



(12)发明专利申请

(10)申请公布号 CN 108900196 A

(43)申请公布日 2018. 11. 27

(21)申请号 201810690646.5

(22)申请日 2018.06.28

(71)申请人 郑州云海信息技术有限公司
地址 450018 河南省郑州市郑东新区心怡路278号16层1601室

(72)发明人 李龙 赵健 尹云峰

(74)专利代理机构 北京集佳知识产权代理有限公司 11227

代理人 罗满

(51) Int. Cl.
H03M 7/30(2006.01)

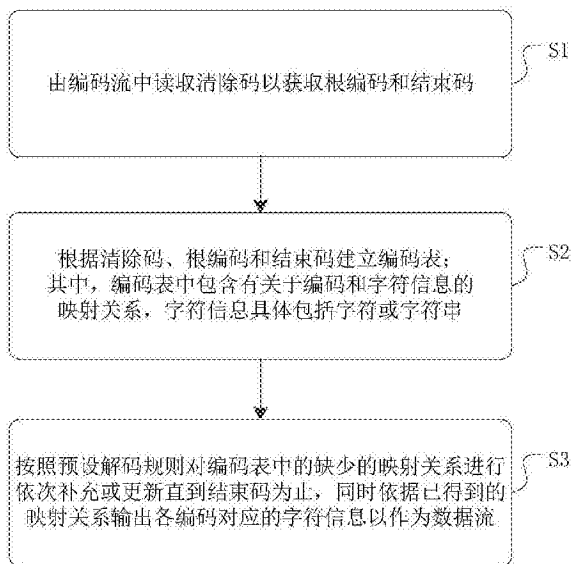
权利要求书2页 说明书10页 附图4页

(54)发明名称

一种基于LZW算法的数据解码方法、装置、设备及介质

(57)摘要

本发明公开了一种基于LZW算法的数据解码方法、装置、设备及介质,包括由编码流中读取清除码以获取根编码和结束码;根据清除码、根编码和结束码建立包含有关于编码和字符信息的映射关系的编码表;对缺少的映射关系进行依次补充或更新直到结束码为止,依据已得到的映射关系输出各编码对应的字符信息以作为数据流。通过编码表维护编码到字符信息的映射关系,以数据表的形式存储编码表,有效提升存取效率,同时由于映射关系中是依次补充的,随着读取编码的个数增加,映射关系也在不断增多,使得当前编码存在于编码表中的概率大大提高,不需要再重复计算其对应的字符信息,提高了解码效率。基于LZW算法的数据解码装置、设备及介质具有上述效果。



1. 一种基于LZW算法的数据解码方法,其特征在于,包括:

由编码流中读取清除码以获取根编码和结束码;

根据所述清除码、所述根编码和所述结束码建立编码表;其中,编码表中包含有关于编码和字符信息的映射关系,所述字符信息具体包括字符或字符串;

按照预设解码规则对编码表中的缺少的所述映射关系进行依次补充或更新直到所述结束码为止,同时依据已得到的所述映射关系输出各编码对应的字符信息以作为数据流。

2. 根据权利要求1所述的基于LZW算法的数据解码方法,其特征在于,所述按照预设解码规则对编码表中的缺少的所述映射关系进行依次补充或更新直到所述结束码为止,同时依据已得到的所述映射关系输出各编码对应的字符信息以作为数据流具体包括:

读取当前编码,依据编码表确定当前编码对应的字符信息以作为数据流输出;

设置当前OLD值为当前编码;

将下一个编码作为当前编码,并读取当前编码;

判断当前编码是否为所述清除码,如果是,则清除编码表,并返回所述根据所述清除码、所述根编码和所述结束码建立编码表的步骤以实现所述映射关系的更新;

如果否,则判断当前编码是否为所述结束码,如果是,则结束;

如果否,则判断编码表中是否存在当前编码;

如果存在,则将当前编码对应的字符信息作为数据流输出,并按照第一预设更新规则更新编码表以进行所述映射关系的补充;返回设置当前OLD值为当前编码的步骤;

如果不存在,则按照第二预设更新规则得到当前编码对应的字符信息以进行所述映射关系的补充,并将当前编码对应的字符信息作为数据流输出;返回设置当前OLD值为当前编码的步骤。

3. 根据权利要求2所述的基于LZW算法的数据解码方法,其特征在于,所述按照所述第一预设更新规则更新编码表具体为:

将当前编码对应的字符信息的首字符和当前OLD值对应的字符信息组成新字符信息顺序添加至编码表中得到一组新的映射关系。

4. 根据权利要求2所述的基于LZW算法的数据解码方法,其特征在于,所述按照第二预设更新规则得到当前编码对应的字符信息以进行所述映射关系的补充,并将当前编码对应的字符信息作为数据流输出具体包括:

将当前OLD值对应的字符信息和当前OLD值对应的字符信息的首字符组成新字符信息作为当前编码的字符信息,并作为数据流输出;

将当前编码和得到的新字符信息顺序添加至所述编码表中得到一组新的映射关系;返回设置当前OLD值为当前编码的步骤。

5. 根据权利要求1-4任意一项所述的基于LZW算法的数据解码方法,其特征在于,编码的起始值为0,编码的上限值为4095。

6. 根据权利要求5所述基于LZW算法的数据解码方法,其特征在于,还包括:保存最终得到的编码表。

7. 一种基于LZW算法的数据解码装置,其特征在于,包括:

编码表读取模块,用于由编码流中读取清除码以获取根编码和结束码;

