IX. COGNITIVE INFORMATION PROCESSING

Academic and Research Staff

Prof. M. Eden
Prof. J. Allen
Prof. B. A. Blesser
Prof. T. S. Huang
Prof. F. F. Lee
Prof. S. J. Mason
Prof. W. F. Schreiber
Prof. D. E. Troxel
Prof. I. T. Young
Dr. D. M. Ozonoff
Dr. O. J. Tretiak
Dr. T. Yamamoto
Dr. M. Yoshida
F. X. Carroll
Deborah A. Finkel
E. R. Jensen
M. Sharon Kohl
R. A. Piankian

Graduate Students

J. Alba
J. C. Borum
J. E. Bowie
B. E. Boyle
P. Coueignoux
C. H. Cox
J. L. Davis
C. J. Drake
J. R. Ellis, Jr.
I. S. Englander
A. M. Fakhr
S. G. Finn
T. Fukuda
D. W. Hartman
M. E. Jernigan
C. K. S. Kau
J. W. Klovstad
T. T. Kuklinski
Y.-T. Lin
C. W. Lynn
H. S. Magnuski
L. C. Makowski
P. L. Miller
B. Minkow
J. A. Myers
D. O'Shaughnessy
J. E. Ostrowski
R. S. Putnam
R. J. Shillman
D. G. Sitler
J. R. Sloan
A. A. Smith
R. D. Solomon
W. D. Stratton
H-m. D. Toong
J. M. Tribolet
B-K. Tye
K. P. Wacks
J. E. Walker
A. W. Wieger
J. A. Whealler, Jr.
H. M. Wolfson

A. USE OF AMBIGUOUS CHARACTERS IN MEASURING FUNCTIONAL INVARIANTS

National Science Foundation (Grant GK-33736X)

R. J. Shillman, B. A. Blesser

Humans are remarkably versatile with regard to the recognition of hand- or machine-printed letters. More than 100 fonts (styles of type) are in common use throughout the United States,[1] and even letters in new or unusual fonts can be recognized with very little effort (see Fig. IX-1). When hand-printed (script) letters are also considered, we realize that humans can identify letters in an infinite variety of styles and variations.

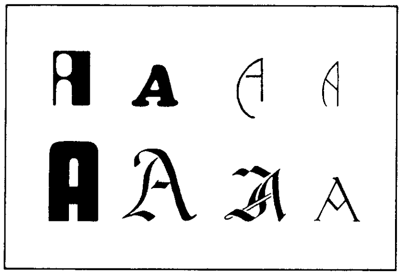

Fig. IX-1. Various styles of the printed letter A.

Many computer algorithms have been developed to recognize machine-printed characters; all are alike, in that they make identifications based upon measurements of various physical components of the unknown characters. The algorithms lack generality in that they perform well only on those fonts that they were specifically designed to read. This specificity to type style is due to the nature of the algorithms that were developed in the absence of knowledge of meaningful descriptions of letters. One possible technique for broadening machine ability in the recognition of characters (without increasing memory size proportionately) is to devise a general letter-recognition program based upon the rules that humans use.

During the past year we have attempted to uncover the critical functional attributes used by humans in the recognition of upper-case English letters of the Roman alphabet. The fact that we can identify letters of a style that we have never previously seen is evidence against a model of recognition based on specific traces or templates and is evidence for a model based on a set of characteristic features.

We recognize all of the characters in Fig. IX-1 as the letter A, even though these specific styles have not been seen before. Although these characters have very little in common physically, it is clear that functional invariants (attributes) must be present because they are all easily categorized as "A". (This generalization is not always true; the Roman numeral II has nothing in common with Arabic numeral 2, even though they both stand for the same thing. In Fig. IX-1, it is clear that simple physical transformations will traverse the given space and that the transformed characters will still be recognizable as members of that space, whereas there is no recognizable character midway between "II" and "2".) We refer to these functional invariants as *discriminata* in contrast to the physical similarities that are called *features*.

Insight into the nature of discriminata can be obtained through the use of ambiguous characters. An ambiguous character is one that is perceptually midway between two letters (it is identified with equal probability as being either of the two letters).

The dimensions in perceptual space along which changes must be made in order to alter the identity of an ambiguous letter are the discriminata involved in resolving the ambiguity. These discriminata are at least some of the functional attributes required for letter recognition. The following examples help to clarify the distinction between features and discriminata.

At first, one might think that the features (physical attributes) of upper closure and two descending legs describe A-ness, but clearly these features are insufficient, because the character R would not be categorized as an A. Also, the hand-printed character A lacks an upper closure (physically) but is still recognized as an A, even though it

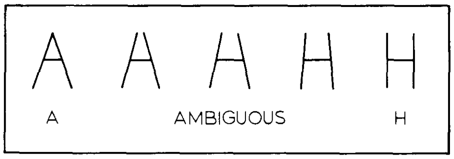

Fig. IX-2. Elements of the A-H continuum.

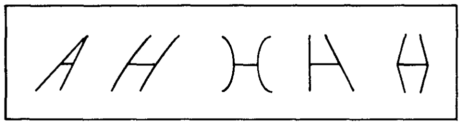

Fig. IX-3. Various examples of A and H.

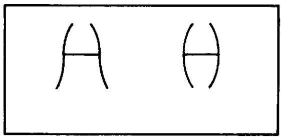

Fig. IX-4. Closure vs symmetry.

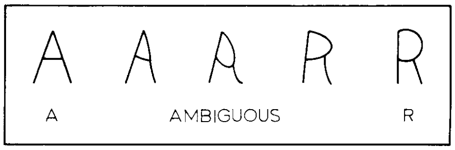

Fig. IX-5. Elements of the A-R continuum.

