

Submit for publication for DCC94

**Adaptive Source Coding Schemes for Geometrically
Distributed Integer Alphabets ***

Kar-Ming Cheung and Padhraic Smyth

Jet Propulsion Laboratory
Pasadena, CA 91109 USA

ABSTRACT

In this article we revisit the Gallager and van Voorhis optimal source coding scheme for geometrically distributed non-negative integer alphabets and show that the various subcodes in the popular Rice algorithm can be derived from the Gallager and van Voorhis code. Next we modify and generalize the Gallager and van Voorhis code for 2-sided geometrically distributed integer alphabets (positive and negative), which are typical input samples to the back-end entropy coding stage of lossless predictive coding schemes and lossy transform coding schemes. Based on this code we propose an adaptive coding scheme with low implementation complexity and present experimental results on compressing planetary images using the proposed method.

Draft #2 11/11/93

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

Adaptive Source Coding Schemes for Geometrically Distributed Integer Alphabets

EXTENDED ABSTRACT

I. Introduction

Predictive coding schemes predict the present sample value based on the previous samples. The error samples, which are the difference between the predicted and actual values, are either sent directly to an entropy coder in the case of lossless compression, or are quantized before they are sent to an entropy coder in the case of lossy compression. Transform coding decorrelates the original signal and this decorrelation generally results in the signal energy being redistributed among a smaller set of transform coefficients. The transform coefficients are quantized before they are sent to an entropy coder. It is well-known that the error samples and the transform coefficients before quantization can be modelled with a Laplacian distribution [1, 2]. In [3] and [4], we introduced an improved modified Laplacian distribution for the unquantized outputs and we showed that the probability distribution of the quantized output integers derived from the modified Laplacian distribution is geometric for all integers except zero. This discrepancy can be shown to be small and the quantized output integers can be modelled using a simple single-parameter discrete 2-sided geometric probability distribution.

In Section II we revisit the Gallager-van Voorhis-Huffman (GVH) optimal source coding scheme for geometrically distributed non-negative integer alphabets [5] where

$$p_{G1}(i) = (1 - \theta)\theta^i \quad \forall i \geq 0, \quad (1)$$

where $\theta = 1 - r(0)$, $r(0)$ is the fraction of zeros in the sample set, and i is a non-negative integer. $r(0)$ can be estimated directly from the image data. We show in Section III that the various subcodes in the popular Rice algorithm can be derived from the GVH code. In fact, the Rice! subcodes are also a subset of the class of optimal codes for runlength encodings proposed by Golomb [6]. In Section IV we modify and generalize the GVH code for 2-sided geometrically distributed integer alphabets (positive and negative), which have the following distribution

$$p_{G2}(i) = \frac{1 - \theta}{1 + \theta} \theta^{|i|} \quad \forall i, \quad (2)$$

where $\theta = \frac{1-r(0)}{1+r(0)}$, and $r(0)$ is the fraction of zeros in the sample set. The 2-sided geometrically distributed integer alphabets are typical inputs to the back-end entropy coding stage of lossless predictive coding schemes and lossy transform coding schemes. In Section V we develop an adaptive coding scheme which has low implementation complexity. We present some experimental results for this scheme using planetary images.

II. Background on GVH Codes

Gallager and Van Voorhis presented an optimal binary prefix code for the set of geometrically distributed nonnegative integers [5]. Here we call this code the Gallager-van Voorhis-Huffman-1 (GVH1) code. This code is a generalization of Golomb's optimal codes for runlength encodings for the special case when $\theta^l = 1/2$ [5]. Let l be the integer satisfying

$$\theta^l + \theta^{l+1} \leq 1 < \theta^l + \theta^{l-1}, \quad (3)$$

where $\theta = 1 - r(0)$ as defined in (1). It is easy to see that for any θ , $0 < \theta < 1$, there is a unique positive integer l satisfying (3). Let a non-negative number i be represented by $i = lj + r$ where $j = \lfloor i/l \rfloor$, the integer part of i/l , and $r = [i] \bmod l$. Gallager and Van Voorhis derived the optimal l (hence the optimal Huffman code) as a function of θ to minimize the code redundancy. They also showed that an optimal code for the non-negative integers is the concatenation of a unary code which is used to encode j , and a Huffman code which is used to encode r , $0 \leq r \leq l - 1$.

