



(12) 发明专利申请

(10) 申请公布号 CN 103748615 A

(43) 申请公布日 2014. 04. 23

(21) 申请号 201280035527. 7

(22) 申请日 2012. 07. 17

(66) 本国优先权数据

PCT/CN2011/077277 2011. 07. 18 CN

(85) PCT国际申请进入国家阶段日

2014. 01. 17

(86) PCT国际申请的申请数据

PCT/CN2012/078750 2012. 07. 17

(87) PCT国际申请的公布数据

W02013/010476 EN 2013. 01. 24

(71) 申请人 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 中科专利商标代理有限责任公司 11021

代理人 袁飞

(51) Int. Cl.

G06T 9/00 (2006. 01)

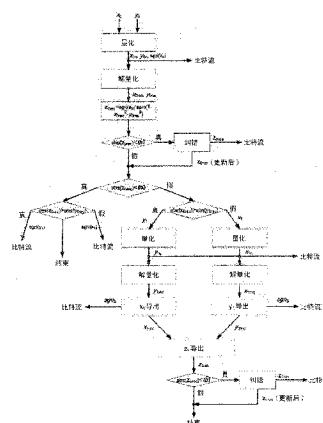
权利要求书3页 说明书11页 附图3页

(54) 发明名称

用于对连接分量的取向向量进行编码的方法和
设备、相应的解码方法和设备、以及携带这种已
编码数据的存储介质

(57) 摘要

本发明是在对连接分量的至少一个取向向量进行编码和解码的领域中做出的。当对向量分量量化以进行编码时,所编码的向量分量的可接受量化偏差有时导致所计算的向量分量的不可接受偏差。因此,提出了一种方法,该方法包括:对该向量的第一分量和该向量的第二分量进行量化和解量化,并对量化后的第一分量和量化后的第二分量以及以信号通知该向量的第三分量的符号的比特进行编码;使用预定长度以及解量化后的第一分量和解量化后的第二分量,来确定该向量的第三分量的近似的所计算的绝对值是否小于第一阈值;并且如果所计算的绝对值小于该第一阈值,则对该第三分量的所计算的绝对值和该第三分量的绝对值之间的残差进行确定、量化和编码。



1. 一种用于对连接分量的取向向量进行编码的方法,所述向量具有预定长度并包括三个分量,所述方法包括:

- 对所述向量的第一分量和所述向量的第二分量进行量化和解量化,并对量化后的第一分量和量化后的第二分量以及以信号通知所述向量的第三分量的符号的比特进行编码,
- 使用所述预定长度以及解量化后的第一分量和解量化后的第二分量,来确定所述向量的第三分量的近似的所计算的绝对值是否小于第一阈值,
- 并且如果所计算的绝对值小于所述第一阈值,则对所述第三分量的所计算的绝对值和所述第三分量的绝对值之间的残差进行确定、量化和编码。

2. 根据权利要求1所述的方法,所述方法还包括:通过执行以下步骤,对所述连接分量的与所述向量垂直的另一取向向量进行编码,所述另一向量具有所述预定长度并包括三个其他分量,

- 使用根据权利要求1编码的数据,确定重构的第三分量,
- 确定所述重构的第三分量小于第二阈值,
- 将解量化后的第一分量的绝对值和解量化后的第二分量的绝对值进行比较,其中,
- 在解量化后的第一分量的绝对值大于解量化后的第二分量的绝对值的情况下,对以信号通知所述其他分量中的第一其他分量的符号的比特进行编码,以及,
- 在解量化后的第一分量的绝对值不大于解量化后的第二分量的绝对值的情况下,对以信号通知所述其他分量中的第二其他分量的符号的比特进行编码,以及,
- 对第三其他分量进行量化和编码。

3. 根据权利要求1所述的方法,所述方法还包括:通过执行以下步骤,对与所述向量垂直的另一向量进行编码,所述另一向量具有所述预定长度并包括三个其他分量,

- 使用根据权利要求1编码的数据,确定重构的第三分量,
- 确定所述重构的第三分量不小于第二阈值,所述第二阈值小于所述第一阈值,
- 使用解量化后的第一分量的绝对值和解量化后的第二分量的绝对值,来对所述其他分量中的第一其他分量和第二其他分量之一进行选择、量化和解量化,
- 使用所述向量的重构、预定长度和解量化后的所选择的其他分量,来计算所述另一向量的第一其他分量和第二其他分量中未选择的其他分量的两个可能值,
- 依赖于所计算的两个可能值中的哪一个更好地近似于所述未选择的其他分量,来设置标记,以及
- 对已量化的所选择的其他分量和所述标记进行编码。

4. 根据权利要求3所述的方法,所述方法还包括:

- 使用所述预定长度、标记和解量化的所选择的其他分量,来确定所述另一向量的第三其他分量的近似的所计算的另一绝对值是否小于所述第一阈值,
- 并且如果所计算的另一绝对值小于所述第一预定阈值,则对所述另一向量的第三其他分量的所计算的绝对值和绝对值之间的另一残差进行确定、量化和编码。

5. 根据权利要求1-4中任一项所述的方法,还包括:将所有已编码的数据存储在非瞬时存储介质上。

6. 一种存储介质,其中,所述存储介质携带根据权利要求5所述的方法存储在所述存储介质上的数据。

7. 一种用于重构连接分量的取向向量的方法,所述向量具有预定长度并包括三个分量,所述方法包括:

- 对以信号通知所述向量的第三分量的符号的比特、所述向量的第一分量和所述向量的第二分量进行解码,并对所述第一分量和第二分量进行解量化,
- 使用所述预定长度以及解量化后的第一分量和解量化后的第二分量,来确定所述向量的第三分量的近似的所计算的绝对值是否小于第一阈值,
- 如果所述所计算的绝对值小于所述第一阈值,则对所述第三分量的所计算的绝对值和所述第三分量的绝对值之间的残差进行确定、解码和解量化,以及
- 使用解码后的数据来重构所述向量的第三分量。

8. 根据权利要求7所述的方法,所述方法还包括:通过执行以下步骤,对所述连接分量的与所述向量垂直的另一取向向量进行重构,所述另一向量具有所述预定长度并包括三个其他分量,

- 确定所述重构的第三分量小于第二阈值,所述第二阈值小于所述第一阈值,
- 将解量化后的第一分量的绝对值和解量化后的第二分量的绝对值进行比较,其中,
- 在解量化后的第一分量的绝对值大于解量化后的第二分量的绝对值的情况下,对以信号通知所述其他分量中的第一其他分量的符号的比特进行编码,以及,
- 在解量化后的第一分量的绝对值不大于解量化后的第二分量的绝对值的情况下,对以信号通知所述其他分量中的第二其他分量的符号的比特进行编码,
- 以及,对所述向量的第三其他分量进行解码和解量化。

9. 根据权利要求7所述的方法,所述方法还包括:通过执行以下步骤,对所述连接分量的与所述向量垂直的另一取向向量进行重构,所述另一向量具有所述预定长度并包括三个其他分量,

