(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 105103192 B
(45)授权公告日 2018.09.25

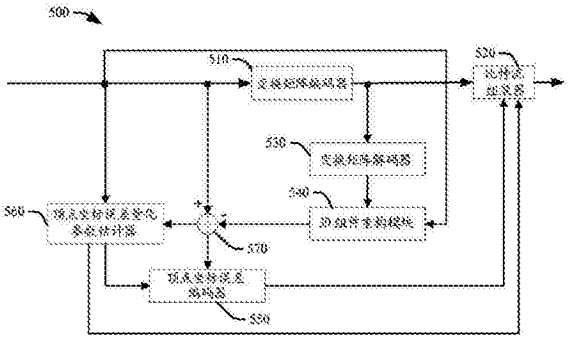
(21)申请号 201380069676.X
(22)申请日 2013.06.18
(65)同一申请的已公布的文献号
申请公布号 CN 105103192 A
(43)申请公布日 2015.11.25
(66)本国优先权数据
PCT/CN2013/070310 2013.01.10 CN
(85)PCT国际申请进入国家阶段日
2015.07.07
(86)PCT国际申请的申请数据
PCT/CN2013/077404 2013.06.18
(87)PCT国际申请的公布数据
WO2014/107944 EN 2014.07.17
(73)专利权人 汤姆逊许可公司
地址 法国伊西莱穆利诺
(72)发明人 江文斐 蔡康颖 田疆
(74)专利代理机构 北京市柳沈律师事务所 11105
代理人 吕晓章
(51)Int.Cl.
G06T 9/00(2006.01)
(56)对比文件
CN 101626509 A,2010.01.13,
CN 102439976 A,2012.05.02,
CN 102460513 A,2012.05.16,
US 2011/0285708 A1,2011.11.24,
Haifeng Yu等.A Segmentation Progressive Mesh Compression Method.《ICALIP》.2012,
审查员 赵玉航
权利要求书2页 说明书10页 附图4页

(54)发明名称

用于顶点误差校正的方法和装置

(57)摘要

可以使用“模式-实例”的表示来建模3D模型，其中实例组件可被表示为模式的变换(例如，旋转、平移和缩放)。量化误差在编码旋转信息时被引入，在实例的不同5顶点处造成不同的顶点坐标误差。为了有效地补偿顶点坐标误差，编码器决定用于补偿顶点坐标误差的量化参数。在比特流中将量化参数表达为量化索引。量化索引、指示量化索引与量化参数之间的映射的量化表，以及顶点坐标误差被10编码到比特流中。量化表可基于统计数据来构建。在解码器处，基于量化参数而解码顶点坐标误差，所述量化参数是根据所接收的量化索引而确定的。



CN 105103192 B

1.一种用于生成表示3D模型的比特流的方法，包括步骤：

访问(330)与实例对应的重构实例；

基于所述实例的顶点与所述重构实例的对应顶点之间的顶点坐标误差，确定(350)量化参数；

响应于所确定的量化参数，确定量化索引；以及

将所述量化索引和所述顶点坐标误差编码(360)成比特流。

2.如权利要求1所述的方法，其中所述确定量化索引的步骤基于指示多个量化索引与多个相应量化参数之间的映射的语法。

3.如权利要求2所述的方法，还包括步骤：

基于统计数据确定所述多个量化索引与所述多个相应量化参数之间的映射。

4.如权利要求3所述的方法，其中比较小的量化索引与所述统计数据中比较频繁的量化参数对应。

5.如权利要求1所述的方法，其中还响应于最大容许误差而确定所述量化参数。

6.如权利要求1所述的方法，其中所述量化参数与量化比特数量和量化步长中的至少一个对应。

7.一种用于解码表示3D模型的比特流的方法，包括步骤：

访问(420)与实例对应的重构实例；

根据所述比特流确定(430)量化索引；

响应于所述量化索引，确定(430)量化参数；

基于所确定的量化参数，解码(440)代表所述实例的顶点与所述重构实例的对应顶点之间的误差的顶点坐标误差；以及

响应于所解码的顶点坐标误差，细化(450)所述重构实例。

8.如权利要求7所述的方法，其中所述确定量化参数的步骤基于指示多个量化索引与多个相应量化参数之间的映射的语法。

9.如权利要求7所述的方法，其中还响应于最大容许误差而解码所述顶点坐标误差。

10.如权利要求7所述的方法，其中所述量化参数与量化比特数量和量化步长中的至少一个对应。

11.一种用于生成表示3D模型的比特流的装置(500)，包括：

3D组件重构模块(540)，用于重构与实例对应的重构实例；

顶点坐标误差量化参数估计器(560)，用于基于所述实例的顶点与所述重构实例的对应顶点之间的顶点坐标误差而确定量化参数，并响应于所确定的量化参数而确定量化索引；以及

顶点坐标误差编码器(580)，用于将所述量化索引和所述顶点坐标误差编码成比特流。

12.如权利要求11所述的装置，其中所述顶点坐标误差量化参数估计器(560)基于指示多个量化索引与多个相应量化参数之间的映射的语法而确定量化索引。

13.如权利要求12所述的装置，其中所述顶点坐标误差量化参数估计器(560)基于统计数据而确定所述多个量化索引与所述多个相应量化参数之间的映射。

14.如权利要求13所述的装置，其中比较小的量化索引与所述统计数据中比较频繁的量化参数对应。

15.如权利要求11所述的装置，其中还响应于最大容许误差而确定所述量化参数。

16.如权利要求11所述的装置，其中所述量化参数与量化比特数量和量化步长中的至少一个对应。

17.一种用于解码表示3D模型的比特流的装置(600)，包括：

3D组件重构模块(630)，用于重构与实例对应的重构实例；

熵解码器(610)，用于根据所述比特流而确定量化索引；

顶点坐标误差解码器(640)，用于确定与所述量化索引对应的量化参数，并基于所确定的量化参数而解码代表所述实例的顶点与所述重构实例的对应顶点之间的误差的顶点坐标误差；以及

