



# (12)发明专利申请

(10)申请公布号 CN 109408260 A

(43)申请公布日 2019.03.01

(21)申请号 201811119377.3

(22)申请日 2018.09.25

(71)申请人 深圳忆联信息系统有限公司

地址 518067 广东省深圳市南山区蛇口街  
道蛇口后海大道东角头厂房D24/F-02

(72)发明人 管金新 郭超

(74)专利代理机构 深圳市精英专利事务所  
44242

代理人 冯筠

(51)Int.Cl.

G06F 11/07(2006.01)

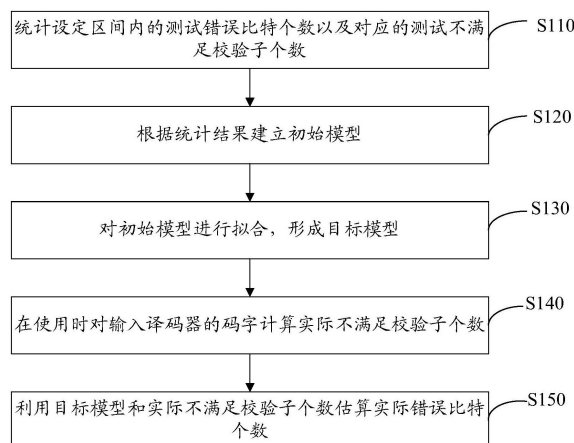
权利要求书2页 说明书12页 附图7页

## (54)发明名称

错误比特个数估计方法、装置、计算机设备及存储介质

## (57)摘要

本发明涉及错误比特个数估计方法、装置、计算机设备及存储介质,该方法包括统计设定区间内的测试错误比特个数以及对应的测试不满足校验子个数;根据统计结果建立初始模型;对初始模型进行拟合,形成目标模型;在使用时对输入译码器的码字计算实际不满足校验子个数;利用目标模型和实际不满足校验子个数估算实际错误比特个数。本发明通过对设定区间内的测试错误比特个数以及对应的测试不满足校验子个数进行统计,获取不同区间内的目标模型,在实际使用过程中,根据实际不满足校验子个数所落入区间的目标模型估计实际错误比特个数,有效进行译码模式的选择或者直接进行下一次读重试,减少不必要的译码迭代和重试的次数,可以提高整体的译码效率。



1. 错误比特个数估计方法,其特征在于,包括:  
统计设定区间内的测试错误比特个数以及对应的测试不满足校验子个数;  
根据统计结果建立初始模型;  
对初始模型进行拟合,形成目标模型;  
在使用时对输入译码器的码字计算实际不满足校验子个数;  
利用目标模型和实际不满足校验子个数估算实际错误比特个数。
2. 根据权利要求1所述的错误比特个数估计方法,其特征在于,所述统计设定区间内的测试错误比特个数以及对应的测试不满足校验子个数,包括:  
获取原始数据;  
对原始数据进行编码,获取测试码字;  
设定随机注入的测试错误比特个数的区间、步长以及最大统计次数;  
利用随机数产生对应的测试错误比特个数;  
将测试错误比特个数加入测试码字;  
根据测试码字计算测试不满足校验子个数;  
记录测试错误比特个数对应的测试不满足校验子个数;  
判断记录次数是否达到最大统计次数;  
若否,则返回所述利用随机数产生对应的测试错误比特步骤;  
若是,则根据步长重新设定注入的测试错误比特个数;  
判断注入的测试错误比特个数是否达到区间的最大值;  
若是,则进入所述根据测试错误比特个数以及对应的测试不满足校验子个数建立初始模型步骤;  
若否,则返回所述利用随机数产生对应的测试错误比特步骤。
3. 根据权利要求1或2所述的错误比特个数估计方法,其特征在于,所述根据统计结果建立初始模型,包括:  
对同一测试错误比特个数对应的测试校验子个数取平均值,以形成目标校验子个数;  
根据测试错误比特个数的区间及对应的目标校验子个数,获取关系曲线,以得到初始模型。
4. 根据权利要求3所述的错误比特个数估计方法,其特征在于,所述对初始模型进行拟合,形成目标模型,包括:  
根据关系曲线对目标校验子个数进行阈值区间划分;  
根据划分结果以及拟合公式采用最小方差进行分段拟合,确定影响因子和偏移值;  
根据影响因子和偏移值确定目标模型。
5. 根据权利要求4所述的错误比特个数估计方法,其特征在于,所述拟合公式包括: $s_{\mu}(x) = a \cdot x + b$ ;其中, $a > 0, b > 0$ ;  $a$ 为比例影响因子,  $b$ 为偏移值。
6. 根据权利要求4所述的错误比特个数估计方法,其特征在于,所述利用目标模型和实际不满足校验子个数估算实际错误比特个数,包括:  
获取实际不满足校验子个数所在的实际区间;  
利用实际区间的目标模型,根据实际不满足校验子个数估算实际错误比特个数。
7. 错误比特个数估计装置,其特征在于,包括:

统计单元,用于统计设定区间内的测试错误比特个数以及对应的测试不满足校验子个数;

初始模型建立单元,用于根据统计结果建立初始模型;

拟合单元,用于对初始模型进行拟合,形成目标模型;

实际个数计算单元,用于在使用时对输入译码器的码字计算实际不满足校验子个数;

比特个数估算单元,用于利用目标模型和实际不满足校验子个数估算实际错误比特个数。

8. 根据权利要求7所述的错误比特个数估计装置,其特征在于,所述统计单元包括:

数据获取模块,用于获取原始数据;

编码模块,用于对原始数据进行编码,获取测试码字;

设定模块,用于设定随机注入的测试错误比特个数的区间、步长以及最大统计次数;

随机产生模块,用于利用随机数产生对应的测试错误比特个数;

加入模块,用于将测试错误比特个数加入测试码字;

计算模块,用于根据测试码字计算测试不满足校验子个数;

记录模块,用于记录测试错误比特个数对应的测试校验子个数;

次数判断模块,用于判断记录次数是否达到最大统计次数;

步长设定模块,用于若是,则根据步长重新设定注入的测试错误比特个数;

区间判断模块,用于判断注入的测试错误比特个数是否达到区间的最大值。

9. 一种计算机设备,其特征在于,所述计算机设备包括存储器及处理器,所述存储器上存储有计算机程序,所述处理器执行所述计算机程序时实现如权利要求1至6中任一项所述的方法。

10. 一种存储介质,其特征在于,所述存储介质存储有计算机程序,所述计算机程序被处理器执行时可实现如权利要求1至6中任一项所述的方法。

## 错误比特个数估计方法、装置、计算机设备及存储介质

### 技术领域

[0001] 本发明涉及固态硬盘,更具体地说是指错误比特个数估计方法、装置、计算机设备及存储介质。

### 背景技术

[0002] 固态硬盘由于其容易产生数据读写错误的缺点,在擦写次数较多之后数据出错的概率显著增大,因此非常需要纠错算法保证,比如,固态硬盘采用ECC算法需要去校验更改数据错误,以及判断是否坏块,准确估算出错误比特个数,对于估计信道环境的好坏具有重要的意义,但是纠错算法的纠错能力基本是固定的,而目前还没有较为合适的算法能够准确地估计在进入LDPC译码器之前码字的错误比特个数,进而无法减少重试次数,使得译码效率低下,

[0003] 因此,有必要设计一种方法,实现减少重试的次数,提高译码效率。

### 发明内容

[0004] 本发明的目的在于克服现有技术的缺陷,提供错误比特个数估计方法、装置、计算机设备及存储介质。

[0005] 为实现上述目的,本发明采用以下技术方案:错误比特个数估计方法,包括:

[0006] 统计设定区间内的测试错误比特个数以及对应的测试不满足校验子个数;

[0007] 根据统计结果建立初始模型;

[0008] 对初始模型进行拟合,形成目标模型;

[0009] 在使用时对输入译码器的码字计算实际不满足校验子个数;

[0010] 利用目标模型和实际不满足校验子个数估算实际错误比特个数。

[0011] 其进一步技术方案为:所述统计设定区间内的测试错误比特个数以及对应的测试不满足校验子个数,包括:

[0012] 获取原始数据;

[0013] 对原始数据进行编码,获取测试码字;

[0014] 设定随机注入的测试错误比特个数的区间、步长以及最大统计次数;

[0015] 利用随机数产生对应的测试错误比特个数;

[0016] 将测试错误比特个数加入测试码字;

[0017] 根据测试码字计算测试不满足校验子个数;

[0018] 记录测试错误比特个数对应的测试不满足校验子个数;

[0019] 判断记录次数是否达到最大统计次数;

[0020] 若否,则返回所述利用随机数产生对应的测试错误比特步骤;

[0021] 若是,则根据步长重新设定注入的测试错误比特个数;

[0022] 判断注入的测试错误比特个数是否达到区间的最大值;

[0023] 若是,则进入所述根据测试错误比特个数以及对应的测试不满足校验子个数建立

初始模型步骤；

[0024] 若否，则返回所述利用随机数产生对应的测试错误比特步骤。

[0025] 其进一步技术方案为：所述根据统计结果建立初始模型，包括：

[0026] 对同一测试错误比特个数对应的测试校验子个数取平均值，以形成目标校验子个数；

[0027] 根据测试错误比特个数的区间及对应的目标校验子个数，获取关系曲线，以得到初始模型。

[0028] 其进一步技术方案为：所述对初始模型进行拟合，形成目标模型，包括：

[0029] 根据关系曲线对目标校验子个数进行阈值区间划分；

[0030] 根据划分结果以及拟合公式采用最小方差进行分段拟合，确定影响因子和偏移值；

[0031] 根据影响因子和偏移值确定目标模型。

[0032] 其进一步技术方案为：所述拟合公式包括： $s_{\mu}(x) = a \cdot x + b$ ；其中， $a > 0$ ， $b > 0$ ； $a$ 为比例影响因子， $b$ 为偏移值。

[0033] 其进一步技术方案为：所述利用目标模型和实际不满足校验子个数估算实际错误比特个数，包括：

[0034] 获取实际不满足校验子个数所在的实际区间；

[0035] 利用实际区间的目标模型，根据实际不满足校验子个数估算实际错误比特个数。

[0036] 本发明还提供了错误比特个数估计装置，包括：

[0037] 统计单元，用于统计设定区间内的测试错误比特个数以及对应的测试不满足校验子个数；

[0038] 初始模型建立单元，用于根据统计结果建立初始模型；

[0039] 拟合单元，用于对初始模型进行拟合，形成目标模型；

[0040] 实际个数计算单元，用于在使用时对输入译码器的码字计算实际不满足校验子个数；

[0041] 比特个数估算单元，用于利用目标模型和实际不满足校验子个数估算实际错误比特个数。

[0042] 其进一步技术方案为：所述统计单元包括：

[0043] 数据获取模块，用于获取原始数据；

[0044] 编码模块，用于对原始数据进行编码，获取测试码字；

[0045] 设定模块，用于设定随机注入的测试错误比特个数的区间、步长以及最大统计次数；

[0046] 随机产生模块，用于利用随机数产生对应的测试错误比特个数；

[0047] 加入模块，用于将测试错误比特个数加入测试码字；

[0048] 计算模块，用于根据测试码字计算测试不满足校验子个数；

[0049] 记录模块，用于记录测试错误比特个数对应的测试校验子个数；

[0050] 次数判断模块，用于判断记录次数是否达到最大统计次数；

[0051] 步长设定模块，用于若是，则根据步长重新设定注入的测试错误比特个数；

[0052] 区间判断模块，用于判断注入的测试错误比特个数是否达到区间的最大值。

[0053] 本发明还提供了一种计算机设备,所述计算机设备包括存储器及处理器,所述存储器上存储有计算机程序,所述处理器执行所述计算机程序时实现上述的方法。

[0054] 本发明还提供了一种存储介质,所述存储介质存储有计算机程序,所述计算机程序被处理器执行时可实现上述的方法。

[0055] 本发明与现有技术相比的有益效果是:本发明通过对设定区间内的测试错误比特个数以及对应的测试不满足校验子个数进行统计,并根据统计结果建立初始模型,对初始模型进行拟合,以获取不同区间内的目标模型,在实际使用过程中,先对译码前的码字进行实际不满足校验子个数的计算,根据实际不满足校验子个数所落入区间的目标模型估计实际错误比特个数,有效进行译码模式的选择或者直接进行下一次读重试,减少不必要的译码迭代和重试的次数,可以提高整体的译码效率。

[0056] 下面结合附图和具体实施例对本发明作进一步描述。

## 附图说明

[0057] 为了更清楚地说明本发明实施例技术方案,下面将对实施例描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

[0058] 图1为本发明实施例提供的错误比特个数估计方法的流程示意图;

[0059] 图2为本发明实施例提供的错误比特个数估计方法的子流程示意图;

[0060] 图3为本发明实施例提供的错误比特个数估计方法的子流程示意图;

[0061] 图4为本发明实施例提供的错误比特个数估计方法的子流程示意图;

[0062] 图5为本发明实施例提供的错误比特个数估计方法的子流程示意图;

[0063] 图6为本发明实施例提供的所有4比特错误落入8环的结果示意图;

[0064] 图7为本发明实施例提供的所有4比特错误落入10环的结果示意图;

[0065] 图8为本发明实施例提供的所有4比特错误落入12环和14环的结果示意图;

[0066] 图9为本发明实施例提供的所有4比特错误互相没有边线连接的结果示意图;

[0067] 图10为本发明实施例提供的码字错误比特个数与不满足的校验子个数的拟合关系示意图;

[0068] 图11为本发明实施例提供的错误比特个数估计装置的示意性框图;

[0069] 图12为本发明实施例提供的计算机设备的示意性框图。

## 具体实施方式

[0070] 下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例是本发明一部分实施例,而不是全部的实施例。基于本发明中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例,都属于本发明保护的范围。

[0071] 应当理解,当在本说明书和所附权利要求书中使用时,术语“包括”和“包含”指示所描述特征、整体、步骤、操作、元素和/或组件的存在,但并不排除一个或多个其它特征、整体、步骤、操作、元素、组件和/或其集合的存在或添加。

[0072] 还应当理解,在此本发明说明书中所使用的术语仅仅是出于描述特定实施例的目

的而并不意在限制本发明。如在本发明说明书和所附权利要求书中所使用的那样,除非上下文清楚地指明其它情况,否则单数形式的“一”、“一个”及“该”意在包括复数形式。

[0073] 还应当进一步理解,在本发明说明书和所附权利要求书中使用的术语“和/或”是指相关联列出的项中的一个或多个的任何组合以及所有可能组合,并且包括这些组合。

[0074] 请参阅图1,图1是本发明实施例提供的错误比特个数估计方法的流程示意图。该错误比特个数估计方法可运用于包含LDPC的ECC算法的固态硬盘内,以对固态硬盘的译码器工作前进行数据纠错,实现减少重试的次数,提高译码效率。

