

AD-A050 684

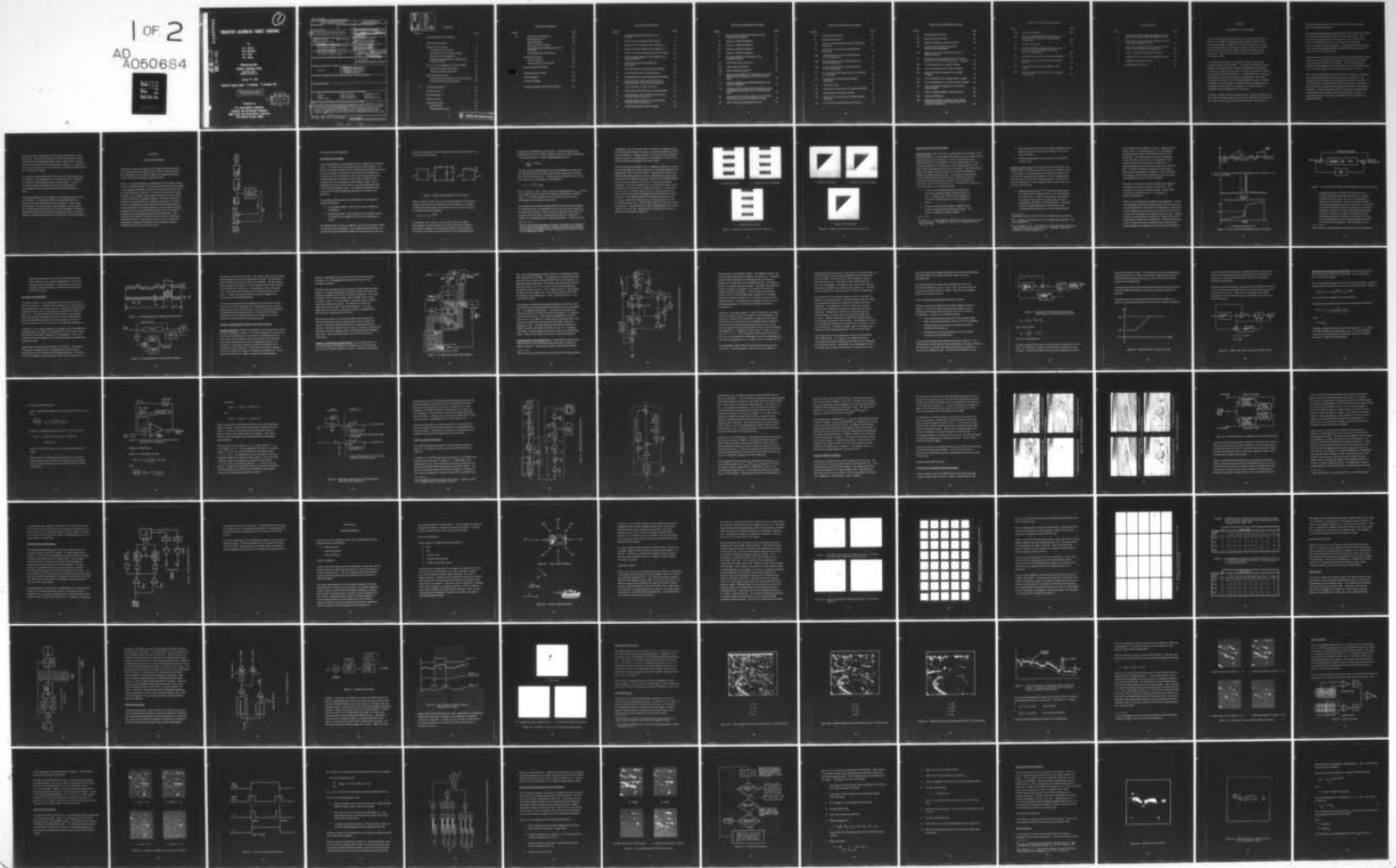
HONEYWELL INC MINNEAPOLIS MINN SYSTEMS AND RESEARCH --ETC F/G 17/5
PROTOTYPE AUTOMATIC TARGET SCREENER.(U)

JAN 78 D E SOLAND, P M NARENDRA, R C FITCH
78SRC6

DAAK70-77-C-0248
NL

UNCLASSIFIED

1 OF 2
AD
A050684



AD A 050684

①
52

PROTOTYPE AUTOMATIC TARGET SCREENER

By

D.E. Soland
P.M. Narendra
R.C. Fitch
D.P. Panda

AD No.
DDC FILE COPY

Honeywell
SYSTEMS & RESEARCH CENTER
2600 RIDGWAY PARKWAY
MINNEAPOLIS, MINNESOTA 55413

January 15, 1978

Quarterly Progress Report 21 September - 31 December 1977

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED.

DDC
RECEIVED
MAR 2 1978
D

Prepared for
U.S. Army Mobility Equipment
Research and Development Command,
Night Vision and Electro-Optics Laboratory
Fort Belvoir, Virginia 22060

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (WHEN DATA ENTERED)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NUMBER	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (AND SUBTITLE)		5. TYPE OF REPORT, PERIOD COVERED
6. Prototype Automatic Target Screener.		Quarterly Progress Report, 21 Sept 1977 - 31 Dec 1977 77
7. AUTHOR(S)		8. PERFORMING ORG. REPORT NUMBER
20 D.E./Soland, D.P./Panda P.M./Narendra, R/C./Fitch		14 78 SRC 6
9. PERFORMING ORGANIZATION NAME/ADDRESS		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
Honeywell Inc., Systems and Research Center 2600 Ridgway Parkway Minneapolis, Minnesota MN 55413		15 DAAK70-77-C-0248 (2a) 16 1E263710DK70/14 010CJ
11. CONTROLLING OFFICE NAME/ADDRESS		12. REPORT DATE
Night Vision and Electro-Optics Laboratory Fort Belvoir, Virginia 22060		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME/ADDRESS (IF DIFFERENT FROM CONT. OFF.)		15. SECURITY CLASSIFICATION (OF THIS REPORT)
		Unclassified
16. DISTRIBUTION STATEMENT (OF THIS REPORT)		15a. DECLASSIFICATION DOWNGRADING SCHEDULE
Unlimited		
17. DISTRIBUTION STATEMENT (OF THE ABSTRACT ENTERED IN BLOCK 20, IF DIFFERENT FROM REPORT)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (CONTINUE ON REVERSE SIDE IF NECESSARY AND IDENTIFY BY BLOCK NUMBER)		
Infrared Target screening Image processing FLIR Target recognition Real time Target cueing Pattern recognition Image enhancement		
20. ABSTRACT (CONTINUE ON REVERSE SIDE IF NECESSARY AND IDENTIFY BY BLOCK NUMBER)		
This report is the first quarterly progress report for contract DAAK70-77-C-0248, Prototype Automatic Target Screener. It describes results of the first three months of a five month design study. The objective of the effort is an automatic target screener to be used with thermal imaging systems employing common module components. This report covers the period from 21 September to 31 December 1977.		

402 349 50B

APPROVED BY	DTIC	DTIC	DTIC
	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ORIGINATOR	DTIC	DTIC	DTIC
JUSTIFICATION			
BY	DISTRIBUTION/AVAILABILITY CODES		
	DISL. AVAIL. CODE/ OR SPECIAL		
	A		

CONTENTS

Section	Page
I INTRODUCTION AND SUMMARY	1
II IMAGE ENHANCEMENT	4
Synthetic DC Restoration	6
Description of the Problem	6
Solutions to the DC Restore Problem	12
Description of the Algorithm	17
Hardware Implementation of Synthetic DC Restore Schemes	19
Local Area Gain/Brightness Control (LAGBC)	26
LAGBC Breadboard Description	35
LAGBC Breadboard Evaluation	39
Global Gain/Bias Control	40
PATS Global Gain/Brightness Control Approach	40
Global Gain/Bias Implementation	45
III TARGET SCREENING	48
Section Summary	48
Data Preparation	49
Imagery Sources	51
Image Segmentation	58
Autothreshold	58
Background Estimator	60

CONTENTS (concluded)

Section		Page
III	Sensitivity to Parameter β	65
	Threshold Selection	65
	Edge Thresholds	72
	Object Interval Extraction	73
	Bin Generation and Elementary Feature Extraction	78
	Smoothing of Object Boundaries	83
	Recognition Features	83
	Moment Features	83
	Fourier Boundary Descriptors	90
	System Simulation Software	101
IV	INTERFRAME ANALYSIS	111
	Frame Alignment	112
	Symbolic Matching	126
V	PLANS FOR NEXT REPORTING PERIOD	133

LIST OF ILLUSTRATIONS

Figure		Page
1	Functional Image Enhancement System on PATS	5
2	Rect Function Response of an RC Circuit	7
3	Synthetic DC Restoration of Test Target No. 1	10
4	Synthetic DC Restoration of Test Target No. 2	11
5	Line-to-Line Pixel Differences from AC Coupling	15
6	Functional Block Diagram of the Histogram DC Restore Algorithm	16
7	The Analog Approach to Synthetic DC Restoration	18
8	All-Analog Scheme Conceptual Block Diagram	18
9	DC Restoration Circuit: Block Diagram	21
10	All-Analog Scheme Implementation Block Diagram	23
11	Functional Flow Description of Local Area Gain/Brightness Control (LAGBC) Algorithm	27
12	Local Area Gain vs. $\frac{m}{\sigma}$ Curve Used	28
13	LAGBC with Linear Recursive Low Pass Filters	29
14	Implementation of Two-dimensional Recursive Low Pass Filter (Approach 1)	32
15	Separable Implementation of Two-dimensional Low Pass Filter (Approach 2)	34
16	LAGBC Implementation Block Diagram	36

LIST OF ILLUSTRATIONS (continued)

Figure		Page
17	Separable Two-dimensional Recursive Filter Implementation Diagram	37
18	Results of LAGBC Breadboard	41
19	Results of LAGBC Breadboard	41
20	Results of LAGBC Breadboard	42
21	Results of LAGBC Breadboard	42
22	Functional Schematic of Global Gain and Brightness Control	43
23	Global Gain/Bias Control Unit	46
24a	Aspect Angle Dependence	50
24b	Elevation Angle Dependence	50
25	Examples of Banding Due to Differences in the Two Channel Bias Variations in the Krebs FLIR Data Base	53
26	Result of Applying the Debanding Algorithm to the Frames in Figure 25	53
27	Examples of FLIR Image Frames Digitized from the Honeywell FLIR Tape for the PATS Training Data Base	54
28	Examples of FLIR Frames Digitized from the NVL Video Tape for the PATS Training Data Base	56
29	PATS Target Screening System Concept	59

LIST OF ILLUSTRATIONS (continued)

Figure		Page
30	Autothreshold Structure	61
31	Background Estimator	62
32	The F1 Controlled Switch Helping in Background Estimation	63
33	The Effect of the Switch in the Background Estimation	64
34a	Brights Obtained by the Autothreshold with $\beta = 0.80$ (No Switch)	66
34b	Brights Obtained by the Autothreshold with $\beta = 0.90$ (No Switch)	67
34c	Brights Obtained by the Autothreshold with $\beta = 0.95$ (No Switch)	68
35	An Actual Intensity Profile Across a FLIR Image Cross Section	69
36	Comparison of Local and Global Bright Thresholds	71
37	Edge Filter Mask	72
38	Comparison of Global and Local Edge Thresholds	74
39	Object Interval Generation Concept	75
40	Alternate Turn on Criterion for Object Interval Extraction	77
41	Two Alternate Interval Extraction Criteria	79

LIST OF ILLUSTRATIONS (continued)

Figure		Page
42	Bin Generation Flowchart	80
43a	Object Intervals Extracted	84
43b	Boundary of Objects Traced by the Bin	85
44a	Boundary Representation Obtained by Autoscreener Segmentation	91
44b	Angle vs. Arc Length Representation of the Segment Boundary	91
45	Derivation of the Equispaced Values of $\theta(l)$ as a Function of l , from the Set $\{(\theta_1, l_1), \dots, (\theta_N, l_N)\}$	93
46a	$\theta(l)$ Representation for a Right Triangle	95
46b	$\varphi'(l)$ Representation for the Right Triangle	96
46c	Fourier Harmonic Amplitudes for the Right Triangle	97
47a	$\theta(l)$ Representation for a Rotated Right Triangle	98
47b	$\varphi'(l)$ Representation for the Rotated Right Triangle	99
47c	Fourier Harmonic Amplitudes for the Rotated Right Triangle	100
48	PATS Screening Simulation Software Outline	103
49a	Results of Simulation	104
49b	Results of Simulation Candidate Object Outline Extracted by the Bin Generation Program from Intervals in Figure 49a	105

LIST OF ILLUSTRATIONS (concluded)

Figure		Page
50a	Results of Simulation	106
50b	Results of Simulation Outline Extracted by the Bin Generation Program from the Intervals in Figure 49a	107
51a	Results of Simulation	108
51b	Results of Candidate Object Outline Extracted by the Bin Generation Program from the Intervals in Figure 51a	109
52	The Five Input Frames for Interframe Analysis	115
53	The Objects Extracted by ATSS from the Input Frames	118
54a	Translation Histogram for Aligning Frames 1 and 2	123
54b	Filtered Translation Histogram Corresponding to Figure 54a	124

LIST OF TABLES

Table		Page
1	Proportion of Target Types and Aspects in Krebs Data (from the Honeywell FLIR) Digitized to Date	57
2	Proportion of Target Types and Aspects in the NVL FLIR Imagery Digitized to Date	57
3	Some PATS Features on Three Targets and Four Clutter Objects Extracted by Simulation	102
4	Frame Translations by the Three Methods	125
5	Alignment Errors and Nearest Neighbor Matches	126
6	Symbolic Matching by Location	131
7	Symbolic Matching by Size	131

SECTION I

INTRODUCTION AND SUMMARY

This is the first quarterly technical progress report for Contract No. DAAK 70-77-C-0248, Prototype Automatic Target Screener (PATS). The report describes results of the first three months of a five month Phase I design study for an automatic target screener that can operate with first generation thermal imagers employing common module components. The period covered by this report is 21 September to 31 December 1977.

The objective of this effort is to produce a design for a prototype automatic target screener (PATS). The screener will reduce the task loading on the thermal imager operator by detecting and recognizing a limited set of high priority targets at ranges comparable to or greater than those for an unassisted observer. A second objective is to provide enhancement of the video presentation to the operator. The image enhancement includes: 1) automatic gain/brightness control, to relieve the operator of the necessity to continually adjust the display gain and brightness controls; and 2) DC restoration, to eliminate artifacts resulting from AC coupling of the IR detectors.

The report consists of three principal sections. Section II describes the effort under the image enhancement part of the study; Section III describes the target screener design activities; and Section IV reports on the

interframe analysis task. Plans for the next three month reporting period are included in Section V.

The image enhancement portion of PATS will consist of circuitry to operate on the Common Module FLIR (MODFLIR) video output signal. This circuitry will provide global gain and bias control in the form of feedback to the MODFLIR to maintain the signal within the dynamic range of the electro-optical multiplexer. The global gain and bias control circuit preliminary design has been completed and will be implemented upon receipt of the GFE MODFLIR.

Image enhancement will also include local area gain and brightness control to enhance local variations of contrast and compress the overall scene dynamic range to match that of the display. This circuitry has been completed and examples of its performance on video taped thermal image data are included, along with the circuit description, in Section II.

The third image enhancement circuit is for DC restoration, to eliminate the streaking associated with loss of line-to-line correlation on the displayed image because of the AC coupling of the detector channels. We have modified our original implementation approach to DC restoration which was based on measuring the histogram of the line-to-line intensity differences. The new scheme provides a much simpler implementation of basically the same approach and is also described in Section II. A breadboard version of the simpler concept has been designed and is being constructed.

Sections III and IV summarize the effort to date on the target screener design task. Section III describes the results of the subtasks involved with detecting and recognizing targets within a single frame of imagery. Section IV on interframe analysis documents research on new algorithms to improve the target screener performance by correlating image features over a sequence of frames.

The target screening tasks reported in Section III are data preparation and analysis, image segmentation, and feature extraction. Remaining are the object classification and target decision tasks. This effort is somewhat behind the planned schedule primarily because of increased effort under the data preparation task over the planned effort.

The data preparation task was expanded to include digitization of taped FLIR imagery. A total of 260 frames of FLIR imagery has been digitized, annotated, and debanded where necessary. These images contain tanks, armored personnel carriers (APCs), and some trucks and represent less than 50 percent of the estimated quantity of imagery required for target screener design, training, and testing. Additional digitized imagery will be added to this data base as it becomes available.

SECTION II

IMAGE ENHANCEMENT

This section reports the progress on the image enhancement tasks of the Prototype Automatic Target Screener (PATS). Specifically, the tasks addressed are: synthetic DC restoration, global gain and bias control, and local area gain and brightness control.

Figure 1 is a functional diagram showing these three functions in PATS. On the DC restoration task, we have analyzed the hardware requirements for the histogram technique of synthetic DC restoration (similar to the NVL scheme) which we proposed. An alternate scheme has also been developed. This scheme is all-analog and appears to be more readily implementable than the histogram approach. Preliminary hardware designs for both approaches and tradeoffs are discussed. The local area gain and brightness function has already been breadboarded and is now being tested. We include schematics of this design and sample imagery which has been processed through this real time hardware. The global gain and bias approaches are discussed briefly. This technique is not computer simulated because it involves modeling the MODFLIR hardware. At this point we feel that the microprocessor-based design proposed for the global control is flexible enough to breadboard as soon as we acquire the FLIR.

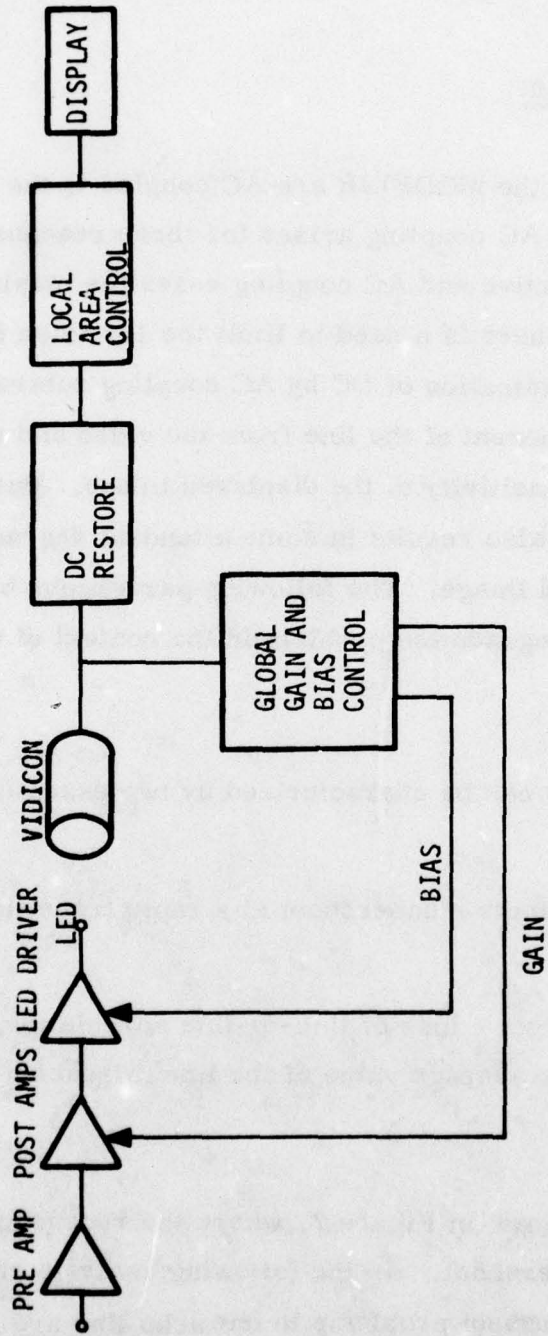


Figure 1. Functional Image Enhancement System on PATS

SYNTHETIC DC RESTORATION

Description of the Problem

The parallel detectors in the MODFLIR are AC coupled to the FLIR electronics. The practice of AC coupling arises for three reasons: 1) the detectors are photoconductive and AC coupling eases the biasing considerations on the detector; 2) there is a need to limit the $1/f$ noise in the detectors; and, 3) the elimination of DC by AC coupling subtracts the average background component of the line from the video and thus increases the contrast sensitivity of the displayed image. But AC coupling of the detectors also results in some attendant degradations on the resultant displayed image. The following paragraphs briefly review the AC coupling degradation problem in the context of the MODFLIR.

AC coupling degradations can be characterized by two essentially separate phenomena:

- The transient effects - undershoot at a rapid transition of temperature.
- Steady state effects - loss of line-to-line correlation because of the loss of the average value of the line (streaking and droop).

The transient effect is shown in Figure 2, where the rect function suffers a droop and an equal undershoot. As the following analysis shows, the transient droop and undershoot problems in one scan line are minor for

the long time constants of the RC coupling circuitry in the MODFLIR, and do not need to be corrected.

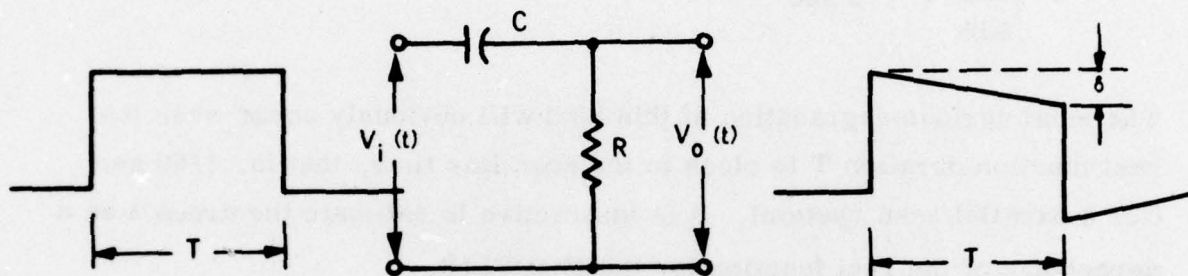


Figure 2. Rect Function Response of an RC Circuit

Figure 2 shows the basic RC circuit that approximates the low frequency response of the entire FLIR electronics from the detector to the light emitting diode (LED) stage. Step response of such a circuit is given by a decaying exponential as follows:

$$U_o(t) = U(t - t_o) e^{-t/RC}$$

The response of the circuit to a rect function (that more closely approximates a hot spot) can be derived from the unit step response and is shown in Figure 2. The important feature in this response is the droop δ of the top of the rect function and the corresponding equal

undershoot δ that follows the rect function. The time constant of the equivalent RC circuit ($T = RC$) is related to the lower 3 dB cutoff frequency of the system (detector to LED) of approximately 8 Hz by*

$$\tau = \frac{1}{f_{3dB}} = 1/8 \text{ sec}$$

The most serious degradation of this kind will obviously occur when the rect function duration T is close to the scan line time, that is, $1/60$ sec (for a parallel scan system). It is instructive to estimate the droop δ as a percentage of the rect function for the MODFLIR.

$$\delta = (1 - e^{-T/\tau}) \times 100\%$$

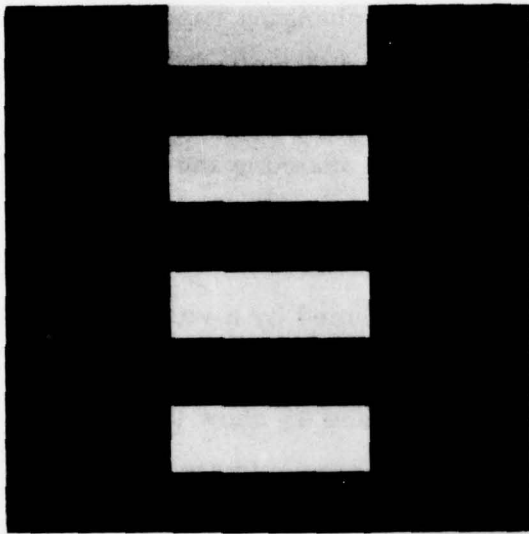
For $T = 1/60$ sec, and $\tau = 1/8$ sec, the above expression gives $\delta_{\max} = 12.5\%$, which indicates that the transient DC droop and undershoot is not at all severe. Very seldom could we have the extreme case of a hot spot extending the entire length of the scan line.

More serious by far than the transient effects is the fact that the parallel AC coupled detectors in the MODFLIR are not DC restored or clamped at the end of each scan line. When the image is viewing a stable scene, the average value of the video in each channel becomes zero within a few fields' time, independently of the neighboring channels. This has serious

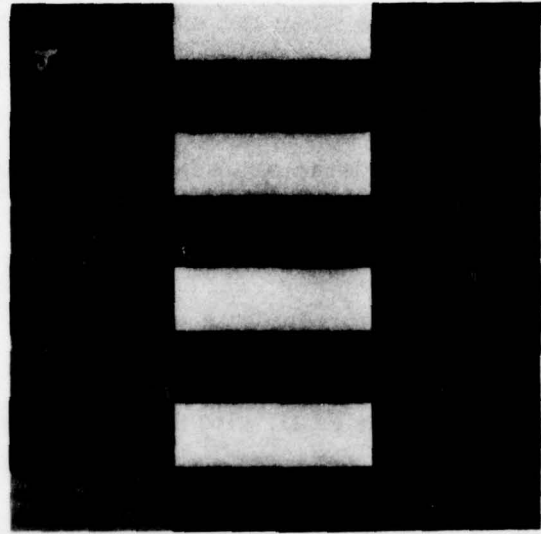
* This is a conservative estimate of the lower 3 dB cutoff of the MODFLIR. The true value is somewhat lower. However, the droop is even smaller at 3 Hz than at 8 Hz and this analysis, showing that the droop in one scan line is insignificant, holds.

consequences when the average scene temperature is changing rapidly in a direction perpendicular to the scan, because this difference is then lost in the display. A classic example is the loss of horizon definition because the detectors scanning the cold sky and those scanning the hot ground yield the same average video signal values.

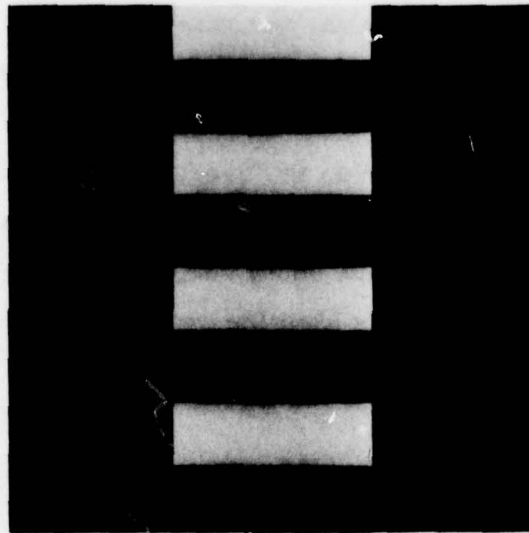
The second steady state effect is the "streaking" produced by a very hot or cold target against a uniform background. This again happens in the parallel scan FLIR with no DC restore. Figures 3b and 4b show this effect on test targets (Figures 3a and 4a) degraded to approximate the loss of DC on the individual scan lines (along with the transient effects). A common bias is added to all the lines to make the video display compatible. Note the streaking evident in the degraded images. This represents the most severe AC coupling degradation, as the presence of a hot spot on a scan line can lower the rest of the scan line video below blacker than black on the display and obscure any detail present in the line. This happens because the average value of a scan line in the top of Figure 4a is lower than the average value of a line with the white-hot portion on it. With the loss of the DC, the average values of these lines in Figure 4b are now equal, depressing the lower line with respect to the upper one. This is also the reason for the shading on the hot and cold targets in Figure 4b.



a. Original test pattern

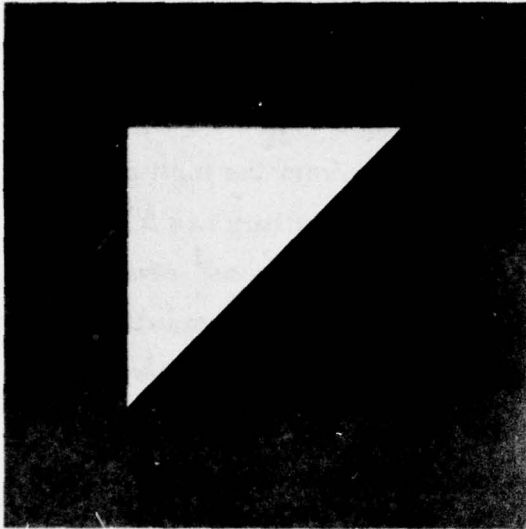


b. Degraded due to AC coupling

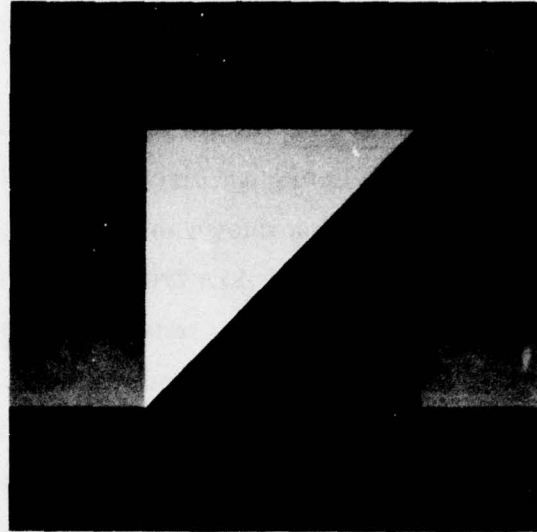


c. Synthetic DC restoration

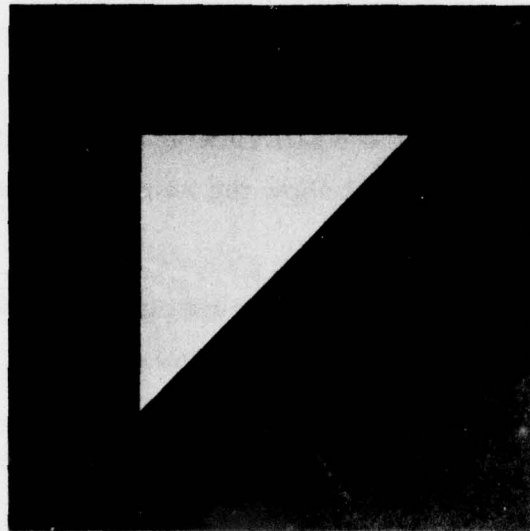
Figure 3. Synthetic DC Restoration of Test Target No. 1



a. Original test pattern



b. Degraded due to AC coupling



c. Synthetic DC restoration

Figure 4. Synthetic DC Restoration of Test Target No. 2

Solutions to the DC Restore Problem

Inverse Filtering--This is the most tempting solution at first sight. This involves designing an inverse sampled data filter to invert the high pass RC transfer function shown in Figure 2. Since this inverse filter has a singularity at very low frequencies (DC) the proposed solutions¹ are not strict inverse filters, but simply boost the low frequencies so that the lower 3 dB cutoff frequency is now lower than before. This will lessen the droop in the single scan line and increase the time of settling (on a stable scene lasting several frames). But complete recovery of the very low frequency information necessary to maintain vertical DC correlation is impossible without making the system unstable and extremely sensitive to low frequency noise. We discarded this approach to DC restoration for the following reasons:

- It will not cure the steady state loss of vertical DC correlation on scenes lasting more than a few frame times. Therefore, the FLIR viewing a test pattern, for example, would still show the same streaking as before.
- It requires access to each detector channel, because each channel has to have its own inverse filter. This involves modifying the common module.

¹T. Noda, et al., "Final Report for Experimental Development of a FLIR Sensor Processor," NORTHROP, Contract No. DAAG53-76-C-0188, January 31, 1977.

- Each channel inverse filter has to exactly compensate for the capacitor/resistor network on that channel. Errors will lead to instability.
- It will make the system very sensitive to $1/f$ noise and to amplifier drift.

Synthetic DC Restoration--This approach uses the properties of the thermal scene to artificially restore the vertical correlation lost due to AC coupling. We have analyzed two techniques for synthetic DC restoration through computer simulation. One is a histogramming approach and the second is an all-analog approach. Both use the fact that background in a scene changes slowly in FLIR scenes from one scan line to the next.

1. Histogramming Approach^{2,3} to Synthetic DC Restoration--
We recognize that the background in a scene varies slowly from line to line. In the AC coupled video, the presence of a significant hot target in one line causes that line to be depressed with respect to the previous line. Therefore, the pixel-to-pixel differences of the two lines will be predominantly distributed at or near the average DC shift

This technique with minor differences was independently conceived by:

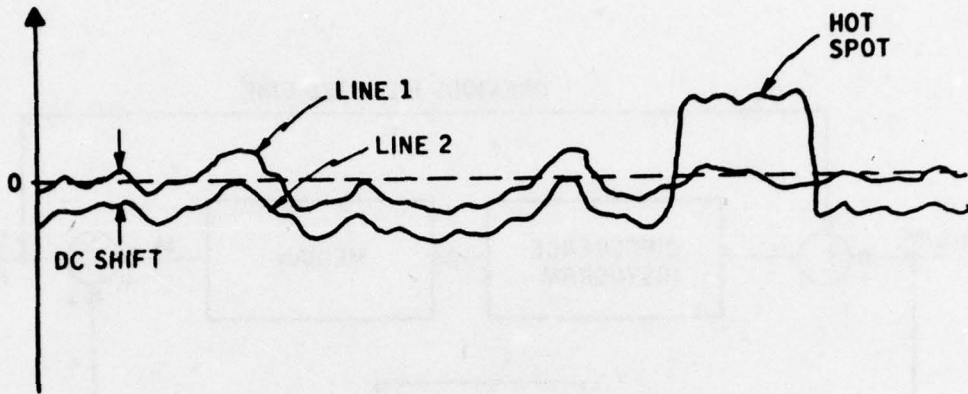
²P. K. Raimondi, "Pseudo-DC Restoration Using Histogramming," NVL, (patent applied for).

³P. M. Narendra, et al., "Final Report on Automated Image Enhancement Techniques for Second Generation FLIR," Honeywell, Contract No. DAAG53-76-C-0195, December 1977.

