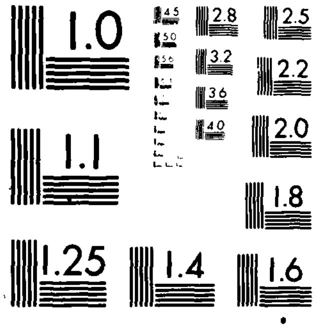
AD-A108 767 GOODYEAR AEROSPACE CORP AKRON OH F/G 9/2
PIPELINE PROCESSOR CONFIGURATION STUDY.(U)
SEP 81 J L POTTER, J R WISEMAN F30602-80-C-0163
UNCLASSIFIED RADC-TR-81-238 NL

1 OF 1
AD
A108767

END
DATE
FILMED
2-82
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

II (12)

RADC-TR-81-238
Final Technical Report
September 1981

PIPELINE PROCESSOR CONFIGURATION STUDY

Goodyear Aerospace Corporation

Dr. J. L. Potter
J. R. Wiseman

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

This report has been reviewed by the RADC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-81-238 has been reviewed and is approved for publication.

APPROVED:

ANDREW R. PIRICH
Project Engineer

APPROVED:

JOHN N. ENTZMINGER
Technical Director
Intelligence & Reconnaissance Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

SUBJECT TO EXPORT CONTROL LAWS

This document contains information for manufacturing or using munitions of war. Export of the information contained herein, or release to foreign nationals within the United States, without first obtaining an export license, is a violation of the International Traffic in Arms Regulations. Such violation is subject to a penalty of up to 2 years imprisonment and a fine of $100,000 under 22 U.S.C 2778.

Include this notice with any reproduced portion of this document.

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (IRRE) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C^3I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER RADC-TR-81-238	2. GOVT ACCESSION NO. AD-A108 [illegible]	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) PIPELINE PROCESSOR CONFIGURATION STUDY		5. TYPE OF REPORT & PERIOD COVERED Final Technical Report 27 May 80 - 27 May 81
		6. PERFORMING ORG. REPORT NUMBER N/A
7. AUTHOR(s) Dr. J. L. Potter J. R. Wiseman		8. CONTRACT OR GRANT NUMBER(s) F30602-80-C-0163
9. PERFORMING ORGANIZATION NAME AND ADDRESS Goodyear Aerospace Corporation 1210 Massillon Road Akron OH 44315		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62702F 45941840
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (RADC/IRRE) Griffiss AFB NY 13441		12. REPORT DATE September 1981
		13. NUMBER OF PAGES 31
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Same		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: Andrew R. Pirich (RADC/IRRE)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Real-Time Image Processor
Multisensor Image Processor
Target Screener
Common Target Screener

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

The pipeline processor configuration study (PPCS) resulted in a high-performance processor system design for automatic screening and image exploitation systems. Key to the success of this design effort was combining the pipelining and parallel processing techniques to overcome typical system problems such as data throughput (I/O) and algorithm flexibility (programmability). The design combines subsystem components that are either already in existence or are currently under development

UNCLASSIFIED

(Cont'd)

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Item 20 (Cont'd)

into a system that contains many features not found in any other exploitation system described in the literature. For example, not only can the system process data at over 150 Mbits per second, but it can also handle multiple-sensor data from different sensor types in an interleaved (time shared) manner at these rates. Thus, much lower exploitation system development and life cycle costs are possible with this one common auto screening system. Since it is 100 percent programmable, it can also accommodate new algorithms and approaches to target recognition such as multiple (group) target analysis.

The results of an image exploitation algorithm analysis and a hardware technology analysis are reported as well as the PPCS system design and a typical mode of automatic screening operation.

TABLE OF CONTENTS

Page

Title

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

The pipeline processing configuration design (PPCS) described in this final report was developed after careful analysis of the technical, functional, and economic needs of the image exploitation environment. The design provides the most cost-effective approach to a real-time, multisensor, image exploitation system.

The key feature of the system is its extreme flexibility, which is achieved by a 100 percent programmable image processor embedded in a system that supports data transfer rates of 1280 Mbits/second, a modular growth capability, and large-volume image storage capacity. Due to the system's flexibility, it will have a long life cycle since it is capable of adapting to the rapid changes in image exploitation, such as higher sensor data rates, new sensor types, new algorithms, and new modes of operation.

For example, as new sensors are developed, the PPCS system can support a number of multisensor processing modes. The PPCS system can be physically time-shared with one team of users at one time and another team at a later time. It can be time-shared on a frame-by-frame basis so that two (or more) different types of imagery can be processed simultaneously, each image type going to its own CRT display(s). Using the same target screener system for multiple sensors would dramatically reduce exploitation systems' life cycle cost.

The PPCS system also can be used for synergistic multisensor and sensor fusion processing; this would be particularly valuable when multiple sensors are aboard the same platform. Synergistic multisensor processing could make use of artificial intelligence techniques to allow combining the information content of multiple sensors on a real-time frame-by-frame basis to improve target detection and recognition. Such techniques would be useful for camouflaged targets, deceptive targets, and even recognition of clusters of targets.

New modes of operation such as real-time interactive image processing would help improve target recognition and can be supported by the PPCS system because of its high data transfer rate and unique image windowing capability. The PPCS short-term image memory will allow the system to directly address any 128 x 128 subregion of the imagery contained in it (this memory can be sized to hold up to 64 Mbytes of data or 8000 x 8000 x 8 bit pixels). This capability allows the system to concentrate its computing power on the subregion of its choice in a real-time mode. Or the user could direct the system in an interactive mode. This would be particularly helpful where the automatic target screening algorithms assign two labels to the same target with approximately the same confidence level. The user could interactively direct additional recognition algorithms.

The flexibility of selectively applying the PPCS computing power to important subimages allows the PPCS subsystem to support continuous real-time processing of FLIR and other TV-formatted imagery. The continuous processing capability adds a new dimension of time-varying data to aid in the classification of difficult targets by artificial intelligence and other sophisticated techniques.

The PPCS subsystem was designed to process imagery at rates of 100 Mbits/second and above and be flexible enough to allow it to be a common element of all present and future real-time IR, radar, and optical image exploitation systems.

BACKGROUND

The nature of the targets (mobile, fleeting, time sensitive) against which the reconnaissance exploitation system of the 1980's and beyond are to operate requires real-time or near real-time processing rates in order to minimize the time required from initial sensor acquisition to subsequent strike. Data from a variety of current sensor systems and many of the sensor systems under development are digital. This digital aspect of these sensors can be used to dramatically reduce the life cycle costs of these systems by using a common computer system capable of real-time digital image processing for the image exploitation functions for all the digital sensor system types.

The sensor types with which such a common design must operate vary in data rates and format. For example, the PAVE TRACK system uses a FLIR sensor. FLIR sensors may produce imagery at either the 525-line or 875-line TV rate (30 frames per second). The AN/AAD-5 sensor is capable of producing up to 10,000 six-bit pixels per line. Some of the electro-optical sensors currently being developed can produce 17,500 pixels per line, 1000 lines per second at four bits per pixel.

Since the sensors of the future will produce even more imagery, the PPCS system as a minimum must be capable of processing data at 100 Mbits/second. Moreover, the system must be capable of using auxiliary information such as input from a moving target indicator; sensor position information such as sensor platform altitude and look angle; and target location cueing data or other intelligence information.

The basic scenario under which the PPCS system analysis was performed assumed a directed flight path (cued mode) with 2-1/2 minutes of 525-line TV-rate FLIR imagery. The imagery would contain 40 targets and consist of groups of tanks, trucks, jeeps, etc. The classification goals were 90 percent of the targets detected, 80 percent of the targets correctly classified, and not more than 10 percent of the targets incorrectly classified. Target density was assumed to average 20 targets per 1000 square miles, with a peak of 100 targets per 1000 square miles. The processing was to be done in real-time (2-1/2 minutes from start of imagery input).

