星上高光谱图像无损压缩方法

舒　锐[1],武少林[2],龚　伟[3],陈　阳[2],张天序[2]

(1.上海卫星工程研究所,上海200240;2.华中科技大学 人工智能与自动化学院,湖北 武汉 430074;
3.武汉工程大学 电气信息学院,湖北 武汉 430062)

摘　要: 针对星上高光谱图像压缩问题,本文从消除高光谱图像的谱间与谱内空间冗余信息入手,提出了基于KLT与自由无损图像格式(FLIF)算法的高光谱图像无损压缩算法。KLT用于去除图像的谱间冗余信息,FLIF算法的MANIAC决策树能够在压缩过程中动态更新上下文环境内容,实时调整上下文环境的数量,多上下文环境建模的设计大幅提升了压缩性能。仿真实验表明:在高分五号卫星上的高光谱数据集上,本方案取得了不错的压缩比特率结果。此外,针对KLT变换后图像的特点,本文提出了多压缩器灵活切换的加速方法,在可见光波段让压缩耗时缩短为未优化前的1/5。

关键词: 高光谱图像; 无损压缩; 可逆整数KLT; 自由无损图像格式(FLIF)

中图分类号: TN 911.73; TP 391.9　　**文献标志码:** A　　**DOI:** 10.19328/j.cnki.2096-8655.2021.04.020

Lossless Compression Method for On-board Hyperspectral Image

SHU Rui[1], WU Shaolin[2], GONG Wei[3], CHEN Yang[2], ZHANG Tianxu[2]

(1.Shanghai Institute of Satellite Engineering, Shanghai 200240, China; 2.School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, Hubei, China; 3.School of Electrical and Information Engineering, Wuhan Institute of Technology, Wuhan 430062, Hubei, China)

Abstract: A lossless compression algorithm based on the Karhunen-Loève transform (KLT) and the free lossless image format (FLIF) algorithm is proposed to eliminate the redundant information between and within the spectra of hyperspectral images, so that the problem of on-board hyperspectral image compression can be solved. The KLT is used to remove the redundant information between the spectra of the images. The MANIAC decision tree of the FLIF algorithm can dynamically update the content of the context environments and adjust the number of the context environments in real time during the compression process. The design of multi-context environment modeling greatly improves the compression performance. Simulation experiments show that the proposed scheme achieves good bit rate compression results on the hyperspectral data set of Gaofen-5 satellite. In addition, in view of the characteristics of images after KLT, an acceleration method for flexible switching of multiple compressors is put forward, which can shorten the time consumption of compression to one-fifth of that before optimization in the visible light band.

Key words: hyperspectral image; lossless compression; reversible integer Karhunen-Loève transform (KLT); free lossless image format (FLIF)

0　引言

JPEG-LS (Joint Photographic Experts Group-Lossless)是图像无损压缩领域中最经典的、基于预测的算法,于2000年由联合图像专家小组JPEG所提出,其核心是LOCO-I(Low Complexity Lossless Compression, LOCO-I)算法。JPEG-LS算法先利用周围左边、左上边及上边的3个像素值做一个简单的中值边界检测预测,然后通过上下文模型修正

收稿日期: 2020-11-11; **修回日期:** 2021-01-05

基金项目: 上海航天科技创新基金(SAST2018-030)

作者简介: 舒　锐(1977—),男,博士,高级工程师,主要研究方向卫星有效载荷与处理技术。

通信作者: 武少林(1995—),男,硕士,主要研究方向为自动目标识别、异源图像红外仿真、高光谱图像压缩。

预测值,最后转化为Golomb-rice编码输出。JPEG-LS算法以较低的复杂度达到了当年(2000年)的最高压缩比。LOCO-I算法仅为单通道的灰度图像设计,SHAWN等[1]于2004年将LOCO-I推广到了高光谱图像压缩,提出了LOCO-3D算法,压缩比提高了约0.2。2015年,在C-DPCM算法的基础上,WU等[2]发现剔除原始高光谱图像数据中的异常像素值后,预测的准确率提升了,此附带局部光谱异常数据剔除的算法被称为C-DPCM-RLSO(C-DPCM-Removal of Local Spectral Outliers)。人工智能的研究热潮将神经网络技术带到了各行各业中,LUO等[3]于2019年提出了C-DPCM-RNN(C-DPCM-Recurrent Neural Networks)方法,将C-DPCM中的线性预测器替换为循环神经网络(Recurrent Neural Network, RNN),该方法在矫正的机载可见光/红外成像光谱仪(Airborne Visible Infrared Imaging Spectrometer,AVIRIS)公开高光谱测试集上取得目前本领域内最高的压缩比。目前,现存的很多高光谱图像压缩算法的通用性不强,很多算法采取单一的上下文模型,这类固定的上下文建模算法存在明显的弊端,容易产生偏好现象。例如,JPEG算法在医学图像上的压缩性能表现明显优于其他类型图像,JPEG 2000算法偏向于场景过度平滑的自然照片[4]。此外,有些算法需要调优参数才能达到较好的压缩性能。例如,C-DPCM类算法虽然取得目前本领域内最好的压缩效果,但是算法需要提前给定图像聚类数目[5],显然,不同类型的图像的最优聚类数目是不同的,不恰当的图像聚类数目会降低算法的压缩性能。

