

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局

(43) 国际公布日
2021年4月8日 (08.04.2021)



(10) 国际公布号
WO 2021/062736 A1

(51) 国际专利分类号:
G06T 9/40 (2006.01) H04N 19/593 (2014.01)
G06T 7/50 (2017.01)

(21) 国际申请号: PCT/CN2019/109653

(22) 国际申请日: 2019年9月30日 (30.09.2019)

(25) 申请语言: 中文

(26) 公布语言: 中文

(71) 申请人: **OPPO 广东移动通信有限公司 (GUANGDONG OPPO MOBILE TELECOMMUNICATIONS CORP., LTD.)** [CN/CN]; 中国广东省东莞市长安镇乌沙海滨路18号, Guangdong 523860 (CN)。

(72) 发明人: **张伟 (ZHANG, Wei)**; 中国广东省东莞市长安镇乌沙海滨路18号, Guangdong 523860 (CN)。

杨付正 (YANG, Fuzheng); 中国广东省东莞市长安镇乌沙海滨路18号, Guangdong 523860 (CN)。

万帅 (WAN, Shuai); 中国广东省东莞市长安镇乌沙海滨路18号, Guangdong 523860 (CN)。

马彦卓 (MA, Yanzhuo); 中国广东省东莞市长安镇乌沙海滨路18号, Guangdong 523860 (CN)。

霍俊彦 (HUO, Junyan); 中国广东省东莞市长安镇乌沙海滨路18号, Guangdong 523860 (CN)。

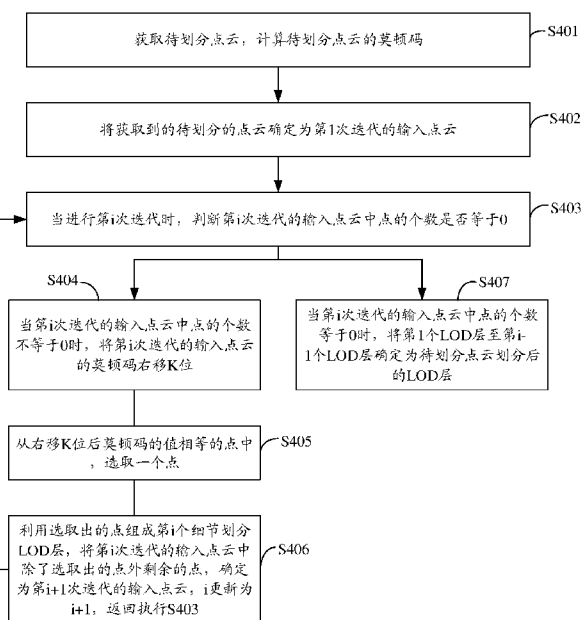
孙泽星 (SUN, Zexing); 中国广东省东莞市长安镇乌沙海滨路18号, Guangdong 523860 (CN)。

(74) 代理人: 北京派特恩知识产权代理有限公司 (CHINA PAT INTELLECTUAL PROPERTY OFFICE); 中国北京市海淀区海淀南路21号中关村知识产权大厦B座2层, Beijing 100080 (CN)。

(81) 指定国 (除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG,

(54) Title: DIVISION METHOD, ENCODER, DECODER, AND COMPUTER STORAGE MEDIUM

(54) 发明名称: 划分方法、编码器、解码器以及计算机存储介质



S401 Obtain a point cloud to be divided, and calculate the Morton code of said point cloud
S402 Determine said obtained point cloud as the input point cloud of the first iteration
S403 During the i-th iteration, determine whether the number of points in the input point cloud of the i-th iteration is equal to 0
S404 If the number of points in the input point cloud of the i-th iteration is not equal to 0, right shift the Morton code of the input point cloud of the i-th iteration by K bits
S405 Select a point from the points where the values of the Morton code after right shifted right by K bits are equal
S406 Use the selected point to form the i-th level of detail (LOD) layer, determine the remaining points in the input point cloud of the i-th iteration except for the selected point as the input point cloud of the (i+1)-th iteration, update i to i+1, and return to the execution of S403
S407 If the number of points in the input point cloud of the i-th iteration is equal to 0, determine the first LOD layer to the (i-1)-th LOD layer as the LOD layers after the point cloud to be divided is divided

图 4

(57) Abstract: Disclosed in embodiments of the present application are a division method, an encoder, a decoder, and a computer storage medium. The method comprises: calculating the Morton code of an obtained point cloud to be divided; determining said point cloud as the input point cloud of the first iteration; during the i-th iteration, determining whether the number of points in the input point cloud of the i-th iteration is equal to 0; if the number of points in the input point cloud of the i-th iteration is not equal to 0, right shifting the Morton code of the input point cloud of the i-th iteration by K bits; selecting a point from the points where the values of the Morton code after right shifted by K bits are equal; using the selected point to form the i-th LOD layer, determining the remaining points in the



WO 2021/062736 A1

BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW。

(84) 指定国 (除另有指明, 要求每一种可提供的地区保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:

— 包括国际检索报告 (条约第21条(3))。

input point cloud of the i -th iteration except for the selected point as the input point cloud of the $(i+1)$ -th iteration, updating i to $i+1$, and returning to the execution of during the i -th iteration, determining whether the number of points in the input point cloud of the i -th iteration is equal to 0; and if the number of points in the input point cloud of the i -th iteration is equal to 0, determining the first LOD layer to the $(i-1)$ -th LOD layer as the LOD layers after the point cloud to be divided is divided.

(57) 摘要: 本申请实施例公开了一种划分方法、编码器、解码器以及计算机存储介质, 该方法包括: 计算获取到的待划分点云的莫顿码, 将待划分的点云确定为第1次迭代的输入点云, 当进行第 i 次迭代时, 判断第 i 次迭代的输入点云中点的个数是否等于0, 当不等于0时, 将第 i 次迭代的输入点云的莫顿码右移 K 位, 从右移 K 位后莫顿码的值相等的点中, 选取一个点, 利用选取出的点组成第 i 个LOD层, 将第 i 次迭代的输入点云中除了选取出的点外剩余的点, 确定为第 $i+1$ 次迭代的输入点云, i 更新为 $i+1$, 返回执行当进行第 i 次迭代时, 判断第 i 次迭代的输入点云中点的个数是否等于0, 当等于0时, 将第1个LOD层至第 $i-1$ 个LOD层确定为待划分点云划分后的LOD层。

划分方法、编码器、解码器以及计算机存储介质

技术领域

本申请实施例涉及基于几何的点云压缩（Geometry-based Point Cloud Compression, G-PCC）的编解码中细节层次（Level of detail, LOD）的划分技术，尤其涉及一种划分方法、编码器、解码器以及计算机存储介质。

背景技术

在 G-PCC 的编码器框架中，点云的几何信息和每个点云所对应的属性信息是分开进行编码的，先进行几何编码，在几何编码完成后，对几何信息进行重建；目前，属性编码主要针对颜色信息进行，在进行属性编码时，首先将颜色信息从 RGB 颜色空间转换到 YUV 颜色空间，然后利用重建的几何信息对点云重新着色，使得未编码的属性信息与重建的几何信息对应起来。在颜色信息编码中，主要有两种变换方法，一是基于 LOD 划分的提升变换，目前 LOD 划分主要分为基于距离进行划分 LOD 和基于固定采样率进行划分 LOD；二是直接对属性进行区域自适应分层变换（Region Adaptive Hierarchical Transform, RAHT）。

由于对点云基于距离进行 LOD 划分的计算复杂度较高，为了降低计算复杂度，在基于距离进行 LOD 划分的基础上使用在当前点所对应的莫顿码的一个范围内进行查找的方法来划分 LOD，然而，在视频编解码中，在对 LOD 划分完成之后，需要使用 LOD 的划分结构对邻居节点属性进行预测，然而，现有的 LOD 的划分方法得到的 LOD 层导致对邻居节点的预测精准度较低。

发明内容

本申请实施例提供一种划分方法、编码器、解码器以及计算机存储介质，能够提高对邻居节点的预测精准度，提高编解码速率。

本申请实施例的技术方案可以如下实现：

第一方面，本申请实施例提供一种划分方法，所述方法包括：

获取待划分点云，计算所述待划分点云的莫顿码；

将所述待划分的点云确定为第 1 次迭代的输入点云；

当进行第 i 次迭代时，判断第 i 次迭代的输入点云中点的个数是否等于 0；其中， i 为初始值等于 1 的正整数；

当所述第 i 次迭代的输入点云中点的个数不等于 0 时，将所述第 i 次迭代的输入点云的莫顿码右移 K 位，其中， K 为正整数；

从右移 K 位后莫顿码的值相等的点中，选取一个点；

利用选取出的点组成第 i 个细节划分 LOD 层，将所述第 i 次迭代的输入点云中除了选取出的点外剩余的点，确定为第 $i+1$ 次迭代的输入点云， i 更新为 $i+1$ ，返回执行所述当进行第 i 次迭代时，判断第 i 次迭代的输入点云中点的个数是否等于 0；

当所述第 i 次迭代的输入点云中点的个数等于 0 时，将第 1 个 LOD 层至第 $i-1$ 个 LOD 层确定为所述待划分点云划分后的 LOD 层。

第二方面，本申请实施例提供一种编码器，所述编码器包括：

第一计算模块，配置为获取待划分点云，计算所述待划分点云的莫顿码；

第一确定模块，配置为将所述待划分的点云确定为第 1 次迭代的输入点云；

5 第一判断模块，配置为当进行第 i 次迭代时，判断第 i 次迭代的输入点云中点的个数是否等于 0；其中， i 为初始值等于 1 的正整数；

第一右移模块，配置为当所述第 i 次迭代的输入点云中点的个数不等于 0 时，将所述第 i 次迭代的输入点云的莫顿码右移 K 位，其中， K 为正整数；

第一选取模块，配置为从右移 K 位后的莫顿码的值相等的点中，选取一个点；

10 第一划分模块，配置为利用选取出的点组成第 i 个细节划分 LOD 层，将所述第 i 次迭代的输入点云中除了选取出的点外剩余的点，确定为第 $i+1$ 次迭代的输入点云， i 更新为 $i+1$ ，返回所述第一判断模块执行当进行第 i 次迭代时，判断第 i 次迭代的输入点云中点的个数是否等于 0；

第二确定模块，配置为当所述第 i 次迭代的输入点云中点的个数等于 0 时，将第 1 个 LOD 层至第 $i-1$ 个 LOD 层确定为所述待划分点云划分后的 LOD 层。

15 第三方面，本申请实施例提供一种解码器，所述解码器包括：

第二计算模块，配置为获取待划分点云，计算所述待划分点云的莫顿码；

第三确定模块，配置为将所述待划分的点云确定为第 1 次迭代的输入点云；

20 第二判断模块，配置为当进行第 i 次迭代时，判断第 i 次迭代的输入点云中点的个数是否等于 0；其中， i 为初始值等于 1 的正整数；

第二右移模块，配置为当所述第 i 次迭代的输入点云中点的个数不等于 0 时，将所述第 i 次迭代的输入点云的莫顿码右移 K 位；其中， K 为正整数；

第二选取模块，配置为从右移 K 位后的莫顿码的值相等的点中，选取一个点；

25 第二划分模块，配置为利用选取出的点组成第 i 个细节划分 LOD 层，将所述第 i 次迭代的输入点云中除了选取出的点外剩余的点，确定为第 $i+1$ 次迭代的输入点云， i 更新为 $i+1$ ，返回所述第二判断模块执行当进行第 i 次迭代时，判断第 i 次迭代的输入点云中点的个数是否等于 0；

第四确定模块，配置为当所述第 i 次迭代的输入点云中点的个数等于 0 时，将第 1 个 LOD 层至第 $i-1$ 个 LOD 层确定为所述待划分点云划分后的 LOD 层。

30 第四方面，本申请实施例提供一种编码器，所述编码器包括：

处理器以及存储有所述处理器可执行指令的存储介质，所述存储介质通过通信总线依赖所述处理器执行操作，当所述指令被所述处理器执行时，执行上述一个或多个实施例所述的划分方法。

第五方面，本申请实施例提供一种编码器，所述编码器包括：

35 处理器以及存储有所述处理器可执行指令的存储介质，所述存储介质通过通信总线依赖所述处理器执行操作，当所述指令被所述处理器执行时，执行上述一个或多个实施例所述的划分方法。

第六方面，本申请实施例提供一种计算机可读存储介质，其中，存储有可执行指令，当所述可执行指令被一个或多个处理器执行的时候，所述处理器执行上述一个或多个实施例所述的划分方法。

