FINAL REPORT

for grant titled:

SSME Propellant Path Leak Detection Real-Time

NAG8-140

RESEARCH SPONSORED by:

George C. Marshall Space Flight Center
Marshall Space Flight Center, AL 35812

NASA Technical Officer: William T. Powers, Code EB22

Principal Investigators:

Dr. R. A. Crawford and Dr. L. M. Smith

University of Tennessee
Center for Space Transportation and Applied Research
UTSI Research Park
Tullahoma, TN 37388-8897

Period: November 20, 1989 through July 31, 1994

Final Technical Report for NAG8-140
SSME Propellant Path Leak Detection Real-Time
PIs: R. A. Crawford and L. M. Smith

The following four documents outline the technical aspects of the research performed on NASA Grant NAG8-140 and, while separate, should provide a complete description of the work performed during the active period November 20, 1989 to July 31, 1994. The corresponding period of work effort and references for each paper are as follows:

1. November 10, 1989 to June 30, 1991:
   J. A. Malone and L. M. Smith, "A System for Sequential Step Detection with Application to Video Image Processing," *IEEE Transactions on Industrial Electronics*, vol. 39, pp. 277-284, published August 1992.

2. July 1, 1991 to June 30, 1992:
   J. A. Malone, L. M. Smith and R. A. Crawford, "Leak Detection from the SSME Using Sequential Image Processing," *Proceedings of the Advanced Earth-to-Orbit Propulsion Technology Conference 1992*, NASA CP-3174, pp. 180-189, published June 1992.

3. July 1, 1992 to July 31, 1992 (and prior effort):
   L. M. Smith and B. W. Bomar, "Digital Image Processor Specifications for Real-Time SSME Leak Detection," Contract Report, published July 1992.

4. July 31, 1992 to May 31, 1994:
   W. A. Hunt and L. M. Smith, "A Color Change Detection System for Video Signals with Applications to Spectral Analysis of Rocket Engine Plumes," *Proceedings of the Advanced Earth-to-Orbit Propulsion Technology Conference 1994*, to be published.

Further information regarding peripheral aspects of this work can be found in the following student theses and dissertations (not included):

1. A. A. Shohadaee, "Leak Detection Feasibility Investigation Using Infrared Radiation Transfer in Absorbing, Emitting and Scattering Media," Doctoral Dissertation, Department of Mechanical Engineering, The University of Tennessee, Knoxville, TN, 1990.

2. J. A. Malone, "A System for Leak Detection Using Sequential Image Processing," Master's Thesis, Department of Electrical Engineering, The University of Tennessee, Knoxville, TN, 1991.

2. W. A. Hunt, "A System for the Detection of Color Changes Using Sequential Image Processing," Master's Thesis, Department of Electrical Engineering, The University of Tennessee, Knoxville, TN, 1994.

A System for Sequential Step Detection with Application to Video Image Processing

Jo Anne Malone, *Member, IEEE*, and L. Montgomery Smith, *Member, IEEE*

Abstract—**A method for detecting the occurrence of an abrupt steplike change in a time sequence of video images is presented. A single-pole recursive high-pass filter cascaded with a moving average filter processes the input data to remove the quiescent background level and accumulate a sustained change in amplitude. The absolute value of the output is compared to a threshold to decide whether a steplike change in signal amplitude has occurred. It is shown that, for a given cutoff frequency of the high-pass filter, an optimal value exists for the number of terms in the moving average. Considerations for implementation of the algorithm on practical image processors are discussed. The results of numerical and laboratory experiments are presented that verify the effectiveness of the method.**

I. Introduction

In dynamic monitoring of a sequence of images, the detection of a sudden but sustained change is often a desired objective. The application giving rise to the method presented in this paper is the detection of high-pressure gas leaks using infrared and visible imaging. However, the same objective applies to any situation in which the intensity at one spatial location in a time sequence of images abruptly changes from its quiescent value to something different. Thus, a motivation exists for developing a system that can quickly and automatically detect a steplike change in intensity occurring within a given field of view of an image. Because of the likely possibility of extraneous intensity fluctuations, the system should also be robust with respect to the noise corrupting the signal. The problem is that of detecting a step function in the presence of noise.

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 processing are directly applicable to this analysis. Earlier research in digital signal processing has considered similar problems to that addressed in this study, but with marked differences in approach and applicability. This previous work roughly falls into two categories: multidimensional methods involving sequential image processing and one-dimensional methods.

Manuscript received July 9, 1991; revised February 15, 1992. This work was supported by the National Aeronautics and Space Administration and the Center for Space Transportation and Applied Research under grant NAG8-140.

The authors are with the Center for Laser Applications, University of Tennessee Space Institute, Tullahoma, TN 37388-8897.

IEEE Log Number 9201270.

Sequential image processing methods for detecting changes in dynamic scene analysis that have been previously implemented are predominately used for monitoring land resources. Although these methods are effective for the purposes for which they were designed, they are not particularly applicable to detecting steplike changes. For example, many land-use techniques employ various forms of temporal differencing [1]–[5]. Other algorithms use thresholding [3] 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 [3]. Since these schemes are used to detect changes that occur slowly over time, only the present image and a few previous images are used. The limited number of sample values makes these systems unsuitable for steplike change detection, since the sustained nature of the change is not exploited.

Another somewhat related application is target tracking systems that use image sequences to detect and follow the presence of targets or objects [6]–[8]. The techniques used include spatial differencing combined with temporal differencing, as well as algorithms using matched filters and peak detection [6]. Furthermore, the system in [7] and [8] determines the trajectory of targets using dynamic programming. Although these techniques monitor changes in image sequences, their objective is markedly different than detecting a steplike change in intensity. For this reason, they are not readily extended to the application considered here.

Among one-dimensional analyses, detection of abrupt steplike changes has been studied as an edge detection problem for jumps occurring spatially within an image [9]–[11]. A noteworthy approach taken by Basseville, Espiau, and Gasnier, [10], [11], 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 and is detected using Hinckley's algorithm. Hinckley's algorithm, as described by Basseville [11], computes the cumulative sum of the sequence and the maximum of the sum to detect positive jumps. A second detector is required for negative jumps. Despite their effectiveness, these spatial processing methods are inherently noncausal. Therefore, they also are not readily extendable to sequential image processing.

In addition to the edge-detection schemes, one-dimensional algorithms have been developed for detecting abrupt changes in discrete-time sequences such as speech,

0278-0046/92$03.00

electrocardiogram, and geophysical signals [12]–[15]. However, these techniques are usually tailored to detect changes in the overall spectral characteristics of the signal and not a specific signal artifact, such as a step function.

The method presented in this paper is an efficient causal algorithm for detecting an abrupt steplike change in intensity in sequential video images. A recursive digital high-pass filter is used to remove slowly varying quiescent intensity levels without the need to compute a mean value estimate. This filter is cascaded with a moving average filter that accumulates a sustained change in amplitude. The number of terms in the moving average is chosen to maximize the signal-to-noise ratio for a given high-pass filter cutoff frequency. The absolute value of the output of the moving average is compared with a threshold to detect the occurrence of either positive or negative jumps. The threshold used in this method is a function of the input noise variance and the number of terms in the moving average and provides explicit control over the probability of false detections. This algorithm has been implemented both in floating-point arithmetic on a digital computer for simulation studies and in fixed-point arithmetic on a digital image processor for practical application to video data. It has been found to be computationally efficient and effective for detecting sudden but sustained changes. Although implementation to date has not realized real-time processing, currently available image processing hardware utilizing full-frame arithmetic logic units (ALU's) should allow video signal processing at standard framing rates.

Section II describes the step detection algorithm. The optimal choices for the parameters of the high-pass filter and moving average will be derived. Some of the considerations involved in the practical implementation of the algorithm on digital image processors are discussed in Section III. The algorithm has been tested in a numerical study using simulated data and has been implemented on an image processing system. The procedures and results of these tests will be presented in Section IV. Summary and conclusions are given in Section V.

II. The Step Detection Algorithm

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 real-time processing of video data.
2) It should be impervious both 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 high-pass filter to remove the slowly or nonvarying background intensity level while allowing any sudden changes to be passed. The output of the high-pass 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 detail. To simplify notation, all signals are written simply as functions of the time variable index n. For video images, all signals depend on two spatial position coordinates in addition to the time variable. However, because processing is carried out only with respect to the time index, the spatial dependence is suppressed in the notation.

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

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

This expression was obtained by applying the bilinear transformation [16, pp. 608–612] to a single-pole continuous transfer function of the form $s/(s+\Omega_c)$. The discrete cutoff frequency of the filter is set by choice of the pole value β. The relationship between β and the -3 db normalized discrete cutoff frequency ω_c is found by making the substitution $z = e^{j\omega}$ in (1) and solving for the value of ω at which the magnitude of the transfer function $|H(e^{j\omega})|$ equals $1/\sqrt{2}$. Explicitly, these parameters are related by

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

Note that the use of a recursive or infinite impulse response (IIR) filter utilizes all of the past information in the input signal.

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

$$y(n) = \frac{1}{J}\sum_{k=0}^{J-1} x(n-k). \tag{3}$$

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 will be shown that for a given high-pass filter cutoff frequency ω_c with corresponding pole value β, an optimal choice exists for the number of terms J in the moving average. To derive this, an input signal of the form

$$w(n) = b(n) + Au(n) + \epsilon(n) \tag{4}$$

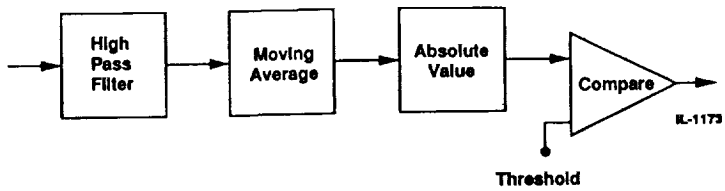


Fig. 1. Block diagram of the step detection system.

is assumed where $b(n)$ represents a slowly varying background intensity level, $u(n)$ is a unit step function, A the amplitude of the step, and $\epsilon(n)$ is a zero-mean white random sequence representing the noise. If it is assumed that the high-pass filter removes the background completely, then from the transfer function given in (1), its output can be shown to be

$$x(n) = \tfrac{1}{2}A(1+\beta)\beta^n u(n) + h(n) * \epsilon(n) \qquad (5)$$

where $h(n)$ is the impulse response of the filter in (1) and the asterisk ($*$) denotes convolution. In practical applications, the cutoff frequency of the high-pass 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

$$x(n) \simeq \tfrac{1}{2}A(1+\beta)\beta^n u(n) + \epsilon(n). \qquad (6)$$

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

$$y(J-1) = \frac{A}{2J}\left(\frac{1+\beta}{1-\beta}\right)(1-\beta^J) + \frac{1}{J}\sum_{k=0}^{J-1}\epsilon(n-k). \qquad (7)$$

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 (7)) to the standard deviation of the processed noise signal:

$$\frac{S}{N} = \frac{|A|}{2\sigma}\left(\frac{1+\beta}{1-\beta}\right)\left(\frac{1-\beta^J}{\sqrt{J}}\right). \qquad (8)$$

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

$$\beta^J(1 - 2\ln\beta^J) = 1 \qquad (9)$$

which can be solved numerically to yield

$$J = -\frac{1.2564}{\ln\beta}. \qquad (10)$$

In practice, J is chosen to the nearest integer to the value computed in (10). Thus, from a given cutoff frequency chosen to remove the background component from the signal, the high-pass filter pole value β is calculated from (2), and a value for the moving average summation J is chosen from (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 (7) exceeds the threshold value. That is, with no noise corrupting the signal, a step must have an amplitude satisfying

$$|A| \geq \frac{2JT}{1-\beta^J}\left(\frac{1-\beta}{1+\beta}\right) \qquad (11)$$

to be detected. This expression provides a lower limit of detection that is useful in evaluating the statistical performance of the step detection algorithm as in the numerical study presented in Section IV.

III. Implementation Considerations

The system on which this algorithm was implemented for image processing in this study was a i386-based personal computer with a CPU speed of 25 MHz. A DT-2861 frame grabber card and a DT-2858 auxiliary frame processor card manufactured by Data Translation were installed in the unit and used for image data acquisition, processing, and display. Several aspects of the step detection algorithm require special consideration when it is implemented on such practical digital image processing systems. One consideration involves the arithmetic operations realizing the filters. Others are concerned with minimizing the effects of finite precision fixed-point arithmetic in the filtering operations and avoiding false detections due to transients at start-up. Also, the threshold value must be chosen to reduce false detections due to noise while maintaining a suitable level of sensitivity. This section discusses these considerations and presents some methods that have been employed to reduce adverse effects.

Because each sequence value processed in the step detection algorithm actually represents one element of a two-dimensional image array with many elements, arithmetic operations in the filter implementation involve a large amount of computation. However, frame processor ALU's can quickly perform addition of image frames. Furthermore, the point-by-point multiplication of array element values can be carried out by noting that frame

buffers in digital image processors are usually configured for fixed-point or integer pixel values (typically 8 b). Thus, only a finite number of products exist for any multiplication by a constant-valued filter coefficient. Multiplications are thus realized via look-up-tables (LUT's) with precomputed values for each possible product.

By realizing the coefficient multiplications via LUT's, and utilizing the ALU for additions, each add/multiply computation pair requires approximately 300 ns per pixel according to the manufacturer's specifications. This execution time can be compared to that of the host processor by assuming that each add/multiply pair in the sum of products consists of a minimum of two data movements, one integer multiply, one arithmetic shift, one integer add, and one loop instruction. This amounts to approximately 46 clock cycles, which, for a CPU speed of 25 MHz, corresponds to 1.84 μs execution time. Thus, a sixfold increase in computation time is a conservative estimate of the advantages of this technique.

Implementation of the recursive high-pass 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

$$\begin{aligned} v(n+1) &= av(n) + bw(n) \\ x(n) &= cv(n) + dw(n) \end{aligned} \tag{12}$$

where a, b, c and d are constant-valued coefficients chosen to realize the transfer function of (1). Specifically, these coefficients must satisfy

$$H(z) = \frac{bcz^{-1}}{1 - az^{-1}} + d. \tag{13}$$

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

$$v(0) = -\frac{d}{c}w(0) = w(0). \tag{14}$$

Because 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 high-pass 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 further assumed that this random signal is Gaussian distributed, the probability of a false detection (i.e., that the magnitude of the processed noise exceeds the threshold) at each sample instant will be 2×10^{-4} for a threshold value of

$$T = \frac{3.70\sigma}{\sqrt{J}}. \tag{15}$$

Other false detection probabilities can be realized by choosing other constants of proportionality in accordance with tabulated Gaussian probabilities [18]. The value given in (15) has been found to provide acceptable performance for practical applications.

IV. Numerical and Experimental Results

To evaluate the statistical performance of the step detection algorithm, a numerical study was conducted by first implementing the algorithm in a FORTRAN program as follows. Tests were conducted by processing 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 (4) was generated with $b(n)$ an arbitrary constant. The step occurred at some random time t_0, uniformly distributed over the 256-point data block. The additive noise was an uncorrelated Gaussian random sequence generated by [16, pp. 132–133]

$$\epsilon(n) = \sigma\sqrt{-2\ln u_1(n)}\cos[2\pi u_2(n)] \tag{16}$$

where $u_1(n)$ and $u_2(n)$ were random numbers uniformly distributed on (0, 1) generated from an intrinsic function within the program. The state-space filter structure in (12) was used to remove the background intensity and update the state variable for the next input value. The moving average was used next to average the present value of the high-pass 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 $|y(n)| \geq T$ and $n \geq t_0$, the program indicated that the step was correctly detected. Otherwise, if $|y(n)|$ exceeded the threshold while $n < t_0$, or if $|y(n)|$ failed to exceed the threshold at any time during the 256-point data block, the program indicated that the algorithm failed to detect the step correctly. The procedure was repeated for each value of n until either a step was detected or $n > 255$.

To obtain a measure of the performance of the algorithm, the foregoing process was performed many times for step amplitudes ranging from 0 to 155 and varying times of occurrence. The algorithm was tested with 100 different input sequences each having the same step amplitude but different times of occurrence and with a different sequence of additive noise. 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. The number of correct detections for each step size was converted to an esti-

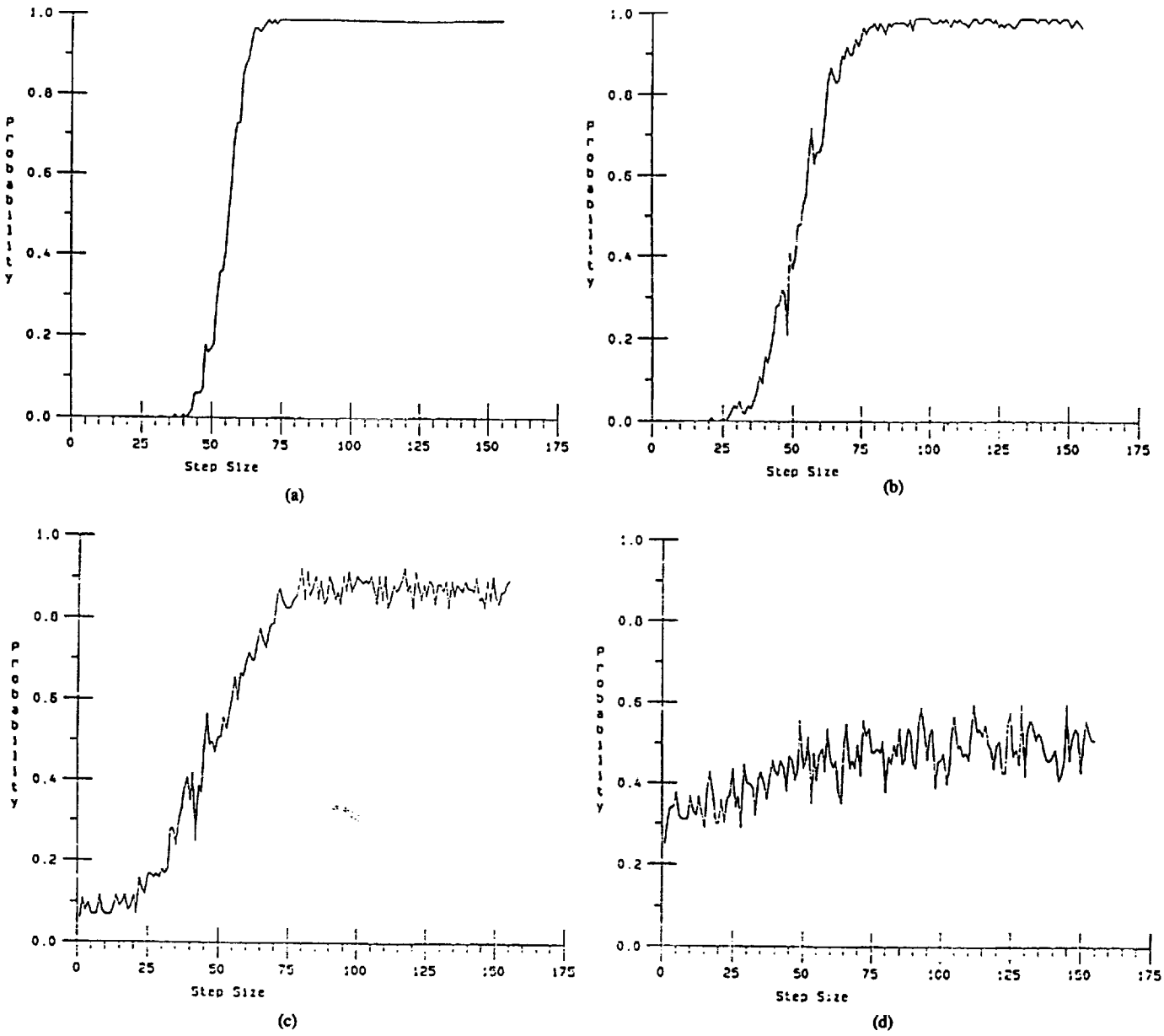


Fig. 2. Results of numerical evaluation of the step detection algorithm. These plots show the calculated probability of correct detection versus step amplitude for processing 256-point input data blocks corrupted by noise with standard deviations of (a) 10, (b) 20, (c) 30, and (d) 40.

mated probability and plotted as a function of the step size. The input noise variance was held constant for a given series of trials.

