XV. PROCESSING AND TRANSMISSION OF INFORMATION*

Prof. R. M. Fano
Prof. R. G. Gallager
Prof. F. C. Hennie III
Prof. E. M. Hofstetter
Prof. D. A. Huffman
Prof. I. M. Jacobs
Prof. A. M. Manders
Prof. W. F. Schreiber
Prof. C. E. Shannon
Prof. J. M. Wozencraft
Dr. C. L. Liu
T. G. Adcock
T. M. Anderson
P. G. Arnold
M. H. Bender
E. R. Berlekamp
D. G. Botha
J. E. Cunningham
H. Dym
P. M. Ebert
D. Ecklein
D. D. Falconer
E. F. Ferretti
G. D. Forney, Jr.
R. L. Greenspan
U. F. Gronemann
A. R. Hassan
J. L. Holsinger
T. S. Huang
N. Imai
R. J. Purdy
J. F. Queenan
J. E. Savage
J. R. Sklar
K. D. Snow
I. G. Stiglitz
W. R. Sutherland
O. J. Tretiak
W. J. Wilson
H. L. Yudkin

A. PICTURE PROCESSING

1. A MEDIUM-SPEED MEDIUM-ACCURACY MEDIUM-COST ANALOG-TO-DIGITAL CONVERTER

An analog-to-digital converter has been constructed. Some of the performance specifications are listed below. Anyone who is interested may obtain a copy of the circuit diagrams.

a. General Specifications

Input: 0 to -10 volts

Output: 8 bits, natural binary code, parallel output

Conversion time: 45 μsec

b. General Description

A conversion is initiated by an external pulse. The intervals between these pulses must be greater than 45 μsec. The conversion is done on a bit-by-bit basis.

c. Construction Details

Most of the circuitry is built with medium-speed switching transistors and general-purpose diodes. The internal voltages are derived from regulated +34-volt and -25-volt power supplies.

The circuit takes up five plug-in cards. The estimated time for wiring is approximately one man-week.

---

*This research was supported in part by Purchase Order DDL B-00368 with Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology, with the joint support of the U. S. Army, Navy, and Air Force under Air Force Contract AF19(604)-7400; and in part by the National Institutes of Health (Grant MH-04737-03), and the National Science Foundation (Grant G-16526).

An approximate list of components includes: 52 pnp switching transistors, 46 general-purpose diodes, 24 special diodes and transistors, 150 resistors (1/2 watt 5 per cent), 18 precision resistors, 20 power resistors, 5 miniature variable resistors, 61 miscellaneous capacitors, 5 electrolytic capacitors, and 7 Zener diodes.

If the components are ordered in reasonable quantities, the total cost would be less than $200.

O. J. Tretiak

2. PICTURE RECORDING AND REPRODUCING EQUIPMENT

The scanner for recording and playing back pictures has been improved on by the addition of a feedback circuit to control the light output of the cathode-ray tube. The light intensity is sensed by a multiplier phototube and compared with the input signal. The error signal is integrated over the duration of a signal pulse, amplified, and fed to the cathode-ray tube.

This arrangement serves two purposes: (a) keeping the light output uniform (for constant signal) over the entire surface of the tube, that is, to eliminate phosphor noise; and (b) during playback, making the light output a linear function of the input signal (which is important for color reproduction by linear addition of primary colors).

The effectiveness of the circuit for purpose (a) is demonstrated by the measurement of a light-intensity variation in the ratio 4:1 for a simulated phosphor response variation in the ratio 45:1.

The linearity of the output is true within 10 per cent in a large-to-small signal amplitude ratio of approximately 100:1 (which is the useful dynamic range of a typical transparency) and should be compared with the highly exponential characteristic of the cathode-ray tube alone.

U. F. Gronemann

3. IMAGE CORRECTION-TRANSMISSION EXPERIMENTS

Usually only small areas of a motion picture change significantly from frame to frame. An obvious exception occurs if the camera is moving relative to a fixed background. Since this is true, a possible way of permitting the reconstruction of a sequence of moving picture frames is the transmission of information concerning the parts of the picture which change significantly.

To determine how such a system would perform, a program was written for the TX-2 computer (located at Lincoln Laboratory, M. I. T.) to simulate the transmission process. The input data were scanned from 35-mm film and recorded on digital computer tape. On this tape pictures consisting of 128 × 128 sample points, each having one of 256 possible brightness levels, were recorded.

