IMAGE INFORMATION, CLASSIFICATION AND CODING

P. D. Dodd
F. B. Wood
International Business Machines Corporation
Advanced Systems Development Division
Los Gatos, California

Summary

Since information content and redundancy vary in documents of different types, these variations must be measured to classify the images for efficient compression and transmission in an automatic document-handling system. Measuring the ϵ-entropy (where ϵ is the scanning resolution), we compared information content and redundancy in numerous examples of three major types of documents. We then plotted the relationships between classes identified and compression ratios reported (theoretical, simulated and actual) to indicate the state of the art. Improved compression ratios may depend on adaptive scanning, which can identify each document by class and switch to an appropriate compression code.

Introduction

In an automated information storage, retrieval and transmission system, encoding of images by linear scanning, as in ordinary facsimile, is not generally satisfactory. It may take too much time on the communication channel or (depending on the trade-off of time and bandwidth) may require a very large memory for storage. Since most document images have considerable redundancy, code compression can profitably be introduced to minimize the time-bandwidth product. Although savings of the order of ten to one hundred are possible, the factor in practical coding systems is much lower (roughly three to five) because signal distribution properties vary so much from image to image. The practicability of code compression depends upon the structure of the total system, particularly upon the availability of processing logic and processing buffer memory. It should be possible to develop an optimum code compression system for each class of documents -- typescripts, drawings, photographs, etc. If a prescanning device could determine the kind of document being processed, the logic could then switch to the appropriate compression coding system and much greater economy would be possible. Before we can design such an adaptive system, we must know the types of documents to be classified, develop means of automatically identifying each class, and select the best code-compression scheme for each class.

The goal implied can be described in this way: We want to obtain the essential information from an image (of a document page, for example); convert this information into a form suitable for transmission and storage in an economical way; and retrieve (then or later) the coded information, transmit it to another station (if required) and decode it to reproduce the original form. By original form, we mean that the reproduced copy must be able to serve the same purpose as the original, whatever that purpose was, and serve it equally well. In the copy of a printed page, each transition between black and white must be within $\pm\epsilon$ of its original location (where ϵ is the resolution of the scanning system). Repeated encoding and reconstruction introduces a decay factor, and this becomes important in the evaluation of a system.

Many factors must be taken into account in solving this problem: buffering requirements for various schemes, for example, and economic trade-offs between complexity of terminal equipment buffer storage, processing logic and transmission costs. Such considerations are beyond the scope of this paper, but wait upon the questions we consider here: (1) How much of the information in any message is needed to represent that message uniquely? (2) How can we strip the redundancy from a message so that only essential information need be transmitted or stored? (3) How can we most efficiently represent (encode) that minimum information?

In our theoretical analysis, we assumed certain restrictions on the reading and writing techniques: that both are incremental, providing a serial flow of information; that a prescan precedes the encoder scanning spot at a variable distance; that reading, or writing as well, can employ à row of scanning spots in a vertical or diagonal line for multiple scans. The prescan and scan can be of different types -- for example the prescan could be linear, with a digital facsimile representation being stored in a computer buffer; the scan could then be the processing of the buffered information. In such a case, the compression coding would be software (a computer program) instead of hardware.

A device employing a single coding system with a few adaptive features could handle a variety of documents by changing speed, by grouping lines vertically, and by skipping blank lines. Ideally, it would be able to revert to straight facsimile whenever the detail of the image departed too far from the conditions for a particular code. In a software version, the mechanical prescan would be linear; variable-speed scanning, grouping of lines, and skipping of blank lines would be alternate paths in the computer program.

Description Capacity, Information, Entropy and ϵ-Entropy

There are several approaches to calculating the information content. If we have a way of estimating the number of different images (w) that can be represented by a page with a specified resolution, we can consider (w) the "description capacity."[1] If we take each of these possible images as equally likely, i.e., $p = 1/w$, the entropy is

$$I = \sum_{i=1}^{w} p_i \log p_i = -w\ (1/w \log w) = -\log w.$$

If the spots are quantized into a binary coding, "black" or "white," and if H is the number of spots per page, then

$$w = 2^H, \text{ and } I = -\log_2 (2^H) = -H.$$

The quantity I is the "entropy," as defined by Shannon.[2] The number of bits (binary digits) H, is variously identified as the information content, communication entropy[3] or negentropy[4] (negative entropy).

The number of bits per page depends upon the size of the page, the coordinate system (i.e., rectangular, polar, bipolar, etc.), and the resolution of the scanning system. In this report we assume rectangular coordinates, and source documents 8 1/2 by 11 inches; and introduce the term "ϵ-entropy" to identify the reference base in terms of scanning resolution, as has been done by Vituskin.[5] The number of bits/page can thus be considered the source entropy, the measure of information content.

Since we do not have detailed knowledge of the conditional probabilities of black and white spots in documents, we must approach the problem by finding coding systems which when used with experimental statistics give upper bounds on ϵ-entropy. For lower bounds we look for character recognition examples of pattern recognition[6] or simple documents constructed in such a way that their images can be redrawn at the receiver.

Definition of ϵ-Entropy for Image Classification

In reviewing the bounds on the communication entropy required for different types of documents, we shall use Vituskin's analysis[5] for simplified cases, and use McLachlan's description mechanics[1] to obtain some upper bounds on the description capacity of certain types of images. For other images particular codes will be used to get upper and lower bounds.

The relative ϵ-entropy concept applies to classes of mathematical spaces which give indications of being useful in image classification, and can be developed to apply to practical cases. Feinstein[7] pointed out a special case of a theorem from Vituskin, and examination indicates that Vituskin's work may be quite relevant to the image classification problem.

