

A Hardware Design for Real-Time Multiple Target Tracking

Frederick Ferguson and Chandra Curtis

Center for Aerospace Research, North Carolina A&T State University
Greensboro, NC 27411

ABSTRACT

This paper describes the use of a simulated real time system of a feed-forward neural network, a recurrent neural network and a set of expert rules in solving the problem of Multiple Target Tracking. It is assumed that data is provided in the form of blips, taken off 3 consecutive focal plane arrays, operating at visible or infrared wavelengths. In this paper, the task of multiple target tracking is transformed from one of blip-frame data association to one of target clustering, which in turn is broken down and solved in four stages. Each stage is described and mapped with the use of a feed-forward, a recurrent neural network or a set of fuzzy rules. The first and second stages of the solution procedure involve the use of two feed-forward neural network modules, while the third and fourth stages use a recurrent neural network module and a set of expert rules module. The Multiple Target Tracking solution procedure is simulated through use of a FORTRAN Code. In principle the number of targets that can be tracked with the routine is unlimited. However, in reality, the number of targets is dictated by the number of neurons, which in turn is constrained by hardware requirements. Software simulation results shows that the Multiple Target Tracking code is capable of tracking an arbitrary number of targets very efficiently. The program was tested and debugged for use in the tracking of sets of multiple targets; ranging from 2 to 14. Results indicated that once the average acceleration of the targets is adequately evaluated, track files could be developed with 100% accuracy.

INTRODUCTION

Air and Missile Defense, and Battlefield Awareness face unprecedented challenges in target detection and tracking. Traditional single-sensor systems experience unacceptable performance degradation when dealing with multiple targets in the highly mobile environment in which today's forces must operate. New developments in multiple sensor data fusion are addressing many of these issues and advanced techniques, such as pre-detection tracking, context-sensitive data association and tracking, adaptive data fusion and tightly-coupled fusion and sensor management, are under development.

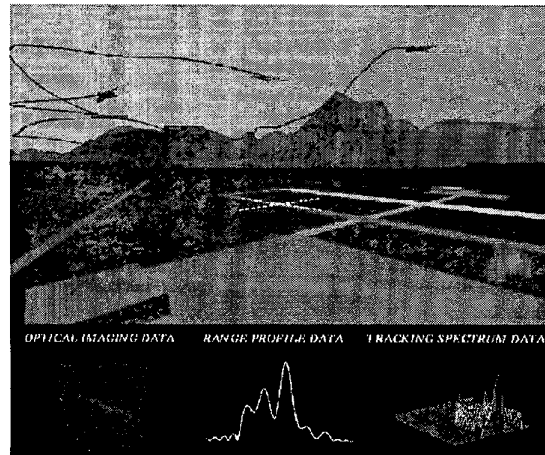


Fig. 1. A Typical Multiple Target Problem [1-3].

One of the key functions performed by a surveillance system is to keep track of all targets of interest within the coverage region of its sensors. For military surveillance systems, the coverage region generally involves several thousand cubic miles containing several hundred targets. The sensors available for use in these surveillance systems operate at visible or infrared wavelengths, and cannot provide perfect information about all targets. In general, sensor measurements of targets tend to be ambiguous, (lack of knowledge as to which target was reported), incorrect (reports of false targets), and imprecise (random errors in the measurements) [4]. A typical multiple target-tracking scenario of interest to our defense forces is illustrated in Figure 1.

A BRIEF REVIEW OF MULTIPLE TARGET TRACKING RESEARCH

Multiple target tracking (MTT) is an essential requirement for surveillance systems employing one or more sensors together with computer subsystems in order to interpret an environment that includes both true targets and false alarms. The objective of MTT [1-6] is to partition the sensor data into sets of observations or tracks produced by the same source. The crux of MTT is to carry out data association process of measurements whose origin is uncertain due to one or all of the following reasons:

- i. random false alarms in the detection process;
- ii. clutter due to spurious reflectors or radiators near the target of interest;
- iii. interfering targets;
- iv. decoys or other counter measures.

The three basic methods for data association are as follows [1-8]:

- i. Nearest Neighbor (NN) method, which is computationally efficient but unreliable for tracking targets in a highly cluttered environment.
- ii. Joint Probabilistic Data association (JPDA), which has its own shortcomings. In tracking closely spaced targets, JPDA delivers poor performance because of the persistent interference from neighboring targets. Also, when the problem size increases, the required computation increases exponentially.
- iii. Multiple Hypotheses Tracking (MHT), which is a multiple scan method and both the memory and computation requirement of which increase exponentially with problem size.

