NEW MEXICO STATE UNIVERSITY

REAL-TIME VIDEO TRACKING CONCEPTS

Department of Electrical and Computer Engineering
New Mexico State University

NMSU-TR-79-1

Final Report

for

Grant DAAG-29-76-G-0231

Submitted to:

U.S. Army Research Office
Box 12211
Research Triangle Park, NC 27709

July 1979

REAL-TIME VIDEO TRACKING CONCEPTS

Department of Electrical and Computer Engineering
New Mexico State University

NMSU-TR-79-1

Final Report

for

Grant DAAG-29-76-G-0231

Submitted to:

U.S. Army Research Office
Box 12211
Research Triangle Park, NC 27709

July 1979

DISCLAIMER

The findings of this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

(14) 1. REPORT NUMBER: NMSU-TR-79-1

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

(6) 4. TITLE (and Subtitle): REAL-TIME VIDEO TRACKING CONCEPTS

(9) 5. TYPE OF REPORT & PERIOD COVERED: Final Report

6. PERFORMING ORG. REPORT NUMBER

(10) 7. AUTHOR(s): G. M. Flachs, P. I. Perez, R. B. Rogers, S. J. Szymanski, J. M. Taylor, W. E. Thompson, and Yee Hsun U.

(15) 8. CONTRACT OR GRANT NUMBER(s): DAAG-29-76-G-0231

9. PERFORMING ORGANIZATION NAME AND ADDRESS
New Mexico State University
Las Cruces, NM 88003

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
U. S. Army Research Office
Box 12211
Research Triangle Park, NC 27709

(11) 12. REPORT DATE: June 1979

13. NUMBER OF PAGES

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

(12) 45p.

15. SECURITY CLASS. (of this report): Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

(18) ARO

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

(19) 13851.4-EL

18. SUPPLEMENTARY NOTES

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

Image Processing
Real-Time
Video Tracking
Image Decomposition
Projections
Video Processing

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

Real-time video (RTV) tracking concepts are developed for tracking high performance targets in noisy background scenes. Image processing techniques are combined with parallel processing methods to perform a statistical analysis of a TV image, partition the picture into target and background components, perform a structural analysis of the target image, and control the tracking optics to form an automatic tracking system.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

410719

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

LIST OF FIGURES

I. INTRODUCTION

The thrust of the research sponsored by this grant (DAAG-29-76-G-0231) is to develop real-time video (RTV) tracking concepts compatible with current technology. The results of the research provided the basic concepts and mathematical foundations for the development of a prototype RTV tracking system [22] currently being tested by White Sands Missile Range (WSMR) in New Mexico. The RTV tracking system, shown in Figure 1, is designed to interface with the Contraves Model F Cinetheodolite tracking mount. The major components of the system include the tracking optics, the RTV processor, and the video tape recorder. The tracking optics includes the tracking mount, image rotation element, zoom lens, and TV camera. The RTV processor receives the video signal, locates the target images, and provides the control signals to the tracking optics to form a closed-loop, automatic tracking system. The video tape recorder records the target flight and stores the tracking data in the vertical retrace interval.

The research focused the solution of the tracking problem onto four major problems in real-time image processing. These problems are:

- real-time image decomposition
- real-time structural characterization
- real-time tracking algorithm
- real-time control algorithm

The first problem concerns the separation of the target image from the background. This problem is mathematically formulated in terms of statistically characterizing the target and background intensity distributions and deciding, for a given intensity, whether it came from the target

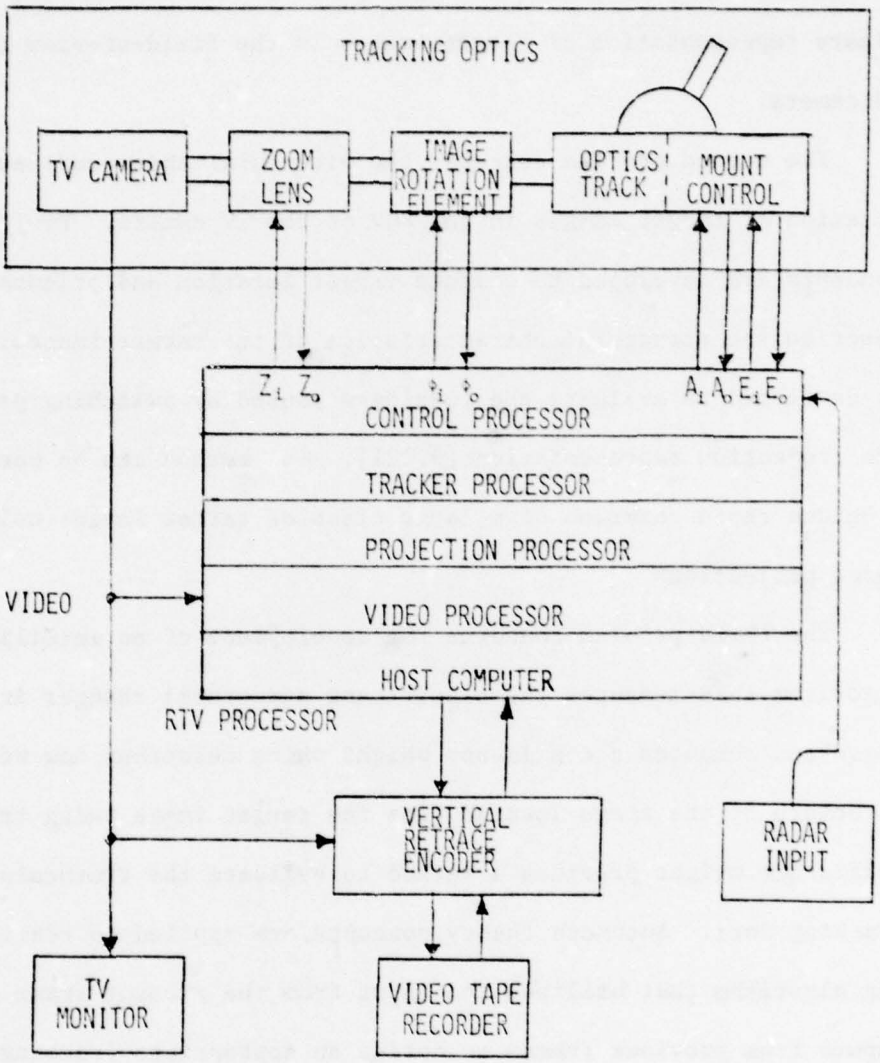

Figure 1. A RTV Tracking System

or background distributions. The results of this research provided the basis for a real-time method for statistically analyzing a digitized TV picture, separating a target image from the background, and providing a binary representation of target images in the field-of-view (FOV) of the TV camera.

The second problem concerns the structural characterization and location of target images in the FOV of the TV camera. Projection theory concepts are developed to compute target location and orientation and to describe the structural characteristics of the target images. A method is developed to evaluate the fuzziness caused by switching patterns in the projection representation [9, 21]. The method can be used to obtain a unique representation of a large class of target images using only three projections.

The third problem concerns the development of an intelligent tracking algorithm that measures the significant structural changes in the target image and computes a confidence weight which describes how well the structure of the image located fits the target image being tracked. This confidence weight provides a method to evaluate the truthfulness of the tracking data. Automata theory concepts are applied to realize a tracking algorithm that utilizes the input from the present frame along with inputs from previous frames to define an appropriate tracking strategy.

The fourth problem concerns the development of a control algorithm for predicting the control signals to point the tracking optics toward the target. A method is developed to utilize the structural confidence weight in a manner similar to that of a Kalman gain for combining measured and predicted values, forming a recursive prediction algorithm. Having

the structural confidence weight, the predictor relies more heavily on the predicted values when the confidence weight is low.

The real-time processing constraint has focused the research on image processing concepts that can be implemented with current LSI Schottky bipolar bit-sliced microprocessor technology. Much effort has been devoted to the implementation of the tracking concepts with LSI technology [14, 15, 17, 19, 20]. This effort has led to the design of four high-speed processors which utilize TI's new 74S481 Schottky bipolar bit-sliced processors for implementing the solutions to the real-time tracking problems. These processors are:

- Video Processor
- Projection Processor
- Tracker Processor
- Control Processor

The Video Processor synchronizes and digitizes the video signal from the TV camera, performs a statistical analysis of the digitized image, and separates the target image from the background. The Projection Processor locates the target image and describes the shape of the target. The Tracker Processor establishes a structural confidence in the tracking data and implements an intelligent tracking strategy. The Control Processor utilizes the structural confidence to combine current target coordinates with previous target coordinates to orientate the tracking optics toward the target, forming a fully automatic system.