40 本申请实施例提供了一种划分方法、编码器、解码器以及计算机存储介质，该方法可以包括：获取待划分点云，计算待划分点云的莫顿码，将待划分的点云确定为第 1 次迭代的输入点云，当进行第 i 次迭代时，判断第 i 次迭代的输入点云中点的个数是否等于 0，其中， i 为初始值等于 1 的正整数，当第 i 次迭代的输入点云中点的个数不等于 0 时，将第 i 次迭代的输入点云的莫顿码右移 K 位，其中， K 为正整数，从右移 K 位后莫

顿码的值相等的点中，选取一个点，利用选取出的点组成第 i 个细节划分 LOD 层，将第 i 次迭代的输入点云中除了选取出的点外剩余的点，确定为第 $i+1$ 次迭代的输入点云， i 更新为 $i+1$ ，返回执行当进行第 i 次迭代时，判断第 i 次迭代的输入点云中点的个数是否等于 0，当第 i 次迭代的输入点云中点的个数等于 0 时，将第 1 个 LOD 层至第 $i-1$ 个 LOD 层确定为待划分点云划分后的 LOD 层；也就是说，在本申请实施例中，通过迭代的方法，将每次迭代的输入点云的莫顿码右移 K 位，再从右移 K 位后的莫顿码的值相等的点中选取一个点，将选取出的点组成本次迭代形成的 LOD 层，通过该划分方法，由于处于莫顿码相邻的点在几何空间中往往处于相邻的位置，那么通过将莫顿码右移，基于右移后的莫顿码选取点，使得在 LOD 划分的过程中，能够将点云的分布和点云之间的空间距离进行综合考虑，进而使得得到的 LOD 层的划分是基于点云的分布和点云之间的空间距离的，这样，提高了基于 LOD 层进行邻居节点预测的精准度，从而提高了编解码的效率。

附图说明

- 图 1 为 G-PCC 编码的流程框图；
图 2 为基于距离划分 LOD 过程示意图；
图 3 为基于莫顿序划分 LOD 过程示意图；
图 4 为本申请实施例提供的一种可选的划分方法的流程示意图；
图 5 为本申请实施例提供的一种可选的划分方法的实例的流程示意图；
图 6 为本申请实施例提供的一种可选的编码器的结构示意图；
图 7 为本申请实施例提供的一种可选的解码器的结构示意图；
图 8 为本申请实施例提供的另一种可选的编码器的结构示意图；
图 9 为本申请实施例提供的另一种可选的解码器的结构示意图。

具体实施方式

下面将结合本申请实施例中的附图，对本申请实施例中的技术方案进行清楚、完整地描述。可以理解的是，此处所描述的具体实施例仅仅用于解释相关申请，而非对该申请的限定。另外还需要说明的是，为了便于描述，附图中仅示出了与有关申请相关的部分。

在本申请实施例中，在点云 G-PCC 编码器框架中，将输入三维图像模型的点云进行 slice 划分后，对每一个 slice 进行独立编码。

图 1 为 G-PCC 编码的流程框图，如图 1 所示，应用于点云编码器中，针对待编码的点云数据，先通过 slice 划分，将点云数据划分为多个 slice。在每一个 slice 中，点云的几何信息和每个点云所对应的属性信息是分开进行编码的。在几何编码过程中，对几何信息进行坐标转换，使点云全都包含在一个 bounding box（包围盒）中，然后再进行量化，这一步量化主要起到缩放的作用，由于量化取整，使得一部分点云的几何信息相同，于是在基于参数来决定是否移除重复点，量化和移除重复点这一过程又被称为体素化过程。接着对 bounding box 进行八叉树划分。在基于八叉树的几何信息编码流程中，将包围盒八等分为 8 个子立方体，对非空的（包含点云中的点）的子立方体继续进行八等分，直到划分得到的叶子结点为 $1 \times 1 \times 1$ 的单位立方体时停止划分，对叶子结点中的点进行算术编码，生成二进制的几何比特流，即几何码流。在基于 trisoup（triangle soup，三角面片集）的几何信息编码过程中，同样也要先进行八叉树划分，但区别于基于八叉

树的几何信息编码, 该 trisoup 不需要将点云逐级划分到边长为 $1 \times 1 \times 1$ 的单位立方体, 而是划分到 block (子块) 边长为 W 时停止划分, 基于每个 block 中点云的分布所形成的表面, 得到该表面与 block 的十二条边所产生的至多十二个 vertex (交点), 对 vertex 进行算术编码 (基于交点进行表面拟合), 生成二进制的几何比特流, 即几何码流。Vertex 还用于在几何重建的过程的实现, 而重建的集合信息在对点云的属性编码时使用。

在属性编码过程中, 几何编码完成, 对几何信息进行重建后, 进行颜色转换, 将颜色信息 (即属性信息) 从 RGB 颜色空间转换到 YUV 颜色空间。然后, 利用重建的几何信息对点云重新着色, 使得未编码的属性信息与重建的几何信息对应起来。在颜色信息编码过程中, 主要有两种变换方法, 一是依赖于 LOD 划分的基于距离的提升变换, 二是 RAHT 的变换, 这两种方法都会将颜色信息从空间域转换到频域, 通过变换得到高频系数和低频系数, 最后对系数进行量化 (即量化系数), 最后, 将经过八叉树划分及表面拟合的几何编码数据与量化系数处理属性编码数据进行 slice 合成后, 依次编码每个 block 的 vertex 坐标 (即算数编码), 生成二进制的属性比特流, 即属性码流。

在相关技术领域中有提到针对 Category3 序列点云在进行属性变换时采用基于距离的方法进行划分 LOD。该基于距离进行划分 LOD 的方法被用于 TMC3V0 编码器中和 TMC13 属性变换 lifting 中。基于距离进行划分 LOD 的方法还被加入到 TMC13 中 predicting 属性变换中, 也即 LOD 划分被应用于点云属性变换 Predicting 和 lifting 两种方式中。

基于距离进行 LOD 划分具体实现如下:

LOD 划分是通过一组定义的距离阈值 d_l , 其中, $l=0, \dots, L-1$, 将输入的点云划分为不同的细化级别 R_l , 其中, $l=0, \dots, L-1$, 即将点云中的点划分到不同的集合 R_l 中, 距离阈值 d_l 需要满足以下两个条件:

$$d_l < d_{l-1} \quad (1)$$

$$d_{L-1} = 0 \quad (2)$$

具体来说, LOD 的划分过程是在点云几何重建之后, 这时候点云的几何坐标信息是已知的, 划分 LOD 的过程同时应用在点云的编码端和解码端, 具体过程如下:

1) 首先, 将所有的点云置于“未访问过”的点集, 并且初始化“访问过”的点集 V 为空集;

2) 该算法过程是通过不断迭代来划分 LOD 层, 第 l 次迭代所对应的细化级别 R_l 生成过程如下所示:

a. 该算法通过迭代遍历所有的点;

b. 如果当前的点已经遍历过, 则忽略该点;

c. 否则, 分别计算该点到集合 V 中每个点的距离, 记其中最近的距离为 D ; 其中, 第 1 次迭代时将第一个点云加入到集合 V 中;

d. 如果距离 D 大于等于阈值 d_l , 则将该点添加到细化级别 R_l 和集合 V 中;

e. 重复 c-d 直至所有的点都被遍历;

3) 第 l 个 LOD 集合, 即 LOD_l 是通过将细化级别 R_0, R_1, \dots, R_l 中的点进行合并得到;

4) 该过程进行不断地迭代直到所有的 LOD 层生成或者所有的点都被遍历过。

图 2 为基于距离划分 LOD 过程示意图, 如图 2 所示, 本次待划分点云包括点 $P_0, P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8$ 和 P_9 , 其中, 原始点云顺序为: $P_0, P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8$ 和 P_9 , 采用上述基于距离划分 LOD 的方法得到的 LOD_0 为 P_4, P_5, P_0 和 P_1 , LOD_1 为 $P_4, P_5, P_0, P_1, P_3, P_2$ 和 P_6 , LOD_2 为 $P_4, P_5, P_0, P_1, P_3, P_2, P_6, P_7, P_8$ 和 P_9 。

相关技术中还提到一种降低 LOD 生成计算复杂度的方案，主要是基于莫顿序进行划分 LOD，其中，莫顿码实现了一维与二维（或多维）的转换，它通过交叉存储两个数的位产生一个数，该方案相比原始搜索全部的点进行划分 LOD 的计算复杂度降低很多，主要步骤如下：

5 首先，假设变量 P_i ，其中， $i=1, \dots, N$ ， P_i 表示输入点云中的点，变量 M_i ，其中， $i=1, \dots, N$ ， M_i 为与每个点相关的莫顿码，计算莫顿码的具体方法描述如下所示，每个点的三维坐标的分量 x 、 y 、 z 用 d 比特二进制数表示为：

$$x = \sum_{l=1}^d 2^{d-l} \cdot x_l \quad (3)$$

$$y = \sum_{l=1}^d 2^{d-l} \cdot y_l \quad (4)$$

10
$$z = \sum_{l=1}^d 2^{d-l} \cdot z_l \quad (5)$$

其中， $x_l, y_l, z_l \in \{0,1\}$ ，且 x_l, y_l, z_l 分别是 x, y, z 的最高位 ($l=1$) 到最低位 ($l=d$) 对应的二进制数值。莫顿码 M 是对 x, y, z 从最高位开始，依次交叉排列 x_l, y_l, z_l 到最低位， M 的计算公式如下所示：

$$M = \sum_{l=1}^d 2^{3(d-l)} \cdot (4x_l + 2y_l + z_l) = \sum_{l=1}^{3d} 2^{3d-l} \cdot m_l \quad (6)$$

15 其中， $m_l \in \{0,1\}$ 分别是 M 的最高位 ($l=1$) 到最低位 ($l=3d$) 的值，在得到点云中每个点的莫顿码 M 后，将点云中的点按照莫顿码由小到大的顺序进行排列。

D_0 (表示初始距离的阈值) 和 ρ (表示相邻 LOD 层划分时的距离阈值比) 分别为用户自定义的初始参数，并且 $\rho > 1$ 。设 I 表示点云中所有点的索引。在第 k 次迭代时 LOD_k 中的点会从 LOD_0 到第 LOD_{k-1} 层中查找最近邻居，即距离最近的点。该算法的具体过程如下：

1) 首先初始化划分距离阈值为 $D = D_0$ ；

2) 在第 k 次迭代时， $k=0,1 \dots N-1$ (N 为 LOD 划分的总层数)， $L(k)$ 保存属于第 k 层 LOD 中的点， $O(k)$ 保存比 LOD_k 层更高细化级别的点集，其中 $L(k)$ 和 $O(k)$ 的计算过程如下：

25 a. 首先， $O(k)$ 和 $L(k)$ 被初始化为空集；

b. 每一次迭代按照 I 中保存点的索引顺序进行遍历。每一次遍历都会计算当前点到集合 $O(k)$ 中一定范围内所有点的几何距离。基于 I 中当前点所对应的莫顿码，在集合 $O(k)$ 中进行查找第一个大于当前点所对应莫顿码点的索引，然后在该索引的一个搜索范围 $SR1$ 内进行查找 ($SR1$ 代表基于莫顿码的搜索范围，一般为 8,16,64)。如果在该范围内查找到与当前点的距离小于阈值 d_l 的点，就将当前点加入到集合 $L(k)$ 中，否则加入到集合 $O(k)$ 中；

3) 在每一次迭代的过程中，集合 $L(k)$ 和 $O(k)$ 分别进行计算，并且 $O(k)$ 中的点会被用来预测集合 $L(k)$ 中点。设集合 $R(k)=L(k) \setminus L(k-1)$ ，即 $R(k)$ 代表 LOD_{k-1} 与 LOD_k 集合相差部分的点集。对位于集合 $R(k)$ 中的点会在集合 $O(k)$ 中进行查找最近的 h 个预测邻居 (目前 h 设置为 3)，查找最近邻居的具体过程如下：

a. 对于集合 $R(k)$ 中的点 P_i ，该点所对应的莫顿码为 M_i ；

b. 在集合 $O(k)$ 查找第一个大于当前点 P_i 所对应的莫顿码 M_i 的点的索引 j ；

c. 基于索引 j 在集合 $O(k)$ 中的一个搜索范围 $[j-SR2, j+SR2]$ 内查找当前点 P_i 的最近邻居节点 ($SR2$ 代表一个搜索范围，一般为 8/16/32/64)。

