JPL Publication 91-2

On the Optimality of Code Options for a Universal Noiseless Coder

Pen-Shu Yeh
Goddard Space Flight Center

Robert Rice
Jet Propulsion Laboratory

Warner Miller
Goddard Space Flight Center

February 15, 1991

NASA

National Aeronautics and
Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

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

Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government, the Goddard Space Flight Center, or the Jet Propulsion Laboratory, California Institute of Technology.

ABSTRACT

Rice developed a universal noiseless coding structure that provides efficient performance over an extremely broad range of source entropy. This is accomplished by adaptively selecting the best of several easily implemented variable length coding algorithms. Variations of such noiseless coders have been used in many NASA applications. Custom VLSI coder and decoder modules capable of processing over 20 million samples per second are currently under development.

In this study, the first of the code options used in this module development is shown to be equivalent to a class of Huffman code under the Humblet condition, for source symbol sets having a Laplacian distribution. Except for the default option, other options are shown to be equivalent to the Huffman codes of a modified Laplacian symbol set, at specified symbol entropy values. Simulation results are obtained on actual aerial imagery, and they confirm the optimality of the scheme. On sources having Gaussian or Poisson distributions, coder performance is also projected through analysis and simulation.

ACKNOWLEDGEMENT

The authors wish to thank John Yagelowich at the Goddard Space Flight Center for making this report more readable and Merv MacMedan at Jet Propulsion Laboratory for his support in publishing this work.

Contents

APPENDICES

List of Figures

List of Tables

ON THE OPTIMALITY OF CODE OPTIONS FOR A UNIVERSAL NOISELESS CODER

I INTRODUCTION

Information preserving source coding, also known as noiseless data compression, has been studied for several decades. This type of coding is especially applicable to the coding of computer files and medical imaging or scientific data, when high fidelity of the data is desirable. The mostly widely known technique, undoubtedly, is the Huffman algorithm which generates variable length codes optimal for fixed known source distributions [1]. A number of papers addressing the properties or subclasses of this coder have been published [2-8].

But most real applications produce source symbol distributions which vary, so the optimality of an individual Huffman code is often insignificant because the optimized code will only exhibit efficient performance over a narrow range of data entropies. Efforts to remedy this limitation are exemplified by the dynamic Huffman code [9-11] and Rice's universal noiseless coding technique, which appeared in its early form in [12] and was generalized in [13-16]. The Rice algorithm is an easily implementable and adaptive scheme that codes data close to the source entropy and can be extended to any entropy range, as desired; it consists of multiple options, each targeted at an entropy range of approximately 1 bit/symbol. Extensions and modifications to the original algorithms have formed a basis of data compression systems for a diverse set of applications [17-20].

In this publication, we explore the intricate relation between Rice's universal noiseless coding technique and the optimal Huffman coder for rates higher than 1 bit/symbol. This relation can be readily derived for sources having a Laplacian Probability Distribution Function (PDF), which is a reasonable assumption among most imagery data, after a linear prediction is performed on adjacent pixels. For other types of PDFs the relation between the coding technique and the Huffman coder is more obvious only after re-segmenting the PDFs, as will be explained later.

A brief description of Rice's technique is given in the early part of the next section.

The relation between the technique and the Huffman coder will be explored and validated by computer simulation. Actual coding results on aerial imagery will be given, along with further investigation into the coder's applicability to other data sources having a Gaussian or a Poisson PDF.

II THE RICE ALGORITHM AND ITS PERFORMANCE ON A LAPLACIAN SYMBOL SET

II.1 Rice's Universal Noiseless Coding Technique

To relate the Rice coding technique to the optimal Huffman code, a brief description of the Rice coding scheme is given first. Interested readers are referred to [13-16] for details. The coder, depicted in Fig. 1, consists of two separate functional parts: The front-end pre-processor is a predictor followed by a symbol mapper, while the second part performs the actual adaptive symbol coding. The function of the front-end pre-processor is to decorrelate the incoming data stream by simply taking the difference between adjacent data (other higher order predictor types can be implemented as well), and also to map all difference values, positive or negative, to a sequence of non-negative integer symbols.

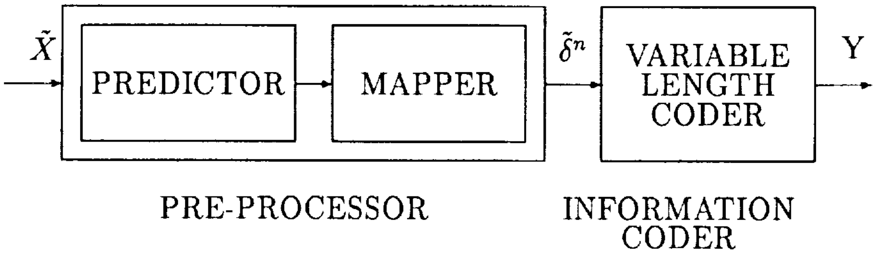
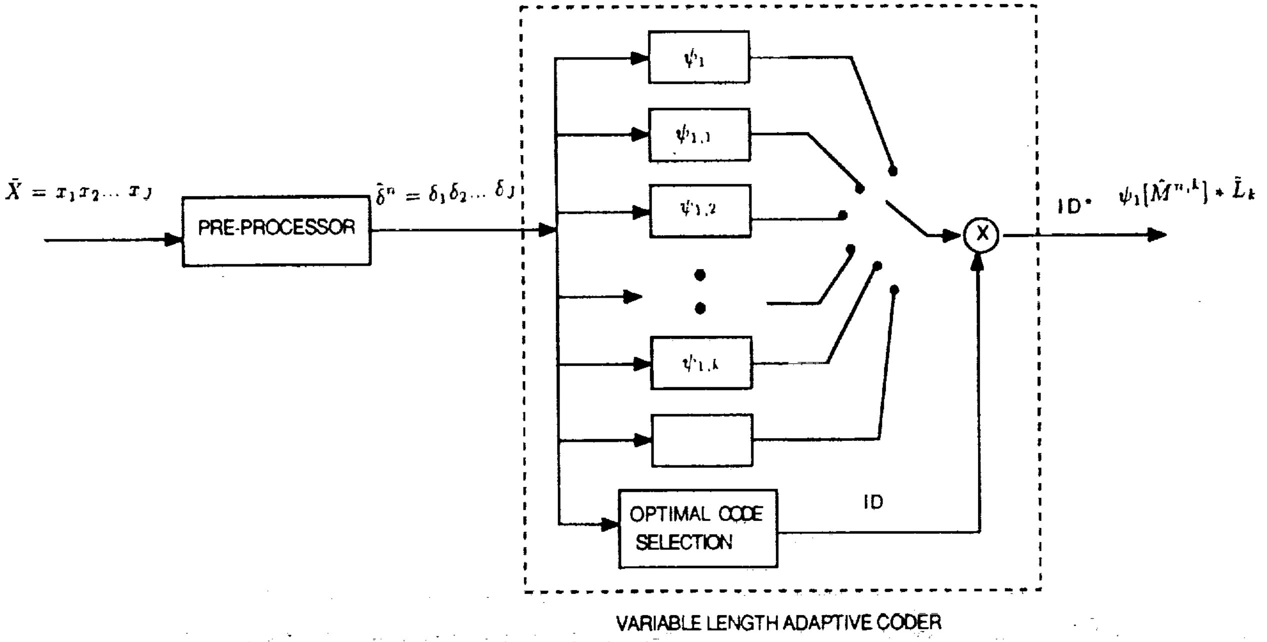


Figure 1: Lossless Source Coder

The second functional block implements a variable length coder of multiple options, each targeted for a different source entropy level. A more detailed structural diagram is given in Fig. 2. Given a block of J input samples $\tilde{X} = x_1 x_2 ... x_J$, the pre-processor outputs J non-negative integer symbols $\tilde{\delta}^n = \delta_1 \delta_2 ... \delta_J$, where n indicates n-bit quantization levels. The codeword length for the J symbols is first calculated for each option, the coder then selects the option which yields the shortest codeword. Of the multiple options, the most basic is a Fundamental Sequence (FS) code ψ_1. For a non-negative integer symbol $s_i \in S$, $S = \{0, 1, 2, ..., 2^n - 1\}$, ψ_1 outputs i 0's for this symbol and terminates with a 1,



$\bar{X} = x_1 x_2 \ldots x_J$
PRE-PROCESSOR
$\bar{\delta}^n = \delta_1 \delta_2 \ldots \delta_J$
ψ_1
$\psi_{1,1}$
$\psi_{1,2}$
$\psi_{1,k}$
X
OPTIMAL CODE SELECTION
ID
ID* $\psi_1[\hat{M}^{n,k}] * \tilde{L}_k$
VARIABLE LENGTH ADAPTIVE CODER

Figure 2: The $\psi_{1,k}$ Coder Structure

as shown in the following:

$$\psi_1(s_i) \equiv fs[i] \equiv \underbrace{0\,0\,0 \cdots 0\,0}_{i\ zeros}1 \tag{1}$$

Thus, $fs[i]$ is equivalent to a *comma code* for an ordered symbol set with the probability ordering

$$p_0 \geq p_1 \geq p_2 \geq \ldots \geq p_N, \tag{2}$$

where

$$p_i = prob[s_i]. \tag{3}$$

Such ordering combined with the comma coding on the source symbol will give shorter codeword lengths to more frequently occurring symbols. Functionally the mapper in the pre-processor tries to map the prediction error into this ordered symbol set so that $\delta_j \in S$. Coding of a complete $\tilde{\delta}^n$ block using the FS will give

$$\psi_1[\tilde{\delta}^n] = fs[\delta_1] * fs[\delta_2] * ... * fs[\delta_J], \tag{4}$$

where $*$ means concatenation.

Other options, as denoted by $\psi_{1,k}$, belong to the split-sample coding scheme. If we let

$$\tilde{M}^{n,k} = m_1 * m_2 * ... m_J \tag{5}$$

denote the J sample sequence of the most significant $n-k$ bit samples extracted from $\tilde{\delta}^n$, and

$$\tilde{L}_k = lsb_1 * lsb_2 * ... lsb_J \tag{6}$$

denote the corresponding sequence of the k least significant bit samples of $\tilde{\delta}^n$, then the coded output from $\psi_{1,k}$ will be

$$\psi_1[\tilde{M}^{n,k}] * \tilde{L}_k. \tag{7}$$

That is, the most significant $n-k$ bits of symbol δ_i will be coded using the FS, while the least significant k bits remain intact. The code-selection module shown in Fig. 2 selects the option with the shortest codeword length. A binary ID specifying the option is attached in front of the coded block for identification. By limiting the number of samples J in a block, the coder achieves adaptability to scene statistics.

II.2 FS as a Class of Huffman Code

The superior performance of the FS, with an entropy range from 1.5 to 2.5 bits/sample on test images, has triggered our interest in comparing it with the optimal Huffman code. Due to its fixed structure of 0's and 1, the FS presents itself as an ideal candidate for infinite source symbols. One might wonder, under what conditions would the performance of the FS code approach, or even be equivalent to, that of a Huffman code? Surprisingly, a generic condition can be logically contrived, and has been derived earlier by Humblet [3] and relaxed in [8]. We re-iterate the condition in Humblet's [3] work as:

Let $p(.)$ be a probability measure on the set of non-negative integers. If there is a non-negative integer m such that for all $j > m$ and $i < j$,

$$p(i) \geq p(j) \tag{8}$$

and

$$p(i) \geq \sum_{n=j+1}^{\infty} p(n), \tag{9}$$

then a binary prefix condition code with minimum average codeword length for the alphabet consisting of the nonnegative integers with the above probabilities is obtained by the following procedure. Consider the reduced alphabet with letters $0, 1, \cdots, m+1$ *whose probabilities are*

$$p_1(i) = p(i), \qquad i \leq m \tag{10}$$

$$p_1(m+1) = 1 - \sum_{i=0}^{m} p(i). \tag{11}$$

Then one can apply Huffman's coding procedure to this reduced alphabet. Denote by $C_1(i)$ *and* $l_1(i)$*, respectively, the codeword and codeword length for letter* i *(* $C_1(i)$ *is a sequence of* $l_1(i)$ *binary symbols),* $0 \leq i \leq m+1$*. From there, construct the codewords* $C(i)$ *for the orignal alphabet by*

$$C(i) = C_1(i) \qquad i \leq m \tag{12}$$

$$C(i) = \{C_1(m+1), (i-m-1)^{*}0, 1\}, \quad i > m \tag{13}$$

where $n^{*}0$ *denotes a sequence of* n *zeroes.*

Both Rice and Humblet consider coding an ordered nonnegative integer symbol set, as Eqs. (2) and (8) reveal. With the Humblet condition in Eq. (9), the procedure in Eqs. (10)-(13) constructs a Huffman code $C(i)$ on this symbol set, which would be equivalent to Rice's ψ_1 code in Eq. (1).

