统计检验方法在无损压缩中的应用

--基于《卫星海洋遥感数据的无损压缩及解压算法》

宋辰菲 邢晨 叶润欣 周晋妤 冯印国

一、 论文概述

1.1 背景

随着卫星遥感技术向高时间分辨率、高空间分辨率和高辐射分辨率方向的快速发展，空间遥感与信息技术已经发展成为满足人类对海洋资源和环境不同尺度和不同层次连续、动态的信息需求的必要手段。但是，在遥感数字化图像分发与处理中，普通城市的高分辨率遥感镶嵌图可达几十 GB 的数据量，超大的数据量给卫星链路的带宽、图像数据的传输、存储、共享、浏览和网络发布带来了时间和网络资源上的巨大压力。具体来说，首先，以文中的海洋遥感与信息技术实验室为例，该室每天可接收和处理 9 颗卫星资料，包括原始数据以及 1 级、2 级、3 级等各种产品的数据和图像，数据量巨大。其次，为了为保证数据资源的长期连续性和可靠性，这些图像及数据均需实施双备份，这样每天的数据量可超过 30 GB。因此，如何将海量的遥感数据进行有效压缩和储存成为人们日益关注的问题。

目前常用的有损压缩方法有赫华颖基于不同小波基的压缩算法，李强的自适应标量、矢量混合量化压缩方法以及周孝宽基于空间重采样的压缩算法。但对于遥感这类科学数据，应该尽可能地保证数据的真实性和可靠性，因而无损压缩具有重要的意义。目前无损压缩有张晓玲提出的遥感图像无损压缩编码以及耿则勋提出的水平与垂直预测相结合的无损压缩方法。但这类压缩算法多是针对陆地航空或卫星遥感领域的。因此，针对海洋遥感图像数据的压缩存储，本文比较研究了不同压缩算法对海表温度、叶绿素 a 浓度等不同遥感产品在压缩比方面的差异，提出了优化游程编码压缩算法。

1.2 游程编码

游程编码是压缩文件最简单的一种方法，它是把一系列的重复值用一个单独的值再加上一个计数值来取代。其中，重复值相当于游程中的 0/1，计数值相当于游程长度。这种算法对于具有长重复值的串的压缩很有效。比如，0001111 的七位数据只需转换为 0314 四位数据。就海洋遥感图像数据而言，因为数据具有非负特征，且陆地信息全为 0，当陆地面积较大时，获得的压缩比较高。游程编码的流程图如下图所示：