加法器(650)，用于响应于所解码的顶点坐标误差而细化所述重构实例。

18.如权利要求17所述的装置，其中所述顶点坐标误差解码器(640)基于指示多个量化索引与多个相应量化参数之间的映射的语法而确定量化参数。

19.如权利要求17所述的装置，其中所述顶点坐标误差解码器(640)还响应于最大容许误差而解码所述顶点坐标误差。

20.如权利要求17所述的装置，其中所述量化参数与量化比特数量和量化步长中的至少一个对应。

21.一种计算机可读存储介质，在其上存储有指令，当该指令被处理器执行使该处理器执行根据权利要求1至10中任一项所述的方法的步骤。

用于顶点误差校正的方法和装置

技术领域

[0001] 本发明涉及一种用于生成代表3D模型的比特流的方法和装置，以及一种用于解码该比特流的方法和装置。

背景技术

[0002] 在实际应用中，许多3D模型由大量的连接组件构成。这些多组件3D模型通常包含许多采用各种变换的重复结构，如图1所示。

[0003] 已知用于多组件3D模型的利用了输入模型中重复结构的压缩算法。3D模型的重复结构以各种位置、方向和缩放因子被发现。于是，3D模型被组织成“模式-实例”(“pattern-instance”)的表示。模式用于标记对应的重复结构的代表性几何图形。属于一个重复结构的组件被标记为对应模式的实例，并由模式ID以及相对于该模式的例如反射、平移、旋转和可能的缩放的变换信息来表示。实例变换信息可被组织成例如反射部分、平移部分、旋转部分和可能的缩放部分。可能存在一些不重复的3D模型的组件，其被称为独特组件。

[0004] 由W.Jiang、K.Cai和J.Tian(PCT/CN2012/074286，代理人案号为No.PA120012，以下简称“Jiang”)共同拥有且名称为“Vertex Correction for Rotated3D Components”的PCT申请，通过引用将其教导明确地并入本文，该PCT申请公开了在编码和解码3D模型时用于顶点误差补偿的方法和装置。

发明内容

[0005] 本原理提供了一种用于生成表示3D模型的比特流的方法，如下所述包括步骤：访问与实例对应的重构实例；基于实例的顶点与重构实例的对应顶点之间的顶点坐标误差而确定量化参数；响应于所确定的量化参数而确定量化索引；以及将量化索引和顶点坐标误差编码成比特流。本原理还提供了一种用于执行这些步骤的装置。

[0006] 本原理提供了一种用于解码表示3D模型的比特流的方法，如下所述包括步骤：访问与实例对应的重构实例；根据比特流而确定量化索引；响应于量化索引而确定量化参数；解码代表实例的顶点与重构实例的对应顶点之间的误差的顶点坐标误差；以及响应于所解码的顶点坐标误差而细化(refine)重构实例。本原理还提供了一种用于执行这些步骤的装置。

[0007] 本原理还提供了一种计算机可读存储介质，在其上存储了用于根据如上所述的方法而生成或解码表示3D模型的比特流的指令。

[0008] 本原理还提供了一种计算机可读存储介质，在其上存储了根据如上所述的方法生成的表示3D模型的比特流。

附图说明

[0009] 图1示出了具有大量的连接组件和重复结构的示例性3D模型；

[0010] 图2A示出了绘出模式的图形例子，图2B示出了绘出对应实例和重构实例的图形例

子；

[0011] 图3是绘出根据本原理的实施例的用于编码3D模型的实例的例子的流程图；

[0012] 图4是绘出根据本原理的实施例的用于解码3D模型的实例的例子的流程图；

[0013] 图5示出了根据本原理的示例性实例编码器；

[0014] 图6示出了根据本原理的示例性实例解码器。

具体实施方式

[0015] 如图1所示，在3D模型中可存在许多重复结构。为了有效地编码3D模型，可将重复结构组织成模式和实例，其中例如使用对应模式的模式ID以及包含诸如平移、旋转和缩放的信息的变换矩阵，可以将实例表示为对应模式的变换。

[0016] 当实例由模式ID和变换矩阵表示时，模式ID和变换矩阵将在压缩实例时被压缩。因此，实例可通过模式ID和解码的变换矩阵来重构，即，实例可被重构为由模式ID进行索引的解码的模式的(根据解码的变换矩阵的)变换。在一个实施例中，在编码变换矩阵时，变换矩阵的旋转部分可例如使用固定数量的比特来量化。由于在量化时引入的损失，解码的旋转部分可能不同于原始的旋转部分。

[0017] 图2A和2B示出了在2D表示下的示例性组件，其中组件210和220是模式，组件250和270(以实线描绘的)是将要被压缩的原始实例，组件260和280(以虚线描绘的)是重构实例。具体地，实例250和270可以分别被表示为模式210和220的变换(例如，旋转和平移)版本。

[0018] 在图2B的例子中，旋转的量化引入大约5°的误差，从而造成原始实例与重构实例之间的差异。正如图2B中可以看到的，虽然(以角度测量的)旋转误差对于实例250和270是类似的，但是由旋转量化而造成的顶点坐标误差(即，原始实例和重构实例之间的顶点偏移，例如图2B中从A到A'、从B到B')在两个实例之间显著变化，其中实例270具有大得多的顶点坐标误差。因此，重构组件的质量可能是不一致的，例如，较大的实例相对于较小的实例具有更低的重构质量。

[0019] 在Jiang中，为了有效地补偿顶点坐标误差，可以为顶点的顶点坐标误差估计上限(upper bound)。根据上限，编码解码器决定顶点的顶点坐标误差是否需要压缩，如果需要压缩，则决定用于压缩顶点坐标误差的量化参数。在编码器和解码器处均可以估计上限，因此不需要明确的信令以指示是否使用顶点坐标误差补偿或指示用于顶点坐标误差的量化参数。

