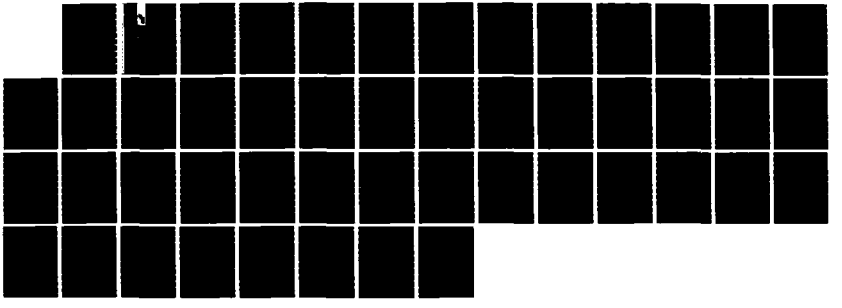
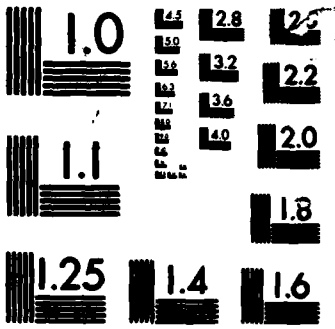
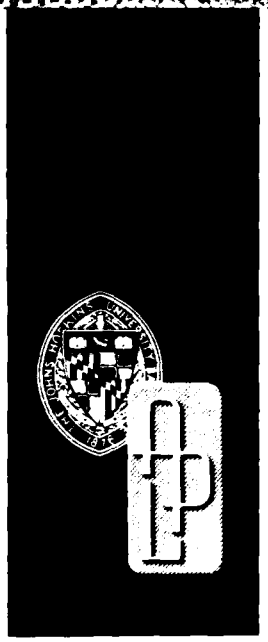
AD-A168 628 AN ERROR DETECTION AND SMOOTHING ALGORITHM FOR INFRARED 1/1
DATA(U) JOHNS HOPKINS UNIV LAUREL MD APPLIED PHYSICS
LAB R A STEINBERG ET AL. APR 86 JHU/APL/TG-1355
UNCLASSIFIED N00024-85-C-5301 F/G 17/5 NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
MICROCOPY
CHART

JHU/APL
TG 1355
APRIL 1986
Copy No.



Technical Memorandum

AN ERROR DETECTION AND SMOOTHING ALGORITHM FOR INFRARED DATA

R. A. STEINBERG*
M. J. McHUGH†

*Formerly of the Naval Research Laboratory
†Sachs-Freeman Associates, Inc.

THE JOHNS HOPKINS UNIVERSITY ■ APPLIED PHYSICS LABORATORY

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

AD-A168628

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE n/a	
4. PERFORMING ORGANIZATION NUMBER(S) JHU/APL TG-1355	5. MONITORING ORGANIZATION REPORT NUMBER(S) JHU/APL TG 1355
6a. NAME OF PERFORMING ORGANIZATION The Johns Hopkins University Applied Physics Laboratory	6b. OFFICE SYMBOL (If Applicable)
7a. NAME OF MONITORING ORGANIZATION NAVPRO, Laurel, Maryland	
6c. ADDRESS (City, State, and ZIP Code) Johns Hopkins Road Laurel, Maryland 20707	7b. ADDRESS (City, State, and ZIP Code) Johns Hopkins Road Laurel, Maryland 20707
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Naval Research Laboratory	8b. OFFICE SYMBOL (If Applicable) 6520
9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00024-85-C-5301	
8c. ADDRESS (City, State, and ZIP Code) 4555 Overlook Ave. SW Washington, DC 20375-5000	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. ZG50 / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
An Error Detection and Smoothing Algorithm for Infrared Data

12. PERSONAL AUTHOR(S)
R. A. Steinberg and M. J. McHugh

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Technical Memorandum	FROM _____ TO _____	1986, April	45

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS
FIELD	GROUP	SUB-GROUP	

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

A new algorithm for error detection and smoothing, EDS, has been developed for infrared image data. Numerical validation indicates that EDS detects more than 99% of all significant errors. Operating in the time domain, EDS exploits smoothness criteria calculated from laboratory point source measurements, and isolates error bursts via a region-growing approach. EDS is applicable to data from any optical scanner whose output is sampled digitally multiple times per dwell.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION	
X UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL Robert W. Irvin	22b. TELEPHONE (Include Area Code) (301) 953-5000	22c. OFFICE SYMBOL NPRO

DD FORM 1473, 84 MAR

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

JHU/APL
TG 1355
APRIL 1986

Technical Memorandum

AN ERROR DETECTION AND SMOOTHING ALGORITHM FOR INFRARED DATA

R. A. STEINBERG*
M. J. McHUGH†

*Formerly of the Naval Research Laboratory
†Sachs-Freeman Associates, Inc.

THE JOHNS HOPKINS UNIVERSITY ■ APPLIED PHYSICS LABORATORY
Johns Hopkins Road, Laurel, Maryland 20707
Operating under Contract N00024-85-C-5301 with the Department of the Navy

THE JOHNS HOPKINS UNIVERSITY
APPLIED PHYSICS LABORATORY
LAUREL, MARYLAND

ABSTRACT

A new algorithm for error detection and smoothing, EDS, has been developed for infrared image data. Numerical validation indicates that EDS detects more than 99% of all significant errors. Operating in the time domain, EDS exploits smoothness criteria calculated from laboratory point source measurements, and isolates error bursts via a region-growing approach. EDS is applicable to data from any optical scanner whose output is sampled digitally multiple times per dwell.

CONTENTS

ILLUSTRATIONS

ILLUSTRATIONS (Cont'd)

TABLES

THE JOHNS HOPKINS UNIVERSITY
APPLIED PHYSICS LABORATORY
LAUREL, MARYLAND

1. INTRODUCTION

The Navy's Background Measurements and Analysis Program (BMAP) was recently begun to obtain the data needed to design surface- and air-based infrared search and track (IRST) devices.[1,2] The data are intended to be a test set of infrared (IR) background images for use in off-line simulation of alternative signal processing techniques for false-alarm suppression. The IR backgrounds data are acquired with a sensor that has high spatial resolution, high sensitivity, and simultaneous 4 to 5 μm and 8 to 11 μm coverage.[3]

Initial IR clutter measurements were obtained at Montauk Point, Long Island, N. Y.[1-5] Unfortunately, the initial measurements were contaminated by a malfunctioning digital recorder with about 4% defective data (3% small-amplitude errors and 1% large-amplitude errors). Thus, each data frame of 6400 samples contains (on the average, prior to data reduction) about 60 large-amplitude errors, i.e., errors of amplitude 10 times the sensor noise or greater. Visual inspection of the data shows that the errors tend to occur *in bursts, i.e., as multiple sequential errors.*

Subsequent to the Montauk Point measurements there was a hiatus in field activities during which software techniques were developed for extracting a limited interim data set for use until better data were obtainable.

The need to recognize "wild points" or "outliers" in data is a requirement of data reduction in all experimental sciences, including, for example, chemistry, speech analysis, spectroscopy, and geophysics.[6-10] The Montauk data analysis is unusual relative to other outlier identification problems in that the Montauk data contain a large number of multiple sequential errors. On the average, each frame of Montauk data contains one error burst of 10 (or more) sequential errors, as well as many error artifacts of lesser length (Table 1).

Table 1
Number of extended artifacts (i.e., multiple errors) in an average frame of Montauk Point data. By "length of artifact" is meant the number of sequential defective samples in an error burst. Thus, an artifact of length 8 is an error burst consisting of 8 sequential defective samples. Each data frame contains 6400 samples.

Length of artifact (samples), L	6	7	8-9	≥ 10
Average number of artifacts of length L (per frame)	1	1	1	1

Identification of multiple sequential and nearly sequential errors is more difficult than identification of isolated single-sample errors. Figure 1 illustrates the difficulties experienced in using a commonly available outlier code, the IMSL routine ICSMOU,[6] against a pair of closely spaced single-sample errors. In the simple example of Fig. 1, ICSMOU damages a good data point. Figure 2 illustrates a compressed notation for representing the results of algorithm performance on short sequences of synthetically damaged data. The single graph in Fig. 2 conveys the same information as the three graphs in Fig. 1.

[1] R. L. Lucke, A. P. Schaum, J. C. Kershenstein, J. Michalowicz, B. V. Kessler, A. B. Blumenthal, I. Goldstein, and A. Krutchkoff, "The Navy's Infrared Background Measurement and Analysis Program," *Proc. IRIS Specialty Group Meeting on Targets, Backgrounds, and Discrimination,* San Diego, 11-14 Feb. 1986.

[2] R. A. Steinberg, "Navy IR Background Measurements and Analysis Program," *Proc. Tri-Services Infrared Backgrounds Symp.*, Oct. 18-20, 1983, Mitre Corporation, pp. 216-223.

[3] M. S. Longmire, "A Final Technical Report on Calibration and Use of Clutter Data for Simulation," Western Kentucky University, Bowling Green, Ky., Contract No. N0014-84-C-2034 (undated draft report to Naval Research Laboratory).

[4] A. Hirschman, "BMAP Surface-Based Background Measurement Activities at Montauk Point, New York, August 1983," Naval Surface Weapons Center, White Oak, Silver Spring, Md., draft report, Nov. 1984.

[5] D. R. Jensen, "Data Report for IR BMAP Experiment, Aug. 1983," Naval Ocean Systems Center, San Diego, Cal., Memorandum DRJ:plk, Serial 5325/109, 14 Nov. 1983.

[6] J. R. Rice, *Numerical Methods, Software, and Analysis,* McGraw-Hill, New York, 1983.

[7] H. Ney, "A Dynamic Programming Technique for Nonlinear Smoothing", *Proc. IEEE International Conf. Acoustics, Speech and Signal Processing, Vol. 1,* Mar. 30-31 and Apr. 1, 1981, Atlanta, Ga., pp. 62-65.

[8] H. Ney, "A Dynamic Programming Algorithm for Nonlinear Smoothing," *Signal Processing,* Vol. 5, No. 2, Mar. 1983, pp. 163-173.

[9] F. Pasian and A. Crise, "Restoration of Signals Degraded by Impulse Noise by Means of a Low-Distortion Non-Linear Filter," *Signal Processing,* Vol. 6, No. 1, Jan. 1984, pp. 67-76.

[10] R. E. Boucher and J. P. Noonan, "Adaptive Detection and Removal of NonGaussian Spikes from Gaussian Data," *IEEE Trans. Pattern Analysis and Machine Intelligence,* Vol. PAMI-4, No. 2, Mar. 1982, pp. 132-136.

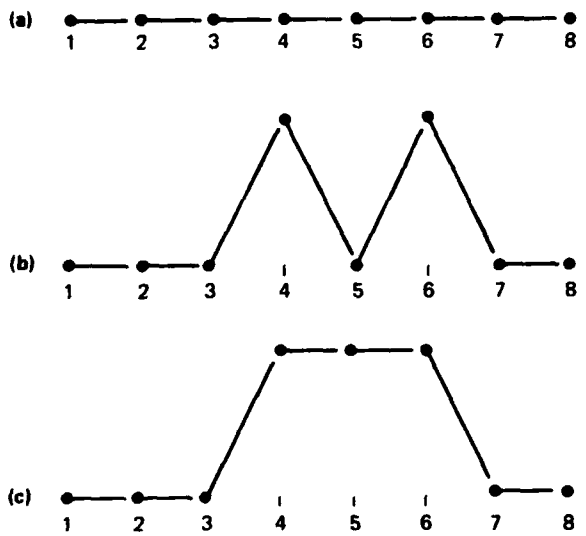


Figure 1. Inadequate treatment of multiple errors can make the data worse. (a) Original data (constant value). (b) Damaged data, with outliers at samples 4 and 6. (c) "Corrected" data, after application of IMSL routine ICSMOU, designed for generic wild point removal applications.[6] ICSMOU fails to identify data errors at samples 4 and 6, and at sample 5 damages what originally had been a good data value.

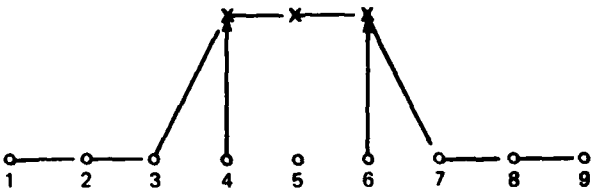


Figure 2. Illustration of compressed notation for representing the results of algorithm performance on short sequences of synthetically damaged data. The single graph in this figure conveys the same information as the three graphs in Fig. 1. O's are original undamaged data; vertical arrows indicate data errors that we intentionally implant; X's are values after application of correction algorithm (ICSMOU in this instance). Where X and O coincide, only O is drawn. Where X falls at the tip of an arrowhead (e.g., samples 4 and 6) the correction software has failed to recognize a data error. Where X appears at a sample with no vertical arrow (i.e., with no error), correction software has damaged a good data point (e.g., sample 5). When X falls anywhere other than at the tail of an arrow, correction software has identified the data sample as an error but has generated an imperfect correction (not illustrated on the figure). The dashed line is the "current waveform," i.e., the waveform as it would appear after the last processing operation performed on the data.

A modified median filter approach[9] and nonlinear smoothing by dynamic programming[7,8] both appear to hold some promise. However, both these techniques require setting thresholds and selecting smoothness parameters that depend in a non-obvious way on the character of the data. The original error detection technique described in this report is based on exploiting smoothness criteria that derive directly from the known point response characteristics of the infrared scanner. The values ascribed to the smoothness parameters are simply calculated from measurements made against a laboratory point source, and do not depend on the highly variable structure of the IR clutter data viewed in the field.

The computation is structured such that the data are pre-screened with a simple algorithm that allows further attention to be concentrated only on "suspect" neighborhoods. There are no computation-intensive steps, such as FFT's or histograms, required for the data set as a whole. Execution time on a VAX 11/780 was about 10 times faster than ICSMOU.

Outlier identification and replacement is the first step in the IR clutter data reduction process. The second step in data reduction involves correction of a structural distortion in the longwave data caused by resistive coupling between the longwave detector elements.[11,12] The remaining steps in data reduction include stagger offset compensation, responsivity nonuniformity compensation, and radiometric calibration.[3]

[11]R. A. Steinberg, "Infrared Background Sensor Characterization and Data Reduction," IRIS Specialty Group on Targets, Backgrounds, and Discrimination, Feb. 1-2, 1984, Camp Pendleton, Cal., pp. 209-231.

[12]R. A. Steinberg, "Elimination of Sensor Artifacts from Infrared Data," Report 8861, Naval Research Laboratory, Washington, D.C., 11 Dec. 1984.

2. OUTLIERS APPEAR IN THE DATA

Prior to the field tests at Montauk Point, laboratory bar-target and point-target measurements were obtained and analyzed to characterize the IR scanner's transient response and radiometric sensitivity.[1] Figure 3 shows a sample of this data, obtained against a laboratory target consisting of six vertical bars of progressively increasing width.

The scan line shown in Fig. 3 contains three single-sample errors, a large positive-going error at sample 315, a very large positive-going error at sample 422, and a small negative-going error at sample 433. Sensor closed-cover noise is about one count, RMS; major y-axis divisions are 500 counts, as shown on the figure. Thus, none of the waveform structure visible in Fig. 3 is attributable to sensor closed-cover noise.

The narrowest bar, left-most in Fig. 3, is spatially unresolved. The physically smallest possible optical source, viz., an ideal point source, induces a pulsed-shaped waveform ("the point response") nearly identical to this narrowest bar. Any pulsed-shaped waveform narrower than the point response must be an artifact of electrical, rather than optical, origin. Thus, applying a simple pulse-width criterion, single-sample errors are detected easily both by eye and by algorithm. Intuitively, the key to simple identification of single-sample errors lies in prior knowledge of the system point response.

Figure 4 depicts a scan line of data from Montauk Point, obtained looking up from the surface against a cloud/sky background. The data shown in Fig. 4 contain about three times as many multiple errors as an average scan line: of the 400 data samples that comprise the waveform, 24 are multiple errors. Three badly damaged data strings, each coincidentally of 11 samples duration, are found to contain 26 of the 30 total errors. The first badly damaged 11-sample interval, centered on sample 171, contains 7 errors and 4 good data points (Fig. 5a). The second error burst, centered on sample 295, contains 8 errors and 3 good data points (Fig. 5b). The third and final error burst, centered on sample 422, contains 11 errors and no good data points (Fig. 5c).

The remaining 4 errors in the scan line are each isolated single-sample errors. These isolated outliers, once identified (detected), can be replaced by interpolation without significantly degrading data quality, due to the high data sampling rate (3.47 samples per dwell).

Although the IR clutter data within the 3 multi-error regions cannot be reconstructed, this only marginally affects the utility of the data. Once the 3 error bursts are identified and excised, the remaining data may then be used for a variety of IRST processor simulation studies.