Each integer r , $0 \leq r \leq l - 1$, represents an equivalence class modulo l . Gallager and Van Voorhis showed that the integer set $\{r : 0 \leq r \leq l - 1\}$ has a distribution $p_r = \frac{1-\theta^l}{1-\theta} \theta^r$, and the sum of the two least likely letters exceeds the probability of the most likely. The length of the optimal codewords can differ by at most one. It can be shown that the optimal coding for this integer set is to use codewords of length $\lfloor \log_2 l \rfloor$ for $i < 2^{\lfloor \log_2 l \rfloor + 1} - l$, and codewords of length $\lfloor \log_2 l \rfloor + 1$, otherwise.

In [3, 4] we proposed a simple construction to generate a Huffman code for the integer set $\{r : 0 \leq r \leq l - 1\}$. The construction algorithm is as follows:

1. Generate the preliminary list L of $2^{\lfloor \log_2 l \rfloor}$ binary sequences $\{00 \dots 0, \dots, 11 \dots 1\}$, each of which has length $\lfloor \log_2 l \rfloor$.
2. Append to each of the last $l - 2^{\lfloor \log_2 l \rfloor}$ binary sequences in L either a 0 or a 1 to generate two binary sequences of length $\lfloor \log_2 l \rfloor + 1$, and **call the** new list L' .

L' has a list of l prefix-conditioned codewords, with $2^{\lfloor \log_2 l \rfloor + 1} - l$ codewords of length $\lfloor \log_2 l \rfloor$, and the rest of length $\lfloor \log_2 l \rfloor + 1$. L' is an optimal Huffman code for $\{r : 0 \leq r \leq l - 1\}$. Thus, each 1-sided geometrically distributed integer can be efficiently encoded using a concatenation of a unary code and a Huffman code. "For the particular case when $l = 2^k$, it is not hard to see that an integer i encoded by the GVH1 code consists of it concatenation of a unary code of length $\lfloor \frac{i}{2^{n-k}} \rfloor + 1$ ($\lfloor \frac{i}{2^{n-k}} \rfloor$ 0's followed by a 1) and a k -tuple (L' consists of $2L$ k -tuples), where n is the length of a symbol in bits (e.g., typically $n = 8$ in image transmission). This particular code construction WILS also described in [6].

III. Relationship Between the 1-sided GVH Code and the Rice Code

Rice developed a predictive lossless coding scheme [7] that consists of two separate stages: the front-end pre-processor is a predictor followed by a symbol mapper, while the second part performs adaptive entropy coding. The first stage takes the difference between the actual values and the predicted values and maps the differences, positive or negative, to a sequence of non-negative integer numbers. The second stage encodes the sequence by adaptively selecting the best of several easily implemented variable length coding algorithms for non-negative integers. The software Rice code was used in the interplanetary Voyager Mission, and the hardware Rice implementation has been baselined for the Cassini Mission (scheduled for launch in the late 1990's timeframe). Recently a new VLSI implementation of the lossless Rice algorithm was reported [8]. The encoder/decoder chip set supported 4 to 14 bits/sample. It was reported that under laboratory conditions, the encoder chip compresses at a rate in excess of 50 Msamples/s, and the decoder operates at 25 Msamples/s. A second Rice encoder has been designed as a gate array and is being fabricated in a 1.2 μ M RAD-hard CMOS process.

Using Rice's notation in [7], it was shown in [9] that the various variable length codes that consists of the fundamental sequence (FS) code Ψ_1 and the split-sample codes $\Psi_{1,k}$ are optimal Huffman codes for data sources that have Laplacian distributions. For a non-negative integer i ,

$$\Psi_{1,k}(i) = \Psi_1\left(\left\lfloor \frac{i}{2^{n-k}} \right\rfloor\right) * LSB_k(i) \quad (3)$$

where

$$\Psi_1(m) = \underbrace{000 \dots 001}_{m \text{ zeros}}, \quad (4)$$

and where $*$ denotes the bit-pattern concatenation operation and LSB_k denotes the k least significant bits of i . From the results of Section II, we can thus observe that the fundamental sequence code Ψ_1 is equivalent to the GVH1 code and the **Go1oIII1** code for $l = 1$, and the split-sample codes $\Psi_{1,k}$ are equivalent to the GVH 1 code and the Golomb code for $l = 2^k$. Hence, one can state that the well-known Rice code can be interpreted as a special case of Golomb's code, which in turn is a special case of the GVH1 code.

