diagram that illustrates an example point cloud decoder, in accordance with some embodiments of the present disclosure;

[0019] Fig. 4 is a schematic diagram illustrating an example of planar regions and blocks in a point cloud;

[0020] Fig. 5 is a schematic diagram illustrating an example of generation blocks for a point cloud;

[0021] Fig. 6 is a schematic diagram illustrating an example of generation reference blocks for current block;

[0022] Fig. 7 is a schematic diagram illustrating an example of generation planar regions for a point cloud;

[0023] Fig. 8 is a schematic diagram illustrating an example of generation reference planar regions for current planar region;

[0024] Fig. 9 illustrates a flowchart of a method for point cloud coding in accordance with some embodiments of the present disclosure;

[0025] Fig. 10 illustrates another flowchart of a method for point cloud coding in accordance with some embodiments of the present disclosure; and

[0026] Fig. 11 illustrates a block diagram of a computing device in which various embodiments of the present disclosure can be implemented.

[0027] Throughout the drawings, the same or similar reference numerals usually refer to the same or similar elements.

DETAILED DESCRIPTION

[0028] Principle of the present disclosure will now be described with reference to some embodiments. It is to be understood that these embodiments are described only for the purpose of illustration and help those skilled in the art to understand and implement the present disclosure, without suggesting any limitation as to the scope of the disclosure. The disclosure described herein can be implemented in various manners other than the ones described below.

[0029] In the following description and claims, unless defined otherwise, all technical and scientific terms used herein have the same meaning as commonly understood by one of ordinary skills in the art to which this disclosure belongs.

[0030] References in the present disclosure to "one embodiment," "an embodiment," "an example embodiment," and the like indicate that the embodiment described may include a particular feature, structure, or characteristic, but it is not necessary that every embodiment includes the particular feature, structure, or characteristic. Moreover, such phrases are not necessarily referring to the same embodiment. Further, when a particular feature, structure, or characteristic is described in connection with an example embodiment, it is submitted that it is within the knowledge of one skilled in the art to affect such feature, structure, or characteristic in connection with other embodiments whether or not explicitly described.

[0031] It shall be understood that although the terms "first" and "second" etc. may be used herein to describe various elements, these elements should not be limited by these terms. These terms are only used to distinguish one element from another. For example, a first element could be termed a second element, and similarly, a second element could be termed a first element, without departing from the scope of example embodiments. As used herein, the term "and/or" includes any and all combinations of one or more of the listed terms.

[0032] The terminology used herein is for the purpose of describing particular embodiments only and is not intended to be limiting of example embodiments. As used herein, the singular forms "a", "an" and "the" are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will be further understood that the terms "comprises", "comprising", "has", "having", "includes" and/or "including", when used herein, specify the presence of stated features, elements, and/or components etc., but do not preclude the presence or addition of one or more other features, elements, components and/ or combinations thereof.

Example Environment

[0033] Fig. 1 is a block diagram that illustrates an example point cloud coding system 100 that may utilize the techniques of the present disclosure. As shown, the point cloud coding system 100 may include a source device 110 and a destination device 120. The source device 110 can be also referred to as a point cloud encoding device, and the destination device 120 can be also referred to as a point cloud decoding device. In operation, the source device 110 can be configured to generate encoded point cloud data and the destination device 120 can be configured to decode the encoded point cloud data generated by the source device 110. The techniques of this disclosure are generally directed to coding (encoding and/or decoding) point cloud data, i.e., to support point cloud compression. The coding may be effective in compressing and/or decompressing point cloud data.

[0034] Source device 100 and destination device 120 may comprise any of a wide range of devices, including desktop computers, notebook (i.e., laptop) computers, tablet computers, set-top boxes, telephone handsets such as smartphones and mobile phones, televisions, cameras, display devices, digital media players, video gaming consoles, video streaming devices, vehicles (e.g., terrestrial or marine vehicles, spacecraft, aircraft, etc.), robots, LIDAR devices, satellites, extended reality devices, or the like. In some cases, source device 100 and destination device 120 may be equipped for wireless communication.

[0035] The source device 100 may include a data source 112, a memory 114, a GPCC encoder 116, and an input/output (I/O) interface 118. The destination device 120 may include an input/output (I/O) interface 128, a GPCC decoder 126, a memory 124, and a data consumer 122. In accordance with this disclosure, GPCC encoder 116 of source device 100 and GPCC decoder 126 of destination device 120 may be configured to apply the techniques of this disclosure related to point cloud coding. Thus, source device 100 represents an example of an encoding device, while destination device 120 represents an example of a decoding device. In other examples, source device 100 and destination device 120 may include other components or arrangements. For example, source device 100 may receive data (e.g., point cloud data) from an internal or external source. Likewise, destination device 120 may interface with an external data consumer, rather than include a data consumer in the same device.

[0036] In general, data source 112 represents a source of point cloud data (i.e., raw, unencoded point cloud data) and may provide a sequential series of "frames" of the point cloud data to GPCC encoder 116, which encodes point cloud data for the frames. In some examples, data source 112 generates the point cloud data. Data source 112 of source device 100 may include a point cloud capture device, such as any of a variety of cameras or sensors, e.g., one or more video cameras, an archive containing previously captured point cloud data, a 3D scanner or a light detection and ranging (LIDAR) device, and/or a data feed interface to receive point cloud data from a data content provider. Thus, in some examples, data source 112 may generate the point cloud data based on signals from a LIDAR apparatus. Alternatively or additionally, point cloud data may be computer-generated from scanner, camera, sensor or other data. For example, data source 112 may generate the point cloud data, or produce a combination of live point cloud data, archived point cloud data, and computer-generated point cloud data. In each case, GPCC encoder 116 encodes the captured, pre-captured, or computer-generated point cloud data. GPCC encoder 116 may rearrange frames of the point cloud data from the received order (sometimes referred to as "display order") into a coding order for coding. GPCC

encoder 116 may generate one or more bitstreams including encoded point cloud data. Source device 100 may then output the encoded point cloud data via I/O interface 118 for reception and/or retrieval by, e.g., I/O interface 128 of destination device 120. The encoded point cloud data may be transmitted directly to destination device 120 via the I/O interface 118 through the network 130A. The encoded point cloud data may also be stored onto a storage medium/server 130B for access by destination device 120.

[0037] Memory 114 of source device 100 and memory 124 of destination device 120 may represent general purpose memories. In some examples, memory 114 and memory 124 may store raw point cloud data, e.g., raw point cloud data from data source 112 and raw, decoded point cloud data from GPCC decoder 126. Additionally or alternatively, memory 114 and memory 124 may store software instructions executable by, e.g., GPCC encoder 116 and GPCC decoder 126, respectively. Although memory 114 and memory 124 are shown separately from GPCC encoder 116 and GPCC decoder 126 in this example, it should be understood that GPCC encoder 116 and GPCC decoder 126 may also include internal memories for functionally similar or equivalent purposes. Furthermore, memory 114 and memory 124 may store encoded point cloud data, e.g., output from GPCC encoder 116 and input to GPCC decoder 126. In some examples, portions of memory 114 and memory 124 may be allocated as one or more buffers, e.g., to store raw, decoded, and/or encoded point cloud data. For instance, memory 114 and memory 124 may store point cloud data.

[0038] I/O interface 118 and I/O interface 128 may represent wireless transmitters/receivers, modems, wired networking components (e.g., Ethernet cards), wireless communication components that operate according to any of a variety of IEEE 802.11 standards, or other physical components. In examples where I/O interface 118 and I/O interface 128 comprise wireless components, I/O interface 118 and I/O interface 128 may be configured to transfer data, such as encoded point cloud data, according to a cellular communication standard, such as 4G, 4G-LTE (Long-Term Evolution), LTE Advanced, 5G, or the like. In some examples where I/O interface 118 comprises a wireless transmitter, I/O interface 118 and I/O interface 128 may be configured to transfer data, such as encoded point cloud data, according to other wireless standards, such as an IEEE 802.11 specification. In some examples, source device 100 and/or destination device 120 may include respective system-on-a-chip (SoC) devices. For example, source device 100 may include an SoC device to perform the functionality attributed to GPCC encoder 116 and/or I/O interface 118, and destination device 120 may include an SoC device to perform the functionality attributed to GPCC decoder 126 and/or I/O interface 128.

[0039] The techniques of this disclosure may be applied to encoding and decoding in support of any of a variety of applications, such as communication between autonomous vehicles, communication between scanners, cameras, sensors and processing devices such as local or remote servers, geographic mapping, or other applications.

[0040] I/O interface 128 of destination device 120 receives an encoded bitstream from source device 110. The encoded bitstream may include signaling information defined by GPCC encoder 116, which is also used by GPCC decoder 126, such as syntax elements having values that represent a point cloud. Data consumer 122 uses the decoded data. For example, data consumer 122 may use the decoded point cloud data to determine the locations of physical objects. In some examples, data consumer 122 may comprise a display to present imagery based on the point cloud data.

[0041] GPCC encoder 116 and GPCC decoder 126 each may be implemented as any of a variety of suitable encoder and/or decoder circuitry, such as one or more microprocessors, digital signal processors (DSPs), application specific integrated circuits (ASICs), field programmable gate arrays (FPGAs), discrete logic, software, hardware, firmware or any combinations thereof. When the techniques are implemented partially in software, a device may store instructions for the software in a suitable, non-transitory computer-readable medium and execute the instructions in hardware using one or more processors to perform the techniques of this disclosure. Each of GPCC encoder 116 and GPCC decoder 126 may be included in one or more encoders or decoders, either of which may be integrated as part of a combined encoder/decoder (CODEC) in a respective device. A device including GPCC encoder 116 and/or GPCC decoder 126 may comprise one or more integrated circuits, microprocessors, and/or other types of devices.

[0042] GPCC encoder 116 and GPCC decoder 126 may operate according to a coding standard, such as video point cloud compression (VPCC) standard or a geometry point cloud compression (GPCC) standard. This disclosure may generally refer to coding (e.g., encoding and decoding) of frames to include the process of encoding or decoding data. An encoded bitstream generally includes a series of values for syntax elements representative of coding decisions (e.g., coding modes).

[0043] A point cloud may contain a set of points in a 3D space, and may have attributes associated with the point. The attributes may be color information such as R, G, B or Y, Cb, Cr, or reflectance information, or other attributes. Point clouds may be captured by a variety of

cameras or sensors such as LIDAR sensors and 3D scanners and may also be computer-generated. Point cloud data are used in a variety of applications including, but not limited to, construction (modeling), graphics (3D models for visualizing and animation), and the automotive industry (LIDAR sensors used to help in navigation).

[0044] Fig. 2 is a block diagram illustrating an example of a GPCC encoder 200, which may be an example of the GPCC encoder 116 in the system 100 illustrated in Fig. 1, in accordance with some embodiments of the present disclosure. Fig. 3 is a block diagram illustrating an example of a GPCC decoder 300, which may be an example of the GPCC decoder 126 in the system 100 illustrated in Fig. 1, in accordance with some embodiments of the present disclosure.

[0045] In both GPCC encoder 200 and GPCC decoder 300, point cloud positions are coded first. Attribute coding depends on the decoded geometry. In Fig. 2 and Fig. 3, the region adaptive hierarchical transform (RAHT) unit 218, surface approximation analysis unit 212, RAHT unit 314 and surface approximation synthesis unit 310 are options typically used for Category 1 data. The level-of-detail (LOD) generation unit 220, lifting unit 222, LOD generation unit 316 and inverse lifting unit 318 are options typically used for Category 3 data. All the other units are common between Categories 1 and 3.

[0046] For Category 3 data, the compressed geometry is typically represented as an octree from the root all the way down to a leaf level of individual voxels. For Category 1 data, the compressed geometry is typically represented by a pruned octree (i.e., an octree from the root down to a leaf level of blocks larger than voxels) plus a model that approximates the surface within each leaf of the pruned octree. In this way, both Category 1 and 3 data share the octree coding mechanism, while Category 1 data may in addition approximate the voxels within each leaf with a surface model. The surface model used is a triangulation comprising 1-10 triangles per block, resulting in a triangle soup. The Category 1 geometry codec is therefore known as the Trisoup geometry codec, while the Category 3 geometry codec is known as the Octree geometry codec.

[0047] In the example of Fig. 2, GPCC encoder 200 may include a coordinate transform unit 202, a color transform unit 204, a voxelization unit 206, an attribute transfer unit 208, an octree analysis unit 210, a surface approximation analysis unit 212, an arithmetic encoding unit 214, a geometry reconstruction unit 216, an RAHT unit 218, a LOD generation unit 220, a lifting unit 222, a coefficient quantization unit 224, and an arithmetic encoding unit 226.

[0048] As shown in the example of Fig. 2, GPCC encoder 200 may receive a set of positions and a set of attributes. The positions may include coordinates of points in a point cloud. The attributes may include information about points in the point cloud, such as colors associated with points in the point cloud.

[0049] Coordinate transform unit 202 may apply a transform to the coordinates of the points to transform the coordinates from an initial domain to a transform domain. This disclosure may refer to the transformed coordinates as transform coordinates. Color transform unit 204 may apply a transform to convert color information of the attributes to a different domain. For example, color transform unit 204 may convert color information from an RGB color space to a YCbCr color space.

[0050] Furthermore, in the example of Fig. 2, voxelization unit 206 may voxelize the transform coordinates. Voxelization of the transform coordinates may include quantizing and removing some points of the point cloud. In other words, multiple points of the point cloud may be subsumed within a single "voxel," which may thereafter be treated in some respects as one point. Furthermore, octree analysis unit 210 may generate an octree based on the voxelized transform coordinates. Additionally, in the example of Fig. 2, surface approximation analysis unit 212 may analyze the points to potentially determine a surface representation of sets of the points. Arithmetic encoding unit 214 may perform arithmetic encoding on syntax elements representing the information of the octree and/or surfaces determined by surface approximation analysis unit 212. GPCC encoder 200 may output these syntax elements in a geometry bitstream.

[0051] Geometry reconstruction unit 216 may reconstruct transform coordinates of points in the point cloud based on the octree, data indicating the surfaces determined by surface approximation analysis unit 212, and/or other information. The number of transform coordinates reconstructed by geometry reconstruction unit 216 may be different from the original number of points of the point cloud because of voxelization and surface approximation. This disclosure may refer to the resulting points as reconstructed points. Attribute transfer unit 208 may transfer attributes of the original points of the point cloud to reconstructed points of the point cloud data.

[0052] Furthermore, RAHT unit 218 may apply RAHT coding to the attributes of the reconstructed points. Alternatively or additionally, LOD generation unit 220 and lifting unit 222 may apply LOD processing and lifting, respectively, to the attributes of the reconstructed

points. RAHT unit 218 and lifting unit 222 may generate coefficients based on the attributes. Coefficient quantization unit 224 may quantize the coefficients generated by RAHT unit 218 or lifting unit 222. Arithmetic encoding unit 226 may apply arithmetic coding to syntax elements representing the quantized coefficients. GPCC encoder 200 may output these syntax elements in an attribute bitstream.