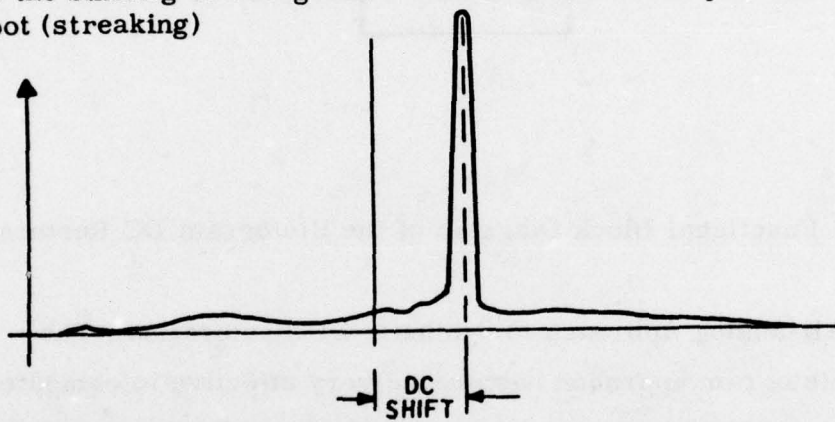
of the second line with respect to the first. Figures 5a and b illustrate this. By recognizing the peak in the histogram, we can identify the shift of the DC level and add it to the second line. The second line now serves as the reference for the next line, and so on down. Peak detection of this histogram may not be very suitable, because this is a sparse histogram (512 levels and about 500 pixels in the scan line). Therefore, finding the true peak of the histogram would involve smoothing of the histogram. Also, when there is no change in the DC levels between successive scan lines, peak seeking will not give a definitive solution.

Actually, we need only to detect the median of the histogram (see Figure 5c) to get an estimate of the DC shift. The median is a more robust measure because it finds the peak if one exists, and yields zero (i. e., no shift) if no significant peak exists.

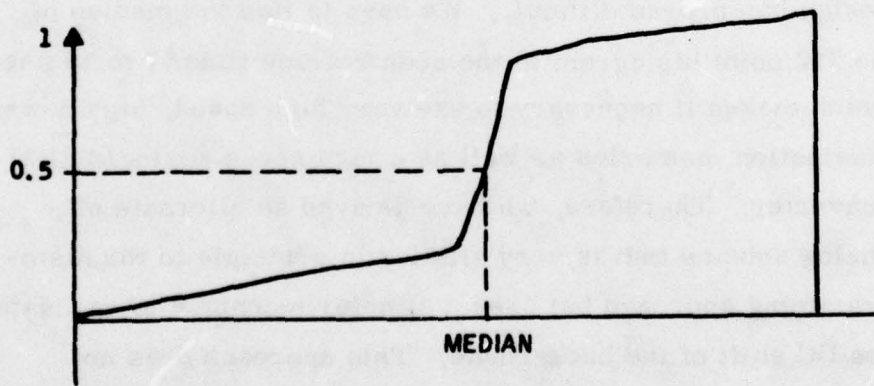
Figure 6 is a functional block diagram of this algorithm. A difference histogram is formed by subtracting the current line from the previous filtered line. The median of the histogram is added to the current delayed line to get the current synthetic DC restored line. The line-to-line background correlation is thus restored. Figures 3c and 4c are the degraded test patterns of Figures 3b and 4b restored using a simulation of this algorithm. We note that the line-to-line correlation has been completely restored and even the gradual shading in the middle of the pattern has been eliminated.



a. Shows the shifting of background level because of the presence of a very hot spot (streaking)



b. Histogram of the difference of line 1 and line 2



c. Cumulative histogram of b

Figure 5. Line-to-Line Pixel Differences from AC Coupling

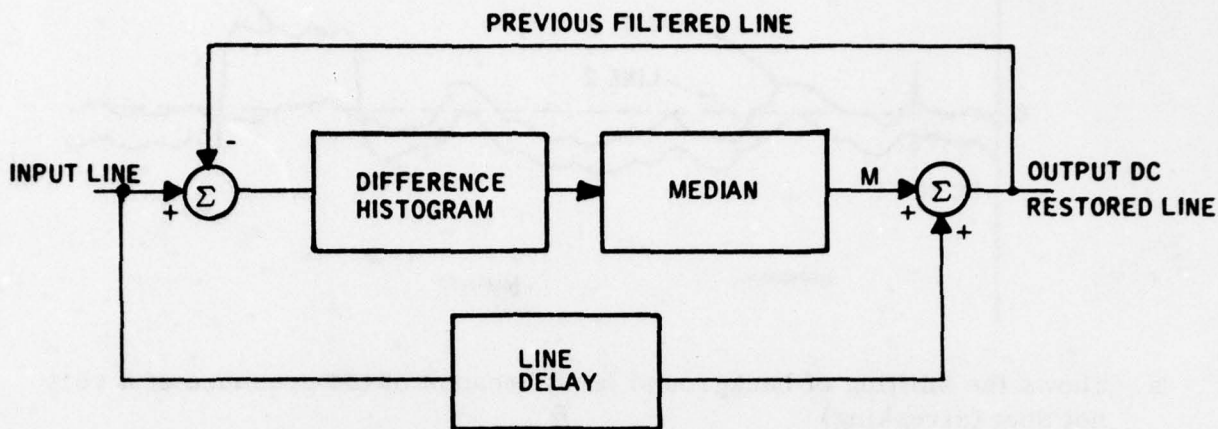


Figure 6. Functional Block Diagram of the Histogram DC Restore Algorithm

2. All-analog Approach to Synthetic DC Restoration--The histogram approach has proved very effective in computer simulations as seen above, but real time implementation design has proved difficult. We have to find the median of the 512 point histogram in the scan retrace time (7 to 10 μsec)*, which makes it necessary to use very high speed, high power dissipation memories as well as a high speed analog/digital converter. Therefore, we have derived an alternate all-analog scheme that is very similar in principle to the histogramming approach but uses a simpler technique to estimate the DC shift of the background. This approach does not

* Seven μsec for an 875-line system and 10 μsec for a 525-line system.

require histogramming and requires only analog integrators, which makes for a very simple implementation. The concept has been computer simulated, and works just as well as the histogram on the test patterns on which NVL has proposed to exercise the scheme.

Description of the Algorithm

Referring to Figure 7, a hot target appears on Scan Line 2 that was not present in Scan Line 1. Because of AC coupling, the average value of the video on Scan Lines 1 and 2 is now the same. Therefore, the background intensities on Scan Line 2 have dropped by a constant offset relative to Scan Line 1. This shift causes the lines with the hot target to appear darker than the rest, resulting in a streak. We need to estimate this shift in the background and add it to the second scan line so that the streak around the hot (or for that matter, very cold) target disappears.

In Figure 7, if we average the positive and negative parts of the difference signal separately, the offset we want should be one of these two averages. We need the average that corresponds to the background. This is the average that lasts the longest over the scan line (targets are smaller than background areas).

Figure 8 shows the functional schematic of this process. The positive and negative averaging is done by integrating over one scan line in analog integrators and dividing by the corresponding counts. The choice of the two averages is made according to which of the two counters (+ or -)

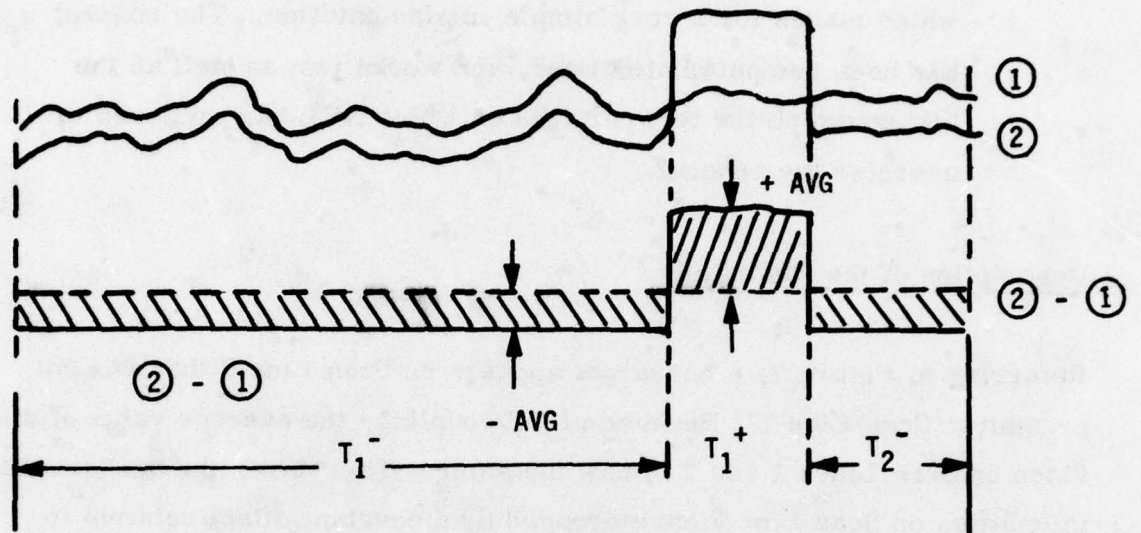


Figure 7. The Analog Approach to Synthetic DC Restoration

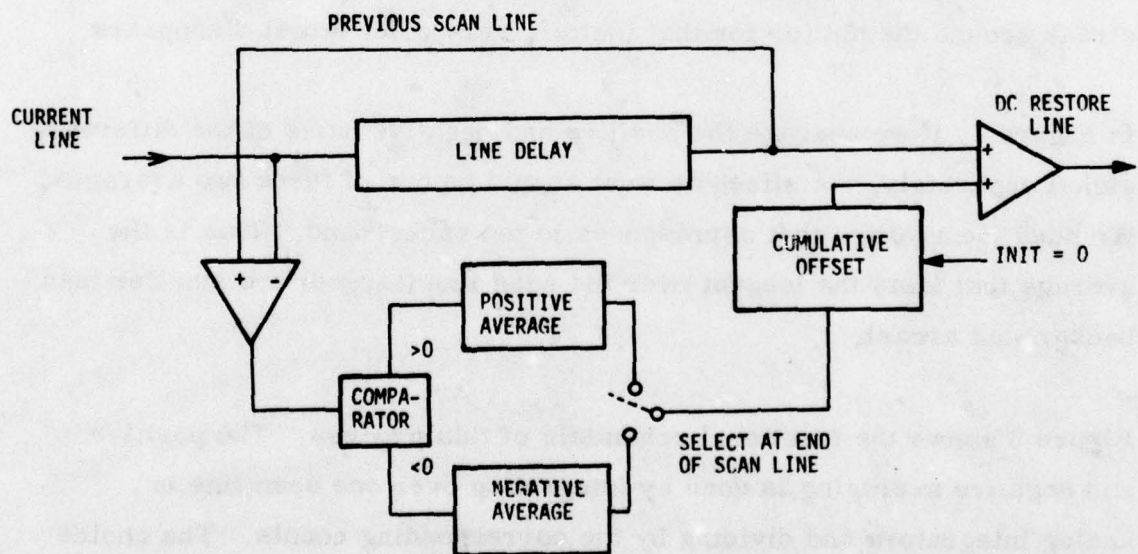


Figure 8. All-Analog Scheme Conceptual Block Diagram

is greater at the end of the scan line. This shift is added to the cumulative offset (initialized to zero at the beginning of the field). The cumulative offset is added to the current scan line (now delayed). Note that only the analog division and selection of the two averages need be made at the end of the scan line (in 7 to 10 μ sec). In the histogram scheme, we would need to step through all 512 histogram bins to find the median in this period. Further, the scheme involves no analog-to-digital (A/D) converters and very little digital logic.

The scheme assumes that the hot spots appear on less than one-half of the scan line (a valid assumption) and the background does not change rapidly from one scan line to the next. These are the same assumptions the original histogram approach made. The scheme was computer simulated on the two candidate test patterns in Figures 3 and 4 and works just as well as the histogramming approach.

Hardware Implementation of Synthetic DC Restore Schemes

Histogram Approach--The functional diagram of the histogram DC restore algorithm was shown in Figure 6. The basic functions are to compute the pixel-by-pixel differences of a line from the corresponding pixels on the previous DC restored line, and find the median of the differences. The median represents the DC shift the background suffered in the presence of a major scene change on successive scan lines, and is added back to the current line to restore the line-to-line correlation. One way to compute the median is to histogram the differences. At the end of the scan line, the median is computed by accumulating the

histogram sequentially in bins until one-half the total line population is reached. The corresponding bin then gives the median of the difference distribution.

This function can be performed in the hardware implementation shown in Figure 9. The input analog signal is differenced from the previous DC restored (and delayed) line and A/D converted to eight bits. This result addresses a 256 x 8 random access memory (RAM) which is used as a histogram counter. To avoid adders, a read only memory (ROM) is used to increment the bin counts. The ROM forms a 256 entry lookup table with the output simply being the input +1. The read update and write cycles for a given point should be completed before the next point is converted by the A/D unit.

This process continues for each of about 500 samples along a line. When the blanking and retrace time at the line's end is reached, the results of successive RAM are added until the histogram median is reached. This value is latched onto the inputs of a digital-to-analog (D/A) converter and the D/A output is added to all samples along the delayed input video. The process continues recursively as each new line comes along.

Analysis of the Histogram Implementation--It can be shown that for a 512-point histogram with eight bits of resolution, an A/D conversion speed of 10 to 16 MHz is needed for the 525- to 875-line video systems.

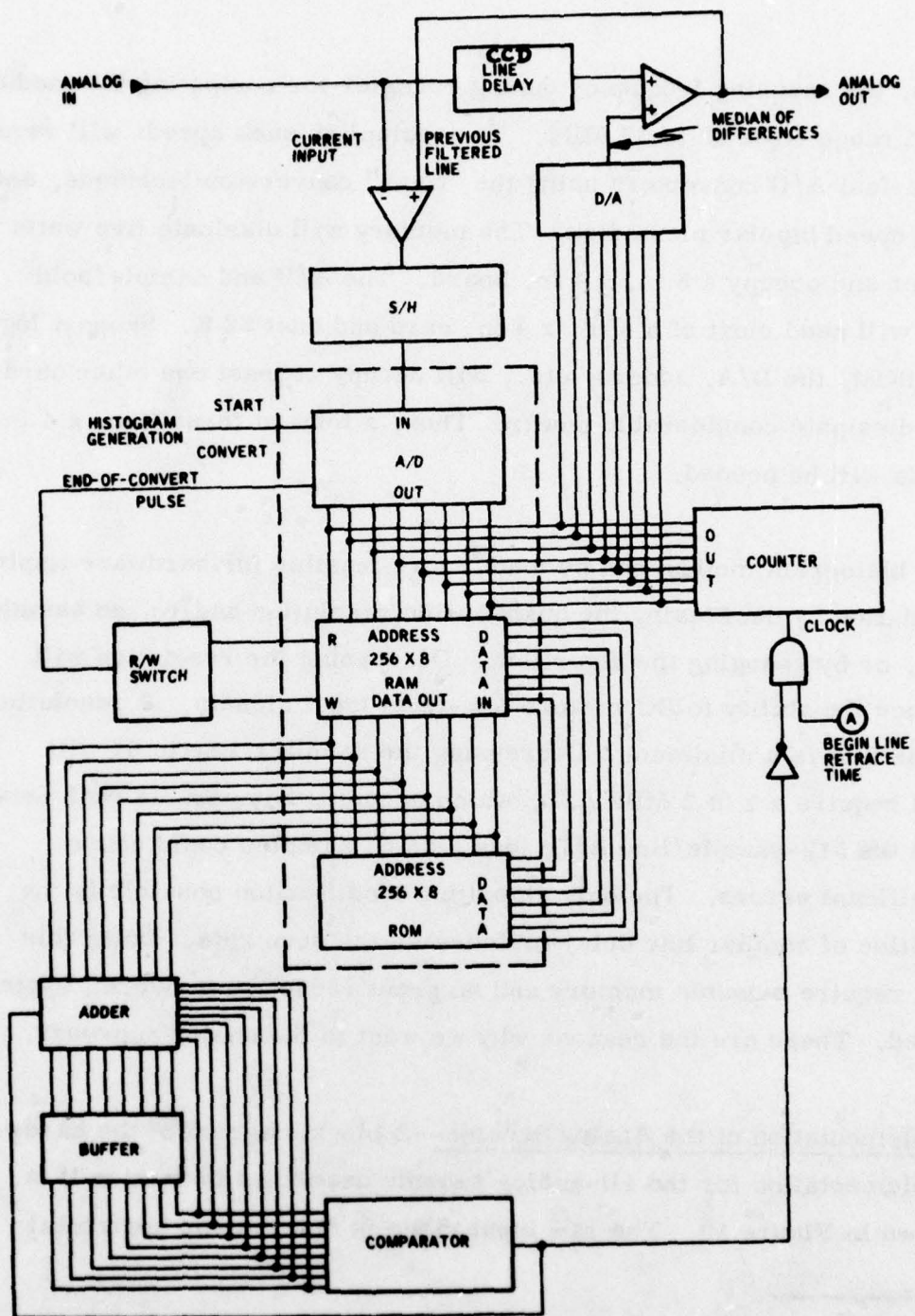


Figure 9. DC Restoration Circuit: Block Diagram

Also, the scanning frequency during retrace* for computing the median must range from 25 to 37 MHz. To accomplish such speeds will require ultra-fast A/D converters using the "flash" conversion technique, and high speed bipolar memories. The memory will dissipate five watts of power and occupy a 6 in. x 4 in. board. The A/D and sample/hold unit will need most of a 6 in. x 4 in. card and cost \$2 K. Support logic, the ROM, the D/A, adders, etc., will occupy at least one other card and dissipate considerable power. Thus, a total of three 6 in. x 4 in. cards will be needed.

The histogram method can be made more feasible for hardware implementation by decreasing the quantization resolution and/or the samples/line, or by changing the algorithm. Decreasing the resolution will reduce the ability to DC restore low-level input signals. A resolution of six bits is a minimum. Decreasing the samples/line to 64 will still require a 1 to 2 MHz A/D, which occupies comparable real estate with the 512-sample/line A/D; less than 64 samples could cause significant errors. The only algorithm modification possible is the addition of another line delay to decrease the scan rate. Doing this will require a double memory and no great reduction in overall system speed. These are the reasons why we went to the analog approach.

Implementation of the Analog Scheme--A block diagram of the hardware implementation for the all-analog scheme described in Section II is shown in Figure 10. The raw input video is delayed one horizontal

* Seven μ sec for an 875-line system and 10 μ sec for the 525-line system.

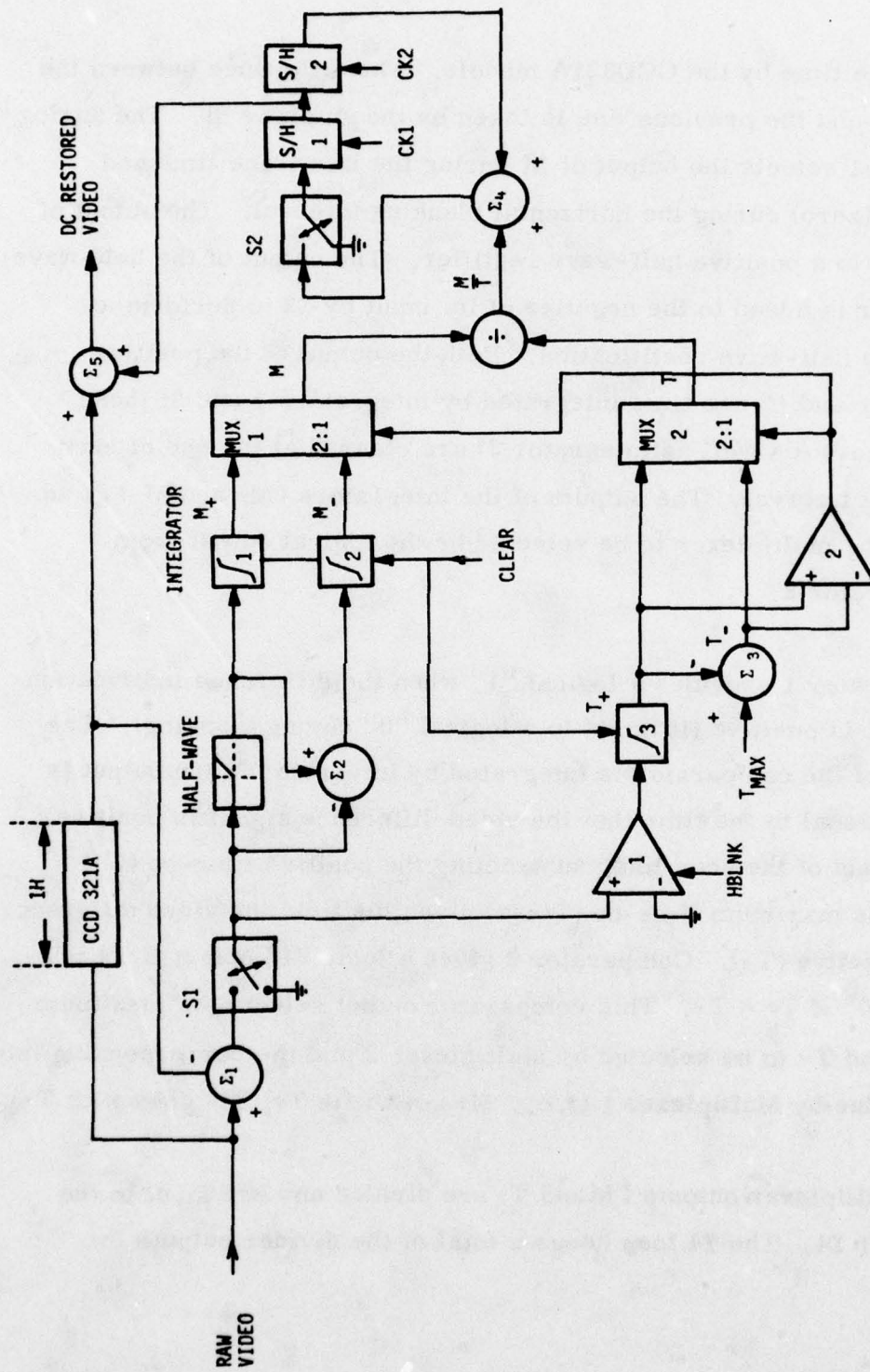


Figure 10. All-Analog Scheme Implementation Block Diagram

scan line time by the CCD321A module. The difference between the current and the previous line is taken by the summer $\Sigma 1$. The analog switch S1 selects the output of $\Sigma 1$ during the line trace time and ground (zero) during the horizontal blanking interval. The output of S1 goes to a positive half-wave rectifier. The output of the half-wave rectifier is added to the negative of its input by $\Sigma 2$ to perform a negative half-wave rectification. Both the output of the positive rectifier and $\Sigma 2$ are time integrated by Integrators 1 and 2; these integrators (as well as Integrator 3) are cleared at the end of each blanking interval. The outputs of the integrators (M+ and M-) go to an analog multiplexer to be selected by the logical output from Comparator 2.

Comparator 1 provides a logical "1" when the difference information from S1 is positive (it is set to a logical "0" during blanking). The output of the comparator is integrated by Integrator 3; the output is proportional to the time that the video difference signal is positive. At the end of the scan line, subtracting the positive time-on (T+) from the maximum time-on (Tmax) gives the time the video difference was negative (T-). Comparator 2 gives a logic "1" output if $T+ > T-$, and a "0" if $T+ < T-$. This comparator output selects the maximum of T+ and T- to be selected by Multiplexer 2 and the corresponding integral value by Multiplexer 1 (i. e., M+ goes with T+, M- goes with T-).

The multiplexer outputs (M and T) are divided and are input to the summer $\Sigma 4$. The $\Sigma 4$ loop keeps a total of the divider outputs for

computing the DC shift estimate. Two sample and hold units (S/H 1 and S/H 2) are used in a master/slave configuration as an analog adder with a line delay. The clock on S/H 1 (CK1) samples during the horizontal retrace time. The clock on S/H 2 (CK2) follows CK1 and is non-overlapping. The output of S/H 1 serves as the DC shift estimate and is added to the delayed video by summer Σ . The S/H units are cleared every frame using S2 to prevent an unstable loop.

All components in the circuit are analog, except for one-shots for clocking the sample/hold units. The summers are made with LM318 type operational amplifiers (op-amps) with 20 MHz small signal bandwidths. The Integrators 1 and 2 are made with low offset LM356 BI-FET amplifiers with 20 V/ μ sec slew rates and 4 MHz bandwidths. Integrator 3 is made with an LM318 op-amp with maximized slew rate (150 V/ μ sec) and minimized settling time ($< 1 \mu$ sec). All three integrators use 4066 CMOS analog switches for clearing the integrator during retrace. The Comparators 1 and 2 are an LM319 dual comparator and have 80 nsec rise times. The output of Comparator 1 is buffered through a gate for strobing during blanking and producing approximately a zero or five volt output.

The analog switches and multiplexers are all low-transient 4066 type CMOS switches. The divider is an Analog Devices 531K transconductance analog multiplier/divider with 2 μ sec settling time at one percent accuracy. This divider can be substituted for since the divider input is always bounded from $1/2 T_{max}$ to T_{max} , and

this small range will not significantly affect most divider settling times. The sample/hold units are Datel SHM-LM-2 types with 4 μ sec acquisition time.

The approximate power required (not including the 312A) is 2 W. The circuit should fit on one 6 in. x 4 in. card. The cost of the parts for the circuit (without the CCD) is \$200. The CCD unit and support circuitry will dominate the cost.

LOCAL AREA GAIN/BRIGHTNESS CONTROL (LAGBC)

Referring to the system diagram for the gain/brightness control in Figure 1, the local area gain/brightness control appears prior to the display of the video, after the synthetic DC restore has been accomplished. It performs the following functions:

- Varies the local average brightness of the displayed image (bias), so that overall dynamic range of scene is compressed.
- Enhances local variations above the contrast sensitivity threshold of the human eye.
- Automatically fits the intensity extremes in the enhanced video scene to the display limits.

A functional description of the algorithm is shown in Figure 11. The image intensity at each point is transformed based on local area statistics-- the local mean M_{ij} and the local standard deviation σ_{ij} computed on a local area surrounding the point. The transformed intensity is then:

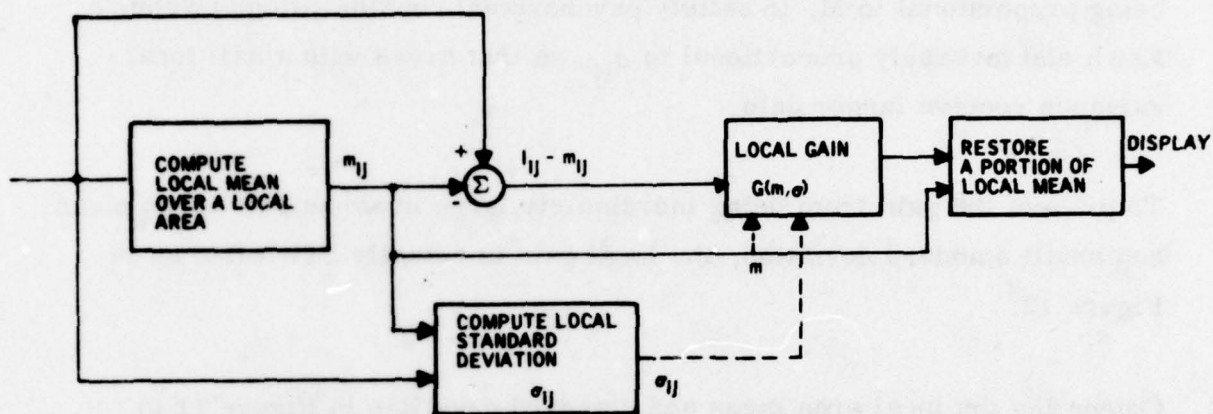


Figure 11. Functional Flow Description of Local Area Gain/Brightness Control (LAGBC) Algorithm

$$\hat{I}_{ij} = G_{ij} [I_{ij} - M_{ij}] + M_{ij}$$

where, the local gain

$$G_{ij} = \alpha \frac{M}{\sigma_{ij}}, \quad 0 < \alpha < 1$$

where M is the global mean.

In words, the local area mean is first subtracted from the image at every point. A variable gain is applied to the difference to amplify the local variation. A portion of the local mean M_{ij} is then added back to restore

the subjective quality of image. The local gain G_{ij} is itself locally adaptive, being proportional to M , to satisfy psychovisual considerations (Weber's Law); and inversely proportional to σ_{ij} , so that areas with small local variance receive larger gain.

To prevent the gain from being inordinately large in areas with large mean and small standard deviation, the local gain is actually controlled as in Figure 12.

Computing the local area mean and standard deviation in Figure 11 is similar to spatial filtering because the local mean is really the convolution

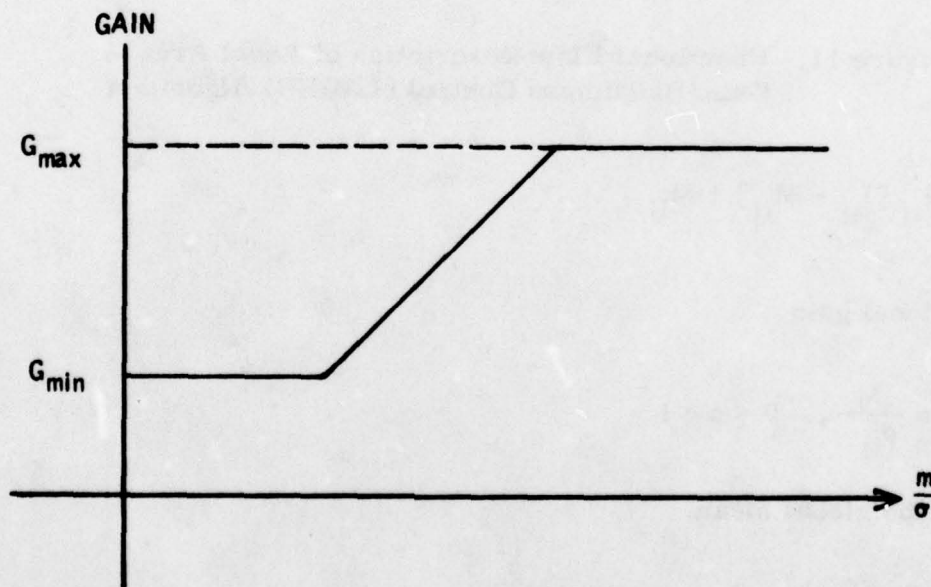


Figure 12. Local Area Gain vs. $\frac{m}{\sigma}$ Curve Used

of a rect function with the image. Convolution with the rect function is really low pass filtering because the sinc function is a low pass filter. Hence we can replace the local average function with an equivalent recursive low pass filter.

The local standard deviation σ_{ij} is approximated by a similarly low passed version of the absolute difference between the image intensity I_{ij} and the local mean estimate M_{ij} . Figure 13 is a realization of the LAGBC using linear recursive low pass filters to estimate the local mean and standard deviation.

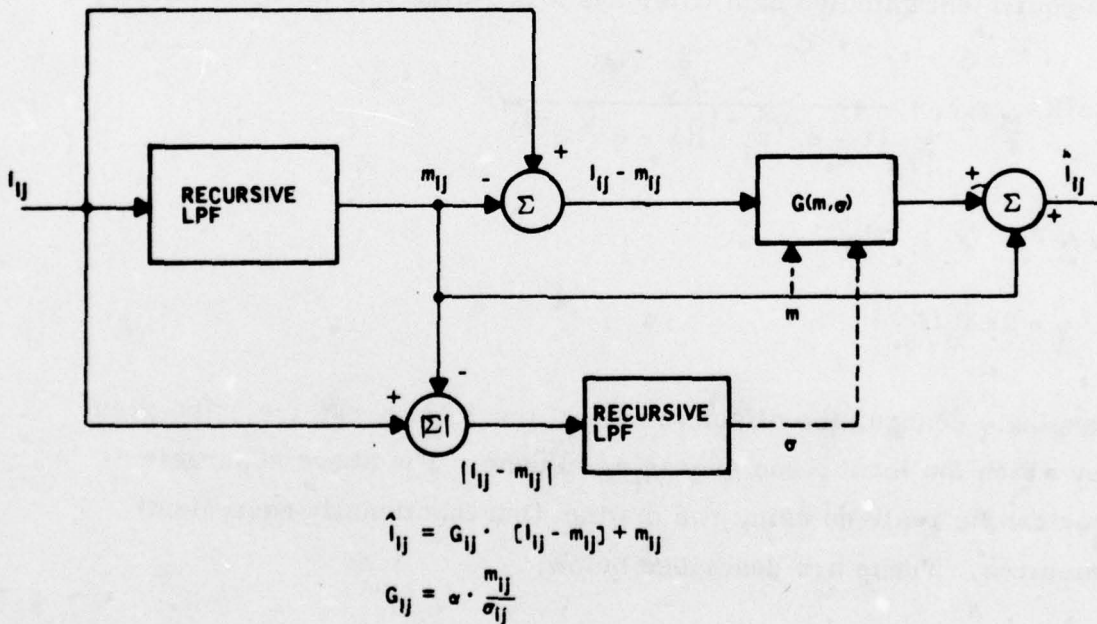


Figure 13. LAGBC with Linear Recursive Low Pass Filters

Realization of the Recursive Low Pass Filter--As demonstrated above, the two-dimensional recursive low pass filter is the basic building block of the LAGBC scheme.

The two-dimensional separable first-order recursive filter has a frequency response given by the product of the two one-dimensional filter responses:

$$|H(f_x, f_y)| = [1 + (f_x/f_c)^2]^{-\frac{1}{2}} [1 + (f_y/f_c)^2]^{-\frac{1}{2}}$$

f_c is the 3 dB cutoff frequency of the low pass filter.

The equivalent sampled data filter has a two-dimensional Z transform,

$$H(z_1, z_2) = \frac{\gamma^2}{(1 - e^{-\gamma} z_1^{-1})(1 - e^{-\gamma} z_2^{-1})}$$

where

$$\gamma = 2\pi f_c / f_s$$

Changing γ changes the effective size of the local area, i. e., the area over which the local mean averaging is done. The above separable filter can be realized using two distinct (but functionally equivalent) structures. These are described below.

1. Nonseparable implementation.

In the Z domain, let the output be $Y(z_1, z_2)$ and the input be $X(z_1, z_2)$.

Then,

$$\frac{Y(z_1, z_2)}{X(z_1, z_2)} = \frac{\gamma^2}{(1 - e^{-\gamma} z_1^{-1})(1 - e^{-\gamma} z_2^{-1})}$$

This form, implemented directly, gives the recursive relation

$$y(m, n) = \gamma^2 x(m, n) + e^{-\gamma} y(m-1, n) + e^{-\gamma} y(m, n-1) - e^{2\gamma} y(m-1, n-1)$$

where m is the row number, and n is the column number in the image.

