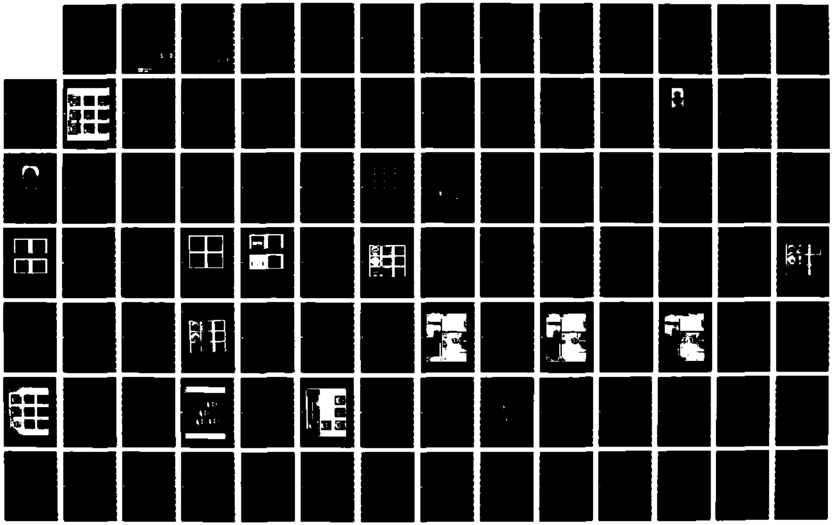
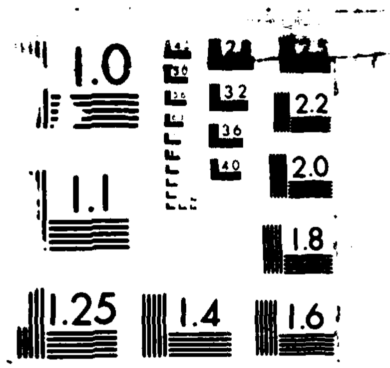
AD-A170 852 DEVELOPMENT OF AN AUTONOMOUS FACE RECOGNITION MACHINE (U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING E J SMITH 08 DEC 86 1/3

UNCLASSIFIED AFIT/GE/ENG/86D-36 F/G 14/5 NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
3.2
3.6
4.0

DEVELOPMENT OF AN AUTONOMOUS

FACE RECOGNITION MACHINE

THESIS

Edward J. Smith
Captain, USAF

AFIT/GE/ENG/86D-36



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

Wright-Patterson Air Force Base, Ohio

[redacted]

AFIT/GE/ENG/86D-36

# DEVELOPMENT OF AN AUTONOMOUS FACE RECOGNITION MACHINE

THESIS

Edward J. Smith
Captain, USAF

AFIT/GE/ENG/86D-36



Approved for public release; distribution unlimited

AFIT/GE/ENG/86D-36

# DEVELOPMENT OF AN AUTONOMOUS FACE RECOGNITION MACHINE

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Electrical Engineering

QUALITY INSPECTED 1

| Accession For | |
|---|---|
| NTIS GRA&I | ☑ |
| DTIC TAB | ☐ |
| Unannounced | ☐ |
| Justification | |
| By | |
| Distribution/ | |
| Availability Codes | |
| Dist | Avail and/or Special |
| A-1 | |


Edward J. Smith, B.S.

Captain, USAF


December 1986

# Acknowledgements

First and foremost I would like to express my sincere thanks to my thesis advisor, Dr. Matthew Kabrisky. From the outset, his very special ability to instill knowledge, confidence and enthusiasm has had a profound effect on this thesis effort. I would also like to extend a special thanks to Captain Robert L. Russel Jr. who introduced me to this topic and spent many hours acquainting me with the intricacies of the Face Recognition Machine.

I am very grateful to Dr. George Batten Jr., of the University of Houston, for donating his time and effort to assist me in obtaining a very important literature reference. He was, without a doubt, my last hope of obtaining that reference.

As in any effort of this kind, the number of people who lend their assistance are simply too numerous to mention by name. That does not prevent me from at least thanking them collectively. So to my thesis readers, the staff of the Signal Processing Lab and all those whose pictures are stored in the data bank, my sincere thanks.

A final word of thanks goes to my wife, Gladys, and my children, Catherine and Christopher. Their understanding and forbearance, with the grace of God, has enabled me to accomplish something I never dreamed possible.

## Table of Contents

## List of Figures

## List of Tables

AFIT/GE/ENG/86D-36

# ABSTRACT

A computer algorithm was developed which successfully locates and identifies human face(s) that are present in a digitized computer image. In the process of finding the facial image, the algorithm simultaneously determines the boundary locations of the sides of the eyes, the center of the face, the tip of the nose and the vertical center of the mouth.

Detection of facial images is based on analysis a digitized scene for the presence of characteristic facial feature signatures for the eyes, nose and mouth. These signatures are generated by the application of a "center of mass" calculation to each pixel row and column for various sub-sections of the digitized scene. The presence of a face is confirmed, and its feature locations are determined based on the presence and location of the local maxima and minima which occur in these curvilinear signatures.

Once the face(s) have been located, individual face recognition is performed by calculating the "gestalt point" for six different regions of the face. The gestalt is a location, in the 2-dimensional facial region, which corresponds closely to the center of mass of the pixel intensity distribution for that region. Identification of

the unknown face is performed by comparing its gestalts with those for known faces, using a distance metric.

The algorithm's face locator function was tested on 139 facial images representing thirty different subjects. The algorithm successfully located and bounded the internal region of the face in 94% of the cases. Further tests, against a limited number of arbitrary backgrounds, indicate that the algorithm is highly specific for faces.

The face recognition portion of the algorithm was tested against a database of 20 different subjects. In two trial runs, using the same 20 person database, 18 of the 20 subjects were in the top three candidates selected. In one trial run, the algorithm successfully identified the unknown individual (selected the proper individual as the number one candidate) 60% of the time. In the other trial run the proper candidate was identified 50% of the time. Recognition was based solely on analysis of the internal facial features.

# I. Introduction

## Background

In his thesis "Performance of a Face Recognition Machine Using Cortical Thought Theory"(22; Appendix A), Capt. Robert L. Russel successfully developed and tested a Face Recognition Machine. The Face Recognition Machine, when given a digitized picture of a human face, is capable of identifying the person to whom the face belongs. Identification is accomplished by first "training" the machine to recognize an individual's face. Another picture (one on which the machine has not been trained) is then input to the machine, and a program is run to identify the individual. Captain Russel tested the machine using a data base of 20 different individuals. Test results indicated that the machine was able to successfully identify the test subject 90% of the time. In the remaining 10%, the test subjects were high on the machines candidate list (second or third in a list of 20).

To train the machine, a series of pictures of an individual's face are taken (with a video camera) and stored in a digital data base on a general purpose computer. The digitized facial pattern data is then processed according to the "Cortical Thought Theory" (CTT) model of the human brain, developed by Captain Richard Routh (21).

Cortical Thought Theory involves the use of an algorithm which calculates the "gestalt" of a given pattern. According to this theory, the "gestalt" represents the essence or "single characterization" uniquely assigned to an entity (in this case a 2-dimensional image) by the human brain. Mathmatically, the gestalt is calculated using a 2-dimensional discrete transform which operates on the pixels (individual picture elements) of the digitized picture. The result, for any given picture, is a set of numbers (cardinality of 2) which represent a location on a 2-dimensional grid. This location varies according to the pattern of the image being transformed. In general, the gestalt location seems to closely correspond to the "center of mass" of the pixel intensity values (with dark pixels having the higher mass).

When initially tested using pictures of the entire face, the system's ability to discriminate between different faces (using the process outlined above) was poor. A technique referred to as "windowing" was then developed to improve the performance of the system. In the windowing technique, the face is automatically divided into characteristic regions and then each region is processed according to the CTT model. Figure 1-1 illustrates the windows used and the results of the gestalt calculation for each window. The numbers in parenthesis (above each respective window) correspond to the "gestalt" or (X,Y) point of the center of mass.

Thus, the training process involves calculating a set of gestalt points (one for each window) for each of several

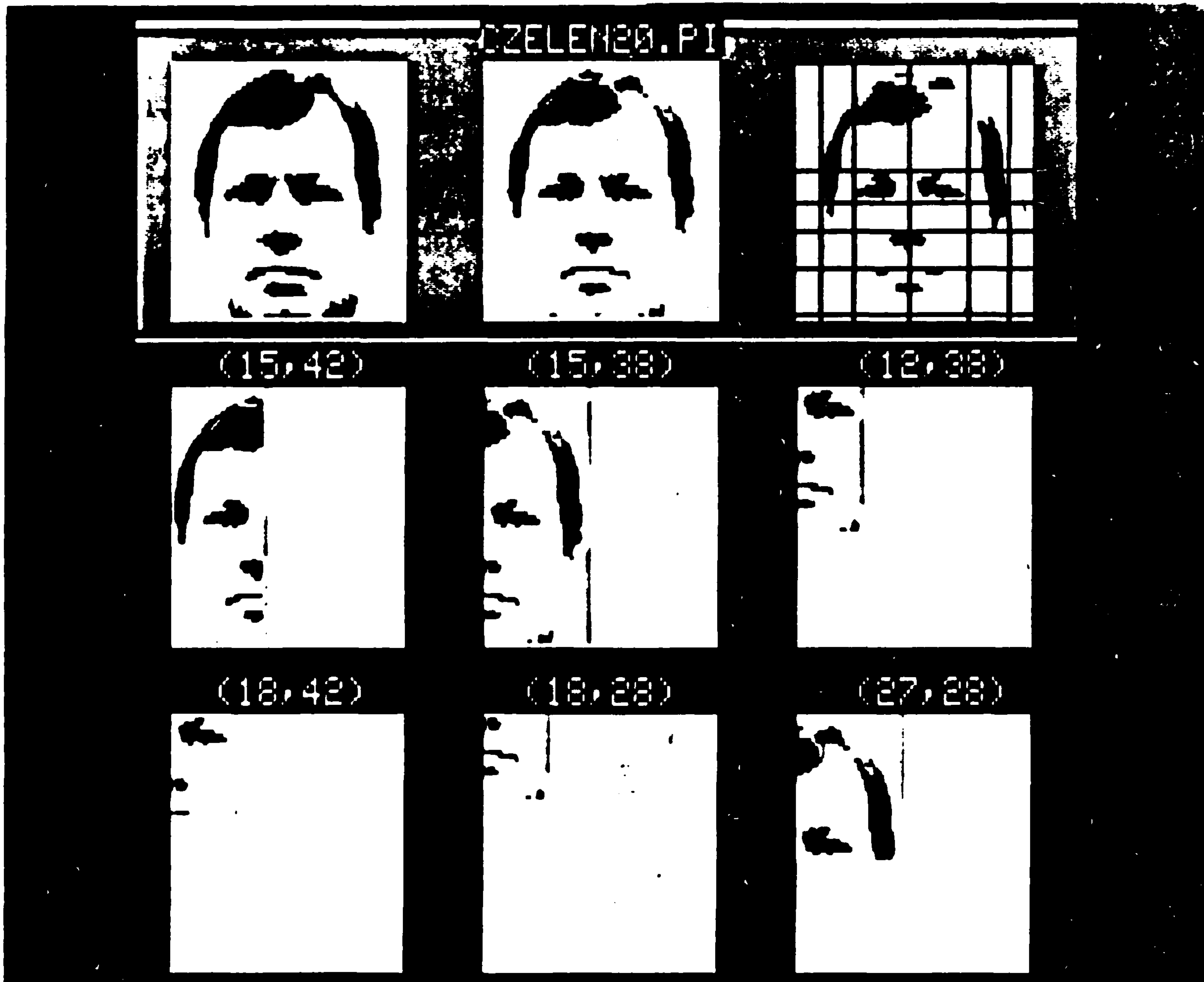



Figure 1-1. Sample Output of the Original Face Recognition Machine.

pictures of an individual, and storing the gestalt points obtained for that individual in the computer memory. When identifying an unknown person, the machine calculates the unknown person's gestalt points and compares these values with those stored from previous training sessions. The person corresponding to the gestalt points closest (overall) to that of the unknown person is selected as the prime candidate.

## Problem

In Russel's implementation, the machine often requires operator intervention in setting up the proper window boundaries. These boundaries are indicated by the horizontal and vertical lines in the picture located in the upper right hand corner of figure 1-1. The operator intervention is subjective in nature (such as having to determine where the bottom of the eyes are located) and the machine's performance is sensitive to it. As the data base continues to grow, it is expected that this sensitivity will become more critical.

For this thesis effort, the author proposes to remove the machine reliance on operator intervention. This requires that the machine be able to locate a face or faces in any given picture and perform the required windowing in a completely autonomous fashion .

## Scope

This thesis effort is concerned with enabling the machine to autonomously locate and window the human face from a digital image, locate the major features of that face, and create various sub-windows of that face for further face recognition processing. The system is designed to process a picture with more than one face in it. No effort will be made to speed up the gestalt calculation process (as developed by Captain Russel) or compensate for variations in face orientation with respect to the camera.

## Assumptions

In any given picture, the subject(s) are looking squarely at the camera (there is no tilt or rotation of the head). Subjects faces are not obscured and lighting is consistent. The subject is not wearing glasses and the facial area is free of unusual markings. The subject is not moving and has a relatively relaxed expression (the face is not being deliberately contorted). Four pictures for each subject are sufficient to characterize a person. The basic CTT algorithm used in the original machine is valid.

## Standards

Test results must meet the same criteria as set out in Captain Russel's thesis:

1. System must demonstrate "human like" classification of human face images.
2. System performance must be repeatable.
3. Recognition performance must be as good as that obtained by Captain Russel (90% successful).
4. All critical components and assumptions remain consistent with CTT.

In addition, the system must never require operator intervention in locating and windowing facial patterns in a picture. Routine operator control such as initiating and terminating the process will still be required. The system must also be able to process pictures with multiple (at least two) faces in them.

General Approach

As stated earlier, the machine has demonstrated a sensitivity to subjective operator intervention. It would be useful, both theoretically and practically, if the machine could be made to autonomously locate and bound a face, as well as perform any necessary windowing. To accomplish this task, a software routine was developed which locates the facial image(s) by performing scene analysis of the digitized picture. The basic algorithm for this technique is a facial signature search routine. The key signature used is one which corresponds to the human eyes. Once a set of possible eyes have been located, the software routine then examines the image for a nose and mouth signature. If all signatures are identified and are located in acceptable positions, then the presence of a face is confirmed. This algorithm enables the machine to locate a face(s) anywhere in a picture, since it is shift invariant. Unlike the original machine, it is necessary to work within the "inner" boundaries of the face. This domain effectively excludes access to information in the hair, ears and bottom of chin. This constraint is necessary to allow for processing of images in which the background is not required to be noise free.

Once the faces have been located and bounded, each face is processed separately. Contrast enhancement of each facial image is implemented using a slightly modified version of Russel's original histogram stretch routine. Ideally, once the proper contrast enhancement has taken place, all and only that information which makes up the essence of the face is displayed. The windowing process is then performed using the original software developed by Captain Russel (modified as necessary). In general, the original software is utilized (for example the data base software) to the maximum degree possible.

Recognition testing will be performed using a minimum of 20 different subjects. For each subject, the system will have been trained with a minimum of 4 different pictures. One picture will be held in reserve for testing (not used for training). Results will be compared with the criteria set forth in the "Standards" section above and with Captain Russel's original results. Several pictures, with one or more subjects in them, will also be tested to verify the AFRM's ability to locate faces in a noisy background.

## Materials and Equipment

The equipment necessary to develop and test the required computer software is as follows:

- Data General Eclipse S/250 Computer System
- Data General Nova 2 Computer System
- Octek 2000 Video Processing Board
- Dage 650 Video Camera (F-Stop and Zoom control)
- Panasonic WV-5490 Monochrome Monitor
- Tektronix 4632 Video Hard Copy Unit.

All equipment is available and located in the AFIT Signal Processing Lab.

## Thesis Structure

The first chapter is intended to serve as an introduction to the Face Recognition Machine (FRM). This chapter contains a brief review of the underlying theory and performance capabilities of the FRM. It also outlines the problem which this thesis addresses and the approach taken to solve this problem.

The purpose of chapter two is to review the literature concerning face recognition. After a brief review of the general literature on automated facial image processing, the reader is acquainted with several previous research efforts which closely parallel the work done in this thesis effort.

In chapter three the reader will find a description of the design and development of the face finding and face recognition algorithms. Since the algorithm used for face recognition is largely the same algorithm that was used in Russel's work (22), discussion on this topic is limited to any necessary modifications. The emphasis of this and the next chapter is on the face finding algorithm which is unique to this thesis.

Chapter four explains the system implementation details. The reader is first given a brief desciption of the hardware and software used in the development and test of this system. He is then provided with an in depth look at how the Autonomous Face Recognition Machine (AFRM) is implemented on the Eclipse/Nova Data General Computer system.

Chapter five contains descriptions and results of testing designed to evaluate the AFRM's ability to find and identify individual faces.

Conclusions based on the results of tests, and recommendations for future development are described in sixth (last) chapter of this thesis.

# II. Background of Facial Image Processing

## General Aspects

"No other object in the visual world is quite so important to us as the human face" (6:1). The previous quote is well substantiated by the existence of an extensive literature base on the general topic of recognition and perception of human faces. In the same text (6), there are several hundred references concerning this topic. Another author (2), has published a separate bibliography on topics concerned with only face recognition. The range of various aspects concerning this topic is also quite remarkable. Studies concerning the legal, developmental, psychological, emotional and other aspects of face perception are readily available.

The reasons for such a extensive interest in this topic are many. Obviously, almost everybody is interested in his own ability to recognize and correctly interpret the information contained in facial images. This is commonly attributed to the fact that man is a highly social animal. Without the aforementioned abilities, the social interactions of any individual are severely handicapped. For those whose interest lie in how we percieve faces, the reasons are sometimes not quite so obvious. In the case of this thesis, the motivation is to see if a machine can be produced which finds and identifies human faces.

In attempting to implement the pattern recognition capabilities of the human brain on a machine, one quickly comes to the realization that the brain is an astounding pattern recognition machine. Many theories have been put forth concerning the mechanism by which the brain performs various (e.g. visual) pattern recognition tasks. And none appear to have been more than, at best, partially successful in mimicking the brain's performance. However, the bottom line is still that the state of the art in machine pattern recognition has not even come close to matching the speed and versatility with which the brain (not necessarily just human) performs pattern recognition.

In the special case of "facial" pattern recognition, there are a wealth of studies available. These studies are concerned with such aspects as spatial frequency analysis (8;19), effects of feature displacement (10), effects of changes in visible area (13), and eye movement strategies (25) associated with facial images. However, in the case of machine analysis of visual images, both for the presence and identification of facial images, relatively few theories or models have been proposed and tested. The next sections of this chapter will acquaint the reader with some of the previous work that has been accomplished in this area. The discussion in the following sections has been divided between finding facial images and identifying to whom those facial images belong. The result of this approach is that the parts of a researcher's work which are pertinent to locating faces appears in one section, and the parts pertaining to facial recognition appear in the following section. The approach of addressing these two tasks in separate sections is taken throughout the remainder of this thesis. The primary reason for this division is that the task of locating a facial image is a qualitatively different task than identifying a facial image. In the next section, efforts associated with the task of locating a facial image within a picture are examined.

## Locating a Face in a Visual Image

Sakai et al (24) reported considerable success in locating human facial images in photographs with a noisy background. The basic premise upon which the technique is based is "The meaningful information exists usually in the drastically changing portions of the gray level in a picture, and we extract these parts as line segments"(24:234). Thus, the initial phase in locating a facial image in a photograph is to transform the image into a line drawing which depicts only the edges that were present in the original image.

The technique used in determining the lines involves examination of the eight nearest neighbor pixels, for each pixel in the image. The result of this investigation is a vector (for each pixel) which indicates the general direction of greatest grey level change and the relative magnitude of that change. The 3 x 3 window used in such a technique is dependent on the overall resolution of the image. In the case of high resolution pictures (more pixels

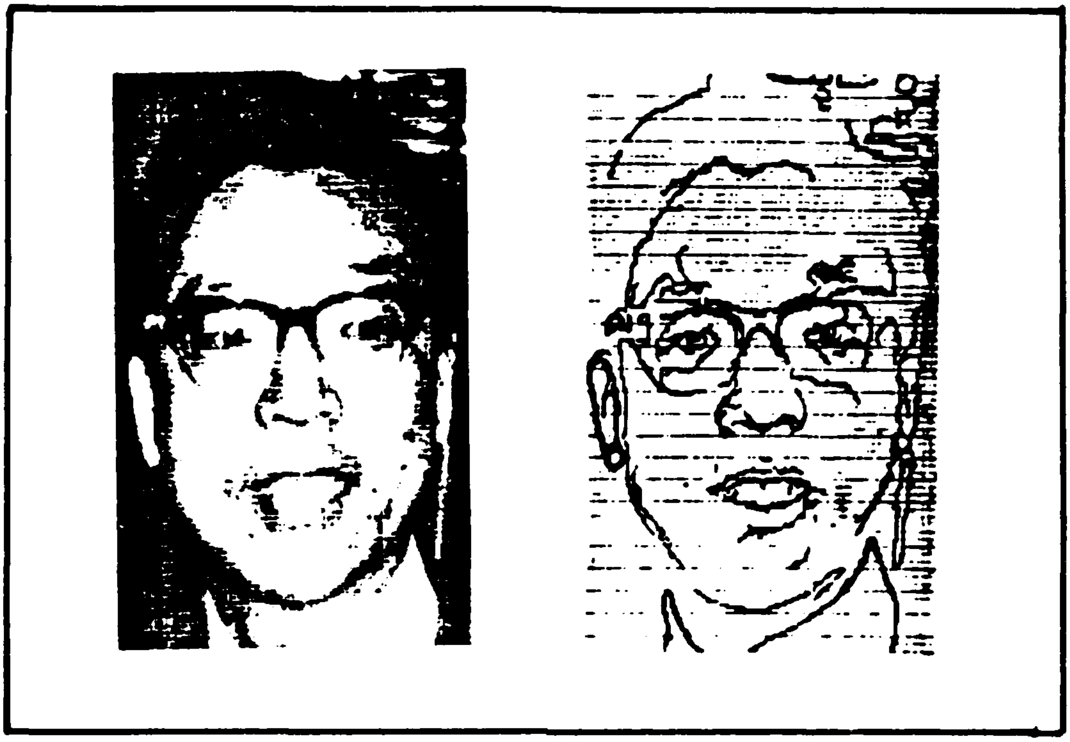


Figure 2-1 Example of a Line Extracted Image (24:243)

per unit area), larger windows (such as 9 x 9) might have to be used since the degree of intensity change across a smaller window may be too gradual to allow accurate edge detection. Through a fairly complex technique of thresholding, basic line elements are then established and extended to yield the final line extracted image. One example of the results of such a process is illustrated in figure 2-1.

The authors admit that this line extraction is not adequate for a number of different types of photographs. Exactly what types of photographs this technique was not suitable for they did not indicate.

The next phase in the process is to locate the face or faces in the line extracted image using a pattern matching technique. Figure 2-2 illustrates the templates used in the pattern matching process.

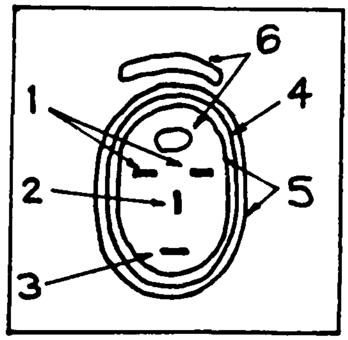



Figure 2-2 Template Used for a Human Face (24:244)

After the appropriate adjustment (apparently by a human operator) of the size of the templates to match the size of the facial images in the line extracted photo, the image is scanned to find a match with templates 4 and 5 (the facial outline). Once this is achieved, the image is evaluated to see how well it matches with template 6. Template 6 describes regions in which no or relatively few lines should be present. Finally, the image is evaluated to see if there is a match between it and templates 1, 2 and 3. This last requirement is very flexible in that if any one of the templates (of 1, 2 or 3) are matched strongly, or if two or three of the set are matched mildly, the result is still considered a match. When all of the aforementioned criteria are met, the presence of a face is considered to be confirmed at the location of the central point of the templates.

Although the authors claim this technique was successful in finding faces in many different pictures, they gave no figures on just how successful they were. As might be expected in any technique which relies on grey level intensity analysis, the authors affirmed the susceptability of this technique to variations in lighting.

In another effort related to locating a face in a visual image, Bromley (4:17) used a facial signature technique to locate the vertical center and sides of the face. The primary emphasis of Bromley´s thesis was to develop the capability to automatically locate various facial features, once the location of the facial image was known. However, as part of the initial processing, her algorithm required the capability to locate a face in a somewhat benign environment. The limited constraints under which the search for the face was conducted are evident by the use of police mug file images which were roughly centered on the face in question and in which the facial image constituted a large part of the entire image. Figure 2-3 illustrates an example of the images used.

Also shown in figure 2-3 (bottom part) is an example of the results of applying a "row signature" technique to the mug file digital image. The signature is generated by simply summing the rows of pixel intensity values for each column and plotting the results. A characteristic peak almost always occurs in coincidence with the center of the face,

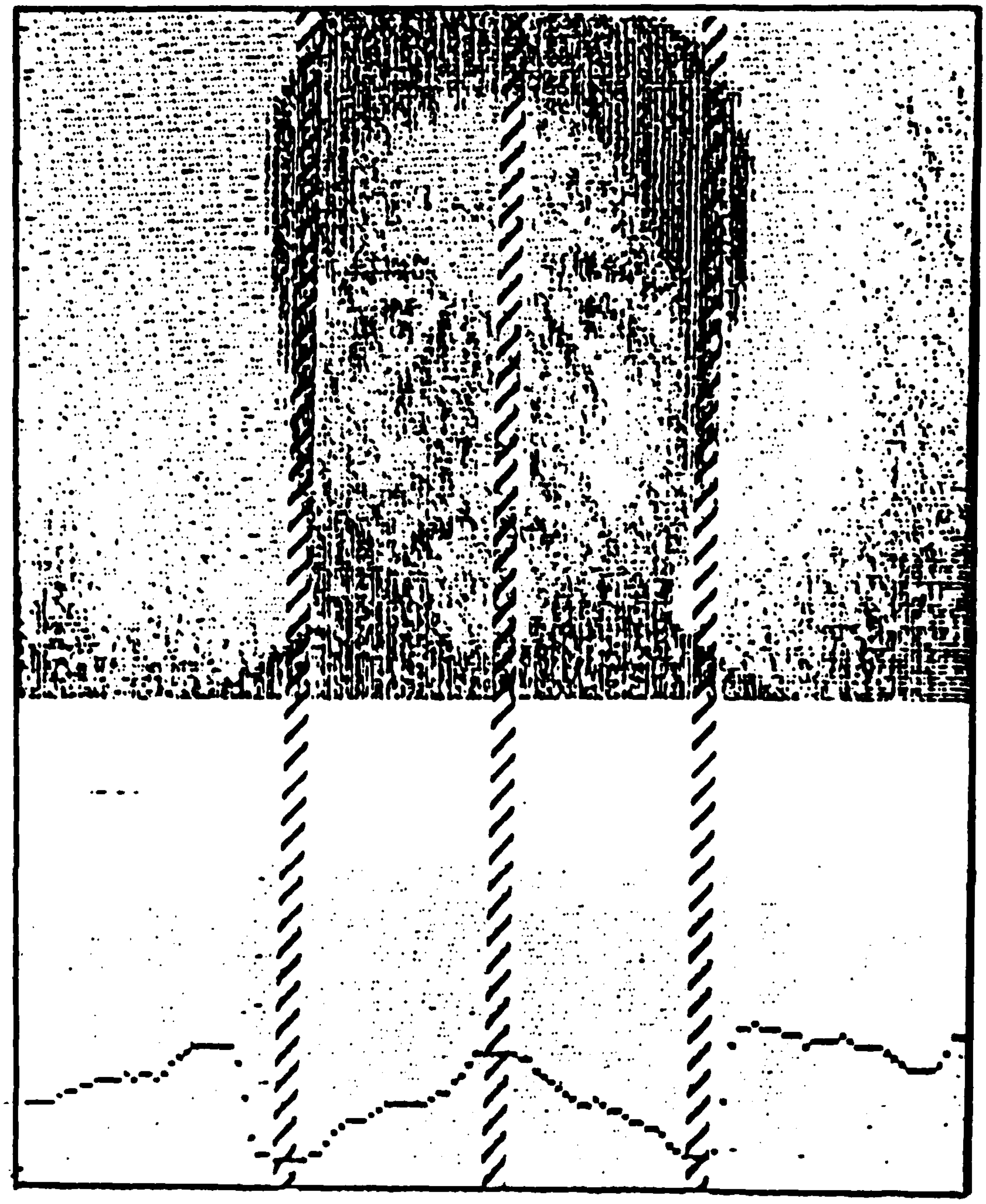


Figure 2-3 Typical Mug File Image (4:26)

and characteristic "vallies" occur at the locations of the sides of the face. Bromley reported (4:47) fairly consistent results using this technique in a limited number of cases. Her algorithm relies on a fairly pristine background and certain physical attributes of the subject face being analyzed. Still, it does suggest a possible technique for locating facial images under much less constrained conditions. In fact, the basic concept used by Bromley is remarkably simiiliar to the technique used to generate the eye signature as discussed in the next chapter.

A third technique related to locating facial images in a digitized image is based on image correlation. In an experiment conducted by Baron at the University of Iowa (1:145-151), a system was devised to perform automatic face recognition from digitized images of faces. As in Bromley's case, the main thrust of the research was not to investigate automatic facial image location, but rather automatic face recognition. Unlike the mug file images used by Bromley, the pictures used in this case did contain background clutter and were taken under much less stringent conditions. The pictures were full face images as illustrated in figure 2-4. Again the author was faced with the task of locating the face (or at least part of the face) as an initial step of the processing. The approach taken was to search the image for the subjects eyes using "eye templates" in a correlation algorithm as illustrated in figure 2-4.

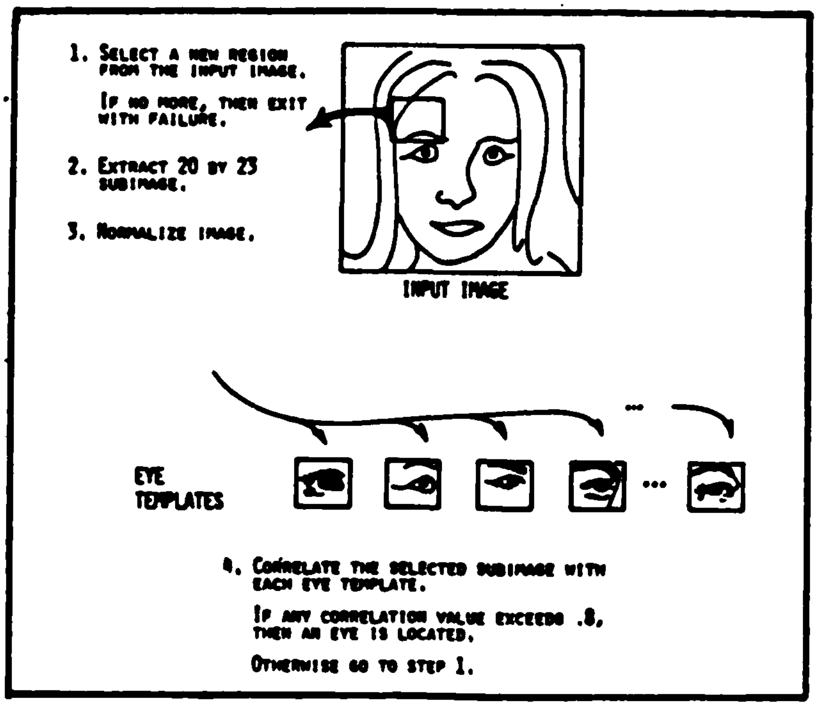



Figure 2-4 Correlation Procedure for Locating Eyes (1:146)

The author indicates that the purpose of locating the eyes is to facilitate standardization of the size of the facial image based on the distance between the eyes. Once the size of the facial image has been standardized, further correlation mechanisms are used to perform recognition as discussed in the next section of this chapter. Baron reports that the technique enjoyed considerable success in both locating the eyes and performing recognition. This technique could be extended to searching for a face in an image in which the face occupied a much smaller portion of the overall image. This would constitute a very large computation since various template sizes as well as types would have to be correlated with every location in the image.

Each of these three approaches is uniquely different from the other. The only one of the three which addresses the more complex problem of extracting facial images from noisy backgrounds is Sakai's approach. However, Sakai's approach is reliant on a human operator to adjust the size of the masks to the size of the facial images in the test picture. In an autonomous machine this would mean that the search would have to be conducted over a range of mask sizes. This approach, like Baron's, would be a very time consuming task. The problem of locating a face in Bromley's work was the least complex of the three. The major distinction between her work and the other two is that she first transforms the facial image into a simple signature and then determines the location of the face. The primary advantage of such a technique is speed.

Recognition of Individual Faces

One of the basic assumptions of this thesis is that the CTT based recognition scheme designed by Russel (22) is valid. To apply that theory and design, it is necessary to window portions of the face based on consistent facial landmarks or features. Automatic facial feature location then becomes an absolute prerequisite to applying the CTT based model. The following is a brief review of previous work done on locating facial features for the purposes of classifying and/or recognizing individual faces.

There have been several attempts made at automated and semi-automated facial feature measurement, usually with the goal of either recognizing a face or at least classifying the type of face. Semi-automated facial feature measurement typically involves a human operator working interactively with a machine (computer) to analyze facial images. It is usually the human operator who performs the pattern recognition task and the computer that does the bookkeeping and classification. Some of the major early contributors in semi-automated measurement were Bledsoe (3) and Harmon (12). The results of their work have been summarized in Captain Russel's thesis (22) and will not be reiterated here.

In the area of fully automated machine classification and/or recognition of human faces, Harmon (11) again was one of the key contributors. Unfortunately the work performed by Harmon was concerned with human face "profiles" and to a large degree is not applicable to this thesis (since this effort is concerned solely with frontal face views).

Another researcher, Bromley (4), developed an automatic feature locating algorithm for use in reducing the number of mug file pictures that had to be reviewed by a witness to a crime. The type of features located by her algorithm are indicated in figure 2-5.

The technique used by Bromley to locate the vertical features (LF or left face,NL or nose line, and IRF or right face) was described earlier in the section on locating faces. The horizontal feature locations (for example the eyes or LE) were acquired by analyzing a curve generated by taking a finite difference derivative of the pixel values along the vertical nose line. This information, when used in conjunction with estimates of where the appropriate features should be, allowed fairly accurate determination of the desired feature locations. For instance, in determining the top of head boundary location, one would search down (along the nose line) until the first maximum negative derivative value was found. Assuming the top of the head is darker than the background (a safe assumption in mugshots), and that the background is relatively noise free, the top of the head would then correspond to the location of the maximum negative derivative. Since the center, sides and top of head are now known, reasonable estimates can be made concerning the location of the rest of the desired features. Bromley apparently had reasonably good success with this technique. Unfortunately, development of an algorithm which could use the feature location information to classify facial images was not a part of her work. Thus no facial classification/identification performance statistics are available.

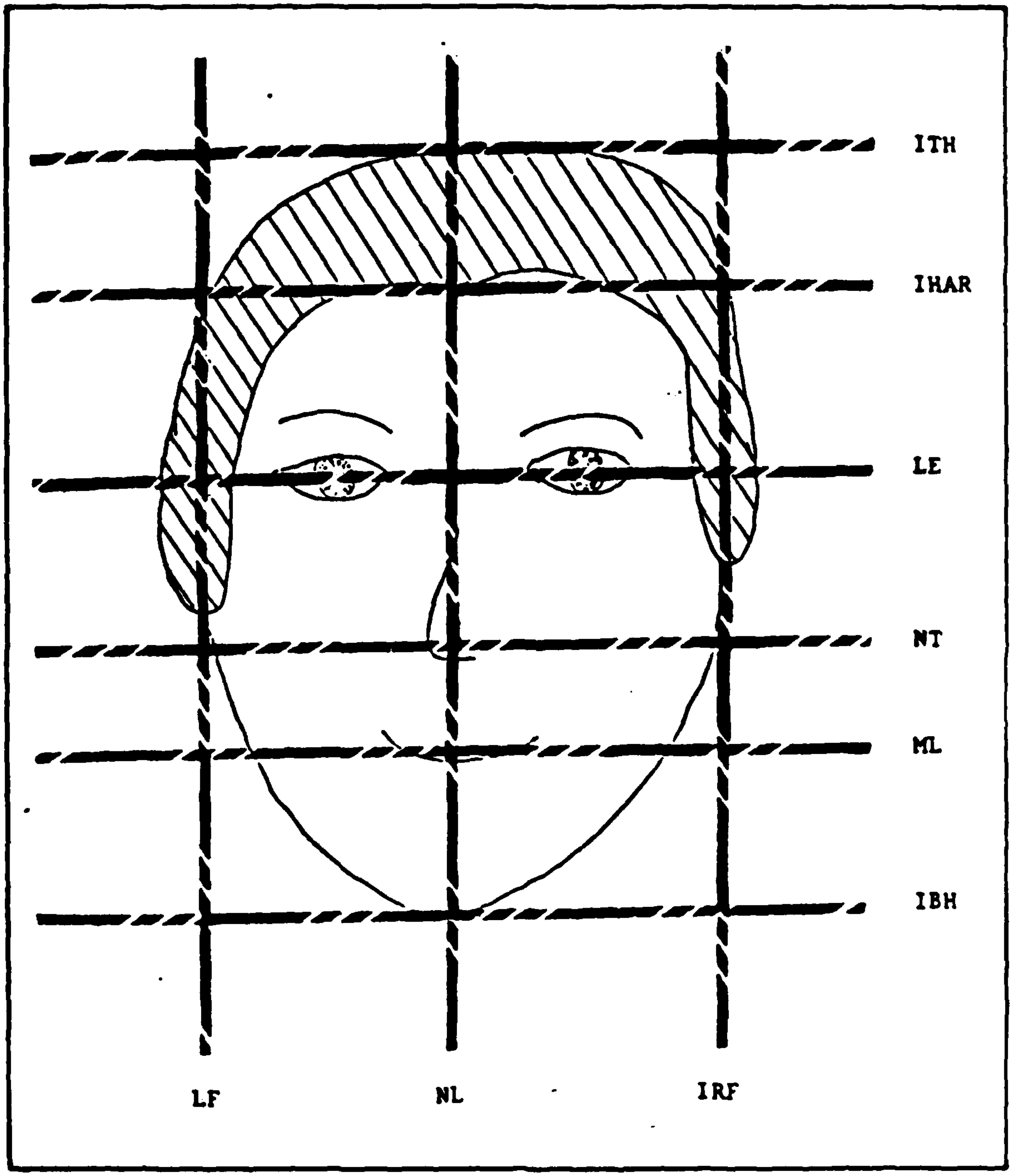



Figure 2-5 Bromley Feature Set (4:25)

In 1965 (18:78), Vander Lugt demonstrated how instantaneous recognition of faces could be accomplished using the optical computer illustrated in figure 2-6. Such a device performs an "optical" correlation of the input image with the test image. The result is a bright spot on the output pattern which corresponds with the location of an image of similiar size and spatial arrangement that is on the test image.

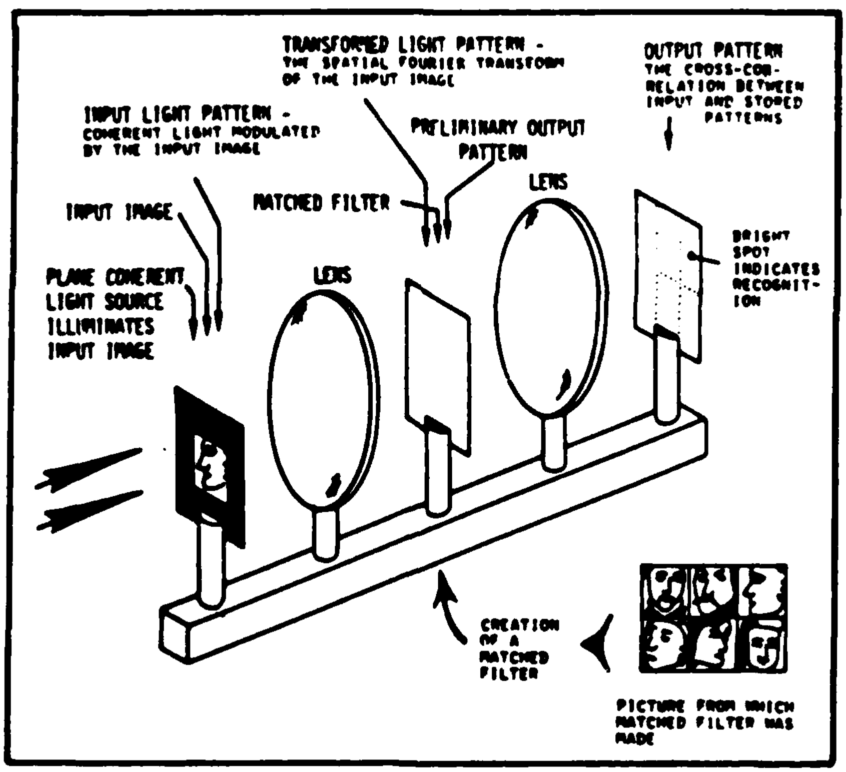



Figure 2-6 Optical Computer for Recognizing Faces (1:139)

Baron (1:140) proposed a model for the way the brain processes information based on the same concept of correlation. As mentioned previously, he implemented and tested an automatic face recognition system based on this theory. Baron et al used an IBM System 360 Model 65 computer in conjunction with a Spatial Data 806 Computer Eye digitizer. The initial images were 512 by 480 pixels which were subsequently reduced to 128 by 120 pixels through averaging of 4 by 4 squares of the original image. After locating the subject's eyes (described previously) and adjusting the size of the facial image to a standard size based on the distance between the eyes, the image was reduced even further to one of 15 by 16 pixels. Further looks at various regions of the face were also acquired and reduced to 15 by 16 pixels as illustrated in figure 2-7.

The only instance where the system was not performing in a fully autonomous mode was in the determination of which various sub-looks (mouth, chin and others) were to be stored in its data base. A human user was required to supply the computer with a list of features of interest and apparently had to point out the exact location of those features during the initial training session. Additional pictures of the same subject were then added to the data base using the control information initially supplied by the user. Each face was represented by up to 20 templates like the ones indicated in figure 2-7. The system was tested using a data base of 42 individuals. Recognition accuracy was 100%. In addition, a set of over 150 faces (which included the original 42 faces) were tested, with the system rejecting the faces not in the data base and recognizing all of the

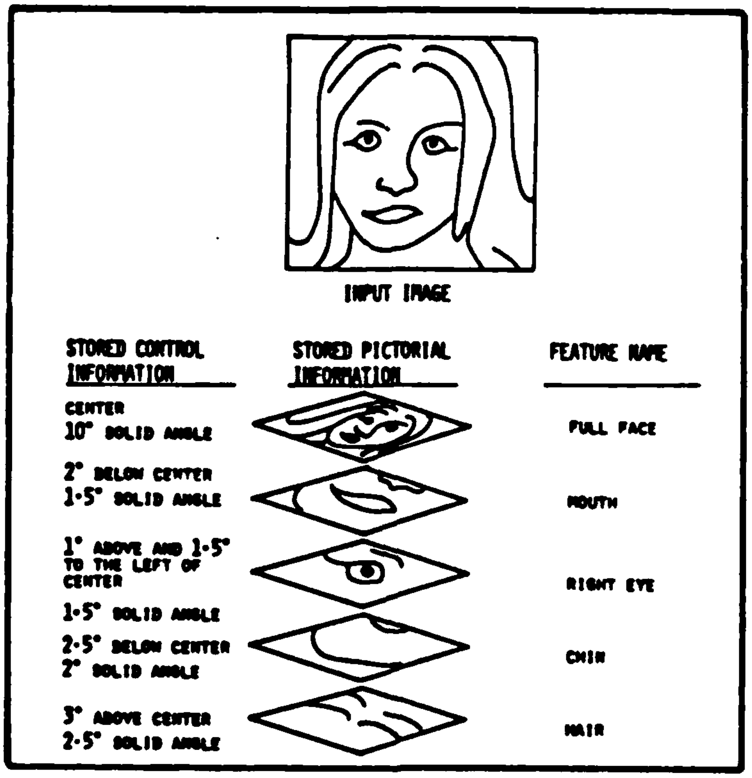



Figure 2-7 Baron's Stored Representations of a Face Image (1:147)

faces that were in the data base. The system performance remained unaffected by face rotations up to 20 degrees, which interestingly is consistent with some recent findings (17:340) concerning the performance of what appear to be face specific neurones in the rhesus monkey brain. Finally, it was found that unlike the performance of human face recognition, the system could not adequately handle large changes in the size of the image.

Although other efforts have been conducted (14; 15; 23), the only remaining effort in face recognition of consequence to this thesis is that conducted by Russel (22). Since the theory, design and performance are covered elsewhere (Appendix A and chapter 1) the reader is referred to those areas for any desired further background information.

# III. System Design

This chapter documents the top level design and the development and design decisions that were made in developing the Autonomous Face Recognition Machine (AFRM). The chapter is divided into two main sections. The first section concerns development and design of the algorithm which determines the presence and location of facial images within a digital image. The second section examines the algorithm used in performing face recognition. The decision to divide the development along these lines is not totally arbitrary. Experts in the field of face recognition seem to agree that these two tasks are indeed separate and serial (5:321). The phrase "separate and serial" refers to the fact that a face must be first identified as a face before further specialized processing (facial recognition) can be performed.

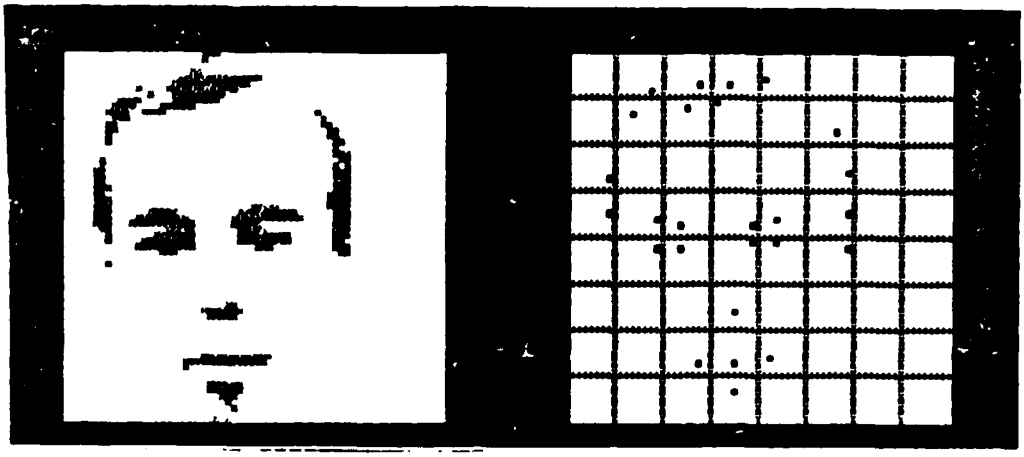
## The Face Finder

As mentioned in the introduction, the approach used to find a face in a visual image is dependent on the use of certain facial area signatures. The general nature of these signatures is such that they are almost always present when a face is in the image and are sometimes present when a face is not in the image. An algorithm was developed which methodically searches for these face specific signatures in some test image. Once a facial image has been found, the system stores the necessary data for later retrieval by the face recognition algorithm. The following sub-sections provide a detailed explanation concerning what these signatures are, how they were developed, and how they are used in locating facial images.