- 确定所述重构的第三分量不小于第二阈值,
- 对标记和所述其他分量之一进行解码,并对所述其他分量之一进行解量化,
- 使用解量化后的第一分量的绝对值和解量化后的第二分量的绝对值,来确定所述其他分量之一是所述另一向量的第一其他分量还是第二其他分量,
- 使用所述向量的重构、预定长度、标记和所述其他分量中解量化后的其他分量,来计算所述另一向量的一个其他分量,所述其他分量之一被确定为不是该其他分量,以及
- 使用所述预定长度、解量化后的一个其他分量以及所计算的其他分量,来确定第三其他分量的近似。

10. 根据权利要求9所述的方法,所述方法还包括:

- 确定所述第三其他分量的所确定的近似的绝对值小于所述第一阈值,
- 对另一残差进行解码和解量化,以及
- 使用解量化的另一残差,更新所确定的近似。

11. 一种设备,包括:处理器,用于执行权利要求1-5之一或权利要求7-10之一所述的方法。

12. 一种设备,包括:重复结构发现模块,确定与连接分量相关联的取向向量,其中,所述连接分量对应于模式的实例,所述模式对应于在3D模型中发生的重复结构;以及编码器,执行权利要求1-5之一所述的方法,以对所述取向向量进行编码。

13. 一种设备,包括:解码器,对根据权利要求7-10之一所述的方法而编码的取向向量进行解码,其中,所述连接分量对应于模式的实例,所述模式对应于在3D模型中发生的重复结构;以及模型重构模块,产生包括所述连接分量的所述3D模型。

14. 根据权利要求1-5中任一项所述的方法,其中,所述连接分量对应于模式的实例,所述模式对应于在3D模型中发生的重复结构,并且所述方法还包括以下步骤:确定与所述连接分量相关联的所述取向向量。

15. 根据权利要求7-10中任一项所述的方法,其中,所述连接分量对应于模式的实例,所述模式对应于在3D模型中发生的重复结构,并且所述方法还包括以下步骤:产生包括所述连接分量的所述3D模型。

用于对连接分量的取向向量进行编码的方法和设备、相应的解码方法和设备、以及携带这种已编码数据的存储介质

技术领域

[0001] 本发明是在对向量分量进行编码的领域中做出的。具体而言,本发明涉及对连接分量的取向向量进行编码,所述向量具有预定义长度并且包括三个分量。

背景技术

[0002] 连接分量的取向向量用作将分量的模板旋转变换为分量的实例,并以很多不同方式用在视听内容的处理中。例如,当对听觉对象进行模型化时,该对象可以代表声源。当对视觉对象进行模型化时,该对象可以代表刚性的身体。

[0003] 当对视觉对象进行模型化时,特别是在用于例如 CAD 系统、3G 游戏、3D 电视或 3D 电影等的三维 (3D) 中,常遇到重复结构。可以通过对这样的重复结构 (例如,发生若干次的对象或对象部分) 的模板进行一次编码,并针对该结构的每个实例,对允许将模板变换为实例的数据进行编码来对重复结构进行压缩编码。模板还被称为模式,并可以由聚类 (clustering) 产生。

[0004] 最一般而言,这种变换是仿射变换,仿射变换可以被分解为旋转、缩放、修剪和 / 或移位 (displacement)。旋转、缩放、修剪是可交换的线性变换,即他们应用的顺序不影响整体变换结果,并且可以对允许每个线性变换的数据独立地进行编码。

[0005] 在线性变换中,在普通三维空间中的旋转可以进一步分解为绕三个不同轴的旋转,即,3D 中旋转的旋转数据一般具有三个自由度。

[0006] 即,仿射变换的旋转变换部分可以由指定相互正交的一对归一化取向向量的参数来表示。由于垂直性约束和归一化约束,这对取向向量具有三个自由度,即,必须确定三个参数以允许两个向量的明确确定,这是因为可以使用所编码的参数和约束计算其他参数。

[0007] 在一个特定旋转是零或 Π (还被记为 Π , 对应于 180 度) 的所谓万向节死锁的情况下,进动 (precession) 旋转和固有旋转绕相同轴 (即在相同面内) 发生。进动旋转和固有旋转是可交换的,并可以由累积旋转表示。因此,在这种情况下,自由度损失,并且旋转由两个参数唯一地指定。

[0008] M.Deering: " Geometry Compression", Proceedings of ACM SIGGRAPH, 1995, pp. 13-20 针对取向向量的编码,提出了法线球体,单位法线的端点位于所述法线球体上,并且所述法线球体被分割为八个八分圆 (每个八分圆被进一步分割为六个六分圆)。

[0009] 针对连接分量的取向轴具有一些占优势的取向的情况,Deok-Soo Kim, Youngsong Cho 和 Hyun Kim: " Normal Compression Based on Clustering and Relative Indexing", Pacific Conference on Computer Graphics and Applications IEEE, 2002 提出了基于具有固定数量的簇的 k 均值聚类的方案。

[0010] Kan gying Cai, Wencheng Wang, Zhibo Chen, Quqing Chen, Jun Teng: " Exploiting repeated patterns for efficient compression of massive models", Proceedings of the 8th International Conference on Virtual Reality

Continuum and its Applications in Industry (VRCAI 2009), 145-150, 2009 中描述了利用重复模式进行压缩的另一方案。

发明内容

[0011] 当对向量分量量化以进行编码时,发明人发现所编码的分量的量化所引入的可接受偏差有时导致所计算的分量的不可接受的偏差。发明人认识到这是由于计算中所包括的划分步骤,在除数接近零的情况下,划分导致除数中出现的误差的不可接受的传播。

[0012] 因此,提出了根据权利要求 1 所述的方法,用于对连接分量的取向向量进行编码,所述向量具有预定长度并包括三个分量。

[0013] 根据本发明的方法有益地用在连接分量的编码/解码中,所述连接分量对应于在 3D 模型中发生的重复模式的实例。

[0014] 所述方法包括对所述向量的第一分量和所述向量的第二分量进行量化和解量化,并对量化后的第一分量和量化后的第二分量以及以信号通知所述向量的第三分量的符号的比特进行编码;使用所述预定长度以及解量化后的第一分量和解量化后的第二分量,来确定所述向量的第三分量的近似的所计算的绝对值是否小于第一阈值;并且如果所计算的绝对值小于所述第一阈值,则对所述第三分量的所计算的绝对值和所述第三分量的绝对值之间的残差进行确定、量化和编码。

[0015] 在实施例中,所述方法还包括,通过执行以下步骤,对所述连接分量的与所述向量垂直的另一取向向量进行编码,所述另一向量具有所述预定长度并包括三个其他分量:使用根据权利要求 1 进行编码的数据,确定重构的第三分量,确定所述重构的第三分量小于第二阈值,比较解量化后的第一分量的绝对值和解量化后的第二分量的绝对值,其中,在解量化后的第一分量的绝对值大于解量化后的第二分量的绝对值的情况下,对以信号通知所述其他分量中第一其他分量的符号的比特进行编码,以及,在解量化后的第一分量的绝对值不大于解量化后的第二分量的绝对值的情况下,对以信号通知所述其他分量中第二其他分量的符号的比特进行编码,并对第三其他分量进行量化和编码。

