0254-6124/2018/38(6)-944–09 *Chin. J. Space Sci.* 空间科学学报

CHEN Cong, ZHANG Xuequan, ZHOU Shengyu. Improvement and hardware implementation for bit rate control algorithm of JPEG-LS (in Chinese). *Chin. J. Space Sci.*, 2018, **38**(6): 944-952. DOI:10.11728/cjss2018.06.944

JPEG-LS 码率控制算法改进及硬件实现 *

陈 聪[1,2] 张学全[1] 周盛雨[1]

1(中国科学院国家空间科学中心 北京 100190)

2(中国科学院大学 北京 100049)

摘 要 JPEG-LS 码率控制算法是一种基于上下文建模的无损/近无损图像压缩标准, 具有实现方法简单、占用资源少等优点, 适用于卫星图像压缩等硬件资源有限的场合. 该算法在近无损压缩模式下码率不可控制. 在对 JPEG-LS 算法进行分析的基础上, 提出一种新的基于动态码率表的 JPEG-LS 码率控制算法, 并将新算法应用于硬件设计中. 测试结果表明, 新算法比现有算法码率控制精度更高, 压缩后恢复图像信噪比及结构相似比更高. 将该算法在 Xilinx Virtex4 平台上实现, 最高运行频率达 60 MHz, 可以满足星载系统对图像压缩的需求.

关键词 JPEG-LS, 星载图像压缩, 码率控制

中图分类号 V 443

Improvement and Hardware Implementation for Bit Rate Control Algorithm of JPEG-LS

CHEN Cong[1,2] ZHANG Xuequan[1] ZHOU Shengyu[1]

1(*National Space Science Center, Chinese Academy of Sciences, Beijing* 100190)

2(*University of Chinese Academy of Sciences, Beijing* 100049)

Abstract JPEG-LS is a lossless/near-lossless image compression standard based on context modeling. Because of the simplicity and little occupation of resource, it is suitable for satellite image compression where the hardware resource is limited. However, in JPEG-LS near-lossless mode, the bit rate is difficult to control. To solve this problem, a new bit rate control algorithm based on dynamic rate table is proposed based on JPEG-LS. Test results show that the new algorithm is more efficient comparing with the existent bit rate control algorithm. The quality of reconstruction image is also better in the aspects of Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index (SSI). The rate control algorithm can be implemented on Xilinx Virtex4, in which the slice occupancy rate is 21% and the synthesized clock frequency is 60 MHz. It can satisfy the demand of satellite image compression.

Key words JPEG-LS, Satellite image compression, Bit rate control

---

* 国家重大科学仪器设备开发专项项目 (2013YQ030595) 和国家自然科学基金项目 (61474123) 共同资助

2017-11-01 收到原稿, 2018-06-14 收到修定稿

E-mail: applecc@vip.qq.com

0　引言

卫星图像压缩是星上数据处理的一个难点. 与常规静止图像相比, 卫星拍摄的图像种类较多, 相关性较弱, 图像信息熵值高, 冗余度小, 具有明显的不可预测性. 因此, 星载图像压缩系统的关键在于核心压缩算法的选取. 目前卫星图像压缩算法主要包括差分脉冲调制编码 (DPCM)、矢量量化 (VQ) 以及变换编码 (DCT 变换、小波变换等). 在这些算法的应用中, 法国遥感卫星 SPOT2–4 采用 DPCM 方法, 其中 SPOT4 的压缩比约为 1.5, SPOT5 采用 DCT 方法, 压缩比约为 3[1,2]. 但是, 实验表明, 对于大压缩比 (超过 3) 的数据传输系统, DPCM 与 DCT 方法得到的压缩图像质量很难满足要求. 为保证大压缩比时的恢复图像质量, 美国 Eyeglass 卫星等采用矢量量化方法, 压缩比达到 12[3]. 但是, 经典的矢量量化方法计算量与所需存贮量较大, 难以满足星上数据的实时处理要求. 同时, 在大压缩比情况下, 原图像的很多细节都将丢失, 难以满足图像高保真的需求.

JPEG-LS 是 HP 实验室提出的基于 LOCO-I 算法的无损/近无损图像压缩标准, 于 1998 年由 ISO-ITU 正式公布, 用于取代 JPEG 的连续色调静止图像无损压缩模式 [4,5]. JPEG-LS 对图像的压缩模式主要有普通模式和游长模式. 普通模式将当前像素几个已经出现过的近邻作为上下文映射到对应的上下文模式中, 根据对应的上下文模式产生预测值并与当前像素的真实值做差, 最后对该差值进行限长 Golomb-Rice 编码. 游长模式则是在上下文之间的差值满足一定条件后, 对连续的像素值之差不超过定值的像素点进行计数, 并对计数值进行编码 [6]. JPEG-LS 相比 JPEG2000 和 SPIHT 等算法的主要优势在于不需要进行复杂的变换和算术编码, 只利用上下文预测及 Golomb-Rice 编码, 算法简单, 易于硬件实现, 不需要外部存储器, 可以很大程度上减少成本 [7,8]. 但是 JPEG-LS 也存在着码率不易调控的缺点. 为实现按需求的图像大小来压缩, 并且保持良好的图像重建质量, 需要在 JPEG-LS 算法基础上进行恰当的码率控制 [9]. 本文主要通过改动 JPEG-LS 算法结构, 针对其近无损模式下的码率控制进行研究.