编码表建立模块,用于根据所述清除码、所述根编码和所述结束码建立编码表;其中,

编码表中包含有关于编码和字符信息的映射关系,所述字符信息具体包括字符或字符串;

编码表操作模块,用于按照预设解码规则对编码表中的缺少的所述映射关系进行依次补充或更新直到所述结束码为止;

编码表输出模块,用于依据已得到的所述映射关系输出各编码对应的字符信息以作为数据流。

8.一种基于LZW算法的数据解码设备,其特征在于,包括存储器,用于存储计算机程序;

处理器,用于执行所述计算机程序时实现如权利要求1至6任一项所述的基于LZW算法的数据解码方法的步骤。

9.一种计算机可读存储介质,其特征在于,所述计算机可读存储介质上存储有计算机程序,所述计算机程序被处理器执行时实现如权利要求1至6任一项所述的基于LZW算法的数据解码方法的步骤。

一种基于LZW算法的数据解码方法、装置、设备及介质

技术领域

[0001] 本发明涉及LZW算法技术领域,特别是涉及一种基于LZW算法的数据解码方法、装置、设备及介质。

背景技术

[0002] LZW算法又叫“串表压缩算法”,通过建立一个字符串表,用较短的代码来表示较长的字符串来,从而实现数据压缩。LZW算法包含三个重要的对象:数据流、编码流和编码表。在进行数据压缩过程中,数据流(原始数据)是输入对象,编码流(经过压缩的数据)是输出对象;与之相反,在进行数据解压缩过程中,编码流是输入对象,数据流则是输出对象。编码表是在数据压缩和数据解压缩过程中都需要借助的对象,用于表示较长的字符串到较短代码的映射。

[0003] 现有技术中采用数据栈的方式动态获取数据流中每个编码所代表的实际字符串。包含两个数组prefix和suffix,每个数组的大小均为4096(即,其可以存储4096的数据值)。其中,数组值suffix[i]表示编码i所代表的字符(串)的最后一个字符,数组值prefix[i]表示编码i的前缀编码。对于根条目i,prefix[i]表示数据i本身,即prefix[i]=i。假设编码表中索引值为i的表项对应的字符串由索引值为p对应的字符(串)和字符k组成,则prefix[i]的值为p,即prefix[i]=p。设置临时变量table_len,表示新的编码表表项的索引值,其初始值为清除码的值+1。设置可以存储4096个数组值的临时数组stack,idx表示数组索引,idx的初始值为0。

[0004] LZW算法的解压缩过程如下:

[0005] 读取第一个编码,为清除码,设置OLD值(上一次读取的编码)为清除码;对于读取的每个编码,执行如下操作:设置prefix[table_len]的值为[[OLD]];根据情况设置suffix[table_len]的值:如果当前编码值[code]小于table_len,则设置suffix[table_len]的值为当前编码对应字符串的首字符;否则,设置suffix[table_len]为[OLD]对应字符串所对应的第一个字符。更改当前编码的值cur:如果当前编码值小于table_len,则当前编码值不变;否则设置当前编码值为[OLD]。迭代获取编码所表示的数据流:如果当前编码大于清除码,循环设置stack[idx]的值为suffix[cur]的值,并将idx值+1,同时设置cur的值为prefix[cur];设置stack[idx]的值为cur的值;table_len的值加1,设置[OLD]的值为读取的当前编码值[code]。最后,逆序输出stack中的值到数据流。

[0006] 综上所述,传统的解码方法,采用数据栈的方案动态获取数据流中每个编码所代表的实际字符串,导致解码过程中对每个编码均需要调用一次这种迭代过程,导致数据解码效率较低。

发明内容

[0007] 本发明的目的是提供一种基于LZW算法的数据解码方法、装置、设备及介质,用于克服在解码过程中解码效率较低的问题。

- [0008] 为解决上述技术问题,本发明提供一种基于LZW算法的数据解码方法,包括:
- [0009] 由编码流中读取清除码以获取根编码和结束码;
- [0010] 根据所述清除码、所述根编码和所述结束码建立编码表;其中,编码表中包含有关于编码和字符信息的映射关系,所述字符信息具体包括字符或字符串;
- [0011] 按照预设解码规则对编码表中的缺少的所述映射关系进行依次补充或更新直到所述结束码为止,同时依据已得到的所述映射关系输出各编码对应的字符信息以作为数据流。
- [0012] 优选地,所述按照预设解码规则对编码表中的缺少的所述映射关系进行依次补充或更新直到所述结束码为止,同时依据已得到的所述映射关系输出各编码对应的字符信息以作为数据流具体包括:
- [0013] 读取当前编码,依据编码表确定当前编码对应的字符信息以作为数据流输出;
- [0014] 设置当前OLD值为当前编码;
- [0015] 将下一个编码作为当前编码,并读取当前编码;
- [0016] 判断当前编码是否为所述清除码,如果是,则清除编码表,并返回所述根据所述清除码、所述根编码和所述结束码建立编码表的步骤以实现所述映射关系的更新;
- [0017] 如果否,则判断当前编码是否为所述结束码,如果是,则结束;
- [0018] 如果否,则判断编码表中是否存在当前编码;
- [0019] 如果存在,则将当前编码对应的字符信息作为数据流输出,并按照第一预设更新规则更新编码表以进行所述映射关系的补充;返回设置当前OLD值为当前编码的步骤;
- [0020] 如果不存在,则按照第二预设更新规则得到当前编码对应的字符信息以进行所述映射关系的补充,并将当前编码对应的字符信息作为数据流输出;返回设置当前OLD值为当前编码的步骤。
- [0021] 优选地,所述按照所述第一预设更新规则更新编码表具体为:
- [0022] 将当前编码对应的字符信息的首字符和当前OLD值对应的字符信息组成新字符信息顺序添加至编码表中得到一组新的映射关系。
- [0023] 优选地,所述按照第二预设更新规则得到当前编码对应的字符信息以进行所述映射关系的补充,并将当前编码对应的字符信息作为数据流输出具体包括:
- [0024] 将当前OLD值对应的字符信息和当前OLD值对应的字符信息的首字符组成新字符信息作为当前编码的字符信息,并作为数据流输出;
- [0025] 将当前编码和得到的新字符信息顺序添加至所述编码表中得到一组新的映射关系;返回设置当前OLD值为当前编码的步骤。
- [0026] 优选地,编码的起始值为0,编码的上限值为4095。
- [0027] 优选地,还包括:保存最终得到的编码表。
- [0028] 为解决上述技术问题,本发明提供一种基于LZW算法的数据解码装置,包括:
- [0029] 编码表读取模块,用于由编码流中读取清除码以获取根编码和结束码;
- [0030] 编码表建立模块,用于根据所述清除码、所述根编码和所述结束码建立编码表;其中,编码表中包含有关于编码和字符信息的映射关系,所述字符信息具体包括字符或字符串;
- [0031] 编码表操作模块,用于按照预设解码规则对编码表中的缺少的所述映射关系进行