[0016] 在不同实施例中,所述方法还包括,通过执行以下步骤,对与所述向量垂直的另一向量进行编码,所述另一向量具有预定长度并包括三个其他分量:使用根据权利要求 1 进行编码的数据确定重构的第三分量,确定所述重构的第三分量不小于第二阈值(所述第二阈值小于所述第一阈值),使用解量化后的第一分量的绝对值和解量化后的第二分量的绝对值,来对所述其他分量中的第一其他分量和第二其他分量之一进行选择、量化和解量化,使用所述向量的重构、预定长度和解量化后的所选择的其他分量,用于计算所述另一向量的第一其他分量和第二其他分量中未选择的其他分量的两个可能值,依赖于所计算的两个可能值中的哪一个更好地近似于所述未选择的其他分量进行近似,来设置标记,并对所选择的其他分量和所述标记进行编码。

[0017] 在所述不同实施例中,所述方法还包括:使用所述预定长度、所述标记和所述解量化的所选择的其他分量,来确定所述另一向量的第三个其他分量的近似的所计算的另一绝对值是否小于所述第一阈值,并且如果所计算的另一绝对值小于所述第一预定阈值,则对所述另一向量的第三其他分量的所计算的绝对值和绝对值之间的另一残差进行确定、量化和编码。

[0018] 所述方法可以但不需要包括：将所有已编码数据存储在非瞬时存储介质上。

[0019] 还提出了一种存储介质，所述存储介质携带根据所提出的方法或所述方法的实施例之一存储在其上的数据。

[0020] 并提出了根据权利要求 7 的用于重构连接分量的取向向量的方法，所述向量具有预定长度并包括三个分量。

[0021] 所述重构方法包括：对以信号通知所述向量的第三分量的符号的比特、所述向量的第一分量和所述向量的第二分量进行解码，并对所述第一分量和第二分量进行解量化；使用所述预定长度以及所述解量化后的第一分量和解量化后的第二分量，来确定所述向量的第三分量的近似的所计算的绝对值是否小于第一阈值，并且如果所计算的绝对值小于所述第一阈值，则对所述第三分量的所计算的绝对值和所述第三分量的绝对值之间的残差进行确定、解码和解量化，并使用所述已解码数据来重构所述向量的第三分量。

[0022] 在实施例中，所述重构方法还包括，通过执行以下步骤，对所述连接分量的与所述向量垂直的另一取向向量进行解码，所述另一向量具有所述预定长度并包括三个其他分量：确定所述重构的第三分量小于第二阈值（所述第二阈值小于所述第一阈值），将解量化后的第一分量的绝对值和解量化后的第二分量的绝对值进行比较，其中，在解量化后的第一分量的绝对值大于解量化后的第二分量的绝对值的情况下，对以信号通知所述其他分量中第一其他分量的符号的比特进行编码，以及，在解量化后的第一分量的绝对值不大于解量化后的第二分量的绝对值的情况下，对以信号通知所述其他分量中第二其他分量的符号的比特进行编码，并对所述向量的第三其他分量进行解码和解量化。

[0023] 在不同实施例中，所述重构方法还包括，通过执行以下步骤，对所述连接分量的与所述向量正交的另一取向向量进行解码，所述另一向量具有所述预定长度并包括三个其他分量：确定所述重构的第三分量不小于第二阈值，对标记和所述其他分量之一进行解码，并对所述其他分量之一进行解量化，使用解量化后的第一分量的绝对值和解量化后的第二分量的绝对值，来确定所述其他分量之一是所述另一向量的第一其他分量还是第二其他分量，使用所述向量的重构、所述预定长度、所述标记和所述其他分量中解量化后的其他分量，来计算所述另一向量的一个其他分量，所述其他分量之一被确定为不是该其他分量，并使用所述预定长度、所述解量化后的一个其他分量以及所计算其他分量，来确定第三其他分量的近似。

[0024] 在所述不同实施例中，所述重构方法还可以包括：确定所述第三其他分量的所确定的近似的绝对值小于所述第一阈值，对另一残差进行解码和解量化，并使用解量化的另一残差更新所确定的近似。

[0025] 此外，提出了一种设备，包括用于执行所提出方法之一的处理器。此外，本发明提供了一种设备，包括用于对连接分量的取向向量进行编码 / 解码的编码器或解码器，其中，所述连接分量对应于 3D 模型中重复模式的实例。

[0026] 在从属权利要求中指定其他有益实施例的特征。

附图说明

[0027] 在附图中示出了并在以下描述中更详细地解释了本发明的示例性实施例。解释示例性实施例仅用于阐述本发明，而不适用于限制权利要求中所限定的本发明的公开或范围。

- [0028] 在附图中：
- [0029] 图 1 示例性地示出了编码方法的实施例的流程图；
- [0030] 图 2 示例性地示了解码方法的实施例的流程图；
- [0031] 图 3 示出了根据本原理的 3D 模型的示例性编码器；以及
- [0032] 图 4 示出了根据本原理的 3D 模型的示例性解码器。

具体实施方式

[0033] 本发明可以在包括相应适配的处理设备在内的任意电子设备上实现。可以实现本发明的示例性设备的非穷尽性列表包括：电视、移动电话、个人计算机、数字静态照相机、数字视频照相机、mp3 播放器、导航系统或汽车音频系统。

[0034] 本发明可以用于对预定长度的向量进行编码，所述预定长度与可以使用已编码向量的任意目的无关。

[0035] 以下所描述的示例性实施例涉及视觉对象的模型化，其中已编码向量是取向向量，但本发明不限于此。

[0036] 在表 1 中，示出了比特流格式的示例，可以通过所述比特流格式表达根据本发明实施例编码的一对垂直的归一化向量：

[0037] `compr_ith_insta_orient_cartesian` 类

[0038]

class compr_ith_insta_orient_cartesian {	比特数	描述符
compr_ith_insta_orient_x0	bit_num_orient_cartesian()	f(bit_num_orient_cartesian())
compr_ith_insta_orient_y0	bit_num_orient_cartesian()	f(bit_num_orient_cartesian())
ith_insta_orient_z0_sgn	1	
if (compute_z0() < threshold) {		
compr_ith_insta_orient_z0_res	bit_num_orient_res_cartesian()	f(bit_num_orient_res_cartesian())
}		
if (compute_z0() ≈ 0) {		
compr_ith_insta_orient_z1	bit_num_orient_cartesian()	f(bit_num_orient_cartesian())
if (abs(ith_x0) < abs(ith_y0)) {		
ith_insta_orient_x1_sgn	1	
}else {		
ith_insta_orient_y1_sgn	1	
}		
}else {		
if (abs(ith_x0) < abs(ith_y0)) {		
compr_ith_insta_orient_x1	bit_num_orient_cartesian()	f(bit_num_orient_cartesian())
}		
}else {		
compr_ith_insta_orient_y1	bit_num_orient_cartesian()	f(bit_num_orient_cartesian())
}		
ith_insta_orient_delta_sgn	1	
if (compute_z1() < threshold) {		
compr_ith_insta_orient_z1_res	bit_num_orient_res_cartesian()	f(bit_num_orient_res_cartesian())
}		
}		
}		

