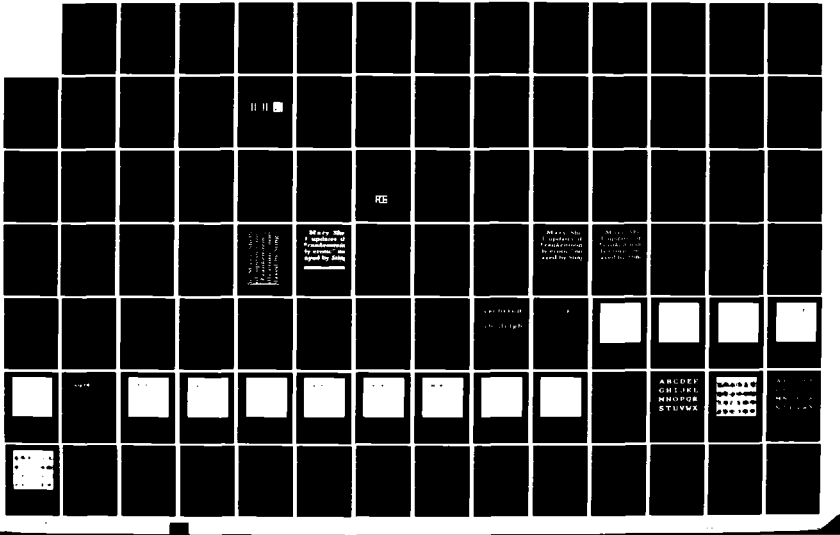
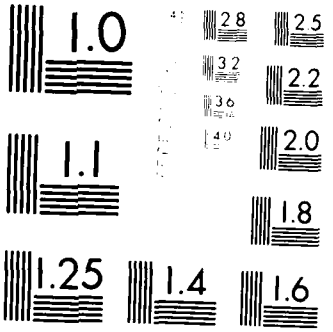
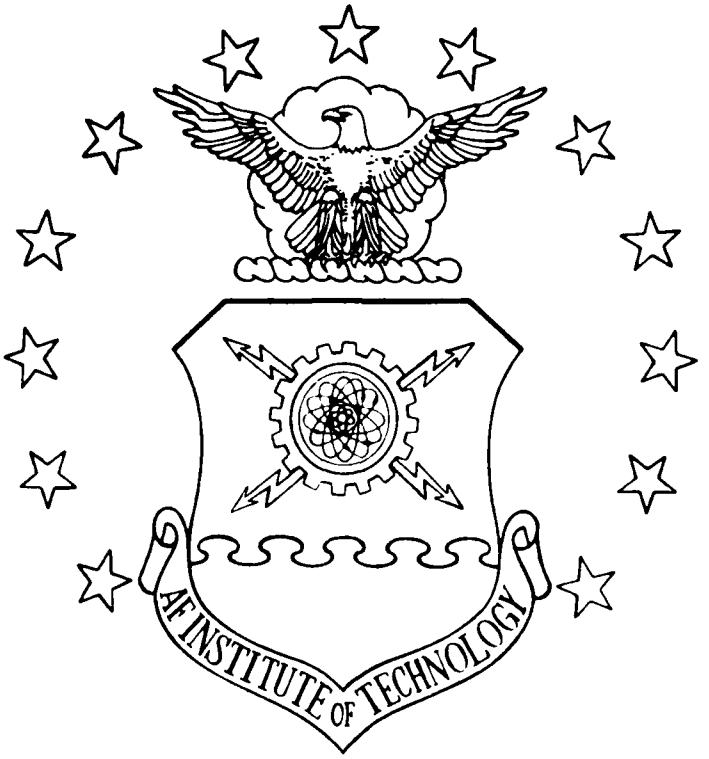
AD-A152 006 CHARACTER RECOGNITION USING CORRELATION TECHNIQUES(U) 1/2
AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL
OF ENGINEERING L SHUM DEC 84 AFIT/GE/ENG/84D-59

UNCLASSIFIED F/G 5/7 NL





MICROCOPY RESOLUTION TEST CHART



CHARACTER RECOGNITION USING CORRELATION TECHNIQUES

THESIS

Lugene Shum
Second Lieutenant, USAF

AFIT/GE/ENG/84D-59

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

CHARACTER RECOGNITION USING CORRELATION TECHNIQUES

THESIS

Lugene Shum
Second Lieutenant, USAF

AFIT/GE/ENG/84D-59

AFIT/GE/ENG/84D-59

CHARACTER RECOGNITION USING CORRELATION TECHNIQUES

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Electrical Engineering

Lugene Shum, B.E.

Second Lieutenant, USAF

December 1984

PREFACE

The following thesis is a presentation of the results obtained during research on the use of correlation techniques for automatic machine recognition of unformatted text. This thesis should be read as the seventh volume in a series of eight other works involving such research by former students at the Air Force Institute of Technology (AFIT). In sequential order (along with their author), the other theses are:

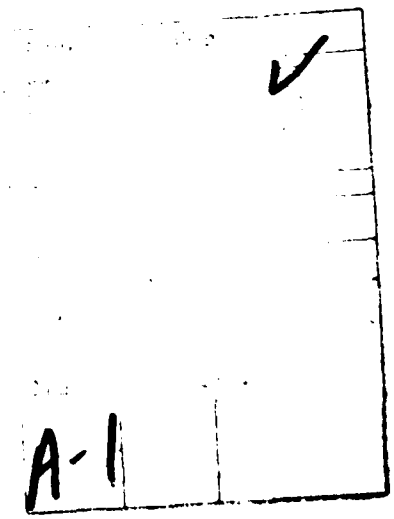
1. PATTERN RECOGNITION BY FOURIER SERIES TRANSFORMS
   C.H. Radoy
2. THE CLASSIFICATION OF VISUAL IMAGES BY SPATIAL FILTERING
   O. H. Tallman
3. FOURIER SPATIAL FREQUENCY COMPONENTS AS DESCRIPTORS FOR RECOGNITION OF VARIABLE FONT TYPESCRIPT
   P. W. Kabler
4. THE DESIGN OF AN OPTIMUM ALPHANUMERIC SYMBOL SET FOR COCKPIT DISPLAYS
   L. F. Bush
5. SEGMENTATION OF TOUCHING ENGLISH LETTERS
   R. E. Bentkowski
6. MACHINE SEGMENTATION OF UNFORMATTED CHARACTERS
   R. A. Simmons
8. WHOLE WORD RECOGNITION BASED ON LOW FREQUENCY FOURIER COMPLEX AND AMPLITUDE SPECTRUMS
   M. Ohair

These other theses provide additional understanding as to why certain parameters exist and why correlation was performed with Fourier Transforms.

During the course of this research, problems were encountered with the OCTEK 2000 and the Eclipse system. With the Octek, random pixel shifts occurred in the digitization of pictures. The source of this problem was still undetermined at the conclusion of this thesis. Problems with

the Eclipse, however, were known to be due to the TSS (Time Sharing System) operating system. Certain programs could not be run under this system.

In order to complete this thesis, I received much help from many people. I esspecially wish to thank Dr. Kabrisky, my thesis advisor, for his guidance. I would also like to thank Capt. King for sharing his expertize on digital signal processing techniques and Mr. Dan Zambon for his technical assistance on the Eclipse. Finally, I would like to thank Capt. Cross for his helpful suggestions.



Lugene Shum

CONTENTS

LIST OF FIGURES

FIGURES (CONT.)

ABSTRACT

This thesis presents an algorithm for recognition of letters of the English alphabet. The letters may be single, isolated letters; or concatenated, groups of letters. The algorithm essentially consists of a two pass correlation in which the size and energy of the unknown characters are set equal to a known template.

1. INTRODUCTION

Since the advent of high-speed computers, digital image processing has become a topic of great interest to both academia and industry. In particular, industry has focussed much attention on that of machine recognition of two- and three-dimensional images. Although much hardware and software have been developed during the last few years to support such efforts, modern computers still cannot locate and identify objects within a cluttered scene. This extends from locating and identifying letters on a page to t of tanks in the desert

This thesis addresses the former problem. Specifically, it examines the problem of locating and identifying printed English letters on a page of text.

1.1 GENERAL BACKGROUND

Human entry of data into digital computers is often slow and erratic. In most cases, it involves the conversion of printed symbols (useable by people) into electronic symbols (useable by computers). A secretary usually performs this task by reading the input data and then toggling a set of switches, or keys, attached to a keyboard to enter the data into the computer.

Since this method of data entry is totally under human control, the abilities of each individual operator greatly affect the performance and operation of the process. The

physical endurance and attention span of each person limits the number of hours of operation. (After several hours of work, a person usually becomes bored and tired.) In addition to human fatigue, the process is also affected by the skills of each individual operator. Thus, speed and accuracy varies accordingly with the competence of each person.

In light of the fact that machines can perform at greater speeds and for longer hours than a person, there exists a need for machine control of the entire data entry process. Automation would improve efficiency and reliability and reduce the total amount of manpower required for the task.

1.1A BACKGROUND ON DATA ENTRY

At present, most applications require that data (i.e. hand-written text and unformatted text) be entered manually by human operators into digital computers via keyboards. This requires many hours of tedious manual labor. An ideal process would be one where the operator merely presents the computer with some printed material and it automatically inputs the data. For ease, the printed material may be presented visually (pages from a document) or verbally (spoken into a microphone).

The development of such alternatives to manual keyboard entry of data include research for Optical Character Recognition (OCR) systems and Voice Recognition systems. For the most part, it appears as if replacement of the keyboard

II. THEORETICAL DEVELOPMENT

One of man's greatest desires has always been to create a machine which functions in human like manner. Even though he still does not fully understand the mechanics of the ubiquitous data processing operations (which seem trivial and commonplace to him) that he routinely performs, he will nevertheless attempt to replicate those processes in a mechanical device. He has thusfar produced mechanical devices which can record, reproduce, and transmit speech and pictures. Now, he must develop machines which can read, understand speech, and analyze images.

With vision as the subject, several researchers have spent much time and effort in developing models for the human visual system. These models may provide the design of future optical recognition machines.

2.1 INITIAL CONSIDERATIONS

In our attempt to build a reading machine, we are trying to imitate the human process of reading. Therefore, the machine process should follow that of the human process. However, since we still do not fully understand (or even have a complete theory on) how the human visual system processes the information it receives, it would be very difficult to design a recognition device which employs the same variables that a person uses to describe his environment.

1. DG Eclipse S/250
   (16-bit minicomputer with 256 kbytes of memmory and
   4 harddisk drives)

2. DG Nova 2
   (16-bit minicomputer with 64 kbytes of memmory)

3. Octek 2000 Image Analyzer

4. Dasher 6053 terminal (for the Eclipse)

5. Heathkit terminal (for the Nova)

6. Tektronix 4632 video hardcopy unit

7. Cohu vidicon camera
   (model 6150-000)

8. Electrohome display monitor

exists or can be easily written at a future time. Such programs will perform the following tasks:

a. determine the angle of a line and make adjustments for it,

b. determine the height of a character and adjust that of the template.

1.6 APPROACH

Research proceeded from the idea that machine reading should be based upon character recognition, i.e., identifying each letter on a page. It was also determined that recognition should be performed using correlation techniques.

In order to simplify the recognition task, it was decided that as much noise as possible be removed from the input text. Thus, the first part of this research examined various methods for the automatic reduction and elimination of noise.

1.7 MATERIAL AND EQUIPMENT

The Signal Processing Laboratory at the Air Force Institute of Technology (AFIT) provided all the necessary material and equipment for this research. Guidance was given by AFIT staff and faculty.

Most of the work was performed on a Data General (DG) computer system which consisted of the following :

uncorrelated, that is, each digitization of the same text will produce a random noise pattern. If the noise pattern is constant, then a simple filter can mask it out.

3. Random noise will not alter a character's shape, i.e., stray marks will not change one letter into another. Usually, stray marks are much lighter in color than that of the letter and can be detected as noise. For example, stray pencil marks will not change a "c" into an "o". However, if such marks are darker than the letter, it will be recognized as an "o".

4. The input text is properly lighted, such that, digitization of the text will produce a bi-modal distribution of greylevels. Much of the background will then be centered around white, and much of the print will be centered around black.

5. It is assumed that a mechanical device exists which can automatically control the focussing of the digitizing camera. Such devices are available commercially.

6. The lines of printed text are placed such that each line is horizontal and not at an angle. Thus, it is assumed that the text is properly placed for digitization and not upside down.

7. The final assumption is that certain software

which implemented most of the functions.

Due to time constraints, this thesis could not examine every aspect of the recognition problem associated with every character shape. This limited the study only to typed letters of the English alphabet. This excluded stylized print (Script, Gothic, etc.), numerics (1, 23, 58, etc.), and special characters (*, &, %, #, etc.). In particular, this thesis did not deal with hand written text or print which has been rotated. For most of the testing, a book was randomly taken out of the library and pages from it were digitized.

The only constraint set upon the shape of the letters was that each character must be recognizable by a person without reference to the word itself. Thus, recognition of a letter must be based solely on the shape of the letter and not on what it should be in a specified word.

1.5 ASSUMPTIONS

Seven assumptions were made at the start of this research in order to simplify the problem:

1. The average width of a character is a given parameter. For the moment, this is a valid assumption because it allows us to search for a solution. On any page of text, there is a high probability that one can find at least one one-lettered word. If one is found, then its width will be used.
2. The noise produced by the digitizer is

classified as spatial noise. Noise incurred during the digitization process include video snow and nonlinear image distortions. Such noise are classified as temporal noise. Noise from either process tends to alter enough of the shape of the letters to cause errors in recognition.

Other problems for reading machines come from varying font styles. All text do not come from the same printing press. Even within the same book one can find multiple font styles. (For instance,the title of a book and its text are usually printed in different styles).

In addition to varying the character shape, modern electronic printing with its proportional spacing causes the virtual concatenation of letters within words. This now creates the problem of finding or separating the letter to be recognized ("segmentation" problem).

This thesis examines those problems associated with the proccess of recognizing letters. Specifically, it examines

1. noise from the digitization and printing process, and
2. character recognition based upon correlation techniques.

1.4 SCOPE

The goal of this research was to start development of an algorithm which would give computers the ability to recognize the letters from any page of text without the intervention of a human operator. The primary output consisted of the algorithm and a series of computer programs

1.3 STATEMENT OF PROBLEM

To date, reading machines based upon Curve Tracing and Template Matching techniques do not work well unless the input text is very precisely controlled in shape and location. The problem lies in the fact that such techniques fail when noise is encountered in the print which causes the input font style (type style) to vary from that of the standard set. In addition, current techniques do not allow for the identification ofletters which do not come from the standard set.

The performance of current reading machines is optimum only when the print on the input text is "perfect". "Perfect" means the following conditions:

1. the ink is of a uniform black color,
2. the paper is of a uniform white color,
3. the letters are all from the same type style,
4. the letters do not touch each other, and
5. the letters are all well formed.

Under these conditions, a reading machine will not have any difficulty in recognizing the letters on a page of printed text.

Most text, however, does not come in perfect condition. One source of problems is that of noise incurred during the printing process and the digitization process. Noise incurred during the printing process include splotches of ink, stray marks, and breaks in the letters. Such noise are

1.2D TECHNOLOGICAL LIMITATIONS

Using today's more advanced technology, computer vision can still at best be compared to that of a blind man probing around his environment with a cane. With respect to reading unformatted text, the scene becomes that of the blind man probing around a page of raised letters with a pin. Both the man and the machine cannot examine an entire page at once to locate the print. They must search around the page until a line of print is found. Then, they must find the beginning of the line and determine the first letter of the first word.

Restricted to using only a pin, finding and then recognizing letters becomes a very difficult task for the man. (Allowing him to use his finger, or hand, would only mean programming the computer with more pointers and performing more computations.) Although it is a difficult task, it is not an impossible task for the man. Thus, building a machine that reads should also not be an impossible task.

Today's technology can provide all the computational tools required to perform the necessary calculations. In conjunction with existing mass storage systems and high resolution digitizers, all the hardware needed to build a reading machine exists. However, complete software required to run the hardware does not exist.

The Kurweil Computer Products company, producers of the Kurweil Data Entry Machine, keeps the designs for their machine a closely guarded secret. The techniques it employs are not made known to the public.

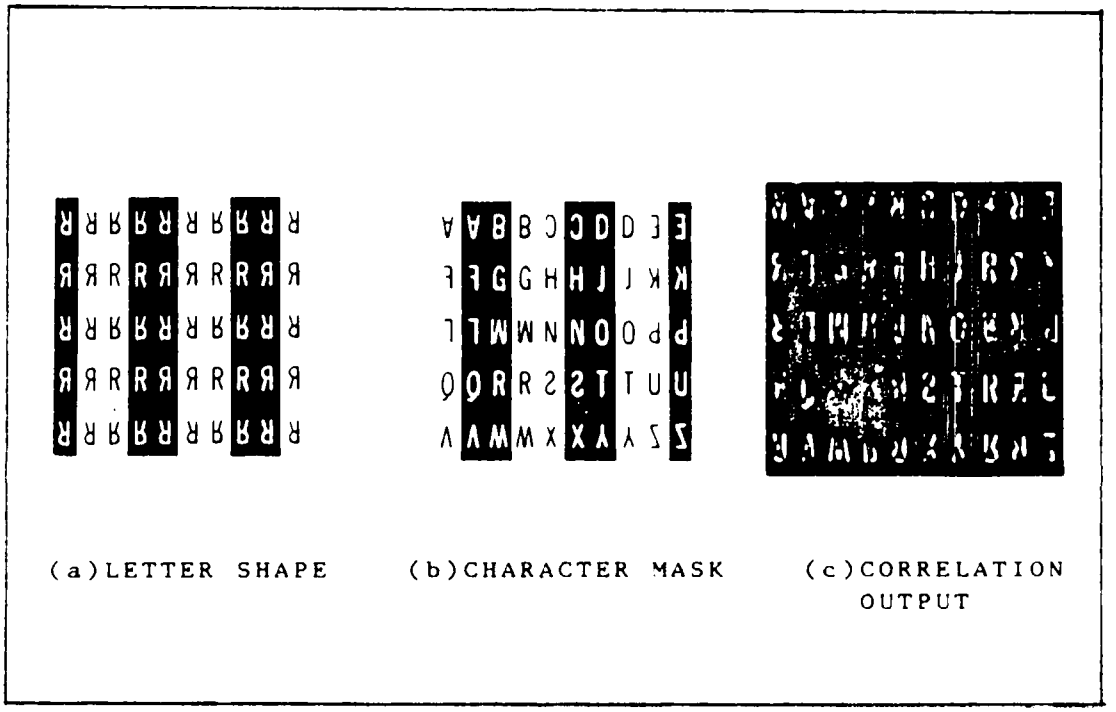
Although they are sold commercially, most machines do not perform as well as expected. Highly varying font styles and nonuniform printing cause intolerable error rates. A human operator is then required to correct the mistakes. Thus, instead of operating at machine speeds, such machines operate at rates slightly higher than that of a secretary.

1.2C AFIT OCR

At the Air Force Institute of Technology (AFIT), one technique for OCR employs a Fourier Transform correlation scheme for its recognition process. Developed by R. Simmons in 1981, the algorithm reconstructed the the magnitude of the Fourier Transform of a template character with the phase of the Fourier Transform of a string of characters (which contains the template character). Shapes which did not have the magnitude of the template were attenuated in the visual domain. Thus, a peak detector could be used to locate the most probable position of a character.

This algorithm seemed to work quite well with "large" letters, but failed when presented with letters that were "smaller" (in comparison with the Fourier window size).

in a given set (see Figure 1.1). Letter shapes which do not match that of the unknown character allow various amounts of light to pass. A perfect match would totally block all light from passing through. Recognition is then based upon the detection of a minimum amount of light.



(a)LETTER SHAPE (b)CHARACTER MASK (c)CORRELATION OUTPUT

Figure 1.1 RCA CORRELATION TECHNIQUE

The IBM machine uses a curve tracing technique in its recognition scheme. A pointer spirals around the boundary of a letter and generates a table of the shape features. These generated feature vectors are then compared against a standard set for recognition.

1.2A APPLICATIONS FOR READING MACHINES

As stated in the previous sections, the primary use of a reading machine is for automatic entry of data into computers. This includes use for text storage and information processing. One specific area of application in information processing involves automatic computer translation of foreign text. As a front-end processor, a reading machine can automatically identify and input foreign text into a computer which can then translate it.

A very well known application area for reading machines involves the Post Office and sorting mail. Such a device can identify the address on a letter and classify each with respect to their zip code.

In addition to easing the data entry task, reading machines can also be used as aids to those visually impaired. Connected to a voice synthesizer, such machines can literally read to persons who cannot see.

1.2B OPERATIONAL READING MACHINES

At the present, there are several reading machines available commercially. They include the RCA Multi-Font Reading Machine, the IBM Reading Machine,, and the Kurzweil Data Entry Machine.

The RCA machine uses an optical correlation technique in its recognition scheme. A "picture" of an unknown character is projected onto a transparency containing negative and positve replicas of all the possible characters

Using Curve Tracing techniques, the template set contains a list of the geometric shapes of each letter. These shapes usually consist of curves, angles, and straight lines. Each letter is then uniquely defined by a list of the given shapes. The features of an input character are obtained by following along the outer edge, or boundary, of the character. Based upon certain rules, a determination is made as to what shapes the input character contains. Recognition is then based upon comparison of the determined shape features with those of the template set.

Using Template Matching techniques, the template set consists of the letters themselves. Each letter thus uniquely defines itself. Instead of identifying the various geometric shapes, the entire input letter is matched against another letter from the template set. Recognition is then based upon which template letter the input character best matches.

1.2 CURRENT KNOWLEDGE

At the present, general reading machines are still in the early development stage. Those which are commercially available now are crude working models which provide limited performance for use in restricted application areas. Funds for additional research come mainly from those interested in increasing office automation.

3. classify the objects within the visual image (e.g. determine if the object is a tank).

Specifically for reading machines, a video digitizer converts the pages of input text into digital images. A computer or microprocessor then analyzes each digital image for characters.

It is important to note that the digital image, or digitized picture, is a bit map of the light intensities reflected from the page. Therefore, the electronic image is only a "photograph" of the text. It contains only greylevel information and does not contain ASCII values for the letters on the page. A computer or microprocessor is required to process the greylevel image to find and identify the letters. Usually, features extracted from the electronic images are compared against some set of templates and thresholded to identify the letter.

1.1C BACKGROUND ON PROCESSING TECHNIQUES

Traditional methods for processing characters have included Curve Tracing and Template Matching techniques. With each, a prototype (or token) is generated with some specific attributes of a given letter. Prototypes for each of the given letters then form a template set (or standard set). Accordingly, each input image is compared against those in the template set and categorized as being similar to one or more prototypes; else sufficiently dissimilar so as to be rejected as an allowed character.

will come from OCR technology in the form of a reading machine.

One major disadvantage of a voice-operated input device is that it, too, operates under human control. (The operator has only exchanged the use of one set of muscles for those of another.) Thus, like keyboard entry, such a system would be susceptible to the physical limitations of the individual operator. However, for reading machines, control is under machine guidance and operation may proceed at machine speeds with limitations due only to mechanical performance.

1.1B BACKGROUND ON READING MACHINES

In this thesis, a reading machine is defined as any optical/electrical/mechanical device designed to recognize characters from a given set of letters for the purpose of automatic input of data into digital computers. (See setion 1.2A for other applications.) Currently, most reading machines are optical systems based upon Optical Character Recognition technology.

Since reading machines deal with identifying letters, they are classified as image processing devices. The basic operation of all such devices consists of the following steps:

1. convert the visual image into a digital image (e.g. a greyscale version of the visual image)
2. store and pre-process the electronic image (e.g. edge-enhance the image)

We do not know how people read; but, we do know that people can read. The human reading process involves input through the visual system and processing in the brain. We must therefore consider the operations of these systems and use them as guides in building a reading machine.

2.1A GUIDELINES FOR RECOGNITION

If the human recognition system exemplifies the best recognition system, then our mechanized model should operate on a parallel level.

Under certain circumstances, our human recognition system fails and produces erroneous outputs. Under other conditions, our human system produces the proper results. Thus, given the same conditions, a mechanical device should see and interpret things only as we see them.

Remembering that the reading machine is a mechanical device, we should expect it to outperform the biological machine in certain areas. This comes about from the fact that many of the parameters in a mechanical device are easily controlled. Hence, problems which may appear difficult, or even impossible to the human system should present no problem to the mechanical one. (For example, a person may distinguish between only approximately 100 greylevels, but a machine may be programmed to distinguish more.)

2.1B STANDARDS FOR OPERATION

If a human operator can isolate and identify any letter on a page of text, then a reading machine must also be able to perform the same task. In addition, the machine must concur with the letter that the human operator recognized and do so consistently.

Since the mechanical device is to replace a human operator, then the machine must perform more efficiently than a person. It must have a higher rate of operation and a lower error rate.

2.2 READING

We do not know exactly what sequence of mental activities occurs within a person during the reading process that gives him the ability to recognize letters of varying sizes and styles. (A mere child of five may be taught to recognize the 26 letters of the English alphabet, yet, a computer cannot perform the same task.) In mechanizing the reading process, we must consider the method of recognition and its implementation.

2.2A WORDS OR LETTERS

Research in developing reading machines (at AFIT) has recently taken on two approaches. One view holds the position of building reading machines based upon word recognition, i.e., locate the words on a page and try to recognize them. The other and more traditional approach

bases recognition upon the letters in a word.

The first approach is much simpler in that one does not have to deal with finding characters in a concatenated sequence of letters. In most text, the words are fairly separated from each other; while the letters are spaced fairly close to each other. The biological reading process seems to tolerate closely spaced words as long as they do not physically touch each other. Letters, on the other hand, may touch, but not overlap more than about 10% of their width (2).

At the present, we cannot tell which approach will lead to the building of the first successful reading machine. In this thesis, we will take the latter approach. Advanced reading skills may perhaps take place at the word level, but, a human reader can always recognize the individual letters on a page of text. (For this reason alone, research was devoted to character recognition.)

2.2B BUILDING A TEMPLATE SET

No two objects in this physical universe are ever exactly alike. Thus, the human recognition system acts accordingly to generate a set of standards or references for interpreting and classifying those inputs from its visual e ironment.

From the fact that, at any instant of time, the human system can intelligently ignore all extraneous details which are unimportant to the recognition process, some researchers

have suggested that the biological recognition system generates a rather loose set of standards which is ill-defined and subject to refinement and adjustment. The exact process whereby it generates and modifies those standards is still unknown to man.

From a totally different point of view, some believe that every visual experience is stored somehow in toto. Thus, the human recognition system compares images from an external scene with those of previous impressions or experiences. There are still others who believe that the answer lies somewhere in-between those two extremes. In any case, we still do not have the total answer.

2.2C STORAGE BY FOURIER TRANSFORMS

Based upon biological and physiological information about the human visual system, Kabrisky (1) has developed a theoretical model for the human recognition system. In this model, all "remembered" patterns (experiences) are stored away topologically (unaltered under a homeomorphism). The system then performs its recognition by correlating, on a point-by-point basis, a given input pattern with all the stored patterns and selecting the pattern which correlates the best.

As pointed out by Kabrisky, the amount of memory required by this model to store all of the patterns which an average person can remember would seem to be prohibitively large. One solution to this memory problem would involve

pattern storage by means of Fourier transforms.

Using Fourier transforms, the human recognition system would only have to evaluate and store part of the transforms of a visual pattern which it is to remember. Recognition will then come from comparing an input with the inverse transform of the stored patterns. From research, selective spatial filtering of the Fourier images permits accurate classification of visual images.

2.2D OPTICAL VS DIGITAL PROCESSING

We know that the human reading process involves input through the visual system and processing in the brain. If we divide this process into two totally separate functions, then we may implement it using only a video camera and a digital processor.

Kabrisky's model assumes that the visual system can perform a two-dimensional Fourier transform calculation. If we follow this method, then reading might well be performed optically and not digitally. For this research, the digital system was chosen because it was readily available.

2.3 DESIGNING THE SOLUTION

As mentioned in the above sections, since a reading machine is a mechanical device, then certain parameters are easily controlled. For example, whereas a person must read the print as is on a document, a digital computer can modify the color (greyscale) of the background and print. In

addition, the size of print can be quickly enlarged or reduced. With the proper manipulations, input data can be transformed into a form more amenable to the recognition scheme.

2.3A NOISE REDUCTION

Before we begin to read a book, we usually turn on a light. This allows us to see the printed text. In much the same way, we must first ensure that the reading machine can "see" the text. Proper illumination and placement of the text for digitization provides one means of this.

Once digitized, the greyscale distribution can be modified for maximum contrast between print and background. Assuming a bimodal distribution of color, a greyscale image can be thresholded to form a binary (black on white) image. Binary images are more tractable for computer processing.

2.3B LOCATING THE LETTERS

After an electronic image has been formed, the computer must read the print on the page. Following the human process, the computer must start at the beginning of the text; reading the first word of the first line. Here, instead of recognizing the entire word, the process is brought to a lower level of recognizing the letters in the first word.

In order to read the letter, the machine must first find the location of the words. This requires finding the

spaces between each word. This is a simple task relative to finding the letters on a page. If the letters within a word touch each other, then there is no space between each letter.

If instead of presenting the recognition scheme with a single letter, we present it with a group of concatenated letters, then the process must recursively identify the letters in the string. This is to say that the recognition scheme must make several passes through the string and judge which characters fit the best. As an example, consider the string of letters presented in Figure 2.1. Restricted to choosing only letters, the string could have several interpretations. It could be FOG, AOG, OR ACG. A recursive correlation routine readily lends itself to this identification method.

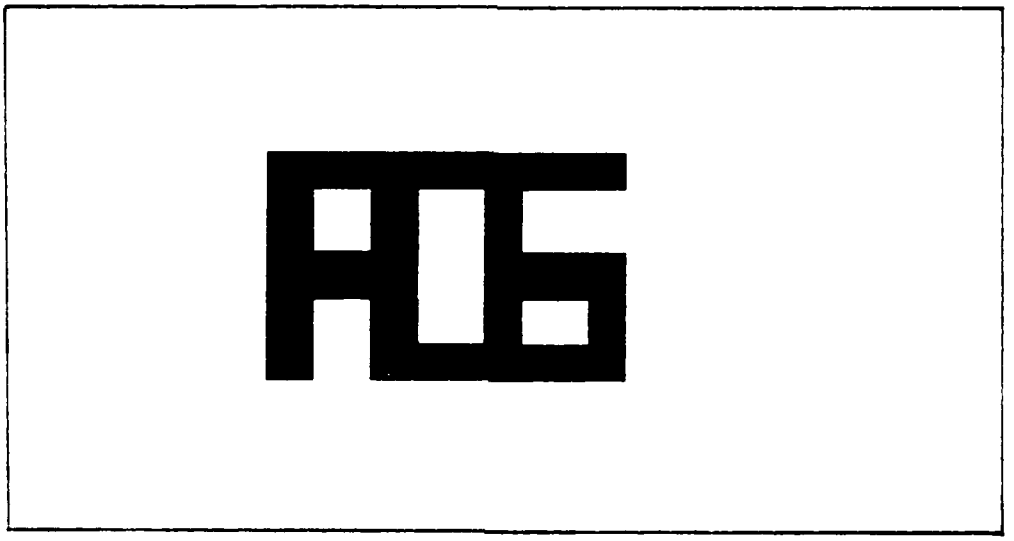


Figure 2.1 THREE CONCATENATED LETTERS.

2.3C RECOGNITION OF LETTERS

Based upon Kabrisky's model, recognition should come from correlating the Fourier transform of an input character with those of a template set. Also, such correlations should take place in the Fourier domain with spatial filtering.

If the human system uses an loosely defined template set for recognition, then the mechanical model should be able to modify its templates. Futhermore, since correlation is similar to template matching, it would be desirable to have both the template shape and input shape as similar as possible. This implies adjustment of the height and width of each template letter. The template height can be set to the input height by measuring that of the input shape. Adjustment of character width means adjusting the thickness of each character.

Adjusting the character width will help correlation in that this will affect the low order harmonics of a Fourier Transform. (A thick character will have more energy content in the low order harmonics than a thinner version of the same letter.) Either the template letters can be thickened to match that of the input, or, the input can be thinned to match that of the template. The reasons for choosing a thinning algorithm is two-fold:

1. Thinning offers a more controllable operation. It is easier to thin a letter to a known pixel size than to

thicken a letter to a varying pixel size. (Some letters are not of the same uniform thickness in certain fonts.)

2. It is safer to modify the input character thickness than vice versa. With a template thicker than any input letter, the possibility exists of correlation with nearly everything in an input scene. (See Chapter III.)

With the height and width adjusted as much as possible, the remaining parameter to adjust is the energy. If an input character and a template contain the same energy, then correlation will produce a maximum peak only if the two figures truely have the same shape. Other shapes which may be similar will produce lower peaks.

III. SOFTWARE DEVELOPMENT

The following chapter formally presents the recognition algorithm examined in this thesis research. Although a total algorithm was described in Chapter 2, only a portion of those functions were fully implemented in software. Rather than devote much of the time to coding, it was decided that time was better spent on testing the validity of the recognition algorithm.

This chapter is divided into two sections. The first section introduces the overall processing procedure of the character recognition algorithm implemented in this thesis. The second section briefly expands upon the steps of the processing procedure and outlines the actual operational process of the recognition scheme. (Chapters 4 and 5 describe in more detail the operation of the software implementation.)

3.1 OVERALL PROCESSING PROCEDURE

The entire processing procedure of the recognition algorithm can essentially be broken down into three stages: the digitization stage, the pre-processing stage, and the recognition stage. Each stage performs its operations independently of the others with the output of one stage feeding directly into that of the next one (see Figure 3.1).

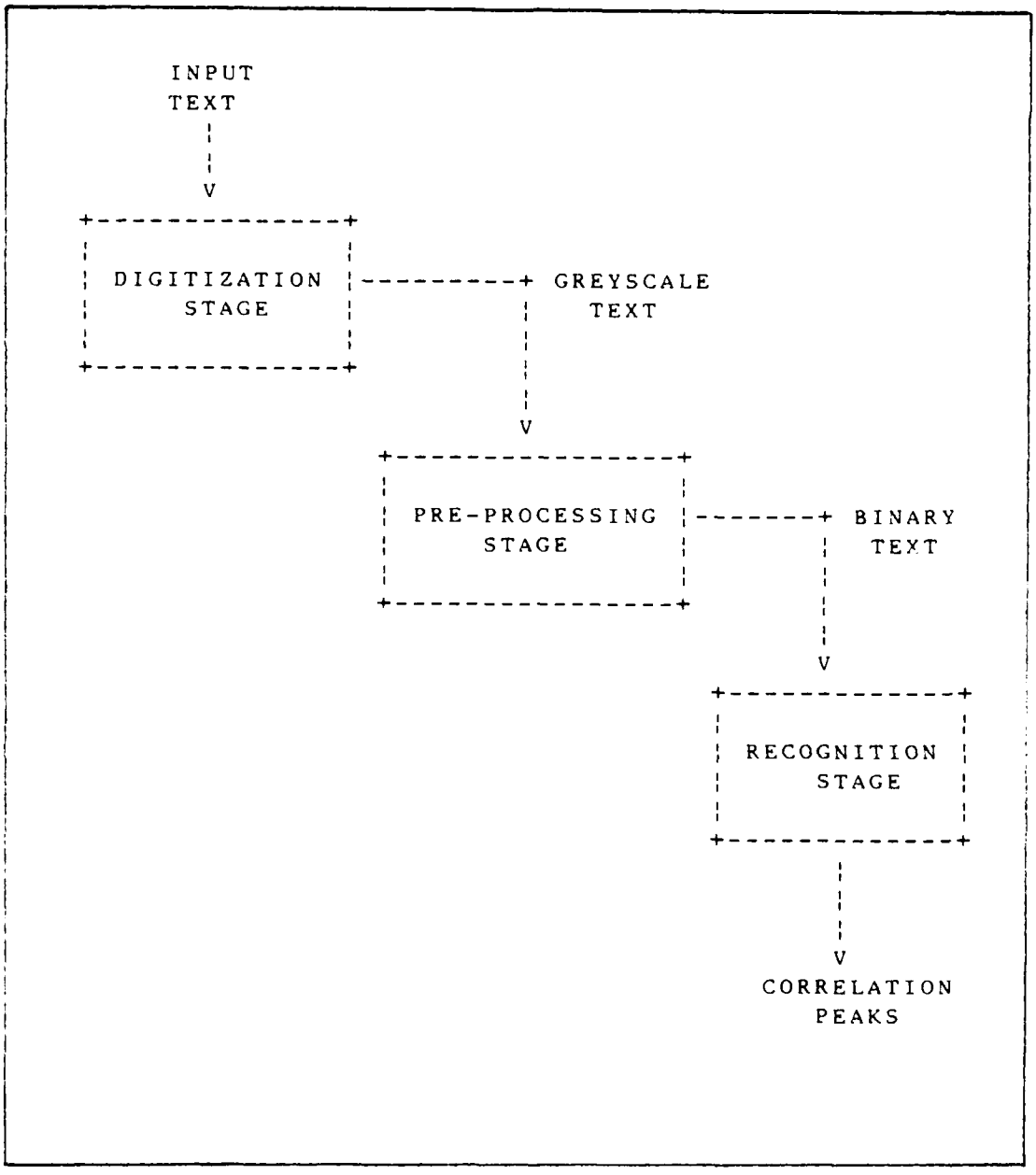


FIG. 3.1 OVERALL PROCESSING PROCEDURE

The Digitization stage performs the actual conversion of analog video text into digitized greyscale text. Most of this stage consisted of hardware which had already been established by former thesis students in the Digital Signal Processing Lab at AFIT.

