Electro-optic tracking R&D for defense surveillance

Article *in* Proceedings of SPIE - The International Society for Optical Engineering · January 2000

DOI: 10.1117/12.218219

Christopher Woodruff
The Walter and Eliza Hall Institute of Medical Research

Electro-Optic tracking R&D for defence surveillance

Stuart Sutherland, Chris Woodruff

Defence Science & Technology Organisation, Salisbury, South Australia

ABSTRACT

Two aspects of work on automatic target detection and tracking for electro-optic (EO) surveillance are described. Firstly, a detection and tracking algorithm test-bed developed by DSTO and running on a PC under Windows NT is being used to assess candidate algorithms for unresolved and minimally resolved target detection. The structure of this test-bed is described and examples are given of its user interfaces and outputs. Secondly, a development by Australian Industry under a Defence-funded contract, of a reconfigurable Generic Track Processor (GTP) is outlined. The GTP will include reconfigurable image processing stages and target tracking algorithms. It will be used to demonstrate to the Australian Defence Force automatic detection and tracking capabilities, and to serve as a hardware base for real time algorithm refinement.

Keywords : Image Processing, Target Detection, Tracking, Simulation, Digital Electronics.

1. INTRODUCTION

Enhancement of the surveillance capability of the Australian Defence Forces will be achieved through various technological developments. A central component in this development will be the widespread use of automatic target detection, tracking, recognition and identification based on electro-optical sensor imagery. This paper describes some of the work being undertaken within the Australian Defence Science and Technology Organisation (DSTO) on automatic target acquisition for the enhancement of surveillance.

The genesis of current efforts lies in a body of work commenced in the mid-1980s to develop passive IR missile detection and tracking capabilities for aircraft self-protection suites. In that project an ambitious program of sensor modelling, development of algorithms for the detection and tracking of point targets, and Australian Industry development of real time hardware to demonstrate automatic target detection and tracking (ATD&T), was undertaken. That work reached a major interim milestone in 1991 with field trials in which an IR sensor with a staring array was flown with purpose-built detection and tracking hardware and an extensive set of data collection tools. At that stage personnel movements and Organisational re-structuring saw a hiatus occur in the development plan.

In late 1992 a revised approach to the development of ATD&T capabilities was adopted in which a more generic development of capability was to be addressed. The current DSTO work described here concerns that development. The two major components are a) the development of a versatile code body for studying total ATD&T processing chains on sequences of imagery from either simulations or from real sensors, and b) the development of a real-time hardware implementation of a generic electro-optical ATD&T processor capable of interfacing to a wide range of rectangular scan imaging sensors, able to implement a range of algorithms at various stages of the processing chain, providing "hooks" to give access to data from interim stages of the processing, and able to incorporate a useful degree of data fusion.

We describe the code body in section 2 of this paper, covering both the function and structure of the code, demonstrate the range of options available in the processing chain, and then illustrate the use of the code in a diagnostic application. In section 3 we outline the generic processor structure, discuss constraints on the development, and provide current information on the stage of development. In section 4 we outline the direction of future efforts in both software and hardware development.

0-8194-1911-7/95/$6.00

2. ALGORITHM TEST-BED

2.1. Background and general requirements

The Australian Defence Forces have requirements for electro-optic surveillance and for platform protection To support this requirement a software simulation of a generic automatic target detection and tracking system has been developed. This simulation is written in C++ and runs on a Windows NT/PC or NT/Alpha platform. It is written in such a way as to allow simple modification and addition of algorithms and data paths. Recognising that there is a need for integration of individual sensor units to achieve an optimum balance between performance and covert operation, the code has been designed to allow for multiple data streams at the image and plot level. Input may also come from other tracking systems, such as radar and ESM (Electronic Support Measures). This allows the code to be used to assess algorithms appropriate to the pre-processing stage of a data fusion system, in the terms of Hall[1] and Llinas and Waltz[2], and also for Level 1 processing in these terms. The graphical user interface allows simple specification of the processing chain and easy specification of parameter values for the various algorithms used. Selected intermediate and final images may be displayed and tracker parameters can be plotted as graphs.

2.2. Functional structure

The functional structure of the algorithm test-bed is shown in Figure 1. The image processor takes image data, enhances features of interest and automatically extracts plots from each image. These plots are then processed by the tracking stage to produce high-confidence detected targets with known trajectories. The test-bed allows input of external plot data which can be combined with the image plots. Feedback from the tracking output can be used to control both the image processing and tracking stages. Figure 2 provides more detail of the system. The upper half of the figure represents the image processing and plot extraction components whilst the lower half represents the target tracking process. Multiple image and plot streams can be used prior to the tracking process with simple time-alignment being provided to allow for different frame/scan-rate streams.

