one second. Even though classification time is not an important factor, a faster rate does help to make the system usable. Recognition of directed paths does seem to be a problem at times since the user cannot have complete control over the input data when sketching on the tablet. However, with a little care while drawing, such a problem can be overcome.

Since the machine used has a comparatively small memory and the drawing is displayed on a storage tube, updating is not as simple a process as desired. Likewise, the system is a little complex to use and guidance is required. All the drawbacks may certainly be improved with equipment more suited to such a system.

The immediate application is limited to drawings with lists of predefined symbols. The recognition technique employed can eventually be expanded to learn new symbols that are not included in the lists. On a larger scale, the system can be applied to architectural and artistic designs as well as shop wiring. To go one step further, edge-detection algorithms can also be added. Thus the system has the following future potential application: to provide robots and such with sensory abilities and to advance man–machine to a more flexible stage.

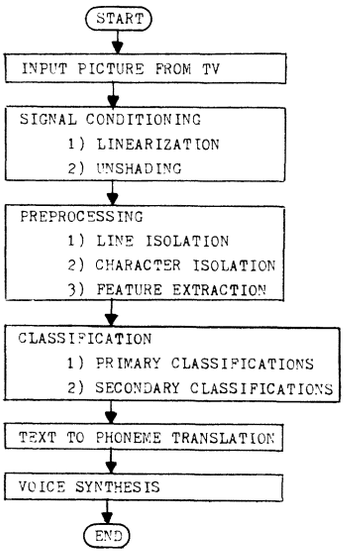

Fig. 1. Speaking system in the testing or demonstration configuration.

References

[1] P. O. Jacobs and J. L. Mueller, "Computerized information processing translates customer requirements into drawings and control systems," *Westinghouse Eng.*, May 1973.
[2] D. P. Greenberg, "Computer graphics in architecture," *Sci. Amer.*, May 1974.
[3] S. K. Chang, "An interactive system for Chinese character generation and retrieval," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, May 1973.
[4] W. C. Lin and T. L. Scully, "Computer identification of constrained handprinted characters with a high recognition rate," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-4, Nov. 1974.

An On-Line Minicomputer-Based System for Reading Printed Text Aloud

ROBERT KOOI AND WEN C. LIN, SENIOR MEMBER, IEEE

Abstract—**An on-line system for reading text aloud from *Newsweek* magazine was developed using a low-cost Vidicon camera and a Votrax voice synthesizer. Of major importance was the development of character separation and recognition algorithms that could handle variable character spacings and large class sizes (upper and lower case characters, numerals, punctuation, etc.) such as those found in books and magazines. In addition the separation algorithm must handle merged characters and the classification algorithm must handle broken or split up characters. A "Synthetic English Speech by Rule" program translates the recognized characters to phonemes which are sent to the voice synthesizer.**

Introduction

Computer imitation of human sensory abilities has been the goal of a vast amount of effort since the beginning of computer development. Great advances have been made in character recognition and speech synthesis and it is only natural that these two areas be combined into a "talking book" machine. Recently available "voice synthesizers" [1] and a program for translating English words into phonemes [2] have bridged the two gaps remaining in such a system. These developments prompted the attempt to combine picture processing, character recognition, text to phoneme conversion and voice synthesis into a system which is capable of reading printed text aloud.

Since the system converts from the printed form of language to the audio form, it is similar to a person reading a book out loud. The system could be used for reading books, newspapers, literature, and letters to blind people or to senior citizens with sight impairments, teaching people to read, entertainment, and transcribing various printed material into machine readable forms with audio feedback for verification.

In this project, *Newsweek* magazine was chosen as the input material because it has a reasonable but not excellent print quality and is therefore probably representative of the quality of most printed text.

The system is split into six major areas (Fig. 1). All areas except the TV input and voice synthesis are implemented in software. After the digitized TV signal is conditioned, the lines and then the characters in the picture are isolated, classified, converted to the proper phonemes, and then spoken, on-line, through the voice synthesizer.

TV Input

The task of digitizing the sample of text to be read is performed by an inexpensive closed-circuit Vidicon-tube television camera [3] interfaced to a PDP-11/45 minicomputer. A page of text (*Newsweek*) is placed on the close-up stand upon which the camera is mounted and four cursors, one each for the top, bottom, left, and right, are used to specify the portion of the page to be processed (Fig. 2). The cursors are adjusted using pushbuttons on a control panel adjacent to a TV monitor. No attempt is made to distinguish between figures and text, therefore only text may be in the area defined by the cursors.