Since the specific automatic detection and identification algorithms depend heavily on the sensor type, most of the algorithms studied were for FLIR imagery; however, optical and radar processing algorithms also were analyzed.

ALGORITHM ANALYSIS

Thirty-nine algorithms for target detection and classification were reviewed (all algorithms reviewed are included in the bibliography). The algorithms were analyzed and classified along two different metrics. One measures the computational complexity of the algorithms from logical to fixed-point mathematical to floating-point mathematical. Logical algorithms require relatively little mathematical operations and consist mostly of comparisons and logical operations (AND, OR, etc.). Most region-growing algorithms are logical. Fixed-point mathematical algorithms require a moderate amount of straightforward arithmetic. For example, convolution requires nine multiplies and adds per pixel. This is sufficiently small so that fixed-point arithmetic is more than adequate for such operations. Floating-point mathematic routines require such extensive calculations that fixed-point arithmetic does not have sufficient precision to handle the dynamic range of the variables. Most transformations such as an FFT fall into this category.

The second metric by which the algorithms were classified was suggested by Hunt (1976). It delineates the nature of the memory accesses of the algorithms. The three classes are pixel mappings, areal mappings, and transformations. Gray-scale remapping is an example of pixel mapping. Convolution is an example of areal mapping, and FFT is an example of transformation.

Pixel mappings are best executed in a pipeline processor since access to the memory associated with only one pixel is needed at any one time. Areal mappings can be handled by pipeline processors if the kernel size is small enough (say, 3 x 3). However, large kernel sizes are difficult for pipeline architectures since all data for each row (or column) of imagery in the kernel must be stored in the pipe or repeatedly passed through it. When windows become large (say, over 9 x 9), as they do for sophisticated target classification, then the memory/multiple-pass aspect of pipeline processors requires special interconnections between stages of the pipe, which reduces the design flexibility and commonality.

Finally, the transformation class is basically similar to the areal class except that the interrelationship of neighboring pixels is much more complex with second transformations than for areal algorithms and therefore is that much more difficult for pipeline processors to handle. However, the floating-point aspect of this class is well suited to pipeline techniques.

The results of these classifications are shown in Table 1. They indicate that most of the algorithms fall into the fixed-point arithmetic classes in the one metric and into the areal class in the other metric.

TABLE 1. ALGORITHM TYPES

Logical	Fixed point	Floating point
6	23	12
Pixel mappings	Areal mappings	Transformations
9	28	4

Most of the algorithms analyzed were for FLIR imagery; however, the general conclusions below apply to all types of image processing. First, there are effective sets of algorithms for the autoscreening function. However, different sensors and even different sensor conditions (day/night FLIR imageing) require different sets of algorithms. The more sophisticated and more successful algorithms used larger kernels (looked at larger areas) and took more computation time. Certain algorithms, especially those used for target recognition, used techniques outside of "traditional" image processing. That is, they are not pixel crunching. Algorithms that model the targets and the sensor's perception of those targets are an example. These are algebraically oriented and require considerable amounts of computer power.

In summary, the PPCS system must be efficient for logical operation, fixed-point operation, and areal operation. Yet, it must be flexible enough to allow concentration of computing power when needed and it must be completely programmable to allow it to be used for different sensors and under different conditions.

TECHNOLOGY SURVEY

The technology survey addressed not only component (VLSI and VHSIC) technology but also architecture and peripheral technology. A well-designed processor system requires that all areas be addressed. The component survey concentrated on expected advancements in IC technology and the anticipated impact of those advancements.

Considerable attention is being paid to speed up chips for military application. The VHSIC program is expected to produce processing rates of 10^{12} to 10^{13} gate Hz in the 1985 time frame. Josephson circuits promise cycle times of 10^{-16} seconds but require very low temperatures. Such speeds invite on-chip algorithms such as convolution. However, there are problems. One is the major design effort required to develop specialized chips. The approach to this problem is the macro-cell design technique, where a library of predesigned (and debugged) circuit modules is used to build the desired chip. However, this approach slows down the chip speed due to the noncustom macro-cell designs. Another problem for VHSIC is that, with very small line widths (for speed), optical techniques for mask generation can no longer be used, but more expensive and time-consuming electron beam processors must be used. Bell Laboratories has recently announced success with an X-ray technique that may alleviate this problem.

In summary, over the past 20 years, a 16,000-fold increase in cost performance has been experienced.* It would appear that the trend will basically continue. However, the concept that this will allow vast new approaches to image processing does not directly follow because the economics of general-purpose versus special-purpose design will not change dramatically. In fact, any change is apt to go against specialized design since, in the foreseeable future, there will be great demand on existing resources; the resources will follow the marketplace, which will be general-purpose chips. Consequently, custom algorithms on a chip will be too expensive compared to alternate approaches except in applications where cost is of no concern.

At the Non-Conventional Computers for Image Processing Workshop, held in Madison, Wisconsin, May 27-30, 1981, four basic architecture classes were discussed: sequential, pipeline, SIMD,** and MIMD.** For real-time image exploitation, sequential computers are too slow and networks for more than six or eight processors (MIMD architecture) cannot be economically built since the operating system overhead added to the system when a new processor is added is almost as large as the added computing power. Thus, only SIMD and pipeline processors are viable candidates for real-time image processing.

SIMD processors achieve their speed from parallel execution of sequential instructions on large amounts of data. In a first-order approximation, it is similar to a hardware loop where all iterations of the loop are processed in parallel. Sixteen thousand processors mean that the equivalent of 16,000 loop iterations is executed at once. Since SIMD processors have a sequential instruction stream, they are basically as easy to program as conventional computers.

---

*Block, E.; Galage, O., "Component Progress: Its Effect on High-Speed Computer Architecture and Machine Organizations," IEEE Computer, 1978, p. 64-76.

**SIMD = single-instruction, multiple-data parallelism; MIMD = multiple-instruction, multiple-data parallelism.

Pipeline processors achieve their speed from stacking processors together so that a memory access is required for only the first and last processors - in effect overlapping or pipelining functional operations.

The tradeoffs between SIMD and pipeline processors are (1) pipeline processors can be designed to be faster for specific predefined algorithms, where SIMD processors are more general and are faster over a broad spectrum of algorithm types; (2) pipeline processors can be built more cheaply for their designed algorithm but cannot be easily modified for new algorithms, where SIMD processors are more expensive but are easily reprogrammable to handle any new algorithm; (3) pipeline processors are efficient with algorithms that do not require large window sizes (7 x 7 or less), while SIMD processors are equally efficient regardless of window size; and (4) pipeline processors are efficient only when the data to be processed flows in a constant stream, while SIMD processors can process randomly accessed data as efficiently as piped data.

In general then, pipeline processors are less expensive and less flexible than SIMD processors. Both architectures can supply sufficient computational throughput. It is estimated that SIMD processors will be capable of 10^6 million operations per second by 1990.*

Current CPU architectures can process data faster than peripherals can deliver it. Therefore, a review of peripheral developments probably is the most important aspect of the survey. Currently, high-density tapes, bubbles and CCD memories, and optical disks hold the best promise for suitable peripherals for real-time image processing. The high-density tapes suffer from the lack of random access while bubble and CCD memories currently are too slow to provide sufficient throughput. Fortunately, optical disks look promising.