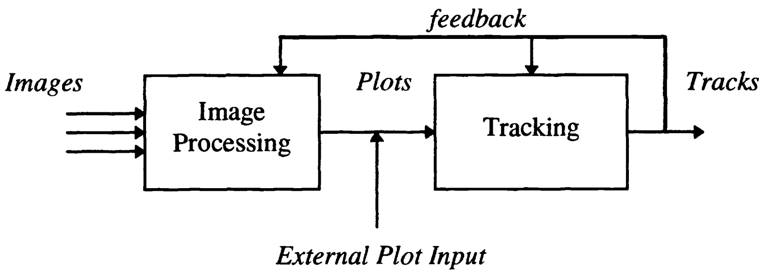


Figure 1- Algorithm test-bed general structure

2.2.1. Data Format

The test-bed accepts as input both raster-format image files and ascii plot files. Image pixels may be bit, byte, word or real. Image size and header size may be specified by the user.

2.2.2. Time Alignment

This process provides for simple time alignment of image and/or plot streams with different frame/scan rates and time offsets. The first stream is used as a reference and is assumed to be the highest frame rate stream. Corresponding frames from other streams are aligned by choosing the frame nearest to the current frame from the first stream. The aligned frames are then processed concurrently.

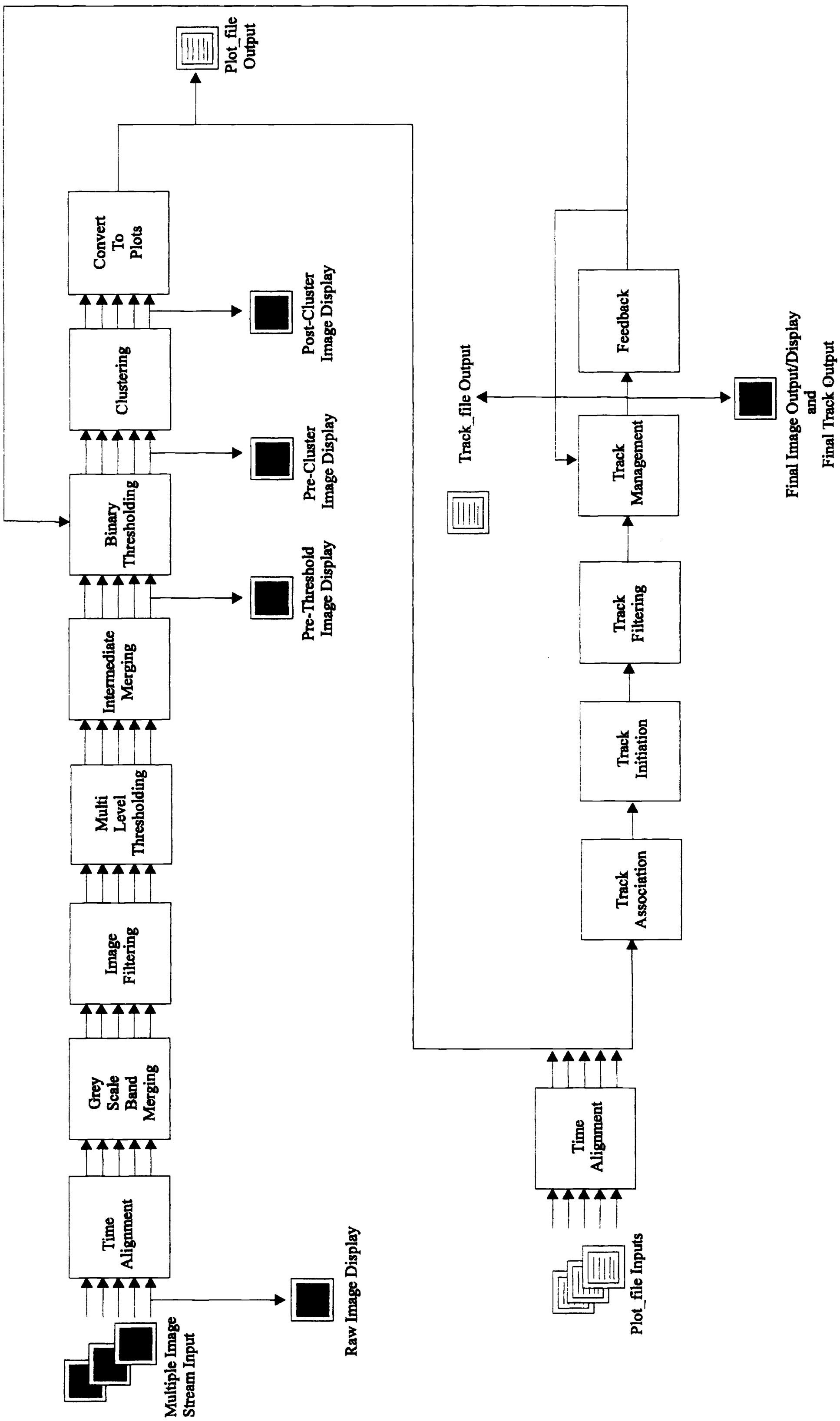


Figure 2 - Algorithm testbed functional structure

2.2.3. Image Processing

This comprises Image Filtering, Stream Merging and Multi-level Thresholding.

