INFRARED AUTONOMOUS ACQUISITION AND TRACKING

Teresa L.P. Olson and Harry C. Lee
Teresa.l.olson@lmco.com (407) 356-7109
Harry.c.lee@lmco.com (407) 356-6997
Lockheed Martin Missiles and Fire Control – Orlando

1.0 Abstract

In order to demonstrate lock-on-after-launch (LOAL) capability, imaging infrared missile systems of the future require the ability to autonomously identify and track targets of interest. A robust algorithm architecture must have the flexibility to accommodate fluid system requirements driving its design. This paper describes a method to autonomously acquire and track an extended range target through its whole flight scenario. A proven ATR approach is used to detect and identify targets of interest, separating them into non-targets and clutter. The methodology uses a down selection strategic to nominate targets for terminal track. Once nominated, a Weighted Edge Tracker is employed. The tracker relies upon correlations of appropriately weighted edge directions from frame-to-frame images and reference templates. This combination of automatic target acquisition and terminal tracking provides a sophisticated yet simple approach to many challenging long range tracking problems.

2.0 Goal

An automatic target acquisition (ATA) algorithm was developed to demonstrate the capability of unaided target detection and identification followed by hand-off to a terminal-tracking algorithm used in closing missile sequence. The tracking algorithm, a Weighted Edge Tracker (WET), makes use of the targets external and internal structure where applicable. The ATA is an algorithm suite which employs an edge/intensity based pre-screen algorithm proceeded by an in-depth feature analysis of all qualified target candidates. Features are the traditional ones used in similar system in use today. While the joint algorithm has demonstrated success, there is a great deal more refinement needed to make it production ready.

3.0 Acquisition Algorithm Design

The algorithm for automatic target acquisition (ATA) can be separated into following parts:
1. - Pre-screener,
2. - Feature analysis, and
3. - Classifier.

The pre-screener acts as a target filter nominating candidate target locations. Candidates that make the cut are processed by a segmentation routine. A feature set is calculated for each of the residue candidate targets using the segmentation as a delineation of each target's boundary. At this step in the process, the candidate targets have been reduced to a simple set of features that describe the nomination's characteristics.
Then a classifier processes each of the candidate targets feature sets and determines which are the desired targets and those that are not.

A sequence of imagery is being presented to this ATA and nominations are being made at frame rate. Advantage is taken of this process to improve the classification performance in a function called Temporal Target Analysis.

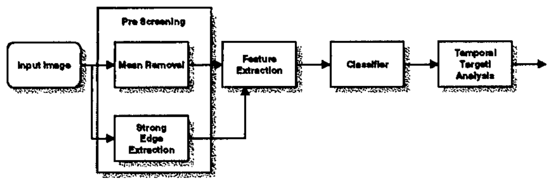

Figure 1: Automatic Target Acquisition Algorithm

3.1 Pre-Screener

The pre-screener extracts candidate target segments by analyzing their contrast and edge strength relative to the immediate background. To achieve this goal the local contrast must be computed. This operation is achieved by local mean removal filtering. The resulting image from this filtering is subtracted from the original image leaving only these regions where the contrast is above or below the local mean average. These residual components make up the image's high frequency contrast elements. By using a fixed threshold for both the cold and hot targets in the image a candidate target segmentation is achieved.

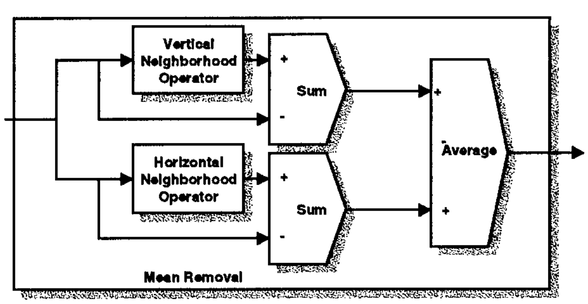

Figure 2: Mean Removal

Figure 3 shows the Strong Edge Extraction algorithm, which is applied to the whole image and will reveal all of the regions of rapidly changing contrast. For its algorithmic efficiency, the Sobel operator was selected. The Sobel determines the image gradient that contains both magnitude and direction at each pixel in the image. To expedite the operations to follow, only the local maximum gradients are retained. This process is achieved using a novel thinning approach based on gradient local pixel neighborhood elements. All edges that are eliminated due to thinning are removed from both the magnitude and direction edge map. Finally, global thresholding removes any stray edges that may cause problems which are attributed to low contrast background or terrain clutter.