1　JPEG-LS 图像压缩算法及现有码率控制算法

JPEG-LS 算法原理如图 1 所示. 首先从图像源读出数据, 对当前像素点已经出现的临近点进行梯度计算. 无损模式下建模时如果上下文的梯度值均为 0, 则进入游长模式计数后编码, 否则当前梯度映射到某一种上下文模式中, 并对利用边界探测器得出的当前像素预测值进行自适应校正, 再将像素真实值与校正后的值做差, 由编码器对该差值进行 Golomb-Rice 编码. 而近无损模式下, 失真量化参数 near 对像素真实值与校正后的值之差进行量化, 编码器对量化后的误差值进行编码.

由以上描述可以看出, 近无损模式添加了失真量化参数 near 对误差值进行量化, 缩小了误差值的范围, 从而使 Golomb 编码的平均码长减小. 由此可以得出以下结论: JPEG-LS 近无损模式下压缩率调整的关键在于失真量化参数 near 的调整. 在现有码率控制算法中, 采用较多的是一阶线性码率控制算法 [10] 和基于先验数据表的码率控制算法 [11]. 一阶

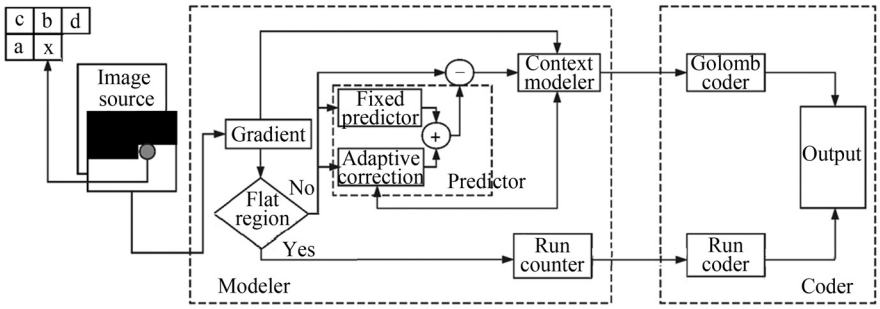


图 1　JPEG-LS 算法原理

Fig. 1　Diagram of JPEG-LS algorithm

线性码率控制算法利用失真量化参数 near 较小时其值与图像压缩比具有近似线性关系, 在图像压缩过程中将图像划分为 N 个 block, 每压缩一个 block, 观察当前码率是否超过较高的码率阈值或小于较低的码率阈值, 如果成立则增加或减少 near 的值, 否则不变, 继续压缩下一个 block, 通过这种操作不断逼近目标码率. 基于先验数据表的码率控制算法则认为, 在 Golomb-Rice 编码中, 当失真量化参数发生变化时, 像素的平均码率变化量只与这两个失真量化参数有关, 与具体的图像复杂度及其他因素基本无关, 从而可以建立一个先验的存有相邻 near 值码率变化量的表, 压缩过程中只需按照一阶线性码率控制算法中的动态调整方法, 在压缩完一个 block 后根据码率调整表进行 near 值的增减. 该方法码率收敛速度和码率精确度均优于一阶线性码率控制算法.

2 JEPG-LS 码率控制算法的改进

2.1 JEPG-LS 码率控制算法分析

基于先验数据表的动态码率调整算法的理论基础是在 Golomb-Rice 编码中, 当失真量化参数由 near1 变化至 near2 时, 像素的平均码率变化量只与 near1 和 near2 有关, 这是由大数据量下 Golomb-Rice 编码的特性决定的 [11]. 然而 JPEG-LS 编码中还存在游长编码部分. 游长编码是针对图像平坦区域, 即像素值变化量不超过容限值的区域进行编码的方法. 如果图像平坦区域较多, 则该先验数据表所显示的压缩码率变化率不准确. 以下对该理论进行验证.

对图 2 和图 3 的图像 (像素尺寸为 512×512) 在 near 值为 0, 1, 2 的情况下分别进行压缩, 统计压缩后的码率以及压缩程序进入游长编码模式的次数, 计算 near 值变化前后压缩码率的差值. 统计及计算结果见表 1 和表 2. 表中码率单位为 bit·pixel^{-1}, 表示每单位像素所需比特数.

