arrangements. Over 150 offices have been equipped at this writing, and planning is under way for many more. System performance to date has been very satisfactory.

Bibliography

[1] A. E. Vitalo, "The concept of automatic number identification," *Bell Lab. Rec.*, vol. 36, pp. 153–156, May 1958.

[2] D. H. Pennoyer, "Automatic number identification and its application to No. 1 crossbar, panel and step-by-step offices," *Bell Sys Tech. J.*, vol. 37, pp. 1295–1318, September 1958.

[3] H. A. Miloche, "Automatic number identification," *Bell Lab. Rec.*, vol. 38, pp. 211–216, June 1960.

[4] C. H. Dagnall, Jr., "Automatic number identification: outpulsers and identifiers," *Bell Lab. Rec.*, vol. 39, pp. 97–102, March 1961.

[5] T. R. Redington, "A low-cost number identifier for small telephone systems," *AIEE Trans.* (*Communication and Electronics*), vol. 80, pp. 117–121, May 1961.

[6] H. D. Cahill, "Maintaining the ANI system," *Bell Lab. Rec.*, vol. 40, pp. 408–412, December 1962.

Harold R. Moore was born in Pittsburgh, Pa., on January 29, 1920. He served in the U.S. Navy from 1944 to 1946 as an electronics technician.

In 1940 he joined the Bell Telephone Company of Pennsylvania, where he worked as a switchman, a circuit layer engineer, and a transmission maintenance engineer. Since joining Bell Telephone Laboratories, Holmdel, N. J., in 1961, he has worked on exploratory development of a high-voltage translator, automatic number identification for small step-by-step offices, and automatic intercept service for panel offices. He is presently a member of the Technical Staff of the Common Systems Switching Laboratory of Bell Telephone Laboratories. He has been granted two patents, and is currently working on the compatibility of E-type single frequency signaling with switching circuits.

Mr. Moore is a Registered Professional Engineer in the State of Pennsylvania.

Prefix Coding of Histograms for Minimal Storage

IRWIN M. JACOBS, MEMBER, IEEE, AND LEONARD KLEINROCK, MEMBER, IEEE

REFERENCE: Jacobs, I. M., and Kleinrock, L.: PREFIX CODING OF HISTOGRAMS FOR MINIMAL STORAGE,[1] University of California, La Jolla, Calif., and University of California, Los Angeles, Calif. Mr. Jacobs was formerly with the Massachusetts Institute of Technology, Cambridge, Mass. Rec'd 8/1/66; revised 8/27/66. Paper 19TP67-916. IEEE TRANS. ON COMMUNICATION TECHNOLOGY, 15-2, April 1967, pp. 149-155.

ABSTRACT: The reduction of experimental data to histograms is often useful, particularly if the experimental site is remote (e.g., in a spacecraft). An implementation utilizing prefix coding is described which permits assembling an N-sample, K-cell histogram directly from sequentially received data with minimal logic and memory requirements. The necessary storage is shown to differ from the minimum number of bits required for unique specification of an (N, K) histogram by less than $3/4\ K$ when $N \gg K$.

The method codes the contents, denoted by n, of a cell into a binary code word of length $f(n)$. A minimax theorem is presented to justify selecting $f(n)$ from a class of linear staircase functions, and the best such function satisfying the Kraft inequality is determined. Graphs are presented of the resulting storage requirements.

***KEYWORDS:* Coding, Communication Theory, Data Storage, Data Transmission Systems, Experimental Data, Information Theory, Processing, Telemeter Systems.**

I. Introduction

The construction of an N-sample, K-cell histogram is a useful data reduction and compression operation for a wide variety of experiments.[1] An N-sample, K-cell histogram, which will be referred to as an (N, K) histogram, is defined as a set of K non-negative numbers $n_1, n_2 \ldots, n_K$ such that $n_1 + n_2 + \ldots + n_K = N$. In this paper, we present an efficient and easily implemented method for assembling, storing, and transmitting histograms in the form of binary-coded sequences. The method is well suited for dynamic (delay line) storage and has interesting implications to list storage of variable-length words in computers.

One obvious method of storing an (N, K) histogram utilizes K blocks of storage, each block containing $[\log_2 N]^+$ bits,[2] for a total storage of $K[\log_2 N]^+$ bits. (Since the total number of samples in the histogram is known to be N, one cell may be omitted and $(K - 1)\ [\log_2 N]^+$ bits will suffice.) This method uses constant length binary representations of each n_k. Since at most one n_k can be as large (or almost as large) as N, this method is rather inefficient in terms of storage.

