OPTIMIZING A VIDEO PREPROCESSOR FOR OCR

M R
IBM Systems Dev elopment Division
Rochester, Minnesota

Summary

This paper describes how optimal video preprocessor performance can be achieved using a software recognition system and a set of controlled experiments. It basically involves optimizing on a constant threshold and then using recognition logic designed from that threshold to optimize an adaptive threshold function or threshold operator. By using this technique, the video preprocessor can be optimized early In the development of an OCR (Optical Character Recognition) machine. The software recognition system is described and experimental data is presented to illustrate the procedures.

Introduction

There are three basic stages in the data path of an optical character recognition (OCR) machine; scanning, video preprocessing, and recognition. The video preprocessor configuration could be one of many possible configurations, but, in general, its function is to improve the image quality prior to the actual recognition process. In the video threshold type of preprocessor discussed here, the analog video is compared to an automatic or adaptive threshold derived from intelligence in the video signal.

Since, for many OCR applications, a wide range of print quality distortions can be expected on the input documents, the circuitry deriving a threshold from video obtained from these documents becomes necessarily complex and difficult to optimize in terms of recognition performance. Optimization techniques in the past have consisted of analysis of video output signals and of two-dimensional video bit patterns, and evaluation using the recognition circuitry of the machine for which the thresholding scheme is being designed. From either of these evaluation techniques one can determine the type of automatic threshold function needed, implement it in hardware, and then repeat the evaluation to determine if the output has improved. If readjustment to the first function is needed, the process must be repeated. This continues on a trial and error basis until the best results are achieved. Use of the recognition circuitry as an evaluation tool has proven to be the most fruitful technique.

This paper describes how optimal video preprocessor performance can be achieved using a software recognition system and a set of controlled experiments. Even though a specific type of preprocessor is assumed here, the techniques can be applied to other preprocessors as well. It basically involves optimizing the recognition performance on a constant threshold and then using recognition logic designed from that threshold to optimize the adaptive threshold functions. Recognition reject rates are used to measure improvement or deterioration In image quality at the preprocessor output. The recognition reject rate, therefore, provides the basis for determining when the preprocessor is optimal.

General Description of Video Preprocessor

The relationship of a video preprocessor to the entire OCR system is shown in Figure 1. An optical scanner provides an analog representation of the scanned image. Circuitry, called threshold operators or adaptive threshold functions, measures certain distinguishing characteristics of the video (average contrast, peak control pulse width, etc.) and produces a threshold to which the analog video is compared using a voltage discriminator. The voltage discriminator output, or amplitude quantized video, is then time sampled and stored in a shift register for recognition. The threshold operator may also use the information stored in the shift register. For a thresholder to perform optimally when the input consists of a wide range of print quality distortions at least one, and usually more, threshold operators are needed.

The operator outputs T_1 T2, T3, T4 in Figure 1 are combined to form a final video threshold. To show how Ti might be produced, consider an operator which measures average contrast $\overline{V}$ over previously scanned characters. The threshold T^ could then be generated by making it a linear function of V,

$$T^1 = K\overline{V} + C_1 \qquad (1)$$

The threshold T_1 would then respond to changes in V and this would be reflected in the final video threshold along with other similarly generated operator outputs.

The basic hardware design of the threshold operators depends heavily on the type of optical scanner being used. Generally, however, the operators will measure contrast, character line width, or any other conceivable video characteristic. Another design parameter is the area of threshold Influence. This may be a local area about a point or an area encompassing several characters.

Video thresholding is used to reduce the variations in the thresholder output caused by print quality distortions of the scanned image. Once the threshold operators are designed, considerable effort still remains in optimizing parameters, such as K and C_1 in Equation 1, for each operator and for other parameters which are involved in mixing the operator outputs to form a final video threshold. Difficulty arises when a subjective decision is made as to when a variation in thresholder output has been significantly reduced by a particular threshold operator. Another difficulty is that a particular threshold operator may have bad as well as good effects on certain types of scanned images. In other words, the thresholder itself may introduce distortions into the video. Here, again, some means of measuring the severity of the distortion is needed.

A thresholder will be optimal when it has adapted to the widest print quality range possible. Thus, the parameters in each operator must be adjusted for the particular type of distortion the operator was designed to improve. The outputs of each operator need to be mixed in such a way that the thresholder adapts to the different types of print quality distortions which occur in the input.

Thresholder Optimization

The basic tool used here for evaluating the thresholder type of video preprocessor is a software character recognition system. Hardware or software recognition could be used, but there are certain advantages to the latter. With a software recognition system, it is not necessary to wait until hardware is designed and built before optimizing the preprocessor. Preprocessor optimization can be completed prior to designing hardware and, therefore, the logic designers can converge more rapidly to an optimal hardware recognition logic design.

For this application, an adaptive computer program was used to automatically design recognition logics. These logics were then used by the software recognition system. Use of the adaptive program allowed the logics to be easily adapted to a large number of scanning devices and also to a wide range of print quality.

