COMPUTATION OF OPTIMAL WEIGHTS FOR AND SIMULATION OF
A HYBRID DECISION ALGORITHM FOR
PATTERN RECOGNITION AS APPLIED TO TRACK EVENT TIMING

by

David D. Bailey

Submitted in Partial Fulfillment of the Requirements
for the Degree of
Master of Science in Engineering
in the
Electrical Engineering
Program

8/25/78

Adviser ______________________ Date

8/31/78

Dean of the Graduate School ______________________ Date

YOUNGSTOWN STATE UNIVERSITY

August, 1978

ABSTRACT

COMPUTATION OF OPTIMAL WEIGHTS FOR AND SIMULATION OF
A HYBRID DECISION ALGORITHM FOR
PATTERN RECOGNITION AS APPLIED TO TRACK EVENT TIMING

David D. Bailey
Master of Science in Engineering
Youngstown State University, 1978

This paper presents the application of a theoretically derived decision algorithm for pattern recognition in the area of automatic track event timing. The proposed system will locate (in time and space) a runner near the end of the course in a track event. A training procedure for determining a set of optimal weights essential to the algorithm is presented along with the results of simulation and testing of the system on an IBM 370 computer.

WILLIAM F. MAAG LIBRARY
YOUNGSTOWN STATE UNIVERSITY

ACKNOWLEDGEMENTS

Much thanks is given to Dr. Alexander and Jeffrey Taft for the direction and advice they gave. Without their help, this paper would not have been possible.

A special word of gratitude is given to Marian Habosky and my mother whose support and encouragement enabled me to finish a thesis that at times looked like it had no end.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

The realization of theoretical ideas in the form of practical applications in the physical world is the role most often taken by the engineer. Maxwell's equations mean very little to the real world on paper but their applications in the areas of microwave technology and field theory affect all of mankind. Many fields of study have been developed, in theory, to a high degree of complexity. However, practical applications are few and far between. One such area is pattern recognition.

Real world applications in pattern recognition exist primarily as projects of the future in the minds of today's mathematicians. Computation time and feature selection are major road blocks in the implementation of complex decision algorithms. It is possible, in certain situations to develop algorithms which are easily implemented. This paper takes a theoretically developed pattern recognition decision algorithm and uses it as a basis for a practical, real time, automatic track event timing system.

The decision algorithm was developed by Mr. Jeffrey Taft and the author and was presented in Mr. Taft's Master's thesis at Youngstown State University. [1] At that time, it was suggested that an application in the area of track timing would be possible. Further investigation by the author verified this possibility and a system for implementing the decision model was proposed. The author's work in feature selection,weight investigation and the simulation and testing of the algorithm verified the ability of the algorithm to identify the runner's position with respect to the finish line. This ability to locate the runner's position enabled the system to determine the finish time of the runner.

Chapter 2 presents the mechanics of track timing along with the current attempts of providing automatic timing systems. Chapters 3 and 4 provide some general background in pattern recognition techniques and the derivation of the decision algorithm. In Chapter 5 the decision algorithm is applied to track timing. The training procedure for providing an optimal set of weights is also presented. The simulation and testing of the algorithm is presented in Chapter 6. Chapter 7 presents a system for implementing the decision algorithm as an automatic track timing system.

CHAPTER II

CURRENT TRACK TIMING PROCEDURES

Track and field has shown tremendous growth in popularity over the past few years. Prime time telecasts of the Olympics have resulted in great exposure to the public, and jogging for fun and health has created a closer tie to track and field for many people. With improved performances by athletes, world class runners differ in finish times by only hundreths of a second. This intense competition at the top has created a need for timing systems with more precision than can be provided by the hand-held stopwatch.

The use of automatic timing systems is constrained by the definition of the finish of the race as given by the National Federation Track and Field Committee. The finish line is a vertical plane extending from the surface of the track located at the end of the course. The white line marked on the track is there as an aid to the finish judges only. A runner has successfully completed the race when the leading edge of his torso has broken the vertical finish plane. The torso is defined as the trunk of the body excluding the head, arms and legs. [2] This definition differs from that of other sports like swimming and horse racing where it is the first part of the participant's body which breaks the finish plane that constitutes the end of the race. In figure 1, the runner is seen approaching the finish plane and actually breaking the plane with his hand. The race is not over, however, until the leading edge of the torso breaks the finish plane.(figure 2)

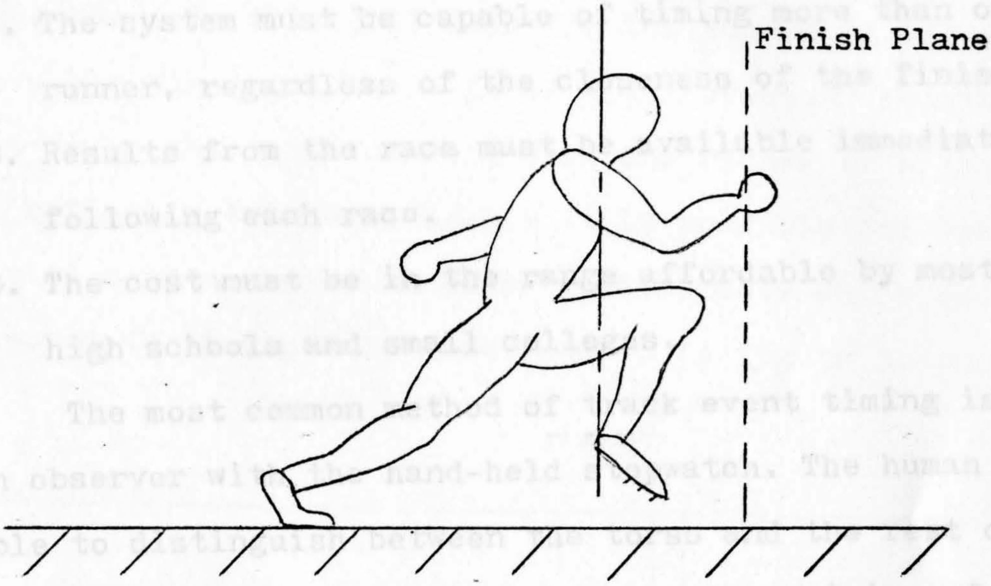


Fig. 1. Illegal finish of runner

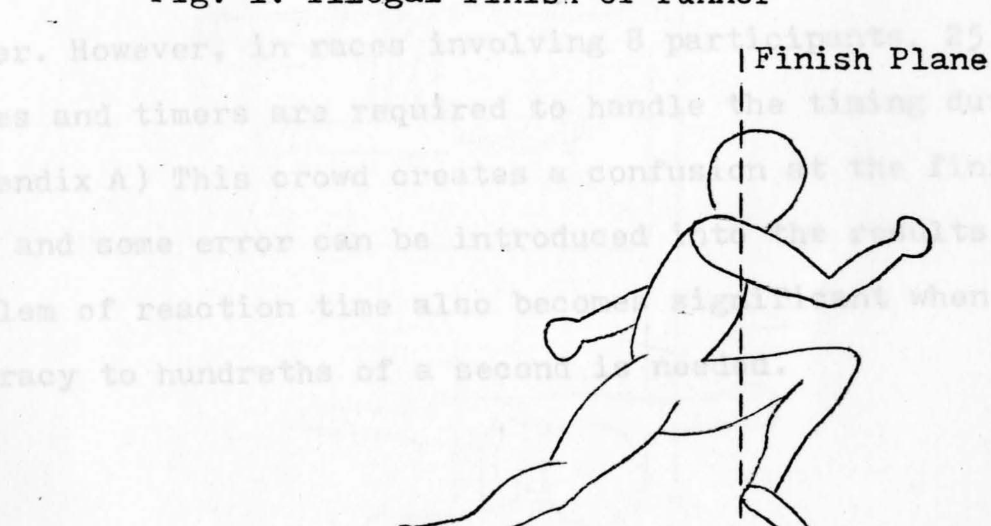


Fig. 2. Valid finish of runner

The basic problem then, in timing a track event, is one of precisely locating the leading edge of the torso, also called the chestline, with respect to the finish line. The timing system must also be able to satisfy the following:

1. Each runner must be timed to an accuracy of .01 seconds.
2. The system must be capable of timing more than one runner, regardless of the closeness of the finish.
3. Results from the race must be available immediately following each race.
4. The cost must be in the range affordable by most high schools and small colleges.

The most common method of track event timing is the human observer with the hand-held stopwatch. The human eye is able to distinguish between the torso and the rest of the body thus eliminating the danger of premature timing of the runner. However, in races involving 8 participants, 25 judges and timers are required to handle the timing duties. (Appendix A) This crowd creates a confusion at the finish line and some error can be introduced into the results. The problem of reaction time also becomes significant when accuracy to hundreths of a second is needed.

Automatic electronic timing systems have been designed to remove the human element from the actual timing. The most common system on the market is one which employs light beams to sense the presence of the runner. This system can be triggered prematurely by an outstretched arm or leg (as shown in figure 1) allowing the runner to illegally improve his time and place. It is also necessary for the system to be able to time more than one runner in a race. In an optically triggered system it is possible for one runner to cast a shadow on other participants which restricts the system to timing only that runner. In figure 3, runner A

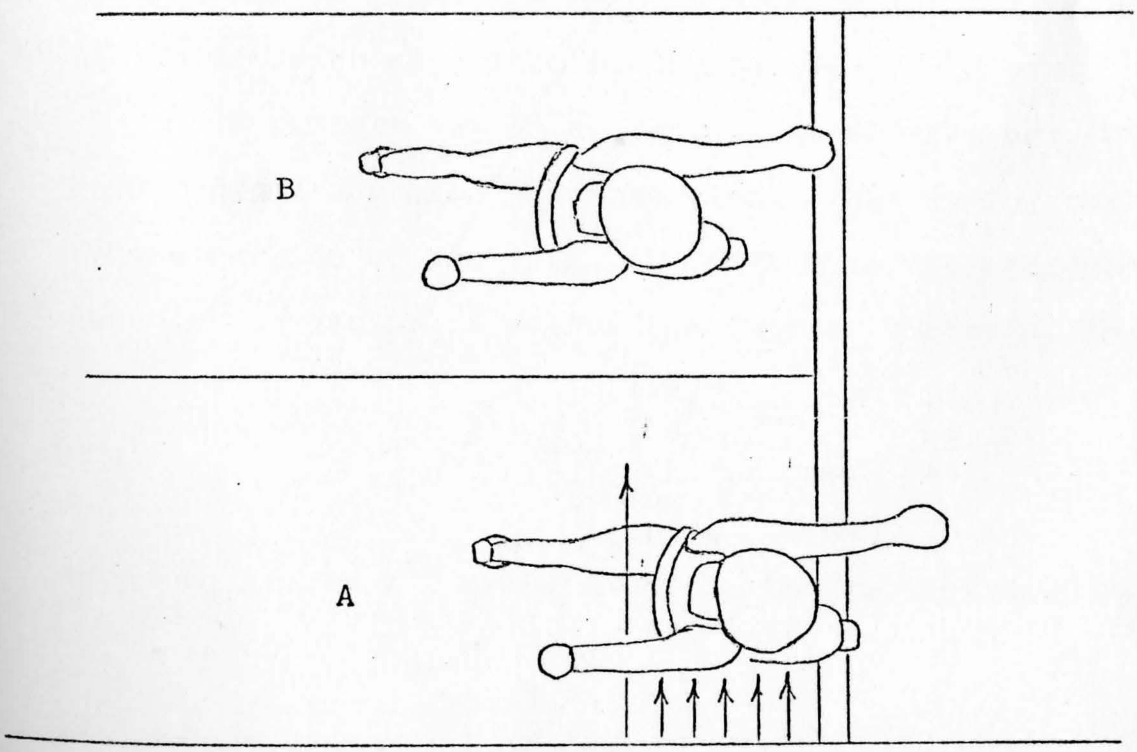


Fig. 3. Runner A shadows runner B

crosses the finish line ahead of runner B. A's shadow on B prohibits the system from timing B. These two problems make the optically triggered system unreliable in situations which are not uncommon in a race.

Other systems on the market handle these reliability problems but as reliability increases so does the cost, with some price tags exceeding $150,000.

WILLIAM F. MAAG LIBRARY
YOUNGSTOWN STATE UNIVERSITY

CHAPTER III

STANDARD APPROACHES IN PATTERN RECOGNITION

Every runner has one and only one chestline. The problem is to locate this chestline with respect to the finish line. The chestline, however, is often camouflaged in a constantly changing pattern of arms,legs and head. The precise location of the chestline in a constantly changing shape suggests a solution through pattern recognition techniques. However, standard pattern recognition techniques fall short of providing the needed accuracy in the system as will be explained later. A brief tutorial on pattern recognition is presented here as a basis for the derivation of the decision algorithm in Chapter 4.

A pattern can be defined as an ordered set of measurements arranged in vector form. This vector must allow reconstruction of the original image from the measurements taken. This vector is called the feature vector $\bar{x}$. Thus

$$\bar{x} = \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_n \end{bmatrix} \qquad (1)$$

where m_k is the k^{th} measurement in the vector. n is refered to as the dimension of the feature space and corresponds to the total number of measurements. The measurements taken for the feature vector may be obtained in several ways. The vector may be assembled from a series of samples on a waveform or from measurements as diversified as the weight, color and length of a runner's body. A vector, composed of a series of components which indicate whether a particular characteristic is present or not, is refered to as a binary vector. Vectors of this type are used in automatic classification schemes for information retrieval.[3]

The n-dimensional feature vector is said to lie in an orthogonal pattern space. A basis set of orthogonal unit vectors corresponding to unit measures of the features defines the pattern space. Patterns which are classified together form "clusters" in the pattern space. These "clusters" are known as pattern classes, denoted ω_i. If a region of pattern space is defined so the entire cluster is included in the area, then the pattern class may be defined in terms of the boundaries of the region.

A wide variety of pattern recognition techniques have been developed to classify sample patterns in terms of known pattern classes. However, no one technique has been developed that will work on all possible patterns.

Statistical Methods

It is possible to compute the probability or likelihood that pattern $\overline{x}$ belongs to class ω_j . This probability acts as an aid in guessing which class the pattern belongs. A loss function can be defined as follows: L(Y,Z) is the payoff or loss for player A when A chooses Y and B chooses Z. If A wins, then his payoff is L(Y,Z). If A loses, his loss is L(Y,Z). A classifer which minimizes loss in making the guess is a Bayes classifer. The decision process is considered a two person zero sum game. In this case, the person providing the samples selects at random rather than trying to optimize strategy.

In a two person game between nature and a classifier, nature selects a class ω_i and from this class produces a vector $\overline{x}$. If the classifier incorrectly selects ω_j as the class from which $\overline{x}$ originated then the classifier has a loss L_{ij}. The expected risk in any classification may be defined as

$$r_j(\overline{x}) = \sum_{i=1}^{M} L_{ij}\, P(\omega_i/\overline{x}) \tag{2}$$

M is the total number of possible pattern classes and $P(\omega_i/\overline{x})$ is the likelihood or probability that $\overline{x}$ came from

ω_i. The Bayes classifier formula is given by

$$P(\omega_i/\overline{x}) = \frac{P(\omega_i)P(\overline{x}/\omega_i)}{P(\overline{x})} \qquad (3)$$

where $P(\overline{x}/\omega_i)$ is the probability function of class ω_i and $P(\omega_i)$ is the a priori probability of occurence of class ω_i. [3] The risk function may now be rewritten as

$$r_j(\overline{x}) = \sum_{i=1}^{M} L_{ij} P(\overline{x}/\omega_i)P(\omega_i) \qquad (4)$$

with the classifier now assigning the pattern to the class with the lowest risk.

Many other methods are possible which are based on the same probability functions. They differ in the forms of threshholds used in making the final classification.

Distance Measurements

One method of solving the pattern recognition problem is to measure the distance between the sample and the pattern class. The sample is then classified to the class which has the minimum distance from the sample. In the case where the patterns can be treated as vectors in an orthogonal space the measure of the distance can be found by the Euclidean distance

$$D_e = \left[(x_1-\omega_1)^2+(x_2-\omega_2)^2+\ldots+(x_n-\omega_n)^2\right]^{\frac{1}{2}} \quad (5)$$

where $x_1, x_2, \ldots x_n$ are elements of the sample vector and $\omega_1, \omega_2, \ldots \omega_n$ are elements of a pattern $\bar{\omega}$. This can be rewritten in vector form as

$$D_{euclid} = \left[(\bar{x}-\bar{\omega})^T(\bar{x}-\bar{\omega})\right]^{\frac{1}{2}} \quad . \quad (6)$$

A more general distance measure is defined as

$$D_{minkowski} = \left[\left|x_1-\omega_1\right|^m+\left|x_2-\omega_2\right|^m+\ldots+\left|x_n-\omega_n\right|^m\right]^{1/m} \quad (7)$$

where m is an integer ≥ 1. [4]

In the case where the pattern classes are not roughly hyperspherical in shape or are not well seperated other distance measure techniques are needed. The Mahalanobis distance may be effective when statistical properties are being used. This measure is defined as

$$D_M = (\bar{x}-\bar{m})^T \underline{\underline{C}}^{-1}(\bar{x}-\bar{m}) \quad (8)$$

where vector $\bar{m}$ is defined as a mean vector

$$\bar{m} = \frac{1}{J}\sum_{k=1}^{J} \bar{x}_k \quad (9)$$

and $\underline{\underline{C}}$ is the covariance matrix estimated by

$$\underline{\underline{C}} = \frac{1}{J} \sum_{k=1}^{J} (\bar{x}_k \bar{x}_k^T - \bar{m}_k \bar{m}_k^T) \tag{10}$$

where $\bar{x}_k$ are training set samples and J is the number of training samples. [3]

The Tanimoto measure is useful in problems dealing with binary vectors. It is defined by

$$D_T = \frac{\bar{x}^T \bar{\omega}}{\bar{x}^T \bar{x} + \bar{\omega}^T \bar{\omega} - \bar{x}^T \bar{\omega}} \ . \tag{11}$$

The quantity $\bar{x}^T \bar{\omega}$ is the number of common attributes between the sample $\bar{x}$ and the pattern protype $\bar{\omega}$.

Discriminant Functions

Strongly related to the distance methods are the discriminant functions. These functions act as partitions between the pattern clusters. The pattern sample is classified depending on which side of the partition the sample falls. The most efficient of the functions is the linear discriminant function or hyperplanes.

To create general hyperplane functions, the pattern vector is augmented in the following way

$$\bar{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ 1 \end{bmatrix} \tag{12}$$

The product of the sample vector and a weight vector forms the linear discriminant function $d(\bar{x})$.

$$d(\bar{x}) = \bar{w}^T\bar{x} \tag{13}$$

where $\bar{w} = \left[w_1, w_2, w_3, \ldots, w_n, w_{n+1}\right]^T$. In the two-class case, the sample is classified by

$$d(\bar{x}) = \bar{w}^T\bar{x} \begin{cases} >0, \text{ if } \bar{x} \in \omega_1 \\ <0, \text{ if } \bar{x} \in \omega_2 \end{cases} . \tag{14}$$

$d(\bar{x})$ represents a hyperplane which divides the pattern space into two parts with one cluster on each side of the partition. [5]

The two-class case is straightfoward. However, the multi-class problems may not be as simple. In the case of multiple clusters, linear combinations of the discriminant functions may be neccessary. This combination may result in indeterminant regions in the pattern space. Figure 4 shows the two-class case along with a multi-class case.

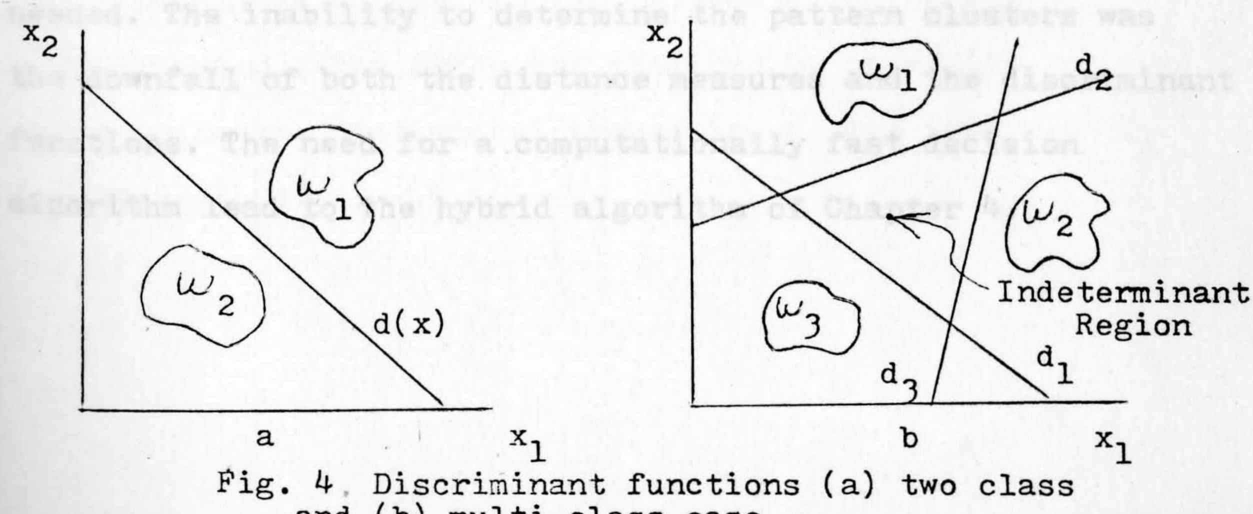


Fig. 4 Discriminant functions (a) two class and (b) multi-class case

The problem of indeterminant regions creates a need for higher-order functions to perform more complex decisions. The clustering properties of the sample sets of pattern classes dictates the need for these functions. With the higher-order functions, the computation involved in classification of the samples becomes much greater than with the linear functions.

