NBSIR 88-3831 R

# AFRS Performance Evaluation Tests

R. T. Moore, R. Michael McCabe and R. Allen Wilkinson

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
Institute for Computer Science and Technology
Gaithersburg, MD 20899

August 1988

Interim Report

DEPARTMENT OF COMMERCE UNITED STATES
75 Years Stimulating America's Progress
1913-1988

Prepared for:

**Federal Bureau of Investigation**
**U.S. Department of Justice**
**Washington, DC 20535**

NBSIR 88-3831

# AFRS PERFORMANCE EVALUATION TESTS

R. T. Moore, R. Michael McCabe and R. Allen Wilkinson

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
Institute for Computer Science and Technology
Gaithersburg, MD 20899

August 1988

Interim Report

Prepared for:
Federal Bureau of Investigation
U.S. Department of Justice
Washington, DC 20535

U.S. DEPARTMENT OF COMMERCE, C. William Verity, *Secretary*
NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Director*

## Table of Contents

# AFRS PERFORMANCE EVALUATION TESTS

by

R. T. Moore, R. Michael McCabe and R. Allen Wilkinson

## INTRODUCTION

The FBI's Automatic Fingerprint Reader Systems (AFRS) are designed to scan life-sized fingerprint images and to detect selected features including, but not always limited to, minutiae. The position and orientation of these minutiae are recorded in units of X, Y, and Theta. For a variety of reasons, an AFRS generally fails to detect all of the true minutiae in a fingerprint and generally makes a number of false detections. Also, the true minutiae that are detected in different readings of the same fingerprint impression are not necessarily found in exactly the same relative positions. Thus, the accuracy, consistency and reliability of minutiae detection are useful measures of reader performance.

## PERFORMANCE METRICS

Historically,[1] reader performance has been evaluated by superimposing a plot of the minutiae detections made by the fingerprint reader on a photograph of the fingerprint enlarged to the same scale as the plot. Then, a fingerprint expert marks the plot indicating each true, false, or missing minutiae in the area of the fingerprint that is being considered. Limits of positional accuracy and angular accuracy are applied and each minutia detected by the reader is classed as either a true or a false minutia. A detection score is then assigned:

$$\text{Detection Score} = (D - M - F) / T \qquad (1)$$

where D = number of true minutiae detected by reader
M = number of true minutiae missed by reader
F = number of false minutiae detected by reader
T = total number of true minutiae in fingerprint

It should be noted that the scoring in Equation 1 penalizes a missed minutia twice as heavily as a false minutia.

Scores can range from 1.00 for perfect performance (no missed or false minutiae) down to large negative values which might result with few true detections and many false detections.

This procedure for evaluating reader performance is useful but quite labor intensive. As a consequence, reader performance evaluation in the past has usually been limited to the

---

[1] R. M. Stock and C. W. Swonger, Development and Evaluation of a Reader of Fingerprint Minutiae, Cornell Aeronautical Laboratory Technical Report CAL No. XM-2478-X-1, Contract J-FBI-6499, January 1969.

examination of only a small number of sample fingerprints. Until fairly recently, the measurement of reader consistency, or repeatability, has received very little attention.

Recent work has been directed toward automating certain portions of the process. In particular, computer programs have been developed[2] that permit two sets of minutiae data to be aligned in translation and rotation to a "best fit" position. These programs are adapted from the M-40 matcher algorithm and are applied in four stages in an iterative manner. From the final positioning, tolerances are applied to the relative displacement and orientation of minutiae pairs that are nearly aligned with each other to determine whether or not they are mates.

If one of the two sets of minutiae data is considered "ground truth" data, then the number of minutiae pairs that are within the tolerance limits establishes the value of D in Equation 1. The total number of minutiae in the "ground truth" data set establishes the value of T. The value of M is equal to T - D, while F is equal to the total number of minutiae detected in the data set being compared minus the value of D.

## COMPARISON WITH "GROUND TRUTH"

In a typical instance, a fingerprint is read manually using any one of several different types of semi-automatic terminals. The reading is performed twice; once by each of two different operators. Next, the minutiae from each reading are plotted out at 10X enlargement. The two plots are then overlaid and any discrepancies are easily identified and can be resolved. The resulting minutiae list, after all anomalies have been resolved, is considered a reasonable estimate of "ground truth" for that fingerprint. It may not exactly coincide with the machine read data for mating pairs of minutiae on that fingerprint simply because the human expert and the machine sometimes use slightly different rules in assigning position and orientation to a minutia, but usually these differences tend to be small.

When the data from a machine read fingerprint is compared with the "ground truth" data, the values for entry in Equation 1 can be developed. In addition, the distribution of the displacements in the position of mating minutiae pairs can be determined. The displacement is calculated as the straight line distance between each of the mating pairs whose X, Y and Theta differences are within the tolerance limits that have been established. It is calculated as:

$$S = \sqrt{(X^2 + Y^2)} \qquad (2)$$

where S = straight line distance
X = displacement in X
Y = displacement in Y

When "ground truth" minutiae data are recorded, each minutia is given a reference number. Each minutia of machine read minutiae data is also assigned a reference number. The comparison programs use these reference numbers to identify mating pairs of minutiae. These reference numbers are also used in recording how frequently each "ground truth" minutiae is

---

[2] A description and listing of these programs appears in the Appendix.

detected on successive readings of the same fingerprint, or on repeated processing of the gray scale data developed during a single reading of the fingerprint.

## COMPARISON WITH "FALSE GROUND TRUTH"

The programs also provide an alternative means for evaluating the consistency of reader performance given identical gray scale fingerprint information as the input data on successive runs. Here, a "false ground truth" data set is calculated and used to determine how repeatedly each minutia, either true or false, is detected.

The recorded gray scale data is input to the reader the first time and the minutiae are detected. Each minutia in the first set of detections from the fingerprint is considered a potential site for a "false ground truth" minutia "cluster" location. Then, the recorded gray scale data is input a second time and the minutiae are detected again. The minutiae of the second set of detections are then translated and rotated for "best fit" with the first set of minutiae. The position and orientation of the minutiae in the second set of detections are compared with those of the first and those that are within the selected tolerance limits are declared mates. Each mating pair of minutiae is the basis for adjustment in the location of its "cluster" site. The new location of the "cluster" site is the mean position of its two members. Minutiae that are beyond the tolerance limits for displacement and/or orientation do not have mates. Their locations establish candidate new sites for additional "clusters" to be formed with minutiae detected from the third and subsequent passes of the recorded gray scale data through the reader.

This process is repeated on the subsequent minutiae detections made from the fingerprint data. As additional minutiae on the subsequent runs are found to be located within the tolerance limits of position and orientation that are established for mates, they become members of that "cluster", or become potential sites for new "clusters". The center point of each "cluster" is continually recalculated on the basis of the positions of the members of that "cluster". Occasionally, minutiae will leave one "cluster" and join another as a result of this recalculation of center position of the "cluster".

An analysis of these "cluster" sizes provides an indication of reader consistency. Ideally, all "clusters" should have a number of members equal to the number of times that the identical gray scale data was passed through the system for minutiae detection. Smaller "clusters" are indicative of inconsistencies in performance.

## COMPARISON WITH FIRST PASS DATA

A third means of evaluating consistency is provided when minutiae data from the first of a series of passes of gray scale data are used as a reference. These data are compared with the minutiae data from each of the succeeding passes of the identical gray scale data. The "cluster" size information developed from this routine is similar but not identical to that developed from use of the "false ground truth" procedure described above. This is because there is no recalculation of "cluster" center position as members are added to it. In addition to cluster size information, this routine also provides information on the distribution of the values of the displacement in the position of the mating minutiae. Since the same gray scale information is being input on each pass, this displacement is assumed to be caused by noise in the minutiae detection electronics.

## TEST RESULTS

In connection with the conversion of the AFRS from a flying spot scanner to a solid state scanner, certain special test materials were prepared. Among these was a recording of the fingerprints of a single individual using three different inking densities, "Light", "Medium", and "Heavy". These fingerprints were scanned by the digital scanner on AFRS No. 4 and the gray scale data were recorded on disk. This permitted the same gray scale data to be directed repeatedly to the AFRS preprocessor for minutiae detection. Ideally, the same gray scale data should produce an identical minutiae list each time it is processed. Differences in the identity or position of the detected minutiae represent imperfect performance of the preprocessor which might be caused by noise or other factors.

This gray scale data recorded on disk was input to the preprocessor of AFRS No. 4 eight times. The detected minutiae were registered and clipped and then recorded as eight fingers of data from each of three cards. The data from the lightly inked card were recorded as card 105, medium as card 203, and heavy as card 301 from system No. 4.

The same procedure was followed using the same fingerprints on AFRS No. 2. This system has had decoupling capacitors installed on the printed circuit cards in the preprocessor.

## "GROUND TRUTH" MATCH RESULTS

The "ground truth" data for these three fingerprints was obtained using the Graphic Pen[3] at NBS. The objective was to record only those minutiae that appeared within the area covered by the clipping box. This would provide minutiae from an area that was common to the area of the machine-read minutiae data. A plot of the machine read data was centered in the field of view of the Graphic Pen in order to approximately define the clipping box boundaries. Then the fingerprint was positioned under the magnifier lenses and adjusted to make the minutiae agree with those on the machine-read plot. Next the outline of the clipping box was centered on the machine-read plot. Finally the minutiae of the fingerprint that were within or even slightly beyond the boundaries of the clipping box were read by two different operators to develop the "ground truth" data base for the minutiae of that finger.

There may be an element of uncertainty in this process. This comes about because in the machine-read data, the clipping box is centered on a position that is found as a result of processing the ridge flow data derived from the machine-read gray scale information. If noise in the system affects minutiae positions, it may also affect ridge flow data. This might cause the clipping box boundaries to be slightly different for each pass of identical gray scale data through the preprocessor. Because of this, some of the minutiae near the boundaries of the clipping box in the "ground truth" data set may actually be outside the boundaries of the clipping box for the machine-read data on one or more of the processing passes. Candidate minutiae have been identified that might have been missed for this reason. These candidates consist of "ground truth" minutiae that were located very close

---

[3] R. T. Moore and J. R. Park, "The Graphic Pen - An Economical Semiautomatic Fingerprint Reader", Proc. 1977 Carnahan Conference on Crime Countermeasures, UKY BU112, ORES, Univ of Kentucky, Lexington, KY 40506.

to the clipping box boundary that were not detected in any of the multiple passes of the gray scale data through the preprocessor.

On the lightly inked fingerprint there were 83 minutiae on the initial "ground truth" list. Of these, nine minutiae, Nos. 1, 2, 3, 4, 8, 12, 23, 56, and 78 were not detected in any pass of the data. It is reasonable to assume that they might have been outside the clipping box. A value of 74 is therefore used for T in Equation 1 for this fingerprint. On the fingerprint with medium inking, there were 75 minutiae on the initial list and three of these, Nos. 1, 49 and 53 were not detected in any pass of the data, so T is assumed to have a value of 72. On the fingerprint with the heavy inking, there were 76 minutiae on the initial "ground truth" list. Four of these, Nos. 5, 6, 19 and 51 had no detections in any pass of the machine-read data so a value of 72 is assumed for T.

The tolerance limits on displacement and orientation for a pair of minutiae to be considered mates are 0.4 mm (four X, Y matcher units) and 12 degrees (Theta).

With the gray scale data from these three fingerprints recorded on disk and then entered as eight independent passes into the preprocessor of AFRS No.4, the values for use in Equation 1 are shown in the following Table 1.

TABLE 1

Detection Scores for AFIS No. 4

| | Light | | | Medium | | | Heavy | | |
|---|---|---|---|---|---|---|---|---|---|
| Pass | D | M | F | D | M | F | D | M | F |
| 1 | 18 | 56 | 60 | 20 | 52 | 54 | 28 | 44 | 60 |
| 2 | 20 | 54 | 46 | 28 | 44 | 49 | 30 | 42 | 57 |
| 3 | 14 | 60 | 48 | 27 | 45 | 43 | 35 | 37 | 58 |
| 4 | 20 | 54 | 51 | 19 | 51 | 47 | 34 | 38 | 49 |
| 5 | 18 | 56 | 57 | 24 | 48 | 51 | 33 | 39 | 53 |
| 6 | 15 | 59 | 57 | 22 | 50 | 40 | 40 | 32 | 46 |
| 7 | 16 | 58 | 51 | 24 | 48 | 53 | 33 | 39 | 54 |
| 8 | 19 | 55 | 50 | 24 | 48 | 45 | 34 | 38 | 49 |
| Tot. | 140 | 452 | 420 | 188 | 388 | 382 | 267 | 309 | 426 |
| Score: | -732/592 = -1.24 | | | -582/576 = -1.01 | | | -468/576 = -0.81 | | |

The mean number of true minutiae detected on the lightly inked print was 17.5 or 24% of the "ground truth" minutiae. On the medium inked print it was 23.5, or 33% of "ground truth". On the heavily inked print it was 33.375, or 46% of "ground truth".

On the lightly inked print, 34 different true minutiae were detected on one or more passes, but only six were consistently detected on all eight passes. On the medium inked print, 46 different true minutiae were detected on one or more passes, and only five were detected on all eight passes. With the heavily inked print, 48 true minutiae were detected on one or more passes and 16 were detected on all eight passes. The distribution of the numbers of minutiae and consistency of their detection with eight passes through the preprocessor of AFRS No. 4 is shown in Table 2.

TABLE 2

Consistency of Detections, AFRS No. 4

| LIGHT | MEDIUM | HEAVY | |
|---|---|---|---|
| 7 | 11 | 3 | Minutiae detected 1 time |
| 5 | 5 | 3 | " " 2 times |
| 6 | 8 | 5 | " " 3 " |
| 3 | 2 | 6 | " " 4 " |
| 2 | 3 | 6 | " " 5 " |
| 1 | 7 | 1 | " " 6 " |
| 4 | 5 | 8 | " " 7 " |
| 6 | 5 | 16 | " " 8 " |
| 34 | 46 | 48 | Minutiae detected one or more times |

Figure 1 shows this same data in a different way. This is a cumulative distribution of the percentage of the minutiae detections that are repeated eight times, seven times, etc.

Figures 2, 3 and 4 show distribution of distances that the minutiae were displaced from the "ground truth" position in each of the eight passes of the gray scale data through the AFRS No. 4 preprocessor for the light, medium and heavily inked image. Figure 5 shows a summary of this same information for the eight passes and all three inkings. This figure shows that the most frequent displacement is two units with the lightly inked images. The heavily inked image shows a noticeable peak at one unit of displacement.

The preprocessor of AFRS No 2. has been modified with the addition of some decoupling capacitors to its printed circuit cards in an attempt to reduce internally generated noise. The same procedures were followed with this reader as with AFRS No. 4. Gray scale data was recorded on disk and passed through the preprocessor eight times and the minutiae detections from each pass compared with the "ground truth" data from the light, medium and heavily inked fingerprint. The results are shown in Table 3. They are very comparable to the performance shown for AFRS No. 4 in Table 1. The lightly inked print provided a mean of 17.5 true minutiae or 24% of "ground truth". On the medium inked print the mean number of detections was 21.625 or 30% of "ground truth". The heavily inked print yielded a mean of 33.125 minutiae which is 46% of the "ground truth".

The distribution of numbers of minutiae and the consistency of detection is shown in Table 4.

TABLE 3

Detection Scores for AFRS No. 2

| | LIGHT | | | MEDIUM | | | HEAVY | | |
|---|---|---|---|---|---|---|---|---|---|
| Pass | D | M | F | D | M | F | D | M | F |
| 1 | 24 | 50 | 53 | 18 | 54 | 55 | 34 | 38 | 50 |
| 2 | 17 | 57 | 41 | 28 | 44 | 57 | 32 | 40 | 56 |
| 3 | 20 | 54 | 51 | 21 | 51 | 53 | 31 | 41 | 60 |
| 4 | 16 | 58 | 47 | 23 | 49 | 52 | 36 | 36 | 45 |
| 5 | 12 | 62 | 48 | 20 | 52 | 41 | 37 | 35 | 46 |
| 6 | 21 | 53 | 62 | 21 | 51 | 57 | 31 | 41 | 54 |
| 7 | 15 | 59 | 58 | 21 | 51 | 51 | 31 | 41 | 54 |
| 8 | 15 | 59 | 62 | 21 | 51 | 55 | 32 | 40 | 58 |
| Tot. | 140 | 452 | 422 | 173 | 403 | 421 | 265 | 311 | 430 |
| Score: | -733/592 = -1.24 | | | -648/576 = -1.13 | | | -476/576 = -0.83 | | |

TABLE 4

Consistency of Detections, AFRS No. 2

| LIGHT | MEDIUM | HEAVY | |
|---|---|---|---|
| 4 | 11 | 6 | Minutiae detected 1 time |
| 8 | 4 | 2 | " " 2 times |
| 2 | 3 | 3 | " " 3 " |
| 1 | 11 | 4 | " " 4 " |
| 2 | 2 | 2 | " " 5 " |
| 5 | 4 | 5 | " " 6 " |
| 6 | 4 | 7 | " " 7 " |
| 3 | 5 | 17 | " " 8 " |
| 31 | 44 | 46 | Minutiae detected one or more times |

Figure 6 shows the cumulative distribution of the repeated minutiae detections shown in Table 4.

Figures 7, 8 and 9 show distribution of distances that the minutiae were displaced from the "ground truth" position in each of the eight passes of the gray scale data through the AFRS No. 2 preprocessor for the light, medium and heavily inked image.

Figure 10 shows a summary of this same information for the eight passes and all three inkings. This figure shows that the most frequent displacement is two units with the light and medium inked images. The heavily inked image again shows a noticeable peak at only one unit of displacement.

## "FALSE GROUND TRUTH" RESULTS

"False ground truth" is the name that has been given to the set of candidate "cluster" sites in (X, Y, Theta) space representing every minutiae detection, true or false, resulting from multiple passes of the same gray scale data through the preprocessor and minutiae detection logic. Each field of detected minutiae is translated and rotated to a "best fit" position with respect to the field of minutiae detected in the first pass. Mating minutiae are identified and the "cluster" site is recalculated to be the mean position and orientation of the mates comprising the cluster. Minutiae that do not have mates within the tolerance limits of displacement and rotation still establish candidate sites that may become populated with subsequent passes of the data.

Since the gray scale data that is used is digital, performance in a noise-free environment would be expected to produce N "clusters" each having a population of M mates, where N is a constant number of minutiae detections and M is the number of times that the same gray scale data is passed through the detector. The fact that this is not the case is a matter of serious concern.

Table 5 shows the performance of AFRS No. 4 on the light, medium and heavily inked prints. The same data is shown in Fig. 11 as the cumulative distribution of the percentage of multiple detections. There is very small difference in the results from the light and medium inked print, while the heavily inked print produced noticeably more consistent detections.

Table 6 and Fig. 12 show the corresponding performance of AFRS No. 2. Here the superiority of the heavily inked print is much less pronounced.

TABLE 5

SUMMARY OF "FALSE GROUND TRUTH" CLUSTER SIZES - AFRS NO. 4

| LIGHT | MEDIUM | HEAVY | |
|---|---|---|---|
| 60 | 69 | 33 | Minutiae detected 1 time |
| 27 | 30 | 16 | " " 2 times |
| 21 | 16 | 9 | " " 3 " |
| 13 | 12 | 19 | " " 4 " |
| 16 | 8 | 11 | " " 5 " |
| 6 | 14 | 7 | " " 6 " |
| 9 | 11 | 12 | " " 7 " |
| 19 | 18 | 43 | " " 8 " |
| 171 | 178 | 150 | Different minutiae detected one or more times |

TABLE 6

SUMMARY OF "FALSE GROUND TRUTH" CLUSTER SIZES - AFRS NO.2

| LIGHT | MEDIUM | HEAVY | |
|---|---|---|---|
| 58 | 53 | 57 | Minutiae detected 1 time |
| 36 | 27 | 17 | " " 2 times |
| 14 | 11 | 12 | " " 3 " |
| 15 | 14 | 13 | " " 4 " |
| 8 | 9 | 8 | " " 5 " |
| 8 | 8 | 16 | " " 6 " |
| 14 | 15 | 12 | " " 7 " |
| 18 | 25 | 37 | " " 8 " |
| 168 | 162 | 172 | Different minutiae detected one or more times |

The software routines that developed the data shown above in Tables 5 and 6 for "false ground truth" performance list the identity of each of the minutiae that formed each of the clusters. Many of these clusters appear to have been formed from false detections that occurred repeatedly. A comparison of the numbers of minutiae detected eight times in Tables 2 and 4 with the number of minutiae detected eight times in Tables 5 and 6 reveals information about how frequently these repeated false detections occurred.

**FIRST PASS MATCH RESULTS**

This comparison takes the minutiae detections resulting from the first pass of the gray scale data from the light, the medium, and the heavily inked fingerprint and treats it as the reference against which the other seven passes of the data from each of these fingerprints are compared. As is the case of the "false ground truth" data there is no distinction made as to whether the minutiae detected are true or false. The chief utility of this comparison is to provide a measure of the displacements in position and orientation of mating minutiae that is not biased by the human position selection rules as it is in the "ground truth" data.

Figures 13 through 15 show the seven individual results obtained from each of the three degrees of inking on AFRS No. 4. Figures 16 and 17 show the summary data for this system.

Figures 18 through 20 show the seven individual results from each of the three inkings on AFRS No.2 and Figs. 21 and 22 cover the summary results for this system.

These data show that the most frequently observed displacement in position for mating minutiae is one matcher unit, although there are displacements greater than that in a few percent of the detections. With no noise, it would be expected that there would be no displacement, since the same digital data is input each time.

## CONCLUSIONS

These test results indicate that despite the improvements resulting from upgrading the readers with the new solid state scanner subsystems, there are serious problems in the AFRS. These are manifest in the form of inconsistencies in the detection of minutiae, both true and false, even when identical gray scale data is input to the preprocessor and minutiae detection circuitry. The performance is more erratic when lightly or moderately inked fingerprints are used than with heavily inked, high contrast, images. In these tests, only about one quarter of the true minutiae were detected in lightly inked prints, one third in the moderately inked prints and one half in the heavily inked prints on any single processing of the fingerprint data. Presumably the observed inconsistencies in performance result from noise in the preprocessor. Minor inconsistencies in detection performance could be expected to be caused by the recursive behavior of the ridge valley filter, but these would be expected to be constrained to the top few per cent of the fingerprint image area. Since several of the instances where minutiae were detected consistently (eight times) occur in the top five percent of the image area, it is not believed that the recursive attributes of the filter are a major contributor to the inconsistencies observed in minutiae detection.