Optimization Technique

Print quality distortions are introduced from four relatively independent sources in the document preparation process; the printer, the ribbon, the scanner itself, and the document handling process. For a conceptual understanding, some hypothetical quality function Q can be defined as a measure of the image quality which finally reaches the recognition system. Q could then be represented as the product of four independent quality functions.

$$Q = Q_p \times Q_r \times Q_h \times Q_s \qquad (2)$$

where p, r, h and s denote printer, ribbon, document handling and scanner, respectively. Q would be maximum when each stage passed the image with minimum distortion.

Attempts have been made to define a unique measure of quality, such as Q, which can be related to measurable characteristics of the image and ultimately predict the performance of an optical character reader. Most of these attempts have been only mildly successful. For example, the average contrast or average stroke width of an image has been used as a measure of quality. These are only two of the many image characteristics which are altered by print quality distortion, and measuring these two alone cannot possibly provide a unique measure of print quality. The evaluation technique described here uses a character recognition system to provide a measure of Image quality.

It should be clear that a wide range of print quality will occur over the lifetime of a ribbon for any given printer The varying contrast levels which occur during this period of time constitute a print quality distortion, and, in addition, they modify the severity of distortions introduced by the printer itself. Shaded printing (light tops with dark bottoms), for example, which normally is a distortion introduced by the printer may be a serious problem to a recognition system only with very old ribbons or when contrast levels are low. On the other hand, if the impression is set too high on a particular type slug, this may only cause recognition problems at high contrast levels where smudged images or even background noise would be expected. The first step in preprocessor evaluation then is to check its performance at different points during ribbon life.

The number of lines printed is used as a measure of ribbon age. To evaluate the dependency of image quality on ribbon age, the number of characters rejected by a recognition system is calculated over thirty intervals of ribbon age. ΔN will denote the interval and $R(N)_m$ will be defined as the number of rejects which occur between $(N - \Delta N)$ and N. The three curves shown in Figure 2 illustrate the dependency of $R(N)_m$ on a constant video threshold (a threshold which is independent of all contrast variations).

The shape of the function $R(N)_m$ is used to determine which of the three thresholds is best. Since it is the goal of the preprocessor to produce an output which has the least amount of variation at different points in ribbon life, Curve b would represent the best choice. It shows the least amount of deviation throughout ribbon life from the minimum reject rate. This choice becomes obvious after observing the extremes in image distortion which occur early in ribbon life for a low threshold, or late in ribbon life for a high threshold.

It is possible to express the conditions for an optimum constant threshold in mathematical terms. By differentiating Curve b of Figure 2, it is possible to determine two points of inflection, N_1 and N_2, which can be used to define boundaries of three contrast ranges. N_1 is the first point at which $dR(N)_m/dN \le 0$ for increasing values of N. N_2 is the first point at which $dR(N)_m dN \le 0$ for decreasing values of N. The three contrast ranges can then be defined as

$$R_H = \sum_{m=1}^{m=N_1/\Delta N} R(N)_m \quad \text{High Contrast Rejects} \qquad (3)$$

$$R_M = \sum_{m=N_1/\Delta N}^{m=N_2/\Delta N} R(N)_m \quad \text{Medium Contrast Rejects} \qquad (4)$$

$$R_L = \sum_{m=N_2/\Delta N}^{m=30} R(N)_m \quad \text{Low Contrast Rejects} \qquad (5)$$

where m is a positive integer ranging from 1 to 30. The optimum constant threshold will occur when $R_H = R_L$.

All discussion to this point have been concerned with a constant video threshold. While this is the most elementary, selecting the optimum constant threshold should be the first design step. When new threshold operators are designed, new values of R_H, R_M, and R_L can be calculated and compared against the old to give a measure of success or failure.

Still another factor in favor of first optimizing on a constant threshold Is that it gives a data set with a proportionate number of high and low contrast problem patterns on which a software recognition logic can be designed. This is an important consideration because the recognition logic will be used for additional preprocessor evaluation. The reliability of a recognition logic design on extremely distorted data such as that which produced Curves a or c of Figure 2 would be very questionable.

Video Evaluation Program

Even though the function $R(^N)$ is very simple in concept, it is very time consuming to determine by manual calculation. For example, consider an optical character reader for which the average reject rate over ribbon life time is required to be 0.1% (1 reject per 1000 characters). Also assume that it is necessary to calculate the reject rate at thirty points throughout ribbon life to accurately determine $R(N)_m$. It seems reasonable to require at least 100 rejects on the average for each point to minimize statistical fluctuations. This means that 3×10^6 characters need to be scanned (1000 x 100 x 30) and 3000 reject patterns have to be analyzed. Thus, a computer program was written to calculate $R(N)_m$.

Several other features were incorporated into the program to allow rapid determination of predominate pattern distortions. In addition to calculating $R(N)_m$, a similar function was calculated for each character class. This allowed determination of which classes, if any, were significantly more troublesome than others. Also the video patterns which were stored on the input tape were sorted according to class and then stored on the output tape. The output patterns can then be printed out in order of increasing values of N so that the type of distortion can be observed. This is necessary in order to determine what type of threshold operator needs to be designed.