Optical disks are being developed that store from 5 x 10^{10} to 1 x 10^{11} bits per disk side. Transfer rates of up to 320 Mbits per second are possible with multiple channels. An optical disk can have as many read/write heads (channels) as needed; they may be switched from read to write and back under software control. Seek time is somewhat slow at one-half second. Being a write-once device is a drawback. However, at the current time, optical disks appear to be the best choice for an image storage medium.

SYSTEM DESIGN

Criteria

The intent of the PPCS subsystem design was to accommodate multiple sensors, provide real-time image screening, and yet be extremely flexible. By accommodating multiple sensors, PPCS is a common image screener capable of processing FLIR, DLIR , radar, and optical imagery. A single system that is capable of processing all types of imagery greatly reduces the cost of sensor exploitation systems through common hardware.

Real-time processing of imagery is required by the nature of the targets the system is to detect. Finding mobile, fleeting, and time-sensitive targets in imagery hours or days old is of little practical value. Moreover, the system should be designed with sufficient reserve capacity to handle new sensors currently under development with much larger data rates in at least real-time (less than five minutes).

---

*Sos, J.; Cennell, E.; McCaleb, F., "Ground Data Processing Technology," in NASA Space Systems Technology Model, Volume II, Space Technology Trends and Forecasts, May 1980, p. 4-101 to 4-108.

The need for flexibility in the PPCS subsystem cannot be overemphasized. This flexibility is needed on two levels. First, flexibility in adapting to new algorithms is required if new sensors and new techniques (such as artificial intelligence for processing camouflaged, deceptive, and multiple targets) are to be accommodated. Basically, this kind of flexibility requires 100 percent programmability of the subsystem.

Second, the new sensors and new techniques and the desire to process multiple sensors simultaneously either for throughput or for sensor fusion effects requires the flexibility to expand the processing power of the system at a later time. The expansion capability needs to be in all three areas of increased computing power, increased storage capacity, and increased bus capacity if increased processing throughput is to be achieved.

All of the above criteria were kept in mind during the design phases of the PPCS subsystem.

System Architecture

The PPCS subsystem design combines the best aspects of pipeline and SIMD processing to achieve a real-time practical, flexible image exploitation target screener. Pipeline techniques are employed on two different organization levels to speed throughput while the SIMD approach is taken in the image processing portion to assure 100 percent flexibility.

On the functional level, pipeline techniques are used to overlap separate tasks such as I/O, control, and image processing. For example, the data flow processor will be inputting imagery from the next frame while it is outputting imagery from the previous frame at the same time it is supplying data from the current frame to the image processing unit.

The image processing unit consists of over 16,000 processors connected in a two-dimensional array (128 x 128). Each processor is functionally complete and is under the control of the program in the image processor control unit. Each processor can have up to 8K bits of data memory. The programmable commands to the image processor control unit are pipelined from the main control unit, which is a conventional sequential computer. Therefore, the 16,000 processors are 100 percent programmable, thus providing complete flexibility. Figure 1 shows the PPCS subsystem design.

All data flow into and out of the PPCS subsystem is under control of the data flow control processor. It controls (1) the input of the sensor data from the sensor buffers to the short-term image store, (2) the movement of data between the short-term image store and the image processor unit, (3) the movement of data between the short-term image store and the medium-term image store, and (4) the movement of data between the short-term and/or medium-term image store and the rest of the image exploitation system.

The image processor control is a programmable unit with 8K words of 64 bits* dedicated to supplying instructions to the 16,000 processors in the image processor unit. It, as well as the data flow control, is in turn directed by the PPCS main control processor.

---

*All PPCS memories are easily expandable.

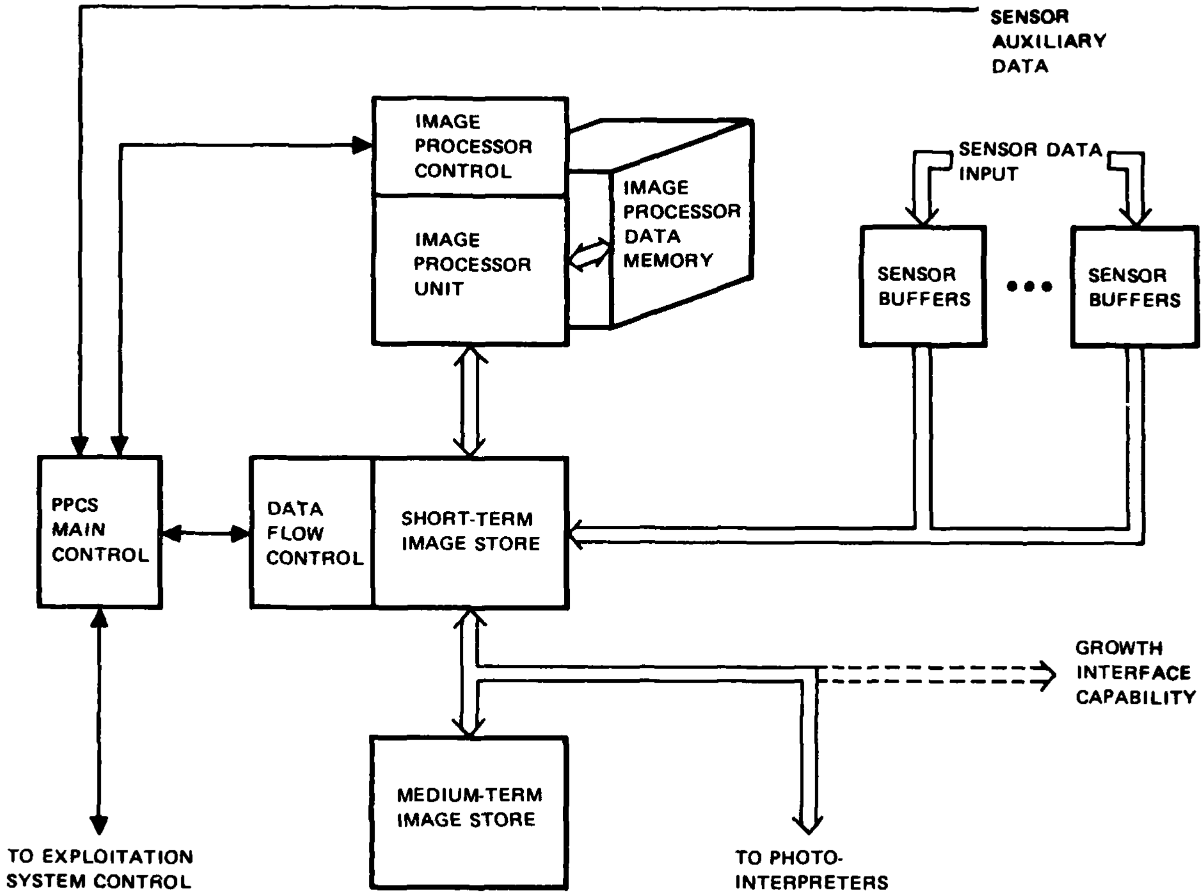


Figure 1 - Block Diagram of PPCS Subsystem Design

Thus, the main control processor with 32K words of 16 bits per word not only directs the flow of data in the PPCS subsystem but also determines what function the image processor unit is to execute on the data and how the processing power resource is to be utilized. For this reason, the auxiliary sensor data and image exploitation control data are input directly to the PPCS main control.

For example, if the image exploitation system has been cued to look for specific target types at specific locations, the main control would direct the image processor to execute only those algorithms best suited for detecting and recognizing the cued targets, direct the data flow control to supply to the image processor only that imagery within the boundaries of the cued region, and direct the data flow control to save the remaining imagery in the medium-term image store for potential later processing.