Other Methods

The methods discussed so far in no way cover the entire range of pattern recognition techniques. A method which has received growing attention in the past few years is syntax analysis or the structural method. [10] Other methods have been developed and it is left for the interested reader to see references [3] - [7] for more details.

In the particular application of track timing these standard methods failed for a variety of reasons. The statistical methods involved tremendous computation time and the reliability of this method seemed to fall short of what was needed. The inability to determine the pattern clusters was the downfall of both the distance measures and the discriminant functions. The need for a computationally fast decision algorithm lead to the hybrid algorithm of Chapter 4.

CHAPTER IV

A HYBRID DECISION ALGORITHM

The types of pattern recognition problems handled in Chapter 3 may be stated as follows: Given a set of pattern classes and given a new sample, determine to which class the sample pattern belongs. In certain applications this problem is changed slightly to read; Given a set of candidate samples and given that one and only one of the samples belongs to a particular pattern class and the rest do not, determine which of the candidate samples belong to the pattern class.

The solution to this class of problems can be handled by a hybrid method, that is, the decision model makes use of both probabilistic and deterministic techniques. First it is necessary to develope a set of probability density functions (PDF). Each PDF will give the probability, x_i, that the sample belongs to the pattern class, given measurement m_i. The PDF's are obtained by carefully determining the statistical properties of the features in interest. [1]

Measurements are made on each candidate. The probabilities are obtained from

$$x_i = P(\omega/m_i) \quad . \tag{15}$$

The probabilities are arranged in a feature vector $\bar{x}$.

$$\bar{x} = \begin{bmatrix} x_1 \; x_2 \ldots x_n \end{bmatrix}^T \tag{16}$$

where n is the number of features.

If J candidates are to be used as samples, then J probability feature vectors must be found. These vectors are arranged in a $\underline{\underline{C}}$ matrix called the candidate matrix.

$$\underline{\underline{C}} = \begin{bmatrix} \bar{x}_1^T \\ \bar{x}_2^T \\ \vdots \\ \bar{x}_J^T \end{bmatrix} \tag{17}$$

where J is the number of candidates and $\bar{x}_j$ is given by (2).

To indicate the relative importance of each feature in determining the class membership, a weight vector $\bar{W}$ is formed. The weight vector postmultiplies $\underline{\underline{C}}$ to yield an evaluation vector $\bar{E}$.

$$\underline{\underline{C}} \, \bar{W} = \bar{E} \tag{18}$$

The $\bar{E}$ vector represents the relative probability that the corresponding candidate is a member of ω. By selecting the candidate for which e_k is a maximum a decision can be made about the membership of the k^{th} candidate in ω.

$$\text{If } e_k > e_i \quad i \neq k \quad 1 \leq i,\ k \leq J \qquad (19)$$
$$\text{then } \overline{x}_k \text{ is a member of } \omega.$$

(18) represents a system with more equations than unknowns. An exact solution therefore, cannot be expected. In order to solve for the values of the weights it is necessary to use a least-squared-error approach. An error signal $\overline{r}$ can be defined as

$$\overline{r} = \overline{E} - \underline{\underline{C}}\,\overline{W} \quad . \qquad (20)$$

with $\overline{E}$ defined as a constant by

$$e_i = 1 \quad \text{if } \overline{x}_i \in \omega$$
$$e_i = -1 \quad \text{if } \overline{x}_i \notin \omega \qquad (21)$$

That is in the ideal case, any candidate $\overline{x}_i$ that is a member of the ideal pattern class ω shall have a corresponding e_i equal to 1 and any candidate that is not a member shall have an e_i equal to -1.

Squaring (20) gives

$$\overline{r}^T\overline{r} = \overline{E}^T\overline{E} - \overline{E}^T\underline{\underline{C}}\overline{W} - \overline{W}^T\underline{\underline{C}}^T\overline{E} + \overline{W}^T\underline{\underline{C}}^T\underline{\underline{C}}\overline{W} \qquad (22)$$

Taking the derivative with respect to $\overline{W}$

$$\nabla_{\overline{W}}\,\overline{r}^T\overline{r} = 0 - \underline{\underline{C}}^T\overline{E} - \underline{\underline{C}}^T\overline{E} + \underline{\underline{C}}^T\underline{\underline{C}}\overline{W} + \underline{\underline{C}}^T\underline{\underline{C}}\,\overline{W} \qquad (23)$$

Setting $\nabla_{\overline{W}} \overline{r}^T \overline{r} = 0$

$$0 = -2\underline{\underline{C}}^T \overline{E} + 2\underline{\underline{C}}^T \underline{\underline{C}}\, \overline{W} \tag{24}$$

and solving for $\overline{W}$ gives

$$\overline{W} = (\underline{\underline{C}}^T \underline{\underline{C}})^{-1} \underline{\underline{C}}^T \overline{E} \quad . \tag{25}$$

This equation can also be determined geometrically. [9]

Equation (25) now allows the direct calculation of the weights from a set of training samples. Once the PDF's are found and the weights calculated, the evaluation of the $\overline{E}$ vector and the subsequent classification of the candidates can be accomplished in a straightfoward manner. [1]

It can be seen from (25) that a new matrix multiply and inversion is needed every time the weights are to be updated. For a large training set these calculations become time consuming. The form of the equations allows for an interesting training procedure. Defining $\overline{P}_j$ as

$$\overline{P}_j = (\underline{\underline{C}}^T \underline{\underline{C}})^{-1} \tag{26}$$

and

$$\overline{P}_{j+1} = \left[\underline{\underline{C}}^T \underline{\underline{C}} + \overline{x}_{j+1} \overline{x}^T_{j+1}\right]^{-1} \tag{27}$$

a simpler approach can be devised. Using (26) and (27) the updated version of $\overline{W}$ may be written as

$$\overline{W}_{j+1} = \overline{W}_j + \overline{P}_j \overline{x}_{j+1} \left[\overline{x}^T_{j+1} \overline{P}_j \overline{x}_{j+1} + 1\right]^{-1} (e_{j+1} - \overline{x}^T_{j+1} \overline{W}_j) \tag{28}$$

Since $\left[\bar{x}^T_{j+1}\bar{P}_j\bar{x}_{j+1} + 1\right]$ is a scalar the inversion is a scalar inversion not a matrix inversion. This allows very simple computations to update $\bar{W}$.

It is now possible to take a small set of candidates and find the initial matrix inverse. Once accomplished, a large data base may be used to train the weights without the need for large, time consuming matrix operations. The interested reader should see references [1] and [9] for more details.

CHAPTER V

THE DECISION ALGORITHM APPLIED TO TRACK TIMING

If the runner is viewed from overhead as he approaches the finish line, it is possible to draw a line through the body of the runner parallel to the finish line that corresponds to the chestline. If a series of lines are drawn, the body is segmented in such a way that a series of body segments are produced. (figure 5) Since a runner can

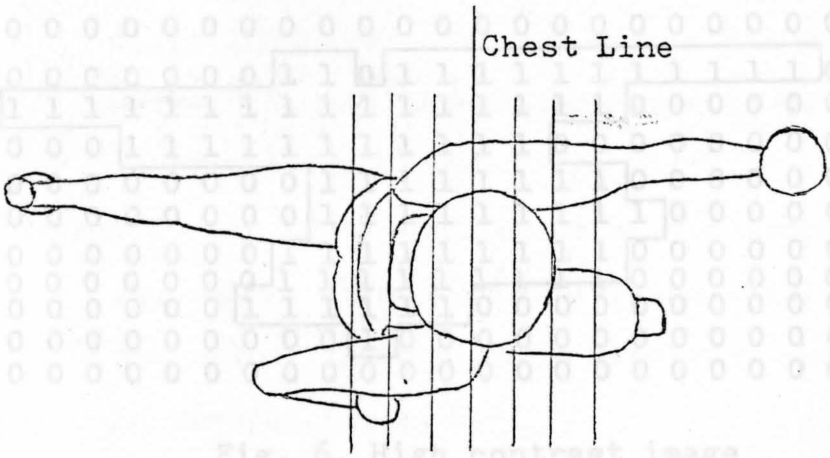


Fig. 5. Segmented body

have one and only one chestline, the problem becomes one of identifying the chestline from the set of body segments. This is very similar to the problem handled by the decision algorithm in Chapter 4.