The Pre-Processing stage takes digitized greyscale text and thresholds the greylevels to produce binary text. In addition to this, it determines the locations of the lines of print on the text and the position of the words within the lines.

The Recognition stage recursively correlates input characters with template characters in order to identify the letters. At each iteration, the characters are energy normalized to produce proper correlation peaks.

The reason for this specific division of stages was twofold. As already mentioned, the first stage was already implemented. There was no need to duplicate the work of others. Also, with this arrangement, time could be appropriately proportioned to each of the other stages. If the Pre-Processing stage was not fully implemented by the alloted time, then its completion would not affect that of the Recognition stage.

3.2 THE RECOGNITION PROCESS

The recognition algorithm examined in this thesis consisted basically of correlating an unknown word with a template character. (It should be pointed out here that in

this section, and those that follow, the term "word" refers to the output of the Pre-Processing stage. A word can be either an individual character or a string of concatenated letters. For an explanation of this, see section 4.2.) The energy of the template which produced a high correlation peak and the energy in the area of the word where the peak occurred would then be set equal to one another. Re-correlation of the two energy equalized images would then produce yet another series of correlation peaks. From the discussion in Chapter 2, the maximum peak produced by the second correlation pass would indicate the most probable identity of the word. (Maximum in the sense that if two objects have the same shape, then normalizing their energies would produce a correlation peak of one.)

More explicitly, the correlation process consisted of the following steps:

1. Individually correlate the characters of a template set with an unknown word.

2. Determine the location where each template correlated well.

3. Center a window around the correlation peak area and set the energy equal to that of a template.

4. Re-correlate the energy modified word with the template.

5. Repeat the energy modification and re-correlation process for each of the templates determined in the first pass.

6. Compare the correlation peak values of the second pass and choose the highest as the best answer.

IV. THE DIGITIZATION STAGE

This chapter discusses the digitization stage. As mentioned before, this level of the algorithm had already been implemented in hardware by other students. Many of the programs for running the digitization hardware were readily available. Instead of describing the details of those written programs, this chapter highlights those features of the digitization stage which are important to the understanding of the software for the next two stages.

4.1 ESTABLISHED WORK

Most of the original software for this stage was written by Robin E. Simmons. As a thesis student, he set the foundation and established the useful software systems for future work in the Digital Signal Processing Lab. The programs he wrote included:

1. UNPACK.FR
2. REPACK.FR
3. VTOC.FR
4. ITOC.FR

These were system dependent Fortran subroutines which allowed for the manipulation of picture elements (pixels) within a digitized picture. (See section below for explanation.)

Since the software already existed, this stage was used as a black box which converted analog video pictures into

The fourth step (FWORD.FR) extracts one word from the line of printed text. The word chosen is up to the user. It copies this word onto a blank background for the next step.

The fifth and final step (FCHAR.FR) further divides the word if it is possible. The third step only examines vertical blank lines between words. If an angular line exists, then it would be missed. The fifth stage accounts for that. It performs an operation much like that of finding a path out of a maze. It continues down a column of pixels until it can no longer do so. From that point it searches for a path to the left and right of the point. This operation continues until the bottom of the line of print is reached.

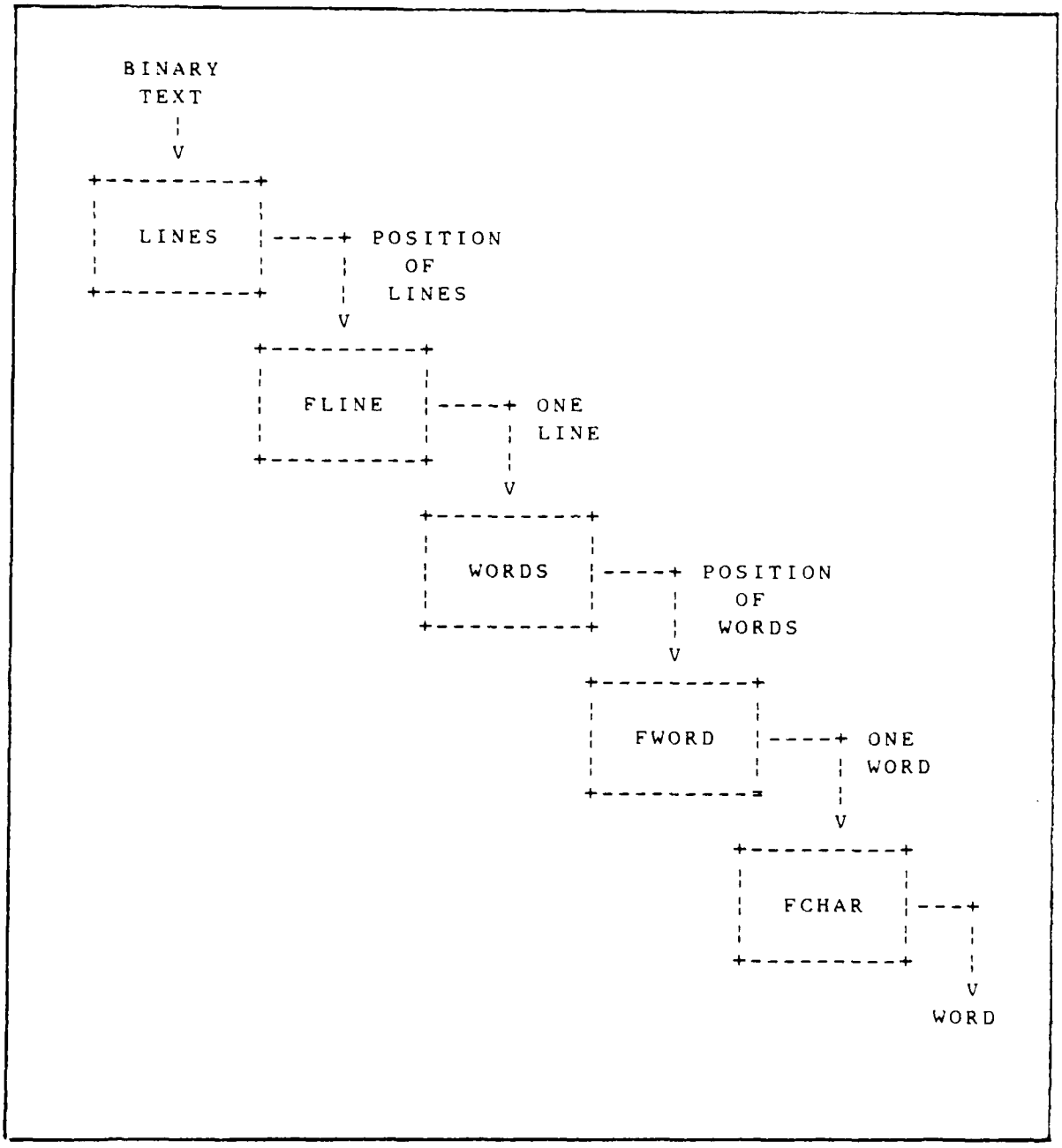


Figure 5.7 EXTRACTION OF WORDS

5.3 LINE/WORD LOCATING ALGORITHM

The input to the LINE/WORD locating process is the noise reduced text. From this, the algorithm determines the locations of the lines of print and proceeds to extract the words from the lines. (In this section and those which follow, the term "word" refers to the output of the LINE/WORD locating process. Thus, a word may be an individual character or a string of concatenated letters.)

This stage performs its operations in five steps (see Figure 5.7). The first step (LINES.FR) determines the locations of the lines of print. It does so by calculating a horizontal histogram of the text. Assuming that the text is properly placed, there should exist a series of blank vertical lines between each line of print. These blank lines would demarcate the location of the lines of print.

The second step (FLINE.FR) extracts one line of print from the text. The choice of which line is up to the user. It copies the specified line onto a blank background for the next step.

The third step (WORDS.FR) calculates a vertical histogram of the chosen line. If a straight vertical line can be drawn between each character, then that character is counted as one word. As mentioned above, if the letters are concatenated, then the string of letters is considered to be one word.

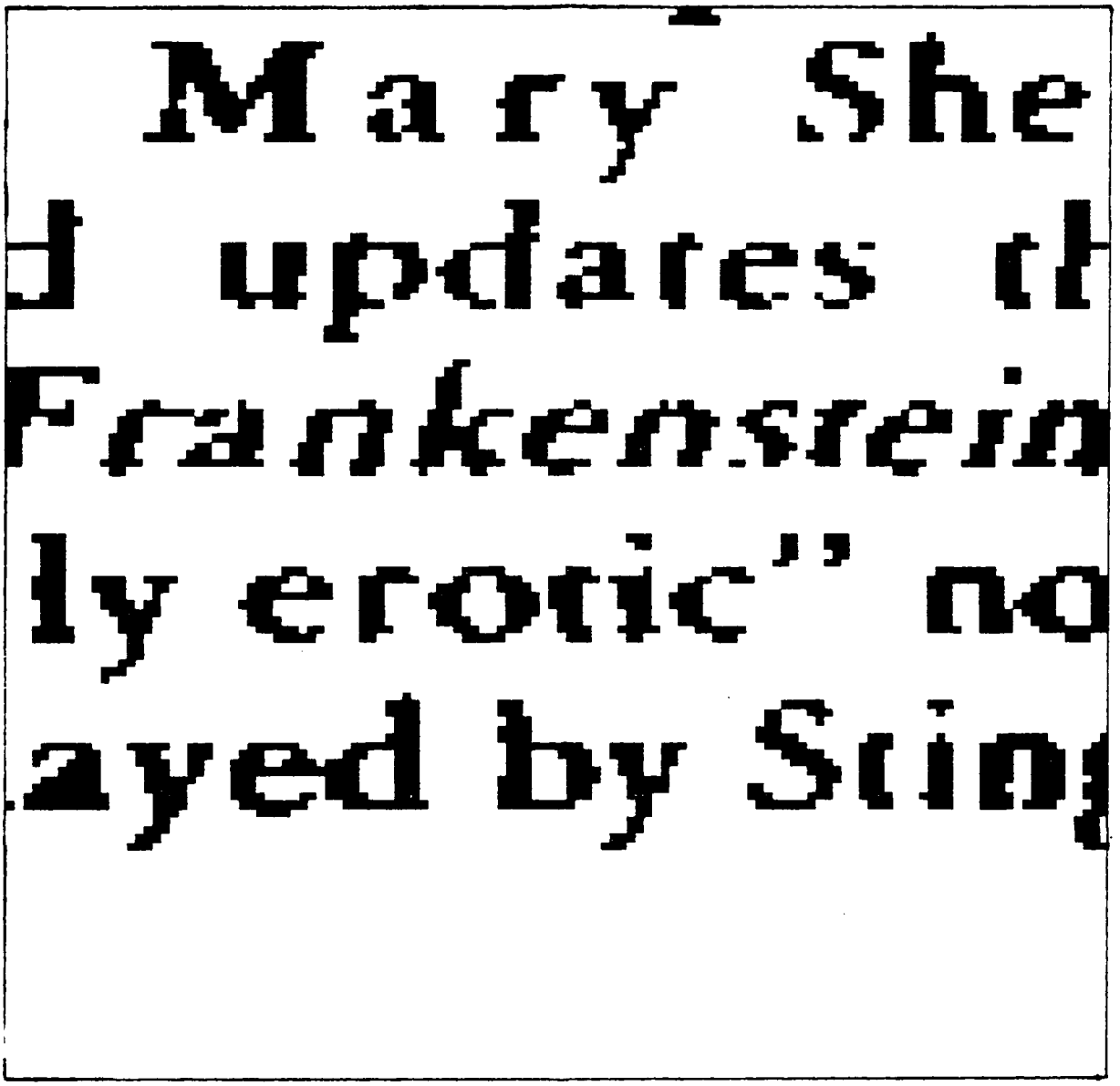


FIGURE 5.6 SPOTTED VERSION OF 5.5

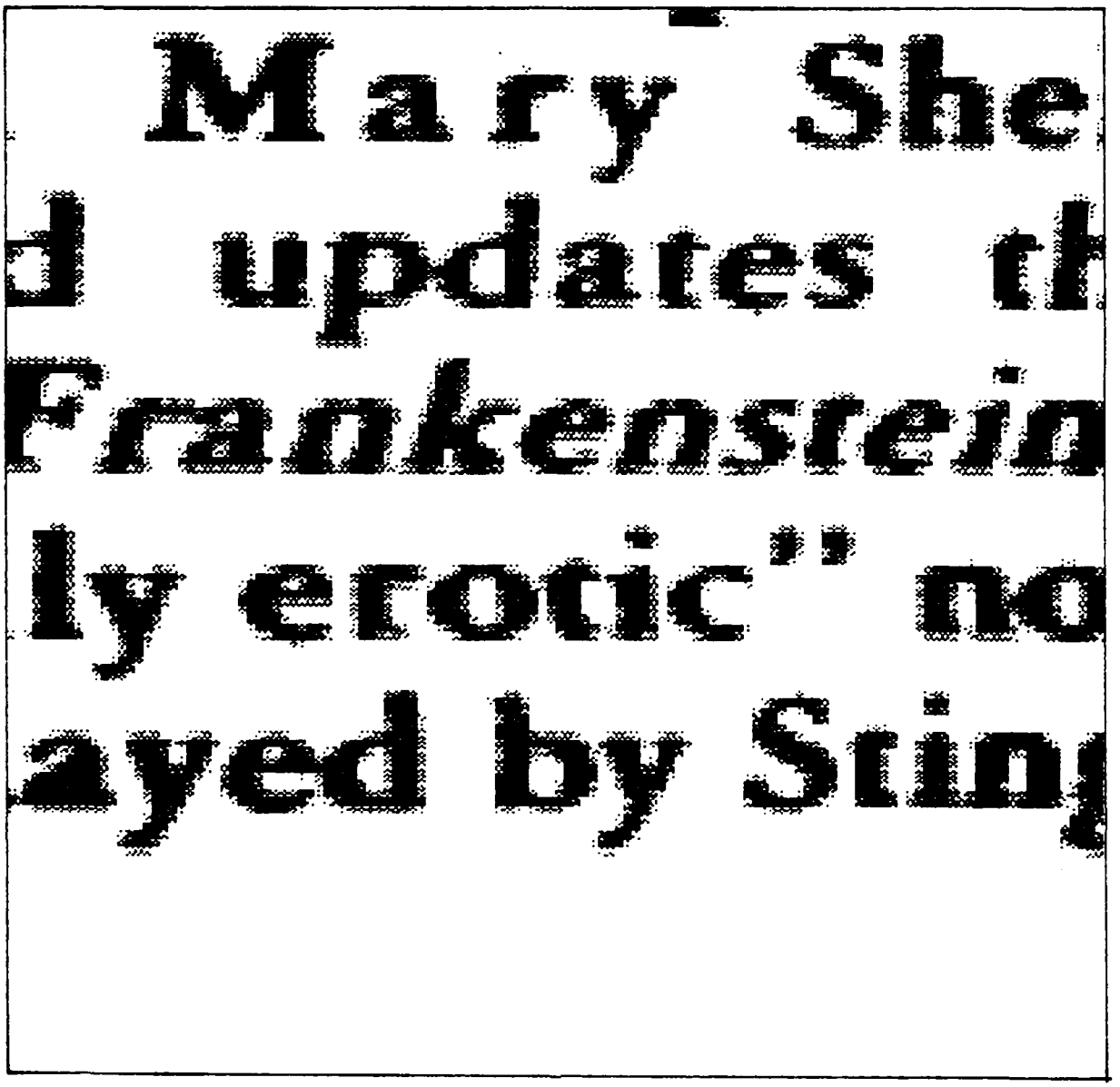


FIGURE 5.5 CLEANED VERSION OF 5.2

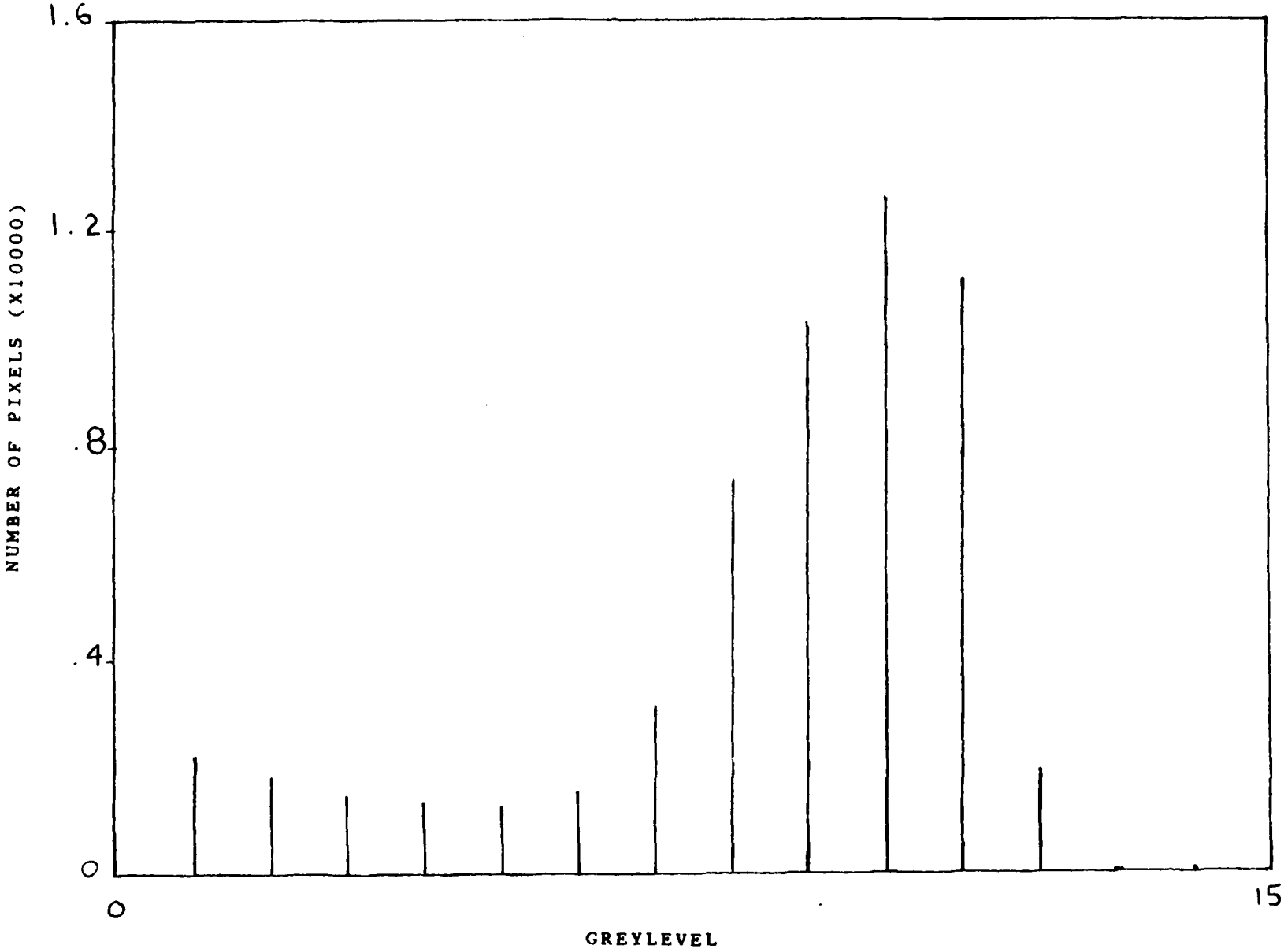


Figure 5.4 HISTOGRAM OF FIGURE 5.2

CLEAN takes a 16 level greyscale image and threshold-averages it to a binary image. It performs this operation under the assumption that the greylevels have a bimodal distribution. This routine first calculates a histogram of the greylevels. It then determines the two peaks corresponding to character color and background color (see Figure 5.4). From the peaks, it determines the first valley after the first peak. It uses this valley point to begin thresholding of the greyscale image. It thresholds up to the tenth level (the start of white on the scale). After each threshold level, the routine averages the clipped values to produce a threshold-averaged value. This final value is then thresholded aginst the valley point. Those below the valley point are set to 0, and those above the point are set to 15. Figure 5.5 shows the CLEANed version of Figure 5.2 .

SPOTS takes the CLEANed image and removes the spikes on the edge of the characters and fills in the potholes caused by thresholding. This routine takes each pixel and calculates a value for its 9 nearest neighbors (8 surrounding the pixel + 1 for the pixel itself). A one corresponds to a pixel without any neighbors and is therefore deleted. Values of zero correspond to holes and are filled in. Values of 2 and 4 are spikes and tips of spikes. They are therefore deleted. Figure 5.6 shows the SPOTed version of Figure 5.5 .

The noise reduction algorithm developed to binarize the text does so in two passes (see Figure 5.3). The first pass (CLEAN.FR) performs the actual conversion of 16 greylevels into two binary levels. The second pass (SPOTS.FR) refines the binarized text image.

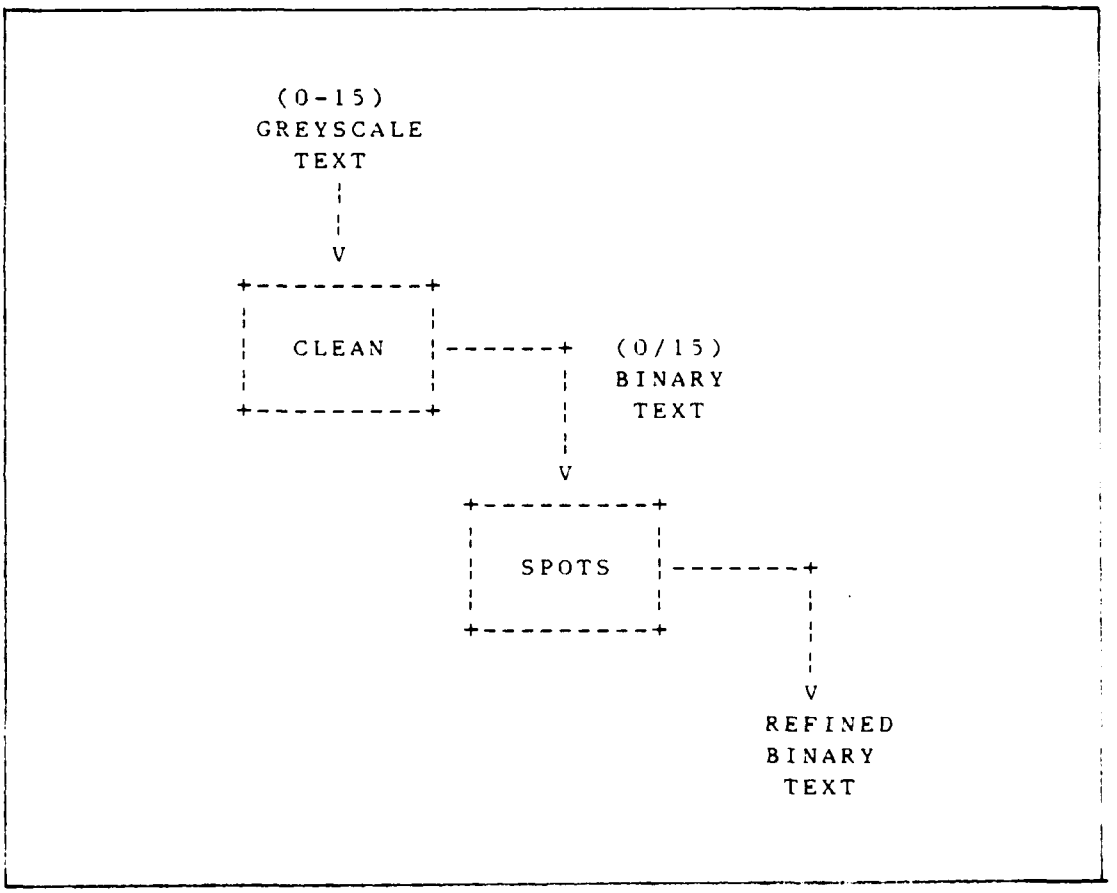


Figure 5.3 NOISE REDUCTION OPERATIONS

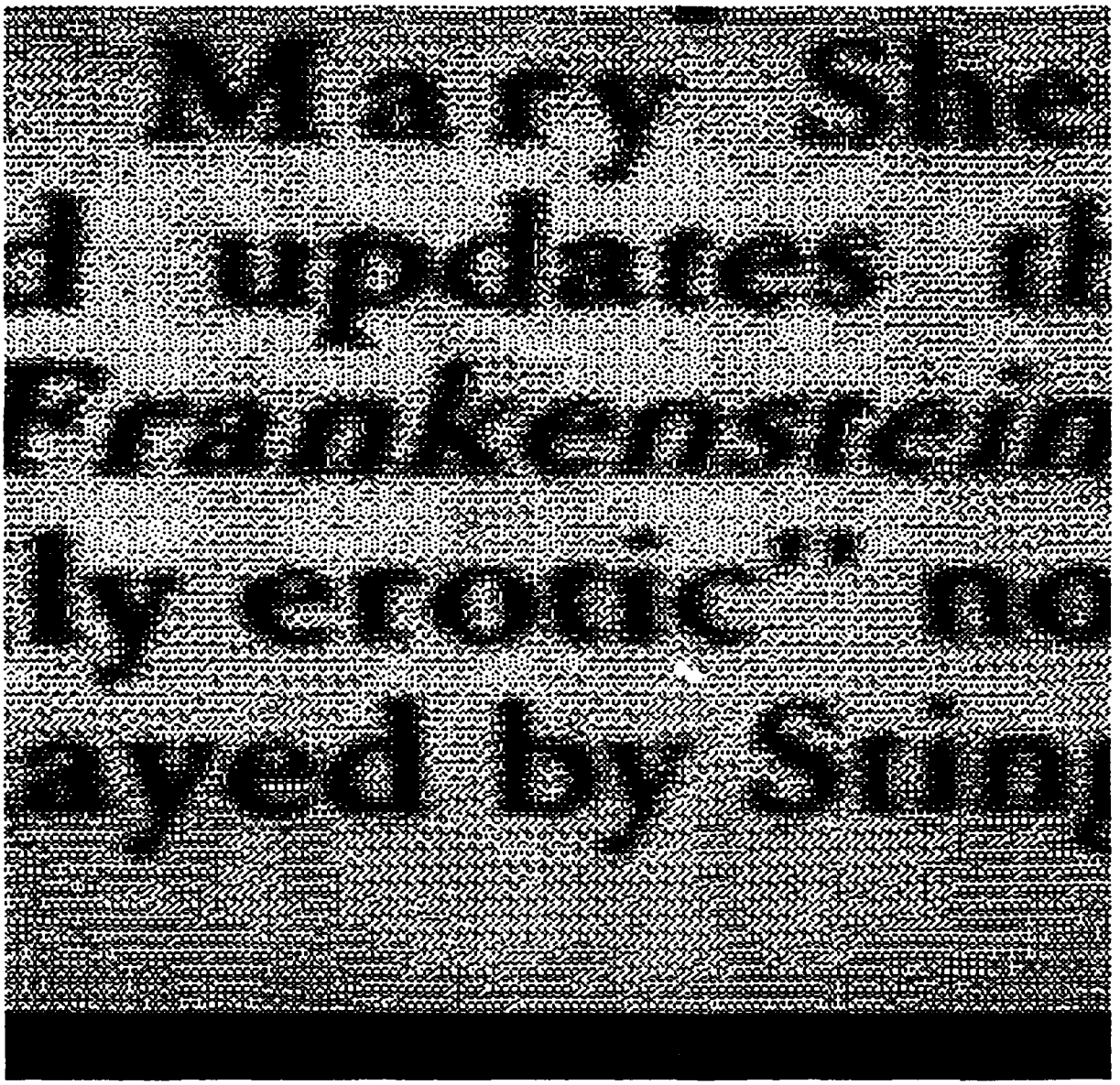


FIGURE 5.2 DIGITIZED GREYSCALE PICTURE

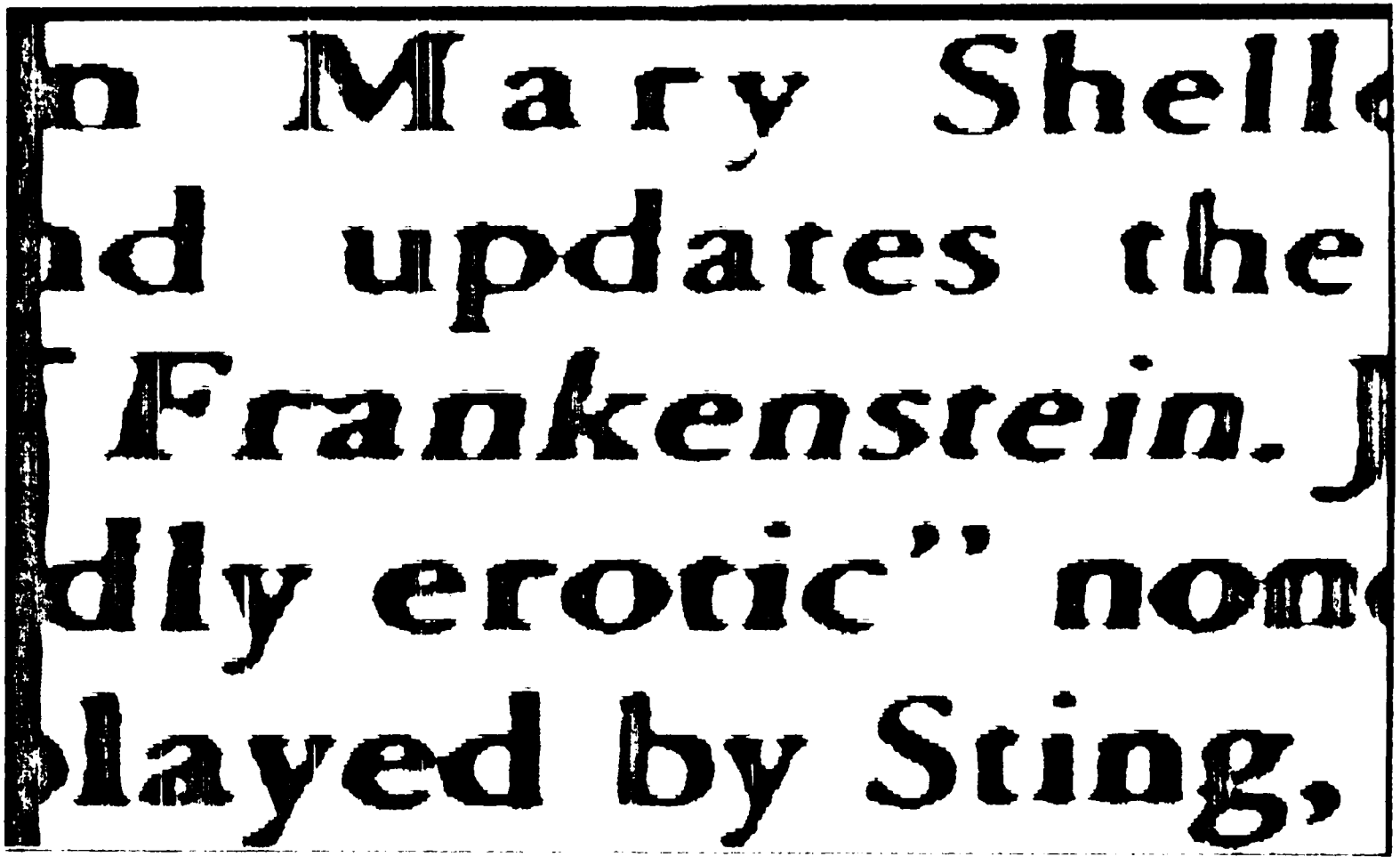


FIGURE 5.1 ORIGINAL ANALOG TEXT

The second algorithm performs a line/word locating process on the binarized text. It examines the page of text, determines the locations of the lines of print, and extracts a word from one of the lines.

5.2 NOISE REDUCTION ALGORITHM

In order to understand why a noise reduction routine is necessary, let us first examine the output of the Digitization stage. Figure 5.1 shows the original analog text. Figure 5.2 shows the raw digitized greyscale text. (The text had been properly positioned for digitization and illuminated with a 120 watt lamp.)

The greyscale version of the print contains unwanted temporal noise from the digitizer. This shows up as the sprinkling of black and white dots on the text. In addition to generating this type of noise, the quantization error of the device causes the distortion of greylevels within each character. The greylevel value of each pixel within a letter can vary from one to two levels of black. Clearly, the contrast between the characters and the background is greater in the original picture.

Conversion of the greyscale text into binary text would allow for the use of readily available digital image processing techniques. (These include line thinning and curve tracing algorithms.) Greyscale text would limit processing to techniques which do not operate on the individual pixels of a picture.

V. THE PRE-PROCESSING STAGE

The following chapter discusses the software implementation of the second level of the recognition algorithm. (See Appendix A for complete source code listings.) At this point, analog text has already been digitized and is contained in a video file. The Pre-processing stage will strip off words from the text and present them to the next stage for identification.

The first section of this chapter describes the overall processing function performed by the Pre-Processing stage. The second and third sections expand upon this overview and provide more details as to the actual operation of each of the individual programs written for implementation.

5.1 OVERALL PROCESS

The Pre-Processing stage accepts as input the digitized greyscale text from the Digitization stage and produces as its output binarized words. This is to say that it extracts one word from a page of text and outputs it as a black character on a white background. (Note that a word may be one or more concatenated letters.)

This level consists of two algorithms, each of which performs a separate operation on its input. The first algorithm performs a noise reduction operation. It takes the input greyscale text and threshold-averages the levels to produce binary text.

