On Redundancy of Rice Coding

A Technical Report at the ITU

Alexandre Krivoulets

IT University Technical Report Series

TR-2003-32

ISSN 1600–6100

9 2003

ISSN 1600–6100

ISBN 87-7949-043-3

Copies may be obtained by contacting:

IT University of Copenhagen
Glentevej 67
DK-2400 Copenhagen NV
Denmark

Telephone:	+45 38 16 88 88
Telefax:	+45 38 16 88 99
Web	www.it-c.dk

On Redundancy of Rice Coding

Alexandre Krivoulets

Abstract

In this paper we derive the relative per-symbol redundancy of the Rice coding algorithm, which is a widely used technique in image compression for very fast entropy coding. We show, that for some important source models, such as the two-sided geometric distribution (TSGD), the redundancy depends on the source entropy H and it tends to zero if $H \to \infty$. The redundancy is upper bounded by 50% if $H \to 0$.

1 Introduction

Rice coding [1] (or the Rice Algorithm) is a widely used technique in image compression for entropy coding due to its efficiency and simple implementation. It is recommended as the base of a standard for space image compression applications [2]. In a modified version it is used in the recent standard for lossless image compression JPEG-LS [3] as a part of the entropy coder. By generalized Rice coding we assume a technique that consists of *Rice preprocessing* [2] followed by run-length coding using *Golomb* [4] or Rice codes, also called *Golomb-power-of-2* [3] (GP2) codes. We describe the technique in details below.

The method is applied to a source modeled by integers with a probability mass function $P(i)$, $i \in \mathbb{Z}$, which satisfies the property:

$$P(0) \geq P(+1) \geq P(-1) \geq P(+2) \geq \dots.$$

Rice coding consists of two steps. In the *preprocessing step* a source symbol $i \in \mathbb{Z}$ is first mapped into an index $j = \{0, 1, \dots\}$ in the sequence of symbols arranged in order $0, +1, -1, +2, -2, \dots$. Each index is then unarily coded and the sequence of index codewords is concatenated to form a sequence of binary symbols, which is called the *fundamental sequence* (FS). (In unary representation, a nonnegative integer $j = \{0, 1, 2, \dots\}$ is mapped into a sequence of j binary symbols 0 followed by a 1). The output of the preprocessing step (the fundamental sequence) is entropy coded using run-length coding based on Golomb or GP2 codes.

The key property of the method is that if the most probable symbol in the FS is zero, which is often the case when one coding prediction residuals, then the algorithm performs symbolwise coding. Otherwise, it performs blockwise coding, thus allowing for entropy coding of sources like the DCT or wavelet transform coefficients after quantization. In all the cases, the attractive feature of the method for practical implementation is that there is no need to store any code tables. Given the source symbol or the block, its codeword can be merely calculated. The calculation is simpler for GP2 codes (although, for some loss in compression performance).

The goal of this paper is to investigate the efficiency of this algorithm in terms of relative per-symbol redundancy as a function of the source parameters. This will allow us to see the potential performance of Rice coding technique for different practical situations.

To find the redundancy, in Section 2 we consider Rice coding from a binary decomposition point of view. This will show more clearly the basics of the method and will lead to a redundancy estimation technique. In Section 3, we introduce a parametric model of the input source that will allow for calculation of the redundancy. We also assume throughout the paper, that we apply the algorithm to a source with *known* parameters[1] and derive the redundancy as a function of these parameters. In Section 4, we derive analytical solution for the redundancy based on the source model and discuss the results.

2 Binary decomposition and Rice coding

Binary decomposition of source symbols combined with binary arithmetic coding is a well known technique for coding of m-ary sources (see, e.g., [5]). A general idea of this method is that any proper and complete binary tree with m leaves can be used to represent symbols from an m-ary source $A = \{a_1, a_2, \ldots, a_m\}$ with any probability distribution. A source symbol is represented by a sequence of binary decisions when passing the tree from the root to the leaf, corresponding to this symbol. The sequence of decisions can be regarded as a sequence of binary symbols generated by a Markov source modeled by this tree. A binary tree with m leaves has $K = m - 1$ nodes. To each node η_k, $k = 1, 2, \ldots, K$, of the decomposition tree there corresponds a parameter q_k, which is the probability of a binary symbol (decision) being '0'. These parameters are uniquely defined by the probability distribution of the source symbols. The sequence of binary decisions can be decomposed into K subsequences of statistically independent binary symbols with probability distributions q_k, corresponding to each node. These subsequences are to be encoded using some kind of binary coding technique.