[0053] In the example of Fig. 3, GPCC decoder 300 may include a geometry arithmetic decoding unit 302, an attribute arithmetic decoding unit 304, an octree synthesis unit 306, an inverse quantization unit 308, a surface approximation synthesis unit 310, a geometry reconstruction unit 312, a RAHT unit 314, a LOD generation unit 316, an inverse lifting unit 318, a coordinate inverse transform unit 320, and a color inverse transform unit 322.

[0054] GPCC decoder 300 may obtain a geometry bitstream and an attribute bitstream. Geometry arithmetic decoding unit 302 of decoder 300 may apply arithmetic decoding (e.g., CABAC or other type of arithmetic decoding) to syntax elements in the geometry bitstream. Similarly, attribute arithmetic decoding unit 304 may apply arithmetic decoding to syntax elements in attribute bitstream.

[0055] Octree synthesis unit 306 may synthesize an octree based on syntax elements parsed from geometry bitstream. In instances where surface approximation is used in geometry bitstream, surface approximation synthesis unit 310 may determine a surface model based on syntax elements parsed from geometry bitstream and based on the octree.

[0056] Furthermore, geometry reconstruction unit 312 may perform a reconstruction to determine coordinates of points in a point cloud. Coordinate inverse transform unit 320 may apply an inverse transform to the reconstructed coordinates to convert the reconstructed coordinates (positions) of the points in the point cloud from a transform domain back into an initial domain.

[0057] Additionally, in the example of Fig. 3, inverse quantization unit 308 may inverse quantize attribute values. The attribute values may be based on syntax elements obtained from attribute bitstream (e.g., including syntax elements decoded by attribute arithmetic decoding unit 304).

[0058] Depending on how the attribute values are encoded, RAHT unit 314 may perform RAHT coding to determine, based on the inverse quantized attribute values, color values for

points of the point cloud. Alternatively, LOD generation unit 316 and inverse lifting unit 318 may determine color values for points of the point cloud using a level of detail-based technique.

[0059] Furthermore, in the example of Fig. 3, color inverse transform unit 322 may apply an inverse color transform to the color values. The inverse color transform may be an inverse of a color transform applied by color transform unit 204 of encoder 200. For example, color transform unit 204 may transform color information from an RGB color space to a YCbCr color space. Accordingly, color inverse transform unit 322 may transform color information from the YCbCr color space to the RGB color space.

[0060] The various units of Fig. 2 and Fig. 3 are illustrated to assist with understanding the operations performed by encoder 200 and decoder 300. The units may be implemented as fixed-function circuits, programmable circuits, or a combination thereof. Fixed-function circuits refer to circuits that provide particular functionality and are preset on the operations that can be performed. Programmable circuits refer to circuits that can be programmed to perform various tasks and provide flexible functionality in the operations that can be performed. For instance, programmable circuits may execute software or firmware that cause the programmable circuits to operate in the manner defined by instructions of the software or firmware. Fixed-function circuits may execute software instructions (e.g., to receive parameters or output parameters), but the types of operations that the fixed-function circuits perform are generally immutable. In some examples, one or more of the units may be distinct circuit blocks (fixed-function or programmable), and in some examples, one or more of the units may be integrated circuits.

[0061] Some exemplary embodiments of the present disclosure will be described in detailed hereinafter. It should be understood that section headings are used in the present document to facilitate ease of understanding and do not limit the embodiments disclosed in a section to only that section. Furthermore, while certain embodiments are described with reference to GPCC or other specific point cloud codecs, the disclosed techniques are applicable to other point cloud coding technologies also. Furthermore, while some embodiments describe point cloud coding steps in detail, it will be understood that corresponding steps decoding that undo the coding will be implemented by a decoder.

1. Summary

This disclosure is related to point cloud coding technologies. Specifically, it is about motion estimation and inter prediction processes in point cloud inter coding. The ideas may be applied

individually or in various combination, to any point cloud coding standard or non-standard point cloud codec, e.g., the being-developed Geometry-based Point Cloud Compression (G-PCC).

2. Abbreviations

G-PCC	Geometry-based Point Cloud Compression
MPEG	Moving Picture Experts Group
3DG	3D Graphics Coding group
CFP	Call For Proposal
V-PCC	Video-based Point Cloud Compression
CE	Core Experiment
EE	Exploration Experiment
inter-EM	inter Exploration Model
LMS	Least Mean Square

3. Background

Point cloud coding standards have evolved primarily through the development of the well-known MPEG organization. MPEG, short for Moving Picture Experts Group, is one of the main standardization groups dealing with multimedia. In 2017, the MPEG 3D Graphics Coding group (3DG) published a call for proposals (CFP) document to start to develop point cloud coding standard. The final standard will consist in two classes of solutions. Video-based Point Cloud Compression (V-PCC) is appropriate for point sets with a relatively uniform distribution of points. Geometry-based Point Cloud Compression (G-PCC) is appropriate for more sparse distributions.

To explore the future point cloud coding technologies in G-PCC, Core Experiment (CE) 13.5 and Exploration Experiment (EE) 13.2 were formed to develop inter prediction technologies in G-PCC. Since then, many new inter prediction methods have been adopted by MPEG and put into the reference software named inter Exploration Model (inter-EM).

Sparse Lidar point cloud data is typically captured by Lidar sensors attached to moving vehicles. In such data, points data always move along with the movement of the objects and the

movement of the moving vehicle. When the motions of the reference frame are estimated and compensated, the residual of inter prediction can be reduced and the efficiency of inter coding can be improved significantly. The process evaluation of these motions is defined as "global motion estimation" in inter-EM.

In one point cloud frame, the points data of road and objects commonly have different motions. A division of a point cloud into road and objects will improve the accuracy of the global motion which leads to a great enhancement of the compression efficiency. In one example, the global motion estimation module can use a two-threshold method to obtain the classification of object points and road points. Two thresholds named *top_thr* and *bottom_thr* are derived based on the histogram of the heights (i.e., z values) of points. If the height of a point is smaller than *bottom_thr* or greater than *top_thr*, it is labeled as an object point, otherwise, it is classified as a road point. The derivation of the thresholds is based on the histogram:

1. Obtain the histogram for the points with the height being in the range of (B, T) with a specific histogram bin size. B and T are two preset thresholds to select the points to be used for calculating *top_thr* and *bottom_thr*.
2. Calculate the standard deviation (*std*) of the histogram.
3. Find the peak bin (P) of the histogram where the number of points achieve the maximum.
4. The thresholds are calculated as follows:

$$top_thr = \mathrm{Min}(T, P + 1.5*std)$$

$$bottom_thr = \mathrm{Max}(B, P - 1.5*std)$$

In classification process, the encoder scans block by block to derive the point array, *pc_likely_world*, which contains object points being used in the global motion estimation process. For a point in the current block, if its height value fit the two-thresholds conditions and there is at least one point in the reference block (collocated block with an extension in the reference frame), this point will be added to *pc_likely_world*.

After the points are classified into the road and objects, only the points with object label are used in the global motion estimation process. The Least Mean Square (LMS) algorithm is used to estimate the global motion. The number of LMS iterations for the estimation and the number of samples used for each iteration of the estimation is preset. In each iteration, the samples are

subsampled from points in *pc_likely_world*. And for each sample, its responding reference sample point will be found in reference frame, which is the point with the smallest distance from the current sample. The global motion matrix is calculated by LMS algorithm with the samples and reference samples. Then the global motion matrix is applied to the object points in reference frame to get the reference frame with motion compensation.

4. Problems

The designs for global motion estimation described above have the following problems:

1. The classification results are not accuracy enough. Some object points with low height value may be labeled as road points, e.g., some object points at the junction of buildings and road. The wrong classification will reduce the accuracy of global motion estimation.
2. The complexity of classification process is high. The classification process that the encoder scans block by block introduces many iterations. These iterations cause high complexity.
3. The matching results between the samples and the reference samples are not accuracy enough. The samples are subsampled from the object points in the current frame. The reference samples are generated by scanning all points in the reference frame, including both object points and road points. Therefore, the reference sample of a sample may be labeled as a road point in the reference frame while the sample is always labeled as an object point. The wrong matching results will reduce the accuracy of global motion estimation.
4. The motion between adjacent frames of the lidar point cloud sequence is mainly manifested as the translation of the object points, while the rotation is small. In current inter-EM, the encoder uses LMS algorithm to obtain the rotation matrix and the translation vector. Due to the small influence of rotation, some bits and calculations are wasted on processing rotation matrix.

5. Exemplary Embodiemnts

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

1) N thresholds may be used to classify the points into the M classes, such as two (M=2) classes: object points and road points wherein N is greater than M.
    a. In one example, N = 5, M = 2.
        i. Alternatively, furthermore, denote the five thresholds by *higher_top_thr*, *lower_top_thr*, *higher_bottom_thr*, *lower_bottom_thr* and *height_change_thr*. Assume that *higher_top_thr* >= *lower_top_thr* >= *higher_bottom_thr* >= *lower_bottom_thr*, the following may be applied in order:
            (1) If the height of a point is smaller than *lower_bottom_thr* or greater than *higher_top_thr*, it is labeled as an object point.
            (2) Otherwise, if the height of a point is between *lower_top_thr* and *higher_bottom_thr*, it is labeled as a road point.
            (3) Otherwise, the point is labeled as an object point if the max height difference between the point and its neighbors is higher than *height_change_thr*.
            (4) Otherwise, the point is labeled as a road point.
    b. In one example, the one or multiple thresholds are pre-defined or derived (e.g., based on the of the height (i.e., z values) of points).
    c. In one example, the one or multiple thresholds may be different from one point to another point.
    d. In one example, the one or multiple thresholds may be signaled from the encoder to the decoder.
    e. In one example, the one or multiple thresholds may be generated at encoder.
    f. In one example, the one or multiple thresholds may be generated at decoder.
    g. In one example, the one or multiple thresholds may be generated based on at least one of the current frame and other frames.
        i. In one example, the other frame may be a reference frame.
        ii. In one example, indication of the other frame may be signalled in the bitstream.
        iii. In one example, the other frame may be pre-defined, e.g., the reference frame with reference picture index being equal to 0.
    h. In one example, a histogram of geometry features (such as the height value of points) may be used to generate the thresholds.
        i. In one example, the histogram may be generated with all or partial points in current picture with a specific histogram bin size.

(1) Alternatively, it may be generated with partial or all previously coded points before current point.

ii. In one example, whether a point can be used to generate the histogram may be depended on the characteristics of the point.

(1) In one example, if the height value of the point is within a preset range, the point may be used.

iii. In one example, the standard deviation (*std*) of the histogram may be calculated to generate the thresholds.

i. In one example, a threshold may be firstly processed, e.g., being clipped, before being used.

j. Any geography or feature value such as the height value for a point may be compared with at least one of the thresholds to classify the point.

k. Any geography or feature value such as the height difference value between a point and the neighbor points may be compared with at least one of the thresholds to classify the point.

l. For above examples, they may also be applicable to current design wherein two thresholds are utilized to category a point to one of the two classes.

2) It is proposed to scan each planar region to perform the classification.

a. In one example, there may be one or multiple reference frames for the current frame.

b. In one example, a planar region is a cuboid space. Fig. 4 is a schematic diagram 400 illustrating an example of planar regions and blocks in a point cloud. The points in a point cloud can be clustered into one or multiple planar regions according to their plane coordinates. A planar region can be divided into one or multiple blocks, which is usually a cube space, based on the height coordinates.

c. In one example, the reference points are in a reference frame.

d. In one example, the reference frame is composed of one or multiple planar regions, and a reference point belongs to one or multiple reference planar regions.

e. In one example, a planar region in the current frame corresponds to one or zero reference planar regions in the reference frame.

f. In one example, whether a point in the scanned planar region can be classified (such as use N thresholds to determine whether it is an object point) may depend on the corresponding reference planar region.

i. In one example, a point in the scanned planar region can only be classified, if the scanned planar region corresponds to one reference planar region and there is at least one reference point belonging to the reference planar region.

g. In one example, how to classify a point that can be classified may depend on classification conditions, such as to use N thresholds to determine whether it is an object point.

3) It is proposed to use multiple types of process units in classification process.

a. In one example, there may be one or multiple types of process units (such as cubic block, planar region) that are used in classification process.

b. In one example, one type of process unit may be used for all reference frames.

c. In one example, the types of process units may be inconsistent for at least two reference frames.

i. In one example, the type of process unit may be selected as the type that minimizes the error (such as mean square error, mean absolute error, root mean square error) between the current frame and the reference frame with motion compensation.

4) It is proposed to scan each point and reference point to perform the classification.

a. In one example, there may be one or multiple reference frames for the current frame.

b. In one example, the reference points are in a reference frame.

c. In one example, the reference frame is composed of one or multiple reference space units (such as blocks or planar regions), and a reference point belongs to one or multiple reference space units.

d. In one example, a reference space unit is marked (such as marked to be occupied) if at least one reference point belonging to it is scanned.

e. In one example, a point in the current frame may correspond to one or zero reference space units.

f. In one example, whether a point in the current frame can be classified (such as to determine whether it is an object point) may depend on its corresponding reference space unit.

i. In one example, a scanned point can only be classified, if the scanned point corresponds to one reference space unit and there is at least one reference point belonging to the reference space unit.

g. In one example, how to classify a point that can be classified may depend on classification conditions, such as to use N thresholds to determine whether it is an object point.

5) It is proposed to apply an estimated global motion to all points in a reference frame.

a. In one example, the global motion matrix may be calculated by LMS algorithm with the samples and reference samples.

b. In one example, a global motion matrix is applied to all points in the reference frame to get the reference frame with motion compensation.

6) It is proposed to generate the reference samples using (e.g., by scanning) the points with the same label in a reference frame.

a. In one example, both samples in current frame and reference samples may be used for global motion estimation.

b. In one example, one or more points (such as the object points) in the current frame may be downsampled to generate samples for global motion estimation.

c. In one example, a sample corresponds to one reference sample. And its reference sample may be found in the points with the same label (such as the object points) in the reference frame.

d. Any geography or feature value such as the Euclidean distance between two points may be used to classify the point.

i. In one example, a point in the reference frame can be marked as the reference sample of a sample, if it has the least Euclidean distance from the sample among all points in the reference frame.

7) It is proposed to use one or multiple rotation matrices to perform the global motion estimation/motion compensation process.

a. In one example, at least one rotation matrix may be a fixed matrix.

i. In one example, only one rotation matrix is used and the rotation matrix is fixed.

b. In one example, a rotation matrix may be pre-defined or indicated in a bitstream or derived-on-the-fly.

c. In one example, a rotation matrix may be identical for at least two reference frames.

d. In one example, a rotation matrix may be fixed for all reference frames.

i. In one example, the rotation matrix may be fixed as $N \times N$ identity matrix (e.g., $N = 3$), which means there is no rotation for the reference frame.

e. In one example, at least one syntax element may be signaled to indicate whether a rotation matrix is shared by two reference frames.

f. In one example, at least one syntax element may be signaled to indicate whether a fixed rotation matrix is used for all reference frames.

g. In one example, at least one syntax element may be signaled to indicate whether a fixed rotation matrix is used for a reference frame.

h. In one example, a rotation matrix may be signaled in a predictive way.

i. In one example, the translation vector may be calculated for each reference frame based on the samples of the current frame and reference samples.

    i. In one example, the translation vector may be set to the vector that minimizes the error (such as mean square error, mean absolute error, root mean square error) between the samples and the reference samples with motion compensation.

j. In one example, a global motion matrix may be composed of or represented or derived by a rotation matrix and a translation vector and the global motion matrix is utilized in the inter-prediction process.

k. In one example, the global motion matrix may be generated at encoder.

l. In one example, the global motion matrix may be generated at decoder.

m. In one example, one or multiple translation vectors may be used for one reference frame.

n. In one example, at least one indication of the translation vector may be signaled from the encoder to the decoder.

    i. In one example, the indication may indicate which translation vector associated with a rference frame may be used for video unit.