A detailed interim technical report [11] was submitted to the U.S. Army Research Office in May 1978 that describes the major tracking concepts developed. During the last year of the grant, emphasis was placed on

implementing the tracking concepts developed with current microprocessor technology. This effort resulted in the design of a prototype RTV tracking system that is currently being tested by White Sands Missile Range. To eliminate duplication of the results already published, this report simply summarizes in Section 2 the major tracking concepts developed with ample published references and describes important aspects of the hardware implementation of the RTV tracking system.

A list of the personnel supported and degrees granted is presented in Section 3. A detailed list of the publications that resulted from the research is given in Section 4.

II. SUMMARY OF MAJOR RESULTS

Image processing and pattern recognition techniques have been combined with parallel processing methods to statistically analyze video signals, to separate the target image from the background, to locate and describe the target shape, and to establish the control signals for pointing the tracking optics toward the target image. In achieving a real-time video tracking system it was necessary to solve significant image processing problems as well as develop microprogrammable parallel processing techniques.

2.1 Video Processor

The task of the Video Processor is to separate the video field image into target, plume and background picture elements. The scene in the field-of-view (FOV) of the television camera is digitized to form an n x m matrix representation of the picture P as

$$P = p_{ij},\ i = 1, 2, \ldots, n;\ j = 1, 2, \ldots, m,$$

where the p_{ij} entry represents the pixel intensity at point (i, j). As the television camera scans the scene, the video is digitized at m equally-spaced points in time, corresponding to equidistant points on the horizontal scan of the camera. During each video field there are n horizontal scans which generate an n-by-m discrete matrix representation at sixty fields per second. A resolution m = 512 pixels per line results in a pixel rate of approximately 96 nanoseconds per pixel. The Video Processor receives the digitized video, statistically analyzes the target, plume, and background pixel intensity distributions, and decides whether a given pixel

is to be classified as a part of the target, the plume, or the background.

Tracking Window

The basic assumption of this video-filtering method is that the target and plume images have some video intensities not contained in the immediate background. A "tracking window" is placed around the target image, as shown in Figure 2, to sample the background intensities immediately adjacent to the target images. The window frame is partitioned into two regions, denoted by BR and PR in the figure. Region BR is used to provide a sample of the pixel intensities that are contained in the background part of the scene, and region PR is used to sample the pixel intensities that the plume of the target contains.

The window frame is partitioned as above to accomodate the expected targets which were assumed to be airborne missiles (or airplanes) that might be expected to have a noticeable plume. This consideration is particularly important when video data are available from television cameras which have infra-red sensitivity, since the hot plume of such a target might indeed be more pronounced than the image of the body of the target object itself.

The region within the tracking window (denoted TR in Figure 2) is called the target region. This region contains the entire target object, part of the target's plume (if it has one), and a large area of background data. Region TR is also referred to as the "active" region or the classification region, since the pixel distribution statistics from the BR, PR, and TR regions are used to perform pixel classification only in the TR region. Thus, the end result of the processing is a set of decisions for each window, stating whether the pixels in the TR are members of the

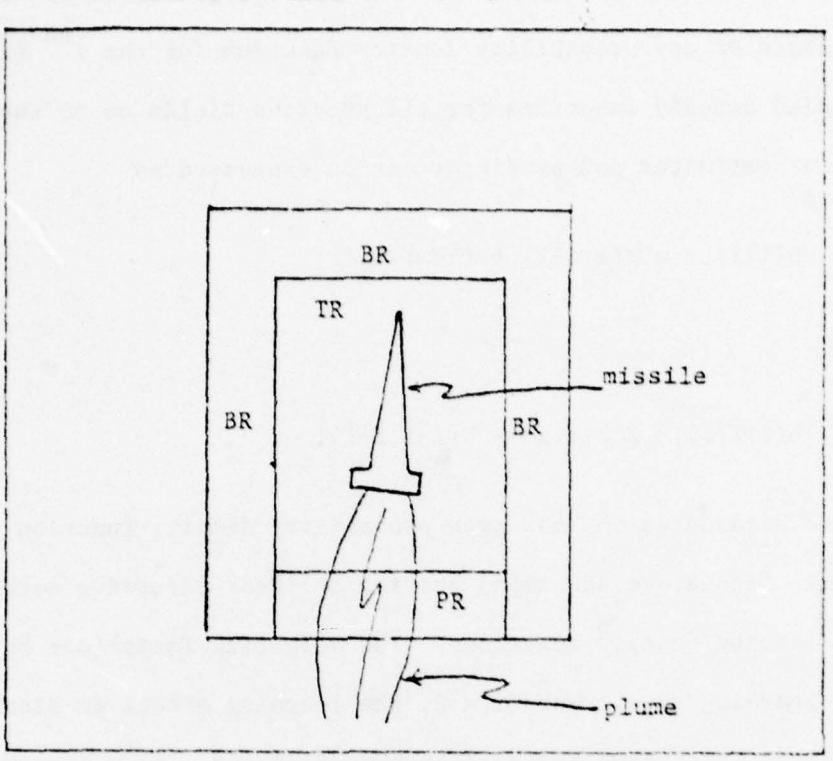

Figure 2. Tracking Window

target, plume, or background elements of the scene.

Learned Intensity Histograms

During each video field histograms are compiled of the intensities in each of the three regions (BR, PR, TR) defined by the tracking window. These histograms are accumulated at video rates by a specialized hardware unit called the Histogram Accumulation Memory (HAM), which uses high-speed memories to realize a multiplexed array of counters. Having field-by-field estimates of the background, plume, and target region probability density functions ($h^{BR}(x)$, $h^{PR}(x)$, $h^{TR}(x)$), a linear recursive estimator and predictor may be utilized to establish learned estimates of the

probability density functions. Letting $h(i|j)$ represent the learned estimate of any probability density function for the i^{th} field using the sampled density functions for all previous fields up to the j^{th} field, a linear estimator and predictor can be expressed as

$$h(i|i) = \omega\, h(i|i-1) + (1-\omega)\, h_i(x)$$

and

$$h(i+1|i) = 2\, h(i|i) - h(i-1|i-1),$$

where h_i denotes the observed probability density function at the i^{th} frame. The above equations provide a linear recursive method for compiling learned density functions. The weighting factor can be used to vary the learning rate. When $\omega = 0$, the learning effect is disabled, and the measured histograms are used by the predictor. As ω increased toward one, the learning effect increases and the measured density functions have a reduced effect. A small ω should be used when the background is rapidly changing; however, when the background is relatively stationary, then ω can be increased to obtain a more stable estimate of the density function.

The predictor provides several important features for the tracking problem. First, the predictor provides a better estimate of the density functions in a rapidly-changing scene (caused, for example, by sudden sun glare on the target). Secondly, the predictor allows the camera to have an automatic gain control to improve the target separation from the background without causing excessive instabilities in the probability density functions. Finally, the a priori probabilities of the states of nature can be estimated based upon a history of statistical information about the scene being analyzed.

Pixel Classifiers

Two pixel classifiers are developed to classify each pixel in the target region (TR) as either target, plume, or background. The first is a very simple to implement threshold decision rule that performs the classification on the basis of background and plume thresholds t_{BR} and t_{PR}. These thresholds are computed on the basis of the noise level and expected target image size. The threshold decision rule decides that a pixel is a target point if

$$h^{BR}(x) < t_{BR} \text{ and } h^{PR}(x) < t_{PR},$$

a plume point if

$$h^{BR}(x) < t_{BR} \text{ and } h^{PR}(x) > t_{PR},$$

and a background point if

$$h^{BR}(x) \geq t_{BR}.$$

A convenient feature of the thresholding decision rule is that the thresholds can be adjusted dynamically on a field-to-field basis to adapt to different tracking environments. Surprisingly good classification accuracy was obtained with this method.

A Bayesian decision rule is developed to classify each pixel in the target region as either target, plume, or background. Using the learned histograms, a method [19, 23] is developed to establish the a priori conditional probability density functions $P(B|x)$, $P(P|x)$, and $P(T|x)$. These density functions are used to implement a standard Bayesian decision rule. The method provides excellent separation of the target image from

the background; however, a considerable computational capability is required to compute the required conditional density functions.

The results of the decision rule are stored in a high-speed decision memory (DM) that is used to make the real-time classifications. Real-time pixel classification is performed by simply letting the pixel intensity address the decision memory location containing the classification. This process can be performed at a very rapid rate.

Storing the pixel classification rule in a memory allows the learning and classification rule construction to be performed simultaneously with the real-time classification process. The only constraint is that the construction of the classification rule must be completed by the beginning of the following video field. This provides the time required to implement the classifier with high-speed bipolar microprocessor components. Furthermore, this allows the flexibility of programmable algorithms to implement the learning classifier.

