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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
23 January 2025 (23.01.2025)

WIPO | PCT

(10) International Publication Number
WO 2025/015486 A1

(51) International Patent Classification:
H04N 19/184 (2014.01) *H04N 19/96* (2014.01)
H04N 19/597 (2014.01)

(21) International Application Number:
PCT/CN2023/107611

(22) International Filing Date:
15 July 2023 (15.07.2023)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **BEIJING XIAOMI MOBILE SOFTWARE CO., LTD.** [CN/CN]; No. 018, Floor 8, Building 6, Yard 33, Middle Xierqi Road, Haidian District, Beijing 100085 (CN).

(72) Inventor: **GAO, Shuo**; No. 018, Floor 8, Building 6, Yard 33, Middle Xierqi Road, Haidian District, Beijing 100085 (CN).

(74) Agent: **FASHENG INTELLECTUAL PROPERTY COMPANY, LTD.**; Room 401, Floor 4, Unit 1, No. 99 Yuexiu Road, Haidian District, Beijing 100085 (CN).

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

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

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

(54) **Title:** METHOD AND APPARATUS FOR PROCESSING A POINT CLOUD, ENCODER, DECODER, AND STORAGE MEDIUM

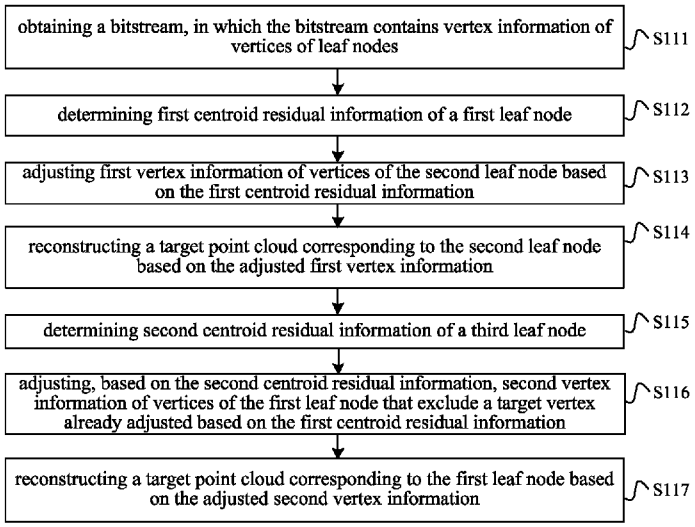


Figure 11

(57) **Abstract:** Method and apparatus for processing a point cloud are provided. The method includes: obtaining a bitstream, in which the bitstream contains vertex information of vertices of leaf nodes; determining first centroid residual information of a first leaf node; adjusting first vertex information of vertices of the second leaf node based on the first centroid residual information, in which the first leaf node and the second leaf node are adjacent nodes in a raster scan order; and reconstructing a target point cloud corresponding to the second leaf node based on the adjusted first vertex information.

WO 2025/015486 A1

Description

Title of Invention: METHOD AND APPARATUS FOR PROCESSING A POINT CLOUD, ENCODER, DECODER, AND STORAGE MEDIUM

Technical Field

[0001] The present disclosure relates to a field of communication technology. In particular, the present disclosure relates to a method and apparatus for processing a point cloud, an encoder, a decoder and a readable storage medium.

Background

[0002] As a format for the representation of 3D data, point clouds have recently gained traction as they are versatile in their capability in representing all types of 3D objects or scenes. Therefore, many use cases can be addressed by point clouds, among which are

[0003] · movie post-production,

[0004] · real-time 3D immersive telepresence or VR/AR applications,

[0005] · free viewpoint video (for instance for sports viewing) ,

[0006] · Geographical Information Systems (aka cartography) ,

[0007] · culture heritage (storage of scans of rare objects into a digital form) ,

[0008] · Autonomous driving, including 3D mapping of the environment and real-time Lidar data acquisition

SUMMARY

[0009] Embodiments of the present disclosure provide a method for processing a point cloud, an apparatus for processing a point cloud, an encoder, a decoder and a readable storage medium, which may decode a bitstream encoded from the point cloud more accurately and improve performance of reconstructing the point cloud.

[0010] In a first aspect a method for processing a point cloud is provided. The method includes: obtaining a bitstream, in which the bitstream contains vertex information of vertices of leaf nodes; determining first centroid residual information of a first leaf node; adjusting first vertex information of vertices of the second leaf node based on the first centroid residual information, in which the first leaf node and the second leaf node are adjacent nodes in a raster scan order; reconstructing a target point cloud corresponding to the second leaf node based on the adjusted first vertex information.

[0011] In the technical solution, the vertex information of the vertices of the current leaf node can be adjusted based on centroid residual information of a leaf node subsequently adjacent to the current leaf node, so that the bitstream encoded from the

point cloud can be decoded more accurately and performance of reconstructing the point cloud can be improved.

[0012] In a second aspect another method for processing a point cloud is provided. The method includes: obtaining vertex information of vertices of leaf nodes; encoding the vertex information into a bitstream; reconstructing a target point cloud by obtaining the vertex information from the bitstream, in which reconstructing the target point cloud includes: determining first centroid residual information of a first leaf node; adjusting first vertex information of vertices of the second leaf node based on the first centroid residual information, in which the first leaf node and the second leaf node are adjacent nodes in a raster scan order; reconstructing a target point cloud corresponding to the second leaf node based on the adjusted first vertex information.

[0013] In a third aspect an apparatus for processing a point cloud is provided. The apparatus includes: a data obtaining unit, configured to obtain a bitstream, in which the bitstream contains vertex information of vertices of leaf nodes; a determining unit, configured to, determine first centroid residual information of a first leaf node; an adjusting unit, configured to adjust first vertex information of vertices of the second leaf node based on the first centroid residual information, in which the first leaf node and the second leaf node are adjacent nodes in a raster scan order; and a reconstructing unit, configured to reconstruct a target point cloud corresponding to the first leaf node based on the adjusted first vertex information.

[0014] In a fourth aspect another apparatus for processing a point cloud is provided. The apparatus includes: an information obtaining unit, configured to obtain vertex information of vertices of leaf node; an encoding unit, configured to encode the vertex information into a bitstream; a point cloud reconstructing unit, configured to reconstruct a target point cloud by obtaining the vertex information from the bitstream, in which reconstructing the target point cloud includes: determining first centroid residual information of a first leaf node; adjusting first vertex information of vertices of the second leaf node based on the first centroid residual information, in which the first leaf node and the second leaf node are adjacent nodes in a raster scan order; reconstructing a target point cloud corresponding to the second leaf node based on the adjusted first vertex information.

[0015] In a fifth aspect of the present disclosure an encoder is provided. The encoder includes a memory and at least one processor, in which instructions are stored in the memory, which when executed by the processor perform the steps of the method according to the first aspect.

[0016] In a sixth aspect of the present disclosure a decoder is provided. The decoder comprises a memory and at least one processor, in which instructions are stored in

the memory, which when executed by the processor perform the steps of the method according to the second aspect.

[0017] In a seventh aspect of the present disclosure a bitstream is provided, wherein the bitstream is encoded by the steps of the method according to the first aspect.

[0018] In an eighth aspect of the present disclosure a computer-readable storage medium is provided including instructions to perform the steps of the method according to the first or second aspect.

[0019] In a ninth aspect of the present disclosure a computer program product including a computer program is provided. When the computer program is running on a computer, the computer is caused to perform the steps of the method according to the first or second aspect.

[0020] In a tenth aspect of the present disclosure a computer program is provided. When the computer program is running on a computer, the computer is caused to perform the steps of the method according to the first or second aspect.

[0021] FIGURES

[0022] In the following the present disclosure is described in more detail with reference to the accompanying figures.

[0023] The figures show:

[0024] Figure 1 an example of vertices on the edges of a cuboid,

[0025] Figure 2 a generation of the triangle by vertices,

[0026] Figure 3 an example of determining the order of the triangles according to Figure 7,

[0027] Figure 4 a generation of the triangle by vertices,

[0028] Figure 5 a schematic drawing for the step of voxelization,

[0029] Figure 6 an example of reconstruction of triangles using the centroid point C as a pivoting point,

[0030] Figure 7 an example of a normal vector,

[0031] Figure 8 an example of 1D residual along the normal vector,

[0032] Figure 9 an example of a refined surface in a leaf node with modelling triangles constructed by vertices and centroid point,

[0033] Figure 10 an example of vertex information adjustment,

[0034] Figure 11 a flowchart of a method for processing a point cloud,

[0035] Figure 12 a flowchart of another method for processing a point cloud,

[0036] Figure 13 a block diagram of an apparatus for processing a point cloud,

[0037] Figure 14 a block diagram of another apparatus for processing a point cloud,

[0038] Figure 15 a decoder or encoder.

DETAILED DESCRIPTION

[0039] Reference will now be made in detail to exemplary embodiments, examples of which are illustrated in the accompanying drawings. The following description refers to the accompanying drawings in which the same numbers in different drawings represent the same or similar elements unless otherwise represented. The implementations set forth in the following description of exemplary embodiments do not represent all implementations consistent with the disclosure. Instead, they are merely examples of apparatuses and methods consistent with aspects related to the disclosure as recited in the appended claims.

[0040] The terms used in the disclosure are only for the purpose of describing specific embodiments, and are not intended to limit the disclosure. The singular forms of "a" and "the" used in the disclosure and appended claims are also intended to include plural forms, unless the context clearly indicates other meanings. It should also be understood that the term "and/or" as used herein refers to and includes any or all possible combinations of one or more associated listed items.

[0041] It should be understood that although the terms "first" , "second" , and "third" may be used in this disclosure to describe various information, the information should not be limited to these terms. These terms are only used to distinguish the same type of information from each other. For example, without departing from the scope of the disclosure, the first information may also be referred to as the second information, and similarly, the second information may also be referred to as the first information. Depending on the context, the term "if" as used herein can be interpreted as "when" , "while" or "in response to determining" .

[0042] A point cloud is a set of points located in a 3D space, optionally with additional values attached to each of the points. These additional values are usually called point attributes. Consequently, a point cloud is a combination of a geometry (the 3D position of each point) and attributes.

[0043] Attributes may be, for example, three-component colours, material properties like reflectance and/or two-component normal vectors to a surface associated with the point.

[0044] Point clouds may be captured by various types of devices like an array of cameras, depth sensors, Lidars, scanners, or may be computer-generated (in movie post-production for example) . Depending on the use cases, points clouds may have from thousands to up to billions of points for cartography applications.

[0045] Raw representations of point clouds require a very high number of bits per point, with at least a dozen of bits per spatial component X, Y or Z, and optionally more bits for the attribute (s) , for instance three times 10 bits for the colours. Practical deployment of point-cloud-based applications requires compression technologies

that enable the storage and distribution of point clouds with reasonable storage and transmission infrastructures.