[0020] 本原理还提供用于有效地补偿由旋转量化而造成的顶点坐标误差的方法和装置。为了降低解码器处的计算负荷，在比特流中用信号表达(signal)量化参数。在一个实施例中，通过量化表在比特流中传输与用于量化顶点坐标误差的量化比特数量对应的索引。

[0021] 图3示出了用于编码3D模型的实例的示例性方法300。方法300在步骤305开始。在步骤310，输入3D模型数据，并执行初始化。还可输入或者从输入中推断出附加数据，诸如质量参数、最大容许顶点坐标误差、变换矩阵的平移部分和旋转部分的量化参数。在一个示例性实施例中，初始化步骤可以将重复结构组织成模式和实例，生成实例的变换矩阵，并将模式编码以形成重构模式。对于特定的待编码实例(标记为C)，对应的原始模式、重构模式和变换矩阵分别被标记为P、P'和T。有可能实例可被精确地表示为模式的变换，即，C＝TP。或者，在某些情况下模式的变换是实例的近似，即，C≈TP。

[0022] 在步骤320，编码变换矩阵(T)。在步骤330，所编码的变换矩阵接着被解码为T'，并且例如使用对应的重构模式和解码的变换矩阵来重构实例(C'＝T'P')。

[0023] 在步骤340，将重构实例中的顶点(V_i')与原始实例中的对应顶点(V_i)之间的顶点坐标误差(E_i)计算为例如$E_i＝V_i-V_i$'。为了编码顶点坐标误差，在步骤350估计量化参数。在步骤360，量化并编码顶点坐标误差。此外，在比特流中用信号表达量化参数。在步骤370，检查是否还有顶点需要处理。如果还有顶点将要处理，则控制返回到步骤340。否则，控制转到结束步骤399。

[0024] 为了有效地用信号表达量化参数，可以将量化参数的索引编码到比特流中，而不是将实际的量化参数编码到比特流中。使用量化比特数量作为示例性量化参数，进一步详细讨论量化过程。本原理还可以在使用其它量化参数时运用，所述其它量化参数例如但不限于量化步长。

[0025] 表1

[0026]

量化索引	量化比特数量
0	2
1	4
2	5
3	6

[0027] 表1示出了示例性量化表，其中量化比特数量被映射为量化索引。具体地，

[0028] QBtable[0]＝2；

[0029] QBtable[1]＝4；

[0030] QBtable[2]＝5；

[0031] QBtable[3]＝6。

[0032] 对于特定顶点，我们将MaxErrorAllow标记为由用户提供的质量要求(即，最大容许顶点坐标误差)，并将Error标记为原始实例和重构实例的顶点坐标之间的差异。对于该特定顶点，我们将量化顶点坐标误差所需要的初始比特数量估计为：

[0033] QB＝ceil[log2(Error/MaxErrorAllow)]。 (1)

[0034] 然后，我们在量化表中搜索最接近QB的量化比特数量。例如，当QB＝7时，量化索引3(QBtable[3]＝6)被选择为对应的量化索引，并且QB被设定为6。随后，Error/MaxErrorAllow被量化成QB比特的二进制码。然后，量化索引和二进制码被编码成比特流。正如从表1可以看到的，量化索引的值与对应的量化比特数量相比通常较小，并且可能要求较少的比特进行传输。因此，不是直接发送量化比特数量，而是发送量化索引，这可降低比特率。

[0035] 在解码器处，最大容许误差(MaxErrorAllow)和量化表可以从比特流中导出。对于顶点，从比特流接收量化索引，量化比特数量QB可以根据量化索引和量化表来确定。然后，可以从比特流中读取QB比特作为量化的顶点坐标误差Q_value。顶点坐标误差可以被计算为：

[0036] Error'＝MaxErrorAllow*Q_value。 (2)

[0037] 例如，我们假设量化索引和代表量化的顶点坐标误差的二进制码在比特流中被连

续地编码，并且它们是“1010100111…”。如果量化索引使用2比特编码，则我们得到量化索引“10”＝2。如果使用如表1所示的量化表，则量化比特数量可以被导出为QBtable[2]＝5。随后，我们从比特流中读取5比特“10100”，并确定量化的顶点坐标误差为Q_value＝“10100”＝20。因此，Error’＝MaxErrorAllow*20。

[0038] 如上面所讨论的，量化表用于在比特流中指示量化参数。在一个实施例中，量化表可以由元数据或用户输入来指定。在另一个实例中，量化表可以基于统计数据来构建。

[0039] 例如，我们可以使用来自不同3D模型的大量组件的不同顶点来基于式(1)计算QB的值。在我们得到一大组QB值之后，我们可以选择n个最频繁出现的QB值作为QB表的元素。如果我们将n个最频繁出现的QB值标记为QB_0、QB_1、…、QB_{n-1}，量化表可以如表2所示地示出。当使用固定长度编码来编码量化索引时，我们可构建表以使得$QB_0<QB_1<\cdots<QB_{n-1}$。当使用可变长度编码来编码量化索引时，为了降低用以发送量化索引的数据量，我们可建立量化表以使得$Prob(QB_0)>Prob(QB_1)>\cdots>Prob(QB_{n-1})$。换言之，较大概率的量化比特数量与较小的索引对应，这通常要求较少的比特来编码。

[0040] 表2

[0041]

量化索引	量化比特数量
0	QB_0
1	QB_1
…	…
n-1	QB_{n-1}

[0042] 图4示出了用于解码3D模型的实例的示例性方法400。方法400的输入可包括比特流，例如，使用方法300而生成的比特流。还可以包括附加数据作为输入，所述附加数据例如为与待解码的实例对应的重构模式(P’)。方法400在步骤405开始。在步骤410，执行初始化，例如，从输入比特流中导出变换矩阵的量化参数和质量参数，并且根据质量参数计算最大容许顶点坐标误差。