It is easy to see now, that Rice preprocessing can be viewed as using a unary tree for decomposition of symbols arranged in order $0, +1, -1, +2, -2, \ldots$ (see Figure 1). From this point of view the FS is in essence a sequence of decisions, which is run-length coded using Golomb or GP2 codes, treating it as a binary *memoryless* source. In this case, a single parameter, which we shall denote $\hat{q}$, characterizes the binary sequence. It corresponds to the zero-order probability of a decision being '0'. Let

$$\bar{n} = \sum_{a \in A} P(a) n(a)$$

be the average number of decisions per symbol, where $n(a)$ is the number of binary decisions required to code the source symbol a. Then, $\hat{q}$ can be found as $\hat{q} = 1 - 1/\bar{n}$. Assuming that the sequence of decisions is a memoryless binary source, its entropy is

$$h = -\hat{q} \log \hat{q} - (1 - \hat{q}) \log(1 - \hat{q}),$$

[1]The parameter estimation technique is outside the scope of our paper. For different techniques the interested reader is referred to [1, 3].

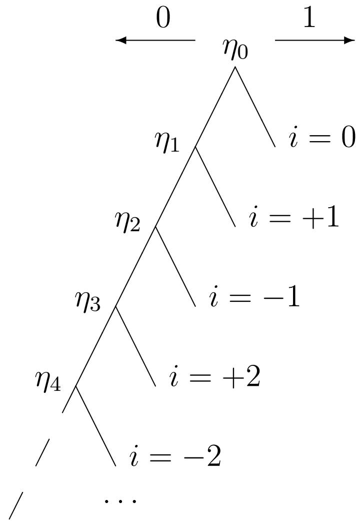

Figure 1: Unary decomposition of source symbols.

and hence we can introduce the *quasi* entropy of the input source by

$$\hat{H} = h\bar{n} = \bar{n}\log(\bar{n}) - (\bar{n} - 1)\log(\bar{n} - 1). \tag{1}$$

We estimate the redundancy, caused by the assumption of memorylessness by the ratio

$$\varrho_0 = \frac{\hat{H} - H}{H}, \tag{2}$$

where $H = -\sum_{i \in Z} P(i) \log P(i)$ is the real entropy of the source. Now we have to define the probability distribution of source symbols in order to calculate the actual redundancy.

3 The source model

We assume, that Rice coding is mainly used to code source symbols, such as prediction residuals in lossless image compression or transform coefficients after quantization in lossy image compression. For this kinds of sources we introduce the generalized two-sided geometric distribution (GTSGD).

The distribution is deduced as a dead-zone quantization of the off-centered continuous Laplacian distribution

$$f(x) = \frac{\alpha}{2} \exp^{-\alpha|x-\varepsilon|},$$

which was shown to be a good approximation of the distribution of prediction errors and transform coefficients [7, 8]. The distribution is specified by the decay parameter α and the offset ε. The dead-zone quantizer is non-uniform and specified by the zero/non-zero quantization intervals Q_0 and Q, respectively. The qunatizer and the distribution are depicted on Figure 2. We assume that $\varepsilon < Q_0/2$, i.e., the distribution center falls into the zero quantization bin. This restriction is justified by the fact, that in practice for transform coefficients it normally holds, and for context-based prediction schemes the