[0075] 如图1所示,该方法包括以下步骤S110至S150。

[0076] S110、统计设定区间内的测试错误比特个数以及对应的测试不满足校验子个数。

[0077] 在本实施例中,测试错误比特个数指的是在测试以建立模型过程中采用的测试数据产生的测试错误比特数量;测试不满足校验子个数指的是测试以建立模型过程中所产生的不满足校验子个数。

[0078] 由于测试错误比特个数与对应的测试不满足校验子个数存在一定的线性关系,因此,通过对这两个数值进行统计,再根据统计结果进行建模,有助于估计出进入LDPC译码器之前的码字的错误比特个数,可以有效进行译码模式的选择或者直接进行下一次读重试操作,这样可以大大减少不必要的译码迭代,可以提高整体的译码效率。错误比特个数也是一个重要的参数,以固态硬盘存储器读取数据为例,在出错较大的情况下,可以提前进行最优读偏移电压的调整,以便获得更准确的读数据。

[0079] 在一实施例中,如图2所示,上述的步骤S110可包括步骤S110a~S110j。

[0080] S110a、获取原始数据;

[0081] 在本实施例中,原始数据指的是写入固态硬盘的测试数据。

[0082] S110b、对原始数据进行编码,获取测试码字。

[0083] 测试码字指的是在写入固态硬盘之前,进行编码以对原始数据进行纠错后形成的码字。在本实施例中,具体是利用LDPC编码器对原始数据进行编码,LDPC(低密度奇偶校验码,Low Density Parity Check Code)具有稀疏校验矩阵的线性分组码,不仅有逼近Shannon限的良好性能,而且译码复杂度较低,结构灵活,并且具有优异的纠错性能。

[0084] S110c、设定随机注入的测试错误比特个数的区间、步长以及最大统计次数。

[0085] 在获取测试不满足校验子个数时,且在不同的测试错误比特个数的区间内,测试错误比特个数与测试不满足校验子个数的关系参数会发生变化,因此,需要设定若干个随机注入的测试错误比特个数的区间,在获取测试不满足校验子个数时,要针对区间内的所有测试错误比特个数进行获取测试不满足校验子个数,最优选地,要重复多次获取,以提高准确率,因此,需要设定最大统计次数。

[0086] 在获取测试不满足校验子个数时,还需要按照设定的步长增加测试错误比特的个数,以完成整个区间的测试错误比特个数获取测试不满足校验子个数的流程。

[0087] S110d、利用随机数产生对应的测试错误比特个数。

[0088] 采用随机数的方式产生的测试错误比特个数才能提高获取准确率。

[0089] S110e、将测试错误比特个数加入测试码字。

[0090] S110f、根据测试码字计算测试不满足校验子个数。

[0091] 在本实施例中,利用 $S=H \cdot m^T$ 进行校验子计算,其中, $m$ 代表测试码字, $H$ 代表LDPC

纠错码采用的校验矩阵,是一种线性分组码矩阵,用于计算校验子。

[0092] S110g、记录测试错误比特个数对应的测试不满足校验子个数;

[0093] 将测试不满足校验子个数记录,以便于后续的处理。

[0094] S110h、判断记录次数是否达到最大统计次数。

[0095] 对同一测试错误比特个数进行多次获取测试不满足校验子个数,以提高准确率,减少误差

[0096] 若否,则返回步骤S110d;

[0097] 若是,则S110i、根据步长重新设定注入的测试错误比特个数。

[0098] 在对同一测试错误比特个数进行最大统计次数获取测试不满足校验子个数后,需要针对下一测试错误比特个数进行上述的流程,因此,需要根据设定的步长增加测试错误比特个数,用同样的流程进行统计,直至当前注入的测试错误比特个数达到区间的最大值。

[0099] S110j、判断注入的测试错误比特个数是否达到区间的最大值;

[0100] 以获取设定区间内若干个测试错误比特个数对应的测试不满足校验子个数,将上述记录的内容作为统计结果。

[0101] 若是,则进入步骤S120;

[0102] 若否,则返回步骤S110d。

[0103] S120、根据统计结果建立初始模型。

[0104] 在本实施例中,统计结果是指统计的设定区间内的测试错误比特个数以及对应的测试不满足校验子个数;初始模型指的是有若干个测试错误比特个数以及对应的测试不满足校验子个数形成的曲线图。

[0105] 在一实施例中,如图3所示,上述的步骤S120可包括步骤S121~S122。

[0106] S121、对同一测试错误比特个数对应的测试校验子个数取平均值,以形成目标校验子个数;

[0107] 以取平均值的方式从多个测试校验子个数获取一个代表该测试错误比特个数对应的测试校验子个数,可以使得提高初始模型的准确率。

[0108] S122、根据测试错误比特个数的区间及对应的目标校验子个数,获取关系曲线,以得到初始模型。

[0109] 在本实施例中,可以采用matlab等工具画出测试错误比特个数的区间及对应的目标校验子个数的关系曲线。

[0110] S130、对初始模型进行拟合,形成目标模型。

[0111] 在本实施例中,目标模型是指针对初始模型进行曲线拟合以形成的曲线。

[0112] 在一实施例中,如图4所示,上述的步骤S130可包括步骤S131~S133。

[0113] S131、根据关系曲线对目标校验子个数进行阈值区间划分。

[0114] 由于不同的区间内,错误比特个数与目标校验子个数之间的关系会不同,因此,需要对目标校验子个数进行阈值区间划分,以获取每个区间内错误比特个数与目标校验子个数之间的关系,从而提高错误比特个数估计的准确率。

[0115] S132、根据划分结果以及拟合公式采用最小方差进行分段拟合,确定影响因子和偏移值。

[0116] 在本实施例中,所述拟合公式包括: $s_{\mu}(x) = a \cdot x + b$ ,其中, $a > 0, b > 0$ ;a为比例影



响因子,  $b$  为偏移值。

[0117] 根据曲线对不满足的校验子设定不同的分段阈值, 对各区间的点利用最小方差算法进行拟合, 确定各个分段区间的影响因子和偏移值。

[0118] S133、根据影响因子和偏移值确定目标模型。

[0119] 当确定各个分段区间的影响因子和偏移值, 则可以采用  $s_{\mu}(x) = a \cdot x + b$  的公式确定错误比特个数和不满足的校验子的关系, 从而确定目标模型。

[0120] S140、在使用时对输入译码器的码字计算实际不满足校验子个数。

[0121] 对输入译码器的码字先进行实际不满足校验子个数计算, 具体采用  $S = H \cdot m^T$  计算。

[0122] S150、利用目标模型和实际不满足校验子个数估算实际错误比特个数。

[0123] 在一实施例中, 上述的步骤S150可包括步骤S151~S152。

[0124] S151、获取实际不满足校验子个数所在的实际区间;

[0125] S152、利用实际区间的目标模型, 根据实际不满足校验子个数估算实际错误比特个数。

[0126] 确定实际不满足校验子个数确定其落入哪个区间, 根据实际不满足校验子个数确定落入的区间对应的目标模型  $s_{\mu}(x) = a \cdot x + b$ , 输入实际不满足校验子个数  $s_{\mu}(x)$ , 对实际错误比特个数进行估计, 可以有效的避免在错误比特个数很多的情况下进入译码器纠不回来而造成的带宽的浪费, 从而提高译码效率, 该过程也同样适用于不规则QC-LDPC的码字出错的错误估计。

