8-1991

A system for leak detection using sequential image processing

Jo Anne Malone

To the Graduate Council:

I am submitting herewith a thesis written by Jo Anne Malone entitled "A system for leak detection using sequential image processing." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

L. Montgomery Smith, Major Professor

We have read this thesis and recommend its acceptance:

Roger Crawford, Bruce Bomar

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a thesis written by Jo Anne Malone, entitled "A System For Leak Detection Using Sequential Image Processing." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

Dr. L. Montgomery Smith,
Major Professor

We have read this thesis
and recommend its acceptance:

Accepted for the Council:

Vice Provost
and Dean of The Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at the University of Tennessee, Knoxville, I agree that the library shall make it available to borrowers under rules of the library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of the source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or in his absence, by the Head of Interlibrary Services when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature Jo Anne Malone

Date 7/91

A SYSTEM FOR LEAK DETECTION USING SEQUENTIAL IMAGE PROCESSING

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Jo Anne Malone

August 1991

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to my advisor and major professor Dr. L. Montgomery Smith. His guidance, suggestions, and advice were vital to the sucess of this thesis. Appreciation is also due Dr. Roger Crawford for his assistance and Dr. Bruce Bomar for serving on my thesis committee. Boyd Stubblefield and Laura Payne are thanked for preparing the photographs in this thesis.

I would also like to thank my husband, Ken, for his encouragement and support during the course of this study. Finally, appreciation is expressed to God for the ability to complete this work.

This work was supported by the Center for Advanced Space Propulsion and NASA under Contract CASP 25-945 and performed at the Center for Lasar Applications, The University of Tennessee Space Institute, Tullahoma, TN.

ABSTRACT

A system for detecting gas leaks is described and tested. The algorithm used by the system detects the occurrence of an abrupt but sustained change in a time sequence of video images. The system consists of a highpass filter cascaded with a moving average filter. The highpass filter removes slowly varying background intensity values and its output is averaged with a set number of previous outputs by the moving average. The absolute value of the average is taken and then compared to a threshold to decide if a step-like change has occurred. It is shown that for a chosen cutoff frequency in the highpass filter, an optimal value for the number of terms in the moving average exists. The cutoff frequency and the corresponding optimal number of terms in the average, as well as, the threshold value determine the smallest step amplitude that can be detected.

The algorithm was tested in a numerical study by implementing it in a FORTRAN program on computer generated input data. The results are shown as the probability of correct detection versus step size and illustrate the effects of noise on the input and the value of the threshold on the performance of the algorithm. Next, the algorithm was tested on a sequence of images of an actual gas leak. A gas leak scene was set up in the laboratory and sequential image data was acquired. The data was processed on a image processor where the algorithm was implemented at every point in an image. The results are shown as binary output images in which

white regions indicate areas where change was detected. The parameters of the algorithm were varied and the results examined. The results presented show the effects of the parameters on the size of the detected region as well as the noise sensitivity of the detector. These results verify that the parameters can be chosen such that this is an effective method for leak detection.

TABLE OF CONTENTS

CHAPTER PAGE

I. INTRODUCTION 1

II. BACKGROUND 5

III. THE STEP DETECTION ALGORITHM 10

3.1 Algorithm Description 10

3.2 Implementation Considerations 16

IV. NUMERICAL STUDY 19

4.1 Procedure 19

4.2 Results 21

4.3 Summary 24

V. EXPERIMENTAL STUDY 26

5.1 Procedure 26

5.2 Results 30

5.3 Summary 51

VI. CONCLUSIONS 53

BIBLIOGRAPHY 56

APPENDIX 59

VITA 62

LIST OF FIGURES

FIGURE PAGE

1. Step detection system. 11

2. Probability of correct detection vs. step size. 22

3. Experimental gas leak scene. 28

4. Original input images. 31

5. Enhanced input images and corresponding output images. 32

CHAPTER I

INTRODUCTION

The detection of changes in a sequence of images is useful for many applications. By comparing satellite images of a forest or desert, land managers can determine if changes have occurred during the time between images [1-4]*. Urban planners can track the progress of cities by comparing images for differences [5]. While these are just a few of the many applications of analyzing a small number of images for changes, applications using longer sequences of images are not as common. Since images in a sequence are usually separated by small amounts of time, applications of image sequence analysis often involve situations where changes can occur in seconds. A large portion of this type of research involves tracking systems [6-7]. A moving object will cause changes in a sequence of images as it moves across a scene. Tracking systems find the object by noting the changes from one image to the next. Robotic vision systems also use methods similiar to tracking systems to determine the location of objects [8].

The method for change detection in a sequence of images presented in this thesis was developed for the application proposed by Shohadaee [9] for detecting gas leaks in rocket engines. In infrared images, a leak should appear as a measurable change in intensity and should extend spatially over the image. Since the gases in the engine are pressurized, a leak can occur in a fraction of a second.

* Numbers in brackets refer to similiarly numbered references in the Bibliography.

Thus, a detection scheme is needed to process images at standard video rates in real-time in order to shut down the engine and minimize damage caused by the leak. Other requirements for the leak detection system include:

1) detection of positive and negative intensity changes

2) detection of relatively small intensity changes

3) suppression of false alarms

4) robustness with respect to noise.

As mentioned above, a leak appearing in a sequence of images occurs quickly and remains in the field of view for some time. Thus, the objective of the system described here is to quickly and automatically detect a step-like change in intensity. Although this system was developed for infrared images, it does not depend upon the type of image data. Thus, it can be implemented in any situation where the intensity at any spatial location in a time sequence of images abruptly changes from its quiescent value to something different.

Time-varying intensity values within a video image constitute an inherently discrete signal with a sampling rate of 30 Hz for standard television format. Thus, techniques of digital signal processsing are directly applicable to this analysis. Currently available image processing hardware utlilizing full-frame arithmetic logic units (ALUs) makes video signal processing at real-time rates possible. The algorithm presented here employs discrete-time filtering and thresholding operations that can be implemented with finite-wordlength arithmetic on small image processing systems to detect abrupt changes in intensity occurring within the field of view of an image.

The detection algorithm used in this thesis was developed as a one-dimensional algorithm implemented at every point in a two-dimensional image array. First, a recursive highpass filter is used to remove slowly changing background intensity values. The highpass filter output is then averaged with a finite number of previous output values. The absolute value of the average is taken and then compared to a threshold value. If the average is greater than the threshold, the decision is made that a change has occurred. For this thesis, a numerical study was conducted using this algorithm. The results of this study are shown as the probability of correct detection versus step size and illustrate the effects of noise on the input and the value of the threshold on the performance of the algorithm. Next, the algorithm was tested on a sequence of images of an actual gas leak. A gas leak scene was set up in the laboratory and sequential image data was acquired. The data was processed on a image processor where the algorithm was implemented at every point in an image. The parameters of the algorithm were varied and the results examined. The results are presented as images indicating regions where change was detected and show the effects of the parameters on the size of the detected region as well as the noise sensitivity of the detector.

The next chapter describes previously developed algorithms for detecting changes and discusses the disadvantages of these algorithms as leak detection systems. In Chapter III, the algorithm is derived and and the methods for choosing the parameters are described. A numerical study was conducted by implementing the algorithm on simulated data. The study and its results will be presented in Chapter IV. The algorithm was also implemented with image processing hardware and laboratory experiments involving actual leaks were performed and analyzed from acquired video data. Chapter V discusses these experiments and shows the results obtained. A summary and recommendations are given in the final chapter.

CHAPTER II

BACKGROUND

Image analysis techniques implemented previously use a variety of detection algorithms. Many of these algorithms involve various forms of temporal differencing [1-4], [10]. Other algorithms described use thresholding [2], [10] or detect changes by image ratioing where the ratio of the previous image to the current image is found and the difference from unity of the ratio indicates a change has occurred [2]. Besides temporal differencing and thresholding, Patterson *et al* [10] have investigated several algorithms which use spatial differencing combined with temporal differencing as well as algorithms using matched filters and peak detection. Another technique, image averaging, involves weighting the images where the weights are chosen to optimize the signal-to-noise ratio [11].

These schemes have several disadvantages which make them unsuitable for detecting leaks in a sequence of images. Since differencing, thresholding, and ratioing algorithms are primarily used to identify changes which occur slowly in time, the frame rate is low, and only the present image and previous image or a few previous images are used by these algorithms. While the spatial differencing algorithms use more sample values to determine that change has occurred, the objective is to detect the location of the jump. Thus, they are computationally intensive and are not causal; hence, they are not amenable to real-time video image processing. Also, a method for dealing with noise effects is not addressed by many of the algorithms.

Algorithms which utilize sequences of images are primarily derived for tracking systems [6-7]. The moving target identification algorithm presented by Reed, Gagliardi, and Stotts [6] computes the summation of a image sequence using a method modeled by the following linear, constant-coefficient difference equation:

$$X_n(k) = Y_n(k) + \alpha(k)X_{n-1}(k) \tag{2.1}$$

where n is the time index, k is the two-dimensional spatial wavenumber vector, $Y_n(k)$ is the Fourier transformed input image, $X_n(k)$ is the image sum Fourier transformed, and $\alpha(k)$ is a function of k and the vector velocity of the target. Since the movement of the object in the images causes a phase shift in the transformed images, multiplying by $\alpha(k)$, the linear shift of the target, rephases the images. A spatial matched filter operates on the summed image and the output image is obtained by inverse Fourier transforming the result. Note that this difference equation implements a recursive digital filter at each point in the transformed image arrays similiar to the method presented here.

Mao and Strickland [7] have also presented a target detection system using sequential image processing which combines spatial and temporal techniques. Each image in the sequence undergoes edge preserving smoothing to reduce noise while preserving the edges of the target. A binary image is then produced by segmenting the image to separate the target from the background; pixels considered as background are set to black, and those of the target are set to white. The binary images are registered using orthogonal projections and then processed temporally. The temporal processing uses a fifth-order nonrecursive median filter given by

$$g(x,y) = \begin{cases} 1, & \text{if } \sum_{j=1}^{5} b_j(x,y) \geq 3; \\ 0, & \text{otherwise.} \end{cases} \tag{2.2}$$

where $b_j(x,y)$ is the binary value at the (x,y) location in the j^{th} registered image. The resulting output image shows the size and shape of the target relative to the background.

These algorithms are also not directly applicable as a leak detection system. Their primary purpose is to locate and identify targets and they are computationally intensive. Furthermore, the algorithm described in [6] requires the rate of change from one image to the next be constant, and the system of [7] assumes that changes occur slowly across the sequence of images. Neither of these assumptions are valid for gas leaks. However, each algorithm is causal and employs image summation which is effective for reducing noise.

Although processing sequences of images is fundamental to the leak detection algorithm, it also involves detecting jumps in the mean value of the intensity of a pixel. The problem of detecting abrupt changes within an image has been studied in the spatial domain as an edge detection problem [4], [12-15]. The approach taken by Basseville, Espiau and Gasnier [13-14] considers each line of the image to be a sequence of independent Gaussian random variables having the same variance. An edge is defined as a jump in the mean value of the sequence ($\mu_0 \rightarrow \mu_0 \pm \Delta\mu$) and is detected using Hinckley's algorithm. If Y_i is the i^{th} sequence sample, Hinckley's algorithm, as described by Basseville [14], computes the cumulative-sum of the sequence, S_n and the maximum of the sum, M_n as shown

$$S_n = \sum_{i=1}^{n}(Y_i - \mu_0 - \frac{\Delta\mu}{2}) \quad (n \geq 1) \tag{2.3}$$

$$M_n = \max_{0 \leq k \leq n} S_k \qquad (S_0 = 0) \tag{2.4}$$

for positive jumps ($\mu_0 \rightarrow \mu_0 + \Delta\mu$). A second detector is required for negative jumps ($\mu_0 \rightarrow \mu_0 - \Delta\mu$) as given by the following equations:

$$U_n = \sum_{i=1}^{n}(Y_i - \mu_0 + \frac{\Delta\mu}{2}) \quad (n \geq 1) \tag{2.5}$$

$$m_n = \min_{0 \leq k \leq n} U_k \qquad (U_0 = 0) \tag{2.6}$$

When $M_n - S_n \geq h$ or $U_n - m_n \geq h \quad (h \geq 0)$, a jump has been detected. The mean value of the sequence, μ_0, is estimated by a Kalman filter. The minimum change amplitude $\Delta\mu$ and the threshold h are set *a priori*.

