SAND2016-11386C

Tracking of Streaking Targets in Video Frames

Andrew Finelli, Peter Willett, Yaakov Bar-Shalom
Department of ECE
University of Connecticut
Storrs, Connecticut 06269
718-608-4408
andrew.finelli@uconn.edu, peter.willett@uconn.edu, yaakov.bar-shalom@uconn.edu

David Melgaard, Raymond Byrne
Sandia National Laboratories
1515 Eubank SE
Albuquerque, NM 87123
dmelga@sandia.gov, rhbyrne@sandia.gov

Abstract— **A method for tracking streaking targets (targets whose signatures are spread across multiple pixels in a focal plane array) is developed. The outputs of a bank of matched filters are thresholded and used for measurement extraction. The use of the Deep Target Extractor (DTE, previously called the MLPMHT) allows for tracking in the very low observable (VLO) environment common when a streaking target is present. A definition of moving target signal to noise ratio (MT-SNR) is also presented as a metric for trackability. The extraction algorithm and the DTE are then tested across several variables, including trajectory, MT-SNR, angle and dropped measurements (dropped frames). The DTE and measurement extraction process performs remarkably well in this difficult tracking environment on these data features.**

TABLE OF CONTENTS

1. INTRODUCTION

The problem we are addressing in this paper is the tracking of targets in an image set where they are streaking in each image. By this, we mean that the target's signature has an elongated or extended spread across several pixels in a frame. This type of imagery arises in situations including those for fast moving targets and those for targets with very low SNR. In the former situation, a streaking target can be created by a target that moves fast relative to the sampling rate of the sensor. This would cause a target to move across several focal plane array (FPA) pixels during a single image (frame) integration period. In the low SNR situation, the integration time may need to be extended in order to observe the signal at all, in which time the target may move across several pixels by the time the integration is completed. In either of these situations (or a combination of the two) a streak will appear in the imagery, making the target difficult to track using standard tracking approaches.

In Section II, we review previous works that compare several tracking and data association algorithms for effectiveness. Next, we present relevant background research on streaking targets in images. Much of the literature presents techniques that require a priori knowledge of the target, which we will not assume is available when designing our algorithm. In Section III, we describe a method to generate streaking target data. We assume a Gaussian point spread function due to the sensor optics and additive Gaussian noise in the FPA. We also define two target SNRs appropriate for streaking targets and show some examples of this generation technique, so that the streak can be easily seen. The measurement extraction is discussed in Section IV. Finally, in Section V, we describe our technical approach: the DTE (deep target extractor, also known as the MLPMHT) which, in addition to its excellent estimation performance seems especially well-suited to a streaking target.

2. BACKGROUND

While the characteristic of the data we are focusing on is the undersampled nature of the image frames, our tracking and data association algorithm also takes into account the characteristics of very low observable (VLO) data. These characteristics include high clutter, sporadic periods of low signal to noise ratio (SNR), and an indeterminable number of targets in each image frame. In order to select a tracker that will best track targets subject to this type of environment, we reference the work presented in [2], [9], [10], [11], [12], [13].

The paper [13] discussed the application of three different algorithms for tracking VLO data. These algorithms were the Maximum Likelihood Probabilistic Data Association tracker (MLPDA), the Maximum Likelihood Probabilistic Multi-Hypothesis Tracker (MLPMHT), and the Interacting Multiple Model Probabilistic Data Association using Amplitude Information (IMMPDAFAI). The last of these was possibly the best at data association of a single target in a single frame, but fell short when it came to tracking, with only the highest amplitudes being accepted for tracking.

The other two data association and tracking algorithms performed significantly better at tracking the VLO test data in [13]. These two trackers operate on the premise that a target's trajectory is deterministic given a kinematic model and certain parameters (usually initial position and velocity of the target). These trackers also have a lower "bandwidth"

978-1-5090-1613-6/17/31.00

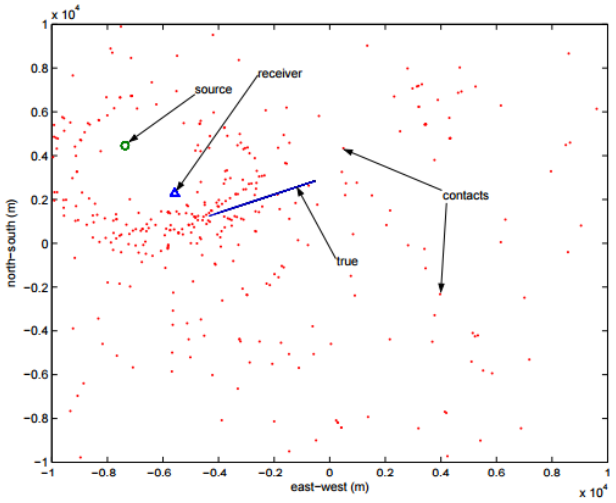


Figure 1: Example DEMUS data from January 2004 sea trial. For purposes of plotting only the top 1% (in amplitude) of the contacts are shown.

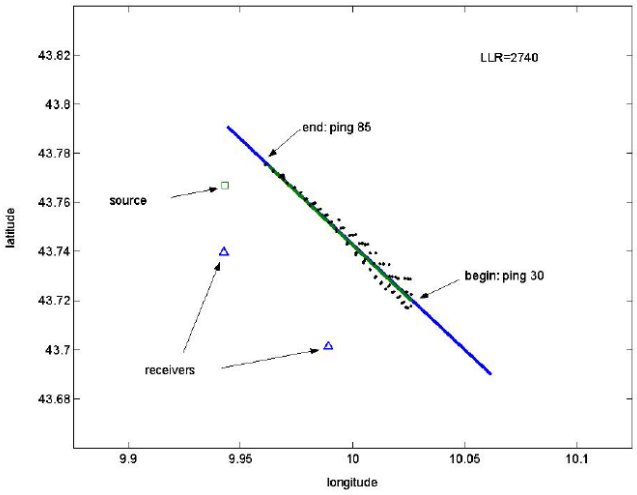


Figure 2: The track detected by MLPDA that corresponds to the true trajectory. The dots indicate measurements associated to this track. In this case the peak-amplitude contacts were used.