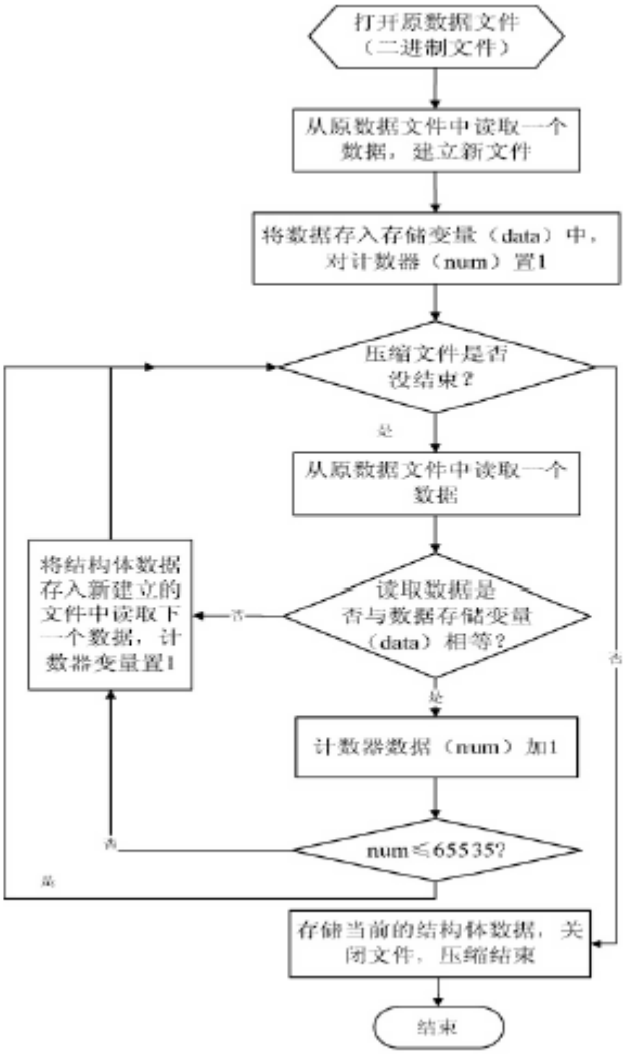

图 1 游程编码流程图

对于一个二进制文件，类似于游程中的 0/1 序列，首先，需要建立一个存储变量（data），一个计数器变量（num）且初值为 1。将第一个数据读入存储变量，再读取第二个数据，如果第二个数据与第一个数据相同，num 加 1，反之，将此时的 num 存入新文件中并重置 1，读取第三个数据与第二个数据进行比较。重复这个流程直到压缩文件结束。例如，对于原始数据 100111，得到存储变量为 101，计数文件为 123。

1.3 优化游程编码

游程编码虽然简单，但是，当数据的连续重复性较差时，该算法会出现“病态”的情况，如 XYZ 三位数据反而会扩充成 X1Y1Z1 六位数据。对于海洋数据而言，同一行的海洋水色水温数值往往不具有连续性，而连续几行具有相同数值的情况更少，因此，需要对游程编码模型进行优化。

考虑到海洋数据非负的特点，可取消游程编中区分压缩数据的最高标志位并将非连续重复的数据取反，即变为负数，这样就可以通过正负来区分哪些是压缩数据哪些是非压缩数据以此来避免非连续重复数据造成的“病态”情况。例如数据列 111010101，采用游程编码得到 13011101110111，位数增加了五位，采用优化游程编码变为 13（-1）（-0）（-1）（-0）（-1）（-0）（-1），仅有九位数。优化游程编码压缩算法如下图所示：

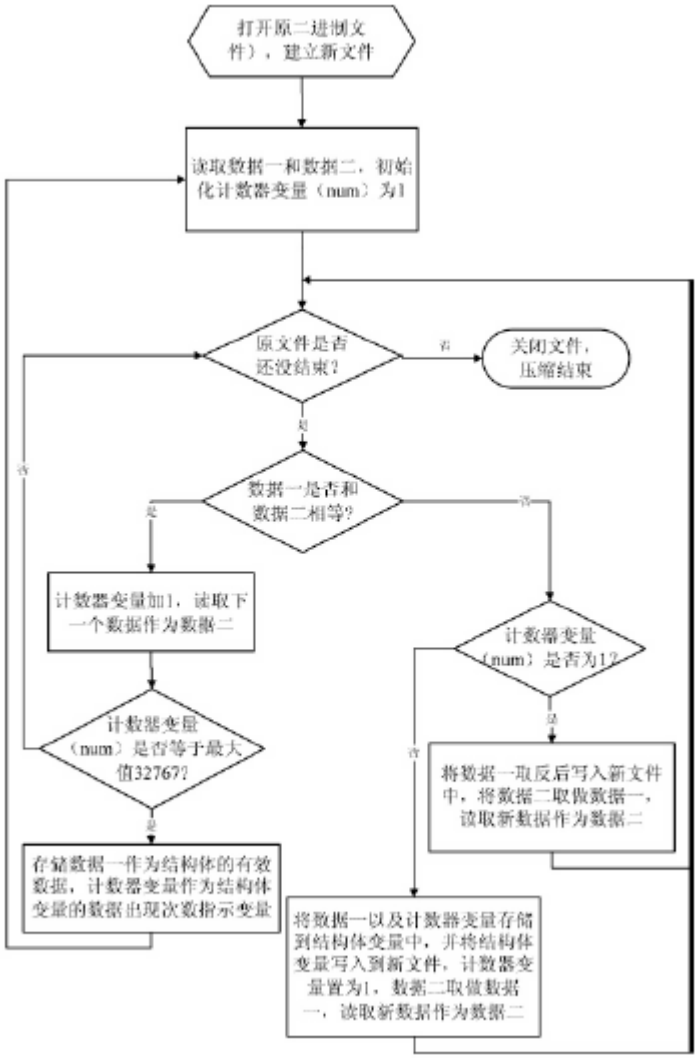

图 2 优化游程编码流程图

类似于游程编码压缩算法，首先，建立计数器变量（num）且初值为 1。读取数据一和数据二，如果第二个数据与第一个数据相同，num 加 1，读取第三个数据作为数据二与之前的数据再进行比较；反之，若 num 为 1，将数据取反写入文件，若 num 不为 1，将 num 和当前数据写入文件， num 重置 1。重复这个流程直到压缩文件结束。