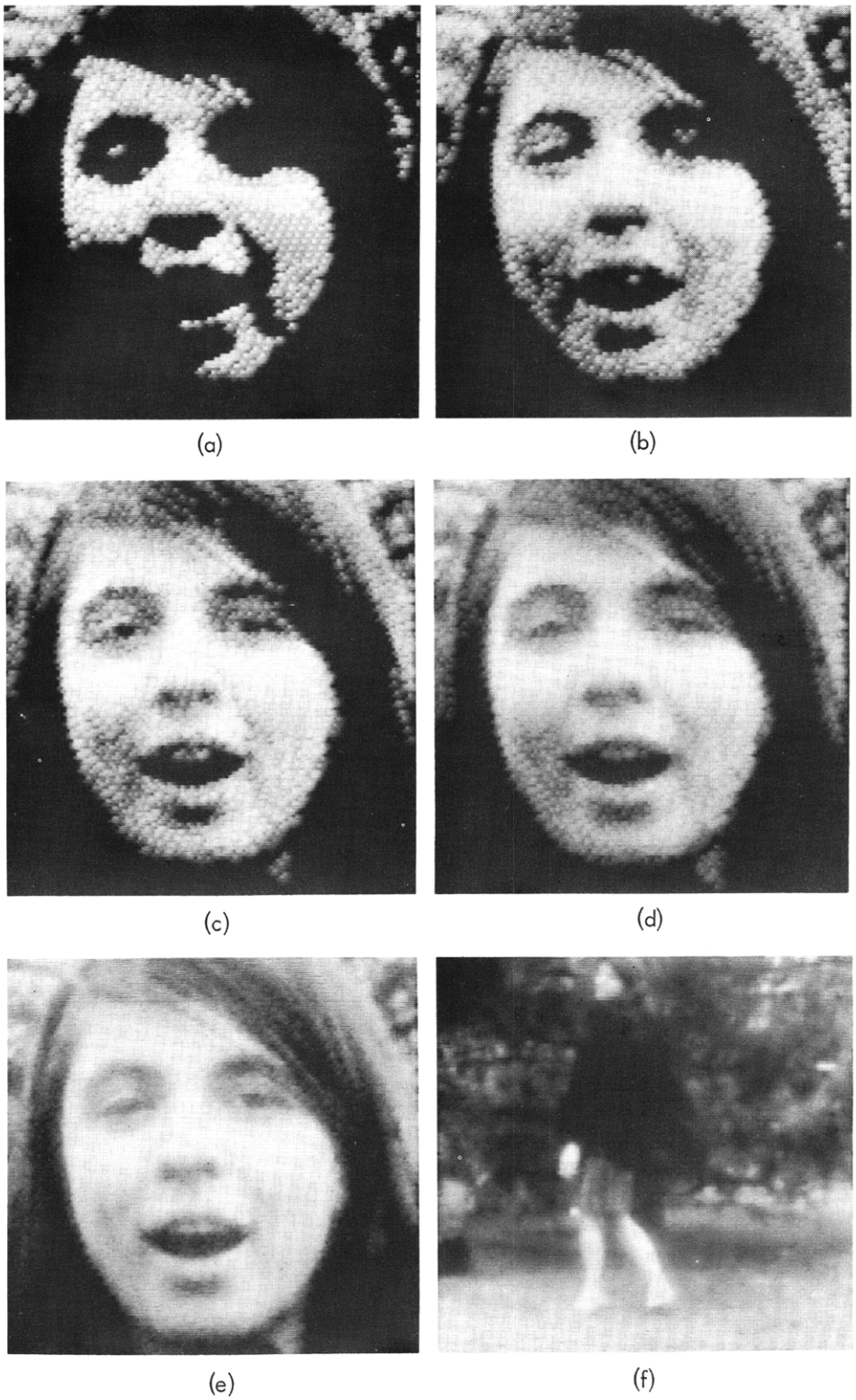


Fig. XV-1. Picture build-up examples.

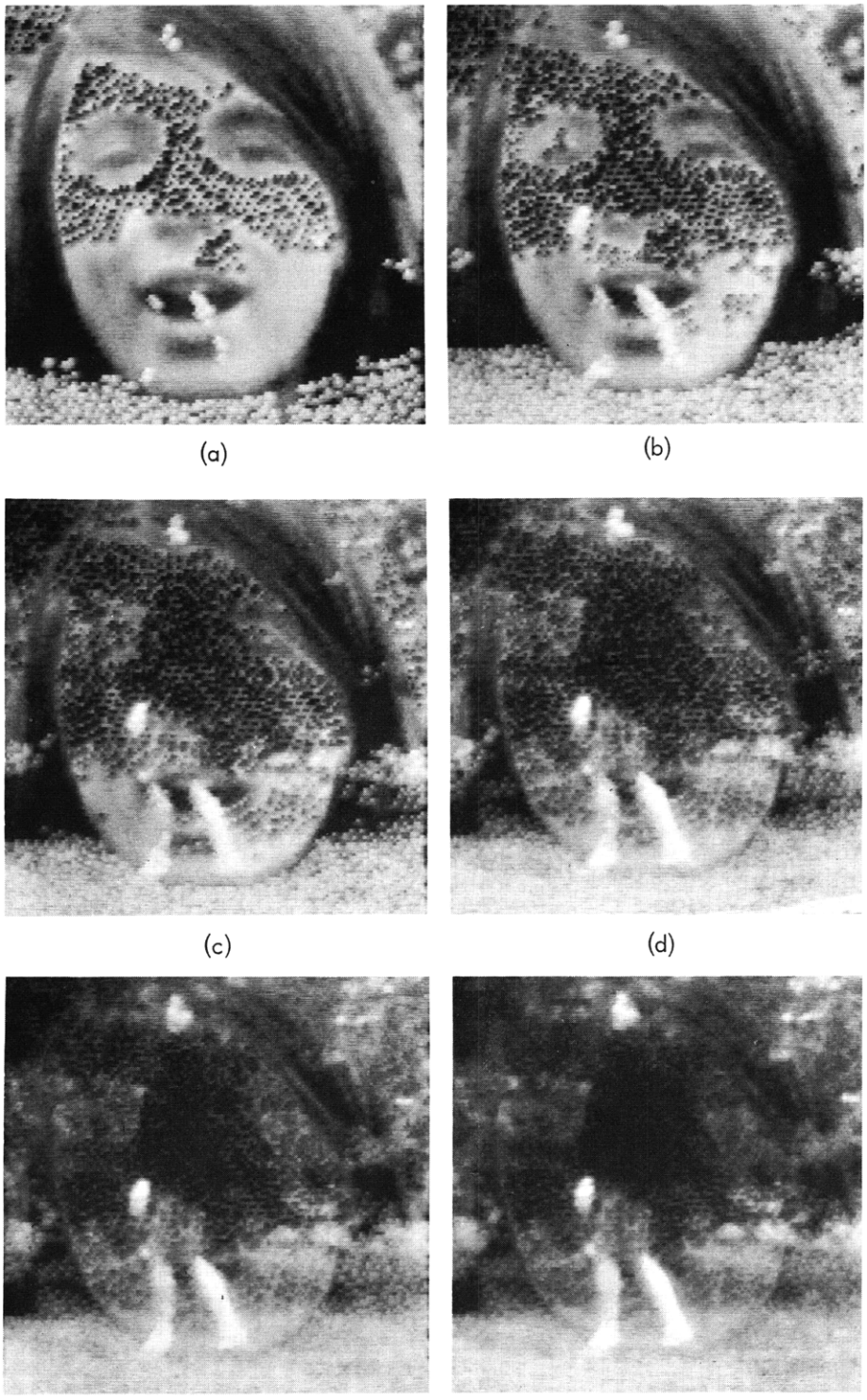


Fig. XV-2. Still pictures of scene transitions.

