A High-Speed Distortionless Predictive Image-Compression Scheme

K.-M. Cheung, P. Smyth, and H. Wang
Communications Systems Research Section

This article introduces a high-speed distortionless predictive image-compression scheme that is based on differential pulse code modulation output modeling combined with efficient source-code design. Experimental results show that this scheme achieves compression that is very close to the difference entropy of the source.

I. Introduction

Digital data compression is the conversion of a stream of high-rate data (digital or analog) into a stream of relatively low-rate quantized data for communication over a digital communication link or storage in a digital memory. The goals are to reduce the volume of data for transmission over a digital channel and to archive to a digital medium. As the volume of speech and image data in the foreseeable future becomes prohibitively large for many communication links or storage devices, the theory and practice of data compression are receiving increasing attention.

Many data sources contain significant redundancy—symbol distribution, pattern repetition, and positional redundancy. Data compression schemes work by removing redundancy and encoding the new information. They can be broadly categorized into two classes: *lossless* and *lossy*. A lossless coder operating on a digital data source achieves compression without losing any of the digital source information, hence, the decoded image is no different from the original one. A lossy coder reduces the data rate by sacrificing some information that is, or is thought to be, of little relevance to the user. As a result, the decoded data from a lossy coder show some coding noise, or error, when compared to the original data.

Image data are a typical example of a data source with significant redundancy. Image-compression techniques are used to reduce the volume of raw image data, which is usually too large for many communication links and storage devices. In the past 30 years, there has been a considerable amount of activity in image coding. Much of this research centers around the theme of using elegant compression algorithms to achieve a better compression ratio in the lossless compression scenario, and to achieve a better compression-distortion trade-off in the lossy compression scenario. Due to the recent advances in very large scale integration (VLSI) technology, it is now feasible to implement these algorithms in practical situations, e.g., differential pulse code modulation (DPCM) coding [1], runlength coding [2], the Rice machine [3], and arithmetic coding [4].

DPCM schemes predict the present sample value based on the previous samples and send an encoded version of the difference between the predicted and actual values.

Because of its simplicity, compression efficiency, and ease of implementation, DPCM has become quite common in compressing speech and image data for transmission and storage. In digital processing, the input sequence to the DPCM scheme is generally digitized at the source itself by a fixed number of bits (typically 8 for images). Because predictor values are required to be integer values, the DPCM output sequence will take integer values having a skewed probability distribution. The integer values can then be entropy-coded without distortion using an entropy coder such as the Huffman code [5] or the Rice algorithm [3]. A schematic diagram of a DPCM/entropy coding scheme is shown in Fig. 1.

This article introduces a high-speed distortionless predictive image-compression scheme that is based on DPCM output modeling and efficient source-code design. In general, this scheme is extremely simple to implement and, hence, has potential for high-rate, low-complexity applications. This scheme is particularly useful for spacecraft imaging data, where exact reproduction is a requirement. Other uses include medical image data, where high speed and good quality are both extremely important. The remainder of this article is organized as follows: Section II describes two probabilistic models for the DPCM output integers and Section III outlines the compression algorithm that is based on the models proposed in Section II. Section IV describes the experimental results. Section V discusses the implementation issues. Section VI discusses the entropy and redundancy analysis for this scheme, Section VII addresses the source–model mismatch, and Section VIII gives the concluding remarks.

II. DPCM Output Modeling

This section describes two probabilistic models for the output samples of a simple DPCM scheme as shown in Fig. 1. To simplify the discussion that follows, a simple predictor consisting of a single delay is chosen. However, the results derived in this article are quite general and apply to DPCM coding schemes with other predictors.

A. The Modified Laplacian Model

The Laplacian distribution of the form

$$f_L(x) = \frac{1}{\sqrt{2}\sigma} \exp^{\frac{-\sqrt{2}|x-\mu|}{\sigma}} \qquad (1)$$

where μ denotes the mean and σ denotes the standard deviation, is traditionally used to model the unquantized output samples of a DPCM scheme [6,7]. By requiring the input sequence and the predictor output samples to be integer values, the DPCM output sequence will take integer values. The probability of the DPCM output integer i, which is denoted by $p(i)$, can be evaluated by integrating from $i - 0.5$ to $i + 0.5$. In most well-behaved speech and image sources, μ is very close to zero.

To simplify the discussion that follows, it is assumed that $\mu = 0$. To verify the validity of the Laplacian model, four images with 8-bit pixels are selected, and the aforementioned simple DPCM coding scheme is applied to encode the images. For each image, the frequency distribution of the DPCM output samples is tabulated and compared to the corresponding hypothetical distribution given in Eq. (1). The comparisons are shown in Fig. 2. The results were found to be unsatisfactory. The Laplacian distribution was then modified by adjusting the standard deviation σ to $\lambda\sigma$ such that $\lambda\sigma = -1/(\sqrt{2}\log(1 - r(0))$, and $r(0)$ is the ratio of the number of zeros in the DPCM output stream to the total number of DPCM output samples. The zero-mean modified Laplacian distribution is given in Eq. (2) as

$$f_{ML}(x) = \frac{1}{\sqrt{2}\lambda\sigma} \exp^{\frac{-\sqrt{2}|x|}{\lambda\sigma}} \qquad (2)$$

The introduction of λ corresponds to adjusting the standard deviation σ of the hypothetical distribution by matching the frequency of occurrences of the most probable event in the experiment, namely the integer zero, to the frequency of occurrences of zero predicted by the hypothetical Laplacian model. The comparisons between the experimental frequency distribution and the hypothetical distribution of the images are shown in Fig. 3. The difference entropy, defined as the entropy of the differences between adjacent pixels, is derived from the DPCM output samples and the entropy calculated from the hypothetical modified Laplacian model of the images. The comparisons are shown in Table 1. This modified Laplacian model can be used (1) to simplify the performance analysis of the DPCM scheme and (2) to design an optimal scalar quantizer (e.g., a Lloyd-Max quantizer) for the DPCM output samples.

