

基于多波段谱间预测的高光谱图像无损压缩算法

孙蕾 罗建书

(国防科学技术大学理学院 长沙 410073)

摘要: 该文提出一种基于多波段谱间预测的高光谱图像无损压缩方案。首先,充分考虑到随着高光谱图像谱间分辨率的提高,其谱间相关性也越来越强烈,推导出由多个波段对当前波段做线性预测的预测器系数,然后给出快速计算求解预测器系数的算法。对 AVIRIS 图像进行压缩,实验结果表明,该算法压缩比高,运算速度快,具有极高的实用价值。

关键词: 谱间预测; 高光谱图像; 无损压缩

中图分类号: TP751.2

文献标识码: A

文章编号: 1009-5896(2007)12-2876-04

Hyperspectral Image Lossless Compression Algorithm Based on Multi-band Prediction

Sun Lei Luo Jian-shu

(Academy of Sciences, National University of Defense and Technology, Changsha 410073, China)

Abstract: A hyperspectral image lossless compression algorithm based on multi-band prediction is proposed in this paper. Considering that as the spectral resolution of hyperspectral images increases, its spectral correlation becomes more and more significant, a group of prediction coefficients are derived, which are able to make use of multi-band to predict the current band linearly. And then a fast algorithm is given to solve the coefficients of the predictor. Experiments show that, with AVIRIS, the ratio of the algorithm is competitive and it runs fast, so it works efficiently in practice.

Key words: Spectral prediction; Hyperspectral image; Lossless compression

1 引言

60年代以来,遥感技术迅猛发展。遥感成像的光谱范围从可见光扩展到红外,紫外甚至更广的范围去。随着成像仪空间和谱间分辨率的增加,遥感数据急剧的增加。遥感数据的传输和储存给遥感领域带来了巨大的压力。

由于遥感信息非常宝贵,同时对压缩信息恢复的要求较高,因而人们对于无损或近无损压缩给予更高的关注。本文研究的是遥感图像的无损压缩。目前,对高光谱遥感图像的无损压缩方法很多,主要有基于预测,变换,矢量量化,以及将多种方法相结合进行压缩。就以上3大类算法而言,基于变换的方法花费的时间较多,而且尽管有提升格式^[1]使其运算量变小,并实现整数到整数的映射,但这些方法主要还是基于渐进式的编码,它们在无损压缩中并没有明显的优势。矢量量化^[2]是高光谱图像压缩的理想方法,在压缩性能上接近理论上图像的最小熵。但是由于矢量量化具有较高的复杂性,计算量随着矢量维数的增加呈指数增长,同时它依赖于图像本身,不同的图像需要构造不同的码书,严重影响了压缩的实时性要求。所以这里对基于预测的方法给予了高度的重视。目前已经有不少学者对基于预测的方法进行研

究,Memon^[3]提出预测树的概念,对原始图像用相邻像素预测,得到差分图,由差分图生成的树称为预测树。其主要缺点是需要附加传输差分树的信息。如果波段相隔远了,预测效果就会变差;如果对相隔较远的波段重新计算差分树,又将增加差分树的传输信息。因而基于预测树的压缩方法的压缩比受到一定的制约。JPEG, LOCO-I^[4]是无损压缩中效率较高的方法,但是它们均适用于二维图像编码。由于它们没有利用到高光谱图像谱间强烈的相关性,并不适用于三维图像。最近, Miguel等^[5]提出的最佳排序(optimal ordering)算法对AVIRIS图像的压缩比均超过了3,其主要思路是用第 k 波段预测第 l 波段($k=1, \dots, L, l=1, \dots, L, L$ 表示波段数),将每一次预测残差进行bzip2压缩,然后从各种结果中选一组最好的作为最终的压缩比,该运算量为 $O(L^2)$ 。尽管最佳排序算法压缩比高,但对于224波段的AVIRIS而言,在运算量方面将花费巨大的代价,实际中不具有可行性。

