AD-A118 076 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 20/6
COMBINED SPATIAL FILTERING AND BOOLEAN OPERATORS APPLIED TO THE--ETC(U)
JUN 82 B E FELTMATE
UNCLASSIFIED AFIT/GCS/EE/82J-8 NL

1 OF 4
AD
A118076

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY (ATC)

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

[illegible]

(1)

AFIT/GCS/EE/82J-8

COMBINED SPATIAL FILTERING AND
BOOLEAN OPERATORS APPLIED TO THE
PROCESSING OF REAL IMAGES

THESIS

AFIT/GCS/EE/82J-8 Blaine Feltmate
Captain
Canadian Armed Forces

AFIT/GCS/EE/82J-8

COMBINED SPATIAL FILTERING AND BOOLEAN OPERATORS APPLIED TO THE PROCESSING OF REAL IMAGES

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

by

Blaine E. Feltmate, B.Eng.

Captain Canadian Armed Forces

Graduate Computer Systems

June 1982

Approved for public release; distribution unlimited.

To My Parents

Acknowledgements

I acknowledge with sincere gratitude the help and guidance I received from my advisor, Dr. Matthew Kabrisky. I am especially grateful to him for the enthusiastic encouragement which he provided throughout the various stages of research that led to accomplishing this thesis.

I would also like to thank Dr T. Hartrum and Major L. Kizer for their time and efforts in reviewing this manuscript and providing many helpful suggestions on the presentation of this work.

I am also indebted to Patricia Sheehan and Joan Tomme for their volunteered time and excellent assistance in the preparation of the large appendices.

Finally, I wish to thank the Professors and Officers of the United States Air Force Institute of Technology for the consistent professionalism and friendship which they showed to me during my course of study in the United States.

Contents

List of Figures

Abstract

Several new and seemingly successful scene analysis techniques for application to 'real' image processing are presented and discussed. These techniques consist of particular combinations of spatial low pass filtering, global thresholding and Boolean operators, specifically the 'AND', 'OR' and 'NOT' operators. These combinatorial operators, hereafter referred to as the "Boolpass" operators, perform the task of picture energy/information reduction, while retaining the fundamental picture primitives such as edges which characterize the images.

Over 150 figures are included which illustrate the results obtained from application of the Boolpass technique to eight different natural scenes. These results indicate that the Boolpass operators do display great potential as important components of a larger more comprehensive pattern recognition machine. Such a machine would encompass further processing (for target classification/recognition) of the resulting Boolpass operator information.

COMBINED SPATIAL FILTERING AND BOOLEAN OPERATORS APPLIED TO THE PROCESSING OF REAL IMAGES

I. INTRODUCTION

Background

Over the past two decades there has been an ever increasing need to design and build machines capable of performing tasks which until now have been performed only by humans. With the large and ever increasing demands being placed on mechanization and information processing, automation of such processes has become an intense and challenging area of current research.

Attempts at modelling human functions and intelligence have lead to many new and diverse fields of study. One such field is Automatic Pattern Recognition and Image/Picture Processing. The importance and emphasis that has been placed on this field of study has been generated because, potentially, it has the capability of revolutionizing the nature of information processing in large areas of scientific, industrial, social, and military activities [Ref. 4]. At present pattern recognition and image processing have been utilized in such areas as: Commercial/Government - for automatic detection of flaws, impurities, etc.; natural resource identification; object recognition and parts handling; character recognition; and automatic processing of documents;

Medical Applications - identification, detection/diagnosis in X-Ray photographs, electorcardiogram and electroencephalogram waveforms; and

Military applications - target detection; interpretation of aerial reconnaissance imagery; guidance applications [Ref. 6]

Further examples could be cited. However, those given should clearly indicate the many diverse and important application areas of pattern recognition research. Advances in computer technology have only recently provided the more powerful hardware and software tools that will be required to find possible solutions to the automatic pattern recognition problem.

One of the major and most difficult fields of contemporary pattern recognition and scene analysis research is in the area of real image processing. 'Real Image' is used here to mean real-world scenes as opposed to modelled scenes or laboratory manufactured scenes which are quasi 'representative' of the real world. Such modelled or representative images have unfortunately proven in the past to often oversimplify the pattern recognition problem, leading to algorithms with limited or no capabilities when applied to the 'real world' [Ref. 1].

Problem and Scope

Scene analysis and pattern recognition techniques for real world object recognition have received a great deal of research and development effort with varying degrees of success [Ref. 1, 2, 4, 8, 9, 11]. A primary objective of scene analysis has been the development of methods to reduce the very large quantity of picture data to suitably relevent feature information. These relevent features may consist, for example, of line segments, edges, regions or other features, any or all of which can be used to represent the scene structure [Ref 1]. The purpose of this thesis was to use a hierarchial approach to the investigation and development of digital operators which would facilitate the extraction of relevent two-dimensional scene structures from digitized real images. The operators had to demonstrate picture independence in their application and no *a priori* knowledge of the picture could be assumed.

"In image pattern recognition one of the most difficult steps is to extract objects from an irrelevant background. The degree of difficulty involved varies greatly with the quality of the picture and the nature of the object." [Ref 3]

In this thesis, several new and seemingly successful scene analysis techniques, for application to 'real' image processing, are presented and discussed. These techniques consist of particular combinations of spatial low pass filtering, global thresholding and Boolean operators, specifically the 'AND', 'OR' and 'NOT' operators. These combinatorial operators were specifically chosen so that their simplicity and speed of operation would allow their adaptation to a real time processing environment via hardware implementation. These combinatorial operators, hereafter referred to as the " Boolpass " operators (explained in Section III), are not to be interpreted as independent pattern recognition operators.

The task of the Boolpass operators is one of picture information reduction in real images while retaining the fundamental picture primatives such as edges which characterize the images. The Boolpass operators do display great potential as important components of a larger more comprehensive pattern recognition machine. Such a machine would encompass further processing of the resulting Boolpass operator information. Further processing could include template matching, moment analysis, size and range discrimination operators, or other relevent operators in order to achieve object(s) recognition.

General Approach

In order for a computer to be able to work with a picture (or image), the picture must first be transformed into a 'digital' image and placed on a suitable storage medium. This task was accomplished in the Air Force

Institute of Technology (AFIT) Signal Processing Laboratory by use of a video digitizer. The digitizer system consisted of a vidicon television camera, an analog to digital converter to convert the signals obtained from the vidicon camera, and a magnetic disk storage unit to store the resulting digital picture array for subsequent processing. The array representing the digitized picture consisted of 256 by 256 'pixel' elements. Each pixel element was a number from Ø to 15 which represented the grey level or brightness of a very small region in the scanned image. In this manner each pixel was represented in the computer by a four bit integer value.

The video digitizer was used to digitize a selection of real scene images which were then stored on disk for further processing. Significant additional noise was introduced into the digitized video images from both the A/D converter and the camera itself. This additional noise was characterized by inconsistent or spurious pixel value information randomly distributed throughout the digitized image. Ambient light conditions also affected the quality of the digitized image. In order to improve the signal to noise ratio of the digitized picture, a series of seven digitizations of each picture were made for each photograph being analyzed. These seven pictures were then averaged , pixel by pixel over the seven pictures, and the resultant image stored as the initial image for processing.

The overall development of the 'Boolpass operators' consisted of: (1) the selection of a simple low pass spatial filter which could be applied 'locally', (as opposed to a global filter); (2) the selection of filter or 'mask' sizes to be applied to the picture; (3) a method of thresholding; (4) the retention of the original pixel values throughout the processing (as opposed to a binary 'black and white' representation after processing); (5) and an iterative approach to examining the potential of various combinations of low pass filtering using Boolean AND operators and of particular importance - both POSITIVE and NEGATIVE images.

A series of interactive computer programs were developed for processing the digitized images. These programs allowed the user to very quickly and easily manipulate the manner and parameters by which a picture was being processed. In this way the hierarchical approach to scene analysis was accelerated and the use of interactive programs proved to be a significant contributing factor in the development of the work reported in this thesis. A detailed description of the function of the computer programs used, along with a Pascal listing for each, can be found in Appendix D.

Sequence of Presentation

Section II of this thesis provides a brief review and discussion of some of the major sources of noise and distortion which occur in a digitized image as a result of the digitization process. The purpose of this discussion is to provide the reader with a better understanding of some of the complexities and difficulties encountered when working with digitized images, and in particular 'natural' or 'real' digitized images.

A detailed explanation of the type and functioning of the operators used in the thesis is given in Section III. The operators described in this section include low pass filtering, the Boolpass operators, and thresholding. The results obtained from processing the digitized pictures by applying specific combinations of these operators appear in Section IV, Appendix A, B and C. Both aerial and ground scenes were considered and used for the operator testing, and examples of each appear in these sections.

Conclusions and recommendations for future research are discussed in Sect:on V. Appendix D contains the listings and documentation for the interactive computer programs used.

II. PROBLEMS ASSOCIATED WITH REAL DIGITAL IMAGE PROCESSING

Noise And Distortion In Digitized Pictures

Noise in digitized images produces variations in both the intensity and position of the pixel points. This noise, which is normally considered undesirable, is introduced into the digitized picture from several sources.

The first source to be considered is the picture, itself, which is being digitized. Noise can already exist in a picture image or photograph due to poor picture quality, processing or reproduction. Improper lighting can produce distortion and improper film speed can produce a 'grainy' or blurred effect on a photograph. These and other effects introduce 'noise' into the picture and detract from the ideal: a clear, crisp image desired for processing.

The digitizer is most frequently the major source of noise in the image processing system. The digitizer consists of five main elements: (1) the sampling aperature, which accesses the individual pixel areas; (2) the scanning mechanism, which sequentially moves the sampling aperature over the image; (3) the sensor, which measures the brightness of each pixel (a linear response is highly desirable here); (4) an analog to digital converter to quantize the output of the sensor; and (5) an output medium for storage and/or display. (If required, a detailed description and discussion of each can be found in Ref. 2.)

Each of the above elements in a digitizer contributes to the amount of noise that will be added to the digitized image. Total elimination of this noise is not physically possible. Therefore an image initially digitized by such a system is inevitably degraded before the actual processing even begins.

"If a uniformly gray image is presented to a digitizer, the noise inherent in the system will cause variations in the output even though the input is constant. Noise introduced by the digitizer is a source of image degradation and should be small relative to the contrast of the image." [Ref. 2]

In natural or real scenes the contrast of the image and the amount of energy in the image can vary widely. In such pictures, image processing becomes an extremely difficult task as the additional presence of noise can effectively hide or distort relevent boundary, edge, and texture information which was present in the original scene. It is believed that these fundamental primatives - boundary, edge, texture and color information are possible keys to the solution of the pattern recognition problem [Ref. 9]. Finding operators which can be applied universally to natural images, where such key features as micro and macro edges are possibly lost or distorted, creates one of the most complex and as yet unsolved problem areas in image processing. (A micro edge, for example, could be defined as two adjacent points having different grey levels, while a macro edge might exist between two large coarsely textured regions having different average pixel greylevels [Ref 10].)

Examples of digitized pictures using the AFIT Vidicon system are shown in figures 1 thru 3. In these examples, the existance of edge noise, distortion and discontinuities are clearly visible. However, the fact that the human visual system can still readily discern distinct objects in these pictures, despite this noise, leads one to believe that object extraction by a machine with this same picture information should still be possible.

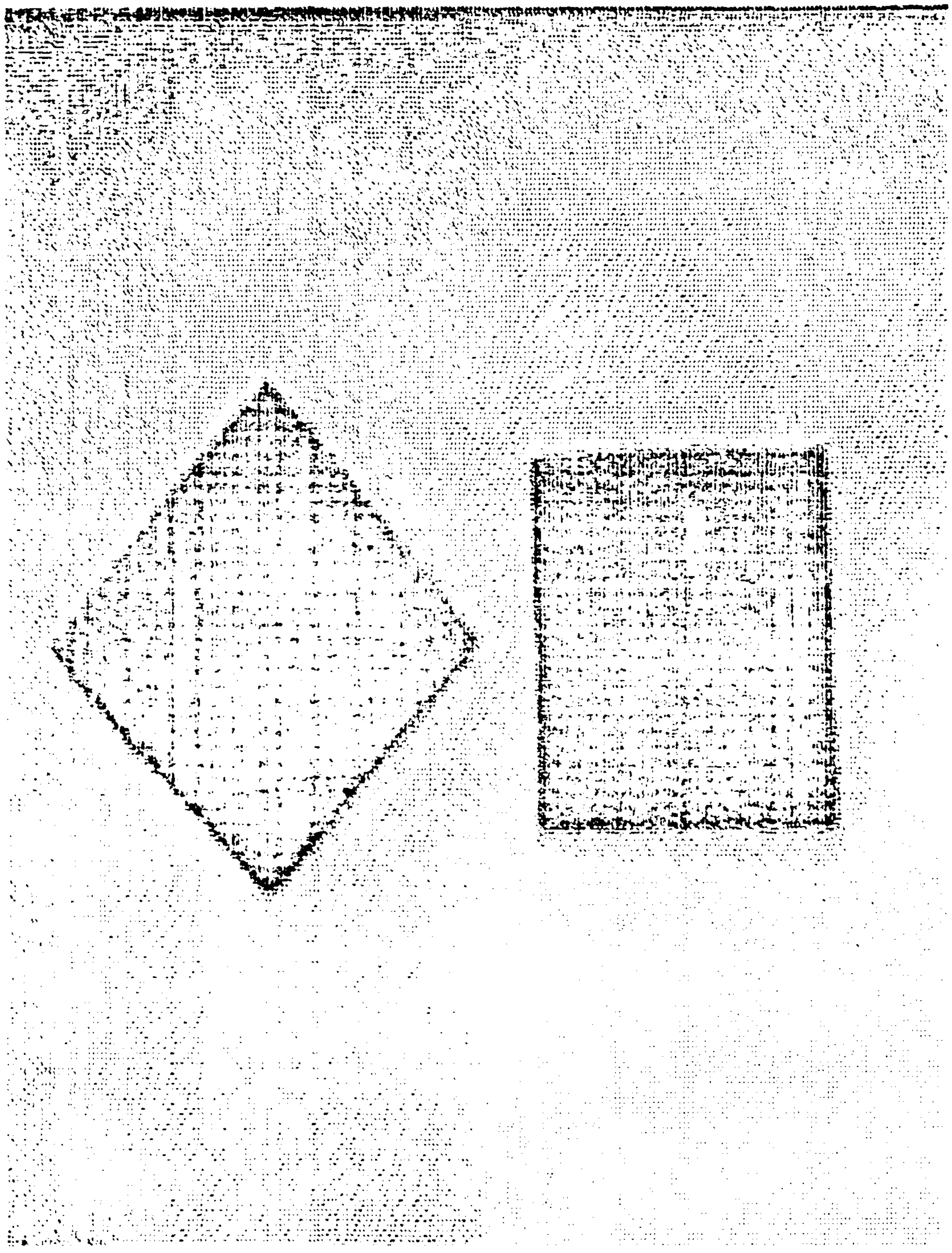

Fig. 1. Digitized Image of Black Blocks on a White Backgroud (Note 'Noise' and Distortion Introduced By Digitization Process)

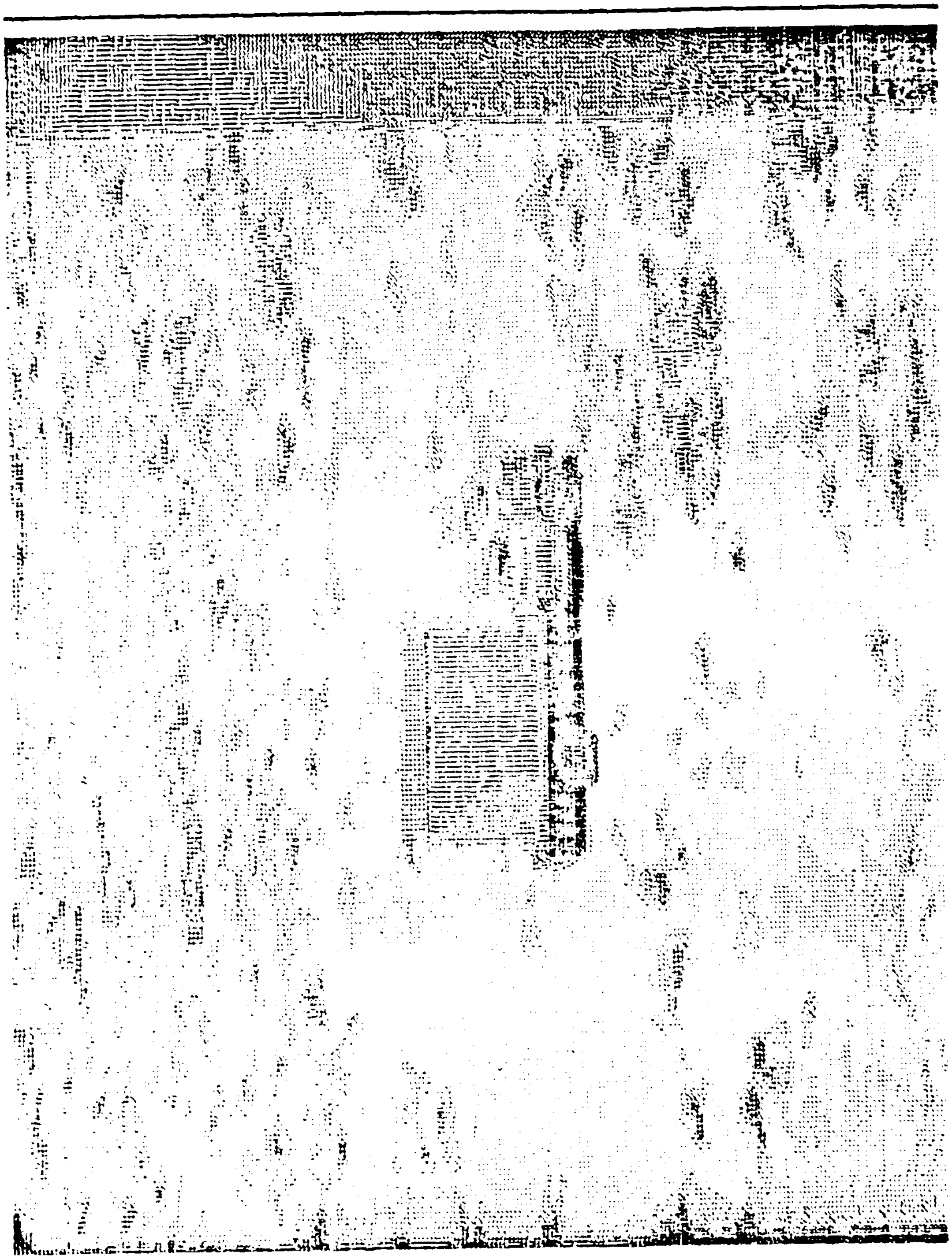

Fig. 2. Digitized Image of Truck in Desert

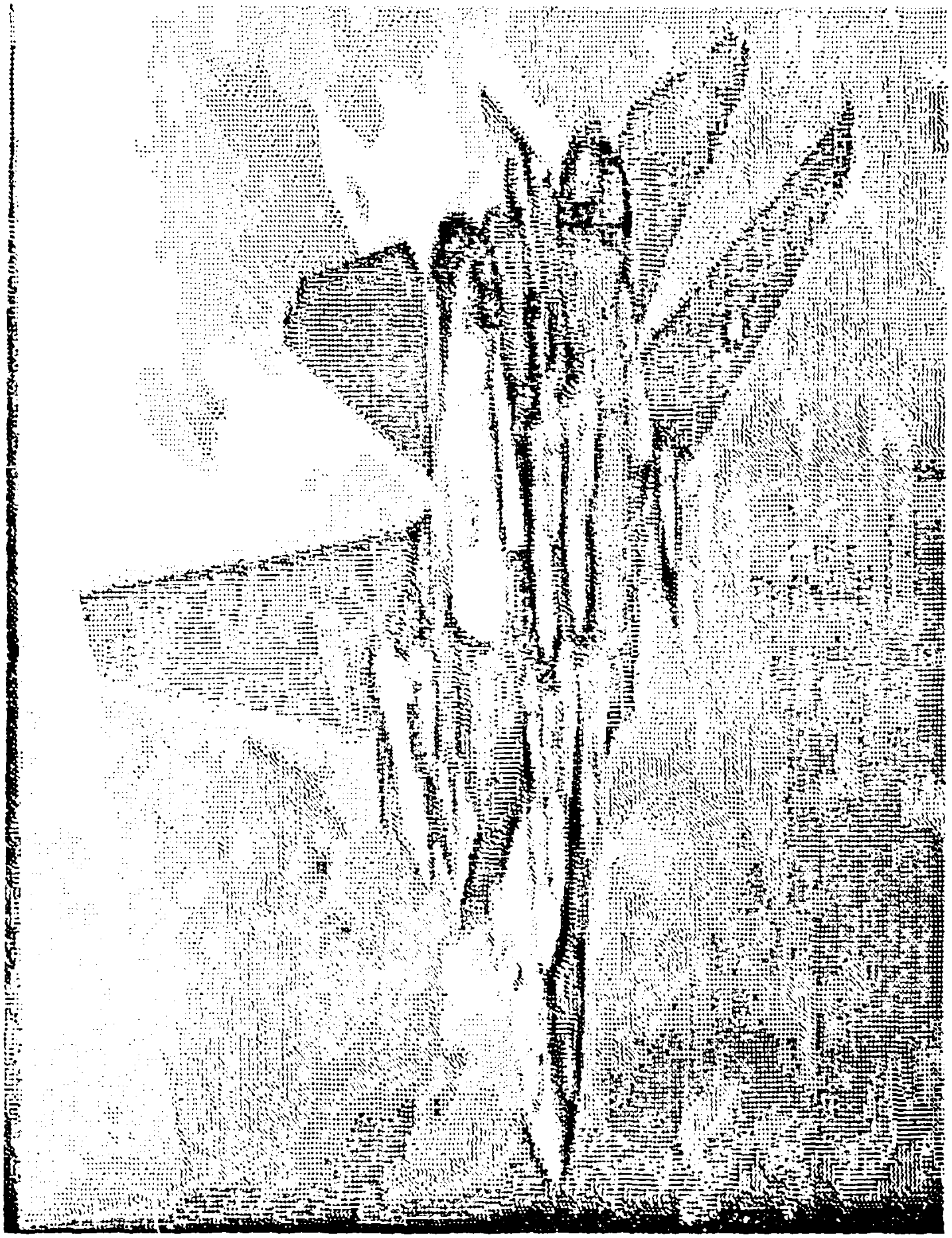

Fig. 3. Digitized Image of United States Navy F14 Tomcat

III. OPERATOR DESCRIPTIONS

Low Pass Filter

A noise smoothing procedure often applied to a digitized picture, before feature or edge extraction, is low pass filtering [Ref. 1]. This type of filtering is particularly useful when applied to real images where the edges frequently suffer from noise or some source of discontinuity. Examples of this were previously illustrated in Section II (figures 1 thru 3).

Low Pass Filtering directly affects the gradient of a picture by smoothing out fast gradients and reinforcing slow gradients. The gradient magnitude operator basically represents the steepness of slope at every point. In areas of steep slope, such as at edges, the gradient takes on large values, while low gradient values result in areas of nearly uniform grey level distribution. Application of the low pass filter results in a blurred image and suppression of isolated point noise sources.

Two low pass filters were used in this thesis. Each filter was computed by a simple averaging of the image points for a given mask size. The filter 'mask' refers to the matrix of pixels being processed by the filter.

The first of the low pass filters used was a two dimensional(2-D) filter which operated in both the vertical and horizontal axis of the picture plane. The second low pass filter was one - dimensional (1-D) and operated only in the horizontal axis of the picture. Five different mask sizes were used and tested for each filter, namely - 3, 5, 7, 9 and 11 pixels. These numbers, for the two-dimensional filter, represent the number of pixels along one edge of a square matrix. As an example, consider image points arranged as follows for a mask of size 3 (or 3x3 mask):

$$\begin{matrix} A & B & C \\ D & X & E \\ F & G & H \end{matrix}$$

The result of the two-dimensional Low Pass Filter (LPF) averaging operation at point X would be:

$$\text{LPF at } X = (A + B + C + D + X + E + F + G + H)/9$$

Using this same example for the one-dimensional low pass filter for a mask size of 3 (or 1x3 mask):

$$\text{LPF at } X = (D + X + E)/3$$

It can be seen that the mask size specified for a one-dimensional low pass filter represents an array symmetric about the point being processed (eg. X). The LPF at pixel X (the matrix center) for the remaining matrix sizes - 5, 7, 9 and 11, were computed in a similar manner.

In this thesis, the Low Pass Filter was applied in the image domain. That is, the LPF operator was applied to each pixel of the original digitized image (i.e. the image domain) as opposed to the transform domain. The transform domain results when one or more operations are performed on the original image, producing an already processed or 'transformed' image.

Many variations and more complex Low Pass Filters are possible. However, the LPF explained for use in this thesis was chosen for its ease of implementation and speed of operation on almost any computer system. Examples of the filter as applied to real images can be found in Section IV RESULTS.

Thresholding

Thresholding was used in conjunction with the Boolpass operators to specify the acceptance/rejection limits for any one particular operation during the processing of the image. The thresholding procedure used here with the Boolpass operators permitted retention only of those gray level values within a certain window of values, determined for each individual pixel in the image being processed. All other values were set to grey level 15 which equals a blank (white).

Threshold values were based on a calculation of the global mean and standard deviation of the pixel values for the entire digitized picture.

That is:

Mean pixel value = sum of pixel grey level values for entire picture divided by the total number of pixels

Standard Deviation = Square root of the average of the squared deviations from the mean

A percentage of the standard deviation was then used to determine the minimum and maximum threshold limits. (Mathematical formulae for these common mean and standard deviation computations are available in the CRC Standard Mathematical Tables(24th Edition), page 473.)

The choice of using a global approach to determine the limits for thresholding, as opposed to a local or locally adaptive threshold technique [Ref. 9] was initially an arbitrary decision based on computational simplicity and speed of operation. However, once the threshold limits were calculated they were applied to each individual pixel so as to determine a threshold window, or window of acceptance, for that particular pixel (grey level). Subsequent processing using this combination of global threshold limits together with a locally determined threshold window for each pixel gray level, yielded positive results while requiring only simple, fast calculations. The following example will help illustrate how, within the interactive programs, the threshold limits and windows were calculated.

When an image was to be processed, the user was prompted at the computer terminal to enter a decimal number between Ø and 1. The standard deviation, having previously been computed for the picture, was then multiplied by this decimal number. The result of this multiplication was then rounded off to an integer value. Plus or minus this integer value then gave the threshold limits (or global threshold limits). This calculation was performed only once, at the start of processing of each picture.

The threshold limits were then used at each pixel processed in the image to calculate a threshold range or threshold window for that particular pixel. Very simply, the threshold limits (range or window) at each pixel was determined by addition and subtraction of the (global) threshold limit to the gray level value of the pixel.

For example:

given a digitized picture 256 x 256 pixels (16 greylevels)

mean of pixel values = 8
standard deviation = 3
decimal number entered by user = .4

THRESHOLD LIMITS = - round(.4 x 3), + round(.4 X 3)

THRESHOLD WINDOW at pixel P = P - round(.4 x 3) ,
P + round(.4 x 3)

Solving:

THRESHOLD WINDOW (min, max) at pixel P = P - 1, P + 1

if P = 10, then

THRESHOLD WINDOW (min, max) = 9, 11

Therefore, if a Boolpass operation is performed using the above threshold window, only pixel values of 9, 10, and 11, at pixel P = 10, will be retained (or otherwise operated upon) while all other values will be set to gray level 15.

The coefficient chosen to establish the threshold limits has a significant impact on the results which are received from applying the Boolpass operators. Examples of various threshold limits used with the Boolpass operators are demonstrated in Section IV - results.

Boolpass Operators and Their Development

"Boolpass" was chosen as an acronym for the combinatorial operators or picture processing techniques that evolved from the research and testing in this thesis. "Boolpass" was based on a contraction of the two principle operations used for the image processing - namely, BOOLean operations and Low PASS filtering operations.

The Boolpass operators, as used in this thesis, were designed to operate simultaneously on two 'input' images and produce a third 'output' or processed image. In this manner the contents (pixel values) of the input image files were not changed after a Boolpass operation therefore permitting additional, separate proceesing of the original image(s), if required.

The Boolean operations used in this thesis consisted of the "AND", "OR" and "NOT" operations. Though frequently considered in the context of binary operations, these Boolean operators were not limited to just binary applications here. Instead, Boolean operations were applied to any or all 16 pixel gray levels. The range of application of these Boolean operations was dependent on (1) the actual picture greylevel content, and (2) the gray level values specified by the threshold window.

A basic illustration of the functioning of these operations, using only binary values 'Ø' (TRUE) and '1' (FALSE), is given by the following truth tables. As was already noted, in this thesis the following operations were extended to as many as 16 values - Ø thru 16. ("A truth table gives the truth values of a logical expression for each combination of the truth values of its variables." [Ref. 7]

A	B	A AND B	A OR B
Ø	Ø	Ø	Ø
Ø	1	Ø	1
1	Ø	Ø	1
1	1	1	1

A	NOT A
Ø	1
1	Ø

These Boolean operators were used to perform two important processing operations: (1) pixel by pixel comparison of two digitized images, and (2) creation of inverse greylevel images, ie. negative pictures from positive pictures and positive pictures from negative pictures.

The first operation, pixel comparison, could be performed with or without thresholding. Without thresholding, the pixel comparison was accomplished by only the 'AND' operator. Using the AND operation, corresponding pixel positions in the two input images were sequentially compared or 'ANDED'. If the two values being compared were the same, that value was stored in a third or output picture file. If the values being compared were not the same, a value of 15 (blank) was stored in the output picture file.

(Note that the pixel values that were stored in the output file did not (necessarily) result in a binary picture - with values Ø (black) and 15 (white) only.) This ANDING operation was performed on all 64k picture pixels and the results stored as a new picture file, leaving the original input images unchanged.

When pixel comparison was performed with thresholding, both the 'AND' and 'OR' operators were required. The OR operation was needed because more than one value, as specified by the threshold window, was acceptable during the AND operation. In performing the pixel comparison with thresholding, a single pixel value in one input image file was compared to a range of pixel values (as specified by the threshold window) for the corresponding pixel in the second input image file. The OR operator was then used for comparison of this range of values in conjunction with the AND operation. Further discussions using this combination of AND and OR operations will be simply referred to as "AND" or "ANDING". The existence of the OR operation will be implicit to any "ANDING" operation where Threshold imposes a range rather than a single value for the comparison. The following example should help illustrate these pixel comparison concepts.

Consider two input file images - Ain, Bin, and an output file image, Pout. Corresponding pixel locations in each are represented by A1, B1 and P1. For this example let

A1 = 6 and B1 = 7. Threshold limits were calculated at .5 (interactive user input) X standard deviation to be equal to the pixel value plus or minus 2. The ANDING operation is now performed between A1 and B1:

First, without thresholding : given A1 = 6 and B1 = 7, does A1 = 6 AND B1 = 6? Obviously not, therefore a value of 15 is stored in P1.

Next, with thresholding : given A1 = 6, B1 = 7 and a threshold limit of + or - 2, does

A1 = 6 AND B1 = 4
OR
A1 = 6 AND B1 = 5
OR
A1 = 6 AND B1 = 6
OR
A1 = 6 AND B1 = 7
OR
A1 = 6 AND B1 = 8

Since A1 = 6 AND B1 = 6, for the threshold window given at B1, the answer is yes, and so a pixel value must be selected and stored in P1. The value of the pixel chosen to be stored in Pout, for example - either A1 or one of the five threshold window 'B' values, can have a significant impact on subsequent processing of Pout. Details of this type of processing will be discussed later in this section.

The initial research of this thesis, into the problem of processing real digitized images, concentrated on a search for a universally applicable method which could be used to help reduce, or at least more evenly distribute, the large amount of noise that was always present in the picture after digitization. Low pass filtering, as previously described, was found to be very useful in helping to solve this noise distribution problem.

The filter or mask size used in the low pass filter (LPF) operation, during picture processing, resulted in very different output picture images. Examples of these differences using 3x3, 7x7, and 11x11 mask sizes are illustrated below in figures 4 thru 6).

Fig. 4. F14 Tomcat (Fig. 3) Processed By a 3x3 Low Pass Filter

Fig. 5. F14 Tomcat (Fig. 3) Processed By a 7x7 Low Pass Filter

Fig. 6. F14 Tomcat (Fig. 3) Processed By an 11x11 Low Pass Filter

During the research, examination of these and many other examples showed a definite tendency of object or feature edge structures to remain while more inconsistently textured, noisy areas were smoothed. The discovery that certain basic structures remained relatively unchanged after processing, while others did not, provoked interest over what features did remain unchanged after processing with different size masks. It was thought that a direct comparison of two low pass filtered images might provide an answer to the question.

The images used for comparison were identical except that each had been previously processed by a different size, but same dimension, low pass filter. Comparison of the two images was accomplished through use of the Boolean AND (and OR) operation. The ANDING operation, depending on the threshold setting, was able to substantially reduce the amount of energy in the picture. However, feature extraction at this level was still considered difficult and further testing using combinations of different size filtered images, thresholding, and ANDING gave interesting but mixed results. For examples see Section IV - Results.

Large areas of unwanted picture information still remained in those areas where the picture gradient was very small. Attempts to reduce these areas using picture thresholding and conventional edge operators (such as Kirsch and 5-level simple masks [Ref. 9]), met with no success.

Further visual study of the digitized pictures showed that at least some of this unwanted picture energy might be removed if the undesired area, of a particular greylevel, was combined with its complement. The solution to this problem was the creation and use of negative images during the combinational image comparisons.

For any given image, the complement or negative of that image ANDED to itself would, of course, yield an output image consisting only of blanks. The problem which then had to be solved was what negative image could be used to help cancel or remove the 'undesired' areas in the image being processed. The solution to this problem was to use a low pass filtered image of a different mask size than that of the image being processed. These filtered images contained similar, but not exactly the same pixel information to that of the image being processed. The negative of this filtered image was then used as one of the input images to the Boolean AND operation. The original image that was being processed was used as the second input image to the AND operation. Therefore, when a positive image and a low pass filtered image were simultaneously processed by the AND operation, a large reduction in pixel energy became possible. The amount of energy reduction achieved was dependent on the size and type of filter used, as well as the threshold window chosen for the Boolean operation.

Application and experimentation with this technique of

ANDing selective positive and negative images proved to yield some very interesting results. Test results showed a significant improvement in total picture energy reduction while relevent feature information was retained. Detailed results are illustrated in Section IV.

It has already been shown, though a previous example, that during the processing of two images by the AND operator, pixel values are stored in a separate picture file to form the new processed picture. During the combinational testing of the Boolpass operators, the values which were chosen to be stored and represent the new processed picture were significant to any subsequent processing done to that picture. The reason for this can be attributed to the threshold window specified for the subsequent processing operations. The following example should best illustrate why.

For this example, consider three input image files - Ain, Bin, Cin, and three corresponding pixel locations in each - A1, B1, C1. If A1 was being ANDED with B1 and the values were exactly equal, it would not matter from which image the pixel value was taken to be stored into the processed picture matrix, Pout. However, if the pixel values were not exactly equal, acceptance or rejection is based upon the limits specified by Threshold. Again, let A1 = 6, B1 = 7, and C1 = 9. Threshold limits were calculated at .5 X standard deviation to be equal to the pixel value plus or minus 2.

The ANDING operation is now performed between A1 and B1, ie given A1 = 6 (and a threshold limit of + or - 2), does B1 = 4 OR 5 OR 6 OR 7 OR 8 ? Since B1 = 7, the answer is yes, and so a pixel value must be selected and stored in 'Pout'. If the value of A1 is chosen, P1 = 6; if B1 is chosen, P1 = 7. Many other methods of pixel value selection could be used based on any number of different criterea. In this thesis, if the result of the AND operation required storage of some pixel value from the Bin file, the Boolpass operators always stored the pixel value of the original input 'B' file. The choice of the 'B' input pixels was based on hierarchical test results obtained during interactive tests using several ad hoc criterea.

Next, the same processing operation is to be applied to a multiprocessing of the pictures 'Cin' and 'Pout'. From the value given for C1 (ie. 9), performing the AND operation exceeds the Threshold limits if A1 had previously been the value for retention and storage in Pout. Therefore, the value stored in another new picture matrix, say P', would be a blank. However, if B1 had been stored in Pout, the value would have been within the threshold limits and a greylevel other than a blank would have been stored.

From this example, it is clear that selection criterea for pixel values stored after processing does affect subsequent processing of that image. It is also for this reason that multi-level processing of pure binary images using the Boolpass operators is not very effective. This is due to the large amount of feature information which is lost for subsequent processing as a result of reducing the picture to a binary image.

IV. RESULTS

General

The number of combinations of processing techniques which are possible using the concepts and techniques of the Boolpass operators is very large. The testing results which follow should not be considered an exhaustive study of these combinations; however, in exercising the Boolpass operators on a total of 8 original images, over 400 pictures were processed. The development of the Boolpass operators was not the result of some totally preconceived theoretical or abstract concept but rather, of an hierarchical approach based on 'man-in-the-loop' real time interactive image processing using a digital computer. The 'loop' referred to here consisted of interactive image processing programs on a digital computer; a video monitor which displayed the processed (output) images; and a 'man' or 'user' who controlled the processing environment and input parameters to the computer programs. The following diagram illustrates the interactive 'loop' concept.

```
                              ____________
                              I          I
user input                  / I computer I\
parameters                 /  I__________I \  processed
& controls                /                 \   (output)
                         /                   \    image
________                /                     \
I      I               /                       \  ____________
I USER I /                                      \ I          I
I      I/________________________________________\I  Video   I
I______I        visual 'feedback'                 I Monitor  I
                to user on image                  I__________I
                   processed

                      : :
                      : :
                    RESULTS
                      IN
                      : :
                      : :
                     \   /
                      \ /
                       :
              Distinct sequence &
                  combination
              of image processing
               operations (using
              low pass filtering,
              & Boolean operators)

              "BOOLPASS OPERATORS"
```

This interactive approach to image processing allowed the user to visually analyze the output images after each stage of the processing. The visual analysis then resulted in a subjective, but discerning human 'feedback' loop within the image processing system. In this way, changes to the input parameters were made, if considered necessary, and used for subsequent processing of the images.

A particular sequence of processing steps, or Boolpass operation, was considered noteworthy if, when applied to several very different images, that same operation(s) performed well at extracting feature information (considered "useful" by the user) in all the test images.

The results illustrated herein represent those combinations of the Boolpass operators which appear to have the greatest potential for use in an image processing/pattern recognition machine. It should be noted that the processed pictures in this section are the result of applying only the Boolpass operators (as previously described in Section III), unless specified otherwise. Further noise reduction and target extraction from the processed images which follow would be facilitated by the use of additional target or feature information. Additional processing data such as range to target information, target size and feature templates would be valuable aides to further refinement of the images or target extraction.

It should be emphasized, again, that the results of the Boolpass operators, as given in this thesis, represents only the first stage of a two stage pattern recognition machine. This, the first stage, accomplishes overall picture energy reduction and feature edge extraction. The second stage, target recognition, is then performed on the output image of the first stage of processing. The advantage to this technique is that the complex task of target recognition is only applied to a small number of probable target areas (as determined by the Boolpass oerators) , as opposed to the entire image. This aids in reducing the complexity of the target recognition task, which in turn enhances the operating speed of the overall pattern recognition machine.

Sequence of Presentation - Results

The results have been divided into three main topic areas: 1) low pass filtering (both one and two dimensional); 2) boolean operators (AND, OR, NOT); and 3) Boolpass operators (with thresholding). Each picture, or sequence of pictures presented, has been preceeded by an outline of those steps and input values which were used to process the input image(s) to the final, output picture shown. Subsection 3), on the Boolpass operators, also includes three dimensional (3-D) picture energy plots of the original input and final output images to the Boolpass operators. These 3-D plots, which graphically show the pixel value distribution over the entire picture, help to illustrate the effectiveness of the Boolpass operators in achieving picture energy reduction.

For ease of reference, page vi begins a detailed listing of all pictures (figures) displayed in this section.

1) Low Pass Filtering

Figures 7 thru 18 show examples of two dimensional low pass filters, using mask sizes of 3x3, 5x5, 7x7, 9x9 and 11x11. One dimensional filtering (in the horizontal axis) is illustrated in figures 19 thru 26 for mask sizes 3, 5, 7 and 11.

It should be noted that the amount of computation required for a two dimensional filter is at least N times that required for a one dimensional filter of size N. (For the computer used in this thesis, a Data General Eclipse S/250, this translated into the two dimensional filter requiring almost 3 times the processing time required for the one dimensional filter.) As might be expected, further, but similar processing of the output of these two filters also yields two distinct images. The one dimensional filter, because of computation speed, might seem the obvious choice for a pattern recognition machine. However, the choice of which filter should be used in a particular algorithm or pattern recognition machine depends on the constraints imposed by that machine or algorithm. For example, a target extraction algorithm may work very effectively on the output of a two dimensional filter, but not as well on the one dimensional ouput. In another case, speed may be critical and the one dimensional filter must be used, even if more errors are introduced into the target extraction algorithm.

Later in the results, examples are given of Boolpass operators using both types of low pass filters. The results show that subsequent processing using the Boolpass operators yield similar ouput images. This observation could imply that the faster, one dimensional filtering might be sufficient for use with the Boolpass operators in a pattern recognition machine.