B. The Two-Sided Geometric Model

The above continuous modified Laplacian distribution can be shown to provide a good foundation for efficient source-code design. However, this article deals with discrete (integer) DPCM outputs rather than a continuous random variable. The probability distribution of DPCM output integers derived from the modified Laplacian distribution is geometric for all integers except zero. To circum-

vent these problems, a simple single-parameter discrete probability distribution for the DPCM output integers is introduced:

$$p_G(i) = \frac{1-\theta}{1+\theta}\theta^{|i|} \quad \forall i \tag{3}$$

where

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

As in the modified Laplacian model introduced in Section II.A, the frequency of occurrences of the most probable event in the experiment, namely the integer zero, matches that predicted by the hypothetical two-sided geometric model. The comparisons between the experimental frequency distribution and the hypothetical distribution of the images are shown in Fig. 4. The two distributions are almost indistinguishable. The difference entropies derived from the hypothetical two-sided geometric model of the images are shown in Table 1. The two-sided geometric model will be used in later sections to derive a simple and efficient lossless coding scheme for the DPCM output samples.

III. Efficient Coding Based on the Two-Sided Geometric Model

Constructing an optimal prefix code, by using the Huffman algorithm for example, is quite a complex operation in hardware. This section presents efficient techniques to design a near-optimal prefix code to encode images with 8 bits/pixel using the two-sided geometric model introduced in the previous section. In contrast, this code is very simple to implement in hardware. For most well-behaved images, frequency(i) $\approx$ frequency($-i$) for $i = 1, 2, \ldots, 255$. Thus, in order to construct a code for the DPCM output samples, which have values ranging from -255 to 255, one can construct a prefix code for the integer set with values from 0 to 255. An additional bit is then appended to each codeword, except the codewords representing 0, to indicate whether integer i or integer $-i$ is sent.

Gallager and van Voorhis presented an optimal binary prefix code for the set of geometrically distributed non-negative integers (single-sided geometric) [8]. This technique is adapted and modified as discussed above to encode the two-sided geometrically distributed DPCM output integers. The Gallager and van Voorhis optimal source code for geometrically distributed integer alphabets is a concatenation of a unary and a Huffman code; here it is named the Gallager–van Voorhis–Huffman (GVH) code. Let l be the integer satisfying

$$\theta^l + \theta^{l+1} \leq 1 < \theta^l + \theta^{l-1} \tag{4}$$

where

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

as defined in Section II.B. It is easy to see that for any θ, $0 < \theta < 1$, there is a unique positive integer l satisfying Eq. (4). 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 showed that an optimal code for the non-negative integers is the concatenation of a unary code that is used to encode j, and a Huffman code that 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

$$p_r = \frac{1-\theta}{1-\theta^l}\theta^r$$

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 1. It can be shown that the optimal coding for this integer set is to use codewords of length $\lfloor \log_2 l \rfloor$ for $i < 2^{\lfloor \log_2 l + 1 \rfloor} - l$, and codewords of length $\lfloor \log_2 l \rfloor + 1$ otherwise.

In this article, a simple construction is proposed 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 $\{00\ldots0, \ldots, 11\ldots1\}$, 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'.

It is not hard to see that L' has a list of l prefix-conditioned codewords, with $2^{\lfloor \log_2 l + 1 \rfloor} - l$ 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 DPCM output integer i can be efficiently encoded using a concatenation of a unary code, a Huffman code, and a sign bit (except for $i = 0$).

IV. Experimental Results

A number of images with 8-bit pixels were selected, including both planetary and nonplanetary images. The difference entropy of these images was evaluated, and the images were compressed using the Rice algorithm and the DPCM/GVH code. The difference entropy and the average numbers of bits/pixel used to encode the images using the Rice algorithm and the DPCM/GVH code are shown in Table 2. Experimental results show that in all cases the DPCM/GVH scheme achieves compression that is very close to the difference entropy of the sources, and the compression performances of the Rice algorithm and the DPCM/GVH code are almost indistinguishable.

V. Some Implementation Issues of the DPCM/GVH Scheme

Like the Rice algorithm [3], the DPCM/GVH code yields a performance (average number of bits per pixel) that approaches the difference entropy of the source. As shown in Sections III and VI, the mathematical theories required to derive this algorithm and to show its efficiency are quite elegant. Its implementation, however, is very simple. This section describes some practical implementation issues of the DPCM/GVH coding scheme.

As discussed in Section III, the decoding process requires finding a unique integer l such that $\theta^l + \theta^{l+1} \leq 1 < \theta^l + \theta^{l-1}$. The range of θ, and thus $r(0)$, that gives a certain l can be precalculated, and l can be found easily by using a table look-up on $r(0)$. The maximum value for l is arbitrary; here it is chosen to be 16. Table 3 gives the ranges of $r(0)$ for $l = 1, \ldots, 16$. Note that this table look-up on l's is optimally designed for the single-sided geometric distribution [5], and there is no guarantee that it is optimal for the two-sided geometrical distribution described in this article. A look-up table that minimizes the redundancy of the code for the two-sided geometric model can be constructed by direct search, and it is given in Table 4. This is discussed further in Section VI. Once l is established, the rest of the encoding is trivial. For each DPCM output integer i, $-255 \leq i \leq 255$, one expresses $|i| = lj + r$. Since $0 \leq |i| \leq 255$, this operation of finding j and r given $|i|$ can be implemented using a table look-up on 256 entities. Since $1 \leq l \leq 16$, there are 16 tables. The total memory size required is $256 \times 16 \times 2 = 8192$ bytes. Once j and r are found, j is encoded using a unary code (a runlength code with j zeros followed by a 1), and r, $0 \leq r \leq l-1$, is encoded using a trivial Huffman code with codewords of lengths $\lfloor \log_2 l \rfloor$ and $\lfloor \log_2 l \rfloor + 1$. The Huffman codes corresponding to different l's can be generated easily, and they may be stored in a read-only memory (ROM). For $l = 1$, $|i| \bmod l = 0$ for all $|i|$ and no Huffman code is needed. For $2 \leq l \leq 16$, the Huffman codes are stored in a ROM with only $(2 + 16) \times 15/2 = 135$ entities, each of which is at most 4 bits long. This is shown in Table 5. Finally, a 0 or 1 is appended to the concatenated codeword to indicate the sign of a nonzero i. Of course, it is obvious from Table 4 that some values of l in the range $1 \leq l \leq 16$ will never be used. The total memory size can be reduced further by omitting those entities that correspond to the unused l's.

