To the Graduate Council:

I am submitting herewith a thesis written by Michael E. Casey entitled "Automated Particle Counting with Applications to Neurology." I recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

E. L. Hall, Major Professor

We have read this thesis
and recommend its acceptance:

Robert E. Bartlemeier

R. C. Gonzalez

Accepted for the Council:

Vice Chancellor
Graduate Studies and Research

Thesis
W
C222
Cop. 2

AUTOMATED PARTICLE COUNTING WITH APPLICATIONS TO NEUROLOGY

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Michael E. Casey

August 1977

1328436

ACKNOWLEDGEMENTS

The author is grateful for the assistance, guidance, and patience of Dr. Ernest L. Hall, committee chairman. Appreciation is also expressed to the members of the committee, Dr. R. C. Gonzalez and Dr. R. E. Bodenheimer for their guidance and contributions.

Recognition is given also to Dr. Gary Frykman of Loma Linda University Hospital without his support this thesis would never have been conceived.

Appreciation is extended to Vicki Bohanan whose patience and skill resulted in a final draft of the text.

ABSTRACT

In this thesis an algorithm is developed to count regular shaped particles in digitized images. The algorithm was applied to counting axons in nerve fasciculi regions.

Images of the data are digitized using a special TV camera. The digitized data is then preprocessed before sent to the recognizer.

Particles are located in the digitized image by matching them with a representative mask. Closeness of the match is determined using binary correlation.

After the slides of the fasciculi regions have been digitized, the system runs completely automatically, taking in data from magnetic tape and storing the results on magnetic disks. Using manually counted regions as a basis for comparison the system is more than 90 percent accurate for the three nerves used.

This algorithm shows promise as a useful tool in any application where multiple particles are counted. As illustrated in this thesis one direct application is in the field of neurology.

TABLE OF CONTENTS

CHAPTER	PAGE
1. INTRODUCTION	1
2. EDGE ENHANCEMENT	5
3. CORRELATION	14
4. THEORETICAL EVALUATION OF COUNTING	22
5. EXPERIMENTAL RESULTS	39
BIBLIOGRAPHY	50
APPENDIX	52
VITA	72

LIST OF FIGURES

FIGURE		PAGE
1.1.	Nerve axon counting system	3
2.1.	Block diagram of image digitizer	6
2.2.	Typical section of a fascicle region	7
2.3.	Thresholded image of fascicle region	9
2.4.	Edge enhanced image.	12
2.5.	Results of thresholding an edge enhanced region.	13
3.1.	A "typical axon" mask used for correlation	18
3.2.	Cursor that can be moved over an image	19
3.3.	Autocorrelation surface.	20
3.4.	A typical mask region.	21
4.1.	Probability density function for correlation of background	26
4.2.	Probability of false alarm for a threshold detector.	27
4.3.	Probability of success for a threshold detector.	28
4.4.	Receiver operating characteristic curves for a threshold detector	30
4.5.	Probability density function for correlation values after peak detection	33
4.6.	Probability of false alarm after peak detection.	34
4.7.	Receiver operating characteristic curves for detector after peak detection	36
4.8.	Probability density function of correlation surface.	37
5.1.	Program flow chart	40
5.2.	A typical nerve showing individual frames.	41
5.3.	A typical frame as digitized	42
5.4.	Least squares fit of Machine vs. Manual for NN3 Proximal	48

CHAPTER 1

INTRODUCTION

Modern digital computers are widely used to take control of simple but tedious and time consuming tasks thus freeing humans to do more creative work. One field of study which is concerned with "teaching" computers to do the simple tasks is pattern recognition. By applying the principles of pattern recognition to different areas of research, computers have played a major role in pushing back the frontiers of science. Medicine is one of the fields of research that the computer is just beginning to enter.

It is a well known fact that arthropods have the capability to regenerate a severed limb. Since humans do not have this capability, a respectable amount of research deals with repairing an appendage after an accident. One of the more lacking areas is that of nerve regeneration. Normally, the severed limb is attached rather crudely by simply applying sutures to the outer layers of tissue, major muscles, ligaments and blood vessels. Usually no attempt is made to match nerve clusters, or at most only a single suture is applied to the cluster. A theory developed by Dr. Gary Frykman of Loma Linda University Hospital is that nerves should regenerate better if the individual fascicle regions are matched and sutured. A fascicle region is a bundle of nerve fibers or axons.

To test Dr. Frykman's theory, a count of the nerve axons in a particule fascicle region proxmal to a cut was to be compared to a count of axons distal to the cut in the same fascicle region. Since there are between sixteen thousand and twenty-five thousand axons in a typical

nerve under consideration, manually counting the axons is a very tedious and boring task for whoever is unlucky enough to be required to do it. Therefore a computer program, the subject of this thesis, was developed to automatically count the axons in a fascicle region.

Previous research in the area includes a system by Kasvand [1] that counts and estimates the size of the nerve fibers. The basis for the system developed by Kasvand is locating the centroid of the fiber by computing a local maximum of the sum of distances to the nearest contours. As implemented, the system required a very high resolution quantization of the input image. Also computing the centroid of the fiber is a very time consuming process.

A different technique for counting the nerve fibers in the optic nerve of a primate was presented by Potts et al. [2]. This technique involved convolving a mask over the entire picture and counting the points where the mask matched. The picture data used was quantized into gray levels, and the mask was simply an x shaped method of weighting the center point and the surrounding points. Again, the picture was quantized into a very high resolution (960 x 630 points) data set. The technique was reasonably fast considering that the computer used was a PDP-10.

The system as presented in this thesis uses a medium resolution quantization of data and a process by which the image is converted to a binary representation (black or white). A mask is used to detect the axons, however, unlike the method presented above, the mask is simply a binary representation of a typical axon. Figure 1.1 shows the block diagram of the system as implemented.

The thesis attempts to present the description of the system in the following manner. Chapter 2 contains a description of the

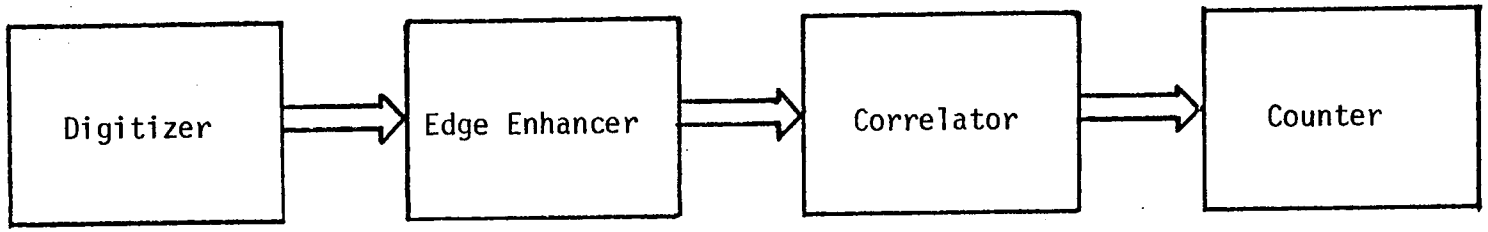


Figure 1.1. Nerve axon counting system.

digitization, display and preprocessing required to ready the data for input to the recognizer. Chapter 3 contains a description and derivation of the matching technique used to recognize an axon. A theoretical evaluation of the system is presented in Chapter 4. Finally, Chapter 5 contains the results of comparing the system with data counted manually. An appendix is included that contains a listing of the program.

CHAPTER 2

EDGE ENHANCEMENT

In order to be able to process data that is in the form of an image, the data has to be digitized and stored in bulk storage. Digitization consists of sampling the intensity of a point in the image. The device used in this system is produced commercially by Spatial Data Systems. Consisting of a television camera and analog to digital converter, the device allows input directly to the computer. Figure 2.1 shows a block diagram of the Spatial Data Systems Digital Camera.

Information flow in the Camera Eye starts when the image is sampled by the vidicon. In order to allow the camera to be focused, the image is displayed on the monitor. The image is then digitized by the analog to digital circuitry and made available to the computer. Data from the Camera Eye is handled by a special program written by B. A. Fittes [9].

Image data stored in bulk storage can be displayed for viewing by a program [8] that handles data transfer from the computer to the Data Disc Display System. This device consists of a fixed head disk rotating at thirty revolutions per second, and a digital to analog converter. Image data is stored on the disk by the computer. Data which is stored on the disk is then displayed on the monitor after going through the digital to analog converter and video generation circuit.

Once the data is stored in the computer, the processing can begin. Figure 2.2 shows a typical section of a fascicle region. Due to the non-linearities in nature, the camera, the film and the digitizer, the average

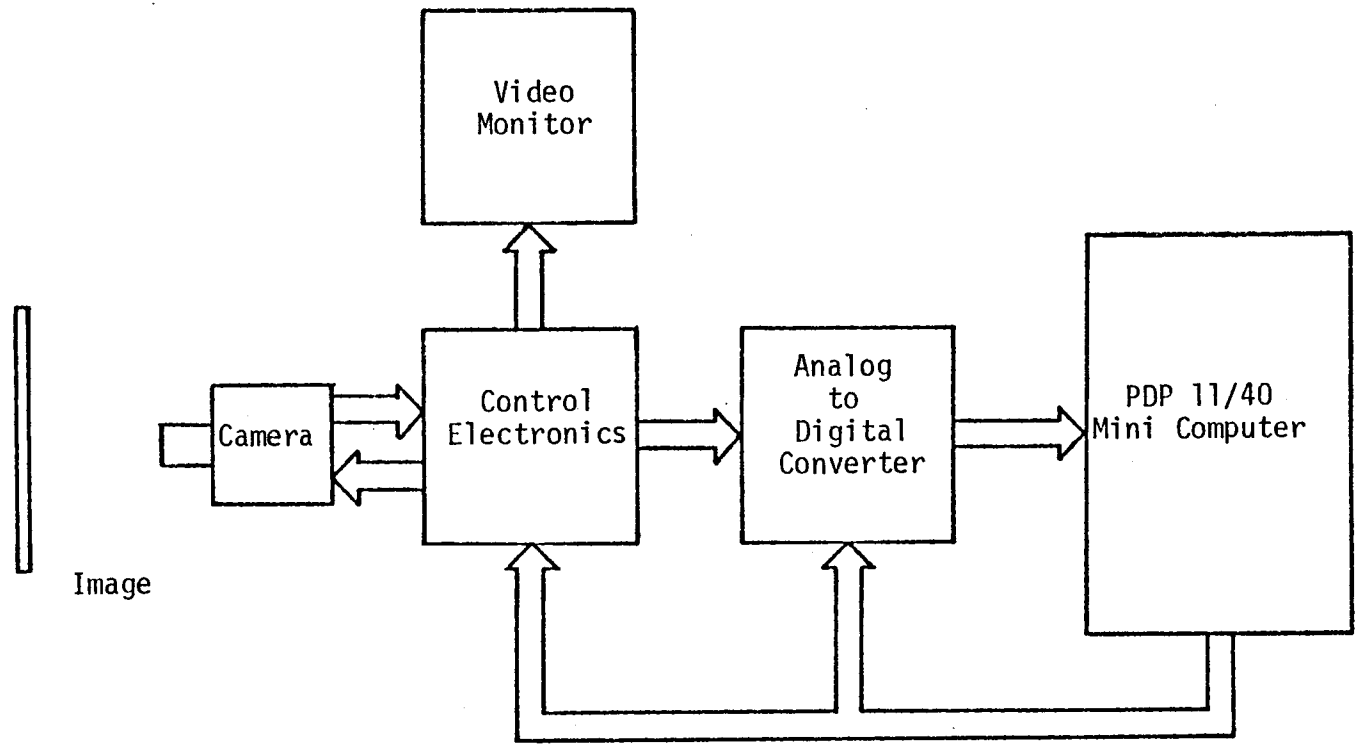


Figure 2.1. Block diagram of image digitizer.

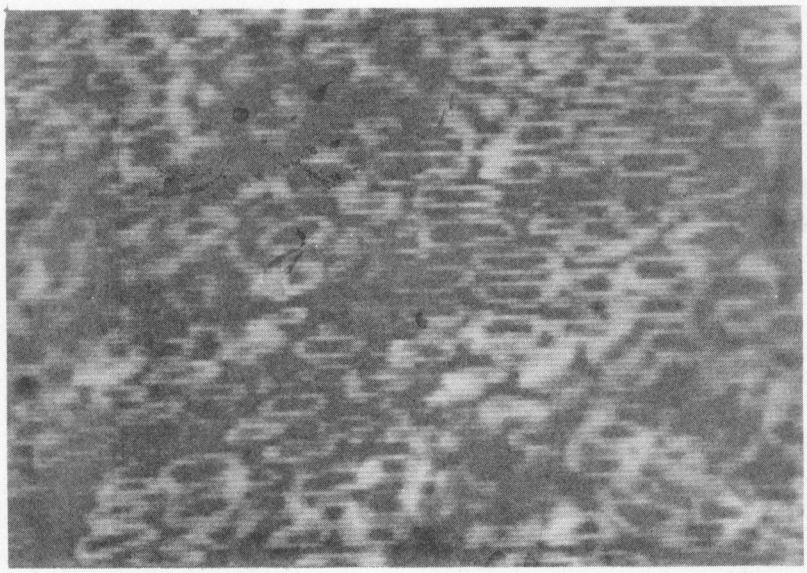


Figure 2.2. Typical section of a fascicle region.

brightness of the image varies across the image. This is demonstrated by thresholding the image. Thresholding assigns the value of white to every pixel whose value is above the threshold and black to every pixel below that value. By defining the input and output images as functions of x and y thresholding can be stated mathematically as:

$$g(x, y) = \begin{cases} 1 & \text{for} \quad f(x, y) \geq T \\ 0 & \text{for} \quad f(x, y) < T \end{cases}$$

where g(x, y) is the resultant binary image and f(x, y) is the input image.

Ideally, thresholding of the image should segment the image into axons (white) and background (black). Figure 2.3 demonstrates that this is not always the case as some of the background is included and some of the axons are deleted.

Nonuniformity in average brightness can be overcome though. Define average brightness of a 3 x 3 pixel region in the image as

$$g(x, y) = \frac{1}{9} \sum_{i=-1}^{1} \sum_{j=-1}^{1} f(x + i, y + j) \qquad (2.1)$$

where g(x, y) is the image that results from the accumulation of the average brightnesses over the input picture f(x, y). Since the problem is nonuniformities in the average brightness of the image, the solution is to subtract the average brightness from the image.

$$h(x, y) = f(x, y) - g(x, y) \qquad (2.2)$$

The resultant image h(x, y) contains information of the differences of adjacent pixels. Substituting Eq. (2.1) into Eq. (2.2) produces:

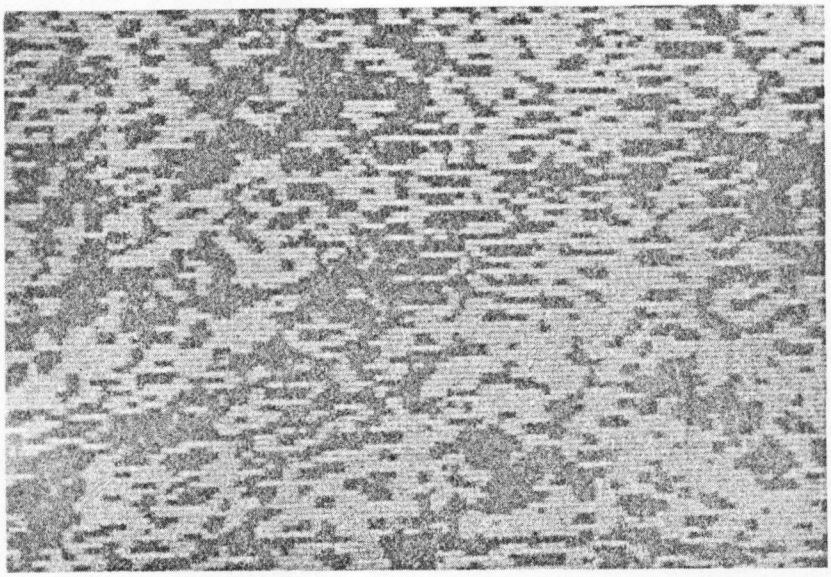


Figure 2.3. Thresholded image of fascicle region.

$$h(x, y) = f(x, y) - \frac{1}{9} \sum_{i=-1}^{1} \sum_{j=-1}^{1} f(x + i, y + j) \qquad (2.3)$$

or

$$h(x, y) = \frac{8}{9} f(x, y) - \frac{1}{9} \sum_{i=-1}^{1} \sum_{j=-1}^{1} f(x + i, y + j),\ i, j \neq 0 \qquad (2.4)$$

Physically, h(x, y) is brightest where the transition between adjacent pixels is the largest and darkest where the transition is the smallest. In fact, Eq. (2.4) states mathematically what is well known by photographers as "unsharp masking." [5].