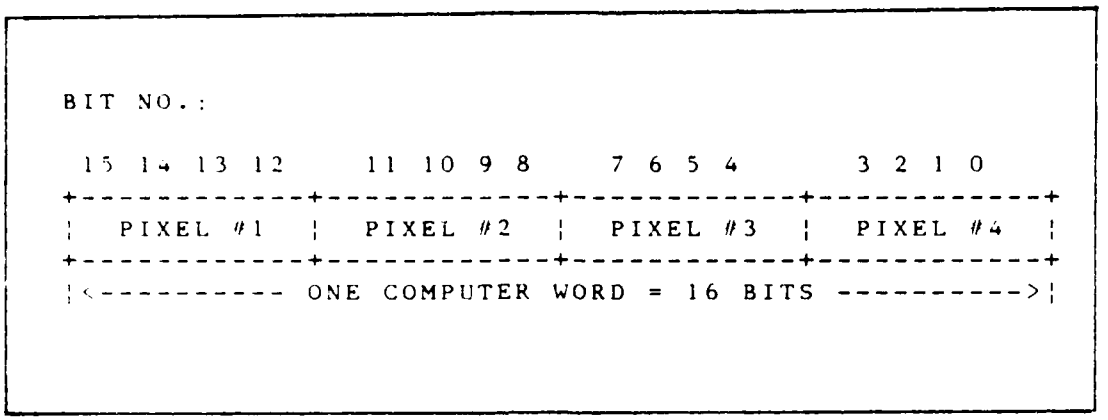


Figure 4.1 PACKED VIDEO FORMAT

As a final note on the digitization stage, it is important to remember that each greylevel in a video file is stored as an integer value of four bits in a computer word. (This becomes an important fact in later processing procedures.)

digitized greyscale pictures. The next stage would have to work with the given output.

4.2 SYSTEM ASPECTS

The digitization equipment consisted of a light table with a video camera mounted onto it. Input text placed onto the device is digitized into a 256x256 video file with 16 greylevels (0 corresponding to black and 15 to white).

In order to understand the source code of certain programs which follow, it is important to mention that the digitizer "packs" four pixels into one computer word (see Figure 4.1). One computer word is 16 bits long. Thus, processing requires the "UNPACK"-ing and the "REPACK"-ing of pixel points.

Even though a complete digitized picture is 256x256 pixels wide, the actual useable video size is only 240x256 pixels wide. The last sixteen rows of each video are zero filled to produce a black strip at the bottom of each picture. (The reason for zero filling the last 16 lines is is that the output display monitor is only 240x256 pixels.) All 65,536 pixels of a video file, however, are accessible for internal processing.

For most of the programs, the actual manipulation of pixel points was handled via PICBUF. This is a library of Fortran-callable subroutines which hides the system-dependent details of unpacking and repacking. (For more details on PICBUF see Appendix A).

VI. THE RECOGNITION STAGE

The following chapter discusses the software implementation of the final level of the proposed recognition algorithm. Since the Recognition stage is the actual level where input characters are identified, it becomes the most important level of the algorithm. Enough code was generated to allow for testing of the scheme. Thus, in most cases, the interface between programs were not written. They would only allow for total automatic processing and could be written at a later date.

Chapter 3 described the overall processing procedure for recognition. This chapter will describe the actual software implementation. Recall that identification comes from re-correlation of energy equalized characters. The first section will discuss the software for correlation and the second section will discuss the software for energy modification.

6.1 SOFTWARE DEVELOPED

The Recognition stage accepts as input words produced by the Pre-processing stage. For the reasons discussed in Chapter 2, it performs a correlation operation on the word using Fourier transforms. Chapter 3 briefly described the details of the correlation process. Figure 6.1 shows a flowchart of the programs which implement the procedure.

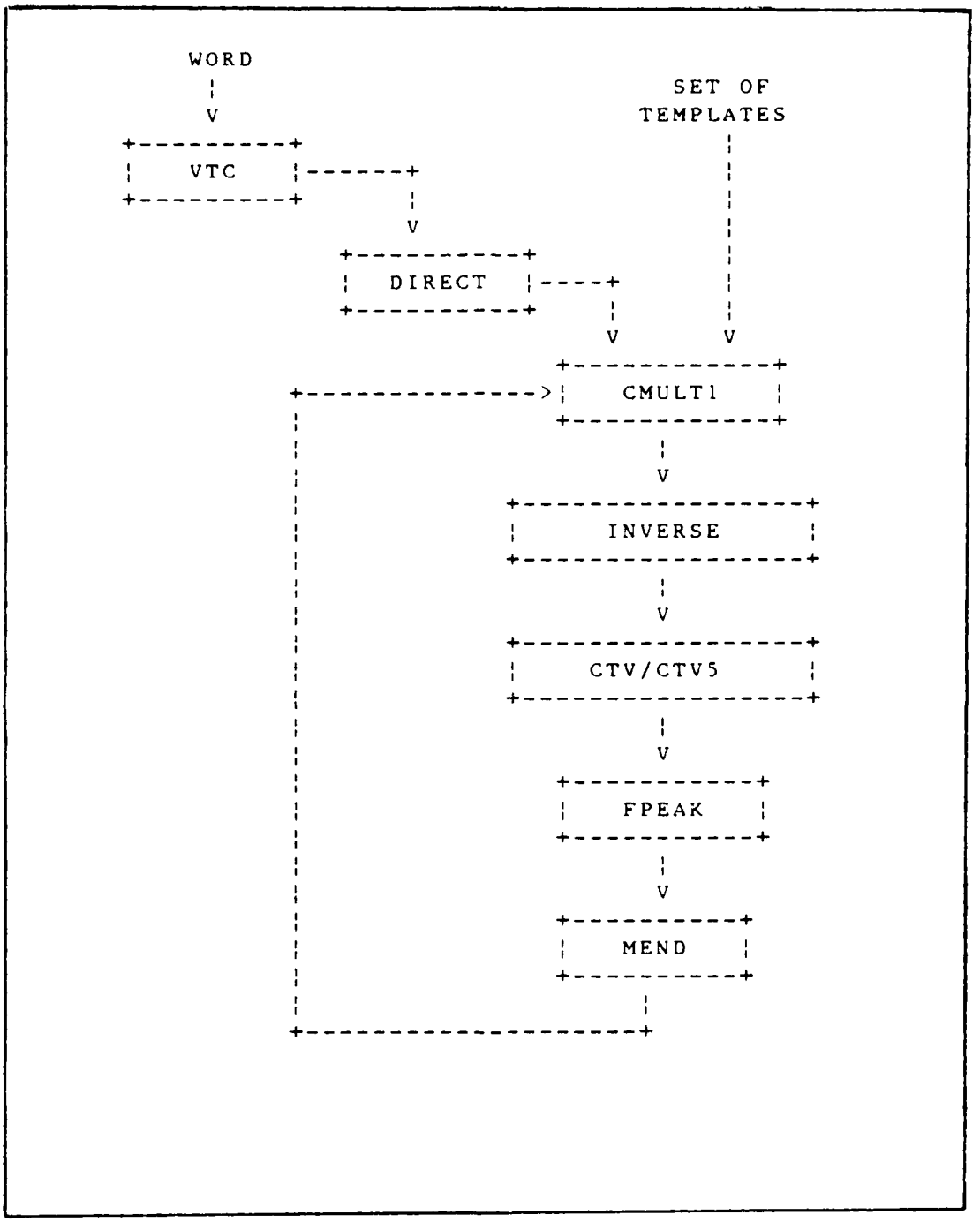


Figure 6.1 FLOWCHART OF RECOGNITION PROGRAMS

Sections 6.2 and 6.3 will describe in detail the operation of each of those programs.

As envisioned, an input word is first correlated with each of the templates in the template set. A record is then kept of the positions where each template peaked with the input. At each peak location, a box is centered on the peak and a second series of correlations is performed. During this second pass, the energy of the box and the respective template are set equal to one another. This time, correlation will produce peaks of varying values. The highest peak value determines which template correlated the best.

Note that at this time of research, some of the functions just described are not fully implemented in software. Instead of being totally automatic, the presented functions require that the user pass the values obtained from one program to that of the next.

6.2 CORRELATION PROCESSS

The first step in the correlation process is to compute the Fourier transform of the input word. This is accomplished by using the programs VTC and DIRECT. The program VTC.FR (Video To Complex) converts a video file of integer numbers into a complex file with the imaginary part set equal to zero.

The program DIRECT.SV performs the actual computation of the Fourier Transform. It was a supplied program which

could only be executed. The original source code was not available.

Since the characters of the template set are known beforehand, the Fourier transform of each template is computed before the beginning of the process. This computation is performed only once and the results are stored in memory. The computation of the transform for the templates requires an additional program, QSHIFT.FR (QuarterSHIFT). In order for the correlation peaks to be centered properly on a letter, the template must be quartered and the two diagonal corners must be interchanged.

The program which performs the actual correlation process is CMULT1.FR (Complex Multiplication). From Digital Signal Processing, recall that correlation in the spatial domain corresponds to multiplication in the frequency domain. The program multiplies the complex value of a pixel from one file with the complex conjugate value of another file. It therefore, performs the correlation process by complex conjugate multiplication on a pixel by pixel level.

The process is completed by inverse Fourier transforming the correlated file and converting the complex file into a video file. The program INVERSE.SV calculates the inverse transform. The program CTV.fr (Complex To Video) performs the conversion.

The program CTV.FR linearly scales the values of the

complex file so that the highest value in the file has a value of 15. Therefore, this program will always produce a peak correlation value of 15. (After the inverse transform operation, the video file contains complex numbers with the imaginary part equal to 0.)

After correlation, the program FPEAK.FR (Find Peak) will determine the highest correlation peak in the correlated scene and determine the location, or loactions, of the peak. (These values can be passed on to the energy modification programs. At this point in time, these values must be manually entered.)

6.3 ENERGY MODIFICATION PROCESS

As discussed in Chapter 2, a correlation peak is maximum if the two shapes have the same energy and the same shape. After the first correlation process determines the locations of peaks, the area centered on the peak must be energy modified to equal that of a template. For the present implementation, if the energy in the area is greater than that of a template, then the energy of the area is set equal to that of the template. If the template has higher energy, then the template energy is set to that of the area. The program which performs this operation is MEND.FR (Modify ENergy Distribution).

For greyscale characters, the energy in a word is computed by taking the square root of the sums of the squares of the individual pixel values within a given area:

$$\text{ENERGY} = \sqrt{\sum_{I}\sum_{J}\left[\ \text{PIXEL VALUE}(I,J)\ \right]^2}$$

FOR ALL (I,J) IN THE AREA

Since all of the characters in this level have been Pre-Processed, the energy becomes the square root of the number of black pixels in a given area.

After the energies are established, both template and input are re-correlated. The same procedure follows as in the first correlation pass, except that at the end, program CTV5.FR is used. Like CTV.FR, this program performs a linear scaling of the complex pixel values, but at a fixed range. Therefore, each correlation will produce a different correlation peak.

VII. RESULTS

The following chapter presents the results of some sample tests conducted to evaluate the performance and limitations of the proposed recognition scheme. It is expected that research will continue until a fully operational device is developed.

The first series of tests were performed with letters from the same font style. The second series of tests were performed with two differing font styles.

Throughout testing, it was assumed that the maximum width and height of a given character was known. This provided the dimensions for the window in the energy modification process.

7.1 SAME FONT STYLE

Figure 7.1 shows the characters of the template set used for testing. Single letters and concatenated letters were generated from this template set and used for the first series of tests. (Appendix A contains the source listing of the programs used to generate the concatenated letters.)

The first test examined the recognition scheme's ability to recognize single letters. The second series of tests examined its ability to recognize letters within a string of letters.

7.1A SINGLE LETTER TEST

Figure 7.2 shows a capital letter F. This character was input into the Recognition stage for identification. This produced a series of correlation peaks. Figures 7.3 through 7.7 show the correlation peaks for the second pass with the letters B, C, E, F, and H. The peak value for the F template was the highest.

7.1B CONCATENATED LETTERS

Figure 7.8 shows five concatenated letters as the output of the second stage. Figures 7.9 through 7.13 shows the results after the first correlation pass with the "A", "B", "C", "D", and "E" templates, respectively. The highest correlation peak at the first position in the word was for the "A" template. The highest peak at the second position was for the "B" template. The highest correlation peaks for the third, fourth, and fifth positions were for the "C", "D", and "E" templates, respectively.

Correlating the word with the "F" template also produced a peak at the fifth positon (see Figure 7.14). A window was therefore centered at the location. Figure 7.15 shows the peaks for re-correlating the "E" template with the window energy set equal to that of the "E" template. Figure 7.16 shows the same peaks for the "F" template. The value of the second correlation peak was highest for the "E" template.