The simulations were performed under the assumption of error-free transmission and sufficient buffer storage at the transmitter and receiver to allow a signal match, as well as storage of complete pictures as required. A description of the operation of the transmission system follows.

The transmitter retains a copy of the picture that the receiver has in its memory; when confronted with the new picture data that are to be transmitted, the transmitter performs calculations to determine the N points that have the N largest brightness difference in the new pictures and the copy of the receiver picture. The transmitter then proceeds to correct the brightness values of these N points; also, each time a point is corrected, the correction value is averaged into its surround by means of a decaying interpolation function. Since the interpolation process usually changes the value of the brightness differences, these differences are recalculated each time that a correction is sent to prevent sending unnecessary corrections. The process of correcting points is carried out in an interlaced pattern that requires 4 passes to check every point in the picture. If the process completes the 4 passes without having sent all N corrections, it starts over with a lower difference threshold so that the full quota of corrections may be sent.

Since there is a constant number of corrections sent for each frame, there is no queing problem when a change of scene occurs. In Fig. XV-1 is shown an example of a picture build-up from a blank screen with the value N = 1024 used, which is equal to 1/16 of the points in the picture. The first four pictures are the frames representing the 4 frames after a blank screen; the fifth and sixth pictures are representative of the quality achieved after approximately 32 frame times. Figure XV-2 shows an example of a scene transition with the same number of corrections per frame. The six frames represent 3/8 second when viewed at the 16-frame/second rate. Since information concerning the location of these points, as well as the brightness values, must be sent, the reduction ratios are not 16:1 but approximately 6:1. This calculation is based on the upper limit of information required if all corrections are sent independently.

We plan to extend this work, and to introduce certain topological constraints on the interpolation process.

J. E. Cunningham

B. DISTANCE PROPERTIES OF TREE CODES

In this report we show that for any two positive integers r and s such that r divides s there exists a "good" tree code in the Galois field of order p^s with p^r branches per node. The code is good in the sense that an appropriately defined minimum distance criterion may be achieved which is numerically identical to that presented by Peterson[1] for the general parity-check code. This result is presented in the form of a theorem.

In the course of the proof, several lemmas are established which in themselves describe properties of tree codes which are thought to be of interest. Finally, it is shown that a "good" tree code may be transformed into a canonic form without destroying its distance properties. We shall first develop suitable notation.

Consider the n-tuple $(g_1, \ldots, g_n)$ whose coordinate entries are chosen from a Galois field of order p^s (G. F.$[p^s]$), where p is a prime and s is a positive integer. Suppose, also, that $g_1 \neq 0$ and that $n = kn_o$, where k and n_o are positive integers. We are interested in the set of n-tuples generated by certain linear combinations of the following basic set of n-tuples.

$$
\begin{array}{ccccccccccccccc}
g_1 & g_2 & \cdots & g_{n_o} & g_{n_o+1} & g_{n_o+2} & \cdots & & g_{2n_o+1} & \cdots & g_{(k-1)n_o+1} & \cdots & g_{kn_o} \\
0 & 0 & \cdots & 0 & g_1 & g_2 & \cdots & & \cdots & \cdots & \cdots & \cdots & g_{(k-1)n_o} \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & g_1 & \cdots & \cdots & \cdots & g_{(k-2)n_o} \\
\vdots & & & & & & & & & & & & \vdots \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \; g_1 & \cdots & g_{n_o}
\end{array}
$$

It is sometimes convenient to represent n-tuples in functional form. Thus we shall represent the n-tuple $(g_1, \ldots, g_n)$ by the function g(t) defined as

$$
g(t) = \begin{cases} g_i & \text{if } t = i \quad \text{for } i = 1, 2, \ldots, n \\ 0 & \text{all other } t \end{cases}
$$

Correspondingly, the array of n-tuples presented above may, at least for $t = 1, \ldots, n$, be represented by the set of functions

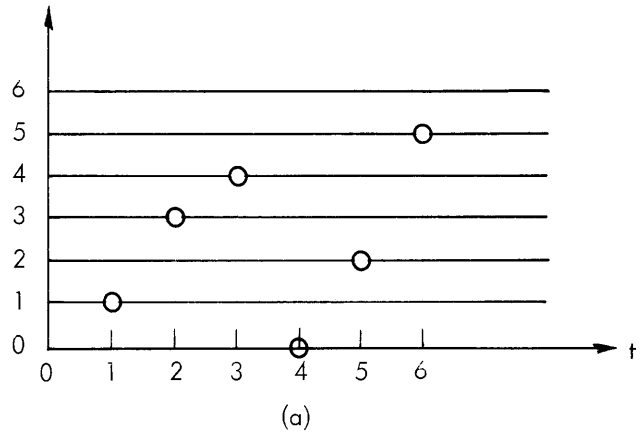
$$g(t),\ g(t-n_o),\ \ldots,\ g(t-(k-1)n_o).$$

Consider the set of functions (n-tuples) of the form[2]

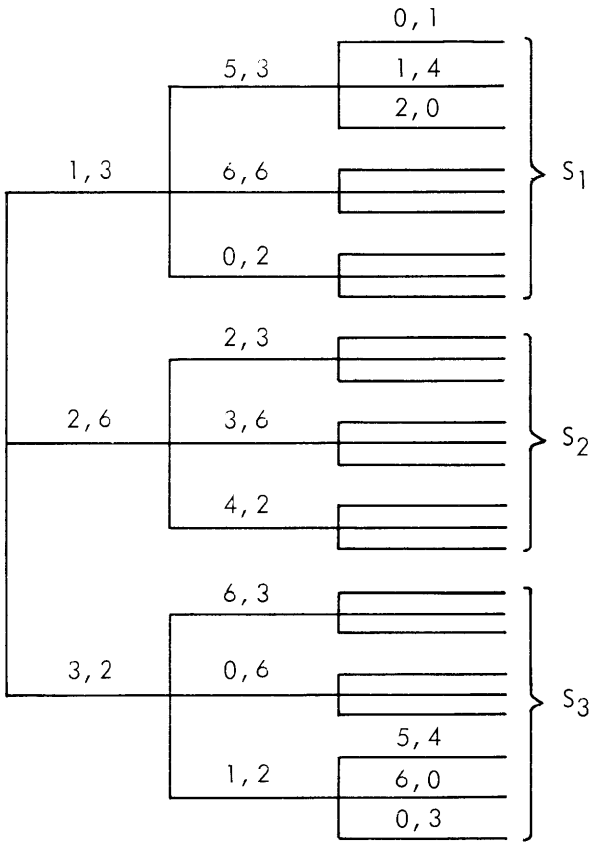
$$p(t) = x_o g(t) + x_1 g(t-n_o) + \ldots + x_{k-1} g(t-(k-1)n_o),$$