The program also calculated the number of high, medium, and low contrast substitutions. Since the number of substitutions is normally small, it is not possible to determine a reliable relationship between substitution errors and ribbon age. The substitution patterns, however, were stored on the output tape so they could be printed for visual inspection.

Input Documents

Input documents which are scanned for preprocessor optimization must be representative of those which the machine is finally expected to handle. If the machine is expected to operate in an uncontrolled print quality environment, documents printed with low quality ribbons, worn out ribbons, and poorly adjusted printers can be expected. For machines that are designed to perform in a so-called controlled environment, the print quality is considerably better but the machine must still be able to handle a large number of print quality distortions.

Documents can be printed for laboratory testing to simulate the range of print quality distortions a machine is expected to handle. For this evaluation technique, the documents must be sampled from the printer output at equal intervals throughout the lifetime of a ribbon. Ribbon lifetime is specified for an OCR ribbon as the number of lines that can be printed with a ribbon before print quality deteriorates to an unacceptable level for OCR. However, ribbons are often used beyond their specified lifetimes in customer applications, but this condition is easily simulated in the laboratory. Scheduled maintenance is normally recommended for printers used in OCR applications and here again, if the maintenance is not done, print quality will deteriorate. To simulate this condition several batches of documents (one batch per ribbon) can be printed without adjusting the printer.

These simulation techniques would apply for multifont applications as well as single font. In multifont applications, there are usually many more printing devices each introducing print quality distortions characteristic of a particular printing mechanism. A particular type style itself may be the source of a unique print quality distortion. To minimize the number of documents required for preprocessor optimization, a preliminary study should be made to determine which printers and fonts can be expected to occur most frequently.

Pattern Collection

A two-dimensional bit pattern of the scanned image is essential for the optimization technique. This, of course, implies that the document transport, scanner and CPU interface be made operational prior to preprocessor optimization. A pattern collection program need also be written to store the two-dimensional bit patterns on magnetic tape. This program would be dependent on a particular machine configuration. All programs used in the optimization scheme beyond this point can be adopted to a variety of video patterns making them useful on any OCR machine.

Segmentation and Registration Program

Pattern segmentation, another type of video preprocessing must be completed before the recognition process. A computer program was written to perform segmentation as well as the pattern registration required by the adaptive program. A sampling feature was included in the program to reduce the amount of data which had to be analyzed by the recognition system. The input documents were divided into small groups with M documents in each group (M can be chosen by the designer).

The segmentation algorithms were kept quite simple to keep computer time minimal. This simplicity caused segmentation errors, but these could be detected automatically by comparing the number of segmented characters per line to the number of characters printed on the document. Error lines determined in this way were not used as input to recognition.

Segmentation errors which could not be detected in the above manner occur infrequently enough for single font applications so as not to significantly affect the final calculation of recognition error rates. This may not be true for multifont applications, but it would be possible to manually eliminate those recognition errors caused by Improper segmentation from the final statistics. Another alternative for multifont machines is to design a sophisticated hardware segmentation system.

Optimization Steps

All of the optimization steps discussed are summarized in a flowchart shown in Figures 3 and 4. The process consists of three phases. During Phase I the optimum constant threshold is determined. This provides a recognition logic capable of handling a reasonable print quality range that can be used for additional optimization in Phase II. One set of documents (about 10, 000 lines) Is required and the same document can be used during Phase I and Phase II.

The documents should be printed with that device which is expected to produce the largest percentage of input for the particular OCR application under consideration. The same is true of the font with which the printer is equipped in the case of multifont applications. The printing device should be properly adjusted so that Its output is typical rather than worst case. This prevents designing threshold operators around distortions which are characteristic of a single printer. Program running times are indicated in Figure 3 and apply to an IBM System/360 Model 30. The number of lines and characters required are also indicated and were arrived at from statistical considerations.

During Phase II (Figure 4), the threshold operators are designed, implemented and evaluated. The criterion for successfully completing Phase II is to make $dR(N)_m/dN = 0$ for all values of N.

A third phase is included to extend the evaluation to more printers of the same type used In Phase I and II and also to other printing devices. This will expose other print quality distortions and if they occur frequestly enough will require the design of additional threshold operators. New fonts can also be investigated in a multifont application. This would require first scanning the documents at the constant threshold derived in Phase I so that a new logic can be designed which Is capable of recognizing the character shapes presented by the new font.

Experimental Results

In order to check the computer programs and techniques described In Figure 3, all of the steps of Phase I were carried out.

A sample of the printing which was scanned for the evaluation is shown In Figure 5. An IBM 1403 NI high-speed printer was used to print 97, 200 lines, and a l/4-length ribbon was used to reduce the number of documents which had to be printed. This represents 1-1/2 times the number of documents recommended for customer applications. (250, 000 lines is the recommended lifetime for a full length ribbon.) The impression setting of the printer was also set to its maximum value. Only half of the printed lines were scanned and this number was again reduced by a factor of ten In the segmentation program prior to recognition (there is a sampling feature in the segmentation program). The recognition logics were designed on 11, 000 patterns sampled uniformly throughout the total number of lines printed.