To implement the algorithm it is necessary to select a set of features which will allow the decision model to classify the candidates as either chestlines or nonchestlines. Rather than work with a multiple gray level image, a high contrast image (figure 6) was generated with the body of the runner classified as a digital "1" and the background classified as a digital "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 0 0 0 0 0 0 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 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 0 0 0 0 0 0 0 0 0 0
```

Fig. 6. High contrast image

The initial selection of the features must be done on a trial and error basis. Five features were selected after considerable testing (Appendix B) as providing sufficient information to classify the candidates. They are:

Magnitude (M) -- The total length of the body segment. This corresponds to the width of the runner at that scan.

Delta Magnitude (dM) -- The change in magnitude between two consecutive body segments.

Total Trapped Zeros (Z) -- Using a high contrast image (two gray levels) it is possible to define the runner's body as a digital "1" and the background as a digital "0". A set of zeros with ones on both sides indicated the scan was in the region about the neck. This was later determined to be a consistantly important feature.

Total Change (Tot) -- The sum of the absolute value of the change in magnitude and the absolute value of the sum of the change in magnitude and the change in trapped zeros.

Distance From End Of Body (Beta) -- The shape of the runner's body was found to be roughly diamond like when viewed from overhead. It was also known that once the body began to narrow the chest line had already been scanned. It was therefore convenient to define a practical end of the body. The end of the body was defined syntactically as the candidate following 3 consecutive decreases in magnitude or a single decrease of 3 units or more in any one candidate. The distance from the practical end of the body to the candidate in question was defined as beta.

These features are further illustrated in figure 7.

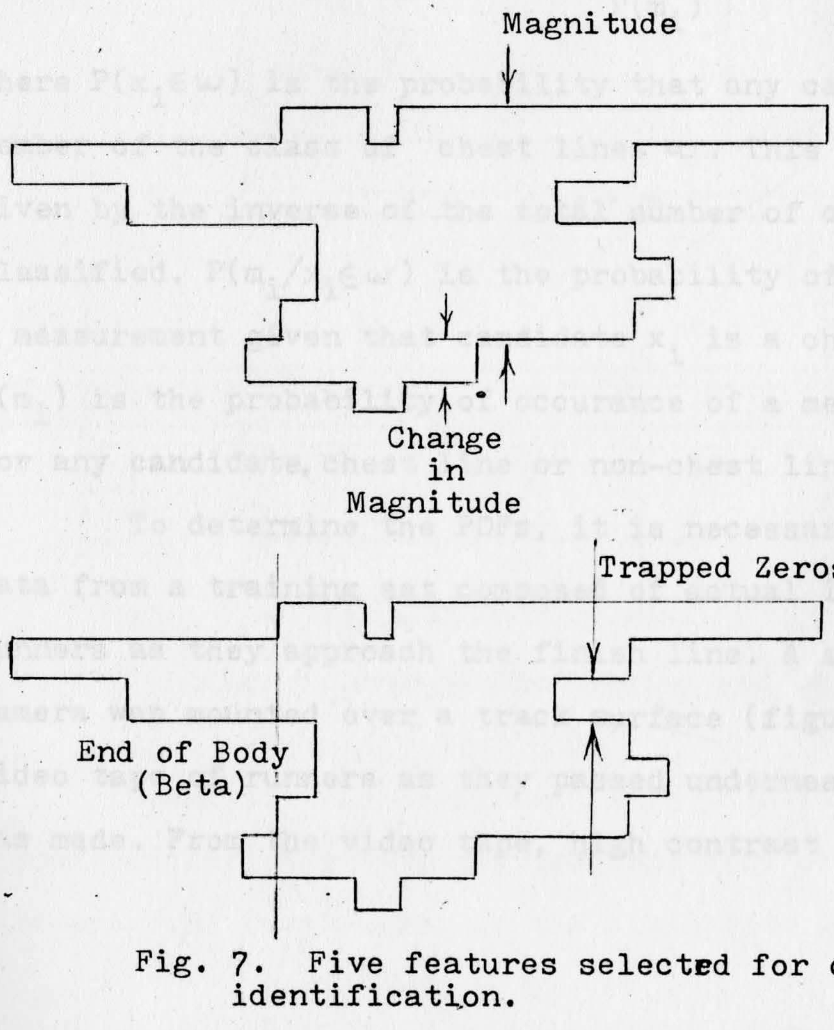


Fig. 7. Five features selected for chest line identification.

The probability that the candidate x_i is a member of the class of chest lines ω given measurement m_i is given as the definition of the probability density functions. This can be written as

$$P(x_i \in \omega / m_i) \tag{29}$$

Using the Bayes Formula as given in (3) and rewriting, the PDFs may be expanded into known terms.

$$P(x_i \in \omega / m_i) = \frac{P(x_i \in \omega) P(m_i / x_i \in \omega)}{P(m_i)} \tag{30}$$

where $P(x_i \in \omega)$ is the probability that any candidate is a member of the class of chest lines ω. This is a constant given by the inverse of the total number of candidates to be classified. $P(m_i / x_i \in \omega)$ is the probability of occurance of a measurement given that candidate x_i is a chest line. $P(m_i)$ is the probability of occurance of a measurement for any candidate, chest line or non-chest line.

To determine the PDFs, it is necessary to extract data from a training set composed of actual images of runners as they approach the finish line. A standard video camera was mounted over a track surface (figure 8) and a video tape of runners as they passed underneath the camera was made. From the video tape, high contrast images were created by hand one frame at a time. These images were arranged into 35 x 25 matrices (Appendix C) and placed in the IBM 370 computer.

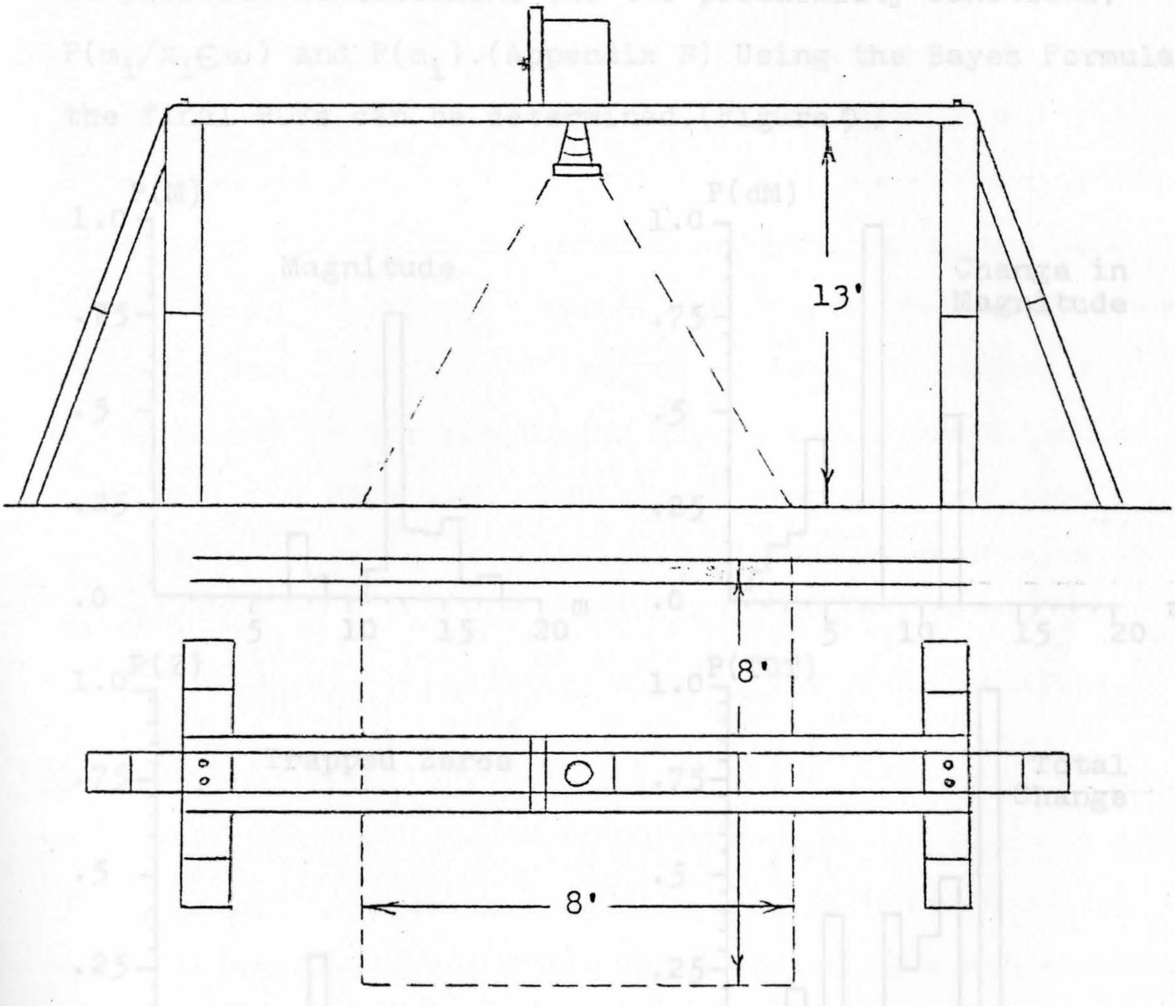


Fig. 8. Camera scaffolding and scanning area

To extract the measurements for the five features defined earlier, two computer programs were created. (Appendix D) These programs later serve as the simulation model for the circuitry used to extract data in the total system simulation. With the feature measurements now available, it is possible to determine the two probability functions, $P(m_i/x_i \in \omega)$ and $P(m_i)$.(Appendix F) Using the Bayes Formula the final PDFs can be determined.(Figure 9)

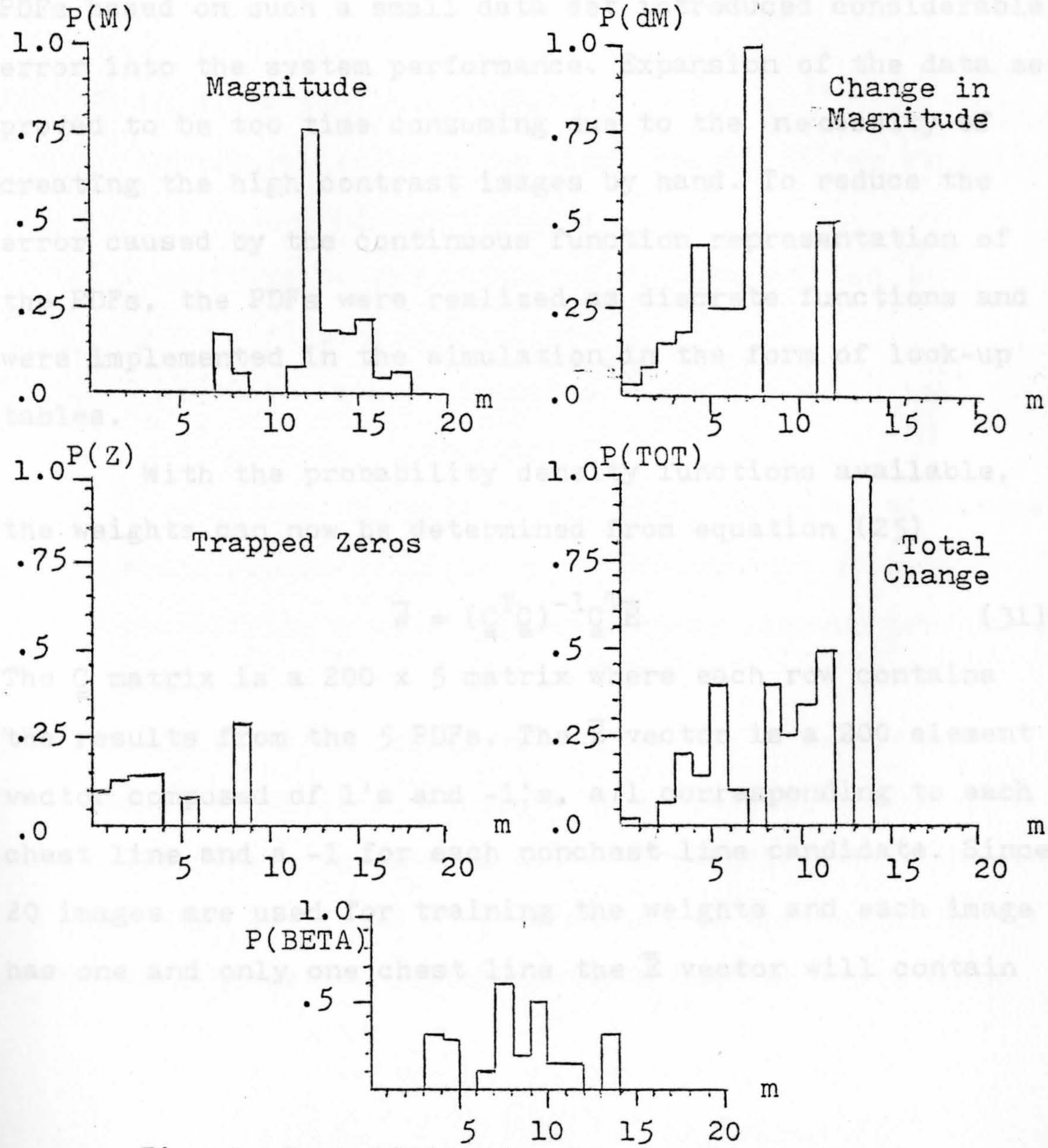


Fig. 9. Probability density functions for five features

The actual PDFs were determined from a data base composed of 20 images of runners. Feature measurements were made on the 9 nearest candidates to the chest line and the chest line itself. This provided 200 data points for the PDFs. It was found that determining continuous functions for the PDFs based on such a small data set introduced considerable error into the system performance. Expansion of the data set proved to be too time consuming due to the necessity of creating the high contrast images by hand. To reduce the error caused by the continuous function representation of the PDFs, the PDFs were realized as discrete functions and were implemented in the simulation in the form of look-up tables.

With the probability density functions available, the weights can now be determined from equation (25)

$$\overline{W} = (\underline{\underline{C}}^T\underline{\underline{C}})^{-1}\underline{\underline{C}}^T\overline{E} \tag{31}$$

The $\underline{\underline{C}}$ matrix is a 200 x 5 matrix where each row contains the results from the 5 PDFs. The $\overline{E}$ vector is a 200 element vector composed of 1's and -1's, a 1 corresponding to each chest line and a -1 for each nonchest line candidate. Since 20 images are used for training the weights and each image has one and only one chest line the $\overline{E}$ vector will contain

20 1's and 180 -1's. The order in which the chest lines and non-chest lines are arranged in the $\underline{\underline{C}}$ matrix and $\overline{E}$ vector makes no difference in the calculations of the weights provided each chest line set of measurements in $\underline{\underline{C}}$ has a corresponding $\overline{E}$ vector entry of 1 and each non-chest line has an $\overline{E}$ entry of -1. The training of the weights was accomplished on the IBM 370 computer. The weight training program is presented in Appendix G.

Investigation into the significance of the weights lead to the following conclusions:

1. A positive weight indicates a feature which is a good chest line classifier.
2. A negative weight indicates a feature which is a good non-chest line classifier.
3. A weight which tends toward zero indicates a feature which provides little or no information to help in classifying a candidate as a chest line or non-chest line.

These conclusions are discussed further in Appendix H. As a result of these conclusions it is possible to use the weight training program as a means of eliminating features which provide little or no information in the classification of the candidates. Since the features must be selected on a hueristic basis, this type of goodness test on a feature is necessary. Removal of a feature with a zero weight does not effect the performance of the algorithm but does reduce the size of the matrix operations performed in the calculation of the $\overline{E}$ vector. Since the original intent was to implement the algorithm in real time on a microprocessor, the size of the matrix operations is a critical factor. Minimal threshholds necessary for acceptance or rejection of a feature cannot be generalized but must be determined in terms of the algorithm"s application.

CHAPTER VI

ALGORITHM SIMULATION AND TESTING

In order to verify the decision algorithm's ability to classify chest lines, a simulation of the algorithm was performed on the IBM 370 computer. The simulation program accepted as input the high contrast images found in Appendix C. The necessary feature measurements were made and the $\underline{\underline{C}}$ matrix was formed from the discrete PDFs. The algorithm generated an $\overline{E}$ vector based on the $\underline{\underline{C}}$ matrix and the largest element in $\overline{E}$ was selected as the predicted chest line. The simulation program appears in Appendix I.

Noise introduced into the system due to the sampling rate prohibited the algorithm from exactly identifying the chest line in every image. The size of the data set used in generating the PDFs and in training the weights also added to the problem. It was determined, however, that in the particular application of track event timing it was not necessary to exactly identify the chest line to maintain a timing accuracy of .01 seconds.(Appendix J) In a worst case example (runners competing in the 100yd. dash) it was only necessary to be within $\pm$ 4 inches of the actual chest line. This corresponds to identifying the predicted chest line in a 5 candidate bandwidth centered about the actual chest line.(figure 10)

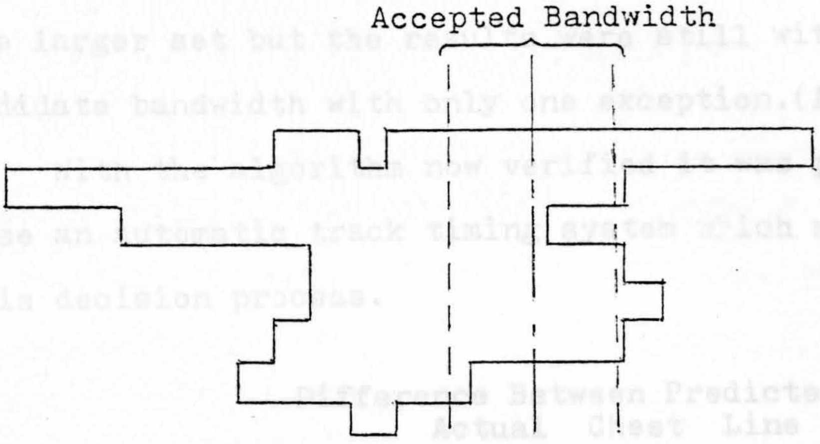


Fig. 10. Acceptable bandwidth of candidates

With this bandwidth as an acceptable tolerance, the algorithm correctly classified the images in every case. (figure 11) The algorithm ideally classified the chest line (that is the predicted chest line and the actual chest line were the same) in 13 out of 20 images and missed by 1 candidate in 5 out of 20 images.

This testing procedure verified the decision algorithm by testing it on the same images the weights were trained on. The results, therefore, were biased in favor of the algorithm. To further establish the credibility of the system 10 images were selected at random and the weights trained on these images. The algorithm was then tested on the entire set of images. The results indicated that the smaller data set could not pick the chest line as well as the larger set but the results were still within the 5 candidate bandwidth with only one exception.(figure 11)

With the algorithm now verified it was possible to propose an automatic track timing system which makes use of this decision process.

Image No.	Difference Between Predicted and Actual Chest Line: Simulation Using 20-Image Weights	Simulation Using 10-Image Weights	
1	0	0	*
2	-1	-1	*
3	0	0	*
4	0	0	*
5	0	0	*
6	-1	-2	*
7	0	0	*
8	0	0	*
9	-1	-1	*
10	0	0	*
11	0	0	
12	0	0	
13	0	-2	
14	-1	-1	
15	0	0	
16	2	3	**
17	2	1	
18	-1	-1	
19	0	0	
20	0	0	

* Images used in training weights
** Predicted chestline outside of tolerance

Fig. 11 Simulation Results

CHAPTER VII

A MICROPROCESSOR BASED AUTOMATIC TIMING SYSTEM

A system which makes use of the decision algorithm is shown in figure 12. This system provides both the precision and reliability needed in track timing. Through the use of the microprocessor and standard video camera it is possible to avoid large price tags on the system.

The camera is mounted over the track and scans an area beginning 8 feet before the finish line. It is oriented so the scan is parallel to the finish line. This, in a sense, slices the body into a series of segments where one of the segments corresponds to the chest line. These segments are regrouped and processed into the high contrast image from which the feature measurements are made. From these measurements, the decision algorithm will determine which body segment corresponds to the chest line.

The decision algorithm provides only the location of the chestline with respect to the finish line. The actual finish time would be calculated by the microprocessor in the following way

1. Locate the chest line with respect to the finish line and record the elapsed time since the start of the race.

2. Repeat Step 1 to obtain a second data point of time and location.
3. Knowing the elapsed time and the distance traveled, determine the average velocity.
4. If the location of the runner is determined close to the finish line, he will be unable to change his velocity significantly before the end of the race. Using his average velocity, it is possible to determine the time it will take to finish the race and from this information, the final finish time is obtained.

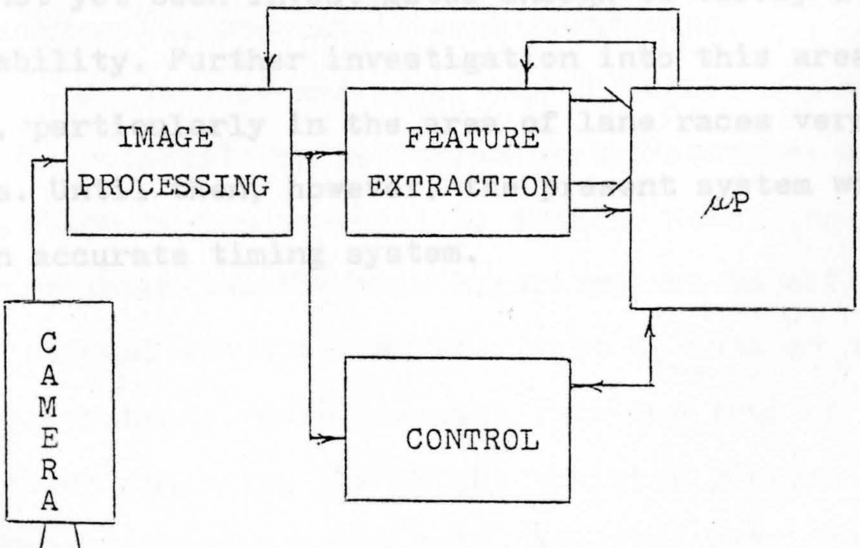


Fig. 12. System Block Diagram

The system will always provide a finish time because the algorithm will always predict a chest line. However, some images are more difficult to classify than others. The camera frame rate provides six complete images of a runner in the worst case. Since only two images are needed to calculate the finish time, it allows the system to throw out images which are difficult to classify.

The combination of video camera, microprocessor and interfacing logic provides a real time, microprocessor based automatic timing system. [9] Through the use of the microprocessor it is possible to provide an accurate and reliable system at a cost affordable by most high schools and small colleges. The system will provide accurate times of runners in any type of race. The ability of the system to place runners has not yet been investigated enough to verify its reliability. Further investigation into this area should be done, particularly in the area of lane races versus non-lane races. Until then, however, the present system will serve as an accurate timing system.

CHAPTER VIII

CONCLUSION

The algorithm presented provides a simple yet powerful means of handling a variety of pattern recognition applications. It is capable of handling highly non-linear features with versatility and speed. The implementation of the algorithm may be accomplished in a variety of ways depending on the complexity of the application.

In regards to track event timing, the algorithm is capable of providing the necessary accuracy in the classification of the chestline within the specified tolerances. Implementation of the algorithm can be accomplished at low cost and in real time through the use of a microprocessor and intelligent programing.

Success of the algorithm in future applications depends primarily on selection of the features and proper generation of the probability density functions. Since feature selection for this algorithm, as in all other pattern recognition applications, must be done on a trial-and-error basis, more research into the area of feature selection employing the weight training program must be done. The necessity of a large data base also requires the investigate into a self adaptive system using the update scheme proposed in Chapter 3.

APPENDIX A

Track Timing Personnel

In order to maintain the needed precision with the hand-held stopwatch method of timing track events, it is necessary to have as many as 25 individuals at the finish line. This is due to the check and counter-eheck system used. That is, each duty of importance is performed by more than one individual with the results being compared after each race. A list of officials need for an 8-lane 100yd dash race is presented below.

Head Finish Judge

2 Assistant Judges for every place scored

1 Assistant Judge for every other place

Head Timer

2 Assistant Timers for every place scored

1 Assistant Timer for every other place

1 Substitute Timer

Finish Line Recorder

Wind Gauge Poerator

APPENDIX B

Feature Selection

Feature selection represents the biggest problem in the implementation of most pattern recognition algorithms. Proper selection of optimal features must be done on a trial and error basis. In the particular application of chest line identification it was possible to select hundreds of possible features. However, limited research time and the need for a computationally fast algorithm restricted the number of features which could be implemented into the system.

Six features originally were selected as containing enough information to classify the chest line of the runner. They were: Magnitude, Change in Magnitude, Trapped Zeros, Change in Trapped Zeros, Total Change and Distance from the End of the Body of the thesis. The change in trapped zeros was discovered to be linearly dependent with the other features, thus it provided no useful information. This problem of linear dependency is one which is often disguised by noise. It is important, however, to recognize these features as providing no information thus reducing the amount of computation needed to execute the algorithms.

Total change was originally defined as the absolute sum of the change in magnitude plus the change in trapped zeros. However, due to an error in programing the feature was redefined as given in the main body of the thesis. Upon testing, it was discovered this feature provided more information than when implemented with the original definition. In pattern recognition it must be understood that optimal features are often stumbled upon and must be obtained any way possible.

APPENDIX C

High Contrast Images

Multiple gray level images created a problem in extracting the feature measurements. Circuitry and simulation programs quickly blew out of proportion when attempts were made to work with the more complex patterns.

In order to simplify programs and computation, high contrast images were produced. The resulting images sacrificed detail for simplicity. 20 images were produced and are presented on the following pages. The images are oriented so the runner is moving left to right across the page. The body segment identified by the broken lines corresponds to the runner's chest line.

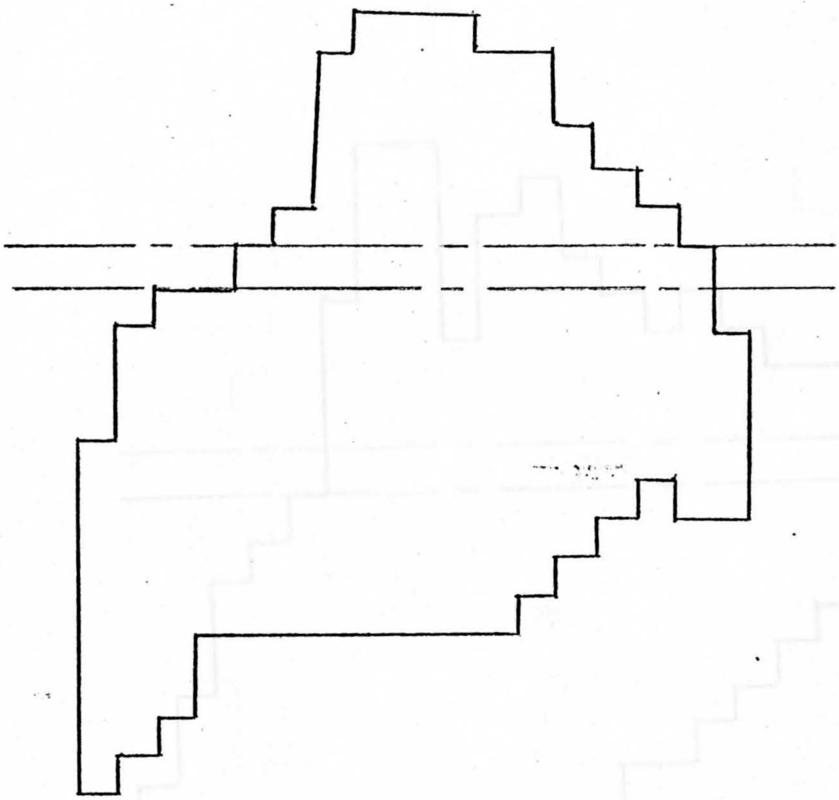


Image 1

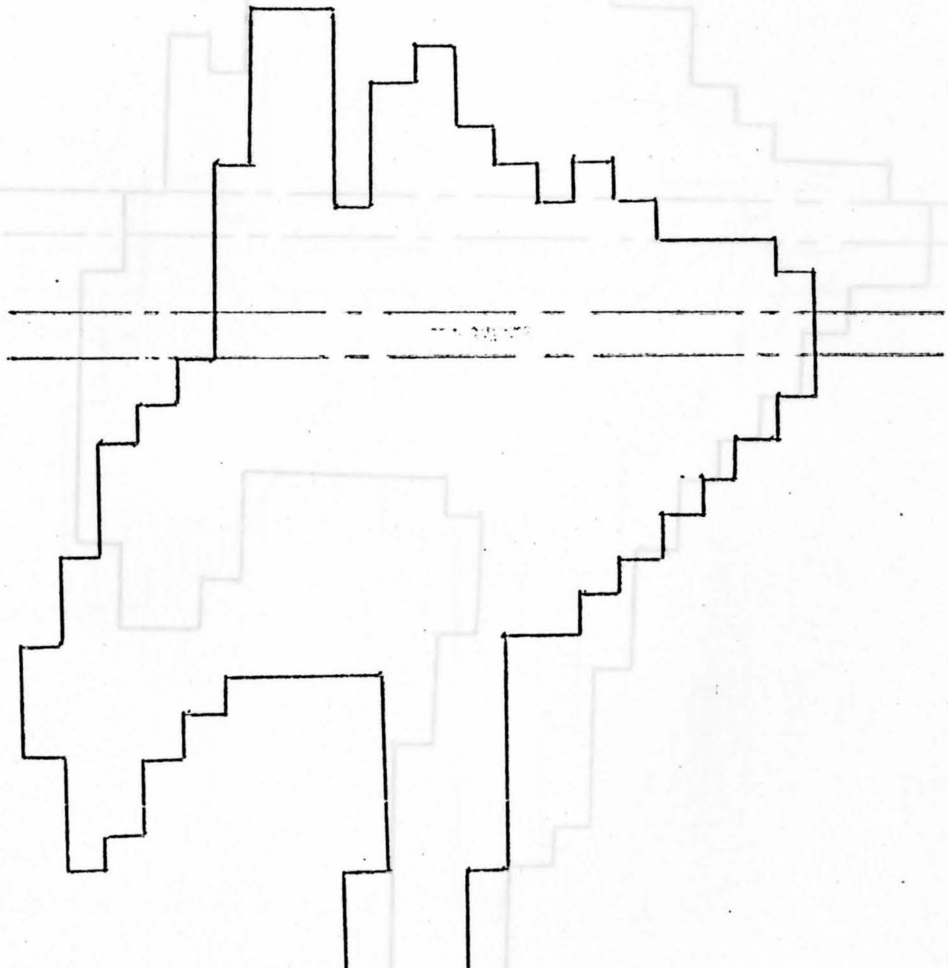


Image 2

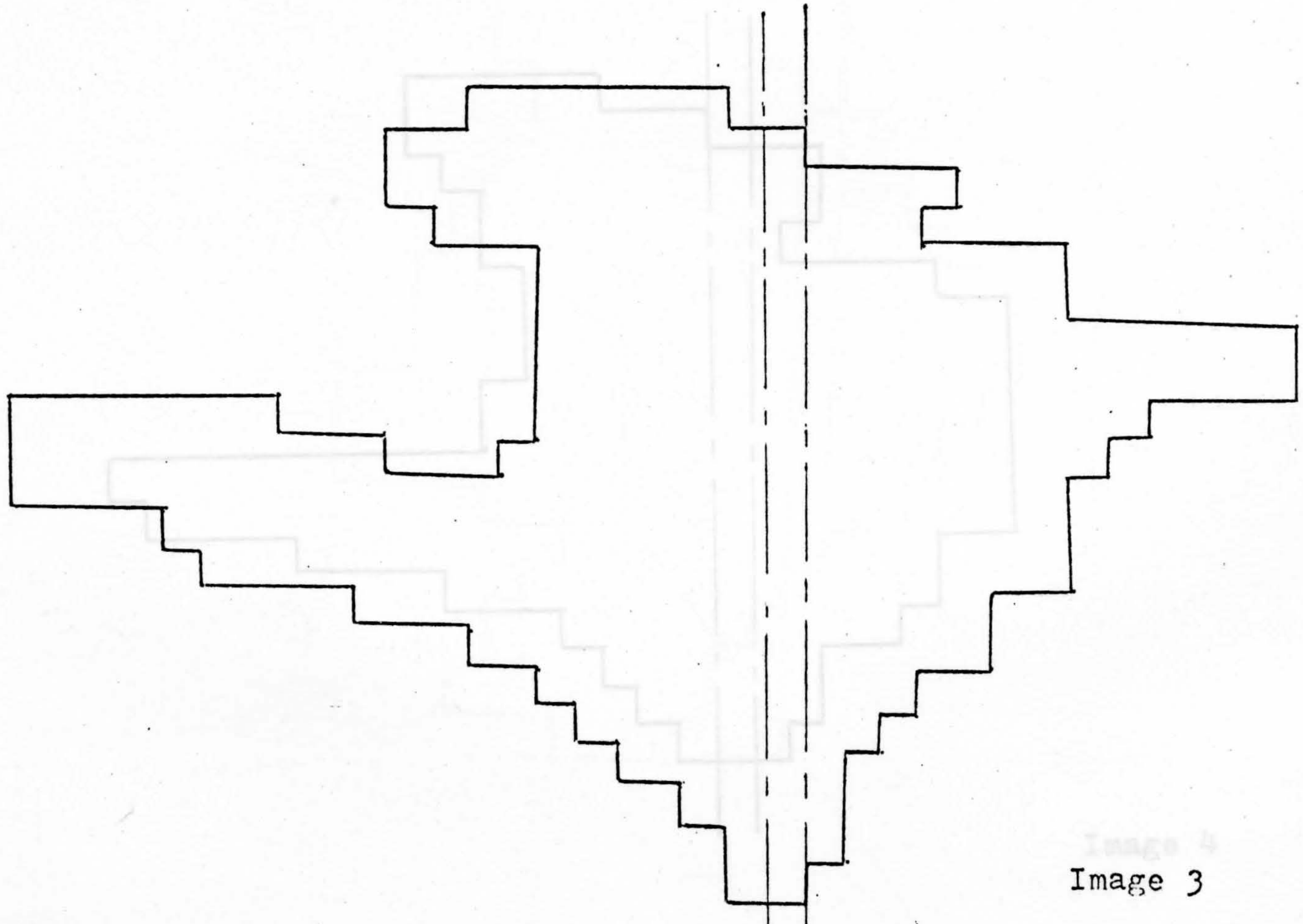


Image 3

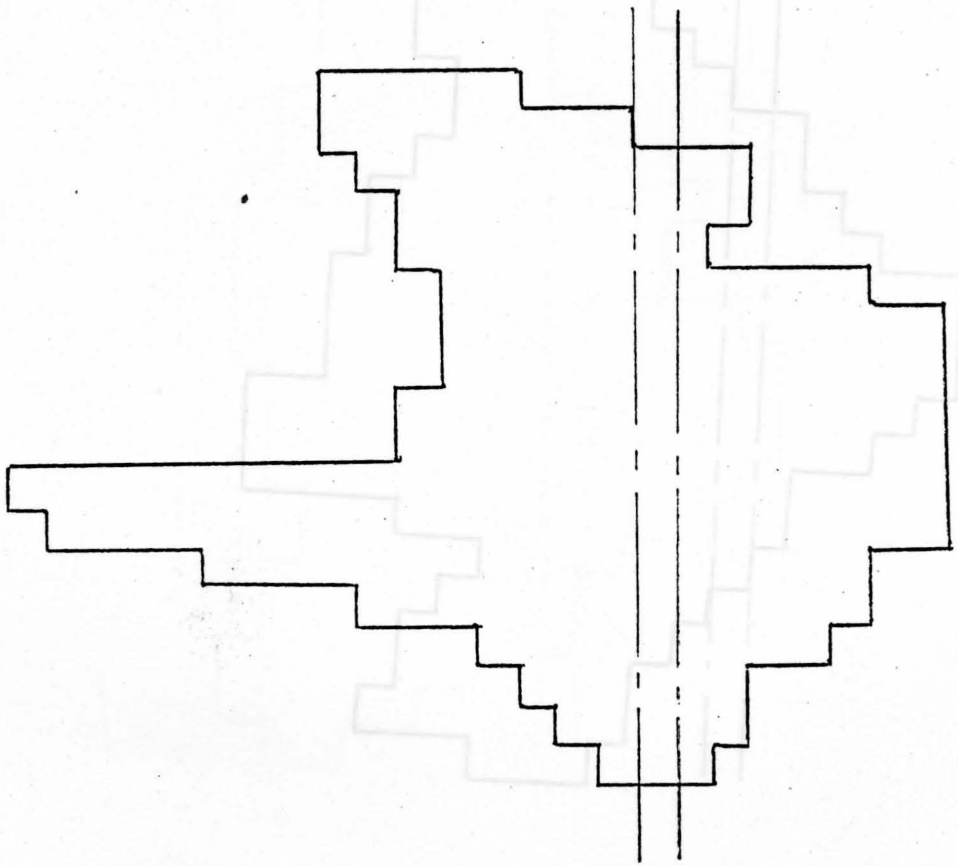


Image 4

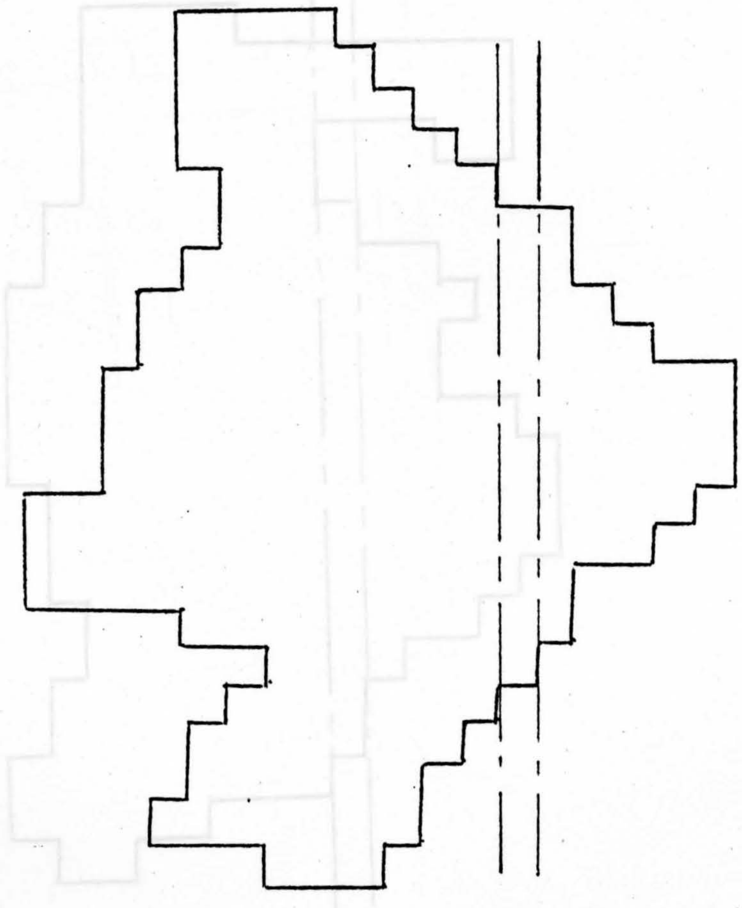


Image 5

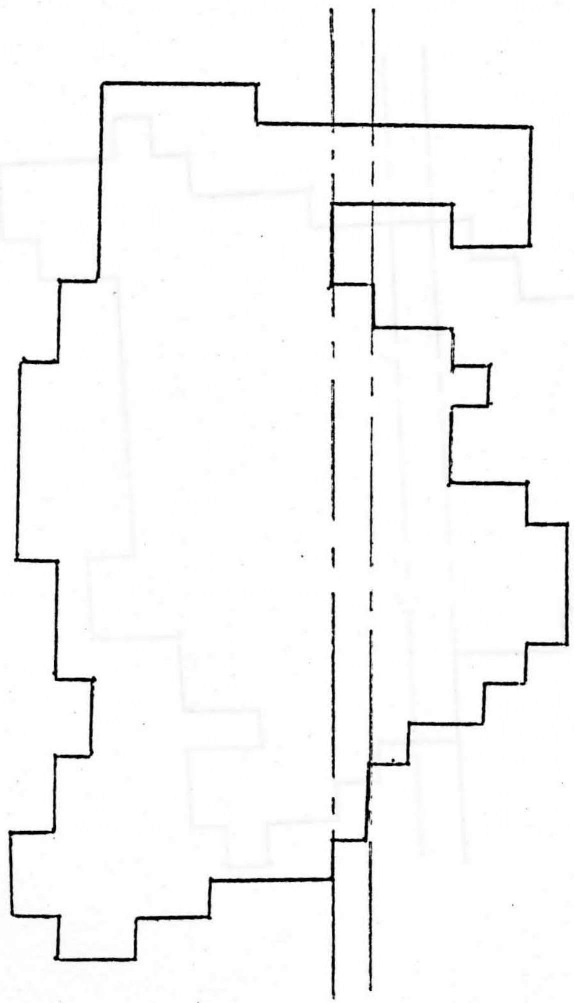


Image 6

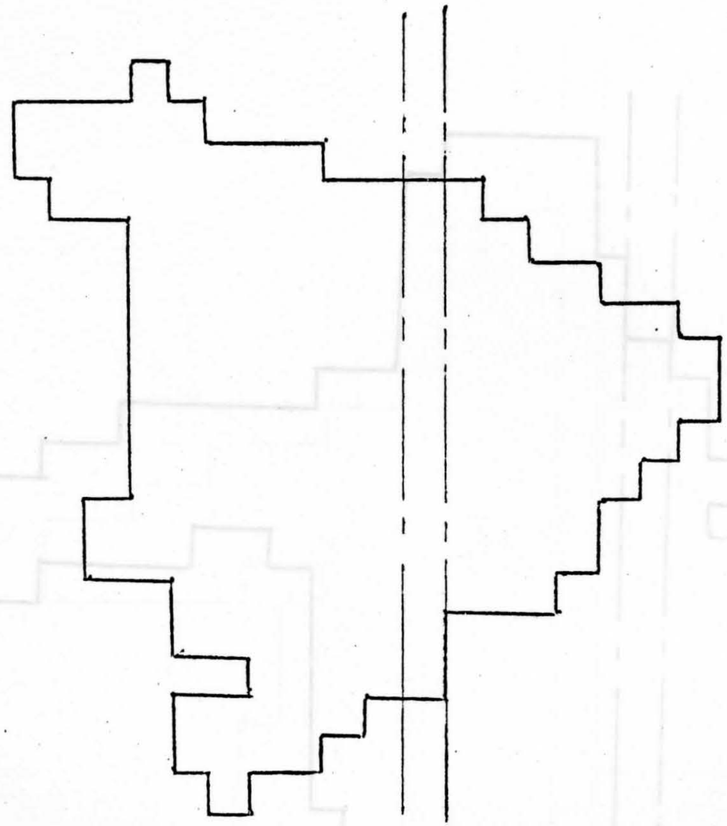


Image 7

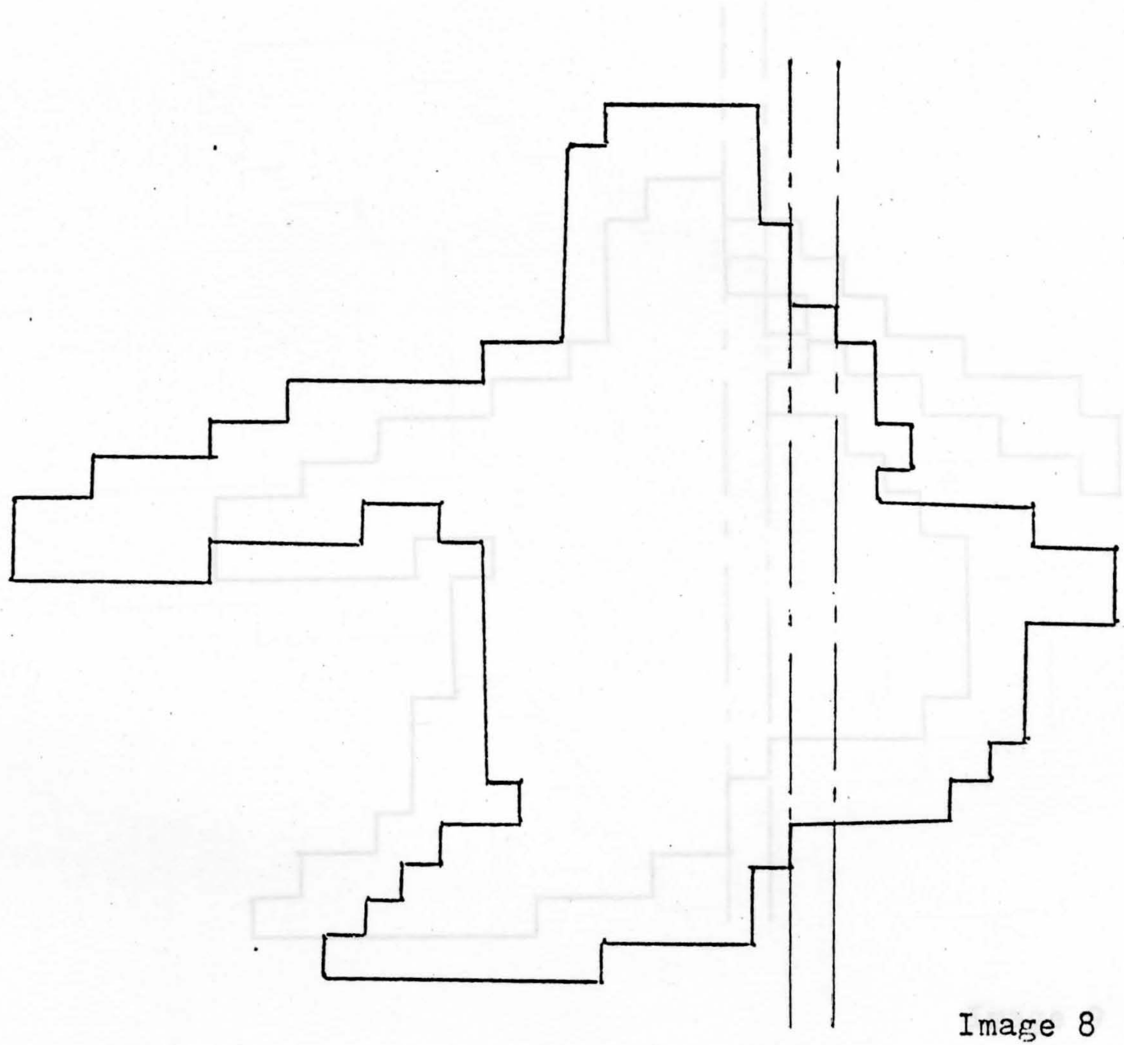


Image 8

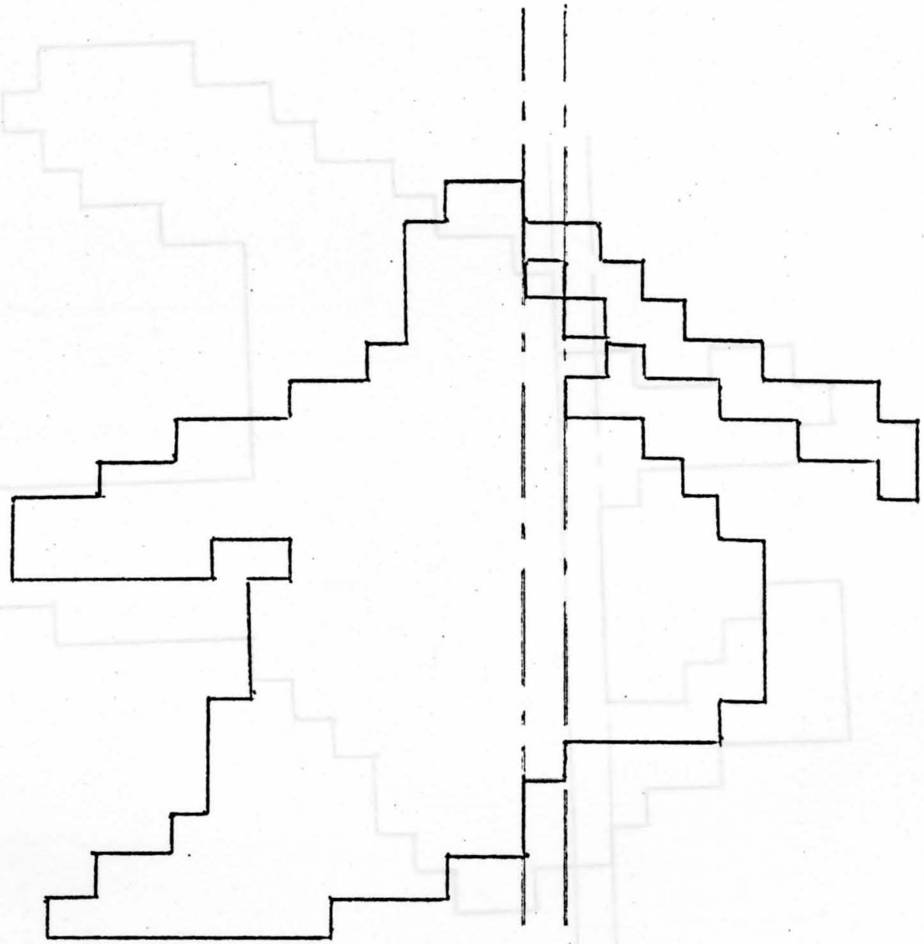


Image 9

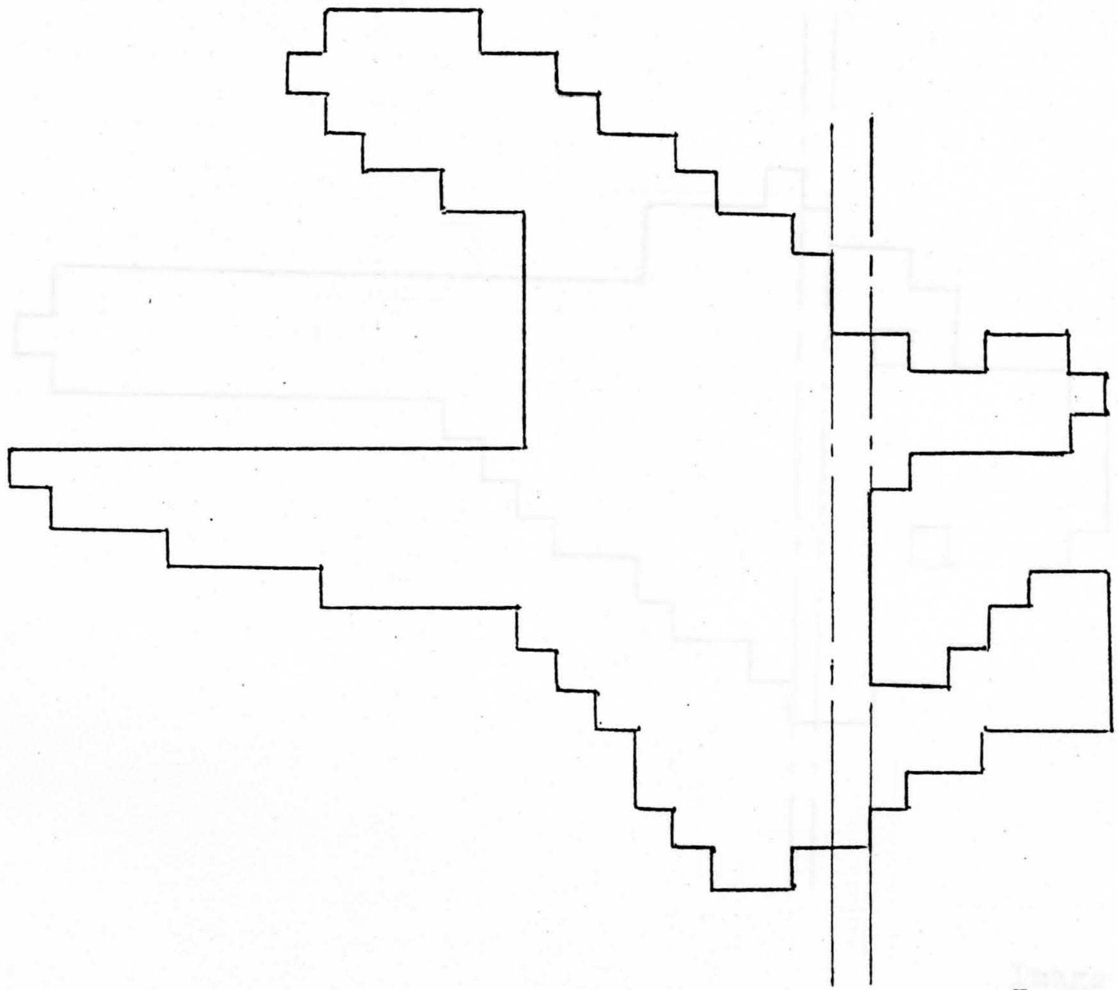


Image 10

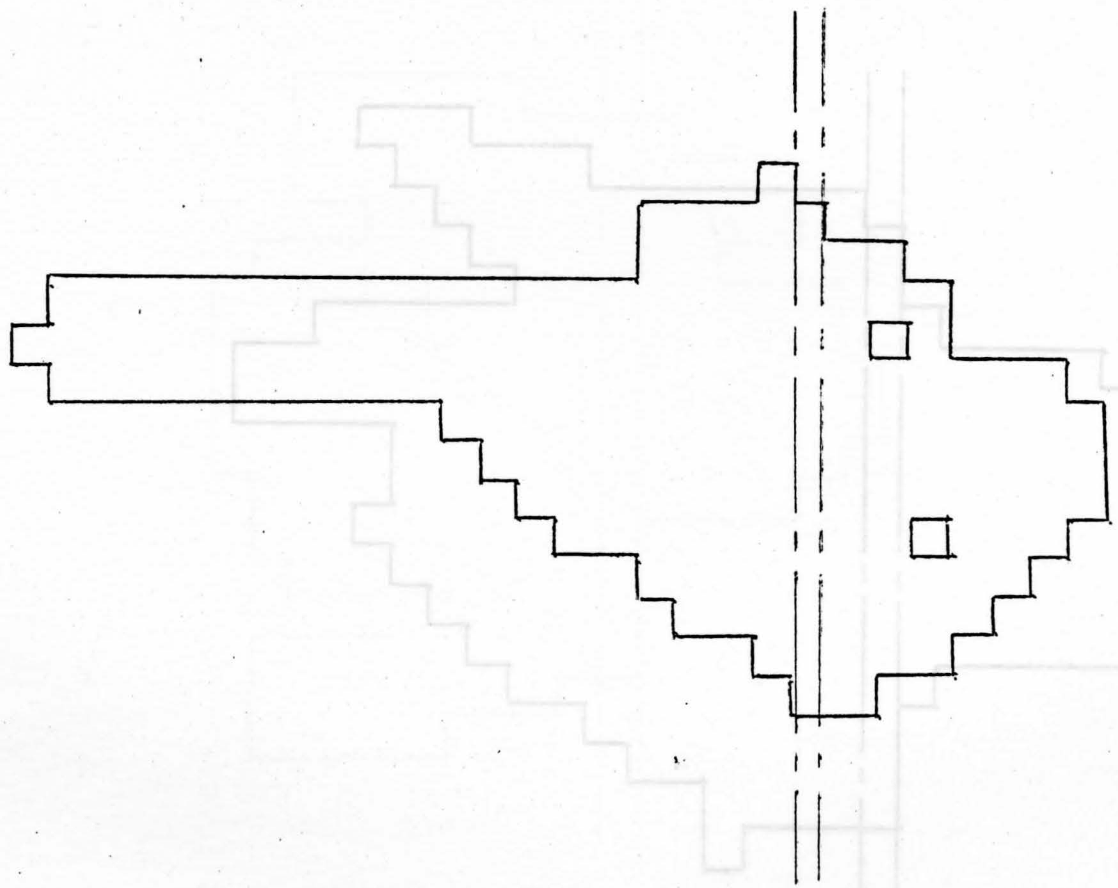


Image 11

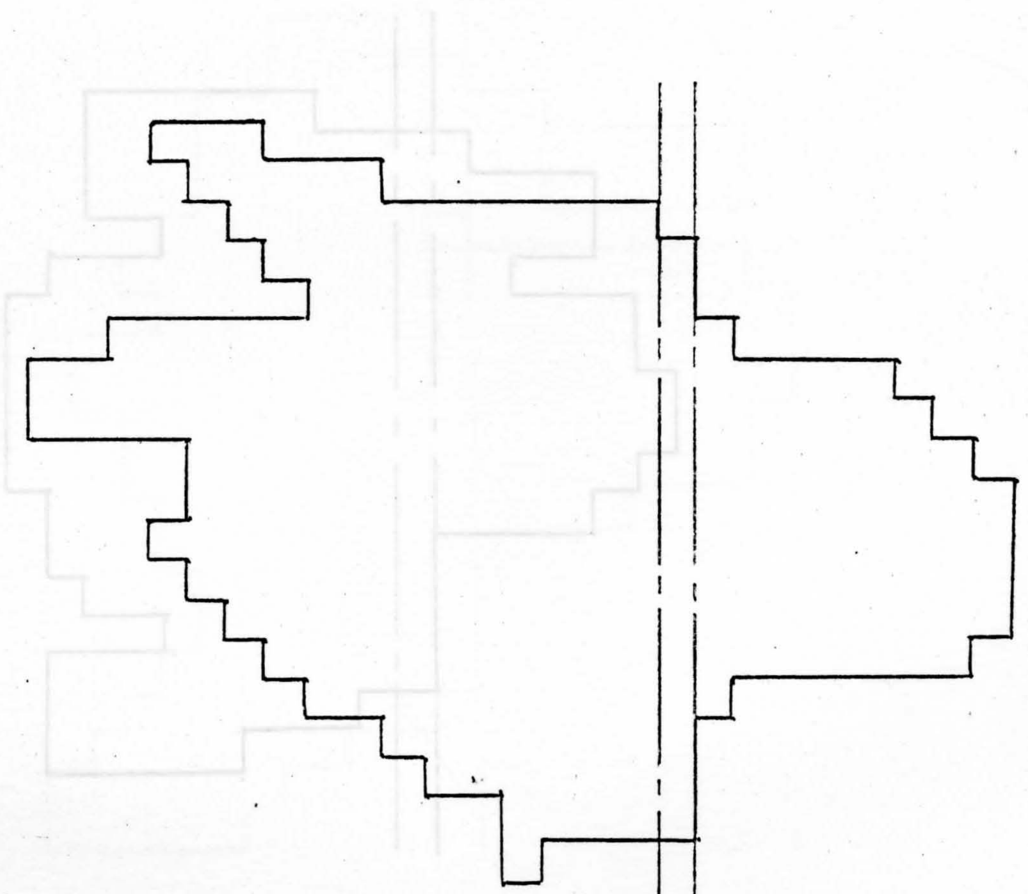


Image 12

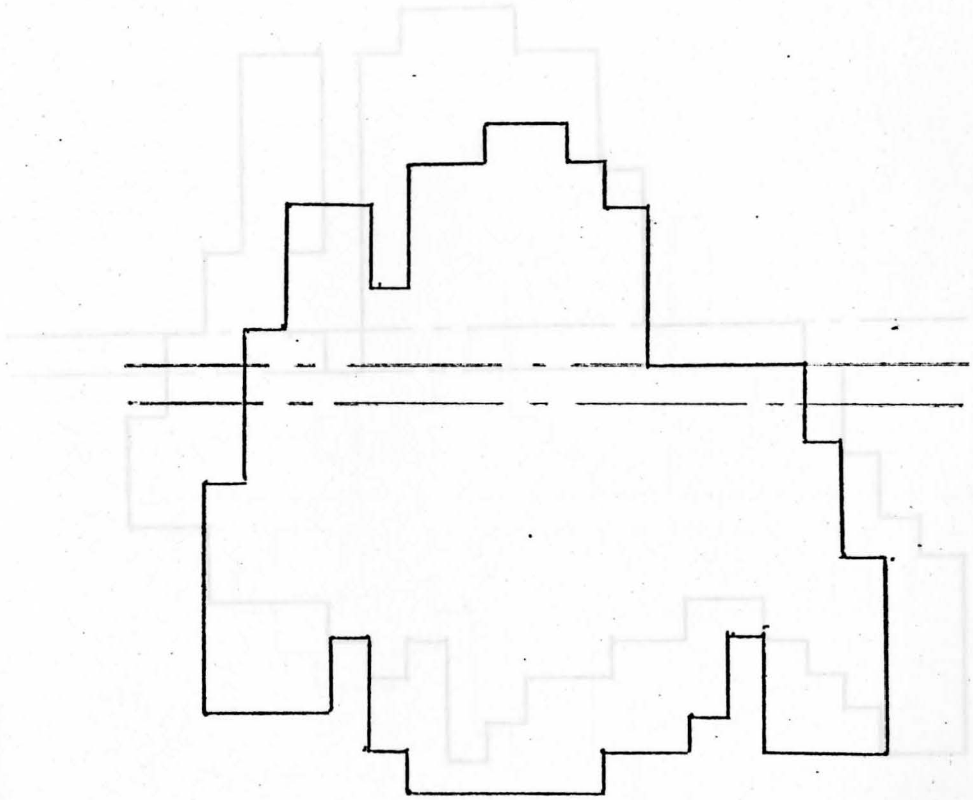


Image 13

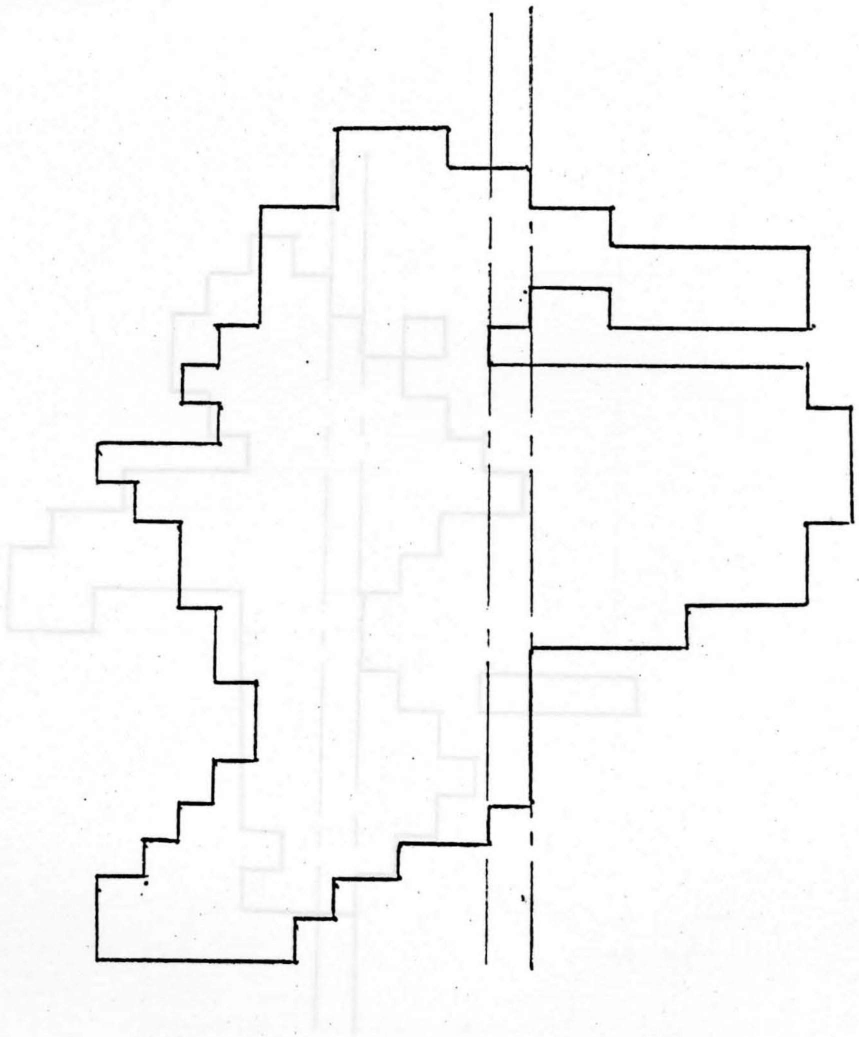


Image 14

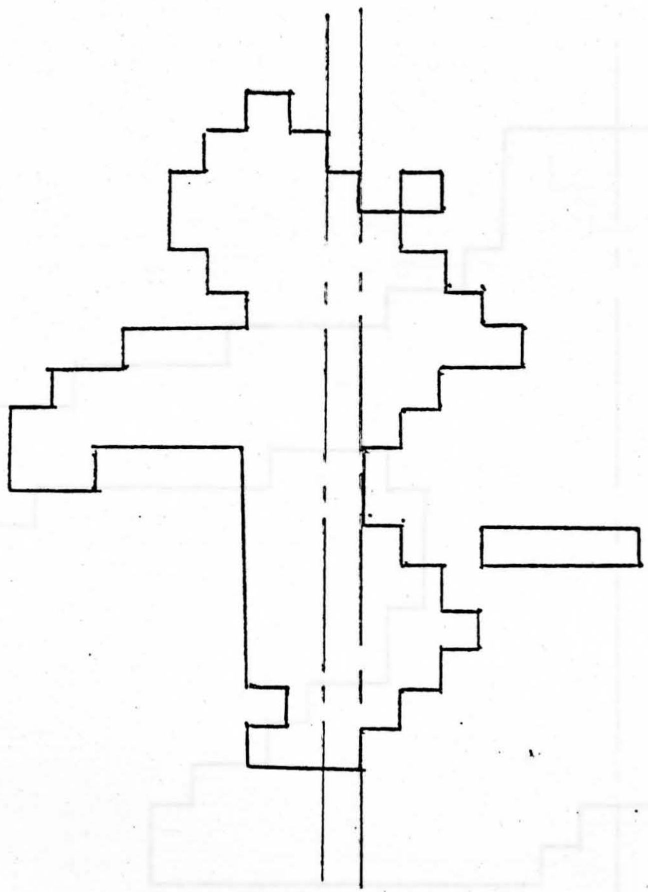


Image 15

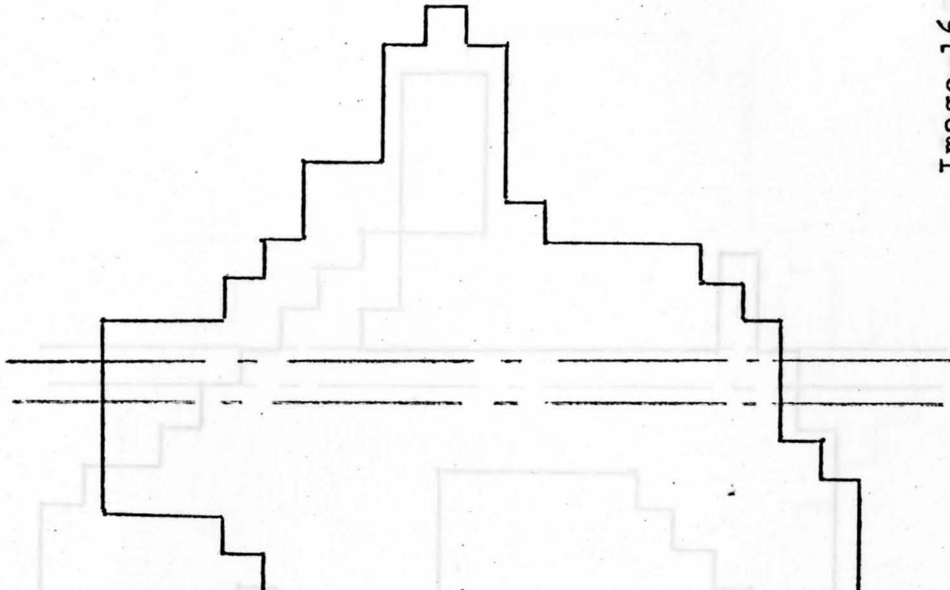


Image 16

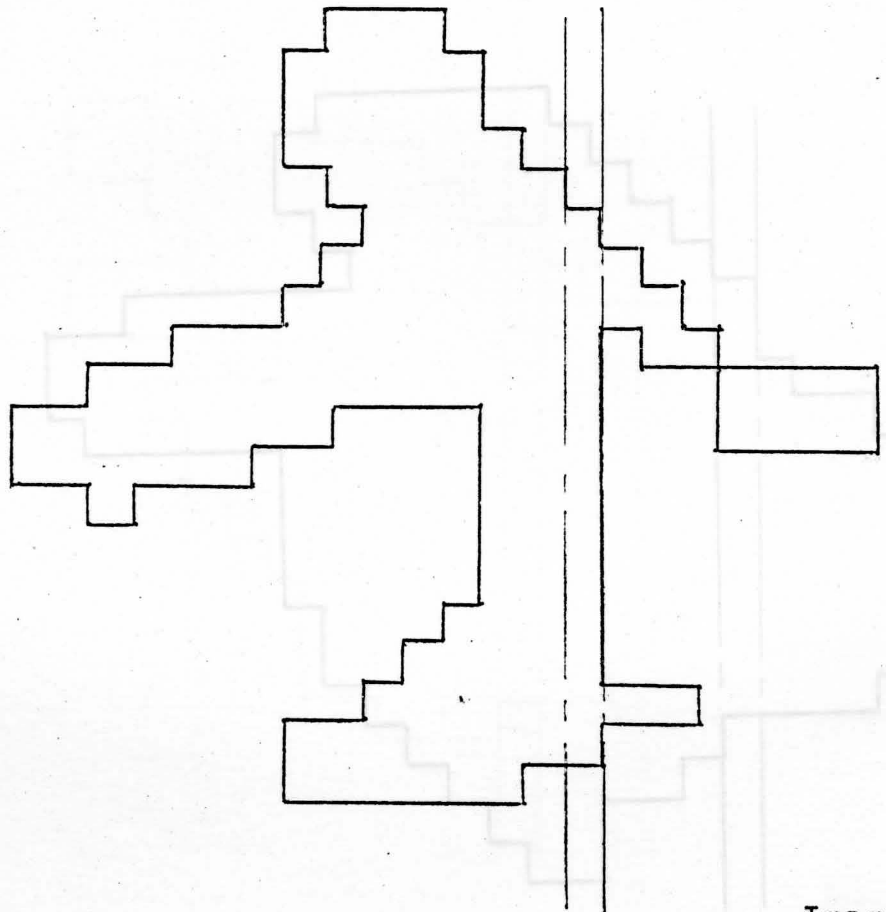


Image 17

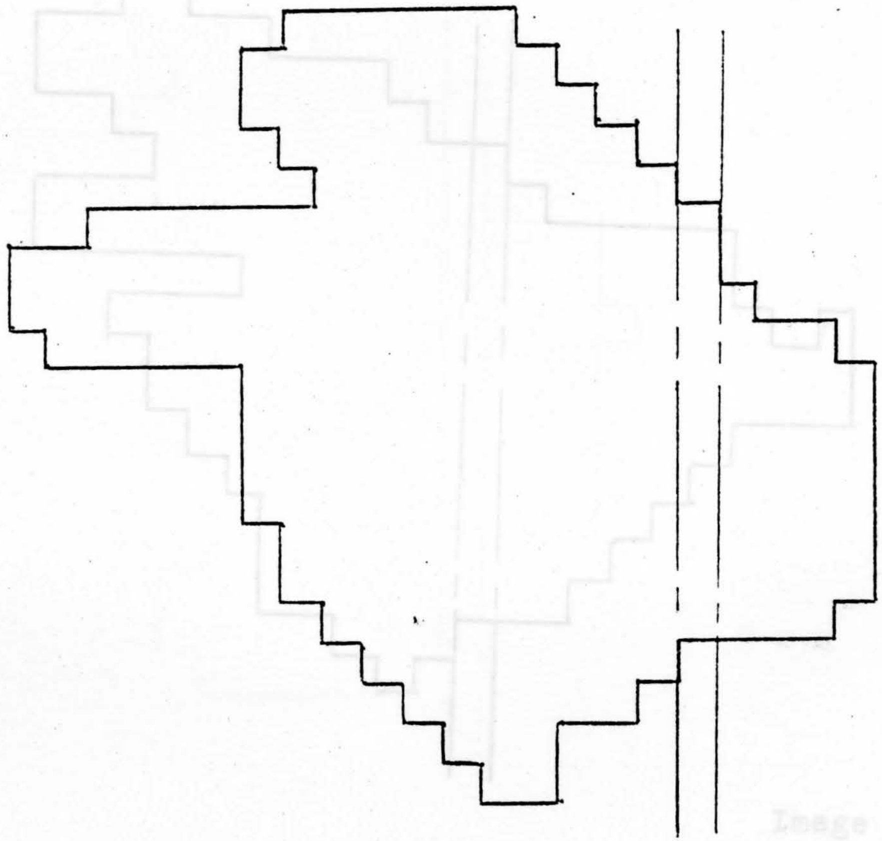


Image 18

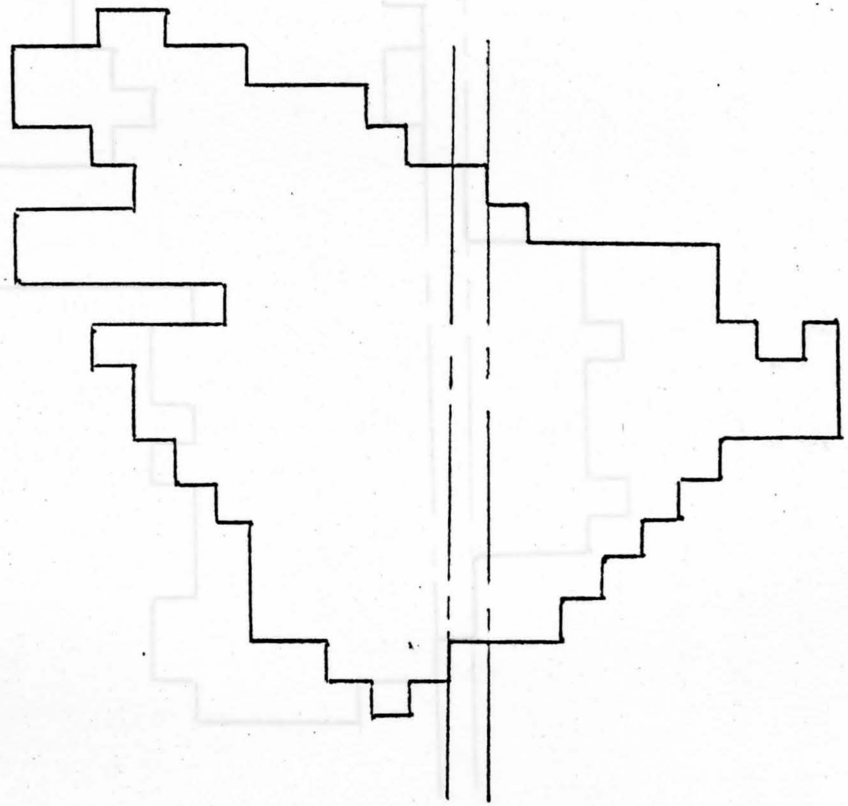


Image 19

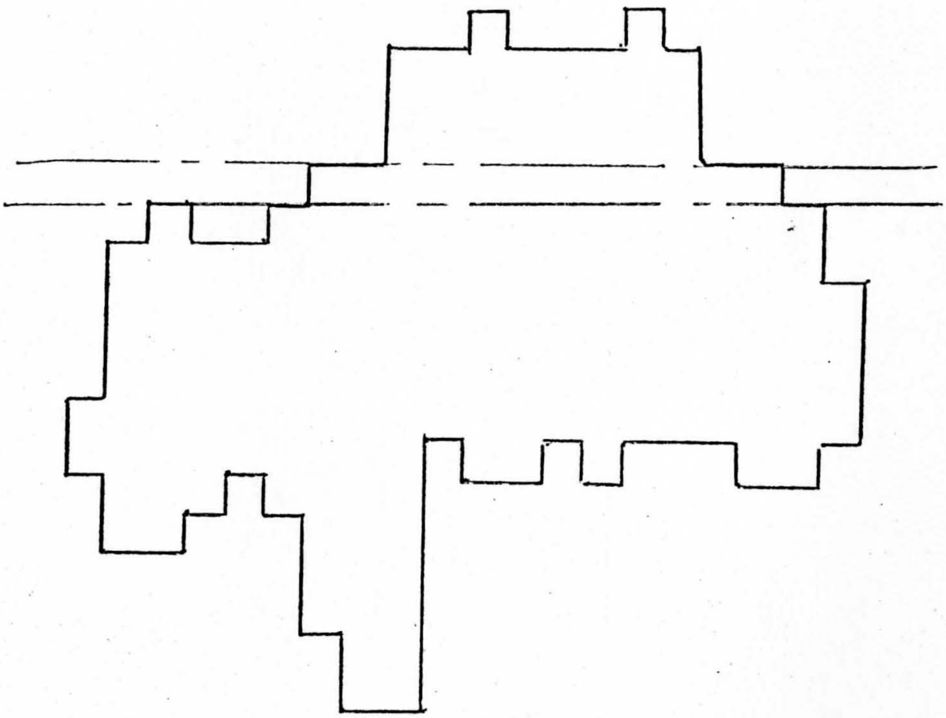


Image 20

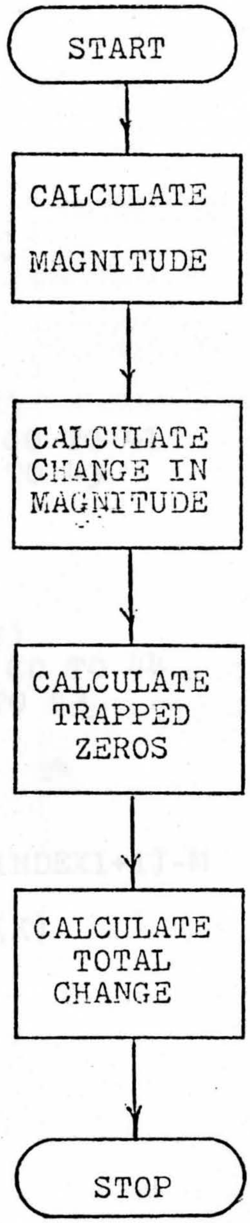
APPENDIX D

Feature Extraction Computer Simulation

The calculation of the PDFs and the training of the weights were based on feature measurements performed on the 20 images of appendix C. To obtain the feature measurements two computer programs were written. Flow diagrams and possible fortran implementations are presented.

Subroutine STAT1

Flow Diagram



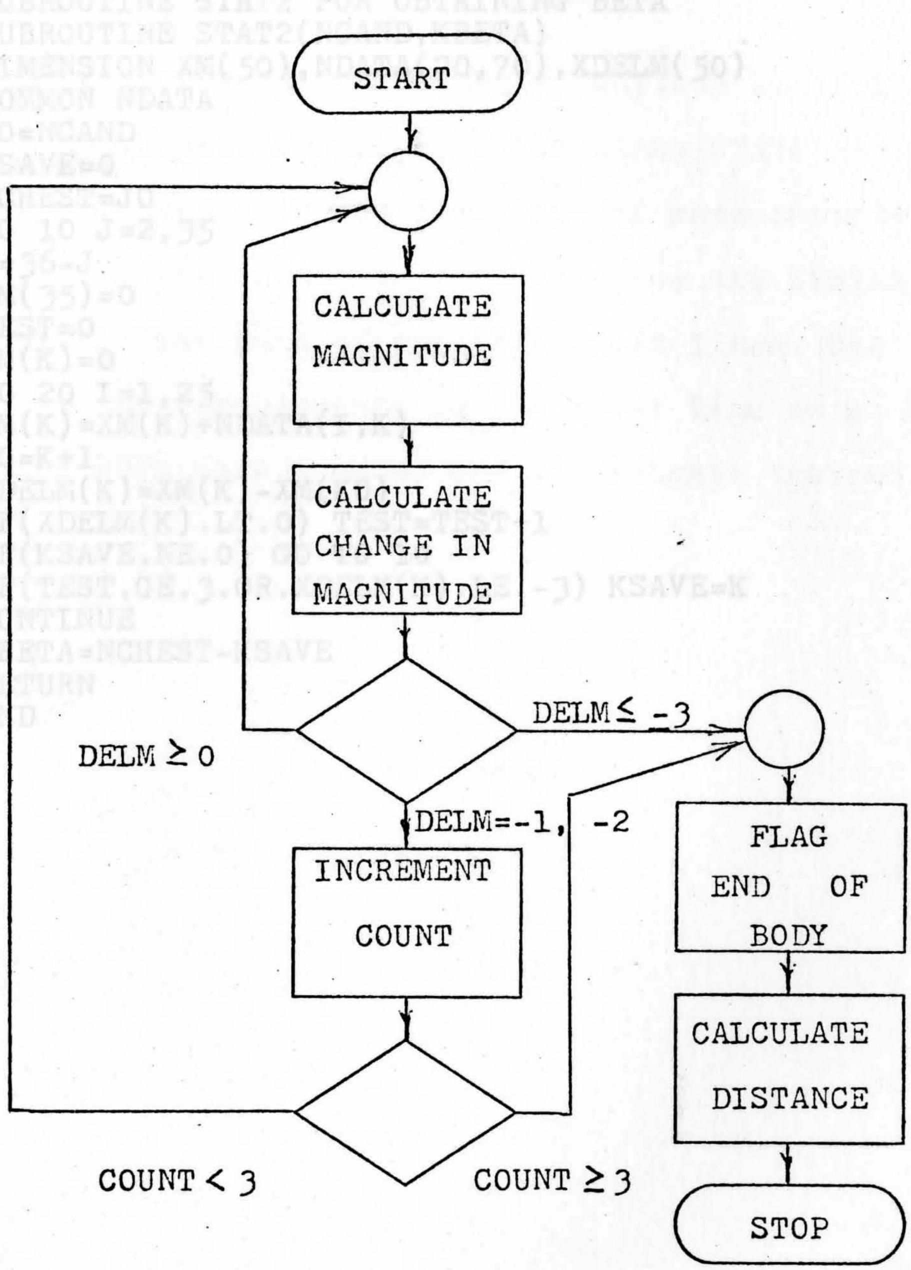
Subroutine STAT1

Computer Program