The Image Filtering function provides multi-stage combinations of both linear and non-linear filtering operations. Both spatial and temporal filter types are provided. The image filtering stage may be used for feature enhancement and for calculation of image statistics.

The merging operation is available at two points in the test-bed and provides functions to form an output stream from a linear weighted sum or ratio of one or more input streams. Thus:

$$o_j = \left(\sum_{i=1}^{5} a_{ij} i_i \right) + a_{6,j} \tag{1}$$

and

$$o_j = \frac{i_j}{\left(\sum_{i=1}^{5} a_{ij} i_i \right) + a_{6,j}} \tag{2}$$

where o_j is the jth output stream
i_j is the jth input stream
$a_{i,j}$ is the weight for input stream i, output stream j
$a_{6,j}$ is a constant corresponding to output stream j

The stream merging function can be looked upon as a weighted crosspoint switch. It allows arithmetic operations on single or multiple streams as well as stream combination by addition or ratioing. As examples, the first stream merging function may be used for multi-spectral image ratioing or for image masking. The second stream merging function allows the formation of a single stream prior to thresholding or, in combination with the multi-level thresholding function, histogram index generation for histogram-based thresholding. This is discussed below.

The multi-level thresholding function can be thought of as a look-up table remapping function for each stream. It may be set to do nothing (linear look-up) or to reduce the dynamic range of each stream independently. This latter function is crucial to the histogram-based thresholding function described below.

All image processing functions include a user-set gain, offset and bit precision facility. This allows more accurate modelling of real-time image processing hardware which may be limited to integer operations.

2.2.4. Binary Thresholding

This may take one of two forms: simple grey-level exceedance testing or multi-dimensional histogram-based thresholding. The simple grey-level exceedance testing compares each pixel in each image stream with a user-defined binary threshold for that stream. If the pixel exceeds the threshold it is unmodified, otherwise it is set to zero.

The multi-dimensional histogram-based thresholding is based upon the two-dimensional histogram scheme described by Cussons[3] but extended to n-dimensions. From each pixel position x,y in the n image streams a vector $\vec{S}(x, y)$ is formed.

$$\vec{S}(x, y) = \{S_1(x, y), S_2(x, y) \ldots S_N(x, y)\} \tag{3}$$

The N component values of this vector are the pixel values at x,y in the N image streams. From this vector a single histogram index, i, is formed using a mapping function, f:

$$i = f\left(\vec{S}(x, y)\right) \tag{4}$$

where f produces a unique i for each unique $\vec{S}$.

$$\text{for example,} \quad f\left(\vec{S}(x,y)\right) = \sum_{n=1}^{N} 2^{(n-1)p} . S_n(x,y) \tag{5}$$

where p is the bit precision of the equal - precision image streams

The histogram building equation is $H(i) = H(i) + 1$ (6)

After forming the complete image histogram, the histogram occupancy is compared with a threshold value to determine whether the image point corresponding to that histogram bin is set or reset, thereby producing a binary image.

To achieve this function in the test-bed, the multi-level thresholding and intermediate merging functions are arranged so as to convert the pixel values of the N image streams into a composite histogram index. The mapping function is formed as in equation (5), as a summation of individual stream mapping functions, f_n. :

$$f\left(\vec{S}(x,y)\right) = \sum_{n=1}^{N} f_n\left(S_n(x,y)\right) \tag{7}$$

The summation is performed using the intermediate merging function. The multi-level thresholding function provides the individual f_n mapping functions. The value, i, is then used as an index into a one-dimensional histogram. The function f_n is chosen so that the calculated value of i is unique to that combination of x,y,n values. f_n may also include a binning function so as to constrain the size of the histogram in which case there will be a subset of x,y,n values which map to the same i. The multi-dimensional histogram-based thresholding option is provided as part of the binary thresholding function.

2.2.5. Clustering and Plot Extraction

This function identifies 8-connected pixel clusters and allows filtering of these based upon size and shape criteria. The ascii plot files contain all cluster data including frame number, x,y coordinates, pre-threshold mean intensity (of the cluster), raw image mean intensity (of the cluster), cluster dimensions and area. Plot data is saved to file since this may be used to exercise only the tracking functions without having to run the image processing functions again.

2.2.6. Tracking

The tracking functions, shown in the lower half of figure 2, may be run with direct plot data input, with plots generated by the image processing functions or with a mixture of the two. Track and plot data may also include range and intensity information which are treated as independent variables.

The tracking functions associate plots from separate frames to form tracks. The tracks which are generated contain information on detected target positions and bearing rates as well as confidence information. The track parameters may be displayed as graphs or, in the case of x and y, overlayed as cross-hairs on the raw image. Tracking is achieved with four functions: association, initiation, filtering and management.

The association function associates new plots with existing tracks based upon proximity criteria. Specific association options include nearest neighbour, Probabilistic Data Association and multi-hypothesis association.

The initiation function assesses unassociated plots and decides whether they constitute a new track. Initiation options include M out of N and extended M out of N.