Print contrast levels were measured on the documents and the results are shown in Figure 6. Contrast is defined by the equation

$$C = \frac{R_P - R_I}{R_P} \times 100\%$$

where

C = Percent contrast

R_I = Ink reflectance

R_P = Paper reflectance

Contrast levels take on a range of values over the document and even over the character. This is illustrated in the figure by upper and lower limits. Notice that the lower limit approaches zero for large values of N. This may or may not be a problem depending on whether the contrast goes to zero in an area of the character that is critical to its identification.

The electroptic device used to scan the documents had a square sampling aperature 5 x 10^{-3} in. wide. The video was time-sampled at a rate which was equivalent to 200 samples/inch on the document in both the horizontal and vertical dimensions. Amplifiers and threshold circuits with appropriate gain and bandwidth were used. A constant threshold equivalent to 20% contrast was used. Table 1 summarizes the parameters involved in this experiment.

Table 1 Experiment Parameters

Printer	IBM 1403 - N1
Ribbon	1/4 length
Lines printed	97,200
Lines scanned	48,600
Lines passed through recognition	4,860
Character passed through recognition	252,720
Characters in design set	11,000
Constant threshold level	20% contrast

Figure 7 shows the relationship between the character reject rate and N.

$$\text{Character reject rate} = \frac{R(N)_m}{\Delta N \times \text{no. of character/line}} \times 100\% \qquad (7)$$

It is obvious from the shape of the curve that the 20% contrast threshold is too high and the documents should be rescanned at a lower threshold in order to produce a symmetrical curve similar to Curve b in Figure 2. Figure 8 shows an example of some reject patterns which contributed to the high reject rate for large values of N.

Conclusion

The techniques and computer programs described in this report provide a means of optimizing the video preprocessor early in a machine development program. Recognition error rates need not be within final machine specification during the optimization process because they are used as a relative measure of image quality. Once optimal performance is achieved in this manner a sophisticated recognition system can be designed which performs within machine specifications.

Acknowledgements

The author would like to acknowledge Messrs. C D Tulledge and R Baumgartner both of the IBM Systems Development Divison Laboratory, Rochester, Minnesota, for their programming assistance.

References

1. Bartz, M. R., Video Preprocessor Design for the IBM 1975, IBM Journal of Research and Development, Vol. 12, No. 4, September, 1968.