Video Processor Architecture

The Video Processor [22] utilizes a standard micoprogrammable processor along with a variety of data acquisition and control components to perform the video processing functions. The major components of the Video Processor, as shown in Figure 3, are:

1. a 16-bit microprogrammable bit-slice microprocessor and its associated memories,
2. a Histogram Accumulation Memory (HAM) which generates the intensity histograms,
3. a Region Definition Logic (RDL) module which defines the locations of the tracking windows in the camera's field-of-view,

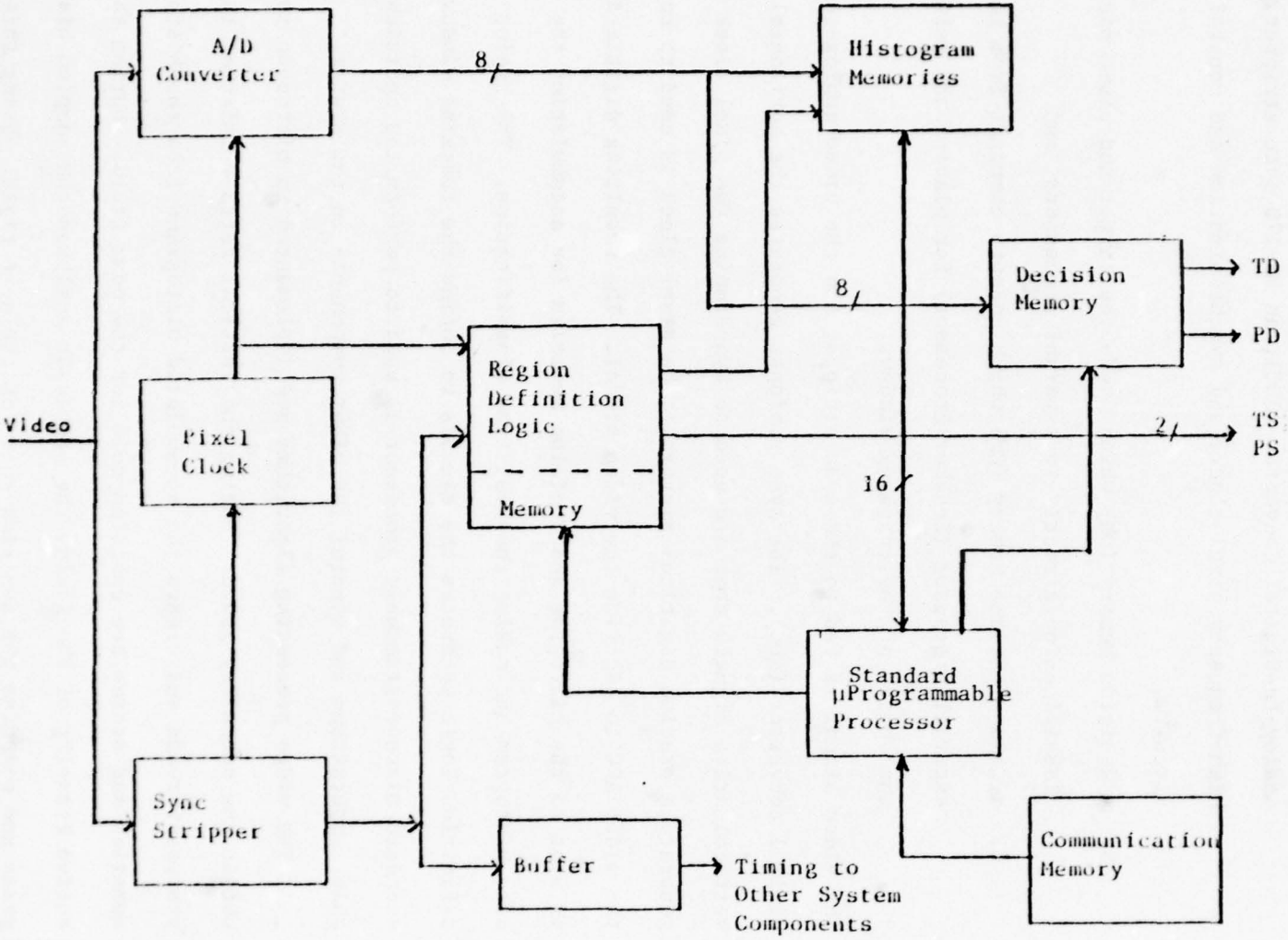

Figure 3. Video Processor

4. a data acquisition and timing package containing a high-speed Analog-to-Digital Converter (ADC), an RS-170 sync stripper, a high-frequency pixel clock, and synchronization and control circuits,
5. a Decision Memory (DM) which feeds the target and plume pixel classification directly to external processors, and
6. a Communications Memory (CM) which accepts commands from an external processor (Tracker Processor) for placing and defining the shapes of the tracking windows.

The video signal is fed to the sync stripper and the video analog-to-digital converter (ADC). The sync stripper generates the horizontal and vertical drive signals that are used to synchronize the pixel clock to generate a precise digitized picture. The pixel clock is used to command the video ADC to digitize the video signal. The resulting digitized video is sent to the histogram and decision memories for accumulating the intensity histogram and making the real-time classification. The region definition logic generates the signals to define the tracking windows. A standard microprogrammable processor is used to perform the decision rule computations and control the other components in the system.

The video processing algorithms are implemented in microcode to obtain the necessary speed. During the vertical retrace interval the processor reads and clears the accumulated histograms from the histogram memories and stores the decision rule for the next field. During the active scanning of the field, the processor analyzes the sampled histograms and computes the decision rule for the next field. Using this technique, the processor is able to perform the real-time pixel classification with a microinstruction cycle time of 200 nanoseconds.

References

The most comprehensive reference for the Video Processor is a dissertation "Real-Time Video Filtering with Bit-Slice Microprogrammable Processors," by Dr. Robert B. Rogers. An interim technical report for Grant DAAG-29-76-G-0231 submitted to the U.S. Army Research Office contains a detailed discussion of the image separation techniques. Several published papers [2, 8, 19, 23] are also good reference for the video processing problem.

2.2 Projection Processor

The task of the Projection Processor is to locate and to describe the shape of the target image. The Video Processor generates a binary picture P for each video field, where target presence at a coordinate is represented by a "1" and target absence by a "0." The target location and structure are characterized by the pattern of "1" entries in the binary picture. To meet the real-time processing constraints imposed by the video rates, the Projection Processor computes two orthogonal projections to locate and characterize the shape of the target image. In general terms, the Projection Processor integrates the intensity levels of a picture along parallel lines through the pattern, generating a function called the projection (See Figure 4). For binary digitized patterns, the projection gives the number of object points along parallel lines; hence, it is a distribution of the target points for a given view angle. These projections represent the signatures of the particular target and allow the target to be precisely located and structurally described.

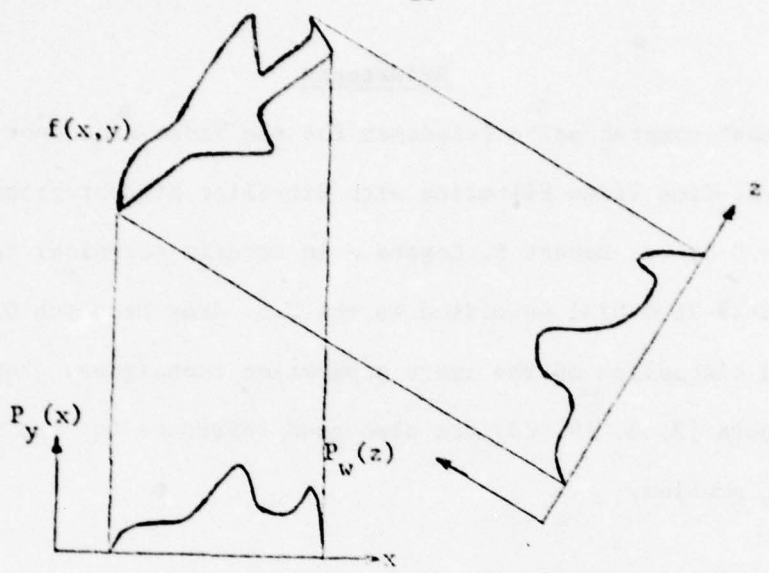

Figure 4. Projections

Tracking With Projections

A projection in the x-y plane of a picture function f(x,y) along a view angle w onto a straight line z perpendicular to w is defined by

