XIV. COGNITIVE INFORMATION PROCESSING[*]

Academic and Research Staff

Prof. M. Eden
Prof. J. Allen
Prof. F. F. Lee
Prof. S. J. Mason
Prof. W. F. Schreiber
Prof. D. E. Troxel

Prof. T. S. Huang
Dr. R. R. Archer
Dr. W. L. Black
Dr. J. E. Green
Dr. K. R. Ingham
Dr. O. J. Tretiak
F. X. Carroll

C. L. Fontaine
E. R. Jensen
A. J. Laurino
D. M. Ozonoff
E. Peper
Sandra A. Sommers

Graduate Students

G. B. Anderson
T. P. Barnwell III
W. L. Bass
B. A. Blesser
J. E. Bowie
A. E. Filip
A. Gabrielian
A. M. Gilkes
R. E. Greenwood
E. G. Guttmann

R. V. Harris III
D. W. Hartman
A. B. Hayes
P. D. Henshaw
M. Hubelbank
W-H. Lee
J. I. Makhoul
G. P. Marston III
G. F. Pfister

D. S. Prerau
G. M. Robbins
S. M. Rose
C. L. Seitz
D. Sheena
W. W. Stallings
R. M. Strong
Y. D. Willems
J. W. Woods
I. T. Young

A. PAGE READER FOR A READING MACHINE FOR THE BLIND

1. Introduction

For several years, our group has worked on the development of a reading machine for the blind. The problem involves the transformation of ordinary printed text into a form that can be sensed by a blind person. The first reading machine system, which was demonstrated in the spring of 1966, was capable of reading a single line of text, recognizing the characters and punctuation, and spelling out these symbols aurally. This experimental system was not intended to be a prototype of a practical reading aid, but rather was a real-time research facility for the experimental investigation of human information requirements and human learning capabilities upon which prototype designs must be based. This experimental system has facilitated research on print scanning, character recognition, control by a blind reader, and auditory displays.[1-5]

One of the crudest features of the initial reading machine system was the opaque scanner, and work commenced immediately to develop an improved scanner and carriage control that would handle whole pages of print rather than individual lines. During the spring of 1968, funds were made available for the purchase of a PDP-9 computer which was to be used primarily for reading machine development. This computer has

formed the nucleus of a second research version of a reading machine for the blind which, while still not a prototype, has considerably improved capabilities and represents a significant step toward the eventual realization of a practical machine.

2. System Description

The system diagram for the second version of the reading machine is shown in Fig. XIV-1. The transformation of a printed page into Braille, spelled speech, or synthesized speech has been demonstrated with this system.
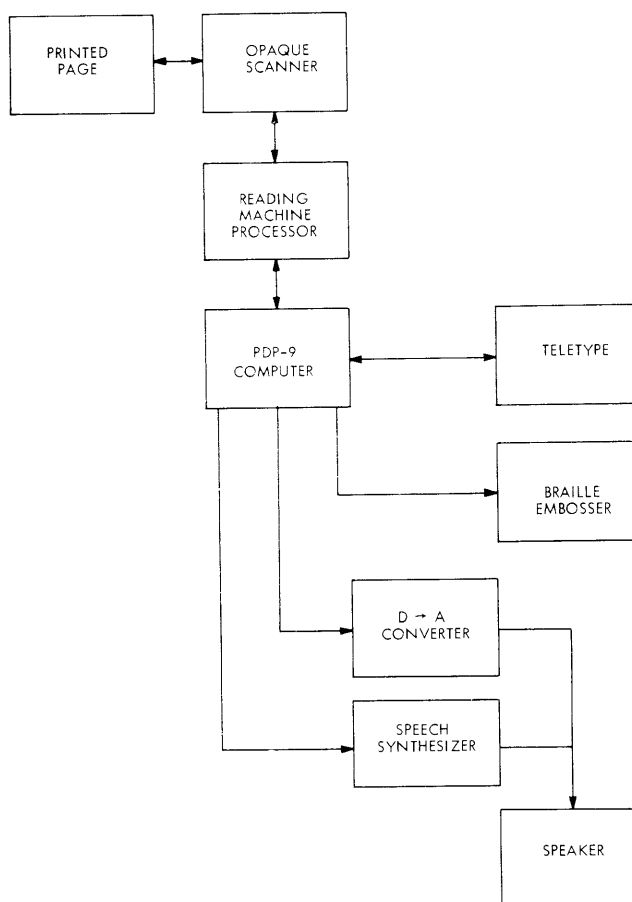
Fig. XIV-1. Reading machine system diagram.

This report is concerned primarily with the page-reading process or the transformation of the printed page into a sequence of character codes that can be further processed to make the information available to a blind person in the form of Grade II Braille, spelled speech, or synthesized speech. The output of the page-reader portion of the reading machine can be typed on a teletype so that sighted persons may conveniently

monitor the performance of the page-reading procedures.

The printed pages used thus far have been removed from a Fourth Grade reader, entitled "Roads to Everywhere," published by Ginn and Company, Boston, 1964 and printed in 14 point TEXTYPE type. Only those pages that do not contain pictures have been used.

The opaque scanner is a flying-spot scanner in which a dot of light positioned on the face of the cathode-ray tube (CRT) is imaged on the page. The total light reflected from the page is sensed by photomultiplier tubes, the output of which is a function of the whiteness of the particular spot on the page. To achieve the necessary resolution, the face of the CRT is imaged onto approximately 1 sq. in. of the page. Access to the whole area is provided by a carriage that is integral to the opaque scanner and permits both horizontal and vertical motion of the page. Both the carriage and opaque scanner are operated by the reading-machine processor (RMP), which in turn receives directives from the computer. This RMP in effect extends the instruction set of the computer and allows various complex functions to be initiated by the computer and to be performed while the computer is engaged in other calculations. One such directive that is often used is, "Given that the dot of light on the CRT is imaged on the contour of a character, find the next sequential contour point and return its coordinates to the computer." It was deemed necessary to provide these functions in the hardware, rather than to perform them by computer program, so that recognition rates could approximate human speaking rates.

The recognition of a alphanumeric characters is a modification and extension of procedures developed by Clemens,[6] in which a number or signature is computed from the exterior contour of a character, and the character is "recognized" by finding a match in a previously generated table of signatures and character codes. The sequence of operation of the page reading programs is indicated in Fig. XIV-2. The first step is page initialization where the optimum threshold for the discrimination of black and white on the page is computed. The size of the type face on the page is determined, and the left and upper boundaries of print on the page are located. Starting from the top of the page, the first line of print is acquired, and then the first character in this line is acquired. Then the exterior contour of the letter is traced; and if the character is deemed large enough, a signature is computed and the signature table is searched to yield the code for the character. Then an attempt is made to acquire the next character. Eventually all of the characters on that line will be processed and no such character will be available, thereby resulting in an end-of-line determination. When this occurs, the program attempts to acquire the next line of print. If no such line is found, then the page is considered finished.

If, after a character has been traced, it has been deemed small enough to be punctuation, control is transferred to a special routine that recognizes the punctuation mark
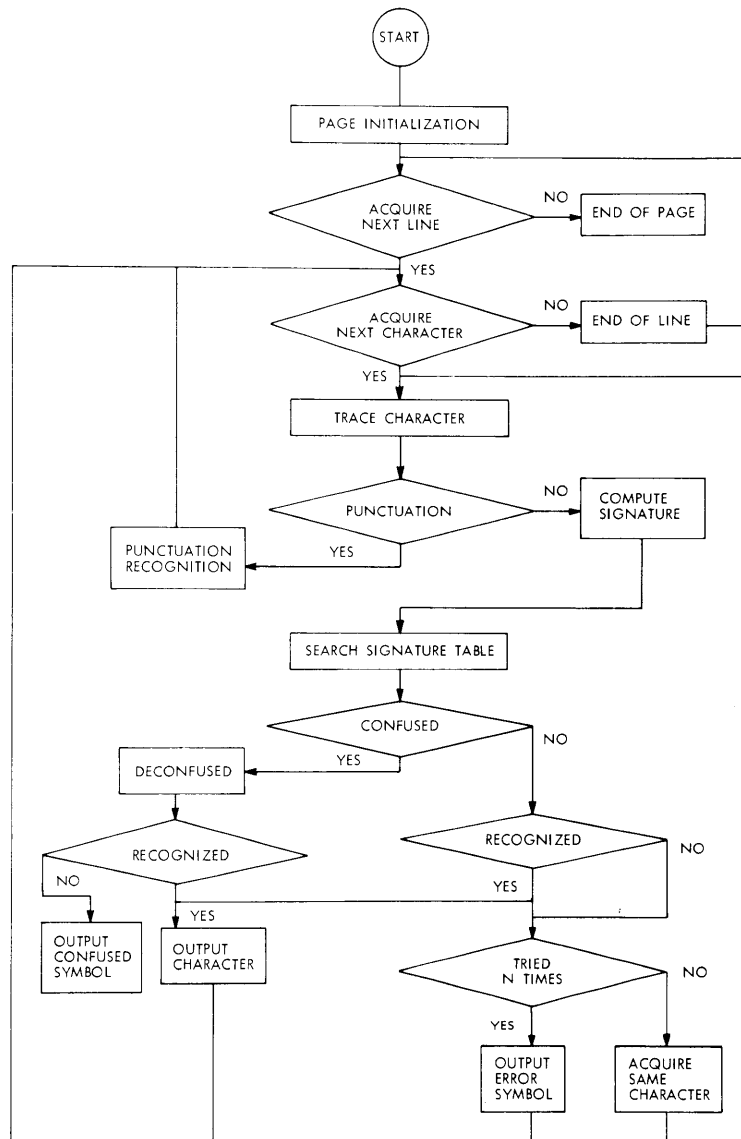
Fig. XIV-2. Page reader flow chart.

and proceeds to acquire the next character.

As seen in the lower portion of Fig. XIV-2, there are three possible outcomes of a signature table search. The most fortunate of these is that the signature is uniquely associated with a particular character. Some shapes, however, that correspond to two different characters result in the generation of identical signatures. Fortunately, there are not very many occurrences of these confused signatures, and it has been possible to resolve these confusions by making auxilliary tests. If a particular signature just generated does not exist in this signature table, then the character is reacquired and traced again. This procedure of trying again appears to be applicable because the

statistics of successive tracings of the same character seem to be similar to the statistics of successive tracings of like characters.

3. Page Initialization

Page initialization performs three main functions: the determination of the optimum threshold for subsequent black-white decisions, the determination of the size of the type face on the page, and the positioning of the carriage for subsequent acquisition of text. As the scanner can only access approximately 1 sq. in. of the page area, it is first necessary to locate the carriage so that this aperture contains some print. In the present system, this is accomplished simply by moving the aperture to the center of the page; then a number of A to D conversions are made over this central portion of the page, and the optimum threshold is computed from a histogram of the results of these A to D conversions, as indicated in Fig.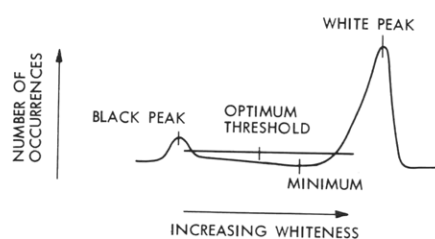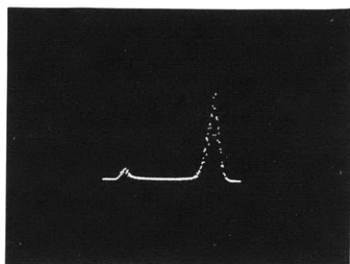 XIV-3. The two peaks of this histogram corresponding to the most common black and the most common white are found. Then the minimum of the central region of the histogram as bounded by these peaks, is determined. The optimum threshold is then computed as the midpoint of a horizontal line drawn a fixed distance above this minimum and bounded by the interior sections of the histogram.