```
C       SUBROUTINE STAT1 FOR OBTAINING M,DELM,XNZ AND DELZ
        SUBROUTINE STAT1(NCAND,M,DELM,XNZ,DELZ)
        DIMENSION NDATA(70,70),NZ(50)
        COMMON NDATA
        JO=NCAND
        M=0
        DO 10 K=1,25
 10     M=M+NDATA(K,JO)
        M1=0
        K=JO+1
        DO 20 L=1,25
 20     M1=M1+NDATA(L,K)
        DELM=M-M1
        DO 30 J=JO,K
        L=0
 40     L=L+1
        IDATA=NDATA(L,J)
        IF(IDATA.EQ.1) GO TO 41
        IF(L.EQ.25) GO TO 43
        GO TO 40
 41     INDEX1=L
        L=26
 42     L=L-1
        IDATA=NDATA(L,J)
        IF(IDATA.EQ.1) GO TO 44
        IF(L.EQ.1) GO TO 43
        GO TO 42
 43     NZ(J)=0
        GO TO 30
 44     INDEX2=L
        NZ(J)=(INDEX2-INDEX1+1)-M
 30     CONTINUE
        DELZ=NZ(JO)-NZ(K)
        XNZ=NZ(JO)
        RETURN
        END
```

Subroutine STAT2

Flow Diagram



Subroutine STAT2

Computer Program

