

基于位平面可预测性的无损图像压缩研究

厉鲁卫¹, 周朔燕², 蔡益宇³

(1. 浙江工业大学信息学院, 杭州 310032; 2. 吉林大学电子科学与工程学院, 长春 130021; 3. 浙江省临海市供电局, 浙江临海 317000)

[摘要] 提出了一种基于嵌入式位平面的静止连续色调图像的无损图像压缩方法: 通过将1幅图像分割成两类位平面(基础层和增强层)使得该图像具有了位平面的可测量性, 并且通过利用平面与平面以及每个平面中各像素之间的相关性减少冗余, 从而获得优秀的压缩性能; 与其他压缩算法的比较表明, 基于嵌入式位平面的无损图像压缩算法由于具有位平面可测量性而体现了巨大的优越性。

[关键词] 信息处理技术; 无损图像压缩; 上下图模型; 嵌入式位平面

[中图分类号] TN911.73 **[文献标识码]** A **[文章编号]** 1009-1742(2005)10-0033-05

近年来, 在图像压缩领域出现了一些优秀的基于上下图模型的编码方法^[1], 其中比较著名的有: CALIC, LOCO-1等^[2,3]。CALIC算法包含了一个自适应的梯度预测器, 以预测误差的能量估计的量化值作为上下图, 对预测误差进行编码, 压缩比较大, 是目前为止一种比较理想的无损压缩方法, 但是它的计算方法较复杂; LOCO-1是JPEG-LS^[4]的标准算法, 它的计算方法比较简单, 但是压缩比要小于CALIC。除此之外, 目前还存在其他一些比较好的无损压缩算法, 如在JPEG2000^[1]标准中, 使用了小波技术, 从而可以实现从有损到无损的渐进压缩。然而, 如果在某一特定环境下, 需要将1幅已经压缩完毕的图像的低位舍去, 从而来适应带宽等条件的限制, 就必须先将该图像解压重建, 然后再进行去除低位图像的压缩步骤。很明显, 这样的压缩步骤大大降低了图像处理的效率。针对此种情况, 笔者提出了一种基于嵌入式位平面的无损压缩算法, 简称LELIC(level embedded lossless image compression), 该算法是一种基于上下图模型的算法, 使用该算法对图像进行压缩后, 如果需要舍去低位, 只要直接将码流中所对应的低位部分取出即可, 存储和计算都简单, 所需要的时间

和资源也少。

1 LELIC的算法

为了说明问题, 先讨论只有1个嵌入位平面的情况, 即将图像分为2个位平面, 称为基础层(高位平面)和剩余层(低位平面)。基础层是通过将每一个像素值 s 除以一个整数 L (L 可以取2, 4, 8, 16, ..., 2^n 中的任何一个)再取整后得到, 记作 $B_L(S) = \lfloor S/L \rfloor$; 剩余层(即嵌入位平面)也叫增强层, 是通过将像素值 S 与 $L \lfloor S/L \rfloor$ 相减得到的, 记作 $r = S - L \lfloor S/L \rfloor$ 。 $L \lfloor S/L \rfloor$ 称为量化像素 $Q_L(S)$ 。

由于所得到的基础层是图像中的最重要的位平面, 且编码过程中完全没有涉及到增强层, 基础层的统计特性与其他全位深图像又非常相似, 所以, 我们可以使用任何一种无损压缩算法对基础层进行压缩(这里使用的是CALIC(context-based, adaptive, lossless image code)自适应预测熵编码。

增强层对应的是一连续色调图像的低位平面。它的压缩方法主要有3个组成部分, 即预测; 根据上下关系建模并进行量化; 进行条件熵编码。

下面定义几个量:

[收稿日期] 2005-03-09; 修回日期 2005-07-05

[基金项目] 浙江省自然科学基金资助项目(M603271), 国家自然科学基金资助项目(39670913)