In summary, this scheme involves first counting the number of occurrences of identical consecutive digits in a block (a line or the whole picture) of pixels. A table look-up operation is then performed to find l, which characterizes the subsequent encoding of the pixels in the block. Each pixel in the block is then encoded with one subtraction operation (difference between the present pixel value and previous pixel value) and two table look-ups.

This scheme can be modified to fit different needs. Instead of encoding the whole image using one l, one can encode one line at a time. This has the following advantages:

(1) It makes the encoding more adaptive to local statistics.

(2) It reduces the buffer size (buffer for one line instead of the whole picture).

(3) It allows on-the-fly encoding.

(4) It complies with the fixed-line rate format.

However, a potential disadvantage is that the estimate of θ is based on fewer samples and may not be as robust as when the estimate is taken over a large number of samples. As the number of samples used n is reduced, the possibility of having a mismatch between the model and the actual data is increased.

VI. The Redundancy of Gallager–van Voorhis–Huffman Codes

It has been demonstrated empirically that the GVH coding scheme yields prefix codes whose average length is very close to the difference entropy of the source, i.e., whose redundancy is quite small. The performance of the GVH codes is now analyzed in more detail, deriving closed-form analytic expressions as a function of θ for the redundancy, the mean codelength, and entropy of both single-sided and two-sided integer geometric distributions.

It is both instructive and convenient to first analyze the redundancy for a single-sided integer geometric source, i.e., a source where the letters are labeled i, $0 \leq i \leq \infty$, where

$$p(i) = (1-\theta)\theta^i$$

The infinite source is treated (rather than the truncated source where $0 \leq i \leq 255$), since the difference that results from the truncation is negligible. In Gallager and van Voorhis's original paper [8], results were stated for the entropy, mean codelength, and redundancy of the GVH code for this source. Define the entropy of the DPCM source to be $H(X_1)$ (where X_1 is the discrete random variable corresponding to the single-sided geometric source), the mean codelength of the GVH code for X_1 to be $\bar{l}_1$, and the redundancy of said code to be r_1. Furthermore, let $l_1(i)$ be the length of the ith codeword, as determined by the GVH procedure. As described in Section III,

$$l_1(i) = j_i + r_i$$

where

$$j_i = \lfloor i/l \rfloor + 1$$

and

$$r_i = \lfloor \log_2(l) \rfloor, \qquad \text{if } i \bmod l < 2^{\lfloor \log_2(l)+1 \rfloor} - l$$

$$= \lfloor \log_2(l) \rfloor + 1, \qquad \text{if } i \bmod l \geq 2^{\lfloor \log_2(l)+1 \rfloor} - l$$

where l is a function of θ as described in Section III.

Appendix A shows that the entropy of this source can be written as

$$H(X_1) = \frac{H_2(\theta)}{1-\theta}$$

where $H_2(p) = -p\log_2(p) - (1-p)\log_2(1-p)$ is the binary entropy function. It is also shown that the mean codelength[1] is

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

where k is defined as

$$k = 2^{\lfloor \log_2(l)+1 \rfloor} - l$$

[1] This result is different from that given in [8]. There appears to be a typographical error in their equation for $\bar{l}_1$, as they have the term $\lceil \log_2(l) \rceil$ instead of $\lfloor \log_2(l) \rfloor$.

Hence, the result is

$$r_1 = \bar{l}_1 - H(X_1)$$

$$= \lfloor \log_2(l) \rfloor + 1 + \frac{\theta^k}{1-\theta^l} - \frac{H_2(\theta)}{1-\theta}$$

Since this is an optimal prefix code, $r_1 \leq 1$; and indeed, for most values of θ, r_1 is quite small.

Next, the topic of relevance to DPCM image coding is addressed, namely the coding of the two-sided geometric model introduced in Section II.B, where for each letter i, $-\infty \leq i \leq \infty$, and

$$p_2(i) = \frac{1-\theta}{1+\theta}\theta^{|i|}$$

Once again, for convenience the infinite source alphabet (rather than the truncated case) is treated: the truncated case differs by additive terms on the order of θ^{255}, which is vanishingly small for all values of θ encountered in practice. It has been found that for planetary images, the difference entropy tends to be in the range of 2.0 to 4.5 bits [3], which corresponds to θ in the range from 0.3 to 0.8 bits.