        (1) The video unit may be a sequence/frame/slice/tile.

        (2) The video unit may be a sub-region within a frame.

    ii. In one example, the translation vector may be signaled in a video unit (covering one or more points) level.

        (1) Alternatively, furthermore, the video unit may be a sequence/frame/slice/tile.

        (2) Alternatively, furthermore, the video unit may be a sub-region within a frame.

o. In one example, the translation vector may be signaled in a predictive way.

    i. In one example, a first element of a first translation vector may be predicted by a second element of first translation vector.

    ii. In one example, a first element of a first translation vector may be predicted by a first element of second translation vector.

        (1) In one example, the two translation vectors may be associated with a same reference frame.

        (2) In one example, the two translation vectors may be associated with two different reference frames.

8) There may be one or zero global motion matrices for a reference frame to perform motion compensation.
    a. In one example, the global motion matrix may be generated at encoder.
    b. In one example, the global motion matrix may be generated at decoder.
9) A piece of information disclosed in this document, which is signaled from an encoder to a decoder, may be binarized with a first coding method.
    a. The first coding method may be fixed length coding, exponential golomb coding, unary coding, truncated unary coding, signed unary coding, signed truncated unary coding, etc.
    b. In one example, the piece of information may be coded by arithmetic coding.
        i. For example, arithmetic coding may be contex-based.
    c. The piece of information may be signaled at any video unit or at any level, such as sequence header/picture header/slice header/block or any other units in PCC.

6. Embodiments

1) This embodiment describes an example of how to classify a point cloud frame into object points and road points by using five thresholds method.

In the example, the input is the current frame, PC_{cur} and the reference frame, PC_{ref}. The output is the point array, *pc_likely_world*, which contains the object points that are used for global motion estimation.

Firstly, generate the five thresholds for road/object classification:

1. Obtain the histogram for the values of the points in PC_{cur} with the height being in the range of (B, T) with a specific histogram bin size. Where B is preset to -4000 and T is preset to -500. The values of B and T depend on the parameters of the point cloud acquisition device
2. Calculate the standard deviation (*std*) of the histogram.
3. Find the peak bin (P) of the histogram where the number of points achieve the maximum.

4. The thresholds are calculated as follows:

$$lower_top_thr = P + 1*std$$

$$higher_bottom_thr = P - 1*std$$

$$higher_top_thr = \mathrm{Min}(T, P + 1.5*std)$$

$$lower_bottom_thr = \mathrm{Max}(B, P - 1.5*std)$$

$$height_change_thr = 4*std$$

Secondly, scan block by block to derive the point array, *pc_likely_world,* which contains object points being used in the global motion estimation process. The scanning process takes the following steps:

1. The bounding box space of PC_{cur} is divided into multiple blocks, and the block size of each block is *bsize* * *bsize* * *bsize*. *bsize* is preset to 4096. Fig. 5 is a schematic diagram 500 illustrating an example of generation blocks for a point cloud.
2. For each block in the current frame, generate its corresponding reference block in the bounding box space of PC_{ref}. The reference block is the collocated block of current block with an extension in the reference frame and its block size is (*bsize*+2**delta*)*(*bsize*+2**delta*)*(*bsize*+2**delta*). *delta* is preset to 1000. Fig. 6 is a schematic diagram 600 illustrating an example of generation reference blocks for current block.
3. For each block, traverse all the points in the current frame and record the points which is in the current block in an array, *current_block*. Move to step 4.
4. Traverse all the points in the reference frame and count the number of points which is in its corresponding reference block. If the number is zero, return to step 3 and scan next block. Otherwise, move to step 5.
5. For each point in the *current_block*, if its height value fits any one of the following conditions, it will be added to *pc_likely_world*. Return to step 3 and scan next block.
    (1) The height of the point is smaller than *lower_bottom_thr*.
    (2) The height of the point is greater than *higher_top_thr*.
    (3) The height of the point is between *lower_top_thr* and *higher_top_thr*, and the max height difference between the point and its neighbors (the point and its neighbors should belong to the same block) is higher than *height_change_thr*.
    (4) The height of the point is between *lower_bottom_thr* and *higher_bottom*_thr, and the max height difference between the point and its neighbors (the point and its neighbors should belong to the same block) is higher than *height_change_thr*.

2) This embodiment describes an example of how to classify a point cloud frame into object points and road points by using a point by point scanning process. The space unit is block in this example.

In the example, the input is the current frame, PC_{cur} and the reference frame, PC_{ref}. The output is the point array, *pc_likely_world*, which contains the object points that are used for global motion estimation.

Firstly, generate the two thresholds for road/object classification:

1. Obtain the histogram for the values of the points in PC_{cur} with the height being in the range of (*B, T*) with a specific histogram bin size. Where *B* is preset to -4000 and *T* is preset to -500. The values of *B* and *T* depend on the parameters of the point cloud acquisition device
2. Calculate the standard deviation (*std*) of the histogram.
3. Find the peak bin (*P*) of the histogram where the number of points achieve the maximum.
4. The thresholds are calculated as follows:

$$top_thr = P + 1.5*std$$

$$bottom_thr = P - 1.5*std$$

Secondly, scan point by point to derive the point array, *pc_likely_world,* which contains object points being used in the global motion estimation process. The scanning process takes the following steps:

1. Count the number of blocks in the current frame, *blocknum*:

$$Bnum = maxBB_Scalled / bsize$$

$$blocknum = Bnum*Bnum*Bnum$$

maxBB_Scalled is the maximum side length of bounding box of PC_{cur}. *bsize* is the side length of a block, which is preset to 4096.

2. A bool array is generated to indicate the occupancy information of each reference block in PC_{ref}, *ref_block_xyz*. It is resized to *blocknum.*

3. For each point (*x*, *y*, *z*) in PC_{ref}, calculate the indexes of its corresponding reference blocks.

$$ID_0 = \left(\frac{x+delta}{bsize} * Bnum + \frac{y+delta}{bsize}\right) * Bnum + \frac{z+delta}{bsize}$$

$$ID_1 = \left(\frac{x+delta}{bsize} * Bnum + \frac{y+delta}{bsize}\right) * Bnum + \frac{z-delta}{bsize}$$

$$ID_2 = \left(\frac{x+delta}{bsize} * Bnum + \frac{y-delta}{bsize}\right) * Bnum + \frac{z+delta}{bsize}$$

$$ID_3 = \left(\frac{x+delta}{bsize} * Bnum + \frac{y-delta}{bsize}\right) * Bnum + \frac{z-delta}{bsize}$$

$$ID_4 = \left(\frac{x-delta}{bsize} * Bnum + \frac{y+delta}{bsize}\right) * Bnum + \frac{z+delta}{bsize}$$

$$ID_5 = \left(\frac{x-delta}{bsize} * Bnum + \frac{y+delta}{bsize}\right) * Bnum + \frac{z-delta}{bsize}$$

$$ID_6 = \left(\frac{x-delta}{bsize} * Bnum + \frac{y-delta}{bsize}\right) * Bnum + \frac{z+delta}{bsize}$$

$$ID_7 = \left(\frac{x-delta}{bsize} * Bnum + \frac{y-delta}{bsize}\right) * Bnum + \frac{z-delta}{bsize}$$

delta is preset to 1000. In *ref_block_xyz*, the elements with these indexes are marked as occupied.

4. For each point (*x,y,z*) in PC_{cur}, calculate the index of its corresponding reference block:

$$ID = \left(\frac{x}{bsize} * Bnum + \frac{y}{bsize}\right) * Bnum + \frac{z}{bsize}$$

If the element corresponding with the index was not found in *ref_block_xyz* or was not marked as occupied, scan next point.

5. If the height value of the point fits any one of the following conditions, it will be added to *pc_likely_world*:

(1) The height of the point is smaller than *bottom_thr*.
(2) The height of the point is greater than *top_thr*.

3) This embodiment describes an example of how to classify a point cloud frame into object points and road points by using a planar region based scanning process.

In the example, the input is the current frame, PC_{cur} and the reference frame, PC_{ref}. The output is the point array, *pc_likely_world*, which contains the object points that are used for global motion estimation.

Firstly, generate the two thresholds for road/object classification:

1. Obtain the histogram for the values of the points in PC_{cur} with the height being in the range of (*B, T*) with a specific histogram bin size. Where *B* is preset to -4000 and *T* is preset to -500. The values of *B* and *T* depend on the parameters of the point cloud acquisition device
2. Calculate the standard deviation (*std*) of the histogram.
3. Find the peak bin (*P*) of the histogram where the number of points achieve the maximum.
4. The thresholds are calculated as follows:

$$top_thr = P + 1*std$$

$$bottom_thr = P - 1*std$$

Secondly, scan each planar region to derive the point array, *pc_likely_world*, which contains object points being used in the global motion estimation process. The scanning process takes the following steps:

1. The bounding box space of PC_{cur} is divided into multiple planar regions, and the region size of each block is *bsize* * *bsize*. *bsize* is preset to 4096. Fig. 7 is a schematic diagram 700 illustrating an example of generation planar regions for a point cloud.
2. For each planar region in the current frame, generate its corresponding reference planar region in the bounding box space of PC_{ref}. The reference planar region is the collocated planar region of current planar region with an extension in the reference frame and its region size is (*bsize*+2**delta*)*(*bsize*+2**delta*). *delta* is preset to 1000.

Fig. 8 is a schematic diagram 800 illustrating an example of generation reference planar regions for current planar region

3. For each planar region, traverse all the points in the current frame and record the points which is in the current planar region in an array, *current_region*.
4. Traverse all the points in the reference frame and count the number of points which is in its corresponding reference planar region. If the number is zero, scan next planar region.
5. For each point in the *current_region*, if its height value fits any one of the following conditions, it will be added to *pc_likely_world*:
   (1) The height of the point is smaller than *lower_bottom_thr*.
   (2) The height of the point is greater than *higher_top_thr*.

4) This embodiment describes an example of how to classify a point cloud frame into object points and road points by using a point by point scanning process. The space unit is planar region in this example.

In the example, the input is the current frame, PC_{cur} and the reference frame, PC_{ref}. The output is the point array, *pc_likely_world*, which contains the object points that are used for global motion estimation.

Firstly, generate the two thresholds for road/object classification:

1. Obtain the histogram for the values of the points in PC_{cur} with the height being in the range of (*B, T*) with a specific histogram bin size. Where *B* is preset to -4000 and *T* is preset to -500. The values of *B* and *T* depend on the parameters of the point cloud acquisition device

2. Calculate the standard deviation (*std*) of the histogram.

3. Find the peak bin (*P*) of the histogram where the number of points achieve the maximum.

4. The thresholds are calculated as follows:

$$top_thr = P + 1.5*std$$

$$bottom_thr = P - 1.5*std$$

Secondly, scan point by point to derive the point array, *pc_likely_world,* which contains object points being used in the global motion estimation process. The scanning process takes the following steps:

1. Count the number of planar regions in the current frame, *Region_num*:

$$Rnum = maxBB_Scalled / bsize$$

$$Region_num = Rnum * Rnum$$

maxBB_Scalled is the maximum side length of bounding box of PC_{cur}. *bsize* is the side length of a planar region, which is preset to 4096.

2. A bool array is generated to indicate the occupancy information of each reference planar region in PC_{ref}, *ref_planar_region*. It is resized to *Region_num*.
3. For each point (*x, y, z*) in PC_{ref}, calculate the indexes of its corresponding reference planar regions.

$$ID_0 = \frac{x + delta}{bsize} * Rnum + \frac{y + delta}{bsize}$$

$$ID_1 = \frac{x + delta}{bsize} * Rnum + \frac{y - delta}{bsize}$$

$$ID_2 = \frac{x - delta}{bsize} * Rnum + \frac{y + delta}{bsize}$$

$$ID_3 = \frac{x - delta}{bsize} * Rnum + \frac{y - delta}{bsize}$$

delta is preset to 1000. In *ref_planar_region*, the elements with these indexes are marked as occupied.

4. For each point (*x, y, z*) in PC_{cur}, calculate the index of its corresponding reference planar region:

$$ID = \frac{x}{bsize} * Rnum + \frac{y}{bsize}$$

If the element corresponding with the index was not found in *ref_planar_region* or was not marked as occupied, scan next point.

5. If the height value of the point fits any one of the following conditions, it will be added to *pc_likely_world*:
   (1) The height of the point is smaller than *bottom_thr*.
   (2) The height of the point is greater than *top_thr*.

5) This embodiment describes an example of how to obtain and process the global motion matrix by fixing the rotaton matrix.

In the example, the input is the sample point array, PC_{sample} and the reference sample point array, $PC_{refsample}$. The output is the global motion matrix, *GM*, which is composed of a rotation matrix, *RM* and a translation vector, *TV*.

Firstly, for each sample, P_i (x_i, y_i, z_i) and their corresponding reference sample, P'_i (x'_i, y'_i, z'_i), calculate the differences between them:

$$Dx_i = x_i - x'_i$$

$$Dy_i = y_i - y'_i$$

$$Dz_i = z_i - z'_i$$

Secondly, calculate the sum of the differences:

$$Sumdx = \sum Dx_i$$

$$Sumdy = \sum Dy_i$$

$$Sumdz = \sum Dz_i$$

Thirdly, generate the translation vector, *TV* and the rotation matrix, *RM* (fixed as a 3×3 identity matrix). Where *SumP* represents the number of sample points.

$$TV = (a, b, c) = \left(\frac{Sumdx * 2}{SumP}, \frac{Sumdy * 2}{SumP}, \frac{Sumdy * 2}{SumP}\right)$$

$$RM = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Finally, generate the global motion matrix, *GM*:

$$GM = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ a & b & c \end{pmatrix}$$

Because the rotation matrix is fixed for all frames, only the translation vector for each reference frame should be coded and signaled to the decoder.

6) This embodiment describes an example of how to obtain and process the global motion matrix by fixing the rotaton matrix.

In the example, the input is the sample point array, PC_{sample} and the reference sample point array, $PC_{refsample}$. The output is the global motion matrix, *GM*, which is composed of a rotation matrix, *RM* and a translation vector, *TV*.