Although a leak could be considered to be a temporal edge similiar to a spatial edge in an image, edge detection schemes have drawbacks making them unsuitable for use as a leak detection algorithm. Edge detection algorithms are computationally intensive since they must determine the location of the edge. Also, they are non-causal which is not applicable to analyzing a time sequence of images in real time.

CHAPTER III

THE STEP DETECTION ALGORITHM

The work performed in this thesis is based on the algorithm proposed by Smith [16]. For completeness and consistency of notation, the following section is included and describes this algorithm*.

3.1 Algorithm Description

The overall goal of the step detection system is to provide binary step-/no-step-occurrence decisions at the input data sampling rate. In addition, the following properties are highly desirable:

1. The algorithm should be computationally efficient, for rapid real-time processing of video data.
2. It should be impervious to different background or quiescent intensity levels at separate locations within the image, and to signal noise.
3. It should be capable of detecting steps over a wide range of amplitudes, both positive and negative.

The system developed to achieve these objectives is shown in Fig. 1. The input data is first filtered with a highpass filter to remove the slowly or non-

---

* The following is an excerpt taken from an unpublished article entitled "A system for sequential step detection with application to video image processing" written by Malone and Smith.

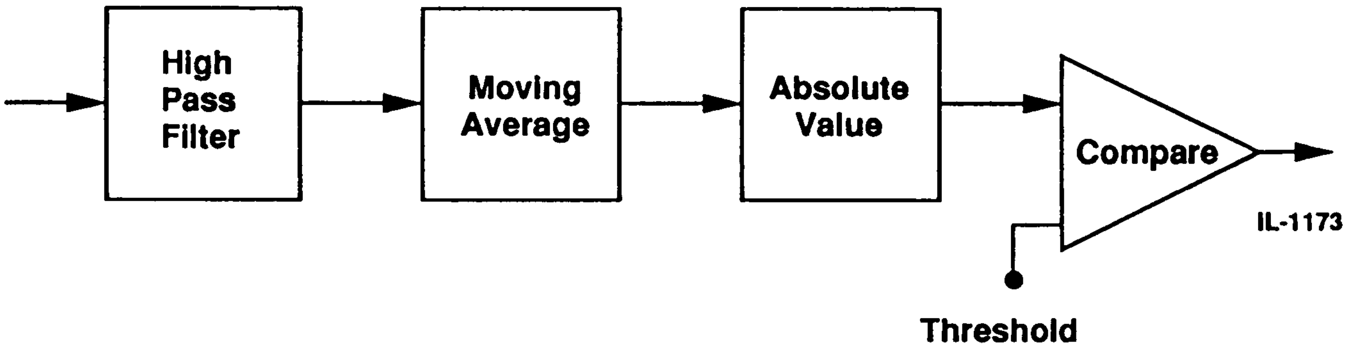


Fig. 1. Step detection system.

varying background intensity level while allowing any sudden changes to be passed. The output of the highpass filter is next input to a moving average filter which sums over the present and previous J sample values. Thus, only a change that is maintained will cause a substantial change in the average. The absolute value block creates a positive value in case the change was negative, and the result is compared to a threshold to decide if a step has occurred. An entire image is analyzed by implementing the algorithm at every point in the image.

The following discussion describes the step detection algorithm in more detail. To simplify notation, all signals are written simply as functions of the time variable index n. For video images, all signals depend upon two spatial position coordinates in addition to the time variable. However, because processing is identical for all spatial coordinates, the spatial dependence is suppressed in the notation.

For computational efficiency and rapid response, the highpass filter was chosen to be a single-pole unity-gain recursive digital filter with a z-domain transfer function given by

$$H(z) = \frac{1+\beta}{2}\left(\frac{1-z^{-1}}{1-\beta z^{-1}}\right). \tag{3.1.1}$$

The cutoff frequency of the filter is set by choice of the pole value β. The relationships between β and the -3 db normalized discrete cutoff frequency ω_c are given by

$$\beta = \frac{1 - \sin\omega_c}{\cos\omega_c}, \qquad \omega_c = 2\sin^{-1}\frac{1-\beta}{\sqrt{2(1+\beta^2)}}. \tag{3.1.2}$$

Note that the use of a recursive or infinite impulse response (IIR) filter utilizes all of the past information in the input signal. Also, use of a single-pole filter produces only a small filter delay; hence, the delay in detecting a sudden change is also small.

If the output of the highpass filter is denoted by $x(n)$, the moving average computes its output as

$$y(n) = \frac{1}{J}\sum_{k=0}^{J-1} x(n-k). \tag{3.1.3}$$

The magnitudes of the sample values of $y(n)$ are then compared to a preset threshold value T and if $|y(n)| \geq T$, the decision is made that a step has occurred, otherwise it is decided that no step has occurred.

It can be shown that for a given highpass filter cutoff frequency ω_c with corresponding pole value β, an optimal choice exists for the number of terms in the moving average J. To show this, assume an input signal of the form

$$w(n) = b(n) + Au(n) + \epsilon(n), \tag{3.1.4}$$

where $b(n)$ represents a slowly-varying background intensity level, $u(n)$ is a unit step that is assumed to occur at $n = 0$ for simplicity of notation, and $\epsilon(n)$ is a zero-mean uncorrelated random sequence modeling the noise corrupting the signal. If it is assumed that the highpass filter completely removes the background, then from the transfer function given in (3.1.1), its output can be shown to be

$$x(n) = \tfrac{1}{2}A(1+\beta)\beta^n u(n) + h(n) * \epsilon(n), \tag{3.1.5}$$

where $h(n)$ is the impulse response of the filter in (3.1.1) and the asterisk $(*)$ denotes convolution. In practical applications, the cutoff frequency of the highpass filter is usually low. Thus, its effect on the spectral and statistical properties of the white noise sequence $\epsilon(n)$ is small. With the assumption that this effect is negligible, to a close approximation, the sequence $x(n)$ can be written

$$x(n) \cong \tfrac{1}{2}A(1+\beta)\beta^n u(n) + \epsilon(n). \tag{3.1.6}$$

The moving average filter sums the previous J values of this sequence. The maximum amplitude of the signal resulting from the step at $n = 0$ occurs at $n = J - 1$. At this instant, $y(J-1)$ is given by

$$y(J-1) = \frac{A}{2J}\left(\frac{1+\beta}{1-\beta}\right)(1-\beta^J) + \frac{1}{J}\sum_{k=0}^{J-1}\epsilon(n-k). \tag{3.1.7}$$

The standard deviation of the J-point average of $\epsilon(n-k)$ is given by $\sigma/\sqrt{J}$, where σ is the standard deviation of the white noise sequence $\epsilon(n)$. Thus, a signal-to-noise ratio for this system can be defined as the ratio of the magnitude of the maximum signal step response (the first term on the right-hand side of (3.1.7)) to the standard deviation of the processed noise signal:

$$\frac{S}{N} = \frac{|A|}{2\sigma}\left(\frac{1+\beta}{1-\beta}\right)\left(\frac{1-\beta^J}{\sqrt{J}}\right). \tag{3.1.8}$$

This signal-to-noise ratio, considered as a function of J, can be maximized in the following manner. Differentiating this expression with respect to J and setting the result to zero yields the optimal value as the solution to the transcendental equation

$$\beta^J(1 - 2\ln\beta^J) = 1, \tag{3.1.9}$$

which can be solved numerically to yield

$$J_{opt} = -\frac{1.2564}{\ln\beta}. \tag{3.1.10}$$

In practice, J is chosen to be the nearest integer to the value computed in (3.1.10). Thus, from a given cutoff frequency chosen to remove the background component from the signal, the highpass filter pole value β is calculated from (3.1.2), and a value for the moving average summation J is chosen from (3.1.10) to maximize the signal-to-noise ratio of the filtered sequence.

As a final comment, note that in the absence of noise, a step will be detected provided that its amplitude is sufficiently large such that the magnitude of the signal term in (3.1.7) exceeds the threshold value. That is, with no noise corrupting the signal, a step must have an amplitude satisfying

$$|A| \geq \frac{2JT}{1-\beta^J}\left(\frac{1-\beta}{1+\beta}\right), \tag{3.1.11}$$

to be detected. This expression provides a lower limit of detection which is useful in evaluating the statistical performance of the step detection algorithm.

3.2 Implementation Considerations

Implementing the step detection algorithm on a practical digital image processing system requires paying special attention to several aspects of the algorithm. Since the algorithm is implemented at every point in an image array, arithmetic operations of the filters involve large numbers of computations. However, frame buffers in digital image processors are usually configured for fixed-point or integer pixel values. Since only a finite number of products result for multiplications by a constant-valued filter coefficient, multiplications are realized via look-up-tables (LUTs) containing precomputed values for each possible product.

Implementation of the recursive highpass filter can be accomplished with minimal storage requirements and with reasonable computational efficiency by using a state-space filter structure. The output $x(n)$ is computed from the input $w(n)$

and a state variable $v(n)$ by means of the following two equations

$$\begin{aligned} v(n+1) &= av(n) + bw(n) \\ x(n) &= cv(n) + dw(n), \end{aligned} \tag{3.2.1}$$

where a, b, c and d are constant-valued coefficients chosen to realize the transfer function of (3.1.1). Specifically, these coefficients must satisfy

$$H(z) = \frac{bcz^{-1}}{1 - az^{-1}} + d. \tag{3.2.2}$$

This determines the values for a and d as $a = \beta$, and $d = \frac{1}{2}(1 + \beta)$.

Because of the fixed-point or integer format of the two-dimensional array element values, overflow in the state variable computation must be eliminated. This is accomplished by using L_∞-norm scaling [17] which, for a stable first-order filter with positive pole value, ensures that the magnitude of the state variable never exceeds that of the input. This sets the values for b and c as $b = 1 - \beta$, and $c = -\frac{1}{2}(1 + \beta)$.

To avoid transients in filtered data at start-up, the state variable is initialized to produce zero output for the first sample. This is accomplished by using the first input sample to compute its initial value by

$$v(0) = -\frac{d}{c}w(0). \tag{3.2.3}$$

Since the step amplitude is not known *a priori*, the threshold value must be chosen to reduce the probability of a false detection to an acceptable level. If the effect of the highpass filter on the input noise is assumed negligible as discussed in the previous section, then the processed data in the absence of a step input consists simply of a random sequence with standard deviation $\sigma/\sqrt{J}$. If it is futher assumed that this random signal is Gaussian distributed, the probability of a false detection at each sample instant will be 2×10^{-4} for a threshold value of

$$T = \frac{3.70\sigma}{\sqrt{J}}. \tag{3.2.4}$$

Other false detection probabilities can be realized by choosing other constants of proportionality in accordance with tabulated Gaussian probabilities. The value given in (3.2.4) has been found to provide acceptable performance for practical applications.

CHAPTER IV

NUMERICAL STUDY

The performance of the algorithm under controlled conditions was tested by implementing it on simulated data in a FORTRAN program. Several parameters of the algorithm were varied and the results displayed as plots showing the probability of correct detection versus amplitude of the sudden step change (step size). The numerical study and the results are discussed below.

4.1 Procedure

To evaluate the statistical performance of the step detection algorithm, a numerical study was conducted by first implementing the algorithm in floating-point arithmetic using the FORTRAN program listed in the Appendix. The inputs to the algorithm were 256-point data blocks where the time variable index n ranged from 0 to 255. For each n, the value of the input $w(n)$ as given by (3.1.4) was generated with $b(n)$ an arbitrary constant, and the step occurred at some random time t_0, uniformly distributed over each 256-point data block. The additive noise term $\epsilon(n)$ was an uncorrelated Gaussian random sequence given by

$$\epsilon(n) = \sigma\sqrt{-2\ln u_1(n)}\cos[2\pi u_2(n)] \tag{4.1.1}$$

where $u_1(n)$ and $u_2(n)$ were random numbers uniformly distributed on the interval

(0,1) generated by subroutine RANUM. The state-space filter structure in (3.2.1) was used to remove the background intensity and update the state variable for the next input value. The moving average filter then averaged the present value of the highpass filter output $x(n)$ with the previous $J-1$ terms of $x(n)$ where $x(n) = 0$ for $n < 0$. The output of the moving average $y(n)$ was then compared to the threshold. If $|y(n)| < T$, the step was not detected, and the next input value was generated and processing continued. However, if $|y(n)| \geq T$ the program then checked for $n \geq t_0$. If $n \geq t_0$, the program indicated that the step was correctly detected and processing stopped. When $|y(n)| \geq T$ but $n < t_0$, the program indicated that the step was not correctly detected and again did not generate or process any more input values for that data block. The procedure was repeated for each value of n until either a step was detected or an entire sequence was processed.