Track filtering provides smoothed estimates of the track position and bearing rate as well as predicted position and rate. Filtering options include both alpha-beta and Kalman filters.

Track management provides track confidence, promotion, deletion and merging functions to manage the processing load associated with tracking and allow declaration of high-confidence targets. Options here include simple counter-based as well as Bayesian likelihood schemes and both single and multi-hypothesis track management.

A feedback function allows control of previous image processing and tracking stages on the basis of calculated tracker statistics. Currently this allows modification of the binary threshold and of track management parameters.

2.2.7. Fusion

Since the test-bed can work with a combination of both image and plot data there are a number of fusion possibilities. If radar data is available as registered intensity-only image then this could be read into one image stream. This could then be combined with EO images to allow the streams to be merged in such a way as to allow beneficial fusion at the image level. Alternatively radar plots or ESM measurements could be read-in as plots and fused with EO plots at the tracker association stage.

In this way an input from another sensor system may be used to modify the operation of the EO tracker. Thus the code allows assessment of certain forms of sensor fusion, including, for example, the effect of radar information to enhance passive sensor performance through the provision of intermittent range and range rate information. Specific algorithms to examine data fusion have not been implemented at this stage, but will be incorporated over the next few months.

2.3. Implementation

The test-bed code has been written using Microsoft Visual C++. It is currently running on both Intel 486 and DEC Alpha platforms under the windows NT operating system. Figure 3 shows an example of the program user interface during program execution.