A second method of storage is incorporated into the Quantiler, a machine built at the Jet Propulsion Labora-

[1] This paper presents the results of one phase of research carried out at the Jet Propulsion Laboratory, California Institute of Technology, under Contract NAS-7-100, sponsored by the National Aeronautics and Space Administration.

[2] In this paper, for any real number k, $[k]^-$ is the integer part of k while $[k]^+$ denotes the smallest integer greater than or equal to k.

tory for the purpose of calculating quantiles.[1],[2] It stores an (N, K) histogram in a delay line register $N + K - 1$ bits long by means of a variable length code. The number of samples in the kth cell n_k is specified by the number of zeros contained between the $(k-1)$st and the kth one, $k = 1, 2, \ldots, K$, except that the 0th and Kth ones are deleted. The beginning and end of the register are fixed and specified by other means. For $K - 1 > N/(\log_2 N - 1)$, this second method has an advantage over the first; for large N, the second method is very inefficient. For example, if $N = 1024$ and $K - 1 = 256$,

$$(K-1)\log_2 N = 2560, \; N + K - 1 = 1280$$

whereas, if $N = 16384$ and $K - 1 = 256$,

$$(K-1)\log_2 N = 3584, \; N + K - 1 = 16640.$$

II. Optimum Storage

The minimum storage size required for (N, K) histograms is easily determined. The number of distinct histograms (partitions of N into K parts) is the binomial coefficient $\binom{N+K-1}{K-1}$. Thus, the minimum binary storage required for unambiguously specifying an (N, K) histogram is

$$S_{\min} = \left[\log_2 \binom{N+K-1}{K-1}\right]^+ \text{ bits.} \tag{1}$$

This minimum storage is achievable by means of a *code book*. Since the storage required for a code book is in general unreasonable, Gordon[3] has given an algorithm and an implementation which permits an optimum code achieving $S_{\min}$ to be computed from the monotonic sequence

$$(n_1, n_1 + n_2, n_1 + n_2 + n_3, \ldots, n_1 + n_2 + \ldots + n_{K-1}).$$

Unfortunately, this method does not appear to be adaptable to the assembly of histograms from samples arriving one at a time unless a coding and storage device similar to that described in the remainder of this report is also available. If the cost of transmission is extremely high, however, it may pay in some cases to use the Gordon coding on the complete (N, K) histogram prior to transmission even though, as demonstrated in the sequel, the additional savings in the number of transmitted bits is small.

To facilitate comparisons, we now determine useful approximations to $S_{\min}$. By Stirling's approximation, we have

$$\binom{N+K-1}{K-1} \approx \left(\frac{N+K-1}{2\pi(K-1)N}\right)^{1/2}\left(1 + \frac{N}{K-1}\right)^{K-1} \times \left(1 + \frac{K-1}{N}\right)^{N}.$$

Thus

$$S_{\min} \approx S^* = (K-1)\log_2\left(1 + \frac{N}{K-1}\right) + N\log_2\left(1 + \frac{K-1}{N}\right) + \frac{1}{2}\log_2\left(\frac{N+K-1}{2\pi(K-1)N}\right). \tag{2}$$

Assuming $N \gg K \gg 1$, the usual case, yields

$$S^* \approx S^{**} = (K-1)\left[\log_2\left(\frac{N}{K-1}\right) + \log_2 e\right] - \frac{1}{2}\log_2(K-1). \tag{3}$$

For example

$N = 1024, K - 1 = 256$

$$S_{\min} = 919, \; S^* = 919, \; S^{**} = 877;$$

$N = 16\,384, K - 1 = 256$

$$S_{\min} = 1903, \; S^* = 1903, \; S^{**} = 1901.$$

Note that the two nonoptimum schemes discussed in Section I fall significantly short of the optimum.

III. Linear Prefix Code

A close approach to the optimum may be obtained by the use of prefix code theory.[4] The prefix condition states that, although code words may differ in length, no code word contains another code word as its initial portion. As a result, the code words are uniquely and instantaneously decipherable.

For coding an (N, K) histogram, a binary code word satisfying the prefix condition is assigned to each number between 0 and N. The code words representing the contents of the K cells can then be placed in a shift register or delay line, one after the other, without the use of additional separators. Since the same prefix code is used for each cell, $S_{\min}$ is not achieved. However, we show in Section V that the best of the prefix codes is very nearly optimum.