$$P_w(z) = \int f(x,y)dw.$$

To precisely determine the position and orientation of a target image, target medians are computed for the top section (X_m^T, Y_m^T) and bottom section (X_m^B, Y_m^B) of the target image as shown in Figure 5. The median Zm of a projection function $P^w(z)$ segments the picture into two equal area parts so that

$$\int_o^{Zm} P^w(z)dz = \int_{Zm}^{Zmax} P^w(z)dz$$

In terms of the target medians, the target location and orientation are given by

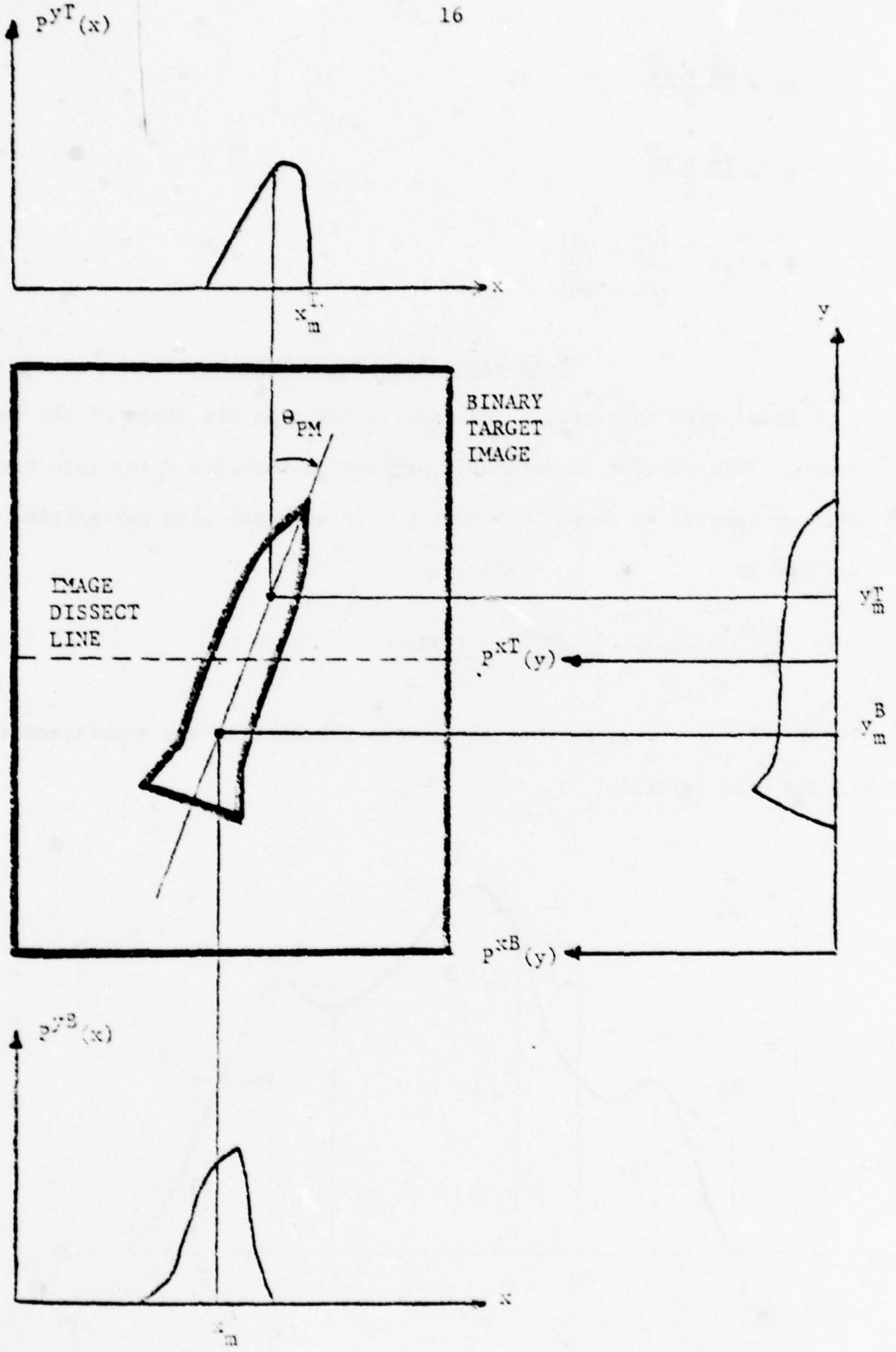

Figure 5. Projection Median Technique

$$Xc = \frac{X_m^T + X_m^B}{2}$$

$$Yc = \frac{Y_m^T + Y_m^B}{2}$$

$$\emptyset = \tan^{-1}\left(\frac{Y_m^T - Y_m^B}{X_m^T - X_m^B}\right)$$

Parametric Structural Model

Equal area percentiles are used to describe the shape of the target image. This process transforms a projection function $P_w(z)$ into k equal area rectangles as shown in Figure 6. These equal area percentiles are defined by

$$\int_{Zi}^{Zi+1} P_w(z)dz = \frac{1}{k}\int_{Z1}^{Zk+1} P_w(z)dz$$

for i = 1, 2, . . ., k. The equal area percentiles are normalized to be target size invariant by

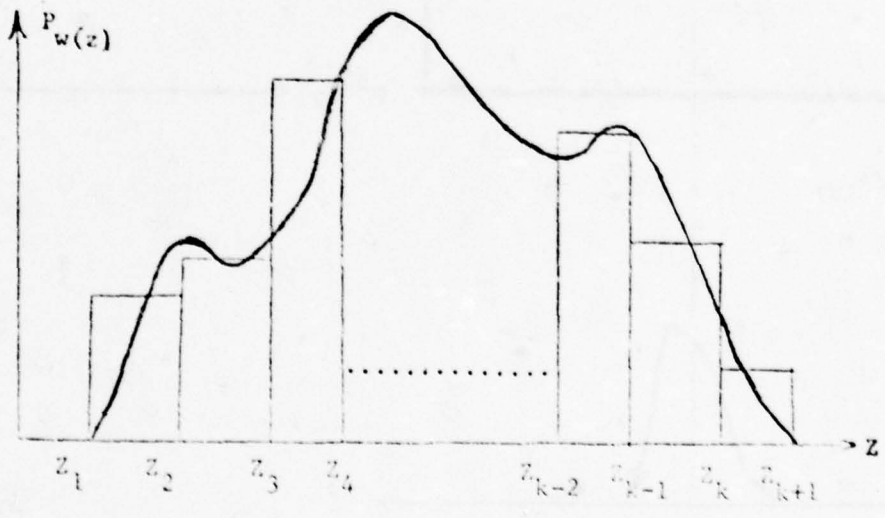

Figure 6. Equal Area Percentiles

$$WTZ(i) = \frac{Zi+1 - Zi}{Zk+1 - Zl}$$

for all i. These are the structural parameters used to locate and describe the target shape.

Projection Processor Architecture

The purpose of the Projection Processor is to reduce the binary images generated by the Video Processor to a parametric model that can be used to implement an intelligent tracking system. The Projection Processor is capable of generating parametric models for two independent images. This is done by forming two orthogonal projections of the target image within each of two tracking windows. Each image can consist of up to 511x511 pixels. The projections along the x-axis are split in two halves at approximately the mid-point of the image to form the top-x and bottom-x projections (See Figure 5). After accumulation of the projections, equal area points for each projection are found. These equal area points are then reduced further to a set of working variables which describe the major characteristics of the image. Although the computations are simple, the amount of data to be reduced is large. At 60 fields per second, there can be as many as 30 million data points per second received from the Video Processor. Therefore, in the design of the Projection Processor, the emphasis is placed on speed.

The architecture of the Projection Processor (Figure 7) has two major components. The first component is a Projection Accumulation Memory (PAM) to compute the projections required to locate and describe the target shape. The second component is a standard bit-slice processor for analyzing the projections to establish the structural parametric model.