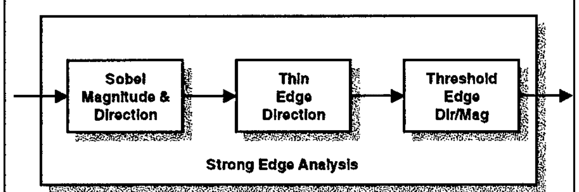

Figure 3: Strong Edge Extraction Routine

At this point in the process the candidate segments from the mean removal algorithm outputs are combined with those from the strong edge routine. A decision process using these two outputs determines which segments contain enough unique edge properties to make the cut. All other segments are eliminated in this processing step. The segments that remain represent the candidate target segments of interest and are passed on to the feature analysis step.

3.2 Feature Extraction

All segments nominated as candidate target objects are parameterized in the Feature Extraction algorithm. Features categories included in the candidate target extraction process are as follows:

1. Contrast
2. Gradient Magnitude
3. Gradient Direction
4. Spatial Extent

There are bounds put on the complete set of extracted features and if one or more is exceeded that particular candidate target is eliminate from further processing.

3.3 Feature Classifier

For this application a 2-class system (target and non-target) Bayes Classifier is being utilized. This classifier makes the assumption that the feature density functions are multivarient Gaussian. Two sets of mean and covariance matrices were built using a sufficient set of ground truthed data set images. Given this knowledge, each feature vector is tested to determine which class it most likely to be a member. This is accomplished by determining the highest conditional probability for the particular feature vector. The candidate target is classified into the i^{th} class where d_i the likelihood is the highest value.

$$d_i = -\frac{1}{2}\ln\left|\Sigma_i\right| - \frac{1}{2}\left(\overline{x}-\overline{\mu_i}\right)^T \Sigma_i^{-1}\left(\overline{x}-\overline{\mu_i}\right) + \ln\{P(\omega_i)\}$$

where Σ_i is the feature set covariance matrix for class i, $\overline{\mu_i}$ is the mean feature vector set for class i, and $\overline{x}$ is the feature vector of the segment. If $d_t > d_{nt}$ then $\overline{x}$ is classified as having similar features as target class t, else $\overline{x}$ is classified as having similar features of objects in non-target class nt.

3.4 Temporal Target Analysis

Once all segments in the frame are classified, the targets are tracked by using a temporal inertial tracker. Segments that have been classified as a target are assigned track id, pixel location, life count, and history. Life count represents the number of frames the track has existed and history represents how the segment was classified during the last ten frames.

All targets in each frame are associated with the current track data. If a target does not associate to a particular track, then it is considered to be a new track and is assigned a unique track id. If a track does not associate to a target in the current frame, then the track "coasts" for that particular frame. By examining the life count and history of the tracks, it is possible to maintain lock targets even if the feature classifier does not classify it as a target 100% of the time.

3.5 Example Results Of The Automatic Target Acquisition Suite

Shown in Figure 4 are the results of running the ATA on an image containing multiple armored vehicles. The ATA nominates three targets in the scene and indicates this by drawing a box around the targets. Also, indicated in the scene are some nominations that passed the pre-screener and were removed by the classifier. They are indicated as red points in the image.