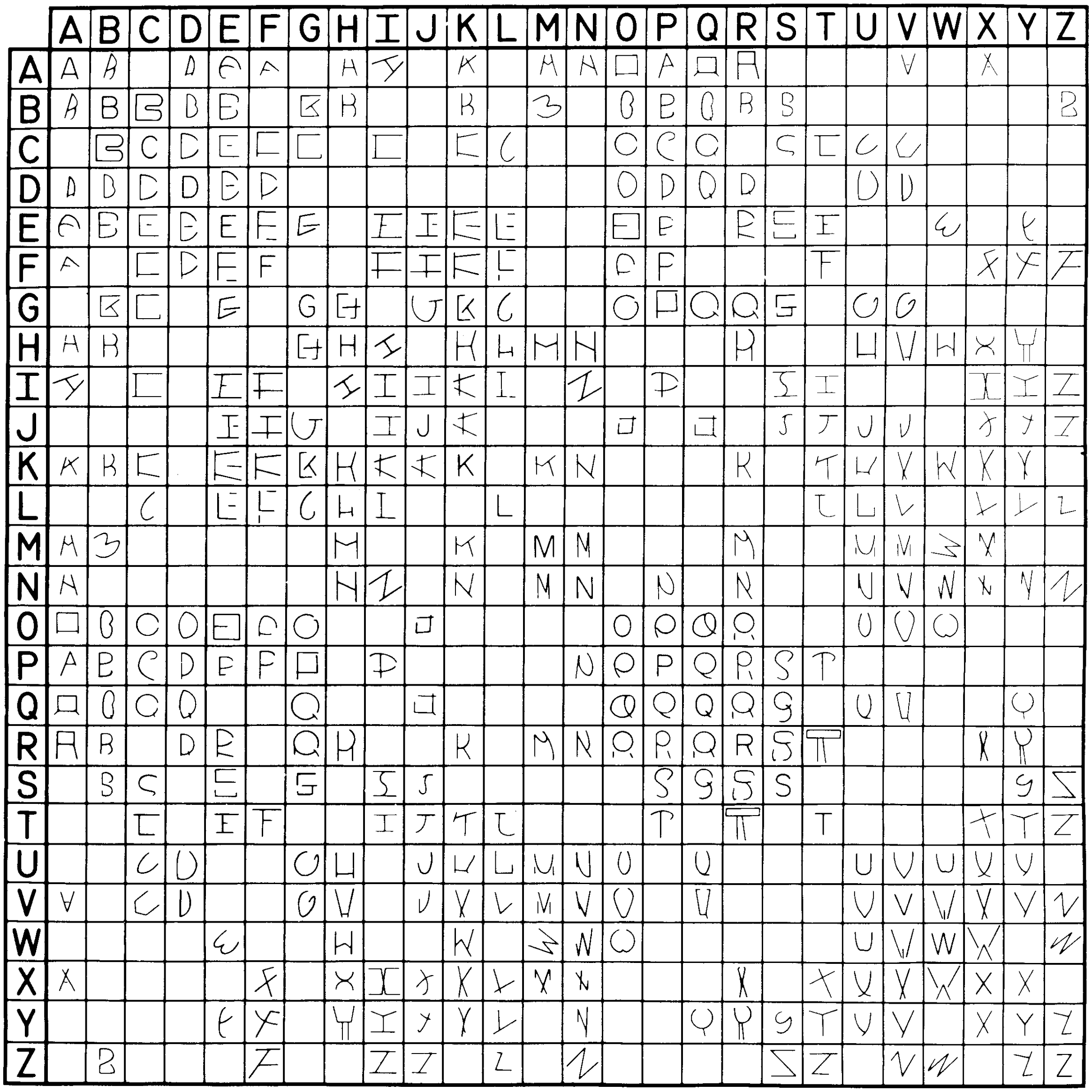

Fig. IX-6. Ambiguous characters.

conforms closely to the physical description of the letter H.

A portion of the A-H continuum is shown in Fig. IX-2. Initially, it appears that the angles of the diagonals are the determining features. How, then, do we decide upon the identity of the characters shown in Fig. IX-3? It is clear that characters composed of straight lines form a small subset of all possible characters and therefore rules based upon more general concepts than angularity need to be developed. With regard to the A-H confusion, we initially concluded that if the top of the character could be closed (functionally), then the character would be identified as an A. We were satisfied with closure as the discriminata until we drew the characters shown in Fig. IX-4. The top portions of both characters are identical, yet the first character is easily recognized as an A, whereas the second character is closer to an H. It appears, then, that at least two discriminata, *closure* and *symmetry*, are necessary to resolve fully the A-H ambiguity.

Again, discriminata does not refer to a physical attribute. The symmetry with which we are dealing here is not geometric symmetry, as can be seen by the second character in Figure IX-3. Although this introspective method yields useful results, we can never know when the list of discriminata is complete; we can only amend the list as novel ambiguities (varying along different functional dimensions) are discovered.

When the A-R continuum shown in Fig. IX-5 was studied in a similar manner, we found that discrimination is based upon *symmetry* *of* *intersection*. The letter A is characterized by the perception of symmetrical intersections on the right and left sides; for example, ⊢ ⊣ for A as opposed to ⊢ ⋏ for R.[2]

From the two previous examples, it is clear that ambiguous characters are useful in determining the dimensions along which letters are perceived. Figure IX-6 shows a matrix of possible ambiguities. The matrix is symmetric and contains 180 different ambiguous characters. We are studying these ambiguities, and the perturbations that resolve them, in an attempt to uncover the underlying functional attributes necessary for letter recognition.

References

1. G. P. Marston III and L. P. A. Henckels, "Survey of Type Font Usage," Quarterly Progress Report No. 95, Research Laboratory of Electronics, M. I. T., October 15, 1969, pp. 105-107.
2. T. W. Sefkow, "Symmetry as an Aspect of Character Recognition," S. B. Thesis, Department of Electrical Engineering, M. I. T., January 1973.

B. PSEUDORANDOM NOISE GENERATION FOR USE IN IMAGE TRANSMISSION

Associated Press (Grant)

K. P. Wacks

Pseudorandom noise [PRN] generation is an important component in many digital image-transmission systems.[1-4] This report considers the required characteristics for PRN and methods for generating it.

The initial application of PRN to picture coding was developed by Roberts[5] in order to reduce the visibility of artificial contours that appear when a sampled picture is quantized to fewer than ~6 bits/sample. He removed this structured distortion and replaced it with unstructured noise, independent of the picture data, by adding PRN to each sample before quantizing and then subtracting the same sequence of PRN at the receiver. It was required that the noise be uniformly distributed over ±.5 of a coarse quantization level and statistically independent on a sample-by-sample basis. The last requirement implies a flat power spectrum. Other applications of PRN with a modified spectrum[2] can be implemented by starting with a sequence that has a flat power spectrum.

A particularly convenient scheme for generating random numbers in a computer is the power-residue method.[6] Each successive random number is obtained from the low-order b bits of the product of the "seed" and the previous number. The appropriate seed depends on b, which usually corresponds to the computer word size. The initial value is any odd b-bit number, and 2^{b-2} terms are produced before the sequence repeats. This PRN generator was implemented on a PDP-9 computer with $b = 18$. The highest 6 bits from each sample were retained to produce integers ranging from -32 to $+31$. The histogram of 1024 values had an rms deviation of 3.5 from the computed (and expected) mean of 16 samples/integer in the range generated.