It is believed that the detection probability displayed in these tests is not capable of supporting an effective automated latent fingerprint identification system. It is suggested that long range planning should contemplate reconversion of the files to be used for an automated latent system. Rigorous quality control measures are suggested to insure that the quality of the re-converted file data is maintained at an acceptable level. It is believed that some of the software routines listed in the Appendix would be appropriate candidates for use in support of this function.

These test data also strongly suggest that there is no significant difference in the performance of AFRS No. 2 and No. 4 and that the FBI made the correct choice in not adding the decoupling capacitors to the remainder of the AFRS preprocessors. It is believed that much more extensive measures will be required to correct the problems in these systems.

Finally, these data show a definite trend to improved performance with increasing ink density. This is in agreement with vendor claims that readers perform better with electronically generated fingerprints which provide images with high contrast.

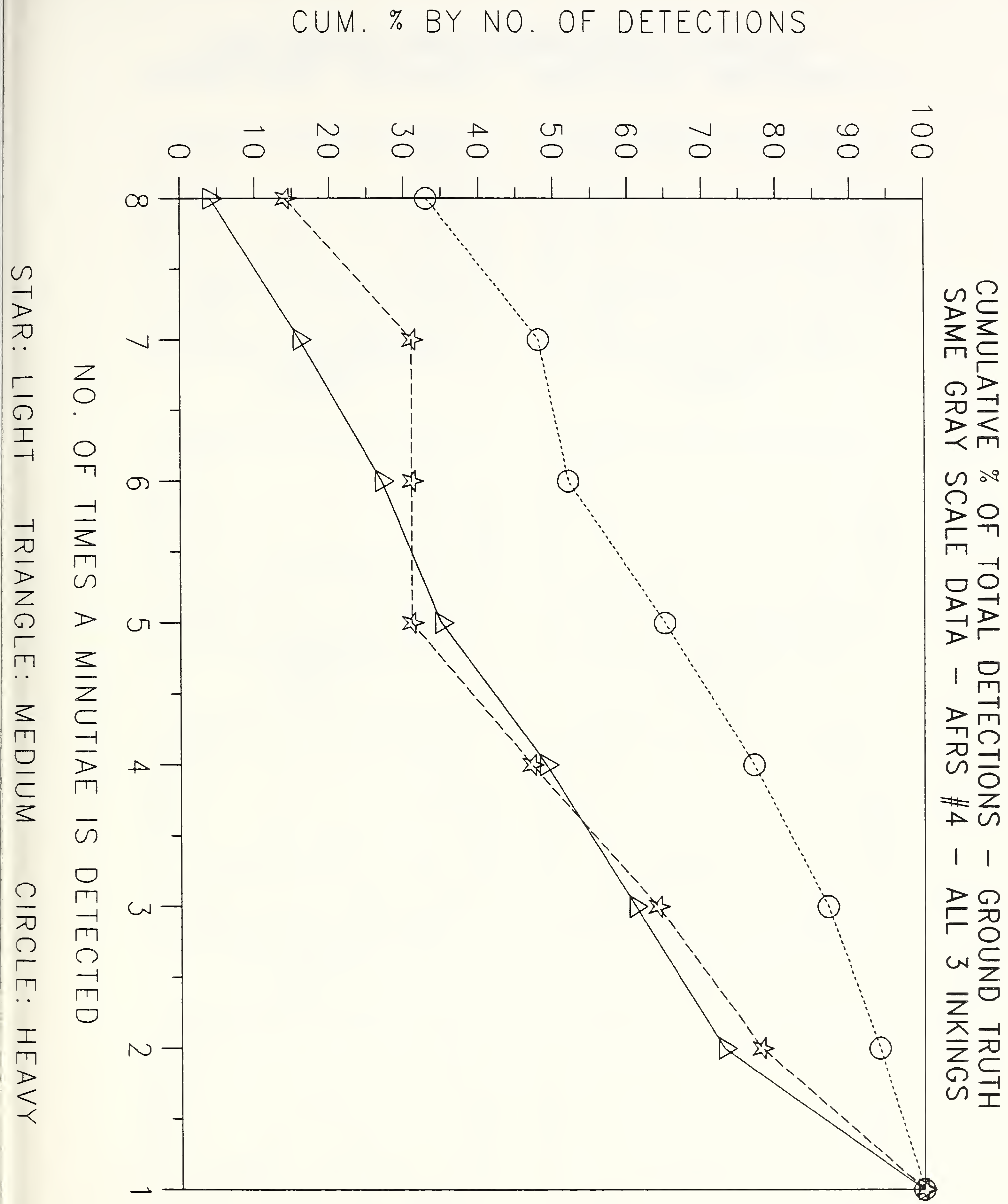

CUMULATIVE % OF TOTAL DETECTIONS - GROUND TRUTH
SAME GRAY SCALE DATA - AFRS #4 - ALL 3 INKINGS
CUM. % BY NO. OF DETECTIONS
0 10 20 30 40 50 60 70 80 90 100
8 7 6 5 4 3 2 1
NO. OF TIMES A MINUTIAE IS DETECTED
STAR: LIGHT TRIANGLE: MEDIUM CIRCLE: HEAVY

Figure 1

# MINUTIAE DISPLACEMENT FROM GROUND TRUTH
# SAME GRAY SCALE DATA – AFRS #4 – LIGHT INKING

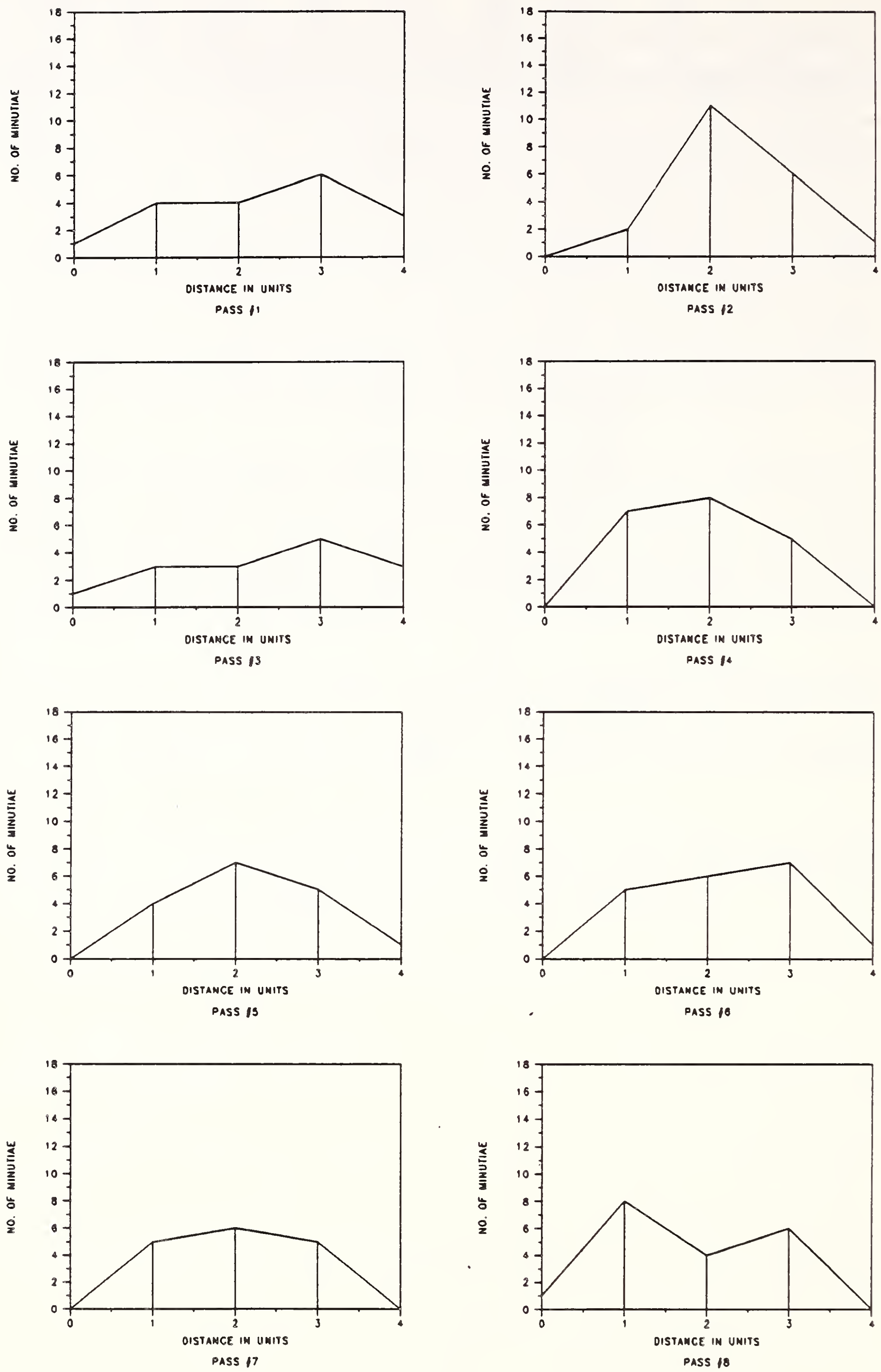



Figure 2

## MINUTIAE DISPLACEMENT FROM GROUND TRUTH
## SAME GRAY SCALE DATA – AFRS #4 – MEDIUM INKING

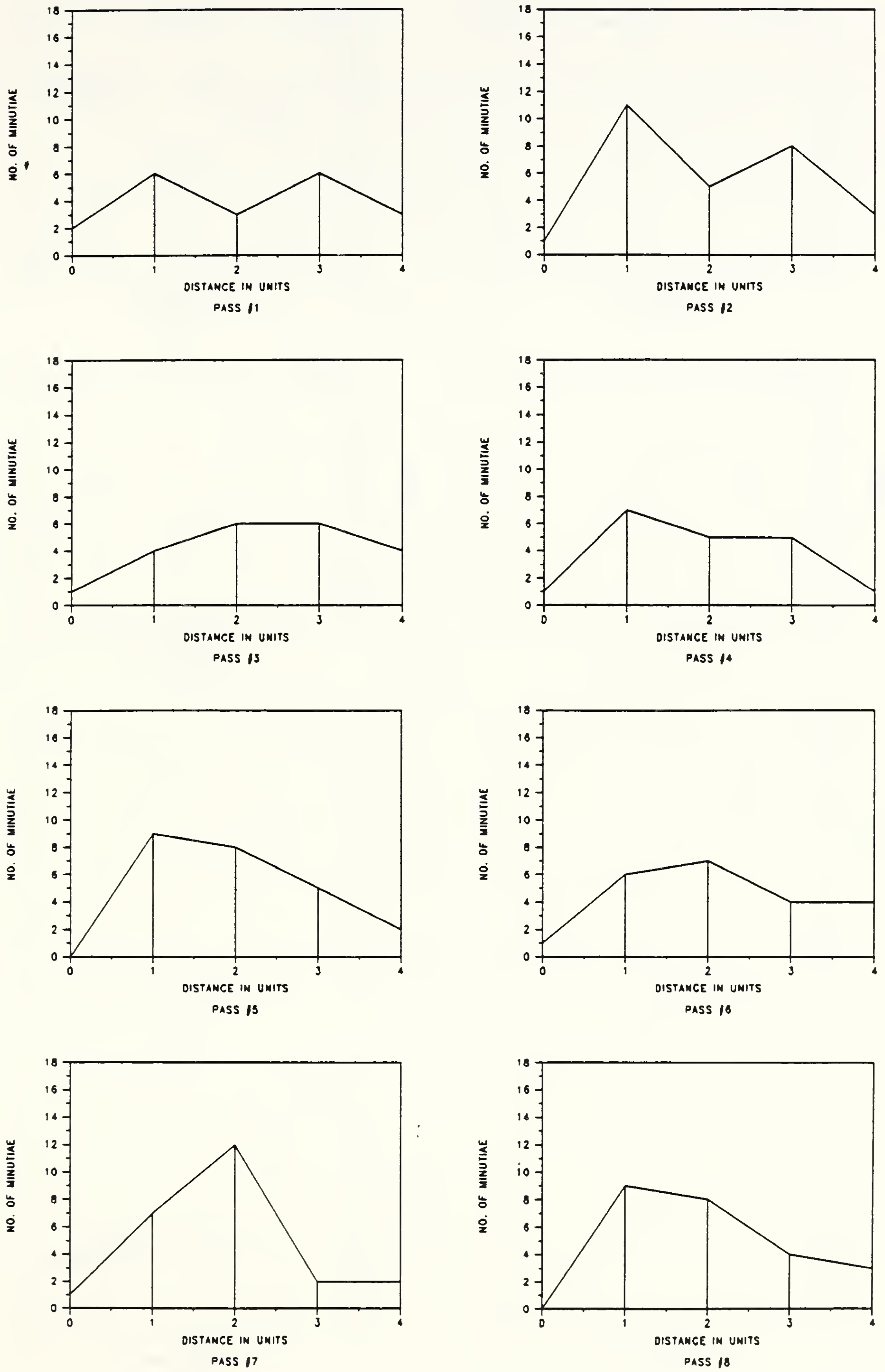



Figure 3

MINUTIAE DISPLACEMENT FROM GROUND TRUTH
SAME GRAY SCALE DATA – AFRS #4 – HEAVY INKING

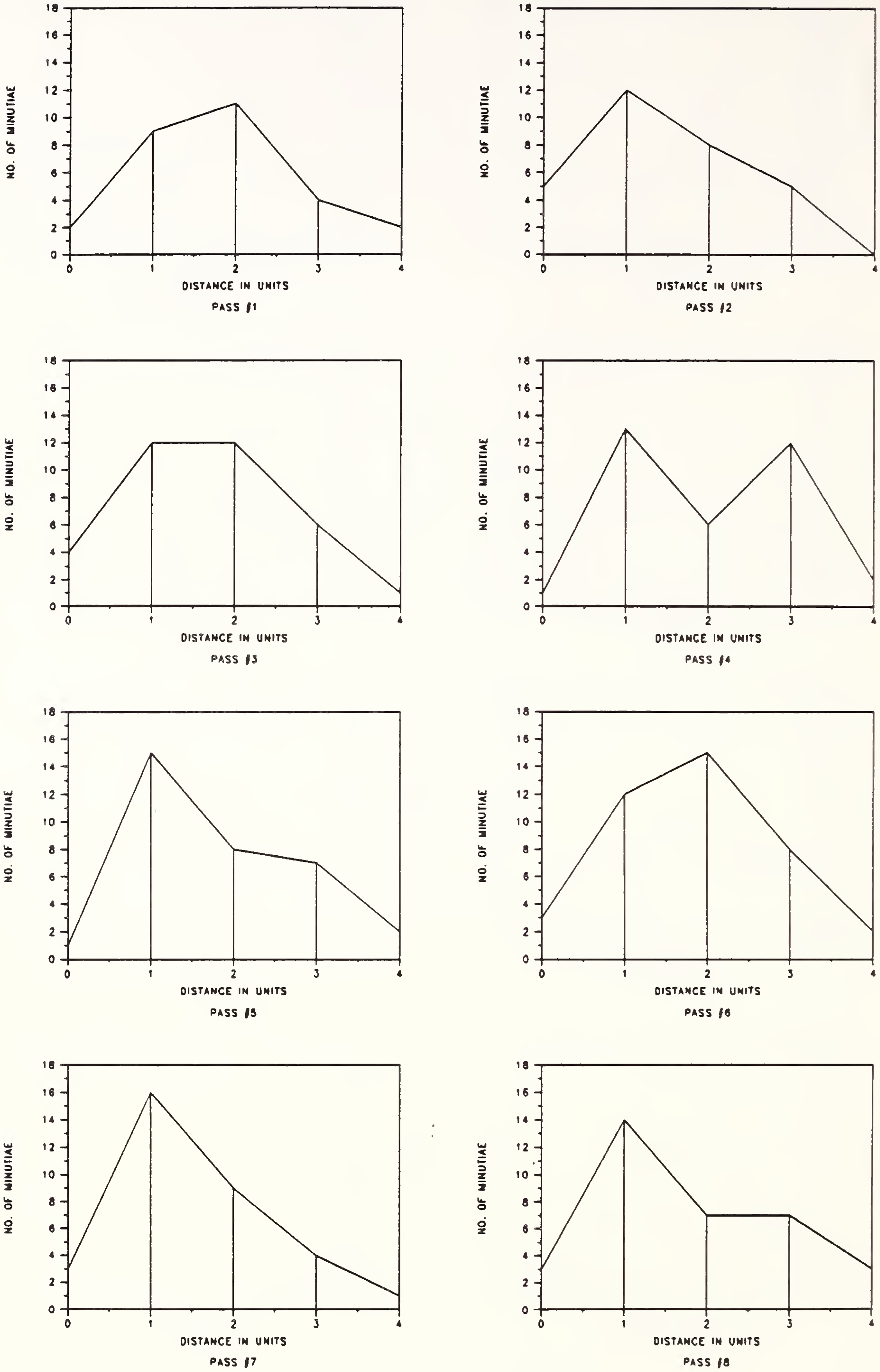



Figure 4

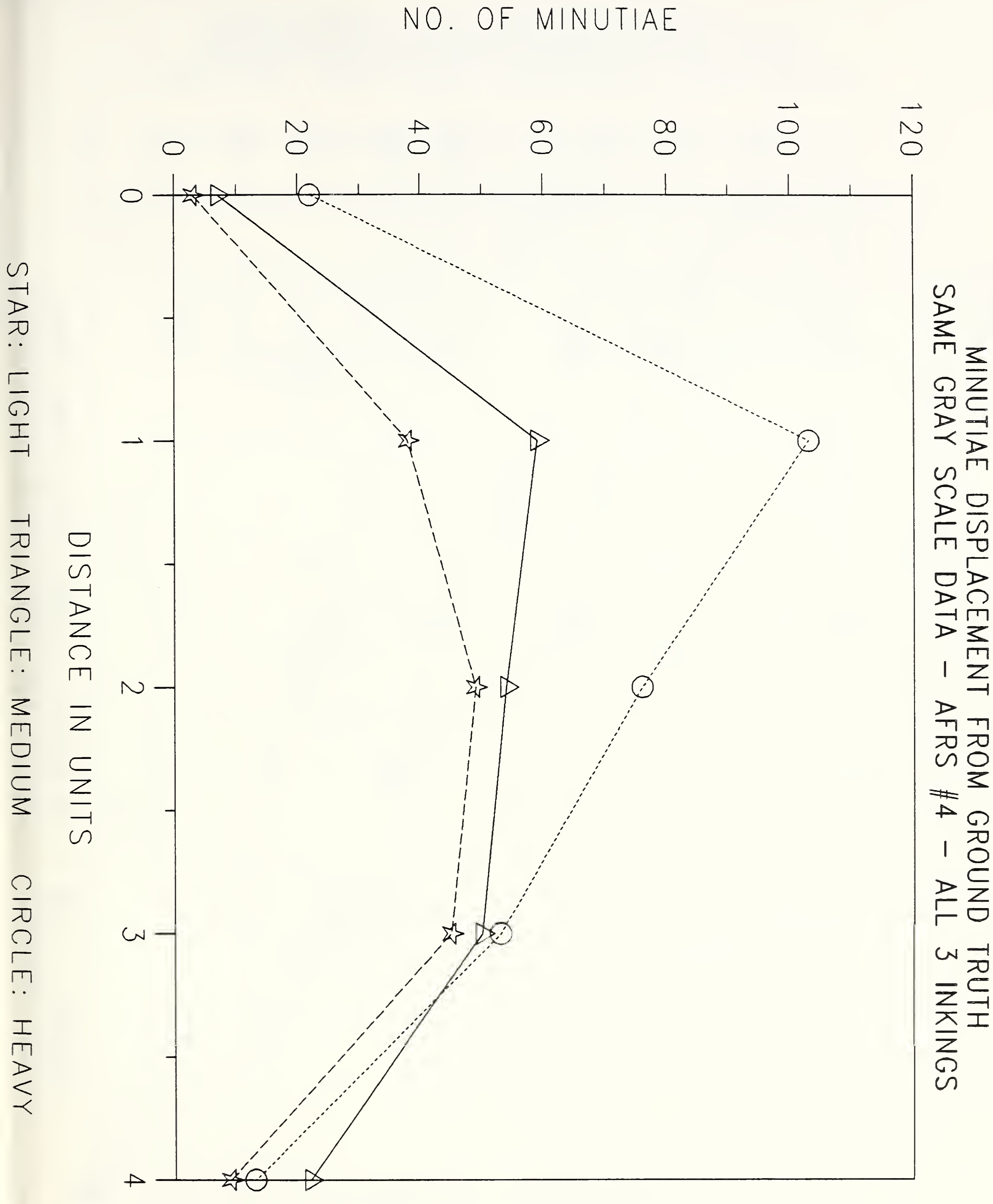

MINUTIAE DISPLACEMENT FROM GROUND TRUTH
SAME GRAY SCALE DATA - AFRS #4 - ALL 3 INKINGS
NO. OF MINUTIAE
0
20
40
60
80
100
120
0
1
2
3
4
DISTANCE IN UNITS
STAR: LIGHT TRIANGLE: MEDIUM CIRCLE: HEAVY

