RCA Review

Special Effects in Digital TV

March 1981 Volume 42 No. 1

RCARCI 42(1) 1-130(1981)

RCA Review, published quarterly in March, June, September and December by RCA Research and Engineering, RCA Corporation, Princeton, New Jersey 08540. Entered as second class matter July 3, 1950 under the Act of March 3, 1879. Second-class postage paid at Princeton, New Jersey, and at additional mailing offices. Effective January 1, 1978, subscription rates as follows: United States and Canada: one year $8.00, two years $14.00, three years $18.00; in other countries, one year $8.60, two years $15.20, three years $19.80. Single copies (except for special ʻssues) up to five years old $3.00.

RCA Review

RCA Review (ISSN 0033-6831) is a technical journal published quarterly by RCA Research and Engineering in cooperation with the subsidiaries and divisions of RCA.

Contents

March 1981 Volume 42 Number 1 ISSN 0033-6831

Spatial Compression and Expansion of Digital Television Images

Glenn A. Reitmeier

RCA Laboratories, Princeton, NJ 08540

Abstract—Video signals are one-dimensional projections of three-dimensional information, two spatial and one temporal. It is desired to process the spatial information contained in a television signal in order to compress and expand video images, thus providing an "electronic zoom" capability. Of course, this must be accomplished while maintaining the color information and the proper video format. Further, it is desired to alter the horizontal and vertical sizes of the picture independently, so as to change the aspect ratio of the image. Digital signal processing techniques are utilized to achieve this goal. Several algorithms are presented, analyzed, and their effects on actual images shown. These algorithms range from simple methods of operating directly on a composite NTSC video signal, which produce fixed size compressions, to a method of operating on a component video signal, which is derived from the application of linear systems theory. Further modifications to this technique produce compressed and expanded images of extremely high quality, while maintaining the computational simplicity required for real time implementation. Various tradeoffs affecting image quality are discussed, and finally, other special effects based on compression are suggested and illustrated.

1. Introduction

While television has been in existence for many years, the advent of digital television is bringing about a new wave of sophistication in the broadcast industry. In particular, the application of digital signal processing techniques to video signals allows practical utilization and processing of the multidimensional nature of video information.

Composite digital television systems perform sampling and quantization of the analog NTSC video signal. The sampling rate is usually constrained to be an integer multiple of the color subcarrier frequency, in order to minimize visible beats associated with the use of practical A/D and D/A converters.[1] The most widely accepted composite digital video system uses linearly quantized 8-bit pulse code modulation (PCM) at a sampling rate of four times the color subcarrier, and this is the system that will be utilized for this investigation. Since sampling is performed at a 14.32 MHz rate, and processing must be done in real time, digital computations must be performed in a time segment of 69 ns. Thus, the use of such a high sampling rate imposes severe constraints on hardware implementation, since the speed required to process the information necessitates the use of state-of-the-art digital technology. Nevertheless, the form of the composite NTSC signal and the four times subcarrier sampling rate lead to several important and convenient relationships.

There are 910 picture samples (or pixels) in each video line, 768 of which are considered visible, or "active", pixels. Because of the relationship of sampling rate and subcarrier frequency, pixels that are separated by multiples of four samples have identical subcarrier phase. Furthermore, vertically adjacent pixels, and those which are separated by two samples, have an inverse phase relationship. These relations occur for two reasons. First, the subcarrier frequency is such that its phase is inverted on adjacent video lines. Secondly, since four samples are taken each subcarrier cycle, they are separated by 90° phase increments. If the sampling phase is referenced to the cosine component of the quadrature modulation (which is referred to as sampling on the in-phase or I axis), then each sample falls on a zero crossing of one of the modulating signals and at the peak of the other, since the two signals also have a 90° phase difference. Because of the form of the composite signal, sampling on the I axis yields sequences of four successive pixels whose values are represented by $Y + I$, $Y + Q$, $Y - I$, $Y - Q$, where, as usual, Y is the luminance signal and I and Q are the in-phase and quadrature color components. Since this results in an easy method of digital demodulation, it is one of the great advantages of four times subcarrier sampling, as is the vertical alignment of pixels on adjacent lines.[2]

Video compression and expansion refers to the function of altering the apparent size of a television picture. Previously, this function could only be performed optically, directly at the television camera, via a zoom lens. However, we wish to develop an "electronic zoom" capability that can produce an image of variable size from an already existing video sequence. Further, it is desirable to alter the horizontal and vertical size independently in order to change the aspect ratio of the picture, which cannot be accomplished using conventional means. This function has various applications in the broadcast industry, ranging from special effects to providing a simple means of translating cinemascope motion pictures to television by appropriately changing the aspect ratio. To accomplish this goal, the spatial and color information contained in the NTSC video signal must be processed and put back into standard video form, certainly not a trivial task given the highly complex nature of the video signal.

Because of the great complexity of video signals, which simultaneously represent three dimensional information for three separate signals (red, green, and blue components), analytic expressions for image quality are simply impossible to deal with. Furthermore, due to the perception of human vision, traditional error criterion have been found to be not applicable for video signals. It is fully accepted in the broadcast industry that the *only* acceptable criterion for the comparison of image quality is the subjective reaction of human observers.[3] Therefore, this is the method used in the evaluation of various algorithms, which are compared to optimize the final observed picture quality. Also, the source images that are used in algorithm evaluation are critical. Typical scenes (scenery, faces, groups of people, etc.) contain very little high-resolution luminance, and are quite forgiving. To provide the most difficult possible case for the evaluation of algorithms, the monoscope test pattern will be utilized, this being a black-and-white test pattern with extremely high

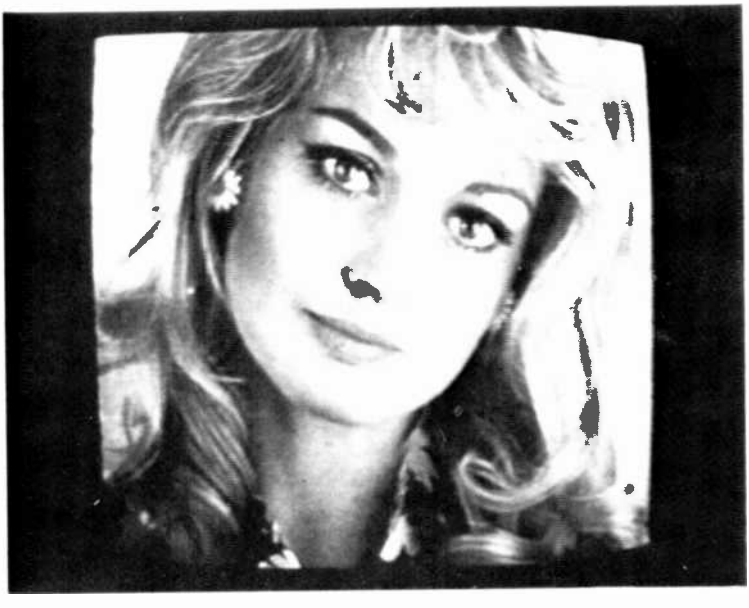


Fig. 1—Typical test slide to be used for algorithm comparison (one field).

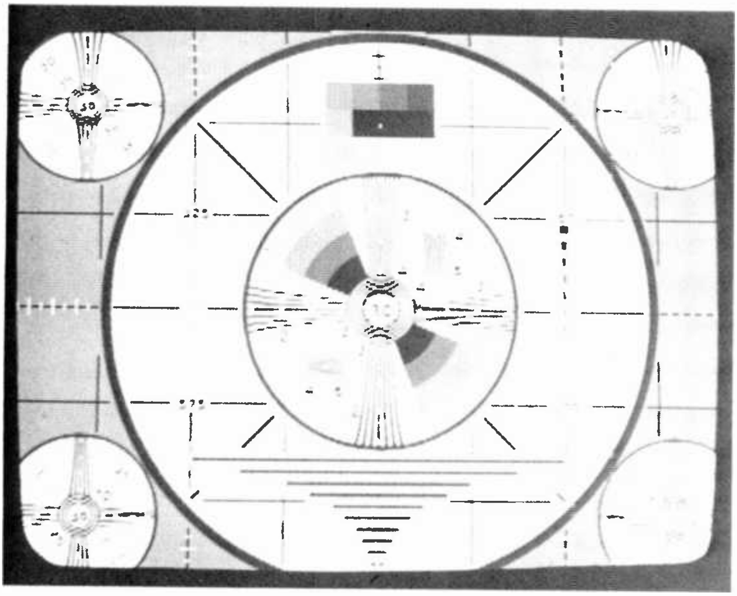


Fig. 2—Monoscope pattern used to test algorithms. The use of a single field is evidenced by the foldover effect in the center of the pattern.

resolution. In fact, the luminance bandwidth of the monoscope exceeds that available for color signals, so we are truly providing the severest possible source image. A typical test slide and the monoscope pattern are shown in Figs. 1 and 2, respectively, as a reference for the quality of their reduced representations.

All of the images presented in this paper were produced using the RCA Laboratories Digital Video Facility, which is a minicomputer based image processing system that allows programmable processing of digital video information (8-bit PCM) at non-real-time rates. Digitization and playback functions are, of course, performed in real time. One field of four-times-subcarrier sampled video was utilized in all cases, resulting in a visible foldover effect in some portions of the monoscope, since only one half of the full vertical resolution is present. However, the comparison of source and processed images is the important criterion, and this is certainly unaffected by the use of a single field.

2. Theoretical Considerations for Image Compression

Consider a continuous, black-and-white image as a two-dimensional spatial signal $f(x, y)$, which can be described by a two-dimensional Fourier transform,

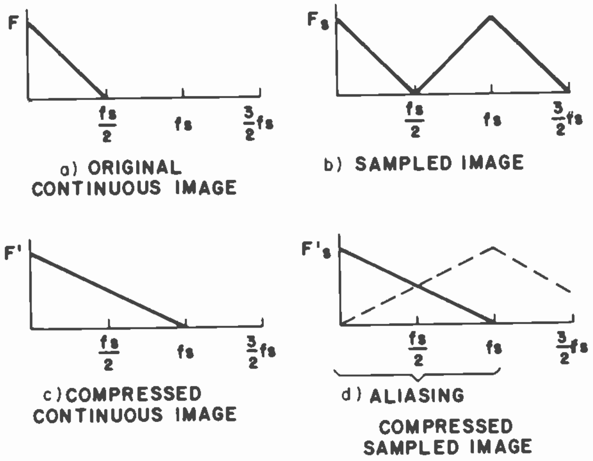
$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\, e^{-j(ux+vy)}dxdy.$$

The scaling theorem indicates that if the signal is compressed in space, then its Fourier transform will expand in frequency, i.e.,

$$f\left(\frac{x}{a},\frac{y}{b}\right) \leftrightarrow |ab|\ F(au, bv).$$

Therefore, to obtain a ¼-size continuous image that maintains its aspect ratio, the image must be reduced to ½ its original size in both the horizontal and vertical dimensions, thus doubling the horizontal and vertical spatial frequencies. If an image is to be composed of discrete picture elements (pixels), it can be considered as the sampled version of an ideal, continuous image, and is subject to minimum sampling at the Nyquist rate in each dimension. Let us examine the problem of reducing the size of a sampled image, and displaying this image with the same spatial resolution (sampling frequency) as the original.

To illustrate, let us examine only one spatial direction, since the effect will be the same in each, and assume that the original, continuous image is bandlimited to a spatial frequency $f_s/2$. Fig. 3a shows the spectrum of the original image and Fig. 3b the sampled version. In the case of ¼-size compression, the spatial frequencies in each direction are doubled, and this causes aliasing over the entire bandwidth of the reduced and sampled image (Fig. 3d). Of course, the solution to this problem is to low-pass filter the original image with cutoff $f_c = f_s/4$, as shown in Fig. 4a. Now the compressed and sampled version exhibits no aliasing at all



a) ORIGINAL CONTINUOUS IMAGE

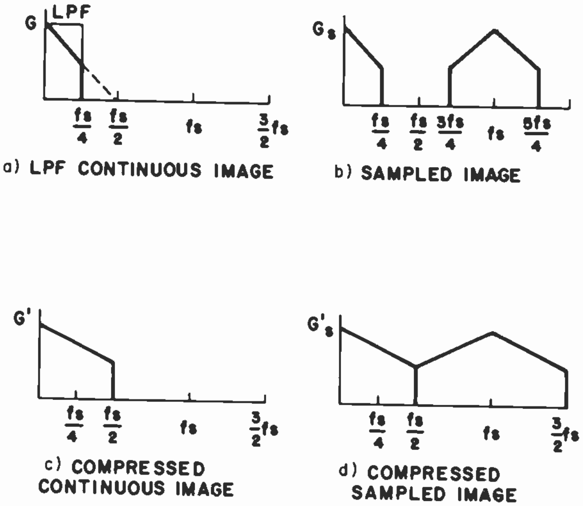
b) SAMPLED IMAGE

c) COMPRESSED CONTINUOUS IMAGE

d) ALIASING

COMPRESSED SAMPLED IMAGE

Fig. 3—Compression effects and aliasing.



a) LPF CONTINUOUS IMAGE

b) SAMPLED IMAGE

c) COMPRESSED CONTINUOUS IMAGE

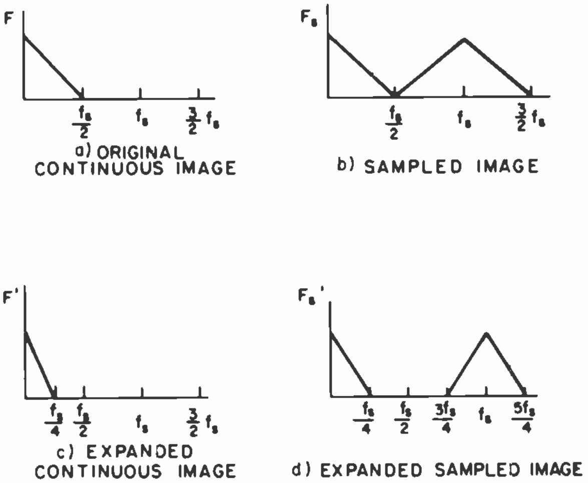
d) COMPRESSED SAMPLED IMAGE

Fig. 4—Proper compression.

(Figs. 4c and 4d). While this image does not contain all of the information of the original, it is the best possible representation under the circumstances.[4] In general, in order to display a compressed image at the same sampling rate as the original, we have the same problem as sampling a continuous image with bandwidth greater than the Nyquist frequency, and the signal must be low-pass filtered in such a manner that aliasing will not occur.

From the preceding discussion, we draw several conclusions about the effects of compression on a video signal. Since the luminance signal represents horizontal and vertical spatial frequencies, it must show a frequency expansion, when an image is compressed in size, corresponding to the projection of the expanded spatial frequencies. Since the color information must align itself with the luminance signal for proper color reproduction, it is evident that the baseband chrominance frequencies must undergo the same frequency expansion as the luminance components. However, since the NTSC format must be preserved, the color subcarrier must remain unchanged and the chrominance frequency expansion must occur relative to it. Further, prefiltering must also be accomplished in the video format, with low-pass and band-pass structures for horizontal luminance and chrominance components, respectively, and comb filters for vertical components. Comb filters are the appropriate representations, since they correspond to projected two-dimensional low-pass filters in the vertical direction.

When an image is spatially expanded, its Fourier transform is com-



a) ORIGINAL CONTINUOUS IMAGE

b) SAMPLED IMAGE

c) EXPANDED CONTINUOUS IMAGE

d) EXPANDED SAMPLED IMAGE

Fig. 5—Expansion effects.

pressed in frequency, as shown in Fig. 5. Naturally, this impacts the video signal in much the same way as compression. Thus, luminance and chrominance components must show the appropriate frequency compression, while the color subcarrier remains unchanged. While expansion does not require prefiltering, as compression does, some postfiltering may be advantageous, and this must again be accomplished in the video format.

Thus far, the case of scaling has been treated mathematically, and how this might be accomplished in the discrete time domain has not been indicated. The signal to be scaled is the four times subcarrier sampled video, and the output must also be of the same format. Thus, we must transform a set of pixels representing a signal into a smaller set of pixels, with the same resolution as the original set, that will represent a scaled version of the original signal. It is this process, and the requirement of preserving the video format, that presents the problem to be solved. Several algorithms to accomplish this function will be presented and analyzed.

3. Compression by Video Data Deletion

Because of the extremely high data rate involved in digital video, which must be processed in real time, it is very desirable from an implementation standpoint to minimize the number of arithmetic operations that must be performed. The simplest possible type of compression algorithm is one that operates directly on the composite video signal, and simply deletes an appropriate number of pixels and entire scan lines, while those maintained are concatenated and output successively. Algorithms of this type have been developed empirically within practical hardware implementation constraints, rather than on a strictly theoretical basis.

Let us consider an algorithm to produce a color picture that is ½-size in both the horizontal and vertical dimensions, i.e., it produces an image that covers ¼ of the original area. The first approach would seem to be to simply delete every other pixel and line for compression in the horizontal and vertical dimensions, respectively. However, this approach is not possible within the constraints of the NTSC video format. Since the subcarrier phase must alternate on adjacent lines, the compressed image would have only lines of one phase if every other line were deleted. Therefore, lines must be deleted in pairs to keep the proper subcarrier relation. Also, since four samples are required to define a single subcarrier cycle, portions of the picture will be deleted and maintained in blocks of 4 pixels, since this will satisfy the proper video format. In summary, then, to produce a ¼-size picture, the following algorithm can be applied:

(1) maintain four samples
(2) delete the next four samples
(3) steps 1–2 are continued until the end of the scan line
(4) the same procedure (1–3) is applied to the next line
(5) the next two scan lines are deleted

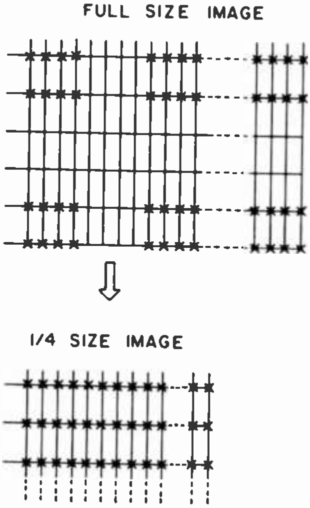


Fig. 6—Video data deletion (¼-size compression algorithm).

Fig. 6 illustrates this compression algorithm.

Because this algorithm utilizes a nonlinear subsampling and concatenation process, it is not easily described by conventional analytic expressions. Therefore, let us show that it does provide an approximation to an ideally compressed image, and discuss the undesirable effects associated with the algorithm.

Consider a low frequency luminance component f_L of the NTSC signal, sampled as shown in Fig. 7. Applying the compression algorithm, we alternately maintain and delete segments of four samples, concatenating them to produce the compressed result, also shown in Fig. 7. Although the signal has been distorted, we see that the dominant effect is to halve the period T_L of the signal, so that $T'_L = T_L/2$ and $f'_L = 2f_L$. Therefore, the resulting waveform will contain a fundamental frequency that has been doubled, with harmonics related to the distortion introduced. As seen from Fig. 7, the amount of distortion will certainly increase at higher luminance frequencies, and we expect this to produce objectionable picture defects.

Since four adjacent samples define a subcarrier cycle, maintaining and deleting segments of four pixels completely maintains the subcarrier frequency, as shown in Fig. 8. Since quadrature modulation is both an amplitude and phase modulation process, any complete color subcarrier

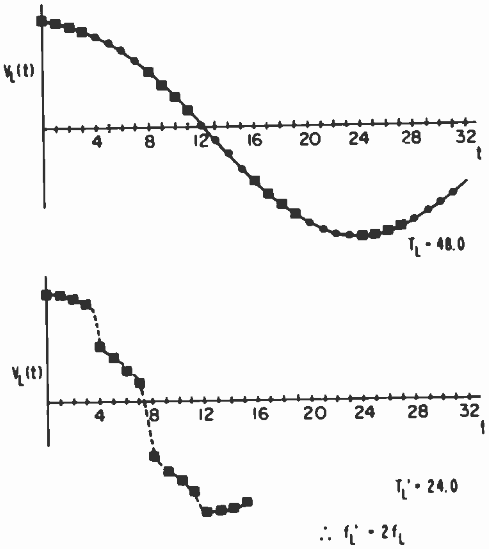


Fig. 7—Effect of video data deletion on a typical luminance component.

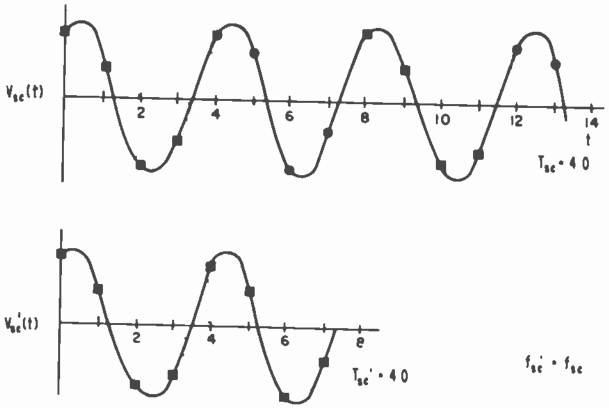


Fig. 8—Effect of video data deletior on the color subcarrier.

cycle is sufficient to define the color content in that area of the picture, and is therefore specified by four samples.

To verify the rather intuitive analysis, this algorithm was implemented in software together with a Fast Fourier Transform in order to provide a numerical confirmation of its effect. The results confirmed the analysis.

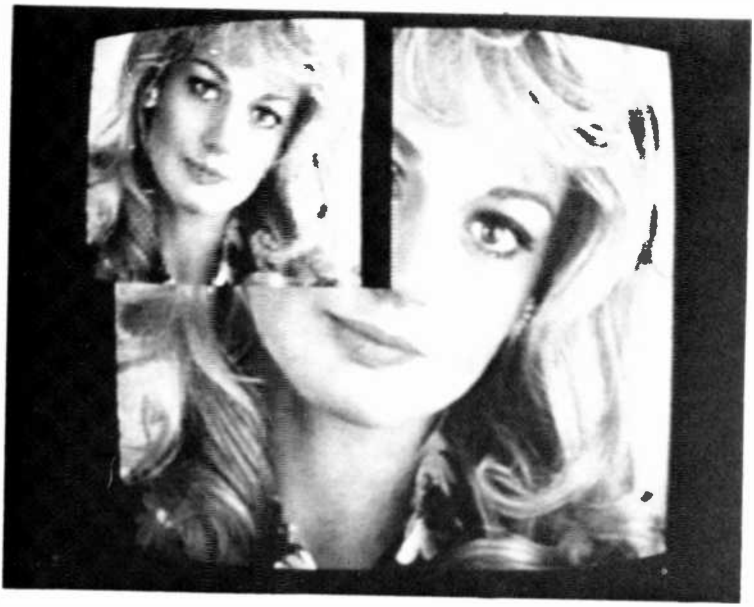


Fig. 9—One-quarter-size image produced by the video data deletion method and inset in the original image.

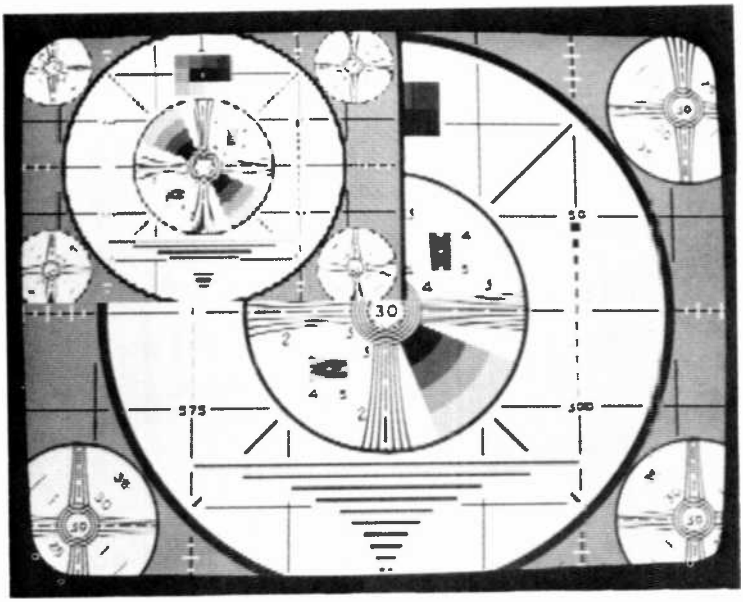
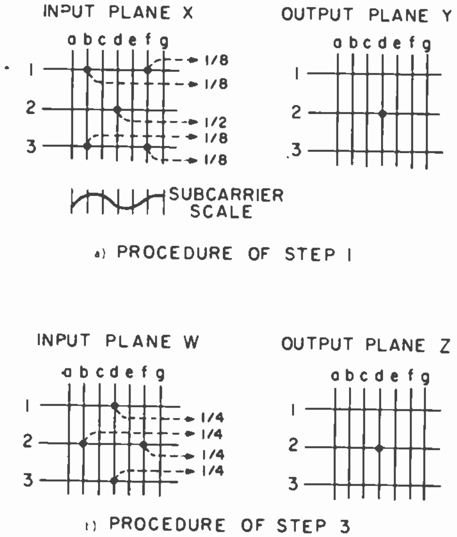


Fig. 10—One-quarter-size monoscope produced by the video data deletion method.

There are two undesirable effects associated with this method of image compression. First, samples have merely been deleted, leaving fewer samples to represent the image. Of course, with no prefiltering, this process introduces severe aliasing effects. Second, those samples maintained are taken and concatenated in a nonlinear manner to approximate what should ideally have been accomplished by a linear subsampling process. The effect is to maintain certain relatively large blocks of the image, introducing a coarse spatial quantization. While this algorithm certainly does not produce a high quality compressed picture, it performs surprisingly well on typical off-air video. since these scenes generally contain very little high frequency information, and are rather forgiving, as previously mentioned. A typical test slide is shown in Fig. 9, with the compressed picture inserted in the upper left hand corner. For comparison, Fig. 10 shows the effect of the algorithm on the monoscope test pattern. where it produces an extremely poor compression. Thus, we see the necessity of using a worst-case image as a standard for algorithm evaluation.

4. Compression by Subcarrier Reconstruction

The following method of ¼-size image compression was developed em-



a) PROCEDURE OF STEP 1

b) PROCEDURE OF STEP 3

Fig. 11—Picture element weightings for subcarrier reconstruction.

pirically by R. Hopkins and T. Gurley of RCA Broadcast Systems. The design philosophy was that in order to obtain a smooth compressed picture, surrounding spatial information would be averaged into the pixels that are maintained to represent the compressed image. Further, the points selected for averaging were selected according to their being either in phase with or 180° out of phase with the maintained pixel. Then, a sequence of averages would be performed that would reproduce the subcarrier frequency, based on the half-wave symmetry of four-times subcarrier sampling. As will be shown, this method satisfies the theoretical requirements for image compression, while operating directly on the composite video signal. The algorithm is as follows.

(1) Apply the procedure shown in Fig. 11a to two consecutive samples. Every x_{ij} and y_{ij} represents a sample. Each "new" sample y_{2d} in the output plane is computed from a weighting of "old" samples x_{1b}, x_{1f}, x_{3b}, x_{3f} in the input plane. That is,

$$y_{2d} = \frac{x_{2d}}{2} + \frac{x_{1b}}{8} + \frac{x_{1f}}{8} + \frac{x_{3b}}{8} + \frac{x_{3f}}{8}.$$

(Note that y_{2d} has the same subcarrier phase as x_{2d}.)

(2) Delete the next two samples.

(3) Apply the procedure shown in Figure 11b to the next two consecutive samples. Here, each "new" sample z_{2d} in the output plane is

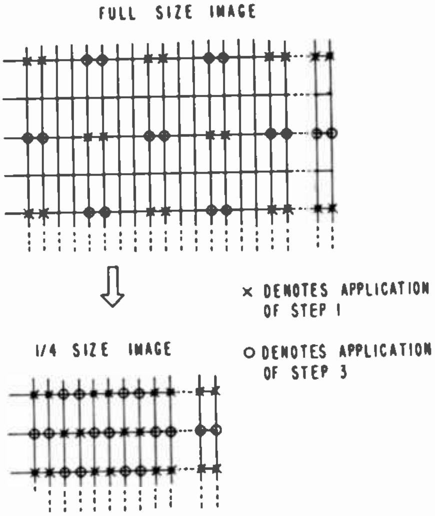

FULL SIZE IMAGE
1/4 SIZE IMAGE
× DENOTES APPLICATION OF STEP 1
O DENOTES APPLICATION OF STEP 3

Fig. 12—Subcarrier reconstruction ¼-size compression algorithm.

computed from a weighting of "old" samples w_{1d}, w_{2b}, w_{2f}, w_{3d} in the input plane. That is,

$$z_{2d} = \frac{w_{1d}}{4} + \frac{w_{2d}}{4} + \frac{w_{2f}}{4} + \frac{w_{3d}}{4}.$$

(Note that z_{2d} is 180° out of phase from w_{2d}.)

(4) Delete the next two samples.

(5) Steps 1–4 are continued until the completion of a scan line.

(6) The following scan line is deleted.

(7) Repeat steps 1–5 on the next scan line, except beginning now with step 3.

Fig. 12 illustrates this ¼ size compression algorithm.

The weighting procedures in steps 1 and 3 of the algorithm are, in fact, digital FIR (finite impulse response) filters. Let us denote the output plane sequences in steps 1 and 3 as $y_1(n)$ and $y_2(n)$, respectively, and the input plane as $x(n)$. Letting m equal the number of samples per scan line, we can write

$$y_1(n) = \frac{x(n)}{2} + \frac{x(n-m-2)}{8} + \frac{x(n-m+2)}{8} + \frac{x(n+m-2)}{8} + \frac{x(n+m+2)}{8}$$

$$y_2(n) = \frac{x(n-m)}{4} + \frac{x(n-2)}{4} + \frac{x(n+2)}{4} + \frac{x(n+m)}{4}$$

Since these equations are the time domain form of FIR filters, the digital

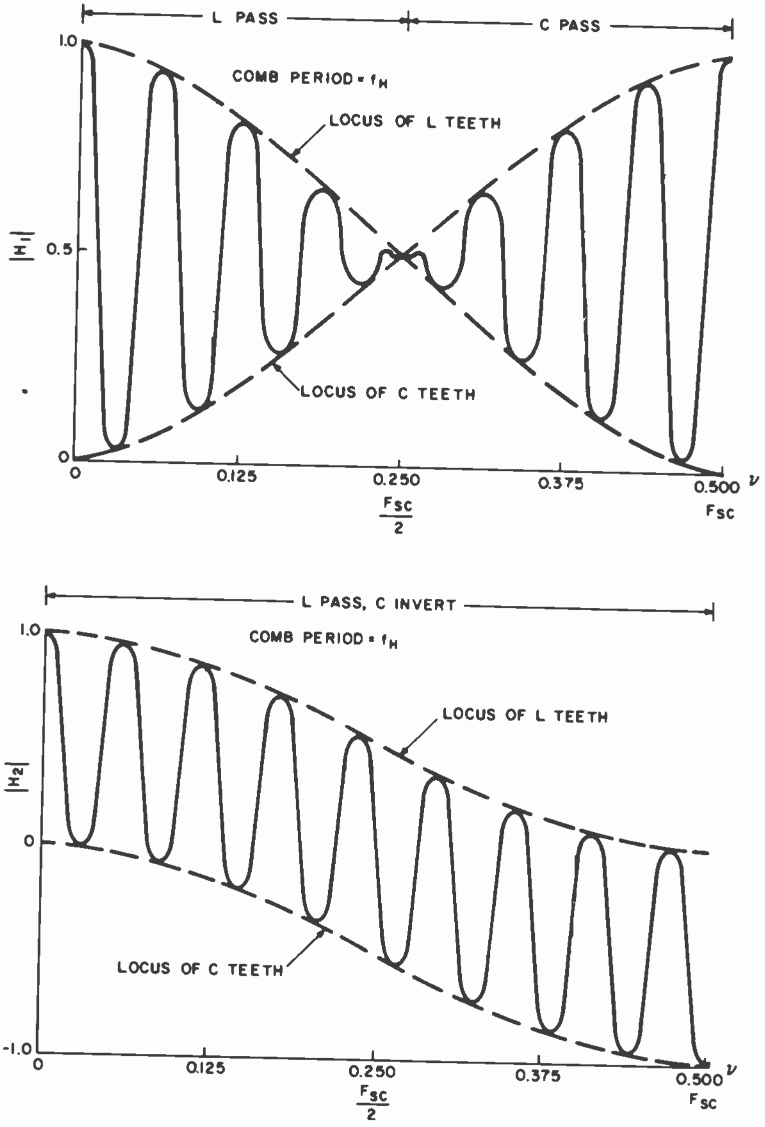


Fig. 13—Filter frequency response (m = 910) (a) for chroma-passing filter and (b) for chroma-inverting filter.

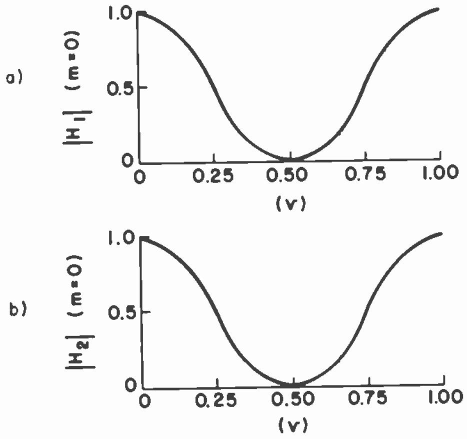


Fig. 14—Horizontal sequence filter responses.

transfer functions $H(Z)$ are specified:

$$H_1(Z) = \frac{Z^{m+2}}{8} + \frac{Z^{m-2}}{8} + \frac{Z^0}{2} + \frac{Z^{-m+2}}{8} + \frac{Z^{-m-2}}{8},$$

$$H_2(Z) = \frac{Z^m}{4} + \frac{Z^2}{4} + \frac{Z^{-2}}{4} + \frac{Z^{-m}}{4}.$$

For reasons that will be evident later, we will denote the transfer functions $H_1(Z)$ and $H_2(Z)$ as the chroma-passing (CP) and chroma-inverting (CI) filters, respectively. Figs. 13a and 13b show the frequency response of these filters for the actual case of horizontal line length, $m = 910$, corresponding to the four times color subcarrier sampling rate. Fig. 14 shows the frequency responses for the case of horizontal line length $m = 0$, thus eliminating all vertical resolution. The reason for considering this case will also soon become evident.

We have just considered filters operating on sequences that progress from left to right and top to bottom. However, the same weighting procedure could also be considered as operating on a sequence progressing first top to bottom and then right to left. Denoting the vertically oriented sequences as primed variables, and l as the vertical line length, the time domain operations can be written

$$y'_1(n) = \frac{x'(n)}{2} + \frac{x'(n-2l-1)}{8} + \frac{x'(n-2l+1)}{8} + \frac{x'(n+2l-1)}{8} + \frac{x'(n+2l+1)}{8},$$

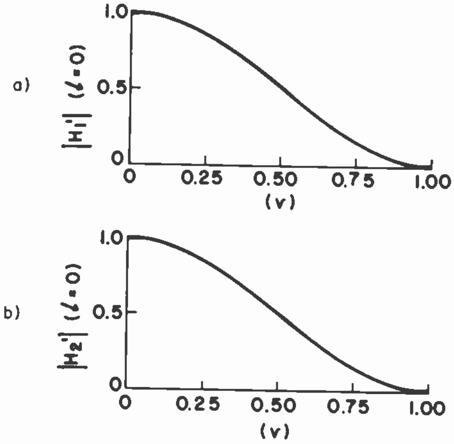


Fig. 15—Vertical sequence filter responses.

$$y_2'(n) = \frac{x'(n-2l)}{4} + \frac{x'(n-1)}{4} + \frac{x'(n+1)}{4} + \frac{x'(n+2l)}{4}.$$

The digital transfer functions are, therefore,

$$H_1'(Z) = \frac{Z^{2l+1}}{8} + \frac{Z^{2l-1}}{8} + \frac{Z^0}{2} + \frac{Z^{-2l+1}}{8} + \frac{Z^{-2l-1}}{8},$$

$$H_2'(Z) = \frac{Z^{2l}}{4} + \frac{Z^1}{4} + \frac{Z^{-1}}{4} + \frac{Z^{-2l}}{4}.$$

Fig. 15 shows the frequency response of $H_1'(Z)$ and $H_2'(Z)$ for the case of $l = 0$, that is, eliminating all horizontal resolution.

So far, the averaging processes have been viewed as one-dimensional problems, and examined from both horizontal and vertical orientations. However, since the averaging is actually done with spatially related picture elements, let us examine these processes as two-dimensional FIR filters. Letting Z_1 represent horizontal delay and Z_2 vertical delay, the two dimensional filters are

$$H_1(Z_1, Z_2) = \frac{Z_1^{-2}Z_2^{-1}}{8} + \frac{Z_1^2 Z_2^{-1}}{8} + \frac{Z_1^0 Z_2^0}{2} + \frac{Z_1^{-2} Z_2^1}{8} + \frac{Z_1^2 Z_2^1}{8},$$

$$H_2(Z_1, Z_2) = \frac{Z_1^0 Z_2^{-1}}{4} + \frac{Z_1^{-2} Z_2^0}{4} + \frac{Z_1^2 Z_2^0}{4} + \frac{Z_1^0 Z_2^1}{4}.$$

The amplitude responses for these filters are shown in Figs. 16a and 16b.

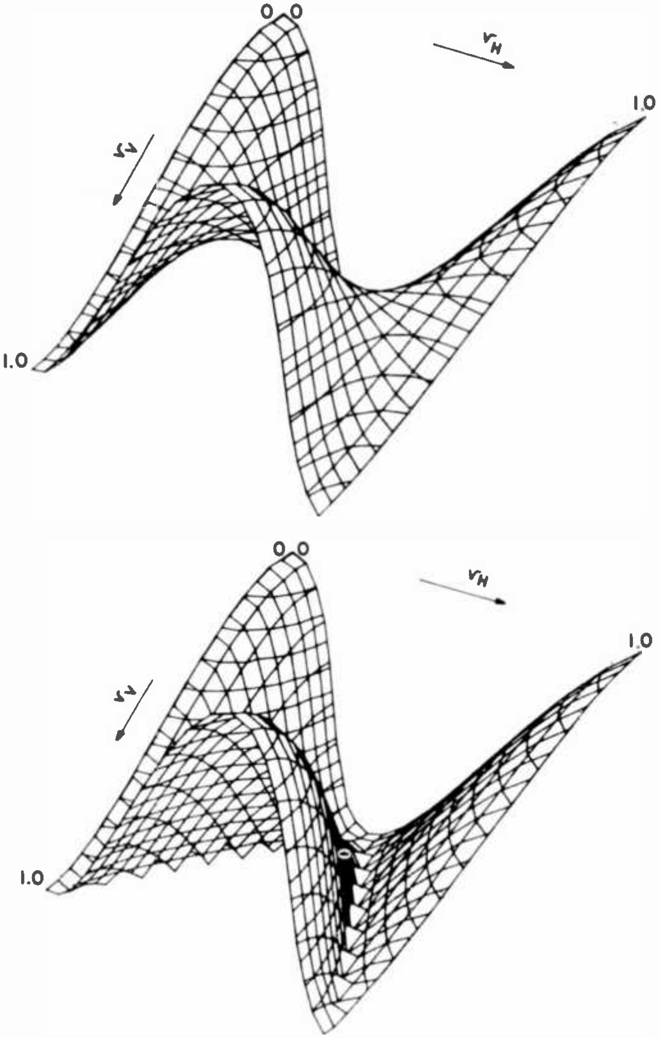


Fig. 16—Two-dimensional frequency response for (a) chroma-passing filter and (b) chroma-inverting filter.