The total area covered by the camera is about 500 vertical by 700 horizontal points of which a 200 by 500 subsection is typically specified as the input area. Fifteen seconds are required to digitize a picture of this size. The resolution of the camera constrains the number of lines that can be input at one time. If seven lines of one column of *Newsweek* magazine are input, the resulting characters (about 250 of them) average fifteen by fifteen points each.

Manuscript received November 17, 1976; revised July 27, 1977.
The authors are with the Department of Electrical Engineering and Applied Physics. Case Western Reserve University, Cleveland, OH 44105.

0018-9472/78/0100-0057$00.75 ©1978 IEEE

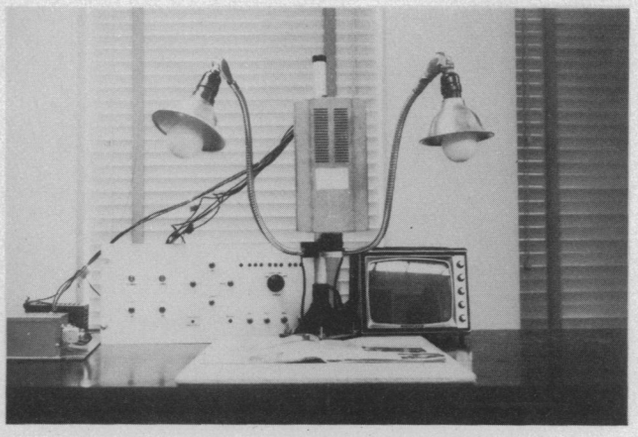

Fig. 2. Closed-circuit TV, monitor, and interface.

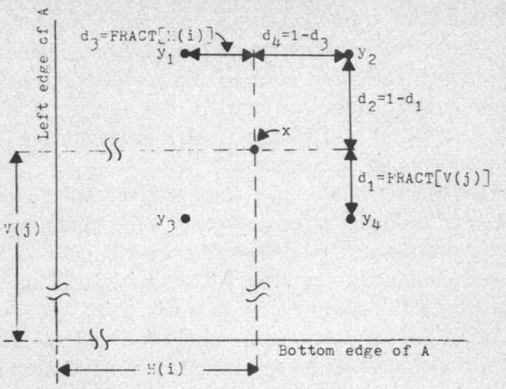

Fig. 3. Determination of gray level value for a point x in the linear picture.

Currently, the input area of 100 000 points resides on disk and accessing portions of it is very slow. Signal conditioning, character isolation and classification take approximately thirteen minutes for 250 characters.

Signal Conditioning

Linearization

When a picture is input from the TV camera there is a certain amount of consistent nonlinearity. This nonlinearity has been measured, and it was found that the distortions in the vertical and horizontal dimensions were relatively independent [4]. (Horizontal and vertical distortions that are not independent will cause a "pincushion" or "barrel" effect.) A reliable program, based on [4], was developed to compensate for independent nonlinearities.

Let i and j represent the position of a point in a picture where i is the horizontal distance from the left edge and j is the vertical distance from the bottom. Let $A(i,j)$ and $B(i,j)$ represent the gray level values of the points at position (i,j) in the nonlinearized and linearized pictures, respectively. The object of linearization is to determine the values of $B(i,j)$ from the values of $A(i,j)$.

Let $V(j)$ represent a vertical correction function such that row j in the linearized picture B may be found at row $V(j)$ in the nonlinearized picture A. Similarly column i in B may be found at column $H(i)$ in A. H and V are table lookup functions which were determined empirically by placing an equally spaced grid in front of the camera and measuring the distance between lines of the resulting picture. While i and j are integers, $H(i)$ and $V(j)$ generally are not. In other words, the gray level value for $B(i,j)$ can be found at position $(H(i),V(j))$ in A but $H(i)$ and $V(j)$ may be noninteger coordinates.

Referring to Fig. 3, it is desired to find the value of $B(i,j)$. To do this, we refer to position $(H(i),V(j))$ in A (point x). Since point x does not generally correspond exactly in position to an existing point in A, its value is determined from the four neighboring points in A (y_1, y_2, y_3, and y_4) by simple interpolation.