依次补充或更新直到所述结束码为止；

[0032] 编码表输出模块,用于依据已得到的所述映射关系输出各编码对应的字符信息以作为数据流。

[0033] 为解决上述技术问题,本发明提供一种基于LZW算法的数据解码设备,包括存储器,用于存储计算机程序；

[0034] 处理器,用于执行所述计算机程序时实现如上述所述的基于LZW算法的数据解码方法的步骤。

[0035] 为解决上述技术问题,本发明提供一种计算机可读存储介质,所述计算机可读存储介质上存储有计算机程序,所述计算机程序被处理器执行时实现如上述所述的基于LZW算法的数据解码方法的步骤。

[0036] 本发明所提供的基于LZW算法的数据解码方法,包括由编码流中读取清除码以获取根编码和结束码;根据清除码、根编码和结束码建立编码表;其中,编码表中包含有关于编码和字符信息的映射关系,字符信息具体包括字符或字符串;按照预设解码规则对编码表中的缺少的映射关系进行依次补充或更新直到结束码为止,同时依据已得到的映射关系输出各编码对应的字符信息以作为数据流。由此可见,本方法中通过编码表维护编码到其所代表实际字符信息的映射关系,以数据表的形式存储编码表,从而有效提升数据的存取效率,更为重要的是,由于编码表中的映射关系中是依次补充的,随着读取编码的个数增加,映射关系也在不断增多,使得当前编码存在于编码表中的概率大大提高,只要当前编码在编码表中,则不需要再重复计算其对应的字符信息,也就大大降低了迭代的过程,提高了解码效率。

[0037] 此外,本发明所提供的基于LZW算法的数据解码装置、设备及介质均与上述方法对应,因此,亦具有上述有益效果。

附图说明

[0038] 为了更清楚地说明本发明实施例,下面将对实施例中所需要使用的附图做简单的介绍,显而易见地,下面描述中的附图仅仅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。

[0039] 图1为本发明实施例提供的一种基于LZW算法的数据解码方法的流程图；

[0040] 图2为本发明实施例提供的另一种基于LZW算法的数据解码方法的流程图；

[0041] 图3为本发明实施例提供的另一种基于LZW算法的数据解码方法的流程图；

[0042] 图4为本发明实施例提供的一种基于LZW算法的数据解码装置的结构图。

具体实施方式

[0043] 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,所描述的实施例仅仅是本发明一部分实施例,而不是全部实施例。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,都属于本发明保护范围。

[0044] 本发明的核心是提供一种基于LZW算法的数据解码方法、装置、设备及介质,用于克服在解码过程中解码效率较低的问题。

[0045] 为了使本技术领域的人员更好地理解本发明方案,下面结合附图和具体实施方式对本发明作进一步的详细说明。

[0046] 图1为本发明实施例提供的一种基于LZW算法的数据解码方法的流程图。如图1所示,该方法包括:

[0047] S1:由编码流中读取清除码以获取根编码和结束码。

[0048] 需要说明的是,本发明实施例中提到的编码流是待解压的编码流,编码流中包含有经过LZW算法压缩得到的多个编码。多个编码中包含有清除码、根编码、结束码以及普通的编码。可以理解的是,对于一个编码流来说,通常情况下,第一个编码为清除码,清除码的编码值+1就是结束码,清除码的编码值之前的编码值是根编码。

[0049] 为了便于说明,本发明中以一个具体的编码流进行说明,该编码流具体为:#4#1#6#6#2#9#9#7#8#10#2#12#1#14#15#6#0#21#0#10#7#22#23#18#26#7#10#29#13#24#12#18#16#36#12#5,其中“#”表示数值,每两个相邻的编码之间用空格表示,只是为了区分两个编码,并无实际含义。对于该编码流来说,第一个编码为#4,即清除码为#4,结束码为#5,根编码为#0-#3。