[0039] 表 1

[0040] 笛卡尔模式中第 i 个实例的取向由 2 个正交轴 (x0, y0, z0) 和 (x1, y1, z1) 表示。

[0041] compr_ith_insta_orient_x0: 包含第 i 个实例的取向的压缩的 x0。

[0042] compr_ith_insta_orient_y0: 包含第 i 个实例的取向的压缩的 y0。

[0043] compr_ith_insta_orient_z0_sgn: 指示使用 x0 和 y0 计算 z0 所需的 z0 的符号的 1 比特无符号整数。对于“-”取 0, 而对于“+”取 1。

[0044] compr_ith_insta_orient_z0_res: 包含通过 (z0-computer_z0()) 计算的 z0 的压缩残差。

[0045] compr_ith_insta_orient_z1: 包含第 i 个实例的取向的压缩的 z1。

[0046] ith_insta_orient_x1_sgn: 指示使用 x0 和 y0 计算 x1 所需的 x1 的符号的 1 比特无符号整数。对于“-”取 0, 而对于“+”取 1。

[0047] ith_insta_orient_y1_sgn: 指示使用 x0 和 y0 计算 y1 所需的 y1 的符号的 1 比特无符号整数。对于“-”取 0, 而对于“+”取 1。

[0048] compr_ith_insta_orient_x1: 包含第 i 个实例的取向的压缩的 x1。

[0049] compr_ith_insta_orient_y1: 包含第 i 个实例的取向的压缩的 y1。

[0050] ith_insta_orient_delta_sgn: 指示使用 x0、y0、z0 和 y1 或 x1 计算 x1 或 y1 所

需的符号的 1 比特无符号整数。对于“-”取 0,而对于“+”取 1。

[0051] `compr_ith_insta_orient_z1_res`:包含通过 `(z1-computer_z1())` 计算的 `z1` 的压缩残差。

[0052] `threshold`:在压缩领域中广泛接受的阈值。

[0053] `compute_z0()`:使用 `x0`、`y0` 和 `z0` 符号计算第 `i` 个实例的 `z0`。

[0054] `bit_num_orient_cartesian()`:基于 QP 计算笛卡尔坐标系中每个取向值的比特数。

[0055] `bit_num_orient_res_cartesian()`:基于 QP 计算笛卡尔坐标系中每个取向残差值的比特数。

[0056] `compute_z1()`:使用 `x0`、`y0`、`z0`、`x1` 和 `y1` 计算第 `i` 个实例的 `z1`。

[0057] 在表 2 中,示出了比特流格式的另一示例:

[0058] `compr_ith_insta_orient_spherical` 类

[0059]

class <code>compr_ith_insta_orient_spherical</code>	比特数	描述符
<code>{</code>		
<code> compr_ith_insta_orient_alpha</code>	<code>bit_num_orient_alpha()</code>	<code>f(bit_num_orient_alpha())</code>
<code> compr_ith_insta_orient_beta</code>	<code>bit_num_orient_beta()</code>	<code>f(bit_num_orient_beta())</code>
<code> compr_ith_insta_orient_gamma</code>	<code>bit_num_orient_gamma()</code>	<code>f(bit_num_orient_gamma())</code>
<code> if (need_correction()) {</code>		
<code> compr_ith_insta_orient_res</code>	<code>6*bit_num_orient_res_cartesian()</code>	<code>f(6*bit_num_orient_res_cartesian())</code>
<code> }</code>		
<code>}</code>		

[0060] 表 2

[0061] 球模式中第 `i` 个实例的取向是通过 3 个角度 (`alpha`, `beta` 和 `gamma`) 表示的。

[0062] `compr_ith_insta_orient_alpha`:包含第 `i` 个实例的取向的压缩的 `alpha`。

[0063] `compr_ith_insta_orient_beta`:包含第 `i` 个实例的取向的压缩的 `beta`。

[0064] `compr_ith_insta_orient_gamma`:包含第 `i` 个实例的取向的压缩的 `gamma`。

[0065] `compr_ith_insta_orient_res`:包含第 `i` 个实例的取向的笛卡尔坐标系中的压缩残差。

[0066] `bit_num_orient_alpha()`:基于 QP 计算每个 `alpha` 值的比特数。

[0067] `bit_num_orient_beta()`:基于 QP 计算每个 `beta` 值的比特数。

[0068] `bit_num_orient_gamma()`:基于 QP 计算每个 `gamma` 值的比特数。

[0069] `need_correction()`:检查取向,如果其处于很可能导致较大误差的边界条件中,则返回真;否则,返回假。

[0070] 归一化向量的编码的必然性出现的示例是表示 3D 连接分量的取向的表示。一般而言,对连接分量的取向轴(在笛卡尔坐标中或在球坐标中)的取向进行编码。因为 3D 连接分量的三个取向轴相互正交,可以通过计算前两个轴的叉积来获得第三轴。将三个轴记为 $d_0(x_0, y_0, z_0)$, $d_1(x_1, y_1, z_1)$ 和 $d_2(x_2, y_2, z_2)$,我们有 $d_0 \cdot d_1 = 0$ 并且 $d_2 = d_0 \times d_1$,则取向轴的编码方法可以包括:

[0071] 1) 压缩 `x0` 和 `y0`。

[0072] 2) `z0` 的编码符号。

[0073] 3) 压缩 x_1 。

[0074] 接着,可以使用下式计算 y_1 、 z_0 和 z_1 :

[0075] $z_1 = \sqrt{1 - x_1^2 - y_1^2}$,

[0076] $z_1 = \sqrt{1 - x_1^2 - y_1^2}$ 以及

[0077] $0 = x_0 * x_1 + y_0 * y_1 + z_0 * z_1$ 。

[0078] 可以使用 $d_2 = d_0 * x_{d_1}$ 确定取向轴 d_2 。

[0079] 但因为在编码之前对 x_0 、 y_0 和 x_1 的浮点值进行量化,所以引入偏差,由于计算中的误差传播,所述偏差导致所计算的分量中的更大误差。

[0080] 例如,在利用 12 比特量化对以下取向进行压缩以后:

[0081] $d_0(-0.984644, -0.174418, -0.00737644)$ 和

[0082] $d_1(-0.121407, 0.714518, -0.689003)$

[0083] 重构将导致

[0084] $d_0(-0.984462, -0.174202, -0.0220894)$ 和

[0085] $d_1(-0.121595, 0.767049, -0.629961)$ 。

[0086] x_0 、 y_0 和 x_1 的量化误差是可接受的:0.000182,0.000216 和 0.000188。然而,所计算的 z_0 、 y_1 和 z_1 值的误差是 0.01471296,0.052531 和 0.059042,这是完全不可接受的。

[0087] 以上问题的首要原因是 z_0 的计算误差。如果 $1 - x_0^2 - y_0^2$ 小,则 z_0 小, x 或 y 上的微小误差发展为 z_0 的较大误差,这是因为 z_0 是 $1 - x_0^2 - y_0^2$ 的平方根。因此,本发明还提出了在 z_0 小的情况下(即,其绝对值在第一阈值以下)对纠正进行编码。