The schematic for implementing this realization on real time video stream is shown in Figure 14. We see that the line delay and the two pixel delays give us the necessary delays to perform the filtering.

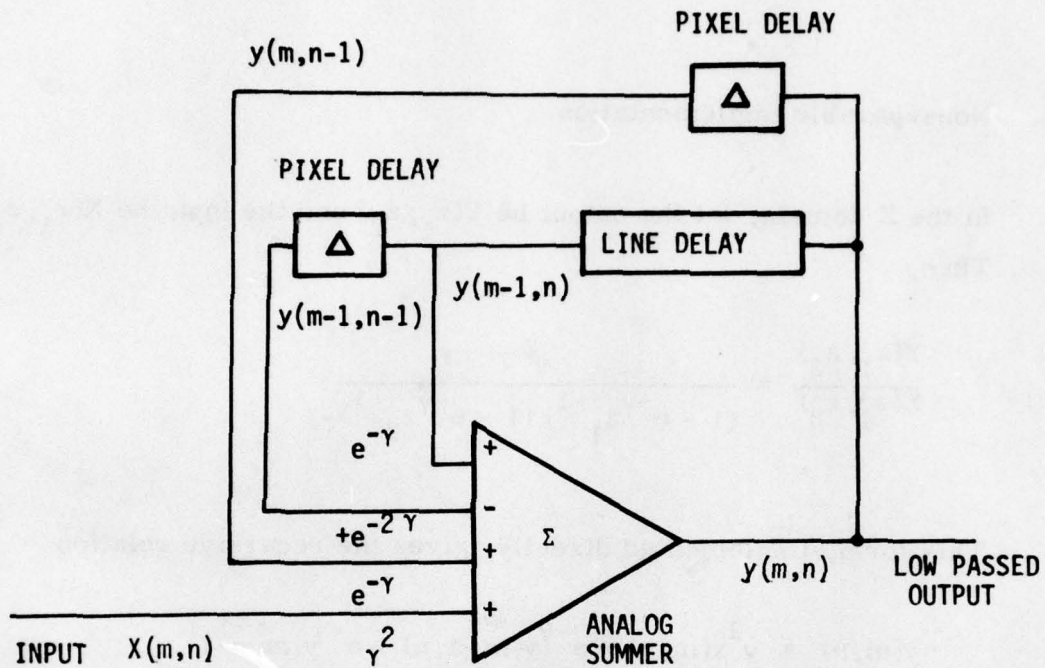


Figure 14. Implementation of Two-dimensional Recursive Low Pass Filter (Approach 1)

2. Separable implementation.

Define a new intermediate variable

$$W(z_1, z_2) = \frac{\gamma}{(1 - e^{-\gamma} z_1^{-1})} \cdot X(z_1, z_2)$$

Then,

$$\frac{Y(z_1, z_2)}{X(z_1, z_2)} = W(z_1, z_2) \cdot \frac{1}{(1 - e^{-\gamma} z_2^{-1})}$$

Therefore,

$$y(m,n) = \gamma w(m,n) + e^{-\gamma}y(m-1,n)$$

and

$$w(m,n) = \gamma x(m,n) + e^{-\gamma}w(m,n-1)$$

Thus we break up the two-dimensional filter into two one-dimensional filters in cascade. Figure 15a shows this filter realized in the above manner. The output of this filter is exactly equivalent to that of Figure 14 because the transfer function is separable. This realization has certain advantages over the nonseparable implementation, described below.

The parameter γ is in the range 0.1 to 0.3, which makes $e^{-\gamma} \approx 1 - \gamma$, in the range 0.9 to 0.7. In the nonseparable realization, the smallest weight, $\gamma^2 = 0.01$ to 0.1 is much smaller than the largest weight $1 - \gamma = 0.7$ to 0.9. The coefficient precision needed in the summing amplifier in Figure 14 is therefore very stringent. On the other hand, coefficient range in the separable formulation is much smaller: $\gamma = 0.1$ to 0.3 for the smallest weight and $1 - \gamma = 0.9$ to 0.7 for the largest. Therefore, the separable structure is much less susceptible to charge coupled device (CCD) noise and amplifier gain variation than is the corresponding nonseparable structure.

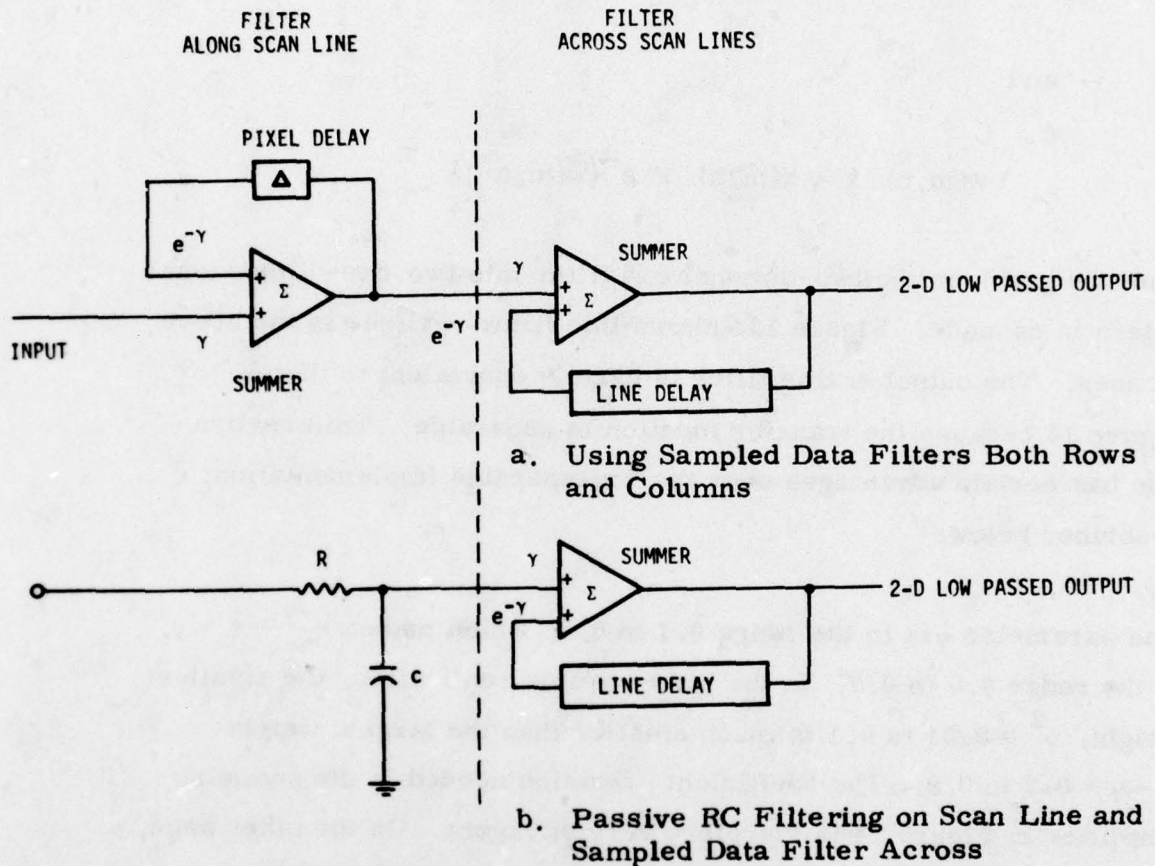


Figure 15. Separable Implementation of Two-dimensional Low Pass Filter (Approach 2)

The separable structure has the additional advantage that the first low pass filter (along the scan direction) can be easily implemented in a passive RC first-order circuit, as shown in Figure 15b. At video frequencies, the required resistor (R) and capacitor (C) values are very reasonable. This eliminates the need for the single pixel delay for the first filter. However, the second low pass filter in the vertical direction is still a sampled data filter as before (Figure 15b).

The above LAGBC scheme, using the separable recursive low pass structure and CCD line delays, has already been breadboarded and is undergoing testing. Following is a description of this hardware and a discussion of the various building blocks of the schematic.

LAGBC Breadboard Description

The LAGBC breadboard implementation block diagram is shown in Figure 16. The different sections are outlined in dashed lines and numbered.

In Figure 16, the LAGBC input intensity information I_{ij} is already sync separated and scaled from "0" (black) to "1" (white) volts* by the global gain/bias unit. This intensity information is processed by the local mean two-dimensional separable low pass filter in Block 1. As shown in the detailed diagram of this filter in Figure 17, filtering is first done horizontally by a first-order analog RC low pass filter with

* See the global gain/bias subsection of this report. Scaling is needed due to dynamic range of the LAGBC circuitry.

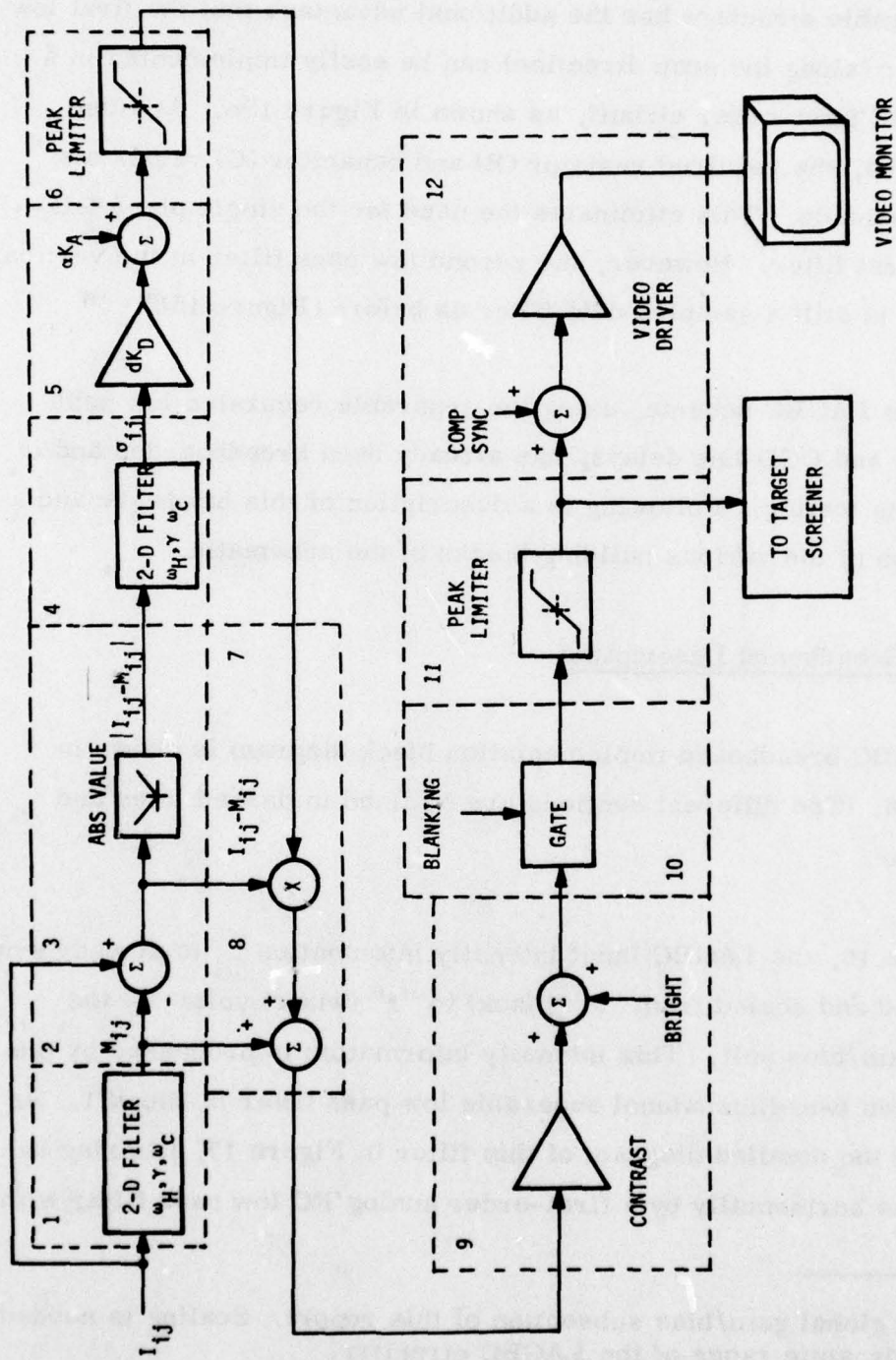


Figure 16. LAGBC Implementation Block Diagram

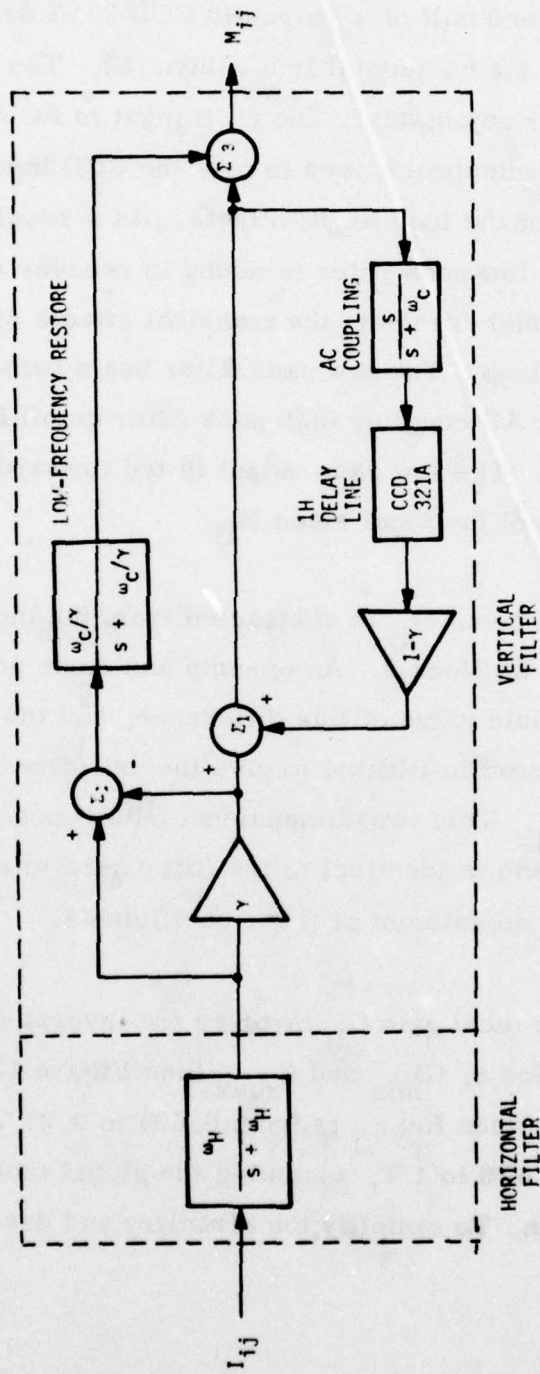


Figure 17. Separable Two-dimensional Recursive Filter Implementation Diagram

cutoff frequency ω_H , and then vertically by a CCD recursive low pass filter with coefficient γ . The CCD filter is implemented with high bandwidth operational amplifiers and half of a Fairchild CCD 321A dual 455-sample CCD, which is used as the horizontal line delay (1H). The values of ω_H and γ are potentiometer adjustable. The CCD input is AC coupled in the feedback loop to eliminate the need to bias the CCD input and to reduce the sensitivity of the loop to DC offsets. As a result of this AC coupling, an additional low pass filter is added to recover lost low frequency information and eliminate the transient effects of having a high pass filter in the loop. The low pass filter has a cutoff frequency of $\frac{\omega_c}{\gamma}$, where ω_c is the AC coupling high pass filter cutoff frequency (approximately -1 Hz). The low pass output is fed forward to summer Σ_3 to give the estimate of the local mean M_{ij} .

In Figure 16, the local mean M_{ij} is subtracted from the input intensity I_{ij} by an op-amp summer in Block 2. An op-amp and diode network in Block 3 takes the absolute value of this difference, and the result is two-dimensionally filtered in Block 4 to give the estimate of the local standard deviation, σ_{ij} . This two-dimensional filter uses the other half of the CCD 321M unit and is identical to the filter used to compute M_{ij} , except for independent adjustment of filter coefficients.

The computation of the local gain G_{ij} involves the inversion of σ_{ij} . If the simulation values for α , G_{\min} and G_{\max} (see Figure 12) are used, the range of relevant values for σ_{ij} is from 0.025 to 0.25 V for an input (intensity) range of 0 to 1 V, assuming the global mean replaces M_{ij} in the gain equation. To simplify the circuitry and desensitize the

gain with respect to the small values of σ_{ij} , a linear approximation to the inversion, shown in Block 5, was used. The slope (αK_0) and bias (αK_A) are adjustable (α is the weighting term in the local gain equation). A single op-amp implements this inversion circuit. The output of the inversion circuit is peak limited between G_{\min} and G_{\max} by an adjustable threshold diode network in Block 6.

The local gain G_{ij} is multiplied by $(I_{ij} - M_{ij})$ with a high bandwidth analog multiplier in Block 7 and the result added to M_{ij} in Block 8 by an op-amp summer to produce an enhanced output. The signal is normalized by adjusting the contrast (AC gain) and brightness (DC level) in Block 9. A black level is set in Block 10 by using an analog switch during blanking. The signal is then peak limited to eliminate "spikes" and overshoot by a diode threshold pair in Block 11. The output of this limiter is used by the target screener. To display the signal on a video monitor, the composite sync signal is added in with an op-amp summer and then buffered with a video driver in Block 12. The resultant output is a composite video signal capable of driving a 75Ω load.

LAGBC Breadboard Evaluation

Preliminary evaluation of the LAGBC breadboard has been done. The values of filter coefficients, gain maximums and minimums, etc., from the simulations were used in the hardware. The images tested were from data furnished by NVL using a Hughes FLIR. The data from a video disk recorder were processed at the 525-line standard television rate, displayed on a video monitor, and photographed.

The input to the LAGBC was first automatically scaled between 0 and 1 V by an AGC unit using the frame video mean and standard deviation; this adjusted the contrast and brightness on a sync separated video input from a video disk, tape, camera, etc. The AGC does not correct saturated inputs as will the global gain/bias control for the FLIR, but it does adjust small signals to the range needed by the LAGBC.

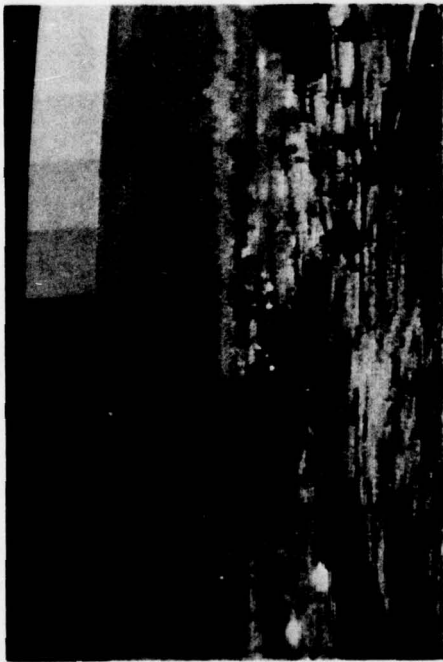
For evaluation, the LAGBC settings were preset and not adjusted for any of the test inputs. Original and enhanced test photos of the video monitor output are shown in Figures 18 to 21. It can be seen in these photos that the contrast between large areas is decreased while small local area contrast increases. This is seen by comparing Figure 18a and 18b, and looking at the black (smoky) area on the left side. In the original image part of a target can be seen, but in the enhanced image there are two obviously better defined target areas. Similar effects are seen in the other photos.

These initial tests on the breadboard show that the LAGBC concept works in relatively simple analog hardware. Tests still must be run to optimize filter coefficients and other adjustable parameters for use with the MODFLIR.

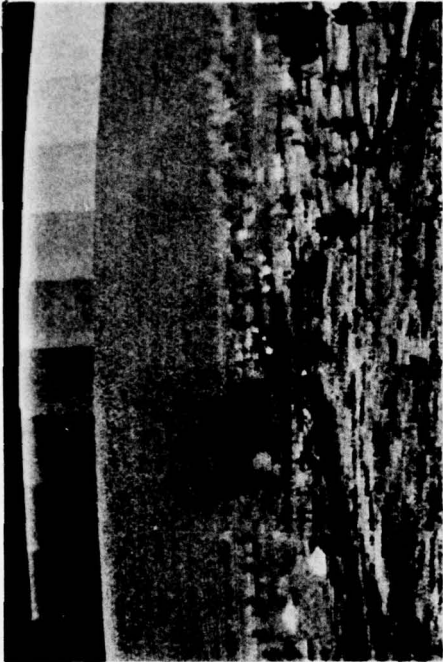
GLOBAL GAIN/BIAS CONTROL

PATS Global Gain/Brightness Control Approach

Thermal imagers such as the MODFLIR have two controls that need constant readjustment for optimum viewing: 1) the gain control that

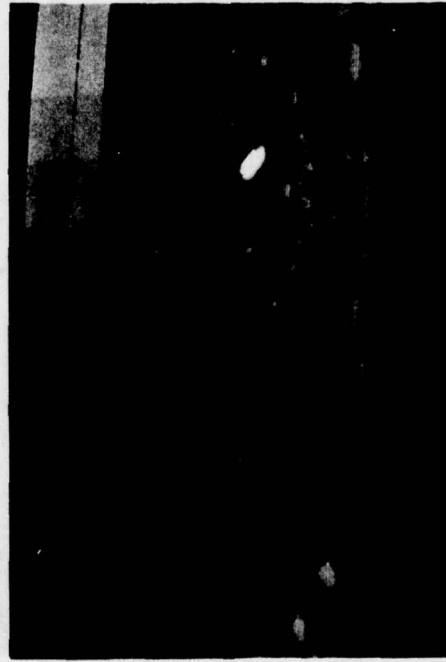


a. Original FLIR frame from video tape



b. Enhanced by real time hardware

Figure 18. Results of LAGBC Breadboard



a. Original FLIR frame from video tape



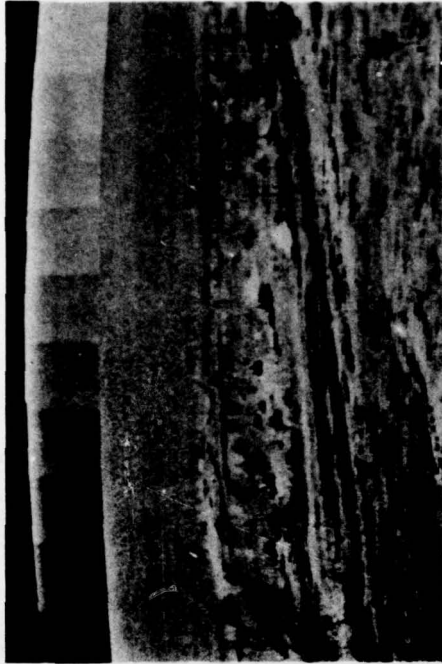
b. Enhanced by real time hardware

Figure 19. Results of LAGBC Breadboard



a. Original FLIR frame from video tape

Figure 20. Results of LAGBC Breadboard



b. Enhanced by real time hardware



a. Original FLIR frame from video tape



b. Enhanced by real time hardware

Figure 21. Results of LAGBC Breadboard

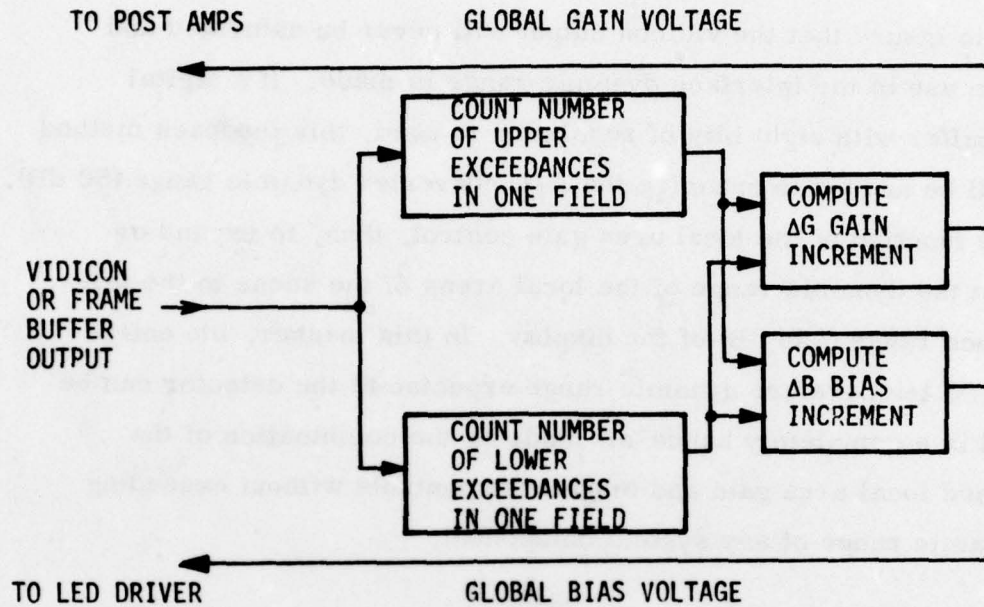


Figure 22. Functional Schematic of Global Gain and Brightness Control

controls the gain of the post-amps and hence the sensitivity to temperature differences in the scene; and 2) the brightness control that adds a bias to the post-amp output (LED driver) stage to shift the temperature range of the scene up and down to match the luminance range of a display or the storage range of a digital frame buffer.

Figure 1 shows the functional diagram for the global gain/brightness control, DC restore and local area gain/brightness control for the ATS. The LED/vidicon interface has a dynamic range of approximately 40 dB, and therefore a global control of the gain and bias is incorporated by feeding the appropriate voltages back to the post-amps and LED driver

cards, to insure that the vidicon output will never be saturated and optimum use of the interface dynamic range is made. If a digital frame buffer with eight bits of resolution is used, this feedback method will still be needed to optimize the A/D converter dynamic range (50 dB). It is the function of the local area gain control, then, to expand or contract the dynamic range of the local areas of the scene to the full luminance range (~20 dB) of the display. In this manner, the entire 10,000 : 1 temperature dynamic range expected at the detector can be handled in a completely hands-off mode by the combination of the global and local area gain and brightness controls without exceeding the dynamic range of any system component.

Figure 22 shows a functional block diagram of the global control algorithm to make optimum use of the LED vidicon or frame buffer interface dynamic range. The number of exceedances of two thresholds corresponding to the upper and lower limits* on the vidicon or A/D output are counted and integrated over a field time (1/60 sec). These counts are used to generate the gain and bias increments ΔG and ΔB which are used to change the gain and bias voltages fed back to the FLIR boards. For example, if the upper exceedances are significantly more than the lower exceedance counts, the bias increment is made negative, and vice versa. The gain increment is determined in a similar fashion and would be a function, for example, of the sum of the lower and upper exceedance counts. No change will be made to the gain and bias if the lower and upper exceedance counts are within

* Lower (black) limit 0, and upper (white) limit 1 V will be typical.

a prespecified range--typically 0.1 percent or less of the entire frame. In this manner isolated noise peaks are allowed but any significant saturation of the vidicon, and subsequent display, will be avoided. Since the scene extrema do not change very rapidly, the feedback process will converge quickly to a stable value.

Global Gain/Bias Implementation

A simple implementation diagram of the scheme featuring a microprocessor system is shown in Figure 23. The input signal from the vidicon or frame buffer is thresholded by an upper and lower threshold comparator; the output of each is a logical level. The comparator outputs are gated by a clock that runs at the pixel sampling rate during the active line time and is off during blanking. Two binary counters are clocked by the number of pixels that the threshold is exceeded, giving a count of the number of exceedances per frame. The most significant bits of the counters are examined by the microprocessor system to make the computation for the gain (G) and bias (B) to the FLIR. The values of G and B are converted to analog format and buffered to drive the FLIR control inputs.

The microprocessor specified for this application is the single-chip Intel 8748 with 1 K of internal program memory. Most instructions are executed in 2.5 to 5.0 μ sec, meaning that 200 to 400 instructions can be executed in the 1 msec vertical retrace time. If necessary, computation

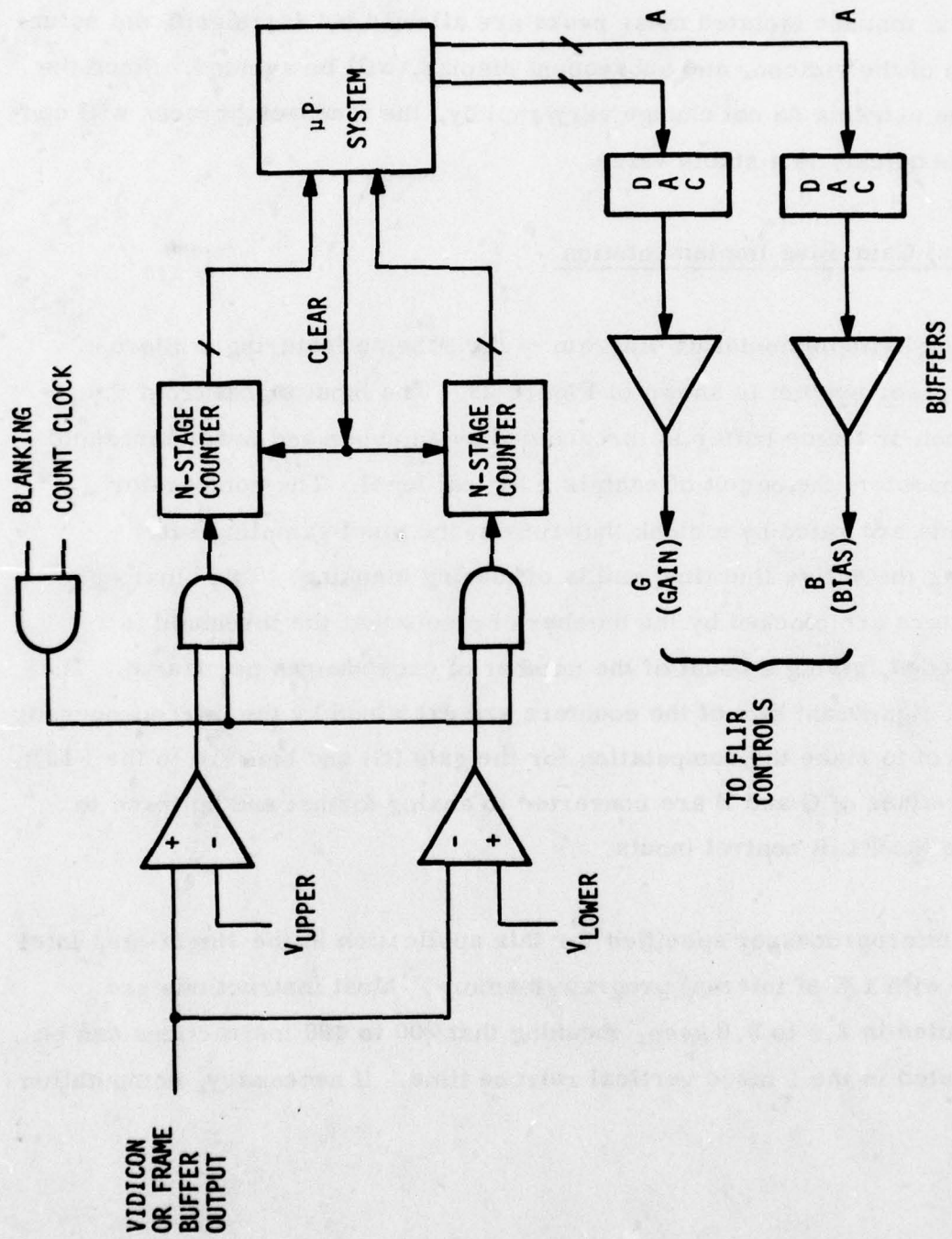


Figure 23. Global Gain/Bias Control Unit

can overlap into the actual trace time. A complex and flexible program can be placed in the 1 K of the Intel 8748 program storage, allowing for numerous algorithm implementations to be tried once the FLIR has arrived.

For initial development, the Intel 8080 MDS simulator will be used for algorithm testing. The 8748 and 8080 have very similar instructions sets and timing. Once operational on the MDS system, the algorithm will be programmed into the 8748 for installation in the final hardware.

SECTION III

TARGET SCREENING

This section covers the progress made in this reporting period on the following target screening tasks:

- Data preparation
- Image segmentation
- Feature extraction

SECTION SUMMARY

A total of 260 FLIR image frames containing tanks, trucks, and armored personnel carriers (APCs) have been digitized to date and annotated in the data preparation task. They have been debanded where necessary. Target positions and types have been identified and recorded on the digital tape header.

The image segmentation part of the simulation software is now fully operational, with a number of improvements to the present Augmented Target Screener system (ATSS) autothreshold, background estimate, and the object interval generation criterion to make the segmentation more robust. Feature extraction software has been developed to extract moment features on object segments as well as Fourier descriptors (FDs) of the object boundary extracted by the segmenter. Analysis software has been developed to plot and analyze the

discriminatory powers of these features. The test images are being run through the simulation to evaluate the recognition features. Following is a more detailed report of the progress on these tasks.

DATA PREPARATION

The five classes of targets of interest to PATS are:

- Tank
- APC
- 2-1/2 ton truck
- Tracked missile launcher
- Tracked anti-aircraft cannon

Training data for statistical classifier design are available to a limited extent on the first three classes, tank, APC, and truck. Targets have strong elevation and aspect angle dependence, as we see in Figures 24a and b. Therefore we not only have five classes of targets to discriminate, but each target class has subclasses that depend on the target aspect and elevation. That is why, when extracting the recognition features and designing the subsequent classifier, we have to keep the target aspect, elevation, and type identities separate. Also, we need training data that statistically represent all the aspect and elevation angle combinations expected.