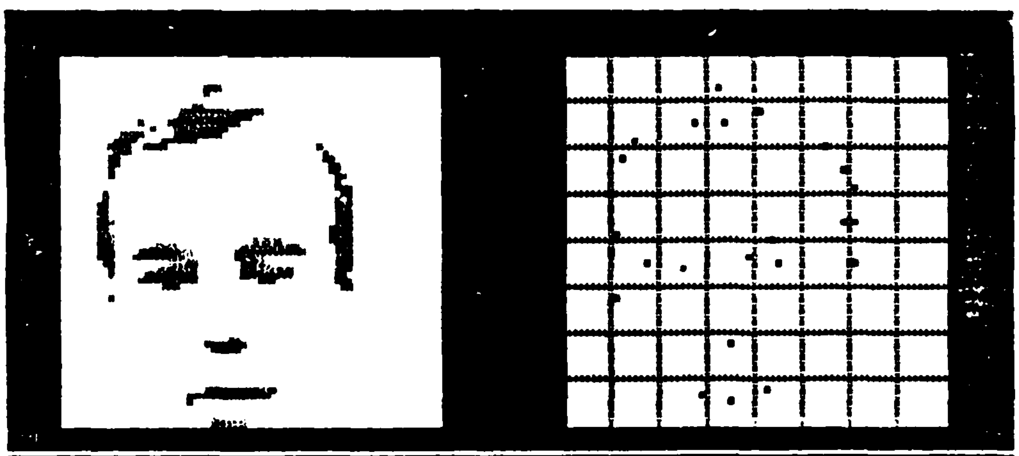
A brief review of the literature reveals that the number of ways of segmenting an image are limited only by the imagination and tools of the researcher. Rather than attempting to evaluate all of these techniques and then pick the best possible one for use in locating faces, the decision was made to use the gestalt calculation (or some variation of it) as was originally introduced in Russel's thesis (22). Since the gestalt calculation has a demonstrated capability for distinguishing between faces, then it should be possible to apply that same calculation to locating faces. Any calculation capable of detecting the nuances between individual faces ought to be capable of distinguishing faces from other classes of objects. Additionally, if the brain is capable of doing this calculation in the performance of recognition, then it is also possible that a similiar calculation is used in locating a face.

The first idea to come to mind is to use the original face recognition algorithm (as developed by Russel), in a direct manner, by developing a set of windows and gestalt values for a "generic face" and evaluating an image to see where and if the gestalt values of the image match up with the gestalt values of the generic face. Assuming that such a generic facial feature set could be developed, the major problem with this appoach is that searching an image in this manner would be a very computation intensive task. Essentially, every pixel location of the image would have to be evaluated using several windows. If the facial size were variable, then additional calculations (for each pixel location) would have to be made over the range of possible sizes. Using such a technique, the number of false alarms would be expected to be high when attempting to confirm the presence of a face. This high false alarm rate would be due mainly to the use of a limited number of large windows (i.e. six). The reduction of information is so great that, for any particular window, a large number of non-facial spatial distributions may result in the required gestalt.

On the hypothesis that perhaps the reduction of information in the original algorithm was too large, the results of performing the gestalt calculation on larger sets of smaller windows was examined. Figure 3-1 illustrates what is obtained when a 64 by 64 image is divided into 64 8 by 8 windows. Except for a modification to perform the calculation on a 8 by 8 instead of a 64 by 64 window, the calculation of the gestalt locations for each 8 by 8 window (in figure 3-1) was done in exactly the same manner as Russel's original calculation (Appendix A:A-3).



a. Original location



b. Shifted Down

Figure 3-1 Facial Gestalts for 8 by 8 windows.

The gestalt values are indicated by the respective single dark pixels within each 8 by 8 box. Boxes which were completely white received no gestalt value. The reader should note that the facial images in figure 3-1 were enhanced to remove unwanted extraneous shadows prior to performing the gestalt transformations. Had this not been the case, then every 8 by 8 area would have a gestalt point in it, making the pattern somewhat more difficult to ascertain. Shifting the facial image slightly down (shown in figure 3-1b) causes a corresponding shift of the gestalt locations, which as was indicated previously, tend to follow the "center of mass" of the pixel intensity. The only discernable and consistent patterns which appeared in images like these (figure 3-1) was for the eyes, nose and mouth and then only very roughly. The patterns obtained using the 8 by 8 window approach were deemed too inconsistent to be useful in locating faces. However, the results did indicate that, if a consistent pattern was to be found, it would have to be done using the "internal feature area" of a facial image. The "internal feature area" refers to the area of the face including only the forehead, eyes, nose and mouth.

Another important observation was the need to enhance the facial image to remove unwanted shadows. In an automated procedure, the enhancement inevitably has to be done based on some characteristic(s) of the facial image. This requirement creates a delemma. In order to use a pattern to locate a facial image, the facial image must be properly enhanced; however, the facial image can't be properly enhanced until the face has been located . This mandates that any characteristic facial gestalt patterns, whatever they may be, must be initially obtained from the unaltered original image.

Development of the Eye Signature. Whenever someone is asked to locate something in a visual image, he typically looks for some unique characteristic of that object. For example, when looking for a tank, one might look for the turret and gun barrel configuration or possibly the treads. When looking for a rabbit the subject might look for two long ears. When looking for a face (at least from a frontal view), the singularly most distinctive characteristic to look for is the eyes.

Motivated by the previous reasoning and the observations made concerning figure 3-1, investigations were conducted to determine what kinds of possible patterns could be generated using the intensity information about the eyes. Figure 3-2 is indicative of the patterns obtained when just the eye portion of the original (unmodified) facial images are subjected to Russel's gestalt calculation (22:5-42).

The "windows" used in figure 3-2 are vertical slices which are 4 pixels wide by 64 pixels long (indicated by the vertical bars). Horizontal windows were not examined since the eyes are symmetric about the vertical axis and thus the center of mass (gestalt) for all points would essentially form a straight vertical line. Since any intensity image that is symmetric about a vertical axis would form this same signature, this would be an almost useless signature.

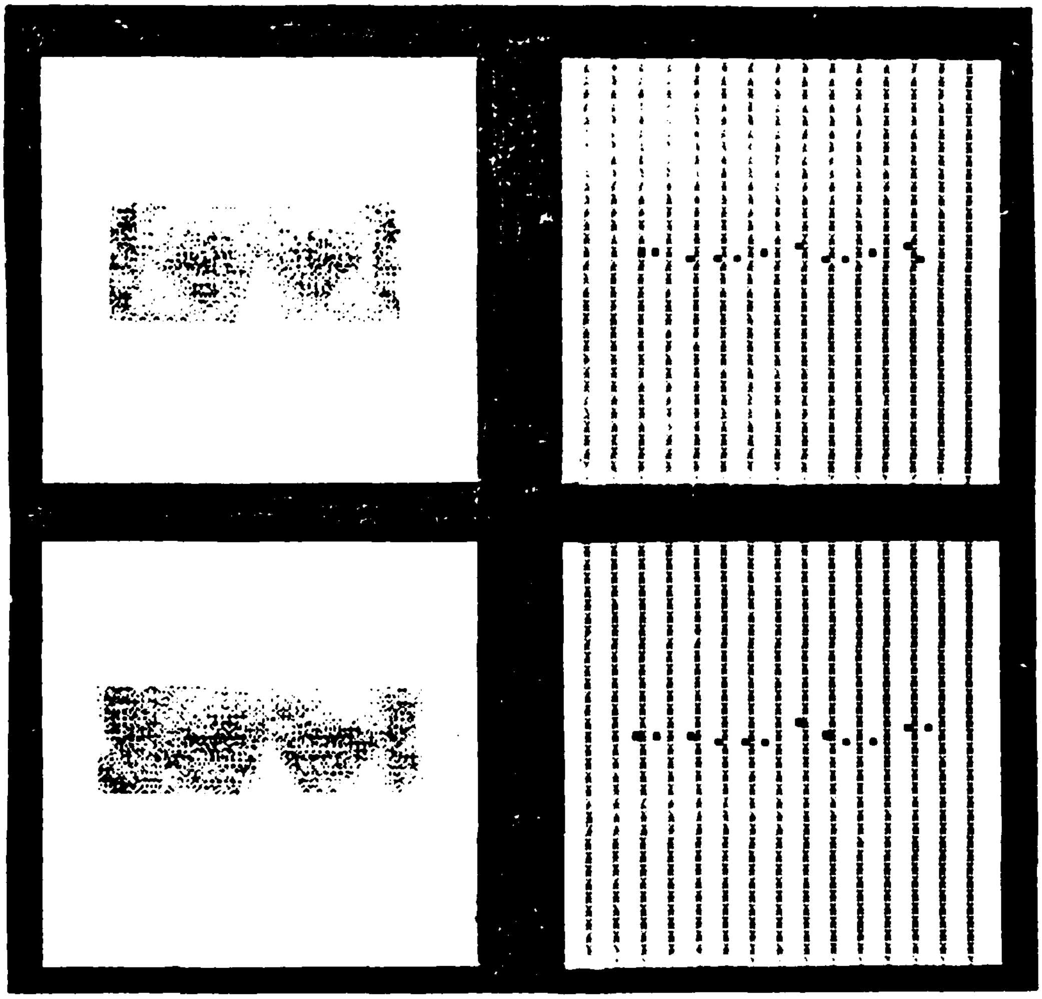
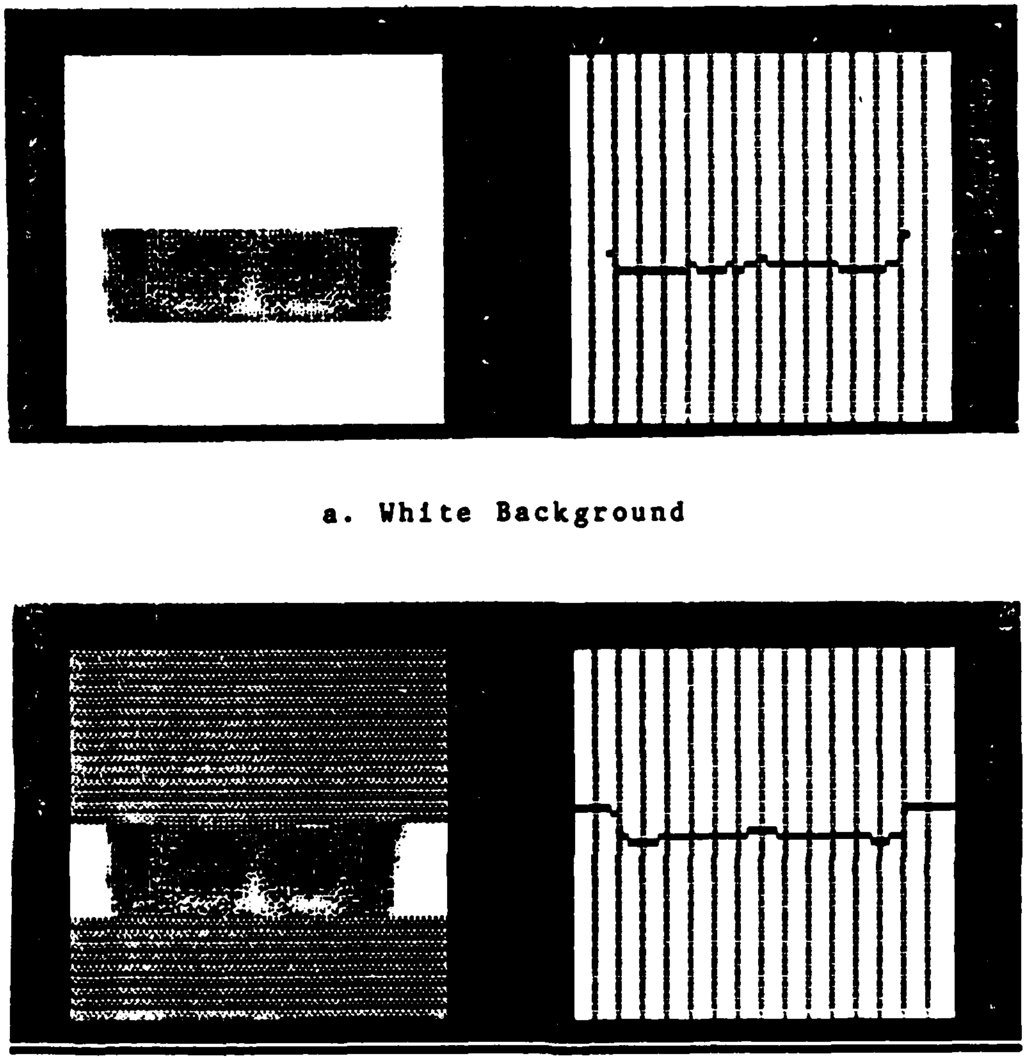


Figure 3-2 Gestalts for Eyes - 4 by 64 vertical windows.

The gestalt pattern illustrated in figure 3-2 is fairly shift invariant. Shift invariance, in this context, is taken to mean that if the eyes are translated to another position, the gestalt pattern is shifted along with the eyes, with no change in the size or shape of the pattern. To maximize the shift invariance, the vertical windows were reduced in size to the highest resolution of the system, namely, 1 by 64 windows. Using such windows resulted in patterns which were



a. White Background

b. Mid-Grey Scale Background

Figure 3-3 Gestalts for Eyes - 1 by 64 vertical windows.

essentially shift invariant. Thus, no matter where the face was placed in the image, the same pattern would emerge. Figure 3-3a illustrates a typical example of what is obtained using 1 by 64 windows on the eye portion of the unaltered original image.

The gestalt pattern illustrated in figure 3-3a can hardly be called a "signature". However, figure 3-3b illustrates that, when the "background" (the non-informative areas above and below the eyes) is filled in with intensity values representing mid-grey scale level instead of pure white, the gestalt pattern does begin to take on a discernable form. In fact, the form of the pattern is not unlike the one obtained by Bromley (see the row signature in figure 2-3). There is the beginning of a characteristic peak at the center of the signature which seems to correspond to the center of the face. In addition, there are characteristic minimums that seem to correspond to the sides of the face.

It was observed that the gestalt patterns, generated by using background fill intensities other than white (15), were position sensitive. Specifically, when the eye image is moved in the vertical direction, the signature changes significantly. This problem is easily solved by ensuring that all eye images are relocated to a standard position within the 64 by 64 image. It was found that the best position was at the top of the 64 by 64 image. This placement of the image coupled with the optimum background setting (also determined empirically) results in very consistent patterns for the eyes. It should be noted at this point that there were a total of 16 grey levels available with "0" intensity representing black and an intensity of "15" representing white. Figure 3-4 illustrates the patterns

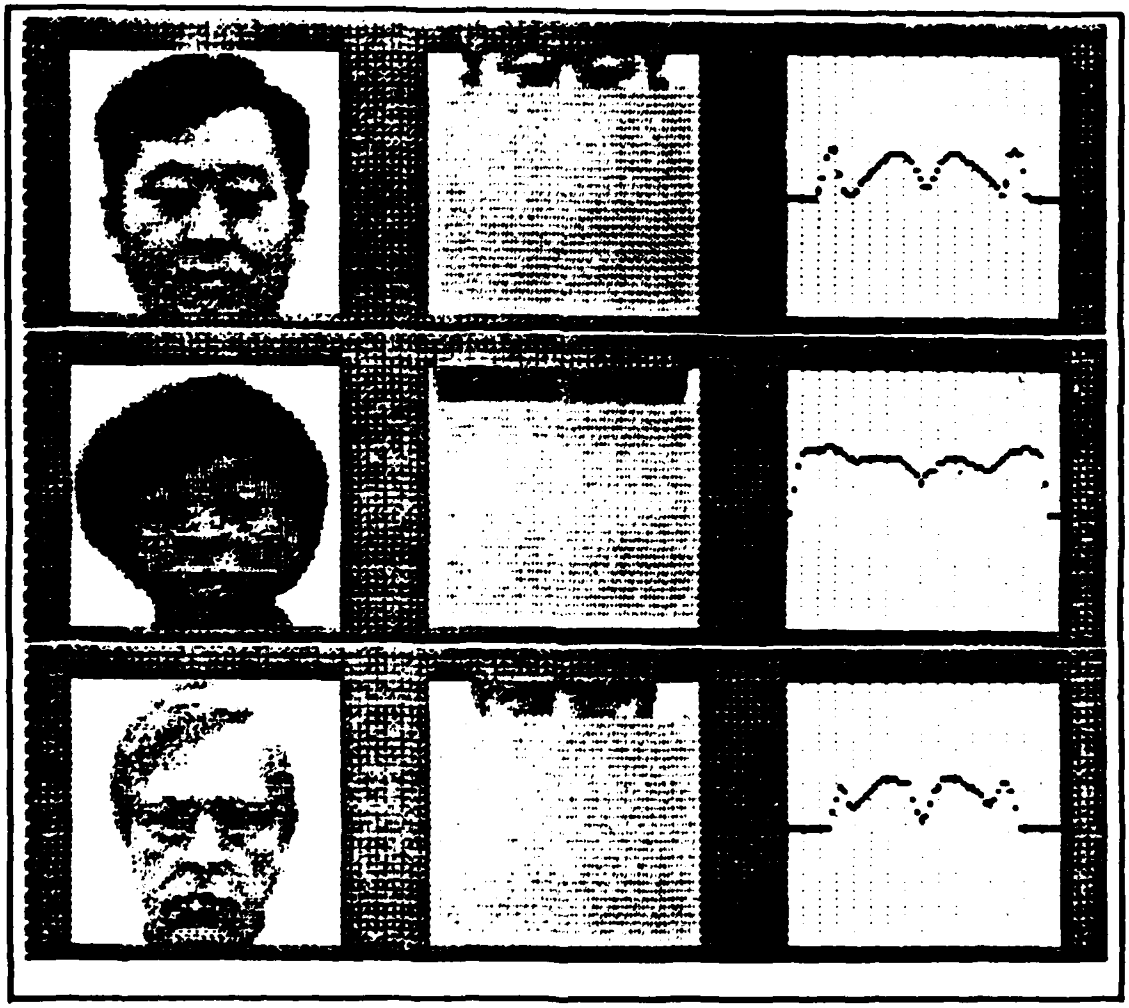


Figure 3-4. Typical Eye Signatures.

obtained for three fairly different subjects using an "optimum" background fill intensity of 12. The "background" is that region of the 64 by 64 image located below the eyes, in the central pictures in figure 3-4. An intensity of "12" was optimum in the sense that it minimized the effects of noise (sharp and sporadic changes in the eye signature) while maintaining fairly well formed peaks and minima.

The consistent pattern evident in the curves in figure 3-4, which from this point on is referred to as the eye signature, is that pattern characterized by the two central peaks and the local minima surrounding them. The outer peaks (on either side of the two central peaks) correspond to peaks due to hair on the side of the subject's head. These outer peaks do not constitute part of the eye signature.

Although the characteristic signature illustrated in figure 3-4 is present using a vertical window of as few as 2 pixels and as many as 16 pixels high, the vertical height of the windows used in each case was 8 pixels. The horizontal width was 1 pixel. The choice of an 8 pixel vertical window height was dictated by the need to limit the information to only the eyes. Since the size of the facial images in the data bank (for examples see Appendix D) are fairly standard, this window worked well for all subjects in the data bank.

An important observation at this point is that not only does this signature indicate the possible presence of eyes, but it also seems to contain information concerning the vertical boundaries of the face. Linking these boundaries to the locations of the three central minima provides exact definition of just where those boundaries lie. The minima to either side of the dual peaks are defined as the left and right sides of the eyes, respectively. The central minimum is defined as the center of eyes.

The manner in which the eye signature is obtained is as follows. An 8 pixel (vertical) by 64 pixel (horizontal) window is extracted from a 64 by 64 whole face image. The 8 by 64 window is placed at the top of a new 64 by 64 image and the remaining rows of the 64 by 64 image are filled in with a background grey level intensity of 12. The resulting image is then transformed into the eye signature pattern by calculating the gestalt point for each of the 64 vertical columns in the image. The calculation used is identical to that used by Russel (22:5-42) in determining the 1-D gestalt transform and is given as follows:

Given the discrete input image "$M_{kh}$",

where $k,h=1,\ldots,64.$

and $0 <= M_{kh} <= 15.$

The transformed image is given as:

$$T'_{ih} = \sum_{k=1}^{64} M_{kh} \exp \frac{1}{2}((k-i)/63\sigma)^2 \qquad (3-1)$$

$i,h=1,\ldots,64.$

$\sigma=0.435$

Once the above transformation has been performed, the gestalt location for each vertical column is determined by finding the location, in each column, which contains the largest value. The 64 gestalt locations can then be directly displayed as in figure 3-4 or saved as a discrete signal to be used in further analysis.

After ensuring (subjectively) that the eye signature was present for at least one of the several images available for each individual in the data bank (45 different facial images), an attempt was made to characterize the eye signature. Although there are several ways to transform and represent the eye signature curve, such as fitting the curve to spline curves, polynomial coefficients or Fourier coefficients, a more direct method was chosen. The method of analysis of the eye signature curves is performed by directly evaluating characteristics of the maxima and minima present within the curve. To avoid problems due to noise, which might shift the location of a peak or create small "false peaks", the discrete signal representing the eye signature is smoothed by convolving it with a gaussian function. This smoothing operation makes the gestalt value for a given column a function of the gestalt value for that given column, plus the gestalt values of the two nearest columns on both sides of the given column. Once the curve has been smoothed, each pair of successive peaks are detected and analyzed to determine if they meet the "eye" criteria. The local maxima and minima are detected based on the instantaneous changes occuring in the discrete signal and the previous changes which have occurred. Thus, if all the previous changes had been of increasing value, and then a change is detected of decreasing value, it is surmised that a peak has just been passed.

The primary criterion of the eye signature is the existence of two successive peaks whose maxima are located equidistant from the minimum between them. Once this criterion has been met, the curve is evaluated to see if the slopes formed between the outer minima and their respective peaks are less than 1.5 times the greater of the two slopes taken between the central minima and the two peaks. When these two criteria have been met, the possibility that eyes are present is confirmed. It is possible that other criteria may also exist in the eye signature, which were not exploited in this design. An exhaustive analysis of all similiarities for the eye signature was not performed due mainly to time constraints. One characteristic similiarity that was considered, but not incorporated as a criterion, was that the two peaks are roughly equal in height. However, due to variations in lighting this is not always the case. Under more controlled lighting conditions, this criterion could and should be used.

Development of Nose/Mouth Signature. The "eye signature" is not unique to just the eyes. Many objects and arrangements of objects could give rise to the eye signature, such as two vertical bars which are darker than the areas to either side of them. In order to specify a total facial signature, which is more specific to the face, further development was necessary.

A number of factors drove the decision to investigate the presence of a nose/mouth signature. First, the eyes are, at least in part, defined by the rest of the face. This is especially true in reduced quality or abstract images. Thus, if one wishes to verify that an object is indeed an eye, they can determine the context in which it appears. If another eye appears next to it, and a nose and mouth appear in roughly their proper positions below the eyes, then one can be reasonably certain that he is looking at an eye, or in fact a face. Second, there is an inevitable need to perform a contrast expansion on the facial image to remove noisy facial shadows. In Russel's original algorithm the initial contrast enhancement is based on the analysis of the intensity data contained in the region whose top and bottom boundaries are defined by the top of the eyes and tip of the nose, respectively. The left and right boundaries can be defined roughly by vertical lines through the left and right eye pupils. Although there does not appear to be an exact correspondence, the location of the maxima for the two peaks in the eye signature are consistently close to the positions of the pupils. In addition, the row corresponding to the top row of the 8 by 64 "eye" window is usually close to what may be called the top of the eyes. Thus, the only bit of information remaining to be determined, in order to apply Russel's contrast expansion, is the location of the tip of the nose (hereafter referred to as the "top of nose").

Using essentially the same technique as was used to develop the eye signature, a vertical facial signature, hereafter referred to as the nose/mouth signature, was developed. In this case, a vertical window (slice) is extracted from the original facial image and placed at the extreme left side of a new 64 by 64 image (reference figure 3-5). That part of the facial image from a vertical line about the center of the left pupil to a vertical line about the center of the right pupil defines the region of interest. Once this sub-image has been placed at the extreme left side, the remaining columns of the 64 by 64 image are filled with a grey level intensity of 12. The resulting image is then transformed into the nose/mouth signature using essentially the same calculation as shown in equation 3-1. The only difference here is that now a gestalt is calculated for each row instead of each column. Figure 3-5 illustrates examples of the gestalt patterns obtained using unaltered original images. The approximate position of the top of the eye window (the window that verifies the presence of the eyes), from a previous analysis of the full facial image, is shown as a solid horizontal line in the gestalt pattern. An investigation of the pattern below the "eyes" reveals that in every case there is a significant increase in pixel intensity (corresponding to the bright area immediately below the eyes), followed by a significant decrease in pixel intensity which corresponds to the darker areas about the top of the nose. Once the location of

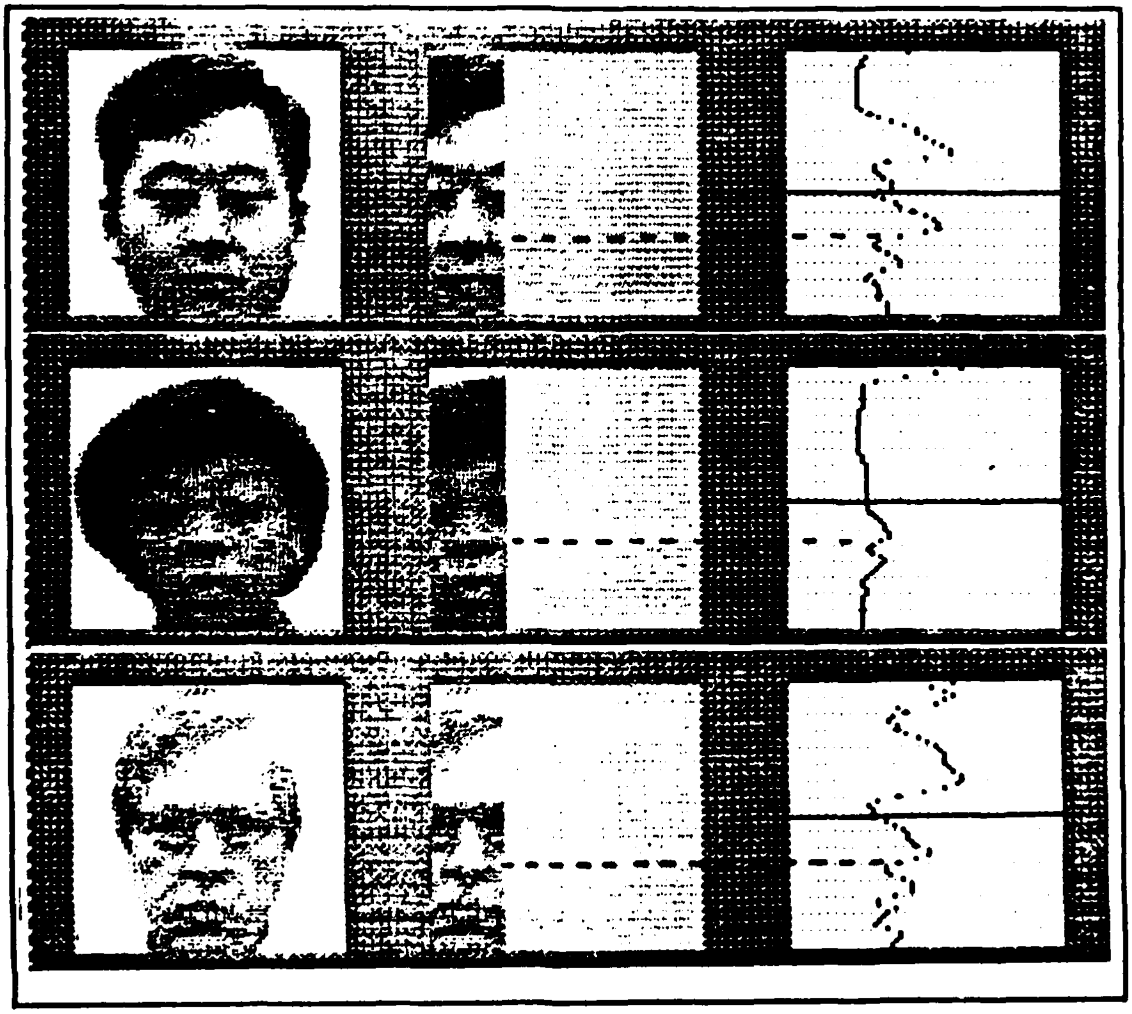


Figure 3-5. Typical Nose/Mouth Signatures.

maximum brightness (below the eyes) has been determined using a simple sort routine, the top of the nose can just as easily be determined. The top of the nose is defined as the point at which the average of the differences between the next three consecutive gestalt points is less than or equal to zero. For the purposes of this calculation, as one proceeds from left to right in the gestalt patterns of figure 3-5, the value of the gestalts increase in value. Applying this definition in a search routine which begins at the point of maximum brightness (determined previously), and proceeds down the nose/mouth signature curve (see figure 3-5), results in the "top of nose" locations illustrated by the dashed lines in figure 3-5.

Combining the results of the eye signature with the results of the nose/mouth signature, the information thus far gleaned from the original, unaltered image is as follows:

1. A good approximation of the vertical location of the eyes within the 64 by 64 image. Given by the location of top row of the eye window.
2. The location of the horizontal center of the face. Given by the location of the central minima in the eye signature.
3. The sides of the eyes. Given by the minima located to either side of the two peaks in the eye signature.
4. A good approximation of where the eye pupils are,

given by the location of the maxima of the two peaks in the eye signature.

5. The top or "tip" of the nose.

The only remaining major internal facial feature that remains to be located is the mouth. The center of the mouth (in the vertical direction) is usually characterized by a sharp transition from a region of low luminous intensity (corresponding to the upper lip) to one of high luminous intensity (the lower lip) under normal lighting circumstances. Although this transition is perceptable in the gestalt patterns of figure 3-5, it is just barely so due to noisy shadows around the mouth area. To overcome this problem and to enhance the overall accuracy of the feature locations, a slightly modified version of Russel's initial enhancement routine (22:5-32) was used. In synopsis, Russel's enhancement technique stretches the intensity histogram of an image by multiplying all pixel intensity values by a constant. The multiplier constant is determined by iteratively multiplying all pixel intensity values, from a sampled region of the face, by test multipliers until a desired average pixel intensity is obtained. The region of interest is a box-like region centered about the nasal area of the face. As mentioned earlier, the only parameter missing up to this point in the development, was the top of nose location. Once the nose location is known, the boundaries of the box can be completely defined relative to the size of the face and the location of the facial features. The left and right boundaries of the box are defined by the position of the left and right maxima for the peaks in the eye signature. The top boundary of the box is defined by the position of the top row of the eye window and the bottom boundary is defined as the top of nose location. It is then a simple matter to apply the enhancement as developed by Russel. Figure 3-6 illustrates the signatures obtained when the entire facial image is enhanced according to the modified Russel enhancement technique.

In figure 3-6, the center pictures contain the patterns for the eye signatures, and the pictures to the extreme right contain the patterns for the nose and mouth. The pictures to the extreme left contain the subject face which has been marked to indicate the feature locations found by the computer. The windows used to generate the eye and nose/mouth signatures (not shown) are the same as described before (ref. figures 3-4 and 3-5). In figure 3-6, the signature for the mouth is more readily apparent. The peak corresponding to the lips of the mouth is the next peak below the one for the nose. The reader will recall that the peak for the nose is directly below that for the eyes (marked by a dark solid line corresponding to the top row of the eye window).

It is important to note that for these images (figure 3-6), the feature locations are found automatically, with no assistance from a human operator. The position of the eye

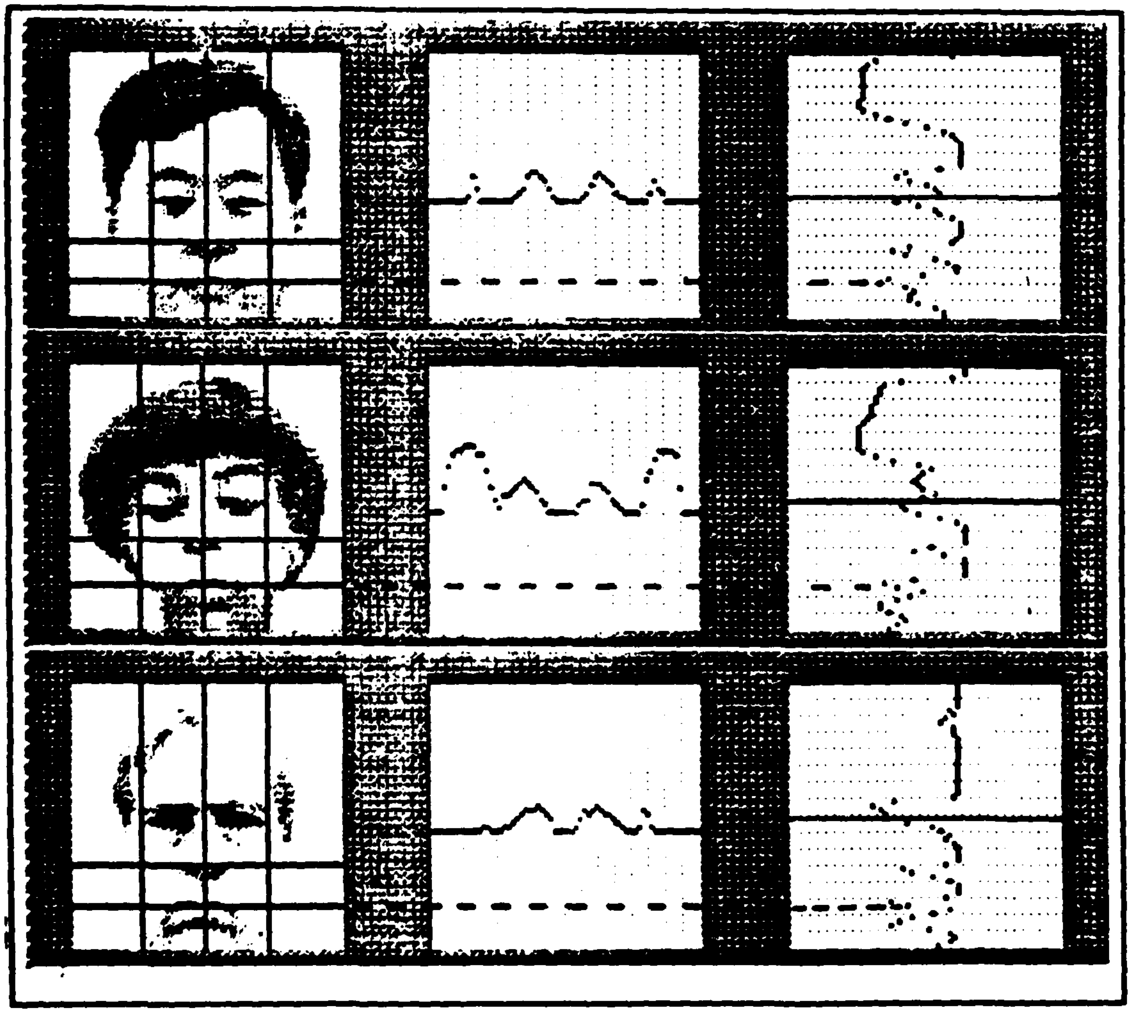


Figure 3-6. Typical Signatures for Enhanced Faces.

window is determined automatically by scanning the facial image (from the bottom up) until an eye signature meeting the characteristics outlined previously is detected. A nose/mouth signature is then generated using the appropriate data from the eye signature. These initial feature locations are determined from the original unaltered image. The facial image is then enhanced, as described previously, and all feature locations (including the mouth location) are determined based on the enhanced image.

The mouth location is determined by evaluating the nose/mouth signature, in a specified range below the nose, for the point of maximum negative slope. The center of mouth location is then taken as the point immediately preceding the point of maximum negative slope. These points are indicated in figure 3-6 by dashed lines. Although the technique of locating the mouth is very simple and straight-forward, the technique does appear to work in the large majority of cases. This method also works in many cases where the subject has a mustache or beard. Further examples of eye and nose/mouth signatures are contained in appendix C.

Searching for a Face. The facial images used in developing the eye and nose/mouth signatures were full face, pristine background, 64 by 64 digital images. These images were obtained from an online data bank of facial images. The procedure used, in searching for the signatures in these full face images, is essentially no different when applied to the larger problem of locating faces in an image where the faces comprise a much smaller percentage of the total image, and where the background can be anything. The designer simply has to make allowances, in the software, for such aspects as a the analysis of a larger image, the presence of more than one face and other such aspects.

Most of the design decisions made at this point were dictated by the practical use of limited computer resources (such as memory), and the desire to have the algorithm perform in at least a "near real-time" manner. The following is a list of the performance constraints imposed:

1. The total size of the search area was limited to a 128 by 192 pixels.
2. The range of facial sizes allowed was from a max size approximately 25% larger than the average face in the data bank, to a min size approximately 30% smaller than that in the data bank.
3. The maximum number of faces that can be processed in any one image is four.
4. Starting at the bottom of the image, and working its way up, the search routine analyzes only every other row for an eye signature. The routine does not analyze every row.

The process of searching for a face is begun by extracting a large central portion of the image displayed on the monitor. The size of the image, as indicated in the above list, is limited to an area of 192 pixels wide by 128 pixels long. This size limitation is due to dynamic memory limitations of the computer, a desire to limit the complexity of the computer code and a desire to limit the time required to process any one picture. This 192 by 128 image is hereafter referred to as the "search area". The location and bounds of the search area is illustrated by the large white rectangular box cursor in Figure 3-7.

Image analysis is begun by extracting the bottom 8 pixel rows of the search area and transforming this 8 by 192 window into a series of gestalt points. The transformation is conducted in the same manner as was described in generating the eye signature. The only difference is that now the gestalt pattern generated consists of 192 instead of 64 discrete points. The array of gestalt points is then analyzed to detect the presence of an eye signature using the criteria set out in the previous section on development of the eye signature. If no eye signature is detected in the gestalt array, the 8 by 192 window is incremented up 2 pixel rows and the whole process is repeated. This process is continued until an eye signature is detected or the search area has been completely scanned. The reason the window is incremented 2 rows, instead of one row, is that the time to process the entire picture is effectively cut in half. As long as the allowed facial image size is not too small,

Figure 3-7. Test Image Search Area.

there is little chance that the eyes will be missed by incrementing 2 rows instead of one row.

If, while scanning the search area, an eye signature is detected, then a 64 by 64 sub-section of the original image (as displayed on the moniter) is accessed and further analyzed. The white box cursor in Figure 3-8 illustrates the 64 by 64 image sub-section which is extracted from the original image. This method does not require that the full facial image be contained within the search area. It only requires that both eyes be contained within the search area.

Analysis of the 64 by 64 image sub-section proceeds as follows. The existence of the eye signature is re-verified and the location of the vertical features (center of face, etc.) are determined relative to the 64 by 64 image. The top of the nose in the unaltered image is determined using the nose/mouth signature, as discussed previously. The image is then enhanced using the modified Russel expansion technique. The enhanced image is reprocessed, once again, to determine all the feature locations (except the mouth). This is done to obtain greater accuracy in the feature locations. The original image is enhanced a second time, using the modified Russel enhancement, except this time the bounds of the box area used for enhancement are those determined from the enhanced picture (not the original). Using this second enhancement, a final determination of all the facial feature locations, including the mouth, is made. If, at any time

Figure 3-8. 64 by 64 Image Sub-Section Extraction.

during this process, it is found that a facial feature no longer exists or no longer falls within acceptable bounds, the process is terminated and the operator is informed that a face was not found in this image sub-section. If a face is found, that is, all features exist and are located in appropriate locations, then the image along with other pertinent data is saved to disk and the operator is informed of the presence of a face within the image sub-section. In addition, when a face is found, that face is marked on the moniter with a line pattern depicting the feature locations found. Thus in figure 3-9, the solid lines on the facial image depict the center of face, sides of eyes, top of nose and center of mouth. In addition, the horizontal line immediately above the eyes corresponds to a location determined by adding the difference between the two maxima (found in the eye signature) to the top of nose location. Adding twice that difference to the top of nose location, results in the line at the top of the forehead. The area within the lined gridwork on the face, comprises the area of interest referred to previously as the "internal facial features". This area is used in further analysis to determine the identity of the individual.

Whether or not a face is found, once analysis of the image sub-section is completed, the algorithm returns to the search routine it was involved in prior to accessing the image sub-section. If a face was found, then the bounds of a

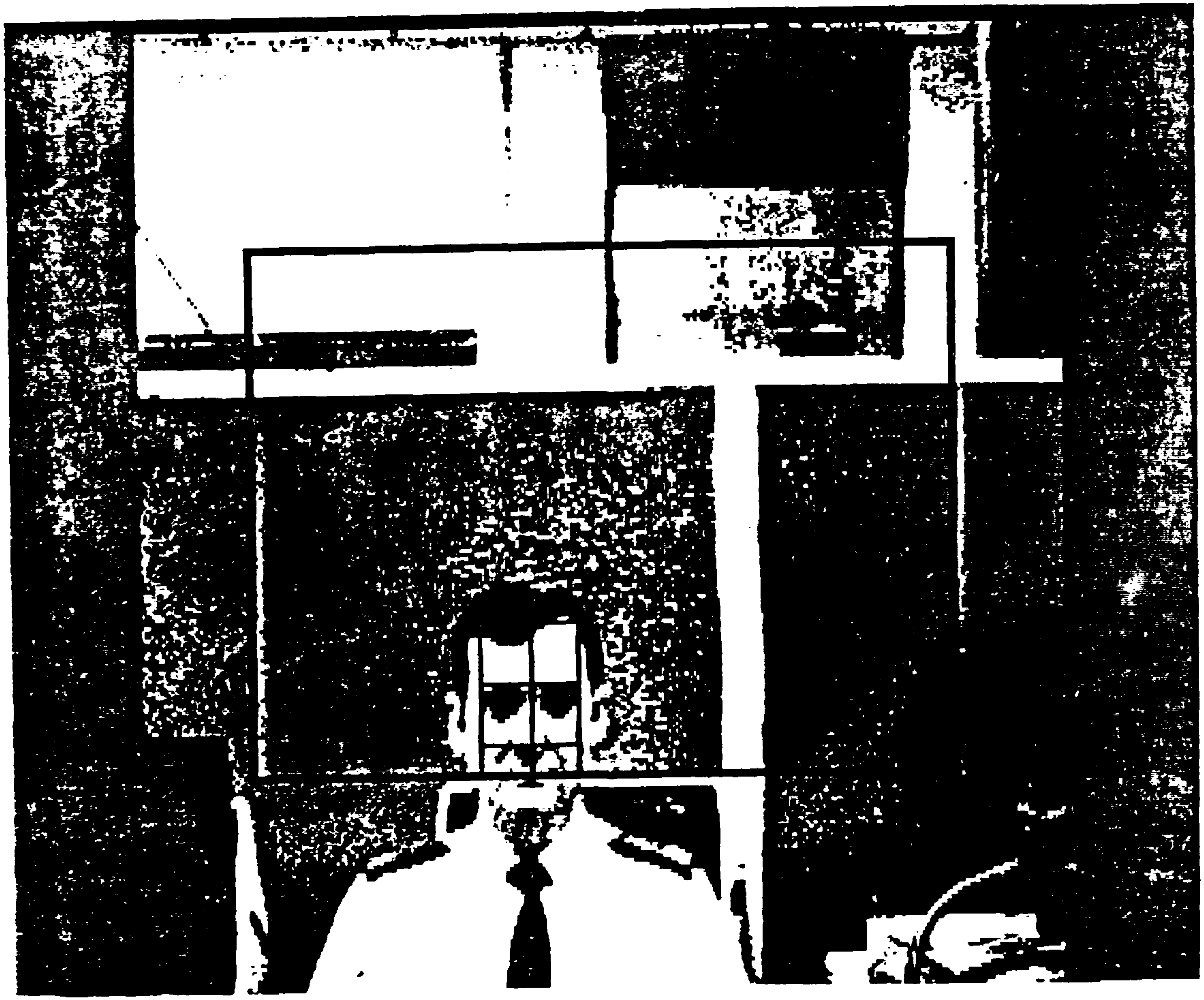


Figure 3-9. Confirmation of Facial Features.

region encompassing the area of the face are stored in memory so that the face will not be "re-found" on a subsequent pass. The search continues from the point it left off at, and proceeds from the bottom of the search area to the top until four facial images have been found or the search area has been completely scanned. The choice of a four face limit was arbitrary. Again, keeping the number of possible faces limited, keeps the speed of performance and the code complexity at a reasonable level.

The Face Recognizer

This thesis assumes that the basic recognition algorithm developed by Russel is valid, and only requires minimal modifications to make it perform in an autonomous fashion. Only the required modifications are discussed in this thesis. For discussion concerning the theory and design of items that are the same as used by Russel, the reader is referred to the appropriate sections of Russel's thesis (22).

Window Generation. The major difference between the face recognition algorithm used by Russel and the algorithm used in this thesis is in the bounds of the facial windows. The windows used for face recognition in this thesis must be constructed from the internal facial features only. In Russel's work, additional facial boundaries were available, such as the sides and top of the head and bottom of the chin. These additional boundaries could be easily determined provided a pristine background was always used. However, in the interest of designing a more universal machine, it was decided to forego the use of these boundaries. As expected, the loss of the information, such as hairstyle, obtained when using the additional boundaries, results in a decrease in the overall recognition performance (see chapter 5). However, it is this author's firm belief that this drawback can be overcome. Indeed, there is ample evidence to show that, at least in the case of familiar faces, recognition is predominantly dependent on analysis of the internal facial features (7;26).

The choice of windows used in this design is illustrated in figure 3-10. The picture in the upper left corner of figure 3-10 is the original image sub-section which was extracted automatically by the face finder algorithm (see figure 3-8). The center upper picture is the same image enhanced based on the final contrast multiplier constant found during the face location process. The picture in the upper right hand corner is the enhanced image with the internal feature boundaries marked.

The windows indicated in the pictures in the middle (row) and left (column) and the middle center positions both have the same top and bottom boundaries. The bottom boundary is the center of mouth location and the top boundary is the boundary formed by adding twice the difference between the maxima (taken from the eye signature), to the top of nose location. The vertical boundaries for these two images are

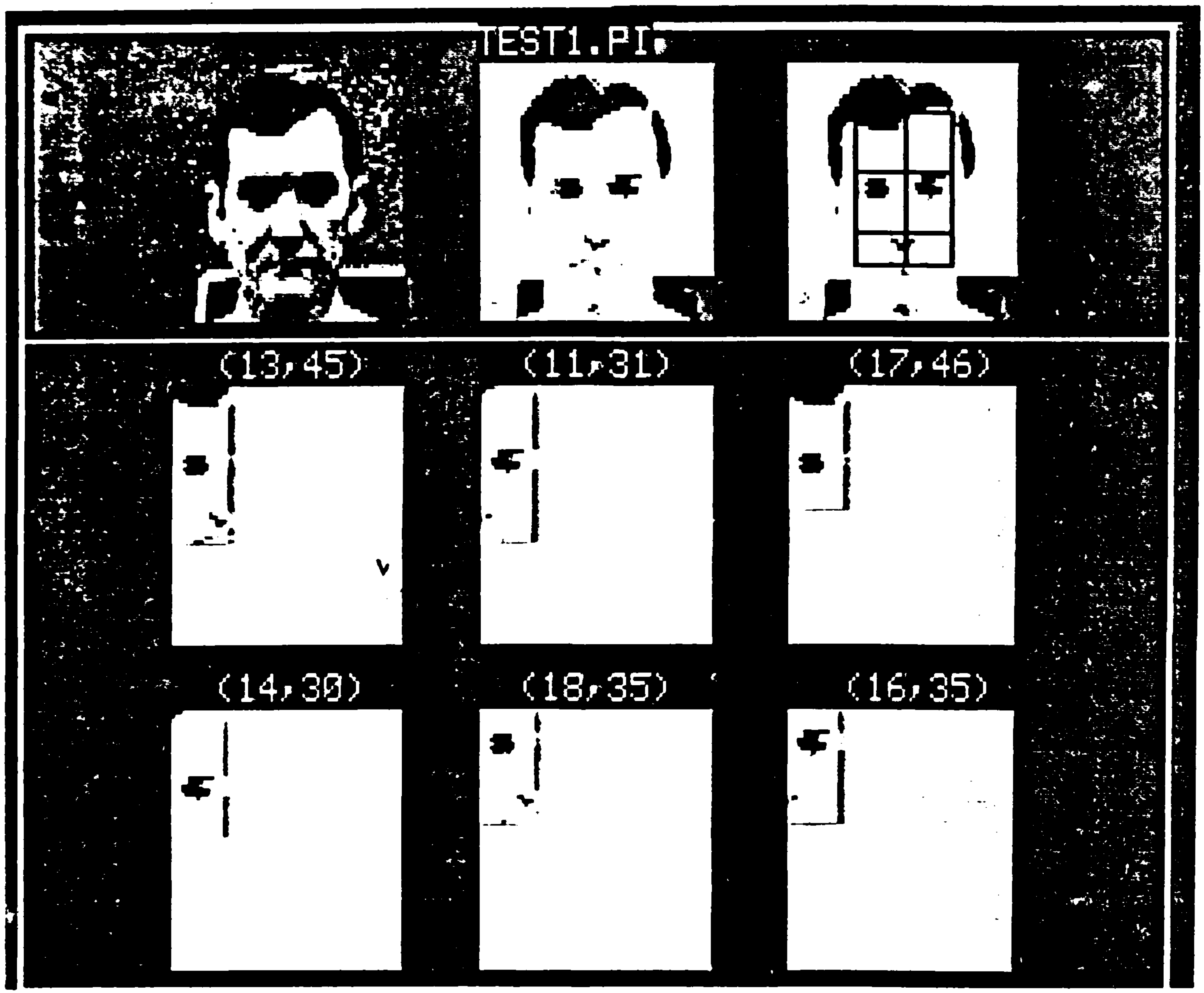



Figure 3-10. Windows Used in Face Recognition.

the center of eyes and sides of eyes. Similiarly, the windows displayed in the pictures at the middle right and bottom left positions have the same boundaries, except that the top of nose location now forms the bottom boundary. The windows depicted in the pictures located at the bottom middle and bottom right positions also have the same vertical boundaries (sides and center of eyes). Their bottom boundaries are determined by the center of mouth. The top boundaries for these two windows are obtained by adding the distance between the sides of the eyes to the center of mouth location.