4)不断地进行迭代，直至 I 中所有的点遍历完为止。

图 3 为基于莫顿序划分 LOD 过程示意图，如图 3 所示，本次待划分点云包括点 P0, P1, P2, P3, P4, P5, P6, P7, P8 和 P9，其中，原始点云顺序为：P0, P1, P2, P3, P4, P5, P6, P7, P8 和 P9，基于莫顿序升序排列的点为：P4, P1, P9, P5, P0, P6, P8, P2, P7 和 P3，采用上述基于莫顿序划分 LOD 的方法得到的 LOD0 为 P4, P5, P0 和 P1, LOD1 为 P4, P5, P0, P1, P3, P2 和 P6, LOD2 为 P4, P5, P0, P1, P3, P2, P6, P7, P8 和 P9。

基于距离进行划分 LOD 的方法使用在当前点所对应的莫顿码的一个范围内进行查找的方法，计算该范围内的点到当前点的空间几何距离并与阈值进行比较，直至所有点到当前点的距离比该阈值都大时，将该点加入到当前 LOD 层中，否则到下一层时再进行划分。经过测试得到的测试数据，基于距离划分 LOD 的方法所用的时间占据总体编码时间的 70%~80%左右，显然，计算复杂度较高。

并且，在现点云属性变换 Predicting 和 lifting 之前，对点云基于距离划分 LOD。由于空间中相邻的点属性之间存在很大的相关性，因此，可以使用基于 LOD 的结构对邻居节点属性的进行预测，从而达到去除相邻点之间属性冗余的目的。尽管如此，由于不同点云具有不同的空间分布，使得不同点云的物体密度不同，因此，现有技术基于距离的划分 LOD 过程中，基于点云的均匀分布（相当于所划分距离阈值相等）进行搜索得到的邻居点不够精准，导致预测残差较大，编码效率较低。

由于对点云基于距离进行 LOD 划分的计算复杂度较高，为了降低计算复杂度，在基于距离进行 LOD 划分的基础上使用在当前点所对应的莫顿码的一个范围内进行查找的方法来划分 LOD，然而，在视频编解码中，在对 LOD 划分完成之后，需要使用 LOD 的划分结构对邻居节点属性进行预测，由于不同点云具有不同的空间分布，使得不同点云的物体密度不同，所以，通过上述划分方法所得到的 LOD 的划分结构预测邻居节点不够精准，从而导致预测残差较大，影响编解码的效率；由此可以看出，现有的 LOD 的划分方法得到的 LOD 层导致对邻居节点的预测精准度较低。

为了提高编解码的效率，需要提高对邻居节点的预测精度，为了提高对邻居节点的预测精度，本申请实施例提供一种划分方法，图 4 为本申请实施例提供的一种可选的划分方法的流程示意图，如图 4 所示，该方法可以应用于编码器或者解码器中，该划分方法可以包括：

S401: 获取待划分点云，计算待划分点云的莫顿码；

本申请实施例提供的划分方法是对上述 lifting 和 Predicting 属性变换中 LOD 的生成过程进行改进，用于编码器和解码器，该方法在进行 lifting 或者 Predicting 变换之前先使用该改进技术进行 LOD 结构的划分，应用于图 1 中生成 LOD 的框图中。

这里，在获取到待划分点云之后，先计算待划分点云的莫顿码，在后续的迭代计算中，利用莫顿码可以知晓点与点之间的空间位置关系，有利于在划分完 LOD 层之后预测邻居节点。

S402: 将待划分的点云确定为第 1 次迭代的输入点云；

具体来说，为了将待划分的点云划分成 LOD 层，这里，采用了迭代方法进行划分，首先，获取待划分点云，其中，待划分点云中包括至少两个点，例如 10 个点，将待划分点云确定为第 1 次迭代的输入点云。

针对待划分点云中点的个数较多的情况下，为了更好更快地完成对待划分点云进行 LOD 的划分，在一种可选的实施例中，在 S402 之前，该方法可以包括：

将待划分点云的莫顿码右移 N 位，并将右移 N 位后的莫顿码的值相等的点组成一组点云，N 为正整数；

将每组点云分别确定为第 1 次迭代的输入点云。

也就是说，当待划分点云中带的个数较多时，不采用直接将待划分点云作为第 1 次迭代的输入点云，而是先计算待划分点云的莫顿码，并将莫顿码右移 N 位，这样，针对右移 N 位后的莫顿码的值相等的点来说，它们之间在空间上相邻，所以将每组右移 N 位后的莫顿码的值相等的点组成一组点云，将每组点云分别确定为第 1 次迭代的输入点云，这样，将待划分点云分成组，将每组点云作为第 1 次迭代的输入点云分贝进行迭代运算，最终得到每组点云划分后的 LOD 层，从而得到待划分点云划分后的 LOD 层。

S403: 当进行第 i 次迭代时，判断第 i 次迭代的输入点云中点的个数是否等于 0;

其中，在本申请实施例中，迭代次数用 i 来表示，i 为初始值等于 1 的正整数;

在 S403 中，在进行每次迭代时，先判断每次迭代的输入点云中点的个数是否等于 0，也就是说，通过迭代是否已经将待划分点云遍历完。如果遍历完成就结束迭代过程，如果还没有遍历完成，需要进行继续迭代，以将待划分点云划分至 LOD 层中。

S404: 当第 i 次迭代的输入点云中点的个数不等于 0 时，将第 i 次迭代的输入点云的莫顿码右移 K 位;

经过 S403 的判断，当第 i 次迭代的输入点云中点的个数不等于 0 时，说明之前的迭代还没有遍历完所有的点，所以，需要执行本次迭代，具体来说，可以根据莫顿码从小到大的顺序点进行排序，以更好的通过莫顿码得到点之间的空间关系。

其中，K 为正整数，这里的 K 的取值可以通过训练得到的固定值，也可以是通过以下任意一种方式获得的值，这里，本申请实施例对此不作具体限定。

为了确定出 K 的值，在一种可选的实施例中，S404 之前，该方法还可以包括:

根据待划分的点云的分布，确定 K 的值。

具体来说，根据待划分的点云的分布来确定 K 的值，这样，在每次迭代中都将莫顿码右移 K 位，使得在莫顿码右移时考虑到点云的空间分布，能够达到更好的预测效果。

为了确定出 K 的值，在一种可选的实施例中，当 i 大于等于第一预设阈值时，S404 之前，该方法还可以包括:

根据第 i 次迭代的输入点云中点的个数和第 i 次迭代的输入点云的分布，确定 K 的值。

这里，由于每次迭代均会将部分点云划分至对应的 LOD 层中，所以，随着迭代次数的增加，剩余的点越来越少，那么，每次迭代的输入点云的分布也在不断变化，为了实现更好的预测效果，这里，每次迭代的 K 值可以根据每次迭代的输入点云中点的个数和每次迭代的输入点云的分布来确定，这样，使得在每次迭代时莫顿码的右移位数考虑到点云的个数和点云的空间分布，能够达到更好的预测效果。

S405: 从右移 K 位后莫顿码的值相等的点中，选取一个点;

在将莫顿码右移 K 位之后，为了划分出 LOD 层，先找到右移 K 位后的莫顿码的值相等的点，可以得到至少一组右移 K 位后的莫顿码的值相等的点，然后从右移 K 位后莫顿码的值相等的点中，选取一个点，其中，可以选取其中任意一个点，也可以选取其中一个指定的点，这里，本申请实施例对此不作具体限定。

为了得到划分后的 LOD 层，以提高对邻居节点的预测精度，在一种可选的实施例中，S405 可以包括:

按照第 i 次迭代的输入点云的莫顿码从小到大的顺序，对第 i 次迭代的输入点云中的点进行排序;

从右移 K 位后的莫顿码的值相等的点中，选取出排序在第一个的点。

这里，对莫顿码按照从小到大的顺序进行排序，然后在找出右移 K 位后的莫顿码的值相等的点，得到至少一组右移 K 位后的莫顿码的值相等的点，然后从每组中选取第一

个点，也就是右移 K 位后的莫顿码的值相等的点中在原始点云顺序中排在最前面的点。

为了得到划分后的 LOD 层，以提高对邻居节点的预测精度，在一种可选的实施例中，S405 可以包括：

确定右移 K 位后的莫顿码的值相等的点的质心；

5 将右移 K 位后的莫顿码的值相等的点中距离质心最近的点，确定为选取出一个点。

具体来说，先确定出右移 K 位后的莫顿码的值相等的点的质心，该质心是通过本次迭代的输入点云中每个点的三维坐标计算出来的，在得到质心之后，然后计算右移 K 位后的莫顿码的值相等的点中距离质心最近的点，将该点选取出来。

10 为了得到划分后的 LOD 层，以提高对邻居节点的预测精度，除了采用上述 S405 来选取点之外，在一种可选的实施例中，在 S404 之后，在 S406 之前，该方法还可以包括：

根据右移 K 位后的莫顿码的值相等的点的个数，确定对应的选取个数；

根据对应的选取个数，从右移 K 位后的莫顿码的值相等的点中选取出选取个数个点。

15 具体来说，可以从右移 K 位后的莫顿码的值相等的点中选取出不止一个点，可以先根据右移 K 位后的莫顿码的值相等的点的个数，确定对应的选取个数，例如，右移 K 位后的莫顿码的值相等的点的个数有 10 个，8 个，6 个，4 个等等，确定对应的选取个数可以按照莫顿码的值相等的点的个数来确定，例如，针对右移 K 位后的莫顿码的值相等的点的个数为 10 个的，对应的选取个数为 5 个，针对右移 K 位后的莫顿码的值相等的点的个数为 8 个的，对应的选取个数为 4 个，针对右移 K 位后的莫顿码的值相等的点的个数为 6 个的，对应的选取个数为 3 个，针对右移 K 位后的莫顿码的值相等的点的个数为 4 个的，对应的选取个数为 2 个，既可以按照正比例关系来确定对应的选取个数。

20 然后，根据选取个数从右移 K 位后的莫顿码的值相等的点中选取出选取个数个点，其中，可以任意选取出选取个数个点，也可以按照指定的方式选取出选取个数个点，这里，本申请实施例对此不作具体限定。

25 为了得到划分后的 LOD 层，以提高对邻居节点的预测精度，除了采用上述 S405 来选取点之外，在一种可选的实施例中，在 S404 之后，在 S406 之前，该方法还可以包括：

按照第 i 次迭代的输入点云的莫顿码从小到大的顺序，对第 i 次迭代的输入点云中的点进行排序；

从右移 K 位后的莫顿码的值相等的点中，按照排序结果，每隔第二预设阈值选取一个点。

30 具体来说，先将莫顿码按照从小到大的顺序进行排序，得到第 i 次迭代的输入点云的排序结果，然后可以找出右移 K 位后的莫顿码的值相等的点，由于每组右移 K 位后的莫顿码的值相等的点也是按照排序结果排列的，所以，这里按照每隔第二预设阈值来选取一个点，以得到选出的点。

其中，上述第二预设阈值为根据需求设置好的值。

35 S406: 利用选出的点组成第 i 个细节划分 LOD 层，将第 i 次迭代的输入点云中除了选出的点外剩余的点，确定为第 i+1 次迭代的输入点云，i 更新为 i+1，返回执行 S403；

40 在得到选出的点之后，用选出的点组成本次迭代的 LOD 层，即第 i 个 LOD 层，从而经过本次迭代将本次迭代的输入点云中的部分点划分至 LOD 层，并将本次迭代的输入点云中除了选出的点外剩余的点确定为第 i+1 次迭代的输入点云，并将 i 更新为 i+1，重新执行 S403。

可见，在本申请实施例中，通过计算第 i 次迭代的输入点云的莫顿码，并右移适当的位数，可以将空间上相邻的点找出来，在形成 LOD 层时，从每组相邻的点中选取出一个或者多个点，这样，使得划分后的 LOD 层考虑的点云之间的空间分布，有利于提高邻居节点的预测精度。

S407: 当第 i 次迭代的输入点云中点的个数等于 0 时, 将第 1 个 LOD 层至第 $i-1$ 个 LOD 层确定为待划分点云划分后的 LOD 层。

另外, 当第 i 次迭代的输入点云中的点的个数等于 0 时, 说明此时待划分点云已经全部遍历完成, 所以, 可以结束迭代, 将得到的第 1 个 LOD 层至第 $i-1$ 个 LOD 层作为待划分点云划分后的 LOD 层。

另外, 为了划分得到 LOD 层, 在一种可选的实施例中, 当 i 大于等于第一预设阈值时, 在利用选取出的点组成第 i 个细节划分 LOD 层, 将第 i 次迭代的输入点云中除了选取出的点外剩余的点, 确定为第 $i+1$ 次迭代的输入点云, i 更新为 $i+1$ 之后, 该方法还可以包括:

根据第 i 次迭代的输入点云, 调用预设的基于距离划分 LOD 的算法进行划分得到 LOD 层;

将划分得到的 LOD 层、第 1 个 LOD 层至第 $i-1$ 个 LOD 层确定为: 待划分点云划分后的 LOD 层。

也就是说, 采用本申请实施例的迭代方法时, 当迭代次数 i 大于等于第一预设阈值时, 针对剩余的未划分的点, 可以基于距离划分 LOD 的算法, 或者基于固定采样率划分 LOD 的算法, 进行划分得到 LOD 层; 这里, 本申请实施例对此不作具体限定。

当迭代次数 i 大于等于第一预设阈值时, 在完成本次迭代的划分之后, 迭代次数 $i+1$ 之后, 根据第 i 次迭代的输入点云, 可以调用预设的基于距离划分 LOD 的算法, 或者调用预设的基于固定采样率划分 LOD 的算法, 对第 i 次迭代的输入点云进行划分, 得到 LOD 层, 最终, 将之前得到的第 1 个 LOD 层至第 $i-1$ 个 LOD 层, 和划分得到的 LOD 层作为待划分点云划分后的 LOD 层。

下面举实例来对上述一个或多个实施例所述的划分方法进行说明。

图 5 为本申请实施例提供的一种可选的划分方法的实例的流程示意图, 如图 5 所示, 该划分方法可以应用于编码器中, 也可以应用于解码器中, 本实例以应用于编码器为例, 该划分方法可以包括:

S501: 针对获取到的待编码点云, 计算莫顿码;

具体来说, 设待编码点云包含 N 个点, 每个点表示为 $P[i]$, 计算每个点 $P[i]$ 所对应的莫顿码为 $M[i]$, $i=0,1\dots N-1$, 并且对待编码点云的莫顿码按照升序进行排列;

S502: 对莫顿码进行右移 K 位;

具体来说, 通过对待编码点云中每个点的莫顿码右移 K 位, 那么处于空间中相邻位置的点所对应的莫顿码就会相等, 因为处于莫顿码相邻的点在几何空间中往往处于相邻的位置, 正是利用莫顿码的这个特点, 通过对空间中待编码点云中所有点对应的莫顿码右移 K 位之后, 将空间中相邻的点聚在一起;

S503: 对莫顿码值相等的点集进行采样;

具体来说, 对 S502 中得到的右移 K 位后的莫顿码的值相等的点进行采样, 通过对聚集在一起的点集每次迭代取一个点, 通过利用预测点的属性对被预测点的属性基于空间距离加权平均得到被预测点的重建属性值, 其次, 利用重建属性值与原始属性值得到被预测点的属性残差, 利用重建属性值和预测残差来进行邻居点的属性预测, 每次采样结束后都会进行预测, 然后综合每次的预测得到一个最终的预测, 从而实现预测待编码点云中剩余的邻居点属性;

S504: 将采样点作为当前 LOD 层中的点;

具体来说, 将 S503 得到的采样点作为当前 LOD 层中的点, 其他剩余的点作为比当前 LOD 层更高层的点; 在实际应用中, 这里, 可以将右移 K 位后的莫顿码的值相等的点中第一个点作为采样点。

S505: 判断剩余的点是否全部遍历或者是否全部划分完; 如果是, 则结束 LOD 层的划分, 否则返回执行 S502, 以继续进行划分;

在继续划分中, 在待编码点云的中莫顿码右移 K 位的基础上, 再进行右移 K 位, 重复上面的 S502 至 S505, 直至划分完所有的 LOD 或者遍历完所有的点。

5 通过上述实例, 在基于莫顿码右移方法进行划分 LOD 时, 每次仅仅需要对原始点云所对应的莫顿码右移一定位数, 然后对右移之后莫顿码相等的点进行采样, 该方法划分 LOD 的时间仅仅占编码时间的 20% 左右, 所以, 基于莫顿序右移划分 LOD 的方法相比原始基于距离划分 LOD 的方法时间复杂度降低很多, 大约是前者的 1/4~1/3 左右。

10 本申请实施例中利用点云空间分布以及点云空间距离划分 LOD, 通过莫顿码右移进行 LOD 的划分, 不仅可以几何空间的距离进行考虑, 并且会将点云的分布进行综合考虑, 过莫顿序右移方法进行划分时, 会在点云较为密集的地方减少采样点数目, 同时在较为稀疏的地方增加采样点数目。在较为稀疏的部分, 由于点和点之间的空间距离较远, 在进行预测的时候相邻点之间的属性残差一般会很大, 如果还是基于固定距离进行采样, 这样就会导致最后基于邻居进行属性预测得到的预测残差很大, 从而导致压缩之后的属性比特流很大, 并且会降低重建之后属性的性能, 在空间中稀疏的部分, 由于点与点之间的空间几何距离较大, 相邻点之间的对应莫顿码值相差很大, 所以在右移莫顿码时, 该部分会有很少的点聚为一类, 从而会得到较多的采样点, 这样在基于邻居属性进行预测时会相比原始仅仅考虑几何空间距离的方案有更好的预测效果。

15 本申请实施例通过莫顿码进行右移一定位数得到不同数目的点集, 然后在每个点集中用采样点来预测邻居点的属性。通过该方案可以将点云的分布和点云之间的空间距离进行综合考虑, 进而提升基于邻居预测属性的效果。

20 本申请实施例在基本不影响性能的前提下, 可以提高属性部分的重建质量, 并且可以降低属性的编解码时间和计算复杂度, 如下表 1 所示:

表 1

| 点云文件 | 初始右移位数 | BD-rate (属性) | | 时间 | |
|--------------------------|--------|--------------|--------|---------|---------|
| | | U | V | 编码占用时间比 | 解码占用时间比 |
| frog_00067_vox12 | 6 | -7.4% | -16.6% | 95% | 89% |
| facade_00064_vox11 | 6 | 13.3% | -4.5% | 96% | 86% |
| house_without_roof_vox12 | 6 | -3.4% | -3.3% | 88% | 78% |
| Average(平均增益) | 6 | -0.8% | -8.1% | 93% | 84% |

25 由上述表 1 可知, 重建点云的 BD-rate 也有较为明显的提升, 峰值信噪比 (PSNR, Peak Signal to Noise Ratio) 是一种图像评价的客观标准, PSNR 越大则图像的质量越好, BD-rate 是用来衡量性能的好坏的参数, BD-rate 为负时表示性能变好, 在此基础上 BD-rate 的绝对值越大, 则性能的增益越大, 且通过本申请实施例提供的划分方法降低了划分 LOD 所需的时间, 从而降低了编解码的时间。

30 本申请实施例提供了一种划分方法, 该方法可以包括: 获取待划分点云, 计算待划分点云的莫顿码, 将待划分的点云确定为第 1 次迭代的输入点云, 当进行第 i 次迭代时,

判断第 i 次迭代的输入点云中点的个数是否等于 0，其中， i 为初始值等于 1 的正整数，当第 i 次迭代的输入点云中点的个数不等于 0 时，将第 i 次迭代中输入点云的莫顿码右移 K 位，其中， K 为正整数，从右移 K 位后莫顿码的值相等的点中，选取一个点，利用选取出的点组成第 i 个细节划分 LOD 层，将第 i 次迭代的输入点云中除了选取出的点外剩余的点，确定为第 $i+1$ 次迭代的输入点云， i 更新为 $i+1$ ，返回执行当进行第 i 次迭代时，判断第 i 次迭代的输入点云中点的个数是否等于 0，当第 i 次迭代的输入点云中点的个数等于 0 时，将第 1 个 LOD 层至第 $i-1$ 个 LOD 层确定为待划分点云划分后的 LOD 层；也就是说，在本申请实施例中，通过迭代的方法，将每次迭代的输入点云的莫顿码右移 K 位，再从右移 K 位后的莫顿码的值相等的点中选取一个点，将选取出的点组成本次迭代形成的 LOD 层，通过该划分方法，由于处于莫顿码相邻的点在几何空间中往往处于相邻的位置，那么通过将莫顿码右移，基于右移后的莫顿码选取点，使得在 LOD 划分的过程中，能够将点云的分布和点云之间的空间距离进行综合考虑，进而使得得到的 LOD 层的划分是基于点云的分布和点云之间的空间距离的，这样，提高了基于 LOD 层进行邻居节点预测的精准度，从而提高了编解码的效率。

15

实施例二

基于同一发明构思下，本申请实施例提供一种编码器，图 6 为本申请实施例提供的一种可选的编码器的结构示意图，如图 6 所示，该编码器可以包括：

- 第一计算模块 61，配置为获取待划分点云，计算待划分点云的莫顿码；
- 20 第一确定模块 62，配置为将待划分的点云确定为第 1 次迭代的输入点云；
- 第一判断模块 63，配置为当进行第 i 次迭代时，判断第 i 次迭代的输入点云中点的个数是否等于 0；其中， i 为初始值等于 1 的正整数；
- 第一右移模块 64，配置为当第 i 次迭代的输入点云中点的个数不等于 0 时，将第 i 次迭代的输入点云的莫顿码右移 K 位；其中， K 为正整数；
- 25 第一选取模块 65，配置为从右移 K 位后的莫顿码的值相等的点中，选取一个点；
- 第一划分模块 66，配置为利用选取出的点组成第 i 个细节划分 LOD 层，将第 i 次迭代的输入点云中除了选取出的点外剩余的点，确定为第 $i+1$ 次迭代的输入点云， i 更新为 $i+1$ ，返回第一判断模块 63 执行当进行第 i 次迭代时，判断第 i 次迭代的输入点云中点的个数是否等于 0；
- 30 第二确定模块 67，配置为当第 i 次迭代的输入点云中点的个数等于 0 时，将第 1 个 LOD 层至第 $i-1$ 个 LOD 层确定为待划分点云划分后的 LOD 层。

在一种可选的实施例中，第一确定模块 62 还可以配置为：

在当第 i 次迭代的输入点云中点的个数不等于 0 时，将第 i 次迭代的输入点云的莫顿码右移 K 位之前，根据待划分的点云的分布，确定 K 的值。

- 35 在一种可选的实施例中，当 i 大于等于第一预设阈值时，第一确定模块 62 还可以配置为：

在当第 i 次迭代的输入点云中点的个数不等于 0 时，将第 i 次迭代的输入点云的莫顿码右移 K 位之前，根据第 i 次迭代的输入点云中点的个数和第 i 次迭代的输入点云的分布，确定 K 的值。

- 40 在一种可选的实施例中，当 i 大于等于第一预设阈值时，

在利用选取出的点组成第 i 个细节划分 LOD 层，将第 i 次迭代的输入点云中除了选取出的点外剩余的点，确定为第 $i+1$ 次迭代的输入点云， i 更新为 $i+1$ 之后，该编码器还可以配置为：

- 45 根据第 i 次迭代的输入点云，调用预设的基于距离划分 LOD 的算法进行划分得到 LOD 层；

将划分得到的 LOD 层、第 1 个 LOD 层至第 $i-1$ 个 LOD 层确定为：待划分点云划分后的 LOD 层。

在一种可选的实施例中，第一选取模块 65 具体可以配置为：

按照第 i 次迭代的输入点云的莫顿码从小到大的顺序，对第 i 次迭代的输入点云中的点进行排序；

从右移 K 位后的莫顿码的值相等的点中，选取出排序在第一个的点。

在一种可选的实施例中，第一选取模块 65 具体可以配置为：

确定右移 K 位后的莫顿码的值相等的点的质心；

将右移 K 位后的莫顿码的值相等的点中距离质心最近的点，确定为选取出一个点。

在一种可选的实施例中，第一选取模块 65 具体可以配置为：

根据右移 K 位后的莫顿码的值相等的点的个数，确定对应的选取个数；

根据对应的选取个数，从右移 K 位后的莫顿码的值相等的点中选取出选取个数个点。

在一种可选的实施例中，第一选取模块 65 具体可以配置为：

按照第 i 次迭代的输入点云的莫顿码从小到大的顺序，对第 i 次迭代的输入点云中的点进行排序；

从右移 K 位后的莫顿码的值相等的点中，按照排序结果，每隔第二预设阈值选取一个点。

在一种可选的实施例中，第一确定模块 62 还可以配置为：

将待划分点云的莫顿码右移 N 位，并将右移 N 位后的莫顿码的值相等的点组成一组点云；其中， N 为正整数；

将每组点云作分别确定为第 1 次迭代的输入点云。

本申请实施例提供一种解码器，图 7 为本申请实施例提供的一种可选的解码器的结构示意图，如图 7 所示，该解码器可以包括：

第二计算模块 71，配置为获取待划分点云，计算待划分点云的莫顿码；

第三确定模块 72，配置为将待划分的点云确定为第 1 次迭代的输入点云；

第二判断模块 73，配置为当进行第 i 次迭代时，判断第 i 次迭代的输入点云中点的个数是否等于 0；其中， i 为初始值等于 1 的正整数；

第二右移模块 74，配置为当第 i 次迭代的输入点云中点的个数不等于 0 时，将第 i 次迭代的输入点云的莫顿码右移 K 位；其中， K 为正整数；

第二选取模块 75，配置为从右移 K 位后的莫顿码的值相等的点中，选取一个点；

第二划分模块 76，配置为利用选取出的点组成第 i 个细节划分 LOD 层，将第 i 次迭代的输入点云中除了选取出的点外剩余的点，确定为第 $i+1$ 次迭代的输入点云， i 更新为 $i+1$ ，返回第一判断模块 63 执行当进行第 i 次迭代时，判断第 i 次迭代的输入点云中点的个数是否等于 0；