To obtain a measure of the performance of the algorithm, the process above was performed for 156 step amplitudes ranging from 0 to 155. Signal and noise amplitude levels in this study were chosen to correspond to the range of gray levels in an 8-bit per pixel digital image processor. For each step amplitude, 100 different input sequences were generated and processed. In the sequences, the step occurred at different times and each had a different sequence of additive noise with the same standard deviation. Each time the algorithm correctly detected the step, a counter was incremented. The number of correct detections out of the 100 trials was recorded for the step size used and the process was repeated for all step sizes from 0 to 155. Thus, a test consisted of 15,600 sequences all having the same input noise variance and compared to the same threshold. Finally, the number of correct detections for each step size was converted to an estimated probability and plotted as a function of the step size. Tests were performed for input noise standard deviations ranging from 10 to 40.

4.2 Results

Figure 2 shows the results of performing these tests with a highpass filter pole value $\beta = 0.7951$ for which the optimal number of terms in the moving average as given by (3.1.10) is $J = 5$. The threshold was fixed at $T = 33.1$ which corresponds to using the criterion in (3.2.4) with an assumed input noise standard deviation of $\sigma = 20$. The four plots shown are the probability of correct detection versus step amplitude for actual input noise standard deviations chosen to be 10, 20, 30 and 40. Some uncertainty exists in these plots since the probabilities were calculated with only 100 trials. Had more sequences been used the results would be more accurate. Futhermore, limiting the sequences to 256 points affects the performance evaluation, since longer sequences would increase the chances of false detections prior to the actual step occurrence.

This series of plots illustrates the relationship between the standard deviation of the input noise (used to generate the additive noise) and the assumed standard

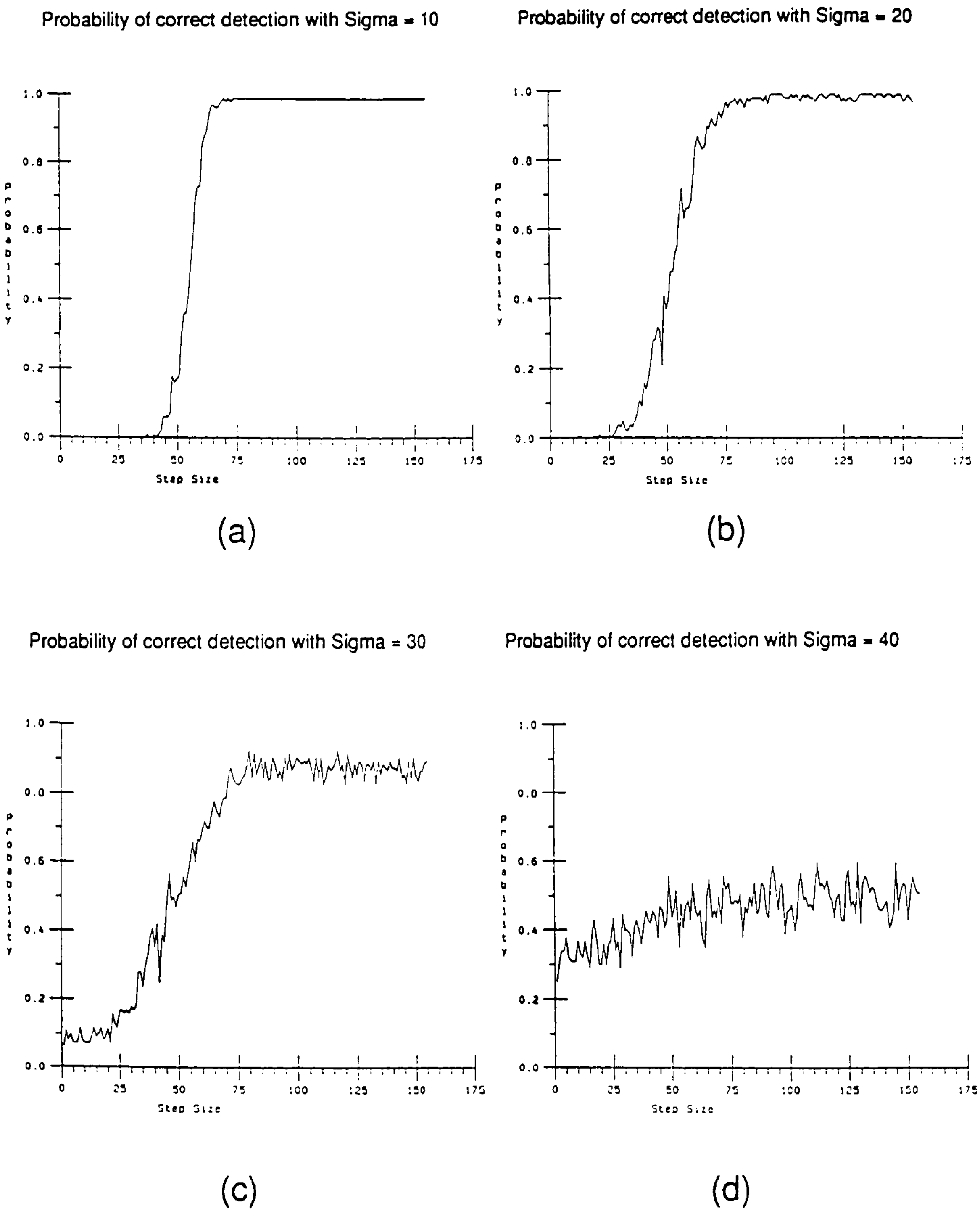


Fig. 2. Probability of correct detection vs. step size.

deviation used to set the threshold. If the level of noise in the input was overestimated, as in Fig. 2(a), the detector was very accurate for large step sizes and the minimum step size detected was close to the value 55.4 given by (3.1.11) for this example. Figure 2(b) demonstrates the performance of the algorithm for the threshold criterion described in Chapter III where the assumed noise standard deviation exactly matched that of the noise actually in the input. As the standard deviation of the input noise was increased, as shown in Fig. 2(c) and 2(d), the algorithm becomes less effective. Although smaller steps were detected more frequently, this increase was due to serendipitous effects of noise causing the processed signal magnitude to exceed the threshold after the occurrence of the step. The noise also caused false detections prior to the occurrence of the step which resulted in decreased performance for large step amplitudes. Note that once the standard deviation of the noise on the input became twice as large as the standard deviation used to set the threshold (Fig. 2(d)), performance of the algorithm deteriorated below acceptable levels.

As the tests described above demonstrate, the performance of the algorithm is affected by the magnitude of the step and the amount of input noise. However, several parameters of the data that do not affect performance are the background value, the time of occurrence of the step, and the direction (positive or negative) of the step. As shown by the transfer function (3.1.1), the highpass filter assures that any constant background intensity value is removed before processing the data block. Also, with no noise on the input, linear shift-invariant processing allows the step to be detected regardless of the time at which it occurs. However, when the input contains noise, false detections prior to the occurrence of the step will prevent detection of the step in the evaluation program. Finally, the the sign of the step is irrelevant since the absolute value of the average is taken before comparing it to the threshold. The study conducted verified these properties. The highpass filter was able to remove the background intensity regardless of magnitude. The time of occurrence was varied from sequence to sequence and did not affect the ability of the algorithm to detect the step provided that a false detection did not occur. The test was conducted with positive and negative step amplitudes and the results were identical for each.

4.3 Summary

This study showed the algorithm to be an effective method for detecting the occurrence of an abrupt change in signal amplitude. Performance of the algorithm depended upon proper choice of the threshold; the best results were obtained when the threshold was set according to (3.2.4) with a standard deviation close to that of the actual input noise. The plots demonstrate that when the noise is accurately estimated, the minimum detectable step amplitude as given by (3.1.11) is the step size for which the detector begins to detect step amplitudes with a probability of fifty percent or greater. This study also showed that in a practical implementation the values for the highpass filter pole value and the number of terms in the moving average must be carefully considered since they help determine the minimum detectable step.

CHAPTER V

EXPERIMENTAL STUDY

The algorithm was implemented on an image processor to evaluate its ability to detect leaks in an actual image sequence. A laboratory scene was created which modeled an actual gas leak in an engine but on a much smaller scale. Then, a gas leak was started with a picture of the space shuttle main engine as a background. A video camera was used to record images of this scene before and after the leak was started. The algorithm was then used to process the acquired image data. The parameters of the algorithm were varied to determine their effect on detecting the leak. Photographs of the input and output images show the results of a test performed on one set of images. The experimental procedure and results are discussed below.

5.1 Procedure

To test the algorithm's ability to detect changes in an actual image sequence, it was implemented on a digital image processor* using the techniques described in Chapter 3. The image processor used for this experiment was a personal-computer-based system which acquired and stored image data in 512×512 pixel

* The detection program was written by Dr. L. Montgomery Smith, a professor at The University of Tennessee Space Institute. This author was granted permission to use the program by Dr. Smith.

format with 8 bits per pixel. The detection program written for this experiment processed a sequence of images off-line which were previously acquired and stored on disk on the system. The program would request the name of the image file and displayed the chosen image on the display monitor. This frame was input to the detection algorithm and processed. Once the image was processed, the output of the algorithm was then displayed as a binary image. At any location where a step-like change was detected the pixel was set white; locations where no change was detected were displayed as black. Once the indication was made to continue, another image file was read from disk, shown on the display monitor, and the process repeated.

For the experiment, an RS-170 format, solid-state charge-injected-device (CID), visible wavelength camera was connected to the image processor for acquiring live video of a gas leak set-up in the laboratory. The experimental set up is shown in Fig. 3. The gas leak was created using a thermos filled with liquid nitrogen. The thermos was sealed with a rubber stopper and vented with copper tubing which directed the leak into the field of view of the camera. The leak was turned on and off electronically by a solenoid valve placed on one end of the copper tubing. A second piece of copper tubing inserted in the stopper was equipped with a mechanical valve and used to regulate the flow of the leak. As the liquid nitrogen warmed, nitrogen gas was forced up the tubing. A neon light was included in the upper right-hand corner of the scene with the leak and was turned on at the same