Figure 5

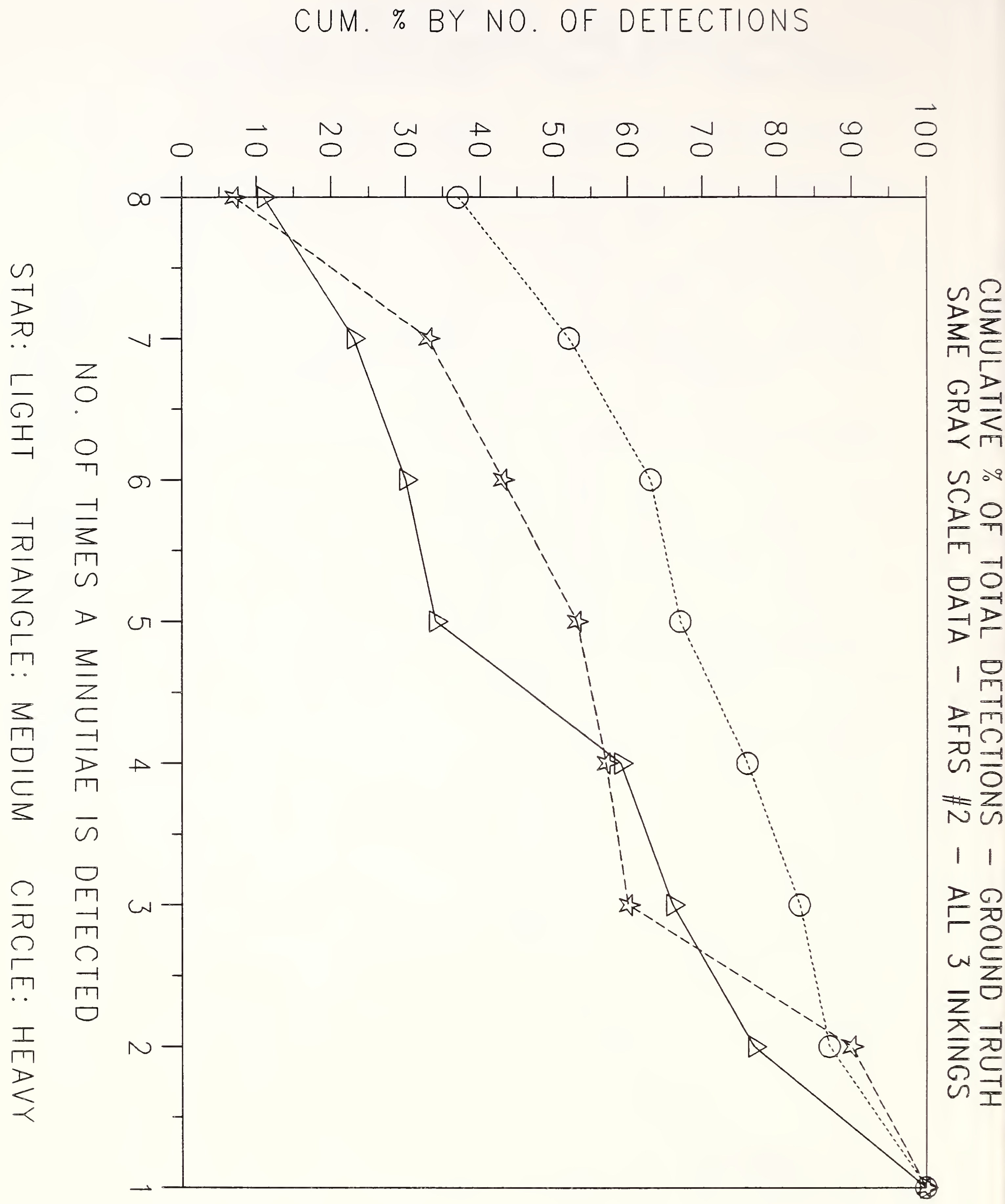

CUMULATIVE % OF TOTAL DETECTIONS - GROUND TRUTH
SAME GRAY SCALE DATA - AFRS #2 - ALL 3 INKINGS
CUM. % BY NO. OF DETECTIONS
NO. OF TIMES A MINUTIAE IS DETECTED
STAR: LIGHT TRIANGLE: MEDIUM CIRCLE: HEAVY

Figure 6

## MINUTIAE DISPLACEMENT FROM GROUND TRUTH
## SAME GRAY SCALE DATA – AFRS #2 – LIGHT INKING

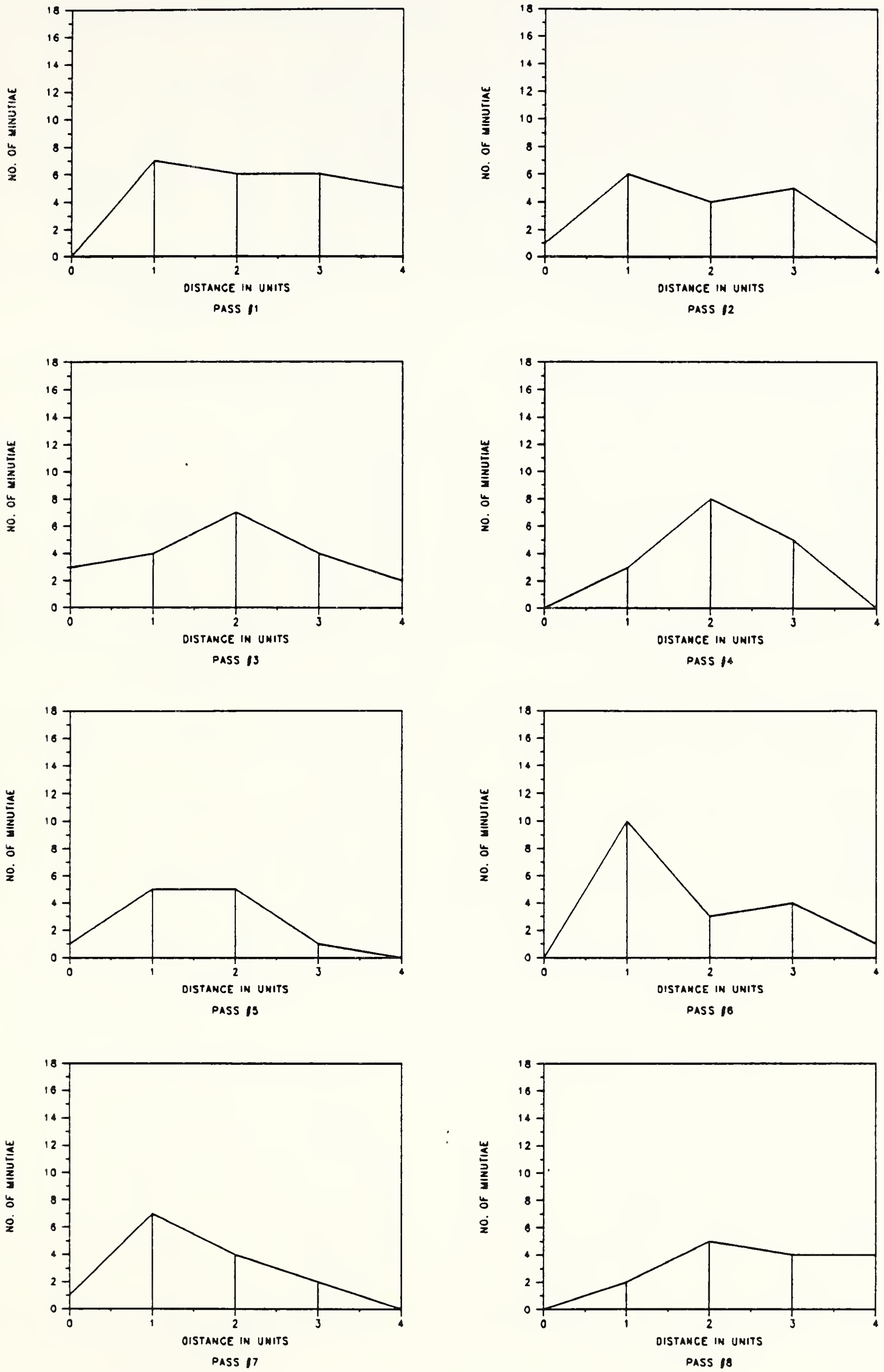



Figure 7

## MINUTIAE DISPLACEMENT FROM GROUND TRUTH
## SAME GRAY SCALE DATA – AFRS #2 – MEDIUM INKING

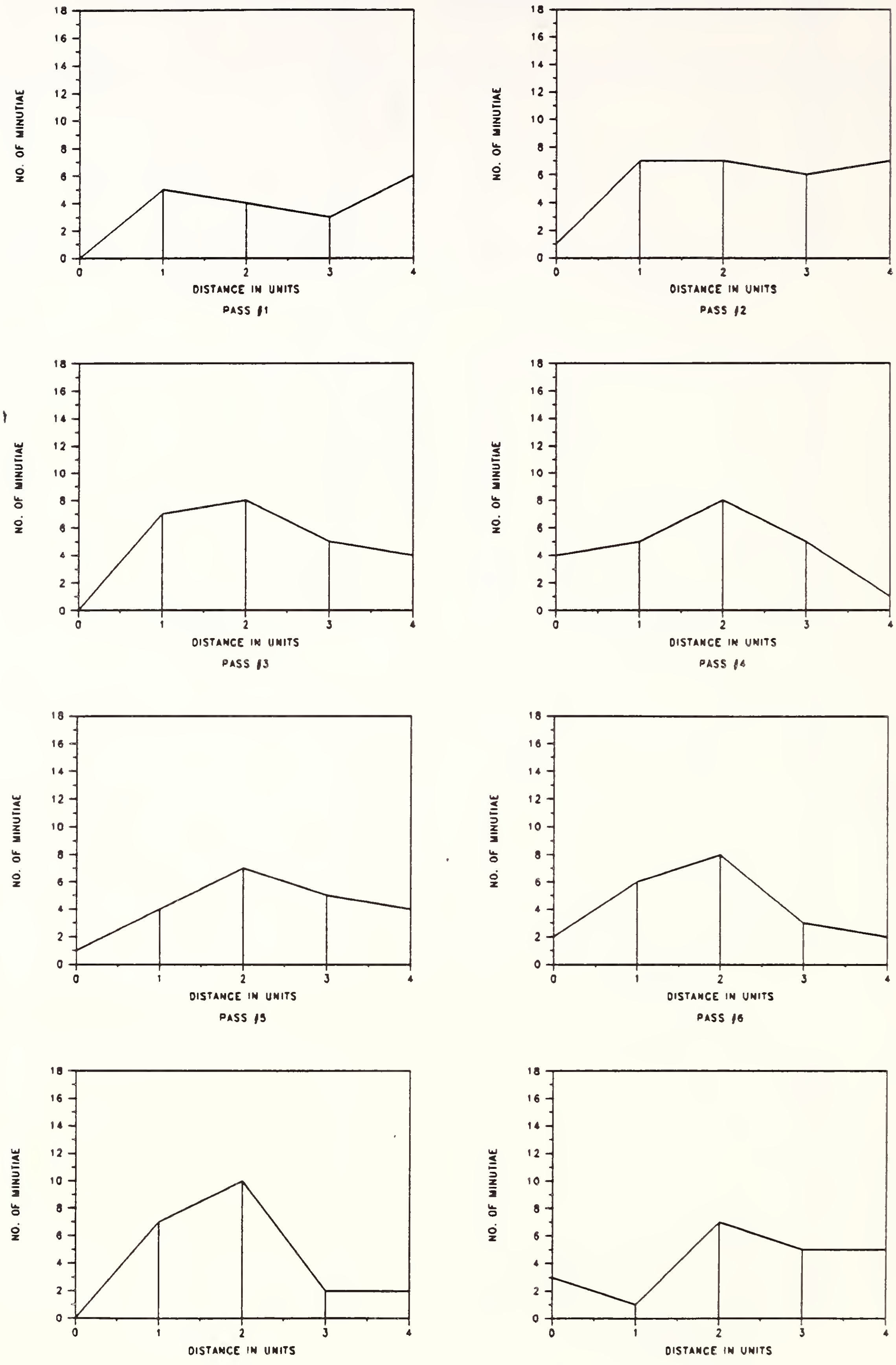



Figure 8

## MINUTIAE DISPLACEMENT FROM GROUND TRUTH
## SAME GRAY SCALE DATA – AFRS #2 – HEAVY INKING

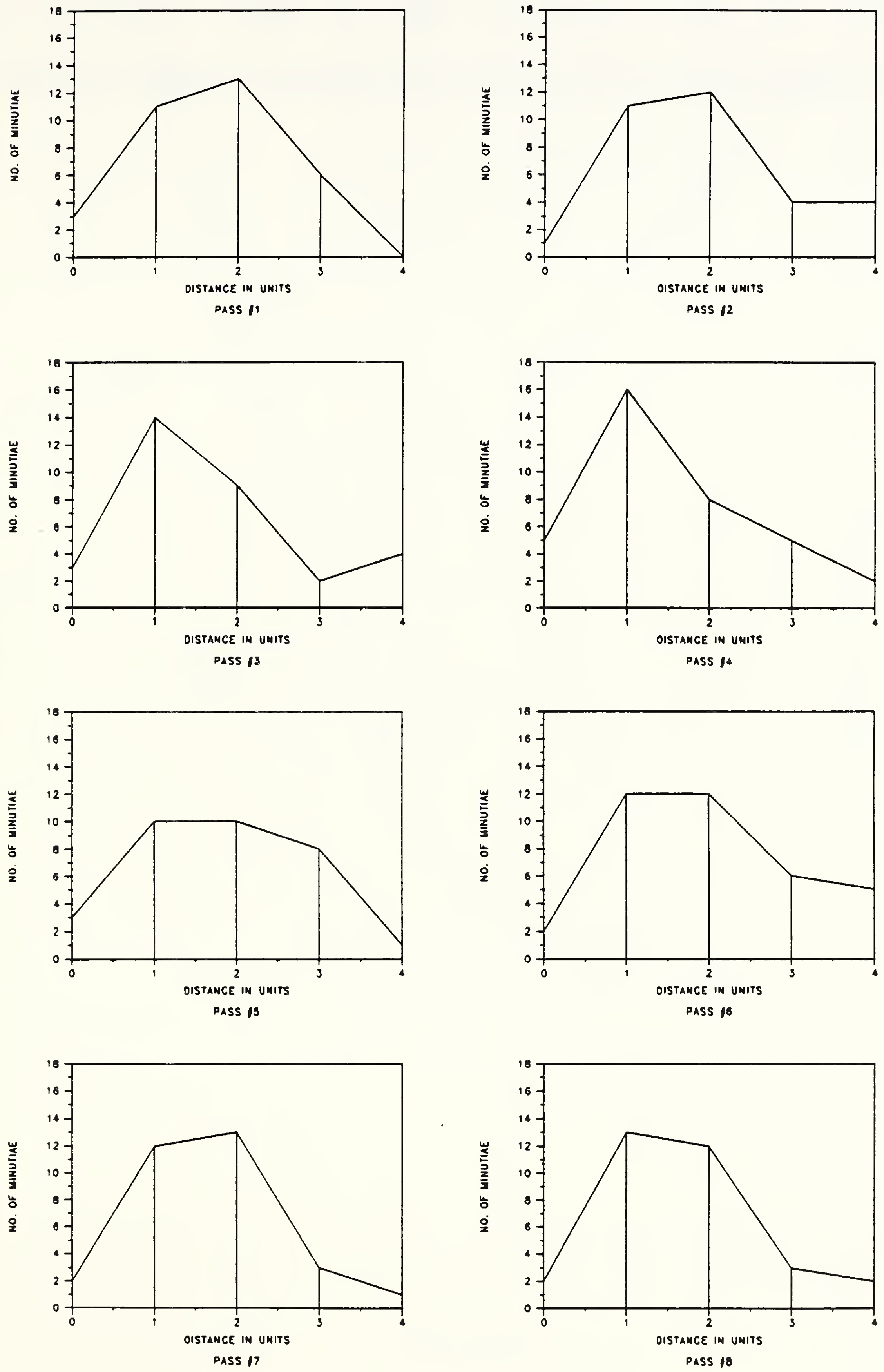



Figure 9

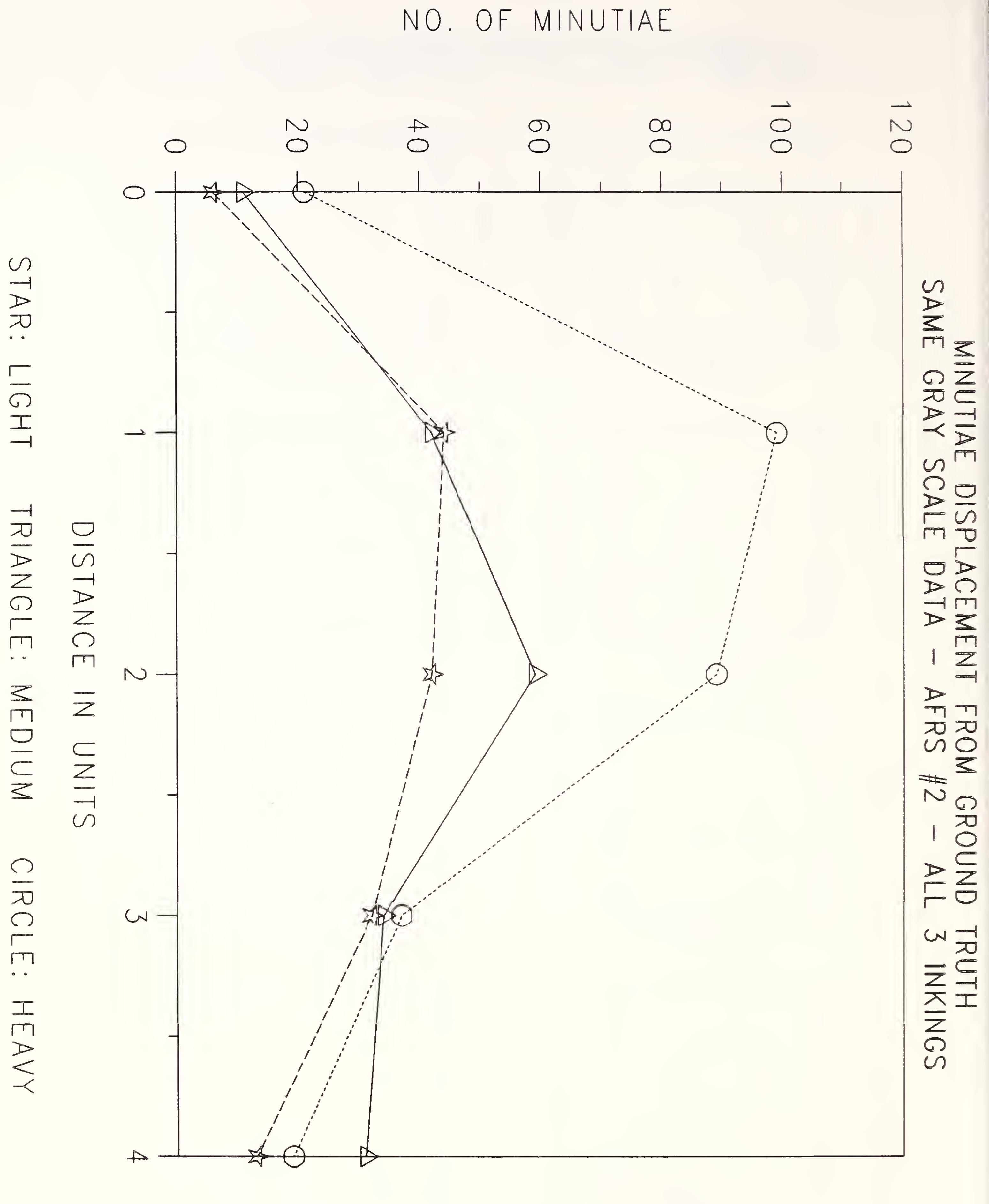

NO. OF MINUTIAE
0
20
40
60
80
100
120
MINUTIAE DISPLACEMENT FROM GROUND TRUTH
SAME GRAY SCALE DATA - AFRS #2 - ALL 3 INKINGS
0
1
2
3
4
DISTANCE IN UNITS
STAR: LIGHT TRIANGLE: MEDIUM CIRCLE: HEAVY