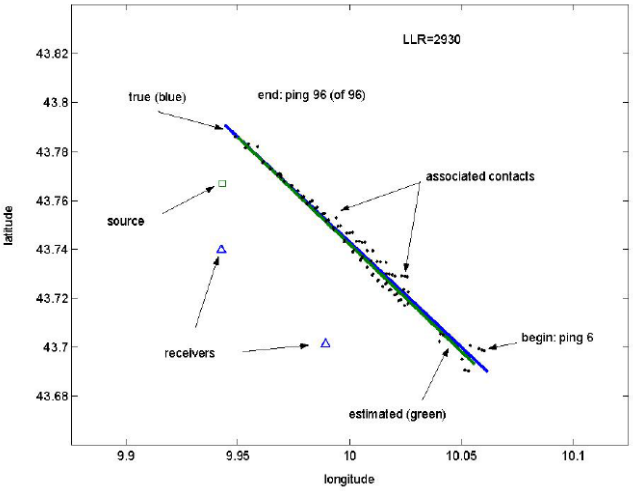


Figure 3: The track detected by MLPMHT that corresponds to the true trajectory. The dots indicate measurements associated to this track. In this case the peak-amplitude contacts were used.

(in the sense of greater structure in the targets sought) and thus are less affected by high amounts of clutter. or false alarms due to sensor noise.

The differences between the two MLP algorithms (MLPDA and MLPMHT) are described in detail in [9], [13] but the main difference is in how each models the measurement generation from each target. The MLPDA makes the assumption that each target generates a single measurement (contact) every scan. On the other hand, the MLPMHT assumes each contact in the frame is independent and, a priori, equally likely to have been generated by a target. The MLPMHT is therefore more advantageous, as the concept of a frame is hard to maintain with long exposure times, something we anticipate to be true with the problem of streaking targets.

Overall, the MLPMHT was chosen to be the best tracker in [13] when it came to several different criteria. When exposure time is long, the tracker's assumptions on data do not become more questionable; and indeed we propose that it is even more desirable. With VLO data, during periods of low SNR, the MLPMHT had more detections than the MLPDA and held tracks on as true tracks for a longer period of time making the tracking performance better. Finally, the MLPMHT tracks were generally the cleanest and the most accurate compared to the true trajectory out of the algorithms tested for VLO data. The imagery in Figures 2 and 3 show the application of these trackers in situations tested in [1], specifically the test on NURC's DEMUS (deployable multistatic sonar) data from its March 2003 Mediterranean Sea trial. The images show thresholded contacts, taken from an active sonar sweep, over several pings from the sonar as black dots. The blue lines show the true trajectory of the target and the green lines show the estimated track based on the associated contacts. The beginning and ending pings are marked as the target moves from the bottom right of the frame to the top left of the frame in the presented orientation. We will use the name of this tracking algorithm as "Deep Target Extractor", or DTE for conciseness. This is the algorithm which we shall use since it is the most effective at data association, target detection, and tracking with appropriate extensions to handle streaking targets.

3. STREAKING TARGETS

There has been very little research done on the topic of streaking target tracking. The research that was done seems to focus more on being able to discern the difference between the features in an image, and less on how to effectively estimate the motion that caused these features. The most recent research directly related to the problem we are addressing was done in 1998 [8]. This paper developed a processing technique which allows for detections based on Poisson statistics and a known starting position and velocity. When the initial position and velocity were unknown, the processing technique required a "bank" of velocity-matched filters to create a two-dimensional likelihood mapping for the target position and velocity. These likelihood maps would then need to be resolved to find the best match for target position and velocity. This process predates the MLPMHT, which would avoid the need for these numerous matched filters using multi-frame

log likelihood ratios (LLRs) in the algorithm. Also, the unresolved multi-target problem is similarly resolved by the MLPMHT LLRs. According to [8] the Poisson statistics accurately reflect the underlying physical processes that cause the streaking target phenomenon, but the Gaussian statistics (which we will be using) better match the field data. For this, he cites the reasoning that the Gaussian modeling lends itself better to data containing arbitrary offsets and scaling, something often found when dealing with real sensor data.

More recently, there was a study done in [7] on how to extract a satellite streak from an image with common signal degradation and sensor artifacts (bloom, dropped pixels, etc.) as well as high intensity clutter, such as a star. This work presented a technique using matched filters to automatically remove the sensor artifacts, noise, and high intensity clutter. The paper then goes on to explain how to design a matched filter to extract a streak from an image by multiple iterations of convolving a generated streak with the processed data and thresholding. The process was even shown to successfully extract a streak when a star (high intensity clutter) was superimposed behind the streaking target. This work showed that the streaking target can be extracted from each frame in most situations in a time efficient manner. With appropriate data association, the streaking target should be able to be tracked. One limiting factor of this algorithm, however, is that one must have a priori information about the movement of the target (i.e. the direction of the streak) in order to design the matched filter. This information is often not available when it comes to real time target tracking and the present work does not assume it.

Ground Truth Data Generation Model

In order to better understand the nature of the streaking phenomenon, and to have a wider range of data on which to test our algorithms, we created a procedure to generate data of streaking targets. In our data model, the streak is generated by a target which moves across several pixels during the frame integration time, T. Some assumptions made to generate this data are first that the position of each target in the focal plane is known (this is the ground truth, which is not known to the tracker). Second, we assume that the amplitude of the target's signature in the focal plane gets spread by the sensor's optics' point spread function (PSF) is known, and that the PSF is a two dimensional circular Gaussian in the focal plane. Thirdly, we assume that the data generation interval (Δ) for the position of the target in the focal plane ($r(t)$) is small enough such that the data generation can be done with r(t) = r(nΔ) with a small enough error that is negligible. Finally, we assume that the target's velocity is, on average, such that more than a single pixel is traversed during the integration period, T, ensuring a streak.

To develop this model, we denote the target's position in the focal plane as $x(t)$ and $y(t)$, with $t = 0$ being the time that taking measurements began. The focal plane is of length w_x x w_y. The integration time for each sensor is T. In order to allow for the generation of these images, we sample the position functions by sampling every Δ seconds, with $\Delta \ll T$. Overall, the total time of observation, Ψ, consists of M frames with integration time T, during which the target moves N times at increments of time Δ, thus $\Psi = MT = MN\Delta$. During the (small) intervals Δ, the target is assumed to be at a fixed location, which allows a quick (but still accurate) calculation of the energy deposited in each pixel via the PSF.