The font size is characterized by the x height, which is the height of the lower-case x. This x height is determined by tracing the contours of a number of characters and is calculated as the average height of the smallest characters, provided that there exists one character which is substantially taller than the smallest character. Care is taken, of course, that specks are not classified as characters.

Fig. XIV-3. Optimum threshold determination.

The left print boundary is determined by moving the carriage so that the left edge of the aperture is quaranteed to be black (i. e., off the page). Then the aperture is moved right until a substantially all white area is found. The motion to the right is continued until a substantial amount of black (the left edge of the print) is found. The aperture is then moved so that the print boundary is near the left edge of the aperture, and the accuracy of this move is checked.

The top print boundary is found in a similar manner, but the top line of print is positioned in the center of the aperture.

4. Line Acquisition

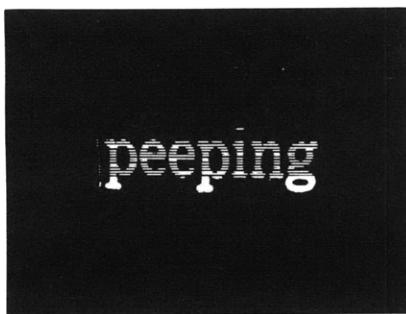Line acquisition makes extensive use of the horizontal histogram (HH) directive, which is issued by the computer and executed by the RMP. When this directive is initiated, the RMP interrogates all points at the specified y value and returns the total number of black points encountered. The heuristic used for the acquisition of a line involves the execution of the HH at selected y values. At first, the search is coarse, that is, with a significant space between successive y values. When a significant number of black points is encountered, the HH is executed approximately every fourth y line (see Fig. XIV-4a), and the maximum number of black points is computed. When the number of black points resulting from an HH is less than one-eighth of the maximum encountered thus far, the base line of the line of text has been passed; and a fine search is performed but for increasing values of y until the number of black points is greater than one-half of the maximum previously encountered (see Fig. XIV-4b). This heuristic has resulted in reliable determination of the base line of a line of text, whether or not it contains letters with descenders.

For second and subsequent lines, the motion of the aperture of the scanner is effected to place the next line approximately in the center of the aperture, as well as to return the aperture to the left edge of the print. An end-of-page determination is made if no line can be found within the scanner aperture.

4a. SEARCH EVERY FOURTH LINE

4b. ADDITION OF FINE SEARCH UPWARD

Fig. XIV-4. Line acquisition.

5. Character Acquisition

Character acquisition again makes use of another directive executed by the RMP. In this directive, a vertical scan is performed between two y values, starting at a particular x value. If a black point is not encountered, then the vertical scan is performed again but indexed to the right in x. This continues until a black point is encountered or the value of x equals that of the x mask. This scan is indicated in Fig. XIV-5. The values of the upper and lower y limits, the starting value of x and the x mask are transmitted to the RMP before the execution of the acquisition scan. Before the acquisition scan is initiated, the aperture is moved if necessary. The x mask is positioned so as to

guarantee that if a character is found before reaching the x mask the character will be fully within the scanner aperture. The contour of the character is then traced. If the extent of the character is small, it is deemed a speck, and the acquisition scan is
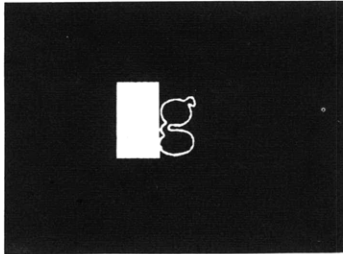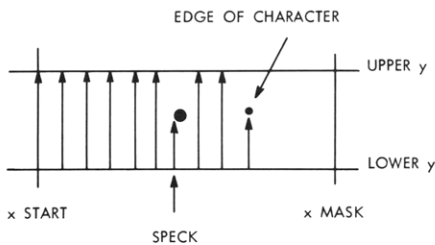

Fig. XIV-5. Acquisition scan.

restarted to the right of that speck. The end-of-line determination is made if the acquisition scan has proceeded for a certain value of x with no more than specks being found. When the first character of a line has been found, the left edge of this character is compared with the left edge of the print on the page to see if a paragraph has been encountered. For succeeding characters, the distance between the left edge of the newly acquired character and the right edge of the previously acquired character is used to determine whether or not a space or spaces should be transmitted.

6. Contour Tracing

The fact that the exterior contour of a letter is traced has often been considered a key feature of the character recognition algorithm used. Contour tracing has little to do, however, with the character recognition algorithm, as it is primarily a means of defining the precise location of the character to be recognized. It is, however, a relatively fast technique for extracting the useful information from a fairly large area if one is confined to an optical transducer that can test only a single point at a time. A character in a 50 × 100 grid would require 5000 tests if every point on the grid were to be examined. It would, however, have only approximately 300 edge points, which would require approximately 600 tests.

There are many ways in which contour tracing can be implemented.[3] Perhaps the

simplest digital method is the square trace (see Fig. XIV-6), which can be described by the following rule. If the last point tested was black (white), then make a 90° left (right) turn to determine which point will be tested next. This contour-trace algorithm
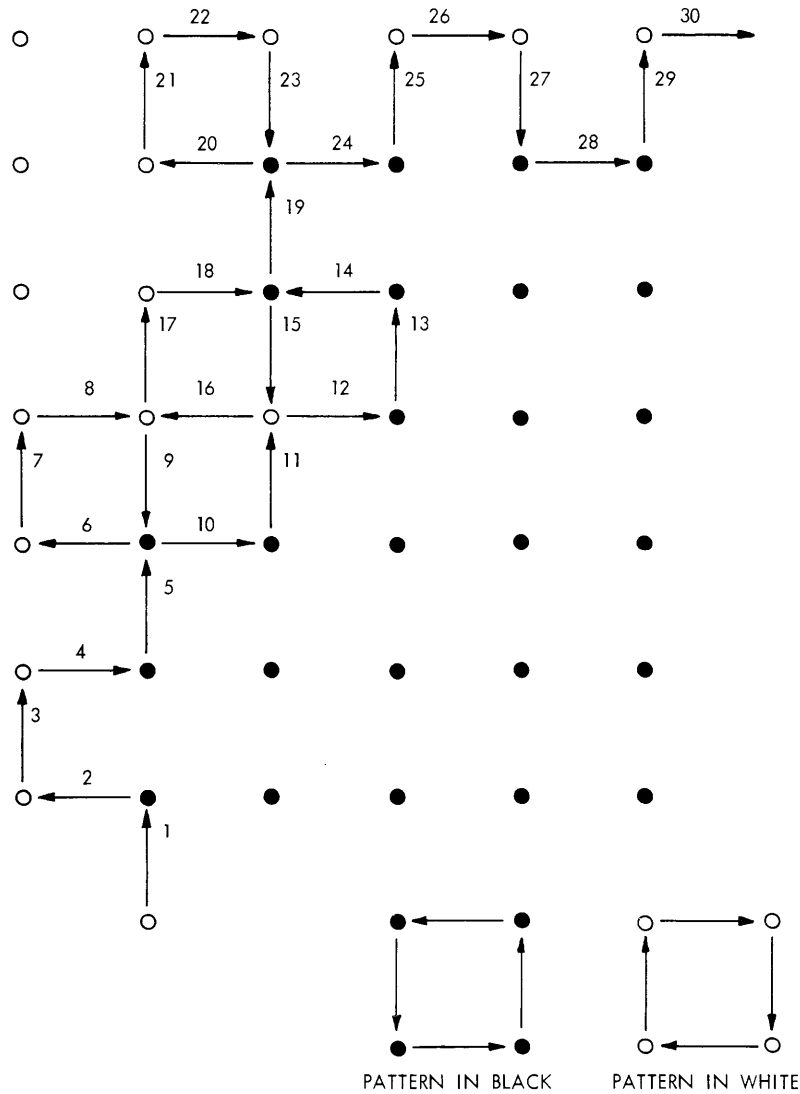

Fig. XIV-6. Square trace.

works well on fixed data. As a matter of choice, the interrogated points are along the edges of a letter. These, of course, are the very points that are on the borderline of being black or white. If there is even a small amount of noise present in the optical scanner, successive interrogations of the same point are likely to reveal different answers.

Because of the square-trace algorithm's propensity to test the same point twice, it is possible to become trapped in black or white. A variation of the square-trace algorithm, which is shown in Fig. XIV-7, eliminates the possibility of becoming trapped

Fig. XIV-7. Modified square trace.

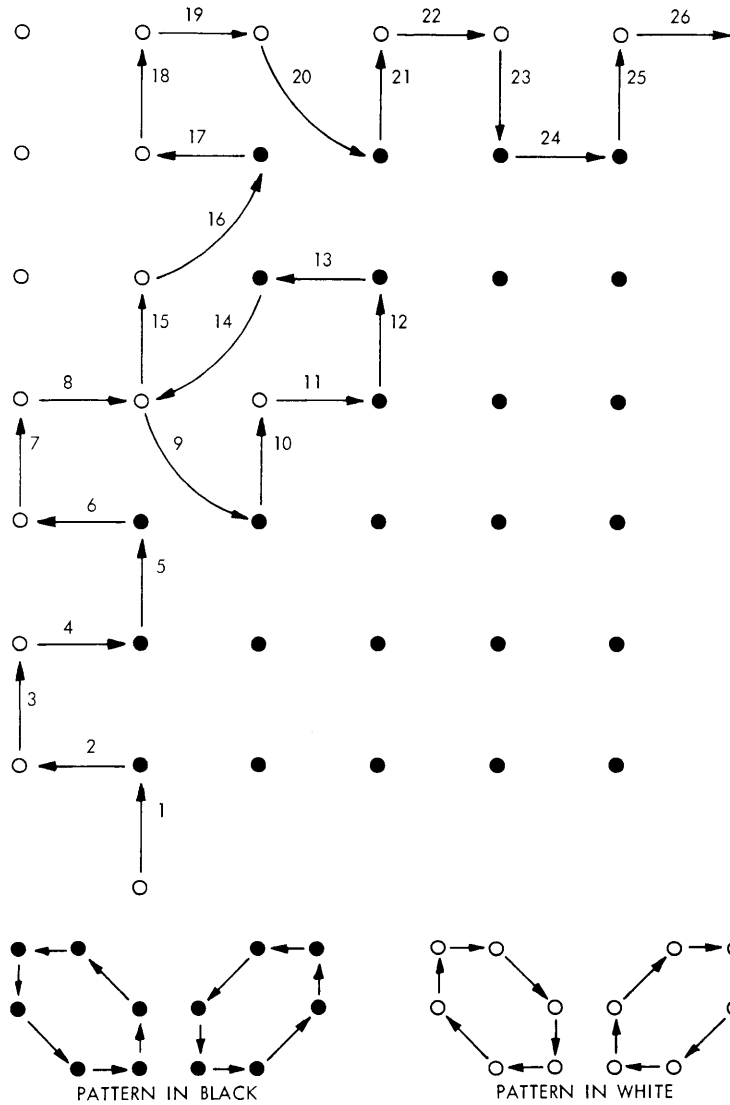in black or white when the spatial quantization and grayness of the letter edges are such that the picture elements on the borderline between being black or white are less than two coordinate units wide. In general, this variation minimizes the square-trace algorithm's tendency to test the same point twice by forcing diagonal moves when the three previous points were either all white or all black.