[0046] Compression may be lossy (like in video compression) for the distribution to and visualization by an end-user, for example on AR/VR glasses or any other 3D-capable device. Other use cases do require lossless compression, like medical applications or autonomous driving, to avoid altering the results of a decision obtained from the analysis of the compressed and transmitted point cloud.

[0047] Until recently, point cloud compression (aka PCC) was not addressed by the mass market and no standardized point cloud codec was available. In 2017, the standardization working group ISO/JCT1/SC29/WG11, also known as Moving Picture Experts Group or MPEG, has initiated work items on point cloud compression. This has led to two standards, namely

[0048] · MPEG-I part 5 (ISO/IEC 23090-5) or Video-based Point Cloud Compression (V-PCC)

[0049] · MPEG-I part 9 (ISO/IEC 23090-9) or Geometry-based Point Cloud Compression (G-PCC)

[0050] Both V-PCC and G-PCC standards have finalized their first version in late 2020 and will soon be available to the market.

[0051] The V-PCC coding method compresses a point cloud by performing multiple projections of a 3D object to obtain 2D patches that are packed into an image (or a video when dealing with moving point clouds) . Obtained images or videos are then compressed using already existing image/video codecs, allowing for the leverage of already deployed image and video solutions. By its very nature, V-PCC is efficient only on dense and continuous point clouds because image/video codecs are unable to compress non-smooth patches as would be obtained from the projection of, for example, Lidar-acquired sparse geometry data.

[0052] The G-PCC coding method has two schemes for the compression of the geometry.

[0053] The first scheme is based on an occupancy tree (octree/quadtree/binary tree) representation of the point cloud geometry. Occupied nodes are split down until a certain size is reached, and occupied leaf nodes provide the location of points, typically at the center of these nodes. By using neighbour-based prediction techniques, high level of compression can be obtained for dense point clouds. Sparse point clouds are also addressed by directly coding the position of point within a node with non-minimal size, by stopping the tree construction when only isolated points are present in a node; this technique is known as Direct Coding Mode (DCM) .

[0054] The second scheme is based on a predictive tree, each node representing the 3D location of one point and the relation between nodes is spatial prediction from parent to children. This method can only address sparse point clouds and offers the advantage

of lower latency and simpler decoding than the occupancy tree. However, compression performance is only marginally better, and the encoding is complex, relatively to the first occupancy-based method, intensively looking for the best predictor (among a long list of potential predictors) when constructing the predictive tree.

[0055] In both schemes, attribute (de) coding is performed after complete geometry (de) coding, leading to a two-pass coding. Thus, low latency is obtained by using slices that decompose the 3D space into sub-volumes that are coded independently, without prediction between the sub-volumes. This may heavily impact the compression performance when many slices are used.

[0056] An important use case is the transmission of dynamic AR/VR point clouds. Dynamic means that the point cloud evolves with respect to time. Also, AR/VR point clouds are typically locally 2D as they most of time represent the surface of an object. As such, AR/VR point clouds are highly connected (or said to be dense) in the sense that a point is rarely isolated and, instead, has many neighbours.

[0057] Dense (or solid) point clouds represent continuous surfaces with a resolution such that volumes (small cubes called voxels) associated with points touch each other without exhibiting any visual hole in the surface.

[0058] Such point clouds are typically used in AR/VR environments and are viewed by the end user through a device like a TV, a smartphone or a headset. They are transmitted to the device or stored locally. Many AR/VR applications use moving point clouds, as opposed to static point clouds, that vary with time. Therefore, the volume of data is huge and must be compressed. Nowadays, lossless compression based on an octree representation of the geometry of the point cloud can achieve down to slightly less than a bit per point (1 bpp) . This may not be sufficient for real time transmission that may involve several millions of points per frame with a frame rate as high as 50 frames per second (fps) , thus leading to hundreds of megabits of data per second.

[0059] Consequently, lossy compression may be used with the usual requirement of maintaining an acceptable visual quality while compressing sufficiently to fit within the bandwidth provided by the transmission channel while maintaining real time transmission of the frames. In many applications, bitrates as low as 0.1 bpp (10x more compressed than lossless coding) would already make possible real time transmission.

[0060] The codec VPCC based on MPEG-I part 5 (ISO/IEC 23090-5) or Video-based Point Cloud Compression (V-PCC) can achieve such low bitrates by using lossy compression of video codecs that compress 2D frames obtained from the projection of the point cloud on a plane. The geometry is represented by a series of projection patches assembled into a frame, each patch being a small local depth map. However, VPCC is not versatile and is limited to a narrow type of point clouds that do not exhibit

locally complex geometry (like trees, hair) because the obtained projected depth map would not be smooth enough to be efficiently compressed by a video codec.

[0061] Purely 3D compression techniques can handle any type of point clouds. It is still an open question whether 3D compression techniques can compete with VPCC (or any projection + image coding scheme) on dense point clouds. Standardization is still under its way toward offering an extension (an amendment) of GPCC that would provide competitive lossy compression that would compress dense point clouds as good as VPCC intra while maintaining the versatility of GPCC that can handle any type of point clouds (dense, Lidar, 3D maps) . This extension is likely to use the so-called TriSoup coding scheme that works over to an octree, as detailed in next sections. TriSoup is under exploration in the standardization working group JTC1/SC29/WG7 of ISO/IEC.

[0062] There are basically three main approaches to obtain a lossy scheme over the octree representation:

[0063] 1. down-sampling + lossless + up-sampling

[0064] 2. modifying the voxels locally on the encoder side

[0065] 3. modelling the point cloud locally.

[0066] The first approach basically includes down-sampling a whole point cloud to a smaller resolution, lossless coding the down-sampled point cloud, and then up-sampling after decoding. Many up-sampling schemes have been proposed (like super resolution, AI or learning-based 3D post-processing, etc.) and good PSNR results may be provided when the down-sampling is not too aggressive, for example, there is no more than a factor 2 in each direction. However, even if the metrics show good PSNR, the visual quality is disputable and cannot be well controlled.

[0067] A second approach is to let the encoder “adjust” the point cloud locally such that the coding of the octree requires less bitrate. For this purpose, points may be slightly moved to obtain occupancy information better predicted by neighbouring nodes, thus leading to a lossless encoding of a modified octree with a lowered bitrate. Unfortunately, this approach leads to only small bitrate reduction.

[0068] The third approach is the most promising, but still requires some work before reaching maturity. The basic idea is to code the geometry by using a tree (an octree) down to some resolution, such as NxNxN blocks where N may be 4, 8 or 16. This tree is coded using a lossless scheme, like GPCC for example. The tree itself does not require too much bitrate because it does not go down to the deepest depth and has a small number of leaf nodes compared to the number of points of the point cloud. Then, in each NxNxN block, the point cloud is modelled by a local model. For example, a model may be a mean plane. Or it may be a set of triangles as described in the so-called TriSoup scheme.

[0069] TriSoup models the point cloud locally by using a set of triangles without explicitly providing connectivity information, thus its name derived from “soup of triangles” .

[0070] Vertices of triangles are coded along the edges of volumes associated with leaf nodes of the tree, as depicted on Figure 1. These vertices on edge are shared among leaf nodes having a common edge. This means that at most one vertex is coded per edge belonging to at least one leaf node. By doing so, continuity of the model is ensured through leaf nodes.

[0071] As mentioned above, the encoding of the TriSoup vertices requires two information per edge:

[0072] · a vertex flag indicating if a TriSoup vertex is present on the edge, and

[0073] · when present, the vertex position along the edge.

[0074] Consequently, the coded data consists in the octree data plus the TriSoup data. The increase of data due to the TriSoup model is more than that compensated by improvement of the reconstruction of the point cloud by the TriSoup triangles as explained hereafter.

[0075] The vertex flag is coded by an adaptive binary arithmetic coder that uses one specific context for coding vertex flags. The position of a vertex on an edge of length $N=2^s$ might be coded with unitary precision by pushing (bypassing/not entropy coding) s bits into the bitstream.

[0076] Inside a leaf node, triangles are constructed from the TriSoup vertices if at least three vertices are present on the edges of the leaf node. Constructed triangles are depicted in Figure 2.

[0077] Obviously, other combinations of triangles are possible. The choice of triangles comes from a three-step process

[0078] 1. determining a dominant direction along one of the three axes

[0079] 2. ordering TriSoup vertices depending on the dominant direction

[0080] 3. constructing triangle based on the ordered list of vertices

[0081] Knowledge about the exact position of the triangles within the current leaf is not necessary and can be deduced from the vertices.

[0082] Figure 3 will be used to explain this process. Each of the three axis is tested and the one maximizing the total surfaces of triangle is kept as dominant axis.

[0083] For simplicity of the figure, only the test over two axis is depicted on Figure 3.

[0084] A first test (top) along the vertical axis is performed by projecting the cube and the Trisoup vertices vertically on a 2D plane. The vertices are then ordered following a clockwise order relative to the center of the projected node (a square) . Then, triangles are constructed following a fixed rule based on the ordered vertices. Here, triangles 123 and 134 are constructed systematically when 4 vertices are involved. When 3

vertices are present, the only possible triangle is 123. When 5 vertices are present, a fixed rule may be to construct triangles. And so on, up to 12 vertices.

[0085] A second test (left) along a horizontal vertical axis is performed by projecting the cube and the Trisoup vertices horizontally on a 2D plane.

[0086] The vertical projection exhibits the 2D total surface of triangles that is maximum, thus the dominant axis is selected as vertical, and the constructed TriSoup triangles are obtained from the order of the vertical projection, as in Figure 3 inside the node. It is to be noted that taking the horizontal axis as dominant would have led to another construction of triangles.

[0087] It is to be noted that taking the horizontal axis as dominant would have led to another construction of triangles as shown in Figure 4.

[0088] The adequate selection of the dominant axis by maximizing the projected surface leads to a continuous reconstruction of the point cloud without holes.

[0089] The rendering of TriSoup triangles into points is performed by ray tracing. The set of all rendered points by ray tracing will make the decoded point cloud.

[0090] For ray tracing as shown in Figure 5, rays are launched along the three directions parallel to an axis. Their origin is a point of integer (voxelized) coordinates of precision corresponding to the sampling precision wanted for the rendering. The intersection (if any, dashed point) with one of the Trisoup triangles is then voxelized (= rounded to the closest point at the wanted sampling precision) and added to the list of rendered points.

[0091] After applying Trisoup to all leaf nodes, i.e., constructing triangles and obtaining points by ray tracing, copies of same points in the list of all rendered points are discarded (i.e. only one voxel is kept among all voxels sharing the same position and volume) to obtain a set of decoded (unique) points.

[0092] Based on the above deprescribed basic concepts, improvements might be applied to the Trisoup coding. For example, by computing a centroid point whose coordinates are the mean coordinates of all (ordered) vertices Vi, see Figure 6 where the centroid point C is depicted using checkboard filling. The centroid point is used as a pivoting point. The triangle are constructed from the ordered vertices (V_1, V_2, …, V_M) by pivoting around the centroid point C. The following M triangles are constructed

[0093] · V_1V_2C

[0094] · V_2V_3C

[0095] · …

[0096] · $V_{M-1}V_MC$

[0097] · V_MV_1C

[0098] Therein, the vertices are ordered clockwise and is not important which of the vertices is chosen as V1.