[0088] 此外,重构 z_1 包括除以 z_0 。在 z_0 接近零的情况下,此除法也导致不可接受的误差传播。类似地,在对 x_1 进行编码的情况下,重构 y_1 包括除以 $(1 - x_0^2)$ 。备选地,在对 y_1 进行编码的情况下,可以使用除以 $(1 - y_0^2)$ 来重构 x_1 。

[0089] 所发明的压缩方法确保向量的重构与向量偏差不超过最大偏差值。

[0090] 如上所述,可以计算取向表示中的一些值,而不是对其进行编码。不幸地,这可能导致不可接受的误差,这是因为:由于量化,计算自身所用的已编码值是不精确的。本发明解决了此问题,并提出了最小化该计算误差的压缩方法,这是由于其包括:对相当小的那些已计算分量的残差数据进行编码。

[0091] 在对垂直向量对进行编码的特定实施例中,所发明的编码方法包括:对向量对之一的第一量化浮点分量值和第二量化浮点分量值以及向量对中另一个的第一量化浮点分量值或第二量化浮点分量值进行编码。在所述特定实施例中,还对两个符号比特或标记比特(即,对单个比特)进行编码,以表示 3D 分量的取向。为了最小化重构误差,基于以下几点设计所述特定实施例的编码方案:

[0092] 1. 使计算中的分母尽可能的大。

[0093] 2. 针对分母极小或为零的情况的特殊处理。

[0094] 3. 自动地标识可能导致不可接受误差的情况,并针对这些情况发送残差数据。

[0095] 图 1 示例性地示出了根据所述特定实施例的编码处理。

[0096] 总是对一个向量的第一分量 x_0 进行量化和编码。至少只要第一量化向量分量的解量化值 x_{0r} 不等于 1,就还对符号比特进行编码,该符号比特以信号通知该一个向量的第三分量 z_0 的符号,并且还对该一个向量的第二分量 y_0 进行量化和编码。

[0097] z0 导出模块使用该一个向量的预定长度和已编码数据的重构, 计算该一个向量的第三分量的近似 z0a。即, 至少只要第一量化向量分量的解量化值 x0r 的绝对值不等以 1, 符号比特以及第一量化向量分量和第二量化向量分量的解量化值 x0r、y0r 就用于确定 z0a。在第一量化向量分量的解量化值 x0r 的绝对值等于 1 的情况下, z0a 可以确定为零。

[0098] 在所计算的 z0a 的值非常小 (即, 小于第一阈值) 因此可能不精确的情况下, 启用误差纠正模块。在这些情况下, 编码器还对原始 z0a 和近似 z0a 之间的量化残差进行编码。即, z0 的重构 z0r 等于 z0a 或与 z0a 相差解量化的残差。

[0099] 在对另一向量的第一分量进行编码的情况下, y1 导出模块使用该一个向量的解量化的第一量化浮点分量值和解量化的第二量化浮点分量值 x0r、y0r 以及另一向量的解量化的第一量化浮点分量值 x1r, 计算另一向量的第二分量的两个可能解:

$$[0100] \quad y1r = (x0r * y0r * x1r + \sqrt{\Delta}) / (1 - x0r^2) \text{ 或}$$

$$[0101] \quad y1r = (x0r * y0r * x1r - \sqrt{\Delta}) / (1 - x0r^2)。$$

[0102] 在这些等式中, Δ 等于 $z0r^2 * (1 - x0r^2 - y0r^2)$, z0r 是可能的残差纠正重构。

[0103] 类似地, 在对另一向量的第二分量进行编码的情况下, x1 导出模块使用该一个向量的解量化的第一量化浮点分量值和解量化的第二量化浮点分量值 x0r、y0r 以及另一向量的解量化的第二量化浮点分量值 y1r, 计算另一向量的第一分量的两个可能解:

$$[0104] \quad x1r = (x0r * y0r * x1r + \sqrt{\Delta}) / (1 - y0r^2) \text{ 或}$$

$$[0105] \quad x1r = (x0r * y0r * x1r - \sqrt{\Delta}) / (1 - y0r^2)。$$

[0106] 在这些等式中, Δ 等于 $z0r^2 * (1 - x0r^2 - y0r^2)$, z0r 是可能的残差纠正重构。

[0107] 在 z0r 的绝对值非常小 (即, 小于第二阈值, 第二阈值甚至小于第一阈值) 的情况下, 既不激活 x1 导出模块也不激活 y1 导出模块。取而代之的, 对 z1 进行量化和编码。此外, 对另一符号比特进行编码, 该另一符号比特指示另一向量的重构第一分量和重构第二分量中绝对值较小的分量的符号。

[0108] 在 z0r 的绝对值不小于第二阈值的情况下是激活 x1 导出模块还是激活 y1 导出模块取决于该一个向量的重构第一分量和重构第二分量 x0r 和 y0r 的绝对值的关系。在 $\text{abs}(x0r) > \text{abs}(y0r)$ 的情况下, 对 x1 进行量化和编码并激活 y1 导出模块。在 $\text{abs}(x0r) \leq \text{abs}(y0r)$ 的情况下, 对 y1 进行量化和编码并激活 x1 导出模块。因为 x1 导出模块和 y1 导出模块中每个提供可能的解, 还对标记比特进行编码, 以向解码器指示要使用的解。

[0109] 在 z0 低于所述第一阈值的情况下, z1 导出模块使用平方根函数 $\text{sqrt}(\cdot)$ 、该一个向量的预定长度以及该一个向量的解量化的第一量化浮点分量值和解量化的第二量化浮点分量值, 计算 z1 的绝对值:

$$[0110] \quad z1 = \text{sqrt}(1 - x1r^2 - y1r^2)。$$

[0111] 在 z0 不低于所述第一阈值的情况下, 如下使用绝对值函数 $\text{abs}(\cdot)$ 计算 z1 的绝对值:

$$[0112] \quad z1 = \text{abs}((-x0r * x1r - y0r * y1r) / z0)。$$

[0113] 在针对 z1 所计算的值小并因此可能不精确的情况下, 还可以启用误差纠正模块。如果在针对 z1 所计算的值小的情况下启用了误差纠正模块, 则编码器还对原始 z1 和所计算的 z1 值之间的量化残差进行编码。

[0114] 图 2 示例性地示出了根据所述特定实施例的解码处理。

[0115] 总是对该一个向量的第一 $x0r$ 浮点分量值进行解码和解量化。此外,总是对标记比特进行解码。至少只要该第一量化向量分量的解量化值 $x0r$ 不等于一,就还对符号比特进行解码,该符号比特以信号通知该一个向量的第三分量 $z0r$ 的符号,并且还对该一个向量的第二分量 $y0$ 进行解码和解量化。

[0116] $z0$ 导出模块使用该一个向量的预定长度和已解码且已量化的数据,计算该一个向量的第三分量的近似 $z0a$ 。即,至少只要第一量化向量分量的解量化值 $x0r$ 的绝对值不等于 1,就使用该符号比特以及该第一量化向量分量和该第二量化向量分量的解量化值 $x0r$ 、 $y0r$ 来确定 $z0a$ 。在第一量化向量分量的解量化值 $x0r$ 的绝对值等于 1 的情况下,可以将 $y0r$ 和 $z0a$ 两者确定为零。