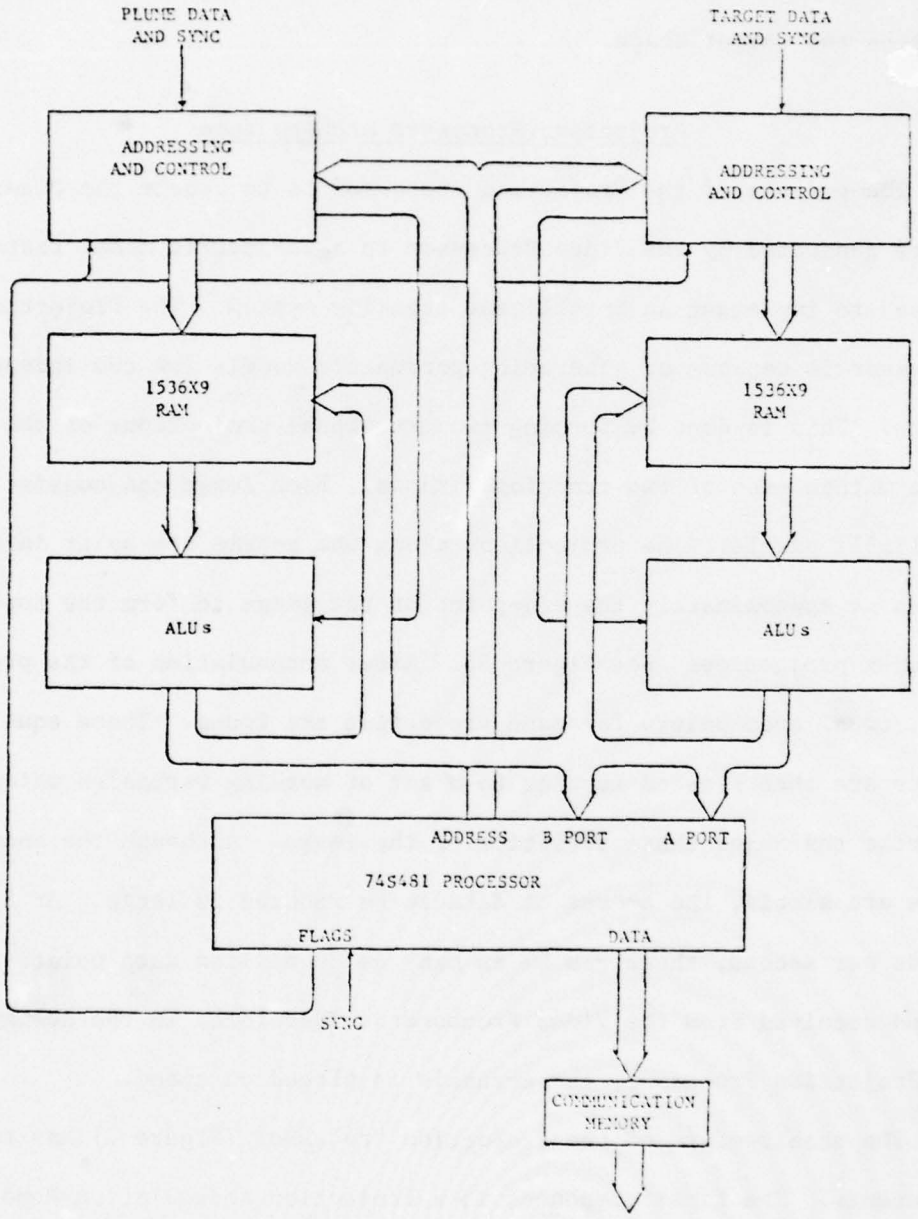

Figure 7. Projection Processor

Separate projection accumulation and processing is used to achieve the speed necessary for high resolution pictures.

The pixel classified binary data from the Video Processor enters the PAM as a serial stream in synchronization with the camera generating the intensity gray-level picture. The projections are formed by the PAM as the data is received, while the processor monitors the y-projections, accumulates the total number of target and plume points, and determines when to split the x-projections. During this time the processor has only limited access to the PAM. In the vertical retrace interval the processor assumes addressing control of the PAM and computes the structural parameters before the first active line of the next field.

The operation of the PAM can best be described as that of an array of addressable counters. To accumulate the x-projections, for example, a counter is assigned to each vertical row of pixels, for up to 511 rows. During the first line of the picture, the previous projection is cleared out by performing a clear on each counter as the pixel corresponding to the address of the counter is received and then conditionally incrementing according to the presence or absence of a target point. For subsequent lines the value stored in each counter is incremented according to the value of the appropriate pixel. Accumulation of the y-projections is similar except that one counter is assigned to each horizontal row of pixels, for up to 511 rows. Thus, after an entire horizontal row has been received, the corresponding counter contains the total number of pixels in that row with a value of one. The addressing of the x-projection counters is controlled during accumulation by a counter which is incremented as each new pixel is received and reset at the end of each horizontal

row, while the y-projection address is controlled by a counter incremented at the end of each active horizontal row and reset at the end of the frame.

Actually using counters to accumulate the projection is impractical. Instead, Random-Access Memories (RAM) and arithmetic logic units (ALU) are used to emulate the counter array by storing the value that would be in the addressed counter in a pre-assigned memory location. The increment and clear-increment functions are realized by reading the memory location corresponding to the incoming pixel and using an ALU to add that value to the pixel value for the increment, or adding the pixel value to zero for the clear-increment, and then writing the results back into the same location. By using a separate high-speed bipolar memory and ALU for each projection, the horizontal and vertical projections can be accumulated simultaneously at a pixel rate of 11 Mhz or more.

The standard processor is microprogrammed to analyze the projections to establish the structural parametrical model. Using a processor with a 200 nanosecond cycle-time, the required computations are completed within one millisecond. Consequently, the projection analysis can be accomplished within the vertical retrace interval.

Much effort was devoted to establish the mathematical foundations and practical implementations for using projections in the tracking environment. A real-time method is developed to measure the fuzziness caused by the ambiguities in the projection representation. To resolve the ambiguity problem associated with two orthogonal projections without accumulating additional projections at arbitrary view-angles, a characterization of the ambiguity problem at the scan line (row) level is

introduced. This characterization forms the basis for the development of a real-time algorithm that identifies the ambiguity introduced when a new line E is added to an unambiguous binary picture matrix, and then modifies a minimum number of entries in E to eliminate the ambiguity introduced. The algorithm can be used to modify a class of pictures, called the pseudo column-connected pictures, so that the original picture can be uniquely reconstructed from only three projections. The algorithm can also be used to dynamically segment a binary picture matrix into vertically contiguous regions, each region being represented by its vertical and horizontal projections. Many new theorems are established for uniquely representing pictures with only three projections.

References

There are several references associated with the Projection Processor. First, the ARO Interim Technical Report [11] presents the basic results for applying projection concepts in the tracking environment. A dissertation [21] "Projection Theory For Real-Time Vision Tracking" by Dr. Yee Hsun U presents a very detailed description of the theoretical aspects of projections in the tracking environment. A masters technical report [20] "A Firmware Module For Projection Computations" by Mr. Steven Szymanski presents a detailed description of the hardware and software architecture of the Projection Processor. Several publications [1, 9, 23, 24] present different aspects of the Projection Processor.

2.3 Tracker Processor

The task of the Tracker Processor is to implement an intelligent tracking strategy for controlling the tracking windows and the tracking

optics. The inputs to the Tracking Processor describe the size, location, dimensions, orientations, density, and shape of the image contained in the tracking windows. The outputs of the Tracking Processor define the location and shape of the tracking window for the Video Processor, and establishes a confidence weight for the tracking data, the boresight correction signals, the image rotation, and the zoom correction signals for the Control Processor. The basic objective of the confidence weight is to recognize fake data caused by rapid changes in the background scene or cloud formations by measuring how well the structural characteristics of the located image match the target image being tracked. When the tracking data is considered false, the confidence weight is reduced and the control algorithm relies more heavily on the previous tracking data to orientate the tracking optics toward the target image.

Intelligent Tracking

The basis of an intelligent tracking algorithm lies in its ability to extract the important information from the input environment, recognize the current state of the environment, and establish an appropriate strategy to effect a desired response from the environment. In the tracking problem, the input environment is restricted to the image in the FOV of the tracking optics as seen by the Projection Processor. From this information, the tracking algorithm extracts the important inputs, classifies the current tracking situation, and establishes an appropriate tracking strategy to control the tracking optics for achieving the goals of the tracking system. The goals that guide the tracking algorithm are:

1. keep the target image within FOV,

2. maintain a desired target image resolution,
3. improve the quality of the tracking data in terms of truthfulness and trackability.

The truthfulness of the target data concerns the reliability of the target input data for extracting tracking information. This quality is characterized by the target shape, density and other physical attributes such as size and dimension. The tracking algorithm can improve on the truthfulness of the target data by controlling the tracking window dimension and shape for the Video Processor, and by adjusting the confidence weight for the Control Processor.

The trackability of the target image concerns the ability of the tracking algorithm to retain the target image within the FOV and to completely enclose it within the tracking window. This quality is measured by the boresight displacements, target dimension, and target image jitter. The tracking algorithm can improve on the trackability of target image by its ability to interpret the current state of the target image within the FOV, by the correct specification of the tracking window location, and by the boresight and zoom correction signals sent to the Control Processor.