本文提出一种基于三维预测的高光谱图像无损压缩的快速算法。高光谱图像是对同一地物不同波段的成像,因此不同波段的图像具有相似的轮廓;同时由于成像仪的分辨率越来越高,谱间相关性也越来越大。因此应该将压缩的重点放在去除谱间相关性上。用更多个波段去预测某一波段的灰度值,而不局限于前人所做的仅用一个波段预测下一个波段。本文首先用前 $m-1$ 个波段预测第 m 个波段的灰度值,由此设计出 $m(m \geq 2)$ 阶最佳线性预测器。而后给出了求解

2006-04-10 收到, 2006-09-21 改回

国家自然科学基金(60572135)资助课题

预测器矩阵系数的快速算法。并由此结合 JPEG-LS 设计出一种基于三维预测的高光谱图像无损压缩算法。实验表明，该算法运算速度快，压缩比高，具有很强的应用前景。

2 谱间预测算法

2.1 多阶线性预测器的设计

高光谱图像的纹理细节丰富，仅仅靠去除谱内相关是远远不够的。随着光谱分辨率的增大，相邻波段光谱图像的像素值变得很相近；同时不同波段的图像所涉及的地物目标相同，它们的空间拓扑结构也相同。因而采用 DPCM 预测的方法去除谱间相关性，可以取得非常好的压缩效果。将 DPCM 预测方法运用于压缩的主要思想是：差分信号 Δ_n 是离散输入信号 x_n 和预测器输出的估算值 \hat{x}_n 之差，如果 Δ_n 的值远远小于 x_n ， Δ_n 就可以用较少的比特数表示，从而达到压缩的效果。

以前的文献中多采用二阶线性预测，即只用前一个波段图像预测后一个波段图像，然后结合其它方法(如波段排序^[5]，整数小波变换^[6]等方法)来提高压缩比。但是这样使得运算量急剧的增加，而压缩比的提高却很有限。以下是本文的预测器设计方法及快速计算方法。

为了使预测残差值尽可能小，考虑 $m+1$ 阶线性最优预测器。设图像的宽度为 M ，高度为 N ，波段数为 L ，用 $x_{n,i,j}$ 表示图像第 n 波段第 i 行第 j 列的像素的灰度值， $\hat{x}_{n,i,j}$ 表示由第 $n-1, n-2, \dots, n-m$ 个波段的像素灰度值预测第 n 波段同一位置上的灰度值。用向量 $\mathbf{a}_n = (a_n^{(1)}, a_n^{(2)}, \dots, a_n^{(m+1)})$ 表示由前 m 个波段的像素灰度值预测第 n 波段同一位置上的灰度值的 $m+1$ 个预测系数，则预测值

$$\hat{x}_{n,i,j} = a_n^{(1)}x_{n-1,i,j} + a_n^{(2)}x_{n-2,i,j} + \dots + a_n^{(m)}x_{n-m,i,j} + a_n^{(m+1)}$$

整幅图像的预测误差平方均值为

$$\epsilon_n^2 = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (x_{n,i,j} - a_n^{(1)}x_{n-1,i,j} - a_n^{(2)}x_{n-2,i,j} - \dots - a_n^{(m)}x_{n-m,i,j} - a_n^{(m+1)})^2$$

要使预测均方差达到最小，则必须满足：

$$\begin{cases} \frac{\partial \epsilon_n^2}{\partial a_n^{(1)}} = -2[r(n, n-1) - a_n^{(1)}r(n-1, n-1) - a_n^{(2)}r(n-2, n-1) - \dots - a_n^{(m)}r(n-m, n-1) - a_n^{(m+1)}u(n-1)] = 0 \\ \frac{\partial \epsilon_n^2}{\partial a_n^{(2)}} = -2[r(n, n-2) - a_n^{(1)}r(n-1, n-2) - a_n^{(2)}r(n-2, n-2) - \dots - a_n^{(m)}r(n-m, n-2) - a_n^{(m+1)}u(n-2)] = 0 \\ \vdots \\ \frac{\partial \epsilon_n^2}{\partial a_n^{(m)}} = -2[r(n, n-m) - a_n^{(1)}r(n-1, n-m) - a_n^{(2)}r(n-2, n-m) - \dots - a_n^{(m)}r(n-m, n-m) - a_n^{(m+1)}u(n-m)] = 0 \\ \frac{\partial \epsilon_n^2}{\partial a_n^{(m+1)}} = -2[u(n) - a_n^{(1)}u(n-1) - a_n^{(2)}u(n-2) - \dots - a_n^{(m)}u(n-m) - a_n^{(m+1)}] = 0 \end{cases}$$