从表 1 和表 2 的数据对比可以看出, 如果一个图像在压缩过程中游长编码使用的次数越多, 则相同 near 值下压缩出来的码率比游长编码使用少的图像码率低. 这符合直观感受. 因为游长编码使用的越多, 说明图像平坦区域越多, 也就更容易压缩. 同时也可以看到, 游长编码使用少的图像, near 值发生变化时码率的变化幅度也更大, 反之则 near 值变化更小. 所以在采用先验码率调整表进行动态码率控制时会产生一些误差, 使得调整幅度过大或过小, 造成重构图像的不平滑.

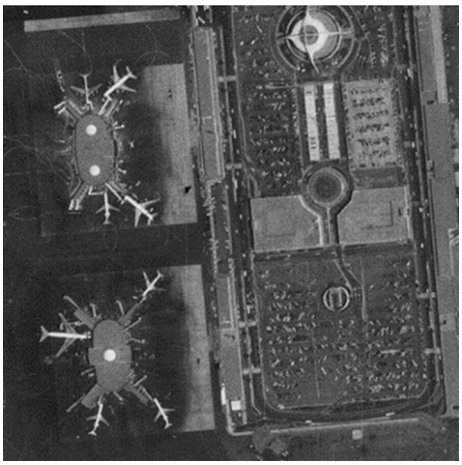


图 2 测试图像 lax

Fig. 2 Test image lax

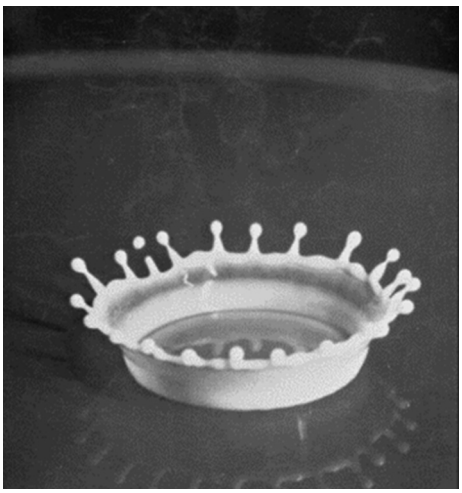


图 3 测试图像 milkdrop

Fig. 3 Test image milkdrop

表 1 测试图像 lax 在不同 near 值下的压缩指标

Table 1 Performance of test image lax under different near value

Value of near	0	1	2
码率/(bit·pixel^{-1})	5.742	4.311	3.739
游长编码个数	0	86	476
码率差/(bit·pixel^{-1})	–	1.431	0.582

表 2　测试图像 milkdrop 在不同 near 值下的压缩指标

Table 2　Performance of test image milkdrop under different near value

Value of near	0	1	2
码率/(bit·pixel^{-1})	3.625	2.671	2.404
游长编码个数	2273	9192	14 004
码率差/(bit·pixel^{-1})	–	0.954	0.267

若要对该方法进行改进, 首先需要对先验数据表进行动态更新. 这意味着每完成一个 block 的压缩后, 不仅要对 near 值进行更新, 也要计算该 block 压缩所用的 near 值与上一个 block 所用 near 值之间的码率差, 从而对码率变化表进行修正.

通过表 1 和表 2 还可以看出, near 值从 0 变化到 1 时码率变化幅度相比其他情况要大得多. 为使码率变化更为平滑, 可在中间设置 near 值为 0.5 的调整节点. JPEG-LS 中误差可以量化为

$$Q_{\mathrm{err}} = \frac{E}{2V_{\mathrm{near}} + 1}. \tag{1}$$

其中 Q_{err} 代表量化后的误差, E 代表原误差. 当 near 值 (为便于计算, 定义数学符号 V_{near}) 取 0.5 时除数为 2, 不会产生除数为非整数的问题.

2.2　基于动态码率调整表的码率控制算法

这里提出的新的码率调整算法是基于动态的码率调整表. 该方法将图像每 n 行划分为一个 block, 每压缩完一个 block, 根据码率调整表调整 V_{near} 压缩下一个 block, 同时更新码率调整表.

基于动态码率调整表的码率控制方法主要包括更新码率调整表和计算下一步失真量化参数两部分, 具体步骤如下.

(1) 设置每一个 block 的行数为 m, 每行包含像素点数目为 p, 目标码率为 b, V_{near} 初始值为 0.5, 根据经验得到初始码率调整表 (见表 3). 该表所示本算法的 V_{near} 上限为 15.

(2) 当压缩至第 n ($n > 1$) 个 block 时, 计算当前 block 的码率 b_1, 图像当前总压缩码率 b_{sum}, 记录当前 near 值为 V_{near1}, 第 $n-1$ 个 block 的 near 值为 V_{near2}, 上一个 block 的码率记为 b_2. 若 V_{near1} 与 V_{near2} 不相等, 则更新码率调整表.