The selection of the windows represents a rough, first cut at the problem. Time constraints did not allow an exhaustive development of "optimal" windows. The selection was based on a subjective analysis of those boundaries that might give the best discrimination in the gestalt feature space.

Gestalt Calculation. The gestalt values for the windows shown in figure 3-10 are calculated in exactly the same manner as was done in Russel's algorithm. The reader is referred to appendix A and Russel's thesis (22:5-42) for further information on how this calculation is performed.

Training for a Face. In order for the system to identify an person, it must be "trained" for that person. As discussed in the introduction to this thesis, training the system consists of processing the facial images of an individual to obtain characteristic gestalt values associated with that individual. These gestalt values are then stored in the systems memory for later recall and comparison with the gestalt values of an unknown individual. If the stored gestalt values match closely with that of the unknown individual, then the system usually is successful in properly identifying the unknown individual.

Since faces are extremely variable (even with respect to internal features), it is necessary to train the system with more than one characteristic facial image. The guideline of training with four pictures for each person, as established by Russel, was also used in this thesis.

The process of training the system is the same process as was described in the preceding sections on searching for a face, window generation and gestalt calculation. Once the facial images have been located, the operator may choose to either train with the facial images that were found or identify the facial images. If the operator chooses to train, then, as the window gestalt calculations for each image are completed, the operator inputs which person these parameters pertain to, then the next image is processed. Figure 3-11 illustrates a sample set of images used to train the system for the test subject of figures 3-7 through 3-10.

Identifying the Face. If the operator desires to identify the faces found by the face finder algorithm, the system attempts to identify the faces in the order that they were found. First, the windows and gestalt values for each

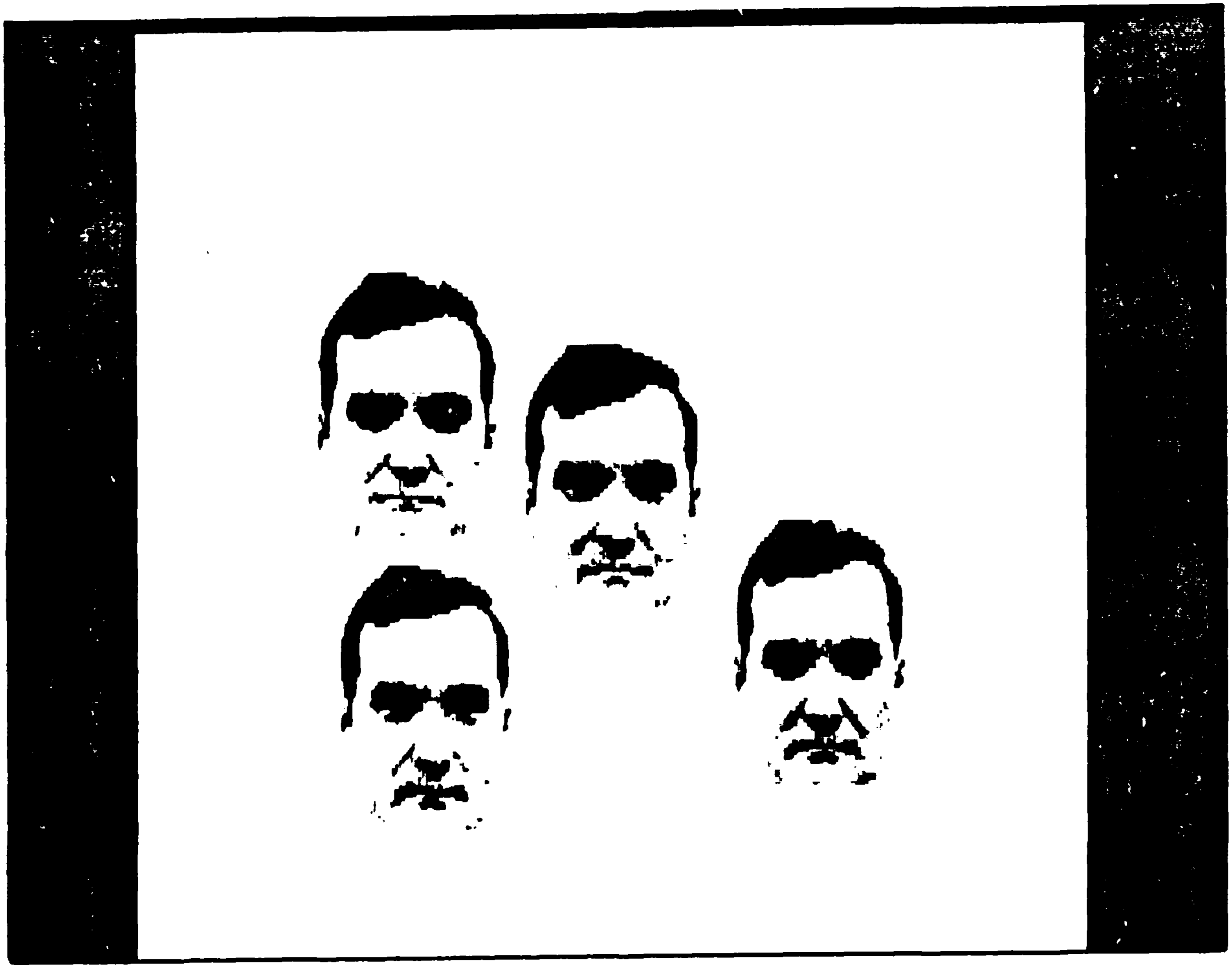


Figure 3-11. Facial Images Used for Training.

image are generated (as in figure 3-10). The system then compares the gestalt values obtained, with those for which it has been trained. The individual that most closely matches the gestalts for the unknown person is chosen by the computer as the number one candidate. The recognition algorithm rank orders the candidates and computes a pseudo-probability measure to indicate how certain it is that the candidate chosen is the proper one. Again, the algorithm used to perform recognition is an unmodified algorithm developed by Russel. The reader is referred to appendix A and Russel's thesis (Appendix A:A-7;22:4-32) for a more detailed explanation of the selection process. Figure 3-12 (on the next page) illustrates the final result of the autonomous face recognition machine algorithm.

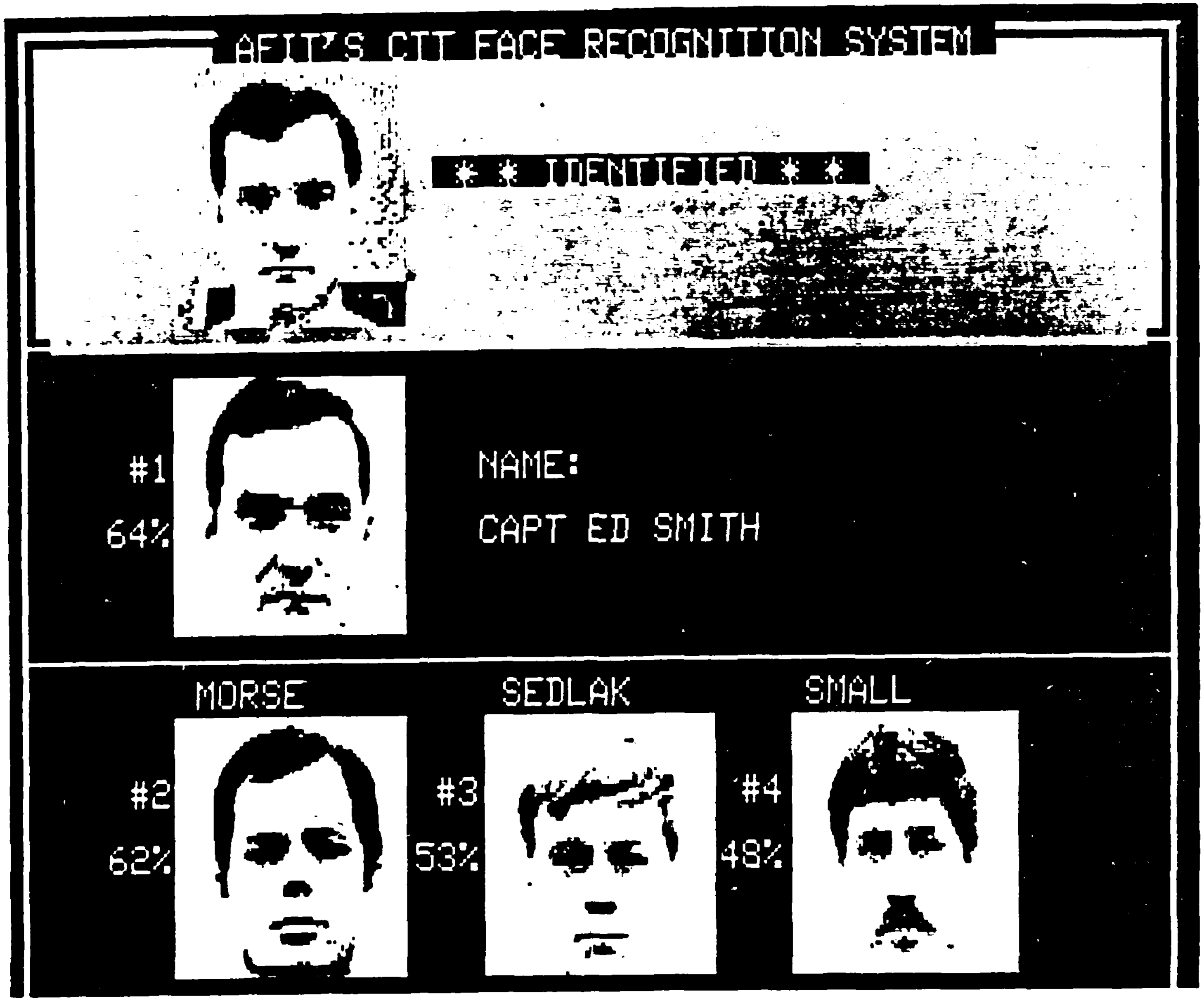



Figure 3-12. Identification of a Person.

# IV. Implementation

This chapter explains how the design discussed in chapter 3 was implemented on the systems available at the AFIT Signal Processing Laboratory. The first section discusses the hardware and software environment used. Subsequent sections discuss the software implementation of the major functions used to locate and identify faces.

## System Environment

The computer used to develop, implement and test the face locator and face recognizer algorithms was a Data General computer system. This system is comprised of two separate computers:

1. A Data General Eclipse S/250, 16-bit minicomputer.
2. A Data General Nova II, 16-bit minicomputer.

The Eclipse computer is the more powerful of the two computers and is primarily used for calculations. The Nova computer, while not as fast or powerful as the Eclipse, is capable of interacting with a video digitizing system and is primarily used to acquire and display image data. The two computers are linked by common disk storage areas and can communicate via "flag" and/or data files which can be created or deleted (by either computer) in the common disk storage area. The amount of user available core memory in either system is approximately 28K bytes, at any one time. This memory limitation mandated the extensive use of such techniques as program overlaying and swapping in order to perform the required algorithms.

The video digitizing system used in conjunction with the Nova computer is an Octek 2000 Image Analyzer Card (IAC). This digitizer system is capable of acquiring or displaying a 320 (horizontal) by 240 (vertical) pixel image. The Octek processes only monochrome (black and white) images. The number of grey levels available on it is 16. The video camera used with this system is a Dage 650 camera which is equipped with an adjustable F-Stop (F2.5-F16) and a zoom lens (18-108mm). The overall system configuration used is identical to that used by Russel (22) and is illustrated in figure 4-1.

The face locator and recognition algorithms were implemented in the Fortran computer language. Specifically, those programs which execute on the Nova are compiled in Data General Fortran IV, and those which execute on the Eclipse are compiled in Data General Fortran V. These Data General Fortran's conform to ANSI standards, thus aiding program transportability from one machine to another. However, the extensive use of library functions in the programs, both Fortran and IAC library functions, may make the task of transporting the software quite difficult.

Top Level Programs

Each computer (Nova and Eclipse) has a top level manager program which is initiated by executing a macro file on the desired system. On the Eclipse the macro file is "FINDFACE.MC" and on the Nova it is "AUTOFACE.MC". These

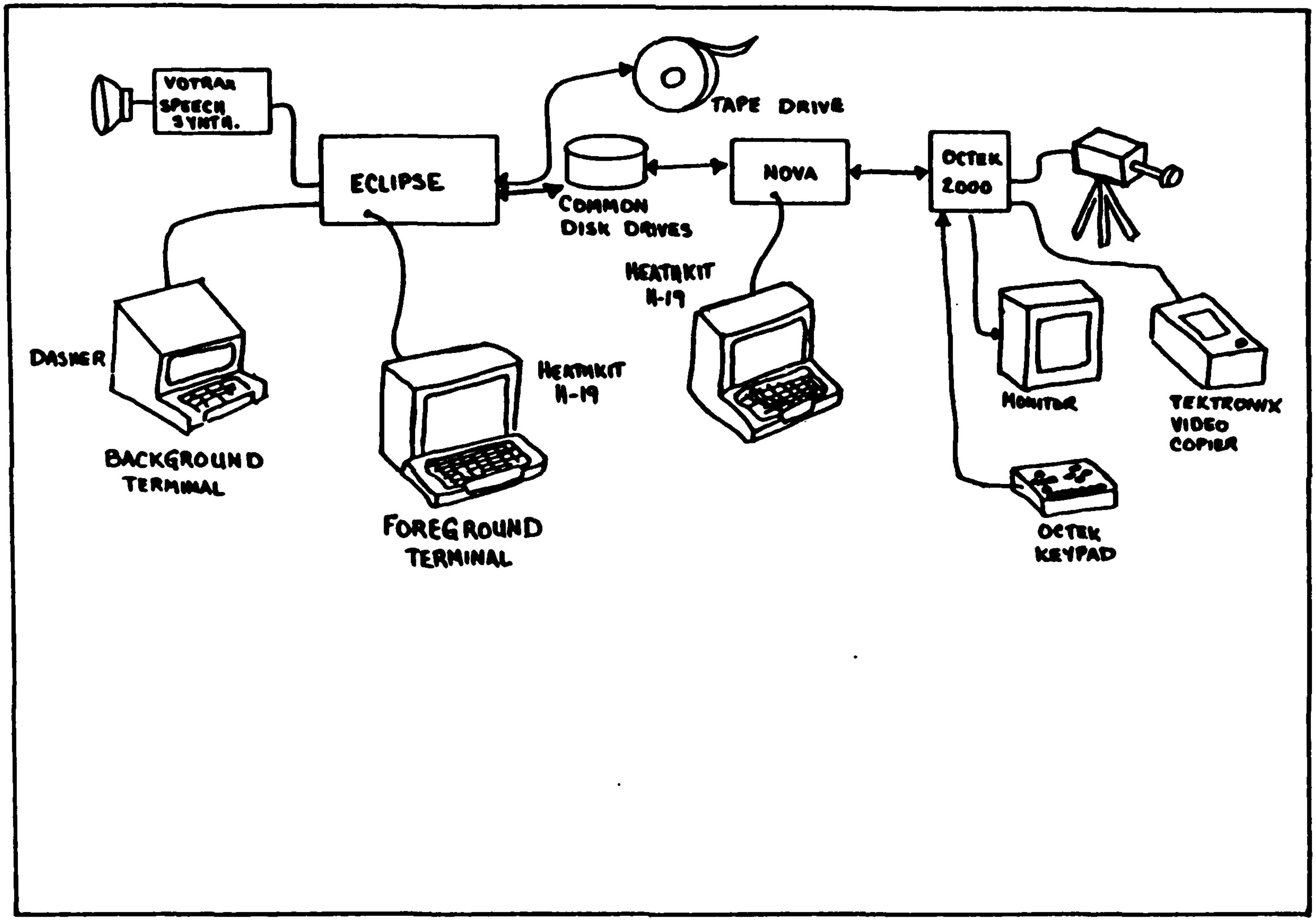



Figure 4-1. System Configuration. (22:5-3)

macro's ensure that the system is initialized correctly before bringing up the top level manager programs. On the Eclipse, proper initialization means ensuring that any needed directories are initialized and any old data or flag files have been deleted. Once this has been accomplished, the top level program "FACEFNDR" is executed on the Eclipse. The Nova is initialized in the same manner, plus the Nova ensures that the program FACEFNDR has been successfully initiated on the Eclipse. To verify that the Eclipse is ready, the Nova checks the common disk area directory (directory NSMITH) for the presence of the flag file "GREENLIGHT". The file GREENLIGHT, is a "flag" file because it is a file which contains no data (a 0 length file). This file is created by FACEFNDR when it begins execution. By virtue of its existence in the directory, the file GREENLIGHT is a signal to the Nova that the Eclipse has successfully performed the FINDFACE initialization routine. Many such flag files are used in this implementation to allow communication and task synchronization between the Eclipse and Nova systems. If the system is to perform correctly, careful housekeeping of these flag files is paramount. Thus, once a flag file has served its purpose, it is immediately deleted from the directory.

When verification of Eclipse initialization has been accomplished, the Nova top level program "GETSUBJ" is executed. Both top level programs (FACEFNDR and GETSUBJ) execute in a continuous loop until receipt of a command from the operator to terminate. The program FACEFNDR is a synchronized program which responds to signals (in the form of flag files) from the Nova. Thus, once the system has been initialized all operator commands are given from the Nova terminal. Figures 4-2a and 4-2b illustrate the top level program flow for both the Eclipse and Nova systems.

Image Acquisition

In order to process an image, the desired image must be loaded into the Octek digitizer. There are two ways to accomplish this. One way is to acquire the image through the use of the video camera. This can be accomplished directly through GETSUBJ by selecting the program option to activate the camera, setting up the picture, and then selecting the program option to turn off the camera. Turning off the camera, freezes the image in the Octek. Another way to load the image in the Octek, is to use the program "OCTEK" to load (and display) a previously saved image.

Once the desired image is displayed on the monitor screen (and thus loaded in the Octek's frame buffer), selection of the GETSUBJ program option "process picture", initiates the acquisition and anaylsis of the image search area. The reader will recall that the image search area is a 192 by 128 pixel area extracted from the central portion of the octek image (see figure 3-7). GETSUBJ stores the search area image data, to the common disk directory "NSMITH", as a file named "SUBJECT.VD". Analysis of the image data (by FACEFNDR on the Eclipse) is then initiated by the creation

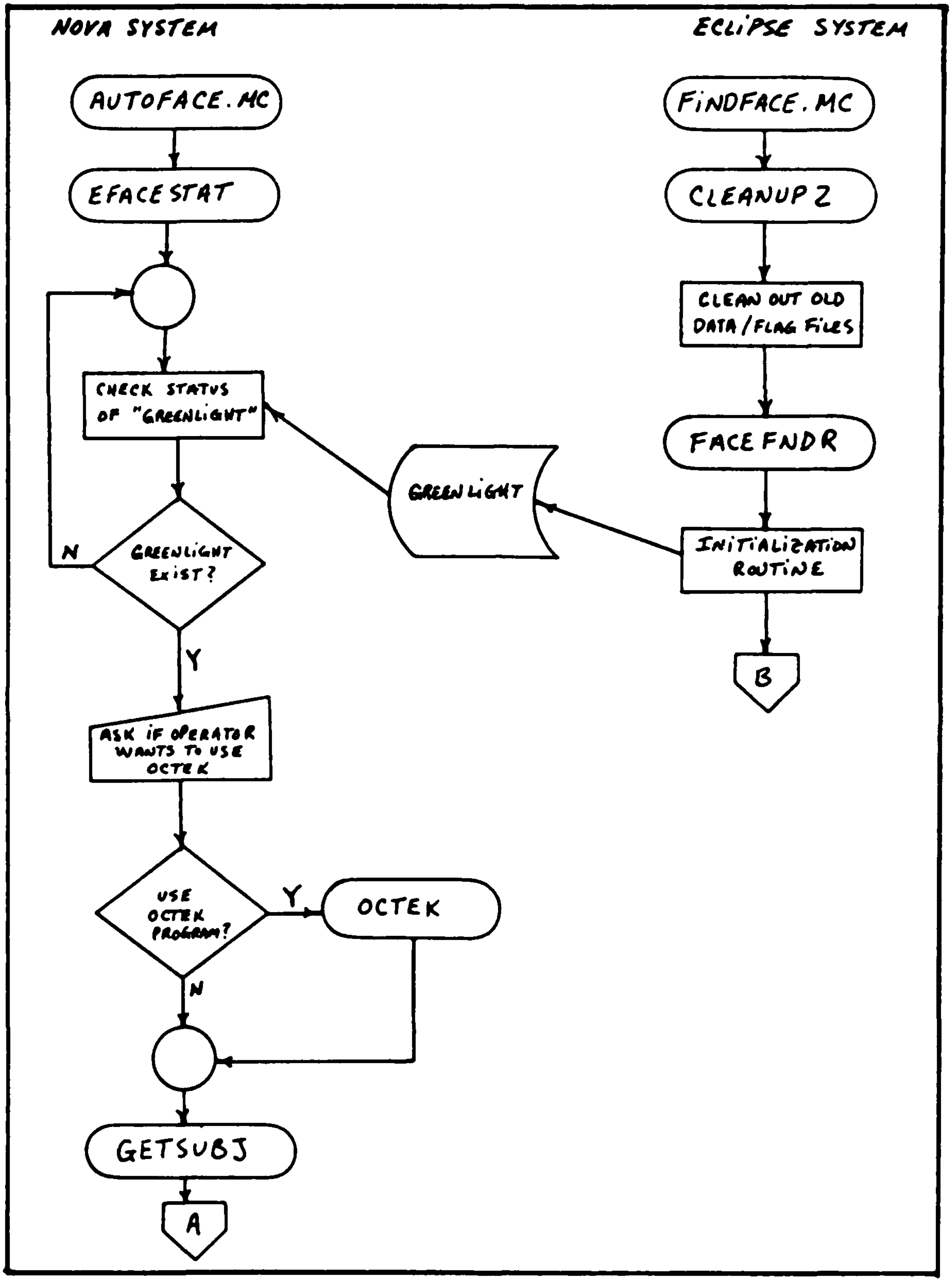



Figure 4-2a. Top Level Program Flow.

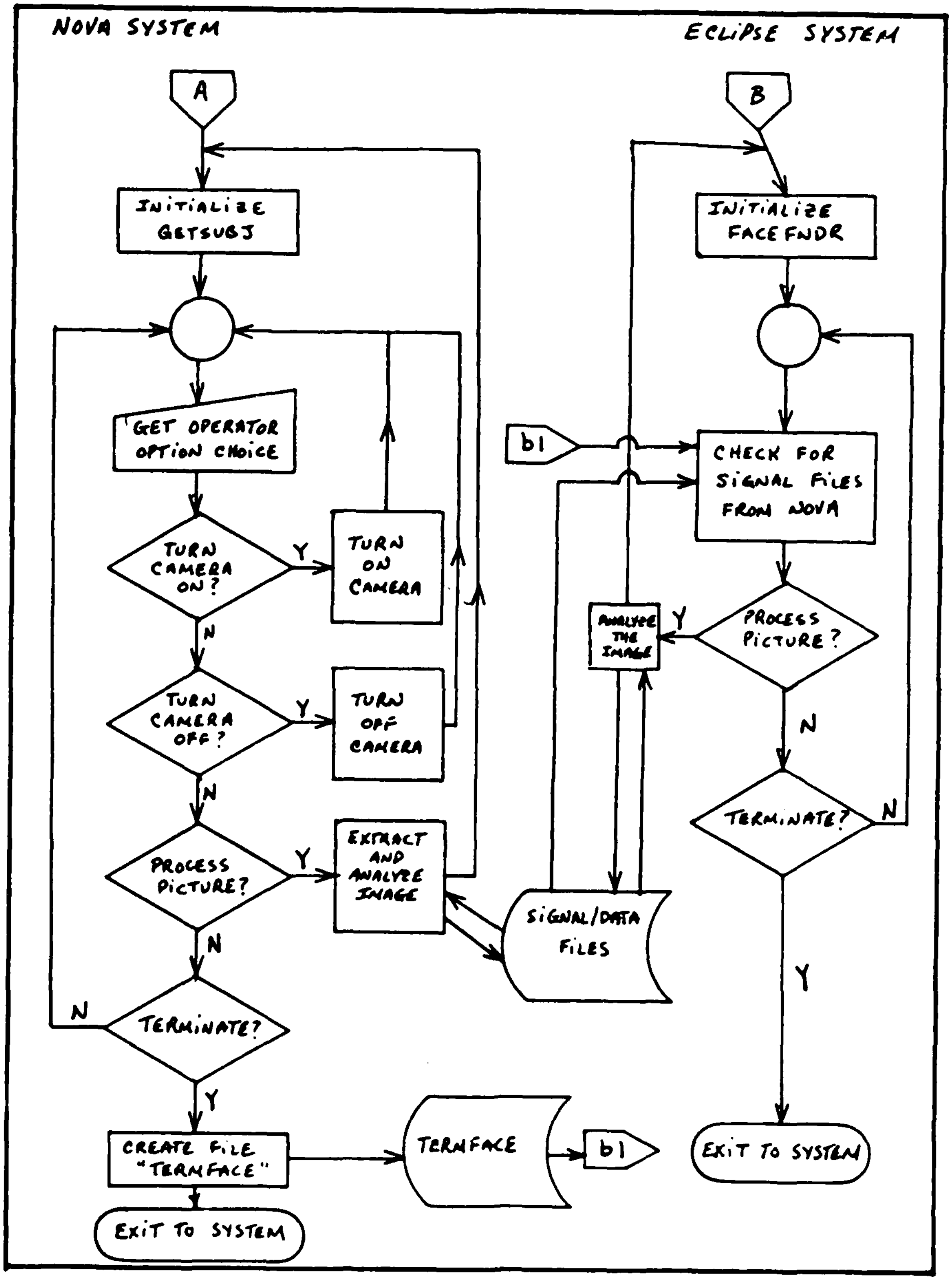



Figure 4-2b. Top Level Program Flow (continued).

of the flag file "SUBJECT1.A". The high level flow chart for the "process picture" option is illustrated in figures 4-3a and 4-3b.

Image Processing to Find Faces

Once the program FACEFNDR has been initiated on the Eclipse, it continuously checks for the existence of various flag files in directory NSMITH. Upon detecting the existence of the flag file SUBJECT1.A, FACEFNDR initiates analysis of the data in SUBJECT.VD, for the eye signature.

Starting with the bottom 8 rows of the image search area, FACEFNDR successively determines the gestalt value for each column of the 8 row by 192 column window. The gestalt values are determined in the same manner as discussed in chapter 3. To the bottom of each column of 8 pixels (from the window on the search area), 56 more pixels are added, to construct a single column of 64 pixels in total length. The intensity value assigned to these added pixels is 12. This single column of 64 pixels is then transformed via the gestalt calculation indicated in equation 3-1. The actual transformation is performed by the subroutine "RTRAM". The transformed column is then evaluated by FACEFNDR to determine the location of the row containing the maximum value. The location of the maximum is then stored as the gestalt value for that column. The determination of the gestalt values proceeds in this manner, column by column, until all 192 columns have been processed. The gestalt values are saved in the array "STRIPDATA" which is a 400

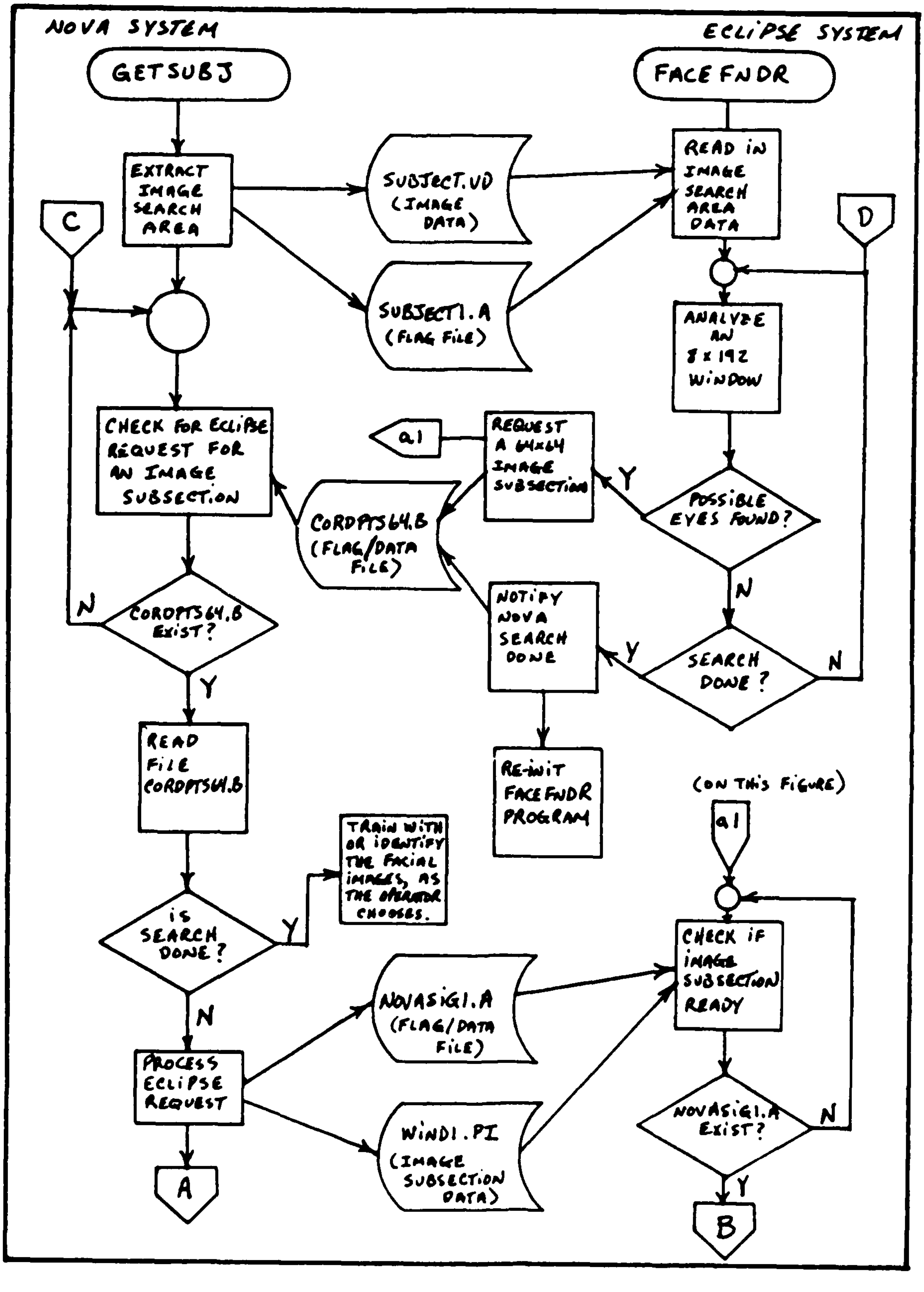



Figure 4-3a. Picture Processing Flow.

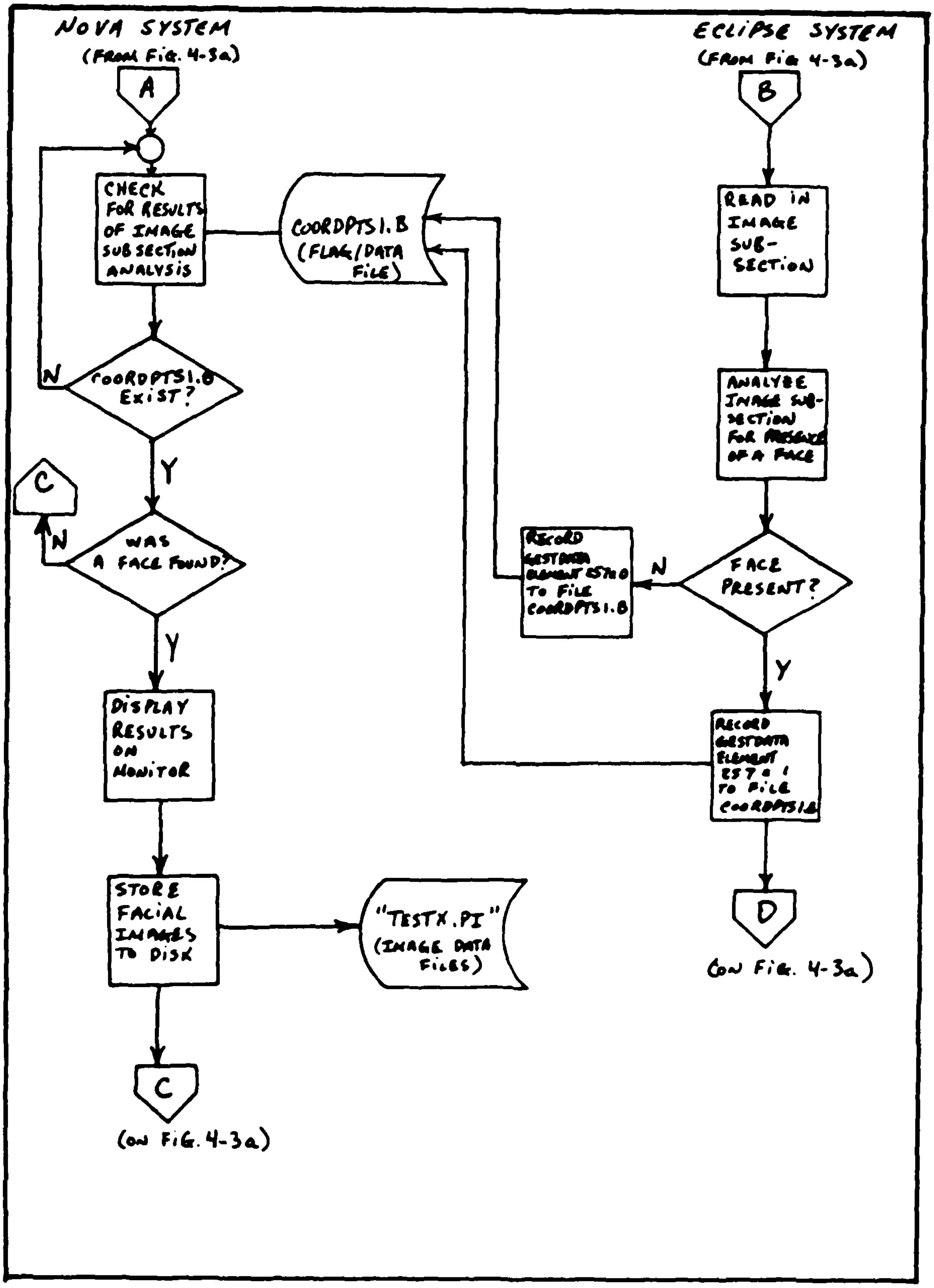



Figure 4-3b. Picture Processing Flow (continued).

element integer array. The first 384 elements of this array are reserved for gestalt location data with odd numbered elements containing the row location and even numbered elements containing the column location of the gestalt point. The column locations are arranged in increasing order within the array. Thus, element 2 of the array contains the number "1" to indicate that the row value in element 1 of the array is associated with column 1. Element 4 of the array contains the column number "2", element 6, the column number "3", etc.

Once the STRIPDATA array has been filled with the gestalt data generated from the 8 by 192 window, this data is passed to the subroutine "EYEF" which analyzes the data for the presence of an eye signature. In order to reduce the problems due to noise in the signature, the data in STRIPDATA is smoothed by convolution with a gaussian function. To perform this operation, the "magitude" of the gestalt must be defined. This magnitude is defined as the difference between the gestalt row value for any given column and a baseline row value equal to 36. The baseline value is the gestalt row value obtained when all 8 pixels (in the window) are white. Using this definition for the magnitude, the smoothed gestalt values for any particular column (at location "N") are determined by adding, to the magnitude at the column of interest, 0.7 times the magnitude of the gestalts for the columns to either side (at locations N+1 and N-1), and 0.3 times the magnitude of the the gestalts for the columns once removed and to either side (at locations N+2 and N-2).

A point by point analysis of the smoothed gestalt curve is then conducted to detect an eye signature. The analysis proceeds from left to right (in relation to the original 8 by 192 window), which corresponds to starting with column 1 and proceeding in increasing order. As each pair of successive peaks is detected, they are tested to see if they meet the criteria for an eye signature. Recall that the criteria for an eye signature are as follows:

1. Two peaks with maxima located equidistant from a central minima.
2. The outer slopes of the two peaks are not greater than 1.5 times the greater of the two inner slopes.

In reference to criterion 2 above, the respective slopes (illustrated in figure 4-4) are determined as the difference between the magnitudes at the respective maxima and minima locations, divided by the difference between the respective maxima and minima column locations. It is important to note how the minima locations are determined when they occur on a "plateau" or in a region where the magnitude is unchanging over several columns. In the case of the central minima, the minima location is taken as the half-way point along the plateau. For the left outer minimum, the location of the minimum is taken as the extreme right of the plateau. For the right outer minimum, the location of the minimum is taken as the extreme left of the plateau. In reference to criterion 1 above, the acceptable tolerance for the equidistant location of the central minima is set at 7.5% of the difference between the two maxima locations.

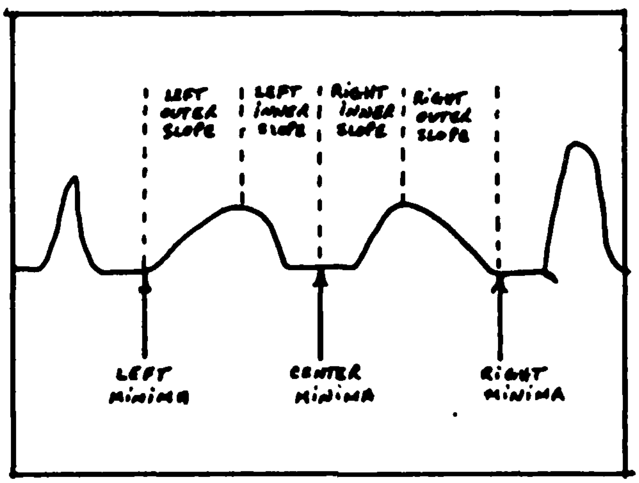



Figure 4-4. Eye Signature Slopes.

Depending on whether or not an eye signature is detected, EYEF updates element 385 of the STRIPDATA array to indicate that possible eyes were found (element 385=1) or no eyes were found (element 385=0). If eyes were found, then the locations of the minima and maxima are also recorded in STRIPDATA. The STRIPDATA array is then passed back to the calling program (FACEFNDR) for evaluation.

If a possible eye signature is not located by EYEF, then FACEFNDR proceeds to access the next 8 by 192 window of the search area. This window is shifted 2 pixel rows up from the first, and likewise, all subsequent windows are incremented up, 2 pixel rows at a time. This process continues until a possible eye signature is found or the search area has been completely scanned.

When a possible eye signature is detected by the subroutine EYEF, FACEFNDR verifies that the location of the eye signature is not within the facial boundaries of a previously found face. If it is, then FACEFNDR ignores the indication and proceeds with scanning the search area. If the eye signature lies in a region where a face has not been previously found, FACEFNDR initiates a request to the Nova to extract a 64 by 64 image subsection of the region of interest. FACEFNDR initiates this request by creating the file CORDPTS64.B which contains the location data necessary for the Nova to extract the proper sub-image.

Upon detecting the presence of the file CORDPTS64.B, the program GETSUBJ obtains the necessary information from the file, extracts and saves the appropriate 64 by 64 image sub-section to disk , and notifies the Eclipse that the image sub-section is ready. The flag file used to indicate that the image sub-section is ready for further processing, is the file "NOVASIG1.A".

When FACEFNDR detects the presence of NOVASIG1.A, it begins the process of analyzing the data in the image sub-section. The locations of the maxima and minima (from the eye signature) are first converted to locations relative to the 64 by 64 matrix. A nose/mouth signature is then generated using the "center of face" window as discussed in chapter 3 (see figure 3-5). In this case the individual row gestalt transformations are performed by the subroutine "RTRANSB". The resulting 64 "row gestalt" values are stored in the 300 element integer array "GESTDATA", beginning at element number 129. The data in GESTDATA are then analyzed to determine the presence and location of the top of the nose. If the nose is found, then the entire image subsection is enhanced and re-analyzed. Enhancement is performed by use of the modified Russel contrast expansion technique (discussed previously). The limits which bound the box-like sample region, upon which the enhancement is based, are given by:

1. Top Limit = top of eye window + 1/2 the difference between location of the eye signature left maximum and the center of eyes (boundary "a" in figure 4-5).
2. Bottom Limit = top of nose location (boundary "b" in figure 4-5).
3. Left Limit = eye signature left maxima location (boundary "c" in figure 4-5).
4. Right Limit = eye signature right maxima location (boundary "d" in figure 4-5).

The significant difference between this technique and the initial enhancement originally used by Russel, is that the enhancement does not require that the image subsection contain a standard size facial image, which is located in

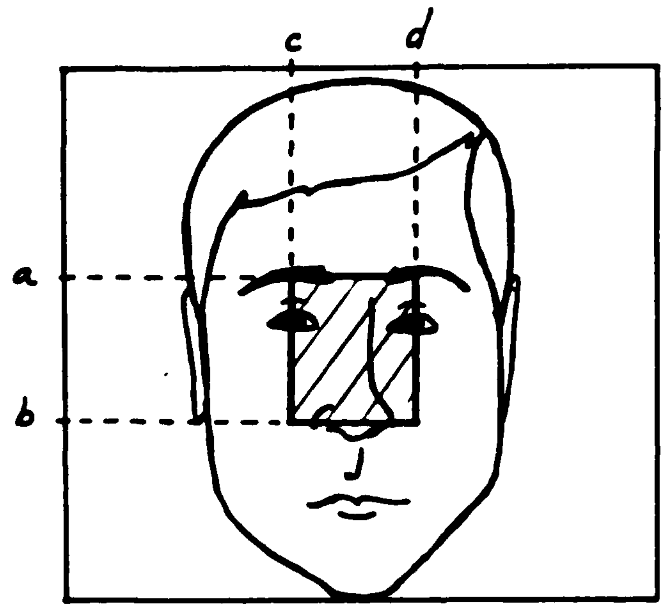



Figure 4-5. Facial Sample Area for Contrast Enhancement.

standard position within the image sub-section. In the "modified" Russel expansion, the location and bounds of the box-like region are determined based on the location of the face and the facial features. The only other difference is that the contrast multiplier is determined based on attaining an average pixel value (for the box-like region) of 14.5, and not 13.0. Otherwise, the contrast expansion is essentially the same as the original Russel expansion technique and the reader is referred to his thesis (22:5-32) for further details.

Re-analysis of the enhanced facial image starts with a determination of where the eye signature is now located. Starting at the vertical position where the eye signature originally occurred and proceeding upwards, the enhanced image is scanned with an 8 by 64 eye window to determine the new location of the eye signature. In many instances, the eye signature location changes because the enhancement process significantly reduces the shadows around the eyes. This causes the eye window to move further up on the facial image, before the eyes are detected. Actual detection of the eye signature is verified by the subroutine "FTHOR", which is essentially a clone of the subroutine EYEF. The only significant difference between the two subroutines is that FTHOR works on 64 pixel wide windows and EYEF works on 192 pixel wide windows. Once the eyes are found again, the process of determining the nose location is repeated.

When the nose is located, the original image is enhanced again (to an average pixel value of 14.5 for the box-like sample region), only this time the bounds of the box-like region are determined from the feature locations as determined from the enhanced image. This second iteration is performed to guard against the possibility that the first sample region was adversely affected by unusual facial shadows. It also improves the accuracy and consistency of the facial feature locations. Based on the second enhancement, the facial features are determined a final time and stored in the GESTDATA array.

With the exception of the subroutine calls indicated in the discussion above, the analysis of the 64 by 64 image sub-section is performed by "in-line" code in the program FACEFNDR. If at any time during the process, the program fails to find what it is searching for, or finds what it is searching for, but in an unacceptable location, the analysis of the image sub-section is terminated. If the analysis is terminated due to a failure, an "eye found" indicator (GESTDATA element number 257) is reset to "0", and the Nova is notified via the flag/data file "COORDPTS1.B", which contains the data from the GESTDATA array.

If, on the other hand, all facial features have been located, then GESTDATA (257) contains a "1". As before, the Nova is notified via the flag/data file COORDPTS1.B. Upon detecting the presence of COORDPTS1.B, the top level program on the Nova, GETSUBJ, reads in the file COORDPTS1.B, storing the values in its GESTDATA array. When it is determined that the value of GESTDATA(257) is non-zero, the operator is informed of a successful location of a facial image. If the facial image is small enough to contain the entire internal facial region in a 64 by 64 image, then the image is saved as TESTX.PI. Where "X" is a number from 1 to 4, depending on whether this is the first, second, third or fourth facial image found. If the bounds of the internal facial region are too large, then the operator is informed that a face was found but the facial image is too large to be processed by the system.

Along with the image file, GETSUBJ also saves the data indicating the facial feature locations and the final contrast multiplier as determined by FACEFNDR. GETSUBJ then displays (on the monitor) a set of grid marks which outline the location of the internal face features. Provided that this was not the fourth facial image to be found, GETSUBJ resumes a cyclic check for further processing requests from the Eclipse. Meanwhile FACEFNDR resumes its scan of the image search area for other locations containing eye signatures.

Generating Facial Gestalts

There are two normal ways in which the scan of the search area (by FACEFNDR) may be terminated. The first is by interrupting the process by striking the octek keypad #7, after a facial image has been found. This action generates the creation of the flag file "FACEDONE" by the program

GETSUBJ. This results in immediate termination of the scan procedure on the Eclipse. The second way the scan is terminated is by normal completion of the scan of the search area. In either case, the FACEFNDR program is reinitialized and returns to its main loop of checking for interrupts from the Nova.

On the Nova, the program GETSUBJ prompts the operator to determine whether or not he would like to train with the facial images, identify the facial images or just quit. If the operator chooses to quit, then GETSUBJ is reinitialized, and the system is ready to start over again. If the operator chooses one of the other two options, then the first thing that will happen (regardless of which option is chosen) is the generation of facial window gestalts for the first face image that was found.

The program GETSUBJ initiates the generation of the facial window gestalts by creating the data file "NOVASIG1" and the flag file "CALCGEST". GETSUBJ then executes the program GTGEST, on the Nova, by performing a program "swap". The use of program swapping is fairly extensive from this point on, so a brief word about this technique appears to be in order. When a program is swapped for another program, the calling program is saved to disk along with most of the parameters associated with it. The program being swapped in, which replaces the calling program, is then loaded into core and executed. When the current program completes execution, or it reaches a "CALL BACK" statement, the calling program is reloaded into core and resumes execution at the next executable statement, after the swap statement. This swap feature makes it possible to execute very lengthy routines in a very limited core memory environment. Thus, when the program GTGEST has completed, control returns to the top level program on the Nova, GETSUBJ.

The flag file CALCGEST is a signal to the FACEFNDR (on the Eclipse) to set up for calculating gestalt values. FACEFNDR accomplishes this by swapping in the program "CORTRAN16". CORTRAN16 is the same program used by Russel (22:5-30) for calculating the window gestalts. The data file NOVASIG1 contains instructions as to which facial image file (found and saved previously by the face locator process) is to be processed. When GTGEST begins execution, it examines the contents of NOVASIG1 to determine which facial image to process. The order of processing is always in increasing order from TEST1.PI up to TEST4.PI.