其中 $r(k, l) = r(l, k) = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} x_{k,i,j}x_{l,i,j}$ ， $u(k) = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} x_{k,i,j}$

将以上方程整理成矩阵形式如下：

$$\begin{bmatrix} r(n-1, n-1) & r(n-2, n-1) & \dots & r(n-m, n-1) & u(n-1) \\ r(n-1, n-2) & r(n-2, n-2) & \dots & r(n-m, n-2) & u(n-2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ r(n-1, n-m) & r(n-2, n-m) & \dots & r(n-m, n-m) & u(n-m) \\ u(n-1) & u(n-2) & \dots & u(n-m) & 1 \end{bmatrix} \begin{bmatrix} a_n^{(1)} \\ a_n^{(2)} \\ \vdots \\ a_n^{(m)} \\ a_n^{(m+1)} \end{bmatrix} = \begin{bmatrix} r(n, n-1) \\ r(n, n-2) \\ \vdots \\ r(n, n-m) \\ u(n) \end{bmatrix} \quad (1)$$

记为 $\mathbf{R}\mathbf{a}_n = \mathbf{b}$

这是一组有关 $\mathbf{a}_n = (a_n^{(1)}, a_n^{(2)}, \dots, a_n^{(m+1)})$ 的线性方程，求解该方程组即可得到所需要的预测器系数。

2.2 系数矩阵的快速算法

由于前面 m 个波段可以用来预测的波段不到 m 个，所以预测算法应该分成两个部分：对第 2 至第 $m+1$ 个波段的预测，以及对第 $m+2$ 及其以后的波段的预测。

2.2.1 第 2 至第 $m+1$ 个波段的预测 设当前需要预测的波段为 n ， $n = 2, \dots, m$ ，由于第 n 波段之前的波段个数少于 m ，所以只能用 n 阶线性预测，即用第 $1, 2, \dots, n-1$ 波段来预测第 n 波段。因此预测器求解的方法和 $m+1$ 阶预测略有不同。其预测值变为

$$\hat{x}_{n,i,j} = a_n^{(1)}x_{n-1,i,j} + a_n^{(2)}x_{n-2,i,j} + \dots + a_n^{(n-1)}x_{1,i,j} + a_n^{(n)}$$

用类似于 2.1 节方法推得 n 阶线性预测矩阵为

$$\begin{bmatrix} r(n-1, n-1) & r(n-2, n-1) & \dots & r(1, n-1) & u(n-1) \\ r(n-1, n-2) & r(n-2, n-2) & \dots & r(1, n-2) & u(n-2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ r(n-1, 1) & r(n-2, 1) & \dots & r(1, 1) & u(1) \\ u(n-1) & u(n-2) & \dots & u(1) & 1 \end{bmatrix} \begin{bmatrix} a_n^{(1)} \\ a_n^{(2)} \\ \vdots \\ a_n^{(n-1)} \\ a_n^{(n)} \end{bmatrix} = \begin{bmatrix} r(n, n-1) \\ r(n, n-2) \\ \vdots \\ r(n, 1) \\ u(n) \end{bmatrix} \quad (2)$$

将该矩阵记为 $\mathbf{A}_{n \times n} \mathbf{X}_{n \times 1} = \mathbf{B}_{n \times 1}$ 。如果直接求矩阵 $\mathbf{A}_{n \times n}$ 及 $\mathbf{B}_{n \times 1}$ ，运算量为 $O(n^2)$ 。事实上，求解当前系数矩阵是可以利用前一波段的系数矩阵，从而将运算量降为 $O(n)$ 。证明如下：已知第 n 波段的预测系数矩阵 $\mathbf{A}_{n \times n}$ 及 $\mathbf{B}_{n \times 1}$ ，预测 $n+1$