假设 $V_{\mathrm{near2}} > V_{\mathrm{near1}}$, 首先计算上一个 block 的

表 3　初始码率调整表

Table 3　Initial bit rate adjustment

失真量化参数 V_{near}	码率差/(bit·pixel^{-1})
0	–
0.5	1.0
1	0.5
2	0.5
3	0.3
4	0.3
5	0.2
6	0.2
7	0.1
8	0.1
9	0.1
10	0.1
11	0.05
12	0.05
13	0.05
14	0.05
15	0.05

码率 b_2 与当前 block 的码率 b_1 的差值. 由于 V_{near} 值调整带来的码率调整幅度不是均匀变化的, 可进行加权计算得出 V_{near1} 至 V_{near2} 范围内相邻 near 值变化导致的码率差. 根据式 (2) 计算由第 $n-1$ 个 blcok 到第 n 个 block 的码率变化情况, near 值为 i ($V_{\mathrm{near1}} \leqslant i < V_{\mathrm{near2}}$) 时, near 值由当前值变化为下一个 near 值时的码率差. 其中 near_dis 代表由当前码率调整表得出的 V_{near2} 变化至 V_{near1} 时的码率差之和, $b_{\mathrm{near}i}$ 代表当前码率调整表下 near 值由 i 变化为下一个 near 值时的码率差, 有

$$\Delta r_i = \frac{b_2 - b_1}{\mathrm{near_dis}} b_{\mathrm{near}i}. \tag{2}$$

将 Δr_i 与 $b_{\mathrm{near}i}$ 进行算术平均, 可得出调整后 near 值由 i 变化为下一个 near 值时的码率差, 并修改码率调整表对应位置的数值.

(3) 压缩完第 $n+1$ 个 block 后的目标码字数为 $(n+1)mpb$, 当前已经压缩完的 n 个 block 码字数为 $nmpb_{\mathrm{sum}}$. 设第 $n+1$ 个 block 的目标码率为 b_3,

根据式 (3) 计算下一个 block 所需码率 b_3, 有

$$b_3 = \frac{(n+1)mpb - nmpb_{\rm sum}}{mp}. \tag{3}$$

简化后可得

$$b_3 = (n+1)b - nb_{\rm sum}. \tag{4}$$

再计算 b_3 与当前 block 的码率 b_1 的差值, 根据当前 near 值 ($V_{\rm near1}$) 在码率调整表中的位置, 找到能够使码率差最符合要求的新 near 值来满足条件, 使新 near 值作用于下一个 block 的压缩过程中.

3 算法性能评价

算法的性能评价标准采用码率收敛步数, 压缩后恢复图像的峰值信噪比 (PSNR) 以及结构相似性 (SSIM) 三个参数. 这里认为当图像压缩至第 n 个 block 后压缩率达到目标压缩率 ±0.05 范围内, 且直至压缩结束时压缩率不会超出该范围, 则该图像压缩过程需要 n 步完成收敛. 对图 4 所示分辨率为 512 pixel×512 pixel 的 8 bit 太阳图像同时采用两种码率控制算法进行压缩, 并对比两种算法在上述标准下的优劣.

取每一个 block 的行数 n 为 16, 目标码率 $b = 2\,{\rm bit\cdot pixel^{-1}}$, 即目标压缩率为 4, 可以得出两种算法的码率调整曲线, 如图 5 所示. 由图 5 可以看出, 两种算法开始时对码率的调整效果相同, 而当接近目标码率时, 基于动态数据表的码率控制算法可以更快地收敛, 且码率波动更小, 最终结果更接近目标码率. 此外, 基于动态数据表的码率控制算法重建图像的 PSNR (峰值信噪比) 值为 34.8821, SSIM (结构相似度) 值为 0.8405; 基于先验数据表的动态码率控制算法重建图像的 PSNR 值为 33.8258, SSIM 值为 0.7727.

再选择两组 16 bit 太阳局部图片, 一组分辨率为 512 pixel×512 pixel, 另一组分辨率为 1024 pixel ×1024 pixel, 取每一个 block 的行数 n 为 16, 目标码率 $b = 5.33\,{\rm bit\cdot pixel^{-1}}$, 即目标压缩率为 3, 进行算法性能测试. 得到的收敛步数如图 6 和图 7 所示.