第四确定模块 77，配置为当第 i 次迭代的输入点云中点的个数等于 0 时，将第 1 个 LOD 层至第 $i-1$ 个 LOD 层确定为待划分点云划分后的 LOD 层。

在一种可选的实施例中，第三确定模块 72 还可以配置为：

在第 i 次迭代的输入点云中点的个数不等于 0 时，将第 i 次迭代的输入点云的莫顿码右移 K 位之前，根据待划分的点云的分布，确定 K 的值。

在一种可选的实施例中，当 i 大于等于第一预设阈值时，第三确定模块 72 还可以配置为：

在第 i 次迭代的输入点云中点的个数不等于 0 时，将第 i 次迭代的输入点云的莫顿码右移 K 位之前，根据第 i 次迭代的输入点云中点的个数和第 i 次迭代的输入点云的分布，确定 K 的值。

在一种可选的实施例中，当 i 大于等于第一预设阈值时，

在利用选取出的点组成第 i 个细节划分 LOD 层，将第 i 次迭代的输入点云中除了选取出的点外剩余的点，确定为第 $i+1$ 次迭代的输入点云， i 更新为 $i+1$ 之后，该解码器还可以配置为：

5 根据第 i 次迭代的输入点云，调用预设的基于距离划分 LOD 的算法进行划分得到 LOD 层；

将划分得到的 LOD 层、第 1 个 LOD 层至第 $i-1$ 个 LOD 层确定为：待划分点云划分后的 LOD 层。

在一种可选的实施例中，第二选取模块 75 具体可以配置为：

10 按照第 i 次迭代的输入点云的莫顿码从小到大的顺序，对第 i 次迭代的输入点云中的点进行排序；

从右移 K 位后的莫顿码的值相等的点中，选取出排序在第一个的点。

在一种可选的实施例中，第二选取模块 75 具体可以配置为：

确定右移 K 位后的莫顿码的值相等的点的质心；

15 将右移 K 位后的莫顿码的值相等的点中距离质心最近的点，确定为选取出一个点。

在一种可选的实施例中，第二选取模块 75 具体可以配置为：

根据右移 K 位后的莫顿码的值相等的点的个数，确定对应的选取个数；

根据对应的选取个数，从右移 K 位后的莫顿码的值相等的点中选取出选取个数个点。

在一种可选的实施例中，第二选取模块 75 具体可以配置为：

20 按照第 i 次迭代的输入点云的莫顿码从小到大的顺序，对第 i 次迭代的输入点云中的点进行排序；

从右移 K 位后的莫顿码的值相等的点中，按照排序结果，每隔第二预设阈值选取一个点。

在一种可选的实施例中，第三确定模块 72 还可以配置为：

25 将待划分点云的莫顿码右移 N 位，并将右移 N 位后的莫顿码的值相等的点组成一组点云；其中， N 为正整数；

将每组点云作分别确定为第 1 次迭代的输入点云。

图 8 为本申请实施例提出的另一种可选的编码器的结构示意图，如图 8 所示，本申请实施例提出的编码器 800 还可以包括处理器 81 以及存储有处理器 81 可执行指令的存储介质 82，存储介质 82 通过通信总线 83 依赖处理器 81 执行操作，当指令被处理器 81 30 执行时，执行上述一个或多个实施例所述的划分方法。

需要说明的是，实际应用时，编码器 800 中的各个组件通过通信总线 83 耦合在一起。可理解，通信总线 83 用于实现这些组件之间的连接通信。通信总线 83 除包括数据总线之外，还包括电源总线、控制总线和状态信号总线。但是为了清楚说明起见，在图 35 8 中将各种总线都标为通信总线 83。

图 9 为本申请实施例提出的另一种可选的解码器的结构示意图，如图 9 所示，本申请实施例提出的解码器 900 还可以包括处理器 91 以及存储有处理器 91 可执行指令的存储介质 92，存储介质 92 通过通信总线 93 依赖处理器 91 执行操作，当指令被处理器 91 40 执行时，执行上述一个或多个实施例所述的划分方法。

需要说明的是，实际应用时，解码器 900 中的各个组件通过通信总线 93 耦合在一起。可理解，通信总线 93 用于实现这些组件之间的连接通信。通信总线 93 除包括数据总线之外，还包括电源总线、控制总线和状态信号总线。但是为了清楚说明起见，在图 9 中将各种总线都标为通信总线 93。

本申请实施例提供了一种计算机存储介质，存储有可执行指令，当所述可执行指令

被一个或多个处理器执行的时候，所述处理器执行上述一个或多个实施例所述的划分方法。

可以理解，本申请实施例中的存储器可以是易失性存储器或非易失性存储器，或可包括易失性和非易失性存储器两者。其中，非易失性存储器可以是只读存储器（Read-Only Memory, ROM）、可编程只读存储器（Programmable ROM, PROM）、可擦除可编程只读存储器（Erasable PROM, EPROM）、电可擦除可编程只读存储器（Electrically EPROM, EEPROM）或闪存。易失性存储器可以是随机存取存储器（Random Access Memory, RAM），其用作外部高速缓存。通过示例性但不是限制性说明，许多形式的RAM可用，例如静态随机存取存储器（Static RAM, SRAM）、动态随机存取存储器（Dynamic RAM, DRAM）、同步动态随机存取存储器（Synchronous DRAM, SDRAM）、双倍数据速率同步动态随机存取存储器（Double Data Rate SDRAM, DDRSDRAM）、增强型同步动态随机存取存储器（Enhanced SDRAM, ESDRAM）、同步连接动态随机存取存储器（Synchlink DRAM, SLDRAM）和直接内存总线随机存取存储器（Direct Rambus RAM, DRRAM）。本文描述的系统和方法的存储器旨在包括但不限于这些和任意其它适合类型的存储器。

而处理器可能是一种集成电路芯片，具有信号的处理能力。在实现过程中，上述方法的各步骤可以通过处理器中的硬件的集成逻辑电路或者软件形式的指令完成。上述的处理器可以是通用处理器、数字信号处理器（Digital Signal Processor, DSP）、专用集成电路（Application Specific Integrated Circuit, ASIC）、现场可编程门阵列（Field Programmable Gate Array, FPGA）或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件。可以实现或者执行本申请实施例中的公开的各方法、步骤及逻辑框图。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。结合本申请实施例所公开的方法的步骤可以直接体现为硬件译码处理器执行完成，或者用译码处理器中的硬件及软件模块组合执行完成。软件模块可以位于随机存储器、闪存、只读存储器、可编程只读存储器或者电可擦写可编程存储器、寄存器 etc 本领域成熟的存储介质中。该存储介质位于存储器，处理器读取存储器中的信息，结合其硬件完成上述方法的步骤。

可以理解的是，本文描述的这些实施例可以用硬件、软件、固件、中间件、微码或其组合来实现。对于硬件实现，处理单元可以实现在一个或多个专用集成电路（Application Specific Integrated Circuits, ASIC）、数字信号处理器（Digital Signal Processing, DSP）、数字信号处理设备（DSP Device, DSPD）、可编程逻辑设备（Programmable Logic Device, PLD）、现场可编程门阵列（Field-Programmable Gate Array, FPGA）、通用处理器、控制器、微控制器、微处理器、用于执行本申请所述功能的其它电子单元或其组合中。

对于软件实现，可通过执行本文所述功能的模块（例如过程、函数等）来实现本文所述的技术。软件代码可存储在存储器中并通过处理器执行。存储器可以在处理器中或在处理器外部实现。

需要说明的是，在本文中，术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含，从而使得包括一系列要素的过程、方法、物品或者装置不仅包括那些要素，而且还包括没有明确列出的其他要素，或者是还包括为这种过程、方法、物品或者装置所固有的要素。在没有更多限制的情况下，由语句“包括一个……”限定的要素，并不排除在包括该要素的过程、方法、物品或者装置中还存在另外的相同要素。

上述本申请实施例序号仅仅为了描述，不代表实施例的优劣。

通过以上的实施方式的描述，本领域的技术人员可以清楚地了解到上述实施例方法

可借助软件加必需的通用硬件平台的方式来实现，当然也可以通过硬件，但很多情况下前者是更佳的实施方式。基于这样的理解，本申请的技术方案本质上或者说对现有技术做出贡献的部分可以以软件产品的形式体现出来，该计算机软件产品存储在一个存储介质（如 ROM/RAM、磁碟、光盘）中，包括若干指令用以使得一台终端（可以是手机、计算机、服务器、或者网络设备等）执行本申请各个实施例所述的方法。

上面结合附图对本申请的实施例进行了描述，但是本申请并不局限于上述的具体实施方式，上述的具体实施方式仅仅是示意性的，而不是限制性的，本领域的普通技术人员在本申请的启示下，在不脱离本申请宗旨和权利要求所保护的范围情况下，还可做出很多形式，这些均属于本申请的保护之内。

10 工业实用性

本申请实施例提供了一种划分方法、编码器、解码器以及计算机存储介质，获取待划分点云，计算待划分点云的莫顿码，将待划分的点云确定为第 1 次迭代的输入点云，当进行第 i 次迭代时，判断第 i 次迭代的输入点云中点的个数是否等于 0，当第 i 次迭代的输入点云中点的个数不等于 0 时，将第 i 次迭代的输入点云的莫顿码右移 K 位，从右移 K 位后莫顿码的值相等的点中，选取一个点，利用选取出的点组成第 i 个 LOD 层，将第 i 次迭代的输入点云中除了选取出的点外剩余的点，确定为第 $i+1$ 次迭代的输入点云， i 更新为 $i+1$ ，返回执行当进行第 i 次迭代时，判断第 i 次迭代的输入点云中点的个数是否等于 0，当第 i 次迭代的输入点云中点的个数等于 0 时，将第 1 个 LOD 层至第 $i-1$ 个 LOD 层确定为待划分点云划分后的 LOD 层，提高了基于 LOD 层进行邻居节点预测的精准度，从而提高了编解码的效率。

权利要求书

1、一种划分方法，其中，包括：

获取待划分点云，计算所述待划分点云的莫顿码；

将所述待划分点云确定为第 1 次迭代的输入点云；

5 当进行第 i 次迭代时，判断第 i 次迭代的输入点云中点的个数是否等于 0；其中， i 为初始值等于 1 的正整数；

当所述第 i 次迭代的输入点云中点的个数不等于 0 时，将所述第 i 次迭代的输入点云的莫顿码右移 K 位；其中， K 为正整数；

从右移 K 位后莫顿码的值相等的点中，选取一个点；

10 利用选取出的点组成第 i 个细节划分 LOD 层，将所述第 i 次迭代的输入点云中除了选取出的点外剩余的点，确定为第 $i+1$ 次迭代的输入点云， i 更新为 $i+1$ ，返回执行所述当进行第 i 次迭代时，判断第 i 次迭代的输入点云中点的个数是否等于 0；

当所述第 i 次迭代的输入点云中点的个数等于 0 时，将第 1 个 LOD 层至第 $i-1$ 个 LOD 层确定为所述待划分点云划分后的 LOD 层。

15 2、根据权利要求 1 所述的方法，其中，在当所述第 i 次迭代的输入点云中点的个数不等于 0 时，将所述第 i 次迭代的输入点云的莫顿码右移 K 位之前，所述方法还包括：