As is indicated in Fig. XIV-7, it is possible for even the second algorithm to become trapped in black or white. The trapped condition is detected by the occurrence of 6 successive black (white) points and is termed a contour-trace error. As the contour-trace algorithm is realized in hardware by the RMP, the contour-trace subroutine merely has to ask for the next contour point. Of course, the subroutine must determine when to terminate the tracing of a character. This is accomplished by a distance calculation of the present contour point relative to the initial contour point. The distance measure that is used is the maximum of the magnitude of the displacement in x or y. To minimize the time taken for the contour trace, this distance calculation is made only as often as is necessary to be sure that the original contour point has not been bypassed. Thus, the
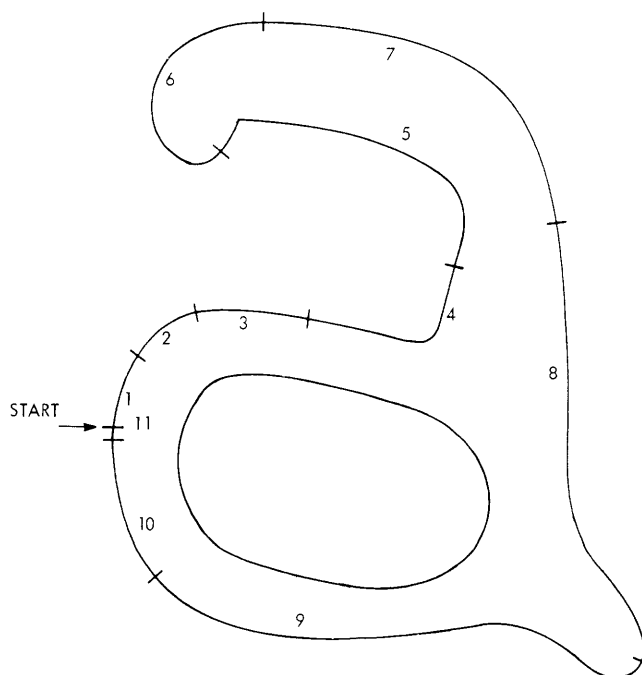
Fig. XIV-8. Segmented contour trace.

contour of a character is traced in segments as indicated in Fig. XIV-8, where the length of the perimeter of the next segment is determined by the distance measure of the present contour point to the initial contour point. Each time the distance calculation is made, a check is also performed to see if any contour-trace errors have occurred on the previous segment; and if such is the case, the contour trace is restarted at the end of the previous segment. The shape whose contour has been traced is considered a speck if the number of contour points is below a fixed threshold or if the initial distance calculation is less than a constant.

7. Punctuation Recognition

If the number of contour points of the shape as traced by the contour-trace routine is smaller than a constant, then the character is considered to be a punctuation mark. Of course, this constant could easily be computed as a function of the type-face size. The allowable punctuation alphabet is shown in Fig. XIV-9, which also indicates the sequence of decisions performed.
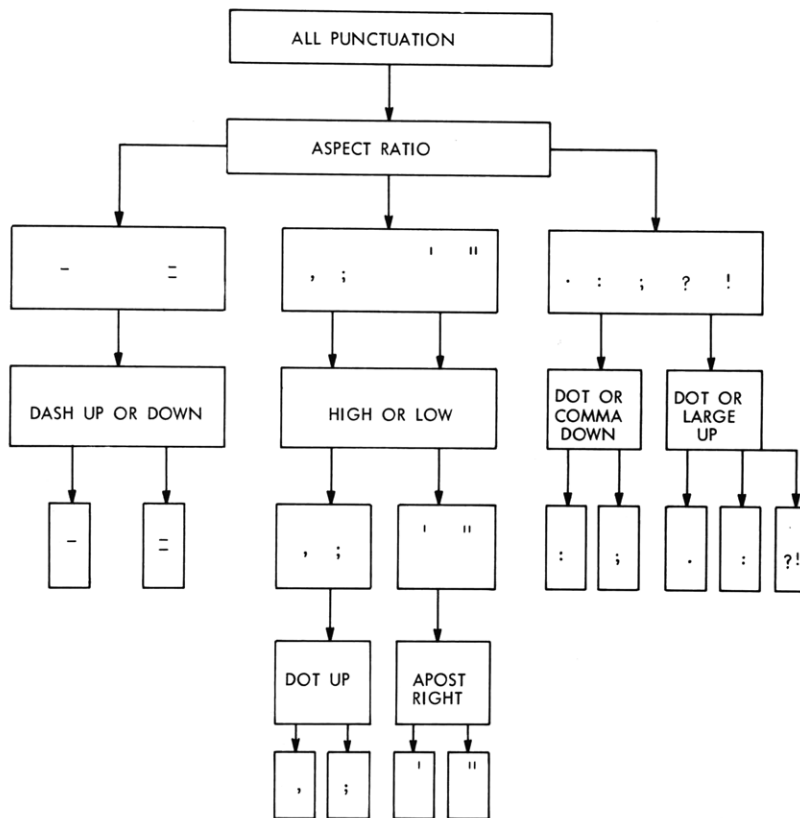
Fig. XIV-9. Punctuation decision tree.

The first decision is based upon the aspect ratio of the shape to divide it into three classes. A further discrimination between a dash and an equality sign is made by searching areas just above and below the shape for the existence of another dashlike character. It is conceivable that a shape that is not dashlike in its aspect ratio could be found, and this would be interpreted as an error. Those punctuation marks with high aspect ratios are further subdivided by their vertical position into two more classes, and these classes are resolved by searching appropriate areas for possible additional pieces of the punctuation mark. Punctuation marks having aspect ratios near unity are
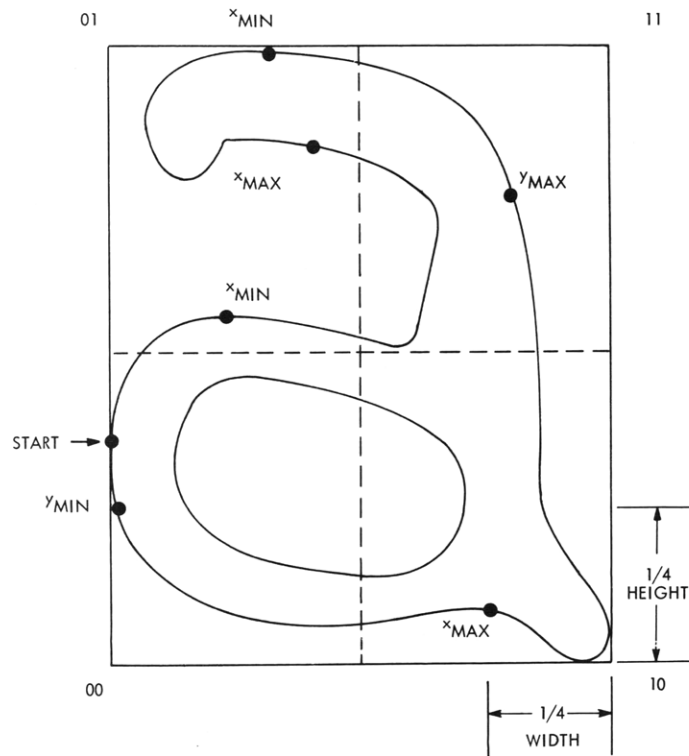
START = 1
x = 1
y = 0

	CODE WORD:	1	1	1	0	1	0
	COORDINATE WORD:	01	01	01	11	10	00

Fig. XIV-10. Determination of extrema.
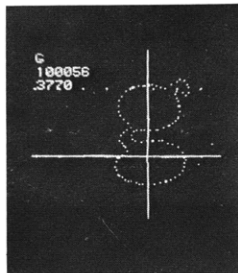
Fig. XIV-11. Training mode display.
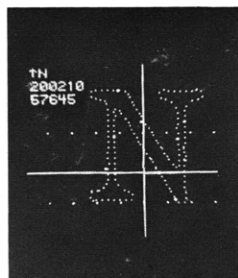
similarly further classified by appropriate searches above and below the shape. The existence of a large shape above a dot is taken to indicate the presence of a question or exclamation mark, and these larger shapes are treated in exactly the same manner as alphanumeric characters.

8. Character Recognition

The basic scheme for character recognition is the same one that was used in the first reading machine system and is described in detail by Clemens[3] and Seitz.[6] After the contour of a letter has been traced, a characteristic number or signature is generated and a table of previously encountered signatures is searched for a match. The signature is computed by the method indicated in Fig. XIV-10.

The starting point is always at the southernmost point. Local extrema are determined for the horizontal and vertical directions if the excursions along the contour are greater than one-quarter of the letter height or width. That is, for example, an x maximum is not denoted unless one had to travel east for at least one-quarter of the letter width and then west for one-quarter of the letter width. With this scheme, minor features of the letter such as the droop in the upper left part of the letter a do not introduce extraneous extrema. The sequence of extrema can be coded with one bit per extremum if the first extremum recorded is an x minimum and the contour trace is clockwise. In addition to this sequence of extrema, the quadrant in which it was determined is stored for each extremum. These binary sequences, along with a small amount of information concerning the aspect ratio of the letter and its position relative to the y lines found by the line acquisition procedure, comprise the signature.

In order for characters to be recognized, the particular signature must have been encountered previously; that is, the machine must have been previously trained by a sighted operator. To enable the sighted operator to recognize the letters, a visual display (see Fig. XIV-11) is made. When the machine is unsuccessful in the search for a match for a particular signature, it can ask the trainer what the character is. Given this information, it can thus add to its experience. There is a certain amount of noise involved in the determination of black and white, and the presence of this noise actually simplifies the training procedure, as it appears that successive tracings of the same a have variations similar to tracings of different a's. Thus a character repeat mode is provided in which one can lock on to a particular letter and build a repertoire of signatures for that particular letter.

9. Multiple-Font Recognition

There are three aspects of the page-reader programs which are peculiar to a specific font. These are the signature table, and the punctuation and deconfusion routines. Separate signature tables for any desired font can be generated easily by use of the

training mode. The deconfusion routines that may be required must be programmed to resolve confusions generated during the training process. Punctuation recognition routines are probably quite universal because punctuation marks have small variations among most fonts. No provision has yet been made for truly multiple-font recognition, i. e. , when a single page contains a mixture of fonts.

