IPN Progress Report 42-159 November 15, 2004

Selecting the Golomb Parameter in Rice Coding

A. Kiely[1]

We examine the use of Golomb-power-of-2 (GPO2) codes to efficiently and losslessly encode sequences of nonnegative integers from a discrete source. Specifically, we're interested in the problem of selecting which GPO2 code to use for a source or a block of samples; this problem is at the heart of the well-known Rice entropy coding algorithm. We're particularly concerned with the case where the mean sample value is known or can be estimated.

We derive bounds on the optimum code parameter as a function of the mean sample value. These bounds establish that no more than three possible code choices can be optimum for a given mean sample value.

We derive a simple method to select the optimum GPO2 code for a geometrically distributed source given the mean value. We also devise a simpler code selection procedure that generalizes previously known methods. Both code selection methods can be implemented using only integer arithmetic and table look-ups, and require no divisions.

We show that, for any source with known mean, the GPO2 code parameter selected under this simple procedure is always within one of the optimum code parameter for the source, and that the added cost due to suboptimum parameter selection under this procedure is never more than 1/2 bit per sample and no more than about 13 percent inefficiency.

We investigate the use of both code selection procedures as lower complexity alternatives to Rice coding within the emerging Consultative Committee for Space Data Systems (CCSDS) image compression standard. For the images tested, both code selection methods produce negligible added rate compared to optimal code selection as in Rice coding.

I. Introduction

We'd like to simply, adaptively, and compactly perform lossless encoding of the output of a discrete source that produces nonnegative integer outputs. This problem arises in many data compression applications. We're specifically interested in selecting the optimum code choice from a particular subset of Golomb codes.

[1] Communications Architectures and Research Section.

The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

For positive integer m, the mth Golomb code [1] defines a reversible prefix-free mapping of nonnegative integers to variable length binary codewords. Golomb codes are optimum for geometrically distributed sources: when δ is a geometrically distributed random variable, the appropriately selected Golomb code minimizes expected codeword length over all possible lossless binary codes for δ [2].

We restrict our choices to codes for which $m = 2^k$ for some nonnegative integer k. As noted in [1], coding in this case becomes especially simple. The codeword for the integer δ consists of the unary representation of $\lfloor \delta/2^k \rfloor$ (that is, $\lfloor \delta/2^k \rfloor$ zeros followed by a one) concatenated with the k least significant bits of the binary representation of δ. Following the convention of [3], we refer to this special case as a Golomb-power-of-2 (GPO2) code with parameter k, and we denote it by $\mathcal{G}_k$. Such codes are used in the Rice entropy coding algorithm [4,5] and the LOCO-I image compressor [3], among myriad other applications.

Our problem of interest is to calculate or estimate the value of the GPO2 code parameter k that minimizes the expected bit rate (the average number of encoded bits per source symbol) for a source. This problem arises in Rice coding. The Rice algorithm encodes a block of samples using the best code option for the block from among several candidate codes that consist mostly of different GPO2 codes. A fixed number of bits are used preceding the encoded block to indicate which code was selected. The Rice algorithm does not specify how to find the best code option, and the most common approach is to exhaustively try each one. One of our interests is to save some computations in this process by making a fast, though sometimes suboptimal, code selection when the mean value of δ is known or can be estimated.

II. Analysis

A. Minimizing Code Rate

Using GPO2 code $\mathcal{G}_k$ to encode a nonnegative integer random variable δ requires $\lfloor \delta/2^k \rfloor + 1$ bits for the unary part and k bits for the remainder. Thus, the code rate (the expected number of coded bits per sample) is

$$R_k = \sum_{j=0}^{\infty} \left(\left\lfloor \frac{j}{2^k} \right\rfloor + 1 + k \right) \text{Prob}[\delta = j] \quad \text{(bits/sample)}$$

It can be shown that, for any fixed nonnegative integer j, the quantity $\lfloor j/2^k \rfloor + 1 + k$ is convex $\cup$ in k. Since the above equation expresses R_k as a convex combination of terms of this form, R_k must also be convex $\cup$ in k. Consequently, a local minimum of R_k is also a global minimum, and so any nonnegative integer k^* satisfying

$$R_{k^*} \leq R_{k^*+1} \quad \text{and} \quad R_{k^*} \leq R_{k^*-1} \tag{1}$$

gives an optimum choice for the code parameter (we treat R_{-1} as ∞ for this calculation). This fact could be used to save some calculations used in the exhaustive approach to selecting the coding option in the Rice algorithm, since once we have found an integer k^* satisfying Condition (1), no further evaluation of R_k is needed.

In the following, we consider the problem of selecting the code parameter when we have an estimate of the mean value $\mu = E[\delta]$. Such an estimate might be determined from previously encoded samples, or in the case of a block-adaptive algorithm like Rice coding, we might explicitly compute the sample mean for each block of samples.

Given μ, the optimum GPO2 code parameter must be in the range $[k^*_{\min}(\mu), k^*_{\max}(\mu)]$, where

$$k^*_{\min}(\mu) = \max\left\{0, \left\lfloor \log_2\left(\frac{2}{3}(\mu+1)\right)\right\rfloor\right\} \qquad k^*_{\max}(\mu) = \max\left\{0, \lceil \log_2 \mu \rceil\right\} \tag{2}$$

Figure 1 shows these bounds on the optimum code parameter value. For all $\mu < 1$, $k^*_{\max}(\mu) = k^*_{\min}(\mu) = 0$, and hence the optimum parameter choice is $k = 0$ over this region. For other values of μ, the number of possible optimum GPO2 parameter choices is limited to no more than three (this fact might save calculations in some coding implementations); that is,

$$k^*_{\max}(\mu) - k^*_{\min}(\mu) \le 2 \quad \text{for all} \quad \mu$$

The expressions for $k^*_{\max}(\mu)$ and $k^*_{\min}(\mu)$ follow from bounds on the difference in rate achieved at two consecutive values of k:

$$\frac{1}{2} - \frac{\mu}{2^k} \le R_k - R_{k-1} \le \frac{3}{2} - \frac{\mu+1}{2^k} \tag{3}$$

The bounds in Eq. (3) are tight whenever $k \le k^*_{\max}(\mu)$. These results are derived in Appendix A.