Students of digital filtering will recognize Eq. (2.4) as a digital highpass filter of the form:

$$h(x, y) = f(x, y) * g(x, y) \qquad (2.5)$$

Convolution of f(x, y), the input picture, and a filter g(x, y) is denoted by "*". The filter g(x, y) then is defined by:

$$\begin{bmatrix} -\frac{1}{9} & -\frac{1}{9} & -\frac{1}{9} \\ -\frac{1}{9} & +\frac{8}{9} & -\frac{1}{9} \\ -\frac{1}{9} & -\frac{1}{9} & -\frac{1}{9} \end{bmatrix}$$

Another form of Eq. (2.5) is,

$$h(x, y) = \sum_{i=0}^{2} \sum_{j=0}^{2} f(x + i - 1, y + j - 1)g(i, j) \qquad (2.6)$$

where the -1 factor included in the subscript is for compatability with Eq. (2.2).

The result of this operation on an image is illustrated in Figure 2.4. The reader should note that the image mainly contains lines where there are intensity edges in the original image. For comparison with Figure 2.3, Figure 2.5 shows the results of thresholding Figure 2.4. Here the segmentation is much more defined in terms of background and axons.

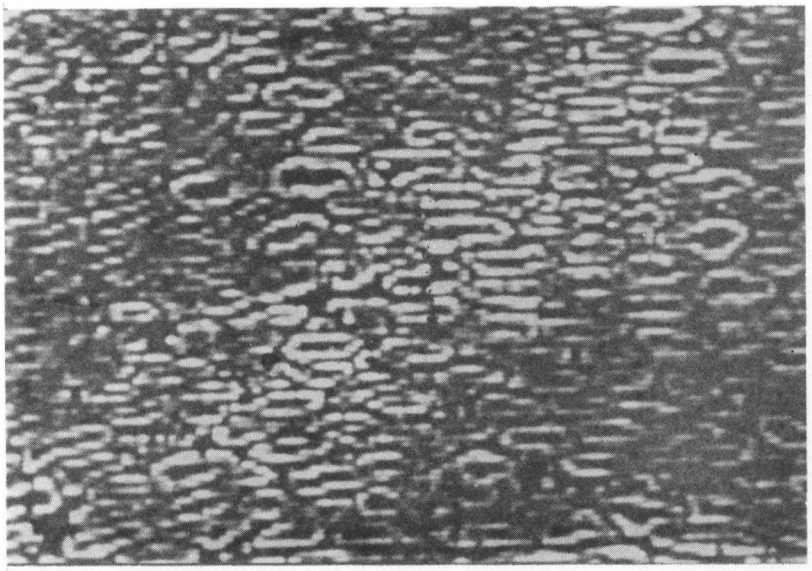


Figure 2.4. Edge enhanced image.

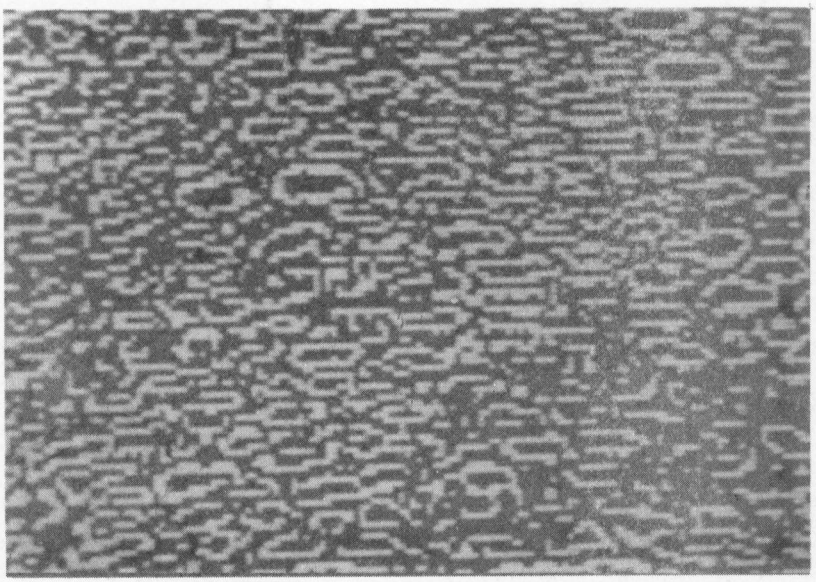


Figure 2.5. Results of thresholding an edge enhanced region.

CHAPTER 3

CORRELATION

Enhancing one of the more prominent features of the axons by edge detection is just a preprocessing step in the counting. In order to be able to count an axon, the computer must first "find" the axon. One method of locating axons is to compare each region of the picture with a representative of an axon.

Consider the comparison of a region Q of a picture P and a representative mask M. A measure of mismatch is the absolute error:

$$\varepsilon = \sum_{Q} \sum |P(i, j) - M(i, j)| \tag{3.1}$$

or the squared error:

$$\varepsilon = \sum_{Q} \sum (P(i, j) - M(i, j))^2 \tag{3.2}$$

The larger the error, the larger the mismatch between P and M. If ε is to be small, P must closely match M. Since the right hand of Eq. (3.2) is a squared quantity ε will always be greater than or equal to zero.

$$0 \leq \sum \sum (P(i, j) - M(i, j))^2 \tag{3.3}$$

Expanding Eq. (3.2) produces:

$$\varepsilon = \sum_{Q} \sum P^2(i, j) - 2\sum_{Q} \sum P(i, j)M(i, j) + \sum_{Q} \sum M^2(i, j) \tag{3.4}$$

From Eq. (3.4) it can be seen that if P and M are constant, for the error to be small $\sum_{Q} \sum P(i, j)M(i, j)$ must be large. Therefore, a measure of match might be:

$$R = \sum_{Q} \sum P(i, j)M(i, j) \tag{3.5}$$

If all the regions over the entire picture are considered then Eq. (3.5) becomes:

$$R(i, j) = \sum_{k=0}^{n-1} \sum_{\ell=0}^{n-1} P(i + k, j + \ell)M(k, \ell) \tag{3.6}$$

which is a well known digital correlation formula. Equation (3.6) can be normalized by first substituting Eq. (3.4) into Eq. (3.3) then rearranging.

$$0 \leq \sum_{Q} \sum P^2(i, j) - 2\sum_{Q} \sum P(i, j)M(i, j) + \sum_{Q} \sum M^2(i, j) \tag{3.7}$$

or

$$2\sum_{Q} \sum P(i, j)M(i, j) \leq \sum_{Q} \sum P^2(i, j) + \sum \sum M^2(i, j) \tag{3.8}$$

If P and M are allowed to go negative then a lower bound can be set by considering:

$$0 \geq -\left[\sum_{Q} \sum P^2(i, j) - 2\sum_{Q} \sum P(i, j)M(i, j) + \sum_{Q} \sum M^2(i, j)\right] \tag{3.9}$$

rearranging produces:

$$-\sum \sum P^2(i, j) - \sum \sum M^2(i, j) \leq -2\sum_{Q} \sum P(i, j)M(i, j) \tag{3.10}$$

Then to normalize divide by

$$-1 \leq \frac{\sum_{Q} \sum P(i, j)M(i, j)}{\frac{1}{2}\left[\sum_{Q} \sum P^2(i, j) + \sum \sum M^2(i, j)\right]} \leq 1 \tag{3.11}$$

If Eq. (3.11) is computed for all regions of the picture as a measure of match, R is known as the normalized correlation:

$$R(i, j) = \frac{\sum_{k=0}^{} \sum_{\ell=0}^{} P(i + k, j + \ell)M(k, \ell)}{\frac{1}{2}\left[\sum_{k=0}^{n-1} \sum_{\ell=0}^{n-1} P^2(i + k, j + \ell) + \sum_{k=0}^{n-1} \sum_{\ell=0}^{n-1} M(i, j)\right]} \quad (3.12)$$

If the picture and the mask are binary, black and white, multiplication of the binary pixels can be approximated by raising negative one to the exclusive-or power:

$$P(i, j) \times M(i, j) = -1^{[P(i, j) \vee M(i, j)]} \quad (3.13)$$

The exclusive-or operator is denoted by a $\vee$. When P and M match the exclusive-or is zero and the product is one, when P and M do not match, then the product is negative one. The normalized correlation then becomes:

$$R(i, j) = \frac{\sum_{k=0}^{n-1} \sum_{\ell=0}^{n-1} -1^{P(i + k, j + \ell) \vee M(i, j)}}{\frac{1}{2}[a + b]} \quad (3.14)$$

where

$$a = \sum_{k=0}^{n-1} \sum_{\ell=0}^{n-1} -1^{P(i + k, j + \ell) \vee P(i + k, j + \ell)}$$

and

$$b = \sum_{k=0}^{n-1} \sum_{\ell=0}^{n-1} -1^{M(k, \ell) \vee M(k, \ell)}$$

The denominator in Eq. (3.14) becomes just the number of pixels in the region because the exclusive-or of a number with itself is always zero. For example, the double sum:

$$\sum_{k=0}^{n-1} \sum_{\ell=0}^{n-1} (-1)^0 = \sum_{k=0}^{n-1} \sum_{\ell=0}^{n-1} (1) = n^2) \quad (3.15)$$

reduces to n^2, the number of pixels in the region. Using Eq. (3.15), Eq. (3.14) then becomes:

$$R(i, j) = \frac{1}{n^2} \sum_{k=0}^{n-1} \sum_{\ell=0}^{n-1} (-1)^{[p(i + k, j + \ell) \forall M(i, j)]} \quad (3.16)$$

The binary correlation was used in the particle counting system because the product is now simply an exclusive-or and the addition can be done in a binary counter.

The reference mask that was used is shown in Figure 3.1. This mask was derived by averaging several axons from the images. A program was written that allowed the user to move a cursor about on the image. Each time the cursor was positioned, the axon at that point was sampled, and added to the total of all of them. As the program was terminated, the average was printed on the lineprinter. Figure 3.2 shows the cursor which was controlled by the joystick.

Figure 3.3 shows the auto-correlation surface produced by the cross-correlation of the reference mask with itself. Notice how the surface has a very definite peak at the point where the two images are actually registered.

Because the correlation has a tendency to peak, as demonstrated in Figure 3.3, the criterion to be considered as an axon requires the region to produce a peak. The peak regions were selected by looking at all five by five regions on the correlation surface. The position was a region if the center point was larger than the surrounding twenty-four points. A new image was created by putting the correlation value of the peak into a blank image. Figure 3.4 shows a typical peak image. Each bright spot in the image corresponds to a tentative axon.

0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0
0	0	1	1	1	1	0	0
0	1	1	0	0	1	1	0
0	1	1	0	0	1	1	0
0	0	1	1	1	1	0	0
0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0

Figure 3.1. A "typical axon" mask used for correlation.

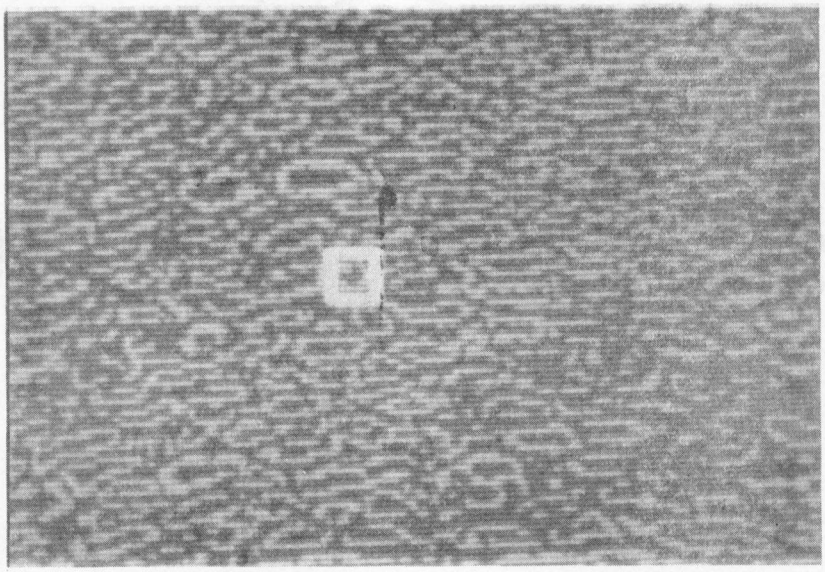


Figure 3.2. Cursor that can be moved over an image.

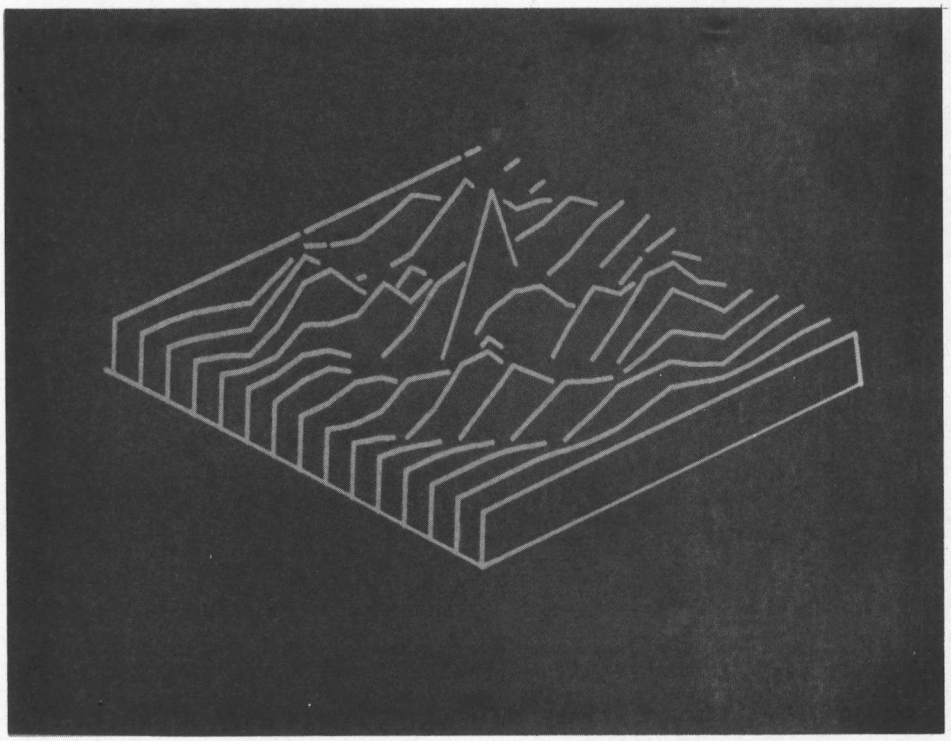


Figure 3.3. Autocorrelation surface.