[作者简介] 厉鲁卫(1949-), 男, 浙江杭州市人, 浙江工业大学信息学院副教授, 主要从事图像处理方面的研究

- (1) \hat{S}_0 当前像素的预测值;
- (2) d, t 测定的上下图值;
- (3) \hat{S}_0 改进后的预测值 (\hat{S}_0, d, t);
- (4) θ 测定的上下图 $\Theta (\hat{S}_0)$;
- (5) 用 S_k 来表示任一像素值 ($k = 1, \dots, 4$)。

1.1 预测

NW	N	NE
W	O	E
SW	S	SE

图 1 像素几何图

Fig.1 Geometric graph of image elements

S_k	S_k	S_k
S_k	\hat{S}_0	$Q_L(S_k)+L/2$
$Q_L(S_k)+L/2$	$Q_L(S_k)+L/2$	$Q_L(S_k)+L/2$

图 2 量化像素几何图

Fig.2 Geometric graph of measurable pixels

如图 1 所示, 对于每一个像素来说, 都有 8 个像素与它邻近。将当前像素的位置记为 0, 邻近的 8 个像素的位置分别为: W, NW, N, NE, E, SE, S, SW。它们的编码顺序和解码顺序都是从左到右, 从上到下。这样的编解码顺序使得当 0 位置的 r_0 被编解码的时候, W, NW, N, NE 位置的像素都已经完成了编解码, 即这些位置的像素值已知。另外, 图像中所有的量化像素值 $Q_L(S)$ 作为一个给定的条件, 便是已知量, 这样, 在 W, NW, N, NE 位置的像素值 $S = Q_L(S) + r$ 和在 E, SE, S, SW 位置的量化像素值 $Q_L(S)$ 就全都知道了 (如图 2)。我们定义一个函数 $f(\cdot)$, W, NW, N, NE 处的像素值用 S_k 表示, 而在其余位置, 则将量化值加上 $L/2$, 来补偿 $Q_L(\cdot)$ 中的偏差。

$$f(S_k) = \begin{cases} S_k & \text{if } k \in \{W, NW, N, NE\} \\ Q_L(S_k) + \frac{L}{2} & \text{otherwise} \end{cases} \quad (1)$$

通过计算 0 位置像素周围最邻近的 4 个像素 (W, E, N, S 位置的像素), 可以得到 0 位置像素值的一个简单的线性预测。

$$\hat{S}_0 = \frac{1}{4} \sum_{k \in \{W, N, E, S\}} f(S_k) \quad (2)$$

然而这个预测经常出现偏差, 于是就会生成一个非零的预测误差: $S_0 - \hat{S}_0$, 需要改进这个预测, 计算公式为:

$$\hat{S}_0 = \text{round}(\hat{S}_0 + \bar{\epsilon}(d, t)) \quad (3)$$

round() 是一个舍入函数 (即将数值四舍五入后取整), $\bar{\epsilon}(d, t)$ 是在给定 d, t 上下图的情况下, 所有预测误差 ($\epsilon = S_0 - \hat{S}_0$) 的平均值。 \hat{S}_0 是一个基于上下图分类, 自适应的非线性预测。

1.2 于上下图建模及量化

所谓上下图, 即本文中的 d, t , 也就是概率中的条件。 \hat{S}_0 是在上下图 d, t 下对 S_0 的预测, 称这样的一组 (\hat{S}_0, d, t) 为一个上下图状态。

典型的自然图像在不同的区域有着不同的统计特性, 如果我们在压缩过程中只建立了一个单一特性的成像模型, 就会使得压缩后的图像产生显著的恶化。而如果像素可以被分离成一系列的基于上下图的上下图状态, 每一个上下图状态的值的统计特性都是非常规则的, 那么, 上下图状态中的一些统计特性 (例如概率分布) 就可以用于相关像素的条件熵编码中。所以, 合适地选择好上下图及其相关统计特性模型, 会大大提高编码的效率。