There are several important observations connecting all of the filters examined. In the one dimensional cases, we see that for $m = 0$,

$$H_1(Z) = H_2(Z),$$

and, similarly, for $l = 0$,

$$H_1'(Z) = H_2'(Z).$$

Note the striking.similarity in the responses of $H_1(Z_1, Z_2)$ and $H_2(Z_1, Z_2)$. Closer investigation shows that the responses of the filters are identical along each axis. Furthermore, those responses are identical to the above one dimensional cases. That is, for $m = 0$,

$$H_1(Z_1, 0) = H_2(Z_1, 0) = H_1(Z_1) = H_2(Z_1),$$

and for $l = 0$,

$$H_1(0, Z_2) = H_2(0, Z_2) = H_1'(Z_2) = H_2'(Z_2).$$

Thus, the response of the two-dimensional filters along the horizontal frequency axis is the same as the response of the horizontally oriented one-dimensional filters when all vertical resolution is removed, leaving only terms representative of horizontal delay. A similar case occurs along the vertical frequency axis of the two-dimensional filters and the vertically oriented one-dimensional filters. This serves to illustrate that the one-dimensional filters for any m or l, respectively, are the appropriate projections of the two dimensional case.

At first glance, the two-dimensional frequency responses certainly do not appear to be low-pass filters. Recall, however, that the horizontal information is in an NTSC format. Therefore, we can assume that the relevant luminance spectrum (and, consequently, horizontal spatial frequencies) is bandlimited to the color subcarrier frequency. This corresponds to $\nu_H = 0.5$ in Fig. 16, and we see that both filters have a low-pass structure in the frequency band of interest, satisfying our requirement to eliminate aliasing in the reduced image. Furthermore, it seems that in processing the NTSC signal, we want to apply a filter that has symmetric responses in the luminance and chrominance bands (as the actual filters indeed do, see Fig. 13). The shape of the two-dimensional filters also fills this requirement. Since we have shown that the two-dimensional forms of the FIR filters satisfy our general requirements for compression, let us examine the mechanism of the 1/4-size compression algorithm.

To understand the compression mechanism, it is necessary to take a closer look at the FIR filters that actually operate on the NTSC signal (horizontally oriented, $m = 910$, Fig. 13) and note several important characteristics and relationships. Recall that we refer to H_1 as the CP filter and H_2 as the CI filter:

(1) both filters are comb filters, of period equal to the horizontal line frequency, f_H;

(2) the locus of comb peaks occurring at luminance frequencies is identical in both the CP and CI filters, rolling off from low frequencies;

(3) the locus of comb peaks occurring at chrominance frequencies has

the same shape in both filters; however, the CI filter inverts all of the chrominance frequencies. Both comb envelopes roll off from the subcarrier ($\nu = 0.5$).

Thus, while both filters have identical luminance response, which satisfies the requirement for a two-dimensional low-pass filter, one is a chroma-passing (CP) and the other a chroma-inverting (CI) filter. With these observations, let us now examine the compression algorithm applied to three cases.

Since the luminance response of both CP and CI filters is identical, the compression algorithm behaves in the following manner. Referring to Fig. 17, the first two samples are filtered, and passed with a certain amplitude. The next two are deleted, and the following two are passed with the same amplitude and placed next to the original two samples. The next pair is deleted and the process continued. Because of the identical luminance response of the CP and CI filters, the effect of deleting alternate pairs of samples is to halve the period of the signal. Thus, for any single luminance frequency f_L, we can express the processed signal as

$$T'_L = \frac{T_L}{2} \text{ and } f'_L = 2f_L.$$

Note that there will be some distortion introduced by this process, and that the algorithm will degrade as the frequencies increase, since there will be a greater difference between adjacent samples, and the effect of

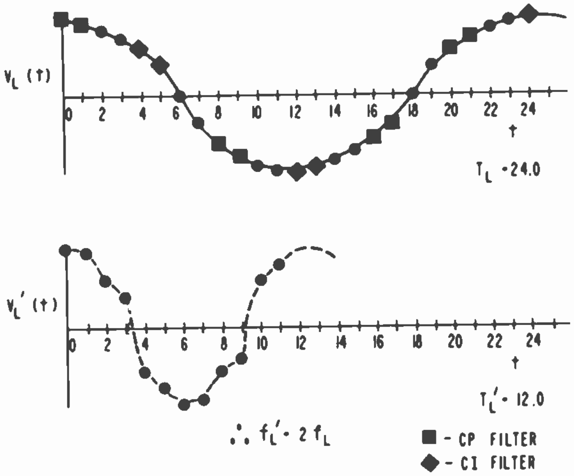


Fig. 17—Effect of the subcarrier reconstruction compression algorithm on a typical lumance component.

dropping two will be to cause a "jump" in the sampled signal. However, $f_L' = 2f_L$ will still be the dominant frequency. We must also observe that at higher frequencies, where distortion will be objectionable, the power content for average scenes is not large to begin with, and the luminance response of the filters will greatly reduce the signal, so that the overall contribution of distortion should not be objectionable.

In the case of the color subcarrier, there are four samples in each cycle (by definition of sampling rate), each pair defining half of cycle. Thus, adjacent pairs of samples exhibit half-wave symmetry. Both CP and CI filters pass the subcarrier frequency ($\nu = 0.5$) with an amplitude of 1, the CI filter inverting, however. We see from Fig. 18 that the algorithm operates in the following manner. Two samples are CP filtered, exactly reproducing their subcarrier phase. Two samples are deleted, so that the next two represent the same portion of a subcarrier cycle as the first two. When these are CI filtered, they are inverted, and because of the half-wave symmetry, "reconstruct" the entire subcarrier cycle. Finally, two samples are deleted, and we repeat the entire process. Thus, the algorithm reconstructs one subcarrier cycle from two cycles of the original signal. Thus, the processed signal f_{sc}' is

$$f_{sc}' = f_{sc}.$$

That this algorithm works at four times subcarrier sampling is of particular importance, because the half-wave symmetry of the subcarrier

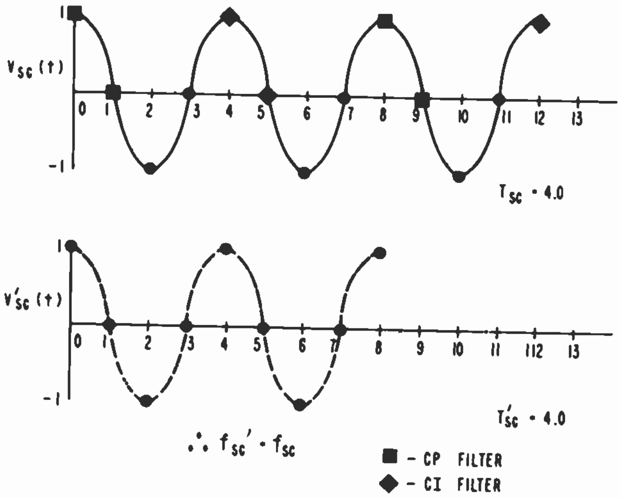


Fig. 18—Effect of the subcarrier reconstruction algorithm on the color subcarrier.

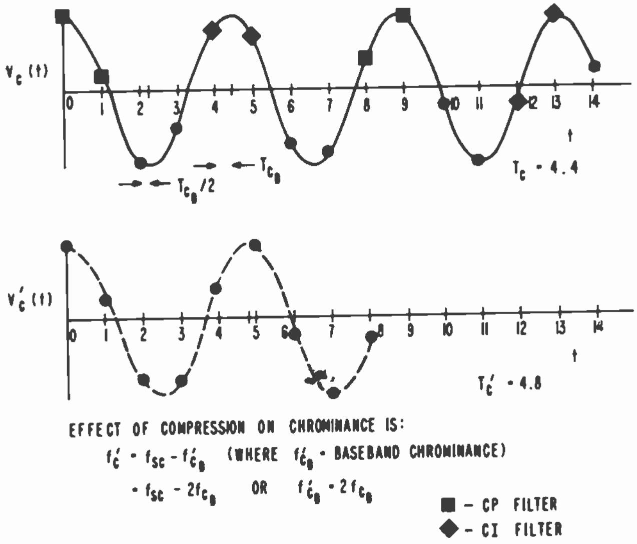


Fig. 19—Effect of the subcarrier reconstruction compression algorithm on a typical chrominance component.

samples is necessary for the proper subcarrier reconstruction and, thus, maintenance of the NTSC format.

The effect of the compression algorithm on the chrominance components is much less obvious than in the preceding cases. Referring to Fig. 19, the filters act as in the case of the subcarrier, except that at chrominance frequencies, half-wave symmetry does not exist. To summarize the process, two samples are CP filtered and passed with a certain amplitude, two deleted, two inverted by the CI filter, and two deleted. Let us express the period of the original chrominance frequency in terms of the subcarrier period and a "differential" period. That is,

$$T_C = T_{sc} + T_D.$$

As seen in the case of Fig. 19, the first-order effect of the compression algorithm is to double the differential period, so that

$$T'_C \doteq T_{sc} + 2T_D.$$

While the effect is not completely obvious, it can be best thought of in the following manner. Since two samples represent some frequency that differs from the subcarrier, the algorithm concatenates pairs of samples that represent different portions of the original cycles. As pairs of samples are deleted, there is a change in the portion of the cycle represented, the change being related to the difference between the subcarrier period and the chrominance signal period, i.e., T_D. The approximate effect of this process will be to create a signal where the differential period is doubled, as indicated above.

Now let us investigate the effect of this process and see if it meets our requirements for image compression. Let us define:

$$f_{sc} = \frac{1}{T_{sc}} \text{ and } f_D = \frac{1}{T_D}.$$

We can now write expressions for the original and processed chroma signals:

$$f_c = \frac{1}{T_{sc} + T_D},$$

$$f_c' \doteq \frac{1}{T_{sc} + 2T_D}.$$

Rewriting in terms of our definitions,

$$f_c = \frac{f_{sc}}{1 + \frac{f_{sc}}{f_D}},$$

$$f_c' = \frac{f_{sc}}{1 + \frac{2f_{sc}}{f_D}}.$$

The baseband versions of these signals are

$$f_{cb} = f_{sc} - f_c$$

$$f_{cb}' = f_{sc} - f_c'$$

or,

$$f_{cb} = f_{sc} - \frac{f_{sc}}{1 + \frac{f_{sc}}{f_D}},$$

$$f_{cb}' = f_{sc} - \frac{f_{sc}}{1 + \frac{2f_{sc}}{f_D}},$$

Now we ask how the original and processed baseband chrominance signals are related. If we write

$$x = \frac{f_{cb}'}{f_{cb}}$$

$$= \frac{f_{sc} - \dfrac{f_{sc}}{1 + (2f_{sc}/f_D)}}{f_{sc} - \dfrac{f_{sc}}{1 + (f_{sc}/f_D)}},$$

we find, upon simplification,

$$x = \frac{2\left[1 + \dfrac{f_{sc}}{f_D}\right]}{\left[1 + \dfrac{2f_{sc}}{f_D}\right]}.$$

Now we see that if the period of the chroma signal is close to the subcarrier period, then the differential period is small. Then, if T_D is small and $f_D \gg f_{sc}$,

$$x \doteq 2$$

$$f'_{cb} \doteq 2f_{cb}.$$

In the case of the subcarrier frequency itself, T_D is 0 and f_D infinite, so from our formulation of the baseband chrominance signals,

$$f'_{cb} = f_{cb} = 0,$$

and the color subcarrier is reproduced exactly, as has already been shown. Thus, the effect of the algorithm is exactly as desired; it will double the baseband chrominance frequencies and still maintain the NTSC format. Note that, similar to the case of luminance components, the algorithm has increasingly nonlinear effects as the signal gets farther away from the subcarrier and will introduce a large amount of distortion. However, also like the luminance case, decreasing the power content of the original signal and the rolloff of the filter chrominance responses should combine to make the overall contribution of distortion not too objectionable.

There are two additional points to address in order to completely cover the compression mechanism. Recall that the compression algorithm called for deleting alternating scan lines and changing the beginning of the alternating CP and CI filtering process. In the original picture, the color subcarrier is 180° out of phase on adjacent lines. By deleting scan lines, adjacent lines would now normally have the same phase, but reversing the CP and CI filter sequence maintains the correct format in the reduced-size picture. Also, the effect of deleting scan lines in conjunction with the low-pass vertical characteristic of the filters provides a picture with increased vertical spatial frequencies without objectionable aliasing.

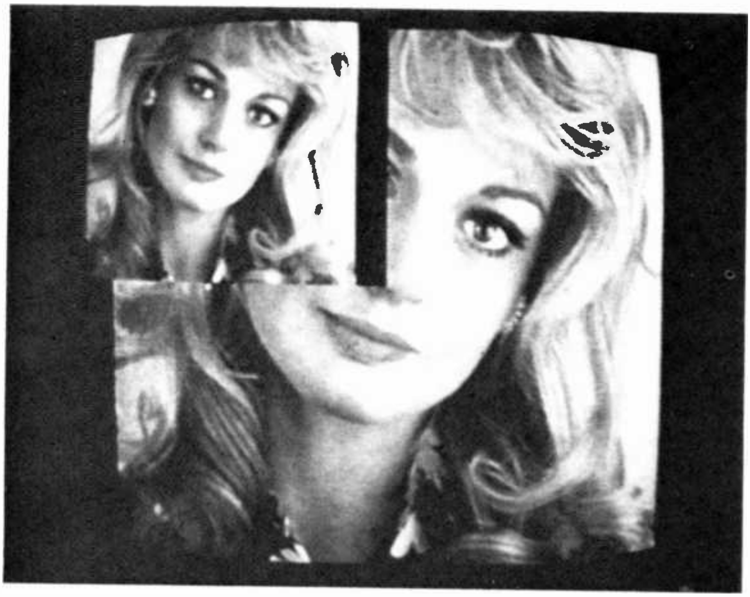


Fig. 20—One-quarter-size image produced by the subcarrier reconstruction method.

To summarize, we have shown how the ¼-size compression algorithm implements the requirements for image compression. Again, because conventional analytic methods do not adequately describe the nonlinear concatenation of samples and the analysis presented is somewhat intuitive, this algorithm has also been implemented in software, using a Fast Fourier transform, to verify the effects of the algorithm. The results, when compared with those of the previous algorithm, show much less distortion in the compressed waveform. For the ultimate comparison, the same test slide and monoscope have been compressed using this algorithm, and are shown in Figs. 20 and 21, respectively. Obviously, this algorithm is a great advance over the video-data-deletion method. It produces excellent results on the slide, but still leaves much room for improvement on the monoscope. While this method could conceivably be extended to a variable ratio algorithm by deriving other CP and CI type filters with varying degrees of two dimensional low-pass filtering, the compressed image is not of sufficient quality, and other methods must be investigated.

5. Compression by Direct Subsampling and Expansion by Direct Repetition

Thus far, we have only considered algorithms that utilize the specific

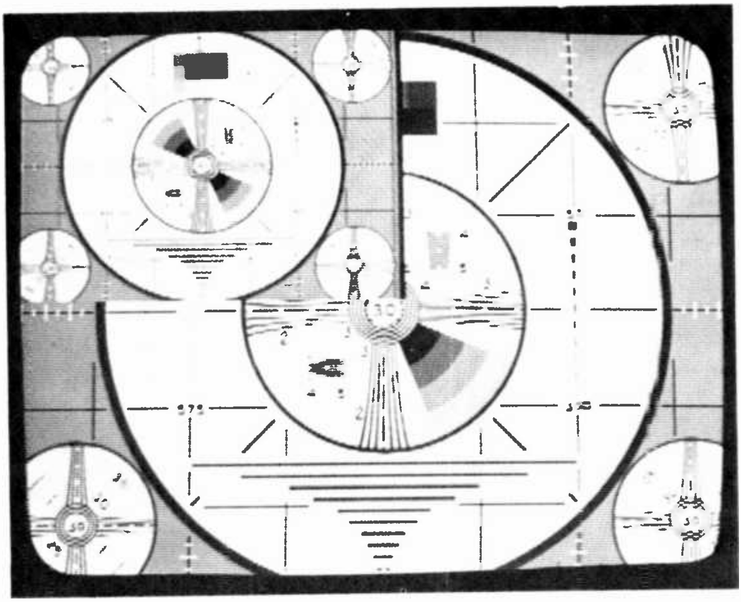


Fig. 21—One-quarter-size monoscope produced by the subcarrier reconstruction method.

relationships of the composite NTSC signal to produce a ¼-size picture. To achieve the goal of a variable compression ratio, it becomes necessary to separate the signal into a set of its component signals (preferably Y, I, and Q), process these in parallel, and recombine them into a compressed NTSC color image at output. Henceforth, only the luminance component will be processed for evaluation, since it has the highest bandwidth of the component signals, and therefore provides the most critical assessment of image quality.

Let us once again consider the approach to the problem where we simply delete an appropriate number of pixels horizontally and entire lines vertically. Now, from the previous consideration of compression requirements, it is obvious that if we want to spatially compress a picture to S times its original size (where $S \leq 1$), it must first be low-pass filtered with cutoff $v_c = S$ before any subsampling operation can be performed, in order to eliminate aliasing. Since we have only a component signal to operate on, and no constraints regarding the color subcarrier, the filtering can be performed in a straightforward manner. To achieve independent compression in each dimension, the two-dimensional filter characteristics must be independently alterable. Utilizing a separable filter results in a simpler structure for implementation, as well as reducing our filtering concepts to that of using one-dimensional filters independently in each dimension. For the horizontal filter, we simply implement a standard LPF. Applying the same filter coefficients to a vertical sequence of data implements a vertical LPF, which is translated to the proper comb structure in the time domain to properly filter the video signal, as has been previously discussed. We have achieved one further advantage, that is, we may now maintain and delete picture data on a pixel by pixel basis.

Examining the case of producing a ¼-size image, it is possible to produce excellent results using this approach. Since we may now delete every other pixel horizontally and every other line vertically, we can achieve a linear subsampling with the proper precompression filtering. Considering the compression process in one dimension, a sampled signal with resolution T is shown in Fig. 22a, after prefiltering at $\nu = 0.5$, which is denoted as $f(t)$. Fig. 22b shows the signal with every other sample deleted. Since the signal has been prefiltered, it is adequately represented by this subset of the original samples. Now the remaining samples are concatenated so that their resolution is again T, which is equivalent to sampling $f(2t)$ at this resolution, as seen in Fig. 22c. Therefore, we can perfectly represent a signal compressed by a factor of two by this combination of prefiltering and subsampling. In fact, this method will work

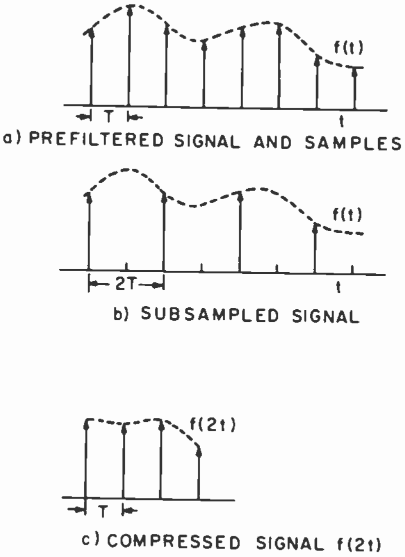


Fig. 22—One-half-size compression by subsampling.

perfectly for image sizes that involve keeping one and deleting an integer number of pixels or lines, corresponding to one-dimensional compressions of 1, ½, ⅓, ¼, and ⅕, etc.

Fig. 23 shows a ¼-size monoscope image that has been compressed using linear subsampling in conjunction with digital Butterworth prefilters. Comparing this result with that of previous algorithms, the image quality has been dramatically improved, since the problems of aliasing and nonlinear subsampling have been corrected. Now that it is possible to implement an ideal compression process, the effects of non-ideal prefiltering must be considered, and this will be done in the next section.

Now, let us examine how the goal of variable ratio compression might be achieved. Since the compressed image must consist of an integer number of pixels, the compression to a size not attainable by direct subsampling can be approximated by a sequence of subsamplings, which, on the average, provide the proper ratio between pixels that are maintained and deleted. For example, a ratio of 3/5 can be achieved by the sequence of keeping one pixel, deleting one pixel, keeping two pixels, deleting one pixel. This process is illustrated in Fig. 24. Similarly, a ratio

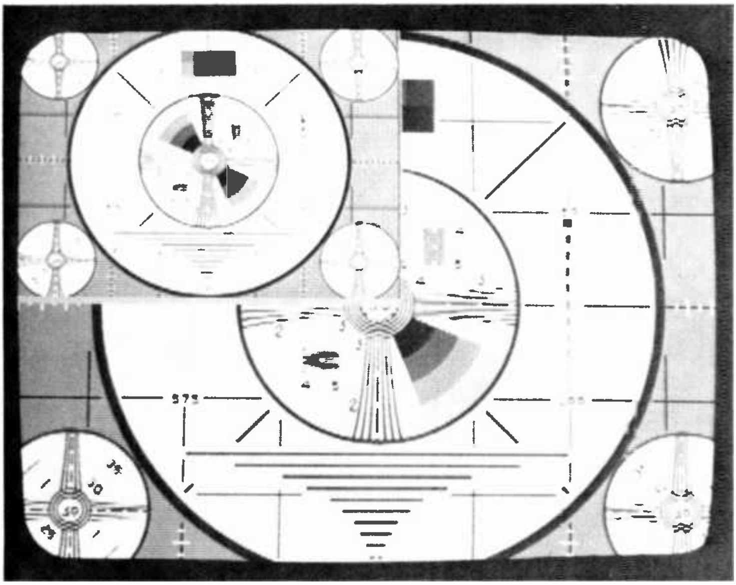
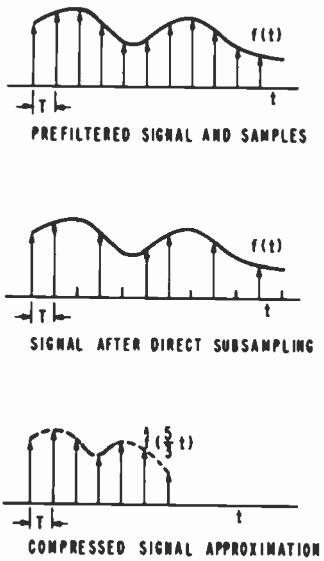


Fig. 23—One-quarter-size monoscope produced by one-half subsampling in each dimension.



PREFILTERED SIGNAL AND SAMPLES

SIGNAL AFTER DIRECT SUBSAMPLING

COMPRESSED SIGNAL APPROXIMATION

Fig. 24—Three-fifth-size compression by direct subsampling.

of 5/7 can be achieved by the sequence of keeping three pixels, deleting one pixel, keeping two pixels, deleting one pixel; or a ratio of 2/7 may be obtained by a sequence of keeping one pixel, deleting three pixels, keeping one pixel, deleting two pixels. This process can be implemented by modifying the number of samples to be maintained and deleted after each subsampling operation. Thus, a linear subsampling is the basis for this compression technique, with changes made occasionally to correct the compression while it is in progress. The following scheme implements this idea. The total number of pixels to be kept in each line is given by

$$T_K = SN,$$

where $S = 1/k$ is the desired size of the compressed signal and N is the number of samples in the original. Then the total number of pixels to be deleted is

$$T_D = N - T_K.$$

Now, the average ratio of pixel deletions is

$$R = \frac{N}{T_D},$$

and the ratio of deletions for the remaining portion of the image (that as yet unprocessed) is

$$R_r = \frac{N_K + N_D}{N_D},$$

where N_K is the remaining number of pixels to be kept and N_D the remaining number to be deleted. Initially, of course, $N_K = T_K$ and $N_D = T_D$. For $S \leq 1/2$, apply the following steps repetitively:

(1) compute the deletion ratio of the remaining portion, R_r;
(2) keep one pixel;
(3) delete int(k) − 1 pixels, when $R_r \geq R$; delete int(k) pixels, when $R_r < R$ (where int(x) denotes an integer truncation);
(4) appropriately decrement the number of pixels remaining to be kept and deleted, N_K and N_D.

When $S > 1/2$,

(1) compute the deletion ratio of the remaining pixels, R_r;
(2) keep int$(k - 1)^{-1}$ pixels, when $R_r \leq R$; keep int$(k - 1)^{-1} + 1$ pixels, when $R_r > R$;
(3) delete one pixel;
(4) appropriately decrement the number of pixels remaining to be kept and deleted, N_K and N_D.

Of course, the same procedure is applied vertically to entire scan lines to achieve variable compression independently in two dimensions.

By considering the operation of the above algorithm, and observing its effect, as shown in Fig. 24, we see that the resulting function is indeed an approximation to the ideally compressed signal, and that the "jumps" caused by changing the number of pixels maintained and deleted is analagous to the effect of sampling jitter. Clearly, the closer a desired size is to one of the ideal subsampling cases, the better the above algorithm performs, since the corrections occur infrequently. Of course, at those cases where linear subsampling produces the desired result, the above algorithm does indeed produce that result. While we still have not yet developed a perfect compression algorithm, this one does allow variable ratio compression, is easily implementable, and its worst effect is that of sampling jitter. For compression to an arbitrary size, it certainly will produce a compressed image of higher quality than potential composite signal algorithms, such as a subcarrier reconstruction method, which must maintain multiple samples. As an example of the worst case effects of this method, Fig. 25 shows the monoscope image, which has been compressed to a size of 0.82 in both the horizontal and vertical dimensions. While the sampling jitter effect is certainly noticeable, it is not as objectionable as one might first expect.

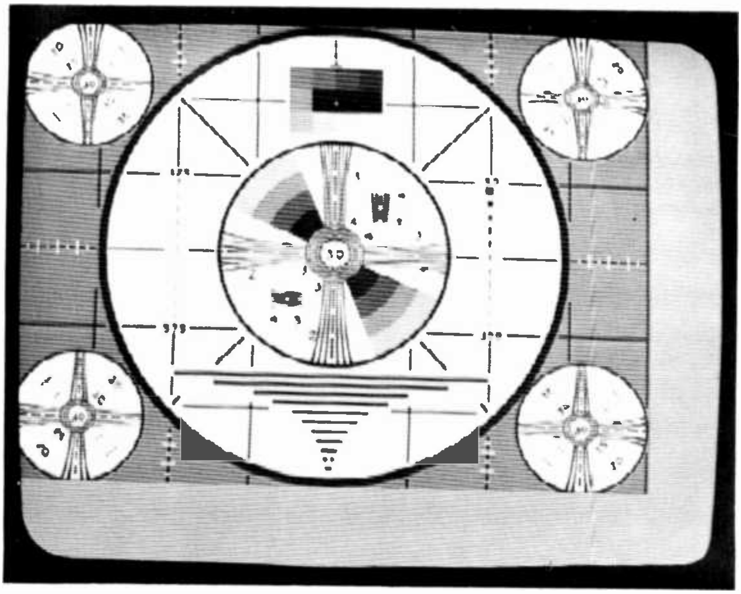


Fig. 25—0.82 × 0.82 compressed monoscope produced by the direct subsampling method.

Analagously to the procedure for obtaining variable ratio image compression by direct subsampling of pixels, let us consider the possibilities of expanding an image by the direct repetition of pixels. Once again examining the problem in one dimension, an expansion ratio of 5/3 can be achieved by the sequence of keeping two original pixels, repeating the last pixel once, keeping the next original pixel, and repeating it once, as shown in Fig. 26. Similarly, the ratio of 7/5 can be achieved by keeping three pixels, repeating one, keeping two pixels, repeating one; or a ratio of 7/2 achieved by keeping one pixel, repeating it three times, keeping another pixel, and repeating it two times. For expansion in the vertical direction, the same procedure is performed with entire lines. Obviously, an algorithm for variable ratio expansions can be developed in a similar manner as was done in the case of compression. However, the case of expansion should provide a much more critical test of this approach than compression did, where the original data was smoothed by prefiltering. The visual effect of pixel repetitions will be most objectionable where a fixed pattern is present, and Fig. 27 shows a 1.33 size monoscope, produced by the sequence of keeping three pixels and repeating one. Clearly, the jitter resulting from this method is more ob-

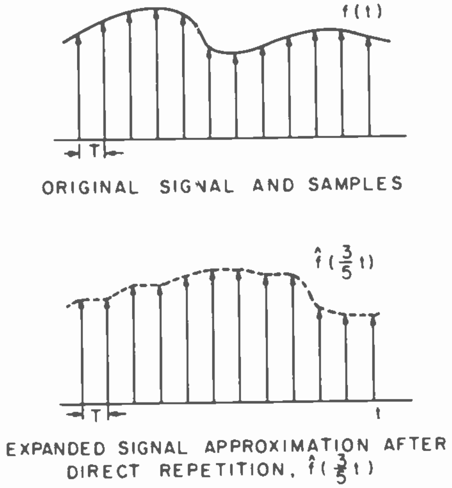

f(t)
T
ORIGINAL SIGNAL AND SAMPLES
$\hat{f}(\frac{3}{5}t)$
T
t
EXPANDED SIGNAL APPROXIMATION AFTER DIRECT REPETITION, $\hat{f}(\frac{3}{5}t)$

Fig. 26—Five-third-size expansion by direct repetition

jectionable than in the case of compression, and produces an image of unacceptable quality.

While the overall results of these approaches to image compression and expansion have not been satisfactory, they are the first step in at-

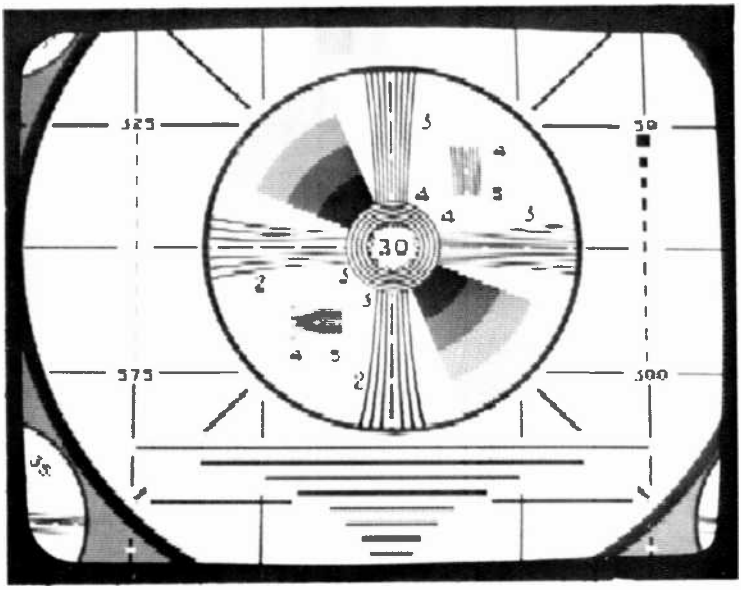


Fig. 27—1.33 × 1.33 expanded monoscope produced by the direct repetition method.

tempting to produce images of variable size. Further, the case of compression has yielded perfect results for picture sizes of $1/k$, where k is an integer, since a linear subsampling results in these sizes.

6. Digital Prefilter Selection and Tradeoffs

Now that ideally compressed images of certain sizes can be produced by linear subsampling, the effects of nonideal prefiltering on resulting image quality will be considered. To simplify the observation of the effects of various prefilters, the case of ¼-size compression in the horizontal dimension will be examined, which of course is achieved by keeping one out of every four pixels. Therefore, a digital LPF with cutoff frequency $\nu_c = 0.25$ is required to prefilter the active video information prior to subsampling and compression. However, the process of compression magnifies and visually illustrates the effects of nonideal filtering, and thus makes the filter selection a critical decision. In the process of

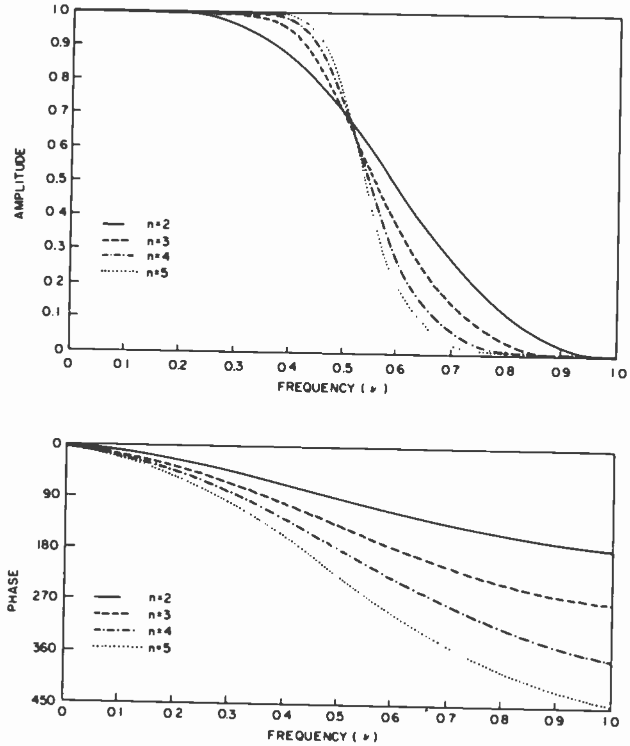


Fig. 28—Butterworth filter response.

compression, signal energy that exists above $\nu = 0.25$ after prefiltering results in aliasing, which is extremely objectionable to the viewer. Therefore, the transition bandwidth and stopband attenuation of a filter become even more critical in this application than in others, and result in interesting tradeoffs with the effects of 3-dB bandwidth and sharpness of cutoff.

In many applications, including broadcast video, frequency selective digital filters are designed by utilizing an analog transfer function as a prototype filter, and then affecting an appropriate transformation to obtain the desired digital transfer function.[5] Such filters are successfully designed by applying the bilinear transformation method. For the purpose of this paper, this procedure is used to design the digital Butterworth and Chebyshev low-pass IIR (infinite-duration impulse response) filters which will be applied to video signals.

It is well known that the Butterworth filter possesses a maximally flat amplitude characteristic, and that the Chebyshev filters are equiripple

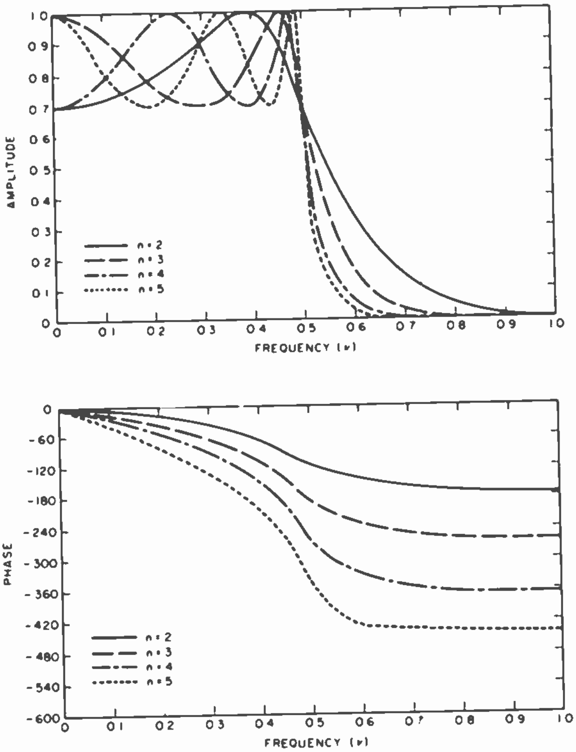


Fig. 29—Chebyshev I filter response.

designs, designated as Chebyshev I and Chebyshev II for ripple present in the passband and stopband, respectively. For easy comparison, both amplitude and phase responses of digital Butterworth, Chebyshev I, and Chebyshev II filters are shown in Figs. 28, 29, and 30, respectively. Looking at the amplitude responses, it is obvious that a sharper amplitude response requires higher-order filters. Unfortunately, the accompanying phase response becomes increasingly nonlinear. The Butterworth phase responses are preferable to those exhibited by the Chebyshev filters, and also, the lower-order filters have more nearly linear phase responses than those of higher order. The digital equivalents of other common analog filters, such as Bessel and elliptic designs, will not be discussed, since it is the Butterworth and Chebyshev designs that make the necessary tradeoffs between amplitude and phase response which are most certainly required for video processing.

It is also well known that digital FIR (finite-duration impulse response) filters can be designed so that their frequency response has an

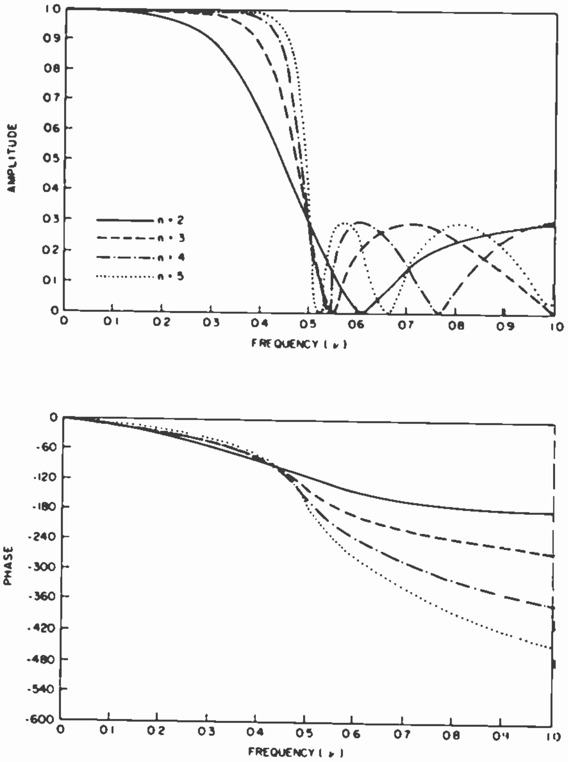


Fig. 30—Chebyshev II filter response.

exactly linear phase characteristic.[6] One technique is to let the filter coefficients be a windowed Fourier series representation of the desired frequency response, and it was this method that was used to design the FIR filter that was implemented for comparison with the IIR filters. However, these FIR filters require a large value of N, the impulse response duration (i.e. filter order), to adequately approximate sharp cutoff filters. Hence, a large amount of processing and storage is required to realize such filters, thus making them undesirable for video applications.