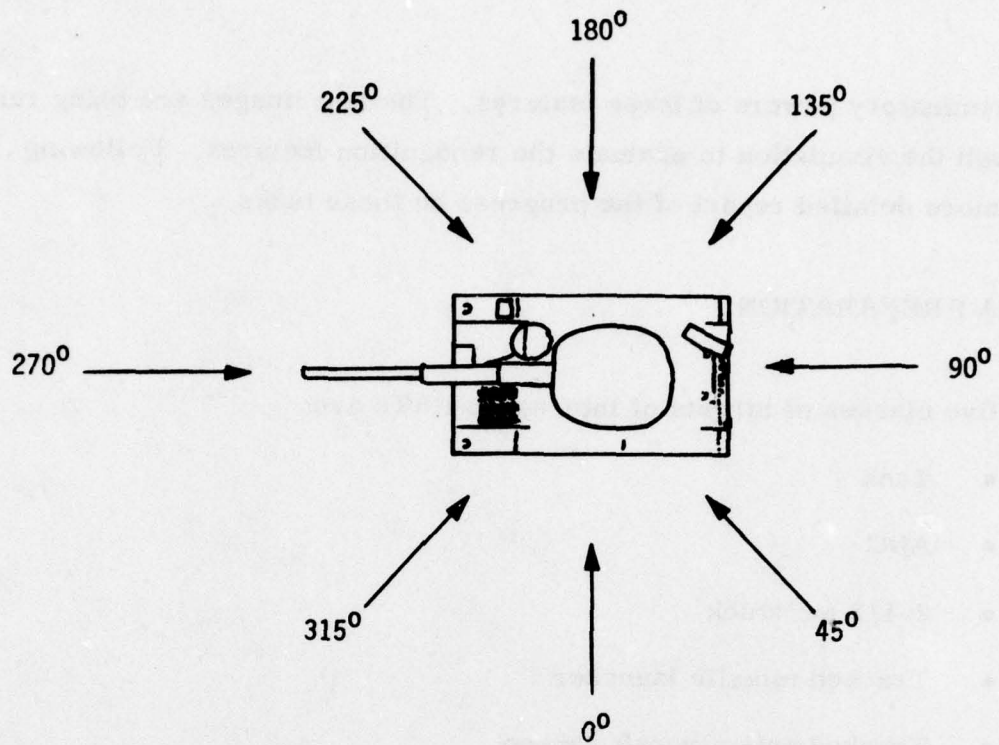


Figure 24a. Aspect Angle Dependence

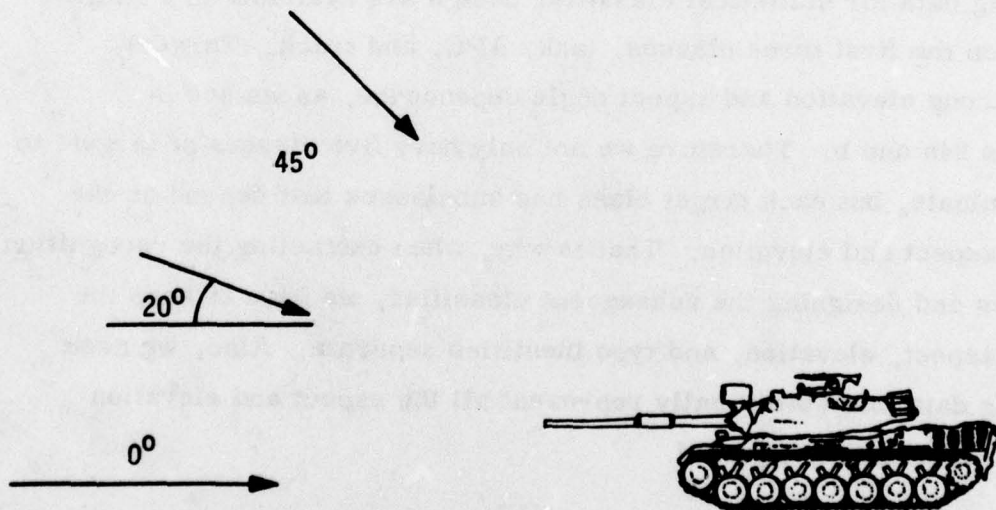


Figure 24b. Elevation Angle Dependence

It is easy to see that shape features extracted cannot be made aspect independent (unless we were looking straight down, in which case rotation invariance could imply aspect invariance). But range invariance, up to a point, is easier because it merely involves scaling. As for elevation, we are interested in low elevation angles for the "pop-up" mission scenarios expected in the Advanced Attack Helicopter (AAH) program.

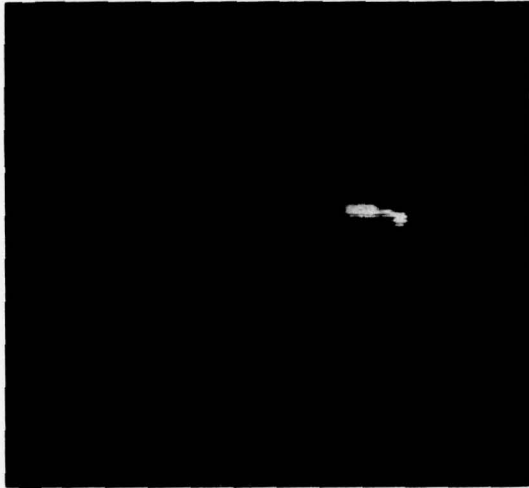
The sensor altitude in these missions is small (treetop, < 20 m), and meaningful ranges (> 200 m) imply very small elevation angles (< 10 degrees). Therefore, in the classifier design, we intend to fix the elevation angle at approximately 0 to 10 degrees, which simplifies the problem somewhat.

IMAGERY SOURCES

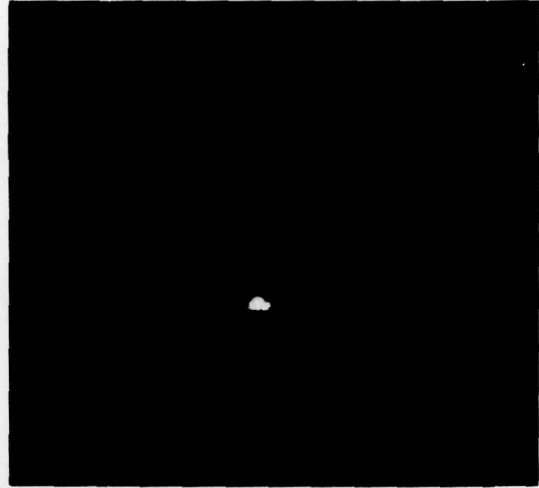
We now have two sources of video taped 525-line imagery: the Honeywell FLIR imagery taken by Marge Krebs of Honeywell, and the new serial scan FLIR data supplied by NVL. The Krebs data consist of imagery acquired from a light aircraft of tanks, APCs, and 2-1/2 trucks, at an altitude of approximately 3,000 feet at ranges of 10 miles to flyover. Because of this high altitude, all frames at useful slant ranges tend to have high elevation angles (typically > 25 degrees). The target aspects represented are primarily side and oblique side. Very few front and rear aspects are present in this test set.

The NVL data, on the other hand, were acquired from a low altitude platform FLIR and therefore the elevation angles are near zero. The target classes here are predominantly tanks with a fair representation of APCs. Multiple targets are present and therefore this represents a good test base. All aspects and ranges are adequately represented. However, there are no 2-1/2 ton trucks in this imagery.

We have digitized a total of 160 frames from the Krebs video tape of medium to far range trucks, tanks, and APCs. These data consist of elevation angles between 30 to 45 degrees and target aspects of mostly side and slant side. The FLIR with which this data was acquired had two channels with different coupling capacitor constants. As a consequence, one channel exhibits a more severe droop than the other along a scan line. This resulted in severe banding of the imagery near the right of the frame. Figures 25a and b show examples of this banding. This banding can play havoc in the segmentation stage by causing spurious vertical edges. Therefore, we devised a statistical debanding algorithm that corrects narrow vertical strips of the digitized image by estimating the average bias of one channel with respect to the other in that strip. This takes care of the variation of the bias level along the scan direction and successfully debands the digitized images. Additional logic is incorporated to account for the saturated regions in the image (which were not banded before the debanding process). Figures 26a and b are the debanded versions of the images in Figures 25a and b. We see that the debanding algorithms successfully restore the differences in the two channels without breaking up saturated targets. Figure 27 is a sample sheet containing

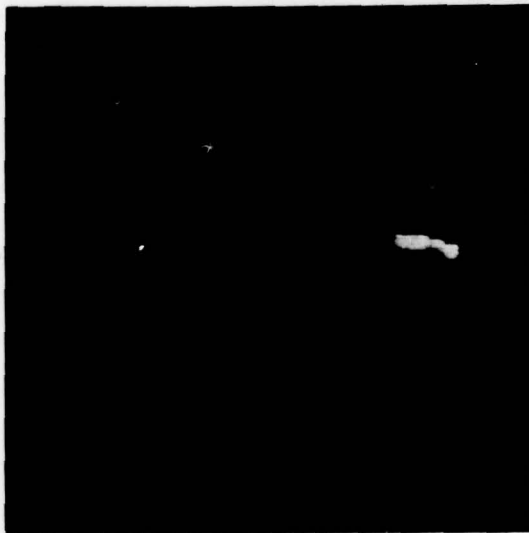


a

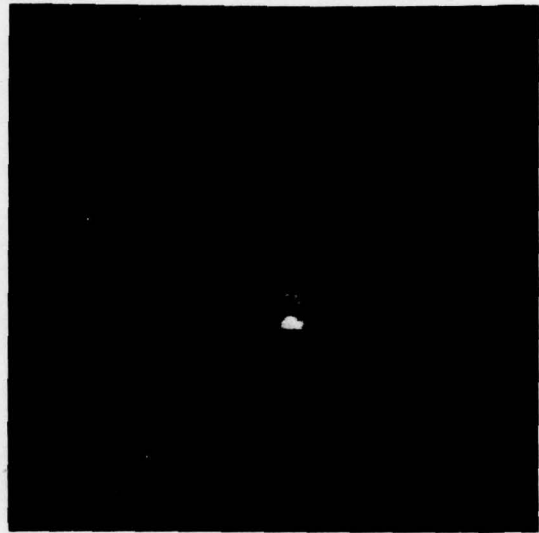


b

Figure 25. Examples of Banding Due to Differences in the Two Channel Bias Variations in the Krebs FLIR Data Base



a



b

Figure 26. Result of Applying the Debanding Algorithm to the Frames in Figure 25

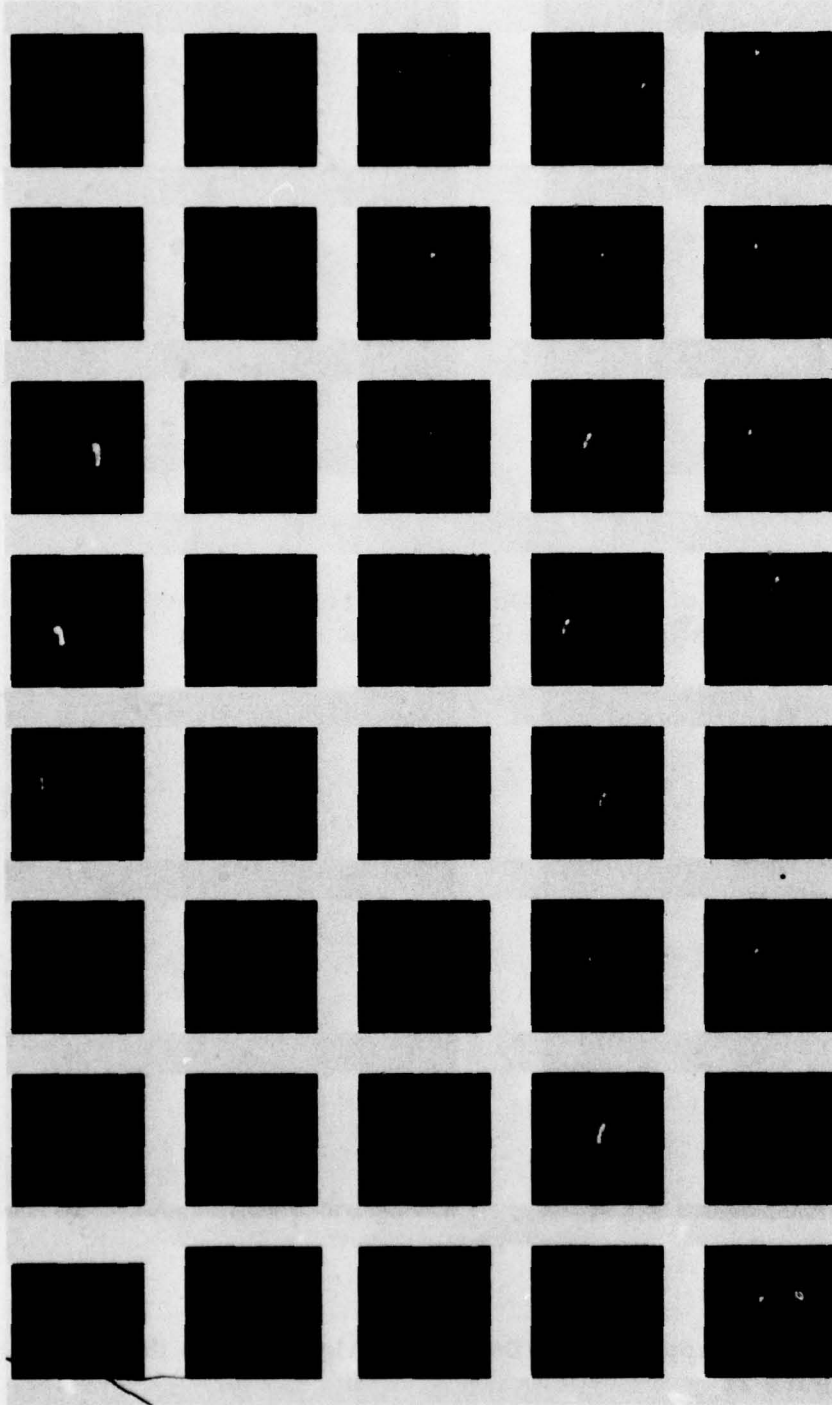


Figure 27. Examples of FLIR Image Frames Digitized from the Honeywell
FLIR Tape for the PATS Training Data Base

some of the 160 subframes that have been digitized and debanded for the PATS simulation effort.

The NVL imagery does not suffer from the banding. Approximately 100 frames of this imagery have been digitized. Figure 28 shows some examples of the digitized frames from this tape.

Tables 1 and 2 show the proportion of tanks, trucks, and APCs at various aspects among the two sets of data digitized so far. We note that there is a preponderance of tanks in both sets. More frames are being digitized from the new NVL video tape to complete representation of all aspect angles of tanks and APCs.

The frames we have digitized so far are "independent" in that they are often more than a second apart and taken during different passes. Therefore, we are also digitizing short sequences of frames (10 to 15 frames) to use in testing the interframe analysis portion of the simulation.

As each frame is digitized, annotative information is written into the magnetic tape header. This information includes for each target in the frame the target type, aspect, size (width and height in pixels), and its position in the image file. This header information and the hardcopy film transparencies (reproduced in Figures 27 and 28) are indispensable for feature analysis, evaluating the image segmentation, and supplying the "ground truth" in training the classifiers.

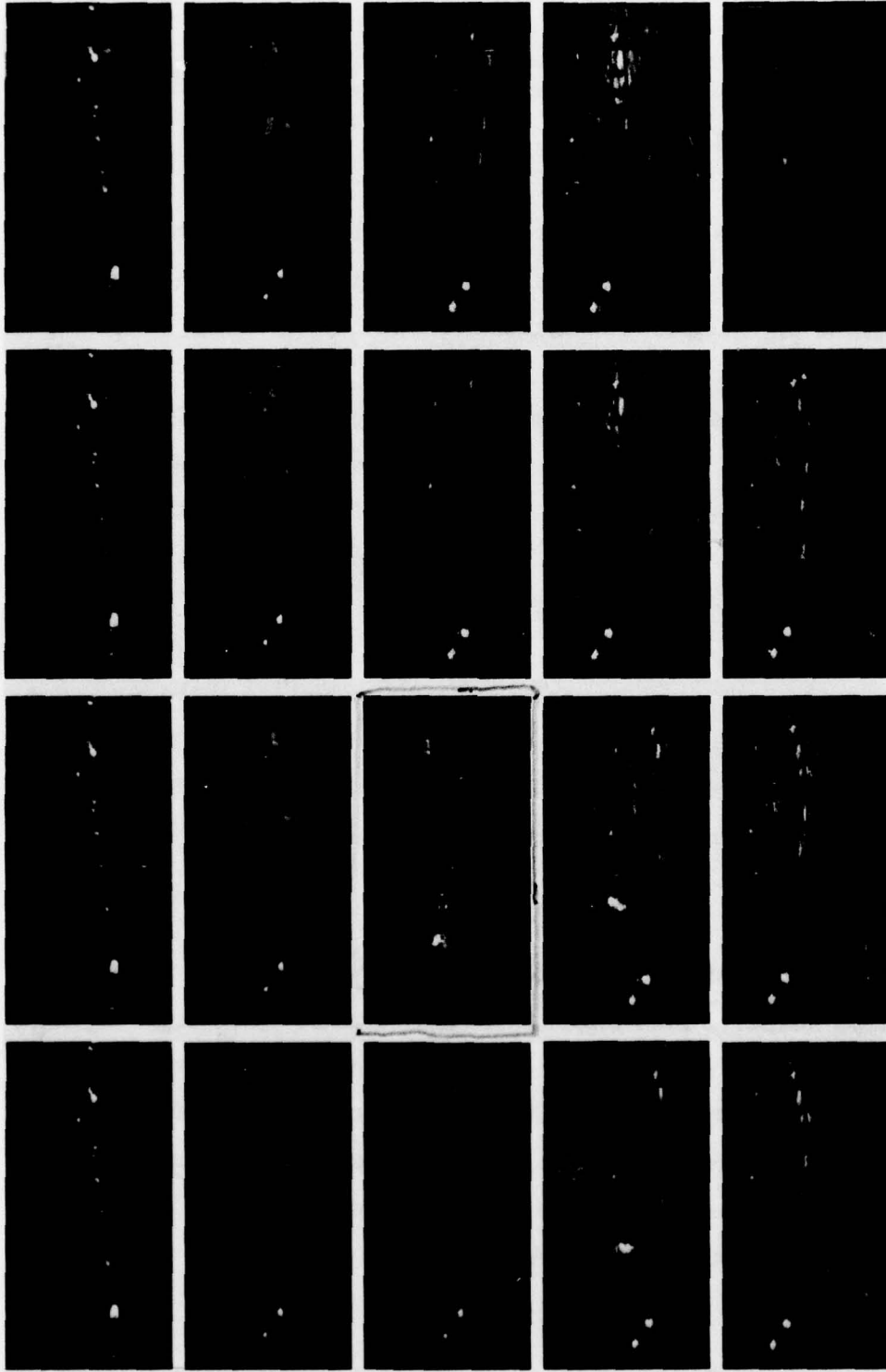


Figure 28. Examples of FLIR Frames Digitized from the NVL Video Tape for the PATS Training Data Base

TABLE 1. PROPORTION OF TARGET TYPES AND ASPECTS IN KREBS DATA (FROM THE HONEYWELL FLIR) DIGITIZED TO DATE (Average elevation angle = 30°)

Target Type	Aspect Angle								Total
	0	45	90	135	180	215	270	315	
Tank	12	0	5	2	32	0	0	0	51
APC	15	0	2	1	16	0	2	6	42
Truck	21	0	2	0	23	0	16	0	62
Jeep	1	0	5	2	4	0	0	4	16

TABLE 2. PROPORTION OF TARGET TYPES AND ASPECTS IN THE NVL FLIR IMAGERY DIGITIZED TO DATE (Average elevation ≈ 0 --Ground mounted FLIR)

Target Type	Aspect Angle								Total
	0	45	90	135	180	215	270	315	
Tank	20	3	13	3	13	12	17	2	83
APC	3	4	2	0	0	3	3	0	15
Truck	0	0	0	0	0	0	0	0	0
Jeep	2	0	1	0	2	0	0	0	5

The digitizer at the Honeywell facility digitizes along columns of the image. As a consequence, each image row on the magnetic tape corresponds to a column of the displayed image (perpendicular to the scan direction). Therefore, we transpose the digitized image files so that the rows in the image file correspond to the scan lines in the original image. This enables a more faithful simulation of the PATS hardware concepts.

IMAGE SEGMENTATION

Figure 29 is an overview of the target screening functions involved in PATS. The function of image segmentation is to extract simple regions from the image that are characteristic of targets. In this subsection we describe the results of our simulation of the segmentation process and then discuss the sensitivity of the segmentation to the parameter changes in the algorithm. The goal of the simulation is to render the first level segmentation (candidate target extraction) robust, so that reliable recognition features can be extracted from the segmentation.

Autothreshold

As we saw in Figure 29, we need to extract the "object intervals" along each scan line; these are then concatenated in the bins to complete the representation of each candidate object. This enables purely sequential real time operation on the serial video. These object intervals are generated by coincidence of "brights" and "edges" along a scan line. Edges are thresholded binary outputs from a two-dimensional edge

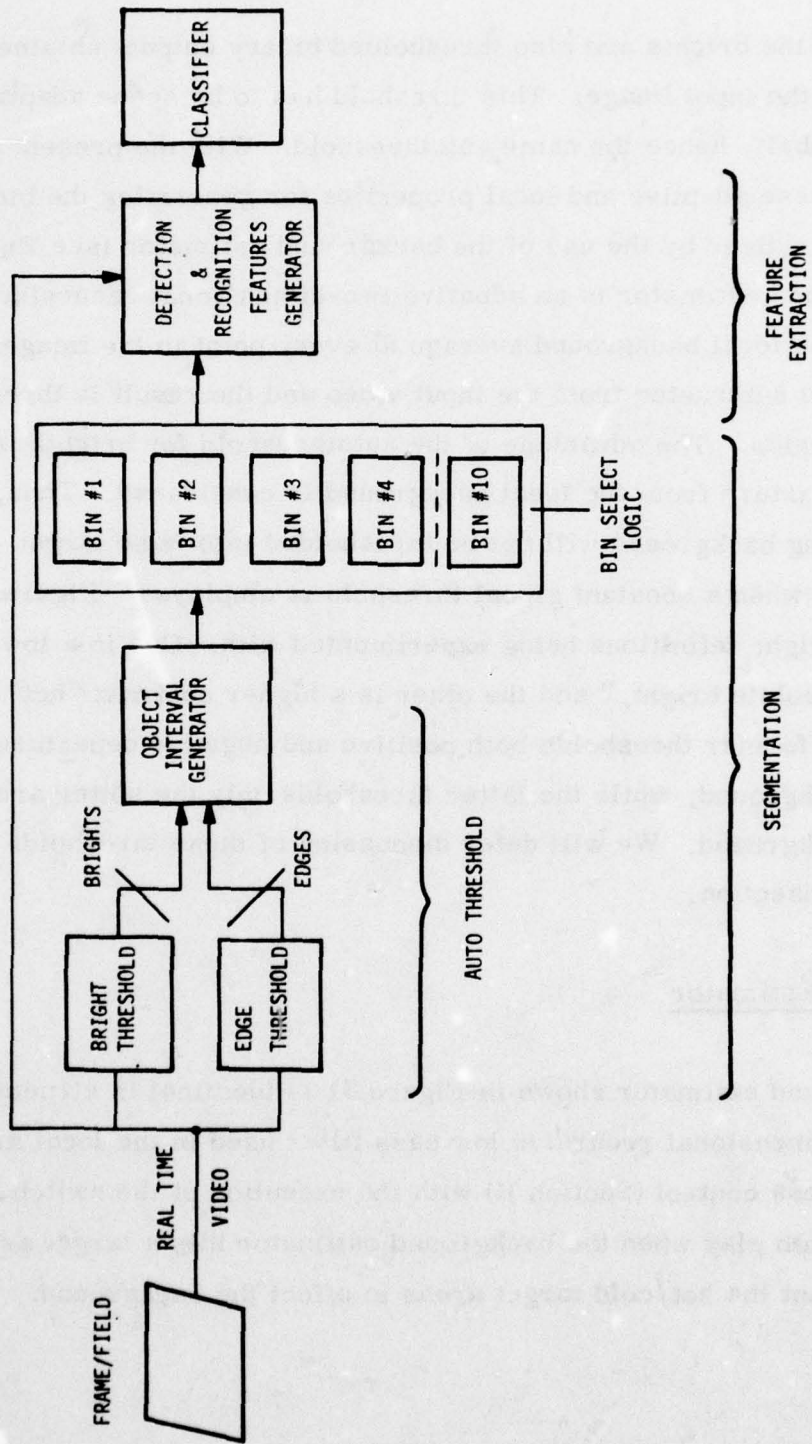


Figure 29. PATS Target Screening System Concept

operator and the brights are also thresholded binary outputs obtained by thresholding the input image. This threshold has to be scene adaptive and local (not global), hence the name autothreshold. With the present autothreshold, these adaptive and local properties for generating the binary brights are realized by the use of the background estimator (see Figure 30). The background estimator is an adaptive two-dimensional recursive filter that yields the local background average at every point in the image. This background is subtracted from the input video and the result is thresholded to get the brights. The advantage of the autothreshold for brights is that only departure from the local background is considered. Thus, slowly varying background will not be thresholded into false target contours, as when a constant global threshold is employed. Figure 30 shows two bright definitions being experimented with. One is a low contrast "absolute bright," and the other is a higher contrast "hot" bright. The former thresholds both positive and negative departures from the background, while the latter thresholds only the hotter areas from the background. We will defer discussion of these thresholds to a later subsection.

Background Estimator

The background estimator shown in Figure 31 is identical in structure to the two-dimensional recursive low pass filter used in the local area gain/brightness control (Section II) with the exception of the switch SW1 that comes into play when the background estimator hits a target area. We do not want the hot/cold target areas to affect the background

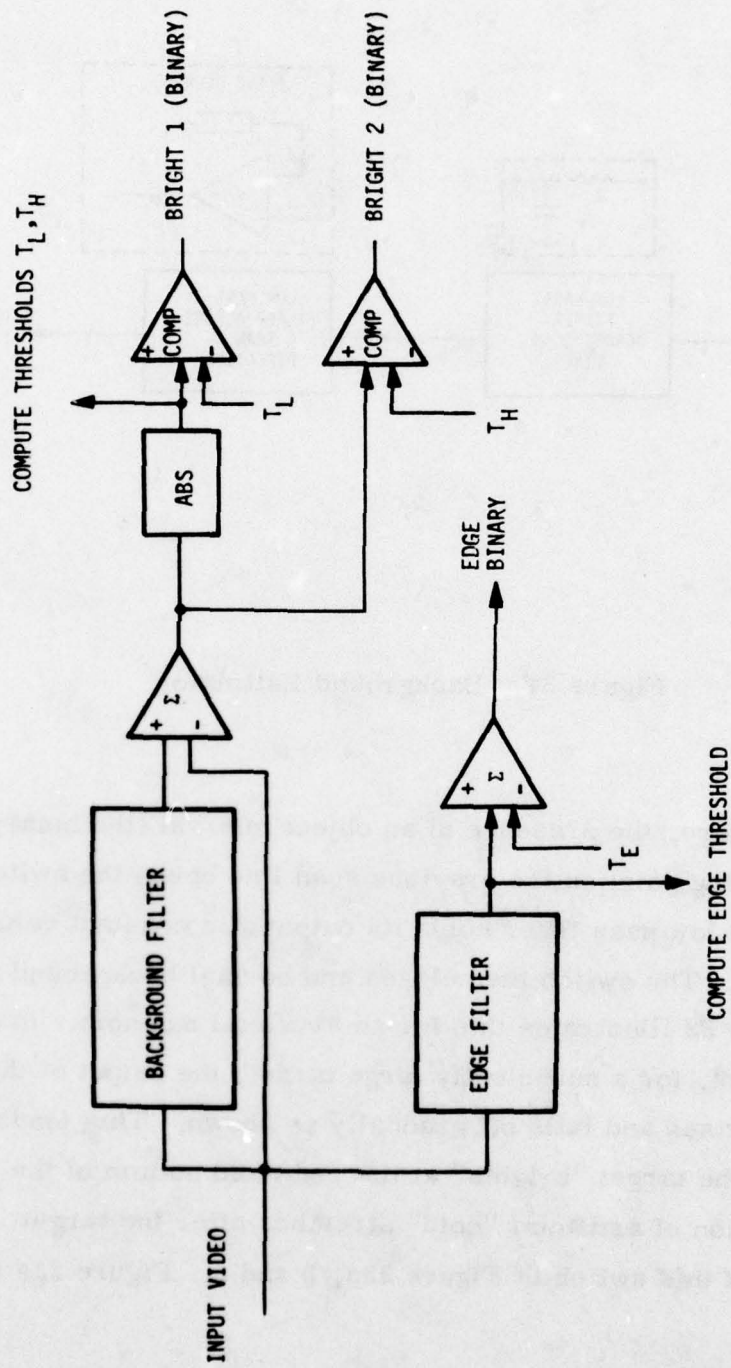


Figure 30. Autothreshold Structure

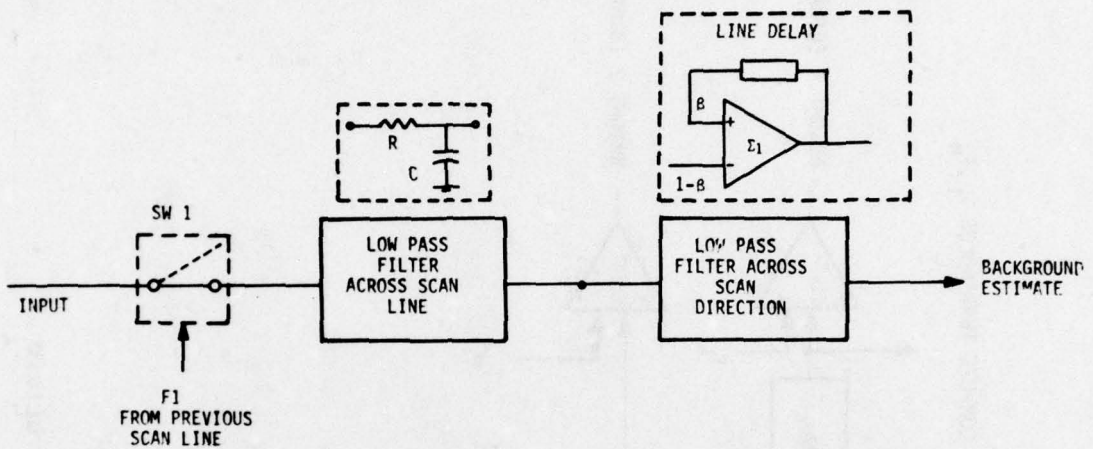


Figure 31. Background Estimator

estimate. Therefore, the presence of an object interval (the binary F1) at the corresponding point on the previous scan line opens the switch SW1, and the horizontal low pass filter holds its output at a constant value until the object ceases. The switch then closes and normal background updating resumes. Figure 32 illustrates this for an artificial example. In the absence of the switch, for a sufficiently large target, the output of the horizontal filter rises and falls off gradually as shown. This tends to result in loss of the target "brights" at the right and bottom of the target, and creation of artificial "cold" stretches after the target. We show the utility of this switch in Figure 33a, b and c. Figure 33a is the

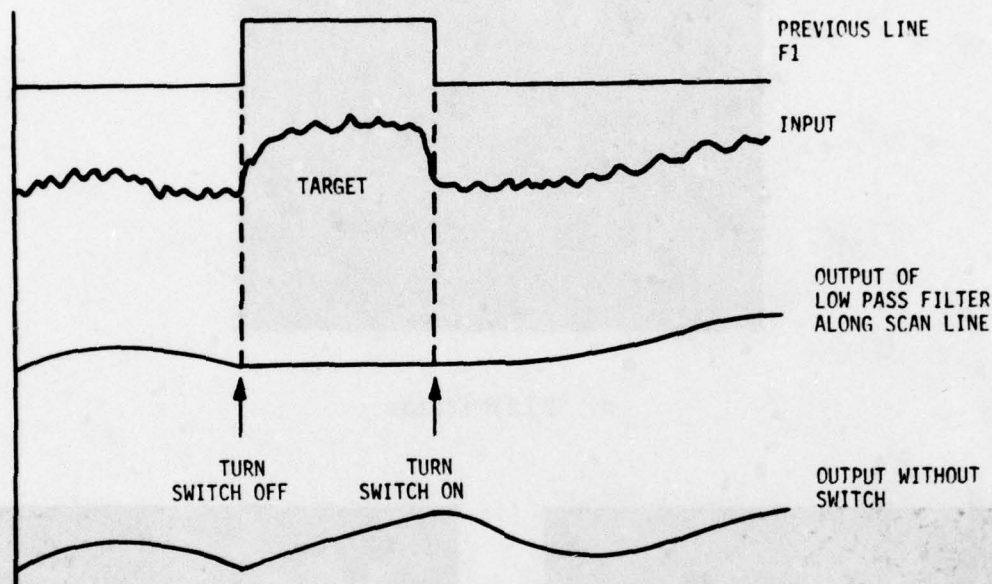
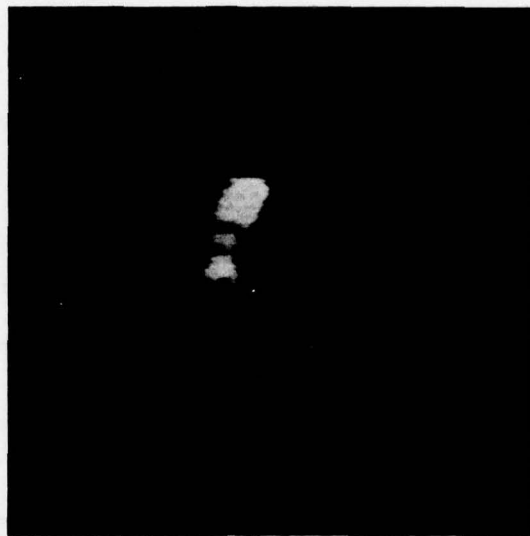
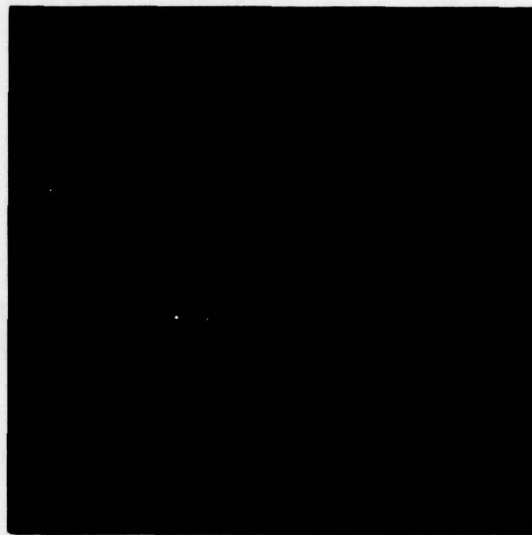


Figure 32. The F1 Controlled Switch Helping in Background Estimation

original FLIR frame containing a hot truck. Figure 33b is the background estimate with the switch. We note that the background is faithfully reproduced in Figure 33c. In Figure 33b we see that the target causes the estimated background to rise with it, which is undesirable in a background estimate.



a. FLIR frame



b. Background estimate without switch

c. Background estimate with switch

Figure 33. The Effect of the Switch in the Background Estimation

Sensitivity to Parameter β

The background filter has the parameter β ($0 < \beta < 1$) and the time constant RC, which need to be tuned for best performance.* Changing β varies the apparent size of the local area over which the background averaging is done. Note that β is the feedback coefficient in the vertical recursive filter. The closer β is to unity, the larger the local area will be. We would like β to be large so that small local variations do not affect the background estimate. But making β too close to unity makes the filter susceptible to CCD noise (generated in the line delay). We experimented with $\beta = 0.8, 0.9$ and 0.95 .

