Integrating correlation filtering and tracking for better target detection

Ryan Kerekes and B. V. K. Vijaya Kumar

A framework that combines correlation arrays over time using probabilistic models can improve automatic target detection and tracking.

Automatic target recognition (ATR) refers to the task of unaided location and classification of targets in a scene (see Figure 1). It is an important component of much defense technology, including reconnaissance systems, avionics, smart weapons, and electronic warfare. For a soldier on a reconnaissance mission, visually scanning and tracking possible adversaries may be overwhelming. However, a computer equipped with ATR can identify and track all targets that enter the vicinity, allowing the soldier to focus on other tasks.

ATR typically includes detection (locating all targets of interest in a scene), recognition (classifying each one), and tracking (connecting a target’s locations over time to determine its path). Integrated methods have increasingly blurred the lines distinguishing these three tasks. The classical approach, to divide them into stages and solve them separately, inevitably results in information loss. For example, a target missed at the detector stage may as a consequence elude the tracking stage, with adverse results. In addition, false detection can complicate the so-called “data association” problem in which decisions must be made that link an existing track (if any) with each data point. Recent work has shown that better results may be obtained by approaching these issues in a more integrated fashion.

In many ATR applications, the scene contains an unknown number of targets of variable aspect and appearance at unknown locations, all of which must be considered when designing a robust algorithm. Also, for real-time applications such as weapons guidance systems, in which computational resources are limited, efficiency of the detection algorithm may be of paramount importance. Correlation filters (CFs) are well-suited to such applications, due to attractive properties that include shift-invariance (which means that targets need not be centered in the image), distortion tolerance, and closed-form solutions. The output of a CF is a 2-D array of correlation values (one per image pixel) that can be compared to a pre-existing threshold for detection. CFs have been used successfully for detection and recognition in scenes with unknown numbers of targets and heavy clutter. They can be designed for specific targets and so are able to solve the detection and recognition problems simultaneously.

We have developed a new framework to improve the link between detection and tracking in the context of correlation filtering. We refer to it as multi-frame correlation filtering (MFCF). The principal difference between this and other approaches is that the goal is not to compute individual target tracks but rather to produce better output arrays by combining

Figure 1. A sample frame from a noisy forward-looking infrared video sequence. The scene contains two targets that must be identified and tracked.
Multi-frame correlation filtering (MFCF) takes a noisy correlation array (top) and combines information from previous frames to produce an enhanced probability array (bottom). PSR: peak to sidelobe ratio.

To calculate enhanced target probabilities for the next frame, convolve enhanced target probabilities for the previous frame (left) with a motion model function (right), such as a 2D Gaussian.

Combined with a correlation filter, MFCF can significantly reduce false alarms generated at each frame (left) as compared to the same filter without MFCF (right). False alarms are represented with the ‘x’ symbol, and ellipses denote the target regions.

information from several correlation outputs over time before thresholding (see Figure 2). This is accomplished using a simple motion model. These “enhanced outputs,” used in the same way as the original outputs, can be thresholded for detections and, if desired, sent to an external tracker that might operate with a more complex motion model. Because the framework uses a target motion model, tracking is in some measure integrated.

Key to MFCF is mapping from correlation output values to probability values. Specifically, we convert each correlation value to the probability that a target is centered at the corresponding pixel location, based on learned models. We then employ a target motion model, represented by a 2D probability mass function (e.g., Gaussian), that gives the probability that a target will jump from its current location to any other nearby location in the next frame. We have shown that, by convolving the converted correlation array with this motion function, it is possible to efficiently compute an array of prior probabilities for the next frame (see Figure 3). These are then combined with the correlation output values from the succeeding frame to yield an enhanced probability array that we can threshold for more robust target detection. This process can be iterated forward indefinitely to enhance all correlation outputs from future frames.

We have tested MFCF on several video sequences similar to the one shown in Figure 1. We observed in some sequences a 95% reduction in false alarms with a 90% detection rate. As an example, we show in Figure 4 the false alarms generated by a correlation filter at 90% detection, both with and without MFCF on a single frame of a video sequence. Such a drastic reduction could make a huge difference to a subsequent tracking algorithm, which must determine whether each new detection corresponds to some tracked target.

Based on preliminary simulations, MFCF appears to be a valuable plug-in to existing correlation-based detectors, especially when noise is prevalent in the video. The framework could also be used with other non-correlation–based detectors, such as support vector machines, provided that the detections are arranged on a grid. Future work will focus on better ways to determine appropriate probability models in the framework, since such models are critical to performance. We also envision adapting our framework to combine information from multiple views of a scene.

Continued on next page
The authors would like to thank Dr. Richard Sims for supplying the forward-looking infrared imagery used in this article.

Author Information

Ryan Kerekes
Computational Sciences and Engineering Division
Oak Ridge National Laboratory
Oak Ridge, TN, 37831
http://www.ornl.gov/~2v7

B. V. K. Vijaya Kumar
Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA, 15213

References