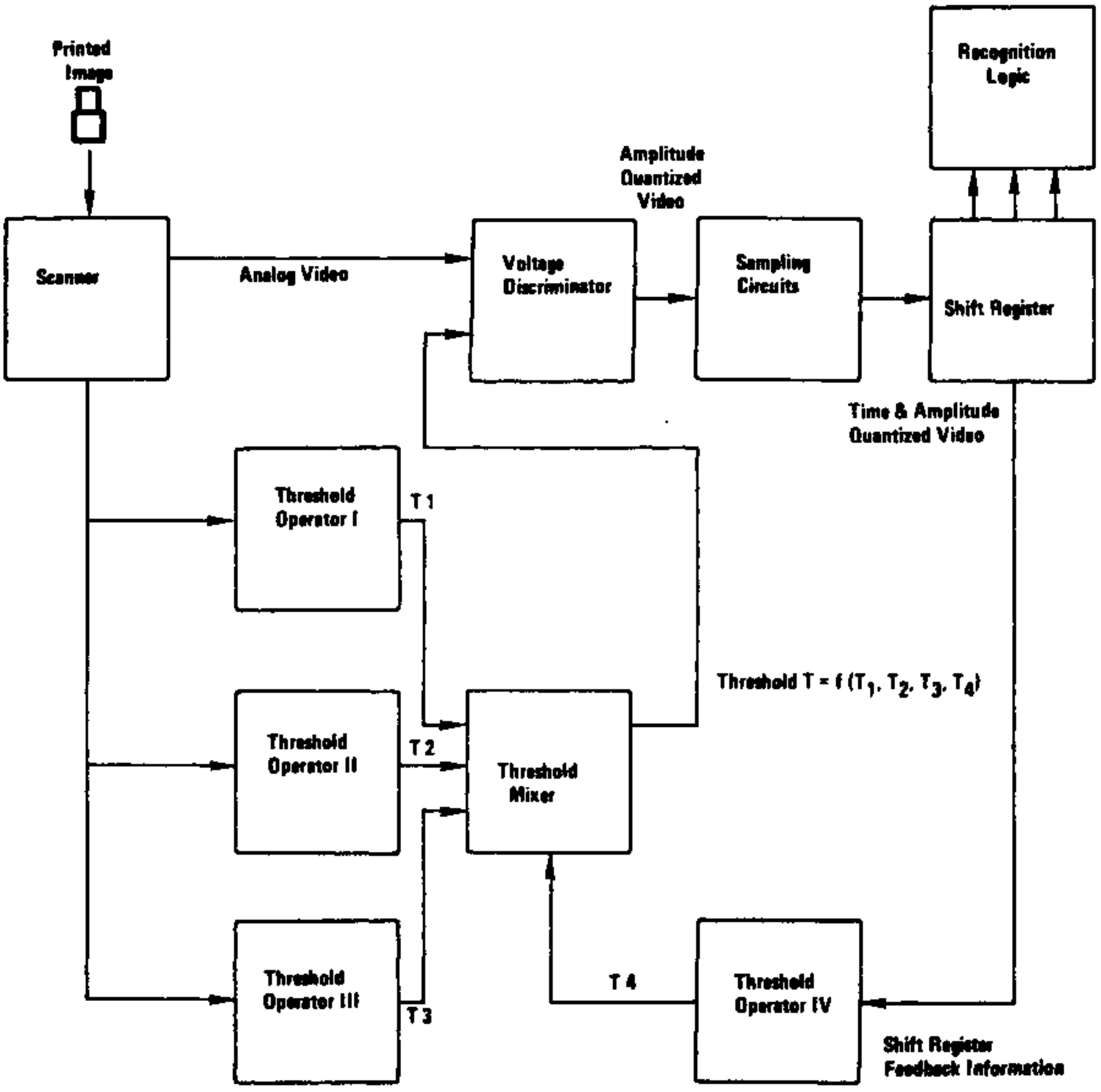

Figure 1 Relationship of video preprocessor to OCR system

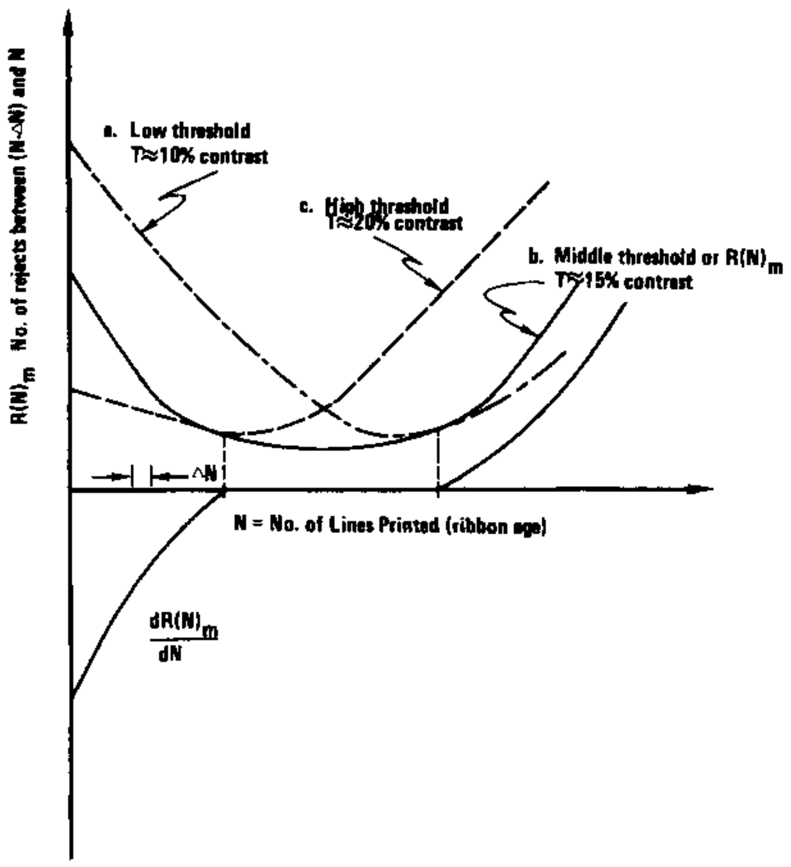

Figure 2 $R(N)_m$ at three constant thresholds, and $R(N)_m$ and $dR(N)_m/N$ vs N (shows inflection points)