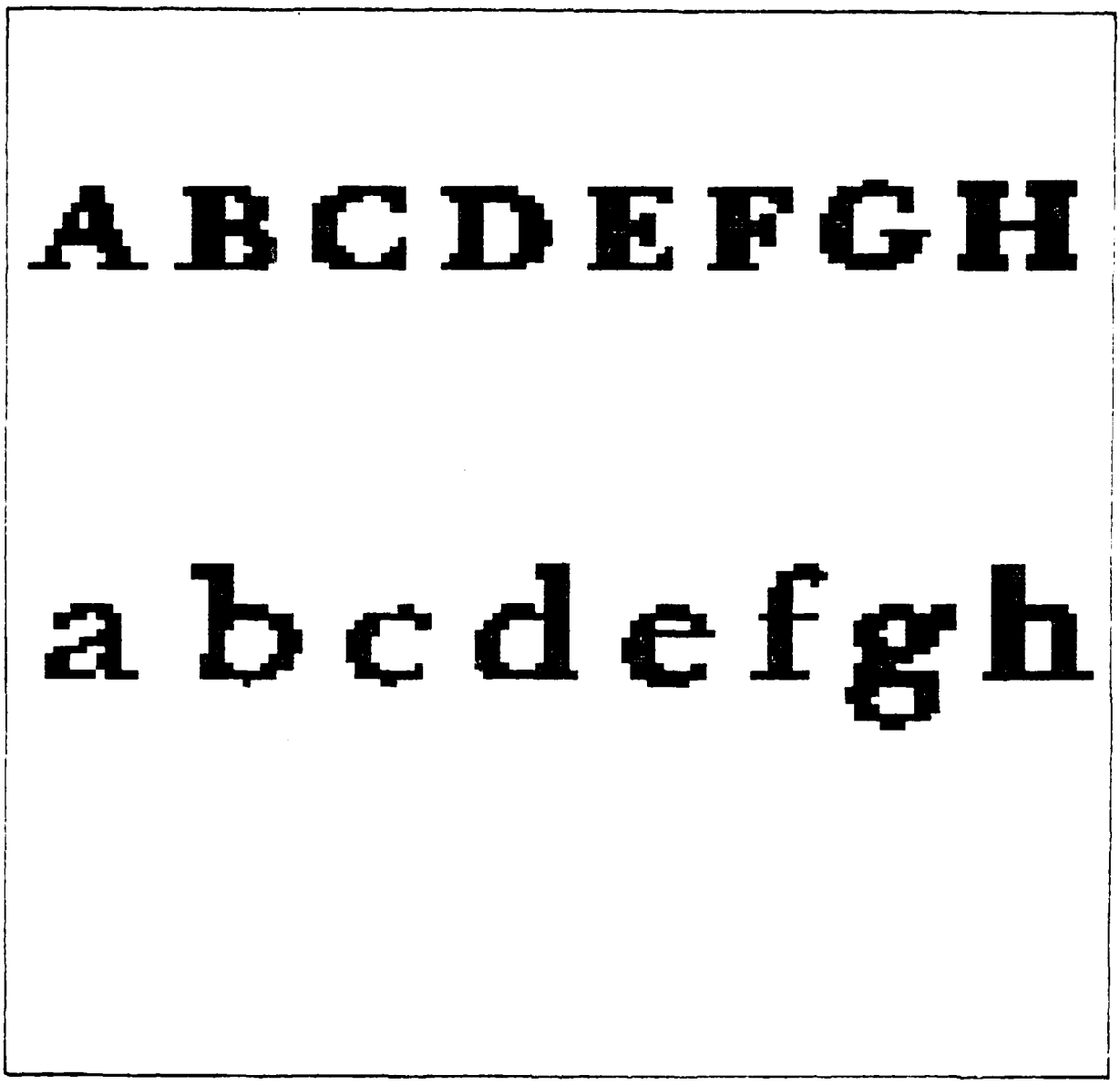


FIGURE 7.1 TEMPLATE SET NUMBER 1

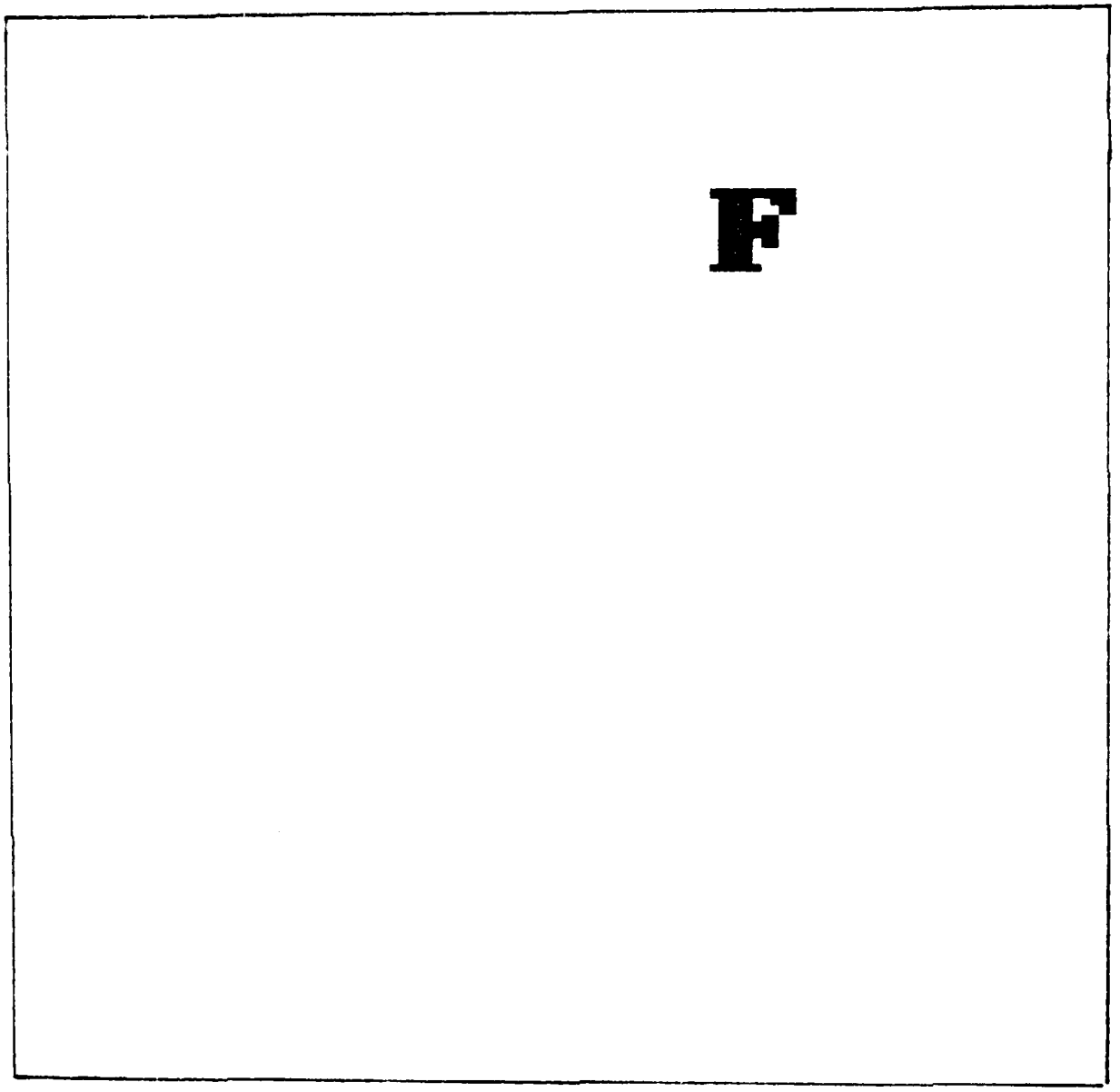


FIGURE 7.2 INPUT FOR RECOGNITION

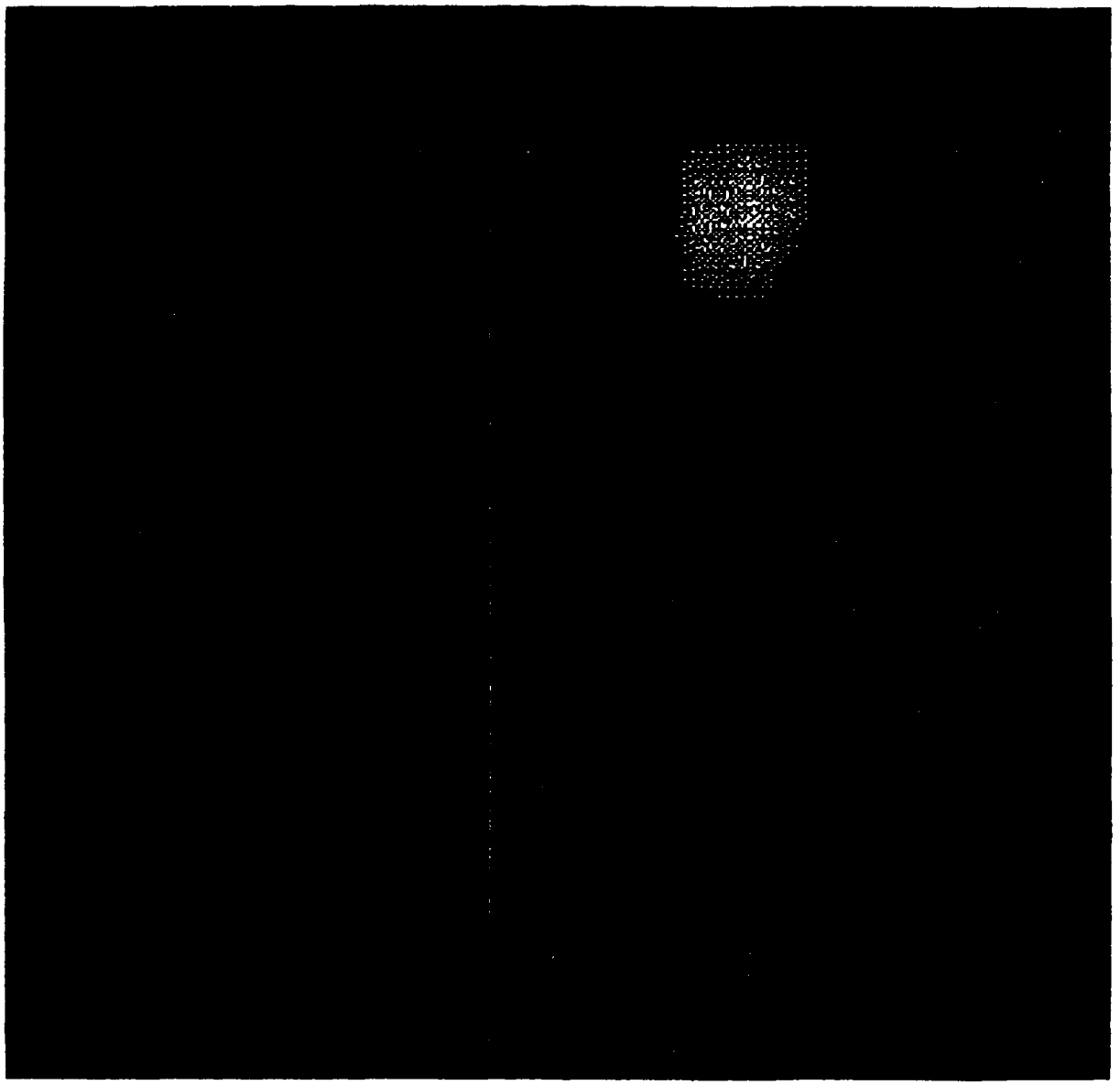


FIGURE 7.3 INPUT CORRELATED WITH TEMPLATE B

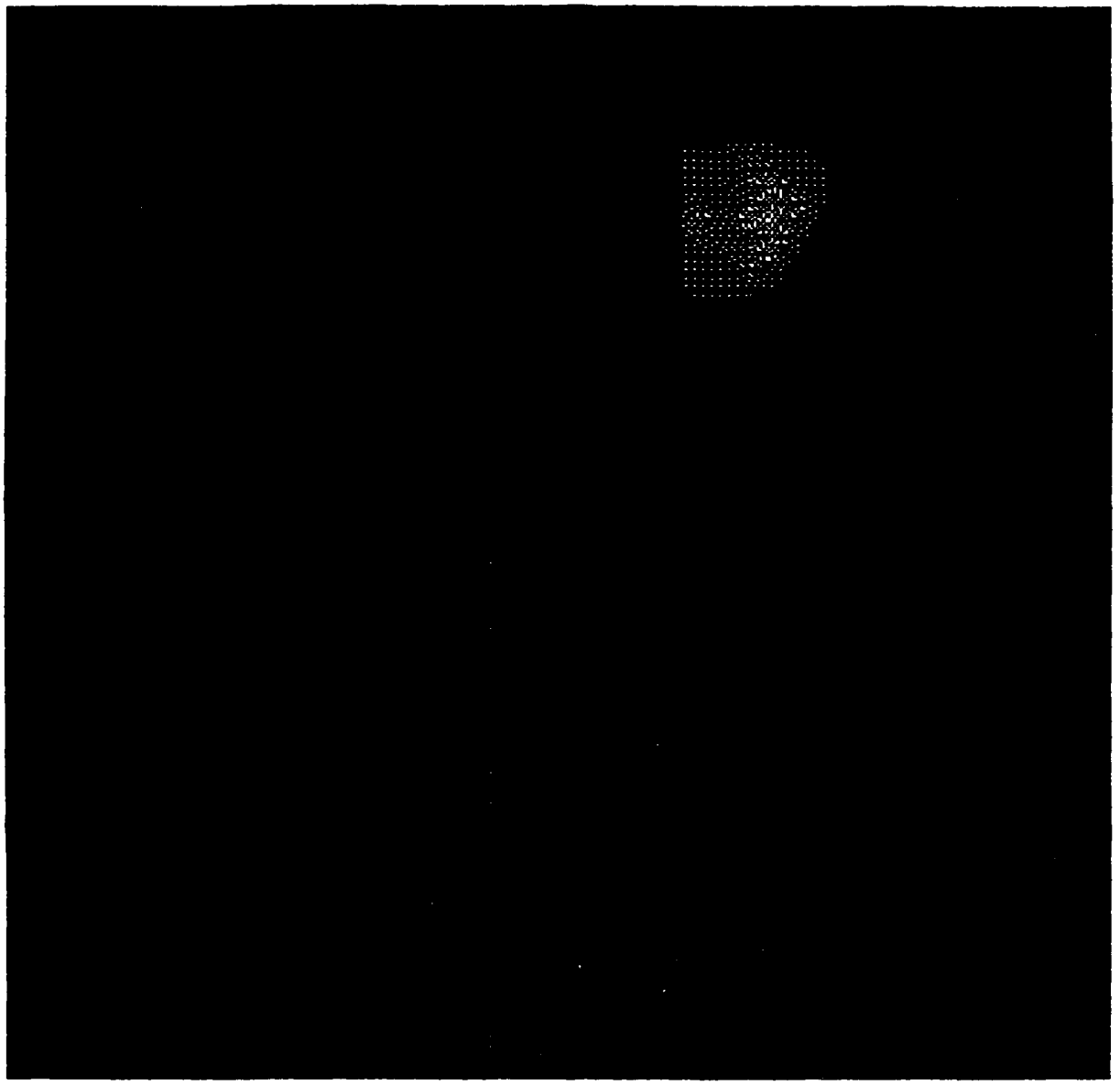


FIGURE 7.4 INPUT CORRELATED WITH TEMPLATE C

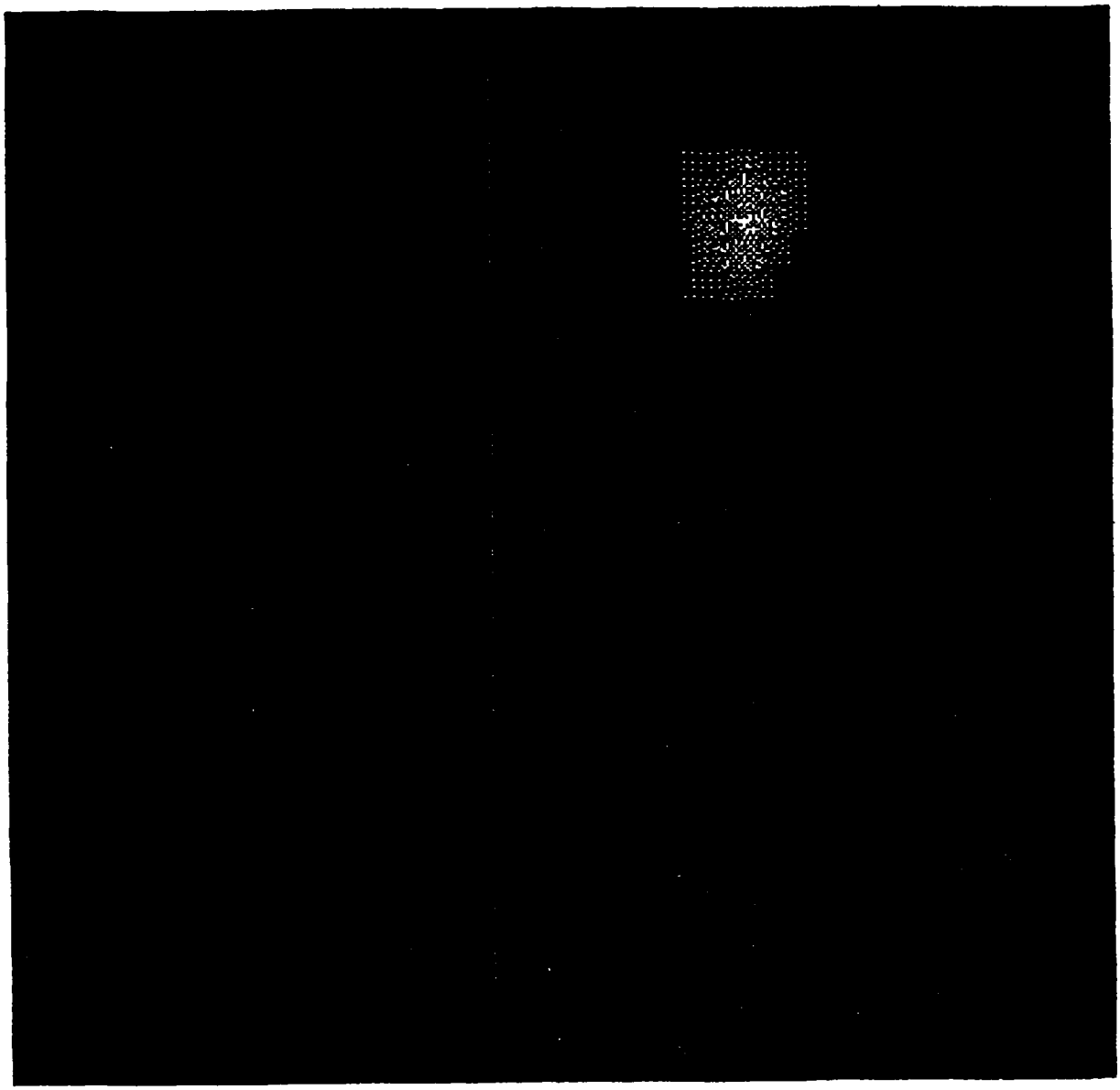


FIGURE 7.5 INPUT CORRELATED WITH TEMPLATE E

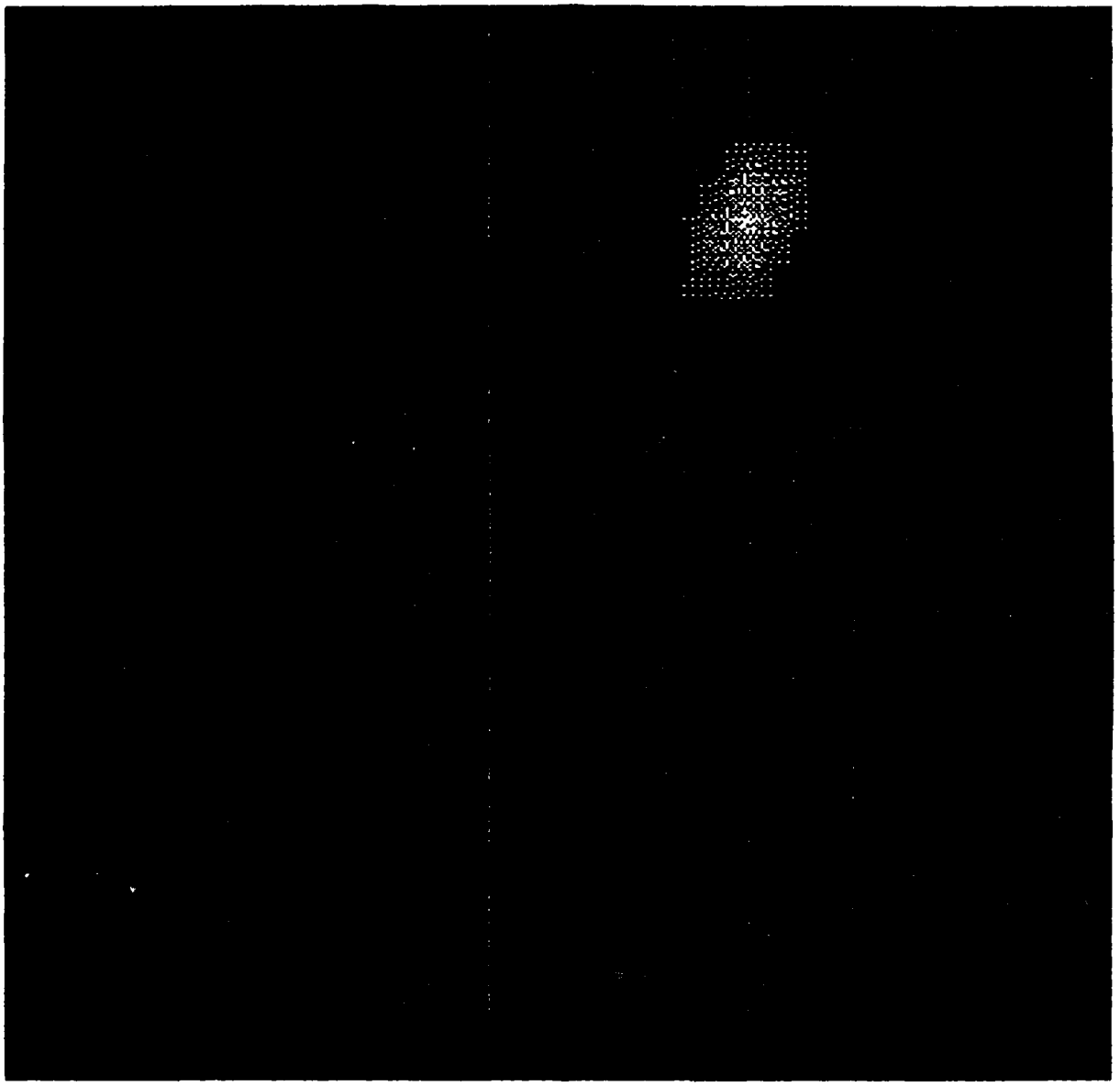


FIGURE 7.6 INPUT CORRELATED WITH TEMPLATE F

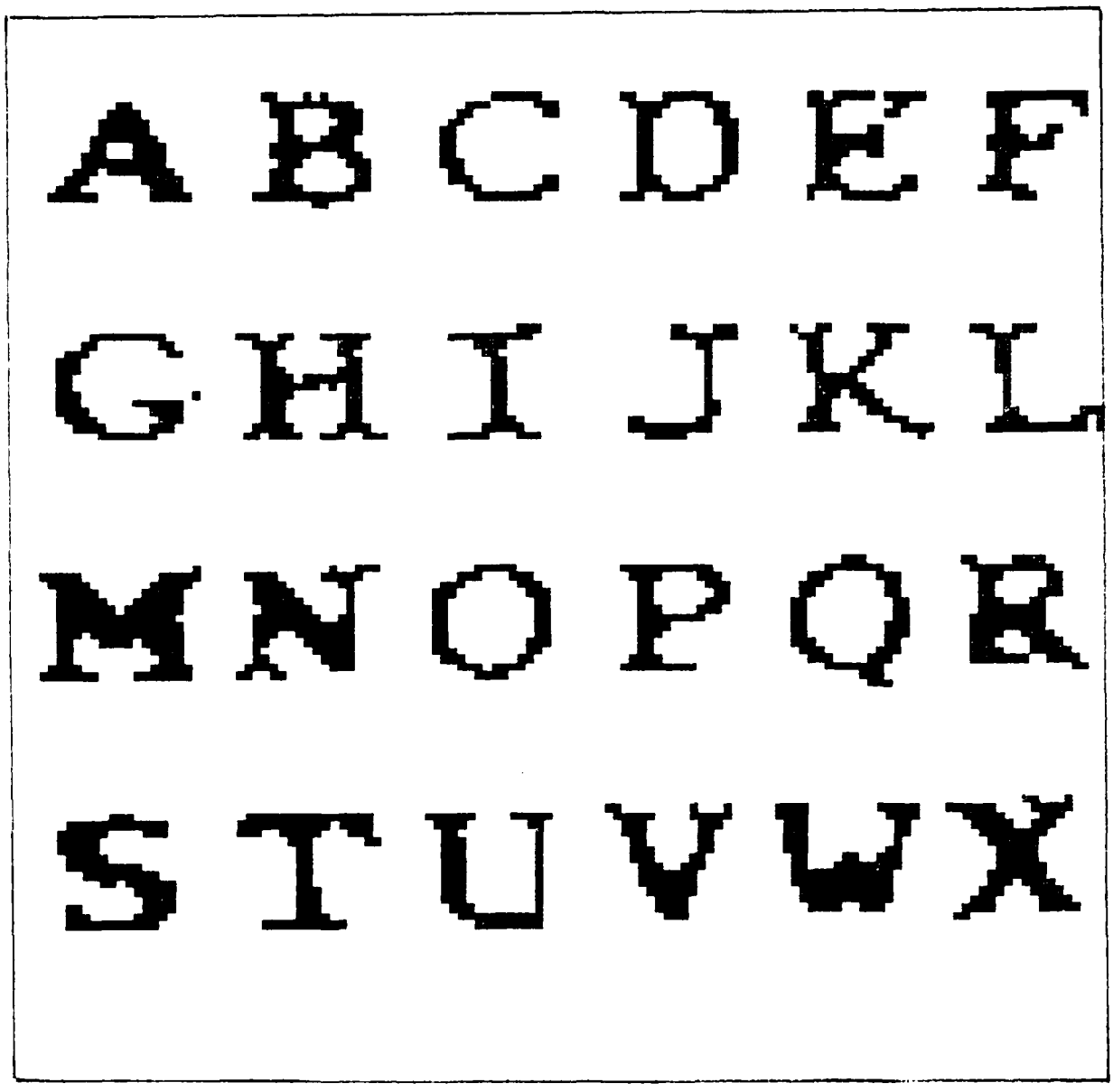


FIGURE 7.19 THINNED VERSION OF TEST SET

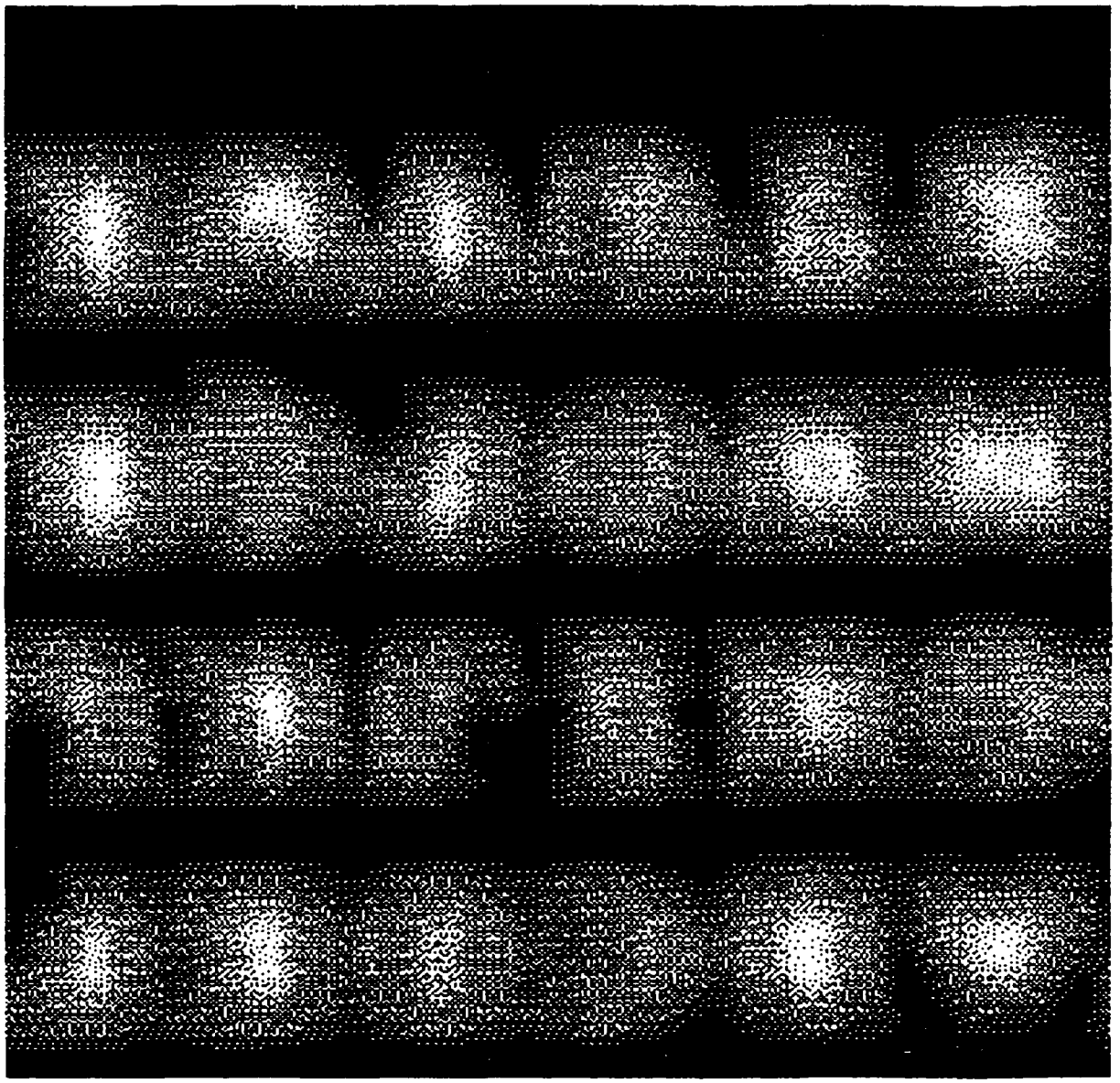


FIGURE 7.18 TEST SET CORRELATED WITH TEMPLATE B

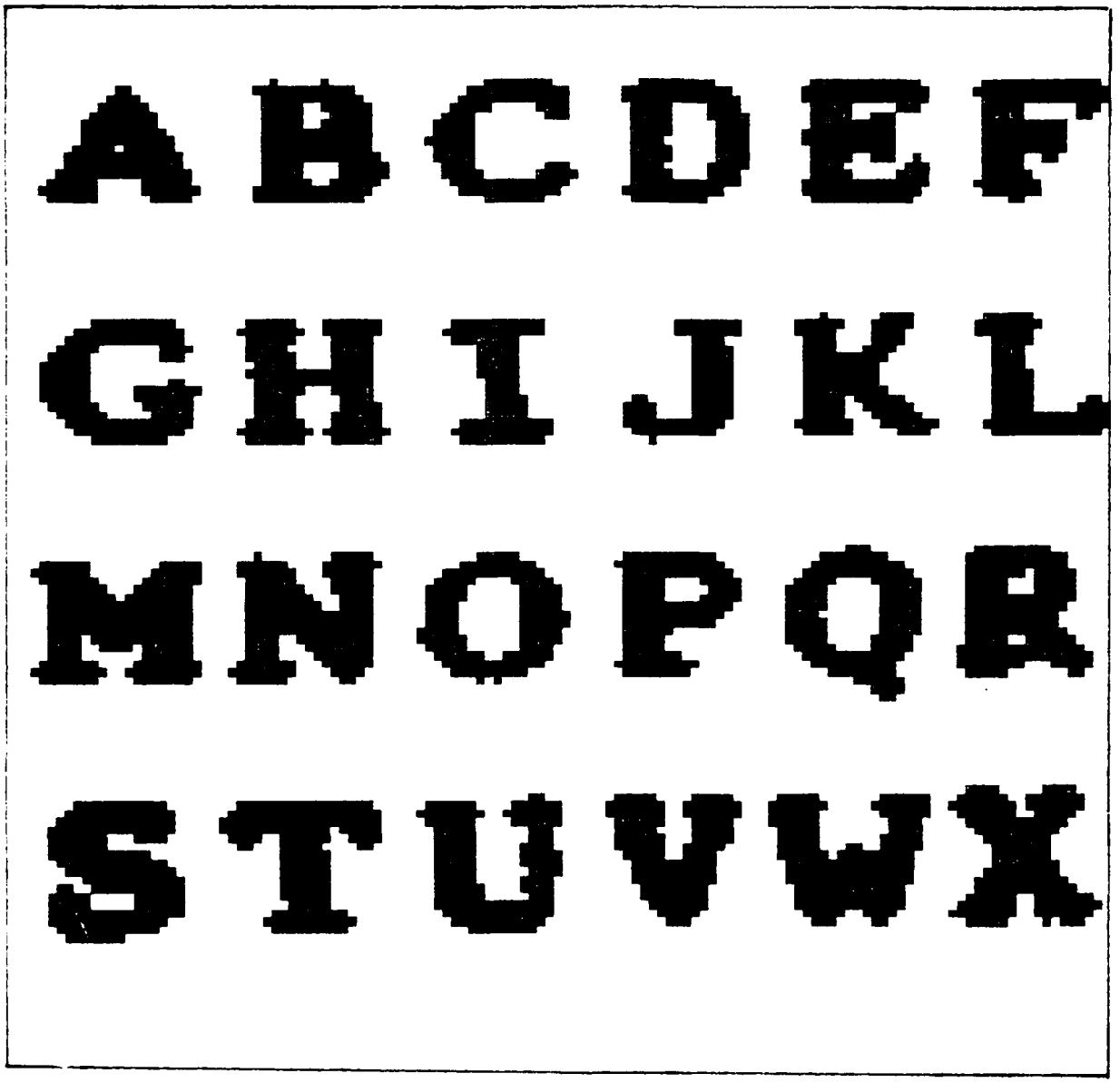


FIGURE 7.17 TEST SET NUMBER 2

7.2 LARGER FONT STYLE

Figure 7.17 shows the characters of another font style. These letters are approximately 30% larger (total amount of pixels contained in a character) than those of the first template set. This second template set was used as input to the Recognition stage.

The first pass correlation of the template "B" (from the first template set) with the second template set produced the peaks shown in Figure 7.18. The template "B" correlated well with every input character.

From the discussion in Chapter 2, it appeared as if the recognition algorithm might be helped if the target letters were not so thick. Figure 7.19 shows a thinned version of the second template set. (The program VTHIN.FR was used to thin the letters. The source code listing for this program is in Appendix A.) The template "B" of the first set was then correlated with the thinned version. Figure 7.20 shows the resulting peaks. The template "B" produced a peak with the letter G.

Since the first correlation pass did not produce a peak at the proper loacation testing was not continued. It appears as if the "template height matching" function of the algorithm must be implemented. However, time limitations did not allow for this.