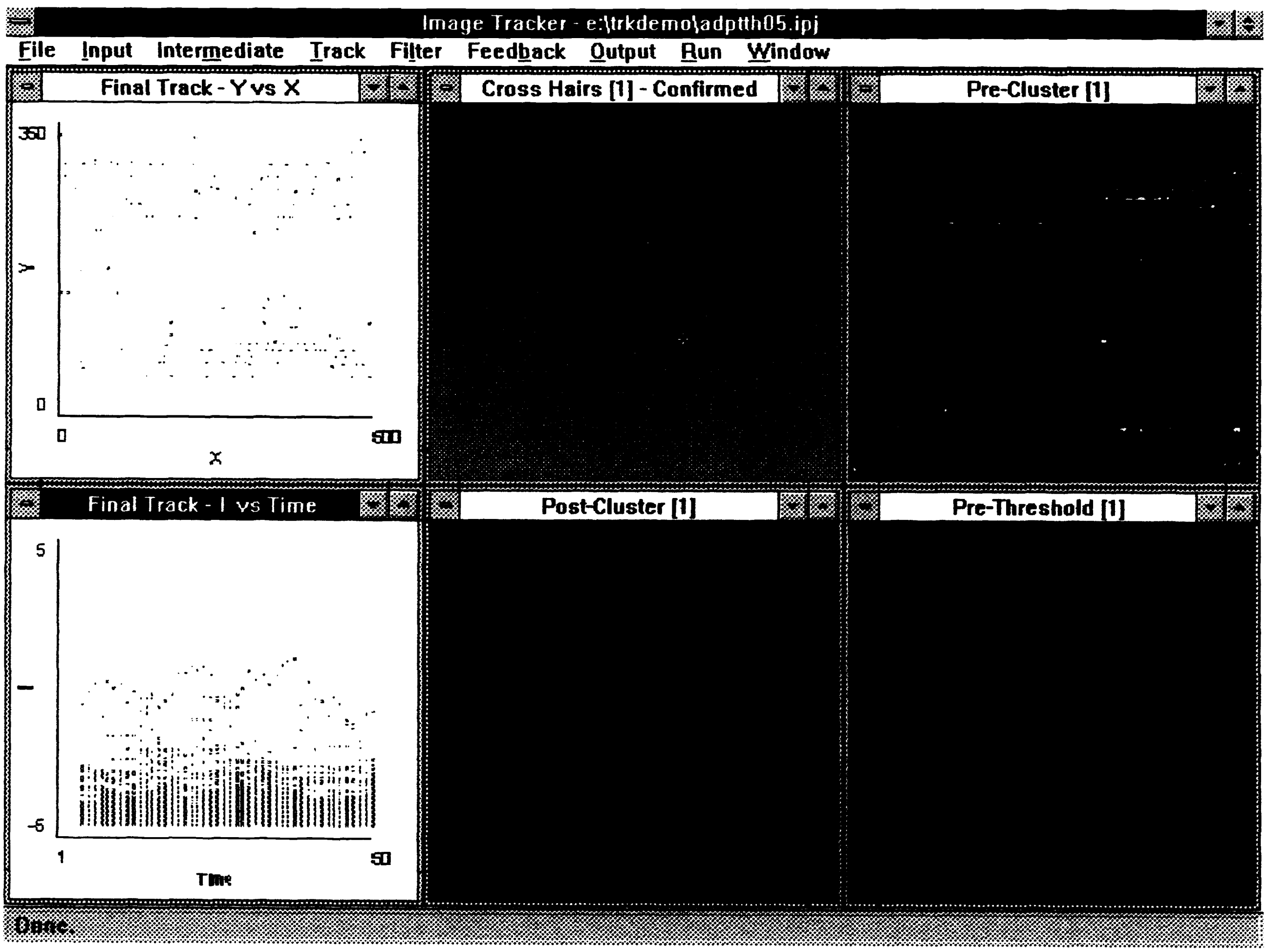


Figure 3 - Test-bed user interface

The program menu structure reflects the functional structure of the code. Under the input menu are submenus for input file specification, time alignment, grey-scale merging, spatial filtering and multi-level thresholding. The intermediate menu provides submenus for intermediate merging, binary thresholding and clustering. The track menu allows association, initiation and management options to be set. The track filtering parameters are set under their own menu. The feedback menu sets the various feedback options and the output menu controls the images and graphs displayed along with their respective parameters.

Figure 4 gives an example of the way in which program parameters are set. The particular example shows the menus used to set-up an intermediate merging operation. The first window (top left) allows the user to select the combination of streams which will contribute to each output stream (from the merge function). The second (bottom left) selects linear or reciprocal merge and the final window allows setting of the merge parameters. An additional option at this stage allows edge pixels to be set to a user-defined value to ensure that undefined pixels do not become supra-threshold after binary thresholding.

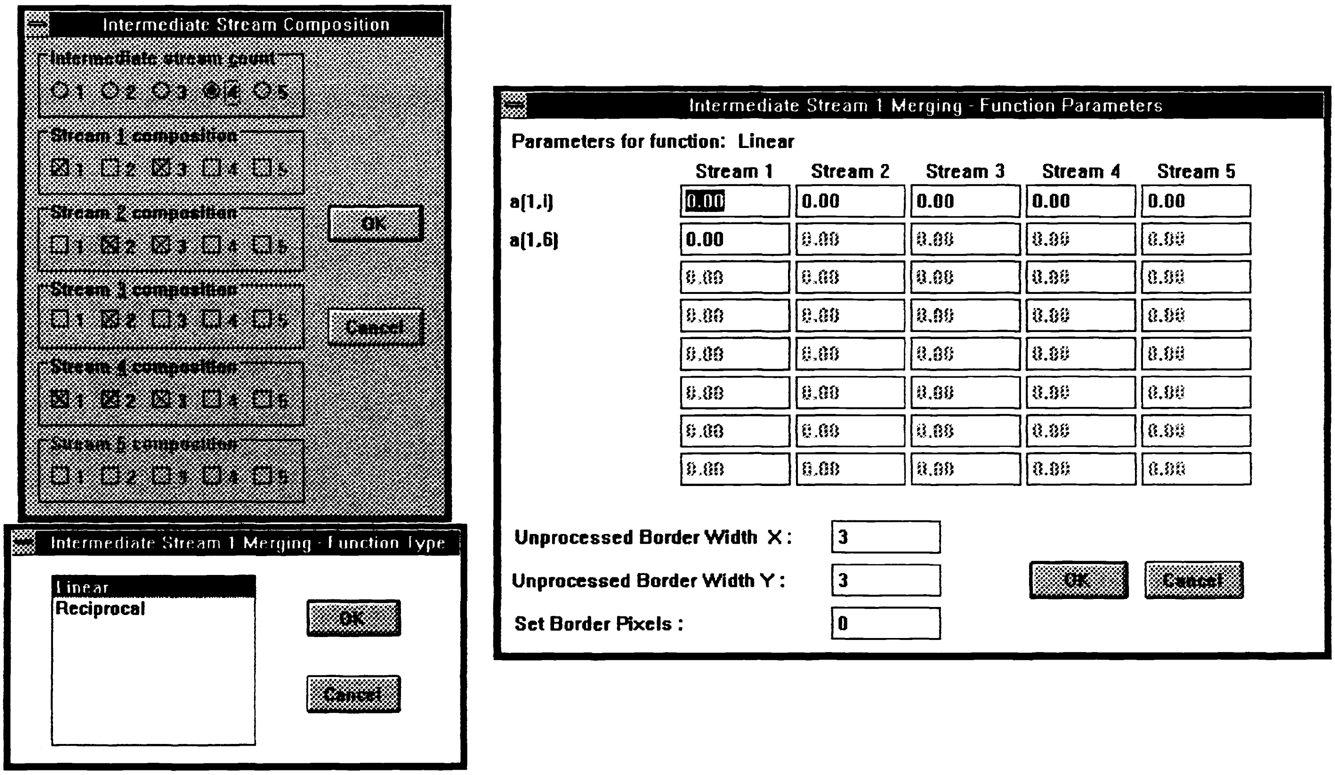
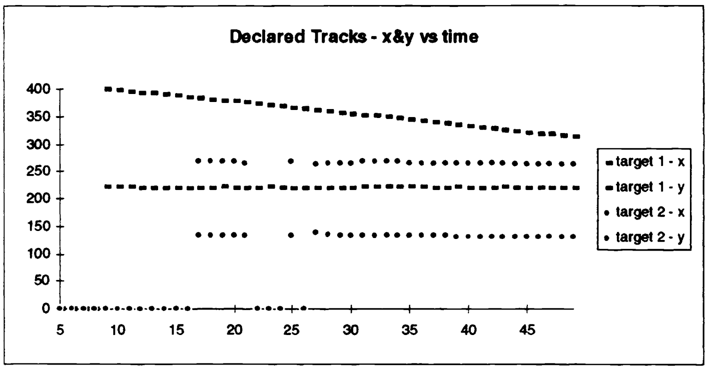


