

基于 LOCO—I 算法的星载图像无损压缩的 FPGA 实现

陈 军^{1,2}, 王怀超^{1,2}, 顾晓东¹, 陈晓敏¹

(1 中国科学院 空间科学与应用研究中心, 北京 100190; 2 中国科学院研究生院, 北京 100190)

摘 要: LOCO—I 图像无损压缩算法是 JPEG—LS 标准的核心算法. 文中针对于星载图像无损压缩的具体应用, 充分利用了流水和并行技术, 给出了基于该算法的高速星载图像无损压缩核的设计和实现. 压缩验证系统实验结果表明, 该设计可每周期处理一个像素, 50MHz 频率下数据处理速度可达 400Mbps, 满足星载图像无损压缩的实际要求.

关键词: 无损压缩; LOCO—I 算法; FPGA

中图分类号: TN919

文献标识码: A

文章编号: 1000—7180(2011)11—0169—05

FPGA Application of Onboard Space Lossless Image Compression Based on LOCO—I Algorithm

CHEN Jun^{1,2}, WANG Huai-chao^{1,2}, GU Xiao-dong¹, CHEN Xiao-min¹

(1 Center for Space Science and Applied Research, Chinese Academy of Sciences, Beijing 100190, China;

2 Graduate University of Chinese Academy of Sciences, Beijing 100190, China)

Abstract: LOCO—I lossless image compression algorithm was selected as the core algorithm of JPEG—LS. This paper focuses on specific applications of onboard image lossless compression and presents the FPGA architectural design based on pipeline and parallel techniques. This design was implemented on a Xilinx XC2V1000. The data throughput is 400 Mbps when the clock is 50 MHz. The experimental results indicate that this design can performed in high—speed, and meet the need of real time onboard image processing.

Key words: lossless compression; LOCO—I algorithm; FPGA

1 引言

为降低海量的图像数据对星载存储和下行通信链路带来的压力, 必须进行在轨的图像压缩. 有损压缩算法可以实现高压缩比, 但不能恢复原始图像. 航天器获得的图像数据极其珍贵, 保证图像数据的完整性是不可或缺的. 虽然许多有损压缩算法将无损压缩作为特殊情况, 但复杂且压缩比较低. 为了解决这个矛盾, 许多航天任务中会针对有损和无损同时采用两种独立的压缩算法.

根据^[1], 已经广泛应用的星载图像无损压缩算法有 DPCM, BTC, RICE 及 LOCO—I/JPEG—LS 等. DPCM 是应用比较早的无损压缩算法, 英国萨

瑞大学研制的卫星中多采用 BTC 算法. RICE 算法已被 CCSDS 的 121.1—B^[2-3] 推荐标准采用, 虽然硬件复杂度低, 但压缩比不高. LOCO—I 是其中压缩性能最好的算法, 已经成为主流的星载图像无损压缩算法.

文中提出了一种基于 FPGA 的 LOCO—I 算法的 FPGA 实现方案. 该方案充分利用 LOCO—I 算法的特点, 采用流水技术和并行技术, 满足星载图像压缩对硬件资源和实时性要求.

2 算法简介

LOCO—I 算法全称是图像的低复杂度无损压缩算法^[4]. 该算法是一种基于上下文的预测差分编

码方法,依据待编码像素的局部梯度,通过量化和对称折叠有效减少了上下文数量,提高了统计效率.每种上下文记录四个统计参数,用以消除双边几何分布的偏差,从而减小预测误差.将双边几何分布进行映射得到单边几何分布,找到对应的最优的 Golomb-Rice 码,从而提高压缩性能和减少复杂度. LOCO-I 将编码分为常规模式和游程模式,编码器首先提取上下文像素,计算局部梯度,据此选择进入常规模式或游程模式.