After accessing the appropriate facial image file, GTGEST displays the unaltered facial image on the monitor. The image is then enhanced based on the final contrast multiplier which was stored with the image. The enhanced image is displayed on the monitor, next to the original image, as is the enhanced image with the feature locations marked, as illustrated in the top row of figure 3-10. GTGEST then performs a program swap with the program PROCESS2. PROCESS2 generates the six facial windows, based on the boundaries described in chapter 3 (see the section on window generation). These 64 by 64 image files are saved to disk as WIND1.PI through WIND6.PI and are also displayed to the monitor (reference bottom two rows of figure 3-10). As each window is saved to disk, a corresponding flag file (NOVASIG1.A through NOVASIG6.A) is generated to initiate processing by CORTRAN16. While CORTRAN16 is determining the six window gestalts, PROCESS2 terminates, and control is returned to the GTGEST program. GTGEST immediately performs another program swap with the program SHOWGEST. The purpose of SHOWGEST is to retrieve the window gestalt values, generated by CORTRAN16, and display these values to the monitor (above the respective window).

Once the process of generating the window gestalt values is completed, control again returns to the top level programs (GETSUBJ on the Nova, FACEFNDR on the Eclipse).

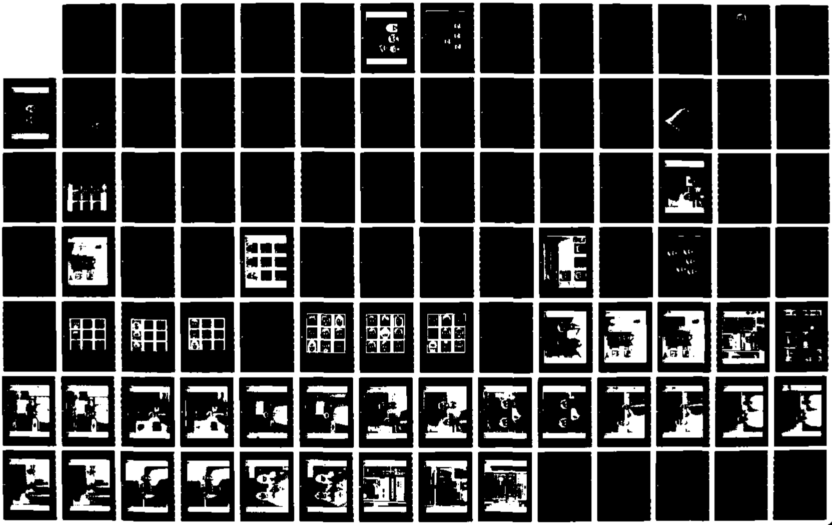
<u>Training the Facial Database</u>

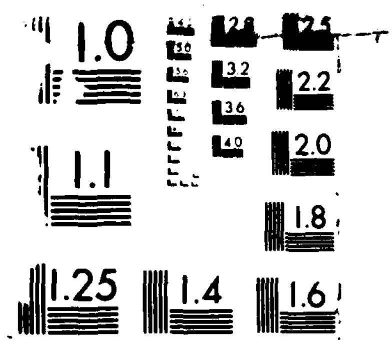
If the operator selected to train with the facial images found by the face locator process, then he is now prompted for a name for this facial image file. The operator is also requested to assign an identification (ID) number to the facial image. This ID number is the way in which the system "knows" an individual. Thus, all facial images assigned to a certain ID number are treated as the same individual.

With the exception of the program GETNAME, [illegible] used to store the gestalts and other per[illegible] in the database, WRNA2 and TRAIN, are [illegible]

AD-A170 052 DEVELOPMENT OF AN AUTONOMOUS FACE RECOGNITION MACHINE (U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING E J SMITH 08 DEC 86 2/3

UNCLASSIFIED AFIT/GE/ENG/86D-36 F/G 14/5 NL






MIC

as described in Russel's thesis. Therefore the reader is referred to Russel's thesis (22:5-31,D-6) for further details concerning these programs. The only significant modification made to TRAIN was to modify it so that it would perform in the directory NSMITH. The purpose of GETNAME is to prompt the user for a filename for the facial image which has just been processed (for which window gestalts were just calculated). The facial image file is then saved to disk with the name given by the operator. All three programs are called into execution by program swap calls from GETSUBJ.

Once the facial image database has been updated, by a single sequence of calls to GETNAME, WRNA2 and TRAIN, the next facial image (if any) is processed. The program GETSUBJ can determine how many pictures need to be processed, based on the value of the variable "IPIXCT" which is incremented during the face location process. If another face needs to be processed then the process starts over again at the point where GETSUBJ initiates the "generation of facial gestalts" (see previous section, 3rd paragraph).

Recognizing Faces

If instead of selecting to train the data base, the operator chose to identify the facial images, then after the facial window gestalts had been determined, as described previously, the algorithm would proceed immediately to the phase of trying to identify the individual.

On the Eclipse, having completed the window gestalt calculations, the program CORTRAN16 terminates and returns control to FACEFNDR. FACEFNDR, in turn, returns to checking for interrupts from the Nova and is ready to begin a new task. On the Nova, GETSUBJ proceeds to initiate the recognition task by creating the flag file "IDCOM" which directs FACEFNDR to begin this task. FACEFNDR activates the program REMID, by program swap, which performs the comparison of the window gestalt values for the unknown facial image with those in the database.

The program REMID is a modified version of the original program REM (used in Russel's thesis). The program has been modified to perform in a non-stop fashion (it no longer requires operator interaction). It has also been modified (by Captain Russel) to perform in an interactive display mode, which allows the results of the recognition process to be displayed on the TV monitor at the Nova. An added feature of this interactive mode is the use of a DECTALK speech synthesizer which announces various milestones the program is about to accomplish. At the end of the recognition process the DECTALK verbally greets the recognized individual. The display program which runs concurrently with REMID (on the Nova) is the program NPROC1. Figure 3-12 illustrates the display generated by NPROC1.

The basic rules of the selection process used in recognizing a person remain unchanged from the original version. In fact, this portion of the code (in REMID) remains unchanged from the original (the program REM). The details of this selection process are fully explained in

Russel's thesis (22:4-32) and to a somewhat lesser degree in Appendix A (starting at page A-7) and will not be repeated here. One parameter has changed slightly from the original version. In the original version, the measure of how certain the computer was (of the correct choice for the unknown person) was indicated by a number which represented the sum (for all six windows) of each window distance metric (discussed in Appendix A, page A-8) times the window performance factor. This parameter has been translated to a "pseudo-probability" measure by multiplying the original sum by 100, and dividing that product by the sum of the six window performance factors.

If more than one face has been found in the original image, the above process is repeated, starting again with the generation of the facial window gestalts for the next unknown image. This continues until all faces that were found have been processed and identified. As the recognition process for each unknown subject is completed, the user is given an opportunity to print the image on the line printer. Printing is performed by the Eclipse system, so it is necessary to use FACEFNDR to print the image. GETSUBJ initiates the print operation by performing a program swap with the program SVPIC. SVPIC then saves the image on the monitor, to disk, and generates a flag file "PRNTIMAGE". The flag file causes FACEFNDR to print the image on the Eclipse printer.

# V. Test Results

As in the previous chapters, this chapter is divided into two main sections. The first section examines the tests and test results associated with the face locator function of the AFRM (Autonomous Face Recognition Machine). The recognition performance of the AFRM is then examined in the second section.

## Locating Faces

Two types of images were used to test the face locator portion of the AFRM. One type was an image containing several facial images pictured against a pristine background. This type of image is illustrated in figures 5-1 and 5-2. Images such as the these were constructed for 30 different subjects. All of the facial images were taken from the on-line facial database (for examples see appendix D). The individual facial images used were all obtained under the following conditions:

1. The F-stop used in all cases was F8.0.
2. Camera zoom was adjusted to exactly fit the vertical dimensions of the head (bottom of chin to top of head) within a 64 by 64 pixel image.
3. White background.
4. Normal laboratory lighting (overhead fluorescent).
5. Subject position was fixed relative to the video camera (9 feet in front of the camera).
6. Slight variations in facial expression were taken for each image of a subject.

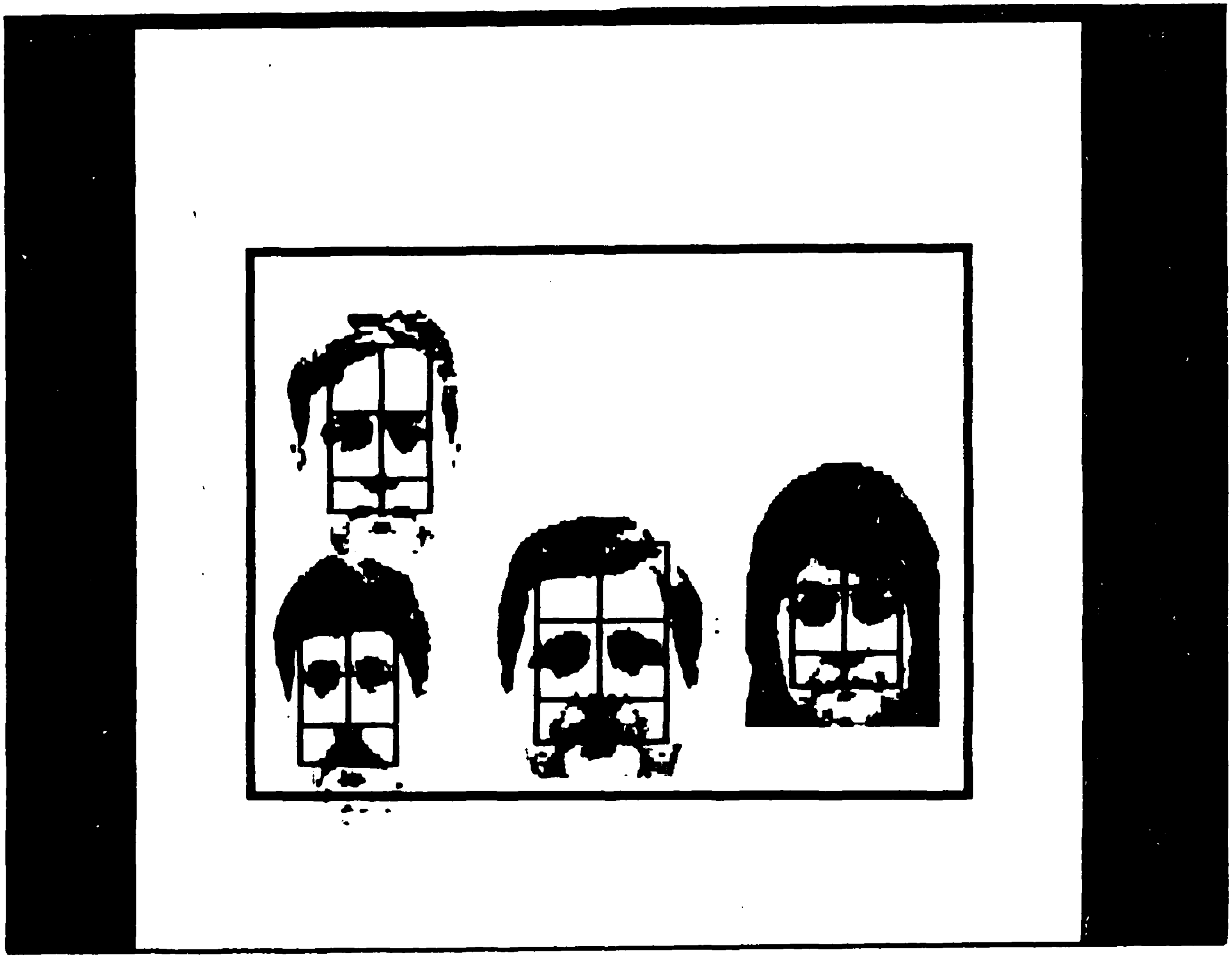


Figure 5-1. Test Image #1 - Pristine Background.

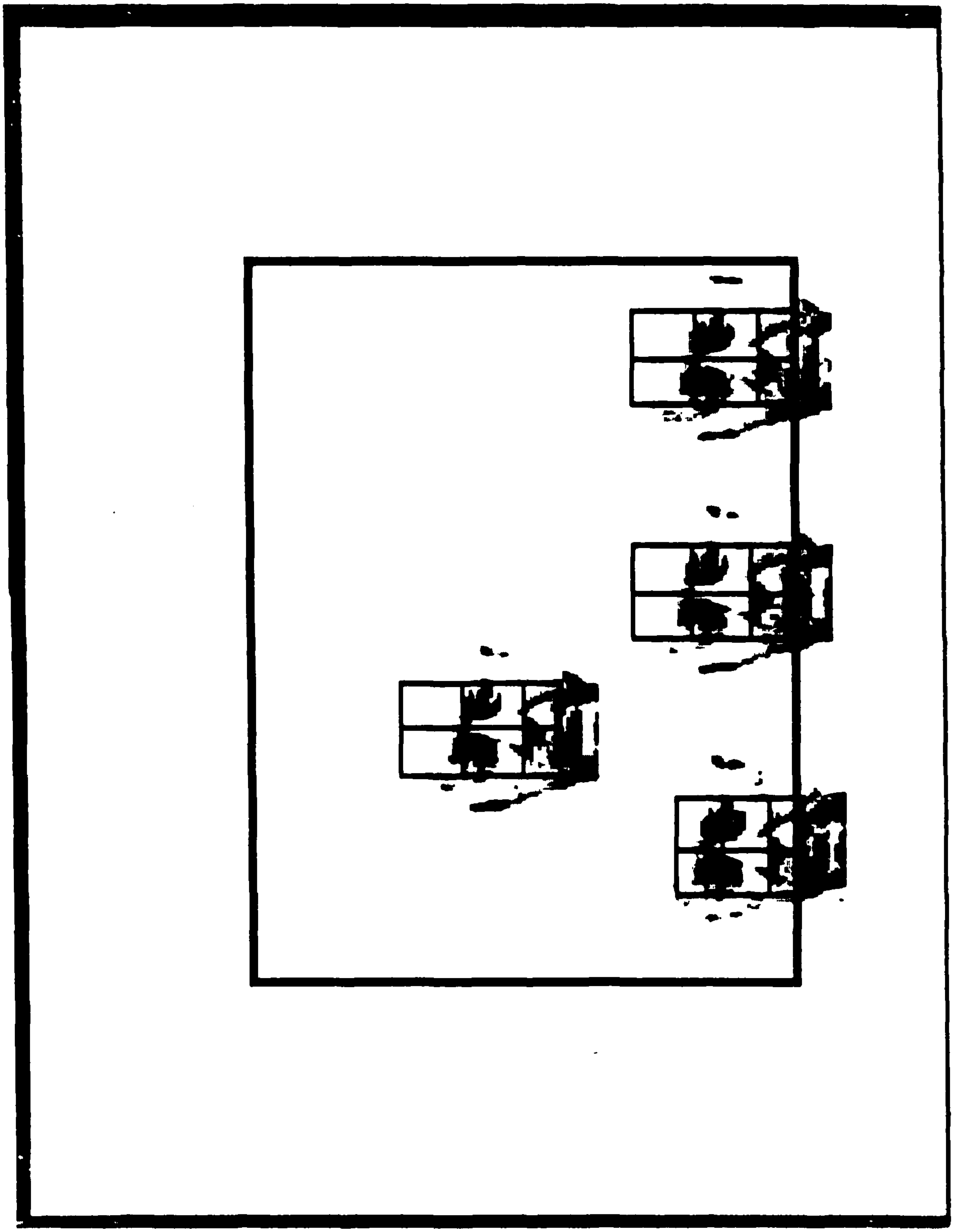


Figure 5-2. Test Image #2 - Pristine Background.

The multiple facial image pictures shown in figures 5-1 and 5-2 were produced by overlaying the facial images from the database onto a white background. This process was performed completely by the computer, using the program PICTURE2. The only criterion for placement of the facial images was that the eyes must appear within the search area (the rectangular box indicated in figure 5-1). Otherwise, the placement of the facial image was arbitrary.

Although the subjects were, for the most part, chosen at random, an attempt was made to ensure that a broad variety of facial types were included. All but two of the individuals illustrated in appendix D were included in this group. The two that were excluded are shown in figure D-3 (reference Appendix D), in the lower right and lower left corners. They were excluded because the majority of the images contained in the data bank, for these subjects, were facial images taken while wearing dark rimmed glasses. Glasses in general, and especially dark rimmed glasses, significantly distort the eye signature. Thus, one of the pre-conditions assumed in this thesis is that the subject is not wearing eye glasses.

The total number of different facial images tested in this manner was 139 (about 4 to 5 different pictures per subject). Of that total, 131 were correctly located. Of the 131 correctly located, only two showed an obvious error in the determination of the location of a single facial feature. That single feature was the location of the mouth.

In these instances the mouth was erroneously determined as being somewhere between the top of the nose and the actual center of mouth. The remaining feature locations were correctly identified. This performance reflects a 94% hit (success) rate for the locator algorithm.

The second type of image used to test the face locator function was one in which the background was not pristine. An example of this type of image was used in chapter 3 (reference figure 3-7). The purpose of this test was not so much to determine the hit rate for finding faces, as it was to determine how "face specific" the face locator is. If the algorithm were to work in an ideal fashion, it would be completely (100%) face specific. In other words, any object (or collection of objects) that did not constitute a facial image, under ideal conditions, would never be confirmed by the algorithm as a face. Unfortunately, there is apparently no exact metric by which to gauge the face specificity of the algorithm. Thus, the results of this test reflect the subjective judgement of the investigator.

Since the AFRM system was designed as an enhancement to the original face recognition machine (developed by Russel), with the goal of removing as much reliance on the operator as possible, it was decided (as a first cut) to test the machine with background conditions which occur normally in the AFIT Signal Processing Lab. Normal conditions means that the operator acquires the picture of a subject by pointing the camera at the subject(s), adjusts the zoom so that the size of the facial image(s) are within an acceptable range, and snaps the picture. The operator need not concern himself about the background or the exact position of the individual with respect to the camera. The assumptions stated in chapter 1 (Assumptions section) must still be observed. That is, the subject is looking square at the camera, with minimal rotation or tilt of the face. The lighting is normal overhead lighting, and the subject is not moving.

Appendix E contains several examples of test images taken under these conditions. These images are presented first as the unaltered image, followed by the same image (in the next figure) showing the results of the AFRM face locator function. In each case, the subjects, the number of subjects, the size of the facial images and the background were varied as much as possible (within the confines of the lab). In the images which show the results of the face locator analysis, facial images are marked, as ususal, by a gridwork of 3 vertical and 4 horizontal dark lines. The search area is marked by the dark boundaries of the large box cursor.

Only two instances of false alarms occurred in these images. One instance occurred in figure E-5, where the computer misinterpreted the dark round control knobs located above and to the right (subject's right) of the subject. Another occurred in figure E-13, where some background noise located to the right (subject's right) of the neck of the taller subject was interpreted as a face. Figures E-26 through E-28 contain no faces, and were designed as "acid" tests to see if any false faces might be found. In no case was the presence of a face confirmed in these images. The results of these preliminary (and somewhat limited) noisy background tests and the previously described test (94% hit rate with a pristine background), indicate that the algorithm used for locating faces has a reasonably high specificity for faces.

Concerning the test images present in appendix E, a few images indicate that a face was not found. In test image #1 (figure E-1, Appendix E), the condition of adjusting the zoom to get roughly the proper size was deliberately violated. In this case, an analysis of the search area by the face locator algorithm yielded the result that no face was found. This result is expected since the facial images were too large to be contained within a 64 by 64 image sub-section. The reader will recall that, once a facial image has been located, it is saved as a 64 by 64 image sub-section. If the facial image is too large, then information necessary for the face recognition process would be lost. Thus an upper limit is placed on the size of the facial image. This limit is governed by the distance (in pixel columns) between the locations of the two maxima in the eye signature. If that distance is greater than 21 pixels, then the signature is not recognized as an eye signature. Figure 5-3 illustrates an example of the 64 by 64 image obtained when the program is operating very close to

Figure 5-3. Example of Maximum Face Size.

the maximum facial size. The sub-image illustrated in figure 5-3 was obtained from test image #8 (figure E-14). The same explanation is true (the facial image was too large) for the missed face in figure E-13.

Figure E-19 illustrates the case where the facial sizes are just at the lower limit. Only one subject face was confirmed in this image. Because the distance between the two maxima (from the eye signature) was less than 10 pixels for the other two subjects (on the left), these faces were not located.

One further test was done to observe the affect of using only an edge enhanced facial image. Figure 5-4 illustrates such an image. This facial image was obtained by photographing the image directly from a book (9:167). In photographing the image, it was necessary to use a very small aperture (F16.0) to obtain reasonable resolution. Use of the video/digitizer system under these conditions generates considerable noise in the image. This is illustrated in figure 5-4. Both images in figure 5-4 are pictures of the same original image taken at different times. In one case (for the left image), the face locator was successful in finding the face. However, for the image on the right, the face locator was unable to find the face. The reason for this failure was investigated by obtaining and analyzing the eye signature for the facial image on the right. The eye signature is illustrated in figure 5-5 (upper right corner). Examination of the eye signature readily indicates why the algorithm failed in this case. There is a small peak at the very center of the signature due to the noise in the image. The smoothing technique used prior to analysis of the eye signature is not sufficient to remove this small peak. Thus, when searching for two successive peaks, the algorithm will erroneously identify this small central peak as the second peak. This results in a failure to meet the eye signature criteria and the search fails.

This type of problem did not occur with images obtained with larger apertures (F8.0). The problem appears to be due to discretization error within the system which is much more marked when operating at small apertures. Although the average system in use would not demonstrate such noisy behavior, the answer to this problem is to filter out as much of the noise as possible as part of a pre-processing step. Alternatively, when searching for the eye pattern in the original image, a darker "background fill intensity" (as

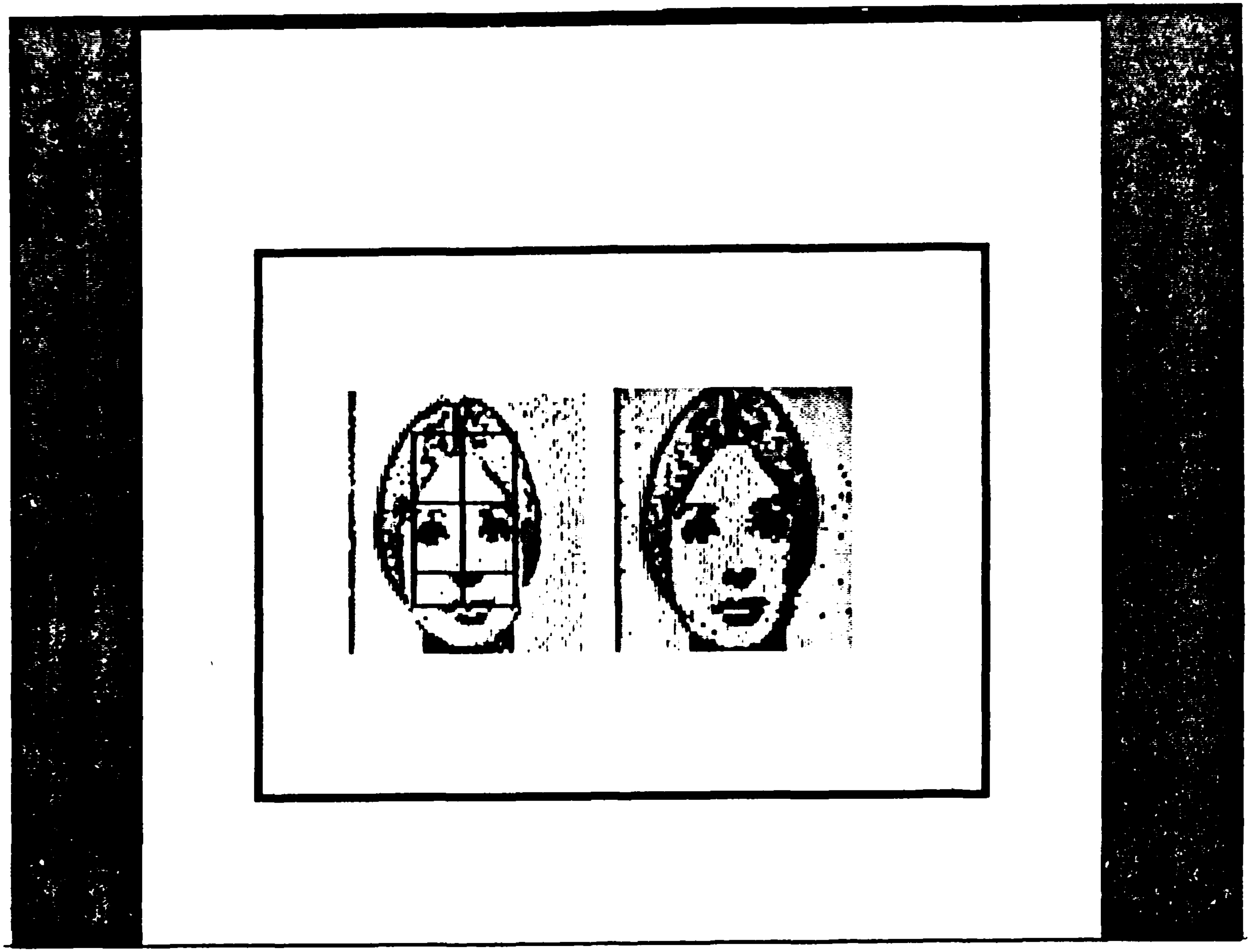


Figure 5-4. Finding an Edge Enhanced Facial Image.

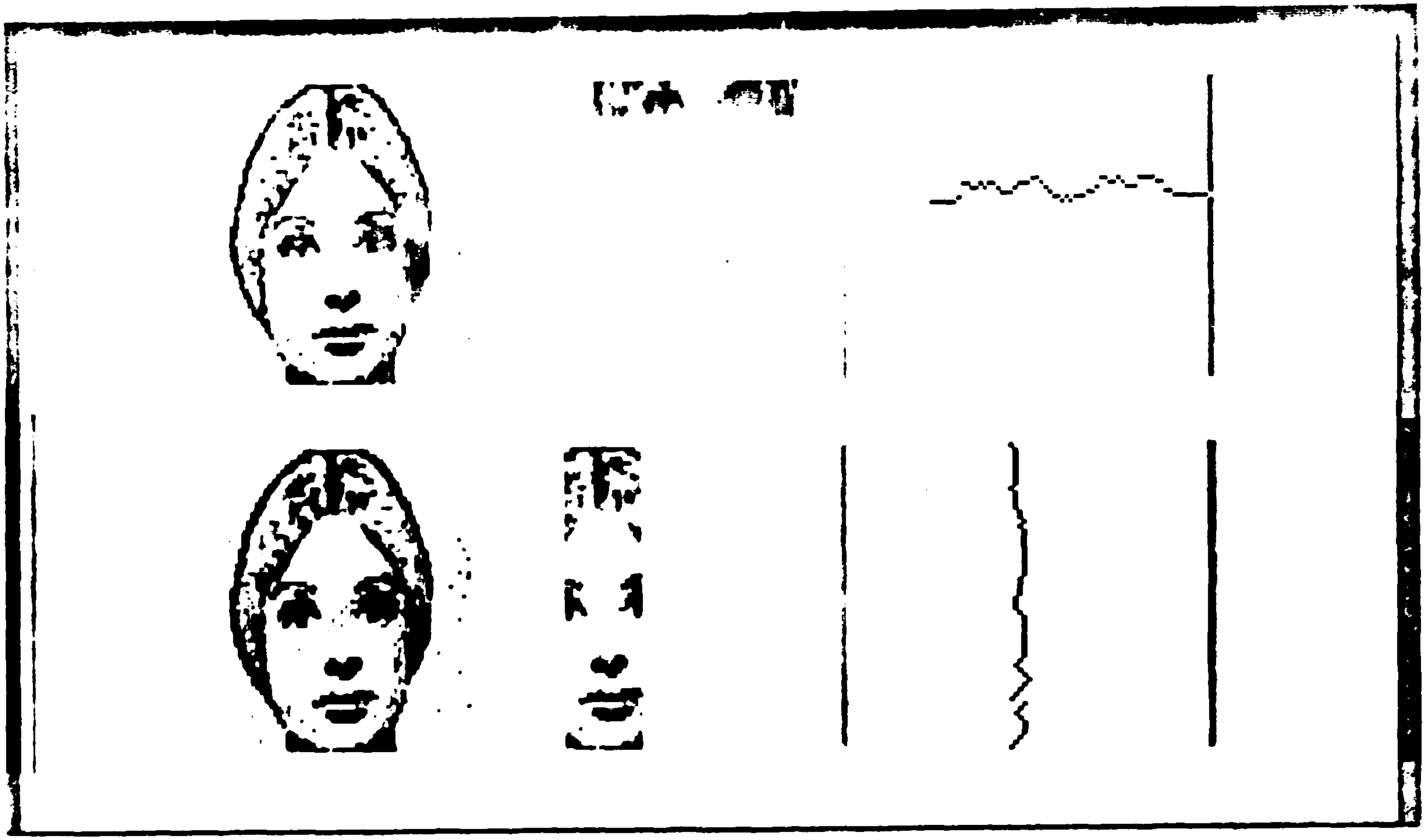


Figure 5-5. Signatures of the Missed Face of Figure 5-4.

discussed in chapter 3, page 3-11) might be used to suppress such noise.

Recognizing Faces

Using a database of twenty individuals, the system was tested for recognition performance. The system was trained with an average of four facial images per subject. A fifth image was held in reserve (the machine was not trained with this image) for testing. All images were acquired using the face locator algorithm, and windowed in the manner described in chapter 3. Two trials were conducted using the same database of 20 people. On the second trial, those images that had been used (on the first trial) as test images, were used as training images. An image (for each subject) that had previously been used for training, was then used to test the system. For example, if the system had been trained on pictures Smith1.PI through Smith4.pi, and tested with picture Smith5.pi in trial #1, then in trial #2 it would be trained on Smith2.PI through Smith5.PI and tested on picture Smith1.PI. The recognition performance results that were obtained are shown in Table 5-1.

Also shown in Table 5-1 (for comparison), are Russel's original results. The metric "average reduction in uncertainty" or "F" was determined as follows:

$$F = 1 - (1/M)\sum_{i=1}^{M}(Si-1)/N \qquad (5\text{-}1)$$

where $S_i$ = number of individuals the correct person is down from the top of an ordered list of candidates.

i = number of the particular individual in the database who is being processed for recognition.

N = total number of individuals in the database.

M = number of individuals for which the recognition system was tested.

Equation 5-1 is identical to that used by Russel (22:6-8). The average reduction in uncertainty is a more accurate measure (than the percent absolute correct) of the machine's performance since it takes into account how close (in rank order) the correct individual was to the top candidate.

Although the recognition performance was relatively poor when gauged in terms of percent absolute correct, the figures obtained for the average reduction in uncertainty were respectably high. In both tests, the correct individual was in the top 3 candidates, in 18 out of 20 cases. These results look very promising, especially considering that the choice of facial gestalt windows was done rather hastily and was not based on a thorough analysis of all possible windows.

**Russel's Results:**

Number in database: 20
Number recognized as 1st Choice: 18
Number recognized as 2nd Choice: 1
Number recognized as 3rd Choice: 1

Absolute Correctness = 0.90
Average Reduction in Uncertainty = 0.9925

**AFRM Results:**

**Trial #1:**
Number in database: 20
Number recognized as 1st Choice: 12
Number recognized as 2nd Choice: 4
Number recognized as 3rd Choice: 2
Number recognized as 5th Choice: 1
Number recognized as 8th Choice: 1

Absolute Correctness = 0.60
Average Reduction in Uncertainty = 0.9525

**Trial #2:**
Number in database: 20
Number recognized as 1st Choice: 10
Number recognized as 2nd Choice: 5
Number recognized as 3rd Choice: 3
Number recognized as 5th Choice: 1
Number recognized as 11th Choice: 1

Absolute Correctness = 0.50
Average Reduction in Uncertainty = 0.9375

Table 5-1. Test Results for Recognition

## VI. Conclusions and Recommendations

### Conclusions

In an attempt to combine the face recognition capabilities of Russel's Face Recognition Machine, with the additional capability of automated scene analysis for faces in a digital image, the Autonomous Face Recognition Machine (AFRM) has been developed. The question that needed to be answered was: "Can a machine, entirely on its own, determine whether or not a person's face is in a picture, and if so, can it determine to whom the face belongs?". The results of this thesis demonstrate very clearly that the answer is yes, on both counts. The results also indicate that much more development needs to be accomplished before such a machine becomes practical.

In terms of its ability to analyze a digitized scene for the presence of faces, the AFRM demonstrated a high success rate of 94%. For the remaining 6% (representing 8 of the 139 facial images tested), the machine initially zeroed in on the face in the large majority of cases. It did not confirm the presence of these faces for various reasons. One such reason is that the contrast expansion of the original image was too great due to dark hair at or below the eyebrow level. Another reason is that the location of various features were outside the expected range by one or two pixels. These problems should be relatively easy to overcome by relaxing and adjusting the constraints of the face locator. Although difficult to quantify, the AFRM's ability to distinguish between faces and other classes of objects (it's face specificity) appears to be quite good. False confirmation of a facial image occurred in only two instances out of 17 test images.

While the face locator is relatively immune to variations in translation and scale of the facial image, it is susceptable to rotation of the facial image. Small degrees of tilting or turning of the head have no ill effect. But larger degrees of rotation (more than 5 to 10 degrees) will cause the face not to be found.

The face locator is also relatively immune to electronic equipment noise. The case demonstrated in figures 5-4 and 5-5 is an extreme case where there is a very high degree of discretization noise at intensity levels low enough to affect the eye signature. Normally, it is not envisioned that the machine would be required to work with such a poor image. The algorithm could be adjusted to compensate for such a problem.

The AFRM's ability to recognize an individual was significantly reduced when compared with Russel's original results for absolute correctness. This is not surprising when one considers that the recognition scheme used in the AFRM is based solely on the internal features of the face. Unlike the original (Russel's) machine, the AFRM does not have available to it the identity specific information contained in the outer bounds of face and the hair style. However, when compared to Russel's results for the average reduction in uncertainty, the machine compared quite favorably. It is highly probable that, through a more judicious choice of windows, the recognition performance can be brought up to the level achieved by Russel.

## Recommendations

Considering the relatively crude method used to analyze the gestalt pattern for the eye signature (point by point analysis of the curve), and the very limited criteria for the eye signature, the high performance rate of the face locator function is very encouraging. A significant improvement to the algorithm would be to first do a least squares fit of the gestalt pattern data to a set of polynomial coefficients. Alternatively, one might also fit the data to a set of Fourier coefficients. In either case, this type of transformation would facilitate a more reliable and consistent analysis for the eye signature. In addition to alleviating such noise problems as that indicated in figure 5-5, an extra bonus might be obtained. The bonus would be a further means of identifying the face. Once the Fourier or polynomial coefficients have been obtained for the eye signature curve, it may be possible to identify an individual (at least in part) by comparing these coefficients with those of previously obtained eye signatures. In addition to the eye signature, the application of this technique to the nose/mouth signature should also be examined.

A consistent problem observed in this thesis, and in previous work of this nature, is the problem of how to get rid of extraneous facial shadows. Facial shadows which vary with the lighting arrangement are usually detrimental to any scheme for identifying faces. The problem is to get rid of these shadows while keeping those which are informative (such as those for the eye sockets and nose). In other words, one has to be careful that they don't "throw the baby out with the bath water". One possible solution to this problem may be to spatially filter the image (once it has been located and bounded by the face locator). The optimal range of spatial frequencies (cycles/facewidth) needed for facial identification have been discussed in the literature (8,19). It should be a relatively simple matter to implement the appropriate filters to remove at least the very low frequency shadows present in the facial image.

In addition to the above recommendations, the reader is referred to Russel's recommendations (22:8-1). Due to time constraints, many of his recommendations were not addressed in this thesis. They also should be considered in any future development of the AFRM.

# Appendix A

## A Face Recognition System Based on Cortical Thought Theory

# A FACE RECOGNITION SYSTEM BASED ON CORTICAL THOUGHT THEORY

Robert L. Russel Jr., Capt, USAF
Rome Air Development Center (RADC/IRDT)
Griffiss AFB, NY

Richard L. Routh, PhD, Capt, USA
U.S. Army Signal Corps (ATZH-CDC)
Ft Gordon, GA

James R. Bolten III, PhD, Capt, USAF
Air Force Weapons Laboratory (AFWL/SISG)
Kirtland AFB, NM

Matthew Kabrisky, PhD
Air Force Institute of Technology (AFIT/ENG)
Wright Patterson AFB, OH

ABSTRACT: A face recognition system was developed at the Air Force Institute of Technology, based on the principles of Cortical Thought Theory (CTT), proposed by Dr. Richard Routh in July 1985 as his doctoral dissertation. CTT is an initial attempt at a unified brain theory. The CTT "gestalt" transformation maps a 2-dimensional image into a 2-D coordinate point. The face recognition system extracts six sub-images from a contrast-expanded image, calculates the 2-D gestalt coordinates, and stores the information in a database. Statistics are then calculated on at least five prototypes processed for each person. An "unidentified" person is recognized by calculating the six gestalt feature vectors, and finding the closest match to previously stored data. Performance testing of the system yielded a reliability of 90% for a database of 20 people.

## I. Introduction.

A face recognition system was developed, based on the principles of Cortical Thought Theory (CTT), recently published by Dr. Richard L. Routh in July 1985 as his doctoral dissertation at the Air Force Institute of Technology (4). CTT claims to be a generic model for sensory information analysis, regardless of the domain or entry level of abstraction. Routh tested the CTT architecture successfully for speech processing. In order to test this architecture as a generic model, CTT was tested for visual processing, specifically for the difficult task of human face recognition.

## II. Background of Cortical Thought Theory.

Since the purpose of this research was to apply Cortical Thought Theory to the domain of vision, it would be instructive to review the major concepts of this theory. For years, those involved in Artificial Intelligence (AI) have tried to model human thinking by using logic and other deductive processes. They have enjoyed considerable success in many areas, but computer systems still have great difficulty reproducing what we call "insight", or taking two pieces of information and inducing a new association. Another problem with conventional AI systems is that the search time increases exponentially with the size of the knowledge base.

Rather than starting with basic operations (primitives) using deduction (a concept well-established in AI), Routh approached the problem by starting with primitives of induction. His theory proposes that information is displayed as a two-dimensional image on the cortex surface of the brain. Then the cortex must extract a two-dimensional vector from the image, which he referred to as the "gestalt" of the image. He maintained that the dimension of the gestalt feature vector set must be "two". This type of representation allows direct memory access, which means basically no increase in search time, even with any increase in size of the knowledge base. This 2-D vector is all that is passed up to the next level of abstraction. He explains this as follows:

> By using the experimental results obtained from the perceptual psychology investigations into the nature of the human gestalt mechanism by Kabrisky, Maher, Ginsburg, Pantle, and Sekuler (among others), it was argued that the two element gestalt vector is probably extracted from some low pass two-dimensional spatial frequency domain representation of the 2-D input image.

But what spatial frequency domain representation was to be used? Several methods of displaying the low-frequency spatial harmonics of a 2D-DFT were investigated so as to find a single identifying 2-space vector characteristic which could be called a "gestalt". The method had to suppress the D.C. value which did not contain useful information for identification.

It also had to deal with how to present both sine and cosine components of a 2D-DFT on a 2-dimensional surface. It was observed that if the Two-Dimensional-Discrete Fourier Sine Transform (2D-DFST) was used (instead of the 2D-DFT), and if the technique of zero-filling was used to produce sub-integral harmonics, a "hump" was usually observed between the zeroeth and the first harmonic. The location of the peak of this hump could easily represent the gestalt value since it can be represented by a two-space vector, and it changes location for different input images (see figure 1). Experiments suggested that it was sufficient to examine the 1/64th harmonics between zero and one. The 2D-DFST gestalt mechanism is specified by the following equations:

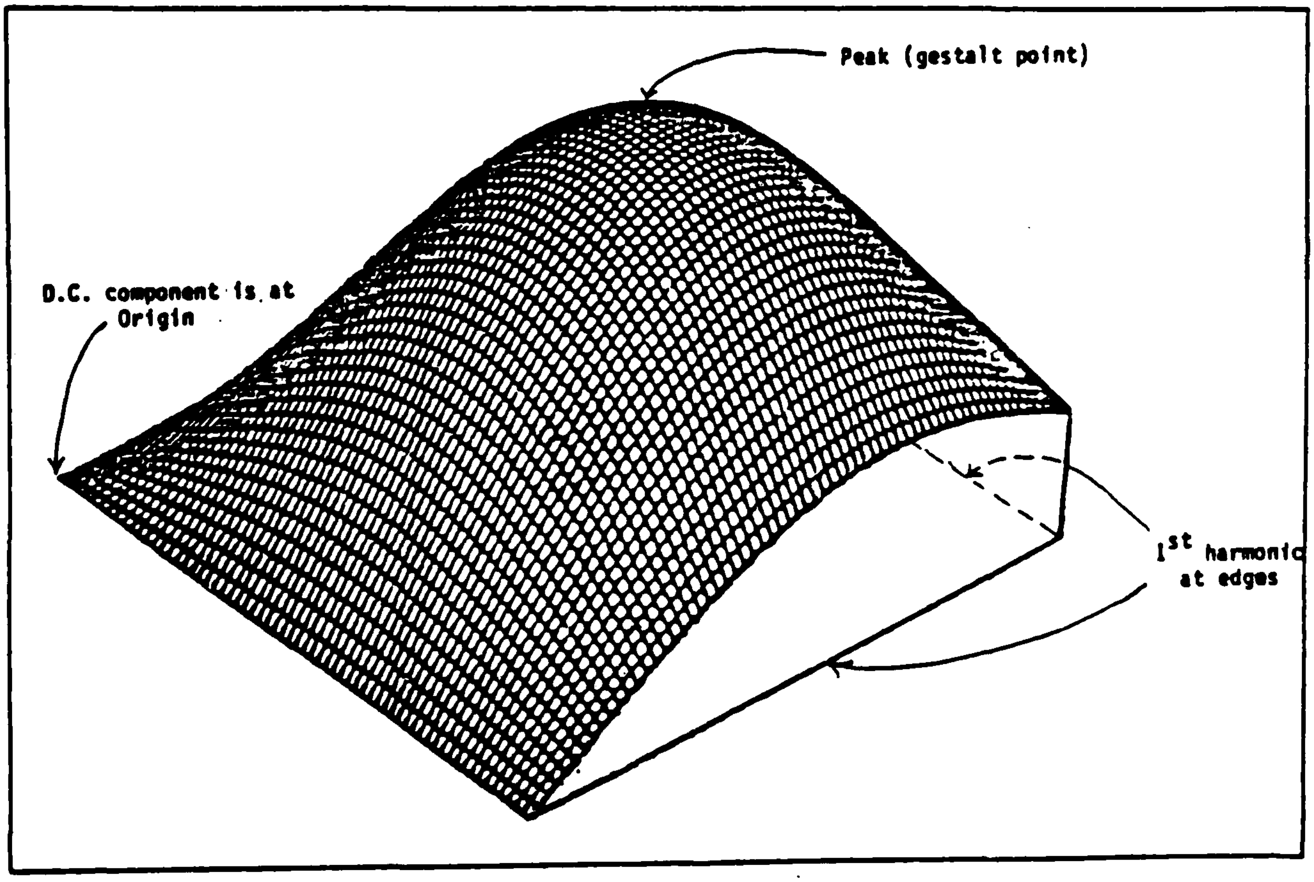



Figure 1. Example of "Gestalt" Transformation

Given the Discrete Input Image :

$M_{kh}$; $k,h=1,\ldots,64$

$$S_{kj} = \sum_{h=1}^{64} M_{kh} \sin \frac{2\Pi j(h-1)}{4096} \; ; \qquad (1)$$

$k,j=1,\ldots,64$

$$T_{ij} = \sum_{k=1}^{64} S_{kj} \sin \frac{2\Pi i(k-1)}{4096} \; ; \qquad (2)$$

$i,j=1,\ldots,64$

$$\Longrightarrow GFVS = (i,j): \max_{(i,j)} T_{ij},$$

where GFVS=(i,j) is the two-space vector identifying the location of the gestalt on the next higher (in the hierarchy of abstraction) local cortex surface.

It was shown that the level one neurons of the cortex could easily perform a very good approximation to the 2D-DFST from the zeroeth to the first harmonic. There would be an error between the true 2D-DFST and the cortex transform, but the cortex transform still preserves the important characteristic: it produces a "hump" whose peak moves in relation to the human-perceived difference in the input images. The gestalt would be the two-space location of the cortical column located at the highest amplitude point (5).

The transform used to simulate this process is as follows:

Given the Discrete Input Image:

$M_{kh}$; $k,h=1,\ldots,64$,

$$S'_{kj} = \sum_{h=1}^{64} M_{kh} \exp -\frac{1}{2}((h-j)/63\sigma)^2 \; ; \qquad (3)$$

$k,j=1,\ldots,64$

$$T'_{ij} = \sum_{k=1}^{64} S_{kj} \exp -\frac{1}{2}((k-i)/63\sigma)^2 \; ; \qquad (4)$$

$i,j=1,\ldots,64$

$$\Longrightarrow GFVS = (i,j): \max_{(i,j)} T'_{ij} \quad \text{where } \sigma = 0.435 .$$

(In this system, $0 <= M_{kh} <= 15$. "0" signifies a "white" pixel, and "15" signifies a "black" pixel, with varying grey scale values in between.)

The above "cortex" model of the transform is the basis of the feature vector used in the CTT Face Recognition System. Routh's theory also embraces the work of Dr. Leslie Goldschlager from the University of Sydney in Australia (1). Goldschlager, studying brain theory on an independent course from Routh, explains how a local cortex surface could reasonably perform the operations of set completion and sequence completion (1,4). Set completion is an operation in which all points of a set are retrieved, given a unique subset. This characteristic may explain such phenomena as recalling many things about a person seemingly simultaneously, given only the person's name. Sequence completion embodies the AI concept of scripts, in which points are stored in the order in which they occur. Given a unique subset of these points <u>in the right order</u>, sequence completion will retrieve the rest of the points in the sequence.

Combining the retrieval characteristics of set completion and sequence completion with Routh's gestalt mechanism, Routh proposed a model of a complete human reasoning system (4,5).

## III. <u>CTT Vision Model</u>.

The first step in designing a vision system based on CTT is to examine the general requirements which CTT outlines for a human-like information processing system (see figure 2):

1) Display the information as a 2-dimensional image.
2) Define the proper boundaries, or "windows", on the image.
3) Extract different sub-looks, or "sub-windows", from the image.
4) Calculate the gestalt of the different sub-looks.
5) Display the gestalts from all the windows as points in a new image.
6) Apply "set completion" to find the set of previously-seen points to which this new set maps to most closely.
7) Find the gestalt of this new image. This will be displayed as a single point on a 3rd level of abstraction, and is the "name" of the original image.