[0127] 举一个例子: 以PEG算法构造的规则QC-LDPC校验阵为例, 此类校验阵具有相同的列重, 由于LDPC码是比较稀疏的分组线性码, 通常可以用二分图进行表示, 如  $G = (V = L \cup R, E)$ ,  $L$  表示所有变量节点的集合,  $R$  表示所有校验节点的集合,  $E$  表示变量节点与校验节点之间的连线。假设LDPC校验阵的列重为  $1 = 4$  以校验阵中其中一种错误模式, 即8环为例来分析, 如图6所示, 黑色方框表示校验阵校验计算不满足的校验节点, 白色方框表示满足的校验节点, 黑色圆圈表示发生错误的变量节点, 每一个变量节点连接4个校验节点, 中间4个校验节点连接偶数个错误的变量节点, 因此校验子满足, 所以不满足的校验子个数为8, 这是一种特例, 在只有4比特错误的情况下, 正好全部落入LDPC校验阵中的一个8环的概率极低, 因为校验阵码长较长, 而且稀疏。

[0128] 当然, LDPC校验阵中不止8环一种错误模式, 如图7所示, 假设所有4比特错误落入10环的错误模式, 则会多出两个校验子不满足。同样的如图8所示, 分别落入12环和14环的错误模式, 分别多出两条和四条不满足的校验子。在LDPC码字只有4比特出错的情况下, 很大概率4个错误比特随机出错不会有任何边线连接, 其结果如图9所示。由此可以得出对于LDPC码字出现随机错的情况, 落入不同的错误模式, 计算得出的校验子是不一样的, 但是在一定的区间范围之内, 理论上统计出所有环的个数, 计算出落入环的概率就可以根据校验子估计错误比特的个数, 通过统计的方法模拟校验子与错误比特个数的关系。由于从一次随机注入  $n$  个错误位置, 来确定不满足校验子个数与错误比特个数的关系不具有一般性, 在不同错误模式下是不一样的, 因此需要从统计意义上来分析, 根据LDPC校验阵的特点, 从统计分析上来说每增加1比特的错误, 即增加一个步长, 不满足的校验节点应该也会增加。假设随机注入的错误比特个数用  $x$  表示, 而不满足的校验子个数可以用  $s_{\mu}(x)$  表示, 可以对  $s_{\mu}$

(x) 进行拟合, 即:  $s_{\mu}(x) = a \cdot x + b$ ; 其中,  $a > 0, b > 0$ ;  $a$  为比例影响因子,  $b$  为偏移值。

[0129] 当错误比特个数较少时, 由于码字本身的稀疏性, 如图10所示, 因此  $a \leq 1$ , 当  $a = 1$  时,  $b = 0$ ; 随着错误比特个数增加, 同一个校验节点可能出现更多比特的错误, 因此影响因子  $a$  会变小。通过对不同区间不满足的校验子个数设置阈值, 从而进行分段拟合, 以获得更高的准确度。其大致的函数拟合情况如图11所示, 针对特定的LDPC校验阵, 曲线中每一段的影响因子  $a$  和偏移值  $b$  都可以通过统计获得。通过实际计算出来的校验子不满足的个数分别与三个阈值  $T_{u1}, T_{u2}, T_{u3}$  进行比较, 确定落入哪个区间, 从而确定  $a$  和  $b$ , 再估计出码字出错的错误比特个数:  $x = (s_{\mu}(x) - b) / a$ 。

[0130] 通过PEG构造列重固定的QC-LDPC校验阵的情况, 这种估计方法同样也适用于不规则的LDPC校验阵的情况, 只要列重的分布大致均匀, 同样可以利用统计的方式来获得对应的影响因子和偏移值, 而且结果不会出现太大的波动。如果列重分布完全没有规律, 也可以通过多划分一些区间来提高拟合的精度, 从而可以估计出更精确的错误比特个数。以实现通过不满足的校验子个数落入的区间来估计码字错误比特的个数。

[0131] 基于LDPC线性分组码估计码字在进入译码器之前出现错误的比特个数, 通过码字和校验矩阵计算出来的校验子信息, 利用不满足的校验子个数, 根据LDPC校验阵的特点, 通过对不满足的校验子个数设置不同的阈值来估计码字的错误比特个数, 对于估计信道环境的好坏具有重要的意义, 也提供给LDPC译码器一个重要的参考因素, 从而可以选择更合适的译码方式, 提高译码效率。通过对LDPC校验阵的特点进行相应的建模分析, 在不同出错情况下错误比特个数对校验子的影响, 以确定出不同错误区间的影响因子, 从而估计出码字出错的个数, 可以获得在不同的错误模式之下, 不满足的校验子的个数和实际的错误比特个数存在一个合理的比例关系, 对不满足的校验子个数设置不同的阈值可以获得不同区间的影响因子和偏移值, 从而可以估计出码字错误比特的个数, 以此类推, 不同其他校验阵也可以估计码字出错的错误比特个数。

[0132] 纠错算法的纠错能力基本是固定的, 因此在译码纠错之前能够借助一定的方法估计出大致出错的比特个数, 从而选择合适的模式进行译码, 这样可以大大减少重试的次数, 提高译码效率。

[0133] 上述的错误比特个数估计方法, 通过对设定区间内的测试错误比特个数以及对应的测试不满足校验子个数进行统计, 并根据统计结果建立初始模型, 对初始模型进行拟合, 以获取不同区间内的目标模型, 在实际使用过程中, 先对译码前的码字进行实际不满足校验子个数的计算, 根据实际不满足校验子个数所落入区间的目标模型估计实际错误比特个数, 有效进行译码模式的选择或者直接进行下一次读重试, 减少不必要的译码迭代和重试的次数, 可以提高整体的译码效率。

[0134] 图11是本发明实施例提供的一种错误比特个数估计装置300的示意性框图。如图11所示, 对应于以上错误比特个数估计方法, 本发明还提供一种错误比特个数估计装置300。该错误比特个数估计装置300包括用于执行上述错误比特个数估计方法的单元, 该装置可以被配置于包含LDPC的ECC算法的固态硬盘内。

[0135] 具体地, 请参阅图11, 该错误比特个数估计装置300包括:

[0136] 统计单元301, 用于统计设定区间内的测试错误比特个数以及对应的测试不满足校验子个数;

- [0137] 初始模型建立单元302,用于根据统计结果建立初始模型;
- [0138] 拟合单元303,用于对初始模型进行拟合,形成目标模型;
- [0139] 实际个数计算单元304,用于在使用时对输入译码器的码字计算实际不满足校验子个数;
- [0140] 比特个数估算单元305,用于利用目标模型和实际不满足校验子个数估算实际错误比特个数。
- [0141] 在一实施例中,所述统计单元301包括:
- [0142] 数据获取模块,用于获取原始数据;
- [0143] 编码模块,用于对原始数据进行编码,获取测试码字;
- [0144] 设定模块,用于设定随机注入的测试错误比特个数的区间、步长以及最大统计次数;
- [0145] 随机产生模块,用于利用随机数产生对应的测试错误比特个数;
- [0146] 加入模块,用于将测试错误比特个数加入测试码字;
- [0147] 计算模块,用于根据测试码字计算测试不满足校验子个数;
- [0148] 记录模块,用于记录测试错误比特个数对应的测试校验子个数;
- [0149] 次数判断模块,用于判断记录次数是否达到最大统计次数;
- [0150] 步长设定模块,用于若是,则根据步长重新设定注入的测试错误比特个数;
- [0151] 区间判断模块,用于判断注入的测试错误比特个数是否达到区间的最大值。
- [0152] 在一实施例中,,所述初始模型建立单元302包括:
- [0153] 平均值获取模块,用于对同一测试错误比特个数对应的测试校验子个数取平均值,以形成目标校验子个数;
- [0154] 关系曲线模块,用于根据测试错误比特个数的区间及对应的目标校验子个数,获取关系曲线,以得到初始模型。
- [0155] 在一实施例中,所述拟合单元303包括:
- [0156] 划分模块,用于根据关系曲线对目标校验子个数进行阈值区间划分;
- [0157] 参数确定模块,用于根据划分结果以及拟合公式采用最小方差进行分段拟合,确定影响因子和偏移值;
- [0158] 目标模型确定模块,用于根据影响因子和偏移值确定目标模型。
- [0159] 在一实施例中,所述比特个数估算单元305包括:
- [0160] 实际区间获取模块,用于获取实际不满足校验子个数所在的实际区间;
- [0161] 个数估计模块,用于利用实际区间的目标模型,根据实际不满足校验子个数估算实际错误比特个数。
- [0162] 需要说明的是,所属领域的技术人员可以清楚地了解到,上述错误比特个数估计装置300和各单元的具体实现过程,可以参考前述方法实施例中的相应描述,为了描述的方便和简洁,在此不再赘述。
- [0163] 上述错误比特个数估计装置300可以实现为一种计算机程序的形式,该计算机程序可以在如图12所示的计算机设备上运行。
- [0164] 请参阅图12,图12是本申请实施例提供的一种计算机设备的示意性框图。该计算机设备500是带有包含LDPC的ECC算法的固态硬盘内的服务器。