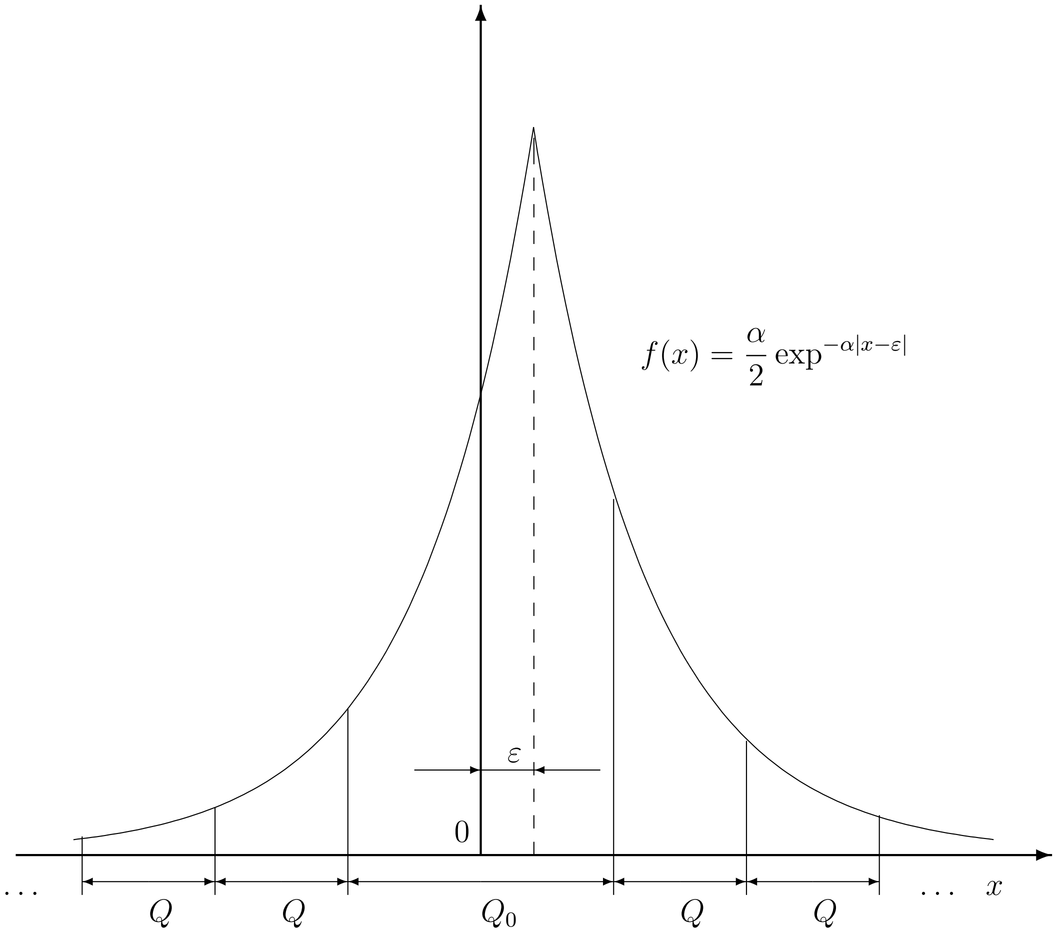

Figure 2: Example of the off-centered Laplacian distribution and the dead-zone quantizer.

unit interval containing the center of the distribution can be located by an error feedback loop [3, 9].

The probability distribution of output symbols $i \in \mathbb{Z}$ of the quantizer is defined as

$$p(i = 0) = \int_{-\frac{Q_0}{2}}^{\frac{Q_0}{2}} f(x)dx,$$

$$p(i > 0) = \int_{\frac{Q_0}{2}+(i-1)Q}^{\frac{Q_0}{2}+iQ} f(x)dx,$$

$$p(i < 0) = \int_{-\frac{Q_0}{2}+iQ}^{-\frac{Q_0}{2}+(i+1)Q} f(x)dx,$$

and after a little algebra we find

$$p(i)=\begin{cases}1-\frac{1}{2}\theta^{\frac{\lambda}{2}}\left(\theta^{\gamma}+\theta^{-\gamma}\right), & i=0\\ \frac{1}{2}\theta^{\frac{\lambda}{2}-\gamma-1}(1-\theta)\theta^{i}, & i>0\\ \frac{1}{2}\theta^{\frac{\lambda}{2}+\gamma-1}(1-\theta)\theta^{|i|}, & i<0\end{cases}\tag{3}$$

where $\theta=\exp^{-\alpha Q}$, $\gamma=\varepsilon/Q$ and $\lambda=Q_0/Q$ are the new distribution parameters. Introducing new parameters, we disengage from the Laplacian distribution and the quantizer parameters. The parameter λ controls the probability of the zero symbol, whereas γ and θ define the off-set and the rate of decay of the distribution, respectively.