We note that our notation on "zoom" plots of errors found in field test data (e.g., Fig. 5) is different

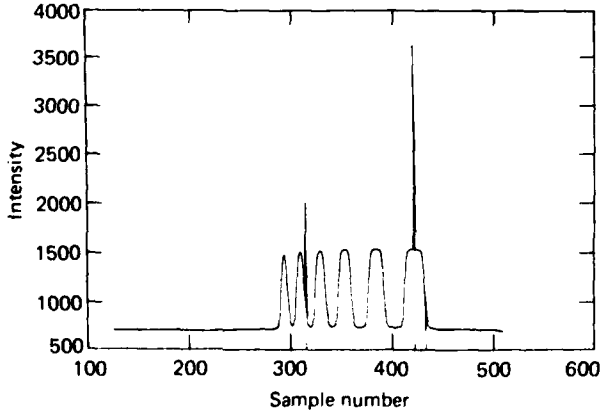


Figure 3. Single scan line (midwave channel 15) obtained against a laboratory target consisting of six vertical bars of progressively increasing width. Single-sample errors occur at sample numbers 315, 422, and 433. A vertical tick mark above the abscissa indicates the location of each data error.

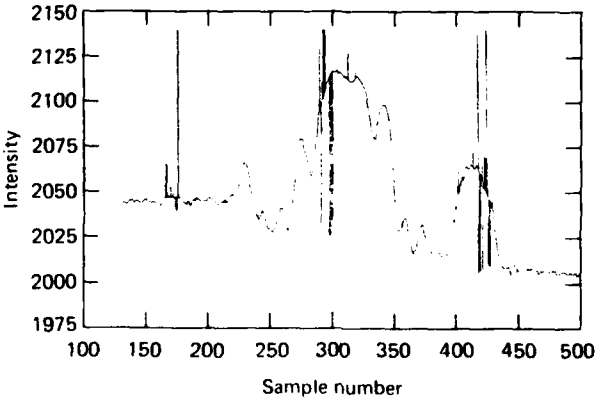


Figure 4. Scan line of badly damaged data. The data shown in this figure contain about four times as many errors as an average scan line: of the 400 data samples that comprise the waveform, 24 are multiple errors. Vertical lines connect data errors to interpolated values on the waveform.

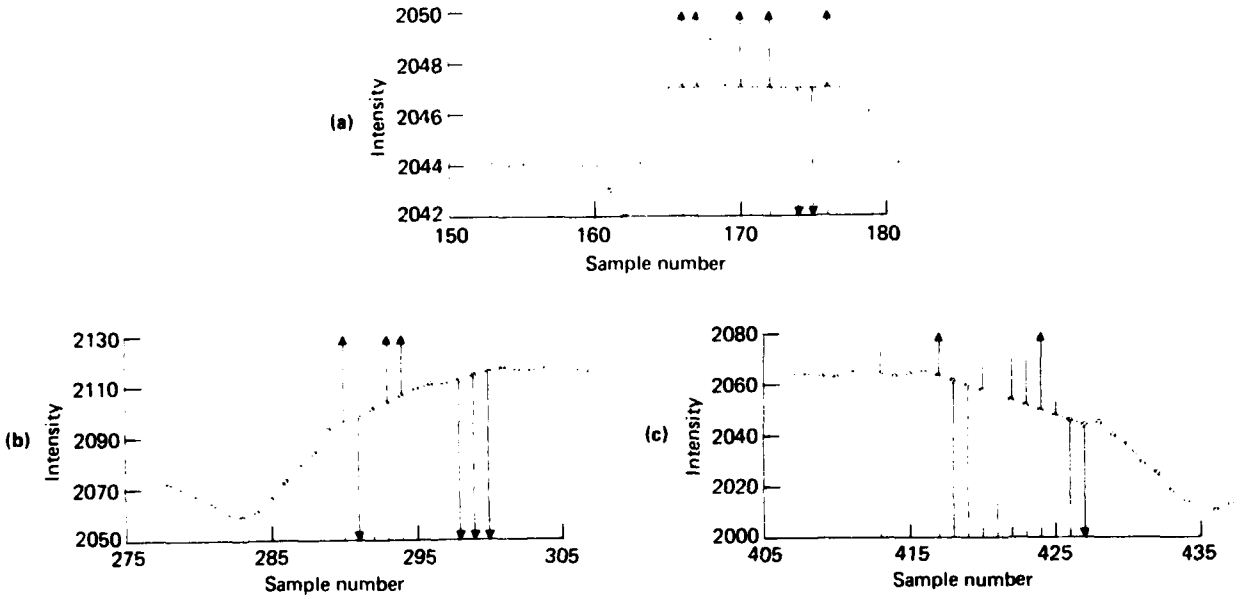


Figure 5. Expanded view of the three error burst intervals in Fig. 4. Circles not connected to vertical lines indicate valid/undamaged data points. Vertical lines connect interpolated values (indicated by circles) to damaged data points. Arrowhead indicates that error value is off-scale. (a) 11-sample interval centered on sample 171 contains 7 errors. (b) 11-sample interval centered on sample 295 contains 8 errors. (c) 11-sample interval centered on sample 422 contains 11 errors.

than for data into which we ourselves have implanted errors for algorithm evaluation purposes ("synthetic errors", e.g., Fig. 2). Our notation for error detection in field test data, illustrated in Fig. 5, is as follows:

- Vertical lines indicate samples identified as errors by the EDS software.
- Open circles attached to vertical lines are interpolated values, i.e., replacement values for identified errors.
- Unattached ends of vertical lines are original, presumably defective, values.
- Arrow heads indicate off-scale error values.
- Open circles not attached to vertical lines indicate "good" measured data.

3. HOW RELIABLY ARE THE ERRORS DETECTED?

The example of Figs. 4 and 5 illustrates the capability of our new error detection algorithm for isolating long sequences of damaged data. In this section we present the results of a validation analysis intended to address the following statistical aspects of algorithm performance:

- What fraction of data errors elude detection by EDS? How many of the undetected errors are significant, in the sense of having large amplitude?
- What fraction of the good data is damaged by EDS?
- For those data errors that are correctly identified as errors by EDS, how accurate are the applied corrections?

Our approach to statistical validation was first described in broad outline in Ref. 2.

The validation process starts with the selection of a data set A that is error-free by the standard of the software error detection and smoothing (EDS) operator, θ, represented as

$$A = \theta(A), \qquad (1)$$

i.e., operating on data set A with the EDS software leaves the data set unchanged.

The data set A used for analysis consisted of 800 scan lines, such as that plotted in Fig. 6.

We numerically generate and add to A a set, S, of error values, in which every nonzero component represents a synthetic data error of known location and amplitude (Fig. 7). The result is a synthetically corrupted data set B (Fig. 8),

$$B = A + S. \qquad (2)$$

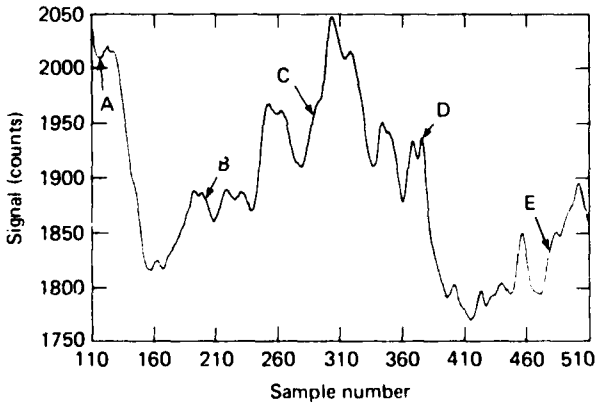


Figure 6. An error-free scan line. 800 such error-free lines of data (collectively referred to as data set A) are used in connection with the statistical validation analysis of the error detection algorithm. The significance of regions labeled A to E is discussed in the text.

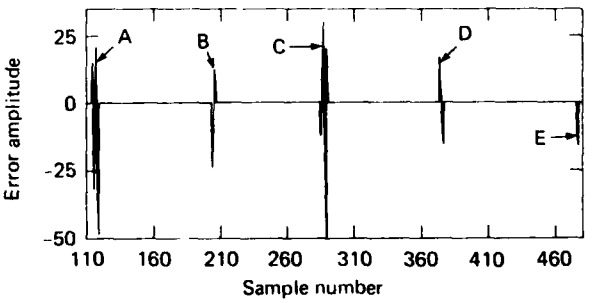


Figure 7. Sample scan line of synthetic data errors, containing two five-fold errors (A and C), two double errors (B and D), and an isolated error (E). The error generation algorithm was designed to provide error statistics specified with respect to an amplitude profile (Fig. 9) and the number of error bursts of a given duration (Fig. 10). The ensemble of 800 such error lines is referred to in the text as data set S.

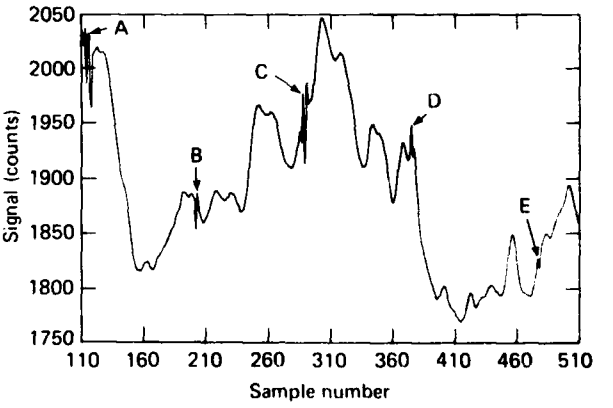


Figure 8. A synthetically corrupted scan line obtained by adding synthetic data errors (Fig. 7) to an error-free scan line (Fig. 6). Error bursts, i.e., multiple sequential errors, are injected at positions denoted as A, B, C, and D, on the waveform; a single error was injected at position E. The multiple error at D occurs on a peak, as seen by comparison with the corresponding region D in Fig. 6.

The error set, S, is described by two histograms (Figs. 9 and 10) that statistically characterize the synthetic errors with respect to amplitude and spatial extent, and that, taken together, constitute a model for the error generation process.

Each data frame used in the statistical validation had 6400 samples:

16 scan lines per frame × 400 samples per line
= 6400 samples per frame.

Fifty statistically independent error frames were numerically generated, containing a total of 11,470 data errors. It follows that the overall error rate is:

$$\text{error rate} = \frac{(11{,}470 \text{ defective samples})}{6400 \text{ samples per frame} \times 50 \text{ frames}}$$
$$= 0.0358 = 3.58\%.$$

Figure 9 shows how the 11,470 synthetic errors were distributed in amplitude. For example, 8458 errors (about three-fourths of the total) had amplitude values between 1 and 9 counts. The number of errors of amplitude ≥ 10, as a fraction of the total data set, is thus given by:

$$[(11{,}470 - 8{,}458)/320{,}000] = 0.94\%.$$

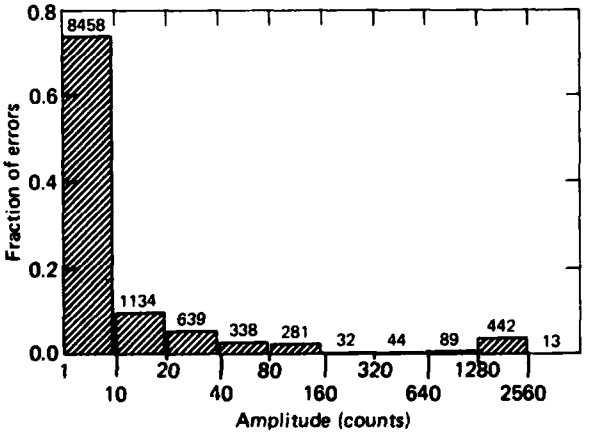


Figure 9. Amplitude histogram of 11,470 synthetically generated errors, dispersed into 50 frames of data (overall error rate = 3.58%). About three-fourths of the errors have values from 1 to 9 counts in amplitude. The data error sign was modeled as statistically independent of the error amplitude, and was assigned positive and negative values with equal likelihood. The synthetic errors are generated by a program that closely matches their amplitude histogram, shown here, to the amplitude histogram of errors found in Montauk Point data.

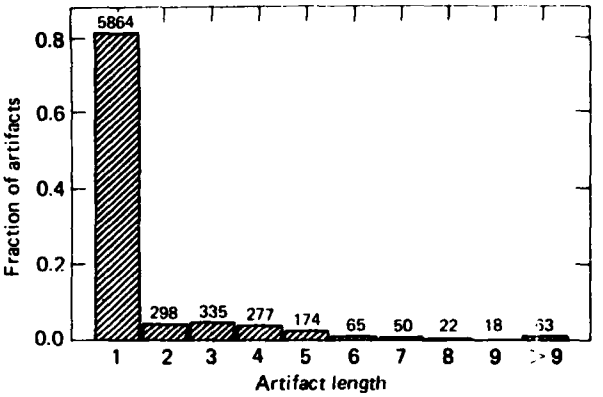


Figure 10. Artifact length histogram of synthetically generated errors. A total of 7166 error artifacts (i.e., error bursts) were dispersed into 50 data frames. As indicated by the left-most bar in the figure, about 82% of the artifacts were single-sample errors. The synthetic errors are generated by a program that closely matches their artifact length histogram, shown here, to the artifact length histogram of errors found in Montauk Point data. Figures 9 and 10 together provide a characterization of the Montauk data as well as a model for the synthetic error generation process.

We consequently speak of the data as having suffered about 1% significant damage.

In this report a group of contiguous errors bounded on each side by a good data value is referred to either as an error burst or as an error "artifact," with the artifact length being defined as the number of sequential errors. The 11,470 errors in error set S were grouped into a total of 7166 artifacts, with the distribution of lengths as shown in Fig. 10. For example, Fig. 10 indicates that 5864 artifacts (82% of the total) were of unit length, i.e., were isolated single-sample errors.

When our EDS algorithm is applied to the synthetically damaged waveform shown in Fig. 8, the smoothed waveform shown in Fig. 11 is obtained. Comparing visually the original and smoothed waveforms (Figs. 6 and 11, respectively) we note that all regions of data damage have been identified and, with varying degrees of success, corrected. The multiple errors at B and C and the single error at E have been corrected with high accuracy. The multiple error at A has been imperfectly corrected, with the result that the trough present at A in the original data (Fig. 6) has been sharpened somewhat in the corrected data (Fig. 11). The largest residual effect of the damage/smoothing process appears at location D. The peak originally present at D

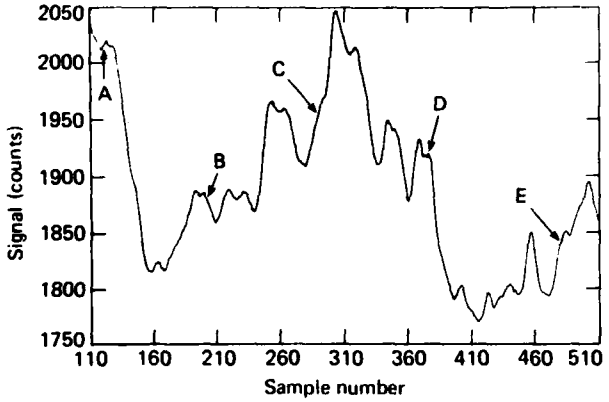


Figure 11. Smoothed scan line obtained by applying Algorithm EDS to the synthetically damaged scan line depicted in Fig. 8. Neighborhoods labeled A to E are compared in the text of the report with corresponding neighborhoods in Fig. 6.

in Fig. 6, badly damaged by a multiple error (Fig. 8), has been sliced away by EDS (Fig. 11). The data errors at D were successfully detected but badly corrected.

Since restoration of the true data structure is generally not possible within regions where multiple errors have occurred, the EDS algorithm is better described as providing error detection and censoring, rather than error correction, within the "blind spots" created by multiple errors.

Considering the example of Figs. 6, 8, and 11, the net effect of data damage/smoothing has been that one structural feature in the original data has been lost (D), another local feature has been somewhat modified in structure (A), and all other features have been either repaired or left unchanged in their original undamaged condition. Whether the smoothed data, Fig. 11, provides a sufficiently accurate representation of the original data, Fig. 6, is apparently application-dependent.

We next consider error correction statistics established over 800 scan lines (i.e., 50 frames) of synthetically damaged data. In the terminology of Eqs. 1 and 2, we operate on the synthetically corrupted data set, B, with algorithm θ,

$$\hat{A} = \theta(B) , \tag{3}$$

to obtain an estimate, $\hat{A}$, for the original error-free data set, A. In our numerical study, an ensemble of 800 corrected waveforms (e.g., Fig. 11) constitutes $\hat{A}$.

Our numerical study uses data sets A, S, and $\hat{A}$ that each consist of 50 frames, i.e., 320,000 values. Equivalently, we may consider corresponding elements in these data sets as forming 320,000 3-tuples, $(a,s,\hat{a})$, where a denotes a single element of A (and similarly for s and $\hat{a}$). A test is performed on each 3-tuple, with four possible outcomes, as indicated in Table 2.

Considering an Outcome 2 result, for example, we see from Table 2 that this corresponds to the following conditions on the 3-tuple $(a,s,\hat{a})$:

$$s = 0 \tag{4a}$$

$$\hat{a} \neq a \tag{4b}$$

Equation 4a indicates that the data sample in question has been modified with a zero-amplitude error, i.e., the data sample is a valid data point. Equation 4b indicates that the processed data value, $\hat{a}$, is not equal to the original data value, a. The interpretation of the Outcome 2 result is, thus, straightforward: a good data value has been incorrectly identified as an error by the EDS algorithm and modified in amplitude, i.e., "damaged," by the amount $\hat{a} - a$.