The power density spectrum of this sequence was calculated by the following method for frequencies up to 256 cycles/dimension (the highest frequency that can be represented by a picture composed of 512×512 samples). We compute the unbiased autocovariance estimate[7] $a(k)$ of the noise $x(k)$.

$$a(k) = \frac{1}{1024 - |k|} \sum_{r=1}^{1024-|k|} x(r)\, x(r+k) - \left[\frac{1}{1024} \sum_{r=1}^{1024} x(r) \right]^2 ,$$

where k ranges from -255 to $+256$. Then we window the estimate with a Gaussian function having 90% of its area between $k = \pm 64$. The estimate of the power density spectrum is the discrete Fourier transform of the windowed autocovariance estimate. It is evident from Fig. IX-7a that the spectrum is relatively flat.

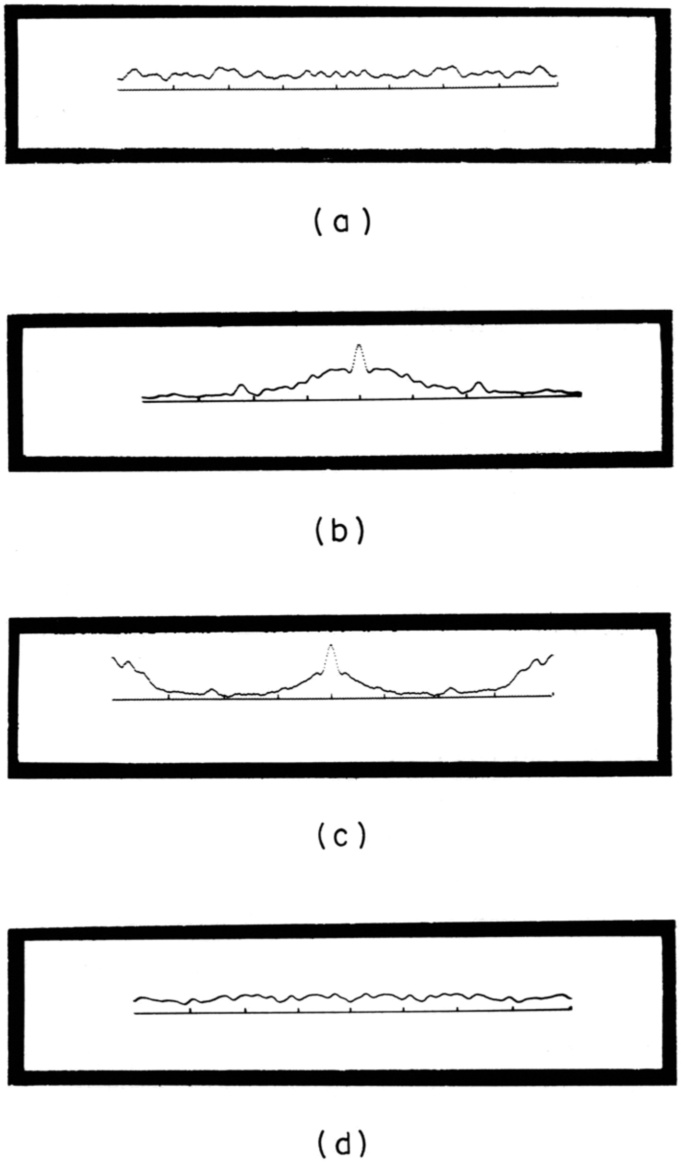

Fig. IX-7.

Power density spectra. Horizontal axis range: -255 to +256 cycles/dimension; dc at center. Range of values in each trace: (a) 87-186, (b) 38-480, (c) 27-474, (d) 83-169.

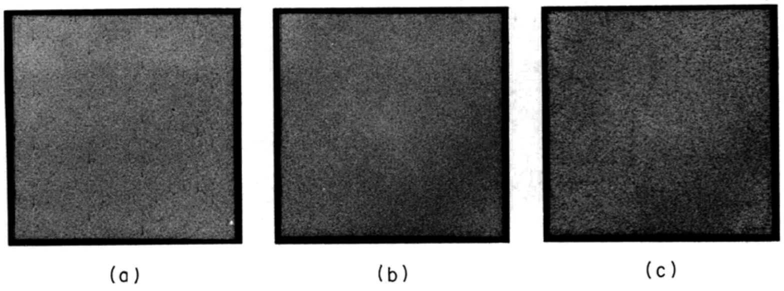

Fig. IX-8. Methods for generating noise: (a) power residue, (b) power residue with extra cycling after each line, (c) shift register.

A flying-spot scanner display was used to convert numbers to brightness on a linear scale (0 black and 255 white). In Fig. IX-8a each of 512 × 512 samples represents 128 + 4 · (noise value). Since the apparent randomness of the noise is the critical factor in picture coding, the vertical stripes shown in Fig. IX-8a make this method unacceptable, especially for a facsimile transmission system. This problem was eliminated without altering the shape of the noise spectrum by generating 5 extra random numbers at the end of each row, as shown in Fig. IX-8b.

If an image-transmission system is to be implemented in special-purpose hardware, there is a more economical PRN generator than the power-residue method. This generator utilizes a shift register with feedback of the exclusive OR of selected cells to simulate an irreducible primitive polynomial.[8] For an 18-bit shift register the simplest arrangement is the exclusive OR of cells 1 and 8 to feed cell 18, which was simulated on the PDP-9 computer. The computed power density spectrum in shown in Fig. IX-7b. The high-frequency roll-off occurred because the register was shifted once between samples so that the low-order bits remained correlated from point to point. Figure IX-8c was generated by using these values. Comparison with Fig. IX-8b demonstrates the correlation of the samples. To decorrelate the bits composing the integers, every other bit was retained after each shift. As shown in Fig. IX-7c, this did not solve the problem. The relatively flat spectrum of Fig. IX-7d was finally produced by shifting the register 6 times between samples composed of the high-order bits. The corresponding noise picture appeared to be similar to Fig. IX-8b. (Calculations on the histogram yielded an rms deviation of 4.0 from a mean of 16.)

This study illustrates the importance of exploiting the inherent pattern-recognition capabilities of the human visual system when designing image-processing components. Not only can design calculations be conveniently corroborated, but unnecessary computation can be avoided if a proposal is unacceptable visually.