Denote the amplitude of the target's Gaussian PSF at each sampling increment as A. The target's PSF also has a "covariance" (spread) matrix that is diagonal and identical in x and y to create a circular Gaussian. So the time varying (moving) position of the PSF center of the target is:

$$r(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} \approx r(n\Delta) = \begin{bmatrix} x(n\Delta) \\ y(n\Delta) \end{bmatrix} \quad (1)$$

The PSF value at location ρ in the focal plane when the target is located at $r(n)$ (short for $r(n\Delta)$ in (1) is:

$$PSF(\rho, r(n)) = A\, N\left[\rho; r(n), \sigma^2_{\text{psf}} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right] \quad (2)$$

Here, $N(\rho; \mu, \Sigma)$ is a two dimensional Gaussian with mean μ and covariance Σ [1] and $\sigma_{\text{psf}} = 0.5\delta$, where each pixel has size $\delta \times \delta$. Next we use this PSF to populate the pixels of our data frames. We do this by integrating the amount of the spread in each pixel for each time step during an integration period and adding them together. This is done for each pixel and the value obtained is the pixel's value, which we will call $\mu(i, j)$. This is implemented in the following way:

```
for n = 1:T/Delta
   for j = 0:w_y
      for i = 0:w_x
```

$$\mu(i+1, j+1) = \mu(i+1, j+1) + \int_{i\delta}^{(i+1)\delta} \int_{j\delta}^{(j+1)\delta} A\, N\left[\rho; r(n), \sigma^2_{\text{psf}} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right] d\rho \quad (3)$$

With the assumption that the PSF is a circular Gaussian, this computationally intensive integral operation in (3) can be simplified via a product of Gaussian Q-functions, in the following way:

```
for n = 1:T/Delta
   for j = 0:w_y
      for i = 0:w_x
         if(pixel i,j shares a row or column with target's
            position at time k)
               Remove absolute value from inside Q-Function of the
               corresponding row and/or column terms in the following
         else
```

$$\mu(i+1, j+1) = \mu(i+1, j+1) + 2\pi A \cdot \left| Q\left(\frac{|i - x(n)|\delta}{\sigma_{\text{psf}}}\right) - Q\left(\frac{|i+1 - x(n)|\delta}{\sigma_{\text{psf}}}\right)\right| \cdot \left| Q\left(\frac{|j - y(n)|\delta}{\sigma_{\text{psf}}}\right) - Q\left(\frac{|j+1 - y(n)|\delta}{\sigma_{\text{psf}}}\right)\right| \quad (4)$$

The "if..." statement prior to (4) prevents an error in calculating the amplitude when the Q-function is used.

Furthermore, the process in (4) is applied to only those pixels within a relevant neighborhood of the target, for instance

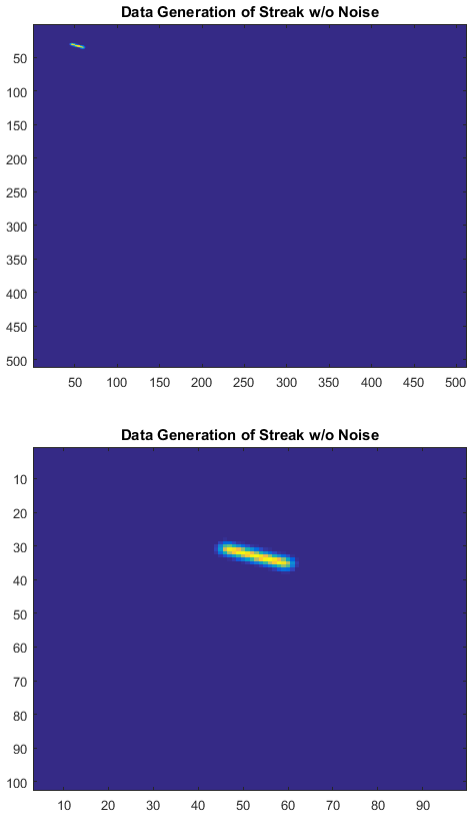


Figure 4: A point target with a large $\sigma_{\text{psf}} = 1\delta$ generated by the method used above. There is no noise added to this 512 x 512 image. The bottom subfigure is a zoomed image of the left.

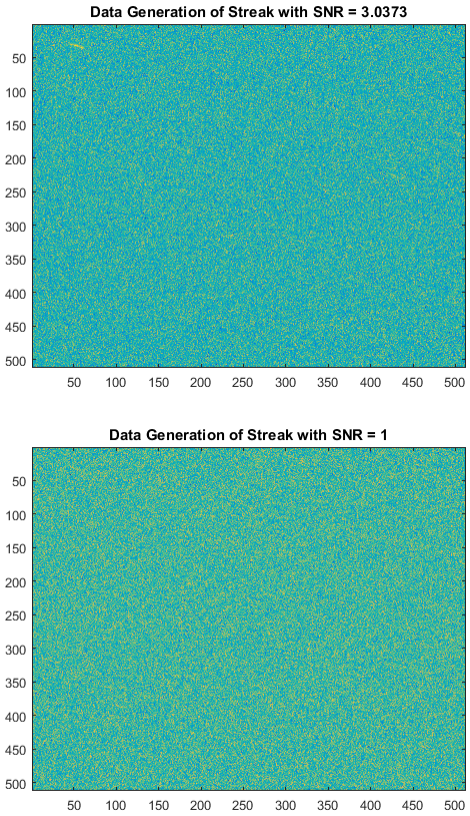


Figure 5: The same streak generated above, however these images show the effects of different additive Gaussian noise levels on the streak. The image is normalized to the noise so the target shown is weaker in the lower SNR case.

within $3\sigma_{\text{psf}}$, in order to lessen the number of computations needed. This process is then repeated for each integration period for which we are observing the target trajectory, with each integration period yielding a single frame.