The image storage function is divided into three levels: short, medium, and long. The short-term store is a solid-state memory of up to 64 Mbytes whose function is to provide a fast buffering capability to keep the PPCS subsystem operating at maximum efficiency. It has a high bandwidth capability (1280 Mbits/second) and zero seek time. The medium-term store is intended to provide storage for the duration of the image processing session (from five minutes to one-half day depending on the utilization of the image exploitation system). It would be an optical disk system with sufficient read/write heads to match the short-term bandwidth. The long-term storage function is not shown and is beyond the scope of the PPCS subsystem. It could be satisfied by the long-term storage of the optical disks as they are filled up, or it could be satisfied by a high-density tape facility. The long-term storage of imagery is not necessary for the functioning of the PPCS subsystem.

Figure 1 shows a growth interface capability, which is a set of ports on the high data rate bus between the short-term image store, the medium-term image store, and the photointerpreter subsystem. These ports can be connected to additional medium-term image stores and/or short-term image store/image processor units.

The detailed designs of the function blocks in Figure 1 are described in GER-16624, 16627, 16650, 16659, 16679, and 16964, which are available from Goodyear Aerospace.

THROUGHPUT ANALYSIS

The throughput analysis is broken down into two parts. In the benchmark section, the most important algorithms, as determined by the algorithm analysis phase, are analyzed to determine the estimated speed at which they can be performed in the PPCS image processor.

Due to the wide variation in sensor data rates, the benchmarks are not based on any one sensor's image format but are expressed as percent of real-time assuming a 100 Mbits/second input rate. Since the image processor is a bit sliced processor, these rates easily can be extrapolated to any sensor's data format. That is, in the PPCS image processor, it takes one-half the time to process 4-bit pixels as it does to process 8-bit pixels.

Due to the importance of FLIR imagery, a further throughput analysis of the time required to process the imagery in the scenario outlined in the Background section was performed. This analysis includes a description of a continuous image processing approach.

Benchmarks

During the algorithm analysis, the algorithms in Table 2 were determined to be representative of the most frequently used algorithms for FLIR, NLIR, radar, and optical imagery. The benchmark values were determined by assuming that the input imagery resided in the short-term image store (STIS) and that the output imagery was to be output to the STIS.

It should be kept in mind that Table 2 expresses throughput in terms of realtime for 100 Mbits/second. However, in any practical implementation of an image screener, it is wasting resources to apply all functions to all imagery. This is a major reason for using a SIMD architecture for the image processing portion as opposed to a pipeline processor. It is difficult to apply algorithms to a run-time selected subsection of an image with a pipeline architecture since the efficiency of a pipeline processor comes from performing the same function on everything in the pipe. However, with a SIMD processor, the efficiency comes from performing the same function in parallel on an entire subportion of the image. If a new subportion of the image is to be processed with a different algorithm, it is simply a matter of executing a different sequence of code.

TABLE 2. ALGORITHM EXECUTION SPEEDS*

Item	Speed (times real-time)
Convolution (3 x 3)	6.21
Convolution (7 x 7)	1.14
Template matching (7 x 7)	142.66
Pseudo-median filter (3 x 3)	74.47
Histogram computation	0.01
Gray-scale thresholding	569.88
Region growing	39.72
Two-dimension cross correlation (13 x 13)	163.46
Target position determination	10485.76
Gray-scale averaging (32 x 32)	1.6384

*Assuming data rates of 100 Mbits/second.

Scenario Analysis

The scenario analysis is based on processing every frame of FLIR imagery in a 2-1/2-minute sequence. Since overlapping frames of imagery is extremely redundant, not every algorithm is applied to every portion of imagery. Because of the 100 percent programmable nature of the image processor, the portion of imagery and the algorithms to be applied can be determined during the processing on a real-time basis. This allows the PPCS system the flexibility to concentrate the computing power of the system on the areas determined to be "most interesting" by the system.

Continuous image processing has the advantage over other approaches of providing a new type of information to image processing - time-varying data. It has been shown that time-varying imagery is a rich source of information;** it is now realized that the application of artificial intelligence techniques is easier to time-varying imagery than static imagery.***

For example, algorithms have been developed for detecting moving objects** in sequences of imagery. These algorithms require no special hardware and, moreover, do not lose targets when they stop. They can easily track any number of targets moving in random directions. This ability adds a new dimension of information that can be used to help recognize clusters of targets. Since these algorithms detect motion only and are immune to optical and IR camouflaging techniques, they also can be helpful for detecting camouflaged targets.

[**] Potter, J. L., "Scene Segmentation Using Motion Information," Computer Graphics and Image Processing, 6, 1977, p. 558-581.

[***] Potter, J. L., "Extraction and Utilization of Motion in Scene Description," Ph. D. Thesis, University of Wisconsin, 1974.

Continuous FLIR Processing - TV-formatted infrared imagery is extremely redundant. It is estimated that, in a typical sortie, from 80 to 95 percent of the ground covered by a frame is also covered in the subsequent frame. This redundancy can be taken advantage of in two ways: (1) process every Nth frame where N is large enough to assure a minimal amount of overlap (typically N=10), and (2) process every frame keeping track of the information content of previous frames to guide the processing of the current frame. This continuous processing approach is discussed below.

The continuous image processing approach to FLIR image exploration has several advantages over the "process every Nth frame" approach. First, there is a lower computing power requirement. Second, it is most likely to allow instantaneous image processing. Third, it can take advantage of the information content present in sequences of imagery not present in a single frame. Fourth, it provides multiple images (one per frame) of a target which can be used for recognition.

The continuous image approach can process every frame in real-time since it remembers what it saw in the previous frames and needs only to update its memory. Since every frame is processed, new data (or status) can be detected on a frame-by-frame basis achieving true frame-by-frame real-time image exploitation.

For example, histogram computation is an extremely time-consuming operation taking perhaps 100 times real-time. However, in a continuous mode, the histogram for a sequence of imagery can be calculated during the first few frames when no targets of interest are present and then updated in real-time throughout the remainder of the sequence on a frame-by-frame basis simply by statistical sampling.

Finally, processing every frame takes advantage of all of the information content of the imagery. In particular, noisy regions in one frame can be smoothed by the corresponding regions in adjoining frames. This may be very important when attempting to classify targets. Moreover, syntactic information at a microscopic and macroscopic level can be easily integrated in a continuous process. For example, the information about the interrelationships of several targets (syntactic information) is easier to extract from a frame in which the targets are too distant to provide the best classification information. However, the information content of two images of the same target (one distant, one up close) would be difficult to correlate if the two frames are taken in isolation. In a continuous process, the target information extracted in the distant frame is assigned to the targets and updated constantly so that, when the closeup frames are processed, the distant information is already associated with the proper target.

(1) Computation Zones - In the idealized world, the targets on FLIR imagery would appear at the tops of the frame, move downward, and exit at the bottom. In the real world, the movement of the platform (sensor) while imaging would affect the flow of the targets. During turns, for example, targets could appear along one side and disappear on the other, in addition to the top and bottom. However, since the side-to-side flow can be processed in exactly the same manner as the top-to-bottom flow, the remainder of this discussion will assume top-to-bottom target flow only.

The imagery will be divided into three sections as shown in Figure 2. If one pixel covers approximately one foot and the top most 128 rows are processed for new targets every 1/30 second, then the combined speed of the sensor and the target

would have to be: (128 pixels - 2 overlaps* (9 pixel/overlap)* 30 frame/sec* 60 sec/min* 60 min/hours* 5280 ft/mile = 2250 miles/hour in order for the target not to be present in the 128-row band (the new target zone) in at least one frame. This seems to be a safe assumption; therefore, only the new target zone of the image need to be processed for new targets.