Firstly, for each sample, P_i (x_i, y_i, z_i) and their corresponding reference sample, P'_i (x'_i, y'_i, z'_i), calculate the differences between them:

$$Dx_i = x_i - x'_i$$

$$Dy_i = y_i - y'_i$$

$$Dz_i = z_i - z'_i$$

Secondly, calculate the sum of the differences:

$$Sumdx = \sum Dx_i$$

$$Sumdy = \sum Dy_i$$

$$Sumdz = \sum Dz_i$$

Thirdly, generate the translation vector, *TV* and the rotation matrix, *RM* (fixed as a 3×3 identity matrix). Where *SumP* represents the number of sample points.

$$TV = (a, b, c) = \left(\frac{Sumdx}{SumP}, \frac{Sumdy}{SumP}, \frac{Sumdz}{SumP}\right)$$

$$RM = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Finally, generate the global motion matrix, *GM*:

$$GM = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ a & b & c \end{pmatrix}$$

Because the rotation matrix is fixed for all frames, only the translation vector for each reference frame should be coded and signaled to the decoder.

[0062] The embodiments of the present disclosure are related to classification-based global motion estimation methods for point cloud coding. As used herein, the term "point cloud sequence" may refer to a sequence of one or more point clouds. The term "frame" may refer to a point cloud in a point cloud sequence.

[0063] Fig. 9 illustrates a flowchart of a method 900 for point cloud coding in accordance with some embodiments of the present disclosure. The method 900 may be implemented during a conversion between a current frame of a point cloud sequence and a bitstream of the point cloud sequence. As shown in Fig. 9, the method 900 starts at 902, where motion compensation is performed on a reference frame of the current frame based on a set of rotation matrixes, so as to obtain a compensated reference frame of the current frame. The set of rotation matrixes comprise at least one fixed rotation matrix or a plurality of rotation matrixes. By way of example, the set of rotation matrixes may comprise one fixed rotation matrix, e.g., a 3×3 identity matrix. A global motion matrix may be generated based on the 3×3 identity matrix, and the motion compensation may be performed on the reference frame based on the generated global motion matrix, so as to obtain the compensated reference frame. It should be understood that the above illustrations and/or examples are described merely for purpose of description. The scope of the present disclosure is not limited in this respect.

[0064] At 904, the conversion is performed based on the compensated reference frame. In some embodiments the conversion may include encoding the current frame into the bitstream. Alternatively or additionally, the conversion may include decoding the current frame from the bitstream.

[0065] According to the method 900, a fixed rotation matrix or a plurality of rotation matrixes may be used for performing motion compensation. If a fixed rotation matrix is used, the proposed method can advantageously reduce bits and resources wasted on processing rotation matrix and improve coding efficiency. If a plurality of rotation matrixes are used, the proposed method can advantageously improve the accuracy of motion compensation and coding efficiency.

[0066] In some embodiments, at least one of the set of rotation matrixes may be predefined. Alternatively, at least one of the set of rotation matrixes may be indicated in the bitstream. Alternatively, at least one of the set of rotation matrixes may be determined on-the-fly. Additionally or alternatively, one of the set of rotation matrixes may be fixed as an identity matrix.

[0067] In some embodiments, one of the set of rotation matrixes may be used for performing motion compensation on a further reference frame of the current frame. Additionally or alternatively, one of the set of rotation matrixes may be used for performing motion compensation on all of reference frames of the current frame.

[0068] In some embodiments, at least one syntax element may be indicated in the bitstream. the at least one syntax element may indicate whether one of the set of rotation matrixes is used for performing motion compensation on a further reference frame of the current frame. Additionally or alternatively, at least one syntax element may be indicated in the bitstream. The at least one syntax element may indicate whether one of the set of rotation matrixes is used for performing motion compensation on all of reference frames of the current frame. In another example, at least one syntax element may be indicated in the bitstream. The at least one syntax element may indicate whether a fixed rotation matrix is used for the reference frame.

[0069] In some embodiments, at least one of the set of rotation matrixes may be determined based on a predictive coding method. By way of example, a residual between a second rotation matrix and a first rotation matrix may be determined and signaled. The second rotation matrix may be determined based on the residual and the first rotation matrix. It should be understood that the above illustrations and/or examples are described merely for purpose of description. The scope of the present disclosure is not limited in this respect.

[0070] In some embodiments, at 902, a global motion matrix for the current frame may be determined based on the set of rotation matrixes and a set of translation vectors for the reference frame. The motion compensation may be performed on the reference frame based on the global motion matrix, so as to obtain the compensated reference frame. In one example, a translation vector in the set of translation vectors may be determined based on samples of the current frame and reference samples of the reference frame. For example, the translation vector may be a vector minimizing an error metric between the samples and the reference samples with motion compensation. The error metric may comprise mean square error, mean absolute error, root mean square error, or the like. In some embodiments, the global motion matrix may be generated at an encoder. Additionally or alternatively, the global motion matrix may be generated at a decoder.

[0071] In some embodiments, the set of translation vectors may comprise a plurality of translation vectors. By using a plurality of rotation matrixes, the proposed method can advantageously improve the accuracy of motion compensation and coding efficiency.

[0072] In some embodiments, at least one indication for the set of translation vectors may be indicated in the bitstream. In one example, the at least one indication indicates a translation vector used for a target unit of the reference frame. By way of example, the target unit may comprise a sequence, a frame, a slice, a tile, or a sub-region in the reference frame.

[0073] In some embodiments, the set of translation vectors may be associated with a target unit of the reference frame, and the set of translation vectors may be indicated in the bitstream. By way of example, the target unit may comprise one of the following: a sequence, a frame, a slice, a tile, or a sub-region in the reference frame. In one example, the set of translation vectors may be signaled at a sequence level.

[0074] In some embodiments, at least one of the set of translation vectors may be determined based on a predictive coding method. By way of example, a first element of a first translation vector of the set of translation vectors may be determined based on a second element of the first translation vector. Alternatively, a first element of a first translation vector of the set of translation vectors may be determined based on a second element of a second translation vector associated with a second reference frame of the current frame. In one example, the second reference frame may be the same as the reference frame. In another example, the second reference frame may be different from the reference frame.

[0075] In some embodiments, the motion compensation may be performed without global motion compensation. By way of example, only local motion compensation is performed on the reference frame, so as to obtain the compensated reference frame.

[0076] In some embodiments, a piece of information for the convention may be binarized based on a coding method. The piece of information may be indicated in the bitstream. In one example, the coding method may be a fixed length coding method. In another example, the coding method may be an exponential golomb coding method. Alternatively, the coding method may be a unary coding method. In a further example, the coding method may be a truncated unary coding method or a signed unary coding method. In yet another example, the coding method may be a signed truncated unary coding method. Alternatively, the coding method may be an arithmetic coding method. In another example, the arithmetic coding method may be context-based. It should be understood that the possible implementations of the coding method described here are merely illustrative and therefore should not be construed as limiting the present disclosure in any way.

[0077] In some embodiments, the piece of information may be indicated at a sequence header. Alternatively, the piece of information may be indicated at a picture header. In another example, the piece of information may be indicated at a slice header. In a further example, the piece of information may be indicated at a block level. It should be understood that the above illustrations and/or examples are described merely for purpose of description. The scope of the present disclosure is not limited in this respect.

[0078] In some embodiments, a bitstream of a point cloud sequence may be stored in a non-transitory computer-readable recording medium. The bitstream of the point cloud sequence can be generated by a method performed by a point cloud processing apparatus. According to the method, motion compensation is performed on a reference frame of the current frame based on a set of rotation matrixes, so as to obtain a compensated reference frame of the current frame. A bitstream of the current frame may be generated based on the compensated reference frame.

[0079] In some embodiments, motion compensation is performed on a reference frame of the current frame based on a set of rotation matrixes, so as to obtain a compensated reference frame of the current frame. A bitstream of the current frame may be generated based on the compensated reference frame. The bitstream may be stored in a non-transitory computer-readable recording medium.

[0080] Fig. 10 illustrates a flowchart of another method 1000 for point cloud coding in accordance with some embodiments of the present disclosure. The method 1000 may be implemented during a conversion between a current frame of a point cloud sequence and a bitstream of the point cloud sequence. As shown in Fig. 10, the method 1000 starts at 1002, where points in the point cloud sequence are classified based on a plurality types of process units for global motion estimation of the current frame. By way of example, the plurality types of process units may comprise a cubic block and/or a planar region. The planar may be three-dimensional and having a height equal to a height of a bounding box of the current frame. With reference to Fig. 4, the cuboid 410 may corresponds to a planar region. In one example, the points in the current frame may be assigned to a set of blocks, while points in the reference frame may be assigned to a set of planar regions. Points in the current frame and the reference frame may be classified based on the above assignment. It should be understood that the above illustrations and/or examples are described merely for purpose of description. The scope of the present disclosure is not limited in this respect.

[0081] At 1004, the conversion is performed based on the classification. In some embodiments the conversion may include encoding the current frame into the bitstream. Alternatively or additionally, the conversion may include decoding the current frame from the bitstream.

[0082] The method 1000 utilizes various types of process units to perform the classification process. Compared with the conventional solution where only a block is used, the proposed method can advantageously reduce the number of iterations in the classification process and thus reduce the complexity of classification process and improve the coding efficiency.

[0083] In some embodiments, the global motion estimation of the current frame may be performed based on a plurality of reference frames. At 1002, points in the plurality of reference frames are classified based on one of the plurality types of process units. By way of example, the points in a first reference frame may be assigned to a set of blocks, and points in a second reference frame may be assigned to a set of blocks. Points in the first reference frame and the second reference frame may be classified based on the above assignment. It should be understood that the above illustrations and/or examples are described merely for purpose of description. The scope of the present disclosure is not limited in this respect.

[0084] In some embodiments, the global motion estimation of the current frame may be performed based at least on a first reference frame and a second reference frame, and the plurality types of process units may comprise a first type of process unit and a second type of process unit. At 1002, points in the first reference frame may be classified based on the first type of process unit, and points in the second reference frame may be classified based on the second type of process unit. By way of example, the points in the first reference frame may be assigned to a set of blocks, while points in the second reference frame may be assigned to a set of planar regions. Points in the first reference frame and the second reference frame may be classified based on the above assignment. It should be understood that the above illustrations and/or examples are described merely for purpose of description. The scope of the present disclosure is not limited in this respect.

[0085] In some embodiments, the first type of process unit may be a type of process unit minimizing an error metric between the current frame and the first reference frame with motion compensation. For example, the error metric may comprise mean square error, mean absolute error, root mean square error, or the like.

[0086] In some embodiments, a bitstream of a point cloud sequence may be stored in a non-transitory computer-readable recording medium. The bitstream of the point cloud sequence can be generated by a method performed by a point cloud processing apparatus. According to the method, points in the point cloud sequence are classified based on a plurality types of process units for global motion estimation of the current frame. A bitstream of the current frame may be generated based on the classification.

[0087] In some embodiments, points in the point cloud sequence are classified based on a plurality types of process units for global motion estimation of the current frame. A bitstream of the current frame may be generated based on the classification. The bitstream may be stored in a non-transitory computer-readable recording medium.

[0088] Implementations of the present disclosure can be described in view of the following clauses, the features of which can be combined in any reasonable manner.

[0089] Clause 1. A method for point cloud coding, comprising: obtaining, during a conversion between a current frame of a point cloud sequence and a bitstream of the point cloud sequence, a compensated reference frame of the current frame by performing motion compensation on a reference frame of the current frame based on a set of rotation matrixes comprising at least one fixed rotation matrix or a plurality of rotation matrixes; and performing the convention based on the compensated reference frame.

[0090] Clause 2. The method of clause 1, wherein at least one of the set of rotation matrixes is predefined, or at least one of the set of rotation matrixes is indicated in the bitstream, or at least one of the set of rotation matrixes is determined on-the-fly.

[0091] Clause 3. The method of any of clauses 1-2, wherein one of the set of rotation matrixes is used for performing motion compensation on a further reference frame of the current frame.

[0092] Clause 4. The method of any of clauses 1-3, wherein one of the set of rotation matrixes is used for performing motion compensation on all of reference frames of the current frame.

[0093] Clause 5. The method of any of clauses 1-4, wherein one of the set of rotation matrixes is fixed as an identity matrix.

[0094] Clause 6. The method of any of clauses 1-5, wherein at least one syntax element is indicated in the bitstream, the at least one syntax element indicating whether one of the set of rotation matrixes is used for performing motion compensation on a further reference frame of the current frame.

[0095] Clause 7. The method of any of clauses 1-5, wherein at least one syntax element is indicated in the bitstream, the at least one syntax element indicating whether one of the set of rotation matrixes is used for performing motion compensation on all of reference frames of the current frame.

[0096] Clause 8. The method of any of clauses 1-5, wherein at least one syntax element is indicated in the bitstream, the at least one syntax element indicating whether a fixed rotation matrix is used for the reference frame.

[0097] Clause 9. The method of any of clauses 1-8, wherein at least one of the set of rotation matrixes is determined based on a predictive coding method.

[0098] Clause 10. The method of any of clauses 1-9, wherein obtaining the compensated reference frame comprises: determining a global motion matrix for the current frame based on the set of rotation matrixes and a set of translation vectors for the reference frame; and performing the motion compensation on the reference frame based on the global motion matrix to obtain the compensated reference frame.

[0099] Clause 11. The method of clause 10, wherein a translation vector in the set of translation vectors is determined based on samples of the current frame and reference samples of the reference frame.

[00100] Clause 12. The method of clause 11, wherein the translation vector is a vector minimizing an error metric between the samples and the reference samples with motion compensation.

[00101] Clause 13. The method of clause 12, wherein the error metric comprises one of the following: mean square error, mean absolute error, or root mean square error.

[00102] Clause 14. The method of any of clauses 10-13, wherein the global motion matrix is generated at an encoder.

[00103] Clause 15. The method of any of clauses 10-13, wherein the global motion matrix is generated at a decoder.

[00104] Clause 16. The method of any of clauses 10-15, wherein the set of translation vectors comprises a plurality of translation vectors.

[00105] Clause 17. The method of any of clauses 10-16, wherein at least one indication for the set of translation vectors is indicated in the bitstream.

[00106] Clause 18. The method of clause 17, wherein the at least one indication indicates a translation vector used for a target unit of the reference frame.

[00107] Clause 19. The method of clause 18, wherein the target unit comprises one of the following: a sequence, a frame, a slice, a tile, or a sub-region in the reference frame.

[00108] Clause 20. The method of any of clauses 10-19, wherein the set of translation vectors are associated with a target unit of the reference frame, and the set of translation vectors are indicated in the bitstream.

[00109] Clause 21. The method of clause 20, wherein the target unit comprises one of the following: a sequence, a frame, a slice, a tile, or a sub-region in the reference frame.

[00110] Clause 22. The method of any of clauses 10-19, wherein at least one of the set of translation vectors is determined based on a predictive coding method.

[00111] Clause 23. The method of clause 22, wherein a first element of a first translation vector of the set of translation vectors is determined based on a second element of the first translation vector.

[00112] Clause 24. The method of clause 22, wherein a first element of a first translation vector of the set of translation vectors is determined based on a second element of a second translation vector associated with a second reference frame of the current frame.

[00113] Clause 25. The method of clause 24, wherein the second reference frame is the same as or different from the reference frame.

[00114] Clause 26. The method of any of clauses 1-9, wherein the motion compensation is performed without global motion compensation.

[00115] Clause 27. The method of any of clauses 1-26, wherein a piece of information for the convention is binarized based on a coding method, the piece of information being indicated in the bitstream.

[00116] Clause 28. The method of clause 27, wherein the coding method is one of the following: a fixed length coding method, an exponential golomb coding method, a unary coding method, a truncated unary coding method, a signed unary coding method, or a signed truncated unary coding method.