针对上述目前很多算法存在的通用性与实用性方面不足的问题,本文从消除谱间信息冗余、谱内空间信息冗余这两个角度展开研究,综合了基于变换和基于预测的压缩技术,提出了一种基于KLT(Karhunen-Loève Transform)和自由无损图像格式(Free Lossless Image Format,FLIF)算法的通用型高光谱图像无损压缩算法。

1　设计思路

KLT,也称为主成分分析(Principal Components Analysis ,PCA)变换,是一种能有效提取关键图像信息以及消除冗余的工具[6]。KLT是一种线性变换,通过旋转坐标系让原始图像数据尽可能地落在新的坐标轴上,算法原理是通过旋转坐标系使得原始数据的协方差矩阵对角化。KLT后的图像数据谱间将会不具备相关性。

本方案首先对高光谱图像进行可逆整数KLT,去除谱间相关性;再对可逆整数KLT后的高光谱图像逐通道计算图像均匀性指标SR;对较均匀的图像通道采用FLIF压缩器编码,对均匀性差的图像通道切换到哈夫曼编码器。通过对高光谱图像做一个可逆的整数KLT变换来去除其谱间相关性,降低了原始数据的信息熵,去除了谱间的冗余信息,然后采取FLIF压缩编码来消除谱内空间信息冗余,从而取得高压缩比。为了优化方案算法的压缩速度,本方案还提出了根据图像特点灵活切换压缩系统的策略,针对均匀图像(高SR值)使用FLIF算法压缩编码,针对非均匀图像(低SR值)切换到简单的哈夫曼编码以加快压缩速度。本方法的算法流程如图1所示。

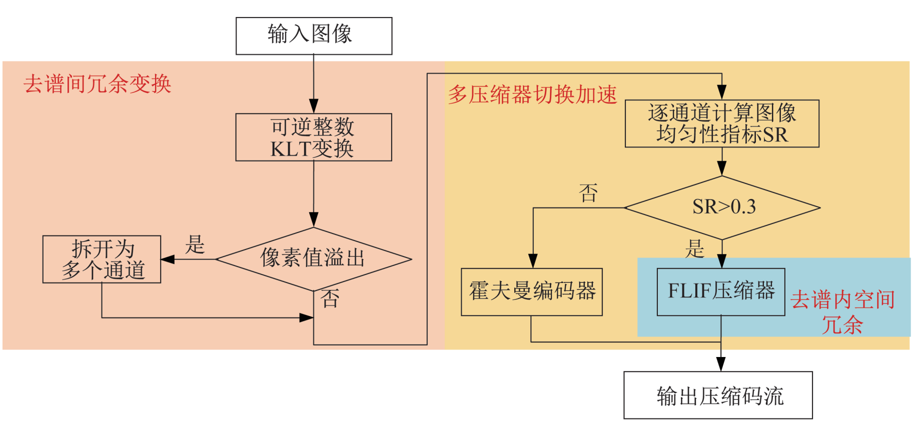

图1　高光谱图像无损压缩总体方案流程图

Fig.1　Flow chart of the overall scheme for lossless compression of hyperspectral images

FLIF 压缩算法的工作流程图如图 2 所示。FLIF 算法采取逐行逐个像素扫描的工作方式，像素首先经过 1 个固定的预测模板得到残差值，残差进一步分解为近零符号编码表示。后续 MANIAC 决策树将计算出上下文环境属性值，决策树根据属性标签值自顶向下遍历为该像素值选出最优的上下文环境，随后更新所有相关上下文环境，并对决策节点进行调整，调整决策树有助于提高下个输入像素的预测准确度。最后，算术编码器根据上下文环境提供的概率分布信息完成编码，并输出压缩码流。