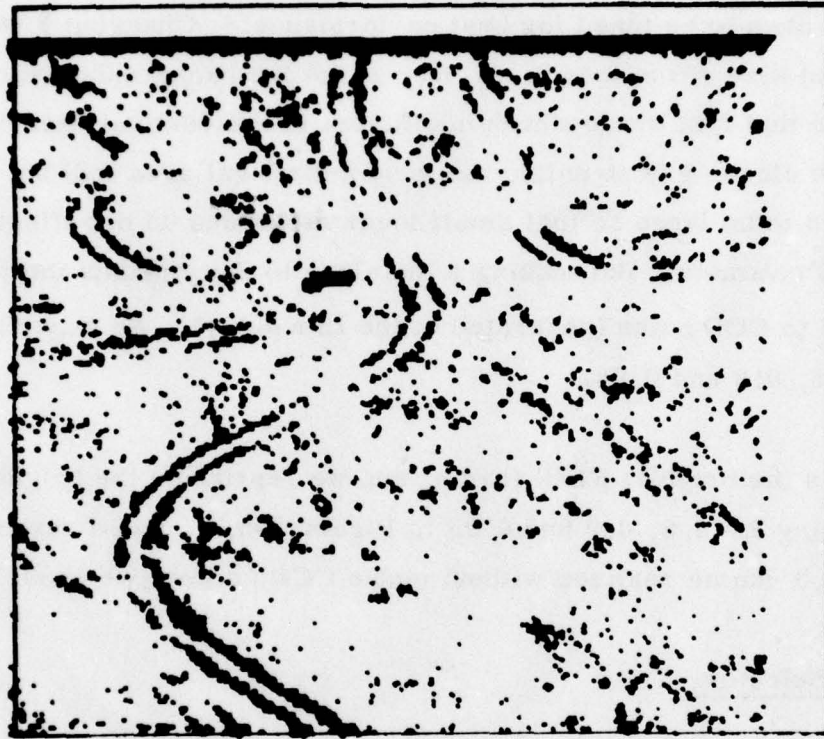
Figure 34 is the original FLIR frame, and we reproduce the brights obtained using $\beta = 0.8, 0.9$ and 0.95 in Figure 34b, c, and d respectively. Since $\beta = 0.9$ can be realized without undue CCD noise**, we chose $\beta = 0.9$.

Threshold Selection

The brights are obtained by thresholding the difference between the image and the local background estimate. We illustrate this in Figure 35 with the intensity profile of an actual thermal image. The usefulness of the background estimate is obvious from this example. Without it, a global threshold would have difficulty extracting the target without getting large chunks of background at the left of the profile.

* For reasons of symmetry, the horizontal filter time constant RC is chosen so that it matches the vertical filter impulse response.

** The vertical recursive filters in the local area gain/brightness broadband use a β of 0.9 .



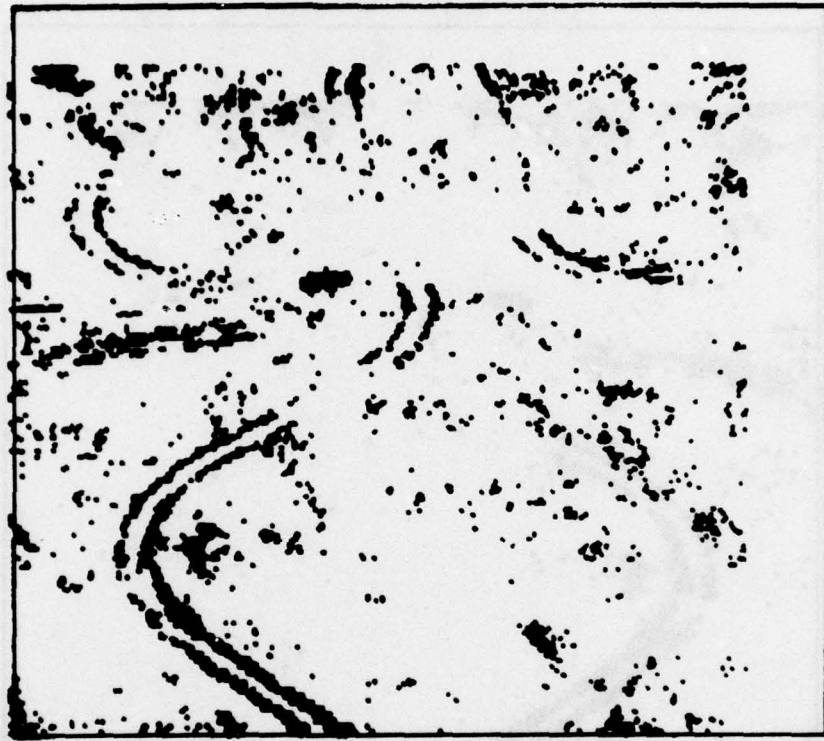
$\alpha = 0.8$

$\beta = 0.8$

$C = 3.0$

$K = 20$

Figure 34a. Brights Obtained by the Autothreshold with $\beta = 0.80$ (No Switch)



$\alpha = 0.9$

$\beta = 0.9$

$C = 3.0$

$K = 40$

Figure 34b. Brights Obtained by the Autothreshold with $\beta = 0.90$ (No Switch)



$\alpha = 0.95$

$\beta = 0.95$

$C = 3.0$

$K = 50$

Figure 34c. Brights Obtained by the Autothreshold with $\beta = 0.95$ (No Switch)

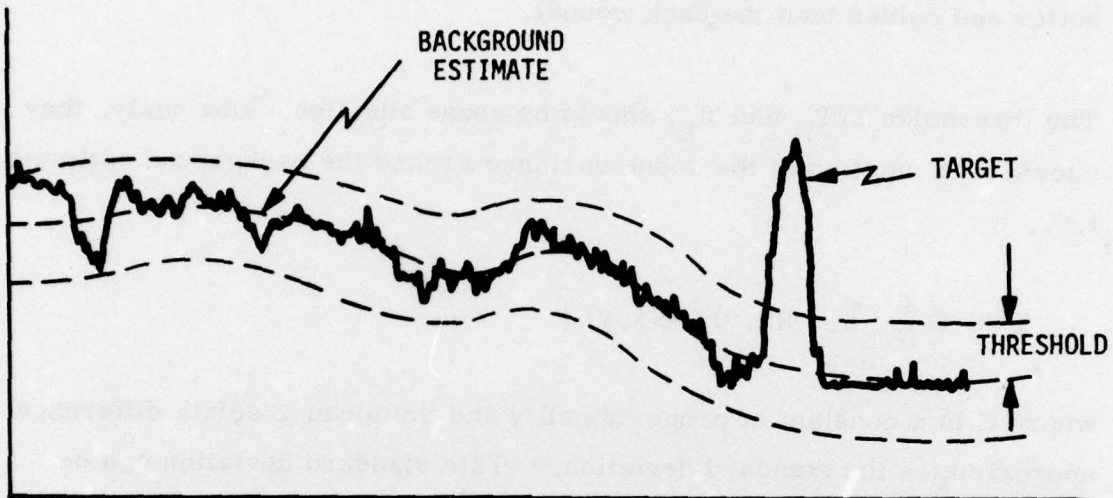


Figure 35. An Actual-Intensity Profile Across a FLIR Image Cross Section. (The background estimate and the threshold contours are shown in broken lines.)

Now, let the image intensity at point (x, y) be $I(x, y)$, and the corresponding background estimate at (x, y) be $b(x, y)$. Then $B(x, y) = I$ (a bright), if

$$I(x, y) - b(x, y) > T_H \quad (\text{Hot threshold})$$

or

$$|I(x, y) - b(x, y)| > T_L \quad (\text{Low contrast threshold})$$

both the threshold definitions are being tested independently.

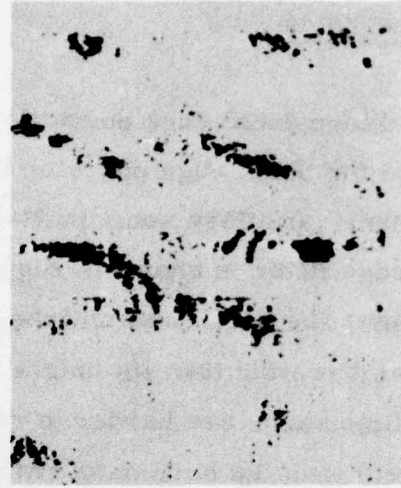
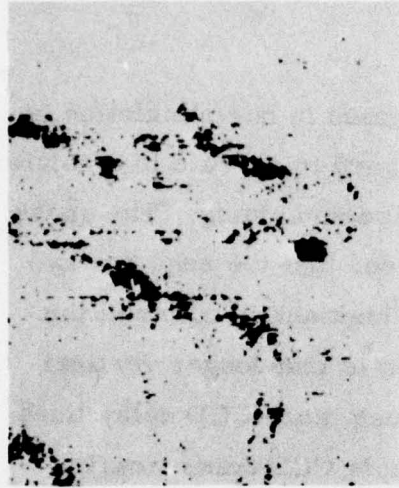
In the first definition, we get only areas that are considerably hotter than the background. In the second we get low contrast differences (both hotter and colder than the background).

The thresholds $T(T_H$ and $T_L)$ should be scene adaptive. Obviously, they should be a function of the scene variance around the background estimate, i. e.,

$$T = C \frac{1}{N} \Sigma | I(x,y) - b(x,y) |$$

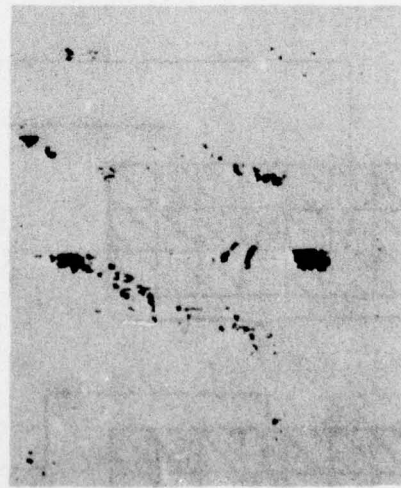
where C is a constant of proportionality and the mean absolute difference approximates the standard deviation.* This standard deviation can be computed either globally over the whole frame, or locally (for example) over the previous scan line. We experimented with both definitions and found that the global estimate gave more robust results. Figure 36a and b are brights obtained using the local and global definitions respectively for $C = 2$. The corresponding results for $C = 3$ are shown in Figure 36c and d. Note that in order to implement the global definition of the threshold, we have to have the background variance over the whole frame. This can be achieved by using the variance estimate from the previous frame as the threshold on the current frame while the new variance is being computed over the current frame.

* It can be shown that, for the Gaussian distribution, the mean absolute difference $= \sqrt{\frac{2}{\pi}} \sigma$, where σ is the standard deviation.



a. Bright image local threshold $C = 2.0$

b. Bright image global threshold $C = 2.0$



c. Bright image local threshold $C = 3.0$

d. Bright image global threshold $C = 3.0$

Figure 36. Comparison of Local and Global Bright Thresholds

Edge Thresholds

The two-dimensional edge operator we incorporated in our simulation is similar to the Sobel edge operator but was designed to operate in a noisier environment. It offers some built-in image noise smoothing. The mask for this edge filter is shown in Figure 37. We see that the edge has two components, the horizontal and the vertical. The mask is wider in the horizontal direction than its height. The reason is that longer vertical window dimensions are harder to realize (because more CCD delay lines are needed) than the horizontal (which have simple CCD transversal filters or tapped CCD delay lines).

The horizontal component of the edge is simply the absolute difference of the average of the nine pixels to the left and right of the current point. The

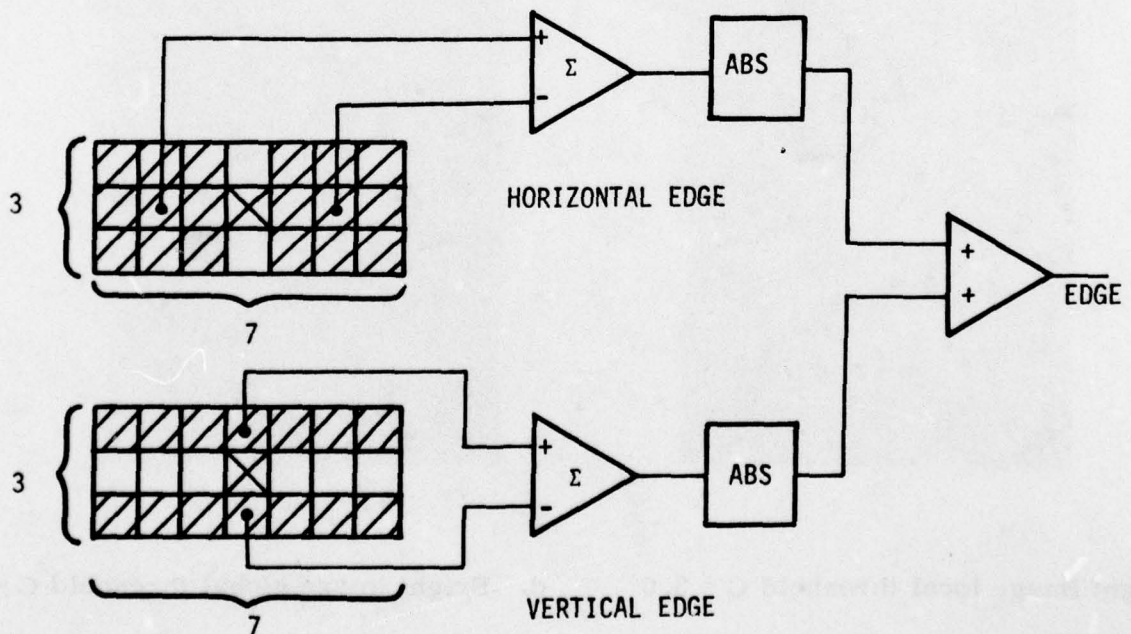


Figure 37. Edge Filter Mask

vertical component is similarly defined in Figure 37. The total edge is then the sum of the two components.

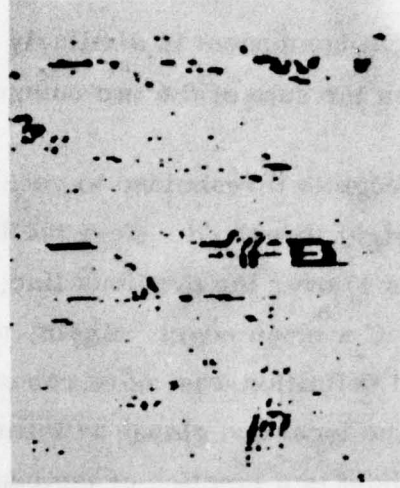
This edge is thresholded to get the binary output in a manner similar to the bright threshold. Here the threshold is a function of the mean edge values 1) over the previous line, or 2) over the previous frame ($T_E = C \times \text{mean edge}$). Again, a comparison of the two showed that the global definition was more robust. We see this in Figure 38a, b, c, and d with the local and global definitions for $C = 2$ and $C = 3$. When the edge threshold is a function of the edges over the previous scan line, the presence of a very "edgy" scan line can cause the threshold to be raised, resulting in missed binary edges on the next scan line.

Object Interval Extraction

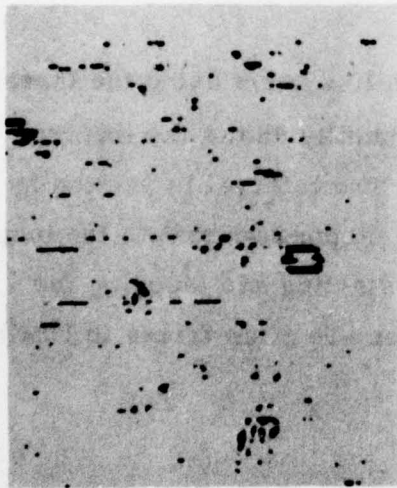
The object intervals are extracted on a scan line basis using the thresholded binary brights and edges. Figure 39 conceptually shows the interval generation on the current ATSS hardware. The interval is started by the coincidence of an edge and the bright. It is stopped when both the bright and edge cease. This lack of symmetry in starting and stopping the object intervals exists because trailing edges are often feeble in FLIR imagery.



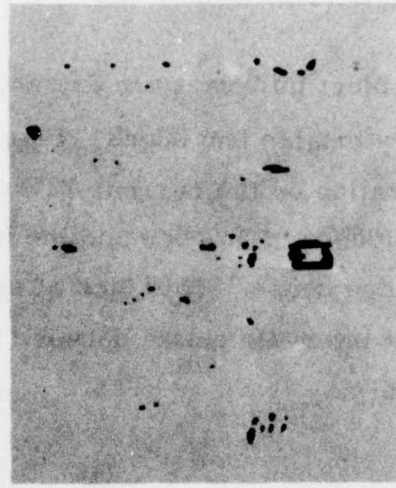
a. Local C = 2.0



b. Global C = 2.0



c. Local C = 3.0



d. Global C = 3.0

Figure 38. Comparison of Global and Local Edge Thresholds

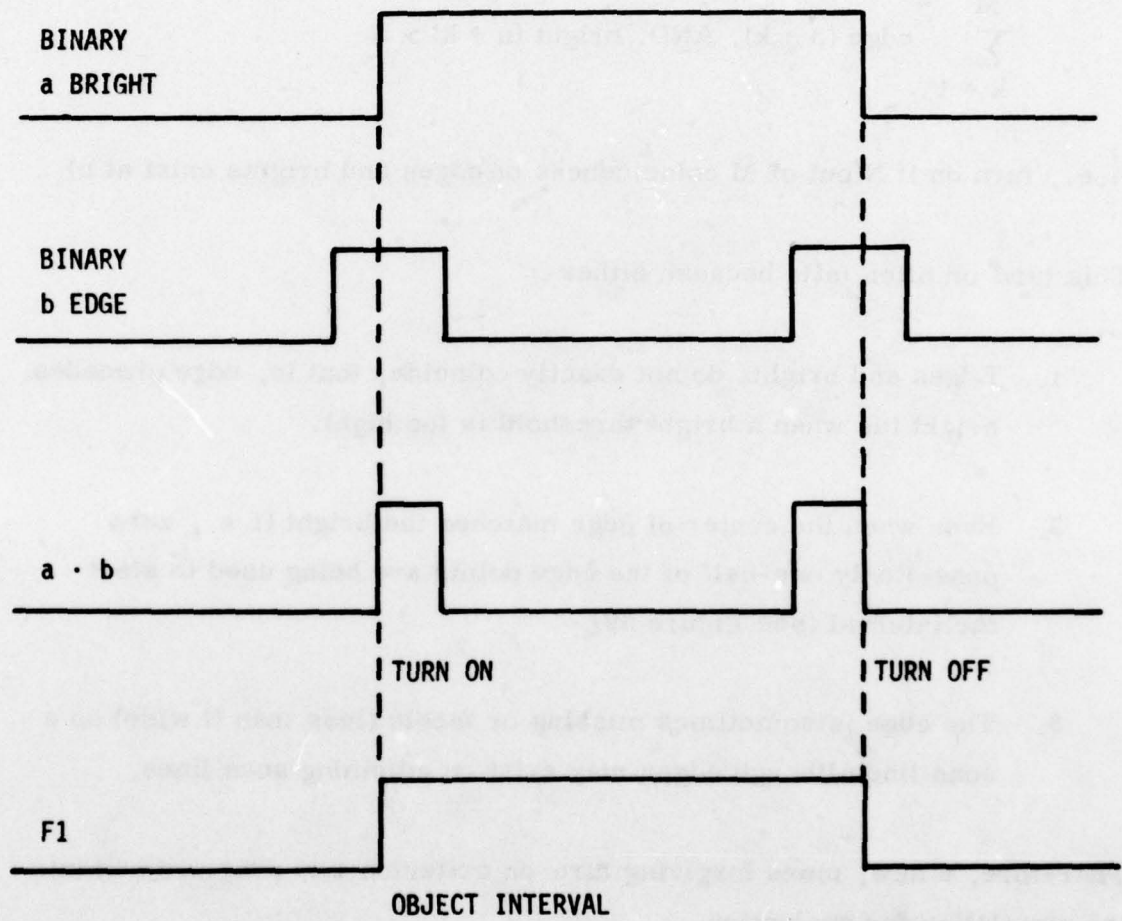


Figure 39. Object Interval Generation Concept

The current turn on criterion on the present ATSS hardware is as follows:

Turn on interval at point n if

$$\sum_{k=1}^M \text{edge}(n+k) \text{ AND } \text{bright}(n+k) > N$$

(i. e., turn on if N out of M coincidences of edges and brights exist at n)

This turn on often fails because either

1. Edges and brights do not exactly coincide, that is, edge precedes bright (as when a bright threshold is too high).
2. Even when the center of edge matches the bright (i. e., zero phase) only one-half of the edge points are being used to start the interval (see Figure 39).
3. The edge is sometimes missing or feeble (less than N wide) on a scan line although edges may exist on adjoining scan lines.

Therefore, a new, more forgiving turn on criterion was programmed into the simulation for evaluation.

The new criterion is illustrated in Figure 40. This was designed to look for the existence of edges on either side of the current scan line and for the presence of edges oriented vertically. This criterion has also been simulated and has proved very robust in detecting the object intervals.

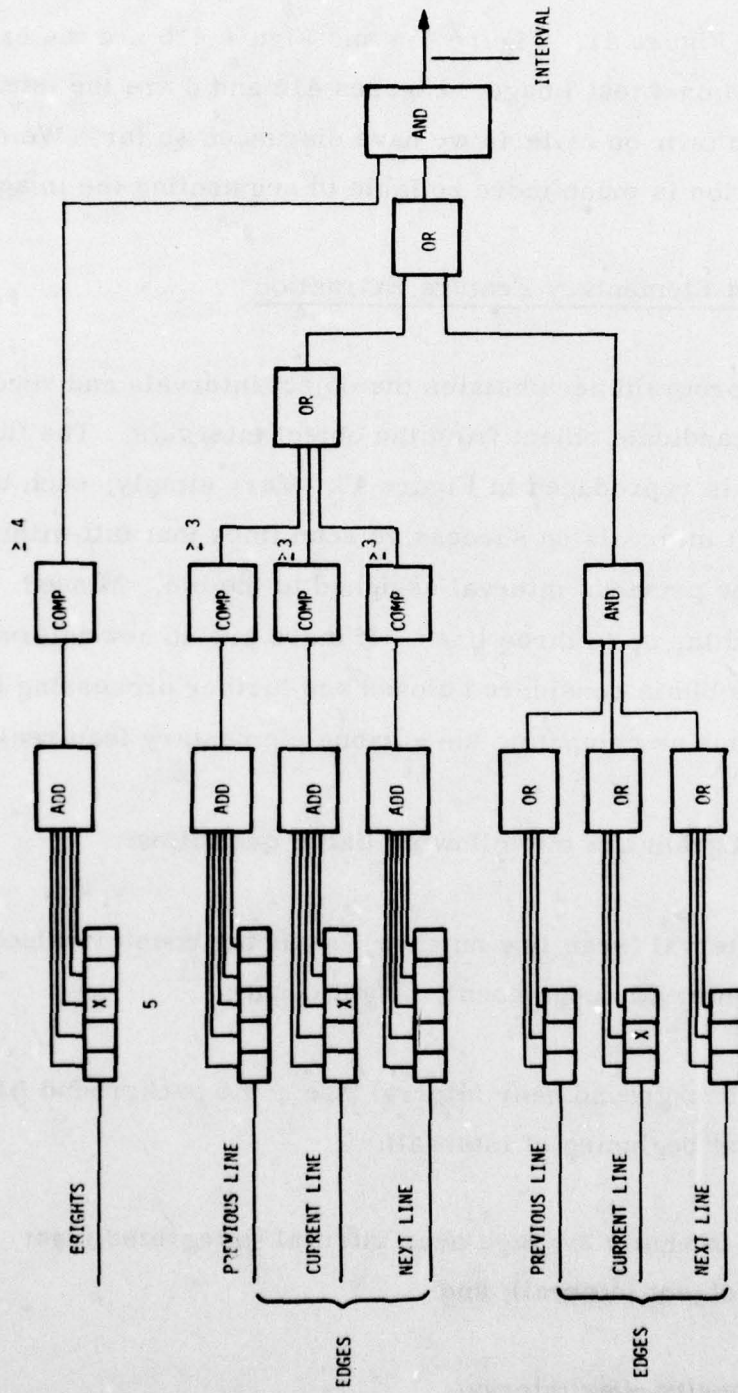


Figure 40. Alternate Turn on Criterion for Object Interval Extraction

We can see this in Figure 41. Figure 41a and Figure 41b are the brights and edges produced on a test image. Figures 41c and d are the intervals obtained by the two turn on criteria we have discussed so far. We note that the new criterion is much more reliable in segmenting the image.

Bin Generation and Elementary Feature Extraction

The bin selection program accumulates the object intervals and records the outline of the candidate object from the object intervals. The flow-chart of this logic is reproduced in Figure 42. Very simply, each bin accumulates object intervals on successive scan lines that fall within the midpoints of the previous interval assigned to the bin. Missed intervals are filled in, up to three lines. If there are no new intervals for scan lines, the bin is considered closed and further processing is done on the bin (such as computing the various elementary features).

Input to the bin program are the following list of quantities:

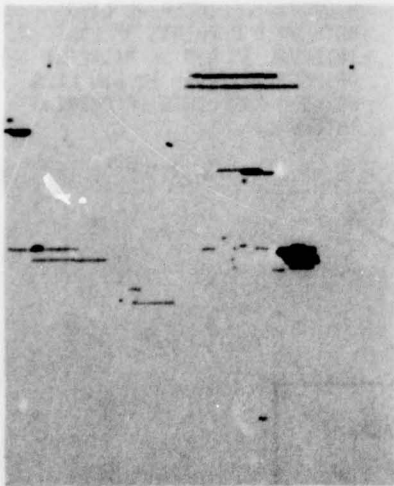
1. Object interval (scan line number, beginning counter value, width of interval, edge count, bright count);
2. Average background near interval (i. e., the background filter sampled at beginning of interval);
3. Absolute intensity average over interval (integrated input over the object interval); and
4. Peak intensity over interval.



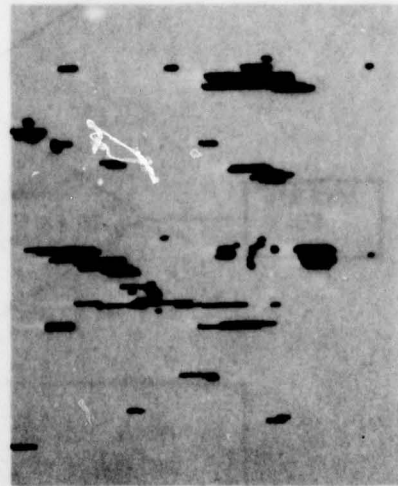
a. Brights



b. Edges



c. Intervals with present ATSS criterion



d. Intervals with alternate criterion

Figure 41. Two Alternate Interval Extraction Criteria

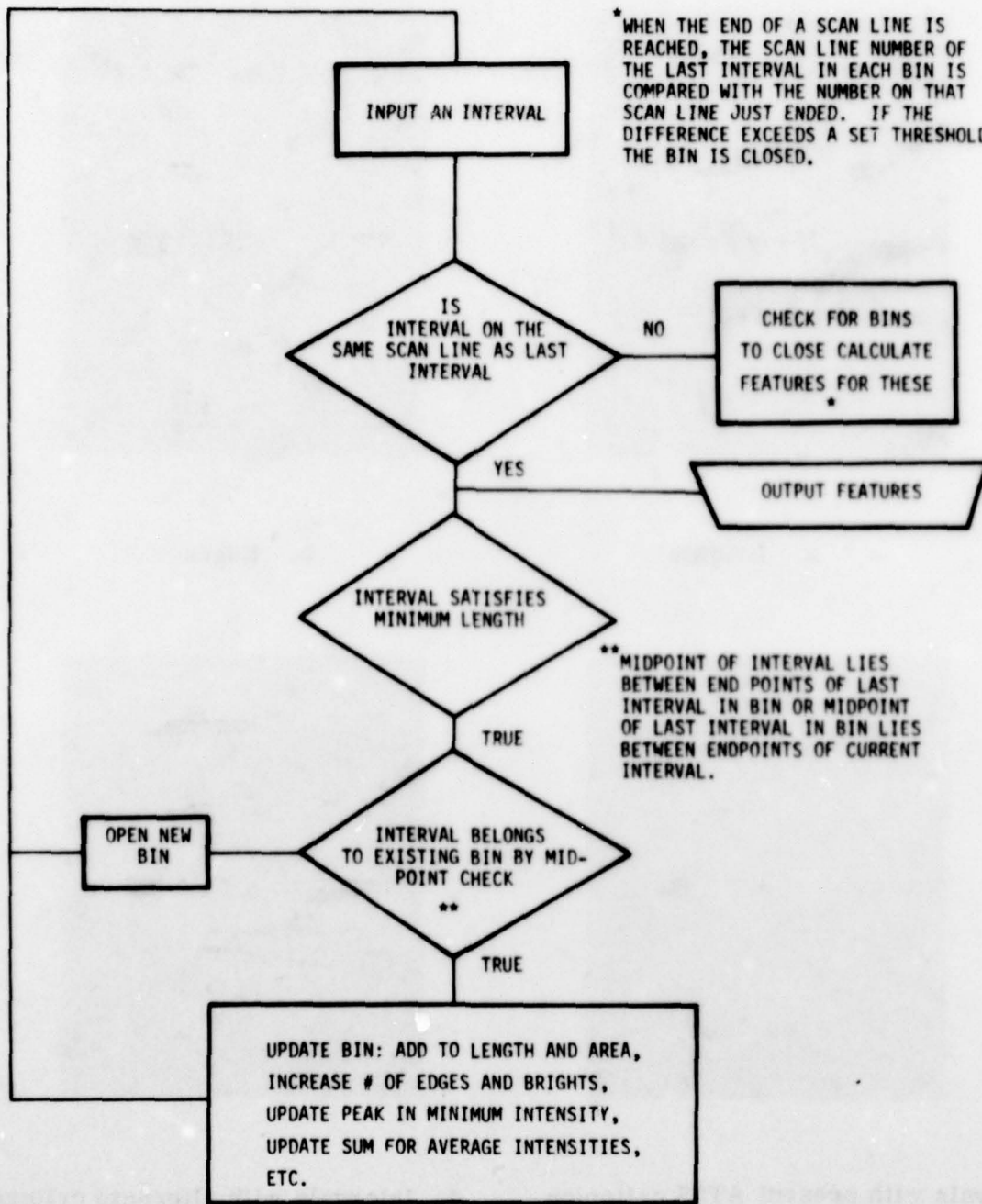


Figure 42. Bin Generation Flowchart

Items 2, 3, and 4 are new to the present ATSS hardware. Their purpose is to introduce intensity (both absolute and contrast) information into the decision making process. These values are further integrated in the bin outputs. The output of the bins are as follows:

1. The outline of the bins (left and right coordinates of all intervals) after median filtering (see below);
2. Total length of target (total number of intervals) including missed intervals;
3. Active length (n), not including missed intervals;
4. Average width $\frac{1}{n} \sum W_i$;
5. Area (sum of active intervals W_i);
6. Edge straightness (S)

$$S = \frac{1}{n-2} \sum |b_{i-2} - 2b_{i-1} + b_i| + |e_{i-2} - 2e_{i-1} + e_i|$$

where b and e are the beginning and end of coordinates of each interval;

7. Edge discontinuity

$$E = \frac{1}{n-1} \sum_{|W_i - W_{i-1}| > 3} |W_i - W_{i-1}|$$

8. Edge count--sum of all edge counts/n;
9. Bright count--sum of all bright counts/Area;
10. Average background--average of all interval background values;
11. Average target intensity

$$\sum_i T_i \cdot W_i / \text{Active Area}$$

where T_i is the average interval intensity, W_i is the interval width.

12. Peak target intensity $\max_i (P_i)$, where P_i is the peak intensity over interval i ;
13. Average width/length ratio;
14. Mean square error of left and right edge fits to straight lines;
15. Slopes of the linear least squares fit to the left and right edges respectively.

Smoothing of Object Boundaries

The extracted object outline is usually far from perfect. There are missed intervals and also intervals that are too long to belong to the outline. The end points of the bin are therefore subjected to a form of boundary smoothing. Rather than linear smoothing that rounds off sharp corners, we are experimenting with median filtering of these boundaries. Instead of averaging the end points over a window three pixels wide, each end point (left or right) is replaced by the median (middle value) of three end point coordinates, including its two neighbors and itself. This gets rid of excessively long lines and very short lines flanked by "normal" end points. Figures 43a and b are the intervals generated and the actual bin outlines obtained for a FLIR image processed through the simulation. Note the end point achieved by this filtering and by the bin generation logic.

RECOGNITION FEATURES

Two classes of recognition features are being evaluated. They are the moment features and the Fourier boundary descriptors (FBD).

Moment Features

Two-dimensional moment features have been used in character recognition⁴ and aircraft identification⁵. These are intensity moments,

⁴M. K. Hu, "Visual Pattern Recognition by Moment Invariants," IRE Transactions on Information Theory, pp. 179-187, February 1962.