Finally, add zero-mean Gaussian random noise, $\mathcal{N}\left(\prime, \sigma^{\in}_{\text{noise}}\right)$, to every pixel of each frame independent of every other frame (i.e. no temporal noise correlation) and pixel. Figures 4 and 5 illustrate the results of this data generation, with an extremely bright point target moving a large distance during frame integration time. The target intensity and σ_{psf} is constant and the trajectory of the target is a straight line. The stated SNR is the ratio of the target's peak amplitude to the r.m.s. noise intensity σ_{noise}.

If one wishes to test algorithms even more robustly, one can add background clutter to the images fairly easily as well, but the process to do so will not be discussed here because it is too scenario specific. We believe the data generation process described above will give valuable insight as to the behavior of these types of targets when it comes to estimation and tracking of streaking targets.

SNR Definitions

Defining the SNR in the case of this type of generation for streaking targets proves difficult. We can define a stationary target SNR (ST-SNR) as what the target's SNR would be if the target was not moving during an integration time (T). In this stationary SNR case, the calculation can be done by multiplying the number of time steps (of length Δ) within an integration time ($\frac{T}{\Delta}$) by the target intensity (assuming a constant target intensity during each Δ) and dividing by the noise intensity (r.m.s. value, σ_{noise}). This number would be larger than the SNR of a moving target because the peak of the PSF would be moving. The SNR would be a function of the overlap of the PSFs during an integration time. The overlap depends on the target velocity and path through the focal plane. In order to approximate the moving target SNR (MT-SNR), we divide the stationary target SNR (assumed to have a fixed intensity A_{Δ}, the deposited energy during Δ in the FPA) by the length of the streak in a single frame in pixel units, L. These two definitions are summarized below.

$$\text{Stationary Target SNR} = \frac{\frac{\text{T}}{\Delta}\text{A}}{\sigma_{\text{noise}}} \triangleq \text{ST} - \text{SNR} \quad (5)$$

$$\text{Moving Target SNR} = \frac{\frac{\text{T}}{\Delta}\text{A}}{\sigma_{\text{noise}}\text{L}} \triangleq \text{MT} - \text{SNR} \quad (6)$$

In the algorithm evaluations, we will use the MT-SNR (6)

4. TARGET MEASUREMENT EXTRACTION

We have discussed how we generate the data to model streaking targets, and now we will discuss how data will be

extracted from image frames. We begin by recognizing that the streaks will be elongated along the direction of motion. We would like to extract these shapes and minimize the artifacts from noise and clutter. We recognized that there are an infinite number of directions that a straight line streak can travel when projected on a two dimensional image plane. However, because of how the image is represented as pixels, we decided that attempting to identify streaks in one of 16 directions and with 8 different lengths would be sufficient to extract the majority of streaks that can exist. The directions of streaks that we chose to focus on extracting were those that started at a point and extended outwards at an angle $\theta = \frac{n\pi}{16}, 0 < n < 15$. The 8 lengths of these streaks that we considered were those that were between a length of 2 pixels and 16 pixels, at an interval of 2 pixels. In order to perform the extraction, the shapes of the matched filters were generated (with no noise) by the same process as described in Section III.1 and then normalized so that the sum squared of the terms in all $N_{MF} = 128$ filters is the same. The σ_{psf} of these shapes was the same as in (2) . Once these shapes were generated, they were used to extract measurements from the noisy image frames. The generated shapes are used as a bank of matched filters to convolve with the noisy image frames. The outputs of these matched filters, denoted $Y_s(i,j)$ for matched filter s and pixel address (i,j), will produce a relative large output at points where there is a streak and a relative small output where one does not exist.

The technique used on the matched filter outputs we call the "top output" technique. There is no way to tell which streak is exhibited in the image before these filters are applied, the output of each is weighed evenly and we threshold the filter outputs of each by taking the mean and standard deviation (σ) of the frame's output from every filter. We then eliminate all measurements from these frames that are less than a certain number κ of standard deviations above the mean. The number of σ is a design parameter of the extractor and the best number to use will change depending on the data and also will depend on the SNR. The selection of this number is based on the maximum number of false measurements per frame that the DTE can handle. For each pixel, if the matched filter output is above the threshold in at least one matched filter, then that pixel is considered a detection. The set of binary pixel detections for each frame is given to the DTE for use in extracting a track. We call the set of binary detection measurements passed to the DTE, $Z(m)$, where m is the frame's index in the batch. The measurements contained in $Z(m)$ are the detections of the center point of the streak. The "top output" technique is defined mathematically next.

$$\text{Define} \quad \text{h}_\text{s} = \text{s}-\text{th Match Filter Impulse Response},$$

$$s = 1, 2, ..., 128 \tag{7}$$

$$x(i,j) = \text{observed intesnsity of pixel(i, j)};$$

$$i = 1, \ldots, W_x, \; j = 1, \ldots, W_y \tag{8}$$

and the indicator function

$$I(\text{cond}) = \begin{cases} 1 & \text{cond} = \text{True} \\ 0 & \text{cond} = \text{False} \end{cases}$$

The output of the s-th matched filter when centered at pixel (i,j) is

$$Y_{ijs} = h_s * x(i,j) \tag{9}$$

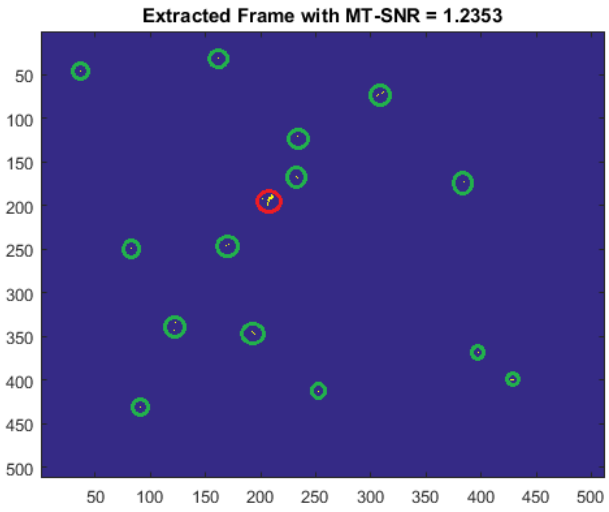


Figure 6: ***Extracted streak using the second technique from data with MT-SNR = 1.24,*** $\sigma_{\text{psf}} = .5\delta$***, and streak length*** 1.94δ***. The target is circled in red. The green circles show false alarms.***