波段时, 相应的矩阵变为

$$\begin{bmatrix} r(n,n) & r(n-1,n) & \cdots & r(1,n) & u(n) \\ r(n,n-1) & r(n-1,n-1) & \cdots & r(1,n-1) & u(n-1) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ r(n,1) & r(n-1,1) & \cdots & r(1,1) & u(1) \\ u(n) & u(n-1) & \cdots & u(1) & 1 \end{bmatrix} \begin{bmatrix} a_{n+1}^{(1)} \\ a_{n+1}^{(2)} \\ \vdots \\ a_{n+1}^{(n)} \\ a_{n+1}^{(n+1)} \end{bmatrix} = \begin{bmatrix} r(n+1,n) \\ r(n+1,n-1) \\ \vdots \\ r(n+1,1) \\ u(n+1) \end{bmatrix} \quad (3)$$

比较式(2)和式(3), 则式(3)可以分块写成如下矩阵形式

$$\begin{bmatrix} \mathbf{D}_{1 \times 1} & \mathbf{B}_{n \times 1}^T \\ \mathbf{B}_{n \times 1} & \mathbf{A}_{n \times n} \end{bmatrix} \mathbf{X}_{(n+1) \times 1} = \mathbf{C}_{(n+1) \times 1} \quad (4)$$

上式表明待解方程的系数有一大部分是利用前面已知结果, 只有 $\mathbf{D}_{1 \times 1}$ 和 $\mathbf{C}_{(n+1) \times 1}$, 共 $n+2$ 个系数是要求的, 所以运算量为 $O(n)$ 。

2.2.2 对第 $m+2$ 及其以后的波段的预测 由式(2)看出, $m+1$ 阶预测的主要运算量集中在求解系数矩阵 \mathbf{R} 和 \mathbf{b} 上。因为 \mathbf{R} 是 $(m+1) \times (m+1)$ 矩阵, 每次预测一个新的波段, 所要求的系数运算量为 $O(m^2)$ 。下面提出一种快速算法求解系数矩阵, 该算法能将运算量降低为 $O(m)$ 。

计算完第 n 个波段的预测系数后, 需要计算第 $n+1$ 波段的预测系数。用 $n+1$ 代替式(2)中的 n , 得到以下方程组:

$$\begin{bmatrix} r(n,n) & r(n-1,n) & \cdots & r(n-m+1,n) \\ r(n,n-1) & r(n-1,n-1) & \cdots & r(n-m+1,n-1) \\ \vdots & \vdots & \ddots & \vdots \\ r(n,n-m+1) & r(n-1,n-m+1) & \cdots & r(n-m+1,n-m+1) \\ u(n) & u(n-1) & \cdots & u(n-m+1) \end{bmatrix} \begin{bmatrix} a_{n+1}^{(1)} \\ a_{n+1}^{(2)} \\ \vdots \\ a_{n+1}^{(m)} \\ a_{n+1}^{(m+1)} \end{bmatrix} = \begin{bmatrix} r(n+1,n) \\ r(n+1,n-1) \\ \vdots \\ r(n+1,n-m+1) \\ u(n+1) \end{bmatrix} \quad (5)$$

观察可以发现, 式(5)中有很多系数是在式(1)中出现的。因此只要将式(1)中出现的系数移到式(5)中相应的位置即可, 从而省掉很大的计算量。

将式(1)的各个矩阵划分成如下分块矩阵

$$\begin{bmatrix} \mathbf{A}_{(m-1) \times (m-1)} & \mathbf{B}_{(m-1) \times 1} & \mathbf{C}_{(m-1) \times 1} \\ \mathbf{B}_{(m-1) \times 1}^T & \mathbf{D}_{1 \times 1} & \mathbf{E}_{1 \times 1} \\ \mathbf{C}_{(m-1) \times 1}^T & \mathbf{E}_{1 \times 1}^T & \mathbf{1} \end{bmatrix} \mathbf{a}_n = \begin{bmatrix} \mathbf{F}_{(m-1) \times 1} \\ \mathbf{G}_{1 \times 1} \\ \mathbf{H}_{1 \times 1} \end{bmatrix}$$