Of the 320,000 trials, an Outcome 2 result is obtained 618 times, i.e., 618 valid data points are damaged. The 618 corresponding values of $|\hat{a} - a|$

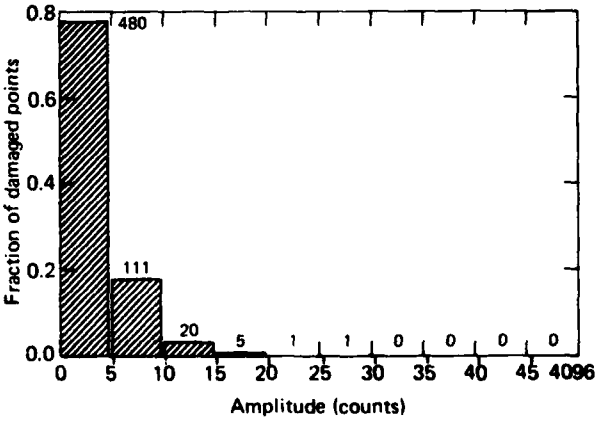


Figure 12. Amplitude histogram of damaged points. Of the 320,000 sample values (i.e., 50 frames) in the validation data set A, 618 valid data points are incorrectly classified as errors and are subsequently modified in amplitude ("damaged") by the EDS Algorithm. The leftmost bar of the histogram indicates that 480 of the damaged samples (78%) were damaged by 4 or less digital counts. Good data values damaged by more than 19 counts were located at the edge of an error burst.

are depicted as a histogram in Fig. 12. If, for a given application, data damage of less than 10 counts is deemed insignificant, it follows from Fig. 12 that 27 data points have been "significantly damaged" by the EDS algorithm.

Table 2
A test is performed on each 3-tuple, $(a,s,\hat{a})$ with four possible outcomes. Since each data set, (A, S, and $\hat{A}$)contains 320,000 values (corresponding to 50 frames), this test was performed 320,000 times. The values of $|s|$ corresponding to outcome 1 (missed errors) were used to assemble the histogram in Fig. 13. Similarly, values of $|\hat{a} - a|$ corresponding to outcomes 2 and 3 were used to assemble Figs. 12 and 14, respectively.

Outcome	s	$\hat{a} - a$	Interpretation	Number of Occurrences
1	$\neq 0$	s	Bad data has not been detected	2,839
2	0	$\neq 0$	Good data has been damaged	618
3	$\neq 0$	$\neq s$	Data error has been detected	8,631
4	0	0	Good data has been left unchanged	307,912

The interpretation of Table 2 for Outcome 1 is only slightly less obvious than Outcome 2. According to Table 2, Outcome 1 corresponds to

$$s \neq 0 \tag{5a}$$

$$\hat{a} - a = s \tag{5b}$$

From Eqs. 5b and 2,

$$\hat{a} = a + s = b\,, \tag{6}$$

where b is a sample from the damaged data set, B.

From Eqs. 5a and 6 it follows that Outcome 1 corresponds to

$$s \neq 0 \tag{7a}$$

$$\hat{a} = b\,. \tag{7b}$$

Equation 7a indicates that the data sample in question has been modified by a nonzero value, i.e., the data sample is an error. Equation 7b indicates that the processed data value, $\hat{a}$, is exactly equal to the defective data value b, i.e., no correction has been performed. Thus, our interpretation of an Outcome 1 result is that a data error of amplitude s has propagated unchanged through EDS, i.e., the error has been "missed."

Of the 320,000 trials, an Outcome 1 result is obtained 2839 times, i.e., 2839 errors are missed. The 2839 corresponding values of $|s|$ are depicted as a histogram in Fig. 13. If, for a particular application, data errors of less than 10 counts are deemed insignificant, it follows from Fig. 13 that 24 "significantly large" errors have not been detected by EDS. However, we see from Fig. 9 that 3012 errors were implanted with amplitudes ≥ 10. Thus, the fraction of significantly large errors not detected by EDS is (24/3012) = 0.8%; slightly better than 99 out of each 100 large amplitude errors are detected.

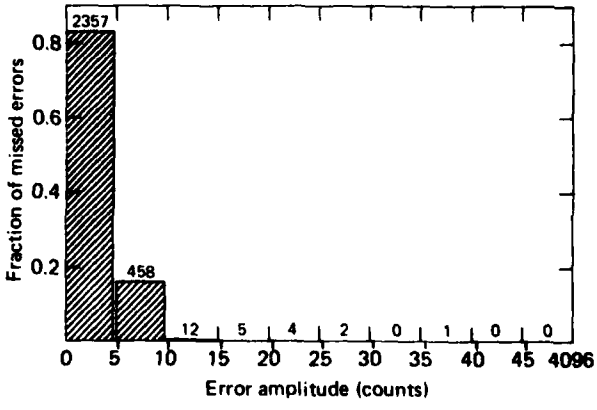


Figure 13. Amplitude histogram of undetected errors. Of the 11,470 errors implanted into 50 frames of data, 2839 errors are undetected by EDS. However, only 24 of the missed errors were of significant amplitude, i.e, amplitude 10 or larger. Thus, after correction there were, on average, 0.5 significant undetected errors per frame. The largest-amplitude undetected error (amplitude 39) occurred at the edge of an error burst.

Similarly, it may be shown that a Table 2/Outcome 3 result corresponds to a trial in which a data error has been correctly identified as an error by EDS, and a correction has been applied to the data. Of the 320,000 trials, Outcome 3 occurs 8631 times. The 8631 corresponding values of $|\hat{a} - a|$ are represented as a histogram in Fig. 14, indicating the accuracy of data correction for those data errors correctly identified as errors by EDS. For example, if correction errors of less than 20 counts are deemed insignificant, it follows from Fig. 14 that only 21 error corrections by EDS were "significantly inaccurate." The poorly corrected samples — the tail of the Fig. 14 histogram — correspond to "sliced peaks," such as occurred at location D in Figs. 6 and 11.

The three histograms presented here as Figs. 12 through 14 together constitute the results of the statistical assessment/validation of our new error detection and smoothing algorithm, EDS.

Additional comments regarding final assessment of EDS are provided in the Conclusions section.

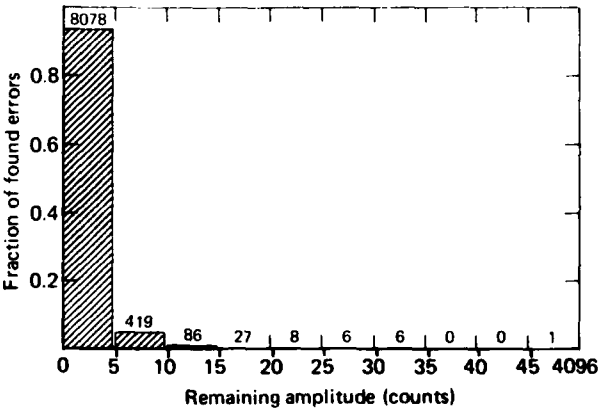


Figure 14. Amplitude histogram of residual errors after data smoothing. Of the 8631 errors that were correctly identified as errors by EDS, 8078 corrections (93.6%) were accurate to within 4 digital counts. Samples accumulated into the tail of the histogram are attributable to "sliced peaks," such as occurred at location D in Figs. 6 and 11, and multiple errors badly corrected at the ends of a scan line.

4. SYSTEM POINT RESPONSE

As discussed above, inspection of Fig. 3 supports the intuitive assertion that prior knowledge of system point response allows us to reliably visually detect single-sample errors: since single-sample errors give rise to spikes of width narrower than the point response, a pulse-width criterion serves to detect single-sample errors. This intuitive observation subsequently serves as the basis for developing an error-detection system for multiple errors as well as single errors. Since the key to our approach lies in exploiting prior knowledge of system point response (SPR) we will first discuss the SPR.

Each of the IR scanner's 32 detectors has a different point response. When we speak of system point response we are referring to the ensemble of measured point responses in which each detector is characterized individually.

The point response measurements presented here were obtained by horizontally scanning the sensor's field of view across a laboratory point source. (The laboratory set-up is described in Ref. 3.) The data are recorded digitally with a sample depth of 12 bits. Examples of measured point responses are given in Fig. 15 for longwave channels 5 and 2.

Inspection of Fig. 15 shows that the point response full-width at 10% peak amplitude is about 6.4 sampling intervals. The zero-amplitude baseline is reliably captured in about 12 sampling intervals, i.e., by representing the point response as a 13-sample waveform.

The sampling interval is 96 μrad (about 0.55% of 1°) in azimuth, established by slaving the A/D converter to a 16-bit shaft angle encoder on the sensor's scanning mirror:

$$(2\pi \text{ rad/rev})/(2^{16} \text{ samples/rev}) = 96 \ \mu\text{rad/sample} \quad (8)$$

The instantaneous field of view of each detector is ⅓ mrad square, as determined by the cell size (2 mils) and optics focal length (6 inches). Thus, the analog detector/postamplifier output voltages are sampled

electronically at a rate of

$$(333\ \mu\text{rad/dwell})/(96\ \mu\text{rad/sample}) = 3.47\ \text{samples/dwell}. \quad (9)$$

The shape of the point response is determined by a number of factors in addition to sample rate, including

- Optics blur,
- Detector size, shape, and spatial responsivity characteristics,
- Postamplifier frequency response,
- Point source position cross-scan, relative to the detector's geometric center, and
- Point source position in-scan relative to the cell geometric center and time-phasing of the A/D sampling clock.

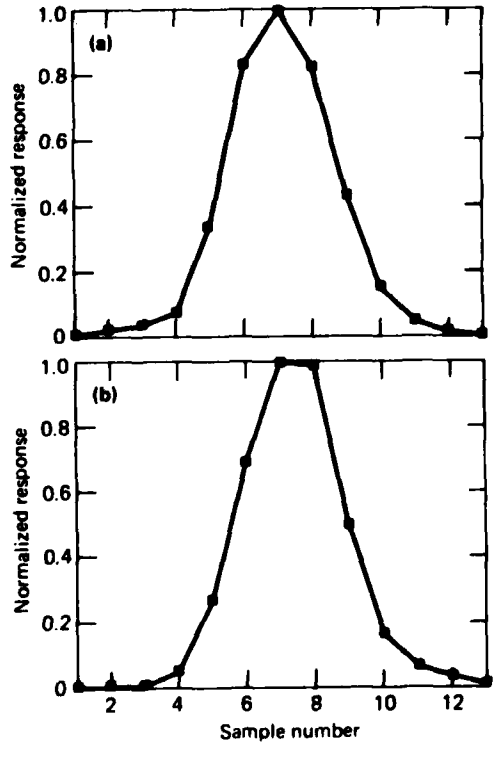


Figure 15. Measured point responses of longwave channel 5 (a) and longwave channel 2 (b). The waveform peak in (a) is very nearly an isosceles triangle while that in (b) is very nearly flat. The difference in structure between the peaks of these two waveforms is attributable to a small displacement of the optical point source in the direction of scan, as discussed in connection with Figs. 16 and 17.

We note that each detector is followed by a wideband preamplifier, followed in turn by an analog postamplifier. The postamplifiers each have a 1-pole lowpass characteristic with a 3 dB cut-off at 1000 Hz. If the postamp response were the only pulse broadening mechanism, the point response in-scan, $p(\theta)$, would thus be a simple exponential,

$$p(\theta) = \begin{cases} 0 , & \theta < 0 \\ \exp(-k_{co}\theta) , & \theta \geq 0, \end{cases} \quad (10)$$

where,

θ = azimuth angle, samples
k_{co} = cut-off spatial frequency, radians/sample,

and the optical point source is presumably located at $\theta = 0$. The cutoff spatial frequency is given by

$$k_{co} = 2\pi f_{co}/\dot{\theta} \quad (11)$$

where

f_{co} = postamp 3 dB cutoff = 1000 sec^{-1}

$\dot{\theta}$ = angular rate of scan, samples/sec.

The angular rate of scan is calculated in appropriate measure (samples/sec) as:

$$\dot{\theta} = (30.7\ \text{degrees/sec})(17.45\ \text{mrad/degree}) \times (1\ \text{sample}/0.096\ \text{mrad}) = 5580\ \text{samples/sec}. \quad (12)$$

From Eqs. 11 and 12,

$$k_{co} = 1.126\ \text{radians/sample}. \quad (13)$$

If the postamp alone were responsible for pulse broadening, the point response full-width at 10% peak amplitude (denoted as θ_{elec}) would be found from Eq. 10,

$$p(\theta_{elec}) = \exp(-k_{co}\theta_{elec}) = 0.1. \quad (14)$$

From Eqs. 13 and 14,

$$\theta_{elec} = 2.3/k_{co} \approx 2\ \text{samples} \quad (15)$$

We see from Fig. 15 that the actual full-width at 10% peak amplitude, θ_{fw}, is about 6.4 samples. Assuming that the separate contributors to the point response each contribute approximately as the square root of

the sum of squares, we find that the postamp response contributes only marginally (about 0.3 samples) to the overall width of the point response:

$$\theta_{fw} - (\theta_{fw}^2 - \theta_{elec}^2)^{1/2} \approx 0.3 \text{ samples.}$$

We conclude from the analysis of Eqs. 10 to 15 that, as expected, the measurement system's electrical frequency response is the smallest of the three major effects that, together, determine the width of the point response (cf. Table 3).

Table 3
Point response width is determined by the optics blur, the detector size, and the postamplifier frequency response. Tabulated here are the full-widths at 10% maximum amplitude of the partial responses due to each of the three main pulse-broadening mechanisms. Entries 2 and 3 are from Eqs. 9 and 15, respectively. Entry 4 is measured from Fig. 15. Entry 1 is inferred by assuming that the partial contributions add as the root of the sum of squares.

	Full width at 10% maximum amplitude
1. Optics blur (longwave)	5.0 samples
2. Detector size	3.5 samples
3. Electrical frequency response	2.0 samples
4. Total (longwave response)	6.4 samples

Visual inspection of the point response waveforms shows that, while their widths at 10% maximum amplitude are nearly the same, their detailed structure near the waveform peak exhibits considerable variation. For example, we see that the waveform peak in Fig. 15a is very nearly an isosceles triangle, while that in Fig. 15b is very nearly flat. We next show that variation in the detailed structure near the waveform peak is attributable to variation in the in-scan position of the optical point source.

An isosceles peak (e.g., Fig. 15a) is produced by the sampling circumstance depicted in Fig. 16. At the bottom of Fig. 16 we schematize a circular optics blur moving at a uniform speed across a stationary square detector. The origin of the time axis in Fig. 16, $t = 0$, is defined as the instant when the optics blur is perfectly centered on the detector, i.e., the time when the detected signal is maximum. Integral values on the "sample" axis in Fig. 16 correspond to times when the analog-to-digital (A/D) converter samples the postamp output. Figures 15a and 16 correspond to the unusual circumstance where one of the sampling times occurs by coincidence simultaneous with the reference time $t = 0$. Thus, a sample is obtained (sample 6 in this case) just as the postamp output reaches its maximum value.

A flat-topped peak (e.g., Fig. 15b) is produced by the sampling circumstance depicted in Fig. 17. In this case the optics blur is by coincidence equally off-center (in opposite directions) at samples 6 and 7.

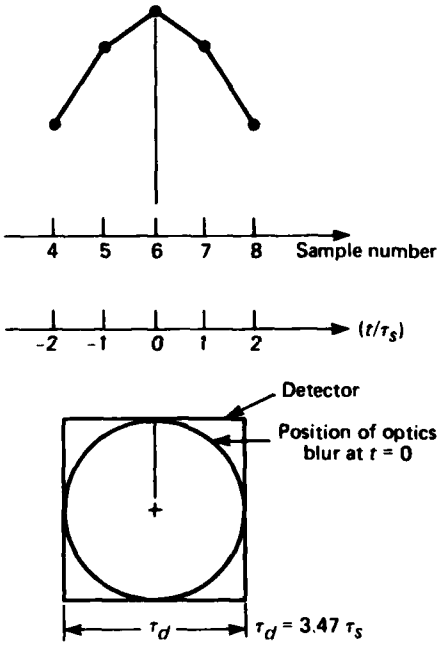


Figure 16. Sampling conditions that result in a symmetrically triangular (isosceles) point response peak, such as Fig. 15a. Quantity τ_s is the sampling interval (0.179 msec); quantity τ_d is the dwell time (0.622 msec).

Most generally, the peak of the point response has an asymmetric triangular shape, having neither the isosceles triangular shape of Fig. 15a nor the flat top of Fig. 15b.

In performing the point response measurements presented in this section cross-scan position was adjusted simply to maximize the point response peak amplitude. No measurements were performed suitable for assessing structural variations of the point response with sub-pixel deviations in point source cross-scan position.