We can express the code rate in a way that emphasizes a dependence on μ:

$$\begin{aligned} R_k &= k + 1 + E\left[\left\lfloor \frac{\delta}{2^k} \right\rfloor\right] \\ &= k + 1 + \frac{1}{2^k}\left(E[\delta] - E\left[\delta - 2^k \left\lfloor \frac{\delta}{2^k}\right\rfloor\right]\right) \\ &= k + 1 + \frac{1}{2^k}\left(\mu - \overline{r}_k\right) \end{aligned} \tag{4}$$

Here we define $\overline{r}_k$ as the expected value of

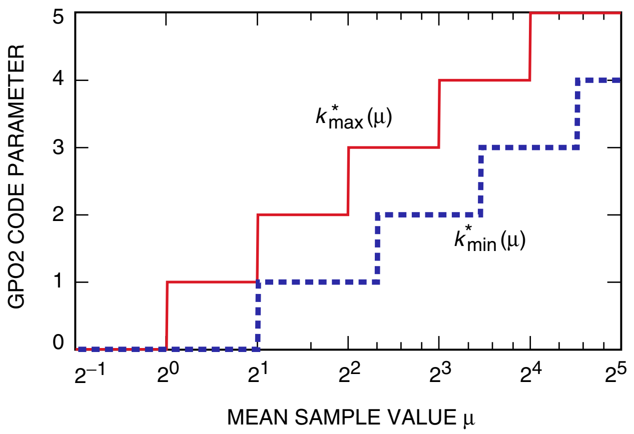

Fig. 1. Upper and lower bounds on optimum GPO2 code parameter choice as a function of sample mean μ.

$$r_k = \delta - 2^k \left\lfloor \frac{\delta}{2^k} \right\rfloor \tag{5}$$

which is equal to the remainder obtained when dividing δ by 2^k.

Equation (4) can be used to derive the following tight upper and lower bounds on R_k as a function of μ:

$$R_k \leq R_k^{\mathrm{up}}(\mu) \triangleq k + 1 + \frac{\mu}{2^k}$$

$$R_k \geq R_k^{\mathrm{low}}(\mu) \triangleq k + \max\left\{1, \frac{\mu+1}{2^k}\right\}$$

The upper bound above follows from Eq. (4) and the fact that $\overline{r}_k \geq 0$. Equality is achieved if all sample values that are not multiples of 2^k occur with probability zero. For any μ, there always exists a distribution with mean μ satisfying this property; hence, this bound is tight. To establish the lower bound on rate, we first note that, since $\overline{r}_k \leq 2^k - 1$, Eq. (4) implies that $R_k \geq k + (\mu+1)/2^k$. This is satisfied with equality only if sample value j occurs with probability zero whenever $j - (2^k - 1)$ is not a multiple of 2^k. This isn't possible when $\mu < 2^k - 1$, when the rate can equal $k+1$. Taking this fact into account produces the lower bound above.

To identify the optimum code parameter using Condition (1), we note from Eq. (4) that $R_k \leq R_{k-1}$ if and only if

$$\mu \geq 2^k + 2\overline{r}_{k-1} - \overline{r}_k \tag{6}$$

Since $\overline{r}_0 = 0$, we would select the code parameter to be zero when $\mu < 2 - \overline{r}_1$; otherwise we would select it to be the largest positive integer k satisfying Eq. (6). We'd like to estimate the function $\overline{r}_k$ so that we can apply Eq. (6) to select the code parameter given an estimate of μ.

In a practical application, this code selection approach can be implemented using a table to store values of μ, where we switch from one code parameter to the next. In Section III we follow this approach using a geometric probability model for δ, and in Section IV we consider a simpler alternative method of selecting the code parameter.

B. Dynamic Range Limit and Uncoded Data

Two practical considerations affect some applications where families of GPO2 codes are used. First, although Golomb codes allow arbitrarily large sample values, in a practical application samples are constrained by some maximum value, and this limits the range of coding options that need to be considered. Second, many entropy coders allow an option for samples to be transmitted uncoded (as is commonly done in Rice coding), and in this case the parameter selection procedure should include a rule for selecting this option.

When sample values are constrained to some maximum number of bits N, sending each sample uncoded (or more precisely, encoded using the conventional unsigned N-bit binary integer representation) costs N bits per sample. The code rate obtained using code $\mathcal{G}_k$ for any $k \geq N-1$ must be at least as large, and so whenever we have the option of sending data uncoded, we would want to impose the constraint that

$$k \le N - 2$$

This constraint isn't completely vacuous. Under either of the parameter selection approaches described in Sections III and IV, an estimate of μ sufficiently close to the maximum sample value can result in selecting GPO2 parameter $k = N - 1$ if we fail to consider the dynamic range limit and the option to send samples uncoded.

III. Coding a Geometric Source

A. Optimum Code Selection

Since Golomb codes optimally encode sources having geometric probability distributions, the use of GPO2 codes in an entropy coder would seem to imply that the source being coded is approximately geometric, and many practical sources fit this description. Consequently, to evaluate $\overline{r}_k$ in Eqs. (4) and (6), we model δ using a geometric probability distribution with parameter $\alpha \in (0,1)$, i.e.,

$$\mathrm{Prob}[\delta = j] = (1-\alpha)\alpha^j$$

Under this model,

$$\overline{r}_k = E\left[\delta - 2^k \left\lfloor \frac{\delta}{2^k} \right\rfloor\right] = \sum_{j=0}^{\infty} \left(j - 2^k \left\lfloor \frac{j}{2^k} \right\rfloor\right)(1-\alpha)\alpha^j$$

$$= (1-\alpha)\sum_{\ell=0}^{\infty} \sum_{j=\ell 2^k}^{(\ell+1)2^k - 1} \left(j - 2^k \ell\right)\alpha^j$$

$$= (1-\alpha)\sum_{\ell=0}^{\infty} \sum_{t=0}^{2^k-1} t\alpha^{t+\ell 2^k}$$

$$= \mu + 2^k\left(1 - \frac{1}{1-\alpha^{2^k}}\right)$$

Here we've made use of the fact that $\mu = \alpha/(1-\alpha)$ for a geometric distribution. Substituting this expression for $\overline{r}_k$ into Eq. (4), we find that, for a geometric source,

$$R_k = k + \frac{1}{1-\alpha^{2^k}} \tag{7}$$

Considering Condition (1) to determine the optimum code parameter, we find that $R_k \le R_{k-1}$ for a geometric source if and only if