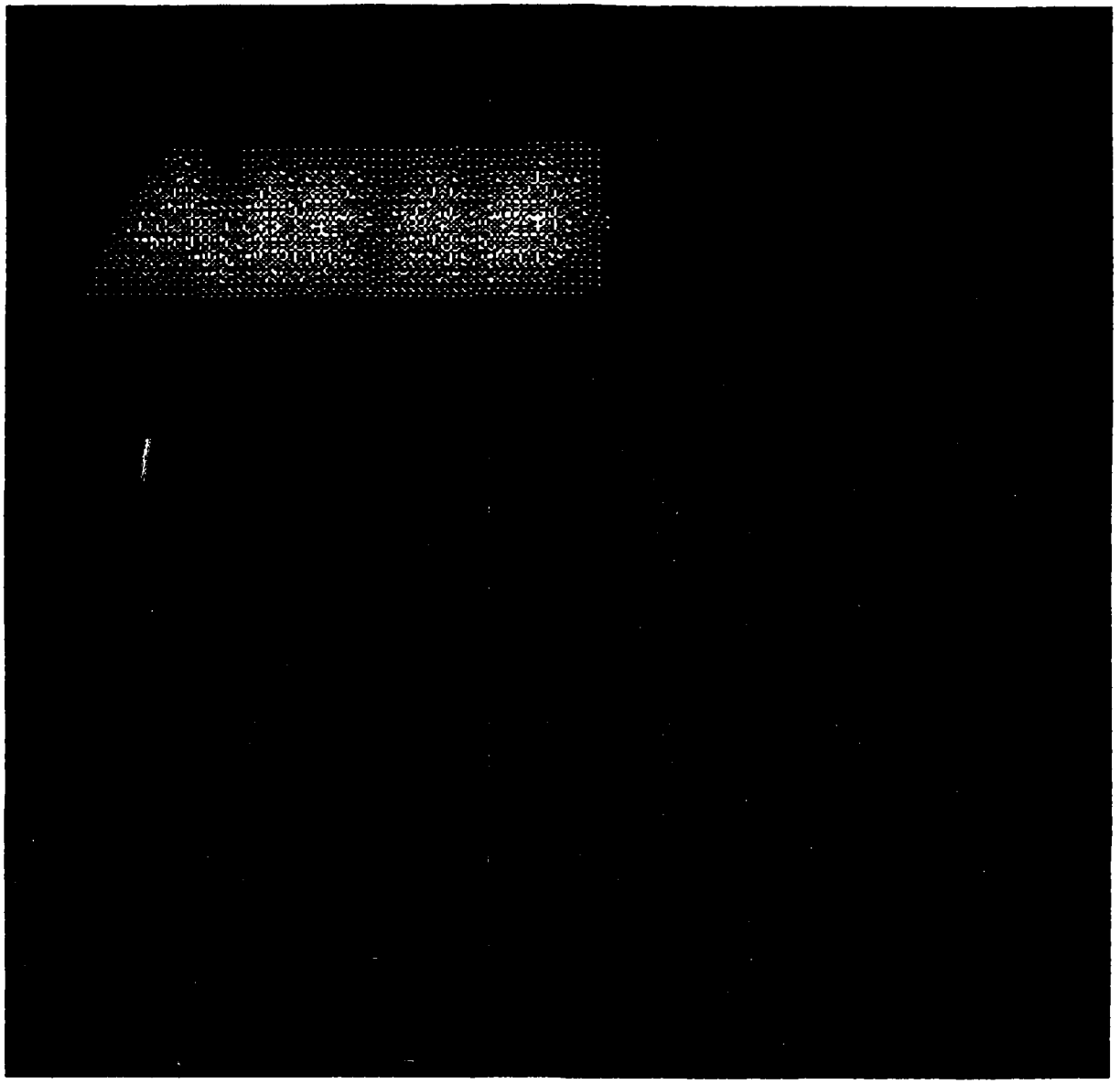


FIGURE 7.16 CORRELATION WITH ENERGY SET TO TEMPLATE F

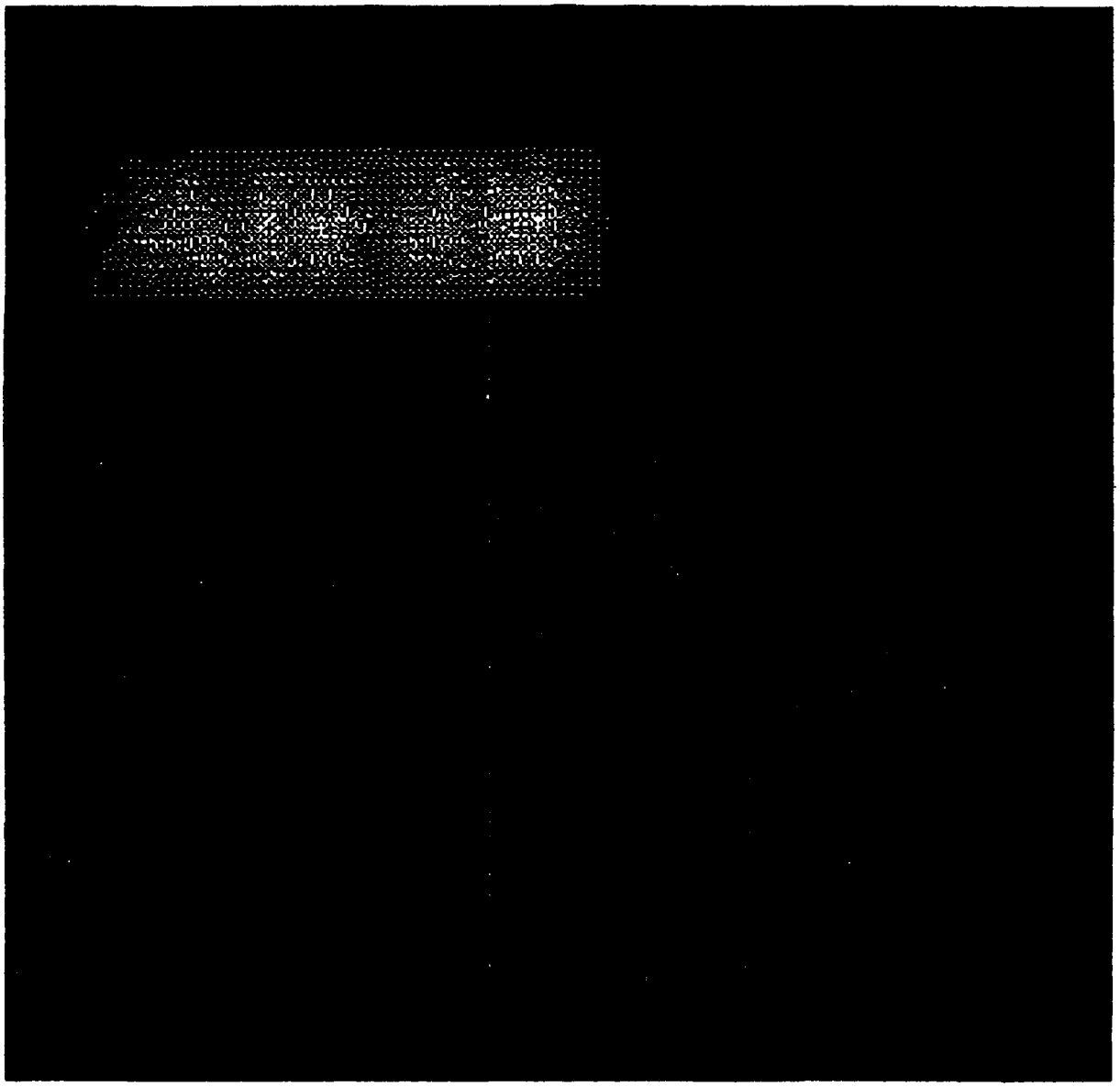


FIGURE 7.15 CORRELATION WITH ENERGY SET TO TEMPLATE E

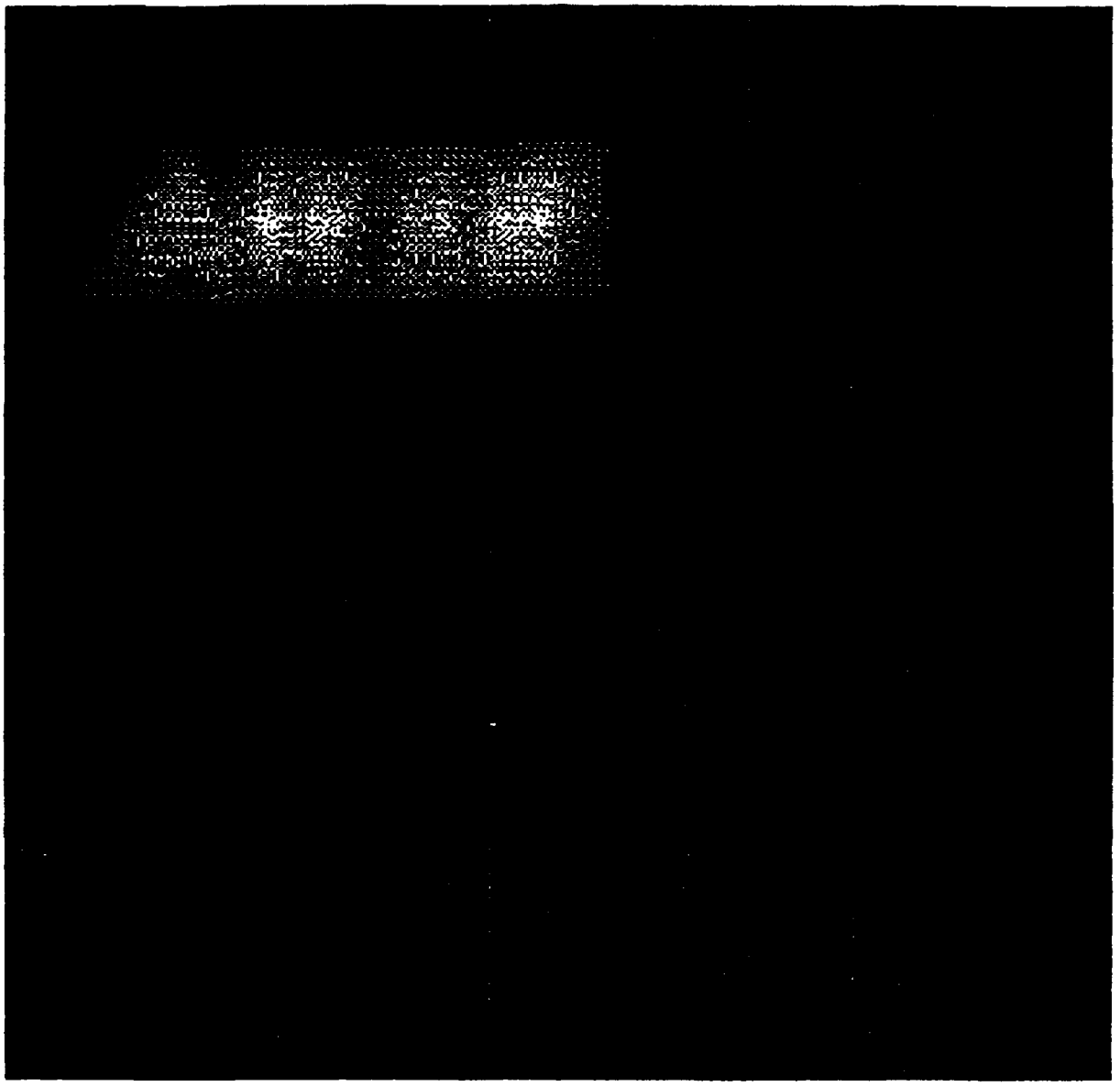


FIGURE 7.14 CORRELATION WITH TEMPLATE F

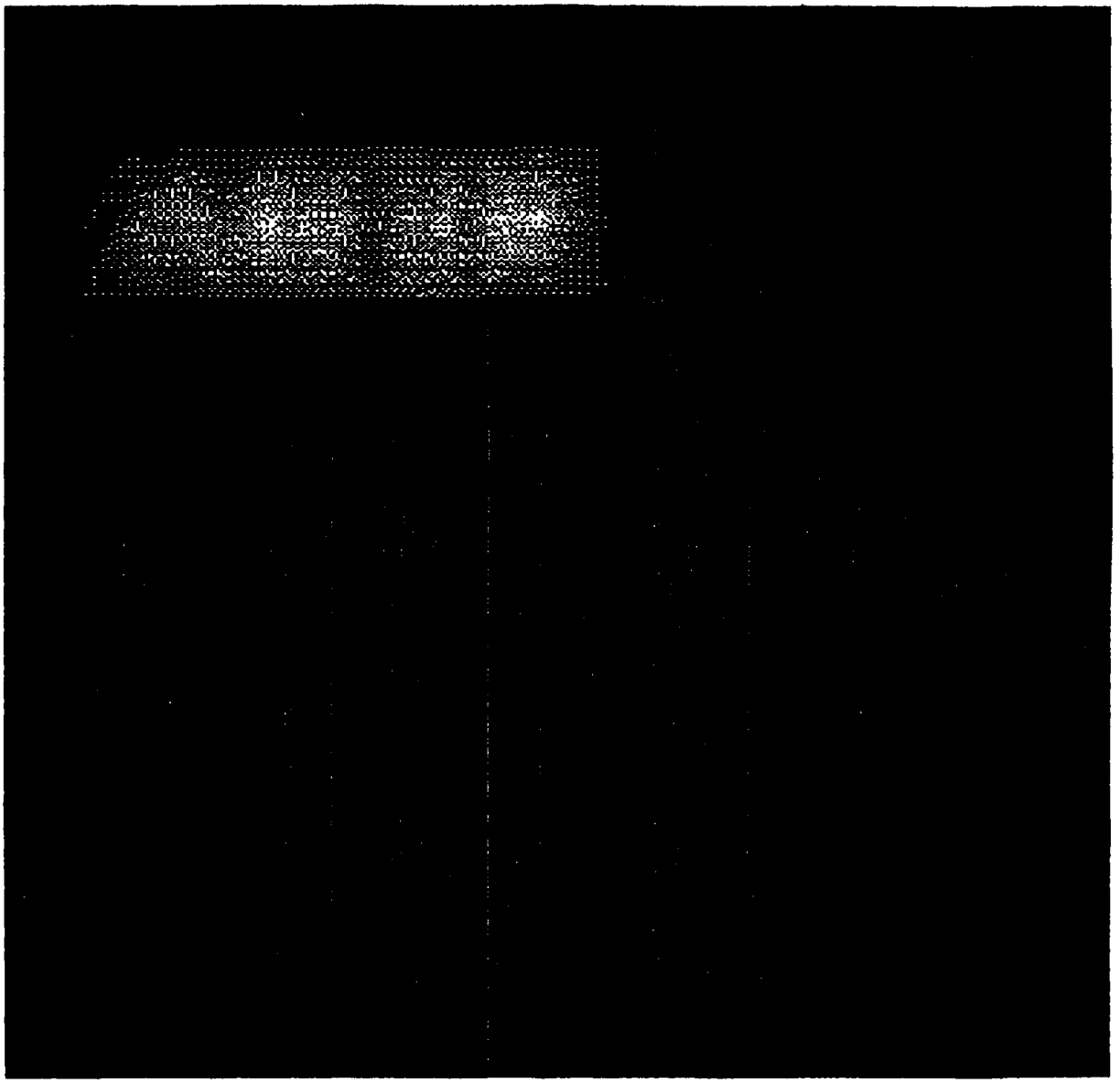


FIGURE 7.13 CORRELATION WITH TEMPLATE E

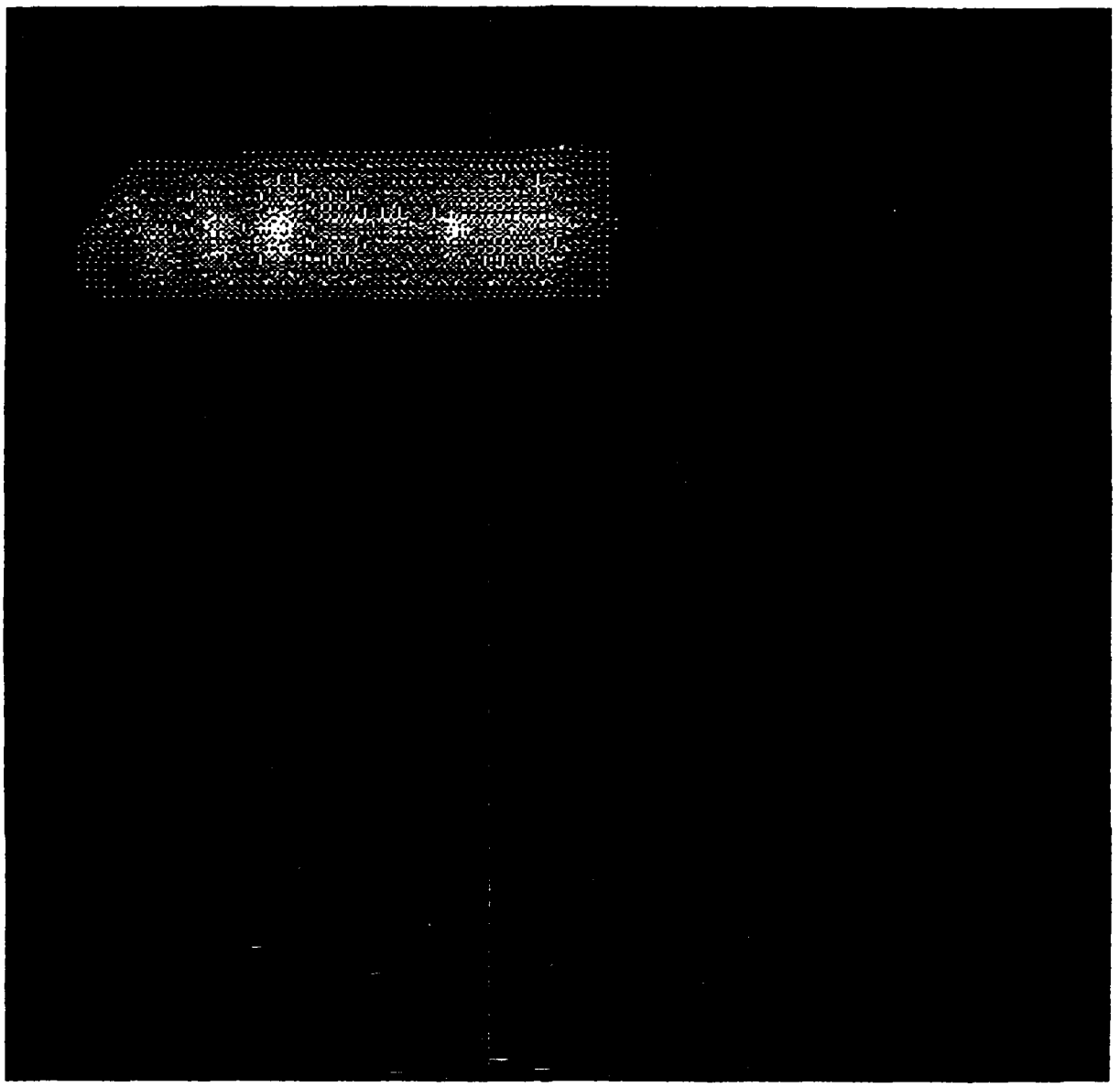


FIGURE 7.12 CORRELATION WITH TEMPLATE D

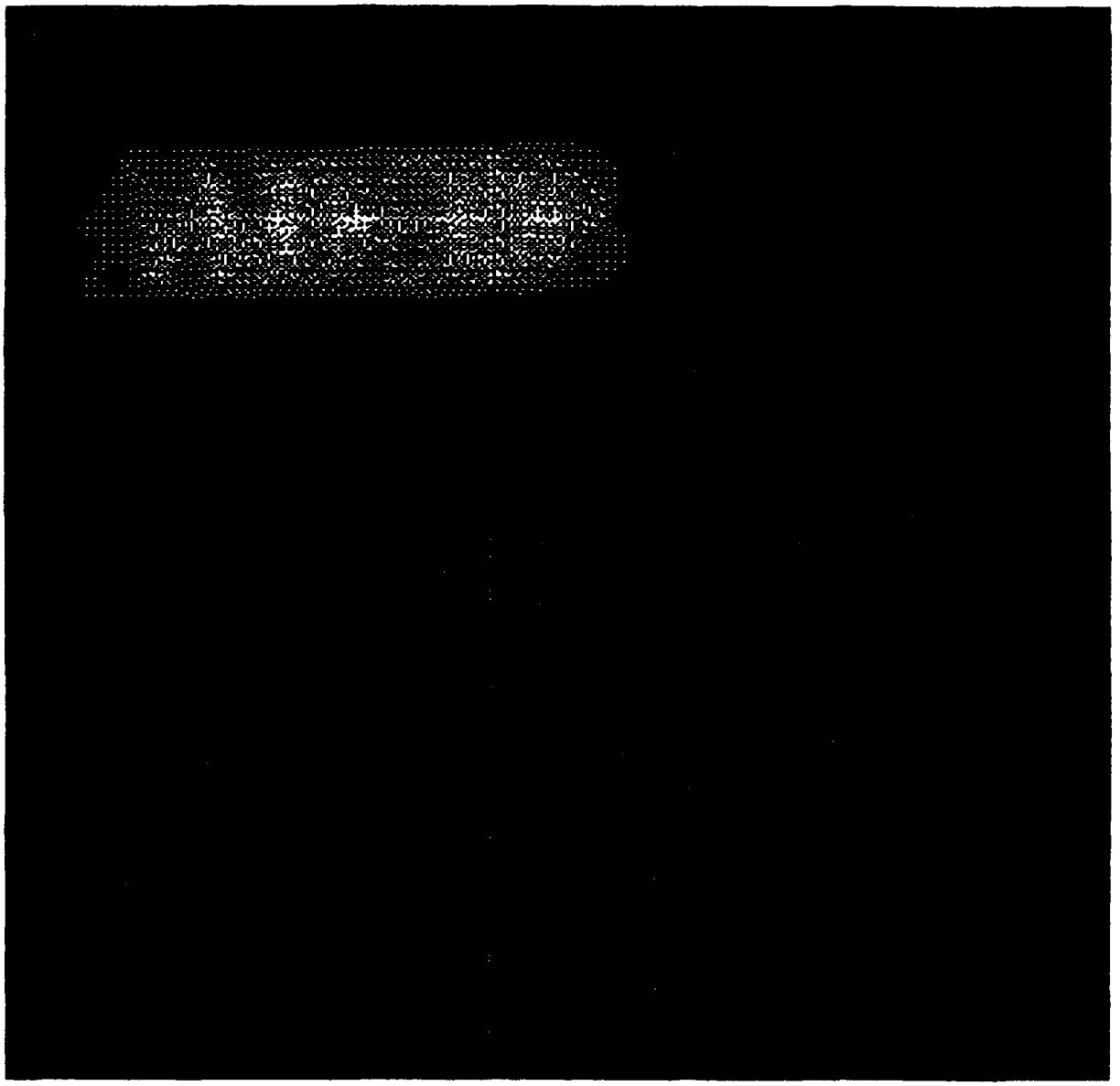


FIGURE 7.11 CORRELATION WITH TEMPLATE C

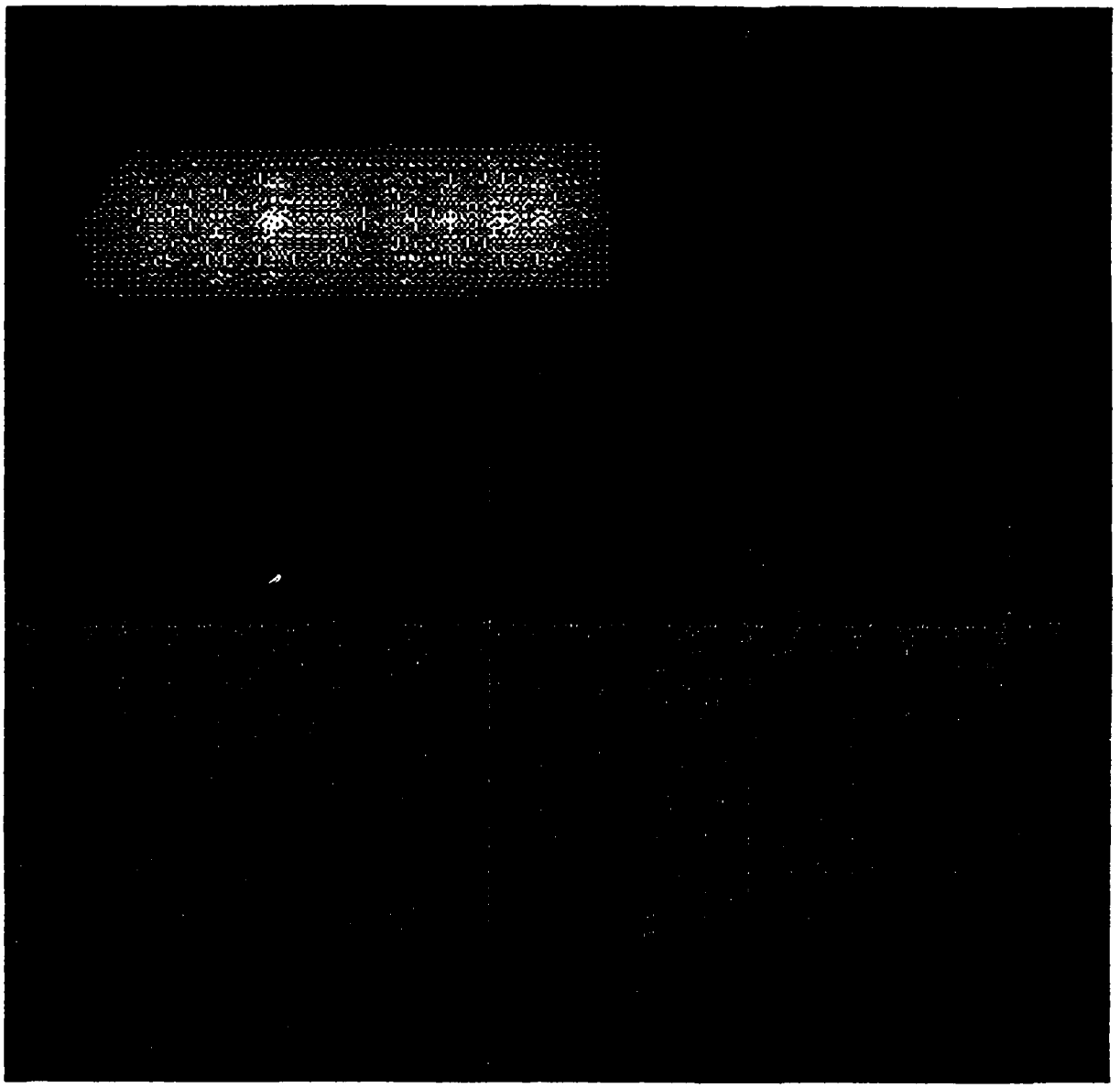


FIGURE 7.10 CORRELATION WITH TEMPLATE B

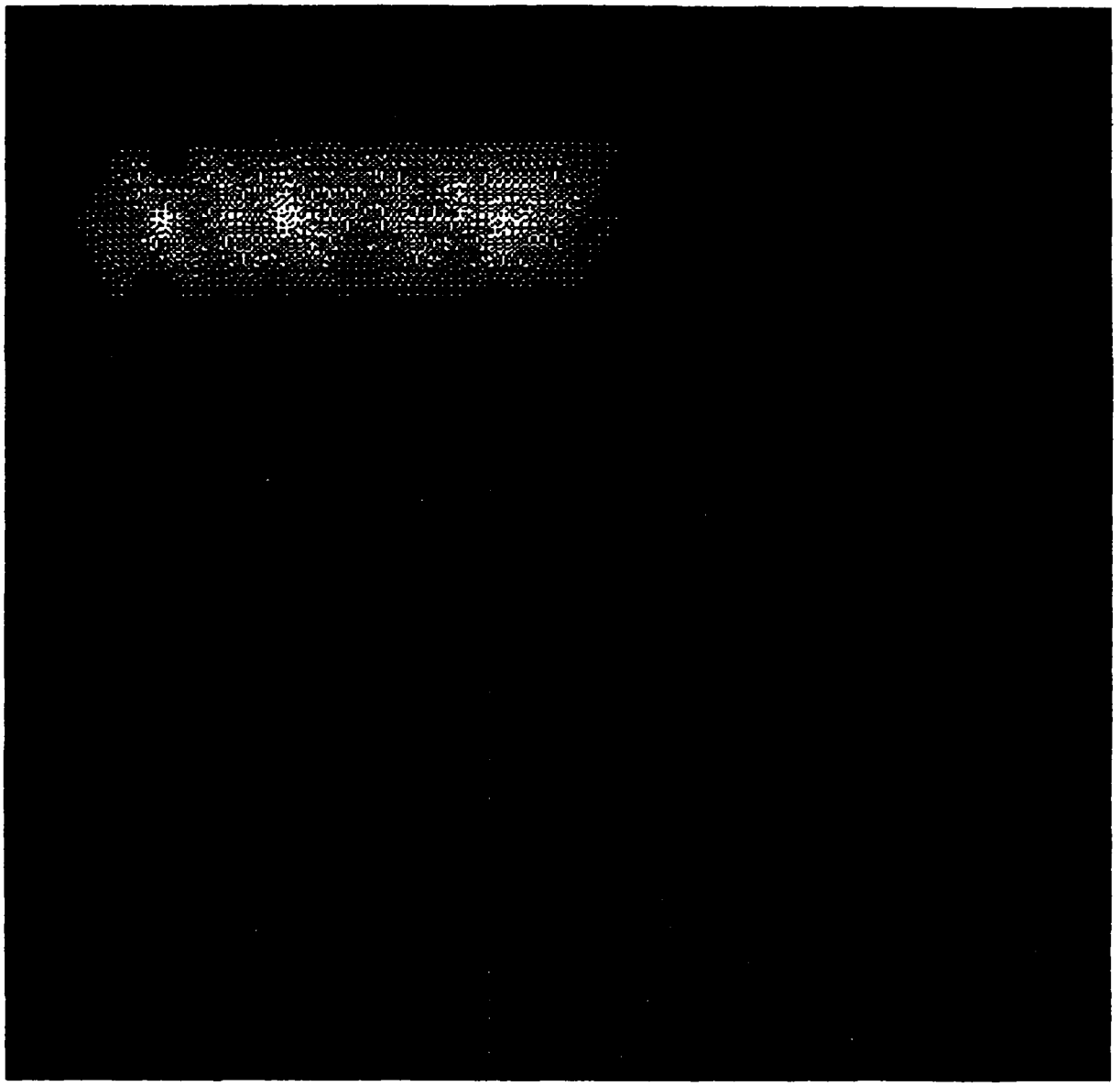


FIGURE 7.9 CORRELATION WITH TEMPLATE A

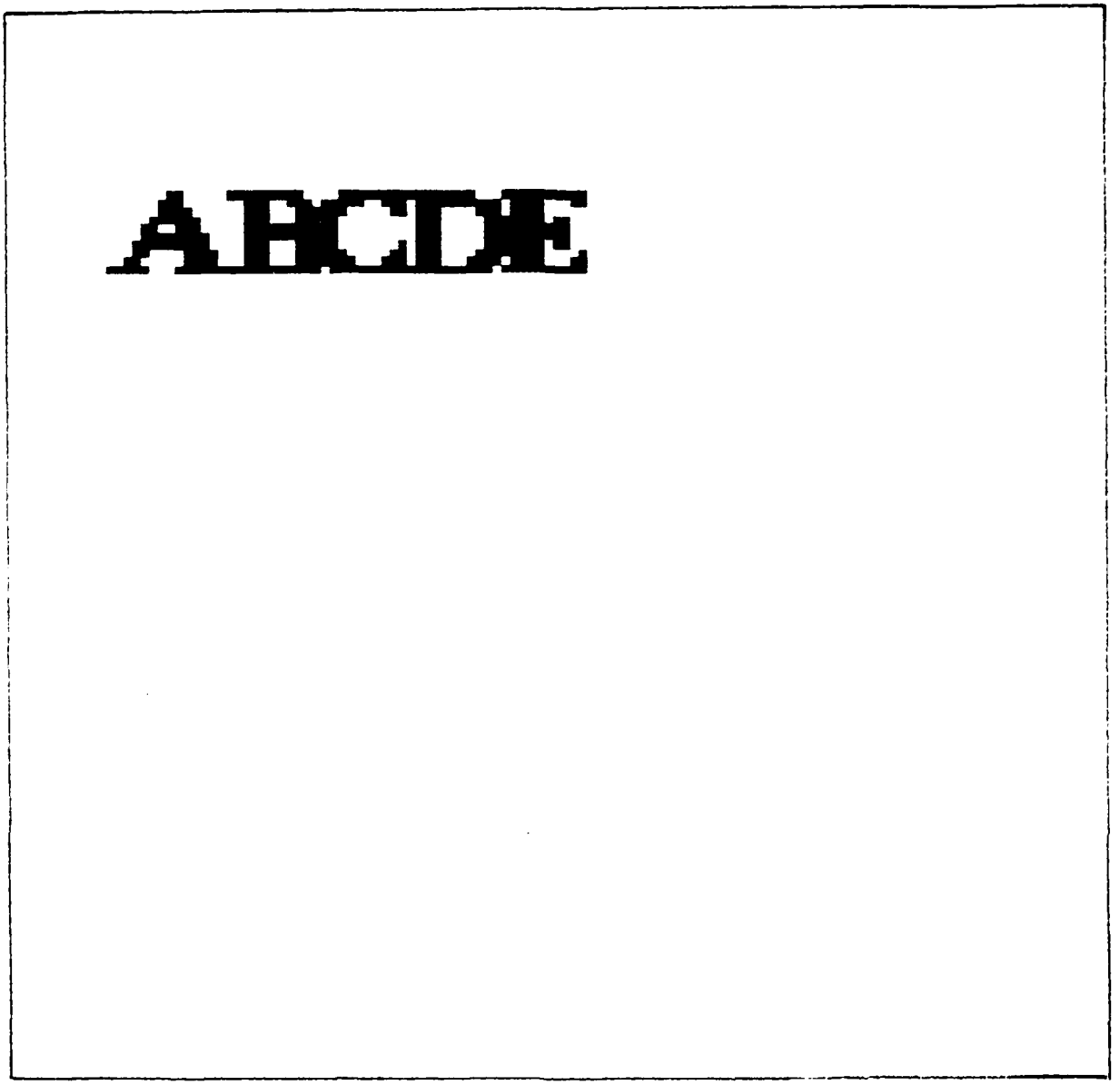


FIGURE 7.8 FIVE CONCATENATED LETTERS

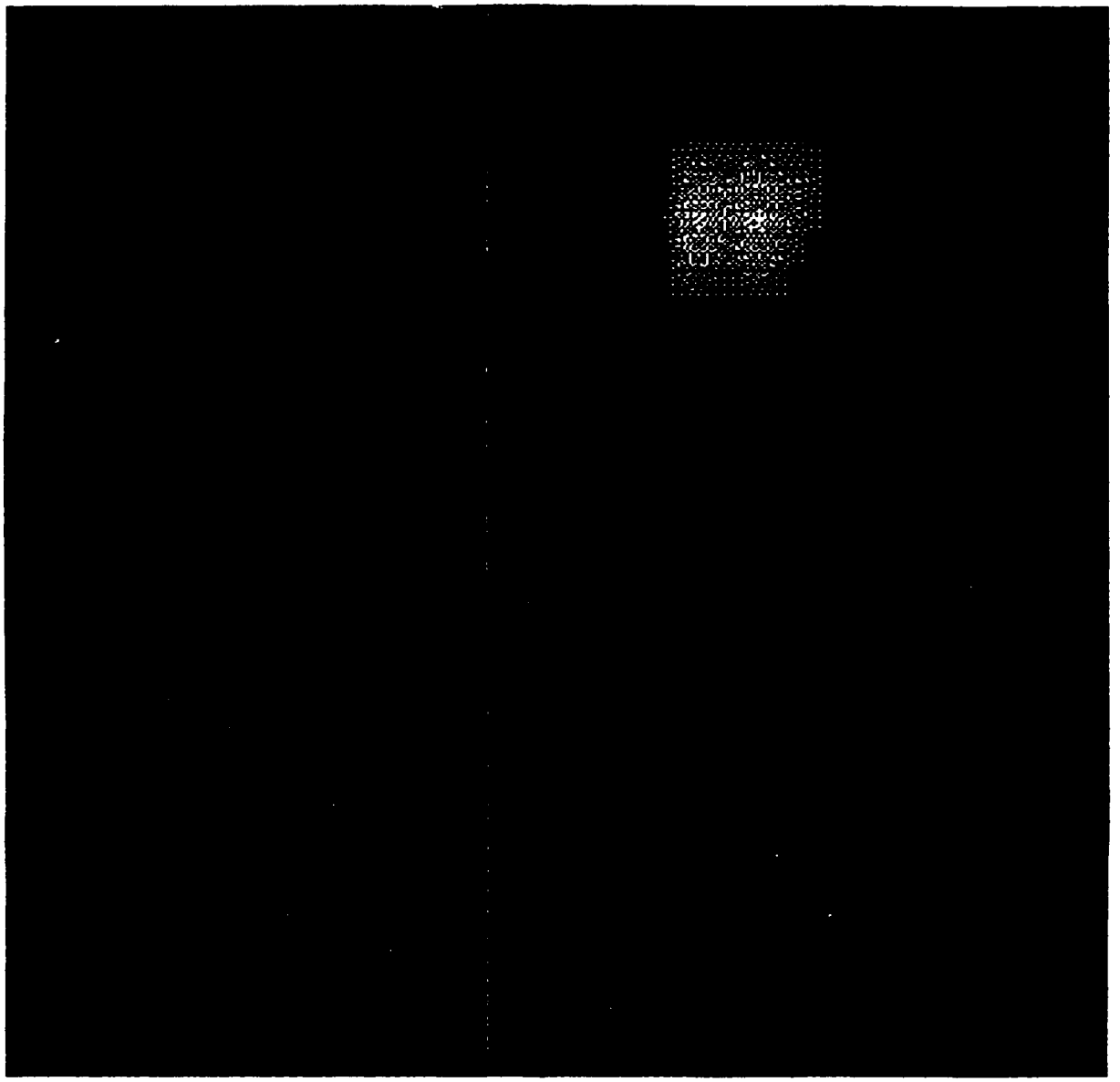


FIGURE 7.7 INPUT CORRELATED WITH TEMPLATE H

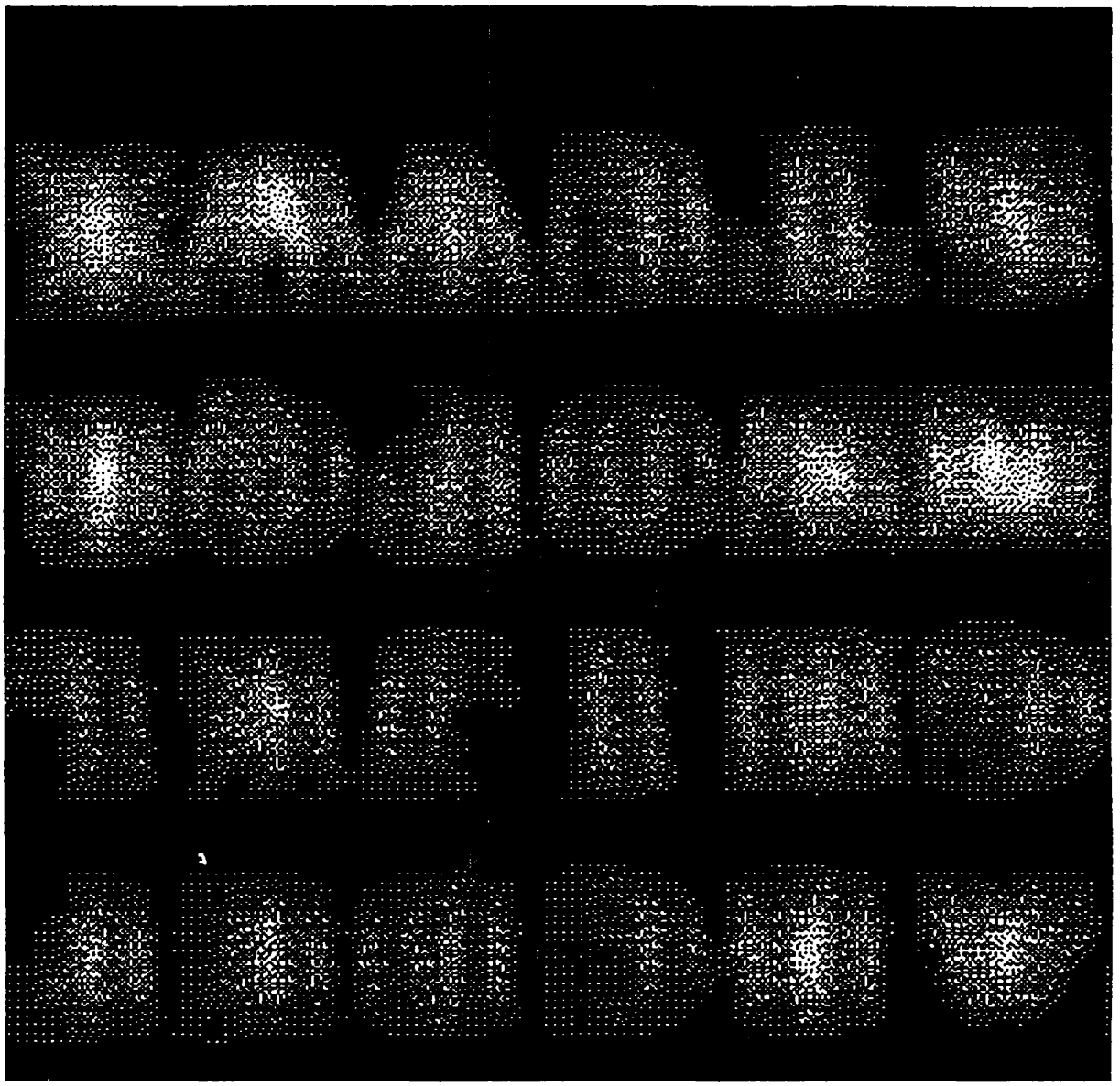


FIGURE 7.20 THINNED TEST SET CORRELATED WITH TEMPLATE B

VIII. DISCUSSION

Those tasks which seem trivial to a human seem to be impossible for a computer. With all their computing power and megabytes of memory, the digital computer is in reality a very stupid machine. It can only perform those tasks which a person allows it to perform (by writing the proper program).

8.1 CONCLUSION

This thesis examined the use of a two pass correlation technique for character recognition. The first pass would produce a series of correlations which showed which templates matched best with the input. During the second pass, the energy of both input and template would be set equal to one another. Correlation would then produce a maximum peak for inputs with the same shape.

Testing with a highly constrained font set showed that the algorithm would work for isolated letters as well as for concatenated letters if the template set was derived from the same font style as that tested.

Testing with a template set different from that of the input test set showed that the algorithm was very sensitive to size variation. Thinning letters to match the template pixel thickness seemed to aid the process. However, recognition was still not possible. It appears as if the

height of the template must also be adjusted. Thus, energy, thickness, and height must all be considered.

8.2 RECOMMENDATIONS

It is apparent that much more research must be performed before a recognition is developed which is truely font invariant. For the algorithm presented, three recommendations are made for future research:

1. Implement in software the function which adjusts the height of a template to match that of the input. The program LINES could be easily modified to calculate the average height of a line of print.

2. Testing should continue, but for a greater number of characters. These tests should contain both the upper and lower case characters. (Since the *height was not* adjusted in this thesis, capital letters did not correlate well with lower case letters.)

3. Try correlation with the Fourier transform clipped after the third harmonic. Most of the information content in a scene is contained in the low order harmonics. Serifs and jagged edges contribute principlely to the high order harmonics.

Bibliography

1. Kabrisky, Matthew. A Proposed Model for Visual Information Processing in the Human Brain. Urbana, Ill.: University of Illinois Press, 1966.

2. Simmons, 2D Lt Robin A. Machine Segmentation of Unformatted Characters. MS thesis. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1981.

APPENDIX A

This part of the thesis contains the software developed for the pre-processing and recognition stages. Each program is listed in alphabetical order with the exception of DIRECT and INVERSE. They were supplied as executable programs without source code.

From documentation in the Signal Processing Lab, DIRECT is a Fortran program written by R. W. Schafer on 30 June 78. The purpose of the program is to compute a 2-D forward FFT on disk using the Eklundh algorithm. The maximum input array size is 512x512. Input must be in binary format and contain complex data with the imaginary part equal to zero.

INVERSE is another Fortran program written by Schafer on 15 Feb 77. This program computed the 2-D inverse FFT on disk using the Eklundh algorithm. The inverse transformed array is normalized by division with N**2.

Two notes are required before this thesis concludes. The first is that the program MEND.FR was written using integer values for output because the OCTEK only accepted integer values.

The final note is that line thinning was examined in the later part of this thesis. The program VTHIN.FR was written to perform this task. The program RLNX.FR must be used first to determine the maximum number of pixels which can be deleted from the character.

```
C*****************************************************************C
C                                                                 C
C                                                                 C
C  PROGRAM: CLEAN.FR          LANGUAGE: FORTRAN V                 C
C                                                                 C
C  WRITTEN: 15 DEC 84         SYSTEM: ECLIPSE                     C
C                                                                 C
C  PURPOSE: THIS PROGRAM WILL CLEAN A GREYSCALE VIDEO FILE        C
C           AND PRODUCE A BINARY VIDEO FILE.                      C
C                                                                 C
C                                                                 C
C                                                                 C
C*****************************************************************C
C                                                                 C
C                                                                 C
C                                                                 C
      PARAMETER IBLK=0
      PARAMETER IWHT=15
      IMPLICIT INTEGER(C,D,E)
      IMPLICIT DOUBLE PRECISION(A,B,T)
      INTEGER IARRAY(256),IBUF(300)
      INTEGER ANDX,BNDX,MNDX,MID,INAME(20)
      DOUBLE PRECISION AHIST(18),AMAX(18),AMIN(18),T
C                                                                 C
C                                                                 C
C.....................................................................C
C                                                   ;             C
C                                                   ;             C
      ACCEPT"ENTER FILENAME: "                      ;             C
      READ(11,199) INAME(1)                         ; INPUT FILENAME C
 199  FORMAT(S40)                                   ;             C
C                                                   ;             C
C...................................................;.............C
C                                                   ;             C
C                                                   ;             C
      CALL PICFMT(IBUF,INAME,IFLG)                  ;             C
      CALL GFMT(IBUF,NROWS,NCOLS,NBITS,IMODE)       ;  PICBUF INPUT C
                                                    ;     BLOCK   C
      CALL MAKB(IBUF)                               ;             C
      CALL PICIN(IBUF,INAME,IFLG)                   ;             C
C                                                   ;             C
C...................................................;.............C
C                                                   ;             C
C                                                   ;             C
      TYPE"CLEAN WORKING!"                          ;     I/O     C
C                                                   ;             C
C...................................................;.............C
C                                                   ;             C
      DO 2 I=1,18                                   ;             C
           AHIST(I)=0.0D0                           ;  INITIALIZE C
           AMAX(I)=0.0D0                            ;  ALL VALUES C
           AMIN(I)=9.0D9                            ;             C
 2    CONTINUE                                      ;             C
C                                                   ;             C
C...................................................;.............C
```

```
C.................................................................C
C                                                 ;               C
      DO 4 I=1,240                                ; GET EACH PIXEL C
           CALL GROW(IBUF,I,1,256,IARRAY)         ;     VALUE      C
           DO 6 J=1,256                           ;  (240 ROWS,    C
                K=IARRAY(J)+2                     ;  256 COLUMNS)  C
                AHIST(K)=AHIST(K)+1.0D0           ; CALCULATE THE  C
 6         CONTINUE                               ;   HISTOGRAM    C
 4    CONTINUE                                    ;               C
C                                                 ;               C
C.................................................;...............C
C                                                     ;           C
C                                                     ; DETERMINE C
      K=0                                             ; IF VIDEO  C
      DO 8 I=1,18                                     ;    IS     C
           IF(AHIST(I).EQ.0)K=K+1                     ;           C
 8    CONTINUE                                        ;           C
                                                      ;           C
      IF(K.NE.17)GOTO 89                              ;           C
      TYPE"THERE IS ONLY ONE LEVEL OF BRIGHTNESS"     ;   BLANK   C
      GOTO 99                                         ;           C
                                                      ;    OR     C
 89   IF(K.NE.16)GOTO 90                              ;           C
      TYPE"THERE ARE ONLY TWO LEVELS OF BRIGHTNESS";    BINARY    C
      GOTO 99                                         ;           C
C                                                     ;           C
C.....................................................;...........C
C                                                     ;           C
C                                                     ;           C
 90   C=1                                             ;           C
      DO 10 I=2,17                                    ;           C
                                                      ;           C
            IF((AHIST(I).GT.AHIST(I-1))               ; DETERMINE C
     /                 .AND.                          ;    THE    C
     /      (AHIST(I).GE.AHIST(I+1)))GOTO 91          ;   PEAKS   C
                                                      ;           C
            GOTO 10                                   ;           C
                                                      ;           C
 91         AMAX(C)=AHIST(I)                          ;           C
            C=C+1                                     ;           C
                                                      ;           C
 10   CONTINUE                                        ;           C
C                                                     ;           C
C.....................................................;...........C
C                                                     ;           C
C                                                     ;           C
C                                                     ;           C
      DO 451 I=1,18                                   ;           C
      WRITE FREE(10)"AHIST",I,AHIST(I)                ;    I/O    C
 451  CONTINUE                                        ;           C
C                                                     ;           C
C                                                     ;           C
C.....................................................;...........C
```

```
C.....C
C                                                                 ;               C
      D=C-2                                                       ;               C
                                                                  ;               C
      DO 12 I=1,D                                                 ;               C
            J=C-I                                                 ;    BUBBLE     C
            DO 14 K=1,J                                           ;     SORT      C
                  IF(AMAX(K).GE.AMAX(K+1))GOTO 14                 ;     THE       C
                  T=AMAX(K)                                       ;     PEAK      C
                  AMAX(K)=AMAX(K+1)                               ;    VALUES     C
                  AMAX(K+1)=T                                     ;               C
 14           CONTINUE                                            ;               C
 12   CONTINUE                                                    ;               C
C.....C
C                                                                 ;               C
        DO 16 I=2,17                                              ; DETERMINE     C
              IF(AHIST(I).NE.AMAX(1))GOTO 16                      ; GREYLEVEL     C
              ANDX=I                                              ; CORRESPONDING C
              GOTO 991                                            ; TO THE FIRST  C
 16     CONTINUE                                                  ;  PEAK VALUE   C
C.....C
C                                                                 ;               C
 991    N=D+1                                                     ;               C
                                                                  ;               C
        J=2                                                       ;   DETERMINE   C
        DO 18 I=2,17                                              ;   GREYLEVEL   C
              IF(AHIST(I).NE.AMAX(J))GOTO 18                      ; CORRESPONDING C
              N=N-1                                               ; TO THE SECOND C
              BNDX=I                                              ;   PEAK VALUE  C
              IF((I.GT.7).AND.(N.GT.0))GOTO 92                    ;               C
              GOTO 992                                            ;               C
 92           J=J+1                                              ;               C
 18     CONTINUE                                                  ;               C
C.....C
C                                                                 ;               C
 992    IF(BNDX.LT.ANDX)GOTO 993                                  ;               C
        E=ANDX                                                    ;  ENSURE THAT  C
        ANDX=BNDX                                                 ;  ANDX > BNDX  C
        BNDX=E                                                    ;               C
C.....C
C                                                                 ;               C
 993    C=1                                                       ;               C
        DO 20 I=BNDX,ANDX                                         ;               C
                                                                  ;               C
              IF((AHIST(I).LE.AHIST(I-1))                         ;               C
     /                        .AND.                               ;   DETERMINE   C
     /        (AHIST(I).LT.AHIST(I+1)))GOTO 93                   ;      THE      C
                                                                  ;    VALLEYS    C
              GOTO 20                                             ;               C
                                                                  ;               C
 93           AMIN(C)=AHIST(I)                                   ;               C
              C=C+1                                               ;               C
                                                                  ;               C
 20     CONTINUE                                                  ;               C
C.....C
C
```

```
C.....C
C
C                                                    ;                 C
C                                                    ;  DETERMINE IF   C
      IF(C.EQ.2)GOTO 994                             ; THERE ARE ONLY  C
C                                                    ;    ONLY TWO     C
C                                                    ;     VALLEYS     C
C                                                    ;                 C
C.....C                                              ;.................C
C                                                    ;                 C
      D=C-2                                          ;                 C
                                                     ;                 C
      DO 22 I=1,D                                    ;                 C
            J=C-I                                    ;   BUBBLE        C
            DO 24 K=1,J                              ;    SORT         C
                  IF(AMIN(K).LE.AMIN(K+1))GOTO 24    ;    THE          C
                  T=AMIN(K)                          ;   VALLEYS       C
                  AMIN(K)=AMIN(K+1)                  ;                 C
                  AMIN(K+1)=T                        ;                 C
 24           CONTINUE                               ;                 C
 22   CONTINUE                                       ;                 C
C.....C                                              ;.................C
C                                                    ;                 C
 994  DO 26 I=2,17                                   ;    DETERMINE    C
            IF(AHIST(I).NE.AMIN(1))GOTO 26           ;    GREYLEVEL    C
            MNDX=I                                   ;  CORRESPONDING  C
            GOTO 995                                 ;   TO THE FIRST  C
 26   CONTINUE                                       ;      VALLEY     C
C.....C                                              ;.................C
C                                                    ;                 C
C                                                    ;DETERMINEC
 995  MNDX=MNDX-2                                    ;    IF           C
      IF(MNDX.GT.10)TYPE"PICTURE CANNOT BE CLEANED"  ;   VIDEO        C
      IF(MNDX.GT.10)GOTO 9999                        ;  CAN BE        C
C                                                    ;PROCESSEDC
C                                                    ;                 C
C.....C                                              ;.................C
C                                                    ;                 C
C                                                    ;    CALCULATE    C
C                                                    ;    MIDPOINT     C
      MID=INT((ANDX+BNDX)/2)                         ;   BETWEEN THE   C
C                                                    ;    TWO PEAKS    C
C                                                    ;                 C
C.....C                                              ;.................C
C                                                    ;                 C
C                                                    ;                 C
      WRITE FREE(10)"ANDX=",ANDX-2                   ;                 C
      WRITE FREE(10)"BNDX=",BNDX-2                   ;       I/O       C
      WRITE FREE(10)"MNDX=",MNDX                     ;                 C
C                                                    ;                 C
C.....C                                              ;.................C
```

```
C.....................................................................C
C                                                    ;                C
C                                                    ;                C
      DO 55 IROW=1,240                               ; GET A ROW      C
            CALL  GROW(IBUF,IROW,1,256,IARRAY)       ; OF PIXELS      C
                                                     ;                C
                                                     ; THRESHOLD      C
            NT=0                                     ;POINT VARIES    C
            DO 60 I=MNDX,10                          ;BETWEEN THE     C
                  NT=NT+1                            ;FIRST VALLEY    C
                                                     ;   AND 10       C
                  DO 65 J=1,256                      ;                C
                        IF(IARRAY(J).LE.I)GOTO 95    ; CLIP PIXELS    C
                        IARRAY(J)=IARRAY(J)+IWHT     ;    BELOW       C
                        GOTO 65                      ; THRESHOLD      C
 95                     IARRAY(J)=IARRAY(J)+IBLK     ; TO 0  AND      C
 65                 CONTINUE                         ; SUM PIXEL      C
                                                     ;   VALUES       C
 60            CONTINUE                              ;                C
                                                     ;                C
                                                     ;   AVERAGE      C
            DO 70 J=1,256                            ; THRESHOLDED    C
                  IARRAY(J)=INT(IARRAY(J)/NT)        ;    VALUE       C
                  IF(IARRAY(J).LE.MID)GOTO 96        ; CLIP VALUES    C
                  IARRAY(J)=IWHT                     ;    BELOW       C
                  GOTO 70                            ;  MIDPOINT      C
 96               IARRAY(J)=IBLK                     ;   TO  0        C
 70         CONTINUE                                 ;                C
                                                     ;                C
              CALL PROW(IBUF,IROW,1,256,IARRAY)      ;                C
                                                     ;                C
 55      CONTINUE                                    ;                C
C.....................................................;................C
C                                                    ;                C
C                                                    ;                C
 99      DO 75 I=1,256                               ;                C
               IARRAY(I)=15                          ;   BLANK OUT    C
 75      CONTINUE                                    ;  LAST 16 ROWS  C
                                                     ;  OF THE VIDEO  C
         DO 85 I=241,256                             ;    FILE        C
               CALL PROW(IBUF,I,1,256,IARRAY)        ;                C
 85      CONTINUE                                    ;                C
C                                                    ;                C
C.....................................................;................C
C                                                    ;                C
         CALL PICOUT(IBUF,INAME,1)                   ;                C
                                                     ; PICBUF OUTPUT  C
         TYPE"<7>FINISHED: PLEASE CALL SPOTS"        ;    BLOCK       C
                                                     ;                C
 9999    CALL RELB(IBUF)                             ;                C
C.....................................................;................C
C
         STOP
         END
```

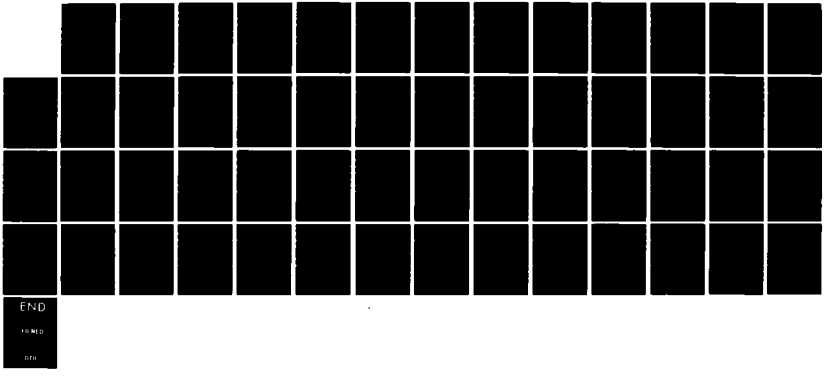
```
C*****************************************************************C
C                                                                 C
C                                                                 C
C    ORIGINAL PROGRAM: CPTOVD.FR  BY R. WELLS                     C
C                                                                 C
C    PROGRAM: CTV.FR              LANGUAGE: FORTRAN V             C
C                                                                 C
C    WRITTEN: 15 DEC 84           SYSTEM: ECLIPSE                 C
C                                                                 C
C    PURPOSE: THIS PROGRAM FINDS THE MAXIMUM OF A 256X256         C
C             COMPLEX FILE AND GIVES THE LOCATION OF THE          C
C             MAXIMUM. ALSO, THE   PROGRAM CONVERTS THE FILE      C
C             TO PACKED VIDEO FORMAT BY  PERFORMING A LINEAR      C
C             SCALING TO THE RANGE OF 0 - 15  AND TRUNCATING.     C
C                                                                 C
C                                                                 C
C                                                                 C
C*****************************************************************C
C                                                                 C
C                                                                 C
      INTEGER IFLNM(7),OFLNM(7)
      INTEGER TEMP(512),OUT(2048),IMAX,JMAX,MAXCOL,MAXROW
      COMPLEX IN(2048)
      REAL RMAG,MAXVAL,MINVAL,RANGE
C                                                                 C
C.....................................................................C
C                                                     ;             C
      ACCEPT "ENTER INPUT FILENAME: "                 ;             C
      READ (11,1995) IFLNM(1)                         ;     ENTER   C
 1995 FORMAT (S40)                                    ; INPUT AND OUTPUT C
                                                      ;   FILENAMES C
      ACCEPT "ENTER OUTPUT FILENAME: "                ;             C
      READ (11,1995) OFLNM(1)                         ;             C
C.....................................................;.............C
C                                                     ;             C
      OPEN 1,IFLNM                                    ;             C
                                                      ;             C
      CALL DFILW (OFLNM,IER)                          ;     OPEN    C
      IF(IER.NE.1)GOTO 90                             ; INPUT AND OUTPUT C
 90   CALL CFILW(OFLNM,2,KER)                         ;     FILES   C
      CALL CHECK(KER)                                 ;             C
      OPEN 2,OFLNM,ATT="OR",ERR=100                   ;             C
C                                                     ;             C
C.....................................................;.............C
C                                                     ;             C
 100  MAXVAL=0.0                                      ;             C
      MINVAL=0.0                                      ;             C
                                                      ;             C
      DO 3 I=0,31                                     ;             C
                                                      ;             C
         CALL RDBLK(1,32*I,IN,32,IER)                 ;             C
         CALL CHECK(IER)                              ;             C
                                                      ;             C
         DO 3 J=1,2048                                ;             C
                                                      ;  DETERMINE  C
```

```
            RMAG=CABS(IN(J))                          ;        MAXVAL        C
                                                      ;          AND         C
            IF(RMAG.LE.MAXVAL)GO TO 3                 ;        MINVAL        C
               MAXVAL=RMAG                            ;                      C
               IMAX=I                                 ;                      C
               JMAX=J                                 ;                      C
                                                      ;                      C
    3 CONTINUE                                        ;                      C
C                                                     ;                      C
C.....................................................;......................C
C                                                     ;                      C
      RANGE=MAXVAL-MINVAL                             ;     CALCULATE        C
                                                      ;       RANGE          C
      MAXCOL=MOD(JMAX-1,256)+1                        ;       MAXCOL         C
      MAXROW=8*IMAX+INT((JMAX-1)/256)+1               ;       MAXROW         C
C                                                     ;                      C
C.....................................................;......................C
C                                                     ;                      C
      TYPE"MAX COL = ",MAXCOL                         ;                      C
      TYPE"MAX ROW = ",MAXROW                         ;                      C
                                                      ;         I/O          C
      TYPE"MAX VAL = ",MAXVAL                         ;                      C
      TYPE"MIN VAL = ",MINVAL                         ;                      C
C                                                     ;                      C
C.....................................................;......................C
C                                                     ;                      C
      DO 5 I=0,31                                     ;                      C
                                                      ;                      C
         CALL RDBLK(1,32*I,IN,32,IER)                 ;                      C
         CALL CHECK(IER)                              ;                      C
                                                      ;                      C
         DO 4 J=1,2048                                ;   SCALE VALUES       C
            OUT(J)=ANINT(15.0*(CABS(IN(J))            ;                      C
     /   -MINVAL)/RANGE)                              ;         TO           C
    4    CONTINUE                                     ;       0 - 15         C
                                                      ;                      C
         CALL REPACK(512,OUT,TEMP)                    ;                      C
         CALL WRBLK(2,2*I,TEMP,2,IER)                 ;                      C
         CALL CHECK(IER)                              ;                      C
                                                      ;                      C
    5 CONTINUE                                        ;                      C
C                                                     ;                      C
C.....................................................;......................C
C                                                                            C
      CALL RESET
      STOP"<7><7><7><7>CPTOVD"
      END
```