对比图 6 与图 7 可以看出, 基于动态数据表的算法收敛步数明显小于基于先验数据表的算法, 并且

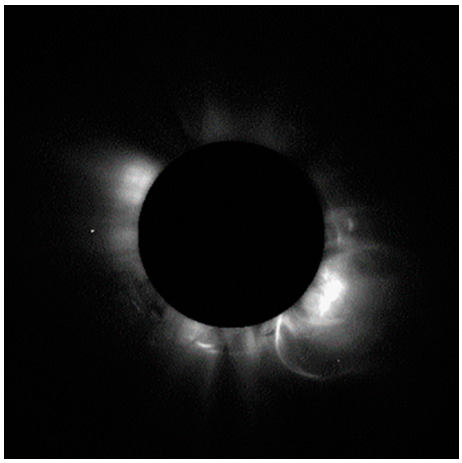


图 4 测试图像 Sun (8 bit)

Fig. 4 Test image Sun (8 bit)

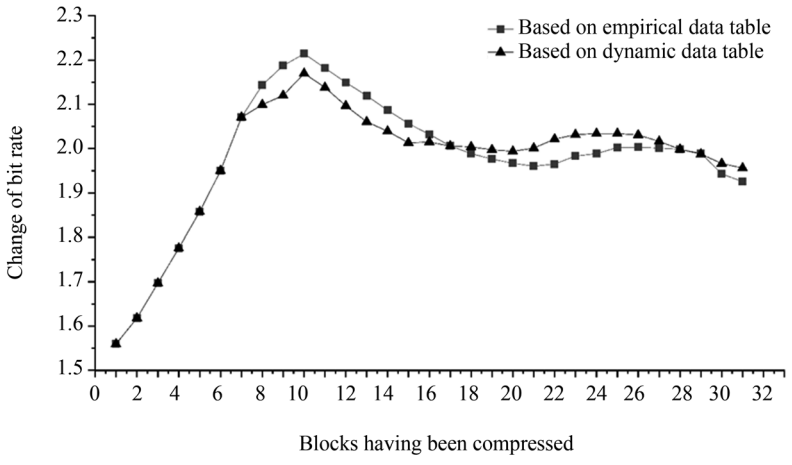


图 5 两种算法的码率调整曲线

Fig. 5 Bit rate control graphs of two algorithms

表 4　图像系列 cor 和 onset 重建图像的 PSNR 值对比

Table 4　Comparison of PSNR between the two algorithms in cor and onset images

图像名称	不同压缩算法重构图形 PSNR 值	
	动态表算法	先验表算法
cor1	40.1527	39.9352
cor2	46.7350	46.2295
cor3	43.9284	43.5900
cor4	47.2992	46.6795
cor5	44.8926	44.3121
cor6	44.9488	44.4505
onset1	45.0393	43.3560
onset2	45.3974	43.5766
onset3	45.3304	43.5421

注　图像 cor 分辨率为 512 pixel×512 pixel, onset 分辨率为 1024 pixel×1024 pixel. 表 5 相同.

表 5　图像系列 cor 和 onset 重建图像的 SSIM 值对比

Table 5　Comparison of SSIM between the two algorithms in cor and onset images

图像名称	不同压缩算法重构图形 SSIM 值	
	动态表算法	先验表算法
cor1	0.9200	0.9067
cor2	0.9675	0.9570
cor3	0.9538	0.9422
cor4	0.9702	0.9594
cor5	0.9581	0.9455
cor6	0.9588	0.9470
onset1	0.9713	0.9552
onset2	0.9726	0.9564
onset3	0.9728	0.9567

其收敛步数的波动更小. 两种算法得到的重建图像 PSNR 值以及 SSIM 值分别列于表 4 和表 5.

由表 4 和表 5 可以看出, 基于动态数据表算法得到的重建图像质量明显优于基于先验数据表的算法, 且图像像素点越多, 差距越明显. 这是由于在同样的目标码率下, 两种算法都是先调大 near 值使码率先接近目标码率, 再对 near 值进行微调使码率收敛至目标码率, 基于动态数据表的算法能更快更准确地通过调整 near 值使压缩码率达到目标码率. 而在近无损压缩模式下, 像素 x 经过误差量化后重建得到的像素值均处于 $[x-V_{\text{near}}, x+V_{\text{near}}]$ 范围内, near 值调整得更快更准确, 使得重建图像与原图像的均方误差更小, 从而 PSNR 值更优.