1.4 实验结果

文章选取了广东海洋大学海洋遥感与信息技术实验室提供的 2011 年 5 月的 NOAA（National Oceanic and Atmospheric Administration）系列卫星海表温度（SST）3A 数据文件 92 个及 MODIS（The Moderate Resolution Imaging Spectroradiometer）卫星叶绿素 a 浓度（CHL）数据文件 70 个，对比了游程编码、优化游程编码等五种压缩算法的压缩率以及压缩时间。

压缩算法	压缩文件大小 70CHL/byte	压缩率 /%	压缩时间 /s	压缩文件大小 92 SST/byte	压缩率 /%	压缩时间 /s
三元组压缩	68 444 660	16.67	8.83	136 789 586	13.09	20.02
哈夫曼压缩	17 739 310	4.32	39.85	160 727 137	15.38	138.69
游程压缩	42 425 056	10.33	7.90	57 910 638	5.54	19.14
优化游程压缩	25 497 534	5.39	8.09	37 219 520	3.56	19.37
WinRAR	16 874 386	4.11	36.00	24 673 779	2.36	60.00

表 1 海洋遥感之海表温度及叶绿素 a 浓度 L3A 数据压缩统计表

由上表可见，对叶绿素浓度数据，压缩率最高的是 WinRAR，其压缩率达 4.11%，优化游程编码也接近 WinRAR 算法；但从时间复杂度来看，最佳的是游程压缩和优化游程压缩，用时不到三元组算法及 WinRAR 算法的 1/4。对海表温度数据，就空间复杂度，压缩率最高的是 WinRAR 工具，其压缩率达 2.36%，其次是优化游程编码，达 3.56%；但从时间复杂度来看，最佳的是游程编码，优化游程编码和三元组算法，它们均约 Win-RAR 算法的 1/3，是哈夫曼编码算法的 1/6。综合比较 5 种算法，不论在空间复杂度还是时间复杂度上，文章提出的针对海洋遥感数据文件的优化游程编码算法都有明显的比较优势。

1.5 非参数统计功能与应用

在游程编码压缩算法中，主要采用的是非参数统计中的游程相关的知识，读取连续重复的字段，用一个文件储存一个游程中相同的元素，用另一个文件储存游程数。这种压缩算法能够做到无损压缩与解压。对于算法中“病态”的情况，将非连续重复的字段取负，可以与其他连续重复的可压缩字段区分出来，有效解决病态的问题，同时满足了空间复杂度以及时间复杂度上的要求。这种压缩算法不需拟合参数和分布，理论上可以适用于大多数的数据，但是对具有大量连续重复字段的数据压缩性更好。

统计方法如非参数统计能够通过分析数据的特点，构建新的模型或对现有的模型进行优化，从而得到问题的解决方案。在本文中，因为海洋数据中的地面数据全为 0 以及存在大量连续可重复数据，可以考虑使用游程的思想存储游程中的重复数据以及重复次数，再因为数据非负性的特点，将游程数为 1 的数据取负，解决了压缩数据“病态”的问题。另外，针对某一特定问题构建的模型可以通过优化扩大其适用范围。虽然本文中的算法流程中要求原数

据文件为二进制文件，但实际上，只要数据非负且含有大量连续重复数据就可以采用这种方法进行压缩。比如 1223334444 可以压缩为 11223344。在存储分类型的问卷数据时，就可以考虑使用这种方法。由此可见，统计方法解决问题的优点在于能够根据数据的特点简化问题，基于统计思想构建基础模型，再逐渐放宽条件对模型进行优化，以扩大其适用范围。