10. Performance

Ten successive readings of the same page were made in order to test the accuracy of the page-reading process. The page that was read comprised 1087 characters, including letters, punctuation, spaces between words, end-of-line determinations, paragraph determinations, and an end-of-page determination. A total of 12 errors was made for a gross error rate of approximately 0.11 per cent. Four of the errors dealt with spaces, either missing or added. A program "bug" has been found and corrected, so that this kind of error no longer occurs. Two punctuation errors occurred in which a single quote was substituted for a double quote. There were 4 letter-substitution errors and 2 letters not recognized. Virtually all of the mistakes concerned with letters resulted from broken or touching letters. If we discount the space-related errors, the error rate was approximately 0.07 per cent. People are normally accustomed to reading text with considerably higher error rates.

The speed attained by the page reader is adequate for Braille or spelled-speech output. It is, however, somewhat marginal for synthesized speech output. The average speed is 75.5 wpm. A large amount of the time spent in reading is "wasted" while the computer is awaiting completion of carriage motions. If we subtract the time spent waiting for completion of carriage motions, the reading speed would be 119 wpm.

D. E. Troxel

References

1. D. E. Troxel, F. F. Lee, and S. J. Mason, "Reading Machine for the Blind," Quarterly Progress Report No. 89, Research Laboratory of Electronics, M. I. T., April 15, 1968, pp. 245-248.

2. D. E. Troxel, F. F. Lee, and S. J. Mason, "Reading Machine for the Blind," Digest of the 7th International Conference on Medical and Biological Engineering (Stockholm, 1967).

3. C. L. Seitz, "An Opaque Scanner for Reading Machine Research," S. M. Thesis, M. I. T., 1967.

4. D. E. Troxel, "Page Position Selection for Use with a Reading Machine for the Blind," Digest of the 8th Annual IEEE Symposium on Human Factors in Electronics (Palo Alto, California, 1967).

5. D. E. Troxel and G. F. Pfister, "Optimum Threshold Level for the Reading Machine Opaque Scanner," Quarterly Progress Report No. 86, Research Laboratory of Electronics, M. I. T., July 15, 1967, pp. 303-305.

6. J. K. Clemens, "Optical Character Recognition for Reading Machine Applications," Ph. D. Thesis, M. I. T., 1965.

B. OUTPUT DISPLAYS FOR THE M. I. T. READING MACHINE

As part of the continuing development of auditory and tactile displays for the M. I. T. Reading Machine system, a new series of Grade 2 translators has been implemented. Following the completion of the combined Mathematics and Literary Braille translators early this year, the need to separate the various modes has arisen. Therefore a software loader and command program has been written which permits the operator to request from dectape storage the binary programs that accept teletype input or scanner input and perform the appropriate translation of text into Mathematics or Literary Braille.

The Literary Grade 2 translator requires no training on the part of the typist. This feature was necessary because a scanner input mode could not be expected to do more than transmit the line information and paragraphing, as well as the full alphanumeric set. The only rule that the operator must follow is that of indenting paragraphs with a tab. The nature of Braille rules requires that a line of Braille be accumulated before embossing, and this rule causes the program to complete the last typed line.

The program, which utilizes a computational dictionary approach and translates the input string from the right, produces Braille contractions close to those of authorized Braille. The chief difficulty with machine translation is the so-called syllabification rule which requires that Braille contractions not be used across syllable boundaries. The relatively infrequent violations of the rule that occur prove, however, to be quite acceptable in reading tests carried out with students at Perkins School for the Blind. All of these programs drive a modified IBM Model D typewriter whose keys have been replaced by Braille characters. This machine is driven by a robothead solenoid pack, mounted above the keyboard at a rate of 8 characters per second. Tests are under way to improve this speed.

It is also possible to replace the Braille program by Spelled Speech programs whose presentation rate is approximately 120 words per minute. All of these output modes may be used to display text from the PDP-9 systems editor.

A set of sounds for the alphanumerics of the standard keyboard is under development whose waveforms are generated with the aid of the speech synthesis utilized by the "synthesis-by-rule" programs. Since the generation of speech by rule in our laboratory is accomplished by combining phonemic elements with linear interpolation for the various parameters, the speech cannot be expected to be of highest quality. The synthesis test program described here permits the operator to insert from console switches any of the relevant parameters while listening to the effect. Each of the formant frequencies and amplitudes, voicing or hiss, may be individually selected and changed at any of the 10-msec output intervals. Once a parameter has been selected, the accumulator test switches may be used to determine the time instant at which the parameters should be

varied. By selecting different bit positions, a contour for the duration of the sound may be generated. It is expected that handmade sounds produced in this manner and tested by the operator will result in utterances whose waveforms include the higher order approximations. Thus two sets of utterances, synthesized speech and PCM sampled speech, will be available for comparison and psychophysical tests. The greatly decreased storage requirements of synthetic speech should lead to the use of such speech as computer output, provided it can approach the quality of compiled speech.

A read-only rote memory has been completed whose data store comprises more than 90 different speech sounds having durations of 100 msec each, which are to be used in the computer-controlled output of alphanumerics and mathematics. The memory contents is PCM sampled speech sounds compressed to an average of one-third of their normal durations. Furthermore, a set of tables is stored which permits the off-line indirect addressing of the speech with ASCII, EBCDIC, and flexocodes. Thus, with electronics components under construction, this device will provide Braille programmers with an off-line readout for cards and paper tape. At present, this device is being installed on the PDP-1 computer for use as a time-shared voice output. Because of the memory cycle time and the duration of the stored sounds, as many as 40 users may simultaneously receive output of different utterances from this device.

Telephone data sets have been acquired and small portable keyboards attached to them in order to permit handicapped users, such as the blind, to utilize the power of a computer at their place of employment. The present system that is being implemented provides for two simultaneous users, and will provide the capability for research in three areas associated with information retrieval and rehabilitation of the handicapped. One of these is the development of suitable local stations, keyboards, and the like. The second would concern itself with the type of software to be made available to the blind user. For example, a linking editor might make possible the employment of blind stenographers in a courtroom setting. The third area of research is that of the development and structuring of limited vocabulary voice output tailored to the vocational area that is being explored. A blind programmer should achieve a high measure of independence with a spoken set of keyboard characters, while a court stenographer might require an entirely different set of sounds. The cost effectiveness of such a program as this would be highly attractive in the light of the possibility of bootlegging custom-engineered packages on existing systems and/or providing a time-shared environment.

K. R. Ingham

C. SPECTRAL TRANSFORMATION AS A PSYCHOLINGUISTIC TECHNIQUE: A SUMMARY

Speech communication, one of the most complex cognitive processes, has been studied from many different viewpoints, each having its own set of biases and assumptions. At the feature-description end of the scale, there are acoustic phoneticians who try to isolate those aspects of the acoustic speech signal which are responsible for the resulting percept. At the other end of the scale are the linguists and psycholinguists who attempt to consider the more global aspects of the speech process in terms of deep structure, surface structure, and phrase units. There is, however, no convincing experimental evidence to indicate that speech perception is a segmentable process.

In the experiment described here, pairs of subjects learned to converse with each other through an acoustic (electronic) medium in which the spectrum of the natural speech was rotated about a center frequency of 1.6 kHz; that is, the high-frequency energy became low-frequency and vice versa, so that a component at 200 Hz would emerge at 3000 Hz. Spectral transformation is a kind of distortion that is not similar to amplitude distortion. As Licklider, Bindra, and Pollack[1] have shown that hard-limited speech — a most severe amplitude distortion — is intelligible because the basic spectral characteristics of the signal are not changed. Spectral rotation, on the other hand, completely changes the spectral shape but does not remove any of the information.

Six pairs of subjects participated in the experiment which ran three sessions per week for seven weeks. During these half-hour conversation sessions, subjects conversed, or attempted to converse, with each other through the transformation medium (no visual contact). The instructions given to the subjects told them that their only goal was learning to communicate. In a certain sense, the subjects were hearing a speechlike signal, which was analogous to a foreign language in that most of the sounds were alien; but the syntax and semantic structure of this pseudo-language just happen to be identical to those of English. The experimental environment was as unstructured as possible and, for this reason, was probably more similar to the normal speech situation than other speech experiments in which a restricted stimulus set is used in a particular task.

The conversation sessions were tape-recorded and later analyzed for characteristic stages of learning. The four stages generated were acoustic probing, high-redundancy utterances, synthetic conversation, and integrated conversation.

1. Acoustic probing was the initial stage during which subjects would be uttering, for example, a continuous diphthonglike vowel as in "aaaaaeeeeoooo." Alternatively, they would practice simple monosyllabic sounds, varying one phoneme at a time, as in "mat, pat, bat, cat ... ." It appeared that subjects were exploring the nature of the

medium and correlating the spoken sound with the perceived sound in much the same way that an infant[2] will talk to himself simply for the pleasure of hearing the funny noises that he can make.

2. High-redundancy utterances generally occurred in the second stage and are characterized by conversations that are highly predictable, as in the sequence "there are 365 days in the year; there are 52 weeks in the year; there are 4 weeks in the month; ... ." Also, subjects discovered that spelling a word that was not understood only required that they be able to identify the 26 elements in the alphabet set. Many of these conversations seemed to correspond to drill practice used in learning a foreign language.

3. Synthetic conversation differed from the previous stage, in that the conversations contained significant information, and the subjects were paying attention to the ideas, rather than to the sounds. Difficult ideas were communicated by using "associated" words. For example, to transmit the misunderstood word "car" in the sentence "Did you buy a car?," one subject went through the sequence "automobile," "Detroit," "Northern United States," "A city in Northern United States." This is very much the same kind of conversation that takes place when one is speaking a foreign language that he does not know very well. If, for example, the sentence "pass the salt" is not understood or if one does not remember the word for salt, one might say "pass the other spice standing next to the pepper."

4. The last stage, integrated conversation, is simply normal conversation with no concern for the fact that the speech signal is spectrally transformed. Actually, only two pair really attained this level of competence, but one of these subjects commented that "This is just like sitting at home chatting."

The learning situation is actually far more complicated than the summary of stages has indicated. Some subjects spent excessive time in one stage, while others skipped a stage completely. The two pairs of subjects who reached a performance level corresponding to integrated conversation spent very little time in either of the first two stages. Rather, they attempted to converse immediately without any form of drill practice.

In order to gain a more objective measure of the learning dynamics, an extensive series of tests was given to each subject following every conversation practice session. These tests attempted to measure performance of sound-unit discrimination, phoneme identification, word recognition, and sentence comprehension.

The results from the vowel tests showed that two distinct vowels almost always remain distinct after transformation, although a semivowel and its neighboring vowel would merge to form a new diphthong. This is illustrated by "bill" which was heard as "boy." Two vowels differing in the tense-lax feature were almost never confused, even before subjects became familiar with the medium. This could have been predicted,

since two of the three cues for this feature, duration and steady-state formant, are independent of the spectral transformation. The front-back place-of-articulation feature, which manifests itself as either high- or low-frequency energy, was initially reversed so that the phoneme /i/ was perceived as /u/, and vice versa. After only half an hour of exposure to the medium, however, subjects had learned to "re-invert" this feature. Only the weakly stressed lax vowels were difficult to identify.