It should be pointed out that it is not always possible to define a response (output) for improving both the truthfulness and the trackability. Often, a certain response can effect an improvement on one but degrade the other. For example, improving the trackability of the target image by enlarging the tracking window could result in the failure to reject background noise clutters, thus perturbing the truthfulness of the target position and orientation measurement.

Tracking Algorithm Implementation

The basis of an intelligent tracking algorithm is the ability to establish a tracking strategy to maximize the truthfulness and trackability of the target image. To formulate an intelligent tracking strategy, the tracking algorithm needs the ability to respond to a current input based upon the sequence of inputs that lead to the current state (or situation).

This state concept can be used to classify the tracking situations in terms of state variables as in control theory, or it can be interpreted as a state in a finite-state automaton. The theory of finite-state automata has been successfully applied to implement an intelligent tracking algorithm for the tracking problem.

Some of the advantages of the finite-state automaton approach are:

1. a finite-state automaton can be easily implemented with a look-up table in a fast LSI memory;
2. a finite-state automaton significantly reduces the amount of information to be processed;
3. the tracking algorithm can be easily adjusted to different tracking problems by changing the parameters in the look-up table;
4. the finite-state automaton can be given many characteristics displayed by human operators.

These advantages are important for the real-time tracking algorithm to achieve the desired tracking performance and satisfy the timing and processing requirements.

In a finite-state sequential machine, each state represents the collection of all input sequences that take the machine from the initial

state to the present state. By defining an equivalence relation R on the tape set as

$$xRy \text{ if } \delta(s_o, x) = \delta(s_o, y) \; \forall \; x, y \in \Sigma^*$$

the tape set Σ^* can be partitioned into equivalent classes

$$[x] = s_i = \{y | xRy \; \forall \; y \in \Sigma^*\}$$

Consequently, a state represents all input sequences that produce a given tracking situation. This interpretation of input sequences transforms the development of the tracking algorithm into a problem of defining a finite-state sequential machine

$$TA = (S, I, Z, \delta, W).$$

The states of the machine $S = \{s_1, s_2, s_3, \ldots, s_n\}$ define the different tracking situations that must be handled by the tracking algorithm. The inputs I to the finite-state machine are derived from the image parameters that characterize the size, shape, and location of the present target image. The output set Z defines a finite set of responses that the tracking algorithm employs for maintaining track and retaining high resolution data. The next-state mapping δ: $S \times I \rightarrow S$ defines the next state $\delta(s_i, i_j) = s_k$ when an input i_j is applied to state s_i. The output mapping W: $S \times I \rightarrow Z$ is a Mealy output that defines the proper tracking strategy (response) for each state.

A sixteen state automaton is used to remember the previous tracking situations that led to the current situation. This allows the Tracking Processor to use previous information in deciding on an appropriate

tracking strategy. The finite-state automaton is implemented with a random access look-up table and most outputs are precomputed and stored in look-up tables associated with the tracking states for fast access. Using this approach the Tracking Processor easily meets the real-time processing constraints imposed by the video tracking environment.

Tracking Processor Architecture

The basic architecture of the Tracker Processor (Figure 8) consists of a standard microprogrammable processor, a 2K x 16 working memory, and communication memories to the other processors. The inputs to the Tracker Processor are stored in the communication memory (CM1) by the Projection Processor, and the finite-state tracking algorithm and the output tracking strategies are stored in the working memory. When the Projection Processor sends the tracking data, the Tracker Processor encodes the inputs and combines them with the present state to form the address in the working memory where the next state and output strategy are stored. The tracking window outputs are stored in the communication memory (CM2), and the outputs to the Control Processor are stored in the CM3 communication memory. When these data are stored in the communication memories, flags are sent to both the Video Processor and the Control Processor to indicate that the control data is ready. The Tracker Processor requires approximately three milliseconds to perform these functions. Consequently, the tracking strategy for the (I + 1) field can be established from the inputs up to and including the (I - 1) field during the active I-th field. This pipeline delay gives the tracking algorithm sufficient response time for reliable tracking of high performance targets.

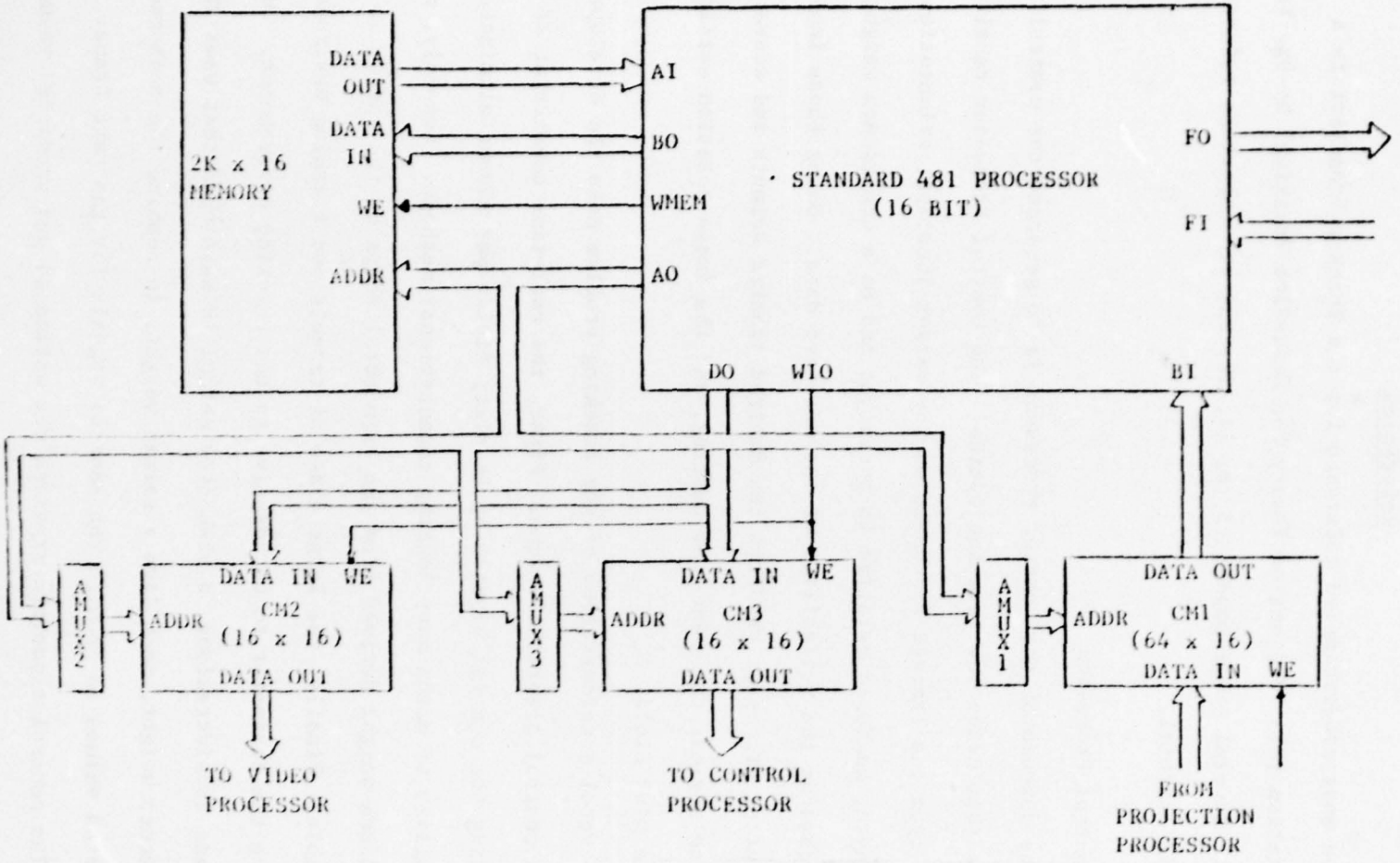

Figure 8. Tracker Processor

References

The most comprehensive reference for the Tracker Processor is a dissertation [31] "Projection Theory For Real-Time Tracking" by Dr. Yee Hsun U. Several other papers [3, 8, 11, 22, 23] are important references for this material.

2.4 Control Processor

The purpose of the Control Processor is to generate the control signals that drive the tracking optics. The Control Processor receives inputs from the Tracker Processor on the target location, orientation, and length, measured relative to boresight and on a confidence weight that measures the reliability of the measured data. Using these inputs, the control processor predicts the desired tracker azimuth and elevation pointing angles, the zoom lens setting, and the image rotation setting for the next field.

Several considerations of the tracking problem make the development of the control algorithm unique. First, the real-time constraint of providing the control signals up to sixty fields per second eliminates the possibility of using many lengthy computational methods. Secondly, a confidence weight derived from the structural shape of the target is available. Finally, the large class of targets and tracking environments require the estimator to be adaptive to the tracking environment. Based on these considerations, a predictor method is developed that uses the confidence weight, much like a Kalman weight, to combine the measured and estimated values to predict the control signals for the next frame.