$$\alpha^{2^k} + \alpha^{2^{k-1}} - 1 \ge 0$$

This inequality is a quadratic in $\alpha^{2^{k-1}}$ and is satisfied if and only if $\alpha^{2^{k-1}} \ge (\sqrt{5}-1)/2$, or equivalently,

$$k \leq 1 + \log_2 \left(\frac{\log(\phi - 1)}{\log\left(\frac{\mu}{\mu+1}\right)} \right)$$

where the constant ϕ denotes the golden ratio $(\sqrt{5}+1)/2$. Thus, if $\mu < \phi$, we would select $k = 0$; otherwise we would select the code parameter to be the largest positive integer k satisfying the above inequality. Thus, the optimum GPO2 code parameter for a geometric source with mean μ is

$$k^*_{\text{geo}}(\mu) = \max\left\{0,\ 1 + \left\lfloor \log_2 \left(\frac{\log(\phi - 1)}{\log\left(\frac{\mu}{\mu+1}\right)} \right) \right\rfloor \right\} \tag{8}$$

A practical way to implement this parameter selection rule is to use a table that stores the values of μ, where we switch from one parameter choice to the next. We define $\mu^*_0 = 0$ and for $k > 0$ define

$$\mu^*_k = \frac{1}{\phi^{2^{1-k}} - 1}$$

which is the smallest value of μ for which parameter k is at least as good as $k-1$. A necessary and sufficient condition for an integer k to be an optimum GPO2 code parameter value is that

$$\mu^*_k < \mu \leq \mu^*_{k+1} \tag{9}$$

and so we can determine the code parameter by comparing an estimate of μ with quantities in the table of μ^*_k values.

Alternatively, instead of using a table of floating point values $\mu^*_0, \mu^*_1, \cdots$, we can eliminate the use of floating point numbers by using a rational approximation to each μ^*_k. We can do this by producing a table of integers $Q_M(0), Q_M(1), \cdots$, where

$$Q_M(k) \triangleq \left\lfloor M \cdot \mu^*_k + \frac{1}{2} \right\rfloor$$

for some fixed integer M, so that $\mu^*_k \approx (1/M)Q_M(k)$. If we're selecting the GPO2 code for a block of J samples $\delta_0, \delta_1, \cdots, \delta_{J-1}$, as is done in Rice coding, we could compute the sum of the sample values in the block

$$S = \sum_{i=0}^{J-1} \delta_i$$

and use the sample mean S/J as our estimate for μ. In this case, the parameter selection rule (9) can be implemented by selecting the code parameter to be the unique integer k such that

$$J \cdot Q_M(k) < M \cdot S \leq J \cdot Q_M(k+1)$$

This approach eliminates the need for divisions or floating point arithmetic. If the block size J doesn't change from block to block, then we can eliminate the multiplications above by selecting $M = J$. Otherwise, selecting M to be a power of two allows multiplications by M to be computed using bit-shift operations.

In Section V, we give an example of the performance of this strategy when used as part of an image compression application.

B. Sending Uncoded Data

As discussed in Section II.B, a couple of modifications to the parameter selection strategy are required when samples are constrained to N bits and we have the option of sending uncoded data. First, we impose the constraint on our parameter selection procedure that $k \leq N-2$, which limits the size of the table needed for parameter selection.

Next, we need to determine a rule for selecting uncoded data. Under the N-bit dynamic range limit, the use of uncoded data beats the GPO2 code with parameter $k = N-2$ whenever $N < R_{N-2}$. Using the code rate for a geometric source given in Eq. (7), this condition turns out to be equivalent to

$$\mu > \mu_N^{\dagger} \triangleq \frac{1}{2^{2^{2-N}} - 1}$$

It can be shown that $\mu > \mu_N^{\dagger}$ implies that $R_{N-3} > R_{N-2}$ for a geometric source, and since R_k is convex $\cup$ in k, we conclude that the above condition is a sufficient test to determine when uncoded data should be used.

Thus, the modified strategy is to first compare μ to $\mu_N^{\dagger}$ to see whether sample values should be transmitted uncoded.[2] If not, then we use the parameter selection strategy of Section III.A with the constraint that $k \leq N-2$.

C. Relative Redundancy

When using code $\mathcal{G}_k$ to encode the output of a geometrically distributed source, the relative redundancy is

$$\frac{R_k - \mathcal{H}}{\mathcal{H}}$$

where the code rate R_k is given in Eq. (7), and $\mathcal{H}$ denotes the entropy of a geometric source:

$$\mathcal{H} = \frac{1}{1-\alpha}\left[-(1-\alpha)\log_2(1-\alpha) - \alpha\log_2\alpha\right] \quad \text{(bits)}$$

$$= (1+\mu)\log_2(1+\mu) - \mu\log_2\mu \quad \text{(bits)}$$

Figure 2 shows relative redundancy versus μ for a geometric source when encoded using the optimum GPO2 code selected according to Eq.(8). The peaks in the curve correspond to values of μ where we switch from one code to the next.

[2] It is straightforward to construct an approximate equivalent test that requires only integer arithmetic, and we omit the details.

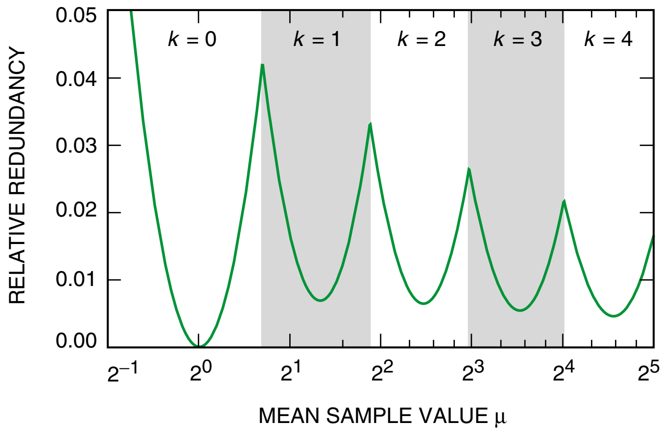

Fig. 2. Relative redundancy for a geometric source with mean μ when coded using the optimum GPO2 code.