The GVH code for the double-sided model is defined as described earlier, i.e., for $i \neq 0$, an extra sign bit is appended to the equivalent codeword for a single-sided source. Define $\bar{l}_2$ as the mean codelength of such a coding scheme, and r_2 as the resulting redundancy. If $l_2(i)$ is defined as the length of the codeword assigned to letter i by this scheme, then

$$l_2(i) = l_1(i) + 1, \qquad i \neq 0$$

$$= l_1(0), \qquad i = 0$$

First derive an expression for the mean codelength $\bar{l}_2$, as a function of θ and l. Note that l need not strictly be a function of θ as before, i.e., there is no guarantee that the condition described in Section III, namely that l uniquely satisfies

$$1 + \theta \leq \theta^{-l} < 1 + \frac{1}{\theta}$$

leads to an optimal code for the two-sided source. Hence, one may treat the optimal selection of l as a separate problem; the derivations to follow are quite valid for any choice of l. The question of how to find the l's that minimize r_2 will be discussed later.

$$\bar{l}_2 = \sum_{i=-\infty}^{i=\infty} p_2(i) l_2(i)$$

$$= p_2(0) l_1(0) + \sum_{i=1}^{i=\infty} p_2(i) \Big(l_1(i) + 1 \Big)$$

$$+ \sum_{i=-\infty}^{i=-1} p_2(i) \Big(l_1(|i|) + 1 \Big)$$

$$= 2 \sum_{i=0}^{\infty} p_2(i) l_1(i) + 1 - p_2(0) - p_2(0) l_2(0)$$

But by definition

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

which leads to

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

Since, in general

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

and

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

and $\bar{l}_1$ has been derived earlier, one can write

$$\bar{l}_2 = \frac{2}{1+\theta} \left(\lfloor \log_2(l) \rfloor + 1 + \frac{\theta^k}{1-\theta^l} \right)$$

$$+ 1 - \frac{1-\theta}{1+\theta} - \frac{1-\theta}{1+\theta} (1 + \lfloor \log_2(l) \rfloor)$$

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

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

It is shown in Appendix B that the entropy of the two-sided source can be written as

$$H(X_2) = \sum_{i=-\infty}^{i=\infty} p_2(i) l_2(i)$$

$$= \log_2 \left(\frac{1+\theta}{1-\theta} \right) - \frac{2\theta \log_2(\theta)}{(1+\theta)(1-\theta)}$$

where X_2 is the discrete random variable corresponding to the double-sided geometric source. Hence, one can write down a closed-form expression for the redundancy of the coding scheme as a function of θ and l, namely

$$r_2 = \bar{l}_2 - H(X_2)$$

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

$$- \log_2 \left(\frac{1+\theta}{1-\theta} \right) + \frac{2\theta \log_2(\theta)}{(1+\theta)(1-\theta)}$$

One can find the value of l that minimizes r_2 for a given θ by minimizing the terms in r_2 that depend on l, namely

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

The optimal l values (over all ranges of θ of interest) can be found by direct search. Table 4 shows the ranges of

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

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 4 and that the ranges are different from the single-sided case (for small values of l) as given by Table 3.

Using the optimal l values, one can plot r_2 as a function of entropy of the two-sided geometric source, with the results shown in Fig. 5. This characteristic of decreasing redundancy as a function of increasing source entropy is quite general and has been bounded by Smyth [9] for arbitrary distributions. The redundancy of this coding scheme yields only a fraction of a bit of inefficiency for ranges of the source entropy of practical interest. The redundancy of the Huffman code as a function of the source entropy for the two-sided geometric model is also plotted in Fig. 5 for comparison. Clearly, the GVH code is near optimal over

all ranges of the source entropy and, from a practical viewpoint, the redundancy improvement by using a Huffman code over the GVH code is negligible.

VII. Source–Model Mismatch

An obvious topic of interest for further investigation is the robustness of the coding procedure subject to deviations of the true source from the parametric model, i.e., one would like to know how the statistical variations that can be expected in practice will affect the actual number of bits sent back. The mean codelength for the model will not be exactly equal to the mean codelength transmitted for the data. Of course, as described earlier, excellent empirical performance in the actual encoding of planetary images has been observed, but a more quantitative understanding of the trade-offs involved is desirable. Gilbert [10] looked at techniques for designing codes that avoid having very long codewords and, hence, minimize worst-case performance in the case of source–model mismatch. Longo and Galasso [11] looked at very general cases for the minimum deviations required before a probability distribution is "nearer" another optimal code. However, their bounds generally depend on the smallest differences between source probabilities and, as such, are not of practical use for DPCM models (where some of the source probabilities are very small). Since neither of these approaches is quite appropriate for this problem, there is considerable room for further work. For flight project applications in particular, it will be very important to understand and quantify the robustness of this type of model-based coding prior to practical implementation.

VIII. Conclusions

The DPCM/GVH coding scheme described in this article effectively replaces the optimal Huffman coding scheme with a near-optimal look-up table operation for the special case of DPCM noiseless coding of image data. Thus, the scheme is eminently suitable for applications that require very high-rate noiseless coding and/or a very simple hardware implementation of the same. Hence, for example, future NASA flight projects such as the proposed Presidential Lunar–Mars Space Initiative[2] may find this scheme very attractive. In addition, very high-rate, ground-based data communication applications such as video conferencing and medical imaging may also be suitable areas for applying this approach.

---

[2]E. C. Posner, "Implications of the Presidential Lunar and Mars Space Initiative for the DSN," JPL IOM ECP 89-67 (internal document), Jet Propulsion Laboratory, Pasadena, California, February 1990.

References

[1] B. N. Oliver, "Efficient Coding," *Bell Syst. Tech. J.*, vol. 31, pp. 724–750, July 1952.

[2] P. Chen and P. Wintz, "Data Compression for Satellite Images," TR-EE-76-9, School of Electrical Engineering, Purdue University, West Lafayette, Indiana.

[3] R. Rice, "Some Practical Universal Noiseless Coding Techniques," Part I, JPL Publication 79-22, Jet Propulsion Laboratory, Pasadena, California, March 1979, and Part II, JPL Publication 83-17, Jet Propulsion Laboratory, Pasadena, California, March 1983.