Let the function FRACT be the fractional part of a real number (e.g., FRACT[10.3] = 0.3) and INT be the integral part of a real number (e.g., INT[10.3] = 10). We see that

y_3 has coordinates (INT[$H(i)$], INT[$V(j)$])

y_1 has coordinates (INT[$H(i)$], INT[$V(j)$] + 1)

y_4 has coordinates (INT[$H(i)$] + 1, INT[$V(j)$])

y_2 has coordinates (INT[$H(i)$] + 1, INT[$V(j)$] + 1).

If we let $A(y_k)$ be the gray level value of point y_k, then the value for point x (the value we will assign to $B(i,j)$) is

$$B(i,j) = d_2[d_4 A(y_3) + d_3 A(y_4)] + d_1[d_4 A(y_1) + d_3 A(y_2)].$$

Subscripts i and j are varied over all values such that $H(i)$ and $V(j)$ remain within the dimensions of picture A, therefore B will generally have different dimensions than A.

Unshading

When any picture is input from the camera, the brightness of the picture varies considerably over its area in an essentially uncontrollable way. The varying brightness is caused by the light source, angle of illumination, camera sensitivity, camera lenses, etc. Since the picture will always be composed of lines of text there is a simple and effective way to normalize the picture with respect to uneven shading. This method will not work with arbitrary pictures because it assumes the picture is made up entirely of black and white segments (no gray) and that there are no large areas that are entirely black. The picture is divided up into a number of rectangles. It is assumed that each rectangle contains some white, this limits the minimum size of the rectangles. The largest valued (white) few points in each rectangle are averaged which gives an accurate indication of the shading of that rectangle. Since the rectangles are relatively small it can be safely assumed that within each rectangle shading is even. The larger the rectangle the less this assumption can be trusted; the smaller the rectangle, the better rapid shading variations can be accommodated. Now, since there is a measure of each rectangle's brightness or shading, every point in each rectangle is multiplied by the appropriate amount and the resulting picture has a very even, repeatable brightness.

Let a rectangular portion of the picture be defined as having corners (v_1,h_1), (v_1,h_2), (v_2,h_1), and (v_2,h_2) where $v_1 < v_2$ and $h_1 < h_2$. Suppose L is the unevenly shaded picture and S is the picture after unshading, then if w_k is the kth largest value in $L_{i,j}$, and

$$f = K \Big/ \sum_{k=1}^{m} (w_k/m)$$

where

$$m \simeq 20 \qquad K = \text{an arbitrary constant}$$

then

$$S_{i,j} = f \cdot L_{i,j}$$

where

$$v_1 \le i \le v_2 \quad \text{and} \quad h_1 \le j \le h_2.$$

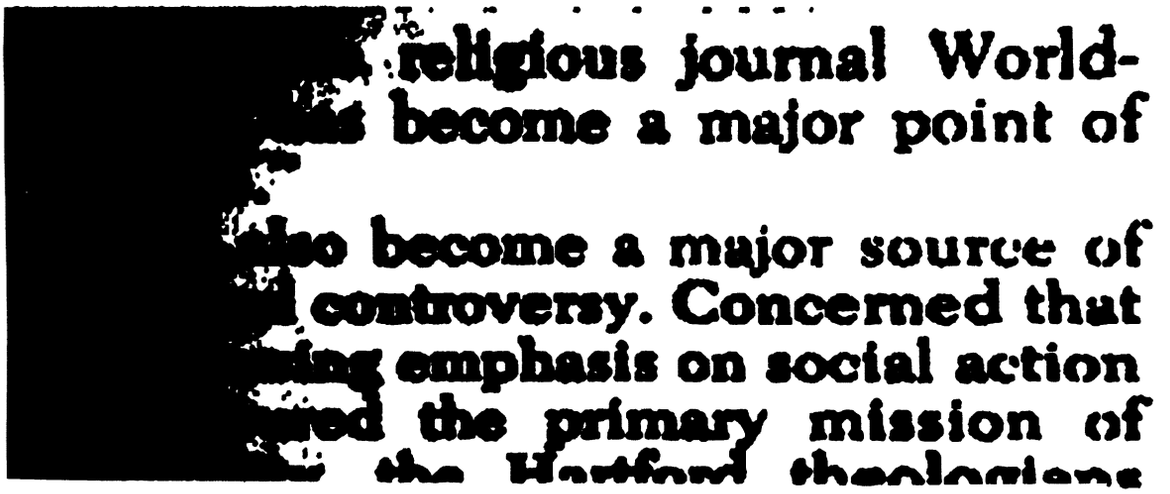

Fig. 4. Text as seen by the camera; note the uneven shading. The picture was quantized to binary values with a fixed threshold.