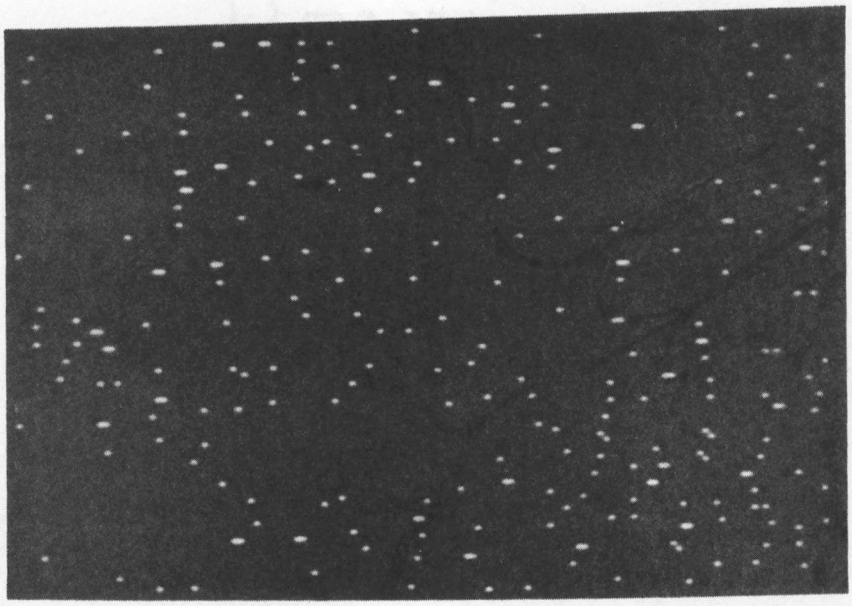


Figure 3.4. A typical peak region.

CHAPTER 4

THEORETICAL EVALUATION OF COUNTING

In order to count the nerve axons, a method for determining system accuracy must first be developed. In this chapter, a theoretical method of determining the receiver operating curves for this system is derived, then an experimental evaluation is made.

Consider the measure of match, the binary correlation derived in Chapter 3:

$$R(i, j) = \sum_{k=0}^{n-1} \sum_{\ell=0}^{n-1} (-1)^{P(i + k, j + \ell) \nabla M(i, j)} \qquad (4.1)$$

This measure is essentially the number of points in the picture, P, that match those corresponding points in the mask, M, minus the number of mismatches. Let U be the set of points in the mask, M, that are one, and Z be the set in M that are zero. Next define $N_U(1)$ to be the number of ones in the picture, P, that match U; $N_U(0)$, the number of zeros in P that correspond to U; $N_Z(1)$, the number of ones in P corresponding to Z; and finally $N_Z(0)$, the number of zeros in P that match the set of zeros, Z. For a given region, the measure of match now becomes:

$$R = N_U(1) + N_Z(0) - N_U(0) - N_Z(1) \qquad (4.2)$$

If n is the total number of points in the mask, then

$$n = N_U(1) + N_Z(0) + N_U(0) + N_Z(1) \qquad (4.3)$$

must hold. Rearranging Eq. (4.3) and substituting into Eq. (4.2) produces:

$$R = 2[N_U(1) + N_Z(0)] - n \tag{4.4}$$

A new measure of match Q can be defined if $Q = \frac{R \div n}{2}$. Q has exactly the same shape as R but on a different interval. The correlation now becomes:

$$Q = N_U(1) + N_Z(0) \tag{4.5}$$

Define q to be the fractional number of ones in the mask, so qn is the number of ones under consideration. In the background, assume the points are ones and zeros and occur independently with probabilities of p and 1 - p, respectively.

The event that h out of k picture points in the background are one is then binomially distributed with mean kp and standard deviation $\sqrt{kp(1 - p)}$. For sufficiently large sample sizes, a binomial distribution can be approximated by a normal distribution with mean μ and standard deviation σ [6, 7]. The accuracy of the approximation is much better for p near 0.5.

Assuming that n = 64 is a sufficiently large sample to make the assumptions valid, it is possible to develop a distribution for the match measure Q. $N_U(1)$, the number of ones in P that match U is a random variable that is normally distributed with mean nqp and standard deviation $\sqrt{nqp(1 - p)}$. This is because nq points are being sampled each with a probability p of being sampled. Similarly, $N_Z(0)$ has a mean $\mu_{X + y} = \sqrt{n(1 - q)p(1 - p)}$.

If two random variables X and Y have means μ_X and μ_Y and standard deviations σ_X and σ_Y then the random variable X + Y has a mean $\mu_{X + Y} = \mu_X + \mu_Y$ and a standard deviation $\sigma_{X + Y} = \sqrt{\sigma_X^2 + \sigma_Y^2}$. Thus, the measure of match, Q, has a normal distribution with parameters

$$\mu_{Q_B} = \mu_{N_U}(1) + \mu_{N_Z}(0) \tag{4.6}$$

and

$$\sigma_{Q_B} = \sqrt{\sigma^2_{N_U}(1) + \sigma^2_{N_Z}(0)} \tag{4.7}$$

Substituting into Eq. (4.3) produces the mean of the distribution:

$$\mu_{Q_B} = qnp + (1 - q)n(1 - p)$$

$$\mu_{Q_B} = n(2qp - q - p + 1) \tag{4.8}$$

Similarly, the standard deviation can be computed to be

$$\sigma_{Q_B} = \sqrt{np(1 - p)} \tag{4.9}$$

To develop the worst case characteristics, assume a threshold detector. This means that an axon is recognized any time that Q is larger than or equal to the threshold, t. Therefore, if the region under question is in the background of the picture, the probability that the correlation, Q, equals the threshold is given by the function:

$$f(t) = \frac{1}{\sqrt{2\pi\sigma^2_{Q_B}}} \exp[-(t - \mu_{Q_B})^2/\sigma^2_{Q_B}] \tag{4.10}$$

The probability that the correlation is greater or equal to the threshold, the probability of false alarm, is given by:

$$p_{FA}(t) = \sum_{x=t}^{n} f(t) \tag{4.11}$$

A good assumption of the probability of occurrence of a one, p, is 1/2. If so, then $\mu_{Q_B} = 32$, and $\sigma_{Q_B} = 4$. Equation (4.11) now becomes:

$$p_{FA}(t) = \frac{1}{\sqrt{2\pi}} \sum_{x=t}^{n} \exp[-(X - 32)^2/16] \qquad (4.12)$$

Equation (4.10) is shown in Figure 4.1 while Eq. (4.12) is illustrated in Figure 4.2.

In order to theoretically develop a measure of success, consider that the mask is exactly positioned over a matching region. Let N_m be the number of points in the picture, P, that match the mask M. In other words, the correlation $Q = N_m$.

Let p now be the probability that a point in the picture will match the corresponding point in the mask, and (1 - p) be the probability that it will not match. The random variable N_m then has a binomial distribution given by:

$$f_{N_m} = \frac{n!}{(n - k)!k!} p^k (1 - p)^{n - k} \qquad (4.13)$$

where k is the number of successes or matching points expected out of n points observed. The distribution has a mean μ_{N_m} given by np and a standard deviation, $\sigma_{N_m} = \sqrt{np(1 - p)}$. Equation (4.13) gives the probability that a value of the correlation Q will occur. The probability of success then is the probability that the correlation will be larger than a certain threshold. From Eq. (4.13), the probability of success for a threshold detector is:

$$p_s(t) = \sum_{k=t}^{64} \frac{64!}{(64 - k)!k!} p^k (1 - p)^{64 - k} \qquad (4.14)$$

Equation (4.14) is illustrated in Figure 4.3 for values of p of 0.5, 0.6, 0.7, 0.8, and 0.9. Notice how the cutoff moves right with larger

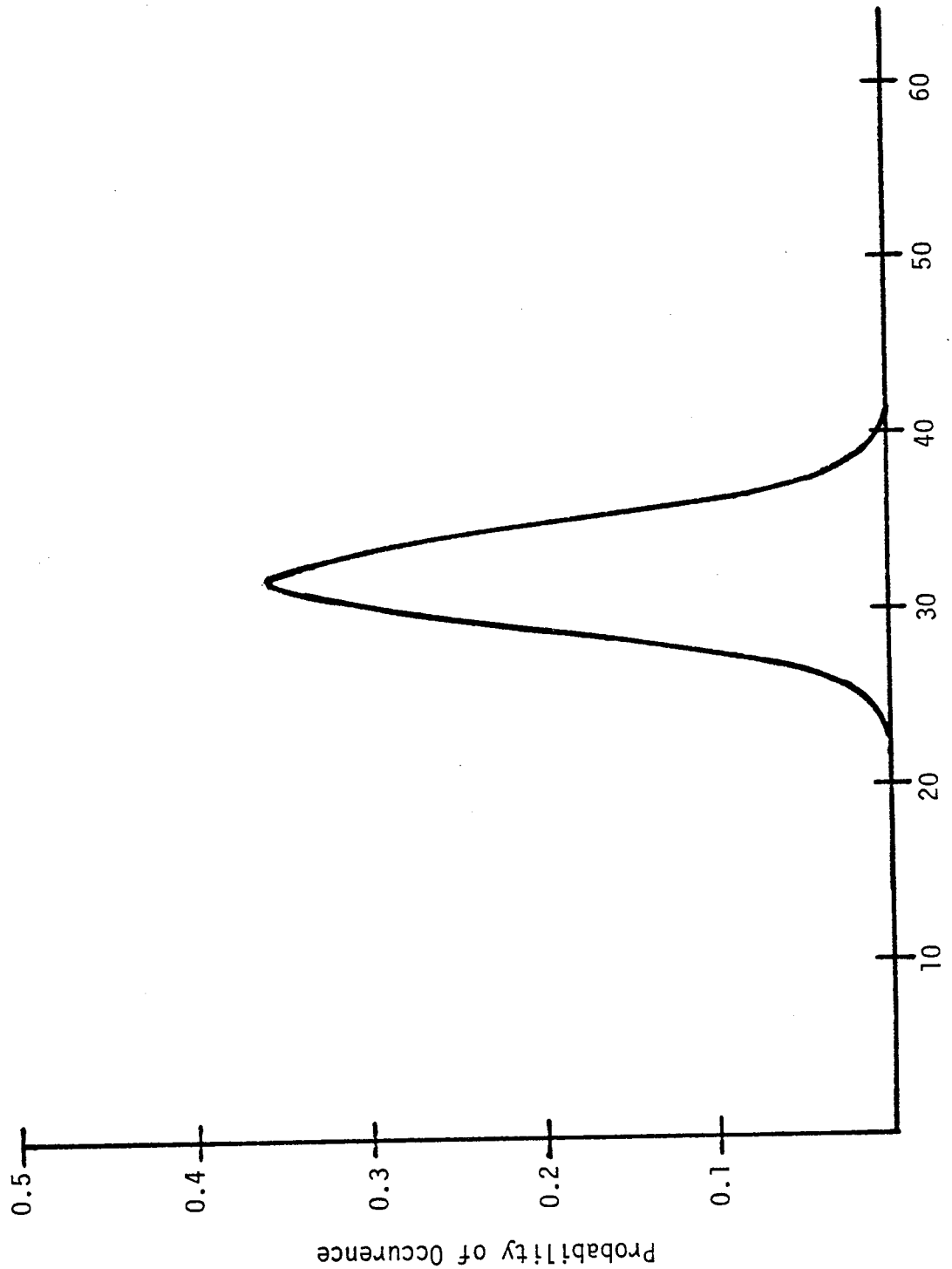


Figure 4.1. Probability density function for correlation of background.

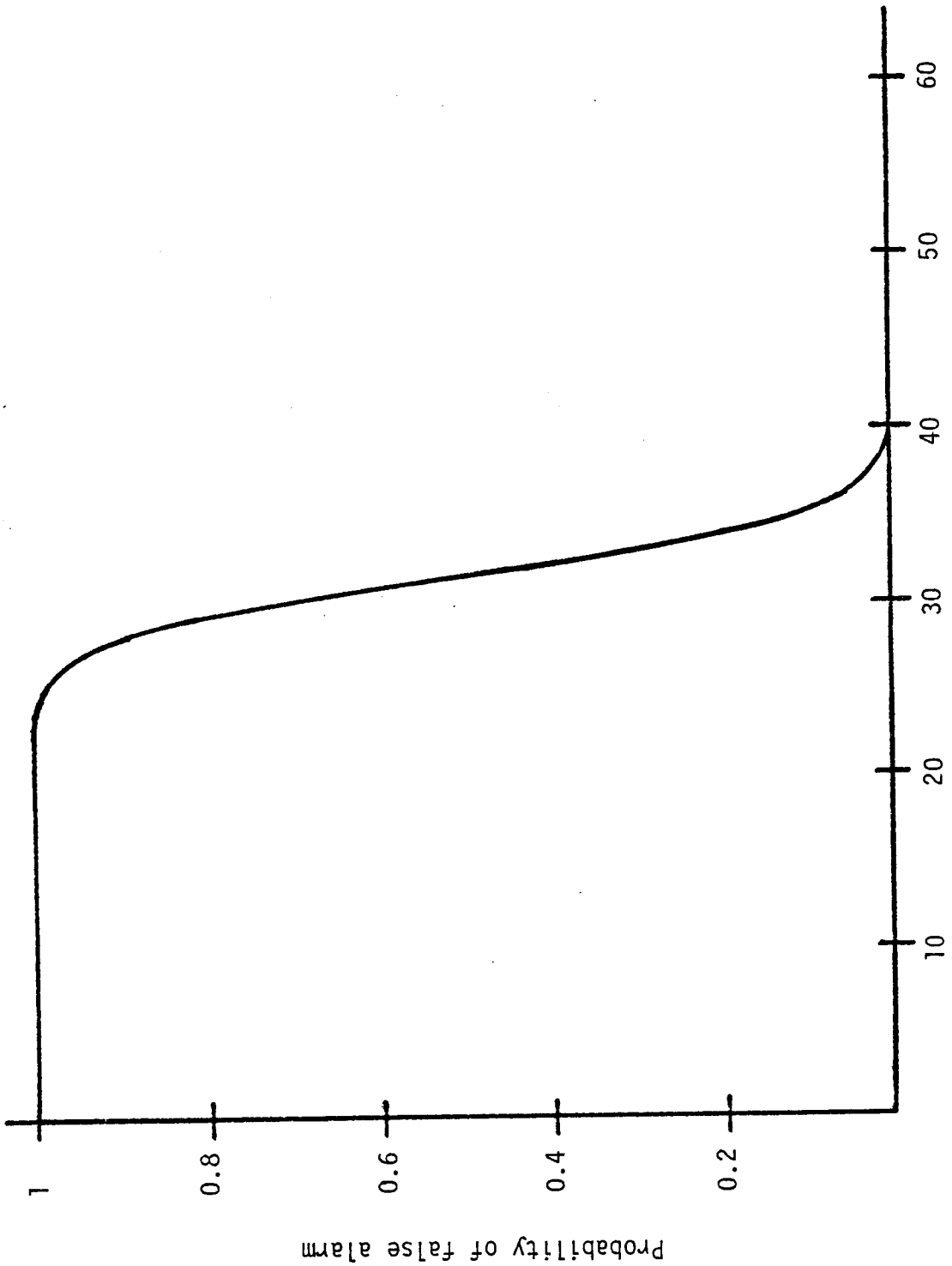


Figure 4.2. Probability of false alarm for a threshold detector.