where the coefficients of the functions $g(t-in_o)$ for $i = 0, \ldots, k-1$ are chosen from a subset E of the Galois field. If E contains m elements, there are m^k such distinct functions, for the condition $g(1) \neq 0$ implies that the set of functions $g(t-in_o)$ $i = 0, \ldots, k-1$ is linearly independent.[3]

We shall refer to the set of functions generated this way as the tree code or convolutional code generated by g(t) and E. The name "tree code" derives from the fact that the elements (functions, code words, paths) of the code can be represented diagrammatically in the form of a tree. This is illustrated in Fig. XV-3 for a generator of length 6



(a)



(b)

Fig. XV-3. (a) Generator function g(t).
(b) Tree code generated by g(t) and the set of field elements 1, 2, 3.

with entries drawn from the Galois field of order 7. In this example it is assumed also that $n_o = 2$, and that the set E consists of the field elements 1, 2, and 3.

Returning to the general case, we denote the m elements of E as $e_o, e_1, \ldots, e_{m-1}$, and consider the subsets of functions (paths):

$$S_o = \left\{ p(t) \,\middle|\, p(t) = e_o g(t) + \sum_{i=1}^{k-1} x_i g(t-in_o) \right\}$$

$$\vdots$$

$$S_{m-1} = \left\{ p(t) \,\middle|\, p(t) = e_{m-1} g(t) + \sum_{i=1}^{k-1} x_i g(t-in_o) \right\},$$

where the x_i vary over all possible values of E.

In effect, we have partitioned the totality of paths into subsets with the same initial prefix. (See, for example, the path sets S_1, S_2, and S_3 in Fig. XV-3b.) Clearly, each subset S_i contains m^{k-1} paths.

We define the distance between any pair of distinct path sets S_i, S_j as

$$D(S_i, S_j) = \min_{\substack{p(t) \in S_i \\ h(t) \in S_j}} d(p(t), h(t)),$$

where $d(p(t), h(t))$ is the conventional Hamming metric,

$$d(p(t), h(t)) = \sum_{i=1}^{n} \| p(i) - h(i) \|.$$

Here, for any element x in the Galois field

$$\| x \| = 1 \qquad \text{if } x \neq 0$$

$$\| x \| = 0 \qquad \text{if } x = 0.$$

We define the weight of a vector to be equal to the distance of the vector from the all-zero vector.

If we let

$$\mathscr{D} = \min_{\substack{i,j \\ i \neq j}} D(S_i, S_j),$$

we can now state the theorem, which is the main result of this report.

THEOREM: If E is a subfield of the Galois field of order q, then there exists a generated tree code for which $\mathscr{D}$ exceeds Δ, where Δ is the largest

integer satisfying the inequality

$$\sum_{j=0}^{\Delta-2} \binom{n-1}{j}(q-1)^j < \frac{q^{n-1}}{m^{k-1}}.$$

REMARK 1: For q = m, this condition is identical to the one presented by Peterson for the general parity-check code.[1]

PROOF: We shall prove the theorem in a series of lemmas that serve as convenient guideposts to the main thread of the argument. After the proof, we shall discuss the effect of weakening the hypothesis by requiring only that E be an additive subgroup of the Galois field and not necessarily a subfield. This is of interest, since the requirement that E be a subfield is somewhat restrictive. The only subfields of a Galois field of order p^m are of order p^r, where r divides m.[4]

LEMMA 1: $D(S_i, S_j)$, $i \neq j$, is independent of the choice of i and j.

By definition,

$$D(S_i, S_j) = \min_{\substack{p(t) \in S_i \\ h(t) \in S_j}} d(p(t), h(t)).$$

But functions belonging to S_i and S_j must have respectively the forms

$$e_i g(t) + \sum_{l=1}^{k-1} x_l g(t-ln_o)$$

and

$$e_j g(t) + \sum_{l=1}^{k-1} y_l g(t-ln_o),$$

in which each of the 2(k-1) coefficients x_l and y_l is chosen from E. Thus

$$D(S_i, S_j) = \min_{\substack{\{x_l\} \\ \{y_l\}}} \left\{ \sum_{v=1}^{n} \left\| (e_i-e_j)\, g(v) + \sum_{l=1}^{k-1} (x_l-y_l)\, g(v-ln_o) \right\| \right\}.$$

Clearly, however, since x_l and y_l are chosen from E, which is a subfield and therefore an additive group, we have

$$D(S_i, S_j) = \min_{\{x_l\}} \left\{ \sum_{v=1}^{n} \left\| (e_i-e_j)\, g(v) + \sum_{l=1}^{k-1} x_l g(v-ln_o) \right\| \right\}.$$

Furthermore, $(e_i - e_j) \neq 0$ implies the existence of a multiplicative inverse $(e_i - e_j)^{-1}$, and thus

$$D(S_i, S_j) = \min_{\{x_l\}} \left\{ \sum_{v=1}^{n} \left\| g(v) + \sum_{l=1}^{k-1} (e_i - e_j)^{-1} x_l g(v - ln_o) \right\| \right\}.$$

But E, a field, implies that

$$\left\{ (e_i - e_j)^{-1} x \right\}_{x \in E} = \{x\}_{x \in E}.$$

Hence

$$D(S_i, S_j) = \min_{\{x_l\}} \left\{ \sum_{v=1}^{n} \left\| g(v) + \sum_{l=1}^{k-1} x_l g(v - ln_o) \right\| \right\}.$$

The right-hand side is independent of the choice of i and j, $i \neq j$, and thus Lemma 1 is proved.