Figure 4 - Example of test-bed user interface - intermediate merging

Other features of the program design include support for cutting and pasting operations on images and graphs and the provision of three levels of diagnostic information which may be written to file for algorithm or code debugging.

2.4. Results

Figure 3 shows the user interface after a typical simulation. Images from various stages of the processing are shown along with graphs of track parameters. Currently the graph plotting routines are quite basic and for more detailed analysis the track output text files are read into a spreadsheet application for data sorting, calculation of track statistics and more elaborate plotting. Figure 5 shows an example of this. The upper graph shows the x and y position of two declared targets plotted against time whilst the lower graph plots the difference between the estimated and measured x and y values for target one in the upper graph. Also in the lower graph trendlines have been calculated using a polynomial fit to the data series'.



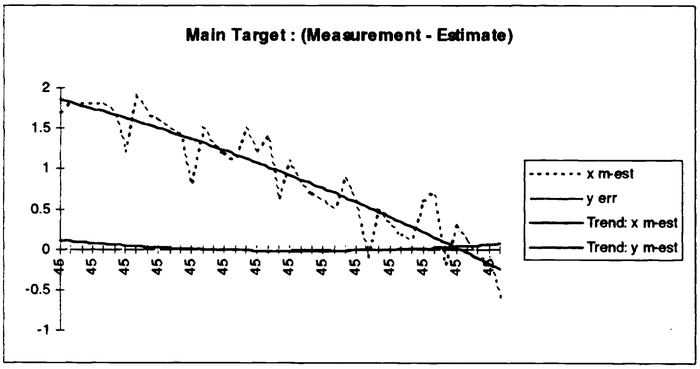


Figure 5 - Example results from testbed track file output.

For a more realistic representation of the likely real time behaviour of the simulated algorithms the output images, with detections superimposed as crosshairs, are compiled into a composite video sequence using a digital video recorder. The sequence may then be replayed in real time.

With reference to the manner of implementation adopted: the choice of an operating system and compiler which are available for a number of platforms and which have built-in graphics features has proved highly successful. The porting of the test-bed code from a 486 PC to a DEC Alpha took less than one hour which represents an enormous cost saving. The adoption of a highly modular and object-oriented implementation has made algorithm enhancements considerably simpler.

3. REAL-TIME HARDWARE DEVELOPMENT - THE GENERIC TRACK PROCESSOR

3.1. Background and general requirements

A key thrust of the EO detection and tracking programme has been to develop a capability within Australian Industry to be a competitive bidder for Defence projects involving the acquisition of advanced EO surveillance hardware, or for the maintenance and upgrading of existing capabilities. This local industry capability must also be able to be demonstrated in the field to a variety of customers, within a time scale of several years. To this end the first of a series of contracts jointly funded by both the Australian Government and Australian Industry is underway. These contracts will seek to produce a real time EO Generic Track Processor (GTP) which will in some ways be a hardware implementation of the algorithm test-bed described above though with somewhat less flexibility. The resulting equipment will be small enough to be easily two-person portable without necessarily requiring the use of application-specific miniaturisation techniques. It is intended that the final hardware will be demonstrated in the field in a variety of detection and tracking roles to several military clients. Figure 6 gives a physical representation of the intended equipment.

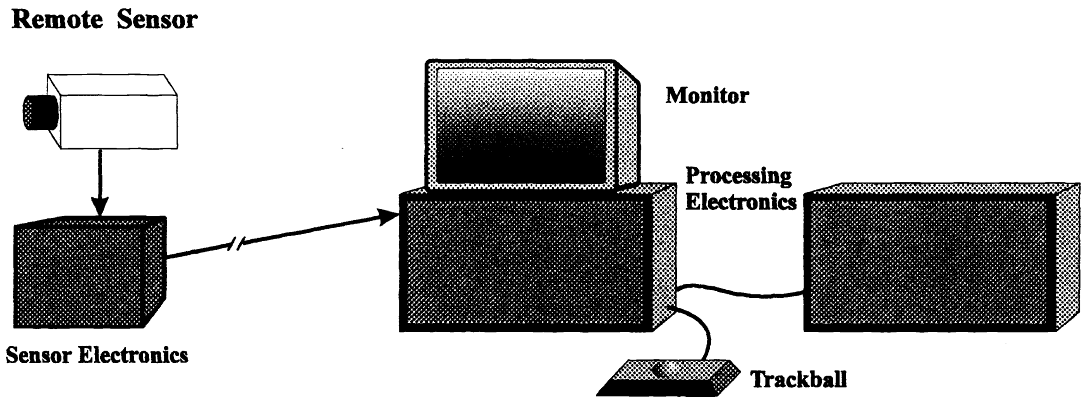


Figure 6 - Physical representation of the Generic Track Processor

3.2. Functional and physical requirements

The GTP system is intended to permit the demonstration of a wide variety of EO detection and tracking techniques for a number of quite different applications. Some of these are listed here:

- Naval Infrared Search and Track Systems using very wide field of view sensors.
- Airborne Missile Approach Warning Systems (MAWS).
- Ground-based Air Defence Systems.