II.3 FS Coding on a Laplacian PDF

How well would the FS coding perform on actual data? It depends on whether the data under test satisfy the Humblet condition. We choose to investigate this issue on a set of imagery data. Most imagery data are known to have high correlation between adjacent pixels. This correlation presents itself as redundant information. Therefore, in data compression studies, the correlation is usually dealt with by pre-processing the data using a prediction scheme. For a large percentage of image data, the statistics of samples after this pre-processing resembles the Laplacian function. A set of typical aerial imagery is shown in Fig. 3. A typical histogram, shown in Fig. 4, is obtained by taking horizontal differences

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

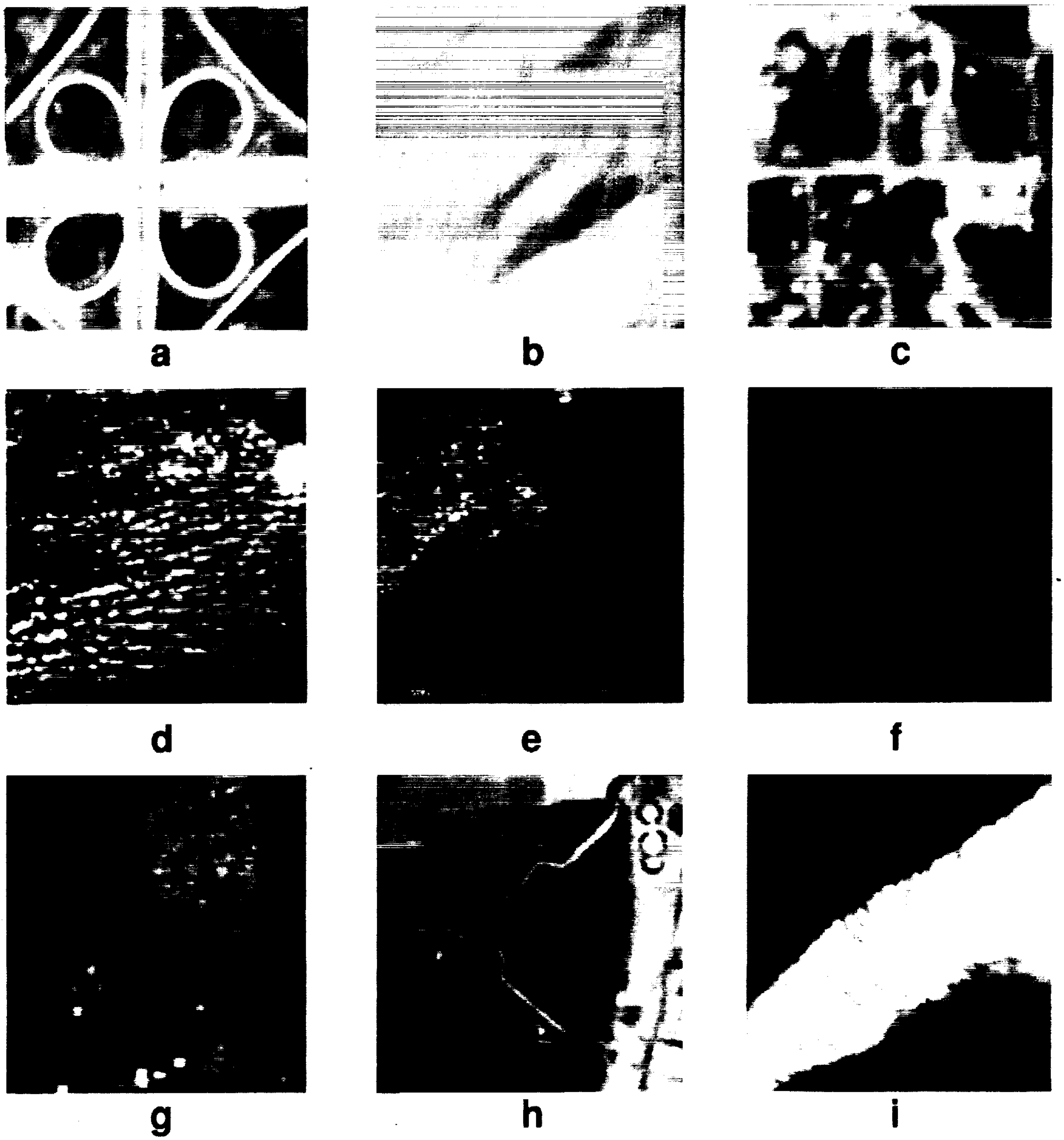


Figure 3: Test Images: Top Row is 8-bit, 3m Resolution, Middle Row is 8-bit, 1km Resolution, Bottom Row is 12-bit, 20m Resolution

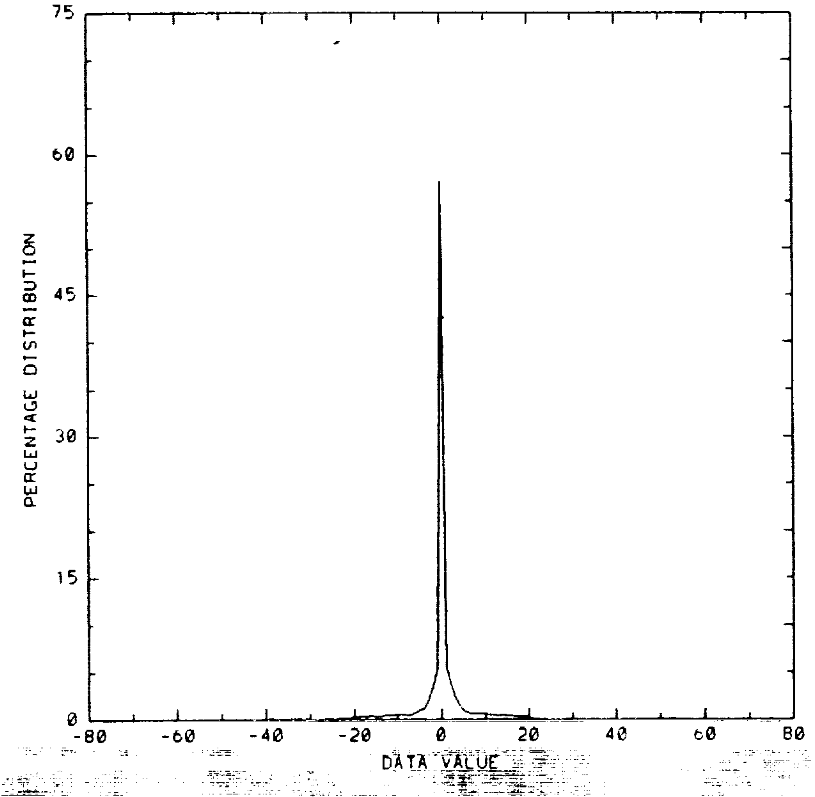


Figure 4: Histogram of the Pixel Differences of Fig. 3(a)

of pixel values on the sub-image in Fig. 3(a). The histogram peaks at value 0 and decreases steeply on both sides. The difference histogram for other images in Fig. 3 resembles Fig. 4, as well. It can be seen that these histograms can be reasonably approximated by a Laplacian function, of the form:

$$p(x) = \frac{a}{2}e^{-ax} \tag{14}$$

where $a^2\alpha^2 = 2$, with α as the standard deviation of the function.

Figure 5 shows a mapped conditional histogram generated from pixel difference values predicated on a fixed value of a predictor (i.e., the previous pixel value), in this case predictor value of 122 from Fig. 3(a). Recall that the mapping procedure performed on the pixel differences produces a sequence of symbols $\delta_1, \cdots, \delta_J$ which are represented by non-negative integers.

When modeling the mapped pixel difference values (so that all symbols are non-negative integers) as a Laplacian PDF, a slight modification is necessary to represent probability for

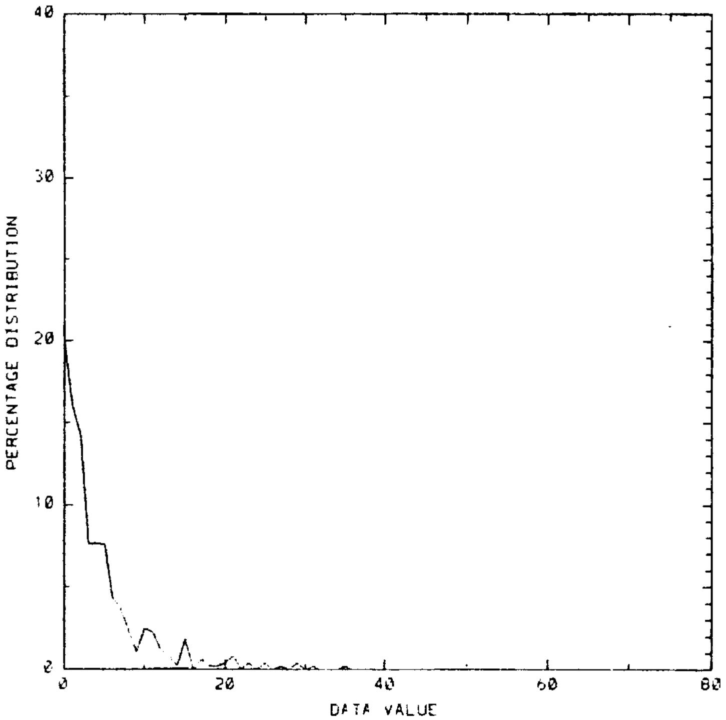


Figure 5: Mapped Conditional Histogram of Pixel Difference from Fig. 3(a) with a Pixel Reference of Value 122

non-negative integers:

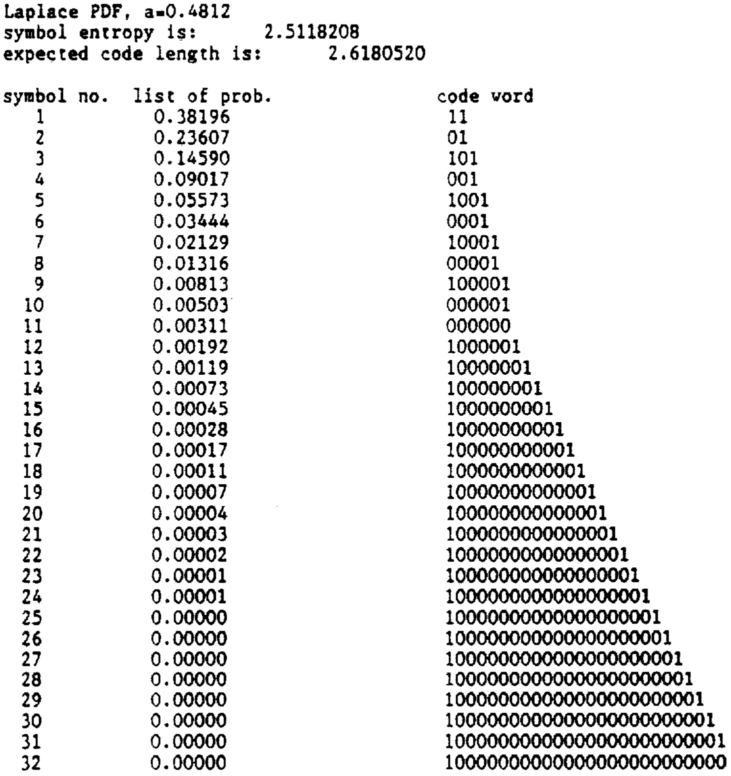
$$p(i) = \frac{a}{A} e^{-ai}, \qquad 0 \leq i \leq N \tag{15}$$

where A, a normalization factor, is given as

$$\begin{aligned} A &= \sum_{i=0}^{N} a e^{-ai} \\ &= \frac{a(1 - e^{-(N+1)a})}{1 - e^{-a}}. \end{aligned} \tag{16}$$

The parameter a of the Laplacian PDF determines the spread of the distribution and affects the entropy measurement of a source symbol set characterized by the PDF. Its value thus determines whether the FS coding of such source symbols will be an optimal Huffman code. One can derive the lower bound of a, above which the Humblet condition holds. Such a bound exists as (see Appendix A for details):

$$\begin{aligned} a &> \frac{\log(\frac{1+\sqrt{5}}{2})}{\log e} \\ &\approx 0.4812118. \end{aligned} \tag{17}$$



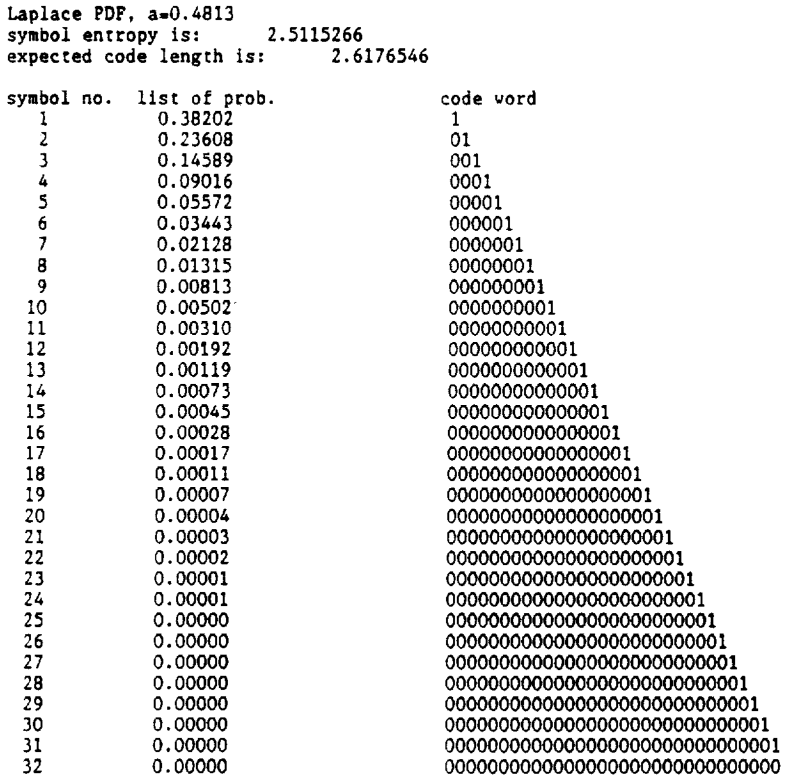
```
Laplace PDF, a=0.4812
symbol entropy is:      2.5118208
expected code length is:      2.6180520

symbol no.  list of prob.               code word
   1          0.38196                    11
   2          0.23607                    01
   3          0.14590                    101
   4          0.09017                    001
   5          0.05573                    1001
   6          0.03444                    0001
   7          0.02129                    10001
   8          0.01316                    00001
   9          0.00813                    100001
  10          0.00503                    000001
  11          0.00311                    000000
  12          0.00192                    1000001
  13          0.00119                    10000001
  14          0.00073                    100000001
  15          0.00045                    1000000001
  16          0.00028                    10000000001
  17          0.00017                    100000000001
  18          0.00011                    1000000000001
  19          0.00007                    10000000000001
  20          0.00004                    100000000000001
  21          0.00003                    1000000000000001
  22          0.00002                    10000000000000001
  23          0.00001                    100000000000000001
  24          0.00001                    1000000000000000001
  25          0.00000                    10000000000000000001
  26          0.00000                    100000000000000000001
  27          0.00000                    1000000000000000000001
  28          0.00000                    10000000000000000000001
  29          0.00000                    100000000000000000000001
  30          0.00000                    1000000000000000000000001
  31          0.00000                    10000000000000000000000001
  32          0.00000                    10000000000000000000000000
```

Figure 6: (a). Huffman Code for a Laplacian PDF at $a = 0.4812$

The condition holds when $a(N-1) \gg 0$. The derived constraint on a can easily be verified through the actual coding of a Laplacian PDF with the Huffman code, as illustrated in Fig. 6. In Fig. 6(a), a Huffman code was generated with $a = 0.4812$; it is obviously not an FS code, but in Fig. 6(b) with $a = 0.4813$, the same procedure generated the Huffman code of a totally different structure. The transition into the highly structured FS code is apparent when the above condition on a is satisfied. In both cases, the Laplacian PDF is only printed to 5 decimal places in Fig. 6. As the value of a decreases, resulting in higher symbol entropy, FS coding for the Laplacian symbol set gives performance much less desirable, as the curve marked $k = 0$ in Fig. 9 reveals.