Figure 10

CUM. % BY NO. OF DETECTIONS

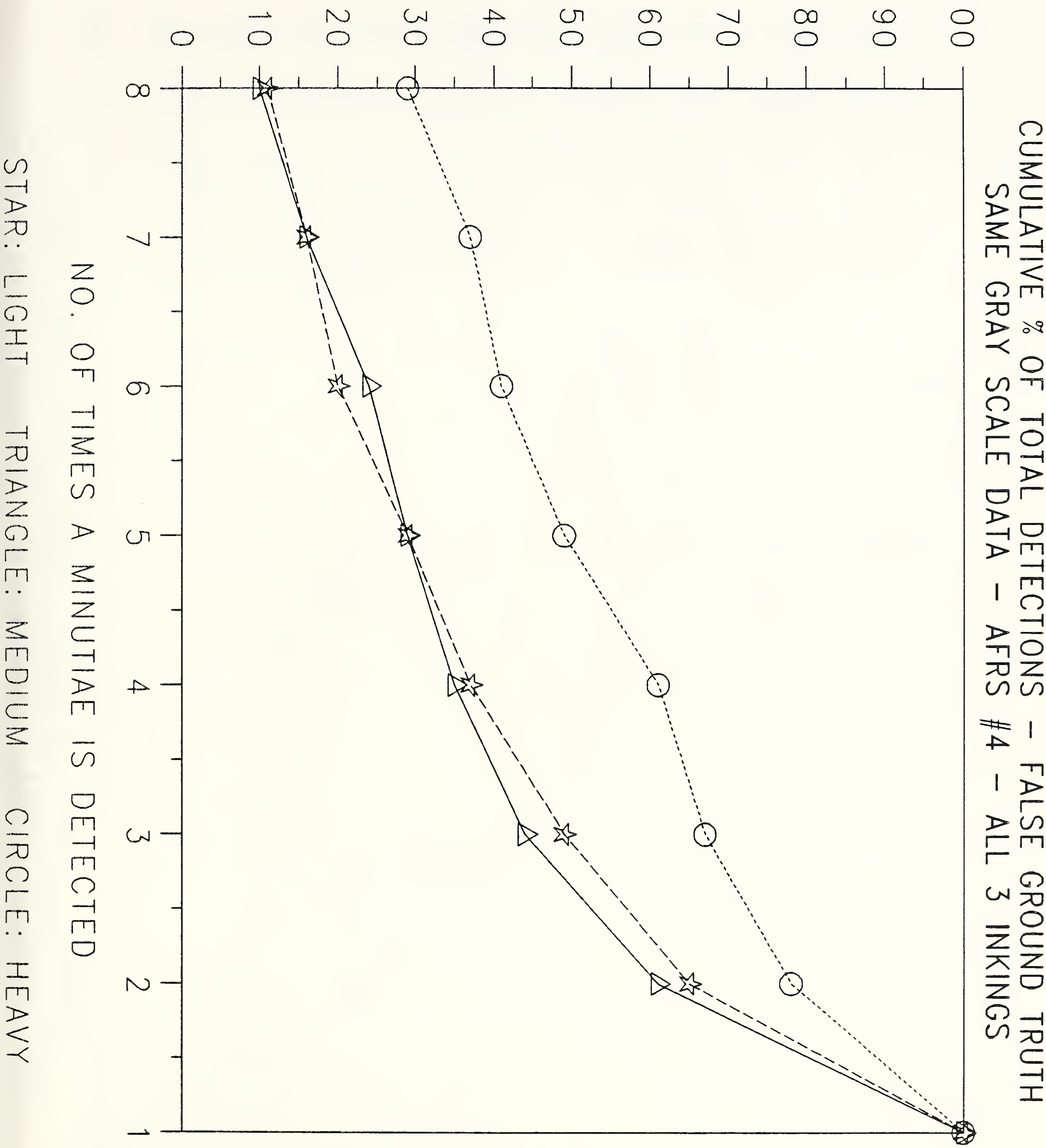



Figure 11

CUM. % BY NO. OF DETECTIONS

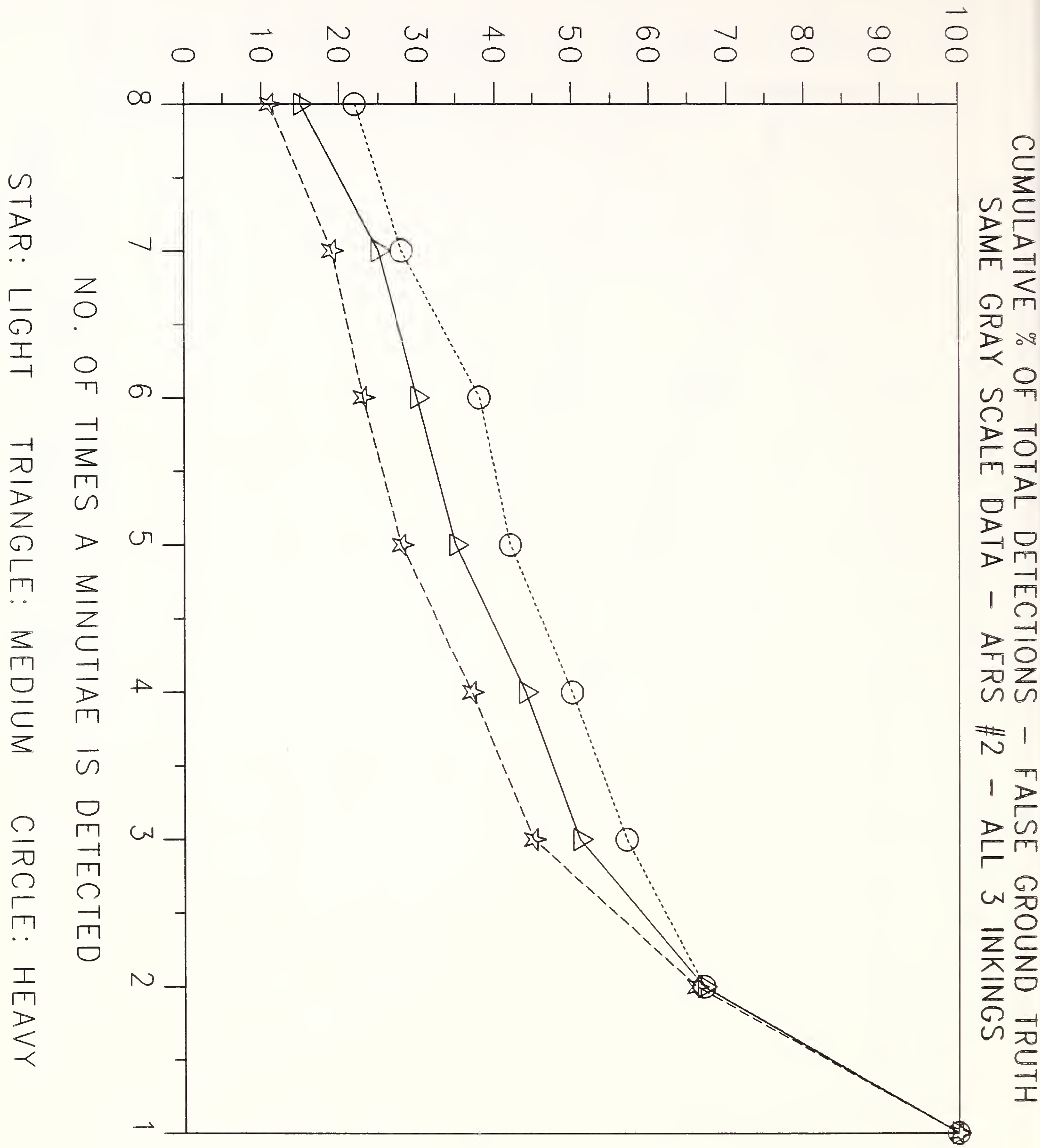



Figure 12

MINUTIAE DISPLACEMENT FROM PASS #1
SAME GRAY SCALE DATA – AFRS #4 – LIGHT INKING

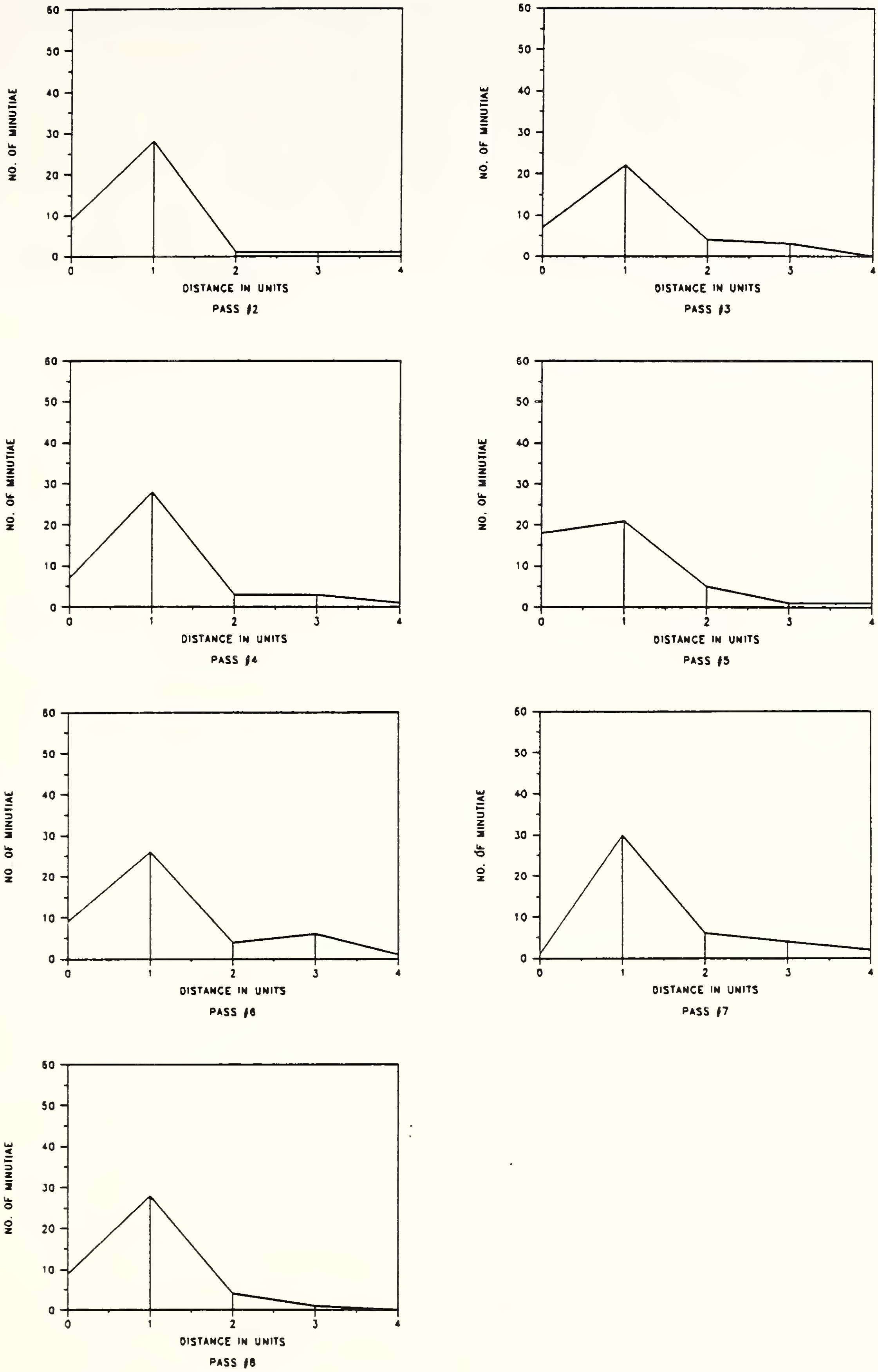



Figure 13

# MINUTIAE DISPLACEMENT FROM PASS #1
# SAME GRAY SCALE DATA – AFRS #4 – MEDIUM INKING

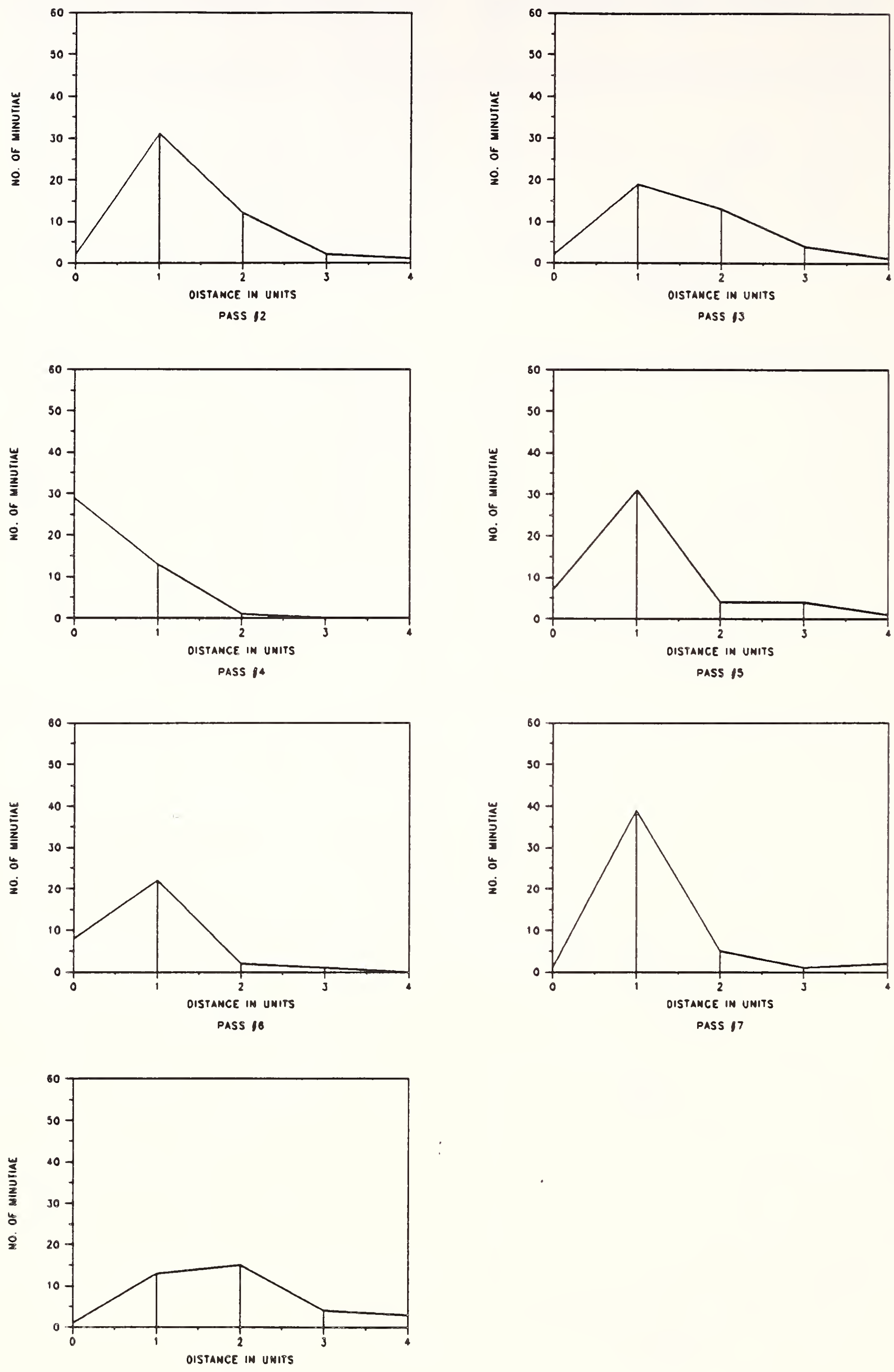



Figure 14

## MINUTIAE DISPLACEMENT FROM PASS #1
## SAME GRAY SCALE DATA – AFRS #4 – HEAVY INKING

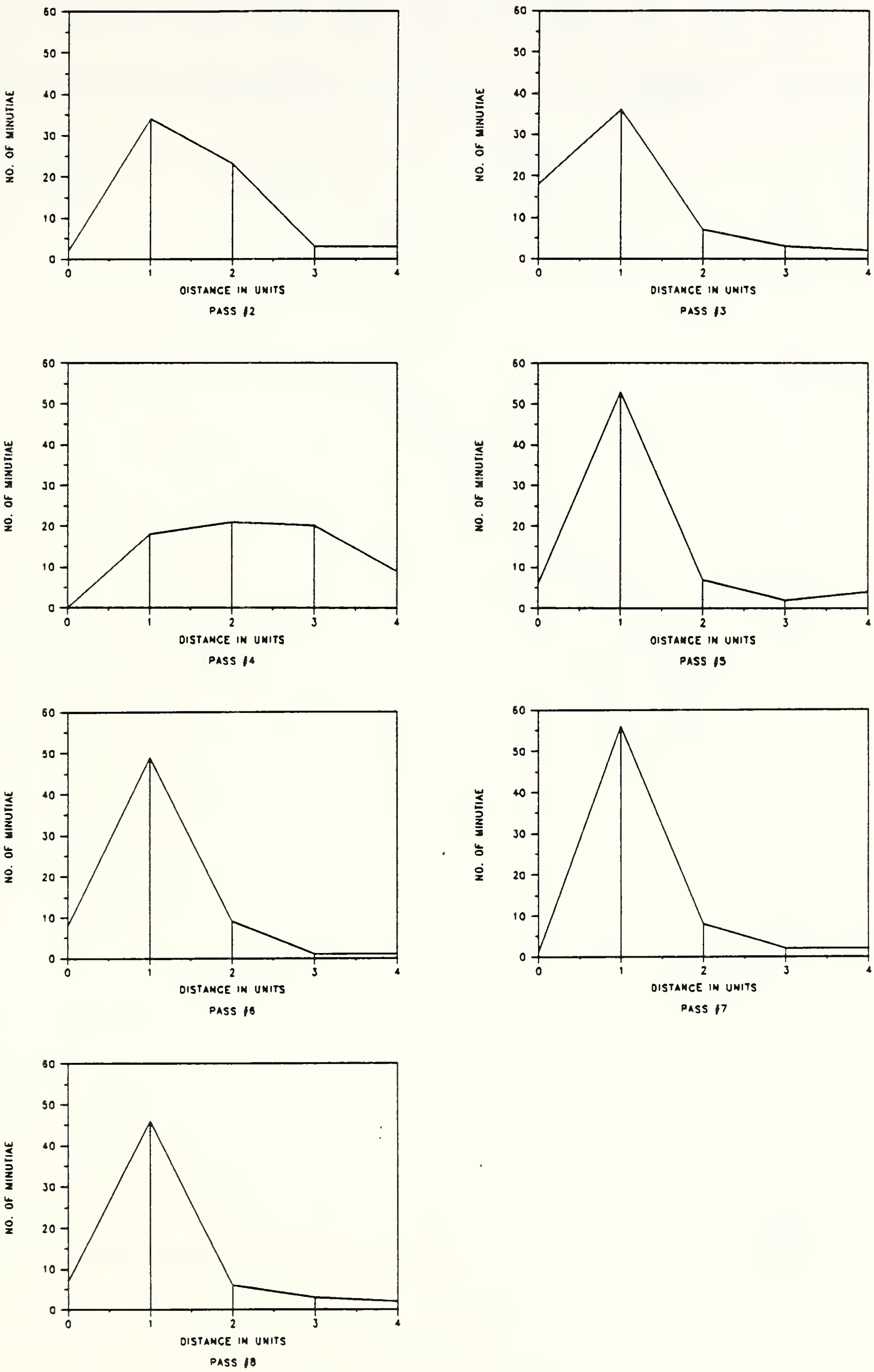



Figure 15

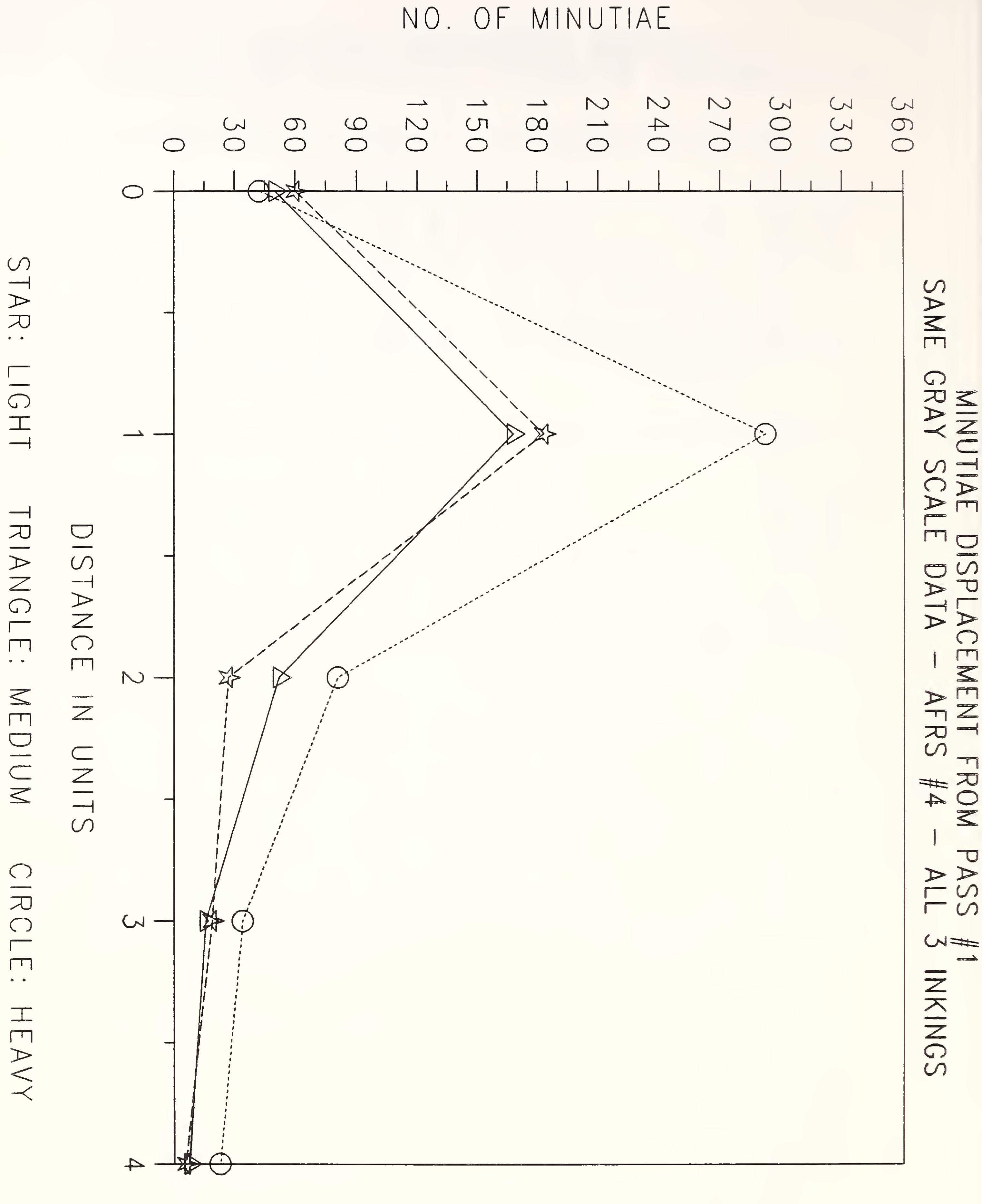
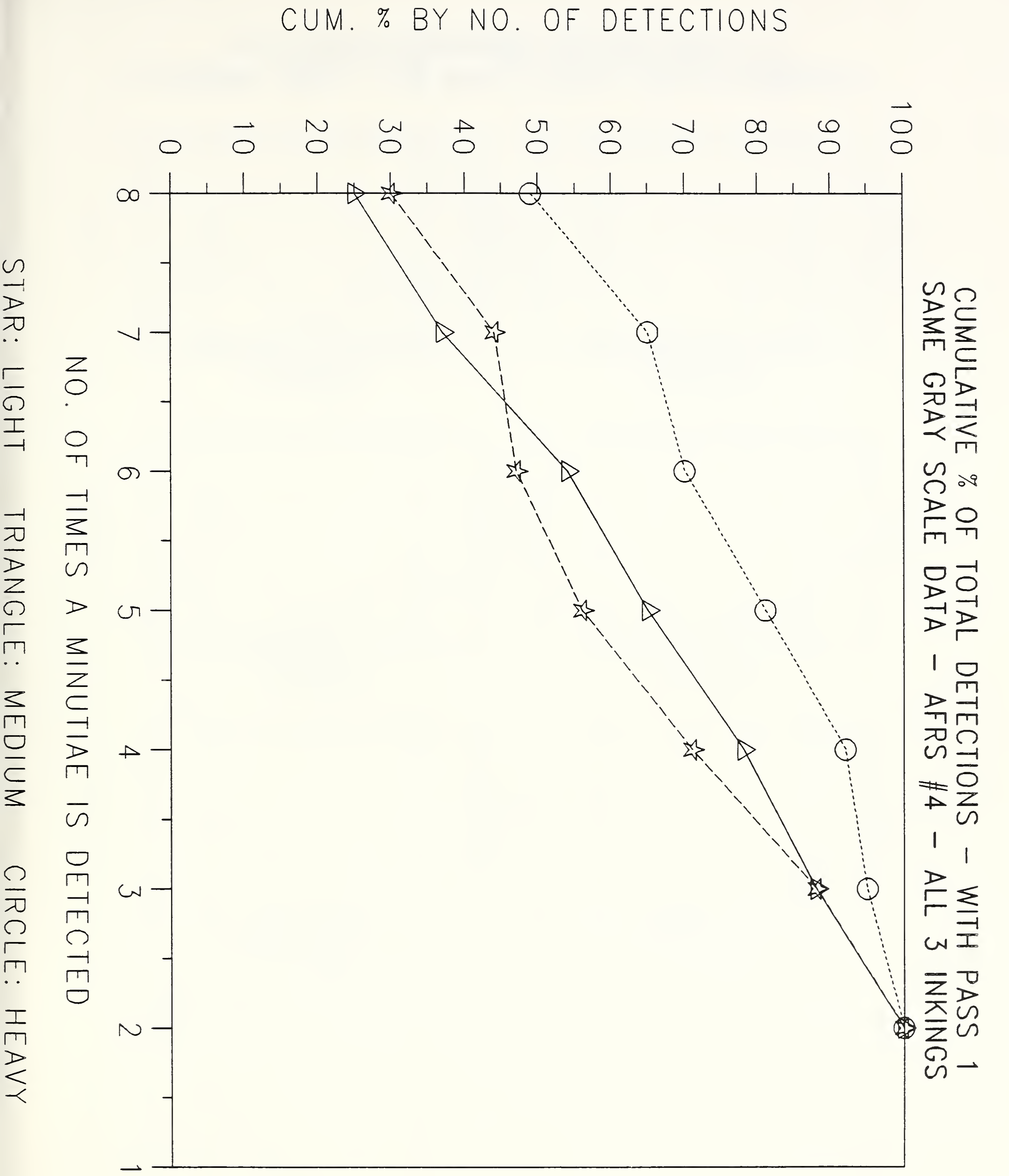