⁵S. A. Dudani, et. al., "Aircraft Identification by Moment Invariants," IEEE Transactions on Computers, pp. 39-46, January 1977.



Figure 43a. Object Intervals Extracted

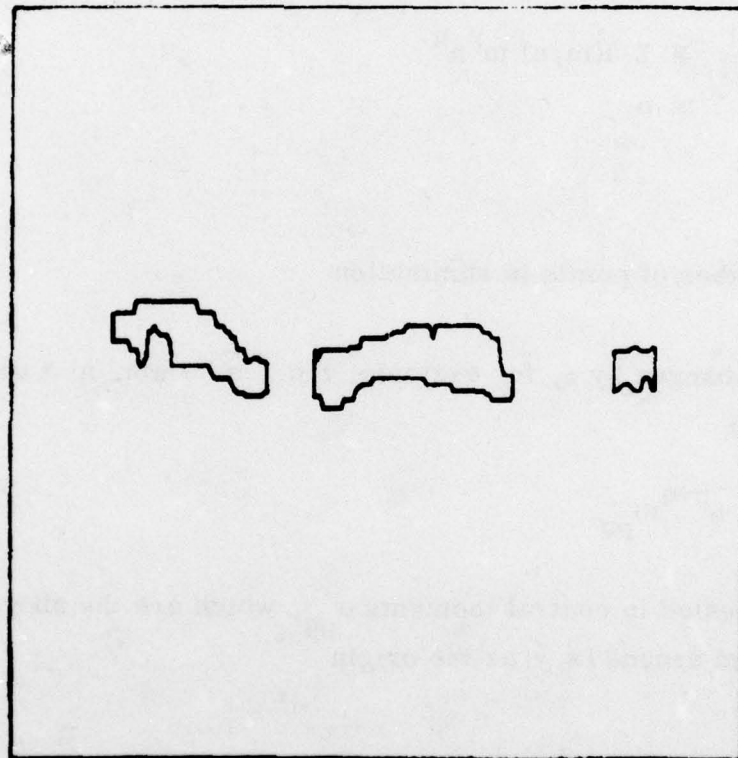


Figure 43b. Boundary of Objects Traced by the Bin
(After Smoothing the Edges)

silhouette (area), and boundary moments (binary). We are evaluating all three classes as candidates.

Let $I(m, n)$ be the object intensity. The $(p, q)^{\text{th}}$ moment given by

$$m_{pq} = \frac{1}{N} \sum_m \sum_n I(m, n) m^p n^q$$

where

N = number of points in summation

If the scale changes by α , for example, i. e., $m' = \alpha m$, $n' = \alpha n$, then we can show that

$$m'_{pq} = \alpha^{p+q} m_{pq} \quad (1)$$

We are interested in central moments μ_{pq} , which are the above moments m_{pq} evaluated around (\bar{x}, \bar{y}) as the origin

where

$$\bar{x} = m_{10} / m_{00}$$

$$\bar{y} = m_{01} / m_{00}$$

It is clear that μ_{pq} can be expressed in terms of m_{pq} and (\bar{x}, \bar{y}) .

AD-A050 684

HONEYWELL INC MINNEAPOLIS MINN SYSTEMS AND RESEARCH --ETC F/G 17/5
PROTOTYPE AUTOMATIC TARGET SCREENER.(U)

JAN 78 D E SOLAND, P M NARENDRA, R C FITCH

DAAK70-77-C-0248

UNCLASSIFIED

78SRC6

NL

2 OF 2

AD
A050684



END
DATE
FILMED
4 -78
DDC

Invariance to Size--From Equation (1), we can see that

$$\mu'_{pq} = \alpha^{p+q} \mu_{pq}$$

$$\mu'_{02} + \mu'_{20} = \alpha^2 (\mu_{02} + \mu_{20})$$

Consider

$$\frac{\mu'_{pq}}{(\mu'_{02} + \mu'_{20})^{\frac{p+q}{2}}} = \frac{\mu_{pq} \alpha^{p+q}}{(\mu_{02} + \mu_{20})^{\frac{p+q}{2}} \alpha^{p+q}} = \frac{\mu_{pq}}{(\mu_{02} + \mu_{20})^{\frac{p+q}{2}}}$$

which is invariant to scale α .

Now, $r^2 = \mu_{02} + \mu_{20}$ is often called the radius of gyration. Therefore, we normalize all μ_{pq} by $\mu_{pq} / r^{(p+q)}$, which makes them invariant to size.

There are three different kinds of moments we are exploring:

1. Intensity moments (as above with $I(x, y)$ representing the target intensities);
2. Silhouette moments ($I(x, y) \equiv 1$ over all points within target boundary); and
3. Boundary moments ($I(x, y) \equiv 1$ on boundary; 0 elsewhere).

For each class, we compute all moments up to and including third order moments, i. e.,

$$\mu_{00}, \mu_{11}, \mu_{20}, \mu_{02}, \mu_{21}, \mu_{03}, \mu_{30}, \mu_{12}$$

Note that $\mu_{10} \equiv \mu_{01} \equiv 0$ because these are central moments.

1. Intensity Moments

Only the target intensities $I(m, n)$, which are bounded by the outline extracted by the autothreshold (bin intervals), are used.

As mentioned before, all μ_{pq} are normalized by r^{p+q} for size invariance. They are also divided by $m_{00} = \frac{1}{N} \sum \sum I(m, n)$ to normalize them by the average target intensity.

2. Silhouette Moments

We do not need the original image intensities for the silhouette moments.

$$m_{pq} = \frac{1}{N} \sum_m \sum_n m^p n^q, \text{ because } I(m, n) = \begin{matrix} 1 & \text{inside object,} \\ 0 & \text{elsewhere} \end{matrix}$$

where (m, n) are all points contained inside the boundary extracted.

The centroid is defined as before, (\bar{x}, \bar{y})

$$\bar{x} = m_{10}, \bar{y} = m_{01}$$

(Note that $m_{00} \equiv 1$)

Again, μ_{pq} are normalized with respect to r^{p+q} .

3. Boundary Moments

$$m_{pq} = \sum_m \sum_n m^p n^q$$

where m, n are points on the boundary extracted by the image segmentation.

In all of the above, it was found convenient to first compute (\bar{x}, \bar{y}) and then compute the central moments directly, instead of computing μ_{pq} from m_{pq} .

Invariance to Rotation--Note that the above moments are not invariant to rotation. In this particular application, we do not want invariance to rotation because a right-side up tank would then be indistinguishable from a piece of clutter that had the same shape as an upside-down tank! Note, however, that invariance to size is assured by the normalization with respect to the radius of gyration.

Fourier Boundary Descriptors

Since human target recognition is often done using the target outline (boundary) in FLIR imagery, we are evaluating boundary shape descriptors as potential recognition features. The boundary shape descriptors are derived from the object boundary extracted by the bin generation program in the form of the beginning and end coordinates of successive intervals in the bin. The approach we are taking is similar to Zahn.⁶ The object boundary is first encoded in a periodic angle versus arc length waveform. The one-dimensional discrete Fourier transform (DFT) of this waveform is then taken. The first few amplitude and phase coefficients are used as shape discriminators. If these Fourier boundary descriptors (FBD) turn out to be useful shape discriminators in the PATS simulation, their implementation via the new CCD Fast Fourier Transform (FFT) module* is deemed simple. Following is a brief description of this process in the current simulation.

Given the object intervals as in Figure 44a, we have to encode the outline in a simple $\theta(l)$ form (θ is the angle with respect to the X axis; l is the arc length from an origin). In a discrete situation, as here, we have to assume that the arc lengths are not all equal. Let the object

⁶C. T. Zahn and R. Z. Roskies, "Fourier Descriptors for Plane Closed Curves," IEEE Transactions on Computers, pp. 269-281, March 1972.

* Reticon 5601 CCD FFT Module.

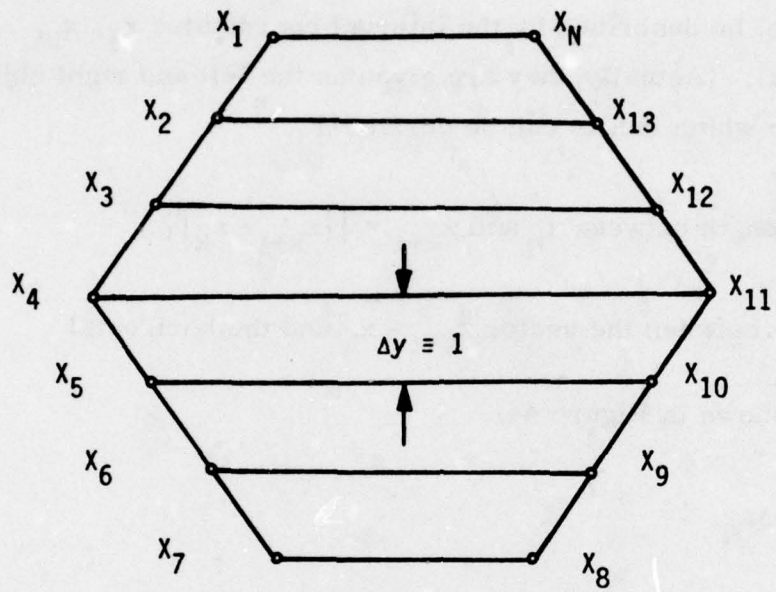


Figure 44a. Boundary Representation Obtained by Autoscreener Segmentation

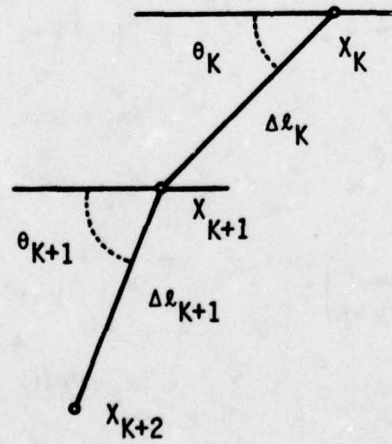


Figure 44b. Angle vs. Arc Length Representation of the Segment Boundary

(a simple curve) be described by the interval coordinates x_1, x_2, \dots, x_N (see Figure 44a). (Actually, they are given as the left and right object intervals, from which the X s can be derived.)

Let $\Delta l_k =$ arc length between x_k and $x_{k+1} = ||x_{k+1} - x_k||$

$\theta_k =$ angle between the vector $x_{k+1} - x_k$ and the horizontal

(measured as shown in Figure 44)

$$\text{and } l_k = \sum_{i=1}^k \Delta l_i$$

Then we need to derive the list $\{(\theta_1, l_1), (\theta_2, l_2), \dots, (\theta_N, l_N)\}$ from x_1, \dots, x_N . Note that the spacing between successive scan lines (intervals) is always 1 (unity); i. e., $|\Delta y| \cong 1$. Therefore, (x_{k+1}, x_k) uniquely determine θ_k and Δl_k as follows:

$$\Delta l_k = \begin{cases} \sqrt{1 + (x_k - x_{k+1})^2} & , \quad x_k, x_{k+1} \in \text{left or right} \\ |x_k - x_{k+1}| & , \quad (x_k \in \text{left}, x_{k+1} \in \text{right}), \text{ or} \\ & , \quad x_k \in \text{right}, x_{k+1} \in \text{left} \end{cases}$$

$$\theta_k = \begin{cases} \arctan\left(\frac{1}{x_k - x_{k+1}}\right) & , \quad x_k, x_{k+1} \in \text{left boundary} \\ 180^\circ & , \quad x_k \in \text{left}, x_{k+1} \in \text{right} \\ \arctan\left(\frac{1}{x_k - x_{k+1}}\right) + 180 & , \quad x_k, x_{k+1} \in \text{right boundary} \\ 0 & , \quad x_k \in \text{right}, x_{k+1} \in \text{left} \end{cases}$$

These are implemented in a lookup table for different values of $x_k - x_{k+1}$, and the sign of Δy (i.e., left or right boundary). (This saves evaluation of square roots and trigonometric functions in the PATS processor.)

The next step is to compute the Fourier transform of the $\theta(l)$ curve defined by the sequence $(\theta_1, l_1), \dots, (\theta_N, l_N)$. But this does not represent equally spaced samples of θ as a function of l (see Figure 45). We also need to normalize the l axis (arc length) to a constant perimeter. Say that total perimeter = 512 (a 512-point DFT can then be used). As we see in Figure 45, these $\theta'_n = \theta(n\Delta l)$ are determined from the list (θ_i, l_i) by replication ($\Delta l = l_N/512$).

Once $\theta'_i, i = 1, \dots, 512$ are determined, we need to arrive at φ_i as follows:

$$\varphi_i = \theta'_i - 360 \frac{i-1}{512} - \theta'_1$$

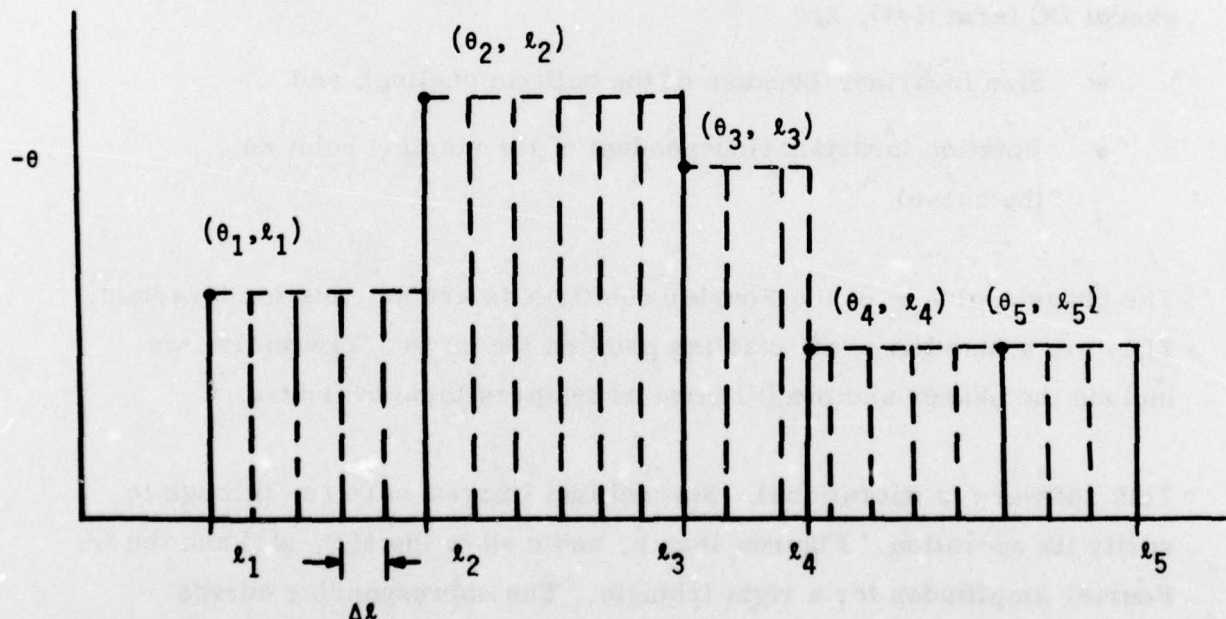


Figure 45. Derivation of the Equispaced Values of $\theta(l)$ as a Function of l , from the Set $\{(\theta_1, l_1), \dots, (\theta_N, l_N)\}$

Note that if we had a circle to begin with, then all the $\phi_i = 0$. This is done to assure that the dynamic range of the FFT is more fully utilized. Thus, ϕ_i measure the departure of the object shape from the circle.

Let $\phi(n) n = 1, \dots, 512$ be the complex FFT of the ϕ_i s obtained above.

The amplitude coefficients are given by

$$A(n) = \sqrt{[\text{Re}\phi(n)]^2 + [\text{Im}\phi(n)]^2}$$

and phase

$$\alpha(n) = \arctan \frac{\text{Im}\phi(n)}{\text{Re}\phi(n)}$$

Properties of the Fourier Descriptors -- These properties are discussed at length by Zahn. For our purposes, we note that the amplitudes A_i , except DC term ($i=1$), are

- Size invariant (because of the built-in scaling); and
- Rotation invariant (independent of the starting point on the curve)

The phase angles α_i of the Fourier coefficients are not rotation invariant. They are a function of the starting point on the curve. Therefore, we include the phases and the DC term as features to be evaluated.

This software is operational. Several test images were run through to verify its operation. Figures 46a, b, and c show the $\theta(L)$, $\phi(1)$ and the Fourier amplitudes for a right triangle. The corresponding curves for a rotated right triangle are reproduced in Figures 47a, b, and c. This shows the invariance of the amplitude features to rotation.

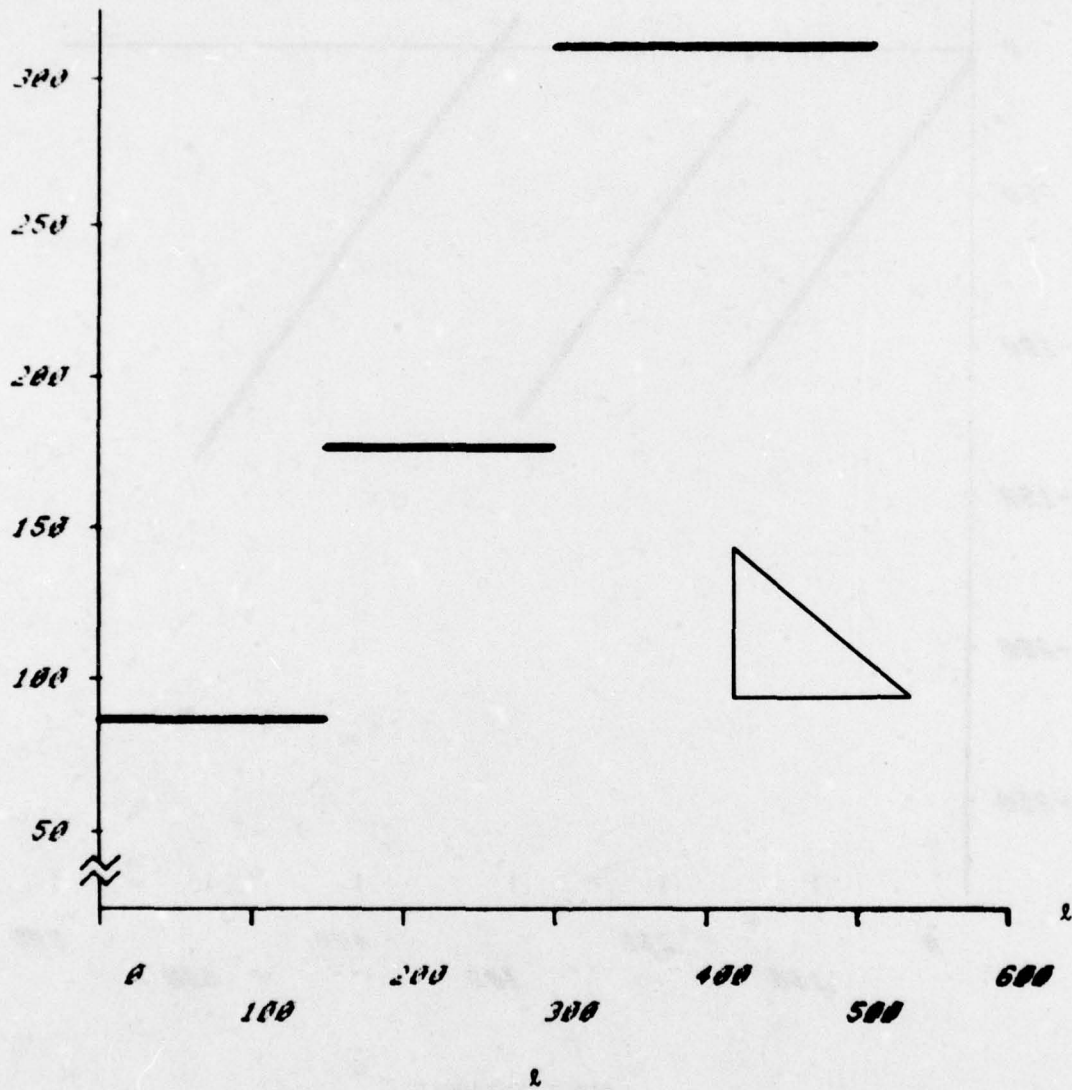
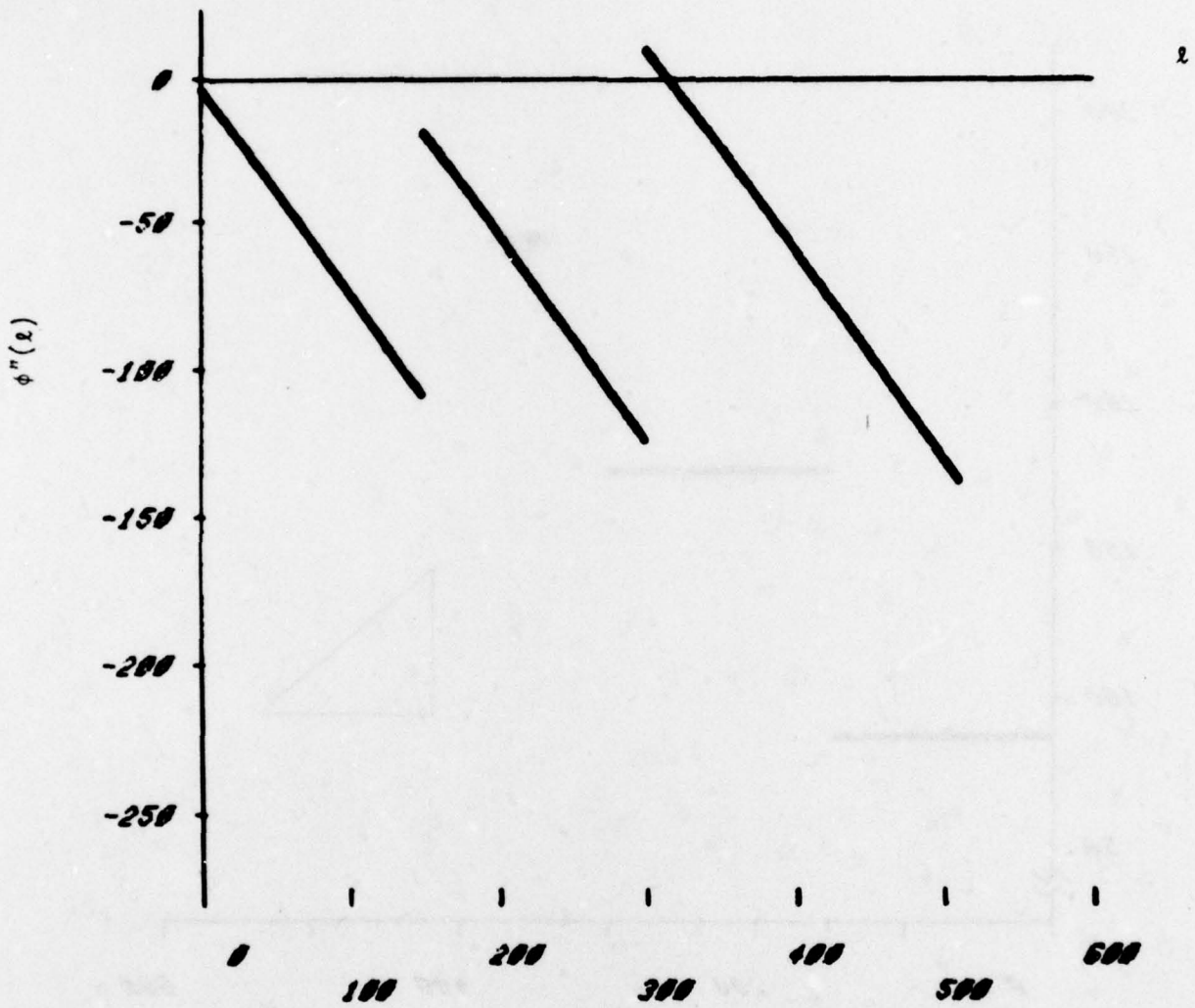