根据所述待划分的点云的分布，确定 K 的值。

3、根据权利要求 1 所述的方法，其中，当 i 大于等于第一预设阈值时，在当所述第 i 次迭代的输入点云中点的个数不等于 0 时，将所述第 i 次迭代的输入点云的莫顿码右移 K 位之前，所述方法还包括：

20 根据所述第 i 次迭代的输入点云中点的个数和所述第 i 次迭代的输入点云的分布，确定 K 的值。

4、根据权利要求 1 所述的方法，其中，当 i 大于等于第一预设阈值时，在利用选取出的点组成第 i 个细节划分 LOD 层，将所述第 i 次迭代的输入点云中除了选取出的点外剩余的点，确定为第 $i+1$ 次迭代的输入点云， i 更新为 $i+1$ 之后，所述方法还包括：

25 根据所述第 i 次迭代的输入点云，调用预设的基于距离划分 LOD 的算法进行划分得到 LOD 层；

将划分得到的 LOD 层、第 1 个 LOD 层至第 $i-1$ 个 LOD 层确定为：所述待划分点云划分后的 LOD 层。

30 5、根据权利要求 1 所述的方法，其中，所述从右移 K 位后的莫顿码的值相等的点中，选取一个点，包括：

按照所述第 i 次迭代的输入点云的莫顿码从小到大的顺序，对所述第 i 次迭代的输入点云中的点进行排序；

从所述右移 K 位后的莫顿码的值相等的点中，选取排序在第一个的点。

35 6、根据权利要求 1 所述的方法，其中，所述从右移 K 位后的莫顿码的值相等的点中，选取一个点，包括：

确定所述右移 K 位后的莫顿码的值相等的点的质心；

将所述右移 K 位后的莫顿码的值相等的点中距离所述质心最近的点，确定为选取出的一个点。

40 7、根据权利要求 1 所述的方法，其中，在当所述第 i 次迭代的输入点云中点的个数不等于 0 时，将所述第 i 次迭代的输入点云的莫顿码右移 K 位之后，在利用选取出的点组成第 i 个细节划分 LOD 层之前，所述方法还包括：

根据右移 K 位后的莫顿码的值相等的点的个数，确定对应的选取个数；

根据所述对应的选取个数，从所述右移 K 位后的莫顿码的值相等的点中选取所述选取个数个点。

8、根据权利要求 1 所述的方法，其中，在当所述第 i 次迭代的输入点云中点的个数
5 不等于 0 时，将所述第 i 次迭代的输入点云的莫顿码右移 K 位之后，在利用选取出的点组成第 i 个细节划分 LOD 层之前，所述方法还包括：

按照所述第 i 次迭代的输入点云的莫顿码从小到大的顺序，对所述第 i 次迭代的输入点云中的点进行排序；

10 从右移 K 位后的莫顿码的值相等的点中，按照排序结果，每隔第二预设阈值选取一个点。

9、根据权利要求 1 所述的方法，其中，在获取待划分点云，计算所述待划分点云的莫顿码之后，在当进行第 i 次迭代时，判断第 i 次迭代的输入点云中点的个数是否等于 0 之前，所述方法还包括：

15 将所述待划分点云的莫顿码右移 N 位，并将右移 N 位后的莫顿码的值相等的点组成一组点云；其中，N 为正整数；

将每组点云作分别确定为所述第 1 次迭代的输入点云。

10、一种编码器，其中，包括：

第一计算模块，配置为获取待划分点云，计算所述待划分点云的莫顿码；

第一确定模块，配置为将所述待划分的点云确定为第 1 次迭代的输入点云；

20 第一判断模块，配置为当进行第 i 次迭代时，判断第 i 次迭代的输入点云中点的个数是否等于 0；其中，i 为初始值等于 1 的正整数；

第一右移模块，配置为当所述第 i 次迭代的输入点云中点的个数不等于 0 时，将所述第 i 次迭代的输入点云的莫顿码右移 K 位；其中，K 为正整数；

第一选取模块，配置为从右移 K 位后的莫顿码的值相等的点中，选取一个点；

25 第一划分模块，配置为利用选取出的点组成第 i 个细节划分 LOD 层，将所述第 i 次迭代的输入点云中除了选取出的点外剩余的点，确定为第 i+1 次迭代的输入点云，i 更新为 i+1，返回所述第一判断模块执行当进行第 i 次迭代时，判断第 i 次迭代的输入点云中点的个数是否等于 0；

30 第二确定模块，配置为当所述第 i 次迭代的输入点云中点的个数等于 0 时，将第 1 个 LOD 层至第 i-1 个 LOD 层确定为所述待划分点云划分后的 LOD 层。

11、一种解码器，其中，包括：

第二计算模块，配置为获取待划分点云，计算所述待划分点云的莫顿码；

第三确定模块，配置为将所述待划分的点云确定为第 1 次迭代的输入点云；

35 第二判断模块，配置为当进行第 i 次迭代时，判断第 i 次迭代的输入点云中点的个数是否等于 0；其中，i 为初始值等于 1 的正整数；

第二右移模块，配置为当所述第 i 次迭代的输入点云中点的个数不等于 0 时，将所述第 i 次迭代的输入点云的莫顿码右移 K 位；其中，K 为正整数；

第二选取模块，配置为从右移 K 位后的莫顿码的值相等的点中，选取一个点；

40 第二划分模块，配置为利用选取出的点组成第 i 个细节划分 LOD 层，将所述第 i 次迭代的输入点云中除了选取出的点外剩余的点，确定为第 i+1 次迭代的输入点云，i 更新为 i+1，返回所述第二判断模块执行当进行第 i 次迭代时，判断第 i 次迭代的输入点云中点的个数是否等于 0；

第四确定模块，配置为当所述第 i 次迭代的输入点云中点的个数等于 0 时，将第 1 个 LOD 层至第 i-1 个 LOD 层确定为所述待划分点云划分后的 LOD 层。

12、一种编码器，其中，所述编码器包括：

处理器以及存储有所述处理器可执行指令的存储介质，所述存储介质通过通信总线依赖所述处理器执行操作，当所述指令被所述处理器执行时，执行上述的权利要求 1 至 9 任一项所述的划分方法。

5 13、一种解码器，其中，所述解码器包括：

处理器以及存储有所述处理器可执行指令的存储介质，所述存储介质通过通信总线依赖所述处理器执行操作，当所述指令被所述处理器执行时，执行上述的权利要求 1 至 9 任一项所述的划分方法。

10 14、一种计算机可读存储介质，其中，存储有可执行指令，当所述可执行指令被一个或多个处理器执行的时候，所述处理器执行所述的权利要求 1 至 9 任一项所述的划分方法。