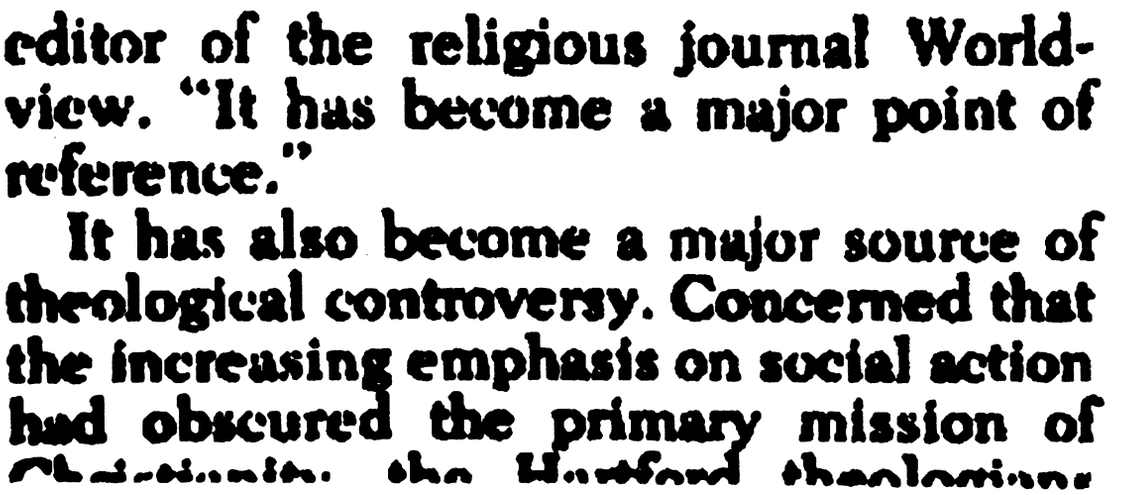

Fig. 5. The picture after linearization and unshading.

Now a single threshold can be passed over the entire picture and characters with even shading and thickness result.

Figs. 4 and 5 show a typical sample of text before and after signal conditioning, respectively.

Preprocessing

Line Isolation

Measurements have shown that if the standard deviation of each row of the picture is calculated, then those rows with the lowest standard deviations can accurately be used as interline markers [4]. Even short lines such as those at the end of paragraphs are accurately detected. The standard deviations are passed to the "line isolation" section which, using a threshold, then determines the interline markers. This section of the system assumes that the lines are all evenly spaced. This basic method for isolating lines was augmented with a heuristic error detection algorithm (Fig. 6).

Character Isolation

The typescript found in books and magazines is unlike that produced by a typewriter. Books and magazines have variable width characters and character separations, i.e., it is "unformatted" text. With unformatted text, three "i"'s can fit in the space of a "w," so character isolation is somewhat complex.

Isolation difficulties increase when either the quality of the text or the resolution of the camera falls to a level where adjacent characters appear to touch (Fig. 7). Normally, the end of a character is marked by a vertical white line that passes from the top of the line to the bottom. But if two adjacent characters touch, and the vertical white line criterion is used exclusively, then the two characters are isolated as one. There are so many possible combinations of adjacent characters that it would be impractical to recognize every character pair as a special class. On the other hand, an imperfectly printed character may contain a break (Fig. 7(c)) in which case it will be considered as two characters.

A solution to character isolation must deal with both split and merged characters. Since no amount of threshold adjustment could satisfactorily eliminate split characters (this was initially attempted), a decision was made to recognize the constituent parts of certain split characters as special classes. Later, during second level classification, these special classes would be converted back to their original characters. For instance, a "u" is frequently split in the middle (due in large part to the resolution of the camera), so there is a special class for the left half and a special class for the right half. In the second level classification section