[0050] S2:根据清除码、根编码和结束码建立编码表;其中,编码表中包含有关于编码和字符信息的映射关系,字符信息具体包括字符或字符串。

[0051] 需要说明的是,编码表随着解压的过程是会变化的。对于S2来说,该步骤中的编码表其实是初始编码表,但是编码表中无论怎么变化,都包含有每个编码和字符信息的映射关系,也就是说,一个编码对应一个字符信息。由于编码的变化,对应的字符信息变化,从而导致编码表的变化。由于有的编码的字符信息为单个字符,简称字符,而有的编码的字符信息为字符串,即多个字符构成,因此,本发明统一用字符信息表示。可以理解的是,字符信息中,至少是包括一个字符。

[0052] 通常情况下,在得到清除码、根编码和结束码之后,是依据这三种码对应的数值作为编码表中的表项值,此时各编码对应的字符信息就是编码本身,且每个表项值对应的表项索引(具体的编码)也与表项值相同。可以理解的是,由于还没有实质读取编码流,因此,对于结束码之前的各表项值而言,是固定的,不作为数据流输出,待真正读取编码后,表项值一栏中就是编码对应的字符信息,而表项索引是依次递增的。这里的表项索引仅用于示例说明,在实际存储中不需要存储该索引值。表1为步骤S2中的编码表的结构。

[0053] 表1

[0054]

表项索引	表项值
#0	#0
#1	#1
#2	#2
#3	#3
#4	#4
#5	#5

[0055] S3:按照预设解码规则对编码表中的缺少的映射关系进行依次补充或更新直到结束码为止,同时依据已得到的映射关系输出各编码对应的字符信息以作为数据流。

[0056] 由于第一个编码为清除码、第二个编码用于得到下一个编码所用的OLD值,因此,从第三个编码开始,每读取一个编码,则对应的编码表中的表项索引就会递增一个,因此,编码表中的编码与字符信息的映射关系会不断增加,直到读到结束码为止,或者读到清除码时,清除整个编码表,则编码表中存在的映射关系就会更新。

[0057] 图2为本发明实施例提供的另一种基于LZW算法的数据解码方法的流程图。作为优选地实施方式,步骤S3具体包括:

[0058] S30:读取当前编码,依据编码表确定当前编码对应的字符信息以作为数据流输出。

[0059] 需要说明的是,在LZW算法中,编码流的第二个编码是在#0-清除码之间的,如上文举例的编码流,第二个编码为#1,由于编码#4已经读取,因此,本步骤中的当前编码就是#1。由于当前编码表中已经存在该编码对应的字符信息,即仍然为#1,因此,该编码的数据流就是#1,也是首次输出数据流。

[0060] S31:设置当前OLD值为当前编码。

[0061] 本步骤中当前编码还是#1,因此,当前OLD值就是当前OLD值就是#1。

[0062] S32:将下一个编码作为当前编码,并读取当前编码。

[0063] 上文中举例的编码流,第三个编码为#6,也是本步骤中所说的当前编码。

[0064] S33:判断当前编码是否为清除码,如果是,则清除编码表,进入S2;如果否,则进入S34。

[0065] 根据例子中的编码流可知,本步骤中所指的当前编码为#6,不是清除码。

[0066] S34:判断当前编码是否为结束码,如果是,则结束;如果否,则进入S35。

[0067] 根据例子中的编码流可知,本步骤中所指的当前编码为#6,不是结束码。

[0068] S35:判断编码表中是否存在当前编码;如果存在,则进入S36,否则,进入S37。

[0069] 编码表中表项索引中的编码是从#0-#5,并不存在#6,因此进入S37。

[0070] S37:按照第二预设更新规则得到当前编码对应的字符信息以进行映射关系的补充,并将当前编码对应的字符信息作为数据流输出;返回设置当前OLD值为当前编码的步骤。

[0071] 编码表中,不存在当前编码,需要补充一组映射关系,如步骤S3中提到的,映射关系需要依次补充,也就是说,编码表中的表项索引需要依次增加,当前表项索引是#5,增加一个就是#6,即当前编码表如表2所示。按照预设更新规则更新了编码表,则当前编码对应的字符信息就确定了,需要说明的是,如何确定当前编码对应的字符信息可以按照如下方式进行。图3为本发明实施例提供的另一种基于LZW算法的数据解码方法的流程图。

[0072] 步骤S37,具体分为子步骤S370和S371,分别为:

[0073] S370:将当前OLD值对应的字符信息和当前OLD值对应的字符信息的首字符组成新字符信息作为当前编码的字符信息,并作为数据流输出。

[0074] S371:将当前编码和得到的新字符信息顺序添加至编码表中得到一组新的映射关系;返回设置当前OLD值为当前编码的步骤。

[0075] 步骤S370中,当前OLD值为#1,当前OLD值对应的字符信息的首字符也为#1,二者组合就是#1#1,因此,编码#6对应的字符信息就是#1#1。在上一次数据流的基础上,本次数据流输出的就是#1#1,为第二次输出数据流。

[0076] 表2

[0077]