二、 优化游程编码算法（基于趋势信息）

2.1 算法原理

基于趋势信息的优化游程编码算法是对优化游程算法的又一次改进，是对近似优化游程编码算法的扩充和拓展。基于趋势信息的优化编码算法是一种有损的算法，在压缩和解压的过程中不能 100%还原数据本来的模样，但却大大提高了压缩率。由于海洋数据在某些维度上具有一定的趋势性特征（如随着纬度的变化，海洋水温的变化存在一定的趋势性特征），利用这些数据的趋势特征，就可以达到用储存趋势的参数特征代替储存整段数据的效果。

2.2 算法设计

举一个简单的例子，数据中有一个数据片段为 1 1.31 1.62 1.91 2.20 2.50。如果通过优化游程算法来储存这段数据，那么储存的是（-1）（-1.31）(-1.62) (-1.91) (-2.2) (-2.5)，并没有达到压缩数据的效果。另外，这些数据之间差距都较大，并没有符合使用近似优化游程编码的算法的条件。我们观察到这些数据实际上具有一定的趋势性特征，前后两个数据之间的差值为 0.31 0.31 0.29 0.29 0.3,两两差值是相对比较稳定的。因此，可以通过储存这段数据的第一个数据 1，最后一个数据 2.5，以及两两差值的均值（0.31+0.31+0.29+0.29+0.3）/5=0.3 来反映这段数据的数值。在解压时，通过累和，可以得到与原数据相似的数据 1 1.3 1.6 1.9 2.2 2.5。

在完善该算法的操作时，由于直接对所有数据两两差值直接进行检验较为繁琐，可能会影响压缩的时间，因此考虑引入 Cox-Staut 趋势存在性检验对数据是否存在具有一定趋势的数据片段进行检验。Cox-Staut 趋势存在性检验过程中原假设和备择假设分别为H_0：该片数据序列无趋势H_1：该数据序列有增长或者下降趋势。对于拒绝了原假设的数据片段，即序列有增长或者下降趋势的数据片段再进行差值的检验。