Fig. 2 shows representative results of performing these tests with $\beta = 0.7951$, $J = 5$, and the threshold fixed at $T = 33.1$. This value of β was chosen somewhat arbitrarily for a cutoff frequency slightly more than 1 Hz in a 30-Hz sampling rate system. The threshold corresponds to using the criterion in (15) with an assumed input noise standard deviation of $\sigma = 20$. Shown are four plots of probability of correct detection versus step amplitude for actual input noise standard deviations of 10, 20, 30, and 40.

This series of plots illustrates the relationship between the standard deviation of the input noise and the assumed standard deviation used to set the threshold. If the level of noise in the input is overestimated, as in plot (a), the detector is very accurate for large step sizes and the minimum step size detected is close to the value 55.4 given by (11) for this example. Plot (b) demonstrates the performance of the algorithm for the threshold criterion described in Section III where the assumed noise standard deviation exactly matches that of the noise actually in the input. As the standard deviation of the input noise is increased, as shown in plots (c) and (d), the algorithm becomes less effective. Although smaller steps are detected more frequently, the increase is due to serendipitous effects of noise causing the processed signal magnitude to exceed the threshold after the occurrence of the step. The noise also causes false detections prior to the occurrence of the step, which results in decreased performance for large step amplitudes. Note that once the

standard deviation of the noise on the input becomes twice as large as the standard deviation used to set the threshold (plot (d)), the probability of detecting any step correctly is effectively that of guesswork. Therefore, a noteworthy property is that although overestimating the standard deviation of the noise (Fig. 2(a)) reduces sensitivity in terms of the minimum detectable step amplitude, the performance of the algorithm is otherwise not severely affected as it is in the case where the standard deviation is underestimated (Figs. 2(c), (d)).

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, parameters of the data that do not affect performance are the background value and the direction (positive or negative) of the step. As shown by the transfer function (1), the high-pass filter assures that any constant background intensity value is removed before processing the data block. The sign of the step is irrelevant since the absolute value of the average is taken before comparing it to the threshold.

An experimental study was performed by implementing and testing the algorithm on the digital image processor using the techniques discussed in Section III. Image data were acquired from RS-170 standard video format signals and stored in 512 × 512 pixel format with 8 b per pixel. The detection program written for this experiment allowed interactive processing of selected input frames as follows. Live video was shown on the display monitor of the image processor until the operator initiated acquisition of one frame of the incoming signal. This frame was input to the detection algorithm and processed. The output of the algorithm was then displayed as a binary image. At any location where a steplike 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, live video was again shown on the display monitor, and the process repeated. For the experiment, a camera was connected to the image processor for acquiring live video of a laboratory scene. A reference test scene was set up and several frames of this image were processed. The scene was subsequently altered and processing continued.

Figs. 3 and 4 are photographs taken during the experiment. For this experiment, the same values $\beta = 0.7951$ and $J = 5$ were used as in the preceding numerical study. Because the noise contributed by the camera and lighting variations was assumed to be low, the threshold was set using (15) for an input noise standard deviation of 5.0 gray levels. (The 8-b input pixel intensity values ranged from 0 to 255.) Fig. 3(a) shows the reference test scene. As expected, no detected changes were indicated in the binary output image during processing of this image. The altered scene is shown in Fig. 3(b). Fig. 4(a)–(h) shows the locations where changes were detected as the sequence of images of the altered scene was processed. Because the scene was altered only once, the input frames for Fig. 4(a)–(h) appear identical to Fig. 3(b) and are not shown.

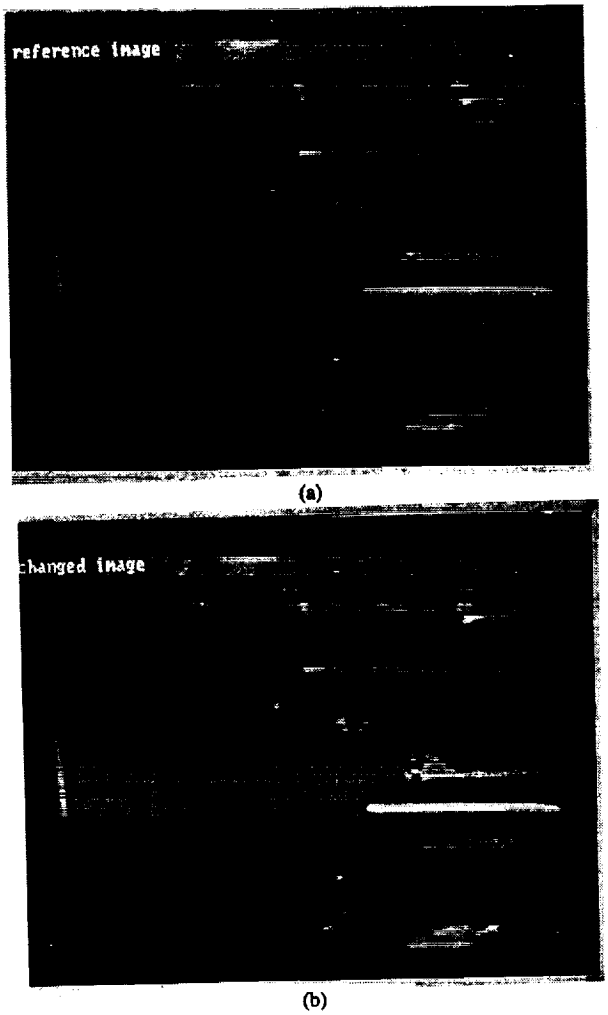


Fig. 3. Photographs of the input video images used in the experimental testing of the step detection algorithm. (a) Reference test scene. (b) Altered scene.

However, it should be noted that real-time video signals including noise were acquired and processed to obtain these images. Because the algorithm is designed to detect a sustained change, the white areas indicating detected change do not immediately appear. Instead, they grow from one output image to the next and reach a maximum in the fifth frame processed after the scene was altered (Fig. 4(e)). They then shrink over Fig. 4(f)–(h) since the altered image itself is not changing and the transient response of the filters is decaying.

Processing time for the algorithm is obviously dependent on the number of pixels in the images and the number of terms in the moving average. For the system used in this study with the parameters as given in the preceding example, execution time was approximately 1.12 s per frame. Although this exceeds the $\frac{1}{30}$-s time required for real-time processing of video signals, the relatively inexpensive and general-purpose nature of the hardware used should be considered. Also, programming was carried out in a higher level language (FORTRAN) utilizing a subroutine library supplied by the manufacturer. It seems reasonable to conclude that real-time implementa-

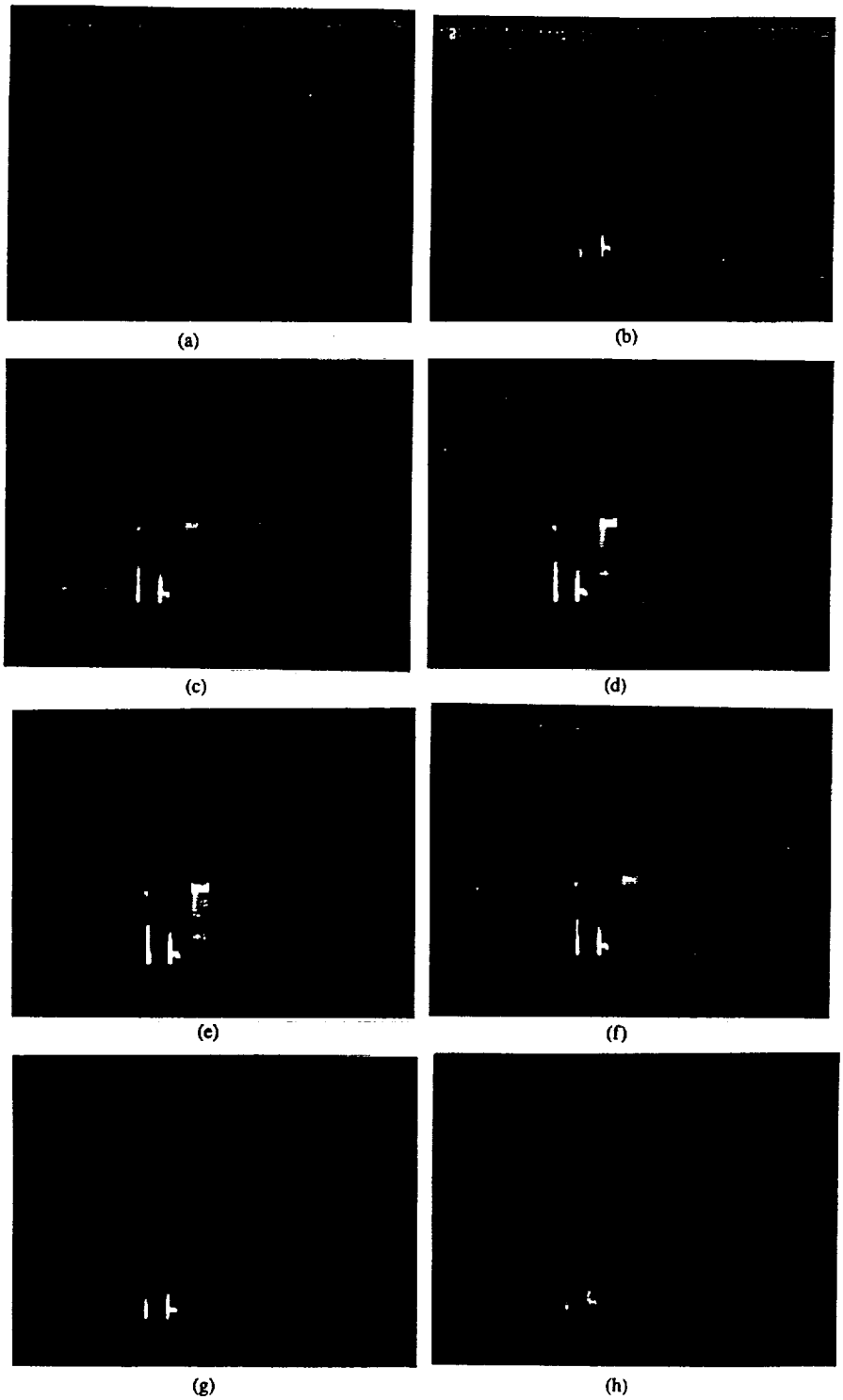


Fig. 4. Sequence of binary output images obtained by processing video images of the altered scene in Fig. 3(b) with the step detection algorithm. White areas represent locations of detected change.

tion of this method is feasible with present technology, provided that more expensive or special-purpose hardware is utilized and programming is carried out in microcode or assembly language.

V. Summary and Conclusions

A method for detecting abrupt, steplike changes in a time sequence of images has been developed, implemented, and tested. The detection algorithm presented functions by 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 and is applicable to implementation on digital image processors using fixed-point or integer arithmetic.

As was shown in the numerical simulation, the algorithm can detect changes in the presence of noise when the threshold is properly chosen. The experimental study demonstrates the ability of the algorithm to detect changes in a full image frame using image-processing hardware. These results indicate that implementation of this scheme for effective real-time video image processing is feasible with existing technology.

Improved performance at the cost of increased computation could be achieved by using a higher order high-pass filter for removing the background based on its spectral properties, if known. This would require reevaluation of the optimal value for the number of terms in the moving average or replacement by a finite impulse response filter. The incorporation of spatial processing of the image data in addition to the temporal processing presented here is of particular interest for many applications. The latter point will be the subject of further investigations where leak detection from infrared video data in real-time is the goal.

REFERENCES

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

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

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

[4] 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," in *Proc. Tenth Int. Symp. Remote Sensing Environment*, Ann Arbor, MI, vol. 1, pp. 359–368, Oct. 1975.

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

[6] 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.

[7] Y. Barniv, "Dynamic programming solution for detecting dim moving targets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-21, pp. 144–156, Jan. 1985.

[8] Y. Barniv and O. Kella, "Dynamic programming solution for detecting dim moving targets Part II: Analysis," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-23, pp. 776–788, Nov. 1987.

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

[10] 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.

[11] 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.

[12] M. Basseville and A. Benveniste, "Sequential detection of abrupt changes in spectral characteristics of digital signals," *IEEE Trans. Inform. Theory*, vol. IT-29, no. 5, pp. 709–724, Sept. 1983.

[13] M. Basseville, A. Benveniste, and G. V. Moustakides, "Detection and diagnosis of abrupt changes in modal characteristics of nonstationary digital signals," *IEEE Trans. Inform. Theory*, vol. IT-32, no. 3, pp. 412–417, May 1986.

[14] M. Basseville and A. Benveniste, "Design and comparative study of some sequential jump detection algorithms for digital signals," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, no. 3, pp. 521–535, June 1983.

[15] A. S. Willsky, "Detection of abrupt changes in dynamic systems," Report No. NASA-CR-173217 NAS, Jan. 1984.

[16] J. G. Proakis and D. G. Manolakis, *Introduction to Digital Signal Processing*. New York: Macmillan, 1988.

[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.

[18] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*. New York: Dover Publications, 1972, pp. 297–329.

Jo Anne Malone (SM'88–M'92) was born in McMinnville, TN, on March 14, 1967. She received the B.S. degree from Tennessee Technological University, Cookeville, TN, in 1989 and the M.S. degree from the University of Tennessee Space Institute, Tullahoma, TN, in 1991, both in electrical engineering.

Her interests include image and signal processing.

L. Montgomery Smith (M'87) received the B.S. degree from Rhodes College, Memphis, TN, in mathematics and physics in 1978, and the B.S., M.S., and Ph.D. degrees from The University of Tennessee, Knoxville, in electrical engineering in 1982, 1984, and 1988, respectively.

He has previously worked for Southern Research Institute, Birmingham, AL, and for the Tennessee Department of Public Health, Nashville, TN. From 1984 to 1989 he was employed as an Engineer for the Center for Laser Applications at the University of Tennessee Space Institute, Tullahoma, TN. Since 1989 he has been an Assistant Professor with the Department of Electrical and Computer Engineering at The University of Tennessee Space Institute. His current activities include teaching and research in the areas of electro-optics, digital signal and image processing, and data analysis.

PREV
ANN
93X 10117

LEAK DETECTION FROM THE SSME USING SEQUENTIAL IMAGE PROCESSING

Jo Anne Malone, L. Montgomery Smith, and Roger A. Crawford *
Center for Laser Applications
The University of Tennessee Space Institute
Tullahoma, TN 37388-8897

ABSTRACT

Initial research using theoretical radiation transport models established that the occurrence of a leak is accompanied by a sudden but sustained change in intensity in a given region of an image. In this study, temporal processing of video images on a frame-by-frame basis was used to detect leaks within a given field of view. The leak detection algorithm developed in this study consists of a digital highpass filter cascaded with a moving average filter. The absolute value of the resulting discrete sequence is then taken and compared to a threshold value to produce the binary leak/no leak decision at each point in the image. Alternatively, averaging over the full frame of the output image produces a single time-varying mean value estimate that is indicative of the intensity and extent of a leak. Laboratory experiments were conducted in which artificially created leaks on a simulated SSME background were produced and recorded from a visible wavelength video camera. This data was processed frame-by-frame over the time interval of interest using an image processor implementation of the leak detection algorithm. In addition, four video sequences from an actual SSME test firing were analyzed using this technique. A hydrogen gas leak was detected before existing sensors initiated shutdown during the test. The resulting output image sequences and plots of the full frame mean value versus time verify the effectiveness of the system.

INTRODUCTION

The rapid detection of propellant leaks from the Space Shuttle Main Engine (SSME) during test firing is crucial to the prevention of catastrophic failures. Ruptures of high-pressure lines and internal components due to thermal shock, mechanical stress, erosion, and material fatigue often result in failure modes with sufficiently long time constants to allow detection and safe shut down. Recent advances in imaging and image processing technology provide the hardware necessary for visual and infrared observation of these phenomena and the computing capability required for processing the signals and detecting the occurrence of a leak within the field of view. Thus, a system capable of detecting leaks from images acquired sequentially during test firing in real time is of value to the development of the SSME and is realizable with current technology. This study investigated this approach and established its feasibility and applicability to the program.

Previous work in this area by Shohadaee and Crawford[1,2] concentrated on establishing the feasibility of observing hot or cold leaks using infrared imaging. The selection of candidate detection methods required the development of analytic models for the leak plumes and the radiation transport through the leak plume. The theoretical models developed were used to predict radiation transport in absorbing, emitting and scattering media. These models predicted the intensity of both background and plume radiation reaching a sensor location, and they were used for designing validation experiments. The feasibility of infrared detection of leak plumes was demonstrated on subscale simulated plumes to determine sensitivity, signal-to-noise ratio, and general suitability. Both hot and cold leaks were readily detected as measurable intensity changes by the sensor.

However, to detect the occurrence of a leak, the temporal aspects of the process must be considered. The previous analysis showed that the occurrence of a leak should result in a sudden change in intensity in a given region of an image. Furthermore, the change should be sustained for a typical persistent leak. The time variation of the intensity at a point within the area of the leak should therefore be similar to that of a step function, although other smaller intensity variations are also present due to normal operating

* Work supported by NAG8-140.

conditions. The problem becomes that of detecting a step function in the presence of additive noise.