2.1 模式判决

当前点 x 的上下文如图 1 所示. 局部梯度描述了像素点周围的平坦程度,编码模式的选择是基于局部梯度的,定义三个梯度值:

$$D1 = d - b, D2 = b - c, D3 = c - a \quad (1)$$

如果三个值均为零,则编码器进入游程模式;否则,进入常规模式.

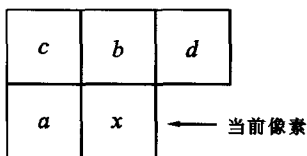


图 1 上下文定义

2.2 常规编码模式

用 $(D1, D2, D3)$ 来表示局部梯度,为了减少上下文的种类,将每个梯度分为大致等概率的区域进行量化,量化结果用 Q_i 表示. 根据对称性假设^[4],有 $P\{\xi = \Delta | \{Q1, Q2, Q3\}\} = P\{\xi = -\Delta | \{-Q1, -Q2, -Q3\}\}$, 通过对称折叠可将上下文个数减半. 如果 $(Q1, Q2, Q3)$ 中第一个非零 Q 是负数,将一标志性变量 $SIGN$ 设置为 -1 ,并将 $(Q1, Q2, Q3)$ 符号翻转为 $(-Q1, -Q2, -Q3)$. 之后将 $(Q1, Q2, Q3)$ 做一一映射得到上下文的索引 Q . 每个上下文需要四个参数 $A[Q], B[Q], C[Q], N[Q]$, 用以统计建模. 其中 A 为偏差绝对值的积累, N 记录对应上下文发生的次数,参数 B 和 C 用以消除模型误差,减小预测偏差,它们在编码结束后进行更新.

预测器分为固定预测部分和自适应预测部分. 固定预测部分是基于边缘准则的非线性预测器,其具体预测值:

$$\hat{x}_{i+1} = \begin{cases} \min(a, b) & \text{if } c \geq \max(a, b) \\ \max(a, b) & \text{if } c \leq \min(a, b) \\ a + b - c & \text{otherwise} \end{cases} \quad (2)$$

自适应预测部分利用参数 C , 对 $\hat{x}_{i+1}$ 进行偏差修正,以使预测偏差符合以零点为中心的双边几何分布.

Golomb-Rice 码是针对单边几何分布的非负整数的一种有效的霍夫曼码,它无需码表且硬件实现简单^[5]. 将预测误差进行单边映射,得到的非负整数近似符合单边几何分布,其编码参数 k 根据上下文参数自适应调整,方法为: $\text{for}(k = 0; (N[Q] \ll k) < A[Q]; k++)$. 具体建模和参数自适应选择原理见^[4].

2.3 游程编码模式

进入游程编码模式后,首先计算游程长度,遇到其他像素或行结束时跳出游程,并对游程长度和中断像素编码. 对游程长度采用分段编码. 用 $RUNcnt$ 表示游程长度,使用一大大小为 32 的入口表 J , 表中的值表示为 rk . 将 $RUNcnt$ 进行分块,块大小为 $rm = 2^k$. 分块编码就是向压缩流中打入比特 1, 每打一个代表相应的块,并从 $RUNcnt$ 中减掉.

如果 $RUNcnt$ 不等于连续的 rm 相加,则先输出 1 位前缀,之后输出 $RUNlen$ 的剩余部分的二进制形式. 如果游程被一个不同的像素值中断,前缀位为 0,如果游程被因行结束终止,则前缀为 1.

编码器的第二个重要任务是编码中断像素,这和常规模式类似.

3 LOCO-I 的 FPGA 实现

为了提高 LOCO-I 算法的吞吐量,满足星载图像无损压缩的实时性要求,文中采用流水设计. 将算法核心部分划分为两条流水线,分别是常规模式流水和游程模式流水. 其中游程模式流水又可细分为游程中断像素编码流水和游程长度编码流水. 为了实现图像大小可调,局部梯度量化参数可调,在相应模块中增加寄存器,可供控制模块配置,从而实现灵活应用. FPGA 实现的整体架构如图 2 所示.