II.4 Split-Sample Coding and Re-segmentation of the PDF

Rice [13] observes that, for source data of entropy more than a few bits/sample, the least significant bits are more randomly distributed than the higher order bits. He then proposes the $\psi_{1,k}$ coding scheme to code only the higher order bits with the FS, and append the lower



```
Laplace PDF, a=0.4813
symbol entropy is:      2.5115266
expected code length is:      2.6176546

symbol no.  list of prob.              code word
    1          0.38202                  1
    2          0.23608                  01
    3          0.14589                  001
    4          0.09016                  0001
    5          0.05572                  00001
    6          0.03443                  000001
    7          0.02128                  0000001
    8          0.01315                  00000001
    9          0.00813                  000000001
   10          0.00502                  0000000001
   11          0.00310                  00000000001
   12          0.00192                  000000000001
   13          0.00119                  0000000000001
   14          0.00073                  00000000000001
   15          0.00045                  000000000000001
   16          0.00028                  0000000000000001
   17          0.00017                  00000000000000001
   18          0.00011                  000000000000000001
   19          0.00007                  0000000000000000001
   20          0.00004                  00000000000000000001
   21          0.00003                  000000000000000000001
   22          0.00002                  0000000000000000000001
   23          0.00001                  00000000000000000000001
   24          0.00001                  000000000000000000000001
   25          0.00000                  0000000000000000000000001
   26          0.00000                  00000000000000000000000001
   27          0.00000                  000000000000000000000000001
   28          0.00000                  0000000000000000000000000001
   29          0.00000                  00000000000000000000000000001
   30          0.00000                  000000000000000000000000000001
   31          0.00000                  0000000000000000000000000000001
   32          0.00000                  0000000000000000000000000000000
```

Figure 6: (b). Huffman Code for a Laplacian PDF at $a = 0.4813$

bits uncoded. This scheme is, in fact, a procedure which re-segments the PDF of the source symbols into fewer regions, each representing a new symbol. With k split bits, we effectively have 2^{n-k} new symbols in this reduced symbol set. To illustrate the effect of segmenting a PDF into fewer regions, the $k = 0$ curve which represents the original Laplacian PDF for 256 symbols in Fig. 7 is re-segmented into 128, 64, 32, 16 and 8 symbols, as equivalent to splitting off 1, 2, 3, 4 and 5 least significant bits, referred to as the k bits in the figure. Narrower curves representing fewer symbols are obtained by integrating the original PDF over every 2, 4, 8, 16 or 32 original symbols. The newly generated PDF becomes more steep and less smooth, as revealed in Fig. 7. The PDF for the reduced symbol set can be derived mathematically by summing over 2^k original Laplacian probability values $p(i)$'s as:

$$\begin{aligned} p'_k(j) &= \sum_{m=0}^{2^k-1} p(i = 2^k \cdot j + m), \qquad j \in (0, \frac{N+1}{2^k} - 1) \\ &= \frac{a}{A} \cdot \frac{e^{-a2^kj}(1 - e^{-a2^k})}{1 - e^{-a}} \end{aligned} \tag{18}$$

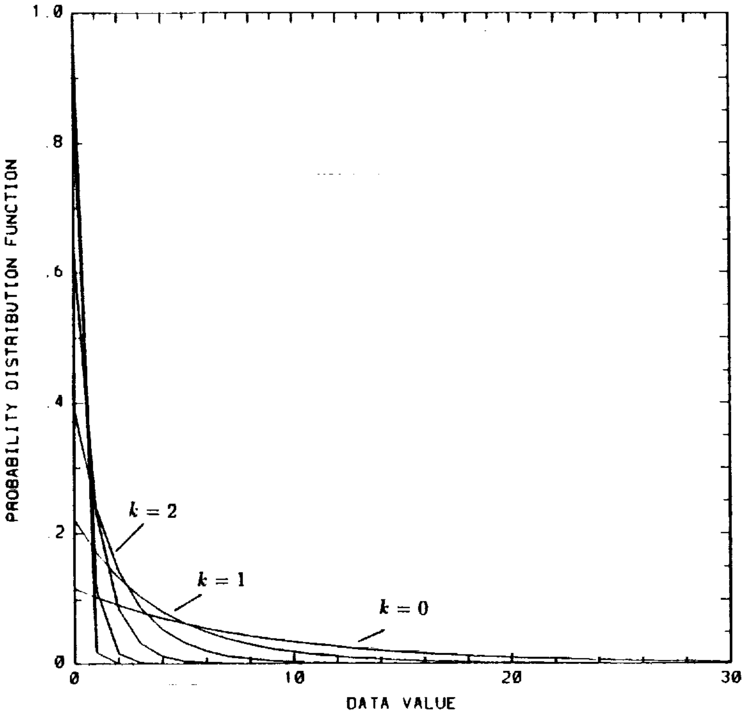


Figure 7: Re-segmented Laplacian PDF at Various Numbers of Split Bits k, from 0 to 5

The Humblet criterion for the reduced symbol set is

$$p_k'(j) \geq \sum_{m=j+2}^{N'} p_k'(m), \qquad N' = \frac{N+1}{2^k} - 1 \tag{19}$$

where for convenience, we have assumed that $N+1$ is of 2's power. Eq. (19) is satisfied when (see Appendix B)

$$2^k \cdot a \geq 0.4812118. \tag{20}$$

Note the great similarity between this value and that of Equation (17). Equation (20) immediately reduces to (17) when $k = 0$. As a decreases, resulting in a Laplacian PDF of wider spread, coding the source symbol by FS no longer guarantees an optimal Huffman code. However, a reduced symbol set, obtained by representing only the higher order bits of the binary symbol representation, may still satisfy the Humblet condition as (20) predicts. The codes resulting from this type of split-sample coding scheme may not equal the Huffman code of the original symbol set. This subject will be explored after we first provide some performance measure on the coding scheme.

II.5 Performance of a Split-Sample Coder on a Laplacian Symbol Set

To compare the performance of a split-sample coder on symbols of Laplacian PDF with an optimal Huffman code, one calculates three quantities: the source symbol entropy, the average codeword length of the $\psi_{1,k}$ coder, and the average codeword length of a Huffman code.

II.5.1 Symbol Entropy

Given the Laplacian PDF in (15), the symbol entropy is derived as:

$$\begin{aligned} H_N[p(i)] &= -\sum_0^N p(i)\log_2 p(i) \\ &= -\log_2\frac{a}{A} + \frac{a^2}{A}\log_2 e \cdot \left[\frac{e^{-a}(1-e^{-aN})}{(1-e^{-a})^2} - \frac{Ne^{-a(N+1)}}{1-e^{-a}}\right]. \end{aligned} \tag{21}$$

Similarly, with the re-segmented PDF in (18), the symbol entropy of the reduced symbol set can be derived as: (details are given in Appendix C)

$$\begin{aligned} H_{N'}[p'_k(j)] &= -\sum_0^{N'} p'_k(j)\log_2 p'_k(j) \\ &= -\log_2\frac{a}{A} - \log_2\frac{1-e^{-a2^k}}{1-e^{-a}} + \frac{a^2}{A}\frac{1-e^{-a2^k}}{1-e^{-a}}2^k\log_2 e \cdot B, \end{aligned} \tag{22}$$

where

$$B = \frac{e^{-a2^k}(1-e^{-a(N+1)+a2^k})}{(1-e^{-a2^k})^2} - \frac{(\frac{N+1}{2^k}-1)e^{-a(N+1)}}{1-e^{-a2^k}}. \tag{23}$$

Again, one can verify that when $k=0$, Eq. (22) reduces to Eq. (21).

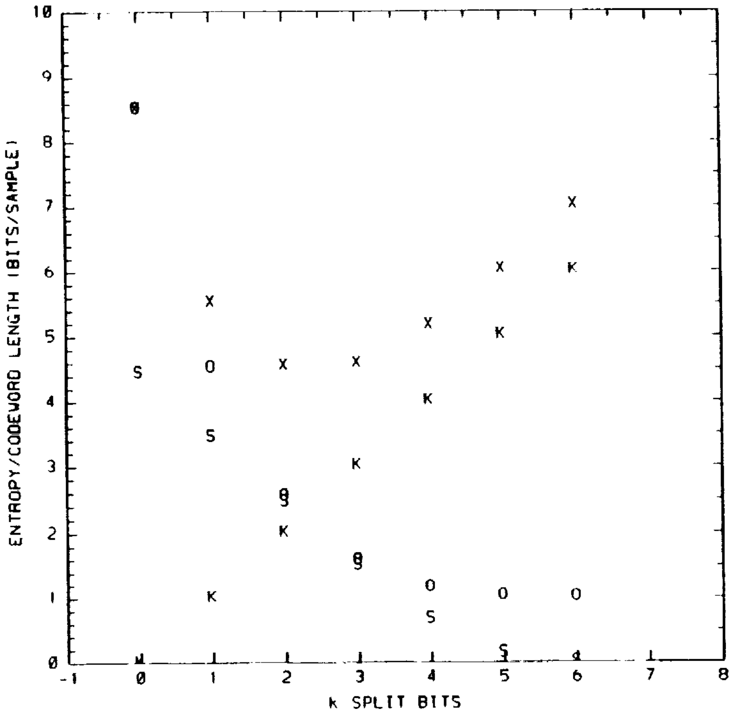


Figure 8: Effect of Sample-Splitting on Symbol Entropy and Codeword Length. S: Symbol Entropy, K: Split Bits, O: FS Codeword Length, X: Split-Sample Codeword Length

II.5.2 Average Codeword Length for Split-Sample Coding

The expected codeword length for split-sample coding consists of two terms: the FS coding length of $(j+1)$ bits for the $(j+1)$th symbol in the distribution, and k bits due to uncoded bits. This can be written as:

$$\begin{aligned} E[p'_k(j)] &= \sum_{j=0}^{\frac{N+1}{2^k}-1} (j+1)\cdot p'_k(j) + k \qquad \text{for} \quad j \in (0, \frac{N+1}{2^k} - 1) \\ &= 1 + k + \frac{1}{1-e^{-\beta(N'+1)}}[\frac{e^{-\beta}(1-e^{-\beta N'})}{1-e^{-\beta}} - N'e^{-\beta(N'+1)}], \end{aligned} \tag{24}$$

where

$$\beta = a2^k \qquad \text{and} \quad N' = \frac{N+1}{2^k} - 1.$$

In Fig. 8, the effect of splitting samples is plotted at several split-bit k values. The symbol entropy value, marked as 'S', decreases as more bits are split off due to the narrowing of the re-segmented PDF. Coding using only FS on the re-segmented Laplacian PDF

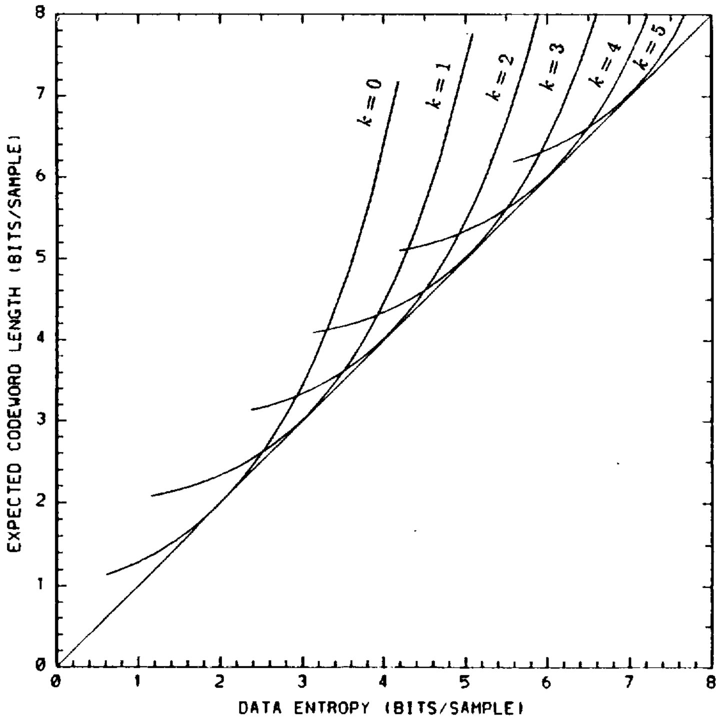


Figure 9: Coder Performance of Split-Sample Coding Technique on a Laplacian PDF

is seen to generate a shorter average codeword, marked as 'O' in the graph. The PDF becomes narrower as k increases, until it reaches a plateau, while the symbol entropy keeps decreasing. The net effect, marked as 'X' in the graph, is a concave curve with a smooth valley. The graph indicates that the optimal choice of k can have more than one value. Here both $k = 2$ and $k = 3$ yield similar coding performance.