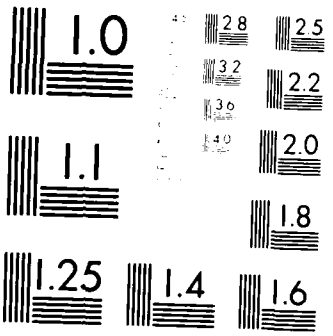
```
C*****************************************************************C
C                                                                 C
C                                                                 C
C                                                                 C
C   ORIGINAL PROGRAM: CPTOVD.FR  BY R. WELLS                      C
C                                                                 C
C   PROGRAM: CTV5.FR             LANGUAGE: FORTRAN V              C
C                                                                 C
C   WRITTEN: 15 DEC 84           SYSTEM: ECLIPSE                  C
C                                                                 C
C   PURPOSE:THIS PROGRAM FINDS THE MAXIMUM AND MINIMUM OF A       C
C           256X256 COMPLEX FILE AND GIVES THE LOCATION OF THE    C
C           MAXIMUM. THE PROGRAM CONVERTS THE FILE TO PACKED      C
C           VIDEO FORMAT BY PERFORMING A LINEAR SCALING OF        C
C           0.0 - 50000.0  TO  RANGE OF 0 - 15.                   C
C                                                                 C
C                                                                 C
C                                                                 C
C*****************************************************************C
C                                                                 C
C                                                                 C
      INTEGER IFLNM(7),OFLNM(7)
      INTEGER TEMP(512),OUT(2048),IMAX,JMAX,MAXCOL,MAXROW
      COMPLEX IN(2048)
      REAL RMAG,MAXVAL,MINVAL,RANGE
C                                                                 C
C.....................................................;...........C
C                                                     ;           C
      ACCEPT "ENTER INPUT FILENAME: "                 ;           C
      READ (11,1995) IFLNM(1)                         ;   ENTER   C
 1995 FORMAT (S40)                                    ; INPUT AND C
                                                      ;  OUTPUT   C
      ACCEPT "ENTER OUTPUT FILENAME: "                ; FILENAMES C
      READ (11,1995) OFLNM(1)                         ;           C
C.....................................................;...........C
C                                                     ;           C
C                                                     ;           C
        MAXVAL=50000.0                                ; SET MAXVAL C
C                                                     ;           C
C.....................................................;...........C
C                                                     ;           C
        OPEN 1,IFLNM                                  ;           C
                                                      ;           C
        CALL DFILW (OFLNM,IER)                        ;   OPEN    C
        IF(IER.NE.1)GOTO 90                           ; INPUT AND C
 90     CALL CFILW(OFLNM,2,KER)                       ;  OUTPUT   C
        CALL CHECK(KER)                               ;  FILES    C
        OPEN 2,OFLNM,ATT="OR",ERR=100                 ;           C
C                                                     ;           C
C.....................................................;...........C
```

AD-A152 006 CHARACTER RECOGNITION USING CORRELATION TECHNIQUES(U) 2/2
AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL
OF ENGINEERING L SHUM DEC 84 AFIT/GE/ENG/84D-59

UNCLASSIFIED F/G 5/7 NL



END



```
C.....;.....C
C     ;     C
C     ;     C
 100  MINVAL=0.0                                    ;   INITIALIZE    C
C     ;     MINVAL      C
C     ;     C
C.....;.....C
C     ;     C
      RANGE=50000.0                                 ;   INITIALIZE    C
C     ;     RANGE       C
C.....;.....C
C     ;     C
      TYPE"MAX VAL = ",MAXVAL                       ;     C
      TYPE"MIN VAL = ",MINVAL                       ;     I/O         C
C     ;     C
C.....;.....C
C     ;     C
      DO 5 I=0,31                                   ;     C
                                                    ;     C
         CALL RDBLK(1,32*I,IN,32,IER)               ;     C
         CALL CHECK(IER)                            ;     C
                                                    ;     C
         DO 4 J=1,2048                              ;  SCALE VALUES   C
            OUT(J)=ANINT(15.0*(CABS(IN(J))          ;     C
     /        -MINVAL)/RANGE)                       ;       TO        C
    4     CONTINUE                                  ;     0 - 15      C
                                                    ;     C
         CALL REPACK(512,OUT,TEMP)                  ;     C
         CALL WRBLK(2,2*I,TEMP,2,IER)               ;     C
         CALL CHECK(IER)                            ;     C
                                                    ;     C
    5 CONTINUE                                      ;     C
C     ;     C
C.....;.....C
C     C

      CALL RESET
      STOP"<7><7><7><7>CPTOVD"
      END
```

```
C*****************************************************************C
C                                                                 C
C                                                                 C
C   PROGRAM: FCHAR.FR            LANGUAGE: FORTRAN V              C
C                                                                 C
C   WRITTEN: 15 DEC 84           SYSTEM: ECLIPSE                  C
C                                                                 C
C   PURPOSE: THIS PROGRAM WILL SEPARATE EACH CHARACTER FROM       C
C            THE NEXT IF THERE EXISTS A SPACE BETWEEN THEM.       C
C                                                                 C
C*****************************************************************C
C                                                                 C
      IMPLICIT INTEGER(C,D,E)
      INTEGER IARRAY(512),IBUF(300),INAME(20)
      INTEGER DCNTR(20),DNTR(20),ECNTR(20),ENTR(20)
      INTEGER ICSTK(100)
C                                                                 C
C.....................................................;...........C
C                                                     ;           C
        ACCEPT"ENTER FILENAME: "                      ;           C
        READ(11,199)INAME(1)                          ;   INPUT   C
 199    FORMAT(S40)                                   ;  FILENAME C
C                                                     ;           C
C.....................................................;...........C
C                                                     ;           C
      CALL PICFMT(IBUF,INAME,IFLG)                    ;           C
      CALL GFMT(IBUF,NROWS,NCOLS,NBITS,IMODE)         ; PICBUF INPUT C
                                                      ;   BLOCK   C
      CALL MAKB(IBUF)                                 ;           C
      CALL PICIN(IBUF,INAME,IFLG)                     ;           C
C                                                     ;           C
C.....................................................;...........C
C                                                     ;           C
        OPEN 1,"LDAT.TX"                              ;           C
        READ BINARY(1)DCNTR,ECNTR,C,LIN               ;  READ LINE C
                                                      ;  AND WORD  C
        OPEN 2,"WDAT.TX"                              ;  POSITIONS C
        READ BINARY(2)DNTR,ENTR,D,IWORD               ;           C
C                                                     ;           C
C.....................................................;...........C
C                                                     ;           C
      TYPE"LINE:",LIN,"     WORD:",IWORD              ;           C
                                                      ;    I/O    C
      TYPE"FCHAR WORKING"                             ;           C
C.....................................................;...........C
C                                                     ;           C
        ISCOL=DNTR(IWORD)                             ;  CALCULATE C
 186    ISCOL=ISCOL+1                                 ;STARTING COLUMN C
C                                                     ;           C
C.....................................................;...........C
C                                                     ;           C
        CALL PICIN(IBUF,INAME,IFLG)                   ;GET VIDEO FILE C
C.....................................................;...........C
```

```
C.....C
        COL=ISCOL                                ;                    C
        E=1                                      ;INITIALIZE COUNTERS C
        ICSTK(E)=COL                             ;      TO STARTING   C
                                                 ;        POSTIONS    C
        DC=DCNTR(LIN)                            ;                    C
        EC=ECNTR(LIN)                            ;                    C
C.....C.....C
C                                                ;                    C
        TYPE"START COL= ",ICSTK(E)               ;                    C
                                                 ;       I/O          C
        TYPE"START ROW= ",DC-1                   ;                    C
C                                                ;                    C
C.....C.....C
C                                                ;                    C
        DC=DC-1                                  ;                    C
        IF(DC.LT.1)DC=1                          ; CALCULATE START    C
        EC=EC+1                                  ; AND END OF ROW     C
        IF(EC.GT.256)EC=256                      ;                    C
C                                                ;                    C
C.....C.....C
C                                                ;                    C
      IST=DC                                     ;                    C
      IEN=EC                                     ;DETERMINE IF PIXEL  C
 1    DO 10 IROW=DC,EC                           ;     IS BLACK       C
           CALL GPNT(IBUF,IROW,COL,IVAL)         ;                    C
           IF(IVAL.EQ.0)GOTO 100                 ;IF WHITE-->         C
           ICSTK(E)=COL                          ;CONTINUE DOWN COLUMN C
           E=E+1                                 ;                    C
 10   CONTINUE                                   ;IF BLACK-->         C
                                                 ;FIND ANOTHER PATH   C
        GOTO 500                                 ;                    C
C                                                ;                    C
C.....C.....C
C                                                ;                    C
 100    DC=IROW                                  ;                    C
        ITEMP=COL                                ;                    C
        IR=IROW                                  ;  FIND CLOSEST      C
        IFR=ITEMP                                ; BLACK PIXEL TO     C
 200    IFR=IFR+1                                ;  TO THE RIGHT      C
        CALL GPNT(IBUF,IR,IFR,IVAL)              ;                    C
        IF(IVAL.EQ.0)GOTO 200                    ;                    C
C                                                ;                    C
C.....C.....C
C                                                ;                    C
C                                                ;                    C
        IB=ITEMP                                 ;  FIND CLOSEST      C
 300    IB=IB-1                                  ; BLACK PIXEL TO     C
        CALL GPNT(IBUF,IR,IB,IVAL)               ;  TO THE LEFT       C
        IF(IVAL.EQ.0)GOTO 300                    ;                    C
C                                                ;                    C
C.....C.....C
```

```
C.....;.....C
C     ;     C
      MR=IR-1
      DO 964 ITT=IB,ITEMP
           CALL GPNT(IBUF,MR,ITT,IVAL)
           IF(IVAL.EQ.0)GOTO 186
964   CONTINUE
C     ;     C
C.....;.....C
C     ;     C
        IFD=IABS(IFR-ITEMP)
        IBD=IABS(ITEMP-IB)
        COL=IFR
        IF(IFD.GT.IBD)COL=IB
        ICSTK(E)=COL
C     ;     C
C.....;.....C
C     ;     C
      GOTO 1
C     ;     C
C.....;.....C
C     ;     C
 500  DO 8 I=1,256
      IARRAY(I)=15
 8    CONTINUE
      J=1
      DO 20 I=IST,IEN
         ILEN=256-ICSTK(J)+1
         CALL PROW(IBUF,I,ICSTK(J),ILEN,IARRAY)
         J=J+1
 20   CONTINUE
C     ;     C
C.....;.....C
C     ;     C
      CTOT=0
      ILEN=IABS(ENTR(IWORD)-DNTR(IWORD))
      DO 30 I=IST;IEN
        CALL GROW(IBUF,I,DNTR(IWORD),ILEN,IARRAY)
        DO 40 J=1,ILEN-1
              IF(IARRAY(J).EQ.0)CTOT=CTOT+1
 40     CONTINUE
 30   CONTINUE
C     ;     C
C.....;.....C
C     ;     C
C     ;     C
        IF(CTOT.LT.30)GOTO 186
C     ;     C
C.....;.....C
C     ;     C
        OPEN 3,"CDAT.TX"
        WRITE BINARY(3)ICSTK
C     ;     C
C.....;.....C
```

```
C.....................................................;...................C
C                                                     ;                   C
 9999    CALL PICOUT(IBUF,INAME,IFLG)                 ; PICBUF OUTPUT     C
                                                      ;     BLOCK         C
         TYPE"<7>FINISHED"                            ;      AND          C
                                                      ;      I/O          C
         CALL RELB(IBUF)                              ;                   C
C                                                     ;                   C
C.....................................................;...................C
C                                                                         C
         STOP
         END
```

```
C*****************************************************************C
C                                                                 C
C                                                                 C
C    PROGRAM: FLINE.FR          LANGUAGE: FORTRAN V               C
C                                                                 C
C    WRITTEN: 15 DEC 84         SYSTEM: ECLIPSE                   C
C                                                                 C
C    PURPOSE: THIS PROGRAM WILL COPY ONE LINE OF PRINTED TEXT     C
C             (WHICH THE USER HAS CHOSEN) AND BLANK OUT THE       C
C                  OTHER LINES.                                   C
C                                                                 C
C                                                                 C
C*****************************************************************C
C                                                                 C
C                                                                 C
      IMPLICIT INTEGER(C,D,E)
      INTEGER IARRAY(512),IBUF(300),IBUF2(300),INAME(20)
      INTEGER DCNTR(20),ECNTR(20)
C                                                                 C
C.....................................................;...........C
C                                                     ;           C
      ACCEPT"ENTER FILENAME: "                        ;           C
      READ(11,199)INAME(1)                            ;   INPUT   C
 199  FORMAT(S40)                                     ; FILENAME  C
C                                                     ;           C
C.....................................................;...........C
C                                                     ;           C
C                                                     ;           C
      OPEN 1,"LDAT.TX"                                ;  READ LINE C
      READ BINARY(1)DCNTR,ECNTR,C                     ; POSITIONS C
C                                                     ;  LDAT.TX  C
C                                                     ;           C
C.....................................................;...........C
C                                                     ;           C
 299  ACCEPT"WHICH LINE TO PROCESS: ",LIN             ;   INPUT   C
      IF((LIN.GT.C-1).OR.(LIN.LE.0))GOTO 299          ; LINE NUMBER C
C                                                     ;           C
C.....................................................;...........C
C                                                     ;           C
      CALL PICFMT(IBUF,INAME,IFLG)                    ;           C
      CALL GFMT(IBUF,NROWS,NCOLS,NBITS,IMODE)         ;           C
                                                      ;           C
      CALL MAKB(IBUF)                                 ; PICBUF INPUT C
      CALL PICIN(IBUF,INAME,IFLG)                     ;   BLOCK   C
                                                      ;           C
      CALL PFMT(IBUF2,NROWS,NCOLS,NBITS,IMODE)        ;           C
      CALL MAKB(IBUF2)                                ;           C
C                                                     ;           C
C.....................................................;...........C
C                                                     ;           C
      TYPE"LINE:",LIN                                 ;           C
                                                      ;   I/O     C
      TYPE"FLINE WORKING"                             ;           C
C                                                     ;           C
C.....................................................;...........C
```

```
C.....................................................................C
C                                                                     C
      DO 5 I=1,256                                                    C
           IARRAY(I)=15                                               C
 5    CONTINUE                                                        C
                                                                      C
      DO 10 I=1,256                                                   C
            CALL PROW(IBUF2,I,1,256,IARRAY)                           C
 10   CONTINUE                                                        C
C                                                                     C
C.....................................................................C
C                                                                     C
      DO 15 I=DCNTR(LIN),ECNTR(LIN)                                   C
           CALL GROW(IBUF,I,1,256,IARRAY)                             C
           CALL PROW(IBUF2,I,1,256,IARRAY)                            C
 15   CONTINUE                                                        C
C                                                                     C
C.....................................................................C
C                                                                     C
      CLOSE 1                                                         C
      OPEN 1,"LDAT.TX"                                                C
      WRITE BINARY(1)DCNTR,ECNTR,C,LIN                                C
C                                                                     C
C                                                                     C
C.....................................................................C
C                                                                     C
      CALL PICOUT(IBUF2,INAME,IFLG)                                   C
      CALL RELB(IBUF)                                                 C
      CALL RELB(IBUF2)                                                C
C                                                                     C
C.....................................................................C
C                                                                     C
        STOP"<7> "
        END
```

```
C*****************************************************************C
C                                                                 C
C                                                                 C
C     PROGRAM: FPEAK.FR                 LANGUAGE: FORTRAN V       C
C                                                                 C
C     WRITTEN: 15 DEC 84                SYSTEM: ECLIPSE           C
C                                                                 C
C     PURPOSE: THIS PROGRAM WILL FIND THE MAXIMUM CORRELATION     C
C              PEAK VALUE IN A VIDEO FILE.                        C
C                                                                 C
C                                                                 C
C*****************************************************************C
C                                                                 C
C                                                                 C
      INTEGER TEMP(512),IN(2048)
      INTEGER IFLNM(7),IMAX(10),JMAX(10),MAXCOL(10),MAXROW(10)
C                                                                 C
C.....................................................................C
C                                                                 C
      ACCEPT "ENTER INPUT FILENAME: "                             C
      READ (11,1995) IFLNM(1)                    ;   ENTER INPUT  C
1995  FORMAT (S40)                               ;    FILENAME    C
C.....................................................;.............C
      OPEN 1,IFLNM                               ; I/O FILE CONTROL C
C.....................................................;.............C
C                                                   ;               C
      MAXVAL=0                                      ;               C
                                                    ;               C
      DO 5 I=0,31                                   ;               C
           CALL RDBLK(1,2*I,TEMP,2,IER)             ;               C
           CALL CHECK(IER)                          ;               C
           CALL UNPACK(512,TEMP,IN)                 ;    DETERMINE  C
                                                    ;     MAXIMUM   C
           DO 10 J=1,2048                           ;    PEAK VALUE C
                IF(IN(J).LE.MAXVAL)GOTO 10          ;               C
                 MAXVAL=IN(J)                       ;               C
10         CONTINUE                                 ;               C
                                                    ;               C
5     CONTINUE                                      ;               C
C.....................................................;.............C
C                                                   ;               C
      K=1                                           ;               C
      DO 15 I=0,31                                  ;               C
            CALL RDBLK(1,2*I,TEMP,2,IER)            ;               C
            CALL CHECK(IER)                         ;               C
            CALL UNPACK(512,TEMP,IN)                ;               C
                                                    ;               C
            DO 20 J=1,2048                          ;    DETERMINE  C
                  IF(IN(J).NE.MAXVAL)GOTO 20        ;    LOCATION   C
                  IMAX(K)=I                         ;        OF     C
                  JMAX(K)=J                         ;   PEAK VALUES C
                  K=K+1                             ;               C
20          CONTINUE                                ;               C
                                                    ;               C
15    CONTINUE                                      ;               C
```

```
      DO   30 I=1,K-1                                                ;          C
           MAXCOL(I)=MOD(JMAX(I)-1,256)+1                            ;          C
           MAXROW(I)=8*IMAX(I)+INT((JMAX(I)-1)/256)+1                ;          C
 30   CONTINUE                                                       ;          C
C                                                                     ;          C
C.....................................................................;.........C
C                                                    ;                          C
        WRITE FREE(10)"MAXVAL IS: ",MAXVAL           ;                          C
        TYPE"AT POSITIONS"                           ;                          C
                                                     ;                          C
        DO 40 I=1,K-1                                ;          I/O             C
        WRITE FREE(10)"ROW: ",MAXROW(I)              ;                          C
        WRITE FREE(10)"COLUMN: ",MAXCOL(I)           ;                          C
 40     CONTINUE                                     ;                          C
C.....................................................;..........................C
C                                                                                 C
      CALL RESET
      STOP "<7> FINISHED"
      END
```

```
C*****************************************************************C
C                                                                 C
C                                                                 C
C    PROGRAM: FWORD.FR          LANGUAGE: FORTRAN V               C
C                                                                 C
C    WRITTEN: 15 DEC 84         SYSTEM: ECLIPSE                   C
C                                                                 C
C    PURPOSE: THIS PROGRAM WILL COPY ONE "WORD" (WHICH THE USER   C
C             HAS CHOSEN) AND BLANK OUT THE OTHER WORDS WITHIN     C
C             THE LINE.                                           C
C                                                                 C
C                                                                 C
C*****************************************************************C
C                                                                 C
      IMPLICIT INTEGER(C,D,E)
      INTEGER IARRAY(512),IBUF(300),IBUF2(300),INAME(20)
      INTEGER DCNTR(20),DNTR(20),ECNTR(20),ENTR(20)
C                                                                 C
C.....................................................;...........C
C                                                     ;           C
C                                                     ;           C
      ACCEPT"ENTER FILENAME: "                        ;           C
      READ(11,199)INAME(1)                            ;   INPUT   C
 199  FORMAT(S40)                                     ;  FILENAME C
C                                                     ;           C
C                                                     ;           C
C.....................................................;...........C
C                                                     ;           C
C                                                     ;  READ WORD C
      OPEN 2,"WDAT.TX"                                ; POSITIONS FROM C
      READ BINARY(2)DNTR,ENTR,D                       ;  WDAT.TX  C
C                                                     ;           C
C.....................................................;...........C
C                                                     ;           C
C                                                     ;           C
 299  ACCEPT"WHICH WORD TO PROCESS: ",IWORD           ;           C
      IF((IWORD.GT.D-1)                               ;   INPUT   C
     /            .OR.                               ; WORD NUMBER C
     /  (IWORD.LE.0))GOTO 299                        ;           C
C                                                     ;           C
C                                                     ;           C
C.....................................................;...........C
C                                                       ;         C
C                                                       ;         C
      CALL PICFMT(IBUF,INAME,IFLG)                      ;         C
      CALL GFMT(IBUF,NROWS,NCOLS,NBITS,IMODE)           ;         C
                                                        ; PICBUF INPUT C
      CALL MAKB(IBUF)                                   ;   BLOCK C
      CALL PICIN(IBUF,INAME,IFLG)                       ;         C
                                                        ;         C
      CALL PFMT(IBUF2,NROWS,NCOLS,NBITS,IMODE)          ;         C
      CALL MAKB(IBUF2)                                  ;         C
C                                                       ;         C
C.....................................................;...........C
```

```
C.....C
C                                                    ;    READ LINE      C
      OPEN 1,"LDAT.TX"                               ;  POSITION FROM    C
      READ BINARY(1)DCNTR,ECNTR,C,LIN                ;     LDAT.TX       C
C                                                    ;                   C
C.....;.....C
C                                                    ;                   C
C                                                    ;                   C
      TYPE"LINE:",LIN,"      WORD:",IWORD            ;                   C
                                                     ;        I/O        C
      TYPE"FWORD WORKING"                            ;                   C
C                                                    ;                   C
C.....;.....C
C                                                    ;                   C
      DO 5 I=1,256                                   ;                   C
         IARRAY(I)=15                                ;                   C
 5    CONTINUE                                       ;    CREATE A       C
                                                     ;     WHITE         C
      DO 10 I=1,256                                  ;   BACKGROUND      C
         CALL PROW(IBUF2,I,1,256,IARRAY)             ;                   C
 10   CONTINUE                                       ;                   C
C                                                    ;                   C
C.....;.....
C                                                                ;   P'     
      ILEN=ENTR(IWORD)-DNTR(IWORD)+1                             ;ORIG    L C
      DO 15 I=DCNTR(LIN),ECNTR(LIN)                              ;  WORD    C
         CALL GROW(IBUF,I,DNTR(IWORD)-1,ILEN,IARRAY)             ;  ONTO
         CALL PROW(IBUF2,I,DNTR(IWORD)-1,ILEN,IARRAY)            ; WHITE    C
 15   CONTINUE                                                   ; BACK-    C
C                                                               ; GROUND   C
C.....;.....C
C                                                    ;                   C
      CLOSE 2                                        ;    WRITE WORD     C
      OPEN 2,"WDAT.TX"                               ;  POSITIONS INTO   C
      WRITE BINARY(2)DNTR,ENTR,D,IWORD               ;     WDAT.TX       C
C                                                    ;                   C
C.....;.....C
C                                                    ;                   C
C                                                    ;                   C
      CALL PICOUT(IBUF2,INAME,IFLG)                  ;  PICBUF OUTPUT    C
      CALL RELB(IBUF)                                ;      BLOCK        C
      CALL RELB(IBUF2)                               ;                   C
C                                                    ;                   C
C.....;.....C
C                                                                        C
        STOP"<7>FINISHED"
        END
```

```
C*****************************************************************C
C                                                                 C
C                                                                 C
C    PROGRAM: LINES.FR          LANGUAGE: FORTRAN V               C
C                                                                 C
C    WRITTEN: 15 DEC 84              SYSTEM: ECLIPSE              C
C                                                                 C
C    PURPOSE: THIS PROGRAM WILL FIND THE LINES OF PRINT ON A      C
C             PAGE OF DIGITIZED TEXT.                             C
C                                                                 C
C           NOTE: A BLANK HORIZONTAL LINE MUST DIVIDE EACH        C
C                 LINE OF PRINTED TEXT.                           C
C                                                                 C
C                                                                 C
C*****************************************************************C
C                                                                 C
C                                                                 C
      PARAMETER BLEV=16.0D0
      IMPLICIT INTEGER(C,D,E)
      IMPLICIT DOUBLE PRECISION(A,B,T)
      INTEGER IARRAY(512),IBUF(300),INAME(20)
      INTEGER DCNTR(20),ECNTR(20)
      DOUBLE PRECISION AVE(258),TAVE
C                                                                 C
C.....................................................;...........C
C                                                     ;           C
      ACCEPT"ENTER FILENAME: "                        ;           C
      READ(11,199)INAME(1)                            ; INPUT FILENAME C
  199 FORMAT(S+0)                                     ;           C
C                                                     ;           C
C.....................................................;...........C
C                                                     ;           C
      CALL PICFMT(IBUF,INAME,IFLG)                    ;           C
      CALL GFMT(IBUF,NROWS,NCOLS,NBITS,IMODE)         ; PICBUF INPUT  C
                                                      ;   BLOCK   C
      CALL MAKB(IBUF)                                 ;           C
      CALL PICIN(IBUF,INAME,IFLG)                     ;           C
C                                                     ;           C
C.....................................................;...........C
C                                                     ;           C
      TYPE"LINES WORKING"                             ;    I/O    C
C                                                     ;           C
C.....................................................;...........C
C                                                     ;           C
      DO 5 I=1,258                                    ;           C
           AVE(I)=0.0D0                               ;           C
    5 CONTINUE                                        ; INITIALIZE C
                                                      ;   ALL     C
      DO 10 I=1,20                                    ; VARIABLES C
            DCNTR(I)=0                                ;           C
            ECNTR(I)=0                                ;           C
   10 CONTINUE                                        ;           C
C                                                     ;           C
C.....................................................;...........C
```

```
                DO 15 J=I,ILN                                  ;                    C
                      IF(IARY(J).EQ.15)GOTO 92                 ;     GET LENGTH     C
                      ID=ID+1                                  ;       OF ROW       C
 15             CONTINUE                                       ;                    C
                                                               ;                    C
 92         IT=IT+1                                           ;                    C
            IVAL(IT)=ID                                       ;                    C
 5          CONTINUE                                          ;                    C
C                                                              ;                    C
C..............................................................;....................C
C                                                              ;                    C
           WRITE FREE(12)"IT",IT                               ;        I/O         C
C                                                              ;                    C
C..............................................................;....................C
C                                                                    ;              C
           JD=IT-1                                                   ;              C
           DO 35 I=1,JD                                              ;              C
                J=IT-I                                               ;              C
                DO 40 K=1,J                                          ;              C
                   IF(IVAL(K).LE.IVAL(K+1))GOTO 40                   ;  BUBBLE SORT C
                   NT=IVAL(K)                                        ;    ROW LENGTHS C
                   IVAL(K)=IVAL(K+1)                                 ;              C
                   IVAL(K+1)=NT                                      ;              C
 40             CONTINUE                                             ;              C
 35         CONTINUE                                                 ;              C
C                                                                    ;              C
C....................................................................;..............C
C                                                                          ;        C
           TEMP=0                                                          ;        C
           IC=1                                                            ;        C
                                                                           ;        C
 96        NCNT=0                                                          ;        C
           DO 45 ICNT=1,IT                                                 ;        C
             IF(IVAL(ICNT).EQ.IVAL(IC))NCNT=NCNT+1                         ;        C
 45        CONTINUE                                                        ;  DETERMINE C
                                                                           ; THICKNESS OF C
           IF(NCNT.LE.TEMP)GOTO 94                                         ; ROW LENGTHS C
           TEMP=NCNT                                                       ;        C
           ITEMP=IC                                                        ;        C
                                                                           ;        C
                                                                           ;        C
94         IC=IC+1                                                         ;        C
           IF(IC.GT.IT)GOTO 95                                             ;        C
           GOTO 96                                                         ;        C
C..........................................................................;........C
C                                                                                ;  C
 95        DO 50 IH=1,IT                                                         ;  C
              WRITE FREE(12)"IVAL",IH,IVAL(IH)                                 ;  C
 50        CONTINUE                                                              ;  C
                                                                                 ;  C
           MAX=IVAL(ITEMP)                                                       ;  C
                                                                                 ;  C
           WRITE FREE(12)"MAX",MAX                                               ;  C
C                                                                                ;  C
C................................................................................;..C
```

```
C*****************************************************************C
C                                                                 C
C     PROGRAM: RLNX.FR              LANGUAGE: FORTRAN V           C
C                                                                 C
C     WRITTEN: 15 DEC 84            SYSTEM: ECLIPSE               C
C                                                                 C
C     PURPOSE: THIS PROGRAM DETERMINES THE THICKNESS OF A LETTER. C
C                                                                 C
C*****************************************************************C
C                                                                 C
      INTEGER IBUF(300)
      INTEGER IARY(256)
      INTEGER INAME(20)
      INTEGER IVAL(100)
C.....................................................................C
C                                                 ;                 C
      ACCEPT"ENTER FILENAME: "                    ;                 C
      READ(11,199)INAME(1)                        ;  ENTER FILENAME C
 199  FORMAT(S40)                                 ;                 C
C.....................................................;.............C
C                                                     ;             C
      ACCEPT"ENTER TOP ROW: ",IST                     ;   ENTER     C
      ACCEPT"ENTER BOTTOM ROW: ",LST                  ;   WINDOW    C
      ACCEPT"ENTER LEFT COLUMN: ",ICO                 ; COORDINATES C
      ACCEPT"ENTER RIGHT COLUMN: ",IRC                ;             C
C.....................................................;.............C
C                                                       ;           C
      CALL PICFMT(IBUF,INAME,IFLG)                      ; PICBUF INPUT C
      CALL GFMT(IBUF,NROWS,NCOLS,NBITS,IMODE)           ;   BLOCK   C
      CALL MAKB(IBUF)                                   ;           C
      CALL PICIN(IBUF,INAME,IFLG)                       ;           C
C.....................................................;.............C
C                                                   ;               C
C                                                   ;               C
      ILN=IRC-ICO+1                                 ;               C
      IT=0                                          ;  INITIALIZE   C
                                                    ;     ALL       C
      DO 1 JN=1,100                                 ;  VARIABLES    C
           IVAL(JN)=99999                           ;               C
 1    CONTINUE                                      ;               C
C                                                   ;               C
C.....................................................;.............C
C                                                       ;           C
C                                                       ;           C
      DO 5 IR=IST,LST                                   ;           C
           CALL GROW(IBUF,IR,ICO,ILN,IARY)              ;           C
                                                        ;           C
           DO 10 I=1,ILN                                ;           C
                 IF(IARY(I).EQ.15)GOTO 10               ; GET TO FIRST C
                 GOTO 91                                ; BLACK PIXEL C
 10        CONTINUE                                     ;  ON LINE  C
                                                        ;           C
           GOTO 5                                       ;           C
                                                        ;           C
 91        ID=0                                         ;           C
```

```
CALL PICOUT(IBUF2,ONAME,IFLG)
CALL RELB(IBUF)
CALL RELB(IBUF2)

STOP"<7>FINISHED"
END
```

```
      DO 5 I=YR(1),YR(3)
           CALL GROW(IBUF,I,1,NCOLS,IARRAY)
           DO 10 J=1,256
                IARRAY(J)=NOT(IARRAY(J))
10         CONTINUE
           CALL PROW(IBUF,I,1,NCOLS,IARRAY)
5     CONTINUE

        TYPE"NEGATIVE FORMED"


C   THIS PROGRAM WILL DIVIDE A PICTURE INTO FOUR QUADRANTS
C   AND INTERCHANGE EACH OF ITS DIAGONAL QUADRANTS. YOU
C   MUST PROVIDE THE UPPER LEFT HAND AND THE LOWER RIGHT
C   HAND COORDINATES OF THE ORIGINAL PICTURE.

        XCT=(XC(3)+XC(1))/2
        XCT=INT(XCT)
        XC(2)=XCT+1


        YR(2)=(YR(3)+YR(1))/2
        YR(2)=INT(YR(2))


        ICLN(1)=XC(2)-XC(1)
        ICLN(2)=XC(3)-XCT


        IC(1)=1
        IC(2)=257-ICLN(1)


        IRW(1)=YR(2)-YR(1)+1
        IR(1)=257-IRW(1)
        IR(2)=1


        DO 15 J=1,2
             KR=IR(J)
             L=1
             DO 20 IROW=YR(J),YR(J+1)
                  CALL GROW(IBUF,IROW,XC(L),ICLN(L),IARRAY)
                  CALL PROW(IBUF2,KR,IC(L+1),ICLN(L),IARRAY)

                  CALL GROW(IBUF,IROW,XC(L+1),ICLN(L+1),IARRAY)
                  CALL PROW(IBUF2,KR,IC(L),ICLN(L+1),IARRAY)

                  KR=KR+1
20           CONTINUE
             YR(2)=YR(2)+1
15    CONTINUE
```

```
C*****************************************************************C
C                                                                 C
C                                                                 C
C   ORIGINAL PROGRAM: NEG.FR BY R. A. SIMMONS                     C
C                                                                 C
C                                                                 C
C   PROGRAM: QSHIFT.FR          LANGUAGE: FORTRAN V               C
C                                                                 C
C   WRITTEN: 15 DEC 84          SYSTEM: ECLIPSE                   C
C                                                                 C
C   PURPOSE: THIS PROGRAM WILL PERFORM THE FOLLOWING OPERATIONS:C
C          1. CREATE THE NEGATIVE OF A SECTION OF A VIDEO FILE    C
C          2. DIVIDE THE SECTION INTO FOUR QUADRANTS              C
C          3. INTERCHANGE EACH DIAGONAL QUADRANT                  C
C                                                                 C
C*****************************************************************C
C                                                                 C
C                                                                 C
      IMPLICIT INTEGER(X,Y,Z)
      INTEGER IARRAY(300),IBUF(300),IBUF2(300)
      INTEGER INAME(20),ONAME(20)
      INTEGER XC(4),YR(4),ICLN(2)
      INTEGER IC(2),IR(2),IRW(2)
C                                                                 C
C                                                                 C
      ACCEPT"ENTER INPUT FILENAME: "
      READ(11,199)INAME(1)

      ACCEPT"ENTER OUTPUT FILENAME: "
      READ(11,199)ONAME(1)
 199  FORMAT (S40)


 991  TYPE"ENTER ROW AND COLUMN OF CORNERS:"
      ACCEPT"UPPER LEFT HAND CORNER -->",YR(1),XC(1)
      ACCEPT"LOWER RIGHT HAND CORNER-->",YR(3),XC(3)


    IF((XC(1).GE.XC(3)).AND.(YR(1).GE.YR(3)))GOTO 991

C ORIGINAL PROGRAM: NEG.FR BY R. A. SIMMONS
C REWRITTEN: NEG1.FR
C THIS PROGRAM WILL CREATE THE NEGATIVE OF A VIDEO FILE.

      TYPE"QSHIFT WORKING"

      CALL PICFMT(IBUF,INAME,IFLG)
      CALL GFMT(IBUF,NROWS,NCOLS,NBITS,IMODE)

      CALL MAKB(IBUF)
      CALL PICIN(IBUF,INAME,IFLG)

      CALL GFMT(IBUF,NROWS,NCOLS,NBITS,IMODE)
      CALL PFMT(IBUF2,NROWS,NCOLS,NBITS,IMODE)
      CALL MAKB(IBUF2)
```

```
C.....................................................;...................C
C                                                    ;                   C
      DO 20 IR=IST,LST                               ;                   C
         CALL GROW(IBUF,IR,1,256,IARY)               ;    CONVERT        C
                                                     ;                   C
         DO 25 J=1,256                               ;     (0/15)        C
            IF(IARY(J).EQ.15)GOTO 25                 ;    GREYSCALE      C
            CALL PPNT(IBUF2,IR,J,1)                  ;       TO          C
 25      CONTINUE                                    ;     (1/0)         C
                                                     ;    GREYSCALE      C
 20    CONTINUE                                      ;                   C
C                                                    ;                   C
C.....................................................;...................C
C                                                    ;                   C
C                                                    ;                   C
      DO 35 IR=IST,LST                               ;                   C
        CALL GROW(IBUF2,IR,1,256,IARY)               ;                   C
        DO 40 J=2,255                                ;                   C
          IF(IARY(J).EQ.0)GOTO 40                    ;                   C
          CALL GROW(IBUF2,IR-1,J-1,3,ICRY)           ;    CALCULATE      C
          CALL GROW(IBUF2,IR,J-1,3,IDRY)             ;    NUMBER OF      C
          CALL GROW(IBUF2,IR+1,J-1,3,IERY)           ;     NEAREST       C
          IX=0                                       ;    NEIGHBORS      C
          DO 45 K=1,3                                ;                   C
            IX=ICRY(K)+IDRY(K)+IERY(K)+IX            ;                   C
 45       CONTINUE                                   ;                   C
          CALL PPNT(IBUF3,IR,J,IX)                   ;                   C
 40     CONTINUE                                     ;                   C
 35   CONTINUE                                       ;                   C
C                                                    ;                   C
C.....................................................;...................C
C                                                                        C
      TYPE"<7>MTRXA FINISHED"
      RETURN
      END
```

```
C*****************************************************************C
C                                                                 C
C                                                                 C
C     PROGRAM: MTRXA.FR              LANGUAGE: FORTRAN V          C
C                                                                 C
C     WRITTEN: 15 DEC 84             SYSTEM: ECLIPSE              C
C                                                                 C
C     PURPOSE:  THIS IS A SUBROUTINE USED BY SPOTS.FR AND         C
C               VTHIN.FR . IT CALCULATES MATRIX A WHERE EACH      C
C               ELEMENT IS THE NUMBER OF NEAREST NEIGHBORS        C
C               THAT A PIXEL HAS (MAXIMUM IS 9, MINIMUM IS 0)     C
C                                                                 C
C                                                                 C
C*****************************************************************C
C                                                                 C
C                                                                 C
        SUBROUTINE MTRXA
C                                                                 C
C.................................................................C
C                                                                 C
      COMMON IBUF(300)
      COMMON IBUF2(300)
      COMMON IBUF3(300)
      COMMON IARY(256)
      COMMON INAME(20)
      COMMON IFLAG
      COMMON IST
      COMMON LST
      REAL   WFILE(5)
      INTEGER ICRY(4)
      INTEGER IDRY(4)
      INTEGER IERY(4)
C.................................................................C
C                                                  ;              C
      TYPE"MTRXA WORKING"                          ;     I/O      C
C                                                  ;              C
C..................................................;..............C
C                                                  ;              C
C                                                  ;              C
      DO 5 I=1,256                                 ;              C
           IARY(I)=0                               ;              C
 5    CONTINUE                                     ;  INITIALIZE  C
                                                   ;     ALL      C
      IX=0                                         ;  VARIABLES   C
                                                   ;              C
      WFILE="BK"                                   ;              C
                                                   ;              C
      CALL PICIN(IBUF2,WFILE,0)                    ;              C
      CALL PICIN(IBUF3,WFILE,0)                    ;              C
C                                                  ;              C
C..................................................;..............C
```

```
C.....................................................................C
 91    CALL RELB(IBUF1)                              ;   PICBUF       C
       CALL RELB(IBUF2)                              ;   OUTPUT       C
C                                                    ;   BLOCK        C
C....................................................;................C
C                                                                     C
       STOP
       END
```

```
                                                    ;                   C
      TYPE"CHANGED TARGET"                          ;                   C
                                                    ;                   C
      CALL PICOUT(IBUF2,INM2,IFLG)                  ;                   C
                                                    ;                   C
      GOTO 91                                       ;                   C
C                                                   ;                   C
C.....................................................;...................C
C                                                   ;                   C
C                                                   ;                   C
 80   ER=E1/E2                                      ;                   C
      WRITE FREE(10)"TEMPLATE IS",ER,               ;       I/O         C
     /  "  TIMES THE TARGET ENERGY"                 ;                   C
C                                                   ;                   C
C                                                   ;                   C
C.....................................................;...................C
C                                                   ;                   C
C                                                   ;   ENERGIES       C
      IF(ER.GE.1.5D0)GOTO 77                        ;     ARE           C
      TYPE"ENERGIES ARE THE SAME"                   ;  THE  SAME        C
      GOTO 91                                       ;                   C
C                                                   ;                   C
C                                                   ;                   C
C.....................................................;...................C
C                                                   ;                   C
                                                    ;                   C
 77   ER=E2/E1                                      ;                   C
      E3=IDINT((ER*15.0D0)+0.5D0)                   ; TEMPLATE ENERGY   C
      E4=E3+1                                       ; IS GREATER THAN   C
                                                    ;   THE WINDOW      C
      IE3D=IABS((IE1*15)-(IE2*E3))                  ;    ENERGY         C
      IE4D=IABS((IE1*15)-(IE2*E4))                  ;                   C
      IF(IE3D.GE.IE4D)E3=E4                         ;                   C
                                                    ;                   C
      IF(E3.LT.0)E3=0                               ;                   C
      IF(E3.GT.15)E3=15                             ;  SET TEMPLATE     C
                                                    ;  ENERGY EQUAL      C
      DO 57 IR=39,65                                ;   TO WINDOW       C
            CALL GROW(IBUF1,IR,ICL,ILN,IARY)        ;     ENERGY        C
                                                    ;                   C
            DO 38 J=1,ILN                           ;    SET THE        C
                  IF(IARY(J).EQ.0)IARY(J)=15-E3     ;   TEMPLATE       C
 38         CONTINUE                                ;  ENERGY EQUAL     C
            CALL PROW(IBUF1,IR,ICL,ILN,IARY)        ; TO THE WINDOW     C
 57   CONTINUE                                      ;     ENERGY        C
                                                    ;                   C
      TYPE"TEMPLATE CHANGED"                        ;                   C
                                                    ;                   C
      CALL PICOUT(IBUF1,INM1,IFLG)                  ;                   C
                                                    ;                   C
C.....................................................;...................C
```

```
C.....C
C                                                                     C
C                                                                     C
      WRITE FREE(10)"TEMPLATE ENERGY",E1                              C
      WRITE FREE(10)"TARGET ENERGY",E2                                C
C                                                                     C
C.....C
C                                                                     C
C                                                                     C
C                                                                     C
      IF(E2.LT.E1)GOTO 80                                             C
C                                                                     C
C                                                                     C
C                                                                     C
C.....C
C                                                                     C
C                                                                     C
      ER=E2/E1                                                        C
      WRITE FREE(10)"TARGET IS",ER,                                   C
     / "  TIMES THE TEMPLATE ENERGY"                                  C
C                                                                     C
C.....C
C                                                                     C
C                                                                     C
      IF(ER.GE.1.01D0)GOTO 67                                         C
      TYPE"ENERGIES ARE THE SAME"                                     C
      GOTO 91                                                         C
C                                                                     C
C.....C
C                                                                     C
C                                                                     C
 67   ER=E1/E2                                                        C
      E3=IDINT((ER*15.0D0)+0.5D0)                                     C
      E4=E3+1                                                         C
                                                                      C
                                                                      C
      IE3D=IABS((IE1*15)-(IE2*E3))                                    C
      IE4D=IABS((IE1*15)-(IE2*E4))                                    C
      IF(IE3D.GE.IE4D)E3=E4                                           C
                                                                      C
                                                                      C
      IF(E3.LT.0)E3=0                                                 C
      IF(E3.GT.15)E3=15                                               C
                                                                      C
                                                                      C
      DO 40 IR=IST,LST                                                C
            CALL GROW(IBUF2,IR,ICO,ILX,IARY)                          C
                                                                      C
            DO 45 J=1,ILN                                             C
                  IF(IARY(J).EQ.0)IARY(J)=15-E3                       C
 45         CONTINUE                                                  C
                                                                      C
       CALL PROW(IBUF2,IR,ICO,ILX,IARY)                               C
                                                                      C
 40   CONTINUE                                                        C
```

```
C.....;.....C
C     ;     C
      E1=0.0D0
      E2=0.0D0
      E3=0
      E4=0
C.....;.....C
      ;     C
      ICOL(1)=2
      ICOL(2)=35
      ICOL(3)=65
      ICOL(4)=96
      ICOL(5)=129
      ICOL(6)=158
      ICOL(7)=185
      ICOL(8)=218

      ILEN(1)=34
      ILEN(2)=31
      ILEN(3)=32
      ILEN(4)=34
      ILEN(5)=30
      ILEN(6)=28
      ILEN(7)=34
      ILEN(8)=34
C.....;.....C
C     ;     C
      ICL=ICOL(INUM)
      ILN=ILEN(INUM)
      DO 5 IR=39,65
           CALL GROW(IBUF1,IR,ICL,ILN,IARY)

           DO 10 J=1,ILN
                IF(IARY(J).EQ.0)E1=E1+1.0D0
 10        CONTINUE

 5    CONTINUE

      E1=DSQRT(E1)
      IE1=IDINT(E1)
C.....;.....C
C     ;     C
      ILX=ILC-ICO+1
      DO 25 IR=IST,LST
           CALL GROW(IBUF2,IR,ICO,ILX,IARY)

           DO 30 J=1,ILX
                IF(IARY(J).EQ.0)E2=E2+1.0D0
 30        CONTINUE

 25   CONTINUE

      E2=DSQRT(E2)
      IE2=IDINT(E2)
C.....;.....C
```