At the Center of Aerospace Research at North Carolina A&T State University research in MTT has been focused on the NN method. Sensor data is provided in the form of blips, taken off three consecutive focal plane arrays, operating at visible or infrared wavelengths. A typical RF-communication unit capable of providing the required data and currently used to characterize sea and surface targets, is illustrated in Figure 2. Our goal is to develop an algorithm that simultaneously traces the paths described by multiple targets in

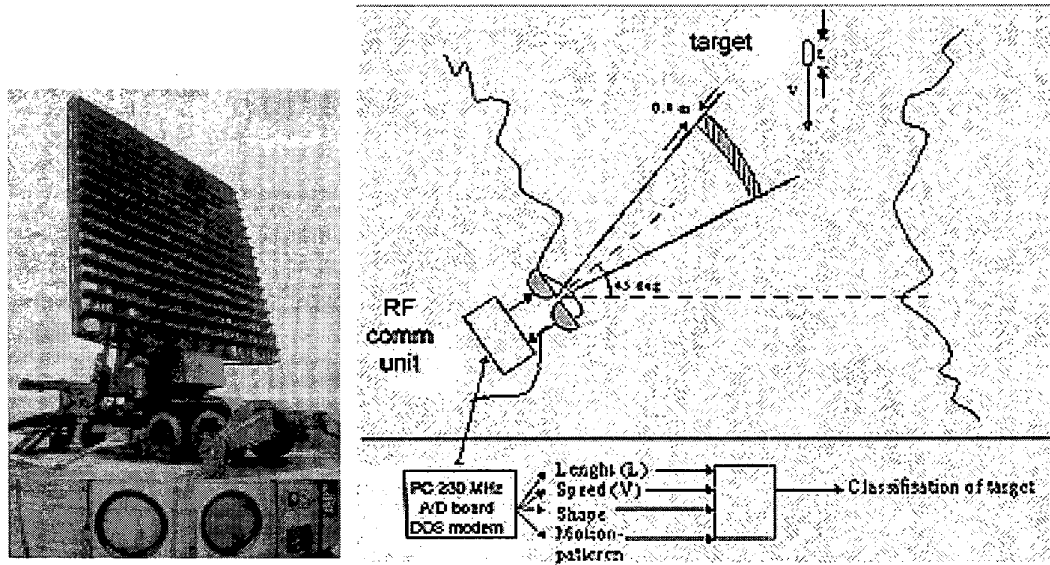


Fig. 2. A typical RF-communication unit used to characterize sea and surface targets.

real time as they traverse a given domain. The algorithm of interest to this study involves the use of the following paradigms; a feed-forward neural network, a recurrent neural network, and a set of fuzzy rules. An interesting advantage of this algorithm is the fact that it can be configured on Digital Signal Processors and used in real time applications.

REAL-TIME MULTIPLE TARGET TRACKING

In a multiple target environment surveillance systems are faced with the prospect of identifying and tracking individual targets, based on models developed from sensor reports and target kinematics. The best choice of target and sensor models will depend on specified missions or applications. Nevertheless, regardless of the choice of sensors and target models, the bottom line capability of any reasonable multiple target tracking algorithm is its ability to trace the appropriate location of all objects of interest over specified time intervals. In general, a simplification of the problem of multiple target tracking can be described as shown in Figure 1.

Over time, target indications move across the field of view of the sensor, and it is required the targets be clustered by their location. To accomplish this task, any multiple target-tracking algorithm must identify, discriminate and locate all targets of interest at specified time instances. However, sensors can only provide focal plane array data of ambiguous targets or blips, i.e., the lack of specific knowledge as to which target was reported, at discreet time intervals. In this analysis, it is also assumed that information on the target location (x, y, z) and target speed with a time resolution of 1/50 second. Available systems [1-4], as illustrated in Figures 1 and 2, capable of delivering these requirements are available off the shelf. Moreover, the surveillance system illustrated in Figure 2 can deliver information on target length, shape, speed, internal motion, sea spectrum and estimated weight. The most common sensors of this type are focal plane arrays, operating at visible or infrared wavelengths. In this paper, it is assumed that each target will be tracked in a 256x256 focal plane array, which can be represented by the Cartesian system of coordinates in the form of blips. A proposed field of view, which includes hardware assembly and data frame, is illustrated in Figure 3.

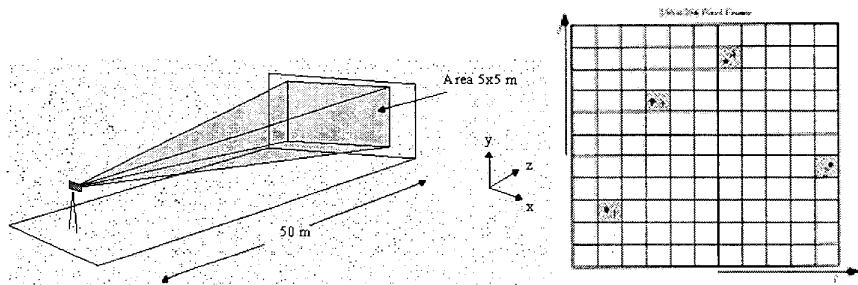


Fig. 3. Illustration of Observation Domain.

The type of preprocessing that is necessary depends greatly on whether the input blip data is already ordered in some fashion. Typical focal plane imagers come out in row scan order. In this paper it is assumed that blip data is extracted and labeled from each pixel, based on the rule of fixed row with varying columns. Only the pixels containing objects are labeled. Results of the blip/frame data extraction using this method are illustrated in Table 1.