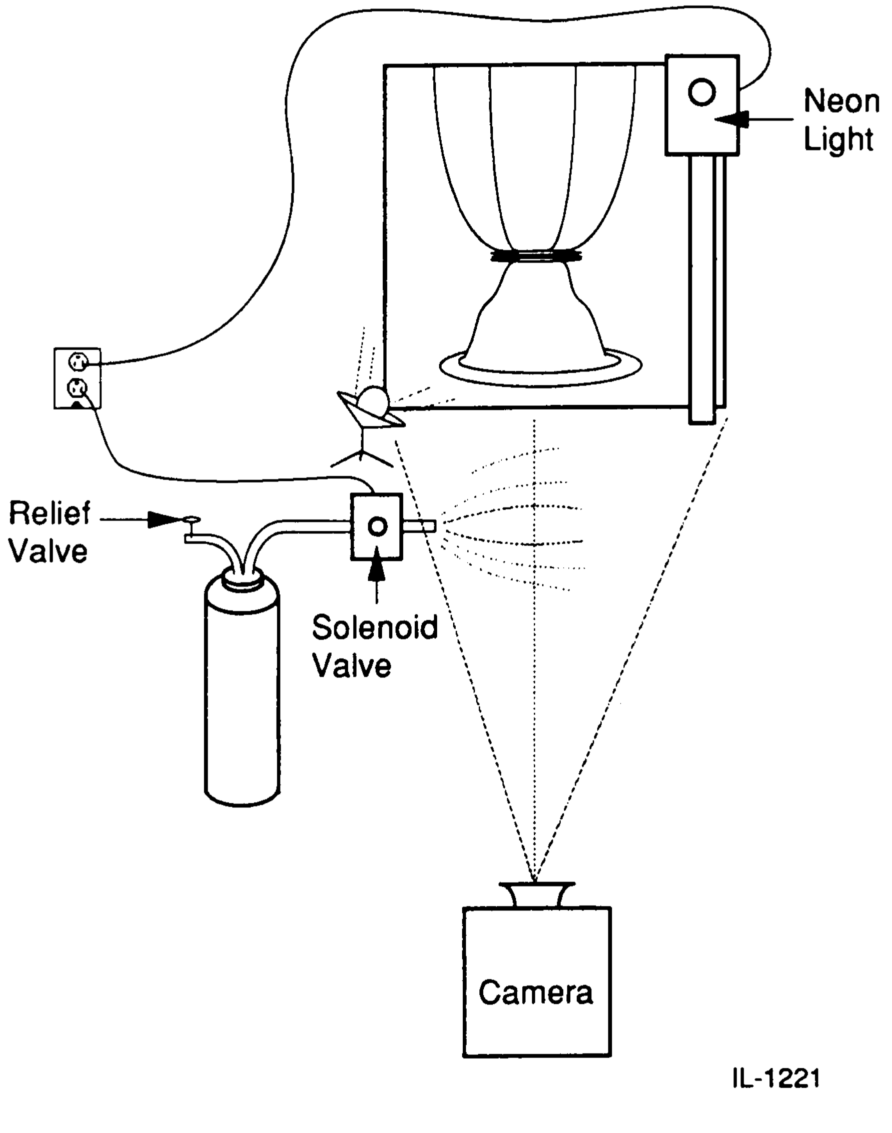


Fig. 3. Experimental gas leak scene.

time as the solenoid valve. This provided a means for determining the image frame in which the leak was initiated. A complicated picture of the space shuttle main engine was used as a background for the scene and was located approximately 33 cm behind the thermos and 170 cm from the camera lens. An incandescent light was positioned at the base of the background picture and directed toward the leak to illuminate the scene. The leak was made visible by the forwardscattering of the light from the condensed water vapor droplets created by the cold nitrogen gas. The field of view of the camera was approximately 40 cm by 40 cm.

For each experimental run, an image sequence of this scene was acquired and stored using the image processor. The software package on the image processor permitted acquisition of a sequence of up to sixteen consecutive frames of live video. Thus, sixteen frame buffers of the image processor were loaded with images of the test scene during one experimental run. To get images before and after the start of the leak, image acquisition was begun and a moment later the switch was thrown to start the leak. The video camera that was used to acquire the image data operated at 30 Hz, and interlaced two fields to create one frame. The camera had a 240×388 detector array, and the information on each row of the detector was converted to an analog signal and transmitted to form one field of the frame. One sixtieth of a second later, the new information on the camera's detector was sent to form the remaining rows in the second field of the frame. Acquired data was digitized into 512×512 arrays. After acquisition, each frame of the sequence was converted to two images by separating the two fields. Each row of the fields was then repeated to create the new images in the same aspect ratio as the acquired image. In this way, the sampling rate was effectively increased to 60 Hz and the image data doubled.

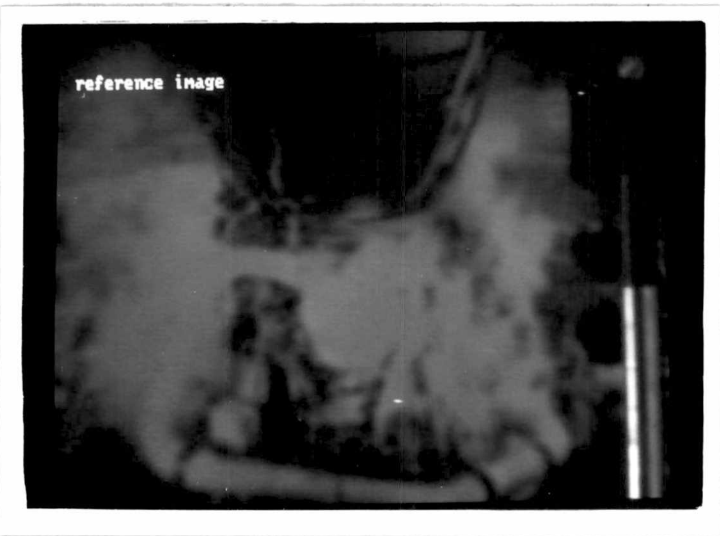
Nine sets of data were acquired in the manner described above. The strength of the gas leak was varied between sets. Also, the lighting on the leak was altered which affected the contrast of the background picture and the leak.

5.2 Results

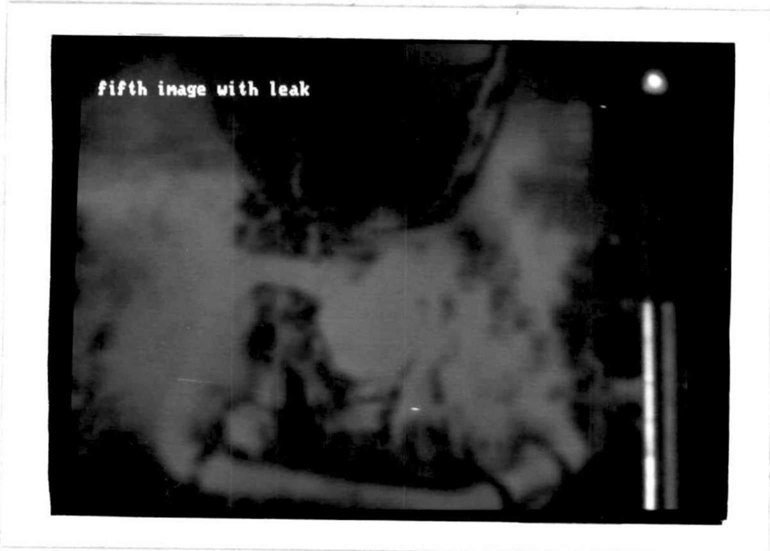
The acquired image data was processed several times using the detection program described above. The algorithm was tested using different values for β, J, and the threshold. The individual tests were compared to determine the effects of varying the parameters of the program. The input and corrsponding output images of one of these tests are presented.

The photographs in Fig. 4 are taken from the display monitor of the image processor and show examples of the type of input image data taken from one run of this experiment. Fig. 4(a) was taken prior to the leak and Fig. 4(b) is the fifth image after the leak was started (Note that the neon light is on in (b) but not in (a).)

Figure 5 contains the enhanced input images and the corresponding output images. Because the changes caused by the leak are difficult to see, these input

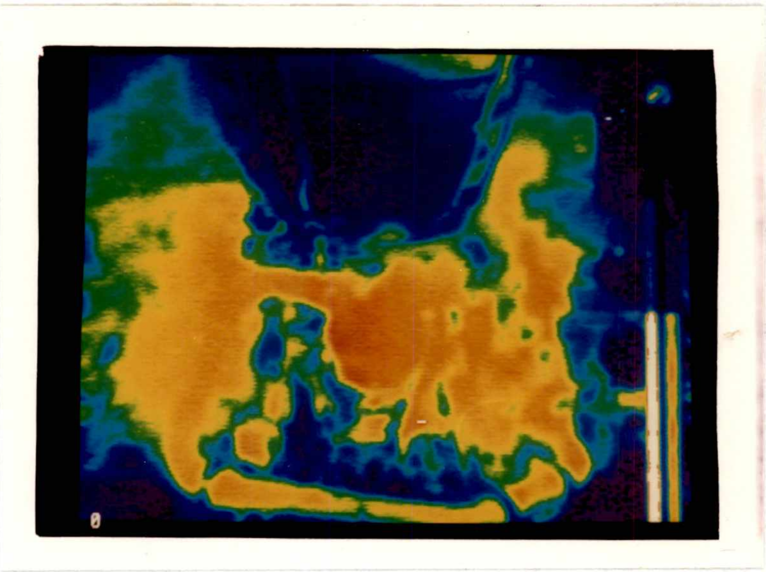


(a)

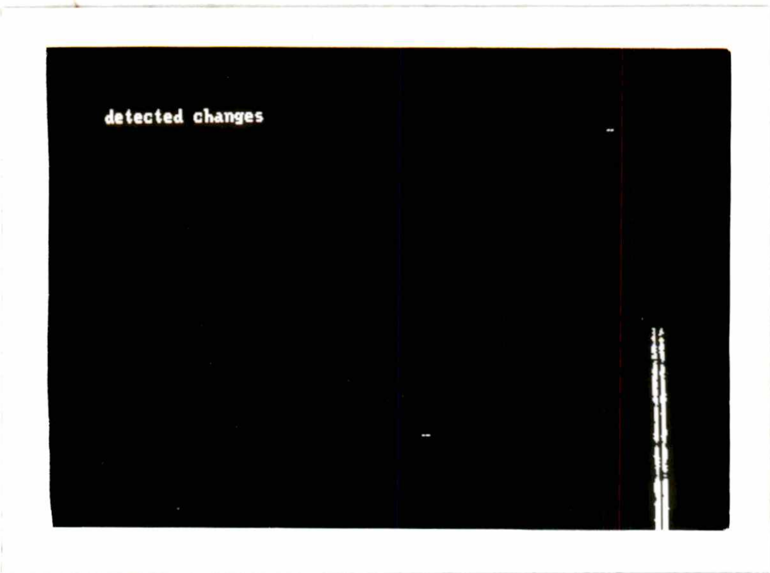


(b)

Fig. 4. Original input images.

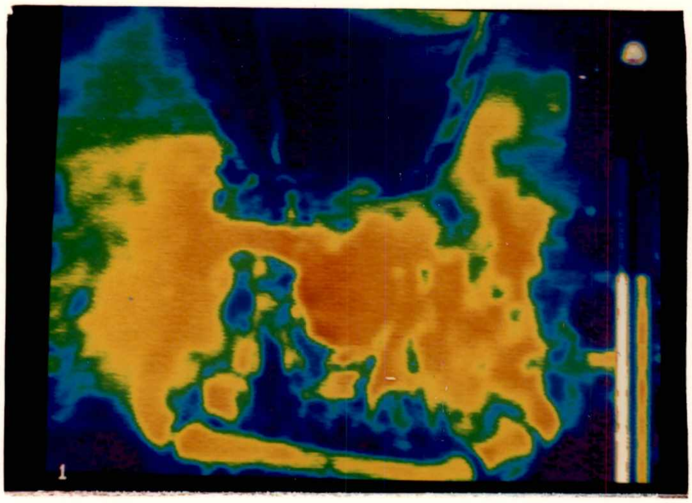


(a)

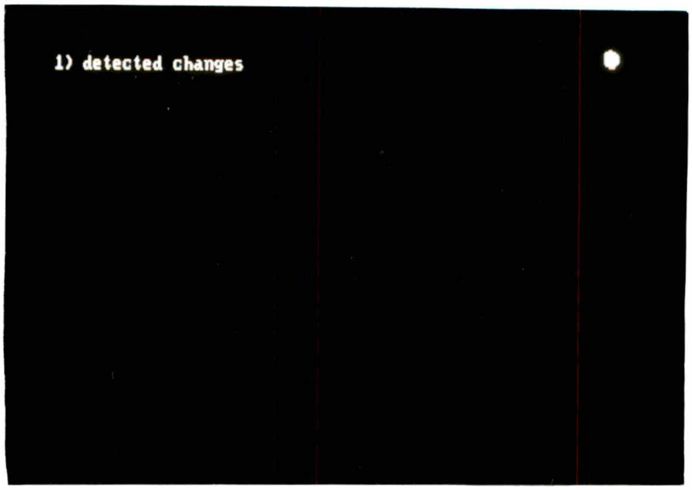


(b)

Fig. 5. Enhanced input images and corresponding output images.