References

1. T. G. Stockham, "Intra-frame Encoding for Monochrome Images by Means of a Psychophysical Model Based on Nonlinear Filtering of Multiplied Signals," Symposium on Picture Bandwidth Compression, Massachusetts Institute of Technology, Cambridge, Massachusetts, April 4, 1969.
2. J. O. Limb and T. W. Mounts, "Digital Differential Quantizer for Television," Bell System Tech. J. 487, 2583-2599 (1969).
3. J. E. Thompson, "A 36-Mbit/s Television Coder Employing Pseudorandom Quantization," IEEE Trans., Vol. COM-19, No. 6, pp. 872-879, December 1971.
4. K. P. Wacks, "Analyzing the Performance of an Image-Processing System," E. E. Thesis, Department of Electrical Engineering, M. I. T., May 1970.
5. L. G. Roberts, "Picture Coding Using Pseudorandom Noise," IRE Trans., Vol. IT-8, No. 2, pp. 145-154, February 1962.
6. IBM, "Random Number Generation and Testing," GC20-8011-0, December 1969.

7. G. M. Jenkins and D. G. Watts, Spectral Analysis and Its Application (Holden Day, San Francisco, 1968).
8. W. W. Peterson, Error Correcting Codes (The M. I. T. Press, Cambridge, Massachusetts, 1961).

C. MINICOMPUTER INTERFACES

National Institutes of Health (Grant 1 PO1 GM19428-01)
Associated Press (Grant)

D. E. Troxel

1. Introduction

It is often desirable to interface equipment to a minicomputer. Although occasionally we wish to interface to large computers, we are much more apt to be concerned with small computers because interfacing is easier, it may be an economic necessity, and minicomputers are often incorporated within a solution to a problem. With the exception of data sets, CRT terminals, or teletypes, each interfacing problem is somewhat unique, in that an existing design is rarely copied verbatim. Fortunately, with respect to CRT terminals, the industry has adhered more or less to a set of standards,[1] the RS-232. It would obviously be desirable to have a standard for other types of interfacing requirements. The adoption of almost any standard would result in greater ease of interconnection of a computer with various peripherals, and computers with other computers. With a standardized interface, the peripherals, which are often much more expensive than the computer, could be reused when the main computer is replaced by a newer model, or shared when an additional minicomputer is acquired. This standard interface would also facilitate maintenance because malfunctions could easily be located by plugging the peripheral into a separate-but-equal standard interface. Now there is often a plethora of minicomputers and a dearth of peripherals, primarily because of the relative costs of these two classes of items. This is particularly annoying because many of the peripherals are used infrequently with a computer. For example, a line printer is used or desirable for producing assembly listings, but this takes a relatively small percentage of the total computer time. If several minicomputers and one line printer were in close proximity, and the line printer could conveniently be plugged into any one of the minicomputers, then users would find little inconvenience in arranging to "time share" the line printer. This arrangement would be far better than having the line printer interfaced to one computer, leaving three others without a line printer, since this would result in having a line of users waiting to use the computer with the line-printer capability, and hence in unused computer capacity. Of course, large time-sharing systems provide another solution to this problem by queuing the files to be printed on a mass-storage device. Adoption of a standardized interface would permit a cheaper, albeit somewhat less powerful, solution for this and similar problems.

2. Serial or Parallel Interface

If we agree that a standardized interface is desirable, we must specify the electrical and mechanical properties of such an interface. Basically, we have two choices, serial or parallel interfaces. Uniformly, computer manufacturers have chosen the parallel interface. They have made this choice primarily because the data source and destination (i.e., the computer and peripherals) are normally organized in a parallel manner. It has appeared to these manufacturers that to impose a serial interface between two parallel devices would be adding unnecessary electronic complexity. There is, however, a price that must be paid for this "simplicity," namely, the multiplicity of cable drivers and receivers and the added expense of multiconductor cables and connectors.

The alternative of a serial interface reduces the number of cable drivers and receivers to a minimum and greatly simplifies the connector and cabling problem, both with regard to reliability and cost. A serial interface requires additional electronic complexity, but the availability of medium- and large-scale integrated circuits has, or soon will, ameliorate this disadvantage. The use of a serial interface simplifies the problem of compatibility of communication between computers and peripherals with different word lengths. Of course, one could design a standard parallel interface which would also solve any compatibility problems, but this has not been done and, I suspect very strongly, will not be done. The subject of compatibility will be considered in detail.

Most parallel interfaces have a constraint on the length of the cabling allowed, either for reasons of timing or for noise considerations. With a serial interface, it is both practical and reasonable to employ appropriate transmission-line drivers and receivers to allow communication essentially over any distance. If proper attention has been paid to this cable-driving problem, there is considerably less trouble with the problem of cross talk, which is often present with parallel interfaces. Finally, a serial interface is considerably easier to connect and disconnect because the cabling connections are both more rugged and easier to handle. Figure IX-9 shows a patch panel that makes possible interconnection of different computers and peripherals. With the small number of wires to be patched, one might even consider the luxury of an electronic switchboard to automate these interconnections.

3. Serial Interface Specifications

The type of serial interface that we are concerned with is self-synchronizing, similar to that used with a teletype[2] or the RS-232 standards; that is, it includes a start bit, data bits, and optional stop bits, as indicated in Fig. IX-10. Basically, there are 3 parameters specifying the serial interface. The logic levels of the cable driver and

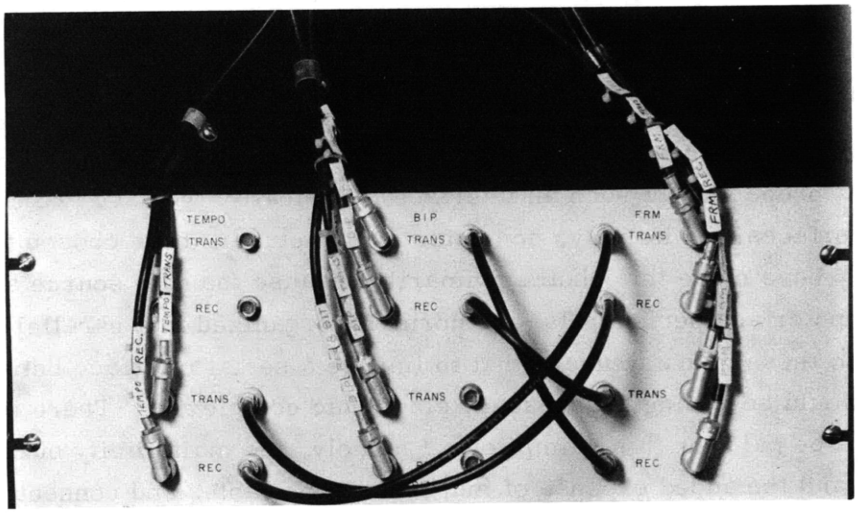

Fig. IX-9. A serial interface patch panel.