Vituskin's concept of relative ϵ-entropy corresponds to the logarithm of McLachlan's descriptive capacity of the document page. McLachlan carries his analysis over a range of physical systems, while Vituskin carried his analysis over a range of mathematical spaces. The convergence of various series representations of functions permits the introduction of two concepts that apply to metrical spaces -- absolute ϵ-entropy and ϵ-capacity.

Let F be a physical space such as the x-y plane in Fig. 1, where F_{ab} is a subsection of specific dimensions (e.g., a = 8 1/2 inches, b = 11). Let θ be the number of bits per sample point of the net, giving 2^θ levels. For example, black and white (two levels) correspond to $\theta = 1$, and $2^\theta = 2$. Three bits per sample point ($\theta = 3$) would give eight shades of gray; five bits ($\theta = 5$) would give a 32-level gray scale. $S_\epsilon{}^\theta(F_{ab})$ is the coordinate net covering space F_{ab} with resolution ϵ; and $W_\epsilon{}^\theta$, the descriptive capacity of the minimum number of points of that net, indicates the total number of different images which $S_\epsilon{}^\theta(F_{ab})$ can describe.

Let $I_\epsilon{}^\theta(F_{ab}) = \log_2 W_\epsilon{}^\theta(F_{ab})$.
$I_\epsilon{}^\theta(F_{ab})$ is the ϵ-entropy of the subset of set $=$ F_{ab} with resolution ϵ. The smallest resolvable spot is a square of side ϵ. This is analogous to Vituskin:

$$H_c{}^\theta(F) = \log_2 N_\epsilon{}^\theta(F)$$

and $W_\epsilon{}^\theta(F_{ab})$ corresponds to McLachlan's descriptive capacity, w.

Redundancy in Basic Classes of Documents

Most document pages are identified as belonging to one of three general classes: continuous tone (including photographs); handwritten, typed or printed text; and line drawings. These are taken to be the kinds of documents to be identified, and we will assume that the images can all be described by rectangular coordinates. The information content of photographs can be indicated by a ten-level gray scale. The second class (usually text composed of alphanumeric characters) has two levels, black and white; 11-point type and pica typing are taken as typical, with black space allowance corresponding to normal typed pages. Line drawings, in the third class, are also two-level, black and white; estimates here have been based on simple sketches and circuit block diagrams.

The whole range of documents is plotted on a logarithmic scale in Fig. 2, where the vertical scales show both description capacity (on the left) and ϵ-entropy (on the right). The latter is for the specific case $H^R_{0.02}(F_{22,28})$, where R means rectangular coordinates, $\epsilon = 0.02$ centimeters (or 125 scans/inch), and the documents are 22 by 28 centimeters (or 8 1/2 by 11 inches).

Assuming that these are human-readable documents and that the output should meet the standards by which the source documents are judged acceptable, resolution compatible with the powers of the human eye is our natural goal. A resolution of 125 lines/inch is taken as standard, which means a capacity of 1.5×10^6 binary digits per page (8 1/2 by 11 inches). Where a lower resolution is acceptable, that number can be reduced. For 100 lines/inch, for example, a page could be represented by 10^6 bits.

The possibility of code compression -- i.e., of reducing the number of bits which must be sent to convey a given message -- depends on identifying and eliminating redundancy in the original message. We have tried to determine the redundancy range for each of these classes, since any adaptive scanning system for general document handling would need to be able to accommodate itself to these ranges.

Figure 2 assembles the pertinent details for more than thirty specific documents or kinds of documents, and organizes these examples under three general classifications: photographs, typescript or print, and drawings. The plotted points are derived as explained in Table 1, from References 8 through 15 cited in the table. We shall first examine one case from each of the three major classes, and then consider intermediate cases.

The third column represents the class of black and white photographs with a ten-level gray scale. The boundary range is based on three sample calculations: (d) represents an upper bound estimated by the change from case (aℓ) binary to case (a), ten levels. (In Table 1, this is rated by the change of units from hartleys to binits, because this case is equivalent to a change of log from base two to base ten.) Case (r) falls slightly lower because McLachlan did not count the margins. Cases repeated in the literature (see Ref. 16, particularly IRE Trans on IT) suggest (q) as the approximate lower bound. Falling below that, case (e) shows the effect of reducing the bandwidth by replacing approximately half the bits by random noise. This economy is practical because the human eye can smooth out enough of such noise fluctuation to recognize the reproduced image without difficulty. Case (e) is not shown as the normal lower bound because the human eye must intervene to integrate the image after one compression and reconstruction sequence; i.e., the possibility of automatic cycling through compression and reconstruction has not been shown.

Exemplifying the second major class, typing or 11-point print has an upper bound (f) which comes close to an estimate by the description mechanics technique and by a run-length coding system developed by Ford Instruments. The lower bound (z) is based on a repeated series of scans for processing in computer memory, equivalent to character recognition. (For a description of similar programs, see Greanias, et al.[17] For later developments in character recognition, see Horowitz and Shelton.[18]) The intermediate cases, (y) and (s) are, respectively, a Shannon-Fano coding of pica type, and a simplified variable-length code.

For drawings, the third class, the upper bound (i) was obtained from a simplified variable-length code designed for typing, and the upper bound line is based on the assumption that a code compression system for one type of document can be applied to other types. The lower bound (k) is based on human coding of a circuit diagram into a computer compiler language like BLODI. The intermediate values are based on case (j), a Huffman code designed for IBM drawings, and case (ac), a straight binary count of run length.

A wide variety of other cases, representing the full range from the digital coding of color photographs to the coding of blank pages, can be analyzed on the basis of Fig. 2 and the identifying notes in Table 1.