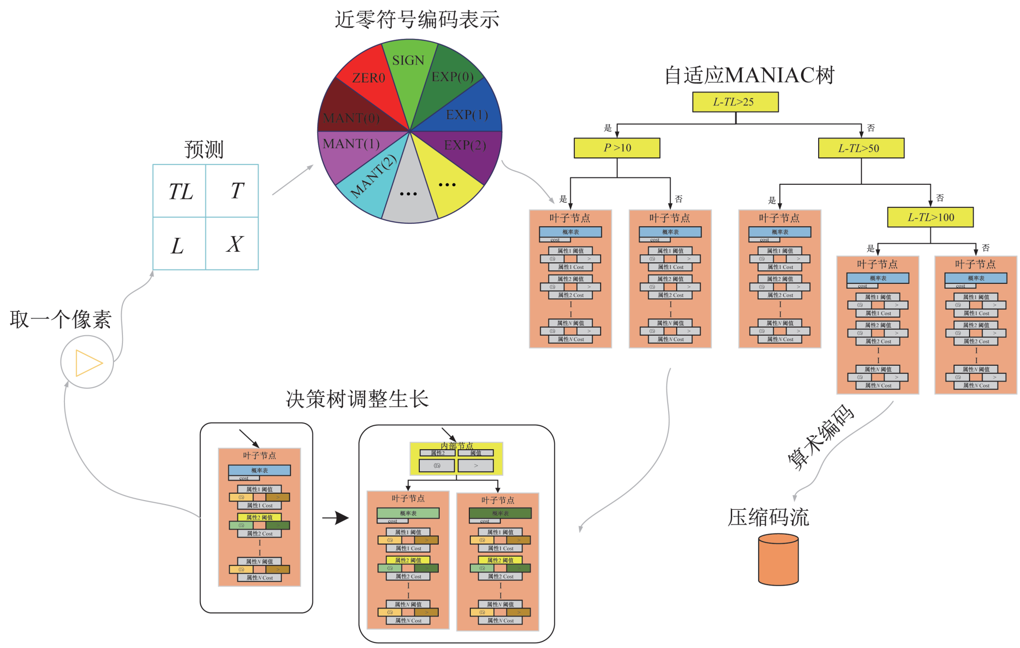

图 2 FLIF 算法的工作流程

Fig.2 Flow chart of the FLIF algorithm

图中，X 表示待预测位置的像素，T 表示 X 的上方像素，L 表示 X 左边的像素，TL 表示左上角的像素，可以根据 T、L、TL 像素值来预测 X 的值。

近零符号编码由零位(ZERO)、符号位(SIGN)、指数位(EXP(k, sign))和尾数位 MANT(k)这几个部分组成。

MANIAC (Meta-Adaptive Near-zero Integer Arithmetic Coding)即元自适应近零整数算术编码。

2 数学模型的建立

2.1 KLT

KLT 的第 1 步是计算原始高光谱图像数据的协方差矩阵 $\boldsymbol{C}_{\mathrm{OV}}$：

$$\boldsymbol{C}_{\mathrm{OV}} = \begin{bmatrix} c_{\mathrm{ov}}(p_1, p_1) & \cdots & c_{\mathrm{ov}}(p_1, p_m) \\ \vdots & \ddots & \vdots \\ c_{\mathrm{ov}}(p_n, p_1) & \cdots & c_{\mathrm{ov}}(p_n, p_m) \end{bmatrix} \tag{1}$$

式中：$c_{\mathrm{ov}}(p_i, p_j)$ 为第 i 个谱段和第 j 个谱段的协方差。

KLT 的目的是去除谱间图像的相关性，可以将去除谱间数据相关性的问题转化为将协方差矩阵 $\boldsymbol{C}_{\mathrm{OV}}$ 中的元素 $c_{\mathrm{ov}}(p_i, p_j)$ 置为 0($i \neq j$)，即为任意两个谱段数据的协方差为 0。KLT 变换的原理是把协方差矩阵 $\boldsymbol{C}_{\mathrm{OV}}$ 对角化。假设存在矩阵 $\boldsymbol{P}$，使得满足

$$c_{\mathrm{ov}} \cdot \boldsymbol{P} = \boldsymbol{P\Lambda} \tag{2}$$

式中：$\boldsymbol{\Lambda}$ 为对角矩阵。

KLT 是一种线性变换，为了使变换后的新坐标系相互垂直，必须对上面的变换矩阵 $\boldsymbol{P}$ 进行正交化，正交化后的矩阵 $\boldsymbol{P}$ 即为 KLT 矩阵。

求解正交变换矩阵 $\boldsymbol{P}$ 的方法分为两步：首先将协方差矩阵 $\boldsymbol{C}_{\mathrm{OV}}$ 对角化得到变换矩阵 $\boldsymbol{P}$；然后再对矩阵 $\boldsymbol{P}$ 正交化。矩阵对角化的详细步骤如下所述。

通过求解下面的线性方程组

$$(\lambda \boldsymbol{E} - c_{\mathrm{ov}})x = 0 \tag{3}$$

式中：λ 为特征值；$\boldsymbol{E}$ 为单位向量

对式(1)运用线性代数的方法，能解出 p 个特征值与 p 个相应的特征向量。再把特征向量按照特征值从大到小的顺序重新排列，得到变换矩阵为

$$P=\left[p_1\ p_2\ \cdots p_n\right],\ (n=1,2,\cdots,P) \tag{4}$$

式中：p_1为最大的特征值λ_1所对应的特征向量；以此类推，p_n为最大的特征值λ_n所对应的特征向量。

对式(4)得到矩阵P依次进行正交化与单位化，使得矩阵P满足

$$P^{\mathrm{T}}P=E \tag{5}$$

采用施密特正交化方法规范化向量$p_1,p_2,\cdots,p_r$。正交化的过程如下：

$$\beta_1=p_1 \tag{6}$$

$$\beta_2=p_2-\frac{(p_2,\beta_1)}{(\beta_1,\beta_1)}\beta_1 \tag{7}$$

$$\beta_3=p_3-\frac{(p_3,\beta_1)}{(\beta_1,\beta_1)}\beta_1-\frac{(p_3,\beta_2)}{(\beta_2,\beta_2)}\beta_2 \tag{8}$$

$$\beta_r=p_r-\frac{(p_r,\beta_1)}{(\beta_1,\beta_1)}\beta_1-\frac{(p_r,\beta_2)}{(\beta_2,\beta_2)}\beta_2-\cdots-\frac{(p_r,\beta_{r-1})}{(\beta_{r-1},\beta_{r-1})}\beta_{r-1} \tag{9}$$