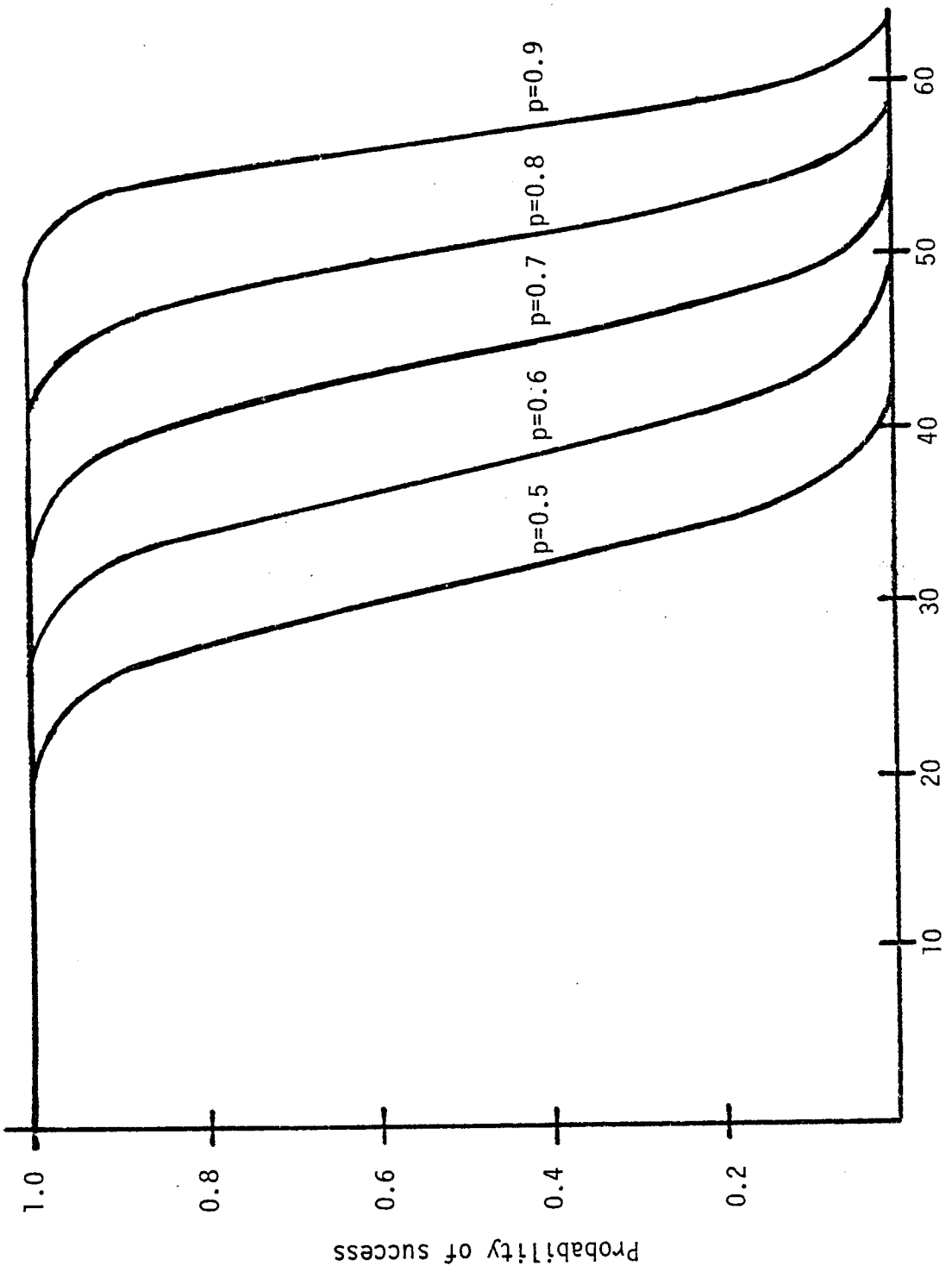


Figure 4.3. Probability of success for a threshold detector.

values of the probability. In fact, the ideal case with no noise where p = 1 can be computed.

$$p_s(t) = 1 \tag{4.15}$$

This is because $(1 - p)^{64 - k} = 0^i = 0$ except when i = 0, then $0^0 = 1$.

A widely used and accepted method of displaying the operating characteristics for a system is the receiver operating characteristics (R.O.C.) curves. The R.O.C. curves for a threshold detector can be drawn using the probability of false alarm (Eq. (4.14)) for the same threshold. Figure 4.4 shows the R.O.C. curves for the threshold detector. The family of curves shown is obtained by using probabilities that a particle will match having values of 0.5, 0.6, and 0.7. For each higher value of probability, the curve moves closer to the ideal in the upper left corner of the graph.

The R.O.C. curves shown were developed for a threshold detector for the entire correlation surface. However, the system as implemented looks for peaks in the correlation surface to identify possible axons. The peak test, as described in Chapter 3, considers a 5 X 5 pixel region. If the center point in that region is larger than its twenty-four neighbors then that point is a peak.

Consider again the probability distribution function for the probability that the correlation on a background region has a certain value.

$$f_{Q_B}(t) = \frac{1}{4\sqrt{2\pi}} e^{\frac{-(t - 32)^2}{16}} \tag{4.16}$$

If t is some value of the correlation, then the probability that the value occurs is given by $f_{Q_B}(t)$. For discrete values of Q_B, the

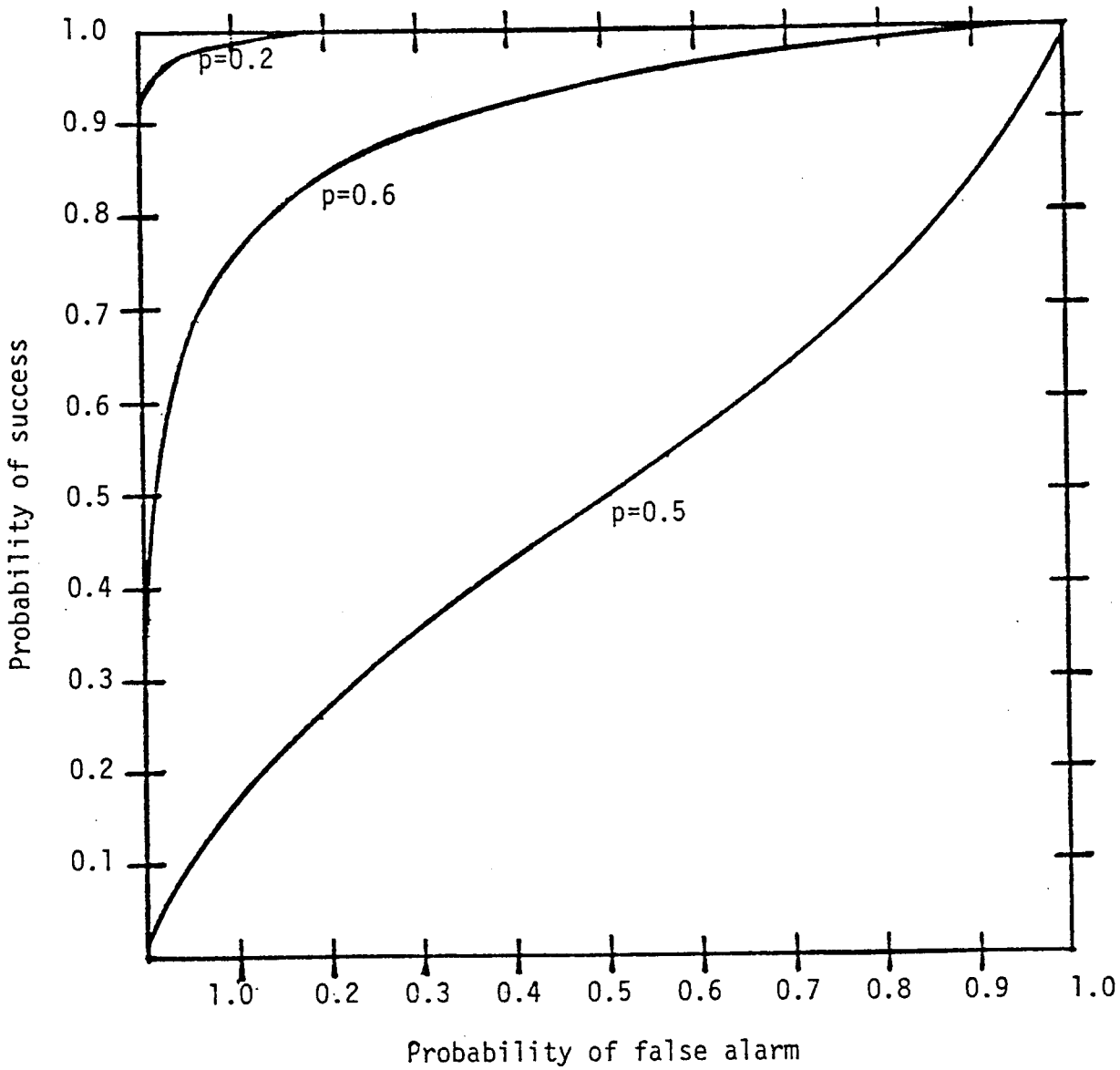


Figure 4.4. Receiver operating characteristic curves for a threshold detector.

correlation, the probability that Q_B is less than or equal to t is given by the discrete cumulative distribution function:

$$F(t) = \sum_{k=0}^{t} f(X) \tag{4.17}$$

In order to satisfy the criterion imposed by the peak detector for some value of the correlation, Q_B, twenty-four surrounding points must be less than Q_B. Using Eq. (4.17) it is possible to compute the probability that the surrounding points will indeed be less than Q_B. Since the correlation only assumes integer values from zero to sixty-four, let t in Eq. (4.17) be Q_B - 1. Therefore the probability that the correlation value in one of the surrounding points is less than Q_B can be found. Assuming that values of the correlation occur independently, the probability that two values will occur simultaneously is the product of the two individual probabilities. This suggests that the probability that twenty-four points be less than or equal to some value is the product of their individual probabilities. From Eq. (4.17) the probability that the twenty-four surrounding points be less than or equal to Q_B - 1 is:

$$P_a = [F(t)]^{24} = \left[\sum_{x=0}^{Q_B-1} f(x)\right]^{24} \tag{4.18}$$

Again the probability that Q_B will occur is given by Eq. (4.16). The probability for some point to pass the peak test is the product of the probability of correlation value of that point and the probability that the twenty-four surrounding points be less than or equal to that value minus one and given by:

$$P_T = f(Q_B) \cdot P_a \tag{4.19}$$

Substituting into Eq. (4.19) produces:

$$f_T(Q_B) = \left[\frac{1}{4\sqrt{2\pi}} e^{\frac{-(Q_B - 32)^2}{16}}\right]\left[\sum_{x=0}^{Q_B-1} \frac{1}{\sqrt{2\pi}} e^{\frac{-(x - 32)^2}{16}}\right]^{24} \quad (4.20)$$

The distribution function of Eq. (4.20) is shown in Figure 4.5.

If a threshold is applied to the points that pass the peak, the probability of false alarm can again be computed by:

$$P_{FA_a} = \sum_{x=Q_B}^{64} f_T(X) \quad (4.21)$$

This probability of false alarm, Eq. (4.21), is illustrated by Figure 4.6.

If the mask is exactly aligned with an axon, the argument on the probability of passing the peak test is slightly different that the argument presented above. Assuming that the mask is positioned over the axon, the probability distribution for the correlation is given by Eq. (4.13), page 15, and is,

$$f_{Q_m} = \frac{64!}{(64 - k)!k!} p^k (1 - p)^{(n - k)} \quad (4.22)$$

However, the probability distribution for the twenty-four surrounding points is given by Eq. (4.16) since for those points the mask is not exactly aligned with an axon. Without restating the argument of the probability that the surrounding points are less than some value (Eq. 4.18), the probability that a point where the mask and an axon are exactly aligned passes the peak test is given by:

$$P_p = P_a \cdot f_{Q_m}(k) = f_p(Q_m) \quad (4.23)$$

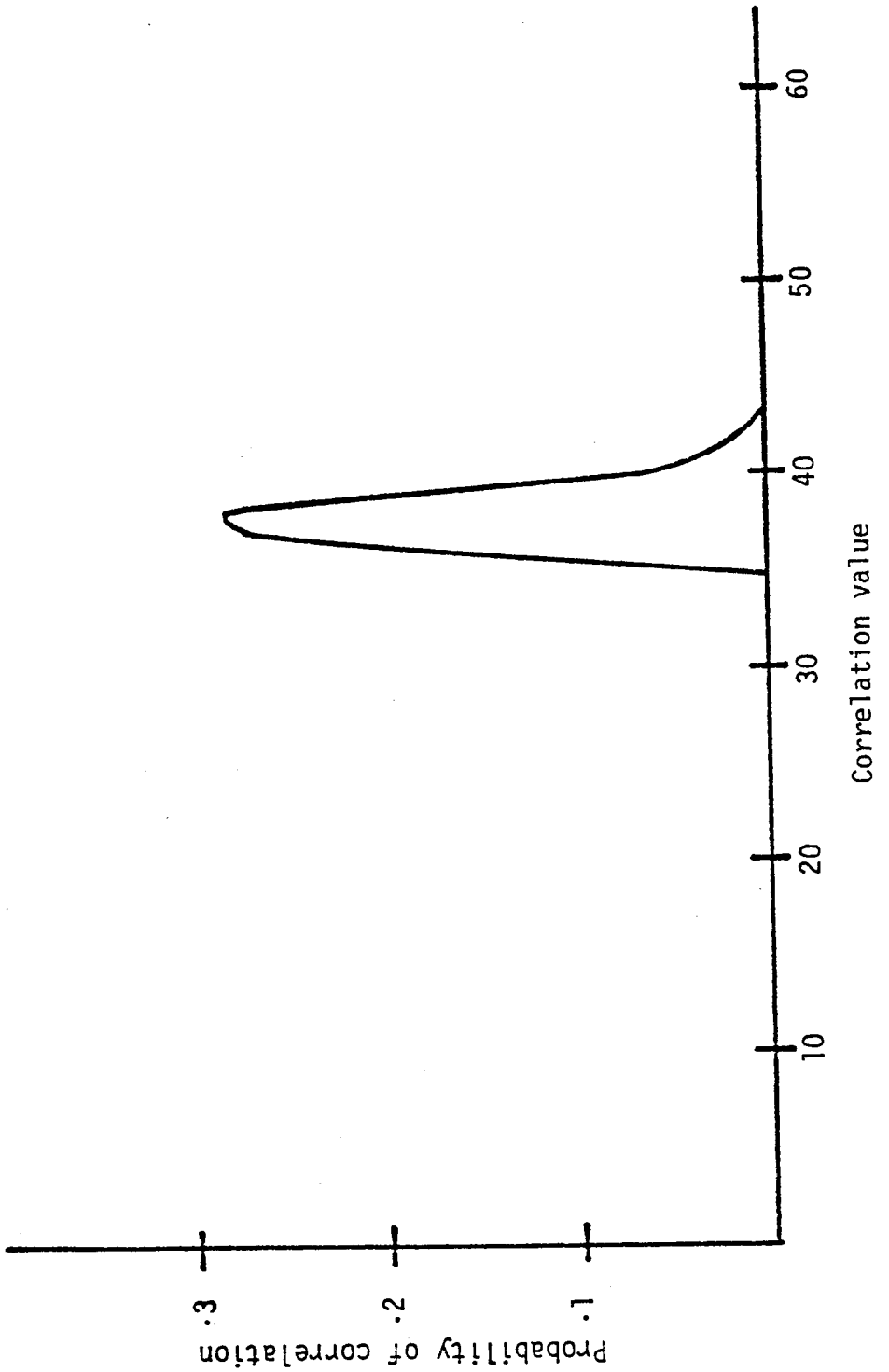


Figure 4.5. Probability density function for correlation values after peak detection.