IV. Efficient Coding Based on the 2-sided Geometric Model

Constructing an optimal prefix code, say by using the Huffman algorithm, is quite a complex operation in hardware. In this section we introduce a class of near-optimal prefix codes to encode data (e.g. differentials of waveform data and image data) with

probability distributions that resemble the 2-sided geometric models discussed in Section II. The construction of this prefix code is simple. For most well-behaved data, $\text{frequency}(i) \approx \text{frequency}(-i)$ for $i = 1, 2, \dots$. Thus in order to construct a code for both the positive and negative values, we use the GVH 1 codes for the non-negative integers. An additional bit is appended to each codeword, except the codewords representing 0, to indicate whether integer i or integer $-i$ is sent. We call this code the Gallager-van Voorhis-Huffman-2 code.

Based on the above code construction, we can evaluate the performance of the GVH2 codes and give closed form analytic expressions as a function of θ for the redundancy r_2 , the mean codelength \bar{l}_2 , and the entropy $H(X_2)$ of the 2-sided integer geometric distribution, where X_2 is the discrete random variable corresponding to the 2-sided geometric source [3, 4]. Hence, from Appendix 1 and 2, we write down a closed form expression for the redundancy of our coding scheme as a function of θ and l , namely,

$$\begin{aligned} \bar{r}_2 &= \bar{l}_2 - H(X_2) \\ &= 1 + \lfloor \log_2(l) \rfloor + \frac{2}{1+\theta} \left(\theta + \frac{\theta^k}{1-\theta^l} \right) - \log_2 \left(\frac{1+\theta}{1-\theta} \right) + \frac{2\theta \log_2(\theta)}{(1+\theta)(1-\theta)} \end{aligned} \quad (5)$$

We find the value of l which minimises \bar{r}_2 for given θ by minimising the terms in \bar{r}_2 which depend on l , namely

$$f(l) = \lfloor \log_2(l) \rfloor + \frac{2}{1+\theta} \left(\frac{\theta^k}{1-\theta^l} \right) \quad (6)$$

We find the optimal l values (over all ranges of θ of interest) by direct search, and we tabulate in Table 1 the ranges of $r(0) = -$ for which each value of l is optimal, $1 \leq l \leq 30$. Note in particular that some values of l are not used in Table 1, and that the ranges are different from the 1-sided case (for small values of l) as given in [5].

v. An Adaptive Coding Scheme Based on the 2-Sided Geometric Distribution

The GVH codes described in previous sections are static compression schemes, and the efficiency of a code depends on how well the code (as a function of l) matches the source statistics (e.g., $r(0)$). In practice, due to the uncertainties associated with the data, a static data compression scheme may cause source-model mismatch. The source-model mismatch can reduce the efficiency of the compression scheme and in some cases, may cause data expansion. In light of this, we have developed an adaptive lossless data compression scheme that does not require prior knowledge of the source statistics. The only requirement is that the source statistics should resemble a 2-sided geometric model. This scheme uses the same basic adaptation strategy as the Rice algorithm: use a number of different codes to compress the data and choose the best one. The GVH-based adaptive data compression method was developed for the Galileo Low Gain Antenna Mission [10] and although is not being used as part of the flight software baseline for Galileo, it may be considered for future missions,

The adaptive lossless data compression scheme is differential-pulse-code-modulation (DPCM) based and uses a Huffman coding strategy similar to the one used to compress the DC differentials of the JPEG [11] and ICT [10] [12] compression schemes. We developed three Huffman codebooks that are based on the '2-sided geometry model: one for low-activity data ($l = 1$), one for medium-activity data ($l=2$), and one for high-activity data ($l=4$). The data are first partitioned into blocks of fixed length (e.g., 16 samples per Mock). The first sample of each block is used as a reference point and is not coded. For the remaining samples the differences between adjacent samples are calculated. The encoder then computes the number of hits that are required to compress the block using each of the predefined codebooks and chooses the codebook that gives the best compression. If all code books give data expansion, the block is sent unencoded. Each block is preceded by a 2-bit tag: 00 for the low-activity codebook, 01 for the medium-activity codebook, 10 for the high-activity code book, and 11 for no compression.