Since the source entropy approaches zero as μ approaches zero, the relative redundancy is unbounded, as is the case for any coding scheme that assigns a distinct codeword to each output value. When $\mu < 1$, approaches such as runlength coding give better performance (see, for example, [6,7]). When $\mu \geq 1$, the relative redundancy is largest when the optimum GPO2 code switches from $\mathcal{G}_0$ to $\mathcal{G}_1$, which occurs when $\mu = \phi$, in which case the relative redundancy is

$$\frac{1}{(3-\phi)\log_2 \phi} - 1 \approx 0.042$$

or about 4.2 percent.

IV. A Simple Code Parameter Selection Rule

A. A Simple Rate Approximation

The quantity r_k, defined in Eq. (5), is nonnegative and has a maximum value of $2^k - 1$. As a simple approximation, we might estimate $\overline{r}_k$ to be some constant fraction $f \in [0,1]$ of this maximum value, i.e., approximate $\overline{r}_k$ as $(2^k - 1)f$. This approximation isn't necessarily very accurate for sources encountered in practice, but it leads to a simple parameter estimation rule that turns out to give good performance.

Under this approximation, Condition (6) reduces to $\mu \geq 2^k - f$, and thus we use $k = 0$ if $\mu + f < 1$; otherwise we select the code parameter to be the unique nonnegative integer k satisfying

$$2^k \leq \mu + f < 2^{k+1}$$

i.e., select

$$\hat{k}_f(\mu) = \max\left\{0, \lfloor \log_2(\mu + f) \rfloor\right\} \tag{10}$$

We'll refer to this method as the simple code selection approach using parameter f. As an indication that the simple approach is reasonable, we note that it always yields a value within the range of possibly optimum parameters identified in Eq. (2):

$$k^*_{\min}(\mu) \le \hat{k}_f(\mu) \le k^*_{\max}(\mu) \quad \text{for all} \quad f \in [0,1] \tag{11}$$

Parameter estimation rules with this same general form have been developed previously by other researchers. The case where $f = 1/2$ was suggested by Pen-Shu Yeh,[3] and a similar parameter selection rule (for a slightly different coding problem, effectively with $f = 0$) is used as part of the LOCO-I algorithm [3]. As noted in [3], the parameter value can be computed using a simple C source code fragment:

```
for (k = 0; (1<<k)<=μ+f;k++)
    ;
```

The simple parameter selection approach is never far from optimum. For any discrete source producing nonnegative integer outputs with mean μ, the optimum GPO2 code parameter never differs from the code parameter value selected under the simple approach by more than one, for any $f \in [0,1]$. This follows from Eq. (11) and from the fact that

$$k^*_{\max}(\mu) - 1 \le \hat{k}_f(\mu) \le k^*_{\min}(\mu) + 1 \quad \text{for all} \quad f \in [0,1] \tag{12}$$

which is proved in Appendix B.

In an application where we have the option of transmitting samples uncoded, we can use the rule for selecting uncoded data described in Section III.B.

B. Performance on a Geometric Source

As an example illustrating the performance of the simple parameter selection rule, Fig. 3 shows the relative redundancy for a geometric source when we choose $f = 1/3$. The relative redundancy is higher than the optimum shown in Fig. 2 whenever the selected parameter differs from the optimum given in Eq. (8). A consequence of Eqs. (11) and (12) is that, for any $f \in [0,1]$, $\hat{k}_f(\mu)$ and $k^*_{\text{geo}}(\mu)$ never differ by more than one:

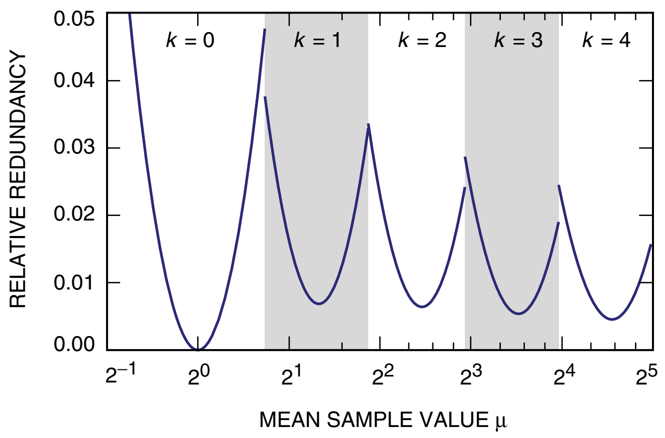

Fig. 3. Relative redundancy for a geometric source with mean μ when coded using the GPO2 code selected using the simple approach with $f = 1/3$.

[3] P.-S. Yeh, personal communication, May 2004.

$$k^*_{\text{geo}}(\mu) - 1 \leq \hat{k}_f(\mu) \leq k^*_{\text{geo}}(\mu) + 1 \quad \text{for all} \quad f \in [0, 1]$$

The simple code selection rule of Section IV.A depends on an unspecified constant f, and so the obvious question is: how should we select the value of f? As we've argued in Section III, it seems reasonable to assume a geometric source model in a coding application that relies on Golomb codes. Since $\hat{k}_f(\mu) = k^*_{\text{geo}}(\mu) = 0$ for all $\mu < 1$, a reasonable approach would be to select f to minimize the maximum relative redundancy obtained for a geometric source with mean $\mu \geq 1$.[4]

When $\mu \geq 1$, the maximum relative redundancy occurs at the point where we switch from code $\mathcal{G}_0$ to $\mathcal{G}_1$, which, for the simple strategy, occurs at $\mu = 2 - f$. Thus, for a geometric source we minimize the maximum additional relative redundancy due to suboptimum parameter selection by selecting f so that the heuristic strategy matches the optimum strategy at this point. This is achieved by setting $f = f^*$, where

$$f^* = 2 - \mu^*_1 = 2 - \phi \approx 0.382$$

Figure 4 shows the relative redundancy under this choice of f. For a geometric source, this choice gives the same maximum relative redundancy (about 4.2 percent) as obtained under the optimum parameter selection strategy when $\mu \geq 1$. As we'll see in Section V, in a typical Rice coding application this inefficiency due to suboptimum parameter selection may be quite small compared to the added rate due to overhead used to indicate the code selected.