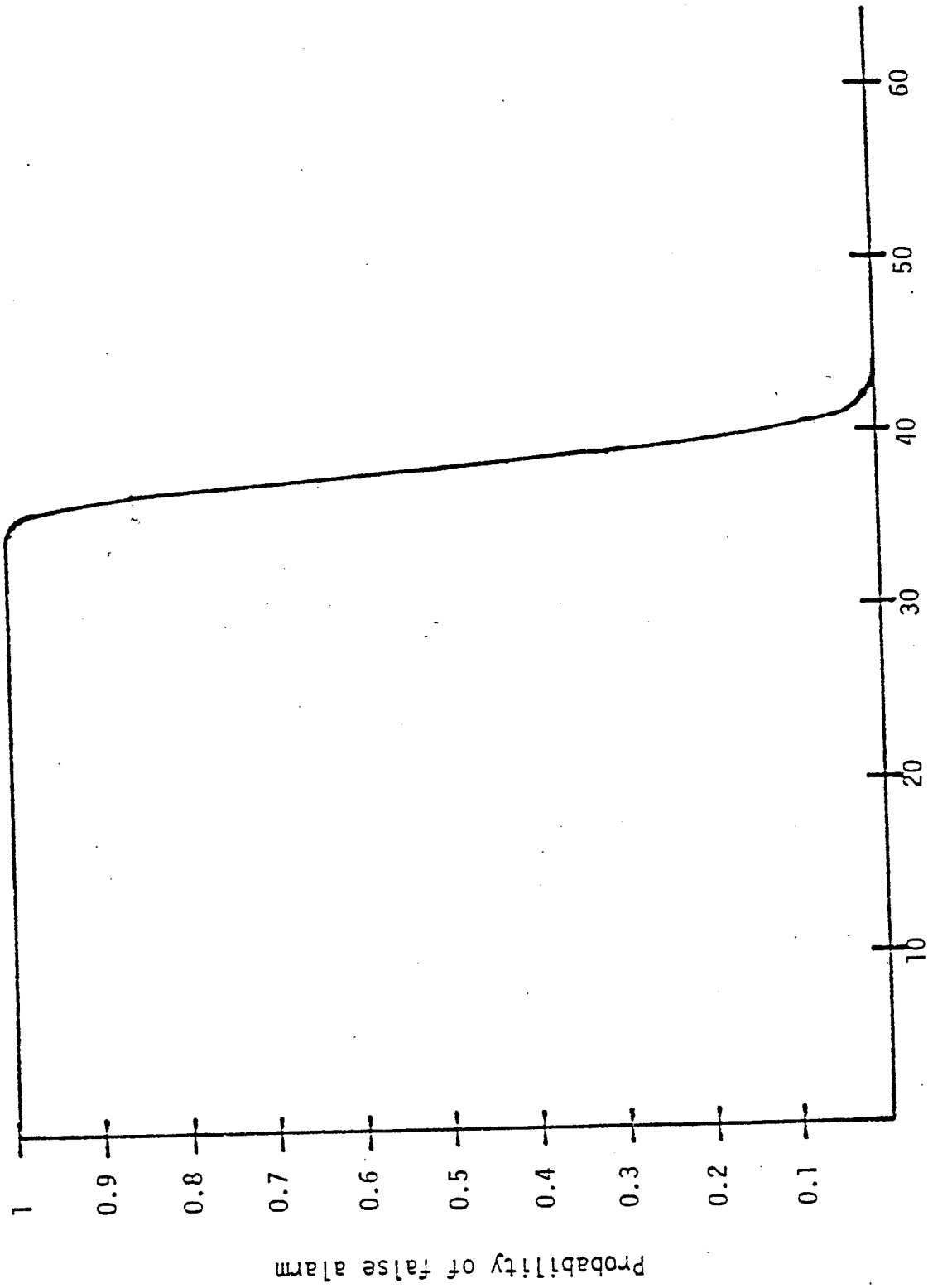


Figure 4.6. Probability of false alarm after peak detection.

Substitution of Eq. (4.22) and Eq. (4.18) into Eq. (4.23) produces:

$$f_p(k) = \left[\frac{64!}{(64 - k)!} p^k (1 - p)^k\right] \left[\sum_{x=0}^{k-1} \frac{1}{\sqrt{2\pi}} e^{\frac{-(x - 32)^2}{16}}\right]^{24} \quad (4.24)$$

In the same manner, the probability of success can be found by finding the probability that Q_m, the correlation value that passed the peak test, lies above a given threshold or from Eq. (4.24).

$$P_{S_a}(Q_m) = \sum_{x=Q_m}^{64} f_p(x) \quad (4.25)$$

Using the probability of false alarm from Eq. (4.18) and the probability of success from Eq. (4.25) above, the R.O.C. curves shown in Figure 4.7 were generated. Note that the curve for a probability of 0.5 that a point will match is closer to the ideal than the corresponding curve in Figure 4.4. This suggests that the peak test actually enhances the performance of the system.

Since the assumption was made that a background point had a probability of 0.5 of being one, then it is logical to assume that in an axon a point has a 0.5 probability of matching the corresponding pixel in the mask. Considering these assumptions, the normalized probability density functions of the correlation in the background and the correlation on an axon are plotted in Figure 4.8. The curve to the left is the density function for the correlation on an axon. The best place to separate the two curves is the threshold where they cross. If this separation is made, however, there is approximately a 0.5 probability of success and a 0.5 probability false detection. The 0.5 probability of success means that for a given number of axons only one-half would actually be counted.

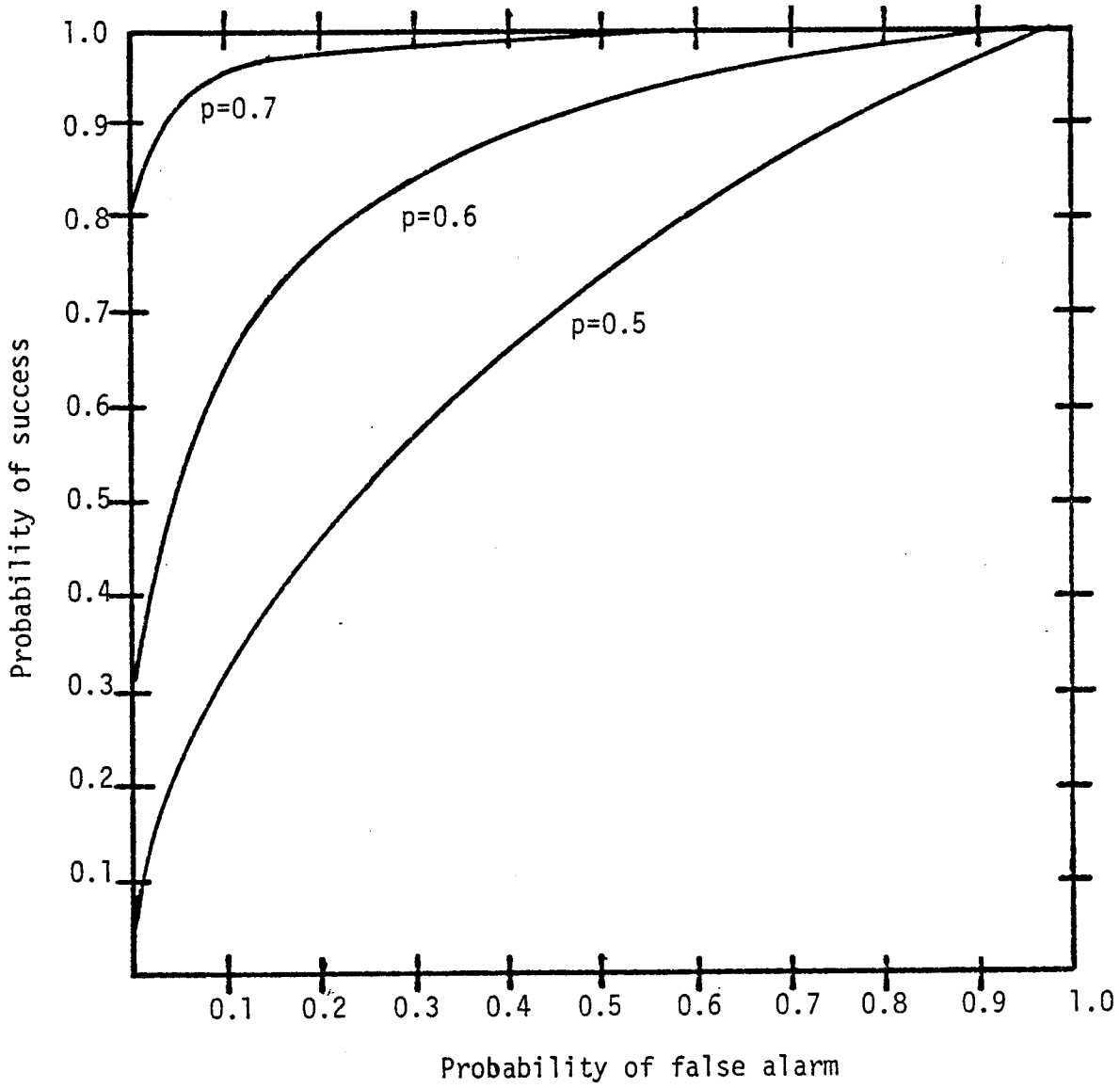


Figure 4.7. Receiver operating characteristic curves for detector after peak detection.

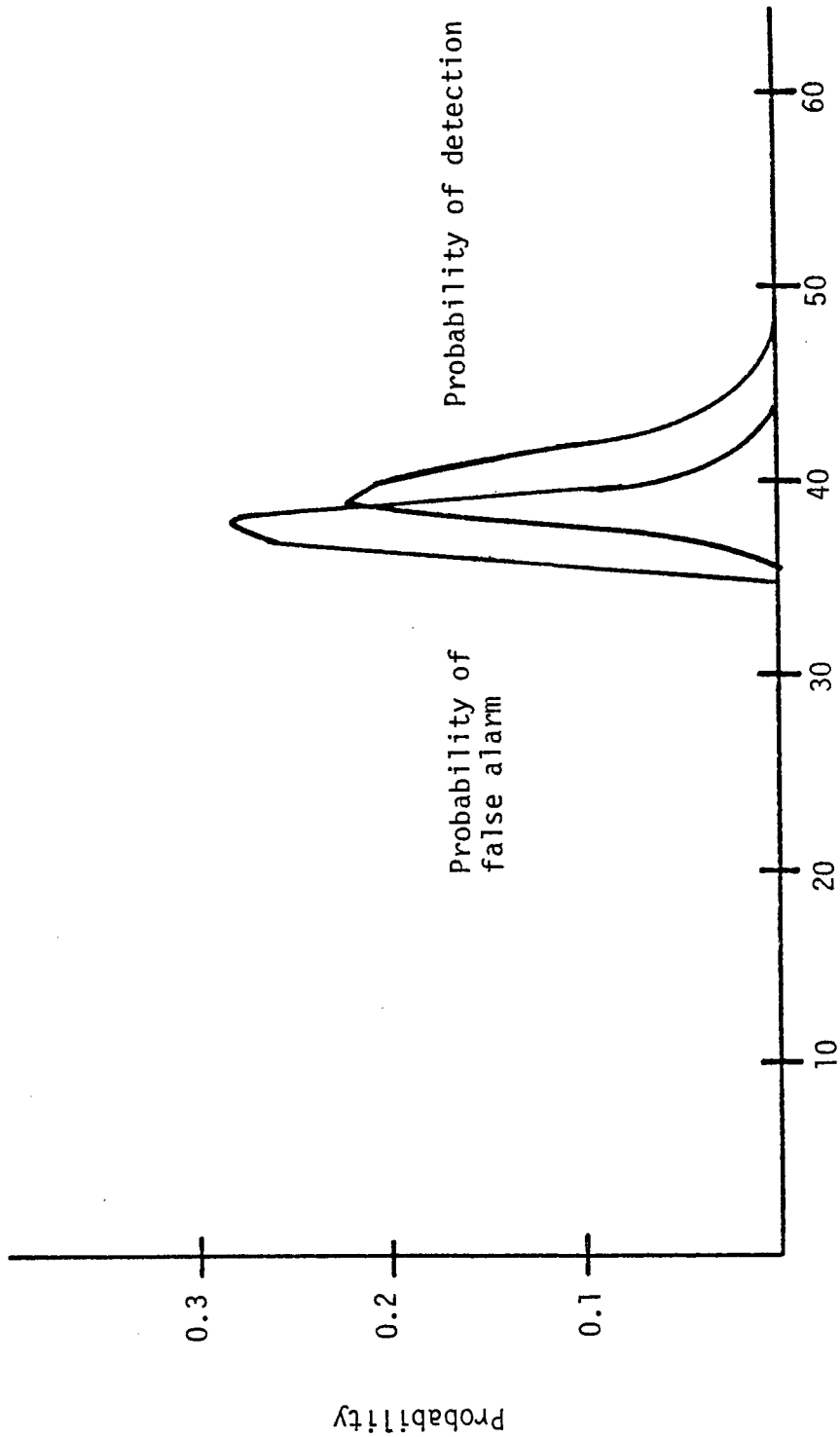


Figure 4.8. Probability density function of correlation surface.

However, there is also a 50 percent chance of false detection. If so, then thresholding at this point would produce a count that is approximately the number of axons in the image. In fact, it was determined experimentally that the threshold should be at the peak of histogram and of the correlation values that passed the peak test.

CHAPTER 5

EXPERIMENTAL RESULTS

Counting nerve cells is done using a program written in Fortran, with several subroutines written in Macro-11. A program flow chart is shown in Figure 5.1. The main program which is included in the Appendix mainly handles the opening of files and other similar functions. Actual implementation of the edge detection and correlation is done in Macro-11 to decrease the time required for the program to execute. Data is handled using subroutine WINDOW. This subroutine is used to keep a section of the image in memory, and to update an offset list. Each subroutine, EDGE, MASK and FLAG, that uses a window type operation is called from WINDOW.

The time required for the program to run is between five to ten minutes per frame depending on the axons in the frame. A typical nerve is shown in Figure 5.2. This nerve consists of twenty-two frames, therefore, the program must loop twenty-two times. Each frame was photographed from the original sample using a microscope with a transport. Since the field of view of the camera is slightly larger than the movement of the transport, a certain amount of overlap resulted. This overlap was considered in the program by simply deleting those lines and columns of the picture that overlapped. In Figure 5.3, a typical frame from a nerve is shown.

Because of the size of the correlation window, eight by eight, the program execution time is very long. To decrease this time, a method of flagging the places in the image where the correlation is to be computed was developed. Essentially this process consists of taking the binary

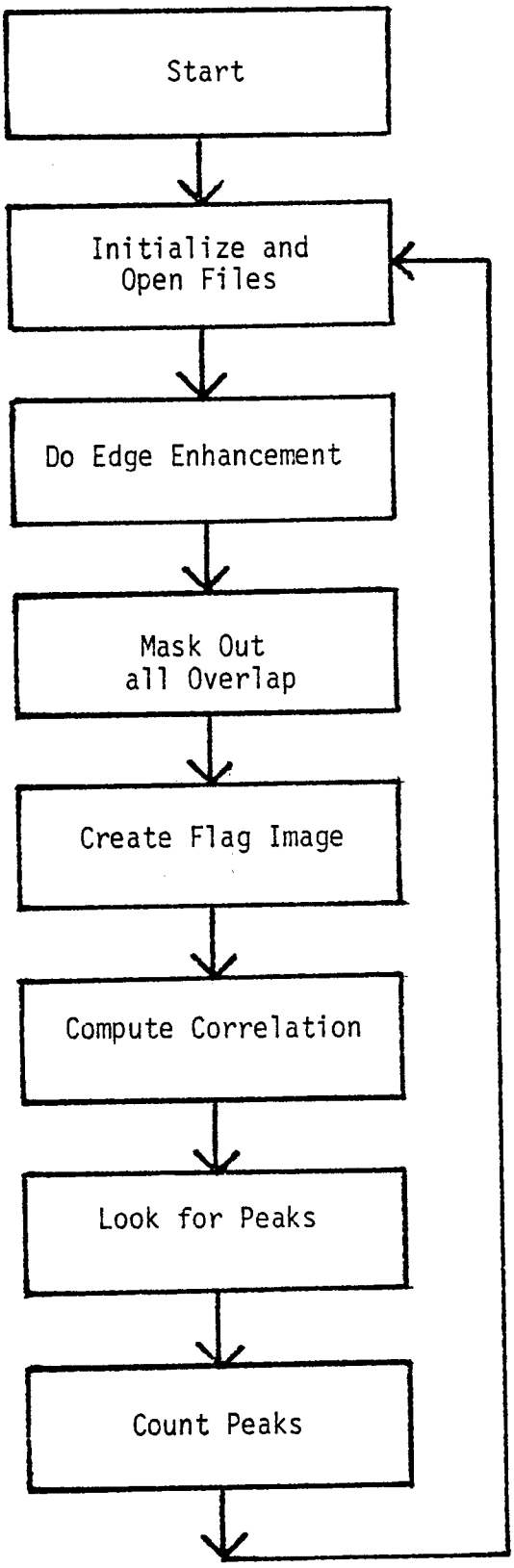


Figure 5.1. Program flow chart.