2.3 算法流程

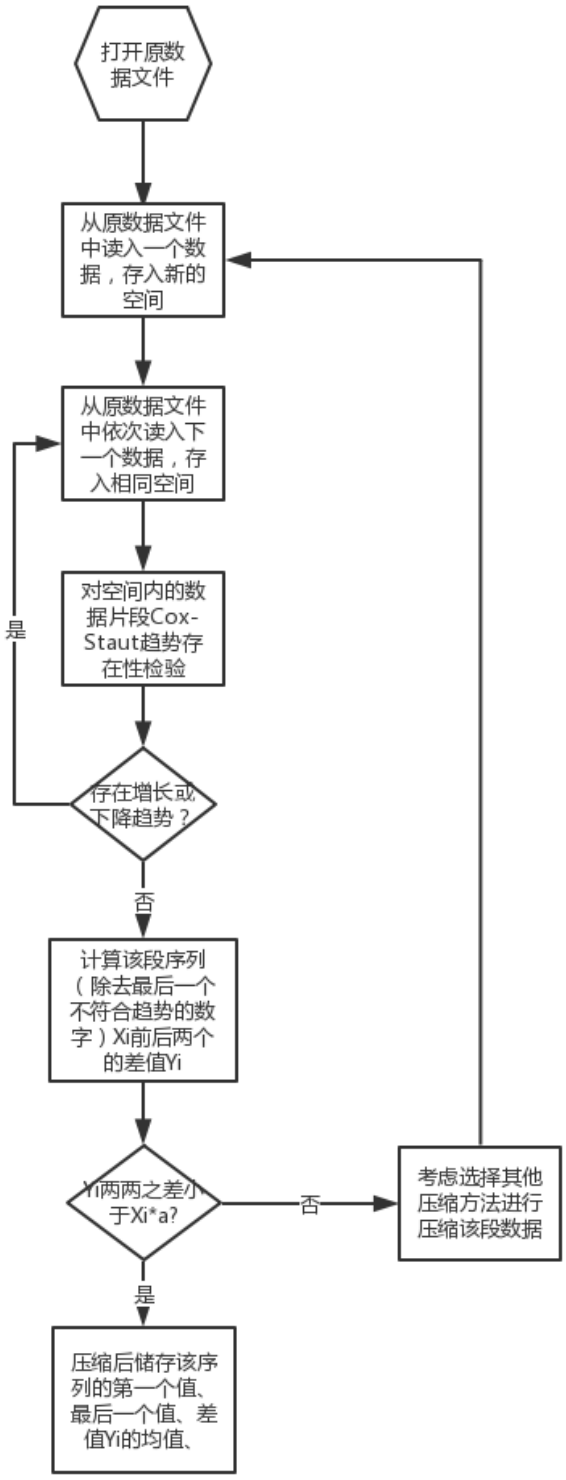

图 3 基于趋势信息的优化游程编码流程图

注：a 的取值决定该压缩过程的精度，可按照压缩精度、压缩后文件大小的要求进行选择。

2.4 算法优劣势分析

该算法通过储存数据的趋势信息代替整段数据达到对数据进行压缩到效果。它是对优化游程编码算法的又一次改进，也是对近似游程编码的扩充。近似游程编码算法考虑了两两差距近似为 0 的数据片段的情况，而本算法将其扩充为可处理两两差距近似为某一常数的数据片段。将整段数据通过趋势特征将其储存，对于具有一定趋势性数据，会大大提高压缩率。

该算法属于有损算法，在压缩和解压的过程中原数据的数值不能得到 100%的复原。另外，为了进行 Cox-Staut 检验和数据差值的检验，会消耗较多的时间，使得压缩时间较其他算法可能会有一定程度的上升。

2.5 统计方法的应用

2.5.1 Cox-Staut 趋势存在性检验

在本算法中，为了提高算法的效率，在进行数据差值的检验之前进行 Cox-Staut 趋势存在性检验。该检验为双边检验，原假设和备择假设分别为H_0：该片数据序列无趋势 H_1：该数据序列有增长或者下降趋势。该假设的原理为：假设数据$x_1, x_2, x_3 \dots x_n$独立，在原假设下，同分布为 F(x)，令

$$c = \begin{cases} \dfrac{n}{2} \text{ 如果 n 是偶数} \\ \dfrac{(n+1)}{2} \text{ 如果 n 是奇数} \end{cases}$$

取x_i和x_{i+c}组成数对（x_i, x_{i+c}）。当 n 为偶数时，共有 c 对。当 n 为奇数时，共有 c-1 对。计算每一数对前后两值之差：$D_i = x_i - x_{i+c}$。用D_i的符号度量增减。令S^+为正的D_i的数目，令S^-为负的D_i的数目。$K = \min(S^+, S^-)$,当正号太多或者负号太多都要拒绝原假设。在没有趋势的假设下，K 服从二项分布。

2.5.2 数据差值的检验

如果一个数据通过了 Cox-Staut 趋势存在性检验，就说明该数据具有增长或者下降的趋势。但是仅仅这个条件并不能使数据得到较为精确的压缩以及解压。在本算法中，我们还对数据的差值进行检验。检验的原理为：首先对通过趋势存在性检验的数据序列 Xi 两两计算差值得到序列 Yi。如果 Yi 取值较为稳定，说明该数据可以近似看成一个等差数列，就可以通过储存等差数列的首项、末项、差来达到储存整个序列的结果，在本算法中，“较为稳定”是通过计算 Yi 两两的差值与 Xi 首项的商，判断它是否小于 a 来度量的。这样的好处在于将数据的差值进行标准化，从而避免了由于原数据本身的数值大小对于稳定性判断的影响。

三、 矩阵优化游程编码算法

3.1 算法原理

原始数据通常在空间上相邻的点更易出现重复数据，并非以行或列的形式出现这一特征，而行程编码逐行读取数据的方式并未很好地利用这一特征。为了充分利用海洋遥感数据这一空间分布特征，可以改为以一个单位小矩阵依次读取，从而可以提高数据的连续重复性。

3.2 算法设计

设原始数据的矩阵大小为 m*n,单位阵的大小为 s*s，先将原始矩阵按照单位阵进行划分，根据 m 或 n 是否为 s 的倍数，将原始数据划分为三个子集（{[m/s]*s}*{[n/s]*s}、{m-[m/s]*s}*n、{[m/s]*s}*{n-[n/s]*s}）。在第一个子集中，每个单位阵内部按行读取，再依次读取下一个单位阵；在其它两个子集中按行读取数据，从而实现对原始数据的重构。