[0117] 在所计算的 $z0a$ 的值非常小(即,比第一阈值小,并因此可能不精确)的情况下,启用误差纠正模块。在这些情况下,解码器还对原始 $z0a$ 和近似 $z0a$ 之间的量化残差进行解码和解量化。即, $z0$ 的重构 $z0r$ 等于 $z0a$ 或与 $z0a$ 相差解量化的残差。

[0118] 在 $z0r$ 不小于小于第一阈值的第二阈值并且 $x0r$ 的绝对值小于 $y0r$ 的绝对值的情况下,对另一向量的第二量化分量进行解码和解量化,以获得 $y1r$,其中该标记比特指示使用两个可能解中的哪一个来计算另一向量的第一分量。然后,计算 $x1r$ 和 $z1r$ 。

[0119] 在 $z0r$ 不小于小于第一阈值的第二阈值并且 $x0r$ 的绝对值小于 $y0r$ 的绝对值的情况下, $y1$ 导出模块使用标记比特来选择两个可能解之一,以计算另一向量的第二分量:

$$[0120] \quad y1r = (x0r * y0r * x1r + \sqrt{\Delta}) / (1 - x0r^2) \text{ 或}$$

$$[0121] \quad y1r = (x0r * y0r * x1r - \sqrt{\Delta}) / (1 - x0r^2)。$$

[0122] 在这些等式中, Δ 等于 $z0r^2 * (1 - x0r^2 - y0r^2)$, $z0r$ 是可能的残差纠正重构。

[0123] 在 $z0r$ 不小于小于第一阈值的第二阈值并且 $x0r$ 的绝对值不小于 $y0r$ 的绝对值的情况下,对另一向量的第一量化分量进行解码和解量化,以获得 $x1r$,其中该标记比特指示要使用两个可能解中的哪一个来计算另一向量的第二分量。然后,计算 $y1r$ 和 $z1r$ 。

[0124] 在 $z0r$ 不小于小于第一阈值的第二阈值并且 $x0r$ 的绝对值不小于 $y0r$ 的绝对值的情况下, $x1$ 导出模块使用标记比特来选择两个可能解之一,以计算另一向量的第一分量:

$$[0125] \quad x1r = (x0r * y0r * y1r + \sqrt{\Delta}) / (1 - y0r^2) \text{ 或}$$

$$[0126] \quad x1r = (x0r * y0r * y1r - \sqrt{\Delta}) / (1 - y0r^2)。$$

[0127] 在这些等式中, Δ 等于 $z0r^2 * (1 - x0r^2 - y0r^2)$, $z0r$ 是可能的残差纠正重构。

[0128] 在 $z0r$ 小于第二阈值(第二阈值小于第一阈值)并且 $y0r$ 的绝对值小于 $x0r$ 的绝对值的情况下,对另一向量的第三量化分量进行解码和解量化,以获得 $z1r$,使用该标记比特来确定 $y1r$ 的 $\text{sign}(y1r)$,并且按下式计算 $x1r$ 和 $y1r$:

$$[0129] \quad y1r = \text{sign}(y1r) * \text{abs}(x0r) * \sqrt{(1 - z1r^2)}$$

$$[0130] \quad x1r = -x0r * x1r / y0r。$$

[0131] 在 $z0r$ 小于第二阈值(第二阈值小于第一阈值)并且 $y0r$ 的绝对值不小于 $x0r$ 的绝对值的情况下,对另一向量的第三量化分量进行解码和解量化,以获得 $z1r$,使用该标记比特来确定 $x1r$ 的 $\text{sign}(x1r)$,并且按下式计算 $x1r$ 和 $y1r$:

$$[0132] \quad x1r = \text{sign}(x1r) * \text{abs}(x0r) * \sqrt{(1 - z1r^2)}$$

$$[0133] \quad y1r = -x0r * x1r / y0r。$$

[0134] 在 z_{lr} 不是由解码和解量化得到而是由计算得到并且具有低于第一阈值的绝对值的情况下,可以启用误差纠正模块。在这些情况下,解码器还对原始 z_{lr} 和 z_{lr} 之间的另一量化残差进行解码和解量化,并根据该解量化的另一残差纠正 z_{lr} 。

[0135] 尽管特定实施例根据笛卡尔坐标描述了本发明,但本发明不限于此。在球坐标中,针对接近 $\pi/2$ (还记作 90 度或 $\pi/2$) 的角度,发生类似的问题。即,在所述所计算的角度的绝对值与 $\pi/2$ 相差不到阈值的情况下,提出对归一化向量的所计算角度的残差进行编码。

[0136] 如上所述,本原理可以有益地应用在对 3D 模型重复结构进行编码的上下文中。为了对 3D 模型高效地进行编码,可以将重复结构组织为模式和实例,其中,使用对应模式的模式 ID 和包含与平移、旋转和缩放有关的信息在内的变换矩阵,将实例表示为例如对应模式的变换。

[0137] 当实例由模式 ID 和变换矩阵表示时,当对该实例进行压缩时要对该模式 ID 和变换矩阵进行压缩。因此,可以通过模式 ID 和所解码的变换矩阵来重构实例,即,可以将实例重构为由模式 ID 索引的解码模式的变换(来自所解码的变换矩阵)。

[0138] 图 3 示出了示例性 3D 模型编码器 300 的方框图。装置 300 的输入可以包括 3D 模型、用于对该 3D 模型进行编码的质量参数以及其他元数据。该 3D 模型首先通过重复结构发现模块 310,重复结构发现模块 310 按照模式、实例和唯一分量输出该 3D 模型。使用模式编码器 320 来对模式进行压缩,并且使用唯一分量编码器 350 来对唯一分量进行编码。针对实例,基于用户所选择的模式对该实例分量信息进行编码。如果选择了实例信息组模式,则使用分组实例信息编码器 340 对该实例信息进行编码;否则,使用基本实例信息编码器 330 对该实例信息进行编码。进一步在重复结构验证器 360 中对已编码的分量进行验证。如果已编码的分量不满足其质量需求,将使用唯一分量编码器 350 来对该已编码的分量进行编码。在比特流组装器 370 处对模式、实例和唯一分量的比特流进行组装。

[0139] 图 4 示出了示例性 3D 模型解码器 400 的方框图。装置 400 的输入可以包括 3D 模式的比特流,例如由编码器 300 产生的比特流。由模式解码器 420 对压缩比特流中的与模式有关的信息进行解码。通过唯一分量解码器 450 对与唯一分量有关的信息进行解码。该实例信息的解码还取决于用户选择的模式。如果选择了实例信息组模式,则使用分组实例信息解码器 440 对该实例信息进行解码;否则,使用基本实例信息解码器 430 对该实例信息进行解码。对解码模式、实例信息和唯一分量进行重构,以在模型重构模块 460 处产生输出 3D 模型。