表项索引	表项值
#0	#0
#1	#1
#2	#2
#3	#3
#4	#4
#5	#5
#6	#1#1

[0078] 上述步骤S37之后需要返回步骤S31,即重复执行S31-S35。

[0079] S31:设置当前OLD值为当前编码。

[0080] 根据例子中的编码流可知,本步骤中所指的当前编码是#6,因此,当前OLD值就是当前OLD值就是#6。

[0081] S32:将下一个编码作为当前编码,并读取当前编码。

[0082] 上文中举例的编码流,第四个编码为#6,也是本步骤中所说的当前编码。

[0083] S33:判断当前编码是否为清除码,如果是,则清除编码表,进入S2;如果否,则进入S34。

[0084] 根据例子中的编码流可知,本步骤中所指的当前编码为#6,不是清除码。

[0085] S34:判断当前编码是否为结束码,如果是,则结束;如果否,则进入S35。

[0086] 根据例子中的编码流可知,本步骤中所指的当前编码为#6,不是结束码。

[0087] S35:判断编码表中是否存在当前编码;如果存在,则进入S36,否则,进入S37。

[0088] 由于第三个编码也是#6,因此,对于当前编码#6来说,编码表中存在该编码,进入到步骤S36。

[0089] S36:将当前编码对应的字符信息作为数据流输出,并按照第一预设更新规则更新编码表以进行映射关系的补充;返回设置当前OLD值为当前编码的步骤。

[0090] 需要说明的是,如果编码表中已经存在当前编码,则该编码对应的字符信息是不变的,即步骤中的编码#6对应的字符信息依然为#1#1。在上一次数据流的基础上,本次数据流输出的就是#1#1,为第三次输出数据流。

[0091] 如步骤S3中提到的,映射关系需要依次补充或更新,也就是说,如果编码流中只有一个清除码,则编码表中的表项索引需要依次增加。S35具体分为两个子步骤,分别为S360和S361。

[0092] S360:将当前编码对应的字符信息作为数据流输出;

[0093] S361:将当前编码对应的字符信息的首字符和当前OLD值对应的字符信息组成新字符信息顺序添加至编码表中得到一组新的映射关系。返回设置当前OLD值为当前编码的步骤。

[0094] 当前编码对应的字符信息为#1#1,其首字符为#1,当前OLD值为#6,对应的字符信息为#1#1,因此,本步骤中得到的新字符信息就是#1#1#1。当前表项索引是#6,增加一个就是#7,即当前编码表如表3所示。也就是说,表3中得到一组新的映射关系,即编码7对应的字

符信息为#1#1#1。

[0095] 表3

[0096]

表项索引	表项值
#0	#0
#1	#1
#2	#2
#3	#3
#4	#4
#5	#5
#6	#1#1
#7	#1#1#1

[0097] 上述步骤S36之后需要返回步骤S31,即重复执行S31-S35。下一个读取的编码为#2,存在于编码表中,对应的字符信息为#2,在上一次数据流的基础上,本次输出的数据流为#2,并且按照S36的步骤,添加的一组新的映射关系就是编码#8和对应的字符信息#2#1#1(将当前编码#2对应的字符信息的首字符#2和当前OLD值#6对应的字符信息#1#1组成新字符信息)。按照上述步骤,读取给出的编码表中的各编码,不断更新编码表,部分映射关系,如表4所示。

[0098] 表4

[0099]

表项索引	表项值
#0	#0
#1	#1
#2	#2
#3	#3
#4	#4
#5	#5
#6	#1#1
#7	#1#1#1
#8	#2#1#1
#9	#2#2
#10	#2#2#2
#11	#1#1#1#1

[0100] 本发明实施例提供的基于LZW算法的数据解码方法,包括由编码流中读取清除码以获取根编码和结束码;根据清除码、根编码和结束码建立编码表;其中,编码表中包含有关于编码和字符信息的映射关系,字符信息具体包括字符或字符串;按照预设解码规则对编码表中的缺少的映射关系进行依次补充或更新直到结束码为止,同时依据已得到的映射关系输出各编码对应的字符信息以作为数据流。由此可见,本方法中通过编码表维护编码到其所代表实际字符信息的映射关系,以数据表的形式存储编码表,从而有效提升数据的

存取效率,更为重要的是,由于编码表中的映射关系中是依次补充的,随着读取编码的个数增加,映射关系也在不断增多,使得当前编码存在于编码表中的概率大大提高,只要当前编码在编码表中,则不需要再重复计算其对应的字符信息,也就大大降低了迭代的过程,提高了解码效率。

[0101] 作为一种优选地实施方式,编码的起始值为0,编码的上限值为4095。

[0102] 编码表中每一项字符串的大小不超过12位,这样编码表有 $2^{12}=4096$ 项,值为0~4095。在具体实施中,编码表中的映射关系虽然是在读取编码之后补充的,但是并不影响预先建立编码表,也就是说,可以在建立编码表的时候可以建立一个最大的编码表,表项索引从0至4095,后续随着读取编码再添加对应的字符信息。

[0103] 作为一种优选地实施方式,还包括:保存最终得到的编码表。

[0104] 考虑到编码表的更新过程中,涉及到一些计算,如果下一个编码流与本次编码流相同的话,则直接调用本次编码流得到的编码表即可,通过编码表中的字符信息实现对下一个编码流的解码。在具体实施中,可以将相同的编码流设置相同的标签,这样在得到一个编码流时,通过标签能够快速检索出是否存在相同的编码流已经进行了解压,如果存在的话,可以直接获取对应数据流,从而快速解压。

[0105] 上文中对于基于LZW算法的数据解码方法的实施例进行了详细说明,本发明还提供与该方法对应的基于LZW算法的数据解码装置。该装置具体是从功能模块的角度进行说明。图4为本发明实施例提供的一种基于LZW算法的数据解码装置的结构图。如图4所示,该装置具体包括:

[0106] 编码表读取模块10,用于由编码流中读取清除码以获取根编码和结束码;

[0107] 编码表建立模块11,用于根据清除码、根编码和结束码建立编码表;其中,编码表中包含有关于编码和字符信息的映射关系,字符信息具体包括字符或字符串;