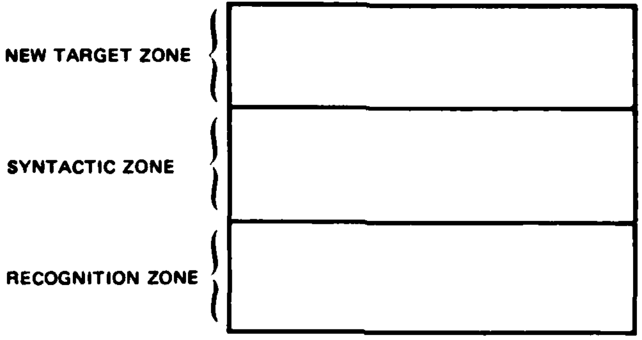


Figure 2. Computational Zones

The bottom portion of the imagery represents the targets closest to the sensor and therefore represents the highest resolution and lowest distortion portion of the imagery. This section should be used to classify targets since the most accurate classification information can be obtained in this area.

The middle portion of the imagery is well suited for extracting syntactic relationships. The only other processing that needs to be performed on the targets in this region is to track them. This allows the syntactic data to be associated with the proper target as it moves into the recognition zone. With this association, the syntactic data can be used for target recognition.

As the three zones of imagery suggest, the computational processes can be divided into three functional pipeline stations: one for detecting targets, one for extracting syntactic data, and one for recognizing/identifying targets. Physical pipeline implementation of these functions is, however, not effective because all imagery must go through every station of the pipe regardless of whether it is to be processed. Thus, the detecting station would be idle over two-thirds of the time when the tracking and recognizing portion of the imagery is being processed. Consequently, functional but not physical pipelining is the best approach to continuous image processing.

(a) New Target Zone

(a-1) Target Detection - Potential new targets in the new target area of the imagery are identified by first passing a median filter over the image to remove salt and pepper noise. A histogram of the area is then computed. Then, a search for the "hottest spots" is made*. These spots are then used as the seeds for region growing. The region growing process is based on the histogram and is designed to find regions of the correct size as a function of the parameters supplied during pre-mission planning such as sensor height and attitude, cued objects, etc.

---

[*] Cold spots also are important in target detection in FLIR imagery. Since a target is likely to be composed of hot and cold spots, if the lower values are complemented (converted to hot values), then the above process will group associated hot and cold spots into the same region.

Regions that pass the size criteria will be labeled potential targets. Every region will be labeled, and all attributes of the region will be entered into the target data base. Some of the attributes available at this time are position (within the sensor frame of reference), length, width, and average intensity.

After a potential target has been detected in a frame, it must be verified in subsequent frames. This is done by tracking the target region as it moves from frame to frame. If a target region cannot be verified by tracking, it is eliminated from the target list. If it is verified, it is labeled as a candidate target.

(a-2) Target Tracking - During the extent calculation of the region growing process, the position of each region is calculated and saved. This position information is used for target tracking. There are two cases to consider. In one, the target region was first identified in the previous frame (it is a potential target). In the second, the target has been firmly established (it is a candidate target).

Candidate targets (case two) are processed first. Based on past history, the relative velocity of the target and the sensor is well known and the position of the target in the previous frame can be easily updated to the predicted position in the current frame. A search is made for a region at the predicted location allowing for noise tolerances. If a positional match is found between the predicted target position and an actual calculated blob position in the current frame, the positional information is updated and the region (target) is removed from the frame.

If a region in the current frame is not found at the predicted location, a wider search is executed. If the region is found, in addition to updating the position, the position prediction parameters are updated. Also, the last four positional fixers are saved for this purpose. If the region is still not found, the target is put on a "missing" list for more detailed analysis later as time permits.

As the candidate target regions are processed, their regions in the current frame are deleted. Thus, the only regions left are those that are new this frame or were new the previous frame. The positions of the regions that were new in the previous frame are translated to the expected position based on sensor motion. These new regions are then tracked in the same manner as candidate targets.

While the tracking process is identical to the case two process, the searching algorithm for the case one target region is much more sophisticated since less information about the relative motion is known. Furthermore, some of the regions could be noise, for which no matching region in the previous frame can be found. The analysis of the situation must be made rapidly.

If a successful match is found, the new target region is marked as a candidate target region and its relative motion is entered in the target data base. If matching is unsuccessful, the target region is deleted from the list of potential targets.

After the newly acquired targets have been verified, they are eliminated from the frame in the same manner as the candidate targets were. The remaining regions of target size in the new target region of the current frame are processed and entered in the data base as potential target regions.

(b) Syntactic Zone - Since the purpose of this system is to recognize targets with a high degree of accuracy, after all positions for the known target regions have been updated, but before the remaining target regions are processed, the known target regions in the syntactic zone are processed for syntactic features. Both global and local features are extracted.

Global features tend to group individual target regions into groups of targets. These features could be extracted directly from the imagery, but most commonly they will be obtained from the target data base. For example, the same relative motion of targets could be used to group vehicles traveling in a convoy. The grouping of targets by proximity can easily be accomplished by searching on the target's coordinate position. These global features can be extracted prior to any knowledge of the specific target type. Indeed, these global features may be very useful in determining what local features to look for to perform target recognition.

Since local features are basically simple subregions within the target region, many of the same algorithms described above for target detection may be used for local feature extraction. For example, if hot spots and cold spots are treated independently and the region size parameters modified, the region growing algorithm can be used to identify the prominent hot and cold spots within a target that are crucial for its identification.

Once these subregions are identified, their syntactic interrelationships can be obtained for comparing subregion coordinate positions. For example, left of, right of, and similar properties in addition to vicinity grouping can be obtained by coordinate comparison.

<u>(c) Target Recognition Zone</u> - By the time the candidate target appears in the target recognition zone, several clues as to its type have already been extracted. These clues will be used to guide the recognition subsystem. One clue is the anticipated target types entered during the premission preparation phase. The other clues are global syntactic in nature.

During the premission briefing, the system can be set to one of several processing modes. In the directed mode, the system will look for and recognize only those targets specified during the briefing. By restricting the system to specific targets, more extensive recognition algorithms can be utilized resulting in a better classification ratio and classification of more difficult targets.

In the cued mode, the system will look for the targets specified during the premission briefing as highest priority. However, if a candidate target region cannot be recognized as one of the cued targets, the system will attempt to classify it as one of the other target types in its system as time permits.

Finally, in the free mode, the system will look for all target types with equal intensity. Extracted image data will be used to direct the recognition process for maximum effect. For example, if a missile acquisition unit was previously recognized in an area, its accompanying tracking/firing units would be given top priority in subsequent searches.

Perhaps one of the most important aspects of continuous FLIR processing is that the target recognition process can be performed over several frames. For example, if a candidate target region in a given frame cannot be accurately identified, the same region in earlier and/or later frames where the resolution is better can also be processed. If none of the frames is sufficiently clear to give a reliable target assignment, the consensus of assignments across all frames should provide a reliable assignment.

(2) Target Presentation

(a-1) Priority Ordering - The priority ordering of targets is done during premission planning. Priority can be based on target type (missile launches are more important than jeeps) or on location (any target within a specified coordinate range has highest priority), or a combination of both.

Once the processing of the imagery begins, the order of target presentation to the photointerpreter is a function of the priorities established during the premission planning phase. Normally, the highest priority targets are displayed first, but with real-time processing lower priority targets may be displayed first because they were found first. If a low-priority target is being displayed when a higher priority target is detected, the user will be alerted by a tone and a display of the priority target type on the CRT. The user can then abort the current image with the provision that it be put back in the queue or that it be abandoned, or he may choose to continue working on the current image.