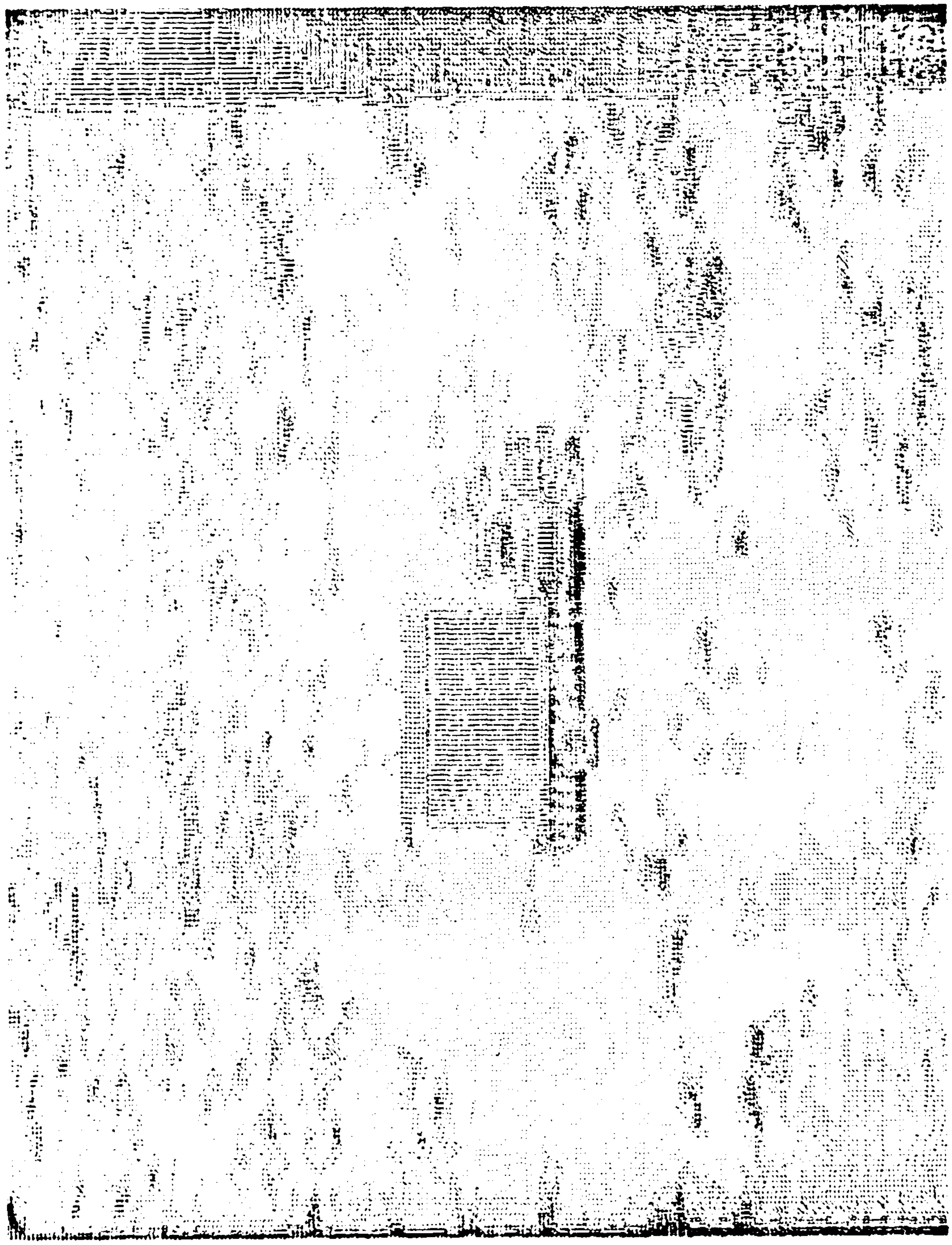

Fig. 7. Truck - No Filtering

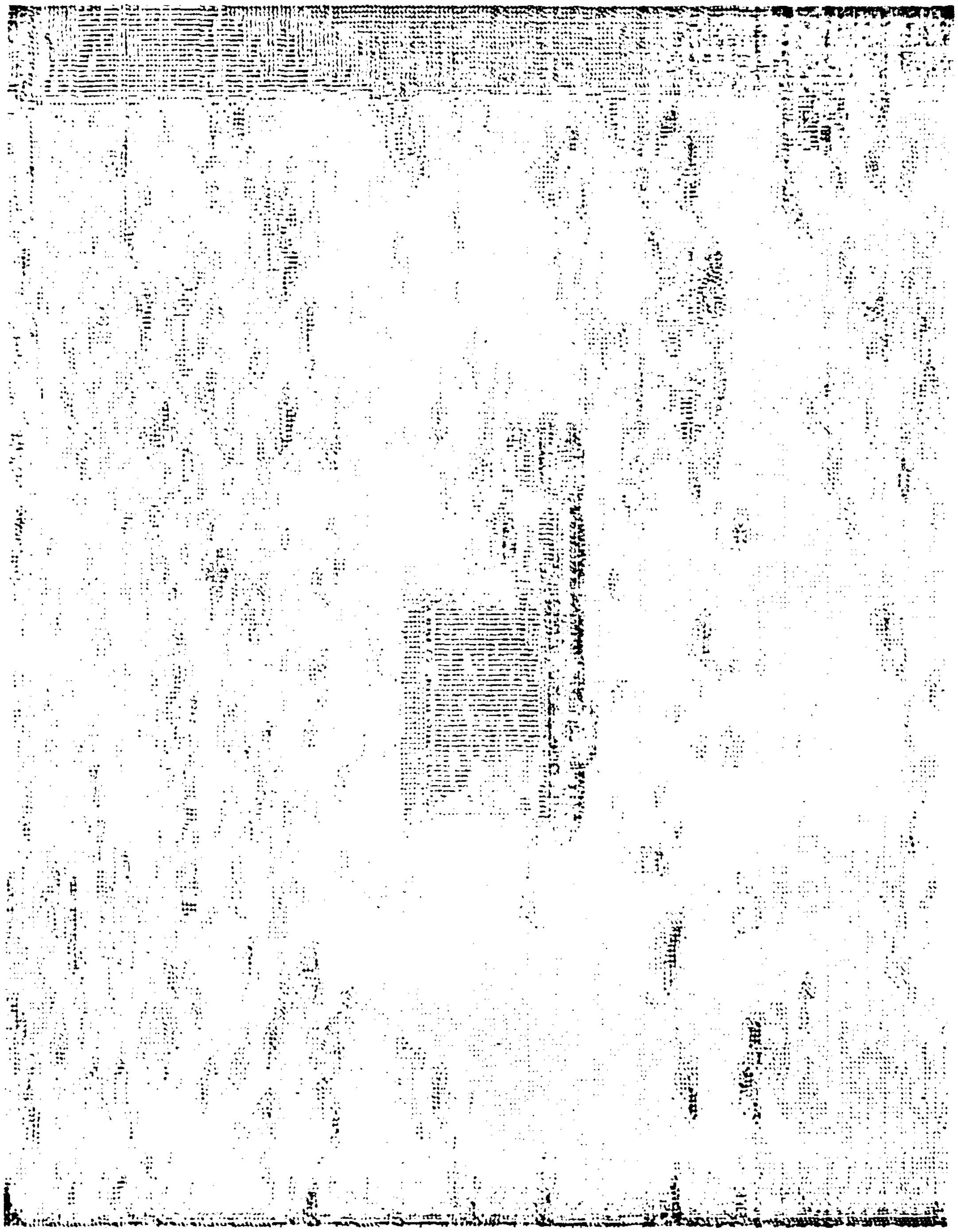

Fig. 8. Truck (Fig. 7) Processed With LPF 3x3

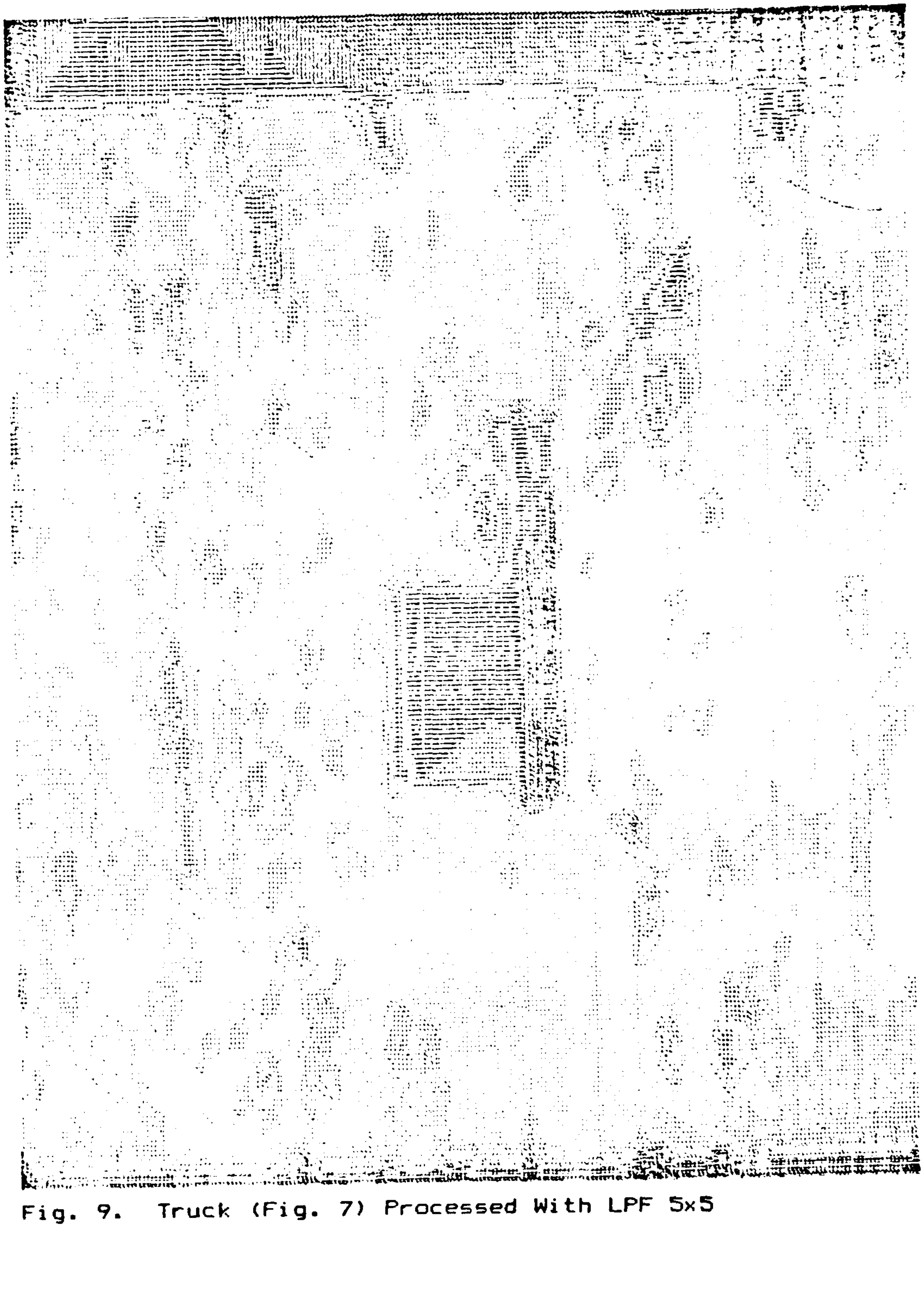

Fig. 9. Truck (Fig. 7) Processed With LPF 5x5

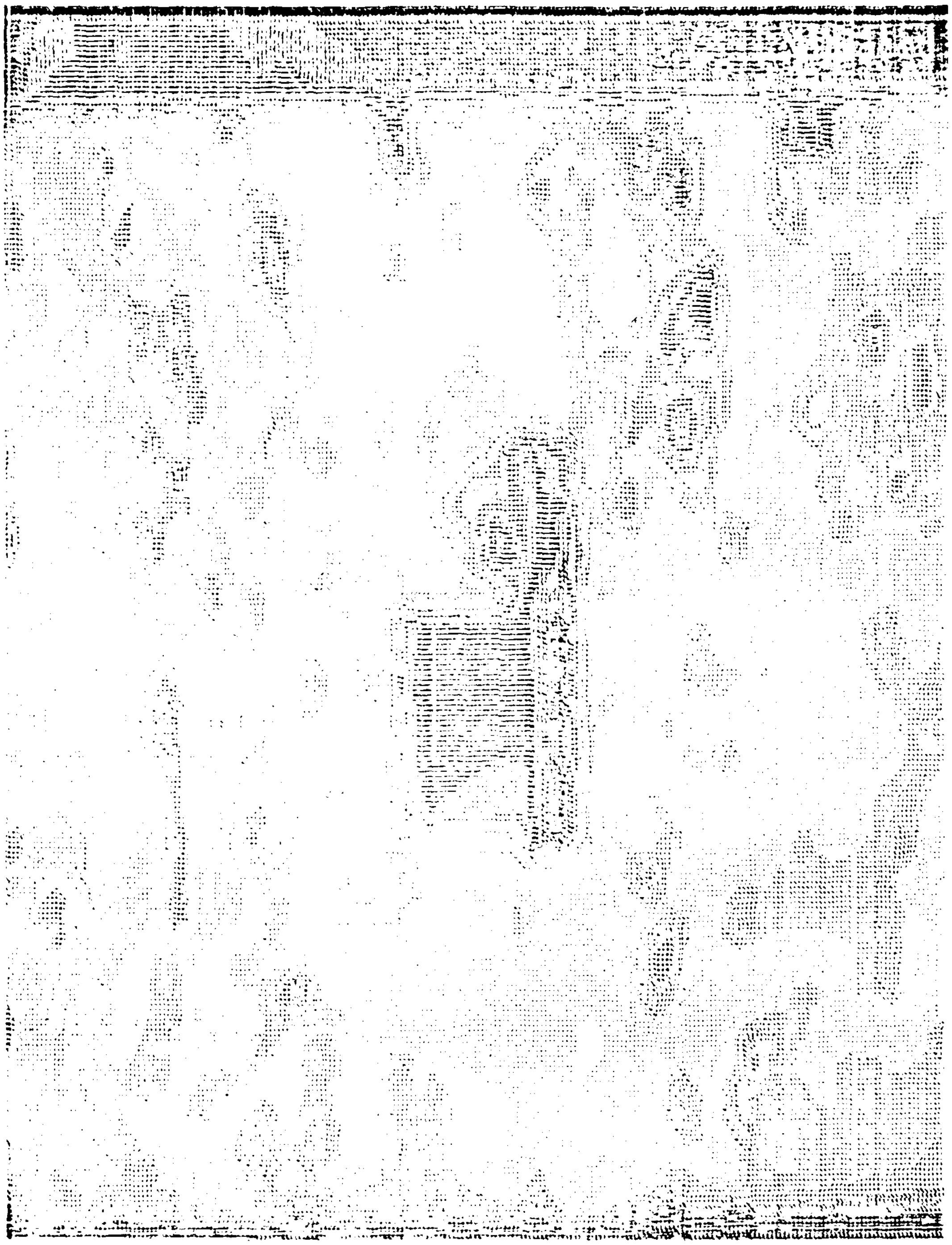

Fig. 10. Truck (Fig. 7) Processed With LPF 7x7

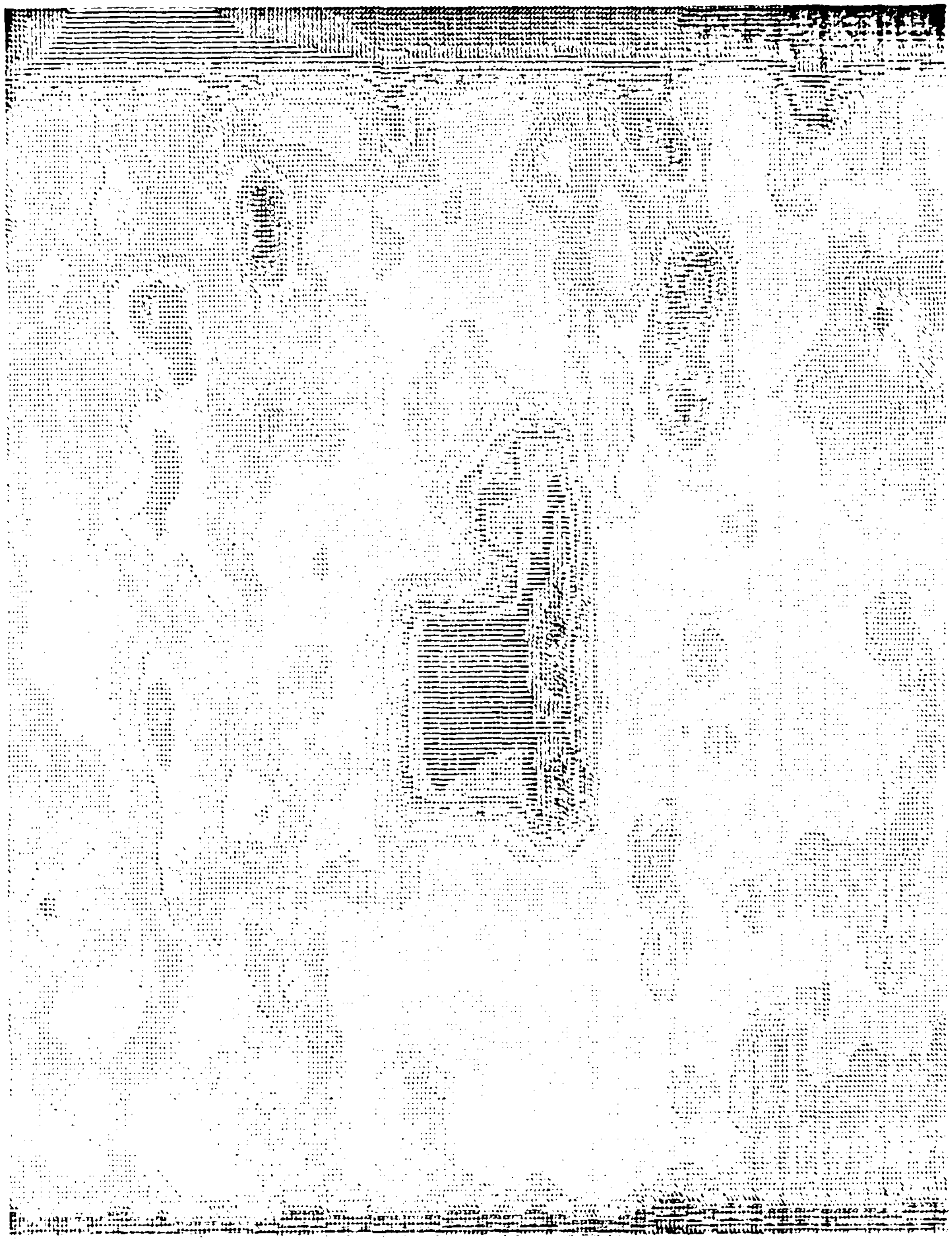

Fig. 11. Truck (Fig. 7) Processed With LPF 9x9

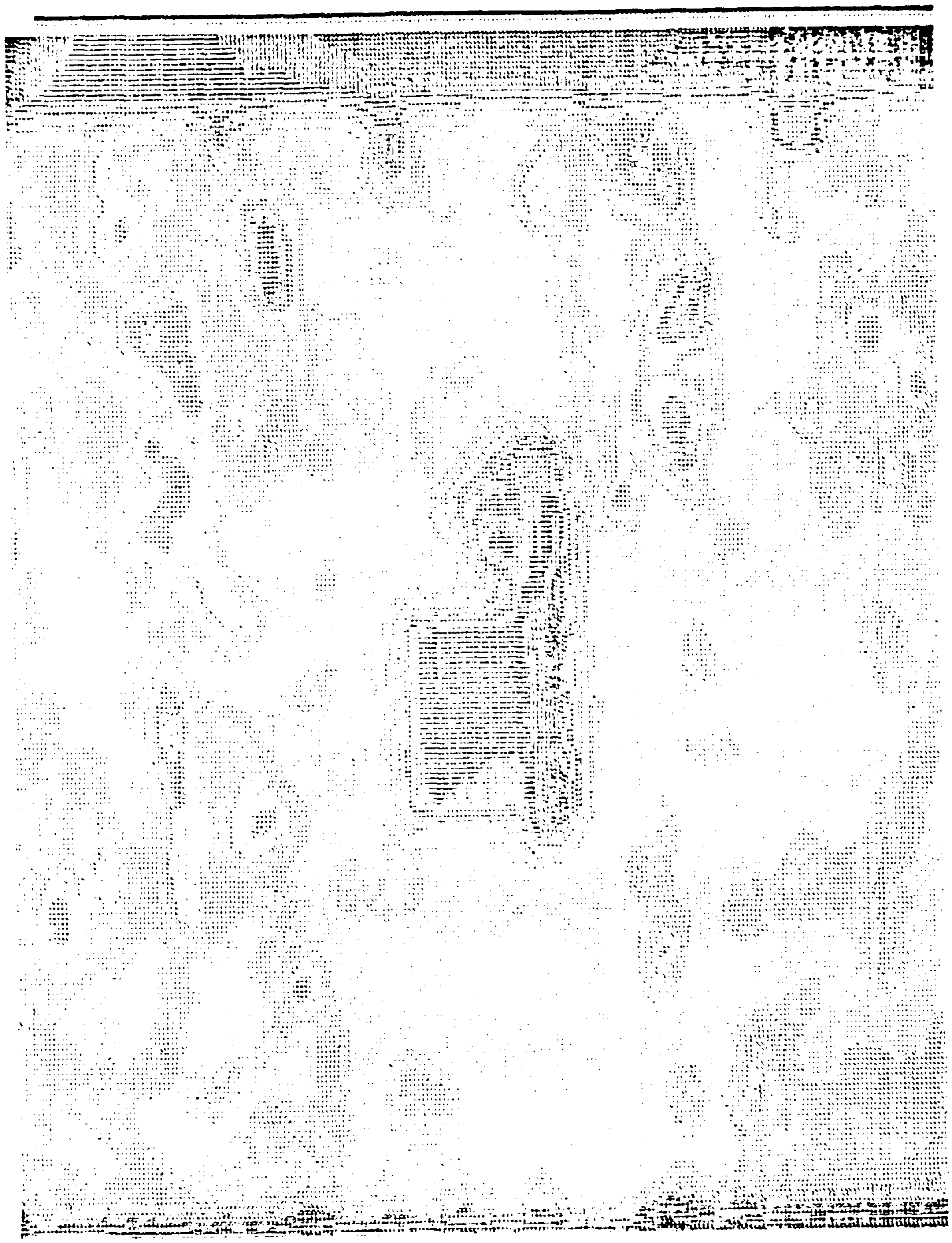

Fig. 12. Truck (Fig. 7) Processed Wth LPF 11x11

Fig. 13. Field - No Filtering

Fig. 14. Field (Fig. 13) Processed With LPF 3x3

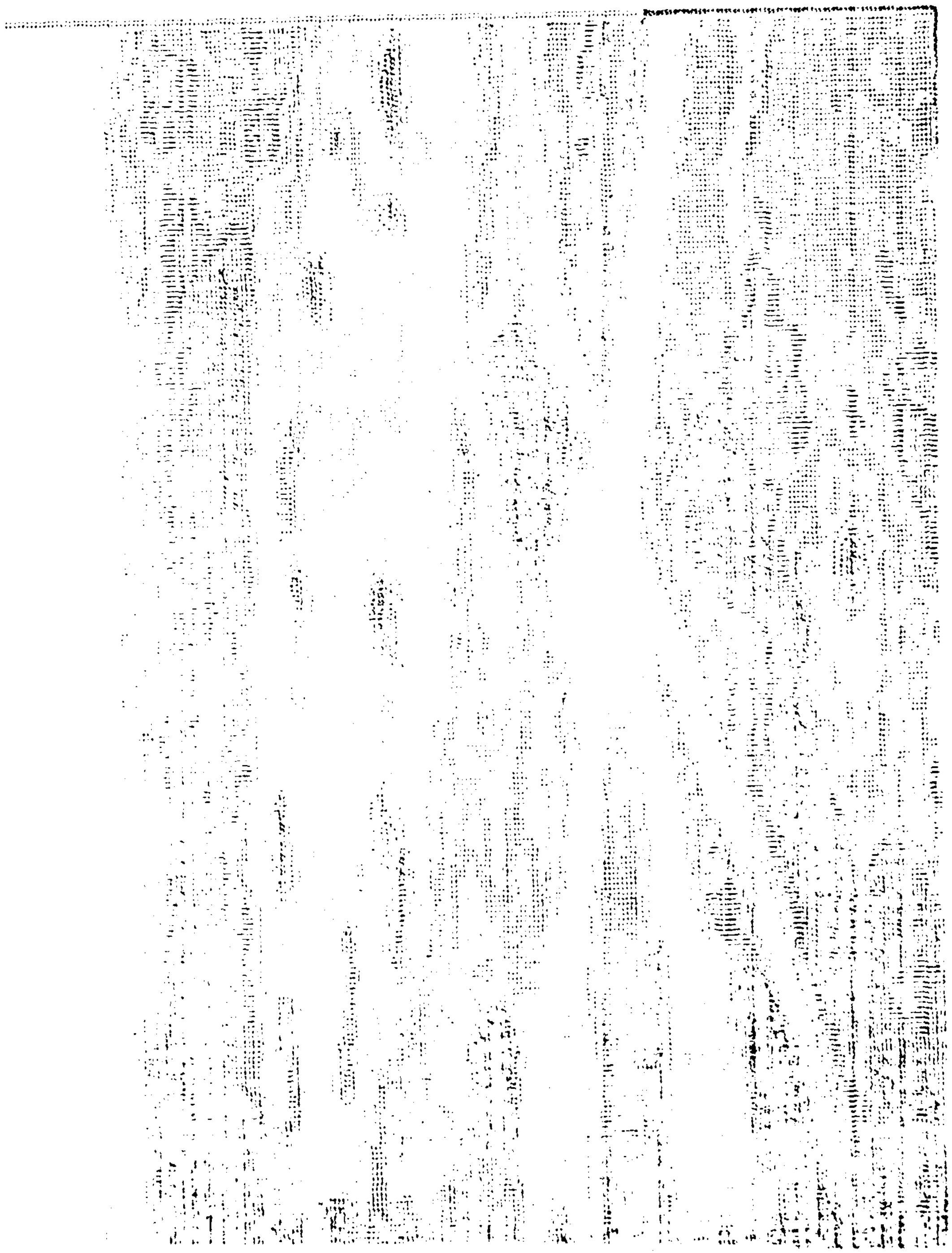

Fig. 15. Field (Fig. 13) Processed With LPF 5x5

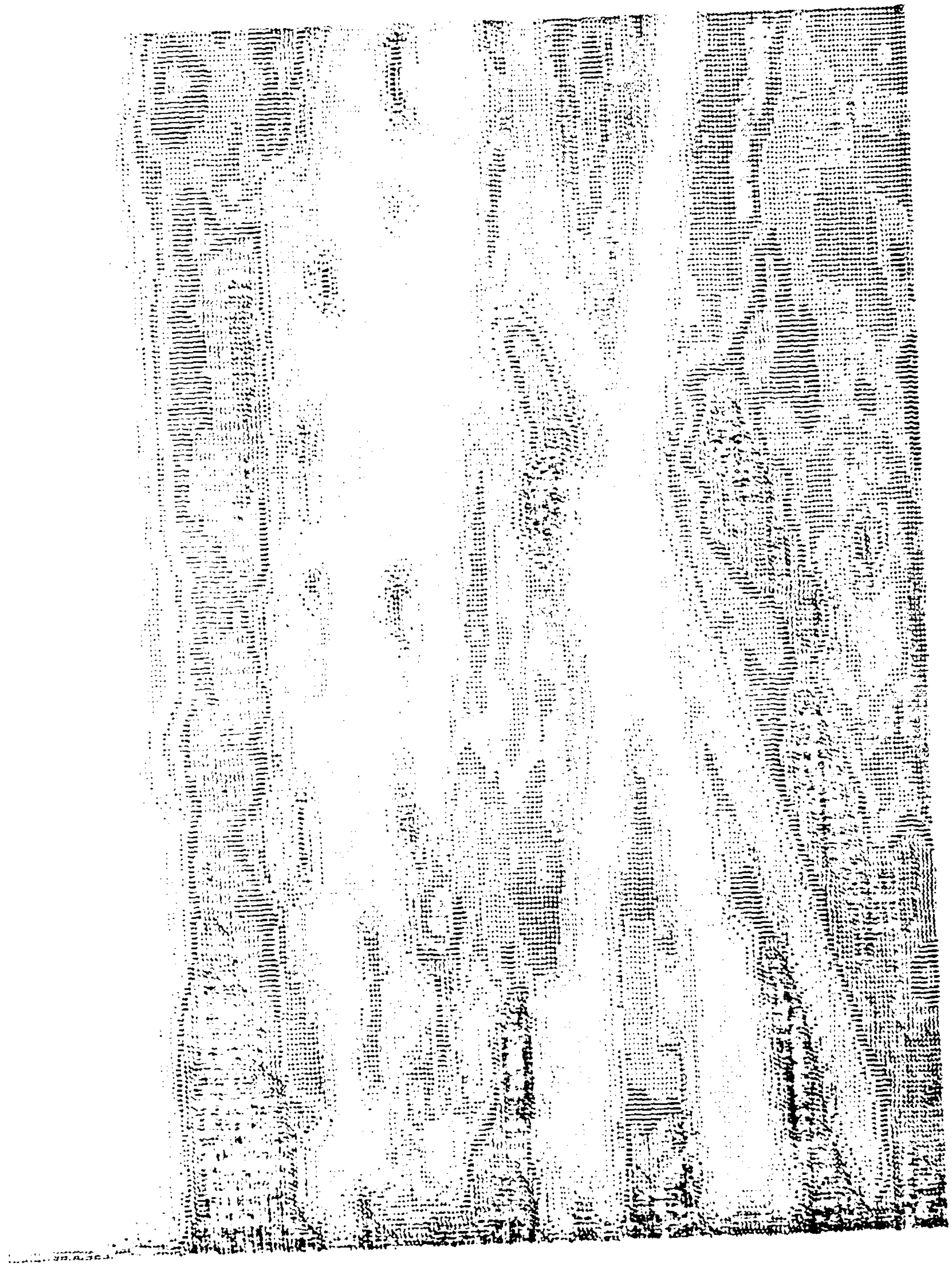

Fig. 16. Field (Fig. 13) Processed With LPF 7x7

Fig. 17. Field (Fig. 13) Processed With LPF 9x9

Fig. 18. Field (Fig. 13) Processed With LPF 11x11

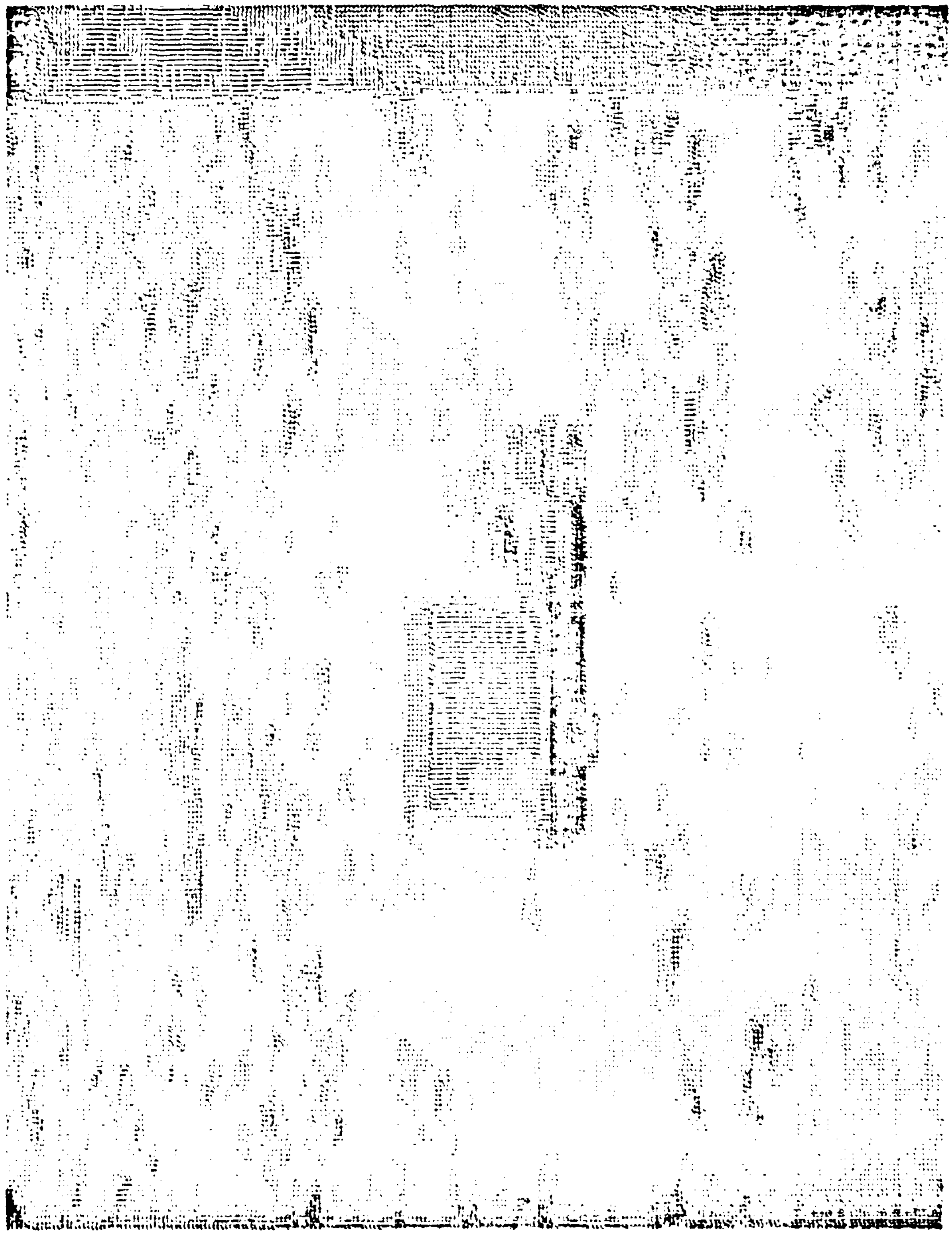

Fig. 19. Truck (Fig. 7) Processed With LPF 1x3

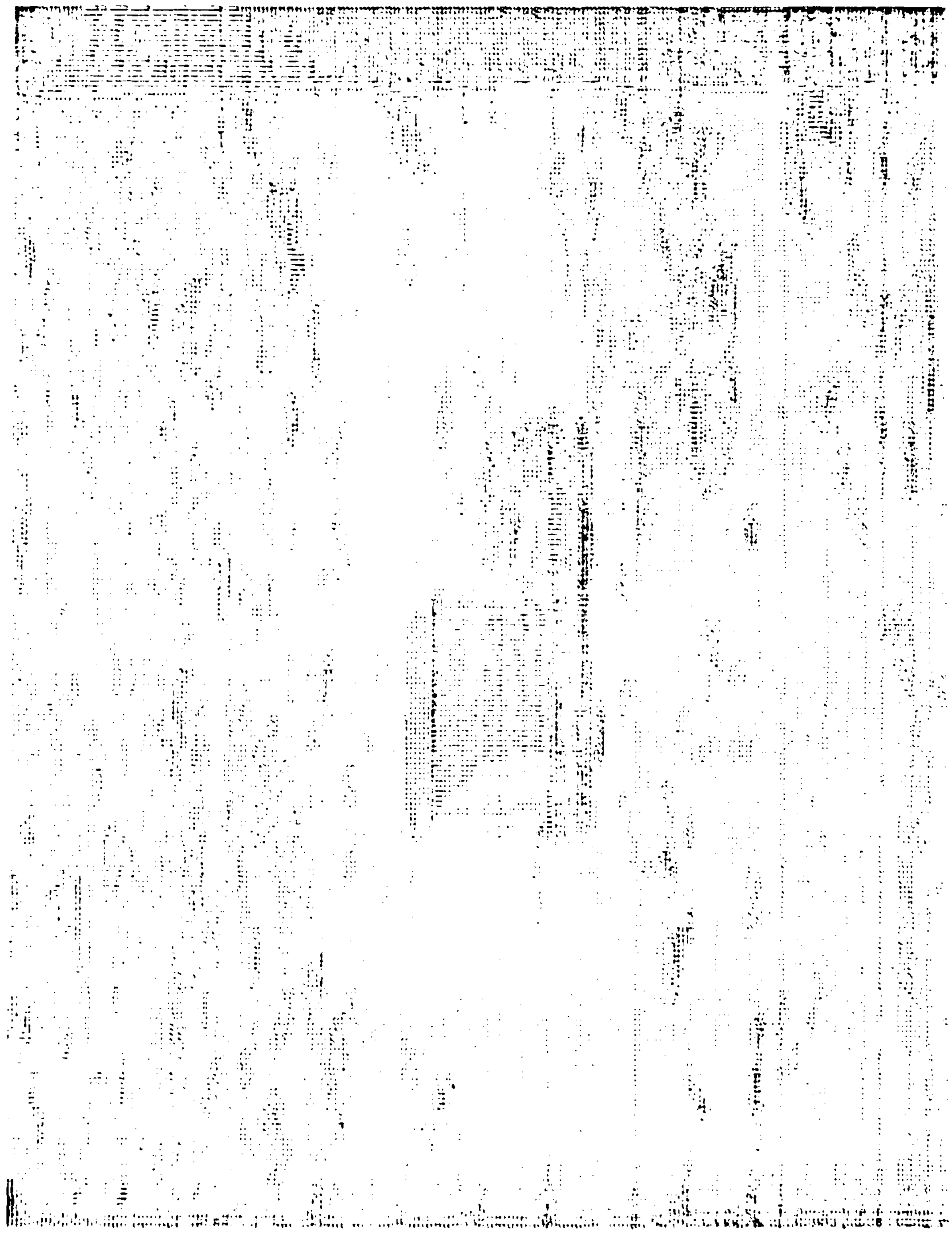

Fig. 20. Truck (Fig. 7) Processed With LPF 1x5

Fig. 21. Truck (Fig. 7) Processed With LPF 1x7

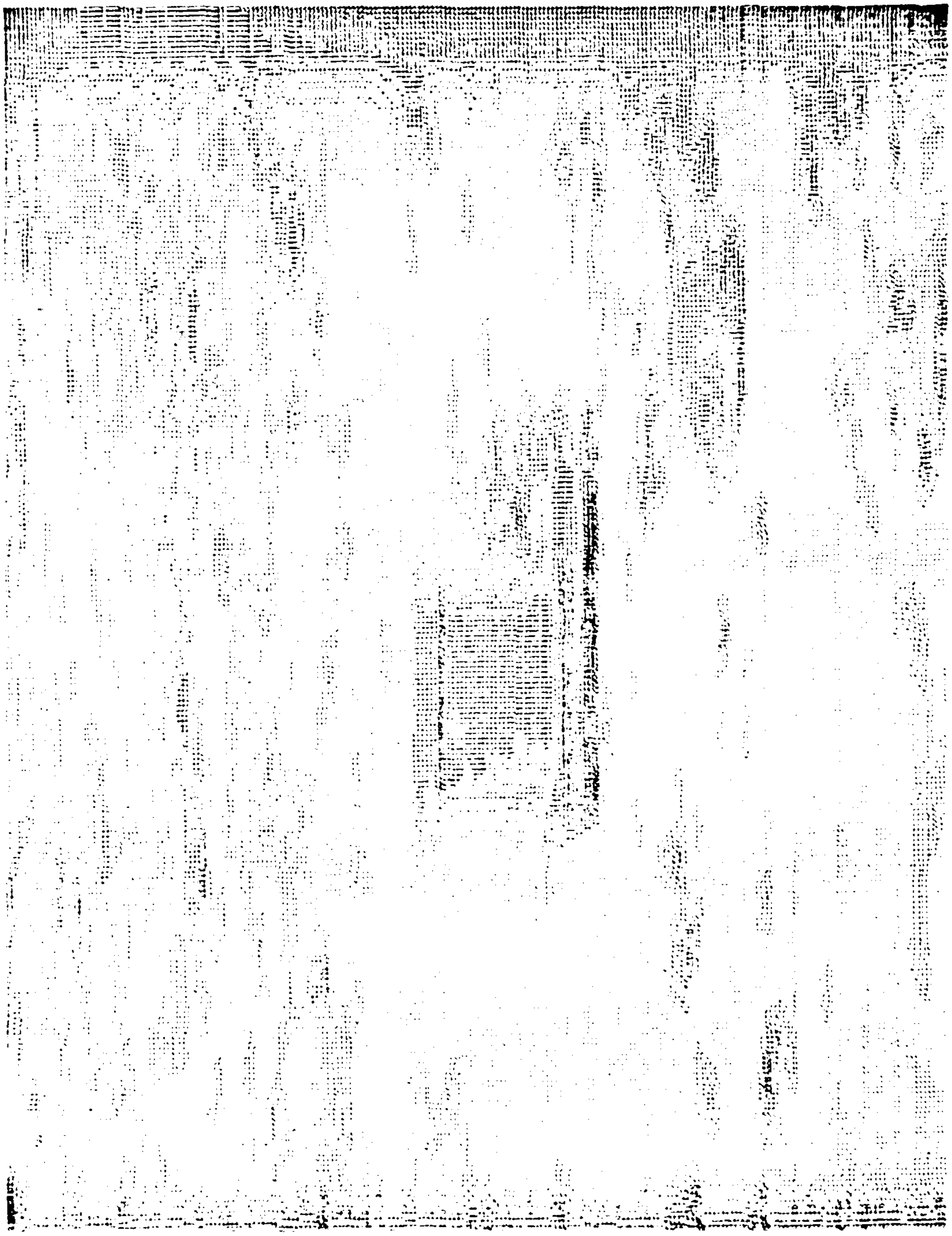

Fig. 22. Truck (Fig. 7) Processed With LPF 1x11

Fig. 23. Field (Fig. 13) Processed With LPF 1x3

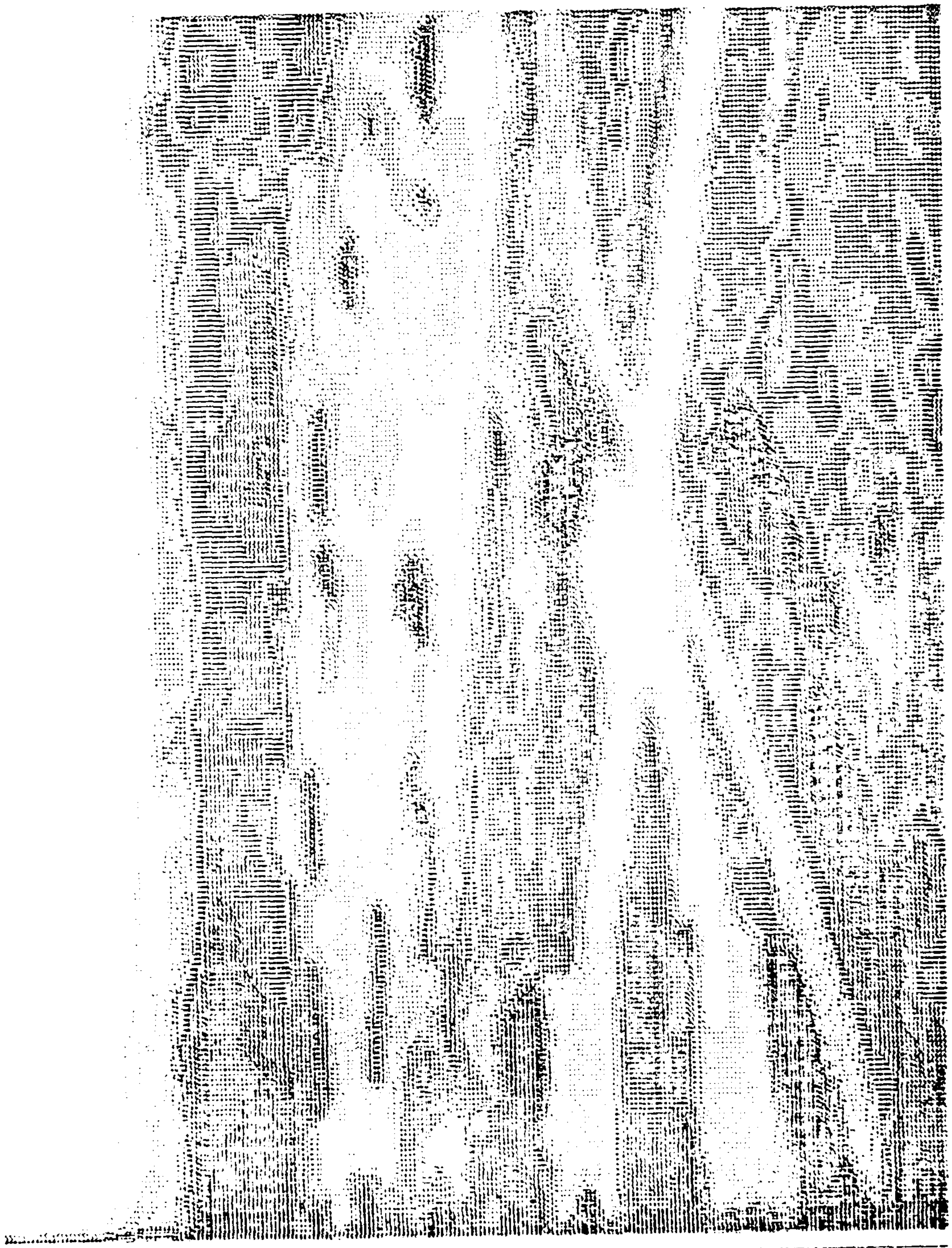

Fig. 24. Field (Fig. 13) Processed With LPF 1x5

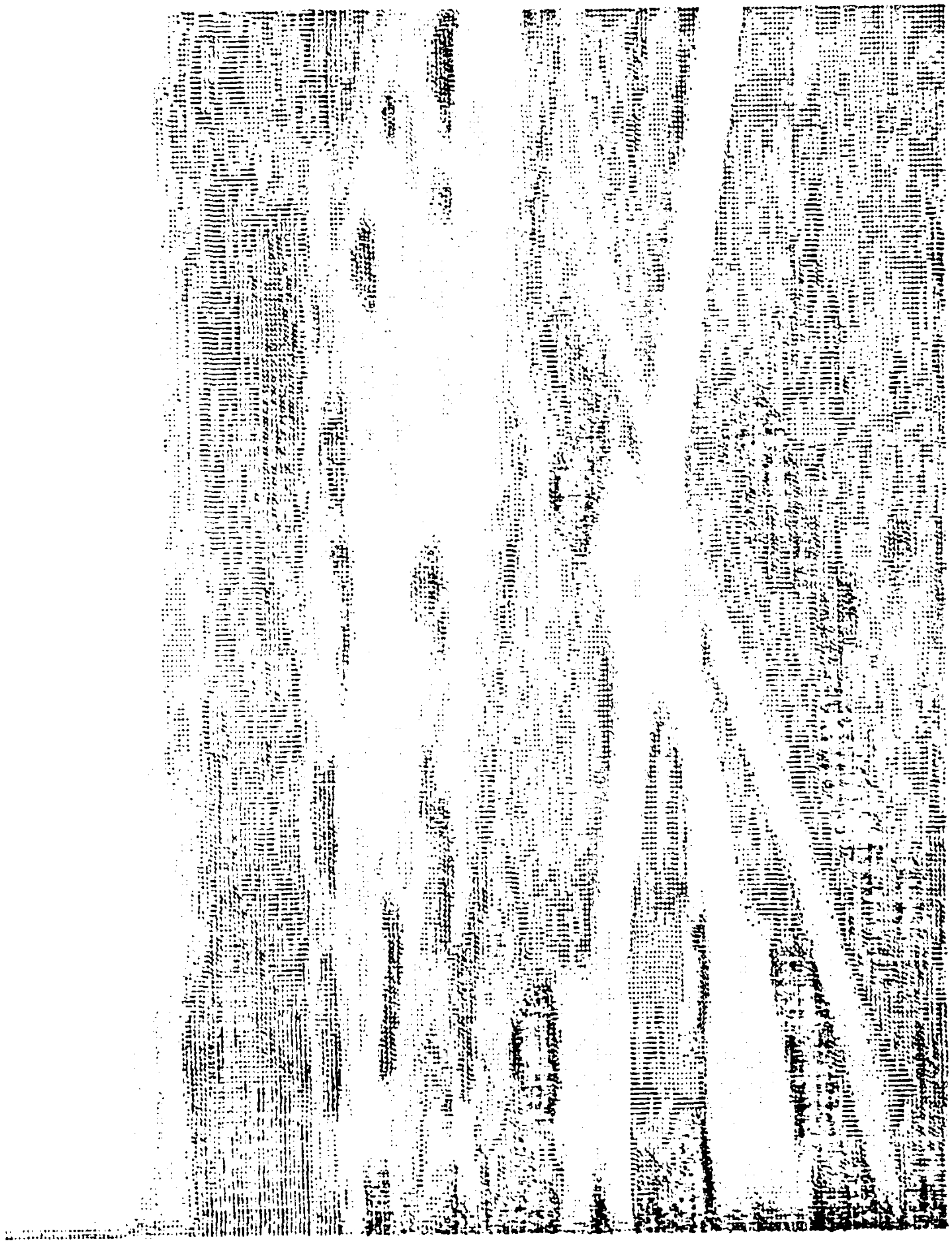

Fig. 25. Field (Fig. 13) Processed With LPF 1x7

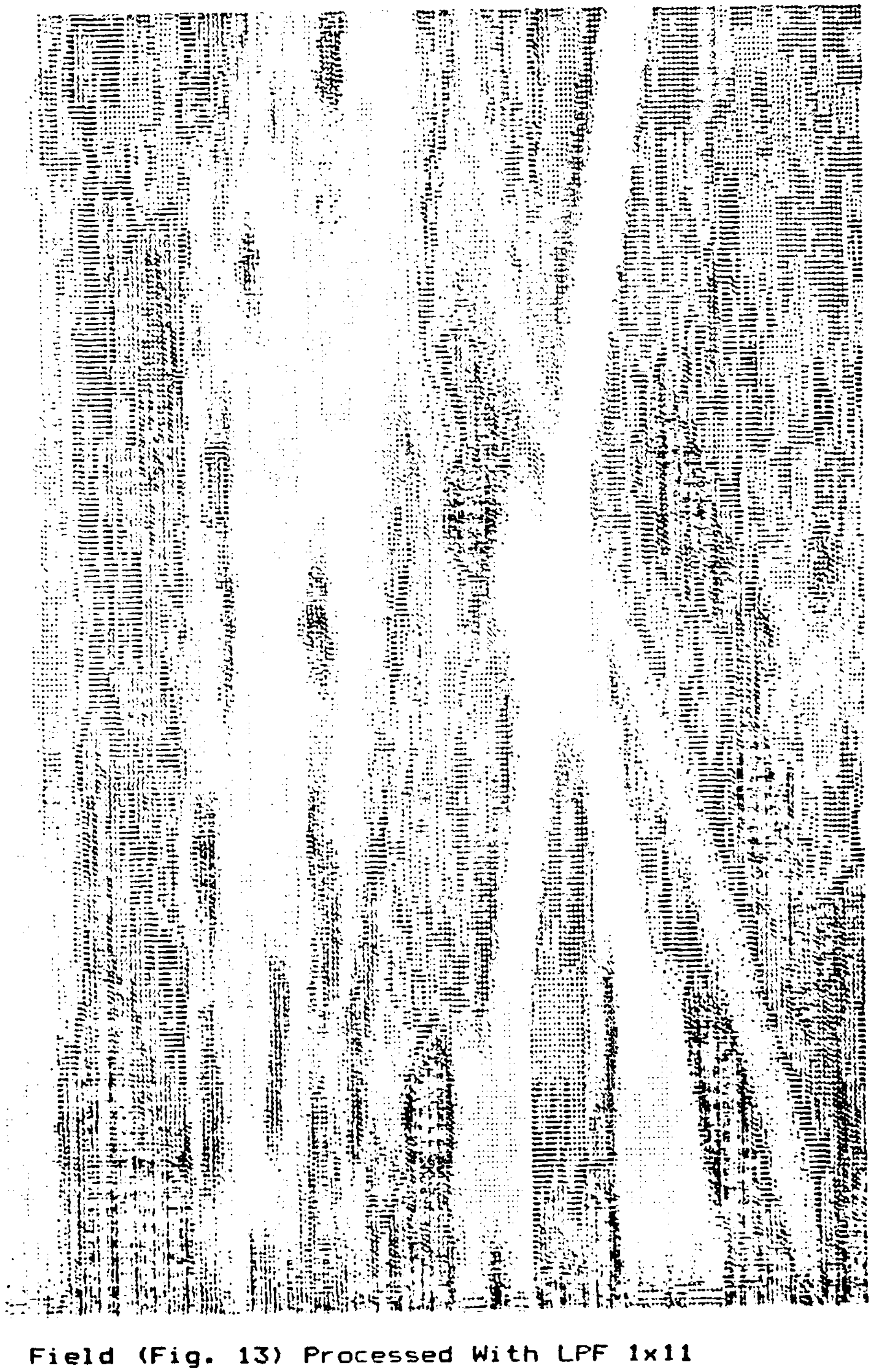

Fig. 26. Field (Fig. 13) Processed With LPF 1x11

2) Boolean Operators

A description of the boolean operations AND, OR and NOT, as applied to image processing, was previously detailed in Section III. Further explanation, beyond that already given, is not considered necessary here to be able to understand and interpret the processed image results which follow.

Examples of images processed using boolean operations , both with and without thresholding, are presented. This provides the reader with an opportunity to compare and contrast the effect of imposing this additional constraint during any particular boolean operation.

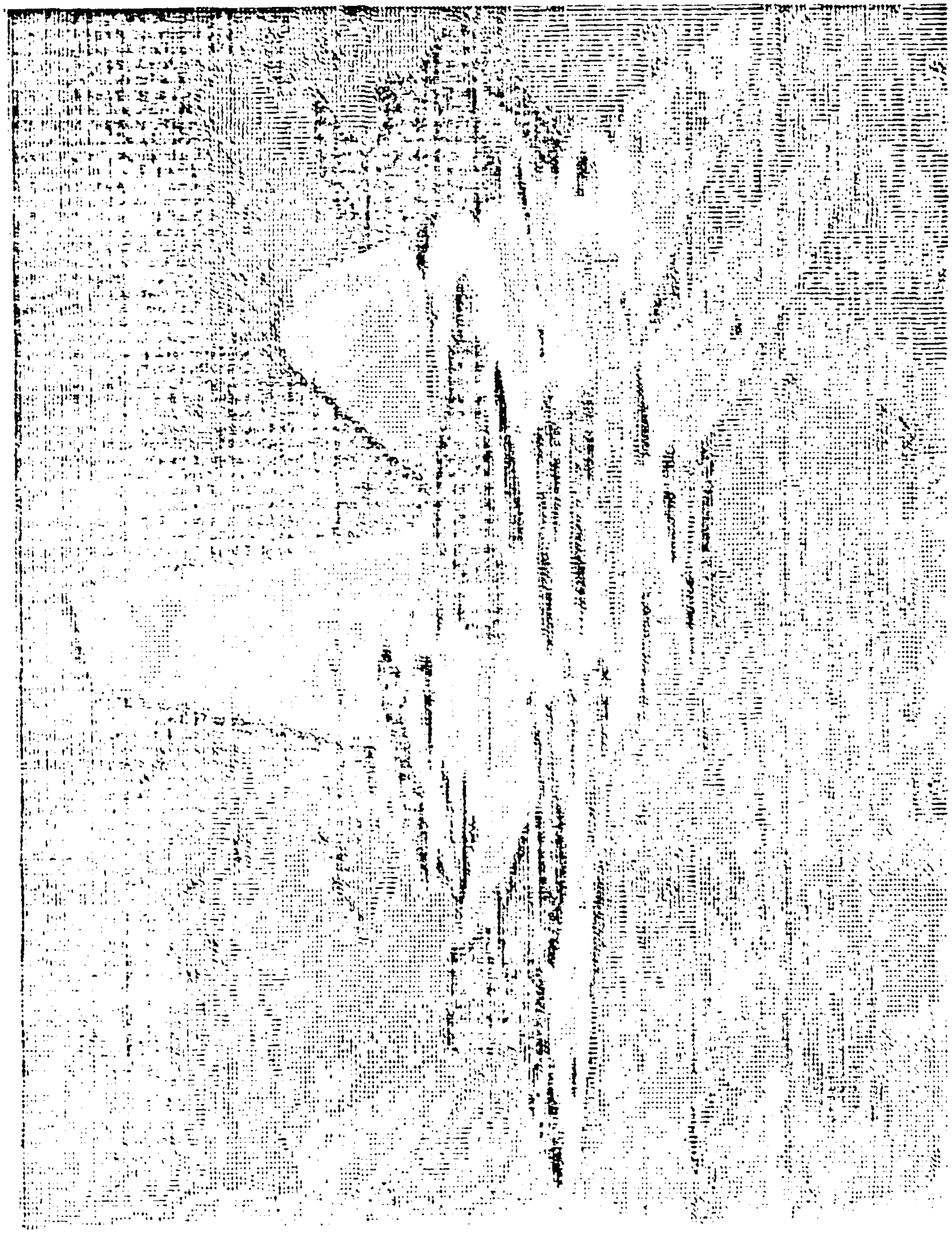

Fig. 27. Boolean NOT Operation Performed on F14 Tomcat (Fig. 5)

Fig. 28. Boolean NOT Operation Performed on Field (Fig. 13)

Fig. 29. 5x5 LPF Truck (Fig. 9) ANDED With Truck (Fig. 7) Threshold = Ø

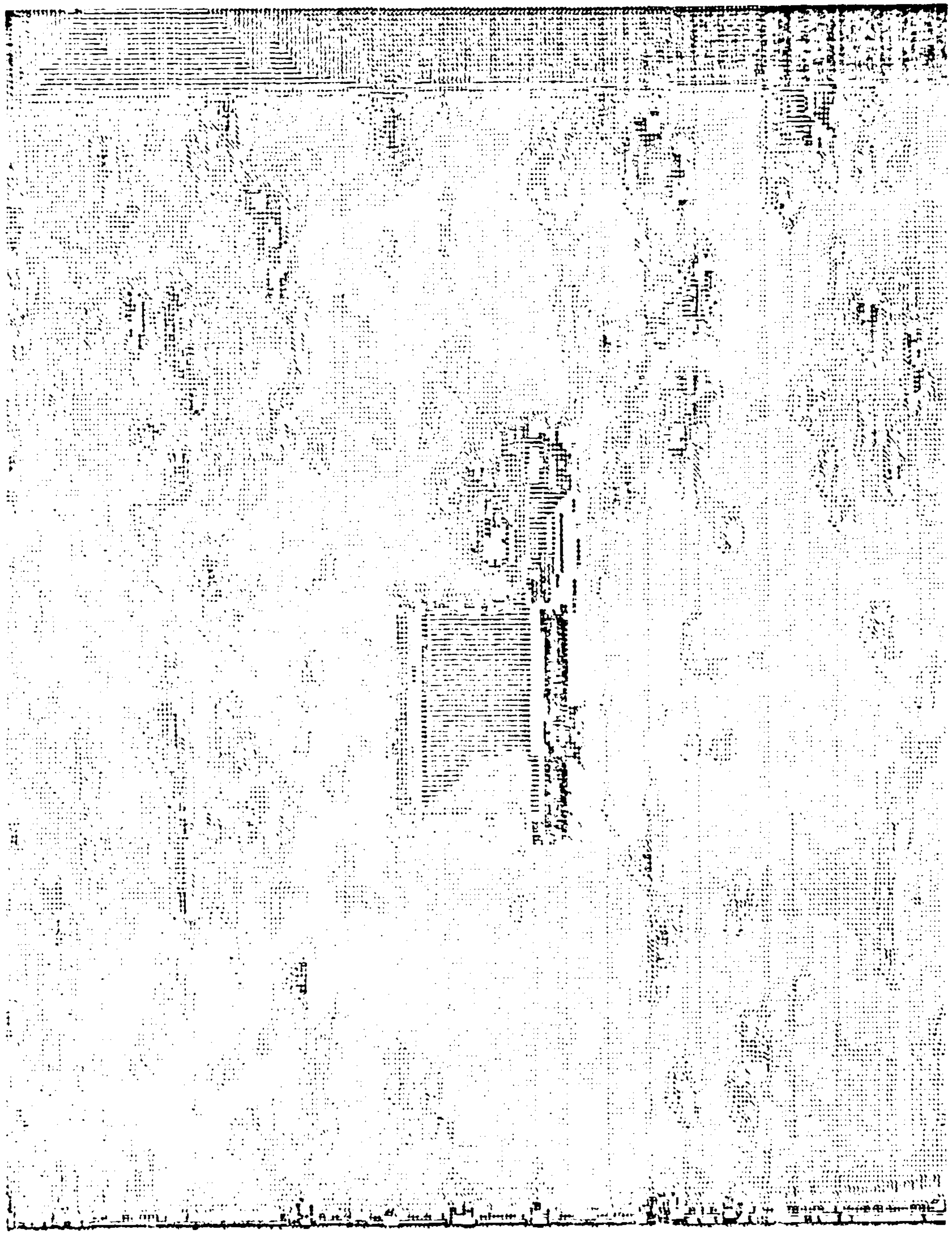

Fig. 30. 5x5 LPF Truck (Fig. 9) ANDED With Truck (Fig. 7)
Threshold = .2

Fig. 31. 5x5 LPF Truck (Fig. 9) ANDED With Truck Threshold = .4

Fig. 32. 5x5 LPF Truck (Fig. 9) ANDED With Truck (Fig. 7)
Threshold = .6

Fig. 33. 5x5 LPF Truck (Fig. 9) ANDED With Truck (Fig. 7) Threshold = .8

Fig. 34. 5x5 LPF Truck (Fig. 9) ANDED With Truck (Fig. 7)
Threshold = 1.0

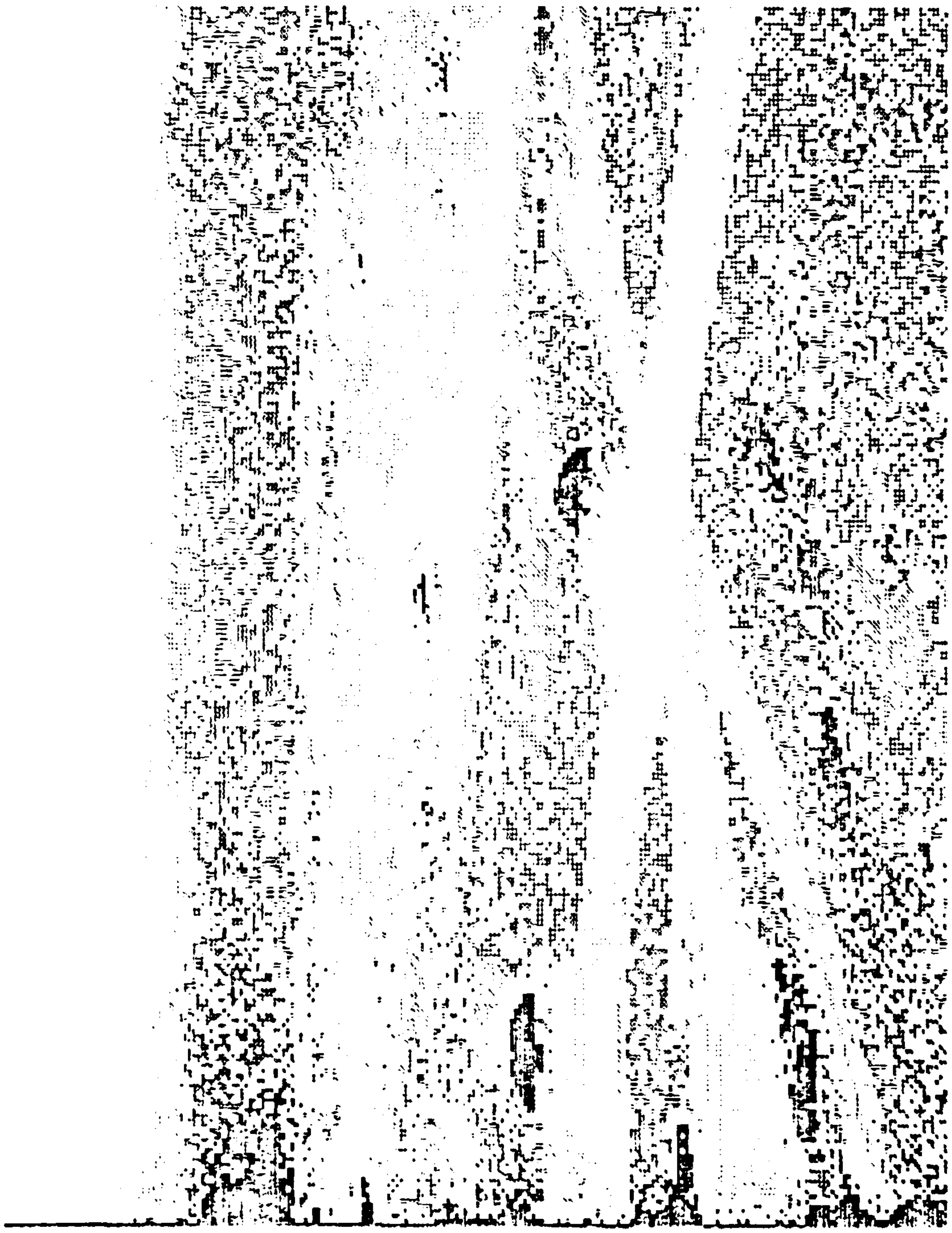

Fig. 35. 1x5 LPF Field (Fig. 24) ANDED With Field (Fig. 1)
Threshold = Ø

Fig. 36. 1x5 LPF Field (Fig. 24) ANDED With Field (Fig. 1) Threshold = .2

Fig. 37. 1x5 LPF Field (Fig. 24) ANDED With Field (Fig. 1)
Threshold = .4

Fig. 38. 1x5 LPF Field (Fig. 24) ANDED With Field (Fig. 1)
Threshold = .6

Fig. 39. 1x5 LPF Field (Fig. 24) ANDED With Field (Fig. 1)
Threshold = .8

Fig. 40. 1x5 LPF Field (Fig. 24) ANDED With Field (Fig. 1)
Threshold = 1.0

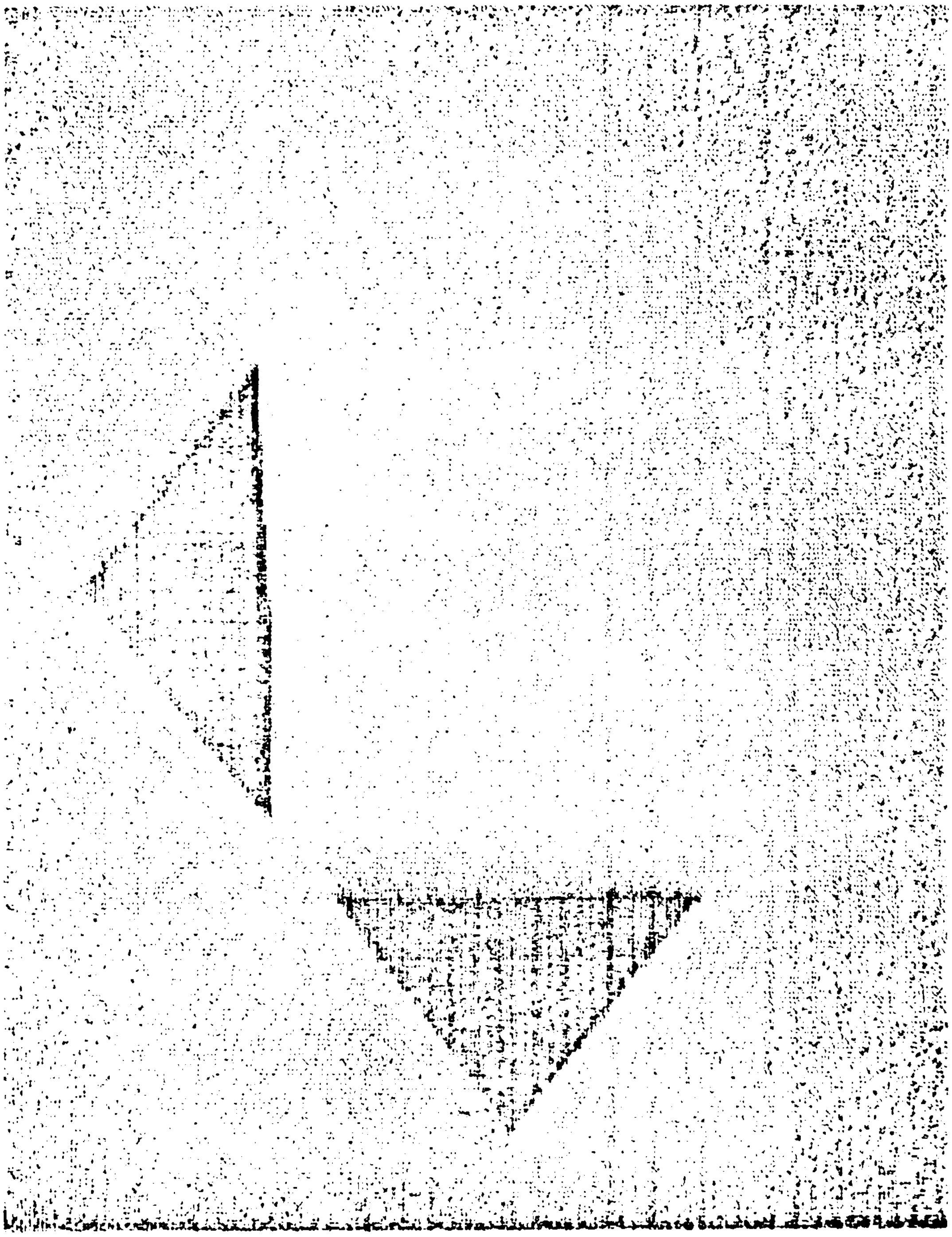

Fig. 41. Digitized Image of Black Triangle on White Background

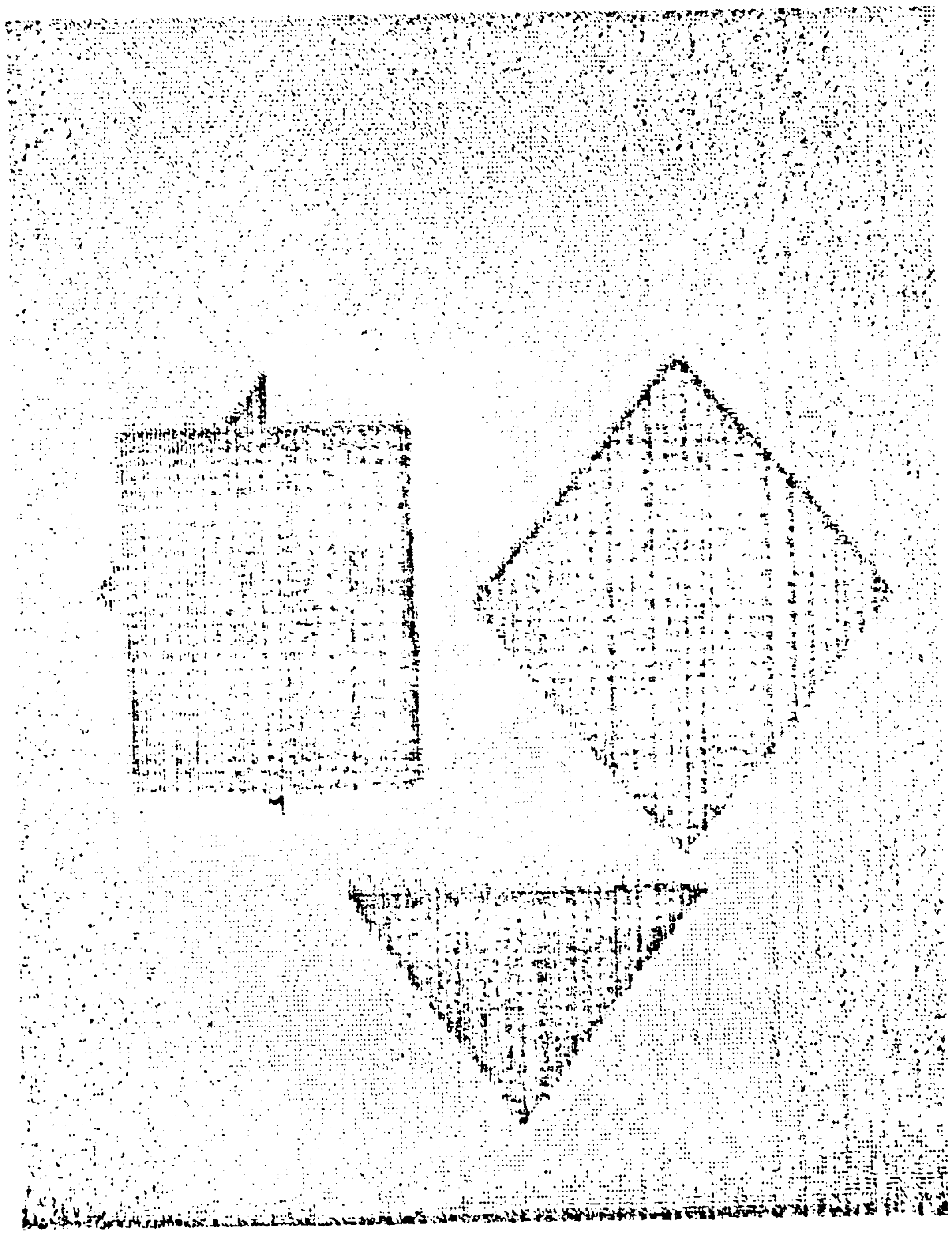

Fig. 42. Boolean OR Operation Performed Between Digitized Blocks (Fig. 1) and Digitized Triangle (Fig. 42)

3) Boolpass Operators

The Boolpass operators use specific combinations of those processing operations demonstrated in the preceeding two subsections. A detailed explanation of the development and functioning of the Boolpass operators was already given in Section III and will not be reiterated here.

Appendix A contains examples of processed image results using two different Boolpass operators. These operators, outlined below and labelled simply Boolpass operator 1 and Boolpass operator 2, are applied with the same parameters and threshold to a number of different images. This approach to presenting the results permits a comparison of each operator's effectiveness when applied to a variety of input images. The results presented in Appendix A include both types of low pass spatial filtering: one dimensional and two dimensional (as demonstrated in subsection 1)). Particular note should be taken of how the threshold value (used in conjunction with the mean and standard deviation of the original picture) affects the final processed image.

The following two image processing (procedural) outlines detail the exact processing steps by which the Boolpass operator results in Appendix A were obtained. The generalized forms such as Fig. A, Fig. B, etc. are replaced by actual figure references in the abbreviated outlines preceeding each set of results in Appendix A.

AD-A118 076   AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC   F/G 20/6
COMBINED SPATIAL FILTERING AND BOOLEAN OPERATORS APPLIED TO THE--ETC(U)
JUN 82   B E FELTMATE

UNCLASSIFIED   AFIT/GCS/EE/82J-8   NL

2 of 4

AD A118076

<u>Outline for Boolpass Operator 1</u>

1. Original picture processed : Fig. A

   Step 1

   Apply 2-dimensional 11x11 low pass filter to original picture, Fig. A

2. Resulting output image : Fig. B

   Step 2

   Form negative image of Fig. B (ouput image from step 1)

3. Resulting ouput image : Fig. C

   Step 3

   Perform 'AND' operation using Fig. C as input 'ONE' to the AND operator and Fig. A as input 'TWO' to the AND operator

   Case 1. Threshold = (1/2 the Standard Deviation of the pixel values for the entire picture, Fig. A) / 10

4. Resulting ouput image : Fig. D

   Case 2. Threshold = (the Standard Deviation of the pixel values fot the entire picture, Fig. A) / 10

5. Resulting ouput image : Fig. E

   NOTE : Pixels stored in output image of AND operation are taken from input 'TWO'

6. Other images : as specified

This same operation is repeated exactly for a 1-Dimensional low pass filter. The only change in the processing procedure occurs in Step 1, as follows:

Step 1

Apply a 1-Dimensional 1x11 low pass filter to original picture, Fig. A

An abbreviated form of this outline, listing only the nu.bers 1 thru 6 and the corresponding correct Figures, is included before the results for any particular image processed.

Outline for Boolpass Operator 2

1. Digitized picture processed : Fig. A

Step 1

Apply 2-dimensional 7x7 low pass filter to original picture, Fig. A

2. Resulting output image : Fig. B

Step 2

Apply 2-dimensional 3x3 low pass filter to original picture, Fig. B

3. Resulting output image : Fig. C

Step 3

Perform 'AND' operation using Fig. B as input 'ONE' to the AND operator and Fig. C as input 'TWO' to the AND operator.
Threshold = Ø, ie. there is no threshold window

4. Resulting output image : Fig. D

Form negative image of Fig. D (ouput image from step 3)

5. Resulting ouput image : Fig. E

Step 4

Perform 'AND' operation using Fig. E as input 'ONE' to the AND operator and Fig. A as input 'TWO' to the AND operator.

Case 1. Threshold = (the Standard Deviation of the pixel values for the entire picture, Fig. A) / 1Ø

6. Resulting ouput image : Fig. F

Case 2. Threshold = Ø, ie. no threshold window

7. Resulting ouput image : Fig. E

NOTE : Pixels stored in output image of AND operation are taken from input 'TWO'

8. Other images : as specified

This same operation is repeated exactly for a 1-Dimensional low pass filter. The only change in the processing procedure occurs in Step 1 and Step 2 as follows:

Step 1

Apply a 1-dimensional 1x7 low pass filter to original picture, Fig. A

Step 2

Apply a 1-dimensional 1x3 low pass filter to original picture, Fig. A

An abbreviated form of this outline, listing only the numbers 1 thru 8, and the corresponding correct Figures, is included before the results for any particular image processed.

Appendix B contains a three dimensional pixel intensity (or pixel/picture energy) plot for each of the original digitized pictures used in Appendix A. An example three dimensional energy plot is also given for one of the Boolpass Operator 1 output images. These three dimensional plots help demonstrate and emphasize the amount of picture energy reduction attained through use of a Boolpass operator.

During the research and testing of the Boolpass operator concept, several other possible useful combinations for Boolpass operators were observed. Time and resource limitations did not allow for more extensive testing of these operators, however, the initial results obtained did warrant further examination for possible future development. Therefore, Appendix C was added to include the results of several of these other possible Boolpass operators and may form the basis for future research and testing.

As previously noted, the examples given in Appendix A are not intended to represent an exhaustive study of the Boolpass operator concept. These pictures were selected, and their Boolpass operation detailed, because (in the mind of the author) they appeared to have the greatest potential for use in an image processing/pattern recognition machine.

For clarity and ease of reference, all results are presented in the following format:

a) Boolpass operator outline (of image processed)
b) digitized image
c) intermediary processed images
   (in sequential order of processing)
d) final output image(s) of Boolpass operator

As noted previously, pages vi thru xviii contain a detailed listing of all pictures displayed in Section IV - Results and Appendices A thru C. This list of figures may be of use to the reader as an additional reference aid.

V. CONCLUSIONS and RECOMMENDATIONS

Development and testing of Boolpass operators, as a new technique for real image processing, was presented in this thesis. The task of the Boolpass operators was one of picture energy reduction, while retaining the fundamental picture primitives such as edges. The Boolpass operators appeared to accomplish this energy reduction task, with a notable degree of constancy, over a number of very different images.

Five factors which had the greatest influence during the overall development and testing of the Boolpass operators are worthy of note. These factors were as follows:

1) The use of real time interactive programming techniques proved to be a powerful tool for image processing operator development. The ability to see (the output pictures), analyze, and respond quickly, throughout all stages of the processing, was instrumental in the development of the Boolpass operators.

2) The nondestructive processing of the original image by use of additional ouput image files allowed the original image to remain unchanged for further processing or reference by a target recognition or feature extraction algorithm.