The control algorithm provides the estimated and predicted values of the variables necessary to orientate the tracking optics toward the target.

Using the confidence weight to combine the predicted variables with current measurements, the algorithm provides estimates of current target position variables. These estimates are combined with previous estimates to predict the target position variables for the next control interval.

Control Algorithms

A simple filter-predictor [10, 11] uses a linear convex combination of a linear-two-point and a quadratic-five-point polynomial filter-predictor. This scheme is based on the reasoning that the linear-two-point polynomial can better anticipate a rapid maneuver from the missile, while the quadratic-five-point polynomial is used to reduce the variance of error. The coefficients of the linear convex combination are chosen to obtain an unbiased minimum variance of error estimate.

In order to evaluate the effectiveness of the polynomial predictor-filter, some simulation studies were made using two considerably more sophisticated filters for comparison purposes. One of the more sophisticated filter predictors considered utilized the famed extended Kalman filter (EKF) in which a nonlinear maneuvering ballistic model is used. These results [10] compared quite favorably.

Control Processor Architecture

The architecture of the Control Processor (Figure 9) includes a standard microprogrammable processor, working memory, and four communication memories. In order to achieve the accuracy and dynamic range required by the control algorithm, the processor has been microcoded with a floating point instruction set. These floating point microroutines

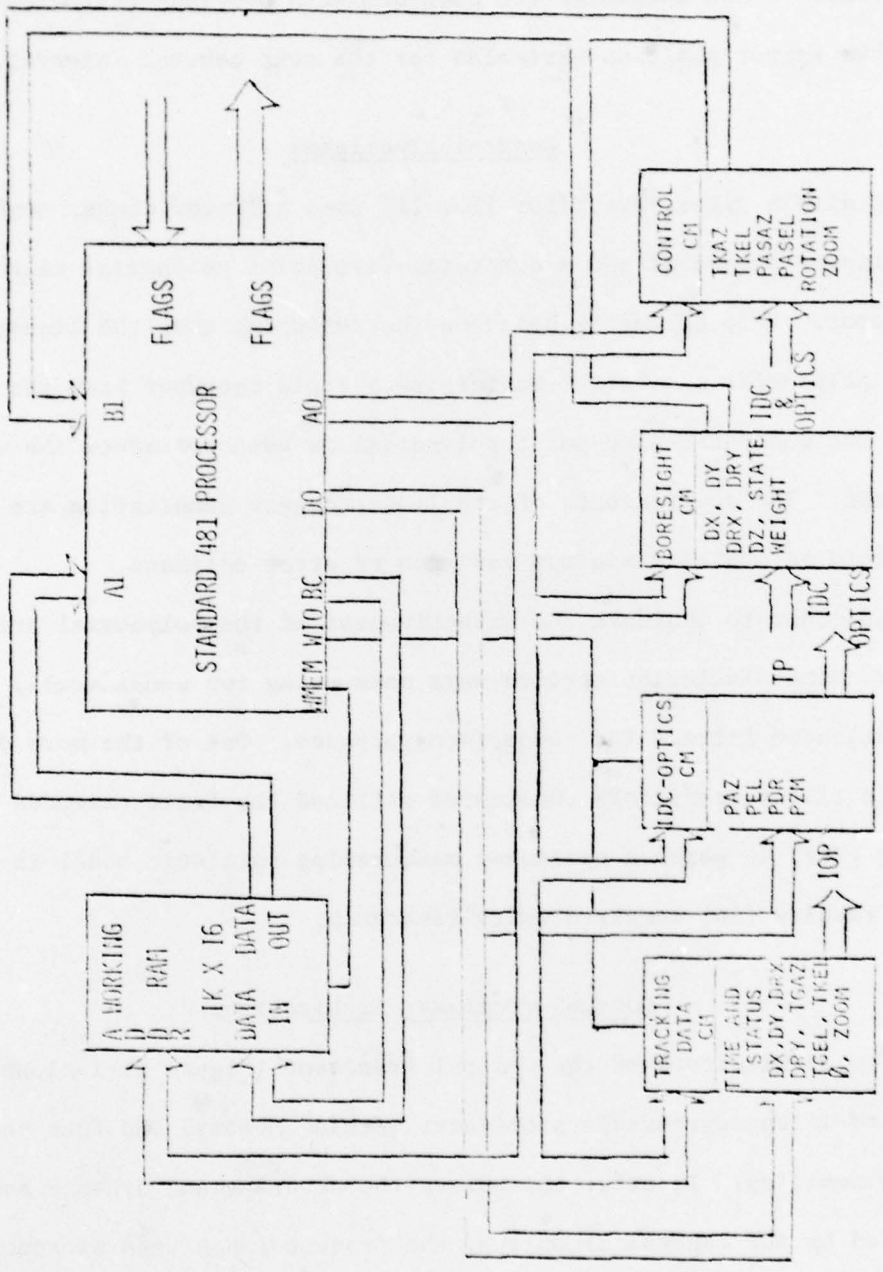

Figure 9. Control Processor

allow floating point computations with 32 bit words. A 1 K x 16 working memory is required for storing the data structure required to realize the control algorithm. The inputs to the control processor are obtained from the boresight communication memory and the optics communication memory. The boresight memory data comes from the Tracking Processor, and it contains the desired boresight corrections, zoom correction, and the confidence weight. The optics memory contains the current optics pointing angles and zoom setting to form a closed loop tracking system. The Control Processor performs the control algorithm and stores the desired controls in the control communication memory. The actual pointing angles, the boresight corrections, the field time, and the tracker status are stored in the tracking data memory so that the data can be stored in the vertical retrace interval on a video tape recorder. Having the recorded video tape of the target flight along with the coordinates of the target image provides an interesting augmentation of current optical film processing techniques.

References

The most comprehensive reference for the Control Processor is contained in an interim technical report [11] submitted to the Army Research Office (May 1978). Several other papers [4, 10, 22, 23] describe and analyze the control algorithm in detail. A dissertation [25] "Some System-Theoretic Properties of Zadel-Fuzzy Sets" by T. R. Kiang contains some applications of fuzzy set concepts to the control problem.

2.5 The RTV Tracking Processor Architecture

The very high-speed computer processing required to carry out the tracking algorithms presented a major research problem in computer

architecture design. The data acquisition rate is too fast and the numerical computations too lengthy to be carried out by any general purpose small-size computer. These considerations created the necessity of utilizing a novel approach in the solution of the problem.

After considering the data rates and computational tasks, a parallel computer architecture was designed to perform the tasks required by the tracking algorithms. The parallel computer architecture consists of four high-speed microprogrammable processors that are programmed with a pipeline architecture so that the major tracking tasks can be performed concurrently. The pipelined architecture provides adequate computational capability to perform the RTV tracking tasks with microprogrammable processors with a microinstruction cycle time of 200 nanoseconds. The cycle time can be achieved with several bit-slice processors currently on the market.

Standard Microprogrammable Processor

A study was made of modern computer technology to select the most appropriate processor for the RTV tracking requirements. To minimize the cost of the system, and also because the individual tasks that need to be performed do not require large programs, it was decided to use bit-slice processors in the implementation of the distributive system. Due to the speed requirements and the very special computations required by the algorithm, the use of high-speed Schottky bit-slice microprogrammable processors is mandatory. The final decision in the selection of the bit-slice microprocessor fell upon the Intel 3002, the Motorola 2901, and the TI 74S481. After evaluating the capabilities of these processors, the decision was made to use the TI 74S481 4-bit processor mainly because

of its strong computational power and its versatile external bus structure.

A general purpose processor architecture has been developed to provide the building blocks of the distributive system. The processor was designed to achieve a microinstruction cycle time of 200 nanoseconds. It has many features to facilitate interprocessor communications. Each processor has a 1 K x 48 writable control store for implementing the tracking algorithms. Having a standard processor architecture greatly simplified the development and maintenance of the system.