To illustrate aliasing at its worst, Fig. 31 shows an image that has been compressed with no prefiltering. Note the translation of high frequency black and white bars to that of a lower frequency. Further, certain pieces of the image can be seen to have simply disappeared from the picture altogether. These effects, of course, are the visual results of aliasing. Now let us apply a prefilter. We will first select a digital Butterworth design, because of its maximally flat amplitude characteristic and fairly good phase response. The filter is composed of cascaded second-order sections and is designed with the reference frequency as the −3 dB point. For our

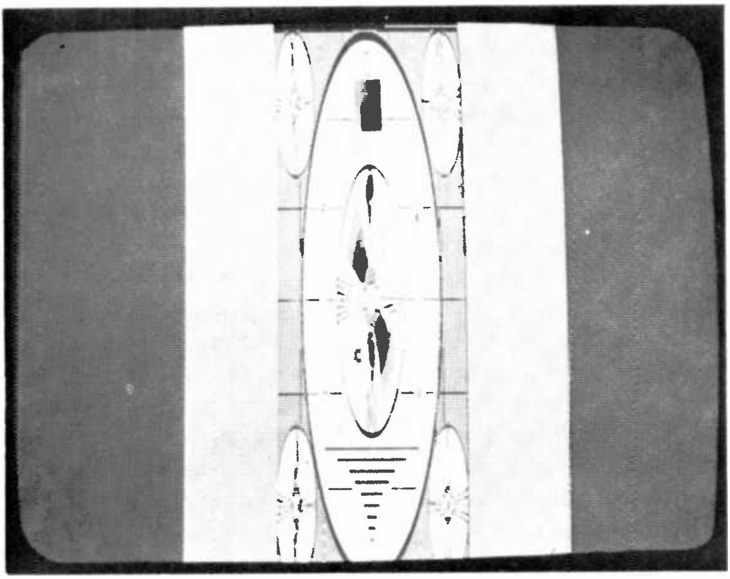


Fig. 31—One-quarter-size horizontally compressed monoscope produced by subsampling with no prefilter. Note the severe effects of aliasing in the image.

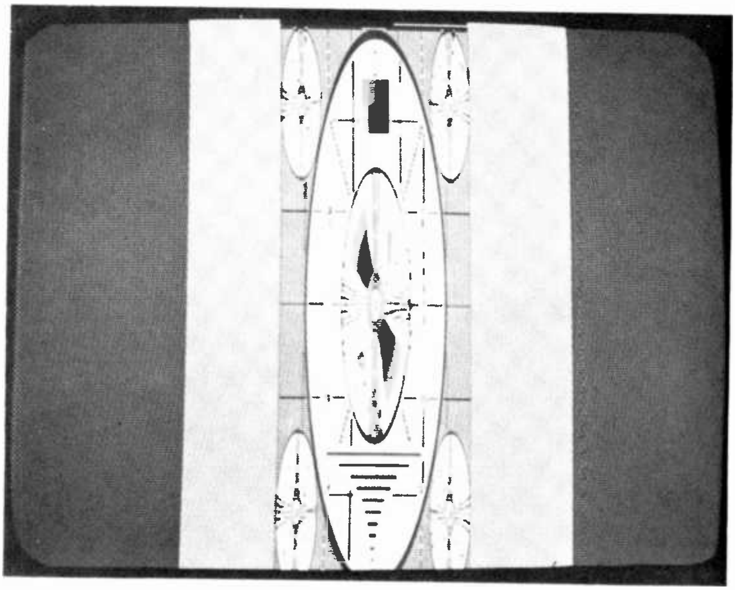


Fig. 32—One-quarter-size horizontally compressed monoscope with sixth-order digital Butterworth prefilter.

first attempt, let us select a sixth-order filter with −3 dB point at ν_c = 0.25. Note in Fig. 32 the odd black rectangles present in the image. Upon close examination, however, there are none of the classic effects of aliasing visible in the picture, nor are any "rectangles" present in the image compressed with no prefilter at all. Before jumping to any conclusion, let us next try a fourth-order Butterworth filter, with ν_c = 0.25. As seen in Fig. 33, our black rectangles are still present in the image, although to a lesser degree. Let us once again reduce the filter order, to a second-order Butterworth with ν_c = 0.25. From Fig. 34, we see that we have now eliminated our black rectangles from the picture. However, a critical examination also reveals indications of aliasing in the picture, particularly in the horizontal and vertical bars. Hence, we have arrived at the tradeoffs to be made in the application of digital filters to the problem of horizontal image compression. A high filter order results in an excellent reduction of aliasing; however, the sharp cutoff of the filter produces ringing in the output waveform and has an associated nonlinear phase response, which together are responsible for the objectionable rectangles introduced into the picture. As the order of the filter decreases, the phase nonlinearity lessens and the overshoots in the output signal are reduced,

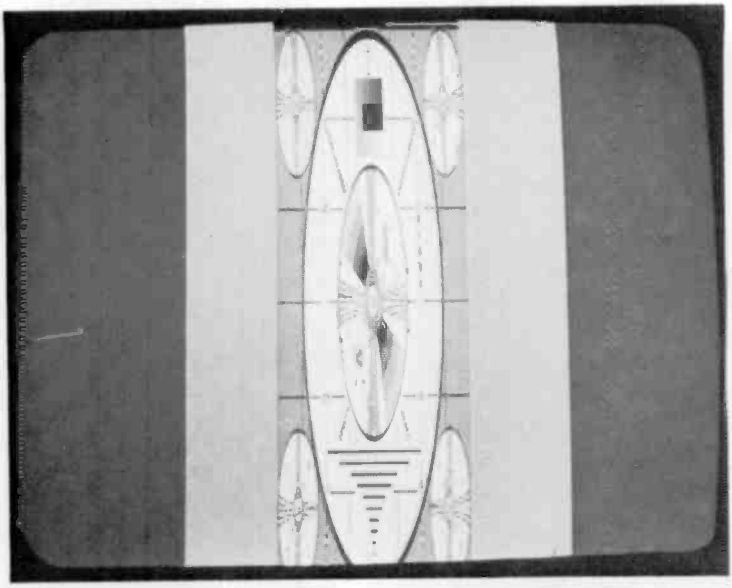


Fig. 33—One-quarter-size horizontally compressed monoscope with fourth-order digital Butterworth prefilter.

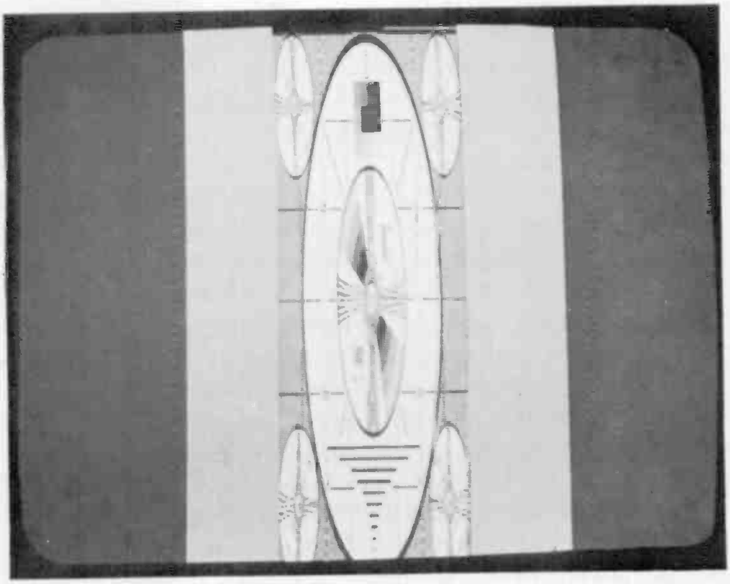


Fig. 34—One-quarter-size horizontally compressed monoscope with second-order digital Butterworth prefilter.

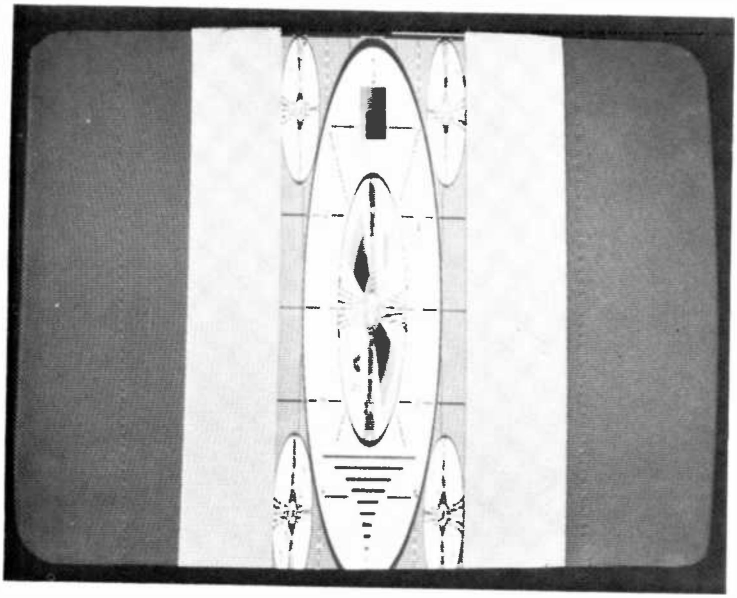


Fig. 35—One-quarter-size horizontally compressed monoscope with third-order digital Chebyshev II prefilter.

but the cutoff sharpness also decreases and admits aliasing that is visible in the picture. Therefore, to obtain an image of optimal visual quality, we must choose a filter to somehow balance the effects of aliasing, ringing, and phase nonlinearity.

The Chebyshev filters are not at all promising for this application, since they exhibit sharp transitions, but with very nonlinear phase response. In fact, all of the even-order Chebyshev I designs are unacceptable, since they introduce an attenuation at low frequencies that would result in a darkening of large picture areas. The Chebyshev II types are clearly not desirable in this application where good stop-band attenuation is necessary to avoid aliasing. Fig. 35 illustrates the effect of a third-order Chebyshev II ($\nu_c = 0.25$) prefilter on a compressed image. A close examination reveals that the undesirable effects of aliasing *and* phase nonlinearity are both present in this case. Clearly, then, a Chebyshev I third-order filter would also be unacceptable due to the phase distortion, thus eliminating any possibility of an acceptable compromise being found in a Chebyshev design.

Let us now examine our alternatives. It seems as if all of our problems could be resolved by the use of any of the higher order IIR filters, cas-

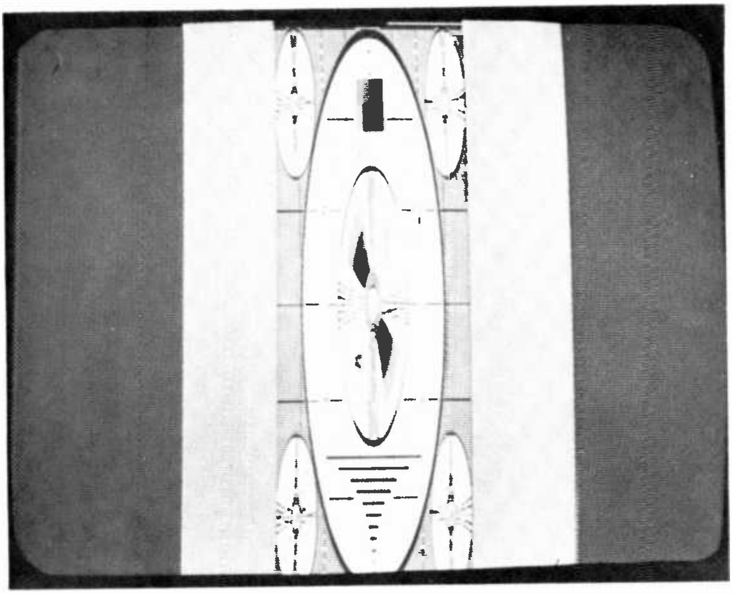


Fig. 36—One-quarter-size horizontally compressed monoscope with scaled second-order digital Butterworth prefilter.

caded with a digital all-pass phase equalization filter. While this is one solution, the use of an initially higher-order filter, in addition to the order of the all-pass network required to correct the phase response to an acceptable characteristic, is undesirable for real-time implementation and should be avoided if we can find a reasonable alternative. The second-order Butterworth design is extremely attractive, except that it does not have a sharp enough rolloff to prevent aliasing energy to be visible when $\nu_c = 0.25$. Therefore, let us scale our cutoff frequency $\nu_c = \beta \cdot 0.25$, so that the aliasing energy is more severely attenuated, at the expense of the 3-dB bandwidth of the image and, correspondingly, its overall sharpness. Subjectively, $\beta = 0.8$ has been found to be a good value; this gives the image a 3-dB bandwidth of $\nu = 0.2$ and has approximately 6-dB attenuation at $\nu = 0.25$. The results of using this as a compression prefilter are shown in Fig. 36, and this seems to be an excellent compromise, since the effects of aliasing are much less noticeable than with the previous second-order filter, while we have not sacrificed much apparent sharpness.

Our final alternative is to utilize a high-order, linear phase FIR filter. For comparison, a nineteenth-order FIR filter (a Hanning windowed Fourier series design) was implemented as a prefilter, with the results shown in Fig. 37. Clearly, this filter produces the best results of all.

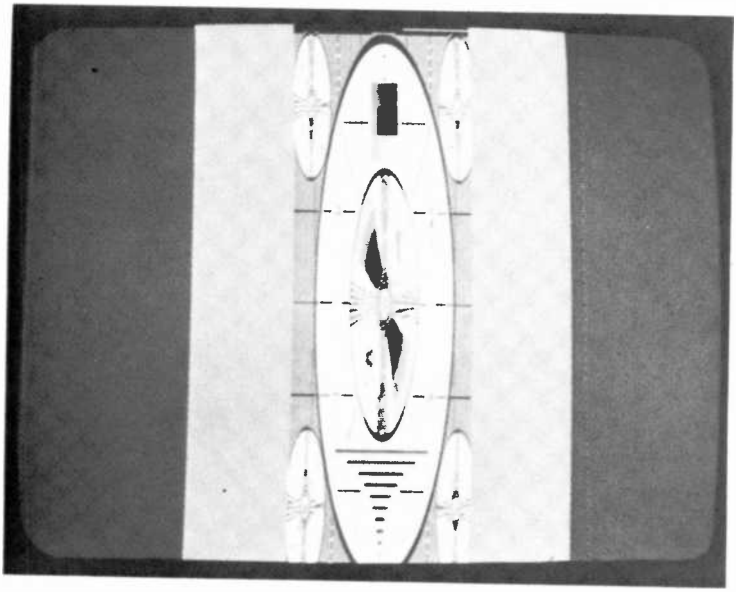


Fig. 37—One-quarter-size horizontally compressed monoscope with nineteenth-order FIR prefilter.

However, the FIR implementation is even worse than the alternative of phase equalization for an IIR filter. Not only is the number of multipliers required for an IIR filter significantly less, but had the compression and filtering been done in the vertical direction, rather than horizontally, instead of 19 samples we would require 19 lines (i.e., 17,290 samples) of storage for the FIR filter, thus making this alternative highly impractical for applications involving the multidimensional processing of video signals.

For the prefilters associated with image compression, we have seen that aliasing energy and phase nonlinearity have a greater impact on subjective image quality than the 3-dB bandwidth. The best compromise seems to be the second-order digital Butterworth filter, with the 3-dB-bandwidth scaled down appropriately for the application. The additional arithmetic operations and storage required for the alternatives of digital phase equalization or FIR filter designs significantly increase the problems of performing digital filtering of video signals in real time, and thus makes it necessary to consider the tradeoffs inherent in the use of low-order IIR filters in order to select the best filter for this application.

7. Compression and Expansion by Resampling Through Interpolation

The algorithm that has been developed, while it does allow variable ratio compression and expansion, is based on simply deleting or repeating appropriate samples and not on a theoretical approach, which will now be developed.

Recall that the previous algorithm did produce theoretically perfect compressions when a linear subsampling operation could be performed. The discrete signal $f_\delta(t)$ can be expressed in terms of the analog signal $f(t)$ and the sampling period T as

$$f_\delta(t) = \sum_{n=-\infty}^{\infty} f(nT)\,\delta(t-nT).$$

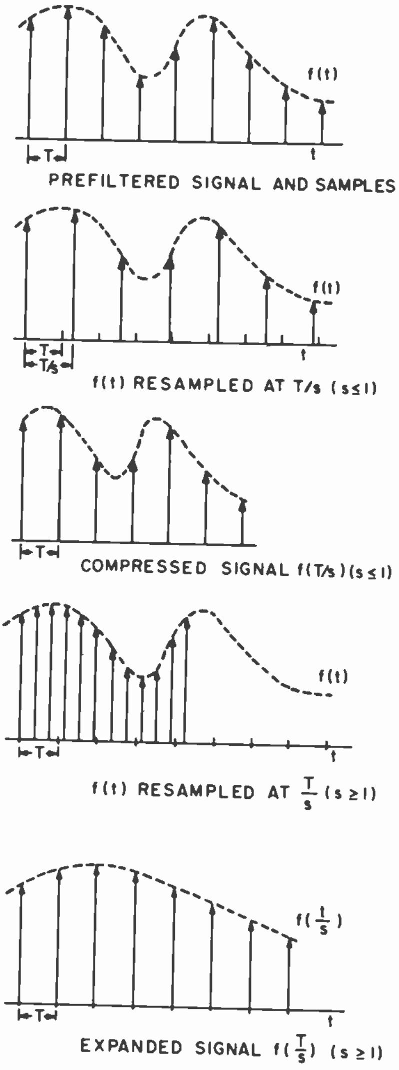
When an integer subsampling occurs corresponding to a picture size of $S = 1/k$, one out of k ($k = 1, 2, 3 \ldots$) samples is maintained, which is equivalent to having originally sampled the signal at a resolution of kT. Hence, the subsampled signal, $g_\delta(t)$, is

$$g_\delta(t) = \sum_{n=-\infty}^{\infty} f(nkT)\,\delta(t-nkT).$$

However, the samples are once again converted to a resolution of T by scaling the output time resolution by a factor of k, so that the output signal, $g_\delta(kt)$, is

$$\begin{aligned} g_\delta(kt) &= \sum_{n=-\infty}^{\infty} f(nkT)\,\delta(kt-nkT) \\ &= \sum_{n=-\infty}^{\infty} f(nkT)\,\delta(t-nT) \end{aligned}$$

The above is, of course, the sampling expansion for the signal $f(kt)$ sampled at a resolution of T, and is the result of the subsampling and discrete time scaling processes. For the case where k is an integer, subsampling produces the appropriate $f(nkT)$. However, the preceding expressions suggest that a perfect compression or expansion of any size can be obtained, as long as the samples of $f(nkT)$ are available, for any k. Thus, to produce the discrete representation of a picture of size S, samples of the original image must be taken with resolution T/S and converted to a resolution of T, as shown in Fig. 38. However, this process must only be performed on the active portion of video waveform after component separation and the appropriate prefiltering. Therefore, it cannot be accomplished via the original sampling, and what is required is a method of interpolating the arbitrarily located samples $f(nT/S)$ from the original samples of $f(nT)$.



PREFILTERED SIGNAL AND SAMPLES

f(t) RESAMPLED AT T/s (s≤1)

COMPRESSED SIGNAL f(T/s)(s≤1)

f(t) RESAMPLED AT $\frac{T}{s}$ (s≥1)

EXPANDED SIGNAL $f(\frac{T}{s})$ (s≥1)

Fig. 38—Perfect compression and expansion by resampling.

Ideally, any signal sampled at or above the Nyquist rate is recoverable by passing the sampled signal through a LPF with cutoff $f_c = f_s/2$. This is the same as convolving the sampled sequence with the filter impulse response, which interpolates between sample points. Therefore, assuming an ideal LPF, $f(t)$ is given by

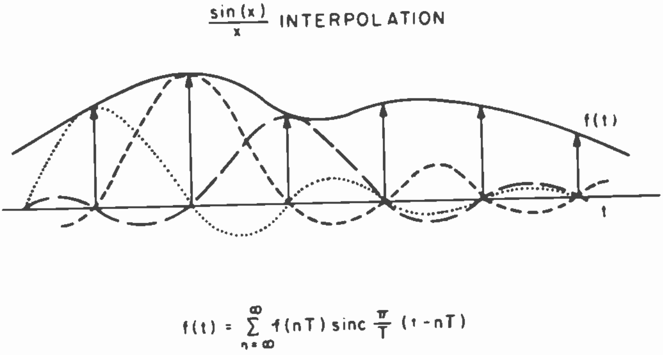

$\frac{\sin(x)}{x}$ INTERPOLATION
f(t)
t
$f(t) = \sum_{n=\infty}^{\infty} f(nT)\,\mathrm{sinc}\,\frac{\pi}{T}(t-nT)$

Fig. 39—sin(x)/x interpolation.

$$f(t) = \sum_{n=-\infty}^{\infty} f(nT)\,\mathrm{sinc}\,\frac{\pi}{T}(t-nT).$$

This interpolation process, illustrated in Fig. 39, gives us a means to calculate the required resampled points $f(nT/S)$ needed to implement an ideal variable-ratio compression.

To summarize this method, then, for a compression to size S the signal

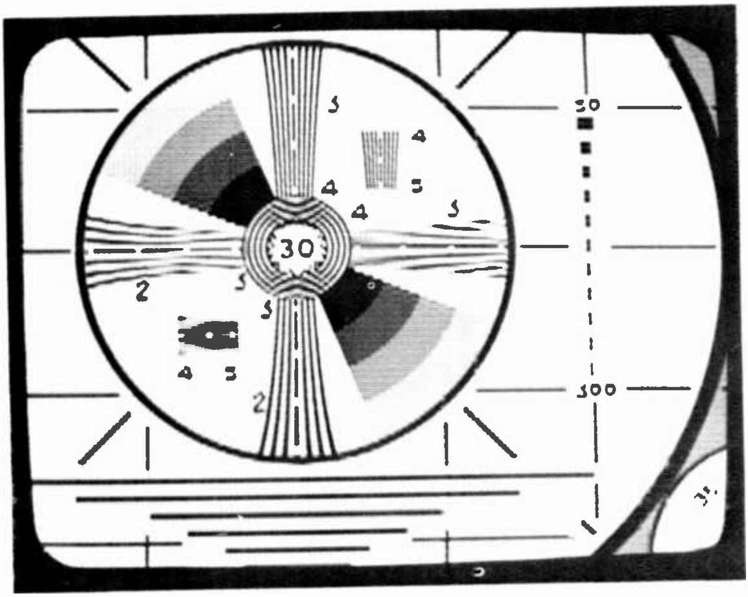


Fig. 40—1.54 × 1.54 expanded monoscope with finite length sin(x)/x interpolation (N = 5).

is prefiltered with cutoff $\nu_c = S$, the interpolated values of $f(nT/S)$ are calculated, and then output with resolution T. In the case of expansion, no prefiltering is required, so that only the interpolation of the samples $f(nT/S)$ and their rescaling to resolution T is necessary. While this method was developed as a one-dimensional problem, it is equally valid to perform the operations independently in the horizontal and vertical dimensions of an image, so that this procedure is performed horizontally by interpolating between adjacent pixels, and vertically by interpolating between lines. When implementing the interpolation, it is, of course, necessary to deal with a finite sum, so that

$$\hat{f}(t) = \sum_{n=-N}^{N} f(nT) \operatorname{sinc} \frac{\pi}{T}(t - nT),$$

where $\hat{f}(t)$ is the approximate value of $f(t)$. Fig. 40 shows a monoscope that has been expanded to 1.54 times its original size in both the horizontal and vertical dimensions, using a finite interpolation with $N = 5$. As can easily be seen, the image suffers a great degree of degradation due to the slow convergence of the sinc function and clearly shows a related ripple structure. At first, the solution would seem to be to take a larger value of N. Fig. 41 shows the effect of doubling N, in a 1.22 times ex-

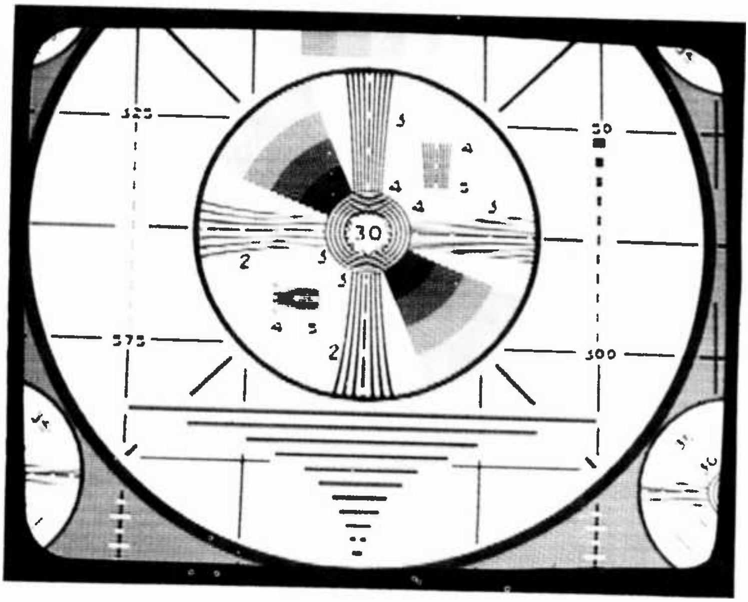
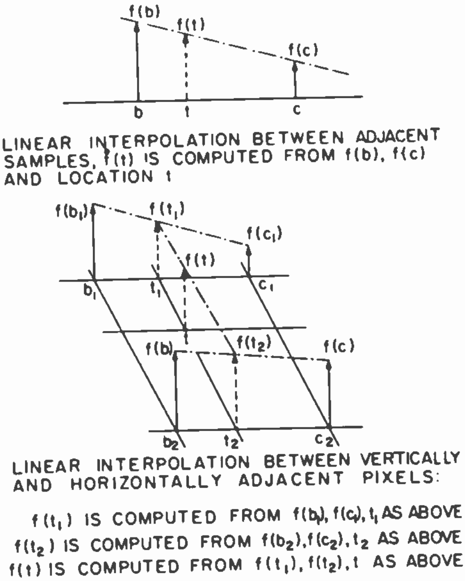


Fig. 41—1.22 × 1.22 expanded monoscope with finite length sin(x)/x interpolation (N = 10).

pansion of both dimensions with $N = 10$. Clearly, the resulting image quality is better, but far from acceptable. A critical problem is that the computational circuitry required for this method to be implemented in real time is hardly conceivable even for small values of N, much less for very large values. While it is possible to derive other interpolators with much faster convergence than the sinc function based on smoother filters of reduced bandwidth, these expressions generally involve rather complex trigonometric calculations, and are simply not suitable for implementation.

Therefore, while linear systems theory allowed a derivation of the ideal method of compression, the interpolation scheme dictated by ideal considerations must be forsaken because of its computational complexity. What is required, then, is a quickly converging method of obtaining a "reasonably good" approximation to $f(nT/S)$ that is computationally tractable. Thus, some method of polynomial interpolation presents itself as a likely candidate, with linear interpolation being particularly attractive.

In the case of linear interpolation, the "new" pixel (i.e. $f(nT/S)$) is computed from the two adjacent pixels and the desired location of the interpolated point in the inter-sampling interval. Referring to Fig. 42,



LINEAR INTERPOLATION BETWEEN ADJACENT SAMPLES, f(t) IS COMPUTED FROM f(b), f(c) AND LOCATION t

LINEAR INTERPOLATION BETWEEN VERTICALLY AND HORIZONTALLY ADJACENT PIXELS:

$f(t_1)$ IS COMPUTED FROM $f(b_1), f(c_1), t_1$ AS ABOVE
$f(t_2)$ IS COMPUTED FROM $f(b_2), f(c_2), t_2$ AS ABOVE
$f(t)$ IS COMPUTED FROM $f(t_1), f(t_2), t$ AS ABOVE

Fig. 42—Linear interpolation of pixels.

note that the desired pixel location t lies between the original sample locations b and c, and that $f(b)$ and $f(c)$ are the pixel values at these locations. Letting point b be zero on our time axis, a first order interpolating polynomial is

$$\hat{f}(x) = \alpha_1 x + \alpha_0.$$

Referencing the desired pixel location to pixel b by substituting $t - b = x$, and solving for α_1 and α_0 so that the line connects $f(b)$ and $f(c)$, we obtain

$$\hat{f}(t) = [f(c) - f(b)]\,(t - b) + f(b).$$

Thus, we have an easily computable approximation to the signal value at any desired time. Once again, we have only considered the one-dimensional case, although, in general, the new pixel location will fall in a grid of vertically- as well as horizontally-adjacent samples of the original picture. Hence, in the general case of two dimensional compression, the original continuous image is approximated at any point by a sequence of two horizontal and one vertical linear interpolations, as shown in Fig. 42.

To visually evaluate the results of this procedure, Fig. 43 shows a monoscope image that has been expanded to a size of 1.54 in each dimension, while Fig. 44 shows an 0.82 size compression. Clearly, this method produces surprisingly good results, with the rather unsophisticated linear interpolation producing much less objectionable effects than the slowly converging sinc interpolator or the jitter resulting from simple pixel deletions. This level of quality is one that would be judged as quite excellent for a special effects product, and the algorithm is particularly attractive from the hardware implementation standpoint, since the linear interpolation can be performed with two subtractions, one addition, and one multiplication.

However, to see if it is possible to improve compressed image quality even further, let us investigate the use of higher-order polynomials as interpolating functions. The general nth order polynomial

$$\hat{f}(x) = \alpha_n x^n + \alpha_{n-1} x^{n-1} + \ldots + \alpha_1 x + \alpha_0$$

has $n + 1$ coefficients to be solved for, thus indicating that $n + 1$ adjacent pixels will be required for interpolation. Both the general nature of the cubic polynomial and the symmetry gained by using four adjacent pixels for estimation (two on each side of the interpolated point) suggest the use of a cubic interpolator. While several methods of polynomial fitting are available, this application seems to dictate that when the polynomial is evaluated at any one of the actual sample points, the actual sample value should be produced, so as not to replace any data that is known

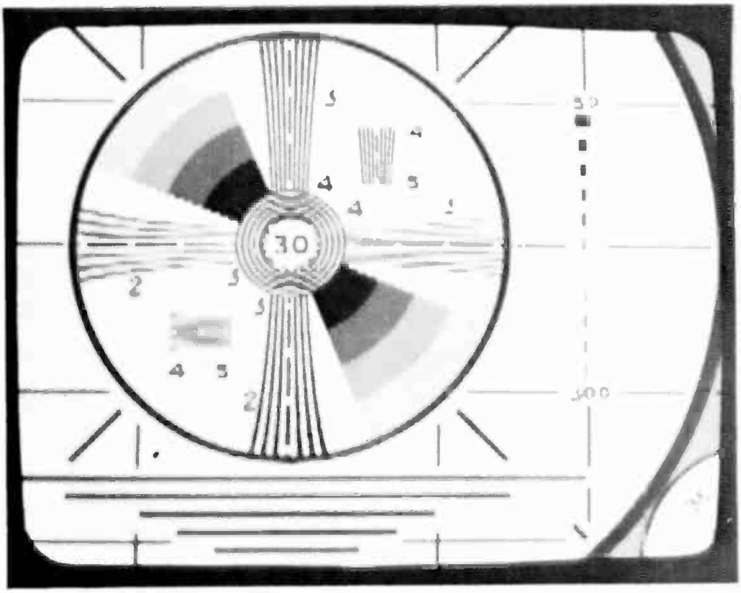


Fig. 43— 1.54 × 1.54 expanded monoscope produced using linear interpolation.

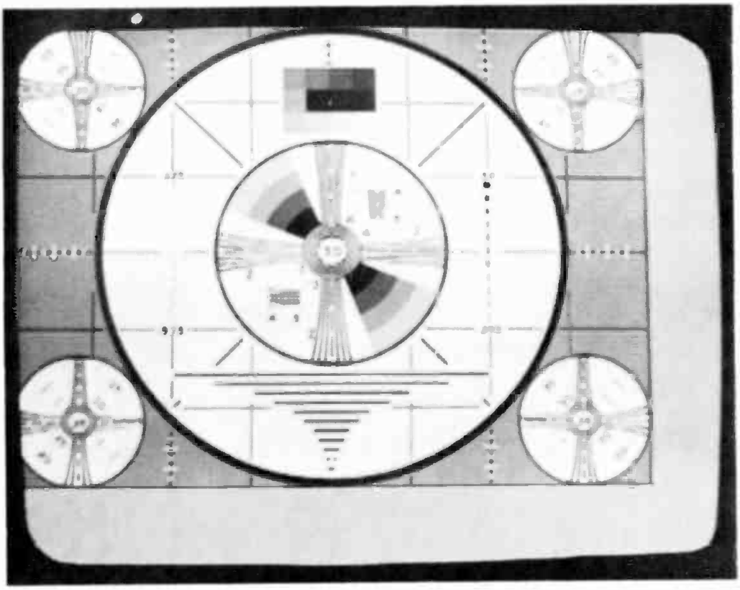
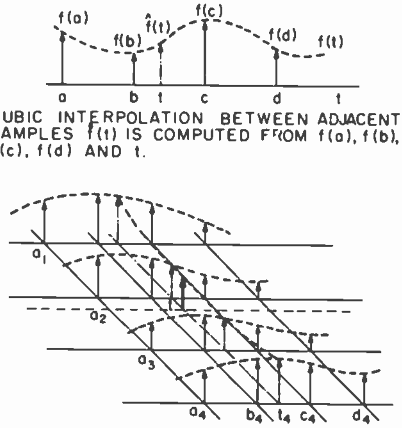


Fig. 44—0.82 × 0.82 compressed monoscope produced using linear interpolation and scaled second-order Butterworth prefiltering.



CUBIC INTERPOLATION BETWEEN ADJACENT SAMPLES $\hat{f}(t)$ IS COMPUTED FROM f(a), f(b), f(c), f(d) AND t.

CUBIC INTERPOLATION BETWEEN VERTICALLY AND HORIZONTALLY ADJACENT PIXELS:

- $\hat{f}(t_i)$ IS COMPUTED FROM $f(a_i)$, $f(b_i)$, $f(c_i)$, $f(d_i)$ AND t_i AS ABOVE, FOR i = 1,2, 3, 4.
- $\hat{f}(t)$ IS COMPUTED FROM $\hat{f}(t_i)$ AND t, AS ABOVE.

Fig. 45—Cubic interpolation of pixels.

to be correct by an approximation. This requirement results in a straightforward method for determination of the polynomial coefficients in terms of the pixel values and interpolated sample location. Referring to Fig. 45, note that the desired pixel location t lies between the original samples b and c, which are adjacent to points a and d, respectively. In order to determine the polynomial coefficients, the following set of equations must be solved:

$$\hat{f}(x) = \alpha_3 x^3 + \alpha_2 x^2 + \alpha_1 x + \alpha_0$$

$$\hat{f}(a) = f(a)$$

$$\hat{f}(b) = f(b)$$

$$\hat{f}(c) = f(c)$$

$$\hat{f}(d) = f(d).$$

where, once again, $f(a)$ represents the pixel value at position a, etc. Since the interpolated pixel location t will always be referenced relative to that of pixel a, and will fall somewhere between pixels b and c, let us set a =

0, $b = 1$, $c = 2$, and $d = 3$, as shown in Fig. 45. We then obtain

$$f(a = 0) = \alpha_0$$

$$f(b = 1) = \alpha_3 + \alpha_2 + \alpha_1 + \alpha_0$$

$$f(c = 2) = 8\alpha_3 + 4\alpha_2 + 2\alpha_1 + \alpha_0$$

$$f(d = 3) = 27\alpha_3 + 9\alpha_2 + 3\alpha_1 + \alpha_0.$$

Solving these equations yields

$$\alpha_3 = \frac{1}{6}[f(d) - f(a)] + \frac{1}{2}[f(b) - f(c)]$$

$$\alpha_2 = \frac{1}{2}[f(a) + f(c)] - f(b) - 3\alpha_3$$

$$\alpha_1 = f(b) - f(a) - \alpha_2 - \alpha_3$$

$$\alpha_0 = f(a).$$

The coefficients were purposely left in a form such that each can be calculated from pixel values and previously computed coefficients. Therefore, the computational circuitry required to perform the coefficient calculations can be minimized by computing the coefficients in the above order. Hence, we have obtained the expressions for the coefficients of an interpolating cubic polynomial in terms of the surrounding pixel values. Since we are referencing the interpolated pixel location to pixel a, our interpolator becomes

$$\hat{f}(t - a) = \alpha_3(t - a)^3 + \alpha_2(t - a)^2 + \alpha_1(t - a) + \alpha_0.$$

where the coefficients are determined as shown above.

Just as in the case of linear interpolation, for the general case of image compression, desired pixels requiring interpolation will fall in between both horizontal and vertical samples. This time, five interpolations are required, four horizontally and one vertically, as shown in Fig. 45. This method also was implemented, and the results of a two dimensional 1.54 expansion and 0.82 compression are shown in Figs. 46 and 47, respectively. A comparison with the images produced by linear interpolation shows no significant improvement in image quality to justify the increase in computational circuitry required to implement the cubic interpolator. It is certainly dubious whether even higher order polynomial interpolators would offer any better results. Thus, while cubic interpolators may produce marginally better results, the compression and expansion of images by resampling through linear interpolation offers both high quality results and ease of implementation.

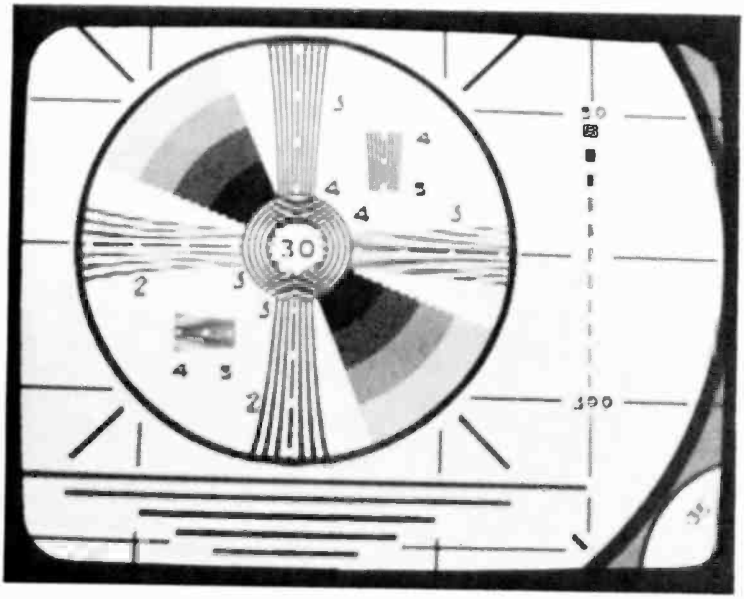


Fig. 46—1.54 × 1.54 expanded monoscope produced using cubic interpolation.