选择上下图的关键在于要适当地确定上下图的个数: 增加上下图个数, 有助于提高编码效率; 而上下图的增长个数和上下图状态的增长个数又呈指数关系, 所以当上下图个数增加时, 上下图状态个数的增长是十分惊人的, 上下图状态的数量太多, 从这些上下图状态中计算统计特性的工作量也将增加, 从而影响了编码效率, 这种因为上下图的数量太多而导致编码效率下降的现象被称为上下图稀释问题。因此, 一个好的上下图模型必须选择一个最佳方案, 使上下图状态的个数尽可能少的情况下, 上下图的个数尽可能的多。

下面就是有关上下图模型建立的步骤:

第一步, 定义一个基于 d, t 上下图的变量 Δ :

$$\Delta = \sum_{k \in \{W, NW, N, NE, E, SE, S, SW\}} \frac{1}{8} |f(S_k) - \hat{S}_0| \quad (4)$$

$$d = Q(\Delta) \quad (5)$$

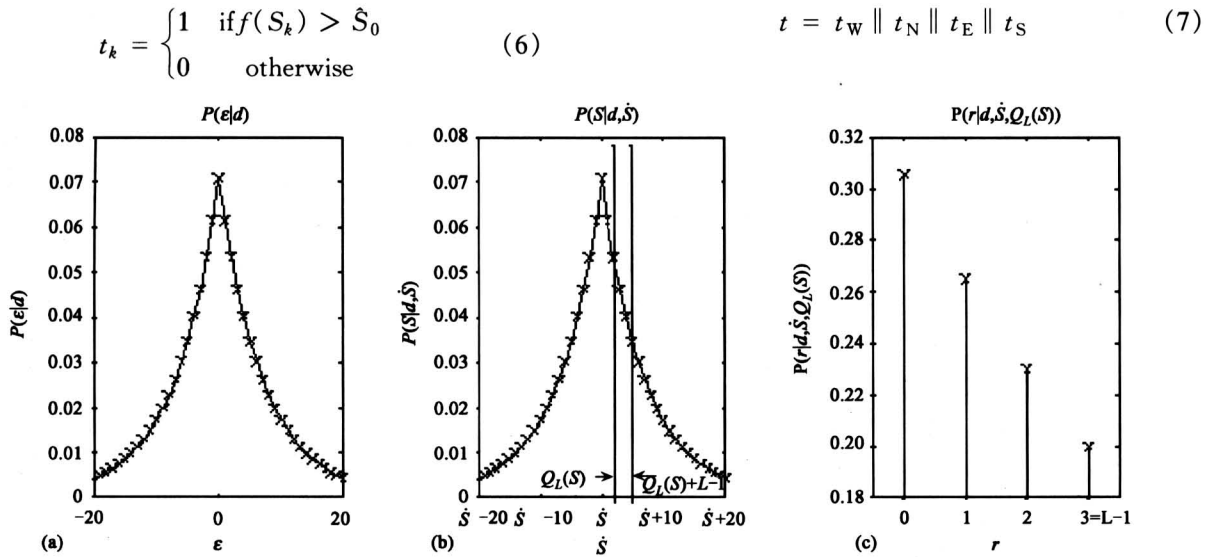


图 3 各种不同概率分布函数

Fig.3 Different PMF

(a) 在拉普拉斯假设 ($\sigma_d=10$) 下 (b) 相关像素的概率分布函数 (c) 增强层的条件概率分布函数的预测误差概率分布函数, $p(\epsilon|d)$; $p(\hat{S} + \epsilon/d, \hat{S})$; ($L=4$), $p(r|d, \hat{S}, Q_L(S))$

为了避免上下图稀释, t 上下图只用于预测中, 而不用在编码过程中。预测误差的分布函数一般用零均值不相关拉普拉斯概率分布函数表示。图 3 (a) 所示的就是预测误差的概率分布函数 (PMF), $p(\epsilon|d)$ 。给定 \hat{S} 后, 可以将预测误差分布 $p(\epsilon|d)$ 平移 \hat{S} , 得到像素的概率分布函数 $p(s = \hat{S} + \epsilon|d, \hat{S})$, 相应的 PMF 如图 3 (b) 所示。