系统整体分成 8 个流水级,第一级进行行缓存,内含计数器,进行图像长宽计数. 最后选择输出上下文像素 a, b, c, d . 第二级计算局部梯度 $D1, D2, D3$, 选择编码模式. 如果进入游程模式,对游程长度进行计数. 常规模式和游程模式均为四个流水级. 常规模式和游程中断像素编码输出的参数直接送入 Golomb-Rice 编码器,而游程长度编码经过一个周期延时后,和 Golomb-Rice 编码器输出的码流同时送入码流组织模块. 码流组织模块将前级输入的不规则码流拼凑成 4 字节长后统一输出.

3.1 上下文像素产生及模式判决

上下文像素产生模块输出上下文像素以及行结束信号,内含计数器以及双口 RAM 控制接口,用于

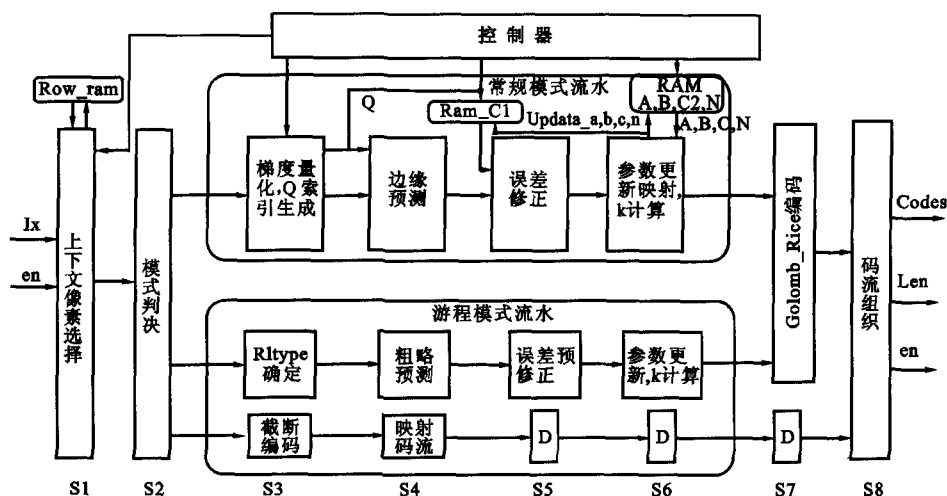


图2 LOCO-I 的 FPGA 实现架构

图像行缓存。像素 x 的上下文像素 a, b, c 分别来自于上个像素及其对应的 b, d , 而 d 需要从行缓存中提取。此外还需要一额外的寄存器, 存储上一行的第一个像素点的上下文 a , 用于配置当前行第一个像素点的上下文 c 。在行结尾时, 输出行结尾信号 END_LINE , 用于下级模式判决模块。

模式判决模块含一 FSM, 共有四个状态:

① 常规模式, 此状态使能常规模式流水, 并输出局部梯度。

② 游程计数, 关闭下级流水, 对游程长度计数, 直至游程结束。

③ 因行结束中断游程, 开启游程模式流水中的截断编码流水线, 输出游程中断模式信号和游程长度。

④ 因不同像素中断游程, 开启游程模式流水, 不仅输出游程中断模式信号和游程长度, 也输出 a, b , 供下级对中断像素进行编码使用。

3.2 常规模式流水

常规模式流水实现上下文建模, 并计算预测残差, 自适应选择 Golomb-Rice 编码参数 k , 将结果送到下级编码模块。该流水线划分为四级, 第一级完成局部梯度量化, 索引 Q 生成, 第二级进行非线性预测, 第三级进行误差修正, 第四级进行误差映射, 编码参数更新, 以及计算 Golomb-Rice 编码参数 k 。上下文参数 A, B, C, N 存储在双口 RAM 中, 第四级将更新后的参数返回到双口 RAM。