比较式(5), 得

$$\begin{bmatrix} \mathbf{J}_{1 \times 1} & \mathbf{F}_{(m-1) \times 1}^T & \mathbf{H}_{1 \times 1}^T \\ \mathbf{F}_{(m-1) \times 1} & \mathbf{A}_{(m-1) \times (m-1)} & \mathbf{C}_{(m-1) \times 1} \\ \mathbf{H}_{1 \times 1} & \mathbf{C}_{(m-1) \times 1}^T & \mathbf{1} \end{bmatrix} \mathbf{a}_{n+1} = \mathbf{K}_{(m+1) \times 1} \quad (6)$$

在这个新的矩阵中, 只有 $\mathbf{J}_{1 \times 1}$ 和 $\mathbf{K}_{(m+1) \times 1}$ 是待求的, 因此每次预测一个新的波段, 所要求的系数运算量仅为 $O(m)$ 。

2.3 压缩编码算法设计

2.3.1 基于 LOCO-I 的压缩模式 JPEG-LS 对于除去谱间相关性的残差图像, 本文使用 JPEG-LS 压缩标准来去除空间相关性。JPEG-LS 是 ISO 和 IEC 合作开发的一种无损压缩标准, 压缩效率较高。由于 JPEG-LS 的编码是基于预测的, 设计简单而快速。它的原理主要基于 LOCO-I^[4] 思想: 对于一幅图像, 先考察当前像素的几个已出现过的邻近点, 用其作为当前像素的上下文, 用上下文结合“边缘检测”的试探法来进行预测, 并根据预测误差的概率分布用 Golomb 码字来编码预测误差。当对含有大片的相似区域的图像进行编码时, JPEG-LS 还提供了另外一种游程模式, 提高了软件的执行速度。

2.3.2 压缩编码算法设计

(1) 对第 1 幅图像不进行预测;

(2) 对第 n 幅图像, $n=2, \dots, m+1$, 用第 1 幅至第 $n-1$ 幅图像预测第 n 个波段的灰度值。具体做法如下: 利用前一组求解预测系数的矩阵公式(2), 求解出预测系数 $(a_n^{(1)}, a_n^{(2)}, \dots, a_n^{(n)})$, 此时预测值为 $\hat{x}_{n,i,j} = \lfloor a_n^{(1)} x_{n-1,i,j} + a_n^{(2)} x_{n-2,i,j} + \dots + a_n^{(n-1)} x_{1,i,j} + a_n^{(n)} \rfloor$, 其中 $\lfloor \cdot \rfloor$ 表示下取整。计算预测误差 $\varepsilon_{n,i,j} = x_{n,i,j} - \hat{x}_{n,i,j}$;

(3) 对第 n 幅图像, $n=m+2, \dots, L$, 用第 $n-m$ 至 $n-1$ 幅图像预测第 n 个波段的灰度值。具体做法如下: 利用前一组求解预测系数的矩阵和公式(6)求得当前所需要的矩阵, 解出预测系数 $\mathbf{a}_n = (a_n^{(1)}, a_n^{(2)}, \dots, a_n^{(m+1)})$, 计算预测值 $\hat{x}_{n,i,j} = \lfloor a_n^{(1)} x_{n-1,i,j} + a_n^{(2)} x_{n-2,i,j} + \dots + a_n^{(m)} x_{n-m,i,j} + a_n^{(m+1)} \rfloor$ 和预测残差 $\varepsilon_{n,i,j} = x_{n,i,j} - \hat{x}_{n,i,j}$;

(4) 找到 $\varepsilon_{n,i,j}$ ($n=2, \dots, L, i=1, \dots, M, j=1, \dots, N$) 的最小值, 记为 ε_{\min} , 将带符号的残差图像映射到非零整数集中, 即令 $\hat{\varepsilon}_{n,i,j} = \varepsilon_{n,i,j} + \varepsilon_{\min}$ ($n=2, \dots, L, i=1, \dots, M, j=1, \dots, N$)。