[00117] Clause 29. The method of clause 27, wherein the coding method is an arithmetic coding method.

[00118] Clause 30. The method of clause 28, wherein the arithmetic coding method is context-based.

[00119] Clause 31. The method of any of clauses 27-30, wherein the piece of information is indicated at one of the following: a sequence header, a picture header, a slice header, or a block.

[00120] Clause 32. A method for point cloud coding, comprising: classifying, during a conversion between a current frame of a point cloud sequence and a bitstream of the point cloud sequence, points in the point cloud sequence based on a plurality types of process units for global motion estimation of the current frame; and performing the conversion based on the classification.

[00121] Clause 33. The method of clause 32, wherein the plurality types of process units comprises at least one of the following: a cubic block, or a planar region being three-dimensional and having a height equal to a height of a bounding box of the current frame.

[00122] Clause 34. The method of any of clauses 32-33, wherein the global motion estimation of the current frame is performed based on a plurality of reference frames, and classifying the points comprises: classifying points in the plurality of reference frames based on one of the plurality types of process units.

[00123] Clause 35. The method of any of clauses 32-33, wherein the global motion estimation of the current frame is performed based at least on a first reference frame and a second reference frame, the plurality types of process units comprise a first type of process unit and a second type of process unit, and classifying the points comprises: classifying points in the first reference frame based on the first type of process unit; and classifying points in the second reference frame based on the second type of process unit.

[00124] Clause 36. The method of clause 35, wherein the first type of process unit is a type of process unit minimizing an error metric between the current frame and the first reference frame with motion compensation.

[00125] Clause 37. The method of clause 35, wherein the error metric comprises one of the following: mean square error, mean absolute error, or root mean square error.

[00126] Clause 38. The method of any of clauses 1-37, wherein the conversion includes encoding the current frame into the bitstream.

[00127] Clause 39. The method of any of clauses 1-37, wherein the conversion includes decoding the current frame from the bitstream.

[00128] Clause 40. An apparatus for processing point cloud data comprising a processor and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to perform a method in accordance with any of Clauses 1-39.

[00129] Clause 41. A non-transitory computer-readable storage medium storing instructions that cause a processor to perform a method in accordance with any of Clauses 1-39.

[00130] Clause 42. A non-transitory computer-readable recording medium storing a bitstream of a point cloud sequence which is generated by a method performed by a point cloud processing apparatus, wherein the method comprises: obtaining a compensated reference frame of a current frame of the point cloud sequence by performing motion compensation on a reference frame of the current frame based on a set of rotation matrixes comprising at least one fixed rotation matrix or a plurality of rotation matrixes; and generating the bitstream based on the compensated reference frame.

[00131] Clause 43. A method for storing a bitstream of a point cloud sequence, comprising: obtaining a compensated reference frame of a current frame of the point cloud sequence by performing motion compensation on a reference frame of the current frame based on a set of rotation matrixes comprising at least one fixed rotation matrix or a plurality of rotation matrixes; generating the bitstream based on the compensated reference frame; and storing the bitstream in a non-transitory computer-readable recording medium.

[00132] Clause 44. A non-transitory computer-readable recording medium storing a bitstream of a point cloud sequence which is generated by a method performed by a point cloud

processing apparatus, wherein the method comprises: classifying points in the point cloud sequence based on a plurality types of process units for global motion estimation of a current frame of the point cloud sequence; and generating the bitstream based on the classification.

[00133] Clause 45. A method for storing a bitstream of a point cloud sequence, comprising: classifying points in the point cloud sequence based on a plurality types of process units for global motion estimation of a current frame of the point cloud sequence; generating the bitstream based on the classification; and storing the bitstream in a non-transitory computer-readable recording medium.

Example Device

[00134] Fig. 11 illustrates a block diagram of a computing device 1100 in which various embodiments of the present disclosure can be implemented. The computing device 1100 may be implemented as or included in the source device 110 (or the GPCC encoder 116 or 200) or the destination device 120 (or the GPCC decoder 126 or 300).

[00135] It would be appreciated that the computing device 1100 shown in Fig. 11 is merely for purpose of illustration, without suggesting any limitation to the functions and scopes of the embodiments of the present disclosure in any manner.

[00136] As shown in Fig. 11, the computing device 1100 includes a general-purpose computing device 1100. The computing device 1100 may at least comprise one or more processors or processing units 1110, a memory 1120, a storage unit 1130, one or more communication units 1140, one or more input devices 1150, and one or more output devices 1160.

[00137] In some embodiments, the computing device 1100 may be implemented as any user terminal or server terminal having the computing capability. The server terminal may be a server, a large-scale computing device or the like that is provided by a service provider. The user terminal may for example be any type of mobile terminal, fixed terminal, or portable terminal, including a mobile phone, station, unit, device, multimedia computer, multimedia tablet, Internet node, communicator, desktop computer, laptop computer, notebook computer, netbook computer, tablet computer, personal communication system (PCS) device, personal navigation device, personal digital assistant (PDA), audio/video player, digital camera/video camera, positioning device, television receiver, radio broadcast receiver, E-book device, gaming device, or any combination thereof, including the accessories and peripherals of these

devices, or any combination thereof. It would be contemplated that the computing device 1100 can support any type of interface to a user (such as "wearable" circuitry and the like).

[00138] The processing unit 1110 may be a physical or virtual processor and can implement various processes based on programs stored in the memory 1120. In a multi-processor system, multiple processing units execute computer executable instructions in parallel so as to improve the parallel processing capability of the computing device 1100. The processing unit 1110 may also be referred to as a central processing unit (CPU), a microprocessor, a controller or a microcontroller.

[00139] The computing device 1100 typically includes various computer storage medium. Such medium can be any medium accessible by the computing device 1100, including, but not limited to, volatile and non-volatile medium, or detachable and non-detachable medium. The memory 1120 can be a volatile memory (for example, a register, cache, Random Access Memory (RAM)), a non-volatile memory (such as a Read-Only Memory (ROM), Electrically Erasable Programmable Read-Only Memory (EEPROM), or a flash memory), or any combination thereof. The storage unit 1130 may be any detachable or non-detachable medium and may include a machine-readable medium such as a memory, flash memory drive, magnetic disk or another other media, which can be used for storing information and/or data and can be accessed in the computing device 1100.

[00140] The computing device 1100 may further include additional detachable/non-detachable, volatile/non-volatile memory medium. Although not shown in Fig. 11, it is possible to provide a magnetic disk drive for reading from and/or writing into a detachable and non-volatile magnetic disk and an optical disk drive for reading from and/or writing into a detachable non-volatile optical disk. In such cases, each drive may be connected to a bus (not shown) via one or more data medium interfaces.

[00141] The communication unit 1140 communicates with a further computing device via the communication medium. In addition, the functions of the components in the computing device 1100 can be implemented by a single computing cluster or multiple computing machines that can communicate via communication connections. Therefore, the computing device 1100 can operate in a networked environment using a logical connection with one or more other servers, networked personal computers (PCs) or further general network nodes.

[00142] The input device 1150 may be one or more of a variety of input devices, such as a mouse, keyboard, tracking ball, voice-input device, and the like. The output device 1160 may

be one or more of a variety of output devices, such as a display, loudspeaker, printer, and the like. By means of the communication unit 1140, the computing device 1100 can further communicate with one or more external devices (not shown) such as the storage devices and display device, with one or more devices enabling the user to interact with the computing device 1100, or any devices (such as a network card, a modem and the like) enabling the computing device 1100 to communicate with one or more other computing devices, if required. Such communication can be performed via input/output (I/O) interfaces (not shown).

[00143] In some embodiments, instead of being integrated in a single device, some or all components of the computing device 1100 may also be arranged in cloud computing architecture. In the cloud computing architecture, the components may be provided remotely and work together to implement the functionalities described in the present disclosure. In some embodiments, cloud computing provides computing, software, data access and storage service, which will not require end users to be aware of the physical locations or configurations of the systems or hardware providing these services. In various embodiments, the cloud computing provides the services via a wide area network (such as Internet) using suitable protocols. For example, a cloud computing provider provides applications over the wide area network, which can be accessed through a web browser or any other computing components. The software or components of the cloud computing architecture and corresponding data may be stored on a server at a remote position. The computing resources in the cloud computing environment may be merged or distributed at locations in a remote data center. Cloud computing infrastructures may provide the services through a shared data center, though they behave as a single access point for the users. Therefore, the cloud computing architectures may be used to provide the components and functionalities described herein from a service provider at a remote location. Alternatively, they may be provided from a conventional server or installed directly or otherwise on a client device.

[00144] The computing device 1100 may be used to implement point cloud encoding/decoding in embodiments of the present disclosure. The memory 1120 may include one or more point cloud coding modules 1125 having one or more program instructions. These modules are accessible and executable by the processing unit 1110 to perform the functionalities of the various embodiments described herein.

[00145] In the example embodiments of performing point cloud encoding, the input device 1150 may receive point cloud data as an input 1170 to be encoded. The point cloud data may

be processed, for example, by the point cloud coding module 1125, to generate an encoded bitstream. The encoded bitstream may be provided via the output device 1160 as an output 1180.

[00146] In the example embodiments of performing point cloud decoding, the input device 1150 may receive an encoded bitstream as the input 1170. The encoded bitstream may be processed, for example, by the point cloud coding module 1125, to generate decoded point cloud data. The decoded point cloud data may be provided via the output device 1160 as the output 1180.

[00147] While this disclosure has been particularly shown and described with references to preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the spirit and scope of the present application as defined by the appended claims. Such variations are intended to be covered by the scope of this present application. As such, the foregoing description of embodiments of the present application is not intended to be limiting.

I/We Claim:

1. A method for point cloud coding, comprising:

obtaining, during a conversion between a current frame of a point cloud sequence and a bitstream of the point cloud sequence, a compensated reference frame of the current frame by performing motion compensation on a reference frame of the current frame based on a set of rotation matrixes comprising at least one fixed rotation matrix or a plurality of rotation matrixes; and

performing the convention based on the compensated reference frame.

2. The method of claim 1, wherein at least one of the set of rotation matrixes is predefined, or

at least one of the set of rotation matrixes is indicated in the bitstream, or

at least one of the set of rotation matrixes is determined on-the-fly.

3. The method of any of claims 1-2, wherein one of the set of rotation matrixes is used for performing motion compensation on a further reference frame of the current frame.

4. The method of any of claims 1-3, wherein one of the set of rotation matrixes is used for performing motion compensation on all of reference frames of the current frame.

5. The method of any of claims 1-4, wherein one of the set of rotation matrixes is fixed as an identity matrix.

6. The method of any of claims 1-5, wherein at least one syntax element is indicated in the bitstream, the at least one syntax element indicating whether one of the set of rotation matrixes is used for performing motion compensation on a further reference frame of the current frame.

7. The method of any of claims 1-5, wherein at least one syntax element is indicated in the bitstream, the at least one syntax element indicating whether one of the set of rotation matrixes is used for performing motion compensation on all of reference frames of the current frame.

8. The method of any of claims 1-5, wherein at least one syntax element is indicated in the bitstream, the at least one syntax element indicating whether a fixed rotation matrix is used for the reference frame.

9. The method of any of claims 1-8, wherein at least one of the set of rotation matrixes is determined based on a predictive coding method.

10. The method of any of claims 1-9, wherein obtaining the compensated reference frame comprises:

determining a global motion matrix for the current frame based on the set of rotation matrixes and a set of translation vectors for the reference frame; and

performing the motion compensation on the reference frame based on the global motion matrix to obtain the compensated reference frame.

11. The method of claim 10, wherein a translation vector in the set of translation vectors is determined based on samples of the current frame and reference samples of the reference frame.

12. The method of claim 11, wherein the translation vector is a vector minimizing an error metric between the samples and the reference samples with motion compensation.

13. The method of claim 12, wherein the error metric comprises one of the following:

mean square error,

mean absolute error, or

root mean square error.

14. The method of any of claims 10-13, wherein the global motion matrix is generated at an encoder.

15. The method of any of claims 10-13, wherein the global motion matrix is generated at a decoder.

16. The method of any of claims 10-15, wherein the set of translation vectors comprises a plurality of translation vectors.

17. The method of any of claims 10-16, wherein at least one indication for the set of translation vectors is indicated in the bitstream.

18. The method of claim 17, wherein the at least one indication indicates a translation vector used for a target unit of the reference frame.

19. The method of claim 18, wherein the target unit comprises one of the following:

a sequence,

a frame,

a slice,

a tile, or

a sub-region in the reference frame.

20. The method of any of claims 10-19, wherein the set of translation vectors are associated with a target unit of the reference frame, and the set of translation vectors are indicated in the bitstream.

21. The method of claim 20, wherein the target unit comprises one of the following:

a sequence,

a frame,

a slice,

a tile, or

a sub-region in the reference frame.

22. The method of any of claims 10-19, wherein at least one of the set of translation vectors is determined based on a predictive coding method.

23. The method of claim 22, wherein a first element of a first translation vector of the set of translation vectors is determined based on a second element of the first translation vector.

24. The method of claim 22, wherein a first element of a first translation vector of the set of translation vectors is determined based on a second element of a second translation vector associated with a second reference frame of the current frame.

25. The method of claim 24, wherein the second reference frame is the same as or different from the reference frame.

26. The method of any of claims 1-9, wherein the motion compensation is performed without global motion compensation.

27. The method of any of claims 1-26, wherein a piece of information for the convention is binarized based on a coding method, the piece of information being indicated in the bitstream.

28. The method of claim 27, wherein the coding method is one of the following:

a fixed length coding method,

an exponential golomb coding method,

a unary coding method,

a truncated unary coding method,

a signed unary coding method, or

a signed truncated unary coding method.

29. The method of claim 27, wherein the coding method is an arithmetic coding method.

30. The method of claim 28, wherein the arithmetic coding method is context-based.

31. The method of any of claims 27-30, wherein the piece of information is indicated at one of the following:

a sequence header,

a picture header,

a slice header, or

a block.

32. A method for point cloud coding, comprising:

classifying, during a conversion between a current frame of a point cloud sequence and a bitstream of the point cloud sequence, points in the point cloud sequence based on a plurality types of process units for global motion estimation of the current frame; and

performing the conversion based on the classification.

33. The method of claim 32, wherein the plurality types of process units comprises at least one of the following:

a cubic block, or

a planar region being three-dimensional and having a height equal to a height of a bounding box of the current frame.

34. The method of any of claims 32-33, wherein the global motion estimation of the current frame is performed based on a plurality of reference frames, and classifying the points comprises:

classifying points in the plurality of reference frames based on one of the plurality types of process units.

35. The method of any of claims 32-33, wherein the global motion estimation of the current frame is performed based at least on a first reference frame and a second reference frame, the plurality types of process units comprise a first type of process unit and a second type of process unit, and classifying the points comprises:

classifying points in the first reference frame based on the first type of process unit; and

classifying points in the second reference frame based on the second type of process unit.

36. The method of claim 35, wherein the first type of process unit is a type of process unit minimizing an error metric between the current frame and the first reference frame with motion compensation.