INITIALIZE ALL VARIABLES

STARTING COLUMN AND LENGTH OF CHARACTERS IN TEMPLATE SET

CALCULATE ENERGY IN TEMPLATE

CALCULATE ENERGY IN WINDOW

```
C*****************************************************************C
C                                                                 C
C                                                                 C
C    PROGRAM: MEND.FR              LANGUAGE: FORTRAN V            C
C                                                                 C
C    WRITTEN: 15 DEC 84            SYSTEM: ECLIPSE                C
C                                                                 C
C    PURPOSE: THIS PROGRAM WILL MODIFY THE ENERGY OF  A LOW       C
C             ENERGY INPUT TO EQUAL A HIGH ENERGY INPUT.          C
C       NOTE: THIS PROGRAM READS FROM A STANDARD TEMPLATE FILE    C
C             (FIGURE 7.1).FOR RESEARCH, THE WINDOW ENERGY WAS    C
C             ALWAYS SET TO THE TEMPLATE ENERGY. TEMPLATE         C
C             CHARACTERS ALL HAD ABOUT THE SAME ENERGY.           C
C                                                                 C
C*****************************************************************C
C                                                                 C
        INTEGER INM1(7),INM2(7)
        INTEGER IARY(256),ICOL(8),ILEN(8)
        INTEGER IBUF1(300),IBUF2(300)
        INTEGER INUM,E3,E4
        DOUBLE PRECISION EN1,EN2,ER,ED
C.....................................................C...............C
C                                                     ;               C
      ACCEPT"ENTER TEMPLATE FILENAME: "               ;               C
      READ(11,199)INM1(1)                             ;    ENTER      C
                                                      ;  FILE NAME    C
      ACCEPT"ENTER TARGET FILENAME: "                 ;               C
      READ(11,199)INM2(1)                             ;               C
 199  FORMAT(S20)                                     ;               C
C.....................................................;...............C
C                                                     ;               C
      TYPE"ENTER TEMPLATE TO MEND: "                  ;    ENTER      C
      TYPE"1->A, 2->B, 3->C, 4->D"                    ;  TEMPLATE     C
      TYPE"5->E, 6->F, 7->G, 8->H"                    ;               C
      ACCEPT"ENTER NUMBER: ",INUM                     ;               C
C.....................................................;...............C
C                                                     ;               C
      ACCEPT"ENTER TOP ROW: ",IST                     ;               C
      ACCEPT"ENTER BOTTOM ROW: ",LST                  ;  ENTER WINDOW C
      ACCEPT"ENTER LEFT COLUMN: ",ICO                 ;  COORDINATES  C
      ACCEPT"ENTER RIGHT  COLUMN: ",ILC               ;               C
C.....................................................;...............C
C                                                       ;             C
C     INM1="TEMPSET.OR"                                 ;             C
      CALL PICFMT(IBUF1,INM1,IFLG)                      ;             C
      CALL PICFMT(IBUF2,INM2,IFLG)                      ;             C
                                                        ;             C
      CALL GFMT(IBUF1,NROWS,NCOLS,NBITS,IMODE)          ;  PICBUF INPUT C
                                                        ;     BLOCK   C
      CALL MAKB(IBUF1)                                  ;             C
      CALL MAKB(IBUF2)                                  ;             C
                                                        ;             C
      CALL PICIN(IBUF1,INM1,IFLG)                       ;             C
      CALL PICIN(IBUF2,INM2,IFLG)                       ;             C
C.....................................................;...............C
```

```
C.....................................................................C
C                                                     ;               C
      DO 75 I=1,C-1                                   ;               C
      TYPE"LINE",I," IS AT ROW",DCNTR(I),"  TO",      ;               C
     /     ECNTR(I)                                   ;      I/O      C
 75   CONTINUE                                        ;               C
C                                                     ;               C
C.....................................................;...............C
C                                                     ;               C
      OPEN 1,"LDAT.TX"                                ; STORE LINE    C
                                                      ;POSITIONS IN   C
      WRITE BINARY(1)DCNTR,ECNTR,C                    ;  LDAT.TX      C
C.....................................................;...............C
C                                                     ;               C
        TYPE"<7>FINISHED "                            ;      I/O      C
C.....................................................;...............C
C                                                     ;               C
C                                                     ;    PICBUF     C
 9999   CALL RELB(IBUF)                               ;    OUTPUT     C
C                                                     ;               C
C.....................................................;...............C
C                                                                     C
        END
```

```
C.....C
C                                                                     C
      DO 20 I=1,240                                                   C
            CALL GROW(IBUF,I,1,NCOLS,IARRAY)                          C
            DO 25 J=1,256                                             C
                  K=IARRAY(J)+1                                       C
                  AVE(I+1)=AVE(I+1)+DFLOAT(K)                         C
 25         CONTINUE                                                  C
            AVE(I+1)=AVE(I+1)/256.0D0                                 C
 20   CONTINUE                                                        C
C                                                                     C
C.....C
C                                                                     C
        AVE(1)=16.0D0                                                 C
        DO 30 I=242,258                                               C
              AVE(I)=16.0D0                                           C
 30     CONTINUE                                                      C
C.....C
C                                                                     C
      TAVE=0.0D0                                                      C
      DO 40 I=2,241                                                   C
            TAVE=TAVE+AVE(I)                                          C
 40   CONTINUE                                                        C
      TAVE=TAVE/256.0D0                                               C
C.....C
C                                                                     C
      WRITE FREE(10)"TAVE= ",TAVE                                     C
      IF(TAVE.GT.10)GOTO 991                                          C
      TYPE"FILE CANNOT BE PROCESSED"                                  C
      GOTO 9999                                                       C
 991  TYPE"FILE MAY BE PROCESSED"                                     C
C.....C
C                                                                     C
        C=1                                                           C
        DO 45 I=2,242                                                 C
              IF((AVE(I).NE.BLEV)                                     C
     /                .AND.                                          C
     /          (AVE(I-1).EQ.BLEV))GOTO 91                           C
              GOTO 45                                                 C
 91           DCNTR(C)=I-1                                           C
              C=C+1                                                   C
 45     CONTINUE                                                      C
C.....C
C                                                                     C
      C=1                                                             C
      DO 50 I=2,257                                                   C
            IF((AVE(I).NE.BLEV)                                       C
     /                .AND.                                          C
     /          (AVE(I+1).EQ.BLEV))GOTO 92                           C
            GOTO 50                                                   C
 92         ECNTR(C)=I                                               C
            C=C+1                                                     C
 50   CONTINUE                                                        C
C                                                                     C
C.....C
```

```
C.....;.....C
C                                                              C
      TYPE"<7>FINISHED"
      CALL RELB(IBUF)
      STOP
      END
```

```
C*****************************************************************C
C                                                                 C
C                                                                 C
C     PROGRAM: SPOTS.FR              LANGUAGE: FORTRAN V          C
C                                                                 C
C     WRITTEN: 15 DEC 84             SYSTEM: ECLIPSE              C
C                                                                 C
C     PURPOSE: THIS PROGRAM WILL REMOVE THE SPOTS IN THE INPUT.   C
C              IT DELETES POINTS OF ONE PIXEL, FILLS HOLES OF     C
C              ONE PIXEL, AND DELETES TIPS OF SPIKES.             C
C                                                                 C
C                                                                 C
C*****************************************************************C
C                                                                 C
      COMMON IBUF(300)
      COMMON IBUF2(300)
      COMMON IBUF3(300)
      COMMON IARY(256)
      COMMON INAME(20)
      COMMON IFLAG
      COMMON IST
      COMMON LST
      INTEGER IBRY(256)
      REAL    WFILE(5)
C                                                                 C
C.................................................................C
C                                                                 C
      ACCEPT"ENTER INPUT FILENAME: "                              C
      READ(11,199)INAME(1)                    INPUT FILENAME      C
 199  FORMAT(S40)                                                 C
C.....................................................;...........C
C                                                     ;           C
      TYPE"SPOTS WORKING"                             ;    I/O    C
C                                                     ;           C
C.....................................................;...........C
C                                                     ;           C
      CALL PICFMT(IBUF,INAME,IFLG)                    ;           C
      CALL GFMT(IBUF,NROWS,NCOLS,NBITS,IMODE)         ;           C
      CALL MAKB(IBUF)                                 ;           C
      CALL PICIN(IBUF,INAME,IFLG)                     ;           C
                                                      ;  PICBUF INPUT C
      WFILE="BK"                                      ;    BLOCK  C
                                                      ;           C
      CALL GFMT(IBUF,NROWS,NCOLS,NBITS,IMODE)         ; (BK IS A BLACK C
      CALL PFMT(IBUF2,NROWS,NCOLS,NBITS,IMODE);  VIDEO FILE)      C
      CALL MAKB(IBUF2)                                ;           C
      CALL PICIN(IBUF2,WFILE,IFLG)                    ;           C
                                                      ;           C
      CALL GFMT(IBUF,NROWS,NCOLS,NBITS,IMODE)         ;           C
      CALL PFMT(IBUF3,NROWS,NCOLS,NBITS,IMODE);                   C
      CALL MAKB(IBUF3)                                ;           C
      CALL PICIN(IBUF3,WFILE,IFLG)                    ;           C
C                                                     ;           C
C.....................................................;...........C
```

```
C.....;.....C
C                                                    ;                 C
      IST=2                                          ;  SET TOP ROW    C
      LST=241                                        ;  SET END ROW    C
C.....;.....C
C                                                    ;                 C
      CALL MTRXA                                     ;  FORM MATRIX A  C
C.....;.....C
C                                                      ;               C
      WRITE FREE(10)"BEGIN 1"                          ;               C
      DO 5 IR=2,241                                    ;               C
           CALL GROW(IBUF3,IR,1,256,IBRY)              ;               C
           DO 10 J=2,255                               ;               C
                IF(IBRY(J).NE.0)GOTO 10                ;               C
                CALL GPNT(IBUF3,IR-1,J,IARY(1))        ;               C
                CALL GPNT(IBUF3,IR,J-1,IARY(2))        ;               C
                CALL GPNT(IBUF3,IR,J+1,IARY(3))        ; DETERMINE IF  C
                CALL GPNT(IBUF3,IR+1,J,IARY(4))        ;  POINT IS A   C
                 DO 15 K=1,4                           ;     HOLE      C
                       IF(IARY(K).LT.1)GOTO 10         ;               C
  15             CONTINUE                              ;               C
                 CALL PPNT(IBUF,IR,J,0)                ;               C
  10        CONTINUE                                   ;               C
  5    CONTINUE                                        ;               C
      WRITE FREE(10)"END 1"                            ;               C
C.....;.....C
C                                                      ;               C
      WRITE FREE(10)"BEGIN 2"                          ;               C
      DO 30 IR=2,241                                   ;               C
            CALL GROW(IBUF3,IR,1,256,IBRY)             ;               C
            DO 35 J=2,255                              ; DETERMINE IF  C
                  IF(IBRY(J).NE.1)GOTO 35              ;  POINT IS A   C
                  CALL PPNT(IBUF,IR,J,15)              ;     SPOT      C
  35        CONTINUE                                   ;               C
  30   CONTINUE                                        ;               C
      WRITE FREE(10)"END 2"                            ;               C
C.....;.....C
C                                                        ;             C
      WRITE FREE(10)"BEGIN 3"                            ;             C
      DO 50 IR=2,241                                     ;             C
            CALL GROW(IBUF3,IR,1,256,IBRY)               ;             C
            DO 55 J=2,255                                ;             C
                  IF(IBRY(J).NE.2)GOTO 55                ;             C
                  CALL GPNT(IBUF3,IR-1,J-1,IARY(1));                   C
                  CALL GPNT(IBUF3,IR-1,J,IARY(2))  ;                   C
                  CALL GPNT(IBUF3,IR-1,J+1,IARY(3));                   C
                  CALL GPNT(IBUF3,IR,J-1,IARY(4))    ; DETERMINE      C
                  CALL GPNT(IBUF3,IR,J+1,IARY(5))    ;     IF         C
                  CALL GPNT(IBUF3,IR+1,J-1,IARY(6));    POINTS       C
                  CALL GPNT(IBUF3,IR+1,J,IARY(7))    ; ARE SPOTS     C
                  CALL GPNT(IBUF3,IR+1,J+1,IARY(8)); AND TIPS        C
                                                     ; OF SPIKES     C
                                                     ;               C
```

```
                  DO 60 K=1,8                                    ;              C
                        IF(IARY(K).EQ.2)GOTO 91                  ;              C
 60               CONTINUE                                       ;              C
                                                                 ;              C
                  DO 65 L=1,8                                    ;              C
                        IF(IARY(L).EQ.5)GOTO 91                  ;              C
                        IF(IARY(L).EQ.6)GOTO 91                  ;              C
                        IF(IARY(L).EQ.7)GOTO 91                  ;              C
 65               CONTINUE                                       ;              C
                  GOTO 55                                        ;              C
 91               CALL PPNT(IBUF,IR,J,15)                        ;              C
 55          CONTINUE                                            ;              C
 50    CONTINUE                                                  ;              C
                                                                 ;              C
        WRITE FREE(10)"END 3"                                    ;              C
C.....................................................................;.........C
C                                                                ;              C
      WRITE FREE(10)"BEGIN 4"                                    ;              C
                                                                 ;              C
      DO 80 IR=2,241                                             ;              C
            CALL GROW(IBUF3,IR,1,256,IBRY)                       ;              C
            DO 85 J=2,255                                        ;              C
               IF(IBRY(J).NE.4)GOTO 85                           ;              C
               CALL GPNT(IBUF3,IR-1,J-1,IARY(1))                 ;              C
               CALL GPNT(IBUF3,IR-1,J,J,IARY(2))                 ;              C
               CALL GPNT(IBUF3,IR-1,J+1,IARY(3))                 ;              C
               CALL GPNT(IBUF3,IR,J-1,IARY(4))                   ;              C
               CALL GPNT(IBUF3,IR,J+1,IARY(5))                   ; DETERMINE IF C
               CALL GPNT(IBUF3,IR+1,J-1,IARY(6))                 ;  POINTS ARE  C
               CALL GPNT(IBUF3,IR+1,J,IARY(7))                   ;    TIPS OF   C
               CALL GPNT(IBUF3,IR+1,J+1,IARY(8))                 ;     SPIKES   C
                                                                 ;              C
               IBRY(1)=IARY(1)+IARY(2)+IARY(3)                   ;              C
               IBRY(2)=IARY(1)+IARY(4)+IARY(6)                   ;              C
               IBRY(3)=IARY(3)+IARY(5)+IARY(8)                   ;              C
               IBRY(4)=IARY(6)+IARY(7)+IARY(8)                   ;              C
                                                                 ;              C
                  DO 90 K=1,8                                    ;              C
                        IF(IBRY(K).EQ.0)GOTO 94                  ;              C
 90               CONTINUE                                       ;              C
                  GOTO 85                                        ;              C
 94               CALL PPNT(IBUF,IR,J,15)                        ;              C
 85          CONTINUE                                            ;              C
 80    CONTINUE                                                  ;              C
C.....................................................................;.........C
C                                                                ;              C
 9999   CALL PICOUT(IBUF,INAME,IFLG)                             ;    PICBUF    C
        CALL RELB(IBUF)                                          ;    OUTPUT    C
        CALL RELB(IBUF2)                                         ;    BLOCK     C
        CALL RELB(IBUF3)                                         ;              C
C.....................................................................;.........C
C                                                                                C
        STOP"<7>SPOTS FINISHED"
        END
```

```
C*****************************************************************C
C    PROGRAM: TDT3.FR              LANGUAGE: FORTRAN V                C
C                                                                     C
C    WRITTEN: 15 DEC 84            SYSTEM: ECLIPSE                    C
C                                                                     C
C    PURPOSE: THIS IS A SUBROUTINE USED BY THIN.FR TO TEST FOR        C
C             D1 AND NEIGHBOR VALUE OF 8 DELETION.                    C
C*****************************************************************C
C                                                                     C
      SUBROUTINE TDT3(II,JJ)
C                                                                     C
      COMMON IBUF(300)
      COMMON IBUF2(300)
      COMMON IBUF3(300)
      COMMON IARY(256)
      COMMON INAME(20)
      COMMON IFLAG
      COMMON IST
      COMMON LST
      INTEGER IVAL(9)
C                                                                     C
      DO 5 I=1,9
           IVAL(I)=0
 5    CONTINUE

      CALL GPNT(IBUF3,II-1,JJ-1,IVAL(1))
      CALL GPNT(IBUF3,II-1,JJ+1,IVAL(2))
      CALL GPNT(IBUF3,II+1,JJ-1,IVAL(3))
      CALL GPNT(IBUF3,II+1,JJ+1,IVAL(4))

      DO 7 I=1,4
           IF(IVAL(I).EQ.8)GOTO 100
 7    CONTINUE

      CALL GPNT(IBUF3,II-1,JJ,IVAL(5))
      CALL GPNT(IBUF3,II,JJ-1,IVAL(6))
      CALL GPNT(IBUF3,II,JJ+1,IVAL(7))
      CALL GPNT(IBUF3,II+1,JJ,IVAL(8))

      IF((IVAL(1).NE.6) .OR. (IVAL(1).NE.7))GOTO 91
      IF((IVAL(5).NE.0) .OR. (IVAL(6).NE.0))GOTO 100

 91   IF((IVAL(2).NE.6) .OR. (IVAL(2).NE.7))GOTO 92
      IF((IVAL(5).NE.0) .OR. (IVAL(7).NE.0))GOTO 100

 92   IF((IVAL(3).NE.6) .OR. (IVAL(3).NE.7))GOTO 93
      IF((IVAL(6).NE.0) .OR. (IVAL(8).NE.0))GOTO 100

 93   IF((IVAL(4).NE.6) .OR. (IVAL(4).NE.7))GOTO 94
      IF((IVAL(7).NE.0) .OR. (IVAL(8).NE.0))GOTO 100

      GOTO 94
100   CALL PPNT(IBUF,II,JJ,15)
 94   RETURN
      END
```

```
C*****************************************************************C
C                                                                 C
C    PROGRAM: TDT4.FR            LANGUAGE: FORTRAN V              C
C                                                                 C
C    WRITTEN: 15 DEC 84          SYSTEM: ECLIPSE                  C
C                                                                 C
C    PURPOSE: THIS IS A SUBROUTINE USED BY THIN.FR TO TEST FOR    C
C                  D2 AND NEIGHBOR VALUE OF 9 DELETION.           C
C                                                                 C
C*****************************************************************C
C                                                                 C
         SUBROUTINE TDT4(II,JJ)
C                                                                 C
         COMMON IBUF(300)
         COMMON IBUF2(300)
         COMMON IBUF3(300)
         COMMON IARY(256)
         COMMON INAME(20)
         COMMON IFLAG
         COMMON IST
         COMMON LST
         INTEGER IVAL(9)
C                                                                 C
         DO 5 I=1,9
              IVAL(I)=0
 5       CONTINUE

         CALL GPNT(IBUF3,II-1,JJ-1,IVAL(1))
         CALL GPNT(IBUF3,II-1,JJ+1,IVAL(2))
         CALL GPNT(IBUF3,II+1,JJ-1,IVAL(3))
         CALL GPNT(IBUF3,II+1,JJ+1,IVAL(4))

         DO 7 I=1,4
              IF(IVAL(I).EQ.9)GOTO 100
 7       CONTINUE

         CALL GPNT(IBUF3,II-1,JJ,IVAL(5))
         CALL GPNT(IBUF3,II,JJ-1,IVAL(6))
         CALL GPNT(IBUF3,II,JJ+1,IVAL(7))
         CALL GPNT(IBUF3,II+1,JJ,IVAL(8))

         IF((IVAL(1).LE.3) .OR. (IVAL(1).GE.9)) GOTO 91
         IF((IVAL(1).EQ.4)
     /              .OR. (IVAL(1).EQ.6)
     /                           .OR. (IVAL(1).EQ.8)) GOTO 91
         IF((IVAL(5).NE.0) .AND. (IVAL(6).NE.0))GOTO 100

 91      IF((IVAL(2).LE.3) .OR. (IVAL(2).GE.9)) GOTO 92
         IF((IVAL(2).EQ.4)
     /              .OR. (IVAL(2).EQ.6)
     /                           .OR. (IVAL(2).EQ.8)) GOTO 92
         IF((IVAL(5).NE.0) .AND. (IVAL(7).NE.0))GOTO 100
```

```
92      IF((IVAL(3).LE.3) .OR. (IVAL(3).GE.9)) GOTO 93
        IF((IVAL(3).EQ.4)
     /                .OR. (IVAL(3).EQ.6)
     /                              .OR. (IVAL(3).EQ.8)) GOTO 93
        IF((IVAL(6).NE.0) .AND. (IVAL(8).NE.0))GOTO 100

93      IF((IVAL(4).LE.3) .OR. (IVAL(4).GE.9)) GOTO 94
        IF((IVAL(4).EQ.4)
     /                .OR. (IVAL(4).EQ.6)
     /                              .OR. (IVAL(4).EQ.8)) GOTO 94
        IF((IVAL(7).NE.0) .AND. (IVAL(8).NE.0))GOTO 100

        GOTO 94

100     CALL PPNT(IBUF,II,JJ,15)

94      RETURN
        END
```

```
C*****************************************************************C
C                                                                 C
C    PROGRAM: TDT5.FR              LANGUAGE: FORTRAN V            C
C                                                                 C
C    WRITTEN: 15 DEC 84            SYSTEM: ECLIPSE                C
C                                                                 C
C    PURPOSE: THIS IS A SUBROUTINE USED BY THIN.FR TO TEST FOR    C
C                  D3 DELETION.                                   C
C                                                                 C
C*****************************************************************C
C                                                                 C
          SUBROUTINE TDT5(II,JJ)
C                                                                 C
          COMMON IBUF(300)
          COMMON IBUF2(300)
          COMMON IBUF3(300)
          COMMON IARY(256)
          COMMON INAME(20)
          COMMON IFLAG
          COMMON IST
          COMMON LST
          INTEGER IVAL(12)
C
          DO 5 I=1,12
               IVAL(I)=0
 5        CONTINUE

          CALL GPNT(IBUF3,II-1,JJ-1,IVAL(1))
          CALL GPNT(IBUF3,II-1,JJ+1,IVAL(2))
          CALL GPNT(IBUF3,II+1,JJ-1,IVAL(3))
          CALL GPNT(IBUF3,II+1,JJ+1,IVAL(4))

          CALL GPNT(IBUF3,II-1,JJ,IVAL(5))
          CALL GPNT(IBUF3,II,JJ+1,IVAL(6))
          CALL GPNT(IBUF3,II,JJ-1,IVAL(7))
          CALL GPNT(IBUF3,II+1,JJ,IVAL(8))

          IVAL(9)=IVAL(3)+IVAL(8)+IVAL(4)
          IVAL(10)=IVAL(1)+IVAL(7)+IVAL(3)
          IVAL(11)=IVAL(2)+IVAL(6)+IVAL(4)
          IVAL(12)=IVAL(1)+IVAL(5)+IVAL(2)

          DO 10 I=1,4
                IF(IVAL(I).NE.8)GOTO 10
                IF((IVAL(I+4).NE.0) .AND. (IVAL(I+8).EQ.0))GOTO 100
 10       CONTINUE

          GOTO 91

 100      CALL PPNT(IBUF,II,JJ,15)

 91       RETURN
          END
```

```
C*****************************************************************C
C                                                                 C
C    PROGRAM: TDT6.FR                LANGUAGE: FORTRAN V          C
C                                                                 C
C    WRITTEN: 15 DEC 84              SYSTEM: ECLIPSE              C
C                                                                 C
C    PURPOSE: THIS IS A SUBROUTINE USED BY THIN.FR TO TEST FOR    C
C                  D4 DELETION.                                   C
C                                                                 C
C*****************************************************************C
C                                                                 C
         SUBROUTINE TDT6(II,JJ)
C                                                                 C
         COMMON IBUF(300)
         COMMON IBUF2(300)
         COMMON IBUF3(300)
         COMMON IARY(256)
         COMMON INAME(20)
         COMMON IFLAG
         COMMON IST
         COMMON LST
         INTEGER IVAL(12)
C
         DO 5 I=1,12
              IVAL(I)=0
 5       CONTINUE

         CALL GPNT(IBUF3,II-1,JJ-1,IVAL(1))
         CALL GPNT(IBUF3,II-1,JJ+1,IVAL(2))
         CALL GPNT(IBUF3,II+1,JJ-1,IVAL(3))
         CALL GPNT(IBUF3,II+1,JJ+1,IVAL(4))

         CALL GPNT(IBUF3,II-1,JJ,IVAL(5))
         CALL GPNT(IBUF3,II,JJ-1,IVAL(6))
         CALL GPNT(IBUF3,II,JJ+1,IVAL(7))
         CALL GPNT(IBUF3,II+1,JJ,IVAL(8))

         IVAL(9)=IVAL(7)+IVAL(8)+IVAL(4)
         IVAL(10)=IVAL(6)+IVAL(8)+IVAL(3)
         IVAL(11)=IVAL(5)+IVAL(7)+IVAL(2)
         IVAL(12)=IVAL(1)+IVAL(5)+IVAL(6)

         IF(IVAL(1).LE.5) GOTO 91
         IF(IVAL(9).NE.0) GOTO 91
         IF((IVAL(5).NE.0) .AND. (IVAL(6).NE.0))GOTO 100

 91      IF(IVAL(2).LE.5)  GOTO 92
         IF(IVAL(10).NE.0) GOTO 92
         IF((IVAL(5).NE.0) .AND. (IVAL(7).NE.0))GOTO 100

 92      IF(IVAL(3).LE.5)  GOTO 93
         IF(IVAL(11).NE.0) GOTO 93
         IF((IVAL(6).NE.0) .AND. (IVAL(8).NE.0))GOTO 100
```

```
93      IF(IVAL(4).LE.5)  GOTO 94
        IF(IVAL(12).NE.0) GOTO 94
        IF((IVAL(7).NE.0) .AND. (IVAL(8).NE.0))GOTO 100
        GOTO 94

100     CALL PPNT(IBUF,II,JJ,15)

94      RETURN
        END
```

```
C*****************************************************************C
C                                                                 C
C                                                                 C
C    PROGRAM: TGEN.FR            LANGUAGE: FORTRAN V              C
C                                                                 C
C    WRITTEN: 15 DEC 84          SYSTEM: ECLIPSE                  C
C                                                                 C
C    PURPOSE: THIS PROGRAM WILL COPY A CHARACTER FROM TSET2.VD    C
C             (SPECIFIED BY THE USER) ONTO A BLANK GACKGROUND.    C
C             THE USER MAY REPEAT THE PROCESS TO GENERATE A       C
C             SEQUENCE OF ISOLATED LETTERS OR CONCATENATED        C
C             LETTERS.                                            C
C                                                                 C
C                                                                 C
C*****************************************************************C
C                                                                 C
C                                                                 C
      PARAMETER INAME="TSET2.VD"
      PARAMETER WFILE="WT"
      IMPLICIT INTEGER(C,D,E)
      INTEGER ICOL(16),ILEN(16)
      INTEGER IARY(256),IBUF(300),IBUF2(300)
      INTEGER IBRY(256)
C
C.....................................................C
C                                        ;            C
      TYPE"PMOVE WORKING"                ;     I/O    C
C                                        ;            C
C.....................................................C
C                                        ;            C
      CALL PICFMT(IBUF,INAME,IFLG)       ;            C
      CALL GFMT(IBUF,NROWS,NCOLS,NBITS,IMODE) ;       C
                                         ;            C
      CALL MAKB(IBUF)                    ;PICBUF INPUT C
      CALL PICIN(IBUF,INAME,IFLG)        ;  BLOCK     C
                                         ;            C
      CALL PICFMT(IBUF2,WFILE,IFLG)      ;            C
      CALL GFMT(IBUF2,NROWS,NCOLS,NBITS,IMODE);       C
                                         ;            C
      CALL MAKB(IBUF2)                   ;            C
      CALL PICIN(IBUF2,WFILE,IFLG)       ;            C
C                                        ;            C
C.....................................................C
C                                        ;            C
1     TYPE"ENTER CHARACTER TO MOVE:"     ;            C
      TYPE"1->A,  2->B,  3->C,  4->D"    ; ENTER LETTER C
      TYPE"UP TO 16->P"                  ;            C
      ACCEPT"ENTER NUMBER: ",INUM        ;            C
C.....................................................C
C                                        ;            C
                                         ;            C
      DATA (ICOL(I),I=1,16)              ;            C
     /  /5,40,71,101,135,163,189,221,    ; LETTER POSITION C
     /   4,25,53,88,120,160,196,229/     ;            C
                                         ;     AND    C
```

```
      DATA (ILEN(I),I=1,16)                    ;                    C
     /  /27,22,22,25,20,19,26,27,              ;       LENGTH       C
     /   14,16,26,22,30,26,25,21/              ;                    C
C.....................................................;.....................C
C                                                      ;                    C
      ITOP=42                                          ;                    C
      IF(INUM.GE.9)ITOP=113                            ;    SET STARTING    C
      IBOT=63                                          ;      ROWS          C
      IF(INUM.GE.9)IBOT=142                            ;                    C
C                                                      ;                    C
C.....................................................;.....................C
C                                                           ;               C
      TYPE                                                  ;  ENTER        C
      TYPE"ENTER POSITION TO PLACE CHARACTER:"              ;  NEW POSITON  C
      ACCEPT"ENTER TOP ROW : ",JTOP                         ;  OF WORD      C
      ACCEPT"ENTER LEFT COL: ",JCOL                         ;               C
C.....................................................;.....................C
C                                                          ;                C
                                                           ;                C
      ILN=ILEN(INUM)                                       ;   MOVE LETTER  C
      IC=ICOL(INUM)                                        ;       TO       C
      J=JTOP                                               ;                C
      DO 10 I=ITOP,IBOT                                    ;       NEW      C
            CALL GROW(IBUF,I,IC,ILN,IARY)                  ;    POSITION    C
            CALL GROW(IBUF2,J,JCOL,ILN,IBRY)               ;                C
                                                           ;                C
                                                           ;                C
             DO 15 K=1,ILEN                                ;                C
                   IF((IARY(K).EQ.0)                       ;                C
     /                    .OR.                            ;                C
     /                (IBRY(K).EQ.0))IBRY(K)=0            ;                C
 15           CONTINUE                                     ;                C
                                                           ;                C
            CALL PROW(IBUF2,J,JCOL,ILN,IBRY)               ;                C
            J=J+1                                          ;                C
                                                           ;                C
 10   CONTINUE                                            ;                C
C                                                          ;                C
C.....................................................;.....................C
C                                                      ;                    C
      TYPE                                             ;                    C
      ACCEPT"CONTINUE (Y=1): ",ICNT                    ; REPEAT FOR MORE    C
      IF(ICNT.EQ.1)GOTO 1                              ;      LETTERS       C
C                                                      ;                    C
C.....................................................;.....................C
C                                                      ;                    C
      CALL PICOUT(IBUF2,WFILE,IFLG)                    ;   PICBUF OUTPUT    C
      CALL RELB(IBUF)                                  ;      BLOCK         C
      CALL RELB(IBUF2)                                 ;                    C
C                                                      ;                    C
C.....................................................;.....................C
C                                                                           C
      STOP
      END
```

```
C*****************************************************************C
C                                                                 C
C                                                                 C
C    PROGRAM: TT2.FR              LANGUAGE: FORTRAN V             C
C                                                                 C
C    WRITTEN: 15 DEC 84           SYSTEM: ECLIPSE                 C
C                                                                 C
C    PURPOSE: THIS IS A SUBROUTINE USED BY THIN.FR TO TEST FOR    C
C                  P1 DELETION.                                   C
C                                                                 C
C                                                                 C
C*****************************************************************C
C                                                                 C
      SUBROUTINE TT2(II,JJ)
C                                                                 C
C                                                                 C
      COMMON IBUF(300)
      COMMON IBUF2(300)
      COMMON IBUF3(300)
      COMMON IARY(256)
      COMMON INAME(20)
      COMMON IFLAG
      COMMON IST
      COMMON LST
      INTEGER IVAL(9)
C
C
      DO 5 I=1,9
           IVAL(I)=0
 5    CONTINUE

      CALL GPNT(IBUF3,II-1,JJ-1,IVAL(1))
      CALL GPNT(IBUF3,II-1,JJ,IVAL(2))
      CALL GPNT(IBUF3,II-1,JJ+1,IVAL(3))

      CALL GPNT(IBUF3,II,JJ-1,IVAL(4))
      CALL GPNT(IBUF3,II,JJ+1,IVAL(5))

      CALL GPNT(IBUF3,II+1,JJ-1,IVAL(6))
      CALL GPNT(IBUF3,II+1,JJ,IVAL(7))
      CALL GPNT(IBUF3,II+1,JJ+1,IVAL(8))


      DO 10 I=1,8
            IF(IVAL(I).EQ.2)GOTO 100
            IF(IVAL(I).EQ 5)GOTO 100
            IF(IVAL(I).EQ.6)GOTO 100
            IF(IVAL(I).EQ.7)GOTO 100
 10   CONTINUE
      GOTO 94

 100  CALL PPNT(IBUF,II,JJ,15)
94    RETURN
      END
```

```
C*****************************************************************C
C                                                                 C
C    PROGRAM: TT3.FR              LANGUAGE: FORTRAN V             C
C                                                                 C
C    WRITTEN: 15 DEC 84           SYSTEM: ECLIPSE                 C
C                                                                 C
C    PURPOSE: THIS IS A SUBROUTINE USED BY THIN.FR TO TEST FOR    C
C                  P1 DELETION.                                   C
C                                                                 C
C                                                                 C
C*****************************************************************C
C                                                                 C
      SUBROUTINE TT3(II,JJ)
C                                                                 C
      COMMON IBUF(300)
      COMMON IBUF2(300)
      COMMON IBUF3(300)
      COMMON IARY(256)
      COMMON INAME(20)
      COMMON IFLAG
      COMMON IST
      COMMON LST
      INTEGER IVAL(9)
C
C
      DO 5 I=1,9
         IVAL(I)=0
 5    CONTINUE

      CALL GPNT(IBUF3,II-1,JJ,IVAL(1))
      CALL GPNT(IBUF3,II,JJ-1,IVAL(2))
      CALL GPNT(IBUF3,II,JJ+1,IVAL(3))
      CALL GPNT(IBUF3,II+1,JJ,IVAL(4))

      DO 10 I=1,4
         IF(IVAL(I).EQ.4)GOTO 90
 10   CONTINUE

      GOTO 94

 90   IF((IVAL(1).EQ.4) .OR. (IVAL(4).EQ.4))GOTO 91
      GOTO 92
 91   IF((IVAL(2).NE.0) .OR. (IVAL(3).NE.0))GOTO 100

 92   IF((IVAL(2).EQ.4) .OR. (IVAL(3).EQ.4))GOTO 93
      GOTO 94
 93   IF((IVAL(1).NE.0) .OR. (IVAL(4).NE.0))GOTO 100

      GOTO 94

 100  CALL PPNT(IBUF,II,JJ,15)

 94   RETURN
      END
```