The leak detection system presented here was designed to quickly and automatically detect a step-like change in intensity in a sequence of images.[3,4] Temporal processing is carried out at each point in full-frame digitized video data. The system consists of a causal, recursive high-pass filter that removes slowly-varying background intensities cascaded with a moving average filter that accumulates transmitted sustained changes. The absolute value of the output is compared to a threshold to provide the binary leak/no leak decision, or the absolute value over the full output frame is averaged to produce a time-varying mean value indicative of the level and spatial extent of any leak. It can be efficiently implemented on standard digital image processors and applied to full-frame video data, with output data each frame. Although it has not been implemented in real time as of this writing, execution times for the off-line processing carried out thus far indicate that such processing is achievable with present technology.

The leak detection system has been successfully applied to laboratory experimental data and actual test-stand firing data. Laboratory experiments using visible wavelength video signals were conducted for an actual controlled gas leak with data processed off-line on a digital image processor. The detection system was found capable of detecting such leaks with proper choice of the operating parameters. In addition, four video image sequences of an SSME test-stand test firing were analyzed with this technique to evaluate its performance under actual firing conditions. A hydrogen leak was detected in three of the four camera views approximately 4 sec before existing sensors initiated an early shutdown.

The following section describes the leak detection algorithm and discusses some of the considerations for implementing it on digital image processing equipment. Section III discusses the laboratory experiments and their results. Section IV describes the processing of the four video sequences from an SSME test firing. A summary and conclusions are given in Section V.

THE LEAK DETECTION ALGORITHM

With the background to this discussed in the previous section, it can be seen that the requirements for an imaging system for SSME leak detection are:

a. it must be causal, since future data are unknown in a real time system,
b. it must be computationally efficient due to the large volume of data acquired in standard video imaging systems,
c. it must be capable of detecting positive and negative changes in intensity corresponding to hot and cold leaks, and
d. it must be robust with respect to background intensity levels and random noise.

Time-varying intensity levels in video images are an inherently discrete signal with a sampling rate of 30 Hz for standard television format. Therefore, techniques of digital signal processing are directly applicable to the analysis. A detailed derivation of the step detection algorithm used in this study was given in [3] along with a numerical analysis of its performance. For completeness, a brief description of it is presented here.

DESCRIPTION

Video images are dependent upon two spatial coordinates in addition to time. However, in this scheme, identical processing is carried out at each point in the two-dimensional image arrays, so the system was developed as a one-dimensional discrete-time algorithm, and is presented here in that manner. To simplify notation, signals are written as depending only upon the integer-valued time index n. The explicit dependence upon spatial coordinates is suppressed.

The leak detection system consists of a series of cascade processing blocks. The first unit in the system is a highpass filter that removes constant and slowly-varying background intensity levels. For computational efficiency and rapid response, the filter was chosen to be a single-pole recursive digital filter with z-transform transfer function given by

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

The cutoff frequency is set by choice of the parameter β. The second block is a moving average filter used to accumulate a sustained change when it occurs. With the output of the highpass filter denoted $x(n)$, the output of the moving average is computed from

$$y(n) = \frac{1}{J}\sum_{k=0}^{J-1} x(n-k), \tag{2}$$

where J is the number of points in the moving average. The third block in the system takes the absolute value of the output of the moving average, thus enabling the system to detect positive or negative intensity changes.

The output of the absolute value block can be processed in two ways. First, at each spatial position within the output image, it can be compared to a preset threshold T. If the value exceeds the threshold, the decision is made that a leak is detected at that point, otherwise no leak is detected. Second, the values in the output image can be averaged over the full field of view to produce an overall mean value estimate. This mean value serves as a single time-varying quantity that is indicative of both the amplitude and extent of the intensity change resulting from a leak. Both methods have proven useful in applying this method.

It was shown[3] that for a given cutoff frequency, or equivalently, filter parameter β, an optimal value exists for the number of terms in the moving average J. This was shown by assuming an input signal of the form

$$w(n) = b(n) + Au(n) + \epsilon(n), \tag{3}$$

where $b(n)$ represents a slowly-varying background intensity, $u(n)$ is a unit step, A the step amplitude, and $\epsilon(n)$ is a zero-mean random sequence modeling the noise corrupting the signal. After processing, the ratio of the peak signal amplitude, which occurs $J-1$ samples later, to the standard deviation of the noise σ was shown to be

$$\frac{S}{N} = \frac{|A|}{2\sigma}\left(\frac{1+\beta}{1-\beta}\right)\left(\frac{1-\beta^J}{\sqrt{J}}\right), \tag{4}$$

which is maximized for a value of J given by

$$J_{\text{opt}} = -\frac{1.2564}{\ln\beta}. \tag{5}$$

In practice, J is chosen to be the closest integer to the value given in (5).

IMPLEMENTATION

Implementation of this algorithm on a digital image processor required special considerations arising from the large amount of computation for two-dimensional image data arrays, and the fixed-point or integer format of the individual array element (pixel) values. Computation times for arithmetic operations were reduced in the following manner. Multiplications were realized by precomputing all possible products for the constant-valued filter coefficients and using look-up-tables (LUTs). Additions were carried out with the full-frame arithmetic logic unit (ALU) available as part of the image processing hardware.

The recursive highpass filter was realized with minimum storage requirements by using a state-space structure L_∞-norm scaled to prevent any overflow in the fixed-point computations. The output $x(n)$ was calculated from the input $w(n)$ and a state variable $v(n)$ by means of the following two equations

$$\begin{aligned} v(n+1) &= \beta v(n) + (1-\beta)w(n) \\ x(n) &= -\frac{1+\beta}{2}v(n) + \frac{1+\beta}{2}w(n). \end{aligned} \tag{6}$$

Note that for a positive bounded input and stable filter ($|\beta| < 1$), the magnitude of the state variable will not exceed the input bound.

For processing where the overall system output is compared to a threshold to make the leak/no-leak decision, it was found that acceptable performance for practical applications was achieved by choosing the threshold to be

$$T = \frac{3.70\sigma}{\sqrt{J}}, \tag{8}$$

where σ and J are defined as before. This choice produces a probability of false alarm of 2×10^{-4} at each sample instant for Gaussian distributed input noise. Other false detection probabilities can be realized by choosing other constants of proportionality in accordance with tabulated Gaussian probabilities.

In the case where the mean value of the overall output image was found, the computation was carried out in image processing hardware designed to sum the pixel values over a full frame. The mean value was found by dividing by the number of pixels in the field of view. This is a standard feature on most image processors and helps in reducing computation time.

LABORATORY EXPERIMENTAL ANALYSIS

To test the ability of the system to detect changes in an actual image sequence, a series of experiments was conducted. In these experiments, an actual gas leak was generated in the laboratory and video image data were acquired. These data were then processed frame-by-frame after acquisition was completed but in a causal, sequential manner to evaluate the performance of the system under actual real-time processing conditions.

For the experiment, an NTSC format, solid-state, visible wavelength camera was used to acquire live video of a gas leak set-up in the laboratory. The experimental set-up is shown in Fig. 1. 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. 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. The leak was initiated manually by throwing the switch on a multiple outlet power strip. This electronically opened a solenoid valve placed on one end of the copper tubing. As the liquid nitrogen warmed, nitrogen gas was forced up the tubing, thus producing the leak.

A neon light connected to the power strip was included in the upper right-hand corner of the scene with the leak and was turned on at the same time as the solenoid valve. This provided a means for determining the image frame in which the leak was initiated. A picture of the SSME 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 it from behind. The leak was made visible by the forward-scattering of the light from the condensed water vapor droplets formed by the escaping cold nitrogen gas. The field of view of the camera was approximately 40 cm by 40 cm.

For each experimental run, a brief (1 to 3 sec) image sequence of this scene was acquired and stored on an optical disc video recorder/player. The use of an optical disc video unit was required in this study in order to provide the single-frame jitter-free playback of images not possible with magnetic tape units. 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.

Several 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.

The detection algorithm was implemented on a digital image processor using the techniques described in the previous section. The image processor used for this experiment was a system with a i386-based host computer operating at a CPU speed of 25 Mhz into which a Data Translation DT-2861 frame grabber and DT-2858 auxiliary frame processor had been installed. This system acquires and stores image data in 512 × 512 pixel format with 8 bits per pixel. The detection program written for these experiments processed the sequence of images which was previously acquired and stored on the optical disc recorder. For convenience, the optical disc recorder was interfaced to the image processor with an RS-232 connection so that individual frame advancing was performed automatically with execution of the detection program.

Execution of the program proceeded by displaying the input image frame on the display monitor. This frame was digitized, 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. At the same time, the mean value of the output image was computed and written to a data file. Once processing of each input frame was completed, the laser disc video recorder was advanced one frame, that image shown on the display monitor, and the process repeated.

The photographs in Fig. 2 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. Figure 2(a) was taken immediately prior to the leak while Figs. 2(b) through 2(f) show the first five images acquired after the leak was started. (Note that the neon light is on in Figs. 2(b)-2(f) but not in Fig. 2(a).) The leak appears as a light cloud stretching from just below the middle of the left-hand side of the image to the lower right-hand corner.

Figure 3 contains the output thresholded images corresponding to the input images in Fig. 2. The processed images were obtained with highpass filter parameter $\beta = 0.795$, number of terms in the moving average $J = 5$, and the threshold set at 12 gray levels corresponding to the use of Eq. (8) with standard deviation $\sigma = 7.25$. The white areas in the output indicating change initially grow as seen in Figs. 3(b) through 3(f). Later on, in images not shown here, they shrink as the algorithm becomes acclimated to the leak as a normal part of the scene. Figure 4 shows a plot of the output mean value (average intensity of the output image) versus frame number. (Each frame corresponds to an elapsed time of 1/30 sec.) The increase in magnitude after initiation of the leak is clearly evident.

The results of this experimental work confirmed the ability of this technique to detect a gas leak under controlled laboratory conditions. The output mean value was found to produce a useful parameter by which a "red-line" condition could be defined. The execution time was approximately 1.1 sec/frame on relatively inexpensive image processing equipment with programming carried out in a higher-level language (FORTRAN). While this exceeds the 1/30 sec/frame processing time required for real-time implementation, it appears that real-time processing with this method is feasible with present technology, provided that special-purpose hardware and software is used.

APPLICATION TO TEST STAND DATA

The system has been applied to visible wavelength image data acquired during an actual SSME test firing in which a premature shutdown occurred. Data supplied by NASA consisted of high-speed film images at approximately 64 frames/sec transcribed onto magnetic video tape in VHS format and re-played at 30

frames/sec. For this test, following the ignition sequence, the engine entered mainstage mode and operated normally for roughly 200 frames (3+ sec). At that time, hydrogen gas began to leak from the area around the low pressure fuel turbopump (LPFTP). This leaked gas ignited intermittently, causing small flames about the powerhead for approximately 250 frames (4 sec) until a large flash fire was detected by an external sensor and the shutdown sequence was initiated.

Four image sequences of this test were processed. These sequences correspond to four different views of the engine powerhead taken during the same test firing. The data acquisition cameras were denoted numbers 1, 6, 7, and 8 by NASA. Figure 5 shows the fields of view for each of these four camera positions. These images were acquired prior to ignition and therefore are clearer than those processed during the firing. The area near the LPFTP from which the leak occurred is visible in the upper right hand corner of Fig. 5(d) (camera position 8). (Because of the wide-angle lenses used on the cameras, features are somewhat distorted.) Figure 5(a) (camera position 1) shows the typical vapor clouds that are present during an engine test.

This image data was transferred to the optical disc recorder and processed frame-by-frame with the leak detection algorithm. Because of the large amount of noise in this data, it was found that values for β and J of 0.91 and 13, respectively, were required for processing. (Use of higher values was limited by the number of frame buffers available in the image processor.) Processing began just prior to ignition following a pre-test synchronization flash and concluded with the post-test mode. For each sequence, a plot of the output mean value (average intensity of the output image) versus frame number was computed and is shown in Fig. 6.

The peaks in the plots of Fig. 6 correspond to the events occurring during the engine test described in Table I. The peaks detected during ignition were caused by lights being turned on and by the plume forming at the bottom of the nozzle. Ignition also caused the nozzle to vibrate which shook loose frost that had accumulated on various cold engine components. These events caused the first large peaks in the output mean plots. As the engine entered mainstage mode, the mean value output reduced to that caused by the noise.

The leak near the LPFTP resulted in a large peak in the output mean value at approximately frame 400 in Fig. 6(d) (camera position 8), and smaller, but still noticeable peaks at the same time in Figs. 6(a) and 6(b) (camera positions 1 and 6). While the leak was detected in three of the four sequences, it was most obvious from camera position 8. The leak caused a flow that purged the vapor clouds normally surrounding the powerhead and cleared the field of view. Flames are visible from frames 400 to 650 in camera position 6 data and resulted in substantial output mean values during that time as shown in Fig. 6(b).

The flash fire at frame 650 is evident in all the plots of Fig. 6. It was detected by the monitoring systems presently in use and the shutdown sequence was initiated. The engine vibration and water spray associated with shutdown and post-test procedures also result in large peaks in the output mean value in all four plots of Fig. 6.

As the plots in Fig. 6 show, if this system been implemented for this engine test, the leak/no leak decision would have probably been positive at approximately frame 400. Thus, the engine would have begun shutdown 4 sec before the systems presently implemented indicated. This analysis of these four data sets thus demonstrates the potential value of this technique for monitoring SSME test firings.

Several anomalous events (e.g., passing large vapor clouds) occurred which also caused the mean value to increase above what would probably be the threshold value for leak detection. Although these false alarms may incorrectly indicate failure conditions, two factors should be considered. First, this system is anticipated to be used in conjunction with other monitoring systems, and integrating the information from all sources should reduce spurious false alarms from any one system. Second, the use of infrared imaging, for which this algorithm was intended should improve performance by reducing interference from water vapor clouds and making the effects of leaking gases more visible. In light of these considerations, it is felt that the performance of the leak detection system is confirmed by this analysis.

CONCLUSIONS

A method for detecting abrupt, step-like changes in a time sequence of images has been developed, implemented and tested. The detection algorithm functions by highpass filtering the input data to remove the quiescent background intensity and computing the absolute value of the output. The resulting image is then compared at each point to a threshold to decide the occurrence of a leak at that point, or is averaged over the field of view to determine a mean value estimate. 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.

The experimental studies demonstrated the ability of this system to detect leaks in a full image frame using image processing hardware. The experiments performed also demonstrated the effects of the various parameters on the detection process. The laboratory experiments show that the system is capable of correctly detecting leaks occurring within a field of view under controlled conditions. The analysis of the test-stand data indicate the applicability of this technique to actual SSME test firings and its ability to identify anomalous events.

REFERENCES

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

[2] A. A. Shohadaee and R. A. Crawford, "SSME leak detection feasibility investigation by utilization of infrared sensor technology," *Center for Advanced Space Propulsion Second Annual Technical Symposium Proceedings*, Tullahoma, TN, Nov. 1990.

[3] J. A. Malone and L. M. Smith, "A system for sequential step detection with application to video image processing," to be published in *IEEE Trans. on Industrial Electronics.*

[4] J. A. Malone, "A system for leak detection using sequential image processing," Master's thesis, Dept. of Elec. Engr., Univ. of Tennessee, Knoxville, TN, 1991.

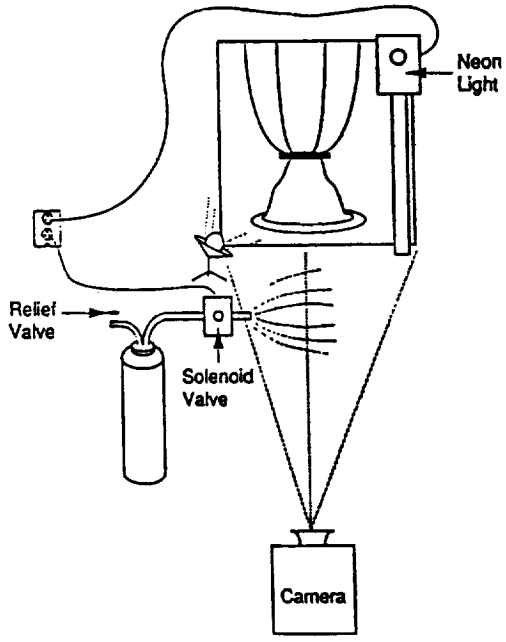


Fig. 1. Experimental laboratory setup for evaluating leak detection system.

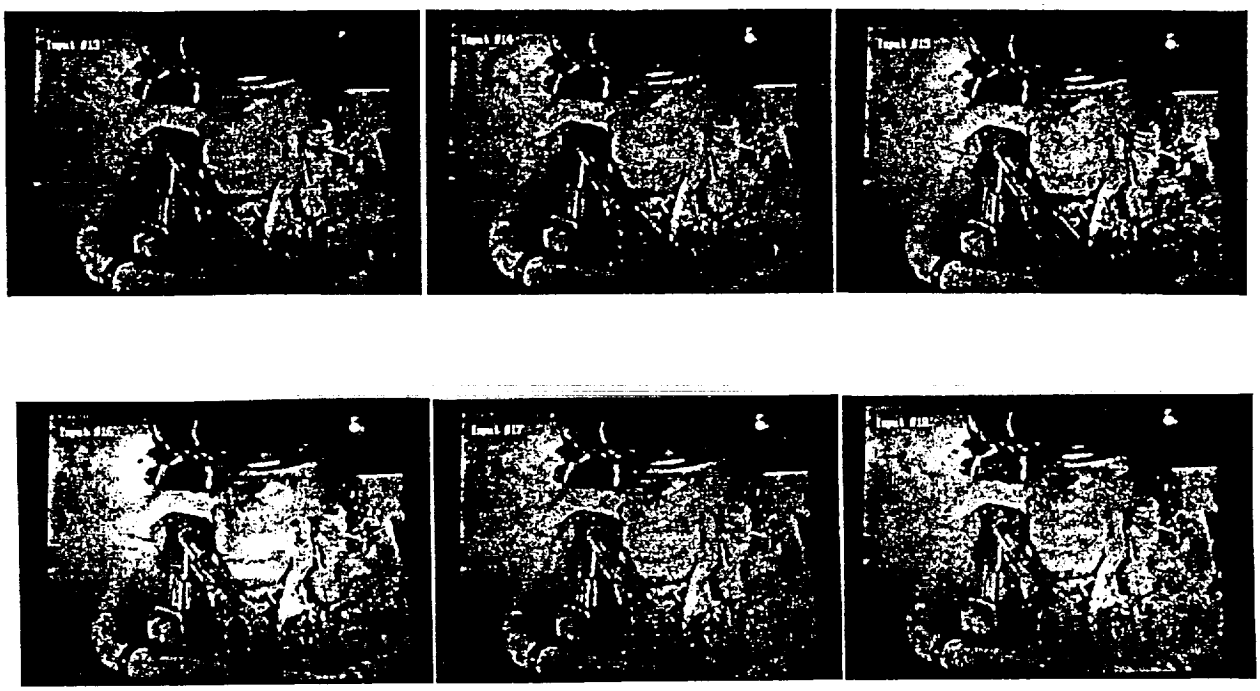


Fig. 2. Sequence of input images showing leak occurrence in laboratory setup.