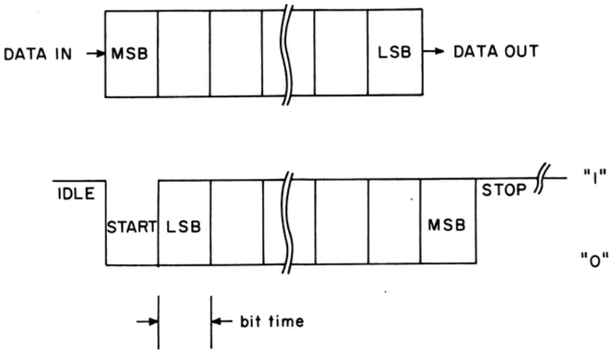

Fig. IX-10. Serial interface format.

receiver have to be compatible. Two reasonable choices are TTL logic levels and the positive and negative levels appropriate to the RS-232 standard. The second parameter that must be common to receivers and transmitters is the bit rate. While bit rates ordinarily used for digital transmission via data sets are possible, there are many circumstances in which it is desirable to have a considerably higher bit rate. Two actual installations, one in private industry and one at M.I.T., employ a bit rate of one megabit per second. This is slow enough so that it is easy to achieve with ordinary logic and so that transmission over suitable cables is not complicated. A familiar coaxial cable, RG 62U, is both convenient and inexpensive. Crimp-on BNC connectors are also extremely rugged and reliable. On the other hand, a 1-megabit rate is high enough so that the transmission time is of little consequence for most peripherals and computers.

It would not be difficult to, say, employ a 10-megabit rate, but it hardly seems worthwhile. On the other hand, a "standard" rate such as 9600 bits (or lower) would result in a more significant transmission time while it would afford little decrease in electronic complexity. The third parameter of a serial interface is the number of data bits. While at first it would appear that the transmitter and receiver should have an identical number of data bits, it is quite feasible to have transmitters of, for example, 18, 16, 12 and 8 bits that are compatible with a peripheral that requires 7 bits of information. Essentially transmitters of any length can communicate with receivers of any length.

4. Compatibility of Different Word Lengths

We first consider the problem of one computer communicating with another. When the output of a 12-bit transmitter is received by an 18-bit receiver, the high-order 6 bits will be read in as the idle condition of the line, that is, as 1's. The receiving computer merely has to AND out the high-order 6 bits in order to interpret properly the data which were sent. In the reverse direction, the sending computer must not attempt to send more bits than can be interpreted by the receiving computer. This can be insured by the sending computer ORing 1's into the high-order bits.

Peripheral interfaces can be designed to be compatible with a range of bit lengths by making the length of the peripheral interface equal to the longest of the possible transmitters. Since it is common practice for short data in a computer to be right-justified, the fact that the data are shifted out of the least significant bit of the word first, insures that the short data will be positioned properly in the peripheral receiver. The normal manner of communication between a computer and a peripheral is to have the computer send a word to the peripheral and to have the peripheral echo this word when it is ready to accept another word. The receipt of the echoed word at the computer is interpreted as the setting of the peripheral ready flag. Occasionally it is desirable that the peripheral notify the computer of some unusual condition such as the fact that a line printer

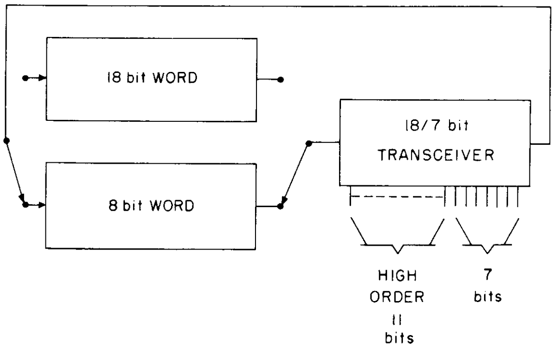

Fig. IX-11. Compatibility of different word lengths.

has run out of paper. In this case, it is convenient for the peripheral to set the sign bit of the computer word, as this is normally an easy thing for the computer program to test. A simple way to insure this capability with a range of possible word lengths is for the peripheral to set all of the unused high-order bits to 1 to signify an unusual condition. In Fig. IX-11 we show a possible arrangement of two different computer word lengths communicating with a peripheral device. When the 18-bit computer is connected, the echoed words have 0's in the high-order 11 bits. When the 8-bit computer is connected, the echoed high-order 10 bits are 1, and the 8th bit is 0. These high-order 10 bits are equivalent to the idle condition of the line for the 8-bit receiver.

5. Programmed Data Transfer vs Direct Memory Access

Since serial data transfers are often thought of in the context of a teletype, the concept of programmed data transfer is often associated with serial interfaces. The essence of a serial interface, however, is unrelated to the question of the use of a programmed data transfer as opposed to a direct memory access arrangement. It will be quite feasible, for example, in the case of a floppy disc, to have all data transfers between the disc control and the computer be of a serial nature. In this case, however, one would probably want to use direct memory access transfers for the reading and writing of records to obviate stringent program timing requirements. If programmed data transfer with a serial interface is not used, then it is convenient to choose IO instructions that are identical to those used for teletype interfaces except for a device code designation (see the appendix). Of course, a serial interface can use the capabilities of whatever interrupt facilities are available on the computer.

6. Peripheral Interfaces

There are many types of peripherals for which serial interfaces are appropriate. These include more-or-less standard computer peripherals such as line printer, magnetic tape, disc, paper-tape reader and punch, xy tablet, remote consoles and plotter. Also appropriate are many less standard interfaces, some of which may be constructed by the user. These include banks of switches and knobs, A-D and D-A converters, automatic controls for audio tape recorders, picture digitizers or scanners, and facsimile transmitters and receivers. For many of the user-constructed equipments, it is very desirable to have a standard interface such as the interface proposed here because of the substantial time and effort expended on development of these equipments. As we have stated, the adoption of a standard interface facilitates the upgrading of computer facilities.