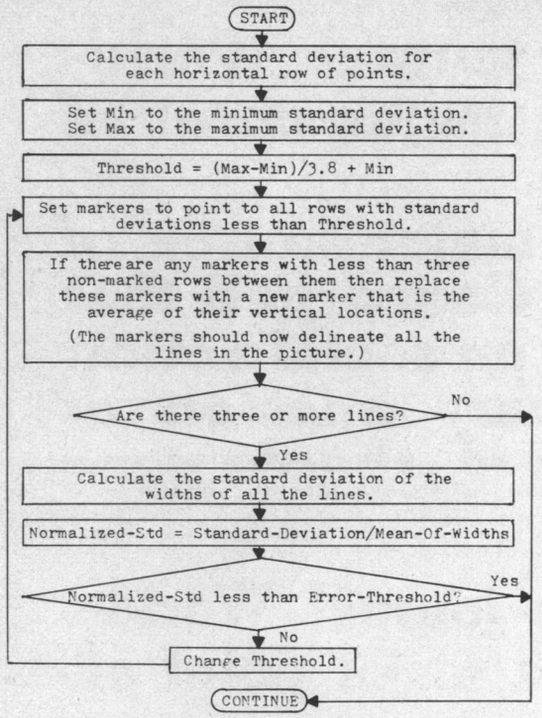

Fig. 6. Flowchart for isolating lines.

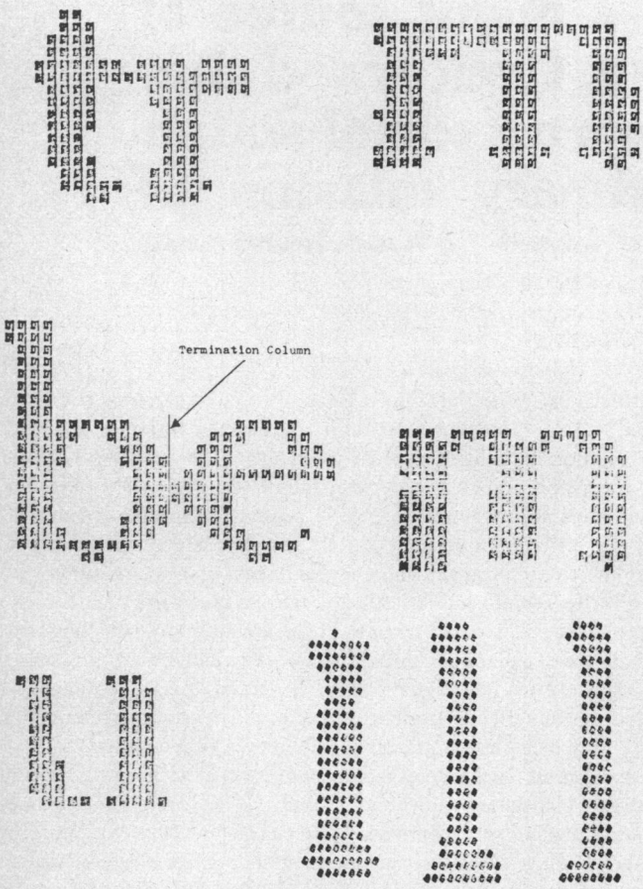

Fig. 7. Examples of difficult characters. (a) Merged characters that are thin enough to pass through width threshold. Recognized as a special character. (b) Merged characters too wide to pass through width threshold. Correctly separated at indicated column. (c) Split "u." Each half is recognized separately. (d) Merged "r" and "n" characters. Note the similarity to the "m." System has difficulty discriminating between the two. (e) "I," one, and "ell" characters, respectively. (f) An "m" character.

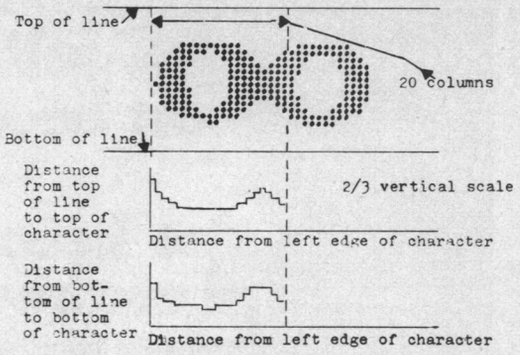

Fig. 8. Separation of merged characters.