The mode-of-articulation feature, which divides the consonants into the following classes: unvoiced fricatives, voiced fricative, unvoiced plosive, voiced plosive, and nasal, was readily perceived without any practice. Moreover, the place-of-articulation feature was never learned, either identification or discrimination. Thus, /p/ would be confused with /t/ or /k/, but never with /m/, /b/, /v/, or /f/. This could have been predicted from our present knowledge of phonetic cues, since the mode cues are predominantly determined by acoustic correlates that are unaffected by the spectral rotation, for example, frication, duration, pause, plosion, and so forth. The place-of-articulation feature manifests itself, however, as a more subtle formant transition and, according to our present understanding, is a very stable feature. Very little improvement takes place with the consonants; the average score for identification increased from 23% to 35% between session 1 and session 8.

The correlation studies showed that a subject who was particularly adept at identifying consonants, for example, was no more likely to do well in identifying vowels than the subject who was not adept at identifying consonants. We also found that the results from the vowel discrimination and consonant discrimination tests were correlated with each other but neither were correlated with the identification tests. Thus these kinds of tasks on isolated sound units do not appear to be perceptually related.

The ability to identify words was rather poor throughout the experiment, and the results, as could have been predicted, were dependent on the kinds of words chosen for test stimuli. Thus, "bed" could easily be perceived as "bed," "dead," "beg," "dig," "bug," etc., since the dependable cues are the consonant place-of-articulation and the vowel tense-lax (to some degree, vowel front-back). If the stimulus word was uncommon, compared with the other words within the confusion class, then it would almost never be identified correctly. For isolated one- and two-syllable words the average score never exceeded 10%.

In contrast, the average scores for identifying the names of geographic locations, such as Chicago, New York, Soviet Union, increased from 28% on the first session to 72% on the last session. This set of stimulus words differs from the previous tests, in that the set of possible elements is limited and each element contains a length and a stress cue. "West Virginia" is never confused with "Greece." Also, when subjects were told that each stimulus word belonged to a particular class such as names of vegetables the average score increased from 10% to 50%.

Even though the ability to perceive words never increased significantly and the ability to identify isolated phonemes ceased to improve after several sessions, average comprehension of sentences continued to increase from 8% on the first session to 40% on the last session.   Moreover, the correlation studies demonstrated that sentence performance was correlated with the geographic test results, but neither was correlated with words or phonemes.   This has been interpreted as meaning that the mechanisms for learning to understand transformed speech are not related to the constituent sound elements.   Rather, the subjects must learn to use other aspects of the signal, such as stress, intonation, semantic redundancy, syntactic structure.

We observed that on the initial sentences tests the words which were perceived correctly were always the small function words, for example, "the," "and," "is," "of." The response to the stimulus sentence "The farmer had many chickens and cows" was, in many cases, "The .... and ...." Recognizing the medial "and" means that the subject was able to segment the utterance based on some physical cues, since he did not perceive enough of the words to use semantic or syntactic constraints. A few sessions later, some content words were perceived but they were always perceived within the phrase.   Sometimes a subject's response would correspond syntactically to the stimulus, even though none of the words was correct.   The best and most powerful example of this was the sentence "Hoist the load to your left shoulder," which was perceived by one of the subjects as "Turn the page to the next lesson." In this example there is almost no phonetic or semantic correspondence between the stimulus and response. The most attractive assumption to be made from this phenomenon is that syntax, in fact, is encoded into the prosodic features in such a way that it is used to specify the function of the words and that it helps the listener to segment the utterance into words.   Much more emphasis is now being placed on the importance of prosodic features in speech perception.[3]

From the analysis of the data it appears that the perception of sentences, at least under the condition of transformed speech, is more dependent on the use of content redundancy, syntactic structure, and prosodic features, than on phonetic cues or phonemic sequence.   A subject who is particularly sensitive to isolated sounds does not necessarily do better in conversing or in comprehending sentences.

A more thorough description and discussion of this experiment may be found in the author's doctoral dissertation.[4]

B. A. Blesser

References

1.   J. C. R. Licklider, D. Bindra, and I. Pollack, "The Intelligibility of Rectangular Speech Waves," Am. J. Psychol. 61, 1-20 (1948).

2.   R. H. Weir, Language in the Crib (Mouton and Co., The Hague, 1962).

3. P. Lieberman, Intonation, Perception, and Language, Research Monograph No. 38 (The M.I.T. Press, Cambridge, Mass., 1967).

4. B. A. Blesser, "Perception of Spectrally Rotated Speech," Ph.D. Thesis, Department of Electrical Engineering, M.I.T., June 1969.

D. IMAGE RESTORATION USING THE MELLIN TRANSFORM

Up to the present time, most works on image restoration have been concerned with degrading systems that are linear and shift-invariant. We encounter in practice, however, many degrading systems that are linear but shift-variant. Images degraded by linear shift-variant systems whose impulse responses are of the form

$$h_1(r, \theta; r_o, \theta_o) = \text{Output at polar coordinates } (r, \theta) \text{ attributable to an input at polar coordinates } (r_o, \theta_o)$$

$$= f(r_o) \, h\left(\frac{r}{r_o^n}, \theta - \theta_o\right) \tag{1}$$

may be restored by using both the Fourier and Mellin transforms.

For a degrading system with an impulse response (1), the input image $u(r, \theta)$ and the output (degraded) image $v(r, \theta)$ are related by

$$v(r, \theta) = \int_0^\infty \int_0^{2\pi} f(r_o) \, h\left(\frac{r}{r_o^n}, \theta - \theta_o\right) u(r_o, \theta_o) \, r_o \, dr_o \, d\theta_o. \tag{2}$$

Let the Fourier transform in the θ coordinate of the function $u(r, \theta)$ be denoted by

$$\overline{u}(r, \lambda) = \int_0^{2\pi} u(r, \theta) \, e^{-j\lambda\theta} \, d\theta.$$

Let the Mellin transform[1] in the r coordinate of the function $u(r, \theta)$ be denoted by

$$\widetilde{u}(s, \theta) = \int_0^\infty r^{s-1} u(r, \theta) \, dr.$$

Thus, using (2), we obtain

$$\overline{v}(r, \lambda) = \int_0^\infty f(r_o) \, \overline{h}\left(\frac{r}{r_o^n}, \lambda\right) \overline{u}(r_o, \lambda) \, r_o \, dr_o.$$

Then, taking the Mellin transform in r and changing the variable of integration to $x = r/r_o^n$, we have

$$\widetilde{\widetilde{v}}(s, \lambda) = \int_0^\infty r_o^{ns-1} f(r_o) \, \bar{u}(r_o, \lambda) \, r_o^2 \, dr_o \int_0^\infty x^{s-1} \bar{h}(x, \lambda) \, dx.$$

Let $g(r_o, \theta_o) = f(r_o) \, u(r_o, \theta_o) \, r_o^2$, then

$$\widetilde{\widetilde{v}}(s, \lambda) = \widetilde{\widetilde{g}}(ns, \lambda) \, \widetilde{\widetilde{h}}(s, \lambda). \tag{3}$$

The original image may be restored[1] by formally inverting (3):

$$u(r_o, \theta_o) = \frac{1}{f(r_o) \, r_o^2} \frac{1}{2\pi} \int_{-\infty}^\infty e^{j\lambda\theta} \frac{1}{2\pi j} \int_{c-j\infty}^{c+j\infty} r_o^{-s} \frac{\widetilde{\widetilde{v}}\left(\frac{s}{n}, \lambda\right)}{\widetilde{\widetilde{h}}\left(\frac{s}{n}, \lambda\right)} \, ds \, d\lambda. \tag{4}$$

In the case of the coma aberration

$$h_1(r, \theta; r_o, \theta_o) = \frac{1}{r_o^2} \, h\left(\frac{r}{r_o}, \theta - \theta_o\right). \tag{5}$$

That is, $f(r_o) = \dfrac{1}{r_o^2}$ and $n = 1$.

Because the ray aberration for coma depends linearly on the distance of the object point from the optical axis, by assuming uniform and equal illumination over the exit pupil from any point source in the object plane, it may be seen that the impulse response for coma must have the argument dependence indicated in (5). For example, if the distance from the axis of an object point is doubled, then the impulse response is spread out by a factor of two in each linear direction. The amplitude at the corresponding point would be reduced by a factor of four (by the $1/r_o^2$ factor) in order to preserve the integral of the impulse response. When the impulse response does have the form (5), the inversion formula simplifies to

$$u(r_o, \theta_o) = \frac{1}{2\pi} \int_{-\infty}^\infty e^{j\lambda\theta} \frac{1}{2\pi j} \int_{c-j\infty}^{c+j\infty} r_o^{-s} \frac{\widetilde{\widetilde{v}}(s, \lambda)}{\widetilde{\widetilde{h}}(s, \lambda)} \, ds \, d\lambda.$$

A second example arises when an image plane is tilted out of the focal plane for a cylindrical lens system. Such a system is shown in Fig. XIV-12 where the axis of symmetry is out of the paper. Assuming that the light distribution from r_1 to r_2 is uniform because of an object line source focused at t, we find that

$$r_1 = t \, \frac{\sec \theta}{1 + \dfrac{R}{c} \tan \theta + \dfrac{t}{c} \tan \theta}$$

$$r_2 = t \, \frac{\sec \theta}{1 - \dfrac{R}{c} \tan \theta + \dfrac{t}{c} \tan \theta}.$$
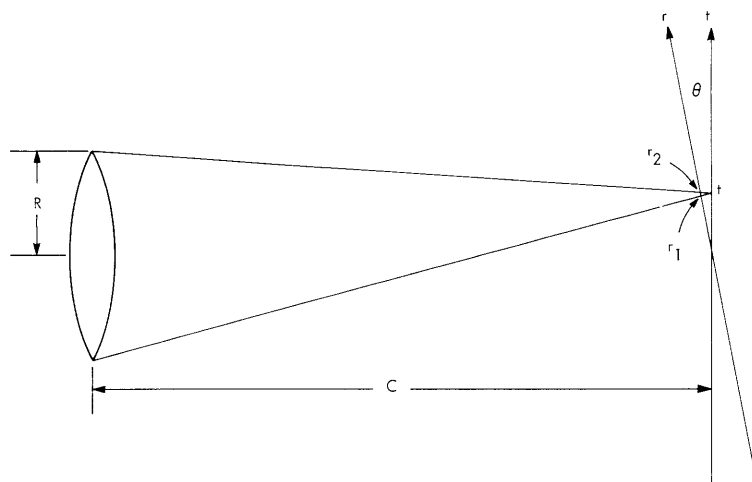
Fig. XIV-12. System geometry.