为了得到增强层的概率分布并利用量化像素 $Q_L(s)$, 这里引入另一个上下图 θ , θ 只用于编码过程而不用用于预测过程。

这里, 已知量化值 $Q_L(S)$ 成为一个附加的上下图。量化像素值 $Q_L(S)$ 将像素 S 的可能的值限制在区间 $[Q_L(S), Q_L(S) + L]$ 内, 如图 3 (b) 中两条垂直界线所包围的区域所示。图 3 (c) 是由图 3 (b) $[Q_L(S), Q_L(S) + L]$ 之间的概率值标准化后得到的条件概率分布函数 $p(r|d, \hat{S}, Q_L(S))$ 。

由于拉普拉斯分布在 \hat{S} 的峰值呈指数下降, 所以条件概率函数 $P(r|d, \hat{S}, Q_L(S))$ 有四种可能的分布, 如图 4 所示: 如果 $\hat{S} \leq Q_L(S)$, 峰值在 $r = 0$ 处。如果 $\hat{S} \geq Q_L(S) + L - 1$, 峰值出现在 $r = L - 1$ 处。当 $Q_L(S) < \hat{S} < Q_L(S) + L - 1$, 峰值在 $r = \hat{S} - Q_L(S)$ 。这样, 上下图状态就减少到只有 4 个, 大大提高了编码效

率。下面我们进一步利用拉斯分布的对称性, 继续减少上下图状态的个数。

如图 4 所示, 峰值在 r_θ 和 $L - 1 - r_\theta$ 位置的分布图是镜像对称的, 因此可以将这些上下图状态合并起来, 这样一来, 上下图状态的数量减少了, 而且不会引起图像质量的恶化。接下来, 将每一对对称的分布图赋值为一对相反数 ($\pm \theta_i$), $|\theta|$ 就和 d 一起成为一个用于编码的上下图了。

第二步, 将 $\hat{S} = Q_L(S)$ 和 $\hat{S} = Q_L(S) + L - 1$ 情况单独赋值到一个 θ 中。虽然这些情况已经包含在 $\hat{S} \leq Q_L(S)$ 和 $\hat{S} \geq Q_L(S) + L - 1$, 但由于公式 (3) 中含入函数 $\text{round}()$ 的特性, 当 $\hat{S} \approx Q_L(s)$ 和 $\hat{S} \approx Q_L(S) + L - 1$ 时可能会导致预测出现偏差, 所以将这两种情况单独赋值给一个, 这样可以使编码更加准确。最后, 上下图 θ 和编码上下图 (d, θ) 的个数为 $\lfloor (L + 1) / 2 + 1 \rfloor$ 和 $8 \lfloor (L + 1) / 2 + 1 \rfloor$ 。

如果 ($\theta \geq 0$),

编码/解码增强层 (r_0, d, θ) ;

如果 ($\theta < 0$),

编码/解码增强层 $(L - 1 - r_0, d, |\theta|)$;

1.3 条件编码

为了提高效率, 使用了基于上下图的自适应算术编码器。该编码器基于上述上下图模型对增强层进行编码。像素的条件概率分布都是由原先已经编

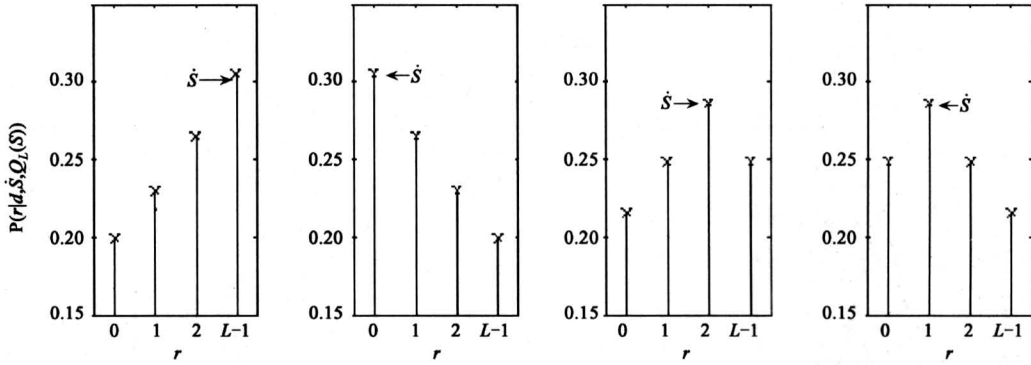


图 4 条件概率分布函数 $P(r|d, \hat{S}, Q_L(S))$, 上下图 $\theta = \{\pm 1, \pm 2\}$ ($L=4$)