37. The method of claim 35, wherein the error metric comprises one of the following:

mean square error,

mean absolute error, or

root mean square error.

38. The method of any of claims 1-37, wherein the conversion includes encoding the current frame into the bitstream.

39. The method of any of claims 1-37, wherein the conversion includes decoding the current frame from the bitstream.

40. An apparatus for processing point cloud data comprising a processor and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to perform a method in accordance with any of Claims 1-39.

41. A non-transitory computer-readable storage medium storing instructions that cause a processor to perform a method in accordance with any of Claims 1-39.

42. A non-transitory computer-readable recording medium storing a bitstream of a point cloud sequence which is generated by a method performed by a point cloud processing apparatus, wherein the method comprises:

obtaining a compensated reference frame of a current frame of the point cloud sequence by performing motion compensation on a reference frame of the current frame based on a set of rotation matrixes comprising at least one fixed rotation matrix or a plurality of rotation matrixes; and

generating the bitstream based on the compensated reference frame.

43. A method for storing a bitstream of a point cloud sequence, comprising:

obtaining a compensated reference frame of a current frame of the point cloud sequence by performing motion compensation on a reference frame of the current frame based on a set of rotation matrixes comprising at least one fixed rotation matrix or a plurality of rotation matrixes;

generating the bitstream based on the compensated reference frame; and

storing the bitstream in a non-transitory computer-readable recording medium.

44. A non-transitory computer-readable recording medium storing a bitstream of a point cloud sequence which is generated by a method performed by a point cloud processing apparatus, wherein the method comprises:

classifying points in the point cloud sequence based on a plurality types of process units for global motion estimation of a current frame of the point cloud sequence; and

generating the bitstream based on the classification.

45. A method for storing a bitstream of a point cloud sequence, comprising:

classifying points in the point cloud sequence based on a plurality types of process units for global motion estimation of a current frame of the point cloud sequence;

generating the bitstream based on the classification; and

storing the bitstream in a non-transitory computer-readable recording medium.

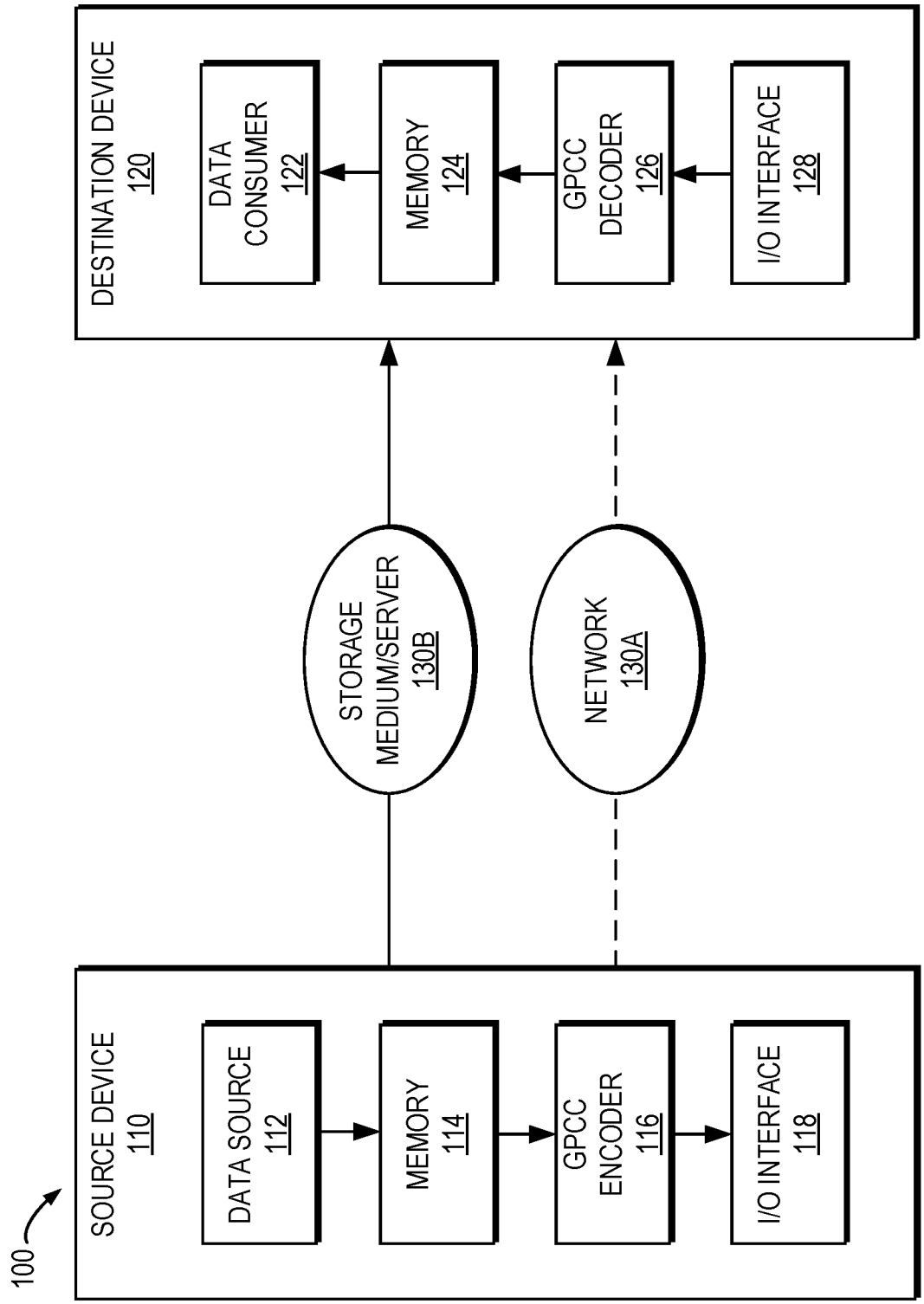

100
SOURCE DEVICE
110
DATA SOURCE
112
MEMORY
114
GPCC
ENCODER
116
I/O INTERFACE
118
STORAGE
MEDIUM/SERVER
130B
NETWORK
130A
DESTINATION DEVICE
120
DATA
CONSUMER
122
MEMORY
124
GPCC
DECODER
126
I/O INTERFACE
128

Fig. 1

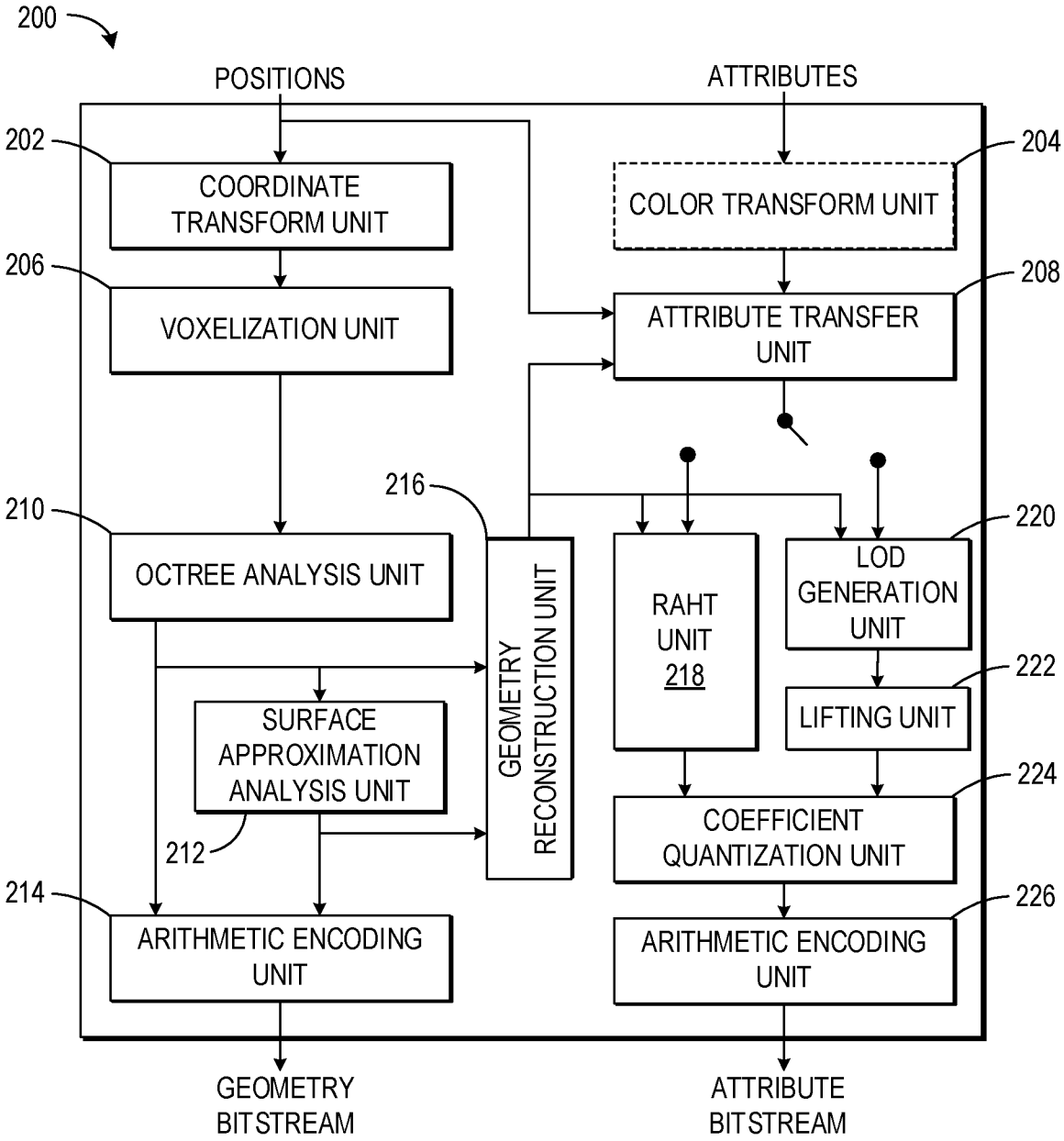

200
POSITIONS
ATTRIBUTES
202
COORDINATE TRANSFORM UNIT
204
COLOR TRANSFORM UNIT
206
VOXELIZATION UNIT
208
ATTRIBUTE TRANSFER UNIT
216
210
OCTREE ANALYSIS UNIT
GEOMETRY RECONSTRUCTION UNIT
RAHT UNIT 218
220
LOD GENERATION UNIT
222
LIFTING UNIT
SURFACE APPROXIMATION ANALYSIS UNIT
212
224
COEFFICIENT QUANTIZATION UNIT
214
ARITHMETIC ENCODING UNIT
226
ARITHMETIC ENCODING UNIT
GEOMETRY BITSTREAM
ATTRIBUTE BITSTREAM

Fig. 2

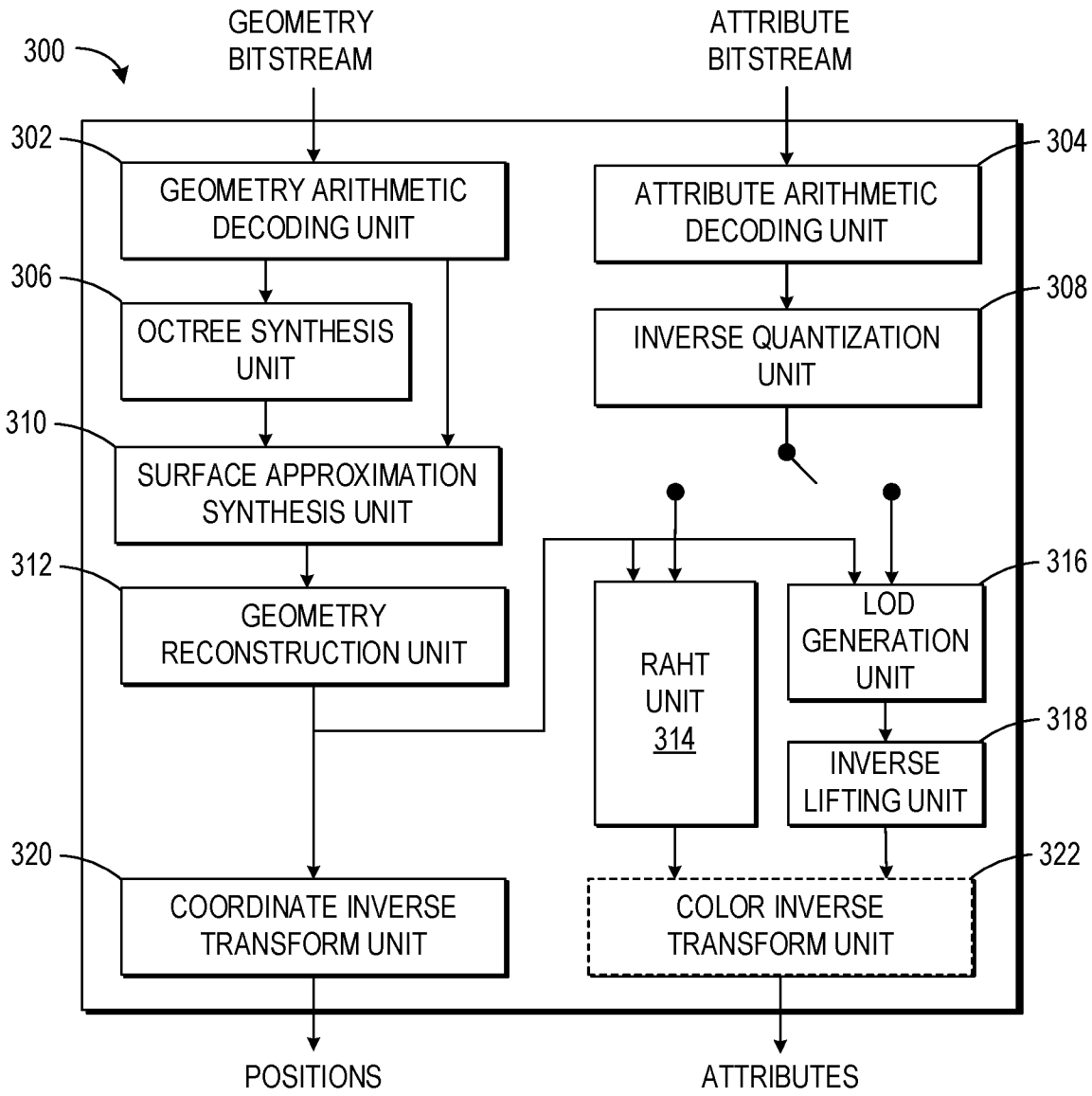

300
GEOMETRY BITSTREAM
ATTRIBUTE BITSTREAM
302
GEOMETRY ARITHMETIC DECODING UNIT
304
ATTRIBUTE ARITHMETIC DECODING UNIT
306
OCTREE SYNTHESIS UNIT
308
INVERSE QUANTIZATION UNIT
310
SURFACE APPROXIMATION SYNTHESIS UNIT
312
GEOMETRY RECONSTRUCTION UNIT
RAHT UNIT 314
316
LOD GENERATION UNIT
318
INVERSE LIFTING UNIT
320
COORDINATE INVERSE TRANSFORM UNIT
322
COLOR INVERSE TRANSFORM UNIT
POSITIONS
ATTRIBUTES