Since E is a field, it must contain a zero element. It is notationally convenient to assume that, in fact, $e_o = 0$ and, correspondingly, S_o is the path set containing the all-zero n-tuple. We shall refer to S_o as the zero-path set. Adopting this notation, we note that the reasoning used to establish Lemma 1 proves almost directly the following lemma.

LEMMA 2: The quantity $\mathscr{D}$ is equal numerically to the weight of the minimum-weight path in the code which does not belong to S_o.

LEMMA 3: The number of distinct n-tuples that, in conjunction with the subfield E, generate the same code is exactly equal to $(m-1)m^{k-1}$.

Assume a code generated by the function g(t). By definition, $g(1) \neq 0$. The number of paths in this code which satisfy the constraint $p(1) \neq 0$ is equal to $(m-1)m^{k-1}$. Any such path, however, may be used to generate the same code. That is,

$$\left\{ \sum_{i=0}^{k-1} x_i p(t - in_o) \right\}_{x_i \in E} = \left\{ \sum_{i=0}^{k-1} x_i g(t - in_o) \right\}_{x_i \in E}.$$

In words, each and every one of the k^m n-tuples that may be expressed as a linear combination of g(t) and its translates may also be expressed as a linear combination of p(t) and its translates. We have thus established the fact that the number of distinct n-tuples that generate the same code is greater than or equal to $(m-1)m^{k-1}$.

However, any code generator must itself belong to the code, and, furthermore, cannot belong to S_o. This establishes the reverse inequality, and hence Lemma 3 is proved.

We shall say for the purposes of this report that two tree codes are essentially distinct if the only paths that they have in common belong to their respective zero path sets. We now formulate

LEMMA 4: Two generated tree codes are either essentially distinct or identical.

Assume two generated tree codes C_1 and C_2 that are not essentially distinct. Then there exists a path p(t) with nonzero first coordinate belonging to both C_1 and C_2. But such a path can serve as a generator both for C_1 and C_2. It follows, therefore, that C_1 must be identical to C_2.

LEMMA 5: The number of essentially distinct tree codes that can be generated with generators of length n in a Galois field of order q is exactly equal to $\frac{(q-1)q^{n-1}}{(m-1)m^{k-1}}$ where m is the order of the subfield E.

Initially, we note that if $q = p^s$, then E is a subfield only if $m = p^r$, where r divides s. This ensures that, in fact, $\frac{(q-1)q^{n-1}}{(m-1)m^{k-1}}$ is a positive integer.

The number of distinct code vectors in G. F.[q] with a nonzero first coordinate is equal to $(q-1)q^{n-1}$. Consider the code, say C_1, which is generated by one such vector. In C_1 there will be a total of $(m-1)m^{k-1}$ vectors with nonzero first coordinates. If $(q-1)q^{n-1} - (m-1)m^{k-1} > 0$, there exists an n-tuple with a nonzero first coordinate that is not in C_1. Consequently, we can generate a second code C_2 that is, by Lemma 4, essentially distinct from C_1. By proceeding in this fashion, it is clear that the number of essentially distinct codes that we can generate is exactly equal to $\frac{(q-1)q^{n-1}}{(m-1)m^{k-1}}$.

The final argument in the proof of the theorem is basically a comparison between the number of low-weight n-tuples with a nonzero first entry and the number of essentially distinct codes. Lemma 2 implies that $\mathscr{D}$, the minimum distance between any pair of distinct path sets in a code, is equal to the weight of the minimum-weight path in the code which does not belong to the zero subset. The total number of n-tuples of weight less than or equal to $\Delta-1$, which have a nonzero first coordinate, is

$$(q-1)\sum_{j=0}^{\Delta-2}\binom{n-1}{j}(q-1)^j.$$

In each code there will be at least m-1 code words of the same weight. Thus the total number of essentially distinct codes that can contain a path (in a nonzero subset) of weight less than Δ is, at most,

$$\left(\frac{q-1}{m-1}\right)\sum_{j=0}^{\Delta-2}\binom{n-1}{j}(q-1)^j.$$

Consequently, as long as the total number of essentially distinct codes exceeds this number, there will always exist a code with $\mathscr{D} \geq \Delta$. The theorem now follows from Lemma 5.

The proof of the theorem depends essentially on Lemma 4, which guarantees that two codes cannot partially overlap in a nontrivial way. More precisely, assume a code C_1 and an n-tuple p(t) with $p(1) \neq 0$ with the property that p(t) does not belong to C_1. Then Lemma 4 implies that the code generated by p(t) is essentially distinct from C_1. This is no longer true necessarily if E is only restricted to be an additive subgroup of the Galois field.

Consider, for example,[5] the Galois field of order 16 which can be represented as the field of polynomials over G. F.[2] with multiplication modulo the polynomial x^4+x+1. The four elements 1, x, 1 + x, 0 form an additive subgroup of G. F.[16]. Assume a generator whose first entry is the Galois-field element $1 + x + x^3$. The corresponding first entries in the four branches stemming from the first node will be

$$1(1+x+x^3) = 1 + x + x^3$$

$$x(1+x+x^3) = 1 + x^2$$

$$(1+x)(1+x+x^3) = x + x^2 + x^3$$

$$0(1+x+x^3) = 0.$$

Now consider a second generator whose first entry is the element $x^2 + x^3$. Such a generator clearly cannot belong to the code generated by the first generator. Yet

$$x(x^2+x^3) = 1 + x + x^3,$$

which coincides with one of the first-branch entries of the first code. It follows that by appropriate juggling it is possible to construct a pair of generators $g_1(t)$ and $g_2(t)$ so that $g_2(t)$ does not belong to the code generated by $g_1(t)$, which we designate C_1. But the code generated by $g_2(t)$, C_2, is still not essentially distinct from C_1. That is, there exists a path p(t) with $p(1) \neq 0$ which is common to both codes. Thus Lemma 4 is no longer true if the assumption that E is a subfield is replaced by the weaker assumption that E is an additive subgroup. Whether or not the theorem is true under these weaker conditions is not yet resolved – at least to the author's knowledge. In this direction we might point out that Lemma 2 can be shown to be true, although Lemma 1 is false, under the assumption that E is an additive subgroup.