之后将得到的新数据文件进行优化游程编码，将重复值用一个单独的值加上一个计数值替代，如果出现病态数据则对数据进行取反处理。具体编码流程可以参考图 1、2。

3.3 算法流程

变量说明：

#data 原始数据
#Image 存储原始数据
#window_size 单位阵的维度
#Block 单位阵
#Main 原始数据的第一个子集（行列为单位阵维度的倍数）
#Right 原始数据的第二个子集（不能被整除的右半部分）
#Bottom 原始数据的第三个子集（不能被整除的下半部分）
#Result 压缩后的数据

伪代码如下：

```
Image <- data
Block <- window_size
Main <- image/block
Right <- right ( image – Main )
Bottom <- image – Main – Right
For block in Main:
    For line in block:
```

```
        If data_continue:
            Result.append( Record ( value, continue_number ))
        Else:
            Result.append(Record ( -value ))
For line in Right&Bottom:
    If data_continue:
        Result.append( Record ( value, continue_number ))
    Else:
        Result.append(Record ( -value ))
Print Result
```

3.4 算法优劣势分析

矩阵优化游程算法对于海洋遥感数据的压缩一方面保证了数据的完整性，使得数据可以完全恢复，另一方面结合了数据空间分布的信息，可以减少游程数从而高效压缩。同时由于单位阵大小的选择不同，必须根据原始矩阵的分割以及编码情况才能对数据的完整重构恢复，因此提高了对科学数据尤其是自主产品的保护，压缩文件的保密性好，安全性可靠度高。

然而由于对数据进行了先分割再优化游程两次处理，可能会增加压缩时间，且单位阵大小不同对于压缩效率有直接影响，还需要结合数据进行具体分析。对于这一问题，可能的优化方案是尝试动态改变单位阵的大小，将原始数据切割成大小不一、矩阵内部元素相同的若干方阵，记录各个矩阵的元素、维度以及位置信息，从而进一步降低数据存储所需的空间。

四、 矩阵优化游程编码算法（基于分布信息）

4.1 算法原理

优化游程算法考虑了分布的信息，但在经纬度的利用上显得不足；而矩阵优化游程编码算法考虑了经纬度的信息但在分布的利用上有所不足，因此综合前两种方法，提出了基于分布信息矩阵优化游程编码算法是一种新的有损算法，但算法的复杂和信息的有损性是其弊端。

4.2 算法设计

海洋数据存在一定的规律性和空间特点，比如靠近陆地的区域温度和叶绿素可能会更高，经纬度等地理信息的分布可以在一定程度上影响数据的分布，所以我们提出了新的改进。首先利用之前的矩阵划分方法，对所需要储存的数据进行重构以及横向和纵向的 Cox-Staut 检验，看是否存在一定的趋势性，并计算间隔。

举一个简单的例子，在二维矩阵中，分别为：

1.00 1.05 1.1

0.95 1 1.05

0.9 0.95 1.0

从横向上来看数据有增加的趋势，从纵向上来看数据有减少的趋势，利用第二部分的趋势存在性检验对其进行分析，看矩阵在一定程度上是否存在趋势。若存在趋势，我们就可对其间隔进行计算，记录初始点的位置，对横向纵向的间隔进行记录。