```
C       SUBROUTINE STAT2 FOR OBTAINING BETA
        SUBROUTINE STAT2(NCAND,KBETA)
        DIMENSION XM(50),NDATA(70,70),XDELM(50)
        COMMON NDATA
        JO=NCAND
        KSAVE=0
        NCHEST=JO
        DO 10 J=2,35
        K=36-J
        XM(35)=0
        TEST=0
        XM(K)=0
        DO 20 I=1,25
 20     XM(K)=XM(K)+NDATA(I,K)
        KO=K+1
        XDELM(K)=XM(K)-XM(KO)
        IF(XDELM(K).LT.0) TEST=TEST+1
        IF(KSAVE.NE.0) GO TO 10
        IF(TEST.GE.3.OR.XDELM(K).LE.-3) KSAVE=K
 10     CONTINUE
        KBETA=NCHEST-KSAVE
        RETURN
        END
```

APPENDIX E

Feature Measurements

The following data provides a complete listing of the feature measurements made by subroutines STAT1 and STAT2. The data is oriented with the first set of measurements corresponding to the candidate farthest from the finish line but still within the band of possible chest lines. The underlined set of measurements is the chest line data. From this data the PDFs were created and the weights trained.

MAGNITUDE	CHANGE IN MAGNITUDE	TRAPPED ZEROS	TOT	BETA
FIGURE 1				
16	0	0	0	3
16	0	0	0	4
16	2	0	4	5
14	2	0	4	6
12	2	0	4	7
10	2	0	4	8
8	1	0	2	9
7	1	0	2	10
6	0	0	0	11
6	3	0	6	12
3	3	0	3	13
FIGURE 2				
14	0	0	0	3
14	-1	0	2	4
15	-1	0	2	5
16	0	0	0	6
16	0	0	0	7
16	1	0	2	8
15	0	0	0	9
15	1	0	2	10
14	3	0	6	11
11	3	0	4	12
8	3	2	7	13
FIGURE 3				
18	0	0	0	3
18	-1	0	2	4
19	-1	0	2	5
20	0	0	0	6
20	2	0	4	7
18	11	0	14	8
7	1	8	2	9
6	2	8	3	10
4	1	9	11	11
3	2	0	4	12
1	-1	0	2	13

FIGURE 4

M	dM	Z	TOT	BETA
15	-1	0	2	3
16	-1	0	2	4
17	8	0	9	5
9	1	7	1	6
8	3	8	4	7
5	4	10	18	8
1	-1	0	2	9
2	1	0	2	10
1	1	0	1	11
0	0	0	0	12
0	0	0	0	13

FIGURE 5

M	dM	Z	TOT	BETA
22	1	0	2	3
21	2	0	4	4
19	2	0	4	5
17	3	0	6	6
14	2	0	4	7
12	1	0	2	8
11	4	0	8	9
7	2	0	4	10
5	2	0	4	11
3	1	0	1	12
2	2	1	3	13

FIGURE 6

M	dM	Z	TOT	BETA
21	1	0	2	3
20	1	0	2	4
19	0	0	0	5
19	3	0	4	6
16	3	2	5	7
13	1	3	2	8
12	1	3	1	9
11	2	4	3	10
9	6	5	17	11
3	3	0	3	12
0	0	0	0	13

M	dM	Z	TOT	BETA
FIGURE 7				
16	0	0	0	3
16	2	0	4	4
14	2	0	4	5
12	0	0	0	6
12	2	0	4	7
10	0	0	0	8
10	1	0	2	9
9	1	0	2	10
8	3	0	6	11
5	1	0	2	12
4	2	0	4	13
FIGURE 8				
21	0	0	0	3
21	0	0	0	4
21	5	0	10	5
16	3	0	6	6
13	1	0	2	7
12	3	0	5	8
9	1	1	3	9
8	2	0	4	10
6	3	0	6	11
3	1	0	2	12
2	2	0	2	13
FIGURE 9				
17	4	0	7	3
13	2	1	3	4
11	1	2	2	5
10	1	2	2	6
9	2	2	3	7
7	1	3	2	8
6	5	3	13	9
1	-1	0	2	10
2	0	0	0	11
2	0	0	0	12
2	0	0	0	13

M	dM	Z	TOT	BETA
FIGURE 10				
16	-1	1	1	3
17	0	0	0	4
17	0	0	0	5
17	0	0	0	6
17	2	0	4	7
15	2	0	4	8
13	7	0	9	9
6	2	5	3	10
4	-1	6	1	11
5	-1	5	1	12
6	-1	4	1	13
FIGURE 11				
11	-2	0	4	3
13	0	0	0	4
13	1	0	2	5
12	2	0	3	6
10	1	1	2	7
9	2	1	5	8
7	1	0	2	9
6	1	0	2	10
5	2	0	4	11
3	3	0	3	12
0	0	0	0	13
FIGURE 12				
12	-2	1	3	3
14	0	0	0	4
14	0	0	0	5
14	-1	0	2	6
15	0	0	0	7
15	-2	0	4	8
17	1	0	2	9
16	0	0	0	10
16	0	0	0	11
16	4	0	8	12
12	2	0	4	13

M	dM	Z	TOT	BETA
FIGURE 13				
17	0	0	0	3
17	1	0	2	4
16	0	0	0	5
16	1	0	2	6
15	1	0	2	7
14	0	0	0	8
14	4	0	8	9
10	1	0	2	10
9	1	0	1	11
8	0	1	0	12
8	3	1	7	13
FIGURE 14				
19	0	0	0	3
19	1	0	2	4
18	1	0	2	5
17	2	0	2	6
15	6	1	11	7
9	0	2	0	8
9	0	2	1	9
9	0	1	0	10
9	1	1	2	11
8	0	1	0	12
8	0	1	0	13
FIGURE 15				
15	4	0	6	3
11	3	2	3	4
8	5	5	9	5
3	1	6	4	6
2	1	4	6	7
1	0	0	0	8
1	0	0	0	9
1	1	0	1	10
0	0	0	0	11
0	0	0	0	12
0	0	0	0	13

M	dM	Z	TOT	BETA
FIGURE 16				
17	0	0	0	3
17	0	0	0	4
17	4	0	8	5
13	2	0	4	6
11	5	0	10	7
6	1	0	2	8
5	2	0	4	9
3	0	0	0	10
3	0	0	0	11
3	2	0	4	12
1	1	0	1	13
FIGURE 17				
14	11	0	13	3
3	0	9	1	4
3	2	8	12	5
1	-1	0	2	6
2	0	0	0	7
2	0	0	0	8
2	0	0	0	9
2	2	0	2	10
0	0	0	0	11
0	0	0	0	12
0	0	0	0	13
FIGURE 18				
17	-1	0	2	3
18	-1	0	2	4
19	-1	0	2	5
20	1	0	2	6
19	3	0	6	7
16	1	0	2	8
15	3	0	6	9
12	1	0	2	10
11	2	0	4	11
9	1	0	2	12
8	0	0	0	13

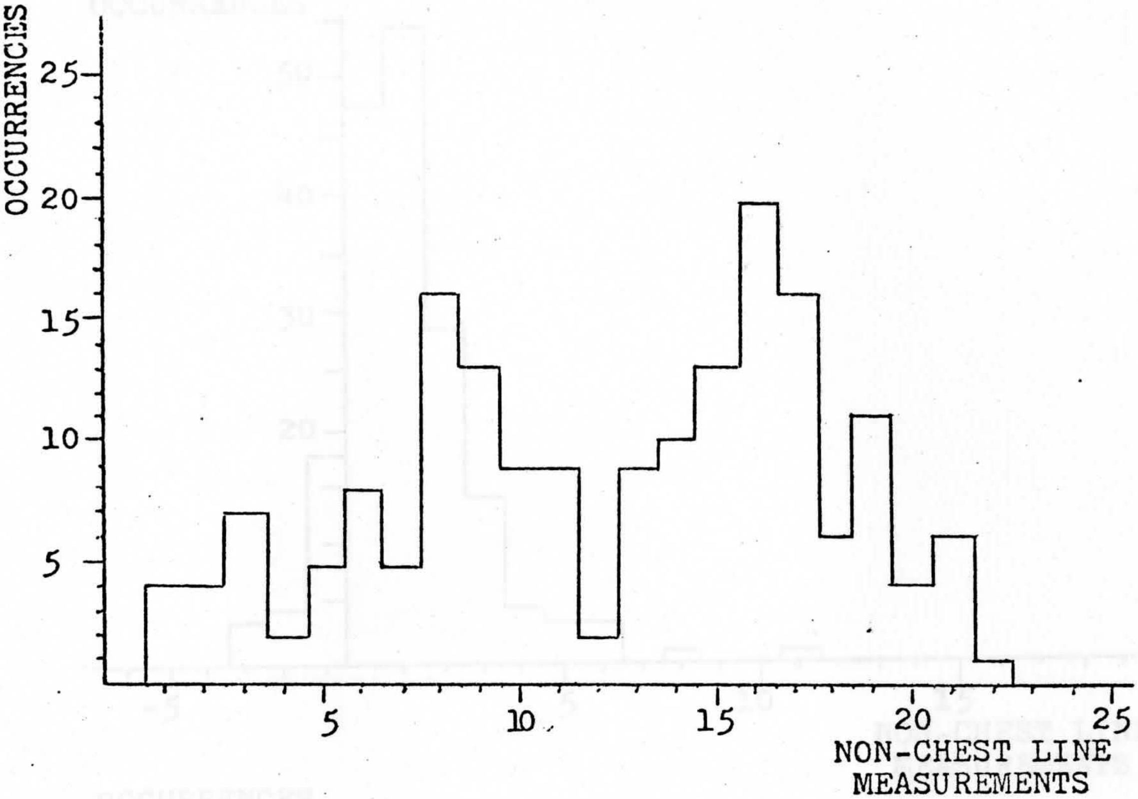
M	dM	Z	TOT	BETA
FIGURE 19				
12	-2	0	4	3
14	0	0	0	4
14	-1	0	2	5
15	0	0	0	6
15	2	0	4	7
13	1	0	2	8
12	1	0	2	9
11	1	0	2	10
10	1	0	2	11
9	1	0	2	12
8	1	0	2	13
FIGURE 20				
19	0	0	0	3
19	1	0	2	4
18	3	0	4	5
15	3	2	8	6
12	4	0	8	7
8	0	0	0	8
8	0	0	0	9
8	6	0	9	10
2	2	3	5	11
0	0	0	0	12
0	0	0	0	13

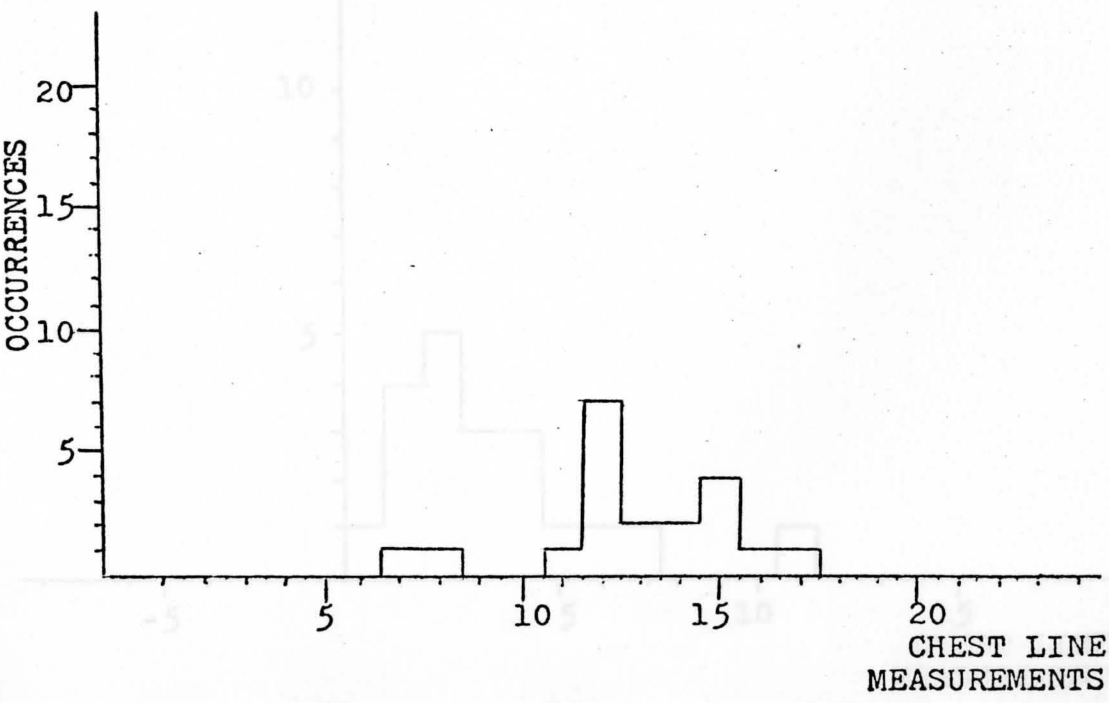
APPENDIX F

Formulation of the PDFs

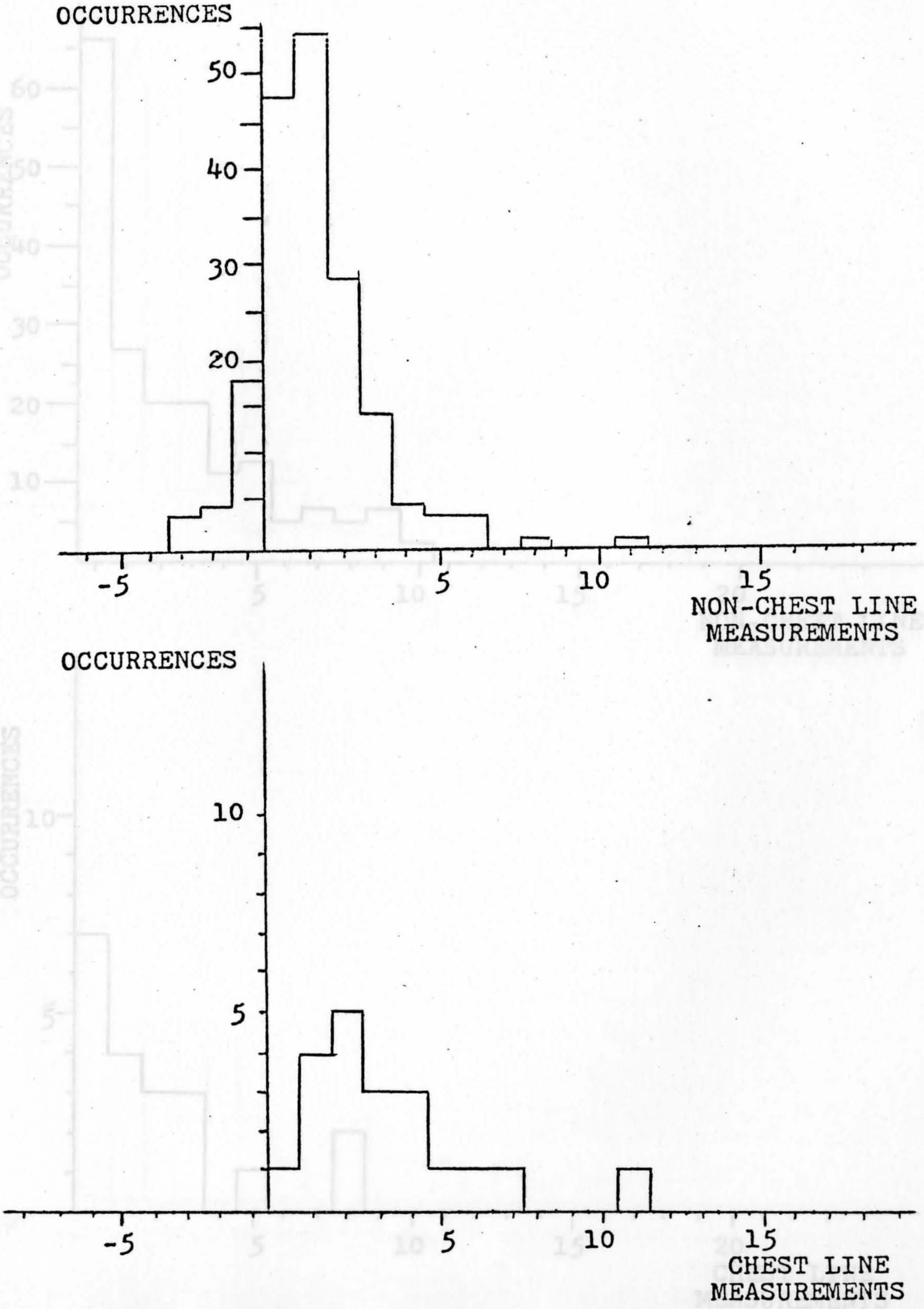
Using the Bayes Expansion, two probability functions are generated for each PDF to be used in the algorithm. Each of these can be easily realized once the feature measurements have been made on the training set. The following graphs present the information needed in generating the PDFs. The top graph in each case gives the number of times a particular measurement occurs if the measurement is made on a non-chest line candidate. The bottom graph shows occurrences for measurements made on chest line candidates. The probability of getting a particular measurement given the candidate is a chest line or non-chest line can be calculated by dividing the graphs by the total possible occurrences of any measurement. However, this term drops out in the calculation of the final PDF thus allowing the use of these curves in evaluating the PDF performances.

MAGNITUDE

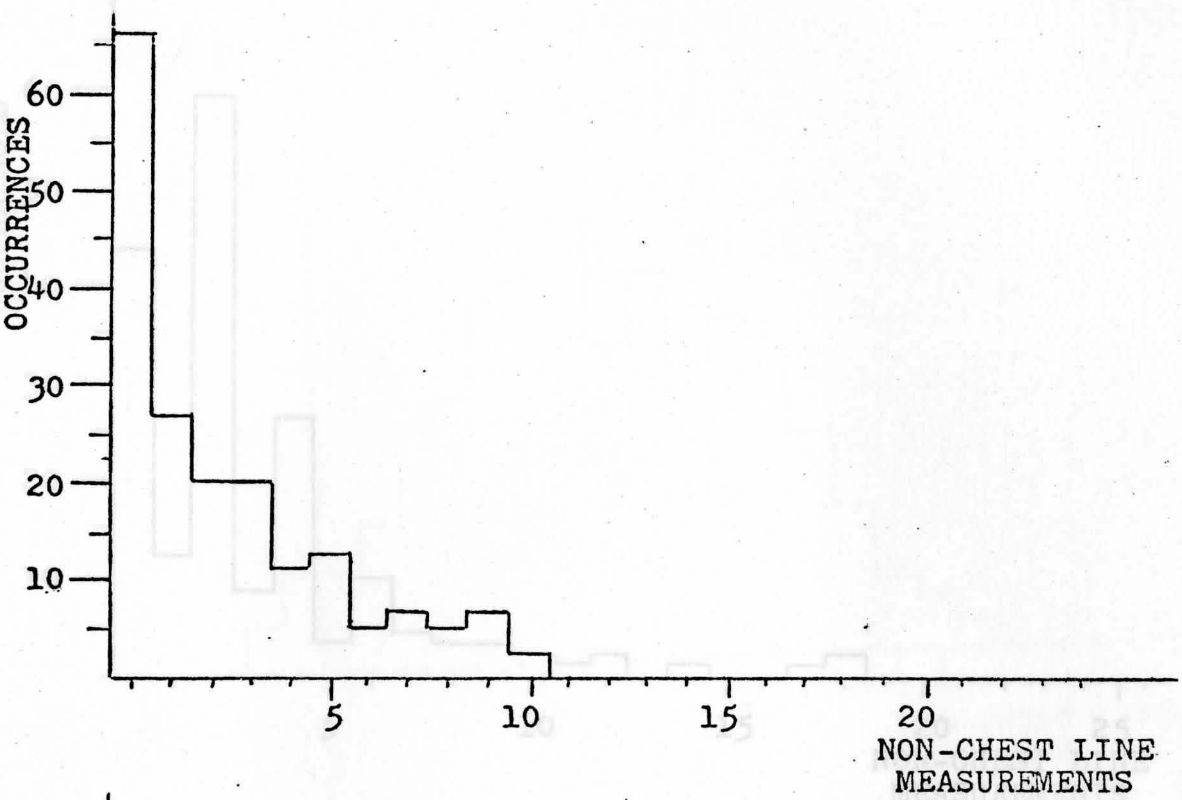

OCCURRENCES
25
20
15
10
5
5
10
15
20
25
NON-CHEST LINE
MEASUREMENTS

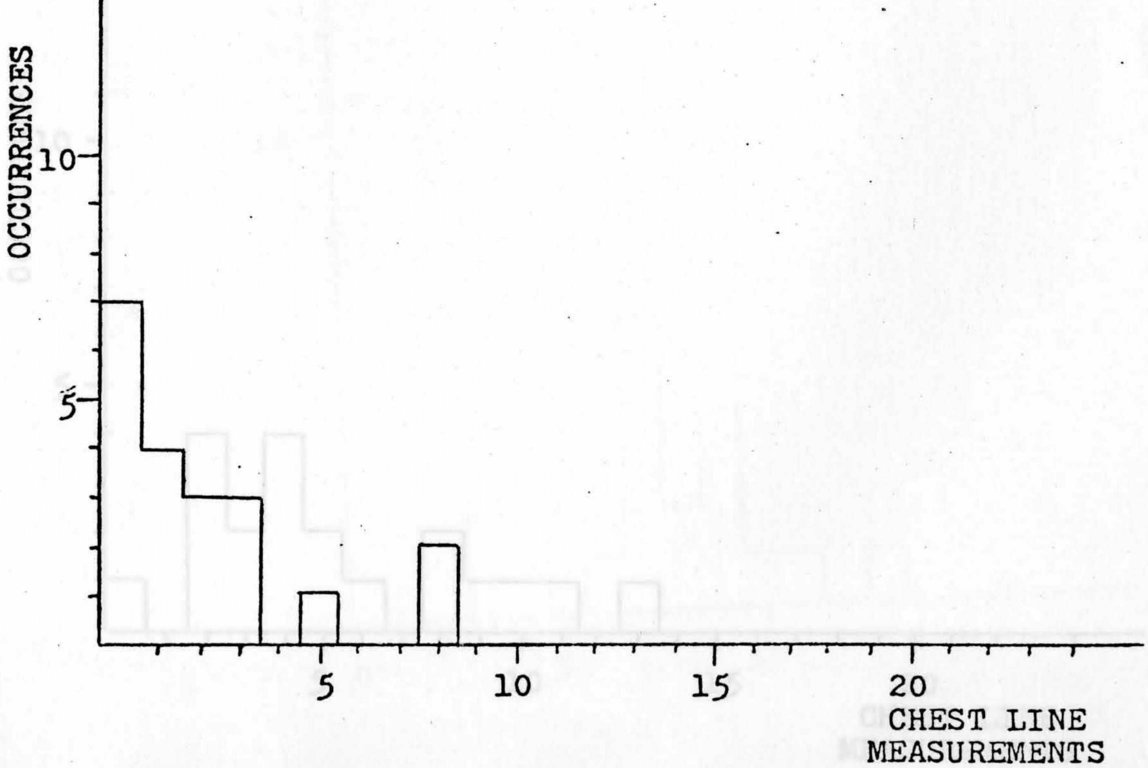

OCCURRENCES
20
15
10
5
5
10
15
20
CHEST LINE
MEASUREMENTS

CHANGE IN MAGNITUDE

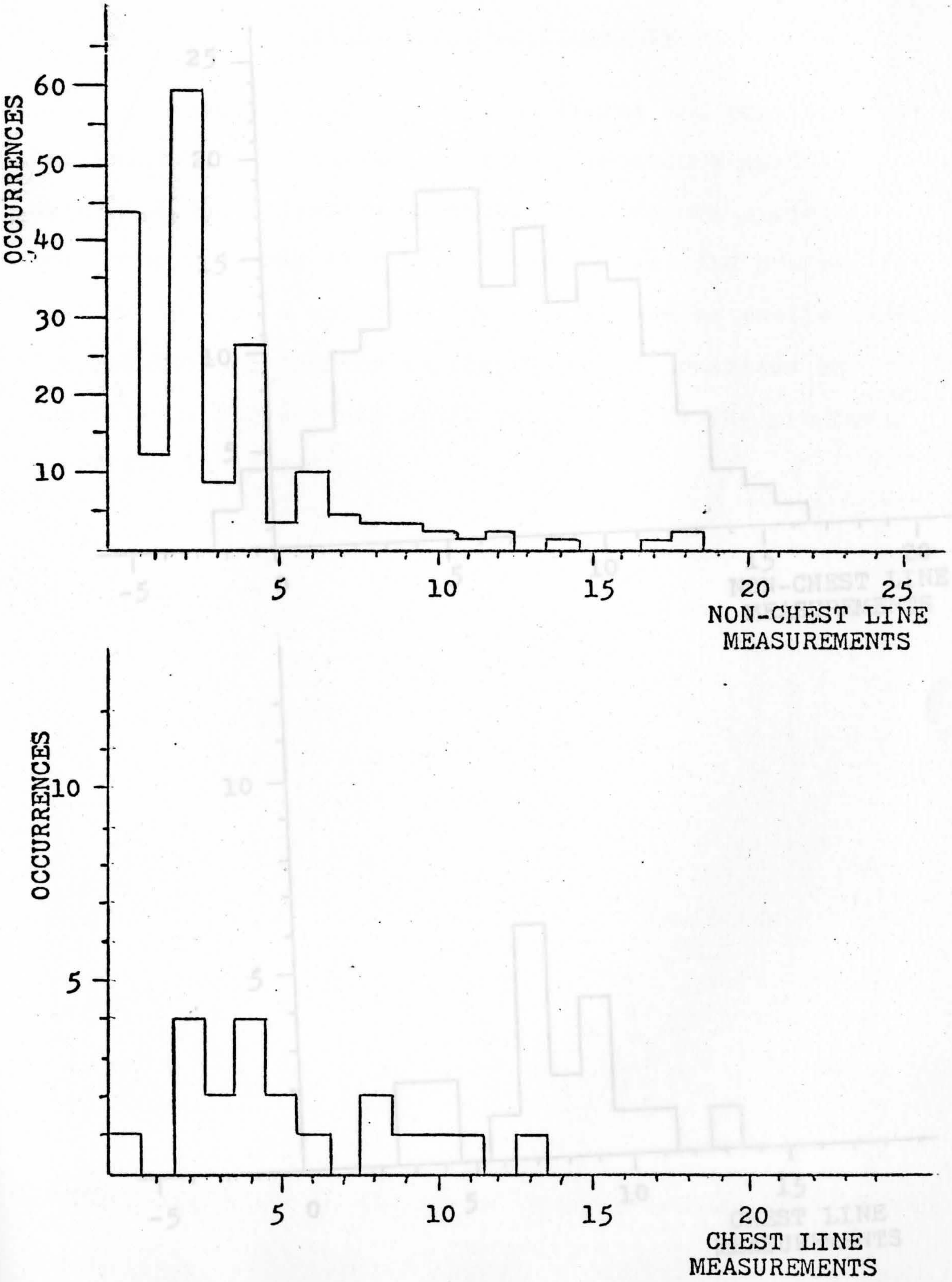


TRAPPED ZEROS

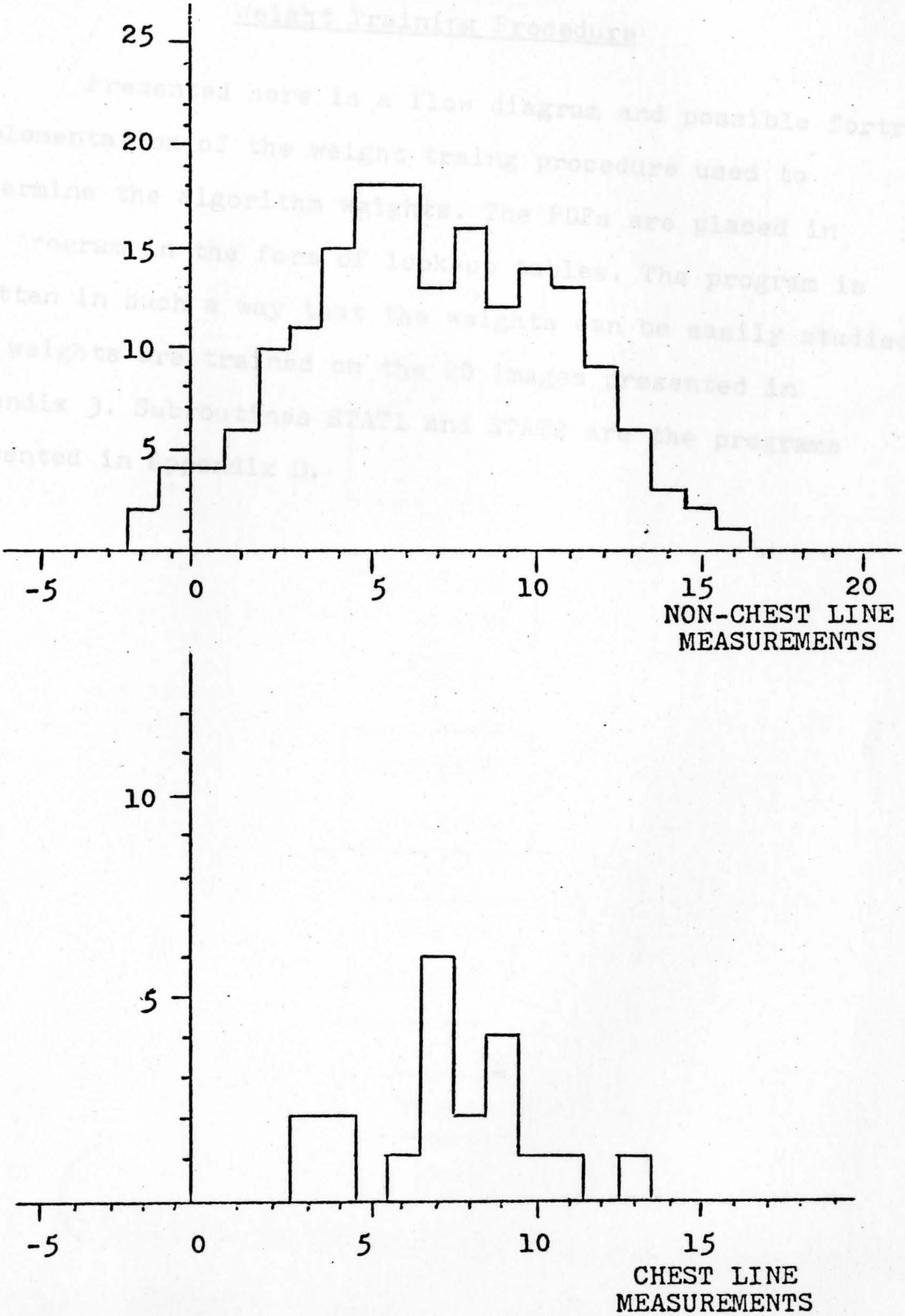




TOTAL CHANGE



END OF BODY (BETA)



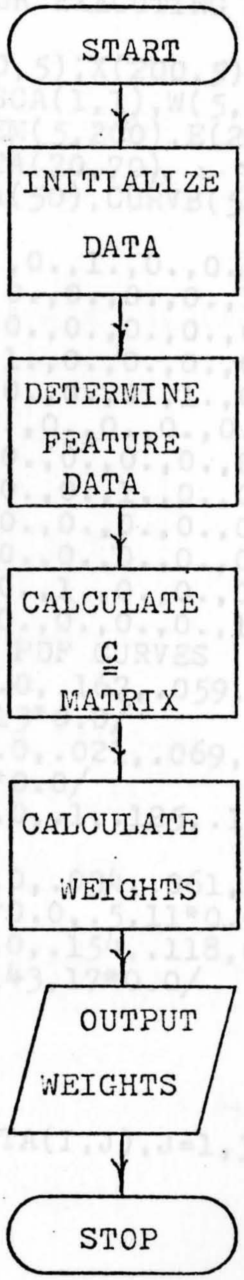
APPENDIX G

Weight Training Procedure

Presented here is a flow diagram and possible fortran implementation of the weight traing procedure used to determine the algorithm weights. The PDFs are placed in the program in the form of look-up tables. The program is written in such a way that the weights can be easily studied. The weights are trained on the 20 images presented in Appendix 3. Subroutines STAT1 and STAT2 are the programs presented in Appendix D.

Weight Training Procedure

Flow Diagram



Weight Training Procedure

Computer Program