At all times, there is a display of the target types waiting in the queue. This display will be ordered according the priority. When the display is idle, the top-priority image would be displayed. However, the user can call the imagery out of the queue in any order he desires.

(a-2) Target Display - The presentation consists of the imagery with the recognized targets circled with a contrasting color (red for example). Unrecognized targets (targets with a recognition confidence level less than the parameter set during premission planning) are circled with a different color (say blue). All target regions will be labeled with a colored label for easy identification. The remainder of the image is black and white.

The targets and target regions can be identified by the user in two different ways. One, the cursor can be positioned over the target and its position on the screen used to identify the target region nearest it. Second, the label of the target displayed on the screen can be used for identification. If the system has a voice recognition unit, the label could be spoken or entered on the console.

The photointerpreter can identify a target and request its attributes displayed. The photointerpreter can modify the attributes and request the recognition process to be repeated, can manually define a target region and request the recognition process to be executed on that region, and can request that all frames with the target of interest be displayed at a rate under his control. If a multiple-frame recognition process was performed, the photointerpreter can request all target assignments and the associated frame to be displayed.

When more than one target type is being processed, all potential targets will be correlated against the target type masters. The highest correlation is used to identify target types. If the correlation between one target master and a potential target is high and the correlation is low for all other targets, the classification can be considered to be correct. However, upon occasion, the correlation distinction between targets may not be clear cut. In these instances, the photointerpreter will be notified of all possibilities in order of probability.

In these situations, the photointerpreter must decide what the target is. However, several classification aids are available. One is classification consistency. In any given sequence, a number of target types will be detected. It is the nature of military science that certain targets will be accompanied by other specific target types. For example, tanks must have support vehicles such as fuel transports in the immediate area. These interrelationships can be reviewed at photointerpreter direction to verify proper classification.

At the user's request, the system will present a list of possible target types with their probability along with the target imagery. The photointerpreter enters his choice with a request for global verification. The photointerpreter's choice of target type will bias the classification of other targets in the area. The effect of this biasing on the adjacent targets will be displayed as an overlay and on a CRT.

(3) I/O Analysis - One of the most important aspects of effectively using large volumes of computing power is inputting and outputting data in an effective manner. The preliminary design calls for I/O to be overlapped with computation in a pipeline fashion. This overlap allows the I/O overhead to be negligible. At 100 nsec/cycle and 128 columns per bit plane, it takes 12.8 microseconds to input a bit plane plus one cycle to store it into memory. If there are eight bits per pixel, then it takes 102.4 microseconds plus 0.8 microseconds, or a total of 103.2 microseconds to input a byte (pixel) plane (6.3 nsec per pixel). However, there are times when a more complex I/O process is desired. For example, many image processing functions require access to neighboring pixel values. The most efficient way to provide this access is to redundantly store the neighboring pixel values with each pixel. Figure 3 illustrates this data storage arrangement for a 3 x 3 window.

In the normal process I/O flow, each column of bits passes by each column of memory; thus, the memory need only to "grab" the neighboring column data as they are shifted by.

Figure 4 illustrates this concept using pixels instead of bits for ease of comparison with Figure 3. The concept is the same; it just takes eight repetitions to input a pixel.

Comparing Figures 3 and 4 shows that, at cycle j-1 in Figure 4, the data is in alignment for a memory store for index (0,1) in Figure 2. At cycle j, the data is in alignment for index (0,0); at cycle j+1, it is in alignment for index (0,1). Since there are eight bits per pixel, it takes just eight additional cycles per redundant column (east-west neighbor) store.

This procedure provides the redundant column data but not the redundant row data. The row data must be shifted up and stored and then shifted down and stored. Since this shift operation requires the P register, it cannot be performed by the I/O controller. It takes three additional cycles per bit per column position (nine cycles altogether).

INDEX	MEMORY (i,j-1)	MEMORY (i,j)	MEMORY (i,j+1)
(-1,-1)	pixel (i-1,j-2)	pixel (i-1,j-1)	pixel (i-1,j)
(0,-1)	pixel (i,j-2)	pixel (i,j-1)	pixel (i,j)
(1,-1)	pixel (i+1,j-2)	pixel (i+1,j-1)	pixel (i+1,j)
(-1,0)	pixel (i-1,j-1)	pixel (i-1,j)	pixel (i-1,j+1)
(0,0)	pixel (i,j-1)	pixel (i,j)	pixel (i,j+1)
(1,0)	pixel (i+1,j-1)	pixel (i+1,j)	pixel (i+1,j+1)
(-1,1)	pixel (i-1,j)	pixel (i-1,j+1)	pixel (i-1,j+2)
(0,1)	pixel (i,j)	pixel (i,j+1)	pixel (i,j+2)
(1,1)	pixel (i+1,j)	pixel (i+1,j+1)	pixel (i+1,j+2)

Figure 3. Redundant 3 x 3 Storage

I/O PLANE	MEMORY (i,j-1)	MEMORY (i,j)	MEMORY (i,j+1)
cycle j-1	pixel (i,j)	pixel (i,j+1)	pixel (i,j+2)
cycle j	pixel (i,j-1)	pixel (i,j)	pixel (i,j+1)
cycle j+1	pixel (i,j-2)	pixel (i,j-1)	pixel (i,j)

Figure 4. Pixel Presentation

Therefore, the time required to produce the data format in Figure 3 is:

$$\text{Total} = 8 \text{ bits/byte}^{\dagger} \text{ (I/O controller time + PE controller time)}$$

$$\text{Total} = 8 \text{ bits/plane}^{\dagger} \text{ (shift into position + store + shift + store + shift + store)}$$

$$+ \begin{array}{c}\text{load P}\\ \text{set topo}\end{array} + \begin{array}{c}\text{shift and}\\ \text{store}\end{array} * \begin{array}{c}\text{no. of}\\ \text{elements}\end{array} -1 * \begin{array}{c}\text{no. of}\\ \text{elements}\end{array}$$

$$= 8^{*}\ (127+1+1+1+1+1(1+1^{*}2^{*}3)) = 1112 \text{ cycles}$$

More generally, let (n,m) be the window size, then

$$\begin{array}{c}\text{Total per}\\ \text{bit plane}\end{array} = 8^{*}\ (128-\frac{n-1}{2} + 2n+nm)$$

$$= 1020+12n+8nm, \text{ where}$$

1020+11n+7nm* of the cycles are performed by the I/O controller and are therefore overlapped with the computational processes. Thus, for n=3, it takes 112.8 microseconds to input a byte plane in the redundant configuration shown in Figure 3,

---

[†] that is, (n+nm) *8 cycles of the PE controller are required.

but only 9.6 microseconds of processing time is required. Table 3 and Figure 5 show the I/O requirements for various window sizes.