If our estimate of μ is computed as the ratio of a sum of sample values (an integer S) divided by a number of samples J, and we approximate f^* as a ratio of integers A and B, then our strategy is to select code parameter 0 when $1 > \mu + f^* \approx S/J + A/B$; otherwise select the code parameter to be the largest nonnegative integer k satisfying

$$2^k \leq \mu + f^* \approx \frac{S}{J} + \frac{A}{B}$$

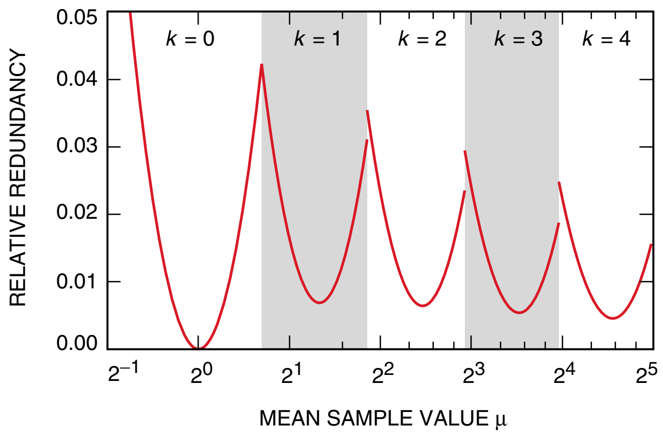

Fig. 4. Relative redundancy for a geometric source with mean μ when coded using the GPO2 code selected using the simple approach with $f = f^*$.

[4] Alternatively, we note that in some implementations setting $f = 0$ reduces the number of calculations required for parameter selection.

or, equivalently,

$$J \cdot B \cdot 2^k \leq B \cdot S + J \cdot A$$

Note that here the parameter selection procedure requires no divisions or floating point operations. In the C programming language, this can be implemented as

```
for (k = 0; ((J * B) << k) <= B * S + J * A; k + +)
    ;
```

A good rational approximation to f^* is 49/128. When the source is geometric with known mean, this approximation results in a negligible increase in relative redundancy over using the exact value of f^*. Other good rational approximations to f^* include 34/89 and 3/8.

As an indication of the practicality of this approach, we consider the size of the arithmetic registers required to implement the simple code selection approach when used as part of the emerging Consultative Committee for Space Data Systems (CCSDS) image compression standard [8]. In this application, the blocklength J is at most 16 samples per block, each sample value δ can be as large as $2^{10}-1$ (i.e., $N \leq 10$), and the GPO2 code parameter k is no larger than 8. In this application, if we set $A = 49$, $B = 128$ (i.e, use 49/128 as our approximation to f^*), then the above multiplications by B can be computed using bit-shift operations. The quantity `B*S+J*A` can be stored using a 21-bit unsigned integer, and `(J*B)<<k` can be stored using a 19-bit unsigned integer.

C. Bounds on Inefficiency

Under the Rice coding algorithm, the optimum GPO2 code is determined for a block of samples; this is typically done by exhaustively evaluating the code rate under each available option. By contrast, the simple code selection approach allows us to select a code based only on the mean (or sum) of samples in the block. Since this approach does not always select the optimum code, we'd like to bound the maximum inefficiency due to suboptimum code selection. In this section, we derive bounds that apply to any probability distribution on the nonnegative integers with known mean μ.

For given mean μ, we'd like to determine the maximum possible performance cost (i.e., increased rate) due to suboptimum code selection when we use the simple rule of Section IV.A—i.e., we'd like to calculate

$$\rho_f(\mu) = \max\left(R_{\hat{k}_f(\mu)} - \min_k R_k\right)$$

where the maximum is taken over all probability distributions on nonnegative integers with mean value μ.

Since $\hat{k}_f(\mu)$ is always within one of the optimum parameter choice, we can bound $\rho_f(\mu)$ via bounds on $R_k - R_{k-1}$ and $R_k - R_{k+1}$ given in Eq. (3). Specifically, we define

$$\Delta_f^-(\mu) \triangleq \begin{cases} \dfrac{3}{2} - \dfrac{\mu+1}{2^{\hat{k}_f(\mu)}}, & \text{if } \hat{k}_f(\mu) > 0 \\ 0, & \text{otherwise} \end{cases}$$

$$\Delta_f^+(\mu) \triangleq \frac{\mu}{2^{\hat{k}_f(\mu)+1}} - \frac{1}{2}$$

then

$$\rho_f(\mu) \le \rho_f^+(\mu) \triangleq \max\left\{0, \Delta_f^-(\mu), \Delta_f^+(\mu)\right\}$$

This bounds the worst-case rate increase arising from using the simple parameter selection rule rather than selecting the optimum code, as in true Rice coding. Figure 5 shows this bound on added rate for $f = 0$ and $f = f^*$. We note that $\rho_f^+(\mu) \le 1/2$; thus, when μ is known, the cost of using this suboptimum code selection procedure is never more than 1/2 bit per sample, for any choice of f. In Section V, we'll see that on a practical source it may be extremely small.

We can also bound the maximum relative inefficiency due to suboptimum code selection when we use the simple rule of Section IV.A. We define

$$\Phi_f(\mu) \triangleq \max \max_k \frac{R_{\hat{k}_f(\mu)} - R_k}{R_k} \le \max_{k \in [k^*_{\min}(\mu), k^*_{\max}(\mu)]} \frac{\rho_f^+(\mu)}{R_k^{\text{low}}(\mu)} \tag{13}$$

where the first maximum is taken over all probability distributions on nonnegative integers with mean value μ. Note that $\Phi_f(\mu)$ is the maximum relative inefficiency compared to coding using the optimally selected GPO2 code, and is not the same as the relative redundancy, which measures the relative inefficiency compared to the source entropy. The bound in Eq. (13) is probably not very tight.

In Fig. 6, we plot the bound of Eq. (13) as a function of μ when $f = 0$ and $f = f^*$. When $f = 0$, the bound has maximum value of $1/5$ at $\mu = 2$, and when $f = f^*$, the bound reaches maximum value of $(4\phi - 5)/11 \approx 0.13$ at $\mu = \phi$. Thus, when we choose $f = f^*$, the maximum inefficiency due to suboptimum parameter selection is no more than about 13 percent.