If $\frac{t}{c} \ll 1$ and $t \ll R$, then

$$r_1 \approx t \; \frac{\sec \theta}{1 + \frac{R}{c} \tan \theta} \quad \text{and} \quad r_2 = t \; \frac{\sec \theta}{1 - \frac{R}{c} \tan \theta} .$$

For simplicity, let

$$\tau = \frac{\sec \theta}{1 + \frac{R}{c} \tan \theta} \; t$$

and

$$k = \frac{1 + \frac{R}{c} \tan \theta}{1 - \frac{R}{c} \tan \theta} .$$

Then the impulse response is

$$h(r) = \begin{cases} \dfrac{1}{k-1} & 1 \leqslant r \leqslant k \\ \\ 0 & \text{otherwise.} \end{cases}$$

If the input is $U(t) = u(\tau)$, then the output is

$$v(r) = \int_{-\infty}^{\infty} \frac{1}{\tau} h\left(\frac{r}{\tau}\right) u(\tau) \, d\tau .$$

Taking the Mellin transform of both sides, we obtain

$$\tilde{v}(s) = \left(\frac{1}{s} \frac{k^s - 1}{k-1} \right) \tilde{u}(s)$$

or

$$s\tilde{v}(s) = \frac{1}{k-1}\left[k^s\tilde{u}(s)-\tilde{u}(s)\right].$$

By inverting this, we obtain

$$u(r) = (k-1)\,rv'(r) + u\left(\frac{r}{k}\right).$$

Just as a time-invariant or spatially invariant system is simply characterized by its action on an exponential function of any frequency, a system with impulse response

$$h_1(r,r_o) = \frac{1}{r_o}\,h(r/r_o)$$

is simply characterized by its action on r_o^{-s} for any complex s. That is, the response to r_o^{-s} is $\tilde{h}(s)\,r^{-s}$; if the input is expressed as a Taylor series, then the output is another Taylor series whose n^{th} term is the product of the nth term of the input and $\tilde{h}(-n)$. Also similar to the exponentials, the functions r^{-s} will be increasing, decreasing, or oscillatory for negative, positive, or imaginary choices of s, respectively.

At present, some computer programs are being written to demonstrate this use of the Mellin transform. One way of finding the Mellin transform of a sampled signal is to make a change of the independent variable, $r = e^{-t}$, then to take the Fourier transform. These computer results will be presented in a future report.

G. M. Robbins

References

1. G. F. Carrier, M. Krook, and C. E. Pearson, Functions of a Complex Variable: Theory and Technique (McGraw-Hill Book Co., New York, 1966).

E. INTERACTIVE ROENTGENOGRAM ANALYSIS

In the detection of pulmonary and cardiovascular diseases and diseases of the mediastinum and bony thorax, the chest X-ray (or chest roentgenogram) serves as an important diagnostic tool. The hard-copy picture also serves as a historical marker noting the absence, progress, or development of disease. As a result, over seventy million chest roentgenograms are prepared in the United States every year. To facilitate the diagnostic process, we have begun a study of techniques for the enhancement, interpretation, and analysis of these images.

Basically we are interested in interactive mechanisms whereby the radiologist can "converse" with the image. As an example of this conversational process the

radiologist, noticing an unusual region in the lung field, may define the region of interest with a light pen and receive in return quantitative measurements concerning mass density or left side-right side image comparison.
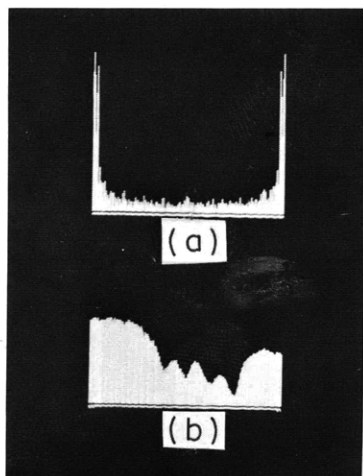

Fig. XIV-13. (a) Magnitude of Fourier spectrum vs frequency for vertical-scan line through chest roentgenogram.
(b) Transmittance amplitude vs vertical position for vertical-scan line of chest roentgenogram.

In our preliminary studies we have considered techniques for removing the perodic rib structures from the chest image. By examining the magnitude of the Fourier spectrum of scan lines taken vertically through a chest roentgenograph (Fig. XIV-13a), we have tried to filter out the periodic component corresponding to the rib cage. This component is quite evident in the transmittance amplitude vs vertical position plot of Fig. XIV-13b.

At present, we are exploring more analytic procedures for the interactive analysis of chest roentgenographs.

D. M. Ozonoff, I. T. Young

F. SPATIAL FILTERING WITH THE NEW SCANNER

The object of the work reported here is to develop an economical and convenient procedure for performing a wide variety of spatial filtering operations.

The new scanner is a digitally controlled CRT flying-spot scanner that uses light feedback to achieve a high signal-to-noise ratio and flat field. IBM 360 compatible tapes are produced from pictures and vice versa. Furthermore, a certain amount of processing can be done directly.

There are several ways in which the New Scanner can be used to achieve spatial filtering, which involves an effective impulse response having negative lobes.

In the first method the scanner is defocussed and a signal proportional to the

logarithm of the defocussed transparency is recorded on tape. This signal is then played back and displayed as a low-contrast negative image on the face of the scanner, which is now in focus. A new output signal is generated with this defocussed image serving as a mask as the transparency is rescanned. The result is that the output
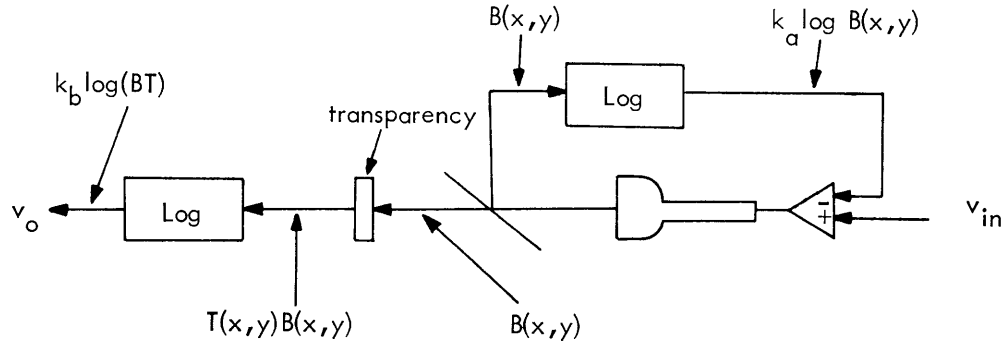

Fig. XIV-14. Scanner arrangement.

is proportional to the logarithm of the transmittance minus a constant times the logarithm of the blurred transmittance. When this signal is displayed in the logarithmic mode the result is an image whose gamma, or contrast, can be controlled by adjusting the contrast of the negative mask.

The scanner arrangement is shown in Fig. XIV-14. By use of feedback around CRT, its brightness is controlled so that

$$\log B(x, y) = \frac{v_{in}}{k_a},$$

where v_{in} is the voltage that was applied at the time the point (x, y) was being scanned. Since the light passing through the transparency is equal to the incident light multiplied by the transmittance T, the output voltage

$$v_o = k_b[\log T + \log B]$$

$$v_o = k_b\left[\log T + \frac{v_{in}}{k_a}\right].$$

In use as an ordinary scanner, v_{in} is a constant, and the gains are adjusted so that $v_o = 0$ for the minimum transmittance in the picture, and $v_o = 1$ (actually -8 V in our system) for the maximum transmittance.

If the scanner is defocussed optically or electrically, we get the same result,

except that T is now replaced by T', a local space-averaged value.

The signal may now be recorded and played back, being displayed on the scanner itself as a negative. For clarity, we use the subscripts 1 for the initial operation and 2 for the second operation. Thus

$$v_{2in} = 1 - v_{10}$$

$$\log B_2(x, y) = \frac{1 - v_{10}}{k_{a2}} - \frac{1 - l_{b1} [\log T'+K]}{k_{a2}}$$

$$v_{20} = k_{b2} \left[\log T + \frac{1}{k_{a2}} (1 - k_{b1} (\log T'+K)) \right]$$

$$v_{20} = k_{b2} \left[\log T - \frac{k_{b1}}{k_{a2}} \log T' + \frac{1 - k_{b1} K}{k_{a2}} \right],$$

which is the desired result. The output is $k_{b2} \log T$ (which is what the output would have been with constant light) minus a variable lowpass-filtered version of the same image. The constant term is of no importance and, in any event, there are enough parameters to set it at an arbitrary value.

For recording, v_{20} may be displayed and photographed on the photodisplay in the usual manner. Note that in order for the image on the photodisplay CRT to have the same characteristics (that is, tone scale) as the original transmittance, the CRT brightness must be directly proportional to T. Thus the logarithmic display mode must be used.

An alternative method of achieving what appears to be an identical result is to combine the logarithms of the focussed and defocussed images electrically rather than optically. The addition of logarithms produces the same result as the multiplication of the original signals, and both the tone scale and filtering operations seem to be performed in the same manner. An additional advantage of the electrical combination is that the coefficients may more easily be adjusted through a range of positive and negative values.

The electrical combination of focussed and defocussed images may also be accomplished if the signals are linear rather than logarithmic. A consideration of the resultant tone scale indicates that the logarithmic and linear combinations, at least between a gamma of zero and +1, differ only slightly. The main difference seems to be that the linear manipulations cause the resultant signal to "pivot" about 50% brightness, while the logarithmic signals pivot around 50% of density, which is typically 10% brightness.

The first application of these techniques has been an attempt to improve the visibility

of fine detail in a 70-mm photo-fluorographic chest film (Fig. XV-15). The film was scanned in the linear mode fashion with the scanner defocussed by moving the film
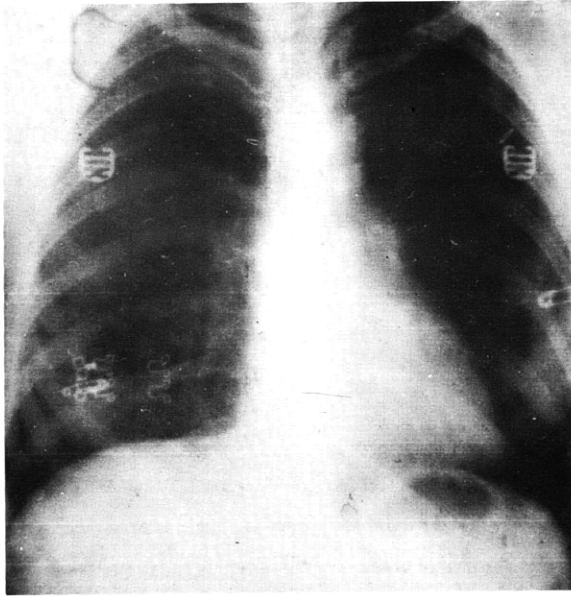

Fig. XIV-15.