This results in a surface displaying the names of all the images the system has seen. To "recognize" an image, the system would calculate its 2-dimensional gestalt coordinates. The "name" of the image is then whichever previously stored

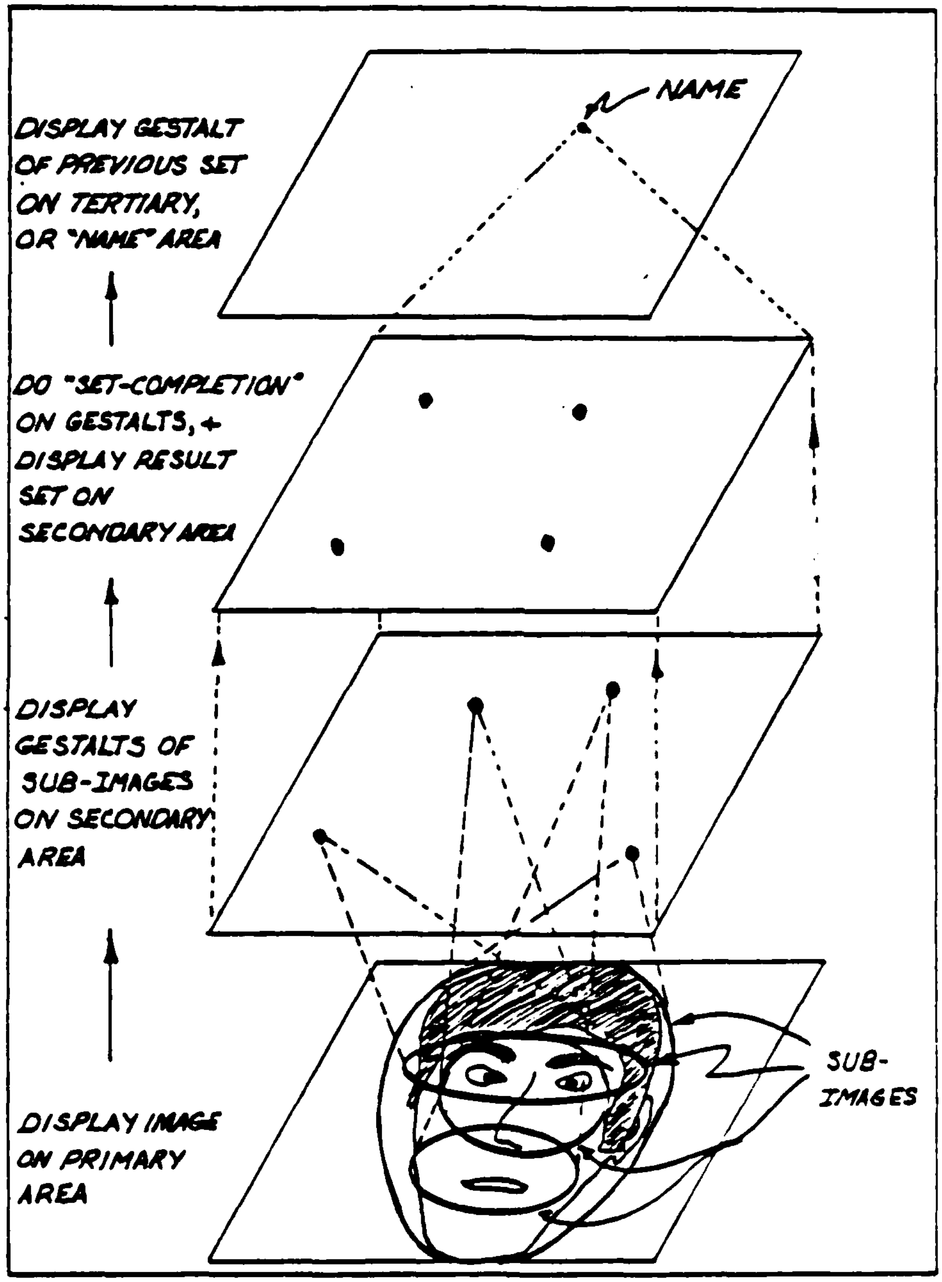



Figure 2. Initial CTT Image Processing Concept

point on the "name surface" to which the coordinates of the unidentified image are closest.

Routh constructed a limited speech recognition system by calculating the gestalt coordinates of time slices of speech signals which were displayed in the log-amplitude by log-frequency format in which audio information is presented to the brain's primary audio cortex. As an initial test, the 64x64 primary audio cortex map was removed from Routh's speech system, and in its place was inserted a 64x64, sixteen gray level, digitized image of a human face. This analysis was applied to five images each of sixteen different people. The results indicated that human faces can be classified and distinguished with the CTT model, and the 2-D CTT mapping (or "gestalt") of the faces is psychologically similar to the way a human would group them.

IV. Development of an Advanced Model.

During the course of the research, it became evident that to get a better separation between individuals, the system needed to look at several sub-parts of the face. The whole face gestalt provided useful, but not sufficient information. The following additional processing steps were added:

A. Contrast Enhancement. When initially taking pictures and processing gestalts, the effect of lighting and f-stop was evaluated. It was found that the best separation came between pictures of different people came from a high-contrast image in which facial lines are bleached out (for the most part) and hair, eyes, nose and mouth appear as dark blobs. The person is usually still recognizable in this form. Pictures were taken at an f-stop of F8 (where all head boundaries were still visible to the human operator and computer), the computer extracted boundary information from this picture, and then artificially expanded the contrast. (See figure 3, top center picture.)

B. Feature Location. Using this contrast-expanded image, the system estimates locations of the major features on the face, and displays them on the screen (see figure 3, top right picture.) The user can at this point readjust the feature locations if the computer chose them incorrectly. The computer will then redisplay the changed values.

The half-face representation was used for a special reason. It was found that the gestalt transform tends to find the center of mass on an image. Given this, then the transform is not sensitive to aspect ratio with the head centered in the transform window. Since a face is basically vertically symmetrical, then a wide face will give the same gestalt in the X direction as a thin face. Unfortunately, people tend to be quite aware of aspect ratio when recognizing someone (determined by an informal survey by one of the authors.)

To handle this problem, it was necessary to divide the image down the center, display the halves as two separate images, and take the gestalts of the separate images (see figure 3.) Now changes in aspect ratio are reflected as changes in the X direction of the gestalt.

Wanting to be consistent with CTT and the physiology, this split-image requirement was found to be a strange restriction of the presentation of a facial image. Then it was realized that the primate visual system also splits images vertically down the center before displaying them on separate left and right primary visual cortexes (2). The reasons for the partial splitting, (or "decussation") of the visual pathway at the optic chiasm are not well understood, and attempted explanations for the phenomenon quickly become complex and convoluted. It is significant that Cortical Thought Theory provides a possible explanation which is simple, straightforward, and is a natural requirement of the theory.

C. Window Extraction. Six different sub-windows on the face were extracted. (See figure 3, bottom six pictures.)

D. Gestalt Calculation. For each of the six windows, a two-dimensional gestalt coordinate is calculated, transformed for scale, and displayed above the sub-image for which it was calculated (see figure 3.)

E. Storage in Database. Once the gestalts are calculated for all six windows on the face, all the data for this picture is put together as a record in the Processed Picture Database. This process is repeated for each picture. When all the pictures are entered for an individual, the system is ready to be "trained" with the data.

F. Training the Database. Training is done by characterizing an individual by the X & Y mean and standard deviations of gestalt values over a number of pictures. In this way the system has an idea of a reasonable range of values to expect for a given individual. For this study, five pictures were taken of each person for training. The authors realized that scores of pictures taken over a period of time (say, a year) would be desirable to thoroughly test the system. However, time constraints prevented this. It was assumed the five pictures would get us "in the ballpark," and a definable cluster was indeed observed with only 5 pictures.

Statistics were calculated for each individual in the database, defining their X & Y mean and standard deviations for each of their six windows. In addition, other statistics for each of the six windows were calculated, giving such information as how big the search area should be, and which windows give the most reliable information. A "Recognizer Database" was set up for each of the six windows, with the ID number and X & Y standard deviations for a person stored at the 2-D coordinate location indicated by the person's average X & Y gestalt values. Once the coordinate database has been trained, it ready to "recognize" an individual.

G. Identification. To identify a person, the "unidentified" person's picture must first be processed for gestalts, as previously described. Using the gestalts, the program now generates an ordered list of candidates. The top person on the list is the winner.

1) Candidate Selection. Each individual only appears once in each window database. When trying to recognize an individual, the program first determines which individuals are within a predetermined range (3.0


Figure 3. Example of a Picture after Processing

standard deviations) from the coordinate values of the unidentified person. Those within the range are selected as candidates for further processing, while thsoe outside this range are rejected. This is in terms of the different standard deviation values associated with each individual. For instance, assume (for just one dimension) the mean coordinate values for Mike Jones and Joe Smith are the same distance away (let's say 2.0) from the values for the unknown person. However, Mike's standard deviation value is 0.5, and Joe's standard deviation is 2.0. Therefore, Mike Jones is 4.0 standard deviations from the unknown person, and will not be selected as a candidate. However, Joe Smith is 1.0 standard deviations away, and is selected as a candidate.

2) Distance Measure. A distance is calculated from the unknown person's values to stored values for each of the six individual windows. The distance measure for each individual window is:

$$V_{iw} = \exp \frac{-1}{\sqrt{2}} \left[ \frac{(G_{ix} - G_{ux})^2}{(2\sigma_{ix})^2} + \frac{(G_{iy} - G_{uy})^2}{(2\sigma_{iy})^2} \right] \quad (5)$$

where i = number of individual in database being considered,

w = number of window being processed,

$G_{ix}$, $G_{iy}$ = X,Y coordinate values of previously stored candidate,

$G_{ux}$, $G_{uy}$ = X,Y coordinate values for an unidentified person, and

$\sigma_{ix}$, $\sigma_{iy}$ = X,Y standard deviations for person "i".

3) Weighting by Window Performance. As mentioned above, each window, or sub-look, has its own database, and a distance measure is calculated for each window. However, when combining values from all the windows, should all windows hold equal weight? Elaine Rich, in her book Artificial Intelligence, points out that the weighting function should take into account the "confidence in the evidence" (3). In this application, the "confidence" is how well the particular window discriminates between individuals, and is referred to in this work as "performance factors." A performance factor is calculated as follows:

$P_{wx}$, $P_{wy}$ =

(Average standard deviation of the mean of (x,y) gestalt values)/ (Average of the standard deviations for all (x,y) gestalts). (6)

Combining the x and y weightings, we have

$$P_w = \sqrt{(P_{wx}^2 + P_{wy}^2)} \quad (7)$$

where $P_w$ is a performance factor which indicates the ability of the particular window to discriminate between individuals.

4) Final Recognition List. By repeating this process on all six windows, summing the values for each window, and sorting them, the result is a list ordered from the most-likely candidates to the least-likely.

$$T_i = \sum_{W=1}^{6} (P_w * V_{iw}), \quad (8)$$

where $T_i$ = list of total values for individuals for all windows,

$P_w$ = Performance factor weighting for window w, and

$V_i$ = Value of individual "i" in window W.

V. TESTING

The system was trained with from 4 to 9 pictures each of 20 individuals. One image for each individual was used to test the system. (This picture was not included in the training set.)

VI. RESULTS.

A. Recognition Performance.

1) The overall recognition results obtained are shown in table 1:

| | |
|---|---|
| Number in database: | 20 |
| Number recognized as 1st choice: | 18 |
| Number recognized as 2nd choice: | 1 |
| Number recognized as 3rd choice: | 1 |
| Absolute Correctness = 0.90 | |
| Average Rank Order = .9925 (with 1.00 being an average of #1 out of 20.) | |

Table 1. Test Results for Recognition

2) Performance of the Individual Windows. This performance is shown in table 2:

| Individual Window | Absolute Correct | Average Rank Order |
|---|---|---|
| 1 | 0.50 | 0.915 |
| 2 | 0.75 | 0.983 |
| 3 | 0.60 | 0.870 |
| 4 | 0.35 | 0.933 |
| 5 | 0.30 | 0.823 |
| 6 | 0.55 | 0.958 |
| All Combined | 0.90 | 0.993 |

Table 2. Test Results for Individual Windows

The data indicates that the performance when combined is much greater than when taken individually. It also shows that although the individual windows had relatively low performance as far as absolute correctness, the correct answer was usually close to the top, as indicated by the Average Rank Order.

3) Performance of Multiple Windows. In order to find the effect of incrementally adding additional windows to the system, the recognition data was recalculated as the number of windows was increased from 1 to 6.

| Windows Used | Absolutely Correct | Average Rank Order |
|---|---|---|
| 1 | 0.50 | 0.910 |
| 1,6 | 0.65 | 0.930 |
| 1,6,2 | 0.70 | 0.980 |
| 1,6,2,3 | 0.35 | 0.988 |
| 1,6,2,3,4 | 0.90 | 0.993 |
| 1,6,2,3,4,5 | 0.90 | 0.993 |

Table 3. Recognition Results from Combining Multiple Windows

B. Other Results Noted during Testing.

1) The system will identify a face with only partially-recognized facial images. In many cases, an individual did not even appear as a candidate in one or two windows, but was still identified as a result of strong performance in the other windows. The system was determined to provide a reasonable engineering approximation to the Goldschlager set completion process.

2) Gestalt calculations of negative images gave little separation, as the system paid more attention to the skin than the hair or features. This is because the system only works where black-colored pixels have the high-energy content. Humans also seem to have problems recognizing negative images. If humans are edging or cartooning the image, as some researchers suggest, then a negative image would give the same result as a positive one. Since a human is indeed sensitive to negative images, the CTT model may provide a possible explanation of why this is so.

3) The performance of the individual windows in the face recognition system provided a reasonably accurate model of human recognition performance for the same sub-parts of the face (6).

VII. SUMMARY.

The CTT Face Recognition System was performance tested with a database of 20 people. The following are some of the significant results:

1) It identified the correct person as 1st choice 90.0% of the time, and the Average Rank Order was 99.25%.

2) Provides an explanation of why the primate visual system splits images vertically before displaying them on separate right and left primary visual cortexes.

3) Highly suggests that the gestalt operation, as proposed by CTT, can indeed provide high-performance form recognition when it is coupled with the use of multiple windows on an image. This is a result predicted by CTT and borne out in this research.

The performance of the face recognition system strongly suggests CTT's general applicability to vision, and increases its credibility as a general model of human sensory information processing. The conclusion of this research is that Cortical Thought Theory is a promising new architecture with demonstrated effectiveness, worth increased research and development by those interested in developing computing systems with human-like sensory information processing capabilities.

REFERENCES.

1. Goldschlager, Leslie M. A Computational Theory of Higher Brain Function. Stanford University: STAN-CS-84-1004, April 1984.

2. Krieg, Wendell J.S. Functional Neuroanatomy. The Blakiston Company, Inc., New York, 1953.

3. Rich, Elaine. Artificial Intelligence. McGraw-Hill: New York, 1983.

4. Routh, Richard LeRoy. Cortical Thought Theory: A Working Model of the Human Gestalt Mechanism. PhD Dissertation, AFIT/DS/EE/85-1, Air Force Institute of Technology, DTIC Document, July 1985.

5. Routh, Richard LeRoy, Robert L. Russel Jr. and Matthew Kabrisky. Cortical Thought Theory: A New Computing Architecture Based on the Human Brain. Conference on Aerospace Applications of Artificial Intelligence (AAAI), 1985.

6. Russel, Robert L. Jr. Performance of a Working Face Recognition Machine using Cortical Thought Theory. Masters Thesis, AFIT/EE/85-D, Air Force Institute of Technology, DTIC Document, December 1985.

For a more complete set of references, see (6).

# Appendix B

## Autonomous Face Recognition Machine

## <u>User's Manual</u>

Owing to the autonomous nature of the machine, use of the Autonomous Face Recognition Machine (AFRM) is a relatively simple task. Once a digital image has been acquired, the operator initiates analysis of the image at the press of a button. After initiating the analysis, the operator's role is limited to deciding what to do with the various results of the analysis. The discussion in this guide follows the normal flow of processing that would usually occur. The special function of generating images that are used specifically for training the AFRM for recognition, is discussed at the end of this user's guide. A complete listing of all necessary directories, programs, links and data files is given at the beginning of appendix F of this thesis.

## Image Acquisition

To begin the analysis of any digital image, the desired image must be loaded in the Octek Image Analyzer Card (IAC). There are basically two ways to do this. One way is to load a previously saved image from memory. The other way is to load the image currently being input to the video camera.

### ACQUIRING AN IMAGE PRIOR TO AFRM STARTUP

If the user desires to load an image which has previously been saved in memory and the user does not recall the name of the image file, or the user is not sure if the image file is present in the current directory

(NSMITH), then the following procedure should be used. This procedure should be used before activating the AFRM.

To examine which image files are in the current directory, type the following at the Nova terminal (while in directory "NSMITH"):

LIST/A/S -.VD

The result of this command is a listing (in alphanumeric order) of all digital image files (files with names ending in ".VD") which exist in the NSMITH directory. If the desired file is not present in the NSMITH directory, it may be moved from directory "FACEPICS" by typing the following command (while in the directory FACEPICS) at one of the Eclipse terminals:

MOVE/V NSMITH filename.VD

As before, while in directory FACEPICS, the user may examine the files available in this directory by use of the "LIST/A/S -.VD" command.

Once the user knows the filename of the digital image he wishes to display, and has confirmed it's presence in directory NSMITH, he may display that file by using the program Octek. This program is activated by typing the following at the Nova terminal:

OCTEK

The user should select option "2" when the Octek menu appears. This option allows the user to enter the filename (which includes the ".VD" suffix) of the digital image to be loaded. Once the user enters the full filename and presses return, the image is simultaneously loaded into the IAC and displayed on the monitor. If the user accidentally selected the wrong image, he may select option "2" again and repeat the process. The user then exits the Octek program by entering "-1" and return at the keyboard.

If the user knows the name of the previously saved image file and is certain that it exists in the current directory (NSMITH), or the user wishes to acquire the desired image via the video camera, he may accomplish this as part of the initialization routine of the AFRM.

INITIALIZING THE AFRM

The AFRM is initialized by typing the following command at the Eclipse terminal (background or foreground terminal) while in directory "ESMITH":

FINDFACE

The user should verify the presence of the following successful initialization message (at the Eclipse terminal):

* * * READY TO PROCESS PICTURE DATA * * *

The above is all that need be done at the Eclipse terminal. The rest of the operator steps are performed at the Nova terminal.

At the Nova terminal the AFRM is initialized by typing the following command:

AUTOFACE

If the Nova detects that the Eclipse has not been initialized properly, it displays instructions to the operator about how to initialize the Eclipse terminal. The Nova then waits until the Eclipse has been initialized.

When the Nova detects that the Eclipse has been properly initialized it outputs the following to the operator:

```
ALL SYSTEMS READY

WOULD YOU LIKE TO SET UP AN IMAGE USING THE OCTEK
PROGRAM ? (1=YES, 0=NO) :
```

ACQUIRING AN IMAGE AFTER AFRM STARTUP

If the user wishes display a previously saved image file, or he wishes to acquire an image through the video camera and save that image in memory before proceeding, then he should select "1". Retrieval of a previously saved image is performed as discussed earlier, except Octek does not give the user an opportunity to see what image files are currently available. The user must know beforehand the exact name of the image file he wants to display. If the user wishes to acquire an image via the video camera and save it to memory, he must perform the save at this point in the process. The user will be given another opportunity to acquire an image via the video camera at a later point in the algorithm. Thus, if the user simply wishes to acquire an image via the video camera, and is not concerned about saving the image, then he may skip this option by selecting "0".

ACQUIRING AND SAVING AN IMAGE FILE

If the user desires to acquire an image and save it to memory, then he must use the Octek program. The user should ensure that the video camera has been turned on and warmed up a few minutes prior to attempting to acquire the picture. Once the Octek menu is displayed, the user may acquire the image by selecting option "1" from the menu. This action causes the monitor to display (in real time) the image being generated in the video camera. At this point, the user should make the following adjustments at the camera:

1. F-Stop setting of F8.0
2. Focus at 30 ft.
3. Zoom - as appropriate (see below).

If faces are present in the desired image, the operator should ensure two things for successful program performance. First, the eyes of any subjects should be located in the central part of the screen. Figure B-1 illustrates the area of the image in which the eyes (both of them) must appear. Second, since the program only finds faces within a certain range of facial sizes, the zoom of the camera must be adjusted so that the subject faces fall within this range. As a rule, the facial image should not be larger than 3.5 inches (when measured from bottom of chin to top of head, on the TV screen) and should not be smaller than 2.0 inches.

When the previous adjustments have been accomplished, the user then freezes the image by hitting the "return" key at the Nova terminal. He may then save the image displayed

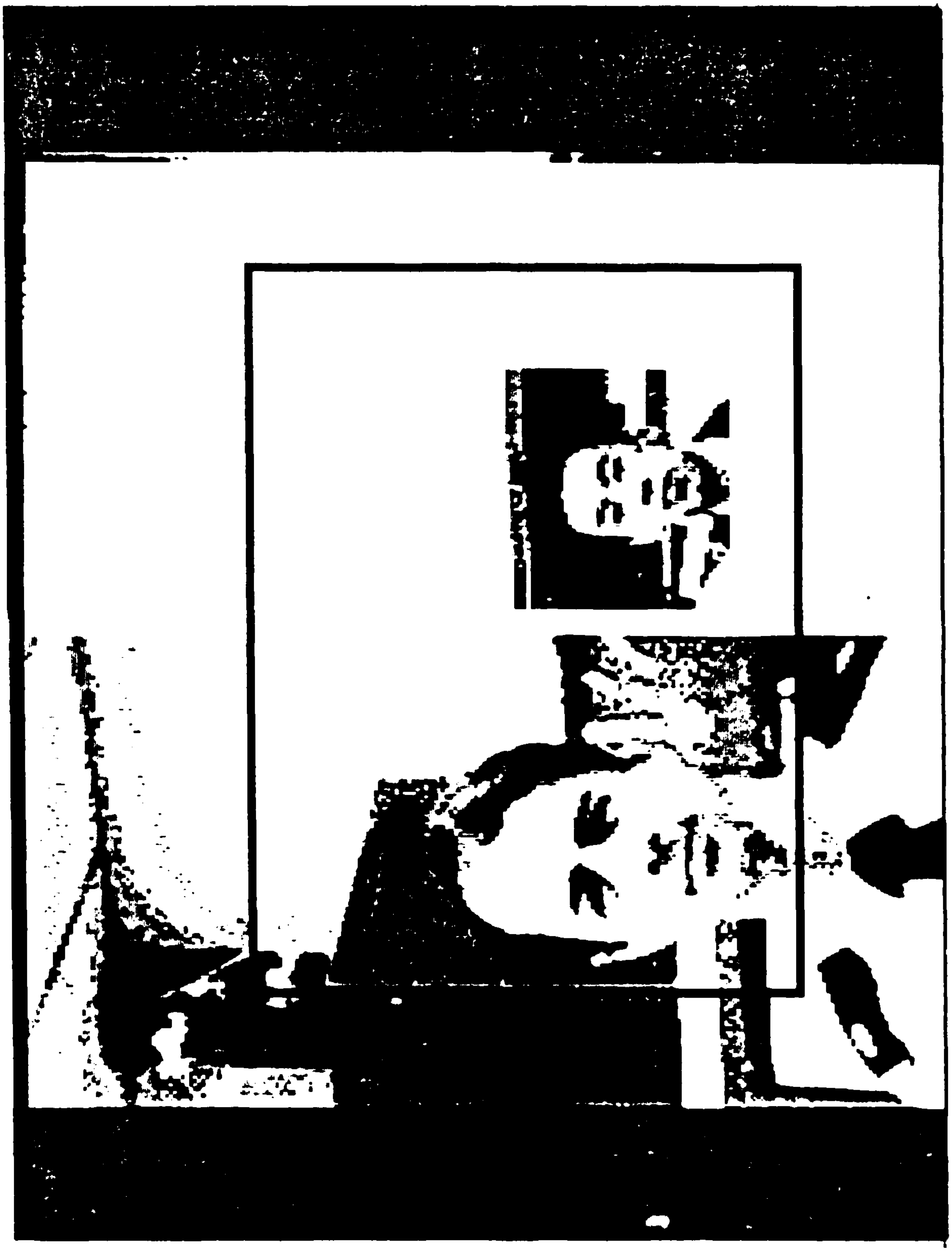


Figure B-1. Illustration of the Search Area.

by selecting option "3" on the Octek menu and entering an appropriate filename for the image. The user then exits the Octek program by entering a "-1" at the terminal. This action returns the user to the main AFRM algorithm.

At this point, initialization of the AFRM is complete and the following AFRM menu is displayed:

```
Autonomous Face Recognition Machine.

Keypad Menu
----------------------------------------
    1   2   3   4       5   6   7   8
----------------------------------------
  1--camera #1 on
  4--camera #1 off
  6--process picture
  8--- terminate and exit to system
```

Here the user is given an additional opportunity to acquire an image via the video camera. If the user desires to do so, he must ensure that the camera is on and warmed up (about 10 minutes) before selecting keypad button #1 on the Octek keypad. Note that the previous instruction refers to the Octek keypad, and not the terminal keyboard "1". The user must still ensure that the proper camera adjustments are performed as described earlier. Once the user is satisfied that he is ready to take the picture, he need only press the Octek keypad button #4. Again, the user is cautioned that an image acquired in this fashion will not be saved. The image will be altered, and eventually erased, once the picture is processed.

Image Processing

Assuming that the image that the user desires to be analyzed is now loaded in the IAC, he is now ready to begin the process of analyzing the image for the presence of faces. To begin this process the user must press the Octek keypad button #6.

ANALYSIS OF THE SEARCH AREA

Analysis of the search area (illustrated in figure B-1) proceeds from the bottom to the top of the search area. This process can take anywhere from 5 to 30 minutes depending upon the nature of the background in the image search area. During this time, the user is kept informed concerning where the program is in the analysis. At the Eclipse terminal, the user is constantly informed about which row of the search area is currently being analyzed. The analysis proceeds from the bottom row (row 176) to the top row (row 56) in increments of 2 rows.

As possible eye signatures are located, messages are displayed at the Nova terminal indicating the event. As the Nova extracts 64 by 64 image sub-sections (for use in further analysis by the Eclipse), it displays a box cursor on the TV monitor which outlines the area of interest. Upon completion of analysis of the image sub-section by the Eclipse, the Nova displays messages indicating the results of the analysis. If a face was successfully found in the image sub-section, then a message to that effect is displayed. In addition, the Nova places a set of grid marks on the displayed image, at the location where the face was found. These grid marks indicate the internal region of the face and the locations of certain facial features. The facial feature locations indicated by the grid marks are:

1. Left and right sides of the eyes
2. Center of face
3. Top of nose
4. Center of mouth

The two horizontal grid marks located above the nose mark do not correspond directly to any facial feature. The position of these marks are determined by the distance between the two maxima in the eye signature and the top of nose location. By adding the distance between the two maxima, once, to the top of nose location, the grid mark centered about the level of the eyes is determined. By adding this distance, twice, to the top of nose location, the top horizontal grid mark is obtained. Figure B-2 illustrates an example of the grid marks displayed when a face is found.

The AFRM will process up to four facial images, for any single test image. As each image is found, in addition to the actions indicated above, the Nova will save (to disk) the 64 by 64 image sub-section containing the facial image. The first such image is saved to disk as the file "TEST1.PI", the second such image is saved as "TEST2.PI", and so on. Appended to these image files are the data

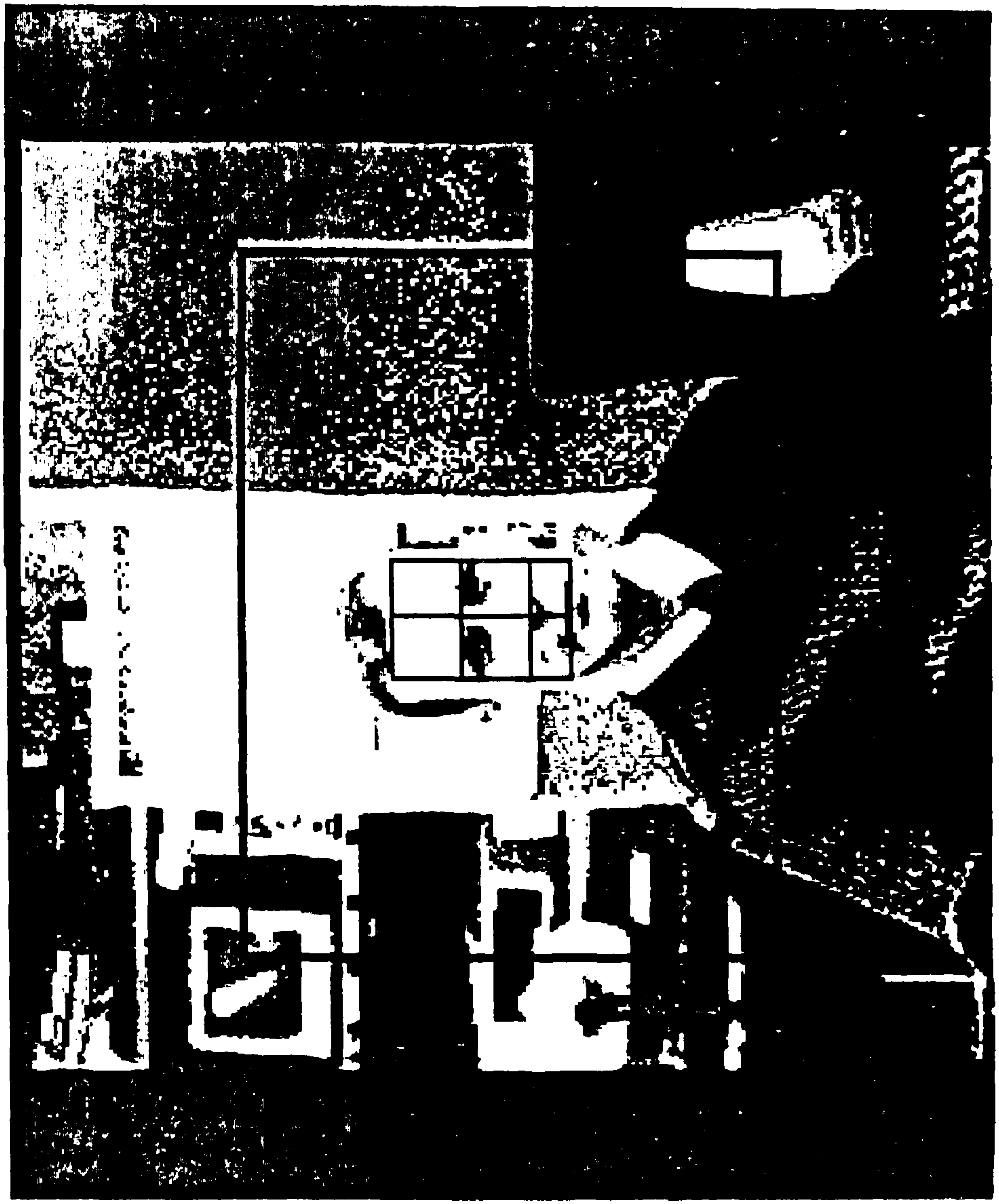


Figure B-2. Example of Grid Marks Indicating a Face was Found.

indicating the locations for the various features (grid marks) that were found, and the enhancement multiplier used.

If a face was not found, or one was found but was too large to fit in the 64 by 64 image, the operator is informed of these events by messages to the Nova console. The AFRM then continues with the analysis of the search area until four facial images are located or analysis of the search area has been completed.

If, at any time after the first facial image has been found, the user chooses to abort the rest of the search, he may do so by pressing the Octek keypad button #7. This action will immediately terminate the process of searching for faces in the test image. The Nova console then displays the following message and prompt:

FACES FOUND WITHIN THE BOX CURSOR AREA = "X"

WOULD YOU LIKE TO PRINT THIS IMAGE ? (1=YES,0=NO) :

The value of "X" is determined by the number of facial images found. The operator is given the option to print the image containing the results of the analysis. If he chooses to print the image, he must wait while the image is being printed (1 or 2 minutes), before proceeding with any other operations. The operator is forced to wait while the image is being printed because the Eclipse is incapable of performing other operations at this time. If the Nova were allowed to proceed, it might very well get out of (task) synchronization with the Eclipse. If no faces were found, the program on the Nova then returns to the initial AFRM menu, and is ready to start again, or quit, as the operator decides.

If faces were found in the image search area, then after responding to the user's selection concerning printing the image, the following prompt is displayed at the Nova console:

Would you like to :

1 - TRAIN WITH THESE FACES ?
2 - IDENTIFY THESE FACES ?
3 - QUIT ?

The operator should then enter a 1, 2 or a 3 (on the Nova keyboard) depending on which option he chooses. If the operator chooses option 3, then the AFRM is re-initialized and the initial menu is displayed again. The operator may then acquire a new image and start the process over again, or terminate and exit to the system.

GENERATING FACIAL GESTALTS

If the operator chooses option 1 or 2, the original test image is removed from the screen, and a new display is generated which illustrates the enhancement and facial windows used in calculating the facial gestalt values. The order of processing follows the order in which the faces were found in the original image. Thus, the image stored in image file "TEST1.PI" contains the first image processed. Figure B-3 illustrates the results of this process.

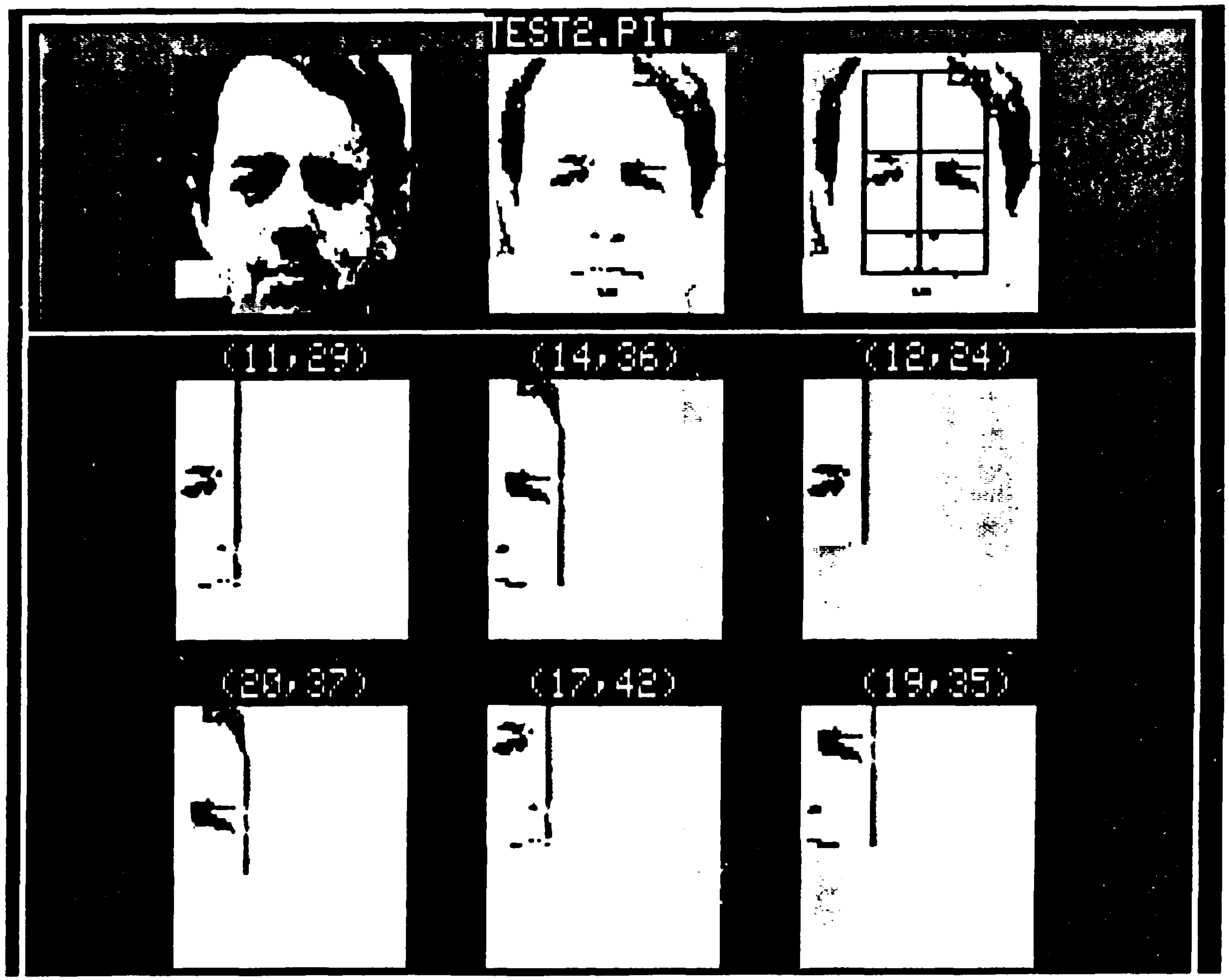



Figure B-3. Example of Facial Window Gestalt Display.

After determining the facial window gestalts for each facial image, the AFRM proceeds in a different manner depending upon whether the operator chose to train or identify the facial images.

STORING THE GESTALT RESULTS

If the operator chose to train with the facial images, then he must supply additional information so that the data may be properly stored in the facial database. The first thing the operator must do is respond to the following prompt:

PLEASE ENTER A NAME FOR THIS PICTURE FILE:

The operator responds by entering a descriptive name for the file which contains no more than 8 characters, plus the ".PI" extension. For example, the last name of the individual illustrated in figure B-3 is "King". Thus, an appropriate filename might be "KING.PI". If a file already exists in the directory which has this filename, then the following prompt is displayed:

```
* * * FILE ALREADY EXISTS * * *

WHAT WOULD YOU LIKE TO DO ?

     1 - TRY ANOTHER FILENAME
     2 - WRITE OVER EXISTING FILE
```

If the operator chooses "1", then he is given another opportunity to enter a filename. If option "2" is selected, the program deletes the existing file, and creates a new file (under the chosen filename) containing the data from

TEST1.PI. The operator is then required to respond to the following prompt:

Do you want to store this record ? (1=Yes,2=no):

If the operator wishes to store the window gestalts for the facial image which has just been processed, then he responds with a "1". Otherwise, he enters a "2", and the AFRM proceeds to the next facial image.

If the operator selects to store the window gestalts, he is presented with the following menu options:

Please enter the ID number of this person:

1 - Enter ID number
2 - Add a New Name
3 - Use last ID number in the system
(LAST ID NUMBER = 20 CAPT ED SMITH)
9 - Look at or Edit Previous Records

If the subject, to whom the face belongs, already has an ID number, but the user does not recall what the ID number is, he should select option "9". This option provides a listing of all the individuals (currently in the system) and their associated ID numbers. He may then return to the above menu and select option "1" to enter the desired ID number of the individual.

If the subject has not been previously entered into the database, then option "2" should be selected. The operator must then respond to various prompts for personal information associated with the subject, such as the subjects name, default picture filename, and speech synthesizer string. The speech synthesizer string determines what the computer will verbally say (via the DECTALK speech synthesizer) when it has determined the identity of the individual (in any future indentification tests). Entries for the speech synthesizer are written pretty much the same as you might write a sentence, such as "Hello, Captain Smith. How are you today?". Sometimes the spelling must be altered to obtain the proper enunciation.

Once the system has an ID number to associate with the subject's facial window gestalt data, the data is stored in the facial database under the given ID number. This data will be used to train the system the next time the database statistics are updated. It is important to note at this point that the system has not been actually "trained" with the image data. The data has merely been stored in the database. To actually train the system, the user must select to update the database statistics using the program "MAIN". This process is accomplished after the AFRM has been terminated. Instructions for the use of the program MAIN are contained in Russel's thesis (22:B-39) and are not repeated here.

At this point the system retrieves the next facial image that was found and proceeds, as before, to generate the facial windows and associated gestalt values for the new image. These values are then stored as described previously. This process continues until all the facial images that were found, have been processed. The system is then re-initialized and the initial AFRM menu is re-displayed to the operator. The operator may then choose to acquire a new test image for subsequent analysis or to exit to the operating system.

IDENTIFYING AN UNKNOWN SUBJECT

If the operator chose to identify the facial images that were originally found by the analysis of the search area, the program proceeds in a different fashion (than for training). After determining the facial window gestalts (as illustrated in figure B-3) for a particular TESTX.PI image, the AFRM immediately proceeds to the "identification phase" of the process (for the facial image in question). If the operator wishes to use the DECTALK speech synthesizer during this phase, he must activate it prior to the identification phase. This is done by turning on the DECTALK unit and ensuring that the appropriate communication channels have been opened between the Eclipse system and the DECTALK. The DECTALK does not have to be activated in order for the identification process to proceed.

The identification process generates a new display on the TV monitor. An example of this display is indicated in figure B-4. The test image (containing the unknown subject) is displayed in the upper left corner of the display. The results of the analysis are displayed as follows:

1. A default picture, representing the computer's first choice for who the subject is, is displayed immediately below the image of the unknown face.
2. The name of the computer's first choice is displayed to the right of this #1 candidates default image.
3. Default images for the 1st, 2nd and 3rd runner's up

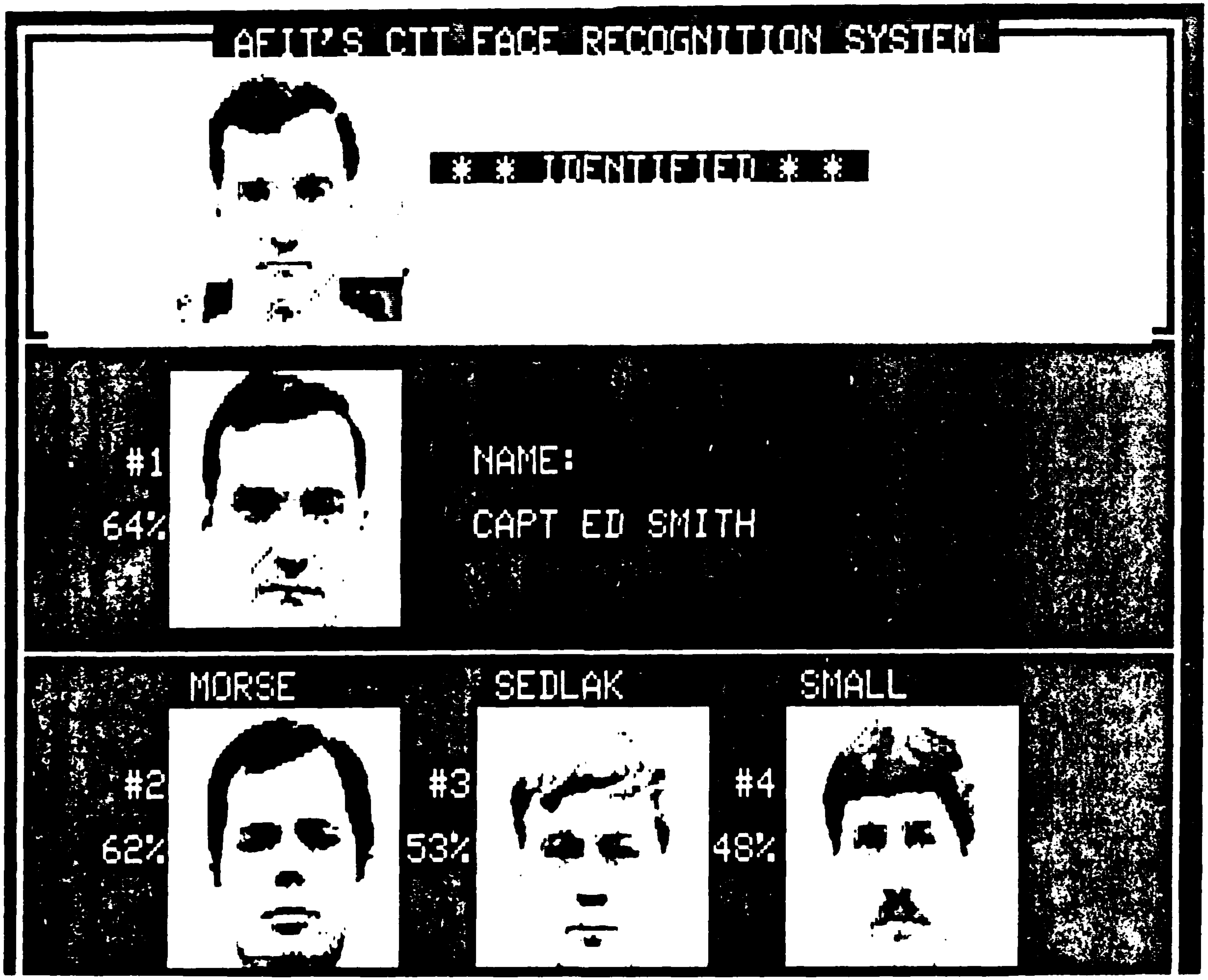



Figure B-4. Example of Identification Display.

are displayed below the #1 candidate's image.

4. To the left of each candidates default image, a pseudo-probability figure is displayed. This figure is a measure of how certain the computer is, that it has correctly identified the unknown individual. The larger the percent difference between the #1 candidate and the runner's up the more certain the computer is of it's selection.

Along with the display of the results, the Eclipse prints out the results of the identification analysis, on the printer. At the completion of the identification analysis, if the DECTALK has been activated, the computer will verbally announce the name of the #1 candidate along with a verbal greeting.

The only option available to the operator, during the identification phase, is whether or not to print the image which displays the identification results.

This entire process is repeated for each facial image that was found in the analysis of the original test image. Once all faces have been processed, the AFRM is re-initialized and is ready to begin again.

## Training Image Generation

The following discussion concerns the generation of images used specifically to train the AFRM recognition database. Images such as the one illustrated in figure B-5 were the type used to train the AFRM recognition database. These images were constructed using the program PICTURE2. This program (which is fully described in appendix B of Russel's thesis) allows the user to construct composite images like that illustrated in figure B-5.

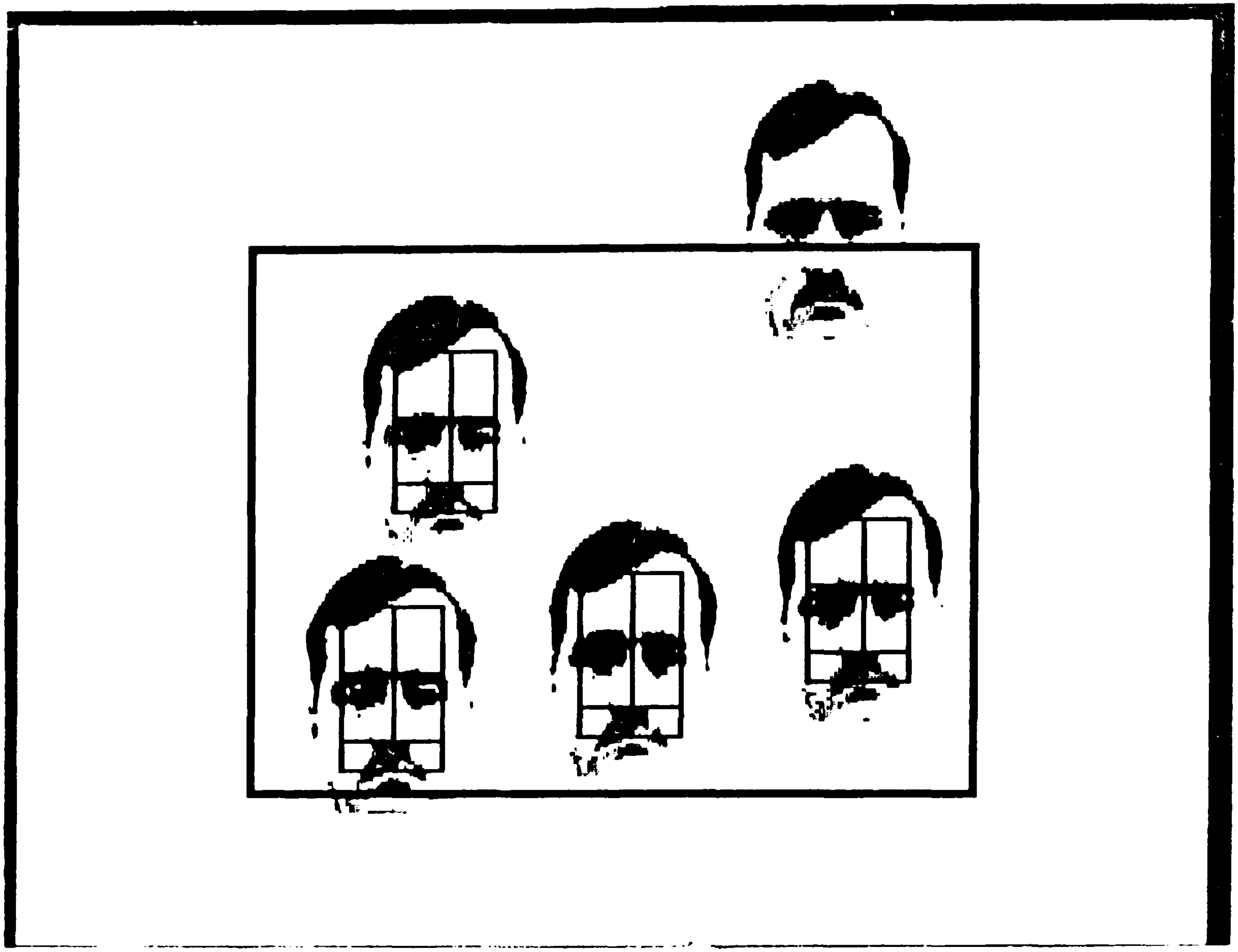


Figure B-5. Example of a Training Image.