The generalized two-sided geometric distribution (3) comprises the two-sided geometric distribution (TSGD) proposed in [6] for modeling prediction residuals in lossless image compression algorithms:

$$p(i)=\frac{(1-\theta)\theta^{|i+\gamma|}}{\theta^{1-\gamma}+\theta^{\gamma}},\tag{4}$$

where θ and γ have the same meaning as in (3). It can be derived from (3) by setting

$$\lambda=\frac{Q_0}{Q}=2\left(1+\frac{\ln 2-\ln(\theta^{1-\gamma}+\theta^{\gamma})}{\ln\theta}\right).\tag{5}$$

4 Redundancy

In Rice coding, the decisions are treated as a *memoryless* binary source, i.e., assuming that all the nodes of the decomposition tree have the same probability of a decision to be '0'. This is true only if the index in the sequence of rearranged source symbols in non-increasing probability order have one-sided geometric probability distribution[2]. In all other cases such a coding will result in some redundancy, depending on the distribution of the source[3].

Given the distribution (3), $\bar{n}$ is defined by

$$\bar{n}=1+\frac{\theta^{\frac{\lambda}{2}}}{2(1-\theta)}\left(2\theta^{\gamma}+\theta^{-\gamma}(1+\theta)\right).\tag{6}$$

and the entropy of the source is

$$H=-\left(1-\frac{1}{2}\theta^{\frac{\lambda}{2}}(\theta^{\gamma}+\theta^{-\gamma})\right)\log\left(1-\frac{1}{2}\theta^{\frac{\lambda}{2}}(\theta^{\gamma}+\theta^{-\gamma})\right)$$
$$-\frac{1}{2}\theta^{\frac{\lambda}{2}}\left[\left(\left(\frac{\lambda}{2}-1\right)\log_2\theta+\log_2(1-\theta)+\frac{\log_2\theta}{1-\theta}-1\right)\left(\theta^{\gamma}+\theta^{-\gamma}\right)+\gamma\log_2\theta\left(\theta^{\gamma}-\theta^{\gamma}\right)\right].\tag{7}$$

[2] $p(j)=(1-\Theta)\Theta^{j}$, $0<\Theta<1$, $j=0,1,2,\dots$ [4].

[3] This redundancy can be thought of as a measure of "closeness" to the one-sided geometric distribution.