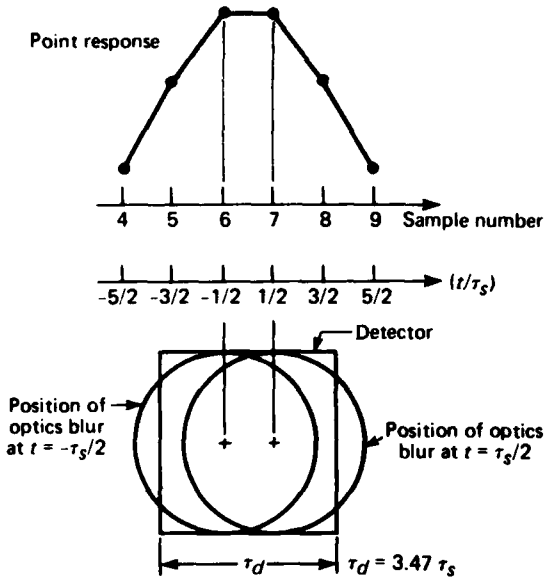


Figure 17. Sampling conditions that result in a flat-topped point response, such as Fig. 15b.

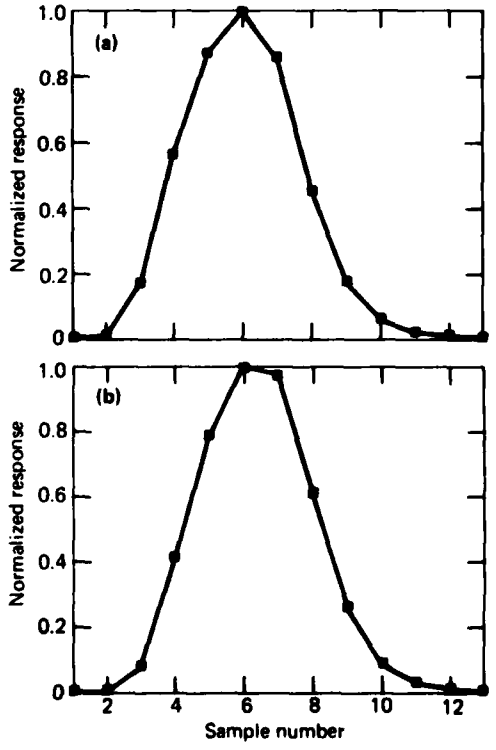


Figure 18. Measured point responses of (a) midwave channel 5 and (b) midwave channel 2.

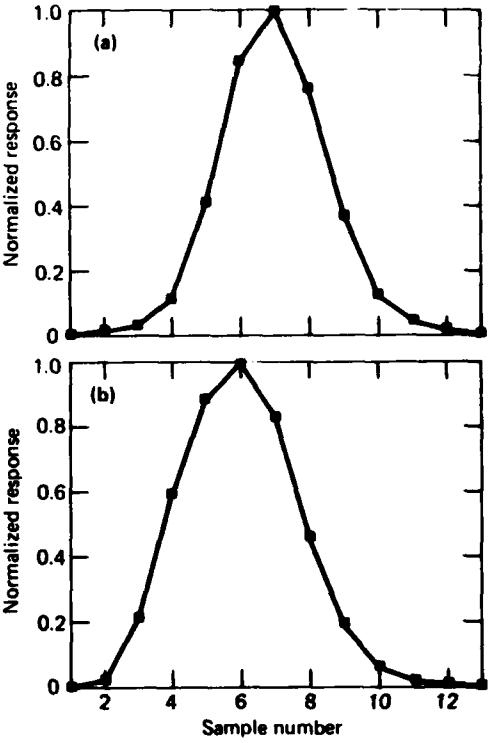


Figure 19. Average of measured point responses for (a) the longwave array and (b) the midwave array. Of the 16 longwave detectors, data from only 13 were available for forming the longwave average response. Data were available from only 12 of the 16 midwave cells.

Figure 18 depicts two sample midwave point responses. The midwave point responses are generally very similar to one another and to the longwave point responses.

Table 4 includes the full complement of point response data available at the time the error detection research was performed. The similarity in structure of the measured point responses in Table 4 suggests that for some applications it may suffice to use an array-average point response (Fig. 19). However, the array average responses are not used or referred to at any later point in this report.

The peak signal-to-noise ratios (S/N) of the point response measurements provided in Table 4 are:

$$S/N \approx \begin{cases} 750, \text{ midwave} \\ 250, \text{ longwave.} \end{cases} \qquad (16)$$

Table 4

Normalized point responses for the midwave and longwave arrays used at Montauk Point. No point responses and no clutter data are available for optically inactive elements: midwave channels 14 and 16, and longwave channel 1. Point response data were unavailable for longwave channel 16. Point response measurements were damaged by data recorder errors and hence are not tabulated for midwave channels 7 and 9, and longwave channel 4.

a. Normalized MWIR Point Responses

Channel	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0.000	0.001	0.073	0.449	0.815	1.000	0.966	0.577	0.244	0.080	0.026	0.008	0.003
2	0.000	0.005	0.075	0.412	0.786	1.000	0.976	0.610	0.262	0.088	0.030	0.013	0.006
3	0.000	0.007	0.102	0.467	0.814	1.000	0.941	0.567	0.245	0.085	0.028	0.010	0.004
4	0.000	0.012	0.171	0.553	0.856	1.000	0.925	0.551	0.233	0.079	0.026	0.011	0.003
5	0.000	0.012	0.169	0.569	0.873	1.000	0.856	0.458	0.178	0.061	0.021	0.009	0.005
6	0.000	0.029	0.258	0.641	0.910	1.000	0.824	0.435	0.166	0.054	0.017	0.005	0.003
8	0.000	0.028	0.244	0.620	0.888	1.000	0.852	0.474	0.193	0.068	0.023	0.008	0.003
10	0.000	0.027	0.269	0.652	0.913	1.000	0.785	0.400	0.153	0.049	0.016	0.007	0.005
11	0.000	0.017	0.219	0.603	0.892	1.000	0.796	0.422	0.176	0.057	0.021	0.008	0.003
12	0.000	0.035	0.307	0.694	0.938	1.000	0.741	0.359	0.131	0.042	0.012	0.005	0.001
13	0.000	0.030	0.284	0.679	0.937	1.000	0.724	0.355	0.129	0.045	0.015	0.007	0.003
15	0.000	0.054	0.376	0.762	0.978	1.000	0.669	0.302	0.107	0.033	0.011	0.005	0.003
Avg	0.000	0.021	0.212	0.592	0.883	1.000	0.838	0.459	0.185	0.062	0.020	0.008	0.003

b. Normalized LWIR Point Responses

Channel	1	2	3	4	5	6	7	8	9	10	11	12	13
2	0.000	0.004	0.008	0.051	0.262	0.688	1.000	0.992	0.489	0.165	0.068	0.038	0.013
3	0.004	0.016	0.033	0.082	0.392	0.890	1.000	0.796	0.392	0.131	0.037	0.012	0.000
5	0.004	0.020	0.035	0.074	0.336	0.836	1.000	0.832	0.434	0.148	0.047	0.012	0.000
6	0.000	0.008	0.051	0.210	0.611	0.946	1.000	0.537	0.183	0.074	0.035	0.008	0.000
7	0.000	0.004	0.019	0.038	0.134	0.622	1.000	0.931	0.641	0.252	0.080	0.019	0.000
8	0.000	0.012	0.044	0.238	0.647	0.968	1.000	0.540	0.198	0.075	0.036	0.008	0.004
9	0.000	0.016	0.032	0.067	0.310	0.817	1.000	0.857	0.480	0.171	0.056	0.012	0.004
10	0.012	0.016	0.048	0.222	0.635	0.964	1.000	0.520	0.171	0.063	0.028	0.012	0.000
11	0.000	0.015	0.031	0.104	0.340	0.861	1.000	0.807	0.417	0.143	0.046	0.027	0.008
12	0.000	0.012	0.035	0.191	0.572	0.911	1.000	0.560	0.198	0.074	0.039	0.012	0.004
13	0.004	0.019	0.039	0.074	0.377	0.879	1.000	0.794	0.381	0.125	0.039	0.004	0.000
14	0.000	0.008	0.019	0.061	0.290	0.718	1.000	0.939	0.420	0.130	0.057	0.031	0.004
15	0.000	0.016	0.033	0.078	0.393	0.898	1.000	0.803	0.377	0.123	0.037	0.012	0.004
Avg	0.002	0.013	0.033	0.115	0.408	0.846	1.000	0.762	0.368	0.129	0.046	0.016	0.003

Thus, the tabulated normalized point response values include a component of zero-mean Gaussian noise of RMS amplitude:

$$\sigma \approx \begin{cases} 0.0013, \text{ midwave} \\ 0.004, \text{ longwave} . \end{cases} \quad (17)$$

A number of detector cells were either totally inactive or too noisy to be useful, viz., midwave channels 14 and 16, and longwave channel 1. Consequently, there are neither point response data nor Montauk Point clutter data corresponding to these channels. Although longwave channel 16 was active at Montauk Point no point response data were available for this channel. Finally, there were a number of channels that were active, but for which the point responses were damaged by data recorder errors, viz., midwave channels 7 and 9, and longwave channel 4.

THE JOHNS HOPKINS UNIVERSITY
APPLIED PHYSICS LABORATORY
LAUREL, MARYLAND

5. ERROR PRIMITIVES: PEAKS AND EDGES

The input tc our data correction algorithm is a 400-sample time series,

$$x(k)\ ,\ k\ =\ 1,\ 2,\ \ldots,\ 400,$$

representing a single uncorrected scan line of IR data. The elements of $x(k)$ are integers, having the range of values

$$0 \le x(k) \le 4095.$$

The waveform $x(k)$ is subjected to an initial processing operation that generates as its product a vector of primitives,

$$p(k),\ k\ =\ 3,\ 4,\ \ldots,\ 398,$$

having 396 values in one-to-one correspondence with elements 3 to 398 of $x(k)$. The elements of $p(k)$ each have one of five possible values, reflecting an initial judgement concerning the likelihood of sample $x(k)$ containing a data error:

$$p(k) = \begin{cases} 0 & \text{No initially obvious error} \\ 1 & \text{Positive error peak} \\ -1 & \text{Negative error peak} \\ 2 & \text{Positive error edge} \\ -2 & \text{Negative error edge} \end{cases} \qquad (18)$$

In this section we explain the workings of an algorithm (PRMTVS) that generates $p(k)$, implicitly defining the terms "error peak" and "error edge."

The test for an error peak at sample i is conducted on four sequential slope values (Fig. 20),

$$\begin{aligned} \mathrm{LL} &= x(i-1) - x(i-2) \\ \mathrm{L} &= x(i) - x(i-1) \\ \mathrm{R} &= x(i+1) - x(i) \\ \mathrm{RR} &= x(i+2) - x(i+1). \end{aligned} \qquad (19)$$

The test for an error edge at sample-pair $(j, j+1)$ is conducted on three sequential slope values (Fig. 21),

$$\begin{aligned} \mathrm{L} &= x(j) - x(j-1) \\ \mathrm{C} &= x(j+1) - x(j) \\ \mathrm{R} &= x(j+2) - x(j+1)\ . \end{aligned} \qquad (20)$$

It is in the nature of our definitions for error peak and error edge that sequential elements of $p(k)$ must satisfy the set of adjacency rules given in Table 5. For example, it may be inferred from Table 5 that edge values must always appear in groups of at least two sequential values (e.g., Table 6, entries 3 and 4), and can never appear in isolation (Table 7, entry 2). Other examples of allowed and disallowed primitive se-

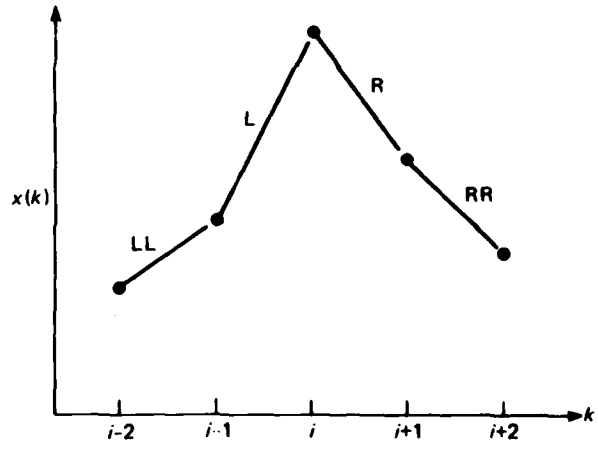


Figure 20. Notation for error peak test. The test for an error peak at sample i is conducted on four sequential slope values defined by the 5-sample neighborhood of i. Sample values of $x(k)$ are indicated by filled circles. Slope values are given by LL, L, R, and RR (Eq. 19).

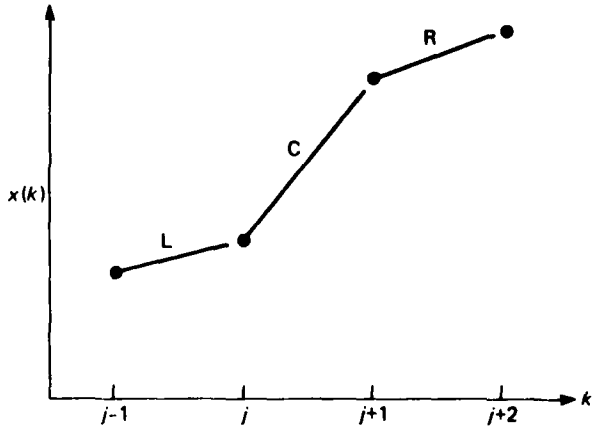


Figure 21. Notation for error edge test. The test for an error edge at the sample-pair $(j, j+1)$ is conducted on three sequential slope values defined by the 4-sample neighborhood $(j-1, j+2)$. Sample values of $x(k)$ are indicated by filled circles. Slope values are given by L, C, and R (Eq. 20).

quences are provided in Tables 6 and 7, respectively.

The algorithm that extracts error peak and error edge primitives from the waveform data is called PRMTVS (Appendix A).

Tracing the logic flow in PRMTVS, we see that indication of an error peak at sample k requires the satisfaction of a necessary condition, step 2.11.3; in addition, one of two auxiliary conditions must also be satisfied, steps 2.12.1 and 2.13.2.

Thresholding the slope absolute value, step 2.11.3a removes from consideration the very large number of low amplitude spikes attributable to closed-cover noise (an additive Gaussian noise having unity standard deviation, $\sigma = 1$). Step 2.11.3b is the basic peak condition, assuring a sign change in slope.

Table 5
Primitive adjacency rules. Table entry O indicates an allowed adjacency; entry X indicates a disallowed adjacency; R indicates a required adjacency.

$p(k)$	+1	−1	+2	−2
+1	X	O	O	O
−1	O	X	O	O
+2	O	O	R	O
−2	O	O	O	R

Table 6
Examples of allowed primitive sequences.

	$p(k)$	Interpretation
1.	{...0 1 0...}	Isolated peak
2.	{...0 1 −1 0...} {...0 1 −1 1 0...}	Adjacent peaks of alternating sign
3.	{...0 2 2 0...}	Isolated edge
4.	{...0 2 2 2 2 0...}	Staircase
5.	{...0 2 2 −2 −2 0...}	Plateau
6.	{...0 1 2 2 0...}	Peak adjacent to edge

Table 7
Examples of disallowed primitive sequences.

$p(k)$	Adjacency rule (Table 5)
1. {...0 1 1 0...}	X
2. {...0 2 0...}	R

The error peak primitives are of two distinct types, referred to by us as PK1 and PK2, corresponding to steps 2.12.1 and 2.13.2, respectively. Examples of PK1 and PK2 error primitives are given in Figs. 22 and 23.

The quantity q_1, defined by step 2.13.1 is interpreted by us as a "peak sharpness metric". Table 8 provides values of q_1 derived from the system point response, Table 4. A threshold Q_1 on peak sharpness, derived from Table 8, is applied in 2.13.2 to assure that PRMTVS does not mistakenly assign a value $p(k) = \pm 1$ to the peak of an optical point source

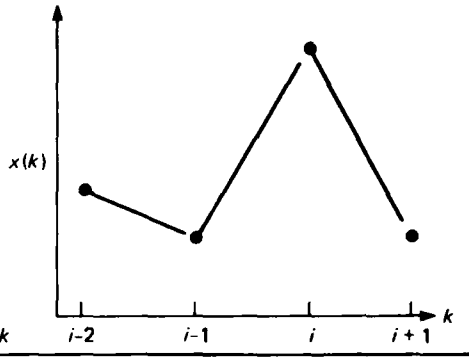


Figure 22. Example of PK1 error peak primitive at sample i. The values given for LL, L, and R may be seen to satisfy steps 2.11.3 and 2.12.1. Asterisks denote indeterminate values of p. In general, just three slope values, (LL, L, and R, or L, R, and RR), are needed to determine a PK1 error peak.

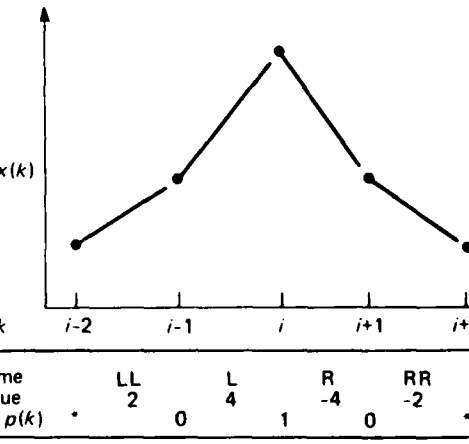


Figure 23. Example of PK2 error peak primitive at sample i. The values given for LL, L, R, and RR may be seen to satisfy steps 2.11.3 and 2.13.2. Asterisks denote indeterminate values of p. Regarding 2.13.2, note that for this example: $4 = q_1 > Q_1 = 1.85$.