[0099] This construction preserves natural symmetries of the model without privileging arbitrary some triangles. Moreover, it provides an additional degree of freedom to improve the accuracy of the model, namely the position of the centroid point C.

[0100] Consequently, the position of the centroid point C might be further improved by coding a residual position in the bitstream such that the position centroid point C is closer from original points of the point cloud.

[0101] For example, $C = C_{mean} + C_{res}$

[0102] where C_{mean} is the mean position obtained by averaging coordinates of all (ordered) vertices, and C_{res} is a coded residual.

[0103] The coded residual may be a 3D residual. However, it is observed that a 3D residual is rarely advantageous because it requires many bits to be coded and this many bits are not fully compensated by the better accuracy of the model. Therefore, it is preferred to code a 1D residual C_{res}.

[0104] For example, a normal vector $\vec{n}$ may be constructed, as shown in Figure 7, and the residual may be determined by the following equation: $C_{res} = \alpha * \vec{n}$

[0105] where α is a 1D signed scalar value coded in the bitstream, see Figure 12. The normal vector $\vec{n}$ may be derived by the following two steps

[0106] 1. $\overrightarrow{m} = \overrightarrow{E_1} \times \overrightarrow{E_2} + \ldots + \overrightarrow{E_{M-1}} \times \overrightarrow{E_M} + \overrightarrow{E_M} \times \overrightarrow{E_1}$

[0107] 2. and then normalization $\vec{n} = \vec{m} / \sqrt{\|\vec{m}\|}$

[0108] where $\times$ is the cross product (also named vector (cross) product) between two vectors, and the edges $\overrightarrow{E_j}$ are: $\overrightarrow{E_j} = \overrightarrow{V_j C_{mean}}$.

[0109] It will be understood that the vector $\vec{n}$ may be taken parallel to an axis in order to simplify computation of its determination and the value α. The vector $\vec{n}$ may be taken parallel to the dominant axis as a good approximation of the vector computed above.

[0110] The value α is determined by the encoder, encoded into the bitstream and obtained by the decoder by decoding the bitstream. The value α may be binarized and each bit may be encoded by using a binary entropy coder such as an arithmetic coder or a context adaptive binary coder like CABAC.

[0111] The value α may be binarized into

[0112] · a flag f_0 indicating if α is equal to 0,

[0113] · a sign that indicates if $\alpha>0$ or $\alpha<0$,

[0114] · a flag f_1 indicating if $|\alpha|$ is equal to 1,

[0115] · a remainder $|\alpha|-2$ coded by an expGolomb coder.

[0116] The value α may be determined by the encoder by considering all points P_k of the point cloud belonging to a current leaf node. For each point P_k, its distance d_k from the line (C_{mean}, $\vec{n}$) is found by $d_k = \|\vec{n} \times \overrightarrow{C_{mean}P_k}\|$,

[0117] and in case this distance d_k is below a pre-defined threshold th (e.g., th =2) , the point P_k is used to compute the value α. The 1D residual r_k of a point P_k relative to the mean point C_{mean} is obtained by the scalar product (also named inner product or dot product) $r_k = \vec{n}.\overrightarrow{C_{mean}P_k}$.

[0118] The value α is thus obtained by $\alpha = \sum_{P_k \in S} r_k \ / \ |S|$

[0119] where S is the set of points P_k such that their distance d_k is below the threshold th, and |S| is the number of points belonging to this set.

[0120] In some embodiments, further improvement may be applied. For example, by refining the triangle modelling in leaf nodes to make the reconstructed surface closer to original surface at the decoder side. To be detail, after obtaining the decoded vertices and decoded centroid point C at the decoder side, the positions of vertices are adjusted towards the direction of vector $\overrightarrow{CC_{mean}}$ in leaf nodes of convex or concave areas.

[0121] An embodiment of the refinement method of triangle modelling in trisoup coding for point cloud data includes the following steps.

[0122] At the decoder side, after decoding vertices of each leaf node from a bitstream, each leaf node is traversed by iteration processing to construct triangles for getting a reconstructed point cloud by the ray tracing method.

[0123] To be detail, for each leaf node,

[0124] · if the processed leaf node has more than 3 vertices,

[0125] ·Firstly, the centroid point C is determined to construct the modelling surface consisted of triangles constructed by vertices and the centroid point C. To be detail, the mean point C_{mean} of vertices (V_0, …, V_i) in the leaf node, and a unit vector $\vec{n}_{Cres}$ of centroid residual are determined, and the magnitude value C_{res} of centroid residual is decoded from bitstream, and the centroid point C can be obtained by $C=C_{mean} + C_{res} * \vec{n}_{Cres}$, where $C_{res} * \vec{n}_{Cres}$ is the vector from C_{mean} to C, which can also be named as $\overrightarrow{CC_{mean}}$.

[0126] ·Then a prominence degree of the modelling surface in the leaf node is determined. In a preferred embodiment, it is judged if the centroid residual C_{res} is large or not relative to a leaf node size.

[0127] · If the centroid residual C_{res} is large relative to the leaf node size (it is more probably that the constructed surface based on the current vertices (V_0, …, V_i)) and centroid point C will have pointy protrusions artifacts in the generated modelling surface) , then among all the vertices V (V_1, …, V_i) in the leaf node, vertices on edges along the axis

where the vector $\overline{CC_{mean}}$ has a maximum value among all three axes, can be refined along the edges to which they belong within the boundary of the edge in the leaf node, and toward the direction that can make reconstructed surface more natural.

[0128] · Otherwise, all vertices (V_0, …, V_i) in the leaf node are not be refined.

[0129] ·Then all vertices in the leaf node (including the refined vertices V' and the other non-refined vertices V) are used to construct triangles combined with the centroid point C, and the ray tracing method is applied onto each triangle to get reconstructed points in the leaf node.

[0130] · Otherwise, if the leaf node has 3 vertices, there is no mean point C_{mean} and centroid point C in the processed leaf node, and only 1 triangle $V_1V_2V_3$ can be constructed, and the ray tracing method is applied onto the triangle to get reconstructed points in the leaf node.

[0131] · Otherwise, if the leaf node has less than 3 vertices, there is no triangle constructed for the leaf node, so no reconstructed points can be generated for the leaf node.

[0132] As described above, if the centroid residual C_{res} is large relative to the leaf node size, then among all the vertices V (V_1, …, V_i)) in the leaf node, vertices on edges along the axis where the vector $\overline{CC_{mean}}$ has the maximum value among all three axes are refined along the edges which they belong to within the boundary of the edge in the leaf node, and toward the direction that can make reconstructed surface more natural. In a preferred embodiment, the vertex refinement method includes the following steps:

[0133] · Firstly, the axis axis_max along which the vector $\overline{CC_{mean}}$ has the maximum value is determined.

[0134] · Then, vertices (V_0,…,V_i) on edges parallel to axis axis_m ax in the leaf node can be found.

[0135] · The axis_max-th dimension of these vertices is changed based on the centroid residual vector $\overline{CC_{mean}}$ of the leaf node, and the obtained vertex V'$_i$ can be described as $V'_i[axis_max]=V_i[axis_max]+offset*\vec{n}_{ofst}$,

[0136] where offset is a shift distance from V_i, and $\vec{n}_{ofst}$ is a unit vector pointing the direction of the shift. In a preferred embodiment, $offset*\vec{n}_{ofst}$ is based on $\overline{CC_{mean}}[axis_max]$ and the edge boundary, where $\overline{CC_{mean}}[axis_max]$ represents a projected vector of the centroid residual vector $\overline{CC_{mean}}$ along the axis_max direction, the edge boundary is a constrain condition to keep the vertex not exceed the leaf node which it belongs to. And in this embodiment, $\vec{n}$ is along the direction of $\overline{CC_{mean}}[axis_max]$, and offset is obtained based on the magnitude value of $\overline{CC_{mean}}[axis_max]$ and the edge boundary of the leaf node. In a preferred embedment, offset can be half of the magnitude value of $\overline{CC_{mean}}[axis_max]$. Figure 9 shows the refined surface produced by preferred embodiment, the blue points on edges are vertices V

Rule 26, 28.07.2023

used to model the surface with the trisoup method, the purple arrow line represents the projected vector of and the black points are refined vertices V' by shifting vertices V along the direction of $\overline{CC_{mean}}[axis_\max]$, the refined surface are more closer to the original surface in the leaf nodes, and the pointy protrusions artifacts has been reduced in some extent.

[0137] The vertices on edges of the leaf nodes are refined one leaf node by one leaf node following an order, the refined vertices in the leaf nodes can be used to reconstruct triangles to get the reconstructed point cloud. However, the refinement method is only based on centroid drift information of the current leaf node, thus, the vertices on edges belonging to two neighbouring leaf nodes may be shifted by different distances along the direction of $\overline{CC_{mean}}[axis_\max]$ in the following cases:

[0138] · case(a): the directions of $\overline{CC_{mean}}[axis_\max]$ along which the centroids of the two neighbouring leaf nodes drift are opposite, or

[0139] · case(b): the values that the centroids of the two neighbouring leaf nodes drift have a large difference even if their directions of $\overline{CC_{mean}}[axis_\max]$ are same.

[0140] However, some visible holes may be caused in the surface of the reconstructed point cloud, as shown in Figure 10. The blue square represents an adjacent surface of two neighbour leaf nodes, and the blue points with dashed outline on the edges represent the vertices without refining. $\overrightarrow{s1}$ and $\overrightarrow{s2}$ each represents a vertex shift vector (including direction and value) , and the dashed areas represent holes caused by vertices shifting.

Rule 26, 28.07.2023

[0141] These visible holes may cause a lower visual quality and a lower compression efficiency. Thus, the problem to be solved is to remove the holes generated by the vertex refinement method.

[0142] The present disclosure provides a method for processing a point cloud. The method includes: obtaining a bitstream, in which the bitstream contains vertex information of vertices of leaf nodes; determining first centroid residual information of a first leaf node; adjusting first vertex information of vertices of the second leaf node based on the first centroid residual information, in which the first leaf node and the second leaf node are adjacent nodes in a raster scan order; reconstructing a target point cloud corresponding to the second leaf node based on the adjusted first vertex information. In the technical solution, the vertex information of the vertices of the current leaf node can be adjusted based on centroid residual information of a leaf node subsequently adjacent to the current leaf node, so that the bitstream encoded from the point cloud can be decoded more accurately and performance of reconstructing the point cloud can be improved.

[0143] The method and apparatus for processing a point cloud, the encoder, the decoder, the readable storage medium provided in the present disclosure will be described in combination with the accompanying drawings.

[0144] With reference to Figure 11, Figure 11 illustrates a flowchart of a method for processing a point cloud. As shown in Figure 11, the method includes, but is not limited to the following steps.