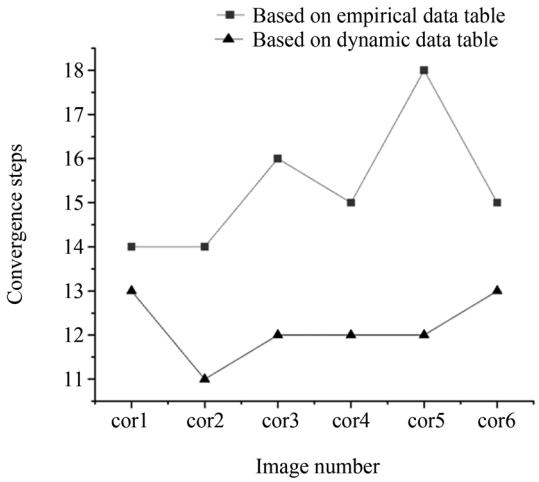


图 6　两种算法对 cor 图像系列的收敛步数对比

Fig. 6　Comparison of convergence steps between the two algorithms in cor image

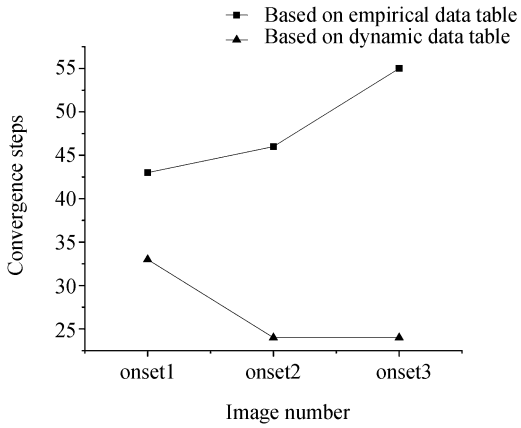


图 7　两种算法对 onset 图像系列的收敛步数对比

Fig. 7　Comparison of convergence steps between the two algorithms in onset image

4 硬件设计中的应用

随着大规模可编程器件的迅速发展, FPGA (Field Programmable Gate Array) 在工程上的应用逐渐普遍起来. 在图像压缩领域, FPGA 与传统器件 (如单片机、PLD 等) 相比, 具有速度快、功耗低、运算能力强、可靠性高及吞吐量大等优点, 在实现图像实时压缩方面具有一定优势 [12]. 这里设计一个基于 FPGA 的 JPEG-LS 近无损实时压缩系统, 并采用基于动态码率表的码率控制方法. 该系统的结构组成如图 8 所示.

由图 8 可以看出, 外部图像数据经数据接口模块缓存后输入编码模块, 编码过程中用到的参数由参数缓存模块存储, 上下文数据由图像缓存模块存储, 压缩时像素点的重建值也输出到图像缓存模块进行缓存, 最后经由码流输出模块整合后输出. 图像压缩系统的核心为编码模块, 其内部结构如图 9 所示.

编码模块的工作流程如下. 外部数据首先经过梯度及 Q 值计算模块处理, 预测 Q 值并判断是否进入游长编码模块. 若进入游长编码模块, 像素值经过编码后直接输入码流输出模块; 若进入正常模式编码, 则经过边缘检测模块得出预测值, 在误差预测及量化模块与原像素值、偏差值分别进行除法运算并融合以得出误差值, 误差值映射后进行 Golomb 编码, 期间进行参数更新, 最后将 Golomb 编码结果输出至码流输出模块. 像素重建值在梯度及 Q 值计算模块进行预测, 在误差映射及参数更新模块进行选择并输出至图像重建值缓存模块.

4.1 梯度计算与 Q 值预测

在近无损压缩方法中, 梯度和 Q 值的计算以及边缘检测依赖于待编码像素已出现的上下文. 其中, 位于待编码像素左侧点 a 的重建值需要通过上一次压缩量化后的误差重建得到. 为实现高效的流水线处理, 这里设计了一种像素重建值预测选择机制, 不需要等待上一次压缩得到的量化后误差再进行本次压缩.

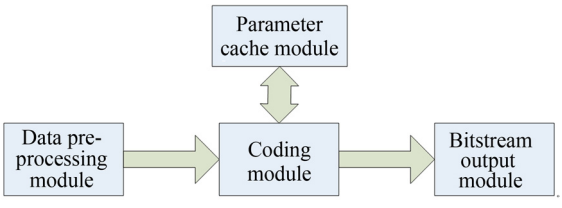


图 8 JPEG-LS 图像压缩系统组成

Fig. 8 Diagram of the JPEG-LS image compression system

梯度计算后量化为索引值 Q 时采用的量化方式如下:

$$Q_i = \mathrm{sign}(D_i) \times \begin{cases} 0, & |D_i| \leqslant V_{\mathrm{near}}; \\ 1, & V_{\mathrm{near}} < |D_i| < T_1; \\ 2, & T_1 \leqslant |D_i| < T_2; \\ 3, & T_2 \leqslant |D_i| < T_3; \\ 4, & |D_i| \geqslant T_3. \end{cases} \tag{5}$$

式中, D_i 代表梯度, Q_i 代表梯度量化后对应的索引值, 参数 T_1, T_2, T_3 设置为

$$\begin{cases} T_1 = (b-7) \times 3 + V_{\mathrm{near}}, \\ T_2 = (b-7) \times 7 + 2V_{\mathrm{near}}, \\ T_3 = (b-7) \times 21 + 3V_{\mathrm{near}}. \end{cases} \tag{6}$$