This adaptive data compression scheme has an escape code that prevents data expansion. Like the Rice algorithm, this scheme is adaptive to local statistics (one codebook per block) rather than depending on global statistics (one codebook for the whole data file). Hence, in principle, it can avoid the source-model mismatch problem by choosing that code (from a family of codes) which performs best on the actual data. The lossless compression performances of this scheme on 19 planetary images are given in Figure 1. The planetary images (of Jupiter) were chosen by the Galileo flight project team to reflect the potentially wide variety of realistic images which the spacecraft may encounter at the planet. On all images except one, the adaptive method outperforms the global non-adaptive GVH2 method of Section IV. The single image where it performed worse was in fact an image of nearly constant background sky; hence, not surprisingly the adaptive approach pays a slight performance penalty over the global method. For some images (such as images 7, 8 and 9) the bits/symbol for the adaptive method are actually lower than the differential entropy of the whole image. This can be explained by the fact that the differential entropy is a global image statistic, whereas the adaptive method is based on **local** statistics. Thus the adaptive method can exploit **local** variations in entropy to improve overall compression performance.

The prototype compressor contains 173 lines of C code which includes three codebooks. The coding architecture is flexible enough to accommodate different codebooks and different numbers of codebooks to fit different applications. The scheme requires 44 bytes per codebook (memory requirement), and one addition per byte per codebook (computational requirement).

This scheme can be generalized to an adaptive combined runlength/Huffman coding algorithm for block transform coding schemes like JPEG, ICT, and Hadamard transform. Other than the DPCM-based schemes, the scheme can also be used as an efficient back-end entropy coder for subband coding.

Acknowledgements

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

References

- [1]. A. Jain, "Image Data Compression: A Review," *Pmt. IEEE*, Vol. 69, No. 3, March 1981.
- [2]. J. Sullivan, "Quantization for MMSE Linear Predictive Image Coding Using Block-Adaptive Gain and Bias," *Proceedings of the International Conference on Digital Signal Processing-87*, Florence, Italy, 7-10 September, 1987.
- [3]. K. Cheung and P. Smyth, "A High-Speed Distortionless Predictive Image Compression Scheme," TDA Progress Report 42-101 vol. January-March, 1990, Jet Propulsion Laboratory, Pasadena, CA.
- [4]. K. Cheung and P. Smyth, "A high speed distortionless predictive image compression scheme," in *Proceedings of the IEEE Symposium on Information, Theory and its Applications*, Hawaii, Dec., 1990.
- [5]. R. Gallager and D. van Voorhis, "Optimal Source Codes for Geometrically Distributed Integer Alphabet s," *IEEE Trans. Inform. Theory*, vol. IT-21, March 1975.
- [6]. S. Golomb, "Run-Length Encodings," *IEEE Trans. Inform. Theory*, vol. IT-12, July 1966.
- [7]. R. Rice, "Some Practical Universal Noiseless Coding Techniques," Part I and II, JPL Publications 79-22 and 83-17, March 1979.
- [8]. J. Venbrux, J. Gambles, D. Wiseman, P. Yeh, and W. Miller, "A VLSI Chip Set Development for Lossless Data Compression," *Proceedings of the AIAA Computing in Aerospace 9 Conference*, San Diego, October 1993.
- [9]. P. Yeh, R. Rice, and W. Miller, "On the Optimality of a Universal Noiseless Coder," *Proceedings of the AIAA Computing in Aerospace 9 Conference*, San Diego, October 1993.
- [10]. K. Cheung and K. Tong, "Proposed Data Compression Schemes for the Galileo S-Band Contingency Mission," *Proceedings of the 1993 Space & Earth Science Data Compression Workshop*, Snowbird, Utah, April 2, 1993.
- [11]. W. Pennebaker and J. Mitchell, "JPEG Still Image Data Compression Standard", New York, van Nostrand Reinhold, 1993.
- [12]. W. Chain, "Development of Integer Cosine Transform by the Principle of Dyadic Symmetry," *IEE Proceedings*, Vol 136, August 1989.

Appendix 1

We define the GVH2 code for the 2-sided model as described earlier, i.e., for $i \neq 0$ an extra sign bit is appended to the equivalent codeword for a single-sided source. If we define $l_2(i)$ as the length of the codeword assigned to letter i by this scheme then we must have

$$\begin{aligned} l_2(i) &= l_1(i) + 1, & i \neq 0 \\ &= l_1(0), & i = 0 \end{aligned}$$

Let \bar{l}_2 be the mean codeword length for the GVH2 code. We have

$$\begin{aligned} \bar{l}_2 &= \sum_{i=-\infty}^{i=\infty} p_2(i) l_2(i) \\ &= p_2(0) l_1(0) + \sum_{i=1}^{i=\infty} p_2(i) (l_1(i) + 1) + \sum_{i=-\infty}^{i=-1} p_2(i) (l_1(|i|) + 1) \\ &= 2 \sum_{i=0}^{\infty} p_2(i) l_1(i) - 1 - p_2(0) - p_2(0) l_2(0) \end{aligned}$$