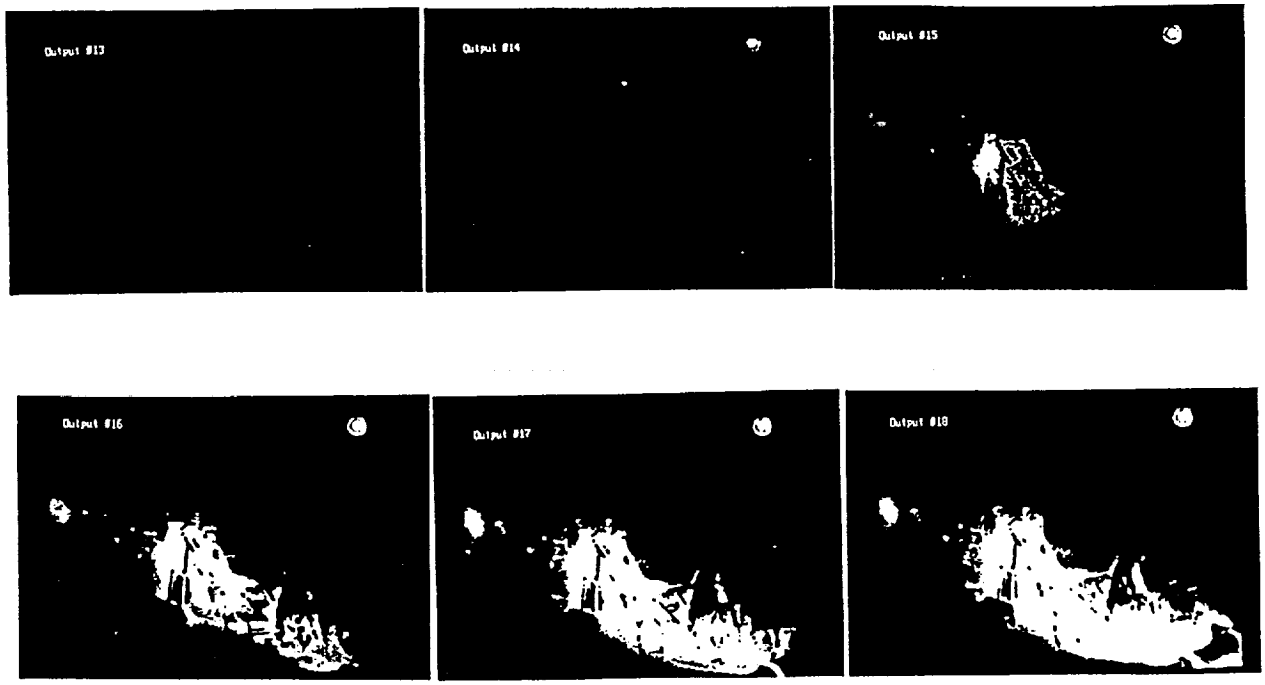


Fig. 3. Sequence of output images showing extent and location of detected laboratory leak.

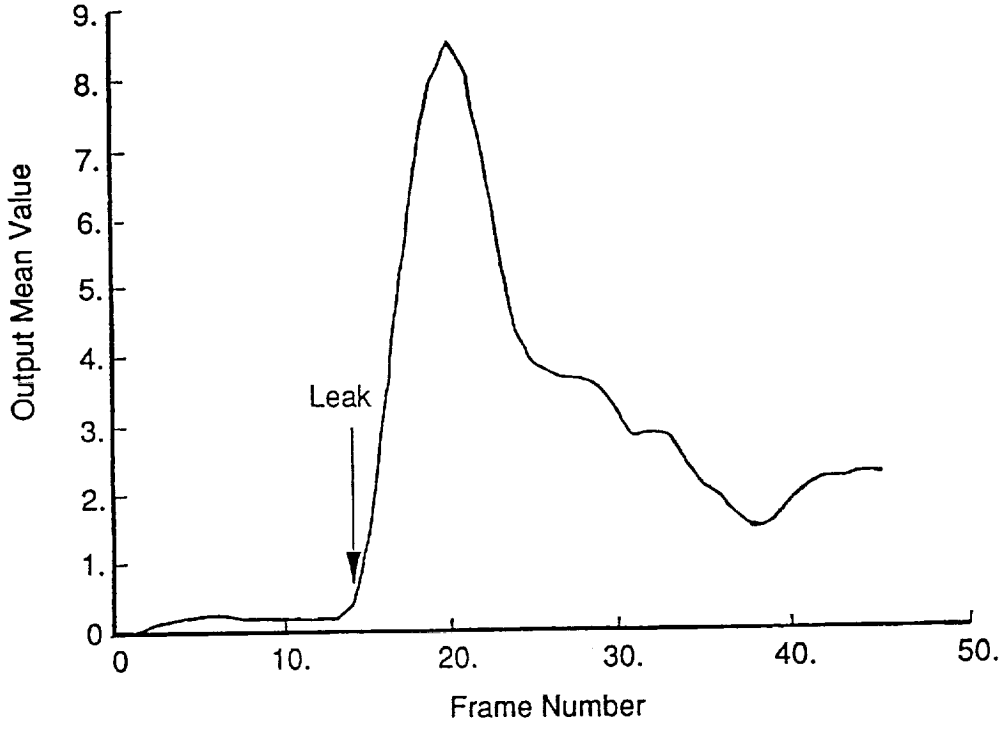


Fig. 4. Mean value of output images versus frame number for laboratory leak of Figs. 2 and 3.

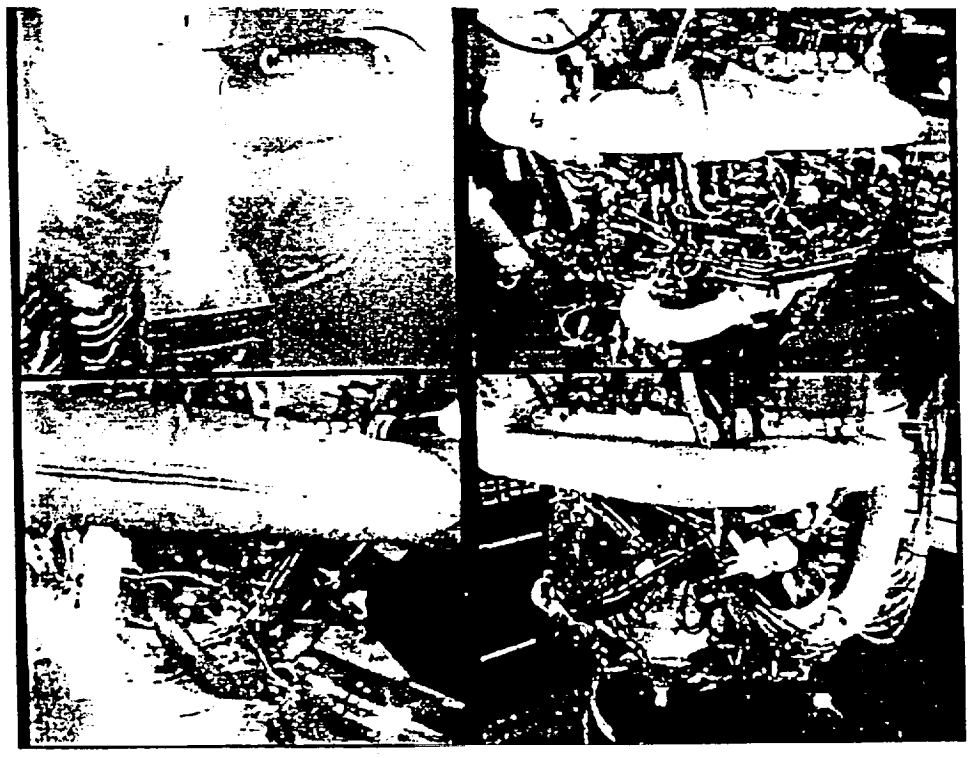


Fig. 5. Fields of view for test-stand video data.

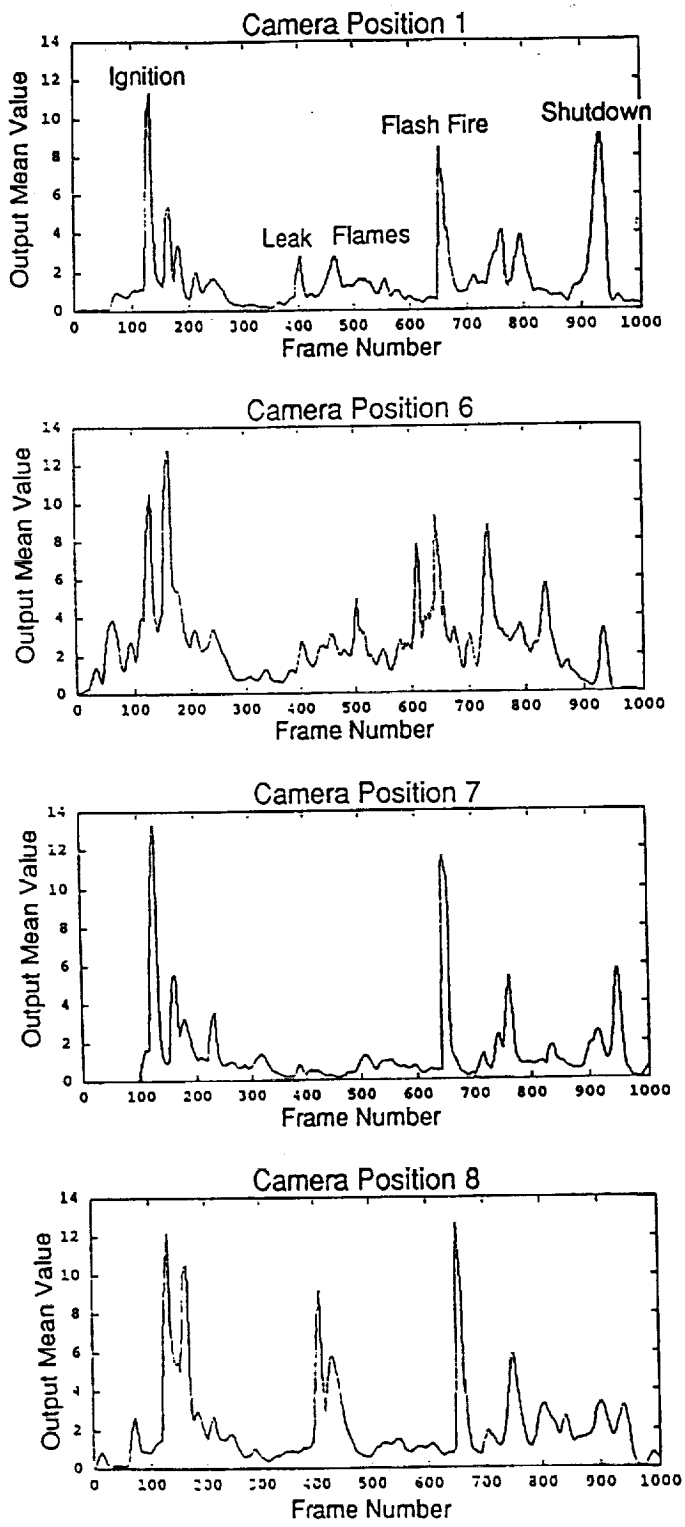


Fig. 6. Mean value of output versus frame number for test-stand video data.

NDB

Digital Image Processor System Specifications for Real-Time SSME Leak Detection

L. Montgomery Smith and Bruce W. Bomar
Department of Electrical and Computer Engineering
The University of Tennessee Space Institute
Tullahoma, Tennessee 37388-8897

Research Sponsored by
Center for Space Transportation and Applied Research
Grant 25-945
and
Marshall Space Flight Center
National Aeronautics and Space Administration
Grant NAG8-140

Report Completed: July 31, 1992

NOTICE CONCERNING THE PRODUCTS MENTIONED IN THIS REPORT

This report was compiled and written by personnel of the University of Tennessee Space Institute (UTSI) in partial fulfillment of the work objectives for Center for Space Transportation and Applied Research (CSTAR) Grant 25-945 and National Aeronautics and Space Administration (NASA) Grant NAG8-140. While its distribution is unrestricted, it is intended primarily for internal use by individuals in those agencies. For this reason, the authors have made extensive reference to products by their manufacturers and brand names. The use of these identifiers should not be construed as an endorsement of any company or product by UTSI, CSTAR or NASA beyond the recommendations contained in this report. Also, the specific capabilities and prices of the products are given based upon the information available to the authors at the time of writing, and are subject to change in the future. The authors assume no responsibility for errors resulting from incorrect or incomplete information supplied to them in the course of preparing this report.

I. INTRODUCTION AND DESCRIPTION OF THE ALGORITHM

Recent work on NASA Grant NAG8-140 has developed a method for the detection of anomalous events such as propellant leaks from the power head of the Space Shuttle Main Engine (SSME) during test firing from a time sequence of video image data. References [1] through [5] present the development and verification aspects of this technique, along with a detailed description of the underlying algorithm. For the purposes of completeness and applicability to real-time implementation, a brief review is given here.

The occurrence of a leak at a given point in the field of view of an image is characterized by a sudden but sustained change in the light intensity level at that point. For SSME testing, other time-varying artifacts (e.g. passing vapor clouds) corrupt the desired signal. The time-varying behavior of the intensity data at one given point in the presence of a leak should thus be similar to a step function, although additional spurious signals are also present. The problem is thus that of rapid detection of a step function in the presence of noise.

The processing algorithm developed for this application can be broken down into four component blocks. The first three blocks involve processes that are applied in parallel at each point (pixel location) in the image, while the fourth incorporates the spatial information over the entire field of view. In concept, the four steps can be described as follows:

1. A highpass filter is applied to remove the constant or slowly-varying background intensity while passing sudden changes.

2. A moving average filter accumulates any sustained changes transmitted through the highpass filter.

3. The absolute value of the output from the moving average is taken to allow for detection of positive or negative intensity changes.

4. The pixel values of the output image are summed or averaged over the field of view to produce a single time-varying quantity indicative of the extent and intensity of a leak.

For speed of response and simplicity of computation, the highpass filter has been chosen as a single-pole recursive filter with a z-domain transfer function given by

$$F(z) = \frac{1+\beta}{2}\frac{1-z^{-1}}{1-\beta z^{-1}}. \tag{1}$$

The cutoff frequency of the filter is determined by choice of the pole value β.

The moving average filter has a transfer function

$$\begin{aligned} G(z) &= \frac{1}{N}\sum_{k=0}^{N-1} z^{-k} \\ &= \frac{1}{N}\frac{1-z^{-N}}{1-z^{-1}}. \end{aligned} \tag{2}$$

These two filtering operations can be combined into a single cascade connection with an effective transfer function given by

$$\begin{aligned} H(z) &= F(z)G(z) \\ &= \frac{1+\beta}{2N}\frac{1-z^{-N}}{1-\beta z^{-1}}, \end{aligned} \tag{3}$$

which corresponds to the difference equation

$$y(n) = \beta y(n-1) + \frac{1+\beta}{2N}\left[x(n) - x(n-N)\right] \tag{4}$$

Equation (4) is the difference equation defining the temporal filtering that is carried out at each point in the image. Processing is completed by taking the absolute value of the resulting signals at all pixel locations and averaging them over the full field of view.

While equation (4) conceptually describes the filtering process at each pixel location, the actual implementation consists of the recursive computation given by

$$y(n) = \beta y(n-1) + x(n) - x(n-N). \tag{5}$$

Multiplication by the gain factor $(1+\beta)/2N$ is performed on the sum of the absolute values of the output pixels in conjuction with normalization to obtain the mean value. This substantially reduces computational requirements.

It has been shown [4] that the signal-to-noise ratio of the output signal can be maximized for a given value of β by choosing the number of terms in the moving average according to

$$N = -\frac{1.2564}{\ln \beta}. \tag{6}$$

This condition sets the memory requirements of the system in terms of previous input images that must be stored.

II. BASIC SYSTEM REQUIREMENTS

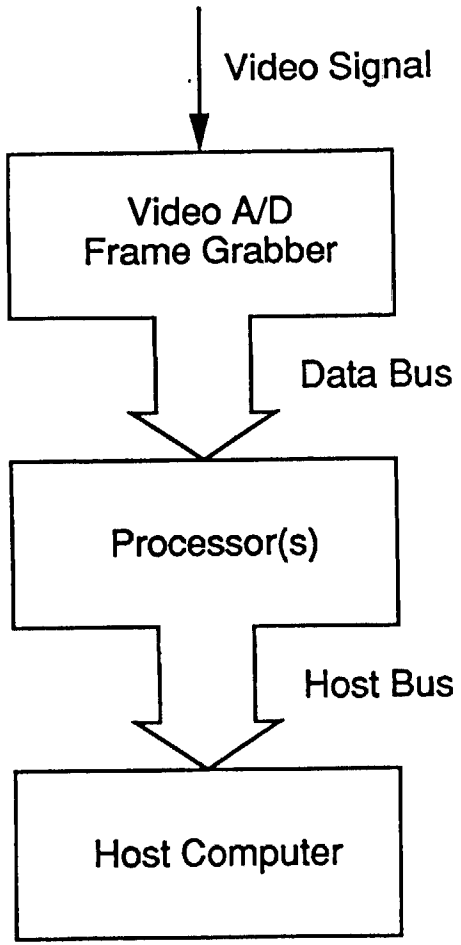
The conceptual structure of the proposed system is shown in Fig. 1 along with a summary of the nominal system specifications. An A/D converter (or frame grabber) digitizes the input video voltage signal. This data is then transferred via an internal data bus to one or more high-speed digital signal processors where the bulk of the numerical computations is carried out. The resulting output sequence of mean values is then transferred to the host computer for display and archival storage.

The proposed system is intended for standard RS-170 format input video signals with a framing rate of 30 Hz and 480 lines/frame. Adequate resolution is acheived by digitizing each line into 512 samples. The dynamic range of the intensity values can be covered with 8 bits (1 byte) per pixel. If this A/D conversion operation is carried out in a dedicated frame grabber and data transferred to other processing hardware, the continuous transfer rate of the internal data bus is nominally 7.4 Mbytes/sec.