[0043] 在步骤420，变换矩阵被解码为T’，并且例如使用对应的重构模式和解码的变换矩阵将实例重构为C’(C’＝T’P’)。在步骤430，根据比特流确定量化参数，例如，量化比特数量(QB)。编码的顶点坐标误差在步骤440被解码，例如，从比特流中读取QB比特，并使用式(2)计算顶点坐标误差。在步骤450，所解码的顶点坐标误差(E_i’)用于补偿最初在步骤420重构的实例的对应顶点(V_i’)，例如，如V_i”＝V_i’+E_i’。换言之，细化了重构实例的顶点。方法400在步骤499结束。

[0044] 顶点坐标误差补偿标记可用于指示是否补偿顶点坐标误差。该标记在编码器和解码器处均应当被知晓。当该标记被设定为1时，使用误差补偿。否则，不使用顶点坐标误差补偿。具体地，在使用方法300和400时，如果该标记为0，则不需要方法300中的步骤340-370和步骤430-460。

[0045] 在一个示例性实施例中，可以使用以下伪代码来描述表示3D模型的比特流的解码过程。

[0046] void PB3DMC_Decoder()

[0047] {

[0048] 读取PB3DMC_stream_header;

[0049] 如果(uni_part_bit==0&&repeat_struc_bit==0)

[0050] {

[0051] 使用由3d_model_compr_mode指示的解码器来解码3D模型;

[0052] }

[0053] 否则

[0054] {

[0055] 如果(uni_part_bit==1)

[0056] {

[0057] 使用由3d_model_compr_mode指示的解码器来解码独特部分;

[0058] 基于连接性通过遍历来分离不同的独特组件;

[0059] 解码独特组件的平移向量;

[0060] 通过将所有重构的独特组件平移到它们的位置上,来重构独特部分;

}

如果(repeat_struc_bit == 1)

{

Repeat_Struc_Decoder ( );

}

[0061]

}

}

void Repeat_Struc_Decoder ( )

{

[0062] 解码所有模式;

[0063] 如果(sym_instance_num>0)

[0064] {

[0065] 解码所有对称实例;

[0066] 解码所有拼接信息;

[0067] }

[0068] 使用重新获得的模式、对称实例和拼接信息,来重构所有未连接的重复结构的模式和包括对称结构的独特组件;

[0069] 解码所有未连接的重复结构的模式和包括对称结构的独特组件的平移向量;使用所解码的平移向量,来重构与所有未连接的重复结构的模式和包括对称结构的独特组件对应的那些组件;

[0070] 如果(insta_trans_elem_bit==1)

[0071] {

[0072] Instance_Elementary_Mode_Decoder();

[0073] }

[0074] 否则

[0075] {

[0076] Instance_Grouped_Mode_Decoder();

[0077] }

[0078] }

[0079] void Instance_Elementary_Mode_Decoder()

[0080] {

[0081] 对于(i＝0;i<numInstance;i++)

[0082] {

[0083] 读取elem_insta_QP_translation_flag;

[0084] 读取elem_insta_QP_rotation_flag;

[0085] 如果(elem_insta_QP_translation_flag＝＝1)

[0086] {

[0087] 解码elem_QP_translation;

[0088] }

[0089] 否则

[0090] {

[0091] elem_QP_translation＝QP_Translation;

[0092] }

[0093] 如果(elem_insta_QP_rotation_flag＝＝1)

[0094] {

[0095] 解码elem_QP_rotation;

[0096] }

[0097] 否则

[0098] {

[0099] elem_QP_rotation＝QP_rotation;

[0100] }

[0101] 解码idPattern;

[0102] 读取elem_insta_flip_flag;

[0103] 读取elem_insta_reflection_flag;

[0104] 读取elem_insta_attribute_header;

[0105] 通过其参数是QB_translation的固定长度的解码器来解码实例平移向量;

[0106] 通过其参数是QB_rotation的固定长度的解码器来解码欧拉角;

[0107] 使用所解码的欧拉角来重新获得旋转矩阵;

[0108] 如果(using_scaling_bit＝＝1)

[0109] 解码缩放因子;

[0110] 如果(error_compen_enable_bit＝＝1)

[0111] {

[0112] 读取elem_insta_error_compen_flag;

[0113] 如果(elem_insta_error_compen_flag==1)

[0114] 解码误差补偿数据;}

[0115] 通过由idPattern指示的模式、重新获得的平移向量、重新获得的旋转矩阵、反射变换(如果存在的话)、缩放因子(如果存在的话)以及误差补偿数据(如果存在的话),来重新获得当前实例的几何图形;

[0116] 如果(elem_insta_flip_flag==1)

[0117] 翻转当前实例的所有三角形;

[0118] 解码当前实例的属性数据;

[0119] }

[0120] }

[0121] void Instance_Grouped_Mode_Decoder()

[0122] {

[0123] 读取所有实例的elem_insta_QB_translation_flag;

[0124] 读取所有实例的elem_insta_QB_rotation_flag;

[0125] 解码那些elem_insta_QB_translation_flag等于1的实例的elem_insta_QB_translation;

[0126] 解码那些elem_insta_QB_rotation_flag等于1的实例的elem_insta_QB_rotation;

[0127] 读取compr_insta_patternID_header;

[0128] 解码所有实例的模式ID;

[0129] 读取所有实例的elem_insta_flip_flag;

[0130] 读取所有实例的elem_insta_reflection_flag;

[0131] 读取compr_insta_transl_header;

[0132] 通过基于八叉树分解的解码器来解码所有实例的平移向量;

[0133] 读取compr_insta_rotat_header;

[0134] 解码所有实例的欧拉角;

[0135] 重新获得所有实例的旋转矩阵;

[0136] 如果(use_scaling_bit==1)

[0137] {

[0138] 读取compr_insta_scaling_header;

[0139] 解码所有实例的缩放因子;

[0140] }

[0141] 如果(error_compen_enable_bit==1)

[0142] {

[0143] 读取所有实例的elem_insta_error_compen_flag;

[0144] 对于(i=0;i<numInstance;i++)

[0145] {

[0146] 如果(对应的elem_insta_error_compen_flag为1)