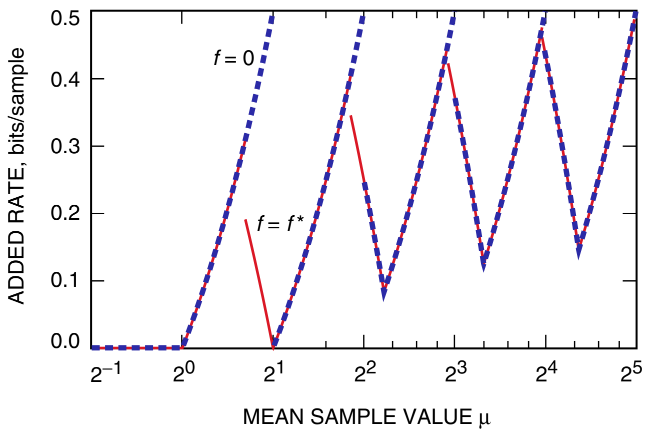

Fig. 5. Plot of $\rho_f^+(\mu)$, an upper bound on $\rho_f(\mu)$, the maximum added rate due to suboptimum GPO2 code selection as a function of source mean μ, when $f = 0$ (dashed curve) and $f = f^*$ (solid curve).

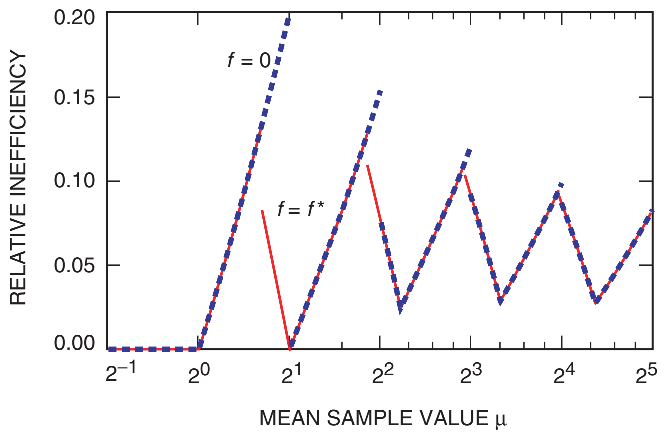

Fig. 6. Bound on $\Phi_f(\mu)$ (maximum relative inefficiency due to suboptimum GPO2 code selection) as a function of source mean μ, when $f = 0$ (dashed curve) and $f = f^*$ (solid curve).

V. Empirical Results

As a practical test, we compare the performance of the parameter selection techniques of Sections III and IV with optimum parameter selection (Rice coding) when applied to block coding of test image data as part of the current draft CCSDS image compression standard [8].

The draft CCSDS standard calls for three stages of two-dimensional discrete wavelet transform (DWT) decomposition of the input image. For our experiment, we use the 9/7-M integer DWT using the implementation described in [9]. In the initial coding stage of the CCSDS compressor, DWT coefficients in the lowest spatial frequency subband are uniformly quantized, and differences between successive quantized coefficients are mapped to nonnegative integers using the procedure described in Appendix C. Each block (of typically 16) of these nonnegative integer values is then encoded using a GPO2 code or transmitted uncoded.

For our test data, we use the "forest_2kb4," "india_2kb4," and "ice_2kb1" images from a set of test images used in the development of the CCSDS standard. Each of these images is taken from a single band of advanced very high resolution radiometer (AVHRR) instrument data. Each image contains 2048×2048 pixels at a bit depth of 10 bits/pixel. Combining data from the lowest spatial frequency subband from each image provides a total of 3×2^{16} samples for our test. Coding (and code selection) is performed on blocks of 16 consecutive samples. In our test, we vary the dynamic range of the input data (i.e., bit depth N) by varying the uniform quantizer applied to the DWT coefficients.

In this application, we evaluated the performance of the code selection method of Section III tuned to a geometric source, and the simple method of Section IV setting the parameter f equal to 0, 1/2, and 49/128 (this last value is chosen to be a rational approximation to f^* as described in Section IV.B). We measured the average difference in code rate between the given code selection method and the optimum code choice, which was determined by exhaustively trying each code option for every block (as is typically done in Rice coding). Figure 7 plots the added rate of each method due to suboptimum code selection as we vary the dynamic range of the input (the bit depth N), and Fig. 8 shows the relative inefficiency due to suboptimum code selection.

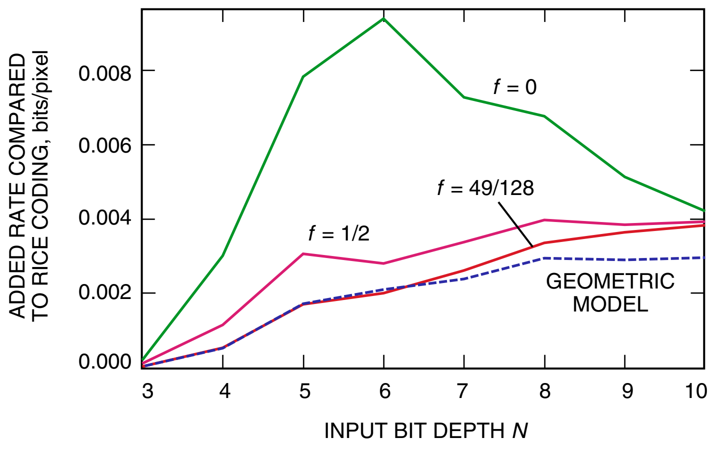

Fig. 7. Added rate due to suboptimum GPO2 code selection on the test data.

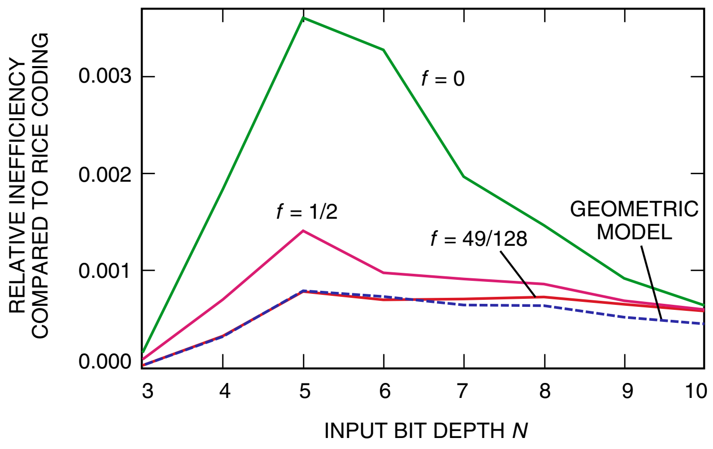

Fig. 8. Relative inefficiency due to suboptimum GPO2 code selection on the test data.