最后，将向量组$\beta_1\ \beta_2\cdots\beta_r$单位化，具体方法如下：

$$\varepsilon_1=\frac{\beta_1}{\left|\beta_1\right|},\varepsilon_2=\frac{\beta_2}{\left|\beta_2\right|},\cdots,\varepsilon_r=\frac{\beta_r}{\left|\beta_r\right|} \tag{10}$$

式中：$\left|\beta_1\right|$为向量β_1的模。

联立式(10)求得单位化后的向量，KLT 矩阵即为所求，记做$A^{\mathrm{T}}=\left[\varepsilon_1\ \ \varepsilon_2\ \ \cdots\ \ \varepsilon_n\right]$，$(n=1,2,\cdots,P)$，那么$A$为

$$A=\left[\varepsilon_1\ \ \varepsilon_2\ \ \cdots\ \ \varepsilon_n\right]^{\mathrm{T}},\ \ (n=1,2,\cdots,P) \tag{11}$$

上面就是 KLT 的全部步骤，使用变换矩阵A去谱间相关性的公式如下，假设输入的高光谱图像数据如下公式的形式，依旧记为X，记去谱间相关性后的数据记为Y，变换公式为

$$Y=(AX^{\mathrm{T}})^{\mathrm{T}} \tag{12}$$

2.2 近零符号编码

近零符号编码是 FLIF 算法数据存储块（包含像素值数据）的一种数据表示形式[7]。任意整型数据都能用近零符号编码表示，近零符号编码由零位、符号位、指数位和尾数位这几个部分组成，其中，指数位部分采取简单的一元编码（Unary Code）。所谓一元编码，即使用$n-1$个二进制数据1与末位尾数0（或者使用$n-1$个二进制数据0与末位尾数1）来表示存储正整数n。在 FLIF 里的近零符号编码中，一元编码只用于表示指数位数据部分[8]。对于任意整数数据x，所需的指数位二进制比特数目为：$e=\lceil \log_2|X|\rceil e=\left[\log_2|x|\right]$（结果如果有小数，往上取整）。因此，假设采取一元编码表示16位的图像数据，近零符号编码的指数位部分二进制长度有上限，即最多需要16个 bpp（bit per pixel）。

近零符号编码表示的每个 bpp 都有它自己的上下文环境，所以每个 bpp 的概率相互隔离，能够自动适应数据。在 FLIF 里的近零符号编码中，指数位数据部分 bpp 的概率不仅取决于它们所在的位置，还取决于它们的符号。

近零符号编码针对每个 bpp 都建立独立的上下文环境，每个 bpp 的可能值只能是要么为0，要么为1。由于 FLIF 的核心是算术编码，近零符号编码使用一个12位的整型数据存储每个 bpp 的概率。12位的无符号（概率不可能为负数）整型数据能表示的整数范围是0～4 095，所以概率值为1时表示该 bpp 为1的概率接近于零（该 bpp 为1几乎是不可能事件），当概率值为4 095时表示该 bpp 为1的概率接近于1（该 bpp 为1是必然事件）。

FLIF 首先对该通道的图像数据逐行从左向右扫描得到预测误差值，扫描时比较存储当前通道图像预测误差值的最大值与最小值。在获得该通道图像预测误差值的最大值与最小值之后，这时我们就能确定k的上限值为

$$k_{\max}=\left[\log_2|x|_{\max}\right] \tag{13}$$

式中：$[x]$为向下取整，注意这里x一定取的是绝对值最大的值。

为了说明近零符号编码表示指数位的方法，下面以正整数1 000为例说明，假设$k_{\max}=12$，先计算$k=\left[\log_2 1\,000\right]=10$，因此 $\exp(k,\mathrm{sign})(k=0,1,\cdots,9)$ 这 10 个 bbp 都置为 1，同时 $\exp(k,\mathrm{sign})(k>9)$也就是剩下的 bpp 都置为0。其实近零符号编码表示指数位$\exp(k,\mathrm{sign})$的本质是记录数据有二进制位的长度，假如$k_{\max}=12$，说明该通道图像的预测误差值最多有12位。根据实际的像素预测经验情况，预测误差值范围通常为正太分布，预测误差值的典型值一般为几百左右，到$\exp(6,\mathrm{sign})$即误差值为128，此时概率值达到最大

值 2 800。相反 exp(k, sign)高位的概率值一般给较小的值(认为预测误差特别小的发生概率小)，到从表 3 的初始概率值可以看出，exp(k, sign)从高位($k=0$)到低位($k=k_{\max}$)的初始概率值基本上是先逐渐增加然后减小的趋势。

近零符号编码的尾数部分用 MANT(k)，$k=1,2,3,\cdots$ 表示，正如上所述近零符号编码表示的指数部分记录数据的二进制长度，所以数据的最高位必为 1，数据尾数用 MANT(k)表示，尾数 MANT(k)中的 k 值一定比数据的二进制长度少一位，故近零符号编码的尾数部分 MANT(k)中的 k 值等于指数部分 exp(k, sign)中的 k 减 1 位。即：$k_{\mathrm{mant}}=k_{\exp}-1$。