[0145] S111, a bitstream is obtained. The bitstream contains vertex information of vertices leaf nodes.

[0146] S112, first centroid residual information of a first leaf node is determined.

[0147] S113, vertex information of vertices of the second leaf node is adjusted based on the first centroid residual information, in which the first leaf node and the second leaf node are adjacent nodes in a raster scan order.

[0148] S114, a target point cloud corresponding to the second leaf node is reconstructed based on the adjusted vertex information of the vertices of the second leaf node.

[0149] S115, second centroid residual information of a third leaf node is determined, in which the second leaf node is immediately preceding the first leaf node in the raster scan order, and the third leaf node is immediately following the first leaf node in the raster scan order.

[0150] S116, vertex information of vertices of the first leaf node that exclude a target vertex already adjusted based on the first centroid residual information is adjusted based on the second centroid residual information (i.e., when adjusting the vertex information of the vertices of the first leaf node based on the second centroid residual information, the target vertex already adjusted based on the first centroid residual information is not adjusted) , in which the target vertex includes a vertex on a common edge of a cuboid of the first leaf node and a cuboid of the second leaf node.

[0151] S117, a target point cloud corresponding to the first leaf node is reconstructed based on the adjusted vertex information of the vertices of the first leaf node.

[0152] In the embodiment of the present disclosure, obtaining the bitstream includes obtaining the bitstream encoded from a point cloud obtained by scanning any item.

[0153] For example, the item may be a desk, a car, a sculpture, and may also be a person, etc.

[0154] It can be understood that encoding the point cloud into a bitstream includes encoding a geometry of the point cloud using a tree (binary tree, quadtree, octree, sixteen tree etc.) .

[0155] The bitstream includes vertex information of vertices of leaf nodes.

[0156] For example, the leaf node may refer to the leaf node of an octree structure of a volume of the point cloud.

[0157] The bitstream may further include structure information of the leaf nodes.

[0158] The structure information of leaf nodes may reflect a position distribution of the cuboids of the leaf nodes.

[0159] It should be understood that, in case of determining the position distribution of the cuboids of the leaf nodes based on the structure information of the cuboids of the leaf nodes, a process order of the leaf nodes may be determined according to a raster scan order, and further the adjacent leaf nodes can be determined.

[0160] It should be noted that a cuboid of the leaf node may be referred to as the leaf node for short. For example, the vertex information of the vertices of cuboids of the leaf nodes may be referred to as the vertex information of the vertices of the leaf nodes; the first centroid residual information of the cuboid of the first leaf node may be referred to as the first centroid residual information of the first leaf node; or the second centroid residual information of the cuboid of the third leaf node may be referred to as the second centroid residual information of the third leaf node.

[0161] In some embodiments, the method for processing a point cloud may include any of steps S111 to S117. For example, S111 may be implemented as an independent embodiment, S112 may be implemented as an independent embodiment, S113 may be implemented as an independent embodiment, S114 may be implemented as an independent embodiment, S115 may be implemented as an independent embodiment, S116 may be implemented as an independent embodiment, S117 may be implemented as an independent embodiment, S111+S112+S113+S114 may be implemented as an independent embodiment, S115+S116+S117 may be implemented as an independent embodiment.

[0162] In some embodiments, S115, S116 and S117 may be optional. In different embodiments, one or more of these steps may be omitted or replaced.

[0163] It should be understood that in the embodiment, after obtaining the target point cloud corresponding to each leaf node, the target point clouds corresponding to all leaf nodes may form the complete point cloud reconstructed after decoding the bitstream.

[0164] In the embodiment of the present disclosure, in case that the first centroid residual information of the cuboid of the first leaf node is determined based on the raster scan order and existence of the second leaf node previously adjacent to the first leaf node and meeting the predetermined condition is determined, the vertex information of the vertices of the cuboid of the second leaf node is adjusted based on the first centroid residual information.

[0165] In some embodiments, determining the first centroid residual information of the cuboid of the first leaf node includes: determining a virtual position based on positions of the vertices included in the vertex information of the cuboid of the first leaf node; constructing triangles based on the positions and the virtual position; determining a normal vector of the cuboid of the first leaf node based on the constructed triangle;

determining a centroid position based on the normal vector; and determining the first centroid residual information based on the virtual position and the centroid position.

[0166] For example, as shown in Figure 6, the virtual position C_{mean} is determined based on the positions (V_1, V_2, V_3, V_4) of the vertices included in the vertex information of the cuboid of the first leaf node. Regarding the method of determining the virtual position C_{mean}, reference can be made to the related description of Figure 6.

[0167] In some embodiments, after determining the virtual position, triangles are constructed based on the positions of the vertices and the virtual position.

[0168] For example, as shown in Figure 6, the triangles constructed based on the positions of the vertices and the virtual position include V_1V_2C, V_1V_4C, V_2V_3C, V_3V_4C.

[0169] In some embodiments, after constructing the triangles based on the positions and the virtual position, a normal vector of the cuboid of the first leaf node may be determined based on the constructed triangle.

[0170] For example, as shown in Figure 7, the normal vector of the cuboid of the first leaf node determined based on the constructed triangle may be $\vec{n}$.

[0171] Regarding the method of determining normal vector $\vec{n}$ of the cuboid of the first leaf node based on the constructed triangle, reference can be made to the related description of Figure 7.

[0172] In some embodiments, after determining normal vector of the cuboid of the first leaf node, a centroid position may be determined based on the normal vector of the cuboid of the first leaf node.

[0173] For example, as shown in Figure 8, the centroid position C may be determined based on the normal vector $\vec{n}$ of the cuboid of the first leaf node.

[0174] Regarding the method of determining the centroid position C based on the normal vector $\vec{n}$ of the cuboid of the first leaf node, reference can be made to the related description of Figure 9.

[0175] In some embodiments, in case of determining the virtual position and the centroid position, the first centroid residual information is determined based on the virtual position and the centroid position.

[0176] For example, as shown in Figure 9, the first centroid residual information may be a vector $\overrightarrow{CC_{mean}}$ with the origin of the centroid position C, the destination of the virtual position C_{mean}, and the mold of α.

[0177] Regarding the method of determining the first centroid residual information based on the virtual position and the centroid position, reference can be made to the related description of Figure 10.

[0178] In some embodiments, reconstructing the target point cloud corresponding to the second leaf node based on the adjusted first vertex information includes: constructing

triangles based on positions of the vertices included in the adjusted first vertex information and a centroid position of the second leaf node; and reconstructing the target point cloud based on the constructed triangles by ray tracing.

[0179] In the embodiment, when reconstructing the target point cloud corresponding to the second leaf node based on the adjusted first vertex information, the centroid position of the second leaf node can be determined, the triangles can be constructed based on positions of the vertices included in the adjusted first vertex information and the centroid position of the second leaf node, and the target point cloud is reconstructed based on the constructed triangles by ray tracing.

[0180] Regarding the method of determining the centroid position of the second leaf node, reference can be made to the related description of the method of determining the centroid position of the first leaf node, which is not elaborated herein.

[0181] In some embodiments, in case of the cuboid of the first leaf node meeting the first condition, the first centroid residual information of the cuboid of the first leaf node is determined.

[0182] Or, in case of determining that the first leaf node is the first one of the leaf nodes and the cuboid of the first leaf node not meeting the first condition, it withholds from determining the first centroid residual information of the cuboid of the first leaf node.

[0183] Or, in case of determining that the first leaf node is not the first one of the leaf nodes and the cuboid of the first leaf node not meeting the first condition, the first centroid residual information of the cuboid of the first leaf node is determined.

[0184] The first condition may be that the vertex information of vertices of the cuboid of the first leaf node includes at least three vertices, i.e., the number of the vertices of the cuboid of the first leaf node is greater than or equal to 3.

[0185] In some embodiments, in case of determining that the vertex information of vertices of the cuboid of the first leaf node including at least three vertices, the first centroid residual information of the cuboid of the first leaf node is determined.

[0186] It should be understood that, when determining whether the first leaf node is the first one of the leaf nodes, it is determined whether the leaf node previously adjacent to the first leaf node exists. If the leaf node previously adjacent to the first leaf node exists, it is determined that the first leaf node is not the first one of the leaf nodes. If the leaf node previously adjacent to the first leaf node does not exist, it is determined that the first leaf node is the first one of the leaf nodes.

[0187] In the embodiment, in case of determining that the first leaf node is the first one of the leaf nodes and the vertex information of the vertices of the cuboid of the first leaf node includes information of less than three vertices, it is unnecessary to determine the first centroid residual information of the cuboid of the first leaf node. In this case,

the first centroid residual information of the cuboid of the first leaf node cannot be determined.

[0188] In the embodiment, in case of determining that the first leaf node is not the first one of the leaf nodes and the vertex information of the vertices of the cuboid of the first leaf node including information of less than three vertices, the first centroid residual information of the cuboid of the first leaf node is determined.

[0189] In some embodiments, in case of the vertex information of the vertices of the cuboid of the first leaf node including information of less than three vertices, the first centroid residual information of the cuboid of the first leaf node is determined as 0.

[0190] In the embodiment, in case of determining that the first leaf node is not the first one of the leaf nodes, it can be determined that there is a leaf node previously adjacent to the first leaf node. If the leaf node previously adjacent to the first leaf node exists and meets the predetermined condition, it can be determined that the leaf node previously adjacent to the first leaf node is the second leaf node. The predetermined condition may be that the vertex information of vertices of the cuboid of the first leaf node includes information of at least three vertices.

[0191] In the embodiment, in case of determining the first centroid residual information of the cuboid of the first leaf node and determining the second leaf node, the vertex information of the vertices of the cuboid of the second leaf node can be adjusted based on the first centroid residual information.

[0192] In some embodiments, the method further includes determining third centroid residual information of the cuboid of the second leaf node; determining a first movement direction and a first movement distance based on the third centroid residual information; and moving vertices of the cuboid of the second leaf node that are located on an edge parallel to the first movement direction and labeled with a first attribute the first movement distance in the first movement direction.

[0193] In the embodiment, in case of determining the second leaf node, before adjusting the vertex information of the cuboid of the second leaf node based on the first centroid residual information, the third centroid residual information of the cuboid of the second leaf node can be determined. The first movement direction and the first movement distance are determined based on the third centroid residual information. In adjusting the vertex information of the vertices of the cuboid of the second leaf node, the vertices that are located on an edge parallel to the first movement direction and labeled with a first attribute are moved the first movement distance in the first movement direction.

[0194] Regarding the method of determining the third centroid residual information of the cuboid of the second leaf node, reference can be made to the related description of determining the first centroid residual information of the cuboid of the first leaf node, which is not elaborated herein.

[0195] In the embodiment, in case of determining the third centroid residual information, the first movement distance and the first movement direction can be determined based on the third centroid residual information.

[0196] In some embodiments, determining the first movement direction and the first movement distance based on the third centroid residual information includes: determining a direction along which the third centroid residual information has a maximum component as the first movement direction; and determining the first movement distance based on a length of the maximum component.

