AN ATR SYSTEM USING AN INTEGER BASED CORRELATION ALGORITHM IN A TIME VARYING ENVIRONMENT

Mr. Carlos Maraviglia
US Naval Research Laboratory
Advanced Techniques Branch, Code 5752
maraviglia@nrl.navy.mil

Mr. Jamie Price
Naval Research Laboratory
Advanced Techniques Branch, Code 5752
price@nrl.navy.mil

Dr. Thomas Taczak
Envisioneering, Inc.
taczak@ninja.nrl.navy.mil

ABSTRACT

In an electro-optical or infrared tracking system, a correlation algorithm offers a robust tracking technique in a time varying scenario. Described in this paper is an Automatic Target Recognition algorithm that employs an operator in the loop as an embedded tracking system. The system described will combine morphological algorithms with an efficient integer based correlation track algorithm. This algorithm will be explored in a time varying search and track scenario using morphological matched filtering to automatically detect and select objects with a handover to a correlation tracker. Development of this algorithm using real and synthetic imagery will be reviewed as well as some preliminary results.

1.0 Introduction

Described in this paper is an Automatic Target Recognition (ATR) algorithm whose initial conditions are defined by an operator and one where the tracking of selected objects is embedded in hardware and software image processing technologies. Using a morphological filter that hands over a selected object to a correlation algorithm the ATR allows a simple integer based system to efficiently use modestly capable computing resources in a time varying environment. Included is a description of how synthetic imagery was used to speed up development of the system. The system and algorithm outlined is one that is currently implemented by the NRL Foxtrot simulator.

1.1 Overview of the hardware

For some time the Advanced Techniques Branch of the Tactical Electronic Warfare Division at NRL has been developing and operating Foxtrot, a programmable real-time video tracking system. The goal of the system is to provide a platform that can be used to develop, study, and simulate various tracking algorithms. The system was designed as a research tool with various imaging front ends and sophisticated image processing capabilities. Configured as either a closed loop or man in the loop system two configurations of Foxtrot are maintained, a flight ready version (Figure 1) and a field portable version.

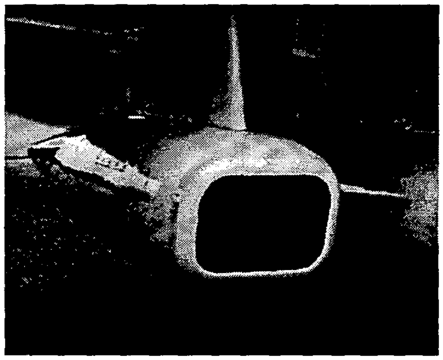

Figure 1: Foxtrot Pod

Foxtrot is composed of several sub-systems. These include a commercially available 3-5μm Infrared (IR) imager, a video image processor and analysis computer, a gyro stabilized gimbal with control loop electronics, data recording devices, and a standard PC for general I/O. One of several different IR imagers along with a standard TV camera can be installed on the gimbal. Each of the IR imagers has a staring focal plane array (FPA). Currently all processed outputs from the IR imagers and TV camera are in the standard RS-170 video format.

1.2 Overview of image processing flow

Starting with the IR energy entering the imager a scene is generated and put into a video stream. The video signal undergoes automatic gain control (AGC) twice, once in the imager at the detector and then again in the image processing hardware. Entering the image processing system the video signal is streamed into a pipeline path that will create a binary image (based on a dynamic digital threshold). Further image analysis is done to produce parameters for groups of contiguous pixels, blobs, for each frame. An accumulated history of blob frame parameters is collected and then used to decide if there are viable blob targets to track. Blob target ratings are generated using their size, width to height ratio, rectangularity, and persistence (cycle count). Information about the location of the highest rated target is then passed to the decision electronics to ensure target retention and to optimally point the gimbal around the chosen target. If no viable targets are found the system will return to a predetermined search pattern.

Various system parameters are overlaid on top of the tracking video for operator feedback and for post analysis of system performance. Included is information such as imager settings, gimbal angles, operational mode, track window, and the primary and secondary target positions.