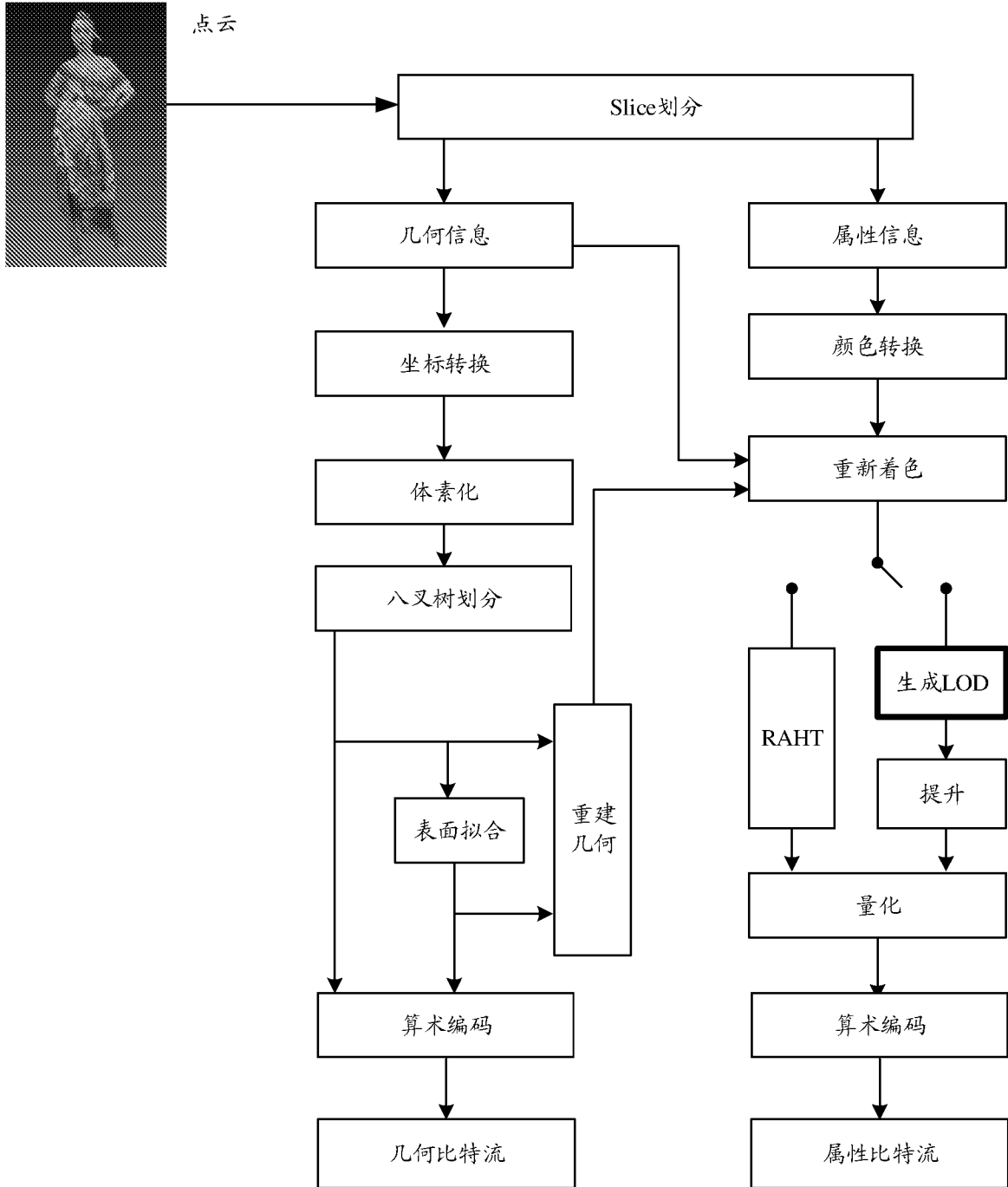


图 1

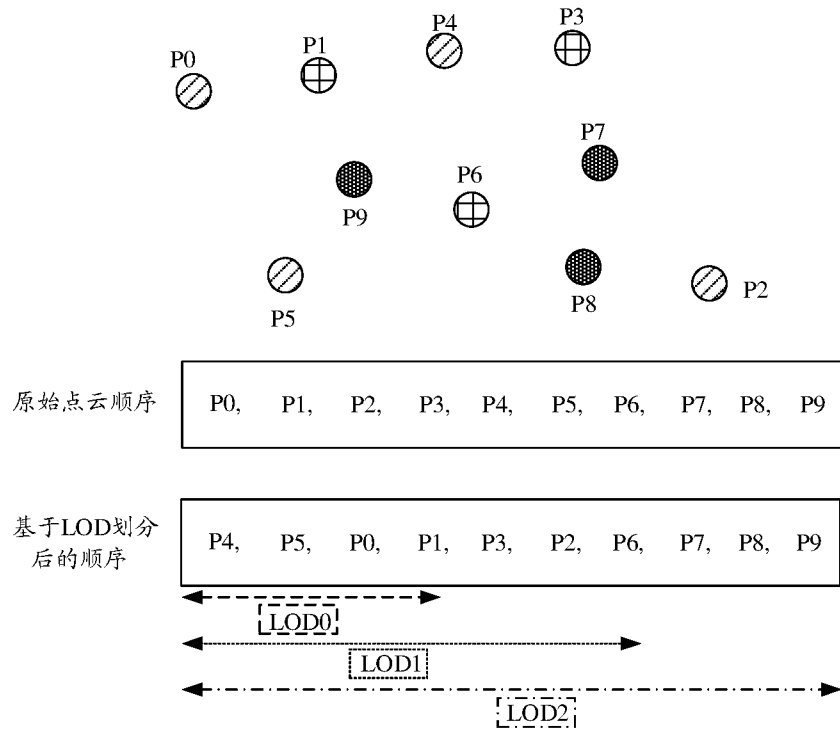


图 2

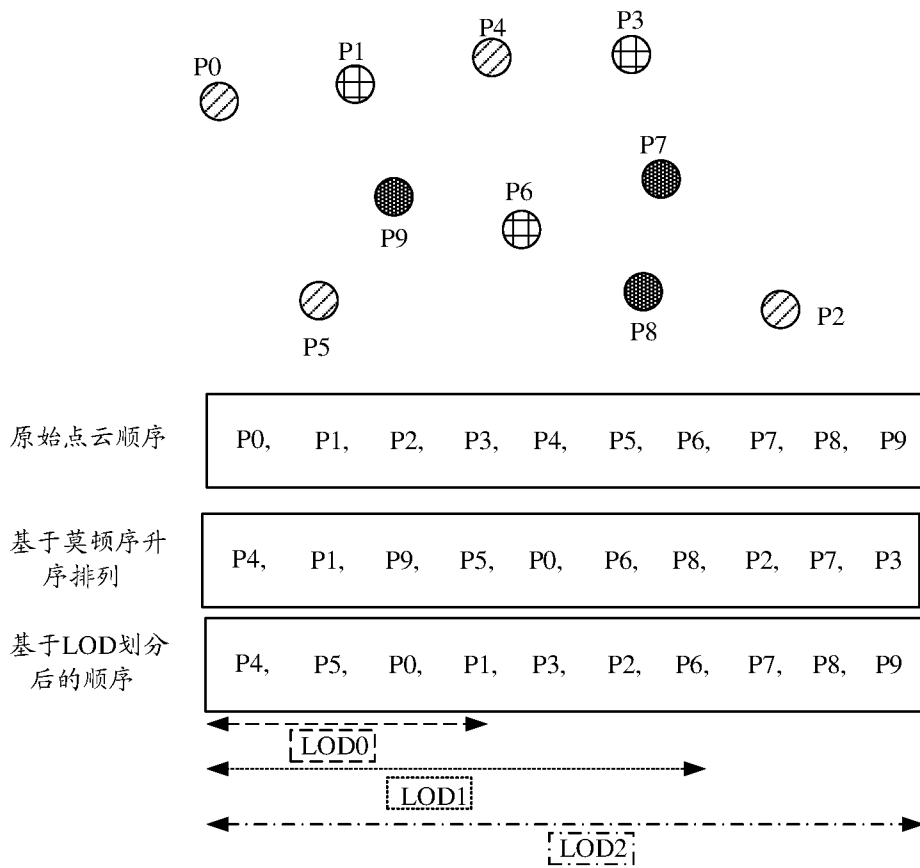


图 3

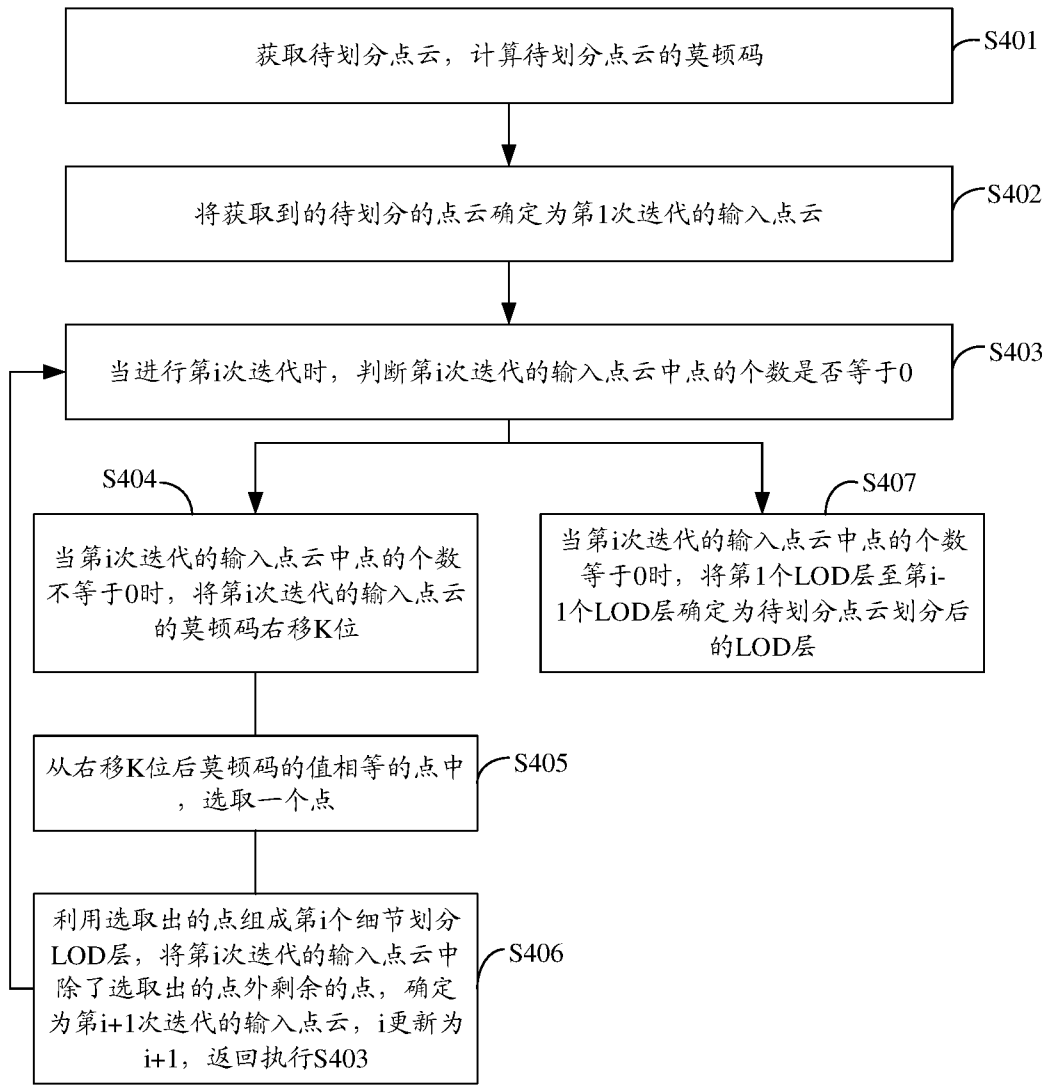


图 4

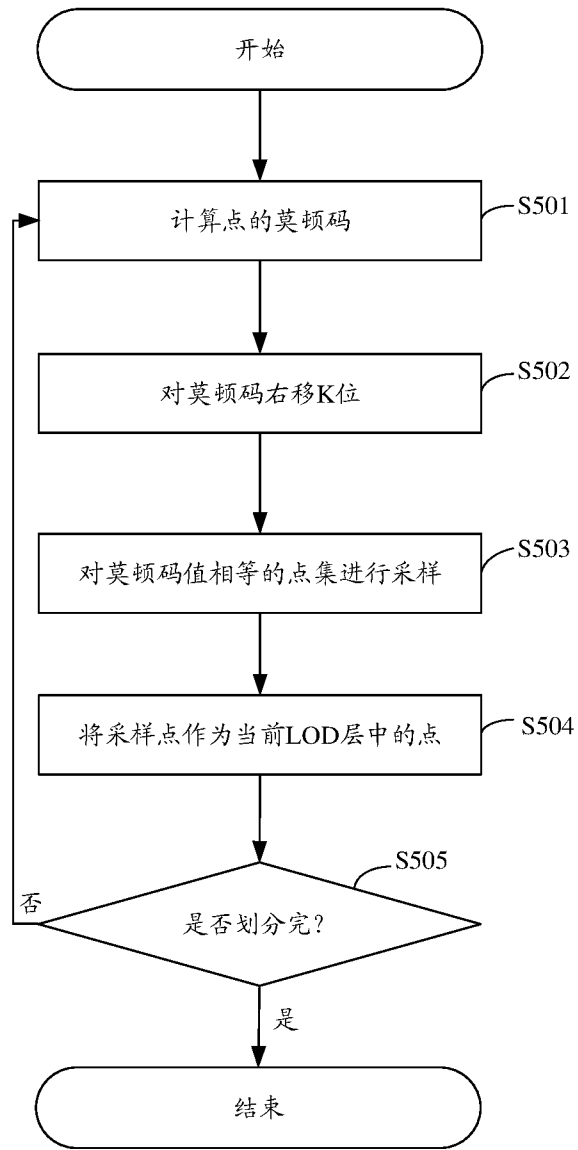


图 5

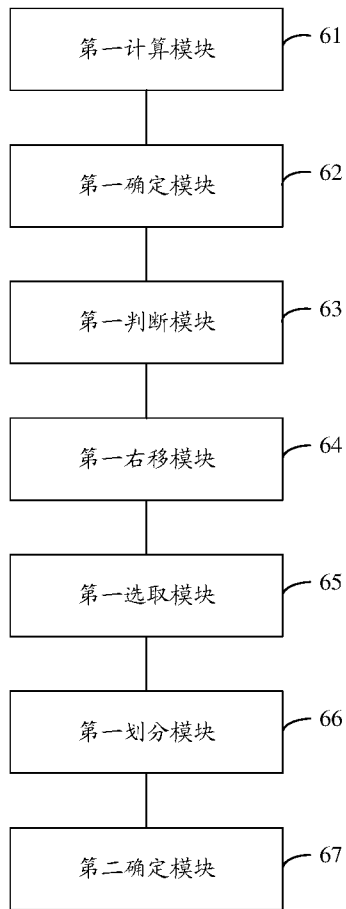


图 6

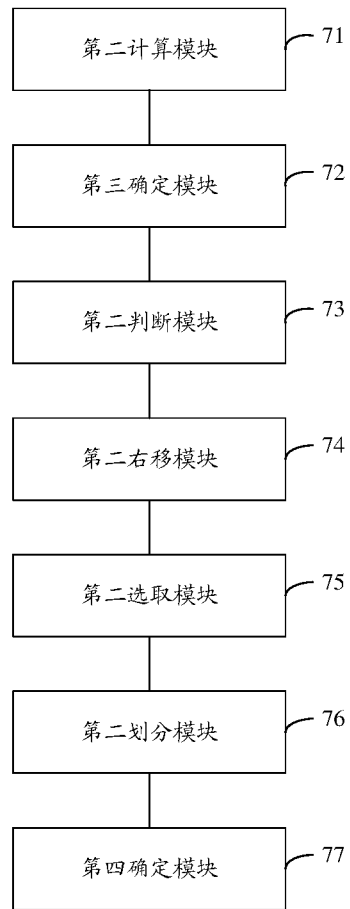


图 7

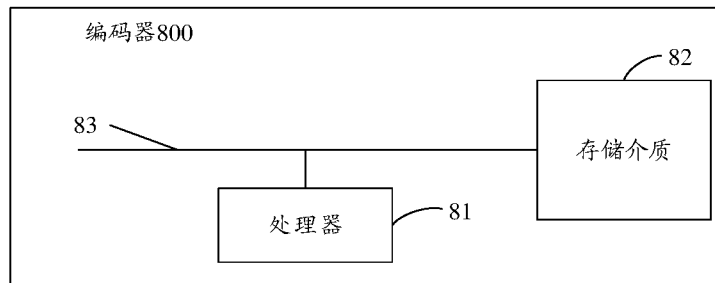


图 8

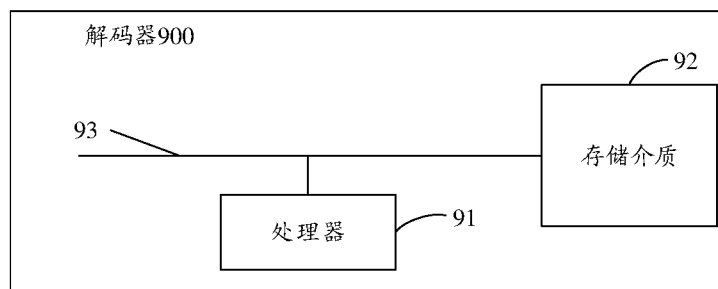


图 9

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2019/109653

| A. CLASSIFICATION OF SUBJECT MATTER | | |
|--|---|---|
| G06T 9/40(2006.01)i; G06T 7/50(2017.01)i; H04N 19/593(2014.01)i | | |
| According to International Patent Classification (IPC) or to both national classification and IPC | | |
| B. FIELDS SEARCHED | | |
| Minimum documentation searched (classification system followed by classification symbols) G06T; H04N | | |
| Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched | | |
| Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) CNPAT, CNKI, WPI, EPODOC, IEEE: 点云, 划分, 压缩, 莫顿码, 迭代, 细节, 层次, 分层, point cloud, partition, divide, compress, Morton, iteration, detail, level, LOD, Level of detail | | |
| C. DOCUMENTS CONSIDERED TO BE RELEVANT | | |
| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
| A | US 2019080483 A1 (APPLE INC.) 14 March 2019 (2019-03-14) description, paragraphs 0377-0380 | 1-14 |
| A | CN 108470374 A (AERIAL PHOTOGRAMMETRY AND REMOTE SENSING CO., LTD.) 31 August 2018 (2018-08-31) entire document | 1-14 |
| A | CN 103077549 A (SOUTH CHINA UNIVERSITY OF TECHNOLOGY) 01 May 2013 (2013-05-01) entire document | 1-14 |
| A | CN 102436654 A (TSINGHUA UNIVERSITY) 02 May 2012 (2012-05-02) entire document | 1-14 |
| A | CN 109257604 A (SHANDONG UNIVERSITY) 22 January 2019 (2019-01-22) entire document | 1-14 |
| A | US 2008238919 A1 (UTAH STATE UNIVERSITY) 02 October 2008 (2008-10-02) entire document | 1-14 |
| A | US 2015113379 A1 (SCHLUMBERGER TECHNOLOGY CORPORATION) 23 April 2015 (2015-04-23) entire document | 1-14 |
| <input checked="" type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> See patent family annex. | | |
| * Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family | | |
| Date of the actual completion of the international search 11 June 2020 | | Date of mailing of the international search report 29 June 2020 |
| Name and mailing address of the ISA/CN China National Intellectual Property Administration (ISA/CN) No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing 100088 China Facsimile No. (86-10)62019451 | | Authorized officer Telephone No. |