[0197] In the embodiment, the direction along which the third centroid residual information has a maximum component can be determined as the first movement direction based on the third centroid residual information. The first movement distance can be determined based on the length of the maximum component.

[0198] In some embodiments, the first movement distance may be 1/2, 1/3, 1/5, 1/8 of the length of the maximum component, or the like.

[0199] In the embodiment, after determining the first movement direction and the first movement distance, the vertices of the cuboid of the second leaf node that are located on the edge parallel to the first movement direction and labeled with the first attribute are moved the first movement distance in the first movement direction.

[0200] In some embodiments, the method further includes labeling attributes of the vertices of the cuboids of the leaf nodes as a first attribute.

[0201] It should be understood that, in the embodiment, in case of obtaining the vertex information of the vertices of the cuboids of the leaf nodes, the attributes of the vertices of the cuboids of the leaf nodes can be labelled as the first attribute.

[0202] For example, the first attribute may be false, which indicates that the vertex labeled with the attribute of false is configured as an adjustable vertex.

[0203] For example, the first attribute may be negative, which indicates that the vertex labeled with the attribute of negative is configured as an adjustable vertex.

[0204] For example, the first attribute may be “0” , which indicates that the vertex labeled with the attribute of “0” is configured as an adjustable vertex.

[0205] For example, as shown in Figure 9, if the cuboid of the second leaf node is the lower cuboid, the third centroid residual information is $\overline{CC_{mean}}$, the first movement direction is represented by $\overline{CC_{mean}}[axis_\max]$, and the attributes of the vertices on the edge parallel to the first movement direction are all the first attribute, the vertices on the edge parallel to the first movement direction can be moved the first movement distance along the first movement direction $\overline{CC_{mean}}[axis_\max]$.

[0206] In adjusting the vertex information of vertices of the cuboid of the second leaf node, regarding the method of moving the vertices of the cuboid of the second leaf node that

are located on the edge parallel to the first movement direction and labeled with the first attribute the first movement distance in the first movement direction, reference can be made to the related description of Figure 9.

[0207] In some embodiments, adjusting the vertex information of the vertices of the cuboid of the second leaf node based on the first centroid residual information includes: determining a second movement direction and a second movement distance based on the first centroid residual information; moving vertices of the cuboid of the first leaf node that are located on an edge parallel to the second movement direction and labeled with the first attribute the second movement distance in the second movement direction; determining a position of one target vertex based on positions of two different vertices when determining existence of the two different vertices on one common edge of the cuboid of the first leaf node and the cuboid of the second leaf node; and replacing the positions of the two different vertices with the position of the one target vertex.

[0208] In the embodiment, when determining the second movement direction and the second movement distance based on the first centroid residual information, the direction along which the first centroid residual information has a maximum component may be determined as the second movement direction, and the second movement distance is determined according to a length of the maximum component.

[0209] In some embodiments, the second movement distance may be 1/2, 1/3, 1/5, 1/8 of the length of the maximum component, or the like.

[0210] In the embodiment, in case of determining the second movement direction and the second movement distance, in adjusting the vertex information of the vertices of the cuboid of the first leaf node, the vertices that are located on an edge parallel to the second movement direction and labeled with the first attribute are moved the second movement distance along the second movement direction.

[0211] For example, as shown in Figure 9, if the cuboid of the first leaf node is the upper cuboid, the first centroid residual information is $\overline{CC_{mean}}$, the second movement direction is represented by $\overline{CC_{mean}}[axis_\max]$, and the attributes of the vertices on the edge parallel to the second movement direction are all the first attribute, the vertices on the edge parallel to the second movement direction can be moved the second movement distance along the second movement direction $\overline{CC_{mean}}[axis_\max]$.

[0212] If there are two different vertices on the common edge of the cuboid of the first leaf node and the cuboid of the second leaf node, the position of one target vertex is determined based on the positions of the two different vertices, and the positions of the two different vertices are replaced with the position of the target vertex.

[0213] For example, as shown in Figure 9, if the upper cuboid is the cuboid of the first leaf node and the lower cuboid is the cuboid of the second leaf node, since the vertices of the cuboid of the first leaf node that are located on the edge parallel to the second movement direction are moved the second movement distance along the second movement direction, the vertices of the cuboid of the second leaf node that are located on the edge parallel to the first movement direction are moved the first movement distance along the first movement direction, and the first movement distance and the second movement distance may be different, there may be two different vertices on the common edge of the cuboid of the first leaf node and the cuboid of the second leaf node.

[0214] Of course, in case that the first movement direction is different from the second movement direction, it may also cause that there are two different vertices on the common edge of the cuboid of the first leaf node and the cuboid of the second leaf node.

[0215] Of course, in case that the first movement distance is different from the second movement distance and the first movement direction is different from the second movement direction, it may also cause that there are two different vertices on the common edge of the cuboid of the first leaf node and the cuboid of the second leaf node.

[0216] Accordingly, in case that there are two different vertices on the common edge of the cuboid of the first leaf node and the cuboid of the second leaf node, the position of one target vertex may be determined based on the positions of the two different vertices, and the positions of the two different vertices are replaced with the position of the target vertex.

[0217] In some embodiments, when the position of one target vertex is determined based on the positions of two different vertices, any one of the two different vertices may be selected as the target vertex, and the position of the any one of the two different vertices is determined as the position of the target vertex. Or a position of a middle point between the positions of the two different vertices may be determined as the position of the target vertex, which are not limited in the embodiments.

[0218] In some embodiments, the method further includes labeling an attribute of the target vertex as a second attribute.

[0219] For example, the second attribute may be true, which indicates that the vertex labeled with the attribute of true is configured as an un-adjustable vertex.

[0220] For example, the second attribute may be positive, which indicates that the vertex labeled with the attribute of positive is configured as an un-adjustable vertex.

[0221] For example, the second attribute may be “1” , which indicates that the vertex labeled with the attribute of “1” is configured as an un-adjustable vertex.

[0222] In the embodiment, after adjusting the vertex information of the vertices of the cuboid of the second leaf node based on the first centroid residual information, the target point cloud corresponding to the second leaf node may be obtained by decoding the bitstream based on the adjusted vertex information of the vertices of the cuboid of the second leaf node.

[0223] In the embodiment, if the second centroid residual information of the cuboid of the third leaf node subsequently adjacent to the first leaf node is determined based on the raster scan order, when adjusting vertex information of the cuboid of the first leaf node based on the second centroid residual information, the target vertex which has been adjusted based on the first centroid residual information is not adjusted. The target vertex includes a vertex on a common edge of the cuboid of the first leaf node and the cuboid of the second leaf node.

[0224] In some embodiments, in case that the cuboid of the third leaf node meets the first condition, the second centroid residual information of the cuboid of the third leaf node is determined. Or in case of determining that the third leaf node is not the first one of the leaf nodes and the cuboid of the third leaf node not meeting the first condition, the second centroid residual information of the cuboid of the third leaf node is determined.

[0225] In the embodiment, in case that the cuboid of the third leaf node meets the first condition, the second centroid residual information of the cuboid of the third leaf node is determined.

[0226] In the embodiment, in case that the third leaf node is determined as a leaf node other than the first one of the leaf nodes and the cuboid of the third leaf node does not meet the first condition, the second centroid residual information of the cuboid of the third leaf node is determined.

[0227] Regarding the first condition, reference can be made to the related description of the foregoing embodiments, which is not elaborated herein.

[0228] In case that the cuboid of the third leaf node does not meet the first condition, the second centroid residual information of the cuboid of the third leaf node is determined as 0.

[0229] In some embodiments, adjusting, based on the second centroid residual information, the vertex information of the vertices of the cuboid of the first leaf node that exclude the target vertex already adjusted based on the first centroid residual information includes: adjusting, based on the second centroid residual information, the vertex information of the vertices labeled with the first attribute of the cuboid of the first leaf node.

[0230] In the embodiment, when adjusting the vertex information of the vertices of the cuboid of the first leaf node based on the second centroid residual information, the vertex information of the vertices labeled with the first attribute of the cuboid of the

first leaf node can be adjusted, but the target vertex that has been adjusted based on the first centroid residual information is not adjusted.

[0231] In order to better understand the present disclosure, an example embodiment is provided.

[0232] A rendering method is proposed. In the method, ray tracing is applied on one leaf node later than current trisoup coding, and when processing the current leaf node to obtain its vertices and centroid point information, the vertex information of a previous leaf node can be refined based on centroid point information (which may be a centroid residual) of the current leaf node, and then the ray tracing can be applied on the previous leaf node based on refined vertices.

[0233] Also, in the proposed method, to keep there is at most 1 vertex on an edge of two neighbouring leaf nodes, the vertices of the previous leaf node that have been refined at previous iteration will not be refined when processing the current leaf node at current iteration.

[0234] By doing so, the performance of the reconstructed point cloud can be improved, and the complexity of the proposed method didn' t increase compared to current trisoup coding.

[0235] In a preferred embodiment, the leaf node is processed in a raster scan order, and in decoding process, each leaf node is traversed by iteration processing to reconstruct the point cloud. Before iterating, one data structure LeafInfor is defined, which contains information below:

[0236] A.the lowest position among 8 corner points of a leaf node, which can be named as Leafpos;

[0237] B.the centroid information CentroidDrift of a leaf node, which can include the mean position Cmean of all vertices within the leaf node, and the refined centroid position C and the centroid residual vector $\overrightarrow{CC_{mean}}$;

[0238] the vertex information of a leaf node, which includes the number TriCount of vertices, position VerticePos of each vertex and a property IsSticked of each vertex in the leaf node, in which the property IsSticked indicates if a vertex belonging to the leaf node can be refined/adjusted or not when processing the current leaf node to get refined vertices of the previous leaf node, and if the IsSticked of a vertex in the leaf node is true, then the vertex cannot be refined/adjusted; otherwise if the IsSticked of a vertex in the leaf node is false, then the vertex can be refined/adjusted, and IsSticked is set as false initially.

[0239] C.two indexes i, j can be defined for iterating of leaf nodes: where i is an index of the processed leaf node to obtain vertex information and centroid residual information iterately, the range of i can be from 0 to N-1, where N is the number of leaf nodes in

trisoup coding; and j is an index of the ray-traced leaf node to get the reconstructed point cloud iterately, the range of j can be from 1 to N, where N is the number of leaf nodes in trisoup coding, where j=i+1.

[0240] And a variable PrevLeafInfor following data structure LeafInfor is defined to store vertex information and centroid point information for ray tracing the previous leaf node when processing the current leaf node.

[0241] It is judged if the current leaf node is the first one of the leaf nodes or not (if i=0 or not) .

[0242] If it is the first one (if i=0) , then the vertex information and centroid information of the current leaf node are determined.