[0147] 解码当前实例的误差压缩数据；

[0148] }

[0149] }

[0150] 通过重新获得的模式、重新获得的平移向量、重新获得的旋转矩阵、反射变换(如果存在的话)、缩放因子(如果存在的话)以及误差补偿数据(如果存在的话)，来重新获得所有实例的几何图形；

[0151] }

[0152] 在表3中，示出了用于量化表的示例性语法和语义，该量化表可以被包含在比特流头中。

[0153] 表3

[0154]

如果 (error_compen_enable_bit = = '1') {
error_compen_QB_table [0..3]
}

[0155] error_compen_enable_bit：该1比特的无符号整数指示在比特流中是否存在某些实例的压缩编码误差补偿数据的数据字段区。0意味着在比特流中不存在实例的压缩编码误差补偿数据的数据字段，1意味着在比特流中存在某些实例的压缩编码误差补偿数据的数据字段。这个比特对应于前面所讨论的方法300和400的顶点坐标误差补偿标记。

[0156] error_compen_QB_table：如果启用误差补偿模式，则每个顶点的补偿值的量化比特数量可在编码器处适应性地确定。编码器将量化比特数量的索引而不是量化比特数量自身传输到比特流中。解码器查找量化表以确定量化比特数量。表中存在4个预定义的量化比特，每个量化比特由一个5比特的无符号整数表示。

[0157] 在表4中，示出了用于顶点误差补偿数据的示例性语法和语义。在该例子中，类compr_elem_insta_error_compen_data包含第i个实例的压缩的顶点误差补偿数据。

[0158] 表4

[0159]

类 compr_elem_insta_error_compen_data {
对于(j=0; j<numofvertex; j++) {
elem_compen_err_QB_id
compr_ver_compen_err_data
}

[0160] elem_compen_err_QB_id：该2比特的无符号整数指示用于error_compen_QB_table中实例的第j个顶点的量化比特数量的索引。

[0161] compr_ver_compen_err_data：该数据字段包含实例的第j个顶点的压缩的补偿值。

[0162] 图5绘出了示例性实例编码器500的方框图。装置500的输入可包括待编码的实例

(C)、对应的模式(P)和重构模式(P')、变换矩阵T、质量参数、以及用于变换矩阵的量化参数。

[0163] 变换矩阵编码器510例如基于用于变换矩阵的不同部分的量化参数来编码变换矩阵T。变换矩阵解码器530将编码器510的输出解码以得到重构的变换矩阵T'。利用对应的重构模式P'和T',可在3D组件重构模块540处将实例重构为C'=T'P'。加法器570求得原始实例和重构实例之间的差异为例如E=C-C'。

[0164] 基于顶点坐标误差E,顶点坐标误差量化参数估计器560例如使用式(1)来估计用于在顶点坐标误差编码器550处量化顶点坐标误差的量化参数。量化参数估计器560可进一步从量化表获得所估计的量化参数的对应索引,可基于量化索引调整量化参数。变换矩阵编码器510和顶点坐标误差编码器550的输出以及量化索引被比特流组装器520组装成比特流,该比特流可以与表示模式或其它组件的其它比特流组合以形成3D模型的整体比特流。

[0165] 图6绘出了示例性实例解码器600的方框图。装置600的输入可包括与实例(C)对应的比特流(例如,根据方法300或由编码器500生成的比特流)以及对应的重构模式(P')。熵解码器610解码比特流以例如得到量化的顶点坐标误差、用于变换矩阵的量化参数、量化表以及用于顶点坐标误差补偿的量化索引。

[0166] 变换矩阵解码器620例如基于用于变换矩阵的不同部分的量化参数来重构变换矩阵T'。利用对应的重构模式P'和T',可在3D组件重构模块630处将实例重构为C'=T'P'。

[0167] 顶点坐标误差解码器640例如基于量化索引和量化表导出量化参数。然后,它可以解码顶点坐标误差。所解码的顶点坐标误差E'用于细化最初在3D组件重构模块630处重构的实例。具体地,加法器650计算解码的坐标误差(E')和最初重构的实例(C')的总和为例如C"=C'+E'。与最初重构的实例C'相比,C"通常提供了原始实例的更精确的表示。

[0168] 在本申请中描述的若干实现和特征可在MPEG 3DGC标准及其扩展的上下文中使用。

[0169] 在此所描述的实现可以例如以方法或过程、装置、软件程序、数据流或信号实现。即使仅在单一形式的实现的上下文中讨论(例如,仅作为方法而讨论),所讨论的特征的实现也可以采用其它形式来实现(例如,装置或程序)。装置可例如以适当的硬件、软件和固件实现。方法可以例如在诸如例如处理器的装置中实现,其中处理器一般是指例如包括计算机、微处理器、集成电路或可编程逻辑器件的处理设备。处理器还包括诸如例如计算机、蜂窝电话、便携式/个人数字助理以及其它促进终端用户之间的信息通信的通信设备。

[0170] 提及本原理的"一个实施例"或"一实施例"或"一个实现"或"一实现"、以及其它变型,意味着结合实施例而描述的特定的特征、结构、特性等被包含在本原理的至少一个实施例中。因此,在整个说明书的各个位置出现的短语"一个实施例"或"一实施例"或"一个实现"或"一实现"以及任何其它变型,不必全都指向同一实施例。

[0171] 另外,本申请或其权利要求可能提及"确定"各种信息。确定信息可包括例如估计信息、计算信息、预测信息或者从存储器取回信息中的一个或多个。

[0172] 另外,本申请或其权利要求可能提及"访问"各种信息。访问信息可包括例如接收信息、取回信息(例如,从存储器取回)、存储信息、处理信息、传输信息、移动信息、复制信息、擦除信息、计算信息、确定信息、预测信息或估计信息中的一个或多个。

[0173] 另外,本申请或其权利要求可能提及"接收"各种信息。与"访问"一样,接收旨在作

为广义术语。接收信息可包括例如访问信息或取回信息(例如,从存储器取回)中的一个或多个。此外,在诸如例如存储信息、处理信息、传输信息、移动信息、复制信息、擦除信息、计算信息、确定信息、预测信息或估计信息的操作期间,“接收”通常以这种或那种方式被涉及。