2.3　自适应 MANIAC 树

MANIAC (Meta-Adaptive Near-zero Integer Arithmetic Coding)即元自适应近零整数算术编码[9]。所谓元自适应，是指 FLIF 编码器的上下文模型能够自动适应数据的特点。

FLIF 算法使用 CABAC(Context-Adaptive Binary Arithmetic Coding)的一种变种方法作为其核心编码器[10]。因此，本文先简要地介绍 CABAC 方法的核心原理。CABAC 算法的基础也是算术编码器，其中的概率模型是自适应的。CABAC 算法逐 bpp 压缩图像数据，所以其中的算术编码器针对 bbp 进行建立概率模型，bpp 只有两种可能值 1 或 0，CABAC 算法对 bpp 的初始概率各赋值为 50%，即认为 bpp1 或 0 位出现的概率相等。CABAC 每压缩一个 bpp，其概率模型就会相应地更新。这种概率随着压缩进程动态调整的策略有利于模型学习到数据的真实概率分布情况，从而提升算法的压缩性能。

CABAC 算法本质上还是只有一个上下文环境模型，如果有多个上下文环境模型信息，那么就能给每个上下文环境分配一个独立的概率模型[11]。参考 FFV1 中的静态上下文环境设计，其上下文环境的数量为属性域的乘积，所以很难得到一个合适的上下文环境数量值。过多的上下文环境不利于压缩，因为模型的自适应能力是有限的(假设极限情况，一个或几个像素共用一个上下文环境)。同理，过少的上下文也不利于压缩，因为处于不同环境条件下的像素，其像素值的分布规律不同却共用同一个上下文环境也是不科学的。此外，固定数量的上下文环境数量会导致很多上下文环境被闲置，完全没被利用上。

与 CABAC 算法相比，FLIF 使用一种动态数据结构作为其核心上下文环境模型。这种动态数据结构的本质是一颗决策树，决策树会随着压缩过程不断地生长。典型的 MANIAC 树示意图如图 3 所示，MANIAC 树内部节点与叶子节点的放大图如图 4 所示。MANIAC 树的每个内部节点(非叶子节点)都有一个测试条件：判断一个上下文属性值(FLIF 含有多个)是否大于(或小于等于)阈值。类似二叉树的数据结构，MANIAC 树的内部节点也有左右两个分支。在压缩图像数据过程中，每个叶子节点包含一个实际的上下文环境，同时叶子节点还包括很多属性条件，每条属性拥有两个虚拟的上下文环境。而在解压图像数据的过程中，FLIF 只会用到实际的上下文环境。

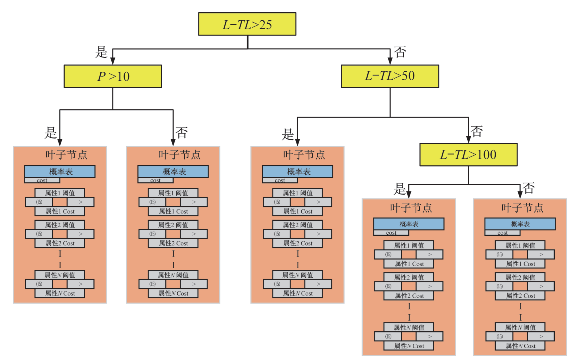

图 3　典型的 MANIAC 树示意图

Fig.3　Schematic diagram of a typical MANIAC tree

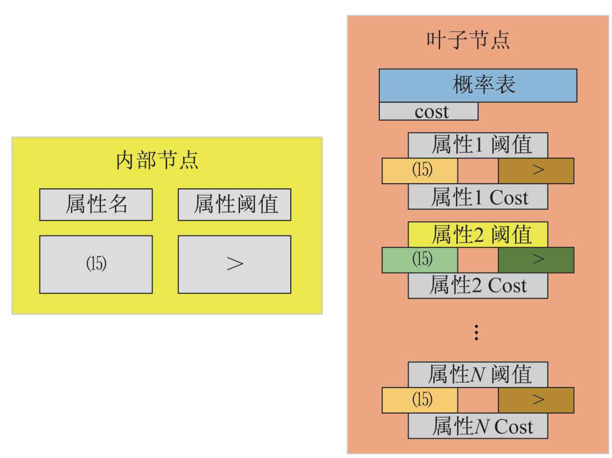

图 4　MANIAC 树内部节点与叶子节点的放大图

Fig.4　Enlarged view of the internal nodes and leaf nodes of the MANIAC tree

所谓自适应 MANIAC 树，下面详细说明 MANIAC 树的动态调整生长过程，也即自适应过程[12]。对于每个输入的待压缩像素数据，FLIF 首先计算每条属性条件值，结合计算好的每条属性值开始遍历 MANIAC 树直到抵达叶子节点。叶子节点包含一个实际的上下文环境，利用其输出压缩比特流，以及一个损失值（输出 bbp 的个数）。FLIF 取每条属性值的平均值作为其比较阈值，该条属性下小于阈值的有一个虚拟的上下文环境，而大于等于阈值的也有一个虚拟的上下文环境。压缩完一个像素值后，每条属性下的两个虚拟的上下文环境会分别更新各自的概率表与损失值。损失值也含义是该上下文环境压缩了多少个像素值。