A detection is declared in pixel (i,j) if there is at least one output with intensity at least τ_1

$$\eta_{ij} = I(\sum_s I(Y_{ijs} \geq \tau_1) \geq 1) \tag{10}$$

$$\text{If } \eta_{\text{ij}} = 1, \text{then } \eta_{\text{ij}} \; \epsilon \; \text{Z(m, n)} \tag{11}$$

The mean intensity of all matched filter outputs is

$$\mu = \frac{\sum_{i,j,s} Y_{ijs}}{N_{\text{MF}} W_x W_y} \tag{12}$$

and their variance is

$$\sigma^2 = \frac{\sum_{i,j,m} Y_{ijs}^2}{N_{\text{MF}} W_x W_y} - \mu^2 \tag{13}$$

These are then used to define the threshold τ_1 in (10)

$$\tau_1 = \mu + \kappa\sigma \tag{14}$$

We set the maximum number of measurements per frame to pass to the DTE as 80. The data seen in Figure 6 has a majority of the extracted measurements as a component of the streak (the red cluster), with the rest resulting from high intensity noise or random "streak like" collections of noise. The latter yielded the green clusters (false measurements) in Figure 6.

5. Deep Target Extractor Tracking (MLPMHT)

As discussed before, the measurements that were extracted from the data will be given to the DTE in order to track the target. For this formulation we assume the target is moving in a constant velocity trajectory. If this is not the case, the trajectory can be taken in segments small enough

that a straight line approximation is close enough to track accurately and apply the DTE on batches. We also have assumed from the target extraction phase that the streaks can be considered to be in a constant velocity motion.

Overall, the DTE algorithm creates a track likelihood function (LF) based on extracted measurements, which is the pdf of the measurements conditioned on the parameter vector related to the target position and motion [1]. We chose to use the starting and ending pixel positions as the parameters to maximize over, namely,

$$\hat{\mathbf{x}} = \begin{pmatrix} \hat{x}_{\text{start}} \\ \hat{y}_{\text{start}} \\ \hat{x}_{\text{end}} \\ \hat{y}_{\text{end}} \end{pmatrix} \tag{15}$$

We will then use the conditional pdf of each of the measurements in the set of measurements for the batch. The batch comprises M frames and each frame in the batch there are N_m measurements. This density function will be conditioned on the parameter vector $\vec{x}$ defined in (10). We will then multiply these likelihood functions together and maximize the total likelihood function of the parameter vector, yielding the estimate of (15) as

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} \left(\prod_{\mathbf{m=1}}^{\mathbf{M}} \prod_{\mathbf{n=1}}^{\mathbf{N_m}} \mathbf{P}(\mathbf{z}(\mathbf{m},\mathbf{n})|\mathbf{x}) \right) \tag{16}$$

In order to allow for simplicity and better computation speed, we can similarly maximize the log-likelihood function (LLF). This reduces (16) to the following:

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} \left(\sum_{\mathbf{m=1}}^{\mathbf{M}} \sum_{\mathbf{n=1}}^{\mathbf{N_m}} \ln(\mathbf{P}(\mathbf{z}(\mathbf{m},\mathbf{n})|\mathbf{x})) \right) \tag{17}$$

Next define the pdf of each measurement conditioned on the parameter vector. We do this by summing the multiplication of the a priori and conditional probabilities that each measurement comes from clutter (or noise) and that it comes from the target. This is defined, according to the MLPMHT [9], as

$$\begin{aligned} P(z(m,n)|\mathbf{x}) &= \Pi_0 P(z(m,n)|\text{clutter}) \\ &+ \Pi_1 P(z(m,n)|\text{target}) \end{aligned} \tag{18}$$

where Π_0 and Π_1 are the prior probabilities that the measurement comes from clutter or the target, respectively. We further define the pdf of the clutter locations in (18) to be uniform over the area of the FPA and for the target to be Gaussian with a covariance matrix R. We can then rewrite (18) as

$$P(z(m,n)|\mathbf{x}) = \frac{\Pi_0}{W_x W_y} + \Pi_1 N(z(m,n); x_m, R) \tag{19}$$

where $z(m,n)$ = (i,j) is the pixel corresponding to measurement n in frame m, $\mathbf{x_m}$ is the putative location in the FPA of the target in frame m (in the middle of the integration interval, T), and R is the covariance matrix of the measurement error between the pixel with the detection and the actual target location.

This finally allows us to rewrite (16) as:

$$\tilde{\mathbf{x}} = \arg\max_{\mathbf{x}} \sum_{m=1}^{M} \sum_{m=1}^{N_m} \ln\left(\frac{\Pi_0}{W_x W_y} + \Pi_1 N(z(m,n); x_m, R) \right) \tag{20}$$

The covariance matrix, R, is determined as follows. The location of a target originated measurement is

$$z(m,n) = x_m + w_{mn} + a_{mn} \cdot l \tag{21}$$

where $\mathbf{x_m}$ is the putative location of the target at the middle of the current frame in the batch, $\mathbf{w_{mn}}$ is a zero-mean, white Gaussian noise (2-D) with identity covariance matrix, a_{mn} is a uniform random variable on -.5 to .5 (1-D), and l is the putative streak length. More specifically, $l = |\vec{v}|$, where $\vec{v}$ is the putative velocity in pixels/frame and M is the number of frames in the batch.

We can then define the covariance matrix R as

$$\begin{aligned} R = \mathrm{E}(\mathrm{z_{mn}} \mathrm{z_{mn}^T}) &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ &+ \frac{1}{12} \begin{bmatrix} \frac{x_{\text{end}} - x_{\text{start}}}{M} \\ \frac{y_{\text{end}} - y_{\text{start}}}{M} \end{bmatrix} \begin{bmatrix} \frac{x_{\text{end}} - x_{\text{start}}}{M} & \frac{y_{\text{end}} - y_{\text{start}}}{M} \end{bmatrix} \delta^2 \end{aligned} \tag{22}$$

In order to perform the maximization of (20), we encounter a non-convex, constrained, global optimization problem. Furthermore, the optimization process must be carried out with respect to the vector (15), i.e., in four dimensions. This type of optimization is computationally costly, so an efficient algorithm must be used to obtain results in a reasonable amount of time. We use a fast global search to obtain an initializing point for a local maximization algorithm to work on.