[4] G. Langdon and J. Rissanen, "Compression of Black-White Images with Arithmetic Coding," *IEEE Trans. Communications*, COM-29, pp. 858–867, June 1981.

[5] D. A. Huffman, "A Method for the Construction of Minimum Redundancy Codes," *Proc. IRE*, vol. 40, pp. 1098–1101, 1952.

[6] A. Jain, "Image Data Compression: A Review," *Proc. IEEE*, vol. 69, no. 3, pp. 349–389, March 1981.

[7] 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, pp. 507–511, September 7–10, 1987.

[8] R. Gallager and D. van Voorhis, "Optimal Source Codes for Geometrically Distributed Integer Alphabets," *IEEE Trans. Inform. Theory*, vol. IT-21, pp. 228–230, March 1975.

[9] P. Smyth, "Entropy-Based Bounds on the Redundancy of Prefix Codes," presented at the IEEE International Symposium on Information Theory, San Diego, California, 1990.

[10] E. N. Gilbert, "Codes Based on Inaccurate Source Probabilities," *IEEE Trans. Inform. Theory*, vol. IT–17, pp. 304–314, 1971.

[11] G. Longo and G. Galasso, "An Application of Informational Divergence to Huffman Codes," *IEEE Trans. Inform. Theory*, vol. IT–28, pp. 36–43, January 1982.

Table 1. Comparison of the difference entropies of the images, modified Laplacian model, and two-sided geometric model

Images	Difference entropy	Difference entropy of modified Laplacian model	Difference entropy of two-sided geometric model
Saturn 6	2.945	2.877	3.041
Mercury	4.059	3.976	4.069
Moon	5.504	5.790	5.822
Baboon	6.348	6.387	6.409

Table 2. Comparisons of the compression performances of the Rice algorithm and the DPCM/GVH code in terms of bit/pixel

Images	Difference entropy	Rice algorithm	DPCM/GVH code
Saturn 6	2.958	3.030	3.281
USC girl	5.061	5.102	5.272
Lena	5.673	5.812	6.010
Moon	5.514	5.663	5.645
Air scene	6.153	6.268	6.174
Peppers	5.094	5.162	5.215
Baboon	6.352	6.339	6.354
Mercury	4.092	4.234	4.162

Table 3. Optimal l values for a single-sided geometric distribution as a function of r_0, the proportion of zeros in the difference-statistic histogram

Start of range $r_1(0)$	End of range $r_2(0)$	Optimal l
1.000000	0.236094	1
0.236094	0.140251	2
0.140251	0.099505	3
0.099505	0.077586	4
0.077586	0.063264	5
0.063264	0.053741	6
0.053741	0.046572	7
0.046572	0.041124	8
0.041124	0.036807	9
0.036807	0.033058	10
0.033058	0.030397	11
0.030397	0.027749	12
0.027749	0.026167	13
0.026167	0.024066	14
0.024066	0.022495	15
0.022495	0.021450	16
0.021450	0.019888	17
0.019888	0.018849	18
0.018849	0.017812	19
0.017812	0.017294	20
0.017294	0.016260	21
0.016260	0.015744	22
0.015744	0.015228	23
0.015228	0.014199	24
0.014199	0.013685	25
0.013685	0.013171	26
0.013171	0.012658	28
0.012658	0.012146	29

Table 4. Optimal l values for a double-sided geometric distribution as a function of r_0, the proportion of zeros in the difference-statistic histogram

Start of range $r_1(0)$	End of range $r_2(0)$	Optimal l
1.000000	0.296176	1
0.296176	0.140251	2
0.140251	0.126126	3
0.126126	0.077586	4
0.077586	0.063264	5
0.063264	0.055966	7
0.055966	0.041124	8
0.041124	0.036807	9
0.036807	0.033058	10
0.033058	0.030397	11
0.030397	0.027749	12
0.027749	0.026167	15
0.026167	0.021450	16
0.021450	0.019888	17
0.019888	0.018849	18
0.018849	0.017812	19
0.017812	0.017294	20
0.017294	0.016260	21
0.016260	0.015744	22
0.015744	0.015228	23
0.015228	0.014199	24
0.014199	0.013685	25
0.013685	0.013171	26
0.013171	0.012658	28
0.012658	0.012146	29

Table 5. Codeword tables for 1 = 1,...,16

Codewords	Table number	Codewords	Table number
0 1	2	00 01 100 101 110 111	6
0 10 11	3	00 010 011 100 101 110 111	7
00 01 10 11	4		
00 01 10 110 111	5	000 001 010	8

Table 5 (contd)

Codewords	Table number
011 100 101 110 111	8
000 001 010 011 100 101 110 1110 1111	9
000 001 010 011 100 101 1100 1101 1110 1111	10
000 001 010 011 100 1010 1011 1100 1101 1110 1111	11
000 001 010 011 1000 1001 1010 1011 1100 1101 1110 1111	12
000 001 010 0110	13

Codewords	Table number
0111 1000 1001 1010 1011 1100 1101 1110 1111	13
000 001 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111	14
000 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111	15
0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111	16

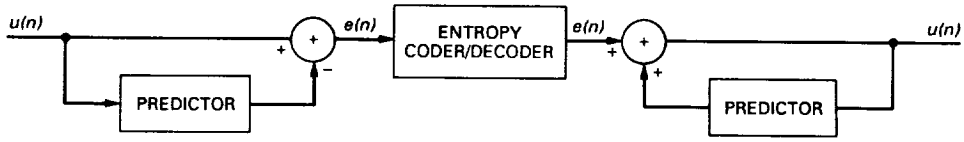

Fig. 1. A DPCM/entropy coder.