1.3 Overview of time varying image-processing algorithm

The system collects imagery that varies over time, typically that seen from an aircraft flying in towards a ship. When such approach imagery is used different image processing techniques become more or less effective in the selection and retention of targets. Therefore a short time after a primary target has been acquired the tracking algorithm is switched to ensure a robust lock on. Assuming track is not lost, the algorithm is only switched the one time for the duration of the fly-in scenario. A typical timeline is illustrated in Figure 2.

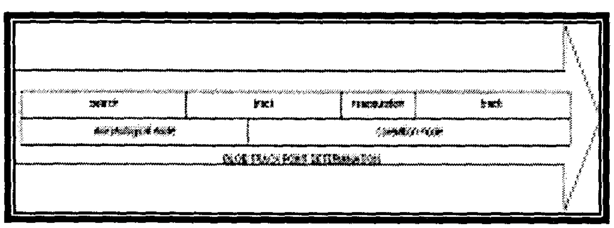

Figure 2: Track Time Line

First a selected object must meet certain criteria under the morphological algorithm while the system is searching. After the system has been tracking the select object for a short time, the system switches to the correlation algorithm. Both of these modes are implemented in a hardware set of image processing cards that use integer-based math. While floating point would offer some advantages, this integer-based method is computationally efficient and was used in the embedded system with previous generations of processors.

2.0 The tracking algorithms

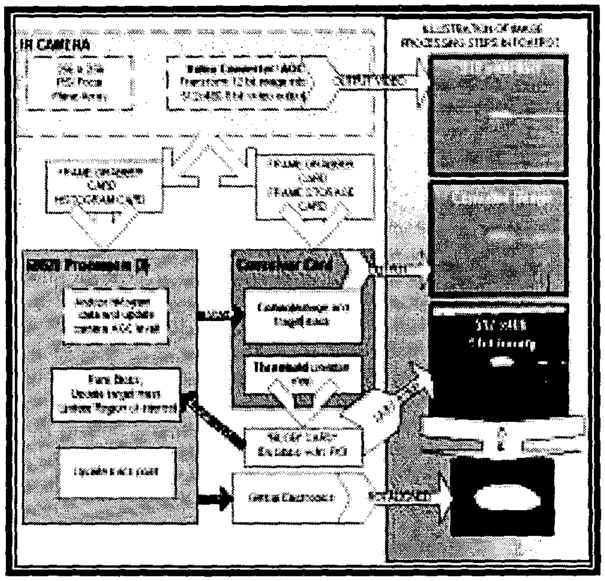

Figure 3: Flow of data through FOXTROT

Implementation of the tracking algorithm can be divided into image processing and target determination. This division is illustrated in Figure 3. Division lines fall within the hardware/software of the multiprocessing of the three 68020 Processors. Embedded in the IR camera, frame grabber, and image processing card blocks of Figure 3 is the image processing /camera AGC (hardware) part of the search/track algorithms. Target determination algorithms in the processor block use reduced image data to determine a primary track point and to feedback correlation masks to the image processing hardware.

Only objects in the search window can be tracked. Since valid ship targets would be located below the horizon all data above the horizon is ignored by all image-processing routines except in the correlation mode. Though the horizon is automatically determined, the operator can specify or offset the row number where the cutoff will occur. Figure 4 illustrates the search path and the region in which the image processor looks for valid objects of interest. A sample target mask is shown to indicate the relative size of the reference kernel.

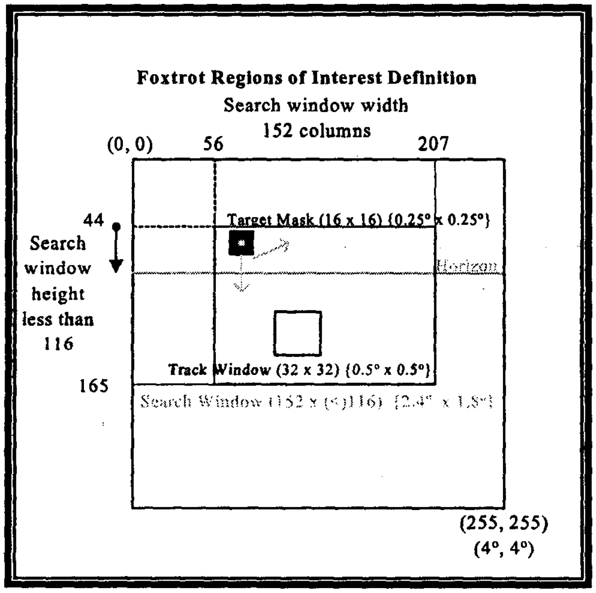

Figure 4: Image Search Pattern

2.1 AGC algorithms

Imager level is dynamically adjusted using feedback from the image processors and is done on a frame-by-frame basis at a 30-hertz (Hz) update rate.