[0108] 编码表操作模块12,用于按照预设解码规则对编码表中的缺少的映射关系进行依次补充或更新直到结束码为止;

[0109] 编码表输出模块13,用于依据已得到的映射关系输出各编码对应的字符信息以作为数据流。

[0110] 由于装置部分的实施例与方法部分的实施例相互对应,因此装置部分的实施例请参见方法部分的实施例的描述,这里暂不赘述。

[0111] 本实施例提供的基于LZW算法的数据解码装置,包括:编码表读取模块由编码流中读取清除码以获取根编码和结束码;编码表建立模块根据清除码、根编码和结束码建立编码表;其中,编码表中包含有关于编码和字符信息的映射关系,字符信息具体包括字符或字符串;编码表操作模块按照预设解码规则对编码表中的缺少的映射关系进行依次补充直到结束码为止,编码表输出模块依据已得到的映射关系输出各编码对应的字符信息以作为数据流。由此可见,本装置中通过编码表维护编码到其所代表实际字符信息的映射关系,以数据表的形式存储编码表,从而有效提升数据的存取效率,更为重要的是,由于编码表中的映射关系中是依次补充的,随着读取编码的个数增加,映射关系也在不断增多,使得当前编码存在于编码表中的概率大大提高,只要当前编码在编码表中,则不需要再重复计算其对应的字符信息,也就大大降低了迭代的过程,提高了解码效率。

[0112] 此外,本发明还提供一种与上述方法对应的基于LZW算法的数据解码设备,与基于

LZW算法的数据解码装置不同的是,本实施例主要从硬件角度说明。该设备包括存储器,用于存储计算机程序;

[0113] 处理器,用于执行计算机程序时实现如上述实施例所述的基于LZW算法的数据解码方法的步骤。

[0114] 可以理解的是,本实施例中的存储器和处理器可以单独添加至服务器,也可以是服务器自身的存储器和处理器。在具体实施中,处理器和存储器可通过总线或其它方式连接。

[0115] 由于装置部分的实施例与方法部分的实施例相互对应,因此装置部分的实施例请参见方法部分的实施例的描述,这里暂不赘述。

[0116] 本实施例提供的基于LZW算法的数据解码设备包括存储器和处理器,处理器在执行过程中能够实现如下方法,由编码流中读取清除码以获取根编码和结束码;根据清除码、根编码和结束码建立编码表;其中,编码表中包含有关于编码和字符信息的映射关系,字符信息具体包括字符或字符串;按照预设解码规则对编码表中的缺少的映射关系进行依次补充或更新直到结束码为止,同时依据已得到的映射关系输出各编码对应的字符信息以作为数据流。由此可见,本方法中通过编码表维护编码到其所代表实际字符信息的映射关系,以数据表的形式存储编码表,从而有效提升数据的存取效率,更为重要的是,由于编码表中的映射关系中是依次补充的,随着读取编码的个数增加,映射关系也在不断增多,使得当前编码存在于编码表中的概率大大提高,只要当前编码在编码表中,则不需要再重复计算其对应的字符信息,也就大大降低了迭代的过程,提高了解码效率。

[0117] 在上述实施例的基础上,本发明还提供一种计算机可读存储介质该介质存储有计算机程序,计算机程序被处理器执行时实现如上述实施例所述的基于LZW算法的数据解码方法的步骤。

[0118] 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,也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现,也可以采用软件功能单元的形式实现。

[0119] 所述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,可以存储在一个计算机可读存储介质中。基于这样的理解,本实施例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,在读取该介质时执行本发明各个实施例所述方法的全部或部分步骤。而前述的存储介质包括:U盘、移动硬盘、只读存储器(Read-OnlyMemory,ROM)、随机存取存储器(Random Access Memory, RAM)、磁碟或者光盘等各种可以存储程序代码的介质。

[0120] 本实施例提供的计算机存储介质,存储有相应的计算机程序能够实现上述实施例所述的基于LZW算法的数据解码方法。本方法中通过编码表维护编码到其所代表实际字符信息的映射关系,以数据表的形式存储编码表,从而有效提升数据的存取效率,更为重要的是,由于编码表中的映射关系中是依次补充的,随着读取编码的个数增加,映射关系也在不断增多,使得当前编码存在于编码表中的概率大大提高,只要当前编码在编码表中,则不需要再重复计算其对应的字符信息,也就大大降低了迭代的过程,提高了解码效率。

[0121] 以上对本发明所提供的基于LZW算法的数据解码方法、装置、设备及介质进行了详

细介绍。说明书中各个实施例采用递进的方式描述,每个实施例重点说明的都是与其他实施例的不同之处,各个实施例之间相同相似部分互相参见即可。对于实施例公开的装置而言,由于其与实施例公开的方法相对应,所以描述的比较简单,相关之处参见方法部分说明即可。应当指出,对于本技术领域的普通技术人员来说,在不脱离本发明原理的前提下,还可以对本发明进行若干改进和修饰,这些改进和修饰也落入本发明权利要求的保护范围内。

[0122] 还需要说明的是,在本说明书中,诸如第一和第二等之类的关系术语仅仅用来将一个实体或者操作与另一个实体或操作区分开来,而不一定要求或者暗示这些实体或操作之间存在任何这种实际的关系或者顺序。而且,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、物品或者设备不仅包括那些要素,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、物品或者设备所固有的要素。在没有更多限制的情况下,由语句“包括一个……”限定的要素,并不排除在包括所述要素的过程、方法、物品或者设备中还存在另外的相同要素。