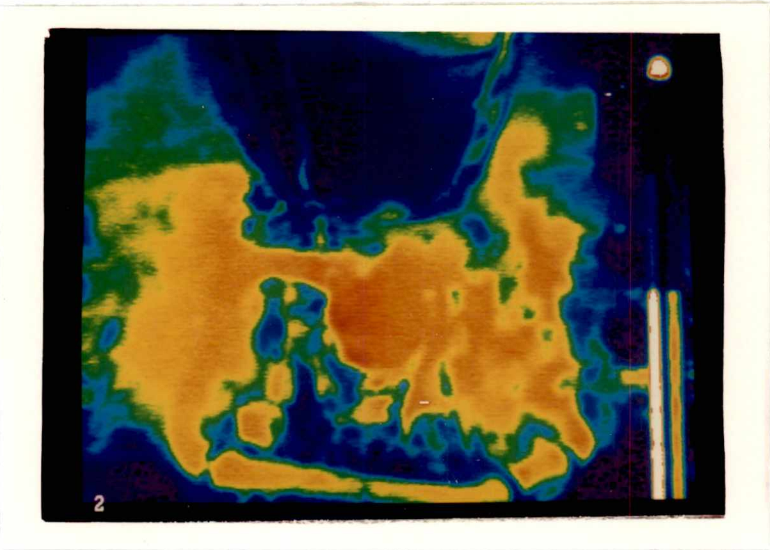


(c)

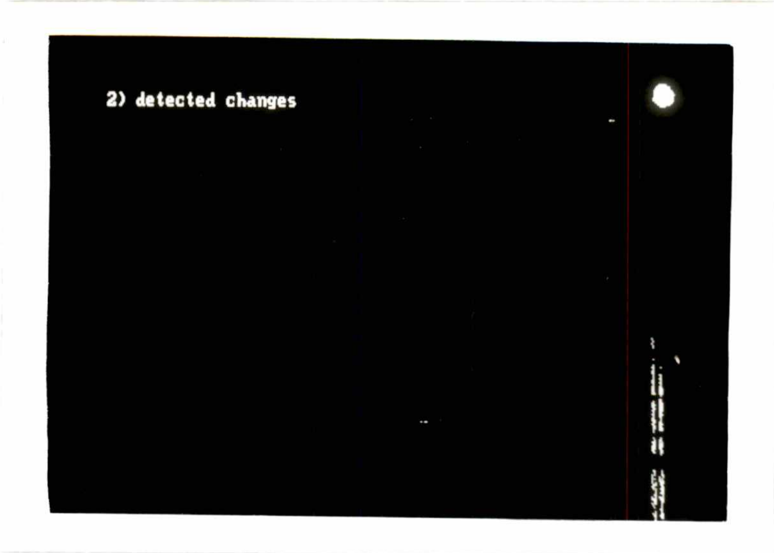


(d)

Fig. 5. (cont.)

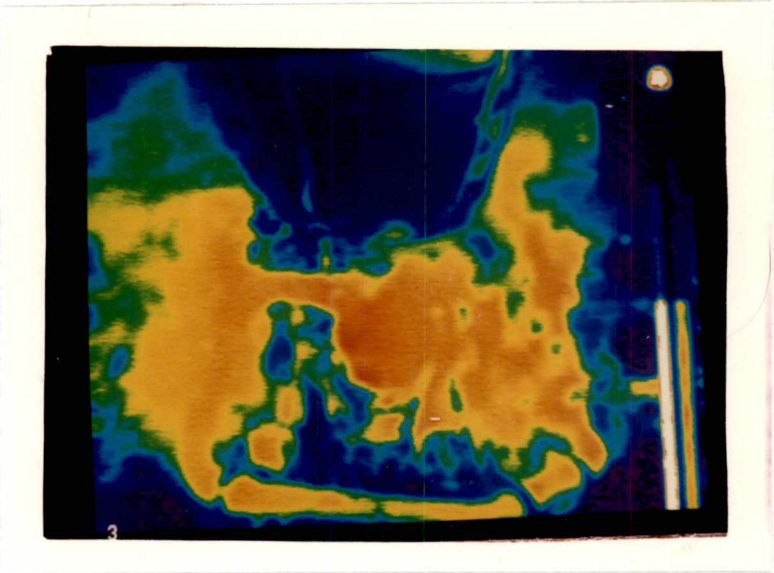


(e)

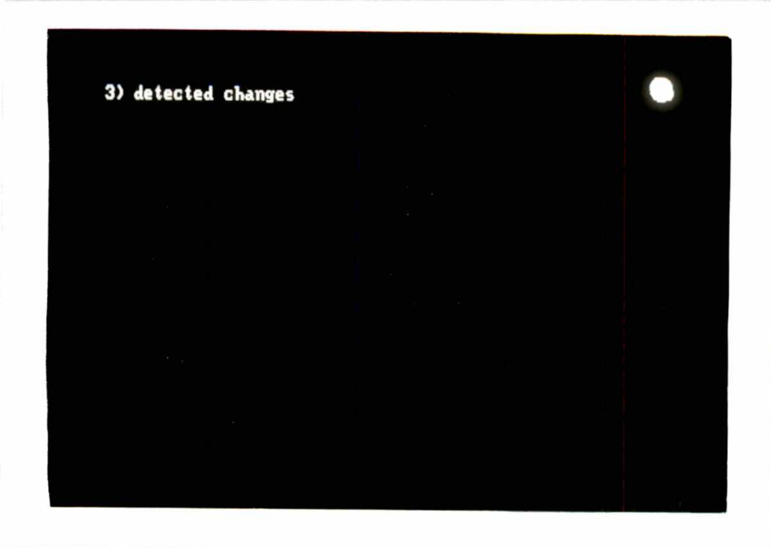


(f)

Fig. 5. (cont.)

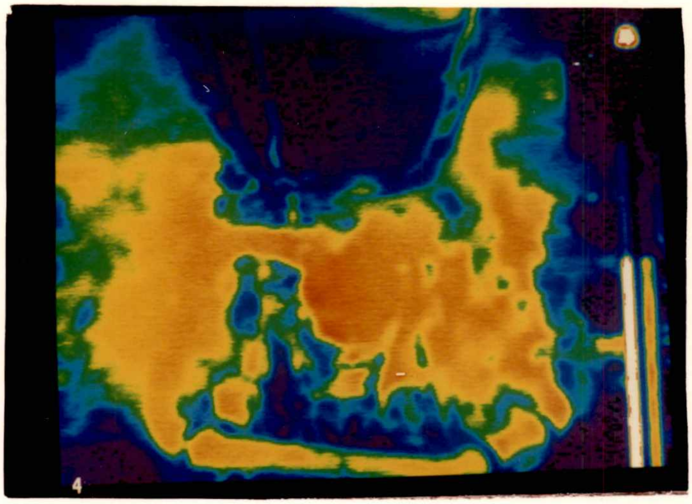


(g)

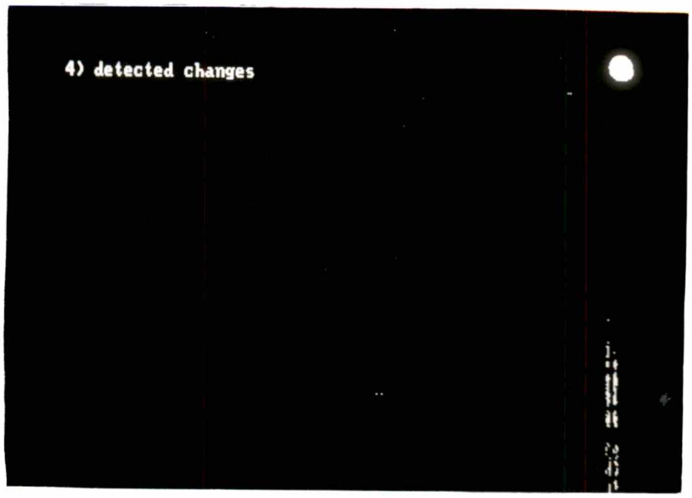


(h)

Fig. 5. (cont.)

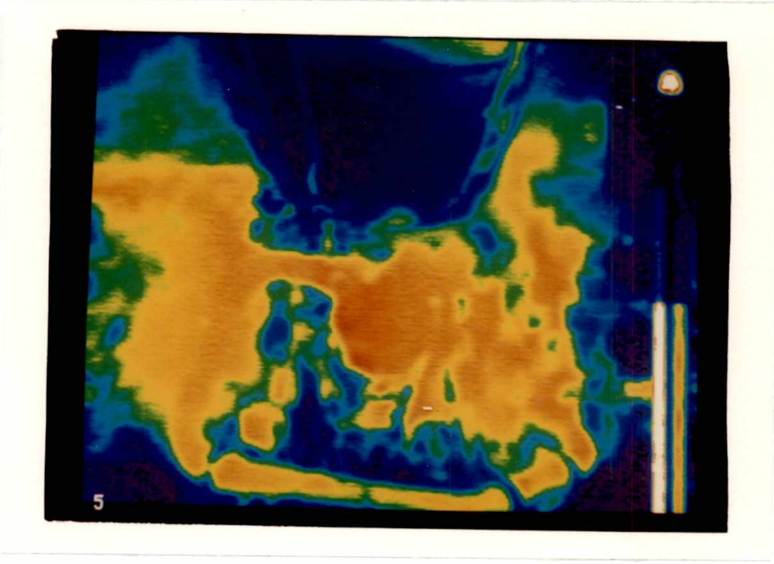


(i)

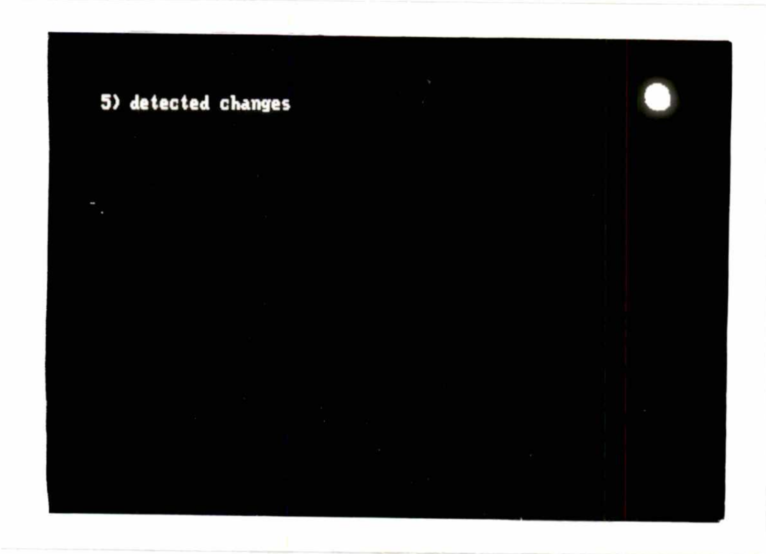


(j)

Fig. 5. (cont.)

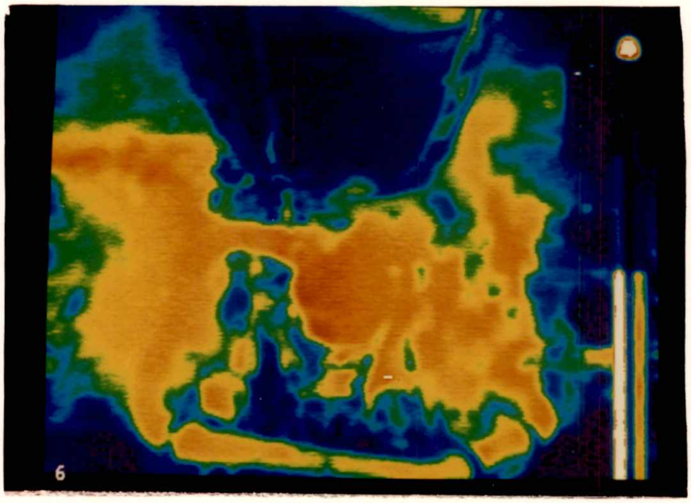


(k)

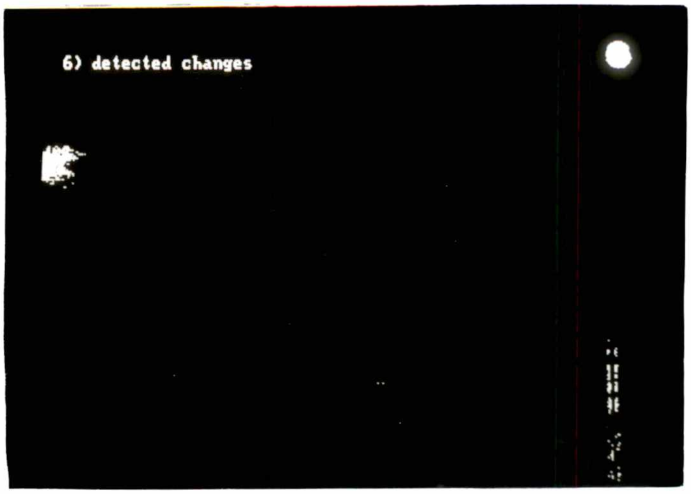


(l)

Fig. 5. (cont.)