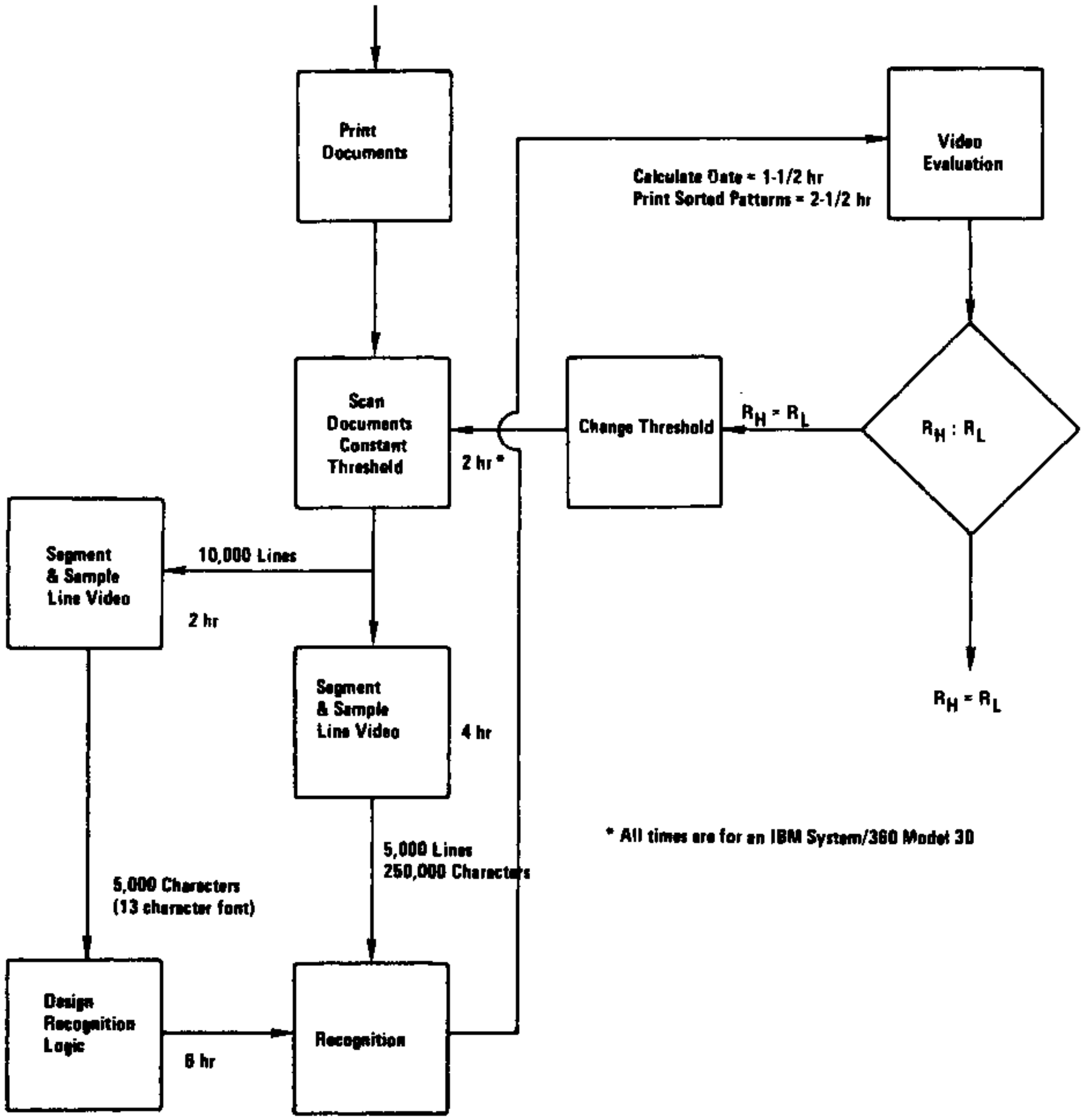

Figure 3 Phase I of preprocessor optimization

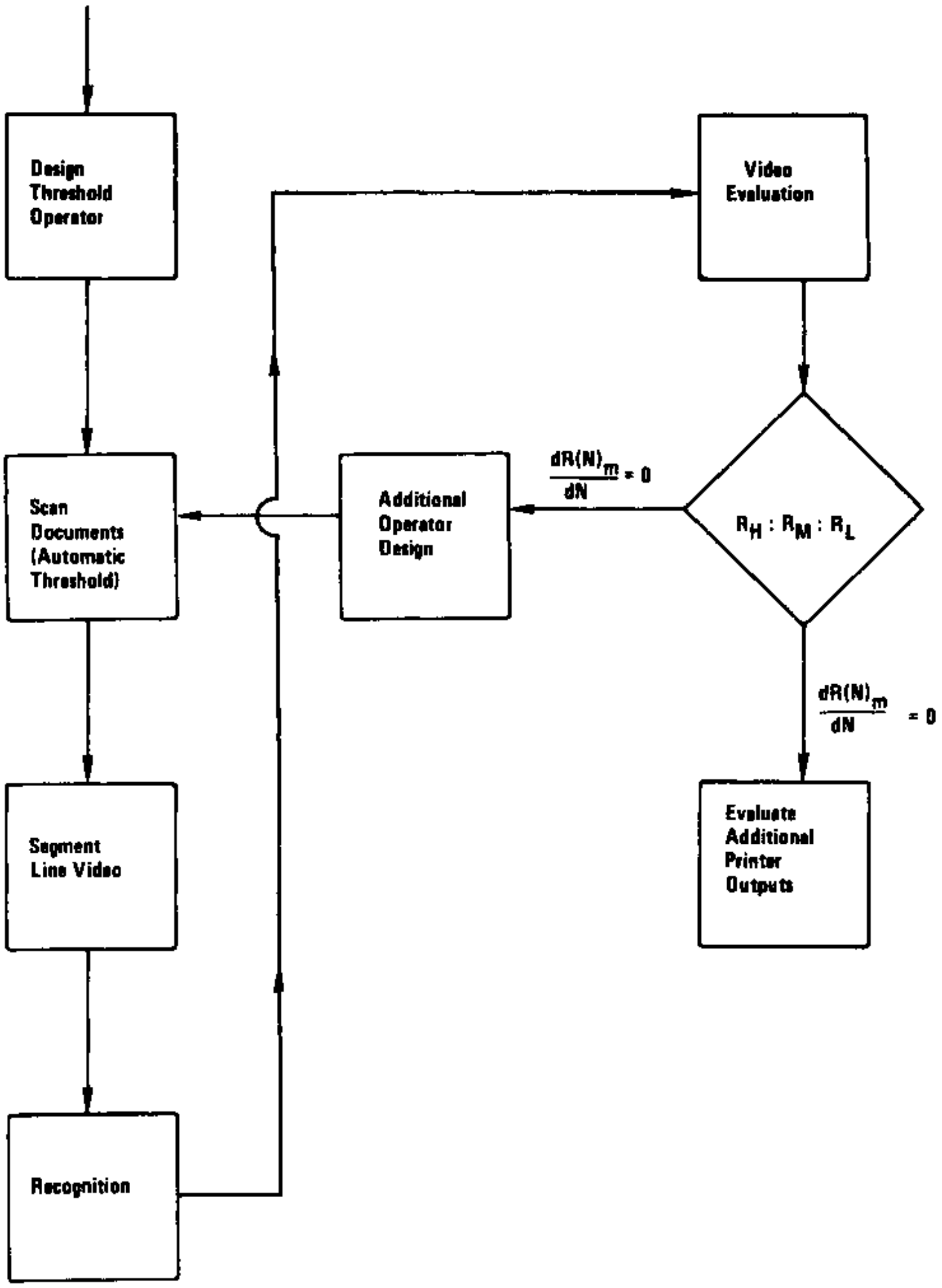
Design Threshold Operator
Video Evaluation
Scan Documents (Automatic Threshold)
Additional Operator Design
$\frac{dR(N)_m}{dN} \neq 0$
$R_H : R_M : R_L$
$\frac{dR(N)_m}{dN} = 0$
Segment Line Video
Evaluate Additional Printer Outputs
Recognition