划分流水时不能出现上下级流水存在数据依赖的问题, 但分析 LOCO-I 算法可知, 如果两个连续像素具有相同的上下文索引 Q , 即具有相同的上下文, 这时对第二个像素进行预测误差纠正时, 如果上下文参数不能及时更新, 则这时不能及时计算新像

素预测残差, 不能实现真正的全流水^[5]。针对这个问题, 可以是引入倍频时钟^[6], 从而使参数得到及时更新, 但引入新的时钟域, 增大设计复杂度, 带来许多新问题。利用算法本身特点, 即每次参数 C 的更新限制在 $[-1, 1]$ 之间, 故每次更新之后, 参数 C 有三个可能的值, 分别是原值增 1, 减 1 或维持不变。所以当遇到连续两像素的 Q 一样时, 可以在残差修正模块进行预计算, 得到三个可能结果, 在下一级中根据上个像素对 C 的更新情况选择正确的残差修正值即可。具体实现如图 3 所示, 对应于常规模式的后两级流水。

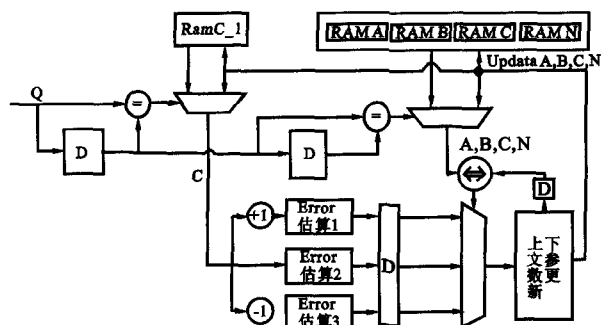
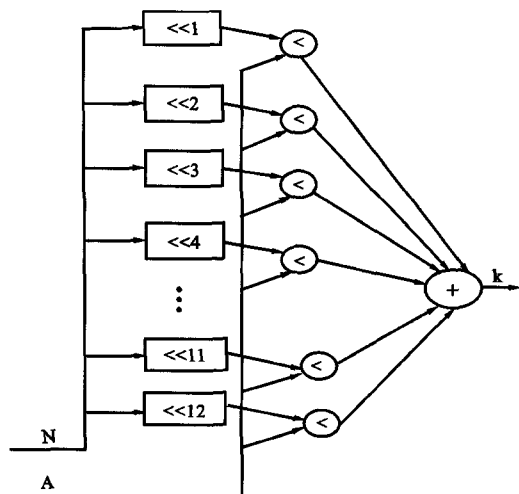


图3 预测误差预计算反馈更新

由于残差修正计算需要参数 C , 而下级中的参数更新也需要参数 C , 故需要两个 ram 来存储 C 。正常情况下, 用从 $RamC_1$ 中获取的参数 C 进行残差修正, 而如果两连续像素的 Q 相同时, 则直接用下级输出的更新后的参数 C 进行残差修正。在参数更新模块中需存储上个像素参数 C 更新情况, 标志三种情况, 以便对上级三个预计算值进行选择。

如上述, Golomb-Rice 编码参数 k 计算为一循环形式, 硬件实现时 A 和 N 位长确定后, 可用并行结构在一个周期实现, 图 4 为具体的实现电路。