Figures 7 and 8 show that, under the parameter selection method of Section IV.B, setting f equal to 49/128 gives the best performance of the choices of f that we tested. The parameter selection method of Section III, which is optimized for a geometric source, gives even better performance when N is sufficiently large. Note that, in all cases tested, the added redundancy is extremely small compared to the bounds of Section IV.C, and the relative inefficiency is less than 0.4 percent in all cases.

For this data set, the cost of using a heuristic parameter selection strategy rather than optimum Rice coding is negligible compared to the overhead of using a block-adaptive approach in the first place. Under the method proposed in the CCSDS standard, the cost of overhead bits used to specify the code option selected exceeds 0.05 bit per sample, which is much higher than the average added redundancy of any of the parameter selection strategies considered.

On the basis of these results, the simple code selection approach, setting f equal to 49/128 currently is being considered as a low-complexity option as part of the CCSDS standard.

VI. Conclusion

The Rice algorithm's brute-force coding approach is perhaps the ultimate in *conceptual* simplicity among adaptive entropy coding techniques. Each block of samples is independently coded, so there are no statistics to track from one block to the next. No parameter estimation is performed. One simply tries several code options and picks the best one for the block.

However, the conceptual simplicity of the Rice coding paradigm brings with it some inherent limitations in terms of implementation complexity and compression effectiveness.

In practice, the complexity of Rice coding implementations is generally high because they tend to rely on separate evaluation of each code option for every block of samples. In Section II we've shown that when the sample mean for a block is known, no more than three GPO2 code options need to be considered. We've also provided evidence that the use of simpler suboptimum code selection methods results in little degradation in performance; however, such a reduced-complexity encoder is not, strictly speaking, a Rice coder, since the simpler code selection methods do not always pick the optimum code option.

By selecting the code option for each block independently, the Rice algorithm fails to exploit useful information from previously coded blocks. This results in an inherent cost in added bit rate due to the overhead bits needed with each block to identify the code selected. Using larger block sizes decreases this bit-rate cost somewhat, but also reduces the ability to quickly adapt to changes in the source data.

The effectiveness of an adaptive entropy coder generally increases when it has more code options available. For example, the relative redundancy shown in Fig. 2 for a geometric source could be made lower for some values of μ if additional Golomb codes (not just GPO2 codes) could be used. However, the extent to which one can add coding options to improve the coding effectiveness of a Rice coder is somewhat limited. Adding code options would increase the complexity of many entropy coders, but the complexity increase becomes particularly significant in a Rice coding implementation where each code option is tried explicitly on each block of samples. Moreover, the Rice coding paradigm imposes a practical limit on the number of code options, because increasing the number of code options increases the overhead required to indicate which code was selected for each block.

A sensible alternative to Rice coding is to adaptively select the code option based on estimates of the mean value (or some other parameter) from recently encoded samples. This simple approach largely avoids the deficiencies of Rice coding as it doesn't require the evaluation of multiple coding options, eliminates the need for overhead bits, and makes it more practical to incorporate additional code options. The success of such an approach has been demonstrated in [3,6] among others.

In light of these considerations, for most applications it seems doubtful that the Rice coding approach of optimizing the code selection for each block of data warrants the added complexity or the significant overhead cost required to indicate the code option selected.

Acknowledgment

The author would like to thank Pen-Shu Yeh at NASA Goddard Space Flight Center, whose previous work on the GPO2 code parameter selection problem inspired this article, and who has provided helpful feedback on this problem.

References

[1] S. W. Golomb, "Run-Length Encodings," *IEEE Transactions on Information Theory*, vol. IT-12, no. 3, pp. 399–401, July 1966.

[2] R. G. Gallager and D. C. Van Voorhis, "Optimal Source Codes for Geometrically Distributed Integer Alphabets," *IEEE Transactions on Information Theory*, vol. IT-21, no. 2, pp. 228–230, March 1975.

[3] M. J. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I Lossless Image Compression Algorithm: Principles and Standardization into JPEG-LS," *IEEE Transactions on Image Processing*, vol. 9, no. 8, pp. 1309–1324, August 2000.

[4] R. F. Rice, "Some Practical Universal Noiseless Coding Techniques," JPL Publication 79-22, Jet Propulsion Laboratory, Pasadena, California, March 15, 1979.

[5] Consultative Committee for Space Data Systems, *CCSDS 121.0-B-1: Lossless Data Compression*, Blue Book, issue 1, May 1997.
http://www.ccsds.org/CCSDS/documents/121x0b1.pdf

[6] A. Kiely and M. Klimesh, "Generalized Golomb Codes and Adaptive Coding of Wavelet-Transformed Image Subbands," *The Interplanetary Network Progress Report 42-154, April–June 2003*, Jet Propulsion Laboratory, Pasadena, California, pp. 1–14, August 15, 2003.
http://ipnpr.jpl.nasa.gov/tmo/progress_report/42-154/154F.pdf

[7] K.-M. Cheung and A. Kiely, "An Efficient Variable Length Coding Scheme for an IID Source," *Proceedings 1995 IEEE Data Compression Conference*, Snowbird, Utah, pp. 182–191, March 28–30, 1995.

[8] P. S. Yeh, P. Armbruster, A. Kiely, B. Masschelein, G. Moury, and C. Schaefer, "The New CCSDS Image Compression Recommendation," *Proceedings IEEE Aerospace Conference 2005*, Big Sky, Montana, March 5–12, 2005 (to appear).

[9] M. D. Adams and F. Kossentini, "Reversible Integer-to-Integer Wavelet Transforms for Image Compression: Performance Evaluation and Analysis," *IEEE Transactions on Image Processing*, vol. 9, no. 7, pp. 1010–1024, June 2000.

Appendix A
Bounds on Rate Differences

Section II.A gives bounds on the difference between rates achieved with consecutive GPO2 codes, along with constraints on the optimum code parameter value that arise from these bounds. In this appendix, we derive these results.