[0165] 参阅图12,该计算机设备500包括通过系统总线501连接的处理器502、存储器和网络接口505,其中,存储器可以包括非易失性存储介质503和内存储器504。

[0166] 该非易失性存储介质503可存储操作系统5031和计算机程序5032。该计算机程序5032包括程序指令,该程序指令被执行时,可使得处理器502执行一种错误比特个数估计方法。

[0167] 该处理器502用于提供计算和控制能力,以支撑整个计算机设备500的运行。

[0168] 该内存储器504为非易失性存储介质503中的计算机程序5032的运行提供环境,该计算机程序5032被处理器502执行时,可使得处理器502执行一种错误比特个数估计方法。

[0169] 该网络接口505用于与其它设备进行网络通信。本领域技术人员可以理解,图12中示出的结构,仅仅是与本申请方案相关的部分结构的框图,并不构成对本申请方案所应用于其上的计算机设备500的限定,具体的计算机设备500可以包括比图中所示更多或更少的部件,或者组合某些部件,或者具有不同的部件布置。

[0170] 其中,所述处理器502用于运行存储在存储器中的计算机程序5032,以实现如下步骤:

[0171] 统计设定区间内的测试错误比特个数以及对应的测试不满足校验子个数;

[0172] 根据统计结果建立初始模型;

[0173] 对初始模型进行拟合,形成目标模型;

[0174] 在使用时对输入译码器的码字计算实际不满足校验子个数;

[0175] 利用目标模型和实际不满足校验子个数估算实际错误比特个数。

[0176] 在一实施例中,处理器502在实现所述统计设定区间内的测试错误比特个数以及对应的测试不满足校验子个数步骤时,具体实现如下步骤:

[0177] 获取原始数据;

[0178] 对原始数据进行编码,获取测试码字;

[0179] 设定随机注入的测试错误比特个数的区间、步长以及最大统计次数;

[0180] 利用随机数产生对应的测试错误比特个数;

[0181] 将测试错误比特个数加入测试码字;

[0182] 根据测试码字计算测试不满足校验子个数;

[0183] 记录测试错误比特个数对应的测试不满足校验子个数;

[0184] 判断记录次数是否达到最大统计次数;

[0185] 若否,则返回所述利用随机数产生对应的测试错误比特步骤;

[0186] 若是,则根据步长重新设定注入的测试错误比特个数;

[0187] 判断注入的测试错误比特个数是否达到区间的最大值;

[0188] 若是,则进入所述根据测试错误比特个数以及对应的测试不满足校验子个数建立初始模型步骤;

[0189] 若否,则返回所述利用随机数产生对应的测试错误比特步骤。

[0190] 在一实施例中,处理器502在实现所述根据统计结果建立初始模型步骤时,具体实现如下步骤:

[0191] 对同一测试错误比特个数对应的测试校验子个数取平均值,以形成目标校验子个数;

[0192] 根据测试错误比特个数的区间及对应的目标校验子个数,获取关系曲线,以得到初始模型。

[0193] 在一实施例中,处理器502在实现所述对初始模型进行拟合,形成目标模型步骤时,具体实现如下步骤:

[0194] 根据关系曲线对目标校验子个数进行阈值区间划分;

[0195] 根据划分结果以及拟合公式采用最小方差进行分段拟合,确定影响因子和偏移值;

[0196] 根据影响因子和偏移值确定目标模型。

[0197] 其中,拟合公式包括: $s_{\mu}(x) = a \cdot x + b$ ;其中, $a > 0, b > 0$ ;a为比例影响因子,b为偏移值。

[0198] 在一实施例中,处理器502在实现所述利用目标模型和实际不满足校验子个数估算实际错误比特个数步骤时,具体实现如下步骤:

[0199] 获取实际不满足校验子个数所在的实际区间;

[0200] 利用实际区间的目标模型,根据实际不满足校验子个数估算实际错误比特个数。

[0201] 应当理解,在本申请实施例中,处理器502可以是中央处理单元(Central Processing Unit,CPU),该处理器502还可以是其他通用处理器、数字信号处理器(Digital Signal Processor,DSP)、专用集成电路(Application Specific Integrated Circuit,ASIC)、现成可编程门阵列(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、分立硬件组件等。其中,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。

[0202] 本领域普通技术人员可以理解的是实现上述实施例的方法中的全部或部分流程,是可以通过计算机程序来指令相关的硬件来完成。该计算机程序包括程序指令,计算机程序可存储于一存储介质中,该存储介质为计算机可读存储介质。该程序指令被该计算机系统中的至少一个处理器执行,以实现上述方法的实施例的流程步骤。

[0203] 因此,本发明还提供一种存储介质。该存储介质可以为计算机可读存储介质。该存储介质存储有计算机程序,其中该计算机程序被处理器执行时使处理器执行如下步骤:

[0204] 统计设定区间内的测试错误比特个数以及对应的测试不满足校验子个数;

[0205] 根据统计结果建立初始模型;

[0206] 对初始模型进行拟合,形成目标模型;

[0207] 在使用时对输入译码器的码字计算实际不满足校验子个数;

[0208] 利用目标模型和实际不满足校验子个数估算实际错误比特个数。

[0209] 在一实施例中,所述处理器在执行所述计算机程序而实现所述统计设定区间内的测试错误比特个数以及对应的测试不满足校验子个数步骤时,具体实现如下步骤:

[0210] 获取原始数据;

[0211] 对原始数据进行编码,获取测试码字;

[0212] 设定随机注入的测试错误比特个数的区间、步长以及最大统计次数;

[0213] 利用随机数产生对应的测试错误比特个数;

[0214] 将测试错误比特个数加入测试码字;

[0215] 根据测试码字计算测试不满足校验子个数;