7. Computer-to-Computer Interfaces

Given the availability of serial interfaces on computers, one computer can use another as its peripheral, and two or more computers can easily be interconnected.

Some sample instances in which this has been desirable are described below.

A microscope scanner was constructed with a small minicomputer used essentially as a controller. This small computer had very little core storage and no bulk storage. Two other computers were available which were powerful enough and had enough bulk storage to process the images acquired by the microscope scanner. By equipping all three of these computers with compatible serial interfaces, it has proved possible to patch the microscope scanner computer to either of the other computers so that microscope images could be transferred to these larger computers for further processing.

Another use of serial interface connections showed the fallibility of "Parkinson's law," which states that programs expand to occupy the available space. In a speech-synthesis research program in our laboratory two different portions of the synthesis problem required programs so large that both could not coexist on any available computer. We used two computers, allowing them to communicate via the serial interfaces. It was then possible to demonstrate the speech-synthesis procedures in real time.

Another use of the serial interface arose in an industrial context where a large number of computers was available, at least temporarily, since each manufactured product included a computer. The product, however, did not include mass storage which is very useful for the debugging of applications programs. By connecting a "spare" computer via a serial interface to the one embedded in the product, it was possible to use the "spare" computer's memory in exactly the same manner as a disc would be used, thereby facilitating debugging and modification of the programs.

8. Implementation of Serial Interfaces

There are many ways in which serial interfaces can be implemented. One of the easiest is to utilize MOS LSI which is currently available.[3,4] While this is a very economical method, it has two significant disadvantages. The achievable bit rates are moderately low, as most of these devices have been designed to be compatible with existing data sets. Of course, it is also true that it is easier for manufacturers to produce slower LSI devices as opposed to faster ones. Clearly, we expect faster devices to become available, and there is now at least one manufacturer who sells a device operating at up to 640 kbauds.[5] The other, perhaps more serious, disadvantage of commercially available LSI is its inflexibility with respect to word length. Most allow selection of 5, 6, 7, or 8 bits of data, but none permits easy expansion to longer word lengths. It is possible, however, to utilize the parity bit to construct a 9-bit serial interface with these devices. This is accomplished by applying the 9th data bit to the control input which selects even or odd parity. It is rather difficult to interpret this 9th bit on a communication line because it is obviously a function of the 8 "real" data bits. After having been processed by a mating receiver, however, the 9th bit may be read in normal fashion from the parity error output of the receiver.

The alternative to using commercial LSI is to construct a serial interface from available MSI. This is not as bad a choice as it may seem because a transmitter and receiver have much in common and most of the circuitry can be shared if receptions and transmissions do not overlap in time. If a transceiver is used with coaxial cable and has a crystal oscillator as its clock source, the stability and accuracy of the clock and the lack of noise associated with the use of coaxial cables enables sampling in the central quarter of the bit time, as opposed to the central sixteenth of the bit time that is used by most commercial LSI receivers.

9. Conclusion

The serial interface described in this report has proved valuable in industrial and academic settings, and has enabled the performance of operations which otherwise would not have been feasible, both for reasons of cost and convenience. It has been possible to share expensive peripherals among several minicomputers without the overhead of a mass-storage device and the associated software required for automatic queuing. We believe that this interfacing scheme can form the basis for a network of small computers.

Appendix

Sample Programmer Information for a Serial Interface

The instructions for the 18-bit Serial Interface on the PDP-9 computer are:

send

IOT 6304 (706403)	loads the transmitter buffer from the AC and sends out the serial data, AC-17 end first

wait

IOT 6301 (706301)	skips when serial data are assembled in the receiver buffer

receive

IOT 6312 (706312)	reads the contents of the receiver into the AC

enable interrupt

IOT 6322 (706322)	AC-0 = 1, enable interrupts from link AC-0 = 0, disable interrupts from link

The IOT instruction takes 4 machine cycles.

The data are serially transmitted at a 1-Mhz rate (start + 18 bits + stop), after it is loaded, i.e., 20 μs/word.

The receiver generates either a (PI) program interrupt request or an (API) automatic program interrupt.

API channel = 21, trap address = 61, priority level = 3.

References

1. RS-232-C EIA Standard, Electronic Industries Association, Washington, D. C., August 1969.
2. D. L. Dietmeyer, Logic Design of Digital Systems (Allyn and Bacon, Boston, 1971), see Teletypes, pp. 763-767.
3. UAR/T Universal Asynchronous Receiver/Transmitter Data Sheet AY-5-1012, General Instrument Corporation, Hicksville, New York, n. d.
4. MC 2257L and ML 2259L Data Sheet for Terminal Transmitter and Receiver, Motorola Semiconductors, Phoenix, Arizona, n. d.
5. PT 1482A Transmitter and PR 1482A Receiver Data Sheets, Western Digital Corporation, Newport Beach, California, n. d.

D. BOUNDS FOR UNIFORM QUANTIZER PERFORMANCE

National Institutes of Health (Grant 1 P01 GM19428-01)

O. J. Tretiak

1. Introduction

The most popular digital encoding of real-valued sources is digitization, so that it is relevant to calculate entropies at the output of such coders. Unfortunately, it is often not easy to evaluate integrals for these entropies. Goblick and Holsinger[1] observed, however, that the entropy and mean-squared distortion for a uniform quantization of a Gaussian random variable obey a simple law that is closely related to the rate-distortion function. Gish and Pierce[2] obtained asymptotic expressions for the distortion and entropy of uniform quantizers which are valid for a broad class of random variables.

In this report we shall give bounds on the performance parameters of a uniform quantizer. These are simply related to the formulas of Gish and Pierce.

2. Terminology

x is a real-valued random variable, with density function $p(x)$.

$h = -\int d\xi\ p(\xi) \log p(\xi)$ is the differential entropy.

A quantizer is a function of x defined as follows. Let $\{x_i\}$ be a monotonic sequence $-\infty \leq i \leq +\infty$. Let $\{y_i\}$ be an arbitrary sequence. The quantizer output y is related to x by

$$x_{i-1} < x \leq x_i \implies y = y_i.$$

We associate the following symbols with the quantizer:

Probabilities of coded values, $p_i = \int_{x_{i-1}}^{x_i} d\xi\ p(\xi)$.

Output entropy, $H = -\sum_i p_i \log p_i$.