The optimal coverage of split-sample coding at various k values is summarized in Fig. 9. Each curve corresponds to a fixed k value. Obviously, each is optimal in an entropy range of approximately 1.0 bit/sample. Thus, an adaptive coder with these options can cover a wide entropy range by varying k. As entropy traverses from the high end of 7.5 bits/sample to the low end, the coder will first select the $k = 5$ curve until at about 6.5 bits/sample, where $k = 4$ will dominate the performance. Figure 9 shows that coder performance shifts from one crossover point defined by two adjacent k values to another, achieving almost the ideal performance defined by the diagonal line. It should be noted that the 'X' curve in Fig. 8 is a vertical cross section of the various k curve data at symbol entropy of 4 bits/sample

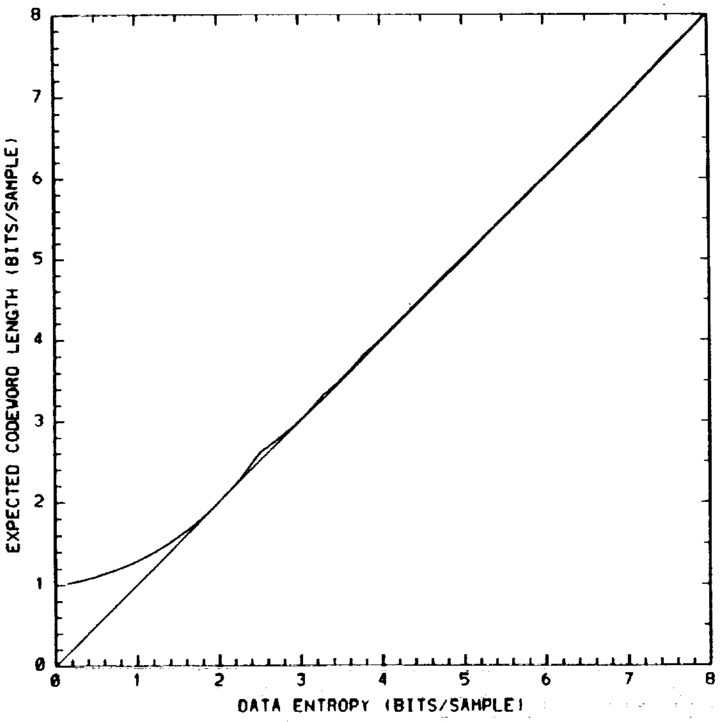


Figure 10: Huffman Coder Performance on a Laplacian PDF

from Fig. 9.

II.5.3 Average Codeword Length for Huffman Codes

A Huffman codebook is only optimal for given data source statistics. With each different value of a for the Laplacian PDF, a new Huffman code has to be generated. The performance of **a set of 50 Huffman codebooks** over an entropy range is shown in Fig. 10 for Laplacian PDFs of varying parameter a. Comparing Fig. 9 with Fig. 10, one easily recognizes the effectiveness of the split-sample coding scheme. Obviously, the major advantage of the split-sample coding technique is its orderly code structure, which greatly simplifies both coding and decoding procedures and reduces hardware design complexity.

III RELATION BETWEEN THE SPLIT-SAMPLE CODE AND THE HUFFMAN CODE

It is evident from Figs. 9 and 10 that the performance of the split-sample coder approaches that of the Huffman code for a Laplacian symbol set. The relation between the split-sample $\psi_{1,k}$ code and a Huffman code can be established by examining the binary code tree structure of both.

III.1 The Binary Code Tree Structure of $\psi_{1,k}$

In split-sample coding of a symbol, one has the option of attaching the uncoded k split bits either to the front or the rear of the coded bits for the $n-k$ most significant bits. Both schemes result in the same performance and will not cause difficulty in decoding. However, they represent totally different binary code trees, one of which is an impossible Huffman tree structure for a Laplacian symbol set.

III.1.1 The Impossible Huffman Tree Structure for $\psi_{1,k}$

The analysis is best understood by an example, such as the $\psi_{1,1}$. The Laplacian symbol set S has elements $\{s_0, s_1, ..., s_N\}$ representing integer symbols 0, 1, 2, ... , N. The $\psi_{1,1}$ binary code tree for a 16-symbol set, with a single least significant bit attached to the end of the FS code of the 3 most significant bits will have the structure shown in Fig. 11. In order for this tree structure to also represent a Huffman code tree, every pair of symbols must first be grouped to form a level-1 parent node. Then the Humblet condition must exist at this level to ensure that the FS construct for the symbols exists at this level.

Let the probability at node level $k-1$, p'_{k-1}, be written from Eq. (18) as

$$\begin{aligned} p'_{k-1}(j) &= \sum_{m=0}^{2^{k-1}-1} p(i = 2^{k-1}\cdot j + m), \qquad j \in (0, \frac{N+1}{2^{k-1}} - 1) \\ &= \frac{a}{A}\cdot\frac{e^{-a2^{k-1}j}(1-e^{-a2^{k-1}})}{1-e^{-a}}. \end{aligned} \tag{25}$$

Grouping of all adjacent pairs of symbols in Fig. 11 before applying the Humblet condition at level 1 to construct a Huffman code tree requires that the parent node probability be larger than the probability of the first symbol at level 0, the leaf nodes. This implies

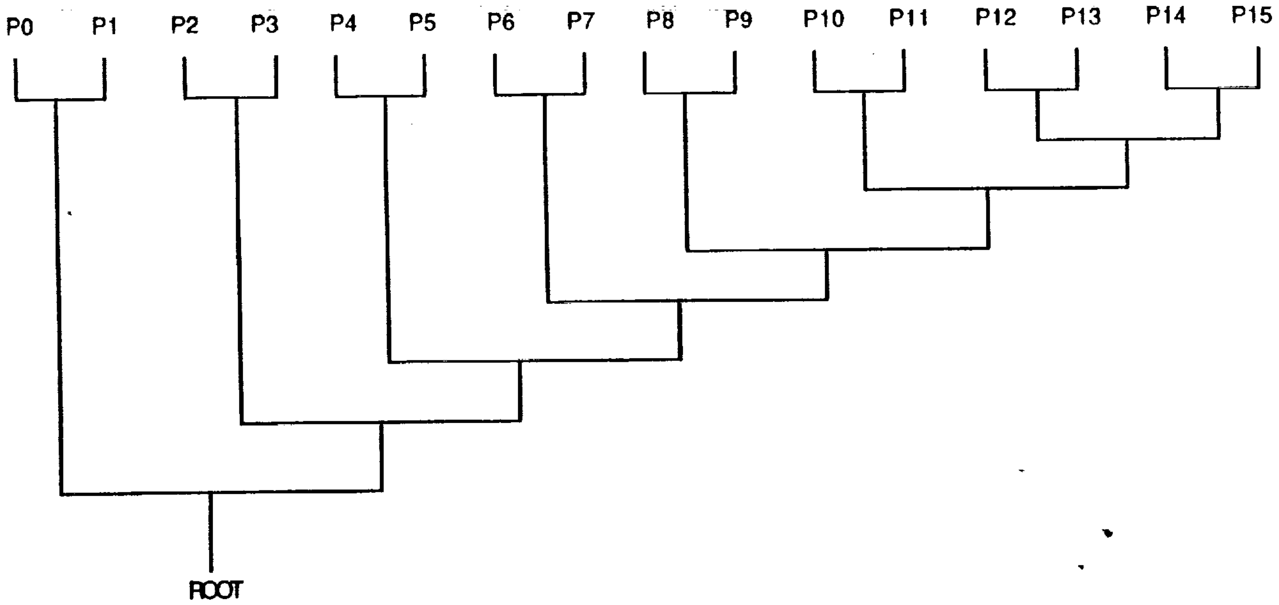


Figure 11: The $\psi_{1,1}$ Binary Tree for a 16-Symbol Set, with Split Bits Appended at the End of the FS Code

that we must have

$$p'_{k-1}(j) \; + \; p'_{k-1}(j+1) > \; p'_{k-1}(0). \tag{26}$$

Or, equivalently,

$$e^{-aj2^{k-1}}(1+e^{-a2^{k-1}}) > 1. \tag{27}$$

Equation (27) is violated when $j \geq 2$ at $k = 1$. Therefore, the tree structure shown in Fig. 11 is not a Huffman code tree for $\psi_{1,1}$.

III.1.2 A Possible Huffman Tree Structure for $\psi_{1,k}$

Can a Huffman tree structure be represented by the $\psi_{1,k}$ code construct? If one starts with the simplest case of $k = 1$, and works from the root of a Huffman code tree while trying intentionally to reach the $\psi_{1,1}$ construct, one arrives at the tree in Fig. 12 for a 16-symbol set. The binary tree produces codewords at leaf nodes exactly the same as $\psi_{1,1}$ codes for non-negative integers except for two observations. First, the ordering of the two

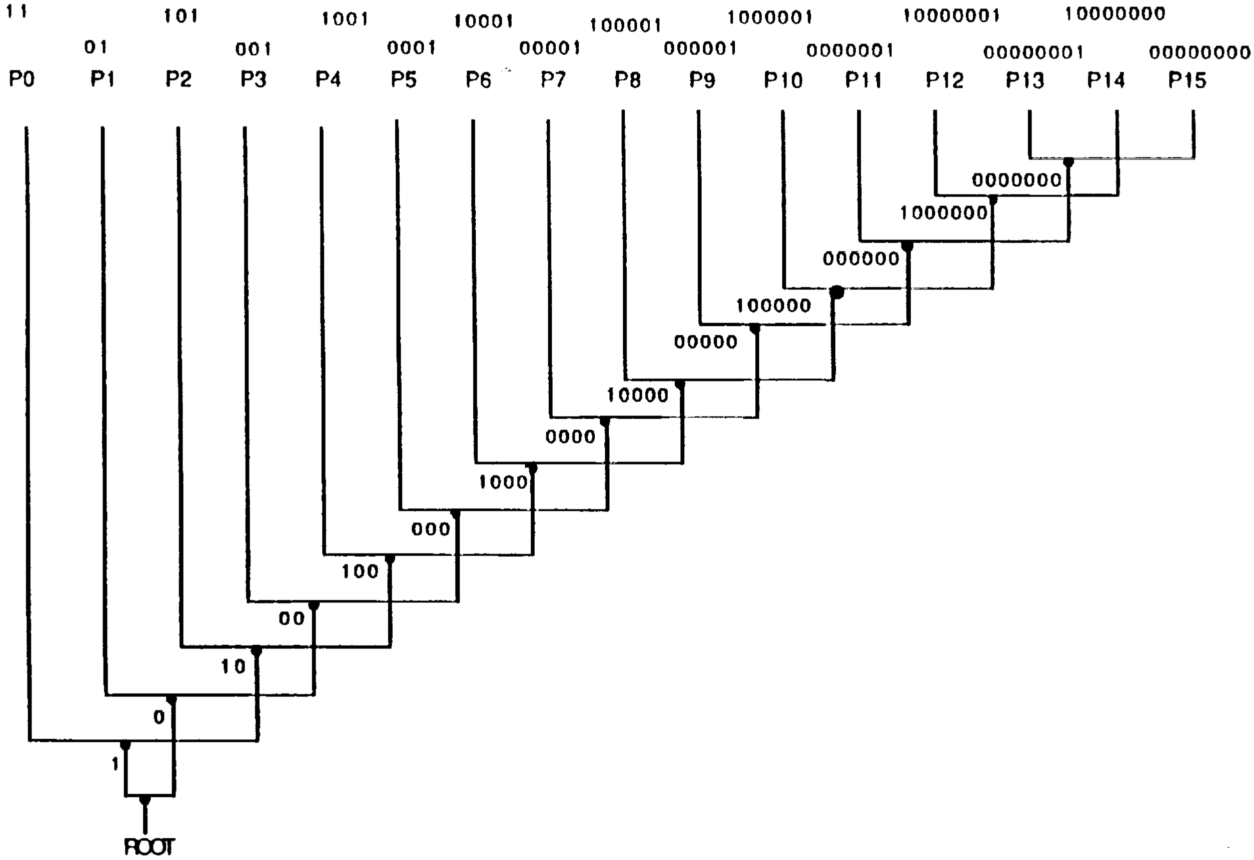


Figure 12: The $\psi_{1,1}$ Binary Tree and Codewords for a 16-Symbol Set, with Split Bits Attached in Front of the FS Code

codewords of the same length is the reversed order of the $\psi_{1,1}$ codewords for the two integer symbols. Second, the last two symbols have to be appended with a '1' to become the $\psi_{1,1}$ codewords. Similarly for $k = 2$, we have a tree structure in Fig. 13. Again, for the last four codewords to become $\psi_{1,2}$ codewords of the last four symbols, we must reverse the ordering of symbols at the same length and append an extra '1' to the codewords.