MANIAC 属性向量组成了 FLIF 算法的上下文环境，MANIAC 树里叶子节点所包含的上下文环境的主要成分也是这些 MANIAC 属性组。假设当前待压缩的像素为 X，X 周围像素值的位置关系如图 5 所示。

		TT	
	TL	T	TR
LL	L	X	R

图 5　像素 X 周围像素的位置关系

Fig.5　Positional relationship of pixels around pixel X

FLIF 定义了一些属性条件，见表 1，这些属性条件构成了 FLIF 算法至关重要的上下文环境。

表 1　MANIAC 属性表

Tab.1　Attribute table of MANIAC

属性	描述
P	像素预测值，即$(L+T-TL, L, T)$3 个值的中值
M_{ni}	中值的索引号
$L-TL$	左边像素值－左上像素值
$TL-T$	左上像素值－上边像素值
$T-TR$	上边像素值－右上像素值
$TT-T$	上上像素值－上边像素值
$LL-L$	左左像素值－左边像素值

上表中 M_{ni} 为像素预测使用哪个预测器，因为像素预测最终取的是$(L+T-T, L, T)$3 个值的中值，故 M_{ni} 定义为

$$M_{ni}=\begin{cases}0, & P=L+T-TL\\ 1, & P=L\\ 2, & P=T\end{cases}\tag{14}$$

计算上下文环境里的损失值其实是在评估哪条属性对压缩任务帮助最大。如果该条属性是无关紧要的，那么该属性下的两个虚拟上下文环境的损失值之和就会大于等于实际上下文环境的损失值。相反，如果该条属性十分关键，那么使用两个不同上下文环境来编码数据将会更有利于压缩。通过比较叶子节点中实际上下文环境的损失值与属性下两个虚拟上下文环境的损失值，当某个属性下两个损失值的差值大于某个阈值的时候，此叶子节点就变成为一个决策节点，此决策节点的测试条件就是本属性。上述的叶子节点转变决策节点的过程，其实也是 MANIAC 树生长的过程，如图 6 所示，叶子节点一分为二，裂变为两个新的叶子节点。

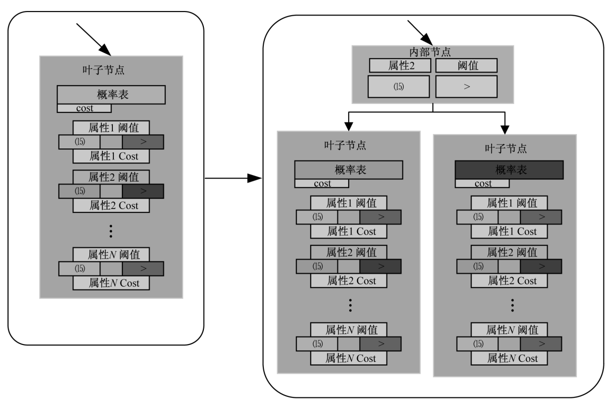

图 6　MANIAC 树生长过程示意图

Fig.6　Schematic diagram of the growth process of the MANIAC tree

2.4　提高压缩速度的方法

考虑到 KLT 能分离出大部分噪声数据的特点，这里提出了针对 FLIF 压缩器的优化加速方法。FILF 压缩器属于复杂压缩系统，压缩性能强，但是内部算法复杂度高，耗时长；而哈夫曼编码器属于简单压缩系统，速度快，但是压缩能力弱。本文提出了根据图像均匀程度灵活地在 FILF 压缩器与哈夫曼编码器之间切换的策略，对

均匀图像应该选择FILF压缩器以提升压缩能力;但是对于噪声大的非均匀图像FILF压缩器与哈夫曼编码器性能相当,此时选择更快的哈夫曼编码器以缩短压缩时间。本文综合图像行间相关性与列间相关性作为图像均匀性指标,图像行间相关性与列间相关性的权重各取一半(即50%)。计算出图像所有相邻行之间的相关系数并取均值$\bar{\rho}_x$作为全图的行间相关性,运用相同的方法计算出全图的列相关系数$\bar{\rho}_y$。定义图像均匀度指标SR为$\left(\bar{\rho}_x+\bar{\rho}_y\right)/2$,其最小值为0,最大值为1,根据统计学知识,一般认为相关系数0.8以上表明具有强的相关性;相关系数0.3~0.8之间认为数据间具有一定的相关性;而相关系数0.3以下则认为数据间没有相关性。相关系数区间含义如下:

$$\begin{cases} SR \geqslant 0.8, & \text{强相关性} \\ 0.3 \leqslant SR < 0.8, & \text{弱相关性} \\ SR < 0.3, & \text{无相关性} \end{cases} \tag{15}$$