The problems of choosing a good prefix code remain. It appears reasonable that a code for which all histograms have essentially the same storage requirement is best. This equi-storage property is achieved if the length, say $f(n)$, of the code word assigned to the number n is a linear function of n, that is, if, ignoring integer constraints for the moment,

$$f(n) = an + b. \tag{4}$$

In this case the storage S required for the histogram $(n_1, n_2, \ldots, n_K)$ is (again observing that the Kth cell need not be sent)

$$\begin{aligned} S &= f(n_1) + f(n_2) + \ldots + f(n_{K-1}) \\ &= a(n_1 + n_2 + \ldots + n_{K-1}) + b(K-1) \\ &\leq aN + b(K-1) \end{aligned} \tag{5}$$

where the inequality follows from the observation that $\sum_{k=1}^{K} n_k = N$. The equality sign is achieved when $n_K = 0$.

The use of a prefix code which is linear is justified by means of the following general theorem.

Let f denote an integer-valued function and let C be a class of functions such that if g is in C and if $f(n) \geq$

$g(n)$, $n = 0, 1, 2, \ldots, N$, then f is also in C. Let p_x be the density function of a discrete random variable x which takes on only the non-negative integer values $0, 1, 2, \ldots, N$, and let an overhead bar denote statistical expectation.

Theorem

Subject to the conditions f in C and $\bar{x} = \mu$,

$$\min_f \max_{p_x} \overline{f(x)} = \max_{p_x} \overline{f^*(x)} = a\mu + b \tag{6}$$

in which the minimax function f^* has the form

$$f^*(x) = [ax + b]^-. \tag{7}$$

To complete the specification of f^*, it is only necessary to select from all pairs (a, b), such that $[ax + b]^-$ satisfies C, the pair of constants for which $a\mu + b$ is a minimum.

The proof of the theorem is contained in the Appendix. We now apply the theorem to the selection of a prefix code for efficiently encoding an (N, K) histogram. In particular, we wish to minimize the maximum required storage

$$S = \max_{\{n_i\}} \{f(n_1) + f(n_2) + \ldots + f(n_{K-1})\} \tag{8}$$

in which, as before, $f(n)$ is the length of the code word assigned to the number n, $0 \leq n \leq N$. A necessary and sufficient condition for a binary prefix code to exist is that the Kraft inequality[5],[6] be satisfied

$$\sum_{n=0}^{N} 2^{-f(n)} \leq 1. \tag{9}$$

We shall work with the slightly weaker sufficient condition

$$\sum_{n=0}^{\infty} 2^{-f(n)} \leq 1 \tag{10}$$

which permits code words to be assigned to *all* non-negative integers. A function f which satisfied (10) is said to be in class C. Note that if g is in C and if $f(n) \geq g(n)$, $n = 0, 1, 2, \ldots$, then f is in C.

As the next step in applying the theorem, recall that

$$n_1 + n_2 + \ldots + n_{K-1} \leq N$$

with equality if and only if $n_K = 0$. Hence

$$\sum_{j=0}^{\infty} jp_j = \mu \leq N/(K-1)$$

in which

$$p_j = \frac{\text{number of } n_i\text{'s equal to } j \text{ exclusive of } n_K}{K - 1}. \tag{11}$$

Hence

$$0 \leq p_j \leq 1; \quad \sum_{j=0}^{\infty} p_j = 1.$$

Thus, we can think of p_j as the probability that a random variable x takes on the value j. The code selection problem may then be phrased: select f in class C as defined by (10) to minimize S, where

$$S = \max_{\{p_j\}} (K-1) \sum_{j=0}^{\infty} p_j f(j) = (K-1) \max_{p_x} \overline{f(x)} \tag{12}$$

and where p_x is such that

$$\bar{x} = \mu \leq N/(K-1). \tag{13}$$

The only thing that now prevents us from applying the theorem and concluding that the optimum f has the form

$$f = [ax + b]^-$$

is the restriction imposed on p_x by (11), that $P[x = j] = p_j$ must be an integer multiple of $1/(K-1)$.

Thus, although the theorem does not prove that $f(x) = [ax + b]^-$ is minimax for histogram encoding, it strongly suggests that such a code length distribution is good. The authors conjecture that this class of linear prefix codes is optimum. In the remainder of this paper, we therefore restrict attention to linear staircase functions; in particular, we restrict attention to the function

$$f(x) = \left[\frac{x}{m}\right]^- + b \tag{14}$$

where b and m are integers. As we shall see, this choice of code permits an especially simple implementation and the cost in terms of storage is small.