Fig.4 Conditional PMFP

码(解码)好了的像素值预测出来, 将每一个像素的概率值用最佳编码长度的算术编码进行编码。

1.4 分解为多个嵌入位平面的方法

以上介绍的是仅将图像分解为一个嵌入位平面(一个基础层和一个增强层)的图像处理方法。多个嵌入位平面的分解可以通过对以上算法的直接递归应用而得到。第一步, 图象被 L_1 分为基础层 B_1 和增强层 r_1 。第二步, 基础层 B_1 被 L_2 分为基础层 B_2 和增强层 r_2 。依次类推……。每一个增强层 r_i 基于相关的基础层 B_i 进行压缩。最后一层 B_n 用 CALIC 算法进行压缩编码。

2 实验结果

图 5 是 4 幅 512×512 的 8 位灰度图像。为了说明问题, 这里取 $L=2$, 用 1.4 节的方法, 将图像分为基础层和增强层。增强层即嵌入位平面, 可以从一个到 7 个不等, 例如: 一个增强层就是将最低位作为增强层而其余 7 位属于基础层; 7 个增强层就是将最高为作为基础层, 而其余 7 位都是增强层, 在这种情况下, 所有的位平面都可以被重建, 因而可以产生完全可测量的码流。如前所述, 基础层使用的压缩算法是 CALIC 算法。

表 1 所描述的是基于嵌入式位平面的压缩算法的压缩性能和其他各种优秀的压缩算法的性能的比较。其中, CALIC, JPEG2000 和 JPEG-LS 算法不具有平面可测量性, JBIG, JBIG (gray) 和本文所提出的 LELIC 算法具有平面可测量性, 我们用 L.E.1, L.E.2, ..., L.E.7 分别表示 1 个增强层, 2 个增强层, ..., 7 个增强层。在无损压缩中, CALIC 的压缩率最大, 所以在表 1 中, 将

CALIC 的压缩率作为标准, 其他压缩算法的压缩率与 CALIC 相比, 计算它们所增长的比特率的百分比。从表中可以看到, JPEG2000 和 JPEG-LS 仅比 CALIC 增长了很小的比特率, 但是他们和 CALIC 一样, 不具备平面可测量性。JBIG 具有平面可测量性, 但是却比 CALIC 增长了平均 37.1% 的比特率。

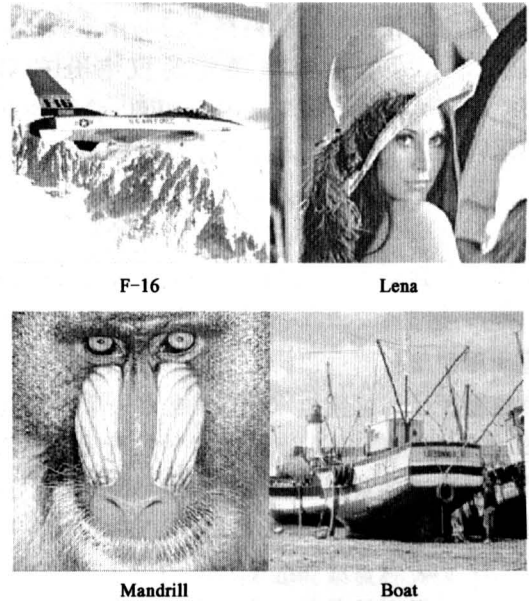


图 5 512×512 的 8 位灰度图像

Fig.5 Each image with 512×512 gray levels (8-bits)