Flowing from these applications are a number of functional and physical requirements:

- Interfacing to a wide variety of sensor data formats and scan patterns for the different applications.
- Pixel data rate of up to 4×10^7 pixels per second.
- A point and extended target detection and tracking capability for land, sea and air backgrounds
- Processing chain reconfigurability and parameter setting to address all applications.
- Data logging capability to allow system performance tuning and analysis
- Provision for different platform power and data interfaces.
- Moderate size and weight to allow portability and demonstration on all platforms.
- Ruggedised sufficiently for trials work though not to military standards.

These are the requirements which, along with financial cost, defined the form and function of the GTP system.

3.3. Functional structure

3.3.1. Data Input

Since the intended applications require the system to be interfaced to quite different sensor types, careful consideration was given to how the GTP would accommodate different data rates and formats. It was decided that the system would not attempt to perform any analogue to digital data conversion since this must be intimately associated with the detailed sensor configuration. Instead it was decided that the system should provide a digital data input interface with a data buffer and reformatter to reorganise the data into a standard format. This will allow acceptance of a wide variety of digital data formats whilst allowing the image processing electronics which follow this stage to be designed to use a single, generic data format. The way in which this is intended to work is shown in figure 7 which illustrates the organisation of the functional components of the GTP. The Data Buffer and Reformatter (DBAR) is able to accept serial and parallel digital signals and includes a fibre-optic interface for applications where the sensor is remote from the GTP. A second DBAR may be configured in a stand-alone mode and located adjacent to the sensor to act as a data transmitter. It can accept data precision's

of up to 16 bits at word rates up to 40MHz. Input scan patterns can be in any direction or line order, including n-order interlaced or non-interlaced. For compatibility with digital broadcast television standards, the DBAR also accepts and generates the SMPTE 259 and 170 standards in PAL and NTSC formats.

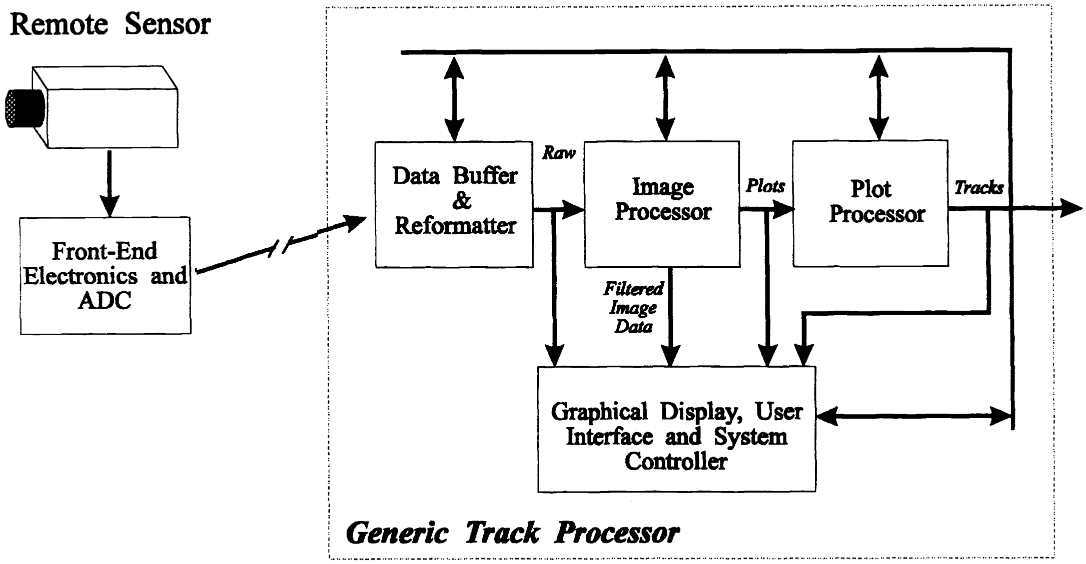


Figure 7 - Functional components of the Generic Track Processor (GTP)

3.3.2. Signal Processing and User Interfaces

The image and plot processing functions perform similarly to the test-bed image processing and tracking functions described above.

The image processor contains linear and nonlinear spatial filtering, feature extraction, a variety of pixel-oriented data stream merging operations (including multiplication and division operations), and binary and histogram-based thresholding. These functions will be functionally separate, allowing any arrangement to form a complete image processing chain. As with the test-bed, the image processor produces plot data which is fed into to the plot processor (tracking function).