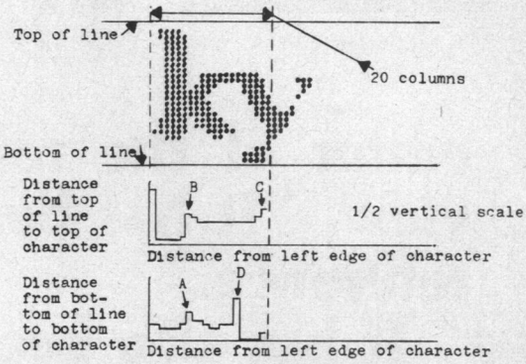

Fig. 9. The more difficult separation of merged "k" and "y" characters.

these two classes (assuming they are correctly recognized) are replaced by the "u" character. The idea of special classes can be extended to cover certain merged characters. For instance, an "f" followed by an "i" is always a single special character. The same holds true for two adjacent "t"'s, two adjacent "f"'s and others. These character pairs are not two characters printed close together, but actually a single special character (ligature). These character pairs are recognized as special classes and during second level classification are replaced by their correct constituent characters. The difficulties of merged characters, other than those special character pairs, still exist.

A threshold was set on the maximum width of a character. If a line is thirty points high then this threshold (for the font in *Newsweek*) was found to be about twenty columns. If there are more than twenty columns without a vertical white line then it is assumed that two or more merged characters have been encountered.

A decision must be made about where to terminate the character if it extends past twenty columns. The topological algorithm is based upon the outside contour of the characters. In Fig. 8 the contours of two touching "o"'s are plotted. The peaks from each graph are extracted (excluding peaks lying on the Y axis) and the peak closest to column 12 (approximately) is chosen. The use of column 12 was determined empirically, as was the number 20 for the maximum number of columns. It was felt that if any peaks extracted from the first and second graph correspond to the same column (as would be likely in this case) then this column should be more reliable than singly occurring peaks. Therefore, this column is weighted a bit more favorably than a column which corresponds to a peak in only one graph. The more difficult separation of the characters "k" and "y" is shown in Fig. 9. In this case there are two peaks from the top and two peaks from the bottom.

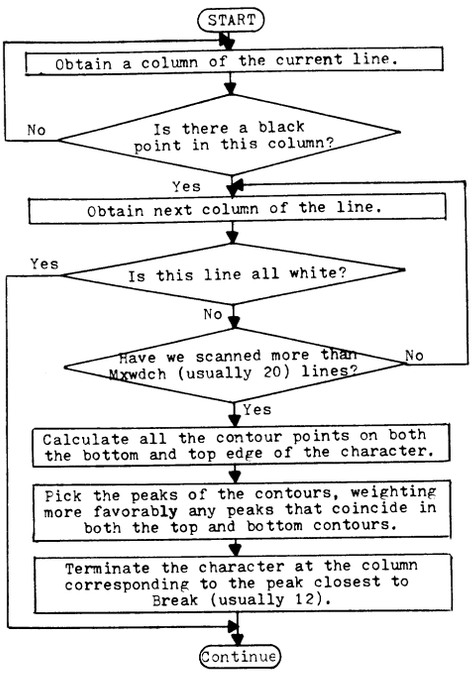

Fig. 10. Flowchart for the segmentation of characters.

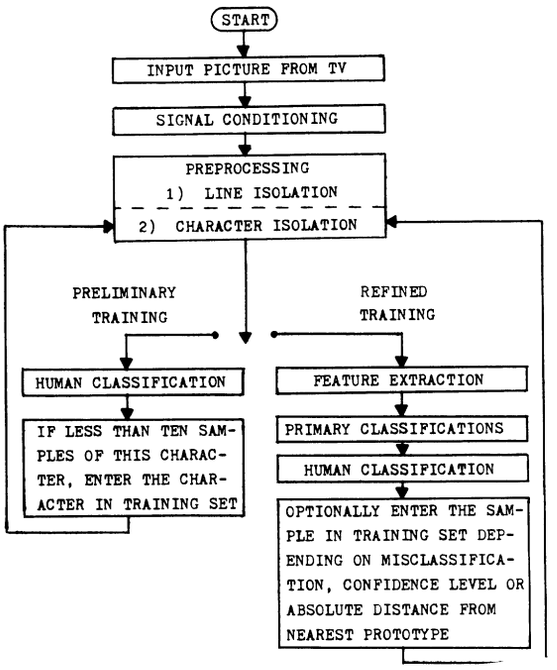

Fig. 11. Flowchart for training the weighted minimum distance classifier.

Peak D will normally be the closest to column twelve (the reason for which twelve was chosen). Unfortunately, in this case peaks A and B occur at the same column and as such are weighted more favorably (the favorable weighting for coinciding peaks is justified in most circumstances). But this weighting is not enough to overcome its much greater distance from column twelve. Therefore, the column corresponding to peak D is chosen to terminate the character.