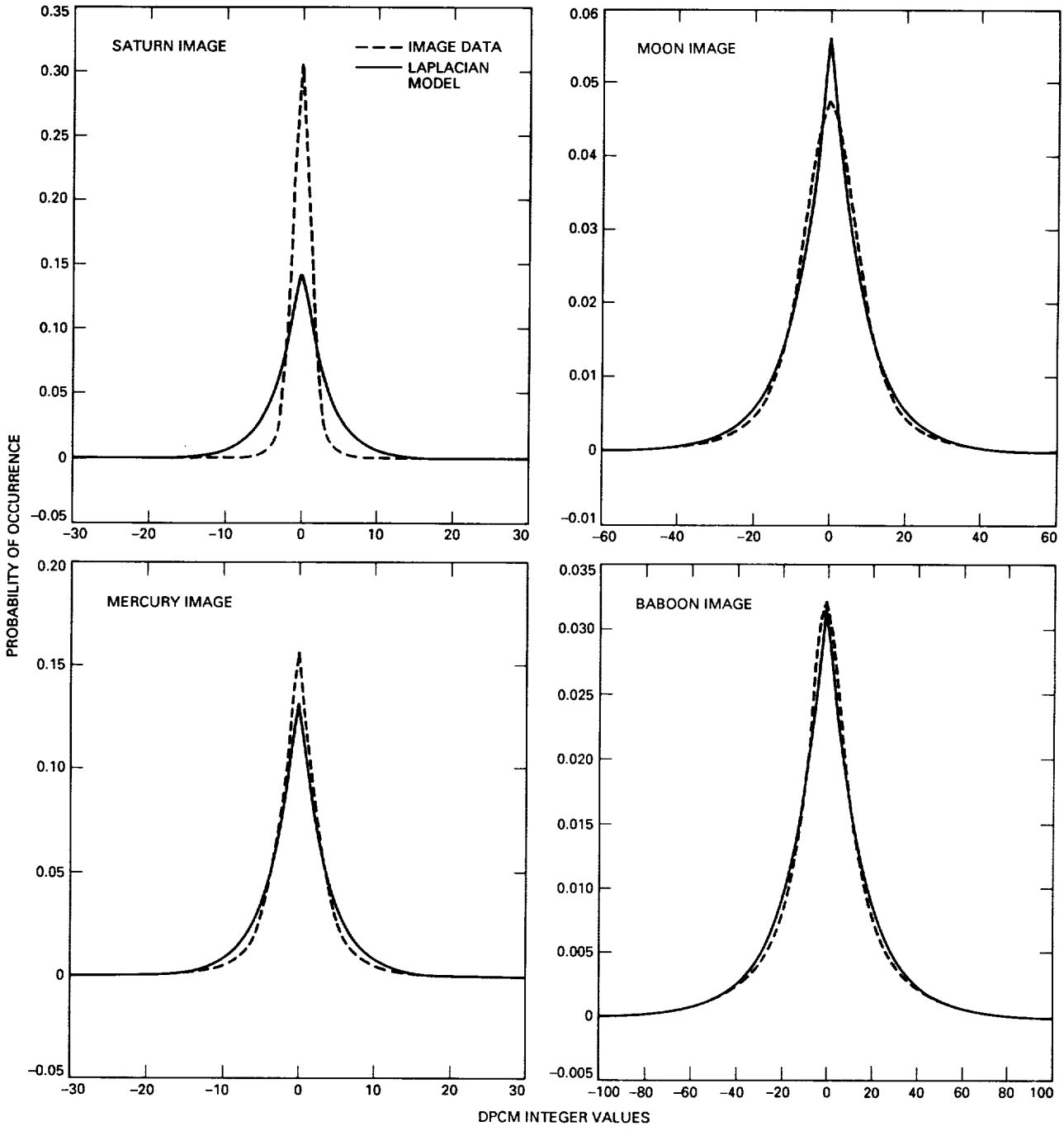

Fig. 2. Comparisons between empirical distributions and simple Laplacian distributions.