Hardware or imager AGC is performed on the video signal in the following operations. First the image is digitized with the Frame Grabber cards. A 256 x 240 image is generated with 8-bit (256) quantization depth per pixel. The image array is then passed to a histogram card where a count of the number of pixels in the upper and lower halves of the 8-bit color range is done. If the sum of pixels in the lower half of the quantization range exceeds 75% of the total number of pixels ((256 x 240) * 0.75 = 46080), then the camera level setting is decremented by 1. Conversely if the sum of pixels in the upper half of the quantization range exceeds 75% of the total number of pixels, then the camera level setting is incremented by 1. (Note that external control of the imager level is limited to 0-256 in increments of 1).

One artifact of this algorithm is that the initial level setting greatly affect where this algorithm stabilizes, which in turn dramatically affects the contrast of the scene. Poor level settings would result from the camera looking at a bright sky within a given area and then being pointed to a darker region. The initial level condition would be relatively high, but then the level would be adjusted down leaving most of the scene excessively bright.

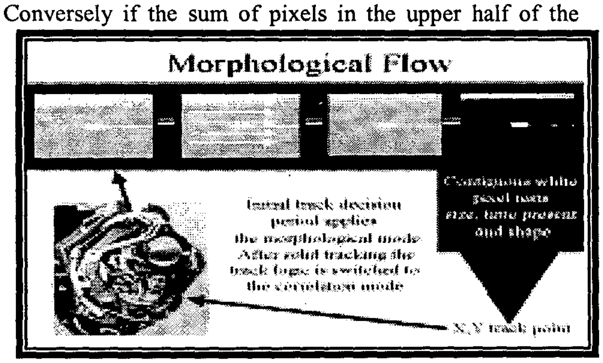

Figure 5: Morphological Flow

2.2 Morphological tracking algorithms

Initial image target processing is based on standard morphological processing techniques. At the beginning of a run the operator chooses from a predefined filter, target mask (kernel). That shape is sought in the scene collected from the imager using convolution techniques. The morphological flow is outlined in Figure 4.

2.3 Correlation tracking algorithms

A Correlation Tracking algorithm is based on a one-pass convolution of the gray scale image with a target mask reference matrix. In the correlation mode of operation, the target mask is automatically updated using data from the highest ranked target in the scene at a rate of once per second. The cross-correlation matrix, resulting from the convolution, is set to a threshold to produce a binary image where only regions of the image with high correlation to the reference mask are evaluated (Figure 6).

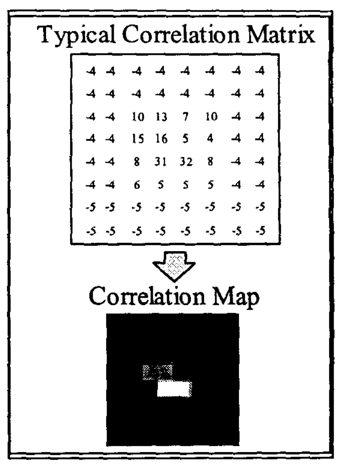

Figure 6: Typical Correlation Matrix

Again details of the flow of data through the Foxtrot system are shown in Figure 3. From Figure 3 it can be seen that the imager produces a stream of 512 x 480 (nominally) 8-bit images converted to an analog video form in a standard RS-170 format. Analog video data is then passed to the Frame Grabber card and digitized at a 30Hz sample rate with 8-bit color depth. Next the image is down sampled from a 512 x 484 to 256 x 242 image using a frame storage card. During this conversion the interlaced video signal is converted to a sequence of non-interlace images. Each image in the sequence is then passed to a convolver card that calculates the convolution of the current "target mask" with each image. The convolution is performed over the entire image, except edge pixels.