Examination of equation (5) shows that at each pixel, 3 arithmetic operations – 1 multiply and 2 adds – are required for the filtering process. Also, the absolute value constitutes one operation/pixel, and the sum over the field of view requires one operation/pixel. For a 480×512 image pixel array and a 30 Hz framing rate, the necessary effective processing speed is thus roughly 37 million operations/sec.

Memory requirements for the processors are governed by the number of terms in the moving average N, which is determined by the criterion given in equation (6). Practical experience with SSME test stand data has shown that up to 15 previous input frames in addition to the present can be required for proper performance. One additional frame is needed for the previous output frame. While input frames can be stored in 1 byte/pixel integer format, the output frame must be in 4 byte/pixel floating-point format. The total memory required is thus 5 Mbyte of access memory.

This system design is intended to compute the sum of the absolute values of the outputs defined by (5), normalize by the appropriate scale factor $((1+\beta)/(2N \times 480 \times 512))$, and transfer those values to the host computer via its internal bus at the framing rate of 30 Hz. This produces the final mean value at discrete time intervals. This value will be written to disk for archival storage. In addition, the value can be compared with a



Input: 30 frames/seconds

Format: 480 x 512 x 1 byte pixels/frame

Transfer Rate: 7.4 Mbytes/second

Memory: 16 previous frame buffers

Algorithm: 5 operations/pixels

Processing Rate: 37 MOps/second

CG-2497

Figure 1. Basic system configuration and nominal specifications.

threshold value during processing to determine whether the "red line" condition has been exceeded indicating a leak has occurred. Another feature is that a manual override from the keyboard will be provided to prevent false alarms from occurring during the ignition sequence and other planned anomalous events.

III. RECOMMENDED SYSTEM

An analysis of available frame grabbers and processor boards resulted in the recommended system to be described in this section. It will be shown that this system definitely satisfies all processing requirements. An alternate system which may marginally be able to satisfy the processing requirements has also been defined and will be described in the next section.

Figure 2 is a block diagram of the recommended system. An overlay frame grabber (OFG) from Imaging Technology, Inc. was selected for the video digitizing function. The OFG was selected because of its ability to continuously output all frames of digitized video data over a standard synchronous digital data bus – the VISIONbus [6],[7]. The VISIONbus data is split into four separate streams, each of which is routed to a separate Texas Instruments TMS320C40 32-bit floating-point digital signal processor. The "C40" processors were chosen because: 1) they have instructions for efficiently manipulating bytes which permits packing 4 bytes per 32-bit word to save memory, 2) their floating-point capability permits the recursive portion of (5) to be carried out in floating-point arithmetic, thus minimizing finite wordlength effects, and 3) they have simple direct-memory-access (DMA) communication ports for interfacing 8-bit data sources at rates of up to 20 Mbytes/sec [8].

The four C40 processors are located on two Spirit-40 AT/ISA dual-C40 PC-plug-in boards from Sonitech International, Inc. [9]. Figure 3 is a block diagram of one of the two identical processors on this board. The C40 has two separate data buses, referred to as the local and global buses. On the Spirit-40 board, an erasable and reprogrammable read only memory (EPROM) is connected to the local bus. This EPROM stores the C40 program which is automatically loaded into internal C40 memory whenever the processor is reset. The local bus is also connected to 256K $\times$ 32-bits of 0-wait-state static memory which will be used to store the $y(n)$ outputs from (5). The global bus is connected to a PC-bus interface and to 1024K $\times$ 32-bits of 1-wait-state static memory which will be used for storing the $x(n)$ input pixel values.

The VISIONbus data is split into four streams, one for each of the C40 processors, using a custom interface board (built in-house) which sends either every fourth pixel or every fourth line of pixels to the same processor. A block diagram of this interface, which

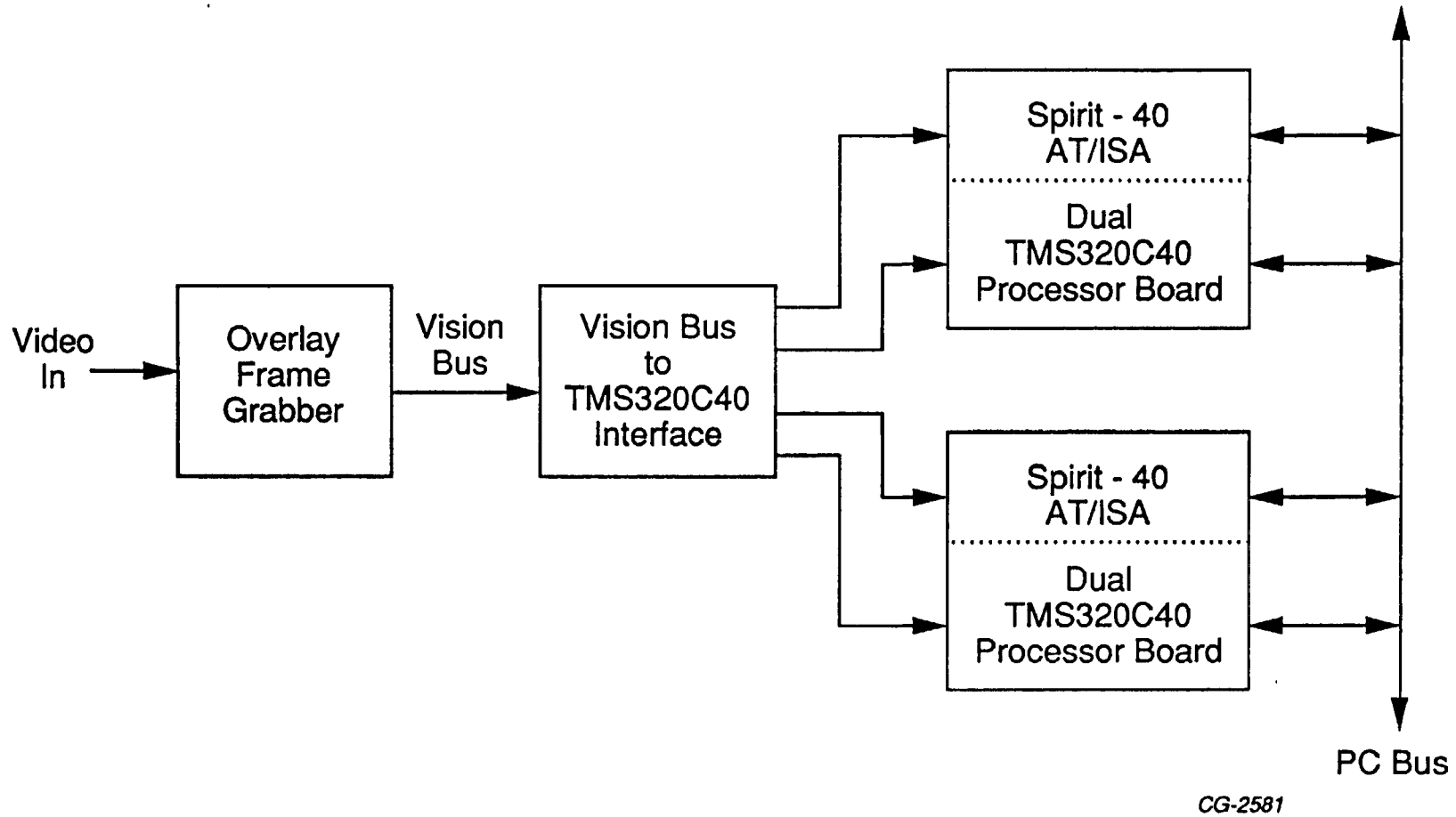


Figure 2. Processing system using TMS320C40 processors.

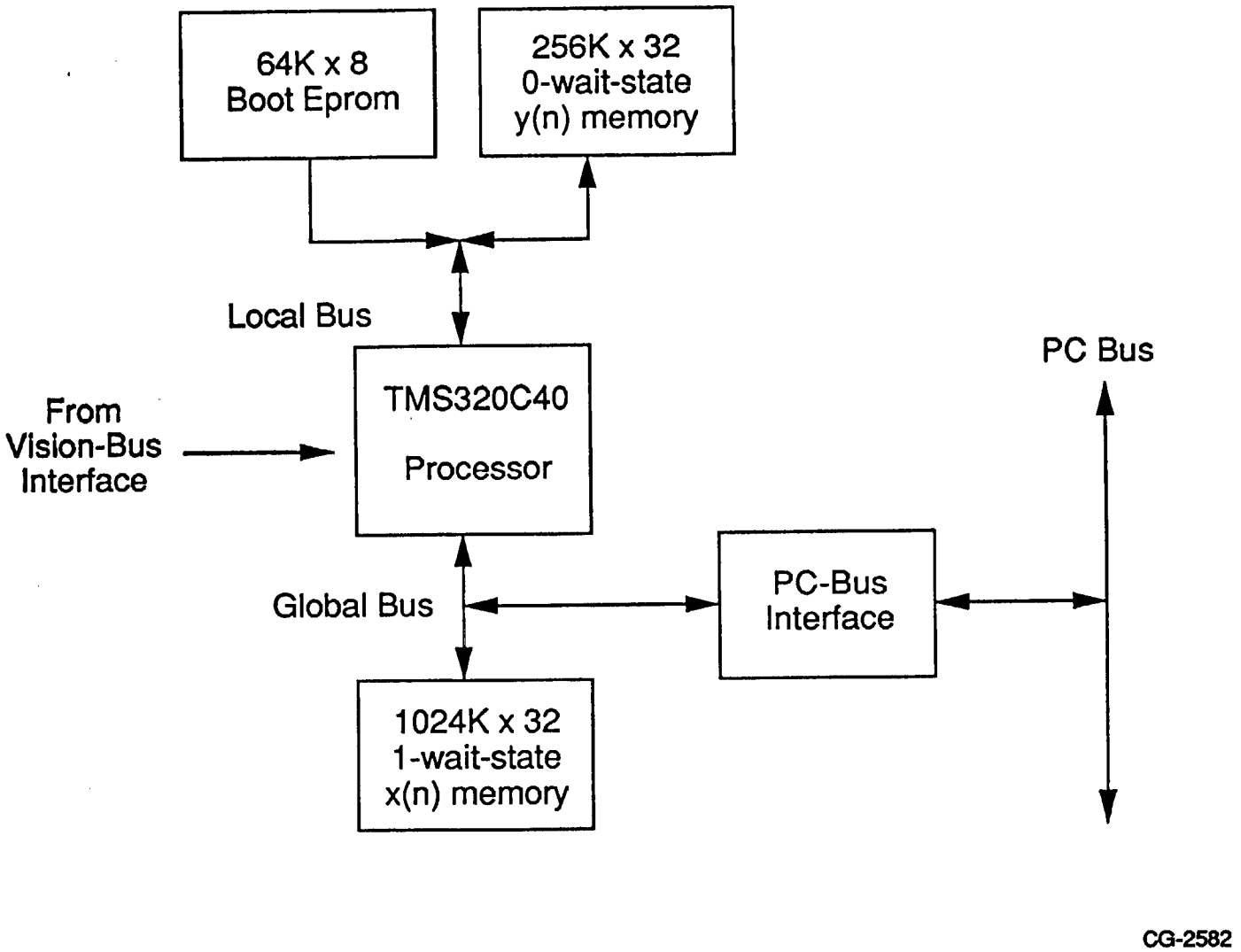


Figure 3. Configuration of each processor on dual-processor Spirit - 40 AT/ISA board.

will be fabricated on a PC-plug-in card, is shown in Figure 4. It consists of four 2K × 9-bit synchronous first-in first-out memories (FIFOs) and five state machines. Each state machine is implemented in a single programmable logic device (PLD) integrated circuit. An input state machine selects which FIFO a pixel of VISIONbus data is clocked into. Each of four output state machines handles clocking data out of one of the FIFOs and into the communication port of a C40 processor where it is automatically packed into a 32-bit word and transferred by DMA to C40 global bus memory.

The C40 assembly language program listed in the Appendix was written to verify that the C40 processors can perform the processing of (5) in the time available. This program carries out the calculations of (5), takes the absolute value of the pixels computed from (5), and sums these absolute values. The program assumes that the $x(n)$ values are stored as 8-bit pixels packed four per 32-bit word in global bus memory and that the $y(n)$ values are stored as 32-bit floating-point numbers in local bus memory. The program was run on a C40 simulator and was optimized to minimize the processing time required per pixel. This optimization involved properly locating data in memory and ordering instruction operands so bus conflicts are avoided and all instructions, except those that reference 1-wait-state memory, execute in one instruction cycle.

With 0-wait-state local bus memory, 1-wait-state global bus memory, and 40 MHz C40 processors, the program in the Appendix requires 9 instruction cycles of 50 nsec each to process one pixel giving a total processing time of 450 nsec/pixel. Although pixels become available on the VISIONbus at an average rate of one every 136 nsec, each of the four C40 processors only receives a new pixel every 544 nsec. Thus, the 450 nsec/pixel processing time is sufficient to guarantee that incoming video data can be continuously processed. (Note that 4 processors performing 5 arithmetic operations in 450 nsec corresponds to an effective processing rate of 44 million operations/sec, thus exceeding the 37 million operations/sec specified in Section II.)

Since input pixels are stored one per byte in global bus memory, the 1024 K × 32-bit global bus memory of each C40 permits 17 input images to be stored at the same time. One of these 17 image locations must be used as the destination of the current input DMA transfer. This leaves 16 image locations for storing the image currently being processed

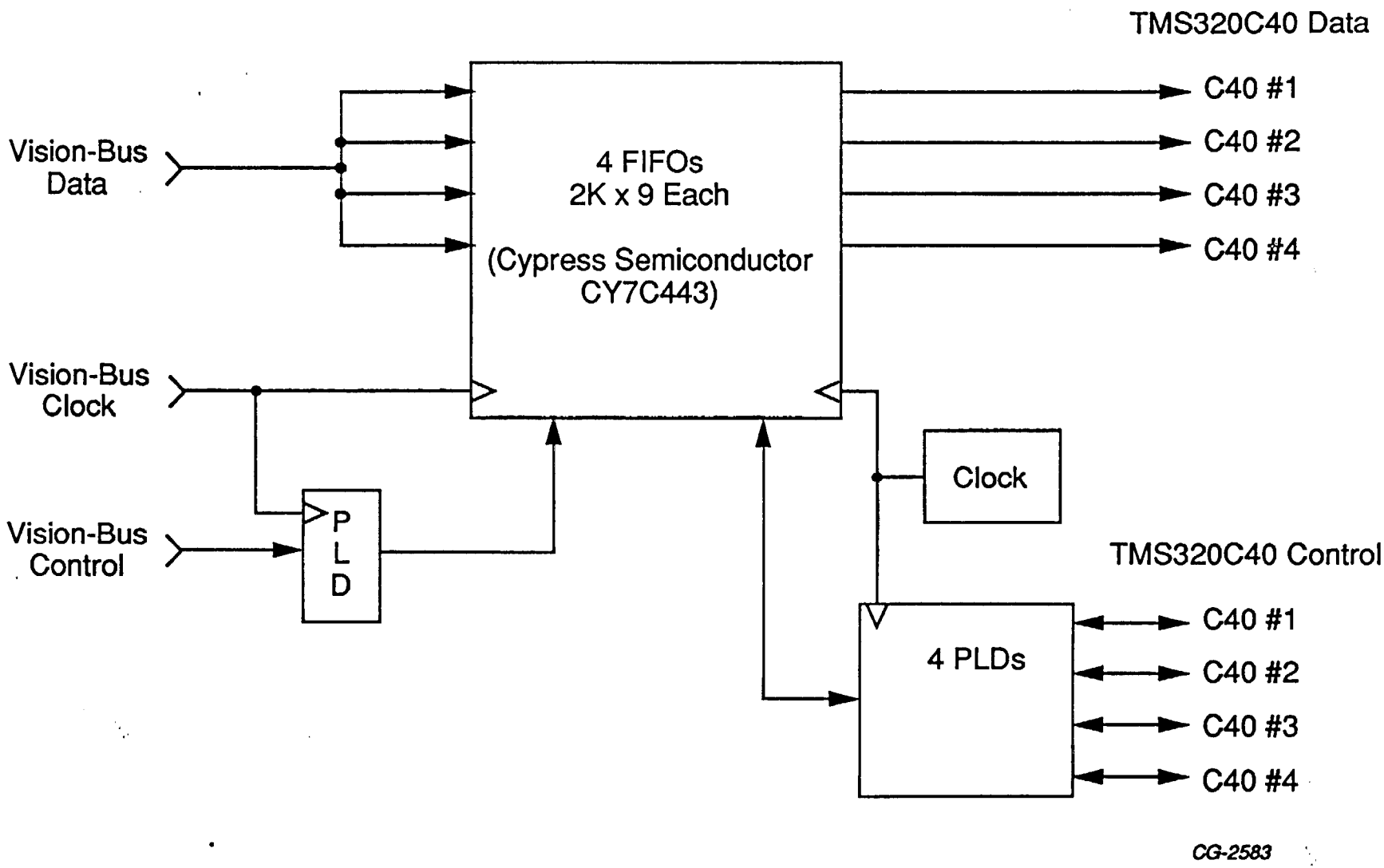


Figure 4. Vision Bus to TMS320C40 interface.

and past images. Therefore, a value as large as $N = 15$ can be used in (5). The 256 K $\times$ 32-bit local bus memory of each C40 permits one output image to be stored at a time. This is satisfactory since the new output image can always overwrite the old output image.

Recommended System Budget

Labor Estimate

Senior Engineer, Ph.D. level (hardware and software design)	100 hrs
Senior Engineer, Ph.D. level (signal analysis and integration)	100 hrs
Technician or Student Assistant, B.S. level	500 hrs

Material Cost Estimate

Host Computer and Image Display System:	
25 MHz 80386-based PC with MS-DOS 5.0	$1600
C Compiler	150
Color Video Monitor	750
Video A/D Frame Grabber:	
Imaging Technology Overlay Frame Grabber Kit	$2300
ITEX-OFG Subroutine Library	550
Interface Board:	
Printed Circuit Board Fabrication	$1000
Components	500
PAL Programmer	300
Processors:	
Sonitech Spirit-40 Boards (2 @ $6000)	$12000
256Kb 0-Wait-State Memory (4 @ $500)	2000
1024Kb 1-Wait-State Memory (4 @ $1200)	4800
C40 C Compiler, Assembler, Linker	1250
C40 Simulator	250
Total Material Cost:	$27450

Note: Figures above are based upon university and educational discounts where available.

IV. ALTERNATE SYSTEM DESIGN

An alternate system which may marginally be able to meet the processing requirements is shown in Figure 5. This system uses the same Imaging Technology OFG for digitizing the video data but uses two 50 MHz Intel i860XP microprocessors for processing the data rather than four TMS320C40 processors. The two i860XP processors are contained on an Alacron AL860-XP-AT PC-plug-in board which also contains an Alacron AL860-VB VISIONbus interface.