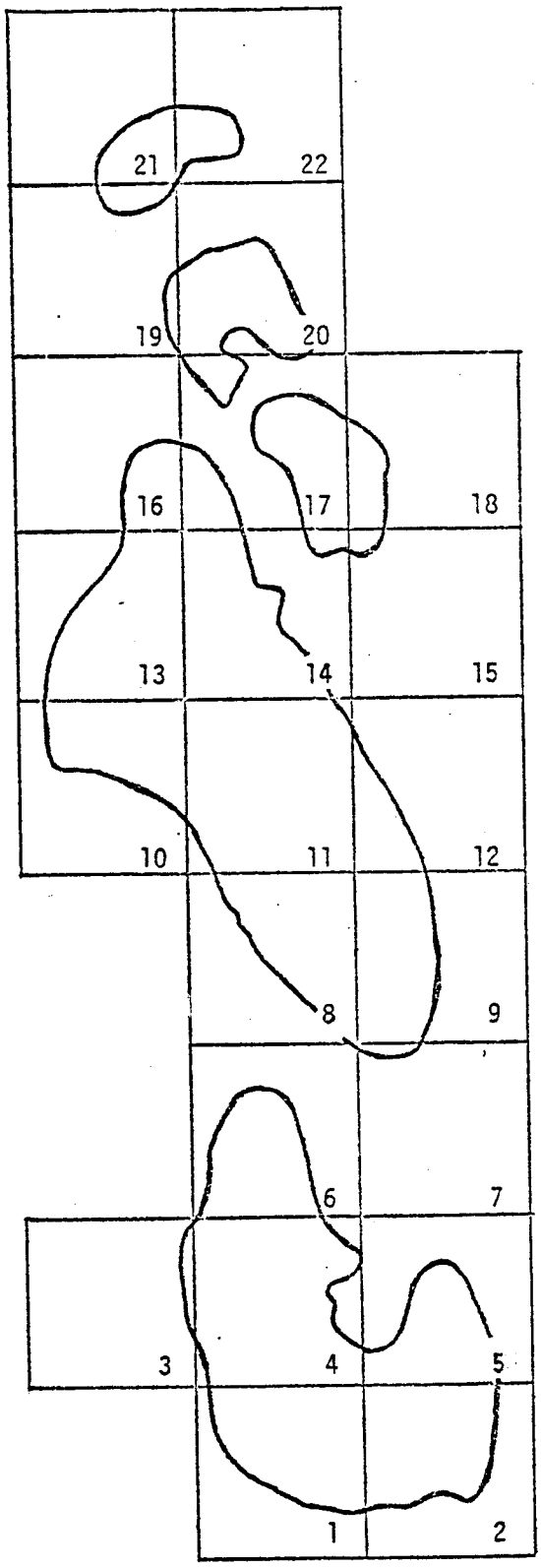


Figure 5.2. A typical nerve showing individual frames.

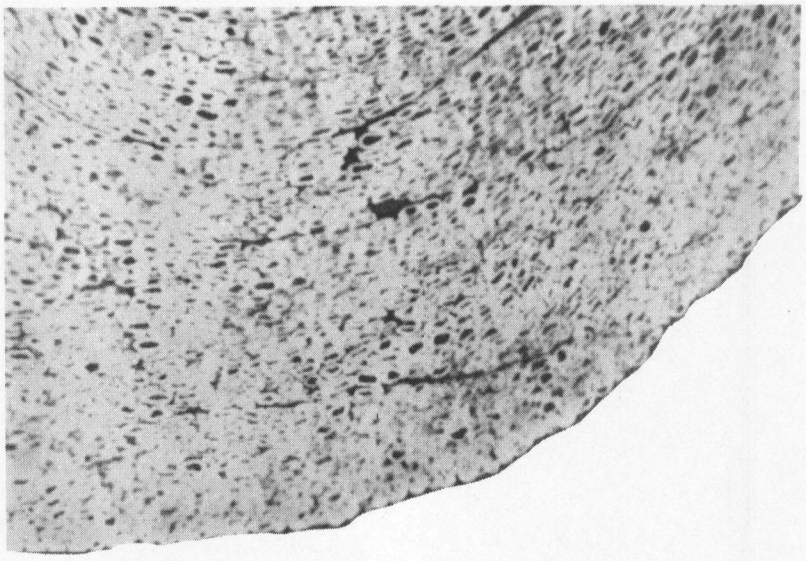


Figure 5.3. A typical frame as digitized.

image before the correlation step and expanding each pixel that is one into a seven by seven square, then writing the resultant image to a new file. When the correlation program executes it examines this image and only computes the correlation where a one appears in the flag image. This technique decreases the execution time of the program by 50 to 70 percent depending on the original image.

Table 5.1 through 5.4 show the results of counting four nerves and comparing the counts with those counts created manually. The absolute error was computed on each frame and included also. The computation was done using:

$$\varepsilon = \left| \frac{\text{Manual Count-Machine Count}}{\text{Manual Count}} \right| \times 100\%$$

where the manual count was taken to be a correct or true value. The mean and variance of the error for all four pictures was computed using:

$$\mu = \frac{1}{N} \Sigma \text{ error}$$

for the mean of the error and similarly:

$$\sigma = \sqrt{\frac{1}{N} \Sigma \text{ error}^2 - \mu^2}$$

for the variance. The mean of the error is $\mu = 57.8$, and the variance is $\sigma = 411.5$.

As another basis for comparison, a least squares fit of a straight line was made to the data. This line is plotted in Figure 5.4. The dashed line is the ideal line desired.

The other nerve pairs, proximal and distal, were counted, but there are no manual counts to be used as a comparison.

TABLE 5.1

Manual and Machine Counts for Nerve 124-D-25.

NERVE IDENTIFICATION 124-P			
No.	Manual	Machine	Error
1	490	525	13
2	486	551	13
3	21	155	37
4	887	784	12
5	984	1015	3
6	694	916	32
7	712	953	34
8	0	36	-
9	884	868	2
10	2003	2236	12
11	1817	2020	14
12	389	578	49
13	42	320	66
14	1414	1368	3
15	397	404	2
16	1798	1309	27
17	908	1053	16
18	431	481	12
19	1989	1440	28
20	249	277	11
Total	16095	17339	7.80%

TABLE 5.2

Manual and Machine Counts for Nerve 23-NN3-P

NERVE IDENTIFICATION			
23-NN3-P			
No.	Manual	Machine	Error
1	1065	593	44
2	1253	757	39
3	0	0	3
4	1738	1177	32
5	1519	1359	10
6	68	5	93
7	626	760	21
8	0	33	-
9	0	80	-
10	437	284	35
11	807	559	31
12	918	773	16
13	2184	2184	0
14	2001	2792	5
15	582	446	23
16	1351	1053	22
17	1132	1019	10
18	51	139	172
19	236	375	59
20	600	848	41
21	13	18	38
22	394	358	91
23	112	207	85
Total	17087	15819	7.4

TABLE 5.3

Manual and Machine Counts for Nerve NN3-D

NERVE IDENTIFICATION			
NN3-D			
No.	Manual	Machine	Error
1	1246	1260	1
2	1101	1042	5
3	295	172	42
4	1687	1201	29
5	1002	912	9
6	26	46	77
7	920	849	47
8	6	130	7
9	911	1049	15
10	570	719	26
11	1142	1218	7
12	1958	2067	6
13	230	253	10
14	1787	2322	30
15	957	1241	30
16	0	67	-
17	696	607	13
18	846	1017	20
19	28	13	54
20	292	541	86
21	371	603	62
22	689	542	31
23	0	65	-
Total	16654	17871	7.3%

TABLE 5.4

Manual and Machine Counts for Nerve 124-D-25.

NERVE IDENTIFICATION 124-D-25			
No.	Manual	Machine	Error
1	1205	1251	4
2	294	294	0
3	0	9	-
4	378	582	54
5	2193	1228	44
6	1272	1260	.4
7	2892	1591	45
8	542	260	51
9	0	25	-
10	380	808	113
11	887	1032	16
12	954	1150	21
13	378	319	16
14	868	610	29
15	705	1073	52
16	1542	1181	23
17	645	579	10
18	15	17	13
19	0	502	-
20	200	391	95
21	1393	988	29
22	1497	1009	33
23	4223	1664	61
24	600	474	21
25	1852	824	55
Total	24924	20718	16.8%

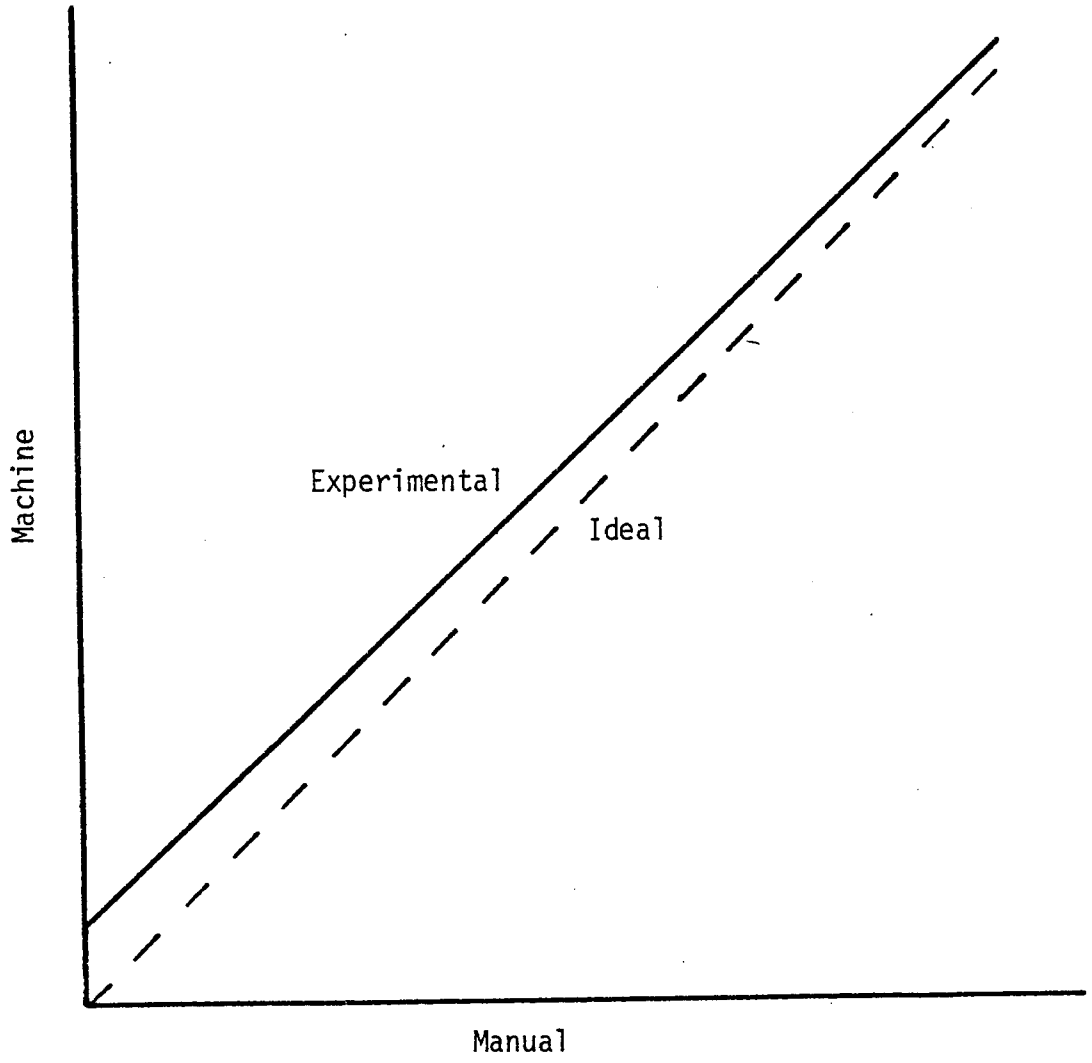


Figure 5.4. Least squares fit of Machine vs. Manual for NN3 Proximal.

The system, as implemented, seems to produce fairly accurate counts for the amount of computation done. For more accurate results, more computation would be required, thus more time. A time of about three hours for a machine count is a respectable decrease from a time of approximately twenty-four hours for a manual count. Another advantage of the computer method of counting the nerve axons is freeing someone from the tiresome task of manually counting the axons. This system demonstrates that an automatic particle count from images is possible.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Kasvand, T. "Pattern Recognition Applied to the Counting of Nerve Fiber Cross-Sections and Water Droplets." International Conference on Methodologies of Pattern Recognition, January, 1968.

[2] Potts, M., Hodges, D., Sherman, C., Fritz, K., Levy, N., Marshall, Y., "Morphology of the Primate Optic Nerve," Investigative Ophothalmogy, vol. 11, no. 10, pp. 980-98 , December, 1972.

[3] Gonzalez, R. C., Wintz, P. A.; Digital Image Processing, Addison-Wesley, in preparation.

[4] Rosenfeld, A., Kak, A. Digital Picture Processing, New York: Academic Press, 1976, p. 297.

[5] Ibid, p. 185.

[6] Vanderburg, G., Rosenfeld, A., "Two State Template Matching IEEE Transactions on Computers," vol. C-26, pp. 384-392, April, 1977.

[7] Thomas, J., Statistical Communication Theory, New York: Wiley, 1969.

[8] Casey, M., "Picture Display Program," Image Processing Lab, 1976.

[9] Fittes, B., "Picture Sample Program," Image Processing Lab, 1975.

[10] Rutland, D., Computer Eye Handbook, Spatial Data Systems, 1974.

[11] Hall, E., Varsi, G., Thompson, W., Gauldin, R., "Computer Measurement of Particle Sizes in Electron Microscope Images," IEEE Transactions on Systems, Man, and Cybernetics, pp. 138-145, February, 1976.

[12] Bernroider, G., "Recognition and Classification of Structure by Means of Stereological Methods in Neurobiology," Fourth International Congress for Stereology, 1975.