One immediately recognizes that such a tree structure represents a most efficient (redundancy = 0 bit/symbol) Huffman tree when equal probability weights are assigned to the two branches emanating from a common parent node. Thus, the expected codeword length equals the input entropy and can be easily calculated for an infinite staircase geometric symbol set at $k = 2$, as

$$\begin{aligned} E_H &= (\frac{1}{8}+\frac{1}{8}+\frac{1}{8}+\frac{1}{8})\cdot 3+(\frac{1}{16}+\frac{1}{16}+\frac{1}{16}+\frac{1}{16})\cdot 4+\cdots \\ &= 4 \ \mathit{bit/symbol}. \end{aligned} \tag{28}$$

Any large symbol set which does not have a PDF of similar geometric form will still have

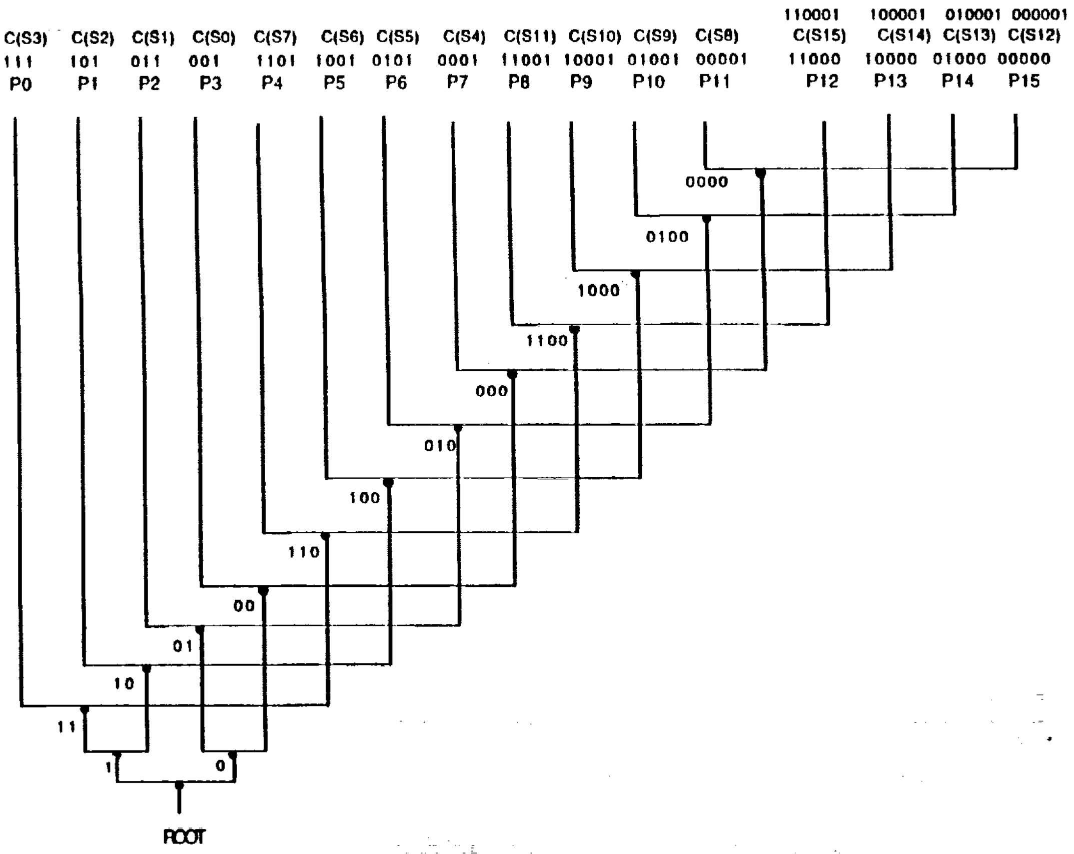


Figure 13: The $\psi_{1,2}$ Binary Code Tree and Codewords of a 16-Symbol Set

an integer expected codeword length for $\psi_{1,k}$ codewords, as long as the probability sum of every 2^k symbols equals $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}$.... For a Laplacian symbol set, this becomes

$$e^{-a2^k} = \frac{1}{2}. \tag{29}$$

Could this binary tree be the actual Huffman tree for a Laplacian symbol set? This is explored next.

III.2 When a $\psi_{1,k}$ Tree Equals a Huffman Tree

The tree structures in Figs. 12 and 13 suggest that each symbol should be paired with a symbol at a separation of 2^k distance when constructing a Huffman code tree of the same structure as the $\psi_{1,k}$ code structure. This pairing imposes a constraint on the Laplacian symbol set. To show that there exists some modified Laplacian PDF whose Huffman code is the $\psi_{1,k}$ code, we first state the following properties:

Property 1. Let $\sum_l$, $\sum_t$ and $\sum_r$ represent different sums of the symbol probabilities defined over a Laplacian PDF given in Eq. (15), where k is a positve integer (the same as the split-sample bits):

$$\sum_l = \sum_{i=N-(k+2)2^k}^{N-(k+1)2^k-1} p(i) \tag{30}$$

$$\sum_t = \sum_{i=N-(k+1)2^k+1}^{N} p(i) \tag{31}$$

$$\sum_r = \sum_{i=N-(k+2)2^k+2}^{N-(k+1)2^k+1} p(i) \tag{32}$$

for $0 \leq i \leq N$, then at $e^{-a2^k} = \frac{1}{2}$,

$$\sum_l > \sum_t > \sum_r. \tag{33}$$

Property 2. Given **Property 1** at a specified k value, and define $\sum_{l+}$, $\sum_{t+}$, $\sum_{r+}$ over the same Laplacian PDF as:

$$\sum_{l+} = \sum_{i=N-(k+2+\ell)2^k}^{N-(k+1+\ell)2^k-1} p(i) \tag{34}$$

$$\sum_{t+} = \sum_{i=N-(k+1+\ell)2^k+1}^{N} p(i) \tag{35}$$

$$\sum_{r+} = \sum_{i=N-(k+2+\ell)2^k+2}^{N-(k+1+\ell)2^k+1} p(i) \tag{36}$$

for a positive integer ℓ, then

$$\sum_{l+} > \sum_{t+} > \sum_{r+}. \tag{37}$$

The properties are proved in Appendix D. A graphical description of each summation is given in Fig. 14(a). If one partitions the symbol set T into segments of 2^k symbols and numbers the segments from the end of the symbol set as depicted in Fig. 14(a), obviously $\sum_t$ is the symbol probability sum of the $k+1$ segments. Now define $\sum_m$ as the probability sum of the the $(k+2)th$ segment,

$$\sum_m = \sum_{i=N-(k+2)2^k+1}^{N-(k+1)2^k} p(i), \quad 0 \leq i \leq N, \tag{38}$$

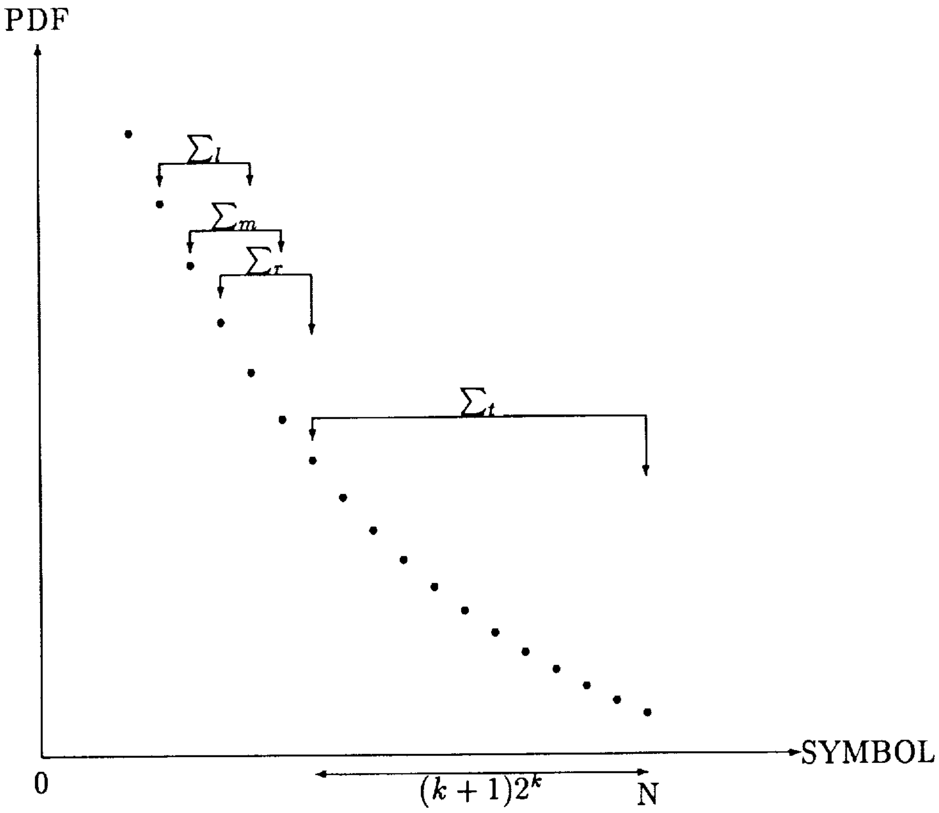


Figure 14(a). Definitions of $\sum_t, \sum_r, \sum_m, \sum_l$ for $k = 2$

then **Property 1** establishes that the probability sum, $\sum_t$, of the $(k+1)$ segments is very close to the probability sum, $\sum_m$, of the $(k+2)th$ segment, whose value, of course, lies between $\sum_l$ and $\sum_r$.

Property 2 extends the definition of $\sum_t$ in **Property 1** to include more 2^k-symbol segments. It guarantees that the probability sum of symbols in the $k+1+\ell$ segments is bounded and is very close to the probability sum of the symbols in the next segment. These two properties will be used to establish the equivalence of a split-sample code to the Huffman code of a modified Laplacian set.

There exist various ways of defining this modified Laplacian set. One such probability assignment which simplifies later derivation and leads to our goal is to impose a slight deviation to the original probability $p(i)$ of symbols in the $k+1$ segments of S in Fig. 14(a), while keeping the same values for symbols in other segments. Thus, as shown in Fig.

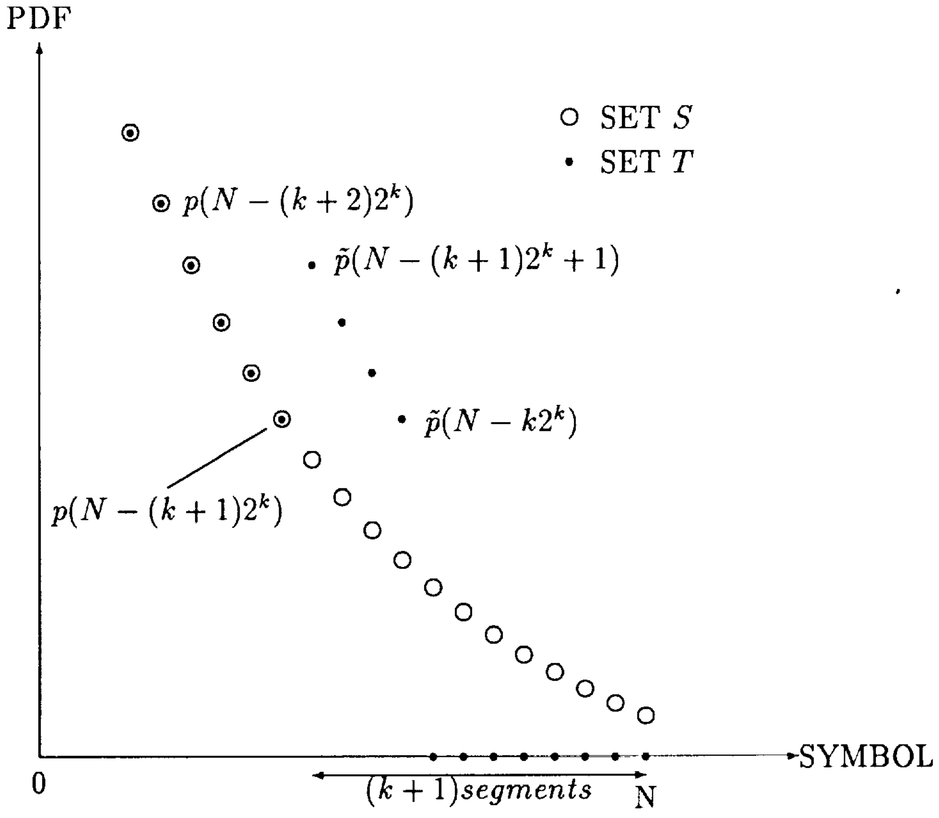


Figure 14(b). Laplacian Set S and the Modified Laplacian Set T

14(b), we define $\tilde{p}(i)$ as the symbol probability for T as

$$\tilde{p}(i) = \begin{cases} p(i) & 0 \le i \le N-(k+1)2^k \\ p(i-2^k) + \delta \cdot e^{-aj} & N-(k+1)2^k+1 \le i \le N-k2^k \\ & j = i-(N-(k+1)2^k+1) \\ 0 & N-k2^k \le i \le N \end{cases} \tag{39}$$

under the constraint that the symbol probability sum over the $(k+1)th$ segment equals $\sum_t$. The deviation from the original probability, δ, is obtained from

$$\sum_t - \sum_m = \delta \cdot \frac{1-e^{-a2^k}}{1-e^{-a}}. \tag{40}$$

One can verify that the symbol probability defined above over the $(k+1)th$ segment of T provides, between symbols within this segment, an exponential relation of the form

$$\tilde{p}(N-(k+1)2^k+1+j) = \tilde{p}(N-(k+1)2^k+1) \cdot e^{-aj}, \quad j = 0, \cdots, 2^k-1 \tag{41}$$

the same as what is between symbols in S. When $e^{-a2^k} = \frac{1}{2}$, one can easily show that $\sum_m > \sum_t$, therefore δ is negative.

With the above definition, we now state the following:

Theorem 1 *The $\psi_{1,k}$ code of symbol set T is its Huffman code at $e^{-a2^k} = \frac{1}{2}$.*

The proof is given in Appendix E. This theorem establishes the equivalence of the split-sample code to the Huffman code of a modified Laplacian symbol set at a particular symbol entropy level dictated by $e^{a2^k} = \frac{1}{2}$. One should be aware that the modified Laplacian symbol set which satisfies our goal is not unique. A slight variation in assigning the probability values to the symbols in the $(k+1)th$ segment of T results in a different set. The closeness of the modified Laplacian PDF to the original Laplacian PDF is examined in the following section.