We can now estimate potential code compression ratios: the straight facsimile entropy divided by the entropy for the upper and lower bounds equals the lower and upper bounds, respectively, for the compression ratios. In Table 2, these bounds are estimated for the sample cases examined above for each major class. The same case designations (a) through (aℓ) facilitate comparison, and, in the first two columns, points plotted in Fig. 2 are represented numerically. The right-hand column shows the redundancy range relative to 1.5×10^6 bits per document (8 1/2 by 11 inch page). In the absence of such information for photographs, a ratio reported for television pictures is listed as possibly indicative.

For typed or printed pages, 12.8 is an experimental value from Bell Telephone Laboratories for a particular type font; 47 is a theoretical limit based on 4.75 bits per character (character recognition) which would require many scans or processing steps. For line drawings, the 25 for case (i) results if a simplified variable-length coding designed for typing is applied to a very simple drawing. The 115 is for case (j), a Huffman code designed for IBM drawings, and the upper bound, 710, is not machine-realizable, being for manual coding into BLODI input language.

The examples of code compression included in Table 2 are based on particular sets of statistics. They fall within the projected ranges, in vicinities which may prove to be typical for each class; and they suggest what is certainly great variation in the redundancy range from one class to another. The same compression code cannot be equally efficient for documents which differ so greatly. The foregoing analysis underscores the need for adaptive coding of some kind. The system designer will aim for minimum buffer requirements, maximum functional overlapping of hardware elements, and at least two scans of the input data, when his goal is a system which can accommodate these classes of documents.

Another possibility suggested, a semi-adaptive system, could work this way: at its point of origin, every document page could be marked to designate the code-compression applicable. (For a sample of adaptive scanning, see Van Blerkom.[19]) The scanning system would accept such pre-identified images, simply reading the label without examining and classifying the contents. This would not actually eliminate steps, but would move the burden of classification to a different and perhaps more logical place in the system. Automatic information-handling systems are bringing about increasing standardization in document preparation, and coding-at-source could be included along with indexing-at-source and standard formatting.

Theoretical Basis for Image Compression

When a message is coded for transmission, i.e., converted into a signal suitable for a given channel, the statistical properties of the message may or may not be taken into account. If they are not, there is simply a one-to-one conversion of the message into a new physical variable, as when a microphone converts sound pressure into proportional voltage or current. Such non-statistical encoding processes require the same transmission time for all messages of the same length; they require no memory; they have a small and constant delay; they are inefficient in their use of channel capacity. By contrast, statistical encoding takes into account the probabilities of a message. Sequences which are likely to occur often are represented by short code designations; less likely sequences are assigned longer codes (as in Morse code, where the shortest code groups represent the most common letters). Statistical encoding generally requires memory; it transmits messages of the same length at different rates, depending on their content, and must therefore have variable delays (buffers) at the sending and receiving ends in order to accept and deliver messages at constant rates. In statistical encoding, then (as Oliver[20] concluded) "the usual inefficiency which results from ignoring the correlation between messages is lessened because this correlation is less in the reduced message." Reducing the information rate (in Shannon's definition, "the average uncertainty as to the next symbol when all the past is known") thus reduces the number of bits per second required to describe the average message.

To see how this works, consider a scanned image where s(t) is an ensemble of the scanner output waveforms and S(f) is the spectrum of s(t) (see Fig. 3). If the image data is black and white only, the signal s(t) will be a random square wave. If s(t) is bandwidth-and-time-limited as shown, it is completely described as a vector in 2FT dimensional signal space.

Now quantize signal samples into n discrete levels so that the n-dimensional sample space contains sample points s. If no knowledge of the past or future is available, the average amount of information or entropy (H, in bits) required to specify a particular present sample s(σ) is, according to information theory,

$$H\ (\sigma) = -\sum_{i} P(s_i)\ \log\ (Ps_i)$$

where $P(s_i)$ is the probability of the present sample value s_i and σ is the space of sample values. An upper bound on the entropy is

$$H\ (\sigma) \leq \log n,$$

with equality if and only if all sample values are equally likely.

If, on the other hand, we can take into account past and future samples, we can reduce the amount of information required to specify the present sample. This reduced quantity

$$H\ (\sigma/\rho) = -\sum_{ij} P(s_i,\ r_j)\ \log\ P(s_i/r_j)$$

where σ is the space of the present sample values (s_i), and ρ is the space of all past and future sample values (r_j). When the samples are correlated,

$$H\ (\sigma/\rho) < H\ (\sigma).$$

We thus define redundancy (σ) as the average mutual information:

$$I\ (\sigma;\rho) = H\ (\sigma) - H\ (\sigma/\rho).$$

Stated in other words, the average number of bits required to send a signal sample is H (σ) if we have no knowledge of past and future sample values; but only H (σ/ρ) bits per sample for transmission of storage. If, however, the first-order density is peaked, the source entropy is reduced. To gain this goal, we must find transformations which make certain sample values more likely than others, and then introduce a source transformation such as Huffman coding[13] to obtain an efficient binary representation of the source.

In summary, efficient coding of a redundant information source (or sequence of signal samples) requires two steps. We first transform the sequence of correlated samples into a sequence of uncorrelated samples, using the past and future sample values to "decorrelate" the present sample. This decorrelation procedure should produce peaked first-order probability densities. We can then convert the sequence of decorrelated samples to an efficient binary sequence (by Huffman coding, for example). The process of Huffman coding produces the desired result -- a sequence of binary digits which conveys a maximum amount of information per sample.