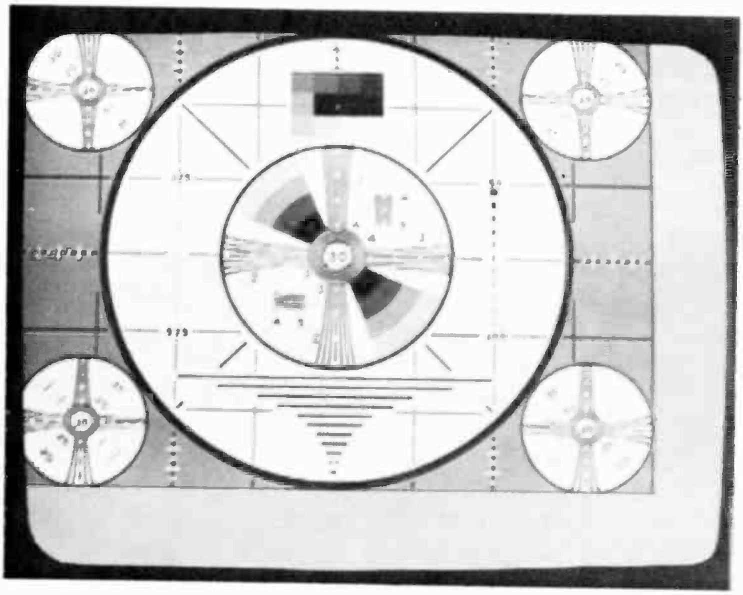


Fig. 47—0.82 × 0.82 compressed monoscope produced using cubic interpolation and scaled second-order Butterworth prefiltering.

8. Conclusions

A number of algorithms for image compression have been examined, ranging from very simple and intuitively developed ones that operate on the composite video signal to the development of a method for variable-ratio compression and expansion of component signals derived from linear systems theory. Further, the short-comings of theoretically perfect interpolation schemes were shown when it was attempted to implement such schemes using finite summations. Hence, the approximate interpolation method of polynomial fitting has been used, and shown to produce excellent results, while being computationally tractable. Also, the effects of nonideal prefiltering have been examined, and the appropriate tradeoffs considered. Clearly, the optimal method of high quality image compression would utilize the combination of high order FIR prefilters and cubic interpolation. However, the cost of producing such a system would probably be prohibitive, due to the amounts of memory and computational circuitry which would be required.

A quite reasonable alternative, which will still produce high quality compressed images, is to utilize the scaled second-order Butterworth prefilter in conjunction with linear interpolation, as was done in the figures illustrating the linear interpolation method. This alternative

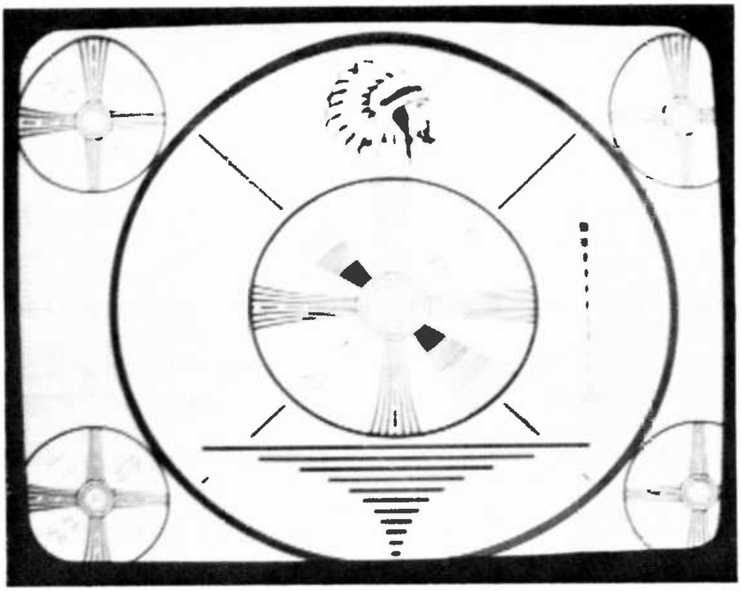


Fig. 48—"Indian head" monoscope pattern used to display video special effects.

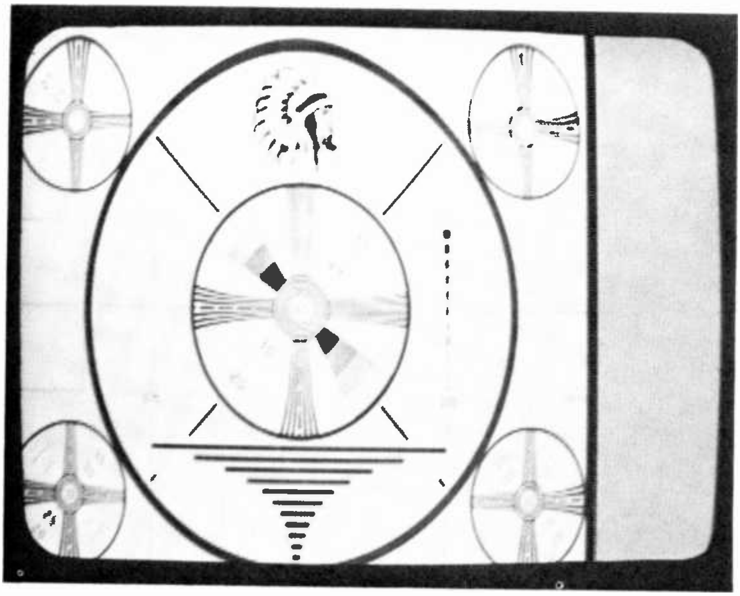


Fig. 49—1 × 0.75 horizontally compressed monoscope pattern.

certainly minimizes the amount of circuitry required for implementation, and produces images of quite high quality. Thus, an implementable scheme for producing high quality, variable ratio compressed images has truly been developed.

To further illustrate the quality of these results, as well as the possibilities for video special effects using the techniques of image compression and expansion, several additional pictures will be presented. Fig. 48 shows the slightly different monoscope pattern used for these examples. Fig. 49 shows the effect of performing a 0.75 compression in the horizontal dimension only, while Fig. 50 shows similar results in the vertical dimension. To show that this method truly maintains high image quality at greatly compressed sizes, Fig. 51 contains images that have been compressed by factors of 0.4, 0.3, and 0.2. Note the excellent quality exhibited even by the 0.2 size image.

To illustrate high quality in expanded images, Fig. 52 shows a 1.54 size image, while Fig. 53 shows an expansion of 2.5 times.

A critical examination of the image reveals what appears to be temporal noise in the image, which has been magnified by the expansion process. For large expansion sizes, temporal noise reduction techniques could be applied to eliminate this noise. Alternatively, a spatial postfilter

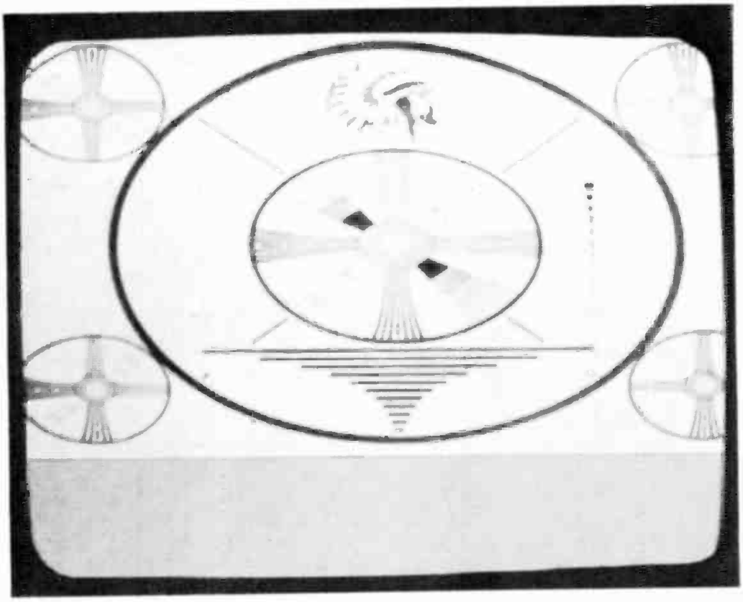


Fig. 50—0.75 × 1 vertically compressed monoscope pattern.

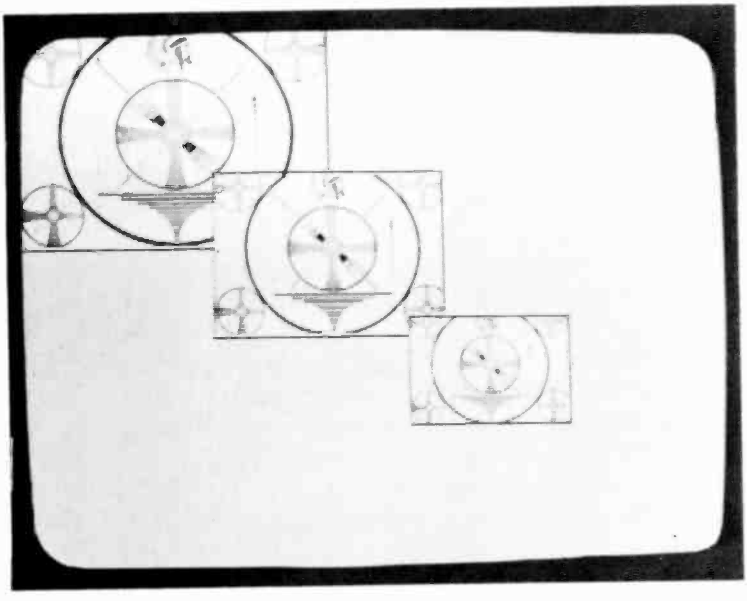


Fig. 51—0.4 × 0.4, 0.3 × 0.3, and 0.2 × 0.2 compressions. Note the high quality of the images even at greatly reduced size.

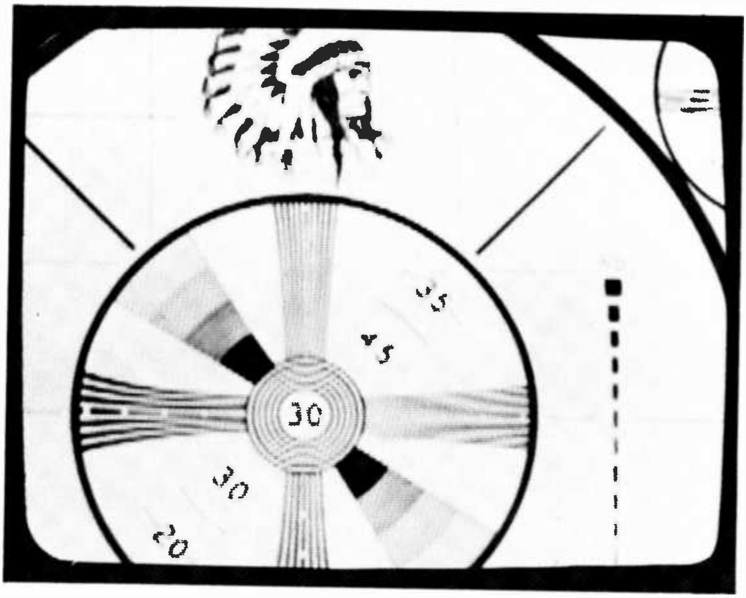


Fig. 52—1.54 × 1.54 expanded monoscope pattern.

could be applied to each of the component signals, just as prefiltering was performed in the case of compression. Thus, postfiltering with cutoff $\nu_c = 1/S$ will eliminate the noise, while passing the entire image spectrum. Fig. 54 shows an expansion of 2.5 times, postfiltered in both spatial directions by second-order digital Butterworth filters. Thus, because of the reduced bandwidth of the expanded image, postfiltering can very effectively remove noise that has been magnified by the expansion.

The variable compression and expansion of images on a dynamic basis can provide a startling special effect, but this can be carried even further. By performing horizontal compressions, and altering the compression ratio on each line by some formula, extremely interesting results can be obtained. Linearly varying the compression ratio produces the effect of Fig. 55, where the image seems to take on a perspective quality. Varying the compressions according to the formula for the chords of a circle transforms a rectangular image into the circular one shown in Fig. 56. Indeed, the effect of gradually transforming the rectangular image into a circle, and then compressing it into a point, or the opposite sequence of operations, would truly be a dramatic effect. Similar operations might be performed with a diamond, or other geometric shapes. Hence, the ability to perform high quality image compression and expansion, a goal

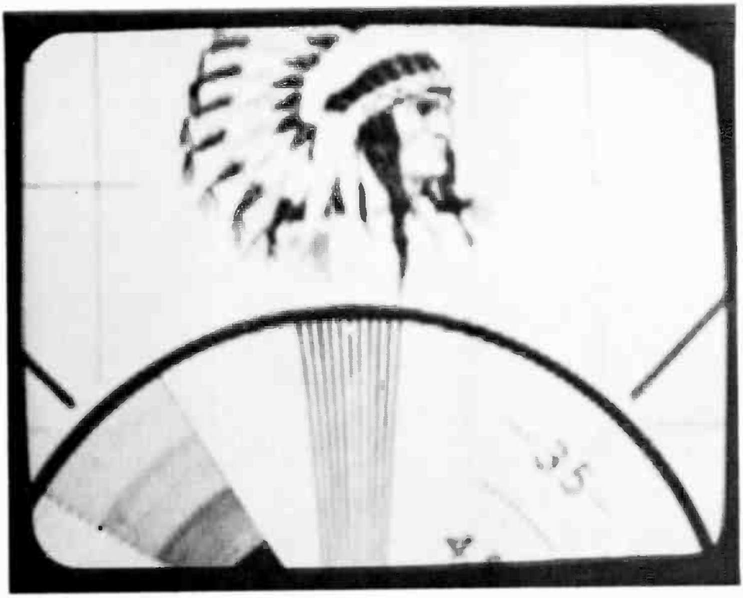


Fig. 53—2.5 × 2.5 expanded monoscope showing noise, particularly in the background.

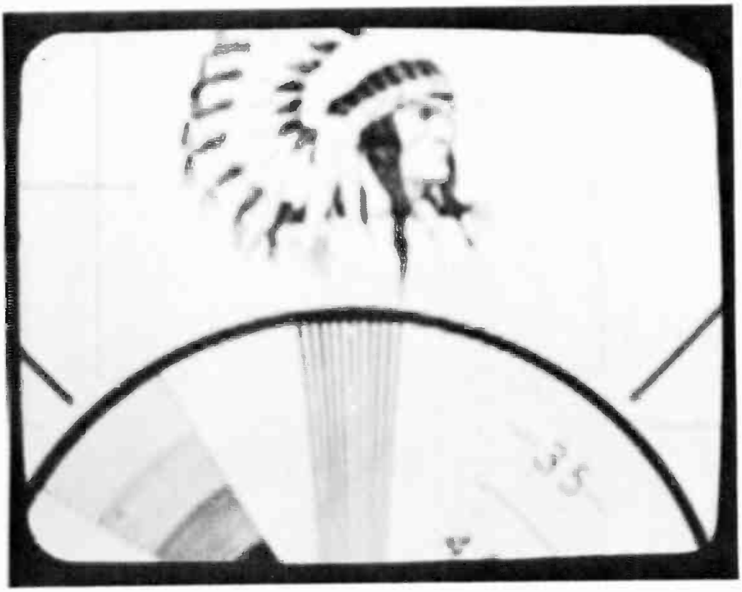


Fig. 54—2.5 × 2.5 expanded monoscope after postfiltering with second-order Butterworth filters. Note the reduction of noise compared to Fig. 53.

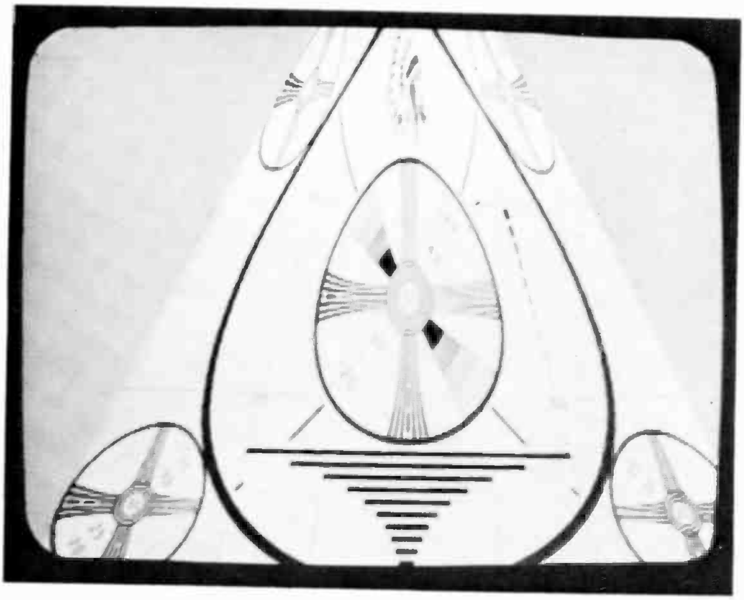


Fig. 55—Altering the compression ratio linearly on a line-by-line bases to give a perspective effect.

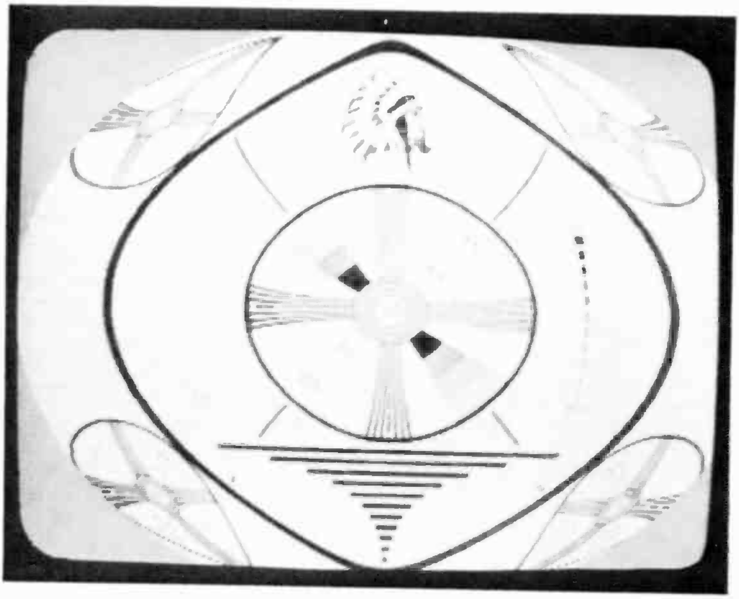


Fig. 56—Altering the compression ration on a line-by-line basis as the chords of a circle produces a circular mapping, or fisheye lens, effect.

which has truly been achieved in this effort, can provide the basis for a whole new class of video special effects.

Acknowledgments

I would like to express my thanks, first of all, to the management of RCA Laboratories, for their continued interest in the area of video special effects, particularly B. J. Lechner, F. J. Marlowe, J. K. Clemens, and C. B. Oakley; also, to R. S. Hopkins, of RCA Broadcast Systems, who has been equally supportive and enthusiastic about this research. Finally, my thanks to S. S. Rao of Villanova University, who first introduced me to the field of digital signal processing, for his always helpful discussions on digital filtering.

References:

[1] A. A. Goldberg, "PCM NTSC Television Characteristics," *J. SMPTE*, **85**, p. 141, March 1976.

[2] J. P. Rossi, "Color Decoding a PCM NTSC Television Signal," *J. SMPTE*, **83**, p. 489, June 1974.

[3] D. E. Pearson, *Transmission and Display of Pictorial Information*, John Wiley and Sons, New York (1975).

[4] Athanasios Papoulis, *Signal Analysis*, McGraw-Hill Book Co., Inc., New York (1977).

[5] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*, Prentice Hall, Inc., Englewood Cliffs, NJ (1975).

[6] Lawrence Rabiner and Bernard Gold, *Theory and Application of Digital Signal Processing*, Prentice Hall, Inc., Englewood Cliffs, NJ (1975).

Bibliography

Bernard Gold and Charles Rader, *Digital Processing of Signals*, McGraw-Hill Book Co., Inc., NY (1969).

A. V. Oppenheim, Ed., *Applications of Digital Signal Processing*, Prentice Hall, Inc., Englewood Cliffs, NJ (1978).

William Stanley, *Digital Signal Processing*, Reston Publishing Co., Inc., Reston, Va. (1975).

Steven Tretter, *Introduction to Discrete-Time Signal Processing*, John Wiley and Sons, Inc., NY (1976).

Visual-Perception-Related Effects in Chinese-Japanese Written Characters

E. O. Johnson and S. Tosima

RCA Research Laboratories, Inc., Machida City, Tokyo, Japan

Abstract—Chinese-Japanese pictorial language characters were analyzed in terms of the standardized writing strokes, or "visual elements," that compose them. When plotted as a function of the number of strokes, a dictionary of characters reveals a normal-type frequency distribution with fine structure arising from stroke grouping patterns. The average stroke group contains about four strokes, an interesting result with counterparts in more familiar examples of how humans organize information. And when strokes are used as a basis for comparison, it is seen that English words and Oriental characters, at least as used in the Japanese language, are approximately equivalent in information content.

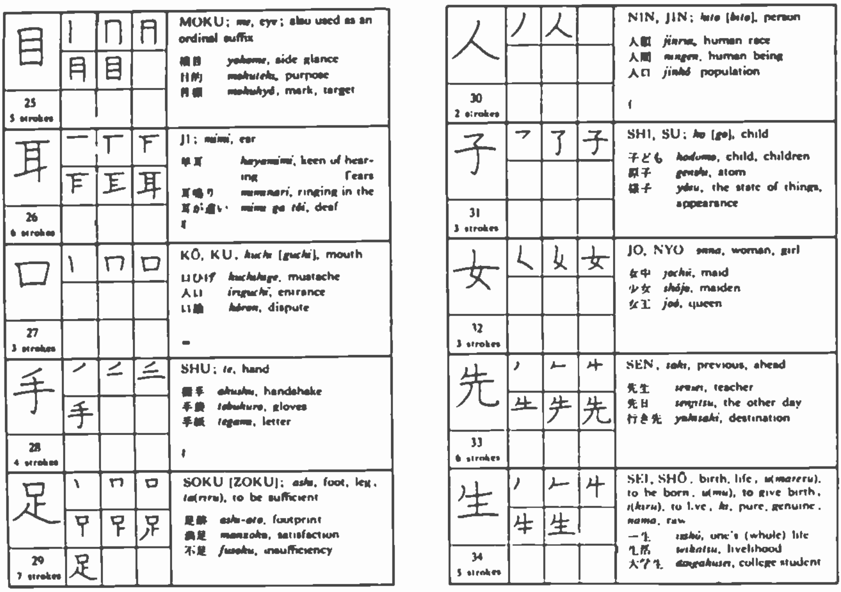
Introduction

The characters in any major written language have evolved over long periods of time and so have a demonstrated practicality in conveying thought information. Long usage should tend to wash away redundancy and ambiguity, leaving only the essential visual elements. Evolution would probably tend to favor characters having economy of construction, ease of recognition, and breadth of meaning. Evolved characters must be the ones that approach the best match to the visual system. In turn, much about this system should be revealed in these written characters.

The Chinese-Japanese pictorial characters, called "Kanji" in Japan, provide a particularly good example of such written characters. These have evolved over thousands of years, and in terms of total usage are of comparable world importance to the purely phonetic characters used in Western alphabets. Some of the simple forms of the Kanji remain unchanged after several thousand years; characters of more complex shapes have been simplified over the years. This simplification process has continued up to the present day in China and in Japan. Moreover, Kanji characters have other qualities that make them particularly meaningful and convenient to study with respect to visual phenomena. They are well standardized, documented, and consist of a very large diversity of configurations that convey the human thought spectrum. Characters are constructed by pen or brush strokes standardized in application sequence, relative size and shape, in orientation, and in location in the character space.[1] Fig. 1 is a copy of two pages taken from a Japanese standard list of 1850 Kanji characters particularly convenient to use as the primary source of data for the following analysis.

It will be noted from the figure that standard stokes are vertical, horizontal, or diagonal lines. In some cases the strokes may be relatively short in length, or bend around a corner. As shall be seen, the number of these standard strokes in a character, or sub-section of a character, are a convenient analytical parameter. In a practical sense, each stroke is a "bit" of visual information.

The main concern in this paper is the analysis of the numerical aspects of the grouping and subgrouping of strokes in the Kanji character



目
25
5 strokes

MOKU; *me*, eye; also used as an ordinal suffix

横目 *yokome*, side glance
目的 *mokuteki*, purpose
目標 *mokuhyō*, mark, target

耳
26
6 strokes

JI; *mimi*, ear

早耳 *hayamimi*, keen of hear-ing Fears
耳鳴り *mimінari*, ringing in the
耳が遠い *mimi ga tōi*, deaf

口
27
3 strokes

KŌ, KU, *kuchi* [*guchi*], mouth

口ひげ *kuchihige*, mustache
入口 *iriguchi*, entrance
口論 *kōron*, dispute

手
28
4 strokes

SHU; *te*, hand

握手 *akushu*, handshake
手袋 *tebukuro*, gloves
手紙 *tegami*, letter

足
29
7 strokes

SOKU [ZOKU]; *ashi*, foot, leg, *ta*(*riru*), to be sufficient

足跡 *ashi-ato*, footprint
満足 *manzoku*, satisfaction
不足 *fusoku*, insufficiency

人
30
2 strokes

NIN, JIN; *hito* [*bito*], person

人類 *jinrui*, human race
人間 *ningen*, human being
人口 *jinkō* population

子
31
3 strokes

SHI, SU; *ko* [*go*], child

子ども *kodomo*, child, children
原子 *genshi*, atom
様子 *yōsu*, the state of things, appearance

女
32
3 strokes

JO, NYO *onna*, woman, girl

女中 *jochū*, maid
少女 *shōjo*, maiden
女王 *joō*, queen

先
33
6 strokes

SEN, *saki*, previous, ahead

先生 *sensei*, teacher
先日 *senjitsu*, the other day
行き先 *yukisaki*, destination

生
34
5 strokes

SEI, SHŌ, birth, life, *u*(*mareru*), to be born, *u*(*mu*), to give birth, *i*(*kiru*), to live, *ki*, pure, genuine, *nama*, raw

一生 *isshō*, one's (whole) life
生活 *seikatsu*, livelihood
大学生 *daigakusei*, college student

Fig. 1—Typical Kanji as they appear in a standard list of 1850 characters used in Japan. Shows stroke order and composition.

spectrum. One conclusion from this study is that there exists a strong tendency on the average for the strokes to have four-fold groupings, a feature that seems to have escaped previous notice. Apparently, if given the choice, the human system prefers to organize its image data in groups limited to about four items. Smaller groupings may not be adequate to carry the required information; larger groupings are inconvenient for rapid comprehension. This result lies within the known human limiting capacity for handling item groupings up to 7 ± 2.[2] Another result is that when strokes are used as a basis for comparison, it is seen that English words and the Oriental characters are approximately equivalent in information content.

Kanji Character Spectrum

Fig. 2 shows the number of Kanji characters $H(N)$ plotted in linear scale as a function of the number of strokes (N) in the characters. The data was taken from the previously noted list of 1850 Japanese standard characters of which Figs. 3 and 4 show typical examples in the various stroke categories.

As can be seen in Figs. 2 and 3, there are only a few single-stroke characters because of the limited number of distinguishable characters

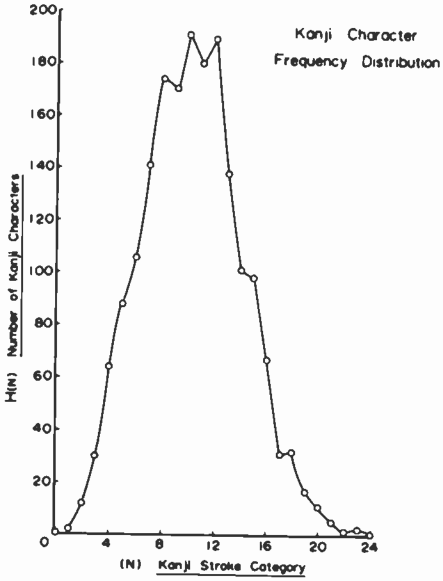
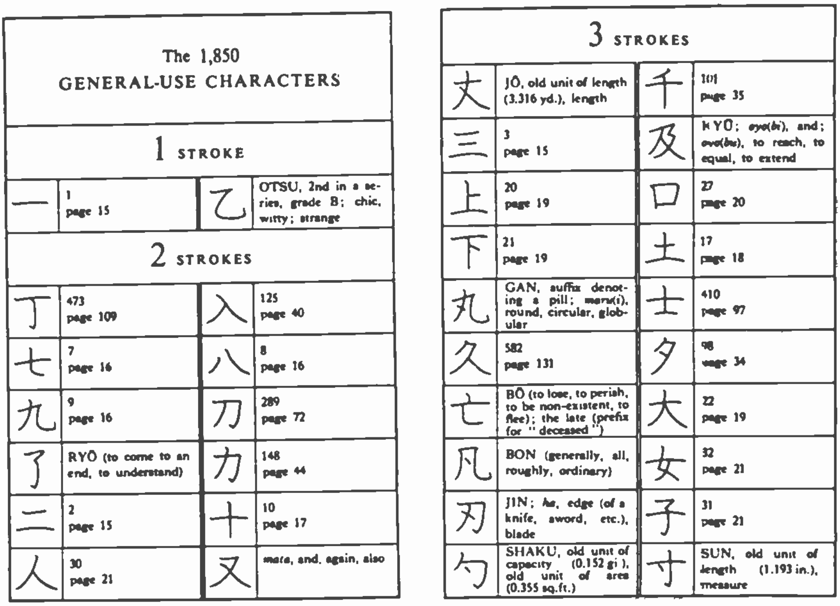


Fig. 2—Linear plot of Kanji character frequency spectrum as function of number of writing strokes.



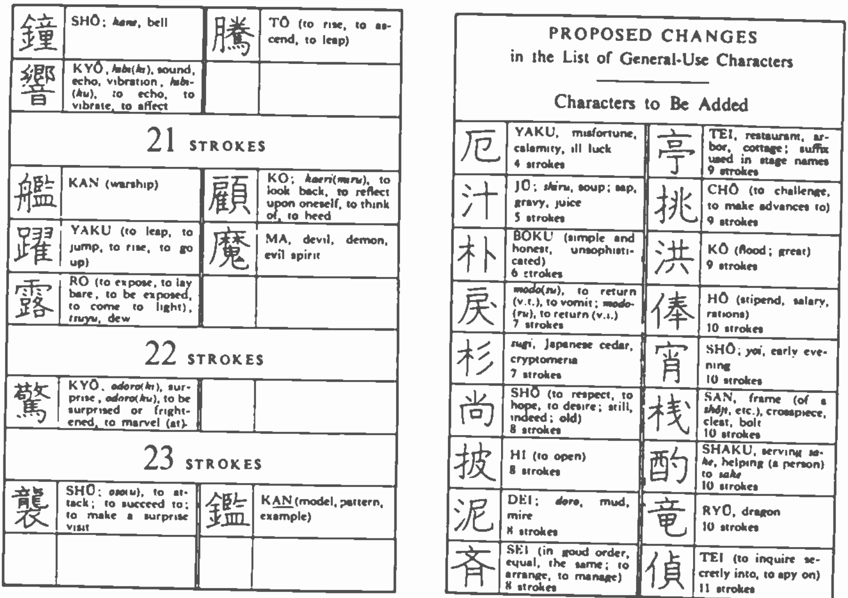
The 1,850 GENERAL-USE CHARACTERS			
1 STROKE			
一	1 page 15	乙	OTSU, 2nd in a series, grade B; chic, witty; strange
2 STROKES			
丁	473 page 109	入	125 page 40
七	7 page 16	八	8 page 16
九	9 page 16	刀	289 page 72
了	RYŌ (to come to an end, to understand)	力	148 page 44
二	2 page 15	十	10 page 17
人	30 page 21	又	*mata*, and, again, also

3 STROKES			
丈	JŌ, old unit of length (3.316 yd.), length	千	101 page 35
三	3 page 15	及	KYŌ; *oyo(bi)*, and; *oyo(bu)*, to reach, to equal, to extend
上	20 page 19	口	27 page 20
下	21 page 19	土	17 page 18
丸	GAN, suffix denoting a pill; *maru(i)*, round, circular, globular	士	410 page 97
久	582 page 131	夕	98 page 34
亡	BŌ (to lose, to perish, to be non-existent, to flee); the late (prefix for "deceased")	大	22 page 19
凡	BON (generally, all, roughly, ordinary)	女	32 page 21
刃	JIN; *ha*, edge (of a knife, sword, etc.), blade	子	31 page 21
勺	SHAKU, old unit of capacity (0.152 gi), old unit of area (0.355 sq.ft.)	寸	SUN, old unit of length (1.193 in.), measure

Fig. 3—Typical Kanji in the 1, 2, and 3-stroke categories as they appear on pages of 1850-character dictionary.

that can be constructed with just one stroke. For example, one might suppose that the location of a single horizontal stroke in the top, middle, or bottom of a character space could be used to signify different characters. In everyday practical usage, however, such different locations of a single stroke in a character space may not be clearly distinguishable from each other, particularly if written in haste, and so would not survive the evolutionary process.

A greater number of practical two-stroke characters is possible than ones having only a single stroke, and even greater numbers are possible with three strokes, four strokes, and so on. However, as more and more strokes are added, the space in the visual square gets so cluttered that the number of perceptually distinguishable characters in a given stroke category reaches a practical limit and starts to decrease. In addition, writing time and effort increases with the number of strokes. Accordingly, the frequency curve reaches a maximum, then decreases. As will be seen later, the perturbations in the region of the maximum are due to rearrangements of the stroke groupings that apparently allow the system to digest more strokes without substantial reduction in the number of visually useful characters. However, when the 13th stroke is added, the number of useful characters drops sharply. The perturbations that appear at 15 and 18 strokes are also due to rearrangements of the stroke groupings.



鐘	SHŌ: *kane*, bell	騰	TŌ (to rise, to ascend, to leap)
響	KYŌ, *hibi(ki)*, sound, echo, vibration, *hibi(ku)*, to echo, to vibrate, to affect		
21 STROKES			
艦	KAN (warship)	顧	KO; *kaeri(miru)*, to look back, to reflect upon oneself, to think of, to heed
躍	YAKU (to leap, to jump, to rise, to go up)	魔	MA, devil, demon, evil spirit
露	RO (to expose, to lay bare, to be exposed, to come to light), *tsuyu*, dew		
22 STROKES			
驚	KYŌ, *odoro(ki)*, surprise, *odoro(ku)*, to be surprised or frightened, to marvel (at).		
23 STROKES			
襲	SHŌ: *oso(u)*, to attack; to succeed to; to make a surprise visit	鑑	KAN (model, pattern, example)

PROPOSED CHANGES
in the List of General-Use Characters

Characters to Be Added

厄	YAKU, misfortune, calamity, ill luck 4 strokes	亭	TEI, restaurant, arbor, cottage; suffix used in stage names 9 strokes
汁	JŪ; *shiru*, soup; sap, gravy, juice 5 strokes	挑	CHŌ (to challenge, to make advances to) 9 strokes
朴	BOKU (simple and honest, unsophisticated) 6 strokes	洪	KŌ (flood; great) 9 strokes
戻	*modo(su)*, to return (v.t.), to vomit; *modo(ru)*, to return (v.i.) 7 strokes	俸	HŌ (stipend, salary, rations) 10 strokes
杉	*sugi*, Japanese cedar, cryptomeria 7 strokes	宵	SHŌ; *yoi*, early evening 10 strokes
尚	SHŌ (to respect, to hope, to desire; still, indeed; old) 8 strokes	桟	SAN, frame (of a *shōji*, etc.), crosspiece, cleat, bolt 10 strokes
披	HI (to open) 8 strokes	酌	SHAKU, serving *sake*, helping (a person) to *sake* 10 strokes
泥	DEI; *doro*, mud, mire 8 strokes	竜	RYŌ, dragon 10 strokes
斉	SEI (in good order, equal, the same; to arrange, to manage) 8 strokes	偵	TEI (to inquire secretly into, to spy on) 11 strokes

Fig. 4—Typical Kanji in the 21, 22, and 23-stroke categories as they appear on pages of 1850-character dictionary.

To summarize, the left-hand portion of the Kanji frequency distribution curve (Fig. 2) is stroke limited, the right-hand portion complexity limited. Limits for perception complexity set in much before the appearance of visual element resolution problems. The left-hand portion of the curve represents simpler and more frequently used characters as judged from the standard list.[1] These simpler characters also tend to be older as judged from typical museum exhibitions of archeological artifacts. Furthermore, as is well known, the right-hand portion of the curve represents characters that are largely composed of groups of the simpler characters. In many cases these simpler characters have been somewhat distorted in shape to achieve a suitable visual fit into the available space.

Curve behavior beyond about 20 strokes does not seem to be significant for our present purposes. Both arm and the eye grow weary beyond this point.

Fig. 5 shows a log-log plot of the data in Fig. 2 (circles). Superimposed on this are data from a more detailed, older, Japanese dictionary[3] (squares) and from a newer Chinese dictionary[4] (triangles). Data for these curves are given in Table 1. The general characteristics of the three curves are quite similar, even though the newer character populations are substantially abbreviated versions of the older one. There is a curve

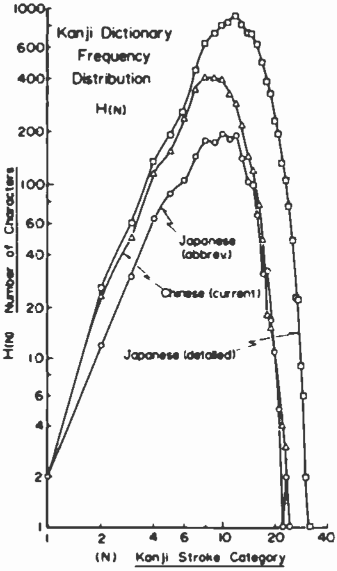


Fig. 5—Log-log plot of Fig. 2 with addition of frequency data taken from a much larger Japanese dictionary, and from a current Chinese dictionary.

maximum between 8 to 12 strokes, and when the low-stroke-number side of this is viewed in detail it is seen to be smoother than the high-stroke-number side. The perturbations on the high side occur roughly at the same number of strokes. On the low-stroke side of the curve maxima, at about four strokes, there is a noticeable change of slope that suggests a change in the basic structure of the characters. In particular, it was felt that this change related to the manner of stroke grouping within the characters. This belief prompted the more detailed analysis that follows.