Host Processor Development System

A great deal of effort was devoted to the development of a Host Processor software and hardware interface (Figure 10) to assist in writing and testing the tracking microprograms [18]. This effort has led to the development of a high-level micro-assembler for the TI-74S481 processor

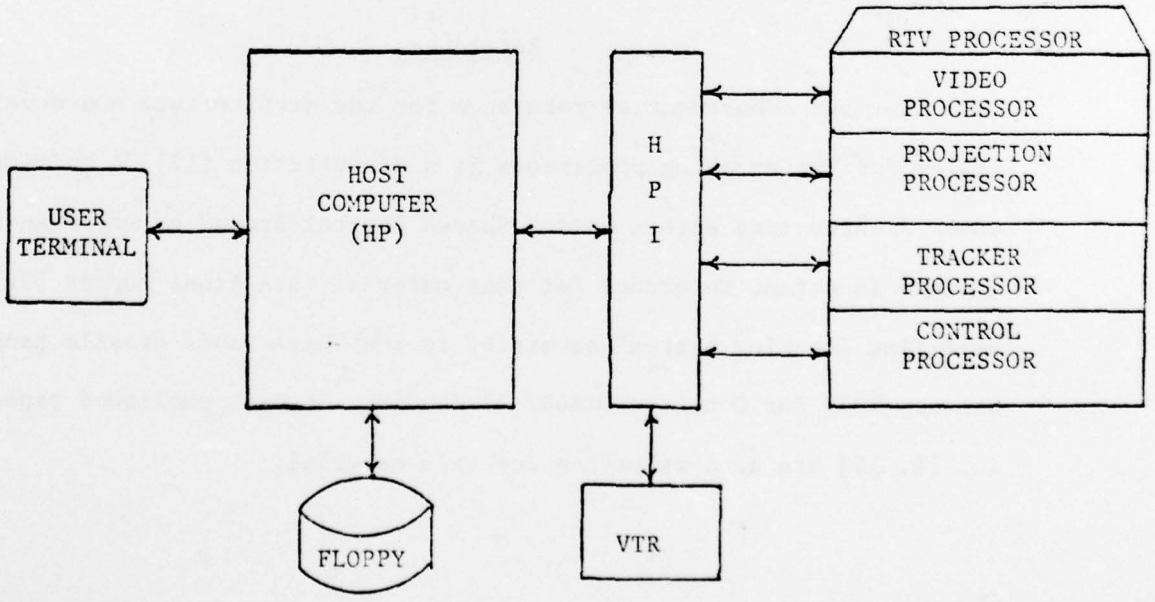

Figure 10. The Host Processor Configuration

[19] which accepts symbolic microinstructions and generates absolute microcode. Furthermore, a Host Processor operating system was developed to allow the user to load, store, and edit microinstructions in the processor control store with our HP21MX satellite computer.

The Host Processor configuration consists of a minicomputer, a user terminal, a floppy disk, and a Host Processor Interface (HPI). The minicomputer provides a programmable interface between the user, the RTV processors, and the VTR system. The user terminal provides standard keyboard entry and CRT display. The floppy disk is required for bulk storage. The HPI maps the writable control stores in the processors and their data structures into the address space of the computer. The Host Processor has the ability to single-step and monitor the performance of any combination of the tracking processors. In effect, the Host Processor provides a development system for the four distributive tracking processors, including a logic analyzing capability for detecting faults.

References

The most comprehensive reference for the architecture and development system for the tracking processors is a dissertation [17] "A Multiprocessor Architecture With A Master Shared Control Store" by Dr. Ivan Perez. Another important reference for this material is a final report [22] "A Real-Time Tracking System" submitted to the White Sands Missile Range in January 1979 for Contract DAAD07-77-C-0046. Several published papers [7, 13, 14, 15] are good reference for this material.

III. Personnel Supported and Degrees Granted

During the three year duration of this grant a considerable number of students were supported, and many of these students received degrees with thesis topics stemming from the research sponsored. A detailed list of the students supported follows.

Name	Degree Granted
Wayne Cannon	M.S.E.E.
Robert Rogers	M.S.E.E. & Ph.D.
Richard Krebs	M.S.E.E.
Otis Soloman	M.S.E.E.
Steven Szymanski	M.S.E.E.
Roger Thompson	M.S.C.S.
Yee Hsun U	Ph.D.
Ivan Perez	Ph.D.
T. R. Kiang	Ph.D.
Rodalfo Gamez	M.S.E.E.
Jerry Chu	M.S.E.E.

Besides the students, two principal investigators, Dr. G. M. Flachs and Dr. W. E. Thompson, were supported twenty-five percent of the academic year throughout the duration of the grant.

IV. Publications

Many publications, dissertations, and technical reports have resulted from the research sponsored by this grant. Several major publications are currently being prepared from the dissertations supported by this grant. These publications will be reported later.

1. 1976 "Structural Feature Extration by Projections," 1976 Region V IEEE Conference Proceedings, pp. 15-19, Austin, Texas.

2. 1976 "A Pre-Prototype Real-Time Video Tracking System," 1976 NAECON Conference Proceedings, pp. 156-160, Dayton, Ohio.

3. 1976 "A Real-Time Structural Tracking Algorithm," 1976 NAECON Conference Proceedings, pp. 161-168, Dayton, Ohio.

4. 1976 "A Structure and Dynamic Mathematical Model of a Real-Time Video Tracking System," 1976 NAECON Conference Proceedings, pp. 169-172, Dayton, Ohio.

5. 1976 "Object Identification with Finite Automata," 1976 NAECON Conference Proceedings, pp. 173-175, Dayton, Ohio.

6. 1976 "A New Concept in Optical Tracking Using Pattern Recognition," Proc. of the International Symposium on Information Theory, June 1976 (Sweden).

7. 1977 "Mathematical Modeling and Simulation in a Programmed Design Methodology," Proc. of First International Conference on Mathematical Modeling, pp. 491-500, St. Louis, Missouri, August 1977.

8. 1977 "An Automatic Video Tracking System," Proc. of NAECON 1977, pp. 361-368, Dayton, Ohio.

9. 1978 "Digital Decomposition and Representation of Video Signals Using Projection Theory," Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 743-746, Tulsa, Oklahoma, April 1978.

10. 1978 "Evaluation of Filtering and Prediction Techniques for Real-Time Video Tracking of High Performance Missiles," Proc. of NAECON 1978, pp. 897-904, Dayton, Ohio, May 1978.

11. 1978 "Real-Time Video Tracking Concepts," Interim Technical Report for Grant DAAG-29-76-G-0231, U.S. Army Research Office, Research Triangle Park, North Carolina, May 1978.

12. 1978 "Video Processing with Microprogrammable Processors," Proc. of the U.S. Army sponsored workshop on Microprocessors and Computer Graphics, Warrenton, Virginia, July, 1978.

13. 1978 "A Tutorial on Distributive Processing Using Microprogrammable Bit-Slice Microprocessors," National Computer Conference 1978, Personal Computing Digest, pp. 377-386.

14. 1978 "Distributive Microprocessor Hardware and Software Architecture for Real-Time Applications," National Computer Conference 1978, Personal Computing Digest, pp. 387-394.

15. 1978 "Using Microprogrammable Bit-Slice Architecture for High-Speed Calculations," National Computer Conference 1978, Personal Computing Digest, pp. 395-400.

16. 1978 "A Real-Time Video Tracking System Using Image Processing," Proc. of the Fourth International Joint Conference on Pattern Recognition, November 1978.

17. 1978 "A Multiprocessor Architecture with a Master Shared Control Store," Ph.D. dissertation by Perez-Mendez, P.I., Department of Electrical and Computer Engineering, New Mexico State University, Las Cruces, New Mexico. 1978.

18. 1978 "The RTV Multiprocessor Development System," Interim Technical Report for Contract DAAD07-77-C-0046, White Sands Missile Range, New Mexico, 1978.

19. 1978 "Real-Time Video Filtering with Bit-sliced Microprogrammable Processors," Ph.D. dissertation by Rogers, R. B., Department of Electrical and Computer Engineering, New Mexico State University, Las Cruces, New Mexico, 1978.

20. 1978 "A Firmware Module for Computing Projections," MSEE Technical Report by Szymanski, S. J., Department of Electrical and Computer Engineering, New Mexico State University, Las Cruces, New Mexico, 1978.

21. 1978 "Projection Theory for Real-Time Vision Tracking," Ph.D. dissertation by U, Yee Hsun, Department of Electrical and Computer Engineering, New Mexico State University, Las Cruces, New Mexico, 1978.

22. 1979 "A Real-Time Video Tracking System," Final Report for Contract DAAD07-77-C-0046, WSMR, January 1979.

23. 1979 "A Real-Time Video Tracking System," Optical Engineering, Vol. 18/No. 1, January/February, pp. 25-32.

24. 1979 "A Real-Time Video Tracking System Using Image Processing Techniques," Accepted for publications in the IEEE Trans. on Pattern Analysis and Machine Intelligence.

25. 1979 "Some System-Theoretic Properties of Zadeh-Fuzzy Sets," Ph.D. dissertation, Department of Electrical and Computer Engineering, New Mexico State University, Las Cruces, New Mexico, 1979.