Q 值的计算公式为

$$Q = 81Q_1 + 9Q_2 + Q_3. \tag{7}$$

其中, b 为待压缩图像单个像素所需的比特数, Q_1 与 Q_2 的值由前行像素值产生, Q 值预测只涉及到 Q_3 的取值. 近无损压缩模式下像素 x 经过误差量化后重建得到的像素值均处于 $[x - V_{\mathrm{near}}, x + V_{\mathrm{near}}]$ 的范围内. 观察式 (5) 可知, 每两个阈值之间的间隔都大于 near, 所以在梯度量化后只会产生三个可能的量化后梯度值 Q_3, $Q_3 - 1$ 和 $Q_3 + 1$, 这里 Q_3 代表由 a 的原值进行梯度计算并量化后得到的索引值. 最终产生三个索引值 Q_1, Q_2 和 Q_3, 可对此设置三个参数缓存模块, 每个模块中含 4 个 RAM 用于存储 4 种参数值. 产生索引值后分别在三个模块中取出压缩过程中需用到的参数, 同时进行误差量化流程. 在误差量化后, 通过上一轮计算所得 a 点的重建值, 得到正确的重建值以及误差量化值, 并以该重建值为依据选择正确的参数更新结果, 写入缓存参数的 RAM 中, 像素重建值也同时写入图像重建值缓存模块中.

4.2 误差量化中的除法运算

由式 (1) 可知, 近无损压缩模式下每一次压缩都需要进行除法运算. 文献 [13] 中采用一种基于查表的除法运算, 将所有可能的除法运算结果分为商和余数都保存在片内 rom 中. 将与误差相关的预测值、原像素值以及 $C[Q]$ 值分别进行该除法运算再进行商

和余数的合成, 即可得到最终误差量化的结果.

由于该文献中输入图像最大为 10 bit, near 值上限为 10, 所以最大只需储存 1024×11=11264 个 10 bit 运算结果. 本文所述系统输入图像上限为 16 bit, 采用此方法则需储存 65 536×11 =720 896 个 16 bit 运算结果, 耗费资源巨大. 这里将 16 bit 除法分为高 8 bit 和低 8 bit 进行运算, 高 8 bit 数和低 8 bit 数的计算结果分别缓存于 ram 中. 先取 16 bit 数中的高 8 bit 进行运算后产生商和余数, 余数与低 8 bit 相加后再进行除法运算, 产生的商与高 8 bit 产生的商相加得到最终的商, 产生的余数为最终的余数. 此方法只需要储存 $256 \times 11 + (256 + 20) \times 11 = 5852$ 个运算结果, 大幅度优化了原有算法.

4.3　码率表的动态调整

编码模块的最后设置一个码率控制模块, 连接码流输出模块与编码模块的梯度计算及量化模块, 动态码率表缓存于该模块中. 编码模块对所产生码流的长度进行计数, 每压缩完一个 block 后将计数值反馈至码率控制模块, 清零后重新开始计数. 该模块首先根据码流长度计算得到该 block 的码率, 码率控制模块依据 2.2 节所述方法更新码率调整表, 其中码率表的调整所用除法也采用查表的方法进行, 除法精确到小数点后两位. 最后根据新的码率调整表修改梯度计算及量化模块处的 near 值. 由于码率的计算以及码率表的调整需要多个时钟周期, 所以在实际的硬件实现中, 需先调整 near 值然后再更新参数.

对该 JPEG-LS 近无损图像压缩系统硬件实现使用的 FPGA 型号为 XC4VSX35, 经综合后该系统的资源占用情况见表 6. 综合后该图像压缩系统最高工作频率可达 60 MHz. 在实际运行过程中, 每秒可处理 60 帧像素分辨率为 1024 pixel×1024 pixel 的 8∼16 bit 图像.

5　结论

通过分析 JPEG-LS 的编码特性, 总结之前关于动态码率控制方法的研究, 改进并提出了一种新的码率控制方法. 实验结果表明, 该算法的码率控制精度以及收敛速度等主要性能指标均超过了现有的码率控制算法, 并且复杂度低, 易于实现, 尤其适用于卫星图像压缩这种带宽受限的情景. 硬件实现后, 综合频率达到 60 MHz, 资源占用合理, 可以满足星载系统的要求.

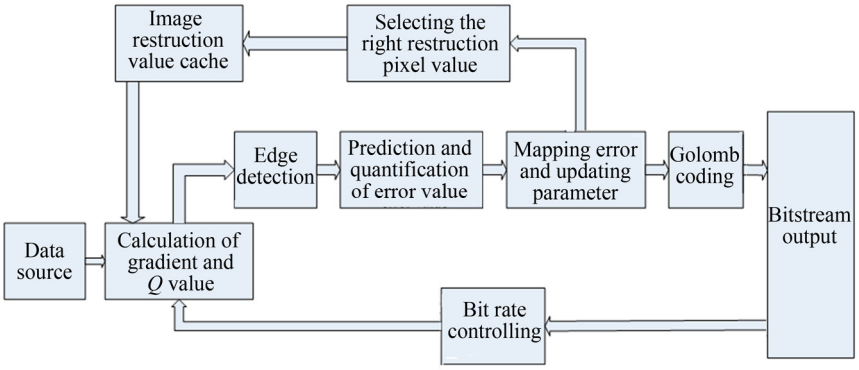


图 9　编码模块系统组成

Fig. 9　Diagram of the coding module

表 6　JPEG-LS 图像压缩 FPGA 系统资源占用情况

Table 6　Device utilization of JPEG-LS image compression system

资源	使用数量	可用数量	利用率/(%)
Slice	7187	15 360	46
LUT	12 094	30 720	39
FIFO16/RAMB16 s	85	192	44