Table 8

Metrics q_1 – q_3 are derived from the system point responses. Values for q_1 are obtained from step 2.13.1, Eq. 19, and Table 4. The value of i to use in Eq. 19 is i = 6 for midwave responses and i = 7 for longwave responses. Values for q_2 are obtained from step 2.25.1, Eq. 20, and Table 4. Values for q_3 are obtained from step 3.32.3 and Table 4.

	q_1		q_2 (left)		q_2 (right)		q_3	
Channel	MW	LW	MW	LW	MW	LW	MW	LW
1	0.593	*	0.857	*	1.059	*	0.578	*
2	0.639	0.750	0.678	0.815	0.983	1.506	0.588	0.701
3	0.696	0.726	0.825	1.184	0.983	0.868	0.584	0.680
4	0.675	*	0.826	*	0.953	*	0.561	*
5	0.777	0.750	0.866	1.174	0.941	0.879	0.589	0.692
6	0.790	1.470	0.769	0.735	0.871	0.619	0.586	0.703
7	*	1.010	*	1.032	*	0.633	*	0.717
8	0.809	1.448	0.778	0.730	0.880	0.585	0.579	0.685
9	*	0.738	*	1.196	*	0.833	*	0.687
10	0.894	1.483	0.760	0.735	0.832	0.595	0.601	0.696
11	0.921	0.762	0.785	1.392	0.831	0.835	0.608	0.709
12	0.936	1.479	0.749	0.719	0.782	0.641	0.607	0.709
13	0.995	0.740	0.773	1.183	0.734	0.891	0.608	0.689
14	*	0.778	*	0.836	*	1.478	*	0.713
15	1.005	0.665	0.717	1.211	0.697	0.945	0.617	0.683
16	*	*	*	*	*	*	*	*
minimum	0.593	0.665	0.678	0.719	0.697	0.585	0.561	0.680
maximum	1.005	1.483	0.866	1.392	1.059	1.506	0.617	0.717

(such as a small target at long range) that could possibly be present in the viewed scene. Applying a threshold to q_1 allows us to discriminate between target-like peaks, e.g., Fig. 16, and error peaks, e.g., Fig. 23.

According to Table 8,

$$0.59 < q_1 < 1.49,$$

from which we surmise that setting a threshold value

$$q_1 < Q_1 = 1.85$$

should assure that the error peak criterion in PRMTVS does not inadvertently cause damage to valid data, except on acceptably rare occasions.

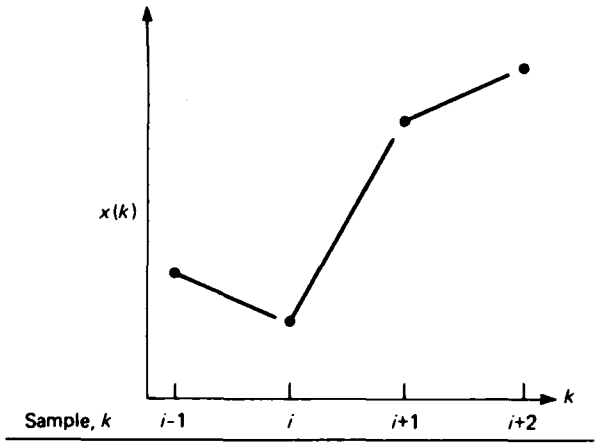
We momentarily defer discussion of step 2.13.3 (invoking the PK2MOD algorithm).

Considering now the second part of PRMTVS, which deals with error edges, we observe that indication of an error edge at sample k requires the satisfaction of a necessary condition, step 2.21.2. In addition, auxiliary conditions must also be satisfied: either 2.22.1, 2.23.1, or 2.25.2.

Thresholding the slope absolute amplitude, 2.21.2a, removes most noise artifacts from consideration. Step 2.21.2b imposes the requirement that neither candidate edge point has previously been labeled by PRMTVS as an error peak.

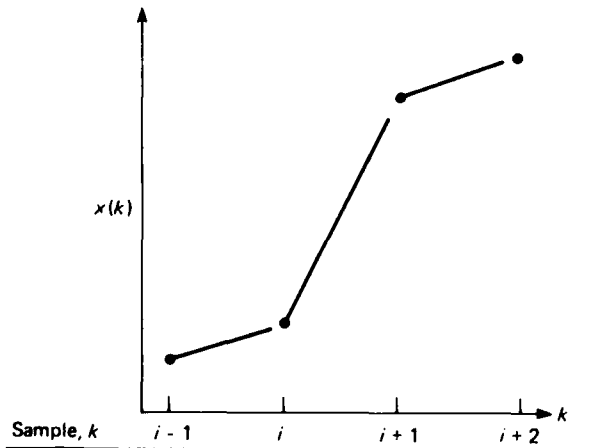
The error edge primitives are of two distinct types, called EDG1 and EDG2, corresponding respectively to steps 2.22.1/2.23.1, and 2.25.2. Examples of EDG1 and EDG2 error primitives are given in Figs. 24 and 25.

The quantity q_2 defined in step 2.25.1 is interpreted by us as an "edge sharpness metric." Table 8 provides values of q_2 derived from the system point responses, Table 4. A threshold, Q_2, on edge sharpness, derived from Table 8, is applied in step 2.25.2 to assure that PRMTVS does not mistakenly assign an error value $p(k) = \pm 2$ to the shoulder of an optical point source possibly present in the viewed scene.



Slope name		L	C	R
Slope value		-1	5	1
Primitive $p(k)$	*	2	2	*

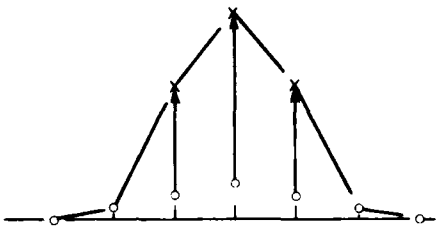
Figure 24. Example of EDG1 error edge primitive at sample pair $(i, i + 1)$. The values given for L, C, and R may be seen to satisfy steps 2.21.2 and 2.22.1. Asterisks denote indeterminate values of p.



Slope name		L	C	R
Slope value		1	6	1
Primitive $p(k)$	*	2	2	*

Figure 25. Example of EDG2 error edge primitive at sample pair $(i, i + 1)$. The values given for L, C, and R may be seen to satisfy steps 2.21.2 and 2.25.2. Asterisks denote indeterminate values of p. Regarding 2.25.2, note that for this example: $3 = q_2 > Q_2 = 2$.

The error primitives defined above are unable to detect a type of length-three artifact illustrated in Fig. 26. (The likelihood of this artifact type assembling by chance, i.e., frequency-of-occurrence statistics, has not been established.) The PRMTVS algorithm at step 2.13.3 invokes a routine, PK2MOD (cf. Appendix A), that augments our original definition of the PK2 error primitive by implementing a six-slope test capable of discovering "flat-footed peak" artifacts such as that depicted in Fig. 26. However, we note that PK2MOD was not operational during the numerical validation studies discussed in connection with Figs. 12 through 14.



Sample number: k	1	2	3	4	5	6	7
Original data: $a(k)$	0	1	2	3	2	1	0
Errors: $s(k)$	0	0	9	14	9	0	0
Damaged data: $b(k)$	0	1	11	17	11	1	0
PK2: $p(k)$	0	0	0	0	0	0	0
PK2MOD: $p(k)$	0	0	0	1	0	0	0

Figure 26. Example of a length-3 artifact that would go undetected but for the action of algorithm PK2MOD. Errors of amplitude $s(k)$ are implanted at $k = 3, 4$, and 5. For $k = 5$, block 2.25 calculates $q_2 = 1.43 < Q_2 = 2$; i.e., the shoulders of the artifact are not steep enough to qualify as error edges. For $k = 4$, block 2.13 calculates $q_1 = 1.2 < Q_1 = 1.85$; i.e., the 4-slope PK2 test also fails to discover the artifact. However, the 6-slope test implemented in PK2MOD returns a non-null value for $p(4)$, as desired. As determined by PK2MOD, the tails of the peak depicted here flatten too quickly (compare with Fig. 15a).

At step 3.32.4 in PK2MOD the shape-related metric q_3 is compared with a threshold value Q_3. The value assigned to Q_3 is developed by applying the six-slope test in PK2MOD to point response data on a channel-by-channel basis (cf. Table 8), in a process strictly analogous to the development of threshold values for q_1 and q_2, the peak and edge sharpness metrics defined earlier.

6. INITIAL ESTIMATES FOR ARTIFACT BOUNDARIES

The vector of primitives generated by the PRMTVS algorithm, $p(k)$, is next operated on by the BOUNDS algorithm to produce estimates for the starting and ending locations of each error artifact in the scan line. We define the beginning sample number of artifact j, $b(j)$, as the first defective datum in artifact j (not the last valid datum preceding the artifact). Similarly, the final sample number of artifact j, $f(j)$, we define as the last defective datum in artifact j (not the first valid datum following the artifact).

The BOUNDS algorithm (Appendix A) begins by searching for the first non-null element of $p(k)$ (step 4.11.2). If a non-null element is found (say, at location k_1), the artifact counter is incremented to unity ($n = 1$) and the beginning location of the first artifact, $b(1)$, is estimated (step 4.12.2) as $b(1) = k_1$. The algorithm then looks for three successive null values of $p(k)$ before deciding that the artifact has terminated (step 4.13.2).

The operation of PRMTVS and BOUNDS is illustrated in Fig. 27 for an isolated error implanted into a constant-valued waveform. For this simple example the artifact boundary estimates generated by BOUNDS precisely indicate the error's location.

We define a "plateau" as an error artifact having initial and final primitives that are edges of opposite sign, i.e., either $[p(b),p(f)] = (2,-2)$, or $[p(b),p(f)] = (-2,2)$, so that $p(b) \times p(f) = -4$ (step 4.12.8). Our estimates for the boundaries of a plateau artifact involve a shrinking process (block 4.14) rationalized in Fig. 28.

We define a first estimate for the length of artifact j, i.e., the number of sequential defective samples in the artifact as

$$L(j) = f(j) - b(j) + 1 . \qquad (21)$$

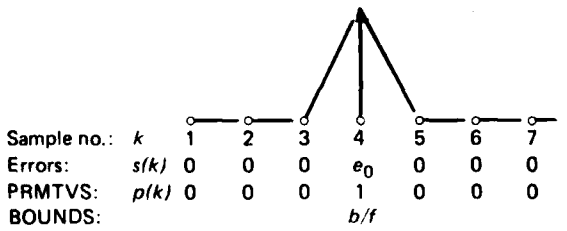


Figure 27. Isolated error implanted at location $k = 4$ in a constant-valued waveform. The PRMTVS algorithm generates a peak error primitive at the location of the error, i.e., $p(4) = 1$, and null primitives elsewhere. The BOUNDS algorithm indicates the existance of an error artifact having beginning and final locations (b and f, respectively) coincident with one another and with the error's location.

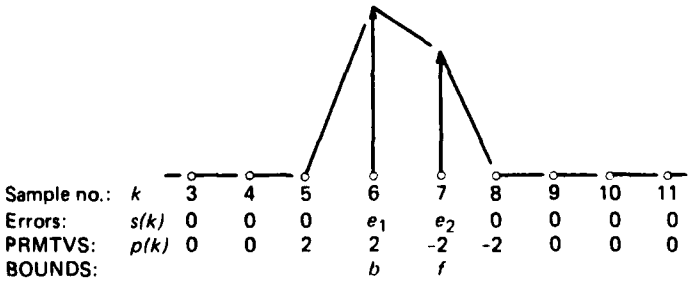
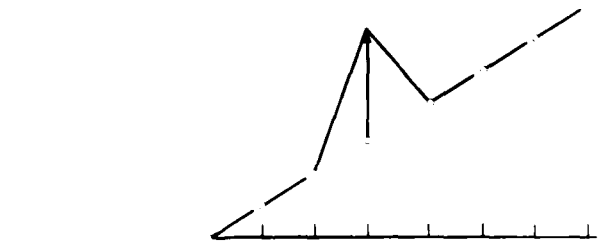


Figure 28. Plateau error artifact implanted at locations k = 6 and 7 in a constant-valued waveform. The BOUNDS algorithm executes a boundary-shrinking operation for this type of artifact such that the estimated beginning and final artifact locations (b and f, respectively) coincide with the locations of the implanted errors (block 4.14). A plateau is defined as an artifact for which $[p(b) \cdot p(f)] = -4$ (step 4.12.8).

7. IDENTIFICATION OF ISOLATED ERRORS

Contrary to the impression perhaps conveyed by Figs. 27 and 28, the artifact boundary estimates generated by BOUNDS often provide an inaccurate indication of precisely which data are in error. This point is illustrated by analysis of five examples (Figs. 29 through 33). The lessons learned from these examples are generalized to a set of rules for refining the initial artifact boundary estimates.

Figure 29 depicts a single error implanted into a constant slope waveform. The slope, though greater than 3 (step 2.11.3a), is much smaller than the amplitude of the implanted error. The simple peak-finding algorithm in PRMTVS has labeled sample $k = 5$ in Fig. 29 as a negative peak, and BOUNDS is designed to retain all such labeled features as candidate errors, i.e., as candidates for correction. In this instance, the artifact boundary estimates provided by BOUNDS encompass the actual error at $k = 4$ and a valid datum at $k = 5$.

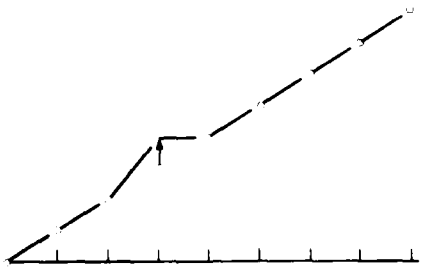


Sample no.:	k	1	2	3	4	5	6	7	8
Error:	$s(k)$	0	0	0	e_0	0	0	0	0
PRMTVS:	$p(k)$			0	1	-1	0	0	0
BOUNDS:					b	f			

Figure 29. A single error is implanted at location $k = 4$ in a constant-slope waveform. The slope amplitude is greater than 3, but much smaller than e_0 (the amplitude of the implanted error). The artifact boundary estimates provided by BOUNDS encompass the error at $k = 4$ and a valid datum at $k = 5$.

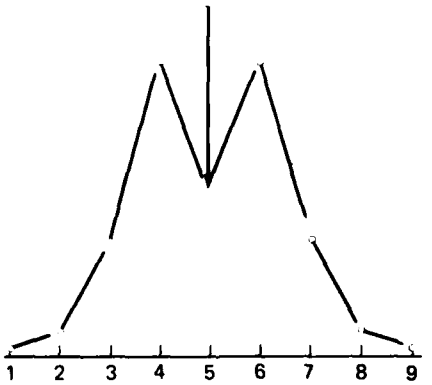
Figure 30 is similar to Fig. 29 in once again depicting a single error implanted into a constant-slope waveform. However, in Fig. 30 the amplitude of the implanted error equals, by coincidence, the slope amplitude. Once again, the boundaries provided by BOUNDS encompass the actual error at $k = 4$. However, three valid data points ($k = 3$, 5, and 6) are also indicated as candidates for correction, indicating that BOUNDS is apparently unable to localize the error.

Figure 31 depicts the implantation of a single error at the peak of the point response, giving rise to a



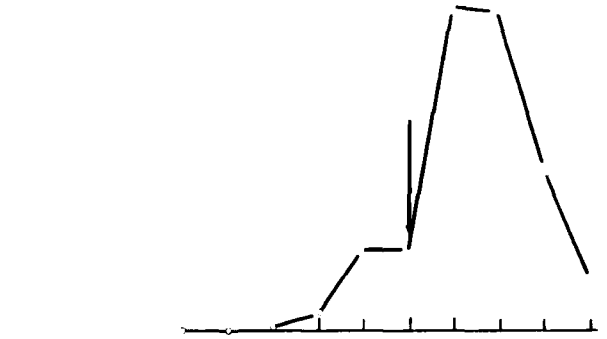
Sample no.:	k	1	2	3	4	5	6	7	8	9
Error:	$s(k)$	0	0	0	e_0	0	0	0	0	0
PRMTVS:	$p(k)$			2	2	2	2	0	0	0
BOUNDS:				b			f			

Figure 30. A single error is implanted at location $k = 4$ in a constant-slope waveform. The amplitude of the implanted error equals by coincidence the slope amplitude. The artifact boundary estimates generated by BOUNDS encompass the error at $k = 4$ and three valid data points as well.