The number twenty for the maximum width of a character was chosen so that very few single characters were larger than twenty while a minimum number of thin merged characters would pass through the threshold. Some characters such as "m"'s and "w"'s, though, are usually greater than twenty columns wide. These characters are then split up, their constituent parts recognized as special classes and then during second level classification, the special classes are replaced by the original character (Fig. 10).

Classification

A weighted minimum distance classifier was chosen for this section because of its simplicity. Let M_i^j and V_i^j be the ith feature mean and ith feature variance, respectively, of the jth class. These values are determined from the training samples. Let

$$F = [f_1, f_2, \cdots, f_N]$$

be a feature vector representing an unknown class. Then F will be classified as belonging to class j if and only if for all k such that $1 \leq k \leq C$

$$D_j \leq D_k$$

where

$$C = \text{number of classes}$$

$$N = \text{number of features}$$

$$D_k = \sum_{i=1}^{N} (f_i - M_i^k)^2 / V_i^k.$$

Second level classification performs two functions; first, it processes any special classes introduced by character isolation or classification and second, it converts certain punctuation into a form acceptable by the text to phoneme translation program.

Data Preparation

All training samples were taken from six different 1975 issues of *Newsweek* magazine. Training was divided into two stages, preliminary training and refined training. For both preliminary and refined training, a sample of text, chosen for its relatively higher content of infrequent characters, is placed in front of the camera. The picture is digitized and the two signal conditioning programs are applied. Line and character isolation are then performed.

For preliminary training, the isolated character is displayed on a CRT and the operator enters the appropriate class designation from the keyboard. For refined training, in addition to the application of the signal conditioning, line isolation, and character isolation programs, feature extraction and primary classifications are performed on each sample. The data base is expanded with particular emphasis on correctly isolated characters that are either misclassified or have a low confidence level (Fig. 11).

Experimental Results and Comments

Approximately 94 percent of the words spoken by the system are intelligible by any listener familiar with the machine's "accent." The text to phoneme conversion program and voice synthesizer account for about 3 percent of the errors while character recognition accounts for he other half of the errors. Note that one misclassified character in a word will generally render the word incomprehensible.

To demonstrate the generality of character segmentation and classification, the classifier was trained on 2660 formatted alphanumeric characters from an IBM "Elite 72" font type ball according to the methods of the last section. Upper and lower case letters, numerals (except "one" which is the same character as "ell") and periods, commas, apostrophes, stars, and question marks were

used (66 classes). The system was then tested on 20 samples of each character (1320 testing samples total) which were not in the training set. There were 16 errors (98.8 percent recognition) in classification (all errors were capital "o"'s misclassified as zeros) given correctly segmented characters. Segmentation was 99 percent correct with the use of no special classes.

Results have been encouraging but there are many directions for improvement. A truly useful system, one for commercial applications, would demand better performance in speed, accuracy, and generality.

Signal conditioning takes approximately 5.25 minutes for a 200 by 500 point picture. A large part of this time is attributable to disk transfer latencies of the disk-resident picture.

The speed of the rest of the system depends on the number of characters input. Character isolation and feature extraction occur together and take about 3.4 minutes for 250 characters. At least 40 percent of this time is a result of the picture's residency on disk.

Primary character classification requires 3.5 minutes of straight computation for 250 characters. Second level classification, text to phoneme translation, and voice synthesis are all performed in better than real-time.

For seven lines of text, from one column of *Newsweek* (about 250 characters), thirteen minutes elapse between the time the picture is input and the text is read aloud.

Errors in the system can be attributed to four areas: character isolation (which in turn depends on the input quality and preprocessing programs), character classifications, text to phoneme translation, and phoneme synthesis. Character isolation, given an adequate number of special classes (about 70 for *Newsweek*), achieves about 99.5 percent correct separations on running text. Merged characters terminated at the wrong column and new types of split characters produce the majority of errors.

Text to phoneme conversion, together with the phoneme synthesis, was claimed to produce about 97 percent acceptable speech on running text [2]; that is, 97 percent of the words could be understood, in context, by a listener familiar with the program and the Votrax Voice Synthesizer.

Objective measurements of the system's intelligibility have not yet been made, but the laboratory and software needed for comprehensive testing is available [5].

Acknowledgment

The authors would like to extend their greatest thanks to Dr. M. D. McIlroy and the Bell Laboratories for making available, in its entirety, the program for the translation of English words into phonemes.