The global optimization method that we used was a Dividing Rectangles method devised in [6] and included in the NLOPT optimization package in the C programming language written by [5]. We use the output of this optimizer to start a local search for the maximum closest to this initialization. The local search that we use is the Matlab command "fminunc", implemented in [3] and [4]. In order to use this function, we chose to minimize the negative of the log-likelihood function in our optimization. We then take the output of this optimization to be the estimated starting and ending potions of the target.

6. RESULTS AND DISCUSSION

This section shows the results of the DTE track extraction, using the "top output only" measurement extraction technique. We will present how the tracker works under various SNR, streak length, and trajectory (velocity). All tests were constant velocity trajectories for the entire duration of the batch. This is a good enough approximation for an appropriately short part of any trajectory.

First, we will look at the effect of varying the SNR, namely the MT-SNR. The DTE performs remarkably well at low SNR. Even when the target is indistinguishable from noise by a human eye, the MLPMHT is able to extract a streaking target track with low RMSE. The following test situation was generated with a 60 frames batch, $\frac{T}{\Delta} = 20$, $\sigma_{\text{psf}} = 0.5\delta$, $x_{\text{start}} = 100$, $y_{\text{start}} = 500$, $x_{\text{end}} = 490$, $y_{\text{end}} = 227$, and $\sigma_{\text{noise}} = 100$. We also assume the inter-frame-integration time to be zero, for simplicity. To change the SNR, we have varied the signal amplitude as A = 50, 30, 20, and 10. The thresholding on target extraction was taken such that the DTE receives approximately 80 measurements per frame. Results are shown in Figure 7.The streak length is given in units of δ, the RMSEV is given in $\frac{\delta}{\text{Frame}}$, and the RMSEP is given in δ from the streaks' center.

The next test looks at the effect of varying the trajectory direction. While we vary the initial positions and velocities of the generated targets, the SNR and streak lengths will remain constant. The target is tracked when moving at any constant velocity trajectory. The situations tested here are horizontal and vertical trajectories, as well as trajectories at angles of $\theta = \frac{\pi}{4}, -\frac{\pi}{5}$, in the FPA. The following test situations were generated with a 60 frames batch, $\frac{T}{\Delta} = 20$, $\sigma_{\text{psf}} = 0.5\delta$, and $\sigma_{\text{noise}} = 100$. The Results are seen in figure 8.

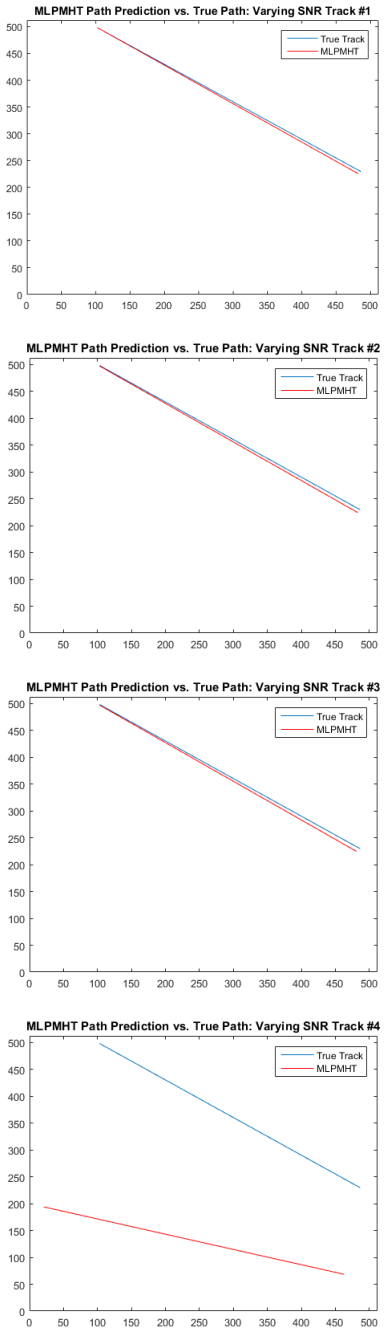
The final test examines the effect of varying the streak length. We will keep the polar angle of velocity and target MT-SNR constant while varying the magnitude of the velocity. This will change the amount of motion during integration time and, thus, the length of the streak. This will also change the full trajectory's length as well. The target is able to be tracked for streaks of most lengths. The following test situations were generated with a 60 frames batch, $\frac{T}{\Delta} = 20$, $\sigma_{\text{psf}} = 0.5\delta$, $x_{\text{start}} = 100$, $y_{\text{start}} = 500$, $x_{\text{end}} = 197.5; 295; 392.5; 490$, $y_{\text{end}} = 431.75; 363.5; 295.25; 227$, and $\sigma_{\text{noise}} = 100$. The Results are seen in figure 9.

7. Conclusion

In this paper, we developed a method for dealing with VLO streaking targets in image frames. We have examined multiple tracking techniques and have decided that the DTE is the best way to track these targets. We presented an algorithm for extracting streaking targets traveling in different directions and at different speeds using a bank of matched filters. Next, we showed the process by which the Deep Target Extractor (DTE, previously called the MLPMHT) is used specifically to track the streaking targets after the measurements have been extracted. Finally, we showed that streaking targets of different trajectories, speeds, and SNR can be tracked accurately by the DTE algorithm developed here. We find that an MT-SNR ≥ 0.4 is needed for the DTE to successfully track over 60 frames