[0174] 正如对本领域技术人员而言显而易见的,实现可产生各种被格式化以携带例如可被存储或传输的信息的信号。信息可例如包括用于执行方法的指令、或者由所描述的实现之一所产生的数据。例如,可以格式化信号以携带所描述的实施例的比特流。这样的信号可例如被格式化为电磁波(例如,使用频谱的射频部分)或基带信号。格式化可例如包括编码数据流以及使用所编码的数据流来调制载波。信号携带的信息可例如是模拟或数字信息。正如已知的,信号可通过各种不同的有线或无线链路来传输。信号可在处理器可读介质上存储。

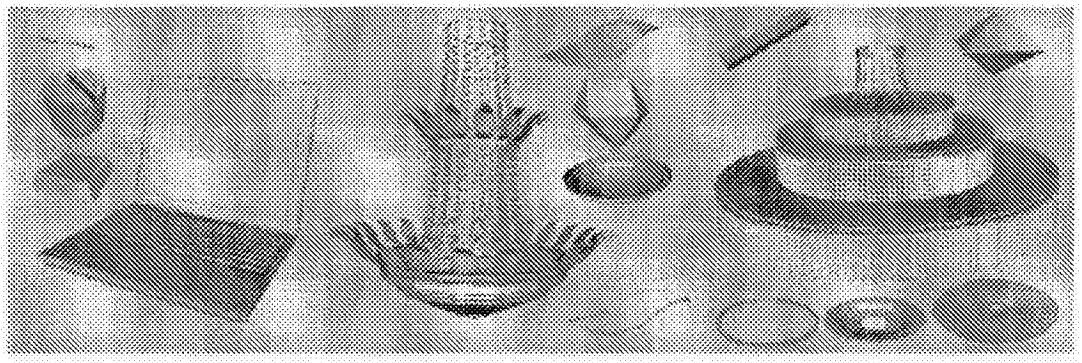


图1

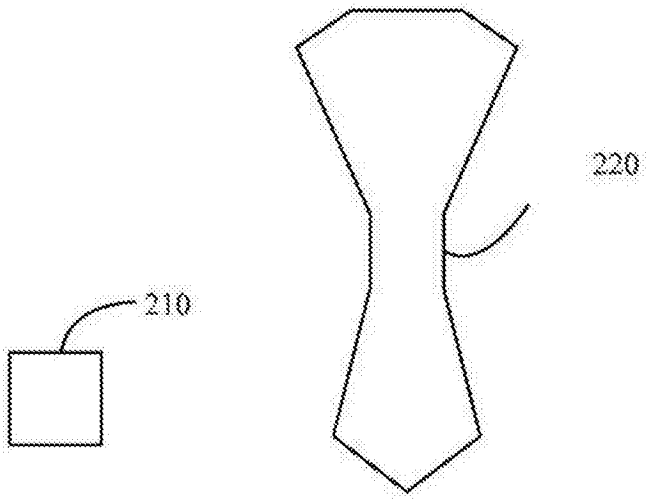


图2A

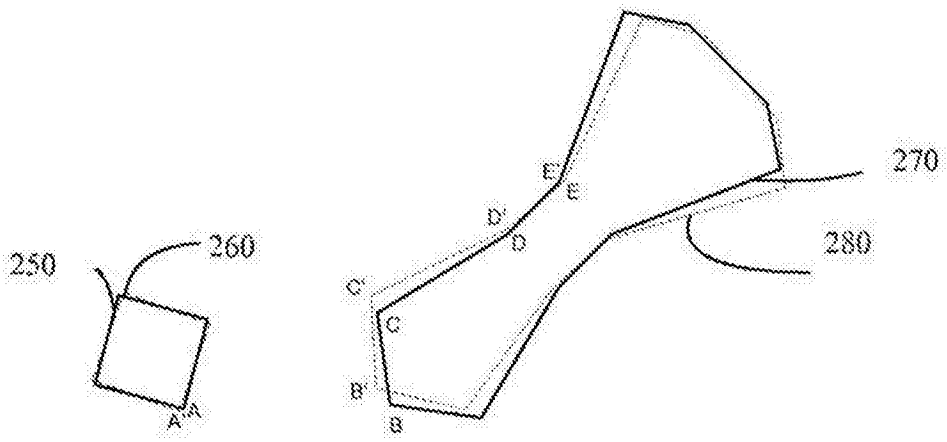