III.3 Simulation and Discussion

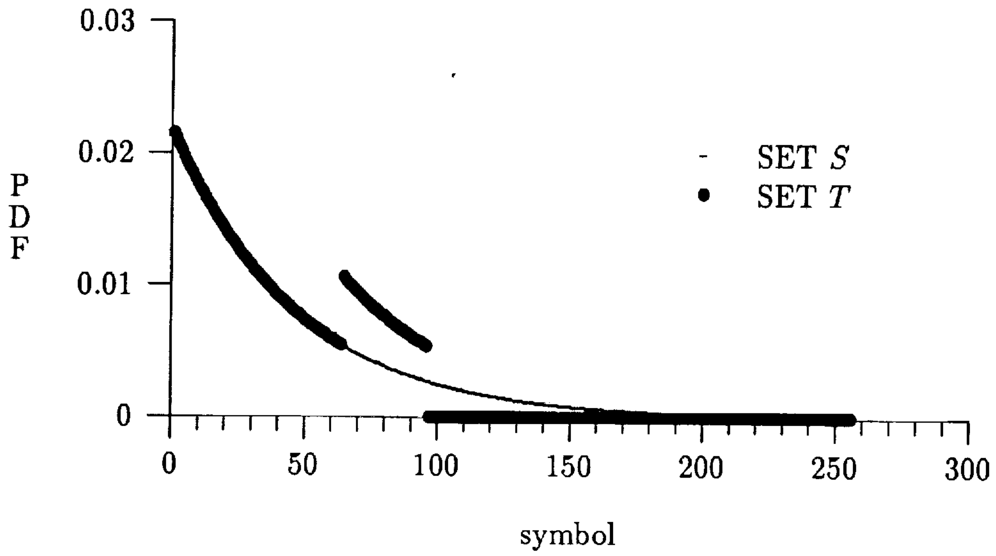


Figure 14(c). PDF of a Laplacian Set of 256 Symbols

Properties 1, 2 can be easily verified for a Laplacian symbol set. The symbol entropy and the expected split-sample codeword length can be calculated using Eqs. (21) and (24). The values are listed in Table 1 for a set of 256 symbols. One notices that, in this example, the expected codeword length is an integer when $k \leq 5$. For this range of k values and 8-bit quantization, the aN value is so large that e^{-aN} approximates zero. Thus Eq. (24) can be simplified to

$$\begin{aligned} E[p'_k(j)] &\approx 1+k+\frac{e^{-\beta}}{1-e^{-\beta}}, \\ &= 2+k. \end{aligned} \tag{42}$$

At $k = 6$, **Theorem 1** still holds, but the expected codeword length is no longer an integer.

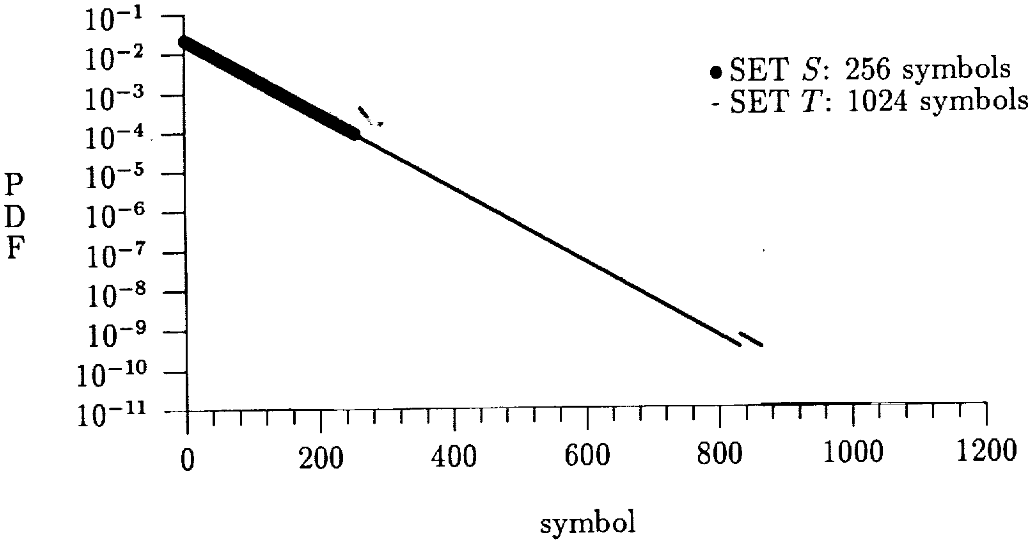


Figure 14(d). PDFs of a Laplacian Set and of a Modified Laplacian Set

For a 256-symbol modified Laplacian symbol set T, the PDF for $e^{-a2^k} = \frac{1}{2}$, at $k = 5$ is given in Fig. 14(c). One notices the apparent difference between this curve and a normal Laplacian distribution. However, as Fig. 14(d) shows, if N is allowed to become a larger number, for instance 1023 for 10-bit data, the PDFs for this 1024-symbol modified Laplacian set T and for a normal 256-symbol Laplacian set S are indistinguishable at the same specified Laplacian parameter a. Theoretically, the two curves in Fig. 14(d) differ only minimally by a normalization factor of $(1 - e^{-a(N+1)})$, as defined in Eqs. (15) and (16). The $(k + 1)th$ segment of set T in Fig. 14(c), which occurs at the 65th symbol of the 256-symbol set $(256 - (5 + 1) \cdot 2^5 + 1)$ has been pushed farther away to the 833rd symbol of the 1024-symbol set $(1024 - (5 + 1) \cdot 2^5 + 1)$. Its value diminishes exponentially fast and lies below 10^{-8}. The last $k2^k = 160$ symbols have 0 values and are plotted as 10^{-11} for visualization on the logarithmic plot. One will, in fact, use the first 256 Huffman codes out of the 1024 Huffman codes constructed for this 1024-symbol set T to code the 256-symbol set S, whose PDF is practically equivalent to those of the set T.

To summarize, we simply state that the $\psi_{1,k}$ codes are a set of Huffman codes at integer expected codeword lengths for an infinitely large modified Laplacian symbol set and for an infinitely large staircase geometric symbol set. One uses these codewords to code a Laplacian symbol set of limited elements. Since $\psi_{1,k}$ coding is a top-down procedure, meaning a codeword can be readily derived by knowing the symbol's order in the set, no

Split Bit k	Symbol Entropy Bit/Symbol	Expected Codeword Length Bit/Symbol
0	2.0000	2.0000
1	2.9787	3.0000
2	3.9733	4.0000
3	4.9719	5.0000
4	5.9713	5.9998
5	6.9710	7.0040
6	7.6117	7.7333

Table 1: Symbol entropy and $\psi_{1,k}$ expected codeword length for a Laplacian symbol set of 256 elements at $e^{-a2^k} = \frac{1}{2}$

codebooks need to be generated before actual coding takes place.

IV PERFORMANCE ON AERIAL IMAGERY

A split-sample coding scheme has been simulated on the VAX computer. A preprocessor was used which simply takes the previous pixel as the prediction value. A test set was selected, shown in Fig. 3, consisting of nine 128 x 128 pixel images. The top row consists of 8-bit, 3m ground resolution images whose differential entropy measurement is the smallest of the image set. The middle portion of Fig. 3 has 8-bit, 1km ground resolution images. Fig. 3(f) is the infrared version of Fig. 3(e). These three images have a medium level of differential entropy. The last row shows 12-bit, 20m ground resolution images, with a much higher differential entropy.

To adapt to the change in scene statistics, an optimal choice of the number of the split k bits is selected and coded, using 3 bits as option identification for every block of $J = 16$ input samples. A reference signal of the first pixel value in each scan line is also retained. The overall system coding performance, including the overhead information of approximately 0.32 bit/sample, is plotted in Fig. 15, against the differential entropy. The closeness of these results to the ideal curve validates the effectiveness of this coding scheme.

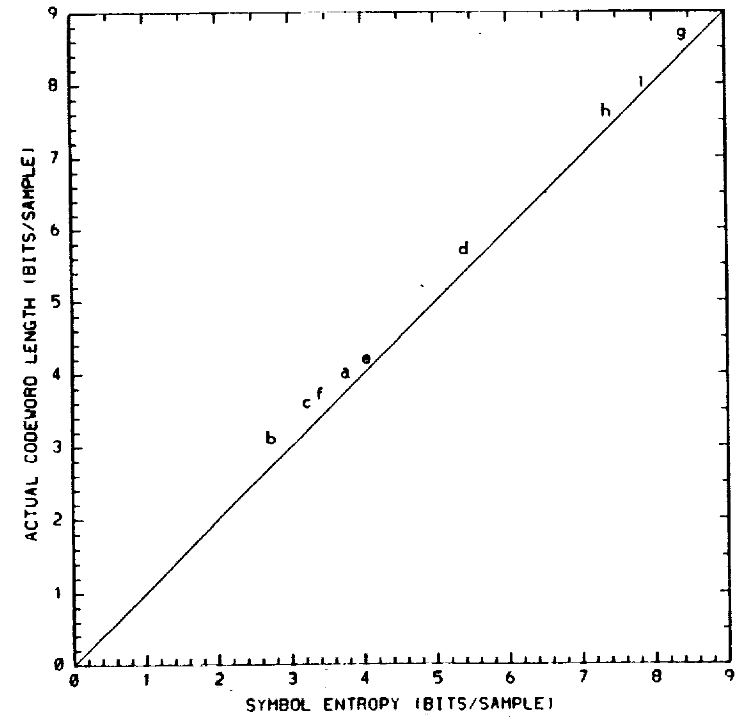


Figure 15: Coding Results for Fig. 3 Images

V SPLIT-SAMPLE CODING ON OTHER SOURCE SYMBOL PDFs

Having observed the effectiveness of the split-sample FS coding on symbols with Laplacian PDF, one wonders if the same scheme can be as effective if applied to other types of PDFs. Two other types of PDFs are considered: a Gaussian PDF, due to its frequent usage in signal modelling, and a Poisson PDF, which is often used to model the hit-rate of high-energy photons in NASA's cosmic ray observations.

V.1 Gaussian PDF

The Humblet condition stated in (8) and (9) demands that the symbol PDF decrease monotonically and fast. For a Gaussian PDF, this condition is only met when the parameter σ is small, resulting in a very steep Gaussian PDF. For a Gaussian PDF of the form:

$$p(i) = \frac{1}{\sqrt{2\pi}\sigma A_g} e^{-\frac{i^2}{2\sigma^2}}, \qquad 0 \leq i \leq N \tag{43}$$

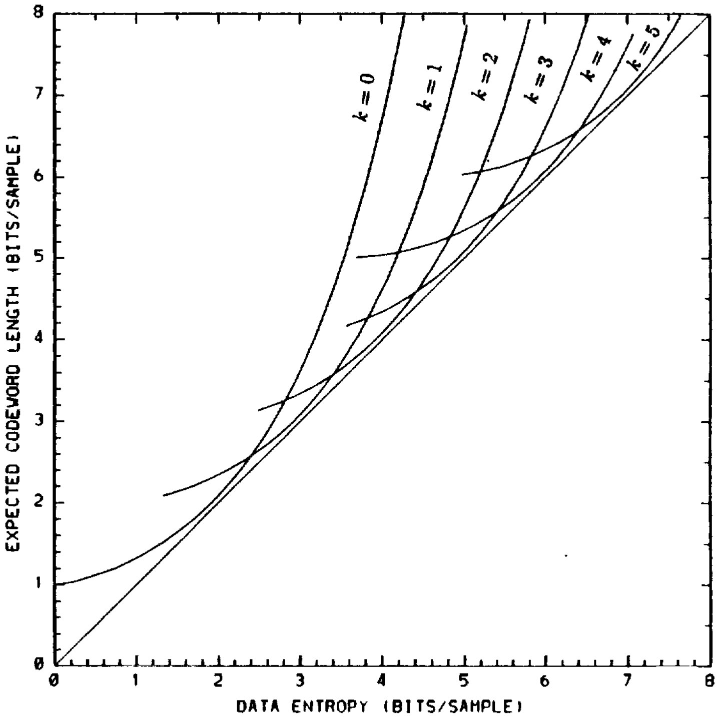


Figure 16: Coder Performance on a Gaussian PDF

where A_g, the normalization factor is given as

$$A_g = \sum_{i=0}^{N} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{i^2}{2\sigma^2}},$$

we can derive the range of σ of the PDF to satisfy the Humblet condition (see Appendix F). Without splitting bits, that is, when $k = 0$,

$$e^{-\frac{1}{2\sigma^2}} \geq \sqrt{2\pi}\sigma A_g - 2, \qquad (44)$$

whereas for $k > 0$

$$e^{-\frac{1}{2\sigma^2}} + e^{-\frac{4}{2\sigma^2}} + \cdots + e^{-\frac{(2^k-1)^2}{2\sigma^2}} \geq \frac{\sqrt{2\pi}\sigma A_g - 2}{2}. \qquad (45)$$

The overall performance of split-sample FS coding on a Gaussian PDF is given in Fig. 16. When compared with Fig. 9, this coding scheme offers comparable coding rates, though the coding rate is slightly higher for a Gaussian source than for a Laplacian source. It is noted that for $k = 0$, the source entropy must be below 0.86 bit/sample for the FS coding

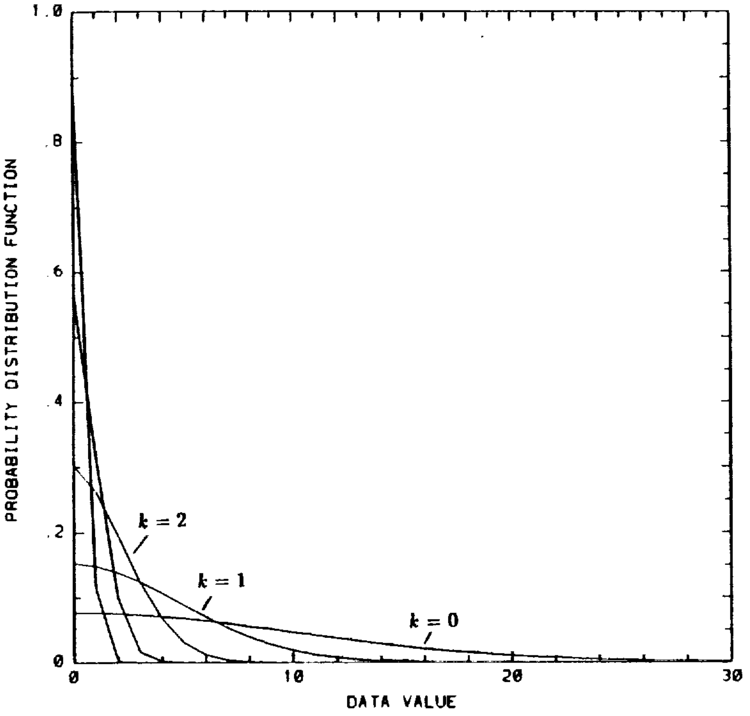


Figure 17: Re-segmented Gaussian PDF for Split Bits from 0 to 4

to be the optimal Huffman code, as compared with 2.51 bits/sample for the Laplacian PDF (see Fig. 6); however, when more bits are split from the symbols, the Gaussian PDF of the reduced symbol set more closely resembles the Laplacian PDF, as can be seen in Fig. 17.