TABLE 3. I/O TIMES FOR REDUNDANT DATA FORMAT

Window size	Total I/O time per pixel plane (microseconds)	Computing time required (microseconds)
1 x 1	103.2	0.1
3 x 3	112.8	9.6
5 x 5	128.0	24.0
7 x 7	149.6	44.8
9 x 9	177.6	72.0

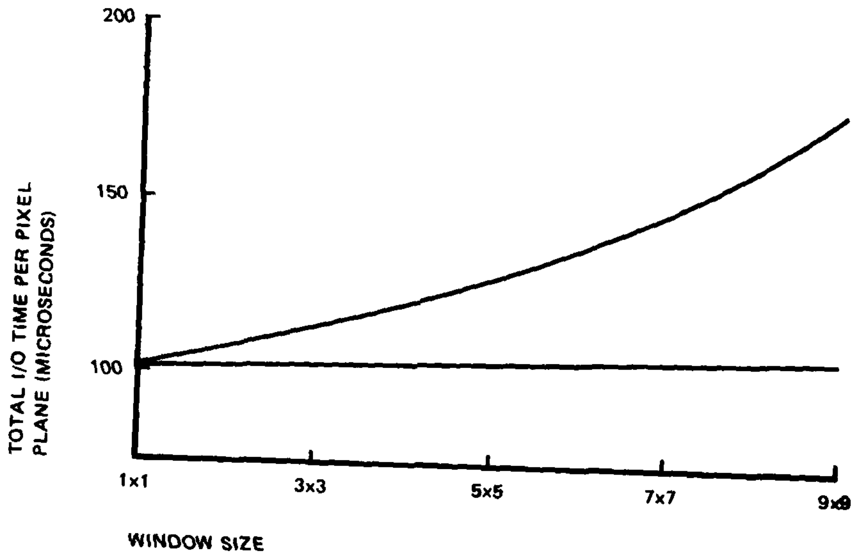


Figure 5. I/O Times

The rationale behind this approach is that, if more than one area process (convolution, median filter, etc.) is to be performed on an image, the inter-pixel communication can be accomplished in one movement of the data during input instead of once for each function. This results in a considerable improvement in execution time.

Factors that must also be accounted for in addition to the number of area functions to be applied to the image are the window size and memory size. For a window size of 9 x 9 and less, this approach is the most efficient and was used when calculating algorithm timing.

(4) Timing Estimates - Table 4 shows the timing estimates for the more important functions required for continuous processing. These estimates are based on 512 x 512 x 8-bit frame size.

A PDL (program design language) of the continuous process was developed and these times applied to it. The total time estimate for processing one frame is 13.78 milliseconds. This estimate was achieved by performing the histogram on approximately 5 percent of the imagery. This reduces the histogram processing time to 10.29 milliseconds/frame and results in a 152 Mbit/second throughput rate.

TABLE 4. ALGORITHM TIME ESTIMATES (MICROSECONDS)

Item	Time
Median filter	281.6 per frame
Histogram	205,712 per frame
Threshold	36.8 per frame
Region growing	528 per frame
Coordinate search	230 per frame
Two-dimensional cross correlation	128.3 per target

RECOMMENDATIONS

Table 5, which is derived from Table A-I in Appendix A to PPCS Status Report No. 11, summarizes the estimated percentage of the PPCS system computing power required under various operating conditions, assuming 100 Mbits/second per sensor. This table demonstrates that the PPCS system can handle single sensors (100 Mbits/second) in real time under all conditions and multiple sensors (200 Mbits/second) in near real-time. The high computational rate and the 100 percent programmability aspect of the PPCS system design allow multisensor operation, thus resulting in a common design for multiple exploitation systems and thereby reducing life cycle costs. Moreover, since multiple-sensor data can be processed simultaneously, new approaches (such as artificial intelligence) to difficult problems (such as camouflaged targets) can be supported in real-time or near real-time.

Goodyear Aerospace recommends that a detailed system design of the PPCS subsystem be undertaken as soon as possible; this would assure the availability of a PPCS subsystem in a test bed environment in the 1984-1985 time frame.

The transition from test bed to a field-deployable unit could be readily accomplished since Goodyear Aerospace is well versed in the state-of-the-art design of military-qualified equipment and maintains IR&D programs to assure continued excellence in this area.

TABLE 5. THROUGHPUT REQUIREMENTS*

Automatic screening

	Single sensor†	Multiple sensor (2 sensors)**	
		Without AI	With AI
Real time	0.66	1.33	1.46‡
Near real time	0.33	0.65	0.67‡

Automatic screening and target locating
Nonreport correlation

	Single sensor†	Multiple sensor (2 sensors)**	
		Without AI	With AI
Real time	0.80	1.59	1.90
Near real time	0.39	0.79	0.87‡

Automatic screening and target locating
Report correlation

	Single sensor	Multiple sensor (2 sensors)**	
		Without AI	With AI
Real time	0.88	1.75	2.09
Near real time	0.43	0.87	0.95‡

BIBLIOGRAPHY

Aggarwal, R. K., et al: "Context Dependent Automatic Image Recognition System," <u>Proceedings: Image Understudy Workshop</u>, April 1979, p. 145-150.

Aggarwal, R. K.: "Image Segmentation for Syntactic Classification of Large Images," <u>Proceedings: Image Understanding Workshop</u>, May 1978, p. 125-132.

---

[*] Computing power requirements are based on 100 Mbits per second per sensor.

[**] Can be time shared during pass; that is, multiple-sensor data processed on a frame-interleaved basis.

[†] Can be time shared on a pass-by-pass basis; that is, sensor Type A during Pass 1, sensor Type B during Pass 2.

[‡] Optical disk required.

Aggarwal, R. K.: "Image Segmentation Using Prototype Similarity, SPIE, Vol 137, 1978, p. 106-113.

Aggarwal, R. K.; Wittenburg, T.M.: "Syntactic Recognition of Tactical Targets," Proceedings: Image Understanding Workshop, November 1977, p. 48-58.

Andrews, H. C.: "Analytic Results of the Coleman Segmentor" (USC Image Processing Institute), Proceedings: Image Understanding Workshop, October 1977, p. 96.

Attridge, C. C.: "Target Isolation Technique Studies," p. 120-125.

Barbe, D. F.: "VHSIC Systems and Technology," IEEE Computer, February 1981, p. 13-22.

Bauer, B. P.: "A Survey of Image Processing Development in Support of Remote Sensing," June 1980.

Block, E.; Galage, D.: "Component Progress: Its Effect on High-Speed Computer Architecture and Machine Organizations," IEEE Computer, 1978, p. 64-76.

Brown, D.; Frei, W.: "Target Segmentation in Infrared Images."

Burr, D. J.: "A Dynamic Model for Image Registration," Computer Graphics and Image Processing, February 25, 1980, p. 102-112.

Deal, B.: "Automatic Target Cuer - First Quarter Report," Prepared for U.S. Army Night Vision and Electro-Optics Laboratories, October 23, 1979.

Dehne, J. S.: "Advanced Target Cueing Techniques," Prepared for U.S. Army Night Vision and Electro-Optics Laboratories, SPIE, Vol 137, 1978, p. 102-105.

Duff, M. J.: "Parallel Algorithms and Their Influence on the Specification of Application Problems," University College, London.

Eversole, W. L.; Hewes, C. P.; Buss, D. D.; Cheek, T. F.; Brasher, W. E.: "CCD Signal processor for Point Target Detection/Location System,"SPIE, Vol 124, 1977, p. 33-40.

Gemmar, P.: "Image Correlation: Processing Requirements and Implementation Structures on a Flexible Image Processing System (FLIP)," Research Institute for Information Processing and Pattern Recognition, Karksruneg, FRG.

Haas, W. H.; Dixon, S.: "Performance on Mosaic Sensor Dim - Target Detection Requirements," SPIE, Vol 178, 1979, p. 25-32.

Habicht, J. C.; Helland, A. R.: "Real Time Classification Technique for Automatic Target Cueing."

Hanaki, S.; Temma, T.: "Template-Controlled Image Processor (TIP) Project," C&C Systems Research Laboratories, NEC, Japan.

Helland, A. R.; Tisdale, G. E.: "The Auto-Q Digital Image Processor for Auto Processing Reconnaissance Images," Proceedings of IEEE 1980 National Aerospace and Electronics Conference (NAECON), 1980.

Holben, R. D.: "A Moving Target Indicator Algorithm for Passive Sensors," Proceedings of IEEE 1980 National Aerospace and Electronics Conference (NAECON), 1980.