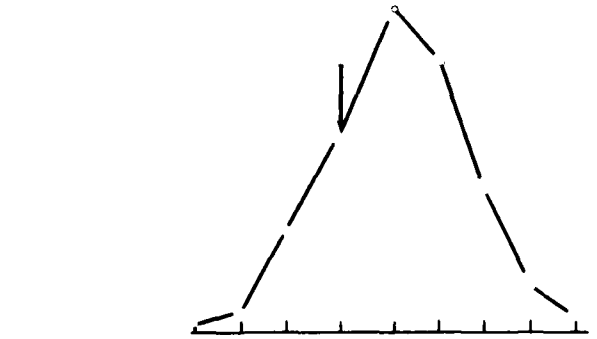
Sample no.:	k	1	2	3	4	5	6	7	8	9
Error:	$s(k)$	0	0	0	0	$-e_0$	0	0	0	0
PRMTVS:	$p(k)$			0	1	-1	1	0	0	0
BOUNDS:					b		f			

Figure 31. A single error occurs at the peak of the point response, giving rise to a double-peaked artifact. Artifact boundaries b and f estimated by BOUNDS encompass the error at $k = 5$, as well as valid data at $k = 4$ and $k = 6$.



Sample no.:	k	1	2	3	4	5	6	7	8	9	10
Error:	$s(k)$	0	0	0	0	0	$-e_0$	0	0	0	0
PRMTVS:	$p(k)$			0	2	2	2	2	0	0	0
BOUNDS:					b			f			

Figure 32. A single error occurs one sample distant from the peak of a flat-topped point response (Fig. 15b), creating a staircase-shaped artifact. BOUNDS labels a 4-sample sequence as suspect.



Sample no.:	k	1	2	3	4	5	6	7	8	9
Error:	$s(k)$	0	0	0	$-e_0$	0	0	0	0	0
PRMTVS:	$p(k)$			0	0	1	0	0	0	0
BOUNDS:						b/f				

Figure 33. A single error occurs one sample distant from the peak of an isosceles point response (Fig. 15a). The boundary estimates b and f provided by BOUNDS do not encompass the error at $k = 4$ for the depicted example.

double-peaked artifact. The artifact boundaries generated by BOUNDS encompass the error at $k = 5$, as well as one valid datum on each side of the error.

In Fig. 32 a single error occurs one sample distant from the peak of a flat-topped point response (e.g., Figs. 15b and 17). The amplitude of the error is such that a staircase artifact (Table 6, entry 4) is created. The four-sample sequence labeled as suspect by BOUNDS encompasses the isolated error.

As a final canonical example of single sample damage we consider Fig. 33, in which the single error occurs one sample distant from the peak of an isosceles point response (e.g., Figs. 15a and 16). The scale of the point response and the amplitude, sign, and location of the error all conspire to produce coincident boundary estimates b and f that, in this pathological instance, do not encompass the actual error.

We draw two conclusions from the preceding examples, relevant to the design of the ISOERR algorithm, whose purpose is to detect and correct isolated errors.

- An artifact having a BOUNDS length estimate as large as 4 may be caused by an isolated error (e.g., Figs. 30 and 32). Thus, a single sample correction should be attempted whenever $L \leq 4$.
- An artifact having a BOUNDS length estimate of 1 may be caused by an isolated error located one sample distant (left or right) from the non-null primitive (Fig. 33).

The ISOERR algorithm (Appendix A) begins with an initial determination of whether a single sample correction should be attempted for artifact j. As noted above, single sample correction is attempted whenever $L(j) \leq 4$ (step 5.11.2). If $L(j) > 4$, the artifact is left untreated by ISOERR.

Quantities B and F, initialized in step 5.12.1, are the boundaries of a search interval: each sample value of $x(n)$ (for $B \leq n \leq F$) is separately considered as a possible isolated error.

As discussed in connection with Fig. 33, when $L(j) = 1$ we must consider the possibility that an isolated error may have occurred at any of three possible locations, viz., $b - 1$, b, and $b + 1$. Thus, ISOERR next tests to see if $L = 1$ (step 5.12.2); if so, ISOERR expands in block 5.125 the search boundaries B and F.

We now draw what we refer to as the "block 5.13 hypothesis." In block 5.13 we test the hypothesis that the jth artifact is caused by an isolated error occurring at the location n within the artifact.

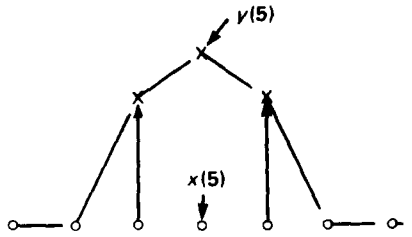
The first action within block 5.13 is to save the original, presumably defective, data value at location n (step 5.13.1). Next, the NTRPL8 algorithm is used in step 5.13.2 to generate a replacement value for $x(n)$, called $y(n)$, by interpolation from the neighboring values $x(n-2)$, $x(n-1)$, $x(n+1)$, and $x(n+2)$, which are hypothesized to be valid data. (The cubic interpolation implemented by NTRPL8 is derived in Appendix B). If the block 5.13 hypothesis is viable, the artifact will now have been eliminated. Thus

where we recall in applying Eq. 22 that waveform $x(k)$ has been modified by replacing $x[N(i)]$ with the interpolated value $y[N(i)]$ (step 5.13.3).

For example, considering Fig. 34, the first of the viable single sample corrections ($i = 1$) occurs at location $k = N(1) = 3$; the second viable single sample correction ($i = 2$) occurs at location $k = N(2) = 4$.

We close this section with one more example illustrating the interaction between the algorithms discussed up to this point: PRMTVS, BOUNDS, ISOERR, and NTRPL8.

Figure 35 depicts a constant-valued waveform into which equal amplitude errors have been inserted at locations $k = 4$ and $k = 6$. The damaged waveform is input to PRMTVS, which generates a vector of primitives, $p(k)$, containing three non-null characters (Fig. 35). The BOUNDS algorithm identifies one error artifact in the waveform, extending from $b = 4$ to $f = 6$, encompassing the data errors at $k = 4$ and $k = 6$, and a valid datum at $k = 5$. Since $L = 3 \leq 4$, ISOERR attempts to eliminate the artifact with a single sample correction applied at each of three locations: $k = 4$, 5, and 6.



Sample no.:	k	2	3	4	5	6	7	8
Errors:	$s(k)$	0	0	e_0	0	e_0	0	0
PRMTVS:	$p(k)$		0	1	-1	1	0	0
Block 5.13								
First pass:	$p'(k)$		0	0	0	1	0	0
Second pass:	$p'(k)$		2	2	0	-2	-2	0
Third pass:	$p'(k)$		0	1	0	0	0	0

Figure 35. Equal amplitude errors are inserted at locations $k = 4$ and $k = 6$ into a constant-valued waveform. The dashed waveform depicted here, passing through $y(5)$, is the waveform's appearance after the second pass through step 5.13.3.

On the first pass through block 5.13 ($n = 4$), ISOERR applies a trial correction at location $k = 4$.

As indicated in Fig. 35, subsequent output generated by PRMTVS at step 5.13.4, $p'(k)$, has a non-null character, $p'(6) = 1$. Consequently, block 5.135 is not executed, and the counter of viable corrections retains its initial value, $i = 0$.

On the second pass through block 5.13 ($n = 5$), ISOERR applies a trial single correction at location $k = 5$. The replacement value for $x(5)$ generated by NTRPL8 is

$$y(5) = x(5) + 4e_0/3$$

as indicated in Fig. 35. Inspection of Fig. 35 shows that the originally separate errors at $k = 4$ and $k = 6$ have been "blended" smoothly with yet a third error produced by ISOERR itself. Indeed, when PRMTVS is applied to the resultant waveform (now containing three errors), it is found that the negative error peak originally present at $k = 5$ has been eliminated, i.e., while originally $p(5) = -1$, after the second pass through block 5.13 we obtain $p'(5) = 0$. However, the "shoulders" of the error artifact are still too abrupt to be optically induced: two error edges now appear in $p'(k)$.

It follows from the foregoing discussion that ISOERR leaves unchanged the multiple error artifact depicted in Fig. 35: ISOERR neither damages good data nor corrects either of the errors.

We note that ISOERR detects and corrects with high accuracy all six examples of single-sample damage presented above in Figs. 29 through 34.

Finally, we note that the version of EDS used in our statistical validation study had a call to PK3MOD following step 5.13.4, rather than at step 2.13.3, where PK3MOD could potentially have had greater impact in reducing undetected errors.

THE JOHNS HOPKINS UNIVERSITY
APPLIED PHYSICS LABORATORY
LAUREL, MARYLAND

8. IDENTIFICATION OF MULTIPLE ERRORS

Multiple sequential errors generally cannot be corrected, i.e., there exists no way to reconstruct with high accuracy the original data values. The most we can accomplish in our treatment of multiple errors is to identify which of the data are defective, and then to smoothly interpolate through the damaged regions. Once data errors are properly labeled they may be excluded from use in subsequent analysis ("excised" from the data set). Consequently, the net effect of having multiple errors is a speckling of "blind spots" in the data, with a consequent reduction in data set size and ease of use, but no degradation in the structural integrity of the waveforms.

Our algorithm for detecting and smoothing multiple sequential errors is called MLTPLS (Appendix A). The waveform data entered as input to MLTPLS have previously been operated on by ISOERR; i.e., the isolated errors have been corrected prior to operating on the data with MLTPLS.

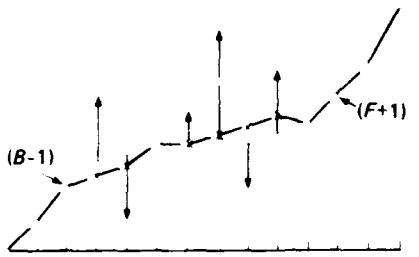
In step 7.12.2 MLTPLS attempts to excise artifact n by linearly interpolating from $[B(n) - 1]$ to $[F(n) + 1]$. The smoothed version of $x(k)$ is then examined by PRMTVS in step 7.12.3 to determine whether any excessively abrupt peaks or edges remain in the waveform. If so, the artifact boundaries are expanded in block 7.31 to obtain new values for $B(n)$ and $F(n)$, and MLTPLS iterates once again through the linear interpolation block. The process of boundary expansion and artifact smoothing continues until PRMTVS returns a vector of null primitives, which causes MLTPLS to terminate.

It was noted during algorithm validation that many of the undetected errors (Fig. 13) occur in between pairs of recognized multiple error artifacts. This observation motivated the development of the MERGER algorithm (Appendix A), invoked from MLTPLS at step 7.31.3. However, our statistical validation, (Figs. 12 through 14) is based on a version of EDS that does not include MERGER.

The values assigned to constants C_1 and C_2 at step 9.11.1 establish a trade-off between data damage and missed errors. Quantifying this trade-off, and thus selecting optimum values for C_1 and C_2, would require an extended Monte Carlo analysis that has not been performed.

We next illustrate the operation of MLTPLS with three examples.

Figure 36 depicts a large error artifact implanted in a slowly varying waveform. In this instance, the original boundary estimates b and f generated by



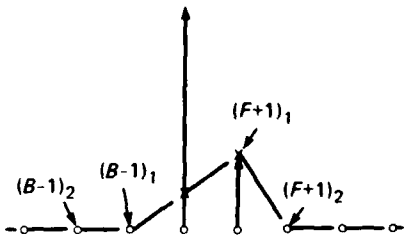
Sample no.:	k	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Errors:	$s(k)$	0	0	0	e_1	$-e_2$	0	e_3	e_4	$-e_5$	e_6	0	0	0	0
PRMTVS:	$p(k)$	0	0	0	1	-1	0	0	1	-1	1	-1	0	0	0
BOUNDS:					b							f			
Step 7.12.3	$p'(k)$	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 36. Example of a large error artifact requiring for excision just one pass through block 7.12. As shown here, linear interpolation through the excised region produces small correction errors when the original data (open circles) are slowly varying. Step 1.7 in executive routine EDS restores the measured value when it differs only slightly from the corrected value. This restoration step prevents damage to samples $x(8)$ and $x(13)$ in the depicted example.

BOUNDS do not need to be modified, and only one pass through block 7.12 is required. Moreover, the differences between the actual data values (O) and the interpolated values (X) (i.e., the "correction errors") are apparently small.

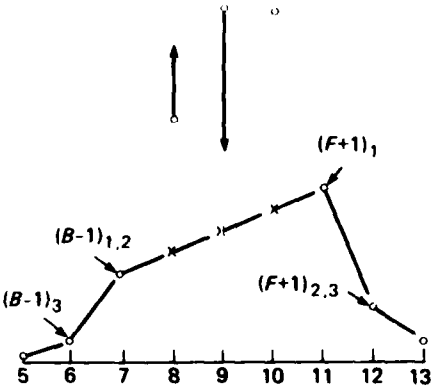
In Fig. 37 two sequential data errors are implanted in an otherwise constant-valued waveform. In this instance, the very large error at $k = 7$ prevents PRMTVS from recognizing the smaller error at $k = 8$ either as an error peak or an error edge. However, after one pass through block 7.12 has largely suppressed the dominant error at $k = 7$, PRMTVS on the second pass through block 7.12 successfully uncovers the smaller error at $k = 8$.

In Fig. 38 we illustrate the point that multiple errors occurring in a high-curvature waveform typically produce significant irrecoverable damage. In this figure, errors occur at two locations in the center of the longwave/channel 2 point response. On the first pass through block 7.12, linear interpolation from $(B - 1)_l$ to $(F + 1)_l$ produces the dashed curve shown in Fig. 38. On successive passes, the residual highly damaged structure is progressively sliced away until, on the third pass, MLTPLS terminates with the data



Sample no.:	k	4	5	6	7	8	9	10	11
Errors:	$s(k)$	0	0	0	$3e_0$	e_0	0	0	0
PRMTVS:	$p(k)$	0	0	0	1	0	0	0	0
BOUNDS:					b/f				
Step 7.12.3									
First pass:	$p'(k)$	0	0	2	2	1	0	0	0
Second pass:	$p'(k)$	0	0	0	0	0	0	0	0

Figure 37. Example of a length-2 artifact requiring for excision two passes through block 7.12. Errors are implanted at two locations, $k = 7$ and $k = 8$, in a constant amplitude waveform. On the first pass, the SMOOTH algorithm performs linear interpolation from $(B - 1)_1$ to $(F + 1)_1$. As shown by the dashed waveform, the artifact has not yet been eliminated. On the second pass through 7.12, SMOOTH performs linear interpolation from $(B - 1)_2$ to $(F + 1)_2$. Algorithm MLTPLS then terminates with the artifact having been wholly excised.



Sample no.:	k	5	6	7	8	9	10	11	12	13
Errors:	$s(k)$	0	0	0	e_1	$-e_2$	0	0	0	0
PRMTVS:	$p(k)$	0	0	0	1	-1	1	0	0	0
BOUNDS:					b		f			
Step 7.12.3										
First pass:	$p'(k)$	0	0	0	0	0	0	1	0	0
Second pass:	$p'(k)$	0	2	2	0	0	0	0	0	0
Third pass:	$p'(k)$	0	0	0	0	0	0	0	0	0

Figure 38. Errors occur at two locations in the center of a point response waveform. On the first pass through block 7.12, linear interpolation from $(B - 1)_1$ to $(F + 1)_1$ produces the dashed curve shown in the figure. Algorithm MLTPLS terminates after the third interpolation, from sample $(B - 1)_3$ to sample $(F + 1)_3$. Thus, a 5-sample blind spot is created, masking out both the data errors and the point response.

errors and point response both having been wholly excised. The final interpolation is between samples $(B - 1)_3 = 6$ and $(F + 1)_3 = 12$. Thus, a 5-sample blind spot has been created in the data, masking out what had originally been a significant structural feature.

The peak-slicing process illustrated in Fig. 38 is also illustrated in Figs. 6, 8, and 11 (feature D).

9. EXECUTIVE ROUTINE EDS

Now that the elements of our error detection system have been discussed, we bring all of our algorithms together under the control of executive routine EDS (Appendix A).

In step 1.1, EDS uses PRMTVS to screen a scan line of data. If PRMTVS detects no excessively abrupt peaks or edges in the data, EDS terminates in step 1.2. Otherwise, the locations of the detected error peaks and error edges are used by BOUNDS in step 1.3 to determine neighborhoods within which error smoothing will be required.

In step 1.4, EDS applies, via ISOERR, a single sample correction to each of the damaged neighborhoods amenable to this simplest expedient. Assuming that at least one neighborhood has been repaired with a single sample fix, block 1.51 operates on the partially

repaired waveform to establish a residual set of multiply damaged neighborhoods. The multiple error regions are smoothed by the MLTPLS algorithm in step 1.6.

The last operation in EDS, step 1.7, compares the smoothed waveform with a stored version of the original unprocessed waveform. Where data values have been changed by only a small amount (nominally, four or fewer digital counts) the original data value is restored to the waveform. (Sensor internal noise is one digital count, RMS; sensor full-scale dynamic range is 4096 digital counts.)

We note that step 1.7 was inactive during the statistical validation discussed in connection with Figs. 12 through 14.

10. CONCLUSIONS

A Navy program was recently begun to obtain a test set of infrared (IR) background images for developing IR search and track clutter suppression techniques. Unfortunately, the initial set of measurements was contaminated by a malfunctioning digital recorder with about 4% defective data (3% small-amplitude errors and 1% large-amplitude errors).