Examples of Image Compression

Modified Huffman Coding of Run Lengths

Variable-length coding can result in the compression shown in Table 3, based on the run-length statistics reported by Michel, et al. Figure 4 shows graphically how this code would be applied. To avoid complex coding of an extremely large alphabet, we have taken only the ten most probable run lengths plus some special symbols. All others are coded with a special prefix plus a ten-digit binary number to designate the count. The achievable compression can be computed from the average run length (T).

$$T = \sum_{i} L_i P_i$$

where L_i is the length of the code word corresponding to the i^{th} symbol, and P_i is the probability that the i^{th} symbol will occur. As Fig. 5 shows, the average code word for coding scheme (2) is less than half as long as the average word in straight binary transmission (1), and the average is even less for schemes (3), (4) and (5).

An Example of Reduction of Redundancy

Figure 6 compares several ways of scanning an 8 1/2 by 11 inch page of typing: from straight facsimile reproduction through successive stages of code compression to the level of character recognition. (Elite type, 6 characters per vertical inch and 12 per horizontal inch, is assumed except where noted.) A straight facsimile scan for ϵ = 125 lines/inch gives 1.5 x 10^6 bits/page (Fig. 6a). When run lengths of solid black or white are counted, the results are as shown in Fig. 6b. Examining the probability distribution of the digits required to represent the run-length count, we saw that recoding could improve the efficiency. Consequently the next step was to recode each string of zeros or ones into a ten-bit binary count (Fig. 6c). As Fig. 6 goes on to show, other codes -- Shannon-Fano, Huffman, or simplified variable-length -- can reduce the average number of bits per run (k) in such a case, even when the necessary bits are added for synchronization.

For optimum two-dimensional recording of typed documents, vertical (spatial) redundancy can be reduced. If s lines are grouped together by fiber optic scanning heads, for example, the vertical line scan (h) is divided by s as in Fig. 6e. Although this increases the average run length count (k), a suitable code can lead to a net increase in compression. The compression suggested in Fig. 6 is typical of the lower bound on information per page for optimum recoding of typed documents.

If we have the logic required for character recognition, we approach the ultimate limit for code compression. Figure 6f assumes entropy of six bits per character, and Figure 6g is based on the average entropy of the English alphabet, 4.75 bits per letter, for full character recognition. For word recognition, the redundancy of English would reduce the average entropy to 2.62 bits per letter.[12]

Selection of Proper Base for Computing Compression Ratio

To estimate or calculate compression ratios accurately, we must be sure we are referring them to the proper base. As noted, the redundancy ranges in Table 1 are relative to 1.5 x 10^6 bits, and redundancy or compression ratios computed in reference to other standards would be expected to vary. The resolution required affects the computation of the compression ratio as shown in Fig. 7 (based on typed copy only). In that figure, examples (A), (B) and (C) show the bits per page for ϵ = 0.0125, 0.020 and 0.031 centimeters, i.e., for 200, 125 and 100 lines/inch respectively. A simplified variable-length code produces approximate compression ratios of 12.8, 10 and 11 for cases (H), (I) and (J). A Shannon-Fano code would give greater compression.

The effect of the resolution requirement can be illustrated in this way: if ϵ were 0.0125 centimeters instead of the 0.031 assumed, the compression for (I), which is a sample of pica type, would appear to be 37.8 rather than the 10 indicated. To prevent this kind of ambiguity, which can invalidate comparisons, we must generally determine the maximum resolution required for actual duplication of characters and take that as the base for all calculations. If the images need not be duplicated exactly, but must be clearly and unambiguously recognizable, then the resolution requirement will change. The solid, fully-formed character shown on line (N) in Fig. 7 may not be required. If, for example, the dotted character on line (P) is acceptable instead, the resolution requirement changes accordingly. It is the assumption that the completely formed character is required which limits ϵ to 0.031, and sets the compression ratio at 10 in this case.

The lines in example (D) represent a half-page of typing requiring 50,000 bits, as the full-page case (I) requires 100,000 bits. Example (E) represents a quarter-page of line drawings, or approximately 8400 bits. Example (F) represents a blank quarter-page, coded by 380 bits. (G) combines the results of (D), (E) and (F). Assuming ϵ to be 0.031, this results in 58,780 bits for the page (50,000 + 8400 + 380) and a compression ratio of 17. If ϵ were 0.020, on the other hand, the compression ratio would be 25, as indicated. This shows again how the true limit can be stated only in relation to the exact resolution required. As (J) illustrates, the compression ratio is the same for elite and pica type, since ϵ naturally changes in a corresponding way. (K), (L) and (M) show the limits for character recognition, which requires more complex logic and operates within the restriction of a fixed alphabet.

Conclusions

As the above examples show, facsimile signals contain a great amount of redundancy, and standard facsimile transmission and storage practices are highly inefficient. Considerations suggested by information theory can lead to techniques for efficient coding of the information in a message, provided that the information can be characterized as a statistical ensemble. Since documents to be scanned, transmitted and reproduced fall into a number of different ensembles due to variations in content, adaptive coding procedures must be introduced if a coded-facsimile transmission and storage system is to be highly efficient for a wide class of documents.

References

1. McLachlan, D., Jr., "Description Mechanics," Information and Control, Vol. 1, 1958, pp. 240-266.

2. Shannon, C., and Weaver, W., A Mathematical Theory of Communication (University of Illinois, Urbana, 1949).

3. Fano, R., Transmission of Information (M.I.T. Press, Cambridge, 1961), p. 42.

4. Brillouin, L., Science and Information Theory (Academic Press, New York, 1962).

5. Vituskin, A.G., Theory of the Transmission and Processing of Information (Pergamon Press, London, 1961) (From Russian).

6. Thomas, F.J., "Character Recognition Literature," IBM Research Report RC-693, June 15, 1962.

7. Feinstein, A., private communication. See also the fundamental theorem in his Foundations of Information Theory (McGraw-Hill, New York, 1958).

8. Roberts, L.B., "Picture Coding Using Pseudo-Random Noise," IRE Transactions on Information Theory, IT-8, No. 2, February 1962, p. 145.