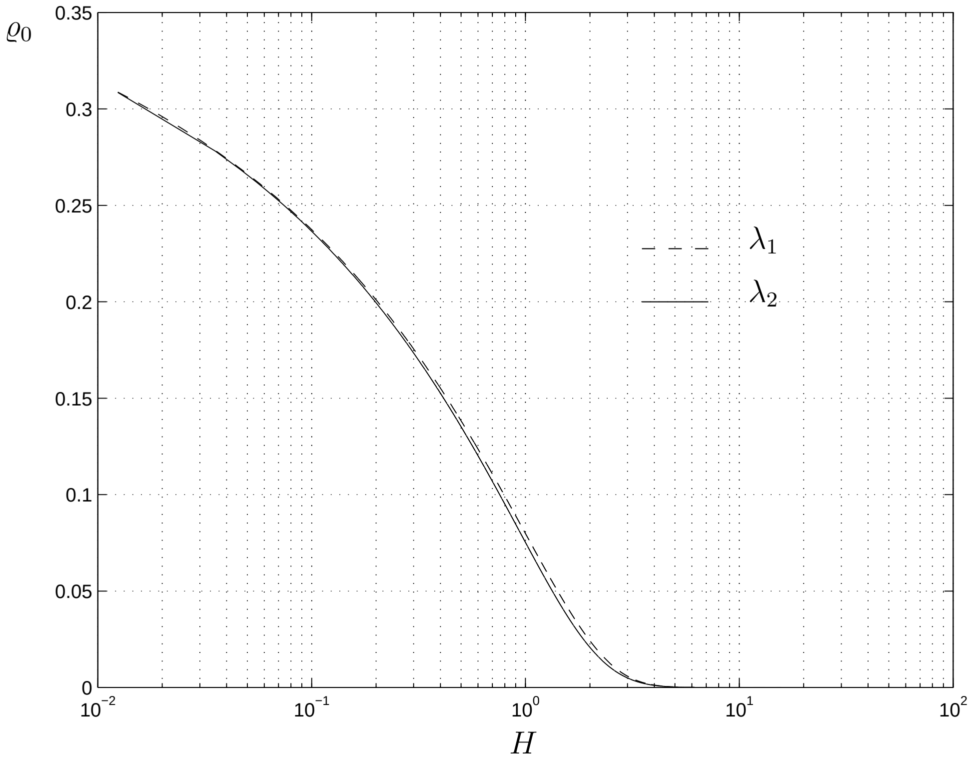

Figure 3: Relative per-symbol redundancy of Rice coding as a function of the source entropy for λ_1 and λ_2.

Using (1), (6) and (7) the redundancy ϱ_0 can be easily calculated by (2).

We shall consider the behaviour of ϱ_0 for λ_1 defined by (5) (i.e., for the distribution (4)) and $\lambda_2 = 1$ (for uniform quantization). Given θ, ϱ_0 has its maximum when $\gamma = 0$ for both cases. Thus, setting $\gamma = 0$ we get the upper bound on ϱ_0. It can be readily shown, that for λ_1 and λ_2: $\lim_{\theta\to 1} \varrho_0(\theta) = 0$ and $\lim_{\theta\to 0} \varrho_0(\theta) = 0.5$.

Figure 3 shows the relative redundancy as a function of the entropy for sources defined by λ_1 and λ_2. One can see, that in both cases the redundancy is approximately the same, being less than 10% for sources with the entropy $H \geq 1$ bit. It tends to zero if $H \to \infty$. The redundancy is also upper bounded by 50% if $H \to 0$.

Note, that ϱ_0 is the "ideal" relative redundancy, that is caused only by the assumption of *memorylessness* of the sequence of decisions. It allows to see potential efficiency of the algorithm. In order to estimate the actual redundancy, we have to add the redundancy, caused by the method of coding. The overall relative redundancy is defined by

$$\varrho = \frac{\hat{H} + \bar{n}\rho_b}{H} - 1,$$

where ρ_b is the *absolute* per (binary) symbol redundancy, caused by the method of binary coding. If GP2 codes are used to code the sequence of the decisions, then