IV. Parameters of Code

Since $f(x) \leq \frac{x}{m} + b$, the maximum value of S, say S_p, is easily bounded.

$$S_p = (K-1) \max_{p_x} \overline{f(x)}$$
$$\leq (K-1)\left(\frac{\bar{x}}{m} + b\right)$$
$$\leq \frac{N}{m} + (K-1)\, b.$$

Since S_p is an integer, a tighter upper bound is $[N/m]^- + (K-1)b$. This value is achieved when $n_1 = N$ and $n_2 = n_3 = \ldots = n_K = 0$. Thus, the maximum storage for the linear prefix code of (14) is

$$S_p = \left[\frac{N}{m}\right]^- + (K-1)b. \tag{15}$$

We now select the integers m and b to minimize S_p subject to the condition, (10), guaranteeing the existence of a prefix code. For the assignment of (14), the condition becomes

$$m[2^{-b} + 2^{-(b+1)} + 2^{-(b+2)} + \ldots] \leq 1$$

or

$$m2^{-b} \leq \frac{1}{2}. \tag{16}$$

For fixed N, K, and b, we see from (15) that S_p is nonincreasing with increasing m. We therefore take (16) to be satisfied with the equal sign

$$m = 2^{b-1}. \tag{17}$$

For $b \geq 1$, m is an integer whenever b is an integer. Since the converse is not true, for purposes of optimization we hereafter consider b to be the independent variable with m determined by (17).

From (15)

$$\frac{N}{m} - 1 + (K-1)b < S_p \leq \frac{N}{m} + (K-1)b. \tag{18}$$

It is reasonable, therefore, to select b to minimize $[(N/m) + (K-1)b]$. Treating b as a continuous variable and holding N and K fixed, we have

$$\frac{d}{db}\left[\frac{N}{m} + (K-1)b\right] = -\frac{N}{m^2}\frac{dm}{db} + (K-1).$$

However, from (17),

$$\frac{dm}{db} = (\ln 2)2^{b-1} = m/\log_2 e.$$

Thus, the derivative is zero whenever

$$-\frac{N}{m \log_2 e} + (K-1) = 0. \tag{19}$$

Furthermore, the second derivative is

$$\frac{d^2}{db^2}\left[\frac{N}{m} + (K-1)b\right] = \frac{N}{m(\log_2 e)^2} > 0$$

so that the function is concave $\cup$.[3] We conclude from (17) and (19) that $[(N/m) + (K-1)b]$ is minimized by setting $b = b_0$ where

$$b_0 = \log_2\left(\frac{N}{K-1}\right) + 1 - \log_2 \log_2 e. \tag{20}$$

Since b_0 is not generally an integer, it is necessary to round up or down to obtain an integer, the choice depending upon the resulting values of S_p. The concavity of $[(N/m) + (K-1)b]$ guarantees that one of these two choices is optimum. The concavity also guarantees that an upper bound to S_p is achieved by substituting $b = b_0 + 1$ in the upper bound of (18). The result is

$$S_p \leq (K-1)\left(\log_2 \frac{N}{K-1} + 2.2\right). \tag{21}$$

Let us determine the parameters of a particular code. In one of the examples of Section I, we considered $N = 1024$, $K - 1 = 256$. For those values, we get from (20)

$$b_0 = \log_2\left(\frac{1024}{256}\right) + 0.47 = 2.47.$$

From (15), we find that $S_p = 1024$ for both $b = 2$ and $b = 3$. This storage compares favorably with $S_{\min} = 919$, and with $S = 2560$ and 1280 for the two schemes mentioned in Section I. Note that the JPL code described in Section I, in which the number of samples in the kth cell is indicated by the number of zeros contained between the $(k-1)$st and kth one is a special case of the codes described here for which $b = 1$; such linear prefix codes are optimum when $N/(K-1) < 2$.

[3] Here and in Appendix A we use $\cup$ and $\cap$ to denote a function which is concave (second derivative is non-negative) and convex (second derivative is non-positive), respectively.

V. Comparative Performance of Codes

In Section II, we showed that the minimum storage required for an (N, K) histogram, $S_{\min}$, is approximated for $N \gg K \gg 1$ by

$$S_{\min} \approx S^{**} = (K-1)\left(\log_2 \frac{N}{K-1} + \log_2 e\right) - \frac{1}{2}\log_2(K-1).$$

In Section IV, we observed that the storage required by the linear prefix code was bounded by

$$S_p \leq (K-1)\left(\log_2 \frac{N}{K-1} + 2.2\right).$$

Comparing S^{**} and S_p, we see that the prefix codes here described have a storage requirement which differs from the minimum by approximately $(K-1)(2.2 - \log_2 e)$ or $3/4\,K$ bits.