MINUTIAE DISPLACEMENT FROM PASS #1
SAME GRAY SCALE DATA - AFRS #4 - ALL 3 INKINGS
NO. OF MINUTIAE
0
30
60
90
120
150
180
210
240
270
300
330
360
0
1
2
3
4
DISTANCE IN UNITS
STAR: LIGHT TRIANGLE: MEDIUM CIRCLE: HEAVY

Figure 16



CUMULATIVE % OF TOTAL DETECTIONS - WITH PASS 1
SAME GRAY SCALE DATA - AFRS #4 - ALL 3 INKINGS
CUM. % BY NO. OF DETECTIONS
0
10
20
30
40
50
60
70
80
90
100
8
7
6
5
4
3
2
1
NO. OF TIMES A MINUTIAE IS DETECTED
STAR: LIGHT TRIANGLE: MEDIUM CIRCLE: HEAVY


Figure 17

MINUTIAE DISPLACEMENT FROM PASS #1
SAME GRAY SCALE DATA – AFRS #2 – LIGHT INKING

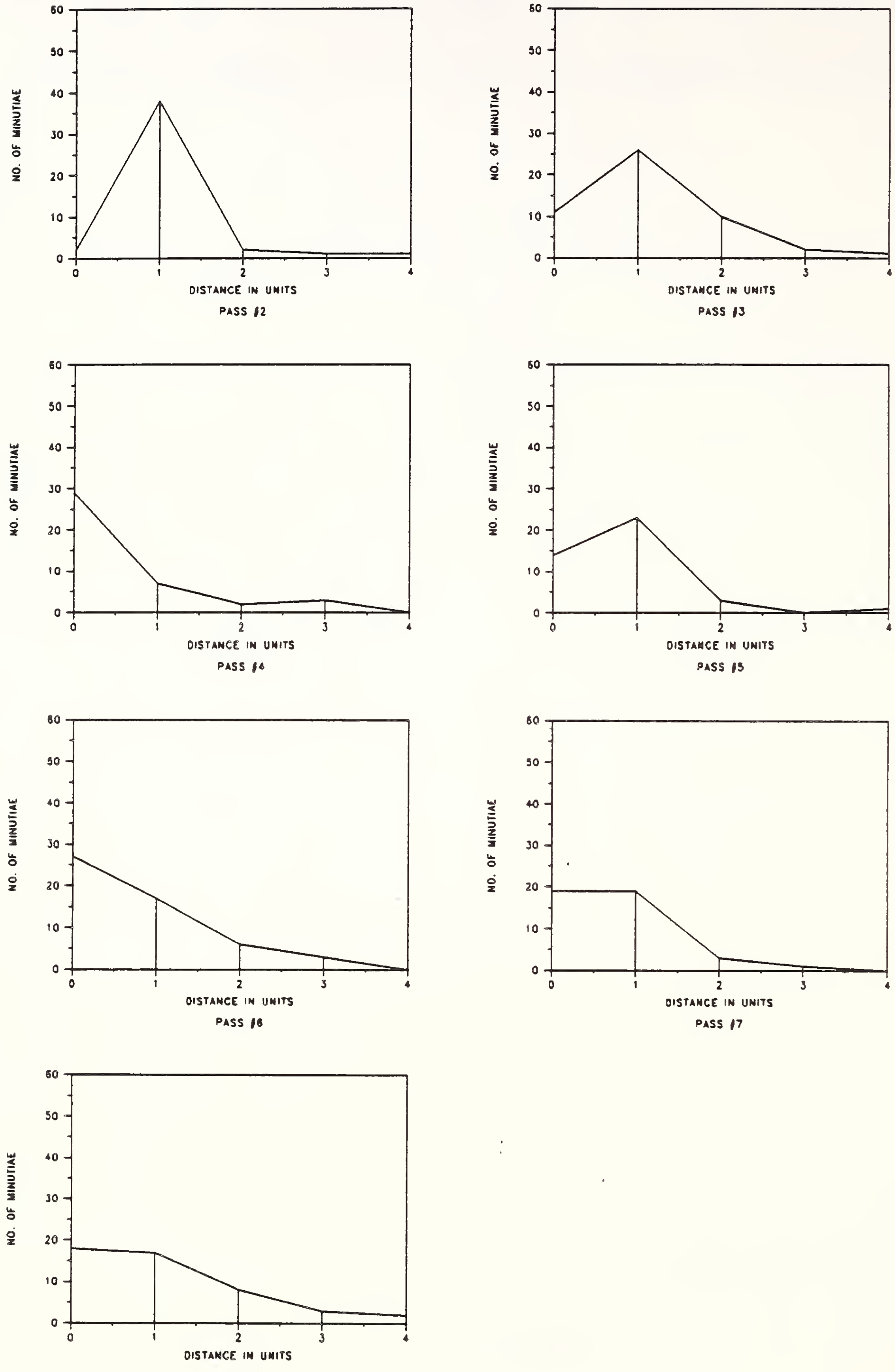



Figure 18

## MINUTIAE DISPLACEMENT FROM PASS #1
## SAME GRAY SCALE DATA – AFRS #2 – MEDIUM INKING

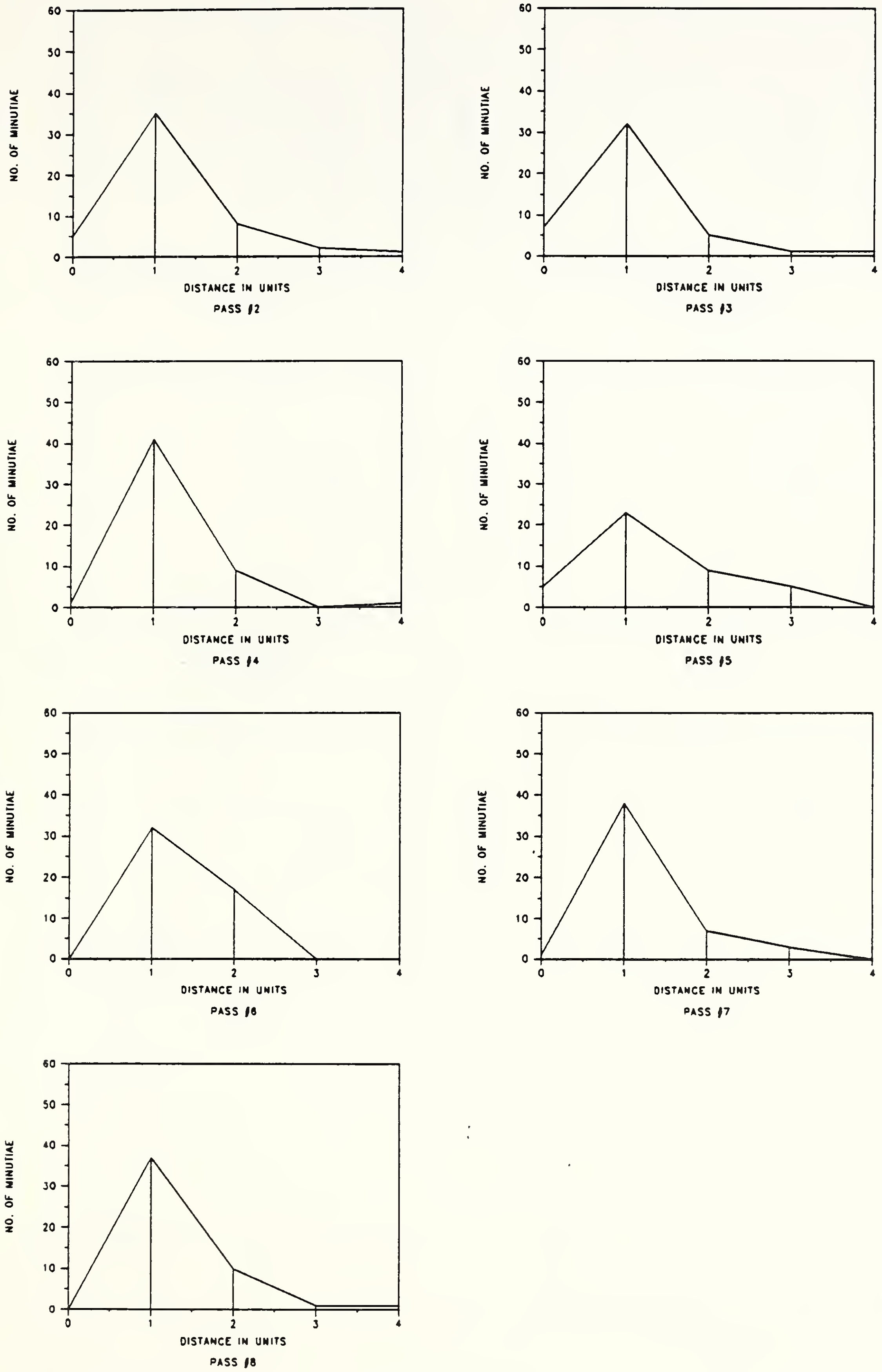



Figure 19

MINUTIAE DISPLACEMENT FROM PASS #1
SAME GRAY SCALE DATA – AFRS #2 – HEAVY INKING

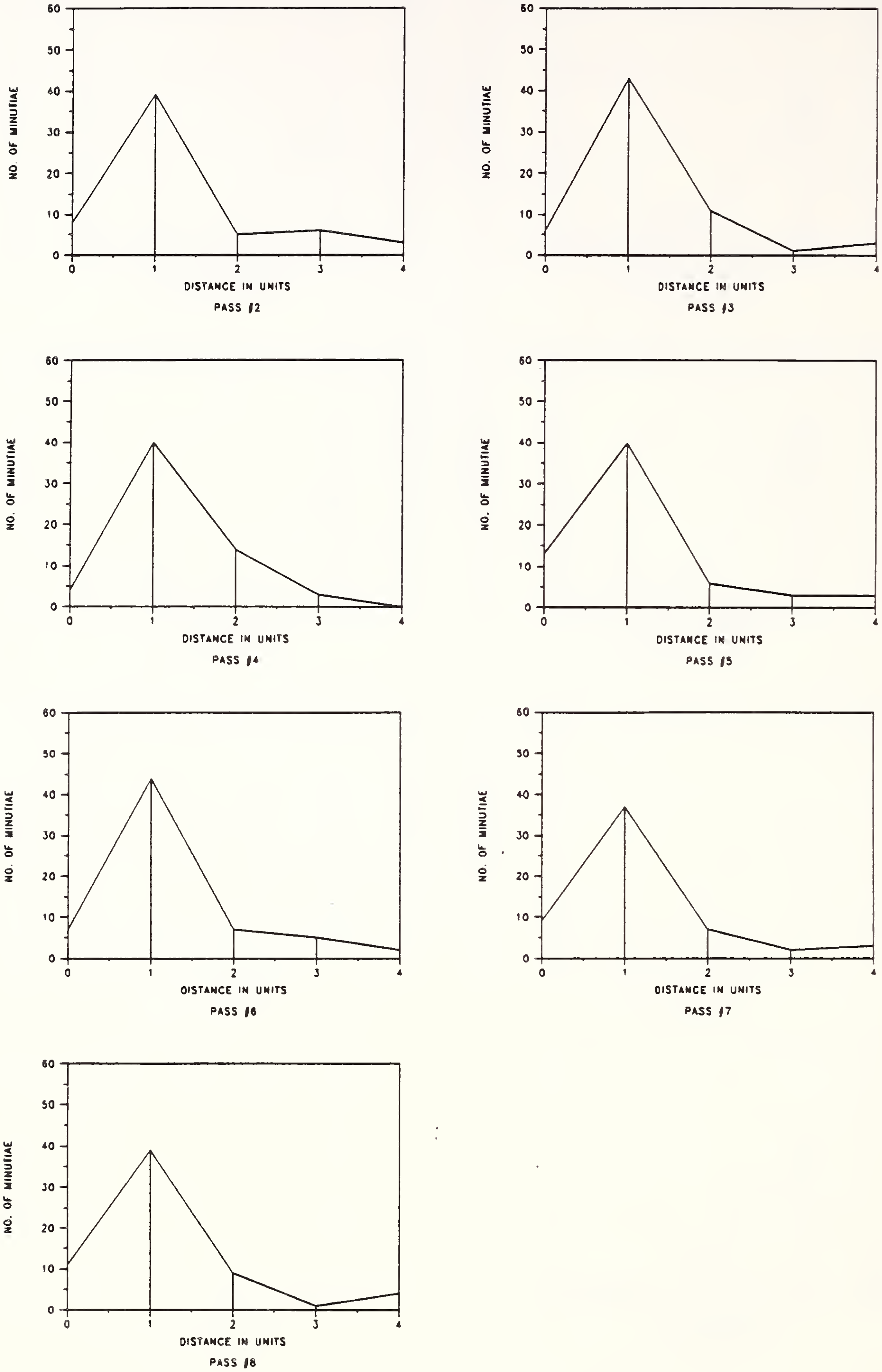



Figure 20

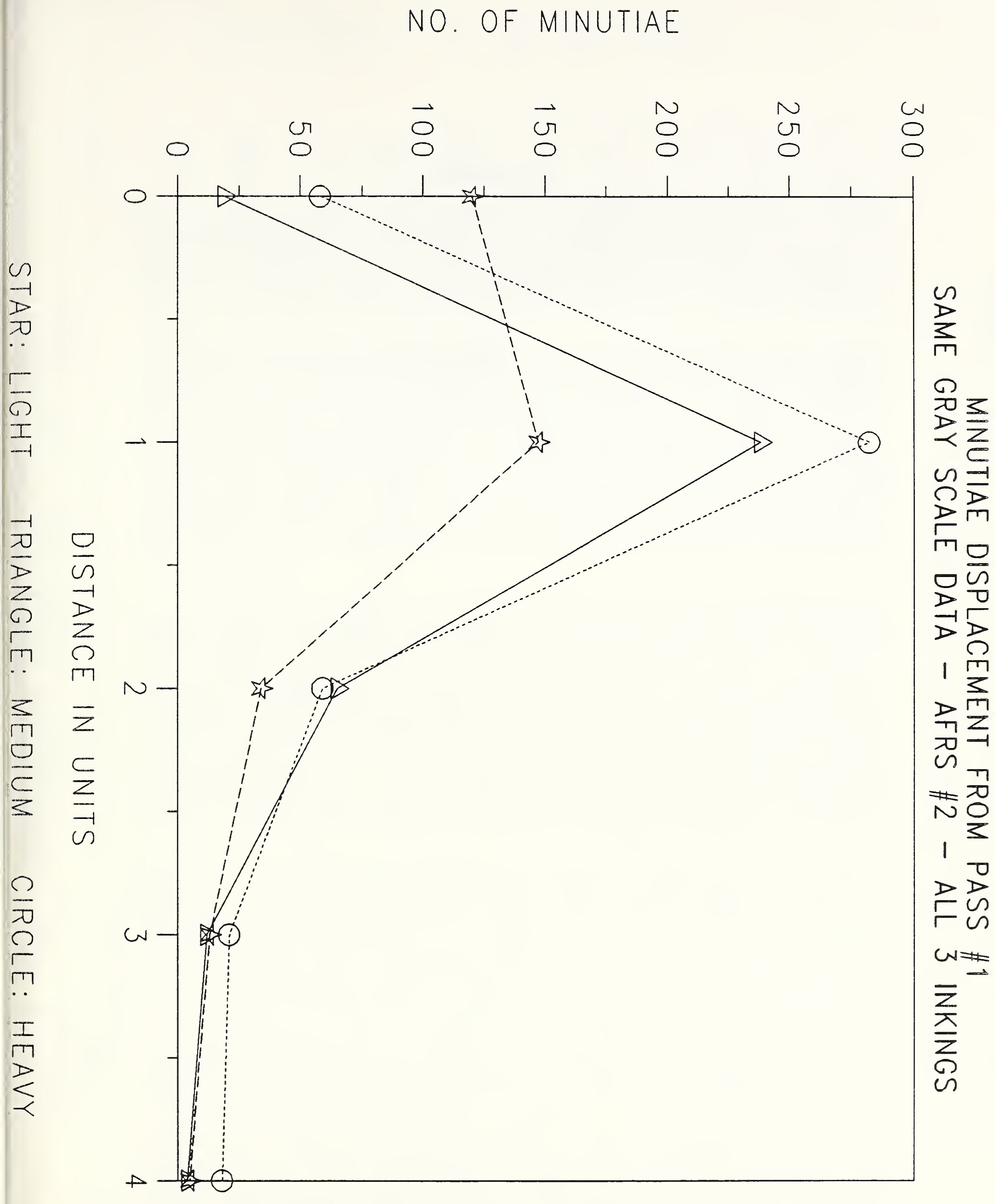

MINUTIAE DISPLACEMENT FROM PASS #1
SAME GRAY SCALE DATA - AFRS #2 - ALL 3 INKINGS
NO. OF MINUTIAE
0
50
100
150
200
250
300
DISTANCE IN UNITS
0
1
2
3
4
STAR: LIGHT TRIANGLE: MEDIUM CIRCLE: HEAVY

Figure 21

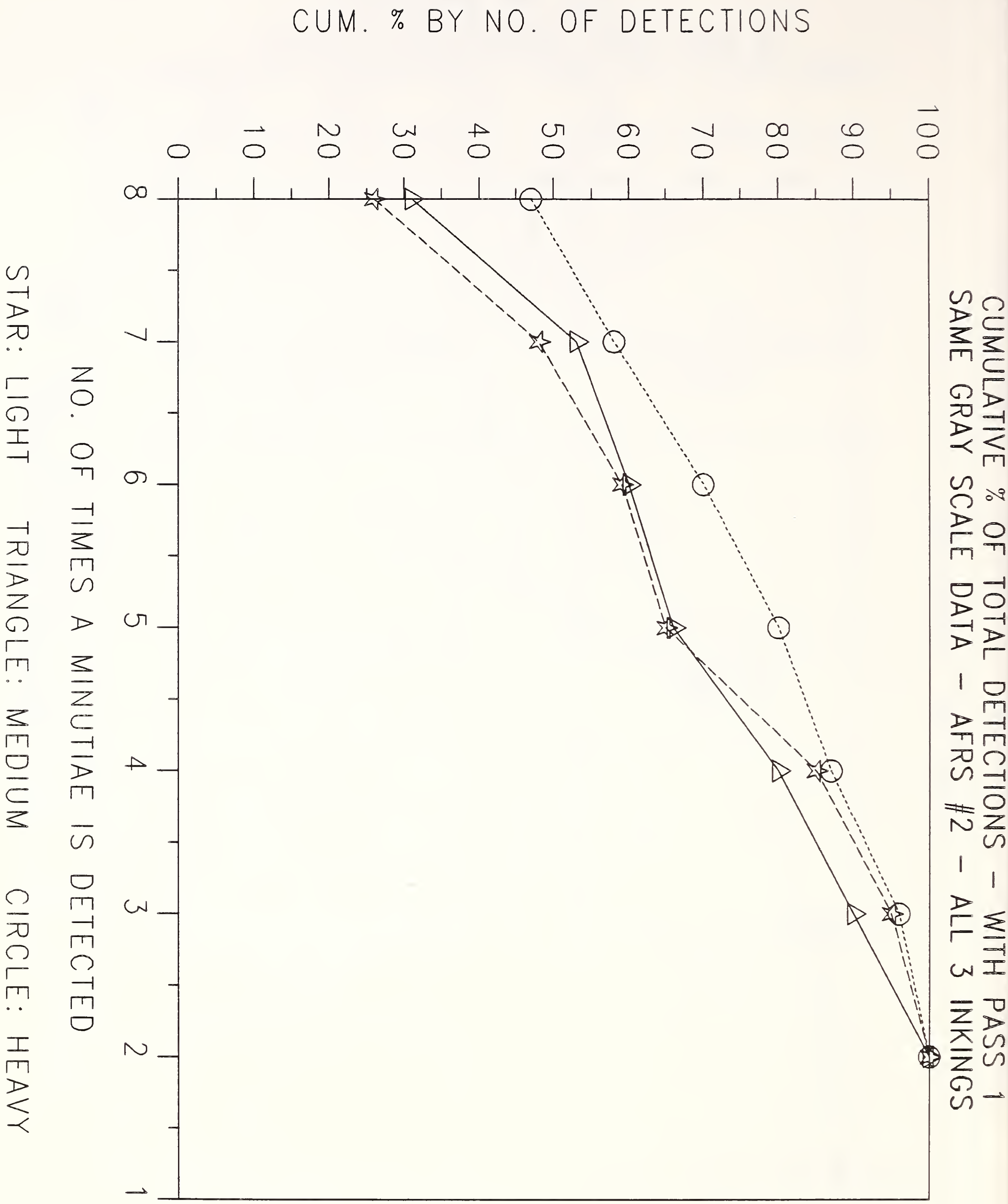

CUMULATIVE % OF TOTAL DETECTIONS - WITH PASS 1
SAME GRAY SCALE DATA - AFRS #2 - ALL 3 INKINGS
CUM. % BY NO. OF DETECTIONS
0
10
20
30
40
50
60
70
80
90
100
1
2
3
4
5
6
7
8
NO. OF TIMES A MINUTIAE IS DETECTED
STAR: LIGHT TRIANGLE: MEDIUM CIRCLE: HEAVY

Figure 22

# APPENDIX

This appendix contains a verbal description, flow diagrams, and the entire code for the program FINDTRANS and its subroutines. The program and subroutines are written in FORTRAN-77. Only two of the subroutines, FBIOPEN and FBIREAD are machine specific and only run on a VAX/VMS system, while the remaining subroutines and program should be portable.