Figure 46a. $\theta(l)$ Representation for a Right Triangle



RIGHT TRIANGLE

Figure 46b. $\phi''(z)$ Representation for the Right Triangle

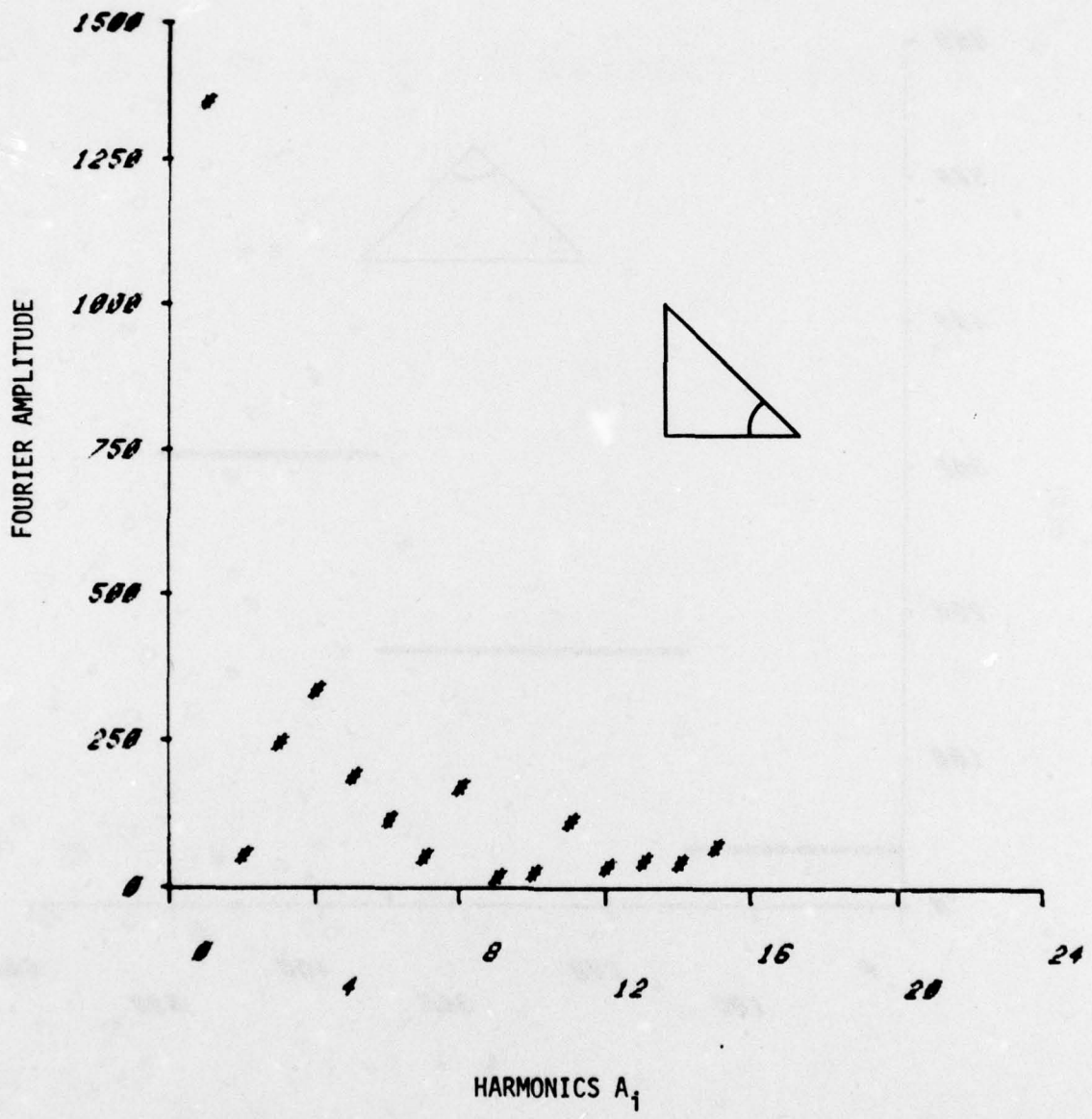


Figure 46c. Fourier Harmonic Amplitudes for the Right Triangle

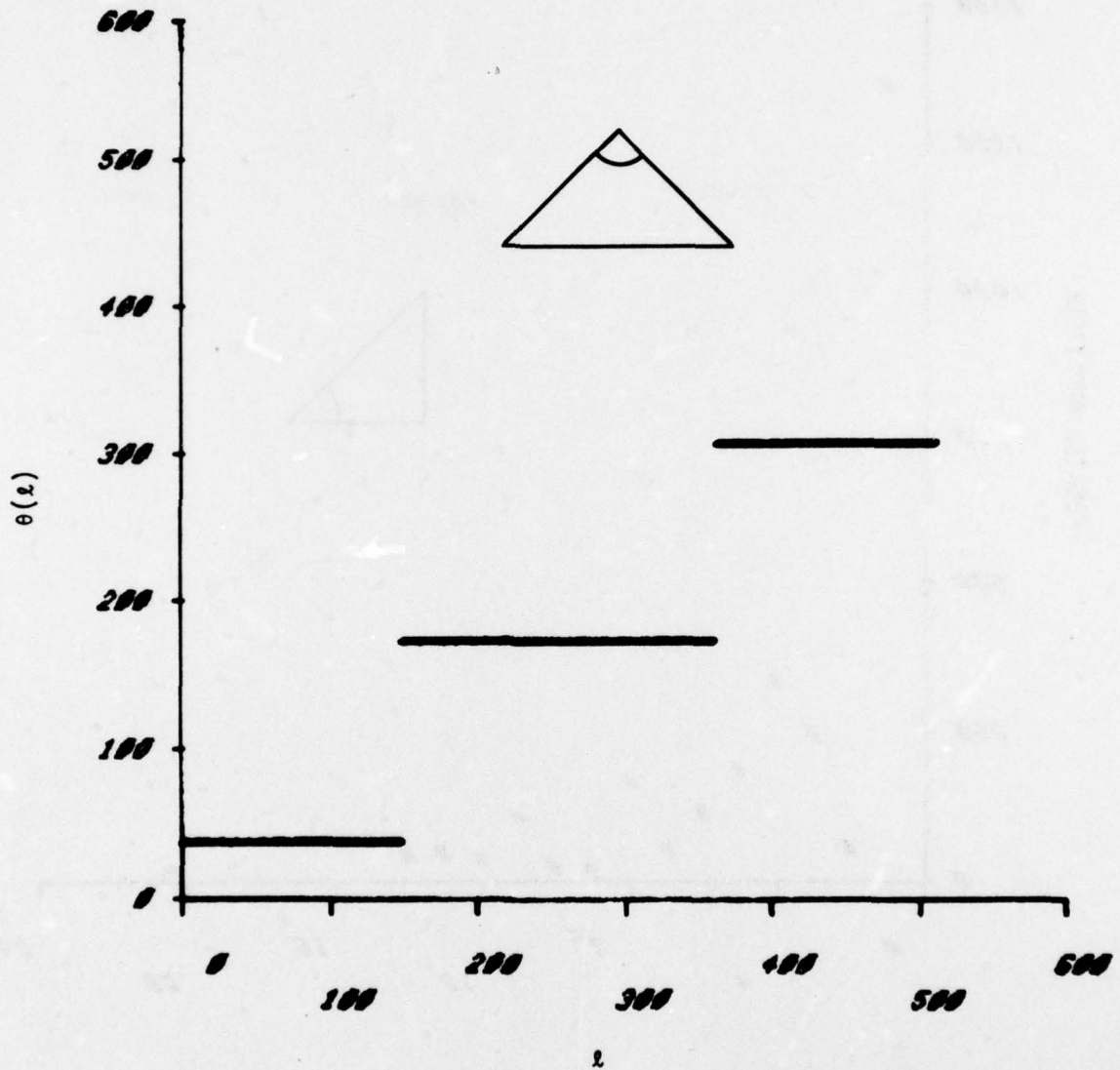


Figure 47a. $\theta(l)$ Representation for a Rotated Right Triangle

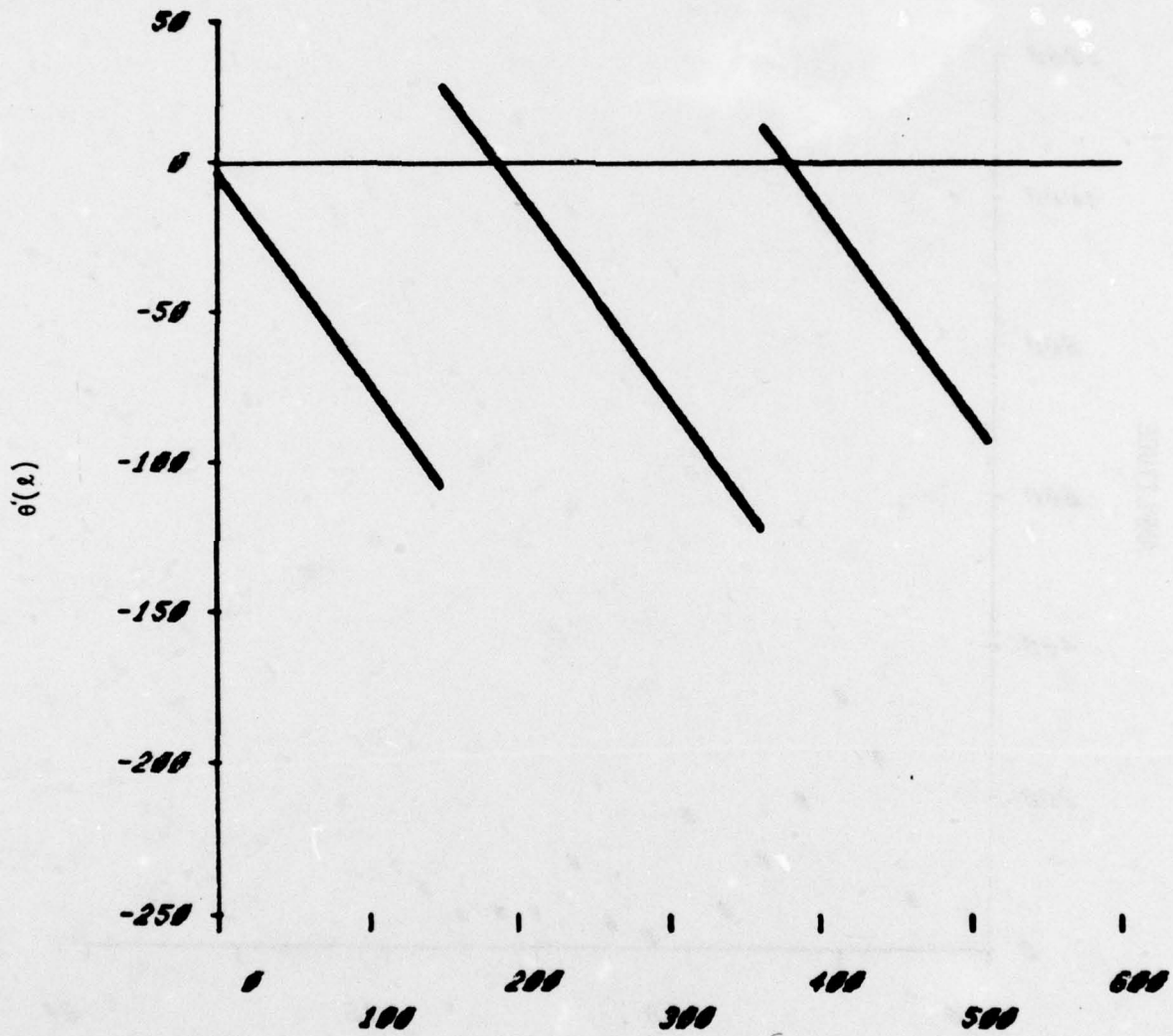


Figure 47b. $\varphi'(l)$ Representation for the Rotated Right Triangle

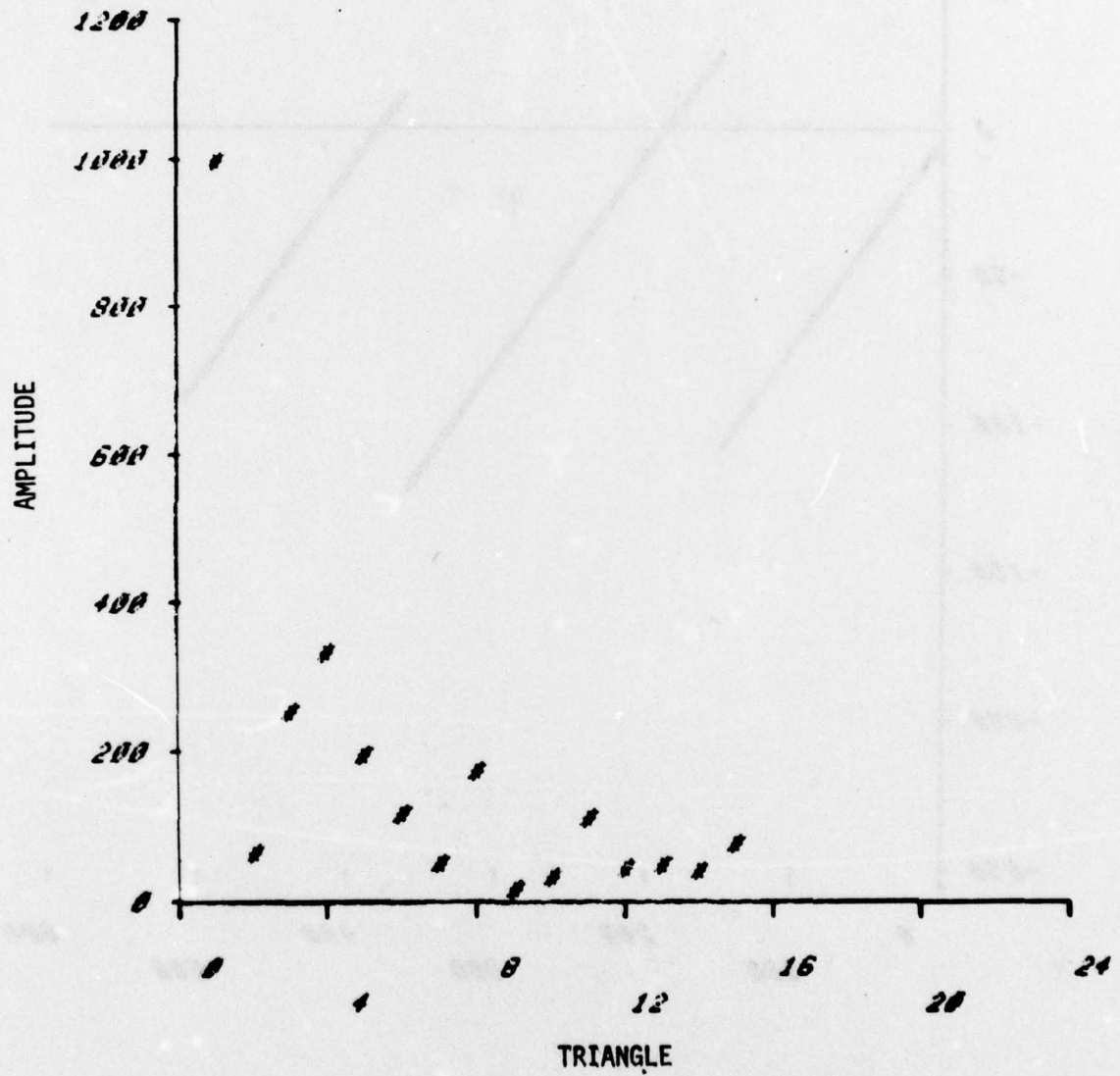


Figure 47c. Fourier Harmonic Amplitudes for the Rotated Right Triangle

SYSTEM SIMULATION SOFTWARE

All the components of the PATS system simulation up to and including the feature extraction/analysis stage have been assembled. The outline of this software is shown in Figure 48. The output of the bin and feature extraction program includes a magnetic tape file consisting of all the bin coordinates (boundary representation), the various elementary features (size, average intensity, edge and bright counts, length/width, etc.), and the recognition features (the moment and Fourier features). These features are combined in the File Supervisor program with the ground truth (class of target or clutter, aspect, size, etc.) to generate a feature tape to be input to the analysis software. The analysis software includes two-dimensional scatter (cluster) plots and histograms of single features. We are now interfacing the classifier software to this output so that discriminant analysis can be used to test the features.

A sample set of three FLIR frames processed through the entire simulation is reproduced in Figures 49 through 51. Note that the high contrast truck in Figure 49 is clearly segmented, as is the lower contrast tank in Figure 50. The bins obtained from these images were processed to get the elementary features and the recognition features. These contain the three targets, as well as four clutter objects identified on the three frames.

We reproduce some of these features measured on the bins in Table 3. Because of space limitations, not all of the features listed in Table 3 are represented. It is of interest that the "edge count" feature is consistently

TABLE 3. SOME PATS FEATURES ON THREE TARGETS AND FOUR CLUTTER OBJECTS EXTRACTED BY SIMULATION

Target Type	Length	Avg. Width	Length/ Avg. Width	Area	Straight- ness	Edge Discontinuity	Edge Count	Bright Count	Avg. Target	Avg. Contrast	Peak Target	Minimum Target
Tank	8	15	533	121	7000	571	11625	1000	182	79	229	127
Truck	32	12	2666	390	7233	1483	8531	969	198	85	255	11
Truck	8	24	333	165	11200	2500	11000	945	211	69	255	131
Clutter	8	10	799	60	9250	3400	5333	883	166	46	206	125
Clutter	15	20	750	284	15333	4461	2571	936	169	65	215	119
Clutter	7	8	875	57	10000	3500	6714	964	145	77	201	98
Clutter	6	5	1.2	30	5000	200	3000	1000	149	70	177	116

Target Type	Fourier Descriptors						Scaled Intensity Moments								
	A ₀	A ₁	A ₂	A ₃	c ₁	c ₂	c ₃	μ ₀₂	μ ₁₂	μ ₂₁	μ ₃₀	Left Edge Fit	Right Edge	θ _L	θ _R
Tank	642	41	320	125	0.140	0.424	1.445	0.23	0.81	2.67	1.60	5299	2460	2.40	0.63
Truck	532	142	507	488	-0.881	1.134	0.127	0.81	-1.34	-1.34	1.12	48277	228352	0.91	0.48
Truck	1607	100	444	309	0.981	-1.33	-0.256	-0.11	4.3	4.3	-15.5	29050	1296	1.28	2.31
Clutter	1270	155	289	577	-1.27	1.05	-1.47	1.2	7.7	7.7	-26.2	37101	6615	1.28	1.99
Clutter	3403	134	244	216	0.174	-0.025	-0.756	-0.65	1.59	1.59	-6.22	23681	11117	1.14	1.02
Clutter	1101	28	157	81	0.089	1.04	0.973	-0.19	-1.96	-1.96	-0.14	16044	18428	0.44	0.38
Clutter	1066	19	47	131	0.862	0.608	1.35	-2.7	-1.96	-1.96	3.19	0	1229	1.57	2.04

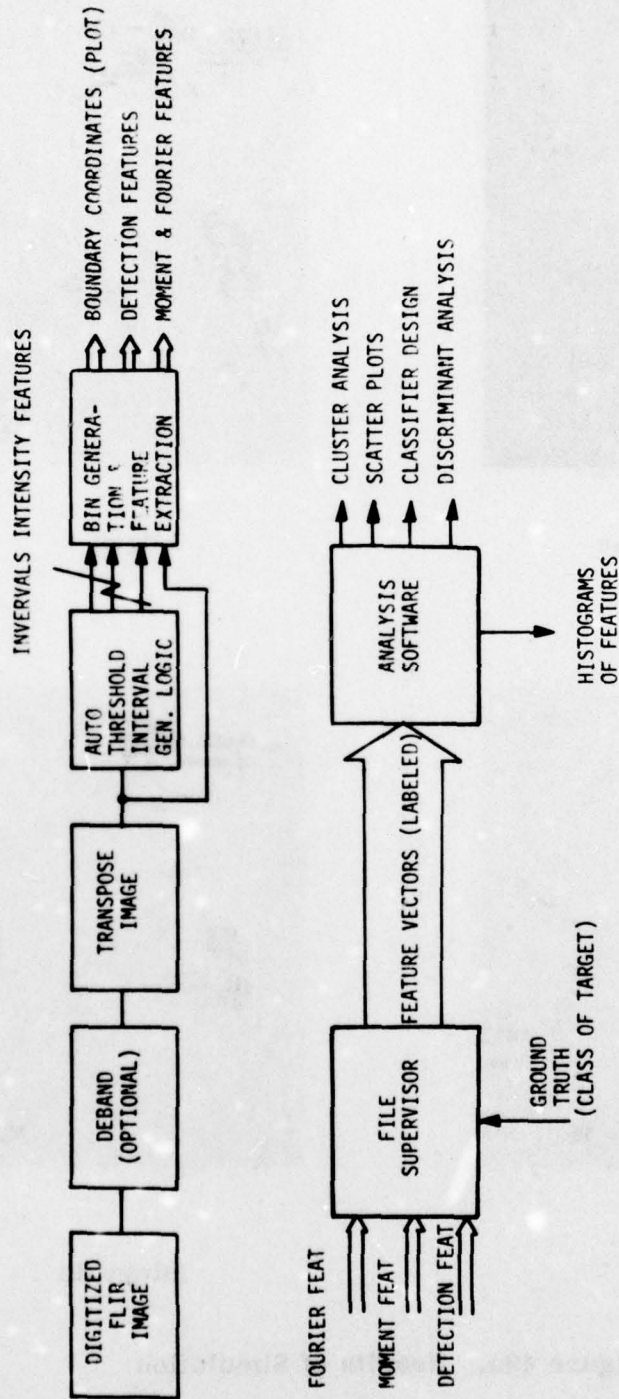


Figure 48. PATS Screening Simulation Software Outline



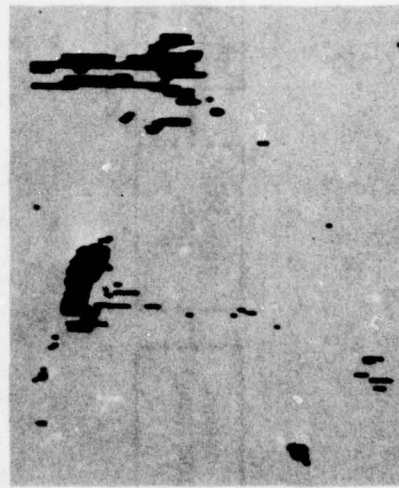
FLIR frame



Edges



Brights



Intervals

Figure 49a. Results of Simulation

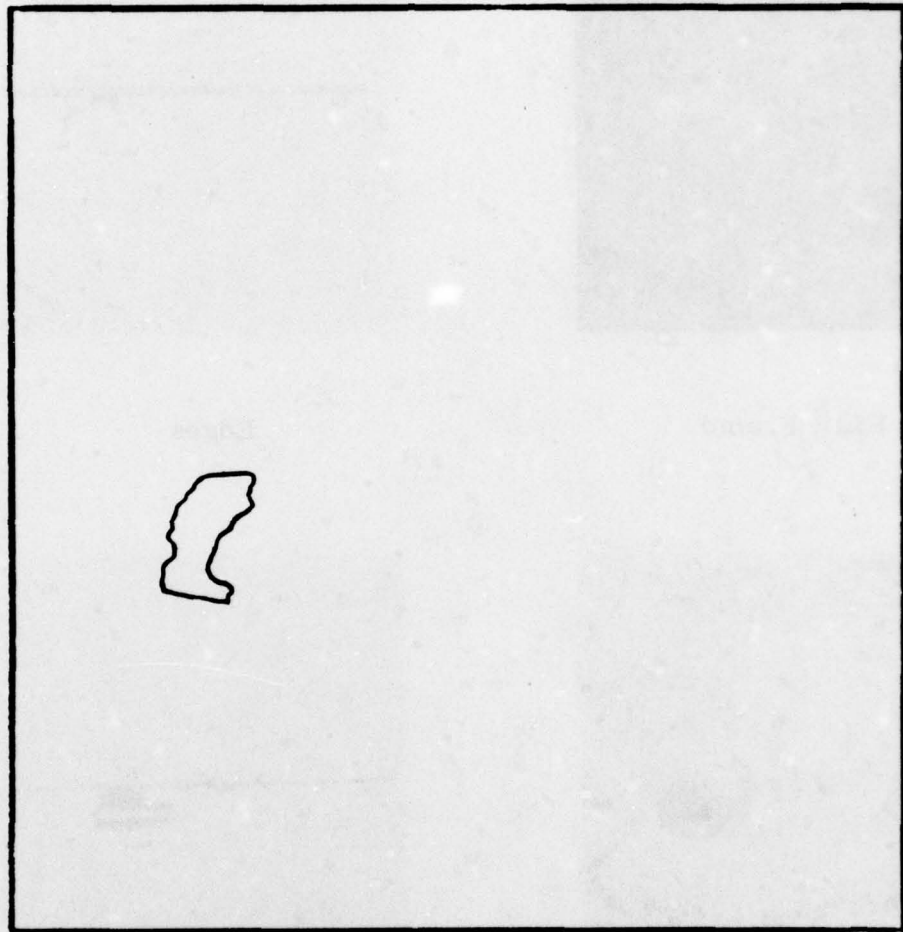
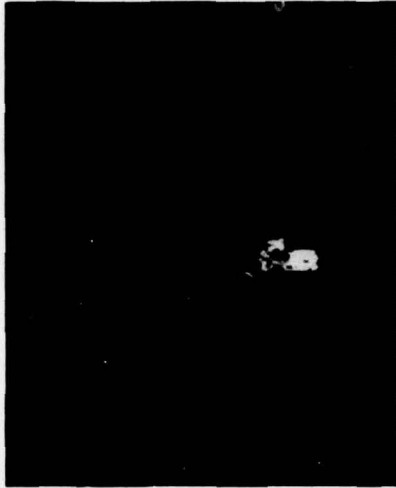


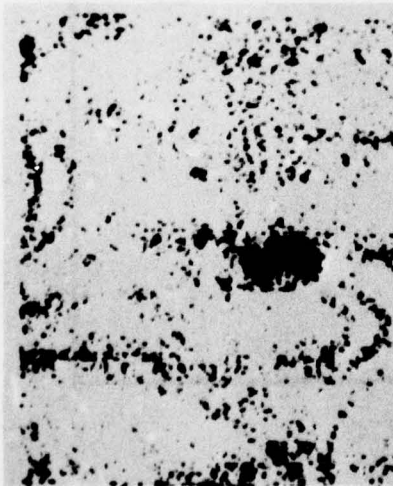
Figure 49b. Results of Simulation Candidate Object Outline Extracted by the Bin Generation Program from Intervals in Figure 49a



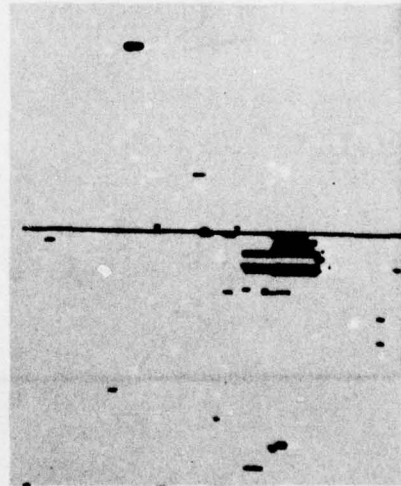
FLIR Frame



Edges



Brights



Intervals

Figure 50a. Results of Simulation

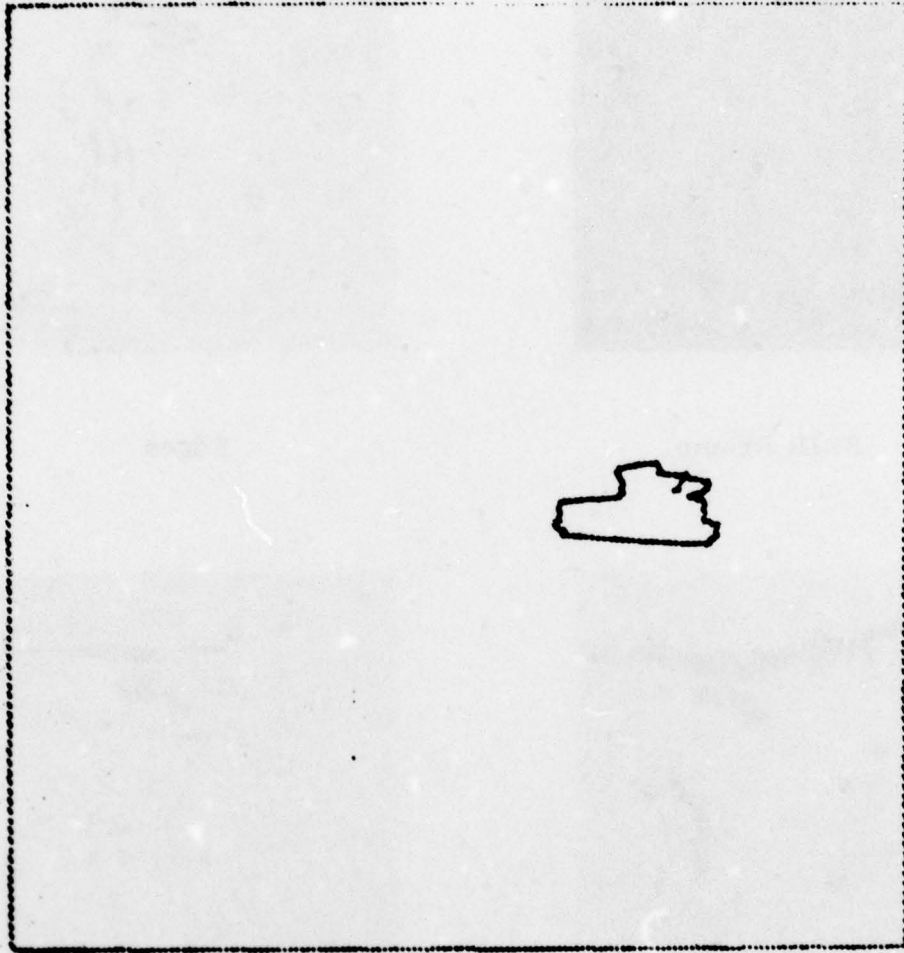
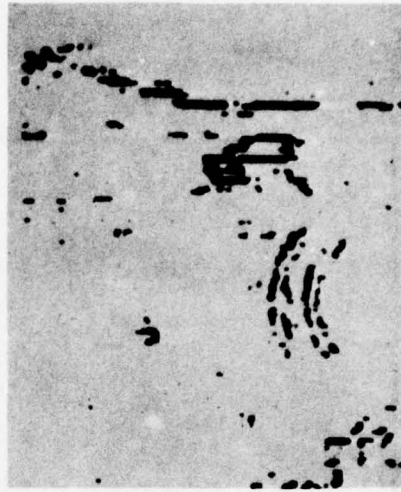


Figure 50b. Results of Simulation Outline Extracted by the Bin Generation Program from the Intervals in Figure 49a



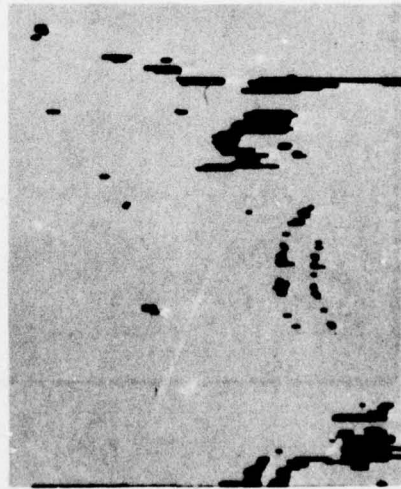
FLIR Frame



Edges



Brights



Intervals

Figure 51a. Results of Simulation

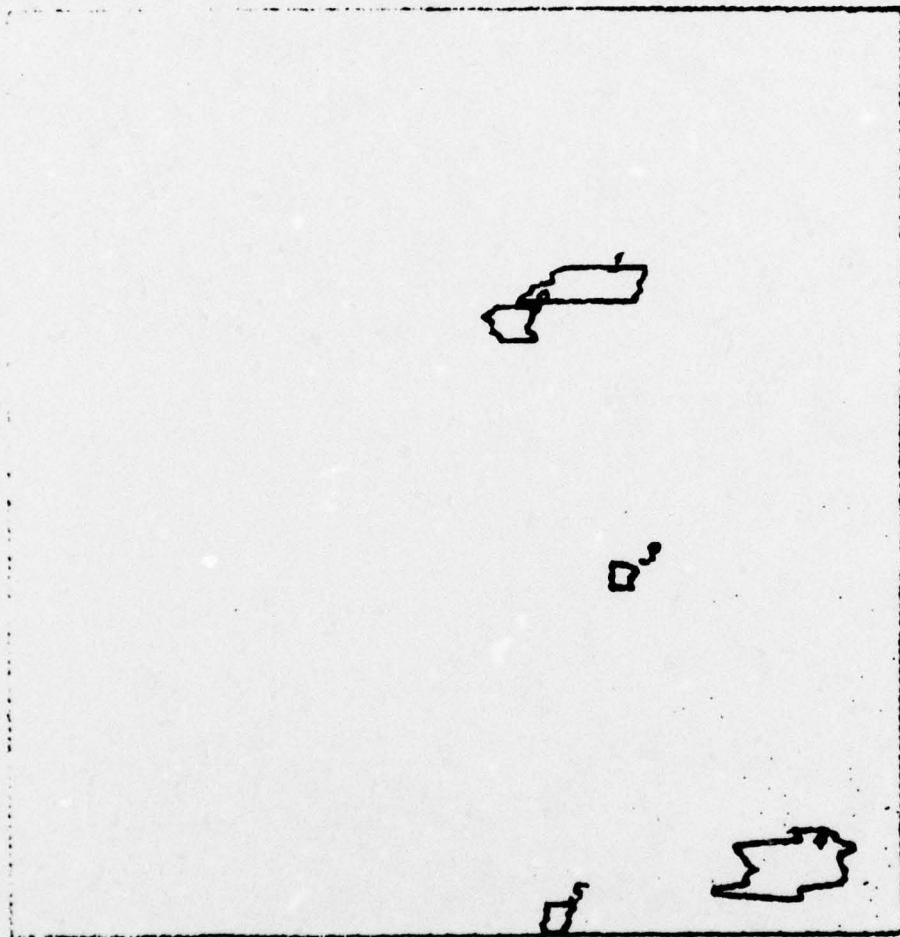


Figure 51b. Results of Candidate Object Outline Extracted by the Bin Generation Program from the Intervals in Figure 51a

higher for the targets (as is the "average contrast"). This, of course, is not a statistically significant sample of features, but serves to demonstrate the state of progress in the simulation software task.

SECTION IV

INTERFRAME ANALYSIS

The major purpose of our interframe analysis is twofold. The first purpose concerns noise reduction and the second is related to target motion. In noise related analysis the goal is to reduce the effect of noise on classifier decisions. This may be done by combining in some suitable way the noise-sensitive measurements of a detected object from several successive or near successive frames. Specifically, we wish to combine various feature values of and classifier decision made on each object in a frame. In motion related analysis the goal may be subdivided as follows: 1) detect targets from their motion, 2) track the targets, and 3) aid the object extraction mechanism by predicting the position and the local background of a target.

For both of the above purposes we need a frame to be matched with another frame. We have tried this interframe registration at a "symbolic" rather than pixel level. The objects are first extracted in the two frames independently. Some of these extracted objects are targets that may or may not be moving; the remaining are parts of the background. Using these extracted objects we first achieve a coarse alignment of the frames. This is followed by a match of the "symbols", i.e., the objects.

FRAME ALIGNMENT

When the sensor is in motion the stationary objects in the frames will have a relative displacement with respect to the frame coordinates. To find a match for an object in a frame, a search has to be made over all the objects in the other frame. The neighborhood of search can be reduced if the two frame coordinate systems are adjusted with respect to each other to correct for the sensor motion. This adjustment will not be necessary when it is known a priori that the sensor is stationary. Such frame alignment, however, may be needed for more general input.

The frame alignment is based on the assumption that most of the objects in the frame are stationary. The alignment is performed by using the locational information of each object in a frame. In general, the rectification due to sensor motion may require translation, rotation, and scale change of a frame. We assume that the sensor motion between the two frames to be matched is small enough so that translation alone may give adequate frame alignment for our purposes. For example, if the frames are successive or near successive the sensor motion may be assumed to be translation only.

Three alternative methods of frame alignment were tried. The first one, called the translation histogram method, conceptually works as follows. The difference in the coordinates of an object in one frame, say F_0 , and an object in the other frame, say F_1 , is computed. Keeping the object in F_0 fixed, this computation is repeated for every object in F_1 . This process is then repeated for all other objects in F_0 .

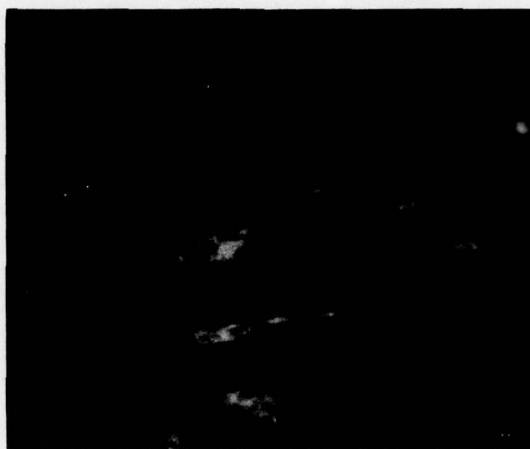
Every computed coordinate difference corresponds to a frame translation that will match an object pair in the two frames. A two-dimensional histogram of all the computed coordinate differences is made. The mode of the histogram corresponds to a frame translation that will match the largest number of object pairs in the two frames. This mode is estimated as the translation necessary for the frame alignment.

The second method is called the mean translation method. In this method the mean location of the objects in a frame is computed. This is the sample mean of the coordinates of all the objects in a frame. This computation is done for both the frames, F_0 and F_1 . The difference between the two mean locations gives the translation for the frame alignment. This method assumes that the mean locations of the objects are matched after the frame alignment. The third method, called the moment translation method, is similar to the mean translation method except that the mean location is replaced by a weighted mean location for each frame. Each object location is weighted by the area of the object. The sample mean of these weighted coordinates, normalized by the total object area in the frame, gives the area moments of the two frames. The difference between the weighted moments of the two frames is the required amount of translation.

The advantage of the mean translation method over the translation histogram method is reduction in computational complexity. However, a tradeoff in accuracy might occur as a result. A single spurious object, or noise, can affect the mean location of a frame. The moment

translation method may bring back some of the accuracy by giving conspicuous large objects more weight. If the "noise" objects are relatively small then this method may yield better accuracy than the mean translation method.

A sequence of five frames from the Krebs FLIR data base was processed by the Honeywell Augmented Target Screener Subsystem (ATSS). The input frames and the objects extracted by the ATSS are shown in Figures 52 and 53. There does not seem to be appreciable target motion in this sequence of frames. The FLIR sensor, however, is nonstationary. It may be noted that in one of the frames the ATSS failed to segment the target from the input image. Figure 54a shows a typical translation histogram. We have assumed that the frame-to-frame displacement is less than 1/8th of the frame dimensions in each of the row and column directions. Consequently, all translations greater than 1/16th or less than -1/16th of the frame dimensions have been ignored in the histogram. In order to achieve strong and robust modes, the histograms were smoothed by a 3 x 3 block filter. The filtered histogram corresponding to Figure 54a is shown in Figure 54b. The resulting frame translations are shown in Table 4. This table also shows the corresponding translations by the moment and the mean methods. F_0 is the frame being aligned (i. e., translated) with the frame F_1 . The translation is shown as (row, column) in the table.

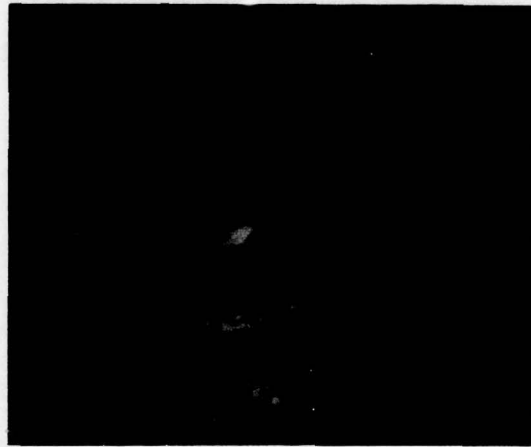


a

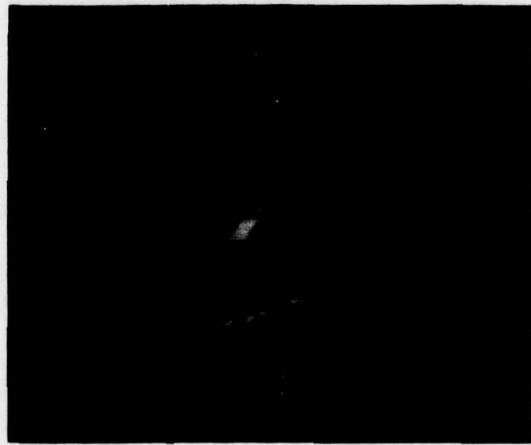


b

Figure 52. The Five Input Frames for Interframe Analysis

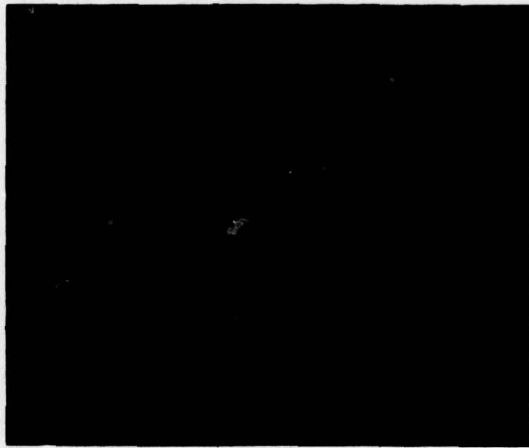


c



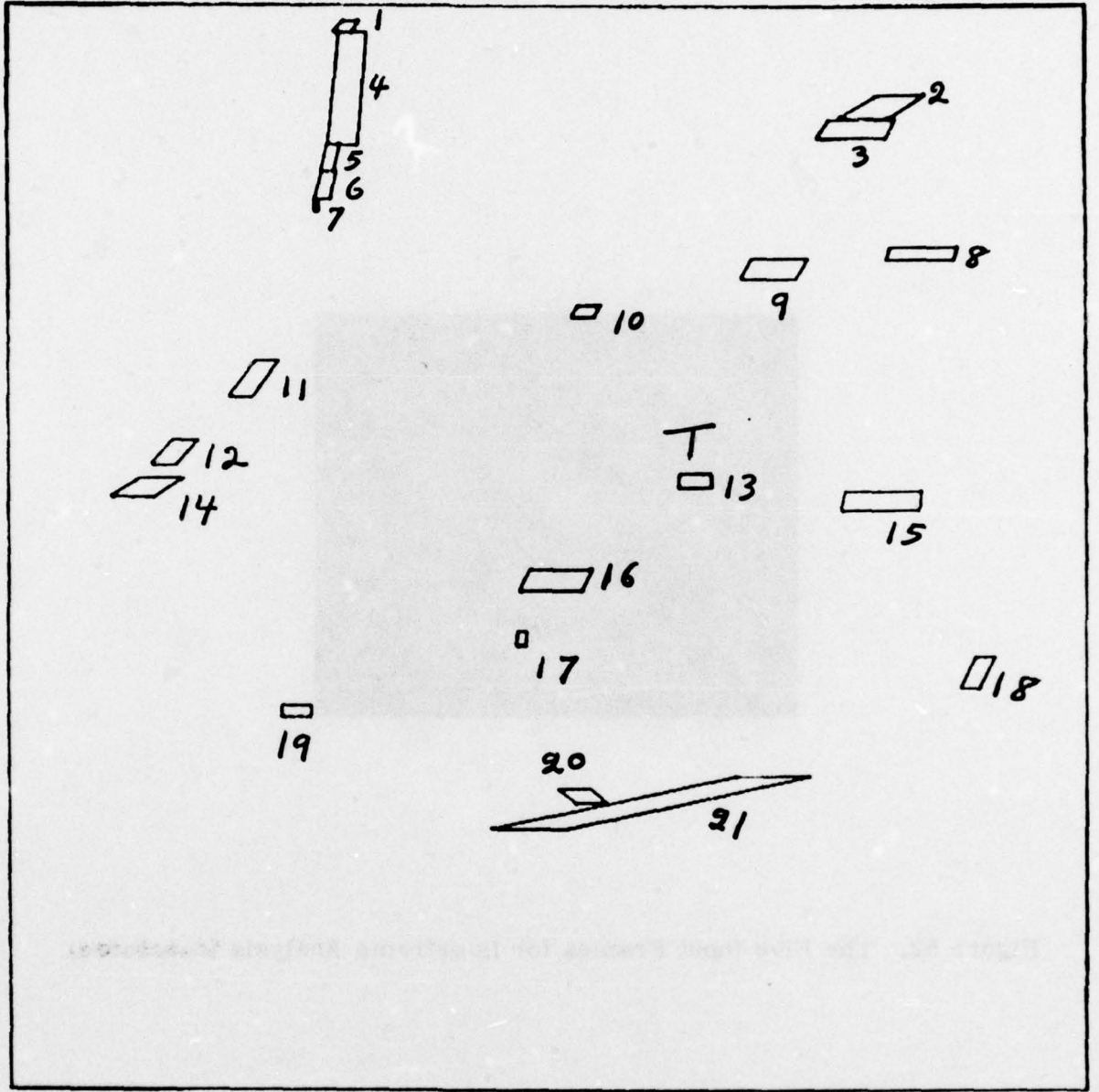
d

Figure 52. The Five Input Frames for Interframe Analysis (continued)