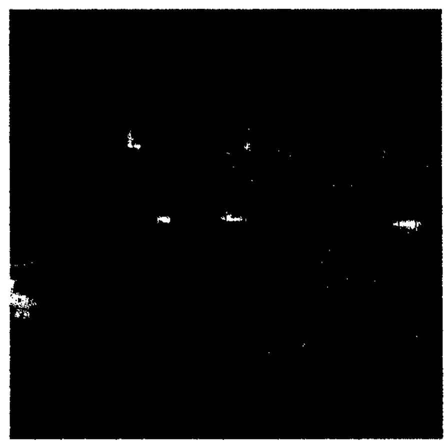

Figure 4: ATA Example Results

4.0 Weighted Edge Tracker

The ATA described above comprise half of the total algorithm suite, the second half is devoted to the introduction of a novel tracking algorithm design utilizing gradient direction based correlation.

4.1 Concept

The Weighted Edge Tracker (WET) algorithm was initially developed for the non-armored tracking case although it has been proven to work just as well for armored target scenarios. It was designed to keep track on buildings or various other urban warfare targets. These targets presented difficult cases due to the fact that they have a great deal of similar structure. This algorithm augments a typical contrast correlation scheme by correlating on appropriately weighted target edges. This eliminates many traditional correlation problems, in particular sliding along strong straight edges.

4.2 Algorithm Design

The WET involves correlating template images with a search window in order to track targets. Relying on cleaned edge directions makes the matching metric less sensitive to contrast than conventional techniques. A key strength of this tracker is its use of weighted edge directions correlation. This allows the algorithm to place equal importance on edge directions occurring several times or a just a few times. This process leads to a stronger tracking lock and is less susceptible to tracker drift. Figure 5 shows WET's top-level flow diagram.

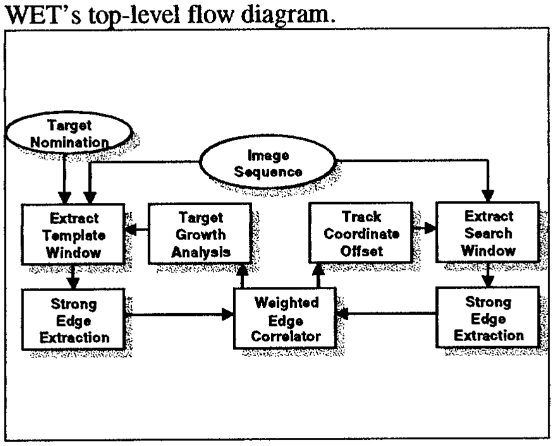

Figure 5: WET Algorithm Description

The first process involves creating a trackable target template. On the first frame of data a region is nominated around the target to be tracked. A template is created from this region along with the calculation of the initial track coordinates. The template is processed by the Strong Edge Extraction routine to obtain robust edges to be used in the WET. A search window of imagery is extracted from the next frame around the track coordinates. This window location is augmented by information from the image plane motion relative to the inertial coordinate system. This delta is used to get the best seed point to start the searching process.

After receiving both processed images, the template and search windows of the WET determines the new track coordinate offset to be used in subsequent frame.

4.3 Strong Edge Extraction

The Strong Edge Extraction routine shown in Figure 6 produces a set of edge magnitudes and their associated directions. The Sobel operator accomplishes this task in the search window. The window is sufficiently sized to encompass the target and some predetermine background percentage.
Or in the non-armored case, it can be a structural piece of the target that should be locked on throughout the tracking process. The edge magnitude and directions are thinned which retains the strongest edges. This in effect skeletonizes the window region and retains more reliable edges that in the correlation process. A threshold is selected and applied to the edge magnitude fixed percentage of the edges are retained. Strong Edge Extraction is used on both the template and the search window.

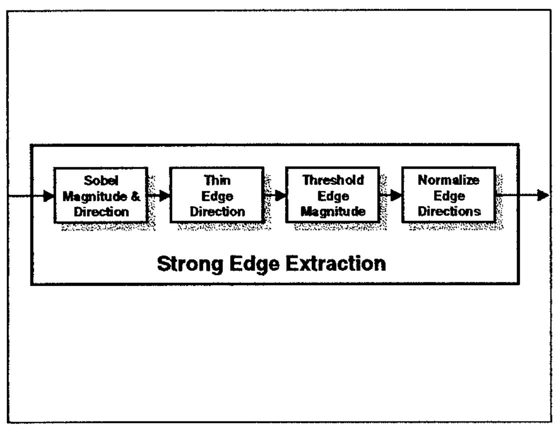

Figure 6: Strong Edge Extraction

4.4 Weighted Template Correlator

Figure 7 shows the correlation process. This involves comparing the template and the search window direction images.