The distortion is evaluated according to the r^{th} power difference measure.

$$\rho_r(x, y) = |x-y|^r \qquad r > 0$$

$$d = E(\rho_r(x, y)).$$

The following special cases are of interest. A uniform quantizer is one for which, for all i, $x_i - x_{i-1} = \delta$. The distortion may be minimized by choosing optimal output levels.

$$d_1(H) = \inf_{\{y_i\}} d.$$

An optimal quantizer is one for which both the $\{x_i\}$ and the $\{y_i\}$ are chosen optimally, constraining the entropy,

$$d_o(H_1) = \inf_{\{x_i\}, H \leq H_1} (d_1).$$

According to Gish and Pierce,[2] as δ, the quantizer increment, becomes small, d_1 of a uniform quantizer approaches d_o. They derived equations for the uniform quantizer

$$H \cong h - \log \delta \qquad \text{as } \delta \to 0 \tag{1}$$

$$d_1 \cong \frac{1}{r+1} \left(\frac{\delta}{2}\right)^r, \tag{2}$$

and for the optimum quantizer

$$H \cong h - \frac{1}{r} \log d_o - \frac{1}{r} \log (r+1) - \log 2 \qquad \text{as } d_o \to 0. \tag{3}$$

For further reference, we state the definition of variation

$$V = \text{var}\,(p(x)) = \sup_{\{x_i\}} \sum_i |p(x_i) - p(x_{i-1})|.$$

The upper bound is taken over monotone sequences $\{x_i\}$. Similarly,

$$V_i = \underset{x_{i-1} < x \leq x_i}{\text{var}} (p(x)).$$

Clearly

$$\sum V_i = V.$$

The following inequalities will be useful. For $x_{i-1} < x < x_i$, $\delta = x_i - x_{i-1}$.

$$\frac{p_i}{\delta} - V_i \leq p(x) \leq \frac{p_i}{\delta} + V_i \tag{4}$$

$$\frac{1}{x} - 1 \leq \log x \leq x - 1. \tag{5}$$

3. Bounds on the Entropy

From the definition of the entropy, for an arbitrary quantizer, we find

$$\begin{aligned} H &= -\sum_i \int_{x_{i-1}}^{x_i} d\xi\ p(\xi) \log p_i \\ &= -\sum_i \int_{x_{i-1}}^{x_i} d\xi\ p(\xi) \log p(\xi) - \sum_i \int_{x_{i-1}}^{x_i} d\xi\ p(\xi) \log \delta \\ &\quad + \sum_i \int_{x_{i-1}}^{x_i} d\xi\ p(\xi) \log \frac{p(\xi)\delta}{p_i} \\ &= h - \log \delta + Q. \end{aligned} \tag{6}$$

In this expression,

$$Q = \sum_i \int_{x_{i-1}}^{x_i} d\xi\ p(\xi) \log \frac{p(\xi)\delta}{p_i}.$$

Lower bound on H. For a uniform quantizer, $x_i - x_{i-1} = \delta$.

$$H \geq h - \log \delta. \tag{7}$$

Proof. From (5) and (6) we obtain

$$Q \geq \sum_i \int_{x_{i-1}}^{x_i} d\xi\ p(\xi) \left(\frac{p_i}{\delta p(\xi)} - 1 \right) = 0.$$

Upper bound on H. For a uniform quantizer, with p(x) of bounded variation V,

$$H \leq h - \log \delta + V\delta. \tag{8}$$

Proof. Again, from (5) and (6) we obtain

$$Q \leqslant \sum_i \int_{x_{i-1}}^{x_i} d\xi\, p(\xi) \left(\frac{\delta p(\xi)}{p_i} - 1 \right)$$

$$\leqslant \delta \sum_i \int_{x_{i-1}}^{x_i} d\xi\, \frac{p(\xi)}{p_i} V_i = \delta \sum_i V_i = \delta V.$$

Note that the lower bound is equal to the asymptotic expression, while the upper bound is the sum of the asymptotic expression and a term that decreases with δ. While it is possible to construct a $p(x)$ for which the lower bound is tight for some δ, inequality (8) is loose for small δ and differentiable $p(x)$. It is possible to obtain the following bound.

Tighter bound on H. For a uniform quantizer,

$$H \leqslant h - \log \delta + \frac{1}{4} \delta^2 \int d\xi\, \frac{(p'(\xi))^2}{p(\xi)}. \tag{9}$$

Proof. We start as before, and obtain

$$Q \leqslant \delta \sum_i \frac{1}{p_i} \int_{x_{i-1}}^{x_i} d\xi\, p(\xi) \left(p(\xi) - \frac{p_i}{\delta} \right).$$

The integral is the same as

$$T_i = \inf_b \int_{x_{i-1}}^{x_i} (p(\xi)-b)^2\, d\xi.$$

The integral can be bounded by the modulus

$$T_i = \delta \inf_b \sup_{x_{i-1} \leqslant \xi \leqslant x_i} (p(\xi)-b)^2$$

$$= \frac{\delta}{4} \left(\sup_{x_{i-1} \leqslant \xi \leqslant x_i} p(\xi) - \inf_{x_{i-1} \leqslant \xi \leqslant x_i} p(\xi) \right)^2$$

$$\leqslant \frac{\delta}{4} \left(\int_{x_{i-1}}^{x_i} d\xi\, |p'(\xi)| \right)^2.$$

and

$$Q \leqslant \frac{\delta^2}{4} \sum_i \frac{\left(\int_{x_{i-1}}^{x_i} |p'(\xi)|\, d\xi\right)^2}{p_i}.$$

We let $u(\xi) = \frac{|p'(\xi)|}{\sqrt{p(\xi)}}$ and $v(\xi) = \sqrt{p(\xi)}$, and apply the Schwartz inequality to the numerator.

$$Q \leqslant \frac{\delta^2}{4} \sum_i \frac{1}{p_i} \int_{x_{i-1}}^{x_i} \frac{|p'(\xi)|^2}{p(\xi)}\, d\xi \int_{x_{i-1}}^{x_i} p(\xi)\, d\xi = \frac{\delta^2}{4} \int_{-\infty}^{+\infty} d\xi\, \frac{|p'(\xi)|^2}{p(\xi)}.$$

This bound is approximately three times the value of Q when $p(\xi)$ is an exponential function.