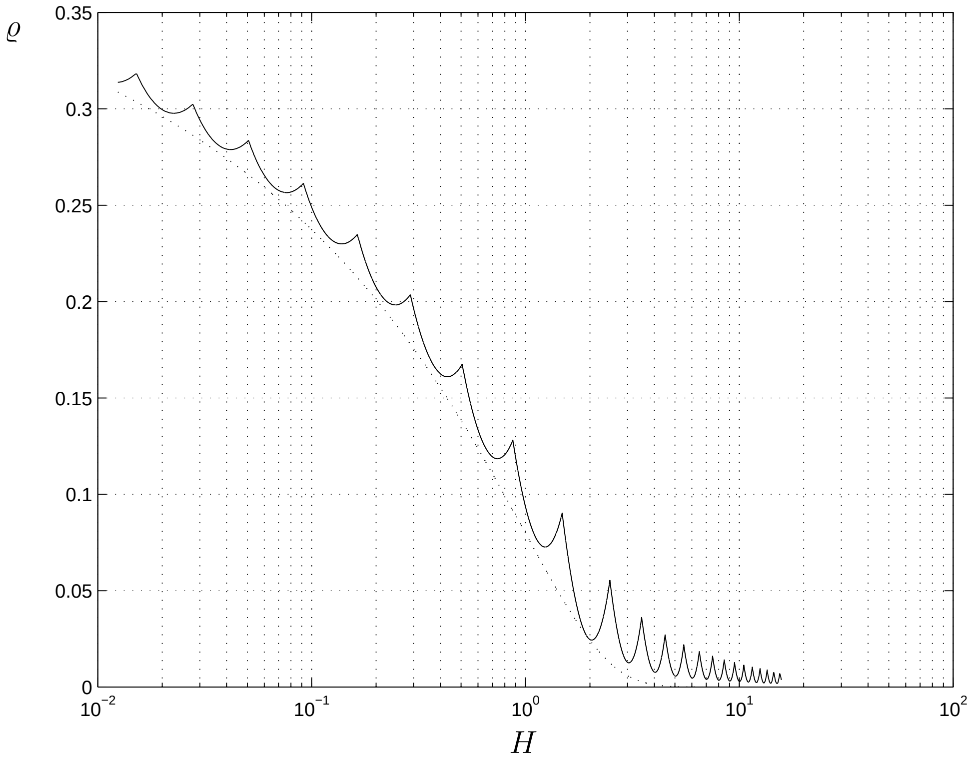

Figure 4: Relative redundancy of the algorithm if GP2 codes are used to code the sequence of binary decisions and the model is defined by λ_1. The dotted line shows the 'ideal' redundancy from Figure 3 for comparison purpose.

$$\rho = \frac{l + 1 + \hat{q}_m^l (1 - \hat{q}_m^l)^{-1}}{1 - \hat{q}_m},$$

where $\hat{q}_m = \max\{\hat{q}, 1 - \hat{q}\}$ is the probability of the most likely symbol (decision) and

$$l = \left\lfloor 1 - \log_2 \left(\frac{\log_2 \hat{q}_m}{\log_2(\sqrt{5} - 1) - 1} \right) \right\rfloor$$

is the parameter[4].

Figure 4 shows the resulting relative redundancy as a function of the entropy for GP2 codes and the source defined by λ_1 (we assumed $\gamma = 0$, thereby, this figure shows the upper bound). The relative redundancy is less than 10% if the entropy $H \geq 1$. In practice, for the low entropy sources one may expect a redundancy of about 10…30%. In some practical implementations this may be reasonable price for possibility of very fast coding.

[4]The derivation of this formula is based on Lemma 4 from [6].

References

[1] R. F. Rice, "Some practical universal noisless coding technique," *JPL Publication 79-22*, March 1979.

[2] R. F. Rice, "Lossless coding standards for space data systems," in *Conference Record of the Thirtieth Asilomar conference on Signals, Systems and Computers*, 1997, vol. 1, pp. 577–585.

[3] M. J. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS," *IEEE Trans. Image Processing*, vol. 9, no. 8, pp. 1309–1324, 2000.

[4] S. Golomb, "Run-length encodings," *IEEE Trans. Inform. Theory*, vol. 12, no. 3, pp. 399–401, 1966.

[5] P. G. Howard, "The design and analysis of efficient lossless data compression systems," Tech. Rep. CS-93-28, Brown Uneversity, Providence, Rhode Island, 1993.

[6] J. O'Neal, "Predictive quantizing differencial pulse code modulation for the transmission of television signals," *Bell Syst. Tech. J.*, vol. 45, no. 1966, pp. 689–722.

[7] J. D. Gibson R. C. Reininger, "Distributions of the two-dimentional DCT coefficients for images," *COM*, vol. 31, no. 6, pp. 835–839, 1983.

[8] X. Wu and N. Memon, "Context-based, adaptive, lossless image codec," *IEEE Trans. Commun.*, vol. 45, no. 4, pp. 437–444, 1997.

[9] N. Merhav, G. Seroussi, and M. J. Weinberger, "Optimal prefix codes for two-sided geometric distributions," *IEEE Trans. Inform. Theory*, vol. 46, no. 1, pp. 121–135, 2000.