9. Wyle, H., et al., "Reduced-Time Facsimile Transmission by Digital Coding," IRE Transactions on Communication Systems, CS-9, No. 3, September 1961, p. 215.

10. Michel, W.S., et al., "A Coded Facsimile System," 1957 IRE WESCON Convention Record, Part 2, p. 92.

11. Shannon, C.E., "A Mathematical Theory of Communication," Bell System Technical Journal, Vol. XXVII, July 1948 and October 1948, pp. 379-423 and pp. 623-656. Also (equivalent method) Fano, R.M., "The Transmission of Information," M.I.T. Research Laboratory of Electronics, Report No. 65, 1949.

12. Shannon, C.E., "Prediction and Entropy of Printed English," Bell System Technical Journal, Vol. 30, January 1951, pp. 50-64.

13. Huffman, D.A., "A Method for the Construction of Minimum Redundancy Codes," Proceedings of the IRE, September 1952, pp. 1098-1101.

14. Kelly, J.R., Jr., et al., "A Block Diagram Compiler," Bell System Technical Journal, May 1961, p. 676.

15. Cherry, C., Electronic News, June 11, 1962.

16. Stumpers, F.L., "A Bibliography of Information Theory," IRE Transactions on Information Theory, IT-2, November 1953; Supplement, IT-2 No. 2, September 1955, pp. 33-44; Second Supplement, IT-3, No. 2, June 1957, pp. 150-166.

17. Greanias, E.C., Hoppel, C.J., Kloomok, M., and Osborne, J.S., "Design of Logic for Recognition of Printed Characters by Simulation," IBM Journal of Research and Development, January 1957, pp. 8-18.

18. Horowitz, L.P., and Shelton, G.L., "Pattern Recognition Using Autocorrelation," Proceedings of the IRE, January 1961, pp. 175-185.

19. VanBlerkom, R., and Blasbalg, H., "Adaptive Redundancy Removal," NEREM (Northeastern Electronics Research and Engineering Meeting), Boston, November 14, 1961.

20. Oliver, B.M., "Efficient Coding," Bell System Technical Journal, July 1952, pp. 724-750.

Table 1. Documents Analyzed for Classification and Coding

CASE	DESCRIPTION	CLASSIFICATION AND CODING DETAILS
(a)	Photographs	Gray-scale resolution: 10 levels. Scanning resolution: 125 lines per inch.
(b)	Typescript or print	Black and white. Pica and 11-point type with normal blank space for typed pages.
(c)	Line drawings	Black and white. Estimated from simple sketch and circuit diagram.
(d)	Photograph	Information content relative to two-level (black and white): I (binits) - I (Hartleys)/log 2 = I/0. 3.
(e)	Picture coding	Estimated from lower value of range (one to three bits) described in Ref. 10.
(f)	Typescript without margin (100 lines per inch)	Calculated by description mechanics. Approximates Ford Instrument Analysis.[11]
(g)	Typescript	Compression: 13.5 for 7.43 percent black. Cf. 12.8 in BTL study[12] with variable-length code approximating Shannon-Fano code.[13]
(h)	Alphabetic code (4.75 bits/character)	Requires techniques of character recognition logic.[14]
(i)	Line drawing (BTL)[12]	Coding: simplified variable-length code (designed for typescript).
(j)	Schematic drawing[15]	Huffman code.[16]
(k)	Circuit block diagram	Coding: translated into BLODI compiler language.[17]
(m)	Television picture (in absence of other data)	2.85:1 compression ratio reported.[18]
(n)	Color photograph	Description capacity for three colors and black, 10 levels each color.
(p)	Black and white	20 levels, no restraints.
(q)	Black and white	10 levels. Upper range of lower bound from range of cases reported in the literature.
(r)	Black and white	10 levels for $H_{0.02}^{R}$ (F 22, 28).
(s)	Black and white	Two levels, for $H_{0.02}^{R}$ (F 22, 28).
(t)	Halftone	(100) estimated.
(u)	4-point type	Margins and blank lines eliminated
(v)	4-point type	Shannon-Fano coding extrapolated from case (y).
(w)	Character recognition	Series of scans; then digital coding (average of 4.75 bits/character).
(y)	Pica typescript (case g)	Shannon-Fano coding.
(z)	Pica typescript	Coding into 6 bits/character recognition.
(aa)	BTL schematic (case g)	Shannon-Fano coding.
(ab)	BTL schematic (case g)	Simplified variable-length coding.
(ac)	Drawing "H" in IBM report*	Straight binary run-length coding.

Table 1. (Continued)

(ad)	Grid, BTL, case (g)	Shannon-Fano coding.
(ae)	Grid	Binary coding of coordinate lines.
(af)	Solid black	Simplified variable-length code (BTL) designed for typing.
(ag)	Solid black	Indicated by 6-bit control signal.
(ak)	Solid white (blank)	Indicated by 6-bit control signal.
(ah)	Blank page coded as for typing statistics	Shannon-Fano coding.
(ai)	Blank page coded as for typing statistics	Simplified variable-length coding.
(aj)	Blank page	Huffman code designed for drawing in case (j).
(aℓ)	Black or white	Digitized facsimile, with $\epsilon \approx 0.02$ cm.