[0140] 这里所描述的实现可以在例如方法或处理、装置、软件程序、数据流或信号中实现。即使仅在单个形式的实现(例如,仅作为方法讨论)的上下文中讨论,所讨论的特征的实现还可以以其他形式(例如,装置或程序)实现。装置可以在例如合适的硬件、软件和固件中实现。方法可以在例如诸如处理器(一般指代处理设备,包括例如计算机、微处理器、集成电路或可编程逻辑设备)的装置中实现。处理器还包括通信设备,例如计算机、蜂窝电话、便携式/个人数字助理(“PDA”)以及有利于最终用户之间信息通信的其他设备。

[0141] 引用本原理的“一个实施例”或“实施例”或“一个实现”或“实现”及其其他变体意味着结合该实施例描述的特定特征、结构、特性等被包括在本原理的至少一个实施例中。因此,出现在整个说明书中各个地方的词组“在一个实施例中”或“在实施例中”或“在一个

实现中”或“在实现中”以及任意其他变体不一定都指代相同实施例。

[0142] 将对本领域技术人员显而易见的是,实现可以产生用于携带可以例如被存储或传输的信息的各种格式的信号。该信息可以包括例如用于执行方法的指令或由所述实现之一所产生的数据。例如,可以规定信号的格式,用于携带所述实施例的比特流。可以将这种信号的格式规定为例如电磁波(例如,使用频谱的射频部分)或规定为基带信号。该规定格式可以包括例如对数据流进行编码和利用已编码的数据流调制载波。该信号携带的信息可以是例如模拟信息或数字信息。可以通过已知的各种不同的有线链路或无线链路传输该信号。该信号可以存储在处理器可读介质中。

[0143] 原则上,所公开的发明还可以应用于其他数据压缩领域。本发明产生唯一的比特流格式。

[0144] 尽管比特流嵌入所有的变换数据,其是高效的并且可以应付若干应用,在所述应用中,有时比特流大小或解码效率或容错性是最重要的。因此,针对如何将一个实例的变换数据(即,其位置、取向和缩放因子)放到比特流中,公开了两种模式选项。在第一模式(选项 A)中,一个实例的位置、取向和可能的缩放因子被包装在一起放在比特流中。在第二模式(选项 B)中,所有实例的位置、取向和可能的缩放因子被包装在一起放在比特流中。

[0145] 将理解,已经纯粹通过示例方式描述了本发明,并且在不脱离本发明的范围的前提下,可以对细节进行修改。

[0146] 说明书以及(当合适时)权利要求和附图中所公开的每个特征可以独立地或以任何合适组合的形式提供。特征可以在合适时以硬件、软件或这两者组合的方式实现。连接可以在可应用时被实现为无线连接或有线(不一定是直接或专用)连接。