- [0216] 记录测试错误比特个数对应的测试不满足校验子个数；
- [0217] 判断记录次数是否达到最大统计次数；
- [0218] 若否，则返回所述利用随机数产生对应的测试错误比特步骤；
- [0219] 若是，则根据步长重新设定注入的测试错误比特个数；
- [0220] 判断注入的测试错误比特个数是否达到区间的最大值；
- [0221] 若是，则进入所述根据测试错误比特个数以及对应的测试不满足校验子个数建立初始模型步骤；
- [0222] 若否，则返回所述利用随机数产生对应的测试错误比特步骤。
- [0223] 在一实施例中，所述处理器在执行所述计算机程序而实现所述根据统计结果建立初始模型步骤时，具体实现如下步骤：
- [0224] 对同一测试错误比特个数对应的测试校验子个数取平均值，以形成目标校验子个数；
- [0225] 根据测试错误比特个数的区间及对应的目标校验子个数，获取关系曲线，以得到初始模型。
- [0226] 在一实施例中，所述处理器在执行所述计算机程序而实现所述对初始模型进行拟合，形成目标模型步骤时，具体实现如下步骤：
- [0227] 根据关系曲线对目标校验子个数进行阈值区间划分；
- [0228] 根据划分结果以及拟合公式采用最小方差进行分段拟合，确定影响因子和偏移值；
- [0229] 根据影响因子和偏移值确定目标模型
- [0230] 其中，所述拟合公式包括： $s_u(x) = a \cdot x + b$ ；其中， $a > 0, b > 0$ ； $a$ 为比例影响因子， $b$ 为偏移值
- [0231] 在一实施例中，所述处理器在执行所述计算机程序而实现所述利用目标模型和实际不满足校验子个数估算实际错误比特个数步骤时，具体实现如下步骤：
- [0232] 获取实际不满足校验子个数所在的实际区间；
- [0233] 利用实际区间的目标模型，根据实际不满足校验子个数估算实际错误比特个数。
- [0234] 所述存储介质可以是U盘、移动硬盘、只读存储器(Read-Only Memory, ROM)、磁碟或者光盘等各种可以存储程序代码的计算机可读存储介质。
- [0235] 本领域普通技术人员可以意识到，结合本文中所公开的实施例描述的各示例的单元及算法步骤，能够以电子硬件、计算机软件或者二者的结合来实现，为了清楚地说明硬件和软件的可互换性，在上述说明中已经按照功能一般性地描述了各示例的组成及步骤。这些功能究竟以硬件还是软件方式来执行，取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能，但是这种实现不应认为超出本发明的范围。
- [0236] 在本发明所提供的几个实施例中，应该理解到，所揭露的装置和方法，可以通过其它的方式实现。例如，以上所描述的装置实施例仅仅是示意性的。例如，各个单元的划分，仅仅为一种逻辑功能划分，实际实现时可以有另外的划分方式。例如多个单元或组件可以结合或者可以集成到另一个系统，或一些特征可以忽略，或不执行。
- [0237] 本发明实施例方法中的步骤可以根据实际需要进行顺序调整、合并和删减。本发

明实施例装置中的单元可以根据实际需要进行合并、划分和删减。另外,在本发明各个实施例中的各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以是两个或两个以上单元集成在一个单元中。

[0238] 该集成的单元如果以软件功能单元的形式实现并作为独立的产品销售或使用时,可以存储在一个存储介质中。基于这样的理解,本发明的技术方案本质上或者说对现有技术做出贡献的部分,或者该技术方案的全部或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中,包括若干指令用以使得一台计算机设备(可以是个人计算机,终端,或者网络设备等)执行本发明各个实施例所述方法的全部或部分步骤。

[0239] 以上所述,仅为本发明的具体实施方式,但本发明的保护范围并不局限于此,任何熟悉本技术领域的技术人员在本发明揭露的技术范围内,可轻易想到各种等效的修改或替换,这些修改或替换都应涵盖在本发明的保护范围之内。因此,本发明的保护范围应以权利要求的保护范围为准。