APPENDIX

```
      PROGRAM COUNT
C
C
C     THIS PROGRAM IS AN ATTEMPT AT A GENERALIZED METHIOD
C     OF PARTICLE COUNTING.  THE THEORY USED IS AN EDGE
C     CORRELATION THAT LOOKS FOR AN 'AVERAGE PARTICLE'.
C     BY COUNTING THE PEAKS IN THE CORRELATION SURFACE
C     IT IS POSSIBLE TO OBTAIN AN ESTIMATE OF THE NUMBER OF
C     PARTICLES IN THE IMAGE.  CHANGING THE MASK USED IN
C     THE CORRELATION ALLOWS THE DETECTION OF DIFFERENT
C     TYPES OF PARTICLES.  INITIALLY THIS PROGRAM WAS
C     DEVELOPED TO COUNT THE AXONS IN NERVE BUNDLES SO THE
C     MASK DEFINED IS A GENERALIZED AXON.
C
C
C     WRITTEN BY MICHAEL CASEY
C
C
C
C     DEFINE THE ARRAYS.
C
      LOGICAL*1 LPIC(8192),LP(8192),TSTRW(512)
      INTEGER*2 FILSPC(39),IND(32),DEFEXT(4),MSK(64)
      INTEGER*2 CROP(4,5),SW(4,1),EMSK(9)
C
C     ALL ARRAYS ARE COMMON TO DECREASE SIZE.
C
      COMMON /ITHRH/ITHRH/NUM/NUM/EMSK/EMSK
      COMMON /LPIC/LPIC/LP/LP/TSTRW/TSTRW/MSK/MSK
C     SUBROUTINE NAMES ARE DECLARED EXTERNAL SO THEY CAN
C     BE CALLED FROM A COMMON ROUTINE.
C
      EXTERNAL EDGE,FLAG,MASK,THIN,PEAK
      DATA DEFEXT/4*3RPIC/SW(1,1)/'A'/
C
C     DEFINE THE CROPPING BOUNDRIES.
C
      DATA CROP/40,500,120,440,40,500,168,440,109,500,120
     1,440,109,500,168,440,3,509,3,509/
      IERCNT=0
C
C     INCREASE THE SIZE OF THE I/O QUE.
C
      IF(IQSET(6).NE.0) STOP 'QUE SIZE'
C
C     PRINT THE INTRODUCTION.
C
      CALL PRINT('PARTICLE COUNTING PROGRAM')
      CALL PRINT(' ')
     1 CALL PRINT('ENTER COMMAND STRING: NEXT FILE<PRESENT
      1 FILE/S')
C
C     ACCEPT A COMMAND STRING FROM THE KEYBOARD.  THEN
```

```
C       PARSE THAT COMMAND STRING.
C
        CALL ICSI(FILSPC,DEFEXT,,SW,1)
C
C       FETCH THE DEVICE HANDLERS.
C
        IF(IFETCH(FILSPC(16)).NE.0) STOP 'BAD FETCH'
C
C       LOOKUP THE INPUT FILE.
C
        IF(LOOKUP(3,FILSPC(16)).EQ.520) GOTO 2
        IF(IERCNT.EQ.10) STOP 'TOO MANY NON-EXISTENT FILES'
        CALL PRINT('FILE NOT FOUND')
        GOTO 1
C
C       SEND THE NEW FILE NAME TO THE FORGROUND JOB.
C
    2   CALL ISDAT(FILSPC,39)
C
C       ASK FOR THE CROPPING NUMBER.
C
        CALL PRINT('ENTER THE CROPING NUMBER')
        ACCEPT 9,ICASE
C
C       INITALIZE THE FILE DESCRIPTORS FOR THE TEMPORARY
C       FILES.
C
        FILSPC(1)=FILSPC(16)
        FILSPC(2)="76725
        FILSPC(3)="64260
        FILSPC(4)="62553
        FILSPC(6)=FILSPC(16)
        FILSPC(7)=FILSPC(2)
        FILSPC(8)="64330
        FILSPC(9)=FILSPC(4)
C
C       CREATE TWO TEMPORARY FILES.
C
        IF(IENTER(1,FILSPC(1),IENTER(2,FILSPC(6),520)).NE.
        1520)STOP 'NO ROOM'
        IF(SW(2,1).EQ.2.OR.SW(2,1).EQ.0) GOTO 3
        CALL PRINT('NO SWITCH ARGUEMENT?')
        GOTO 1
    3   IF(SW(4,1).EQ.1.OR.SW(4,1).EQ.3) CALL DEFEMS
        IF(SW(4,1).EQ.2.OR.SW(4,1).EQ.3) CALL DEFMSK
        NUM=1
C
C       CONVRT SIMPLY REASSIGNS THE MASK VALUES TO A
C       DIFFERENT FORMAT.
C
        CALL CONVRT
C
C       DO THE EDGE ENHANCEMENT.
```

```
C
      CALL WINDOW(EDGE,3,3,1,LPIC,LP)
C
C     THRESHOLD THE PICTURE.
C
      DO 4 I=8,479,32
      IF(IREADW(8192,LPIC,I,1).NE.8192) STOP 'RD 1'
      CALL THRESH(70)
      IF(IWRITW(8192,LPIC,I,1).NE.8192) STOP 'WT 1'
    4 CONTINUE
C
C     CROP OUT ALL THAT OVERLAPS.
C
      DO 8 I=8,479,32
      II=I-8
      IF(IREADW(8192,LPIC,I,1).NE.8192) STOP 'RD 2'
      DO 7 J=1,32
      KK=(J-1)*512
      JJ=J+II
      IF(JJ.GE.CROP(3,ICASE).AND.JJ.LE.CROP(4,ICASE))
      1 GOTO 5
      CALL ROWCLR(LPIC(KK+1),512)
      GOTO 7
    5 DO 6 K=1,512
      IF(K.LE.CROP(1,ICASE).OR.K.GE.CROP(2,ICASE))
      1 LPIC(KK+K)="200
    6 CONTINUE
    7 CONTINUE
      IF(IWRITW(8192,LPIC,I,2).NE.8192) STOP 'WT 2'
    8 CONTINUE
C
C     CREATE A FILE OF THE PLACES TO PREFORM THE
C     CORRELATION.
C
      CALL WINDOW(FLAG,7,2,2,LPIC,LP)
C
C     PREFORM THE CORRELATION.
C
      CALL WINDOW(MASK,8,1,1,LPIC,LP)
C
C     LOOK FOR THE PEAKS.
C
      CALL WINDOW(PEAK,5,1,1,LPIC,LP)
      ITHRH="201
C
C     COUNT THE PEAKS.
C
      CALL PCOUNT
C
C     PURGE THE TEMPORARY FILES.
C
      CALL PURGE(1)
      CALL PURGE(2)
      CALL PURGE(3)
```

```
      GOTO 1
      STOP 'END OF COUNTING'
    9 FORMAT(I1)
      END
      BLOCK DATA
C
C     THIS BLOCK DATA SEGMENT IS USED TO DEFINE
C     THE EDGE MASK AND THE CORRELATION MASK.
C
      INTEGER*2 MSK(64),EMSK(9)
      COMMON /MSK/MSK/EMSK/EMSK
      DATA MSK/8*0,0,0,0,1,1,0,0,0,0,0,1,1,1,1,0,0,0,1
     1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,0,1,1,1,1,0,0,0,0
     1,0,1,1,0,0,0,8*0/
      DATA EMSK/-1,-1,-1,-1,8,-1,-1,-1,-1/
      END
```

```
      SUBROUTINE WINDOW(SRTN,JNUM,IC1,IC2,LPIC,LP)
C
C
C     THIS SUBROUTINE IS USED TO CONTROL THE PICTURE
C     DATA IN SUCH A WAY AS TO ALLOW A WINDOW TO BE
C     PASSED OVER THE PICTURE.  THE SECRET IS AN INDEXING
C     SCHEME THAT USES AN ARRAY.  THE FIRST N ELEMENTS
C     ALWAYS POINT TO THE LINES OF THE PICTURE THAT
C     ARE UNDER CONSIDERATION.  AS THE WINDOW MOVES DOWN
C     THROUGH THE DATA, OLD DATA IS WRITTEN TO THE DISK,
C     AND NEW IS READ IN.
C
C             SRTN    SUBROUTINE NAME
C             JNUM    ROW DIMENSION OF THE WINDOW
C             IC1     INPUT CHANNEL
C             IC2     OUTPUT CHANNEL
C             LPIC    INPUT ARRAY
C             LP      OUTPUT ARRAY
C
C
C     ARRAY DEFINATIONS.
C
      LOGICAL*1 LPIC(8192),LP(8192)
      INTEGER*2 IND(16)
C
C     MAKE THESE COMMON SO THE SUBROUTINES HAVE ACCESS
C     ALSO.
C
      COMMON /IND/IND/ILIN/ILIN
C
C     READ AND WRITE THE HEADER.
C
      IF(IWRITW(IREADW(2048,LPIC,0,IC1),LPIC,0,IC2)
     1.NE.2048) STOP 'WT ERR SW1'
C
C     CLEAR THE WORK SPACE
C
      CALL ROWCLR(LP,8192)
      J=0
C
C     SET UP THE INDEX ARRAY
C
      DO 1 I=1,16
      IND(I)=J
    1 J=J+512
C
C     READ IN FIRST PORTION OF THE PICTURE.
C
      IF(IREADW(2048,LPIC,8,IC1).NE.2048) STOP 'RD ERR SW1'
      IBLK=24
      ILIN=8
C
C     CALL WHATEVER SUBROUTINE IS SPECIFIED.
```

```
C
    2   CALL SRTN
        ILIN=ILIN+1
C
C       ROTATE THE INDEXES.
C
        DO 3 I=1,16
        IF(IND(I).EQ.7680) IND(I)=-512
    3   IND(I)=IND(I)+512
        ITEM=IND(JNUM)
C
C       CHECK TO SEE IF THERE IS ENOUGH DATA IN MEMORY.
C
        IF(ITEM.NE.0.AND.ITEM.NE.4096) GOTO 2
        IF(IBLK.EQ.16) GOTO 4
C
C       WRITE OUT OLD DATA.
C
        IF(IWRITW(2048,LP(ITEM+1),IBLK-16,IC2).NE.2048)
        1 STOP'WT ERR SW2'
        CALL ROWCLR(LP(ITEM+1),4096)
    4   IF(IBLK.GE.520) GOTO 5
C
C       READ IN SOME NEW DATA.
C
        IF(IREADW(2048,LPIC(ITEM+1),IBLK,IC1).NE.2048)
        1 STOP'RD ERR SW2'
    5   IF(IBLK.GT.520) GOTO 6
        IBLK=IBLK+8
        GOTO 2
    6   RETURN
        END
```

```
      SUBROUTINE DEFEMS
C
C
C     SUBROUTINE DEFEMS IS USED TO ALLOW AN EDGE MASK TO
C     BE ENTERED BY INCLUDING AN 'A' SWITCH WITH EITHER A
C     ':1' OR ':3' ARGUEMENT.
C
C
      INTEGER*2 EMSK(9)
      COMMON /EMSK/EMSK
      CALL PRINT('INPUT EDGE MASK')
      ACCEPT 10,((EMSK(J),J=I,I+2),I=1,7,3)
10    FORMAT(3I4)
      RETURN
      END
      SUBROUTINE DEFMSK
C
C
C     SUBROUTINE DEFMSK IS USED TO SPECIFY A NEW MASK FOR
C     THE CORRELATION.  DEFMSK IS CALLED WHEN EITHER A '/A:2'
C     OR A '/A:3' SWITCH IS USED IN THE COMMAND STRING.
C
C
      INTEGER*2 MSK(64)
      COMMON /MSK/MSK
      CALL PRINT('INPUT BINARY MASK FOR CORRELATION')
      ACCEPT 10,((MSK(J),J=I,I+7),I=1,57,8)
10    FORMAT(8I2)
      STOP
      END
      SUBROUTINE CONVRT
C
C
C     CONVRT IS USED TO CHANGE THE BINARY FORMAT OF THE
C     CORRELATION MASK FROM 0 AND 1 TO -128 AND 127
C     RESPECTIVELY TO ALLOW EASIER SPECIFICATION OF
C     THE MASK.
C
C
      INTEGER*2 MSK(64)
      COMMON /MSK/MSK
      DO 1 I=1,64
      IF(MSK(I).EQ.1) MSK(I)=127
      IF(MSK(I).EQ.0) MSK(I)=-128
1     CONTINUE
      RETURN
      END
```

```
        .NLIST  TTM
        .TITLE  EDGE MASK SUBROUTINE
        .SBTTL  EDGE SUBROUTINE


;  THIS SUBROUTINE CONVOLVES A 3X3 MASK WITH EACH 3X3
;  NEIGHBORHOOD OF AN IMAGE.  THE MASK IS DEFINED
;  IN THE CALLING PROGRAM.  EDGE IS CALLED FROM FORTRAN
;  USING THE FOLLOWING STATEMENT:
;       CALL EDGE()
;  THE SUBROUTINE NEEDS THE FOLLOWING ARRAY
;  NAMES DEFINED USING A LABELED COMMON STATEMENT:
;       LPIC -  THE INPUT ARRAY
;       LP --   THE OUTPUT ARRAY
;       IND -   ARRAY THAT CONTAINS THE ROW OFFSETS
;       EMSK -- THE ARRAY THAT CONTAINS THE MASK
;       NUM --  INTEGER USED TO NORMALIZE THE RESULT

        .MCALL  ..V2..,.REGDEF
        ..V2..  ; VERSION 2
        .REGDEF ; DEFINE THE REGISTERS

;  DEFINE THE GLOBAL SYMBOLS
        .GLOBL LPIC,LP,IND,EMSK,NUM,EDGE

;  LOAD POINT
EDGE:   MOV     #511.,CNT       ; NO. OF POSITIONS PER ROW
E1:     CLR     R0              ; REG IS USED AS ACCUMULATOR
        MOV     #3,R2           ; NO. OF ROWS IN THE MASK
        MOV     #LPIC,R3        ; ADD OF THE INPUT ARRAY
        ADD     CNT,R3          ; OFFSET FOR THE COLUMN
        DEC     R3              ; CENTER THE MASK
        MOV     R3,(PC)+        ; STORE FOR USE LATER
T1:     0
        MOV     #EMSK,R5        ; GET ADDRESS OF THE MASK
        MOV     #IND,R4         ; GET THE ADDRESS OF OFFSETS
E2:     MOV     T1,R3           ; START WITH PREVIOUS OFFSET
        ADD     (R4)+,R3        ; ADD THE ROW OFFSET
        MOVB    (R3)+,R1        ; GET A PIXEL
        ADD     #200,R1         ; RESCALE
        MUL     (R5)+,R1        ; MULTIPLY BY MASK ELEMENT
        ADD     R1,R0           ; ADD TO ACCUMULATOR
        MOVB    (R3)+,R1        ; REPEAT THE LAST 4 INSTS
        ADD     #200,R1         ; TWICE MORE TO TAKE
        MUL     (R5)+,R1        ; CARE OF ONE WHOLE ROW.
        ADD     R1,R0
        MOVB    (R3)+,R1
        ADD     #200,R1
        MUL     (R5)+,R1
        ADD     R1,R0
        SOB     R2,E2           ; COUNT THE ROWS IN THE MASK
        MOV     R0,R1           ; SET UP FOR DIVISION
        CLR     R0
```

```
        DIV     NUM,R0          ; DIVIDE TO NORMALIZE
        BPL     .+4
        CLR     R0              ; FOR NOW JUST CLEAR THE NEG
        SUB     #200,R0         ; RESCALE
        MOV     #2,R4           ; OFF SET TO OFF SET
        MOV     IND(R4),R4      ; GET ROW OFFSET
        ADD     #LP,R4          ; ADDRESS OF OUTPUT BUFFER
        ADD     CNT,R4          ; OFFSET TO COLUMN
        MOVB    R0,(R4)         ; STORE THE PIXEL
        DEC     (PC)+           ; DECREMENT THE COLUMN COUNT
CNT:    0
        BNE     E1              ; BRANCH IF NOT DONE
        RTS     PC              ; GO TO YOUR CALLER

        .END    EDGE
```

```
        .NLIST  TTM
        .TITLE THRESHOLD SUBROUTINE
;
;
; THIS SUBROUTINE IS USED TO THRESHOLD AN IMAGE.
; THE ROUTINE OPERATES ON A 16384 BYTE BUFFER
; WHICH IS SUPPLIED BY THE CALLING PROGRAM.  THE
; THRESHOLD VALUE IS SUPPLIED AS AN ARGYEMENT TO THE
; SUBROUTINE.
;
; MACRO DEFINATIONS
        .MCALL  ..V2.,.REGDEF
        ..V2..
        .REGDEF
; GLOBAL DEFINATIONS.
        .GLOBL  THRESH,LPIC

; ENTRY POINT
THRESH: MOV     #LPIC,R0        ; IMAGE ADDRESS
        MOV     @2(R5),R1       ; GET THE THR VALUE
        MOV     #16384.,R3      ; NUMBER OF BYTES
        SUB     #200,R1         ; RESCALE THR VALUE
1$:     MOV     #200,R2         ; NEW PIXEL VALUE
        CMPB    R1,(R0)         ; TEST THE PIXEL