Fig. 3

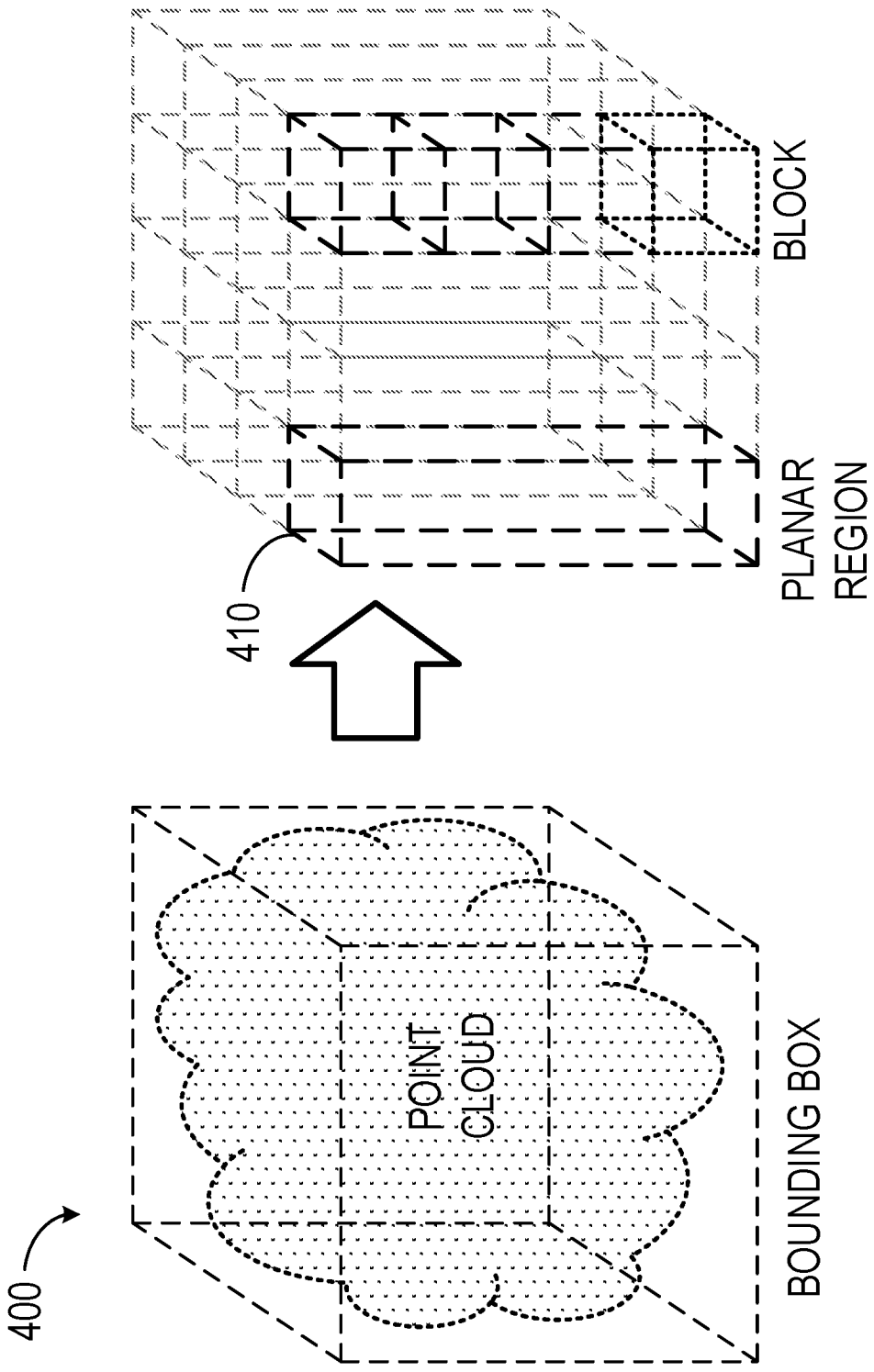


Fig. 4

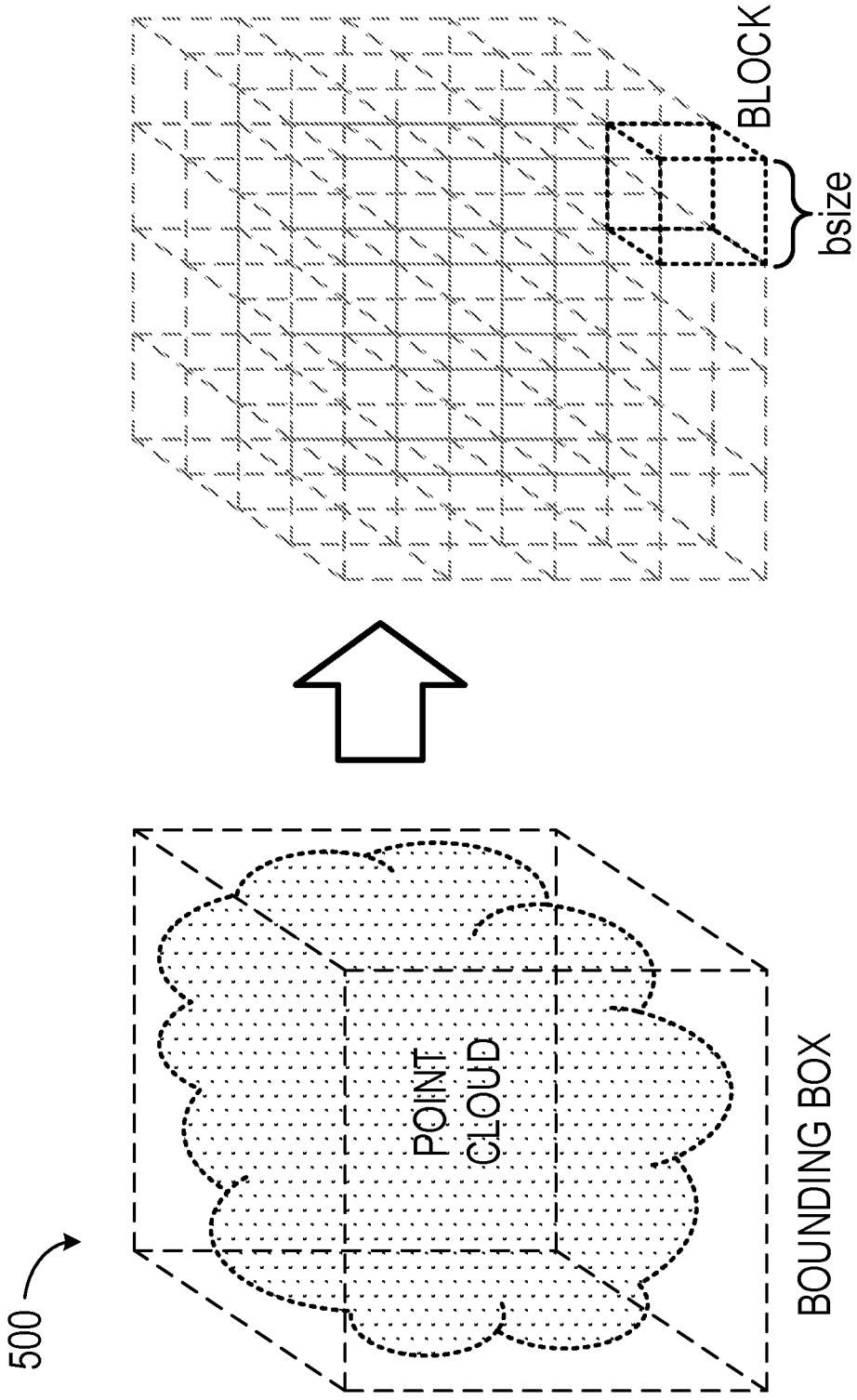


Fig. 5

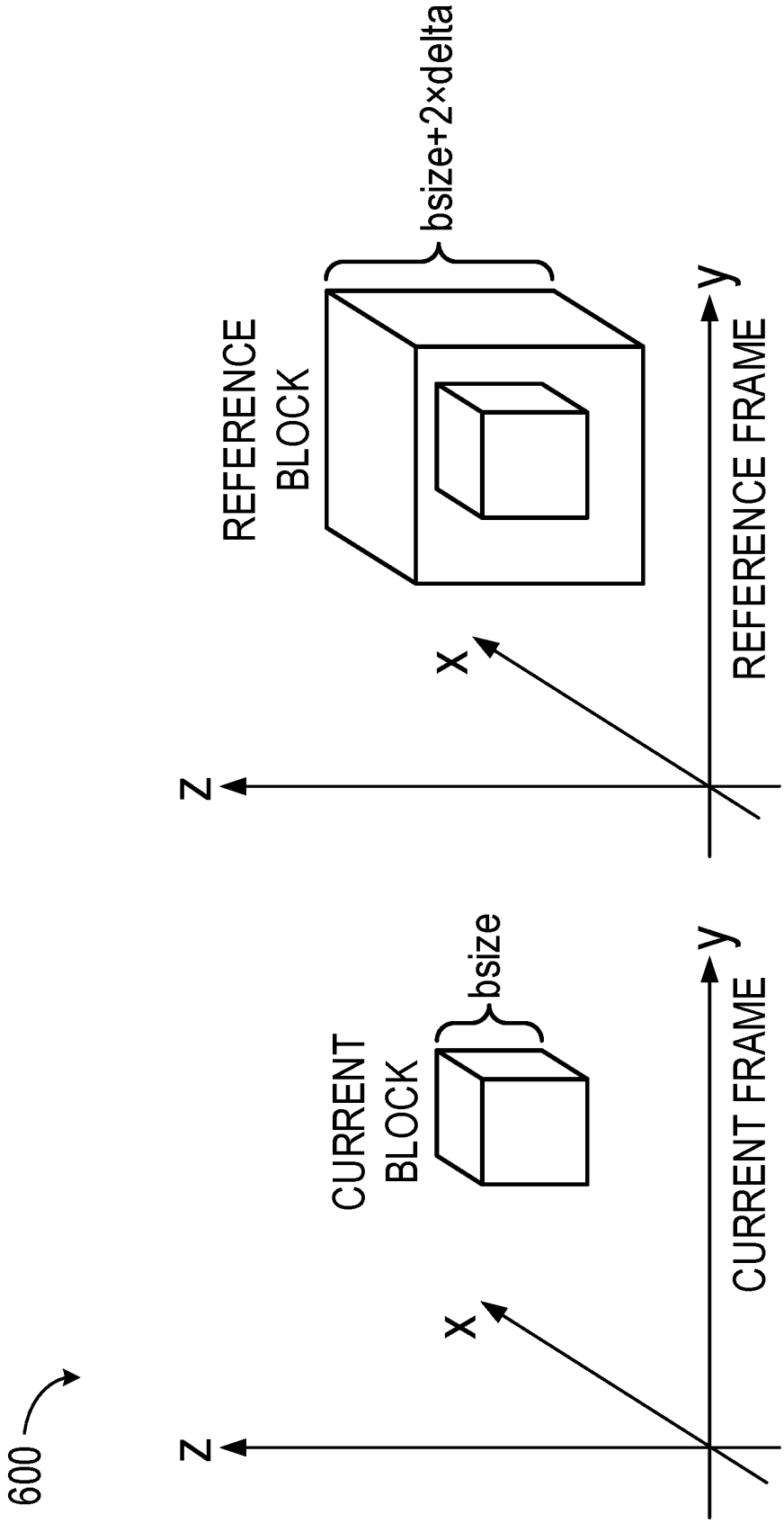

600
z
x
y
CURRENT FRAME
CURRENT BLOCK
bsize
z
x
y
REFERENCE FRAME
REFERENCE BLOCK
bsize+2×delta