Table 2. Estimated Bounds for Code Compression

DOCUMENT CLASSES	BITS/PAGE		COMPRESSION POSSIBLE		
	Upper Bound	Lower Bound	(For 1.5 x 10^6 Bits, $\epsilon \approx 0.02$ cm)		
(a) Photographs (10 levels of gray)	(d) 1.5 x 10^6 decimal digits; 5 x 10^6 binary digits.	(e) 1.0 x 10^6 binary digits (estimated)	(m)		
			One Scan	Two	Many
(b) Typescript or print (2 levels)	(f) 3.5 x 10^5	(h) 3.2 x 10^4	(f) 4.3	(g) 12.8	(h) 47.0
(c) Line Drawing (2 levels)	(i) 6 x 10^4	2.1 x 10^3	(i) 25	(j) 115	(k) 710

Table 3. Modified Huffman Coding of Run Lengths

SYMBOL	PROBABILITY	MODIFIED HUFFMAN CODE
(black)	0.200	10
(2)	0.180	000
(3)	0.180	001
(4)	0.100	110
(5)	0.070	0100
(all other lengths)	0.066	0101 + x . . . *
(6)	0.050	0111
(margin)	0.050	1110
(7)	0.031	01100
(1)	0.025	11110
(8)	0.022	11111
(9)	0.015	011010
(10)	0.011	011011
	Σ = 1.000	