Procedure for Counting Stroke Groups

The grouping of items to facilitate mental processing is a familiar phenomenon and has been extensively studied.[2] The grouping of strokes in patterns inside Kanji characters is conventionally used as a mnemonic aid for students. The simpler characters with a relatively small number of strokes are usually taught first. These form the basis for dealing later with the more complicated characters mainly composed of subgroups of the simpler ones fitted together in various ways. For example, the 20-stroke Kanji "SHŌ," is composed of three readily distinguishable subgroups that are characters by themselves, and appeared as such

Table 1—Kanji Character Frequency Data as a Function of Number of Strokes for Japanese Standard 1850-Character List,[1] for Japanese Dictionary of 9768 characters,[3] and for Current Chinese Dictionary of 3372 Characters.[4]

Stroke Category		Kanji characters	
	Japan Abbrev.[1]	Japan Detailed[3]	Current Chinese[4]
(N)		$H(N)$	
1	2	2	2
2	12	26	23
3	30	60	50
4	64	133	117
5	88	192	155
6	106	256	243
7	141	442	345
8	174	621	400
9	170	703	390
10	191	784	385
11	180	825	320
12	189	883	287
13	138	798	214
14	101	703	142
15	98	704	118
16	67	604	76
17	31	494	48
18	32	377	18
19	17	322	15
20	11	224	11
21	5	192	5
22	1	131	4
23	2	104	3
24	—	75	1
25	—	48	—
26	—	23	—
27	—	22	—
28	—	9	—
29	—	6	—
30	—	2	—
31	—	1	—
32	—	1	—
33	—	1	—
Sum $\Sigma H(N)$	1850	9768	3372
Dispersion (σ)	3.77	4.63	3.43
Median	9.72	11.95	8.90

earlier in the dictionary listing (top left in Fig. 4). In fact, all of the Kanji in the figure are composed of subgroups in this same manner. Subgroup shape distortions to achieve "visual balance" are relatively minor in these particular examples; it is easy to recognize the subgroups as characters that appeared earlier in the Kanji spectrum.

In counting the subgroups in a character to provide the data for our analysis we proceeded as follows. The entire standard list of characters was visually scanned starting with one-stroke items and continuing throughout the entire dictionary list, terminating with 23-stroke characters. The characters with one, two, and three strokes are quite simple and are considered to consist of only one visually inseparable group of strokes. In the four-stroke category, however, it was readily seen that many characters were either composed of two of the simpler characters, or one of these plus a new configuration of grouped strokes. In the succeeding stroke categories, such "building block" buildup becomes increasingly pronounced; in some of the more complicated characters we could recognize up to five building blocks. In almost all cases it was possible to make a clear distinction of such groupings, particularly after the dictionary had been scanned several dozen times to improve familiarity with the stroke patterns. In addition, to refine the perception of stroke groups, the presence of many particular subgroups was tracked through the character spectrum.

Whereas the preceding part of the procedure is straightforward, there remains a question of definition when the subgroups, themselves, are made part of a larger subgroup. This happens, for example, when "千" and "口" are combined into "舌", which is visually, if not historically, a subgroup of the character "話". Our procedure in such a case is to count "舌" as two groups when it first appears in the Kanji frequency spectrum, and then count it as a single group when it appears further on in the spectrum as a clearly recognizable component in a more complicated Kanji such as "話" noted above. This manner of counting favors the natural human tendency to group items at ever higher combinatorial levels, and is consistent with the concept of grouping discussed in this paper.

Notations and Relationships

There are various simple relationships between numbers of Kanji, strokes, Kanji subgroups, and the various useful averages that will be considered in this study. These, together with the parameter notations, are as follows.

Every Kanji is composed of a given number of strokes N. The total number of Kanji in the category of N-strokes is $H(N)$. The total number of Kanji in all stroke categories, $N = 1$ to $N = N$, is the summation of all Kanji having one stroke, two strokes, and so on, as given by

$$H = \sum_{N=1}^{N} H(N). \qquad [1]$$

A Kanji having N-strokes may have one grouping, or cluster, of strokes ($n = 1$) or, as shall be seen, there may be as many as five different groups of strokes ($n = 5$). The N total strokes in a Kanji may be allocated differently between various subgroups or, as a special case, the allocation may be equal. In any case, the average stroke allocation in a Kanji is N/n.

Among the $H(N,n)$ Kanji having N strokes, some Kanji $H(N,1)$ will have $n = 1$ subgroups of strokes, some will have $n = 2$, $n = 3$, and so on. The sum of all these Kanji $H(N,n)$ having the same stroke number N, but different numbers of subgroups, $H(N,1)$, $H(N,2)$, etc., will be $H(N)$ as described by

$$H(N) = \sum_{n=1}^{n} H(N,n). \qquad [2]$$

The total number of strokes in the Kanji class having N strokes is $N \cdot H(N)$, and the total number of strokes summed over all values of N divided by the total number of Kanji from Eq. [1] gives the average number of strokes per Kanji:

$$\langle N \rangle = \frac{\sum_{N=1}^{N} N \cdot H(N)}{H}. \qquad [3]$$

In a Kanji $H(N,n)$ the total number of stroke groups is $nH(N,n)$, or $S(N,n)$. If this is summed up over N for a fixed value of n, say $n = 1$, we get the total number of groups that exist as singles in the Kanji spectrum. If $n = 2$, we get the total number of groups that exist as doubles, and so on for $n = 3, 4$, and up to n. The summation of all of these gives the total number of stroke groups G in all the Kanji:

$$G = \sum_{n=1}^{n} \sum_{N=1}^{N} nH(N,n). \qquad [4]$$

If this is divided by the total number of Kanji H, we get the average number of stroke groups per Kanji:

$$\left\langle \frac{G}{H} \right\rangle = \frac{\sum_{n=1}^{n} \sum_{N=1}^{N} nH(N,n)}{\sum_{N=1}^{N} H(N)}. \qquad [5]$$

Returning to the discussion of average numbers of strokes, note that

$$\sum_{N=1}^{N} \frac{1}{n} N \cdot H(N,n) = \frac{1}{n} \sum_{N=1}^{N} N \cdot H(N,n) \qquad [6]$$

for a given value of n gives the average number of strokes in all the groups of class n Kanji. If this is divided by the total number of groups of class n, we get the average number of strokes per group:

$$\left\langle \frac{N}{G_n} \right\rangle = \frac{\frac{1}{n} \sum_{N=1}^{N} N \cdot H(H,n)}{\sum_{N=1}^{N} H(N,n)}, \qquad [7]$$

where G_n is the number of groups of class n. The average number of strokes in all groups, $G_1, G_2, G_3, \ldots G_n$, is obtained from the summation

$$\left\langle\frac{N}{G}\right\rangle = \frac{\sum_{n=1}^{n}\frac{1}{n}\sum_{N=1}^{N} N\cdot H(N,n)}{\sum_{n=1}^{n}\sum_{N=1}^{N} H(N,n)} = \frac{\sum_{n=1}^{n}\frac{1}{n}\sum_{N=1}^{N} N\cdot H(N,n)}{H}. \qquad [8]$$

The same relations hold for later discussion of the Kanji building block stroke configurations $B(N,n)$, sometimes called "radicals," that are used to build up the Kanji characters.

The next step is to apply the above to actual data from the various dictionaries.

Grouping Results

Fig. 6 shows the Kanji frequency distribution $H(N,n)$ for $n = 1, 2, 3, 4$, and 5 as obtained from Table 2. The bounding curve is $H(N)$. As mentioned earlier, all characters up to $N = 3$ exist only with a single group of strokes ($n = 1$); there is no subgrouping. But when $N = 4$, dual

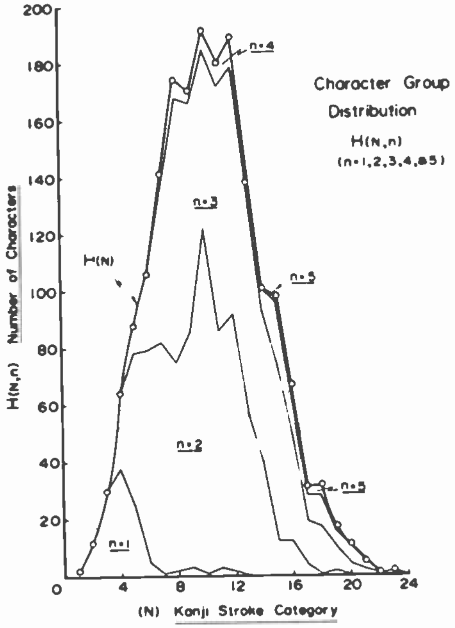


Fig. 6—Linear plot of character group distribution $H(N,n)$ for $n = 1, 2, 3, 4$, and 5.

Table 2—Character Group Data

Strokes	Kanji characters	Total strokes	$H(N,n)$					
			$n = 1$	2	3	4	5	$\sum_{n=1}^{n} H(N,n)$
(N)	H(N)	N·H(N)						
1	2	2	2	—	—	—	—	2
2	12	24	12	—	—	—	—	12
3	30	90	30	—	—	—	—	30
4	64	256	38	26	—	—	—	64
5	88	440	25	53	10	—	—	88
6	106	636	5	74	27	—	—	106
7	141	987	1	81	52	7	—	141
8	174	1392	—	75	93	6	—	174
9	170	1520	3	83	80	4	—	170
10	191	1910	1	121	63	6	—	191
11	180	1980	3	83	86	8	—	180
12	189	2268	2	89	88	10	—	189
13	138	1794	—	57	70	11	—	138
14	101	1414	—	39	54	8	—	101
15	98	1470	—	12	62	22	2	98
16	67	1072	—	12	37	16	2	67
17	31	527	—	4	15	9	3	31
18	32	576	—	1	16	11	4	32
19	17	323	—	2	8	5	2	17
20	11	220	—	—	4	7	—	11
21	5	105	—	—	3	2	—	5
22	1	22	—	—	1	—	—	1
23	2	46	—	—	1	1	—	2
	1850	19084	122	812	770	133	13	1850

$$\text{Av. strokes per Kanji} = \frac{19084}{1850} = 10.32$$

subgrouping ($n = 2$) appears; at $N = 5$, triple; at $N = 7$, quadruple; at $N = 15$, quintuple. There is evident association of these groupings with the shape perturbations on the bounding curve $H(N)$.

Fig. 7 shows the subgroup frequency distribution $S(N,n)$ for $n = 1$, 2, 3, 4, and 5 as obtained from Table 3 (see previous section for evaluation of $S(N,n)$). The bounding curve $S(N)$ represents the sum $S(N)$. The perturbation effect of the subgrouping on the character distribution curve is particularly evident (Fig. 2).

Table 4 shows the distribution $N{\cdot}H(N,n)$ and the calculation of Eqs. [7] and [8] at the bottom of the page. Calculation of Eq. [5] is shown at the bottom of Table 3; calculation of Eq. [3], at the bottom of Table 2.

Fig. 8 shows the Kanji building block characters $B(N,n)$ as listed on the inside cover of the large Japanese dictionary.[3] These are numerically listed (Table 5). The notations and parameter interrelationships are analogous to those discussed in the preceding section. The data in Table 5 is plotted in Fig. 9. This plot only shows the curves for $n = 1, 2$, and 3; data for $n = 4$ and 5 are not shown because that would complicate the figure. However, all data is taken into account as noted below.

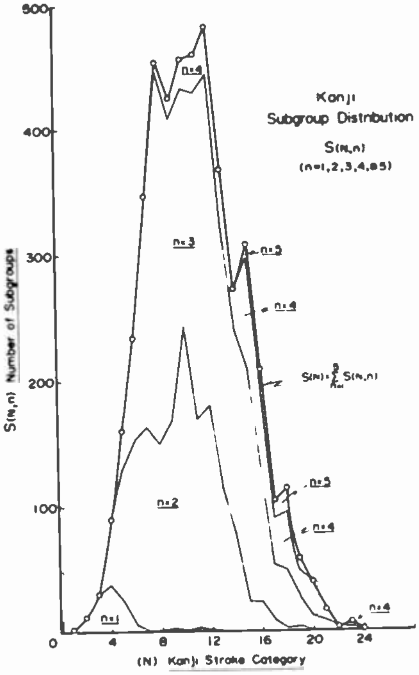


Fig. 7—Linear plot of Kanji subgroup stroke distribution $S(N,n)$ for $n = 1, 2, 3, 4$, and 5.

Table 3—Character Group Distribution and Results (from 1850 Character List[1]). Shows Kanji Group Distribution $S(N,n) = n{\cdot}H(N,n)$ and Calculation of Average Number of Groups per Kanji.

Strokes (N)	Kanji characters $H(N)$	Total strokes $N{\cdot}H(N)$	$n{\cdot}H(N,n) = S(N,n)$				
			$n = 1$	2	3	4	5
1	2	2	2	—	—	—	—
2	12	24	12	—	—	—	—
3	30	90	30	—	—	—	—
4	64	256	38	52	—	—	—
5	88	440	25	106	30	—	—
6	106	636	5	148	81	—	—
7	141	987	1	162	156	28	—
8	174	1392	—	150	279	24	—
9	170	1520	3	166	240	16	—
10	191	1910	1	242	189	24	—
11	180	1980	3	166	258	32	—
12	189	2268	2	178	264	40	—
13	138	1794	—	114	210	44	—
14	101	1414	—	78	162	32	—
15	98	1470	—	24	186	88	10
16	67	1072	—	24	111	64	10
17	31	527	—	8	45	36	15
18	32	576	—	2	48	44	20
19	17	323	—	4	24	20	10
20	11	220	—	—	12	28	—
21	5	105	—	—	9	8	—
22	1	22	—	—	3	—	—
23	2	46	—	—	3	4	—
	1850	19084	122	1624	2310	532	65

Total Groups = 4653

$$\text{Average Number of groups per Kanji} = \frac{4653}{1850} = 2.52$$

Table 6 shows the distribution of $N{\cdot}B(N,n)$ and calculation of the average number of strokes for each class $n = 1, 2, 3, 4$, and 5, as well as the average number of strokes for all groups. The average number of stroke groups per building block B is calculated at the bottom of Table 5 along with the average number of strokes per building block, for all blocks.

Discussion of Grouping Results

The first feature of note is the evident relationship between the manner of stroke grouping and the bumps on the Kanji distribution curves (Figs. 2 and 5). This relationship is particularly evident near the 8, 10, 15, and 18 stroke categories (Fig. 7). The more gentle bump near $N = 4$ (Fig. 2) is associated with the start of multiple grouping, a feature consistent with the marked tendency toward the four-fold grouping of strokes discussed below in more detail.

The calculations at the bottom of Table 4 show that the average number of strokes per group for the groups that appear as singles in a Kanji is 4.3; for doubles, 4.74; for triples, 3.78; for quadruples, 3.56; and for the relatively few quintuples found, 3.43. Of particular interest is the overall average for all groups, 4.22 strokes per group.

This overall average for all groups is to be related to the average number of stroke groups per Kanji character, 2.52, calculated at the bottom of Table 3; and with the average number of strokes per Kanji, 10.32, whose calculation is shown at the bottom of Table 2. That is, one

Table 4—Character Stroke-Grouping Distribution and Results. Shows Total Strokes in Each Kanji Group Class: Kanji with n = 1, 2, 3, 4, or 5 groups of strokes. Shows Average Number of Strokes for Each Class and Overall Average.

Strokes	Kanji characters	Total strokes	$N \cdot H(N,n)$				
(N)	$H(N)$	$N \cdot H(N)$	$n = 1$	2	3	4	5
1	2	2	2	—	—	—	—
2	12	24	24	—	—	—	—
3	30	90	90	—	—	—	—
4	64	256	152	104	—	—	—
5	88	440	125	265	50	—	—
6	106	636	30	440	162	—	—
7	141	987	7	567	364	49	—
8	174	1392	0	600	744	48	—
9	170	1520	27	747	720	36	—
10	191	1910	10	1210	630	60	—
11	180	1980	33	913	946	88	—
12	189	2268	24	1068	1056	120	—
13	138	1794	0	741	910	143	—
14	101	1414	—	546	756	112	—
15	98	1470	—	180	930	330	30
16	67	1072	—	192	592	256	32
17	31	527	—	68	255	153	51
18	32	576	—	18	288	198	72
19	17	323	—	38	152	95	38
20	11	220	—	—	80	140	—
21	5	105	—	—	63	42	—
22	1	22	—	—	22	—	—
23	2	46	—	—	23	23	—
	1850	19084	524	7697	8743	1893	223
	$\frac{1}{n}\sum_{N=1}^{23} N \cdot H(N,n)$		524	3848.5	2917	473.3	44.6
						$\sum = 7804.4$	

Av Strokes (Eq. [7]): for $n = 1$, $\frac{524}{122} = 4.30$; for $n = 2$, $\frac{3848.5}{812} = 4.74$; for $n = 3$.

$\frac{2914}{770} = 3.78$; for $n = 4$, $\frac{473.3}{133} = 3.56$; for $n = 5$, $\frac{44.6}{13} = 3.43$

Av. Strokes for Total Groups: $\frac{7804.4}{1850} = 4.22$

might expect that

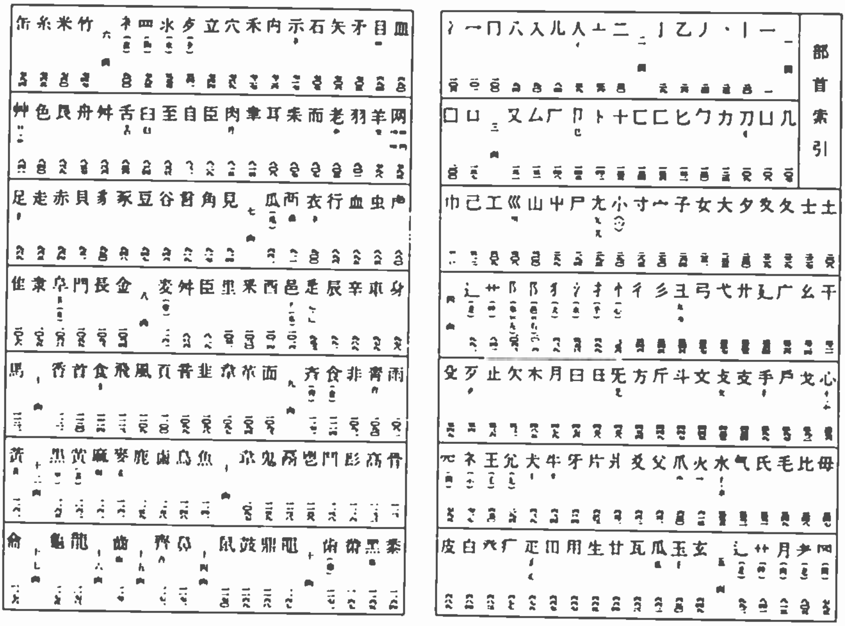
$$\left\langle\frac{\text{Strokes}}{\text{Kanji}}\right\rangle = \left\langle\frac{\text{Groups}}{\text{Kanji}}\right\rangle \cdot \left\langle\frac{\text{Strokes}}{\text{Group}}\right\rangle;$$

and indeed this is nearly the case:

$$\begin{aligned} 10.32 &\cong 2.52 \cdot 4.22 \\ &\cong 10.63. \end{aligned}$$

Actually, the degree of equality depends upon the Kanji and group frequency distributions. If these distributions are severely skewed or highly irregular, the equality does not hold as can be seen for particular examples, one of which is discussed later.

We interpret the above noted allocation of strokes and groups as follows. If a number of visual items is to be grouped to facilitate mental processing, these items will be divided up, like digits on license plates or in telephone numbers, so that the largest grouping will not exceed approximately four items. Kanji characters with their thousands of years of evolution seem a particularly good vehicle to demonstrate this type of human behavior. In the case of Kanji, it seems unlikely that this four-fold average division was done by conscious design as it has been with license plates or phone numbers.



部首索引

Fig. 8—Kanji building block elements (radicals) as they appear on inside cover of large Japanese dictionary.

Table 5—Building Block Group Distribution and Results. Shows Distribution of Blocks $B(N,n)$ for $n = 1, 2, 3, 4,$ and 5, Number of Strokes Per Average Block, and Number of Stroke Groups.

Stroke category	Building blocks	Total strokes	$B(N,n)$				
(N)	$B(N)$	$N{\cdot}B(N)$	$n = 1$	2	3	4	5
1	6	6	6	—	—	—	—
2	23	46	18	5	—	—	—
3	39	117	18	21	—	—	—
4	43	172	15	28	—	—	—
5	27	135	6	21	—	—	—
6	30	180	7	22	1	—	—
7	23	161	2	16	5	—	—
8	11	88	6	4	1	—	—
9	11	99	1	7	3	—	—
10	9	90	2	4	1	1	1
11	8	88	2	2	4	—	—
12	5	60	—	2	3	—	—
13	4	52	2	1	1	—	—
14	2	28	—	—	1	1	—
15	1	15	—	—	—	1	—
16	2	32	1	—	—	1	—
17	1	17	—	—	—	1	—
	245	1386	86	133	20	5	1 $\sum = 246$
times n:			86	266	60	20	5 $\sum = 437$

$$\frac{\sum_{N=1}^{17} N{\cdot}B(N)}{\sum_{N=1}^{17} B(N)} = \frac{1386}{246} = 5.66 \text{ Strokes/Av. Block; } \sigma = 3.16 \text{ Strokes}$$

$$\text{Av. Number of Strokes Groups per Building Block } B = \frac{437}{245} = 1.78$$

We can only speculate on the reasons why the 10.32 strokes per Kanji were not allocated in a "square" manner, that is, as the square root of 10.32, or 3.21. Possibly, the reason for this is that the pictorial dictionary started with the more simple characters so that the strokes-per-group limit of about 4 was reached first. Then the number of groups per Kanji evolved to whatever it had to, but did not reach above about 2.52 for the small Japanese dictionary. The product, 4.22 × 2.52, meets the total requirement of 10.32 strokes per average Kanji for this dictionary of 1850 characters. But what about the larger dictionary of almost 10,000 characters?

The tiresome exercise of analyzing this larger dictionary in the direct fashion carried out with the smaller dictionary was avoided by analyzing the standard building block information provided in original form (Fig.

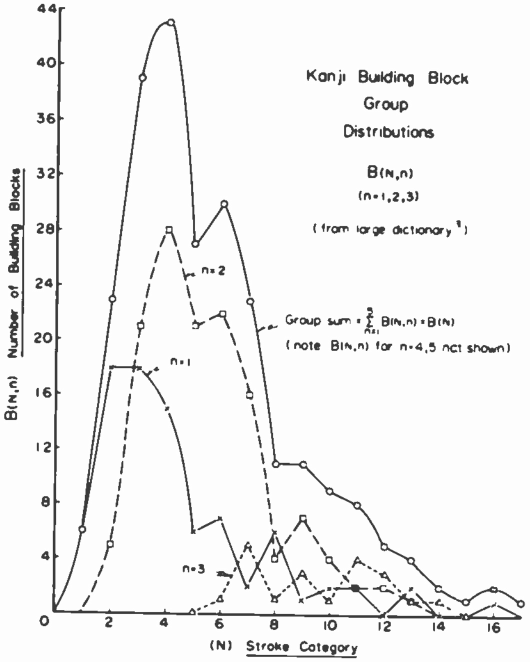


Fig. 9—Linear plot of Kanji building block frequency distribution $B(N,n)$ for n = 1, 2, 3.

8), and summarized for our purposes in Tables 5 and 6, and in Fig. 9. The curves (Fig. 9) with their dropoff beyond N = 4, seem to demonstrate an aversion to groupings in excess of 4. This is further substantiated by the allocation of the 5.66 strokes per average building block (Table 5) into the average number of groups per building block of 1.78 (Table 5), and the average number of strokes per building block group, 3.39 (Table 6). The product, 1.78 × 3.39, which equals 6.03, is well above the 5.66 figure, a peculiarity of the irregular building block frequency distributions (Fig. 9).

Although the well-known building blocks (Fig. 8) were listed only in the larger dictionary,[3] with some exceptions they also apply to the smaller one.[1] Furthermore, this list is only a compilation of building blocks available for building Kanji; it contains no information about the frequency or preferences in actual use. Available building blocks have 5.66 strokes on the average. The blocks actually used are smaller; we found they have 4.22 strokes on the average, as noted previously. In other words, there is a usage preference for building blocks having fewer strokes than 5.66.

The current Chinese dictionary[4] was not analyzed in detail. However, the intimate connection between Chinese and Japanese characters, and

Table 6—Building Block Group Stroke Distribution and Results. Shows N and n Distribution of Strokes $N{\cdot}B(N,n)$ With Average Number of Strokes for All Groups.

Stroke category (N)	Building blocks $B(N)$	Total strokes $N{\cdot}B(N)$	$N{\cdot}B(N,n)$				
			$n = 1$	2	3	4	5
1	6	6	6	—	—	—	—
2	23	46	36	10	—	—	—
3	39	117	54	63	—	—	—
4	43	172	60	112	—	—	—
5	27	135	30	105	—	—	—
6	30	180	42	132	6	—	—
7	23	161	14	112	35	—	—
8	11	88	48	32	8	—	—
9	11	99	9	63	27	—	—
10	9	90	20	40	10	10	10
11	8	88	22	22	44	—	—
12	5	60	—	24	36	—	—
13	4	52	26	13	13	—	—
14	2	28	—	—	14	14	—
15	1	15	—	—	—	15	—
16	2	32	16	—	—	16	—
17	1	17	—	—	—	17	—
	245	1386 $\sum = 1386$	383	728	193	72	10
$\frac{1}{n}\sum N{\cdot}B(N,n)$			383	364	64.33	18 $\sum =$ 831.33	2

Av. Strokes, $\dfrac{\sum N{\cdot}B(N,n)}{n\sum B(N,n)}$: for $n = 1$, $\dfrac{383}{86} = 4.45$; for $n = 2$, $\dfrac{364}{133} = 2.74$; for $n = 3$, $\dfrac{64.33}{20} = 3.22$; for $n = 4$, $\dfrac{18}{5} = 3.6$; for $n = 5$, $\dfrac{2}{1} = 2.0$.

Av. Strokes for All Groups: $\dfrac{831.33}{245} = 3.39$.

the generally analogous shapes of the distribution curves (Fig. 5), strongly suggest similar behavior with respect to the preceding results on stroke and group behavior.

We conclude that there exists a strong visual preference for stroke groupings not exceeding about 4.

Kanji Frequency Distribution

The linear scale comparison in Fig. 10, and the log-log scale comparison in Fig. 11, suggest that the Kanji distribution function in the abbreviated Japanese dictionary is quite close to the normal form. The distribution function of the Kanji in the other two dictionaries is more skewed, and

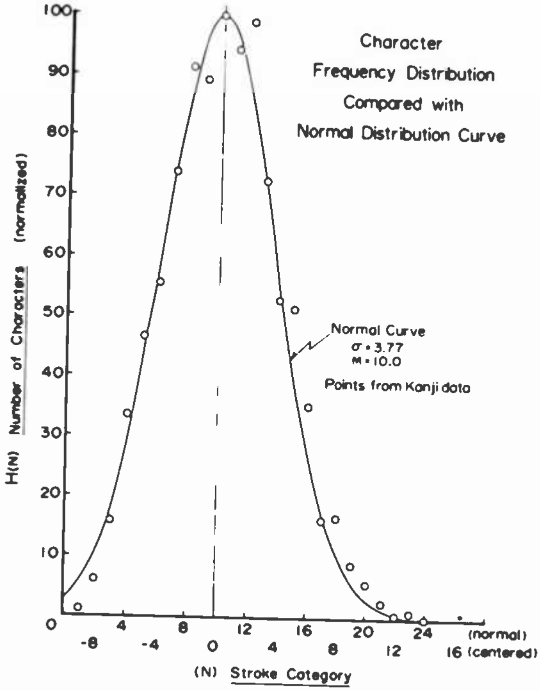


Fig. 10—Linear plot of Kanji character frequency distribution compared with the theoretical normal distribution curve having a medium value at $N = 10$, and a dispersion, $\sigma = 3.77$ strokes.

a cursory attempt at curve fitting showed that the agreement was not quite as good. However, symmetry about the peaks of all the distributions was surprisingly good in the middle stroke range when local irregularities are neglected.

The relatively good match to the normal distribution, or perhaps more properly to the binomial distribution because of the discrete nature of the curve data, suggests that the number of Kanji $H(N)$ at a particular value of N can be represented in terms of the binomial coefficient[5]

$$H(N) \propto \begin{bmatrix} N' \\ N \end{bmatrix} p^{N'} q^{N'-N}. \qquad [9]$$

Here N' is the approximate largest value of N in the $H(N)$ distribution (in this case about 20 strokes), N is the stroke category, p is the success probability of a stroke composing a dictionary character, and $q = 1 - p$ is the probability of failure. Viewed physically, N' is the total number of available cells for strokes to fit into, and Eq. [9] gives the relative number of successful ways that N strokes can be fitted into the N' cells to provide a visually acceptable combination.[6]

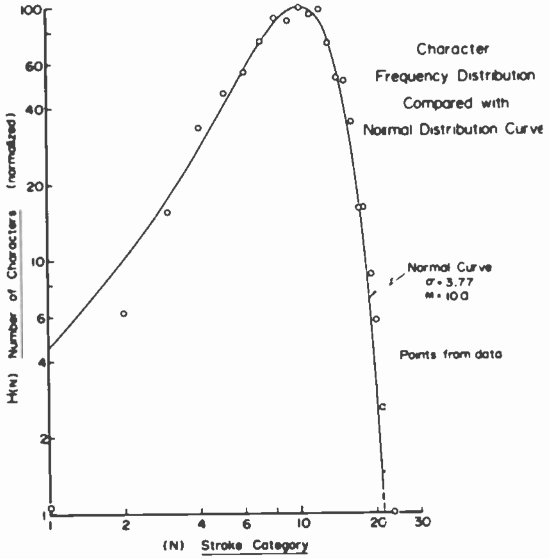


Fig. 11—Log-log plot of Fig. 10.

As is well known, the form of Eq. [9] is bell-shaped with a peak at $N = N'/2$ for the case where $p = q = 0.5$. Perhaps the most familiar practical example of such behavior is the case of a head-tail symmetrical coin. Here N' would be the total number of tosses, and $N = 0, 1, 2, 3$, etc., applied in Eq. [9] would give the relative probability of observing 0, 1, 2, 3, . . . N heads.

More detailed versions of the cell occupancy concept seem quite possible, but we have not pursued them.

Other Results

The average number of strokes per Kanji in the 1850-character dictionary is 10.32. It is interesting to compare this with the number of "strokes" in the average word in a small English–Japanese and Japanese–English dictionary[7] containing about 2000 English words in the first section. In this section the average English word length is about 5.7 letters as determined by a sample from each of the 122 pages. English block letters average about 2.5 strokes if counted in a manner analogous to that used with the Kanji. Accordingly, the average English word in this size of dictionary has ~5.7 × 2.5 = 14.3 strokes, a value not much different than the 10.32 value for the Kanji dictionary calculated in Table 2.

With larger English dictionaries the average word length increases to somewhat over 6 letters bringing the average number of strokes to about 15. But an increase also occurs with the larger Japanese dictionary[3] where the average number of strokes per Kanji rises to about 12. Again, there is approximate equivalence.

The above suggests that "strokes" are a meaningful parameter to use in evaluating written languages and, at least as a crude first approximation, the Chinese-Japanese and English languages require about the same number of visual information "bits" to express a thought. We are not aware of any previous mention of this interesting possibility in the literature, at least with respect to the Japanese language. However, textual comparisons, such as discussed below, may be more valid than dictionary-like ones because Kanji tend to be used phoneme-wise in pairs or larger groupings to express a concept, particularly in the Chinese written language. This tendency in Japanese is reduced by the use of an additional set of symbols, Kana.

The validity of using strokes as an analytical parameter is indeed corroborated more closely by character usage in actual text, as in the case of the Japanese language. Data on this was kindly supplied privately by W. English of the Xerox Corporation and is shown in Table 7. We normalized his percentage results to English character usage, per 100 characters, and made the stroke comparisons. Note that when the Japanese text moves to higher Kanji content, as in the newspaper text, the total number of strokes more closely approximates those used in English. Indeed, this equivalence is better than with the dictionary-like word comparison made above. Note also that the purely phonetic characters, Kana, have about the same number of strokes, 2.3, as the English pho-

Table 7—Comparison of Stroke Count in English and Japanese Texts.

	Semi-technical business text			Newspaper text		
Language	Characters[1]	Strokes per character[2]	Total strokes[2]	Characters[1]	Strokes per character[2]	Total strokes[2]
English	100	2.5	250	100	2.5	250
Japanese						
Kanji	21	10.3	216	21	10.3	216
Kana[3]	49	2.3	113	26	2.3	60
	70		329	47		276

[1] Information kindly supplied by W. English of Xerox Corp.

[2] Calculation made by authors.

[3] The two Japanese phonetic alphabets, Katakana and Hiragana. These perform foreign word translation and grammatical functions in text. Kana were derived from Kanji characters and in the Katakana form consist of very simple stroke configurations such as マ, ヤ, and レ; in the cursive form, Hiragana, the characters would be ま, や, and れ. The strokes per character of these were derived solely from the Katakana component since the strokes here are not of the cursive type, as in the Hiragana, and can counted more easily. Accordingly, the 2.3 value is an approximation for Kana that is probably quite accurate since the average stroke count in Hiragana differs very little from that in the Katakana. On the other hand, English cursive characters have significantly fewer strokes than the block letters: approximately 2 strokes per letter.

netic characters. This is subject to the qualification footnoted in Table 7. The third and most important item of note is that in both types of text, regardless of relative Kanji content, both written languages have about the same stroke count. This comparison (Table 7) is neither intended to be precise nor a measure of the relative effectiveness of these very different languages to convey written information under varying circumstances. However, it is quite suggestive. To make a more precise comparison, account would have to be taken of language translation efficiency.

Inspection of ancient and modern Mediterranean phonetic alphabets listed in the Encyclopedia Britannica shows very few letters with more than four strokes. The average is much less than this, roughly 2.5, just as in modern English and Japanese Kana. Indeed, a phonetic alphabet seems characterized by an average of approximately 2.5 visual stroke elements per letter character. Although the pictorial type "alphabet" is significantly different in terms of having more strokes per average character, the total number of strokes per average mental concept is nevertheless about the same as for the phonetic method. This situation can be viewed in terms of building blocks in a manner analogous to that discussed previously with respect to the Kanji characters. That is, the phonetic characters are, themselves, small standardized visual building blocks. But instead of being assembled a few at a time inside a character "square" as in Kanji, they are assembled linearly. Each linear grouping, a word, performs roughly the same concept function as a Kanji. Also, just as in Kanji, building blocks elements can appear in English words as easily recognizable groupings, for example, as the familiar prefixes and suffices, and as the visual cadences between lower case letters having long stems such as "b," "d," "g," and "p".[8]

Discussion

We are tempted to speculate about the four-fold grouping tendency revealed in these studies. The human tendency for grouping information is well-known and very basic, but why should it be four-fold?

Possibly there is a clue in the personal observation that one can quickly recognize, without consciously counting or subgrouping, one, two, three, or four simple items such as short lines or outstretched fingers. When a fifth item appears, one consciously starts to count, or subgroup (4 + 1, 2 + 3, or 3 + 2), and then add together to determine the total. It is probably not a coincidence that money denominations are set up the way they are to facilitate quick recognition and handling. We are not aware of any money denomination system wherein the intervals are larger than five. That is, the next higher denomination never occurs later than after four units of the lower denomination. This is also true in the abacus, the ancient calculating device. Phone numbers and license plates numbers are grouped such that the largest group contains no more than four digits. The earlier days of motoring in the U.S. featured license numbers with their digits all in one group, a grouping that often appreciably exceeded four. It was something of a novelty when the four-fold grouping appeared for the convenience of the police in identifying speeders.

Observation of a Kanji dictionary, portions of which were shown (Figs. 1, 3, 4, and 8), makes it quite obvious that four directions of lines dominate: horizontal, vertical, and two approximately orthogonal diagonals. These seem to be the most perceptually distinguishable set of directions that fit into the four-sided character space. The addition of only one more line direction would considerably increase the probability of error in distinguishability.[9] Possibly there is some connection between these four directions and the preferred four-fold grouping of strokes.

At this time we can only speculate on the connection between the "7 ± 2" rule[2] and our observation of the average tendency to group in fours, and to avoid larger groupings if at all possible. While 7 ± 2 is a limiting case for information channel capacity, the 4-grouping seems a strong practical preference. As noted by others,[2] up to four items can be recollected and handled with virtually no errors; beyond four, errors increase rapidly. In addition, as the number of items to perceive increases, the number of items in the mnemonic subgroups increase. For example, for a total of five items one can group as 3 + 2; for six items as 3 + 3; for seven as 4 + 3, 3 + 3 + 1, 3 + 2 + 2, 2 + 2 + 2 + 1, or 5 + 2; and so on for even larger numbers of items to be handled. If we now assume that a minimum number of groups or elements in a group is best, and that the largest desirable group, or grouping of groups, is four-fold, then we are led to an upper maximum of 4 + 4 + 4 + 4 = 16. But empirical evidence supports the 7 ± 2 rule as an upper limit. Accordingly, in terms of all the above restrictions we have 4 + 4, 2 + 2 + 2 + 2, and 3 + 3 + 3 as the limiting symmetrical cases that can exist within the 7 ± 2 rule limit. None violate the 4-condition, and for each combination, symmetry minimizes the number of items in each group. However, we feel that 4 + 4 is preferable to either 2 + 2 + 2 + 2 or 3 + 3 + 3 because fewer conscious steps, and time intervals, are involved in which to forget or to make errors. The average number of stroke groups per Kanji, 2.5 (Table 3), may have some relation to this. Also, additional groups would require more interstitial space within a character. Four groups of four strokes produce Kanji that are on the descending portion of the frequency distribution curve, three groups of four produce Kanji at the peak of the curve, and one or two groups of four strokes produce Kanji on the rising part of the curve.

Many other examples of the tendency for four-fold groupings can be given for the case when the total number of items exceed, or could exceed, four in number. This tendency is apparently not restricted to the visual process.[2] For example, there is a four-fold duple structure in classical music.[10] Moreover, standard harmonic chords in classical music seldom have more than 4 different notes. When 5-chord notes are found, one is usually melodic and not harmonic, as in the ninth chord. A symphony orchestra is four-fold organized: strings, brass, woodwinds, and percussion. Inspection of classical music history and present-day record catalogs reveals a noticeable preference for quartet music compared to that for duos, trios, and especially for instrument groupings larger than four. Piano and orchestral groupings were excluded in this observation. With respect to taste, a biochemist usually speaks of four parameters—sour, bitter, salty, and sweet.[11]

Does this grouping tendency have something to do with brain organization, a simple reflection of how our fingers are grouped, or some combination of both?