```
C      MAIN PROGRAM FOR EXECUTING TRAINING PROCEDURE OF
C      WEIGHTS
       DIMENSION C(200,5),X(200,5),XVEC(5,5),XTRA(5,1),
     1 P(5,5),R(1,5),SCA(1,1),W(5,1),G(1,1),H(5,1),
     2 CTRAN(5,200),TEM(5,200),E(200,1),PCPY(5,5),DENT(5,5),
     3 L(5),MO(5),NDATA(70,70)
       DIMENSION CURVA(50),CURVB(50),CURVC(50),CURVE(50)
       COMMON NDATA
       DATA E/0.,0.,0.,0.,1.,0.,0.,0.,0.,0.,0.,0.,0.,0.,1.,
     1 0.,0.,0.,0.,0.,0.,0.,0.,0.,1.,0.,0.,0.,0.,0.,0.,
     2 0.,0.,1.,0.,0.,0.,0.,0.,0.,0.,0.,1.,0.,0.,0.,0.,
     3 0.,0:,0.,0.,0.,1.,0.,0.,0.,0.,0.,0.,0.,0.,1.,0.,
     4 0.,0.,0.,0.,0.,0.,0.,0.,1.,0.,0.,0.,0.,0.,0.,0.,
     50.,1.,0.,0.,0.,0.,0.,0.,0.,0.,1.,0.,0.,0.,0.,0.,
     6 0.,0.,0.,0.,1.,0.,0.,0.,0.,0.,0.,0.,0.,1.,0.,0.,
     7 0.,0.,0.,0.,0.,0.,0.,1.,0.,0.,0.,0.,0.,0.,0.,0.,
     8 1.,0.,0.,0.,0.,0.,0.,0.,0.,0.,1.,0.,0.,0.,0.,0.,
     9 0.,0.,0.,1.,0.,0.,0.,0.,0.,0.,0.,0.,1.,0.,0.,0.,
     a 0.,0.,0.,0.,0.,0.,1.,0.,0.,0.,0.,0.,0.,0.,0.,1.,
     b 0.,0.,0.,0.,0.,0.,0.,0.,0.,1.,0.,0.,0.,0.,0./
C      CREATE DISCRETE PDF CURVES
       DATA CURVA/26*0.0,.167,.059,2*0.0,.1,.778,.182,.167,
     1 .235,.048,.059,13*0.0/
       DATA CURVB/19*0.0,.021,.069,.151,.176,.429,.25,.25,
     1 1.0,2*0.0,.5,20*0.0/
       DATA CURVC/19*0.0,.1,.125,.13,.13,0.0,.077,2*0.0,
     1 .125,22*0.0/
       DATA CURVD/19*0.0,.034,.061,.079,.133,.2,.182,.111,
     1 .25,4*0.0,1.0,6*0.0,.5,11*0.0/
       DATA CURVE/22*0.0,.154,.118,0.0,.05,.273,.111,.25,
     1 .067,.071,0.0,.143,17*0.0/
C
       DO 1 N=1,20
       READ(5,101) JO
 101   FORMAT(1I2)
       READ(5,100)((NDATA(I,J),J=1,35),I=1,25)
 100   FORMAT(35I1)
```