Figure 6 gives a block diagram of the AL-860-XP-AT with AL860-VB VISIONbus interface. Since the VISIONbus interface does not use DMA, the i860-XP procssors must use programmed transfers to move the 7.4 Mbyte/second of VISIONbus data to main memory for processing. Only a limited amount of data buffering is available on the VISIONbus interface (up to 8 Kbytes maximum) so these transfers must take place at least once every msec.

Data in main memory will be processed by the two i860-XP processors. Since the i860XP was designed for general-purpose computing rather than real-time digital signal processing, it is difficult to predict processing times accurately. This is further complicated by the use of dynamic rather than static memory on the AL-860-XP-AT board and by the use of one shared bus for both of the processors and data input. However, using assembly language instruction timings from [10] and memory access timings from [11] it is possible to predict a best case processing time of 170 nsec/pixel for a 50 MHz i860-XP processor operating with 0-wait-state sole access to the memory bus. A similar estimate of the data input timing gives a best-case value of 50 nsec/pixel for a total *best case* time of 220 nsec/pixel.

Since two i860-XP processors are available and VISIONbus data arrives at a rate of one pixel every 136 nsec, all operations on one processor for one pixel must be performed in 272 nsec. Although the 220 nsec best case time is within this limit, bus conflicts due to the use on one shared bus for all data transfers, and dynamic memory refresh operations, could easily increase the actual processing time to more than the available 272 nsec. Presently, the only way to determine for certain whether this system will meet the timing constraints is to implement the system and develop optimum assembly language code. Since there is

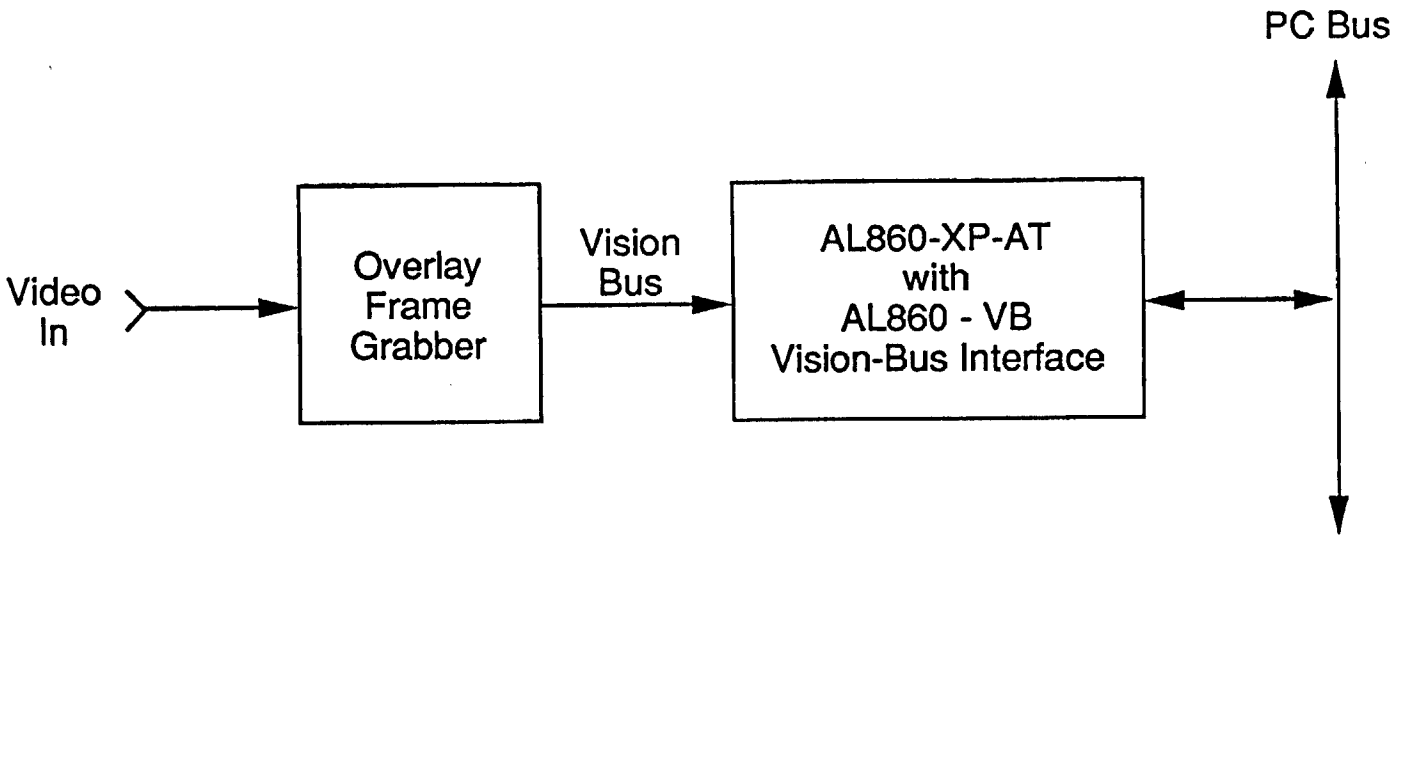


Figure 5. Processing system using i860XP processors.

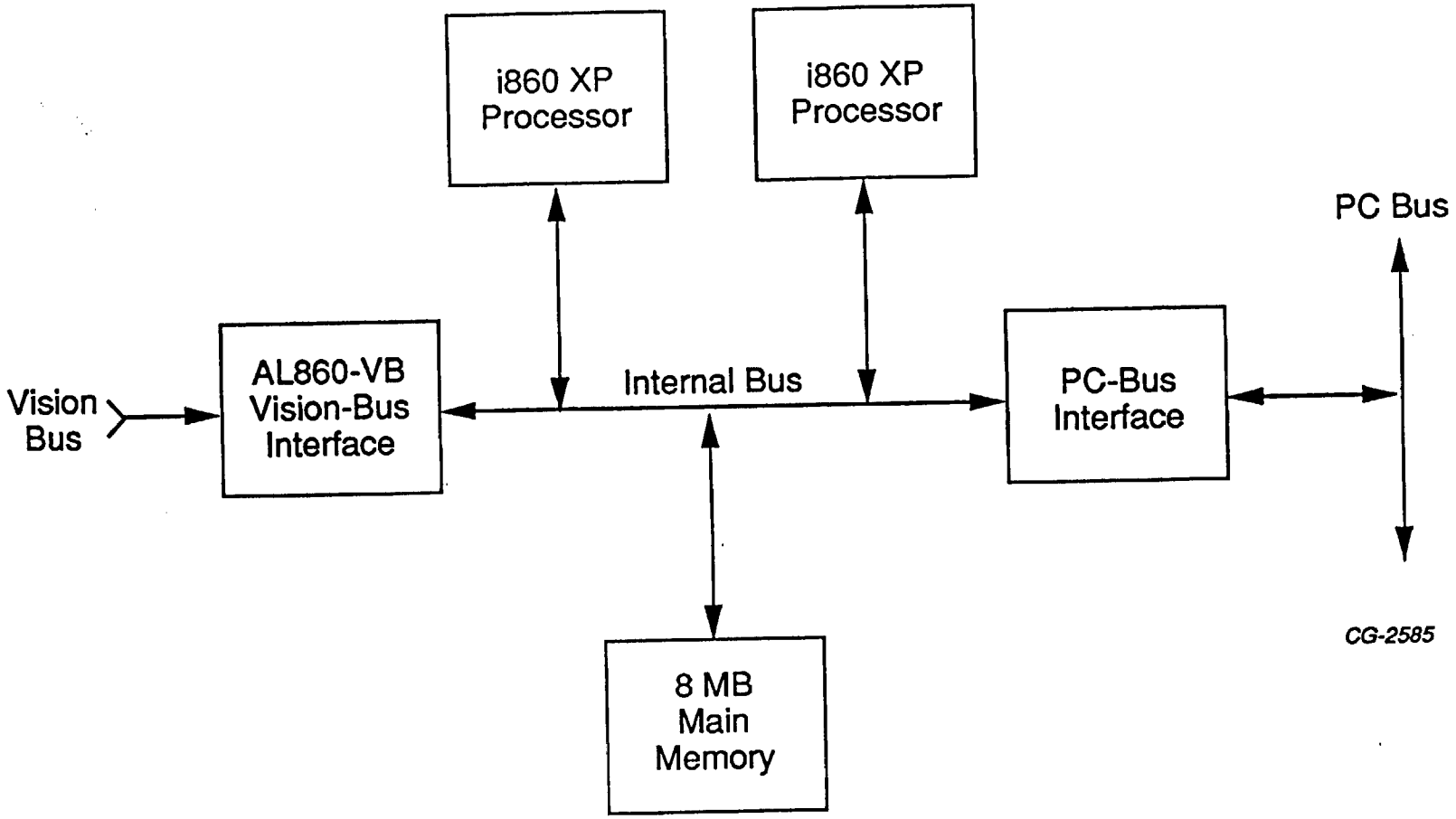


Figure 6. Block diagram of AL860-XP-AT board with vision-bus interface.

a significant probability that the implemented system will not be capable of meeting the timing requirements, we consider this approach to be high risk. However, as shown below, the cost of the i860-XP system is lower than that of the recommended system which uses four C40 processors.

Alternate System Budget

Labor Estimate

Senior Engineer, Ph.D. level (hardware and software design)	100 hrs
Senior Engineer, Ph.D. level (signal analysis and integration)	100 hrs
Technician or Student Assistant, B.S. level	500 hrs

Material Cost Estimate

Host Computer and Image Display System:	
25 MHz 80386-based PC with MS-DOS 5.0	$1600
C Compiler	150
Color Video Monitor	750
Video A/D Frame Grabber:	
Imaging Technology Overlay Frame Grabber Kit	$2300
ITEX-OFG Subroutine Library	550
Interface Board:	
Alacron AL860-VB	$750
Processors:	
Alacron AL860-AT-8-XP50-2	$10200
RT860-C-D C Compiler, Linker	2250
i860 Assembler	1000
Total Material Cost:	$19550

Note: Figures above are based upon university and educational discounts where available.

V. CONCLUSIONS

Two candidate systems have been identified and specified for real-time image processing for SSME leak detection. Both systems use an IBM compatible personal computer as a host platform and utilize the Imaging Technology Overlay Frame Grabber with the VISIONbus for data acquisition and transfer. The first system uses four TMS320C40 processsors located on two Sonitech Spirit-40 add-in cards and requires that a custom interface be fabricated to transfer data from the frame grabber to the processors. The second system uses a dual i860 Alacron AL860-XP-AT processor card and a commercially available AL860-VB interface for data transfer.

Despite the requirement for fabricating the data transfer interface and additional material cost, the first system is the recommended option because of the low risk associated with assuring adequate performance. The processing algorithm has been programmed and proper timing to achieve the required effective throughput rate has been verified. In the case of the second candidate system, performance cannot be verified without actual procurement of the components. Thus, the possibility exists that it may not be capable of meeting the basic system specifications, and is therefore a high risk alternative.

REFERENCES

[1] J. A. Malone, "A system for leak detection using sequential image processing," Master's thesis, Dept. of Elec. Engr., Univ. of Tennessee, Knoxville, TN, Aug. 1991.

[2] L. M. Smith, J. A. Malone and R. A. Crawford, "Space shuttle main engine propellant path leak detection using sequential image processing," *Proceedings of the CSTAR Third Annual Technical Symposium*, pp. 1-11, Jan. 1992.

[3] L. M. Smith, J. A. Malone and R. A. Crawford, "Leak detection from the space shuttle main engine using sequential image processing," *1992 Conference on Advanced Earth-to-Orbit Propulsion Technology*, May 1992.

[4] J. A. Malone and L. M. Smith, "A system for sequential step detection with application to video image processing," accepted for publication in *IEEE Trans. Indust. Electron.*

[5] J. A. Malone, L. M. Smith and R. A. Crawford, "Detection of leaks from the space shuttle main engine using sequential image processing," submitted to *IEEE Trans. Aerosp. Electron. Syst.*

[6] Imaging Technology, Inc., *OFG Hardware Reference Manual.* Bedford, MA: Imaging Technology. Document number 47-H30001-03, Aug. 1991.

[7] Imaging Technology, Inc., *VISIONbus Video Bus Specification.* Bedford, MA: Imaging Technology. Document number 47-T30000-00, Oct. 1991.

[8] Texas Instruments, Inc., TMS320C4x User's Guide. Dallas, TX: Texas Instruments, 1991.

[9] SONITECH International, Inc., *DSP Solutions Product Catalog.* Wellesley, MA: SONITECH International, 1992.

[10] Intel, *i860 Microprocessor Family Programmer's Reference Manual.* Mt. Prospect, IL: Intel, 1991.

[11] Intel, *i860 64-bit Microprocessor Hardware Reference Manual.* Mt. Prospect, IL: Intel, 1990.

APPENDIX - C40 PROGRAM LISTING

```
;
;       TMS320C40 program for time-filtering image pixels and summing the
;       absolute value of the filtered outputs.  The program uses 9
;       instruction cycles or 450 nanoseconds per pixel with 0-wait-state
;       local bus memory and 1-wait-state global bus memory.  The y(n) output
;       values are stored in local bus memory as 32-bit words.  The
;       x(n), x(n-1), ..., x(n-N) input values are stored as 8-bit bytes in
;       global bus memory.  The computer program and constants (like beta)
;       are stored in memory internal to the TMS320C40 chip.
;
        .data                           ;Internal memory RAM0 starting at 0x02ff800
LB1     .word   3e528050h               ;Constant for initializing the global bus
LB2     .word   3e528050h               ;Constant for initializing the local bus
LA      .word   100000h                 ;Bus interface control register base address
GLBL1   .word   80000000h               ;Global memory bus address of x(n)
GLBL2   .word   80000200h               ;Global memory bus address of x(n-N)
OUTP    .word   300000h                 ;Local memory bus address of y(n)
BETA    .float  0.905                   ;Value of beta
CONST   .float  2.98133e-7              ;Value of (1+beta)/(2*N*480*512) for N=13
BETAPTR .word   BETA                    ;Pointer to beta
COUNT   .word   87040                   ;Number of pixels to process on one processor
;
        .text                           ;Internal memory RAM1 starting at 0x02ffc00
START:  LDP     LB1                     ;Program 0 external wait states on both the
        LDI     @LA,AR0                 ;  local and global buses.  The  value of LB1
        LDI     @LB1,R0                 ;  changes for 1 wait state on the global bus.
        STI     R0,*AR0++(4)            ;Program the global bus
        LDI     @LB2,R0
        STI     R0,*AR0                 ;Program the local bus
;
        LDI     @GLBL1,AR0              ;Load pointer to input x(n) on the global bus
        LDI     @GLBL2,AR1              ;Load pointer to input x(n-N) on global bus
        LDI     @OUTP,AR2               ;Load pointer to input y(n-1) on the local bus
        LDI     AR2,AR3                 ;Load pointer to output y(n) on the local bus
        SUBI    1,AR3                   ;Reduce output pointer -- first output is zero
        LDI     @BETAPTR,AR4            ;Pointer to beta stored in internal memory
        LDI     @COUNT,RC               ;Number of pixels to process = loop count RC
        RPTBD   LOOP                    ;Initiate delayed loop
        LDF     0.0,R0                  ;Initialize R0, R6, and R7 to zero
        LDF     0.0,R6
        LDF     0.0,R7                  ;R7 = running sum of output absolute values
;
        LDI     *AR0++,R1               ;Start of loop --
        LDI     *AR1++,R3               ;  Get 4 pixels of x(n) and x(n-N)
        LB0     R1,R5                   ;Unpack first of 4 pixels of x(n)
        LB0     R3,R4                   ;Unpack first of 4 pixels of x(n-N)
        SUBI    R4,R5                   ;Form x(n)-x(n-N) for first of 4 pixels
        MPYF3   *AR2++,*AR4,R0          ;Form y(n-1)*beta in R0 and simultaneously add
||      ADDF3   R6,R0,R2                ;  last pixel's x(n)-x(n-N) to its y(n-1)*beta
        STF     R2,*AR3++               ;Store y(n) for previous loop's fourth pixel
        ABSF    R2,R2                   ;Take absolute value of y(n)
        ADDF    R2,R7                   ;Add absolute value to running sum
        FLOAT   R5,R6                   ;Float first pixel's x(n)-x(n-N)
        LB1     R1,R5                   ;Unpack second of 4 pixels of x(n)
        LB1     R3,R4                   ;Unpack second of 4 pixels of x(n-N)
        SUBI    R4,R5                   ;Form x(n)-x(n-N) for second of 4 pixels
        MPYF3   *AR2++,*AR4,R0          ;Form y(n-1)*beta in R0 and simultaneously add
||      ADDF3   R6,R0,R2                ;  last pixel's x(n)-x(n-N) to its y(n-1)*beta
        STF     R2,*AR3++               ;Store first pixel's output y(n)
        ABSF    R2,R2                   ;Take absolute value of y(n)
```

```
        ADDF    R2,R7           ;Add absolute value to running sum
        FLOAT   R5,R6           ;Float second pixel's x(n)-x(n-N)
        LB2     R1,R5           ;Unpack third of 4 pixels of x(n)
        LB2     R3,R4           ;Unpack third of 4 pixels of x(n-N)
        SUBI    R4,R5           ;Form x(n)-x(n-N) for third of 4 pixels
        MPYF3   *AR2++,*AR4,R0  ;Form y(n-1)*beta in R0 and simultaneously add
||      ADDF3   R6,R0,R2        ;  last pixel's x(n)-x(n-N) to its y(n-1)*beta
        STF     R2,*AR3++       ;Store second pixel's output y(n)
        ABSF    R2,R2           ;Take absolute value of y(n)
        ADDF    R2,R7           ;Add absolute value to running sum
        FLOAT   R5,R6           ;Float third pixel's x(n)-x(n-N)
        LB3     R1,R5           ;Unpack fourth of 4 pixels of x(n)
        LB3     R3,R4           ;Unpack fourth of 4 pixels of x(n-N)
        SUBI    R4,R5           ;Form x(n)-x(n-N) for fourth of 4 pixels
        MPYF3   *AR2++,*AR4,R0  ;Form y(n-1)*beta in R0 and simultaneously add
||      ADDF3   R6,R0,R2        ;  last pixel's x(n)-x(n-N) to its y(n-1)*beta
        STF     R2,*AR3++       ;Store third pixel's output y(n)
        ABSF    R2,R2           ;Take absolute value of y(n)
        ADDF    R2,R7           ;Add absolute value to running sum
LOOP    FLOAT   R5,R6           ;Float fourth pixel's x(n)-x(n-N)
;
        ADDF    R6,R0           ;Add x(n)-x(n-N) to y(n-1)*beta for fourth
        STF     R0,*AR3++       ;  pixel of last loop pass and store result
        ABSF    R0,R0           ;Take absolute value of last output
        ADDF    R0,R7           ;Add absolute value to running sum
;
        MPYF    @CONST,R7       ;Scale sum by (1+beta)/(2*N*480*512)
        TOIEEE  R7,R7           ;Convert final result to IEEE floating point
;
;       At this point R7 contains the scaled absolute-value-sum of filtered
;       pixels in IEEE 754 standard floating-point format.
;
        .end
```

NDB

A COLOR CHANGE DETECTION SYSTEM FOR VIDEO SIGNALS WITH APPLICATIONS TO SPECTRAL ANALYSIS OF ROCKET ENGINE PLUMES

W. A. Hunt and L. M. Smith
Department of Electrical and Computer Engineering
The University of Tennessee Space Institute
Center for Laser Applications
Tullahoma, TN 37388

Abstract

A system to detect step-like color changes in sequential image data has been developed and tested. The exhaust of the space shuttle main engine(SSME) or other rocket engines typically maintains a nearly constant color during normal operation. If at some point internal erosion becomes significant or other anomalous event occurs, a noticeable color change often appears in the exhaust. The color change detection system applies a step-change detection algorithm to each color channel of standard video. After applying the algorithm, each channel output is squared and weighted and then combined to form a single channel output signal. This output signal is compared to a threshold value to determine if a color change has occurred. The results of numerical and laboratory experiments are presented to verify the performance of the system.