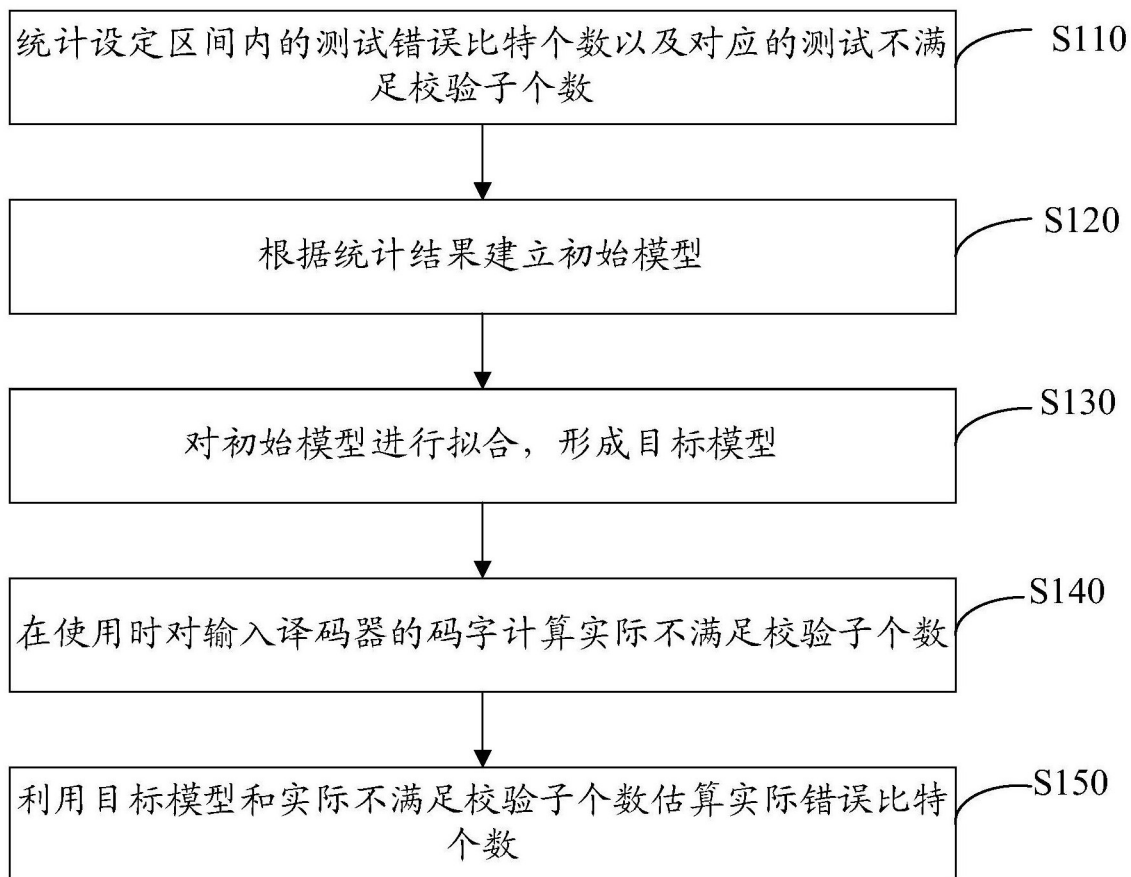


图1



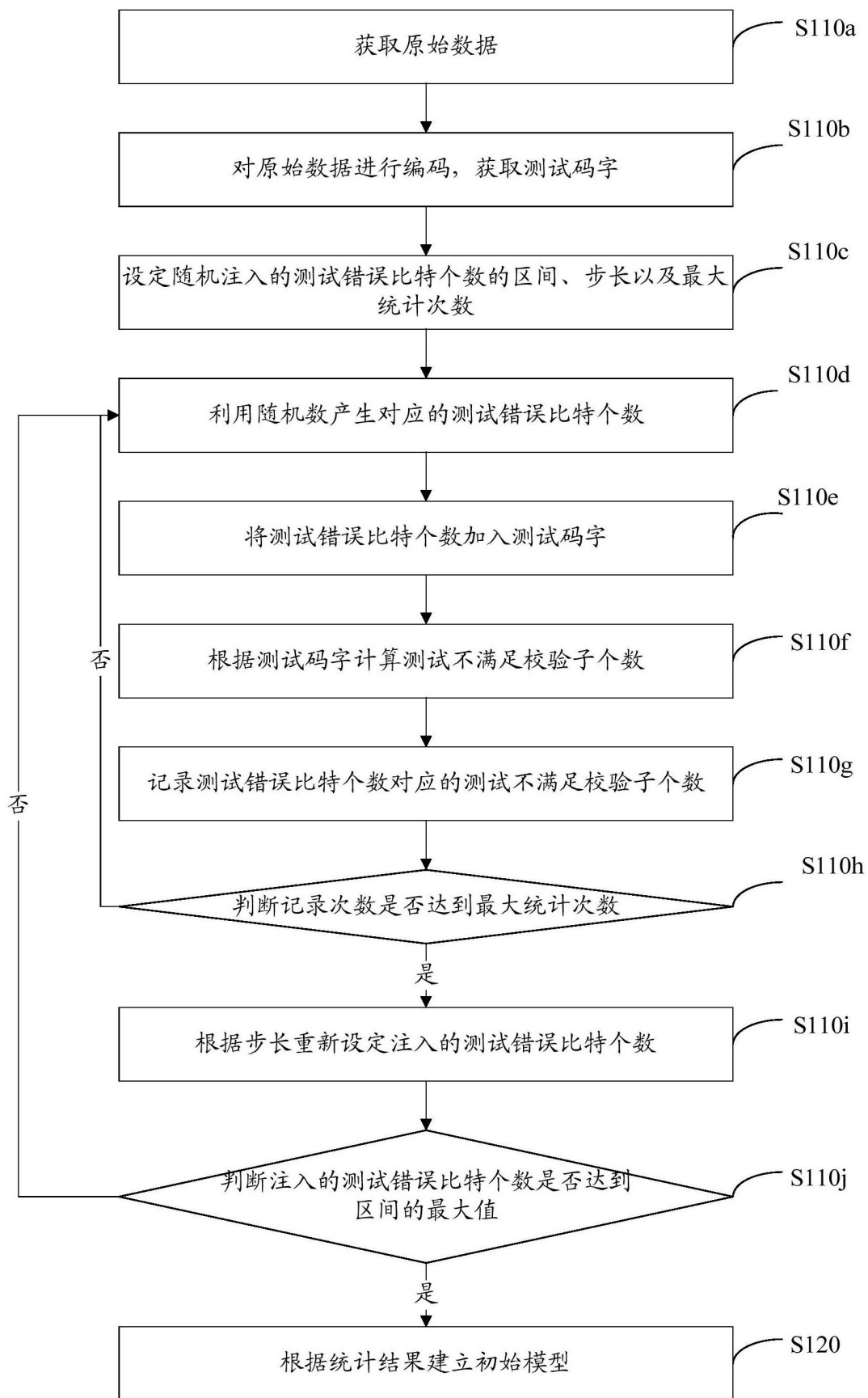


图2

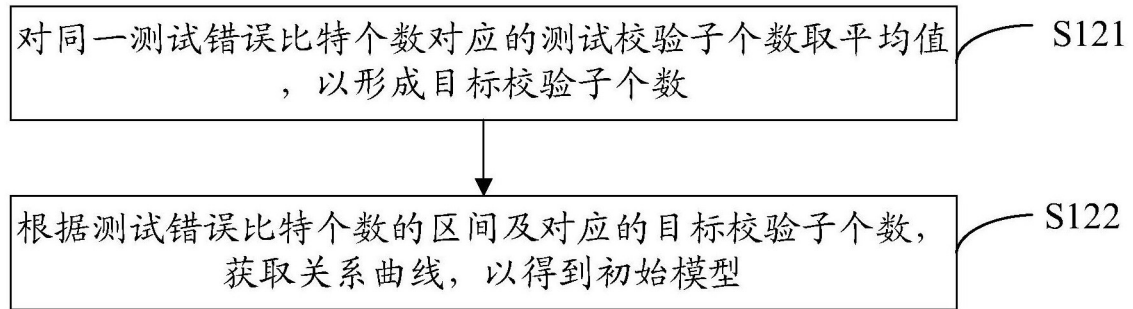


图3

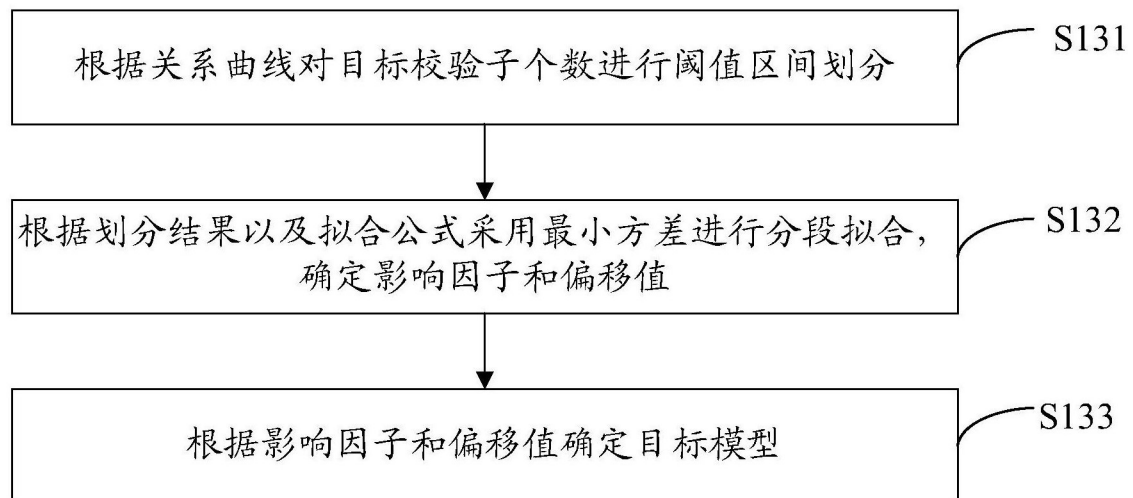


图4

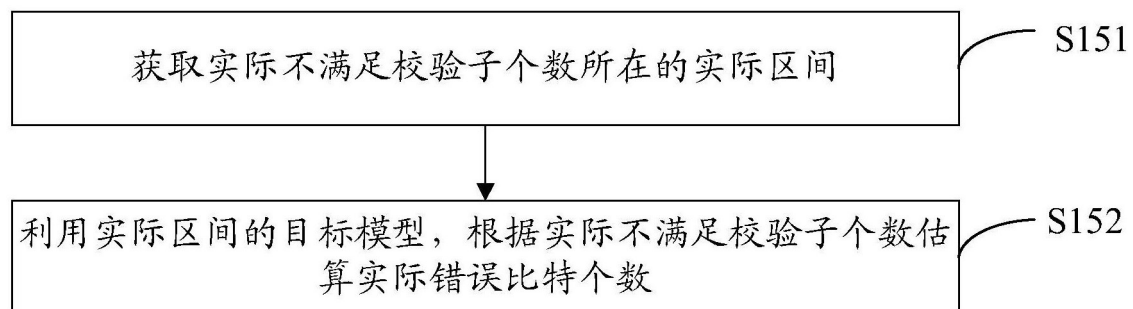


图5