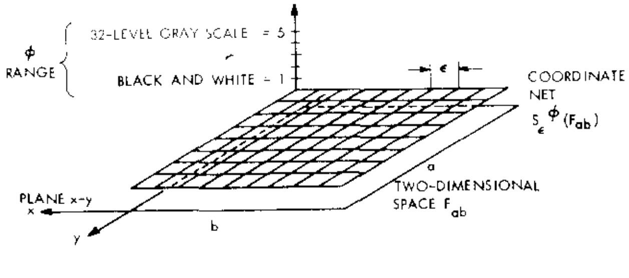


Fig. 1. Graphical Definition of ϵ-Entropy

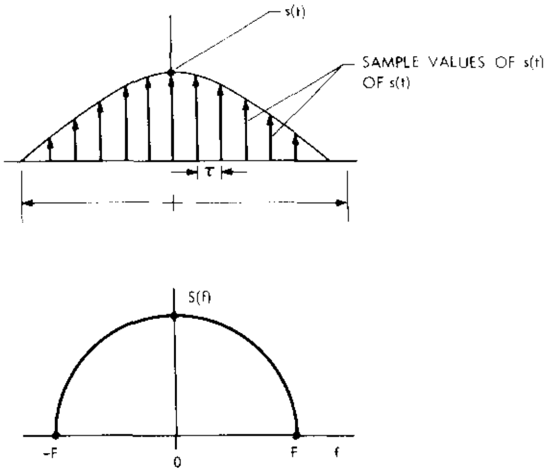


Fig. 3. Scanner Output Waveforms and Spectrum

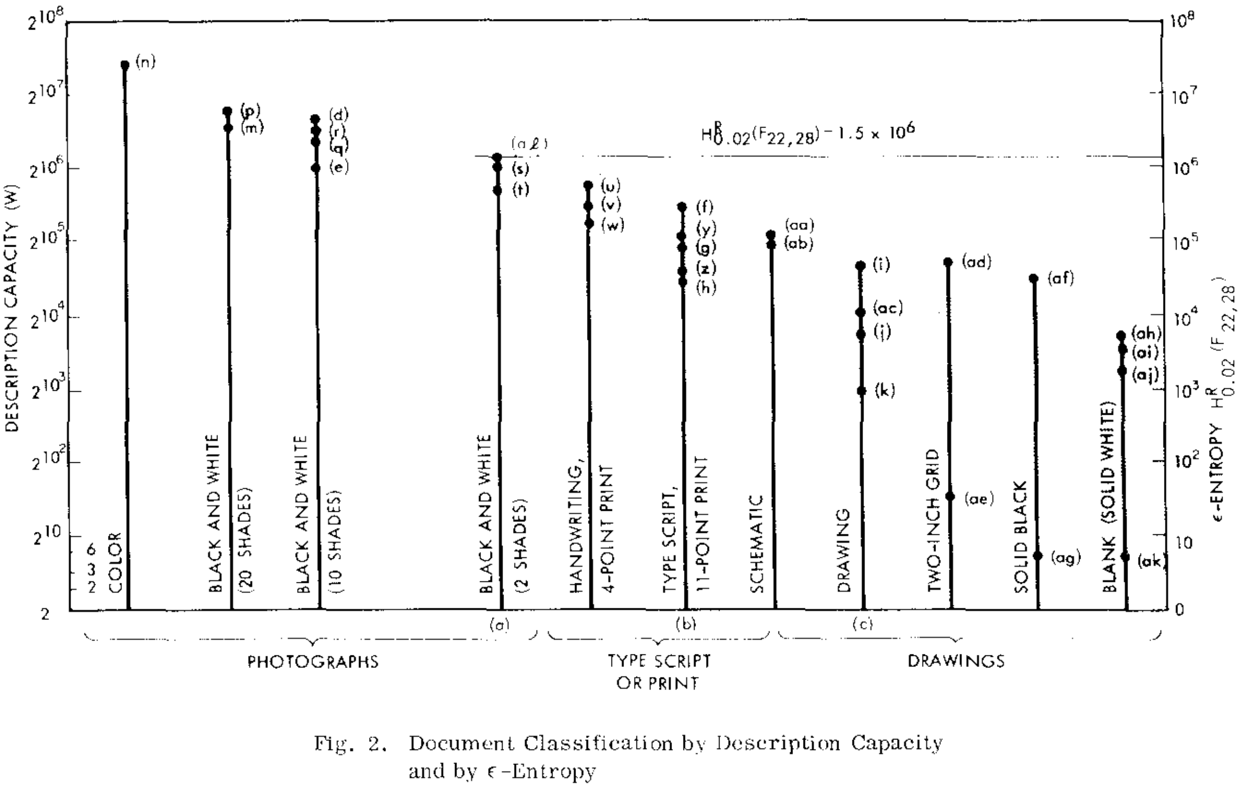


Fig. 2. Document Classification by Description Capacity and by ϵ-Entropy

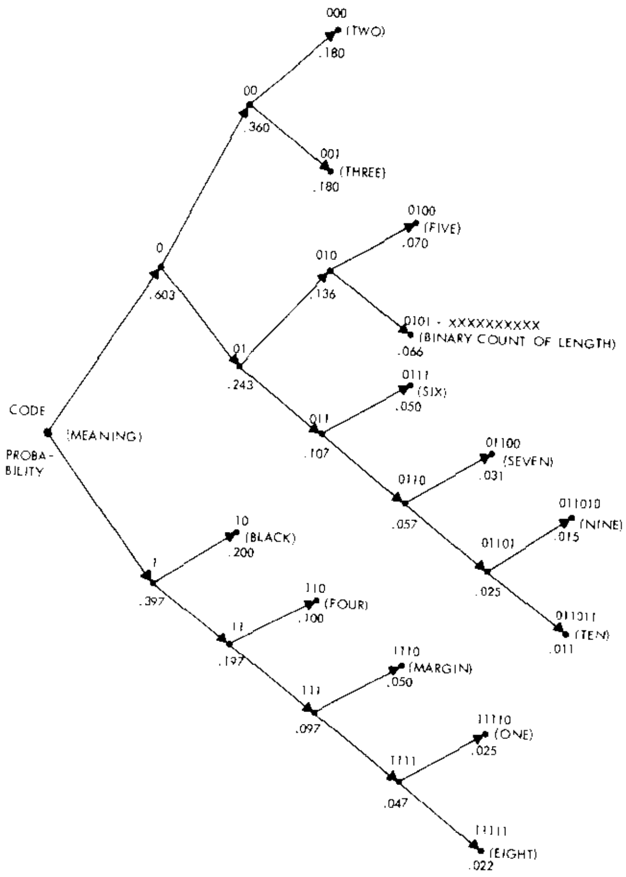


Fig. 4. Sample Huffman Code (Example from Table 3)

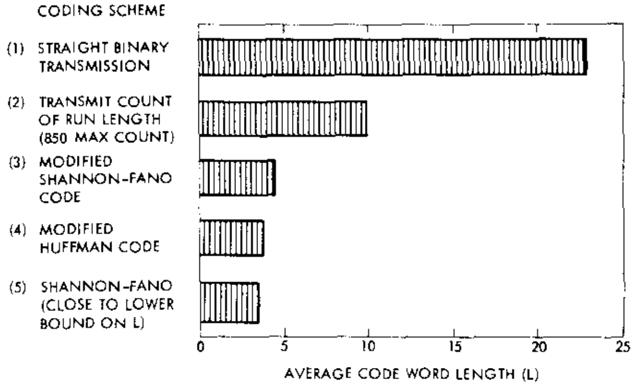
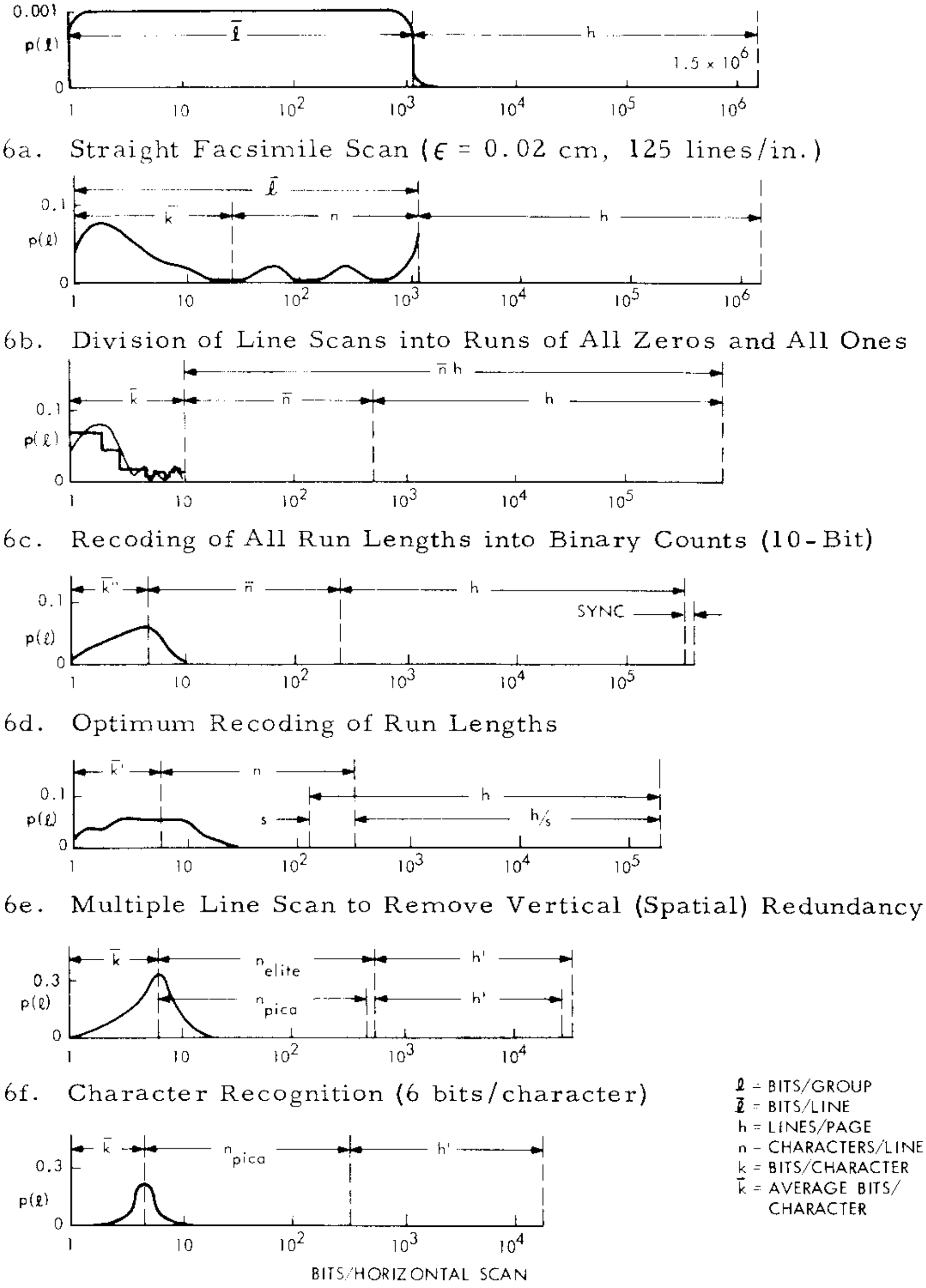