图4 Golomb-Rice 编码参数 k 计算

3.3 游程模式流水

游程模式对中断游程像素和游程长度进行编码。中断游程像素编码方法和常规模式类似。对于游程长度的截断编码,为了提高编码速度,考虑用并行方式实现。对于大小为 32 的入口表 $J\{0,0,0,0,1,1,1,1,2,2,2,2,3,3,3,3,4,4,5,5,6,6,7,7,8,9,10,11,12,13,14,15\}$,元素在该表中的指针定义为 $RUNindex$ 。将 J 中元素 rk 对应得到的 rm 累加可得 $L\{1,2,3,4,6,8,10,12,16,20,24,28,36,44,52,60,76,92,124,156,220,284,412,540,796,1308,2332,4380,8476,16668,33052,65820\}$ 。如果图像宽度不超过 2332,只使用编码入口表的 27 个元素即可。算法要求每次对游程长度 $Runcnt$ 进行编码时,均是在上次 $RUNindex$ 处开始进行分块切割。将 $Runcnt$ 和 L 中第 $RUNindex$ 个元素相加的得到 $Runcnt_tmp$,通过与 L 中元素比较可得到新的 $RUNindex$,新旧 $RUNindex$ 的差即是对 $Runcnt$ 划分的块数,也就是要打入码流中 1 的个数。 $Runcnt_tmp$ 减去 $L[RUNindex]$ 得到游程长度 $Runcnt$ 分块后的余量,将其送入下级,根据游程中断模式加上前缀后打入码流。硬件结构如图 5 所示。

4 FPGA 验证及性能分析

文中搭建了验证系统,验证系统由一片 Xilinx 的 XC2V1000 FPGA 完成。验证系统通过 USB 与 PC 进行通信^[7],PC 向验证系统发送图像数据,FPGA 完成压缩后将码流送回 PC,并由其进行解码。

用 LOCO-I 算法和 CCSDS 推荐的 RICE 算法及基于小波的算法对图 6、图 7 卫星图像进行无损压缩。压缩性能结果如表 1 所示。LOCO-I 算法结果明显优于其他两种算法。

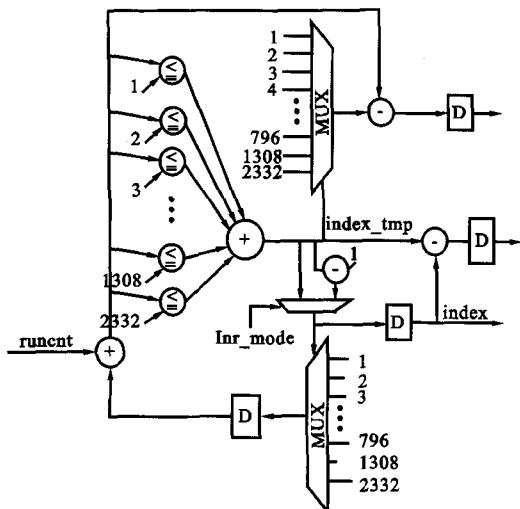


图5 截断编码结构

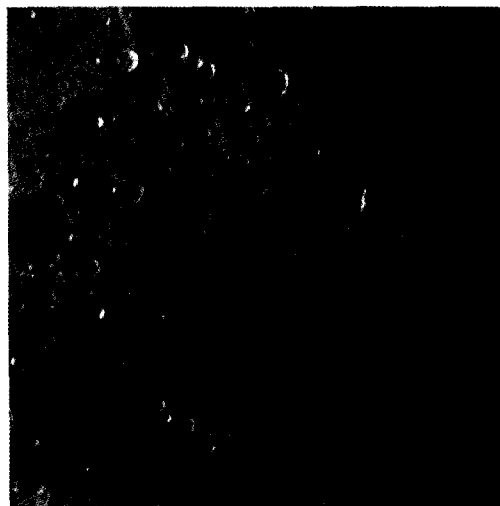


图6 Lunar

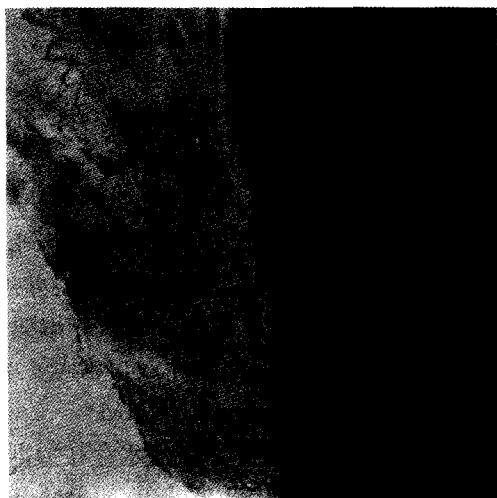


图7 Coastal

表 1 算法性能比较

图像	源数据	压缩后	LOCO	CCSDS	CCSDS
			-I	-Rice	-DWT
Lunar	256KB	139KB	1.84	1.53	1.75
Coastal	256KB	80KB	2.76	2.40	2.48