In this report we present a new Error Detection and Smoothing algorithm, EDS, and we demonstrate that EDS is capable of eliminating from the IR data almost all large amplitude errors (amplitude greater than 9 times the standard deviation of the sensor closed-cover noise).

The EDS algorithm is generally capable of discriminating electrically induced burst noise (including burst noise from electromagnetic interference) from structural features induced by optical variations in the viewed scene. Moreover, EDS is generally applicable to data from any optical scanner whose output is sampled digitally multiple times per dwell.

The EDS algorithm operates entirely in the time domain. The bases for error detection are smoothness criteria extracted from measurements made against a laboratory point source, and do not depend on the highly variable structure of the IR clutter data viewed in the field. In effect, EDS exploits the redundancy inherent in highly sampled image data.

The computation is structured such that the data are pre-screened with a simple algorithm that allows further attention to be concentrated only on "suspect" neighborhoods. There are no computation-intensive steps, such as FFT's or histograms, required for the data set as a whole.

The data are sampled at a sufficiently high rate that isolated errors can be corrected by EDS to high accuracy. However, multiple sequential errors generally cannot be corrected; i.e., there exists no way to reconstruct with high accuracy the original data values. Our approach to treating multiple errors is to identify which of the data are defective, and then to smoothly interpolate through the damaged regions. Once data errors are properly labeled they may be excluded from use in subsequent analysis ("excised" from the data set). Consequently, the net effect of having multiple errors is a speckling of "blind spots" in the data, with a consequent reduction in data set size and ease of use, but no degradation in the structural integrity of the waveforms.

A valuable feature of the new algorithm is its ability to accurately isolate error bursts (i.e., long sequences of defective data) by means of a region growing approach.

We have presented a statistical validation based on evaluating the performance of EDS against a synthetically degraded data set. We have found that, on the average, less than 1% of the larger-amplitude errors propagate undetected through the error detection algorithm. The density of large amplitude errors is thus reduced from about 60 per frame to about 0.5 per frame (counting only those errors whose amplitude is at least 10 times the sensor noise).

We have observed that the errors undetected by EDS tend to be of the following kinds:

- Errors at the edge of a recognized error burst, where the error burst abuts a region of high curvature in the IR data,
- Errors sandwiched closely between a pair of recognized error artifacts (typically, of large extent),

- Errors occurring in waveforms that are identified by EDS as having exceptionally high error content.

These observations could be used to derive useage guidelines for the data. Formalized as algorithms (e.g., the MERGER algorithm in Appendix A), it is probable that these guidelines could reduce the rate of undetected errors significantly below the nominal 0.5 errors per frame quoted above.

Finally, it must be said that the block-structured algorithms presented as EDS in Appendix A are completely restructured from the FORTRAN source that was used in our statistical validation. It is possible that, in the process of casting EDS into a more easily explained form, we may have introduced logic errors.

We conclude that our immediate objectives were met in the sense that EDS proved adequate for extracting from the damaged IR data a limited interim data set for use until better data were obtainable. From a larger perspective, we have successfully demonstrated the utility of a structural approach to error detection for highly sampled image data. Moreover, we have built in EDS a structure upon which could be based even more powerful error detection algorithms.

ACKNOWLEDGMENTS

The concepts underlying EDS and the original FORTRAN source implementing a version of EDS were developed by R. A. Steinberg while employed by the Naval Research Laboratory (NRL), Washington, D.C. The numerical validation of EDS was performed on an NRL VAX 11/780 computer.

Financial support provided to R. A. Steinberg by Dr. John C. Kershenstein, NRL, is gratefully acknowledged. Essential support in the documentation phase of this work was provided by Dr. William J. Tropf, of APL.

Thanks are due to Brian Sweeney (Sachs/Freeman Associates, Bowie, Md.) for software development and computer graphics.

The IR scanner used to obtain the data in this report is owned and operated by Raytheon Missile Systems Division, Bedford, Mass. Thanks are due to Raytheon employees Irving Goldstein, Andre Krutchkoff, and Joseph DiBiaso, for their enthusiastic and highly competent technical assistance.

The data errors discussed in this report were due to a damaged data recorder that is being replaced by an entirely new digital data acquisition system being developed by NRL. The data errors in no way subtract from Raytheon's remarkable technical accomplishments in fabricating the IR scanner — a uniquely valuable tool for the development of advanced IR search and track devices.

REFERENCES

[1]R. L. Lucke, A. P. Schaum, J. C. Kershenstein, J. Michalowicz, B. V. Kessler, A. B. Blumenthal, I. Goldstein, and A. Krutchkoff, "The Navy's Infrared Background Measurement and Analysis Program," *Proc. IRIS Specialty Group Meeting on Targets, Backgrounds, and Discrimination,* San Diego, 11-14 Feb. 1986.

[2]R. A. Steinberg, "Navy IR Background Measurements and Analysis Program," *Proc. Tri-Services Infrared Backgrounds Symp.,* Oct. 18-20, 1983, Mitre Corporation, pp. 216-223.

[3]M. S. Longmire, "A Final Technical Report on Calibration and Use of Clutter Data for Simulation," Western Kentucky University, Bowling Green, Ky., Contract No. N0014-84-C-2034 (undated draft report to Naval Research Laboratory).

[4]A. Hirschman, "BMAP Surface-Based Background Measurement Activities at Montauk Point, New York, August 1983," Naval Surface Weapons Center, White Oak, Silver Spring, Md., draft report, Nov. 1984.

[5]D. R. Jensen, "Data Report for IR BMAP Experiment, Aug. 1983," Naval Ocean Systems Center, San Diego, Cal., Memorandum DRJ:plk, Serial 5325/109, 14 Nov. 1983.

[6]J. R. Rice, *Numerical Methods, Software, and Analysis,* McGraw-Hill, New York, 1983.

[7]H. Ney, "A Dynamic Programming Technique for Nonlinear Smoothing", *Proc. IEEE International Conf. Acoustics, Speech and Signal Processing, Vol. 1,* Mar. 30-31 and Apr. 1, 1981, Atlanta, Ga., pp. 62-65.

[8]H. Ney, "A Dynamic Programming Algorithm for Nonlinear Smoothing," *Signal Processing,* Vol. 5, No. 2, Mar. 1983, pp. 163-173.

[9]F. Pasian and A. Crise, "Restoration of Signals Degraded by Impulse Noise by Means of a Low-Distortion Non-Linear Filter," *Signal Processing,* Vol. 6, No. 1, Jan. 1984, pp. 67-76.

[10]R. E. Boucher and J. P. Noonan, "Adaptive Detection and Removal of NonGaussian Spikes from Gaussian Data," *IEEE Trans. Pattern Analysis and Machine Intelligence,* Vol. PAMI-4, No. 2, Mar. 1982, pp. 132-136.

[11]R. A. Steinberg, "Infrared Background Sensor Characterization and Data Reduction," IRIS Specialty Group on Targets, Backgrounds, and Discrimination, Feb. 1-2, 1984, Camp Pendleton, Cal., pp. 209-231.

[12]R. A. Steinberg, "Elimination of Sensor Artifacts from Infrared Data," Report 8861, Naval Research Laboratory, Washington, D.C., 11 Dec. 1984.

[13]T. Pavlidis, *Structural Pattern Recognition,* Springer-Verlag, Berlin, Heidelberg, New York, 1977.

Appendix A
ALGORITHMS

Nine algorithms are provided in this Appendix (see Table A-1), in a block-structured format adopted from Ref. 13.

Individual steps within each algorithm are referenced with statements of the form:

Step (X.YY.Z),

where

X = algorithm number
YY = block number
Z = step number within block YY.

For example, step 2.13.1 corresponds to the statement (from PRMTVS):

Set q_1 = {(L/LL) + (R/RR)}.

The version of EDS evaluated statistically in the text did not include algorithms PK2MOD and MERGER. In addition, step 1.7 was inactive during the numerical validations.

Table A-1
Number, name, and a brief description of nine algorithms.

Number	Name	Description
1	EDS	Error detection and smoothing (executive routine)
2	PRMTVS	Detects error peak and edge primitives
3	PK2MOD	Modifies definition of PK2 primitive
4	BOUNDS	Develops estimates for artifact boundaries
5	ISOERR	Detects and smoothes isolated errors
6	NTRPL8	Single sample interpolation
7	MLTPLS	Detects and smoothes multiple-error artifacts
8	SMOOTH	Linear interpolation through multi-error artifact
9	MERGER	Merges proximate artifacts

Algorithm 1: EDS Error detection and smoothing, executive routine

```
Input:    Data points              {x(k), k=1,2,...,400}
Output:   Smoothed data points     {x(k), k=3,4,...,398}

Note:     This routine uses Algorithms PRMTVS, BOUNDS, ISOERR, and MLTPLS.

1.   Execute PRMTVS.
2.   If all primitives are null then terminate.
3.   Execute BOUNDS.
4.   Execute ISOERR: Corrects isolated errors.
5.   If any isolated errors were found in step 4 then do block 51.

          Begin block 51;
          1.  Execute PRMTVS.
          2.  If all primitives are null then terminate.
          3.  Execute BOUNDS.
          End block 51;

6.   Execute MLTPLS: Smoothes multiple-error artifacts.
7.   If corrected value differs from original value by 4 or less,
     restore the pre-correction value (optional).
```

---

[13] T. Pavlidis, *Structural Pattern Recognition*, Springer-Verlag, Berlin, Heidelberg, New York, 1977.

Algorithm 2 : PRMTVS Detects error peak and edge primitives

```
Input:      Data points {x(k), k=1,2,...,400}
Output:     Primitives {p(k), k=3,4,...,398} (cf. Eq. 13)
Note:  This routine uses Algorithm PK2MOD.

1.   For k=3,4,...,398 do block 11.
       Begin block 11 : Identify error peaks;
       1.   Obtain slopes LL, L, R, and RR (cf. Eq. 19 and Fig. 20).
       2.   Set p(k)=0.
       3a.  If |L| and |R| are both ≥ 4, and
        b.  if L·R < 0
            then do block 12.
              Begin block 12;
              1.  If LL·L ≤ 0, or if RR·R ≤ 0, or both,
                  then p(k) = L/|L|.
                  else do block 13.
                    Begin block 13;
                    1.  Set q1 = {(L/LL) + (R/RR)}.
                    2.  If q1 > Q1 = 1.85, then p(k) = L/|L|.
                    3.  Execute PK2MOD.
                    End block 13;
              End block 12;
       End block 11;

2.   For k=2,3,...,398, do block 21.
       Begin block 21 : Identify error edges;
       1.   Obtain slopes L, C, and R (cf. Eq. 20 and Fig. 21).
       2a.  If |C| ≥ 5, and
        b.  if |p(k)| and |p(k+1)| ≠ 1,
            then do block 22.
              Begin block 22;
              1.  If L·C≤0, and if R/C≤1,
                  then p(k) = p(k+1) = 2·C/|C|,
                  else do block 23.
                    Begin block 23;
                    1.  If R·C≤0 and if L/C≤1,
                        Then p(k) = p(k+1) = 2·C/|C|,
                        else do block 24.
                          Begin block 24;
                          1.  If L, C, and R have the same sign,
                              then do block 25.
                                Begin block 25;
                                1.  Set q2 = C/(L+R).
                                2.  If q2 > Q2 = 2,
                                    then p(k) = p(k+1) = 2·C/|C|.
                                End block 25;
                          End block 24;
                    End block 23;
              End block 22;
       End block 21;
```

Algorithm 3: PK2MOD Modifies definition of PK2 primitive

Input: a) Data points {x(i), i=1,2,...,400}
b) Sample number k
c) Initial value of p(k)

Output: Revised value of p(k)

Note: This routine is used by Algorithm PRMTVS.

1. If $|p(k)| = 1$, k=3, or k=398, then return.
2. Set $h_1 = x(k) - [x(k-2) + x(k+2)]/2$.
3. Do block 31.

Begin block 31: assure that peak is out of noise;
1. If $h_1 \geq 15$ do block 32;
Begin block 32: extend PK1 test to 7-point neighborhood;
1. Set $h_2 = x(k) - [x(k-3) + (k+3)]/2$.
2. Set $L = [x(k) - x(k-1)]$.
3. Set $q_3 = h_1/h_2$.
4. If $q_3 > Q_3 = 0.8$,
then set $p(k) = L/|L|$.
End block 32;
End block 31;

Algorithm 4: BOUNDS Calculates estimates for artifact boundaries

```
Input:    Primitives {p(k), k = 3,4,...,398}

Output:   Estimates for
          a)   number of artifacts, J
          b)   beginning sample number of jth artifact, {b(j), j=1,2,...,J}
          c)   final sample number of jth artifact, {f(j), j=1,2,...,J}

1.   Set n=0 to initialize the artifact counter.
2.   Set k=2 to initialize the p(k) pointer.
3.   While k<395 do block 11.

       Begin block 11;
       1.  Set k = k+1 to move the p(k) pointer along the scan line.
       2.  If p(k) = 0 do block 12.

             Begin block 12: an artifact has been found;
             1.  Set n = n+1 to increment the artifact counter.
             2.  Set b(n) = k and FLAG = 1.
             3.  Do block 13 for i = k, 395.

                   Begin block 13: search for end of artifact;
                   1.  Set f(n) = i.
                   2.  If p[f(n)+1] = p[f(n)+2] = p[f(n)+3] = 0 go to 12.7
                   End block 13;

             4.  Set f(n) = 398.
             5.  If p(398) = 0 set f(n) = 397.
             6.  If p(397) = p(398) = 0 set f(n) = 396.
             7.  If b(n) ≤ 4 or f(n) ≥ 397 set FLAG = 0.
             8.  If p[b(n)] · p[f(n)] · FLAG = -4 do block 14.

                   Begin block 14: shrink plateau artifact boundaries;
                   1.  Set b(n) = b(n) + 1.
                   2.  Set f(n) = f(n) - 1.
                   End block 14;

             9.    Set k = f(n)+3; search for next artifact will start at
                   f(n)+4.
             End block 12;
       End block 11;

4.   Set J = n.
```

Algorithm 5: ISOERR Detects and smoothes isolated errors

Input: a) Data points {x(k), k=1,2,...,400}
b) Artifact boundaries {b(j), f(j), j=1,2,...,J}

Output: Data points {x'(k), k=1,2,...,400}, with isolated errors corrected over the range 3 ≤ k ≤ 398.

Note: This routine uses Algorithms NTRPL8 and PRMTVS.

```
1.  Do block 11 for j=1,2...,J.
        Begin block 11;
        1.  Set L(j) = [f(j) - b(j) + 1] to estimate artifact length.
        2.  If L(j) ≤ 4 do block 12.

            Begin block 12 : Attempt single sample correction of artifact j.
            1.  Set i=0, B(j) = b(j), and F(j) = f(j).
            2.  If L(j)=1 do block 125.

                Begin block 125 : Expand search region;
                1.  If B(j) > 3 set B(j) = [B(j) - 1].
                2.  If F(j) < 398 set F(j) = [F(j) + 1].
                End block 125;

            3.  Do block 13 for n = B(j), F(j).
                Begin block 13 : Attempt single sample correction at n;
                1.  Set z = x(n) to save original value.
                2.  Apply NTRPL8 to {x(k)} to get y(n).
                3.  Set x(n) = y(n) to perform replacement.
                4.  Apply PRMTVS to {x(k)} to get {p'(k)}.
                5.  If p'(k)=0, B≤k≤F, do block 135.
                        Begin block 135 : Single correction at n is
                        viable;
                        1.  Set i=i+1.
                        2.  Set N(i) = n.
                        3.  Calculate a(i) from Eq. (22).
                        End block 135;

                6.  Set x(n) = z to restore original value.
                End block 13;

            4.  If i≠0 do block 14.
                Begin block 14 : Apply best of the i viable corrections.
                1.  Find m such that a(m) = min{a(1), a(2),...,a(i)}.
                2.  Set x[N(m)]) = y[N(m)], to perform correction.
                End block 14;
            End block 12;
        End block 11;
```

Definitions of Variables in Algorithm ISOERR

j	indexes the "current artifact", i.e., the artifact currently being considered for single sample correction.
J	maximum value of j; total number of artifacts in the waveform (input to Algorithm ISOERR, provided by Algorithm BOUNDS).
B(j),F(j)	bound the argument range of $x(\kappa)$ over which the search will be conducted for a single sample correction to artifact j.
n	running index of position within current artifact.
z	dummy variable for temporarily storing a pre-correction value of $x(\kappa)$.
y(n)	a replacement value for x(n), generated by interpolation from adjacent samples (output of Algorithm NTRPL8).
$p'(\kappa)$	sequence of error primitives for waveform in which sample x(n) has been replaced by interpolated value (output of Algorithm PRMTVS).
i	counts the number of viable single sample corrections within the current artifact.
N(i)	location of the ith viable single sample correction within the current artifact.
a(i)	measure of smoothness of sequence {x(k)} after ith viable isolated correction to the current artifact (cf. Eq. 22).
m	value of i that indexes the "best" of the viable isolated corrections, in that the resulting corrected sequence $\{x(\kappa)\}$ is "smoothest".