[0243] If the number TriCount of vertices of the current leaf node is smaller than 3, then the vertex refinement and centroid refinement process of the current leaf node are skipped, and the information (Leafpos, TriCount, the centroid residual information CentroidDrift, VerticePos, and property information IsSticked) of the current leaf node is stored in the variable PrevLeafInfor, which will be used to ray trace the current leaf node in next iteration.

[0244] If the number TriCount of vertices of the current leaf node is larger than 3 or equal to 3, then the centroid residual information will be determined and be used to refine vertices of the current leaf node, the detail process will be described later, and then the information (Leafpos, TriCount, the centroid residual information CentroidDrift, position of refined vertices VerticePos, and property information IsSticked) of the current leaf node is stored in the variable PrevLeafInfor, which will be used to ray trace the current leaf node in next iteration.

[0245] If it is not the first one, firstly, it is judged if the number TriCount_prev of vertices of the previous leaf node is smaller than 3 or not to determine if ray tracing of the previous leaf node is needed. If TriCount_prev<3, then ray tracing will not be applied on the previous leaf node. Otherwise, ray tracing will be applied on the previous leaf node based on the refined vertices of the previous leaf node using centroid information of the current leaf node in the variable PrevLeafInfor.

[0246] Then, the vertex information and centroid information of the current leaf node are determined.

[0247] If the number TriCount of vertices of the current leaf node is smaller than 3, then the vertex refinement and centroid refinement process of the current leaf node are skipped, and the information (Leafpos, TriCount, the centroid residual information CentroidDrift, VerticePos, and property information IsSticked) of the current leaf node is stored in the variable PrevLeafInfor, which will be used to ray trace the current leaf node in next iteration.

[0248] If the number TriCount of vertices of the current leaf node is larger than 3 or equal to 3, then the centroid residual information will be determined and be used to refine vertices of the current leaf node, the detail process will be described later, and then the information (Leafpos, TriCount, the centroid residual information CentroidDrift, position of refined vertices VerticePos, and property information IsSticked) of the current leaf node is stored in the variable PrevLeafInfor, which will be used to ray trace the current leaf node in next iteration.

[0249] It is judged if the number TriCount of vertices of the current leaf node is smaller than 3 or not to determine if the centroid information needs to be calculated for the current leaf node and is also used to refine vertices of the previous leaf node. If TriCount>3, then the centroid point C and centroid residual vector information are calculated as in current trisoup coding.

[0250] And if the centroid residual magnitude value is larger than a threshold Th, then an axis direction M that the centroid residual vector has the largest component is determined.

[0251] Vertices V of the current leaf node that along edges of the axis direction M are adjusted by an offset, and let it to be V' , then it iterates vertices of the previous leaf node to refine vertices V_prev along the direction M of the previous leaf node and let it to be the same position as V' only if the vertices V_prev satisfy a condition that it is not refined during previous iteration (when processing previous leaf node) .

[0252] Then the refined vertices will be stored in a memory to be used in next iteration for ray tracing.

[0253] Figure 12 illustrates a flowchart of another method for processing a point cloud. As shown in Figure 12, the method includes, but is not limited to the following steps.

[0254] S121, vertex information of vertices leaf nodes is obtained.

[0255] S122, the vertex information is encoded into a bitstream.

[0256] S123, a target point cloud is reconstructed by obtaining the vertex information from the bitstream.

[0257] The target point cloud can be reconstructed by performing the followings:

[0258] determining first centroid residual information of a first leaf node;

[0259] adjusting first vertex information of vertices of the second leaf node based on the first centroid residual information, wherein the first leaf node and the second leaf node are adjacent nodes in a raster scan order;

[0260] reconstructing a target point cloud corresponding to the second leaf node based on the adjusted first vertex information.

[0261] In the embodiment of the present disclosure, the vertex information of the vertices of leaf nodes can be obtained by encoding a geometry of the point cloud using a tree

(binary tree, quadtree, octree, sixteen tree etc.) . For example, the leaf node may refer to the leaf node of an octree structure of a volume of the point cloud.

[0262] Further, the vertex information of the vertices and the structure information of leaf nodes can be obtained.

[0263] The structure information of leaf nodes may reflect a position distribution of the cuboids of the leaf nodes.

[0264] It should be noted that a cuboid of the leaf node may be referred to as the leaf node for short. For example, the vertex information of the vertices of cuboids of the leaf nodes may be referred to as the vertex information of the vertices of the leaf nodes; the first centroid residual information of the cuboid of the first leaf node may be referred to as the first centroid residual information of the first leaf node; or the second centroid residual information of the cuboid of the third leaf node may be referred to as the second centroid residual information of the third leaf node.

[0265] In the embodiment, in case of obtaining the vertex information of the vertices and the structure information of the cuboids of leaf nodes, the vertex information and the structure information can be encoded into the bitstream.

[0266] In the embodiment, the target point cloud can be reconstructed by obtaining the vertex information from the bitstream.

[0267] Regarding the specific process of reconstructing the target point cloud, reference can be made to the related description of the foregoing embodiments, which is not elaborated herein.

[0268] Figure 13 illustrates a block diagram of an apparatus for processing a point cloud according to an embodiment of the present disclosure. As shown in Figure 13, the apparatus 10 includes a data obtaining unit 11, a determining unit 12, an adjusting unit 13 and a reconstructing unit 14.

[0269] The data obtaining unit 11 is configured to obtain a bitstream, in which the bitstream contains vertex information of vertices of leaf nodes.

[0270] The determining unit 12 is configured to determine first centroid residual information of a first leaf node;

[0271] The adjusting unit 13 is configured to adjust first vertex information of vertices of the second leaf node based on the first centroid residual information, in which the first leaf node and the second leaf node are adjacent nodes in a raster scan order.

[0272] The reconstructing unit 14 is configured to reconstruct a target point cloud corresponding to the second leaf node based on the adjusted first vertex information .

[0273] In some embodiments, the determining unit 12 is further configured to determine second centroid residual information of a third leaf node, in which the second leaf node is immediately preceding the first leaf node in the raster scan order, and the third leaf node is immediately following the first leaf node in the raster scan order.

[0274] The adjusting unit 13 is configured to adjust, based on the second centroid residual information, second vertex information of vertices of the first leaf node that exclude a target vertex already adjusted based on the first centroid residual information, in which the target vertex includes a vertex on a common edge of a cuboid of the first leaf node and a cuboid of the second leaf node.

[0275] The reconstructing unit 14 is further configured to reconstruct a target point cloud corresponding to the first leaf node based on the adjusted second vertex information.

[0276] With the apparatus for processing a point cloud according to the embodiment of the present disclosure, the vertex information of the vertices of the cuboid of the current leaf node can be adjusted based on centroid residual information of a leaf node subsequently adjacent to the current leaf node, so that the bitstream encoded from the point cloud can be decoded more accurately and performance of reconstructing the point cloud can be improved.

[0277] In some embodiments, the determining unit 12 is further configured to: determine the second centroid residual information when the third leaf node meets a first condition; or determine the second centroid residual information when determining that the third leaf node is not the first one of the leaf nodes and the third leaf node does not meet the first condition.

[0278] In some embodiments, the determining unit 12 is is further configured to: determine third centroid residual information of the second leaf node; determine a first movement direction and a first movement distance based on the third centroid residual information; and movinge vertices of the second leaf node that are located on an edge parallel to the first movement direction and labeled with a first attribute the first movement distance in the first movement direction.

[0279] In some embodiments, the determining unit 12 is further configured to: determine third centroid residual information of the second leaf node; determine a first movement direction and a first movement distance based on the third centroid residual information; and move vertices of the second leaf node that are located on an edge parallel to the first movement direction and labeled with a first attribute the first movement distance in the first movement direction.

[0280] In some embodiments, the determining unit 12 is further configured to: determine a direction along which the third centroid residual information has a maximum component as the first movement direction; and determine the first movement distance based on a length of the maximum component.

[0281] In some embodiments, the adjusting unit 13 is further configured to: determine a second movement direction and a second movement distance based on the first centroid residual information; move vertices of the first leaf node that are located on an edge parallel to the second movement direction and labeled with the first attribute the second

movement distance in the second movement direction; determine a position of one target vertex based on positions of two different vertices when determining existence of the two different vertices on one common edge of a cuboid of the first leaf node and a cuboid of the second leaf node; and replace the positions of the two different vertices with the position of the one target vertex.

[0282] In some embodiments, the apparatus further includes a processing unit configured to: label an attribute of the target vertex as a second attribute.

[0283] In some embodiments, the adjusting unit 13 is further configured to: adjust, based on the second centroid residual information, the second vertex information of the vertices labeled with the first attribute of the cuboid of the first leaf node.

[0284] In some embodiments, the determining unit 12 is further configured to: determine the first centroid residual information when the first leaf node meets a first condition; or withhold from determining the first centroid residual information when determining that the first leaf node is the first one of the leaf nodes; or determine the first centroid residual information when determining that the first leaf node is not the first one of the leaf nodes and the first leaf node does not meet the first condition .

[0285] In some embodiments, the determining unit 12 is further configured to: determine a virtual position based on positions of the vertices included in second vertex information of vertices of the first leaf node; construct triangles based on the positions of the vertices and the virtual position; determine a normal vector of the first leaf node based on the constructed triangle; determine a centroid position based on the normal vector; and determine the first centroid residual information based on the virtual position and the centroid position.

[0286] In some embodiments, the reconstructing unit 14 is further configured to: construct triangles based on positions of the vertices included in the adjusted first vertex information and a centroid position of the second leaf node; and reconstruct the target point cloud based on the constructed triangles by ray tracing.

[0287] In some embodiments, the apparatus further includes a processing unit configured to: label attributes of the vertices of the cuboids of the leaf nodes as a first attribute.

[0288] Figure 14 illustrates a block diagram of another apparatus for processing a point cloud according to an embodiment of the present disclosure. As shown in Figure 14, the apparatus 100 includes an information obtaining unit 101, an encoding unit 102, and a point cloud reconstructing unit 103.

[0289] The information obtaining unit 101 is configured to obtain vertex information of vertices leaf nodes.

[0290] The encoding unit 102 is configured to encode the vertex information into a bitstream.

[0291] The point cloud reconstructing unit 103 is configured to reconstruct a target point cloud by obtaining the vertex information from the bitstream.

[0292] The target point cloud is reconstructed by performing the followings:

[0293] determining first centroid residual information of a first leaf node;

[0294] adjusting first vertex information of vertices of the second leaf node based on the first centroid residual information, wherein the first leaf node and the second leaf node are adjacent nodes in a raster scan order;

[0295] reconstructing a target point cloud corresponding to the second leaf node based on the adjusted first vertex information.

[0296] In order to implement the above embodiments, the present disclosure also proposes a computer storage medium.

[0297] The computer storage medium provided by the embodiment of the present disclosure stores an executable program; after the executable program is executed by a processor, the method for processing a point cloud provided by any of the foregoing technical solutions can be implemented, for example, as shown in at least one of Figure. 11 to 12.