图2B

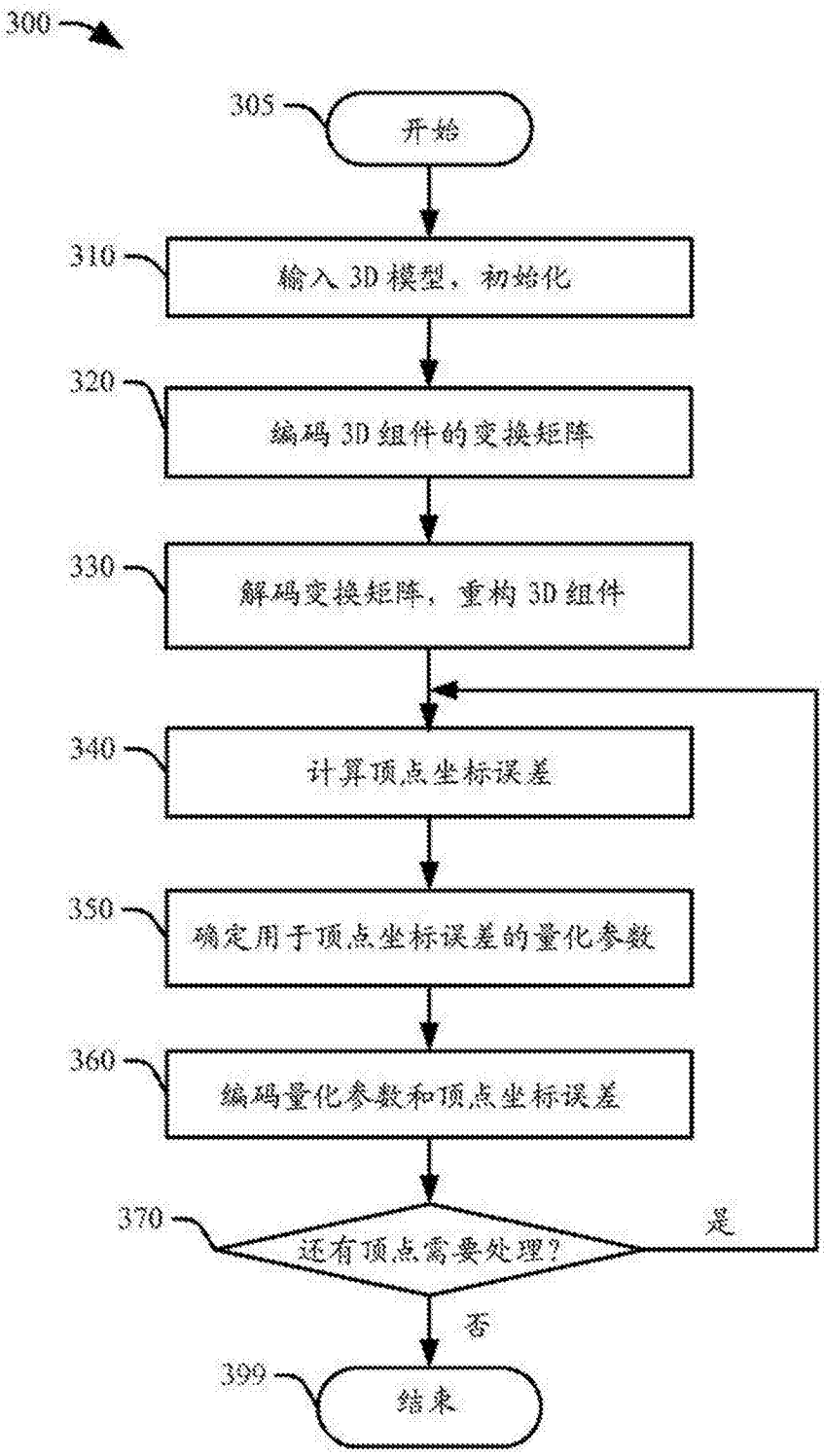

300
305
开始
310
输入 3D 模型，初始化
320
编码 3D 组件的变换矩阵
330
解码变换矩阵，重构 3D 组件
340
计算顶点坐标误差
350
确定用于顶点坐标误差的量化参数
360
编码量化参数和顶点坐标误差
370
还有顶点需要处理？
是
否
399
结束

图3

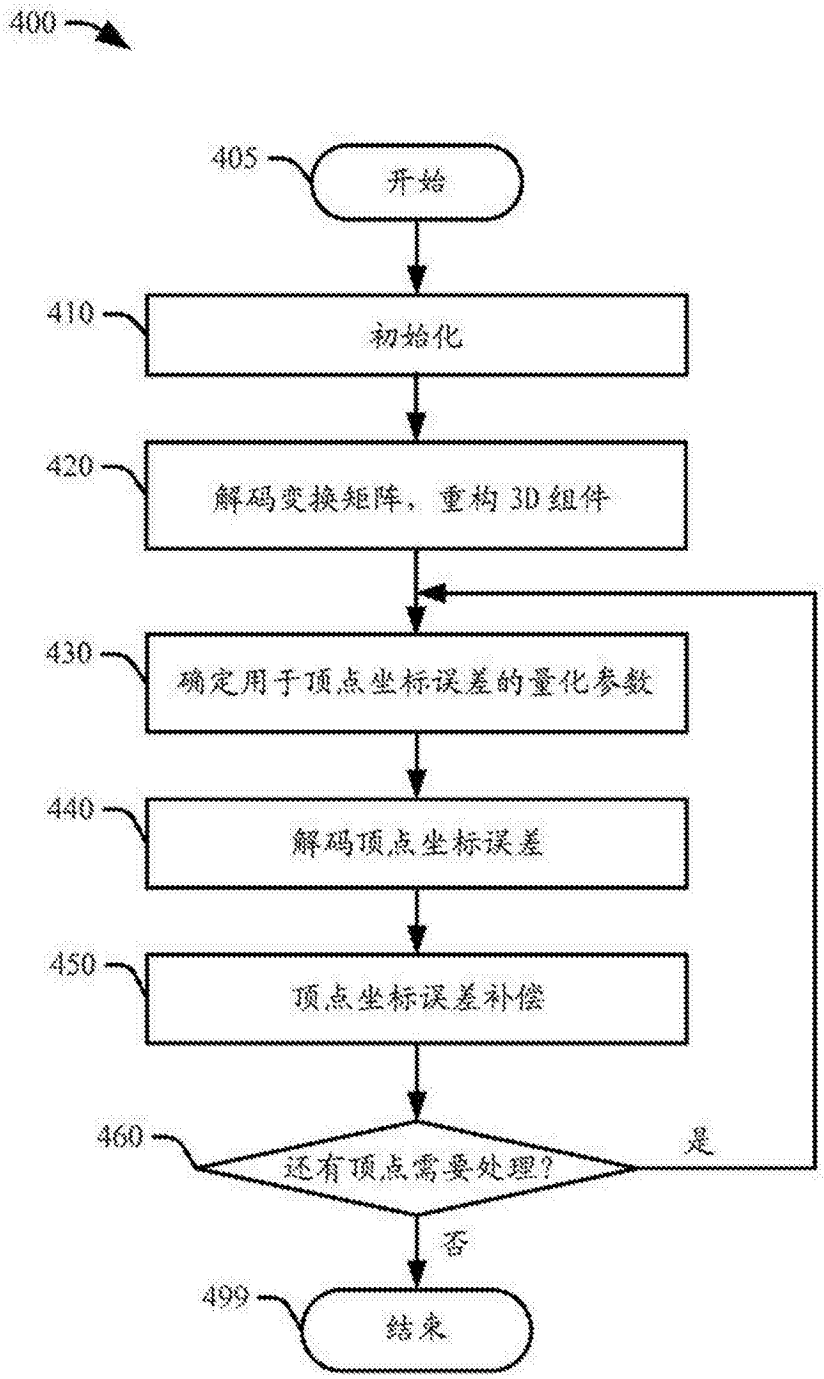

400
405
开始
410
初始化
420
解码变换矩阵，重构 3D 组件
430
确定用于顶点坐标误差的量化参数
440
解码顶点坐标误差
450
顶点坐标误差补偿
460
还有顶点需要处理?
是
否
499
结束