使用 LELIC 算法对图像进行压缩时, 随着增强层数量的增多, 图像所产生的恶化程度基本是一个线性的增长。如果将一幅 8 位的图像截断成 6 位的图像, 则需要有 2 个嵌入式位平面(即 2 个增强层), 这样一来, 于 CALIC 算法相比, 平均会增

表 1 基于嵌入式位平面的压缩算法和其他经典无损压缩算法的性能比较

Table 1 Performance of LELIC and CALIC

		F-16	Mandrill	Boat	Lena	平均
用于比较的压缩算法		最佳无损压缩率				
	CALIC	3.54	5.66	4.15	4.08	4.36
用以下算法压缩后, 与 CALIC 算法相比, 比特率增加的百分比						
无嵌入 式平面	CALIC	0.0	0.0	0.0	0.0	0.0
	JPEG2000	7.6	4.0	6.2	5.2	5.8
	JPEG-LS	1.9	2.8	2.4	3.4	2.6
	JBIG	46.6	26.2	35.8	39.7	37.1
有嵌入 式平面	L.E.1	2.0	0.2	1.6	1.1	1.2
	L.E.2	4.1	0.9	4.1	3.7	3.2
	L.E.3	7.0	2.2	6.3	5.8	5.3
	L.E.4	10.6	3.4	9.9	8.6	5.6
	L.E.5	13.7	5.3	12.4	11.7	10.8
	L.E.6	15.9	6.6	14.5	14.4	12.9
	L.E.7	18.8	7.6	16.4	17.5	15.1

加 3.2% 的比特率。这一压缩率已经基本可以和 JPEG-LS 以及 CALIC 相媲美。如果将图像截断成只有一位, 即需要 7 个增强层, 这样平均会比 CALIC 算法增加 15.1% 的比特率, 即使这样的压

缩率也大大优于了 JBIG 算法。

3 结语

到目前为止, 并不存在绝对好的图像压缩方法: JPEG2000 具有非常丰富的功能, 但它的无损压缩比低于 JPEG-LS; JPEG-LS 和 CALIC 具有较高的无损压缩比, 但是不具备平面可测量性; 而笔者提出的 LELIC 算法, 虽然它的压缩比略微低于 JPEG-LS 和 CALIC, 但是由于它所具有的位平面可测量性, 使得在带宽不稳定的情况下仍然可以完整地访问图像这一愿望成为了可能。因此, 在网络带宽经常变化的不佳条件下, LELIC 算法是一种比较理想的压缩算法。

参考文献

- [1] 刘建华, 张昌年. JPEG2000 压缩编码标准介绍[J], 桂林航天工业高等专科学校学报, 2002; (1): 41~43
- [2] Wu X., Memon B. Context-based, adaptive, lossless image codec[J], IEEE Trans. Comm. 1997, 45 (4): 437~444
- [3] Wu X. ISO/IEC 10918-1 图片测试集的编准测试图片[EB/OL]. Ftp://ftp.csd.uwo.ca/pub/from-wu/image, 2002-5-20
- [4] 曹青, 吴乐南. 静止图像无失真编码的新标准 JPEG-LS[J]. 电子工程师, 1999, (2): 32~34

Research of Lossless Image Compression Base on Level-scalability

Li Luwei¹, Zhou Shuoyan², Cai Yiyu³

(1. School of Information, Zhejiang Univ. of Technology, Hangzhou 310032, China;

2. School of Electronic Technology and Engineering, Jilin Univ., Changchun 130021, China;

3. Zhejiang Province Linhai Power Supply Bureau, Linhai, Zhejiang 317000, China)

[Abstract] A level-embedded lossless image compression method for continuous-tone still images is presented. Level (bit-plane) scalability is achieved by separating the image into two layers (the base layer and the residual layer) before compression. Excellent compression performance is obtained by exploiting both spatial and inter-level correlations. A comparison of the proposed scheme with a number of scalable and non-scalable lossless image compression algorithms indicates that the level-embedded compression incurs only a small penalty in compression efficiency over non-scalable lossless compression, while offering the significant benefit of level-scalability.

[Key words] data processing techniques; lossless image compression; context-based model; embedded level