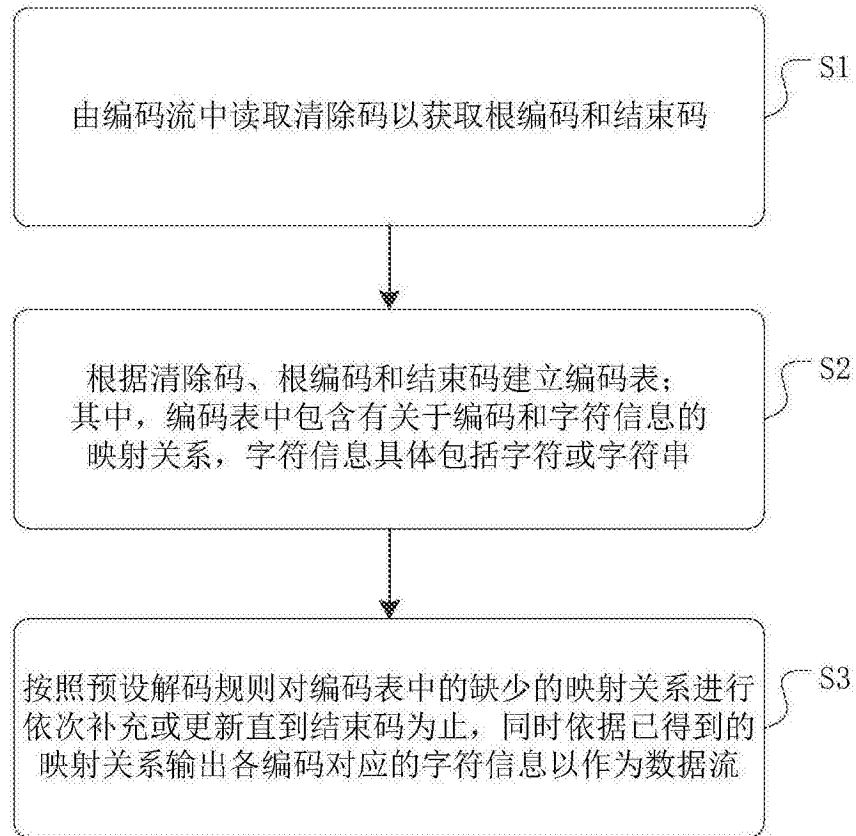


图1

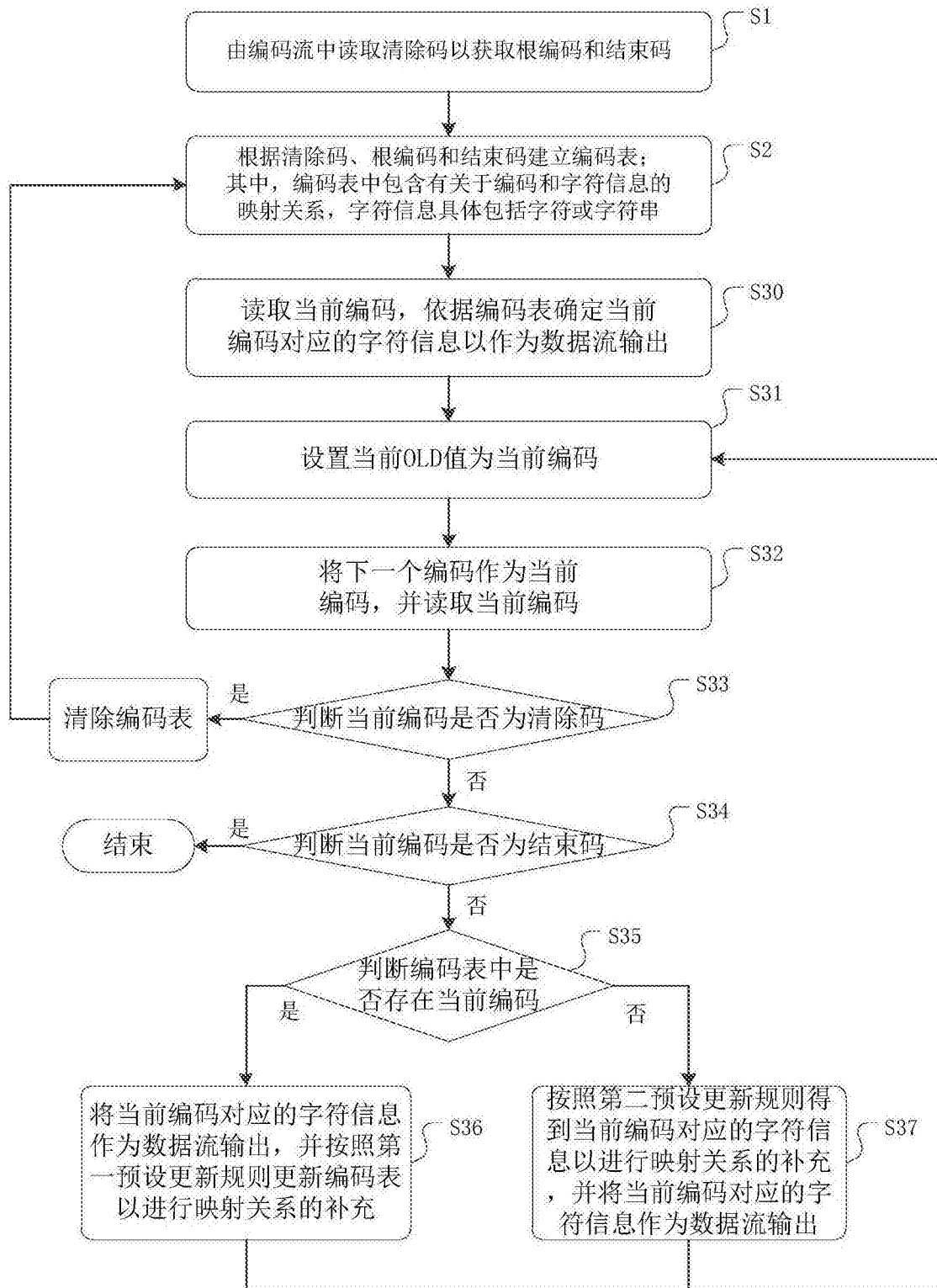


图2