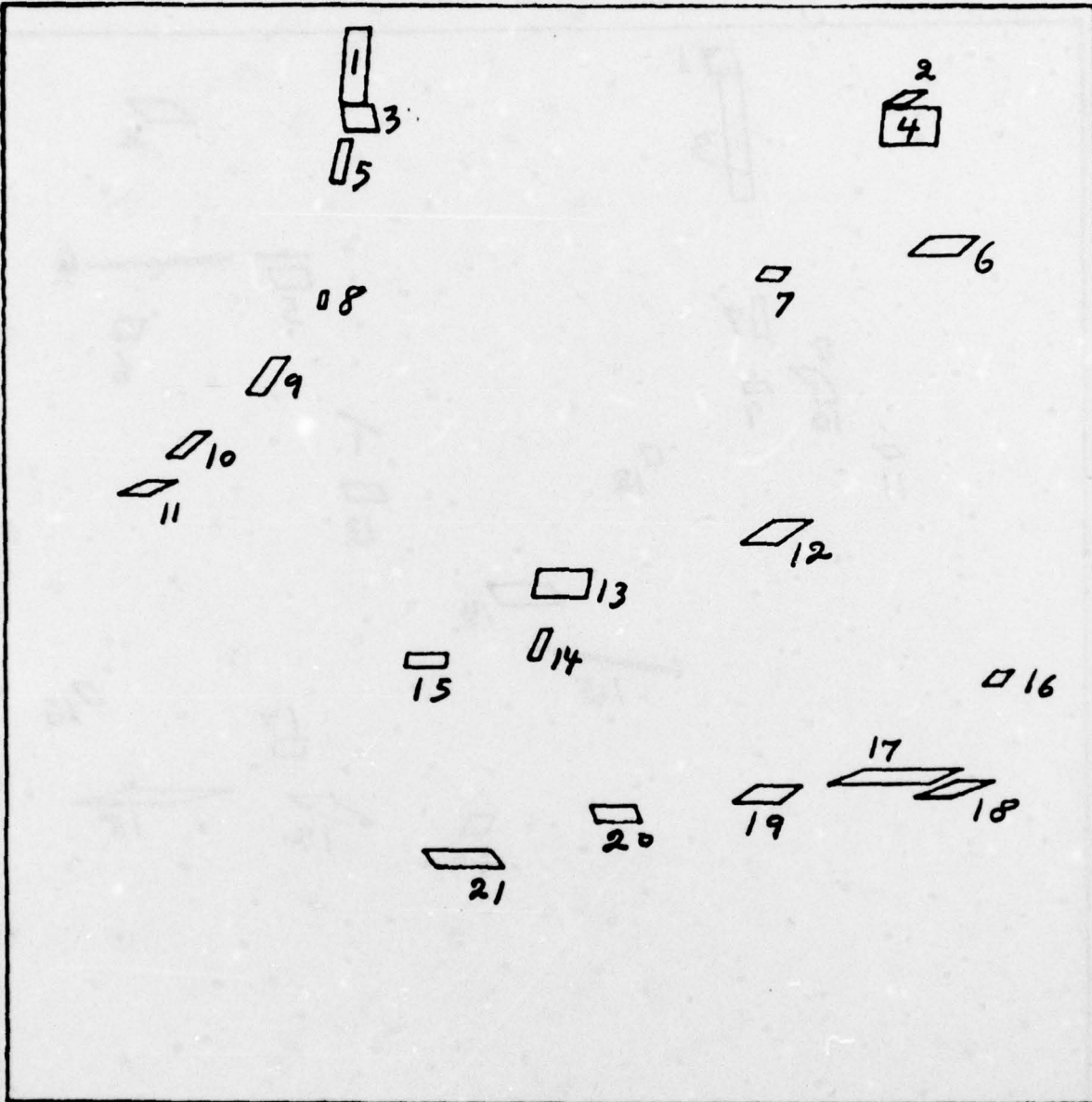
e

Figure 52. The Five Input Frames for Interframe Analysis (concluded)



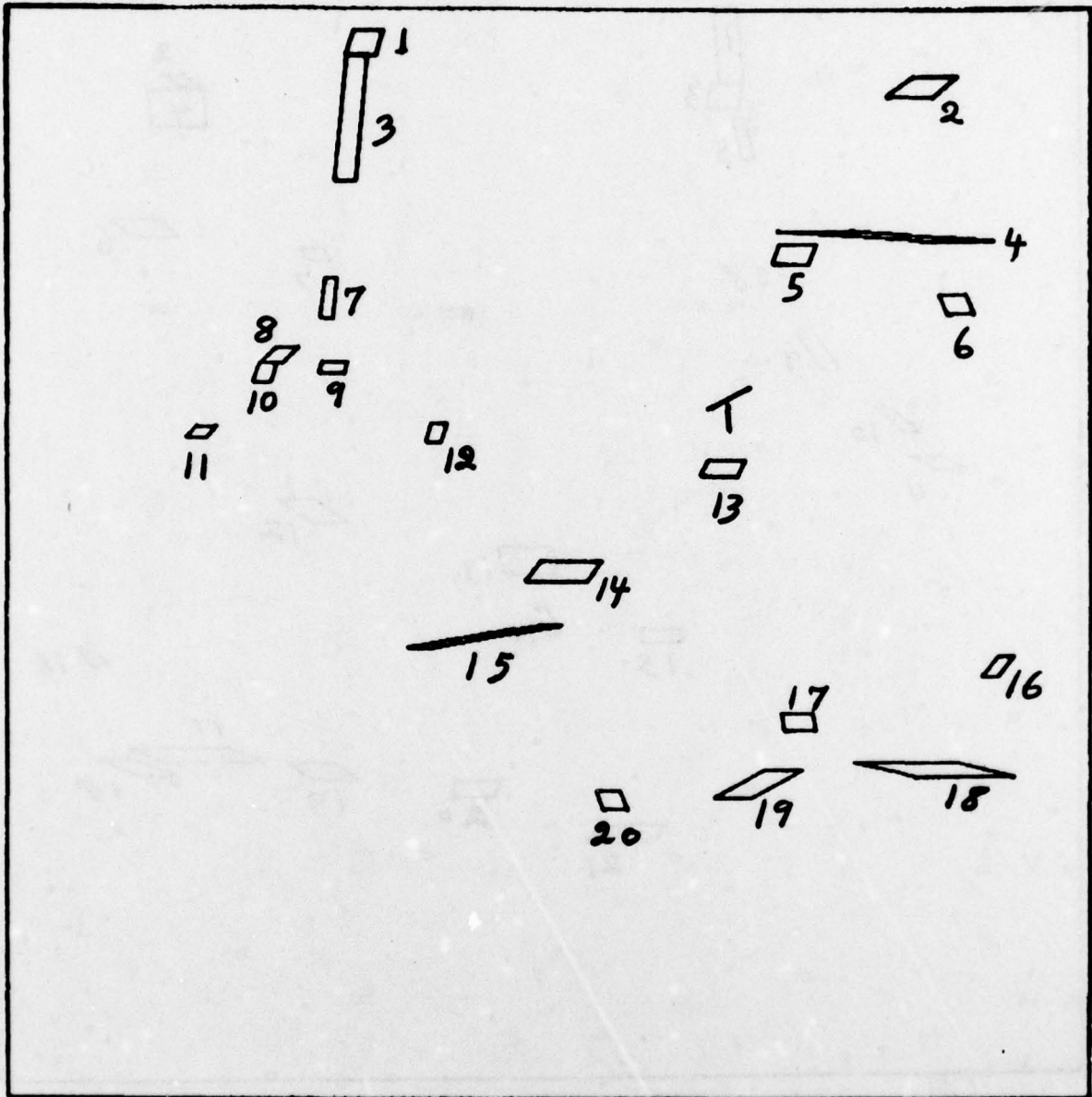
a. Frame #1

Figure 53. The Objects Extracted by ATSS from the Input Frames
(T = target)



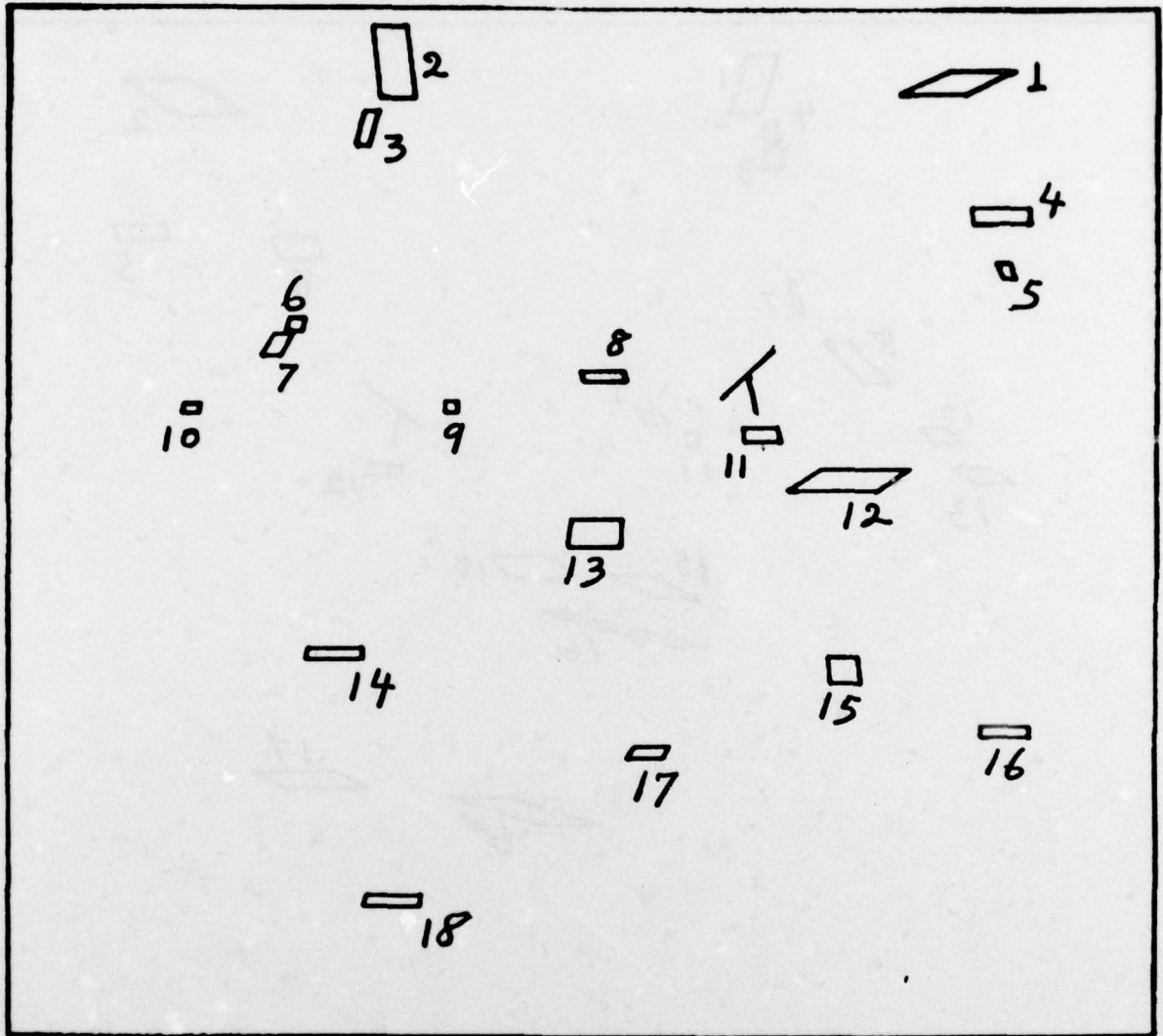
b. Frame #2

Figure 53. The Objects Extracted by ATSS from the Input Frames (T = target) (continued)



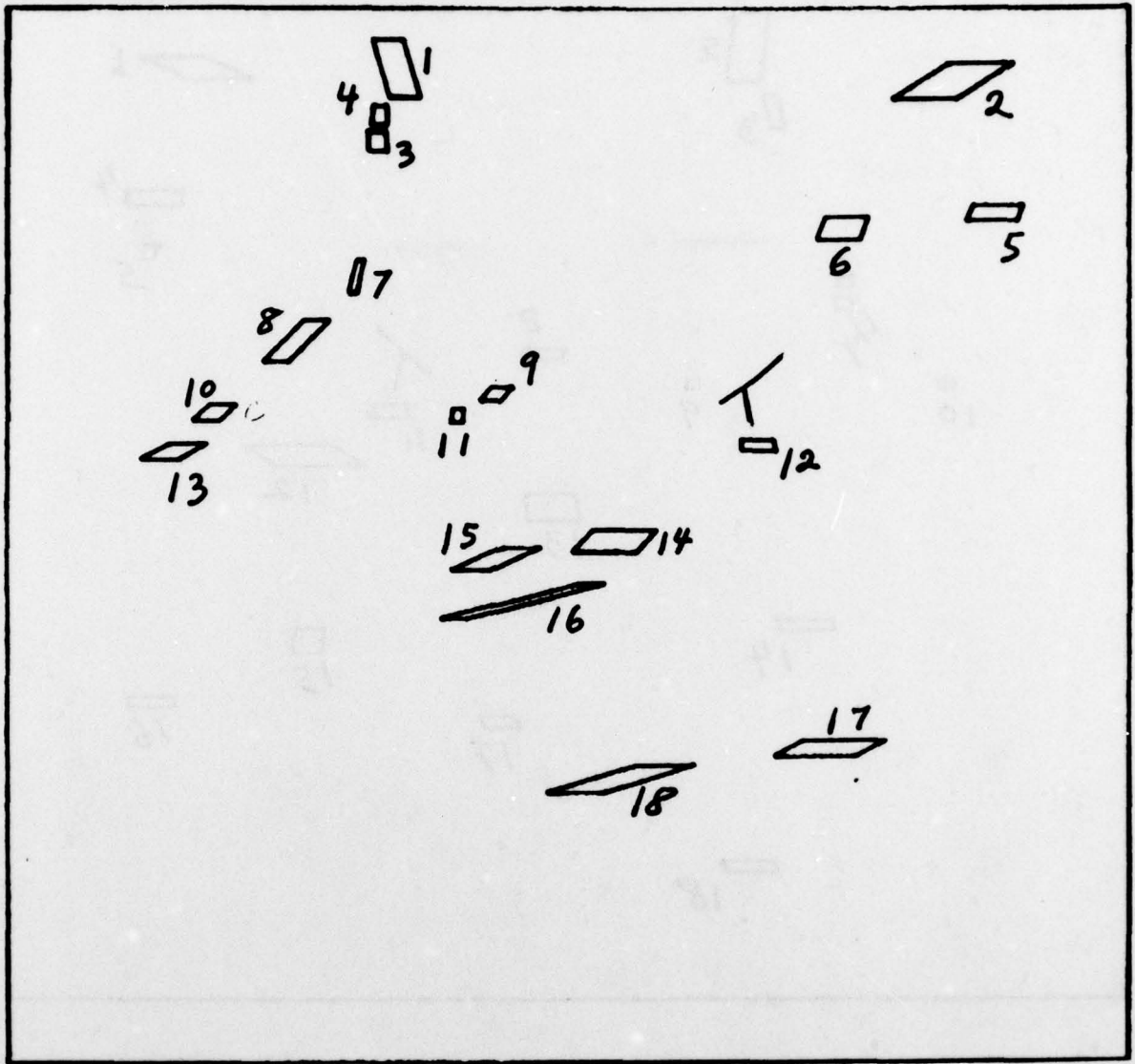
c. Frame #3

Figure 53. The Objects Extracted by ATSS from the Input Frames (T = target) (continued)



d. Frame #4

Figure 53. The Objects Extracted by ATSS from the Input Frames (T = target) (continued)



e. Frame #5

Figure 53. The Objects Extracted by ATSS from the Input Frames (T = target) (concluded)

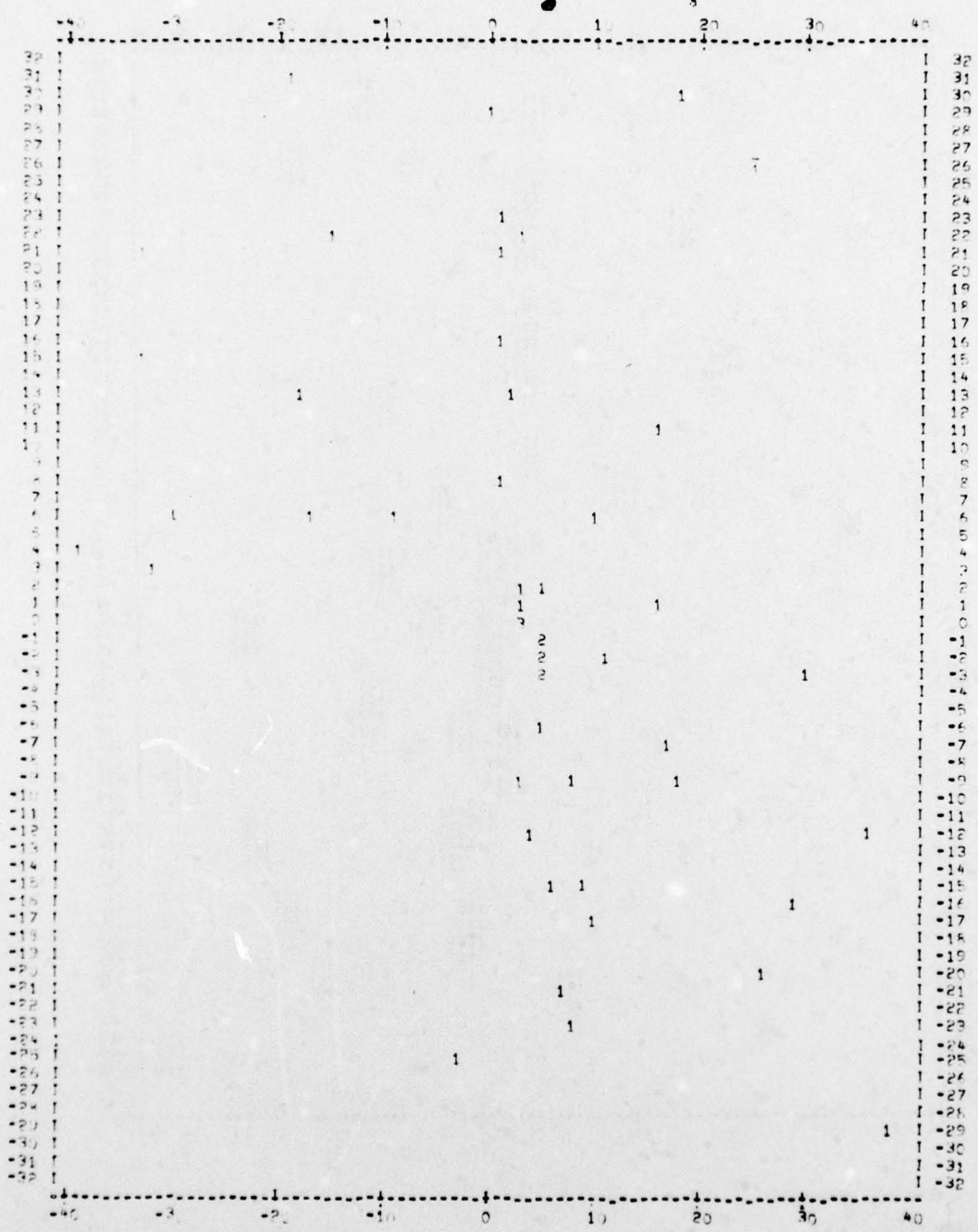
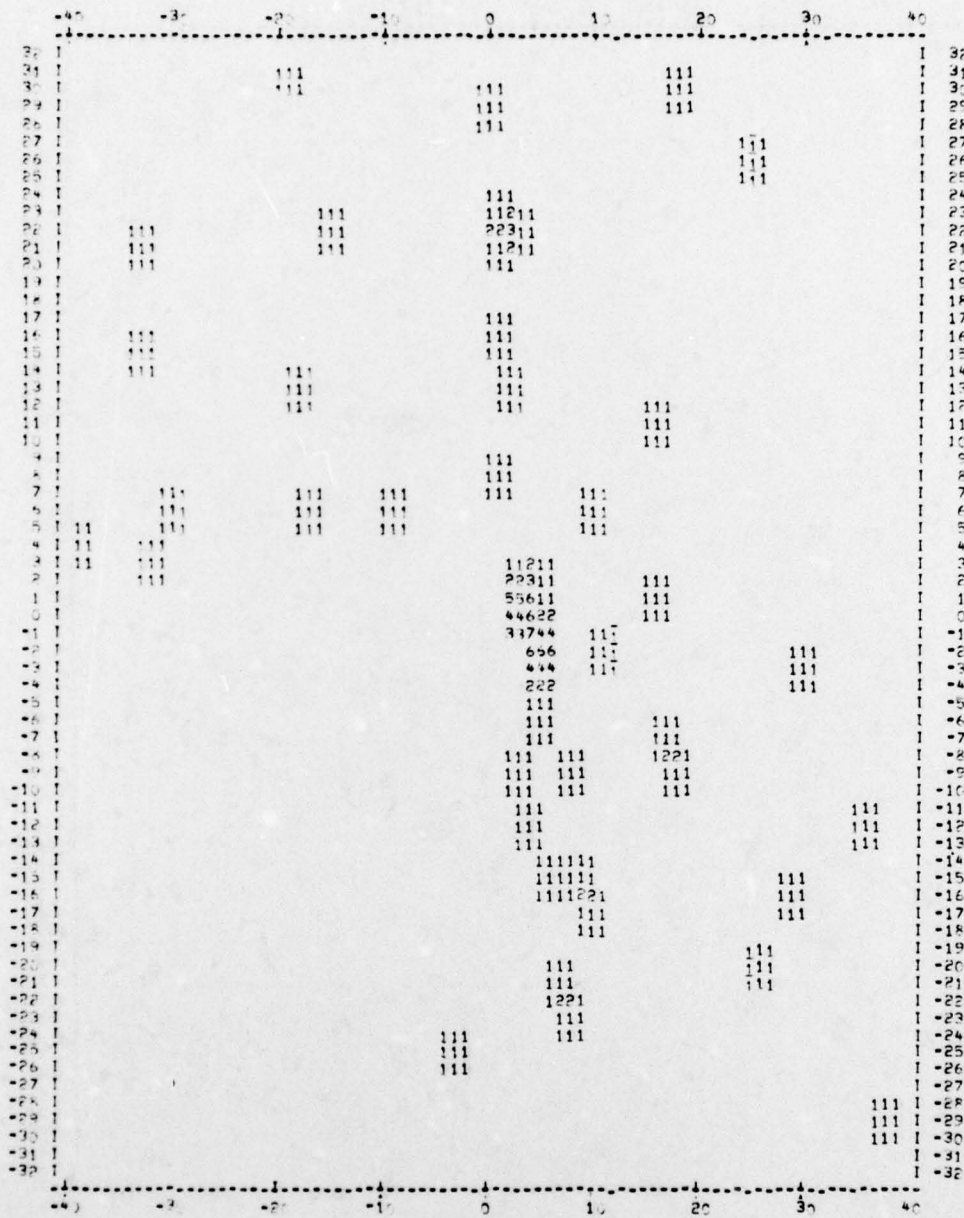


Figure 54a. Translation Histogram for Aligning Frames 1 and 2



MAX PEAK IN ABOVE HIST

IMIS(-1, 4) * 7
 SCALE * 2*001010000

Figure 54b. Filtered Translation Histogram Corresponding to Figure 54a

TABLE 4. FRAME TRANSLATIONS BY THE THREE METHODS

$F_0 - F_1$	Histogram	Moment	Mean
1 - 2	(-2, 8)	(24, 17)	(31, 28)
2 - 3	(-4, 4)	(-26, 0)	(-14, 4)
3 - 4	(-2, 2)	(5, 4)	(10, -15)
4 - 5	(10, 0)	(-9, -5)	(-30, -29)

The differences in the translations due to the three methods are substantial. The following procedure was devised to compare the accuracies of the three methods. For each frame-pair under consideration, a list of object pairs that visually appeared to match each other was made. This was treated as the "ground truth". After the frame alignments the distances between these ground truth object pairs were computed. These distances were regarded as the alignment errors. The total alignment error for each frame pair and each method is shown in Table 5. Also shown in the table in parentheses are the number of objects in the ground truth that were nearest neighbors to their matching pairs after alignments. From the table it appears that the histogram method is more accurate than the other two. Computationally, however, the histogram method is the most costly of the three methods.

TABLE 5. ALIGNMENT ERRORS AND NEAREST NEIGHBOR MATCHES

$F_0 - F_1$	Histogram	Moment	Mean
1 - 2	54 (100%)	310 (79%)	451 (78%)
2 - 3	63 (100%)	119 (100%)	71 (100%)
3 - 4	50 (100%)	86 (100%)	118 (100%)
4 - 5	101 (100%)	145 (100%)	362 (86%)

SYMBOLIC MATCHING

A major task in symbolic matching is the selection of a suitable set of attributes or features of the objects that should be used in matching. Another major task is the matching procedure itself. Typically, features that are usually used in symbolic matching are^{7,8} size, shape, color, texture, and location. The speed restriction in real time application may allow only a few and simple features to be extracted. Other considerations in extracting the features are the computational cost and the effectiveness of the features for the specific applications and image qualities in mind. In the application of noise suppression by interframe analysis, the

⁷K. Price and D. R. Reddy, "Symbolic Image Registration and Change Detection," Proceedings: Image Understanding Workshop, April 1977, pp. 28-31.

⁸B. Glish, W. Kober, and G. Swanlund, "Image Registration Experiments," Ibid, pp. 32-37.

selection of appropriate matching features is especially difficult. This is so because the features that change from frame to frame due to noise and which need to be averaged in some manner may not be used as matching features. The task of selecting a good set of features for our application needs to be investigated in detail. Here we shall demonstrate the result of using some simple features in object matching. The features are the size and the location of the object. These two features are easily obtained in the ATSS.

Our present effort is confined to the use of only one feature at a time in matching the objects. The degree of mismatch, which is termed as the cost of matching, may be of two kinds. The first, the static cost, arises due to mismatch in the features of the two objects under consideration. The second degree of mismatch, the dynamic cost, is due to mismatch or inconsistency in the interobject structural relationships.⁹ In our application the static cost is the absolute difference in the feature values of the two objects being matched. Since the objects, e.g., targets, may be moving with respect to each other, there is no constraint on the structural relationship in our application. The only interobject constraint is that no two different objects in one frame may be matched with the same object in a second frame. This will dictate the dynamic cost in our search procedure. Specifically, in matching the i th object in F_0 with the k th object in F_1 , and matching the j th ($j \neq i$) object in F_0 with the m th object in F_1 the dynamic cost = $\begin{cases} 0, & \text{if } k \neq m \\ \infty, & \text{if } k = m \end{cases}$ for all object indices i and j .

⁹M. Fischler and R. Elschlager, "The Representation and Matching of Pictorial Structures," IEEE Transactions on Computers, Vol. C-22, no. 1, January 1973, pp. 67-92.

An optimum matching procedure should minimize the total cost of matching all objects in a frame. This may be done by computing all possible static and dynamic costs and selecting the particular set of object matches that has the lowest total cost. However, the storage and the computational requirements are too high for this procedure. If we know the maximum distance a target may have moved between two frames, then we can restrict our search for a match to a neighborhood of corresponding size. In this regard the frame alignment helps save search time by cutting down the neighborhood size. Even then, the storage and computational requirements for finding the optimum matches for all objects in the frame may be very high. If there are N objects to be matched in a frame and each object has at least K objects in its neighborhood for searching, then there are at least N^K combinations to consider, each combination having N static costs and $\frac{N(N-1)}{2}$ dynamic costs to compute. The Linear Embedding Algorithm of Fischler and Elschlager⁹ is aimed at cutting down such a computational requirement by trading it off with the global optimality of matching. In particular, the method may fail to find the globally optimum match if the objects with low indices in F_0 incur a high static cost when matched with their optimal object matches in F_1 . We have adopted a matching procedure with similar suboptimality but one that is computationally more suited for our application. The procedure is independent of object indices but depends on the relative magnitudes of the static costs.

Let the i th object in F_0 have K_i objects in F_1 in its neighborhood of search. We shall call these K_i object indices in F_1 the possible K_i "labels" of the i th object. In our search procedure the static costs for

all possible labels are computed for each of N objects in F_0 . In total there will be K_T different static costs, where

$$K_T = \sum_{i=1}^N K_i .$$

Each of these K_T costs corresponds to an object-label match. We arrange these K_T costs in increasing order. In case of a tie the costs are arranged in increasing order of the object index i . We shall accept at the most N of these static costs and corresponding object-label pairs. In our procedure, the lowest of the K_T costs is always accepted. We then proceed to the next higher cost. If the label corresponding to this cost has already been taken by previously accepted object-label pairs, then we discard this object-label (infinite dynamic cost). If, instead, the object corresponding to this cost has already been taken, then this object-label pair has a higher static cost. Hence we discard this object-label pair and proceed to the next higher cost. If certain cost did not get discarded by the above two methods then the corresponding object-label pair is accepted as the next matching object-label pair. This procedure continues until all the K_T static costs are exhausted. This algorithm will not give the globally optimum match if the static cost corresponding to an optimum object-label pair is higher than that of another object-label pair having the same label. Consider, for example, two objects, A and B , being matched with two labels, a and b , with the following static costs:

	a	b
A	3	7
B	5	11

The object-label pairs arranged in increasing order of static cost are: Aa, Ba, Ab, and Bb. The pairs that will get accepted are Aa and Bb, even though the optimum pairs are Ab and Ba. It is possible that several iterations of a similar procedure in some suitable manner, e.g., by relaxation labelling,¹⁰ will asymptotically yield the global optimum match.

An accelerated version of this ordered static cost method has been implemented in the XDS9300 computer. The previously mentioned sequence of Krebs frames was aligned according to pairs using the translation histogram method. The result was processed by the above matching algorithm using location and size as matching features. Table 6 shows the result with the location feature, and Table 7 shows the result with the size feature. A label of "0" implies no match, and the number following the # sign is the frame number in the tables. These interim results appear to be excellent. We are currently also studying the use of other features.

Further study needs to be done in determining appropriate features for matching objects. The effectiveness of the matching procedure

¹⁰ A. Rosenfeld, R. Hummel, and S.W. Zucker, "Scene Labelling by Relaxation Operations," IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-6, 1976, pp. 420-433.

TABLE 6. SYMBOLIC MATCHING BY LOCATION

#5 F ₀	#4 LABEL	#4 F ₀	#3 LABEL	#3 F ₀	#2 LABEL	#2 F ₀	#1 LABEL
1	2	1	2	1	1	1	0
2	1	2	1	2	2	2	2
3	0	3	3	3	5	3	4
4	3	4	4	4	6	4	3
5	4	5	6	5	7	5	5
6	0	6	8	6	0	6	8
7	0	7	10	7	8	7	9
8	7	8	0	8	0	8	0
9	0	9	12	9	0	9	11
10	10	10	11	10	9	10	12
11	9	11	13	11	10	11	14
12	11	12	0	12	0	12	0
13	0	13	14	13	0	13	16
14	13	14	0	14	13	14	17
15	0	15	17	15	15	15	0
16	0	16	18	16	16	16	18
17	0	17	0	17	0	17	0
18	0	18	0	18	18	18	0
				19	19	19	0
				20	20	20	0
					21	21	0

TABLE 7. SYMBOLIC MATCHING BY SIZE

#5 F ₀	#4 LABEL	#4 F ₀	#3 LABEL	#3 F ₀	#2 LABEL	#2 F ₀	#1 LABEL
1	2	1	2	1	1	1	4
2	5	2	1	2	2	2	3
3	0	3	3	3	3	3	6
4	3	4	4	4	6	4	2
5	4	5	6	5	7	5	5
6	1	6	8	6	4	6	8
7	6	7	10	7	0	7	9
8	0	8	0	8	9	8	7
9	8	9	12	9	5	9	11
10	10	10	11	10	0	10	12
11	9	11	13	11	10	11	14
12	11	12	5	12	8	12	13
13	7	13	14	13	12	13	16
14	13	14	15	14	13	14	17
15	0	15	17	15	15	15	19
16	14	16	18	16	16	16	0
17	15	17	0	17	17	17	15
18	17	18	0	18	18	18	18
				19	19	19	20
				20	20	20	0
					21	21	21

also needs to be examined and evaluated in detail with data typical for our application. We are also studying methods of speeding up the translation histogram methods.

SECTION V

PLANS FOR NEXT REPORTING PERIOD

During the next quarter of the program we will complete the five month Phase I design study. The tasks which will be completed during that period include: Task 1.1, MODFLIR/PATS interface analysis; Task 1.2, algorithm selection; Task 1.3, system modeling; and Task 1.4, design review.

The final algorithm selection and performance evaluation will be made using the available thermal imagery data set described in Section III. It should be noted that the available imagery essentially contains only two (tanks, and APCs) of the five target classes for which the automatic target screening function is required. When imagery for the other three classes becomes available, the target screener can be trained for the additional classes. However, the final selection of algorithms will essentially have been made on the presently available data.

We also plan to complete during the next quarter a breadboard model of the global gain and brightness control circuit and its interface to the MODFLIR, and a breadboard model of the DC restoration circuit. With these breadboards, and that of the already completed local area gain/brightness control, the image enhancement subsystem will be completed except for repackaging into the final configuration.