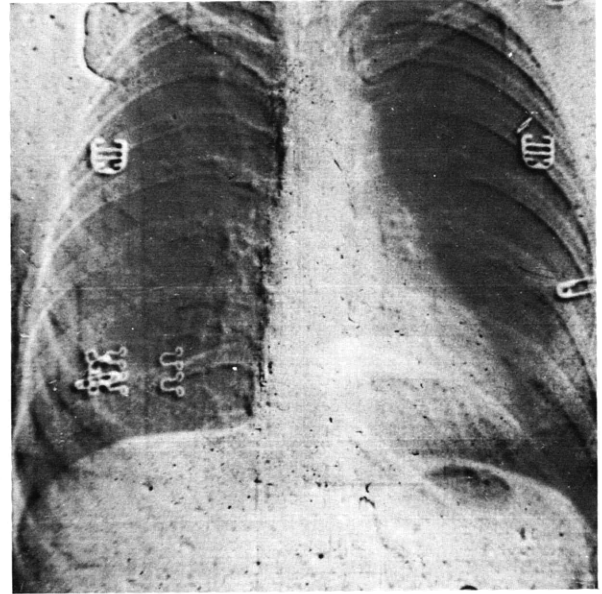Original x-ray. Sampled 1024 × 1024 elements, digitized at 8 bits/element, displayed and photographed.

Fig. XIV-16.

Processed x-ray. High-frequency components × 2.0; low-frequency components × 0.5.

holder 0.5 cm to the left and the CRT holder 0.85 mm to the right. The defocussed image was recorded on tape and then placed back in the register with a sharp image as the latter was being scanned. The photographs (Fig. XV-16) were produced by subtracting from the focussed image a certain proportion of the defocussed image, the relative amplitude of the two-signal components which is indicated below the pictures. The best results were achieved with amplitudes 2.0 and 1.5, respectively, where 1.0 is the normal amplitude as shown in the original picture.

These preliminary studies have indicated that some improvements are necessary in the scanner in order to prevent a loss of some detail which may be visible in the original x ray; in particular, in order to record detail rendered at densities above 2.0, it is necessary to improve the signal-to-noise ratio of the scanner by eliminating some low-level pickup, as well as certain errors in the operation of the circuit that integrates the photomultiplier tube current.

W. F. Schreiber

G. MODULAR PICTURE PROCESSING PACKAGE (MP^3)

This report describes the modular picture processing package (MP^3) which has been developed in the Cognitive Information Processing group. MP^3 was initiated to provide the ability to computer process microscopic images of biological/medical significance such as chromosome spreads, peripheral blood smears, Papanicolaou smears, and others. These images are characterized by having a relatively uniform background upon which several objects are located. In some cases, the objects themselves contain one or more discrete subobjects. The objects are separated from their surrounding background by a detectable change in optical density.

Although a sampled and digitized image is usually thought of as a two-dimensional array of brightness values, MP^3 treats the image as a one-dimensional array, A

$$A_0, A_1, A_2, \ldots, A_{(M \times N)}$$
$$[N], [\text{Picture} \ldots \qquad] \qquad (1)$$

where each row of samples has been juxtaposed edge to edge between its two nearest neighbors.

$$A_0, A_1, \ldots, A_N, A_{(N+1)}, \ldots, A_{2N}, \ldots, A_{((M-1 \times N)+1)}, \ldots, A_{(M+N)}$$
$$[N], [\text{ Row } 1], [\text{ Row } 2], \ldots, [\text{ Row } M] \qquad (2)$$

Here N = the number of columns in the original image, and M = the number of rows. This scheme is a convenient way of treating a picture both conceptually and computationally. For example, consider that the location of a point in a two-dimensional array requires the storage of two coordinates, while a point in a one-dimensional array can be located with only one index, a saving of 50% of storage space.

With the linear, line-by-line scheme shown in (2), an object in the picture which is more than one row high will be represented in the array by isolated segments.

$$A_0, A_1, \ldots, A_\ell \ldots \qquad A_m, \ldots, A_n, \ldots,$$
$$[N], [\ldots \quad (\text{part of object in row i}), \ldots, (\text{part of object in row j}),$$
$$\ldots, A_p, \ldots \qquad A_q, \ldots, A_{(M \times N)}$$
$$\ldots, (\text{part of object in row k}), \ldots, \text{etc.}, \ldots] \qquad (3)$$

Such an object could be defined by simply listing the index of each point within the object. For large objects, however, this would be somewhat wasteful of space. A more compact scheme would be to list pairs of beginning and ending points for connected sequences of object points in the array.

..., (beginning point of sequence, ending point),

(beginning point, ending point), ... etc. (4)

For the object in (3), the list would contain

... (ℓ, m), (n, o), (p, q), ... (5)

As the picture is represented as a one-dimensional array, and the last point of one row (in the original picture) is next to the first point of the next row, sequences that begin on one row and end on another row are possible.

A slight variation on (4) is a list of the beginning point of a series of connected points, and the number of points in the sequence.

..., (beginning point, # of points in sequence), (beginning, #), ... etc. (6)

Both (4) and (6) contain the same information in only slightly different forms. In different computational situations, one or the other form will be the more convenient form to work with, and simple algorithms are available to convert from one form to another. Form (6) has been chosen as the standard form, although (4) could just as easily have been.

The table (6) has been named an XOFY table, and is represented in our MP^3 as follows:

Tab_0, Tab_1, Tab_2, Tab_3, Tab_4,

[# entries in, [beginning, # of points], [beginning, # of points],
 table   ]

..., $Tab_{(W-1)}$ $Tab_{(W)}$

..., [beginning, # of points] (7)

where W is the number of entries in the table Tab.

Most of the programs in MP^3 are organized around the XOFY table as their frame of reference in the picture within which they work. This feature can be most easily understood by considering the hypothetical analysis of a simple picture consisting of several dark blobs on a light background. The system user wishes to identify the blobs, and subject their areas to further analysis. The first step is to construct an XOFY table for the picture, which is simply

Tab_0, Tab_1, Tab_2

[2],   [1, $M \times N$] (8)

Using this table as an argument, HISANA (see subroutine descriptions) can now calculate the brightness histogram for the picture, determine the average brightnesses of

the background and the blobs, and choose an appropriate brightness threshold for sepa-
rating the two. The picture is then searched point by point by SEARCH for a point darker
than the threshold. The search is made within the XOFY table which is also supplied
as an argument. When such a point is found, its location is passed to CONTUR which,
by using the same brightness threshold to define the edges of the object, traces the con-
tour of the object within which the point was contained. CONTUR returns a list of the
locations of the contour points. This list is converted by XOFY into an XOFY table
describing the object.

This XOFY table can now be used in a number of ways. It may be used with DP, a
display subroutine, to display the blob, or with MOMNT to extract features from the
blob, or the blob itself may be reanalyzed by HISANA and searched for dark or light
subobjects. In the last case, the blob XOFY table serves the same purpose as (8) did
previously.

Finally, when analysis of the first blob has been exhausted, COMBIN can be used to
remove the blob's XOFY table from (8), which yields a new XOFY table that defines
the original picture minus the first blob.

Thus with MP^3, one can perform a very involved analysis of a picture quite easily
by calling the same small group of subroutines repeatedly and only varying the exact
sequence of the calls and the arguments in each call. This is most easily illustrated
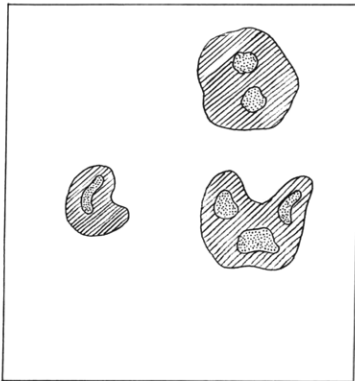by the following example.

Fig. XIV-17. Illustrating a sample picture that can
be analyzed by the sample program in
Table XIV-1. It shows several dark
blobs on a light background. Each blob
has one or more inclusions.

Table XIV-1 illustrates a sample program using MP^3 which analyzes the picture dia-
grammed in Fig. XIV-17. The picture consists of a light background on which are sev-
eral dark objects. Each object contains one or more light inclusions. Let us suppose
that we need to measure the areas of these inclusions. The problem is to isolate each
object in turn, and then locate each inclusion and measure its area.

The program first dimensions the necessary arrays and sets up an XOFY table
(PICTAB) defining the whole picture. (It is assumed that some provision has already

Table XIV-1. Sample program.

```
C   SAMPLE PROGRAM USING MP3
        INTEGER A, PICTAB, BLOB, THRSH1, PGD1, POINT, LIMIT, OBJTAB,
        CTAB, THRSH2, HOLETB
C   DIMENSION STATMENT ASSUMES THAT MEASURES HAVE BEEN TAKEN TO ALLOW
C   USE OF ZERO SUBSCRIPTS
        DIMENSION A(5000), PICTAB(200), CTAB(200), OBJTAB(100), HOLETB(50)
        M = (# OF ROWS IN PICTURE)
        N = (# OF COLUMNS IN PICTURE)
C   SET UP X OF Y TABLE DESCRIBING PICTURE
        PICTAB (0) = 2
        PICTAB (1) = 1
        PICTAB (2) = M*N
C   HISTOGRAM PICTURE AND SELECT THRESHOLD LEVEL
        CALL HISANA (A, PICTAB, 5,5, BLOB, THRSH1, BGD1)
10      CALL SEARCH (A, PICTAB, 0, THRSH1, POINT, LIMIT)
        IF (POINT.EQ.0) GOTO40
        CALL CONTUR (A, POINT, CTAB, 0, THRSH1)
        CALL XOFY (CTAB, OBJTAB)
C   HISTOGRAM OBJECT JUST FOUND
        CALL HISANA (A, OBJTAB, 5,3, BLOB, THRESH2, HOLE)
20      CALL SEARCH (A, OBJTAB,-1, THRSH2, POINT, LIMIT)
        IF (POINT, EQ. 0) GOTO 30
        CALL CONTUR (A, POINT, CTAB, -1, THRSH2)
        CALL XOFY (CTAB, HOLETB)
C   MOMNT IS SET UP TO CALCULATE AREA
        CALL MOMNT (A, HOLETB, BDG)
C   AFTER CALCULATING AREA OF HOLE, REMOVE IT FROM OBJECT
        CALL COMBIN (OBJTAB, HOLETB, CTAB)
        CALL MOVE (CTAB, OBJTAB)
        GOTO 20
C   WHEN ALL HOLES HAVE BEEN FOUND, REMOVE OBJECT FROM PICTURE
30      CALL COMBIN (PICTAB, OBJTAB, CTAB)
        CALL MOVE (CTAB, PICTAB)
        GOTO 10
C   WHEN ALL OBJECTS HAVE BEEN FOUND, JOB IS OVER
40      STOP
        END
```

been made to allow use of zero subscripts.) A call to HISANA produces a brightness-level histogram of the whole picture and determines an appropriate clipping threshold for the boundary of the objects. The picture is then searched until a point darker than the threshold is found. If POINT is returned zero, no objects remain in the field, and the program branches to the end and stops. If a point is found, the contour of the object is traced (TAB contains the contour points), and an XOFY table is constructed for the object (OBJTAB).

With OBJTAB as a frame of reference, the object is histogrammed and a new threshold, THRSH2, is selected. Changing the value of the parameter SWITCH from 0 to -1, the object is searched for a bright inclusion. If POINT is returned zero, no more inclusions are present in the object, and the program branches to statement 30 where the object XOFY table OBJTAB is removed from the picture XOFY table PICTAB. The program then searches PICTAB for another object. If an inclusion was found, its boundary is contoured and an XOFY table, HOLETB, formed for it. MOMENT, which has been set up to calculate the inclusion's area, uses HOLETB to do so. HOLETB is then removed from OBJTB, and the program branches to 20 to search for another inclusion in the object.