After activating the PICTURE2 program, the training images are generating by first selecting the additional options menu (Octek keypad #5) from the PICTURE2 main menu. The TV monitor screen is then cleared (to a white background) by selecting option #4 while in the additional options menu. This action also returns the user to the PICTURE2 main menu. The 64 by 64 box cursor is then activated by pressing the Octek keypad button #7. Once activated, the box cursor which is now visible on the screen, may be positioned anywhere on the screen. Positioning of the box cursor is accomplished using the upper Octek keypad buttons. The box cursor should be positioned, for each subject facial image, such that the subjects eyes will fall within the search area indicated by the large rectangular box in figure B-5.

After positioning the 64 by 64 box cursor at the desired position, the user should again select the additional options menu by pressing the Octek keypad button #5. He may then select to retrieve and display a 64 by 64 facial image which has previously been acquired (by the program PICTURE2) and saved as file named "filename.PI". The program then displays the chosen facial image, on the screen, at the chosen location of the 64 by 64 box cursor. This process of retrieving and displaying a facial image may be repeated until an image like figure B-5 is generated. The resulting image may then be saved using the OCTEK program, which saves the full screen image. Subsequent to this, the image may be analyzed by the AFRM, and the resulting facial images processed and stored in the AFRM recognition database.

Since all of the facial images in the face databank (in excess of 300 images representing more than 50 subjects) were initially acquired and saved by PICTURE2, as 64 by 64 images, this allows the AFRM to access all of that data.

It is recommended that future potential users continue to use the PICTURE2 program to acquire facial images that will be used for training. This will keep the facial image databank (found in directory FACEPICS) consistent.

# Appendix C

## Facial Signature Examples

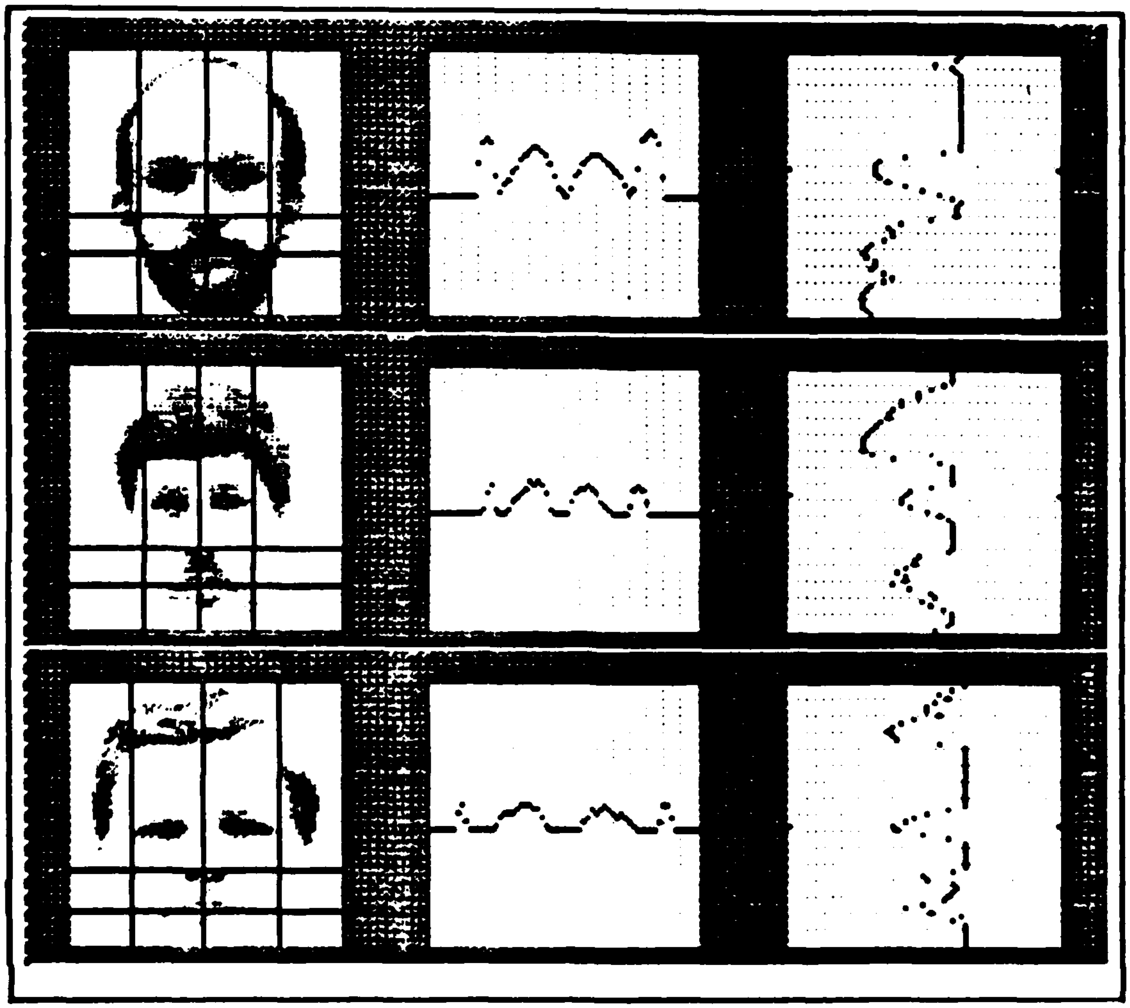


Figure c-1. Facial Signatures Sample #1.

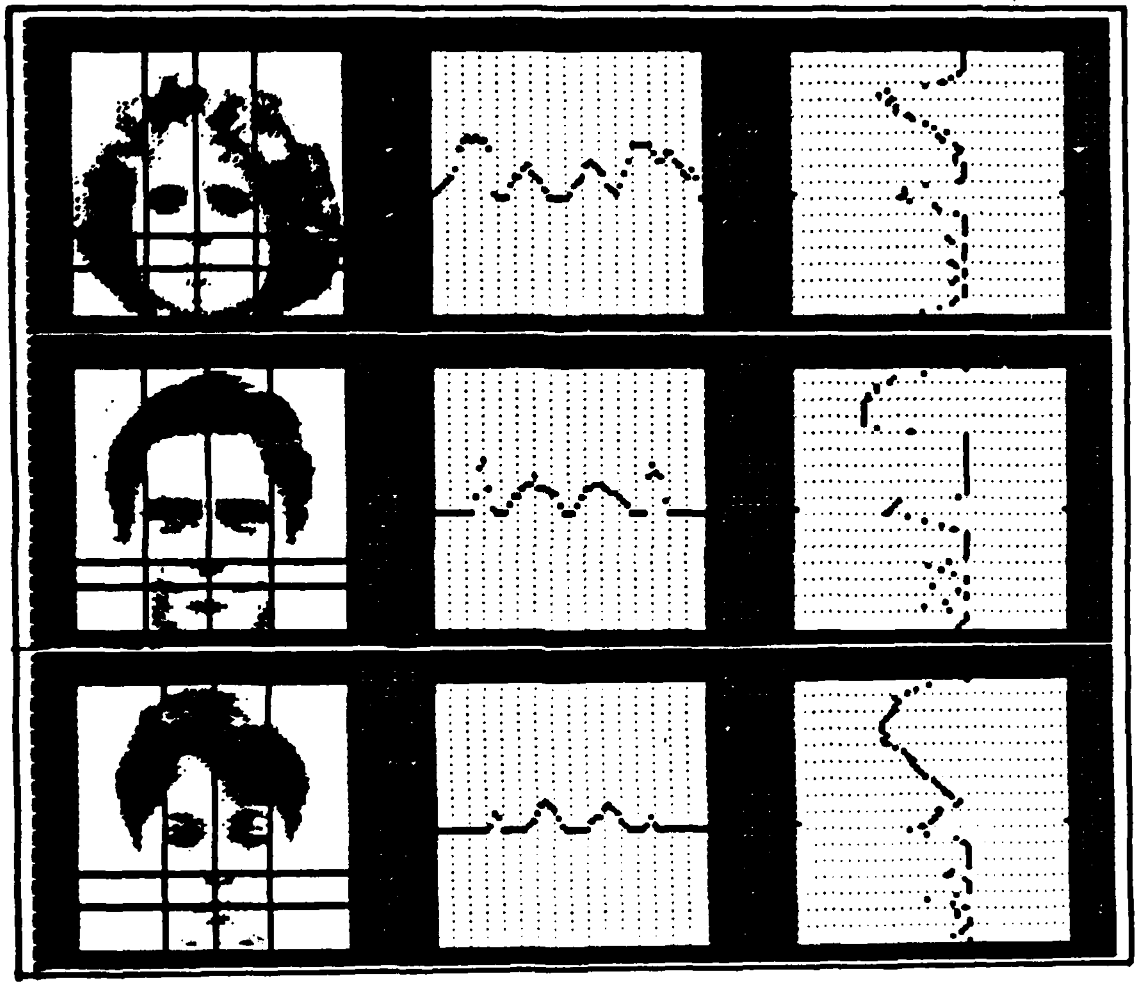


Figure c-2. Facial Signatures Sample #2.

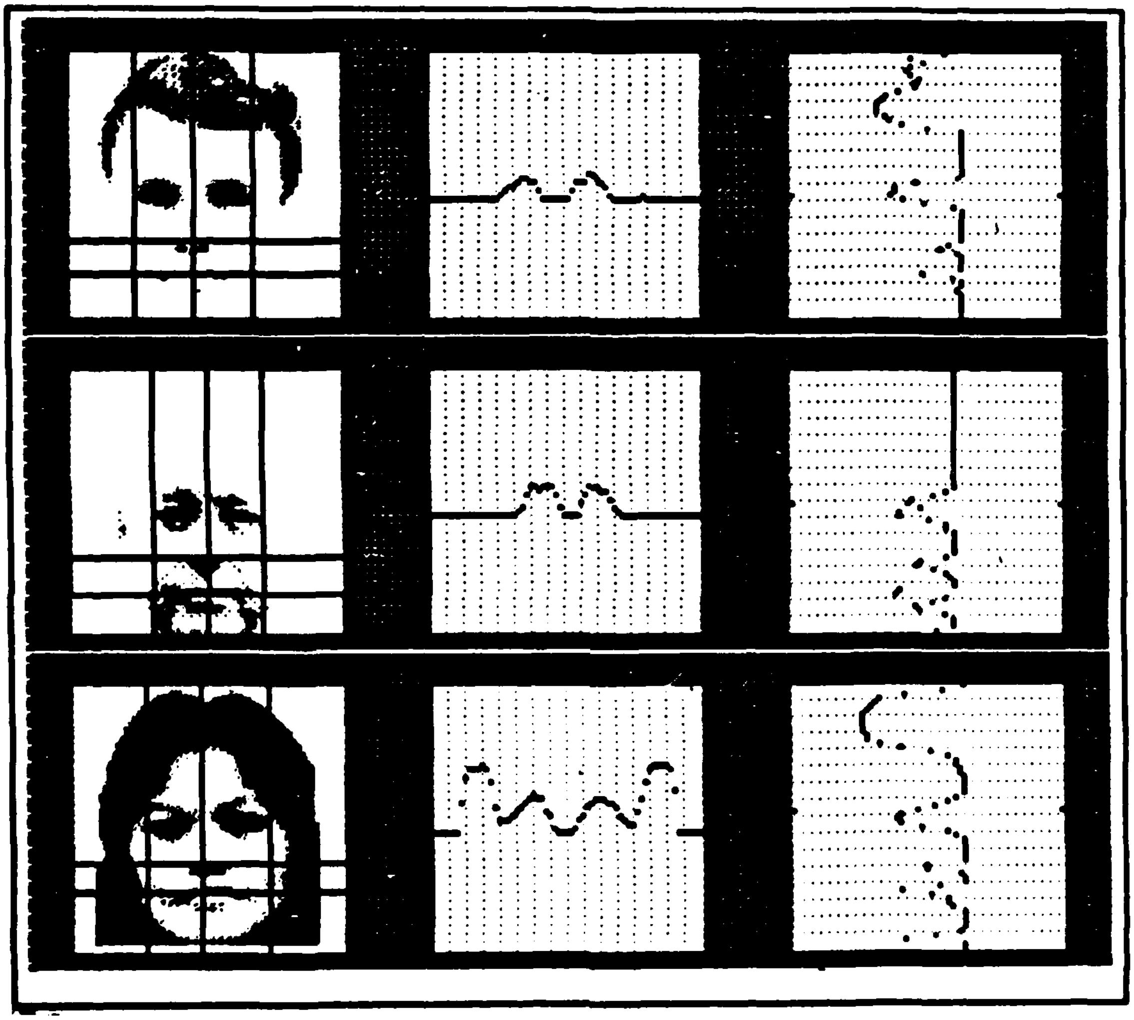


Figure c-3. Facial Signatures Sample #3.

# Appendix D

## Face Image Samples from the Data Bank

Figure D-1. Facial Image Samples #1.

Figure D-2. Facial Image Samples #2.

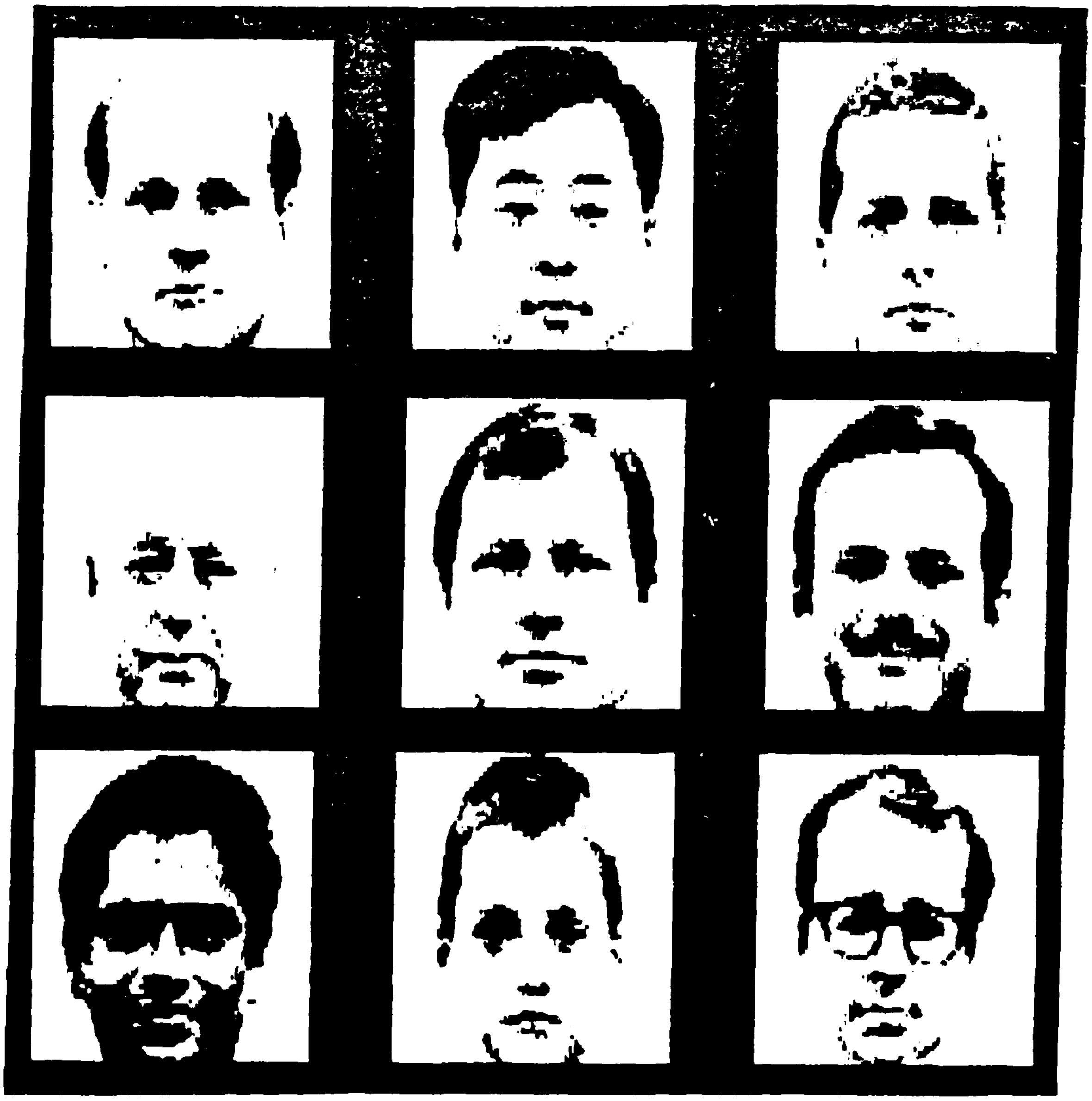


Figure D-3. Facial Image Samples #3.

# Appendix E

## Face Locator Test Images

Figure E-1. Test Image #1.

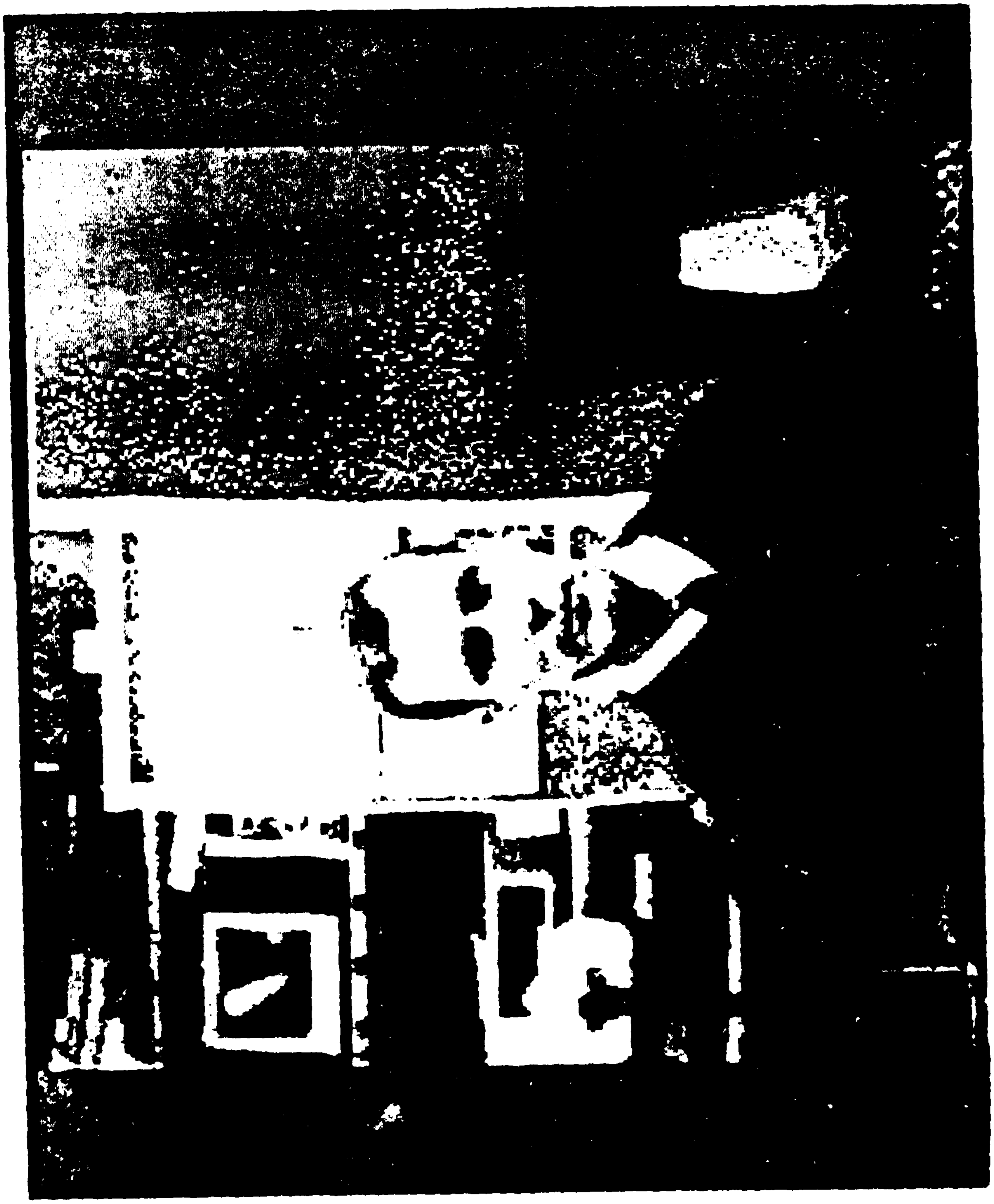


Figure E-2. Test Image #2.

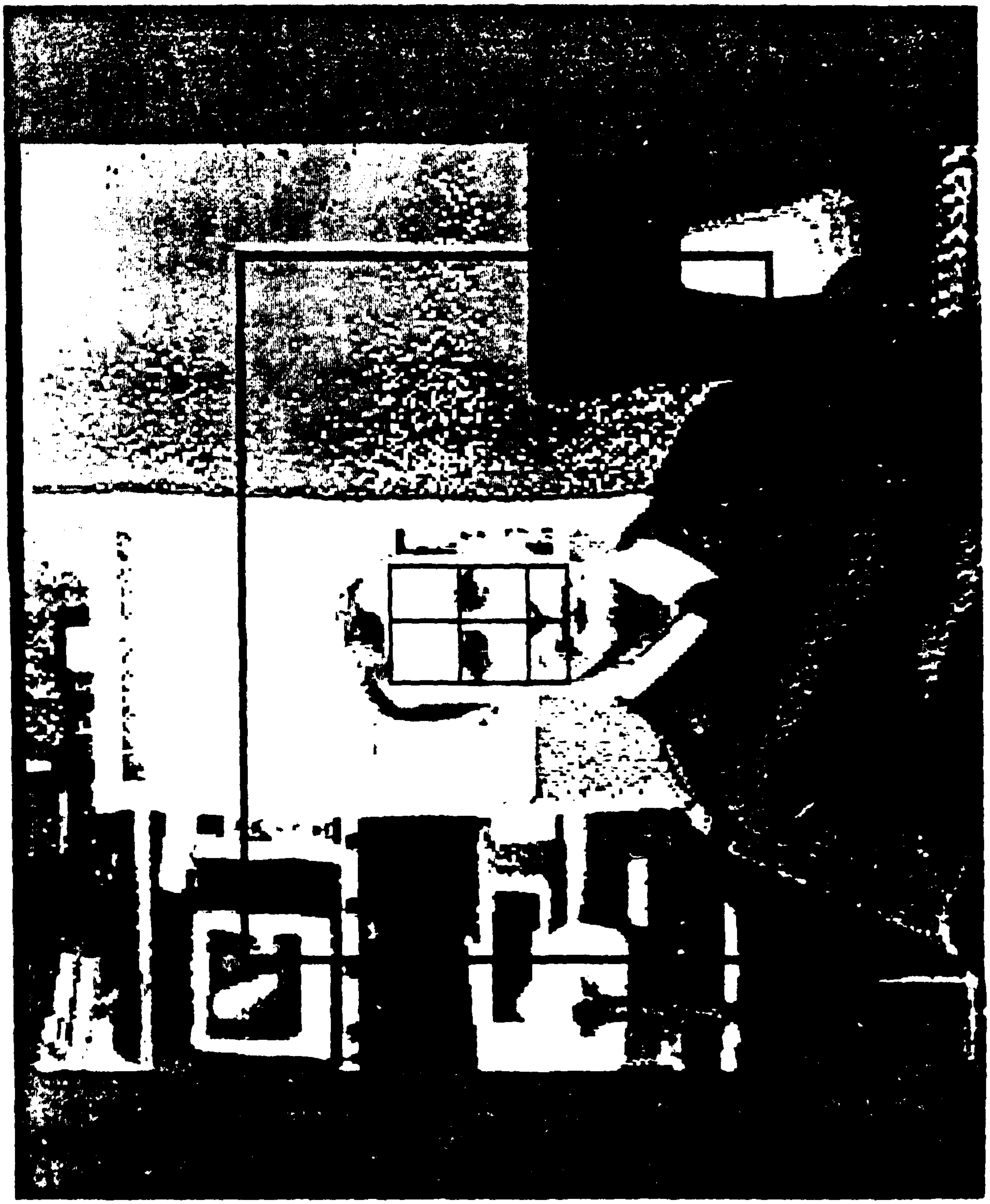


Figure E-3. Test Image #2 Results.

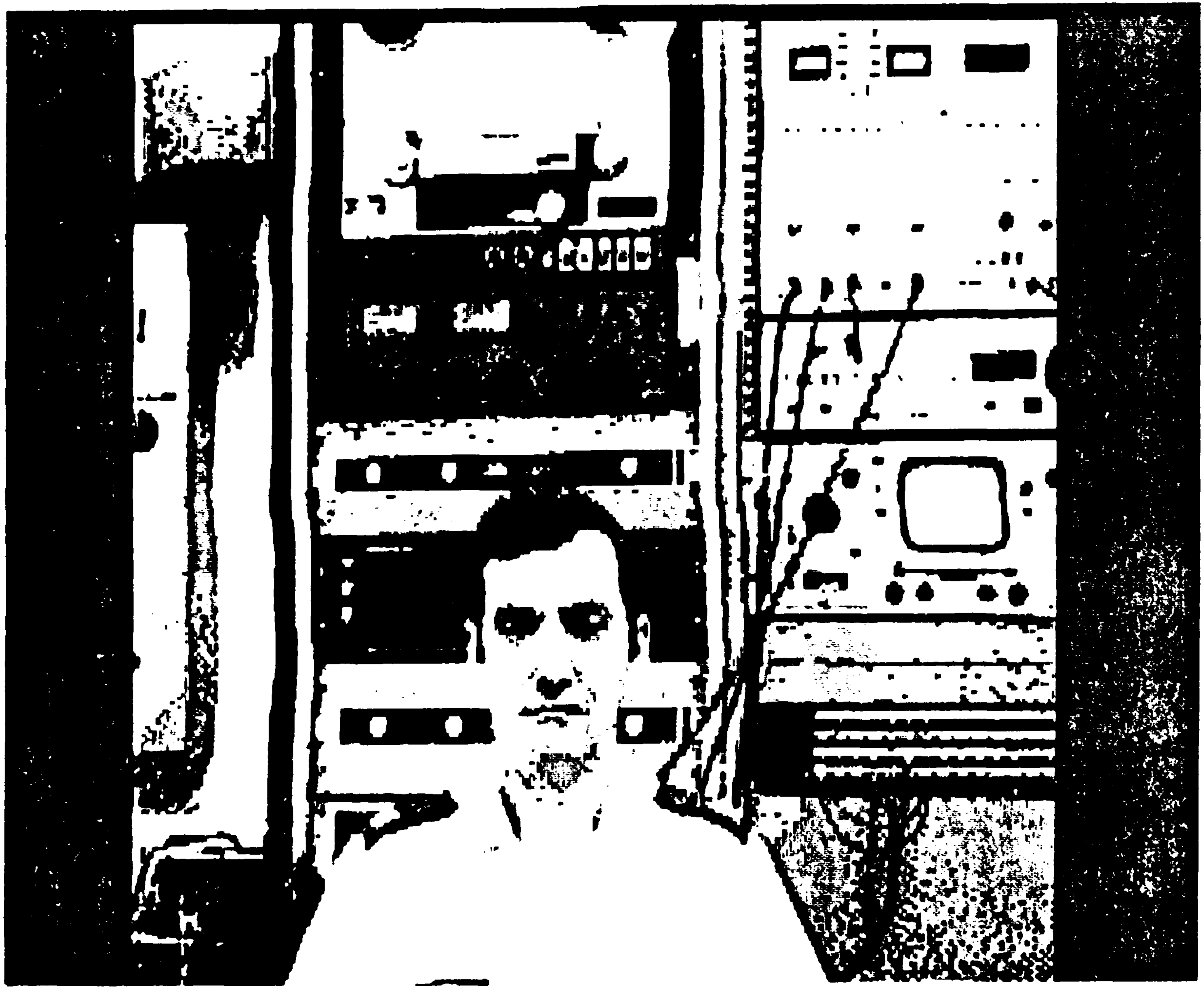


Figure E-4. Test Image #3.

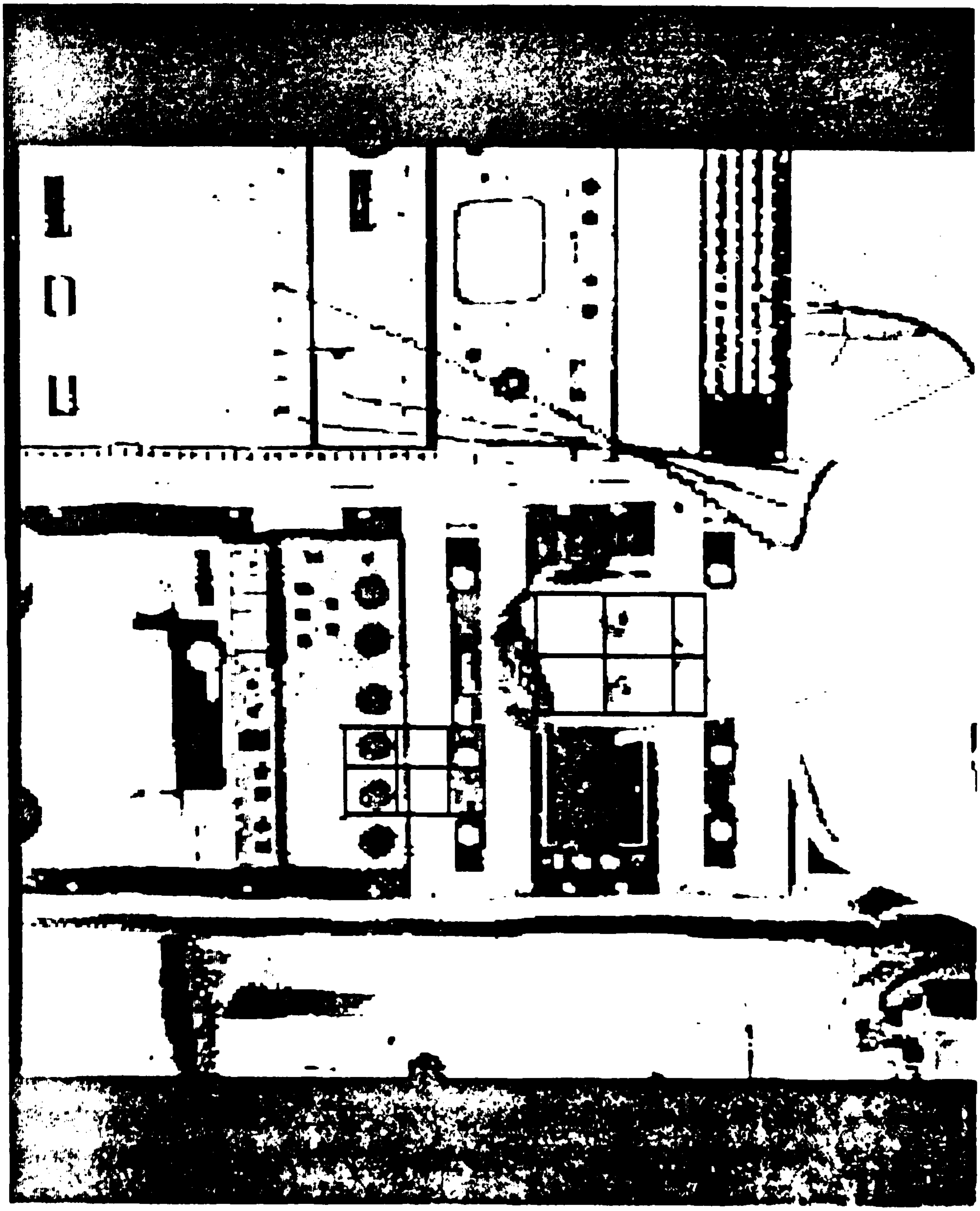


Figure E-5. Test Image #3 Results.

Figure E-6. Test Image #4.

Figure E-7. Test Image #4 Results.

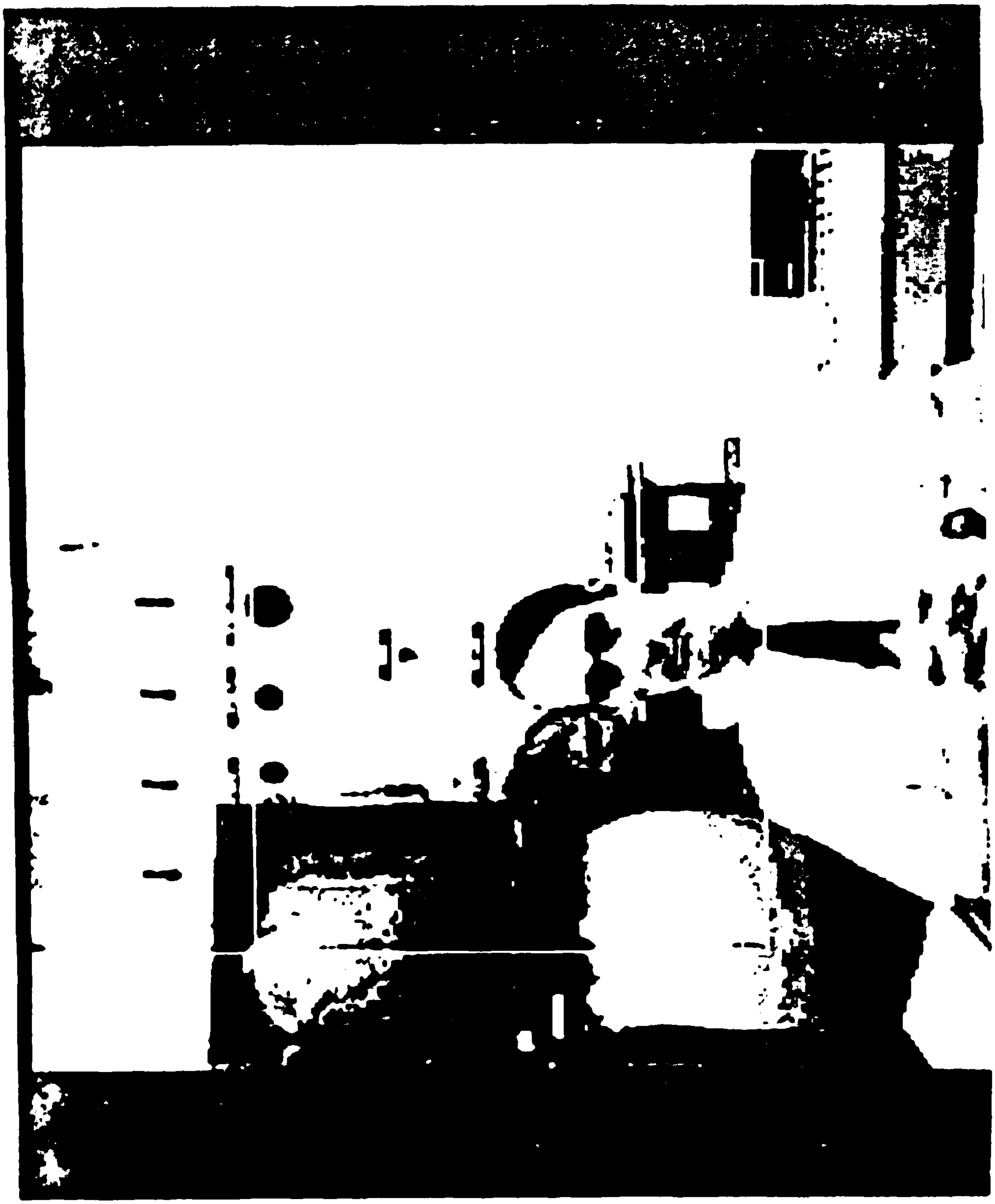


Figure E-8. Test Image #5.

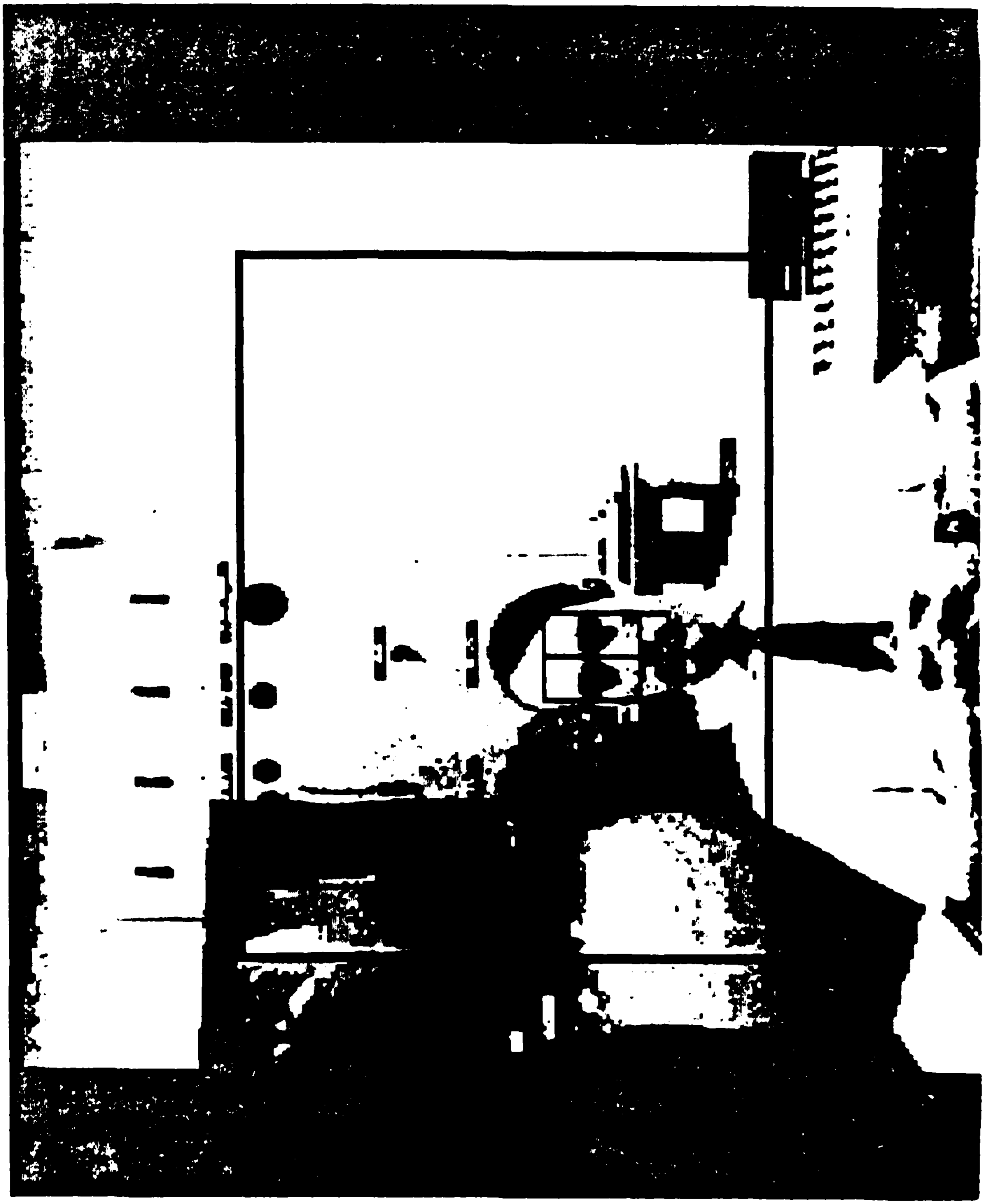


Figure E-9. Test Image #5 Results.

Figure E-10. Test Image #6.

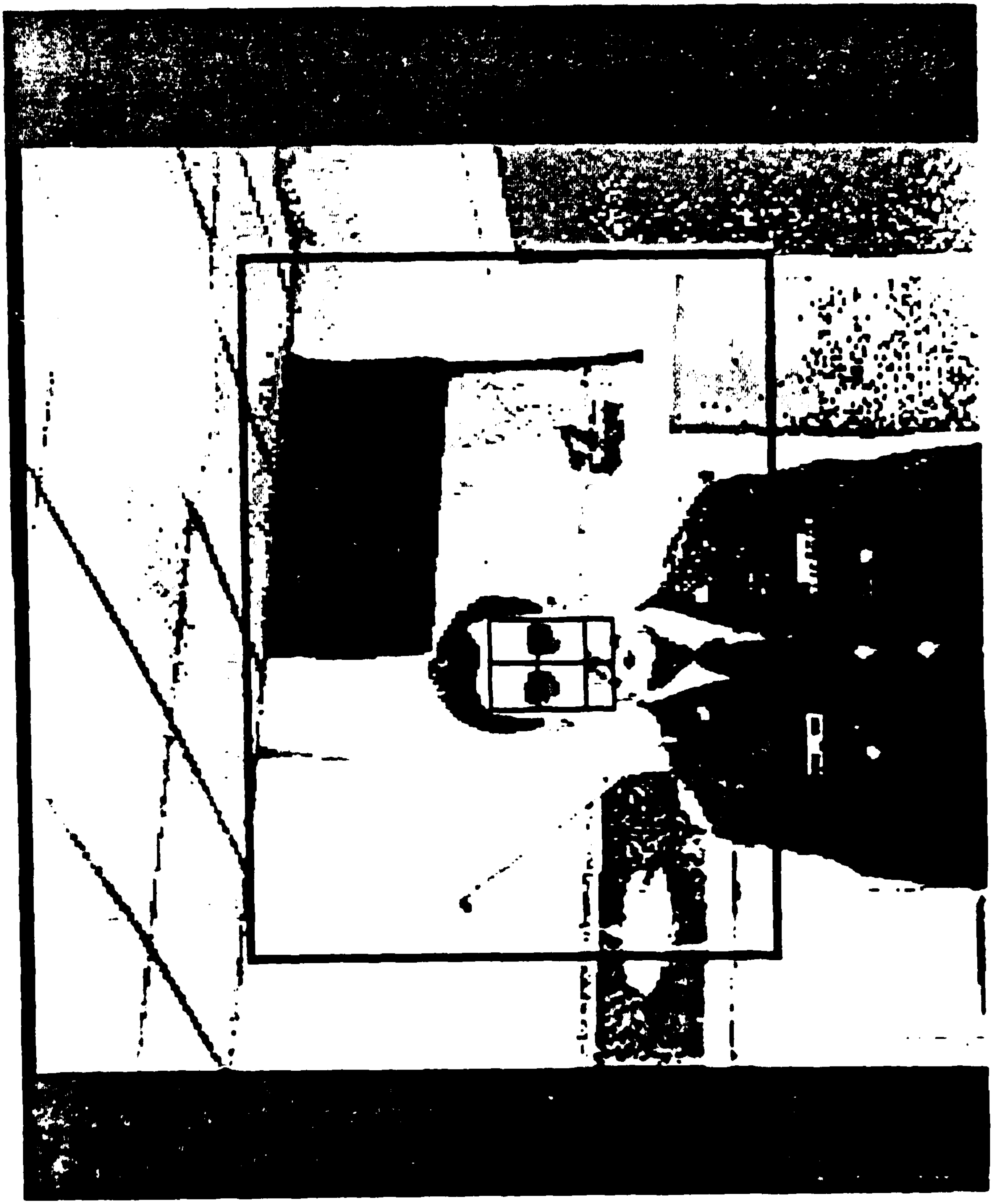


Figure E-11. Test Image #6 Results.

Figure E-12. Test Image #7.

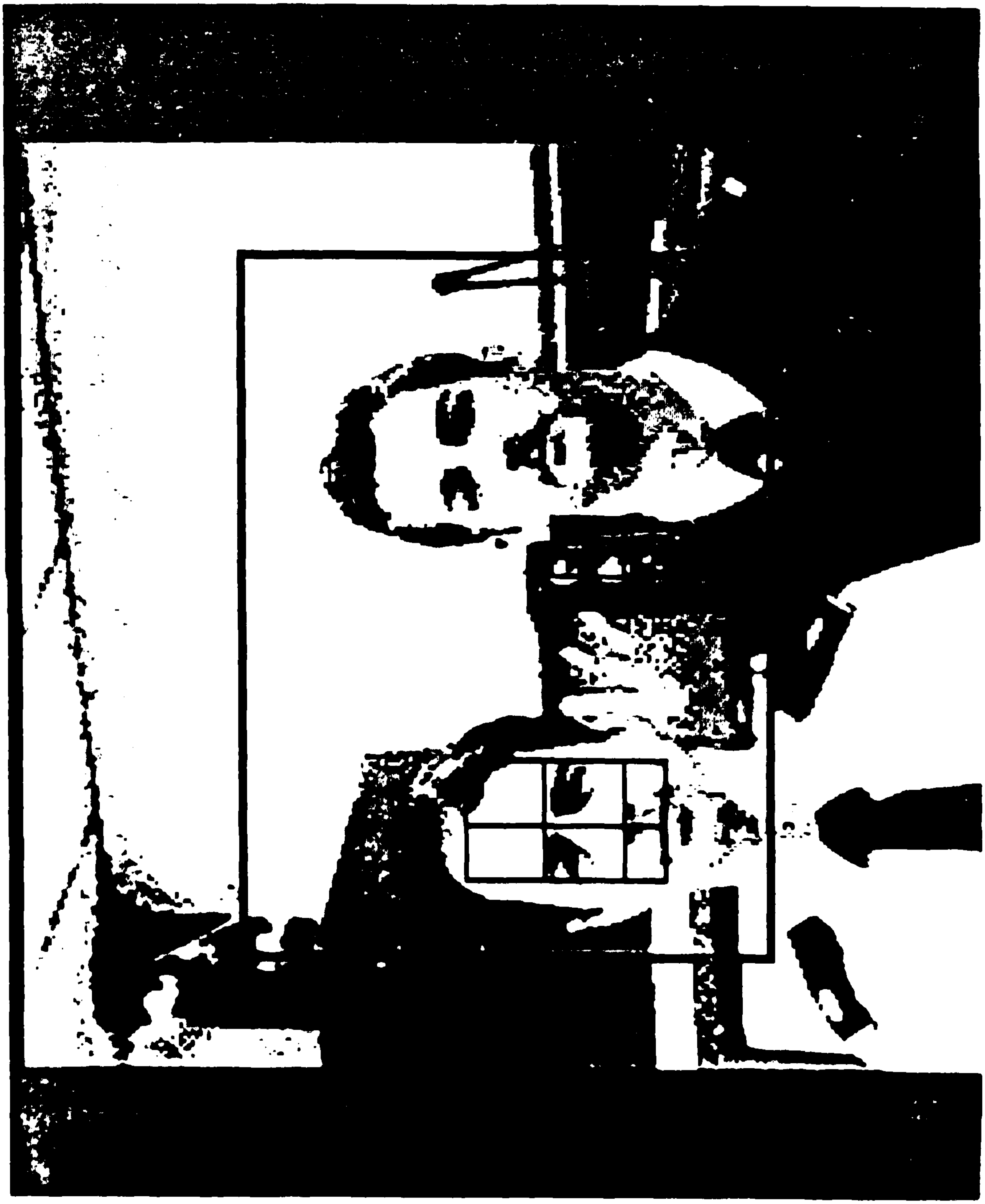


Figure E-13. Test Image #7 Results.

Figure E-14. Test Image #8.