In Figs. 1 and 2, $S_{\min}$ and S_p are compared. Figure 1 shows $S_{\min}$ and S_p as functions of N for several values of K while Fig. 2 shows both as functions of K for several values of N. We see that $S_{\min}$ and S_p are quite close and that, for given values of K and N, their difference is approximately $3/4\,K$ as predicted.

VI. Implementation

By restricting attention to linear prefix coding, we suffer a storage penalty on the order of $3/4\,K$ bits. In accepting this suboptimality, we generate codes which are extremely simple to implement and thus interesting for practical applications. In this section one method of carrying out the implementation is discussed.

We first specify the code for a given b and $m = 2^{b-1}$. Recall that, according to (14), the code words assigned to $0, 1, \ldots, m-1$ are of length b; those assigned to $m, m+1, \ldots, 2m-1$ are of length $b+1$; those assigned to $2m, 2m+1, \ldots, 3m-1$ are of length $b+2$; and so on.

A prefix code is perhaps most easily visualized as a tree (see Fano[6]). A linear prefix code with $b = 3$ and $m = 2^{b-1} = 4$ is presented in Fig. 3. Each terminal node is assigned an integer shown in parentheses. The sequence of branches leading from the tree root to terminal node n specifies the code word assigned to n by the convention that a branch with positive slope denotes 0 and a branch with negative slope denotes 1. The code words for each integer are also given in the figure. The generalization of this code to other values of b is straightforward. It is obvious from the construction that no code word is the prefix of any other.

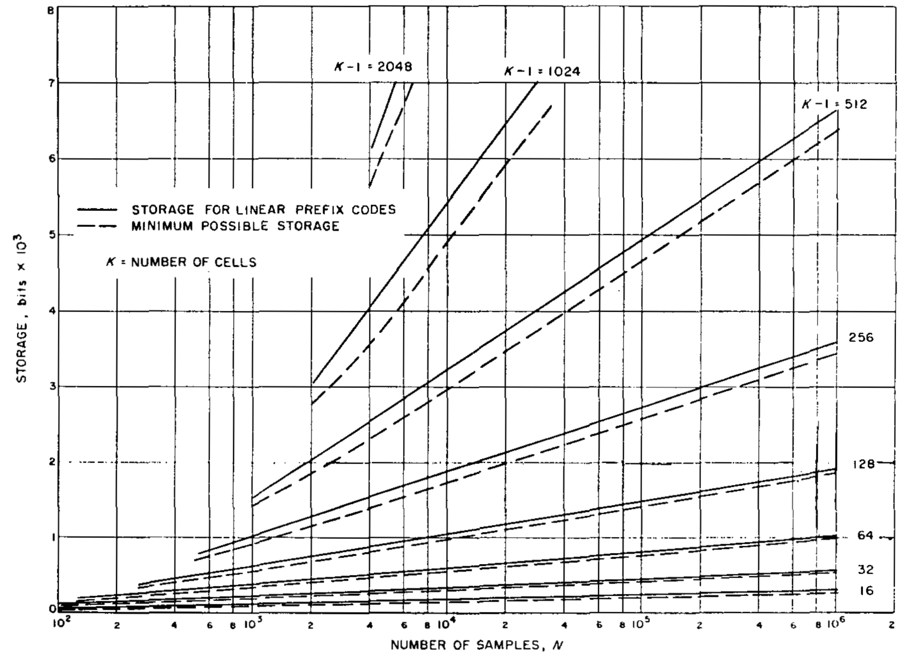

Fig. 1. Storage requirements for linear prefix codes compared to theoretical minimum.

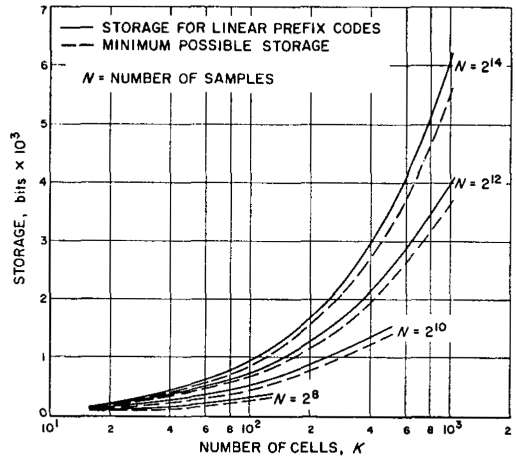

Fig. 2. Storage requirements for linear prefix codes compared to theoretical minimum.