Hsu, Y. S., et al: "Useful Image Transform Coefficients for Pattern Recognition," Proceedings of Society of Photo-Optical Instrumentation Engineers, Vol 149, p. 112-117.

Huany, T. S.; Fu, K. S.: "Image Understanding and Information Extraction," Proceedings: Image Understanding Workshop, May 1978, p. 147-149.

Hunt, B. R., "Computer and Images," SPIE, Vol 74, 1976.

Ishii, M.; Inamoto, Y.: "Memory Structures for an Image Processing System," Fujitsu Laboratory, LTD, Japan.

King, J.: "Image Segmentation and Object Detection by a Syntactic Method," Proceedings: Image Understanding Workshop, April 1977, p. 38-43.

Kung, H. T.; Picard, R. L.: "Hardware Pipelines for Multi-Dimensional Convolution and Resampling," ESLInc., May 1981.

Kung, H. T.; Song, S. W.: "A Systolic Two-Dimensional Convolution Chip," Department of Computer Science, Carnegie Mellon University, March 1981, p. 1-8.

Lantuejoul, C.: "Geodesic Segmentation," Centre de Geostatistique et de Morphologie Mathematique, Ecole des Mines de Paris.

Larson, R.: "Automatic Image Recognition System," Proceedings: Image Understanding Workship, May 1978, p. 145-146.

Larson, R.: "1976-1977 Program Review," Proceedings: Image Understanding Workshop, April 1977, p. 96-97.

Liang, A. C.: "A Technique for Infrared Background Suppression," SPIE, Vol 124, 1977, p. 2-9.

Lo, C. M.: "FLIR Image Enhancement for the Automatic Target Cuer System," Northrup Corporation, March 25, 1980.

Lo, C. M.: "A Survey of FLIR Image Enhancements."

Lundgren, K.; Granlund, G. H.: "Image Processing Applications Enabled by MIMD Processor Structures," Linkoeping University, Sweden.

Martin Marietta: "Digital Image Processing and Autonomous Target Recognition at Martin Marietta."

Milgram, D. L.: "Region Extraction Using Convergent Evidence," Proceedings: Image Understanding Workshop, April 1977, p. 53-62.

Minter, T. C.: "A Discriminant Procedure for Target Recognition in Imagery Data," Proceedings of IEEE 1980 National Aerospace and Electronics Conference (NAECON).

Mitchell, D. R.; Lutton, S. M.: "Segmentation and Classification of Targets in FLIR Imagery," Proceedings: Image Understanding Workshop, November 1978, p. 59-65.

Mitchell, D. R.: "Target/Background Segmentation and Classification in FLIR Imagery," Proceedings: Image Understanding Workshop, May 1978, p. 115-117.

Morefield, C. L.: "Decision-Directed Multitarget Tracking," 1978 Conference on Decision and Control, San Diego, California, January 10-12 1979 (TORAC Corporation).

Otazo, J. J. (Parenti): "Digital Filters for the Detection of Resolved and Unresolved Targets Embedded in Infrared (IR) Scenes," SPIE, Vol 178, 1979.

Panda, D. P.: Aggarwal, R. K.: "Advanced Techniques for Airborne Target Screener," Proceedings of IEEE 1980 National Aerospace and Electronics Conference (NAECON), 1980.

Panda, D. P.: "Target Screener System Noise Removed by Interface Analysis," Proceedings: Image Understanding Workshop, May 1978, p. 22-27.

Potter, J. L.: "Scene Segmentation Using Motion Information," Computer Graphics and Image Processing, 6, 1977, p. 558-581.

Potter, J. L.: "Extraction and Utilization of Motion in Scene Description," Ph. D. thesis, University of Wisconsin, 1974.

Quam, L. H.: "Road Tracking and Anomaly Detection in Aerial Imagery," Proceedings: Image Understanding Workshop, May 1978, p. 51-55.

Raimondi, P. K. (Jones): "Performance of Image Enhancement and Target Cueing Techniques on Grafenwoefr II Imagery," SPIE, Vol 178, 1979.

Reeves, A. P.: "Parallel Algorithms for Real-Time Image Processing," Purdue University.

Reid, D. B.: "An Algorithm for Tracking Multiple Targets," IEEE Transactions on Automatic Control, Vol AC-24, No. 6, December 1979.

Rosenfeld, A.: "Cellular Architectures: From Automata to Hardware," Computer Vision Laboratory, University of Maryland.

Rubin, L. M.: "Automatic Target Cueing," Proceedings of Fourth International Joint Conference on Pattern Recognition, p. 774-779.

Sato, M.; Matsuula, H.; Ogawa, H.; Ijjima, T.: "Multiprocessor System PX-1 For Pattern Information Processing," Tokyo Institute of Technology.

Sea, R. G.: "Optimal Correlation of Sensor Data with Tracks in Surveillance Systems," Hawaii International Conference on System Sciences, 6, Honolulu, Hawaii, January 9-11, 1973.

Serreya, D.; Larson, R.: "Adaptive Threshold for an Image Recognition System," Proceedings: Image Understanding Workshop, October 1977, p. 73-74.

Shneier, M.: "Using Pyramids to Define Local Thresholds for Blob Detection," Proceedings: Image Understanding Workshop, November 1979, p. 31-35.

Snyder, W. E.; Cowart, A. E.: "An Iterative Approach to Region Growing."

Soland, D. E. (Narendra): "Prototype Automatic Target Screener," SPIE, Vol 178, 1979.

Sos, J.; Cennell E.; McCaleb, F.: "Ground Data Processing Technology," in NASA Space Systems Technology Model, Vol II, Space Technology Trends and Forecasts, May 1980, p. 4-101 to 4-108.

Tanimoto, S. L.: "Programming Techniques for Hierarchical Parallel Image Processing," University of Washington.

Tao, T. F., et al: "Focal Plane Processing Techniques for Background Clutter Suppression and Target Detection," SPIE, Vol 178, 1979.

Tavakoli, M.: "Toward the Recognition of Cultural Features."

Tisdale, G. E.: "A Digital Image Processor for Automatic Target Cueing, Navigation, and Change Detection," Proceedings of Society of Photo-Optical Instrumentation Engineers, Vol 101, p. 112-119.

Tisdale, G. E.: "Evolution of an Automatic Target Cueing System," SPIE, Vol 137, 1978.

Uhr, L.; Schmitt, L.; Hanrahan, P.: "Cone/Pyramid Perception Programs for Arrays and Networks," University of Wisconsin.

Van Atta, P. W.: "Army Prototype Automatic Target Screener," SPIE, Vol 137, 1978.

Westinghouse: "Algorithms and Hardware Technology for Image Rcognition." Computer Science Center, University of Maryland.

Westinghouse: "A Discussion of Hardware Implementation and Fabrication for an Automatic Target Cueing System, April 30, 1977.

Willett, T. J.: "Automatic Target Cueing on the Focal Plane," Proceedings: Image Understanding Workshop, April 1977, p. 87-88.

Willett, T. J.; Tisdale, G. E.: "Hardware Implementation of a Smart Sensor: A Review," Proceedings: Image Understanding Workshop, May 1978, p. 1-8.

Willett, T. J.; Helland, A. R.; Tisdale, G. E.: "Higher Level Algorithms: Evaluation and Implementation," Proceedings: Image Understanding Workshop, November 1979, p. 15-23.

Wood, R.: "Hybrid Optical/Digital Processing for Target Identification," Proceedings of Society of Photo-Optical Instrumentation Engineers, Vol 209, p. 74-79.

DATE
FILMED

2-8