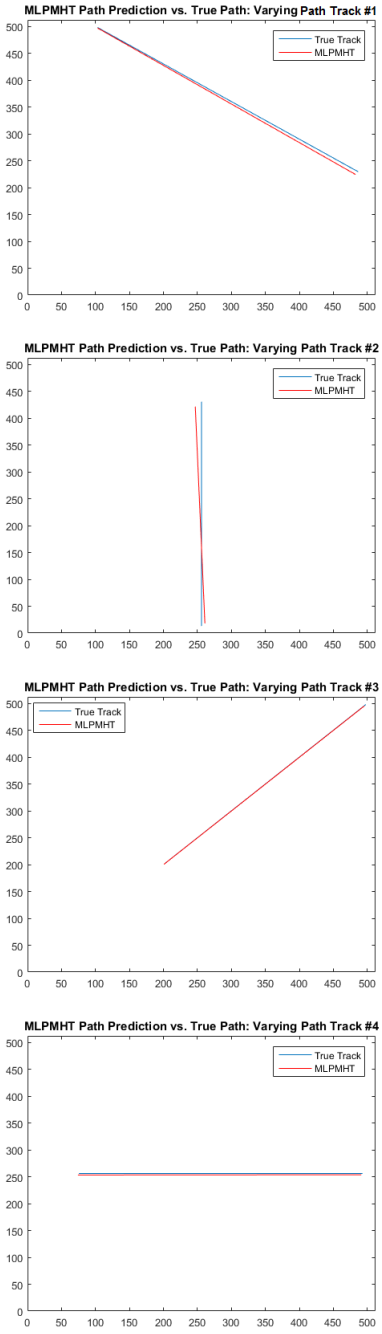
Acknowledgments

This research was funded by the Laboratory Directed Research and Development (LDRD) program at Sandia National Laboratories. Sandia National Laboratories is a multiprogram laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.



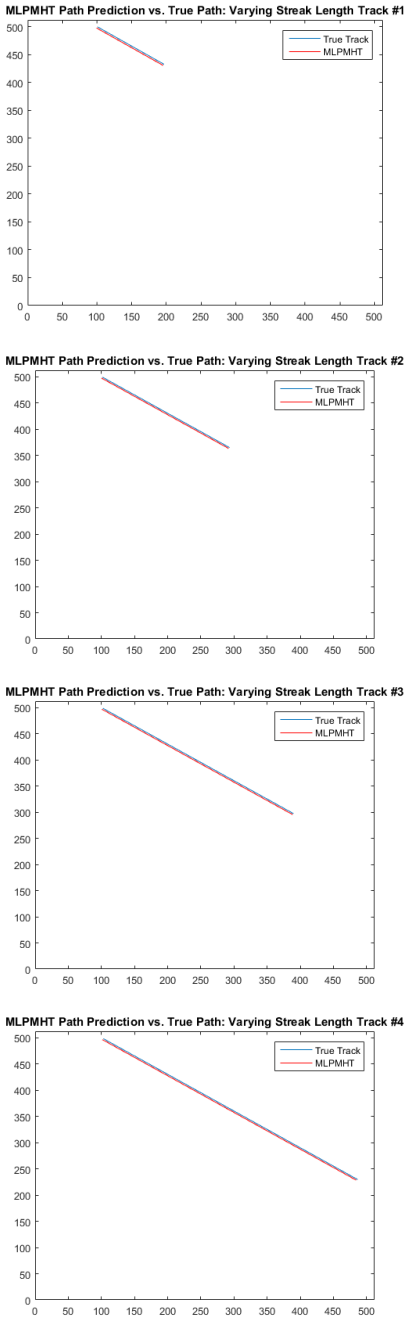
$M = 60$	Track 1	Track 2	Track 3	Track 4
ST-SNR	10	6	4	2
MT-SNR	1.1193	0.6716	0.4477	0.2239
MT-SNR(dB)	.9789	-3.4578	-6.9803	-12.9989
Streak Length	8.9343	8.9343	8.9343	8.9343
RMSEV	0.7047	0.6806	0.8076	1.2862
RMSEP	2.7543	3.6117	4.0235	239.7935
κ	4.5	4.5	4.5	4

Figure 7: ***A set of four tracks, with different SNRs, compared to their ground truth trajectories. The track information is within the table above. Track 4 shows an inability to track due to extremely low SNR. The angle of the trajectory is*** $\theta = -\frac{\pi}{5}$.



$M = 60$	Track 1	Track 2	Track 3	Track 4
ST-SNR	6	6	6	6
MT-SNR	0.7434	0.7434	0.7434	0.7434
MT-SNR(dB)	-1.2878	-1.2878	-1.2878	-1.2878
Streak Length	8.0711	8.0711	8.0711	8.0711
RMSEV	0.7047	1.0399	0.3622	0.3668
RMSEP	2.7543	2.7498	3.2447	3.0103
κ	4.5	4.5	4.5	4.5

Figure 8: ***A set of four tracks, with varying trajectories, compared to their ground truth trajectories. The track information is contained within the table above. Track 1 moves towards the bottom right (of the frame) at an angle $\theta = -\frac{\pi}{5}$. Track 2 moves straight upwards. Track 3 moves to the top left at an angle $\theta = \frac{\pi}{4}$. Track 4 moves to the straight to the left.***



$M = 60$	Track 1	Track 2	Track 3	Track 4
ST-SNR	3	4.9944	6.9889	8.9834
MT-SNR	1.0055	1.0055	1.0055	1.0055
MT-SNR(dB)	0.0238	0.0238	0.0238	0.0238
Streak Length	2.9836	4.9671	6.9507	8.9343
VRMSE	0.1380	0.3038	0.4595	0.6817
PRMSE	2.4255	2.2531	2.3280	2.3540
κ	5.5	5.5	5.5	5.5

Figure 9: ***A set of four tracks, with varying streak lengths, compared to their ground truth trajectories. The track information is contained within the table above. The intensity of the object (ST-SNR) was increased to keep the MT-SNR the same for longer streaks.***

REFERENCES

[1] Y. Bar-Shalom, X. R. Li and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*, J. Wiley and Sons, 2001.

[2] Y. Bar-Shalom, P. K. Willett and X. Tian, *Tracking and Data Fusion: A Handbook of Algorithms*, YBS Publishing, 2011.

[3] Coleman, T. F. and Y. Li. "An Interior, Trust Region Approach for Nonlinear Minimization Subject to Bounds." *SIAM Journal on Optimization, Vol. 6*, 1996, pp. 418-445.