4.3 算法流程

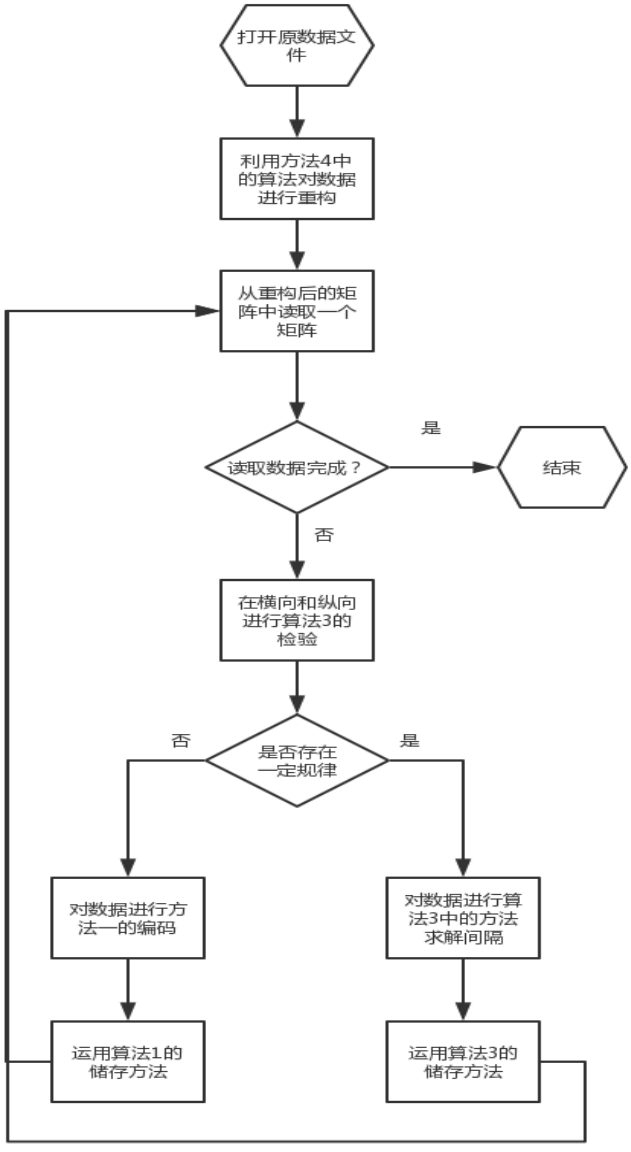

图 4 基于分布信息的矩阵优化游程编码流程图

4.4 算法优缺点分析

优点：该算法通过对横向和纵向的信息，以及分布信息的利用，相比以往的数据有更大的压缩性，同时综合了之前算法的优点。

缺点：在进行数据压缩的时候涉及到较多计算，压缩时间可能更长。同时由于是有损编码，信息的完整度也会下降。

五、 近似优化游程编码算法

5.1 算法原理

优化游程编码无损压缩算法的提出降低了数据的空间复杂度及时间复杂度，极大地节约了海洋遥感数据的存储和共享发布空间。但是由于海洋遥感数据中相邻数据（相邻像素）的数据值比较相近，比如相邻海域的海水温度比较相似，叶绿素浓度比较相近，因此为了达到更高的压缩率，我们基于优化游程编码算法提出了一种有损的近似优化游程编码算法。

5.2 算法设计

我们在数据中经常发现一系列数据也许不是相同的，但是极其相近，如果按照原来无损的优化游程编码算法，这一连串数据要花大于 2 个数据记录，但是我们将其记录成同一个数据，那么我们只需要 2 个数据，第一个是元素，第二个是重复个数。

具体做法是利用历史数据进行模拟，找到一个最优的数据临界差距 δ ，使得既能控制数据损失在一定范围内，又能保证压缩率得以有效提高。当数据一和数据二之差小于δ 时，我们在压缩时将其看成同一个数。

例如，假设我们规定 $\delta = 0.1$，有一串数据，1、1、1.02、1.08、0.97、0.8、0.87、0.76、1.2、1.2、1.9、2.1，按照优化游程编码算法，得到的压缩数据是 1、2、-1.02、-1.08、-0.97、-0.8、-0.87、–0.76、1.2、2、-1.9、-2.1，几乎没有压缩效果。但利用近似优化游程编码算法，结果是 1、5、0.8、3、1.2、2、-1.9、-2.1，极大地提高了数据压缩率。