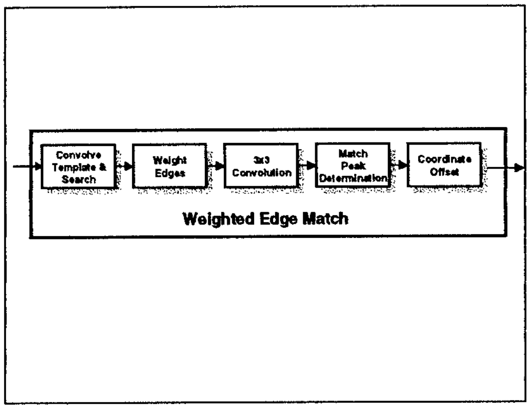

Figure 7: Weighted Edge Matching

The direction weights are applied to each pixel, which are common to both the template and the search windows. The weights are summed to form the correlation value. The template image 'slides' around the search window and a correlation value is computed for each location of the template inside the search window. A 3x3 convolution filter is applied to the correlation surface image. This smoothes the correlation surface prior to maximum point extraction. If the maximum correlation point is less than a predetermined threshold then the track coasts for that frame, otherwise the location of the maximum correlation is marked as the target location.

4.5 Template Updating Based on Image Growth

The update rate of the template image is based on the size growth of features as the closing sequence progress. This size expansion is a function of the range change and the absolute range. Shown in Figure 8 is the formula for calculating the template number. The start range is r0 and k is the percentage growth that is acceptable before a template update is required.

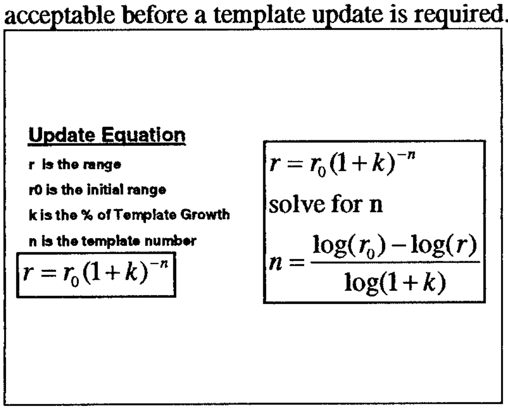

Figure 8: Growth Rate for a Closing Sequence

4.6 Example Output From the Weighted Edge Tracker

Figure 9 shows a composite image. The large image in the upper left is the image being tracked by WET. The small window on the left edge shows the stabilized portion of the image. The colored line image in the upper right is the template image and the colored line image in the lower left is the search window. The image next to the search window is the correlation accumulation image and it shows a definite peak. This peak means that there is a good correlation between the template and search windows. The coloring coding signifies unique directions.

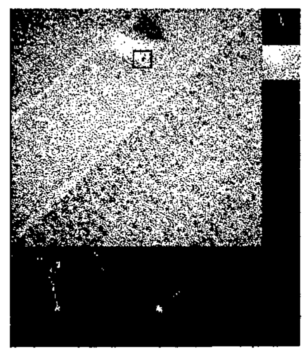

Figure 9: Composite Image

5.0 Summary

This paper has outlined a methodology for automatically detecting, classifying, and tracking an armored target in a closing sequence. While the automatic acquisition technique is based on a classical approach, it is a robust real-time algorithm. An additional strength of this approach is that the ATA has been appropriately integrated with a novel terminal-tracking algorithm, WET.

6.0 Acknowledgements

The authors would like to thank Dat Nguyen for his dedicated work in implementing the algorithms in Matlab and creating the output data runs.

They would like to thank Dr. Richard Sims from US Army Aviation and Missile Command (AMCOM), Redstone Arsenal, Alabama for providing the imagery being used in this algorithm development.