本文通过灵活地切换复杂压缩器(FLIF)与简单压缩器(哈夫曼编码器)来加速压缩过程,关于切换压缩器的时机如何选择的问题,结合上述的相关系数区间含义,本文设置图像均匀度指标SR的阈值为0.3。当图像均匀度指标SR大于0.3时,说明图像具备较强的空间相关性的特点,故采取FLIF压缩这类图像;当图像均匀度指标SR小于0.3时,说明图像受噪声污染严重,故切换到哈夫曼编码器来压缩这类图像以缩短压缩时间。

哈夫曼编码属于变长编码技术,能够有效地消除数据的编码冗余,是一种基于二叉树的实现输出最短二进制码长度的方法。哈夫曼编码需要数据分概率的先验信息,故要先统计图像所有像素值的频率,为了节省存储空间,频率信息使用一个无符号的16位整型数据存储,16位无符号整型数据所能表示的范围仅为0~65 535,因为可能存在频率超过65 535的情况,频率表最后会统一归一化到0~65 535之间。为了减少先验频率分布信息的体积,本文仅对该图像存在的值存储统计的频率分布信息,即仅存储像素值最小值到最大值之间的频率分布信息。有了数据频率分布之后,先建立哈夫曼编码树(数据结构是一颗二叉树),再遍历哈夫曼树得出每个数据的二进制编码,哈夫曼树会为高频率的数据分配较短的编码,进而实现最优编码过程,接下来根据编码表对待压缩像素重新生成二进制码,消除编码冗余。解码方法跟上述的压缩过程基本相同,也是先建立哈夫曼树,然后根据压缩码流逐位读取遍历哈夫曼树直到遍历到叶子节点便恢复出一个像素数据,接着继续取压缩码流重复遍历过程直到读尽压缩码流(或者恢复出最后一个像素)便停止。

3 仿真试验及结果分析

3.1 仿真结果

本文的实验数据来自高分五号卫星高光谱数据,数据分两部分:可见光近红外(VN)和短波红外(SW)。VN有150个波段,SW有180个波段,总共330个波段。

VN波段范围0.39~1.03 μm左右,5 nm一个间隔。SW波段范围1.0~2.5 μm左右,10 nm一个间隔。空间分辨率30 m。试验数据选取北京某机场,选取6组数据,分别为VN①~VN③,SW①~SW③,如图7所示。

图7 VN①(第50、60、70波段合成)高光谱实验测试图像

Fig.7 Hyperspectral experimental test image for VN ① (integrated 50th, 60th, and 70th waveband)

压缩结果指标统一采取每个像素所占的bpp指标,本文对多组高光谱图像测试用例进行压缩实验,见表2。

表 2　高光谱测试图像的无损压缩实验的 bpp 结果

Tab.2　The bpp results of the lossless compression experiment of hyperspectral test images

算法名	VN①	VN②	VN③	平均值
RKLT-FLIF	4.12	4.19	4.16	4.16
算法名	SW①	SW②	SW③	平均值
RKLT-FLIF	4.06	4.21	4.17	4.15

本文提出的RKLT-FLIF方案的另一个优点是能够根据图像的均匀程度（噪声情况）在复杂FLIF与简单哈夫曼编码器之间切换，从而加快压缩速度，缩短压缩时间。本文增加了一个对照实验组来分析自适应切换压缩器策略的效果，对照实验组禁用了哈夫曼编码器。针对上述的高光谱图像测试样本，在相同的测试实验平台上，本文做了相关实验统计出了RKLT-FLIF方案与RKLT-FLIF（无哈夫曼编码）方案完成压缩所耗费的时间，见表3。

表 3　高光谱测试图像的压缩实验耗时结果

Tab.3　Time-consuming results of the compression experiment for hyperspectral test images

算法名	VN①	VN②	VN③	平均
RKLT-FLIF	224.0	226.0	225.0	225.0
RKLT-FLIF（有哈夫曼编码）	50.5	52.3	54.3	52.4
算法名	SW①	SW②	SW③	平均
RKLT-FLIF	268.7	273.9	273.6	272.0
RKLT-FLIF（有哈夫曼编码）	251.7	219.2	244.5	238.5

3.2　结果分析

通过表2中的bpp结果，实验结果表明，针对可见光bpp达到4.16，短波红外bpp达到4.15。据文献[13]显示，本文算法相比于C-DPCM更具通用性。C-DPCM算法的原理是先对高光谱bpp图像进行聚类，聚类算法通常是*k*-Means或LBG聚类算法，再针对不同的类别的图像块分别进行预测压缩[14]。然而，图像聚类算法有个明显的缺陷，难以确定合适的类别数，过多或过少的类别都会降低压缩性能，C-DPCM（order 20）括号里的order 20表明算法的聚类数目为20[15]，20类的数量是经过多次试参才确定的，这意味着若更换测试样本，C-DPCM类算法性能可能会出现下滑，与C-DPCM类算法相比，本算法的优点是无须根据输入图像特点手动调整任何参数，通用性更强。