Fig. 5. Relative Code Word Lengths



6a. Straight Facsimile Scan (ϵ = 0.02 cm, 125 lines/in.)

6b. Division of Line Scans into Runs of All Zeros and All Ones

6c. Recoding of All Run Lengths into Binary Counts (10-Bit)

6d. Optimum Recoding of Run Lengths

6e. Multiple Line Scan to Remove Vertical (Spatial) Redundancy

6f. Character Recognition (6 bits/character)

6g. Character Recognition (k = 4.75 bits)

Fig. 6. Comparison of Scanning Methods for Typescript

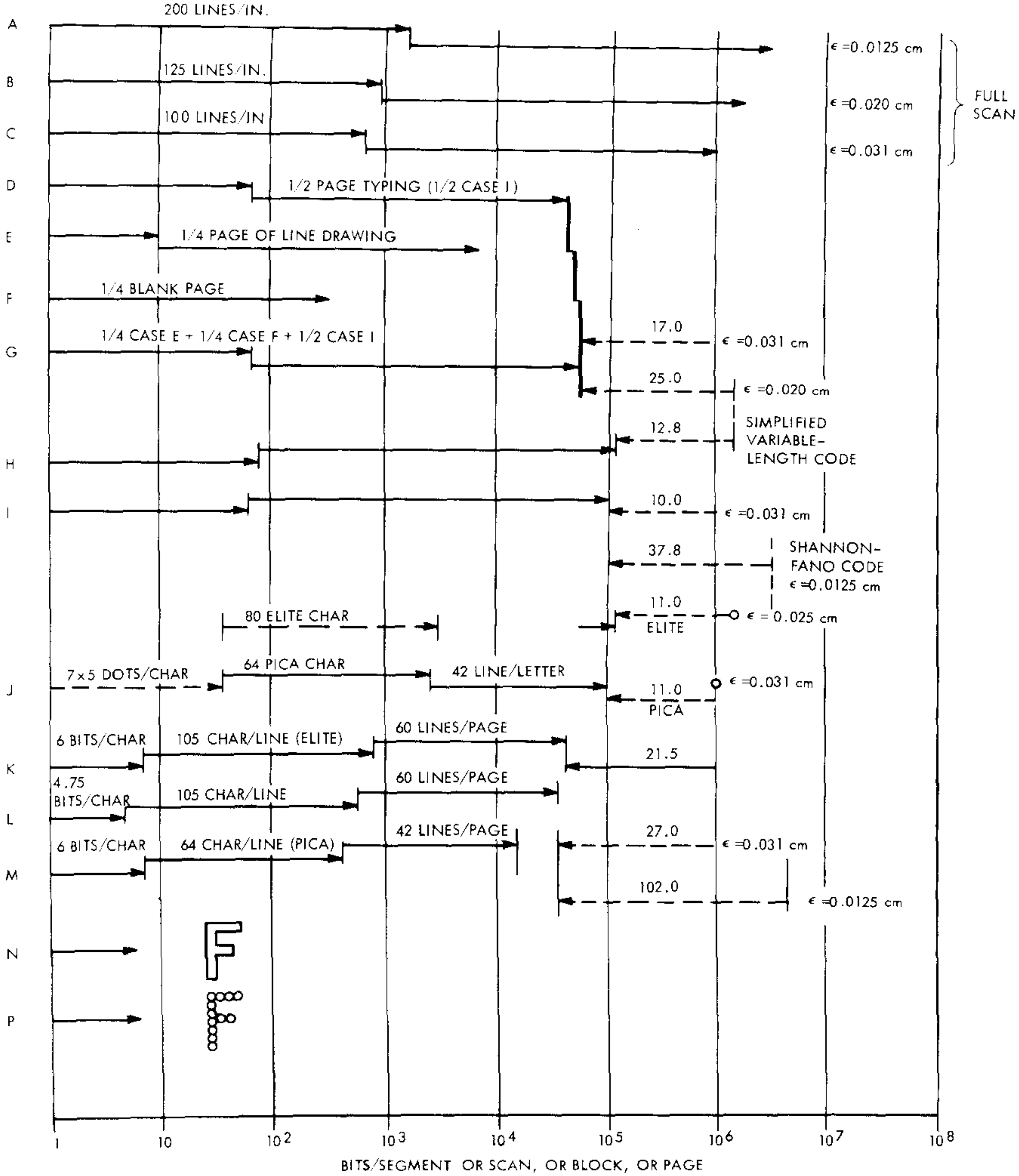


Fig. 7. Resolution Requirements and Compression Limits for Typescript

Reprinted from:
1966 IEEE INTERNATIONAL CONVENTION RECORD
Part 7