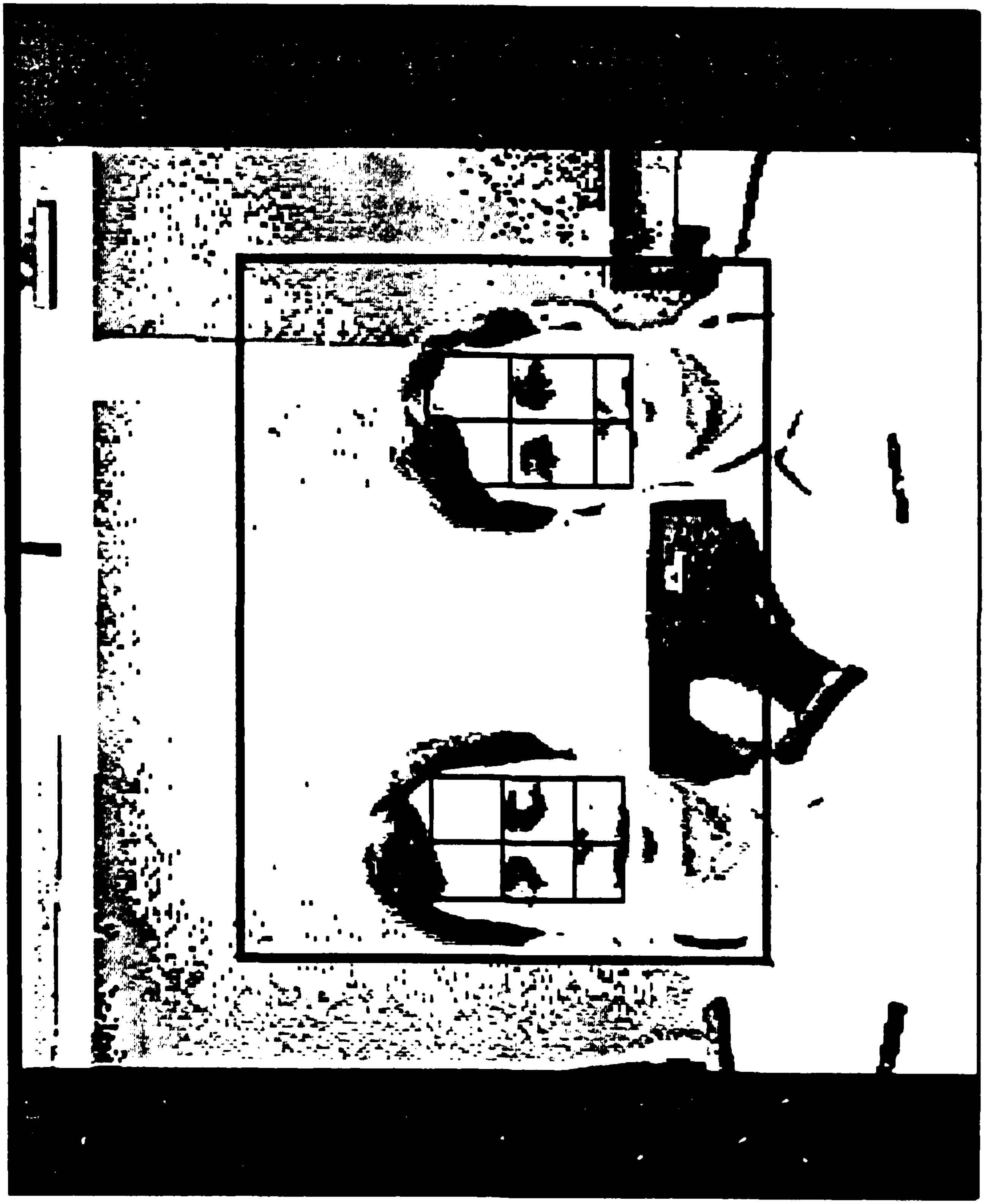


Figure E-15. Test Image #8 Results.

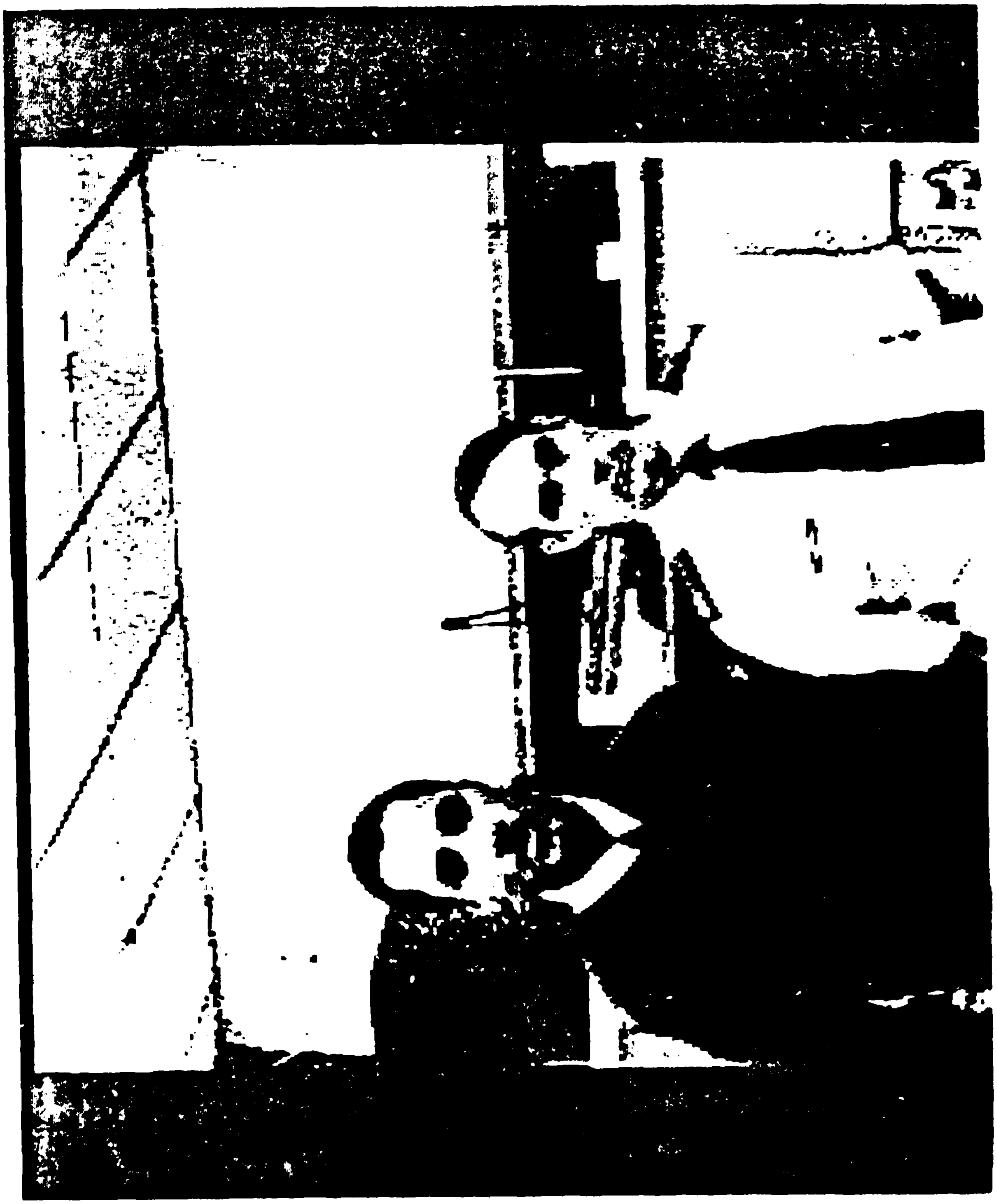


Figure E-16. Test Image #9.

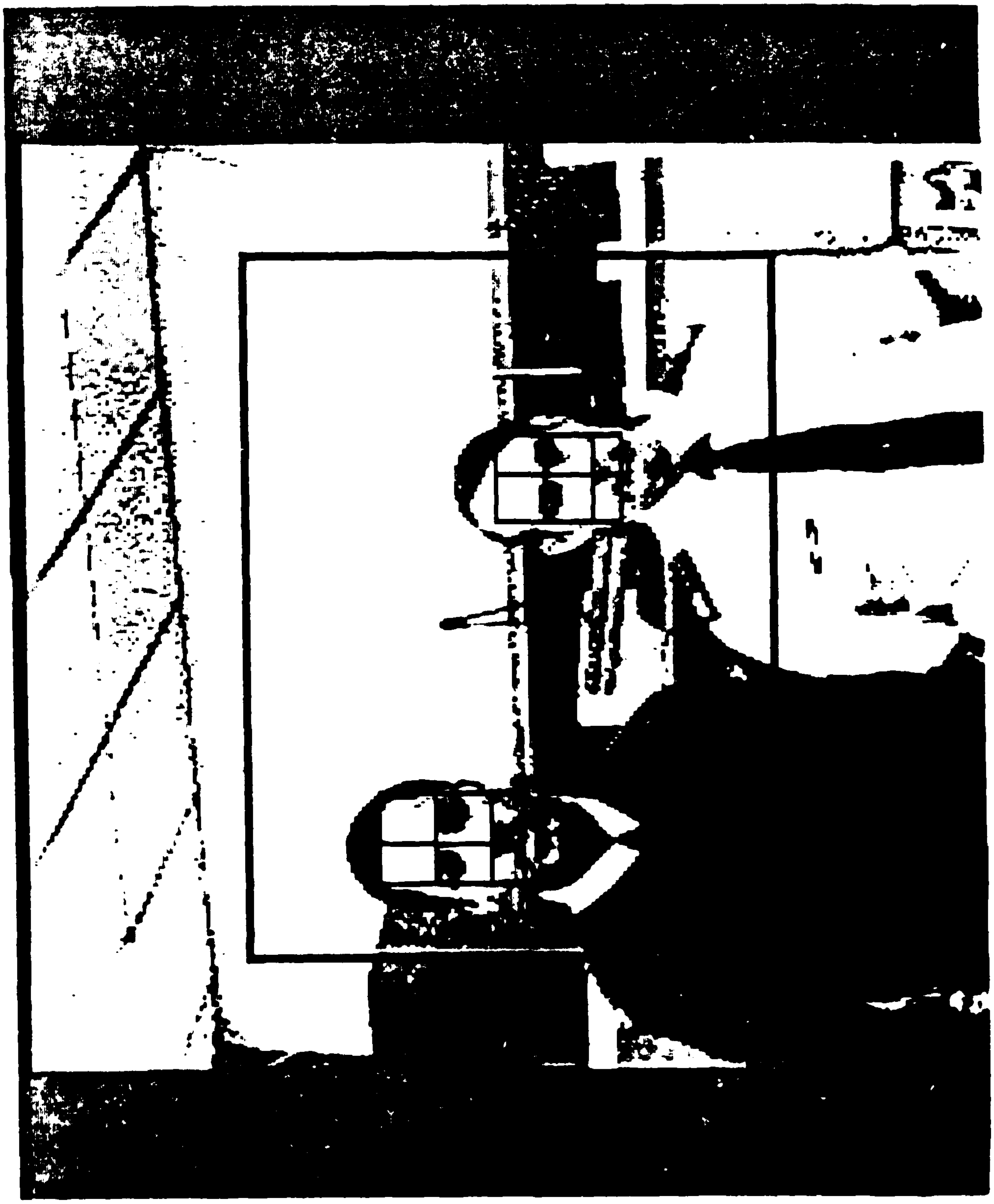


Figure E-17. Test Image #9 Results.

Figure E-18. Test Image #10.

Figure E-19. Test Image #10 Results.

Figure E-20. Test Image #11.

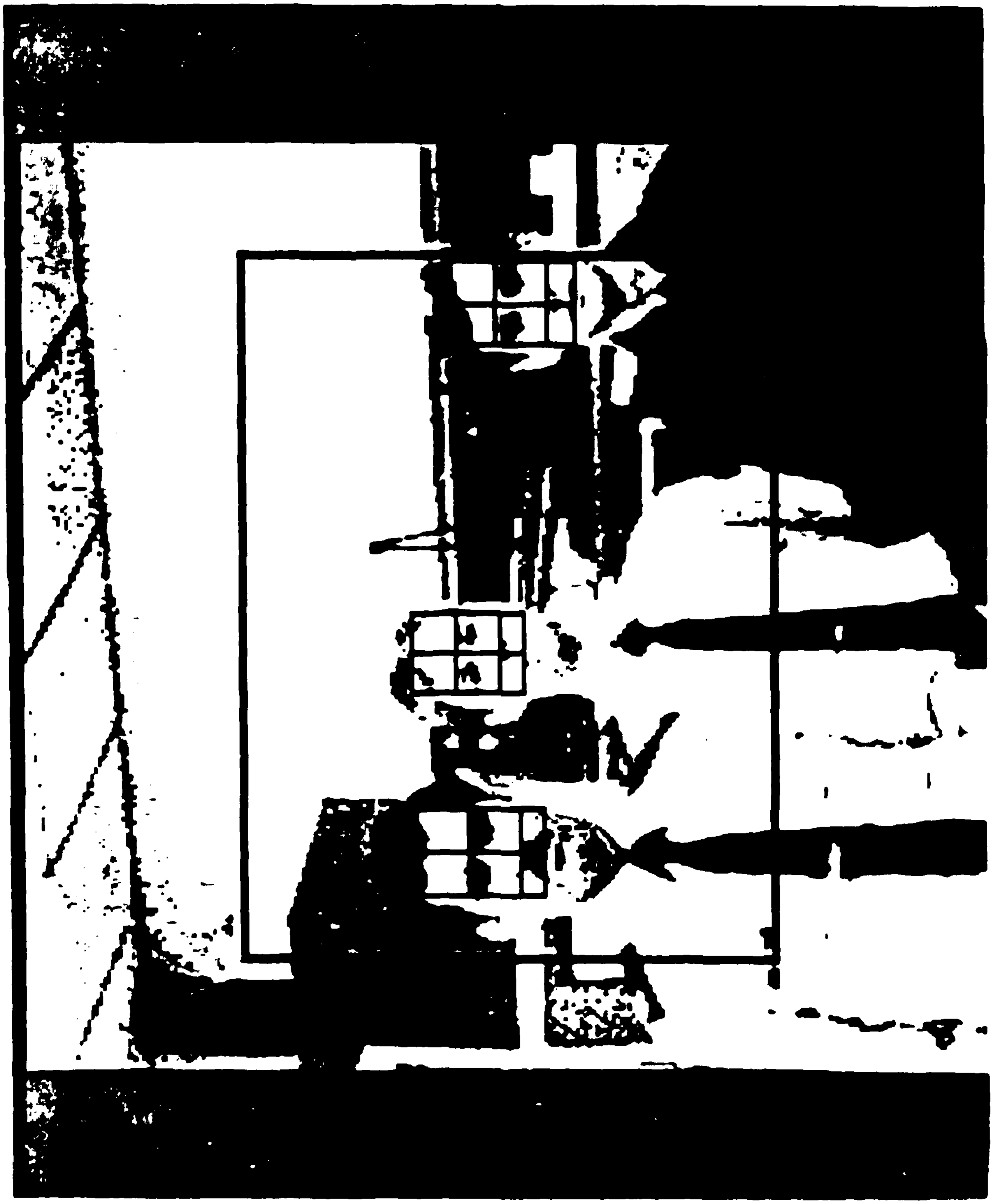


Figure E-21. Test Image #11 Results.

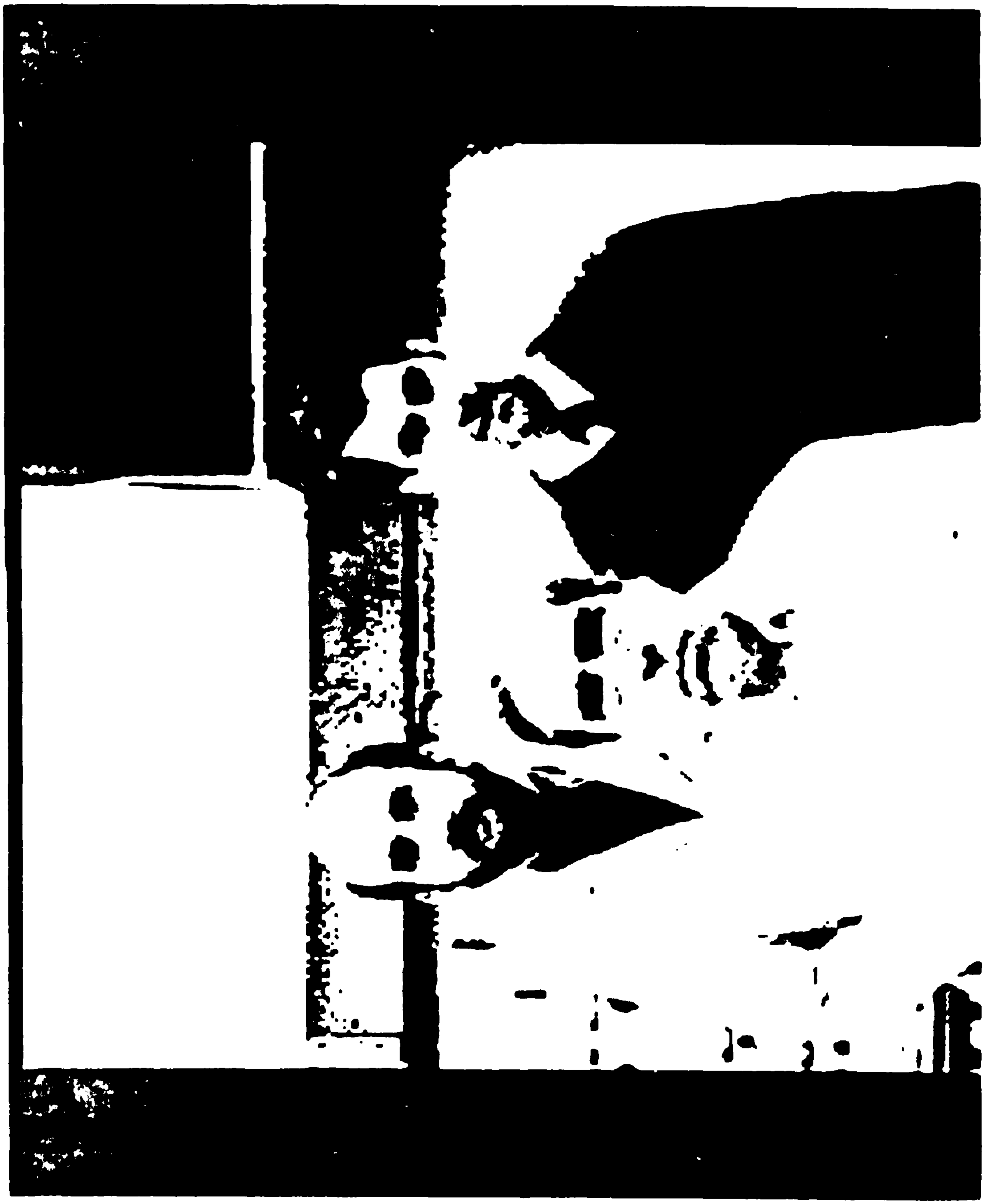


Figure E-22. Test Image #12.

Figure E-23. Test Image #12 Results.

Figure E-24. Test Image #13.

Figure E-25. Test Image #13 Results.

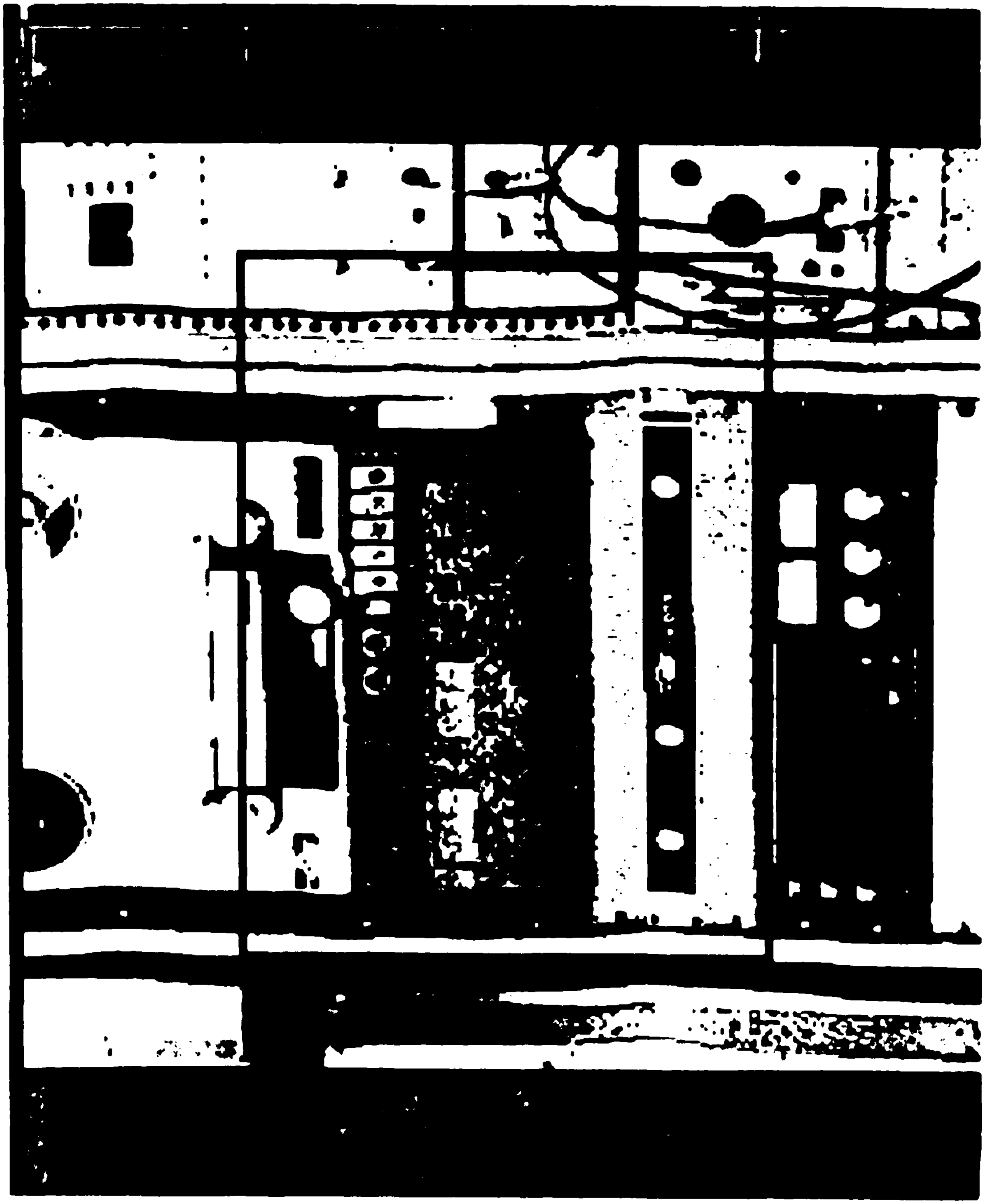


Figure E-26. Test Image #14 Results.

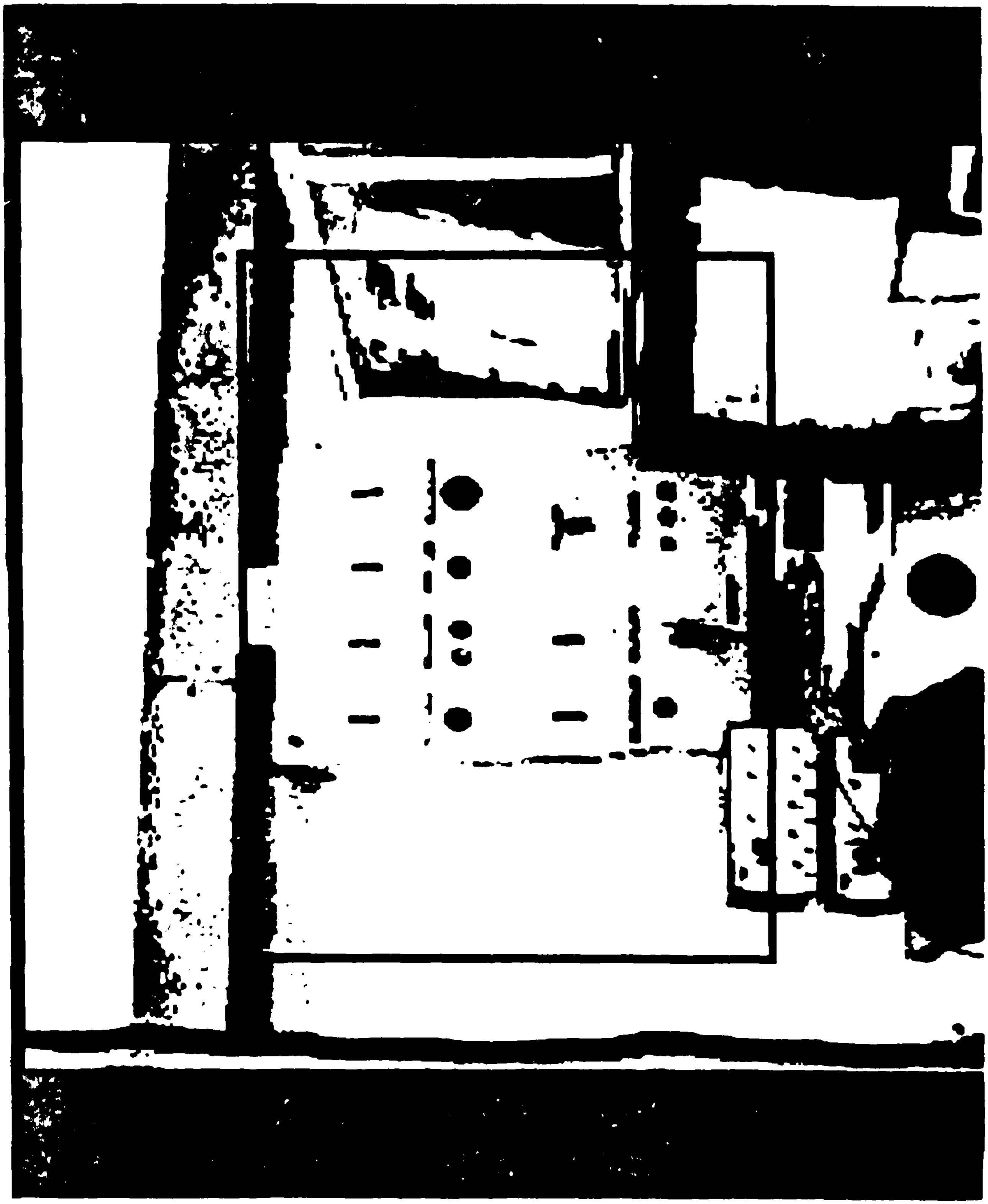


Figure E-27. Test Image #15 Results.

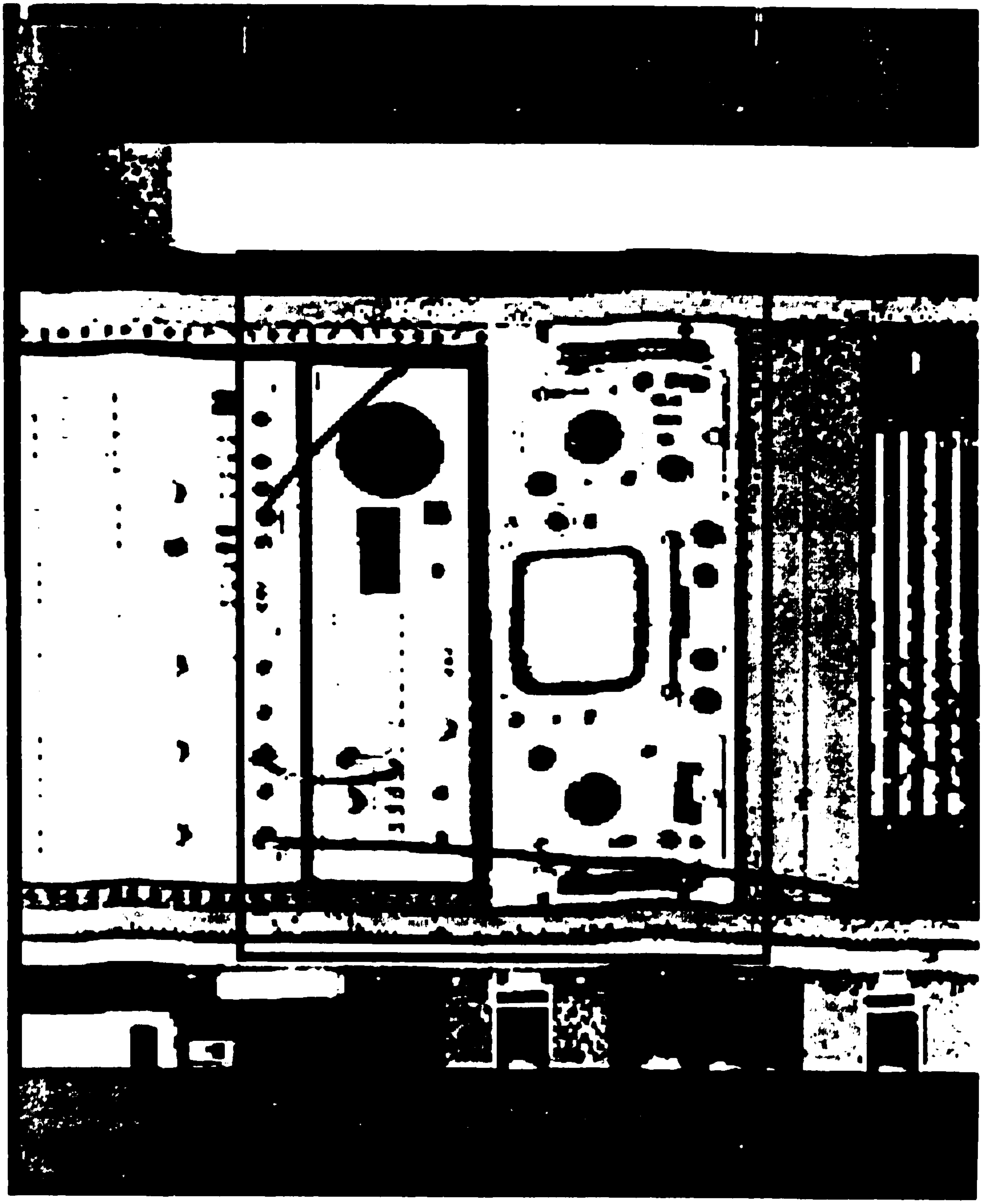


Figure E-28. Test Image #16 Results.

# Appendix F

## Software Listings

## Software Listings:

Listed below are all directories, programs, links and data files which comprise the AFRM. Reference is made to actual compiled code. Source code filenames are the same except that the extensions (".ol" ,".rb" and ".sv") all change to ".fr". Two major listings are provided, one for the Nova and one for the Eclipse.

Following the two listings, a similiar listing is provided which applies to the facial database management software. This listing shows the programs, etc., necessary to execute the program "MAIN".

At the end of this appendix, the Fortran source code for the two top level manager programs (Facefndr.fr and Getsubj.fr) is provided.

### AFRM Program Listings

Note: Indentation indicates programs which are called as part of a higher level program. Since some low level programs are called by more than one higher level program, the low level programs may appear more than once in the following list.

Computer system : Nova

```
       Directory: NSMITH

        Programs: Autoface.mc
                      Efacestat.sv
                          Octek.sv
                      Getsubj.sv
                          Svpic.sv
                          Gtgest.sv
                              Newtext.rb
                              Process2.sv
                                  O.ol
                                  D.ol
                                  R.ol
                              Showgest.sv
                                  Gestalt.rb
                          Getname.sv
                              Newtext.rb
                          Wrna2.sv
                              Rdnal.rb
                          Train.sv
                          Cleanup.sv
                          Iddisp.sv
                          Showdisp.sv
                              Nprocl.rb
```

Nprocl.rb (continued)
Newtext.rb
Rdnal.rb
Rdna4.rb

Links: Link to Dir Nsave: Octek.sv

Data Files: Emainpix
Idfile
Idfile2
Idfile3
Idfile4
Idnum
Rememl.tk
Remem2.tk
Sigmas
Windowl
Window2
Window3
Window4
Window5
Window6
Windowl.lu
Window2.lu
Window3.lu
Window4.lu
Window5.lu
Window6.lu
Windowl.sp
Window2.sp
Window3.sp
Window4.sp
Window5.sp
Window6.sp
Wfactor

---

Directory: OCTEK

Programs: None.

Other: IACF4.LB (Octek library of functions)

---

Directory: NSAVE

Programs: Octek.sv

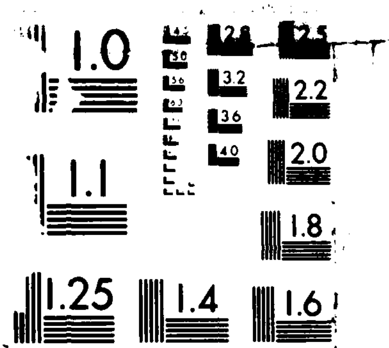
End Nova System

---

```
Computer system : Eclipse

      Directory: ESMITH

       Programs: Findface.mc
                     Cleanup2.sv
                     Facefndr.sv
                         Cleanup3.sv
                         Rtransa.rb
                         Newpix.rb
                             Out4x3.rb
                             Out3x3.rb
                         Cortran16.sv
                             Rtransa.rb
                             Rtransb.rb
                         Remid.sv
                             Talkfile.rb
                             Retrieve.rb
                             Add.rb
                                 Sort.rb
                             Probab.rb
                             Rdname.rb
                             Rdname2.rb
                             Copyfile.rb
                                 Appendit.rb
                             Rdname3.rb
                             Countit.rb
                             Talker.rb
                         Rtram.rb
                         Eyef.ol
                         Rtransb.rb
                         Fthor.ol

     Data Files: None.

                 ----------------------------------------

      Directory: NSMITH

        Program: None.

          Links: Link to Esmith:Cleanup2.sv
                 Link to Esmith:Facefndr.sv
                 Link to Esmith:Facefndr.ol
                 Link to Esmith:Cleanup3.sv
                 Link to Esmith:Cortran16.sv
                 Link to Esmith:Remid.sv

     Data Files: Same as for Nova (see Nova System/NSMITH
                 Data Files)

                 ----------------------------------------
```

```
  Directory: Facepics

   Programs: None.

      Links: None.

 Data Files: Emainpix
             Emainpix.bu
             -.pi (all 64 by 64 face image files)

End Eclipse System.
------------------------------------------------------------
```

AD-A178 852 DEVELOPMENT OF AN AUTONOMOUS FACE RECOGNITION MACHINE 3/3
(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH
SCHOOL OF ENGINEERING E J SMITH 08 DEC 86
UNCLASSIFIED AFIT/GE/ENG/86D-36 F/G 14/5 NL




MIC

N

## Facial Database Management Program Listing

The following is a program listing for the software used to maintain the face recognition database. Except for changes to incorporate speech synthesis and a display mode (when performing recognition), the software is the same as that documented in Russel's thesis. Reference Appendix B in Russel's thesis (page B-39).

Computer System: Eclipse

Directory: Esmith

```
Programs: Main.mc
              Main.ol
              Main.sv
                  S.ol
                      Createit.rb
                      Rdname.rb
                      Addit.rb
                  Rem.ol
                      Talkfile.rb
                      Retrieve.rb
                      Probab.rb
                      Rdname.rb
                      Add.rb
                          Sort.rb              Same files
                      Rdname2.rb               as for
                      Copyfile.rb              Remid.sv
                          Appendit.rb         (see p.F-4)
                      Rdname3.rb
                      Countit.rb
                      Talker.rb
                  Mark.ol
                  Pac.ol
                  Loa.ol
                      Copyfile.rb
                  Listr.ol
                  Selec.ol
                  Changev.ol
                  Testa.ol
                      Addit.rb
                  Exa.ol
                      Dataret.rb
                  Newmemory.rb
                      Createit.rb
```

Links: None.

Data Files: None.

---

Directory: Nsmith

Programs: None.

Links: Link to Esmith: Main.sv
Link to Esmith: Main.ol

Data Files: Emainpix
Idfile
Idfile2
Idfile3
Idfile4
Idnum
Rememl.tk
Remem2.tk
Sigmas
Windowl
Window2
Window3
Window4
Window5
Window6
Windowl.lu
Window2.lu
Window3.lu
Window4.lu
Window5.lu
Window6.lu
Windowl.sp
Window2.sp
Window3.sp
Window4.sp
Window5.sp
Window6.sp
Wfactor

Note: Data files are the same files listed for the AFRM under Nova/Nsmith.

End Eclipse System.

---

Computer System: Nova

Note: The following programs are used if the user wishes to display (on the TV monitor) the results of a recognition test which has been performed using "MAIN" on the Eclipse. Main will work just fine without them, if the user chooses not to use the display mode.

Directory: Nsmith

Programs: Display.mc
Cleanup.sv
Clearit.sv

```
Demo1.mc
    Showdisp.sv
    Dtitle.sv
    DProc1.sv
    DFeat.sv
    Proc1b.sv
    DProc2.sv
    Showgest.sv
```

Links: None.

Data Files: None.

End Nova System

---

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                            C
C                                                            C
C       NAME:  FACEFNDR.FR      10/12/86                     C
C       AUTHOR:  E.SMITH                                     C
C       PURPOSE: GIVEN A 192X128 PICTURE OF A HUMAN FACE AS  C
C       ACQUIRED BY USING THE "GETSUBJ" PROGRAM ON THE       C
C       NOVA, THIS PROGRAM WILL DETERMINE THE LOCATION OF    C
C       ALL THE FACIAL FEATURES REQUIRED FOR FURTHER PROC-   C
C       ESSING OF THE FACE RECOGNITION MACHINE.              C
C                                                            C
C       TO COMPILE AND LINK, USE MACRO: FACEFNDM.MC          C
C       HISTORY: COMES FROM CORTRAN16.FR                     C
C                                                            C
C                                                            C
C                                                            C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

        EXTERNAL EYEF
        EXTERNAL FTHOR

        INTEGER SUMFILE(27),IPIX(6144)
        REAL RINP(64),ARRAY(130),PARAM(16),CARRAY(64)
        INTEGER ISTAT(20),DUMMY(64),GESTDATA(300),PRNTARRAY(6,10),IFOUND(15)
        INTEGER STRIPDATA(400)
        INTEGER WHATFILE
        REAL MINMEAN,MAXMEAN
        IPROCPIC=0

C       SET UP FOR OVERLAYS
        CALL OVOPN(14,"FACEFNDR.OL",IER)

C       NOTIFY NOVA THAT THIS PROGRAM IS UP AND RUNNING.
        CALL CFILW("GREENLIGHT",2,IER)

C       INIT FACE FOUND ARRAY COUNTER TO "1"
5       IFDCT=1

C       ZERO OUT "PARAM" ARRAY

        DO 10 I=1,16
           PARAM(I)=0
10      CONTINUE

C       ZERO OUT "GESTDATA" ARRAY

        DO 20 I=1,300
           GESTDATA(I)=0
20      CONTINUE

C       ZERO OUT "STRIPDATA" ARRAY
```

```
          DO 30 I=1,400
             STRIPDATA(I)=0
30        CONTINUE

C         ZERO OUT "IFOUND" ARRAY

          DO 32 I=1,15
             IFOUND(I)=0
32        CONTINUE

          CALL SWAP("CLEANUP3.SV",IER)

C         Perform reinitialization, and bypass prompts.
          IF(IPROCPIC.EQ.0)GO TO 33
          IPROCPIC=0
          GO TO 50

C         CALCULATE TRANSFORM VALUES
33        CALL RTRANSA(ARRAY)

C         Calculate constants for CARRAY

          DO 36 I=1,64
             IFACTOR=I-1
             CARRAY(I)=0
                DO 34 J=9,64
                   CARRAY(I)=CARRAY(I)+3.0*ARRAY(J-IFACTOR+63)
34              CONTINUE
36        CONTINUE


          TYPE
40        CONTINUE
          TYPE
          TYPE
          TYPE "<7>  * * * FACEFNDR -- Gestalt Processor Program * * *"
          TYPE
          TYPE
          TYPE
          TYPE
          TYPE "<7>"
          IPRN=0
          TYPE
          IWHATPROP=2

50        CONTINUE
          IPIXCT=0

          TYPE
          TYPE " <07> * * * READY TO PROCESS PICTURE DATA * * *"
          TYPE
          TYPE
          TYPE
          TYPE
```

```
        TYPE
        TYPE
60      CONTINUE

C       Print the output file TEMP.VD if the Eclipse recieves PRNTIMAGE
        CALL STAT("PRNTIMAGE",ISTAT,IER)
        IF(IER.EQ.1)TYPE "<7>SIGNAL RECIEVED TO START PRINTING FILE"
        IF(IER.EQ.1)CALL NEWPIX(DUMMY)

C       Now delete the file
        IF(IER.EQ.1)CALL DFILW("PRNTIMAGE",IER)
        IF(IER.EQ.1)TYPE "<7>PRINTING IS COMPLETED"
        IF(IER.EQ.1)GOTO 50

C       Reinitialize if Face Recognition Program on Nova is done
        CALL STAT("FACEDONE",ISTAT,IER)
        IF(IER.EQ.1)CALL DFILW("FACEDONE",IER)
        IF(IER.EQ.1)GO TO 5

C       Terminate the program
        CALL STAT("TERMFACE",ISTAT,IER)
        IF(IER.EQ.1)CALL DFILW("TERMFACE",IER)
        IF(IER.EQ.1)STOP "Face finder terminated"

C       Calculate Gestalt Values (activate CORTRAN16)

        CALL STAT("CALCGEST",ISTAT,IER)
        IF(IER.EQ.1)CALL DFILW("CALCGEST",IER)
        IF(IER.NE.1)GO TO 63
        CALL SWAP("CORTRAN16.SV",IER)
        GO TO 50

C       Recognize an individual

63      CALL STAT("IDCOM",ISTAT,IER)
        IF(IER.EQ.1)CALL DFILW("IDCOM",IER)
        IF(IER.NE.1)GO TO 65
        CALL SWAP("REMID.SV",IER)
        GO TO 50

C       Check for "subject.vd" picture ready

65      CALL STAT("NSMITH:SUBJECT1.A",ISTAT,IER)
        IF(IER.NE.1)GO TO 210
        TYPE "<15><7>ACCESSING IMAGE FILE"
        IPROCPIC=1
        CALL DFILW("NSMITH:SUBJECT1.A",IER)
70      CALL OPEN(1,"SUBJECT.VD",3,IER)
        IF(IER.EQ.1)GO TO 74
        TYPE
        TYPE "* * ERROR OPENING IMAGE FILE * *"
74      IF(STRIPDATA(386).EQ.1)GO TO 80
        ISBLKO=1
        GO TO 90
```

```
80        ISBLKO=(90-STRIPDATA(395))/18

90        DO 180 K=ISBLKO,5
             ISBLK=72-(K-1)*18
             STRIPDATA(395)=ISBLK
             CALL RDBLK(1,ISBLK,IPIX,24,IER)

          IF(STRIPDATA(386).EQ.1)GO TO 100
          ISHFT20=0
          GO TO 110
100       ISHFT20=24-STRIPDATA(394)

110          DO 178 ISHFT2=ISHFT20,24,2
                ISHFTINV=24-ISHFT2
                STRIPDATA(394)=ISHFTINV
                M=1
                DO 150 J=1,192
                   IFACTOR=J+ISHFTINV*192

                   DO 120 I=1,8
                      IVAL=(I-1)*192+IFACTOR
                      RINP(I)=15-IPIX(IVAL)
120                CONTINUE

                   CALL RTRAN(ARRAY,CARRAY,RINP)

                   BMAX=0
                   IR3D=0
                   JR3D=0
                   DO 140 I=1,37
                      IF(RINP(I).LE.BMAX)GO TO 140
                      BMAX=RINP(I)
                      IR3D=I
                      JR3D=J
140                CONTINUE

                   STRIPDATA(M)=IR3D
                   M=M+1
                   STRIPDATA(M)=JR3D
                   M=M+1
150             CONTINUE

                ISTOPEYE=176-(24-ISHFTINV)-(K-1)*24
                CALL OVLOD(4,EYEF,1,IER)
                CALL EYEFIND(STRIPDATA)

C            Reset false alarm indicator
                STRIPDATA(386)=0

                IF(STRIPDATA(385).EQ.0)GO TO 176

C         VERIFY THIS ISN'T A PREVIOUSLY FOUND FACE
                IF(IFOUND(1).EQ.1)GO TO 152
                GO TO 190
```

```
152           IF(ISTOPEYE.GE.IFOUND(2).AND.ISTOPEYE.LE.IFOUND(3))GOTO 154
              GO TO 158
154           IF(STRIPDATA(387).GE.IFOUND(4).AND.STRIPDATA(387).LE.IFOUND( 5))
     1  GO TO 156
              GO TO 158
156     STRIPDATA(386)=1
        GO TO 80

158           IF(IFOUND(6).EQ.1)GO TO 160
              GO TO 190
160           IF(ISTOPEYE.GE.IFOUND(7).AND.ISTOPEYE.LE.IFOUND(8))GOTO 162
              GO TO 166
162           IF(STRIPDATA(387).GE.IFOUND(9).AND.STRIPDATA(387).LE.IFOUND( 10))
     1  GO TO 164
              GO TO 166
164     STRIPDATA(386)=1
        GO TO 80

166           IF(IFOUND(11).EQ.1)GO TO 168
              GO TO 190
168           IF(ISTOPEYE.GE.IFOUND(12).AND.ISTOPEYE.LE.IFOUND(13))GO TO 170
              GO TO 190
170           IF(STRIPDATA(387).GE.IFOUND(14).AND.STRIPDATA(387).LE.IFOUND (15))
     1  GO TO 172
              GO TO 190
172     STRIPDATA(386)=1
        GO TO 80

176           TYPE
              TYPE "<7>EYES NOT FOUND WITH TOP EYE WINDOW AT ROW ",ISTOPEYE

C             Check if operator wishes to end search

              CALL STAT("FACEDONE",ISTAT,IER)
              IF(IER.NE.1)GO TO 178
              CALL RESET
              CALL DFILW("FACEDONE",IER)
              GO TO 5

178        CONTINUE

180     CONTINUE
        TYPE "<15><15><15><7>ECLIPSE SEARCH ROUTINE COMPLETED"
        IF(STRIPDATA(385).EQ.0.AND.IFOUND(1).EQ.1)
     1    TYPE "<15>NO OTHER FACES WERE FOUND"

C       NOTIFY THE NOVA COMPUTER TO EITHER SEND A 64X64 SUBSECTION OR
C       INFORM THE OPERATOR THAT NO FACE WAS FOUND.