In this way, an object is found, all inclusions in the object are located and analyzed, the object is removed from the picture, another object is found, and so forth through the picture.

The subroutines in MP^3 may be divided into three general categories: (i) main subroutines, (ii) slave subroutines, and (iii) peripheral subroutines. Table XIV-2 lists the programs comprising the three categories. The main subroutines comprise the working core of MP^3. Slave subroutines are programs that are called by the main subroutines; normally, they would not be called by the user. Peripheral subroutines are user-called programs which, although not necessary for operation of MP^3, make its use more convenient. This category includes the display routines and data-packing routines. As the peripheral subroutines would be of interest only to workers who have a PDP-9 with 34-H display, they will not be included in the following descriptions.

MP^3 is, at present, implemented in PDP-9 Fortran IV. Some of the simpler routines, such as UN, STAND, SORT, MOVE, etc. also exist in Macro-Language versions to speed execution. A few routines, such as the display package, are hardware-dependent, and are implemented in assembly language only.

There are several departures from normal Fortran practice in our package. A saving in storage space has been effected by packing the 6-bit picture points three per word into our PDP-9's 18-bit word. In order to easily reference the data in packed form, special fetch and deposit routines are used in place of the normal Fortran IV subscripting feature. Thus, to reference Array (i), instead of using the normal Fortran call X = ARRAY(I), we call our own external function X = GET(ARRAY, I).

Table XIV-2. Subroutines in the modular picture processing package.

Main Subroutines

1. HISANA

2. SEARCH

3. CONTUR

4. XOFY

5. COMBIN

6. MOVE

7. MOMNT

Slave Subroutines

1. UN

2. STAND

3. CLEAN

4. SORT

5. XTREMA

6. LOGIPK

Peripheral Subroutines

1. Display subroutines

2. Data packing subroutines

There is one additional departure from normal FORTRAN IV practice in our system; the use of the Zero subscript. Normally Fortran does not reserve a location for Array (0). We overcome this difficulty by defining ARRAY with only one location and a dummy array B.

DIMENSION ARRAY(1), B(N+1)

We then equivalence ARRAY(1) to B(2), thereby leaving us with

ARRAY(0) equivalent to B(1) and ARRAY(N) equivalent to B(N+1).

The following descriptions of the subroutines in the package list the purpose of the subroutine, a description of the arguments, a note on any special requirements or limitations, the language of implementation, and approximate running times on the PDP-9 computer.

SUBROUTINE HISANA (A, TAB, UPPER, LOWER, CELLPK, CLPLVL, BGDLVL)

Purpose of Subroutine

HISANA forms and analyzes the brightness histogram for an area of a picture. It is usually used to determine the appropriate brightness threshold for a subsequent search.

Description of Argument

A brightness level histogram of a portion of A defined by XOFY table TAB is produced. The histogram is then smoothed with an exponential smoothing function and the smoothed histogram is searched for the most significant maxima and minima. The number of maxima and minima found is between UPPER and LOWER. BGDLVL, CELLPK, and CLPLVL are set to the brightness value for the maximum nearest 63, the next darker maximum, and the trough between the two maxima, respectively. BGDLVL usually correspond to the mean brightnesses for the picture background, CELLPK to the mean brightness for the cells, and CLPLVL to the appropriate clip threshold to separate cells from background.

Special Requirements or Limitations

If display of raw and smoothed histogram is desired, subroutine HISDPY must be provided.

Implementation

PDP-9 FORTRAN IV.

Running Time on PDP-9

$\cong$ 8 sec for 20,000 point picture, proportionally less for smaller fields.

SUBROUTINE SEARCH (ARRAY, TAB, SWITCH, THRESH, POINT, LIMIT)

Purpose of Subroutine

SEARCH searches a specific portion of a picture for the first point which is less than (or greater than) a given threshold.

Description of Argument

SEARCH searches the interior of the area defined by XOFY table TAB for a point less than THRESH (or greater than or equal to THRESH if switch = -1). If such a point is found, its location is returned as POINT, otherwise POINT = 0. The location of the first point in the X of Y segment where POINT was found is returned in LIMIT.

Special Requirements or Limitations

Picture size limited by available core, SWITCH is a logical variable with values -1 (true) or 0 (false).

Implementation

PDP-9, FORTRAN IV.

Running Time

Varies with # of points searched. <1 sec for 15K points.

SUBROUTINE CONTUR (A, LOC, TAB, SWITCH, THRESH)

Purpose of Subroutine

CONTUR is used to find the outline of an object in a field. It may be used either on a dark object in a light field, or a light object in a dark field. Also, for either case it can find either the interior contour (the points just inside the object), or the exterior contour (the points just outside the object).

Description of Arguments

In the picture array A, CONTUR traces the outline of an object beginning with the point LOC. For inside contours, LOC should be on the object's edge; for outside contours, LOC should be a point adjacent to the object's edge. Switch is True (0) for dark objects on light fields, and False (-1) for light objects on dark fields. Thresh is the boundary threshold for tracing the contour. The contour points are returned in table TAB, beginning with TAB (1) = LOC. TAB (0) contains the number of points in the contour.

Special Requirements or Limitations

LOC is usually provided by a previous call of SEARCH. LOC must actually be a point on the object contour (as defined by the threshold THRESH, or the contour returned will be a 4-point diamond. SWITCH is a logical variable true (0) or false (-1).

Implementation

PDP-9 FORTRAN IV.

Running time

«1 sec for objects with less than 400 point contours.

SUBROUTINE XOFY (LIST, TAB)

Purpose of Subroutine

XOFY is used to convert the list of contour points of an object supplied by CONTUR into an XOFY table defining the object.

Description of Arguments

The list of points in table LIST is used to form the XOFY table TAB. LIST (0) contains the number of contour points in the table.

Special Requirements or Limitations

LIST must actually be a series of adjacent points or the program will fail.

Implementation

PDP-9 FORTRAN IV.

Running Time

≪1 sec for objects with less than 400 points in the contours.

SUBROUTINE COMBIN (TABO, TABN, TABX)

Purpose of Subroutine

COMBIN produces a new XOFY table TABX, which defines an area defined by TABO minus the area defined by $\overline{\text{TABN}}$.

Description of Arguments

COMBIN produces TABO · TABN = TABX. It is usually used to remove an object from a picture after it has been analyzed.

Special Requirements or Limitations

Tables must be standard form XOFY tables.

Implementation

PDP-9 FORTRAN IV.

Running Time

≪1 sec for objects with less than 400 points in the contour.

SUBROUTINE MOVE (IN, OUT)

Purpose of Subroutine

MOVE copies table IN into table OUT.

Description of Arguments

OUT (0, ..., IN(0)) — IN(0, ..., IN(0)), IN remains unchanged. This subroutine is convenient when a limited number of arrays is used for all tables and data at different times in a main program, and it is necessary to transfer data from one table to another.

Special Requirements or Limitations

IN(0) must contain the number of entries in the table to be moved.

Implementation

PDP-9 FORTRAN IV, PDP-9 MACRO.

Running Time

$\ll 1$ sec.

SUBROUTINE MOMNT (A, TAB, BGDLVL)

Purpose of Subroutine

MOMNT calculates moments of inertia and other features for an object. It is used for feature extraction. The exact set of features must be tailored to the user's particular needs.

Description of Arguments

The area of picture A defined by XOFY table TAB which has a mean surrounding brightness BGDLVL is analyzed. At present, features calculated are useful in classifying human erythrocytes. The values are printed on an appropriate I-O device.

Special Requirements or Limitations

At present, set up for erythrocyte features. Care should be taken that the machine's wordlength is sufficient to prevent overflow when calculating moments, and appropriate double precision packages provided. On our PDP-9 (18-bit word length) special rapid double precision add, multiply and conversion routines were written to provide fast, accurate calculations.

Implementation

PDP-9 FORTRAN IV.

Running Time

<1 sec.

SUBROUTINE UN(TAB)

Purpose of Subroutine

UN converts XOFY table TAB from standard (Eq. #7) to unstandard (Eq. #4) form.

Description of Arguments

TAB(0) $\leftarrow$ TAB(0)

TAB (2I-1) $\leftarrow$ TAB (2I-1) $I = 1, \ldots, \dfrac{TAB(0)}{2}$

TAB (2I) $\leftarrow$ TAB (2I-1) + TAB (2I) - 1

Special Requirements or Limitations

None.

Implementation

PDP-9 FORTRAN IV, PDP-9 MACRO.

Running Time

$\ll 1$ sec.

SUBROUTINE STAND (TAB)

Purpose of Subroutine

STAND converts XOFY table from unstandard form (Eq. #4) to standard form (Eq. #7).

Description at Arguments

TAB(0) ← TAB(0)

$$\text{TAB (2I-1)} \leftarrow \text{TAB (2I-1)} \qquad I = 1, \ldots, \frac{\text{TAB(0)}}{2}$$

TAB (2I) ← TAB (2I) – TAB (2I-1) + 1

Special Requirements or Limitations

None.

Implementation

PDP-9 FORTRAN IV and MACRO.

Running Time

«1 sec.

SUBROUTINE CLEAN (TAB)

Purpose of Subroutine

Cleans out singular points in an XOFY table.

Description of Arguments

CLEAN combines any entry in XOFY table TAB any that is adjacent to another (i.e., if one segment ends at point j, and the next segment begins at point j + 1, the segments are combined into one).

Special Requirements or Limitations

None.

Implementation

PDP-9 FORTRAN IV and MACRO.

Running Time

«1 sec.

SUBROUTINE SORT (TAB)

Purpose of Subroutine

SORT sorts the entries in TAB from smallest to largest.

Description of Arguments

TAB(0) contains the number of entries in the table. SORT is a typical radix exchange sort procedure.

Special Requirements or Limitations

None.

Implementation

PDP-9 MACRO.

Running Time

Varies with length of table.

SUBROUTINE XTREMA (HIST, IH, HSTR, NA, NH)

Purpose of Subroutine

XTREMA finds the extremes (peaks & troughs) in the histogram array HIST.

Description of Arguments

The peaks and troughs are found which are more than HSTR greater or less than the previous trough or peak, respectively, and their locations are returned in the array IH. IH(1) is set to zero if the first extremum is a trough, and to one if it is a peak. IH(2) contains the first extremum, etc. The number of items in the HIST array is NA.

Special Requirements or Limitations

None.

Implementation

PDP-9 FORTRAN IV.

Running Time

«.1 msec for 64-level histogram.

FUNCTION LOGIPK

Purpose of Subroutine

Provides logical operators.

Description of Arguments

LOGIPK is a multiple entry Macro Language subroutine which provides certain logical operators not available in PDP-9 FORTRAN IV. Among these are exclusive or equivalence, and bit by bit and routines.

Special Requirements or Limitations

None.

Implementation

PDP-9 Macro.

Running Time

<50 μsec at each entry.

J. E. Green, O. J. Tretiak