[0298] In order to implement the above embodiments, the present disclosure also proposes a computer program product, including a computer program. When the computer program is executed by a processor, the method for processing a point cloud as described above can be implemented.

[0299] In order to implement the above embodiments, the present disclosure further provides a computer program, which, when executed by a processor, implements the method for processing a point cloud described in the embodiments of the present disclosure.

[0300] Reference is now made to Figure 15, which shows a simplified block diagram of an example embodiment of an encoder or decoder 300. The encoder or decoder 300 includes a processor 301 and a memory storage device 303. The memory storage device 303 may store a computer program or application containing instructions that, when executed, cause the processor 301 to perform operations such as those described herein. For example, the instructions may encode and output bitstreams encoded or decode bitstreams and output points of a point cloud in accordance with the methods described herein. It will be understood that the instructions may be stored on a non-transitory computer-readable medium, such as a compact disc, flash memory device, random access memory, hard drive, etc. When the instructions are executed, the processor 301 carries out the operations and functions specified in the instructions so as to operate as a special-purpose processor that implements the described process (es) . Such a processor may be referred to as a "processor circuit" or "processor circuitry" in some examples.

[0301] It will be appreciated that the decoder and/or encoder according to the present application may be implemented in a number of computing devices, including, without limitation, servers, suitably programmed general purpose computers, machine vision systems, and mobile devices. The decoder or encoder may be implemented by way of software containing instructions for configuring a processor or processors to carry out the functions described herein. The software instructions may be stored on any suitable non-transitory computer-readable memory, including CDs, RAM, ROM, Flash memory, etc.

[0302] It will be understood that the decoder and/or encoder described herein and the module, routine, process, thread, or other software component implementing the described method/process for configuring the encoder or decoder may be realized using standard computer programming techniques and languages. The present application is not limited to particular processors, computer languages, computer programming conventions, data structures, other such implementation details. Those skilled in the art will recognize that the described processes may be implemented as a part of computer-executable code stored in volatile or non-volatile memory, as part of an application-specific integrated chip (ASIC) , etc.

[0303] Certain adaptations and modifications of the described embodiments can be made. Therefore, the above discussed embodiments are considered to be illustrative and not restrictive. In particular, embodiments can be freely combined with each other.

Claims

[Claim 1] Method for processing a point cloud, including:
obtaining a bitstream, wherein the bitstream contains vertex information of vertices of leaf nodes;
determining first centroid residual information of a first leaf node;
adjusting first vertex information of vertices of the second leaf node based on the first centroid residual information, wherein the first leaf node and the second leaf node are adjacent nodes in a raster scan order; and
reconstructing a target point cloud corresponding to the second leaf node based on the adjusted first vertex information.

[Claim 2] The method according to claim 1, further including:
determining second centroid residual information of a third leaf node, wherein the second leaf node is immediately preceding the first leaf node in the raster scan order, and the third leaf node is immediately following the first leaf node in the raster scan order;
adjusting, based on the second centroid residual information, second vertex information of vertices of the first leaf node that exclude a target vertex already adjusted based on the first centroid residual information, wherein the target vertex includes a vertex on a common edge of a cuboid of the first leaf node and a cuboid of the second leaf node; and
reconstructing a target point cloud corresponding to the first leaf node based on the adjusted second vertex information.

[Claim 3] The method according to claim 2, wherein determining second centroid residual information of the second leaf node includes:
determining the second centroid residual information when the third leaf node meets a first condition; or
determining the second centroid residual information when determining that the third leaf node is not the first one of the leaf nodes and the third leaf node does not meet the first condition.

[Claim 4] The method according to any of claims 2-3, further including:
determining third centroid residual information of the second leaf node;
determining a first movement direction and a first movement distance based on the third centroid residual information; and
moving vertices of the second leaf node that are located on an edge parallel to the first movement direction and labeled with a first attribute the first movement distance in the first movement direction.

[Claim 5] The method according to claim 4, wherein determining the first movement direction and the first movement distance based on the third centroid residual information includes:
determining a direction along which the third centroid residual information has a maximum component as the first movement direction; and
determining the first movement distance based on a length of the maximum component.

[Claim 6] The method according to claim 4 or 5, wherein adjusting the first vertex information of the vertices of the second leaf node based on the first centroid residual information includes:
determining a second movement direction and a second movement distance based on the first centroid residual information;
moving vertices of the first leaf node that are located on an edge parallel to the second movement direction and labeled with the first attribute the second movement distance in the second movement direction;
determining a position of one target vertex based on positions of two different vertices when determining existence of the two different vertices on one common edge of a cuboid of the first leaf node and a cuboid of the second leaf node; and
replacing the positions of the two different vertices with the position of the one target vertex.

[Claim 7] The method according to claim 6, further including:
labeling an attribute of the target vertex as a second attribute.

[Claim 8] The method according to any of claims 2-7, wherein adjusting, based on the second centroid residual information, the second vertex information of the vertices of the first leaf node that exclude the target vertex already adjusted based on the first centroid residual information includes:
adjusting, based on the second centroid residual information, the second vertex information of the vertices labeled with a first attribute of the first leaf node.

[Claim 9] The method according to any of claims 1-8, wherein determining first centroid residual information of the first leaf node includes:
determining the first centroid residual information when the first leaf node meets a first condition; or

withholding from determining the first centroid residual information when determining that the first leaf node is the first one of the leaf nodes; or

determining the first centroid residual information when determining that the first leaf node is not the first one of the leaf nodes and the first leaf node does not meet the first condition.

[Claim 10] The method according to any of claims 1-9, wherein determining the first centroid residual information of the first leaf node includes:

determining a virtual position based on positions of the vertices included in second vertex information of vertices of the first leaf node;

constructing triangles based on the positions of the vertices and the virtual position;

determining a normal vector of the first leaf node based on the constructed triangle;

determining a centroid position based on the normal vector; and

determining the first centroid residual information based on the virtual position and the centroid position.

[Claim 11] The method according to any of claims 1-10, wherein reconstructing the target point cloud corresponding to the second leaf node based on the adjusted first vertex information includes:

constructing triangles based on positions of the vertices included in the adjusted first vertex information and a centroid position of the second leaf node; and

reconstructing the target point cloud based on the constructed triangles by ray tracing.

[Claim 12] The method according to any of claims 1-6, further including:

labeling attributes of the vertices of the cuboids of the leaf nodes as a first attribute.

[Claim 13] Method for processing a point cloud, including:

obtaining vertex information of vertices of leaf nodes;

encoding the vertex information into a bitstream;

reconstructing a target point cloud by obtaining the vertex information from the bitstream,

wherein reconstructing the target point cloud includes:

determining first centroid residual information of a first leaf node;

adjusting first vertex information of vertices of the second leaf node based on the first centroid residual information, wherein the first leaf node and the second leaf node are adjacent nodes in a raster scan order;

reconstructing a target point cloud corresponding to the second leaf node based on the adjusted first vertex information.

[Claim 14] Apparatus for processing a point cloud, including:
a data obtaining unit, configured to obtain a bitstream, wherein the bitstream contains vertex information of vertices of leaf nodes;
a determining unit, configured to, determine first centroid residual information of a first leaf node;
an adjusting unit, configured to adjust first vertex information of vertices of the second leaf node based on the first centroid residual information, wherein the first leaf node and the second leaf node are adjacent nodes in a raster scan order; and
a reconstructing unit, configured to reconstruct a target point cloud corresponding to the first leaf node based on the adjusted first vertex information.

[Claim 15] Apparatus for processing a point cloud, including:
an information obtaining unit, configured to obtain vertex information of vertices of leaf node;
an encoding unit, configured to encode the vertex information into a bitstream;
a point cloud reconstructing unit, configured to reconstruct a target point cloud by obtaining the vertex information from the bitstream, wherein reconstructing the target point cloud includes:
determining first centroid residual information of a first leaf node;
adjusting first vertex information of vertices of the second leaf node based on the first centroid residual information, wherein the first leaf node and the second leaf node are adjacent nodes in a raster scan order;
reconstructing a target point cloud corresponding to the second leaf node based on the adjusted first vertex information.

[Claim 16] Bitstream encoded by the method according to claim 13.

[Claim 17] Decoder comprising at least one processor and a memory, wherein the memory stores instructions when executed by the processor perform the steps of the method according to any of claims 1 and 12.

[Claim 18] Encoder comprising at least one processor and a memory, wherein the memory stores instructions when executed by the processor perform the steps of the method according to claim 13.

[Claim 19] Computer-readable storage medium comprising instructions when executed by a processor to perform the steps of the method according to any of claims 1 to 12.