We have shown that for any two positive integers r and s such that r divides s there exists a generator for a tree code in the Galois field of order p^s with p^r branches per node, which is good in the sense that an appropriately defined minimum distance criterion can be satisfied. Furthermore, the proof is constructive in the sense that by

simply choosing generators and testing the resulting codes one is guaranteed that ultimately a good code will be found.[6] Let us suppose that such a code has, in fact, been found with generator function g(t).

REMARK 2: (a) The code can be implemented by using only modulo p addition and multiplication.

(b) From such a code one can derive a code having the same distance properties which is in canonic form (to be described below).

We proceed to discuss Remark 2a.

A field of order p^r is a vector space of dimension r over the prime field of order p. We thus can select r elements from E, say $e_1, \ldots, e_r$ such that every element in E may be written in the form

$$\lambda_1 e_1 + \ldots + \lambda_r e_r$$

for some choice of $\lambda_1, \ldots, \lambda_r$ where $\lambda_1, \ldots, \lambda_r$ are elements in G. F.[p].

Consider the multiples of the generator function $e_1 g(t), \ldots, e_r g(t)$ (where multiplication is performed according to the rules of G. F.[p^s]), and designate the corresponding products $g_1(t), \ldots, g_r(t)$.

With each such function $g_1(t), \ldots, g_r(t)$ we can associate an n-tuple

$$\begin{matrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{r1} & \cdots & a_{rn} \end{matrix}$$

where the a_{ij} are elements of G. F.[p^s].

The Galois field of order p^s is, however, a vector space of dimension s over the prime field of order p. Correspondingly, we, in fact, can associate with each function $g_1(t), \ldots, g_r(t)$ an sn-tuple of prime-field elements

$$\begin{matrix} b_{11} & \cdots & b_{1(sn)} \\ \vdots & & \vdots \\ b_{r1} & \cdots & b_{r(sn)} \end{matrix}$$

where the b_{ij} are elements of G. F.[p].

Correspondingly, letting $\hat{g}_i(t)$ denote the expansion of $g_i(t)$ in prime-field elements, we can describe our code as the set of sn-tuples of the form

$$\sum_{i=0}^{k-1} \lambda_{1i} \hat{g}_1(t-isn_o) + \sum_{i=0}^{k-1} \lambda_{2i} \hat{g}_2(t-isn_o) + \ldots + \sum_{i=0}^{k-1} \lambda_{ri} \hat{g}_r(t-isn_o),$$

where the λ_{ij} are elements of the prime field.

In this formulation all addition and multiplication is done modulo p. Thus, for example,

$$\lambda_{11}(b_{11}, b_{12}, \ldots, b_{1(sn)}) = (\lambda_{11}b_{11}, \lambda_{11}b_{12}, \ldots, \lambda_{11}b_{1(sn)}),$$

where each entry on the right is interpreted modulo p.

We thus have shown that a code corresponding to a single generator function g(t) and the subfield E of order p^r may be viewed as a code corresponding to r generator functions $g_1(t), \ldots, g_r(t)$ and the prime field of order p. We turn now to Remark 2b.

We are concerned here with showing that it is possible to transform a code that is known to have good distance properties into canonic form and still preserve those distance properties.

We suppose that the n-tuple corresponding to g(t) has been transformed, as discussed in Remark 2a, into a set of r sn-tuples with entries in G. F.[p]. We again designate these sn-tuples by $\hat{g}_1(t), \ldots, \hat{g}_r(t)$.

Clearly, we do not alter the code generated by $\hat{g}_1(t), \ldots, \hat{g}_r(t)$ and the prime field G. F.[p] by

(1) replacing any generator $\hat{g}_i(t)$ by $\lambda\hat{g}_i(t)$ if λ is a nonzero element of G. F.[p];

(2) replacing $\hat{g}_i(t)$ by $\hat{g}_i(t) + \lambda\hat{g}_j(t)$, where λ is any element of G. F.[p];

(3) interchanging $\hat{g}_i(t)$ with $\hat{g}_j(t)$ for any i, j $1 \leq i, j \leq r$.

These operations are analagous to the "elementary row operations" of matrix theory by means of which it is possible to reduce any matrix into a row-reduced echelon form without affecting the space spanned by the matrix.[7]

If, in addition to these three operations, we allow column permutations of the generator array, then it is possible to derive from the original set of generators $\hat{g}_1(t), \ldots, \hat{g}_r(t)$ a new set of generators, say $p_1(t), \ldots, p_r(t)$, which have the forms

$$\begin{array}{ccccccc} 100 & \ldots & 0 & c_{11} & \ldots & c_{1(sn-r)} \\ 010 & \ldots & 0 & c_{21} & \ldots & c_{2(sn-r)} \\ \vdots & & & & & \\ 000 & \ldots & 1 & c_{r1} & \ldots & c_{r(sn-r)} \end{array}$$

respectively, where the c_{ij} belong to the prime field of order p.

In general, column permutations of the generator array will alter the code. Thus the code generated by $\hat{g}_1(t), \ldots, \hat{g}_r(t)$ may differ from the code generated by $p_1(t), \ldots, p_r(t)$. However, if the set of allowable column permutations is suitably restricted, the two codes will have the same metric structure.

It is important to keep in mind here that we are still measuring distance according to the rules of the Galois field of order p^s, even though all computations are carried out

modulo p. Thus to calculate the Hamming distance between two vectors it is necessary to compare their entries in successive blocks of length s. Each such block comparison can contribute only 1 to the cumulative Hamming distance between the two vectors, regardless of the actual number of discrepancies in the block which are in excess of 1.

Consider, for example, the two code vectors

$$d_1, d_2, \ldots, d_{sn}$$

$$f_1, f_2, \ldots, f_{sn}.$$

There are n blocks of length s to be compared. The first comparison involves a check of

$$d_1, \ldots, d_s$$

$$f_1, \ldots, f_s.$$

If there is disagreement in one or more places, these two blocks are distance 1 apart.