References

[1] VS-6 Votrax Voice Synthesizer, Vocal Interface Division, Federal Screw Works, Detroit, MI.

[2] M. D. McIlroy, "Synthetic English speech by rule," Computing Science Tech. Rep. 14, Bell Laboratories, Murray Hill, NJ, Mar. 1974.

[3] K. Spikowski, "A television camera input system for a digital computer," M.S. thesis, Case Western Reserve Univ., Cleveland, OH, June 1972.

[4] K. Mantey, "An automated feature selection algorithm and application to printed character recognition," M.S. thesis, Case Western Reserve Univ., Cleveland, OH, Aug. 1973.

[5] W. C. Lin and A. Agrawal, "Mini-computer based laboratory for speech intelligibility research," *Proc. IEEE*, vol. 61, pp. 1583–1588, Nov. 1973.

[6] R. N. Ascher, G. M. Koppelman, M. J. Miller, G. Nagy, and G. L. Shelton, Jr., "An interactive system for reading unformatted printed text," *IEEE Trans. Comput.*, vol. C-20, pp. 1527–1543, Dec. 1971.

[7] R. Kooi, "An on-line mini-computer based system for reading unformatted printed text aloud," unpublished M.S. thesis, Case Western Reserve Univ., Cleveland, OH, June 1976.

Smoothing Algorithms for Two-Dimensional Image Processing

TOHRU KATAYAMA, MEMBER IEEE, AND MICHITAKA KOSAKA

Abstract—**The smoothing algorithms for images that can be modeled by a discrete, two-dimensional random field with a separable autocorrelation function of exponential form is considered. By using a state–space representation for the image, a recursive image processing algorithm is derived by applying the fixed-lag smoothing result. A simplified smoothing algorithm is also obtained by use of an orthogonal transformation to the coordinate of the state space. Digital simulation studies are carried out to show the feasibility of the present approach.**

I. Introduction

An image processing or enhancement is to improve the quality of the image that is distorted by the inaccuracy of sensing instruments and/or the noise incurred during the transmission [1]. Any monochromatic image can be modeled by specifying the gray level or the brightness at each grid point in the spatial domain. Therefore, the digital image processing deals with the data that is naturally represented by a double indexed sequence. But this two-dimensional nature of pictorial data makes the data processing very difficult. To alleviate the difficulty associated with two-dimensional data processing, recursive image processing techniques have been presented by Nahi, Assefi, and Franco [2]–[4]. In these papers, two-dimensional image fields are transformed into scalar or vector stochastic processes by scanning, and then the Kalman filtering technique [5] is applied. Jain and Angel [6] obtained a smoothing algorithm for image enhancement by using the nearest neighbor model for the image field. Alternative approaches based on a double indexed linear dynamic system are developed by Habibi [7] and Attasi [8].

In this correspondence, we derive smoothing algorithms for the two-dimensional image processing by using the nearest neighbor model due to Jain and Angel [6]. Since the nearest neighbor model is a discrete version of a two-dimensional diffusion, it well takes into account the property of the homogeneous random field [9]. As shown in Section III, however, this model leads to an unstable linear dynamic system with a time delay; this fact was not well recognized in [6]. For deriving smoothing algorithms, we have applied the fixed-lag smoothing result due to Priemer and Vacroux [10]. A simplified algorithm is also derived by applying an orthogonal transformation to the coordinate of the state space. A digital simulation study is carried out by using an artificial image to show the feasibility of the present approach.

II. Model of Image

Let $x_{n,m}$ be the gray level of the original image at spatial coordinate (n,m), where n and m denote the vertical and horizontal position variables, respectively, and $n = 1, \cdots, N$; $m = 1, \cdots, N$. Since the ensemble of such images can be regarded as a two-dimensional homogeneous random field with a second-order

Manuscript received March 14, 1977; revised July 19, 1977. This work was supported in part by the Ministry of Education, Japan, under the Special Project Research on "Evaluation and Control of Environmental Systems" Grant No. 011913.

T. Katayama is with the Department of Applied Mathematics and Physics, Faculty of Engineering, Kyoto University, Kyoto, Japan.

M. Kosaka was with the Department of Applied Mathematics and Physics, Faculty of Engineering, Kyoto University, Kyoto, Japan. He is now with the Systems Development Laboratory, Hitachi Ltd., Totsuka, Yokohama, Japan.

0018-9472/78/0100-0062$00.75 ©1978 IEEE