        BGT     .+4             ; BRANCH IF PIXEL TO SMALL
        COMB    R2              ; NEGATE IF NOT
        MOVB    R2,(R0)+        ; STORE WHICH EVER
        SOB     R3,1$
        RTS     PC              ; RETURN

        .END    THRESH
```

```
        .NLIST  TTM
        .TITLE  FLAGING FOR CORRELATION

;  THIS PROGRAM IS DESIGNED TO FLAG ALL THE POSITIONS WHERE
;  THE CORRELATION IS TO BE RUN.  THIS IS DONE BY FLAGGING
;  ALL THE NEIGHBORS OF ANY LOCATION THAT IS ONE.

        .MCALL  ..V2..,.REGDEF
        ..V2..
        .REGDEF

        .GLOBL  FLAG,LPIC,IND,ROWCLR,LP

;  LOAD POINT
FLAG:   MOV     #0,R0           ;  STARTING POINT
        MOV     #511.,R1        ;  ENDING POINT
        SUB     R0,R1           ;  NO.  OF POINTS IN BETWEEN
        MOV     #6,R2
        MOV     IND(R2),R2      ;  OFF SET TO START OF LINE
        ADD     R0,R2           ;  ADD THE STARTING POINT
        TSTB    LPIC(R2)        ;  IS IT TRUE?
        BPL     SET
CONT:   INC     R2
        SOB     R1,..-10        ;  COUNT THEM ALL
        RTS     PC              ;  RETURN

SET:    MOV     #512.,R3        ;  NUMBER OF POINTS IN A LINE
        SUB     R1,R3
        SUB     R0,R3
        SUB     #3,R3
        ADD     #LP,R3          ;  GETS US IN THE BALL PARK
        MOV     R3,(PC)+        ;  WE NEED THIS LATER
TEMP:   0
        MOV     #7,R4           ;  NUMBER OF LINES TO ZERO
        MOV     #IND,R5         ;  GET THE INDEX ARRAY
1$:     MOV     TEMP,R3         ;  GET TEMP
        ADD     (R5)+,R3        ;  COMPUTE ADDRES
        .REPT   7
        MOVB    #177,(R3)+      ;  FLAG THE LOCATION
        .ENDR
        SOB     R4,1$           ;  DO FOR SEVEN LINES
        BR      CONT            ;  SEARCH FOR SOME MORE
```

```
;
;
;       SUBROUTINE ROWCLR IS USED TO CLEAN UP THE BUFFERS
;       AS THE PROGRAM PROGRESSES.  THE ROUTINE IS CALLED
;       USING THE COMMAND 'CALL ROWCLR(BUFFER,NUMBER)'.
;       BUFFER IS THE NAME OF THE ARRAY TO BE CLEANED.
;       NUMBER IS THE NUMBER OF BYTES INVOLVED.
;
;
ROWCLR: MOV     2(R5),R0
        MOV     @4(R5),R1
        MOVB    #200,(R0)+
        SOB     R1,.-4
        RTS     PC

        .END    FLAG
```

```
        .TITLE  CORRELATION SUBROUTINE
        .NLIST  TTM

;  THIS PROGRAM COMPUTES THE NOT-EXCLUSIVE-OR CROSS
;  CORRELATION BETWEEN A BINARY MASK AND A BINARY PICTURE.
;  IT IS CALLED FROM FORTRAN USING THE FOLLOWING
;  STATEMENT:  CALL MASK
;

        .MCALL  ..V2..,.REGDEF,.READW
        ..V2..
        .REGDEF

        .GLOBL  MASK,MSK,LP,IND,LPIC,TSTRW,ILIN

;  MSK,LPIC,LP ARE THE MASK,  THE INPUT ARRAY,  AND THE OUTPUT
;  ARRAY RESPECTIVLY.  THEY ARE DECLARED COMMON
;  IN THE CALLING PROGRAM

;  LOAD POINT

        .NLIST  ME

;  TO SPEED UP THE CORRELATION,  A PICTURE IS READ IN
;  THAT CONTAINS THE LOCATION WHERE THE CORRELATION
;  IS TO BE PREFORMED.
MASK:   .READW  #LIST,#2,#TSTRW,#256.,ILIN
        MOV     #512.,R0            ;  NUMBER OF POINTS IN ROW
        MOV     #6.,R1
        MOV     IND(R1),R1
        ADD     #LP,R1
        MOVB    #200,(R1)+          ;  ZERO THE LOCATION
        SOB     R0,.-4              ;  FOR EVERY LOCATION
        MOV     #498.,R5            ;  NUMBER OF POSITIONS
M1:     CLR     R1                  ;  HOLDS THE COUNT
        TSTB    TSTRW(R5)           ;  DO WE DO THIS ONE?
        BMI     M4                  ;  IF NOT,  BRANCH
        MOV     #8.,R2              ;  NUMBER OF ROWS IN THE MASK
M2:     MOV     #8.,R4              ;  NUMBER OF COLS IN THE MASK
M3:     MOV     R2,R3               ;  GET THE COLUMN NUMBER
        ASL     R3                  ;  WORD OFF-SET
        MOV     IND-2(R3),R3        ;  GET THE OFF-SET TO ROW
        ADD     R5,R3               ;  ADD COLUMN NUMBER
        ADD     R4,R3               ;  ADD MASK COLUMN NUMBER
        MOVB    LPIC-1(R3),R0       ;  GET THE BYTE
        MOV     R2,R3               ;  GET MASK ROW NUMBER
        ASL     R3                  ;  WORD OFF-SET
        MOV     IND1--2(R3),R3      ;  GET THE MASK OFFSET
        ADD     R4,R3               ;  ADD COLUMN NUMBER
        ADD     R4,R3               ;  TWICE FOR WORDS
        TST     -(R3)               ;  MINUS TWO
        MOV     MSK-2(R3),TEMP      ;  GET THE MASK VALUE
        XOR     R0,(PC)+            ;  EXCLUSIVE OR
```

```
TEMP:   0
        BNE     .+6                 ; RESULT IS A 1, BRANCH
        INC     R1                  ; IF ZERO, INCREMENT SUM
        BR      .+4                 ; SHOULD SKIP NEXT INS ALWAYS
        DEC     R1                  ; UNLESS RESULT IS ZERO
        SOB     R4,M3               ; COUNT THE COLUMNS IN MASK
        SOB     R2,M2               ; COUNT THE ROWS IN THE MASK
        MOV     IND+6,R3            ; GET THE OFFSET
        ADD     R5,R3               ; ADD THE COLUMN COUNT
        SUB     #100,R1
        MOVB    R1,LP+6(R3)         ; STORE IN THE OUTPUT ARRAY
M4:     SOB     R5,M1               ; COUNT THE COLUMNS
        RTS     PC                  ; RETURN


IND1:   .WORD   0,16.,32.,48.,64.,80.,96.,112.
LIST:   .BLKW   6

        .END    MASK
```

```
        .NLIST  TTM
        .TITLE  PEAK DETECTION ROUTINE
;
;
; THIS SUBROUTINE IS USED TO DETECT THE PEAKS IN THE
; CORRELATION SURFACE.  THE PROGRAM LOOKS AT A 5X5
; AREA.  IF THE CENTER POINT IS LARGER THAN ALL SURROUNDING
; POINTS, THE CENTER POINT IS THEN A PEAK.  THAT POINT IS
; INCLUDED IN THE OUTPUT PICTURE.
; THE SUBROUTINE IS CALLED USING A 'CALL PEAK' STATEMENT.
; THE FOLLOWING ARRAYS ARE ASSUMED TO BE COMMON:
;       LPIC(8192)      INPUT PICTURE (BYTE ARRAY)
;       LP(8192)        OUTPUT PICTURE (BYTE ARRAY)
;       IND(32)         INTEGER ARRAY USED FOR INDEXING.
;
;
        .MCALL  ..V2..,.REGDEF
        ..V2..
        .REGDEF
;
;
        .GLOBL  PEAK,LPIC,LP,IND
;
;
PEAK:   MOV     #508.,R0        ; NO. OF COLUMNS TO LOOK IN.
        MOV     IND+4,R2        ; INDEX TO LINE OF INTREST.
1$:     MOVB    LPIC+4(R2),R1   ; GET CENTER POINT.
        DECB    LPIC+4(R2)      ; NO CAUSE FALSE ALARM.
        MOV     #508.,R3        ; GET NO. OF COLUMNS TO TRY.
        SUB     R0,R3           ; THIS THEN COMPUTES OFFSET.
        ADD     #LPIC,R3        ; PLUS START ADDRESS.
        MOV     R3,-(SP)        ; SAVE FOR LATER.
        MOV     #5,R4           ; NUMBER OF ROWS TO SEARCH.
        MOV     #IND,R5         ; ADDRESS OF INDEXES.
2$:     MOV     (SP),R3         ; GET THE START ADDRESS.
        ADD     (R5)+,R3        ; ADD THE INDEX.
        .REPT   5
        CMPB    R1,(R3)+        ; TEST THE OTHER POINTS.
        BLE     3$              ; IF THEY FAIL, BRANCH.
        .ENDR
        SOB     R4,2$           ; COUNT THE COLUMNS
        MOVB    R1,LP+4(R2)     ; NOW PUT IN THE VALUE.
3$:     TST     (SP)+           ; BUMP THE STACK
        INCB    LPIC+4(R2)      ; RESTORE CENTER POINT.
        INC     R2              ; NEXT POSITION
        SOB     R0,1$           ; LOOP AGAIN
        RTS     PC              ; RETURN TO FORTRAN
        .END
```

```
      SUBROUTINE PCOUNT

C
C
C     T PROGRAM COUNTS THE NUMBER OF AXONS.  THE INPUT IS
C     IS THE PROCESSED PICTURE THAT CONTAINS THE PEAKS
C     ENCODED WITH DIFFERENT VALUES.  ESSENTIALLY WHAT
C     HAPPENS IS THAT A PROBIBILITY DENSITY FUNCTION IS
C     FORMED.  FROM THIS A MAXIMUM IS FOUND.  THE COUNT IS
C     THE INTEGRAL OF THE PDF FROM THE MAX TO 255.
C
C     SUBROUTINES INCLUDED ARE FOUND IN FILE CALLED EQ.MAC
C

C
C     ARRAY DEFINATIONS
C
      LOGICAL*1 LPIC(8192),LP(8192)
      INTEGER*2 ITRA(256)
      REAL*4 TRA(256)
      COMPLEX*8 CTRA(256)
C
C     THESE ARRAYS ARE USED BY THE SUBROUTINES.
C
      COMMON /LPIC/LPIC/LP/LP
C
C     A ROSE BY ANY OTHER NAME...
C
      EQUIVALENCE (ITRA(1),TRA(1),CTRA(1))
C
C     SET UP THE ARRAY FOR THE SUBROUTINE
C
      DO 1 I=1,256
    1 CTRA(I)=CMPLX(1.,0.)
      DO 2 I=8,488,32
      IF(IREADW(8192,LPIC,I,1).NE.8192) STOP'READ 1'
C
C     CALL THE COUNTING SUBROUTINE.
C
    2 CALL PDF(LPIC,CTRA)
C
C     TAKE OUT THE UNESSESARY.
C
      CALL COMPRS(CTRA)
      AMAX=0.
C
C     LOOK FOR THE MAXIMUM OF THE HISTOGRAM.
C
      DO 3 I=256,2,-1
    3 AMAX=AMAX1(AMAX,TRA(I))
      RTEMP=0.
```

```
C
C       THIS LOOP DOES THE INTEGRATION, AND PRINTS OUT
C       THE HISTOGRAM IF ASKED FOR.
C
        DO 4 I=256,2,-1
        RTEMP=RTEMP+TRA(I)
C
C       THIS STATEMENT IS INCLUDED AT COMPILE TIME IF
C       A '/D' SWITCH IS INCLUDED IN THE COMMAND STRING.
C
D       IF(TRA(I).GT.0) TYPE 5,I-1,RTEMP,TRA(I)
        IF(TRA(I).LT.AMAX) GOTO 4
        TYPE 6,I-1,RTEMP,TRA(I)
    4   IF(I.EQ.M) RM=RTEMP
        TYPE 7
    5   FORMAT(1H ,I3,5X,F10.2,10X,F10.2)
    6   FORMAT(1H ,'THE MAX IS AT ',I3,' THE COUNT IS ',F10.
        12,' THE PEAK VALUE IS ',F10.2)
    7   FORMAT(1H0)
        RETURN
        END
```

```
        .NLIST  TTM
        .TITLE  HISTOGRAM MODIFICATION SUBROUTINES
;
;
; THIS SUBROUTINE PACKAGE IS USED WITH HISTOGRAM
; MODIFICATION PROGRAMS.  THE PACKAGE INCLUDES A ROUTINE TO
; CONSTRUCT A PROBIBALITY DENSITY FUNCTION, AN INTEGRATION
; ROUTINE THAT INTEGRATES THE PDF AND A ROUTINE TO MAP THE
; IMAGE TO ANOTHER SET OF GREY LEVELS.
;
;
        .GLOBL  PDF,CDF,TFORM,COMPRS,JSTK
        .MCALL  ..V2..,.REGDEF
        ..V2..
        .REGDEF

; THE FOLLOWING SUBROUTINE IS USED TO FORM THE PDF OF
; 16K BYTES STORED IN IPL FORMAT.  THE CALLING SEQUENCE
; IS:   CALL PDF(LPIC,CTRA).
;       LPIC IS THE 16384 BYTE IMAGE ARRAY.
;       CTRA IS A 256 NUMBER COMPLEX ARRAY THAT IS INATILIZED
;       USING CTRA(I)=COMPLX(PROB,0.).  PROB IS THE
;       PROBIBALITY OF ONE OCCURRENCE.  (1./16384.  FOR A
;       128X128 IMAGE)
;
PDF:    MOV     2(R5),R0
        MOV     4(R5),R1
        MOV     #16384.,R2
LOOP:   MOVB    (R0)+,R3
        ADD     #200,R3
        BIC     #-400,R3
        ASH     #3,R3
        ADD     R1,R3
        FADD    R3
        SOB     R2,LOOP
        RTS     PC
```

```
;
;
;  SUBROUTINE COMPRS IS USED TO FORM THE PDF FROM
;  THE COMPLEX ARRAY USED BY PDF.  ESSENTIALLY COMPRS
;  FORMS A 256 NUMBER REAL ARRAY THAT IS THE
;  IMAGINARY PART OF OF CTRA.
;THE CALLING STATEMENT IS:
;         CALL COMPRS(CTRA)
;         CTRA IS THE COMPLEX ARRAY, THE REAL ARRAY IS
;         RETURNED IN THE FIRST 256 NUMBERS OF CTRA.
;
;
COMPRS:  MOV      2(R5),R0
         MOV      R0,R1
         CMP      (R1)+,(R1)+
         MOV      #256.,R2
         MOV      (R1)+,(R0)+
         MOV      (R1)+,(R0)+
         CMP      (R1)+,(R1)+
         SOB      R2,.-6
         RTS      PC
```

VITA

Michael Edward Casey was born in Los Angeles, California on November 22, 1952. He moved to Paris, Tennessee in 1955. In Paris, he attended elementary school and in June 1970, was graduated from Henry County High School. The following September he entered the University of Tennessee at Knoxville.

College education was interrupted by the four month tour active duty required when he joined the Army National Guard in 1972. In June 1976, he graduated with a Bachelor of Science degree in Electrical Engineering. The following September after accepting a Research Assistantship at the University of Tennessee, he began study toward a Masters Degree. This degree was awarded in August 1977.

Mr. Casey is married to the former Bonnie Charlene Johnson of South Charleston, West Virginia.