V.2 Poisson PDF

The Poisson PDF, often used to characterize the number of random discrete hits i within a given collection time τ and a given average hit rate of λ, is given by:

$$p_\tau(i) = \frac{(\lambda\tau)^i}{i!}e^{-\lambda\tau}. \tag{46}$$

The Poisson PDF is very narrow, and the distribution skews towards $i = 0$, when $\lambda\tau$ is small. It resembles a Gaussian PDF when $\lambda\tau$ becomes large. No closed form mathematical condition can be derived with the PDF to predict when the Humblet condition is satisfied. However, the split-sample coding performance is very close to that of a Gaussian PDF, as shown in the simulation results in Fig. 18.

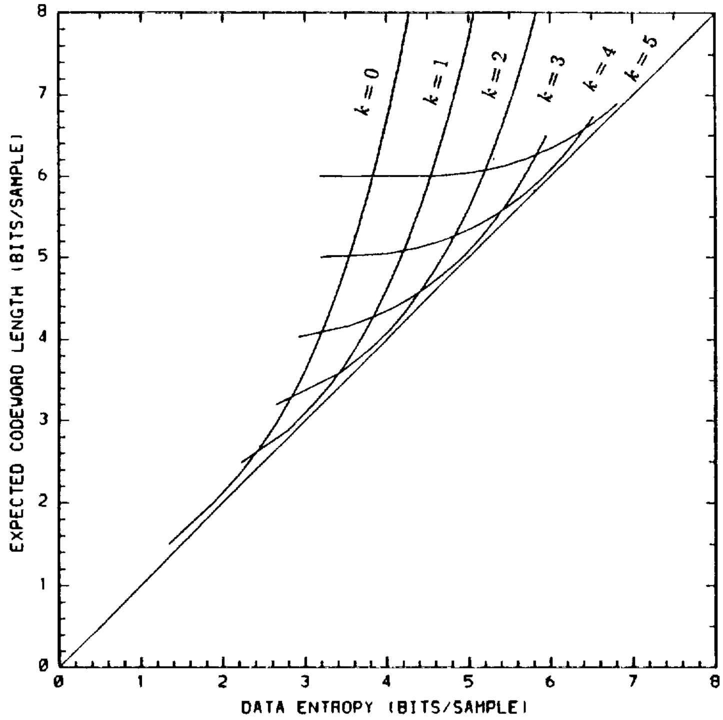


Figure 18: Coder Performance on a Poisson PDF

VI DISCUSSION

To summarize, we state that the $\psi_{1,k}$ codes are a set of Huffman codes with expected performance centered at integer codeword lengths for an infinitely large modified Laplacian symbol set. For real world applications for which symbol distributions are well modeled as Laplacian, the practical results are both simple and profound: *The best code to use at each integer entropy value* $(k+2)$ *is the corresponding* $\psi_{1,k}$ *code.* Further, the codeword for each symbol is completely specified by knowing its order in the set. No codebooks are needed.

Clearly, the implementation of any individual split-sample coder is extremely simple, in both software and hardware. After all, the coding of an N-bit sample requires no more than splitting off k least significant bits and then replacing the remaining $n-k$ most significant bits with i binary zeroes and a 1, where i is the integer value of the most significant bits. This simplicity extends to the implementation of adaptive coders built around these split-sample modes as options to choose from. Yet as we have shown, this simplicity does not sacrifice performance.

Split Bit k	Symbol Entropy Bit/Symbol	Expected Codeword Length Bit/Symbol
0	2.0000	2.0000
1	2.9787	3.0000
2	3.9733	4.0000
3	4.9719	5.0000
4	5.9716	6.0000
5	6.9715	7.0000
6	7.9715	8.0000
7	8.9714	9.0000
8	9.9714	10.0000
9	10.9714	11.0000
10	11.9711	11.9998

Table 2: Symbol entropy and $\psi_{1,k}$ expected codeword length for a Laplacian symbol set of 16,384 elements at $e^{-a2^k} = \frac{1}{2}$

Two different hardware implementations are being developed as custom CMOS VLSI chip sets. The Jet Propulsion Laboratory is implementing a mission specific coder, whereas the Goddard Space Flight Center (GSFC), in collaboration with the University of Idaho, is implementing a multi-mission encoder/decoder chip set, capable of broader applicability. Both implementations select code options over blocks of 16 samples.

The 1.0μm GSFC chip set is designed to operate at up to 20 million samples/sec when used on 14 bits/sample data while adaptively providing coding performance close to the local entropy over an entropy range of from 1.5 to 12.5 bits/sample.

To provide the flexibility to support broad mission requirements, the GSFC version will:

- allow the use of an externally supplied predictor;
- allow the use at an externally supplied pre-processor;
- operate on data quantized from 4 to 14 bits/sample;
- optionally provide automatic insertion of reference samples between data blocks.

This coder implements twelve options: one default in addition to eleven split-sample modes. The performance of the coder on a Laplacian symbol source of 14 bits/sample quantization is listed in Table 2, for input entropy values at $e^{-a2^k} = \frac{1}{2}$, $k = 0, \cdots, 10$.

References

[1] D. A. Huffman, "A method for the construction of minimum redundancy codes," *Proc. IRE*, Vol. 40, pp. 1098-1101, 1952.

[2] R. G. Gallager, "Variations on a theme by Huffman," *IEEE Trans. on Information Theory*, Vol. IT-24, No. 6, pp. 668-674, 1978.

[3] P. A. Humblet, "Optimal source coding for a class of integer alphabets," *IEEE Trans. on Information Theory*, Vol IT-24, No. 1, pp. 110-112, 1978.

[4] O. Johnsen, "On the redundancy of binary Huffman codes," *IEEE Trans. on Information Theory*, Vol. IT-26, No. 2, pp. 220-222, 1980.

[5] G. V. Cormack and R. N. Horspool, "Algorithms for adaptive Huffman codes," *Information Proc. Lett.* Vol. 18, pp. 159-165, 1984.

[6] R. M. Capocelli, R. Giancarlo, and I. J. Taneja, "Bounds on the redundancy of Huffman codes," *IEEE Trans. on Information Theory,* Vol. IT-32, No. 6, pp. 854-857, 1986.

[7] H. Tanaka, "Data structure of Huffman codes and its application to efficient encoding and decoding," *IEEE Trans. on Information Theory*, Vol. IT-33, No. 1, pp. 154-156, 1987.

[8] B. L. Montgomery and J. Abrahams, "On the redundancy of optimal binary prefix-condition codes for finite and infinite sources," *IEEE Trans. on Information Theory*, Vol. IT-33, No. 1, pp. 156-160, 1989.

[9] G. V. Cormack and R. N. Horspool, "Algorithms for adaptive Huffman codes," *Information Processing Letters*, Vol. 18, pp. 159-165, 1984.

[10] D. E. Knuth, "Dynamic Huffman coding," *J. of Algorithms*, Vol. 6, 163-180, 1985.

[11] J. S. Vitter, "Design and analysis of dynamic Huffman codes," *J. of the ACM*, Vol. 34, No. 4, pp. 825-845, 1987.

[12] R. F. Rice and J. R. Plaunt, "Adaptive variable-length coding for efficient compression of spacecraft television data," *IEEE Trans. on Communication Technology*, Vol. COM-19, No. 6, pp. 889-897, 1971.

[13] R. F. Rice, "Some practical universal noiseless coding techniques," JPL Publication 79-22, Jet Propulsion Laboratory, Pasadena, California, 1979.

[14] R. F. Rice and J.-J. Lee, "Some practical universal noiseless coding techniques, part II," JPL Publication 83-17, Jet Propulsion Laboratory, Pasadena, California, 1983.

[15] R. F. Rice, "Some practical universal noiseless coding techniques, Part III, Module PSI14,k+," JPL Publication 91-3, Jet Propulsion Laboratory, Pasadena, California, to be published.

[16] R. F. Rice, P.-S. Yeh, and W. H. Miller, "Algorithms for a very high speed universal noiseless coding module," JPL Publication 91-1, Jet Propulsion Laboratory, Pasadena, California, Feb. 15, 1991.

[17] R. F. Rice, E. Hilbert, J.-J. Lee, and A. Schlutsmeyer, "Block adaptive rate controlled image data compression," *Proc. 1979 National Telecommunications Conf.*, Washington, D.C., 1979.

[18] R. F. Rice and A. Schlutsmeyer, "Data compression for NOAA weather satellite systems," *SPIE Proc: Advances in Image Transmission II,* Vol. 249, 1980.

[19] R. F. Rice and J.-J. Lee, "Noiseless coding for the Gamma ray spectrometer," JPL Publication 85-53, Jet Propulsion Laboratory, Pasadena, California, 1985.

[20] R. F. Rice and J.-J. Lee, "Noiseless coding for the magnetometer," JPL Publication 87-19, Jet Propulsion Laboratory, Pasadena, California, 1987.

APPENDICES

A Derivation of the Humblet Condition on a Laplacian PDF

To satisfy Equation (9) for all nonnegative $n = j + 1$ values, it suffices to show that (9) holds for $j = i + 1$. Therefore we can rewrite (9) as

$$p(i) \geq \sum_{n=i+2}^{N} p(n). \tag{A.1}$$

Substitute the definition in (15) for Laplacian PDF and (A.1) becomes

$$\frac{a}{A}e^{-ai} \geq \frac{a}{A}\sum_{n=i+2}^{N} e^{-an}. \tag{A.2}$$

Carrying out the summation in (A.2), we further reduce it to

$$1 - e^{-a} - e^{-2a} \geq -e^{-a(N-i+1)}, \qquad 0 \leq i \leq N \tag{A.3}$$

Since an exponential function is always nonnegative, then (A.3) is guaranteed as long as

$$1 - e^{-a} - e^{-2a} \geq 0. \tag{A.4}$$

Solving for a in (A.4), we arrive at

$$a \geq \frac{\log(\frac{1+\sqrt{5}}{2})}{\log e} \tag{A.5}$$

which is (17) in this publication.

B Derivation of the Humblet Condition on a Re-segmented Laplacian PDF

If we substitute the re-segmented Laplacian PDF of (18) in (19), we have

$$\frac{a}{A} \cdot \frac{e^{-a2^k j}(1-e^{-a2^k})}{1-e^{-a}} \geq \frac{a}{A} \cdot \frac{1-e^{-a2^k}}{1-e^{-a}} \cdot \sum_{m=j+2}^{N'} e^{-2^k am}. \tag{B.1}$$

The summation in (B.1) is equal to

$$\frac{e^{-2^k a(j+2)}[1-e^{-2^k a(N'-j-2)}]}{1-e^{-2^k a}}$$

where $N' = \frac{N+1}{2^k} - 1$. After rearranging terms on both sides of (B.1), we obtain

$$1 - e^{-2^k a} - e^{-2^{k+1} a} \geq -e^{-2^k a(\frac{N+1}{2^k} - j - 2)}. \tag{B.2}$$

Again, (B.2) holds when the left side of (B.2) always exceeds 0, that is

$$1 - e^{-2^k a} - e^{-2^{k+1} a} \geq 0. \tag{B.3}$$

Solving for $2^k \cdot a$, we obtain

$$2^k \cdot a \geq \frac{\log(\frac{1+\sqrt{5}}{2})}{\log e} \tag{B.4}$$

which is Equation (20) in this publication.

C Derivation of Symbol Entropy for a Re-segmented Laplacian PDF

Equation (18) gives the re-segmented Laplacian PDF, which can be rewritten as

$$p'_k(j) = \frac{\beta}{\Gamma} e^{-\beta j}, \tag{C.1}$$

where $\beta = a \cdot 2^k$ and $\frac{1}{\Gamma} = \frac{a\beta}{A} \cdot \frac{1-e^{-\beta}}{1-e^{-a}}$.

Using the entropy derived for the Laplacian PDF in (21), one can readily write down the following form as the entropy for the re-segmented Laplacian PDF:

$$H_{N'}[p'_k(j)] = -\log_2 \frac{\beta}{\Gamma} + \frac{\beta^2}{\Gamma} \log_2 e \cdot [\frac{e^{-\beta}(1 - e^{-\beta N'})}{(1 - e^{-\beta})^2} - \frac{N' e^{-\beta(N'+1)}}{1 - e^{-\beta}}]. \tag{C.2}$$

Now, replacing β and Γ in (C.2), it immediately reduces to (22) and (23).