4. Bounds on the Distortion

Lower bound

$$d_1 \geqslant \left(\frac{\delta}{2}\right)^r \frac{1}{(r+1)} (1-V\delta). \tag{10}$$

Proof.

$$d_1 = \inf_{\{y_i\}} \sum_i \int_{x_{i-1}}^{x_i} d\xi\, p(\xi) |y_i-\xi|^r.$$

For each i, according to (5),

$$p(\xi) \geqslant \max\left(\frac{p_i}{\delta} - V_i, 0\right)$$

$$d_1 \geqslant \min_{\{y_i\}} \sum_i \int_{x_{i-1}}^{x} d\xi\, |y_i-\xi|^r \max\left(\frac{p_i}{\delta} - V_i, 0\right).$$

The minimum over y_i occurs when $y_i = x_{i-1} + \delta/2$, and is equal to

$$\frac{2}{(r+1)} \left(\frac{\delta}{2}\right)^{r+1} \sum_i \max\left(\frac{p_i}{\delta} - V_i, 0\right).$$

By bounding the maximum by $\frac{p_i}{\delta} - V_i$, we obtain

$$d_1 \geq \frac{1}{(r+1)} \left(\frac{\delta}{2}\right)^r (1-V\delta).$$

Upper bound. For a uniform quantizer

$$d_1 \leq \frac{(\delta/2)^r}{(r+1)} (1+\delta V). \tag{11}$$

Proof

$$d = \inf_{\{y_i\}} \sum_i \int_{x_{i-1}}^{x_i} d\xi\, p(\xi) |\xi - y_i|^r$$

$$\leq \sum_i \int_{x_{i-1}}^{x_i} d\xi\, p(\xi) \left| \xi - \frac{x_i + x_{i-1}}{2} \right|^r .$$

The result follows from Eq. 5. As before, the bounds are equal to the asymptotic expression, plus a term that decreases with δ.

5. Bounds on the Entropy-Error Curve

The inequalities can be used to determine bounds on the entropy for a given distortion. The algorithm is the following.

1. For a given distortion, use (10) and (11) to find a range for δ, the quantizer step size.
2. Given the step size range, use (7) and (8) to find the range for entropy.

A formula for this is somewhat harder to obtain, since only (7) can be inverted in closed form. By relaxing the inequalities, however, we can obtain closed expressions for the bounds.

Upper bound. For $d \leq 2(2V)^{-r} (1+r)^{-1}$,

$$H \leq h - \frac{1}{r} \log (d2^r(r+1)) + (c+V)\, 2(d(r+1))^{1/r},$$

where

$$c = V/r \qquad r \geq 1$$

$$c = V(2^{1/r}-1) \qquad r < 1.$$

Proof. From (11) with $x = d2^r(1+r)^{1/r}$ and $T(\delta) = (1+V\delta)^{1/r}$

$$x \leqslant \delta T(\delta).$$

The term $T(\delta)$ is a concave downward (c. d.) function of δ if $r \geqslant 1$, so that

$$r \geqslant 1 \implies T(\delta) \leqslant 1 + \delta T'(0) = 1 + \frac{V\delta}{r}.$$

For $r < 1$, $T(\delta)$ is concave upward (c. u.) and we can overbound it over any interval of δ with a linear function that matches it at the limits of the interval.

$$\left(0 \leqslant \delta \leqslant \frac{1}{V}\right)(r<1) \implies T(\delta) \leqslant 1 + ((2)^{1/r}-1)V\delta$$

or, by defining c as in the statement of the bound,

$$x \leqslant \delta(1+c\delta).$$

Solving for δ, we obtain

$$\delta \geqslant \frac{x}{\frac{1}{2}\left(1+\sqrt{1+4cx}\right)}.$$

Inequality (8) is a monotone decreasing function of δ over $0 \leqslant \delta \leqslant 1/V$, so that we may substitute and obtain

$$H \leqslant h - \log x + \log\left(\frac{1}{2}\left(1+\sqrt{1+4cx}\right)\right) + \frac{Vx}{\frac{1}{2}\left(1+\sqrt{1+4cx}\right)}.$$

The denominator of the last term is greater than or equal to 1. Using the usual bounds for the log and the square root, we obtain

$$H \leqslant h - \log x + (c+V)x.$$

The inequality follows by substituting y for x.

Lower bound. For

$$d \leqslant \left(\frac{r}{r+1}\,\frac{1}{V}\,2\left(1-\frac{zr}{(r+1)V}\right)\right)^r \frac{1}{r+1}$$

$$z = \frac{V}{r} \qquad r \geqslant 1$$

$$z = \left(1-\left(\frac{1}{r+1}\right)^{1/r}\right)\left(\frac{r+1}{r}\right) \qquad r < 1$$

$$H \geq h - \frac{1}{r}\log\left(d2^r(r+1)\right) - 4z\left(d2^r(r+1)\right)^{1/r}$$

<u>Proof.</u> Starting with (10) and $T(\delta) = (1-\delta V)^{1/r}$, we get

$$\delta T(\delta) \leq x$$

and as before

$$\delta < \frac{r}{(r+1)V}, \qquad r \geq 1 \qquad T \geq 1 - \frac{\delta V}{r}$$

$$\delta < \frac{r}{(r+1)V}, \qquad r < 1 \qquad T \geq 1 - \left(1 - \frac{1}{(r+1)}\,\frac{1}{r}\right)\left(\frac{r+1}{r}\right)^{\delta} V.$$

Or, more briefly, with the proper definition of z

$$x \geq \delta(1-z\delta).$$

By solving for δ, we obtain

$$x \leq \frac{1}{4z} \Longrightarrow \delta \leq 2x\left(1+\sqrt{1-4zx}\right)^{-1}.$$

We can ensure the proper range for δ by requiring

$$x \leq \frac{r}{(r+1)V}\left(1 - \frac{zr}{(r+1)V}\right).$$

From inequality (8) we now obtain

$$H \geq h - \log x + \log \frac{1}{2}\left(1+\sqrt{1-4zx}\right).$$

Using the bound for the log, we obtain

$$H \geq h - \log x + \frac{\sqrt{1-4xz} - 1}{\sqrt{1-4xz} + 1}.$$

The last term here is c.d., so that it bounds a linear function over the interval $0 \leq x \leq 1/4z$ which gives

$$H \geq h - \log x - 4zx.$$

References

1. T. J. Goblick, Jr., and J. L. Holsinger, "Analog Source Digitization: A Comparison of Theory and Practice," IEEE Trans., Vol. IT-13, No. 2, pp. 323-326, April 1967.
2. H. Gish and J. N. Pierce, "Asymptotically Efficient Quantizing," IEEE Trans., Vol. IT-14, No. 5, pp. 676-683, September 1968.