This thesis presents an algorithm for recognition of letters of the English alphabet. The letters may be single, isolated letters; or concatenated, groups of letters. The algorithm essentially consists of a two pass correlation in which the size and energy of the unknown characters are set equal to a known template.

UNCLASSIFIED

REPORT DOCUMENTATION PAGE

4 PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/CI/NR 84-54	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Air Force Institute of Technology	6b OFFICE SYMBOL AFIT/NR	7a NAME OF MONITORING ORGANIZATION
6c ADDRESS (City, State and ZIP Code) Wright-Patterson AFB, Ohio 45433	7b ADDRESS (City, State and ZIP Code)
8a NAME OF FUNDING/SPONSORING ORGANIZATION	8b OFFICE SYMBOL	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State and ZIP Code)	10 SOURCE OF FUNDING NOS.	

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT NO

11 TITLE (Include Security Classification)
See Box 19

12 PERSONAL AUTHOR(S)
Eugene Shum, B.E., 2d Lt, USAF

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Yr., Mo., Day)	15 PAGE COUNT
		1984 December	149

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB GR	Character Recognition, Correlation Techniques
06	04		

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

Title: Character Recognition Using Correlation Techniques

Thesis Advisor: Matthew Kabrisky, Ph.D.
Professor of Electrical Engineering

Approved for public release: IAW AFR 190-17.
Director of Public Affairs Development
Air Force Institute of Technology (AIC)
Wright-Patterson AFB OH 45433

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED/UNLIMITED SAME AS RPT. DTIC USERS	UNCLASSIFIED

22a NAME OF RESPONSIBLE INDIVIDUAL	22b TELEPHONE NUMBER (Include Area Code)	22c OFFICE SYMBOL
Matthew Kabrisky, Ph.D.	513-255-5276	AFIT/ENG

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

VITA

Lugene Shum was born in New York City, New York, on 20 April 1961. He graduated from Manhattan College, New York, in 1983. Upon graduation, he was commissioned as a Second Lieutenant through the AFROTC program and was assigned to the School of Engineering, Air Force Institute of Technology. Upon graduation from AFIT, he was assigned to Edwards Air Force Base.

Permanent Address: 50 Bayard St.
Apt. 2H
New York, New York 10013

```
        TYPE"VTHIN WORKING"                                ;      I/O       C
C.....................................................;.............C
        DO 1 ICNT=1,MAX                                    ;    REPEAT     C
                                                           ;PROCESS MAX    C
                                                           ;     TIMES     C
        CALL MTRXA                                         ;     FORM      C
                                                           ;   MATRIX A     C
        DO 5 II=IST,LST                                    ;               C
                                                           ;               C
           DO 10 JJ=2,255                                  ;               C
             CALL GPNT(IBUF3,II,JJ,IV)                     ;               C
             GOTO (81,82,83,84,85,86,87,88,10)IV           ;               C
                                                           ;               C
             GOTO 10                                       ;               C
                                                           ;               C
 81          CALL PPNT(IBUF,II,JJ,15)                      ;   FILL IN     C
             GOTO 10                                       ;   HOLES       C
                                                           ;               C
 82          CALL TT2(II,JJ)                               ;   TEST P2     C
             GOTO 10                                       ;               C
                                                           ;               C
 83          CALL TT3(II,JJ)                               ;   TEST P3     C
             CALL TDT3(II,JJ)                              ;   TEST D3     C
             GOTO 10                                       ;               C
                                                           ;               C
 84          CALL TT4(II,JJ)                               ;   TEST P4     C
             CALL TDT4(II,JJ)                              ;   TEST D4     C
             GOTO 10                                       ;               C
                                                           ;               C
 85          CALL TT5(II,JJ)                               ;   TEST P5     C
             CALL TDT5(II,JJ)                              ;   TEST D5     C
             GOTO 10                                       ;               C
                                                           ;               C
 86          CALL TT6(II,JJ)                               ;   TEST P6     C
             CALL TDT6(II,JJ)                              ;   TEST D6     C
             GOTO 10                                       ;               C
                                                           ;               C
 87          CALL TT7(II,JJ)                               ;   TEST P7     C
             GOTO 10                                       ;               C
                                                           ;               C
 88          CALL TT8(II,JJ)                               ;   TEST P8     C
 10        CONTINUE                                        ;               C
                                                           ;               C
 5       CONTINUE                                          ;               C
 1       CONTINUE                                          ;               C
C.....................................................;.............C
C                                                                  ;
 9999    CALL PICOUT(IBUF,INAME,IFLG)                              ;     PICBUF OUTPU
         CALL RELB(IBUF)                                           ;         BLOCK
         CALL RELB(IBUF2)                                          ;
         CALL RELB(IBUF3)                                          ;
C.....................................................;.....................
C
         STOP"<7>THIN FINISHED"
         END
```

```
C*****************************************************************C
C    PROGRAM: VTHIN.FR                LANGUAGE: FORTRAN V           C
C                                                                   C
C    WRITTEN: 15 DEC 84               SYSTEM: ECLIPSE               C
C                                                                   C
C    PURPOSE: THIS PROGRAM THINS A PATTERN TO PRODUCE A SKELETON.C
C                                                                   C
C         NOTE:1. THE INPUT PATTERN MUST BE BINARY (0/15 FOR        C
C                 THE OCTEK).                                       C
C              2. THE INPUT PATTERN MUST FIRST BE PRE-PROCESSED     C
C                 USING SPOTS.FR AND MEDIAN.FR.                     C
C              3. SOURCE OF ALGORITHM:                              C
C                   LINE THINNING ALGORITHM                         C
C                   G. FEIGIN AND N. BEN-YOSEF                      C
C                   PROCEEDINGS OF SPIE, VOL 397                    C
C*****************************************************************C
C                                                                   C
      COMMON IBUF(300)
      COMMON IBUF2(300)
      COMMON IBUF3(300)
      COMMON IARY(256)
      COMMON INAME(20)
      COMMON IFLAG
      COMMON IST
      COMMON LST
      REAL    WFILE(5)
      INTEGER MAX
C.....................................................................C
      ACCEPT"ENTER INPUT FILENAME: "    ;                             C
      READ(11,199)INAME(1)              ;        ENTER                C
199   FORMAT(S40)                       ;      FILENAME               C
C.....................................;.....................C
      ACCEPT"ENTER STARTING ROW: ",IST  ;      ENTER ROW    C
      ACCEPT"ENTER END ROW: ",LST       ;      POSITIONS    C
C.....................................;.....................C
C                                       ;   ENTER           C
C                                       ;  THICKNESS        C
      ACCEPT"ENTER NUMBER OF ITERATIONS: ",MAX ; OF CHARACTER C
C.....................................;.....................C
C                                     ;                     C
      WFILE="BK"                      ;                     C
      CALL PICFMT(IBUF,INAME,IFLG)    ;                     C
      CALL GFMT(IBUF,NROWS,NCOLS,NBITS,IMODE) ;             C
      CALL MAKB(IBUF)                 ;                     C
      CALL PICIN(IBUF,INAME,IFLG)     ;                     C
                                      ;                     C
      CALL PICFMT(IBUF2,WFILE,IFLG)   ;  PICBUF INPUT       C
      CALL GFMT(IBUF2,NROWS,NCOLS,NBITS,IMODE);    BLOCK    C
      CALL MAKB(IBUF2)                ;                     C
      CALL PICIN(IBUF2,WFILE,IFLG)    ;                     C
                                      ;                     C
      CALL PICFMT(IBUF3,WFILE,IFLG)   ;                     C
      CALL GFMT(IBUF3,NROWS,NCOLS,NBITS,IMODE);             C
      CALL MAKB(IBUF3)                ;                     C
      CALL PICIN(IBUF3,WFILE,IFLG)    ;                     C
```

```
C*****************************************************************C
C                                                                 C
C   ORIGINAL PROGRAM: VTOC.FR  BY R. SIMMONS                      C
C   ORIGINAL PROGRAM: UNPACK.FR  BY R. SIMMONS                    C
C                                                                 C
C   PROGRAM: VTC.FR                    LANGUAGE: FORTRAN V        C
C                                                                 C
C   WRITTEN: 15 DEC 84                 SYSTEM: ECLIPSE            C
C                                                                 C
C   PURPOSE: THIS PROGRAM CONVERTS A 256X256 PACKED VIDEO FILE    C
C            INTO A 256X256 UNPACKED COMPLEX VIDEO FILE.          C
C                                                                 C
C*****************************************************************C
C                                                                 C
      INTEGER IFLNM(7),OFLNM(7)
      INTEGER TEMP(512),IN(2048)
      COMPLEX OUT(2048)
C.................................................................C
C                                       ;                         C
      ACCEPT "ENTER INPUT FILENAME: "   ;                         C
      READ (11,199)IFLNM(1)             ;        ENTER            C
                                        ;    INPUT AND OUTPUT     C
      ACCEPT "ENTER OUTPUT FILENAME: "  ;       FILENAMES         C
      READ (11,199)OFLNM(1)             ;                         C
 199  FORMAT (S40)                      ;                         C
C.....................................;.........................C
C                                       ;                         C
      OPEN 1,IFLNM                      ;                         C
                                        ;          OPEN           C
      CALL DFILW (OFLNM,IER)            ;    INPUT AND OUTPUT     C
      IF(IER.NE.1)GOTO 90               ;          FLIES          C
 90   CALL CFILW(OFLNM,2,KER)           ;                         C
      CALL CHECK(KER)                   ;                         C
      OPEN 2,OFLNM,ATT="OR",ERR=100     ;                         C
C.....................................;.........................C
C                                           ;                     C
 100  DO 5 I=0,31                           ;                     C
                                            ;                     C
        CALL RDBLK(1,2*I,TEMP,2,IER)        ;                     C
        CALL CHECK(IER)                     ;                     C
        CALL UNPACK(512,TEMP,IN)            ;                     C
                                            ;                     C
        DO 3 J=1,2048                       ; CONVERT INTEGER     C
             OUT(J)=CMPLX(FLOAT(IN(J)),0.0) ; INTO COMPLEX        C
 3      CONTINUE                            ;  (A + j 0.0)        C
                                            ;                     C
        CALL WRBLK(2,32*I,OUT,32,IER)       ;                     C
        CALL CHECK(IER)                     ;                     C
                                            ;                     C
 5    CONTINUE                              ;                     C
C.....................................;.........................C
C                                                                 C
      CALL RESET
      STOP "<7> FINISHED"
      END
```

```
        GOTO 94

100     CALL PPNT(IBUF,II,JJ,15)

94      RETURN
        END
```

```
C*****************************************************************C
C    PROGRAM: TT8.FR              LANGUAGE: FORTRAN V                C
C                                                                    C
C    WRITTEN: 15 DEC 84           SYSTEM: ECLIPSE                    C
C                                                                    C
C    PURPOSE: THIS IS A SUBROUTINE USED BY THIN.FR TO TEST FOR      C
C             P8 DELETION.                                           C
C*****************************************************************C
C                                                                    C
         SUBROUTINE TT8(II,JJ)
C                                                                    C
         COMMON IBUF(300)
         COMMON IBUF2(300)
         COMMON IBUF3(300)
         COMMON IARY(256)
         COMMON INAME(20)
         COMMON IFLAG
         COMMON IST
         COMMON LST
         INTEGER IVAL(9)
C                                                                    C
         DO 5 I=1,9
              IVAL(I)=0
 5       CONTINUE

         CALL GPNT(IBUF3,II-1,JJ,IVAL(1))
         CALL GPNT(IBUF3,II,JJ-1,IVAL(2))
         CALL GPNT(IBUF3,II,JJ+1,IVAL(3))
         CALL GPNT(IBUF3,II+1,JJ,IVAL(4))

         CALL GPNT(IBUF3,II-1,JJ-1,IVAL(5))
         CALL GPNT(IBUF3,II-1,JJ+1,IVAL(6))
         CALL GPNT(IBUF3,II+1,JJ-1,IVAL(7))
         CALL GPNT(IBUF3,II+1,JJ+1,IVAL(8))

         IF(IVAL(1).NE.7) GOTO 91
         IF(IVAL(2).NE.7) GOTO 91
         IF((IVAL(6).EQ.9) .OR. (IVAL(3).EQ.9) .OR. (IVAL(8).EQ.9)
     /    .OR. (IVAL(4).EQ.9) .OR. (IVAL(7).EQ.9))GOTO 100

 91      IF(IVAL(1).NE.7) GOTO 92
         IF(IVAL(3).NE.7) GOTO 92
         IF((IVAL(5).EQ.9).OR.(IVAL(2).EQ.9) .OR. (IVAL(7).EQ.9)
     '    .OR. (IVAL(4).EQ.9).OR.(IVAL(8).EQ.9))GOTO 100

 92      IF(IVAL(4).NE.7) GOTO 93
         IF(IVAL(2).NE.7) GOTO 93
         IF((IVAL(5).EQ.9).OR.(IVAL(1).EQ.9) .OR. (IVAL(6).EQ.9)
     /    .OR. (IVAL(3).EQ.9).OR.(IVAL(8).EQ.9))GOTO 100

 93      IF(IVAL(4).NE.7) GOTO 94
         IF(IVAL(3).NE.7) GOTO 94
         IF((IVAL(7).EQ.9).OR.(IVAL(2).EQ.9) .OR. (IVAL(5).EQ.9)
     /    .OR. (IVAL(1).EQ.9).OR.(IVAL(6).EQ.9))GOTO 100
```

```
93      IF(IVAL(4).NE.9) GOTO 94
        IF(IVAL(1).NE.0) GOTO 94
        IF(((IVAL(6).EQ.0).OR.(IVAL(3).NE.0)) .AND.
     /  ((IVAL(5).EQ.0).OR.(IVAL(2).NE.0)))GOTO 100

94      IF(IVAL(2).NE.8) GOTO 95
        IF(IVAL(7).NE.7) GOTO 95
        IF((IVAL(5).EQ.9).OR.(IVAL(1).EQ.9))GOTO 100

95      IF(IVAL(3).NE.8) GOTO 96
        IF(IVAL(8).NE.7) GOTO 96
        IF((IVAL(1).EQ.9).OR.(IVAL(6).EQ.9))GOTO 100

96      IF(IVAL(2).NE.8) GOTO 97
        IF(IVAL(5).NE.7) GOTO 97
        IF((IVAL(7).EQ.9).OR.(IVAL(4).EQ.9))GOTO 100


97      IF(IVAL(3).NE.8) GOTO 98
        IF(IVAL(6).NE.7) GOTO 98
        IF((IVAL(4).EQ.9).OR.(IVAL(8).EQ.9))GOTO 100


        GOTO 98


100     CALL PPNT(IBUF,II,JJ,15)

98      RETURN
        END
```

```
C*****************************************************************C
C                                                                 C
C                                                                 C
C    PROGRAM: TT7.FR              LANGUAGE: FORTRAN V             C
C                                                                 C
C    WRITTEN: 15 DEC 84           SYSTEM: ECLIPSE                 C
C                                                                 C
C    PURPOSE: THIS IS A SUBROUTINE USED BY THIN.FR TO TEST FOR    C
C             P8 DELETION.                                        C
C                                                                 C
C                                                                 C
C*****************************************************************C
C                                                                 C
      SUBROUTINE TT7(II,JJ)
C                                                                 C
      COMMON IBUF(300)
      COMMON IBUF2(300)
      COMMON IBUF3(300)
      COMMON IARY(256)
      COMMON INAME(20)
      COMMON IFLAG
      COMMON IST
      COMMON LST
      INTEGER IVAL(9)
C                                                                 C
      DO 5 I=1,9
         IVAL(I)=0
 5    CONTINUE

      CALL GPNT(IBUF3,II-1,JJ,IVAL(1))
      CALL GPNT(IBUF3,II,JJ-1,IVAL(2))
      CALL GPNT(IBUF3,II,JJ+1,IVAL(3))
      CALL GPNT(IBUF3,II+1,JJ,IVAL(4))

      CALL GPNT(IBUF3,II-1,JJ-1,IVAL(5))
      CALL GPNT(IBUF3,II-1,JJ+1,IVAL(6))
      CALL GPNT(IBUF3,II+1,JJ-1,IVAL(7))
      CALL GPNT(IBUF3,II+1,JJ+1,IVAL(8))

      IF(IVAL(1).NE.9) GOTO 91
      IF(IVAL(4).NE.0) GOTO 91
      IF(((IVAL(7).EQ.0).OR.(IVAL(2).NE.0)) .AND.
     /  ((IVAL(8).EQ.0).OR.(IVAL(3).NE.0)))GOTO 100


91    IF(IVAL(2).NE.9) GOTO 92
      IF(IVAL(3).NE.0) GOTO 92
      IF(((IVAL(8).EQ.0).OR.(IVAL(4).NE.0)) .AND.
     /  ((IVAL(6).EQ.0).OR.(IVAL(1).NE.0)))GOTO 100


92    IF(IVAL(3).NE.9) GOTO 93
      IF(IVAL(2).NE.0) GOTO 93
      IF(((IVAL(5).EQ.0).OR.(IVAL(1).NE.0)) .AND.
     /  ((IVAL(7).EQ.0).OR.(IVAL(4).NE.0)))GOTO 100
```

```
C*****************************************************************C
C    PROGRAM: TT6.FR              LANGUAGE: FORTRAN V                C
C                                                                    C
C    WRITTEN: 15 DEC 84           SYSTEM: ECLIPSE                    C
C                                                                    C
C    PURPOSE: THIS IS A SUBROUTINE USED BY THIN.FR TO TEST FOR      C
C             P7 AND NEIGHBOR VALUE OF 9 DELETION.                   C
C*****************************************************************C
C                                                                    C
         SUBROUTINE TT6(II,JJ)
C                                                                    C
         COMMON IBUF(300)
         COMMON IBUF2(300)
         COMMON IBUF3(300)
         COMMON IARY(256)
         COMMON INAME(20)
         COMMON IFLAG
         COMMON IST
         COMMON LST
         INTEGER IVAL(9)
         INTEGER T(3)
C                                                                    C
         DO 5 I=1,9
              IVAL(I)=0
 5       CONTINUE

         CALL GPNT(IBUF3,II-1,JJ,IVAL(1))
         CALL GPNT(IBUF3,II,JJ-1,IVAL(2))
         CALL GPNT(IBUF3,II,JJ+1,IVAL(3))
         CALL GPNT(IBUF3,II+1,JJ,IVAL(4))

         DO 7 I=1,4
              IF(IVAL(I).EQ.9)GOTO 100
 7       CONTINUE
C
C        CALL GROW(IBUF3,II+1,JJ-1,3,T)
C        IVAL(5)=T(1)+T(2)+T(3)
C        CALL GROW(IBUF3,II-1,JJ-1,3,T)
C        IVAL(8)=T(1)+T(2)+T(3)
C        CALL GPNT(IBUF3,II-1,JJ+1,T(1))
C        CALL GPNT(IBUF3,II+1,JJ+1,T(2))
C        IVAL(6)=T(1)+T(2)+IVAL(3)
C        CALL GPNT(IBUF3,II-1,JJ-1,T(1))
C        CALL GPNT(IBUF3,II+1,JJ-1,T(2))
C        IVAL(7)=T(1)+T(2)+IVAL(2)
C        DO 10 I=1,4
C             IF(IVAL(I).LE.5) GOTO 10
C             IF((IVAL(I).EQ.6) .OR. (IVAL(I).EQ.8)) GOTO 10
C             IF(IVAL(I+4).EQ.0) GOTO 100
C10      CONTINUE
         GOTO 91
 100     CALL PPNT(IBUF,II,JJ,15)
 91      RETURN
         END
```

```
82      IF(IVAL(3).NE.5)GOTO 83
        IF((IVAL(1).EQ.0) .OR. (IVAL(5).EQ.0))GOTO 83
        IF(IVAL(7).EQ.0)GOTO 100

83      IF(IVAL(4).NE.5)GOTO 84
        IF((IVAL(3).EQ.0) .OR. (IVAL(6).EQ.0))GOTO 84
        IF(IVAL(5).EQ.0)GOTO 100

84      IVAL(9)=IVAL(4)+IVAL(7)+IVAL(8)
        IVAL(10)=IVAL(3)+IVAL(6)+IVAL(8)
        IVAL(11)=IVAL(2)+IVAL(5)+IVAL(7)
        IVAL(12)=IVAL(1)+IVAL(5)+IVAL(6)


        DO 10 I=1,4
              IF((IVAL(I).LT.5) .OR. (IVAL(I).GT.7))GOTO 10
              IF(IVAL(I+8).EQ.0)GOTO 100
10      CONTINUE


      GOTO 91

100     CALL PPNT(IBUF,II,JJ,15)

91      RETURN
        END
```

```
C*****************************************************************C
C                                                                 C
C                                                                 C
C    PROGRAM: TT5.FR              LANGUAGE: FORTRAN V             C
C                                                                 C
C    WRITTEN: 15 DEC 84           SYSTEM: ECLIPSE                 C
C                                                                 C
C    PURPOSE: THIS IS A SUBROUTINE USED BY THIN.FR TO TEST FOR    C
C             P5, P6, AND NEIGHBOR VALUE OF 8 DELETION.           C
C                                                                 C
C                                                                 C
C*****************************************************************C
C                                                                 C
      SUBROUTINE TT5(II,JJ)
C                                                                 C
      COMMON IBUF(300)
      COMMON IBUF2(300)
      COMMON IBUF3(300)
      COMMON IARY(256)
      COMMON INAME(20)
      COMMON IFLAG
      COMMON IST
      COMMON LST
      INTEGER IVAL(12)
C
      DO 5 I=1,9
          IVAL(I)=0
 5    CONTINUE

     CALL GPNT(IBUF3,II-1,JJ,IVAL(1))
     CALL GPNT(IBUF3,II,JJ-1,IVAL(2))
     CALL GPNT(IBUF3,II,JJ+1,IVAL(3))
     CALL GPNT(IBUF3,II+1,JJ,IVAL(4))


     DO 7 I=1,4
          IF(IVAL(I).EQ.8) GOTO 100
 7   CONTINUE


      CALL GPNT(IBUF3,II-1,JJ-1,IVAL(5))
      CALL GPNT(IBUF3,II-1,JJ+1,IVAL(6))
      CALL GPNT(IBUF3,II+1,JJ-1,IVAL(7))
      CALL GPNT(IBUF3,II+1,JJ+1,IVAL(8))


      IF(IVAL(1).NE.5)GOTO 81
      IF((IVAL(2).EQ.0) .OR. (IVAL(7).EQ.0))GOTO 81
      IF(IVAL(8).EQ.0)GOTO 100


 81   IF(IVAL(2).NE.5)GOTO 82
      IF((IVAL(4).EQ.0) .OR. (IVAL(8).EQ.0))GOTO 82
      IF(IVAL(6).EQ.0)GOTO 100
```

```
C*****************************************************************C
C                                                                 C
C                                                                 C
C    PROGRAM: TT4.FR                LANGUAGE: FORTRAN V           C
C                                                                 C
C    WRITTEN: 15 DEC 84             SYSTEM: ECLIPSE               C
C                                                                 C
C    PURPOSE: THIS IS A SUBROUTINE USED BY THIN.FR TO TEST FOR    C
C             P2, P3, P4 DELETION.                                C
C                                                                 C
C                                                                 C
C*****************************************************************C
C                                                                 C
      SUBROUTINE TT4(II,JJ)
C                                                                 C
      COMMON IBUF(300)
      COMMON IBUF2(300)
      COMMON IBUF3(300)
      COMMON IARY(256)
      COMMON INAME(20)
      COMMON IFLAG
      COMMON IST
      COMMON LST
      INTEGER IVAL(9)
C
      DO 5 I=1,9
           IVAL(I)=0
 5    CONTINUE

      CALL GPNT(IBUF3,II-1,JJ,IVAL(1))
      CALL GPNT(IBUF3,II,JJ-1,IVAL(2))
      CALL GPNT(IBUF3,II,JJ+1,IVAL(3))
      CALL GPNT(IBUF3,II+1,JJ,IVAL(4))

      IF((IVAL(2).EQ.0) .OR. (IVAL(3).EQ.0))GOTO 91
      IF((IVAL(1).GE.4) .AND. (IVAL(1).LE.7))GOTO 100
      IF((IVAL(4).GE.4) .AND. (IVAL(4).LE.7))GOTO 100

 91   IF((IVAL(1).EQ.0) .OR. (IVAL(4).EQ.0))GOTO 92
      IF((IVAL(2).GE.4) .AND. (IVAL(2).LE.7))GOTO 100
      IF((IVAL(3).GE.4) .AND. (IVAL(3).LE.7))GOTO 100

 92   IF((IVAL(2).NE.0) .OR. (IVAL(3).NE.0))GOTO 93
      IF((IVAL(1).EQ.7) .OR. (IVAL(4).EQ.7))GOTO 100

 93   IF((IVAL(1).NE.0) .OR. (IVAL(4).NE.0))GOTO 94
      IF((IVAL(2).EQ.7) .OR. (IVAL(3).EQ.7))GOTO 100
      GOTO 94

 100  CALL PPNT(IBUF,II,JJ,15)

 94   RETURN
      END
```

END

FILMED

5-85

DTIC