[0147] 权利要求中出现的附图标记是仅通过示意方式的,并且不应当对权利要求的范围具有限制作用。

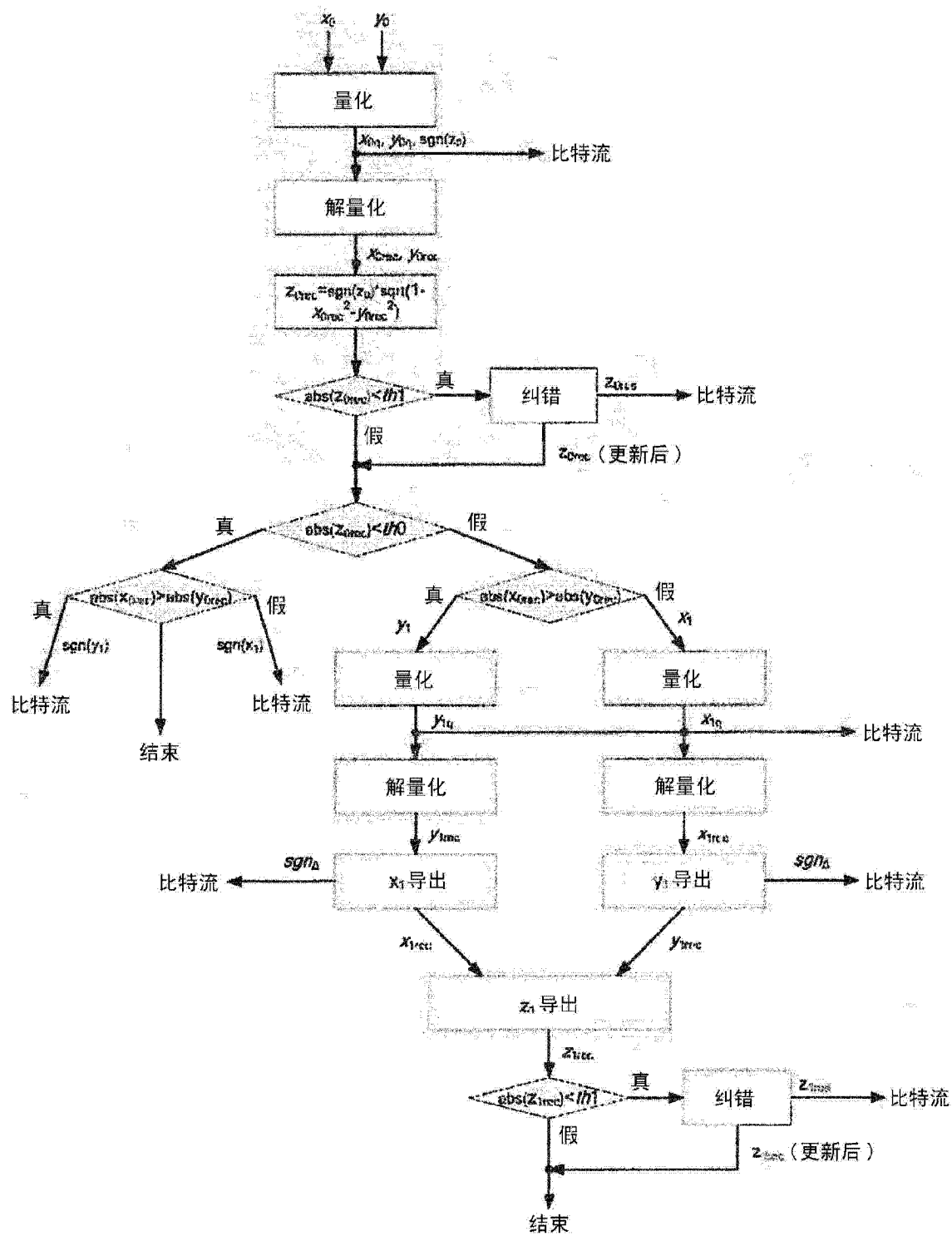


图 1

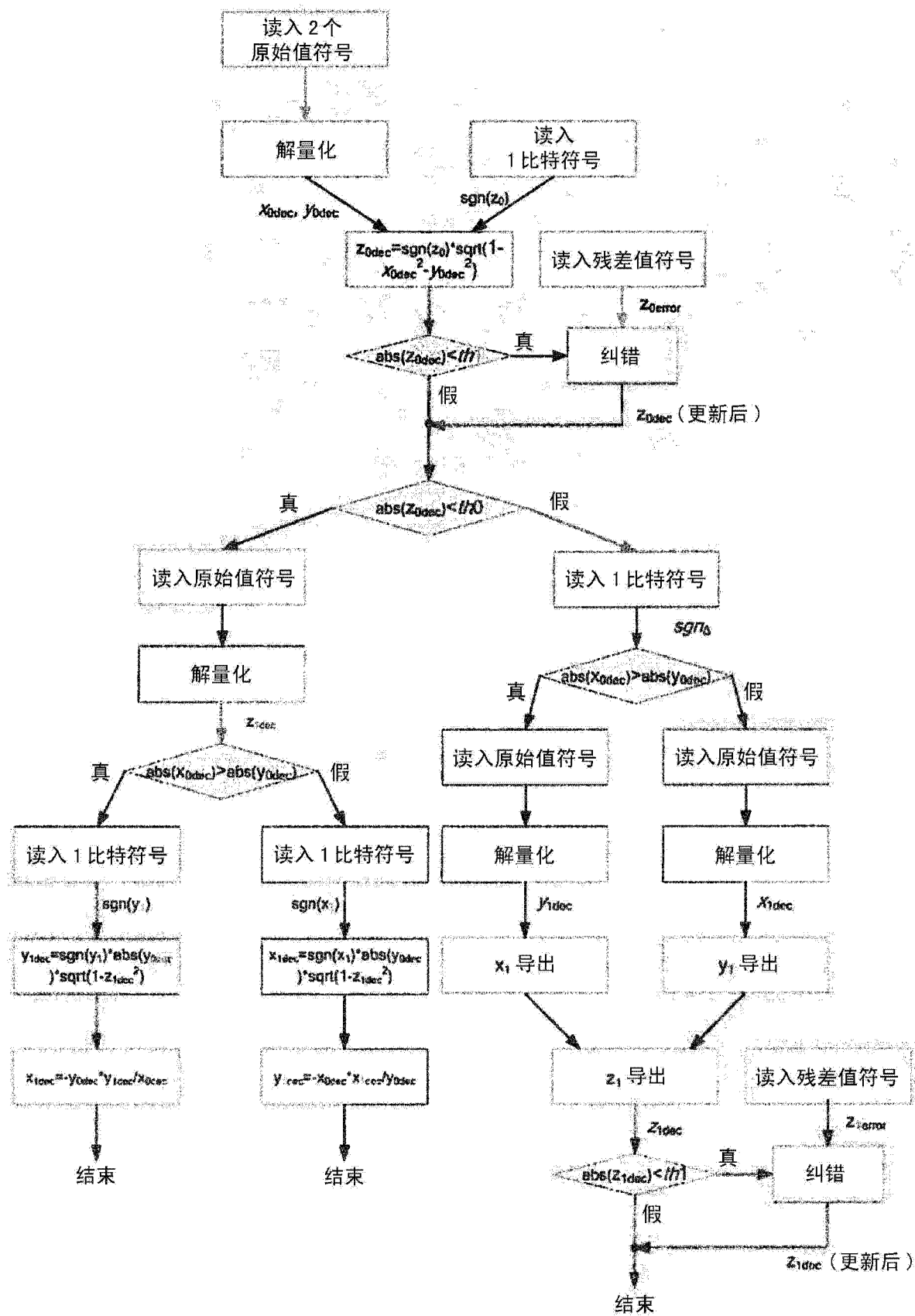


图 2

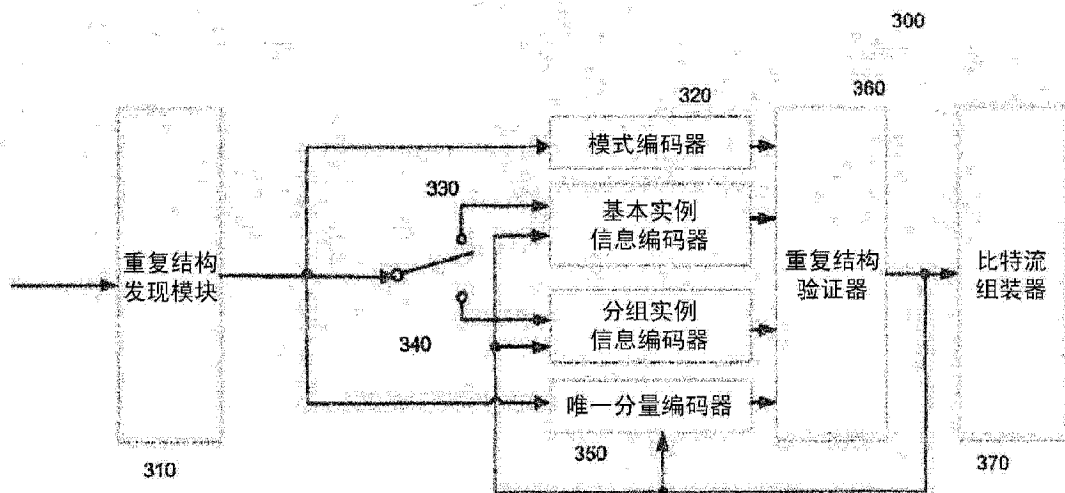


图 3

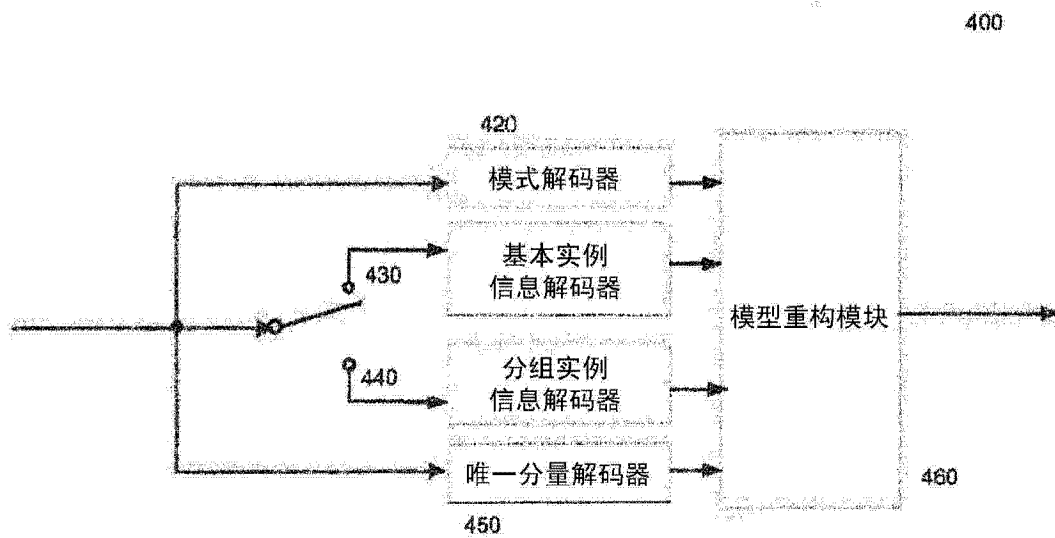


图 4