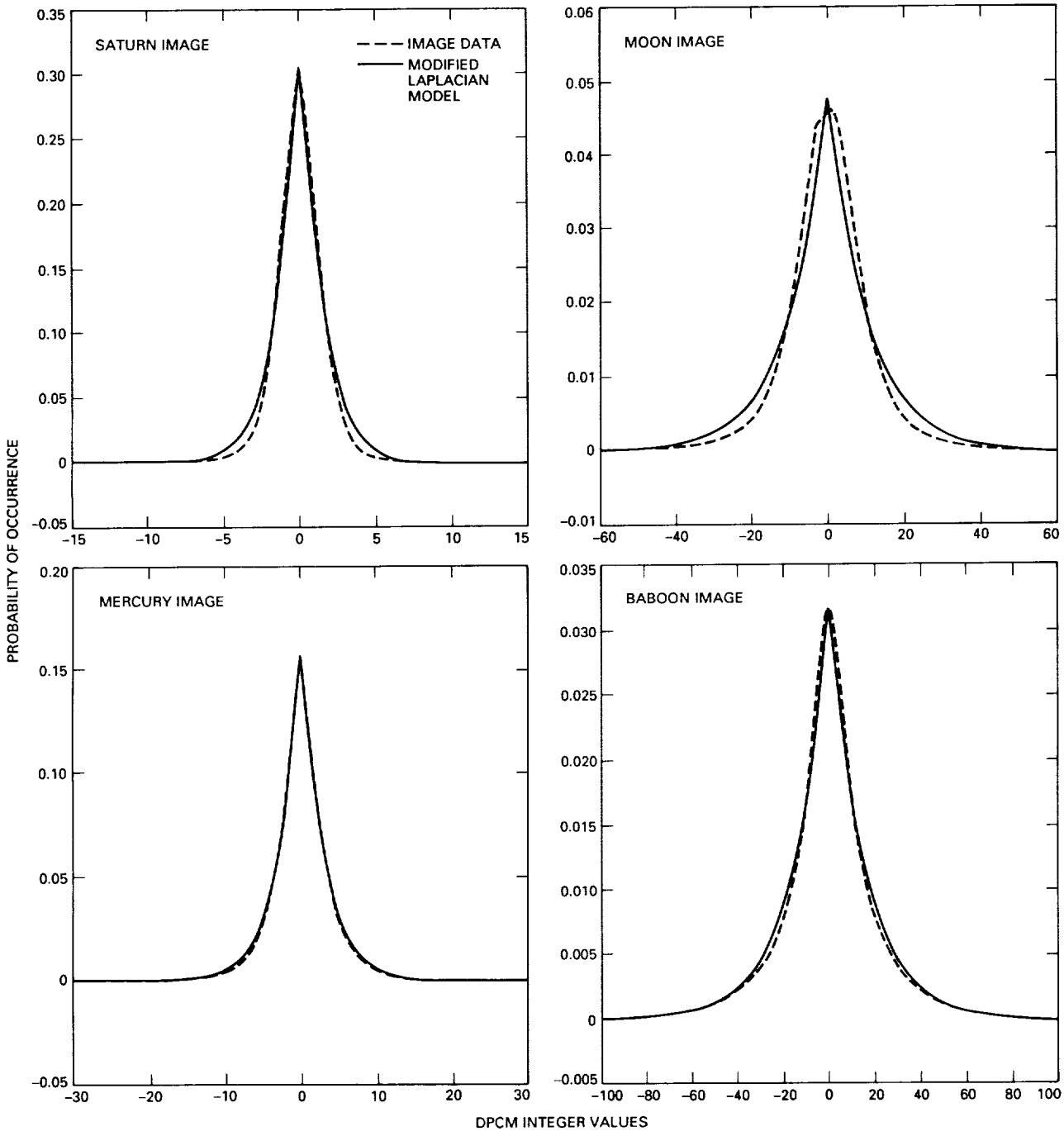

Fig. 3. Comparisons between empirical distributions and modified Laplacian distributions.

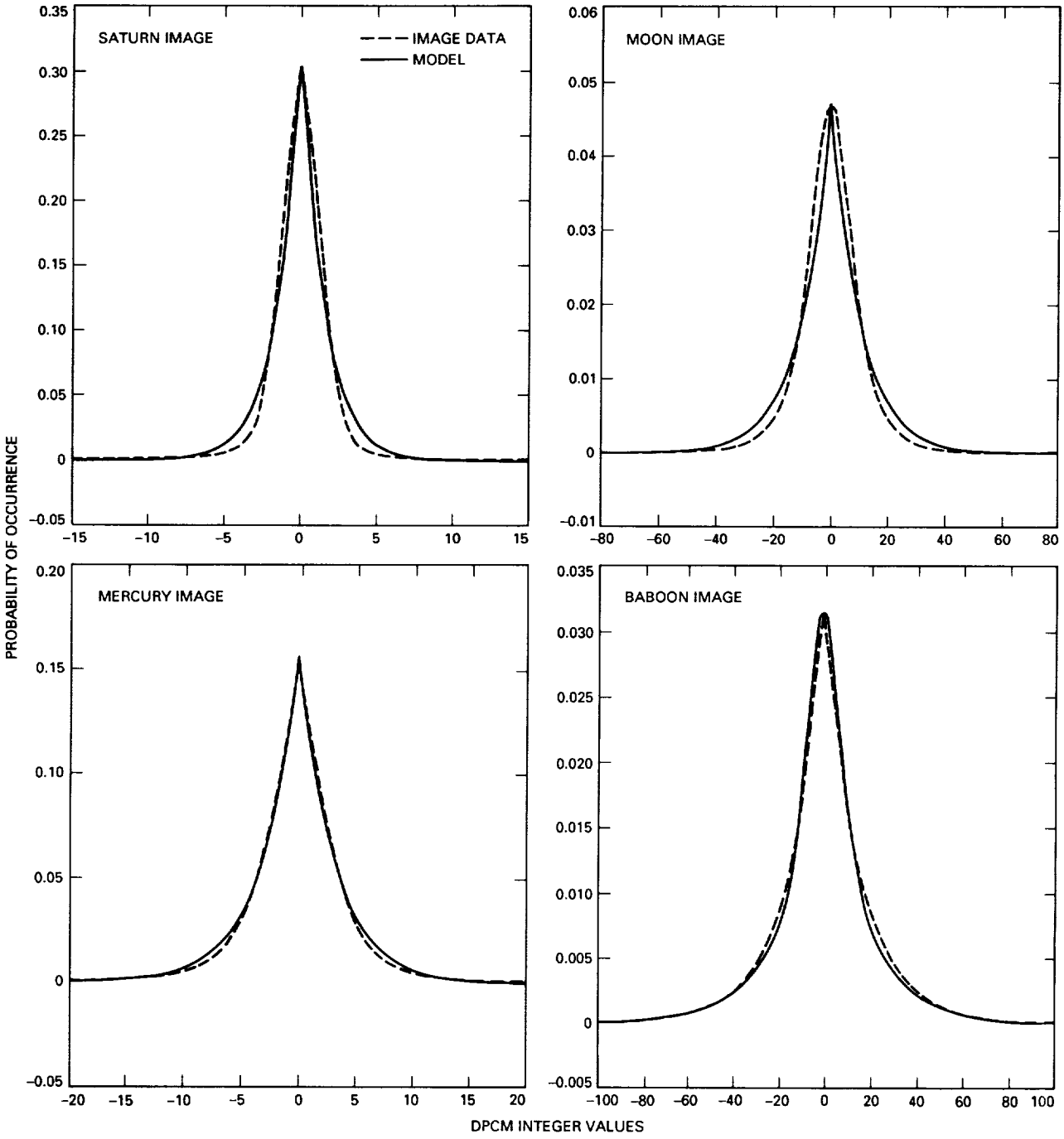

Fig. 4. Comparisons between empirical distributions and two-sided geometric distributions.