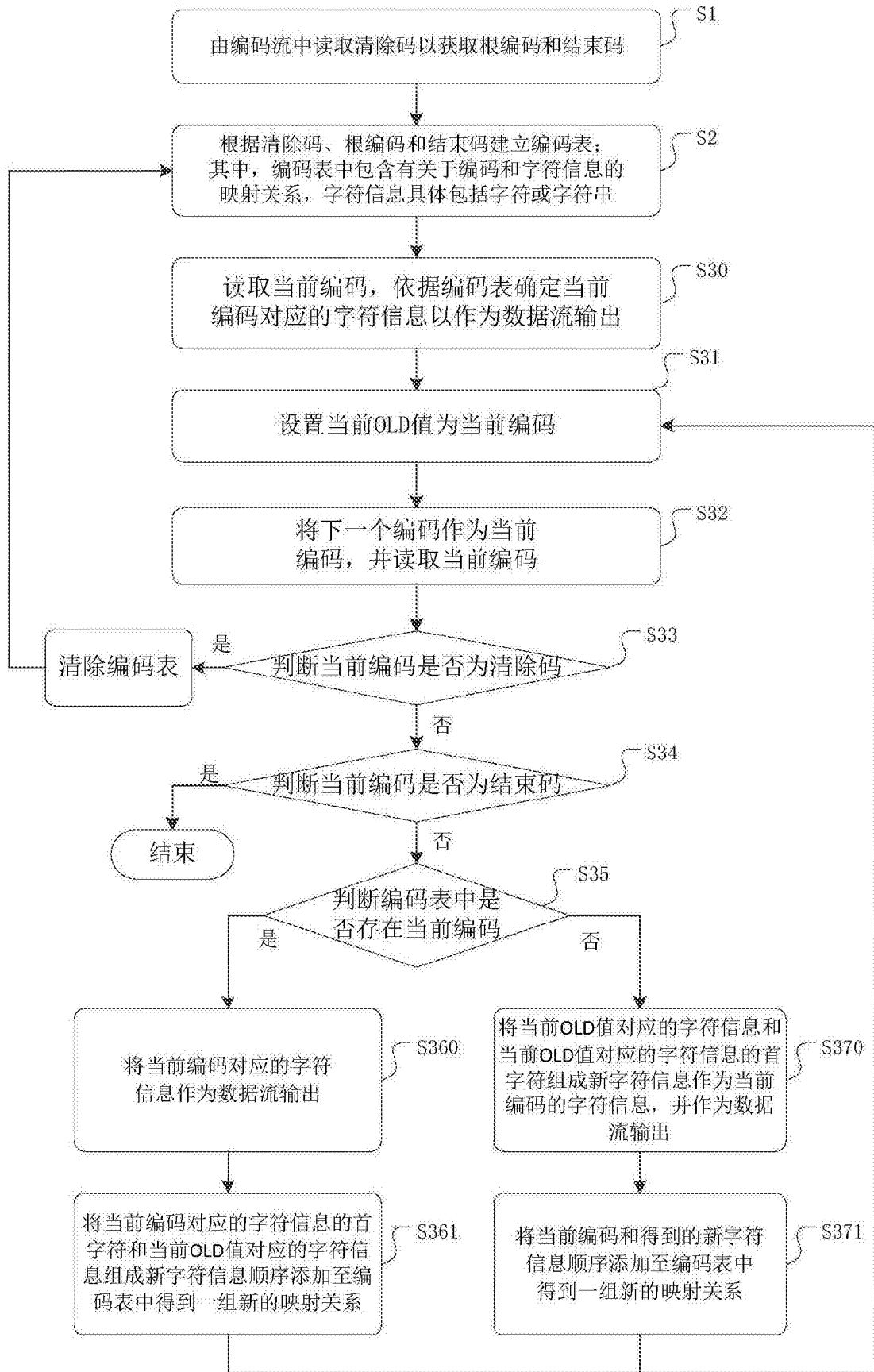


图3

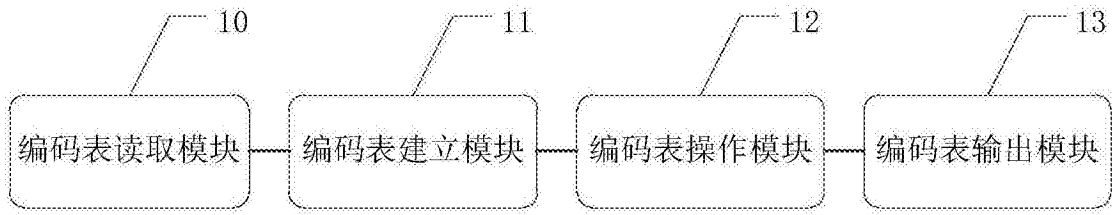


图4