The program FINDTRANS calculates the "best fit" transformation values for delta X, delta Y and delta THETA to be used in matching a pair of fingerprints. Additionally, the program evaluates the accuracy of a match performed using these transformation values. A "best fit" may be defined as the orientation in translation and rotation of a comparison fingerprint with a mating base fingerprint, such that the positions of the majority of minutiae from the comparison fingerprint are relatively close to those of the base fingerprint as evidenced by highest matching score. A "best fit" transformation may be one of several possible "best fit" transformations because of the many combinations of transformations that can be generated with equal scores. In this situation the first "best fit" transformation found is the transformation used.

As input, FINDTRANS requires the name of the file containing the base fingerprint (the one which will not be transformed) and the quantity of other prints to be compared against this base fingerprint. For each comparison print, the minutiae positions are read in and the initial "best fit" is assumed at the transformation values of 0 in X, 0 in Y, 0 in rotation. FINDTRANS then issues four calls to the subroutine BESTFIT, which calculates new transformation values. Each time BESTFIT is called, a constant theta difference limit of 12 degrees is used. However, the straight line distance tolerance decreases with each call to BESTFIT. The values that are used for the four calls are 30, 15, 8, and 4 respectively. Only four calls of BESTFIT are used as the "best fit" values after the fourth iteration show very little sign of change. When FINDTRANS is done, the "best fit" transformation is known and diagnostic output of the match performed using this transformation is generated.

BESTFIT is the primary subroutine called by FINDTRANS, as it performs the majority of the processing. It initially saves the assumed "best fit" values that were passed to it and asks the user to input the beginning and ending rotational range with values stated in degrees. This range is used to test for the best transformation rotation value. This will be a value between the assumed "best fit" rotation plus the beginning value, increasing in increments of one degree, to the assumed "best fit" rotation plus the ending value. The beginning and ending values may be negative but the beginning value must always be less than the ending value. The comparison print is transformed using the present rotation value of the programming loop. After the transformation, the two prints are compared to find possible matches in minutiae positions. Any minutiae position in the base print that has a minutiae position in the comparison print located within the straight line distance tolerance and with the difference in theta values less than or equal to 12 degrees is considered to be a possible match. These possible matches are saved in a table called the S-table. Each entry in the S-table is given a score which rates the closeness of the match between the two minutiae. Using a scoring strategy, a score is generated for the entire transformation. If this is the first transformation then the score is assumed to be the highest score and is saved with the transformation values that generated the score. If this is not the first transformation then the score is compared against the saved score. If the new score is higher than the saved score, then the values and score are saved as the new "best fit" and high score. When the last rotation value of the loop has been processed, the highest score and the corresponding transformation values are saved. This rotation value is used to retransform the comparison print and then recreate the S-table for this transformation. The data in the S-table is used to determine where the concentration of S-table entries are located. This point of concentration is the negative offset needed for the "best fit" X and Y values. If the point of concentration is at position (3,-1) then the new "best fit" values are the previous "best fit" X value minus 3, and the previous "best fit" Y value plus 1. BESTFIT returns to FINDTRANS, its caller, the "best fit" values it has generated and the score of the match generated with these transformation values.

Several scoring strategies have been tested in an effort to evaluate the accuracy of a match between a base fingerprint and a transformed comparison fingerprint. The strategies that were tested include highest S-table entry score, average of the S-table entry's scores and total of the S-table entry's scores. The highest S-table score was found to be an effective method until the prints had at least one minutiae position match exactly on both prints. This caused the highest possible score to be generated. Many transformation values caused this situation. Due to this, the highest score method could not accurately choose a "best fit". To solve this problem more than one S-table entry value had to be used so the total and average methods were tested. Averaging the S-table on some matches cause a selection of two or more "best fit" transformations. With examination, these were not found to be very accurate. By using the total method the best results were obtained. The total method summed the S-table scores to generate the overall score. This method was used in BESTFIT to create a match score.

Example of results using all three methods

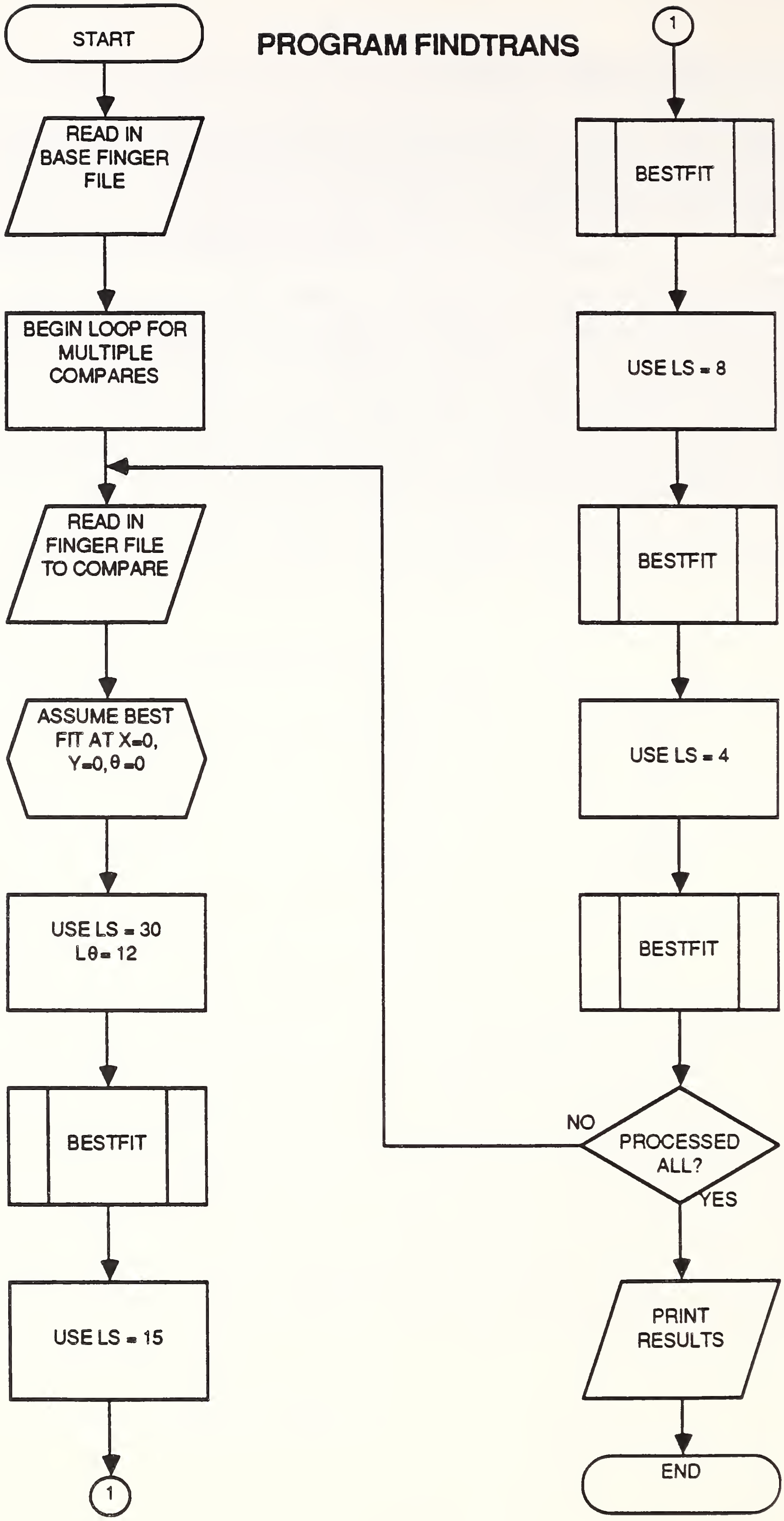
| Entry # | S-table 1 scores | S-table 2 scores | S-table 3 scores |
|---|---|---|---|
| 1 | 100 | 120 | 180 |
| 2 | 100 | 110 | 100 |
| 3 | 50 | 70 | 80 |
| 4 | 60 | 80 | 40 |
| 5 | 150 | 90 | 60 |
| 6 | 20 | No entry | No entry |
| Total method | 480 | 470 | 460 |
| Average method | 80 | 94 | 92 |
| Highest method | 150 | 120 | 180 |

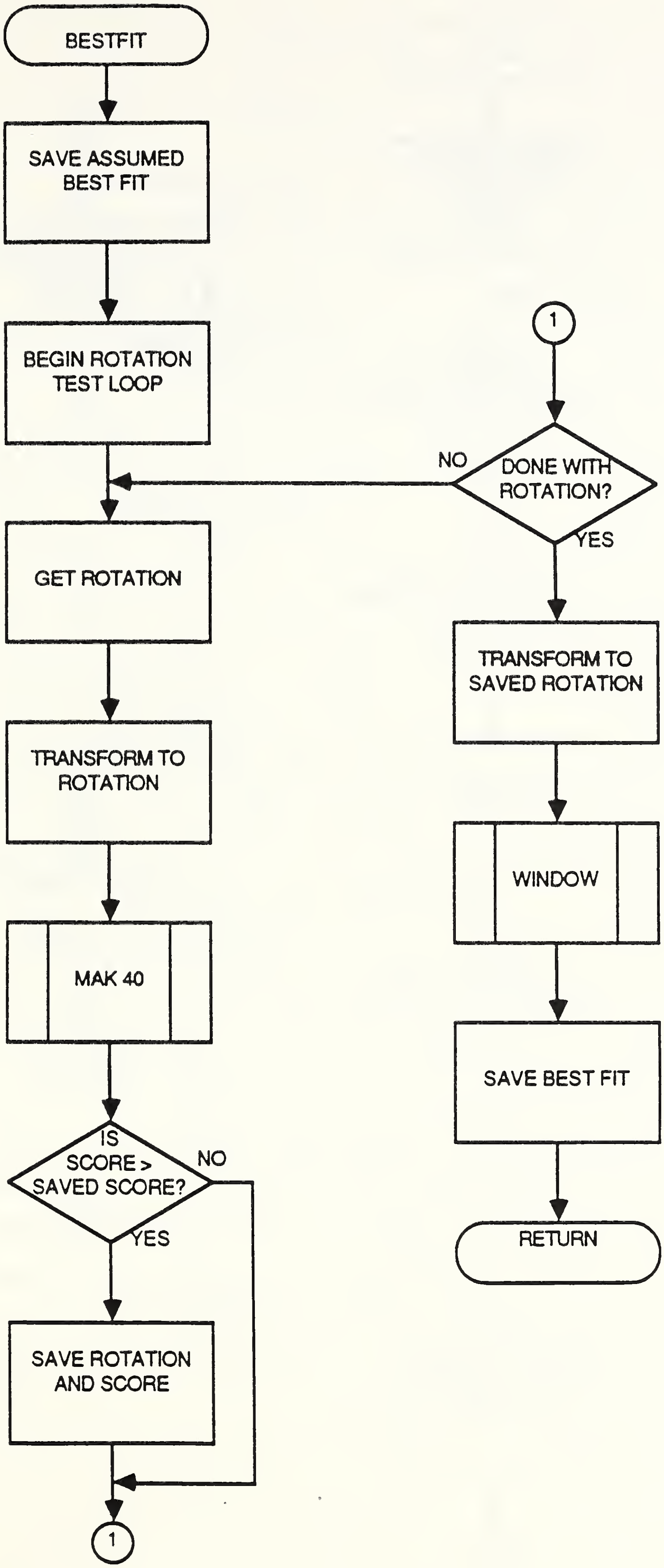
To create the S-table for a comparison, the routine MAK40 is called. This routine creates a table of pairs of minutiae whose delta X, delta Y and delta theta differences are within the tolerance levels. The tolerance limits are linear distance, usually no more than four units, and angular difference, usually no more than 12 degrees. This table keeps track of the minutiae identity numbers for the transformed print and base print, the differences in X, and Y, the delta theta of these two minutiae, and the score for the entry. When an entry is added to the list it is assigned a score of zero. Once all the entries have been created, MAK40 generates a score for each one. To do this, the linear distance between these two entries is calculated using their X and Y difference values [1]. If the distance is not within the straight line distance tolerance level or is greater than five, it has no effect on the score. If the distance is within the straight line distance tolerance and less than five, the score is adjusted by adding five minus the distance to the score [2]. When all the entry scores have been generated, MAK40 returns to its caller the completed S-table and the number of entries in the S-table.

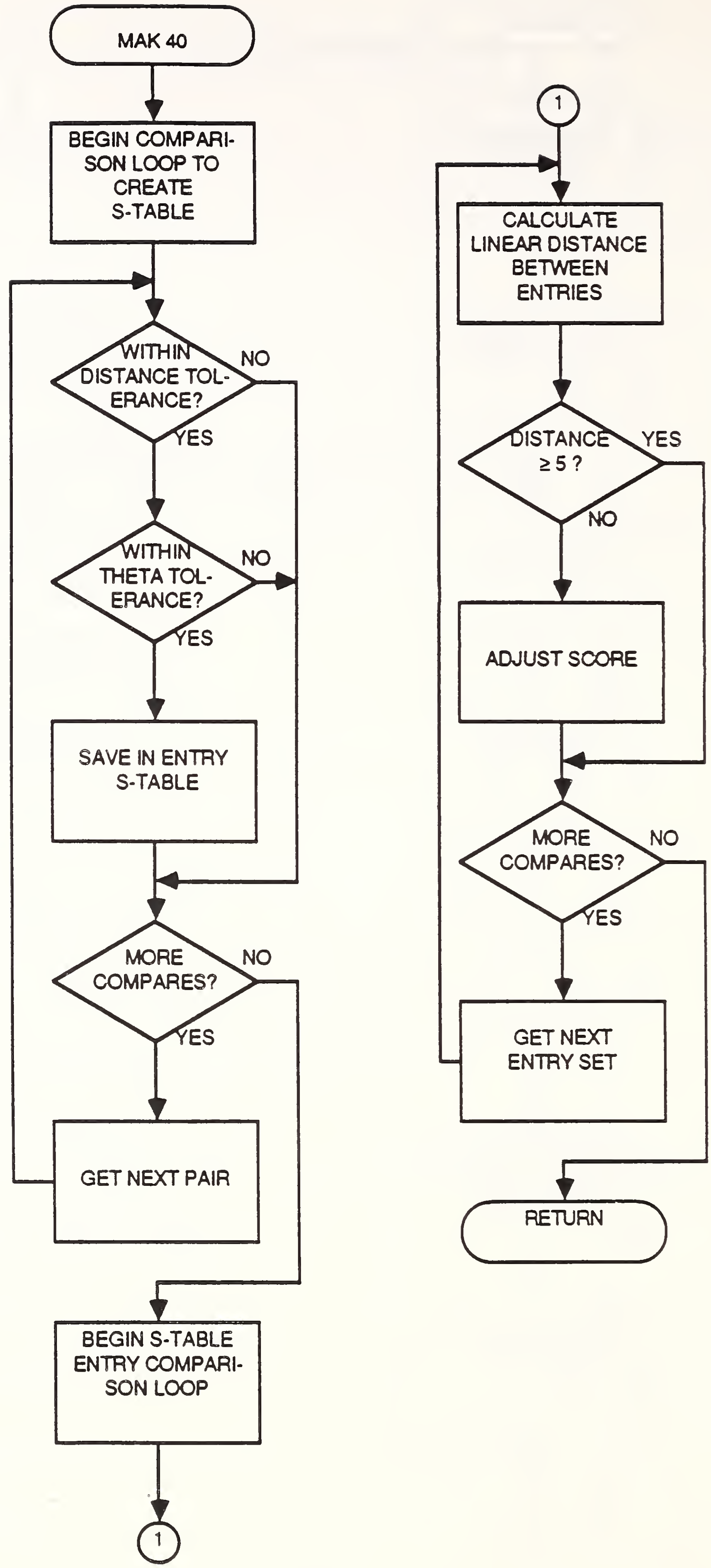
$$\text{Distance} = \sqrt{[(\text{delta } X_2 - \text{delta } X_1)^2 + (\text{delta } Y_2 - \text{delta } Y_1)^2]} \qquad [1]$$

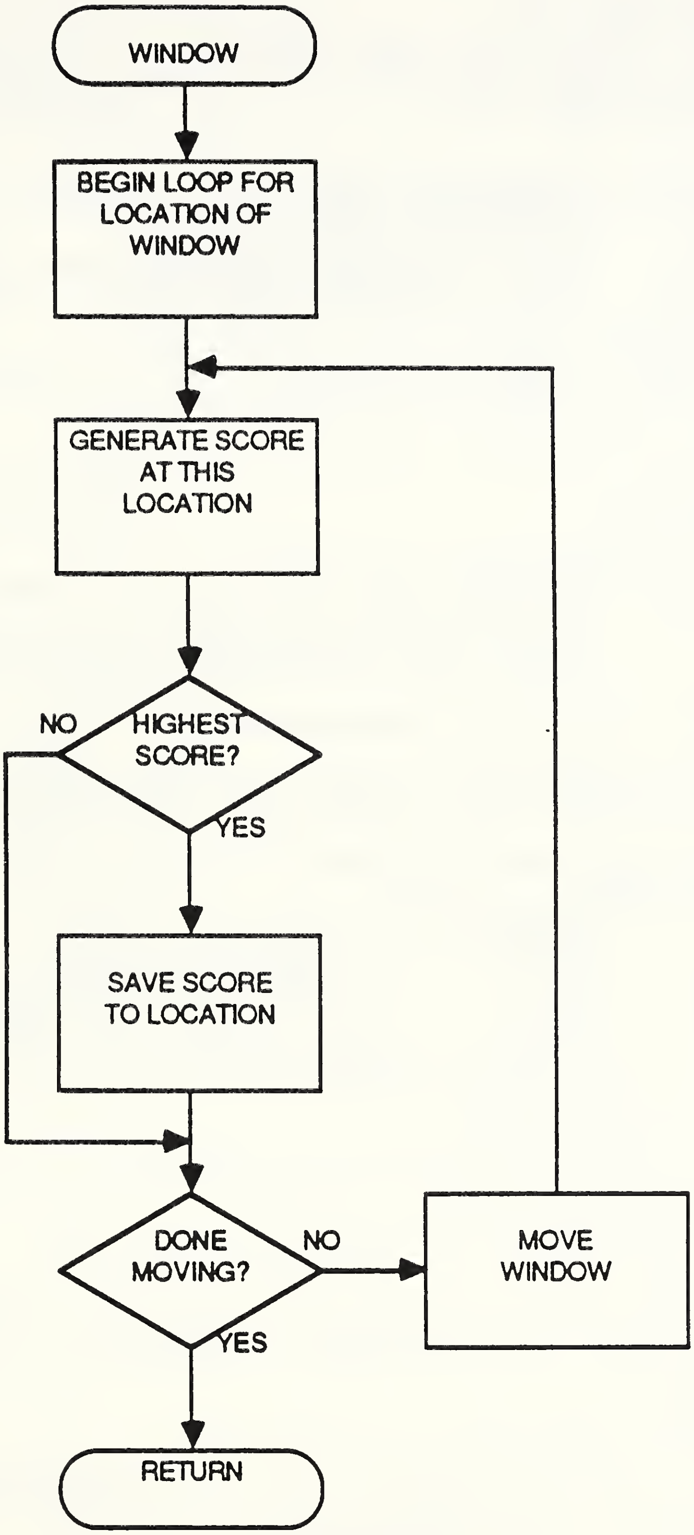
$$\text{Score} = \text{Score} + (5 - \text{Distance}) \qquad [2]$$

To determine the "best fit" X and Y values, the routine WINDOW is called. WINDOW generates values to be placed in a table indexed by delta X and delta Y values from the S-table. The S-table entries with the same delta X and delta Y values are counted and that number is placed in the proper delta X, delta Y location. This routine moves a window over the delta X, delta Y table and generates three values for each position in that table. These values are for the specified window size, a window size that is two increments smaller, and a window size that is four increments smaller. A starting window size must always be an odd number and be greater than or equal to 5 units. This forces the center of the window to be at an integer coordinate position, not located between coordinates. If the specified size is 7x7 then the other two values are for windows 5x5 and 3x3. WINDOW returns the location of the highest score for a large window. If two or more locations tie, then the delta X, delta Y position of the one out of this group with the highest medium window score is returned. If a tie still exists, the highest score from the smallest window has its location returned. Finally, if a tie exists in all three window sizes, the average of delta X, delta Y locations that tie in the smallest window are the values that are returned.