The convolution involves the calculation of the cross-correlation coefficient between the reference mask and the region of image overlapped by the reference mask. This mask is then moved across every pixel in the image and the cross-correlation coefficient is recalculated. Equation 1, as implemented in FOXTROT, defines the calculation of the cross-correlation coefficient. This calculation is performed at every pixel (I,J) in the image. Where the sensitivity is a scaling factor. (I,J) defines the current location of the

$$C_{(I,J)} = \sum_{\text{Each pixel in the target mask}} \mathit{Sensitivity} * \frac{\text{Image}_{(K,L)} * \text{Mask}_{(i,j)}}{256} \qquad \textbf{Equation 1}$$

mask in the image, (i,j) defines the position within the mask, and (K,L) define the pixels in the image overlapped by the (i,j) pixel in the reference mask.

One selected object mask in FOXTROT is a 16 x 16 kernel that contains a 8 x 8 center region with data describing the current selected object and an outer ring containing normalization terms. A sample selected object mask is shown in Figure 6. Since currently calculations are limited to 8-bit integers, normalization terms in the outer ring ensure that over-flow errors do not occur during the processing. Overall the integer arithmetic helps reduce the processing time.

Before any output is sent from the convolver card it is passed through a thresholding algorithm. As an index value the threshold algorithm uses the integer value resulting from the convolution to reference a value in a 1 x 256 dimensional (1-D) look-up-table (LUT). For this particular FOXTROT correlation tracker, a value of 0 is assigned to all convolved pixel values of 16 or less. All pixel values greater than 16 are set to a full-scale value. The resulting image is a binary image. An example of the binary image is shown in Figure 3.

The binary image is passed to an image-processing card that groups contiguous grayscale pixels into objects called blobs. This "blob" information is used to analyze the key geometric features of all contiguous regions of pixels. The output from this image analysis card is a table of data on each blob, including the number of pixels in the blob, the width and the height of each blob (referred to as window-x and window-y, respectively), and the horizontal and vertical centroid of each blob.

For each blob object that is located within the track window, its respective data is then used to calculate a rating. Each object is scored based on the size of the blob (# pixels in blob, $\mathbf{P_b}$), the fill factor of the blob (# pixels/ Area of rectangular region subtending the blob, $\mathbf{P_a}$), the width-to-height ratio ($\mathbf{R_{wh}}$) of the blob, and the length of time the blob has existed. C_1 and C_2 are two weighting constants that emphasize horizontal objects. A "blob score", $\mathbf{B_S}$ is calculated using Equation 2.

For this implementation the Blob size is significantly larger than the other two factors. Therefore producing a large blob in the track window is the principal requirement for generating a larger blob target score. All of the blob target scores are then ranked compared to each other. The centroid of the highest ranked blob then becomes the location for the center of the track window. Any error between the current pointing vector of the gimbal and the new calculated track- pointing vector is used to generate the error signals that drive the gimbal to its updated position. At a rate of once a second the 8-bit values of the pixels surrounding the best blob (±4 pixels about the centroid) are used to generate a new target mask.

3.0 Summary of system use of algorithms

This section summarizes the coordinated operation of the Foxtrot system as a whole. The algorithms described thus far are assembled together and placed into Foxtrot and have been operated and tested under various conditions. Over time they have been optimized at field trials for use in maritime environments searching for ship targets. Several parameters may vary significantly over the time period of a single run.

The dynamic nature of flying offers many problems to overcome. These challenges include the change of aspect angle over the period of a run, different atmospheric transmission conditions that can vary in a run or day-to-day, variable target sizes and unknown target ranges. All these challenges introduce different image processing needs. Foxtrot (the algorithms and system described) attempts answer to the challenge of all of these parameters and their effects to produce a reliable programmable system.

3.1 Summary of system target approach

Before a test run the operator initially tunes the system to use the proper matched filter. Choosing the proper filter is usually based on past experience with the weather conditions, aircraft approach angle, and estimated range to target. Predetermined and entered into the system are the scan pattern and system scan rate. Once the proper parameters are selected the system is set into an automatic search mode. No other operator action should be needed to scan, acquire and then track a given target. Of course the goal is to observe if the system with its set system parameters behaves according to the expected.