参考文献

[1] ZHAO Jiaxin. Design and Efficient Hardware Implementation of Onboard Satellite Image Compression System [D]. Xi'an: Xidian University, 2010 (赵嘉鑫. 星载图像压缩系统设计与高效硬件实现 [D]. 西安: 西安电子科技大学, 2010)

[2] SUN Xinzhe, Tang Shengquan. The development of optical remote sensing technology and its application to the active tectonics research [J]. *Seismol. Geology*, 2016, **38**(1): 211-220 (孙鑫喆, 唐声权. 光学遥感技术的发展及其在活动构造研究中的应用 [J]. 地震地质, 2016, **38**(1): 211-220)

[3] XIAO Xuemin. Research and Implementation of JPEG-2000 Satellite Image Compression [D]. Changsha: National University of Defense Technology, 2007 (肖学敏. JPEG-2000 卫星图像压缩方法的研究与实现 [D]. 长沙: 国防科学技术大学, 2007)

[4] WEINBERGER M J, S EROUSSI G, S APIRO G. The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS [J]. *IEEE Trans. Image Proc.*, 2000, **9**(8): 1309-1324

[5] RANE S D, SAPIRO G. Evaluation of JPEG-LS, the new lossless and controlled-lossy still image compression standard, for compression of High-resolution elevation data [J]. *IEEE Trans. Geosci. Remote Sens.*, 2001, **39**(10): 2298-2306

[6] LI Hengjian, WANG Lianhai, ZHANG Jiashu. Secure palmprint image coding algorithm for JPEG-LS [J]. *Computer Science*, 2013, **40**(09): 141-146, 181 (李恒建, 王连海, 张家树. 基于 JPEG-LS 的高效掌纹图像安全编码算法 [J]. 计算机科学, 2013, **40**(09): 141-146, 181)

[7] PADONIKOLAKIS M E, KAKAROUNTAS A P, GOUTIS C E. Efficient High-Performance ASIC implementation of JPEG-LS encoder [J]. *J. Real-Time Image Proc.*, 2008, **3**(4): 303-310

[8] MAO Jiafa, NIU Xinxin, YANG Yixian, *et al.* JPEG2000 image maximum steganography payload based on MPUI model [J]. *J. Software*, 2014, **25**(7): 1606-1620 (毛家发, 钮心忻, 杨义先, 等. 基于 MPUI 模型的 JPEG2000 图像最大隐写容量 [J]. 软件学报, 2014, **25**(7): 1606-1620)

[9] CHEN Yigang, DENG Jiaxian, XIE Kaiming. Image compression encryption algorithm based on compression ratio control of JPEG-LS [J]. *J. Comput. Appl.*, 2015, **35**(01): 93-98 (陈益刚, 邓家先, 谢凯明. 基于 JPEG-LS 压缩比控制的图像压缩加密算法 [J]. 计算机应用, 2015, **35**(01): 93-98)

[10] XU Yanling, LIU Bei. Dynamic rate control scheme for JPEG-LS image compression [J]. *Comput. Eng.*, 2008, **34**(7): 238-239, 271 (徐燕凌, 刘蓓. JPEG-LS 图像压缩动态码率控制策略 [J]. 计算机工程, 2008, **34**(7): 238-239, 271)

[11] ZHANG Yi, LEI Jie, LI Yunsong. A novel dynamic rate control algorithm for JPEG-LS based on empirical data table [J]. *J. Electron. Inf. Technol.*, 2014, **36**(4): 823-827 (张毅, 雷杰, 李云松. 一种新的基于先验数据表的 JPEG-LS 动态码率控制算法 [J]. 电子与信息学报, 2014, **36**(4): 823-827)

[12] YANG Haigang, SUN Jiabin, WANG Wei. An overview of FPGA device design technologies [J]. *J. Electron. Inf. Technol.*, 2010, **32**(3): 714-727 (杨海钢, 孙嘉斌, 王慰. FPGA 器件设计技术发展综述 [J]. 电子与信息学报, 2010, **32**(3): 714-727)

[13] ZHANG Yi. Hardware Implementation of Rate Controlled JPEG-LS Near Lossless Image Compression Encoder [D]. Xi'an: Xidian University, 2014 (张毅. 码率可控的 JPEG-LS 近无损图像压缩编码器硬件实现 [D]. 西安: 西安电子科技大学, 2014)