But by definition

$$p_2(i) = \frac{p_1(i)}{1 + \theta}$$

which leads to

$$\bar{l}_2 = \frac{2}{1 + \theta} \bar{l}_1 + 1 - p_2(0) - p_2(0) l_2(0),$$

where \bar{l}_1 is the mean code length of the GVH1 code, and is given by the following expression [5]: * is

$$\bar{l}_1 = \lfloor \log_2(l) \rfloor + 1 + \frac{\theta^k}{1 - \theta^l}$$

Since we also have in general that

$$l_2(0) = 1 + \lfloor \log_2(l) \rfloor$$

and

$$p_2(0) = \frac{1 - \theta}{1 + \theta}$$

* We note that this result is different from that given in Gallager and Van Voorhis' original paper [5] - there appears to be a typographical error in their equation for \bar{l}_1 , they have the term $\lfloor \log_2(l) \rfloor$ instead of $\lfloor \log_2(l) \rfloor$.

We can write

$$\begin{aligned}\bar{l}_2 &= \frac{2}{1+\theta}([\log_2(l)] + 1 + \frac{\theta^k}{1-\theta^l}) + 1 \frac{1-\theta}{1+\theta} \frac{1-\theta}{1+\theta} (1 + [\log_2(l)]) \\ &= 1 + [\log_2(l)] + \frac{2}{1+\theta}(\theta + \frac{\theta^k}{1-\theta^l})\end{aligned}$$

Hence we see that the mean codelength for the 2-sided GVH coding scheme is quite similar in form to the 1-sided GVH result. Clearly however the difference in the two forms may lead to different optimal values of the parameter 1, for fixed θ , i.e., \bar{l}_1 and \bar{l}_2 may be minimised by different values of 1 over certain ranges of θ .

Appendix 2

We seek an expression for the entropy of a 2-sided geometric source as a function of θ . We have

$$\begin{aligned}H(X_2) &= - \sum_{i=-\infty}^{i=\infty} p_i \log_2(p_i) \\ &= - \sum_{i=-\infty}^{i=\infty} \left(\frac{1-\theta}{1+\theta}\right) \theta^{|i|} \log_2\left(\left(\frac{1-\theta}{1+\theta}\right) \theta^{|i|}\right) \\ &= \left(\frac{1-\theta}{1+\theta}\right) \left(\log_2\left(\frac{1-\theta}{1+\theta}\right) \left(1 + 2 \sum_{i=1}^{i=\infty} \theta^i\right) - \log_2(\theta) \left(2 \sum_{i=1}^{i=\infty} i \theta^i\right)\right) \\ &= \left(\frac{1-\theta}{1+\theta}\right) \left(\log_2\left(\frac{1-\theta}{1+\theta}\right) \left(1 + \frac{2\theta}{1-\theta}\right) - \frac{2 \log_2(\theta) \theta}{(1-\theta)^2}\right) \\ &= \log_2\left(\frac{1+\theta}{1-\theta}\right) - \frac{2\theta \log_2(\theta)}{(1-\theta)(1+\theta)}\end{aligned}$$