```
      DO 2 I=1,10
            NCAND=J0+5-I
            K=((N-1)*10)+I
            CALL STAT1(NCAND,M,DELM,XNZ,DELZ)
            CALL STAT2(NCAND,KBETA)
            TOT=ABS(DELM)+ABS(DELZ)
            X(K,1)=M
            X(K,2)=DELM
            X(K,3)=XNZ
            X(K,4)=TOT
            X(K,5)=KBETA
            IF(E(K,1).EQ.0.) E(K,1)=-1.0
 2          CONTINUE
 1    CONTINUE
C
C     CALCULATION OF MATRIX P=(C(TRANS)C)**-1
      DO 501 I=1,200
            IA=X(I,1)+20
            IB=X(I,2)+20
            IC=X(I,3)+20
            ID=X(I,4)+20
            IE=X(I,5)+20
            C(I,1)=CURVA(IA)
            C(I,2)=CURVB(IB)
            C(I,3)=CURVC(IC)
            C(I,4)=CURVD(ID)
            C(I,5)=CURVE(IE)
 501        CONTINUE
      CALL MTRA(C,CTRAN,200,5,0)
      CALL MPRD(CTRAN,C,P,5,200,0,0,5)
      CALL MINV(P,5,DET,L,MO)
      PRINT 622,DET
      IF(DET.EQ.0) GO TO 502
C
C     CALCULATION OF VECTOR W=P*CTRAN*E
      CALL MPRD(P,CTRAN,TEM,5,5,0,0,200)
      CALL MPRD(TEM,E,W,5,200,0,0,1)
      PRINT 601
 601  FORMAT(10X,'WEIGHTS')
      PRINT 611,W(1,1),W(2,1),W(3,1),W(4,1),W(5,1)
 611  FORMAT(5G10.3)
      PRINT 603
 603  FORMAT(10X,'P MATRIX')
```