No attempt was made to relate our results to the "Rank-Size Law" of G. K. Zipf[12] nor the "Fractals" of B. B. Mandelbrot.[13]

Conclusions

The ancient Chinese-Japanese pictorial language characters seem to be an unusually good medium for studying the behavior of the human visual system. These characters are well documented and standardized, and are constructed with standardized line strokes particularly convenient for use as an analytical parameter. Indeed, one might view a stroke as a "bit" of visual information. Plotting a dictionary of characters as a function of the number of strokes in a character reveals a peaked, fairly symmetrical, frequency distribution with interesting structure. This peculiar structure arises from the manner of stroke grouping in the characters. The average stroke group contains approximately four strokes, an interesting phenomenon that seems to have its counterpart in other more familiar examples of how humans organize information.

The character frequency distribution approximates a normal distribution suggesting that a character can be viewed in terms of a cell-stroke occupancy model. Visually acceptable characters increase in number as the number of strokes increase. But as the character cell space becomes increasingly filled with strokes, the strokes become grouped in various configurations to maintain visual distinguishability and acceptability. Nevertheless, the frequency distribution reaches a maximum value at about 11 strokes, then decreases rapidly as the stroke cell space gets used up.

The average character has about 11 strokes, a value approximately the same as the number of block letter strokes in the average English word. This equivalence holds in ordinary text as well as in dictionary words, and suggests that strokes may indeed be a useful parameter in evaluating visual information.

We have been surprised at the apparent dearth of literature references to the type of study described above. Aside from purely scientific interest, the results may have applicability to information display technology and human interface design of instruments and equipment.

Acknowledgments

We are grateful for the many useful suggestions made by C. R. Carlson and R. Williams at the Princeton Laboratories, and appreciate the assistance of W. Chu, librarian, in providing us with numerous references. Helpful discussions with H. Martin, formerly associated with the Music Department at Princeton University, are a pleasure to acknowledge.

References and Notes:

[1] F. Sakade et al., *A Guide to Reading and Writing Japanese*, 2nd ed. rev. Charles Tuttle, Rutland, Vermont and Tokyo (1961).

[2] G. A. Miller, "Information and Memory," *Scientific American*, Aug. 1956; G. A. Miller, "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information," *Psych. Rev.*, **63**, No. 2, p. 81, March 1956; K. Anders Ericsson, et al., "Acquisition of a Memory Skill," *Science*, **208** p. 1181, 6 June 1980.

[3] T. Ogawa et al., *Kadogawa Shinjigen*, Kadogawa, Tokyo (1968).

[4] *Current Chinese-English Dictionary*, Cosmos Books Ltd., Hongkong (1978). Called to author's attention by R. Williams, Princeton.

[5] W. Feller, *An Introduction to Probability Theory and Its Applications*, Vol. 1, John Wiley, New York (1950), Chapman and Hall, London.

[6] The symmetry of the Kanji distribution curve and its description in terms of the cell-stroke occupancy model suggests that unoccupied spaces in a Kanji are as important in visual recognition as the stroke-filled space. Suitably configured, unoccupied space must be retained between stroke groups as well as between the strokes inside a stroke group. The configuration of unoccupied space between groups would seem to serve the same function in visual recognition as do the unoccupied spaces surrounding words in English. In the Orient, the importance of unoccupied space in calligraphy and painting has long been overtly recognized.

[7] S. E. Martin, *Basic Japanese Conversation Dictionary*, Charles E. Tuttle, Rutland, Vermont and Tokyo, forty-third printing (1979).

[8] See, for example, P. Dunn-Rankin, "The Visual Characteristics of Words," *238* No. 1, p. 122 January 1978 and P. A. Kolers, "Experiments in Reading," *Scientific Amer.*, **227**, p. 84, July 1972.

[9] D. H. Hubel and T. N. Wiesel, "Brain Mechanisms of Vision," *Scientific Amer.*, p. 130, Sept. 1979. This article presents evidence that significant groups of visual cortex brain cells respond to specifically oriented line segments, rather than spots of light. Furthermore, response resolution is limited to 10° to 20° clockwise or counterclockwise line rotation. As if in recognition of this, the noncursive letters in most alphabets, old or modern, largely consist of line segments that are clearly recognizable as vertical, horizontal, or diagonal. Here a closed circle might be considered as a rounded square.

[10] Private communication from H. Martin, professional musician, and researcher in field of musical structure.

[11] F. R. Dastoli, "Taste," *Science and Technology*, p. 42, March 1968.

[12] G. K. Zipf, *Human Behavior and the Principle of Least Effort*, Addison-Wesley (1949). Also, G. K. Zipf, *The Psycho-biology of Language*, Houghton-Mifflin (1935).

[13] B. B. Mandelbrot, *Fractals: Form, Chance, and Dimension*, W. H. Freeman Co., San Francisco (1977).

Maximum Entropy Modeling for Identification and Detection of Certain Classes of Dynamic Events

N. A. Macina

RCA Government Communications Systems, Somerville, NJ 08876

Abstract—As a result of their continuous refinement, time domain signal processing techniques deserve consideration as potentially more effective tools than their more popular frequency domain (FFT) counterparts when used in classification, identification, and detection of certain classes of dynamic events and targets. This paper discusses the investigation of the maximum entropy modeling technique as applied to a specific problem of dynamic target identification and detection, for which actual recordings of a seismic sensor were available. A heuristic introduction of the maximum entropy method and its relation to linear prediction analysis is presented, for the benefit of the reader not too familiar with the technique.

From the interpretation of the data base, an algorithm outline is derived that uses time domain features only, which are shown to be obtainable from the waveform model. Considerations of computational requirements together with an estimate of an 1802 microprocessor implementation of the recursive subroutine conclude the paper.

1. Introduction

Many problems of target classification and detection are today still approached as pattern recognition problems leading to the development of classifiers from target features most frequently extracted in the frequency domain (FFT). Since the observed events are usually nonstationary, the resultant algorithms threaten to become complex and cumbersome both in storage and computation requirements.

Rather powerful, new time-domain techniques have been developed and perfected over the past decade, which—if properly understood and applied—can provide elegant and compact solutions for certain classes of targets and problems. These techniques go under a variety of names, the most common of which are linear prediction and maximum entropy modeling or filtering; they represent slightly different embodiment of the basic whitening filter (Wiener) theory, according to which for any waveform a matched filter can be derived from the second order statistics (i.e., autocorrelation or power spectrum) of that waveform.

This paper describes the investigation of one of these modeling techniques when applied to a specific problem of event identification and detection related to a certain class of targets. Actual recordings of typical events were available and used in the investigation. In Sec. 2 the problem is described and some design considerations are derived from the analysis of recorded data. A brief review of the maximum entropy modeling technique is introduced in Sec. 3. The paper concludes with a section on the application of the time domain modeling technique, with considerations on computational requirements. The investigation reported here was purely exploratory, with no intent to optimize an algorithm for any preset performance objectives at this time. The purpose of this paper, therefore, is to disseminate these new ideas and stimulate interest in these potentially very powerful time-domain signal processing techniques.

2. The Problem and the Data Analysis

2.1 The Problem

It is desired to monitor the traffic of jet-aircraft on an active runway, as they go by a reference point, where a geophone is emplanted, within, say, 100 ft. from the runway. What is wanted then is an identifier/detector that will

(1) discriminate between targets (take-off, taxiing, landing) and non-targets (overflight and nuisances, e.g., firetrucks)
(2) detect the crossing of the closest point of approach (CPA) by a target,
(3) ignore the traffic on another runway located more than 200 ft. away from the sensor, on the other side relative to the runway under observation (range containment).

2.2 Data Analysis

The analysis of the recorded data showed rather early that the traces of the various events featured certain distinctive patterns amenable to intuitive physical interpretation:

(1) a vehicle rolling on the ground generates a relatively strong low frequency (10–50 Hz) coherent component,

(2) the aircraft jet-engines generate a very broadband colored acoustic noise of sufficiently high intensity to be picked up by the geophone;

(3) the jet noise is prevalently directed toward the back of an aircraft so that as it crosses CPA a sudden increase in noise level occurs in the seismic signature,

(4) since the seismic component attenuates through the ground more rapidly than the acoustic contribution through the air, the ratio of the former (signal) to the latter (noise) deteriorates as the distance of the source from the sensor increases.

These physical events affecting the sensor traces can be correlated with the actual events on which the identifier/detector must operate, as illustrated in Fig. 1. The figure shows three representative geophone time records and (at the bottom) the trace of a time marker identifying CPA. The first time series is typical of a target. It shows a rather strong low frequency component that, after CPA, is suddenly submerged by

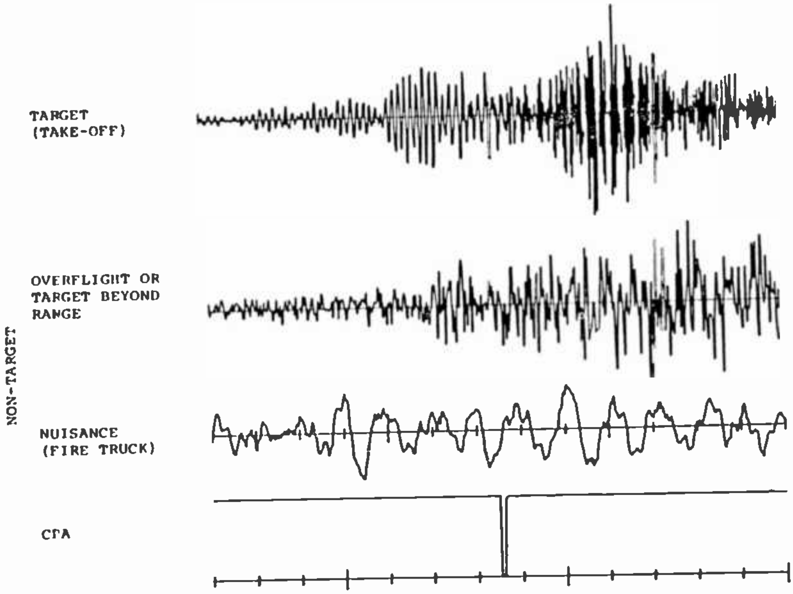


Fig. 1—Representative 0–100 Hz seismic traces around closest point of approach (CPA).

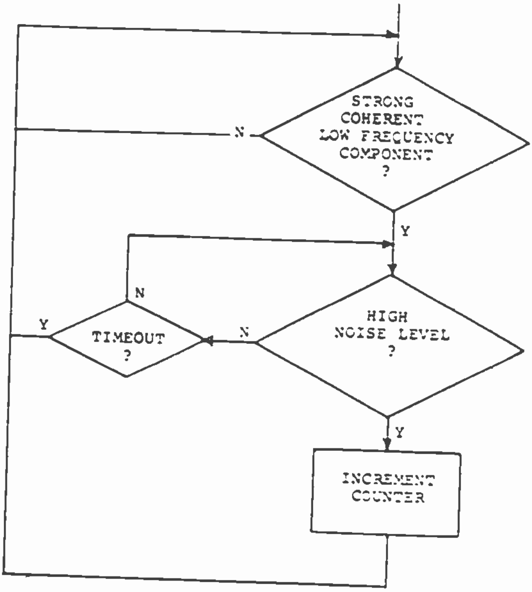


Fig. 2—Algorithm outline.

a high noise level. The second trace is representative of two classes of nontargets, namely, (1) overflights and (2) targets on a distant runway. The acoustic jet noise prevails in either case. The third time record is representative of a firetruck signature. It shows strong, coherent, low-frequency energy, that remains unchanged as the truck goes past CPA.

It seems obvious then that for this problem the solution can be found by just observing and interpreting the various event signatures strictly in the time domain. The algorithm for the identifier/detector can be outlined as follows (Fig. 2):

(1) Look for a coherent low frequency component in the geophone output; when its presence is detected (which means "target" or "firetruck"), go to (2);
(2) Look for a burst of noise; if detected (which means target), step traffic counter and go to (1), else go to (3);
(3) Look for time out; when detected, reset to (1), else go to (2).

One can obviously improve on this algorithm by taking into account additional time features, such as the average signal amplitude; a steady increase in amplitude then may warn of a potential approaching target; the rate of increase may be used to discriminate, if required, between take-off and taxiing; and, finally, the occurrence of the amplitude peak could be used as a reference for the time-out delay. We shall, however, concentrate on the main time feature, namely the degree of correlation of the low-frequency component of the geophone trace.

2.3 The Main Time Feature

How does one monitor the level of low-frequency correlation of a waveform? One way is to measure the correlation between one waveform sample and its immediate neighbor. With a strong low-frequency component the correlation between adjacent samples is close to one; as the noise level increases or the frequency of the major component shifts upward toward the Nyquist frequency, the correlation drops to zero or

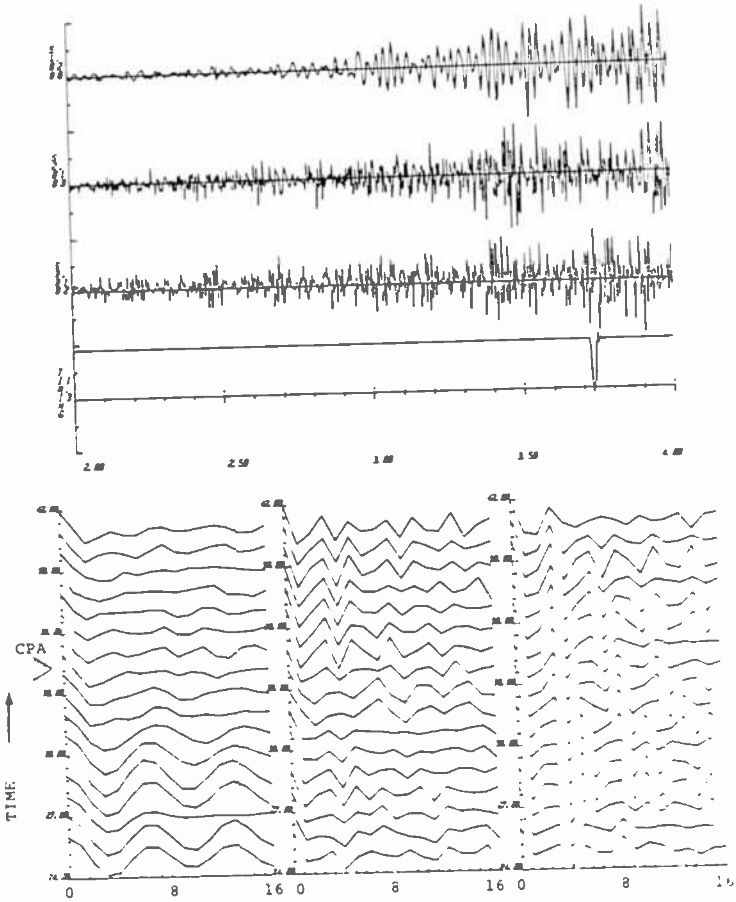


Fig. 3—Traces and autocorrelations for take-off at 100, 200, and 300 feet.

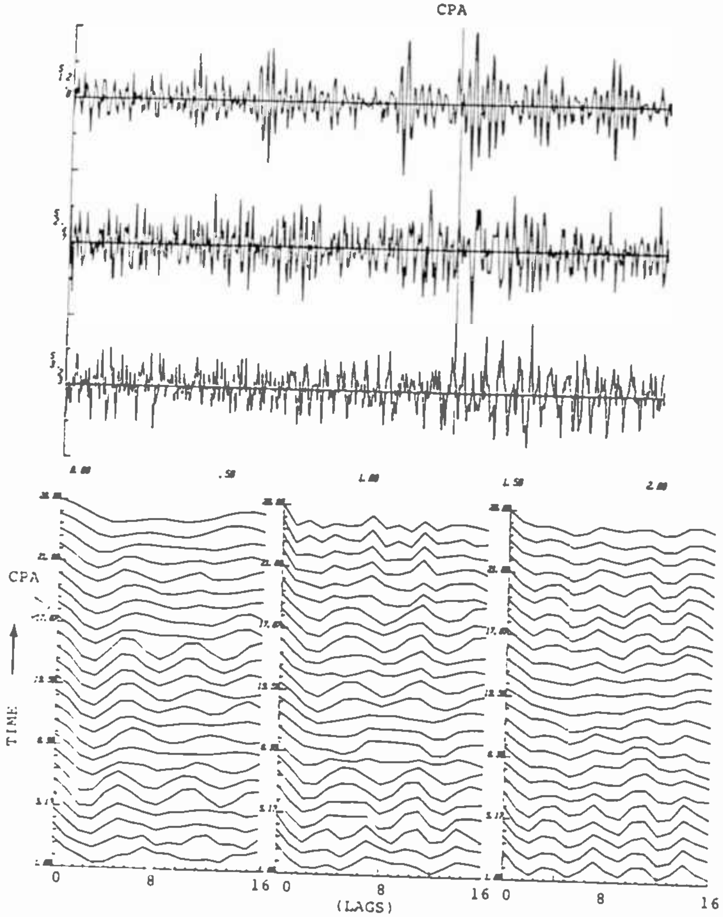


Fig. 4—Traces and autocorrelations for fire truck at 100, 200, and 300 feet.

down further to negative values. A simple enough solution, therefore, may be to just compute periodically the first auto-correlation (first lag) coefficient of the sensor output, and compare it to a threshold of, say, 0.5. An auto-correlation coefficient greater than 0.5 would indicate the presence of a strong low-frequency component.

The realization of the extractor of such a feature can be equally simple. It requires the computation of two sample auto-correlation values: at zero lag (power) R_0 and at the first lag R_1. These can be computed over adjacent or overlapped data windows, or more simply recursively.[1] The normalizing division R_1/R_0 can be avoided if a threshold of 0.5 is adopted; one just compares R_1 to $R_0/2$ (right shift).

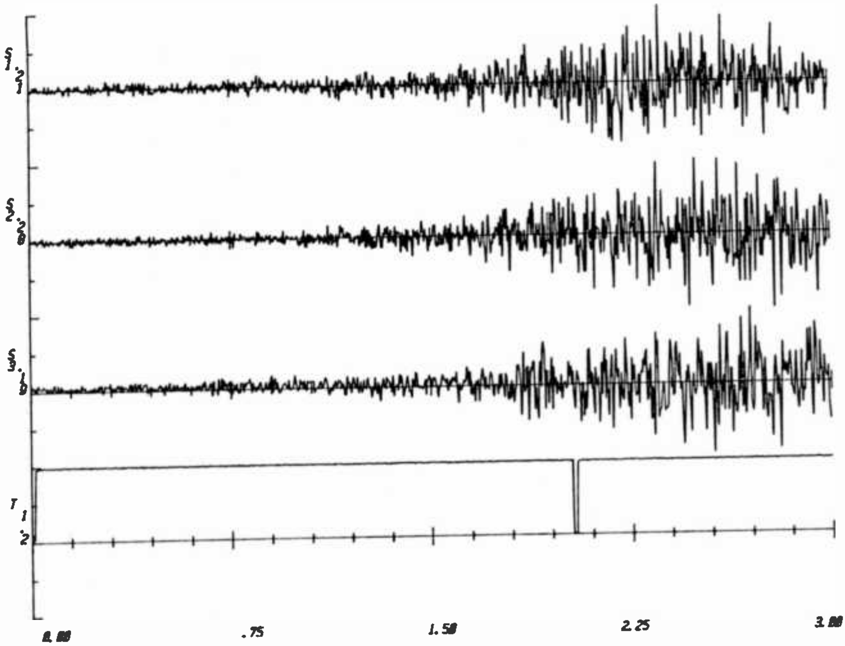


Fig. 5—Traces of an overflight at 100, 200, and 300 feet.

An illustration of the behavior of the first correlation coefficient for an aircraft taking off is given in Fig. 3. At the top are the traces of 3 geophones, 100, 200, and 300 ft. from the runway, and the CPA time marker; in the lower half of the figure are plots of successive sample auto-correlation (from lag 0 to 16) functions versus time (ordinates) for the three sensor traces. Note how the first-lag coefficient drops after CPA for the 100-ft. trace, but stays negative before and after CPA for the sensors further away from the runway.

Fig. 4 illustrates the case of a firetruck. The value of the first-lag coefficient at 100 feet does not change as the truck goes past the CPA. The three traces resulting from an overflight are shown in Fig. 5. They are so obviously noisy that their auto-correlation functions were not derived.

The extreme simplicity of this solution may make the reader somewhat uncomfortable, since it does not seem to provide either sufficient visibility on what is going on or enough parameters to control (data window size and threshold, only) for the subsequent process of algorithm optimization on the basis of the data base statistics and performance objectives. A more satisfying approach would be to monitor and track the full waveform rather than just the correlation between two adjacent samples. One can do this by generating a sequential model of the waveform (matched filter), which we subtract from the waveform itself, and observe the filter coefficients and the subtraction output (error). The higher the coherence of the signal, the better the match, the smaller the error; the higher the noise, the poorer the match, the higher the error. The filter coefficients contain information about the spectrum of the signal.

The techniques available to perform this sequential waveform modeling as well as the specific choice made for a representative algorithm are discussed in next section.

3. The Maximum Entropy Modeling Technique

3.1 The Maximum Entropy Method

The maximum entropy method (MEM), popularly attributed to Burg, was originally developed as a special procedure for spectral estimation, to circumvent the limitations of the direct method (FFT) and other indirect methods (MEM is one of them) based on the computation of the auto-correlation function of the observed waveform, first. Those limitations arise from the fact that in practice these second-order statistics (power spectrum or autocorrelation) have to be computed from a time-limited observation of the waveform. This requires that certain assumptions be made as to the behavior of the waveform outside the observation interval, such as to be identically zero (windows) or periodic (FFT). These assumptions would introduce waveform distortions (weighting windows) and leakages.

With the support of an analysis based partially on information theory the MEM claimed that its spectral estimate, while consistent with the known information about the waveform, expressed maximum uncertainty (entropy) with respect to the unknown information. In other words, with the MEM there was no assumption to be made as to waveform behavior outside the observed interval. It turned out, however, that the MEM spectral analysis is equivalent to fitting an all-pole (autoregressive, or AR) model to the random process under observation; i.e., the MEM and the all-pole linear prediction method are equivalent.

We can only present a brief heuristic introduction to linear prediction, while recommending to the interested reader familiar with linear sampled systems and z-transforms two very useful references: (1) the classic tutorial by Makhoul,[2] and (2) chapter 8 of the Rabiner and Shafer book on speech processing.[3] For MEM, the not too easily available Ref. [4] is recommended.

3.2 Autoregressive (AR) Processes and Linear Prediction

Assume that a waveform containing some coherent energy and some noise is characterized by the fact that any sample $s(n)$ can be approximated, except for the noisy component $e(n)$, by a linear combination of its p preceding samples:

$$s(n) = a_1 s(n-1) + a_2 s(n-2) + \ldots + a_p s(n-p) + e(n). \quad [1]$$

This difference equation reflects an AR-system (Fig. 6, $p = 4$); namely, $s(n)$ is the output of a p-pole filter when $e(n)$ is the input. The transfer function of the filter is

$$H(z) = \frac{1}{1 - \sum_p a_i z^{-i}}, \quad [2]$$

and the set of coefficients $\{a_i\}$ are adequate to model the waveform.

Now call

$$\hat{s}(n) = \sum_p a_i s(n-i); \quad [3]$$

then

$$s(n) - \hat{s}(n) = e(n). \quad [4]$$

In words, $\hat{s}(n)$ is the estimate or predicted value of $s(n)$ from its previous p samples, while $e(n)$ is the error of the estimate. From Eq. [4] we obtain the linear predictor realization of Fig. 7.

Note also that from Eq. [4], $e(n)$ can be regarded as the output of a filter with $s(n)$ as input and with transfer function

$$A(z) = 1 - \sum_p a_i z^{-i}.$$

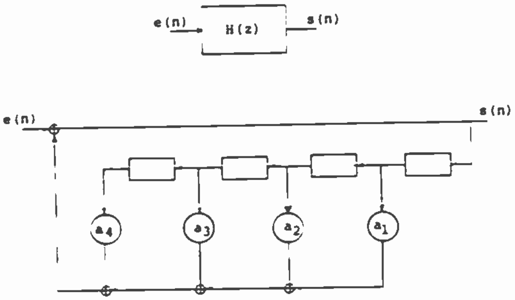


Fig. 6—Autoregressive system realization.

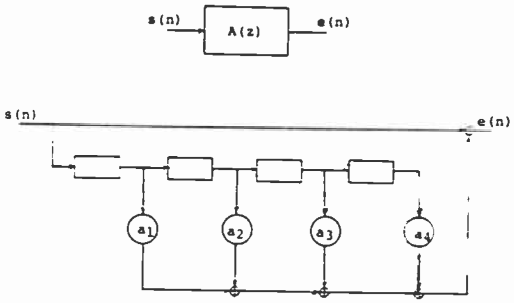


Fig. 7—Linear predictor realization.

This filter is called the inverse filter, since

$$H(z) = 1/A(z).$$

Hence, if a waveform can be approximated by an AR-process, one can model it by finding the set of coefficients $\{a_i\}$. Indeed, knowing the filter coefficients, one can synthesize a realization of the process as in Fig. 6 (synthesis).

On the other hand, given a realization of an AR-process, one can find the modeling coefficients as the set of $\{a_i\}$ that minimize the error $e(n)$ in Eq. [4] in the mean square sense (analysis). This operation of error minimization leads to a set of p linear equations, called normal equations,

$$\sum_{k=1}^{p} a_k R(k-i) = R(i), \quad 1 \leq i \leq p, \qquad [5]$$

where $R(i)$ is the autocorrelation value of the AR-process at lag i.

There are various algorithms available for the solution of the normal equations, once the $p + 1$ values of the autocorrelation functions are estimated from a finite segment of the waveform. Most of the difficulties arise in carrying out this estimate so as to guarantee accuracy, stability, and well conditioning of the autocorrelation matrix in Eq. [5]. The problem is further complicated if the algorithm is to be realized in fixed-point arithmetic, since the coefficients a_i can be greater than unity.

Two computational methods are available: (1) the autocorrelation method guarantees stability but requires a window, which takes computation times, degrades resolution, and may distort the waveform and (2) the covariance method, which does not require data windowing but does not guarantee stability.

Once the linear prediction coefficients $\{a_i\}$ are found, the power spectrum distribution of the waveform $S(\omega)$ can be obtained from Eq. [2] as (ω is the angular frequency)

$$S(\omega) = \frac{E}{|1 - \sum_{p} a_i e^{-j\omega i}|^2}. \qquad [6]$$

The gain E is the error power or 'residual,' which can be viewed as an index of how well the waveform is modeled or matched by our linear predictor. It is the mean power of the output in Fig. 7 and can be obtained as

$$E = R(0) - \sum_{p} a_i R(i). \qquad [7]$$

One can then use these techniques as an indirect method of computing the power spectrum of an observed time series, or as a direct way to model the latter through its predictor coefficients and its residual.

The computational methods outlined thus far may be categorized as block processing. An N-point segment of the observed time series is taken, the autocorrelation values are computed, and the normal equations solved.

The coefficients can also be generated recursively,[5] taking one sample at a time through an adaptive version of the filter of Fig. 7, the coefficients of which are forced to provide the minimum error output. In contrast with the block method, where the reference (autocorrelation) is smoothed over N samples, here the reference is noisy, thus requiring more care in the design of the computation to insure convergence of the filter.

3.3 Back to the Maximum Entropy

It has already been said that the MEM and linear prediction, as indirect methods of spectral estimation, provide comparable results. MEM, however, still retains its identity as a modeling method for a combination of significant algorithm improvements, novel at the time Burg (1967–1968) developed his approach.

Burg's method does not require prior estimate of the autocorrelation function. The AR coefficients are generated recursively, first for 1-pole, then for 2-pole, and so on up to the p-pole solution. At each step the prediction error power (to be minimized) is calculated by running the prediction error filter over the data in a forward and backward direction. This is a significant difference from the autocorrelation method in which the error power is calculated by running the prediction filter over the

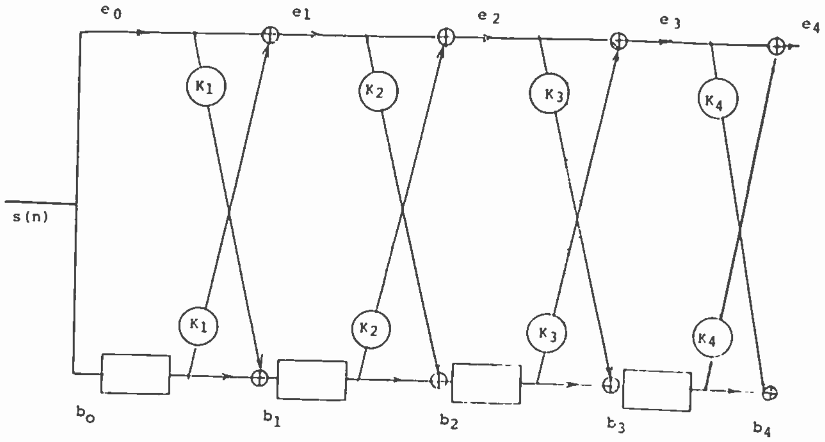


Fig. 8—Lattice inverse filter.

auto-correlation function segment $R(-p)$ to $R(p)$ in the forward direction only.

The Appendix lists the FORTRAN program (BURGO) of a most recent version of the Burg algorithm as designed by Andersen (reference given with program) for minimum computation.* The algorithm equations are too cumbersome to reproduce here and add little insight as to the computation steps involved (Ref. [4] is recommended).

A better insight into this more powerful predictive process can be gained by replacing the transversal filter structure of Fig. 7 with the lattice structure of Fig. 8. It has p stages, one per pole. With the Burg algorithm, one starts with the first stage, ignoring all the others, and feeds into it successively the N sample. For each sample, $s(n)$, the errors are computed in the forward, $e(n)$, and backward, $b(n)$, direction.

$$\begin{aligned} e_1(n) &= s(n) - k_1 s(n-1) \\ b_1(n) &= s(n-1) - k_1 s(n). \end{aligned} \quad [8]$$

The coefficient k_1 is the value that minimizes the sum of the two mean-squared errors:

$$E_1 = \sum^{N} (e_1(n))^2 + (b_1(n))^2.$$

Setting the derivative to zero and solving for k_1,

* The number of multiplications is of the order of $5Np$.

$$k_1 = \frac{2\sum^{N} s(n)s(n-1)}{\sum (s(n))^2 + \sum (s(n-1))^2}. \qquad [9]$$

One then considers the first two stages and again runs the N samples through them and computes the two output errors:

$$\begin{aligned} e_2(n) &= e_1(n) - k_2 b_1(n-1) \\ b_2(n) &= b_1(n-1) - k_2 e_1(n). \end{aligned} \qquad [10]$$

Again, minimizing the sum of the mean-squared errors, one obtains

$$k_2 = \frac{2\sum^{N} e_1(n)b_1(n-1)}{\sum (e_1(n))^2 + \sum (b_1(n-1))^2}. \qquad [11]$$

The process then continues iteratively up to the final solution for p poles.

Substituting Eq. [8] in Eq. [10],

$$\begin{aligned} e_2(n) &= s(n) - k_1(1-k_2)s(n-1) - k_2 s(n-2) \\ b_2(n) &= s(n-2) - k_1(1-k_2)s(n-1) - k_2 s(n), \end{aligned} \qquad [12[$$

or

$$\begin{aligned} e_2(n) &= s(n) - a_1 s(n-1) - a_2 s(n-2) \\ b_2(n) &= s(n-2) - a_1 s(n-1) - a_2 s(n) \end{aligned} \qquad [13]$$

where $a_1 = k_1(l - k_2), a_2 = k_2$.

Note now the following:

(1) In spite of its appearance, the lattice filter indeed combines a forward (along the top) and a backward (along the bottom) predictor, as shown by Eqs. [12] and [13].

(2) The $\{k_i\}$ coefficients, except for the last stage at each iteration, are not the same as the $\{a_i\}$ coefficients of the transversal predictor, but the latter can be derived from the former iteratively.

(3) Eq. [11] for k_i reminds one of a normalized correlation coefficient between the forward and backward error; hence, its more usual name of PARCOR, for partial correlation coefficient; the $\{k_i\}$ are also called reflection coefficients, in analogy to a transmission line.

(4) It has been shown that $|k_i| \le 1$, which guarantees the stability of the filter, and easier coefficient handling in fixed-point arithmetic.

(5) In contrast with the transversal structure, which admits a global minimum (minimum error is achieved by computing an optimal coefficient set for the entire structure), the lattice structure decouples one stage from another, so that if an adaptive version is wanted, minimization at one stage can be achieved without affecting any other.

In summary, the MEM algorithm has circumvented so many of the limitations of the autocorrelation method, that it has become, with the alternative name of lattice method, an important and viable approach to the implementation of linear predictive analysis and modeling. Its main disadvantage relative to the autocorrelation and co-variance methods is a lower computational efficiency. The required number of multiplications are of the order of $N_p + p^3$ (covariance), $N(p + 1) + p^2$ (autocorrelation), and $5Np$ (lattice). Consequently, MEM and lattice method are in a sense synonymous. Still, the term MEM is widely used to stress the significant implications of the original method as against the transversal filter structure and its limitations, most commonly associated (although incorrectly) with the most general term of linear prediction.

We may conclude this review of MEM with two considerations:

(1) The lattice method can also be implemented as an adaptive filter, with the same advantages and disadvantages relative to its transversal counterpart as the block method just discussed, namely, greater stability, local minima, but more computation.

(2) Strictly speaking, MEM would imply all-pole modeling only. Since it is slowly dissolving into the more general lattice method, however, that limitation is being removed. Very active research is currently going on to extend these methods to a more general process with poles and zeros (the ARMA processes, for autoregressive-moving-average).

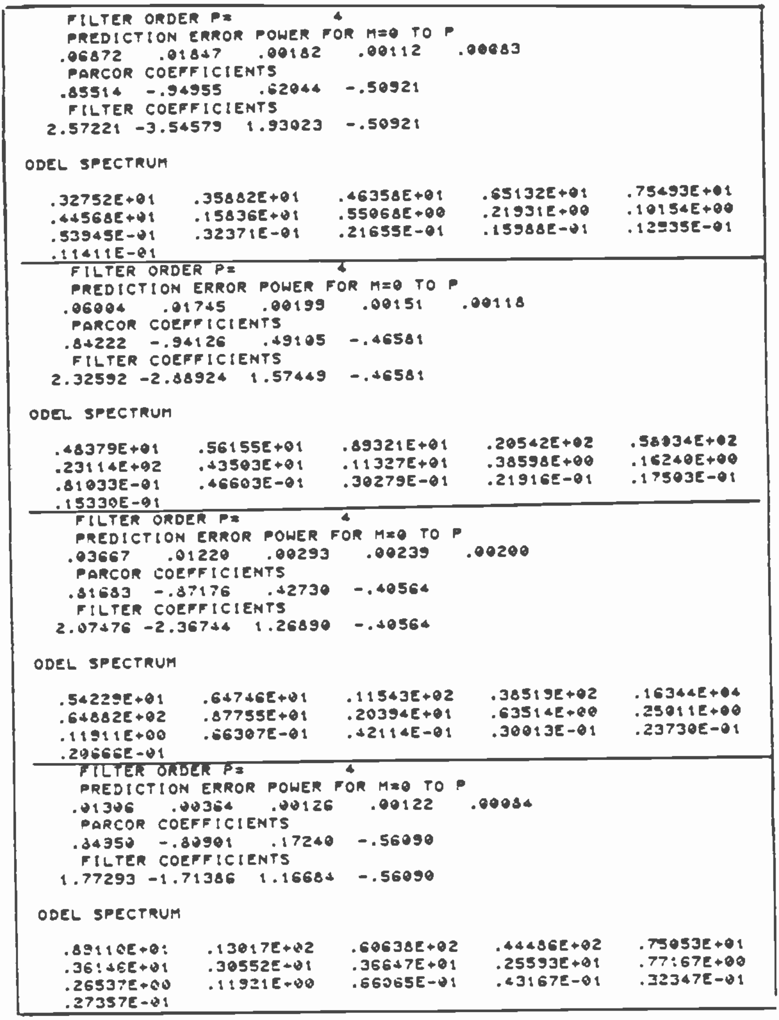
4. Application

4.1 The Time Domain Modeling

How does one apply the modeling method just described to the problem at hand? The data base analysis had suggested the need to monitor and interpret the sensor output in order to detect the presence of a strong low-frequency, coherent, component against noise or a higher frequency component.

Suppose the waveform is fed into the 4-pole lattice inverse filter of Fig. 8. (We elect to use for this application the block processing, but for ease of explanation we may as well assume that the filter is recursively adapting to the waveform.) As the filter adapts to the waveform, we monitor the (forward) residuals $\{e_i\}$ and the parcor coefficients $\{k_i\}$. Consider now the following alternative input waveforms:

(1) *Pure sinewave:* The pure sinewave can be modeled precisely with two poles (two stages) from which, if needed, we can estimate the frequency without FFT (a look-up table, perhaps, depending on the



```
    FILTER ORDER P=            4
    PREDICTION ERROR POWER FOR M=0 TO P
   .06872   .01847   .00182   .00112   .00083
    PARCOR COEFFICIENTS
   .85514  -.94955   .62044  -.50921
    FILTER COEFFICIENTS
  2.57221 -3.54579  1.93023  -.50921

ODEL SPECTRUM

  .32752E+01   .35882E+01   .46358E+01   .65132E+01   .75493E+01
  .44568E+01   .15836E+01   .55068E+00   .21931E+00   .10154E+00
  .53945E-01   .32371E-01   .21655E-01   .15988E-01   .12535E-01
  .11411E-01
    FILTER ORDER P=            4
    PREDICTION ERROR POWER FOR M=0 TO P
   .06004   .01745   .00199   .00151   .00118
    PARCOR COEFFICIENTS
   .84222  -.94126   .49105  -.46581
    FILTER COEFFICIENTS
  2.32592 -2.88924  1.57449  -.46581

ODEL SPECTRUM

  .48379E+01   .56155E+01   .89321E+01   .20542E+02   .58034E+02
  .23114E+02   .43503E+01   .11327E+01   .38598E+00   .16240E+00
  .81033E-01   .46603E-01   .30279E-01   .21916E-01   .17503E-01
  .15330E-01
    FILTER ORDER P=            4
    PREDICTION ERROR POWER FOR M=0 TO P
   .03667   .01220   .00293   .00239   .00200
    PARCOR COEFFICIENTS
   .81683  -.87176   .42730  -.40564
    FILTER COEFFICIENTS
  2.07476 -2.36744  1.26890  -.40564

ODEL SPECTRUM

  .54229E+01   .64746E+01   .11543E+02   .38519E+02   .16344E+04
  .64882E+02   .87755E+01   .20394E+01   .63514E+00   .25011E+00
  .11911E+00   .66307E-01   .42114E-01   .30013E-01   .23730E-01
  .20666E-01
    FILTER ORDER P=            4
    PREDICTION ERROR POWER FOR M=0 TO P
   .01306   .00364   .00126   .00122   .00084
    PARCOR COEFFICIENTS
   .84350  -.80901   .17240  -.56090
    FILTER COEFFICIENTS
  1.77293 -1.71386  1.16684  -.56090

ODEL SPECTRUM

  .89110E+01   .13017E+02   .60638E+02   .44486E+02   .75053E+01
  .36146E+01   .30552E+01   .36647E+01   .25593E+01   .77167E+00
  .26537E+00   .11921E+00   .66065E-01   .43167E-01   .32347E-01
  .27357E-01
```

Fig. 9—4-Pole modeling printout of four time snapshots.

estimate accuracy). From Eq. [9], one will note also that the first parcor k_1 is the same as the first-lag autocorrelation; thus, it can be used as a rough indicator of a low versus a high-frequency component. Most significant, the perfectly matched inverse filter will produce a zero residual after two stages ($e_2 \simeq 0$), irrespective of the input power.