GVH Compression Performance on Planetary Images

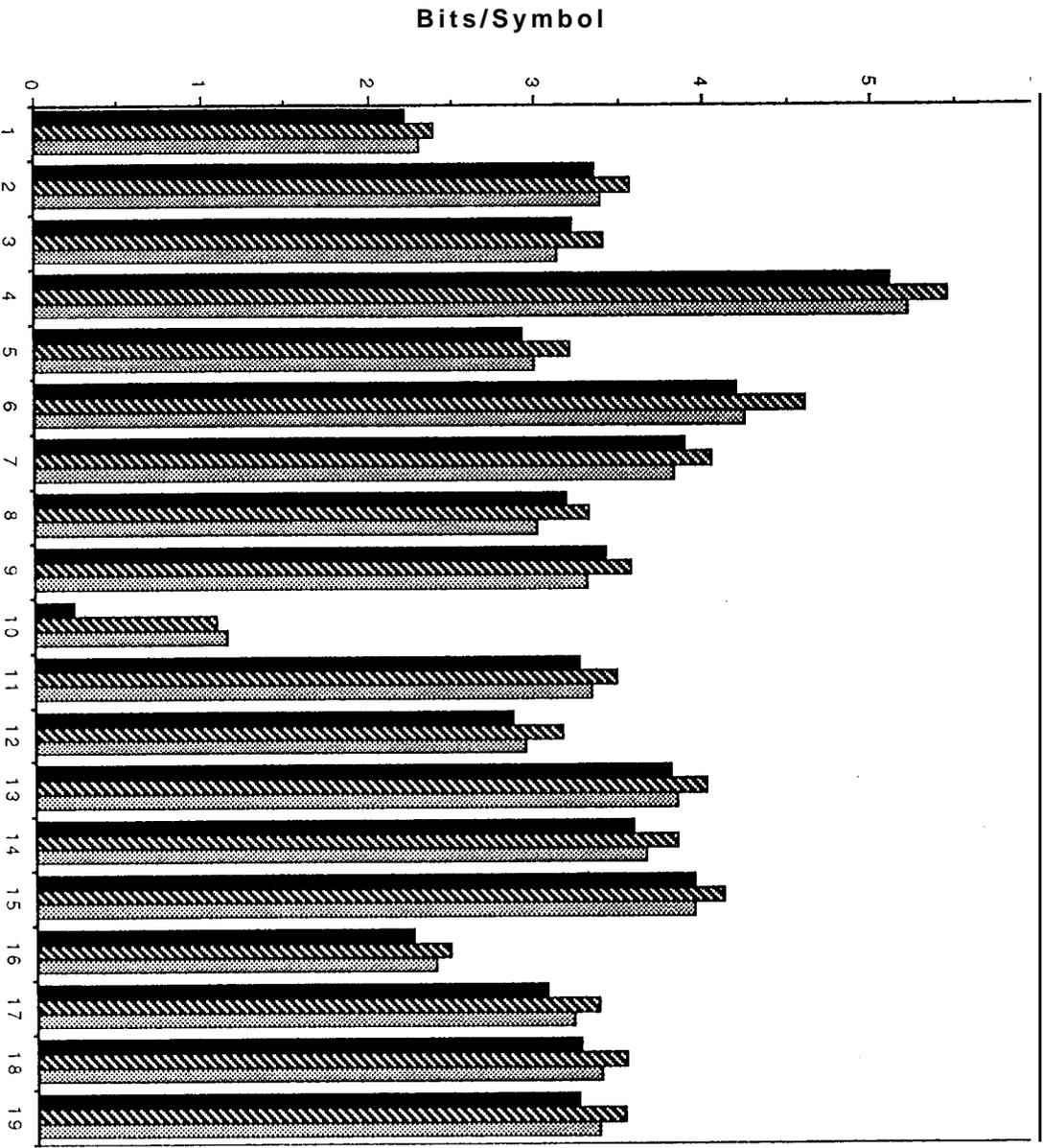


Image Number r.x

Figure 1

■ Dif Entropy #
 ▨ Best Cf
 ▩ Adaptive

Differential entropy is measured for the whole image, by taking differences between adjacent pixels and gather the statistics

Best compression ratio using an optimal codebook for the :r:le image

Start of Range $r_1(0)$	End of Range $r_2(0)$	Optimal l
1\$000000	0.296176	<i>1</i>
<i>0.296176</i>	<i>0.140251</i>	<i>2</i>
<i>0.140251</i>	0.126126	<i>3</i>
<i>0.126126</i>	<i>0.077586</i>	<i>4</i>
0.077586	0.063264	<i>5</i>
0.063264	<i>0.055966</i>	<i>7</i>
0.055966	0.041124	8
0.041124	0.036607	<i>9</i>
0.036807	0.033058	10
0.033058	0.030397	<i>11</i>
0.030397	0.027749	12
0.027749	0.026167	<i>15</i>
0.026167	0.021450	<i>16</i>
0.021450	0.019888	<i>17</i>
0.019888	0.018849	18
0.018849	0.017812	19
0.017812	0.017294	<i>20</i>
0.017294	0.016260	21
0.016260	0.015744	22
0.015744	0.015228	<i>23</i>
0.015228	0.014199	<i>24</i>
0.014199	0.013685	<i>25</i>
0.013685	0.013171	<i>26</i>
0.013171	0.012658	<i>28</i>
<i>0.012658</i>	0.012146	<i>29</i>

Table 4 Optimal l values for a double-sided geometric distribution as function of r_0 , the proportion of zero's in the difference statistics histogram'