Algorithm 6: NTRPL8 Single sample interpolation

Input: a) Data points {x(k), k=1,2,...,400}
b) Location, n, of sample for which replacement is desired.

Output: Interpolated value, y(n), to replace the presumably defective value, x(n).

Note: This routine is used by Algorithm ISOERR.
Integer n is always within the range $3 \leq n \leq 398$.

1. Set y(n) = [x(n-1) + x(n+1)]/2, to obtain linear interpolation.
2. Set C = [x(n-2) + x(n+2)]/2, to obtain correction term.
3. Set y(n) = y(n) + [y(n)-C]/3, to obtain cubic interpolation.

Algorithm 7: MLTPLS Detects and smoothes multiple-error artifacts

Input: a) Data points {x(k), k=1,2,...,400} with isolated errors already corrected.
b) Primitives {p(k), k=3,4,...,398}
c) Estimated number of multiple-error artifacts J
d) Estimated artifact boundaries {b(j), f(j), j=1,2,...,J}

Output: Smoothed data points {x(k), k=3,4,...,398}

Note: This algorithm uses Algorithms PRMTVS, SMOOTH, and MERGER.

```
1.  Set j=0 and n=0.
2.  While j < J do block 11.
      Begin block 11;
      1.  Set j=j+1 and n=n+1.
      2.  Set B(n) = b(j) and F(n) = f(j).
      3.  Set m=1.
      4.  Set p'(2) = p'(399)=0.
      5.  If m≠0 do block 12.
            Begin block 12;
            1.  Set m=0.
            2.  Execute SMOOTH to interpolate linearly through multi-error
                artifact.
            3.  Apply PRMTVS to {x(k)} to obtain {p'(k)}.
            4.  Do block 21 for k = [B(n)-1], [F(n)+1].
                  Begin block 21: Count the non-null entries in p'(k);
                  1.  If p'(k)≠0 do block 22.
                        Begin block 22: Error artifact still exists;
                        1.  Set m=m+1.
                        2.  Set k_m=k.
                        End block 22;
                  End block 21;
            5.  If m≠0 do block 31.
                  Begin block 31: Artifact still exists - grow the
                  boundaries:
                  1.  Set B(n) = min{B(n),k_1}.
                  2.  Set F(n) = max {F(n),k_m}.
                  3.  Execute MERGER to merge proximate artifacts.
                  End block 31;
            End block 12;
      End block 11;
```

Algorithm 8: SMOOTH Linear interpolation through multi-error artifact

```
Input:    a)  Index n
          b)  Data points {x(k), k=1,2,...,400}
          c)  Boundaries {B(i), F(i), i=1,2,...,n}

Output:       Smoothed data {x(k), k=1,2,...,400}

Note:     This routine is used by Algorithm MLTPLS.

1.  If B(n)≤5 do block 11 for k=1,...,F(n).
        Begin block 11: blank beginning of line;
        1.  Set x(k) = x[F(n)+1].
        End block 11;
2.  If F(n)≥396 do block 21 for k=B(n),...,400.
        Begin block 21: blank end of line;
        1.  Set x(k) = x[B(n)-1].
        End block 21;
3.  If B(n)≥6 and F(n)≤395 do block 31.
        Begin block 31: linear interpolation from (B-1) to (F+1);
        1.  Set L(n) = F(n) - B(n) + 1.
        2.  Set δ = {x[F(n)+1] - x[B(n)-1]}/[L(n)+1].
        3.  Do block 32 for i=1, L(n).
              Begin block 32;
              1.  Set k = B(n)+i-1.
              2.  Set x(k) = x[B(n)-1] + i*δ.
              End block 32;
        End block 31;
```

Algorithm 9: MERGER Merges proximate artifacts

```
Input:    a)  Indices j and n
          b)  Boundary estimates {b(i), f(i), i=1,2,...J}
          c)  Improved boundaries {B(i), F(i), i=1,2,...n}

Output:   a)  Revised values of j and n
          b)  Revised values of B(n) and F(n)

Note:         This routine is used by Algorithm MLTPLS.

1.  Do block 11.
      Begin block 11;
      1.  Set C1=2 and C2=5.
      2.  Set L(n) = F(n)-B(n)+1.
      3.  If L(n)≥C2 set C1=C1+1.
      End block 11;
2.  If j≤(J-1) and F(n)≥[b(j+1) - C1] then do block 21.
      Begin block 21: Region merging, forward;
      1.  Set j=j+1.
      2.  Set F(n) = f(j).
      End block 21;
3.  If n≥2 and B(n)≤[F(n) + C1] then do block 31.
      Begin block 31: Region merging, backward;
      1.  Set n=n-1.
      2.  Set F(n) = F(n+1).
      End block 31;
```

THE JOHNS HOPKINS UNIVERSITY
APPLIED PHYSICS LABORATORY
LAUREL, MARYLAND

Appendix B

DERIVATION OF CUBIC INTERPOLATION FORMULA

Algorithm NTRPL8 generates a replacement value for $x(n)$, called $y(n)$, by interpolating from the four neighboring values: $x(n-2)$, $x(n-1)$, $x(n+1)$, and $x(n+2)$. In this appendix we derive the interpolation formula used by NTRPL8.

Since it is our intent to interpolate between four "known" data points, we select as our interpolation function a cubic polynomial,

$$y(k) = a + b(k-n) + c(k-n)^2 + d(k-n)^3, \qquad |k-n| \le 2, \quad \text{(B-1)}$$

where the coefficients (a, b, c, and d) are at present treated as "unknowns" in need of determination.

The four data values neighboring sample n are substituted into Eq. B-1 to obtain four equations:

$$x(n-2) = y(n-2) = a - 2b + 4c - 8d \quad \text{(B-2)}$$

$$x(n-1) = y(n-1) = a - b + c - d \quad \text{(B-3)}$$

$$x(n+1) = y(n+1) = a + b + c + d \quad \text{(B-4)}$$

$$x(n+2) = y(n+2) = a + 2b + 4c + 8d\,. \quad \text{(B-5)}$$

We next define quantities $\hat{x}$ and C as follows:

$$\hat{x} = [x(n-1) + x(n+1)]/2\,, \quad \text{(B-6)}$$

and

$$C = [x(n-2) + x(n+2)]/2\,. \quad \text{(B-7)}$$

From Eqs. B-3, B-4, and B-6,

$$\hat{x} = a + c\,. \quad \text{(B-8)}$$

From Eqs. B-2, B-5, and B-7,

$$C = a + 4c\,. \quad \text{(B-9)}$$

From Eqs. B-8 and B-9,

$$a = (4\hat{x} - C)/3 = \hat{x} + (\hat{x} - C)/3\,. \quad \text{(B-10)}$$

However, from Eq. B-1,

$$y(n) = a\,. \quad \text{(B-11)}$$

Thus, from Eqs. B-10 and B-11,

$$y(n) = \hat{x} + (\hat{x} - C)/3\,, \quad \text{(B-12)}$$

where

$$\hat{x} = [x(n-1) + x(n+1)]/2 \quad \text{(B-13)}$$

$$C = [x(n-2) + x(n+2)]/2\,. \quad \text{(B-14)}$$

The linear interpolation estimate, Eq. B-13, is formed in step 6.1; the correction term C, Eq. B-14, is formed in step 6.2; the cubic interpolation, Eq. B-12, is formed in step 6.3.

It also follows from Eqs. B-12 through B-14 that

$$y(n) = [-x(n-2) + 4x(n-1) + 4x(n+1) - x(n+2)]/6\,. \quad \text{(B-15)}$$

INITIAL DISTRIBUTION EXTERNAL TO THE APPLIED PHYSICS LABORATORY*

The work reported in TG 1355 was done under Navy Contract N00024-85-C-5301 and under Naval Research Laboratory Job Order Number 65-1299-A5. This work is related to JHU/APL task ZG50.

ORGANIZATION	LOCATION	ATTENTION	No. of Copies
DEPARTMENT OF DEFENSE			
Secretary of Defense	Washington, DC 20301	G. C. Kopesak, OUSDRE (ET)	1
Office of the Under Secretary of Defense, Research and Engineering	Washington, DC 20301	J. MacCallum	1
Defense Technical Information Center	Alexandria, VA 22314	Accessions	12
Defense Advanced Research Projects Agency	Arlington, VA 20331	S. Karp	1
DEPARTMENT OF THE NAVY			
CNO	Washington, DC 20350	OP 98	1
		OP 983	1
		OP 987	1
		OP 987B	2
		OP 35	1
		OP 35E	1
		OP 352L	1
		OP 355W	1
		OP 507D	1
Office of the Assistant Secretary of the Navy	Washington, DC 20350	E. Donaldson	1
		C. Kincaid	1
		R. E. Metry	1
		R. L. Rumpf	1
Office of Naval Research	Arlington, VA 22217	W. C. Condell, Jr., ONR-421	1
		E. Wegman, ONR-411SP	1
Office of Naval Technology	Arlington, VA 22209	ONT 07C	1
		ONT 0712	1
		ONT 0713	1
Naval Air Systems Command	Washington, DC 22202	AIR 320	1
		AIR 320B	1
		AIR 320D	1
		AIR 320R	1
		AIR 340J	1
		AIR 06	1
		AIR 0623	1
		Library, AIR 7226	2
		W. Goodell, PMA 253, JP-1	1
Naval Air Development Center	Warminster, PA 18974	S. Campana, 3011	1
Naval Electronic Systems Command	Arlington, VA 20360	D. M. Ferreira	1
		CDR T. French, 3W20	1
		R. Thompson, 6152A	1
Naval Postgraduate School	Monterey, CA 92940	T. Tao	1
Naval Research Laboratory	Washington, DC 20375	J. Blodgett, 6535	1
		T. G. Giallorenzi, 6500	1
		J. Kershenstein, 6520	1
		R. Lucke, 6520	1
		T. H. May, 6535	1
		M. J. McHugh, 6520	1
		J. V. Michalowicz, 6521	1
		R. A. Patten, 6530	1
		E. J. Stone, 6520	1
		H. H. Szu, 6535	1
		E. H. Takken. 6550.2	1
Naval Sea Systems Command	Washington, DC 22202	SEA 06	1
		SEA 06A	1
		SEA 06AT	1
		SEA 06AX	1
		SEA 06P	1
		SEA 06R	1
		SEA 62	1
		SEA 62B	1

Requests for copies of this report from DoD activities and contractors should be directed to DTIC. Cameron Station, Alexandria, Virginia 22314 using DTIC Form 1 and, if necessary, DTIC Form 55.

*Initial distribution of this document within the Applied Physics Laboratory has been made in accordance with a list on file in the APL Technical Publications Group.

INITIAL DISTRIBUTION EXTERNAL TO THE APPLIED PHYSICS LABORATORY

ORGANIZATION	LOCATION	ATTENTION	No. of Copies
DEPARTMENT OF THE NAVY (continued)			
		SEA 62R	2
		SEA 62R1	2
		SEA 62R11	1
		SEA 62R2	1
		SEA 62R4	1
		SEA 62R5	2
		SEA 62R52	13
		SEA 62Z	1
		SEA 62Z2	1
		SEA 62Z3	1
		SEA 62Z3B	1
		SEA 62Z31	2
		SEA 62Z31F	1
		Library, SEA 9961	2
		PMS 400	1
		PMS 400B	1
Naval Surface Weapons Center	Dahlgren, VA 22448	NSWC/DL D2W	1
		NSWC/DL G21	1
		NSWC/DL G22	1
		NSWC/DL G24	1
		Library	2
	White Oak, MD 20903	NSWC/WO CR42	1
		NSWC/WO G40	1
		NSWC/WO K22	1
		NSWC/WO N12	1
		B. Katz	1
		C. W. Larson	1
		D. Matlack, R-42	1
		S. K. Petropoulos	1
		Library	2
Naval Weapons Center	China Lake, CA 93555	3906	2
		3921	1
		3943	1
		S. Breil, 3943	1
		D. Kappelman, 39	1
		L. Wilkins, 39	1
		Library	2
Pacific Missile Test Center	Pt. Mugu, CA 93042	4045	1
U. S. Naval Academy	Annapolis, MD 21402	Director of Research	2
NAVPRO	Laurel, MD 20707		1
DEPARTMENT OF THE ARMY			
Ballistic Missile Defense Advanced Technology Center	Huntsville, AL 35807	ATC-R	1
		P. Boyd, ATC-RN	1
		HNV	1
		LEH	1
Missile Command	Huntsville, AL 35898	DRSMI OD	1
		DRSMI REI	1
		DRSMI RR	1
Night Vision and Electro-Optics Lab.	Fort Belvoir, VA 22060	J. A. Ratches,	1
		R. Buser	1
		F. T. Doepel	1
		W. R. Lawson	1
		J. Olson	1
DEPARTMENT OF THE AIR FORCE			
Air Force Armament Division	Elgin AFB, FL 32542	AD/XRG	1
Air Force Avionics Laboratory	Wright-Patterson AFB, OH 45433	V. Bart, AART	1
		R. Nieter	1
		R. Sanderson, WRP-1	1
Aeronautical Systems Division	Wright-Patterson AFB, OH 45433	YYM	1

INITIAL DISTRIBUTION EXTERNAL TO THE APPLIED PHYSICS LABORATORY

ORGANIZATION	LOCATION	ATTENTION	No. of Copies
CONTRACTORS			
Aerodyne Research	Billerica, MA 01821	J. Conant	1
		J. Draper	1
Aerospace Corporation	Los Angeles, CA 90009	C. Q. Ho	1
		P. Mazaika	1
		M. Schlessinger	1
		F. S. Simmons	1
		I. Spiro	1
		S. J. Young	1
Charles Stark Draper Laboratory	Cambridge MA 02139	T. H. Brooks	1
Eaton Corp., AIL Division	Melville, NY 11747	A. J. DiNardo	1
Environmental Research Institute of Michigan	Ann Arbor, MI 48107	J. R. Maxwell	1
General Dynamics	Pomona, CA 91769	D. W. Blay, 44-18	1
General Electric	Syracuse, NY 13221	B. A. Boerschig	1
		A. F. Milton	1
General Research Corp.	McLean, VA 22101	R. Zirkind	1
Grumman Aerospace	Bethpage, NY 11714	P. E. Barry, A08-35	1
		J. Krassner	1
		J. E. A. Selby	1
Honeywell, Inc.	Minneapolis, MN 55440	P. M. Narendra	1
		D. P. Panda	1
		A. Cox	1
Hughes Aircraft Co.	Canoga Park, CA 91340	A. V. Funari	1
		J. T. Hall	1
		R. L. Sendall	1
		R. Briggs	1
		W. A. Bundy	1
		J. Fisk	1
	El Segundo, CA	D. Rey	1
IBM Federal Systems Division	Westlake Village, CA 91361	J. C. Day	1
Institute for Defense Analyses	Alexandria, VA 22311	I. W. Kaye	1
		R. R. Legault	1
		H. G. Wolfhard	1
Jamison Science and Engineering	Washington, DC 20814	J. A. Jamison	1
Lockheed Missile and Space Co.	Palo Alto, CA 94304	J. D. Matthews	1
		L. H. Wald	1
		J. Davis	1
		N. Kulgein	1
Lou Massa	New York, NY 10021		1
Martin Marietta	Orlando, FL 32855	C. Layne	1
MIT Lincoln Laboratory	Lexington, MA 02173	H. Kleiman	1
Nichols Research Corp.	Huntsville AL 35802	R. J. Nichols	1
		D. L. Patz	1
North Carolina State University	Raleigh, NC 27709	W. E. Snyder	1
Northrop Corp.	Hawthorne, CA 90250	S. Zakanycz	1
Ontar Corp.	Brookline, MA 02146	J. Schroeder	1
Opti Metrics	Ann Arbor, MI 48104	R. Meredith	1
Optical Science Co.	Placentia, CA 92670	D. Fried	1
Polytechnic Institute of New York	Farmingdale, NY 11735	A. Papoulis	1
Rand Corporation	Santa Monica, CA 90406	L. G. Mundie	1
Raytheon Missile Systems Division	Bedford, MA 01730	I. Goldstein	1
		A. Krutchkoff	1
Riverside Research Institute	New York, NY 10036	R. G. Cestaro	1
Rockwell International	Downey, CA 90241	M. F. Sentovich	1
Science Applications	Ann Arbor, MI 48104	R. Turner	1
Sensor Systems Group	Waltham, MA 02154	M. Anapol	1

INITIAL DISTRIBUTION EXTERNAL TO THE APPLIED PHYSICS LABORATORY

ORGANIZATION	LOCATION	ATTENTION	No. of Copies
CONTRACTORS (continued)			
Spectral Sciences, Inc.	Burlington, MA 01803	L. S. Bernstein	1
SRI International	Menlo Park, CA 94025	J. Malick	1
Syn Optics Associates	Long Beach, CA 90803	R. Donohue	1
SWL, Inc.	McLean, VA 22102	W. Peters	1
Visidyne Co.	Burlington, MA 01803	T. F. Zehnpfennig	1
W. J. Schafer Associates	Arlington, VA 22209	D. H. Leslie	1
		W. S. Watt	1
Whittier College	Whittier, CA 90608	R. Rubin	1

END

DTIC

7-86