First we derive the bounds of Eq. (3). For $k > 0$, we have

$$R_k - R_{k-1} = 1 + \sum_j \left(\left\lfloor \frac{j}{2^k} \right\rfloor - \left\lfloor \frac{j}{2^{k-1}} \right\rfloor \right) \mathrm{Prob}[\delta = j] = 1 + \sum_j \left\lfloor \frac{1}{2} - \frac{j+1}{2^k} \right\rfloor \mathrm{Prob}[\delta = j]$$

The upper and lower bounds on this difference, given in Eq. (3), follow immediately from this expression combined with the following straightforward tight upper and lower bounds on the term in the above sum:

$$-\frac{1}{2} - \frac{j}{2^k} \le \left\lfloor \frac{1}{2} - \frac{j+1}{2^k} \right\rfloor \le \frac{1}{2} - \frac{j+1}{2^k} \quad \text{for any integer } j \tag{A-1}$$

The upper bound in Eq. (A-1) is met if and only if $j - 2^{k-1} + 1$ is a multiple of 2^k. As a result, the upper bound in Eq. (3) holds with equality if and only if $j - 2^{k-1} + 1$ is a multiple of 2^k for each j that occurs with nonzero probability. When $\mu < 2^{k-1} - 1$, this isn't possible and the upper bound in Eq. (3) is weak; otherwise the upper bound is tight. Applying the definition of $k^*_{\mathrm{max}}(\mu)$ from Eq. (2), we see that $k \le k^*_{\mathrm{max}}(\mu)$ is a sufficient condition for the upper bound in Eq. (3) to be tight.

The lower bound in Eq. (A-1) is met if and only if $j - 2^{k-1}$ is a multiple of 2^k, and similarly, the lower bound of Eq. (3) is met with equality if and only if j is an odd multiple of 2^{k-1} for each j that occurs with nonzero probability. Thus, the lower bound of Eq. (3) is tight when $\mu \ge 2^{k-1}$, and so it's tight when $k \le k^*_{\mathrm{max}}(\mu)$.

The bounds in Eq. (3) can be used to determine conditions on k and μ that ensure that the difference $R_k - R_{k-1}$ must be positive or must be negative, and thus constrain the range of code parameters that can be optimum given μ. Specifically, using $k^*_{\mathrm{min}}(\mu)$ and $k^*_{\mathrm{max}}(\mu)$ as defined in Eq. (2), the upper bound of Eq. (3) implies that $R_{k^*_{\mathrm{min}}(\mu)} \le R_{k^*_{\mathrm{min}}(\mu)-1}$, and the lower bound implies that $R_{k^*_{\mathrm{max}}(\mu)} \le R_{k^*_{\mathrm{max}}(\mu)+1}$. From the convexity of the rate function, $\left[k^*_{\mathrm{min}}(\mu), k^*_{\mathrm{max}}(\mu)\right]$ thus includes the range of GPO2 code parameter values that might be optimal given μ.

The fact that the optimum GPO2 code parameter value lies in the range $\left[k^*_{\mathrm{min}}(\mu), k^*_{\mathrm{max}}(\mu)\right]$ constrains the optimum parameter to at most three possible choices. That is, we can show that

$$k^*_{\mathrm{max}}(\mu) - k^*_{\mathrm{min}}(\mu) < 3$$

It's straightforward to show that this bound holds when $\mu < 3$. When $\mu \ge 3$, we derive the bound as follows:

$$k^*_{\mathrm{max}}(\mu) - k^*_{\mathrm{min}}(\mu) = \lceil \log_2 \mu \rceil - \left\lfloor \log_2 \frac{2}{3}(\mu + 1) \right\rfloor < 1 + \log_2 \mu - \left(\log_2 \left(\frac{2}{3}\mu \right) - 1 \right) = 2 - \log_2 \frac{2}{3} < 3$$

Appendix B

The Simple Code Selection Approach is Always Nearly Optimum

We now derive the bounds given in Eq. (12) of Section IV.

To establish Eq. (12), it is sufficient to show that $k^*_{\max}(\mu) - \hat{k}_f(\mu) < 2$ and $\hat{k}_f(\mu) - k^*_{\min}(\mu) < 2$. For $\mu \leq 3$, it's straightforward to show that $\hat{k}_f(\mu) - k^*_{\min}(\mu) \leq 1$. For $\mu > 3$,

$$\begin{aligned}\hat{k}_f(\mu) - k^*_{\min}(\mu) &= \lfloor \log_2(\mu + f)\rfloor - \left\lfloor \log_2\left(\frac{2}{3}(\mu+1)\right)\right\rfloor < \log_2(\mu+f) - \left(\log_2\left(\frac{2}{3}(\mu+1)\right) - 1\right) \\ &= \log_2\left(\frac{\mu+f}{\mu+1}\right) + \log_2 3 \leq \log_2 3 < 2\end{aligned}$$

Next, it's straightforward to show that $k^*_{\max}(\mu) - \hat{k}_f(\mu) \leq 1$ whenever $\mu \leq 2$ or $f = 0$. When $f > 0$ and $\mu > 2$,

$$k^*_{\max}(\mu) - \hat{k}_f(\mu) = \lceil \log_2 \mu\rceil - \lfloor \log_2(\mu+f)\rfloor < 1 + \log_2 \mu - (\log_2(\mu+f) - 1) = 2 + \log_2 \frac{\mu}{\mu+f} < 2$$

❐

Appendix C

Mapping Sample Differences to Nonnegative Integers in the CCSDS Image Compression Standard

The emerging CCSDS image compression standard [8] relies on GPO2 codes to encode the values of quantized DWT coefficients in the lowest spatial frequency subband of a DWT-transformed image. To do this, differences between successive quantized coefficients, taken in raster-scan order, are mapped to nonnegative integer values using essentially the same procedure incorporated in the CCSDS lossless data compression standard [5].

In this encoding process, in the lowest spatial frequency subband, each quantized DWT coefficient c_m is a signed N-bit quantity. Differences between successive quantized coefficients, $c_m - c_{m-1}$, are mapped to nonnegative N-bit integers δ_m according to

$$\delta_m = \begin{cases} 2(c_m - c_{m-1}), & \text{if } 0 \leq c_m - c_{m-1} \leq \theta_m \\ 2|c_m - c_{m-1}| - 1, & \text{if } -\theta_m \leq c_m - c_{m-1} < 0 \\ \theta_m + |c_m - c_{m-1}|, & \text{otherwise} \end{cases}$$

Here $\theta_m = \min\{c_{m-1} + 2^{N-1}, 2^{N-1} - 1 - c_{m-1}\}$, and we can treat c_{-1} as equal to zero. A block of J consecutive values of δ_m are then encoded using a GPO2 code.