3) In those pictures where multiple-levels of processing were involved, the retention of the image greylevel values,

as opposed to going to a binay picture representation, was an important factor in the 'succesful' application of a Boolpass operator. Also, as was explained in Section III, the choice of which input image pixel values should be stored in the output image file can have a profound effect on the further processing of that image.

4) The size and type of low pass filter which is used in processing an image can be used to bias an image for certain, specific features or orientations of those features. This was illustrated in several of the results where only one dimensional filtering in the horizontal axis was used. In such images, the Boolpass operators retained and emphasized horizontal lines while supressing many vertical edges. This type of selective filtering could have important applications for aiding target recogntion. Similarly, with some insight into a particular pattern/target recognition problem, design or choice of a low pass filter specific to the user's requirements could further improve the Boolpass operator performance.

5) The use of negative images is an important step which should not be omitted if large energy reduction is desired through use of the Boolpass operators in image processing.

This thesis has presented the initial development and testing of a new technique, the Boolpass operators, for future use in image processing. It is recommended that further testing and refinement of these operators be continued and a more quantitative evalutaion of their effectiveness be conducted. More specifically, it is recommended that further research include at least these following areas:

1) The Boolpass operators should be exhaustively tested on a very wide range of pictures, with large variations in pixel mean and standard deviation for the pictures.

2) Other types of thresholding techniques should be investigated, particularly a locally adaptive threshold technique.

3) Different types of low pass filtering should be exercised.

4) More combinations of multilevel processing should be attempted. This might yield useful results when used with a different filtering technique.

5) The performance of the Boolpass operators should be quantitatively compared to existing edge operators such as the Sobel and Roberts operators.

6) Research into the possible use of additional, but different operations in the Boolpass technique should be investigated. For example, using high pass filtering of the images in some intermediary stage of the processing. It might also be found that other Boolean operations, such as exclusive OR, may be of some use in multiprocessing the images.

From the results given herein, the concept and techniques of the Boolpass operators readily demonstrate a potential for future development and possible use in a pattern recognition machine.

Bibliography

1. Bullock, Bruce. The Performance of Ege Operators On Images With Texture. Hughes Research Laboratories, Malibu, California, October 1974. (AD/A - 006-230) [This report is available from the Defense Documentation Center.]

2. Castleman, Kenneth R. Digital Image Processing. Englewood Cliffs, N.J.: Prentice Hall, Inc., 1979.

3. Chow, C. K. and Kaneko, T. "Automatic Boundary Detection of the Left Ventricle from Cineangiograms". Computers and Biomedical Research, 5: 388-410 (1972).

4. Dudani, Sahibsingh A. An Experimental Study of Moment Methods for Automatic Identification of Three-Dimensional Objects From Television Images. Columbus, Ohio: Department of Electrical Engineering, Ohio State University, August 1973. (AD 771 662) [The report is available from the Defense Documentation Center.]

5. Hueckel, Manfred H. "A Local Visual Operator Which Recognizes Edges and Lines". Jounal of the Association for Computing Machinery, Vol 20, No. 4: 634-647 (October 1973).

6. Kanal L. N. and Agrawala A. K. "Pattern Recognition" in Encyclopedia of Computer Science (First Edition), edited by Anthony Ralston. New York: Van Nostrand Reinhold Company, 1976.

7. Korfhange R. R. "Boolean Algebra" in Encyclopedia of Computer Science (First Edition), edited by Anthony Ralston. New York: Van Nostrand Reinhold Company, 1976.

8. Pratt, William K. Digital Image Processing. New York: John Wiley and Sons, Inc., 1978.

9. Robinson, Guner S. "Detection and Coding of Edges Using Directional Masks". Optical Engineering, Vol 16, No. 6: 580-585 (1976).

10. Rosenfeld, Azriel and Thurston, Mark. "Edge and Curve Detection for Visual Scene Analysis". IEEE Transactions on Computers, Vol c-20, No. 5: 562-569 (May 1971).

11. Tou, J. T. and Gonzalez R. C. Pattern Recognition Principles. Reading, Mass: Addison-Wesley Publishing Company, 1979.

Appendix A

Boolpass Operators 1 and 2 - Results

Contained in this appendix are the results of applying Boolpass operators 1 and 2 to eight different digitized images. The examples which follow demonstrate the use of both one and two dimensional filtering and various threshold values. For ease of reference, the general outline of the processing steps for both Boolpass operators 1 and 2 are repeated herein.

Outline for Boolpass Operator 1

1. Digitized Image processed : Fig. A

Step 1

Apply 2-dimensional 11x11 low pass filter to original picture, Fig. A

2. Resulting output image : Fig. B

Step 2

Form negative image of Fig. B (ouput image from step 1)

3. Resulting ouput image : Fig. C

Step 3

Perform 'AND' operation using Fig. C as input 'ONE' to the AND operator and Fig. A as input 'TWO' to the AND operator

Case 1. Threshold = (1/2 the Standard Deviation of the pixel values for the entire picture, Fig. A) / 10

4. Resulting ouput image : Fig. D

Case 2. Threshold = (the Standard Deviation of the pixel values fot the entire picture, Fig. A) / 10

5. Resulting ouput image : Fig. E

NOTE : Pixels stored in output image of AND operation are taken from input 'TWO'

6. Other images : as specified

This same operation is repeated exactly for a 1-Dimensional low pass filter. The only change in the processing procedure occurs in Step 1, as follows:

Step 1

Apply a 1-Dimensional 1x11 low pass filter to original picture, Fig. A

An abbreviated form of this outline, listing only the numbers 1 thru 6 and the corresponding correct Figures is included before the results for any particular image processed.

Appendix A Quick Reference to Boolpass Operators 1 and 2

Boolpass Operator 1

	Page	Figure
2 dimensional low pass filtering	89 to 136	43 to 83
1 dimensional low pass filtering	137 to 160	84 to 103

Boolpass Operator 2

	Page	Figure
2 dimensional low pass filtering	163 to 219	104 to 153
1 dimensional low pass filtering	220 to 267	154 to 195

BOOLPASS OPERATION I

1.	Digitized Image Truck	:	Fig. 43
2.	11x11 low pass filter	:	Fig. 44
3.	Negative of 2.	:	Fig. 45
4.	AND (1/2 Std Dev)/10	:	Fig. 46
5.	AND (Std Dev)/10	:	Fig. 47

Mean: 10 Standard Deviation: 4

Fig. 43. Digitized Image of Original Picture Truck

Fig. 44. Truck Processed by an 11x11 Low Pass Filter

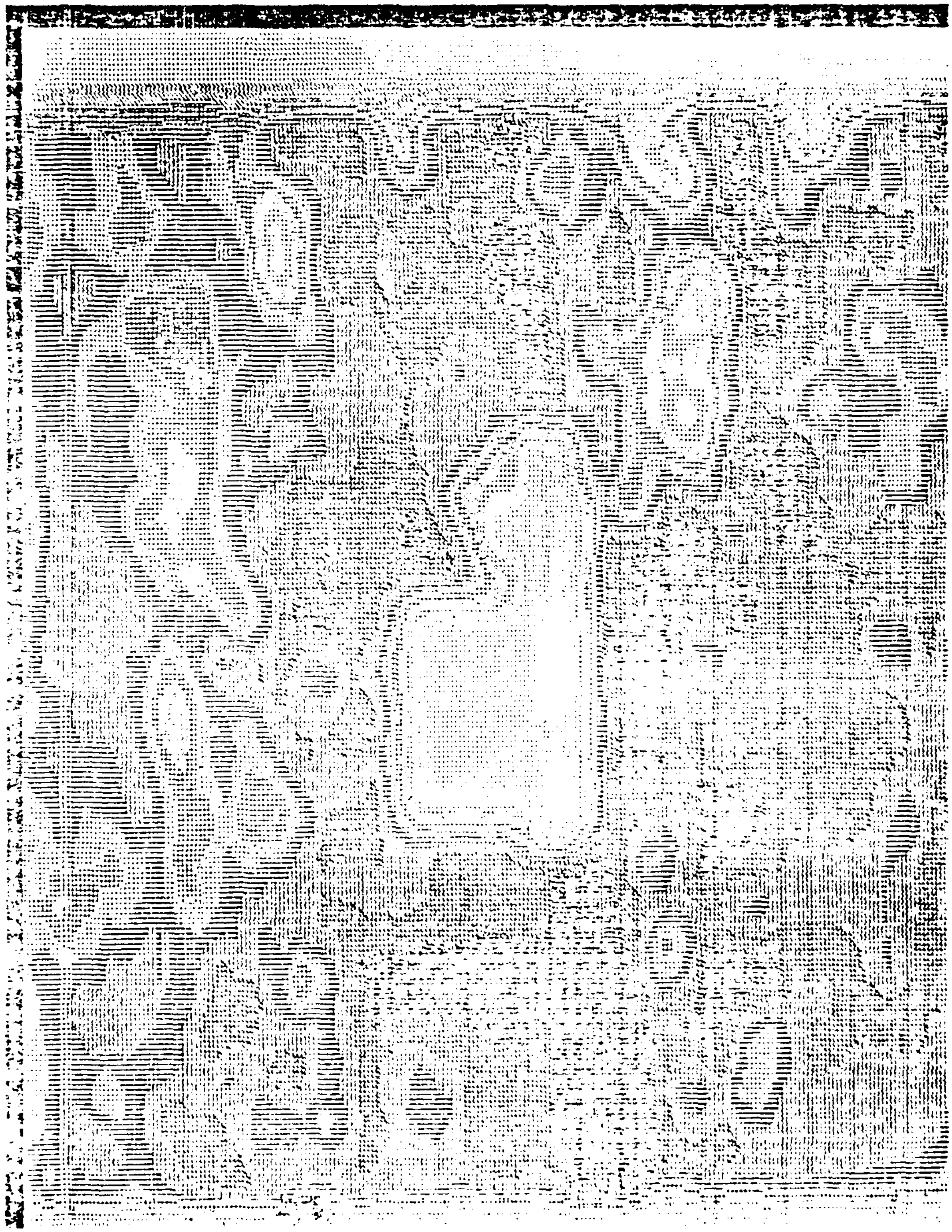

Fig. 45. Negative Image of Fig. 44

Fig. 46. Output Image of Boolpass Operation 1
Threshold = (1/2 Std Dev)/10

Fig. 47. Output Image of Boolpass Operation 1
Threshold = (Std Dev)/ 10

BOOLPASS OPERATION I

1.	Digitized Image Tomcat	:	Fig. 48
2.	11x11 low pass filter	:	Fig. 49
3.	Negative of 2.	:	Fig. 50
4.	AND (1/2 Std Dev)/10	:	Fig. 51
5.	AND (Std Dev)/10	:	Fig. 52

Mean: 10 Standard Deviation: 4

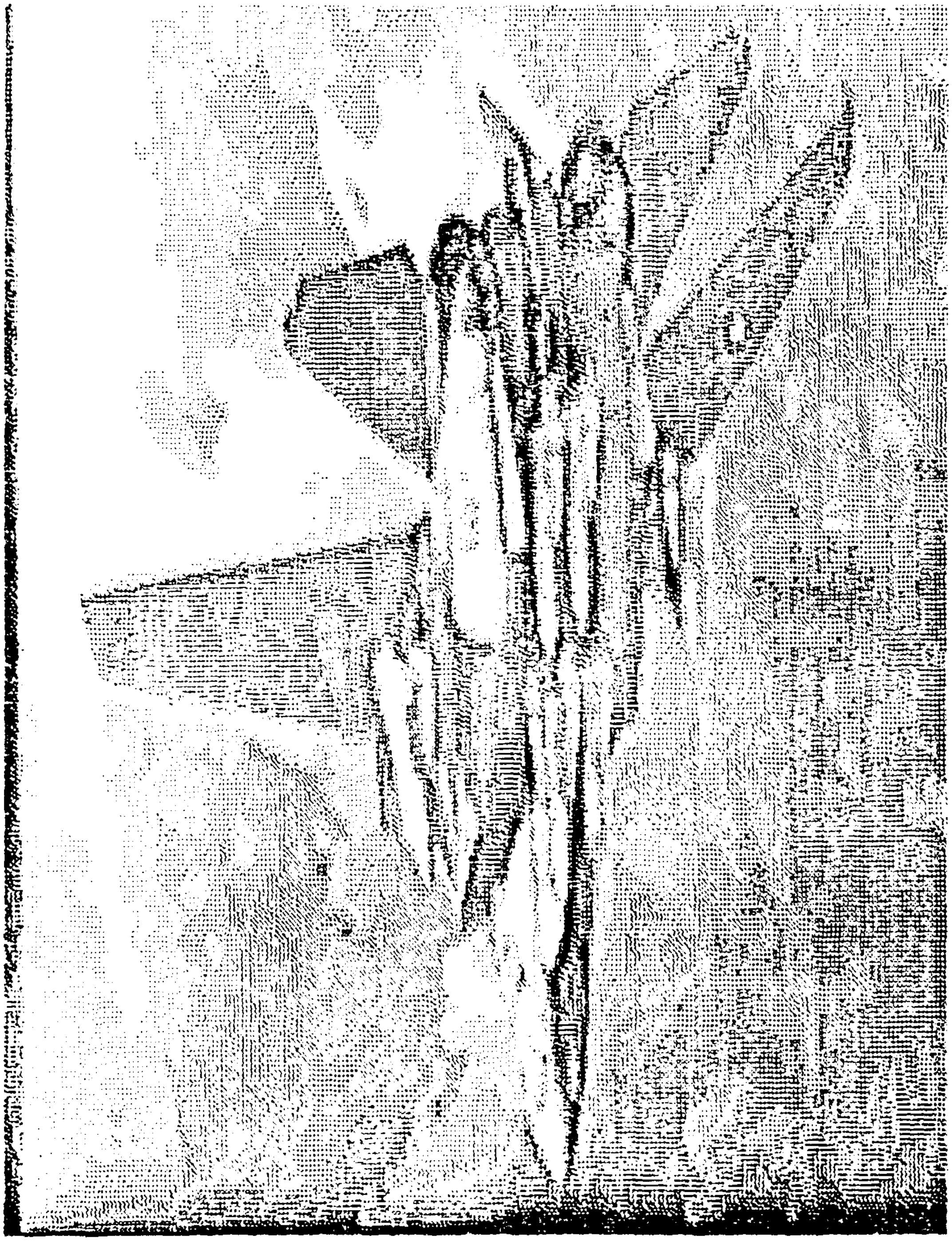

Fig. 48. Digitized Image of Original Picture Tomcat

Fig. 49. Tomcat Processed by an 11x11 Low Pass Filter

Fig. 50. Negative Image of Fig. 49

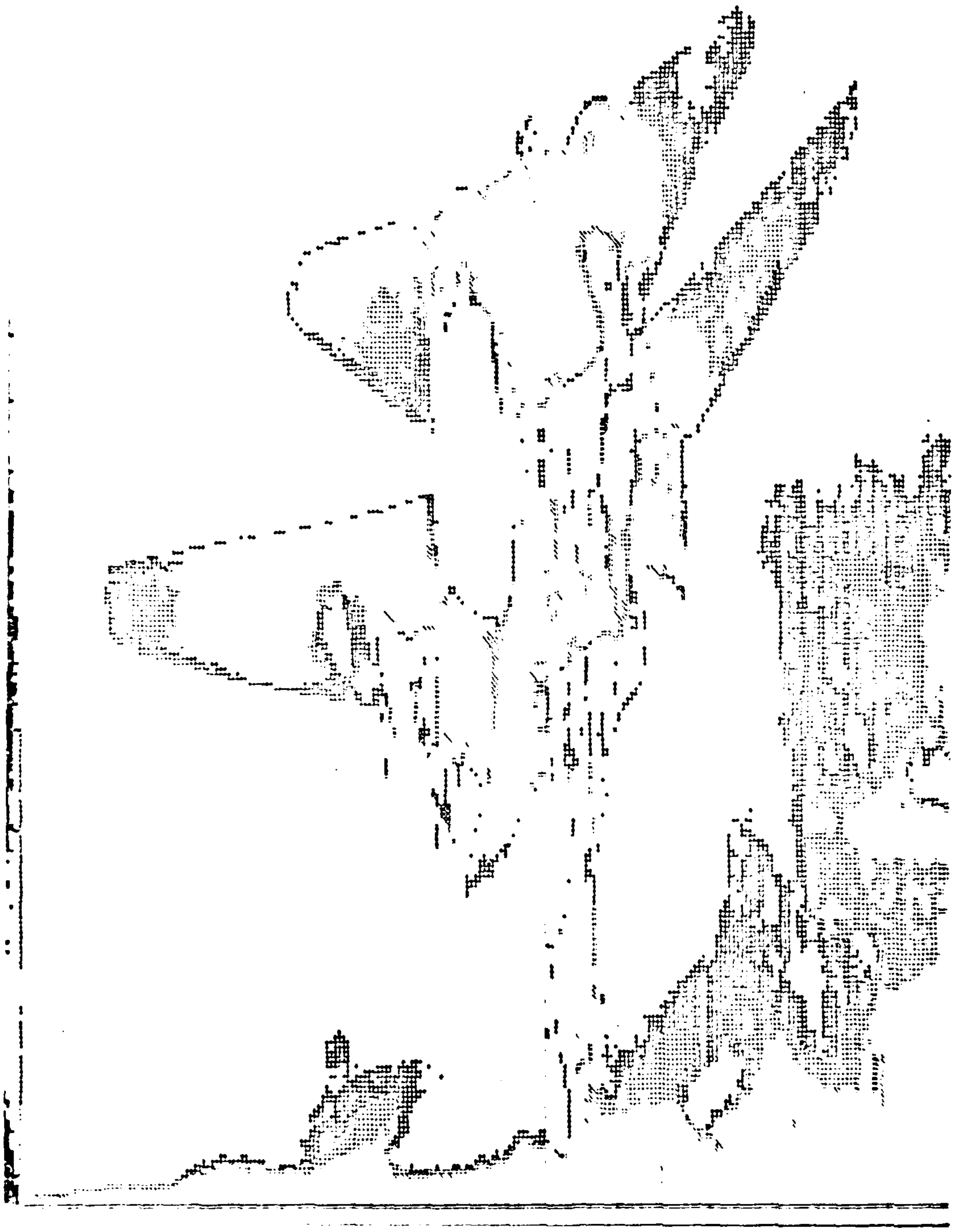

Fig. 51. Output Image of Boolpass Operation 1
Threshold = (1/2 Std Dev)/10

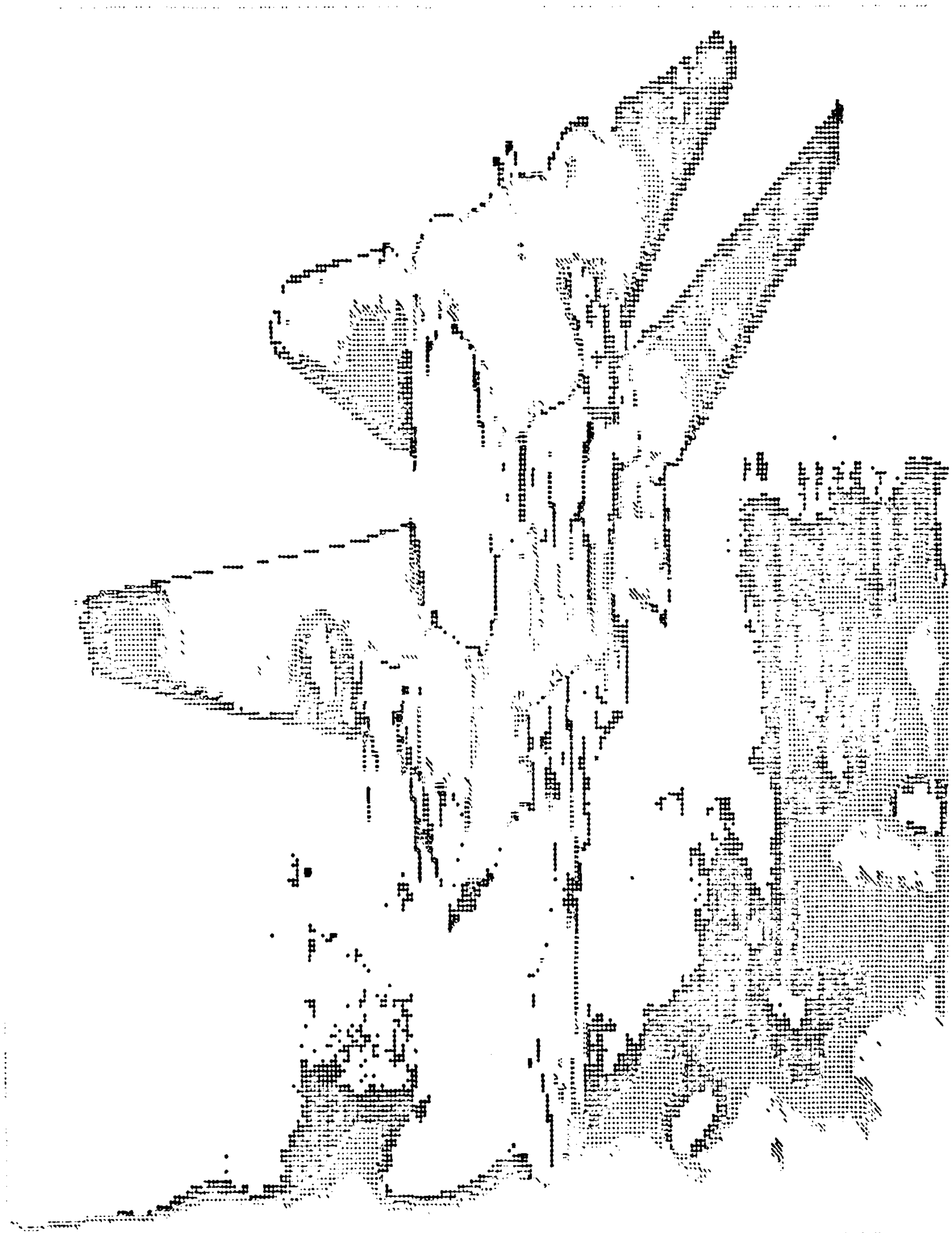

Fig. 52. Output Image of Boolpass Operation 1
Threshold = (Std Dev)/ 10

BOOLPASS OPERATION I

1.	Digitized Image Field	:	Fig. 53
2.	11x11 low pass filter	:	Fig. 54
3.	Negative of 2.	:	Fig. 55
4.	AND (1/2 Std Dev)/1Ø	:	Fig. 56
5.	AND (Std Dev)/1Ø	:	Fig. 57

Mean: 8   Standard Deviation: 5

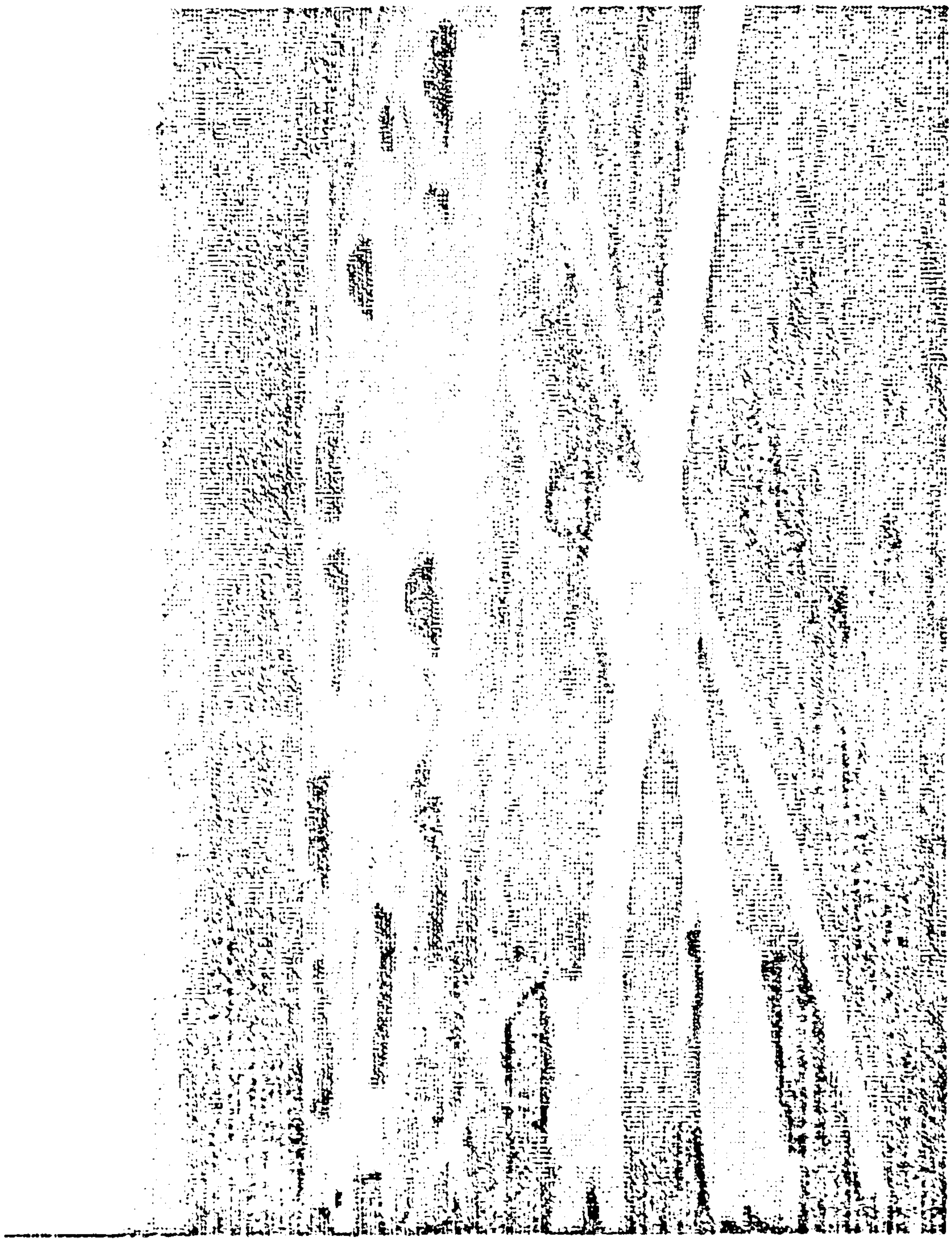

Fig. 53. Digitized Image of Original Picture Field

Fig. 54. Field Processed by an 11x11 Low Pass Filter

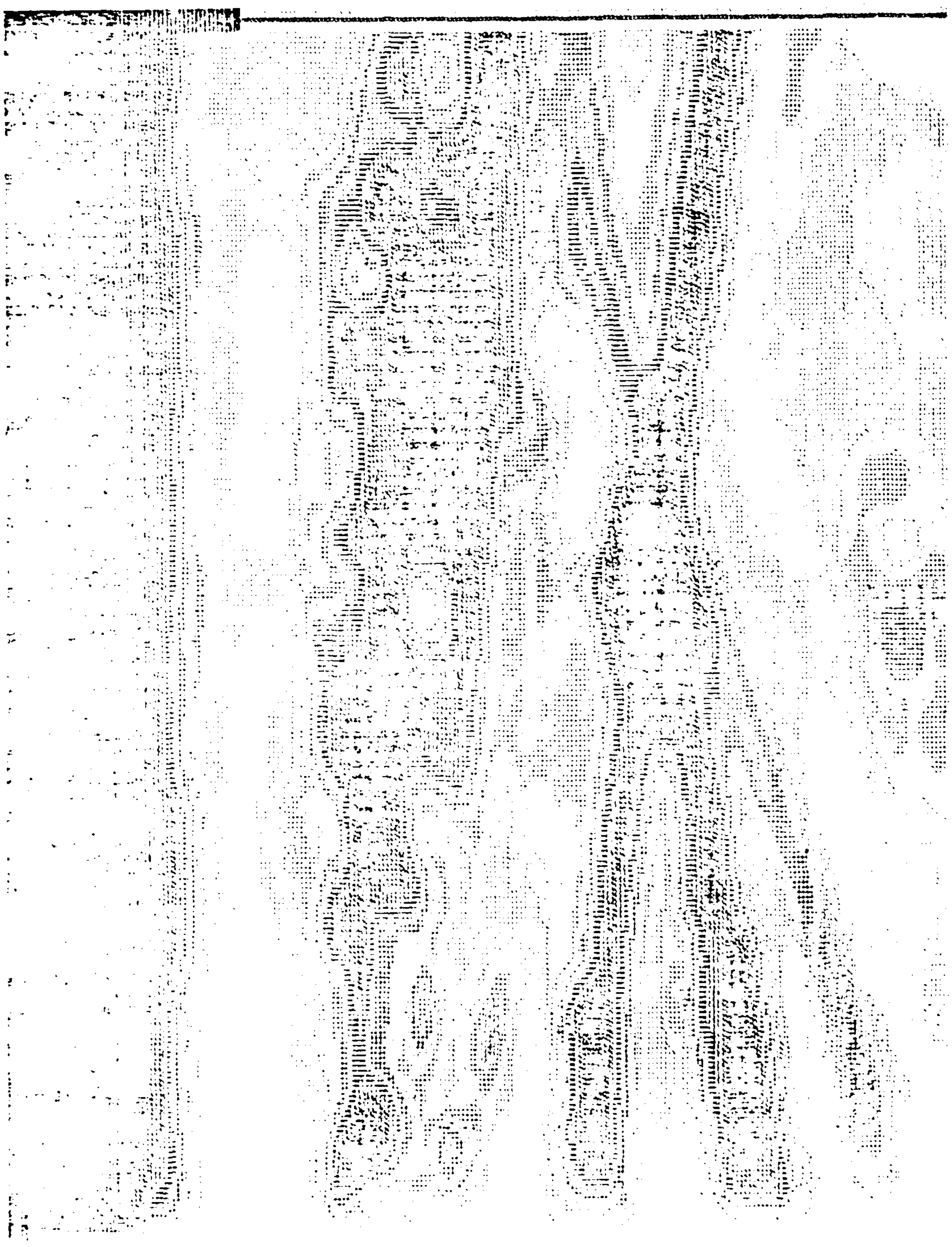

Fig. 55. Negative Image of Fig. 54

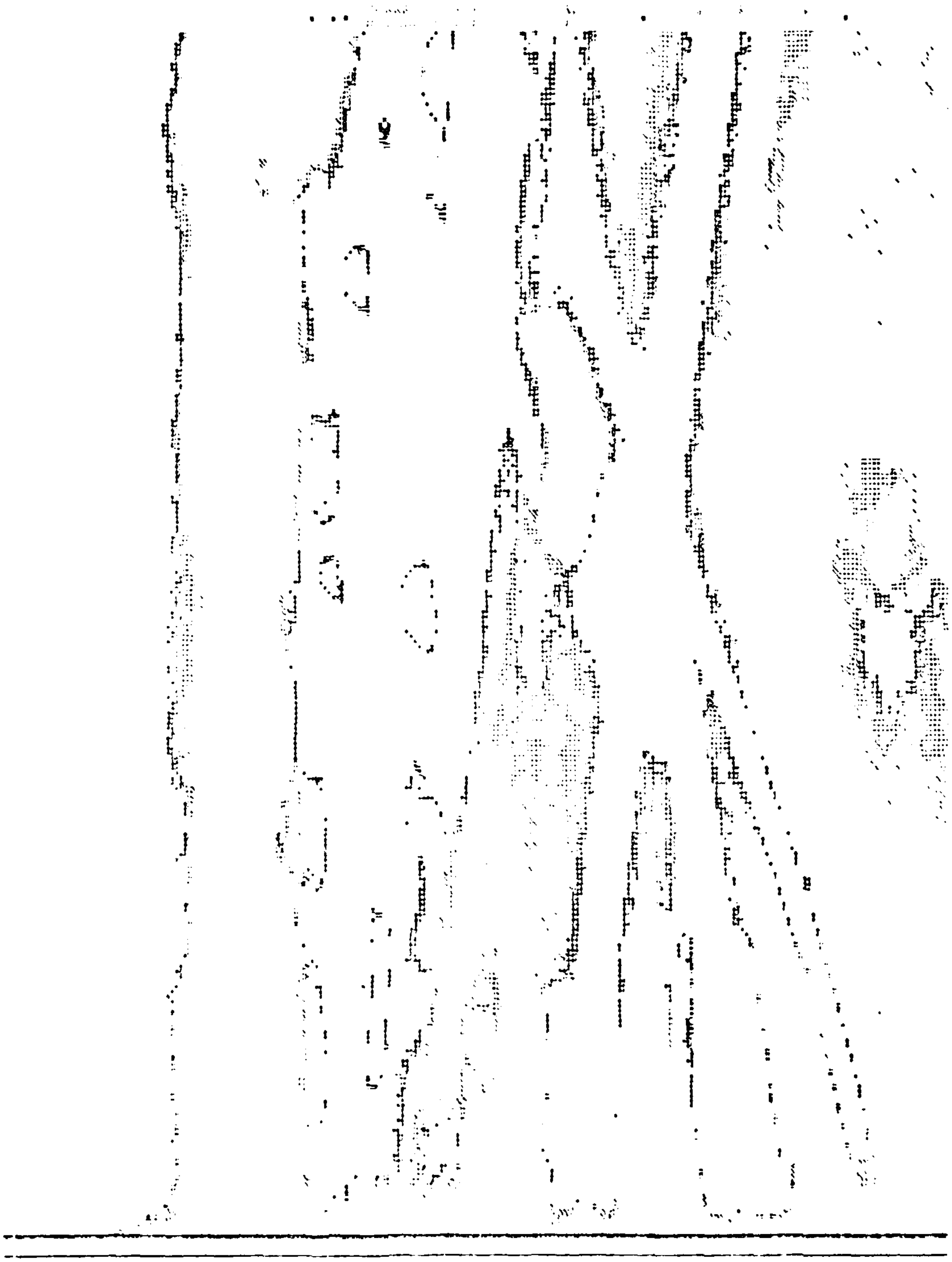

Fig. 56. Output Image of Boolpass Operation 1
Threshold = (1/2 Std Dev)/10

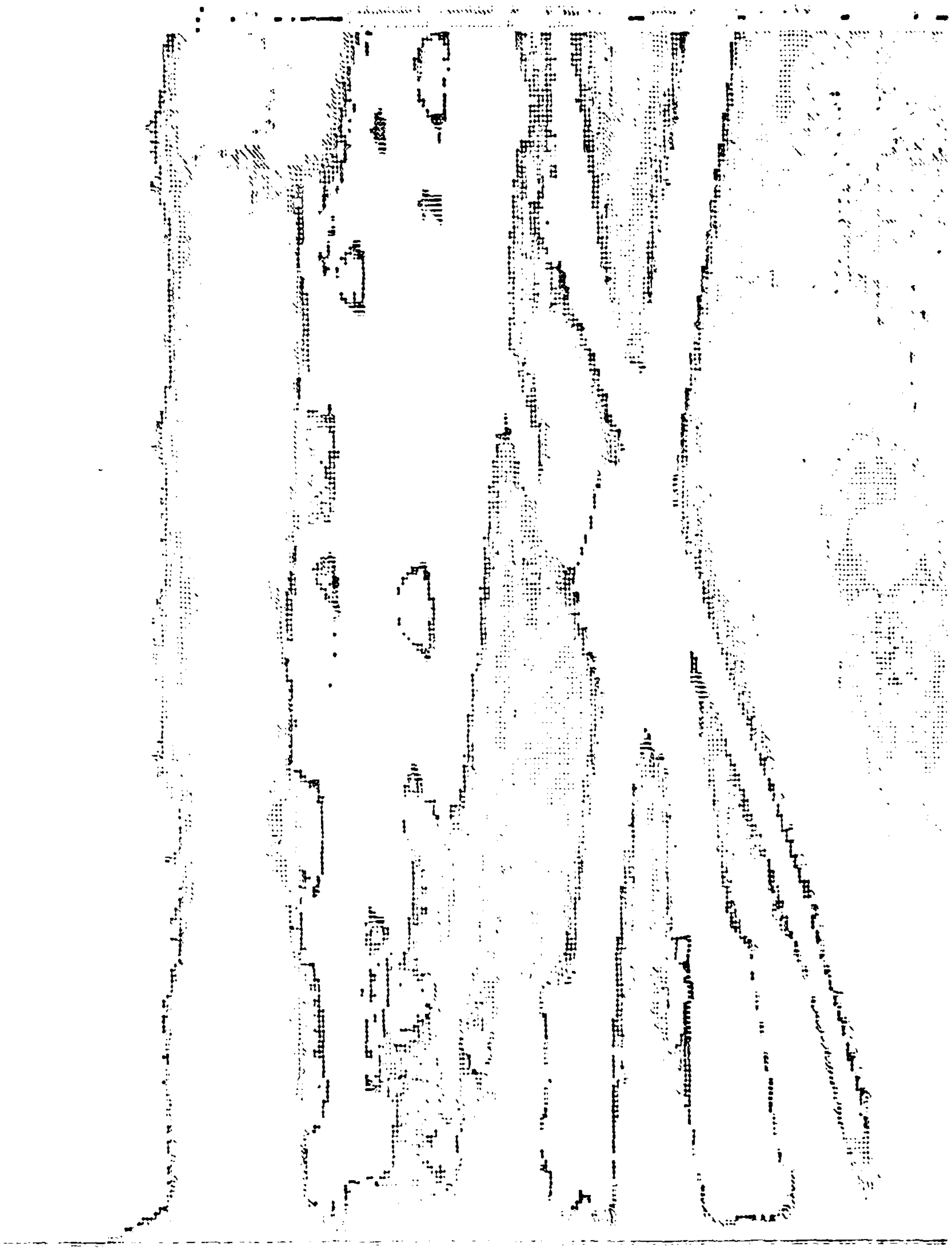

Fig. 57. Output Image of Boolpass Operation 1
Threshold = (Std Dev)/ 10

BOOLPASS OPERATION I

1.	*Digitized Image Field2*	:	Fig. 58
2.	11x11 low pass filter	:	Fig. 59
3.	Negative of 2.	:	Fig. 60
4.	AND (1/2 Std Dev)/10	:	Fig. 61
5.	AND (Std Dev)/10	:	Fig. 62

Mean: 9   Standard Deviation: 5

Fig. 58. Digitized Image of Original Picture Field2

Fig. 59. Field2 Processed by an 11x11 Low Pass Filter

Fig. 60. Negative Image of Fig. 59

Fig. 61. Output Image of Boolpass Operation 1
Threshold = (1/2 Std Dev)/1Ø

Fig. 62. Output Image of Boolpass Operation 1
Threshold = (Std Dev)/ 10

BOOLPASS OPERATION I

1.	Digitized Image Hornet	:	Fig. 63
2.	11x11 low pass filter	:	Fig. 64
3.	Negative of 2.	:	Fig. 65
4.	AND (1/2 Std Dev)/10	:	Fig. 66
5.	AND (Std Dev)/10	:	Fig. 67

Mean: 9   Standard Deviation: 3

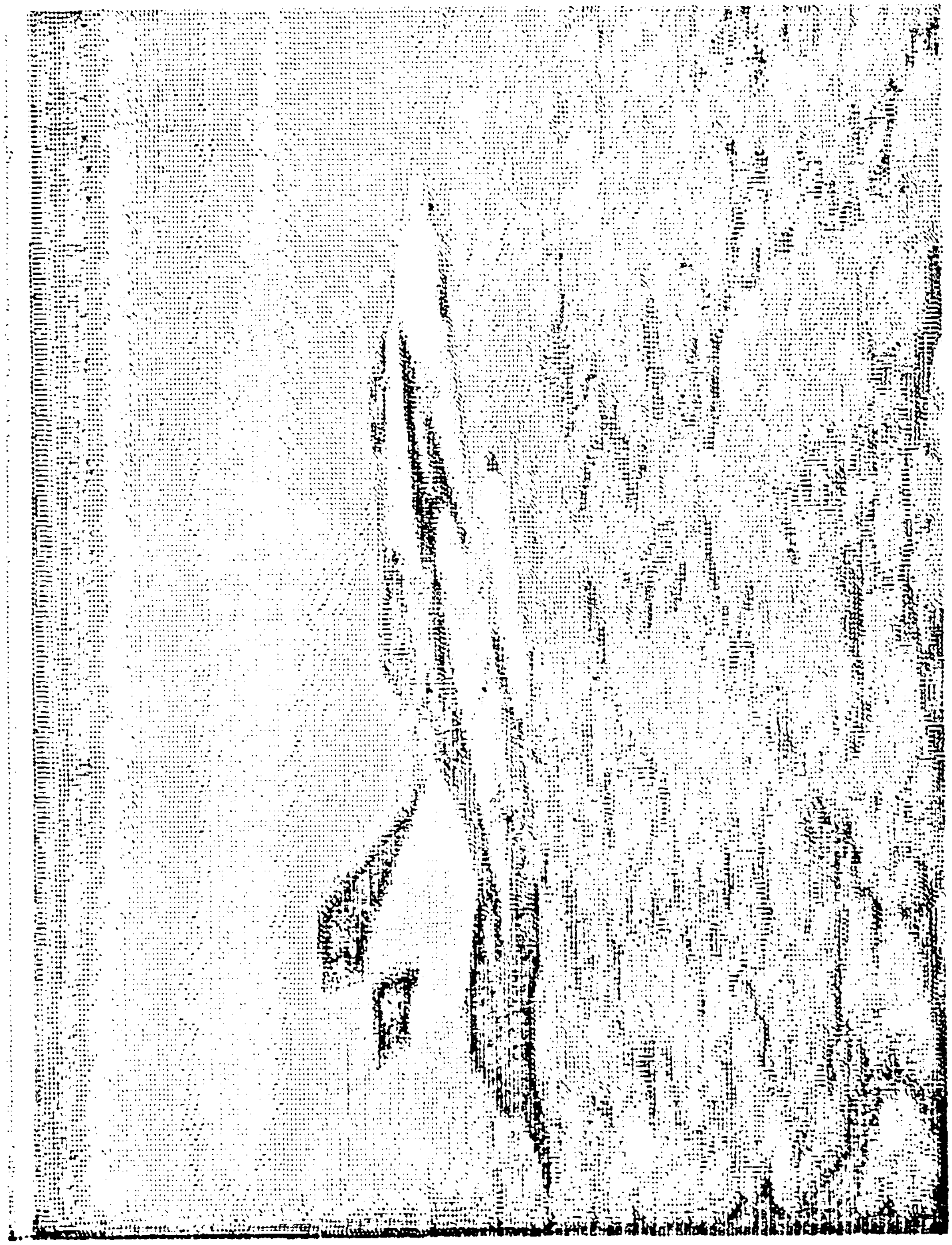

Fig. 63. Digitized Image of Original Picture F18 Hornet

Fig. 64. F18 Hornet Processed by an 11x11 Low Pass Filter

Fig. 65. Negative Image of Fig. 64

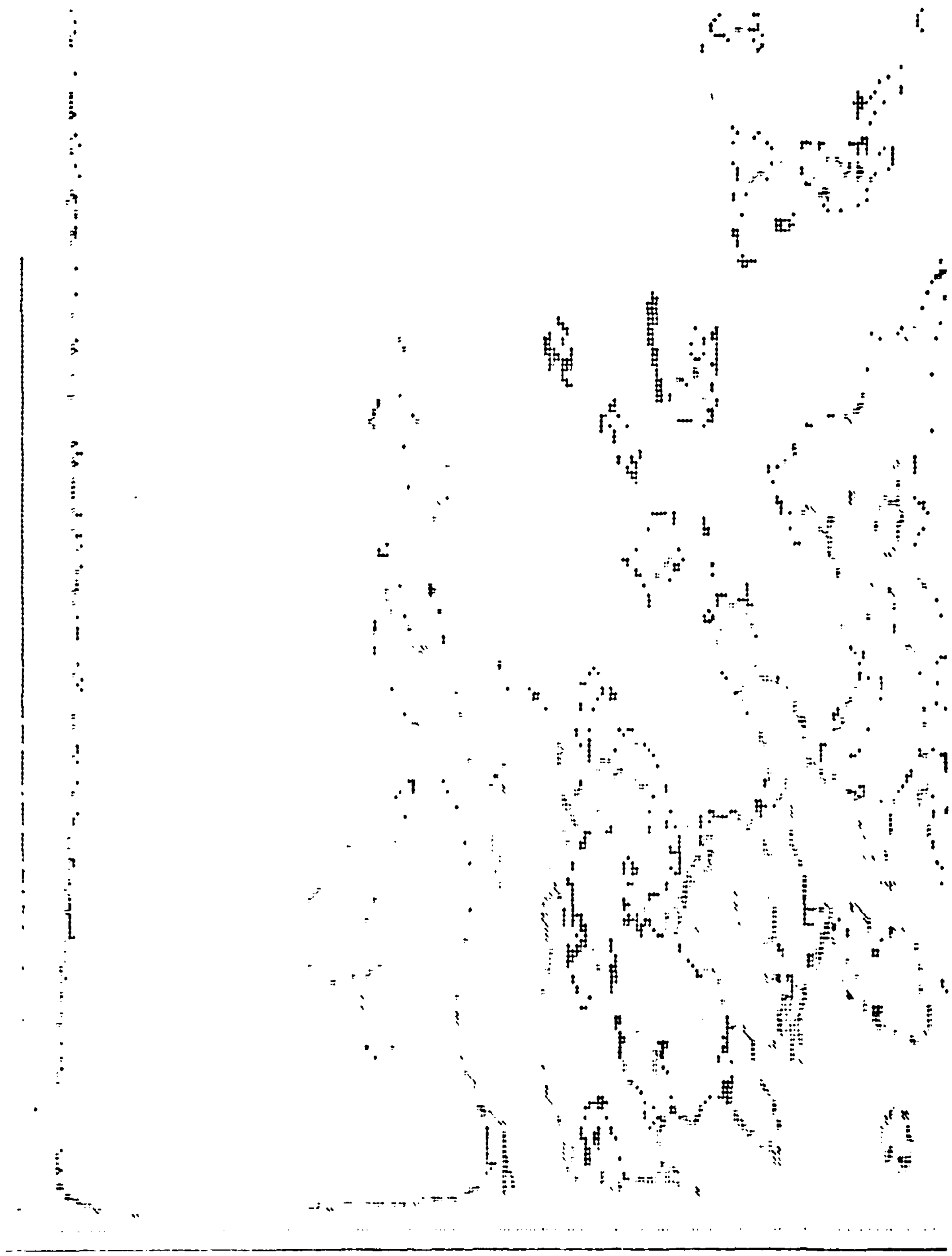

Fig. 66. Output Image of Boolpass Operation 1
Threshold = (1/2 Std Dev)/10

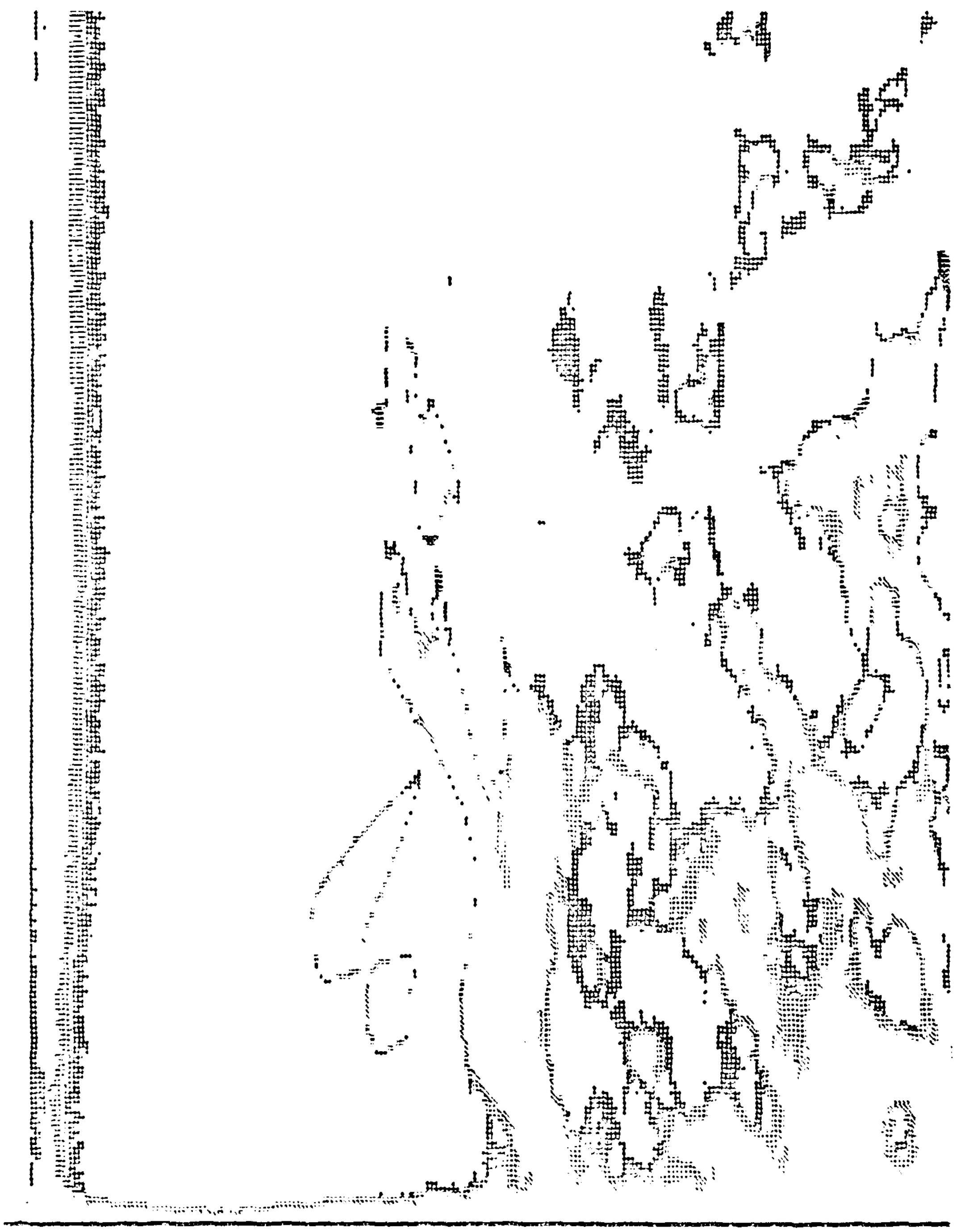

Fig. 67. Output Image of Boolpass Operation 1
Threshold = (Std Dev)/ 10

BOOLPASS OPERATION I

1.	Digitized Image Tank	:	Fig. 68
2.	11x11 low pass filter	:	Fig. 69
3.	Negative of 2.	:	Fig. 70
4.	AND (1/2 Std Dev)/10	:	Fig. 71
5.	AND (Std Dev)/10	:	Fig. 72

Mean: 10 Standard Deviation: 5

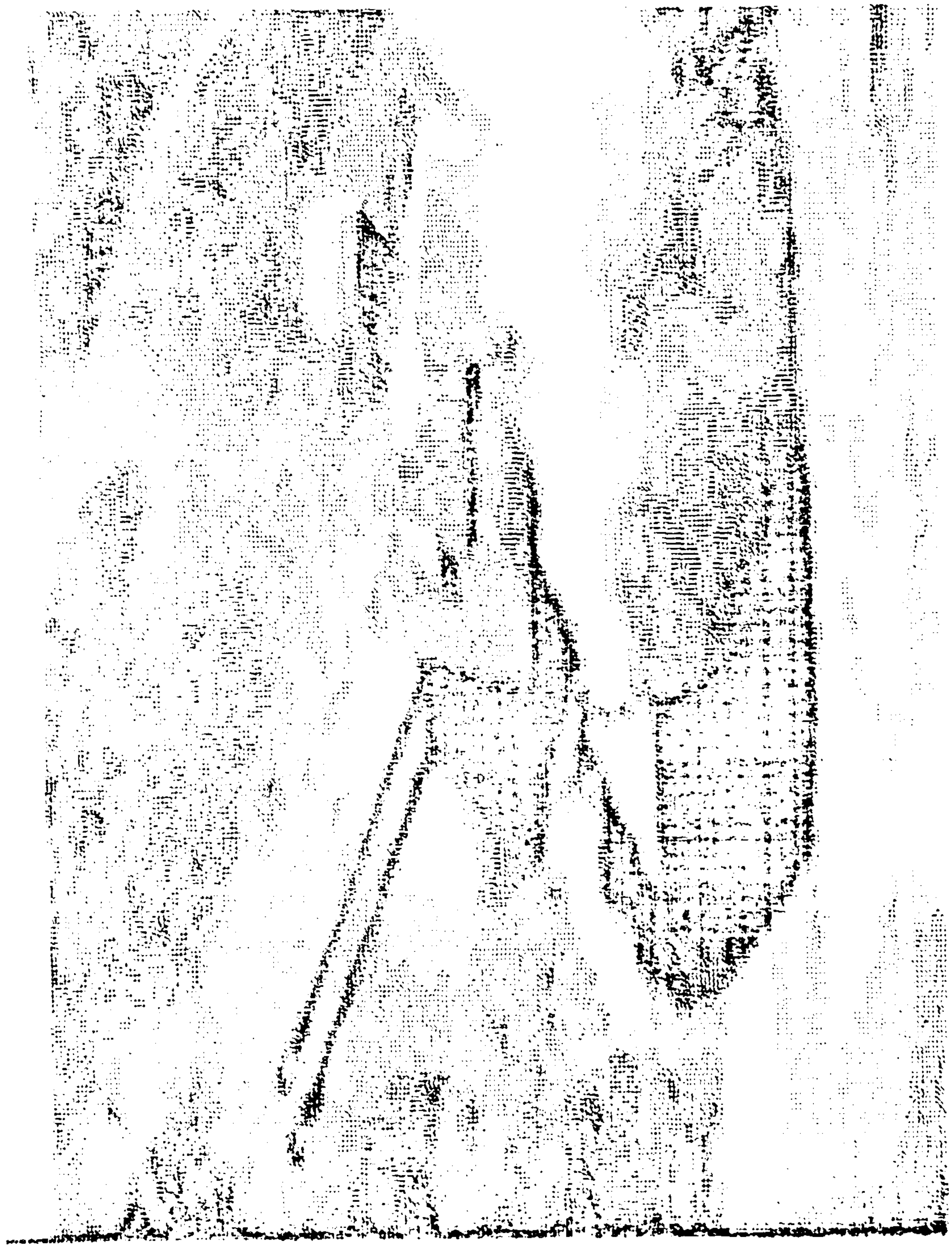

Fig. 68. Digitized Image of Original Picture Tank

Fig. 69. Tank Processed by an 11x11 Low Pass Filter

Fig. 7Ø. Negative Image of Fig. 69

Fig. 71. Output Image of Boolpass Operation 1
Threshold = (1/2 Std Dev)/10

Fig. 72. Output Image of Boolpass Operation 1
Threshold = (Std Dev)/ 10

BOOLPASS OPERATION I

1.	Digitized Image Dispersal	:	Fig. 73
2.	11x11 low pass filter	:	Fig. 74
3.	Negative of 2.	:	Fig. 75
4.	AND (1/2 Std Dev)/1Ø	:	Fig. 76
5.	AND (Std Dev)/1Ø	:	Fig. 77

Mean: 9   Standard Deviation: 5

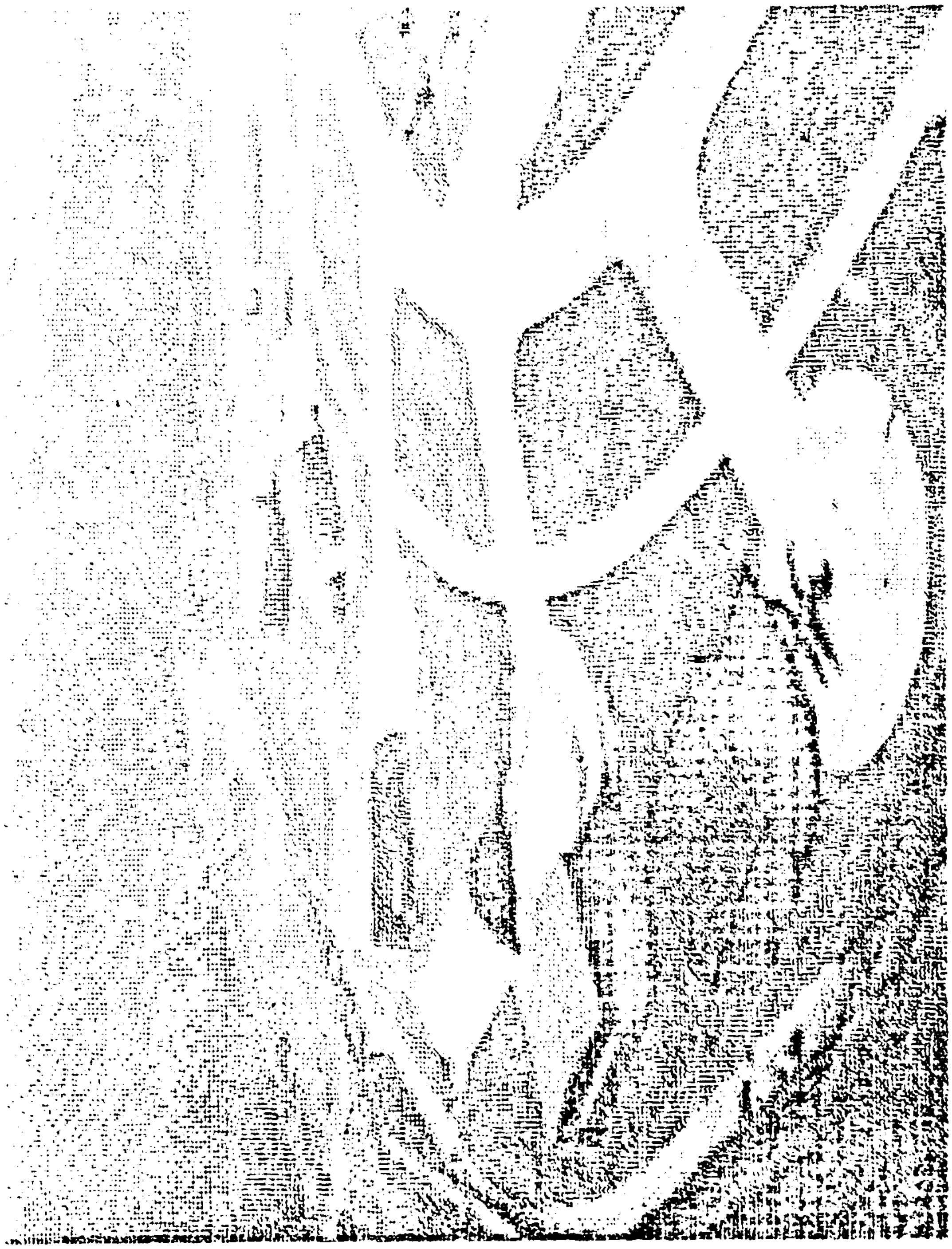

Fig. 73. Digitized Image of Original Picture Dispersal

Fig. 74. Dispersal Processed by an 11x11 Low Pass Filter

Fig. 75. Negative Image of Fig. 74

Fig. 76. Output Image of Boolpass Operation 1
Threshold = (1/2 Std Dev)/10

Fig. 77. Output Image of Boolpass Operation 1
Threshold = (Std Dev)/ 10

BOOLPASS OPERATION I

1.	Digitized Image Radar	:	Fig. 78
2.	11x11 low pass filter	:	Fig. 79
3.	Negative of 2.	:	Fig. 80
4.	AND (1/2 Std Dev)/10	:	Fig. 81
5.	AND (Std Dev)/10	:	Fig. 82
6.	AND (Threshold = .4)	:	Fig. 83