[4] Coleman, T. F. and Y. Li. "On the Convergence of Reflective Newton Methods for Large-Scale Nonlinear Minimization Subject to Bounds." *Mathematical Programming, Vol. 67*, Number 2, 1994, pp. 189-224.

[5] Steven G. Johnson, *The NLopt nonlinear-optimization package*, http://ab-initio.mit.edu/nlopt .

[6] D. R. Jones, C. D. Perttunen, and B. E. Stuckmann, "Lipschitzian optimization without the lipschitz constant," *J. Optimization Theory and Applications*, vol. 79, p. 157 (1993).

[7] Levesque, Martin P., and Sylvie Buteau. Image processing technique for automatic detection of satellite streaks. No. DRDC-V-TR-2005-386. *DEFENCE RESEARCH AND DEVELOPMENT CANADA VALCARTIER* (QUEBEC), 2007.

[8] J. N. Sanders-Reed, Maximum likelihood detection of unresolved moving targets, *IEEE Trans. Aerosp. Electronics Systems*, AES-34(3): 844–859,July 1998.

[9] S. Schoenecker, P. Willett and Y. Bar-Shalom, "ML-PDA and ML-PMHT: Comparing Multistatic Sonar Trackers for VLO Targets Using a New Multitarget Implementation". *IEEE Journal of Oceanic Engrg.*, 39(2):303–317, April 2014.

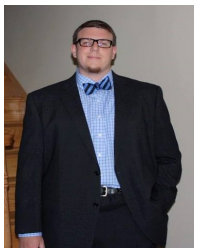
[10] S. Schoenecker, P. Willett and Y. Bar-Shalom, "Comparing Multitarget Multisensor ML-PMHT with ML-PDA for VLO Targets", *Proc. 16th Intn'l Conf. on Information Fusion*, Istanbul, Turkey, July 2013.

[11] S. Schoenecker, P. Willett and Y. Bar-Shalom, "Extreme-Value Analysis for ML-PMHT, Part 1: Target Trackability", *IEEE Trans. Aerosp. Electronic Systems*, 50(4):2500–2514, Oct. 2014.

[12] S. Schoenecker, P. Willett and Y. Bar-Shalom, "Extreme-Value Analysis for ML-PMHT, Part 2: False Track Probability", *IEEE Trans. Aerosp. Electronic Systems*, 50(4):2515–2527, Oct. 2014.

[13] P. Willett , and S. Coraluppi. "MLPDA and MLPMHT applied to some MSTWG data." , *Proc. 9th Intn'l Conf. on Information Fusion*, Florence, Italy, July 2006.

BIOGRAPHY



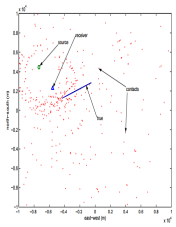
Andrew Finelli *is a PhD student in the Electrical and Computer Engineering department at the University of Connecticut. He graduated with a B.S. degree in Electrical Engineering from the University of Connecticut in 2016. His research interests include estimation theory, target tracking, image processing, and machine learning.*

Yaakov Bar-Shalom *received the B.S. and M.S. degrees from the Technion in 1963 and 1967 and the Ph.D. degree from Princeton University in 1970, all in EE. From 1970 to 1976 he was with Systems Control, Inc., Palo Alto, California. Currently he is Board of Trustees Distinguished Professor in the Dept. of Electrical and Computer Engineering and Marianne E. Klewin Professor in Engineering at the University of Connecticut. His current research interests are in estimation theory, target tracking and data fusion. He has published over 500 papers and book chapters. He coauthored/edited 8 books, including Tracking and Data Fusion (YBS Publishing, 2011), He has been elected Fellow of IEEE for "contributions to the theory of stochastic systems and of multi target tracking". He served as Associate Editor of the IEEE Transactions on Automatic Control and Automatica. He was General Chairman of the 1985 ACC. He served as Chairman of the Conference Activities Board of the IEEE CSS and member of its Board of Governors. He served as General Chairman of FUSION 2000, President of ISIF in 2000 and 2002 and Vice President for Publications in 2004-13. In 1987 he received the IEEE CSS Distinguished Member Award. Since 1995 he is a Distinguished Lecturer of the IEEE AESS. He is co-recipient the M. Barry Carlton Award for the best paper in the IEEE TAESystems in 1995 and 2000. In 2002 he received the J. Mignona Data Fusion Award from the DoD JDL Data Fusion Group. He is a member of the Connecticut Academy of Science and Engineering. In 2008 he was awarded the IEEE Dennis J. Picard Medal for Radar Technologies and Applications, and in 2012 the Connecticut Medal of Technology. He has been listed by academic.research.microsoft (top authors in engineering) as #1 among the researchers in Aerospace Engineering based on the citations of his work. He is the recipient of the 2015 ISIF Award for a Lifetime of Excellence in Information Fusion.*

Peter K. Willett *Peter Willett has been a faculty member in the Electrical and Computer Engineering Department at the University of Connecticut since 1986. Since 1998 he has been a Professor, and since 2003 an IEEE Fellow. His primary areas of research have been statistical signal processing, detection, machine learning, communications, data fusion, and tracking. He is editor-in-chief of IEEE Signal Processing Letters, 2014-2016. He was editor-in-chief for IEEE Transactions on Aerospace and Electronic Systems from 2006-2011, and AESS Vice President for Publications 2012-2014. For 1998-2005 he was associate editor for three active journals - IEEE Transactions on Aerospace and Electronic Systems (for Data Fusion and*

Target Tracking) and IEEE Transactions on Systems, Man, and Cybernetics, parts A and B. He is remains associate editor for the IEEE AES Magazine. He is a member of the IEEE AESS Board of Governors and of the IEEE Signal Processing Society's Sensor-Array and Multichannel (SAM) technical committee (and is now Chair).

Raymond Byrne *is a Distinguished Member of the Technical Staff at Sandia National Laboratories. He holds a B.S. from the University of Virginia, an M.S. from the University of Colorado, and a Ph.D. from the University of New Mexico in electrical engineering (control theory), as well as an M.S. in finance from the University of Chicago. Interests include dim target tracking algorithms and control of power systems.*



David Melgaard *Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.*