Fig. 6

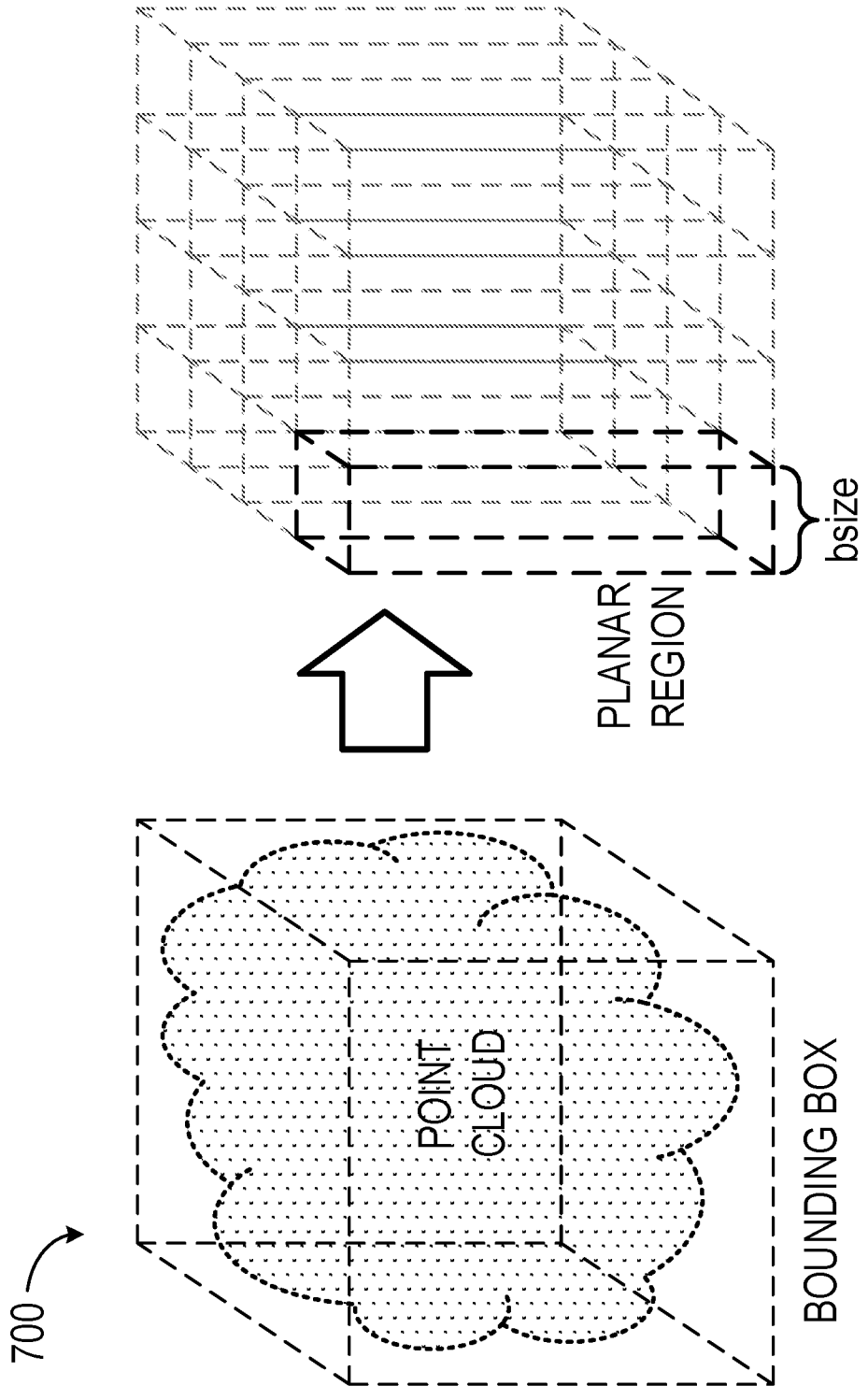


Fig. 7

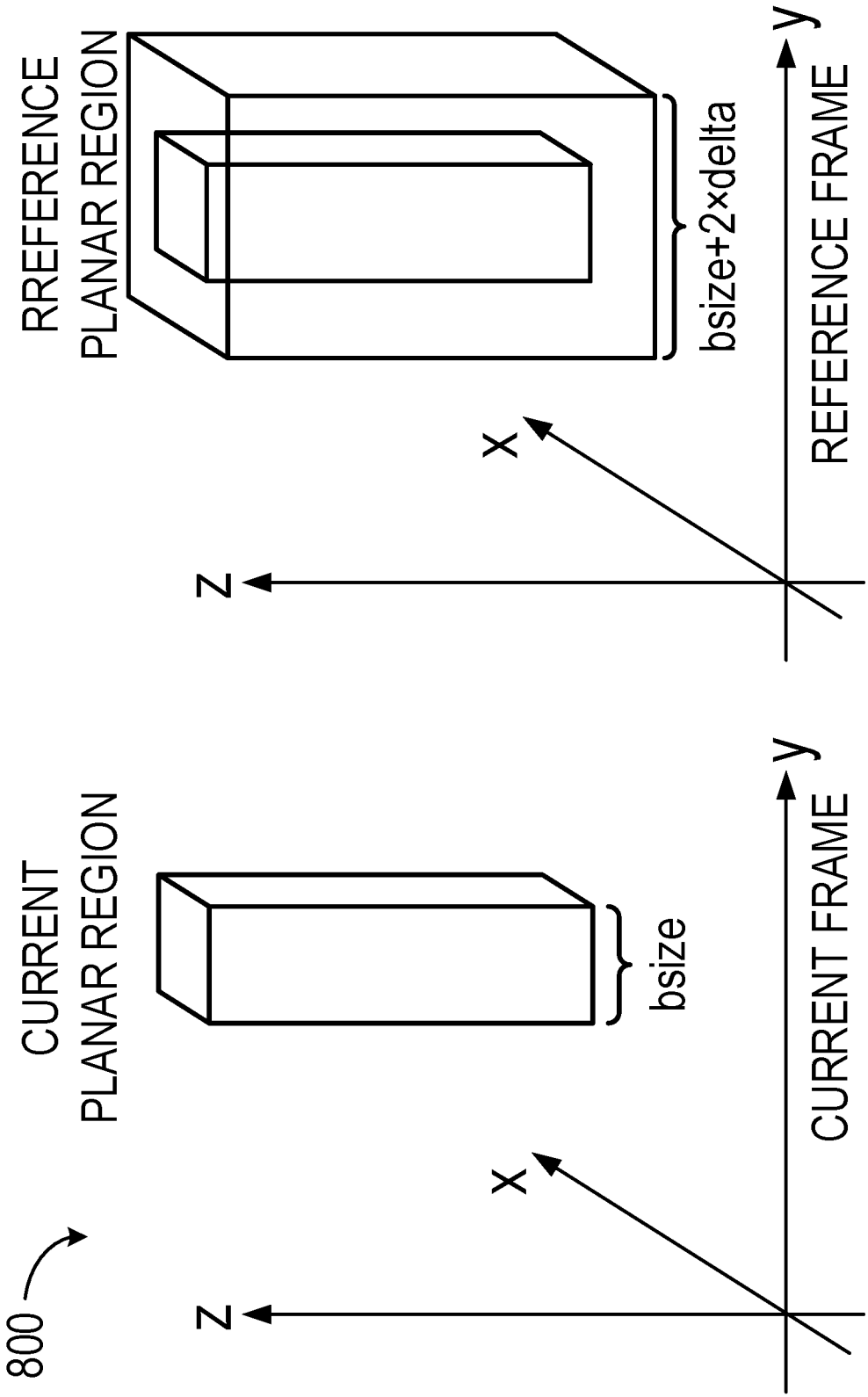

800
CURRENT PLANAR REGION
bsize
z
x
y
CURRENT FRAME
RREFERENCE PLANAR REGION
bsize+2×delta
z
x
y
REFERENCE FRAME

Fig. 8

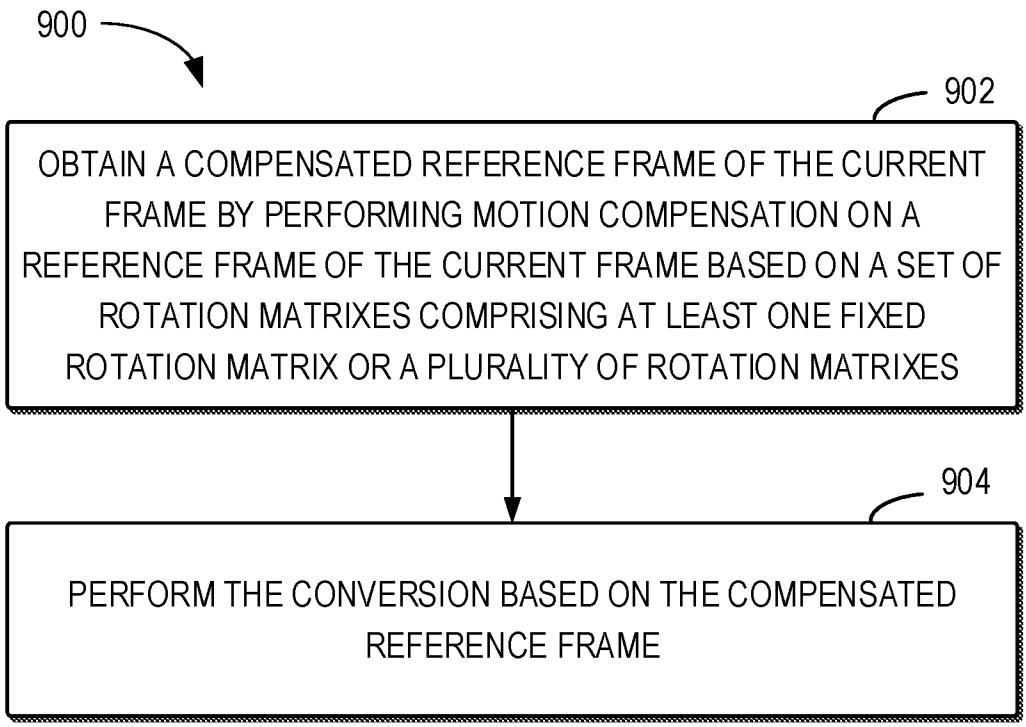

900
902
OBTAIN A COMPENSATED REFERENCE FRAME OF THE CURRENT FRAME BY PERFORMING MOTION COMPENSATION ON A REFERENCE FRAME OF THE CURRENT FRAME BASED ON A SET OF ROTATION MATRIXES COMPRISING AT LEAST ONE FIXED ROTATION MATRIX OR A PLURALITY OF ROTATION MATRIXES
904
PERFORM THE CONVERSION BASED ON THE COMPENSATED REFERENCE FRAME

Fig. 9

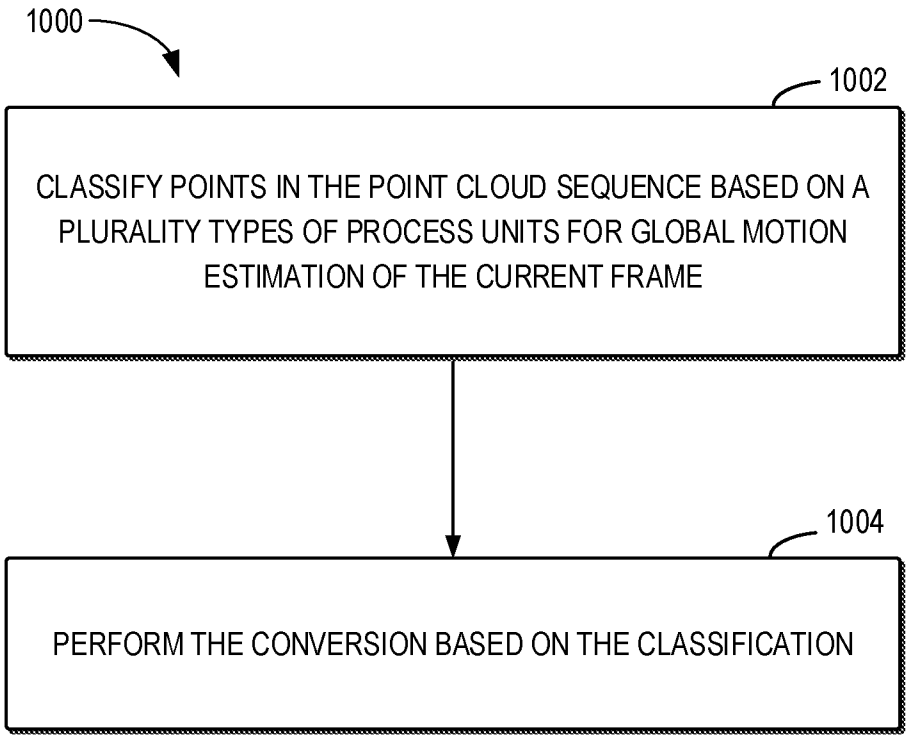

1000
1002
CLASSIFY POINTS IN THE POINT CLOUD SEQUENCE BASED ON A PLURALITY TYPES OF PROCESS UNITS FOR GLOBAL MOTION ESTIMATION OF THE CURRENT FRAME
1004
PERFORM THE CONVERSION BASED ON THE CLASSIFICATION

Fig. 10

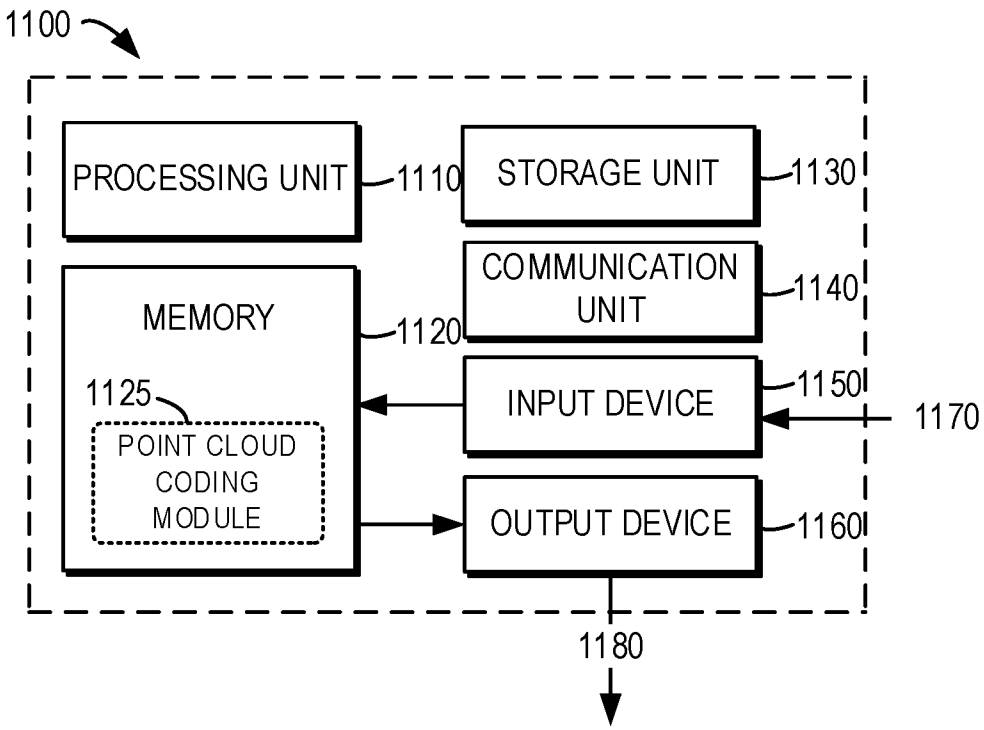

1100
PROCESSING UNIT
1110
STORAGE UNIT
1130
COMMUNICATION UNIT
1140
MEMORY
1120
1125
POINT CLOUD CODING MODULE
INPUT DEVICE
1150
1170
OUTPUT DEVICE
1160
1180

Fig. 11

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2022/103834

A. CLASSIFICATION OF SUBJECT MATTER

G06T 7/246(2017.01)i; G06T 9/00(2006.01)i; H04N 19/119(2014.01)i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06T; H04N

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

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

CNPAT,CNKI,WPI,EPODOC,HEVC,VVC: point cloud, frame, compensate, reference frame, motion compensation, rotation matrix, fixed, classify, global motion estimation,

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2020021847 A1 (APPLE INC.) 16 January 2020 (2020-01-16) claims 1, 12, description, paragraphs [0004]-[0016]	1-45
A	US 2020043121 A1 (INTEL CORPORATION) 06 February 2020 (2020-02-06) the whole document	1-45
A	EP 3301930 A1 (THOMSON LICENSING) 04 April 2018 (2018-04-04) the whole document	1-45
A	CN 112640417 A (HUAWEI TECHNOLOGIES CO., LTD.) 09 April 2021 (2021-04-09) the whole document	1-45
A	CN 104715496 A (UNIVERSITY OF SCIENCE AND TECHNOLOGY OF CHINA) 17 June 2015 (2015-06-17) the whole document	1-45
A	CN 105143915 A (ISIS INNOVATION LTD.) 09 December 2015 (2015-12-09) the whole document	1-45

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

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

Date of the actual completion of the international search	Date of mailing of the international search report
09 September 2022	**23 September 2022**
Name and mailing address of the ISA/CN **National Intellectual Property Administration, PRC** **6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing** **100088, China** Facsimile No. **(86-10)62019451**	Authorized officer **YU,Lina** Telephone No. **(86-10)53961756**

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2022/103834

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	ZAKHARCHENKO.V et al. "Encoder-only Supplemental Motion Vector Estimation for Point cloud Coding content" *Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 36th Meeting: Gothenburg , SE Document: JCTVC-AJ0028-v2,* 12 July 2019 (2019-07-12), the whole document	1-45

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2022/103834

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
US	2020021847	A1	16 January 2020	US	2021266597	A1	26 August 2021
US	2020043121	A1	06 February 2020	CN	110855971	A	28 February 2020
				DE	102019117469	A1	06 February 2020
EP	3301930	A1	04 April 2018	KR	20190055101	A	22 May 2019
				WO	2018060207	A1	05 April 2018
				US	2019238853	A1	01 August 2019
				CN	109792536	A	21 May 2019
				JP	2019534616	A	28 November 2019
				EP	3520415	A1	07 August 2019
CN	112640417	A	09 April 2021		None		
CN	104715496	A	17 June 2015		None		
CN	105143915	A	09 December 2015	EP	2959315	A2	30 December 2015
				HK	1212450	A1	10 June 2016
				KR	20150119337	A	23 October 2015
				WO	2014128498	A2	28 August 2014
				JP	2016516977	A	09 June 2016
				GB	201303076	D0	10 April 2013
				US	2015379766	A1	31 December 2015