Mean: 8   Standard Deviation: 5

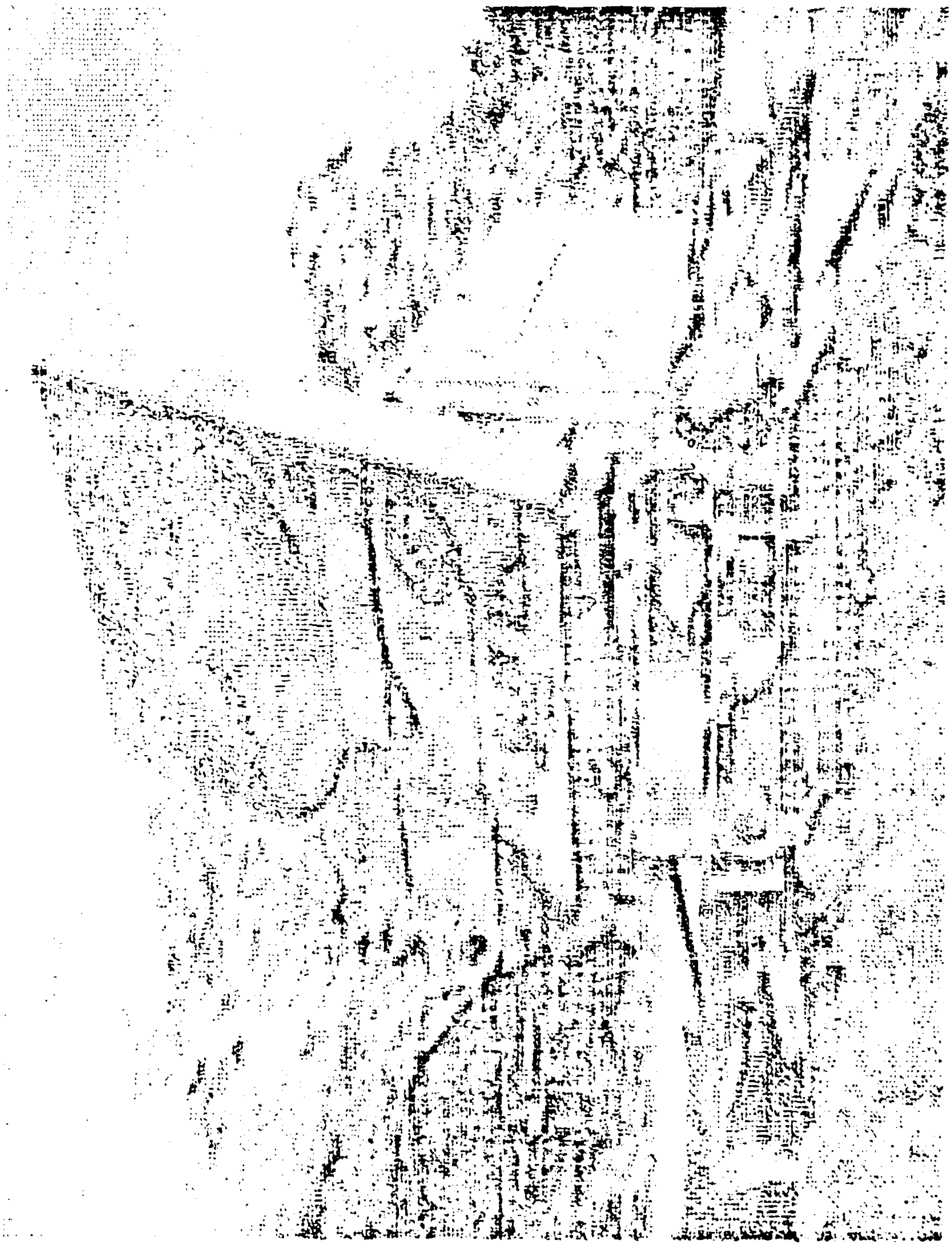

Fig. 78. Digitized Image of Original Picture Radar

Fig. 79. Radar Processed by an 11x11 Low Pass Filter

Fig. 80. Negative Image of Fig. 79

Fig. 81. Output Image of Boolpass Operation 1
Threshold = (1/2 Std Dev)/10

Fig. 82. Output Image of Boolpass Operation 1
Threshold = (Std Dev)/ 10

Fig. 83. Output Image of Boolpass Operation 1
Threshold = .4

BOOLPASS OPERATION I

1.	Digitized Image Truck	:	Fig. 84
2.	1x11 low pass filter	:	Fig. 85
3.	Negative of 2.	:	Fig. 86
4.	AND (1/2 Std Dev)/10	:	Fig. 87
5.	AND (Std Dev)/10	:	Fig. 88

Mean: 10 Standard Deviation: 4

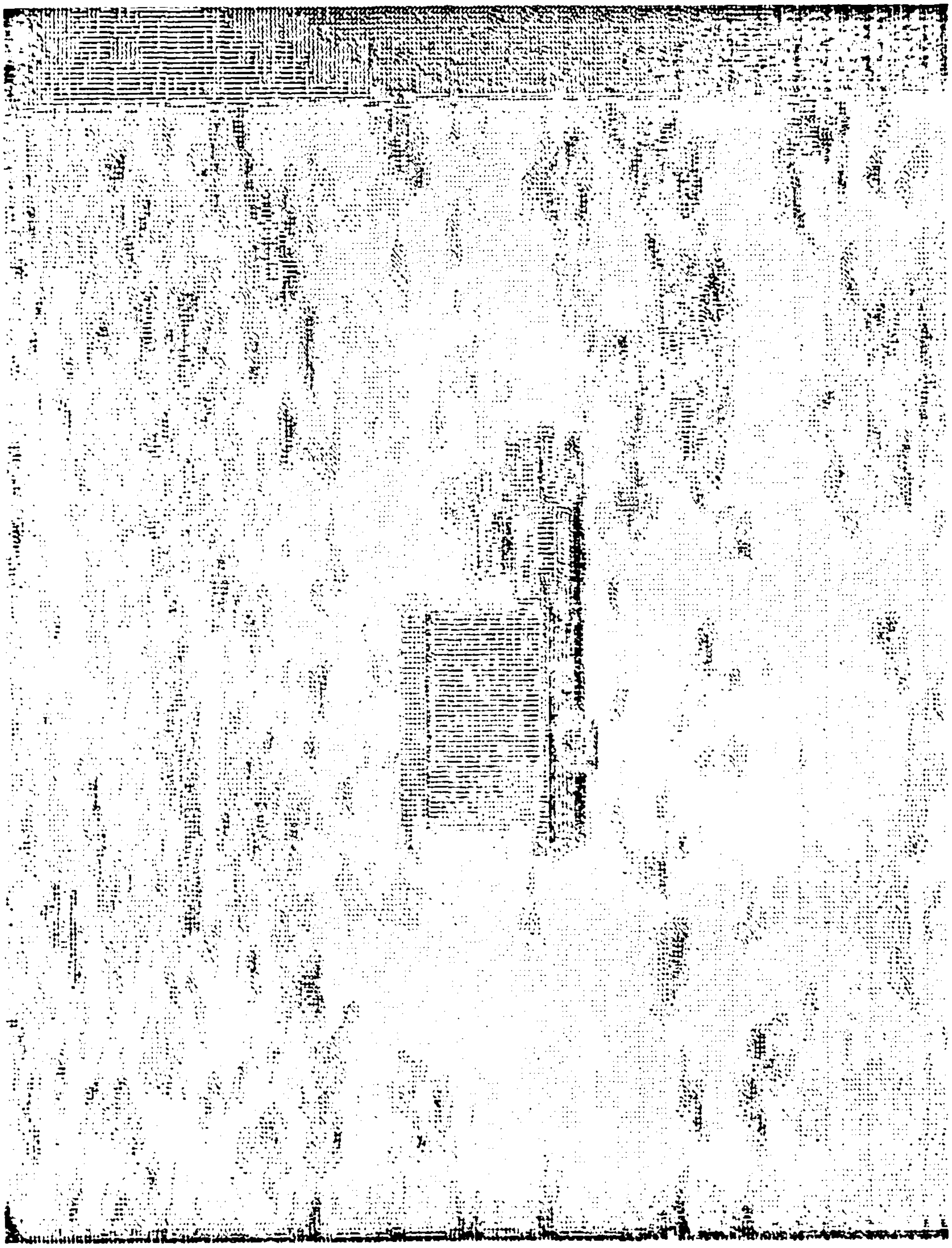

Fig. 84. Digitized Image of Original Picture Truck

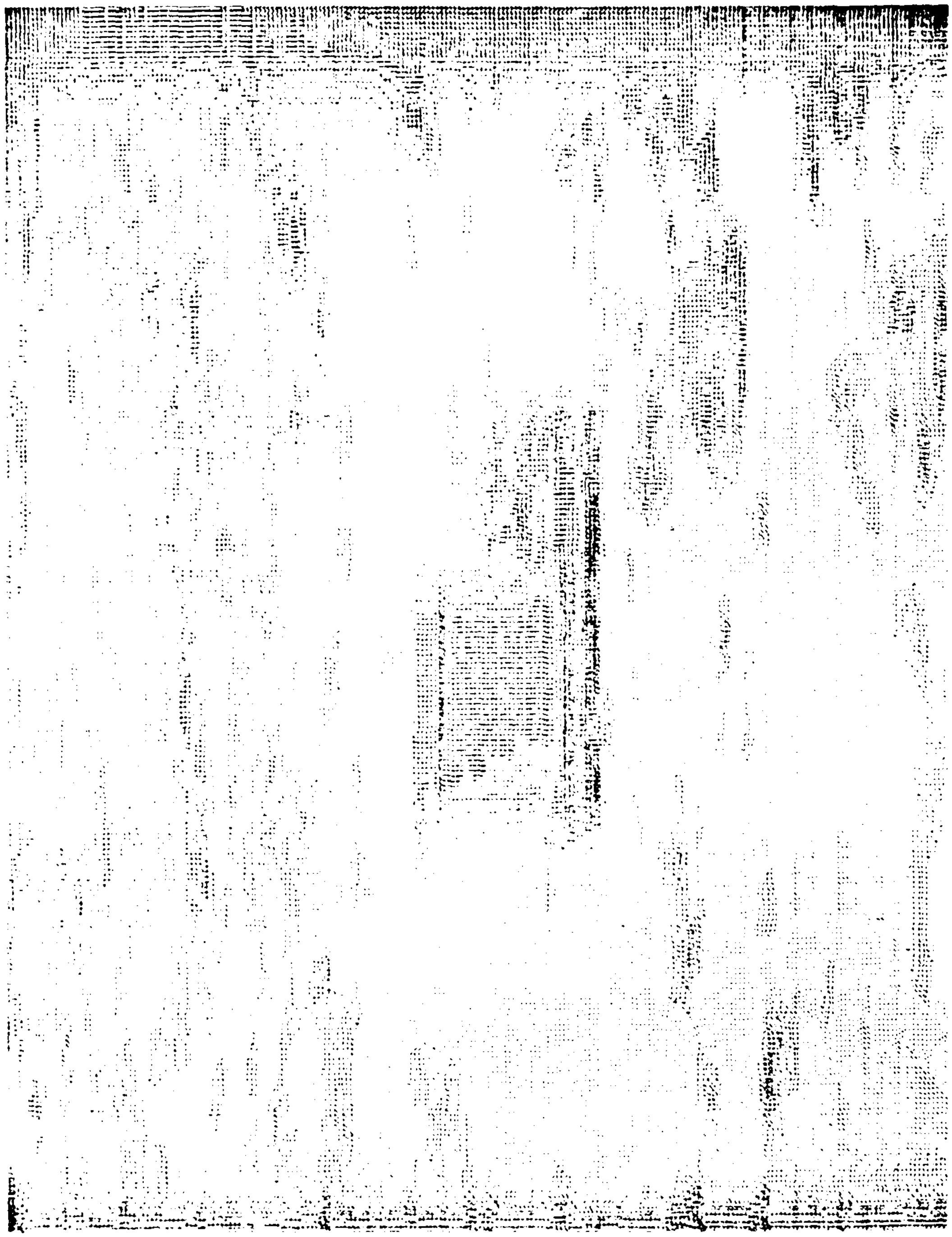

Fig. 85. Truck Processed by a 1x11 Low Pass Filter

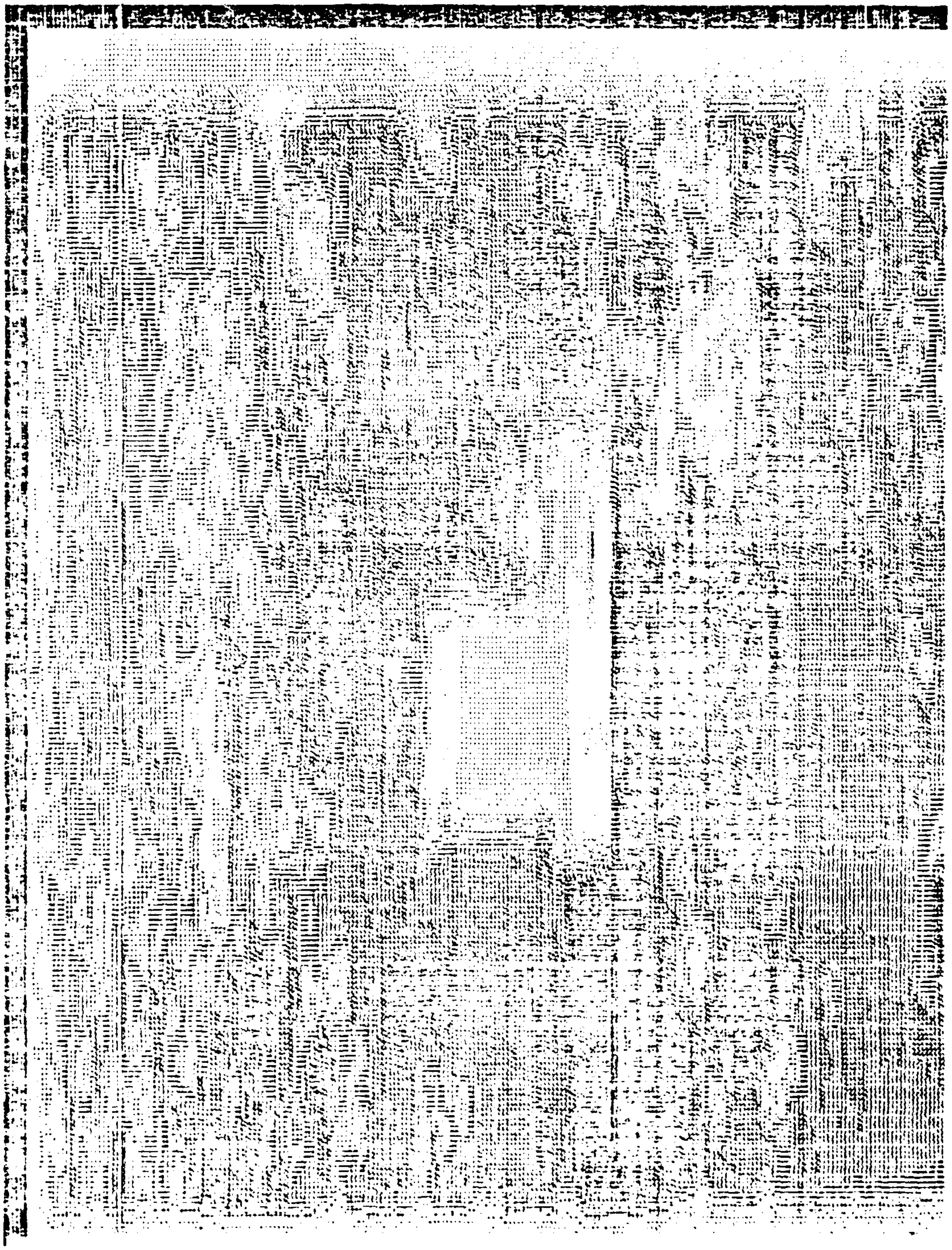

Fig. 86. Negative Image of Fig. 85

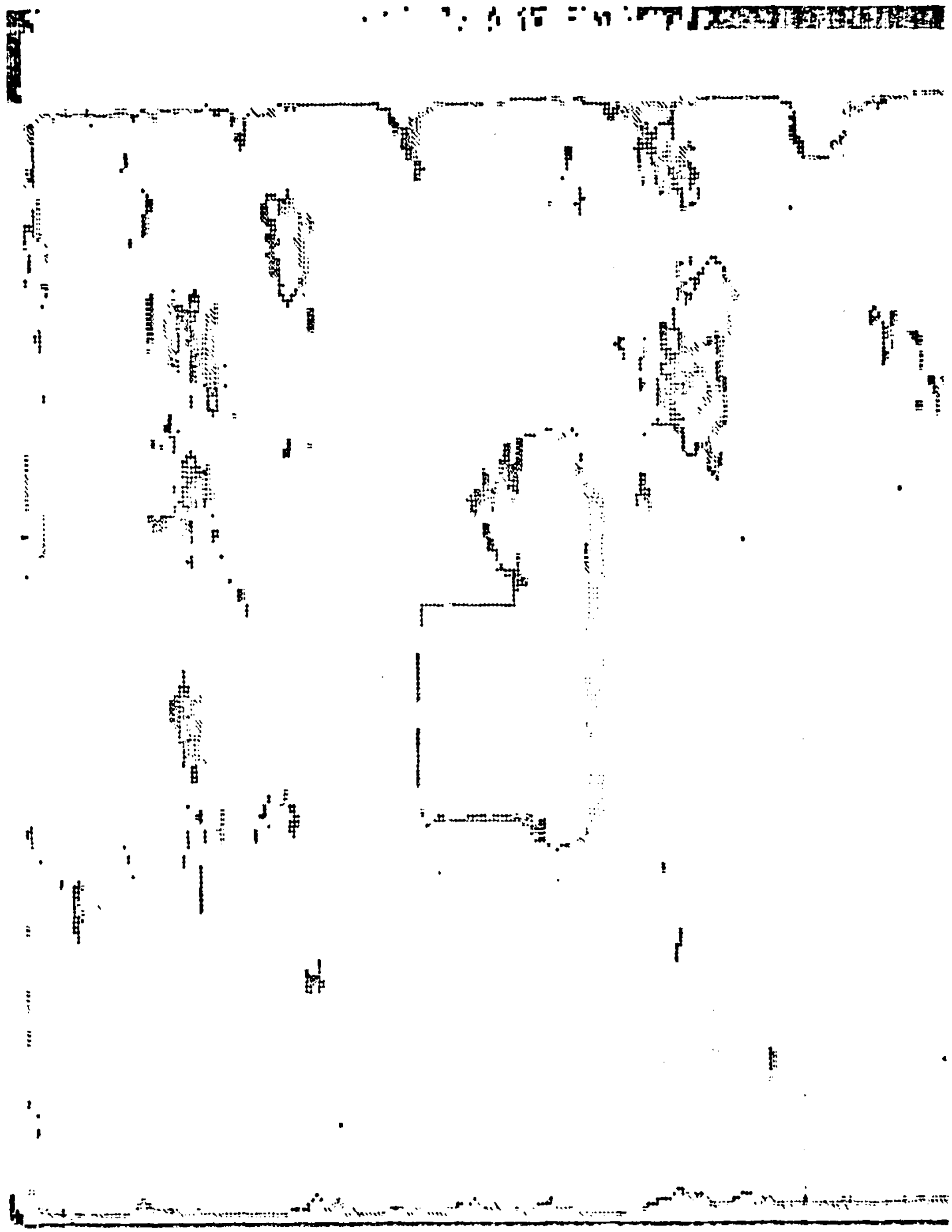

Fig. 87. Output Image of Boolpass Operation 1
Threshold = (1/2 Std Dev)/10

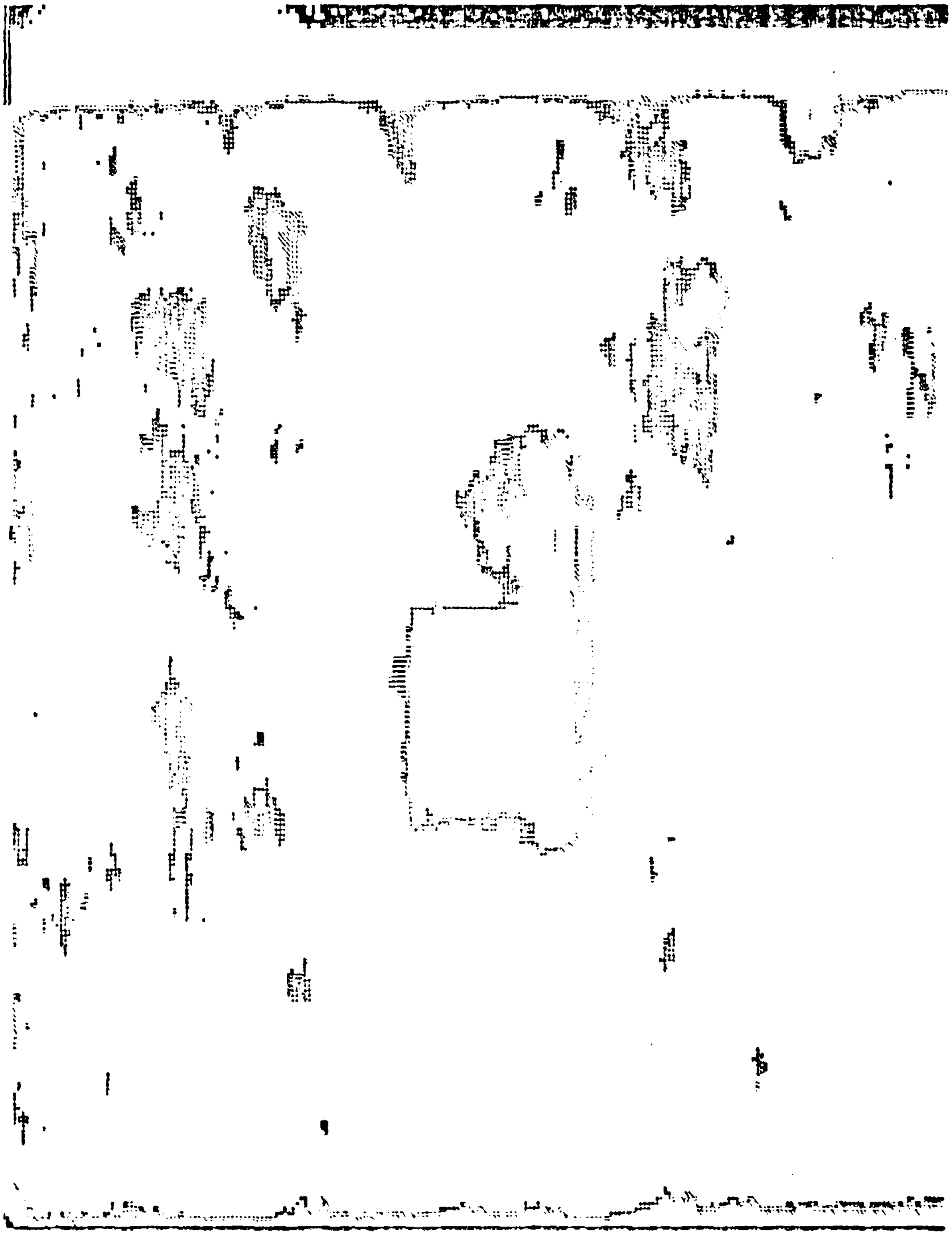

Fig. 88. Output Image of Boolpass Operation 1
Threshold = (Std Dev)/ 10

BOOLPASS OPERATION I

1.	Digitized Image Tomcat	:	Fig. 89
2.	1x11 low pass filter	:	Fig. 90
3.	Negative of 2.	:	Fig. 91
4.	AND (1/2 Std Dev)/10	:	Fig. 92
5.	AND (Std Dev)/10	:	Fig. 93

Mean: 10 Standard Deviation: 4

Fig. 89. Digitized Image of Original Picture F14 Tomcat

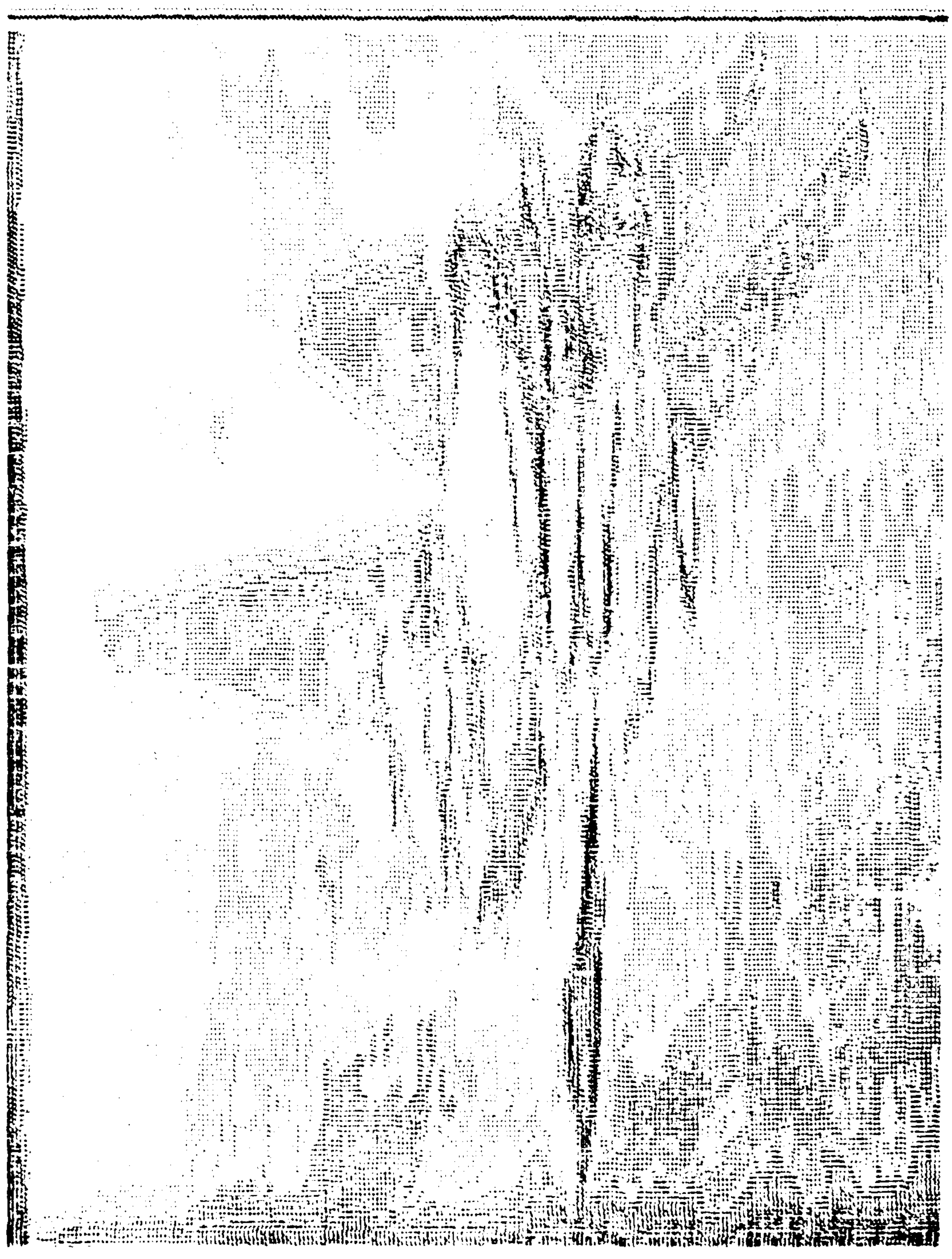

Fig. 90. F14 Tomcat Processed by a 1x11 Low Pass Filter

Fig. 91. Negative Image of Fig. 90

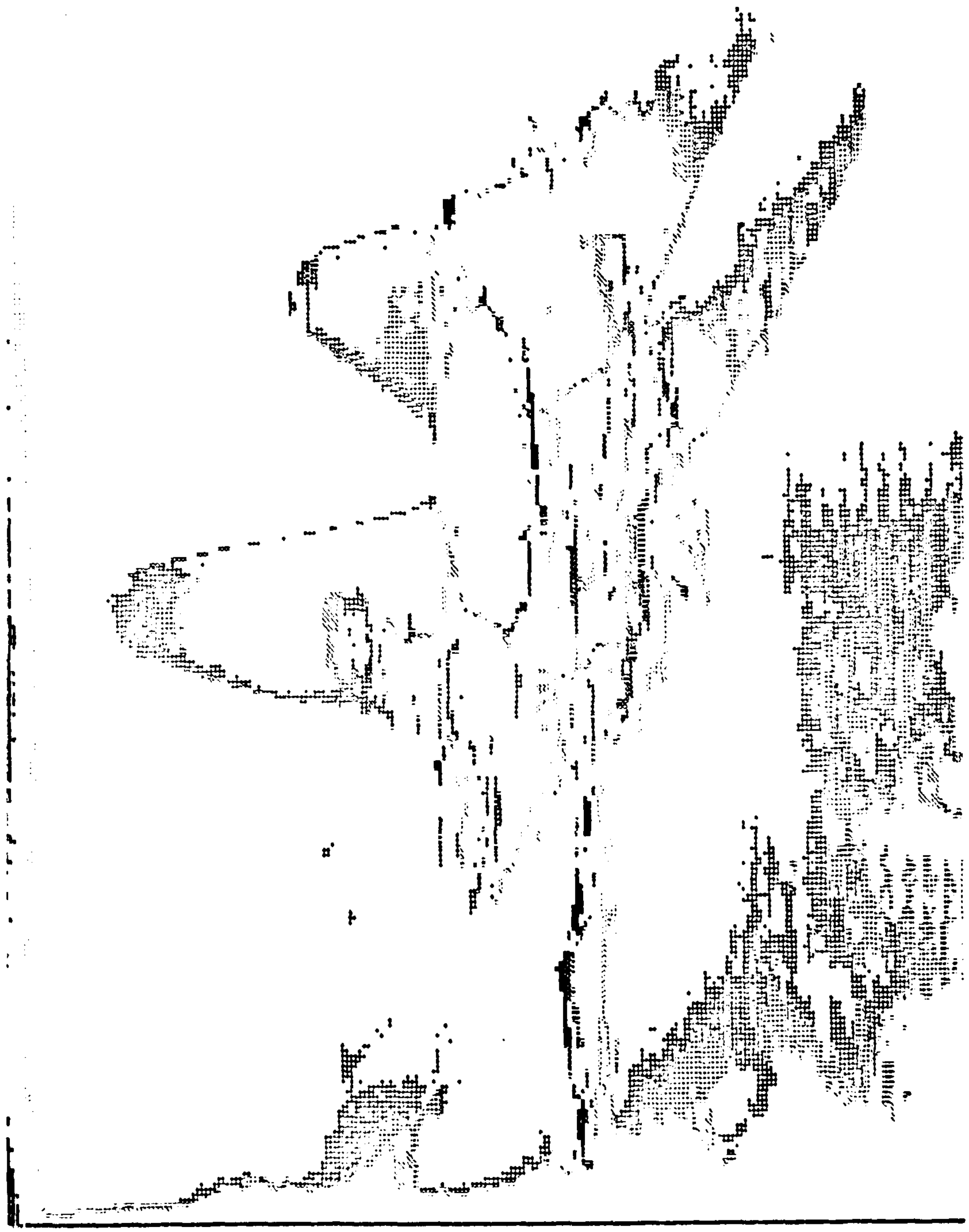

Fig. 92. Output Image of Boolpass Operation 1
Threshold = (1/2 Std Dev)/10

Fig. 93. Output Image of Boolpass Operation 1
Threshold = (Std Dev)/ 10

BOOLPASS OPERATION I

1.	Digitized Image Field2	:	Fig. 94
2.	1x11 low pass filter	:	Fig. 95
3.	Negative of 2.	:	Fig. 96
4.	AND (1/2 Std Dev)/10	:	Fig. 97
5.	AND (Std Dev)/10	:	Fig. 98

Mean: 9   Standard Deviation: 5

Fig. 94. Digitized Image of Original Picture Field2

Fig. 95. Field2 Processed by a 1x11 Low Pass Filter

Fig. 96. Negative Image of Fig. 95

Fig. 97. Output Image of Boolpass Operation 1
Threshold = (1/2 Std Dev)/10

Fig. 98. Output Image of Boolpass Operation 1
Threshold = (Std Dev)/ 10

BOOLPASS OPERATION I

1.	Digitized Image Tank	:	Fig. 99
2.	1x11 low pass filter	:	Fig. 100
3.	Negative of 2.	:	Fig. 101
4.	AND (1/2 Std Dev)/10	:	Fig. 102
5.	AND (Std Dev)/10	:	Fig. 103

Mean: 10 Standard Deviation: 5

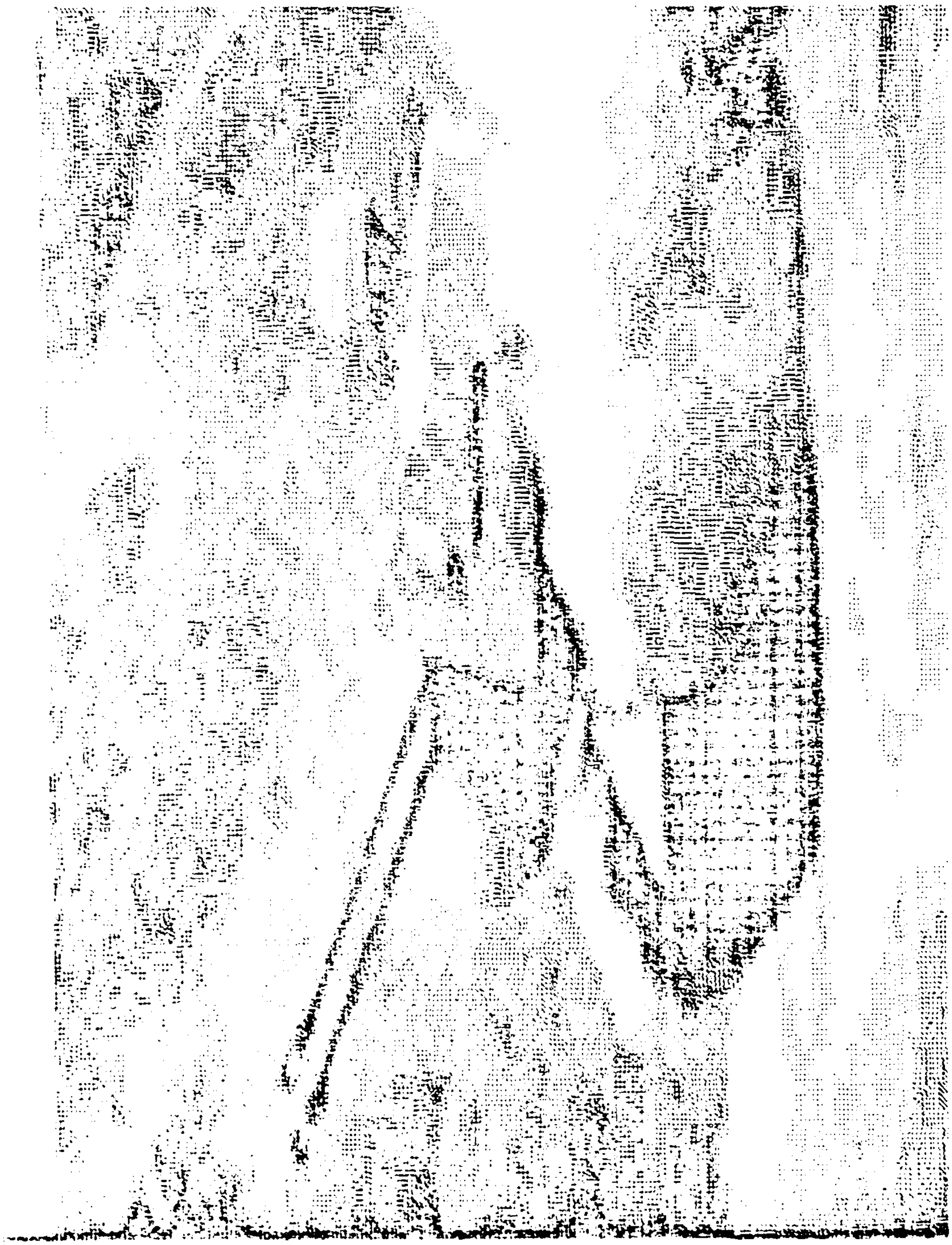

Fig. 99. Digitized Image of Original Picture Tank

Fig. 100. Tank Processed by a 1x11 Low Pass Filter

Fig. 1Ø1. Negative Image of Fig. 1ØØ

Fig. 102. Output Image of Boolpass Operation 1
Threshold = (1/2 Std Dev)/10

Fig. 103. Output Image of Boolpass Operation 1
Threshold = (Std Dev)/ 10

Outline for Boolpass Operator 2

1. Digitized picture processed : Fig. A

   Step 1

   Apply 2-dimensional 7x7 low pass filter to original pictur·, Fig. A
2. Resulting output image : Fig. B

   Step 2

   Apply 2-dimensional 3x3 low pass filter to original picture, Fig. B
3. Resulting output image : Fig. C

   Step 3

   Perform 'AND' operation using Fig. B as input 'ONE' to the AND operator and Fig. C as input 'TWO' to the AND operator Threshold = Ø, ie. there is no threshold window
4. Resulting output image : Fig. D

   Form negative image of Fig. D (ouput image from step 3)
5. Resulting ouput image : Fig. E

   Step 4

   Perform 'AND' operation using Fig. E as input 'ONE' to the AND operator and Fig. A as input 'TWO' to the AND operator

   Case 1. Threshold = (the Standard Deviation of the pixel values for the entire picture, Fig. A) / 1Ø
6. Resulting ouput image : Fig. F

   Case 2. Threshold = Ø, ie. no threshold window
7. Resulting ouput image : Fig. E

   NOTE : Pixels stored in output image of AND operation are taken from input 'TWO'

8. Other images : as specified

This same operation is repeated exactly for a 1-Dimensional low pass filter. The only change in the processing procedure occurs in Step 1 and Step 2 as follows:

Step 1

Apply a 1-Dimensional 1x7 low pass filter to original picture, Fig. A

Step 2

Apply a 1-dimensional 1x3 low pass filter to original picture, Fig. A

An abbreviated form of this outline, listing only the numbers 1 thru 8, and the corresponding correct Figures, is included before the results for any particular image processed.

BOOLPASS OPERATION 2

1.	Digitized image Truck	:	Fig. 1Ø4
2.	7x7 low pass filter	:	Fig. 1Ø5
3.	3x3 low pass filter	:	Fig. 1Ø6
4.	AND (Ø Std Dev)	:	Fig. 1Ø7
5.	Negative of 4.	:	Fig. 1Ø8
6.	AND (Std Dev)/1Ø	:	Fig. 1Ø9
7.	AND (Ø Std Dev)/1Ø	:	Fig. 11Ø

Mean: 1Ø   Standard Deviation: 4

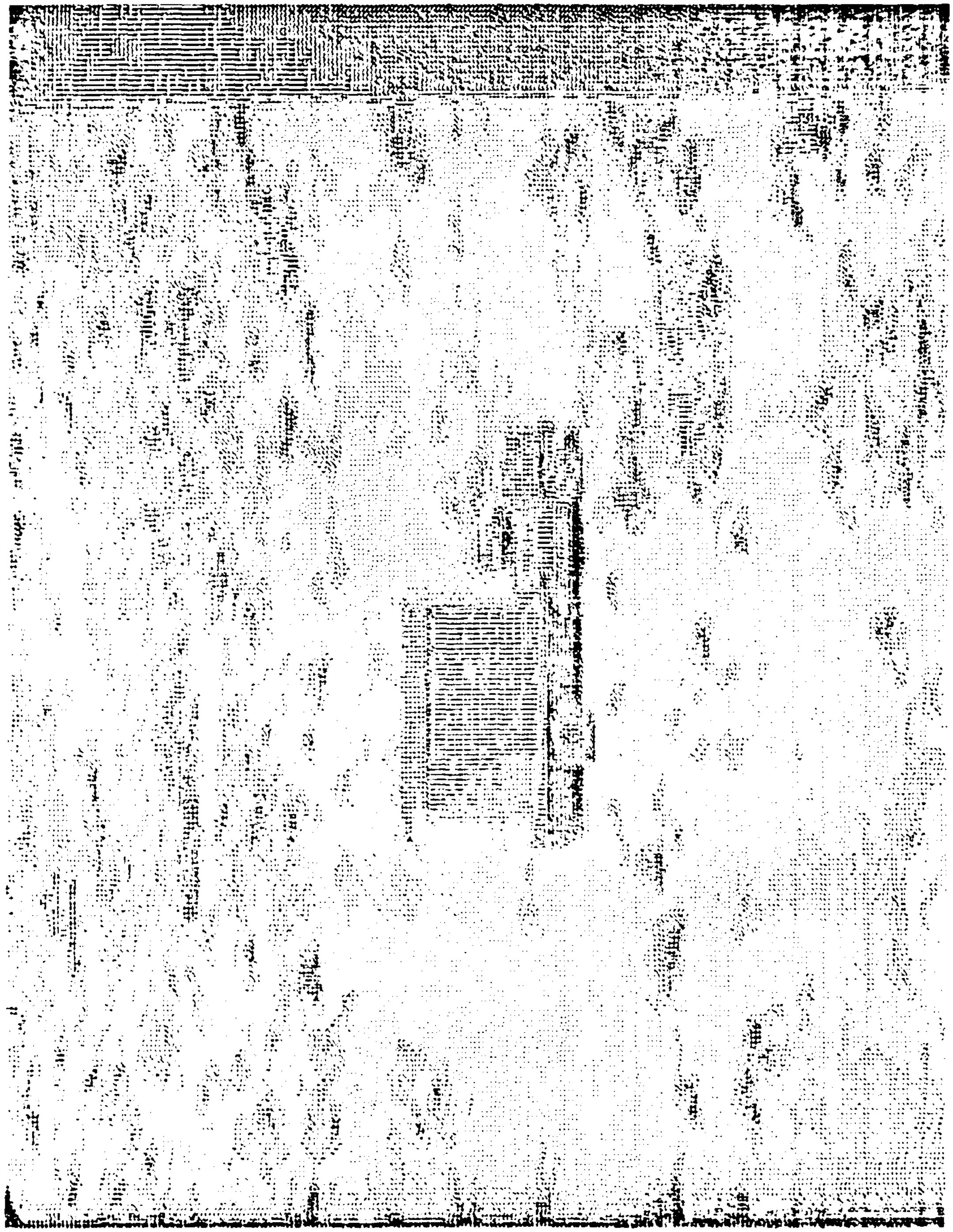

Fig. 1Ø4. Digitized video image of Truck

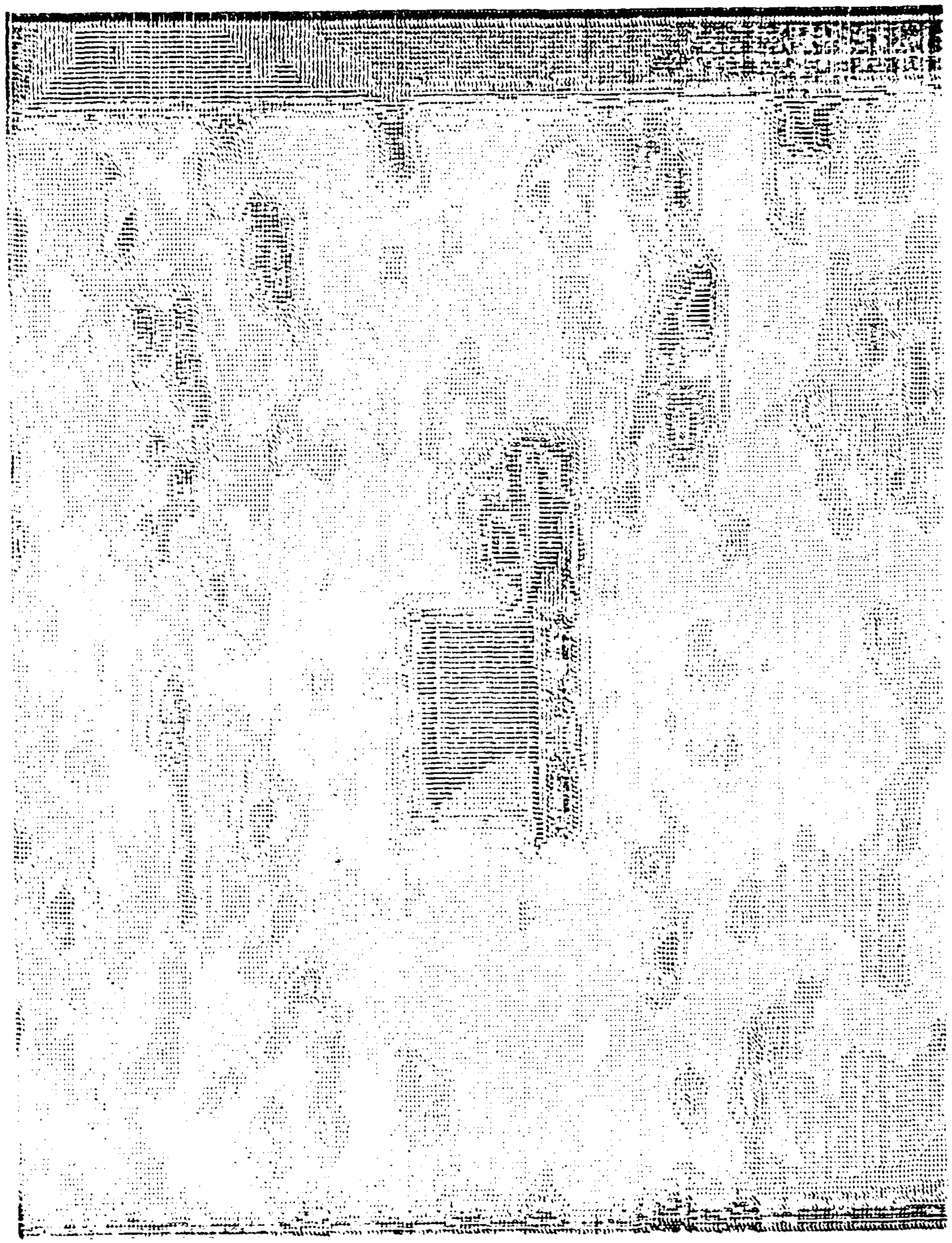

Fig. 1Ø5. Truck Processed By A 7x7 Low Pass Filter

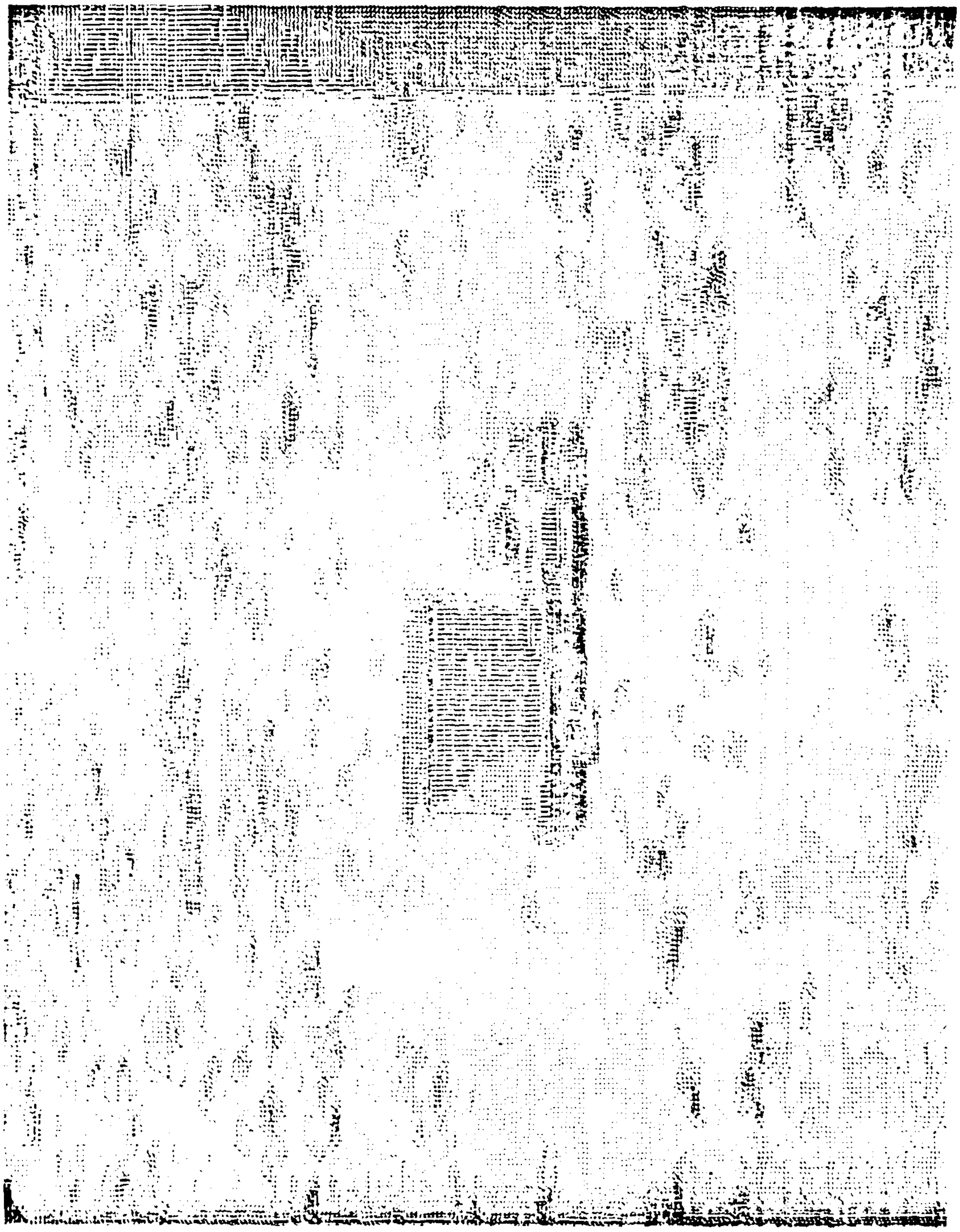

Fig. 106. Truck Processed BY A 3x3 Low Pass Filter

Fig. 107. Boolean AND Operation Performed On Prereeding Two Figures. Threshold = Ø

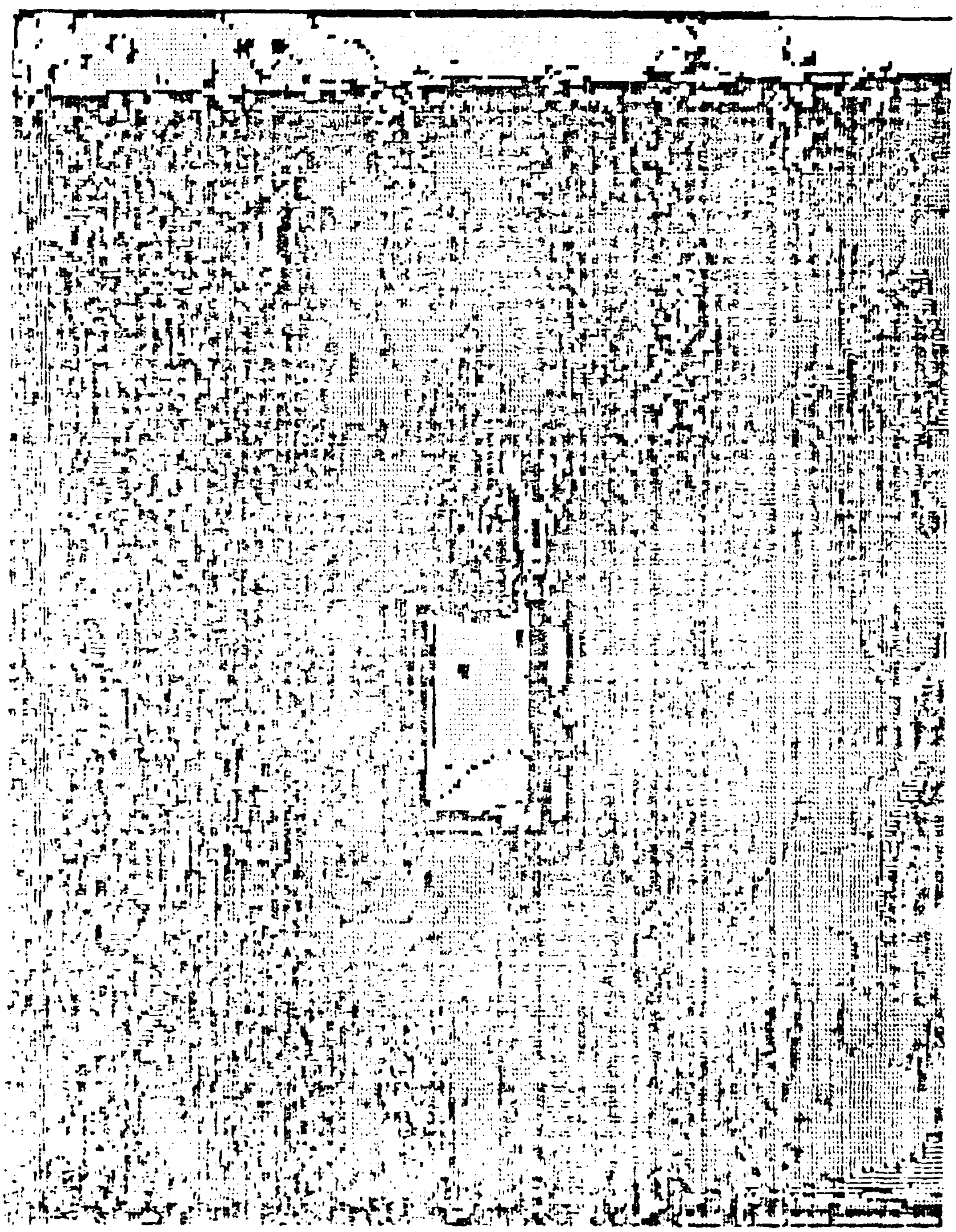

Fig. 108. Negative Image of Fig. 107

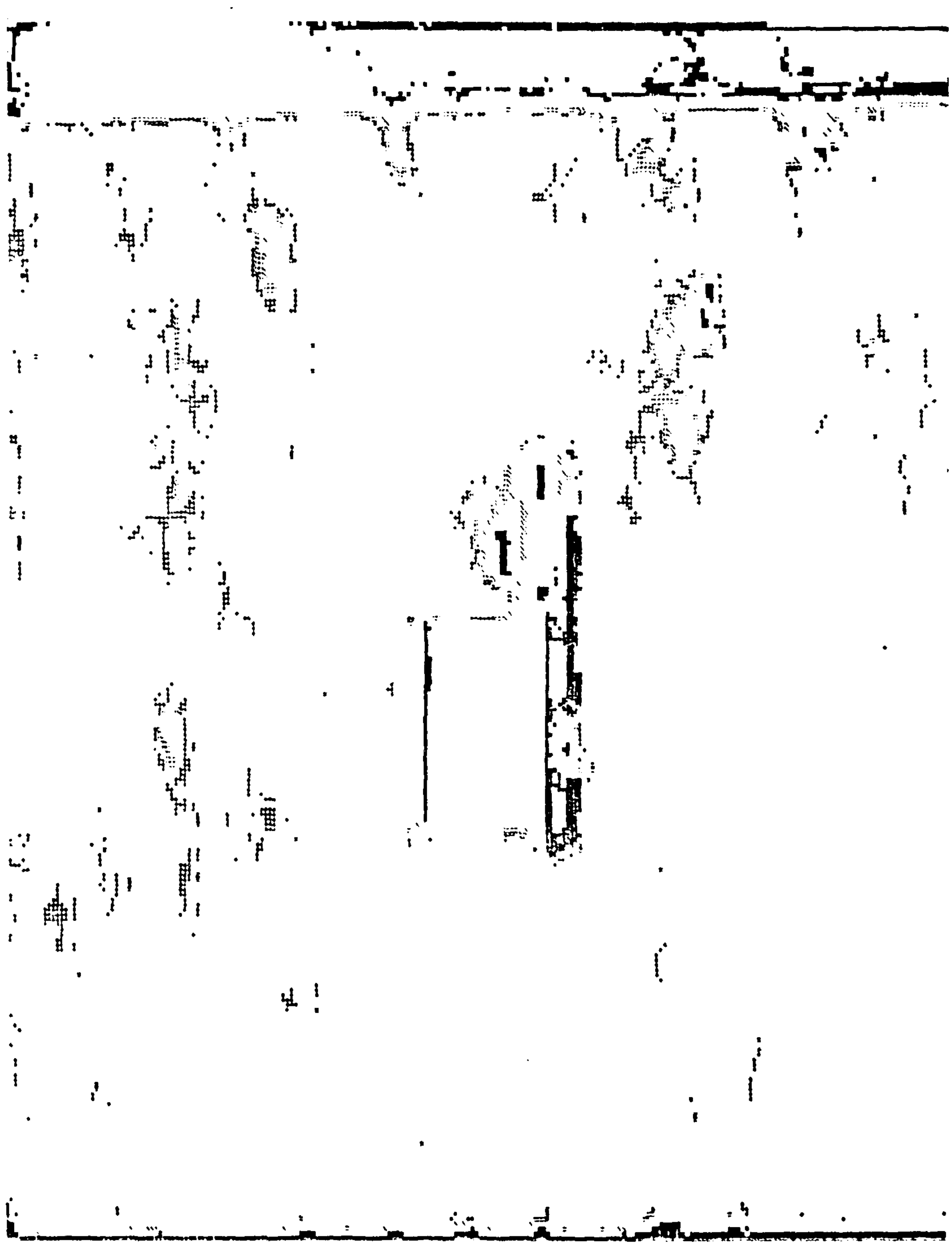

Fig. 109. Output Image Of Boolpass Operation 2
Threshold = (Std Dev)/10

Fig. 110. Output Image Of Boolpass Operation 2
Threshold = 0

BOOLPASS OPERATION 2

1.	Digitized image Tomcat	:	Fig. 111
2.	7x7 low pass filter	:	Fig. 112
3.	3x3 low pass filter	:	Fig. 113
4.	AND (Ø Std Dev)	:	Fig. 114
5.	Negative of 4.	:	Fig. 115
6.	AND (Std Dev)/1Ø	:	Fig. 116
7.	AND (Ø Std Dev)/1Ø	:	Fig. 117

Mean: 1Ø   Standard Deviation: 4

AD-A118 076 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 20/6
COMBINED SPATIAL FILTERING AND BOOLEAN OPERATORS APPLIED TO THE--ETC(U)
JUN 82 B E FELTMATE
UNCLASSIFIED AFIT/GCS/EE/82J-8 NL

3 OF 4
AD A118076

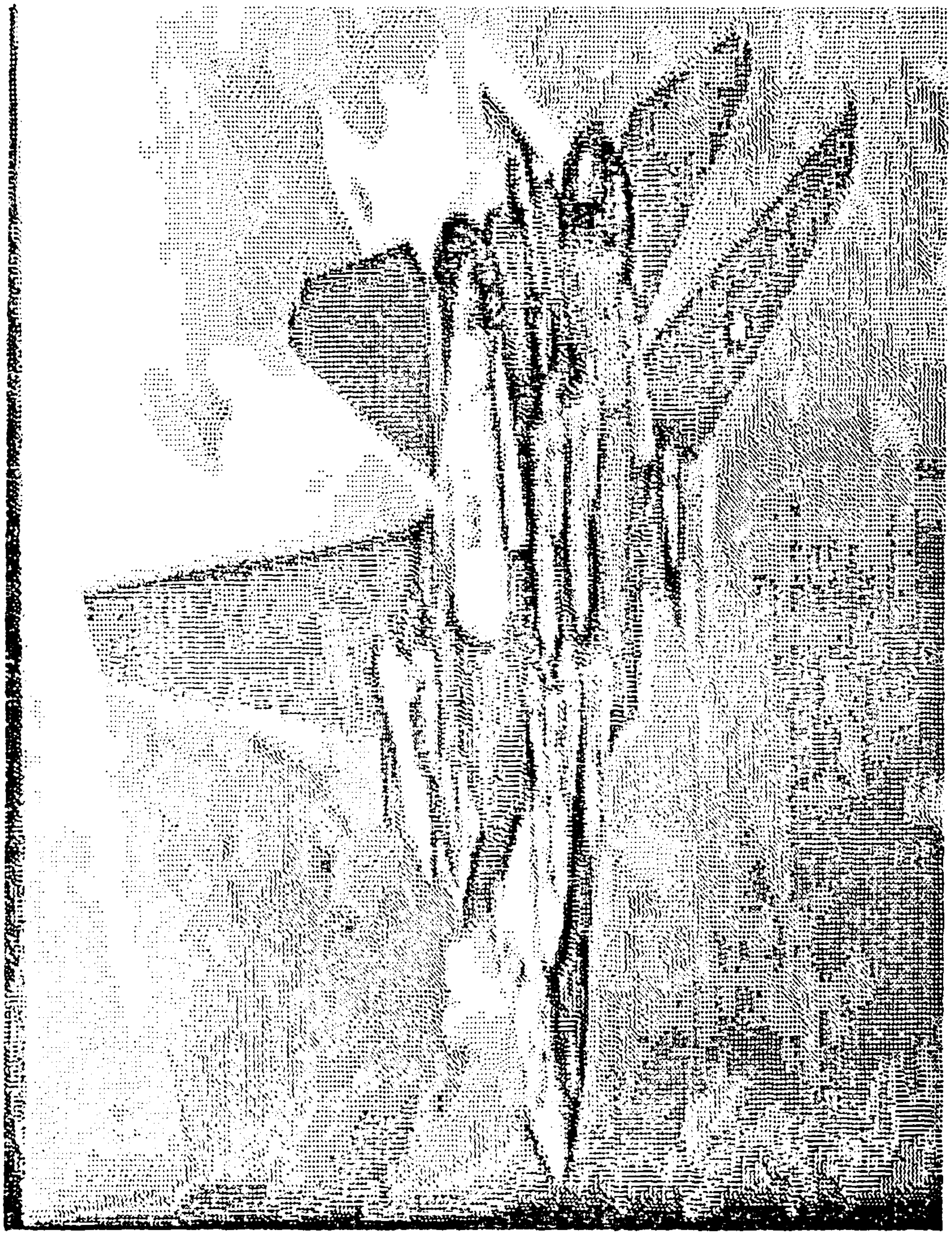

Fig. 111. Digitized video image of Tomcat

Fig. 112. Tomcat Processed By A 7x7 Low Pass Filter

Fig. 113. Tomcat Processed BY A 3x3 Low Pass Filter

Fig. 114. Boolean AND Operation Performed On Preceeding Two Figures. Threshold = Ø