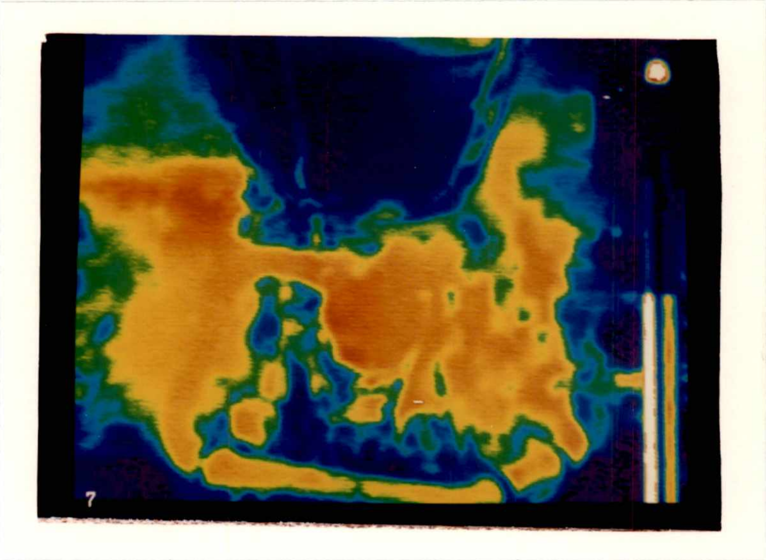


(m)

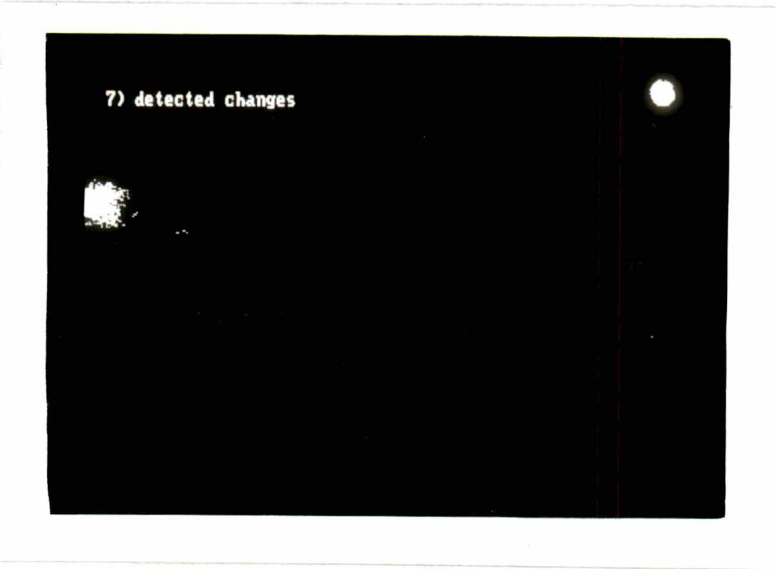


(n)

Fig. 5. (cont.)

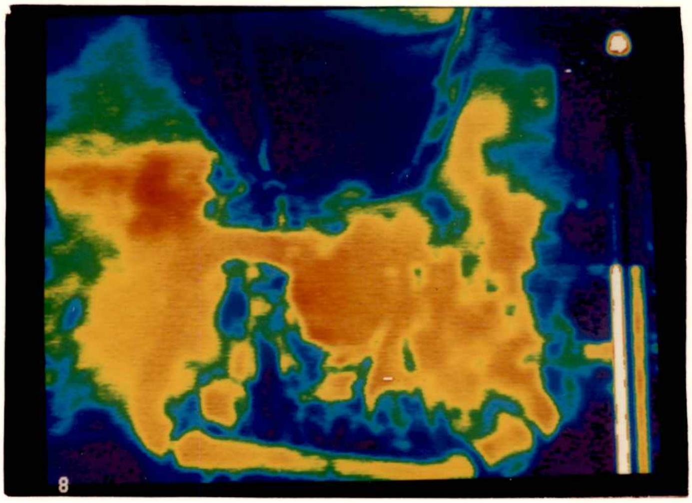


(o)

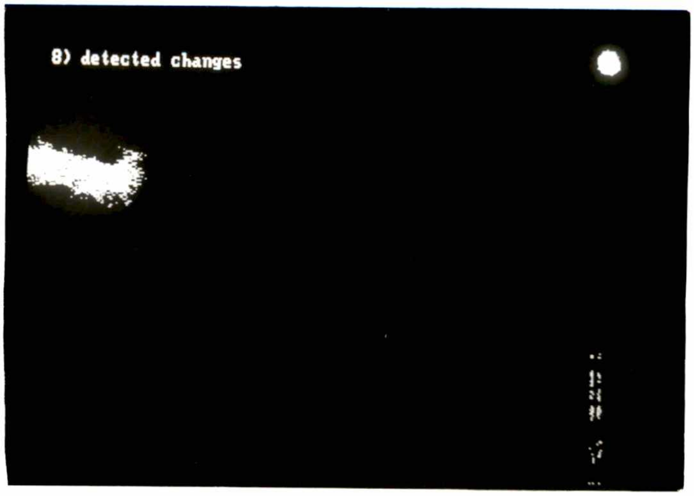


(p)

Fig. 5. (cont.)

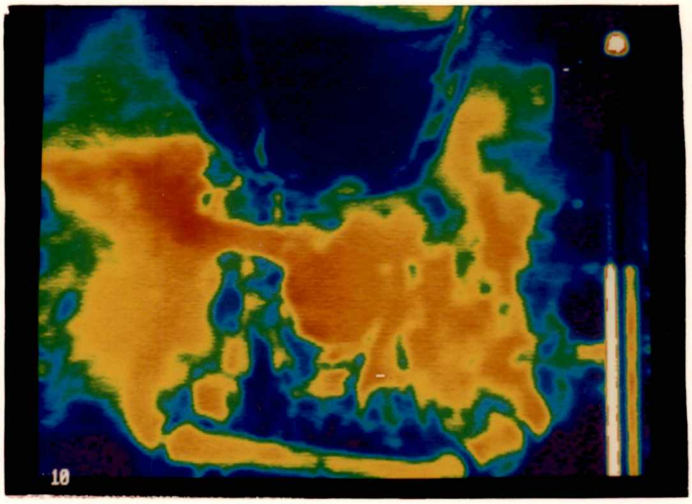


(q)

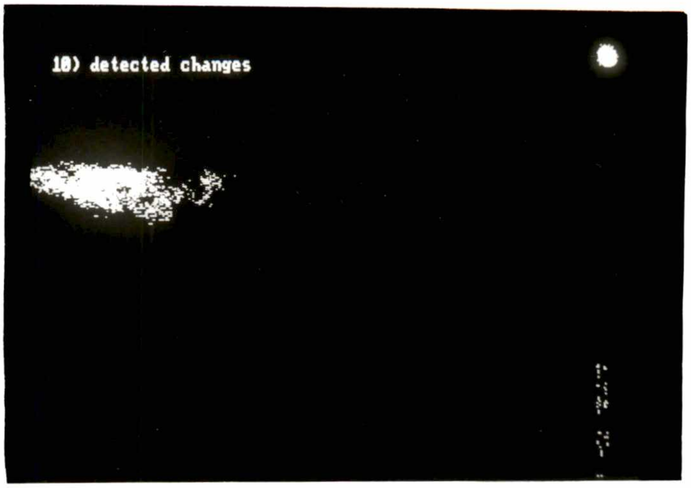


(r)

Fig. 5. (cont.)

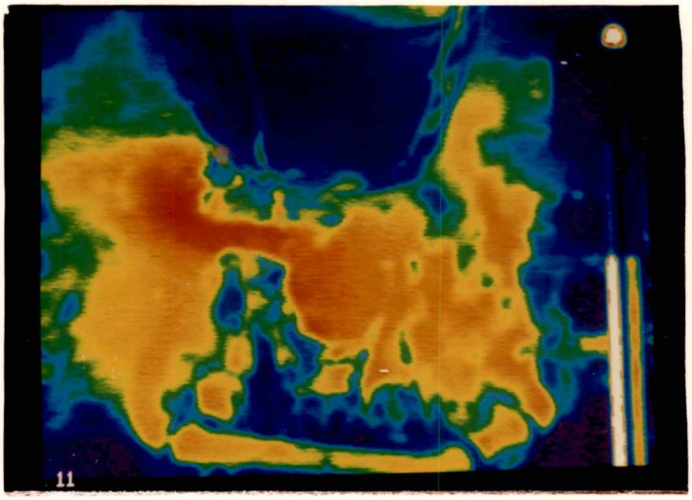


(u)

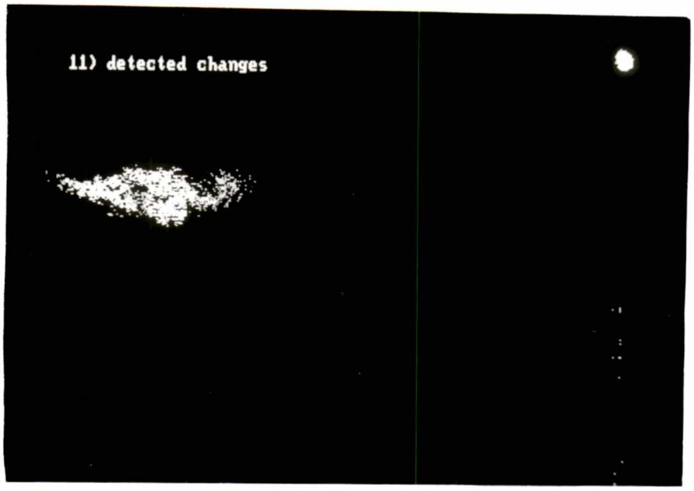


(v)

Fig. 5. (cont.)

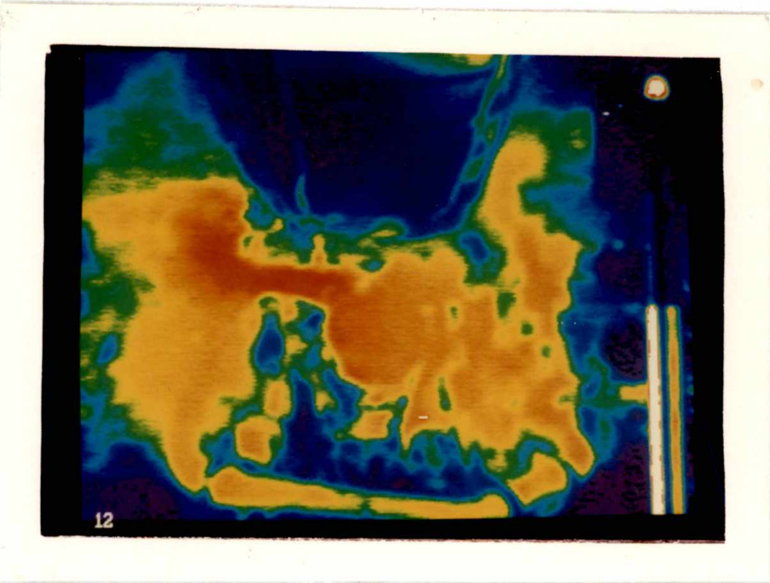


(w)

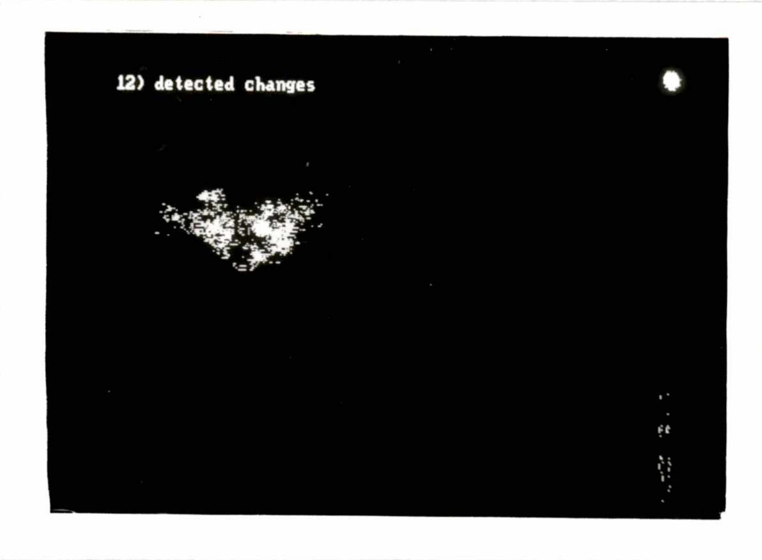


(x)

Fig. 5. (cont.)