190     IF(STRIPDATA(385).EQ.0)GO TO 195
        TYPE
        TYPE "POSSIBLE EYES FOUND AT ROW ",ISTOPEYE
        TYPE
        TYPE "REQUESTING 64X64 IMAGE SUBSECTION FROM NOVA"
```

```
195     STRIPDATA(393)=ISTOPEYE
        CALL CLOSE(1,IER)
        CALL CFILW("ETEMP3",2,IER)
        CALL OPEN(3,"ETEMP3",2,IER)
        IF(IER.EQ.1)GO TO 200
        TYPE "* * ERROR OPENING STRIPDATA FILE * *"

200     CONTINUE
        WRITE BINARY(3) (STRIPDATA(I),I=385,400)
        CALL CLOSE(3,IER)
        IF(IER.NE.1)TYPE "CLOSE ERROR ON STRIPDATA FILE"
        CALL DFILW("CORDPTS64.B",IER)
        RENAME "NSMITH:ETEMP3","NSMITH:CORDPTS64.B"
        IF(STRIPDATA(385).NE.0)GO TO 210
        IPROCPIC=0
        IF(ISTOPEYE.LE.55)GO TO 685
        GO TO 50

210     CONTINUE
C       Check for IMAGE SUBSECTION ready
        IF(IPROCPIC.NE.1)GO TO 60
        CALL STAT("NSMITH:NOVASIG1.A",ISTAT,IER)
        IF(IER.NE.1)GOTO 60
        CALL OPEN(1,"WIND1.PI",3,IER)
        WHATFILE=1
220     CONTINUE


        TYPE "<07> * * * FILENAME RECEIVED FROM NOVA * * *"
        TYPE

230     CONTINUE
        CALL RDBLK(1,0,IPIX,16,IER)
        IFRSTIME=0
        IENHAN2=0
240     CONTINUE

        TYPE "Input File Read"

250     GESTDATA(257)=0

C       PERFORM GESTALT CALCULATIONS ON SUCCESSIVE, 8 PIXEL WIDE,
C          HORIZONTAL WINDOWS.
        IBKUP=0
        ISHFLO=24-IFRSTIME*(31-GESTDATA(272))
        ISHFHI=ISHFLO+6-IFRSTIME*(31-GESTDATA(272))
        DO 310 ISHFDWN=ISHFLO,ISHFHI,2
          M=1
          ISHIFT=ISHFDWN+6-IBKUP
          DO 290 J=1,64
           IFACTOR=J+ISHIFT*64
           DO 260 I=1,8
              IVAL=(I-1)*64+IFACTOR
              RINP(I)=15-IPIX(IVAL)
```

```
260          CONTINUE

C   Do Gestalt on column

           CALL RTRAM(ARRAY,CARRAY,RINP)
           BMAX=0
           IR3D=0
           JR3D=0
           DO 280 I=1,64
             IF(RINP(I).LE. BMAX)GO TO 280
             BMAX=RINP(I)
             IR3D=I
             JR3D=J
280        CONTINUE

           GESTDATA(M)=IR3D
           M=M+1
           GESTDATA(M)=JR3D
           M=M+1
290      CONTINUE
            ITOPEYE=ISHFDWN+7-IBKUP
            CALL OVLOD(4,FTHOR,1,IER)
            CALL FEATHOR(GESTDATA)
            IF(GESTDATA(257).EQ.0)GO TO 300
            GO TO 330

C        Since previous attempt failed, move up two rows and repeat process

300      CONTINUE
         IBKUP=IBKUP+4

         TYPE
         TYPE "<7>EYES NOT FOUND WITH TOP OF EYE WINDOW AT ROW ",ITOPEYE

310       CONTINUE

320      IF(IFRSTIME.EQ.0)TYPE "EYES NOT FOUND"
         IF(IFRSTIME.EQ.1)TYPE "EYES NOT FOUND AFTER ENHANCEMENT"
         ITOPEYE=0

C        GESTDATA FIELDS DEFINITION:

C        GESTDATA (1) - (128) = GESTALT DATA FOR THE 64 COLUMNS
C        GESTDATA (129)-(256) = GESTALT DATA FOR THE 64 ROWS
C        GESTDATA (257) = 0 IF EYES NOT FOUND
C        GESTDATA (258) = LEFT SIDE OF HEAD  (not used)
C        GESTDATA (259) = RIGHT SIDE OF HEAD (not used)
C        GESTDATA (260) = CENTER OF EYES
C        GESTDATA (261) = LEFT SIDE OF EYES
C        GESTDATA (262) = RIGHT SIDE OF EYES
C        GESTDATA (263) = IDELMX1
C        GESTDATA (264) = IDELMNN
C        GESTDATA (265) = IDELMNP
C        GESTDATA (266) = IDELMX2
```

```
C       GESTDATA (267) = IDELPK
C       GESTDATA (268) = LOSLOPE
C       GESTDATA (269) = LISLOPE
C       GESTDATA (270) = RISLOPE
C       GESTDATA (271) = ROSLOPE
C       GESTDATA (272) = ITOPEYE
C       GESTDATA (273) = MAXLL
C       GESTDATA (274) = MAXRL
C       GESTDATA (275) = TOP OF NOSE
C       GESTDATA (276) = CTR OF MOUTH
C       GESTDATA (277) = FALSE EYE OCCURENCE INDICATOR
C       GESTDATA (278) = MINCV

C       PARAM Field definition: (real numbers)

C       PARAM (1) = CONTRAST MULTIPLIER

330     CONTINUE
        GESTDATA(272)=ITOPEYE
        TYPE
        TYPE "* * GESTALT CALCULATIONS ON EYES COMPLETED * *"
        TYPE

C       ********************************************

C       IF EYE'S NOT FOUND, NOTIFY NOVA AND TERMINATE.
        IF(ITOPEYE.EQ.0)GO TO 650


C       Determine the "top of nose" location


C       SET-UP "CENTER OF FACE" SLICE FOR PROCESSING

        ILLIM=GESTDATA(273)
        IRLIM=GESTDATA(274)
        IENDW=IRLIM-ILLIM+1
        ISENDW=IENDW+1

        M=129
        DO 370 K=1,64
           IFACTOR=(K-1)*64+ILLIM-1
               DO 340 I=1,IENDW
                  IVAL=IFACTOR+I
                  RINP(I)=15-IPIX(IVAL)
340            CONTINUE
               DO 350 I=ISENDW,64
                  RINP(I)=3.0
350            CONTINUE
               CALL RTRANSB(ARRAY,RINP)
               BMAX=0
               IR3D=0
               JR3D=0
               DO 360 I=1,64
```

```
                 IF(RINP(I).LE.BMAX)GO TO 360
                 BMAX=RINP(I)
                 IR3D=K
                 JR3D=I
360           CONTINUE
              GESTDATA(M)=IR3D
              M=M+1
              GESTDATA(M)=JR3D
              M=M+1
370      CONTINUE
         TYPE
         TYPE "* * GESTALT CALCULATIONS FOR VERT WINDOW COMPLETED * *"
         TYPE

C        Find the lowest point (on the face) of the brightest area
C        immediately below the eyes.

         IF(IFRSTIME.EQ.1)GO TO 410

C        RESET THE FALSE EYE INDICATOR
         GESTDATA(277)=0

         IGTOPEYE=130+2*ITOPEYE
         TYPE "IGTOPEYE = ",IGTOPEYE
         IHINOSE=IGTOPEYE+30
         IHIVAL=GESTDATA(IGTOPEYE)
         ILOVAL=IHIVAL
         DO 380 I=IGTOPEYE,IHINOSE,2
            IF(GESTDATA(I).LT.ILOVAL)GO TO 380
            ILOVAL=GESTDATA(I)
            IHILOC=I-1
380      CONTINUE

C        IF NO MIN, SET FALSE ALARM INDICATOR AND CONTINUE SEARCH

         IHIVATST=ILOVAL-IHIVAL
         IF(IHIVATST.GT.1)GO TO 390
         TYPE
         TYPE "NO INTENSITY MINIMUM DETECTED BELOW THE EYES"
         GESTDATA(257)=0
         GO TO 650

390      IOHILOC=IHILOC
         TYPE "IHILOC = ",IHILOC

C        Now, locate the top of the nose which is usually located
C        just a few pixel rows below the point in "IHILOC"

         DO 400 I=1,12
            IVAL1=GESTDATA(IHILOC+1)
            IVAL2=GESTDATA(IHILOC+3)
            IVAL3=GESTDATA(IHILOC+5)
            IDIFF1=IVAL1-IVAL2
            IDIFF2=IVAL2-IVAL3
```

```
            AVGOF2=(IDIFF1+IDIFF2)/2.0
            IF(AVGOF2.LE.0)GO TO 470
            IHILOC=IHILOC+2
400      CONTINUE
            TYPE
            TYPE "TOP OF NOSE NOT FOUND IN ORIGINAL PICTURE"

C        SINCE NOSE NOT FOUND, SET FALSE ALARM INDICATOR AND CONTINUE SEARCH

         GESTDATA(257)=0
         GO TO 650

C        Find lowest point of the bright area below the eyes (after enhance)

410      IHILOC=IOHILOC
         TYPE "ITOPEYE = ",ITOPEYE
         DO 420 I=1,10
            IF(GESTDATA(IHILOC+3).LT.GESTDATA(IHILOC+1))GO TO 430
            IHILOC=IHILOC+2
420      CONTINUE

C        Find the top of nose (after enhance)

430      IVAL=0
         DO 460 I=1,5
            IDIFF3=GESTDATA(IHILOC+1)-GESTDATA(IHILOC+3)
            IF(IDIFF3.GT.0.AND.IDIFF3.GT.IVAL)GO TO 440
            GO TO 450
440         IVAL=IDIFF3
            INOSLOC=IHILOC
450         IHILOC=IHILOC+2
460      CONTINUE
         IHILOC=INOSLOC+2

C        Check if top of nose found after enhancing the image
         IF(IVAL.GT.1)GO TO 470
         TYPE
         TYPE "TOP OF NOSE NOT FOUND IN ENHANCEMENT"
         GESTDATA(257)=0
         GO TO 650

470      ITOPNOSE=GESTDATA(IHILOC-2)

C        Check if top of nose is in proper location wrt eyes
         ICTRTOI=GESTDATA(260)-GESTDATA(273)
         ITONOSE=ITOPNOSE-ITOPEYE
         INOSLL=1.2*ICTRTOI
         IF(ITONOSE.GE.INOSLL)GO TO 475
         TYPE
         TYPE "TOP OF NOSE TOO CLOSE TO EYES"
         TYPE
         TYPE "MIN VALUE MUST BE = OR GT  ",INOSLL
         TYPE "ACTUAL EYE TO NOSE DISTANCE = ",ITONOSE
         GESTDATA(257)=0
```

```
      GO TO 650
475   ICTRTOI=2.6*ICTRTOI
      IF(ITONOSE.LE.ICTRTOI)GO TO 480
      TYPE
      TYPE "TOP OF NOSE TOO FAR FROM EYES"
      TYPE
      TYPE "DISTANCE FROM EYE TO NOSE = ",ITONOSE
      TYPE "MAX ALLOWED               = ",ICTRTOI
      GESTDATA(257)=0
      GO TO 650

480   IFRSTIME=IFRSTIME+1
      TYPE "ITOPNOSE = ",ITOPNOSE
      GESTDATA(275)=ITOPNOSE
      INOSLOC=ITOPNOSE
      IF(IENHAN2.EQ.2)GO TO 590
      IF(IENHAN2.EQ.1)IFRSTIME=1
      IENHAN2=IENHAN2+1

C     TYPE
C     TYPE "<07>* * * Now Creating Picture with Expanded Contrast * * *"
C     TYPE

      CALL RDBLK(1,0,IPIX,16,IER)

      MIN=15
C     FIND VALUE OF DARKEST SPOT IN PICTURE
      DO 490 J=1,4096
       IF(IPIX(J).LT.MIN)MIN=IPIX(J)
490   CONTINUE


C     EXPAND CONTRAST USING MODIFIED RUSSEL EXPANSION TECHNIQUE

      AMEAN=14.5
      SLOP=.05
      MAXITERATE=12
      ITERATE=0
      PRESENT=1
      DELTA=8
      MINMEAN=AMEAN-SLOP
      MAXMEAN=AMEAN+SLOP
      ITOP=ITOPEYE-(GESTDATA(260)-GESTDATA(273))/2
      IBOT=ITOPNOSE
      ILEFT=GESTDATA(273)
      IRIGHT=GESTDATA(274)
      TYPE "ITOP IBOT ILEFT IRIGHT = ",ITOP,IBOT,ILEFT,IRIGHT

500   CONTINUE


      SUM=0
      SUMSQUARE=0
      IF(ITERATE.GT.MAXITERATE)GOTO 550
```

```
      ITERATE1=ITERATE+1
      TYPE "Iteration ",ITERATE1
      INUMPTS=0
      DO 510 I=ITOP,IBOT
        DO 510 J=ILEFT,IRIGHT
          IVAL=(I-1)*64+J
          VALUE=(IPIX(IVAL)-MIN+1)*PRESENT
          IF(VALUE.GT.16)VALUE=16
          IF(VALUE.LT.1)VALUE=1
          SUM=SUM+VALUE
          SUMSQUARE=SUMSQUARE+VALUE**2
          INUMPTS=INUMPTS+1
510   CONTINUE


      AVERAGE=SUM/INUMPTS
      ITERATE=ITERATE+1
      TYPE "Average pixel value (1 TO 16 SCALE)=",AVERAGE
      TYPE
      IF(AVERAGE.GT.MAXMEAN)GOTO 520
      GOTO 530
520   CONTINUE
      PRESENT=PRESENT-DELTA
      DELTA=DELTA/2
      GOTO 500

530   CONTINUE
      IF(AVERAGE.LT.MINMEAN)GOTO 540
      GOTO 550
540   CONTINUE
      PRESENT=PRESENT+DELTA
      DELTA=DELTA/2
      GOTO 500
550   CONTINUE

      TYPE "PRESENT = ",PRESENT
      PARAM(1)=PRESENT

      DO 560 I=1,4096
         VALUE=(IPIX(I)-MIN+1)*PRESENT
         IPIX(I)=VALUE
560   CONTINUE

C     FIND VALUE OF DARKEST PIXEL IN ENHANCED PICTURE

      MIN=16
      DO 570 I=1,4096
         IF(IPIX(I).LT.MIN)MIN=IPIX(I)
570   CONTINUE

C     ADJUST PICTURE DOWN TO PUT DARKEST PIXELS AT ZERO INTENSITY

      DO 580 I=1,4096
         VALUE=IPIX(I)-MIN
```

```
          IF(VALUE.GT.15)VALUE=15
          IPIX(I)=VALUE
580     CONTINUE

        GO TO 240

C       FIND CENTER OF MOUTH

590     IHILOC=IHILOC+10
        IVAL=0
        IMOUSWT=0
        IMOULOC=0
        DO 630 I=1,7
          IDIFF4=GESTDATA(IHILOC+3)-GESTDATA(IHILOC+1)
          IF(IDIFF4.LE.1)GO TO 610
          IMOUSWT=1
          IF(IDIFF4.GE.IVAL)GO TO 600
          GO TO 620
600       IVAL=IDIFF4
          IMOULOC=IHILOC
          GO TO 620
610       IF(IMOUSWT.EQ.1)GO TO 640
620      IHILOC=IHILOC+2
630     CONTINUE

C       Check if mouth located
640     IF(IMOULOC.GT.0)GO TO 644
        TYPE
        TYPE "MOUTH NOT FOUND"
        GESTDATA(257)=0
        GO TO 650

C       Check if mouth located in proper location
644     IMOUTH=GESTDATA(IMOULOC)
        ICTRTOI=GESTDATA(260)-GESTDATA(273)
        IMOUTHUL=ITOPNOSE+ICTRTOI-ICTRTOI/3
        IMOUTHLL=ITOPNOSE+ICTRTOI*2
        IF(IMOUTH.GE.IMOUTHUL.AND.IMOUTH.LE.IMOUTHLL)GO TO 646
        TYPE
        TYPE "MOUTH LOCATED, BUT NOT IN PROPER POSITION"
        TYPE
        TYPE "MOUTH LOCATION LOWER LIM/UPPER LIM = ",IMOUTHLL,IMOUTHUL
        TYPE "ACTUAL MOUTH LOCATION AT  ",IMOUTH
        GESTDATA(257)=0
        GO TO 650
646     TYPE "CTR OF MOUTH AT ",IMOUTH
        GESTDATA(276)=IMOUTH

C       ******************************************

650     CALL CFILW("ETEMP3",2,IER)
        CALL OPEN(3,"ETEMP3",2,IER)
        IF(IER.EQ.1)GOTO 660
        TYPE "* * ERROR IN OPENING COORDINATE FILE ETEMP3 * *"
```

```
660     CONTINUE

C       Write coordinate points out to disk, so NOVA can use them.
        WRITE BINARY(3)(GESTDATA(I),I=1,300),(PARAM(I),I=1,16)

C       Close coordinate file

        CALL CLOSE(3,IER)
        IF(IER.NE.1)TYPE "CLOSE ERROR ON COORDINATE FILE ETEMP3, IER=",IER

C       Delete any present coordinate files

        IF(WHATFILE.EQ.1)CALL DFILW("COORDPTS1.B",IER)

C       Rename our temporary file to a coordinate file
C       (This is necessary because the NOVA is constantly scanning for
C       these filenames, and must not see them until they have been properly
C       filled with data.  Otherwise, it will try to open them when they
C       are first created.)

        IF(WHATFILE.EQ.1)
     1  RENAME "NSMITH:ETEMP3","NSMITH:COORDPTS1.B"


670     CALL CLOSE(1,IER)
        IF(IER.NE.1)TYPE "CLOSE ERROR ON DATA FILE, IER=",IER

C       Now that we're done processing the file, delete the flag file from the
C         Nova.

        IF(WHATFILE.EQ.1)CALL DFILW("NSMITH:NOVASIG1.A",IER)
C       IF(IER.EQ.1)GOTO 680
C       TYPE "NOVASIG NOT DELETED, IER=",IER

680     CONTINUE

        STRIPDATA(386)=1
        IF(GESTDATA(257).EQ.0)GO TO 685
        IPIXCT=IPIXCT+1
        IF(IPIXCT.GE.4)ISTOPEYE=0
        IF(IPIXCT.GE.4.AND.IPRN.EQ.1)GO TO 687
        IF(IPIXCT.GE.4)GO TO 685
        IFOUND(IFDCT)=1
        IFOUND(IFDCT+1)=ISTOPEYE-4*ICTRTOI
        IFOUND(IFDCT+2)=ISTOPEYE+2*ICTRTOI
        IFOUND(IFDCT+3)=STRIPDATA(388)-2*ICTRTOI
        IFOUND(IFDCT+4)=STRIPDATA(389)+2*ICTRTOI
        IFDCT=IFDCT+5
        IF(IPRN.EQ.1)GO TO 687

685     IF(ISTOPEYE.GT.55)GO TO 70

        TYPE
```

```
      TYPE "TEST IMAGE SEARCH FOR FACE COMPLETED"
      TYPE
      GO TO 9999

C     The following section of code (for printing) was stranded by setting
C     IPRN=0 at the beginning of this program. It has been left in since,
C     in the future, a user may wish to use it. It may be used by coding
C     an ACCEPT statement at the beginning of this program, which will
C     allow the operator to determine the value of the variable "IPRN".

C**********************************************************************
C     Print the Data on the Printer (if requested)                    *
C**********************************************************************

C     Get date and time
687   CALL FGDAY(IMONTH,IDAY,IYEAR)
      CALL FGTIME(IHOUR,IMINUTE,ISEC)

      IF(IPRN.EQ.1)WRITE(12,690)IPIXCT
      IF(IPRN.EQ.1)WRITE(12,7091)
      IF(IPRN.EQ.1)WRITE(12,700)IMONTH/10,MOD(IMONTH,10),IDAY/10,M OD(IDAY,10),
     1 IYEAR/10,MOD(IYEAR,10),IHOUR/10,MOD(IHOUR,10),IMINUTE/10,MOD (IMINUTE,10)
690   FORMAT("1      * * * Feature Locations for TEST SUBJECT #",I2," * * *")
700   FORMAT("  Date: ",I1,I1,"/",I1,I1,"/",I1,I1,5X,"Time: ",I1,I1,":",I1,I1)

710   CONTINUE

C     Display Critical parameters for horiz features

      IF(IPRN.EQ.1)WRITE(12,7095)GESTDATA(261)
      IF(IPRN.EQ.1)WRITE(12,7107)GESTDATA(273)
      IF(IPRN.EQ.1)WRITE(12,7094)GESTDATA(260)
      IF(IPRN.EQ.1)WRITE(12,7108)GESTDATA(274)
      IF(IPRN.EQ.1)WRITE(12,7096)GESTDATA(262)
      IF(IPRN.EQ.1)WRITE(12,7109)GESTDATA(275)
      IF(IPRN.EQ.1)WRITE(12,7110)GESTDATA(276)
C     IF(IPRN.EQ.1)WRITE(12,7102)GESTDATA(268)
C     IF(IPRN.EQ.1)WRITE(12,7103)GESTDATA(269)
C     IF(IPRN.EQ.1)WRITE(12,7104)GESTDATA(270)
C     IF(IPRN.EQ.1)WRITE(12,7105)GESTDATA(271)
      IF(IPRN.EQ.1)WRITE(12,7106)GESTDATA(272)


C     Close printer file to let someone else use it.
      IF(IPRN.EQ.1)CALL CLOSE(12,IER)

7077  FORMAT(1X,"* * * COORDINATE POINTS FOR ",S14," * * *")
7090  FORMAT(1X," Coordinate Points for SECTION #",I2," = (",I2,",",I2,")")
7091  FORMAT(1X,"  ")
7094  FORMAT(1X,"    CENTER OF EYES  = ",I2)
7095  FORMAT(1X,"     LEFT SIDE EYES  = ",I2)
7096  FORMAT(1X,"   RIGHT SIDE EYES  = ",I2)
7109  FORMAT(1X,"       TOP OF NOSE   = ",I2)
7110  FORMAT(1X,"      CTR OF MOUTH  = ",I2)
```

```
C102    FORMAT(1X,"           LOSLOPE  = ",I2)
C103    FORMAT(1X,"           LISLOPE  = ",I2)
C104    FORMAT(1X,"           RISLOPE  = ",I2)
C105    FORMAT(1X,"           ROSLOPE  = ",I2)
7106    FORMAT(1X,"           ITOPEYE  = ",I2)
7107    FORMAT(1X,"             MAXLL  = ",I2)
7108    FORMAT(1X,"             MAXRL  = ",I2)

        IF(IPIXCT.GE.4)GO TO 685

9999    CONTINUE

        TYPE "<7>"
        TYPE
        TYPE
C       Go back for another...
        IF(ISTOPEYE.GT.55)GO TO 70
        GO TO 50

        END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                              C
C        GETSUBJ.FR 10/13/86 BY CAPT ED SMITH                  C
C                                                              C
C       Top level program used in performing autonomous face   C
C       location and recognition in a visual image.            C
C                                                              C
C         Uses the OCTEK 2000 Image Analyzer Card in the NOVA  C
C                                                              C
C        Adapted from "Picture1.fr"                            C
C        by James R. Holten III (6 May 85)                     C
C                                                              C
C        Use "GETSUBM.MC" to compile and load                  C
C                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

        COMMON ICTPIX,ICHOICE
        INTEGER ICT(120),STRIPDATA(400),GESTDATA(300)
        INTEGER IPIX(6150)
        INTEGER OUTFILE(40)
        INTEGER SUMFILE(20)
        REAL PARAM(16)
        IXL=320
        IXC=160
        IYL=240
        IYC=120
        NEWREC=0
        IC=1
        LASTW=0
        IPIXCT=1


        CALL DFILW("CORDPTS64.B",IER)
10      CONTINUE
        TYPE "<33>E <15><15><15><7>"
        TYPE "  Autonomous Face Recognition Machine."
        TYPE
        TYPE "  Keypad Menu"
        TYPE "  ----------------------------------------"
        TYPE "       1   2   3   4        5   6   7   8"
        TYPE "  ----------------------------------------"
        TYPE "   1--camera #1 on"
        TYPE "   4--camera #1 off"
        TYPE "   6--process picture"
        TYPE "   8--- terminate and exit to system"

        IF (NEWREC.EQ.1 ) GOTO 40
        NEWREC=1
        CALL SINTRO (ICT,63K,IER)
        IF(IER.EQ.1) GO TO 30
        TYPE 'INTRO ECODE:',IER
        STOP "UNABLE TO INITIALIZE"

30      CONTINUE
```

```
        CALL OPEN (3,"IACHON.XB",2,IER)
        IF(IER.NE.1) TYPE "WARNING: Unable to access IACHON.XB"
        IF(IER.EQ.1) CALL LXB (ICT,3)

        CALL MPRUN (ICT,0,1)
C
C         Initialize the variables and devices.
        INF=0
        TYPE
        TYPE " Interactive video input control"
        TYPE "<33>j"
        CALL XHAIR(ICT)
        CALL GREYSCALE(ICT,1)
        CALL GREYSCALE(ICT,2)


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C              Loop until operator chooses an option
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


40      CONTINUE
        TYPE "<33>k"
C         Sound the annunciator
        CALL SOB(ICT,0,0)
        CALL SOB(ICT,1,0)

50      CONTINUE

C         Input the bottom key row.

        CALL SOB(ICT,0,1)
        IWRD=IRDEXT(ICT)
        IF (IWRD .EQ. LASTW) GOTO 50
        LASTW=IWRD
        IF (IWRD .EQ. 0) GOTO 50

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C         Check for camera control (left half bottom row keys).
        IF (IWRD .LT. 16) GOTO 80
C         Turn on the camera?
        IF (IWRD .AND. 128) GO TO 60
C         Turn off camera?
        IF (IWRD .AND. 16) GOTO 70

60      CONTINUE
C         Turn on currently selected camera.
        IC=1
        CALL INTLACE(ICT,1)
        CALL SYNC(ICT,IC,ILOCK)
        IF (ILOCK .EQ. 0) GOTO 200
```

```
      CALL VON(ICT,0)
      CALL BOXCUR(ICT,192,128)
      CALL HCTAB(ICT,IXC-96,IYC-64)
      TYPE
      TYPE "Ensure subject faces are located within the box cursor"
      TYPE "then press button #4 on the keypad to freeze the image."
      GOTO 40

70    CONTINUE
C       Turn off camera.
      CALL VOFF(ICT)
      CALL SYNC(ICT,0,ILOCK)
      CALL INTLACE(ICT,0)
      GOTO 10

80    CONTINUE
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       Check lower row of buttons, right half.
C
C       Check for exit selection
      IF(IWRD.AND.1)GO TO 190
C       Check for Process Picture command.
      IF(IWRD.AND.4)GO TO 144

      GOTO 40

144   INUMPASS=1
      TYPE "<33>E <15><15><15><7>"
      TYPE "CREATING FILE OF IMAGE CONTAINED WITHIN THE BOX CURSOR"
      CALL BOXCUR(ICT,192,128)
      CALL HCTAB(ICT,IXC-96,IYC-64)
148   CONTINUE
      GO TO(150,153,155,157,160),INUMPASS
150   CALL DFILW("SUBJECT.VD",IER)
      CALL CFILW("SUBJECT.VD",3,96,IER)
      CALL OPEN(1,"SUBJECT.VD",3,IER)
      IF (IER .EQ. 1) GOTO 151
      TYPE "---- error opening file--- IER=",IER
      GOTO 159

C     Extract the 192 by 128 pixel search area from the image
C     on the TV screen.

151   CONTINUE
      CALL RVBLK(ICT,IPIX,IXC-96,192,IYC-64,32)
C     TYPE "IPIX(1) = ",IPIX(1)
      IF(INF.NE.3)CALL WRBLK(1,0,IPIX(2),24,IER)
      GO TO 159

153   CONTINUE
      CALL RVBLK(ICT,IPIX,IXC-96,192,IYC-32,32)
      IF(INF.NE.3)CALL WRBLK(1,24,IPIX(2),24,IER)
```

```
      GO TO 159
155   CONTINUE
      CALL RVBLK(ICT,IPIX,IXC-96,192,IYC,32)
      IF(IMF.NE.3)CALL WRBLK(1,48,IPIX(2),24,IER)
      GO TO 159
157   CONTINUE
      CALL RVBLK(ICT,IPIX,IXC-96,192,IYC+32,32)
      IF(IMF.NE.3)CALL WRBLK(1,72,IPIX(2),24,IER)
159   CONTINUE
      INUMPASS=INUMPASS+1
      GO TO 148
160   CALL CLOSE(1,IER)
C     Tell Eclipse that image search area is ready and to begin
C     analysis to find faces.
      CALL CFILW("SUBJECT1.A",IER)
      TYPE
      TYPE "<7>IMAGE FILE READY, SEARCH ROUTINE NOW STARTING ON ECLIPSE"
C     Await Eclipse request for a 64 by  64 image sub-section or
C     indication that the search for faces is complete.
164   CALL STAT("NSMITH:CORDPTS64.B",ISTAT,IER)
      IF(IER.EQ.1)GO TO 166
      CALL SOB(ICT,0,1)
      IWRD=IRDEXT(ICT)
      IF(IWRD.AND.2)GO TO 169
      GO TO 164
166   CALL OPEN(1,"CORDPTS64.B",3,IER)
      IF(IER.EQ.1)GO TO 168
      TYPE "* * ERROR IN OPENING STRIPDATA FILE * *"
168   CONTINUE
      READ BINARY(1) (STRIPDATA(I),I=385,400)
      TYPE
      TYPE "STRIPDATA READ"
      CALL CLOSE(1,IER)
      CALL DFILW("CORDPTS64.B",IER)
      TYPE "<33>E <15><15><15><7>"
      TYPE "<7>MESSAGE RECIEVED FROM ECLIPSE:"
C     If search is done (STRIPDATA(385)=0), proceed to next phase
C     (train or identify). If search is not done, then process the
C     Eclipse request for an image sub-section.
      IF(STRIPDATA(385).EQ.0)GO TO 170
      GO TO 174
169   CALL CFILW("FACEDONE",2,IER)
170   IPIXCT=IPIXCT-1
      CALL PXFILL(ICT,0,62,196,54,2)
```

```
      CALL PXFILL(ICT,0,62,196,185,2)
      CALL PXFILL(ICT,0,62,2,56,129)
      CALL PXFILL(ICT,0,256,2,56,129)
      CALL HCTAB(ICT,255,255)
      TYPE "<33>E <15><15><15><7>"
      TYPE
      TYPE "FACES FOUND WITHIN THE BOX CURSOR AREA = ",IPIXCT
      TYPE
171   ACCEPT "WOULD YOU LIKE TO PRINT THIS IMAGE ? (1=YES,0=NO)",IPRN
      IF(IPRN.NE.1.AND.IPRN.NE.0)GO TO 171
      IF(IPRN.EQ.0)GO TO 1711
      CALL SWAP("SVPIC.SV",IER)
1711  IF(IPIXCT.LE.0)GO TO 10
172   TYPE
      TYPE
      TYPE "<7>WOULD YOU LIKE TO : "
      TYPE
      TYPE "   1 - TRAIN WITH THESE FACES?"
      TYPE "   2 - IDENTIFY THESE FACES?  "
      TYPE "   3 - QUIT?"
      TYPE
      ACCEPT "        CHOICE: ",ICHOICE
      IF(ICHOICE.LT.1.OR.ICHOICE.GT.3)GO TO 172
      IF(ICHOICE.EQ.3)GO TO 10
      ICTPIX=0
1734  ICTPIX=ICTPIX+1
         IF(ICTPIX.EQ.1)SUMFILE(1)=1
         IF(ICTPIX.EQ.2)SUMFILE(1)=2
         IF(ICTPIX.EQ.3)SUMFILE(1)=3
         IF(ICTPIX.EQ.4)SUMFILE(1)=4

C     Signal Eclipse to set up (activate CORTRAN16) to calculate
C     facial window gestalts.

         CALL CFILW("CALCGEST",2,IER)
         CALL STAT("NOVASIG1",ISTAT,IER)
         IF(IER.NE.1)GO TO 1733
         CALL DFILW("NOVASIG1",IER)
         IF(IER.NE.1)TYPE "DELETE FILE ERROR ON NOVASIG1"
1733     CONTINUE
         CALL CFILW("NOVASIG1",2,IER)
         IF(IER.NE.1)TYPE "CREATE FILE ERROR ON NOVASIG1"
         CALL OPEN(1,"NOVASIG1",2,IER)
         IF(IER.NE.1)TYPE "OPEN FILE ERROR ON NOVASIG1"
         WRITE BINARY(1) SUMFILE(1)
         CALL CLOSE(1,IER)

C      Activate program which displays the facial window gestalts.

         CALL SWAP("GTGEST.SV",IER)
         TYPE
         IF(ICHOICE.EQ.2)GO TO 1736
         CALL SWAP("GETNAME.SV",IER)
         CALL SWAP("WRNA2.SV",IER)
```

```
          CALL SWAP("TRAIN.SV",IER)
          GO TO 1732
1736      CALL DFILW("IDCOM",IER)
          CALL SWAP("CLEANUP.SV",IER)
          CALL SWAP("IDDISP.SV",IER)
          CALL CFILW("IDCOM",2,IER)
          CALL SWAP("SHOWDISP.SV",IER)
1732      TYPE
          ACCEPT "WOULD YOU LIKE TO PRINT THIS IMAGE ? (1=YES,0=NO)",IPRN
          IF(IPRN.NE.1.AND.IPRN.NE.0)GO TO 1732
          IF(IPRN.EQ.0)GO TO 1731
          CALL SWAP("SVPIC.SV",IER)
          TYPE
1731    IF(ICTPIX.LT.IPIXCT)GO TO 1734
        GO TO 10

C       Process an Eclipse request for an image sub-section.

174     TYPE
        TYPE "PROCESSING ECLIPSE REQUEST FOR 64X64 IMAGE SUBSECTION"
        IYCDEL=STRIPDATA(393)-150
        IXCDEL=STRIPDATA(387)-128
        CALL BOXCUR(ICT,64,64)
        CALL HCTAB(ICT,IXC+IXCDEL,IYC+IYCDEL)
        CALL RVBLK(ICT,IPIX,IXC+IXCDEL,64,IYC+IYCDEL,64)
        CALL DFILW("WIND1.PI",IER)
        CALL CFILW("WIND1.PI",3,18,IER)
        CALL OPEN(1,"WIND1.PI",3,IER)
        IF(IER.EQ.1)GO TO 176
        TYPE "* * ERROR OPENING WIND1.PI FILE * *"
176     CALL WRBLK(1,0,IPIX(2),16,IER)
        CALL CLOSE(1,IER)
        CALL CFILW("NOVASIG1.A",IER)
        TYPE
        TYPE "64X64 IMAGE SUBSECTION SENT TO ECLIPSE"

C       Await results of the Eclipse analysis of the image sub-
C       section. Display these results to the operator.

178     CALL STAT("NSMITH:COORDPTS1.B",ISTAT,IER)
        IF(IER.NE.1)GO TO 178
        TYPE "<15><15>"
        TYPE "<7>IMAGE SUBSECTION PROCESSING COMPLETED"
        CALL OPEN(1,"COORDPTS1.B",3,IER)
        IF(IER.EQ.1)GO TO 180
        TYPE "<15>* * ERROR OPENING SUBSECTION FILE * *"
180     READ BINARY(1) (GESTDATA(I),I=1,300),(PARAM(I),I=1,16)
        CALL CLOSE(1,IER)
        IF(IER.NE.1)TYPE "CLOSE ERROR ON SUBSECTION FILE"
        CALL DFILW("COORDPTS1.B",IER)
        IF(GESTDATA(257).NE.0)GO TO 184
        TYPE
        TYPE "FACE NOT FOUND IN THIS SUBSECTION"
        TYPE
```

```
        TYPE "CONTINUING SEARCH"
        TYPE
        TYPE "HIT OCTEK KEYPAD #7 TO DISCONTINUE THE SEARCH"
        CALL BOXCUR(ICT,192,128)
        CALL HCTAB(ICT,IXC-96,IYC-64)
        GO TO 164
184     TYPE
        TYPE "FACE FOUND IN THE BOX CURSOR SUBSECTION"

        ITOPNDEL=0
        ITOPNOSE=GESTDATA(275)

C       IF FACE LOCATION WAS TOO HIGH IN FIRST IMAGE, ADJUST CURSOR DOWN
C         AND RESNAP IMAGE.
        IF(ITOPNOSE.GE.43)GO TO 192
         TYPE "<15>CENTERING IMAGE"
         ITOPNDEL=43-ITOPNOSE
         IXCDEL=STRIPDATA(387)-128
         IYCDEL=STRIPDATA(393)-150-ITOPNDEL
         CALL HCTAB(ICT,IXC+IXCDEL,IYC+IYCDEL)
         CALL RVBLK(ICT,IPIX,IXC+IXCDEL,64,IYC+IYCDEL,64)
         CALL DFILW("WIND1.PI",IER)
         CALL CFILW("WIND1.PI",3,18,IER)
         CALL OPEN(1,"WIND1.PI",3,IER)
         IF(IER.EQ.1)GO TO 194
         TYPE "* * ERROR OPENING WIND1.PI FILE * *"
194      CALL WRBLK(1,0,IPIX(2),16,IER)
         CALL CLOSE(1,IER)

192     ITOPNOSE=GESTDATA(275)+ITOPNDEL
        IMOUTH=GESTDATA(276)+ITOPNDEL
        GESTDATA(275)=ITOPNOSE
        GESTDATA(276)=IMOUTH
        ILFTEYE=GESTDATA(261)
        IRHTEYE=GESTDATA(262)
        MAXDEL=GESTDATA(274)-GESTDATA(273)
        ITOPEYE=ITOPNOSE-MAXDEL
        ITOPHEAD=ITOPEYE-MAXDEL
        IF(ITOPHEAD.LT.1)GO TO 205
        GO TO 196
205     TYPE
        TYPE "FACIAL IMAGE IS TOO LARGE TO SAVE AS A 64X64 PICTURE"
        TYPE
        CALL BOXCUR(ICT,192,128)
        CALL HCTAB(ICT,IXC-96,IYC-64)
        GO TO 164
196     GESTDATA(285)=ITOPEYE
        GESTDATA(286)=ITOPHEAD

C       The next 3 gestdata values represent default values for the
C         dimensions of the image (64X64) and the FSTOP used (8.0).
          GESTDATA(45)=64
          GESTDATA(46)=64
          GESTDATA(47)=80
```

```
      CALL OPEN(1,"WIND1.PI",3,IER)
      CALL WRBLK(1,16,GESTDATA(45),1,IER)
      CALL WRBLK(1,17,PARAM,1,IER)
      CALL CLOSE(1,IER)
      TYPE
      TYPE "IMAGE AND FEATURE LOCATIONS SAVED AS TEST",IPIXCT
      TYPE

      IF(IPIXCT.EQ.1)CALL RENAM("WIND1.PI","TEST1.PI",IER)
      IF(IPIXCT.EQ.2)CALL RENAM("WIND1.PI","TEST2.PI",IER)
      IF(IPIXCT.EQ.3)CALL RENAM("WIND1.PI","TEST3.PI",IER)
      IF(IPIXCT.EQ.4)CALL RENAM("WIND1.PI","TEST4.PI",IER)
      IPIXCT=IPIXCT+1

C     Draw grid marks on the face at the location it was found.

      DO 210 I=ILFTEYE,IRHTEYE
         IELEM=(ITOPHEAD-1)*64+I+1
         IPIX(IELEM)=0
210   CONTINUE
      DO 220 I=ILFTEYE,IRHTEYE
         IELEM=(ITOPEYE-1)*64+I+1
         IPIX(IELEM)=0
220   CONTINUE
      DO 230 I=ILFTEYE,IRHTEYE
         IELEM=(ITOPNOSE-1)*64+I+1
         IPIX(IELEM)=0
230   CONTINUE
      DO 240 I=ILFTEYE,IRHTEYE
         IELEM=(IMOUTH-1)*64+I+1
         IPIX(IELEM)=0
240   CONTINUE

C     DELINEATE THE CENTER OF EYES
      ICTREYE=GESTDATA(260)
      DO 250 I=ITOPHEAD,IMOUTH
         IELEM=ICTREYE+64*(I-1)+1
         IPIX(IELEM)=0
250   CONTINUE

C     DELINEATE THE LEFT SIDE OF THE EYES
      DO 260 I=ITOPHEAD,IMOUTH
         IELEM=ILFTEYE+64*(I-1)+1
         IPIX(IELEM)=0
260   CONTINUE

C     DELINEATE THE RIGHT SIDE OF THE EYES
      DO 270 I=ITOPHEAD,IMOUTH
         IELEM=IRHTEYE+64*(I-1)+1
         IPIX(IELEM)=0
270   CONTINUE

C     Display the outer boundaries of the facial image.
```

```
        IYCDEL=STRIPDATA(393)-150-ITOPNDEL
        IXCDEL=STRIPDATA(387)-128
        CALL WVBLK(ICT,IPIX,IXC+IXCDEL,64,IYC+IYCDEL,64)

        IF(IPIXCT.GE.5)GO TO 170

        TYPE
        TYPE "CONTINUING SEARCH"
        TYPE
        TYPE "HIT OCTEK KEYPAD #7 TO DISCONTINUE THE SEARCH"
        CALL BOXCUR(ICT,192,128)
        CALL HCTAB(ICT,IXC-96,IYC-64)
        GOTO 164

190     CONTINUE

C       Terminate Eclipse Processing
        CALL CFILW("TERMFACE",2,IER)

C       Exit.

        TYPE "<33>J Processing done."
        GOTO 999

200     CONTINUE
        TYPE "<33>J ** Camera Input is missing (unable to find video sync)"
        GO TO 40


999     CONTINUE
        CALL OREMOVE (ICT)
        CALL RESET
        END
```

## Bibliography

1. Baron, Robert J. "Mechanisms of Human Facial Recognition," International Journal of Man-Machine Studies, Vol 15, 137-178 (1981).

2. -----. "A Bibliography on Face Recognition," The SISTM Quarterly Incorporating The Brain Theory Newsletter, Vol II, No. 3: 27-36 (1979).

3. Bledsoe, Woodrow W. Man-machine facial recognition: Report on a large scale experiment. Palo Alto; Panoramic Research, August 1,1966.

4. Bromley,L.K. Computer-Aided Processing Techniques For Useage in Real-Time Image Evaluation. Masters Thesis, University of Houston, May 1977.

5. Bruce, V. and Young, A. "Understanding Face Recognition," British Journal of Psychology, Vol 77, part 3, 305-327 (August 1986).

6. Davies, Graham et al. Percieving and Remembering Faces. London: Academic Press, 1981.

7. Ellis, Haydn D. et al. "Identification of Familiar and Unfamiliar Faces from Internal and External Features: Some Implications for Theories of Face Recognition," Perception, Vol 8, 431-439 (1979).

8. Fiorentini, Adriana et al. "The Role of High Spatial Frequencies in Face Perception," Perception, Vol 12, 195-201 (1983).

9. Ginsburg, Arthur P. Visual Information Processing Based on Spatial Filter Constrained by Biological Data. Phd Dissertation, AMRL-TR-78-129, Aerospace Medical Research Lab, DTIC Document, December 1978.

10. Haig, Nigel D. "The Effect of Feature Displacement on Face Recognition," Perception, Vol 13, 505-512 (1984).

11. Harmon, L.D. "Machine Identification of Human Faces," Pattern Recognition, Vol 13, No. 2, 97-110 (1981).

12. -----. "Some Aspects of Recognition of Human Faces," Pattern Recognition in Biological and Technical Systems, Springer Verlag, New York, 1971. Not in circulation.

13. Inui, T. and Miyamoto, K. "The Effect of Changes in Visible Area on Facial Recognition," Perception, Vol 13, 49-56 (1984).

14. Kanade,T. Picture Processing System by Computer Complex and Recognition of Human Faces. PhD Dissertation, Department of Information Science, Kyoto University, Kyoto, Japan, 1973.

15. Kelly,M.D. Visual Identification of People by Computer. PhD Dissertation, Computer Science Department, Stanford University, Palo Alto, California, 1970.

16. Perrett, D.I. et al. "Neurones Responsive to Faces in the Temporal Cortex: Studies of Functional Organization, Sensitivity to Identity and Relation to Perception," Human Neurobiology, Vol 3, 197-208 (1984).

17. Perrett, D.I. et al. "Visual Neurones Responsive to Faces in the Monkey Temporal Cortex," Experimental Brain Research, Vol 47, 329-342 (1982).

18. Preston, K., Jr. "Computing at the Speed of Light," Electronics, Vol 38, 72-83 (1965).

19. Rolls, E.T. et al. "Role of Low and High Spatial Frequencies in the Face-Selective Responses of Neurons in the Cortex in the Superior Temporal Sulcus in the Monkey," Vision Research, Vol 25, No. 8, 1021-1035 (1985).

20. Rolls, E.T. "Neurons in the Cortex of the Temporal Lobe and in the Amygdala of the Monkey with Responses Selective for Faces," Human Neurobiology, Vol 3, 209-222 (1984).

21. Routh, Richard LeRoy. Cortical Thought Theory: A Working Model of the Human Gestalt Mechanism. Phd Dissertation, AFIT/DS/EE/85-1, Air Force Institute of Technology,DTIC Document, July 1985.

22. Russel, Robert L. Jr. Performance of a Working Face Recognition Machine using Cortical Thought Theory. Masters Thesis, AFIT/EE/85-D, Air Force Institute of Technology, DTIC Document, December 1985.

23. Sakai,T., Nagao,M. and Kanade,T. "Computer Analysis of Photographs of Human Faces," Transactions of the Institute of Electronics and Communication Engineers of Japan Abstracts, Vol 56, 29 (1973).

24. Sakai,T. Nagao,M. and Fujibayashi,S. "Line Extraction and Pattern Detection in a Photograph," Pattern Recognition, Vol 1, 233-248 (1969).

25. Walker-Smith, G.J. et al. "Eye Movement Strategies Involved in Face Perception," Perception, Vol 6, 313-326 (1977).

26. Young, Andrew W. et al. "Matching Familiar and Unfamiliar Faces on Internal and External Features," *Perception*, Vol 14, 737-746 (1985).

# Vita

Captain Edward J. Smith was born on 4 April 1951 in Brooklyn, New York. He recieved a Bachelor of Science degree in Chemistry from Eastern Illinois University in 1976. After working in the chemical industry for two years, Captain Smith attended Officer Training School (OTS) and recieved his commission in the United States Air Force in August 1979.

He recieved a Bachelor of Science degree in Electrical Engineering from the Air Force Institute of Technology (AFIT) in March 1981. Upon graduation from AFIT, Captain Smith served as manager for the development, test and evaluation of applications software to be used in the checkout, control and monitor of space shuttle electrical systems at Vandenberg Air Force Base. He entered the AFIT School of Engineering Master's Program in June 1985.

Permanent address: 2088 Florence Ave.
Hazlet, New Jersey 07735

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

AD-A1[illegible]-85

# REPORT DOCUMENTATION PAGE

**1a. REPORT SECURITY CLASSIFICATION:** UNCLASSIFIED

**1b. RESTRICTIVE MARKINGS:** N/A

**2a. SECURITY CLASSIFICATION AUTHORITY:**

**2b. DECLASSIFICATION/DOWNGRADING SCHEDULE:**

**3. DISTRIBUTION/AVAILABILITY OF REPORT:** APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED.

**4. PERFORMING ORGANIZATION REPORT NUMBER(S):** AFIT/GE/ENG/86D-36

**5. MONITORING ORGANIZATION REPORT NUMBER(S):**

**6a. NAME OF PERFORMING ORGANIZATION:** School of Engineering, Air Force Inst. of Tech.

**6b. OFFICE SYMBOL (If applicable):** AFIT/ENG

**7a. NAME OF MONITORING ORGANIZATION:**

**6c. ADDRESS (City, State and ZIP Code):** Wright-Patterson AFB, OH 45433

**7b. ADDRESS (City, State and ZIP Code):**

**8a. NAME OF FUNDING/SPONSORING ORGANIZATION:**

**8b. OFFICE SYMBOL (If applicable):**

**9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER:**

**8c. ADDRESS (City, State and ZIP Code):**

**10. SOURCE OF FUNDING NOS.**

| PROGRAM ELEMENT NO. | PROJECT NO. | TASK NO. | WORK UNIT NO. |
|---|---|---|---|
| | | | |

**11. TITLE (Include Security Classification):** See box 19

**12. PERSONAL AUTHOR(S):** Edward J. Smith, Captain, USAF

**13a. TYPE OF REPORT:** Masters Thesis

**13b. TIME COVERED:** FROM ______ TO ______

**14. DATE OF REPORT (Yr., Mo., Day):** 86 DEC 08

**15. PAGE COUNT:** 225

**16. SUPPLEMENTARY NOTATION:**

**17. COSATI CODES**

| FIELD | GROUP | SUB. GR. |
|---|---|---|
| 09 | 02 | |

**18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number):** COMPUTERS, IMAGE PROCESSING, ARTIFICIAL INTELLIGENCE, IMAGE SEGMENTATION

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


TITLE: Development of an Autonomous Face Recognition Machine

Thesis Chairman: Matthew Kabrisky, PhD
Professor of Electrical Engineering


Approved for public release: IAW AFR 190-1.
LYNN E. WOLAVER 5 Mar 87
Dean for Research and Professional Development
Air Force Institute of Technology (ATC)
Wright-Patterson AFB OH 45433

**20. DISTRIBUTION/AVAILABILITY OF ABSTRACT:** UNCLASSIFIED/UNLIMITED [X] SAME AS RPT. [ ] DTIC USERS [ ]

**21. ABSTRACT SECURITY CLASSIFICATION:** UNCLASSIFIED

**22a. NAME OF RESPONSIBLE INDIVIDUAL:** Matthew Kabrisky, Professor, GS-15

**22b. TELEPHONE NUMBER (Include Area Code):** 513-255-5276

**22c. OFFICE SYMBOL:** AFIT/ENG

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

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

A computer algorithm was developed which successfully locates and identifies human face(s) that are present in a digitized computer image. In the process of finding the facial image, the algorithm simultaneously determines the boundary locations of the sides of the eyes, the center of the face, the tip of the nose and the vertical center of the mouth.

Detection of facial images is based on analysis a digitized scene for the presence of characteristic facial feature signatures for the eyes, nose and mouth. These signatures are generated by the application of a "center of mass" calculation to each pixel row and column for various sub-sections of the digitized scene. The presence of a face is confirmed, and its feature locations are determined based on the presence and location of the local maxima and minima which occur in these curvilinear signatures.

Once the face(s) have been located, individual face recognition is performed by calculating the "gestalt point" for six different regions of the face. The gestalt is a location, in the 2-dimensional facial region, which corresponds closely to the center of mass of the pixel intensity distribution for that region. Identification of the unknown face is performed by comparing its gestalts with those for known faces, using a distance metric.

The algorithm's face locator function was tested on 139 facial images representing thirty different subjects. The algorithm successfully located and bounded the internal region of the face in 94% of the cases. Further tests, against a limited number of arbitrary backgrounds, indicate that the algorithm is highly specific for faces.

The face recognition portion of the algorithm was tested against a database of 20 different subjects. In two trial runs, using the same 20 person database, 18 of the 20 subjects were in the top three candidates selected. In one trial run, the algorithm successfully identified the unknown individual (selected the proper individual as the number one candidate) 60% of the time. In the other trial run the proper candidate was identified 50% of the time. Recognition was based solely on analysis of the internal facial features.

END
5-87
DTIC