5.3 算法流程

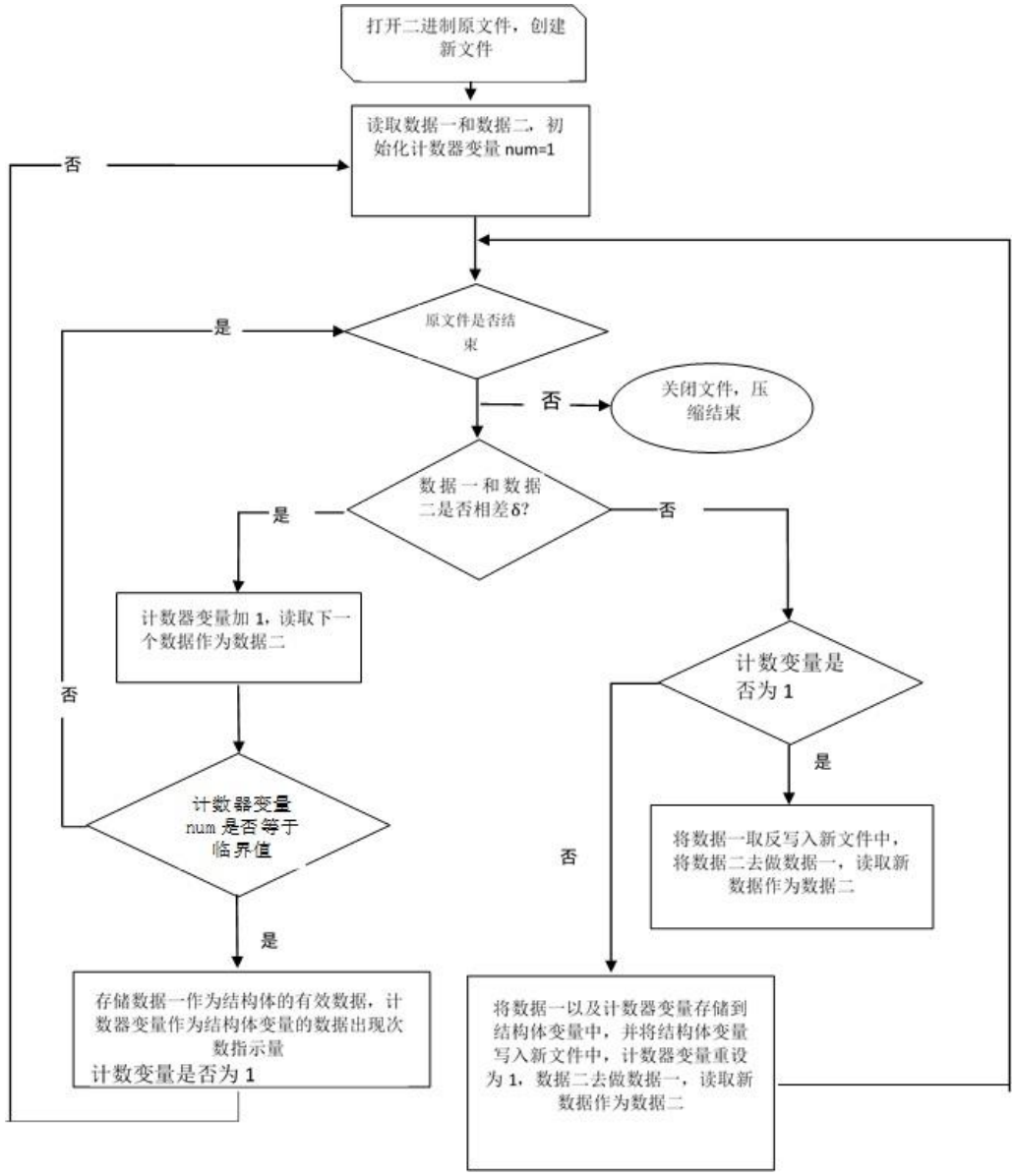

图 5 近似优化游程编码流程图

5.4 算法优劣势分析

近似优化游程编码算法的优点在于提高了压缩率，对于相近但不相等的数据序列具有明显的压缩效果。并且相对于优化游程编码算法，在压缩率上有了明显的提高，而卫星海洋遥感数据的数据特点也比较符合相邻数据取值相近的特点，适合近似优化游程编码算法。

但这个算法的缺点也很明显。首先，近似优化游程编码算法是牺牲了准确率的有损压缩，也即压缩数据无法百分百还原成原始数据。其次，近似优化游程编码算法要求通过对历史数据进行分析，得到合适的临界值 δ，且不同时间不同海域的合适的数据临界差距不同，并需要不断更新，这都增加了数据建模和编码的工作量。