I. Introduction

A robust, inexpensive method of real-time spectrum shift detection is desirable to monitor rocket engine plumes during test firing. During normal operation, the exhaust of the space shuttle main engine (SSME) exhibits a nearly constant color. If at some point an anomalous event occurs there is a distinctive, visible color change within the exhaust plume. This color change can be attributed to internal components erosion or other some type of failure.

The color change detection system design uses standard RGB+Sync video as its measurement signal. During SSME tests, such as technology test bed (TTB) tests, many cameras which produce such video are used to monitor and record the test. It is the ultimate application of the color change detection system to use these systems already in place to monitor the exhaust in real time. The advantages of such a system are readily apparent: 1) measurement devices, color video cameras, are already in place during tests, 2) the data acquisition and processing can be performed in real-time, 3) the system detects color shifts caused by a variety of reasons, and 4) the system automatically gives a reading based on the area in the field of view showing a change; Therefore, the system gives an output based on the magnitude and spatial extent of the anomalous event.

The color change detection system is based on an algorithm that was developed and tested by Malone [2] and by Malone and Smith [3] for application to detect step-like changes in intensity in sequential image data. In their study they found that while methods existed that would detect changes in a series of image data, such as image averaging techniques, none were suitable for their requirements. These requirements included real-time detection, noise reduction, suppression of false alarms, and detection of small change amplitudes. The dynamics of the color change occurrences are similar in that the color change occurrence is a sudden, sustained change in a sequence of images. In the case of a color change, it is very possible for there to be no change in overall intensity while having a large change in hue. It is necessary then to use sequential *color* image data versus the use of monochrome image data as in the leak detection system developed by Malone and Smith.

The color change detection system described here applies Malone and Smith's algorithm to the three channels (red, green, and blue) of standard analog video to detect color changes. The algorithm consists of a highpass filter to filter quiescent intensity values, cascaded with a moving-average filter to decrease the noise variance on each channel. The output of each moving-average filter is squared and weighted. The weighted squared outputs are then summed and the square root taken to yield a single, time-index varying signal. This signal is compared to a threshold to determine if a color change has occurred. The weighting of each channel was chosen by the National Television Standards Committee (NTSC) weightings to approximate the response of the human eye.

The system developed to detect color changes was implemented on a PC in FORTRAN as part of a numerical study to ensure the operation of the system

for many different conditions. The amplitude of color change was varied and the probability of correct detection by the system was calculated for various noise conditions. The overall intensity was also varied to determine the system response to such occurrences.

After the numerical study was completed, an experimental trial of the system was undertaken. A salt solution was introduced into a Bunsen burner airstream to cause a color change in the flame. The flame was recorded by video camera and stored for later processing. The concentration of the solution was varied to yield different color change intensities. The recorded video sequences were processed using a PC based image processing system. The system parameters were varied to determine their effect on system performance. The resulting amplitudes were plotted versus frame number to quantitatively analyze system performance.

II. Background

In rocket engines, the color exhibited by the exhaust plume remains roughly constant during normal operation. If at some point an anomalous event within the engine causes an increase of foreign matter in the fuel, such as an increase of metal concentrations, there will be a noticeable change in the color of the exhaust. This can be associated with either the absorption of light or the emission of light at another wavelength. Such an event often occurs in a very short time and can lead to catastrophic failure. The form of this spectral shift is therefore a step-like change, that, if detected, can be used to initiate shut-down and prevent further engine damage.

Processing of video signals is conveniently carried out digitally. The RS-170 video format consists of 480 lines of video data in each frame. When digitized by standard commercial frame grabber cards, 512 pixels per line with 256 possible intensity levels per pixel are obtained. Thus a digitized color video signal consists of three sequences of 480 by 512 8 bit image data. Utilizing the entire field of view given by video provides an effective 480 by 512 detector array as compared to a single-detector system. This yields redundancies that increases the reliability of the system results. For example, if a step-change is detected at one pixel, but not at other pixels, it can be assumed to be a step-change caused by noise. If the change is detected at multiple pixels, one has much greater confidence that it is not a false detection.

Previously, Malone [2] and Malone and Smith [3], developed a system to detect step-like changes of intensity in sequential image data. In their study, they found that an effective method of detecting such changes that met their criterion did not exist. Their requirements for such a method were 1) must be computationally efficient for potential real-time application, 2) must remove non-changing or slowly changing intensities, and 3) must be robust in terms of noise and background intensities. In their effort, Malone and Smith developed a very effective system that meets the requirements.

Their system consists of a highpass filter cascaded with a moving-average filter. The absolute value is taken of the output of the moving-average filter and the result is compared to a threshold. The highpass filter removes slow-changing or non-changing intensities while allowing the higher frequency components of a step-change to pass unattenuated. The moving-average filter minimizes the effects of noise by averaging the incoming values with a set number of past input values. Hence the average is said to "move" because the average is for only the present input value with a set number of past values. Thus only a sustained change will yield a significant output. Malone and Smith determined an optimum number of terms in the moving-average filter based on the cutoff frequency of the highpass filter to maximize the signal-to-noise ratio at the output. A single-pole highpass filter was used to minimize the computational complexity of the system. The algorithm developed by Malone and Smith coincides with the requirements for a color-change detection system applied to each channel separately. The optimum relations obtained by Malone and Smith are compared to those obtained for the color change detection in the next section.

III. The Color Change Detection System

This system is based on the application of a monochrome step-change detection algorithm proposed by Smith [3] and later developed and applied by Malone [2]. The basic algorithm was developed to be computationally efficient, insensitive to noise and background intensity levels, and able to detect step changes with varying amplitudes, both positive and negative. Though Smith and Malone [3] developed the algorithm for a leak detection system, the basic algorithm can find uses in many other applications.

The system developed to detect color changes is shown in Fig. 1. This system is applied to each pixel of each image using the same system parameters. The input for each channel is first highpass filtered. The output from each highpass filter is then passed through a moving average filter to decrease the effects of noise and accumulate any sudden signal change transmitted through the highpass filter. Each channel is then weighted by values determined *a priori*. These values were chosen by the NTSC weightings to approximate the response of the human eye. After being weighted, the three channels are summed to produce a single, time-varying value for each pixel position. The individual pixel values are averaged over the entire frame to produce a single value per frame. This value then is not only a measure of change in the image, but also a measure of the area where the change occurs. This value is compared to a predetermined threshold value to decide whether a step change has occurred.

The z-transfer equation for a single-pole highpass filter is

$$H_{HP}(z) = \frac{1+\beta}{2}\left(\frac{1-z^{-1}}{1-\beta z^{-1}}\right) \qquad (1)$$

where β represents the pole location. This structure was selected because of its computational efficiency and small delay. It was also found that in terms of its application, a narrow transition region does not significantly enhance system performance. The input-output relation for a moving average filter is

$$y(n) = \frac{1}{J}\sum_{k=0}^{J-1} x(n-k) \qquad (2)$$

whose recursive form is

$$y(n) = \frac{1}{J}[x(n) - x(n-J)] + y(n-1). \qquad (3)$$

The related z-transfer equation for a recursive moving average filter is

$$H_{MA}(z) = \frac{1}{J}\left(\frac{1-z^{-J}}{1-z^{-1}}\right). \qquad (4)$$

Cascading this with the single pole highpass filter shown in (1) results in

$$H_C(z) = \frac{1+\beta}{2J}\left(\frac{1-z^{-J}}{1-\beta z^{-1}}\right). \qquad (5)$$

The input-output relationship for the cascaded highpass filter and moving average filter is

$$y(n) = \frac{1+\beta}{2J}[x(n) - x(n-J)] + \beta y(n-1). \qquad (6)$$

The weighting value for each channel is determined *a priori*. The weighting values were chosen by the NTSC weighting to approximate the response of the human eye. These values are equal to 0.3, 0.59, and 0.11 for the red, green, and blue channels, respectively and are designated as *a*, *b*, and *c*.

The system response to an input consisting of a signal with additive noise, signal with no noise, and noise only are analyzed in the following sections. In the case of a signal with noise, the goal is to determine system parameters to maximize the signal-to-noise ratio at the output of the system. After system parameters are determined to maximize the signal-to-noise ratio, the system using these parameters is analyzed to determine the minimum detectable step in the absence

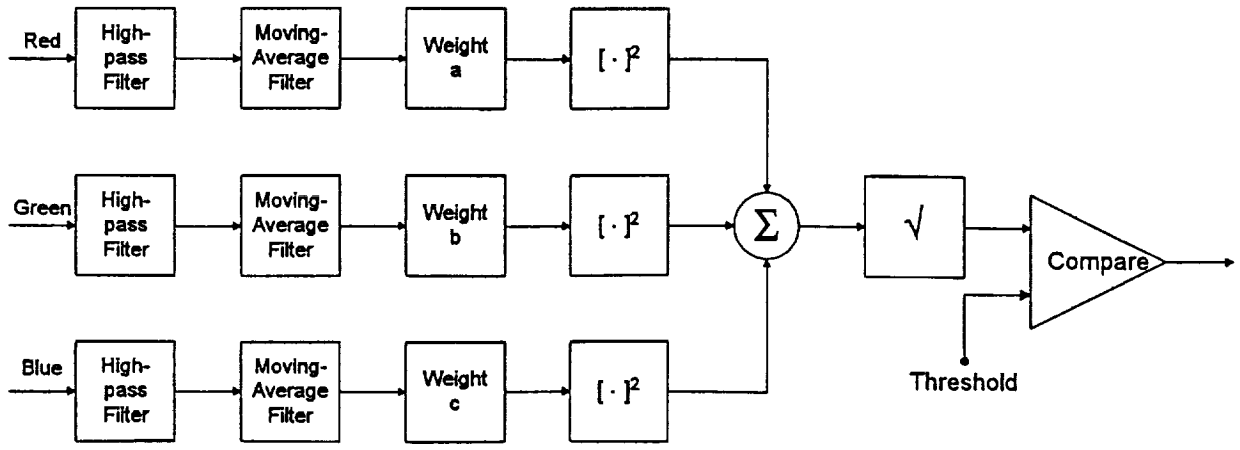


Fig. 1. Block diagram of color change detection system

of noise. The system response to a noise only input is then considered to determine the probability of false detection at any given instant.

Signal with Noise Input

The input to the system is assumed to be

$$w_1(n) = R_0 + Ru(n-n_0) + \varepsilon_1(n), \quad (7)$$

$$w_2(n) = G_0 + Gu(n-n_0) + \varepsilon_2(n), \quad (8)$$

and

$$w_3(n) = B_0 + Bu(n-n_0) + \varepsilon_3(n) \quad (9)$$

where R_0, G_0, and B_0 are the constant or slowly varying signal portion, R, G, and B are the step changes in the respective color channels, and ε_1, ε_2, and ε_3 are the noise portions of the input. The instant of step occurrence is denoted by n_0. As was shown in [3], after the highpass filter and moving-average filter have been applied to the input, the maximum signal value of the output occurs at a time index value equal to J-1 samples later and assuming the step occurred at n_0=0 for simplicity of notation is given by

$$y_1(J-1) = \frac{R}{2J}\left(\frac{\beta+1}{\beta-1}\right)(1-\beta^J) + \frac{1}{J}\sum_{k=0}^{J-1}\varepsilon_1(J-1-k), \quad (10)$$

$$y_2(J-1) = \frac{G}{2J}\left(\frac{\beta+1}{\beta-1}\right)(1-\beta^J) + \frac{1}{J}\sum_{k=0}^{J-1}\varepsilon_2(J-1-k), \quad (11)$$

and

$$y_3(J-1) = \frac{B}{2J}\left(\frac{\beta+1}{\beta-1}\right)(1-\beta^J) + \frac{1}{J}\sum_{k=0}^{J-1}\varepsilon_3(J-1-k). \quad (12)$$

This response is the same as determined by Malone and Smith [3], only here there are three channels. In their analysis, they determined that the maximum signal-to-noise ratio could be obtained by choosing the number of terms in the moving average, J, by the expression

$$J = -\frac{1.2564}{\ln(\beta)}. \quad (13)$$

The signal-to-noise ratio was defined by Malone and Smith as the maximum signal response divided by the standard deviation of the noise at the output of the moving-average filter. The optimum relation shown in (13) will be shown to hold true even after each channel output is weighted, squared, summed together and the square root is taken.

Noise Only Input

The signal performance for inputs consisting of only noise will be considered next. The noise in each channel will be considered as a random sequence with a Gaussian distribution. The noise is considered independent in each channel with the standard deviation of the noise in each channel to be equal. After each channel is weighted and squared, the probability density function for noise in each channel becomes

$$f_{\zeta_1}(\zeta_1) = \frac{1}{2\sigma_1\sqrt{2\pi\zeta_1}} e^{\frac{-\zeta_1}{2\sigma_1^2}}, \quad (14)$$

$$f_{\zeta_2}(\zeta_2) = \frac{1}{2\sigma_2\sqrt{2\pi\zeta_2}} e^{\frac{-\zeta_2}{2\sigma_2^2}}, \quad (15)$$

and

$$f_{\zeta_3}(\zeta_3) = \frac{1}{2\sigma_3\sqrt{2\pi\zeta_3}} e^{\frac{-\zeta_3}{2\sigma_3^2}} \quad (16)$$

where

$$\sigma_1 = a\frac{\sigma_0}{\sqrt{J}}, \quad (17)$$

$$\sigma_2 = b\frac{\sigma_0}{\sqrt{J}}, \quad (18)$$

and

$$\sigma_3 = c\frac{\sigma_0}{\sqrt{J}} \quad (19)$$

are the standard deviations of noise in each channel, respectively and σ_0 is the standard deviation of the input noise sequences $\varepsilon_1(n)$, $\varepsilon_2(n)$, and $\varepsilon_3(n)$. After summing the three channels of noise together, a new noise sequence can be defined as

$$\zeta(n) = \zeta_1(n) + \zeta_2(n) + \zeta_3(n) \quad (20)$$

whose probability density function becomes

$$f_\zeta(\zeta) = f_{\zeta_1}(\zeta) * f_{\zeta_2}(\zeta) * f_{\zeta_3}(\zeta) \quad (21)$$

which is the convolution of the density functions from the three channels. This operation is very complex for Gaussian distributed variables with different variances. Although the variances are different as a result of the weighting factors, they are all within an order of

magnitude of one another. To a reasonable approximation, they can be assumed to be equal variance with an effective variance defined as

$$\sigma_{eff} = \sqrt[3]{\sigma_1\sigma_2\sigma_3} = \sqrt[3]{abc}\,\frac{\sigma_0}{\sqrt{J}}. \quad (22)$$

After the square root of the sum of the three channels is taken, the final output noise sequence can be defined as $\eta(n)$. The probability density function of this variable after transformation will then be approximately

$$f_\eta(\eta) = \sqrt{\frac{2}{\pi}}\frac{\eta^2}{\sigma_{eff}^3}e^{\frac{-\eta^2}{2\sigma_{eff}^2}} \quad (23)$$

whose standard deviation is found to be

$$\sqrt{\left(3-\frac{8}{\pi}\right)}\,\sigma_{eff}. \quad (24)$$

Using (23), the probability the output of the system will exceed a given threshold in the absence of a signal (a false detection) can be found by the integration

$$P(\eta > T) = 1 - \int_0^T f_\eta(\eta)\,d\eta \quad (25)$$

which, when evaluated, equals

$$P_{fd} = P(\eta > T) = 1 + \frac{T}{\sigma_{eff}}\sqrt{\frac{2}{\pi}}e^{\frac{-T^2}{2\sigma_{eff}^2}} - erf\left(\frac{T}{\sqrt{2}\sigma_{eff}}\right). \quad (26)$$

The function *erf()* is the error function.

Signal Only

Now a noise-free input will be considered to determine the minimum detectable color change amplitude. The maximum value of the signal after weighting becomes

$$a\cdot\frac{R}{2J}\left(\frac{1+\beta}{1-\beta}\right)(1-\beta^J), \quad (27)$$

$$b\cdot\frac{G}{2J}\left(\frac{1+\beta}{1-\beta}\right)(1-\beta^J), \quad (28)$$

and

$$c\cdot\frac{B}{2J}\left(\frac{1+\beta}{1-\beta}\right)(1-\beta^J) \quad (29)$$

for each channel. After being squared, summed together, and the square root taken, the maximum signal value becomes

$$S_{max} = \frac{B}{2J}\left(\frac{1+\beta}{1-\beta}\right)(1-\beta^J)\sqrt{\left[(aR)^2+(bG)^2+(cB)^2\right]}. \quad (30)$$

In order to quantify the performance analysis, the color change is parameterized using spherical coordinates. By using spherical coordinates, single color change amplitudes can be defined by one variable. This also allows for easier numerical simulation of the color change. The numerical simulation is discussed in greater detail in the next chapter. The color change in each channel, shown by R, G, and B becomes

$$R = \frac{A}{a}\sin(\theta)\cos(\phi), \quad (31)$$

$$G = \frac{A}{b}\sin(\theta)\sin(\phi), \quad (32)$$

and

$$B = \frac{A}{c}\cos(\theta) \quad (33)$$

where A is defined as the color change amplitude. When (31), (32), and (33) are substituted into (30), the maximum signal amplitude in terms of the color change amplitude becomes

$$S_{max} = \frac{|A|}{2J}\left(\frac{1+\beta}{1-\beta}\right)(1-\beta^J). \quad (34)$$

Note that $|A| = \sqrt{(aR)^2+(bG)^2+(cB)^2}$ involves the weighted step change amplitude from each channel. A color change is said to have occurred when the output of the system, S, has become greater than the threshold. Thus, (34) must be greater than the threshold, which when solved for the amplitude yields the minimum detectable color change amplitude given by

$$|A| > \frac{2JT}{1-\beta^J}\left(\frac{1-\beta}{1+\beta}\right), \quad (35)$$

which is the same result obtained by Malone and Smith [3] for a single channel system.