D Proof of Properties 1 and 2

For a Laplacian symbol set defined in Eq. (15), the various sums are:

$$\begin{aligned}\Sigma_l &= \sum_{i=N-(k+2)2^k}^{N-(k+1)2^k-1} p(i) \\ &= \frac{a}{A}\frac{e^{-a[N-(k+2)2^k]}(1-e^{-a2^k})}{1-e^{-a}},\end{aligned} \tag{D.1}$$

$$\begin{aligned}\Sigma_t &= \sum_{i=N-(k+1)2^k+1}^{N} p(i) \\ &= \frac{a}{A}\frac{e^{-a[N-(k+1)2^k+1]}(1-e^{-a(k+1)2^k})}{1-e^{-a}},\end{aligned} \tag{D.2}$$

$$\begin{aligned}\Sigma_r &= \sum_{i=N-(k+2)2^k+2}^{N-(k+1)2^k+1} p(i) \\ &= \frac{a}{A}\frac{e^{-a[N-(k+2)2^k+2]}(1-e^{-a2^k})}{1-e^{-a}}.\end{aligned} \tag{D.3}$$

We need to show

$$\Sigma_l > \Sigma_t > \Sigma_r, \tag{D.4}$$

which is equivalent to

$$1.0 > \frac{e^{-a(2^k+1)}(1-e^{-a(k+1)2^k})}{1-e^{-a2^k}} > e^{-2a}. \tag{D.5}$$

Substituting $e^{-a2^k} = \frac{1}{2}$, the above equation becomes

$$1.0 > e^{-a}(1-(\frac{1}{2})^{k+1}) > e^{-2a} \tag{D.6}$$

The left side of the relation always holds because $1.0 > e^{-a} > 0.0$, and $1.0 > 1.0-(\frac{1}{2})^{k+1} > 0.0$, therefore we only need to show that the right side of the above equation is valid. Equivalently, we must prove

$$1-(\frac{1}{2})^{k+1} > e^{-a} \tag{D.7}$$

is valid for all integer $k > 0$. We will use induction for this purpose. At $k = 1$ and $e^{-a2^k} = \frac{1}{2}$, Eq. (D.7) is valid. Now we need to prove that

$$\begin{aligned} If \quad 1-(\tfrac{1}{2})^{k+1} &> (\frac{1}{2})^{\frac{1}{2^k}}, \quad \text{for any } k > 1 && \text{(D.8)} \\ Then \quad 1-(\tfrac{1}{2})^{k+2} &> (\frac{1}{2})^{\frac{1}{2^{k+1}}}. && \text{(D.9)}\end{aligned}$$

Let $u = (\frac{1}{2})^{k+1}$ and $v = (\frac{1}{2})^{\frac{1}{2^k}}$, from (D.8)

$$1 - u > v. \tag{D.10}$$

Taking the square root will keep the relation because both u and v are positive, that is,

$$(1-u)^{\frac{1}{2}} > v^{\frac{1}{2}}. \tag{D.11}$$

Using binomial series expansion,

$$\begin{aligned} (1-u)^{\frac{1}{2}} &\approx 1 - \frac{1}{2}u - \frac{1}{8}u^2 - \cdots \\ &< 1 - \frac{1}{2}u. \end{aligned} \tag{D.12}$$

Therefore we have

$$1 - \frac{1}{2}u > (1-u)^{\frac{1}{2}} > v^{\frac{1}{2}}, \tag{D.13}$$

which is Eq. (D.9). This proves **Property 1**. Similarly, from the definition of $\sum_{l+}, \sum_{t+}, \sum_{r+}$, we can write ·

$$\begin{aligned} \textstyle\sum_{l+} &= \sum_{i=N-(k+2+\ell)2^k}^{N-(k+1+\ell)2^k-1} p(i) \\ &= \frac{a}{A}\frac{e^{-a[N-(k+2+\ell)2^k]}(1-e^{-a2^k})}{1-e^{-a}}, \end{aligned} \tag{D.14}$$

$$\begin{aligned} \textstyle\sum_{t+} &= \sum_{i=N-(k+1+\ell)2^k+1}^{N} p(i) \\ &= \frac{a}{A}\frac{e^{-a[N-(k+1+\ell)2^k+1]}(1-e^{-a(k+1+\ell)2^k})}{1-e^{-a}}, \end{aligned} \tag{D.15}$$

$$\begin{aligned} \textstyle\sum_{r+} &= \sum_{i=N-(k+2+\ell)2^k+2}^{N-(k+1+\ell)2^k+1} p(i) \\ &= \frac{a}{A}\frac{e^{-a[N-(k+2+\ell)2^k+2]}(1-e^{-a2^k})}{1-e^{-a}}. \end{aligned} \tag{D.16}$$

We need to show

$$\textstyle\sum_{l+} > \sum_{t+} > \sum_{r+}, \tag{D.17}$$

which is equivalent to

$$1.0 > \frac{e^{-a(2^k+1)}(1-e^{-a(k+1+\ell)2^k})}{1-e^{-a2^k}} > e^{-2a}. \tag{D.18}$$

The left side holds following the same reasoning used earlier. The right side is equivalent to proving

$$1-(\frac{1}{2})^{k+1+\ell} > e^{-a} \tag{D.19}$$

for a positive integer ℓ. Since Eq. (D.7) holds under **Property 1**, one can easily write

$$1-(\frac{1}{2})^{k+1+\ell} > 1-(\frac{1}{2})^{k+1} > e^{-a}, \tag{D.20}$$

which establishes **Property 2**.

E Proof of Theorem 1

To construct the Huffman code for the modified Laplacian symbol set defined by Eqs. (39) and (40), we start pairing the two symbols with the smallest probabilities. Because the last $k2^k$ symbols have 0 probability value, they can be paired in arbitrary order. One such order is to start with the last symbol in the first segment, i.e., the segment at the end of the symbol set, and pair it with the last symbol in the second segment. This procedure continues until reaching the $(k+1)th$ segment in Fig. 14(b). The definition of T assigns $\tilde{p}(i)$ to each symbol in this segment in an exponentially decreasing way, the same as for a Laplacian set. Because all the Huffman tree nodes resulting from the last $k2^k$ symbols have 0 probability value, all symbols in the $(k+1)th$ segment will be paired in the same fashion. This results in 2^k nodes with node values the same as the symbol probabilities in the $(k+1)th$ segment. These nodes preferably will pair with the 2^k symbols in the $(k+2)th$ segment. To deduce the desired relation, we first prove two lemmas.

Lemma 1 $\tilde{p}(N-(k+2)2^k+1+j) > \tilde{p}(N-(k+1)2^k+1+j) > p(N-(k+2)2^k+2+j)$, for $0 \le j \le 2^k - 1$

Lemma 2 $\tilde{p}(N-(k+1)2^k+1+j)+\tilde{p}(N-(k+2)2^k+1+j) = \tilde{p}(N-(k+3)2^k+1+j)+\delta e^{-aj}$, for $0 \le j \le 2^k - 1$

Essentially, **Lemma 1** sets upper and lower probability bounds for the jth symbol in the $(k+1)th$ segment of T, respectively, by the jth and the $(j+1)th$ symbol probability in the $(k+2)th$ segment. **Lemma 2** extends the relation between symbols in the $(k+2)th$ segment and nodes resulting from pairing all symbols in the $k+1$ segments, to the symbols in the $(k+3)th$ segment and the newly formed nodes. The proof is as follows:

Proof of **Lemma 1**: The left-side inequality is immediately established from the definition in Eqs. (39) and (40) and the fact that δ is negative. For the right-side inequality, we need to show that a single symbol which contradicts the inequality will violate **Property 1**, and thus is not allowed. We will outline the proof for the case of the symbol when $j = 0$. A similar procedure can be applied to cases when $0 < j \le 2^k - 1$.

If at $j = 0$, we have

$$\tilde{p}(N-(k+1)2^k+1) \le p(N-(k+2)2^k+2), \tag{E.1}$$

then multiplying both sides by e^{-a} will give

$$\tilde{p}(N-(k+1)2^k+1)\cdot e^{-a} \leq p(N-(k+2)2^k+2)\cdot e^{-a}, \tag{E.2}$$

which from Eq. (41) is equivalent to

$$\tilde{p}(N-(k+1)2^k+2) \leq p(N-(k+2)2^k+3). \tag{E.3}$$

Continue multiplying by e^{-a} on both sides and making use of Eq. (41), we will have

$$\tilde{p}(N-(k+1)2^k+1+j) \leq p(N-(k+2)2^k+2+j), \quad 0 \leq j \leq 2^k-1. \tag{E.4}$$

Summing both the left and right sides of the above equation for the 2^k j values, we arrive at

$$\sum_t \leq \sum_r, \tag{E.5}$$

which contradicts **Property 1**.

Proof of **Lemma 2:** From the definition of T, we can write

$$\begin{aligned} \tilde{p}(N-(k+1)2^k+1+j) &+ \tilde{p}(N-(k+2)2^k+1+j) \\ &= \tilde{p}(N-(k+2)2^k+1+j)+\delta e^{-aj}+\tilde{p}(N-(k+2)2^k+1+j) \\ &= 2\tilde{p}(N-(k+2)2^k+1+j)+\delta e^{-aj} \\ &= 2\tilde{p}(N-(k+3)2^k+1+j)e^{-a2^k}+\delta e^{-aj} \\ &= \tilde{p}(N-(k+3)2^k+1+j)+\delta e^{-aj}, \end{aligned} \tag{E.6}$$

where we have used the fact that $e^{-a2^k} = \frac{1}{2}$.

From **Lemma 1**, the 2^k nodes resulting from pairing symbols in the $(k+1)th$ segment with previous nodes will pair with symbols in the $(k+2)th$ segment in an orderly way. The newly formed nodes, from **Lemma 2**, hold a relation to the symbols in the $(k+3)th$ segment similar to that between the symbols in the $(k+2)th$ segment and the old nodes. Due to **Property 2**, the jth new node value is also bounded above and below, respectively, by $\tilde{p}(N-(k+3)2^k+1+j)$ and $\tilde{p}(N-(k+3)2^k+2+j)$, when $0 \leq j \leq 2^k-1$. Therefore, the same pairing procedure will be followed for the $(k+3)th$ segment and will continue until all symbols are paired. This results in a Huffman code structure for the modified Laplacian set T.

F Derivations of (44) and (45)

The Gaussian PDF given in (43) generally does not satisfy the Humblet condition in (9) due to its zero derivative at the origin. However, when the σ value is small or when split-sample coding is applied, the PDF becomes sufficiently steep that the Humblet condition is possible. To derive the constraint on σ and k for Gaussian PDF, we first prove that it suffices to apply (9) when $i = 0$ to guarantee cases when $i > 0$; then we explicitly derive the constraint for $i = 0$.

Applying (9) to Gaussian PDF in (43) and letting $j = i + 1$, we obtain

$$e^{\frac{-i^2}{2\sigma^2}} > e^{-\frac{(i+2)^2}{2\sigma^2}} + e^{-\frac{(i+3)^2}{2\sigma^2}} + \cdots + e^{-\frac{N^2}{2\sigma^2}}, \tag{F.1}$$

which can be simplified to

$$1 > e^{-\frac{4i+4}{2\sigma^2}} + e^{-\frac{6i+9}{2\sigma^2}} + \cdots + e^{-\frac{N^2-i^2}{2\sigma^2}}. \tag{F.2}$$

There are different numbers of terms in (F.2) when i has different integer values. Since the first term on the right side of (F.1) is always greater than the following $N - i - 2$ terms, (F.2) is guaranteed when

$$1 > (N - i - 1)e^{-\frac{4i+4}{2\sigma^2}} \tag{F.3}$$

is valid. Noting that the exponential term in (F.3) is larger when $i = 0$ than when $i > 0$, and that there are more terms of this form when $i = 0$, we conclude that as long as (F.3) holds for $i = 0$, it is also valid when $i > 0$. Therefore, we only have to apply the Humblet condition to the Gaussian PDF and derive the constraint on σ when $i = 0$.

Instead of working with (F.3) directly, applying (9) and setting $i = 0$, we can address the Humblet condition in the following easier way:

$$p(0) \geq 1 - p(0) - p(1). \tag{F.4}$$

Using the Gaussian PDF in (43), we obtain

$$\frac{1}{\sqrt{2\pi}\sigma A_g} \geq 1 - \frac{1}{\sqrt{2\pi}\sigma A_g} - \frac{1}{\sqrt{2\pi}\sigma A_g} e^{-\frac{1}{2\sigma^2}}. \tag{F.5}$$

(F.5) can be written as

$$e^{-\frac{1}{2\sigma^2}} \geq \sqrt{2\pi}\sigma A_g - 2, \tag{F.6}$$

which is (44).

For the reduced symbol set resulting from k split bits, applying (9) and rearranging terms, we obtain

$$\sqrt{2\pi}\sigma A_g 2(1 + e^{-\frac{1}{2\sigma^2}} + e^{-\frac{4}{2\sigma^2}} + \cdots + e^{-\frac{(2^k-1)^2}{2\sigma^2}}) + p'_k(1) \geq 1, \tag{F.7}$$

since the second term $p'_k(1)$ is smaller than the first one, a more strict condition of (F.7) is

$$\frac{1}{\sqrt{2\pi}\sigma A_g} 2(1 + e^{-\frac{1}{2\sigma^2}} + e^{-\frac{4}{2\sigma^2}} + \cdots + e^{-\frac{(2^k-1)^2}{2\sigma^2}}) \geq 1, \tag{F.8}$$

which can be written as

$$e^{-\frac{1}{2\sigma^2}} + e^{-\frac{4}{2\sigma^2}} + \cdots + e^{-\frac{(2^k-1)^2}{2\sigma^2}} \geq \frac{\sqrt{2\pi}\sigma A_g - 2}{2}, \tag{F.9}$$

the same as (45).