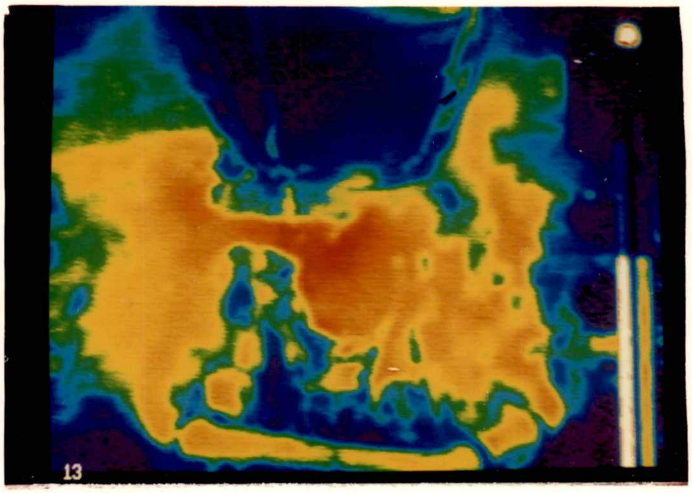


(y)

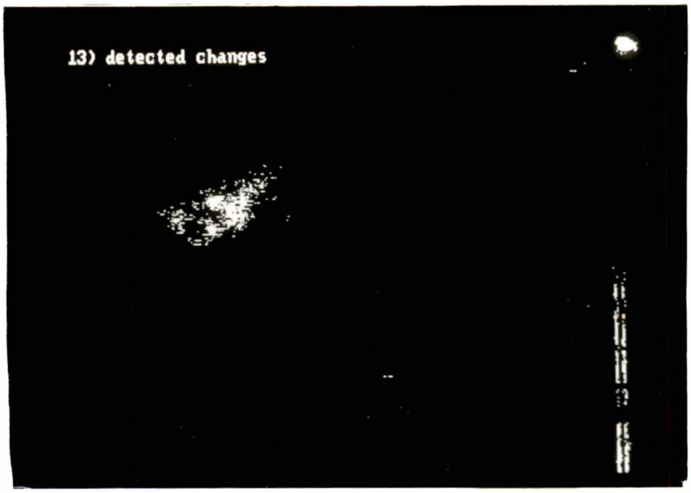


(z)

Fig. 5. (cont.)

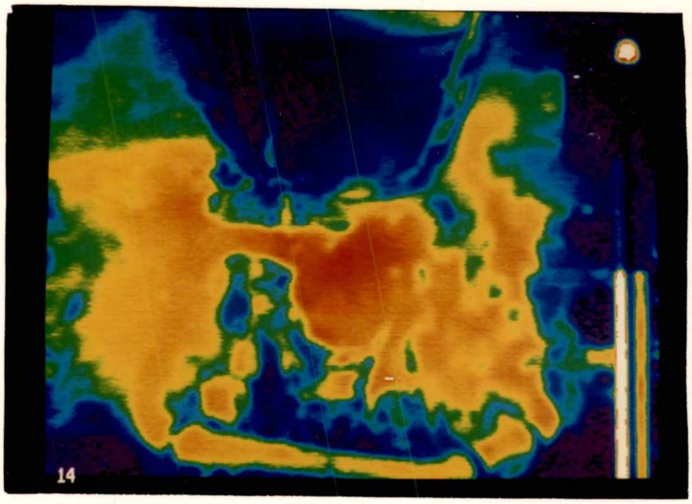


(aa)

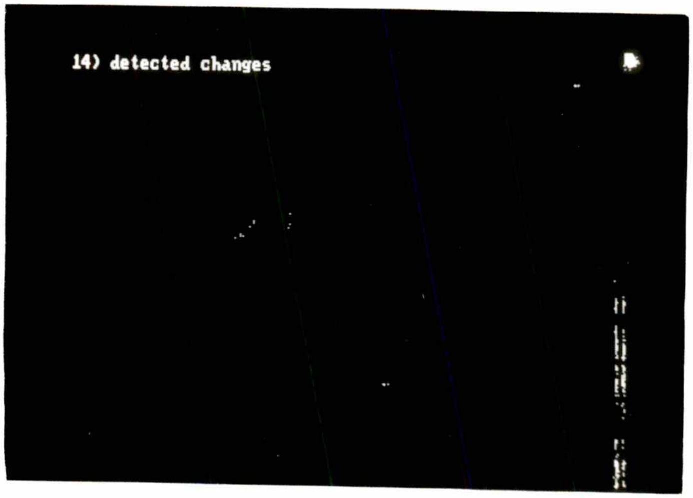


(bb)

Fig. 5. (cont.)

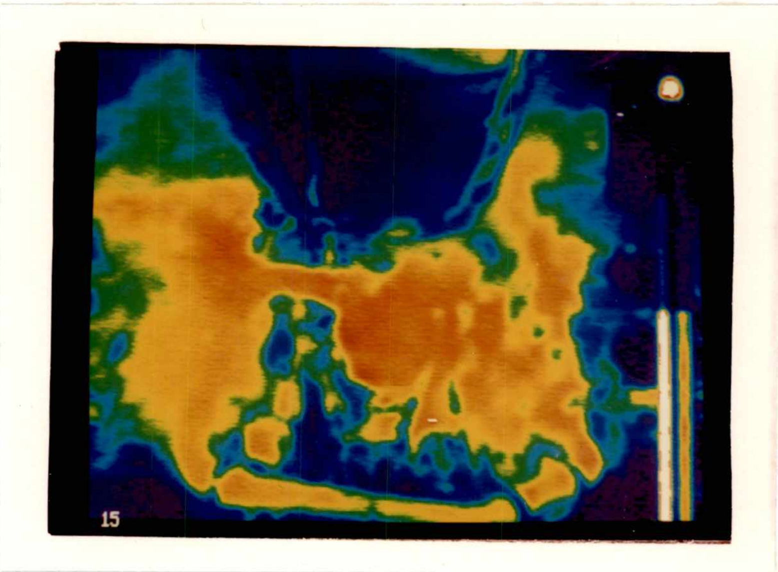


(cc)

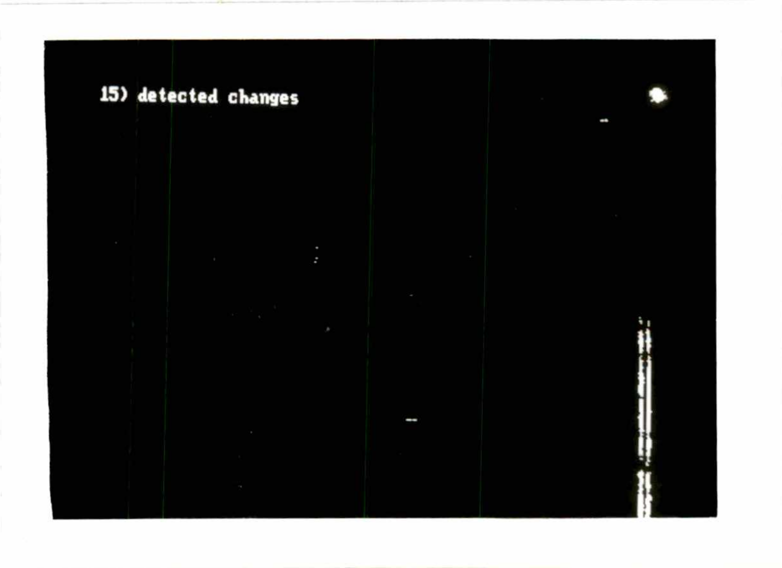


(dd)

Fig. 5. (cont.)

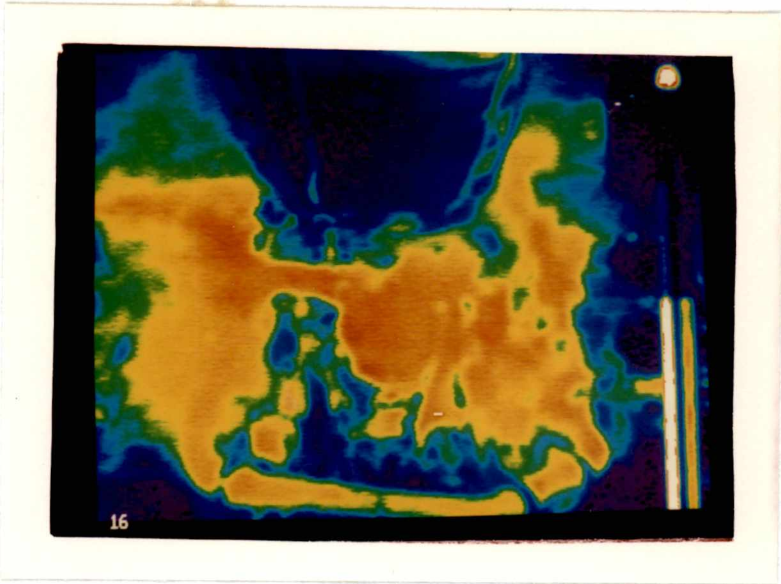


(ee)

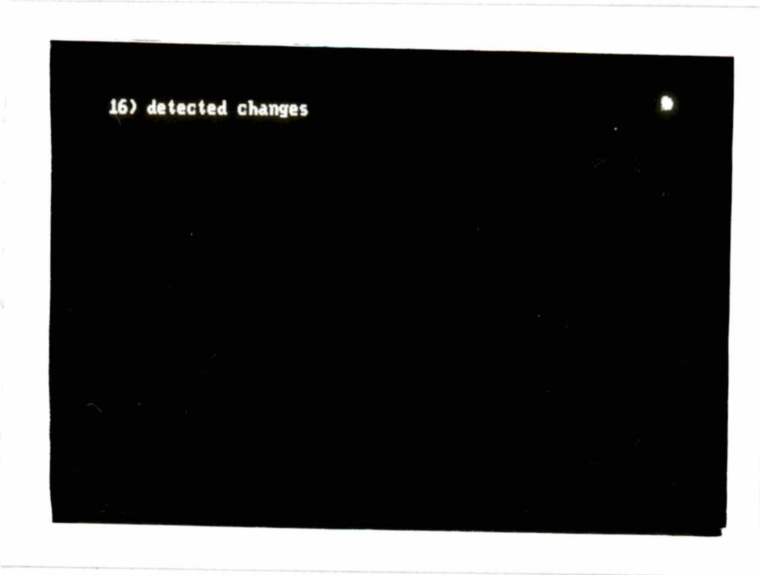


(ff)

Fig. 5. (cont.)