图4

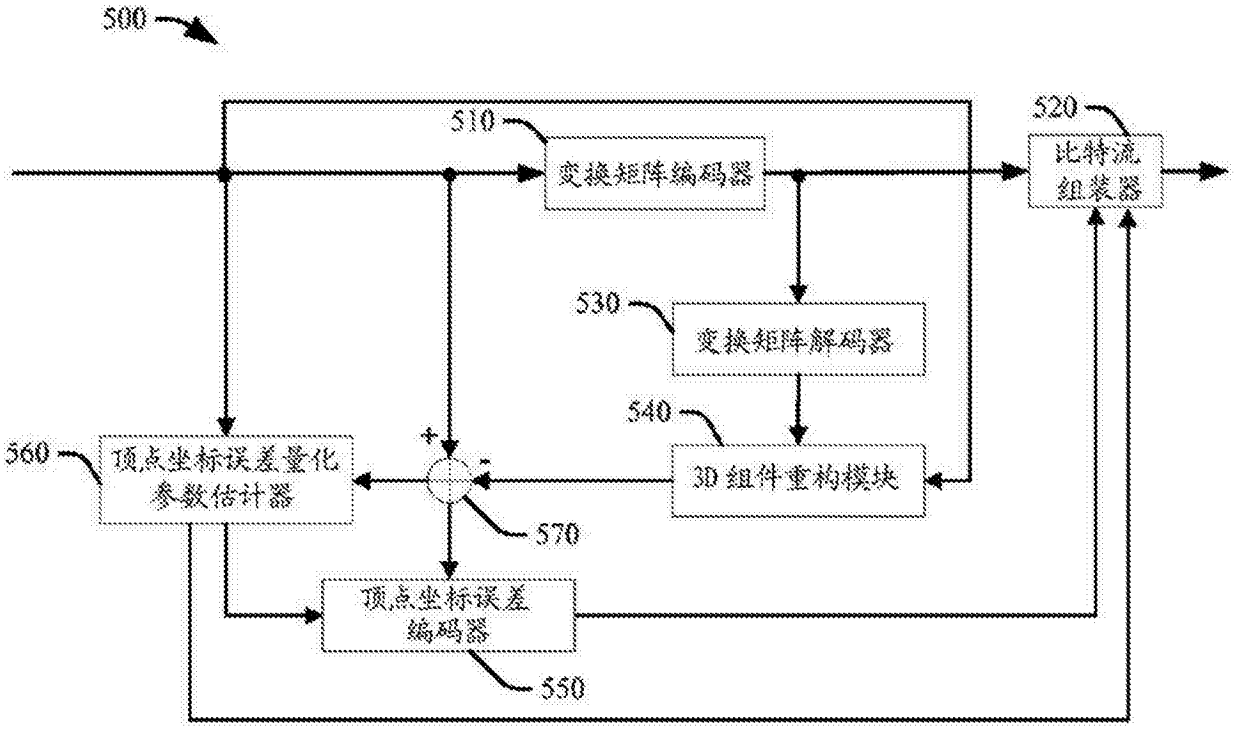

500
510
变换矩阵编码器
520
比特流组装器
530
变换矩阵解码器
540
3D 组件重构模块
560
顶点坐标误差量化参数估计器
+
-
570
顶点坐标误差编码器
550

图5

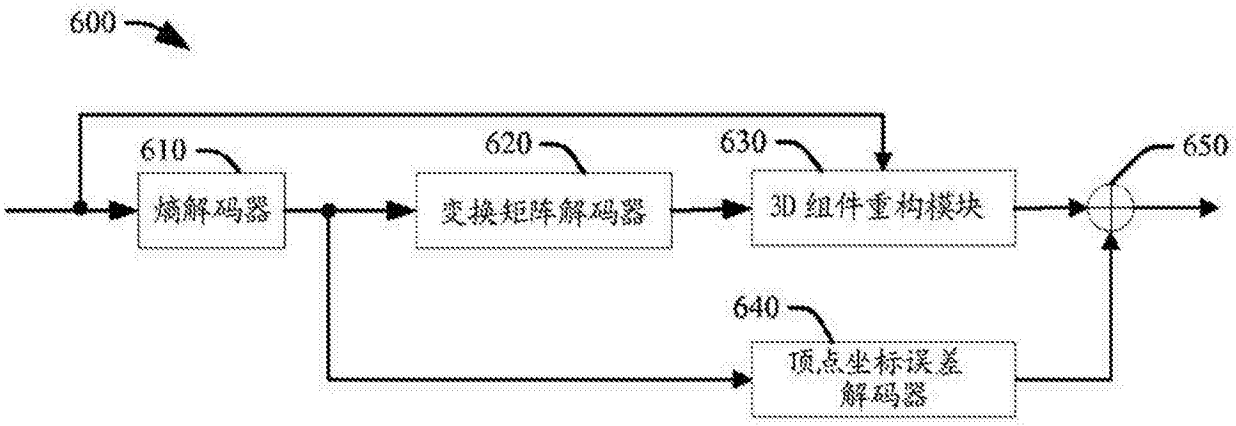

600
610
熵解码器
620
变换矩阵解码器
630
3D 组件重构模块
650
640
顶点坐标误差解码器

图6