Compare, next, the s entries

$$d_{s+1}, \ldots, d_{2s}$$

$$f_{s+1}, \ldots, f_{2s} \quad .$$

Again, disagreement in one or more places adds 1 to the cumulative Hamming distance between the two code vectors. Clearly, the maximum distance between the two code vectors is equal to n.

It should be clear that any two columns that appear in the same block of length s can be permuted without changing the distance between the two code words.

Thus, for example, the distance between the vectors

$$d_1, d_2, \ldots, d_s, d_{s+1}, \ldots, d_{sn}$$

$$f_1, f_2, \ldots, f_s, f_{s+1}, \ldots, f_{sn}$$

is equal to the distance between the two vectors

$$d_s, d_2, \ldots, d_{s-1}, d_1, d_{s+1}, \ldots, d_{sn}$$

$$f_s, f_2, \ldots, f_{s-1}, f_1, f_{s+1}, \ldots, f_{sn}.$$

This is not true if columns from distinct blocks are interchanged. Thus, in general, the distance between the last two vectors given above is not equal to the distance between the vectors

$d_{s+1}, d_2, \ldots, d_s, d_1, d_{s+2}, \ldots, d_{sn}$

$f_{s+1}, f_2, \ldots, f_s, f_1, f_{s+2}, \ldots, f_{sn}$.

Consider now the first s entries of the generator functions $\hat{g}_1(t), \ldots, \hat{g}_r(t)$, namely,

$b_{11}, \ldots, b_{1s}$

$b_{21}, \ldots, b_{2s}$

$b_{r1}, \ldots, b_{rs}$.

In accordance with this discussion, we can permute the columns of this matrix of entries without affecting the metric structure of the portion of the code consisting of elements of the form

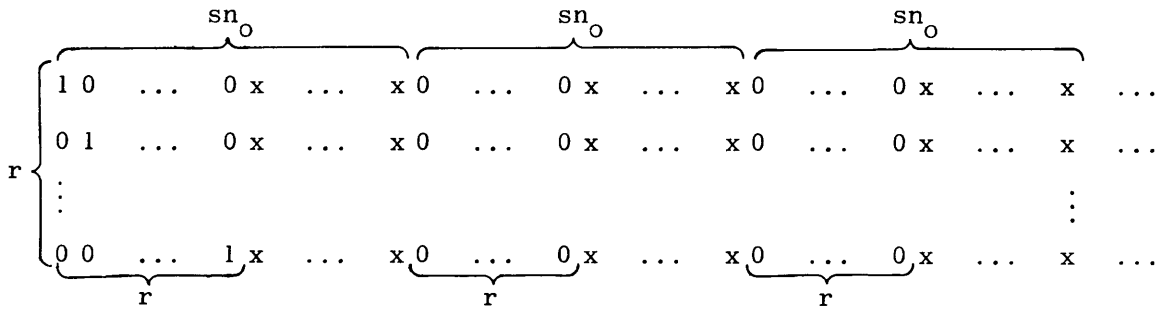
$$\lambda_1\hat{g}_1(t) + \ldots + \lambda_r\hat{g}_r(t),$$

where the λ_i are chosen from the prime field of order p.

The total code, however, involves translates of the generator functions. Thus the elements that appear in the first block of generator entries, in the process of forming the code, will interact with elements that appear in blocks numbered n_o+1, $2n_o+1$, ..., $(k-1)n_o+1$, and, in fact, only with these blocks. It follows that the metric structure of the code will be preserved if any permutations of the columns of block 1 are paralleled in these blocks. Thus if the first column of block 1 is interchanged with the third column of block 1, then the first column of block n_o+1 should be interchanged with the third column of block n_o+1, and similarly for blocks numbered $2n_o+1$, ..., $(k-1)n_o+1$.

The manipulations discussed above permit us to derive the set of generators $p_1(t)$, ..., $p_r(t)$ whose first r entries are in diagonal form, as shown earlier.

Finally, we note that if $h_1(t), \ldots, h_r(t)$ are any paths belonging to the code subsets containing $p_1(t), \ldots, p_r(t)$, respectively, then the code generated by $h_1(t), \ldots, h_r(t)$ is identical to the code generated by $p_1(t), \ldots, p_r(t)$.[8] Using this fact, we can extract a set of generators for the code which are in canonic form. That is to say, the generators may be written in the form



where the symbol x denotes some entry from the prime field. (Note that entries in different positions will, in general, have different values.)

We conclude this report with an example designed to illustrate Remark 2. We consider a generator with entries drawn from G. F.[16] (see Peterson[5]) and assume that $n = 6$, $n_o = 2$, and $m = 4$. As noted earlier, G. F.[16] may be represented as the field of polynomials over G. F.[2] with multiplication performed modulo the polynomial $1 + x + x^4$. The subfield E of order 4 consists of the elements 1, $x + x^2$, $1 + x + x^2$, 0. Clearly E, considered a vector space of dimension 2 over the binary field, is spanned by the two basis vectors 1 and $x + x^2$. Let us assume a generator function g(t) of the form

$$1 + x + x^3,\ x + x^2 + x^3,\ 1 + x^2 + x^3,\ x + x^3,\ x^2 + x^3,\ 1 + x^2.$$

Consider the pair of functions (1)g(t) and $(x+x^2)$g(t):

$$1 + x + x^3,\ x + x^2 + x^3,\ 1 + x^2 + x^3,\ x + x^3,\ x^2 + x^3,\ 1 + x^2$$

$$1 + x + x^2 + x^3,\ x,\ x^3,\ 1 + x^3,\ x + x^2 + x^3,\ 1 + x^2 + x^3.$$

Denoting these two 6-tuples by $g_1(t)$ and $g_2(t)$, respectively, we may write the corresponding 24-tuples $\hat{g}_1(t)$ and $\hat{g}_2(t)$ as

110101111011010100111010

111101000001100101111011.

Replacing $\hat{g}_2(t)$ by $\hat{g}_1(t) + \hat{g}_2(t)$, we get the pair

110101111011010100111010

001000111010110001000001.

We can put the first two entries of this array into diagonal form by interchanging columns 2 and 3. Correspondingly, in order to preserve the distance properties of the tree code generated by these two vectors, we must further interchange column 10 with 11 and column 18 with column 19. We denote the resulting pair of vectors

101101111101010101011010

010000111100110000100001

as $p_1(t)$ and $p_2(t)$, respectively.