```
      PRINT 622,((P(I,J),J=1,5),I=1,5)
622   FORMAT(5G10.3)
C
      GO TO 503
502   PRINT 605
605   FORMAT(10X,'DETERMINANT EQUAL ZERO')
503   STOP
      END
```

APPENDIX H

Weight Investigation

Before the simulation of the algorithm could begin it was necessary to determine the significance of the weighting coefficiants and their effects on the accuracy and reliability of the algorithm. A series of 3 tests were run on the weights using the training program in which the results of the PDFs were controlled. These tests showed the relationship between the weights and the PDFs. The tests are explained below and the resulting conclusions given.

Test 1 One of the PDFs was modified to give a result of 1 every time a chest line measurement was taken and a -1 for every non-chest line measurement. This indicated that the feature for the modified PDF ideally selected the chest line in every case. The resulting set of weights for this test went to 0 for every feature except that which corresponded to the modified PDF. In the case of the modified PDF, the weight went to 1. The coefficient of 1 for the weight was a result of the definition of the ideal E vector. Since, for a chest line the resulting E vector valve must be 1 and the PDF had been modified as given the weight valve was forced to equal 1. If the PDF had been modified to give a result of .5 in the case of a chest line and -1 otherwise the corresponding weight value would have been a 2. The significance of this test was in the driving to '0' of all weights for features which did not ideally select the chest line. This result held true for all features. If two features were modified to ideally select the chest line the corresponding weight values were 0.5 with all other weights driven to 0.

<u>Test 2</u> In this case the PDF was modified so the feature ideally selected non-chest lines. The PDF gave a result of 1 for every non-chest line measurement and a -1 for all chest line measurements. The resulting set of weights for this test also went to 0 for every case except that which corresponded to the modified PDF. However, the weight value instead of being 1 had a value of -1.

<u>Test 3</u> The PDF for a particular feature was replaced with a random number generator. In this case the weights were determined to have finite values with the exception of the modified PDF which tended toward 0.

From these tests, it can be seen that the weights are well behaved with respect to the PDFs. It is also possible to use the weight training procedure as a form of feature selection algorithm. If the weight for a particular feature tends toward 1, the feature is a good chest line indicator. If the weights tend toward -1, the feature is a good non-chest line indicator. A weight which tends toward zero indicates a feature which provides little or no useful information in the classification of the chest line. If the weight is significantly close to zero, the feature may be removed from the algorithm with no effect on the results.

The weight training procedure is not a convergence method. However, as the number of images is increased the weights continue to react to the new images. At some point in time no new data can be presented to the weight trainer. Only images already seen by the trainer will be possible. At this time the weights will be stable at their values. this is true the weights should be well behaved, that is the there should be little or no oscillation as the weights approach their final value.

To test this a random sample of 10 images was selected and the weights determined. It can be seen that the weights have already approached the values obtained from 20 images. As more samples are added to the training set, the weights will appear to converge on a set of values which will contain all possible data relating the PDFs to one another. More work in this area is needed to understand the effect of the training set size on the weights.

This is a table listing the weight values for the two training sets.(20 and 10 images)

	WEIGHTS	
FEATURE	10 IMAGES	20 IMAGES
MAGNITUDE	1.57	1.51
CHANGE IN MAGNITUDE	.621	1.07
TRAPPED ZEROS	-9.92	-9.77
TOTAL CHANGE	-.498	-.735
END OF BODY(Beta)	.760	.517

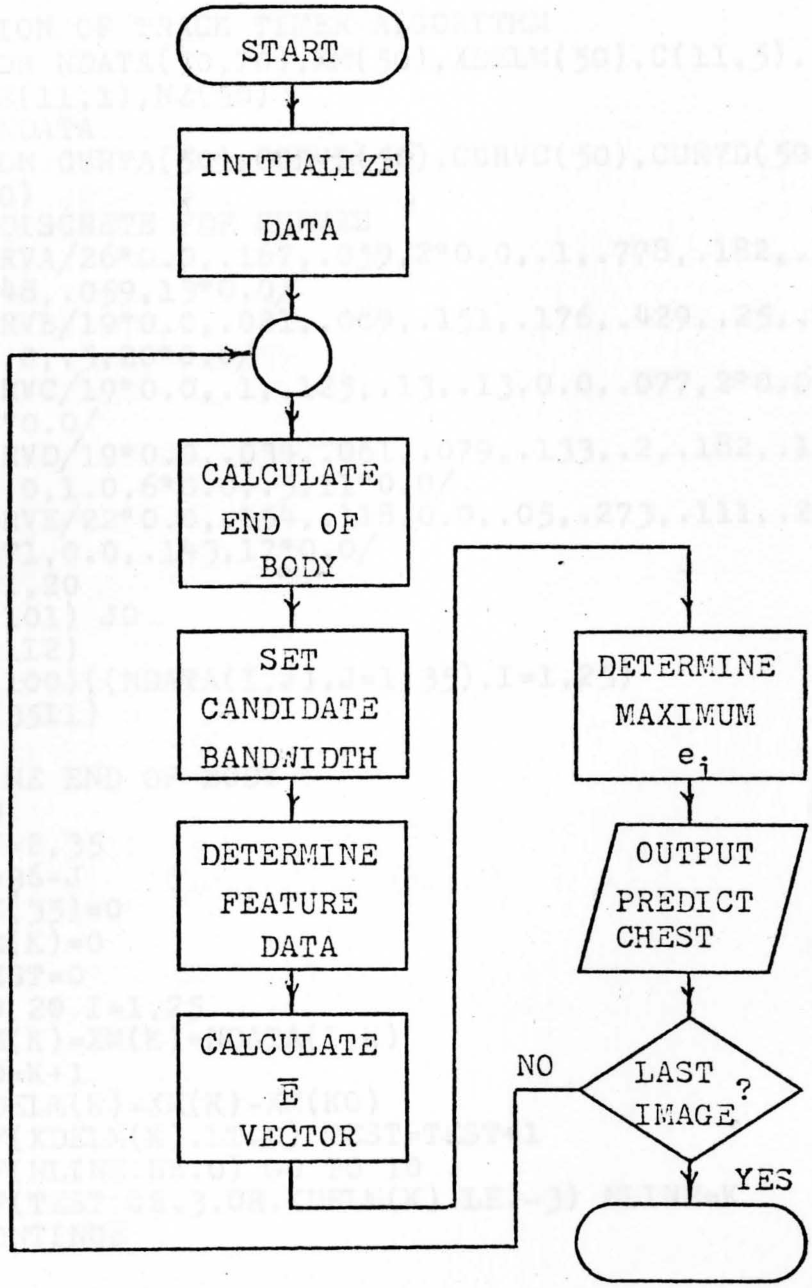
APPENDIX I

Algorithm Simulation

Presented here is the flow diagram and a possible fortran program implementing the decision algorithm simulation. The simulation was developed to test the algorithm only and in no way attempts to simulate the microprocessor software or the feature extraction circuitry.

Algorithm Simulation

Flow Diagram



Algorithm Simulation

Computer Program

```
C         SIMULATION OF TRACK TIMER ALGORITHM
          DIMENSION NDATA(70,70),XM(50),XDELM(50),C(11,5),
     1    W(5,1),E(11,1),NZ(50)
          COMMON NDATA
          DIMENSION CURVA(50),CURVB(50),CURVC(50),CURVD(50)
     1    CURVE(50)
C         CREATE DISCRETE PDF CURVES
          DATA CURVA/26*0.0,.167,.059,2*0.0,.1,.778,.182,.167,
     1    .235,.048,.059,13*0.0/
          DATA CURVB/19*0.0,.021,.069,.151,.176,.429,.25,.25,
     1    1.0,2*0.0,.5,20*0.0/
          DATA CURVC/19*0.0,.1,.125,.13,.13,0.0,.077,2*0.0,
     1    .125,22*0.0/
          DATA CURVD/19*0.0,.034,.061,.079,.133,.2,.182,.111,
     1    .25,4*0.0,1.0,6*0.0,.5,11*0.0/
          DATA CURVE/22*0.0,.154,.118,0.0,.05,.273,.111,.25,
     1    .067,.071,0.0,.143,17*0.0/
          DO 1 N=1,20
          READ(5,101) JO
 101      FORMAT(1I2)
          READ(5,100)((NDATA(I,J),J=1,35),I=1,25)
 100      FORMAT(35I1)
C
C         DETERMINE END OF BODY
          NLINE=0
          DO 10 J=2,35
                K=36-J
                XM(35)=0
                XM(K)=0
                TEST=0
                DO 20 I=1,25
 20             XM(K)=XM(K)+NDATA(I,K)
                KO=K+1
                XDELM(K)=XM(K)-XM(KO)
                IF(XDELM(K).LT.0) TEST=TEST+1
                IF(NLINE.NE.0) GO TO 10
                IF(TEST.GE.3.OR.XDELM(K).LE.-3) NLINE=K
 10             CONTINUE
C
```

```
C         SETTING THE BAND OF CANDIDATES
          NCAND=NLINE+2
          DO 102 K=1,11
              NCAND=NCAND+1
C         FILLING THE C MATRIX
              CALL STAT1(NCAND,M,DELM,XNZ,DELZ)
              CALL STAT2(NCAND,KBETA)
              TOT=ABS(DELM)+ABS(DELZ)
              IA=M+20
              IB=DELM+20
              IC=XNZ+20
              ID=TOT+20
              IE=KBETA+20
              C(K,1)=CURVA(IA)
              C(K,2)=CURVB(IB)
              C(K,3)=CURVC(IC)
              C(K,4)=CURVD(ID)
              C(K,5)=CURVE(IE)
  102         CONTINUE
C
C         SPECIFICATION OF WEIGHTS
          W(1,1)=1.51
          W(2,1)=1.07
          W(3,1)=-9.77
          W(4,1)=-.735
          W(5,1)=0.517
C
C         CALCULATION OF E VECTOR
          CALL MPRD(C,W,E,11,5,0,0,1)
C         LOCATION OF CHEST LINE
          BIG=-99.0
          NCHEST=1
          DO 103 I=1,11
              IF(BIG-E(I,1)) 105,104,103
  103         CONTINUE
          GO TO 112
  104     IF(C(I,1)-C(NCHEST,1)) 103,106,105
  106     IF(C(I,2)-C(NCHEST,2)) 103,107,105
  107     IF(C(I,3)-C(NCHEST,3)) 103,108,105
  108     IF(C(I,4)-C(NCHEST,4)) 103,109,105
  109     IF(C(I,5)-C(NCHEST,5)) 103,111,105
  105     NCHEST=I
          BIG=E(I,1)
          GO TO 103
```

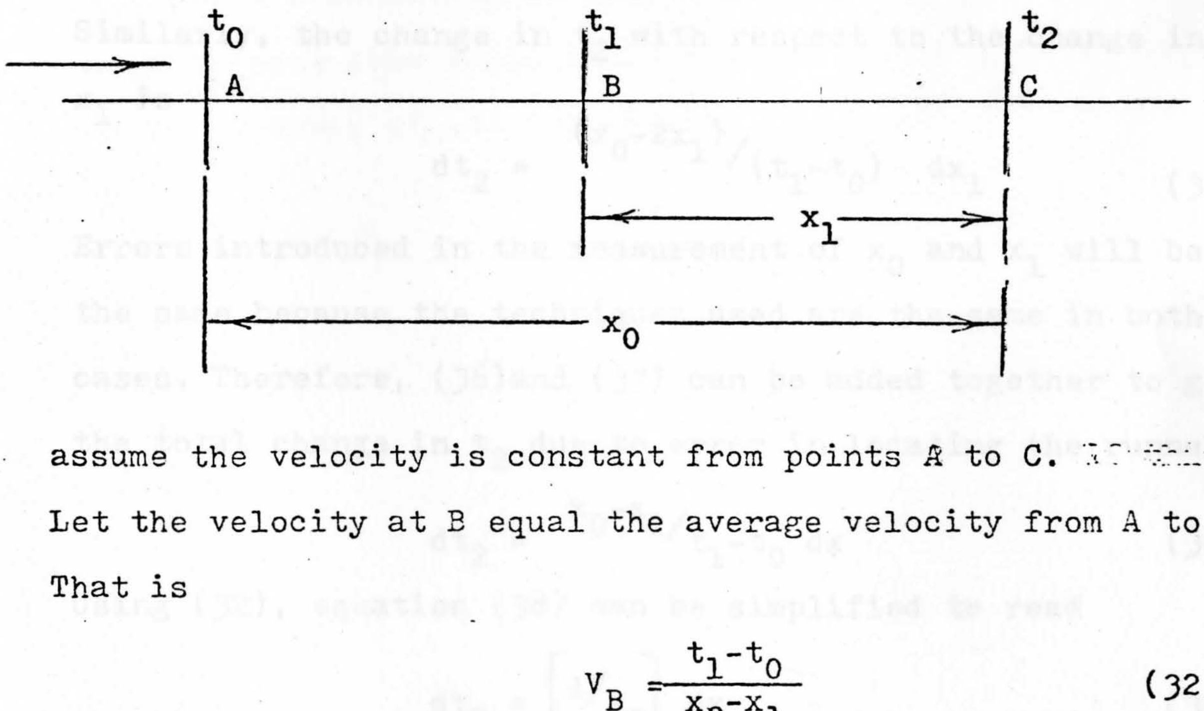
```
111     PRINT 113,I,NCHEST
113     FORMAT(10X,'THE FOLLOWING LINES ARE TIED',2I4)
        GO TO 105
112     PRINT 114
114     FORMAT(10X,'E VECTOR')
        PRINT 115,(E(I,1),I=1,11)
115     FORMAT(11G10.3)
        NCHEST=NCHEST+NLINE+2
        PRINT 116,NCHEST
116     FORMAT(10X,'PREDICTED CHEST LINE IS',1I4)
        NDIFF=NCHEST-JO
        PRINT 117,NDIFF
117     FORMAT(10X,'DIFFERENCE FROM ACTUAL',1I4)
1       CONTINUE
        STOP
        END
```

APPENDIX J

Accuracy Requirements In Locating The Chest Line

Due to the injection of noise into the system it is not possible for the algorithm to exactly identify the chest line in every case. However, in the particular application of track event timing it is not necessary to exactly identify the chest line. The following derivation shows the developement of the tolerances needed for the system.

Given the following diagram,



assume the velocity is constant from points A to C. Let the velocity at B equal the average velocity from A to B. That is

$$V_B = \frac{t_1-t_0}{x_0-x_1} \tag{32}$$

The finish time can then be calculated to be(if the finish line is at point C)

$$t_2 = t_1+x_1/\,V_B \tag{33}$$

From (32), equation (33) can be rewritten as

$$t_2 = t_1 + x_1(x_0-x_1)/(t_1-t_2) \tag{34}$$

or

$$t_2 = (x_1 x_0 - x_1^2)/(t_1-t_0) \tag{35}$$

The accuracy with which t_0 and t_1 are measured is determined by the accuracy of the timer clock. Therefore, error in t_2 can enter equation (35) only in the measurement of x_0 and x_1.

Taking the derivitive of t_2 with respect to change in x_0 gives

$$dt_2 = {}^{x_1}/(t_1-t_0)\ dx_0 \tag{36}$$

Similarly, the change in t_2 with respect to the change in x_1 is

$$dt_2 = {}^{(x_0-2x_1)}/(t_1-t_0)\ \ dx_1 \tag{37}$$

Errors introduced in the measurement of x_0 and x_1 will be the same because the techniques used are the same in both cases. Therefore, (36)and (37) can be added together to give the total change in t_2 due to error in locating the runners.

$$dt_2 = {}^{x_0-x_1}/_{t_1-t_0}\ dx \tag{38}$$

Using (32), equation (38) can be simplified to read

$$dt_2 = \left[{}^{1}/_{V_B}\right]\ dx \tag{39}$$

From (39), it is seen that the total error in t_2 is given by the error in locating the runner times the inverse of the velocity of the runner. In order to maintain a timing accuracy of .01 seconds the error in the finish time must be less than .005 seconds. Using this value and a worst case velocity for the hundred yard dash, it is possible to calculate the maximum allowable error in locating the chest line. The maximum error in dx was calculated to be 4.2 inches. This corresponds to missing the actual chest line by two scans on either side. From this, it is possible to establish a bandwidth of body segments which if selected as the chest line would still allow the runner to be timed to an accuracy of .01 seconds.

BIBLIOGRAPHY

1. Taft, Jeffrey D.,"An Optimally-Weighted Hybrid Stochastic Decision Algorithm for Pattern Recognition in Noisy Digitized Video Images", Master's Thesis, YSU, 1978.

2. Durbin, Bruce B., ed., The National Federation Track and Field Rules, 1978 ed. National Federation of State High School Associations.

3. Tou, Julius T. and R.C. Gonzalez, Pattern Recognition Principles, Addison-Wesley Publishing Company, 1974.

4. Fam, A.T. and J. Sklansky, "Cellularly Straight Images and Hausdorff Metric", Proceedings of the 1977 Pattern Recognition and Image Processing Conference, June 6-8 RPI Troy, NY.

5. Bongard, M., Pattern Recognition, Spartan Books, 1970.

6. Chen, Chi-hau, Statistical Pattern Recognition, Spartan Books, New Jersey, 1973.

7. Young, Tzay Y. and Thomas W. Calvert, Classification, Estimationand Pattern Recognition, American Elsevier Publishing Co. Inc., New York, 1974.

8. DeRusso, Paul M., Rob J. Roy and Charles M. Close, State Variables for Engineers, John Wiley and Sons,Inc., New York, 1965.

9. Bailey, David D.,"A Physical Model for a Real Time Pattern Recognition System", Proceedings of the Modeling and Simulation Conference, Pittsburgh, Pa., 1978.

10. Fu, K.S.,"Syntactic Methods and Image Processing", Proceedings of the 1977 Pattern Recognition and Image Processing Conference, June 6-8, RPI Troy, NY.