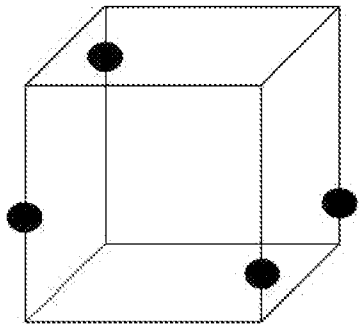


Figure 1

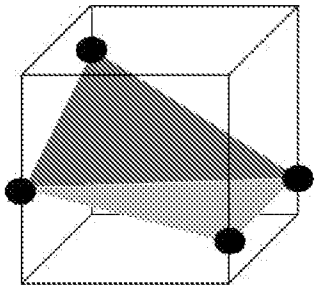


Figure 2

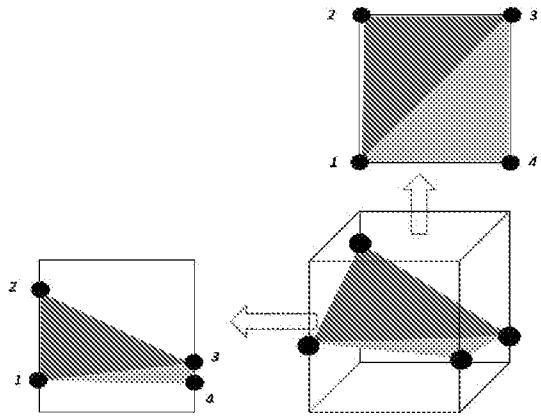


Figure 3

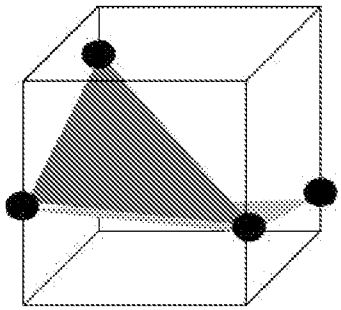


Figure 4

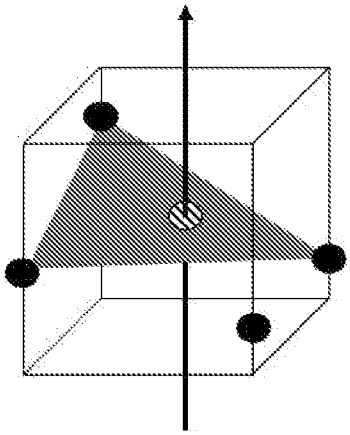


Figure 5

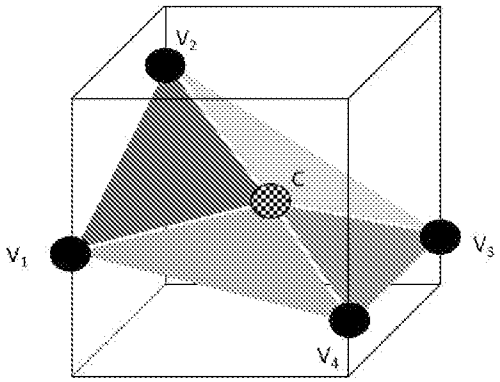


Figure 6

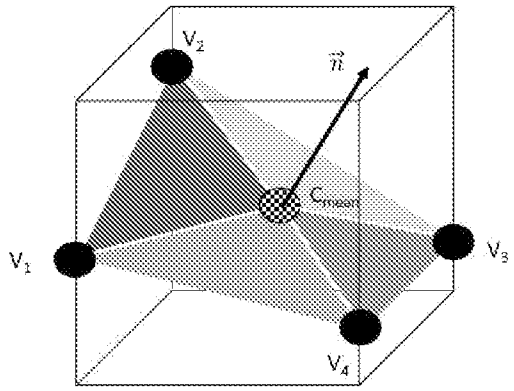


Figure 7

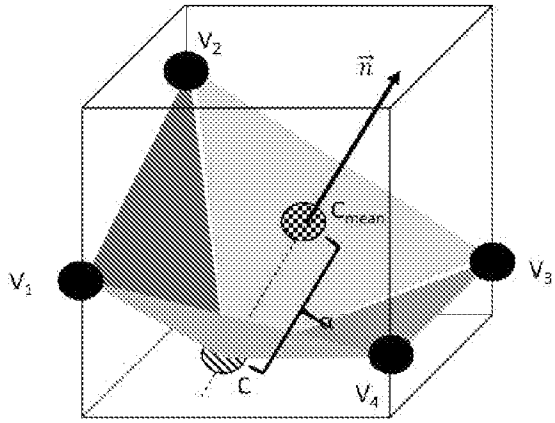


Figure 8

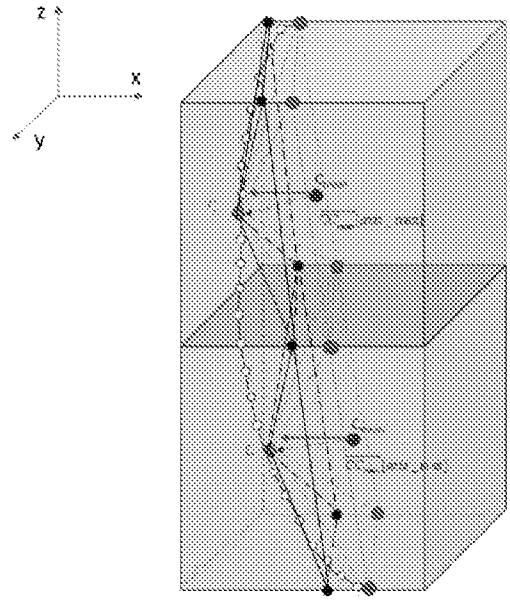


Figure 9

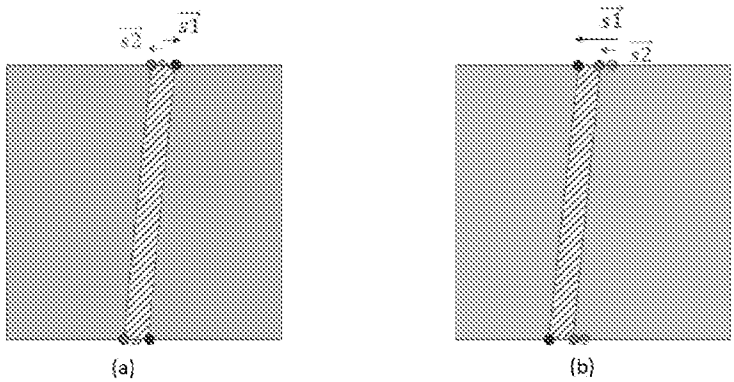


Figure 10

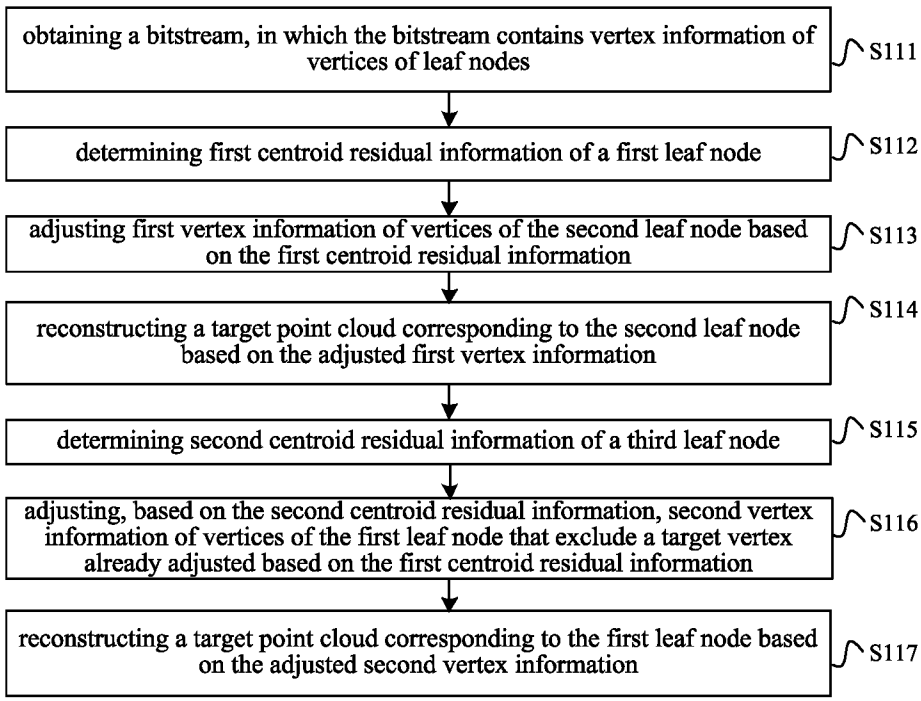


Figure 11

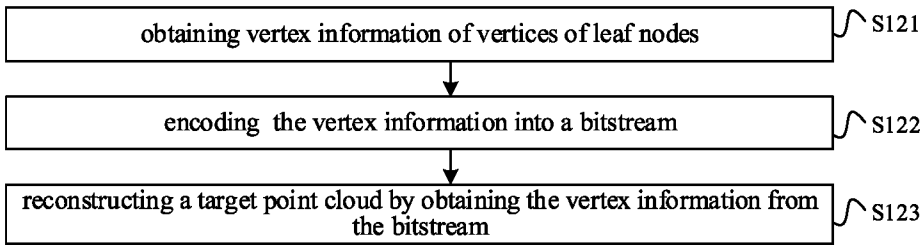


Figure 12

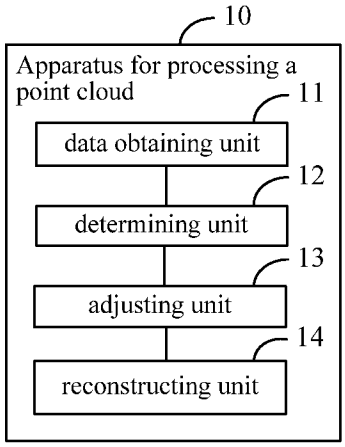


Figure 13

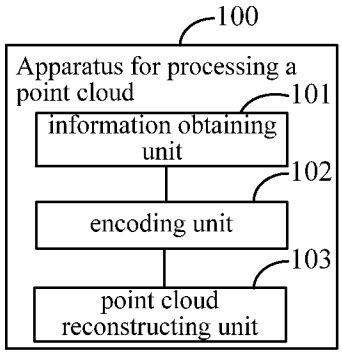


Figure 14

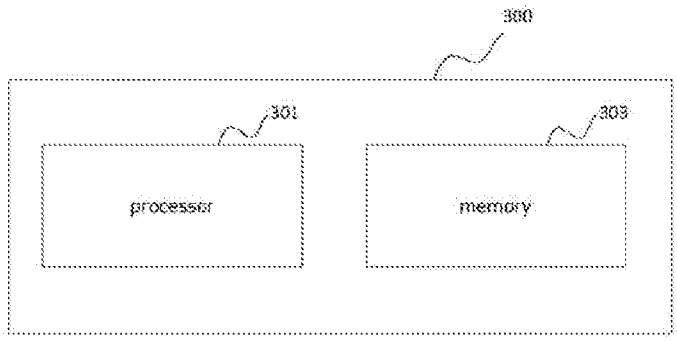


Figure 15

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2023/107611

A. CLASSIFICATION OF SUBJECT MATTER

H04N19/184(2014.01)i; H04N19/597(2014.01)i; H04N19/96(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)

IPC: 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)

CNTXT, CNABS, WPABS, DWPI, ENTXT, IEEE, CNKI, JVET: code, decode, encode, point cloud, vertex, leaf, node, centroid, center, residual, adjust, modulate, modify, rectify, adjacent, neighboring

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2022286713 A1 (LG ELECTRONICS INC.) 08 September 2022 (2022-09-08) description, paragraphs 0090-0435	1-19
A	WO 2020197086 A1 (LG ELECTRONICS INC.) 01 October 2020 (2020-10-01) the whole document	1-19
A	CN 108668136 A (HUAWEI TECHNOLOGIES CO., LTD.) 16 October 2018 (2018-10-16) the whole document	1-19
A	CN 116033186 A (TENCENT TECHNOLOGY SHENZHEN CO., LTD.) 28 April 2023 (2023-04-28) the whole document	1-19
A	CN 116112688 A (OPPO GUANGDONG MOBILE TELECOMMUNICATIONS CO., LTD.) 12 May 2023 (2023-05-12) the whole document	1-19

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

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

Date of the actual completion of the international search	Date of mailing of the international search report
11 March 2024	**15 March 2024**

Name and mailing address of the ISA/CN	Authorized officer
CHINA NATIONAL INTELLECTUAL PROPERTY ADMINISTRATION **6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing 100088, China**	**XU,Wei** Telephone No. **(+86) 010-53961818**

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2023/107611

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
US	2022286713	A1	08 September 2022	WO	2020190090	A1	24 September 2020
WO	2020197086	A1	01 October 2020		None		
CN	108668136	A	16 October 2018	US	2020021831	A1	16 January 2020
				WO	2018177254	A1	04 October 2018
				TW	201904301	A	16 January 2019
				TWI	678104	B	21 November 2019
CN	116033186	A	28 April 2023		None		
CN	116112688	A	12 May 2023	WO	2021184380	A1	23 September 2021
				US	2023019767	A1	19 January 2023
				CN	115299057	A	04 November 2022