(gg)



(hh)

Fig. 5. (cont.)

images were enhanced and shown in pseudo-color for this thesis. In pseudo-color display, regions shaded from dark to light are shown in blue, green, yellow to red, respectively. However, the algorithm was applied to the raw image data as shown in Fig. 4 with no enhancement performed. The processed images were obtained with highpass filter pole value $\beta = 0.795$ and number of terms in the moving average $J = 5$ and the threshold set at 8 gray levels corresponding to the use of (3.2.4) with standard deviation $\sigma = 4.83$. The first photograph, Fig. 5(a), shows an image acquired prior to the leak, thus no changes are detected in the region of the leak. Although Fig. 5(a) is the only photograph shown of the scene prior to the leak, several images prior to the leak were processed but are not shown here since they appear identical to Fig. 5(a). The remaining images were acquired after the leak was started which is indicated by the neon light being on in the upper right corner of the images. Since the algorithm averages the present output of the highpass filter with four previous outputs, the leak is not immediately detected. Furthermore, examining the enhanced input images reveals that the gas leak does not substantially alter the field of view of the camera until Fig. 5(g) and 5(i) which are the third and fourth input images after the leak was started. This is probably due to warm air in the tubing which must be forced out before the nitrogen gas can escape to condense the moisture in the air of the laboratory. The presence of the leak is indicated in pseudo-color by the change from green to yellow to red in the left side of the input images. As shown in the next few output images, the entire region of the leak does not immediately appear since the algorithm is designed to detect sustained changes. Instead, the white areas in the output indicating change grow from one output image to the next until reaching a maximum in Fig. 5(s). From Fig. 5(u) to the remainder of the output images, the area of detected changes decreases since at this point all the images in the average contain the leak. Thus, the algorithm accepts the leak as a normal part of the scene. However, the changes do not immediately disappear due to the transient response of the filters.

Several observations were made by running the leak detection algorithm with different parameters than those in the example above. First, β was varied and the threshold was held constant. Lowering β increased the cut-off frequency of the highpass filter and reduced the number of terms in the moving average. Thus, for low values of β, more noise effects were observed than in the above example since fewer than five images were averaged by the moving average. Furthermore, the region of change was detected earlier and was larger. It also faded away more quickly than the leak in the example. Using a substantially larger value of β allowed more changes to pass through the highpass filter since the cut-off frequency was lowered. A larger value of β also increased the number of terms in the moving average. Although the large number of images in the average reduced the effects of noise, often very little of the leak was detected since the minimum detectable step amplitude as given by (3.1.11) also increased. Thus, the best choice for β involved balancing the performance of the highpass filter with the number of terms in the moving average.

Next, β was held constant while the threshold was varied. As expected, decreasing the threshold allowed more change to be detected, however, the detector was also more sensitive to noise. Thus, increasing the threshold value allowed less noise through. Larger thresholds also caused the leak to be detected several images later and decreased the size of the detected area.

5.3 Summary

This study showed that the algorithm was effective in detecting leaks in a sequence of images. As the sequence of output images in Fig. 5 demonstrate, the detected region will grow spatially from image to image until reaching a maximum. The region will then fade away. The detected region will be largest in roughly the J^{th} output image after the leak was started, and a number of images is required for the region to fade away. The transient behavior of the highpass filter is the cause of the gradual growth and decay of the detected region. Likewise, this growth and decay does not happen in exactly J images since the input images are from a dynamic system.

The performance of the detector was controlled by β, the pole value (and hence the cut-off frequency) of the highpass filter, by J, the number of terms in the moving average filter, and by T, the value of the threshold. Thus, these parameters must be chosen carefully. Although increasing β allows changes to pass through the highpass filter with less distortion, it also increases the number of terms in the moving average. Increasing the number of terms in the average decreased noise effects, however, averaging too many terms caused very little of the leak to be detected. As more terms are averaged, the amplitude of the step must increase to be detected since the minimum detectable step as given by (3.1.11) increases as β and J are increased. The threshold value could however be lowered to increase the amount of detected changes. But, lowering the threshold increases the sensitivity of the detector to noise.

CHAPTER VI

CONCLUSIONS

A method for detecting abrupt, step-like changes in a time sequence of images has been described, implemented and tested. The detection algorithm presented functions by highpass filtering the input data to remove the quiescent background intensity and comparing the output of a moving average to a threshold. When the average exceeds the threshold, a step has been detected. This method is computationally efficient, is applicable to implementation on digital image processors using fixed-point or integer arithmetic, and is potentially capable of being implemented in real time.

As was shown in the numerical simulation, the algorithm can detect changes in the presence of noise when the threshold is properly chosen. Furthermore, the threshold and the cut-off frequency factor of the highpass filter determine the minimum detectable step amplitude; thus, they must be carefully chosen. It was shown that the performance of the detector is very poor when the standard deviation of the actual input noise is twice as large as the standard deviation used to set the threshold.

The experimental study demonstrates the ability of the algorithm to detect changes in a full image frame using image processing hardware. The experiments performed also demonstrate the effects of the various parameters on the detection process. The highpass filter pole value β can be chosen to remove the background intensity while preserving any changes that occur. However, if it is chosen too large, the number of terms in the moving average will also be large. As the number of terms in the average increase, the effects of noise are reduced, however, less of the leak is detected. Proper choice of the threshold value will allow most of the changes to be detected while decreasing the amount of detected noise.

Further numerical studies could be conducted to determine the effect of different numbers of terms in the moving average on the probability of detection versus step size. For different levels of input noise, optimum results could be obtained by changing J as well as the threshold. Also, improved performance of the algorithm at the cost of increased computation could be achieved by using a higher-order highpass filter for removing the background based upon its spectral properties, if known. This would require re-evaluation of the optimal value for the number of terms in the moving average, or replacement of the recursive filter with a finite impulse response filter. Since the number of terms in the moving average and the threshold value affect the amount of noise detected, a method for estimating the amount of noise in the input is needed. A Kalman filter could be added to the detection system and used to estimate the amount of noise on the input signal before the leak is started. Also, atmospheric changes in the observed scene could cause false detections. Thus, the addition of an adaptive threshold could prevent these false detections while still detecting leaks. Another area of consideration involves incorporating spatial processing with the temporal processing described here. Also, the experiments conducted here need to be repeated using an infrared camera with simulated atmospheric conditions. Finally, the detection algorithm should be implemented in real-time since that is the ultimate goal.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] R. A. Weismiller, S. J. Kristof, D. K. Scholz, P. E. Anuta, and S. A. Momin, "Change detection in coastal zone environments," *Photogrammetric Engineering and Remote Sensing*, vol. 43, no.12, pp. 5133-1539, Dec. 1977.

[2] A. Singh, "Digital change detection techniques using remotely-sensed data," *Int. J. Remote Sensing*, vol. 10, no. 6, pp. 989-1003, June 1989.

[3] S. D. De Gloria, S. J. Daus, N. Tosta, and K. Bonner, "Utilization of high altitude photography and LandSat-1 data for change detection and sensitive area analysis," *Proc. Tenth Inter. Symp. Remote Sensing Environment*, Ann Arbor, MI, vol. 1, pp. 359-368, Oct. 1975.

[4] S. H. Paine and G. D. Lodwick, "Edge detection and processing of remotely sensed digital images," *Twenty-First International Symposium on Remote Sensing of Environment*, Ann Arbor, MI, Oct. 26-30, 1987.

[5] J. R. Jensen and D. L. Toll, "Urban change detection procedures using LandSat digital data," *Proc. of Seventh Pecora Symp. Amer. Soc. Photogrammetry*, pp. 230-251, 1982.

[6] I. S. Reed, R. M. Gagliardi, and L. B. Stotts, "A recursive moving-target-indication algorithm for optical image sequences," *IEEE Transactions on Aerospace and Electronic Systems*, vol.26, no. 3, pp. 434-439, May 1990.

[7] Z. Mao and R. N. Strickland, "Image sequence processing for target estimation in forward-looking infrared imagery," *Optical Engineering*, vol. 27, no.7, pp.541-549, July 1988,

[8] M. Kabuka, J. Desoto, and J. Miranda, "Robot vision tracking system," *IEEE Trans. Indust. Elect.*, vol. 35, no.1, Feb.1988.

[9] A. Shohadaee, "Leak detection feasibility investigation using infrared radiation transfer in absorbing, emitting and scattering media," Doctoral Dissertation, Dept. of Mech. Engr., The Univ. of Tenn., Knoxville, TN, 1990.

[10] T. J. Patterson, D. M. Chabris, and R. W. Christiansen, "Detection algorithms for image sequence analysis," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, no. 9, pp. 1454-1458, Sept. 1989.

[11] M. Unser and M. Eden, "Weighted averaging of a set of noisy images for maximum signal-to-noise ratio," *IEEE Trans. on Acoust., Speech, Signal*

Processing, vol. 38, no. 5, pp. 890-895, May 1990.

[12] J. J. Reis, G. S. Robinson, and A. B. Lucero,"Real-time edge detection in noisy imagery," *SPIE Image Understanding Systems Industrial Appl.*, vol. 155, pp. 23-30, 1978.

[13] M. Basseville, B. Espiau, and J.Gasnier, "Edge detection using sequential methods for change in level–Part I: A sequential edge detection algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, no. 1, pp. 24-31, Feb. 1981.

[14] M. Basseville, "Edge detection using sequential methods for change in level–Part II: Sequential detection of change in mean," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, no. 1, pp. 32-50, Feb. 1981.

[15] E. C. Hildreth, "Edge detection for computer vision system," *Mechanical Engineering*, pp. 48-53, Aug. 1982.

[16] J. A. Malone and L. M. Smith, "A system for sequential step detection with application to video image processing," Submitted to *IEEE Trans. on Industrial Electronics*, June 1991.

[17] S. Y. Hwang, "Dynamic range constraint in state-space digital filtering," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, no. 6, pp. 591-593, Dec. 1975.

APPENDIX

APPENDIX

```
*************************************************************
* This program generates a step response sequence and      *
* processes it using the step detection algorithm.         *
* of the step in noise.                                    *
*************************************************************
* Input values (read from external file):                  *
*              beta: cutoff frequency factor               *
*              sig : variance of the input noise           *
*              Bg  : offset of the step input              *
*              J   : number of terms in moving average     *
*************************************************************
* Output is the probability that the step was detected     *
*       correctly.                                         *
**************************************************************

        integer t0,s,iseed,j,k,true,p,total,step,Bg
        real w,x,a,b,c,d,pi,sig,u1,u2,v,beta
        real T,y,y0,y1,y2,y3,y4

        read(5,*)beta,sig,J,Bg,k

        T=3.70*20.0/sqrt(float(J))

         pi=4*atan(1.0)
         iseed=12345
         a=beta
         b=1.0-beta
         c=-0.5*(beta+1.0)
         d=(beta+1)/2.0


c   increase the step size

         do 40 step=1,255-Bg
            true=0
            t0=0

c   for each step size calculate 100 sequences
c    each with a different time of occurrance

         do 20 i=1,100
              t0=t0+4
              if(t0.gt.255)t0=t0-255
              y=0.0
              y0=0.0
              y1=0.0
              y2=0.0
              y3=0.0
              y4=0.0
```

```
c  sequence  (k denotes the sign of the leak)

           do 10 h=1,255
             n=h-1
             if(n.lt.t0)s=Bg
             if(n.ge.t0)s=Bg+k*step
             u1=ranum(iseed)
             u2=ranum(iseed)
             xp=-2.0*alog(u1)
             w=s + sig*sqrt(xp)*cos(2*pi*u2)
               if(n.eq.0)v=-d*w/c
            x=c*v + d*w
            v=a*v + b*w
              y4=y3
              y3=y2
              y2=y1
              y1=y0
              y0=x
              y=(y0+y1+y2+y3+y4)/J
              if(y.ge.T)go to 99
10        continue

c   leak detected correctly when sum > threshold
c      and n >= time of occurrance

99       if(y.ge.T.and.n.ge.t0)true=true + 1
20       continue

c   calculate probability of correct detection of leak

         prob=float(true)/100.0
         write(6,*)step,prob
40       continue
          stop
          end


          function ranum(n)
          n=1907*n+2003
          n=iand(n,524287)
          ranum=1.9073486328E-06*float(n)
          return
          end
```

VITA

Jo Anne Malone was born on March 14, 1967. She attended school in Coffee County, Tennessee and graduated from Manchester Central High School in June, 1985. From September, 1985 to August, 1989, she attended Tennessee Technological University in Cookeville, TN and obtained a Bachelor of Science degree in Electrical Engineering. In January 1990, she enrolled in The University of Tennessee Space Institute in Tullahoma, TN and was granted a graduate research assistanceship. She was awarded the Master of Science degree in Electrical Engineering in August, 1991.

Currently, she resides in Murfreesboro, TN with her husband and three cats. She is a member of the Institute of Electronics and Electrical Engineers and Eta Kappa Nu.