(2) *Pure white noise:* In this case, the parcor coefficients will settle around a zero value and the residuals at the various stages will equal the input power.

(3) *Sinewave with noise:* Depending on the signal-to-noise ratio, the filter parameter patterns will be between the two previous extreme cases.

It should be noted that the residuals $\{e_i\}$ have monotonically nonincreasing values after each stage; furthermore, depending on the number and relative power of the sinusoidal components in the waveform, residual values would drop more markedly every second stage (two poles per sinewave component).

In Fig. 9, four successive parameter snapshots of a sensor signature are tabulated. Each snapshot processes 256 samples (at 256 Hz sampling rate) using the BURGO subroutine given in the Appendix. For a 4-pole inverse filter, it gives

(1) the five residuals (prediction error power), including the input power e_0,

(2) the four parcor coefficients,

(3) the four (transversal, Fig. 7) inverse filter coefficients,

(4) the spectral estimate (the first 16 values of a 32-point DFT) computed from Eq. [6].

One may note the following:

(1) There is a strong component around 32 Hz (model spectrum), that seems to shift down in the last snapshot.

(2) A significant residual drop (more than 10 to 1) after the first two stages.

(3) High absolute values of the first (same as 1-lag autocorrelation, from Eq. [9]) and second parcor.

(4) The parcor coefficients are always less than unity in absolute value irrespective of the signal power (e_0). This is not true for the transversal filter coefficients.

We can proceed now with an algorithm outline.

4.2 The Algorithm Outline

The outline presented in Fig. 2 is still valid and we can fit the time domain modeling as follows:

1. Time domain processing:
   (a) seismic trace, 0–100 Hz, sampled at 256 Hz,
   (b) data window: 64 samples, 0.5 overlaps,
   (c) 4-pole auto-regressive model,
   (d) model features monitored:
   —input power (e_0),
   —ratio of input power to error power after second stage (e_0/e_2),
   —four parcor coefficients k_1 through k_4.
2. Algorithm tests (losanges in Fig. 2):
   (a) presence of strong coherent low frequency component; either/or a combination of following:
   —high e_0/e_2 ratio, say, ≥ 2
   —high absolute value of k_1 and k_2, say, ≥ 0.6

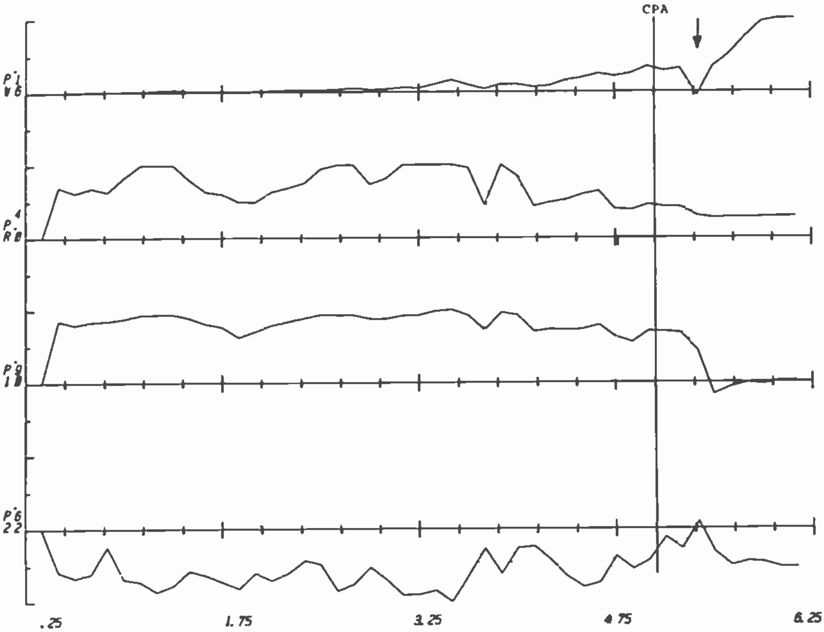


Fig. 10—Time domain features of take-off signature.

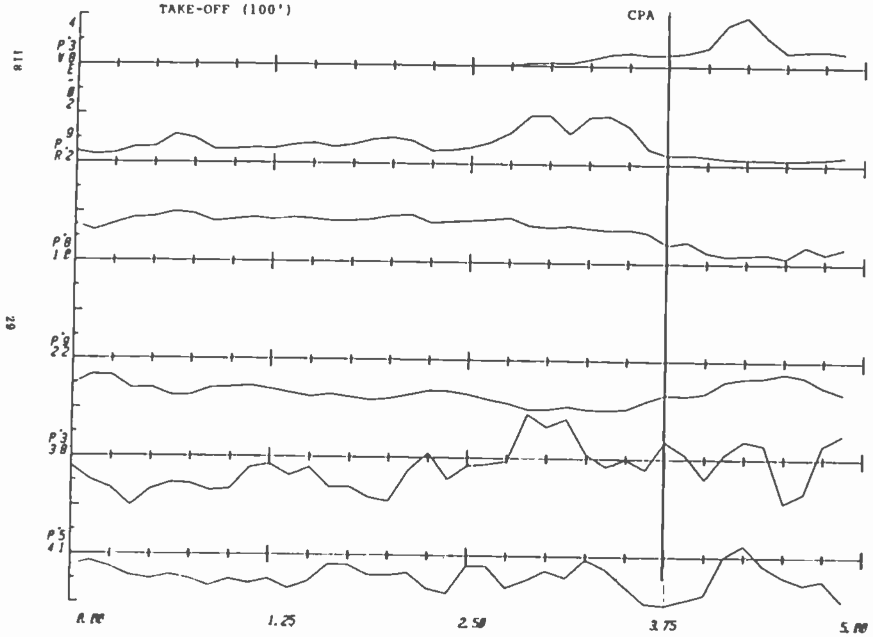


Fig. 11—Time domain features for take-off (100 feet).

(b) high noise level:
—collapse of k_1 and k_2 toward zero,
—e_0/e_2 ratio between 1 and 2.

(c) time out: we can add the monitoring of the input power e_0, looking for its peak.

An extremely simple algorithm along the criteria just described was programmed and run on a take-off trace; the four parameters are plotted versus time (in sec.) in Fig. 10.* From the top, PW, input power (e_0); PR, power ratio (e_0/e_2); P_1, parcor k_1; and P_2, parcor k_2. The algorithm includes the turn-on test, 2(a) above (noticeable in Fig. 10 a fraction of a second from the start of the lower three traces), and the increment-counter test, 2(b) above, which when true forces the input power trace slightly below zero (arrow).

This simple example highlights some of the factors affecting algorithm design and optimization; namely,

* The reader is reminded that the plotter scales each trace to its peak value which is printed vertically along the ordinate axis.

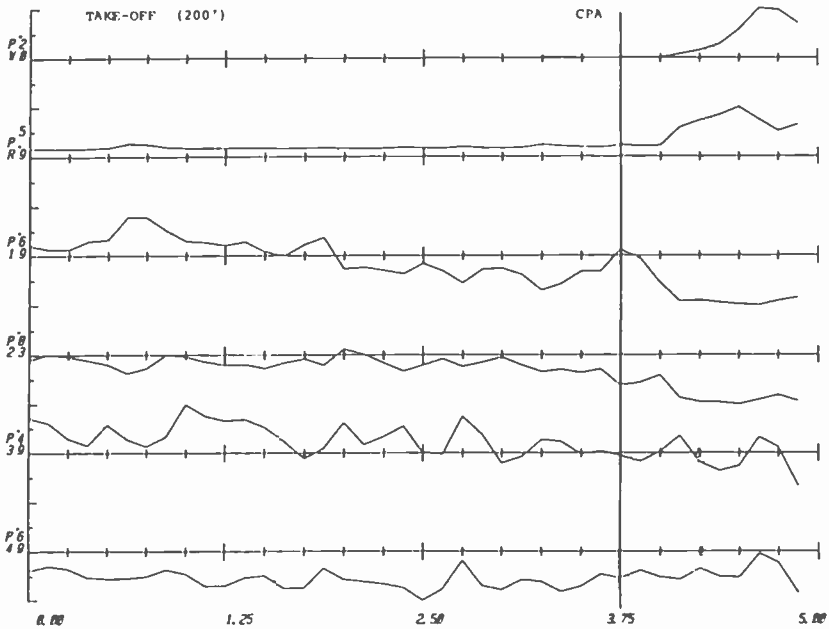


Fig. 12—Time domain features for take-off (200 feet).

(1) optimal choice of data window length, so as to smooth the inevitable fluctuations of model parameters, but not too short to miss desired transients,
(2) optimal choice of thresholds, based on statistics from data base, so as to design for certain detection objectives (probability of miss versus false alarm),
(3) optimal combination and weighting of parameters (e.g., power-ratio and parcors) for most reliable testing of events.

The algorithm design was not part of this initial investigation.

4.3 Typical Time Domain Feature Profiles

Some representative time plots of the model features are shown in Figs. 11 through 15. The six features have already been described as (1) the input power, (2) the power ratio of input to second-stage output, and (3–6) the four parcor coefficients.

Figs. 11 through 13 show the modeling of a take-off as seen from 100, 200, and 300 ft. from the target, respectively. Note at 100 ft.; the high values of PR, P1, and P2 before CPA, which collapse toward zero after CPA; the behaviour of the same parameters is reversed at 200 and 300

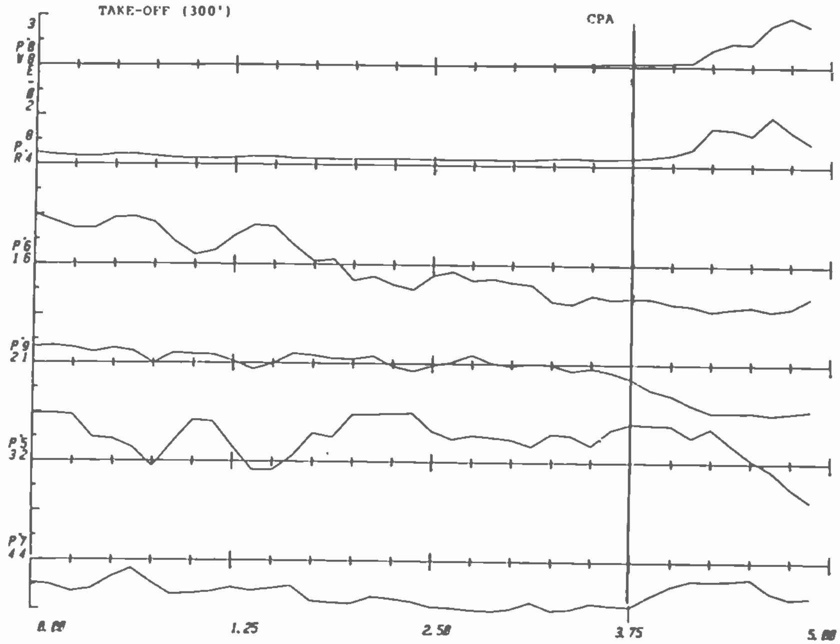


Fig. 13—Time domain features for take-off (300 feet).

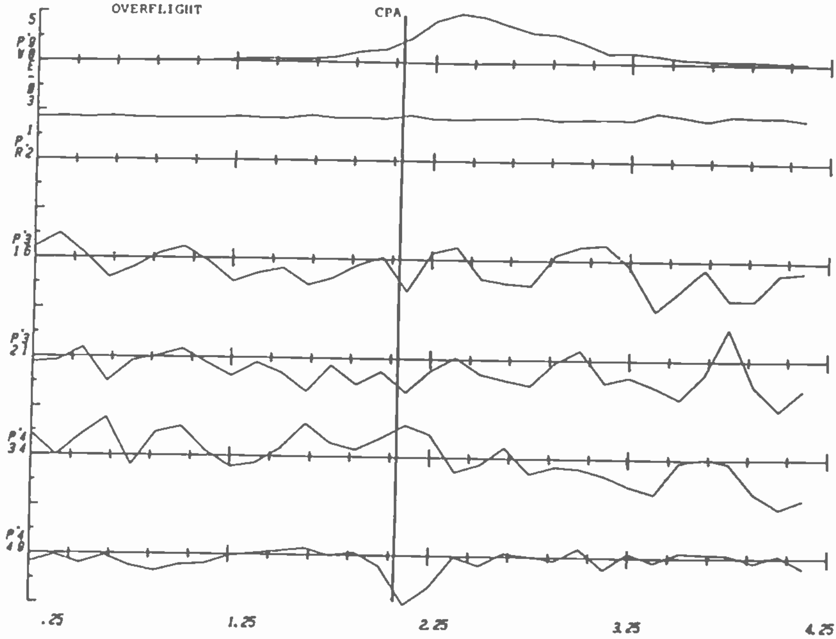


Fig. 14—Time domain features for overflight.

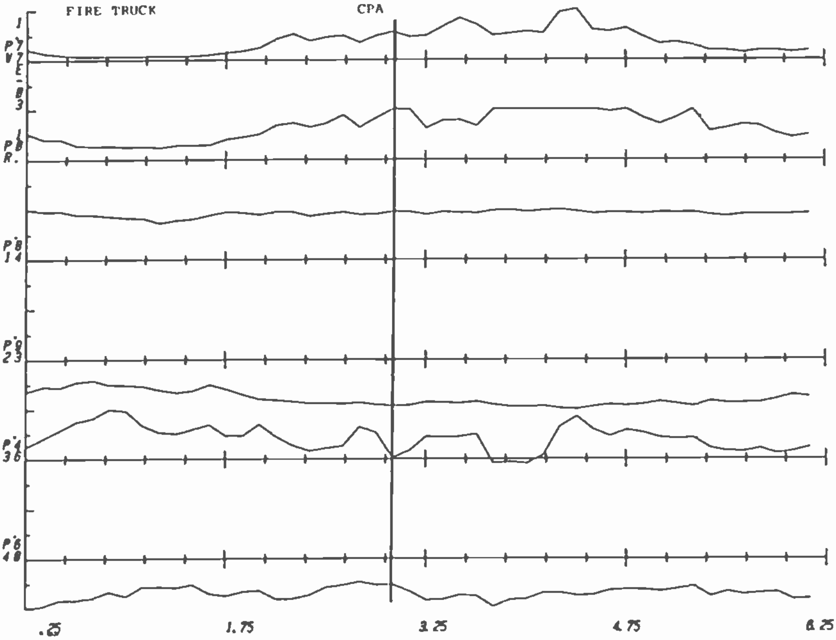


Fig. 15—Time domain features for fire truck.

ft. (note, however, that P1 after CPA is negative, denoting a high frequency coherent component).

The plots for an overflight in Fig. 14 show clearly that PR never goes above 1.5, denoting a rather noisy input to the inverse filter; this is confirmed by the rather low values of the parcor coefficients randomly oscillating around zero.

Finally, a firetruck (nuisance) produces the plots shown in Fig. 15. Very high and rather stable values of the three major features (PR, P1, and P2) are generated, not affected by the CPA crossing.

4.4 The Time Domain Modeling Computation Requirements

In choosing time domain features in preference to frequency domain counterparts for the solution of a particular problem, one has to make sure that (1) time domain features apply and work better and (2) the computation load is not prohibitive.

The tools for frequency domain feature extraction have been well refined and tested over mary years for a very wide variety of applications. The corresponding tools for time domain feature extraction have been sharpened thus far for one specific application, speech processing. Al-

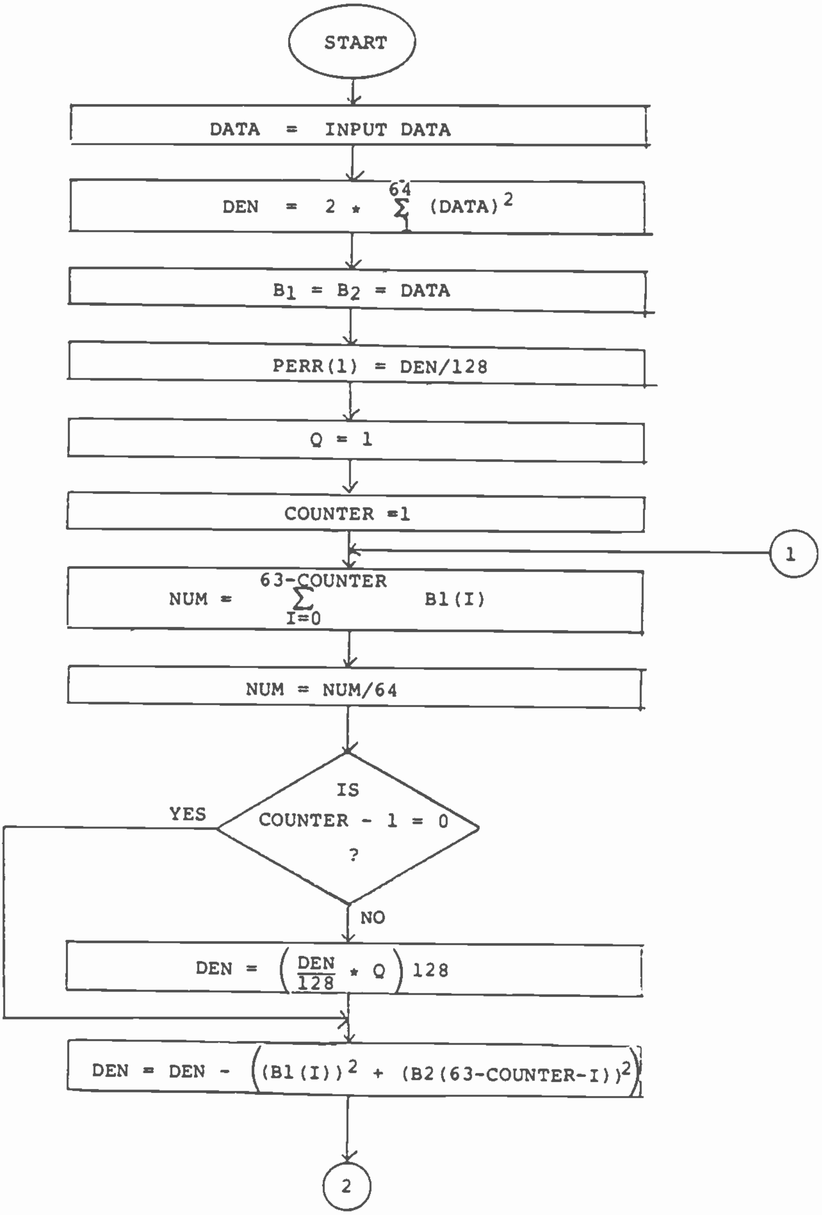

START
DATA = INPUT DATA
DEN = 2 * $\sum_{1}^{64}$ (DATA)^2
B$_1$ = B$_2$ = DATA
PERR(1) = DEN/128
Q = 1
COUNTER =1
1
NUM = $\sum_{I=0}^{63-COUNTER}$ B1(I)
NUM = NUM/64
IS COUNTER - 1 = 0 ?
YES
NO
DEN = $\left(\frac{DEN}{128} * Q\right)$ 128
DEN = DEN - $\left((B1(I))^2 + (B2(63-COUNTER-I))^2\right)$
2

Fig. 16—Maximum entropy (2-pole) algorithm flowchart.

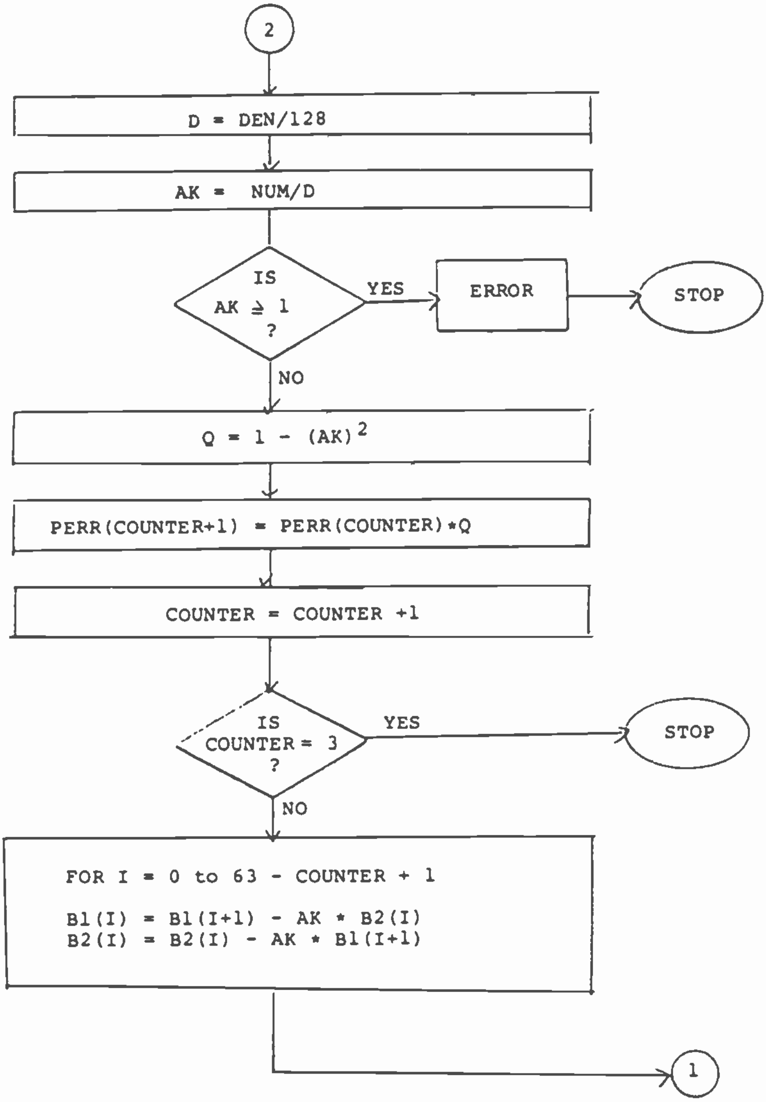

2
D = DEN/128
AK = NUM/D
IS AK ≥ 1 ?
YES
ERROR
STOP
NO
Q = 1 - (AK)²
PERR(COUNTER+1) = PERR(COUNTER)*Q
COUNTER = COUNTER +1
IS COUNTER = 3 ?
YES
STOP
NO
FOR I = 0 to 63 - COUNTER + 1
B1(I) = B1(I+1) - AK * B2(I)
B2(I) = B2(I) - AK * B1(I+1)
1

Fig. 16 (Contd.)

though this experience is applicable elsewhere, new developments have been optimized for the specific "data base," namely, speech.

An increasing amount of research is going into the development of time domain computational methods for a wide variety of applications, including those considered in this paper. Of real significance is that the effects of fixed-point arithmetic and finite length registers is well understood and analyzed for an FFT, but not yet for linear prediction analysis.

If the number of multiplications required is taken as an index of computation load,* an FFT requires for the same data window roughly as many real multiplications as a 3-pole batch lattice filter (plus one division per pole). It can be argued in favor of the latter that further processing of the FFT output is required to identify or extract the actual features; in contrast, the filter would provide the features directly. On the other hand, the range of the intermediate results during the filter computation is not as predictable as in the case of the FFT. Thus, until the filter computation dynamics are better understood, it seems advisable to simulate the fixed-point arithmetic behavior of a particular filter algorithm before proceeding to the design of its actual realization.

A limited effort was devoted to an estimate of the computational requirements, if a 2-pole inverse filter (subroutine BURGO, Appendix) had to be implemented on an 1802 (COSMAC) microprocessor. The program in assembly language was coded along the flow chart of Fig. 16, but not assembled and debugged. The main purpose at this time was to estimate time and memory requirements. The characteristics of the implementation where:

(1) Algorithm: 64-sample window, 2-pole inverse filter,
(2) Microprocessor: 1802, 4-MHz clock,
(3) Arithmetic Operations: 16-bit fixed-point,
  —24-bit sum accumulation
  —32-bit/16-bit division (software)
  —16 × 16 = 16 multiplication (hardware multiplier TCS-132)
(4) Memory: 380 8-bit bytes
(5) Execution:

	Instructions	Time (ms)
Algorithm	3409	13.63
2 Divisions	838	3.35
Total	4247	16.98

For a p-pole algorithm the number of instructions is estimated at

$$\text{(no. of instruction for } p \text{ poles)} = 459 + 1894p$$

of which $419p$ are required for the division.

It is worth noting that the execution of a 64-point FFT, implemented with the same design criteria, would require 17,824 instructions and 71.3

* This rather popular yardstick is becoming less and less realistic since with a hardware multiplier the time to multiply is the same as to add.

ms. This is a clear example that, with a hardware multiplier, the FFT algorithm (with its 6 additions per butterfly, and significant housekeeping needs) may not be the fastest way to characterize a waveform. An 8-pole filter may still be faster than a same size FFT.

A word of caution should be added to these rather encouraging estimates. The algorithm just described has not been tested for the effects of fixed-point arithmetic on the dynamic range of the computation process consistent with the parameter accuracy requirements.

A final consideration should be made regarding the range of signal bandwidths for which these time domain processing techniques are applicable. It obviously depends on (1) how frequently one needs to take a time snapshot, (2) how much additional processing of the time features is needed, and (3) how complex an implementation is acceptable. For the specific application reported here, the very modest signal bandwidth (100 Hz, sampled at 256 Hz) and algorithm involved pose no strain on a real-time implementation. For high speeds, a good term of reference would be the present state-of-the-art in speech analysis, where 4-KHz voice sampled at 8 or 10 KHz is processed through a 10-pole filter in consecutive snapshots of 20ms.

Acknowledgment

Credit for the design of the maximum entropy algorithm programming for the 1802 goes to Dominick Imbesi.

Appendix—MEM Subroutine (FORTRAN Listing)

```
FTN4,L
        SUBROUTINE BURGO (XDAT,NPNT,IPOL,PERR,AK,AA)
C
C       MEM ALGO FROM ANDERSEN, PIEEE NOV 1978,P.1581; REAL :NPUT,N SAMPL
C       S S( I ), FILTER ORDER IP (POLES); GENERATES PREDICTION ERROR POWER
C       (P) FOR ORDER 0 TO IP, PARCOR COEFF.AK( I ) AND FILTER COEFF.(AA( ).
C       N. MACINA, 5-12-79
C
C
        DIMENSION XDAT(500),B1(500),B2(500),PERR(40),AK(40),AA(40)
        M=0
        PP=0.0
        DO  10 I=1,NPNT
 10     PP=PP+XDAT( I )**2
        DEN=2*PP
        PERR( 1 )=PP/NPNT
        Q=1
        DO 20 I=1,NPNT
        B1( I )=XDAT( I )
 20     B2( I )=XDAT( I )
        DO 100 M=1,IPOL
        XNUM=0
        NM=NPNT-M
        DO 30 I=1,NM
 30     XNUM=XNUM+B1( I+1 )*B2( I )
        DEN=DEN*Q-B1( 1 )**2-B2( NPNT-M+1 )**2
        AK( M )=2*XNUM/DEN
        Q=1-AK( M )**2
        PERR( M+1 )=PERR( M )*Q
```

```
      M1=M-1
      IF(M.EQ.1) GOTO 45
      DO 40 I=1,M1
40    AA(I)=AA(I)-AK(M)*AA(M-I)
45    AA(M)=AK(M)
      IP1=IPOL+1
      IF(M.EQ.IPOL) GOTO 100
      DO 50 I=1,NM
      B1(I)=B1(I+1)-AK(M)*B2(I)
50    B2(I)=B2(I)-AK(M)*B1(I+1)
100   CONTINUE
      RETURN
      END
```

References:

[1] T. Barnwell, "Recursive Autocorrelation Computation for LPC Analysis," 1977 International Conf. on ASSP, Hartford, Conn., May 1977, pp. 1–4.

[2] J. Makhoul, "Linear Predictor: A Tutorial Review," *Proc. IEEE*, **63**, p. 561, April 1975.

[3] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*, Chapter 8, Prentice-Hall Book Co., Inc., N. Y. (1978).

[4] T. J. Ulrych and T. N. Bishop, "Maximum Entropy Spectral Analysis and Autoregressive Decomposition," *Reviews of Geophysics and Space Physics*, **13**, p. 183, Feb. 1975.

[5] B. Windrow, et al., "Stationary and Nonstationary Learning Characteristics of the LMS Adaptive Filter," *Proc. IEEE*, **64**, p. 1151, Aug. 1976.

An Analytic Solution of a Magnetic Electron-Beam Bender

K. K. N. Chang

RCA Laboratories, Princeton, NJ 08540

Abstract—A magnetic bender featuring a hyperbolically varied transverse field is proposed to bend and collimate electrons into a very sharp angle. An analytic solution to reveal the bending and collimation physics is presented. The bending angle is found to be a measure of the total number of the flux lines cut by the traversing electron. Three in-line beams bent in a sharp 90° angle are observed with full collimation. The bender has potential application to commercial kinescopes in reducing their form factors.

Introduction

There has long been a conceptual interest in electron beams bent through a sharp right angle for use in advanced cathode-ray tubes.[1] In standard kinescopes, also, sharp beam bending, if accomplished in a collimated manner, could result in a significant reduction of the structural form factor.

In electron optics, an electron beam can be bent either by an electrostatic field or a magnetic field. Because of the complex nature of the lens configuration, determination of the bent trajectory by analytic solutions is rare and numerical solutions through computer simulating must be resorted to.

This paper describes a permanent-magnet structure that bends and collimates electron beams in a 90° sharp turn. The bender is a rectangular parallel-plate pole structure that produces a uniform field in one direction and a hyperbolically varied field in the transverse direction.

We have arrived at an analytic solution by which the electron trajectory in such field can be rigorously determined. Furthermore, the analytical model described enables us to present a physical picture of the field-beam interaction more easily than is possible with computer simulation. The analytical results obtained have been substantiated in an experimental magnetic bender.

Magnetic Beam Bender

The magnetic beam bender under investigation is a slot-shaped rectangular structure of two parallel-plate pole pieces (Fig. 1). Two permenent magnets at two ends supply the magnetic field B_z in the slot at the same polarity. The magnetic field is assumed to be z-directional in the midplane between the two parallel plates. Beam bending of interest is confined in that plane, i.e., $z = 0$, where the field distribution is considered to be constant in the x direction. In the y direction, the field is approximated as a hyperbolic function, which has been verified by measurements:[2]

$$B_z(y) = B_o \operatorname{sech} k_o y,$$

where k_o is defined by the y and z dimensions of the magnetic field region.

Analytic Model

Electron motion in a nonuniform transverse magnetic field is described by the following force equations,

$$\begin{aligned} \ddot{x} &= \omega_z(y)\dot{y} \\ \ddot{y} &= -\omega_z(y)\dot{x}, \end{aligned} \qquad [1]$$

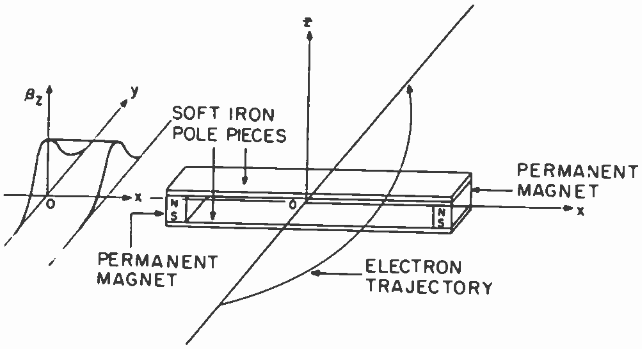


Fig. 1—A magnetic electron-beam bender.

where $\dot{x}$ and $\dot{y}$ are the velocity components and $\ddot{x}$ and $\ddot{y}$ are the force components. $\omega_z(y) = (e/m)B_z(y)$ is the cyclotron frequency, e/m being the electron charge-to-mass ratio.

The solution of interest is the beam bending curve in the x-y plane at $z = 0$. The trajectory velocity v_o, the time derivative of the trajectory length(S), is conserved for drifting electrons under the influence of a transverse magnetic field, i.e.,

$$v_o^2 = \left(\frac{dS}{dt}\right)^2 = \dot{x}^2 + \dot{y}^2. \qquad [2]$$

The trajectory slope can be written

$$\tan\theta = \frac{dx}{dy} = \frac{\dot{x}}{\dot{y}}. \qquad [3]$$

To arrive at the solution of the bending trajectory, one formulates from Eq. [1] as follows:

$$\dot{y}\ddot{x} - \dot{x}\ddot{y} = \omega_z(\dot{y}^2 + \dot{x}^2) = \omega_z\dot{y}^2(1 + x'^2) = \dot{y}^2\frac{d}{dt}\left(\frac{\dot{x}}{\dot{y}}\right),$$

which leads to

$$x'' = (1 + x'^2)^{3/2}\left(\frac{\omega_o}{v_o}\right)\operatorname{sech} k_o y, \qquad [4]$$

where $\omega_o = (e/m)B_o$. Since

$$\frac{d(\sin\theta)}{dy} = \frac{x''}{(1 + x'^2)^{3/2}}$$

Eq. [4] becomes

$$d(\sin\theta) = \frac{\omega_o}{v_o}\operatorname{sech} k_o y dy. \qquad [5]$$

Integration of Eq. [5] with limit $\theta = \theta_0$ when $y = -y_o$ yields

$$\sin\theta - \sin\theta_0 = \frac{\omega_o}{k_o v_o}[\tan^{-1}(\sinh k_o y) + \tan^{-1}(\sin k_o y_o)] \qquad [6]$$

Results

Figs. 2 and 3 show several electron trajectories under the influence of a magnetic bender according to the analytic solution given by Eq. [6]. These trajectories in the x-y plane are derived first by plotting dx/dy ($= \tan\theta$) versus y and then by computing $x (= \int \tan\theta\, dy)$ with numerical integration. The graphs use normalized coordinates of y/y_o and x/x_o,

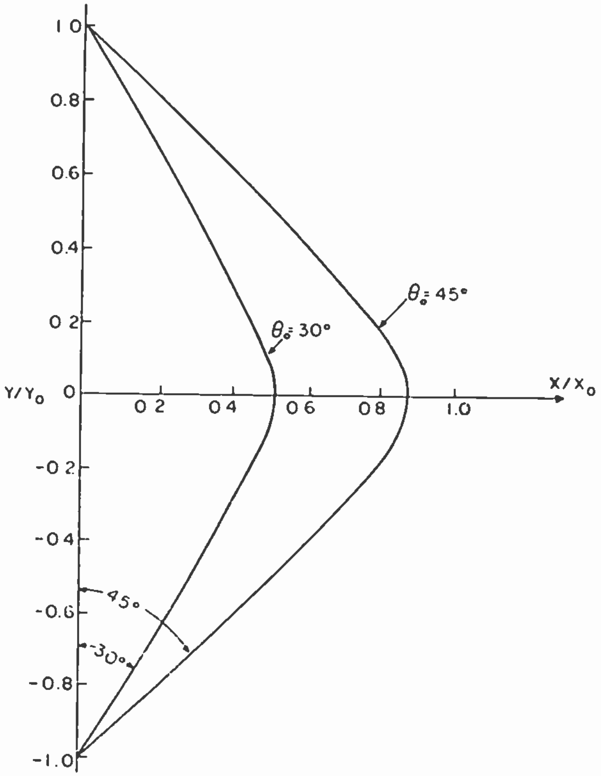


Fig. 2—Symmetrical bent trajectory as a function of initial launching angle.

where x_o is the x-intercept of the line passing through the initial point $(0, y_o)$ and with an initial slope of $\tan\theta_0$. In both figures, the magnetic field is adjusted such that all the trajectory slopes are zero at $y = 0$. It is interesting to note that all trajectories are symmetrical with respect to the x axis. To show mathematically this symmetry, one writes, from Eq. [6], at $y = 0$

$$\sin\theta_0 = -\frac{\omega_o}{k_o v_o} \tan^{-1} (\sinh k_o y_o) \qquad [7]$$

and converts Eq. [6] into

$$\sin\theta = -\frac{\omega_o}{k_o v_o} [\tan^{-1} (\sinh k_o y)] \qquad [8]$$

Eq. [7] gives the relationship among the peak magnetic field, the entrance angle, and the initial y coordinate. Trajectories for $\theta_0 = 30°$ and $\theta_0 = 45°$ at a given initial position $y = y_o$ are plotted in Fig. 2.

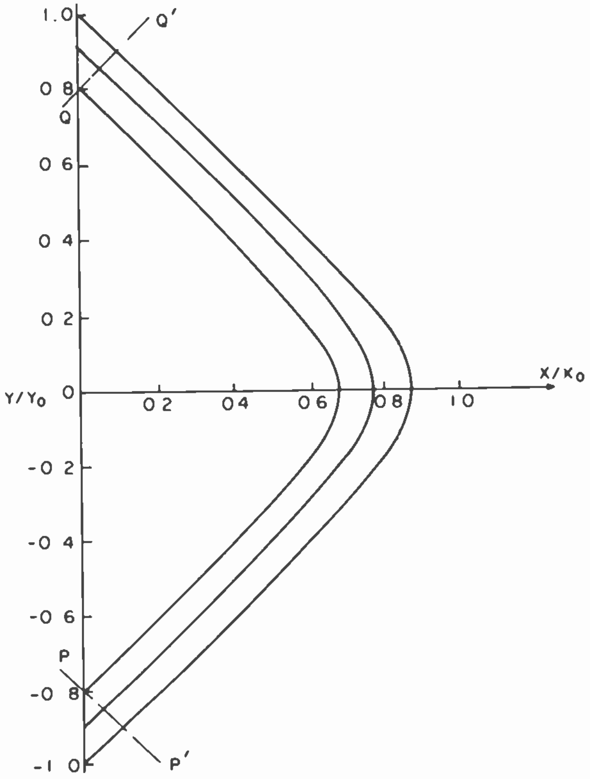


Fig. 3—Collimated trajectories in a 90° magnetic bender.

If the initial position $y = y_o$ is considered to be practically out of the magnetic field, i.e., $k_oy_o \gg 1$ and $(\sinh k_oy_o)^{-1} \approx 0$, Eq. [7] becomes

$$\sin\theta_0 = -\frac{\pi}{2}\frac{\omega_o}{k_ov_o}. \qquad [9]$$

If launched at a given angle, all trajectories with $k_oy_o \gg 1$, which are dependent only on the y position, are parallel according to Eq. [8]. Fig. 3 shows these trajectories in a 90° angle bending.