Fig. 115. Negative Image of Fig. 114

Fig. 116. Output Image Of Boolpass Operation 2
Threshold = (Std Dev)/10

Fig. 117. Output Image Of Boolpass Operation 2
Threshold = Ø

BOOLPASS OPERATION 2

1.	Digitized image Field	:	Fig. 118
2.	7x7 low pass filter	:	Fig. 119
3.	3x3 low pass filter	:	Fig. 120
4.	AND (0 Std Dev)	:	Fig. 121
5.	Negative of 4.	:	Fig. 122
6.	AND (Std Dev)/10	:	Fig. 123
7.	AND (0 Std Dev)/10	:	Fig. 124

Mean: 8 Standard Deviation: 5

Fig. 118. Digitized video image of Field

Fig. 119. Field Processed By A 7x7 Low Pass Filter

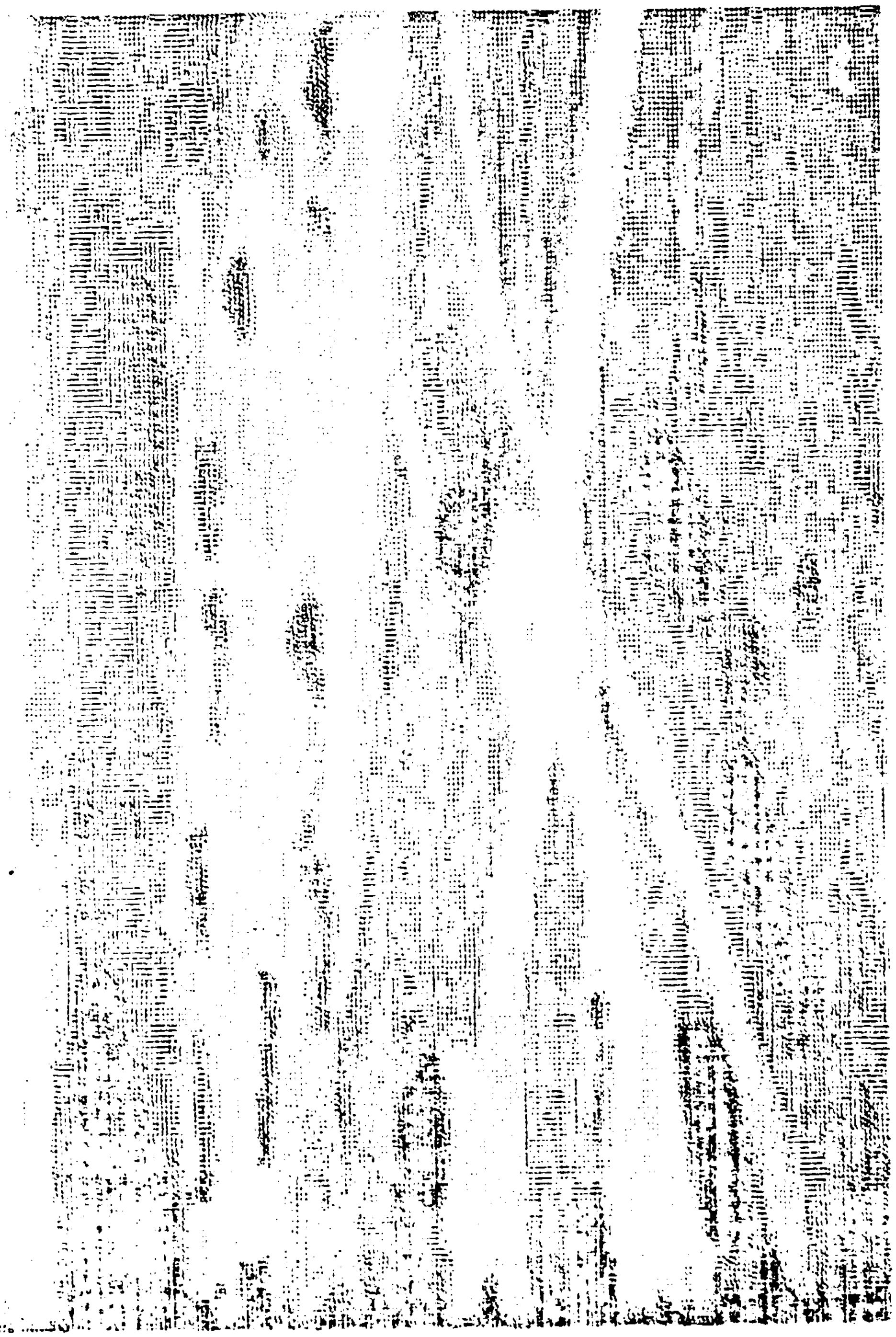

Fig. 120. Field Processed BY A 3x3 Low Pass Filter

Fig. 121. Boolean AND Operation Performed On Preceeding Two Figures. Threshold = Ø

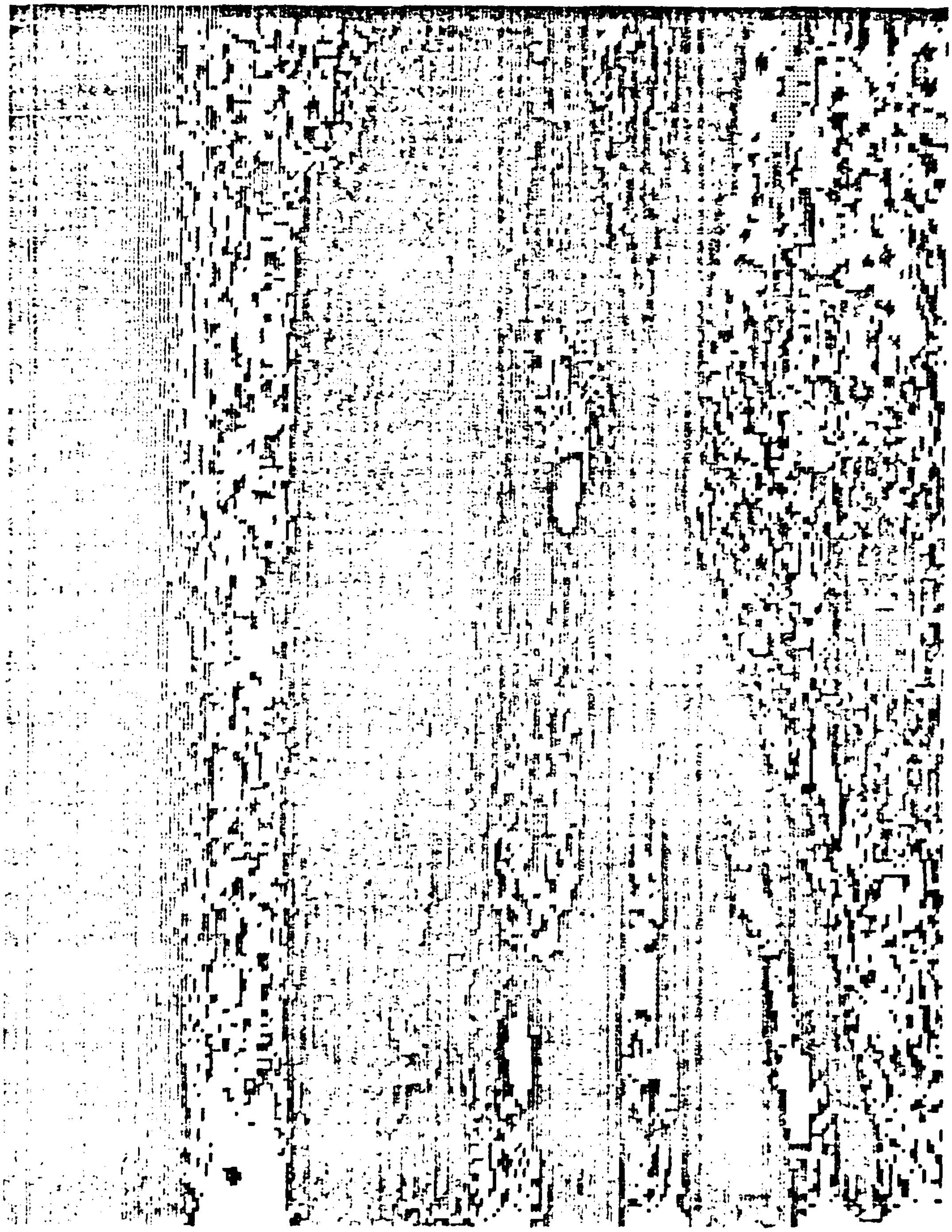

Fig. 122. Negative Image of Fig. 121

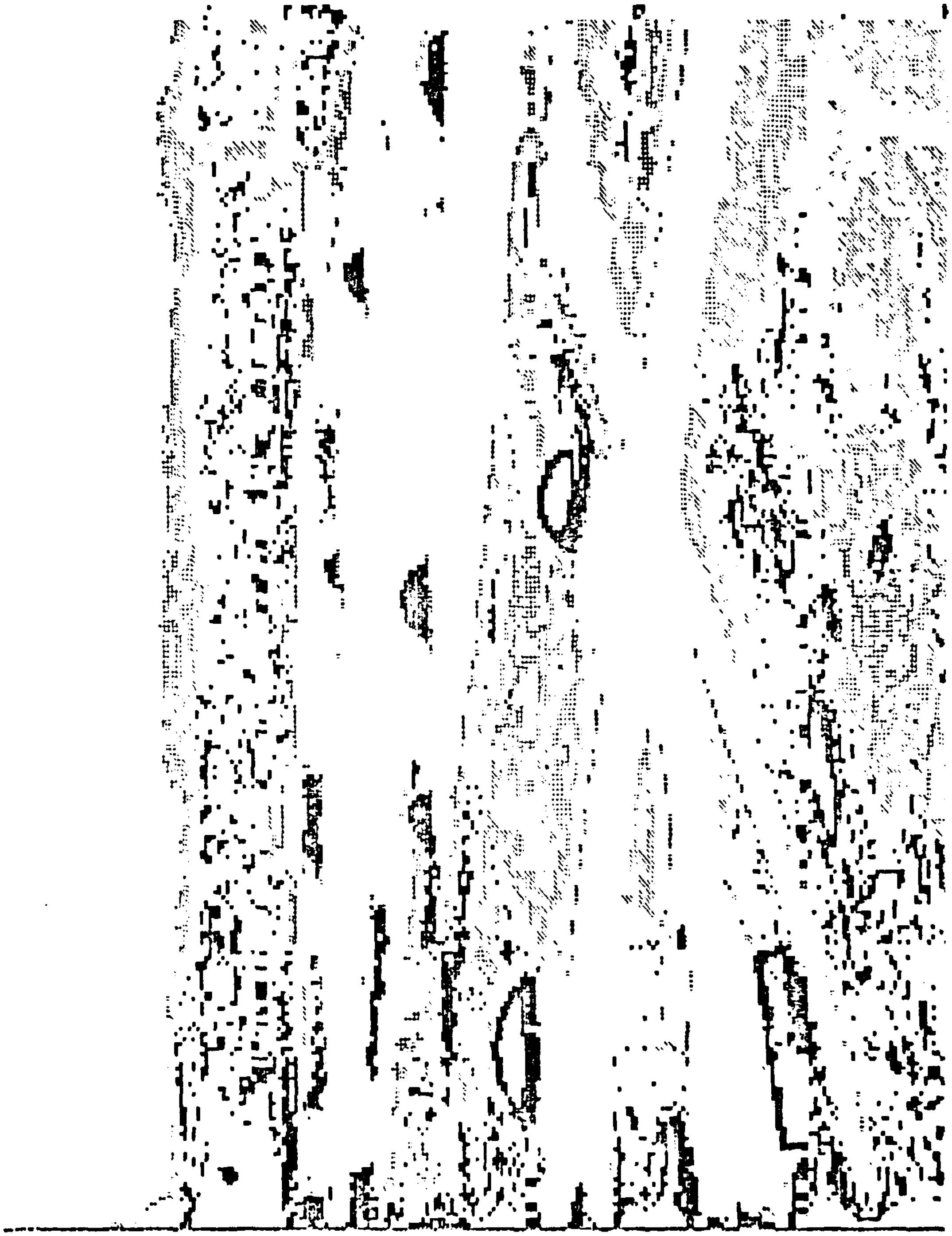

Fig. 123. Output Image Of Boolpass Operation 2
Threshold = (Std Dev)/10

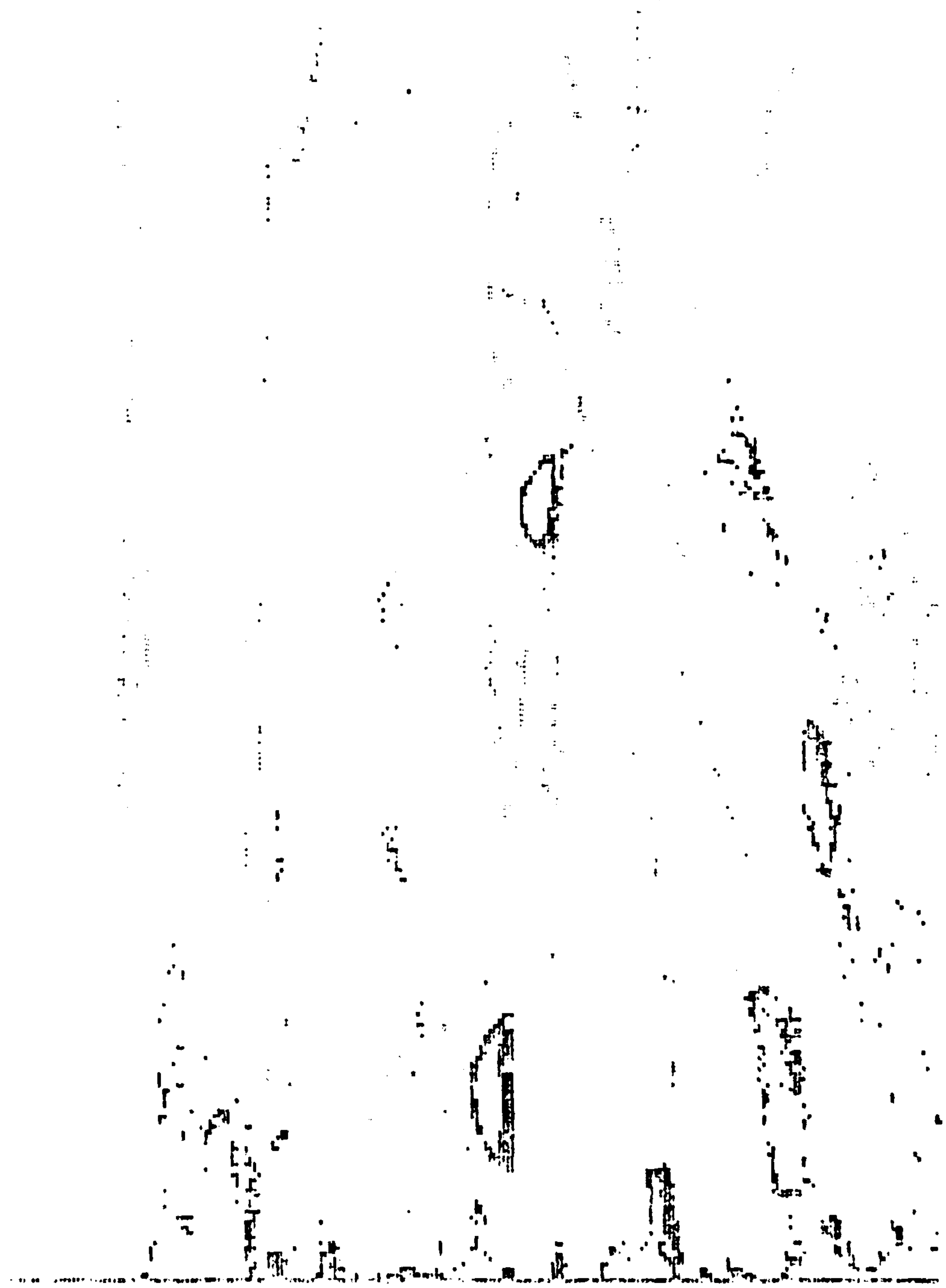

Fig. 124. Output Image Of Boolpass Operation 2
Threshold = Ø

BOOLPASS OPERATION 2

1.	Digitized image Field2	:	Fig. 125
2.	7x7 low pass filter	:	Fig. 126
3.	3x3 low pass filter	:	Fig. 127
4.	AND (0 Std Dev)	:	Fig. 128
5.	Negative of 4.	:	Fig. 129
6.	AND (Std Dev)/10	:	Fig. 130
7.	AND (0 Std Dev)/10	:	Fig. 131

Mean: 9 Standard Deviation: 5

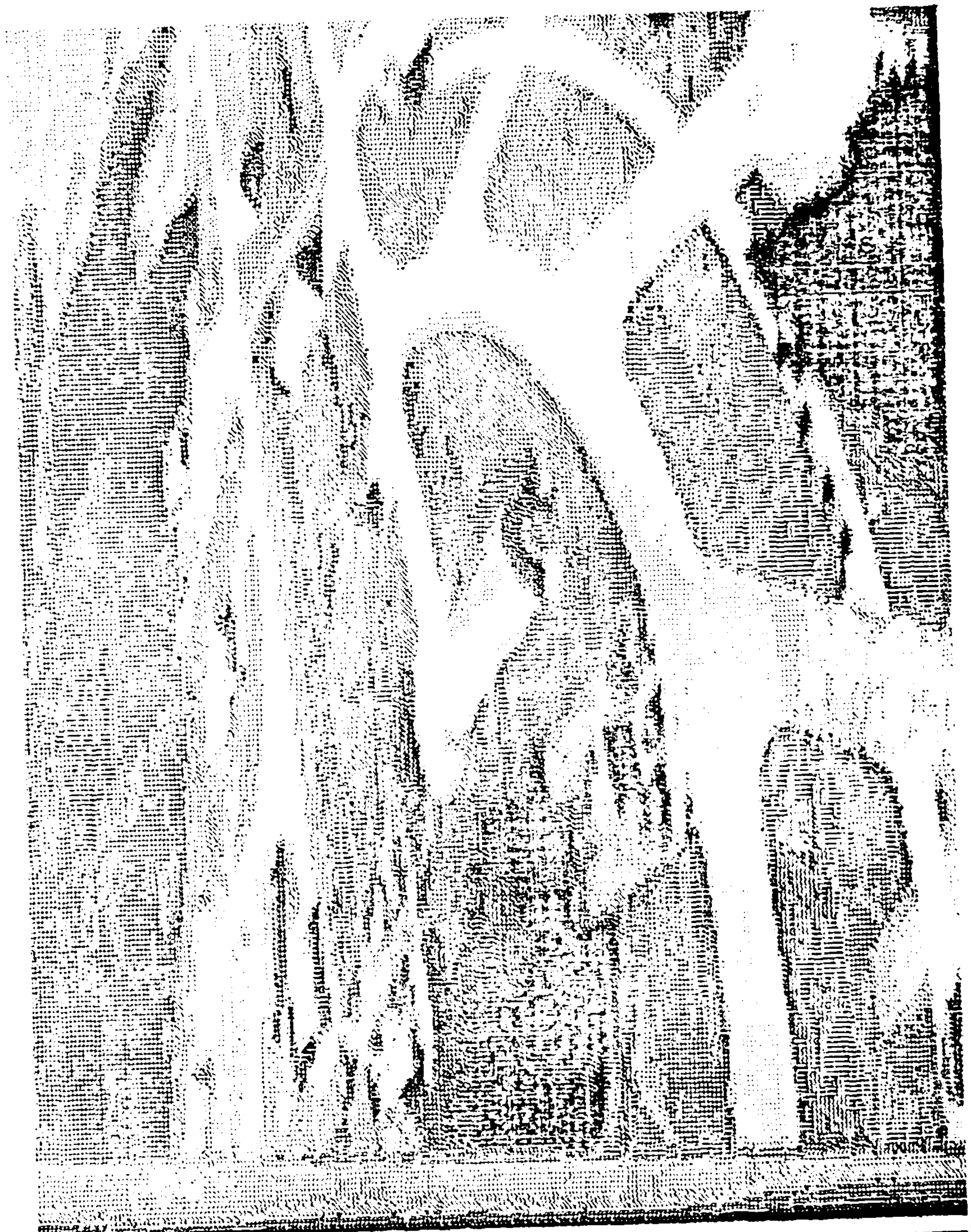

Fig. 125. Digitized video image of Field2

Fig. 126. Field2 Processed By A 7x7 Low Pass Filter

Fig. 127. Field2 Processed BY A 3x3 Low Pass Filter

Fig. 128. Boolean AND Operation Performed On Preceeding Two Figures. Threshold = Ø

Fig. 129. Negative Image of Fig. 128

Fig. 130. Output Image Of Boolpass Operation 2
Threshold = (Std Dev)/10

Fig. 131. Output Image Of Boolpass Operation 2
Threshold = Ø

BOOLPASS OPERATION 2

1.	Digitized image Hornet	:	Fig. 132
2.	7x7 low pass filter	:	Fig. 133
3.	3x3 low pass filter	:	Fig. 134
4.	AND (Ø Std Dev)	:	Fig. 135
5.	Negative of 4.	:	Fig. 136
6.	AND (Std Dev)/1Ø	:	Fig. 137
7.	AND (Ø Std Dev)/1Ø	:	Fig. 138

Mean: 9 Standard Deviation: 3

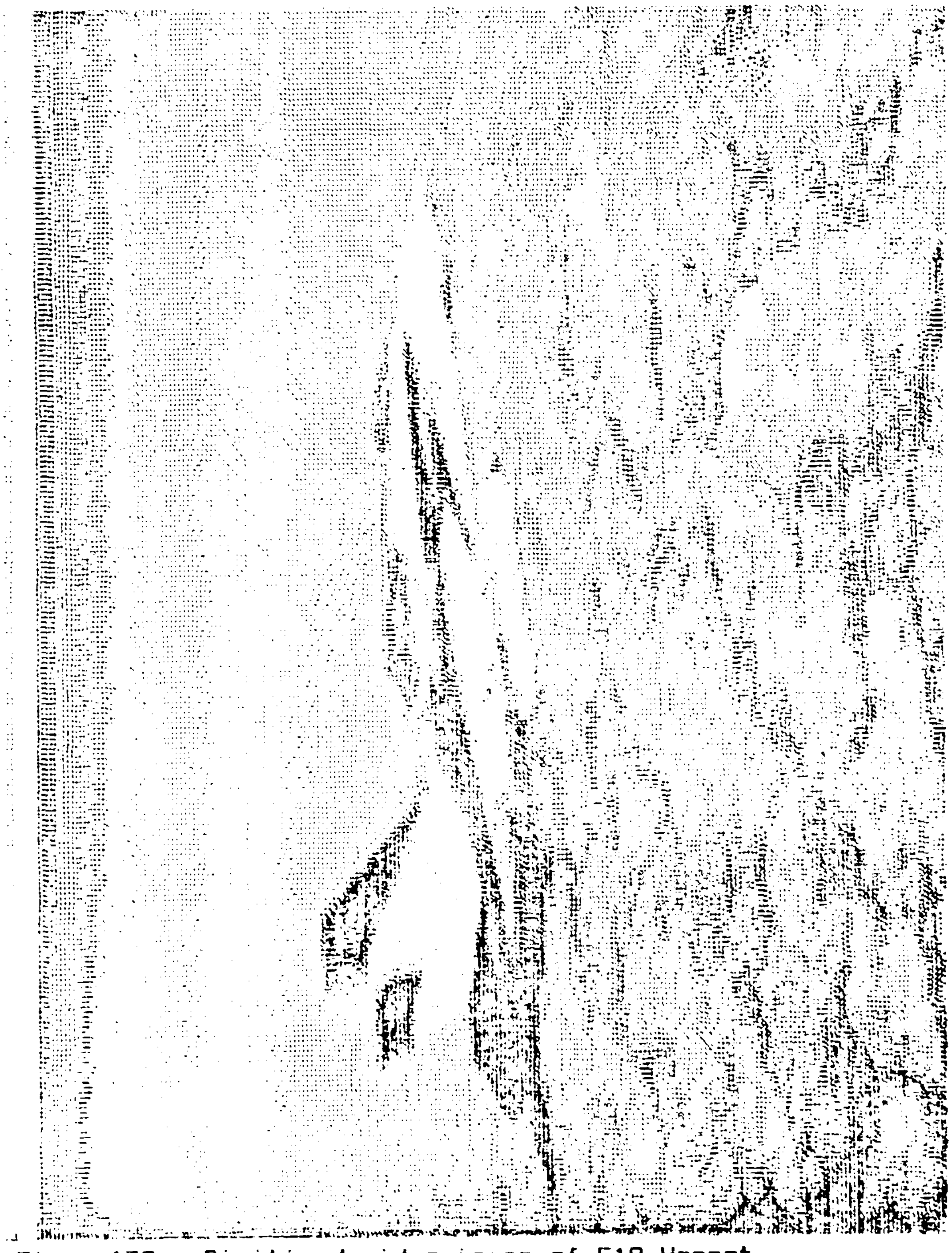

Fig. 132. Digitized video image of F18 Hornet

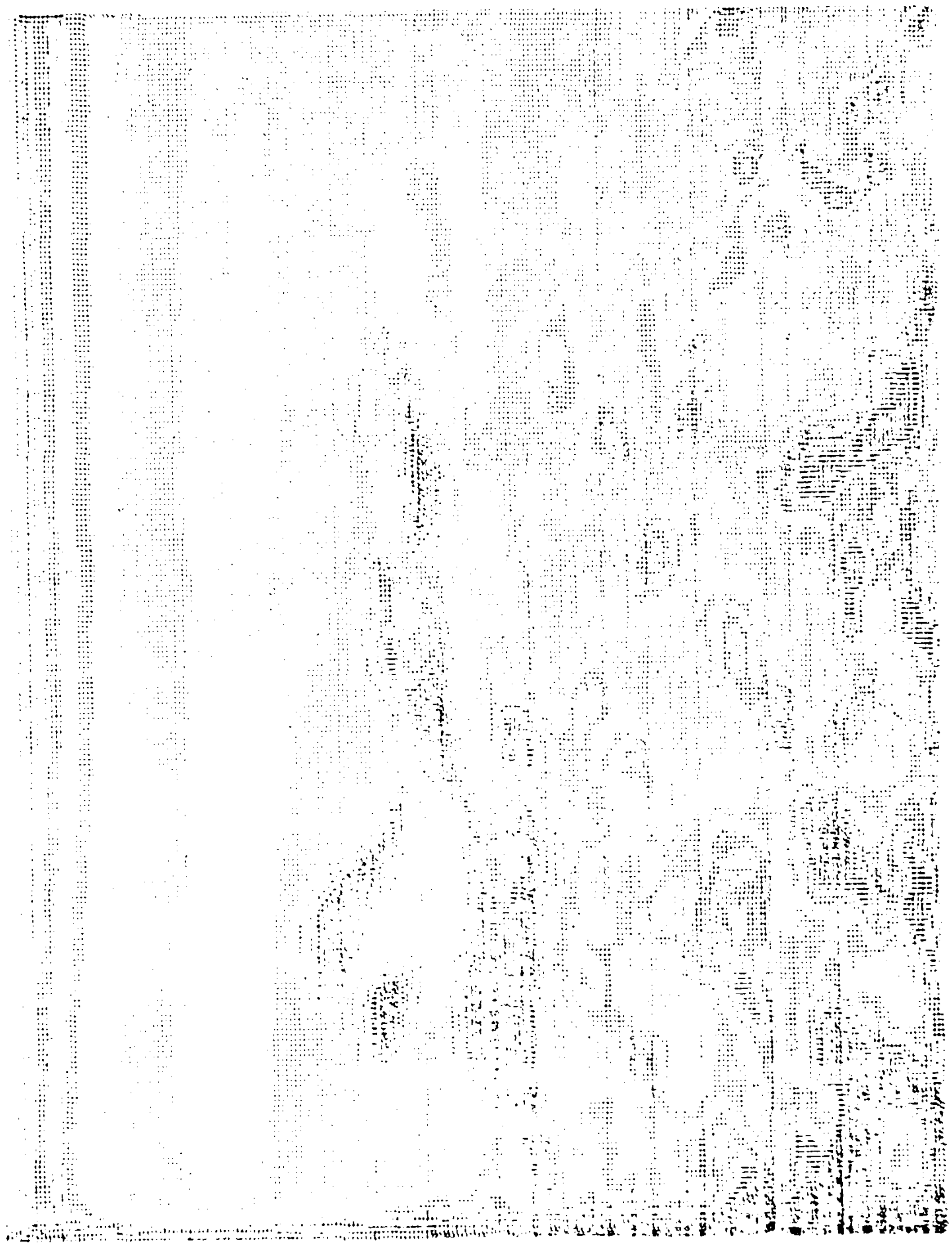

Fig. 133. F18 Hornet Processed By A 7x7 Low Pass Filter

Fig. 134. F18 Hornet Processed BY A 3x3 Low Pass Filter

Fig. 135. Boolean AND Operation Performed On Preceeding Two Figures. Threshold = Ø

Fig. 136. Negative Image of Fig. 135

Fig. 137. Output Image Of Boolpass Operation 2
Threshold = (Std Dev)/10

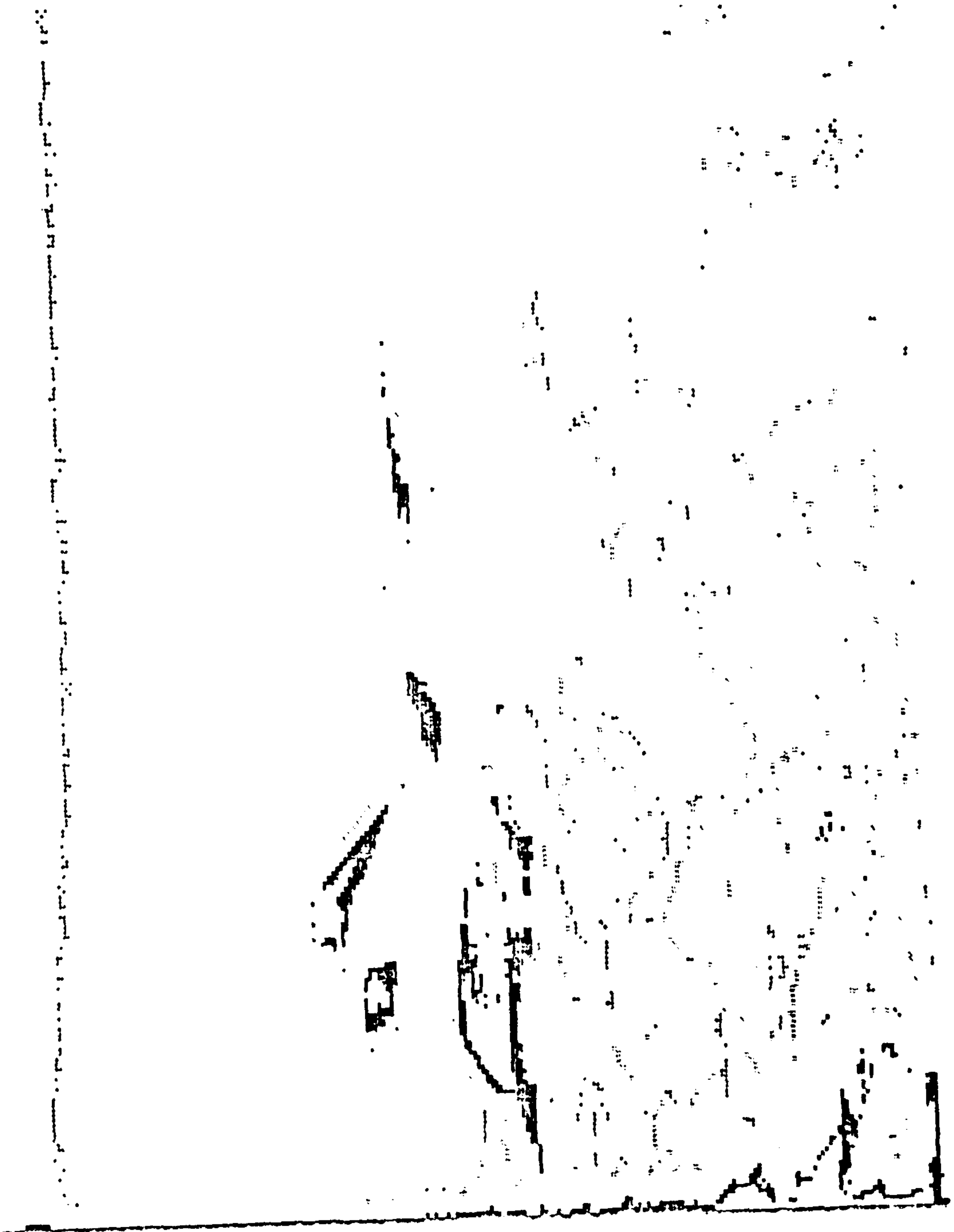

Fig. 138. Output Image Of Boolpass Operation 2
Threshold = Ø

BOOLPASS OPERATION 2

1.	Digitized image Tank	:	Fig. 139
2.	7x7 low pass filter	:	Fig. 140
3.	3x3 low pass filter	:	Fig. 141
4.	AND (0 Std Dev)	:	Fig. 142
5.	Negative of 4.	:	Fig. 143
6.	AND (Std Dev)/10	:	Fig. 144
7.	AND (0 Std Dev)/10	:	Fig. 145
8.	AND (Threshold = .4)	:	Fig. 146

Mean: 10 Standard Deviation: 5

Fig. 139. Digitized video image of Tank

Fig. 140. Tank Processed By A 7x7 Low Pass Filter

Fig. 141. Tank Processed BY A 3x3 Low Pass Filter

Fig. 142. Boolean AND Operation Performed On Preceeding Two Figures. Threshold = Ø

Fig. 143. Negative Image of Fig. 142

Fig. 144. Output Image Of Boolpass Operation 2
Threshold = (Std Dev)/10

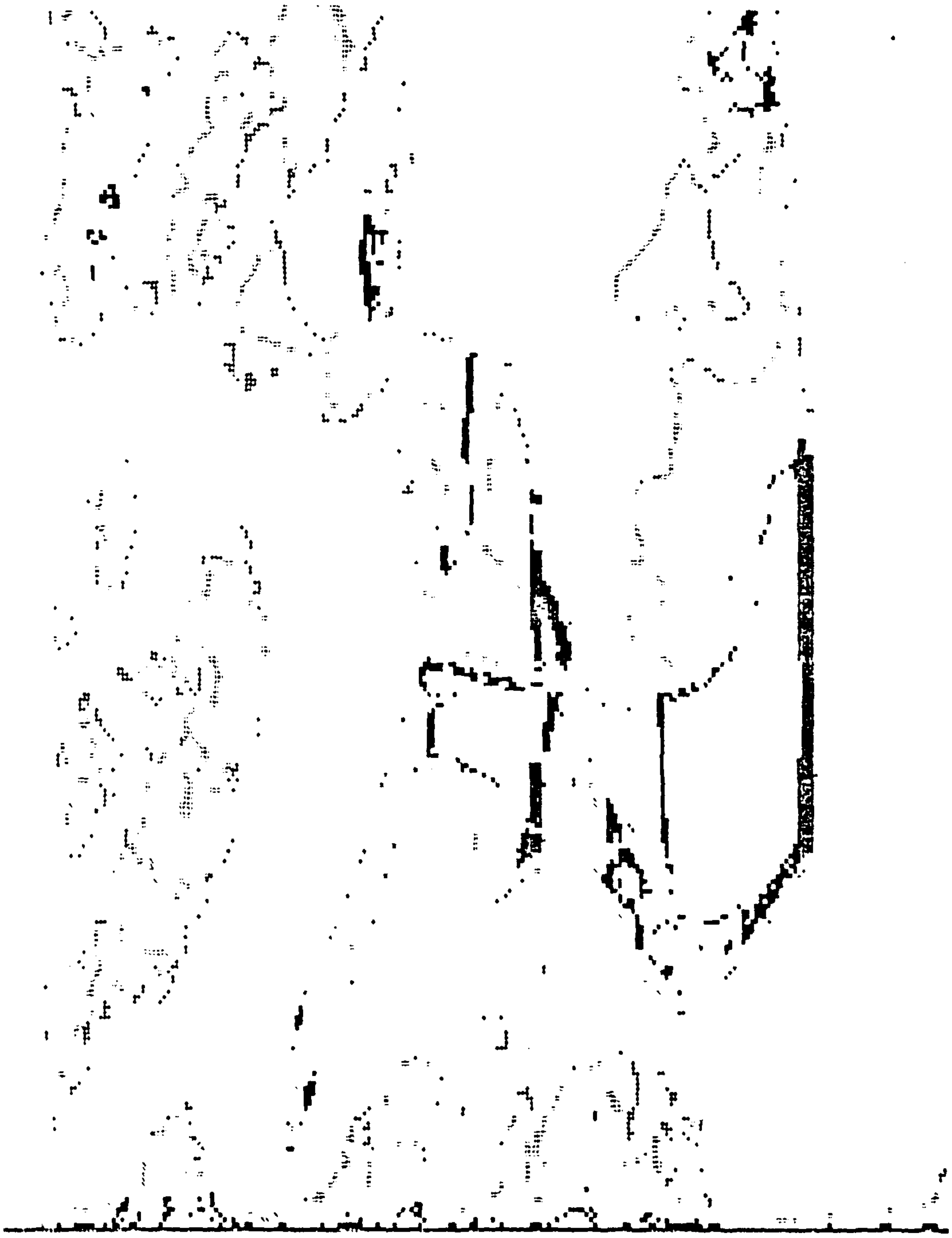

Fig. 145. Output Image Of Boolpass Operation 2
Threshold = Ø

Fig. 146. Output Image Of Boolpass Operation 2
Threshold = .4

BOOLPASS OPERATION 2

1.	Digitized image Dispersal	:	Fig. 147
2.	7x7 low pass filter	:	Fig. 148
3.	3x3 low pass filter	:	Fig. 149
4.	AND (Ø Std Dev)	:	Fig. 15Ø
5.	Negative of 4.	:	Fig. 151
6.	AND (Std Dev)/1Ø	:	Fig. 152
7.	AND (Ø Std Dev)/1Ø	:	Fig. 153

Mean: 9 Standard Deviation: 5

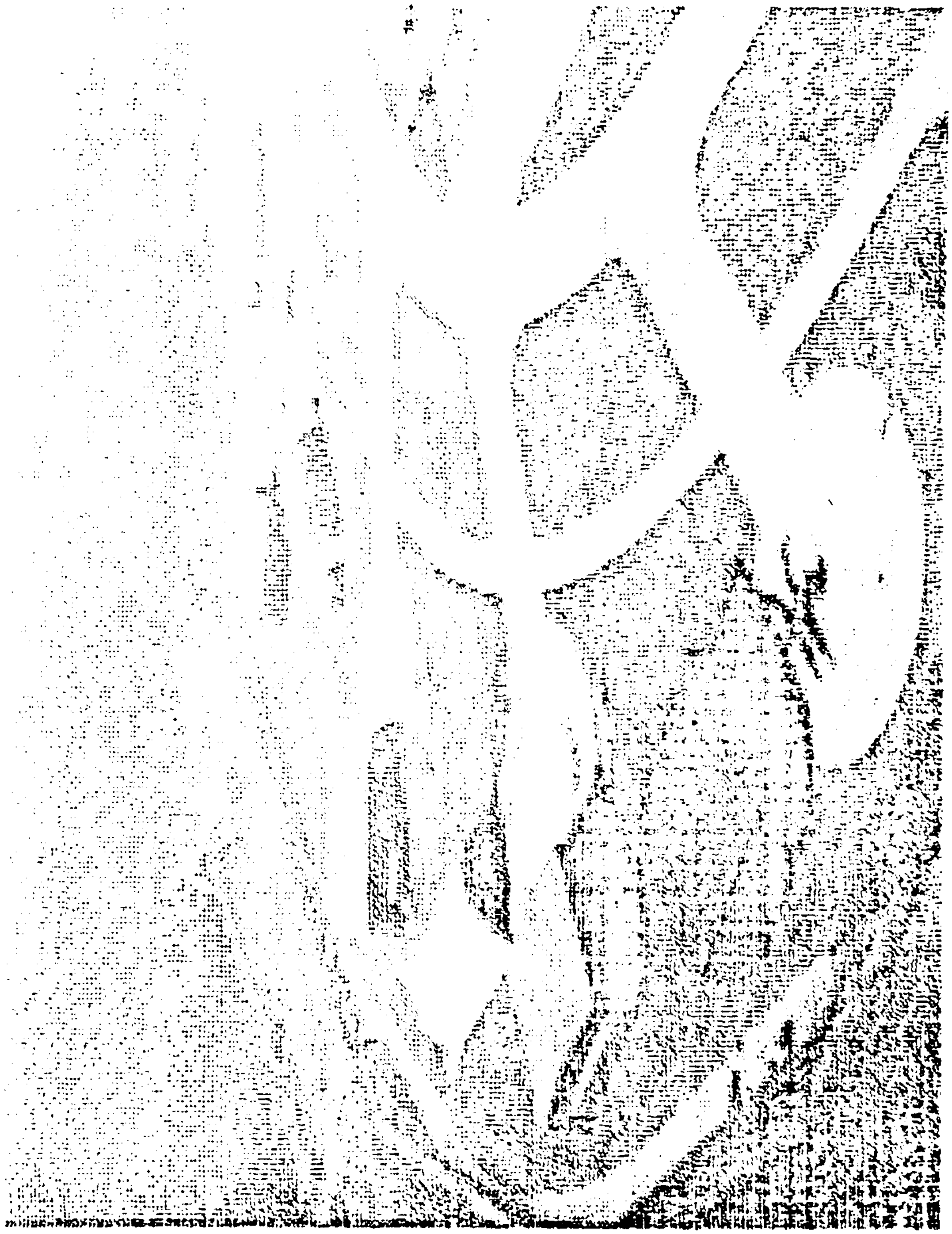

Fig. 147. Digitized video image of Dispersal

Fig. 148. Dispersal Processed By A 7x7 Low Pass Filter

Fig. 149. Tank Processed BY A 3x3 Low Pass Filter

Fig. 15Ø. Boolean AND Operation Performed On Preceeding Two Figures. Threshold = Ø

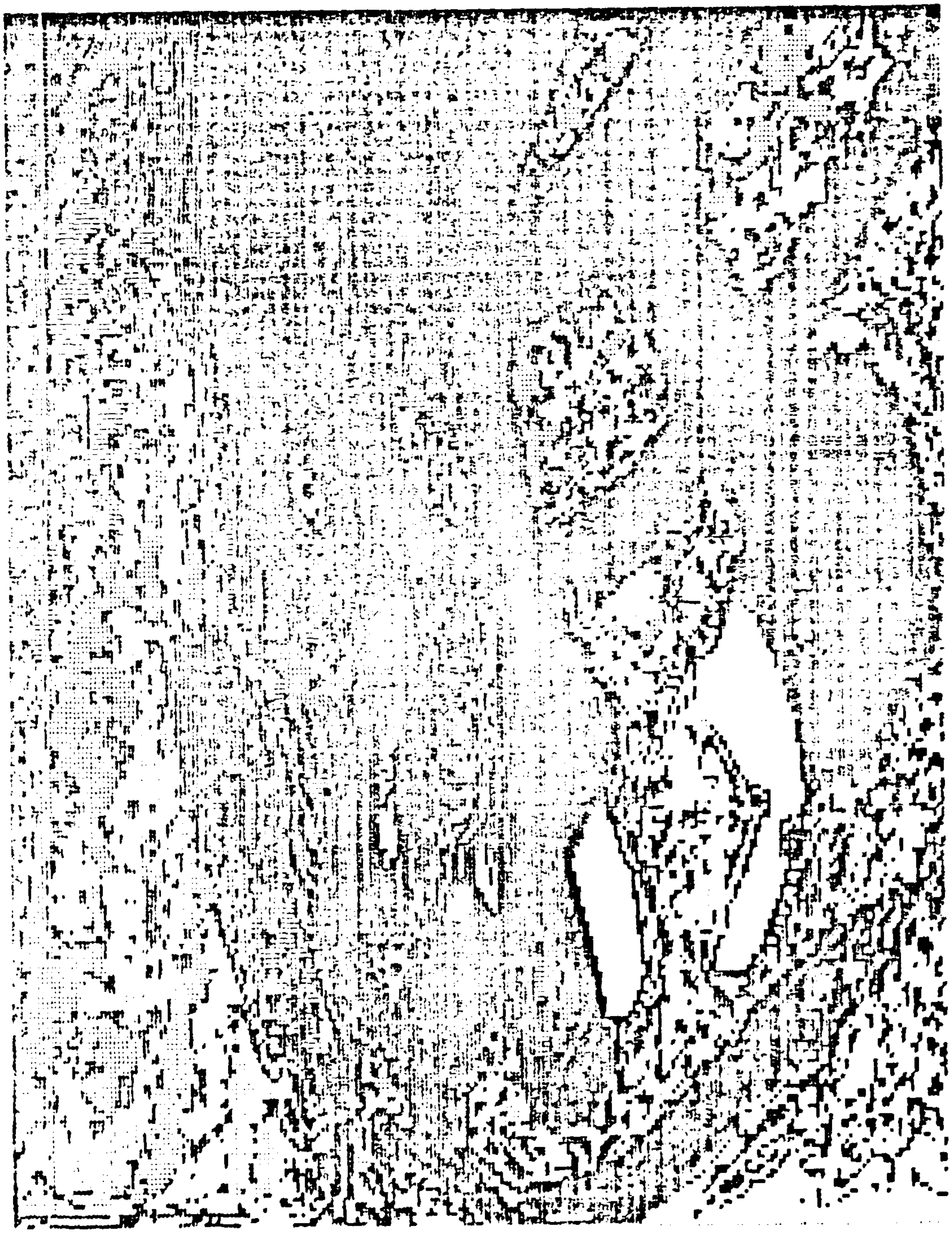

Fig. 151. Negative Image of Fig. 150

Fig. 152. Output Image Of Boolpass Operation 2
Threshold = (Std Dev)/10

Fig. 153. Output Image Of Boolpass Operation 2
Threshold = Ø

BOOLPASS OPERATION 2

1.	Digitized image Truck	:	Fig. 154
2.	1x7 low pass filter	:	Fig. 155
3.	1x3 low pass filter	:	Fig. 156
4.	AND (Ø Std Dev)	:	Fig. 157
5.	Negative of 4.	:	Fig. 158
6.	AND (Std Dev)/1Ø	:	Fig. 159
7.	AND (Ø Std Dev)/1Ø	:	Fig. 16Ø

Mean: 1Ø   Standard Deviation: 4

Fig. 154. Digitized video image of Truck

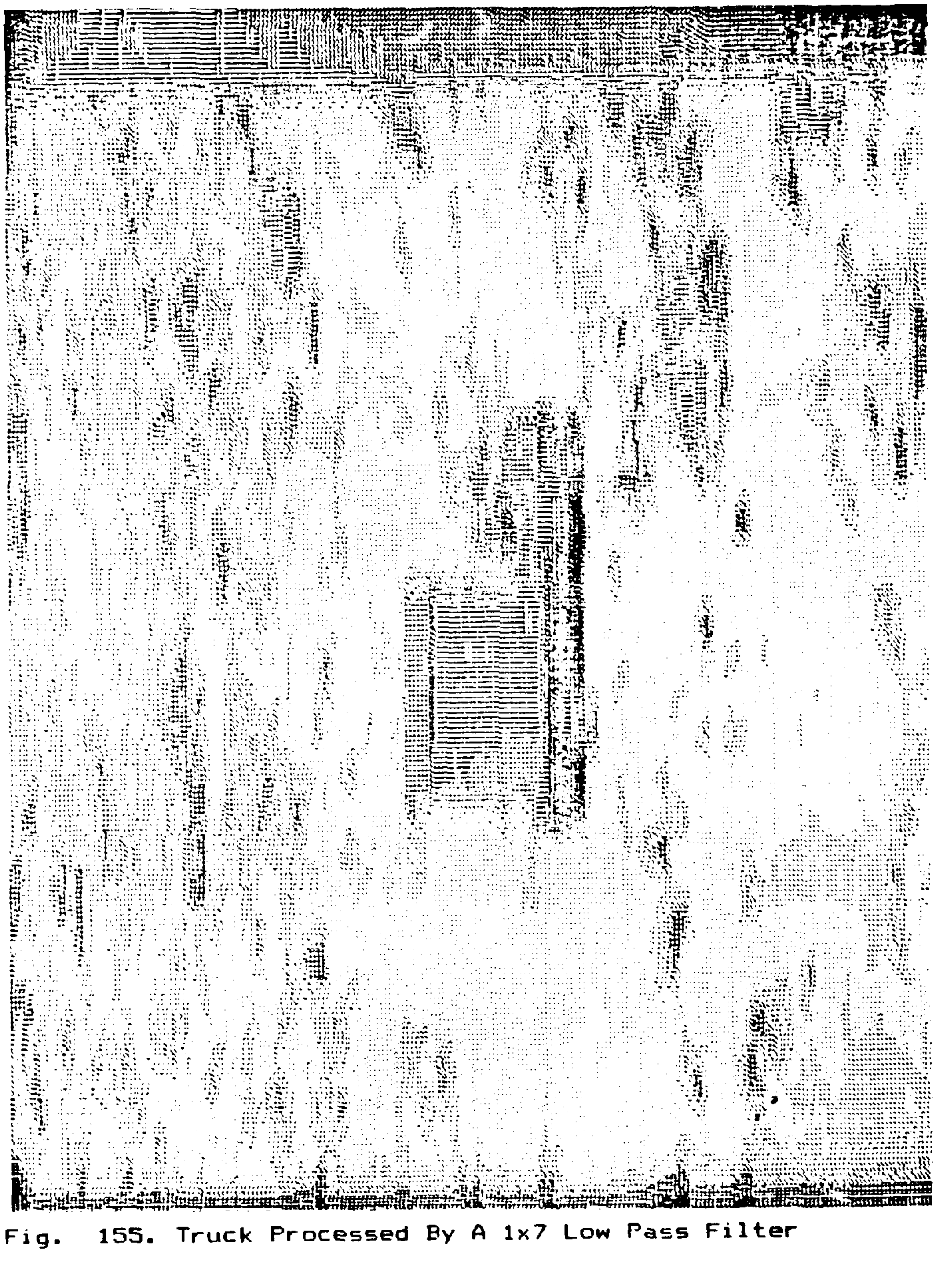

Fig. 155. Truck Processed By A 1x7 Low Pass Filter

Fig. 156. Truck Processed BY A 1x3 Low Pass Filter

Fig. 157. Boolean AND Operation Performed On Preceeding Two Figures. Threshold = Ø

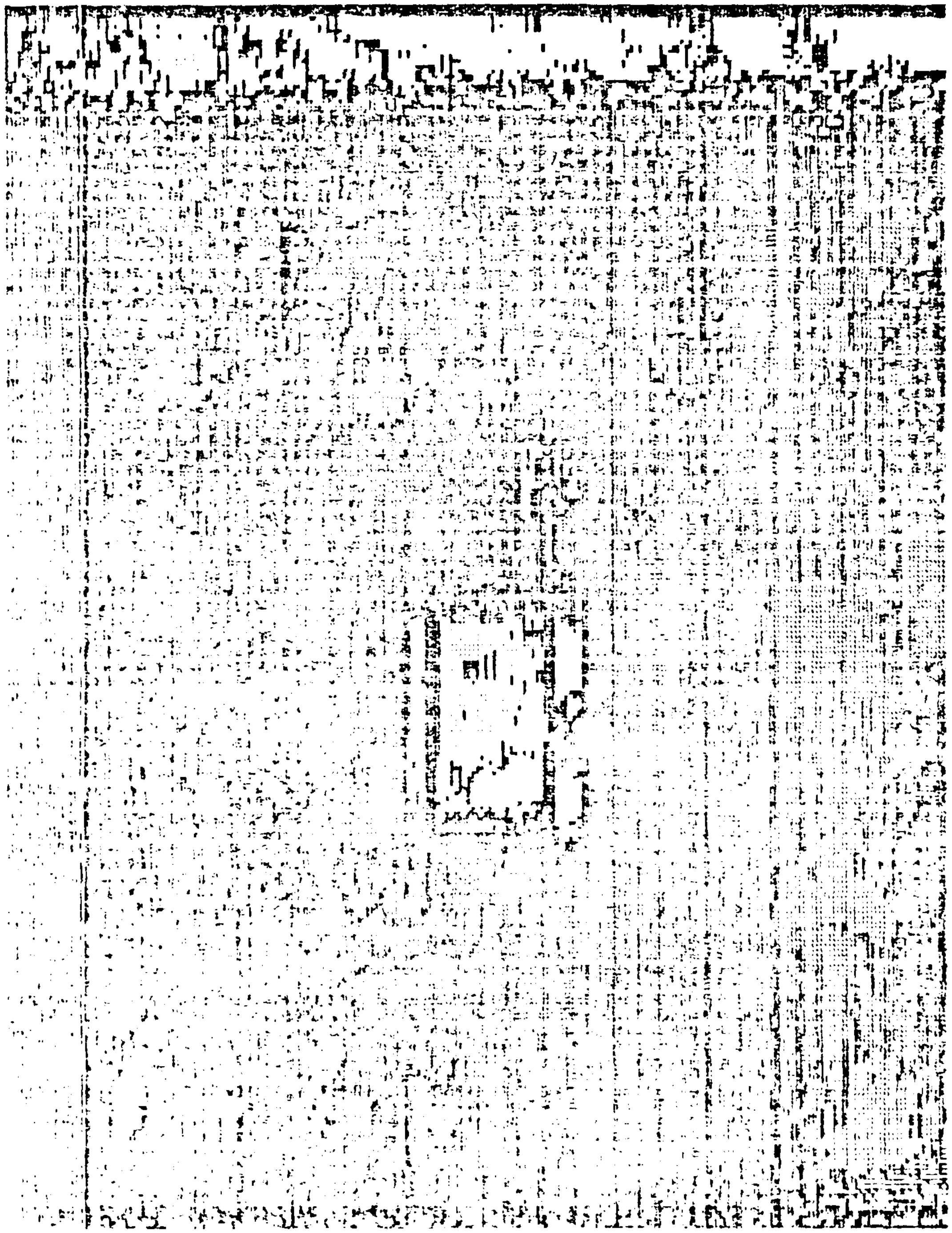

Fig. 158. Negative Image of Fig. 157

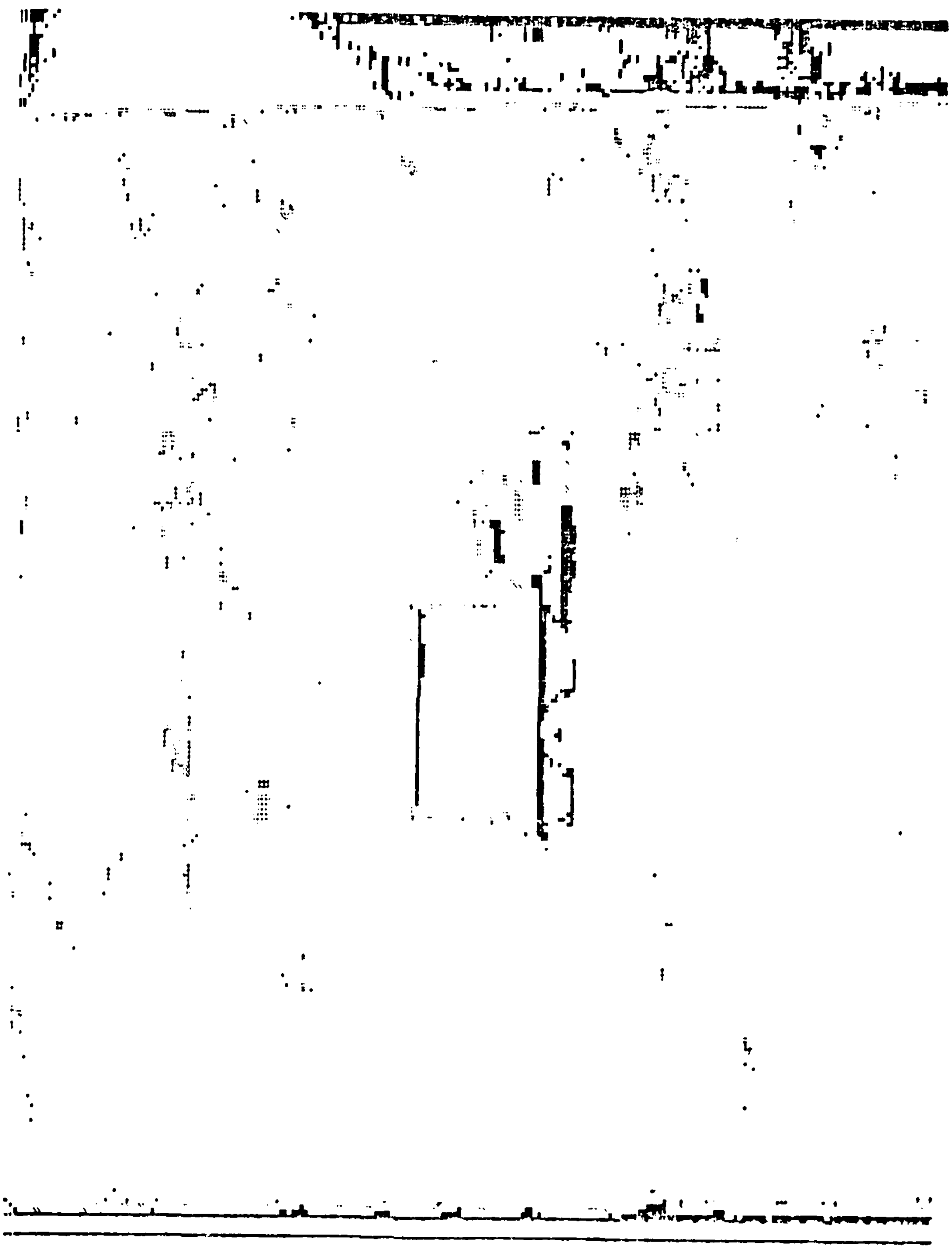

Fig. 159. Output Image Of Boolpass Operation 2
Threshold = (Std Dev)/1Ø

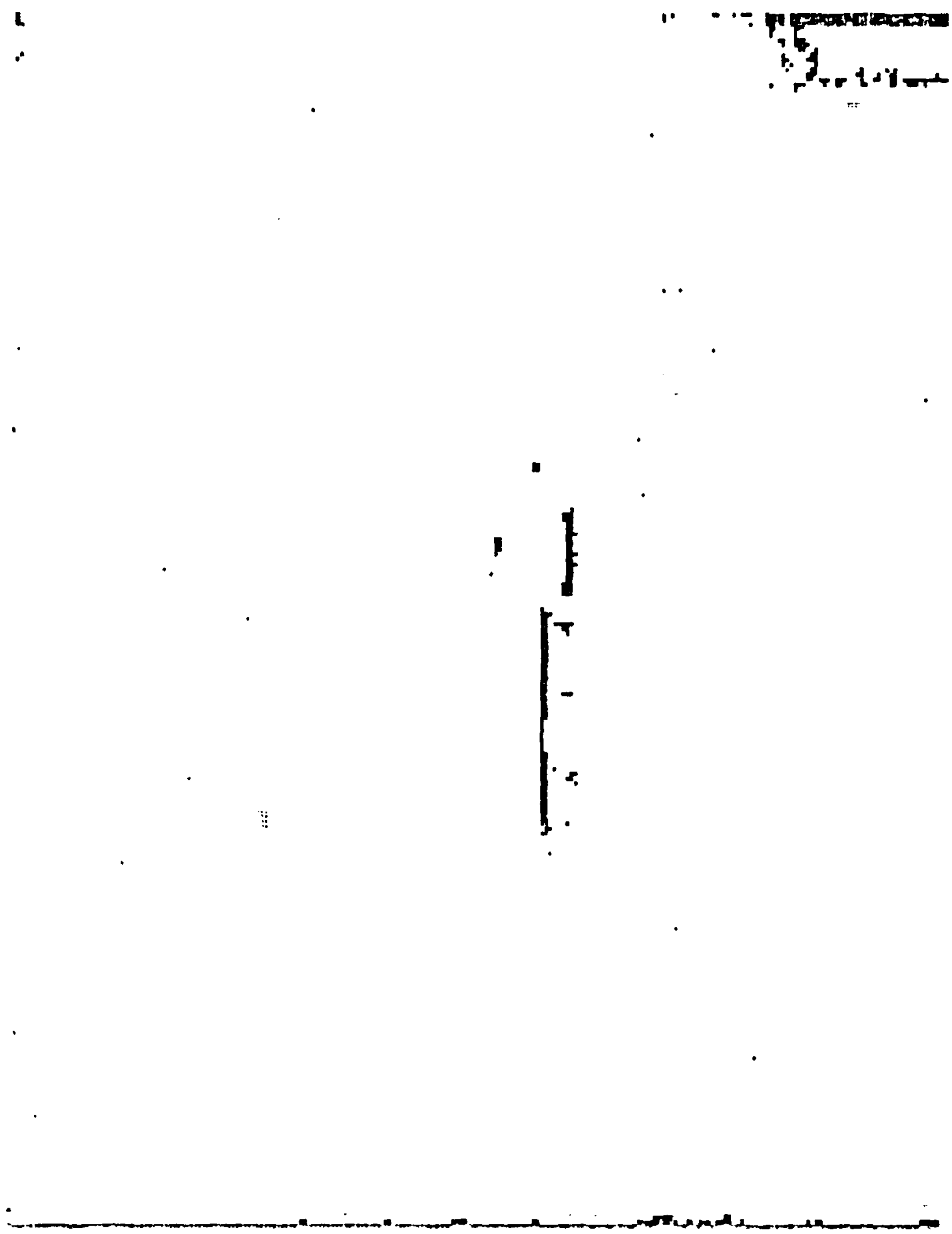

Fig. 160. Output Image Of Boolpass Operation 2
Threshold = 0

BOOLPASS OPERATION 2

1.	Digitized image Tomcat	:	Fig. 161
2.	1x7 low pass filter	:	Fig. 162
3.	1x3 low pass filter	:	Fig. 163
4.	AND (Ø Std Dev)	:	Fig. 164
5.	Negative of 4.	:	Fig. 165
6.	AND (Std Dev)/1Ø	:	Fig. 166
7.	AND (Ø Std Dev)/1Ø	:	Fig. 167