本文提出的RKLT-FLIF方案的另一个优点是能够根据图像的均匀程度（噪声情况），在复杂FLIF与简单哈夫曼编码器之间切换，从而加快压缩速度，缩短压缩时间。本文增加了一个对照实验组来分析自适应切换压缩器策略的效果，对照实验组禁用了哈夫曼编码器。针对上述的高光谱图像测试样本，在相同的测试实验平台上，本文做了相关实验统计出了RKLT-FLIF方案与RKLT-FLIF（无哈夫曼编码）方案完成压缩所耗费的时间，见表3。从表3可以看出：针对可见光波段图像，RKLT-FLIF（无哈夫曼编码）的耗时平均值接近于RKLT-FLIF的5倍；针对短波红外波段图像，RKLT-FLIF（无哈夫曼编码）的耗时平均值接近于RKLT-FLIF的1.14倍。因此，实验结果表明，自适应切换压缩器策略的加速效果可以，加快了压缩速度，同时又没有降低压缩质量。

4　结束语

本论文提出了基于KLT和FLIF算法的高光谱图像无损压缩方法，高光谱图像先经过可逆整数KLT去除图像的谱间冗余信息，再由谱内压缩器FLIF去除图像空间冗余信息得到压缩码流。FLIF算法中的MANIAC决策树能够在压缩过程中适应输入图像的特征，MANIAC决策树的每个叶子节点都维护着一个实际上下文环境和两个虚拟上下文环境，每编码完一个像素便更新一次上下文环境，叶子节点随着压缩进度逐渐分裂从而增加了上下文环境的数量，谱间去相关变换与多上下文环境的设计使得方案取得的实验结果还是不错的。C-DPCM类算法需调参确定合适聚类数目，而多上下文环境设计的自适应能力使得本算法更具通用性和实用性。本文还提出了多压缩器灵活切换的策略，通过逐通道计算图像均匀性指标SR，针对高SR图像使用复杂的FLIF压缩器，对低SR图像切换到简单的哈夫曼编码器，在可见光波段将压缩时间平均缩短为原来的1/5。

参考文献

[1] SHAWN H, LEILA S. Rodríguez. Fast piecewise linear predictors for lossless compression of hyperspectral imagery [C]// IEEE International Geoscience and Remote Sensing Symposium. 2004: 309-312.

[2] WU J J, KONG W Q, JARNO M, et al. Lossless compression of hyperspectral imagery via clustered differential pulse code modulation with removal of local spectral outliers [J]. IEEE Signal Processing Letters, 2015, 22: 2194-2198.

[3] LUO J Q, WU J J, ZHAO S H, et al. Lossless compression for hyperspectral image using deep recurrent neural networks [J]. International Journal of Machine Learning and Cybernetics, 2019, 10: 2619-2629.

[4] GOETZ A F H, SRIVASTAVA V. Mineralogical mapping in the cuprite mining district [M]. CA: JPL Publication, 1985: 85-141.

[5] GOETZ A F H. Three decades of hyperspectral remote sensing of the earth: a personal view [J]. Remote Sensing of Environment, 2009, 113: 5-16.

[6] VANE G, GOETZ A F H, WELLMAN J B. Airborne imaging spectrometer: a new tool for remote sensing [J]. IEEE Transactions on International Geoscience and Remote Sensing, 1984, 22: 546-549.

[7] VANE G, GREEN R O, CHRIEN T G, et al. The airborne visible/infrared imaging spectrometer (AVIRIS) [J]. Remote Sensing of Environment, 1993, 44: 117-126.

[8] GOETZ A F H, HERRING M. The high resolution imaging spectrometer (HIRIS) for EOS [J]. IEEE Transactions on Geoscience and Remote Sensing, 1989, 27: 136-144.

[9] GOETZ A F H. Hyperspectral imaging and quantitative remote sensing [J]. Land Satellite Information in the Next Decade, 1995, 180: 1-10.

[10] ABRAMS M J, ASHLEY R P, ROWAN L C, et al. Mapping of hydrothermal alteration in the cuprite mining district, nevada, using aircraft scanner imagery for the 0.46-2.36 μm spectral region [J]. Geology, 1977, 5: 713-718.

[11] CLARK R N, KING T V V, KLEJWA M, et al. High spectral resolution reflectance spectroscopy of minerals [J]. Journal Geophysical Research: Solid Earth Planets, 1990, 95: 12653-12680.

[12] PAINTER T H, ROBERTS D A, GREEN R O, et al. The effect of grain size on spectral mixture analysis of snow-covered area from AVIRIS data [J]. Remote Sensing of Environment, 1998, 65: 320-332.

[13] SWAYZEKATHLEEN G A, SMITHROGER S, CLARKSTEPHEN N, et al. Using imaging spectrometry to map acidic mine waste [J]. Environmental Science and Technology, 2000, 34: 47-54.

[14] ABOUSLEMAN G P , ARCELLIN M W, HUNT B R. Compression of hyperspectral imagery using the 3-D DCT and hybrid DPCM/DCT [J]. IEEE Trans Geosci Remote Sensing, 1995, 33:26-34.

[15] MIELIKAINEN J, HUANG B M. Lossless compression of hyperspectral images using clustered linear prediction with adaptive prediction length [J]. IEEE Geoscience and Remote Sensing Letters, 2012, 9 (6):1118-1121.