START
PROGRAM FINDTRANS
READ IN BASE FINGER FILE
BEGIN LOOP FOR MULTIPLE COMPARES
READ IN FINGER FILE TO COMPARE
ASSUME BEST FIT AT X=0, Y=0, θ=0
USE LS = 30 Lθ= 12
BESTFIT
USE LS = 15
1
1
BESTFIT
USE LS = 8
BESTFIT
USE LS = 4
BESTFIT
NO
PROCESSED ALL?
YES
PRINT RESULTS
END


BESTFIT
SAVE ASSUMED BEST FIT
BEGIN ROTATION TEST LOOP
GET ROTATION
TRANSFORM TO ROTATION
MAK 40
IS SCORE > SAVED SCORE?
NO
YES
SAVE ROTATION AND SCORE
1
1
DONE WITH ROTATION?
NO
YES
TRANSFORM TO SAVED ROTATION
WINDOW
SAVE BEST FIT
RETURN


MAK 40
BEGIN COMPARISON LOOP TO CREATE S-TABLE
WITHIN DISTANCE TOLERANCE?
NO
YES
WITHIN THETA TOLERANCE?
NO
YES
SAVE IN ENTRY S-TABLE
MORE COMPARES?
NO
YES
GET NEXT PAIR
BEGIN S-TABLE ENTRY COMPARISON LOOP
1
1
CALCULATE LINEAR DISTANCE BETWEEN ENTRIES
DISTANCE ≥ 5 ?
YES
NO
ADJUST SCORE
MORE COMPARES?
NO
YES
GET NEXT ENTRY SET
RETURN



WINDOW
BEGIN LOOP FOR LOCATION OF WINDOW
GENERATE SCORE AT THIS LOCATION
HIGHEST SCORE?
NO
YES
SAVE SCORE TO LOCATION
DONE MOVING?
NO
YES
MOVE WINDOW
RETURN

```
          PROGRAM FINDTRANS
    C
    C **  CALCULATES POSSIBLE TRANSFORMATION VALUES
    C
    C **  BY R. ALLEN WILKINSON
    C
    C **  INITALIZE VARIABLES
    C
          IMPLICIT INTEGER (A-Z)
          INTEGER UNIT,FIN
          INTEGER NUM,I,J,K,L
          INTEGER ISTAB(500,6),ISCNT,CHART(200,10)
          INTEGER BEST(4),TOTAL(25),TABLE(10,500,6),TSCNT(10),GROUP(20)
          INTEGER*2 B(250,3,2)
          CHARACTER TYPE
    C
    C **  NUM          NUMBER OF FINGERS TO BE COMPARED
    C **  ISTAB(K,6)   S TABLE ENTRIES
    C **  ISCNT        NUMBER OF ENTRIES IN S TABLE
    C **  CHART(K,L)   ARRAY FOR MINUTIAE MATCHED CHART
    C **  TOTAL(I)     TOTAL OF DISTANCE CHART OVER I FINGERS
    C **  GROUP(I)     MINUTIAE MATCHED SUMMARY ARRAY
    C
          DATA UNIT/7/
          DATA TOTAL/0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0/
    C
          INCLUDE '[WILKINSON.LIB]COM004.FOR/LIST'
  1 C
  1       INTEGER*2 A
  1       INTEGER IMX
  1 C
  1       COMMON /BLK004/A(250,3,2),IMX(2)
  1 C **  A(250,3,2)   MINUTIAE ARRAY(MINUTIAE #,DATA TYPE,PRINT TYPE)
  1 C **               DATA TYPE- X=1,Y=2,THETA=3
  1 C **               PRINT TYPE- SEARCH PRINT=1,FILE PRINT=2
  1 C **  IMX(2)       NUMBER OF MINUTIAE(PRINT TYPE)
  1 C
          INCLUDE '[WILKINSON.LIB]COM007.FOR/LIST'
  1 C
  1       COMMON /BLK007/LN(51),LT(51),LS,LTHETA,MTM2,IPRMT,UPPER(2),
  1      * ISBZ(2),ISRZ(2),ISUNIT,SCORES(40),IPRM2,IJUMP
  1 C
          INCLUDE '[WILKINSON.LIB]COM012.FOR/LIST'
  1       LOGICAL DIAG
  1       COMMON /BLK012/DIAG(100)
  1 C **  FLAG         SUBROUTINE    DIAGNOSTIC
  1 C **  DIAG(1)      BESTFIT       S TABLE INFO PRINTED
  1 C **  DIAG(2)      BESTFIT       S TABLE DX,DY VALUES PLOTTED
  1 C **  DIAG(3)      BESTFIT       DISTANCE SUMMARIES PRINTED
  1 C **  DIAG(4)      BESTFIT       USING SCORE TOTALS NOT AVERAGE
          INCLUDE '[WILKINSON.LIB]COM013.FOR/LIST'
  1 C
  1       CHARACTER TMPFILE*40
  1       INTEGER DISTAN
  1 C
  1       COMMON /BLK013/TMPFILE,DISTAN(25)
  1 C **  USED IN OPENFBI
  1 C **   TMPFILE    ORIGINAL FILE NAME UPON OPENNING
  1 C **  USED IN BESTFIT
  1 C **   DISTAN(I)  DISTANCE FROM CENTER COUNTS ON A FINGER (I=DISTANCE)
  1 C
          INCLUDE '[WILKINSON.LIB]COMDATA.FOR/LIST'
  1 C
  1 C *  DATA ARRAYS  *
  1       INTEGER*2 FINGER,QUALITY,MINCNT,XYT
  1 C
  1       COMMON /BLKDATA/FINGER(10),QUALITY(10),MINCNT(10),
  1      *                XYT(250,3,10)
  1 C **  USED IN NEW FBI DATA ORGANIZATION
  1 C **    FINGER(I)    Ith FINGER'S ACTUAL NUMBER
  1 C **    QUALITY(I)   Ith FINGER'S DATA QUALITY
  1 C **    MINCNT(I)    Ith FINGER'S NUMBER OF MINUTIAE
  1 C **    XYT(J,K,I)   Ith FINGER'S Jth MINUTIAE DATA
  1 C **                   WHERE K=
  1 C **                    1 FOR X DATA VALUE 0-255
  1 C **                    2 FOR Y DATA VALUE 0-255
  1 C **                    3 FOR THETA DATA VALUE 0-359
  1 C
          INCLUDE '[WILKINSON.LIB]COMIO.FOR/LIST'
  1 C
  1       STRUCTURE /FBIDAT/
  1         UNION
  1           MAP
```

```
1               CHARACTER*10 PCNUM
1               CHARACTER*40 CLASS
1               CHARACTER*9  CDATE
1               CHARACTER*8  CTIME
1               CHARACTER*3  FILL
1               INTEGER*2    CARD
1               INTEGER*2    OFFSET(10)
1               INTEGER*2    VALUES(7550)
1             END MAP
1             MAP
1               CHARACTER*15192  BUF
1             END MAP
1           END UNION
1         END STRUCTURE
1         RECORD /FBIDAT/INPUT
1 C
1         COMMON /BLKIO/INPUT
1 C **    USED IN NEW FBI I/O ROUTINES - FBIOPEN,FBIREAD,FBIWRITE
1 C **      PCNUM        PROCESS CONTROL NUMBER
1 C **      CLASS        CLASSIFICATION
1 C **      CDATE        FILE CREATION DATE
1 C **      CTIME        FILE CREATION TIME
1 C **      FILL         JUST USED TO FILL TO WORD BOUNDARY
1 C **      OFFSET(I)    OFFSET FROM PCN TO FINGER #I'S DATA
1 C **      VALUES(I)    ACTUAL MINUTIAE DATA ARRAY
1 C
  C
          DIAG(1)=.FALSE.
          DIAG(2)=.FALSE.
          DIAG(3)=.FALSE.
          DIAG(4)=.TRUE.
  C
  C **    OPEN FILES
  C
          WRITE(6,1100)
  1100    FORMAT(1X,'BASE FINGER')
          CALL FBIOPEN(UNIT)
  C
  C **    READ RECORD
  C
          CALL FBIREAD(UNIT)
          CLOSE(UNIT)
          CALL FINNUM(FIN)
          IMX(1)=MINCNT(FIN)       !SET NUMBER OF MINUTIAE
          DO 10 I=1,IMX(1)
            DO 15 J=1,3
              A(I,J,1)=XYT(I,J,FIN)
              B(I,J,1)=XYT(I,J,FIN)
  15        CONTINUE
  10      CONTINUE
  C
  C **    ENTER LOOP FOR ALL TRANSFORMING FINGERS
  C
          WRITE(6,1101)
  1101    FORMAT(1X,'ENTER NUMBER OF FINGERS TO COMPARE TO BASE')
          READ (5,'(I2)')NUM
          DO 20 L=1,NUM
  C
  C **    OPEN FILE
  C
            WRITE(6,1102)
  1102      FORMAT(1X,'FINGER TO TRANSFORM')
            CALL FBIOPEN(UNIT)
  C
  C **    READ RECORD
  C
            CALL FBIREAD(UNIT)
            CLOSE(UNIT)
            CALL FINNUM(FIN)
            BEST(1)=0
            BEST(2)=0
            BEST(3)=0
            IMX(2)=MINCNT(FIN)       !SET NUMBER OF MINUTIAE
            DO 30 J=1,IMX(2)
              DO 35 K=1,3
                A(J,K,2)=XYT(J,K,FIN)
                B(J,K,2)=XYT(J,K,FIN)
  35          CONTINUE
  30        CONTINUE
  C
  C **      DO FINE ALIGNMENT CHANGES TO FIND BEST FIT
  C
            LS=30
```

```
                LTHETA=12
                DIAG(1)=.FALSE.
                DIAG(2)=.FALSE.
                DIAG(3)=.FALSE.
                CALL BESTFIT(BEST,B) !FINDS BEST FIT FOR PRESENT LS AND LTHETA VALU
                LS=15
                CALL BESTFIT(BEST,B) !FINDS BEST FIT FOR PRESENT LS AND LTHETA VALU
                LS=8
                CALL BESTFIT(BEST,B) !FINDS BEST FIT FOR PRESENT LS AND LTHETA VALU
                LS=4
                DIAG(1)=.TRUE.
                DIAG(2)=.TRUE.
                DIAG(3)=.TRUE.
                CALL BESTFIT(BEST,B) !FINDS BEST FIT FOR PRESENT LS AND LTHETA VALU
                DO 21 Z=1,25
                  TOTAL(Z)=TOTAL(Z)+DISTAN(Z)    !ADD CURRENT DISTANCES TO TOTALS
      21        CONTINUE
                CALL TRNSFX(A,2,IMX(2),BEST(3),BEST(2),BEST(1))
                CALL MAK40(ISTAB,ISCNT)
                TSCNT(L)=ISCNT
                DO 23 Z=1,ISCNT
                  DO 22 K=1,6
                    TABLE(L,Z,K)=ISTAB(Z,K)
      22          CONTINUE
      23        CONTINUE
      20      CONTINUE
              WRITE(6,1000)
      1000    FORMAT(1H1,'********** OVERALL DISTANCE SUMMARY **********')
              WRITE(6,1001)
      1001    FORMAT(1X,'   0   1   2   3   4   5   6   7   8   9  10  11'
           * ,'  12  13  14  15  16  17  18  19  20  21  22  23  24')
              WRITE(6,1002)(TOTAL(K),K=1,25)
      1002    FORMAT(1X,25I4)
      C
      C **     INITIALIZE CHART VARIABLES
      C
              DO 880 L=1,NUM
                GROUP(L)=0
      880     CONTINUE
              DO 890 L=1,IMX(1)
                DO 895 Z=1,10
                  CHART(L,Z)=0
      895       CONTINUE
      890     CONTINUE
      C
      C **     CREATE MINUTIAE MATCHED ARRAY
      C
              DO 900 L=1,NUM
                DO 910 Z=1,TSCNT(L)
                  CHART(TABLE(L,Z,4),L)=TABLE(L,Z,5)
      910       CONTINUE
      900     CONTINUE
      C
      C **     PRINT MINUTIAE MATCH CHART
      C **     COUNT NUMBER OF MATCHES PER MINUTIAE
      C **     CREATE MINUTIAE MATCHED SUMMARY
      C
              DO 920 Z=1,IMX(1)
                WRITE(6,1004)Z,(CHART(Z,L),L=1,10)
      1004      FORMAT(1X,I4,10I4)
                CNT=0
                DO 930 Y=1,10
                  IF (CHART(Z,Y).EQ.0) GOTO 930
                  CNT=CNT+1
      930       CONTINUE
                IF (CNT.EQ.0) GOTO 920
                GROUP(CNT)=GROUP(CNT)+1
      920     CONTINUE
      C
      C **     PRINT MINUTIAE MATCHED SUMMARY
      C
              DO 940 Z=1,NUM
                WRITE(6,1005)GROUP(Z),Z
                FORMAT(1X,I3,' MINUTIAE MATCHED',I3,' TIMES')
      940     CONTINUE
              STOP
              END
```

```
            SUBROUTINE BESTFIT(BEST,B)
      C
      C **  BY R. ALLEN WILKINSON
      C
      C **  INITALIZE VARIABLES
      C
            IMPLICIT INTEGER (A-Z)
            INTEGER*2 B(250,3,2)
            INTEGER I,J,J1,K,K1,VALUE,S,Q,TOT,CNT
            INTEGER ISTAB(500,6),ISCNT,SAVE(4),DIST,NOCNT
            INTEGER DX,DY,DT,BEST(4)
            REAL RTEMP
            CHARACTER*8 WORD(3)
            CHARACTER OUT(-45:45,-45:45),OVER*97
      C
      C **  SAVE(I)     SAVE OF INITIAL BEST FIT VALUES
      C **  BEST(I)     BEST FIT VALUES
      C                      I=1   DELTA THETA
      C                      I=2   DELTA Y
      C                      I=3   DELTA X
      C                      I=4   SCORE
      C **  DX          DELTA X
      C **  DY          DELTA Y
      C **  DT          DELTA THETA
      C **  CNT         COUNTER
      C **  NOCNT       COUNTER   SCORES NOT WORTH COUNTING
      C **  DIST        DISTANCE BETWEEN TWO S TABLE ENTRIES
      C
      C
            DATA WORD/'ROTATION','DELTA Y ','DELTA X '/
      C
            INCLUDE '[WILKINSON.LIB]COM004.FOR/LIST'
    1 C
    1       INTEGER*2 A
    1       INTEGER IMX
    1 C
    1       COMMON /BLK004/A(250,3,2),IMX(2)
    1 C **  A(250,3,2)  MINUTIAE ARRAY(MINUTIAE #,DATA TYPE,PRINT TYPE)
    1 C **              DATA TYPE- X=1,Y=2,THETA=3
    1 C **              PRINT TYPE- SEARCH PRINT=1,FILE PRINT=2
    1 C **  IMX(2)      NUMBER OF MINUTIAE(PRINT TYPE)
    1 C
            INCLUDE '[WILKINSON.LIB]COM007.FOR/LIST'
    1 C
    1       COMMON /BLK007/LN(51),LT(51),LS,LTHETA,MTM2,IPRMT,UPPER(2),
    1      * ISBZ(2),ISRZ(2),ISUNIT,SCORES(40),IPRM2,IJUMP
    1 C
            INCLUDE '[WILKINSON.LIB]COM008.FOR/LIST'
    1       COMMON /BLK008/STABLE(-45:45,-45:45),  !TABLE OF DELTA Y,X SCORES
    1      * SCUBE(-45:45),             !POINTER TO HEAP BY VALUE OF DELTAY
    1      * HEAP(600,4),               !UP TO 300 DELTA X,THETA,SCORE,FLG/PTR
    1      * IRNG(120),                 !RANGES FOR EACH PLANE
    1      * HIGHS(-29:30,3),           !CONTIANS MAX SCORES OF WINDOWS
    1      * HWIND,VWIND,AWIND,         !DIMENSIONS OF WINDOW
    1      * MAXRG1,MAXRG2,MAXRNG,MAXWIN !MAXIMUM WINDOW VARIABLES
            INCLUDE '[WILKINSON.LIB]COM012.FOR/LIST'
    1       LOGICAL DIAG
    1       COMMON /BLK012/DIAG(100)
    1 C ** FLAG        SUBROUTINE   DIAGNOSTIC
    1 C ** DIAG(1)     BESTFIT      S TABLE INFO PRINTED
    1 C ** DIAG(2)     BESTFIT      S TABLE DX,DY VALUES PLOTTED
    1 C ** DIAG(3)     BESTFIT      DISTANCE SUMMARIES PRINTED
    1 C ** DIAG(4)     BESTFIT      USING SCORE TOTALS NOT AVERAGE
            INCLUDE '[WILKINSON.LIB]COM013.FOR/LIST'
    1 C
    1       CHARACTER TMPFILE*40
    1       INTEGER DISTAN
    1 C
    1       COMMON /BLK013/TMPFILE,DISTAN(25)
    1 C ** USED IN OPENFBI
    1 C **  TMPFILE    ORIGINAL FILE NAME UPON OPENNING
    1 C ** USED IN BESTFIT
    1 C **  DISTAN(I)  DISTANCE FROM CENTER COUNTS ON A FINGER (I=DISTANCE)
    1 C
      C
            OVER=' '
            DO 19 Q=1,25
              DISTAN(Q)=0
      19    CONTINUE
            DO 20 Q=1,4
              SAVE(Q)=BEST(Q)   !SET SAVE TO VALUES ASSUMED BEST FIT
      20    CONTINUE            !VALUES FOR TRANSFORMATION
            BEST(4)=0
```

```
          PRINT *,'ENTER BEGINNING AND ENDING VALUES FOR ',WORD(1)
          READ (5,'(I3)')K,K1
          WRITE (6,150)WORD(1),K,K1
  150     FORMAT (1H1,'TABLE FOR ',A8,I4,' THRU',I3)
          CNT=1
          WRITE(6,180)BEST(3),BEST(2),BEST(1)
  180     FORMAT (1X,'STARTING TRANSFORMATION IS X=',I3,' Y='
     *    ,I3,' THETA=',I3)
          DO 60 VALUE=K,K1
            DX=0
            DY=0
            DT=0
            TEST=SAVE(1)+VALUE
            CALL TRNSFX(A,2,IMX(2),SAVE(3),SAVE(2),TEST)
            CALL MAK40(ISTAB,ISCNT)
            S=0
            MAX=1
            NOCNT=0
            DO 40 J=1,ISCNT
              IF(ISTAB(J,6).EQ.0) THEN
                NOCNT=NOCNT+1
                GOTO 40
              ENDIF
              DX=DX+ISTAB(J,1)
              DY=DY+ISTAB(J,2)
              DT=DT+ISTAB(J,3)
              S=S+ISTAB(J,6)
  40        CONTINUE
            IF (ISCNT.EQ.0) GOTO 70
            IF (ISCNT.LT.NOCNT) GOTO 70
            IF (ISCNT.EQ.NOCNT) GOTO 70
            RTEMP=DX/(ISCNT-NOCNT)
            DX=JNINT(RTEMP)
            RTEMP=DY/(ISCNT-NOCNT)
            DY=JNINT(RTEMP)
            RTEMP=DT/(ISCNT-NOCNT)
            DT=JNINT(RTEMP)
            IF (ISTAB(J,6).GT.ISTAB(MAX,6)) MAX=J
            IF (.NOT.DIAG(4)) THEN        !DIAG(4) TRUE DO TOTAL
              RTEMP=S/(ISCNT-NOCNT)
              S=JNINT(RTEMP)
            ENDIF
            WRITE (6,50)DX,DY,DT,S,WORD(1),TEST
  50        FORMAT (1X,'DX:',I3,' DY:',I3,' DT:',I3,' SCORE:',I5,
     *      ' AT ',A8,' VALUE:',I3)
            IF (BEST(4).LT.S) THEN
              BEST(4)=S
              TOT=TEST
              CNT=1
            ELSE
              IF (BEST(4).EQ.S) THEN
                CNT=CNT+1
                TOT=TOT+TEST
              ENDIF
            ENDIF
            GOTO 51
  70        WRITE (6,52)WORD(1),TEST
  52        FORMAT (1X,'ISCNT=0 NO SCORE AT ',A8,' VALUE',I4)
  51        DO 53 J=1,250
              DO 55 I=1,3
                A(J,I,2)=B(J,I,2)
  55          CONTINUE
  53        CONTINUE
  60      CONTINUE
          RTEMP=TOT/CNT
          BEST(1)=JNINT(RTEMP)
          WRITE (6,181) BEST(3),BEST(2),BEST(1)
  181     FORMAT (1X,'BEST TRANSFORMATION IS X=',I3,' Y='
     *    ,I3,' THETA=',I3)
          CALL TRNSFX(A,2,IMX(2),BEST(3),BEST(2),BEST(1))
          CALL MAK40(ISTAB,ISCNT)     !GENERATE ISTAB
          DO 189 J=1,IMX(2)
            DO 188 J1=1,3
              A(J,J1,2)=B(J,J1,2)
  188       CONTINUE
  189     CONTINUE
          DO 190 J=-45,45
            DO 195 J1=-45,45
              STABLE(J,J1)=0
  195       CONTINUE
  190     CONTINUE
  C
  C **   LOAD PLOT ARRAY
```

```
C
      DO 200 J=1,ISCNT
        IF (.NOT.DIAG(1)) GOTO 202    !PRINT S TABLE IF TRUE
        WRITE (6,201)J,(ISTAB(J,K),K=1,6)
201     FORMAT(1X,'ISCNT:',I3,' DX:',I3,' DY:',I3,' DT',I3,
     *  ' SM#:',I3,' FM#:',I3,' SCORE:',I5)
202     STABLE(ISTAB(J,2),ISTAB(J,1))=
     *  STABLE(ISTAB(J,2),ISTAB(J,1))+1
200   CONTINUE
C
C **  CONVERT STABLE ARRAY TO BE OUTPUTTED
C **  0-9, A-Z , a-z
C **  0-9,10-35,36-61
C
      MAX=0
      DO 250 J=-45,45
        DO 240 J1=-45,45
          IF (STABLE(J,J1).EQ.0) THEN
            OUT(J,J1)=' '
            GOTO 240
          ENDIF
          IF (STABLE(J,J1).LT.10) THEN
            OUT(J,J1)=CHAR(STABLE(J,J1)+48)
          ELSE
            IF (STABLE(J,J1).GT.35) THEN
              OUT(J,J1)=CHAR(STABLE(J,J1)+60)
            ELSE
              OUT(J,J1)=CHAR(STABLE(J,J1)+55)
            ENDIF
          ENDIF
240     CONTINUE
250   CONTINUE
      CALL WINDOW(5,5)
      BEST(3)=BEST(3)-HWIND
      BEST(2)=BEST(2)-VWIND
      CALL TRNSFX(A,2,IMX(2),BEST(3),BEST(2),BEST(1))
      CALL MAK40(ISTAB,ISCNT)     !GENERATE ISTAB
      DO 251 J=1,IMX(2)
        DO 252 J1=1,3
          A(J,J1,2)=B(J,J1,2)
252     CONTINUE
251   CONTINUE
C
C **  S TABLE CHART
C
      IF (.NOT.DIAG(2)) GOTO 261        !PRINT S TABLE PLOTTING IF TRUE
      WRITE (6,225)
225   FORMAT(1X,'     ....-4.........-3.........-2.........-1',
     *    '.........0.........1.........2.........3.........4',
     *    '.....')
      DO 260 J=-45,45
        WRITE (6,235)J,(OUT(J,K),K=-45,45),J
235     FORMAT(1X,I3,91A1,I3)
        IF (J.EQ.VWIND) THEN
          OVER(49+HWIND:49+HWIND)='/'
          WRITE(6,236)OVER
236       FORMAT('+',A97)
        ENDIF
260   CONTINUE
      WRITE (6,225)
C
C **  SUMMARY DISTANCE CHART
C
261   IF (.NOT.DIAG(3)) GOTO 300        !PRINT DISTANCE SUMMARY CHART
      WRITE(6,210)
210   FORMAT(1H1,'SUMMARY CHART OF BEST FIT DISTANCES')
      DO 209 J=1,ISCNT
        DIST=(ISTAB(J,1)-HWIND)**2
        DIST=DIST+(ISTAB(J,2)-VWIND)**2
        DIST=JNINT(SQRT(FLOAT(DIST)))
        IF (DIST.GT.24.OR.DIST.LT.0) GOTO 209
        DISTAN(DIST+1)=DISTAN(DIST+1)+1
209   CONTINUE
      WRITE (6,212)
212   FORMAT(1X,'   0   1   2   3   4   5   6   7   8   9  10'
     * ,'  11  12  13  14  15  16  17  18  19  20  21  22  23  24')
      WRITE (6,230)(DISTAN(K),K=1,25)
230   FORMAT(1X,25I4)
      TEMP=DISTAN(1)+DISTAN(2)+DISTAN(3)+DISTAN(4)+DISTAN(5)
      PRINT *,'TOTAL 0-4:',TEMP
      K=TEMP-(IMX(1)-TEMP)-(IMX(2)-TEMP)
      RTEMP=(FLOAT(K))/(FLOAT(IMX(1)))
      WRITE(6,183)RTEMP
```

```
 183    FORMAT (1X,'DETECTION SCORE: ',F6.3)
        WRITE(6,184)ISCNT,IMX(1),IMX(2)
 184    FORMAT (1X,'ISCNT: ',I4,' IMX(1): ',I4,' IMX(2): ',I4)
 300    WRITE(6,182)BEST(3),BEST(2),BEST(1)
 182    FORMAT (1X,'BEST TRANSFORMATION IS X=',I3,' Y='
     * ,I3,' THETA=',I3)
        RETURN
        END
```

```
            SUBROUTINE FBIOPEN(UNIT)
      C
      C **   OPENS A FINGERPRINT FILE
      C ** !! FOR VAX ONLY !! **
      C
      C **   BY R. ALLEN WILKINSON
      C
      C **   INITALIZE VARIABLES
      C
            INTEGER UNIT
      C
            INCLUDE '[WILKINSON.LIB]COM013.FOR/LIST'
    1 C
    1       CHARACTER TMPFILE*40
    1       INTEGER DISTAN
    1 C
    1       COMMON /BLK013/TMPFILE,DISTAN(25)
    1 C **  USED IN OPENFBI
    1 C **   TMPFILE    ORIGINAL FILE NAME UPON OPENNING
    1 C **  USED IN BESTFIT
    1 C **   DISTAN(I)  DISTANCE FROM CENTER COUNTS ON A FINGER (I=DISTANCE)
    1 C
      C
      C **   OPEN FILES
      C
            PRINT *,'ENTER FILE IN SINGLE QUOTES'
            READ (5,*)TMPFILE
            OPEN(UNIT,FILE=TMPFILE,STATUS='UNKNOWN',BLOCKSIZE=15192,
     *           BUFFERCOUNT=1,RECL=15192,RECORDTYPE='VARIABLE',
     *           ORGANIZATION='SEQUENTIAL',FORM='FORMATTED',
     *           IOSTAT=IOS,ERR=1000)
            RETURN
      1000  WRITE(6,1010)IOS
      1010  FORMAT(1X,'### ERROR: ',I3,' ###')
            RETURN
            END
```

```
              SUBROUTINE FBIREAD(UNIT)
        C
        C **   READS A FINGERPRINT FILE
        C ** !! FOR VAX ONLY !! **
        C
        C **   BY R. ALLEN WILKINSON
        C
        C **   INITALIZE VARIABLES
        C
              IMPLICIT INTEGER (A-Z)
              INTEGER IOS,TEMP,UNIT,CARD,K,PNT
        C
              INCLUDE 'COMIO.FOR/LIST'
      1 C
      1       STRUCTURE /FBIDAT/
      1         UNION
      1           MAP
      1             CHARACTER*10 PCNUM
      1             CHARACTER*40 CLASS
      1             CHARACTER*9  CDATE
      1             CHARACTER*8  CTIME
      1             CHARACTER*3  FILL
      1             INTEGER*2    CARD
      1             INTEGER*2    OFFSET(10)
      1             INTEGER*2    VALUES(7550)
      1           END MAP
      1           MAP
      1             CHARACTER*15192  BUF
      1           END MAP
      1         END UNION
      1       END STRUCTURE
      1       RECORD /FBIDAT/INPUT
      1 C
      1       COMMON /BLKIO/INPUT
      1 C **   USED IN NEW FBI I/O ROUTINES - FBIOPEN,FBIREAD,FBIWRITE
      1 C **     PCNUM        PROCESS CONTROL NUMBER
      1 C **     CLASS        CLASSIFICATION
      1 C **     CDATE        FILE CREATION DATE
      1 C **     CTIME        FILE CREATION TIME
      1 C **     FILL         JUST USED TO FILL TO WORD BOUNDARY
      1 C **     OFFSET(I)    OFFSET FROM PCN TO FINGER #I'S DATA
      1 C **     VALUES(I)    ACTUAL MINUTIAE DATA ARRAY
      1 C
              INCLUDE 'COMDATA.FOR/LIST'
      1 C
      1 C *  DATA ARRAYS  *
      1       INTEGER*2 FINGER,QUALITY,MINCNT,XYT
      1 C
      1       COMMON /BLKDATA/FINGER(10),QUALITY(10),MINCNT(10),
      1      *                 XYT(250,3,10)
      1 C **  USED IN NEW FBI DATA ORGANIZATION
      1 C **    FINGER(I)    Ith FINGER'S ACTUAL NUMBER
      1 C **    QUALITY(I)   Ith FINGER'S DATA QUALITY
      1 C **    MINCNT(I)    Ith FINGER'S NUMBER OF MINUTIAE
      1 C **    XYT(J,K,I)   Ith FINGER'S Jth MINUTIAE DATA
      1 C **                      WHERE K=
      1 C **                      1 FOR X DATA VALUE 0-255
      1 C **                      2 FOR Y DATA VALUE 0-255
      1 C **                      3 FOR THETA DATA VALUE 0-359
      1 C
        C
        C **   SELECT CARD
        C
              WRITE(6,*)'ENTER CARD TO BE READ'
              READ(5,*)CARD
        C
        C **   READ FILES
        C
        10    CONTINUE
              READ(7,40,IOSTAT=IOS,ERR=1000,END=1100)
             * NBYTES,INPUT.BUF
        40    FORMAT(Q,A)
              IF (CARD.NE.INPUT.CARD) GOTO 10
              DO 100 I=1,10
                IF (INPUT.OFFSET(I).EQ.0) THEN
                  MINCNT(I)=0
                ELSE
                  FINGER(I)=INPUT.VALUES(INPUT.OFFSET(I)-46+2)
                  QUALITY(I)=INPUT.VALUES(INPUT.OFFSET(I)-46+3)
                  MINCNT(I)=INPUT.VALUES(INPUT.OFFSET(I)-46+4)
                  PNT=INPUT.OFFSET(I)-42
                  DO 150 J=1,MINCNT(I)
                    K=(J-1)*3+1
```

```
                 XYT(J,2,I)=INPUT.VALUES(PNT+K)
                 XYT(J,1,I)=INPUT.VALUES(PNT+K+1)
                 XYT(J,3,I)=INPUT.VALUES(PNT+K+2)
      150      CONTINUE
             ENDIF
      100   CONTINUE
            RETURN
      1000  WRITE(6,1010)IOS
      1010  FORMAT(1X,'### ERROR: ',I3,' DURING READ ###')
            RETURN
      1100  WRITE(6,1110)
      1110  FORMAT(1X,'END OF FILE REACHED!!!!')
            WRITE(6,1120)CARD
      1120  FORMAT(1X,'CARD #',I7,' MAY NOT BE IN FILE')
            RETURN
            END
```

```
              SUBROUTINE MAK40(ISTAB,ISCNT)
              PARAMETER(IPRM1=500)
      C
      C **    ROUTINE TO FORM TABLE OF CLOSENESS...VALUES
      C
      C **        MAY 22, 1984   1640
      C **        AUGUST 25, 1986
      C **        SEPTEMBER 18, 1986
      C **        TR CALCULATED AS THE HYPOTENUSE RATHER THAN BY JUST LIMITS
      C **        IN X AND Y........
      C
                  IMPLICIT INTEGER(A-Q,S-Z)
                  REAL TR,TEMP(IPRM1)
                  DIMENSION ISTAB(IPRM1,6)
                  INCLUDE 'COM004.FOR/LIST'
    1 C
    1         INTEGER*2 A
    1         INTEGER IMX
    1 C
    1         COMMON /BLK004/A(250,3,2),IMX(2)
    1 C **    A(250,3,2)  MINUTIAE ARRAY(MINUTIAE #,DATA TYPE,PRINT TYPE)
    1 C **                DATA TYPE- X=1,Y=2,THETA=3
    1 C **                PRINT TYPE- SEARCH PRINT=1,FILE PRINT=2
    1 C **    IMX(2)      NUMBER OF MINUTIAE(PRINT TYPE)
    1 C
      C
                  INCLUDE 'COM007.FOR/LIST'
    1 C
    1         COMMON /BLK007/LN(51),LT(51),LS,LTHETA,MTM2,IPRMT,UPPER(2),
    1       * ISBZ(2),ISRZ(2),ISUNIT,SCORES(40),IPRM2,IJUMP
    1 C
      C
      C **    SETUP THE FIRST 2 STEPS.  WHEN AN ENTRY IS MADE IN THE ISTAB ON ONE
      C       LINE REPRESENTING THE PAIR (i,j), THEN THE SAME ENTRY WILL BE MADE
      C       EVENTUALLY IN THE SLOT REPRESENTING THE PAIR (j,i).
      C

              ISCNT=0                           ! NUMBER OF SLOTS USED
              DO 100 I1=1,IMX(1)                ! CYCLE TO 100 FOR EACH SEARCH MIN.
              LSL=A(I1,1,1)-LS                  ! LEFT EDGE OF SRCH BOX
              LSR=A(I1,1,1)+LS                  ! RIGHT EDGE
              LSB=A(I1,2,1)-LS                  ! BOTTOM EDGE
              LST=A(I1,2,1)+LS                  ! TOP EDGE
              DO 90 J1=1,IMX(2)                 ! LOOK AT EACH FILE MINUTIA
              IF(A(J1,2,2).GE.LSB)THEN          ! ABOVE BOTTOM EDGE
                IF(A(J1,2,2).LE.LST) THEN       ! BELOW TOP EDGE
                  IF(A(J1,1,2).GE.LSL) THEN     ! TO THE RIGHT OF LEFT EDGE
                    IF(A(J1,1,2).LE.LSR)THEN    ! TO THE LEFT OF RIGHT EDGE
                      DIST=(A(J1,1,2)-A(I1,1,1))**2+
           *          (A(J1,2,2)-A(I1,2,1))**2
                      IF(DIST.LE.(LS**2)) THEN
                        DELTHE=A(J1,3,2)-A(I1,3,1)      ! DELTA THETA
                        IF(IABS(DELTHE).GT.LTHETA)THEN
                          IF(DELTHE.GE.0)THEN
                            DELTHE=DELTHE-360
                          ELSE
                            DELTHE=DELTHE+360
                          ENDIF
                        ENDIF
                        IF(IABS(DELTHE).LE.LTHETA)THEN
                          ISCNT=ISCNT+1         ! FILE MIN. WITHIN BOX
                          IF(ISCNT.GT.500) THEN    ! RAN OUT OF ROOM IN ISTAB
                            PRINT 60,I1,J1
      60                    FORMAT(//1X,'### MATCH87 ERROR --EXCEEDED ',
           *                'ISTAB-- I1/J1=',2I4)
                            STOP
                          ENDIF
                          ISTAB(ISCNT,1)=A(J1,1,2)-A(I1,1,1)  ! DELTA X
                          ISTAB(ISCNT,2)=A(J1,2,2)-A(I1,2,1)  ! DELTA Y
                          ISTAB(ISCNT,3)=DELTHE               ! DELTA THETA
                          ISTAB(ISCNT,4)=I1           ! SEARCH MIN. #
                          ISTAB(ISCNT,5)=J1           ! FILE MINUTIA #
                          TEMP(ISCNT)=0               ! TS SCORE
                        ENDIF
                      ENDIF
                    ENDIF
                  ENDIF
                ENDIF
              ENDIF
      90      CONTINUE
      100     CONTINUE
      C
      C **    COMPARE THE LIST OR SLOTS A FORM INITIAL SCORES
```

```
C
      IF(LS.LT.10) THEN
        KR=LS+1
      ELSE
        KR=10
      ENDIF
      KR=5
      DO 150 I2=1,ISCNT
        DO 140 J2=I2+1,ISCNT                ! LOOK @ THEM ALL
          IF(I2.EQ.J2) GO TO 140            ! DON'T COMPARE TO SELF
          TR=(ISTAB(I2,1)-ISTAB(J2,1))**2+(ISTAB(I2,2)-ISTAB(J2,2))**2
          TR=SQRT(TR)
          IF((KR-TR).GT.0)THEN
            TEMP(I2)=TEMP(I2)+KR-TR
            TEMP(J2)=TEMP(J2)+KR-TR
          ENDIF
140     CONTINUE
150   CONTINUE
      DO 160 J2=1,ISCNT
        ISTAB(J2,6)=JNINT(TEMP(J2))
160   CONTINUE
      RETURN
      END
```

```
C
C
      SUBROUTINE TRNSFX(A,IK,IMX,DELTAX,DELTAY,DELTHE)
C
C **  USED FOR ROTATING AND TRANSLATING ABOUT 128 x 128
C
C **  THIS ROUTINE CONSISTS OF A SERIES OF TRANSFORMS WHICH SUPERIMPOSES
C **  THE CENTRAL MINUTIA OVER THE FILE MINUTIA AND REPOSITIONS AND
C **  RECALCULATES THE NEW COORDINATES AND ANGLE FOR EACH IMX(1) MINUTIA
C
C **  THE TRANSFORM TO BE APPLYIED CONSISTS OF 4 SEPARATE ONES.  THE FIRST
C **  [T1] TRANSLATES THE SEARCH MINUTIA OVER THE CORRESPONDING FILE MINUTI
C **  THE SECOND [T2] TRANSLATES THE SUPERIMPOSED MINUTIAE TO THE ORIGIN; T
C **  THIRD [T3] ROTATES THE MINUTIAE BY -DELTHE DEGREES ( NEG. SIGN INDICA
C **  A COUNTERCLOCKWISE ROTATION); THE FOURTH TRANSFORM [T4] RETRANSLATES
C **  THE SEARCH MINUTIAE GROUP OVER THE CORRESPONDING CENTRAL FILE MINUTIA
C
C **  [X',Y',1] = [X,Y,1] [T1] [T2] [T3] [T4]
C
C **  THE TRANSFORMS MAY BE EXPRESSED AS:
C             --          --    --          --    --            --    --         --
C            | 1    0    0|    | 1    0    0|    |COS   -SIN   0|    |1    0    0|
C     [T] =  | 0    1    0| *  | 0    1    0| *  |SIN    COS   0| *  |0    1    0|
C            | RX0  RY0  1|    |-RX  -RY   1|    | 0      0    1|    |RX   RY   1|
C             ---         --    --          --    --            --    --         --
C
C     WHERE: RX0/RY0 ARE DELTAX AND DELTAY AND
C            RX / RY ARE THE X & Y COORDINATES OF THE CENTRAL FILE MINUTIA
C
C **  AFTER ALL THE 4 TRANSFORMS ARE SIMPLIFIED (MULTIPLIED) TOGETHER THE
C **  NEW X AND Y VALUES OF THE MINUTIA MAY BE EXPRESSED AS :
C
C     X' = COS( X+X1) + SIN( Y+Y1) + RX
C     Y' = COS( Y+Y1) - SIN( X+X1) + RY
C
C     WHERE X1 = DELTAX-RX ; AND Y1 = DELTAY-RY
C
C **  TRNSFX(MINUTIAE DATA ARRAY,FINGER,NUMBER OF MINUTIAE
C **         OF FINGER,CHANGE IN X,CHANGE IN Y,CHANGE IN
C **         THETA)
C
      IMPLICIT INTEGER (A,Z)
      INTEGER*2 A,X1,X2,Y1,Y2,RX,RY
      INTEGER*2 DELTAX,DELTAY,DELTHE
      REAL RSIN,RCOS,RAD,RTEMP
      DIMENSION A(250,3,10)
      RAD=57.29578                      !RADIANS PER DEGREE
      RSIN=SIN(-DELTHE/RAD)
      RCOS=COS(-DELTHE/RAD)
      RX=128                            !CENTER X COORDINATE
      RY=128                            !CENTER Y COORDINATE
      X1=DELTAX-RX                      !CALCULATE X OFFSET FROM CENTER
      Y1=DELTAY-RY                      !CALCULATE Y OFFSET FROM CENTER
      DO 103 K=1,IMX
      X2=A(K,1,IK)+X1                   !MOVE BY X OFFSET
      Y2=A(K,2,IK)+Y1                   !MOVE BY Y OFFSET
      RTEMP=RCOS*X2+RSIN*Y2+RX          !CALCULATE NEW X VALUE
      A(K,1,IK)=JNINT(RTEMP)
      RTEMP=RCOS*Y2-RSIN*X2+RY          !CALCULATE NEW Y VALUE
      A(K,2,IK)=JNINT(RTEMP)
      A(K,3,IK)=A(K,3,IK)+DELTHE+360 !MAKE SURE IT IS NOT NEGATIVE
      A(K,3,IK)=MOD(A(K,3,IK),360)   !CHANGE TO 0-359 NUMBER
103   CONTINUE
      RETURN
      END
```

```
          SUBROUTINE WINDOW(OFFX,OFFY)
     C
     C **   FINDS HIGHEST TOTAL SCORE IN A WINDOW SIZE
     C **   OFFX BY OFFY
     C **   RETURN COORDINATES OF CENTER IN VARIABLE
     C **   HWIND AND VWIND (X,Y RESPECTIVELY)
     C
          IMPLICIT INTEGER (A-Z)
          REAL RTEMP
          DIMENSION MAX(3),SUM(3)
     C
     C **   MAX(I)    HIGHEST SCORE IN WINDOW
     C **   SUM(I)    SCORE OF CURRENT WINDOW
     C
          INCLUDE '[WILKINSON.LIB]COM007.FOR/LIST'
   1 C
   1      COMMON /BLK007/LN(51),LT(51),LS,LTHETA,MTM2,IPRMT,UPPER(2),
   1     * ISBZ(2),ISRZ(2),ISUNIT,SCORES(40),IPRM2,IJUMP
   1 C
          INCLUDE '[WILKINSON.LIB]COM008.FOR/LIST'
   1      COMMON /BLK008/STABLE(-45:45,-45:45),  !TABLE OF DELTA Y,X SCORES
   1     * SCUBE(-45:45),             !POINTER TO HEAP BY VALUE OF DELTAY
   1     * HEAP(600,4),               !UP TO 300 DELTA X,THETA,SCORE,FLG/PTR
   1     * IRNG(120),                 !RANGES FOR EACH PLANE
   1     * HIGHS(-29:30,3),           !CONTIANS MAX SCORES OF WINDOWS
   1     * HWIND,VWIND,AWIND,         !DIMENSIONS OF WINDOW
   1     * MAXRG1,MAXRG2,MAXRNG,MAXWIN !MAXIMUM WINDOW VARIABLES
          MAX(1)=0
          MAX(2)=0
          MAX(3)=0
          DO 100 I=-LS,LS                     !CENTER Y
            DO 150 J=-LS,LS                   !CENTER X
              SUM(1)=0
              SUM(2)=0
              SUM(3)=0
              X=(OFFX-1)/2
              Y=(OFFY-1)/2
              DO 160 K=I-Y,I+Y                !Y EDGE LIMITS
                DO 170 L=J-X,J+X              !X EDGE LIMITS
                  SUM(1)=SUM(1)+STABLE(K,L)      !SUM OF WINDOW
                  IF (K.GT.(I-Y).AND.K.LT.(I+Y).AND.L.GT.(J-X).
     *            AND.L.LT.(J+X)) SUM(2)=SUM(2)+STABLE(K,L)
                  IF (K.GT.(I-Y+1).AND.K.LT.(I+Y-1).AND.L.GT.(J-X+1).
     *            AND.L.LT.(J+X-1)) SUM(3)=SUM(3)+STABLE(K,L)
     170        CONTINUE
     160      CONTINUE
              IF (SUM(1).LT.MAX(1)) GOTO 150     !SAVE HIGHEST SUM
              IF (MAX(1).EQ.SUM(1)) THEN
                IF (SUM(2).LT.MAX(2)) GOTO 150
                IF (SUM(2).EQ.MAX(2)) THEN
                  IF (SUM(3).LT.MAX(3)) GOTO 150
                  IF (SUM(3).EQ.MAX(3)) THEN
                    CNT=CNT+1
                    TEMPH=TEMPH+J
                    TEMPV=TEMPV+I
                    GOTO 150
                  ENDIF
                ENDIF
              ENDIF
              MAX(1)=SUM(1)                   !SAVE SUM
              MAX(2)=SUM(2)                   !SAVE SUM
              MAX(3)=SUM(3)                   !SAVE SUM
              CNT=1                           !HOW MANY AT THIS VALUE
              TEMPH=J                         !SAVE Y COORDINATE OF SUM
              TEMPV=I                         !SAVE X COORDINATE OF SUM
     150    CONTINUE
     100  CONTINUE
          RTEMP=TEMPH/CNT
          HWIND=JNINT(RTEMP)
          RTEMP=TEMPV/CNT
          VWIND=JNINT(RTEMP)
          RETURN
          END
```