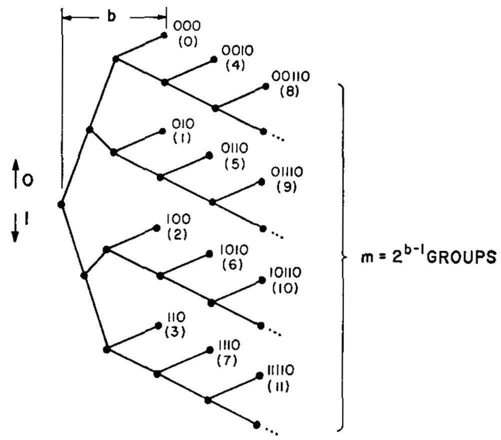

Fig. 3. The linear prefix codes represented as a tree, $b = 3$, $m = 4$.

Implementation of this code is facilitated by observing that each code word has three distinct segments. Consider the code word assigned to n. The first segment, consisting of $b - 1$ digits, is the binary representation of the number m modulo 2^{b-1}. The second segment is a sequence of t 1's where

$$t = \left[\frac{n}{2^{b-1}}\right]^{-}.$$

Note that if n 1's are fed into a binary counter with $b - 1$ stages, the counter overflows exactly t times and the resulting count is n modulo 2^{b-1}. The third segment for every code word is a single 0, indicating the end of the code word. For example, if $b = 4$, the code word for 17 is 0 0 1 1 1 0 while the code word for 85 is 1 0 1 1 1 1 1 1 1 1 1 1 1 0.

In Fig. 4 we diagram one method for generating the coded histogram. The basic storage unit is an S-bit shift register that is initially set to all zeros. Each time a sample arrives from the experiment generating the histogram, it is assigned a histogram cell. The number k of this cell is stored in the *New Cell Number* register and a start pulse given. The start pulse serves to set flag x, to enter zero into the *cell counter*, and to start the shift register shifting right. The bit stream from the shift register bypasses the unit delay, enters the 1-stage adder, appears unchanged at the adder sum-output, and is returned to the left end of the shift register. Flag x remains set during the first segment of each code word; following the first $(b - 1)$ bits x is reset to 0 and is not set again until a zero arrives marking the end of the code word. The end-of-word zero also causes the cell counter to be increased by 1.

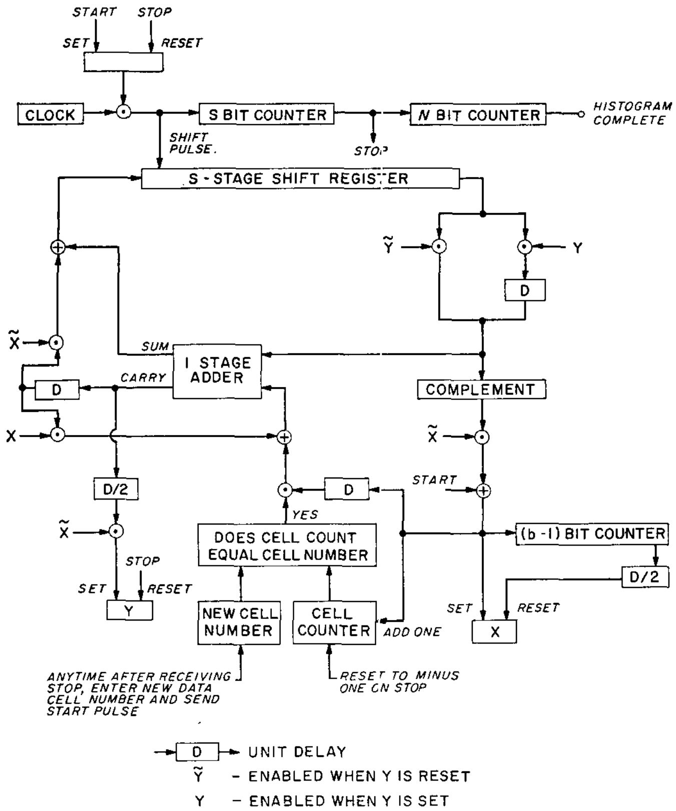

Fig. 4. Implementation of linear prefix encoder.

When the cell count equals the new cell number, a 1 is added, via the carry input of the 1 stage adder, to the least significant digit of the first segment of the codeword representing the present contents of the cell.[4] The updated least significant digit is returned to the S-stage shift register while a carry, if present, is added to the next more significant digit.