验证系统时钟为 50MHz,数据吞吐量达到 400Mbps,压缩核占用 FPGA 资源如表 2 所示. 验证结果表明,该设计满足星载图像压缩对实时性和硬件资源的要求.

表 2 所耗 FPGA 资源

	Slice	Slice Flip Flop	LUT	BRAM
个数	2413	1019	4554	6
总个数	5120	10240	10240	40
百分数	47%	9%	44%	15%

5 结束语

文中通过对 LOCO-I 无损压缩算法的深入研究,给出满足星载图像无损压缩需求的 FPGA 实现架构,并进行验证. 该结构充分利用了算法特点,实现了低延时和全流水,经过 8 个时钟延时后可一个时钟处理一个像素,有效提高了数据吞吐量. 整个验证系统的码速率可达 400Mbps,且所耗硬件资源较少,能够满足空间图像实时压缩的要求.

参考文献:

[1] Guoxia Yu, Tanya Vladimirova, Martin N. Sweeting image compression systems on board satellites[J]. Acta Astronautica,2009(64):988-1005.

[2] 王怀超,张学全. 一种适合空间应用的高速图像无损压缩核设计[J]. 微电子学与计算机,2009,26(4):47-49.

[3] Consultative Committee for Space Data Systems. CCSDS 121.0-B-1, Blue Book, issue 1. CCSDS Recommendation for Lossless Data Compression[S]. 1997.

[4] Weinberger M J, Sapiro G and Seroussi G. The LOCO-I Lossless image compression algorithm: Principle and standardization into JPEG-LS[J]. IEEE Trans on Image Processing, 2000(9):1309-1324.

[5] 李晓雯,陈新凯. 低功耗全流水线 JPEG-LS 无损图像编码器的 VLSI 设计[J]. 清华大学学报:自然科学版,2007,47(10):1654-1657.

[6] Pierantonio Merlino, Antonio Abramo. A fully pipelined architecture for the LOCO-I compression algorithm [J]. IEEE transctions on VLSI systems, 2009,17(7):967-971.

[7] 付华杰,刘丽君. 高速 USB IP 核的设计与开发[J]. 微电子学与计算机,2008,25(7):127-129.

作者简介:

陈 军 男,(1986-),硕士研究生. 研究方向为空间图像压缩技术.

(上接第 168 页)

[3] 李涵. 一种改进的聚类方法在异常检测中的应用[J]. 微电子学与计算机,2010,27(8):66-69.

[4] 鄢喜爱,杨金民,常卫东. 基于蜜罐技术的计算机动态取证系统研究[J]. 微电子学与计算机,2010,27(1):135-140.

[5] 诸葛建伟,韩心慧,周勇林,等. HoneyBow: 一个基于高交互式蜜罐技术的恶意代码自动捕获器[J]. 通讯学报,2007,28(12):8-13.

[6] 韩家伟,孟小峰,王静,等. Web 挖掘研究[J]. 计算机研究与发展,2001,38(4):405-414.

[7] Ю. И. ВОРОТНИЦКИЙ,谢金宝. 基于源代码分析的

Web 恶意代码探测方法[J]. 计算机与信息技术,2010(Z1):49-53.

作者简介:

邢 容 男,(1988-),硕士研究生. 研究方向为网络安全、网络管理.

李 俊 男,(1968-),研究员,博士生导师. 研究方向为网络安全、下一代互联网.

荆 涛 男,(1979-),博士研究生,工程师. 研究方向为网络安全、入侵检测.