| C. DOCUMENTS CONSIDERED TO BE RELEVANT | | |
|--|--|-----------------------|
| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
| A | 杨振发 等 (YANG, Zhenfa et al.). "多分辨率LOD 的海量点云显示技术研究 (Research on Rendering of Massive Point Cloud Based on Multi-resolution LOD)" <i>地理空间信息 (Geospatial Information)</i> , Vol. 14, No. 10, 31 October 2016 (2016-10-31), entire document | 1-14 |
| A | HUBO, Erik et al. "The Quantized kd-Tree: Efficient Ray Tracing of Compressed Point Clouds" <i>IEEE Symposium on Interactive Ray Tracing 2006 18 - 20 September, Salt Lake City, UT, USA</i> , 20 September 2006 (2006-09-20), entire document | 1-14 |

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2019/109653

| Patent document cited in search report | | | Publication date (day/month/year) | Patent family member(s) | | | Publication date (day/month/year) |
|--|------------|----|-----------------------------------|-------------------------|------------|----|-----------------------------------|
| US | 2019080483 | A1 | 14 March 2019 | WO | 2019055772 | A1 | 21 March 2019 |
| CN | 108470374 | A | 31 August 2018 | None | | | |
| CN | 103077549 | A | 01 May 2013 | None | | | |
| CN | 102436654 | A | 02 May 2012 | None | | | |
| CN | 109257604 | A | 22 January 2019 | None | | | |
| US | 2008238919 | A1 | 02 October 2008 | None | | | |
| US | 2015113379 | A1 | 23 April 2015 | GB | 201418829 | D0 | 03 December 2014 |
| | | | | GB | 2523869 | A | 09 September 2015 |

国际检索报告

国际申请号

PCT/CN2019/109653

| <p>A. 主题的分类</p> <p>G06T 9/40(2006.01)i; G06T 7/50(2017.01)i; H04N 19/593(2014.01)i</p> <p>按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类</p> | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|--|--|--|--|---------|---|---|------|---|---|------|---|--|------|---|--|------|---|---|------|---|--|------|---|--|------|
| <p>B. 检索领域</p> <p>检索的最低限度文献(标明分类系统和分类号)</p> <p>G06T; H04N</p> <p>包含在检索领域中的除最低限度文献以外的检索文献</p> <p>在国际检索时查阅的电子数据库(数据库的名称, 和使用的检索词(如使用))</p> <p>CNPAT, CNKI, WPI, EPDOC, IEEE:点云, 划分, 压缩, 莫顿码, 迭代, 细节, 层次, 分层, point cloud, partition, divide, compress, Morton, iteration, detail, level, LOD, Level of detail</p> | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <p>C. 相关文件</p> <table border="1"> <thead> <tr> <th>类型*</th> <th>引用文件, 必要时, 指明相关段落</th> <th>相关的权利要求</th> </tr> </thead> <tbody> <tr> <td>A</td> <td>US 2019080483 A1 (APPLE INC.) 2019年 3月 14日 (2019 - 03 - 14) 说明书第0377-0380段</td> <td>1-14</td> </tr> <tr> <td>A</td> <td>CN 108470374 A (中煤航测遥感集团有限公司) 2018年 8月 31日 (2018 - 08 - 31) 全文</td> <td>1-14</td> </tr> <tr> <td>A</td> <td>CN 103077549 A (华南理工大学) 2013年 5月 1日 (2013 - 05 - 01) 全文</td> <td>1-14</td> </tr> <tr> <td>A</td> <td>CN 102436654 A (清华大学) 2012年 5月 2日 (2012 - 05 - 02) 全文</td> <td>1-14</td> </tr> <tr> <td>A</td> <td>CN 109257604 A (山东大学) 2019年 1月 22日 (2019 - 01 - 22) 全文</td> <td>1-14</td> </tr> <tr> <td>A</td> <td>US 2008238919 A1 (UTAH STATE UNIVERSITY) 2008年 10月 2日 (2008 - 10 - 02) 全文</td> <td>1-14</td> </tr> <tr> <td>A</td> <td>US 2015113379 A1 (SCHLUMBERGER TECHNOLOGY CORPORATION) 2015年 4月 23日 (2015 - 04 - 23) 全文</td> <td>1-14</td> </tr> </tbody> </table> | | | 类型* | 引用文件, 必要时, 指明相关段落 | 相关的权利要求 | A | US 2019080483 A1 (APPLE INC.) 2019年 3月 14日 (2019 - 03 - 14) 说明书第0377-0380段 | 1-14 | A | CN 108470374 A (中煤航测遥感集团有限公司) 2018年 8月 31日 (2018 - 08 - 31) 全文 | 1-14 | A | CN 103077549 A (华南理工大学) 2013年 5月 1日 (2013 - 05 - 01) 全文 | 1-14 | A | CN 102436654 A (清华大学) 2012年 5月 2日 (2012 - 05 - 02) 全文 | 1-14 | A | CN 109257604 A (山东大学) 2019年 1月 22日 (2019 - 01 - 22) 全文 | 1-14 | A | US 2008238919 A1 (UTAH STATE UNIVERSITY) 2008年 10月 2日 (2008 - 10 - 02) 全文 | 1-14 | A | US 2015113379 A1 (SCHLUMBERGER TECHNOLOGY CORPORATION) 2015年 4月 23日 (2015 - 04 - 23) 全文 | 1-14 |
| 类型* | 引用文件, 必要时, 指明相关段落 | 相关的权利要求 | | | | | | | | | | | | | | | | | | | | | | | | |
| A | US 2019080483 A1 (APPLE INC.) 2019年 3月 14日 (2019 - 03 - 14) 说明书第0377-0380段 | 1-14 | | | | | | | | | | | | | | | | | | | | | | | | |
| A | CN 108470374 A (中煤航测遥感集团有限公司) 2018年 8月 31日 (2018 - 08 - 31) 全文 | 1-14 | | | | | | | | | | | | | | | | | | | | | | | | |
| A | CN 103077549 A (华南理工大学) 2013年 5月 1日 (2013 - 05 - 01) 全文 | 1-14 | | | | | | | | | | | | | | | | | | | | | | | | |
| A | CN 102436654 A (清华大学) 2012年 5月 2日 (2012 - 05 - 02) 全文 | 1-14 | | | | | | | | | | | | | | | | | | | | | | | | |
| A | CN 109257604 A (山东大学) 2019年 1月 22日 (2019 - 01 - 22) 全文 | 1-14 | | | | | | | | | | | | | | | | | | | | | | | | |
| A | US 2008238919 A1 (UTAH STATE UNIVERSITY) 2008年 10月 2日 (2008 - 10 - 02) 全文 | 1-14 | | | | | | | | | | | | | | | | | | | | | | | | |
| A | US 2015113379 A1 (SCHLUMBERGER TECHNOLOGY CORPORATION) 2015年 4月 23日 (2015 - 04 - 23) 全文 | 1-14 | | | | | | | | | | | | | | | | | | | | | | | | |
| <p><input checked="" type="checkbox"/> 其余文件在C栏的续页中列出。</p> <p><input checked="" type="checkbox"/> 见同族专利附件。</p> <table border="0"> <tr> <td> <p>* 引用文件的具体类型:</p> <p>“A” 认为不特别相关的表示了现有技术一般状态的文件</p> <p>“E” 在国际申请日的当天或之后公布的在先申请或专利</p> <p>“L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件(如具体说明的)</p> <p>“O” 涉及口头公开、使用、展览或其他方式公开的文件</p> <p>“P” 公布日先于国际申请日但迟于所要求的优先权日的文件</p> </td> <td> <p>“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件</p> <p>“X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性</p> <p>“Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性</p> <p>“&” 同族专利的文件</p> </td> </tr> </table> | | | <p>* 引用文件的具体类型:</p> <p>“A” 认为不特别相关的表示了现有技术一般状态的文件</p> <p>“E” 在国际申请日的当天或之后公布的在先申请或专利</p> <p>“L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件(如具体说明的)</p> <p>“O” 涉及口头公开、使用、展览或其他方式公开的文件</p> <p>“P” 公布日先于国际申请日但迟于所要求的优先权日的文件</p> | <p>“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件</p> <p>“X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性</p> <p>“Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性</p> <p>“&” 同族专利的文件</p> | | | | | | | | | | | | | | | | | | | | | | |
| <p>* 引用文件的具体类型:</p> <p>“A” 认为不特别相关的表示了现有技术一般状态的文件</p> <p>“E” 在国际申请日的当天或之后公布的在先申请或专利</p> <p>“L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件(如具体说明的)</p> <p>“O” 涉及口头公开、使用、展览或其他方式公开的文件</p> <p>“P” 公布日先于国际申请日但迟于所要求的优先权日的文件</p> | <p>“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件</p> <p>“X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性</p> <p>“Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性</p> <p>“&” 同族专利的文件</p> | | | | | | | | | | | | | | | | | | | | | | | | | |
| <p>国际检索实际完成的日期</p> <p>2020年 6月 11日</p> | | <p>国际检索报告邮寄日期</p> <p>2020年 6月 29日</p> | | | | | | | | | | | | | | | | | | | | | | | | |
| <p>ISA/CN的名称和邮寄地址</p> <p>中国国家知识产权局(ISA/CN) 中国北京市海淀区蓟门桥西土城路6号 100088</p> <p>传真号 (86-10)62019451</p> | | <p>授权官员</p> <p>于利娜</p> <p>电话号码 (86-10)53961756</p> | | | | | | | | | | | | | | | | | | | | | | | | |

| C. 相关文件 | | |
|---------|--|---------|
| 类型* | 引用文件, 必要时, 指明相关段落 | 相关的权利要求 |
| A | 杨振发 等. "多分辨率LOD 的海量点云显示技术研究" 地理空间信息, 第14卷, 第10期, 2016年 10月 31日 (2016 - 10 - 31), 全文 | 1-14 |
| A | HUBO, Erik 等. "The Quantized kd-Tree: Efficient Ray Tracing of Compressed Point Clouds" IEEE Symposium on Interactive Ray Tracing 2006 18 - 20 September, Salt Lake City, UT, USA, 2006年 9月 20日 (2006 - 09 - 20), 全文 | 1-14 |

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2019/109653

| 检索报告引用的专利文件 | | | 公布日 (年/月/日) | 同族专利 | | | 公布日 (年/月/日) |
|-------------|------------|----|----------------|------|------------|----|----------------|
| US | 2019080483 | A1 | 2019年 3月 14日 | WO | 2019055772 | A1 | 2019年 3月 21日 |
| CN | 108470374 | A | 2018年 8月 31日 | 无 | | | |
| CN | 103077549 | A | 2013年 5月 1日 | 无 | | | |
| CN | 102436654 | A | 2012年 5月 2日 | 无 | | | |
| CN | 109257604 | A | 2019年 1月 22日 | 无 | | | |
| US | 2008238919 | A1 | 2008年 10月 2日 | 无 | | | |
| US | 2015113379 | A1 | 2015年 4月 23日 | GB | 201418829 | D0 | 2014年 12月 3日 |
| | | | | GB | 2523869 | A | 2015年 9月 9日 |