This continues until $(b - 1)$ bits have been updated. At this point flag x is reset. If there was no overflow of the adder, that is, no carry from the $(b - 1)$st addition, the remaining contents of the shift register pass unchanged through the adder and back to the shift register. If there is a carry, however, an additional 1 must be inserted into the second segment of the code word. This is accomplished by setting flag Y and, after a unit delay, injecting a 1 (from the carry output) into the bit stream returning to the shift register. Flag Y causes a unit delay to be inserted into the bit stream coming from the shift register, thereby maintaining the proper bit spacing.

After S shifts, the updating of the histogram is complete. We then wait for a new experimental result and repeat. After N such cycles, the histogram is completely assembled and the coded form is ready for any processing or transmission.

VII. Conclusion

A method for assembling and storing histograms using prefix codes has been described. The method is rather easily implemented, requires little more storage than that absolutely necessary for uniquely specifying an (N, K) histogram, and permits construction of the histogram as data arrives.

The theorem used to justify the linear staircase assignment of code word lengths, $f(n) = \left[\frac{n}{m}\right]^- + b$, did not prove that this choice was optimum. Whether another assignment of lengths for the prefix code can reduce storage further is an open question.

[4] For the implementation, the order of the digits in the first segment of the code word is reversed, with the least significant of the $b - 1$ digits appearing first.

Appendix
Proof of Theorem

Let f be an integer-valued function in class C. Since $\overline{f(x)}$ depends only on the values $f(n)$, $\{n = 0, 1, 2, \ldots, N\}$, we may ignore the values of f for noninteger arguments. Define $h(x)$, $0 \leq x \leq N$, as the convex $\cap$ hull of the sequence of samples $\{f(0), f(1), f(2), \ldots, f(N)\}$ (see Fig. 5 and Wozencraft and Jacobs[7] Appendix 7B). Since $h(n) \geq f(n)$, $n = 0, 1, 2, \ldots, N$, we have

$$\max_{p_x} \overline{f(x)} \leq \max_{p_x} \overline{h(x)}. \tag{22}$$

But, since $\bar{x} = \mu$ and h is convex $\cap$,

$$\max_{p_x} \overline{h(x)} = h(\mu). \tag{23}$$

Let N_1 be the largest integer less than μ for which $f(N_1) = h(N_1)$ and let N_2 be the smallest integer greater or equal to μ for which $f(N_2) = h(N_2)$. Then, h is a segment of a straight line between N_1 and N_2; say

$$h(x) = ax + b, N_1 \leq x \leq N_2. \tag{24}$$

A function which achieves the maximum in (23) is [5]

$$p_x(\alpha) = p\delta(\alpha - N_1) + (1 - p)\delta(\alpha - N_2) \tag{25}$$

in which p is chosen so that

$$\bar{x} = pN_1 + (1 - p)N_2 = \mu. \tag{26}$$

As a check, note that for this density function

$$\overline{h(x)} = a\bar{x} + b = a\mu + b = h(\mu).$$

Since for this density function $\overline{f(x)} = \overline{h(x)}$, we conclude that

$$\max_{p_x} \overline{f(x)} = a\mu + b. \tag{27}$$

Next define

$$f^*(x) = [ax + b]^-. \tag{28}$$

Then (see Fig. 5)

$$f^*(n) \geq f(n), n = 0, 1, 2, \ldots, N \tag{29}$$

[5] Here, $\delta(t)$ is the unit impulse function, or delta function. (See Papoulis.[8])

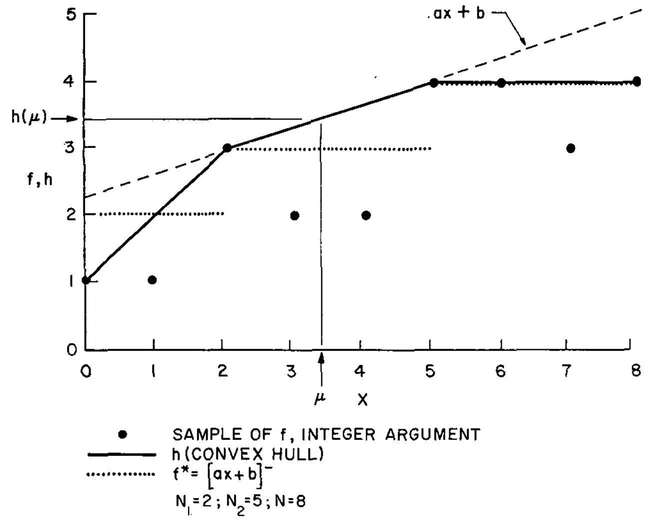

Fig. 5. Convex hull of samples of $f(x)$.

so that f^* is also in class C. Furthermore, for all p_x with $\bar{x} = \mu$,

$$\overline{f^*(x)} \leq \overline{ax + b} \qquad (30)$$
$$= a\mu + b,$$

and equality is achieved for the density function of (25).