Mean: 1Ø   Standard Deviation: 4

Fig. 161. Digitized video image of Tomcat

Fig. 162. Tomcat Processed By A 1x7 Low Pass Filter

Fig. 163. Tomcat Processed BY A 1x3 Low Pass Filter

Fig. 164. Boolean AND Operation Performed On Preceeding Two Figures. Threshold = Ø

Fig. 165. Negative Image of Fig. 164

Fig. 166. Output Image Of Boolpass Operation 2
Threshold = (Std Dev)/10

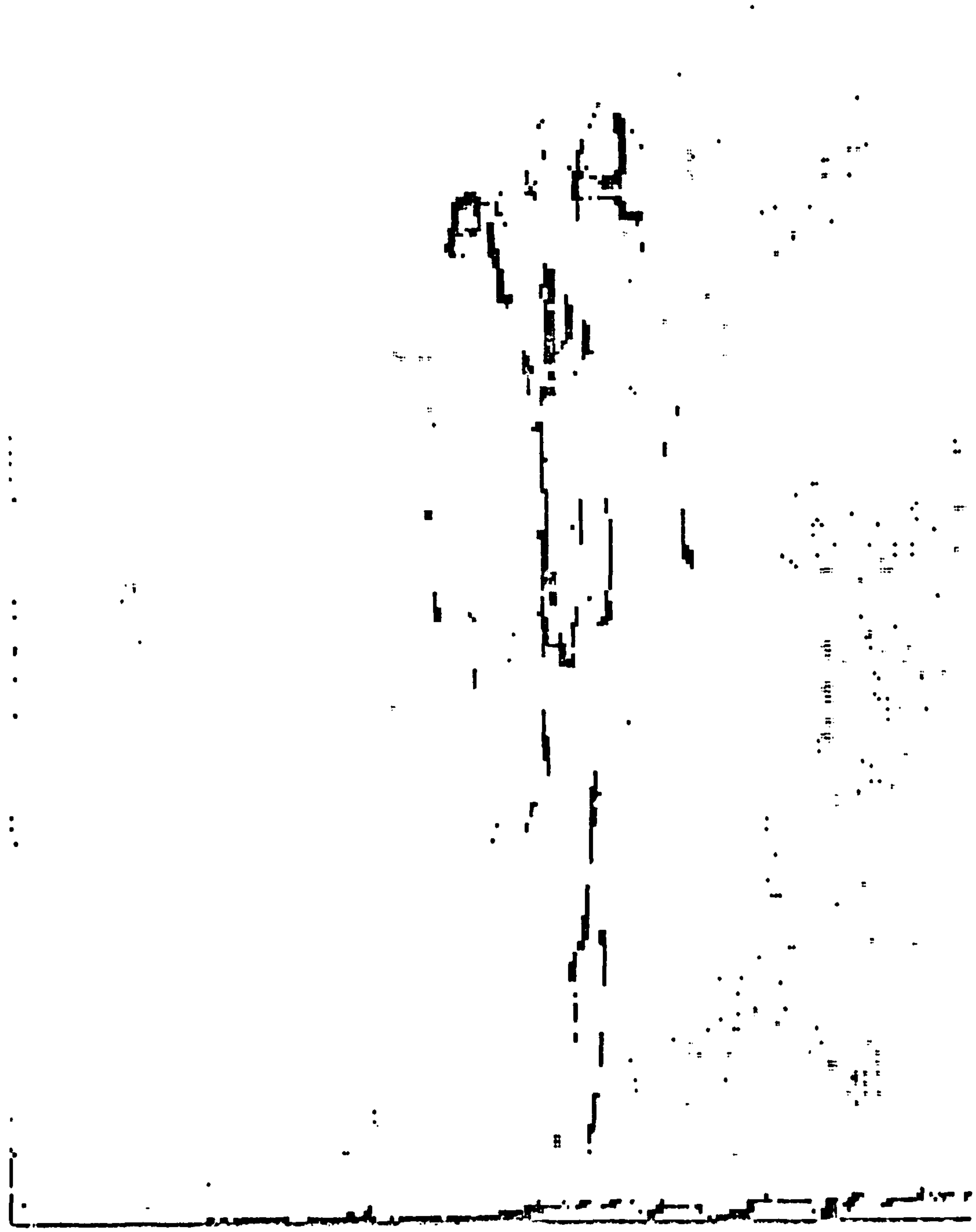

Fig. 167. Output Image Of Boolpass Operation 2
Threshold = Ø

BOOLPASS OPERATION 2

1.	Digitized image Field	:	Fig. 168
2.	1x7 low pass filter	:	Fig. 169
3.	1x3 low pass filter	:	Fig. 170
4.	AND (Ø Std Dev)	:	Fig. 171
5.	Negative of 4.	:	Fig. 172
6.	AND (Std Dev)/10	:	Fig. 173
7.	AND (Ø Std Dev)/10	:	Fig. 174

Mean: 8 Standard Deviation: 5

Fig. 168. Digitized video image of Field

Fig. 169. Field Processed By A 1x7 Low Pass Filter

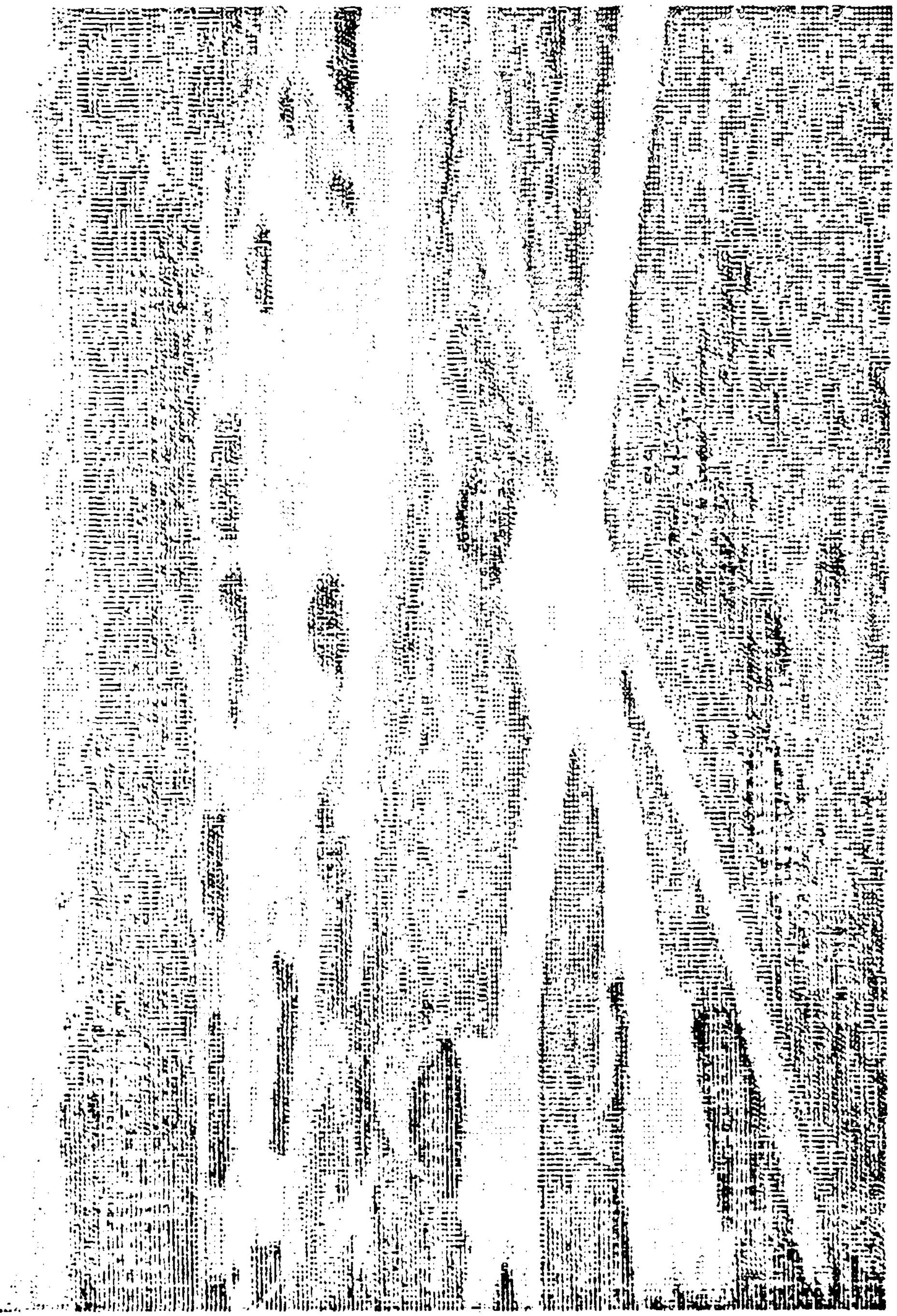

Fig. 17Ø. Field Processed BY A 1x3 Low Pass Filter

Fig. 171. Boolean AND Operation Performed On Preceeding Two Figures. Threshold = Ø

Fig. 172. Negative Image of Fig. 171

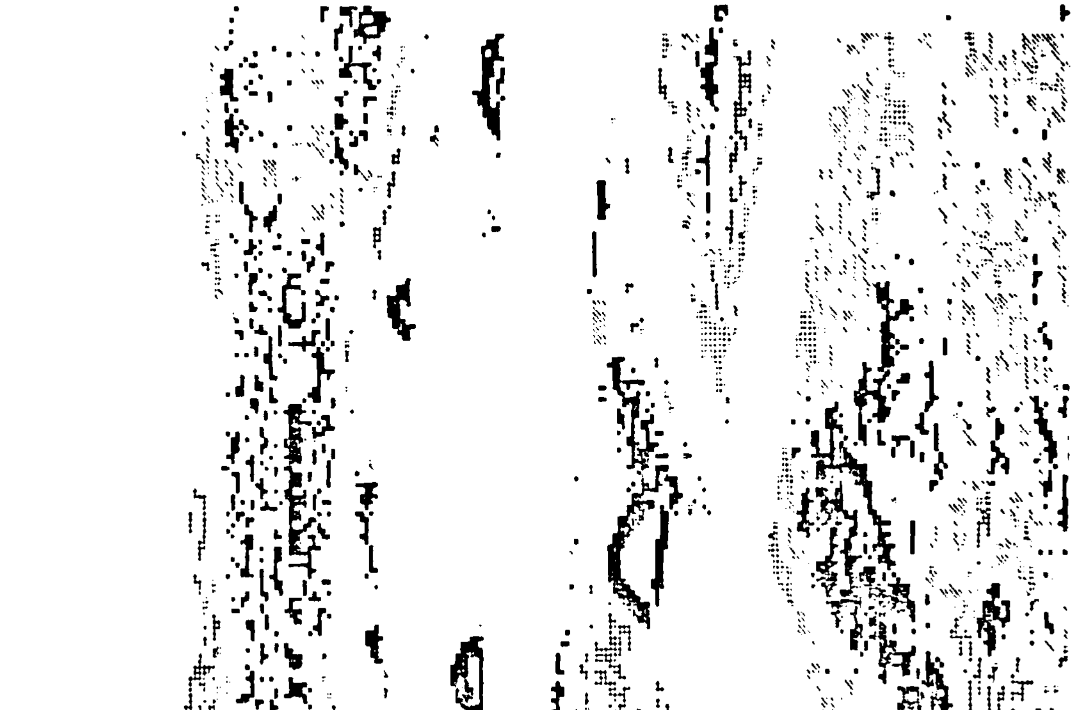

Fig. 173. Output Image Of Boolpass Operation 2
Threshold = (Std Dev)/10

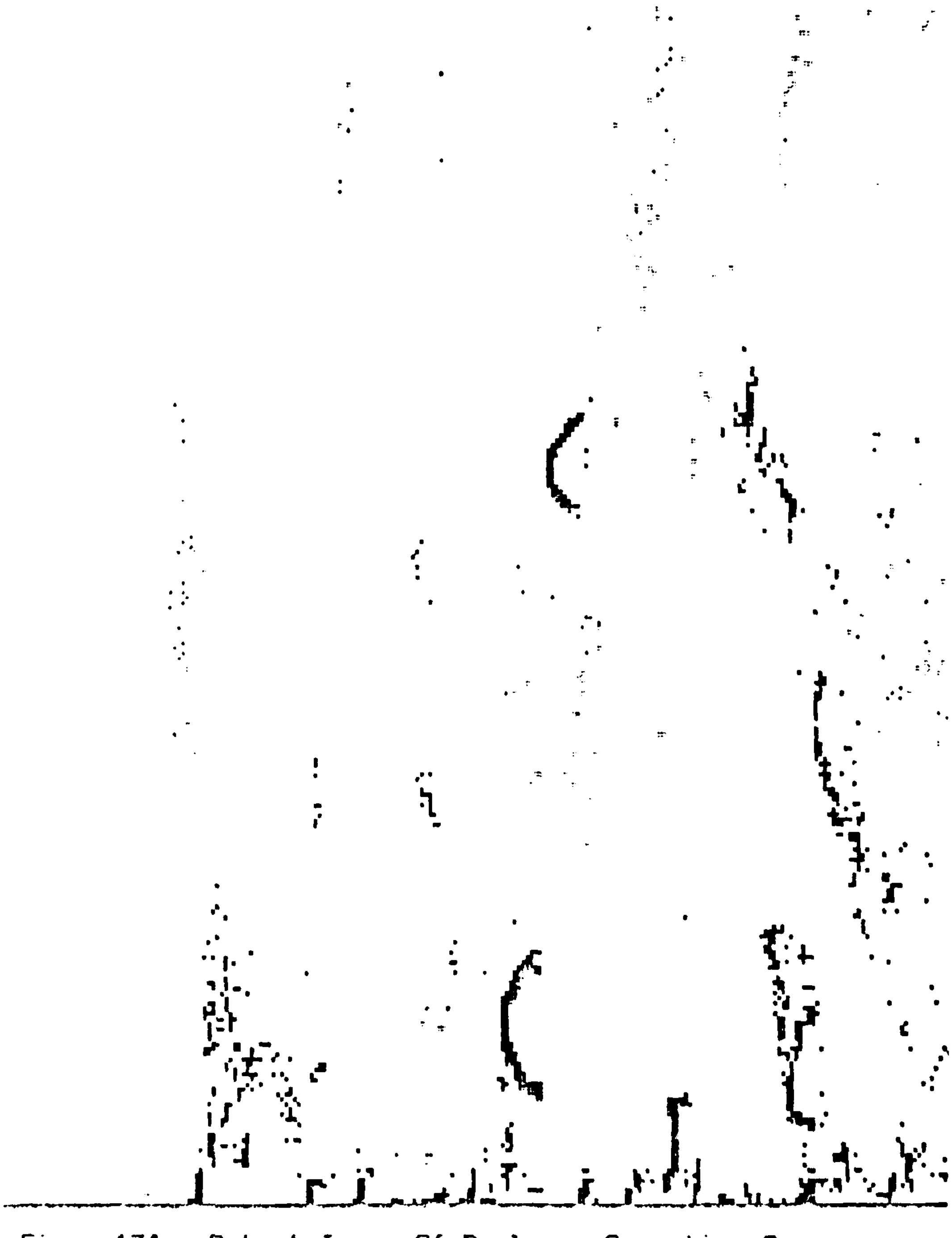

Fig. 174. Output Image Of Boolpass Operation 2
Threshold = Ø

BOOLPASS OPERATION 2

1.	Digitized image Field2	:	Fig. 175
2.	1x7 low pass filter	:	Fig. 176
3.	1x3 low pass filter	:	Fig. 177
4.	AND (Ø Std Dev)	:	Fig. 178
5.	Negative of 4.	:	Fig. 179
6.	AND (Std Dev)/1Ø	:	Fig. 18Ø
7.	AND (Ø Std Dev)/1Ø	:	Fig. 181

Mean: 9 Standard Deviation: 5

Fig. 175. Digitized video image of Field2

Fig. 176. Field2 Processed By A 1x7 Low Pass Filter

Fig. 177. Field2 Processed BY A 1x3 Low Pass Filter

Fig. 178. Boolean AND Operation Performed On Preceeding Two Figures. Threshold = Ø

Fig. 179. Negative Image of Fig. 178

Fig. 180. Output Image Of Boolpass Operation 2
Threshold = (Std Dev)/10

Fig. 181. Output Image Of Boolpass Operation 2
Threshold = Ø

BOOLPASS OPERATION 2

1.	Digitized image Tank	:	Fig. 182
2.	1x7 low pass filter	:	Fig. 183
3.	1x3 low pass filter	:	Fig. 184
4.	AND (Ø Std Dev)	:	Fig. 185
5.	Negative of 4.	:	Fig. 186
6.	AND (Std Dev)/1Ø	:	Fig. 187
7.	AND (Ø Std Dev)/1Ø	:	Fig. 188

Mean: 1Ø   Standard Deviation: 5

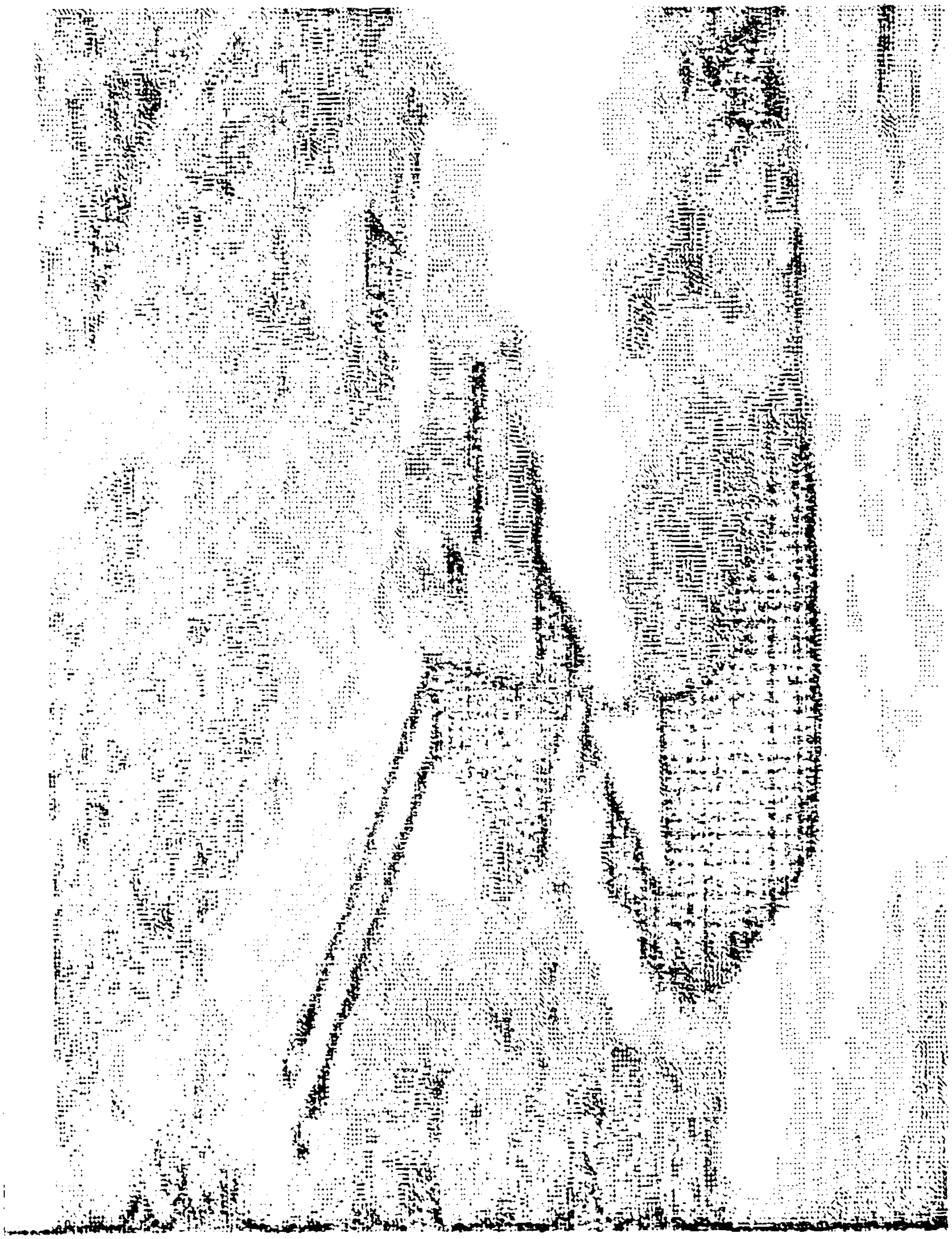

Fig. 182. Digitized video image of Tank

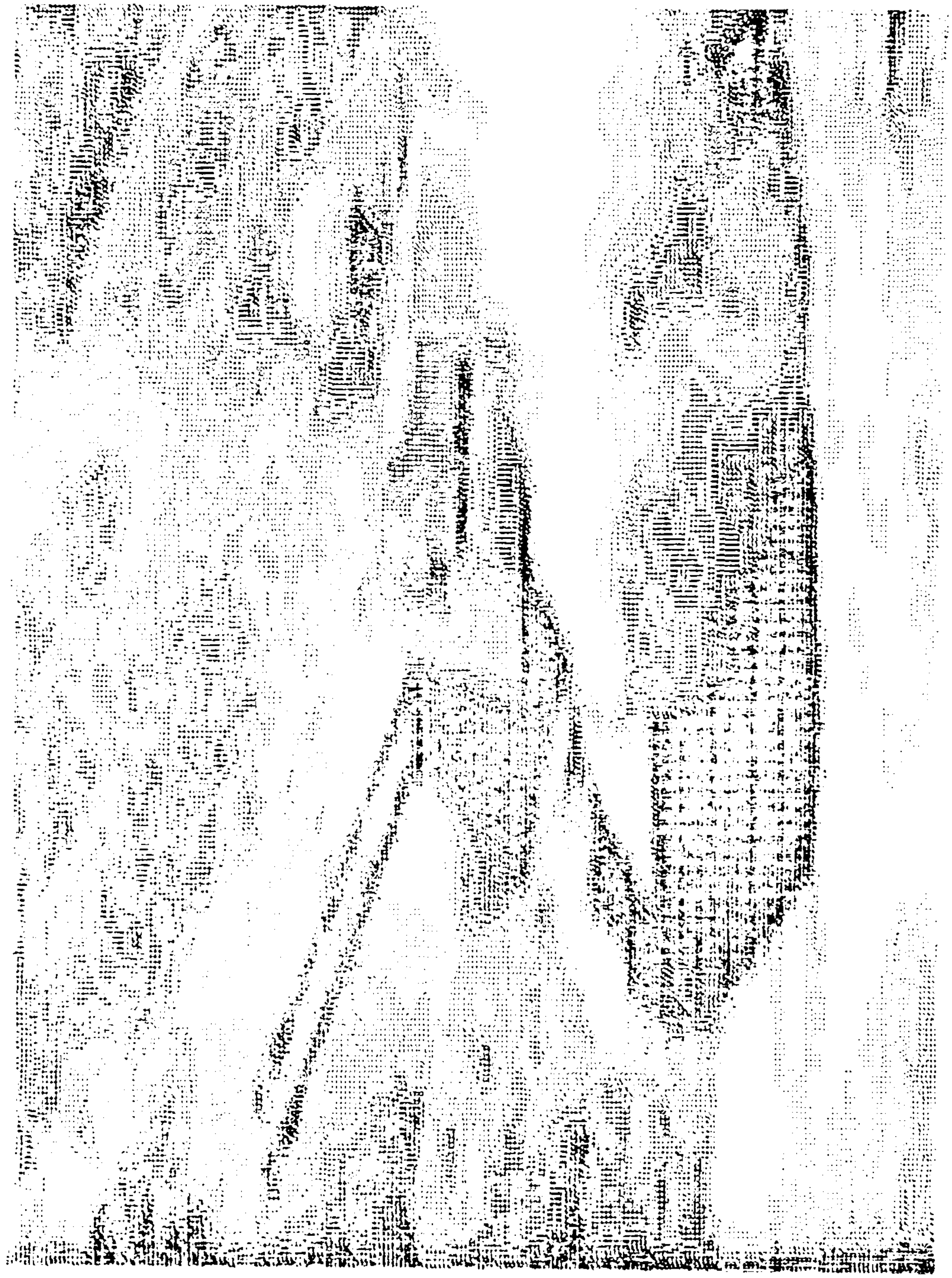

Fig. 183. Tank Processed By A 1x7 Low Pass Filter

Fig. 184. Tank Processed BY A 1x3 Low Pass Filter

Fig. 185. Boolean AND Operation Performed On Preceeding Two Figures. Threshold = Ø

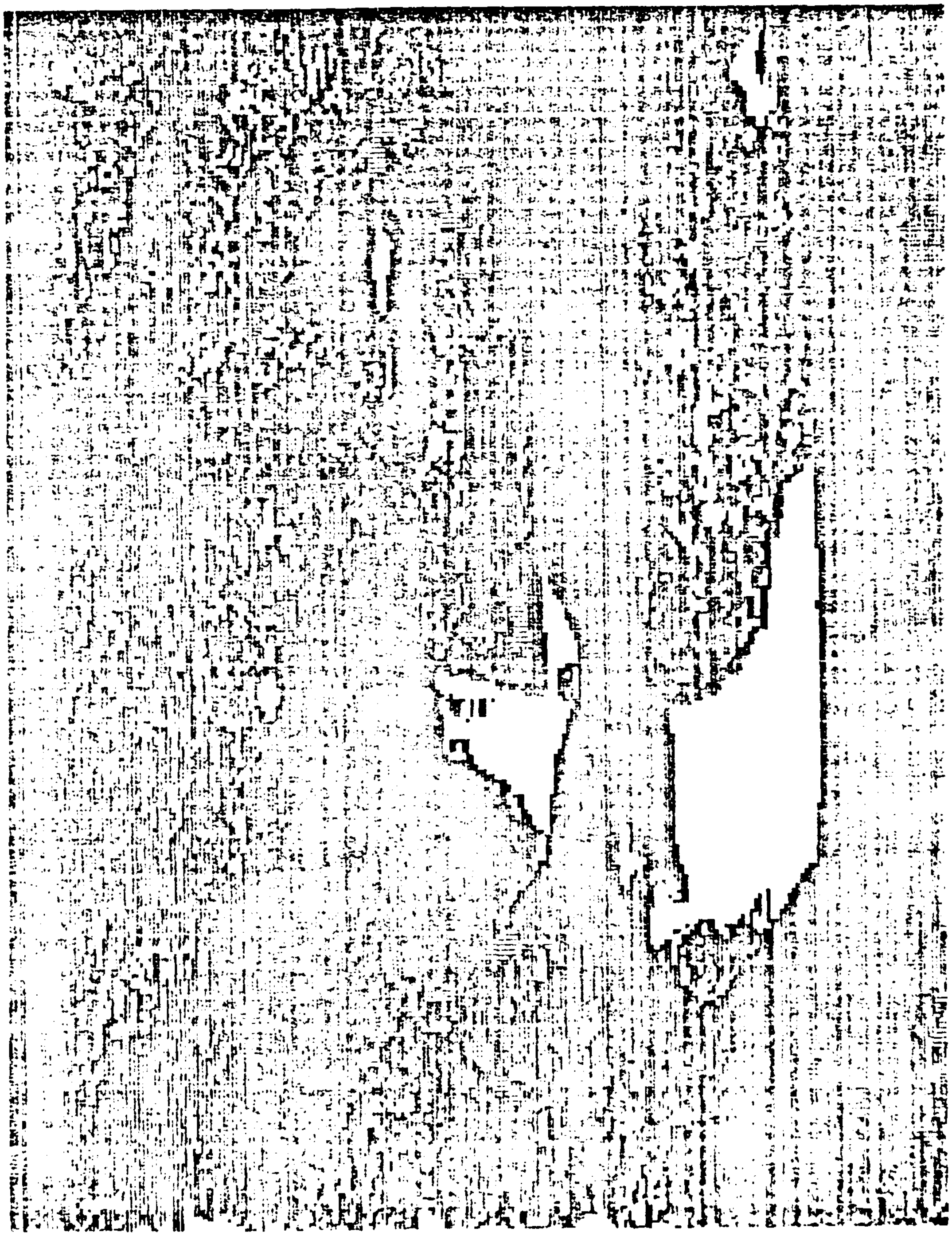

Fig. 186. Negative Image of Fig. 185

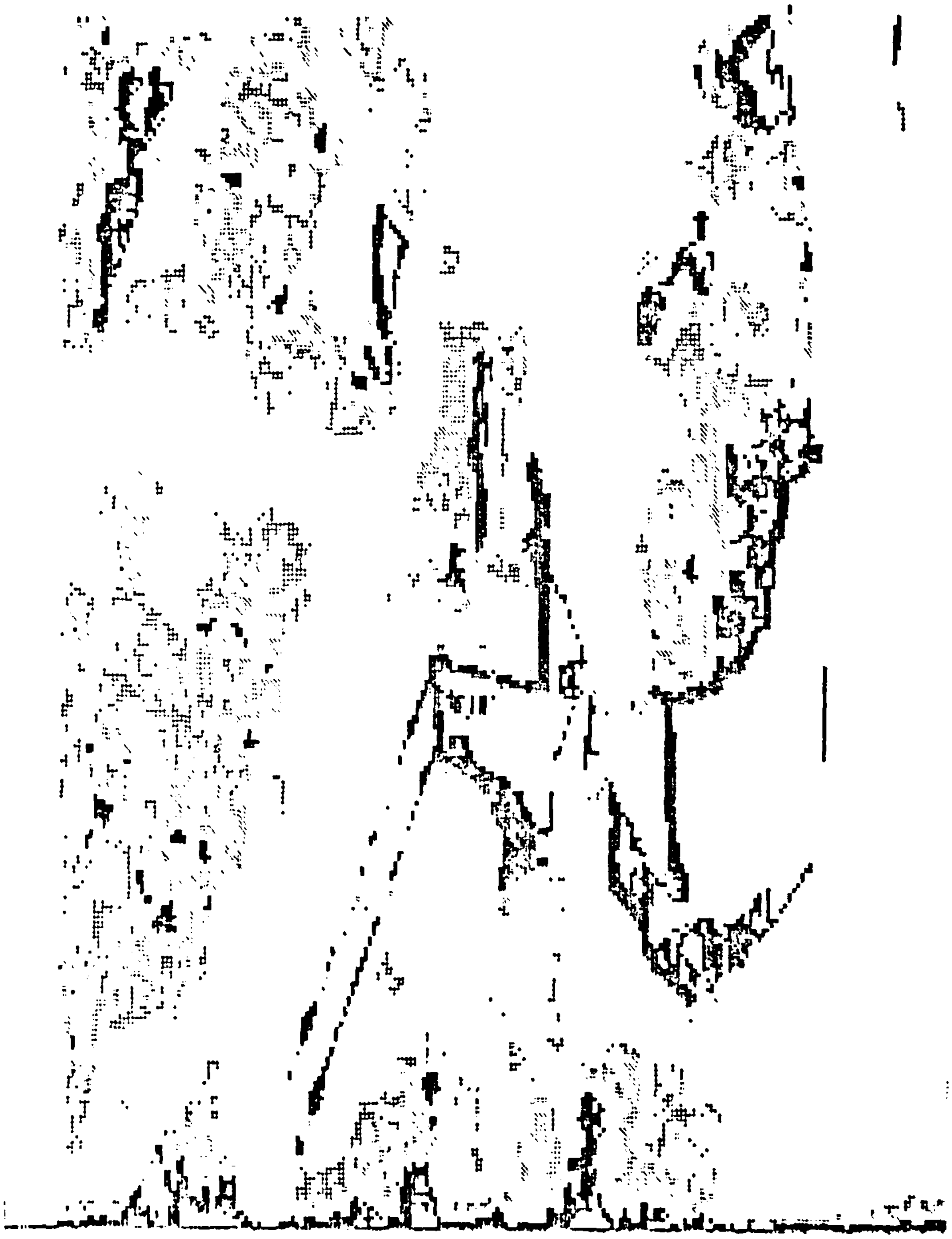

Fig. 187. Output Image Of Boolpass Operation 2
Threshold = (Std Dev)/10

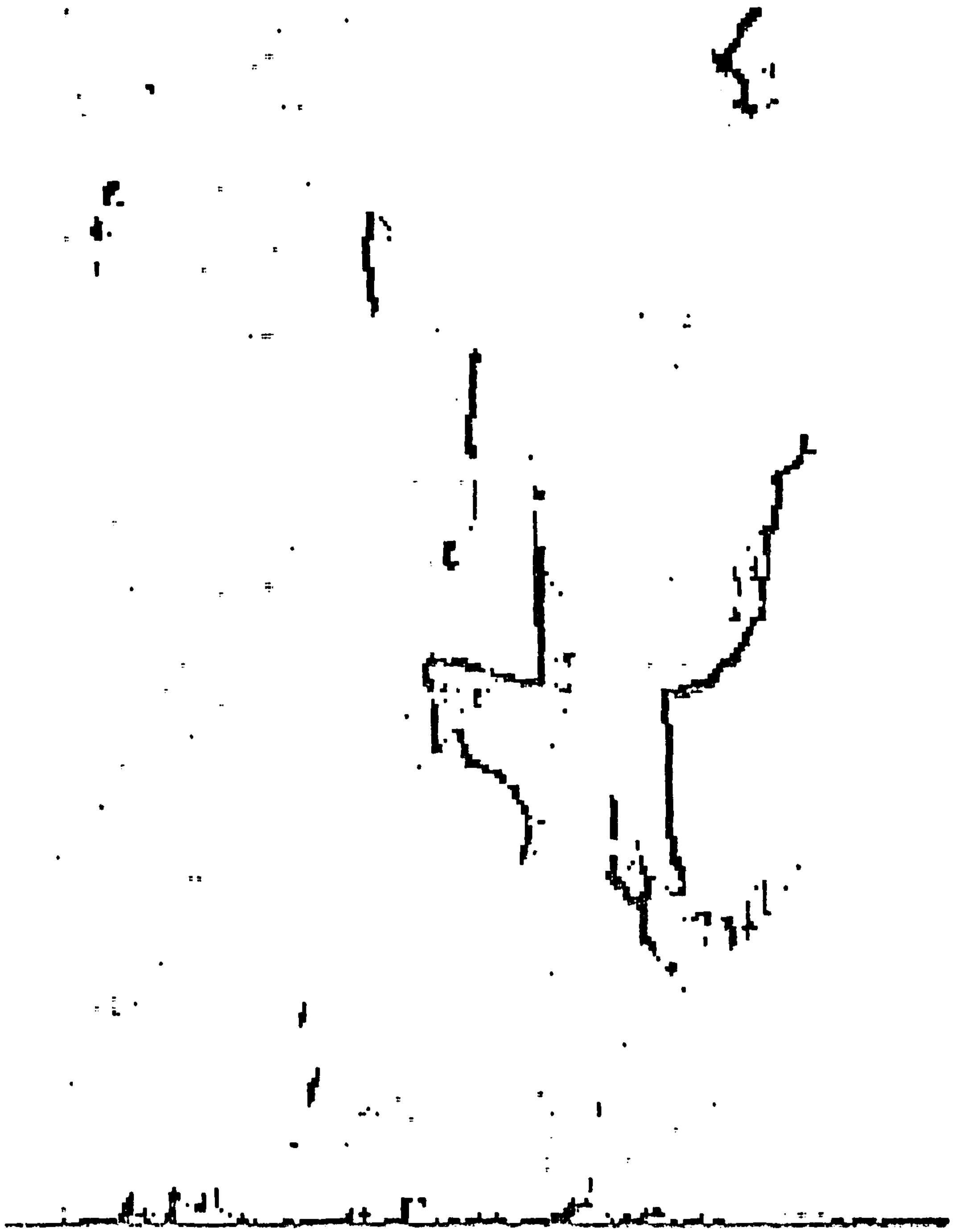

Fig. 188. Output Image Of Boolpass Operation 2
Threshold = Ø

BOOLPASS OPERATION 2

1.	Digitized image Dispersal	:	Fig. 189
2.	1x7 low pass filter	:	Fig. 190
3.	1x3 low pass filter	:	Fig. 191
4.	AND (0 Std Dev)	:	Fig. 192
5.	Negative of 4.	:	Fig. 193
6.	AND (Std Dev)/10	:	Fig. 194
7.	AND (0 Std Dev)/10	:	Fig. 195

Mean: 9 Standard Deviation: 5

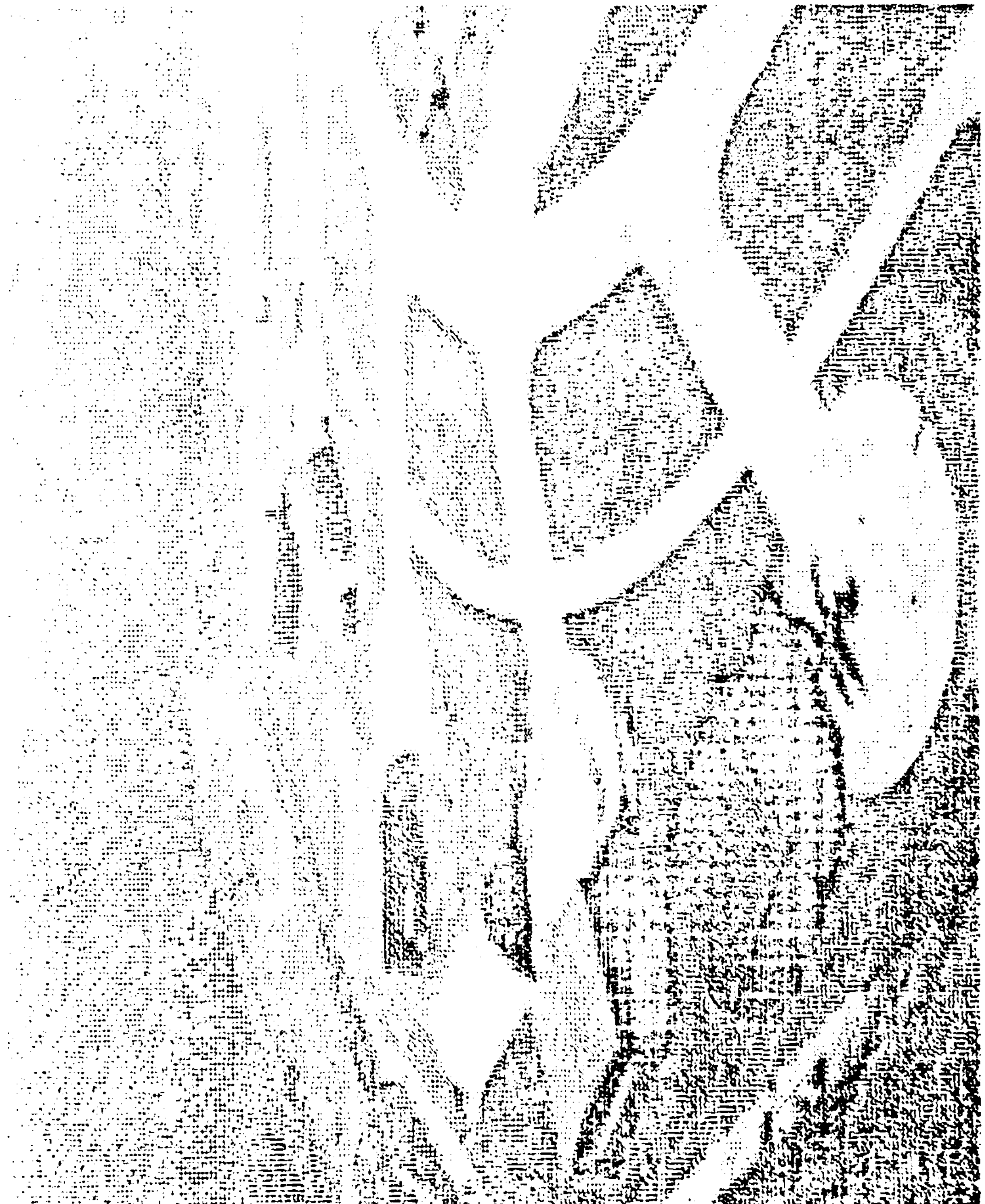

Fig. 189. Digitized video image of Dispersal

Fig. 190. Dispersl Processed By A 1x7 Low Pass Filter

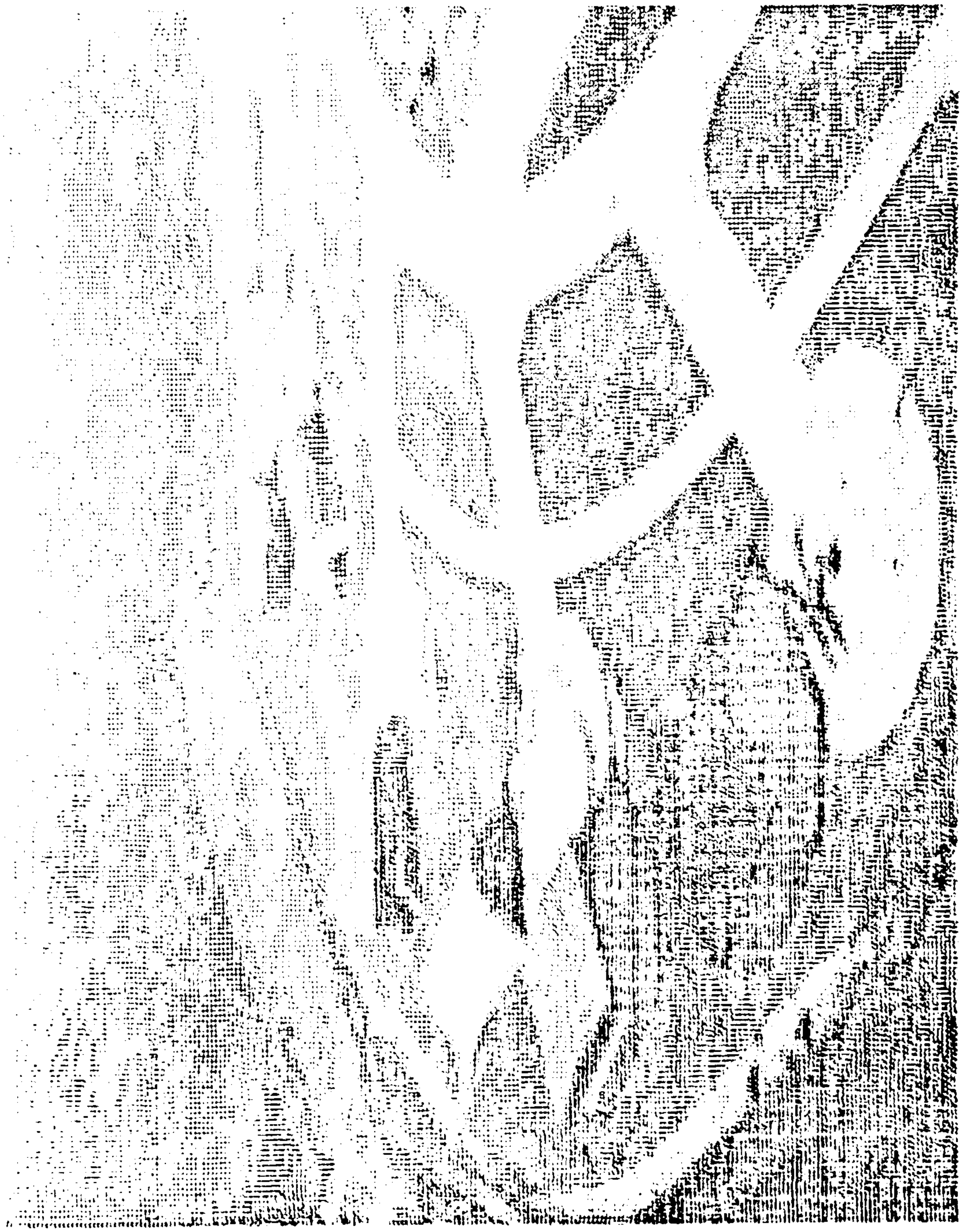

Fig. 191. Dispersal Processed BY A 1x3 Low Pass Filter

Fig. 192. Boolean AND Operation Performed On Preceeding Two Figures. Threshold = Ø

Fig. 193. Negative Image of Fig. 192

Fig. 194. Output Image Of Boolpass Operation 2
Threshold = (Std Dev)/10

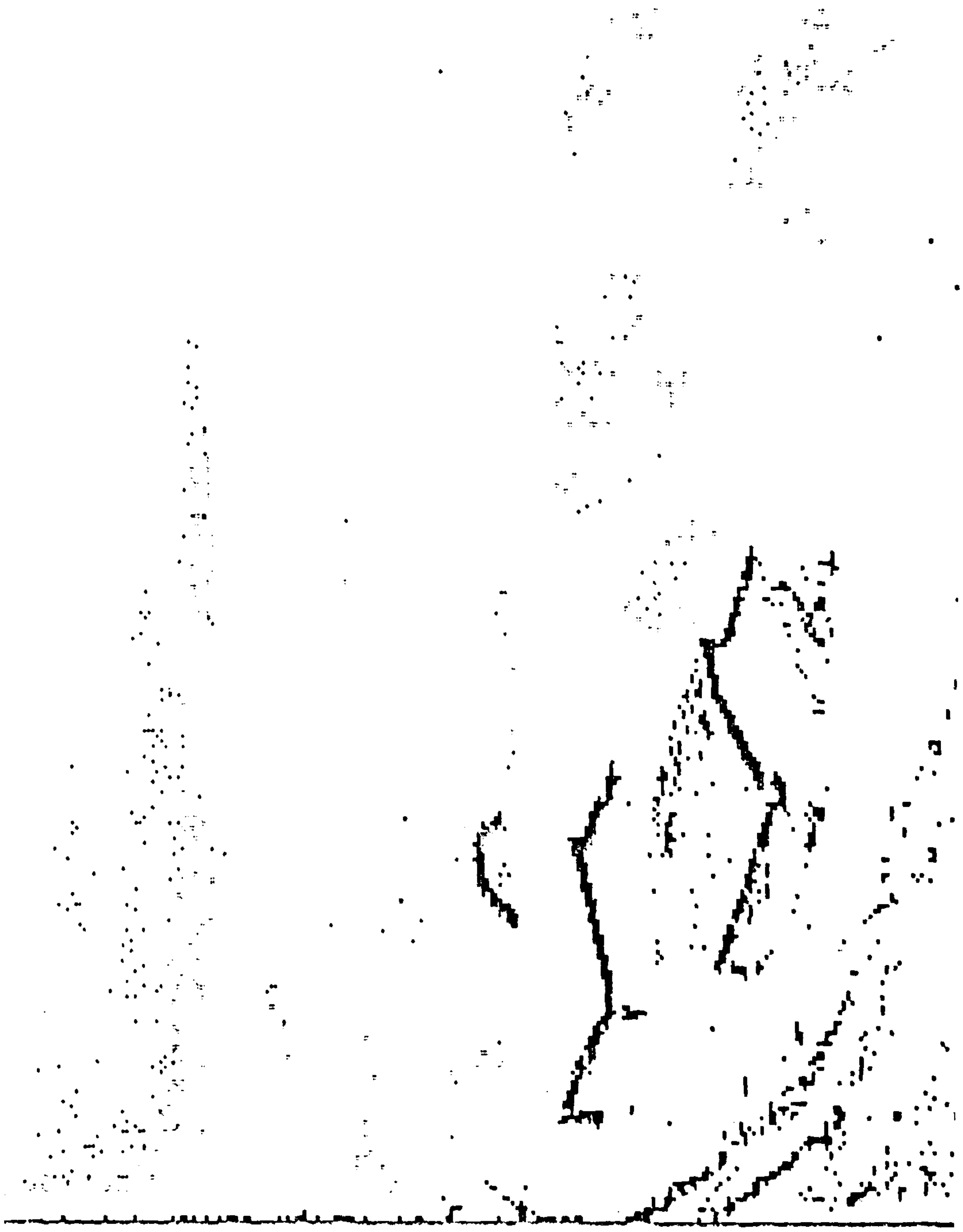

Fig. 195. Output Image Of Boolpass Operation 2
Threshold = Ø

AD-A118 076 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 20/6
COMBINED SPATIAL FILTERING AND BOOLEAN OPERATORS APPLIED TO THE--ETC(U)
JUN 82 B E FELTMATE
UNCLASSIFIED AFIT/GCS/EE/82J-8 NL

4 OF 4
AD A118076

END
7-82

Appendix B

3-Dimensional Picture/Pixel Intensity Plots

The following 3-dimensional plots help to visually display to the reader the exact pixel intensity distribution over an entire digitized image. This appendix contains pixel intensity plots for each of the original digitized images used in Appendix A. An example 3-dimensional intensity plot is also given for one of the Boolpass Operator 1 output images (for each of the eight original images used in Appendix A).

A large amount of overall energy reduction is apparent in all images processed by Boolpass Operator 1.

NOTE: Anomalies existed in the computational routines used for making these plots. Therefore, the graphs which follow are reversed from the original image. For example, an aircraft pointed toward the right side of the original image points to the left side of the image after being processing by the 3-dimensional plot routines. Despite this distraction, the plots are still considered constructive visual aids to the better understanding of the image processing operation done by a Boolpass operator.

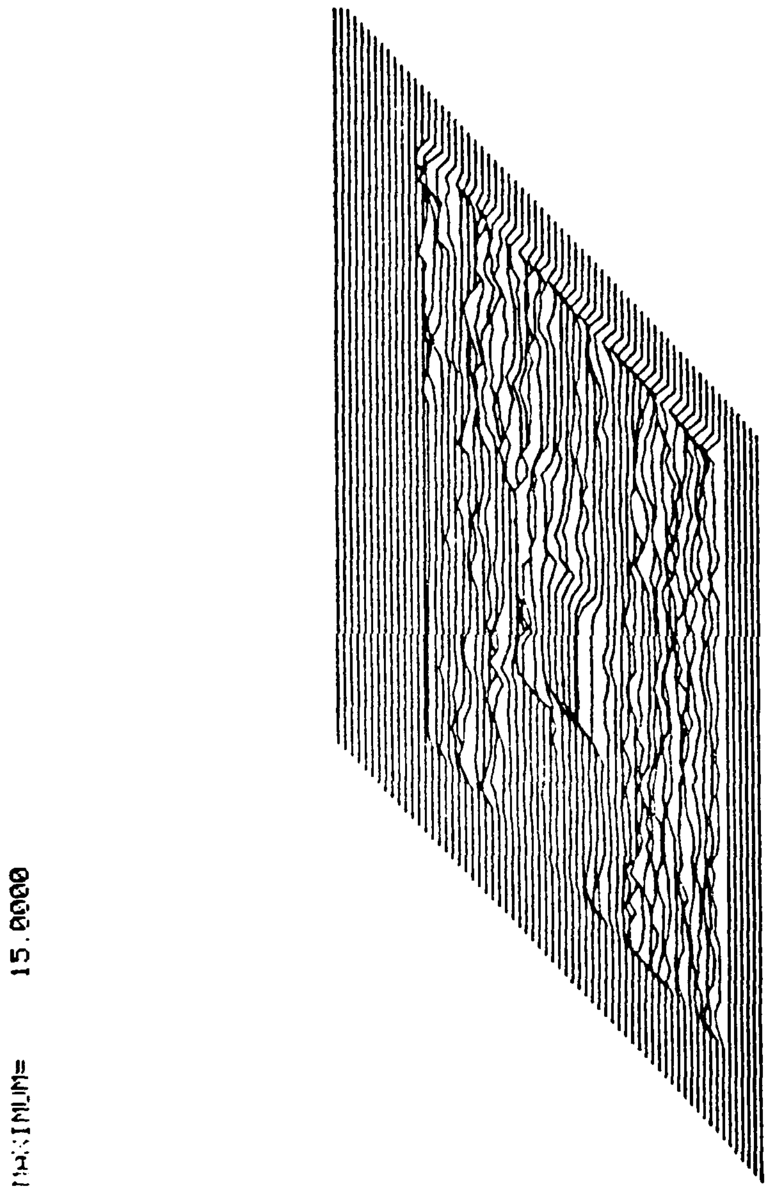

Fig. 196. 3-Dimensional Pixel Plot of Truck - Fig. 43

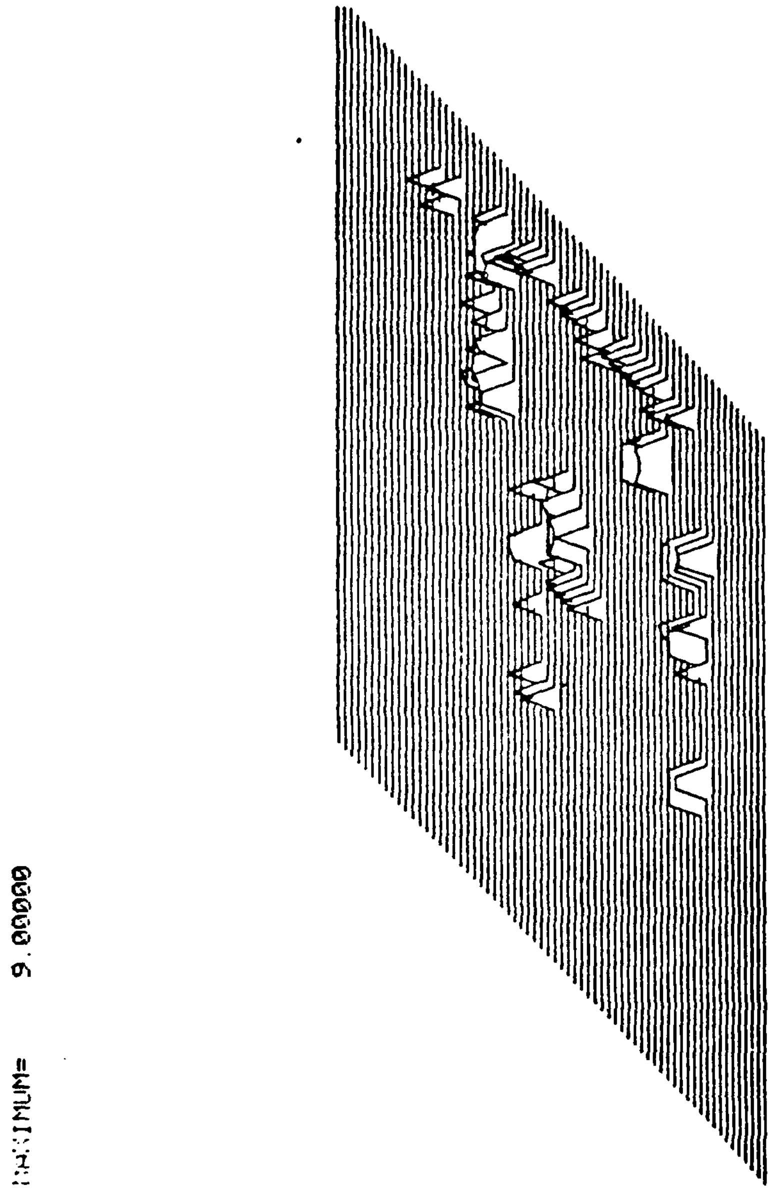

Fig. 197. 3-Dimensional Pixel Plot of Fig. 46
Boolpass Opeator 1 Used For Processing

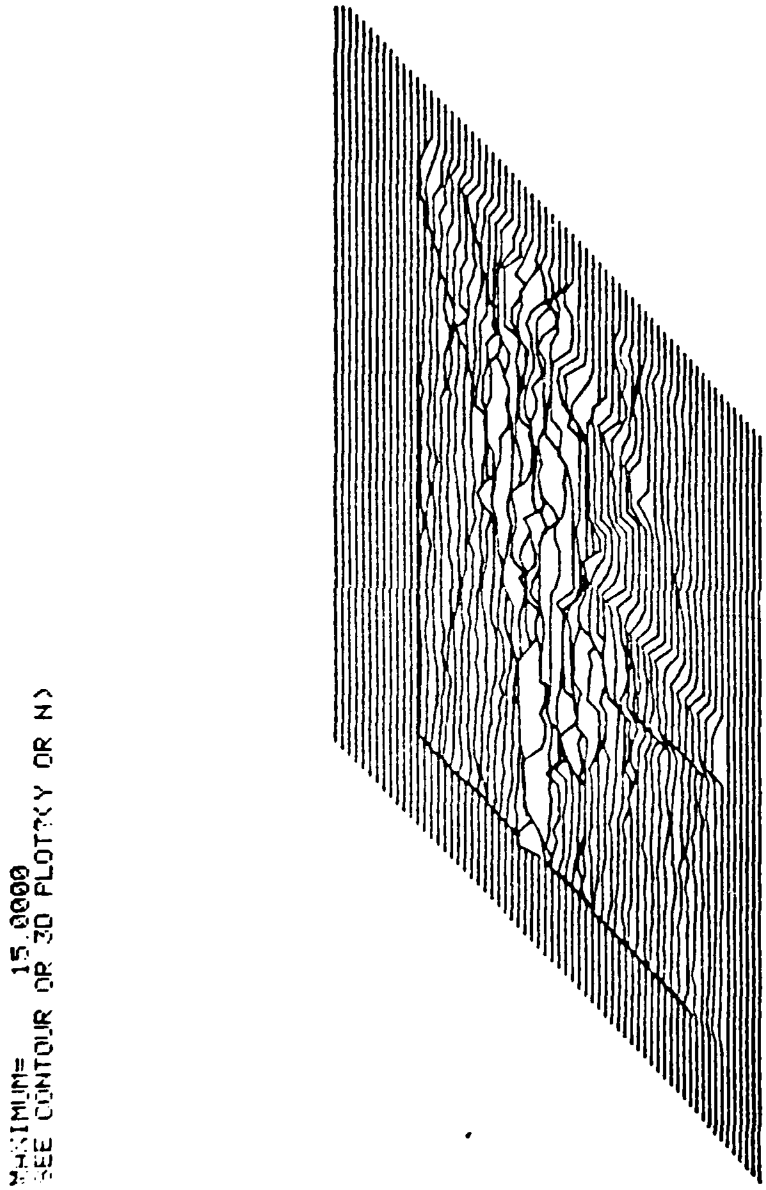

Fig. 198. 3-Dimensional Pixel Plot of F14 Tomcat, Fig. 48

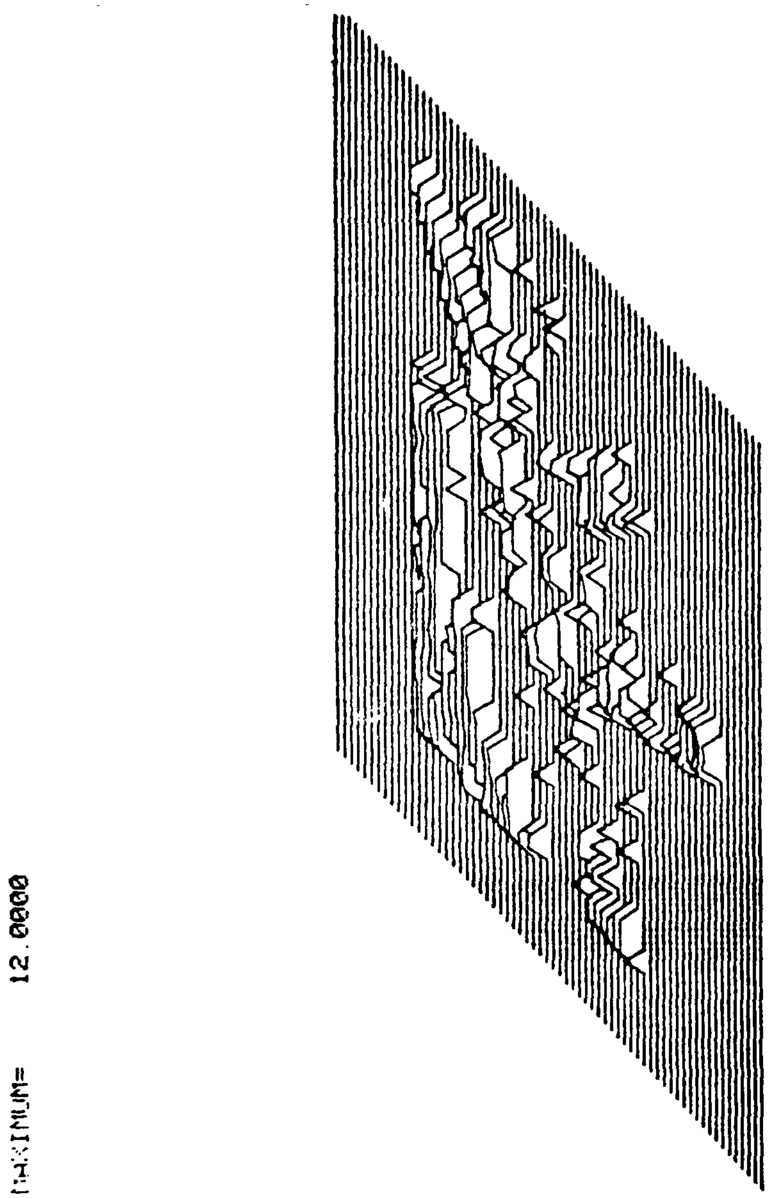

Fig. 199. 3-Dimensional Pixel Plot of Fig. 51
Boolpass Operator 1 Used for Processing