Signal-to-Noise Ratio

Because of the non-linearities introduced into the system by the use of square law elements, the signal-to-noise ratio is redefined as the maximum signal output in the absence of noise as shown in (34) divided by the standard deviation of output noise shown in (20) and (18) with no signal present which is given by

$$SNR = \frac{|A|}{2\sqrt[3]{(abc)}\sigma_0}\left(\frac{1+\beta}{1-\beta}\right)\frac{1-\beta^J}{\sqrt{J(3-8/\pi)}}. \qquad (36)$$

When (36) is differentiated with respect to J and set to zero it yields

$$\beta^J(1-2J\ln\beta) = 1.0. \qquad (37)$$

This result is the same as obtained by Malone and Smith [3] which was used to determine the value of J that will maximize the signal-to-noise ratio shown in (13). This shows that the signal-to-noise ratio optimized at the output of the moving-average filter will remain optimized after squaring each channel, summing the channels and taking the square root of the resulting sum under the assumption of roughly equal weighting described earlier.

IV. Numerical Results

Once the system was analyzed theoretically, a numerical study was undertaken to verify the system performance and to better understand the effects of varying system parameters. The system was implemented on a PC using code written in FORTRAN.

Simulated Input

The input to the system was generated as a signal with additive noise. The signal portion of the input consisted initially as three arbitrary values corresponding to the quiescent colors in analog video images. The equations shown in (7), (8), and (9) were used to model the signal part of the simulated input to the system. The constants R_0, G_0, B_0 shown are the initial values corresponding to the quiescent values in the red, green, and blue channel. Color change properties were determined by the parameters A, ϕ, and θ as given in (31), (32), and (33). An arbitrary value for ϕ was chosen to yield a small color change for each color change amplitude integer A. The values a, b, and c represent the channel weighting values. The step function portion of these equations, $u(n-n_0)$, shows a step occurrence at n_0. The value for θ depended on whether there would be an overall intensity shift, or whether the input would simulate only a color change with the intensity remaining constant. The value for θ which yields a constant intensity is calculated by (31)-(33):

$$\theta = -\tan^{-1}\left(\frac{1}{\cos(\phi)+\sin(\phi)}\right). \qquad (38)$$

To simulate a decrease in intensity, a value less than the right hand side of the equation was used. A value of θ greater than the right hand side of the equation was used to simulate an increase in intensity.

The noise portion of the simulated input was produced using a Gaussian random number generator. For each time index value, a subroutine that contains the Gaussian random number generator is called. The subroutine uses the specified variance to generate a random number with a Gaussian distribution function. This Gaussian random generator uses two uniformly distributed random number generators and is adapted from a method described by Proakis and Manolakis [4].

The composite system input-output relation shown in (6) was implemented directly. This same difference equation was used for all three channels identically. As was previously discussed, the past input and output values were initialized to yield a zero for the first output value. This was necessary to avoid startup transients at the output and would be necessary in a true application of the system. The system was implemented in floating-point arithmetic.

Simulation Results

The time index ran from 0 to 500. During this index sequence, the sequence according to the previous description for the step portion and additive noise was entered into the system. First, the output of the system was compared to the established threshold. If the output had exceeded this threshold, the time index was then compared to the time index value of the time occurrence, n_0. If the time index n was greater than or equal to n_0, then a counter representing the number of correct detections was incremented. If not, the counter was not incremented. The step occurrence time index, n_0, was varied to produce 250 sequences with different values for n_0. After these 250 sequences were processed, the counter was divided by 250 to

yield a value representing the probability of correct detection for that step change amplitude. This process was repeated for a step-change amplitude from 1 to 175 in steps of 1. The results were plotted as probability of correct detection versus color-change amplitude. These results were obtained for noise with different variances and plotted. Fig. 2 (a) shows plots of the results for a change with an increase in intensity, constant intensity, and a decrease in intensity. The plots show that the results were virtually identical for each of the three cases. The noise added to these signals had a standard deviation of 10. These results were obtained again for noise with a standard deviation of 20, 30, and 40 and are shown in Fig. 2 (b), (c), and (d), respectively. The pole position, β, was set at 0.7951. The number of moving-average terms was chosen to be 5 using (13). By choosing b to be 0.7951 only a small amount of delay is introduced into the system. The noise standard deviation was assumed to be 20. From (35), the minimum detectable amplitude is calculated to be approximately 55 using a threshold value of 33.1. The probability of false detection is less than 10^{-8} for noise having the assumed standard deviation of 20. As the noise standard deviation increases, the probability steadily increases. Using (26), the probability of false detection can be determined for any given noise standard deviation having chosen a threshold. The results show that for low noise, this holds true. As the standard deviation increases, noise effects cause a degradation in performance.

V. Experimental Results

After the system was tested through various numerical simulations, an experimental study was undertaken. The experimental study consisted of three stages. In the first stage the experiment was performed and data was acquired. In the second stage, the data was processed using the system for the second stage. In the third stage, the results of the second stage were interpreted.

Experimental Procedure

An apparatus consisting of a video camera, Bunsen burner, solenoid valve, bubbler, and laser disc recorder was constructed to model the color change phenomenon in an engine plume. The video camera output NTSC format video which was recorded on a laser disc recorder. The video camera was set up so the field of view included only the flame and a small portion of the burner. A compressed air line was split. A solenoid valve was placed on one of the split lines. The valve outlet was fed into a bubbler apparatus. The bubbler bubbles the gas being fed into it through the liquid placed in it. In this experiment the liquid was a salt solution. The air stream exiting the bubbler then had a very small amount of the solution with it. The air exiting the bubbler was rejoined with the other split line to form a single compressed air line again, which was fed into one inlet of the Bunsen burner. This setup allowed the burner to burn normally when the solenoid valve was not open and no solution was in the air stream. A fuel line was fed into the other burner inlet. The fuel gas used for this experiment was methane.

Data Acquisition

After the camera was mounted in a proper viewing location (image composed of mostly flame) and the laser disc had been advanced to a blank frame location, the flame was lit. After the flame had stabilized, recording was begun. At some time after recording had began, the solenoid valve switch was thrown. This allowed the solution to enter the air stream of the Bunsen burner. The approximate frame number corresponding to the switch throw time was noted and recorded. This was repeated several times for salt solutions of various concentrations. Note that the salt concentrations are given as a matter of comparison, not as a measurement of actual concentration at the flame. Because the wavelength emitted by the salt solution is constant, the color of the flame after the solution was introduced also remained constant. Therefore, for different concentrations the color change was constant while the intensity of the color in the flame changed. This is caused by more sodium atoms being excited and emitting photons for the higher concentrations. The recorded sequences show color changes ranging from a slight decrease in overall intensity to a very strong increase in intensity.

Data Processing

After the aforementioned data had been taken, it was processed. The laser disc player outputs color video signals as four channel RGB + sync as well as in the NTSC format. The sync channel signal was combined with one of the RGB color channel signals to produce a RS-170 monochrome video signal. The resulting signal was fed into a PC-based image processor. Software previously developed for use in the leak detection system was modified and used to process the data from each channel separately.

The software operated by implementing the composite system difference equation shown in (6) on

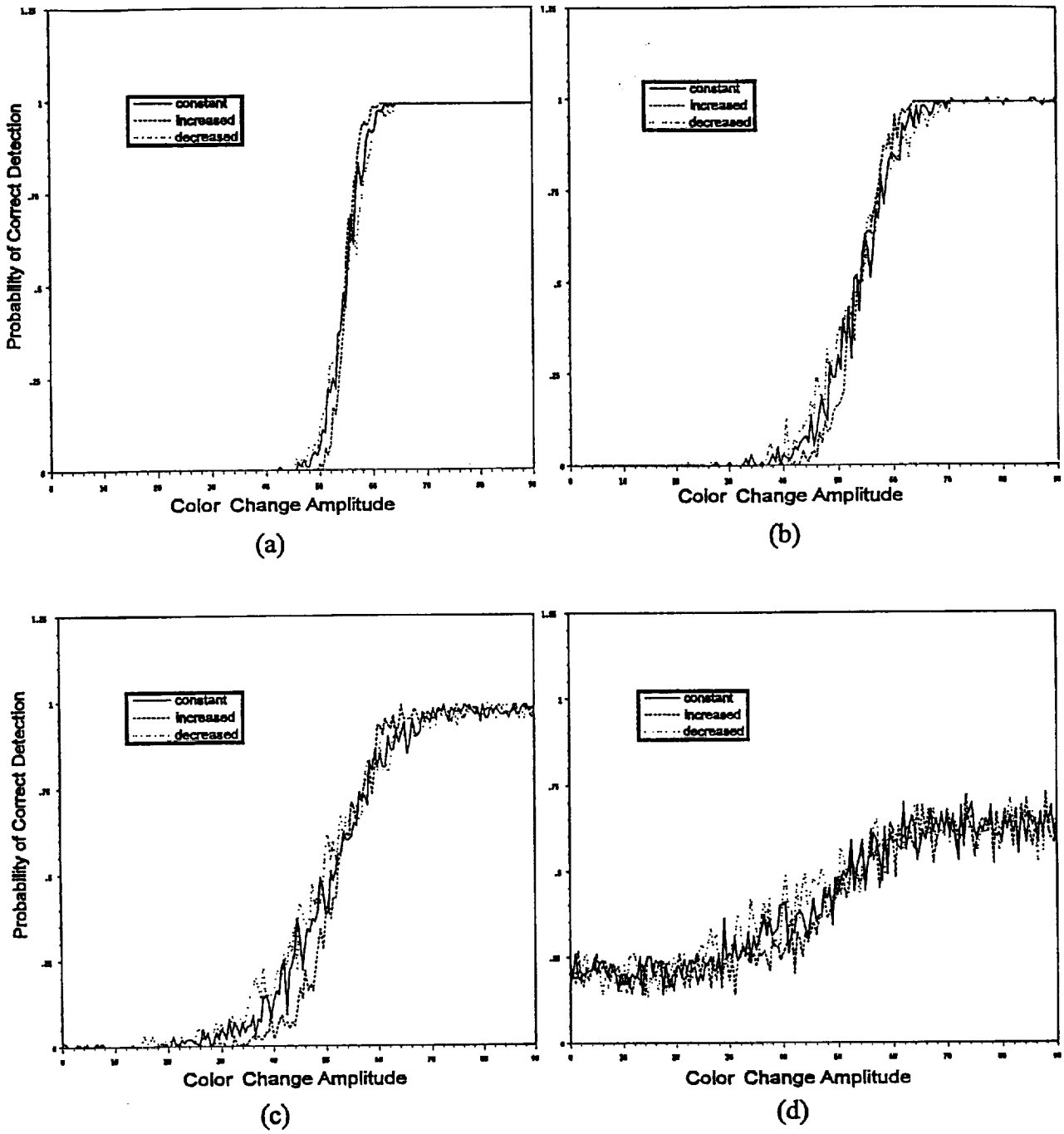


Fig. 2. Results of numerical evalution of the color change detection system.

the entire image sequentially. That is, the difference equation is applied to all pixels simultaneously by means of a full frame (image) arithmetic logic unit (ALU). The result from processing was squared pixel by pixel and the average squared value for the entire image was calculated. This gives an indication of how much of a change is occurring over the entire image. Thus, if a change occurs at only one point from noise, its effect is diminished by the averaging of the full image. This average value is output and stored for each frame of the sequence. This process is repeated for the remaining two color channels. After the processing has been completed, the output average values from each channel are combined by weighting each channel and adding the result. The square root is then taken. The results are output and stored.

Results

Fig. 3 shows a plot of the results of processing one of the recorded sequences using two different sets of weighting values. The solution concentration for this sequence was 5 grams of NaCl per liter of water and the color change occurred at frame 118. This concentration yielded a noticeable change in the intensity of the flame. The pole location β, was set at 0.7951. The plot shows a significant change after this point. The NTSC weighting values used to process the previous results were not optimum. Because the solution introduced into the flame was a NaCl salt solution, the wavelengths of light emitted by the flame are known. From the tables in [1], sodium emits light in the visible spectrum at 5890Å. Using the camera response values at this wavelength, a weighting function of 0.48, 0.48, 0.04 for the red, green, and blue channels respectively, was selected. The results show that before a color change has occurred, using the selected values decreases the output level a small amount. After the color change occurs, however, using the selected weighting values increases the signal output level. These results show that if the color expected (or elements expected are known) after a color change is known weighting values can be chosen to optimize detection.

VI. Summary and Conclusions

A system has been developed to detect color changes in sequences of video data for implementation in real-time for the application of monitoring the exhaust of the space shuttle main engine (SSME). The system consists of a highpass filter cascaded with a moving-average filter applied to each channel (red, green, blue) video data. The results of the filtering in each channel are weighted, squared, and the three channels are summed together to produce a single-time varying output. This output is compared to a threshold value to determine if a color change has occurred.

A numerical study was conducted to test the system for a variety of noise and color change conditions. The time of the color change occurrence was varied along with the noise conditions to produce a plot of correct detection probability versus color change amplitude. 'Correct detection' was said to have occurred if the system output exceeded the set threshold after the color amplitude change. This plot was produced for noise with a standard deviation of 10, 20, 30, and 40. The system parameters were set assuming the input noise had a standard deviation of 20. The results show that for low noise inputs the system probability of correct detection is approximately one for amplitude changes above the minimum detectable color change amplitude. As the noise standard deviation increased to 40, the probability of correct detection above the minimum detectable step amplitude was not much different than the correct detection probability below the minimum.

An experimental study was undertaken to test system performance on a color change in a flame. A Bunsen burner burning a mixture of methane and compressed air was used as the flame source. A branch of the compressed air stream was passed through a salt solution at the throw of a switch to cause the flame color change. The flame was recorded by a video camera and stored on laser disc. The salt solution concentration was varied to obtain video sequences with varying color change intensities. The system was implemented on a PC-based image processor where the video sequences were processed a frame at a time. The results of the processed sequences were plotted. The results show that system can detect color changes accurately, even though the laboratory had substantial noise conditions. Selecting weighting values based on the expected color change improved performance. A criterion for selecting these weighting values was not determined and is an area for further research.

Considering the conditions during engine firing, some steps would have to be undertaken to improve the system operation. The field of view of the camera should contain only plume and as close up as possible. This ensures that every pixel is being utilized to detect color changes. The camera should also be placed in such a position, if possible to ensure, that as

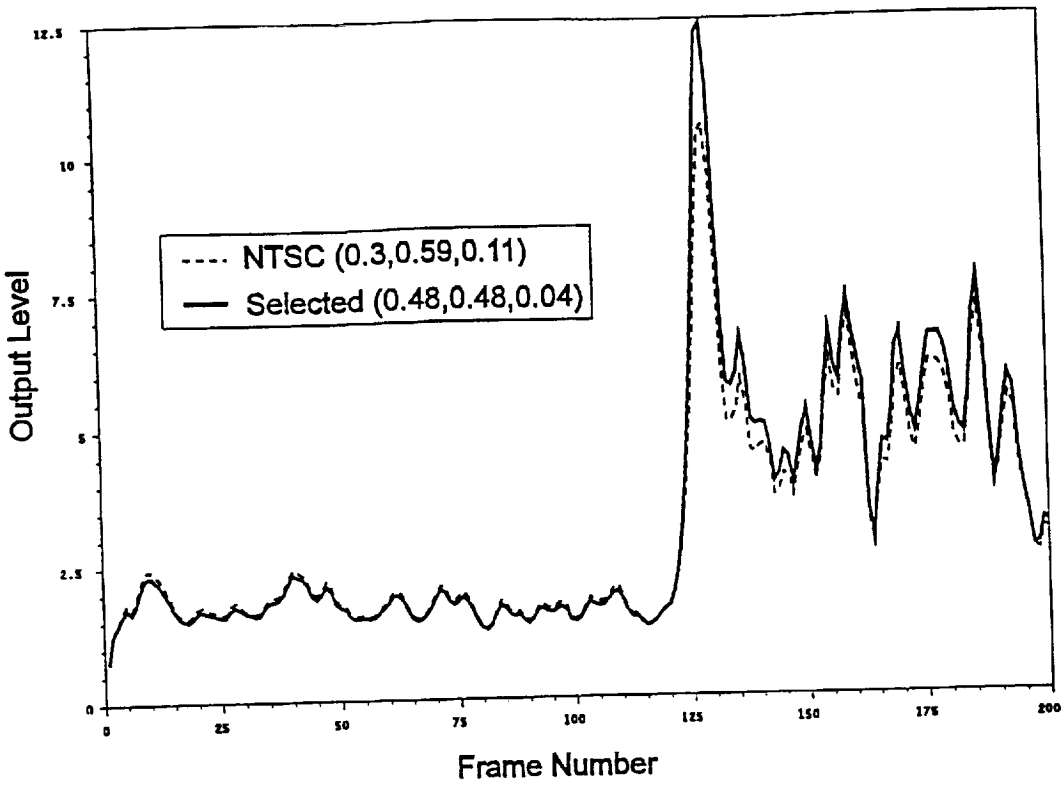


Fig. 3. Experimental Results for 5 gram/liter salt solution. Plots for NTSC weighting values and selected values are shown for comparision.

little steam and other debris as possible should pass through the field of view since this causes massive changes in the images which in turn cause false alarms.

Another possible method to improve system performance is to incorporate a median filter. As of now, the mean change was calculated for the entire frame. By determining the median value, any large changes caused by noise are minimized.

References

1. Mavrodineanu, Radu and Boiteux, Henri, *Flame Spectroscopy*, Wiley, New York 1965.

2. Malone, JoAnne, *A System for Leak Detection Using Sequential Image Processing*, Masters Thesis, University of Tennessee Space Institute, August 1991.

3. Malone, JoAnne and Smith, L. Monty, "A System for Sequential Step Detection with Application To Video Image Processing," *IEEE Transactions on Industrial Electronics*, vol. 39, no. 4, pp. 277-284, August 1992.

4. Proakis, John G. and Manolakis, Dimitris G., *Digital Signal Processing(Principles, Algorithms, and Applications)*, Second Edition, MacMillan, New York, 1992