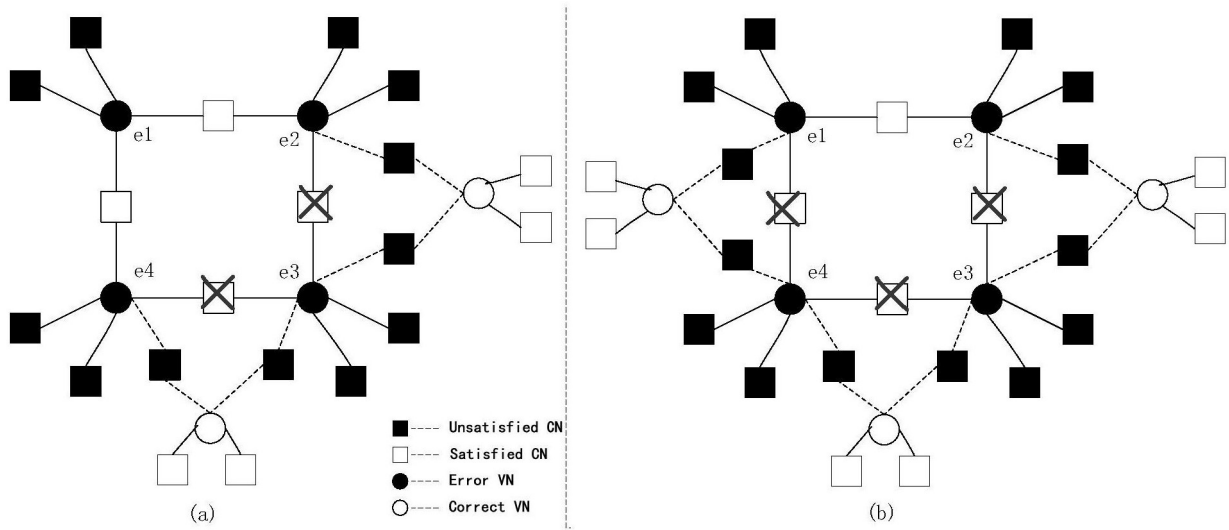


图8

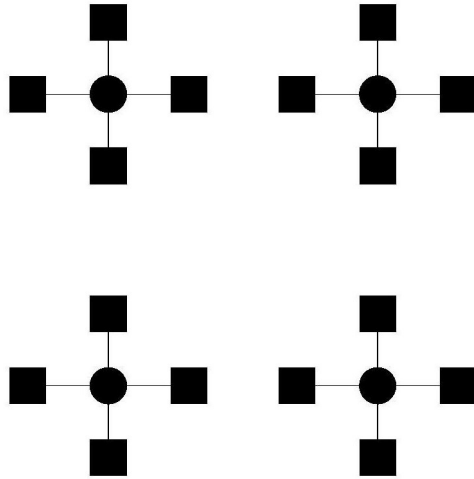


图9

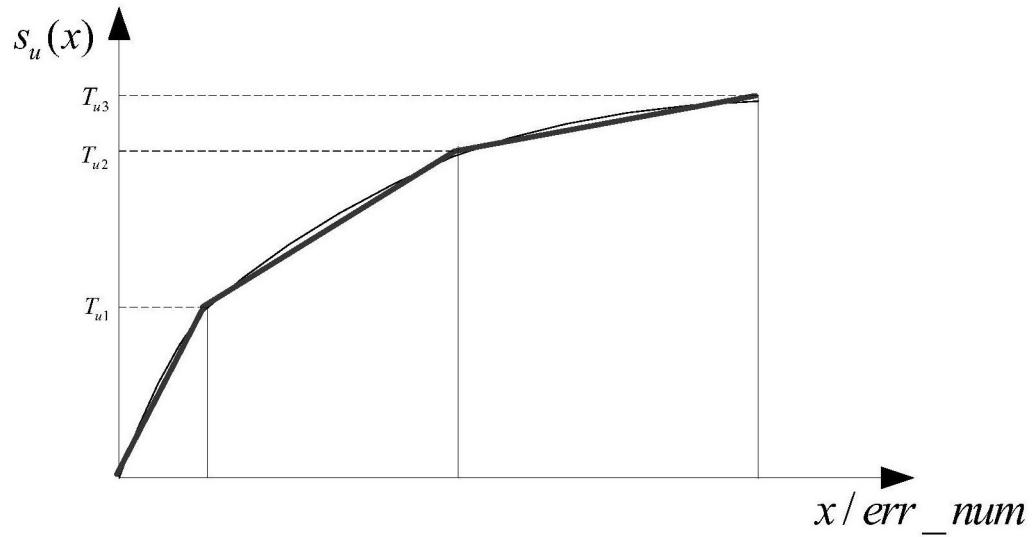


图10

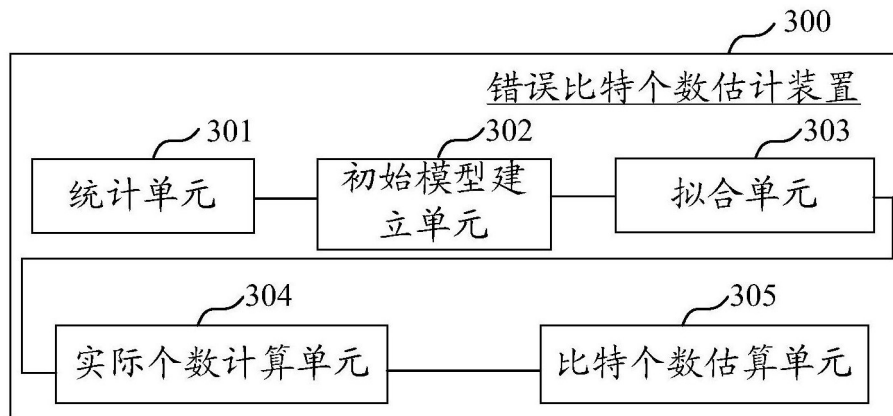


图11

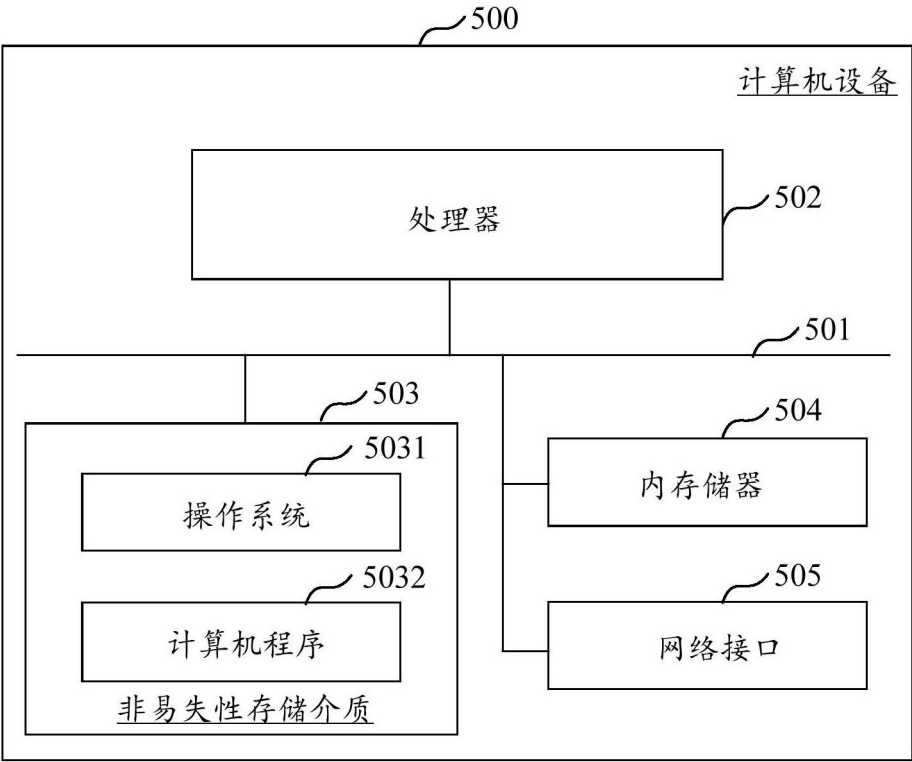


图12