The code generated by $p_1(t)$ and $p_2(t)$ consists of the 2^6 vectors

$$\sum_{i=0}^{2} \lambda_{1i} p_1(t-i8) + \sum_{i=0}^{2} \lambda_{2i} p_2(t-i8),$$

where the λ_{ji} are elements of the binary field.

The canonic generators for this code are the two vectors

$$h_1(t) = p_1(t) + p_1(t-8) + p_2(t-8) + p_2(t-16)$$

$$h_2(t) = p_2(t) + p_1(t-8) + p_2(t-8).$$

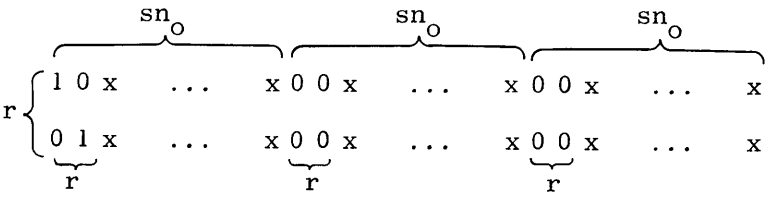
Note that $h_1(t)$ belongs to the path subset containing $p_1(t)$, and $h_2(t)$ belongs to the path subset containing $p_2(t)$. Computing $h_1(t)$ and $h_2(t)$ explicitly, we find that they are equal to the pair of 24-tuples

101101110010000100000000

010000110011100000111000,

respectively.

The important thing to note about this array is that it is of the form



The author wishes to acknowledge his indebtedness to Professor John M. Wozencraft for stimulating this research. The basic comparison of the number of low-weight code words and essentially distinct codes, which was used to prove the theorem, is a generalization of his argument for binary fields.[9] The observation that codes may be transformed into canonic form without destroying their metric structure is also due to him.

H. Dym

References

1. W. W. Peterson, Error-Correcting Codes (The M. I. T. Press, Cambridge, Mass., and John Wiley and Sons, Inc., New York, 1961), p. 51.

2. p(t) is only of interest for t = 1, ..., n; the n-tuple corresponding to p(t) is, of course, (p(1), p(2), ..., p(n)).

3. It actually would have been sufficient to require only that one of the first n_o entries of g(t) be nonzero in order to guarantee the linear independence of g(t) and its translates.

4. R. D. Carmichael, Introduction to the Theory of Groups of Finite Order (Dover Publications, Inc., New York, 1956), p. 260.

5. This example, for ease of reference, was chosen to coincide notationally with a discussion of G. F.[16] which appears in Peterson, op. cit., p. 100.

6. This may require a very long time, however, especially when n and k are large.

7. K. Hoffman and R. Kunze, Linear Algebra (Prentice-Hall, Inc., Englewood Cliffs, N. J., 1961).

8. Recall, for example, that $h_1(t)$ belongs to the same subset as $p_1(t)$ if and only if $h_1(t) = p_1(t)$ for $t = 1, 2, \ldots, sn_o$.

9. J. M. Wozencraft and B. Reiffen, Sequential Decoding (The M. I. T. Press, Cambridge, Mass., and John Wiley and Sons, Inc., New York, 1961), pp. 51-54.

C. MOMENTS OF THE SEQUENTIAL DECODING COMPUTATION

Recent investigations of a sequential decoding algorithm[1] for the memoryless binary erasure channel provide results with implications for the behavior of sequential decoding on the general, memoryless channel.

The behavior of the moments of the sequential decoding computation as a function of rate on the erasure channel has been determined. The n^{th} moment grows exponentially with constraint length for rates $R > R_n$, $n = 1, 2, 3, \ldots$. The rates $\{R_n\}$ form a monotonically decreasing sequence with an interesting geometrical interpretation. Plot the exponent, E(R), on the "sphere-packed" probability of error versus R (see Fig. XV-4).

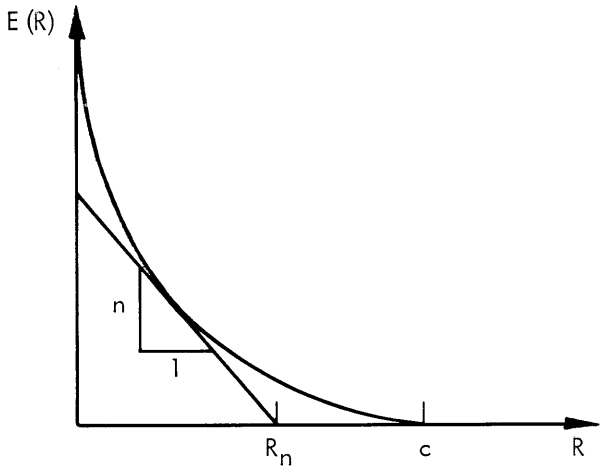


Fig. XV-4. Rate construction.

Draw a line with slope equal to -n tangent to E(R). Then the rate-axis intercept of this straight line is the rate R_n. This is a very natural extension of the geometrical interpretation of $R_{comp} \equiv R_1$. Because the rates $\{R_n\}$ have such a natural interpretation and because sequential decoding on the erasure channel exhibits the fundamental features that are exhibited on more general channels, we are inclined to believe that these results also apply to the general, memoryless channel. Preliminary investigations indicate that this is true.

The behavior of the rates $\{R_n\}$ implies something about the character of the

distribution of computation, $P_R[x \geq N]$. In particular, if we assume that the fractional rates can be obtained by the same construction technique as the integral rates, then the Pareto distribution

$$P_R[x \geq N] = \frac{A}{N^{\beta}}, \text{ N large and } \beta \text{ such that } R = R_{\beta}$$

provides the correct moment behavior. This problem, as well as those mentioned earlier, is still under study.

J. E. Savage

References

1. For a description of the algorithm employed, see J. E. Savage, Sequential decoding for an erasure channel with memory, Quarterly Progress Report No. 69, Research Laboratory of Electronics, M. I. T., April 15, 1963, pp. 149-154.