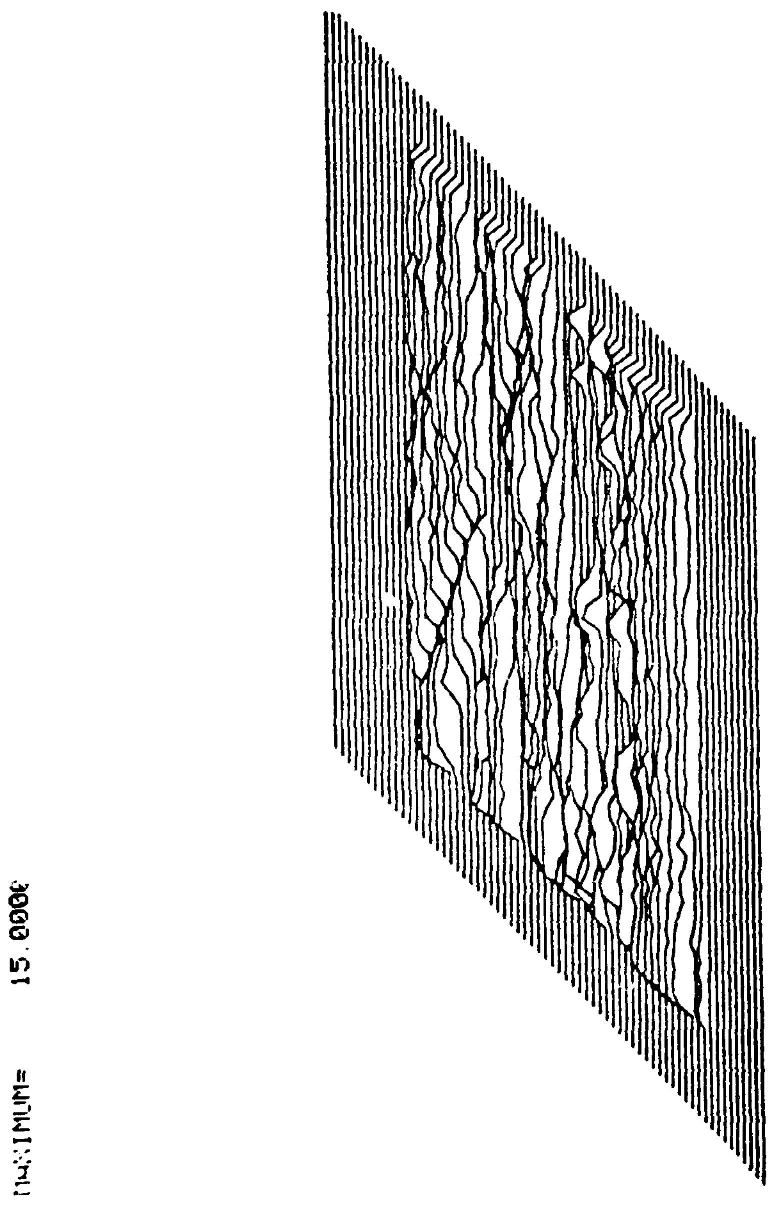

Fig. 200. 3-Dimensional Pixel Plot of Field, Fig. 53

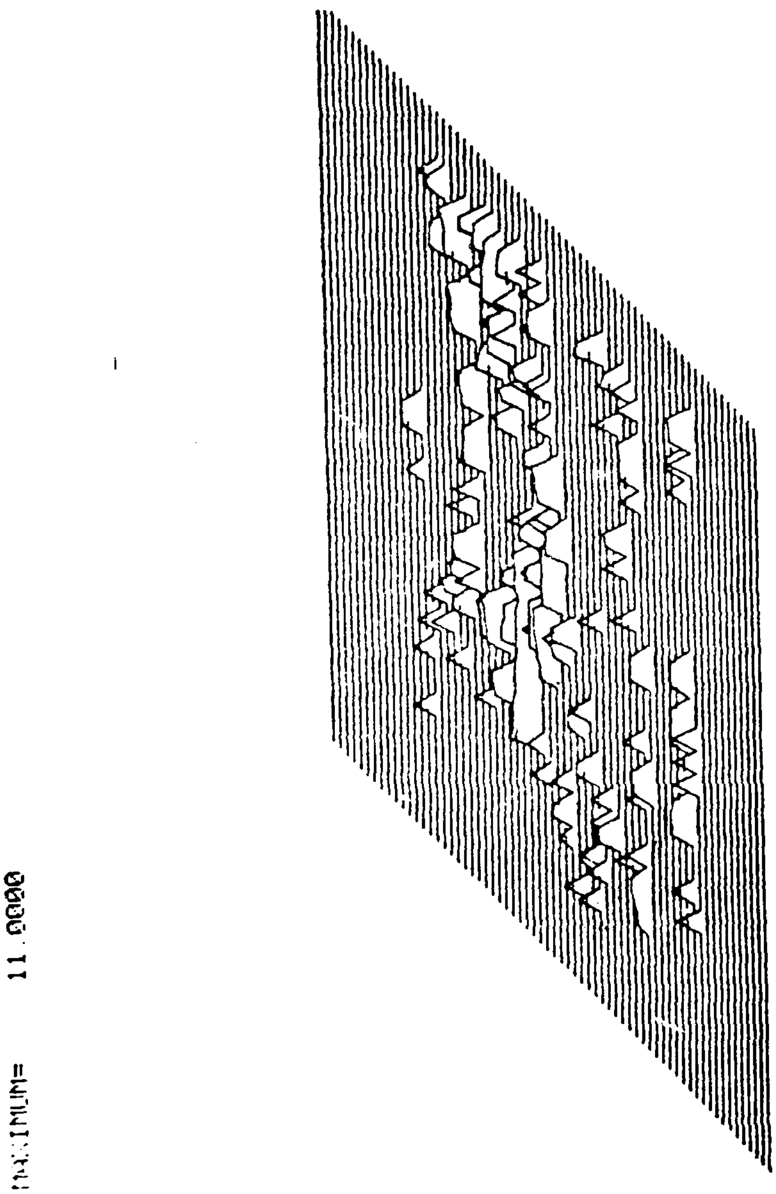

Fig. 201. 3-Dimensional Pixel Plot of Fig. 56
Boolpass Operator 1 Used For Processing

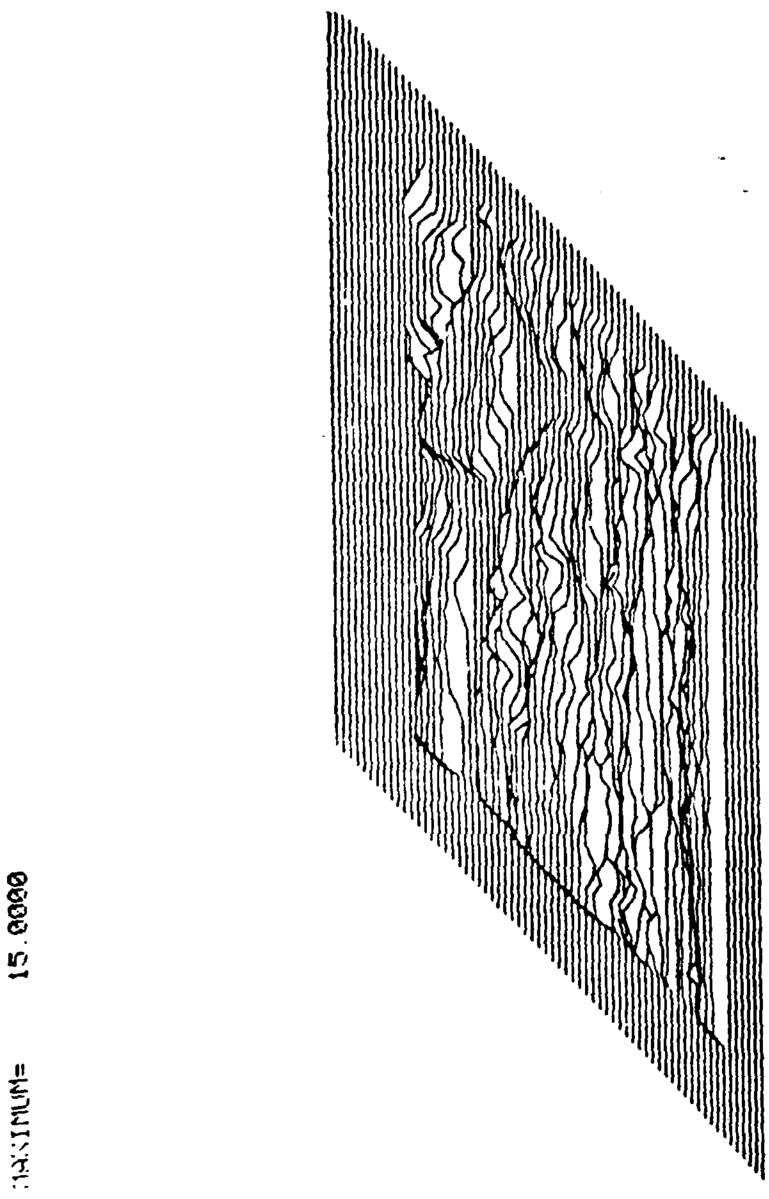

Fig. 202. 3-Dimensional Pixel Plot of Field2, Fig. 58

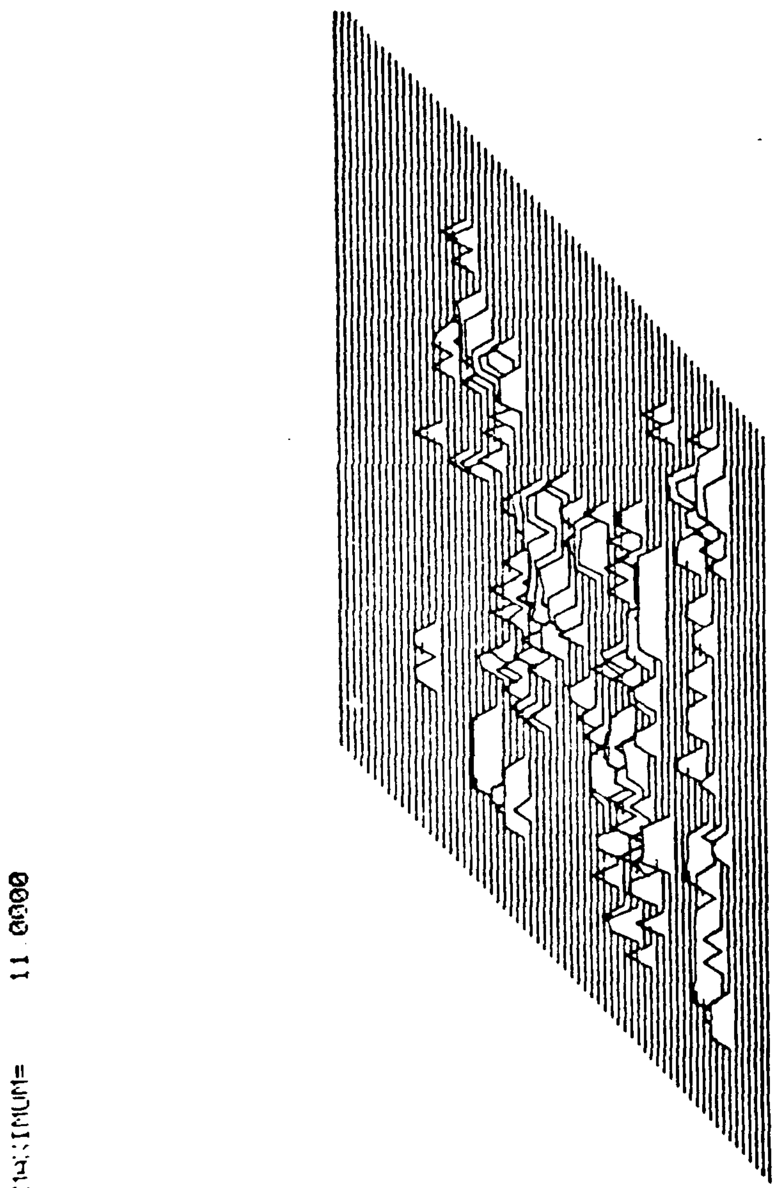

Fig. 203. 3-Dimensional Pixel Plot of Fig. 61
Boolpass Operator 1 Used For Processing

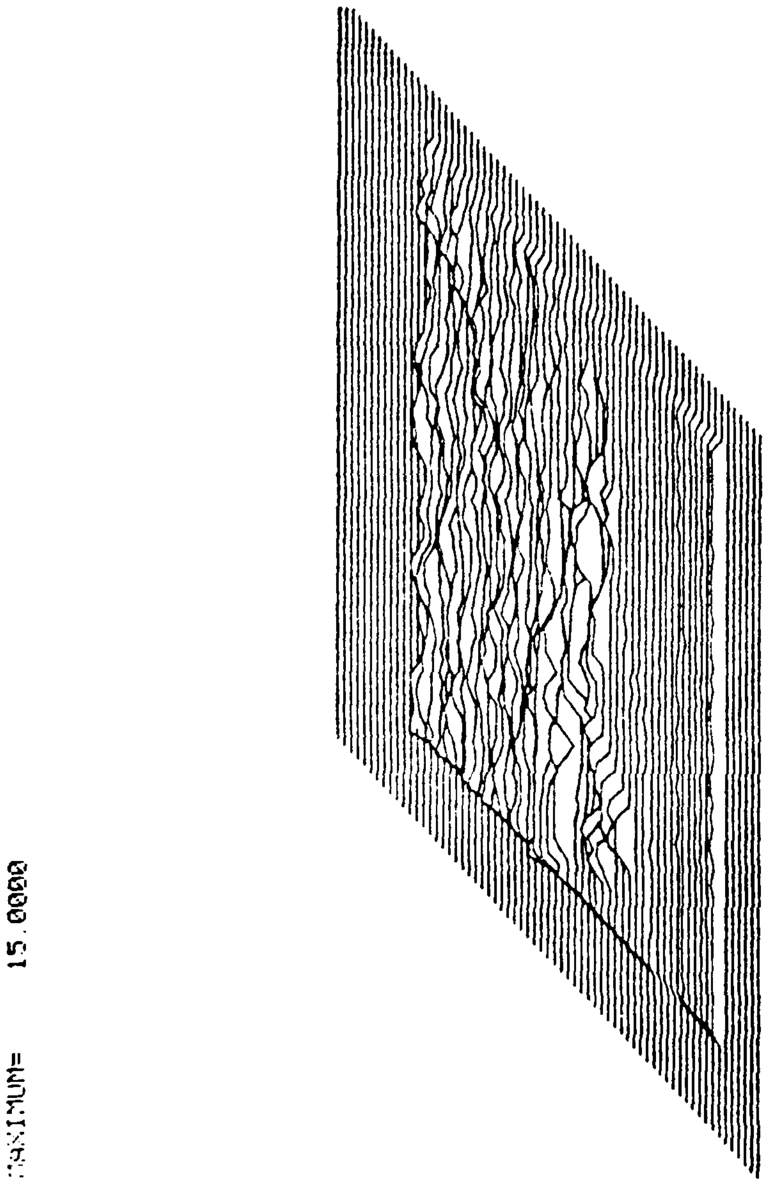

Fig. 204. 3-Dimensional Pixel Plot of F18 Hornet, Fig. 63

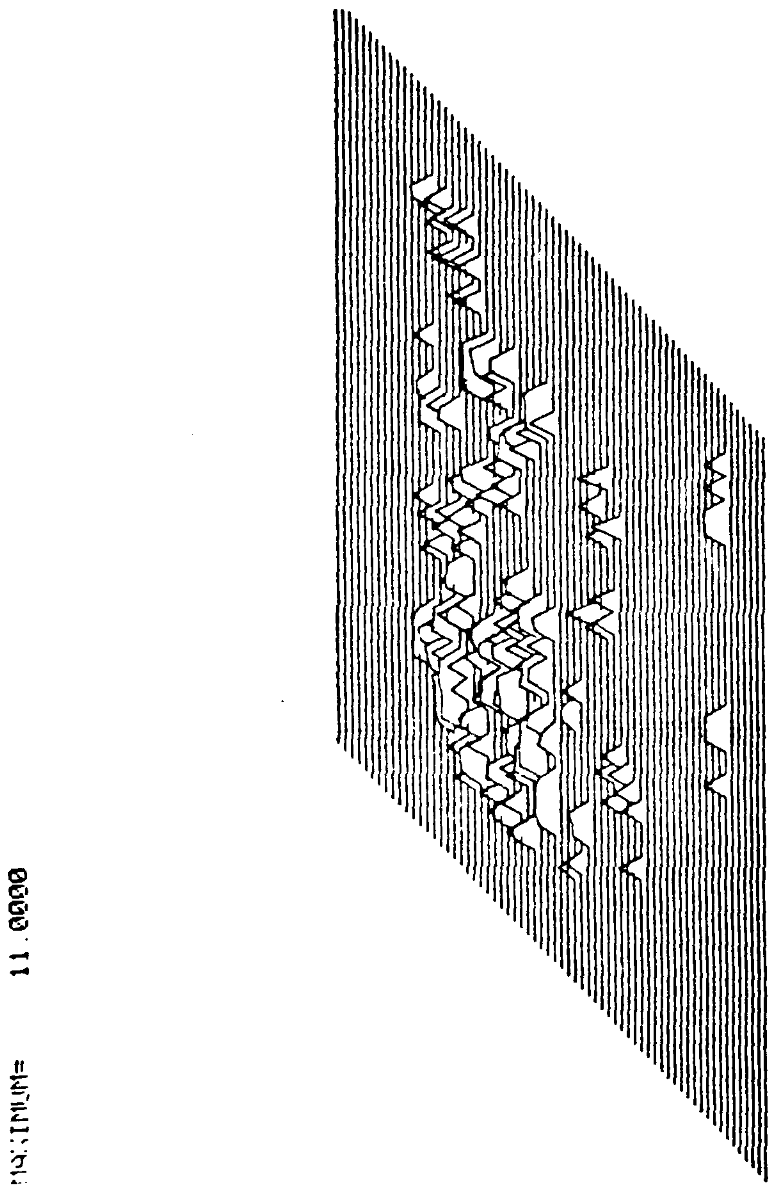

Fig. 205. 3-Dimensional Pixel Plot of Fig. 67
Boolpass Operator 1 Used For Processing

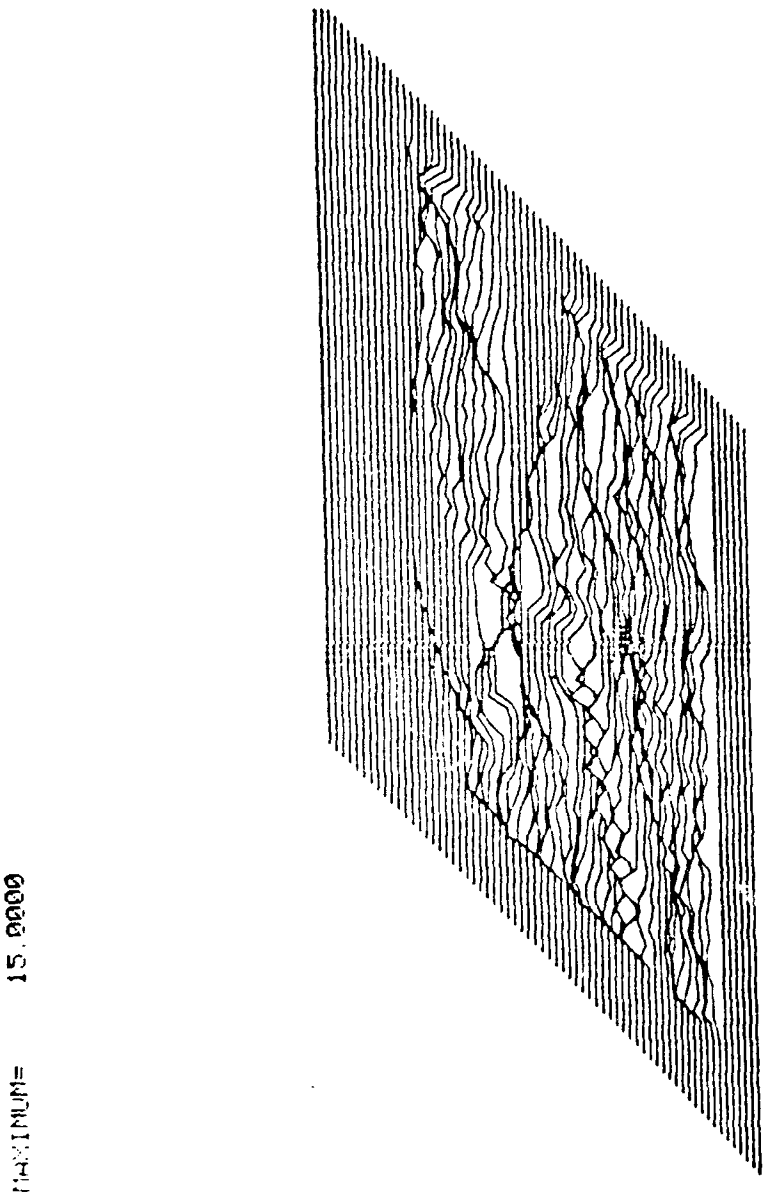

Fig. 206. 3-Dimensional Pixel Plot of Tank, Fig. 68

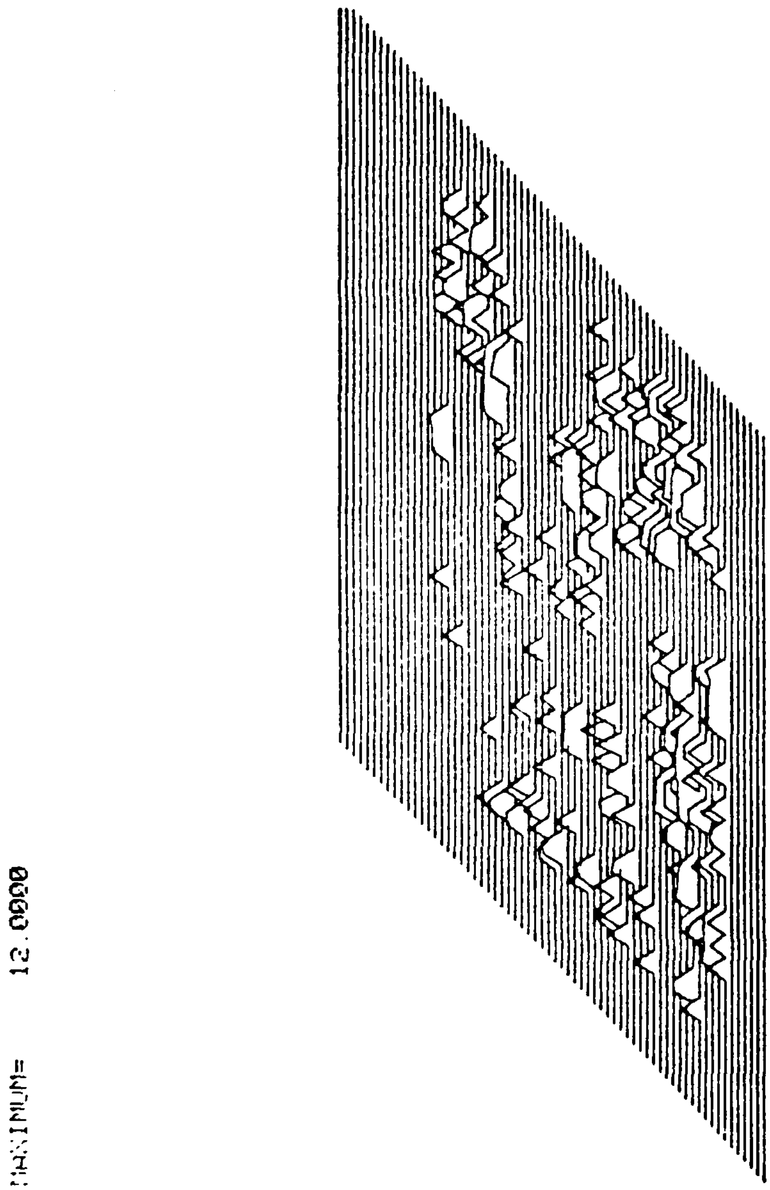

Fig. 207. 3-Dimensional Pixel Plot of Fig. 71
Boolpass Operator 1 Used For Processing

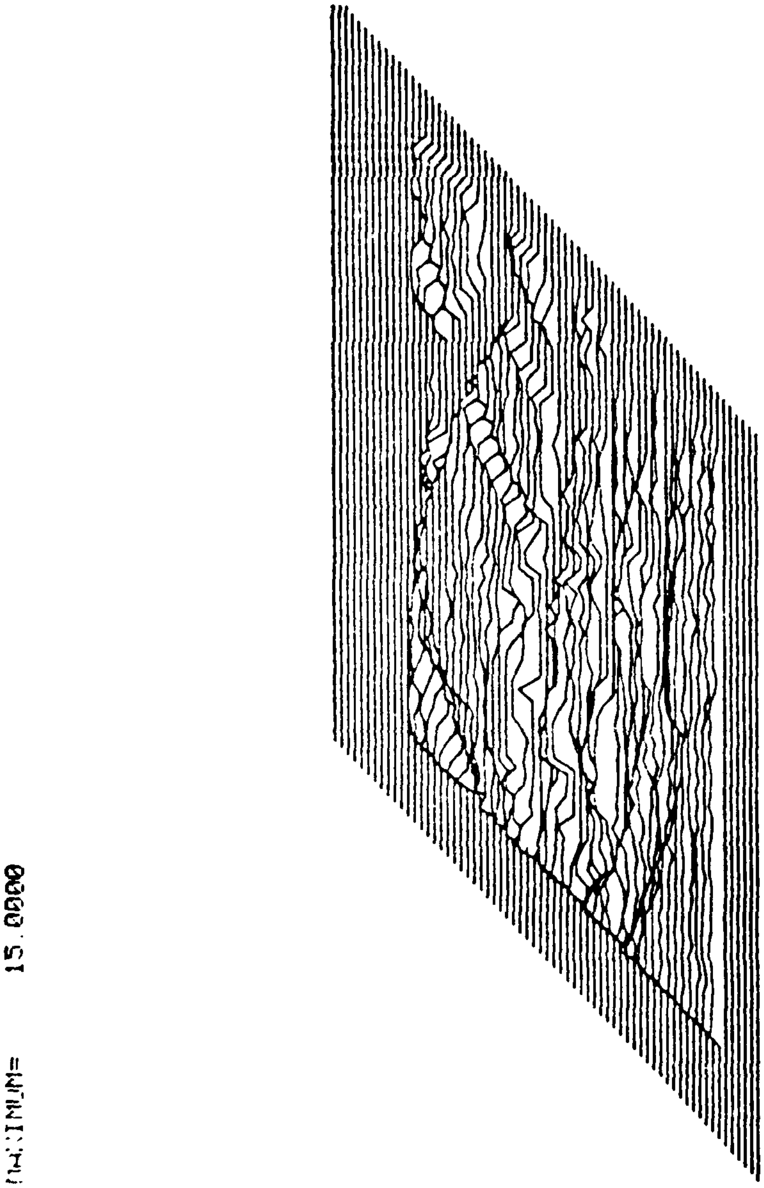

Fig. 208. 3-Dimensional Pixel Plot of Dispersal, Fig. 73

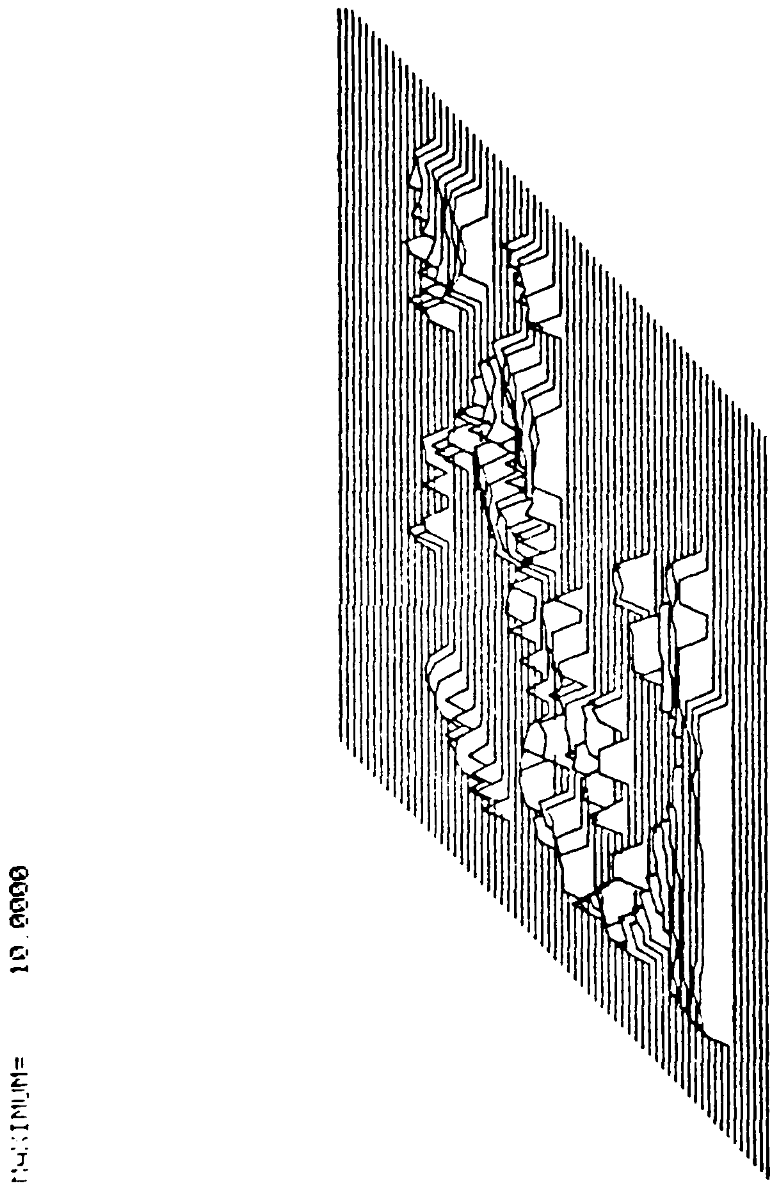

Fig. 209. 3-Dimensional Pixel Plot of Fig. 77
Boolpass Operator 1 Used For Processing

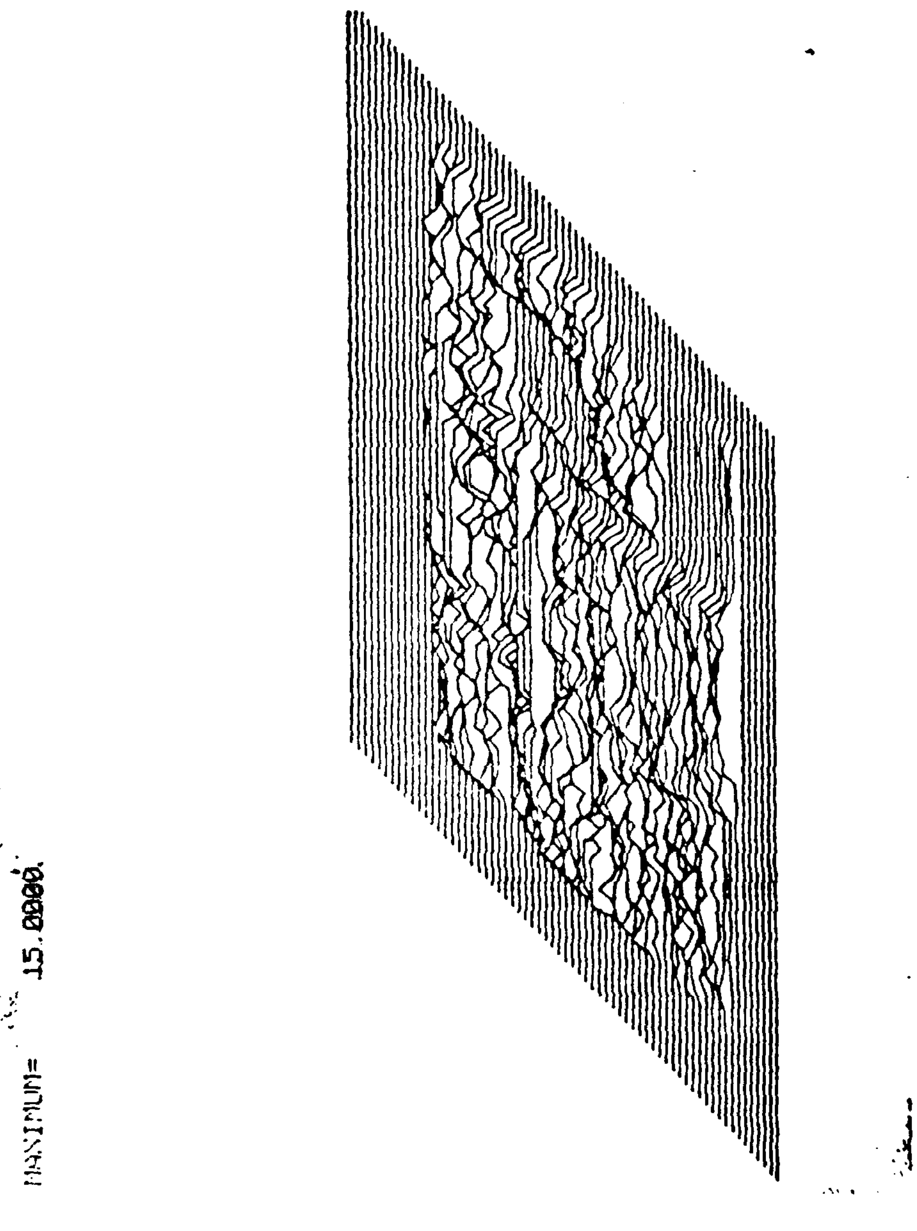

Fig. 210. 3-Dimensional Pixel Plot of Radar, Fig. 78

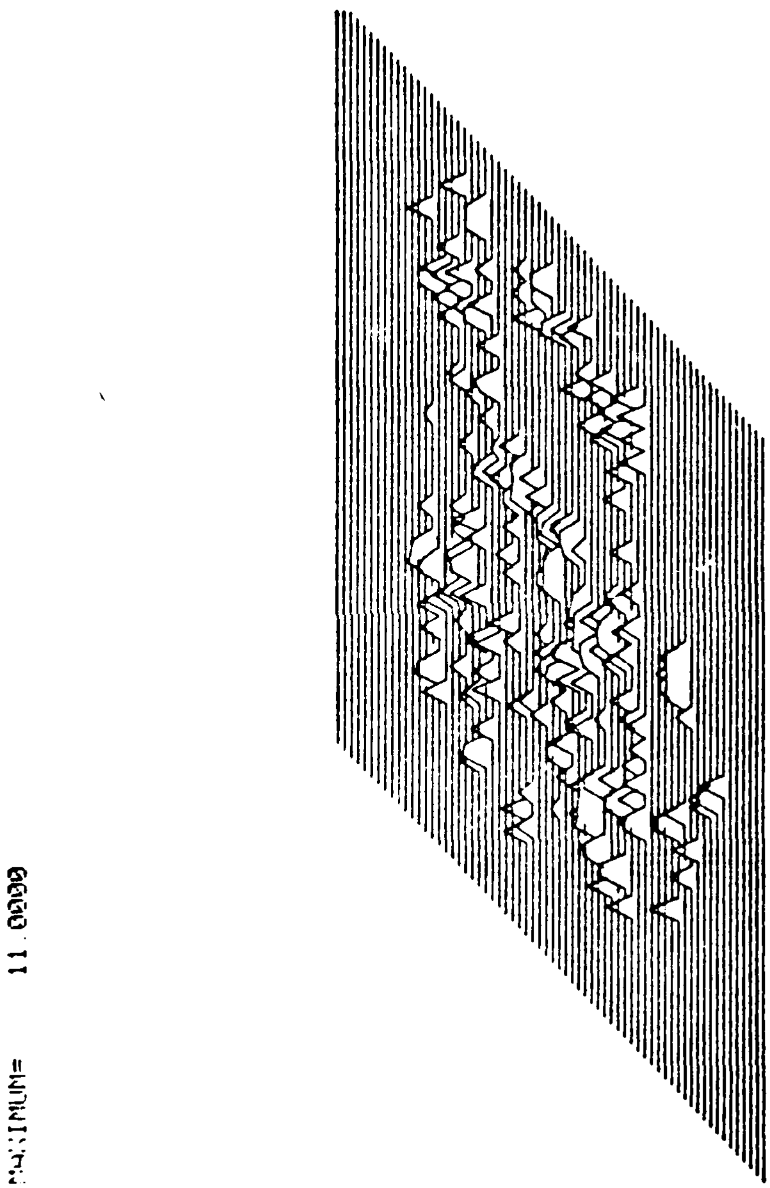

Fig. 211. 3-Dimensional Pixel Plot of Radar, Fig. 81

Appendix C

Additional Boolpass Operators - Results

The number of combinations of low pass filtering techniques and boolean operations which might constitute a Boolpass operator are very large. Appendix A demonstrated two specific combinations which were labelled Boolpass operators 1 and 2. These operators were then applied in a consistent manner to 8 different digitized images so that relative comparisons of the output images were possible.

The results given in this section demonstrate other possible Boolpass operators that were briefly investigated during the research and development of this thesis. These additional Boolpass operators were chosen for inclusion in this appendix based on their potential for possible future testing and refinement. At this time, however, they have not been extensively tested over a large number of images.

Figures 212 thru 216 are variations on Boolpass Operation 1, as previously outlined in Appendix A. The differences in the processing steps for these six images are detailed below. Only the output images from the resulting Boolpass operation are given in this section.

Variations On Boolpass Operation 1

Figure 212:

Original digitized Image - Field (Fig. 53). Image Field is replaced (as input TWO to the AND operation) by 3x3 low pass filtered image of Field (Fig. 120).

Threshold = .2

Figure 213:

Original digitized image - Field (Fig. 53). Image Field is replaced (as input Two to the AND operation) by 7x7 low pass filtered image of Field (Fig. 119).

Threshold = .2

Figure 214:

Original digitized image - Field (Fig. 53). 11x11 low pass filtered image of Field is replaced (as input ONE to the AND operation) by a 7x7 low pass filtered image of Field (Fig. 119)

Threshold = .5

Figure 215:

Original digitized image - Field2 (Fig. 58). 11x11 low pass filtered image of Field2 is replaced (as input ONE to the AND operation) by a 7x7 low pass filtered image of Field2 (Fig. 126). Similarly, image Field2 is replaced (as input Two to the AND operation) by 5x5 low pass filtered image of Field2.

Threshold = .3

Figure 216:

Original digitized image Dispersal (Fig. 73). Image Dispersal is replaced (as input Two to the AND operation) by 7x7 low pass filtered image of Dispersal (Fig. 148).

Threshold = .2

Figure 217:

Original digitized image Truck (Fig. 7). Image Truck is replaced (as input Two to the AND operation) by 7x7 low pass filtered image of Truck.

Threshold = .4

Figure 218:

Original digitzed image Tomcat (Fig. 48). Image Tomcat is replaced (as input Two to the AND operation) by 7x7 low pass filtered image of Tomcat.

Threshold = Ø

Figure 219: Threshold = .2

Figure 220: Threshold = .4

Figure 221:

Original digitized image Dispersal (Fig. 73) 1x5 low pass filtered image of Dispersal is AND'ed with a 1x7 low pass filtered image of Dispersal. Threshold for this first AND operation = .3 . The ouput image of this first AND operation is used as input ONE to a second AND operation, and Dispersal (the original digitized image) is used as input Two to the AND operation. Figure 221 is the ouput of this second and operation performed at a threshold = .3 .

Combined Images Using Boolean OR Operation

Figures 222 and 223 are each composed of a combination of two output images from Boolpass operations. The combining of the two images was performed by a simple boolean OR operation. For the two examples which follow, one image for the OR operation was an ouput image from Boopass operation 1, while the second image for the OR operation was an ouput image from Boolpass operation 2.

Figure 222:

Original digitized image - Field (Fig. 53). Boolpass operation 1 output image, Fig. 213 OR'ed with Boolpass operation 2 ouput image, Fig. 124.

Figure 223:

Original digitized image - Dispersal (Fig. 73). Boolpass operation 1 output image, Fig. 76 OR'ed with Boolpass operation 2 ouput image, Fig. 153.

Boolpass Operation 3

Because Figure 224 uses a more complex procedure for processing, a complete outline of the processing steps is included for this image.

Outline for Boolpass Operator 3

1. Digitized picture processed : Fig. A

   Step 1

   Apply 2-dimensional 3x3 low pass filter to original picture, Fig. A

2. Resulting output image : Fig. B

   Step 2

   Apply 2-dimensional 7x7 low pass filter to original picture, Fig. A

3. Resulting output image : Fig. C

   Step 3

   Perform 'AND' operation using Fig. B as input 'ONE' to the AND operator and Fig. C as input 'TWO' to the AND operator Threshold = Ø, ie. there is no threshold window

4. Resulting output image : Fig. D

   Step 4

   Apply 2-dimensional 5x5 low pass filter to original picture, Fig. A

5. Resulting output image : Fig. E

   Step 5

   Form negative image of Fig. D (ouput image from step 3)

6. Resulting ouput image : Fig. F

   Step 6

   Perform 'AND' operation using Fig. E as input 'ONE' to the AND operator and Fig. F as input 'TWO' to the AND operator

   Threshold = (the Standard Deviation of the pixel values for the entire picture, Fig. A) / 1Ø

7. Resulting ouput image : Fig. G

   NOTE : Pixels stored in output image of AND operation are taken from input 'TWO'

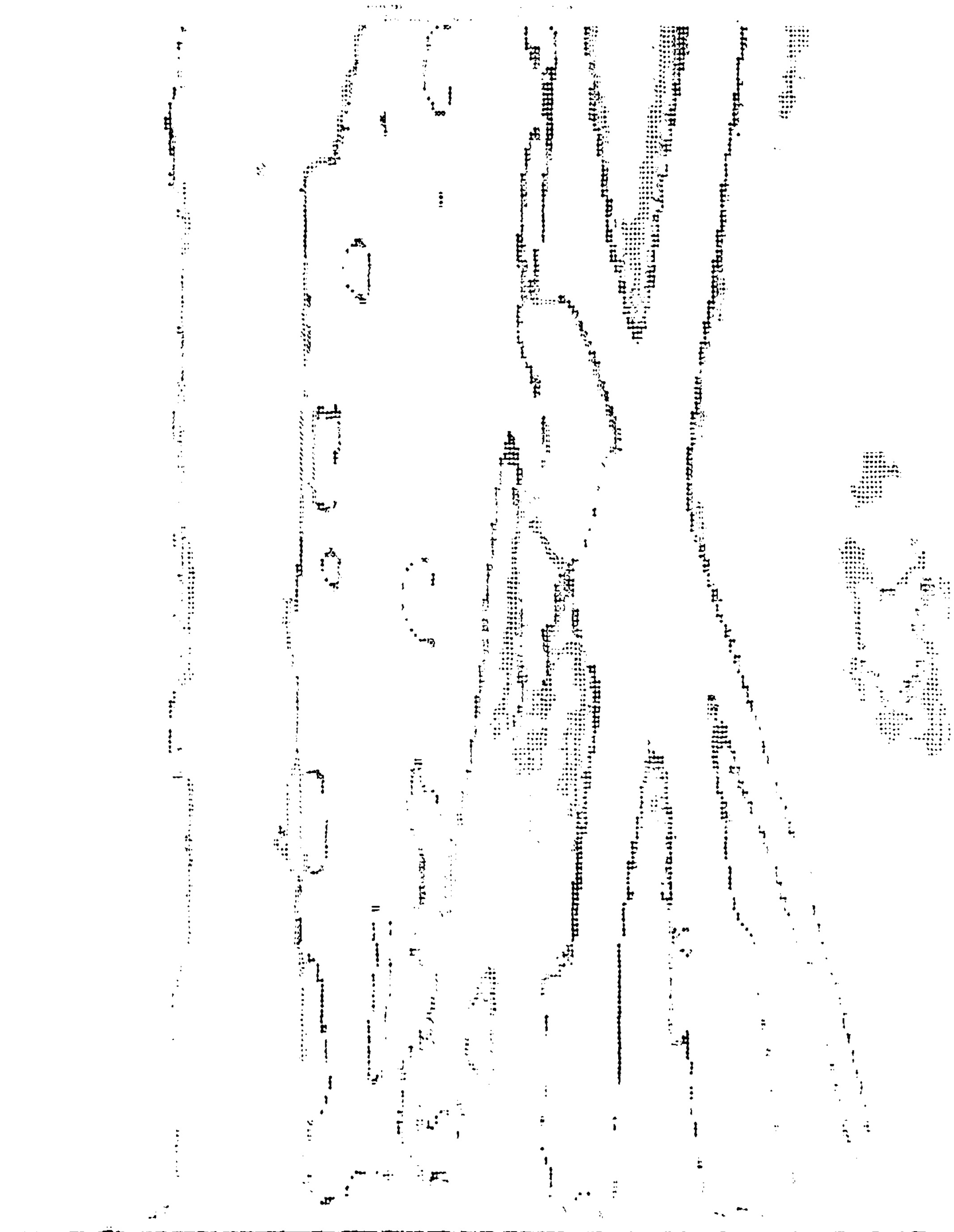

Fig. 212. Variation on Boolpass Operation 1. Image Field is Replaced (as input TWO to the AND operation) by 3x3 Low Pass Filtered Image of Field (Fig. 120). Threshold = .2

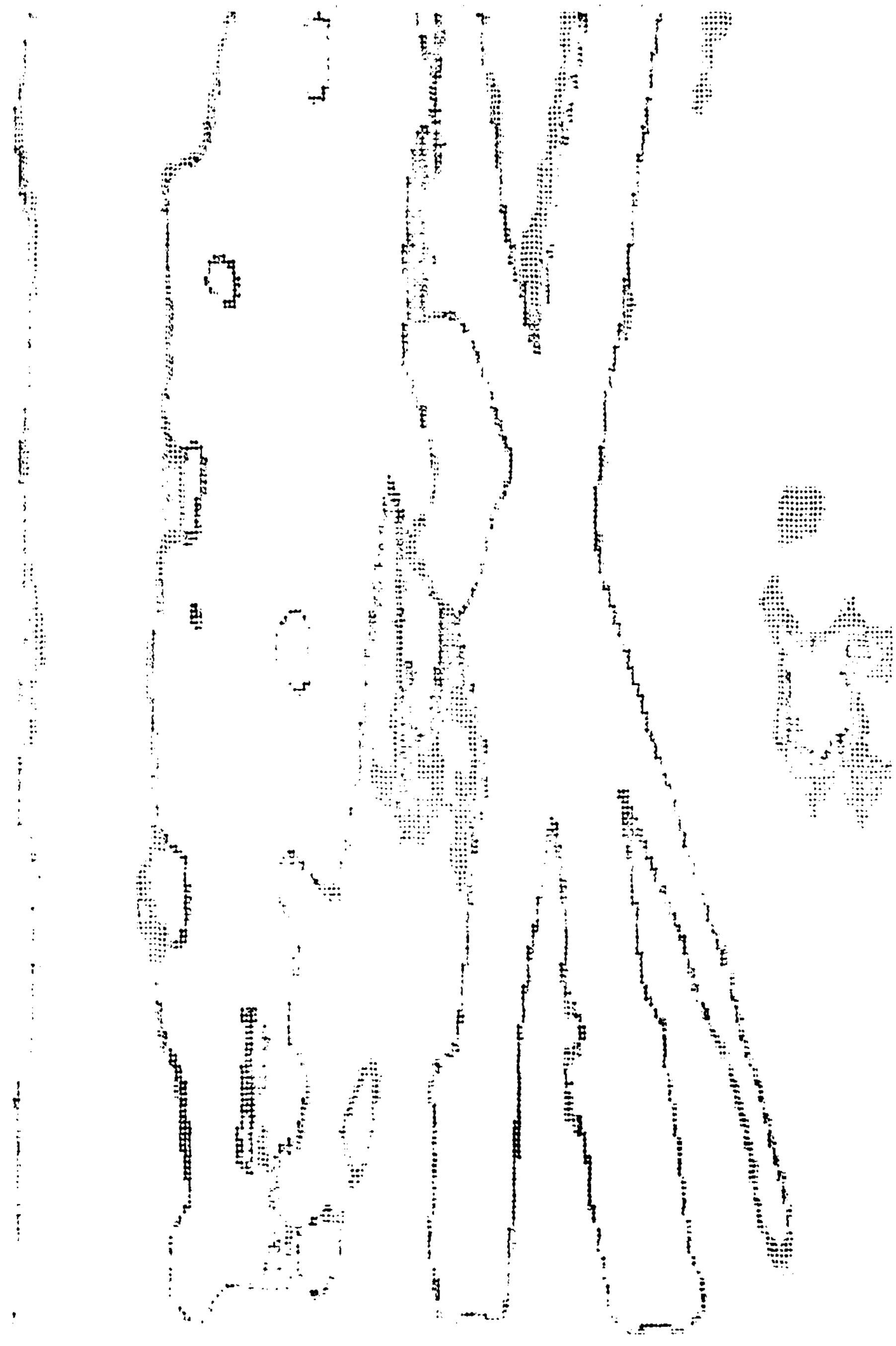

Fig. 213. Variation on Boolpass Operation 1. Image Field is Replaced (as input Two to the AND operation) by 7x7 Low Pass Filtered Image of Field (Fig. 119). Threshold = .2

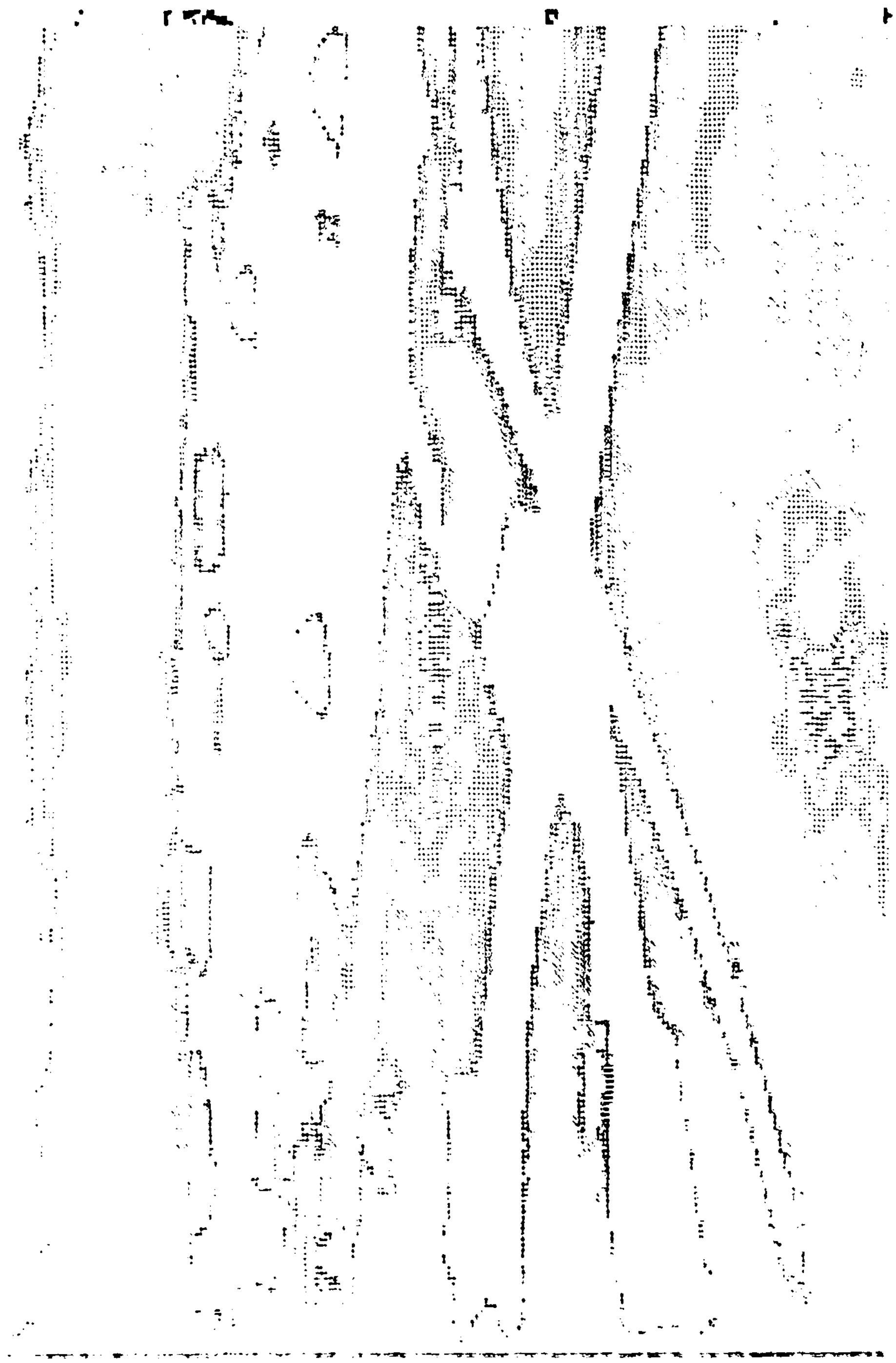

Fig. 214. Variation On Boolpass Operation 1. 11x11 Low Pass Filtered Image of Field is Replaced (as input ONE to the AND operation) by a 7x7 LPF Image of Field (Fig. 119). Threshold = .5

Fig. 215. Variation On Boolpass Operation 1. 11x11 LPF Image of Field2 is Replaced (as input ONE to the AND operation) By a 7x7 LPF Image of Field2 (Fig. 126). Image Field2 is Replaced (as input Two to the AND operation) by 5x5 LPF Image of Field2. Threshold = .3

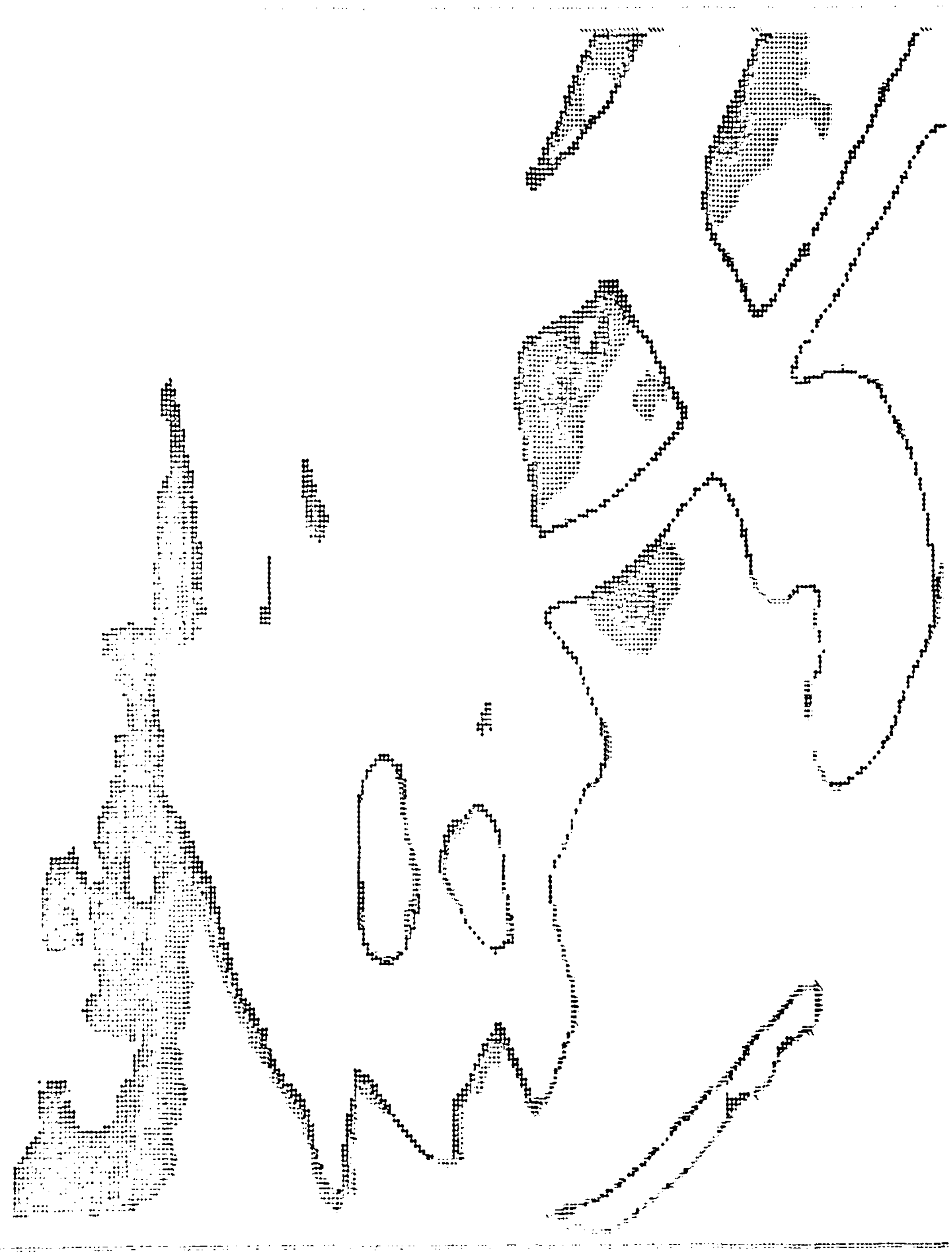

Fig. 216. Variation On Boolpass Operation 1. Image Dispersal Is Replaced (as input Two to the AND operation) By 7x7 Low Pass Filtered Image of Dispersal (Fig. 148). Threshold = .2

Fig. 217. Original digitized image Truck (Fig. 7). Image Truck is replaced (as input Two to the AND operation) by 7x7 low pass filtered image of Truck. Threshold = .4

Fig. 218. Original digitzed image Tomcat (Fig. 48). Image Tomcat is replaced (as input Two to the AND operation) by 7x7 low pass filtered image of Tomcat. Threshold = Ø

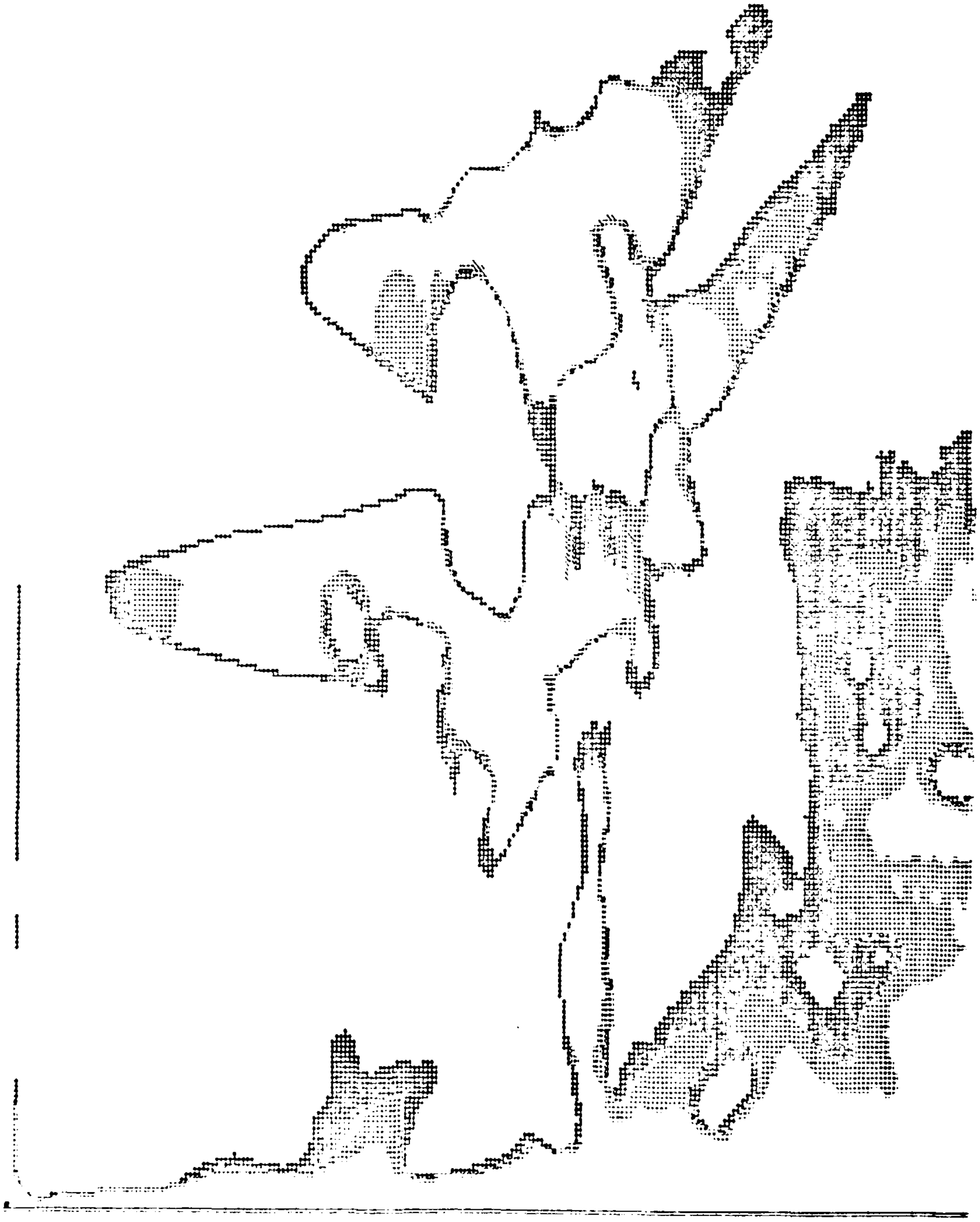

Fig. 219. Same as Fig. 218 except Threshold = .2 .