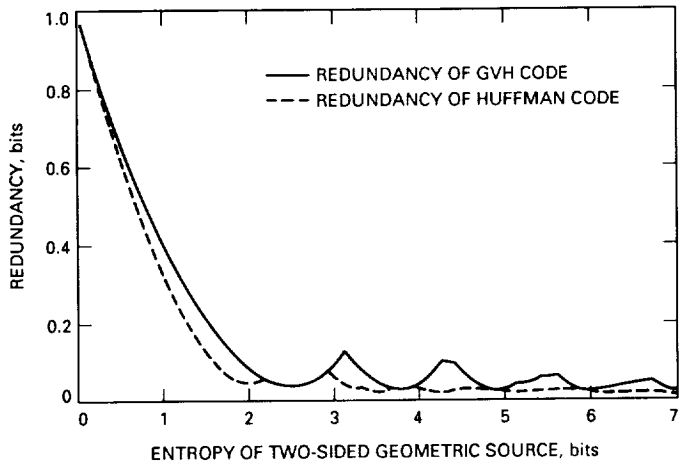

Fig. 5. Redundancy versus entropy for a two-sided geometric source.

Appendix A

Mean Codelength of GVH Code for Single-Sided Geometric Distribution

This Appendix gives an expression for the mean codelength of the GVH code for a single-sided geometric distribution as defined in Section V. One has

$$\bar{l}_1 = \sum_{i=0}^{\infty} p_i l_i$$

$$= (1-\theta)\sum_{i=0}^{i=\infty} \theta^i l_i$$

$$= (1-\theta)\left(\sum_{i=0}^{i=\infty} \theta^i j_i + \sum_{i=0}^{i=\infty} \theta^i r_i\right)$$

$$= (1-\theta)\left(\sum_{i=0}^{i=\infty} \theta^i \left(\lfloor \frac{i}{l} \rfloor + 1\right) + \sum_{i=0}^{i=\infty} \theta^i \left(\lfloor \log_2(l) \rfloor + \delta_i\right)\right)$$

where

$$\delta_i = 1 \quad \text{if } i \pmod{l} \geq k$$

$$= 0 \quad \text{otherwise}$$

and

$$k = 2^{\lfloor \log_2(l)+1 \rfloor} - l$$

and so one has

$$\bar{l}_1 = (1-\theta)\left(\sum_{i=0}^{i=\infty} \theta^i \lfloor \frac{i}{l} \rfloor + \sum_{i=0}^{i=\infty} \theta^i + \sum_{i=0}^{i=\infty} \theta^i \lfloor \log_2(l) \rfloor + \sum_{i=0}^{i=\infty} \theta^i \delta_i\right)$$

$$= S_1 + S_2 + S_3 + S_4$$

Clearly

$$S_2 + S_3 = \frac{1 + \lfloor \log_2(l) \rfloor}{1-\theta}$$

Looking at the first sum,

$$S_1 = \sum_{i=0}^{i=\infty} \theta^i \lfloor \frac{i}{l} \rfloor$$

$$= 0(\theta^0 + \theta^1 + \theta^2 + \ldots \theta^{l-1}) + 1(\theta^l + \theta^{l+1} + \theta^{l+2} + \ldots \theta^{2l-1}) + 2(\theta^2 l + \theta^{2l+1} + \theta^{2l+2} + \ldots \theta^{3l-1}) + \ldots$$

$$= \sum_{i=0}^{i=\infty} i\left(\theta^{il} \sum_{j=0}^{l-1} \theta^j\right)$$

$$= \frac{1-\theta^l}{1-\theta} \sum_{i=0}^{i=\infty} i\left(\theta^l\right)^i$$

Since

$$\sum_{i=0}^{i=\infty} i\alpha^i = \frac{\alpha}{1-\alpha^2} \qquad 0 < \alpha < 1$$

one gets

$$S_1 = \frac{1-\theta^l}{1-\theta} \frac{\theta^l}{(1-\theta)^2}$$

$$= \frac{\theta^l}{(1-\theta)(1-\theta^l)}$$

For the fourth sum, S_4, one gets

$$S_4 = \sum_{i=0}^{i=\infty} \theta^i \delta_i$$

$$= (\theta^k + \theta^{k+1} + \ldots + \theta^{l-1}) + (\theta^{k+l} + \theta^{k+l+1} + \ldots + \theta^{2l-1}) + \ldots$$

$$= \sum_{i=0}^{i=\infty} \theta^{il} \left(\sum_{j=k}^{l-1} \theta^j \right)$$

$$= \theta^k \sum_{i=0}^{i=\infty} \theta^{il} \left(\sum_{j=0}^{l-1-k} \theta^j \right)$$

$$= \theta^k \left(\frac{1-\theta^{l-k}}{1-\theta} \sum_{i=0}^{i=\infty} (\theta^l)^i \right)$$

$$= \theta^k \left(\frac{1-\theta^{l-k}}{1-\theta} \frac{1}{1-\theta^l} \right)$$

Adding $S_1 + S_4$, one gets

$$S_1 + S_4 = \frac{1}{1-\theta} \frac{\theta^k}{1-\theta^l}$$

Hence

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

where k is defined as

$$k = 2^{\lfloor \log_2(l)+1 \rfloor} - l$$

Appendix B

Entropy of a Two-Sided Geometric Source as a Function of Θ

This Appendix gives an expression for the entropy of a two-sided geometric source as a function of θ. One has

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