The plot processor will support both single and multiple-hypothesis tracking and will include Kalman filtering as well as a final 'target declaration' function to filter high confidence tracks. The graphical display, user interface and system controller will incorporate a real time image display with declared targets and provide both user and developer interfaces to allow control and tuning of the system. It is this component which is most likely to require changing to produce a system suitable for military use. For instance, the airborne MAWS may have no image display facility and no in-service equipment is likely to have a manual algorithm tuning function.

3.4. Project constraints and system limitations

Finance and size have imposed considerable constraints on the achievable goals. Firstly, although the DBAR will accept sensor data at up to 4×10^7 words/s, the image processor pixel rate has been down-graded to 2×10^7 pixels per second. This will allow the use of a variety of specialist image processing chips to be used. The alternatives would have required more extensive use of FPGA's, making the system too big, or the design of ASIC's, which would have been too expensive. The system will still allow demonstration for the applications listed above however, by using the DBAR to spatially window the data.

Only a rather limited capability for sensor fusion has been included - there is no sensor spatial registration capability in the processor though there is a very crude data stream time-alignment function.

3.5. Status

The current status of the development is that the DBAR is an operational unit - with a broad range of applications well beyond those of being a smart front end for the data stream processor. It occupies two standard 6U rack slots and is configured using an IEEE 488 interface. It may be operated in a computer-controlled mode or in a stand-alone mode for remote placement. Frame grabbing and downloading operations are available for all data formats. This provides a powerful diagnostic facility since we may use it to generate test data for real-time playback and to 'grab' in real-time an arbitrary format digital video stream. The DBAR was developed under contract to GEC-Marconi Systems Australia with funding from the Australian Department of Defence.

Contract negotiations for the remaining components are well advanced, but final commitment of funds is not yet guaranteed. It is expected that a decision to commit funds will be finalised this calendar year.

4. FUTURE

4.1. Algorithm Test-Bed development

The tracker code we have developed is already proving to be a valuable tool for studies involving detection and tracking. Major extensions are currently in progress to provide a capability to carry out extensive analyses of fusion with multiple EO sensors and tracking radar. This should be in place by October 1995.

A more extensive track analysis package is currently under development. This will provide track statistics calculations, more extensive graphing and graphical representations of multiple-hypothesis trees. This work will be completed in August 1995.

It is intended that the test-bed used for studies involving laser radar as well. Since plot range is already included as a tracked variable the required extensions to the test-bed for this use will not be great. It is not likely that the test-bed would be extended to study track-to-track fusion.

The code structure is amenable to dealing with higher level classification - for instance, an individual plot is actually a vector with values assigned to each of a set of parameters(vector components), such as azimuth and elevation angles, signal amplitude, range and size. The track processor then operates on this plot data primarily by applying temporal processing to it. Since the existing code structure allows for both histogramming operations and multi-level thresholding it is possible to develop plots which are actually feature vectors consisting of a wide range of types. The temporal processing (tracking) could be replaced by a statistical or model-based classifier to provide a code body for studies on automatic target recognition.

The possibility of providing a near real-time image sequence playback facility on the test-bed is being investigated. Since images may contain in excess of 0.5M pixels with frame rates of up to 30Hz this will probably utilise MPEG or fractal-based video compression techniques.

4.2. Hardware developments

With respect to hardware developments, it is premature to plan significant developments until we have experience with the processor currently planned. However the likely next step would be to incorporate more sensor fusion capability at both the sensor data level and at the track level, and also to address extended target recognition Depending on the developments in optical processing technology for target recognition it may be appropriate to implement complementary approaches to recognition - such as model-based techniques - in digital electronic hardware.

5. CONCLUDING REMARKS

We have presented software and hardware developments in the area of automatic target detection.

The implementation approach taken with the algorithm test-bed - utilising a platform-independent operating system (Windows NT) and a compiler with built-in user interface features (Microsoft Visual C++) - has proved to be an economically efficient one with platform porting taking less than one hour. The provision of clipboard interfaces allows data from the test-bed to be used with more sophisticated analysis applications running on the same platform.

The modular approach taken with the test-bed has consistently made the algorithm upgrade process a simple one and it is hoped that the same modular approach being taken with the Generic Track Processor will prove likewise.

The Data Buffer and Reformatter has allowed the GTP image processing components to be independent of sensor format. Additionally it is likely to be an invaluable interfacing and diagnostic tool for any digital video system.

6. REFERENCES

1. D. L. Hall, *Mathematical Techniques in Multisensor Data Fusion*, Artech House, Norwood MA, 1992

2. J. Llinas & E. Waltz, *Multisensor Data Fusion*, Artech House, Norwood MA, 1990

3. S. Cussons. "Automatic Acquisition of Aircraft Targets using Infra-Red Imaging Sensors", *IEE International Conference on Advanced Infrared Detectors and Systems*, London, pp. 131-138, October 1981.