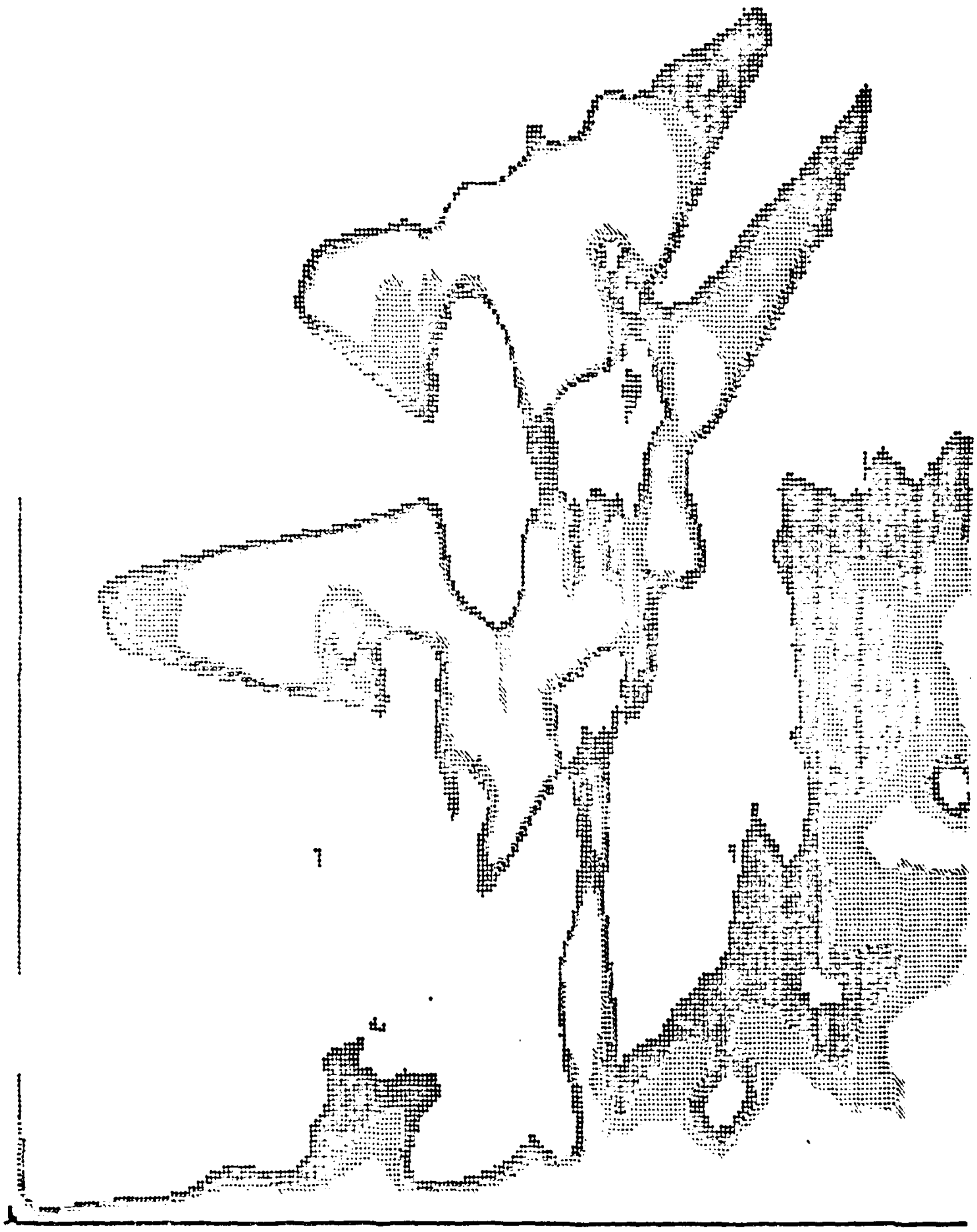

Fig. 220. Same as Fig. 219 except Threshold = .4 .

Fig. 221. Multiple And Operation on Dispersal, Fig. 73.
Threshold = .3

Fig. 222. Original Digitized Image - Field (Fig. 53). Boolpass Operation 1 Output Image, Fig. 213 OR'ed With Boolpass Operation 2 Output Image, Fig. 124.

Fig. 223. Original Digitized Image - Dispersal (Fig. 73). Boolpass Operation 1 Output Image, Fig. 76 OR'ed With Boolpass Operation 2 Output Image, Fig. 153.

Fig. 224. Tank (Fig. 68) Processed By Boolpass Operation 3

Appendix D

Computer Program Listings

The computer programs used for image processing with the Boolpass operators performed two main functions: 1) low pass filtering, and 2) boolean operations. The programs, as presently written, are general in nature and have not been performance optimized to any Boolpass operator in particular. It should also be noted that this RDOS version of Pascal was limited to a maximum stack size of 16K, imposing large disk I/O times for the processing of 64K digitized images.

In order to overcome the limitations imposed by the maximum stack size, the 64K video disk files were divided into four equal files of 16K each for subsequent processing. After processing, these four files were then sequentially appended back into a 64K video file. This latter step was required to be able to display the picture file on the AFIT signal processing laboratory video monitor.

Two Fortran programs were used during the processing of the Boolpass operators. One program was used for dividing the 64K video files into four 16K files. The second Fortran program was used for creating the negative of a video file. These programs already existed as utility programs in the AFIT signal processing laboratory and are not included in this appendix.

```
(****************************************************************************************

PROGRAM NAME   AVGMASKS.PA

WRITTEN BY        CAPTAIN BLAINE FELTMATE

DATE            :  SEP 1981            REVISED 21 MAY 1982

PURPOSE         THE PURPOSE OF THIS PROGRAM, AVGMASKS.PA, IS TO ACT AS A LOWPASS
FILTER IN PROCESSING DIGITIZED VIDEO IMAGES.  THE FILTER IS A SIMPLE AVERAGING
OF THE MASK PIXEL VALUES, WITH THE CENTER PIXEL BEING REPLACED BY
THE 'AVERAGE' VALUE.   THIS PROGRAM CAN WORK WITH A MAXIMUM PICTURE SIZE OF
 256 X 256 PIXEL ELEMENTS, AND A MAXIMUM OF 16 GREYLEVEL VALUES.
        DUE TO THE STACK LIMITATIONS IMPOSED BY THE RDOS VERSION OF
PASCAL, THE 64K PICTURE FILE MUST BE EQUALLY DIVIDED INTO FOUR SEPARATE
FILES NAMED DATA1, DATA2, DATA3, DATA4.   AFTER PROCESSING BY THIS PROGRAM
THE NEW PICTURE WILL BE STORED IN FILES NEWD1, NEWD2, NEWD3, NEWD4.
IN ORDER TO DISPLAY THE PICTURE ON THE MONITOR VIA THE SIGNAL PROCESSING
LAB PROGRAM "VIDEO" THE SEPARATE FILES MUST AGAIN BE RECOMBINED INTO ONE
SINGLE PICTURE FILE.  THIS IS EASILY DONE BY THE FOLLOWING COMMAND
     " APPEND <FILENAME> NEWD1 NEWD2 NEWD3 NEWD4"
TO INITIALLY DIVIDE THE PICTURE FILE INTO FOUR - USE THE FORTRAN PROGRAM
LABELLED "V4".  BY RUNNING V4 YOU WILL BE PROMPTED FOR THE PICTURE
FILENAME, AND ONCE THE FILENAME IS ENTERED EVERYTHING ELSE PROCEEDS AUTOMAT-
ICALLY WITHIN SECONDS.

     AT THIS TIME THE SIZE MASKS ALLOWED FOR THIS PROGRAM ARE 3X3, 5X5, 7X7,
 9X9, 11X11
IF LARGER MASKS SIZES ARE REQUIRED MODIFICATIONS TO THE PROGRAM WILL
BE REQUIRED - ( BUT COULD BE VERY EASILY DONE).

****************************************************************************************)


PROGRAM AVGMASKS(INPUT, OUTPUT, DATA1, DATA2, DATA3, DATA4,
                 NEWD1, NEWD2, NEWD3, NEWD4);


LABEL      82;

CONST
     YESCHAR = 'Y';
     NOCHAR = 'N';

TYPE
     DATAFILE = PACKED RECORD
                LINE : PACKED ARRAY[0..255] OF 0..15
                END;
     DATAF = FILE OF DATAFILE;
     ROWCOL = 0..255;
     PIXEL = 0..15;
     RECROWDATA = PACKED ARRAY[1..11] OF DATAFILE;
     RECPIXM = PACKED RECORD
               PIXMATRIX : PACKED ARRAY[1..121] OF 0..15;
               END;
     RECMATRIXMASK = PACKED ARRAY[0..255] OF RECPIXM;

VAR
     GRECMASK   RECMATRIXMASK;
     MATRIXDATA   RECROWDATA;
     DATAOBJECT   DATAFILE;
     DATA1, DATA2, DATA3, DATA4    DATAF, (*THESE ARE THE INPUT IMAGE FILES TO BE PROCESSED*)
     NEWD1, NEWD2, NEWD3, NEWD4    DATAF, (*THESE ARE THE OUPUT IMAGE FILES WHICH CONTAIN THE PROCESSED IMAGE*)
     GREYLEVEL   PIXEL;
```

```
ROW.COL   ROWCOL,
MAXCOL.MINCOL.MAXROW.MINROW   INTEGER,
ANSWER   CHAR,
LINECOUNT.KOUNT.MASK.NUMBER   INTEGER,
OFFSET.SIZE   INTEGER,
MEAN.STDEVIATION   INTEGER,

(* ********************************************************************************)


(*PROCEDURE BLANKFILE IS DESIGNED TO FILL THE 'OUTPUT' DISK FILES
NEWD1 AND NEWD2 WITH THE LOWEST GREYLEVEL VALUE.I E.
'15'. WHICH IS EQUIVALENT TO A BLANK   THIS ENSURES THAT NO 'GARBAGE'
IS CONTAINED IN THESE FILES BEFORE PICTURE ELEMENT INFORMATION IS STORED*)


PROCEDURE BLANKFILE,

VAR
     NUM,INDEX  0  255,
     N1PICTURE  PACKED ARRAY[0  63] OF DATAFILE,
     N2PICTURE  PACKED ARRAY[64  127] OF DATAFILE,


BEGIN
     REWRITE(NEWD1),
     FOR INDEX  = 0 TO 63 DO
     BEGIN
          WITH N1PICTURE[INDEX] DO
          BEGIN
               FOR NUM  = 0 TO 255 DO
               LINE[NUM]  = 15,
          END,
          NEWD1^  = N1PICTURE[INDEX],
          PUT(NEWD1),
     END,
     CLOSE(NEWD1),
     REWRITE(NEWD2),
     FOR INDEX  = 64 TO 127 DO
     BEGIN
          WITH N2PICTURE[INDEX] DO
          BEGIN
               FOR NUM  = 0 TO 255 DO
               LINE[NUM]  = 15,
          END,
          NEWD2^  =N2PICTURE[INDEX],
          PUT(NEWD2),
     END,
     CLOSE(NEWD2),
END,

(**************************************************************************************)


(*BLANK2FILE IS DESIGNED TO FILL THE OUTPUT DISK FILES NEWD3 AND NEWD4
 WITH THE LOWEST GREYLEVEL VALUE. I E  '15', WHICH IS EQUIVALENT TO A BLANK *)

PROCEDURE BLANK2FILE,

VAR
     NUM.INDEX   0  255,
```

```
     N3PICTURE   PACKED ARRAY[128  191] OF DATAFILE,
     N4PICTURE   PACKED ARRAY[192  255] OF DATAFILE,

BEGIN
     REWRITE(NEWD3),
     FOR INDEX := 128 TO 191 DO
     BEGIN
          WITH N3PICTURE[INDEX] DO
          BEGIN
               FOR NUM := C TO 255 DO
               LINE [NUM] := 15,
          END,
          NEWD3^ := N3PICTURE[INDEX],
          PUT(NEWD3);
     END,
     CLOSE(NEWD3);
     REWRITE(NEWD4);
     FOR INDEX := 192 TO 255 DO
     BEGIN
          WITH N4PICTURE[INDEX] DO
          BEGIN
               FOR NUM := 0 TO 255 DO
               LINE [NUM] := 15,
          END,
          NEWD4^ := N4PICTURE[INDEX],
          PUT(NEWD4);
     END;
     CLOSE(NEWD4),
END (* PROCEDURE BLANKFILE*)

(************************************************************************************)

(*PROCEDURE CHECKLIMITS IS DESIGNED TO CHECK THE ROW AND COLUMN LIMITS
WHICH ARE USED FOR PROCESSING A PICTURE   IF AN ERROR IS DETECTED
.SUCH AS WHEN A  ROW OR COLUMN IS TO BE PROCESSED - LESS THAN 0 OR GREATER THAN
 256, THE LIMITS ARE AUTOMATICALLY CHANGED TO THE MIN/MAX ALLOWABLE   THIS
 PROCEDURE ALSO ADJUSTS THE LIMITS BASED ON THE SIZE OF MASK USED AND
 MIN MAX EDGE BOUNDARIES PERMITTED FOR PROCESSING *)

PROCEDURE CHECKLIMITS (VAR CKROW,CKCOL   ROWCOL);



BEGIN
     IF MASK = 0 THEN SIZE := 0 ELSE
     IF MASK = 3 THEN SIZE := 1 ELSE
     IF MASK = 5 THEN SIZE := 2 ELSE
     IF MASK = 7 THEN SIZE := 3 ELSE
     IF MASK = 9 THEN SIZE := 4 ELSE
     IF MASK = 11 THEN SIZE := 5;

     IF ((CKROW - SIZE) < 0) THEN
     BEGIN
          WHILE  ((CKROW - SIZE ) < 0) DO
          BEGIN
               CKROW := CKROW + 1,
          END,
         WRITELN(' CKROW LIMIT TOO LOW - CHANGED TO  ',CKROW),
          MINROW := CKROW,
     END,

     IF ((CKROW + SIZE) > 255) THEN
     BEGIN
          WHILE ((CKROW + SIZE ) > 255) DO
```

```
          BEGIN
               CKROW  := CKROW - 1;
          END;
     WRITELN(' CKROW LIMIT TOO HIGH - CHANGED TO  ',CKROW);
     MAXROW  := CKROW;
     END;

     IF ((CKCOL - SIZE ) < 0) THEN
     BEGIN
          WHILE ((CKCOL - SIZE) < 0 )  DO
          BEGIN
               CKCOL  := CKCOL + 1;
          END;
          WRITELN(' CKCOL LIMIT TOO LOW - CHANGED TO  ',CKCOL);
          MINCOL  := CKCOL;
     END;

     IF ((CKCOL + SIZE ) > 255 ) THEN
     BEGIN
          WHILE ((CKCOL + SIZE) > 255) DO
          BEGIN
               CKCOL  := CKCOL - 1;
          END;
          WRITELN('CKCOL LIMIT TOO HIGH - CHANGED TO  ',CKCOL);
          MAXCOL  := CKCOL;
     END;


   IF ANSWER = YESCHAR THEN
     BEGIN
       MINROW  := SIZE;
       MAXROW  := 255 - SIZE;
       MINCOL  :=  SIZE;
       MAXCOL  := 255 - SIZE;
     END;

END;

(*****************************************************************************************)


PROCEDURE SETUP;

LABEL
     10,20 ;

VAR
     DUM1,DUM2 : INTEGER;

BEGIN
     DUM1  := 1;
     DUM2  := 1;
     WRITELN;
     WRITELN('DO YOU WANT TO FILL THE OUTPUT DISK FILES -');
     WRITELN(' NEWD1 TO NEWD4 WITH BLANKS?  Y/N ');
     READLN(ANSWER);
     IF ANSWER = YESCHAR THEN
     BEGIN
          WRITELN;
          WRITELN(' THE "WORKING" DISK FILES ARE NOW BEING FILLED WITH BLANKS ');
          BLANKFILE;
          WRITELN;
          WRITELN('            NEWD1,NEWD2 - READY ... ');
          BLANK2FILE;
          WRITELN('                        NEWD3,NEWD4 - READY');
```

```
        WRITELN,
    END,
    10   WRITELN(' ALLOWABLE MASK SIZES FOR PICTURE PROCESSING ARE 3X3,5X5, 7X7,9X9, OR 11X11'),
    WRITE(' ENTER MASK SIZE    "3"  "5"  "7"  "9"  "11"         '),
    READLN(MASK),
    IF (MASK <> 3) AND (MASK <> 5) AND (MASK <> 7) AND (MASK <> 9) AND (MASK <> 11) THEN
    BEGIN
        WRITELN,
        GOTO 10,
    END,
    WRITELN(' MASK SIZE    ',MASK,'    BEING USED'),
    WRITELN,
    20   WRITELN(' DO YOU WANT TO PROCESS ENTIRE PICTURE - Y OR N ?'),
    READLN(ANSWER),
    IF ANSWER = NOCHAR THEN
    BEGIN
        WRITELN,
        WRITELN('          ENTER - " MINROW MAXROW MINCOL MAXCOL " ');
        WRITELN(' MIN/MAX LIMITS ON VALUES ENTERED = 0 THRU 255 INCLUSIVE'),
        WRITELN('            FOR EXAMPLE    2 20 3 125 '),
        READLN(MINROW,MAXROW,MINCOL,MAXCOL),
        CHECKLIMITS(MINROW,MINCOL);
        WRITELN,
        CHECKLIMITS(MAXROW,MAXCOL);
        WRITELN('MINROW   ',MINROW,'        MINCOL   ',MINCOL);
        WRITELN('MAXROW   ',MAXROW,'        MAXCOL   ',MAXCOL),
        WRITELN,
    END

    ELSE

    IF ANSWER = YESCHAR THEN
    BEGIN
        CHECKLIMITS(DUM1,DUM2),
        IF MASK <> 3 THEN
        WRITELN(' IGNORE ERROR MESSAGES ON ROW/COL PARAMETERS');
        WRITELN(' PICTURE WILL PROCESS FROM ROW ',MINROW, 'TO ',MAXROW);
        WRITELN(' AND FROM COL ',MINCOL,' TO ',MAXCOL);
    END

    ELSE

    BEGIN
        WRITELN,WRITELN,WRITELN,WRITELN;
        WRITELN(' INCORRECT ENTRY - REPLY WITH "Y" OR "N" - TRY AGAIN'');
        GOTO 20,
    END,
    WRITELN,
    WRITELN('          *****  SETUP COMPLETE   *****'),
    WRITELN,
END.


(*********************************************************************************)

(* PROCEDURE THRESHOLD IS DESIGNED TO FIND BOTH THE MEAN AND STANDARD DEVIATION
OF ALL THE PIXELS IN A 256 BY 256 PICTURE   AT THIS TIME ALLOWANCE HAS
NOT BEEN MADE FOR DOING SUB-IMAGE AVERAGING. SUCH AS MAY BE SPECIFIED BY
THE LIMITS MIN,MAXROW AND MIN,MAXCOL   BECAUSE OF INACCURACIES ON THE LEFT
 AND BOTTOM BORDERS THE AVERAGE IS COMPUTED FROM 12 TO 256 AND FROM
0 TO 233   *)
```

```
PROCEDURE THRESHOLD;

VAR
     THRFILE : DATAFILE;
     TSQR, TSUM, TNUM, PIXPT : REAL;
     T1FILE, T2FILE, T3FILE, T4FILE : DATAF;
     TAVG, STANDEV : REAL;
     TBLOCK : PACKED ARRAY [0..63] OF DATAFILE;
     THRROW, THRCOL : 0..255;


BEGIN;
     TSQR := 0;      (* EQUALS THE SUM OF THE PIXEL VALUES SQUARED*)
     PIXPT := 0;     (*COUNTS THE # OF PIXELS PROCESSED*)
     TSUM := 0;      (*EQUALS SUM OF ALL PIXEL VALUES*)
          FILETITLE(T1FILE, 'DATA1');
          RESET(T1FILE);
          FOR THRROW := 0 TO 63 DO
          BEGIN
          GET(T1FILE);
          TBLOCK[THRROW] := T1FILE^;
          THRFILE := TBLOCK[THRROW];
          WITH THRFILE DO
          BEGIN
               FOR THRCOL := 13 TO 255 DO
               BEGIN
                    TNUM := LINE[THRCOL];
                    TSUM := TSUM + TNUM;
                    PIXPT := PIXPT + 1;
                    TSQR := TSQR + (TNUM * TNUM);
               END;
          END;
     END;
          CLOSE(T1FILE);
     WRITELN;
     WRITELN('         1/4  DONE');
     FILETITLE(T2FILE, 'DATA2');
     RESET(T2FILE);
     FOR THRROW := 0 TO 63 DO
     BEGIN
          GET(T2FILE);
          TBLOCK[THRROW] := T2FILE^;
          THRFILE := TBLOCK[THRROW];
          WITH THRFILE DO
          BEGIN
               FOR THRCOL := 13 TO 255 DO
               BEGIN
                    TNUM := LINE[THRCOL];
                    TSUM := TSUM + TNUM;
                    PIXPT := PIXPT + 1;
                    TSQR := TSQR + ( TNUM * TNUM);
               END
          END;
     END;
     CLOSE(T2FILE);
     WRITELN('         1/2  DONE');
     FILETITLE(T3FILE, 'DATA3');
     RESET(T3FILE);
     FOR THRROW := 0 TO 63 DO
     BEGIN
          GET(T3FILE);
          TBLOCK[THRROW] := T3FILE^;
          THRFILE := TBLOCK[THRROW];
          WITH THRFILE DO
          BEGIN
               FOR THRCOL := 13 TO 255 DO
```

```
                    BEGIN
                         TNUM  := LINE[THRCOL];
                         TSUM  := TSUM + TNUM;
                         PIXPT  := PIXPT + 1;
                         TSQR  := TSQR + ( TNUM * TNUM );
                    END
               END;
     END;
     CLOSE(T3FILE);
     WRITELN('                3/4  DONE');
     WRITELN;
     FILETITLE(T4FILE, 'DATA4');
     RESET(T4FILE);
     FOR THRROW := 0 TO 40 DO
     BEGIN
          GET(T4FILE);
          TBLOCK[THRROW] := T4FILE^;
          THRFILE := TBLOCK[THRROW];
          WITH THRFILE DO
          BEGIN
               FOR THRCOL := 13 TO 255 DO
               BEGIN
                    TNUM  := LINE[THRCOL];
                    TSUM  := TSUM + TNUM;
                    PIXPT  := PIXPT + 1;
                    TSQR  := TSQR + ( TNUM * TNUM);
               END
          END;
     END;
     CLOSE(T4FILE);
     TAVG  := (TSUM / PIXPT);
     MEAN  := ROUND(TAVG);
     WRITELN(' MEAN PIXELVALUE FOR 256 X 256 PICTURE = ',TSUM,' / ',PIXPT,' = ',MEAN);

(*COMPUTE STANDARD DEVIATION*)

     STANDEV  := SQRT(((PIXPT*TSQR) - (TSUM*TSUM))/(PIXPT*(PIXPT-1)));
     STDEVIATION  := ROUND(STANDEV);
     WRITELN(' STANDARD DEVIATION  =  ',STDEVIATION);
END;

(*****************************************************************************************)

(* PROCEDURE READPIX IS DESIGNED TO RETURN A GREYLEVEL VALUE FOR
EACH PIXEL IN A PICTURE MATRIX OF 256 BY 256 DISCRETE PICTURE
POINTS.  THE PARAMETERS PASSED INTO THE PROCEDURE ARE THE 'X,Y'
COORDINATES OF THE POINT/PIXEL IN THE DIGITIZED PICTURE*)

PROCEDURE READPIX( PIXROW : ROWCOL;
                   VAR ROWDATA : RECROWDATA);

(* ROWDATA IS THE DATA WHICH IS PASSED BACK FROM READPIX.  IT
CONSISTS OF A PACKED RECORD OF PICTURE ROWS - EACH CONTAINING 256
DISCRETE PICTURE GREYLEVEL VALUES.  THE SIZE OF ROWDATA IS COMPLETELY
DEPENDENT ON THE "MASK" SIZE:     EITHER 3X3 - 3ROWS, 5X5 - 5 ROWS,
OR 7X7 - 7 ROWS  *)


VAR
     DONE, DTWO, DTHREE, DFOUR : DATAF;
     RECINDEX : INTEGER;
     NUMPIXROW, TOPROW, BOTROW : INTEGER;

BEGIN
```

```
RECINDEX  := 0;
IF PIXROW < 0 THEN
 WRITELN ('ERROR IN PROCEDURE READPIX - VALUE < 1');
IF PIXROW > 255 THEN
 WRITELN(' ERROR IN PROCEDURE READPIX - VALUE > 255');


IF MASK = 0 THEN OFFSET  := 0 ELSE
IF MASK = 3 THEN OFFSET  := 1 ELSE
IF MASK = 5 THEN OFFSET  := 2 ELSE
IF MASK = 7 THEN OFFSET  := 3 ELSE
IF MASK = 9 THEN OFFSET  := 4 ELSE
IF MASK = 11 THEN OFFSET  := 5;
TOPROW :=( PIXROW - OFFSET);
BOTROW  :=( PIXROW + OFFSET);
SIZE  := OFFSET;
FOR PIXROW := TOPROW TO BOTROW DO
BEGIN

RECINDEX := RECINDEX + 1;
IF (PIXROW >=0) AND (PIXROW <= 63) THEN
BEGIN
      FILETITLE(DONE, 'DATA1');
      RESET(DONE);
      SEEK(DONE,PIXROW);
      GET(DONE);
      ROWDATA[RECINDEX] := DONE^;
      CLOSE(DONE);
END
ELSE
IF(PIXROW >= 64) AND (PIXROW <= 127) THEN
BEGIN
      NUMPIXROW := PIXROW - 64;
      FILETITLE(DTWO, 'DATA2');
      RESET(DTWO);
      SEEK(DTWO,NUMPIXROW);
      GET(DTWO);
      ROWDATA[RECINDEX] := DTWO^;
      CLOSE(DTWO);
END
ELSE

IF(PIXROW >= 128) AND (PIXROW <= 191) THEN
BEGIN
      NUMPIXROW  := PIXROW - 128;
      FILETITLE(DTHREE, 'DATA3');
      RESET(DTHREE);
      SEEK(DTHREE,NUMPIXROW);
      GET(DTHREE);
      ROWDATA[RECINDEX]  := DTHREE^;
      CLOSE(DTHREE);
END

ELSE

IF(PIXROW >= 192) AND (PIXROW <= 255) THEN
BEGIN
      NUMPIXROW  := PIXROW - 192;
      FILETITLE(DFOUR, 'DATA4');
      RESET(DFOUR);
      SEEK(DFOUR,NUMPIXROW);
      GET(DFOUR);
      ROWDATA[RECINDEX]  := DFOUR^;
      CLOSE(DFOUR);
```

```
    END.
  END.
END. (*READPIX*)

(*****************************************************************************)

(*PROCEDURE STORELINE IS DESIGNED TO STORE ONE COMPLETE LINE (OR ROW) OF
A PICTURE (IE 256 PIXELS) INTO A DISK FILE .  THE
DISK FILE IS COMPOSED OF 64 SUCH LINES AND IS OF TYPE DATAF AS
DESCRIBED ABOVE  EACH LINE , ITSELF, IS A PACKED RECORD OF PIXELS OF
 TYPE "DATAFILE" .*)


PROCEDURE STORELINE(ROWSTORE : ROWCOL;
                    DATALINE : DATAFILE);

VAR
    D1FILE,D2FILE,D3FILE,D4FILE : DATAF;
    STROBJECT : DATAFILE;

BEGIN
    IF(ROWSTORE >=0) AND (ROWSTORE <= 63) THEN
    BEGIN
         STROBJECT := DATALINE;
         FILETITLE(D1FILE,'NEWD1');
         RESET(D1FILE);
         SEEK(D1FILE,ROWSTORE);
         WRITE(D1FILE,STROBJECT);
         CLOSE (D1FILE);
    END

    ELSE
    IF(ROWSTORE >= 64) AND (ROWSTORE <= 127) THEN
    BEGIN
         ROWSTORE := ROWSTORE - 64;
         STROBJECT := DATALINE;
         FILETITLE(D2FILE,'NEWD2');
         RESET(D2FILE);
         SEEK(D2FILE,ROWSTORE);
         WRITE(D2FILE,STROBJECT);
         CLOSE(D2FILE);
    END

    ELSE
    IF(ROWSTORE >= 128) AND (ROWSTORE <= 191) THEN
    BEGIN
         ROWSTORE := ROWSTORE - 128;
         STROBJECT := DATALINE;
         FILETITLE(D3FILE,'NEWD3');
         RESET(D3FILE);
         SEEK(D3FILE,ROWSTORE);
         WRITE(D3FILE,STROBJECT);
         CLOSE (D3FILE);
    END

    ELSE
    IF(ROWSTORE >= 192) AND (ROWSTORE <= 255) THEN
    BEGIN
         ROWSTORE := ROWSTORE - 192;
         STROBJECT := DATALINE;
         FILETITLE(D4FILE,'NEWD4');
         RESET(D4FILE);
         SEEK(D4FILE,ROWSTORE);
         WRITE(D4FILE,STROBJECT);
         CLOSE(D4FILE);
         END
```

```
     ELSE
          WRITELN('"ROWSTORE" VALUE PASSED ---OUT OF RANGE IN STORELINE').
END.

(****************************************************************************)

(*PROCEDURE FILLMASK TAKES A SPECIFIC PIXEL IN THE PICTURE AND BUILDS
 A MATRIX OR 'MASK' WITH THE SPECIFIED PI EL AS THE CENTER
MATRIX/MASK SIZE IS ENTERED INTERACTIVELY BY THE USER
ALLOWABLE MASK SIZES ARE  3X3, 5X5,  7X7, 9X9 OR 11X11
THIS PROCEDURE DOES NOT COMPENSATE FOR 'EDGE' PIXEL VALUES*)

PROCEDURE FILLMASK( FROW : ROWCOL;
                    PIXPOINTS : RECROWDATA;
                    VAR FMASK : RECMATRIXMASK);

VAR
     FCOL, ECOL, SCOL : ROWCOL;
     TOPCORNER, BOTCORNER, LEFTCOL, RTCOL, LEFTCORNER : INTEGER;
     FINDEX, FCOUNTER, FNUMBER : INTEGER;
     FILLPIXREC : 0..255;
     FSIZE, MARKER : INTEGER;
     TESTINDEX : 1..121;

BEGIN
     FSIZE := OFFSET;
     FCOL := MINCOL;
     SCOL := MINCOL;
     ECOL := MAXCOL;
     TOPCORNER := FROW - FSIZE;
     BOTCORNER := FROW + FSIZE;
     LEFTCOL := MASK - FSIZE;
     RTCOL := MASK + FSIZE;
     LEFTCORNER := FCOL - SIZE;
     MARKER := LEFTCORNER;

(* THIS PROCEDURE LABELS THE POINTS IN THE MATRIX / MASK
  STARTING AT ONE "1" UP TO THE  (MASK * MASK) LOCATION
  NOTE HOWEVER THAT RECORDS OF THESE MATRIX FILES GO FROM
  0 TO 255 *)

     BEGIN
       FOR FILLPIXREC := SCOL TO ECOL DO
       BEGIN
          FINDEX := 0;
          FCOUNTER := 0;
          FOR FROW := TOPCORNER TO BOTCORNER DO
          BEGIN
               FCOUNTER := FCOUNTER + 1;
               FNUMBER := MARKER;
               FOR FCOL := LEFTCOL TO RTCOL DO
               BEGIN
                    FINDEX := FINDEX + 1;
                    FMASK[FILLPIXREC].PIXMATRIX[FINDEX] := PIXPOINTS[FCOUNTER].LINE[FNUMBER];
                    FNUMBER := FNUMBER + 1;
               END
          END;
          MARKER := MARKER + 1;
       END;
     END;
(*     FOR FILLPIXREC := MINCOL TO MAXCOL DO
     BEGIN
          WRITELN('FMASK USING MASK ',FILLPIXREC,' AND MASK SIZE ',MASK).
```

```
        FOR TESTINDEX  = 1 TO (MASK * MASK) DO
        BEGIN
           WRITE(FMASK[FILLPIXREC] PIXMATRIX[TESTINDEX] 3),
        END.
      WRITELN,
      WRITELN('*********************************************'),
      END, *)
END.


(**************************************************************************************************)


PROCEDURE AVERAGE(AVGMASK RECMATRIXMASK);

VAR
     AVGFILE   DATAFILE,
     AVGREC   0. 255;
     AVERAGE, AVDEX : INTEGER;
     AVGSUM   REAL;
     AVP   PACKED ARRAY[1..121] OF 0..15;
     MASKSQUARED : INTEGER,

BEGIN
     WRITELN('PROCESSING WITH A M SK SIZE OF ', MASK);
     AVERAGE  =0,
     MASKSQUARED := MASK * MASK,
     FOR AVGREC  = MINCOL TO MAXCOL DO
     BEGIN
          AVGSUM  = 0,
          FOR AVDEX  = 1 TO MASKSQUARED DO
          BEGIN
               AVP[AVDEX]  = AVGMASK[AVGREC].PIXMATRIX[AVDEX];
               AVGSUM  = AVGSUM + AVP[AVDEX];
          END.
          AVERAGE  = ROUND(AVGSUM/MASKSQUARED);
          AVGFILE.LINE[AVGREC]  = AVERAGE,
     END,
     STORELINE(ROW, AVGFILE),
     AVERAGE  = 0;
END,
(**************************************************************************************************)

BEGIN (* MAINLINE*)
     LINECOUNT := 0;
     WRITELN, WRITELN, WRITELN, WRITELN; WRITELN, WRITELN, WRITELN,
     WRITELN(' WELCOME TO PROCESS PICTURE - YOU ARE CURRENTLY USING THE PASCAL PROGRAM.....');
     WRITELN;
     WRITELN('          "AVGMASKS"      (VERSION MAY 1982)');
     WRITELN;
     WRITELN,
     WRITELN('NOTE   FOR THIS PROGRAM TO WORK A PICTURE FOR PROCESSING')
     WRITELN('MUST ALREADY EXIST ON DISK IN FILES SPECIFICALY LABELLED - DATA1, ');
     WRITELN(' DATA2, DATA3 AND DATA4'),
     WRITELN('TO CREATE THESE FILES SIMPLY ENVOKE THE FORTRAN PROGRAM "V4" BY TYPING "V4"');
     WRITELN('AT THE TERMINAL. YOU MUST EXIT THIS PROGRAM FIRST, HOWEVER, SO');
     WRITELN(' TYPE "CONTROL A", AND REENTER THIS PROGRAM AFTER DATA1 THRU 4 ARE CREATED');
     WRITELN, WRITELN, WRITELN, WRITELN, WRITELN,
     62 WRITELN('BESIDES PERFORMING LOW PASS (AVERAGING) FILTERING OPERATIONS '),
     WRITELN('THIS PROGRAM CAN CALCULATE THE MEAN AND STANDARD DEVIATION');
     WRITELN('FOR THE PIXEL VALUES OF THE ENTIRE PICTURE.  DO YOU WISH TO'),
     WRITE('HAVE THESE VALUES CALCULATED?    Y/N       '),
```

```
      READLN(ANSWER);
      IF ANSWER = YESCHAR THEN
      BEGIN
            WRITELN;
            WRITELN('  ROGER, USER ... WAIT ONE ');
            THRESHOLD;
            WRITELN;WRITELN;WRITELN;WRITELN;WRITELN;;
      WRITELN('  PHEW *#@' ... - I HOPE  I DO NOT HAVE TO DO THAT AGAIN ''');
      WRITELN;WRITELN;WRITELN;WRITELN;WRITELN;
      END
      ELSE
       IF ANSWER <> NOCHAR THEN GOTO 82;
      WRITELN;
      SETUP;
      FOR ROW := MINROW TO MAXROW DO
      BEGIN
            WRITE('ENTERING READPIX - ROW = ',ROW);
            READPIX(ROW,MATRIXDATA);
            WRITELN('       ENTERING FILLMASK - ROW = ',ROW);
            FILLMASK(ROW,MATRIXDATA,GRECMASK);
            AVERAGE(GRECMASK);
            WRITELN('LINECOUNT -   ',LINECOUNT);
            LINECOUNT := LINECOUNT +1;
      END;
      WRITELN;WRITELN;
      WRITELN('            PROCESSING OF PROGRAM AVGMASKS  COMPLETE  ');
      WRITELN;WRITELN;
      WRITELN('                 THANK YOU, HAVE A GOOD DAY''');
      WRITELN;WRITELN;
END
```

```
(*****************************************************************************

PROGRAM NAME   PROCESS PICTURE  "ANDOR PA"

BY             : CAPT BLAINE FELTMATE

DATE           : SEP 1981   REVISED 21 MAY 1982

PURPOSE        : THE PURPOSE OF THIS PROGRAM, ANDOR PA, IS TO PERFORM BOOLEAN
"AND" AND "OR" OPERATIONS ON TWO INPUT IMAGE FILES AND CREATE A THIRD
OUTPUT IMAGE FILE OF THE RESULTING PROCESSED PICTURES.  THE FIRST INPUT IMAGE
FILE TO THE ANDOR PROGRAM, SIMPLY CALLED IMAGE ONE,
IS CREATED BY RUNNING THE FORTRAN PROGRAM VPIC BY TYPING "VPIC" AT THE
TERMINAL. THIS MUST BE DONE BEFORE ENTERING THIS PROGRAM  VPIC TAKES A
64K PICTURE FILE AND DIVIDES IT EVENLY INTO FOUR 16K FILES WHICH
CAN BE USED FOR PROCESSING BY THIS PROGRAM  THIS IS DUE TO THE 16K STACK
 LIMITATION IMPOSED BY THE RDOS VERSION
OF PASCAL RUNNING ON THE AFIT SIGNAL PROCESSING LABORATORY COMPUTER.
VPIC THEN CREATES FILES PIC1 THRU PIC4 ON DISK.  THE SECOND INPUT
IMAGE, IMAGE TWO, IS PREPARED FOR PROCESSING IN A SIMILAR MANNER.
TO CREATE THE 4 DISK FILES FOR IMAGE TWO SIMPLY TYPE "VD" AND FILES
D1 THRU D4 WILL BE CREATED   AGAIN THIS SHOULD HAVE BEEN DONE PRIOR TO
ENTERING THIS PROGRAM.
     THE OUTPUT OF THE ANDOR PROGRAM, EITHER "AND" OR "OR"
OPERATION SPECIFIED BY THE USER, WILL BE STORED IN FILES N1, N2, N3 AND
N4 ON DISK.  SIMPLY TYPE "APPEND <FILENAME> N1 N2 N3 N4"
AT THE TERMINAL TO REFORM THE 4 PROCESSING FILES BACK INTO A SINGLE
PICTURE FILE.  ALL OTHER PROMPTS SHOULD BE SELF EXPLAINATORY AS THEY
APPEAR INTERACTIVELY TO THE USER.
NOTE : THE VALUES STORED IN THE OUTPUT FILES ARE TAKEN FROM
THE INPUT IMAGE TWO FILES (VD - D1 D2 D3 D4).


*****************************************************************************)


PROGRAM PIXVALUE(INPUT,OUTPUT,DATA1,DATA2,DATA3,DATA4,
                 D1,D2,D3,D4,PIC1,PIC2,PIC3,PIC4,
                  N1,N2,N3,N4);


LABEL
     1111;

CONST
     YESCHAR = 'Y';
     NOCHAR = 'N';
     ANDCHAR = 'A';
     ORCHAR = 'O';

TYPE
     DATAFILE = PACKED RECORD
                LINE : PACKED ARRAY[0..255] OF 0..15
                END;
     DATAF = FILE OF DATAFILE;
     ROWCOL = 0..255;

VAR
     DATA1,DATA2,DATA3,DATA4 : DATAF;
     N1,N2,N3,N4,D1,D2,D3,D4,PIC1,PIC2,PIC3,PIC4 : DATAF;
     ANSWER : CHAR;
     MEAN,STDEVIATION : INTEGER;
     ANDTYPE : BOOLEAN;
     KONST : REAL;
```

```
(*****************************************************************************************************

(* PROCEDURE THRESHOLD IS DESIGNED TO FIND BOTH THE MEAN AND STANDARD DEVIATION
OF ALL THE PIXELS IN A 256 BY 256 PICTURE   AT THIS TIME ALLOWANCE HAS
NOT BEEN MADE FOR DOING SUB-IMAGE AVERAGING, SUCH AS MAY BE SPECIFIED BY
THE LIMITS MIN,MAXROW AND MIN,MAXCOL.  BECAUSE OF INACCURACIES ON THE LEFT
 AND BOTTOM BORDERS THE AVERAGE IS COMPUTED FROM 12 TO 256 AND FROM
0 TO 233   *)


PROCEDURE THRESHOLD;


VAR
     THRFILE : DATAFILE;
     PIXPT, TNUM, TSUM, TSQR : REAL;  (*REQUIRED DUE TO SIZE OF NUMBERS WHICH ACCUMULATE*)
     T1FILE, T2FILE, T3FILE, T4FILE : DATAF;
     TAVG, STANDEV : REAL;
     TBLOCK : PACKED ARRAY [0..63] OF DATAFILE;
     THRROW, THRCOL : 0..255;

BEGIN;
     TSQR := 0;      (* EQUALS THE SUM OF THE PIXEL VALUES SQUARED*)
     PIXPT := 0;      (*COUNTS THE # OF PIXELS PROCESSED*)
     TSUM := 0;      (*EQUALS SUM OF ALL PIXEL VALUES*)
          FILETITLE(T1FILE, 'DATA1');
          RESET(T1FILE);
          FOR THRROW := 0 TO 63 DO
          BEGIN
          GET(T1FILE);
          TBLOCK[THRROW] := T1FILE^;
          THRFILE := TBLOCK[THRROW];
          WITH THRFILE DO
          BEGIN
               FOR THRCOL := 13 TO 255 DO
               BEGIN
                    TNUM := LINE[THRCOL];
                    TSUM := TSUM + TNUM;
                    PIXPT := PIXPT + 1;
                    TSQR := TSQR + (TNUM * TNUM);
               END;
         END;
     END;
          CLOSE(T1FILE);
     WRITELN('          1/4  DONE');
     FILETITLE(T2FILE, 'DATA2');
     RESET(T2FILE);
     FOR THRROW := 0 TO 63 DO
     BEGIN
          GET(T2FILE);
          TBLOCK[THRROW] := T2FILE^;
          THRFILE := TBLOCK[THRROW];
          WITH THRFILE DO
          BEGIN
               FOR THRCOL := 13 TO 255 DO
               BEGIN
                    TNUM := LINE[THRCOL];
                    TSUM := TSUM + TNUM;
                    PIXPT := PIXPT + 1;
                    TSQR := TSQR + ( TNUM * TNUM);
               END
          END;
     END;
```

```
     CLOSE(T2FILE);
     WRITELN('                    1/2  DONE');
     FILETITLE(T3FILE, 'DATA3');
     RESET(T3FILE);
     FOR THRROW := 0 TO 63 DO
     BEGIN
          GET(T3FILE);
          TBLOCK[THRROW] := T3FILE^;
          THRFILE := TBLOCK[THRROW];
          WITH THRFILE DO
          BEGIN
               FOR THRCOL := 13 TO 255 DO
               BEGIN
                    TNUM := LINE[THRCOL];
                    TSUM := TSUM + TNUM;
                    PIXPT := PIXPT + 1;
                    TSQR := TSQR + ( TNUM * TNUM );
               END
           END;
     END;
     CLOSE(T3FILE);
     WRITELN('                         3/4  DONE');
     FILETITLE(T4FILE, 'DATA4');
     RESET(T4FILE);
     FOR THRROW := 0 TO 40 DO
     BEGIN
          GET(T4FILE);
          TBLOCK[THRROW] := T4FILE^;
          THRFILE := TBLOCK[THRROW];
          WITH THRFILE DO
          BEGIN
               FOR THRCOL := 13 TO 255 DO
               BEGIN
                    TNUM := LINE[THRCOL];
                    TSUM := TSUM + TNUM;
                    PIXPT := PIXPT + 1;
                    TSQR := TSQR + ( TNUM * TNUM);
               END
          END;
     END;
     CLOSE(T4FILE);
     WRITELN('                              4/4  DONE');
     TAVG := (TSUM / PIXPT);
     MEAN := ROUND(TAVG);
     WRITELN(' MEAN PIXELVALUE FOR 256 X 256 PICTURE = ',TSUM,' / ',PIXPT,' = ',MEAN);

(*COMPUTE STANDARD DEVIATION*)

     STANDEV := SQRT(((PIXPT*TSQR) - (TSUM*TSUM))/(PIXPT*(PIXPT-1)));
     STDEVIATION := ROUND(STANDEV);
     WRITELN('  STANDARD DEVIATION := ',STDEVIATION);
     WRITELN;
END;

(**************************************************************************************************)


(*PROCEDURE MATCHPIC DOES THE ACTUAL BOOLEAN "AND" AND "OR" OPERATIONS
THE CHOICE OF WHICH OPERATION IS PERFORMED IS ENTERED AT THE TERMINAL
BY THE USER, AFTER AN APPROPRIATE PROMPT REQUESTING THIS INFORMATION
IS GIVEN BY THE COMPUTER *)


PROCEDURE MATCHPIC;

LABEL  767;
```

```
VAR
     STAT, PCOUNTER, DNUM, PNUM, DUL, DLL    INTEGER;
     PROW, PCOL    0  255;
     DBLOCK, PICBLOCK    PACKED ARRAY[0  63] OF DATAFILE;
     NFILE, PFILE, DFILE    DATAFILE;
     FNAME, ORIGINAL, PICNAME  :  DATAF;

BEGIN
     767 WRITELN('   ENTER THE CORRESPONDING LETTER FOR THE OPERATION TO BE PERFORMED  -  ');
     WRITELN;
     WRITELN('            " A " FOR THE * AND OPERATION * ');
     WRITELN('            " O " FOR THE * OR  OPERATION *');
     READLN(ANSWER);
     IF (ANSWER <> ANDCHAR) AND (ANSWER <> ORCHAR) THEN GOTO 767;
     IF ANSWER = ANDCHAR THEN
     BEGIN
          WRITELN; WRITELN;
          WRITELN('YOU CHOICE OF THE AND OPERATOR REQUIRES THAT YOU ');
          WRITELN('ENTER A DECIMAL NUMBER FROM 0 TO 9 OF THE');
          WRITELN('STANDARD DEVIATION TO BE USED EG.   .7 OR .3');
          WRITELN('  THE CHOICE OF 0 WILL INITIATE AN EXACT AND OPERATION WITH NO THRESHOLD WINDOW');
          READLN(KONST);
          STAT := ROUND(KONST * STDEVIATION );
     END
     ELSE
     BEGIN
          WRITELN; WRITELN;
          WRITELN('YOUR SELECTION - THE OR OPERATOR -  IS NOW EXECUTING');
     END;
     FILETITLE(FNAME, 'N1');
     FILETITLE(PICNAME, 'PIC1');
     FILETITLE(ORIGINAL, 'D1');
     PCOUNTER := 1;
     WHILE PCOUNTER <= 4 DO
     BEGIN
          IF PCOUNTER = 1 THEN
          WRITELN('PROCESSING FILEBLOCK 1');
          RESET(PICNAME);
          RESET(ORIGINAL);
          FOR PROW := 0 TO 63 DO
          BEGIN
               GET(PICNAME);
               PICBLOCK[PROW] := PICNAME^;
               GET(ORIGINAL);
               DBLOCK[PROW] := ORIGINAL^;
               PFILE  := PICBLOCK[PROW];
               DFILE  := DBLOCK[PROW];
               WITH PFILE DO
               BEGIN
                 FOR PCOL := 13 TO 255 DO
                 BEGIN
                    IF ANSWER = ANDCHAR THEN
                    BEGIN
                         WITH DFILE DO
                         BEGIN
                              DNUM  := LINE[PCOL];
                              DUL  := DNUM + STAT;
                              DLL  := DNUM - STAT;
                         END;
                         PNUM  := LINE[PCOL];
                         IF ((DLL <= PNUM) AND (PNUM <= DUL)) THEN
                         BEGIN
                              IF ANDTYPE = TRUE THEN
                              BEGIN
```

```
                        NFILE LINE[PCOL] := 0;
                    END
                    ELSE
                    BEGIN
                        NFILE LINE[PCOL] := PNUM;
                    END
                END
                ELSE
                BEGIN
                    NFILE LINE[PCOL] := 15;
                END
            END   (* ANSWER = ANDCHAR*)
            ELSE
            BEGIN
                WITH DFILE DO
                BEGIN
                    DNUM := LINE[PCOL];
                END;
                PNUM := LINE[PCOL];
                IF (PNUM = 15) AND (DNUM = 15) THEN
                BEGIN
                    NFILE LINE[PCOL] := 15;
                  END
                  ELSE
                  BEGIN
                        IF DNUM < PNUM THEN
                        BEGIN
                           NFILE LINE[PCOL] := DNUM;
                        END
                        ELSE
                        BEGIN
                           NFILE LINE[PCOL] := PNUM;
                        END
                    END;
                END;
              END;
            END;
        RESET(FNAME);
        SEEK(FNAME,PROW);
        WRITE(FNAME,NFILE);
        CLOSE(FNAME);
END;
        CLOSE(PICNAME);
        CLOSE(ORIGINAL);
IF PCOUNTER = 1 THEN
BEGIN
         FILETITLE(FNAME, 'N2');
          FILETITLE(PICNAME, 'PIC2');
          FILETITLE(ORIGINAL, 'D2');
  WRITELN('FILEBLOCK 2 NOW BEING PROCESSED');
END
ELSE
IF PCOUNTER = 2 THEN
BEGIN
        FILETITLE(FNAME, 'N3');
         FILETITLE(PICNAME, 'PIC3');
         FILETITLE(ORIGINAL, 'D3');
        WRITELN('FILEBLOCK 3 NOW BEING PROCESSED');
END
ELSE
IF PCOUNTER = 3 THEN
BEGIN
        FILETITLE(FNAME, 'N4');
      FILETITLE(PICNAME, 'PIC4');
      FILETITLE(ORIGINAL, 'D4');
        WRITELN('FILEBLOCK 4 NOW BEING PROCESSED');
```

```
            END;
            PCOUNTER  = PCOUNTER + 1.
        END;
END.

(*****************************************************************************************************)


BEGIN (* MAINLINE*)
     WRITELN.WRITELN.WRITELN.WRITELN.WRITELN.WRITELN.WRITELN.WRITELN.
WRITELN.WRITELN.WRITELN.
     WRITELN('WELCOME TO PROCESS PICTURE - YOU ARE CURRENTLY USING THE PASCAL PROGRAM .     ').
     WRITELN.
     WRITELN('                "ANDOR PA"      (VERSION MAY 1982)').
     WRITELN.WRITELN;
     WRITELN('ENSURE ALL PICTURE FILES ARE PROPERLY SETUP');
     WRITELN('ON DISK BEFORE PROCESSING WITH THIS PROGRAM.  IE. FILES MUST EXIST');
     WRITELN('FOR PIC1 THRU PIC4 AND D1 THRU D4.  (USE FORTRAN PROGRAMS - ').
     WRITELN('"VPIC" AND "VD" TO CREATE THESE FILES IF YOU HAVE NOT');
     WRITELN('ALREADY DONE SO).  THE PICTURE FOR WHICH THE PICTURE');
     WRITELN('MEAN AND STANDARD DEVIATION WILL BE COMPUTED , IF DESIRED BY THE USER, MUST');
     WRITELN('EXIST IN FILES DATA1 THRU 4 (VIA "V4" FORTRAN PROGRAM)');
     WRITELN.WRITELN;
     1111  WRITELN('BESIDES PERFORMING "AND" AND "OR" OPERATIONS').
     WRITELN('THIS PROGRAM CAN ALSO CALCULATE THE MEAN AND STANDARD DEVIATION').
     WRITELN('FOR THE PIXEL VALUES OF THE ENTIRE PICTURE.  DO YOU WISH TO').
     WRITELN('HAVE THESE VALUES CALCULATED?   Y/N    ').
     READLN(ANSWER);
     IF ANSWER = YESCHAR THEN
     BEGIN
          WRITELN, WRITELN('        ROGER, USER ... WAIT ONE, PLEASE').
          WRITELN.WRITELN;
          THRESHOLD;
          WRITELN.WRITELN.WRITELN.WRITELN;
          WRITELN('    YUK!!!      WHAT A PAIN IN THE #@$% THAT TURNED OUT TO BE'"').
     WRITELN.WRITELN.WRITELN.WRITELN;
          END
          ELSE
          IF ANSWER = NOCHAR THEN
          BEGIN
               WRITELN('ENTER MEAN AND STANDARD DEVIATION').
               READLN(MEAN, STDEVIATION).
               WRITELN('MEAN = ',MEAN,'          STANDARD DEVIATION = ',STDEVIATION).
          END
          ELSE
          BEGIN
              WR;TELN(' ERROR !  RE-ENTER VALUES');
              GOTO 1111.
          END.
     WRITELN.WRITELN.WRITELN;
     WRITELN('DO YOU WANT THE OUTPUT OF THE "AND" OPERATION TO BE').
     WRITELN('BINARY - IE GREYLEVEL 0 & 15 ONLY - (RECOMMEND "N")     Y/N');
     READLN(ANSWER);
     IF ANSWER = YESCHAR THEN
     ANDTYPE .= TRUE
     ELSE
     ANDTYPE  = FALSE.
     MATCHPIC.
WRITELN.WRITELN.WRITELN.
WRITELN('  PROCESSING OF ANDOR IS  COMPLETE  ').
WRITELN.
WRITELN('PARAMETERS USED FOR PROCESSING WERE AS FOLLOWS ');
WRITELN.WRITELN.
WRITELN('PICTURE MEAN = ',      MEAN);
WRITELN('PICTURE STANDARD DEVIATION = ',      STDEVIATION).
IF ANDTYPE = TRUE THEN

WRITELN('OUTPUT IMAGE VALUES WERE SELECTED TO BE BINARY. IE 0 AND 15 ONLY').
IF ANDTYPE = FALSE THEN
WRITELN('OUPUT IMAGE VALUES ARE THOSE OF PICTURE STORED IN D1 THRU D4');
WRITELN('DECIMAL NUMBER USED FOR THRESHOLD WINDOW = '.      KONST).
END
```

Vita

Blaine Feltmate was born on 16 December 1953 in Glace Bay, Nova Scotia, Canada. He graduated from Donkin - Morien District High School in June of 1972 and was selected to attend the Royal Military College of Canada (RMC) in August of the same year. In May of 1976 he graduated from RMC with a Bachelor of Engineering degree, Second Class Honours, and was commissioned as a First Lieutenant in the Canadian Armed Forces. After completion of his AERE (Aeronautical and Aerospace Engineering) classification training, he was assigned to 1 Canadian Air Group Maintenance Squadron, Baden, West Germany in January 1977.

While part of the NATO forces in Europe, he served as Line Servicing Officer for 421 Tactical Fighter Squadron and later became the 1 CAG Maintenace Squadron Avionics Officer. He was promoted to the rank of Captain in May of 1979 and in July 1979 received the Chief of the Defence Staff Commendation. He was assigned to the United States Air Force Institute of Technology in August 1980.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFIT/GCS/EE/82J-8
2. GOVT ACCESSION NO.: AD-A118 076
3. RECIPIENT'S CATALOG NUMBER:
4. TITLE (and Subtitle): COMBINED SPATIAL FILTERING AND BOOLEAN OPERATORS APPLIED TO THE PROCESSING OF REAL IMAGES
5. TYPE OF REPORT & PERIOD COVERED: MS Thesis
6. PERFORMING ORG. REPORT NUMBER:
7. AUTHOR(s): Blaine Feltmate, Capt Canadian AFB
8. CONTRACT OR GRANT NUMBER(s):
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Air Force Institute of Technology (AFIT/EN) Wright-Patterson AFB OH 45433
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS:
11. CONTROLLING OFFICE NAME AND ADDRESS:
12. REPORT DATE: June 1982
13. NUMBER OF PAGES: 343
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office):
15. SECURITY CLASS. (of this report): UNCLASSIFIED
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:
16. DISTRIBUTION STATEMENT (of this Report): Approved for public release; distribution unlimited
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report):
18. SUPPLEMENTARY NOTES: APPROVED FOR PUBLIC RELEASE; IAW AFR 190-17 23 JUL 1982

Lynn E. Wolaver

LYNN E. WOLAVER
Dean for Research and Professional Development

AIR FORCE INSTITUTE OF TECHNOLOGY (ATC)
WRIGHT-PATTERSON AFB, OH 45433

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Pattern Recognition	Artificial Intelligence
Image Processing	Spatial Filtering
Boolpass Operators	Picture Processing
Scene Analysis	Edge Operators

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

In this thesis, several new and seemingly successful scene analysis techniques, for application to 'real' image processing are presented and discussed. These techniques consist of particular combinations of spatial low pass filtering, global thresholding and Boolean operators, specifically the 'AND', 'OR' and 'NOT' operators. These combinatorial operators, hereafter referred to as the "Boolpass" operators, perform the task of picture energy/information reduction, while retaining the

DD FORM 1473, 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Block 20. ABSTRACT

fundamental picture primitives such as edges which characterize the images.

Over 150 figures are included which illustrate the results obtained from application of the Boolpass technique to 8 different natural scenes. These results indicate that the Boolpass operators do display great potential as important components of a larger more comprehensive pattern recognition machine. Such a machine would encompass further processing (for target classification/recognition) of the resulting Boolpass operator information.