Figure 4 Phase II and III of preprocessor optimization

```
        1029H38Y47J564657Y83J9201H    1234J5678H90Y098765H43Y21J

811698

        1029H38Y47J564657Y83J9201H    1234J5678H90Y098765H43Y21J
```

Figure 5 Sample printing from IBM 1403 N1

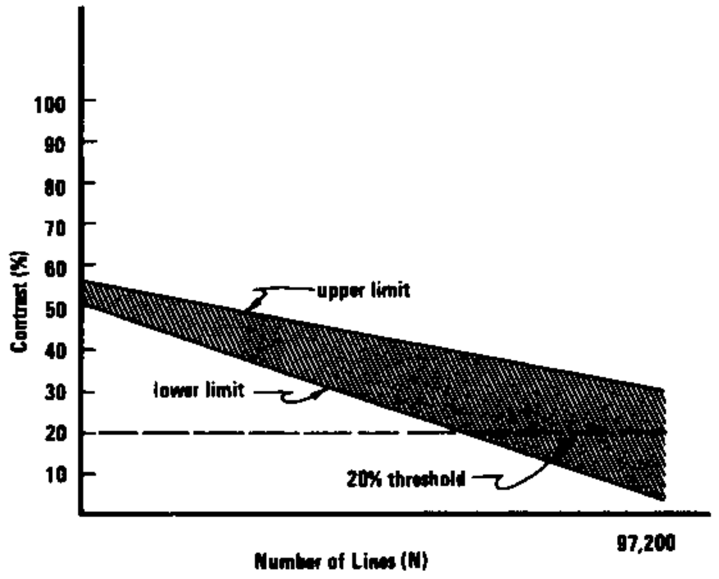

Figure 6 Contrast vs ribbon age

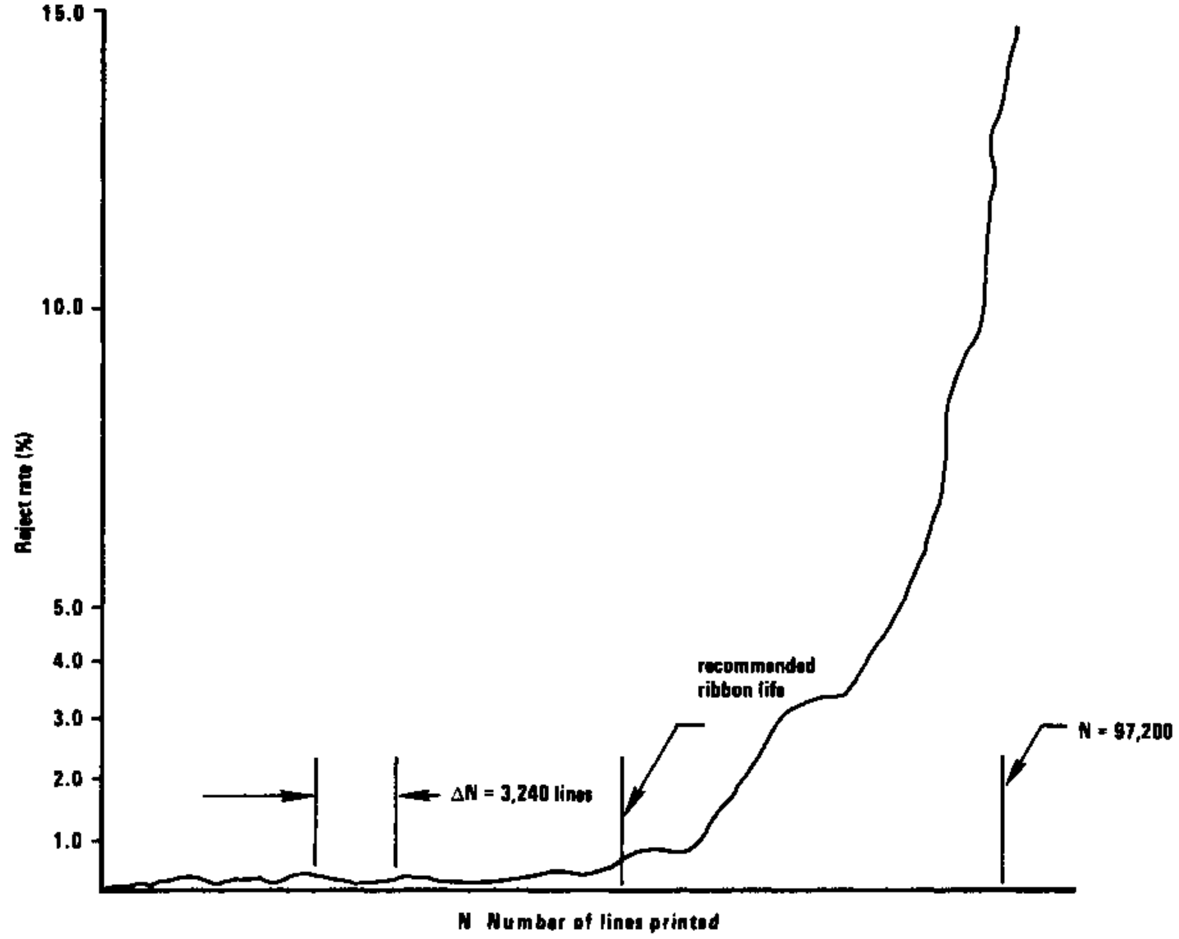

Figure 7 Relationship between character reject rate and N

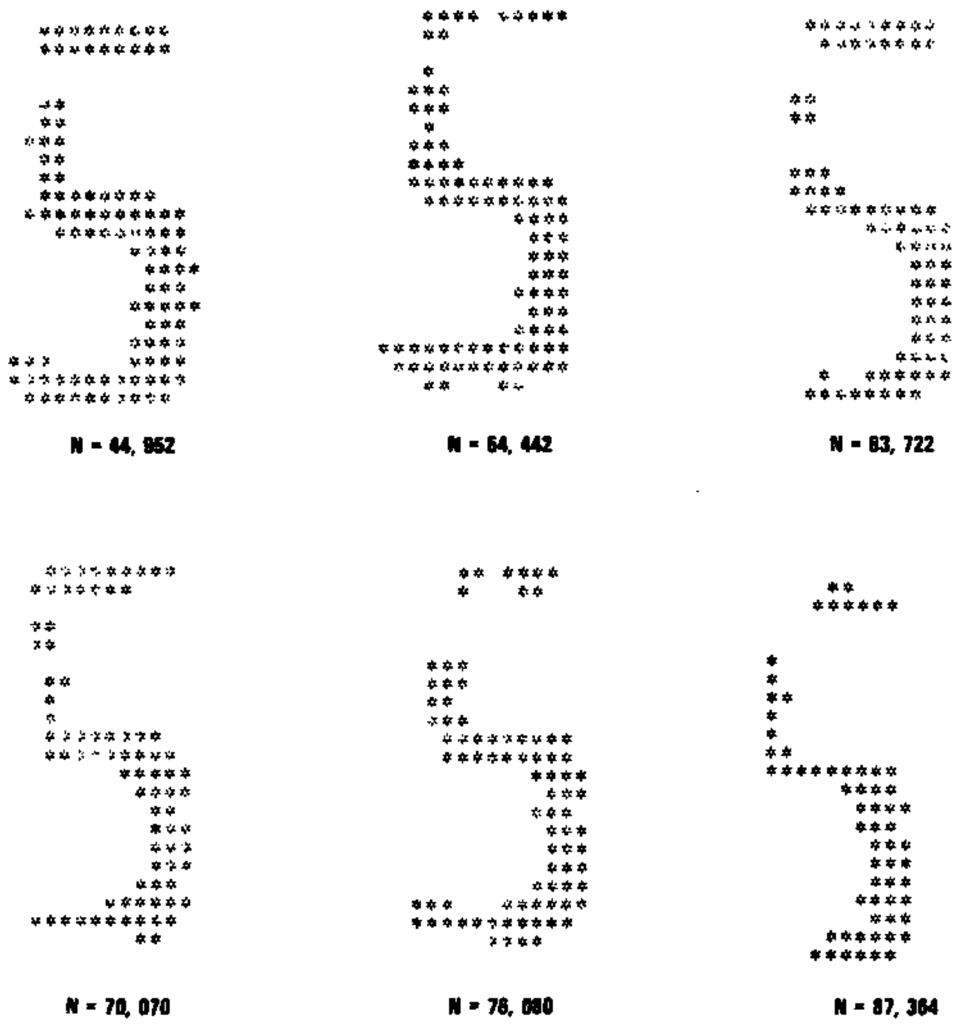

Figure 8 Examples of reject patterns