We conclude that for every function f in C there is a function f^* of the form $[ax + b]^-$ in C which has the same maximum as f, namely, $a\mu + b$. It then suffices to search for the minimax function f^* from among functions of the form $[ax + b]^-$ in C.

Acknowledgment

The authors gratefully acknowledge the motivation and several suggestions provided by W. Tveitan and W. A. Lushbaugh of the Jet Propulsion Laboratory.

References

[1] E. C. Posner, "The use of quantiles for space telemetry data compression," *Proc. 1964 Nat'l Telemetry Conf.*
[2] Jet Propulsion Laboratory, California Institute of Technology, Pasadena, Calif., Space Program Summary 37-27, vol. III, pp. 103–112.
[3] G. A. Gordon, "An algorithmic code for monotonic sources," private communication.
[4] E. N. Gilbert, "Synchronization of binary messages," *IRE Trans. on Information Theory*, vol. IT-6, pp. 470–477, September 1960.
[5] L. G. Kraft, "A device for quantizing, grouping, and coding amplitude modulated pulses," S.M.E.E. thesis, Massachusetts Institute of Technology, Cambridge, Mass., March 1949.
[6] R. M. Fano, *Transmission of Information.* New York: Wiley, 1961.
[7] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering.* New York: Wiley, 1965.
[8] A. Papoulis, *The Fourier Integral and Its Applications.* New York: Wiley, 1960.

Irwin M. Jacobs (S'55–M'60) was born in New Bedford, Mass., on October 18, 1933. He received the B.E.E. degree from Cornell University, Ithaca, N. Y., in 1956, and the S.M. and Sc.D. degrees from the Massachusetts Institute of Technology, Cambridge, in 1957 and 1959, respectively. He was the recipient of a McMullin Regional Scholarship and a General Electric Teachers Conference Scholarship at Cornell, and participated in the engineering cooperative program in association with the Cornell Aeronautical Laboratory, Buffalo, N. Y. In graduate school, he was a General Electric Fellow and an Industrial Fellow of Electronics.

In 1959, he was appointed an Assistant Professor of Electrical Engineering at M.I.T. and a member of the staff of the Research Laboratory of Electronics. He was promoted to Associate Professor in 1964. On leave from M.I.T., he spent the academic year 1964–1965 as a NASA Resident Research Fellow at the Jet Propulsion Laboratory, Pasadena, Calif., and was concerned principally with coding for deep-space communications. In 1966, he accepted an appointment as Associate Professor of Applied Electrophysics at the University of California, San Diego, Calif., and is currently working in the area of information and computer science.

Dr. Jacobs is a member of Phi Kappa Phi, Sigma Xi, Eta Kappa Nu, and Tau Beta Pi, and the Association for Computing Machinery.

Leonard Kleinrock (S'55–M'64) was born in New York, N. Y., on June 13, 1934. He received the B.E.E. degree from the College of the City of New York, N. Y., in 1957, and the S.M.E.E. and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in 1959 and 1963, respectively, while participating in the Lincoln Laboratory Staff Associate Program.

From 1951 to 1957, he was employed at the Photobell Company, Inc., New York, N. Y., an industrial electronics firm. He spent the summers from 1957 to 1961 at the M.I.T. Lincoln Laboratory, Lexington, Mass., first in the Digital Computer Group and later in the Systems Analysis Group. At M.I.T., he was a research assistant, initially with the Electronic Systems Laboratory, and later with the Research Laboratory for Electronics where he worked on communication nets in the Information Processing and Transmission Group. After completing his graduate work at the end of 1962, he worked at Lincoln Laboratory on communication nets and on signal detection. Since 1963 he has been an Assistant Professor of Engineering at the University of California at Los Angeles. His main interests are in communication nets, data compression, priority queueing theory, and theoretical studies of time-shared systems.

Dr. Kleinrock is a member of Tau Beta Pi, Eta Kappa Nu, Sigma Xi, and the Operations Research Society of America.