The system scans and processes the scene looking for possible targets that meet requirements. Once potential targets are found the system starts determining object value. Value or rating is done by looking at the associated processed contiguous pixel object (blob) for simple geometric properties, temporal factors and spatial relationship to the scene. The blob that rates the highest triggers the start of track and determines the track point. Four seconds later the system hands over track to the correlation mode. Starting with the image provided under initial track and the features of the prime target the correlation mode maintains track for the duration of the approach or run.

$$B_S = P_b + C_1 * P_a + C_2 * R_{wh} \qquad \textbf{Equation 2}$$

After correlation mode starts the correlation mask is periodically updated. A more robust correlation mask results from the target growth and increased resolution that occurs with the approach to the target. The system typically maintains a strong track until some point at close range. Unfortunately during the later part of the final approach the 8-bit integer method used in processing presents some difficulties. At this time further work is still ongoing to ensure the maintenance of a strong track.

3.2 Advantages of track algorithm handover

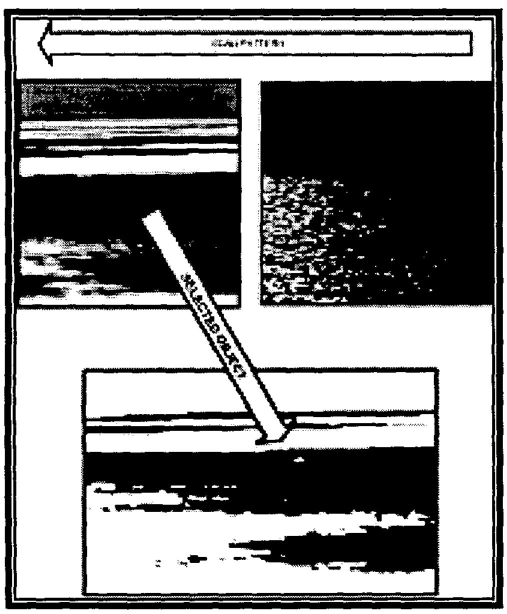

Figure 7: System performance

These algorithms offer different strengths at different times and different conditions. Figure 7 illustrates that the scan pattern in a very challenging infrared scene. The selected object is embedded within the dark patch. On using the morphological filter, with luck, the desired target will be selected. After a brief time of track, the correlation filter will then be used. The advantage in the algorithm handover in this situation is demonstrated by the fact that a few seconds later the sea-glint pattern changes to another different challenging pattern. The handoff improves the capability to maintain track on the desired target instead of being lost in the noise or background.

3.3 Synthetic Imagery used to develop dynamic system

In the past Foxtrot algorithms were developed using real time imagery or that which was collected in the field and played through the system at a later time. To enable testing with a wide variety of scenarios, Foxtrot algorithms are also being developed and tested with synthetically modeled imagery. Modeling is being developed to provide the dynamic environment and well featured scenes that will reduce future development and study costs. Data from past field-tests aid in the creation of modeled imagery. For verification of the modeled scene fidelity, it is necessary to collect more field data to increase the overall database. Field trials will continue to be used in the development and testing of tracking algorithms along with being use to verification of modeled data.

4.0 Conclusion

The use of multiple tracking algorithms presents difficult challenges when integrated into a real system and used in a continuously dynamic linear environment. Overcoming the difficulties encountered when switching tracking algorithms offers the advantage of applying the better algorithm at the most appropriate time. Knowing well the environment that will be encountered and the overall system response is essential in good tracking algorithm development. In time, better, faster hardware will increase the limits of sophistication over the current software thus enabling a system to truly behave in the desired fashion at the desired time. For this application the ideal system would be an integrated system that could be setup to track in a prescribed fashion looking at various types of targets under varying environmental conditions. Ideally all of that would have to be done with minimal notice, at little cost, and completed in a short amount of time. At this time we will have to be satisfied with deriving, testing, and analyzing the strength of different tracking algorithms with the system we have. We are confident these efforts will make our next generation programmable tracker significantly more robust.