In practice, a parallel electron beam makes its entrance to the magnetic bender at a normal plane PP′, shown in Fig. 3, and then leaves the bender with the precise collimation at a symmetrical normal plane QQ′, which is at right angles with respect to the entrance plane PP′. In the case of three parallel beams, as those originating, for example, from an in-line gun, they would also be simultaneously bent and collimated in the same manner as the single beam. The parallelism of three beams bent through a magnetic bender has been experimentally observed.

Physical Interpretation

In electron dynamics, a drifting electron would rotate in a transverse uniform field at a cyclotron frequency. The angle displacement is related to the total number of flux lines that the transversing electron cuts during its circular course. Expressed in mathematical terms, the angle displacement is

$$\theta = \frac{\omega}{v_o} S, \qquad [10]$$

where ω is the cyclotron frequency, S is the traversed path length, and v_o is the electron drift velocity.

Under the influence of a transverse nonuniform magnetic field, such as the hyperbolic function being considered here, the angle displacement of a drift electron would also be expected to be related to the number of flux lines traversed. The physical interpretation appears to be simple and plausible. The mathematical proof is, however, not easily attainable for most of the nonuniform fields dealt with in practical magnet geometries. Nevertheless, an analytic solution happens to exist in the present case and the angle displacement can be derived forthwith.

It follows from Eq. [5] that

$$d\theta = \frac{\omega_o}{v_o} \frac{\mathrm{sech} k_o y}{\cos\theta} dy.$$

Thus, by making use of the cosine relation $dy/dS = \cos\theta$,

$$\theta = \int \frac{\omega_o}{v_o} \frac{\mathrm{sech} k_o y}{\cos\theta} dy = \int \frac{\omega_o}{v_o} \mathrm{sech} k_o y dS, \qquad [11]$$

which is similar to Eq. [10] except that the total flux in Eq. [11] is derived through the integration of the nonuniform field over the traversed path.

The number of flux lines as a measure of the angle displacement also explains the trajectory symmetry with respect to the magnet axis where $\theta = 0$. Since the field is symmetrical, the amount of angular displacement, according to Eq. [11], would also be expected to be symmetrical.

Conclusion

A sharply bent electron beam, long sought for kinescopes for the purpose of achieving reduced form factors, is advanced from concept to reality. The achieved practical device is a permanent magnet bender featuring a hyperbolically varied field that bends and collimates a single beam or multiple in-line beams in a sharp 90° angle. To explore the bending physics, an analytic solution is achieved. The bending angle is found to be a measure of the total number of flux lines cut by the traversing electron.

References:

[1] D. Gabor, et al, "A New CR Tube for Monochrome and Color Television," *Proc. IEE Part B.*, **105**, p. 581 (1958); W. R. Aiken, "A Thin CR Tube," *Proc. IRE*, **45**, p. 1599 (1957)

[2] K. K. N. Chang, "Collimated Electron Trajectory in a Nonuniform Transverse Magnetic Field," *RCA Review*, **40**, No. 4, p. 390, Dec. 1979.

Carrier Mobility in Laser-Annealed Silicon-on-Sapphire Films

E. W. Maby* and C. P. Wu

RCA Laboratories, Princeton, NJ 08540

Abstract—Average carrier mobilities determined by Hall-resistivity measurements are reported for ion-implanted silicon-on-sapphire (SOS) films that were annealed with a pulsed Nd-YAG laser. The results demonstrate significant mobility improvement when laser annealing follows ion implantation. Although less impressive, mobility improvements are also realized for an SOS processing sequence in which laser annealing preceeds ion implantation and furnace annealing.

Introduction

Silicon-on-sapphire (SOS) is a widely-used heteroepitaxial material for the fabrication of metal-oxide-silicon (MOS) integrated circuits. Compared with bulk silicon technology, SOS technology features an exceptional degree of electrical isolation between individual or groups of MOS circuit elements as well as resistance to the deleterious effects of ionizing radiation. However, as-grown SOS films often exhibit relatively poor crystalline quality, particularly near the silicon-sapphire interface.[1,2] Crystal imperfections are presumed to be responsible for low carrier mobility and associated MOS speed limitations.[3,4]

Recently, there has been interest in improving the crystalline and electrical quality of SOS films by a laser annealing process.[5-9] For example, one study, in which n-channel MOS transistors were fabricated on SOS islands that had been annealed by a pulsed excimer laser ($\lambda =$

* Present address: Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139.

0.249 μm), suggested a 30% improvement in channel mobility on the basis of saturation-current—gate-voltage transistor characteristics.[7] In this paper, we present electron and hole mobility data, as determined by direct Hall-resistivity measurements, for ion-implanted furnace-annealed SOS films that were also annealed by a pulsed, frequency-doubled Nd-YAG laser ($\lambda = 0.530\ \mu m$).

Experiment

Hall bar mesas (240 × 1260 μm) with six metallized contact pads were formed from a 0.6-μm SOS film following ion implantation and annealing. The n-type hall bars received a 90-keV phosphorus implant at a dose of $2.05 \times 10^{13}\ cm^{-2}$. The p-type hall bars received a 30-keV boron implant at a dose of $8.95 \times 10^{13}\ cm^{-2}$. The average depth of ion penetration for both implants was 0.20 μm. After implantation, the hall bars were capped with a clean CVD oxide and furnace annealed at 1000°C for thirty minutes in a dry nitrogen ambient. Laser annealing was performed with a pulsed, frequency doubled ($\lambda = 0.53\ \mu m$) Nd-YAG laser. Each pulse duration was 90 nsec and the beam size was approximately $5 \times 10^{-5}\ cm^2$. In one experiment, the laser annealing followed the furnace anneal of the implant and removal of the oxide cap. In a separate experiment, the laser annealing preceeded implantation. Hall resistivity procedures and analysis followed those outlined by Johansson et al.[10] Measurements were performed with a magnetic field of ± 10 kilogauss. At this high field, Hall and drift mobilities in extrinsic silicon are equal.[11] The mobility values reported necessarily reflect a weighted average over the implanted layer.

The mobility data for each experiment appear in Table 1. When the laser anneal follows implantation (Experiment 1), there is a dramatic improvement (93%) in average electron mobility to a value that would be typical in bulk silicon, given an annealing radiation energy density of 2.0 J/cm^2. At an annealing radiation energy density of 1.5 J/cm^2, the improvement is only marginal (9%). The behavior of average hole mobilities is consistent with these observations, but the improvement following the 2.0 J/cm^2 anneal is less impressive. When the laser annealing is performed before implantation (Experiment 2), there is a general improvement in average carrier mobility by about 5%. This improvement is not particularly sensitive to the energy density of the annealing radiation.

The quantity N_{avg} which appears in Table 1 is defined as the total integrated carrier concentration. Under ideal circumstances, all ion-implanted impurities will be ionized following the anneal, and N_{avg} will equal the implantation dose, N_{dose}. Second-order perturbations are

Table 1—Mobility Data for Laser Annealing Before (Experiment 1) and After (Experiment 2) Implantation

		Experiment 1			Experiment 2	
Anneal	Mobility (sm^2/V-sec)	Resistivity (Ω/□)	N_{avg} (cm^{-2})	Mobility (cm^2/V-sec)	Resistivity (Ω/□)	N_{avg} (cm^{-2})
Electron Mobility						
L.A. 1.5 J/cm^2	247 ± 10	2400 ± 110	1.05 × 10^{13}	270 ± 0.5	1310 ± 5	1.77 ± 10^{13}
Control	227 ± 1	1820 ± 30	1.51 ± 10^{13}	251 ± 2	1370 ± 10	1.82 × 10^{13}
L.A. 2.0 J/cm^2	423 ± 13	1700 ± 135	8.66 ± 10^{12}	243 ± 1	1530 ± 10	1.68 × 10^{13}
Control	219 ± 7	1940 ± 65	1.47 ± 10^{13}	231 ± 0.5	1510 ± 5	1.79 × 10^{13}
Hole Mobility						
L.A. 1.5 J/cm^2	77.5 ± 0.4	853 ± 7	9.42 × 10^{13}	70.5 ± 0.2	735 ± 2	1.21 × 10^{14}
Control	68.9 ± 1.0	801 ± 4	1.13 × 10^{14}	67.4 ± 0.2	748 ± 3	1.24 × 10^{14}
L.A. 2.0 J/cm^2	98.2 ± 6.7	863 ± 65	7.37 × 10^{13}	71.6 ± 0.3	773 ± 1	1.13 × 10^{14}
Control	67.4 ± 0.2	833 ± 2	1.11 × 10^{14}	68.5 ± 0.3	738 ± 5	1.23 × 10^{14}

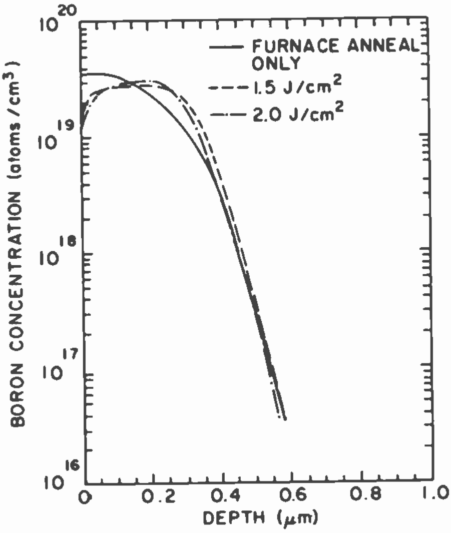


Fig. 1—SIMS concentration profiles of boron implanted into SOS at 30 keV and laser-annealed at different energy densities. The implantation dose of 8.95×10^{13} cm^{-2} exceeds the dose for Hall-resistivity measurements by a factor of ten.

dependent upon the exact shape of the impurity distribution. The N_{avg} values for the conditions of Experiment 2 and the control conditions of Experiment 1 are greater than N_{dose} for p-type implanted layers and less than N_{dose} for n-type ion-implanted layers. These observations properly reflect the presence of p-type aluminum that has diffused into the silicon from the sapphire substrate during the higher-than-normal-temperature furnace anneal. The N_{avg} values for the laser-annealed conditions of Experiment 1 monotonically decrease with increasing laser radiation power density from otherwise consistent levels for both p-type and n-type ion-implanted layers. This observation suggests the simultaneous, possibly amphoteric, presence of acceptor-like and donor-like damage imperfections, with energy levels deep within the forbidden energy band, which are introduced by the laser-annealing process. The validity of this model has yet to be verified by other, more direct measurements.

Discussion

The discussion which follows assumes that the total mobility can be written in the form

$$\frac{1}{\mu} = \frac{1}{\mu_l} + \frac{1}{\mu_i} + \frac{1}{\mu_N}, \qquad [1]$$

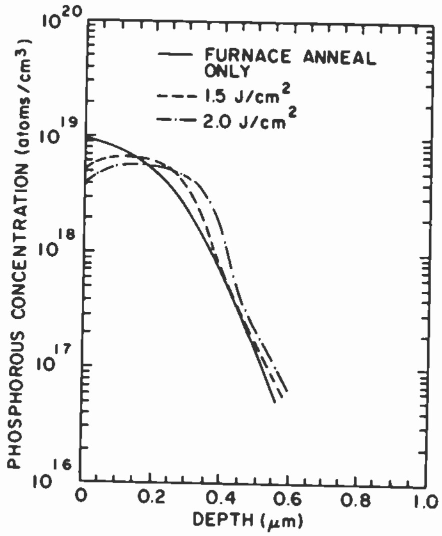


Fig. 2—SIMS concentration profiles of phosphorus implanted into SOS at 90 keV and laser-annealed at different energy densities. The implantation dose of 2.05 × 10^{13} cm^{-2} exceeds the dose for Hall-resistivity measurements by a factor of ten.

where μ_l is the carrier mobility limited by phonon scattering, μ_i is the carrier mobility limited by ionized imperfection scattering, and μ_N is the mobility limited by neutral imperfection scattering. The factor μ_l is assumed to be independent of the annealing sequence chosen. The factor μ_i has the Brooks-Herring[12,13] form

$$\mu_i = 2^{7/2}\pi^{-3/2}(kT)^{3/2}\epsilon^2 e^{-3} m^{*-1/2} N_I^{-1}\left[\ln\left(\frac{24m^*(kT)^2\epsilon}{e^2h^2N'}\right)\right]^{-1}. \qquad [2]$$

In this expression, k is Boltzmann's constant, T is the absolute temperature, ϵ is the dielectric permittivity, e is the fundamental unit of charge, m^* is the carrier effective mass, h is Planck's constant, N_I is the total density of ionized imperfections, and N' is the total density of carriers. The μ_N mobility factor has the form[14]

$$\mu_N = \frac{2m^*e^3\pi^3}{5\epsilon h^3 N_n}, \qquad [3]$$

where N_n is the concentration of neutral imperfections.

Figs. 1 and 2 show the control and laser-annealed concentration profiles of Experiment 1 for boron and phosphorus respectively. In order to improve sensitivity, the implantation doses for the Secondary Ion Mass Spectroscopy (SIMS) analysis were greater than the respective doses for Hall-resistivity measurements by a factor of ten. The dose differences are presumed not to affect the diffusive character of the implants. Changes in the average impurity concentration are not sufficient, given the form of Eq. [2] and assuming complete ionization, to explain the dramatic mobility improvements that are observed on the basis of an increased μ_i factor. Indeed the total concentration of ionized imperfections in Experiment 1 must be greater than the concentration of impurities by ion implantation, since these impurities appear to be compensated by other charged imperfections, thereby resulting in a decrease rather than in the μ_i mobility factor given by Eq. [2]. One would hope, therefore, that the mobility improvements of Experiment 1 could be further enhanced by removal of excess ionized imperfections through subsequent low-temperature furnace annealing steps or hydrogenation.[15] These experiments were not performed. The mobility improvements of Experiment 2 are less likely to be influenced by changes in μ_i due to the nature of that experiment.

It is likely that the mobility improvements of both experiments are dominated by changes in μ_N. The concentrations of neutral imperfections required of Eq. [3] in order to support this hypothesis are not unreasonable for SOS material.[16] As a limiting example, let the average electron mobility of 423 cm^2/V-sec for the 2.0 J/cm^2 laser anneal of Experiment 1 represent a case of no mobility limitation by neutral imperfection scattering ($\mu_N \rightarrow \infty$) and let the control average electron mobility of 219 cm^2/V-sec represent a particular concentration of neutral imperfections N_N. The N_N concentration required for this total mobility difference is approximately 2.6×10^{19} cm^{-3}. This paper makes no claim as to the nature of neutral imperfections at comparable concentrations or the specific mechanism for the elimination by laser annealing. The interested reader is referred to the discussion of Ref. [16].

The mobility differences between Experiment 1 and Experiment 2 suggest that little may be gained by processing pre-laser-annealed SOS but that mobility degradation due to imperfect annealing regrowth following ion implantation can be minimized when the final annealing mechanism is an appropriately chosen laser annealing process. Nevertheless, the small improvement in Experiment 2 is encouraging for electronic applications, given the following explanation. Any improvement in the starting SOS material will be primarily in the top 0.2 to 0.3 μm. If the laser processing is followed by a relatively heavy implantation, as is required in this experiment for a direct Hall mobility measurement, the damaged silicon must subsequently regrow by solid-phase epitaxy from a region near the Si/Al$_2$O$_3$ interface, where there is little or no difference between the control and the laser-annealed samples. More favorable performance results of MOS transistors fabricated in pre-laser-annealed SOS will be presented in a separate communication.[17]

Acknowledgments

It is a pleasure to acknowledge Ethel Moonan and Frank Kolondra for their technical assistance throughout this work and Dr. Charles Magee for providing the SIMS analysis.

References:

[1] M. Abrahams and C. J. Buiocchi, "Cross-Sectional Electron Microscopy of Silicon on Sapphire," *Appl. Phys. Lett.*, **27**, p. 324 (1975).

[2] M. S. Abrahams, C. J. Buiocchi, J. F. Corboy, Jr., and G. W. Cullen, "Misfit Dislocations in Heteroepitaxial Si on Sapphire," *Appl. Phys. Lett.*, **28**, 275 (1976).

[3] S. T. Hsu and J. H. Scott, Jr., "Mobility of Current Carriers in Silicon-on-Sapphire (SOS) Films," *RCA Review*, **36**, p. 240 (1975).

[4] S. T. Hsu, "Electron Mobility in SOS Films," *IEEE Trans. Electron Devices*, **ED-25**, p. 913 (1978).

[5] M. E. Roulet, P. Schwob, K. Affolter, W. Luthy, M. von Allmen, M. Fallavier, J. M. Mackowski, M. A. Nicolet, and J. P. Thomas, "Laser Annealing of Silicon on Sapphire," *J. Appl. Phys.*, **50**, 5536 (1979).

[6] W. Luthy, K. Affolter, H. P. Weber, M. E. Roulet, M. Fallavier, J. P. Thomas and J. Mackowski, "Dynamics of Nd:YAG Laser Annealing of Silicon on Sapphire," *Appl. Phys. Lett.*, **35**, p. 873 (1979).

[7] G. Yaron and L. D. Hess, "Application on Laser Annealing Techniques to Increase Channel Mobility in Silicon on Sapphire Transistors," *Appl. Phys. Lett.*, **36**, 220 (1980).

[8] G. Yaron and L. D. Hess, "Laser Annealed Silicon on Sapphire," *IEEE Trans. Electron. Devices*, **ED-27**, p. 573 (1980).

[9] G. Yaron, L. D. Hess, and S. A. Kororowski, "Laser Annealing Effects on the Electrical Characteristics of SOS Transistors," *Solid State Electron*, **23**, p. 893 (1980).

[10] N. G. E. Johansson, J. W. Mayer, and O. J. Marsh, "Technique Used in Hall Effect Analysis of Ion Implanted Si and Ge," *Solid State Electron.*, **13**, p. 319 (1970).

[11] F. J. Blatt, "Theory of Mobility of Electrons in Solids," *Solid State Physics*, **4**, 199 (1957).

[12] H. Brooks, "Scattering by Ionized Impurities in Semiconductors," *Phys Rev.*, **83**, p. 879 (1951).

[13] H. Brooks, "Theory of the Electrical Properties of Germanium and Silicon," *Adv. Electronics and Electron Phys.*, **7**, p. 85 (1955).

[14] C. Erginsoy, "Neutral Impurity Scattering in Semiconductors," *Phys. Rev.*, **79**, p. 1013 (1950).

[15] J. I. Pankove and M. L. Tarng, "Amorphous Silicon as a Passivant for Crystalline Silicon," *Appl. Phys. Lett.*, **34**, p. 156 (1979).

[16] A. C. Ipri and J. N. Zemel, "Impurity Centers in Silicon Films on Sapphire," *J. Appl. Phys.*, **44**, p. 744 (1973).

[17] C. P. Wu and B. Leung, to be published.

Patents Issued to RCA Inventors—Fourth Quarter 1980

October

A. Abramovich Relative Humidity Measurement (4,227,411)

A. A. Ahmed Television Vertical Ramp Generator (4,230,998)

A. A. Ahmed Oscillator Incorporating Negative Impedance Network Having Current Mirror Amplifier (4,230,999)

F. Aschwanden Self-Stabilizing Analog to Digital Converter Useful in Phase Locked Loop Tuning Systems (4,227,186)

L. R. Avery Video Games Color Synthesis (4,229,760)

A. R. Balaban and S. A. Steckler Multiplexing Arrangement for a Television Signal Processing System (4,227,217)

W. H. Barkow Deflection Yoke With a Magnet for Reducing Sensitivity of Convergence to Yoke Position (4,231,009)

W. Bohringer Regulated Deflection System (4,227,125)

F. J. Campbell and W. H. Barkow Saddle-Toroid Deflection Winding for Low Loss and/or Reduced Conductor Length (4,228,413)

D. E. Carlson Method of Enhancing the Electronic Properties of an Undoped and/or N-Type Hydrogenated Amorphous Silicon Film (4,226,643)

R. R. Demers Method of Slicing a Wafer (4,227,348)

R. R. Demers and M. A. Leedom System for Regulating the Applied Blade-to-Boule Force During the Slicing of Wafers (4,228,782)

W. F. Dietz Switching Amplifier for Driving a Load Through an Alternating-Current Path with a Constant-Amplitude, Varying Duty Cycle Signal (4,227,123)

M. Ettenberg and C. J. Nuese III-V Direct-Bandgap Semiconductor Optical Filter (4,228,349)

T. M. Gurley, W. D. Fischer, and R. S. Hopkins, Jr. Television Picture Positioning Apparatus (4,227,215)

L. A. Harwood and E. J. Wittmann Signal Detector Including Sample and Hold Circuit With Reduced Offset Error (4,229,759)

W. M. Kramer Reduced Blooming Device Having Enhanced Quantum Efficiency (4,228,446)

C. J. Martin, R. J. Ryan, and M. J. Voelker Conductive Molding Composition (4,228,050)

J. D. Mazgy Logic Circuit (4,228,371)

D. C. McCarthy Improving Etch Resistance of a Casein-Based Photoresist (4,230.794)

L. S. Napoli Solar Cell Grid Patterns (4,228,315)

J. A. Olmstead Engine Timing Circuit With Noise Immunity (4,226.219)

P. C. Olsen Wire Coil Assembly for an Electrical Circuit (4,229,722)

R. W. Paglione Temperature Controller for a Microwave Heating System (4,228,809)

W. Philips, C. C. Neil, and J. M. Hammer Planar Optical Waveguide Comprising Thin Metal Oxide Film Incorporating a Relief Phase Grating (4,227,769)

J. J. Piascinski and E. E. Doerschuk, 3rd Method for Making Etch-Resistant Stencil With Dichromate-Sensitized Alkali-Caseinate Coating (4,230,781)

J. S. Pistiner Closed Loop Roll Control for Momentum Biased Satellites (4,230,294)

A. Rosen and W. P. Santamore Method of Measuring Blood Perfusion (4,228,805)

S. A. Steckler and A. R. Balaban Switched AFPC Loop Filter With Offset Voltage Cancellation (4,228,463)

M. W. Stewich Charge-Coupled Device Imager System (4,231,062)

T. Takahashi and O. Yamada High Temperature Cadmium Boracite Semiconductor Device (4,228,454)

L, N. Thibodeau Flyback Transformer (4,229,787)

A. J. Tocci Cleaving Apparatus (4,228,937)

L. A. Torrington Manual Scanning Mechanism for VideoDiscs Player (4,227,699)

C. P. Wu and R. K. Smeltzer Low-Resistivity Polycrystalline Silicon Film (4,229.502)

C. P. Wu and A. Rosen Method of Making an Impatt Diode Utilizing a Combination of Epitaxial Deposition, Ion Implantation and Substrate Removal (4,230,505)

J. P. Yu RF Connector Assembly With Provision for Low Frequency Isolation and RFI Reduction (4,229,714)

November

B. Astle and R. A. Dischert Automatic Setup System for Television Cameras (4,234,890)

A. V. Bedford Method of and Means for Generating Complex Electrical Coding Waves for Secret Communication (4,232,186)

A. E. Bell Playback Information Record Using Phase Cancellation for Reading (4,233,626)
D. Botez, M. Ettenberg, and H. Kressel Light Emitting Diode (4,233,614)
H. Chen and R. H. Hughes Electron Gun With Astigmatic Flare-Reducing Beam Forming Region (4,234,814)
T. L. Credelle Flat Display Tube Having Shielding Member Between Beam Guide and Screen (4,234,815)
W. Denhollander Combined Linearity and Side Pincushion Correction Arrangement (4,234,824)
W. F. Dietz Synchronous Switched Vertical Deflection Driven During Both Trace and Retrace Intervals (4,234,826)
D. F. Dion and D. R. Wojcik Raster Display Refresh System (4,232,376)
B. W. Faughnan and R. S. Crandall Method for Making Electrochromic Films Having Improved Etch Resistance (4,233,339)
J. S. Fuhrer Defect Compensation for Color Television (4,232,340)
R. A. Gange System for Controlling Brightness Uniformity in Display Devices (4,234,825)
W. H. Groeneweg SECAM-PAL Converter (4,233,622)
P. E. Haferl Regulated Deflection Circuit (4,232,254)
F. Z. Hawrylo and H. Kressel Method of Making a Laser Diode (4,233,090)
W. N. Henry Method of Making a Silicon Diode Array Target (4,231,820)
W. Hinn SECAM Chroma Demodulator Circuit (4,232,268)
W. Hinn and J. Hinderling Combined Kinescope Grid and Cathode Video Drive System (4,233,624)
S. T. Hsu Extended Drain Self-Aligned Silicon Gate MOSFET (4,232,327)
S. Larach and D. H. Vilkomerson Echocardiographic Apparatus for Myocardial Disease Diagnosis by A-Wave Quantification (4,233,989)
A. J. Leidich PNP Output Short Circuit Protection (4,232,273)
R. W. Longsderff and D. V. Henry Alkali Metal Dispenser (4,233,936)
B. B. McCue Internal Metal Stripe on Conductive Layer (4,232,248)
S. Pearlman CRT Screen Structure Produced by Photographic Method (4,234,669)
K. J. Phillips Minimization of Residual Spacecraft Nutation Due to Disturbing Torques (30,429)
E. D. Savoye, T. W. Edwards, and L. F. Wallace Reduced Blooming Devices (4,232,245)
O. H. Schade, Jr. Self-Balancing Bridge Network (4,234,841)
R. G. Stewart Regulated High Voltage Power Supply (4,236,199)
J. A. Strother Interference Suppression for Imaging Optical Systems (4,233,501)
M. Toda and S. Osaka Light Control Device Using a Bimorph Element (4,234,245)
P. P. Webb and R. J. McIntyre Light Detector Housing for Fiber Optic Applications (4,233,619)
J. L. Werst Electron Beam Moving Apparatus for a Color Cathode Ray Tube (4,232,283)
D. H. Willis Regulated Deflection Circuit With Regulator Switch Controlled by Deflection Current (4,234,827)
D. H. Willis High Voltage Disabling Circuit for a Television Receiver (4,234,829)
M. H. Woods Stepped Oxide, High Voltage MOS Transistor (4,236,167)

December

A. A. Ahmed Television Vertical Ramp Generator (4,241,265)
H. Blatter and J. Tults Automatic Turn-Off Apparatus for a Radio or Television Receiver (4,241,450)
A. Bloom and W. J. Burke Ablative Optical Recording Medium (4,241,355)
A. Bloom and R. A. Bartolini Ablative Optical Recording Medium (4,242,689)
W. A. Bosenberg Method of Defining a Photoresist Layer (4,239,790)
W. J. Burke, M. Ettenberg, and H. Kressel Optical Memory With Injection Laser as Light Source and Detector (4,241,423)
D. J. Carlson and J. P. Yu RF Connector Assembly With Low Frequency Isolation (4,242,655)
C. F. Coleman and L. A. Torrington VideoDisc Caddy (4,239,108)
C. F. Coleman Record Extracting Mechanism for Caddy Type VideoDisc Player (4,239,238)
P. Datta and R. N. Friel VideoDiscs and Molding Compositions Therefor (4,241,120)
J. M. David, Jr. Creel (4,240,594)
C. A. Deckert, R. B. Comizzoli, and G. L. Schnable Method for Inspecting Electrical Devices (4,237,379)
N. V. Desai and E. S. Poliniak Silane Electron Beam Resists (4,237,208)
W. F. Dietz Switched Vertical Deflection With Triac (4,238,712)
A. G. Dingwall Integrated Circuit Device Including Both N-Channel and P-Channel Insulated Gate Field Effect Transistors (4,240,093)
A. G. Dingwall Look Ahead High Speed Circuitry (4,242,738)
R. S. Dougherty Aqueous Photoresist Method (4,237,210)
P. D. Griffis Stereophonic Sound Synthesizer (4,239,939)

W. H. Groeneweg Identification System for SECAM or SECAM/PAL Color Television Receivers (4,240,102)

J. R. Harford Signal Overload Prevention Circuit (4,237,490)

R. J. Hollingsworth High Performance Electrically Alterable Read Only Memory (EAROM) (4,237,472)

F. R. Hughes and P. Nyul Electroluminescent Semiconductor Device With Fiber-Optic Face Plate (4,240,090)

R. W. Jebens Solar Powered Dehumidifier Apparatus (4,242,112)

H. C. Johnson Pulse Radar Transmitting Oscillator (4,241,345)

W. F. Kosonocky and R. L. Angle High Speed Loading of Output Register of CCD Array System (4,237,383)

I. Ladany Electroluminescent Diode and Optical Fiber Assembly (4,237,474)

A. J. Leidich Variable Gain Current Amplifier (4,242,643)

D. D. Mawhinney Radio Frequency Pulse Generator (4,238,796)

J. F. McCoy High Resistance Continuous Shield for Reduced Capacitive Coupling in a Deflection Yoke (4,237,438)

W. H. Meise Vertical Sync Separator (4,240,111)

W. H. Moles Vertical Deflection Circuit (4,238,713)

C. L. Olson and J. F. Hacke Television Camera Highlight Discharge Apparatus (4,237,491)

G. D. Pyles Turntable Drive System for VideoDisc Player (4,239,237)

R. D. Rhodes Digital Phase Detector (4,237,423)

W. R. Roach and P. Sheng Diffractometer for Measuring Signal Depth and Width (4,236,823)

T. J. Robe Bandwidth Limited Large Signal IC Amplifier Stage (4,240,042)

A. Rosen, A. M. Gombar, and E. Mykietyn Method for Fabricating Stacked Semiconductor Diodes for High Power/Low Loss Applications (4,237,600)

J. O. Schroeder Pulse Modulator for Operating a Device in the Avalanche Mode (4,242,644)

A. Schwarzmann Phase Shifter (4,238,745)

R. G. Stewart Asymmetrically Precharged Sense Amplifier (4,239,994)

C. E. Tracy, W. Kern, and R. D. Vibronek Sprayable Titanium Composition (4,241,108)

W. F. Wedam Horizontal Deflection and Power Supply Circuit With a Start-Up Arrangement (4,240,013)

P. K. Weimer Charge Transfer Imager (4,237,477)

P. K. Weimer Line-Transfer CCD Imagers (4,242,700)

D. H. Willis Regulated Deflection Circuit (4,240,012)

R. W. Zborowski Parallel Operation of Multiple TV Transmitters (4,238,855)

AUTHORS

Kern K. N. Chang received his BS degree from National Central University, Chungking-Nanking, China, in 1940, his MS degree in Electrical Engineering from the University of Michigan in 1948, and his PhD degree from the Polytechnic Institute of Brooklyn, New York, in 1954. Since 1948, Dr. Chang has been a Member of Technical Staff at RCA Laboratories, Princeton, New Jersey. From 1962 to 1971, he was Head of Solid State Microwave Devices. In 1967, he was appointed a Fellow of RCA Laboratories. He has been engaged in research on TV receivers, magnetrons, traveling-wave tubes, beam-focusing devices, parametric amplifiers, tunnel-diode devices, Hall effect solid-state devices, bulk light-emitting sources, avalanche diodes, and display devices. His current work is on TV kinescope research. His significant technical contributions have included a pioneering effort on periodic field focusing that culminated in an RCA commercial line of traveling-wave tubes, original work on parametric and tunnel diodes as amplifiers and converters for military and industrial applications, and discovery of a new mode of high-power and high-efficiency operation of avalanche diodes for radar application.

Dr. Chang holds 33 U.S. patents in the field of microwave tubes and solid-state devices and is the author of the book *Parametric and Tunnel Diodes* (Prentice Hall, Inc.). He was the recipient of three RCA Laboratories Achievement Awards (1955, 1960, and 1972) for outstanding theoretical and experimental research on electron-beam focusing, on parametric and tunnel-diode devices, and on avalanche diodes. He was also the 1964 Achievement Award winner of the Chinese Institute of Engineers, New York, Inc., for his outstanding contribution to the field of electron devices. In 1967 he received the David Sarnoff Outstanding Achievement Individual Award in Science. Dr. Chang is a member of Sigma Xi.

Edward O. Johnson attended U.S. Naval Electronics schools in Texas and later was with the Pacific carrier fleets in communications, navigation, and radar electronics during World War II. He received his BSEE from Pratt Institute in Brooklyn, N.Y., in 1948 and later had graduate courses in physics and engineering at Princeton University and the Swiss Federal Institute of Technology in Zurich. After joining RCA Research Laboratories at Princeton in 1948 he did research in gaseous electronics and semiconductor devices and surfaces. In various managerial capacities, from 1959 to 1972 he was with the Solid-State and Tube Product Divisions and from 1973 to 1975 with Licensing Operations. From 1975 to the present he has been director of the RCA Research Laboratories in Tokyo, Japan.

In 1953 with W. M. Webster he received the IEEE Editor's Award, and in 1960 the Eta Kappa Nu Recognition. In 1966 he was a member of the U.S. Department of Commerce Panel on Electronic Automobiles. He has approximately 30 patents.

Edward W. Maby received tne S.B. and S.M. degrees in Electrical Engineering and the S.B. degree in Physics in 1975 and the Ph.D. degree in Electrical Engineering in 1979, all from the Massachusetts Institute of Technology, Cambridge, Mass. From 1972 to 1979, he worked summers at RCA Laboratories in Princeton, N.J., becoming a member of the technical staff in 1979. While at RCA Laboratories, he worked on ion implantation problems for semiconductor research and development. Since July 1980, he has been a research associate in the Department of Electrical Engineering and Computer Science at M.I.T.

Dr. Maby is a member of Sigma Xi and of the American Physical Society.

N. A. Macina holds two MS degrees, one in Civil Engineering from the University of Rome, Italy, (1942), the other in Electrical Engineering from the Polytechnic Institute of Brooklyn, (1969). With RCA since 1954, Mr. Macina is a telecommunication system specialist with extensive experience in signal design, modulation, coding, and discrete-time signal processing. In recent years, as a principal member of the engineering staff of the Advanced Communications Laboratory, he has been primarily involved in applications of advanced digital filtering and spectral analysis techniques to communications, to feature extraction for classification, and to smart sensor and detecting devices.

Glenn A. Reitmeier obtained his B.E.E. degree summa cum laude from Villanova University in 1977, and received the Dean's Award for Academic Excellence. He joined RCA Laboratories in 1977, and received his M.S.E. in Systems Engineering from the University of Pennsylvania in 1979. Since joining the Laboratories, Mr. Reitmeier has been performing research in the field of digital television, particularly in the areas of multidimensional signal processing and the development of system concepts for digital video tape recording and for digital television studios. He received an RCA Laboratories Achievement Award in 1979 for his contributions to digital television. Mr. Reitmeier is currently the Secretary of the Working Group on Digital Video Standards in the Society of Motion Picture and Television Engineers, and is a member of Tau Beta Pi, Eta Kappa Nu and Phi Kappa Phi.

Soitiro Tosima received the B.Sc. degree in physics from Kyushu University in 1951, and the D.Sc. degree in physics from Osaka University in 1962. From 1953 to 1961, he was a Research Associate in the Physics Department of Kyushu University. He joined RCA Research Laboratories, Tokyo, in 1961. In 1964, he spent six months as a visiting research scientist at RCA Laboratories in Princeton. In 1965, Dr. Tosima became manager of the Applied Physics Group. His principal areas of interest have been the theories of transport phenomena in semiconductors, semimetals, and ferromagnetic semiconductors, particularly at high electric fields, instabilities in solid state plasmas, and magnetic bubbles. His present areas of interest are high intensity surface acoustic waves and the physics of the soldering process.

Dr. Tosima is a member of the Physical Society of Japan.

Chung P. Wu received the B.S., M.S., M.Phil., and Ph.D. degrees in Physics from Yale University in 1965, 1966, 1967, and 1968, respectively. He was a research staff physicist at the Yale Electron Linear Accelerator Laboratory from 1968 to 1970, working on photonuclear reactions and neutron time-of-flight spectroscopy. From 1970 to 1972 he was an Assistant Professor of Physics at Nanyang University in Singapore. Since 1973, Dr. Wu has been a Member of Technical Staff at RCA Laboratories, Princeton, N.J. He has worked on different techniques for determining the electrically active ion-implanted doping profiles, the utilization of ion implantation in the fabrication of semiconductor devices, and the characterization and evaluation of such devices. He is currently working on the laser annealing of semiconductor materials, including polysilicon, crystalline silicon, SOS, and GaAs, for the fabrication of solar cells, devices, and circuits.

Dr. Wu has more than 20 technical publications and 15 patents granted or pending. He is a member of the American Physical Society, IEEE, and Sigma Xi.

STATEMENT OF OWNERSHIP

Statement of Ownership, Management and Circulation (Act of August 12, 1970, Section 3685, Title 39, United States Code).

1. Title of Publication: RCA REVIEW. 2. Date of Filing: October 3, 1980. 3. Frequency of Issue: Quarterly. 4. Location of Known Office of Publication: RCA—David Sarnoff Research Center, Princeton, Mercer County, New Jersey 08540. 5. Location of Headquarters or General Business Offices of the Publishers: RCA—David Sarnoff Research Center, Princeton, Mercer County, New Jersey 08540. 6. Names and Addresses of Publisher and Editor: Publisher, RCA Corporation, Research & Engineering, Princeton, New Jersey 08540. Editor, Ralph F. Ciafone, RCA, David Sarnoff Research Center, Princeton, New Jersey 08540. 7. Owner: Cede & Company, Box 20, Bowling Green Station, New York, New York 10004; Kray & Company, 120 S. LaSalle Street, Chicago, Illinois 60603; Pacific & Company, Box 7877, San Francisco, California 94120. 8. Known Bondholders, Mortgagees, and Other Security Holders: Metropolitan Life Insurance Company, 1 Madison Avenue, New York, New York 10010; New York Life Insurance Company, 51 Madison Avenue, New York, New York 10010; Connecticut General Life Insurance Company, Hartford, Connecticut 06115; The Travelers Insurance Company, One Tcwer Square, Hartford, Connecticut 06115; John Hancock Mutual Life Insurance Company, Hancock Place, Boston, Massachusetts 02117. 10. Extent and Nature of Circulation: Average Number of Copies Each Issue During Preceding 12 Months, and of the Single Issue Nearest to Filing Date, respectively, are as follows: (A) Total Number of Copies Printed (Net Press Run) 3,815–3,632; (B) Paid Circulation—Mail Subscriptions 3,138–3,003; (C) Total Paid Circulation 3,138–3,003; (D) Free Distribution by Mail, Carrier or Other Means (including Samples, Complimentary and Other Free Copies) 131–139; (E) Total Distribution 3,269–3,142; (F) Office Use, Left Over, Unaccounted, Spoiled After Printing 546–490; (G) Total 3,815–3,632.

I certify that the statements made by me above are correct and complete.

Ralph F. Ciafone, Editor