| U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET (See instructions) | 1. PUBLICATION OR REPORT NO. NBSIR 88-3831 | 2. Performing Organ. Report No. | 3. Publication Date AUGUST 1988 |
|---|---|---|---|

**4. TITLE AND SUBTITLE**

AFRS Perfornace Evaluation Tests

**5. AUTHOR(S)**

R.T. Moore; R. Michael McCabe; R. Allen Wilkinson

**6. PERFORMING ORGANIZATION** *(If joint or other than NBS, see instructions)*

NATIONAL BUREAU OF STANDARDS
U.S. DEPARTMENT OF COMMERCE
GAITHERSBURG, MD 20899

**7. Contract/Grant No.**

**8. Type of Report & Period Covered**

**9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS** *(Street, City, State, ZIP)*

Federal Bureau of Investigation
10th 8 Pa. Ave., NW
Washington, DC 20537

**10. SUPPLEMENTARY NOTES**

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.


**11. ABSTRACT** *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)*

Automatic Fingerprint Reader Systems (ADRS) are designed to scan life-sized fingerprint images and to detect selected features such as minutiae. Position and orientation identifiers are detected and recorded for each of these minutiae. Generally, for a given AFRS, all of the true minutiae are not detected while several false minutiae are generated. Furthermore, true minutiae are not always detected on consecutive readings or are not detected at exactly the same locations.

Methods of measuring the reader performance as regards minutiae detections are described. Lightly, medium, and heavily inked sets of fingerprints from a single subject were used as input to two of the FBI's readers. Measures of accuracy, consistency and reliability were determined and compared between these three inkings and two systems.



**12. KEY WORDS** *(Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)*

accuracy; cluster; consistency; detection; evaluation; fingerprint; ground truth; minutiae; performance; position; reliability.


**13. AVAILABILITY**

☐ Unlimited
☒ For Official Distribution. Do Not Release to NTIS
☐ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.
☐ Order From National Technical Information Service (NTIS), Springfield, VA. 22161

**14. NO. OF PRINTED PAGES**

54

**15. Price**

USCOMM-DC 6043-P80