(5) 对首幅图像及所有残差图像进行 JPEG-LS 编码。

传输过程中除了传输 JPEG-LS 编码的码流, 还要附加传输预测系数 $\mathbf{a}_n = (a_n^{(1)}, a_n^{(2)}, \dots, a_n^{(m+1)})$ ($n=1, \dots, L-1$), 其中 L 表示该高光图像光谱图像的波段数。用 4 个字节(32bit)表示一个浮点数, 这些系数共需要 $32(m+1)L$ bit。由于实验中 $m=2$ 或 $m=3$ 效果较好, 所以预测系数所占的存储空间非常小, 可以忽略不计。

3 实验结果

本文用高光谱遥感图像对该算法进行了测试。实验数据来源于 JPL 实验室的 AVIRIS 高光谱遥感图像 Jasper Ridge, 每个像素用 16bit 表示, 选取图像的大小为 $204 \times 256 \times 224$ (长 \times 宽 \times 波段)。

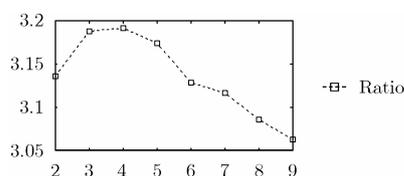


图 1 用不同阶预测的压缩比

图 1 表明并不是预测的阶数越高越好的, 因为尽管高光谱图像谱间有着强烈的相关性, 但是波段之间相隔远了, 相关性自然的会减弱, 尽管运算量增大了, 过多波段的预测却没有取得好的效果。因此本文选择了预测效果最好的 3 或 4 阶。和文献[5]的结果进行比较如下:

表 1 和文献[5]的压缩比比较

	文献[5]最佳排序	4 阶预测
压缩比	3.18	3.192

选择最好的 4 阶预测, 尽管最佳排序算法的压缩比令人满意, 但其运算量为 $O(L^2)$, 这对于 $L = 224$ 波段的 AVIRIS 图像而言, 在运算量方面将花费巨大的代价, 实际中不具有可行性。本文所设计的算法在运算量方面, 远远优于最佳排序算法, 并且压缩比也略微高于最佳排序算法。

4 结束语

本文提出的高光谱图像无损压缩算法在同类算法中具有极强的竞争力。它的主要特点是: (1) 利用高光谱图像谱

间强烈的相关性, 采用多波段进行谱间预测, 提高了压缩比; (2) 设计出求解预测系数的快速算法, 满足实时性要求。

参 考 文 献

- [1] Swelends W. The lifting scheme: a custom-design construction of biorthogonal wavelets [J]. *Journal of Appl. And Comput. Harmonic Analysis*, 1996, 3(2): 186-220.
 - [2] Ryan M J and Pickering M R. An improved M-NVQ algorithm for the compression of hyperspectral data[A]. *Geoscience and Remote Sensing Symposium, GARSS'2000, Proceedings, Honolulu, HI, USA, 2000, 2: 600-602.*
 - [3] Memon N D, Sayood K, and Magliveras S S. Lossless compression of multispectral image data[J]. *IEEE Trans. on Geoscience and Remote Sensing*, 1994, 32(2): 282-289.
 - [4] Weinberger M J, Seroussi G, and Sapiro G. LOCO-I: A low complexity, context-based, lossless image compression algorithm. *Proc.1996 Data Compression Conf., Snowbird, UT, Mar. 1996: 140-149.*
 - [5] Miguel A C, Askew A R, and Chang A, *et al.* Reduced complexity wavelet-based predictive coding of hyperspectral images for FPGA implementation. *Data Compression Conference 2004 (DCC 2004), Snowbird, UT, USA, 23-25 March 2004. IEEE Computer Society 2004, ISBN 0-7695-2082-0 2004: 469-478.*
 - [6] 孙蕾, 罗建书. 基于最佳线性预测器的高光谱图像无损压缩算法. *遥感信息*, 2005, 6: 7-9.
Sun L and Luo J S. Hyperspectral image lossless compression based on optimal linear predictor. *Remote Sensing Information*. 2005, 6: 7-9.
- 孙 蕾: 女, 1980年生, 博士生, 研究方向为小波分析及其在图像压缩中的应用。
罗建书: 男, 1956年生, 教授, 博士生导师, 研究方向为多元小波及图像压缩。