Table 1: Typical Ambiguous Data Clusters

Blip Label	Blip x Location	Blip y Location
1	X_1	Y_1
2	X_2	Y_2
...
n	X_n	Y_n

DATA PROCESSING DURING MULTIPLE TARGET TRACKING

The task of real time multiple target tracking is equivalent to the re-organization of the input blip/frame data into track files. The mapping of input to output is based on the use of three very important mathematical paradigms, namely, the feed-forward neural network, the recurrent neural networks, and a set of fuzzy rules. The uses of these paradigms in the multiple target tracking algorithm presented herein is described in the following five sub-sections. The mapping of the input data to track files, and the details of the mapping process are illustrated in Figure 4.

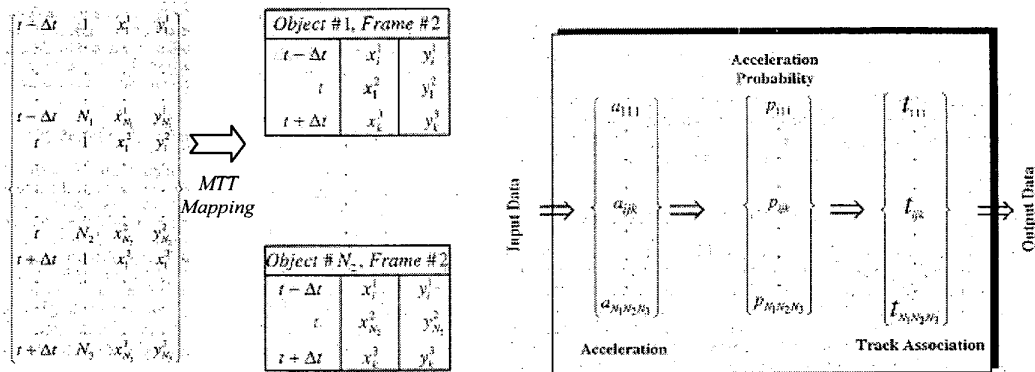


Fig. 4. The three stages of input to output MTT data mapping.

Elementary Data Unit for Multiple Target Tracking

The algorithm described herein is based on 3 frames of data, which is provided by a sensor capable of mapping a two-dimensional image of the observation. No depth information is needed, only speed and/or acceleration. Figure 5 illustrates a single data plane, and its use with a previous and past neighbors forming a basic unit of three consecutive data frames from which appropriate track files are developed.

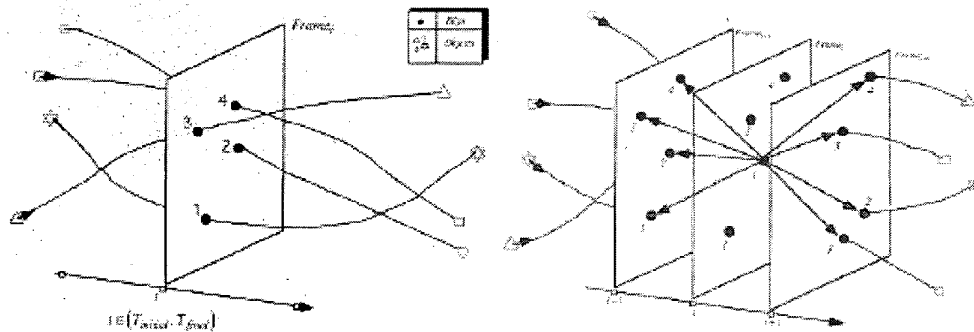


Fig. 5. Consecutive arrays of ambiguous targets.

Track files contains information about the targets and their location history. The successful tracking of a given target specified by the i^{th} blip in frame², involves the identification of two corresponding blips located in frame¹ and frame³ respectively. When i^{th} blip in frame² combines with the identified blips in frame¹ and frame³ the result is the most likely path of the target for the duration of time, $t-dt$, through time, $t+dt$.

Notation Convention for the Multiple Target Tracking Process

In figure 5, and through out this paper a convention on the use the indices i, j , and k is adopted. The indices i, j , and k are such, that, $i, i = 1, N_1$; represents any blip in frame¹, at time $t-dt$; $k, k = 1, N_2$ represents any blip in frame², at time t ; and $j, j = 1, N_3$, represents the blips in frame³, at time $t+dt$. Besides, the indices, i, j , and k carry a special implication when used with the coordinate of a vector. Under the terms of this convention any parameter or variable labeled by the indices, i, j , and k , refers to all of the possible combination of the three indices as they vary independently from their minimum to maximum values. As a result any parameter labeled by the indices, i, j , and k , refers to a vector of $n_1 n_2 n_3$ coordinates. A coordinate of the vector described by the indices, i, j , and k , will be described by the expression, $P_{m:ijk}$, where the index m varies from 1 through $n_1 n_2 n_3$.

The notations adopted in this analysis are outlined as follows:

Convention # 1: P_{ijk} = vector of $n_1 n_2 n_3$ coordinates

Convention # 2: $P_{m:ijk}$ = m^{th} coordinate of vector P_{ijk}

Acceleration and Probability Mapping with Feed-Forward Neural Networks

In this analysis it is assumed that the time interval between frames is uniform, and moreover, that the frame rate is high enough so that differentiation can be conducted by finite differences. With these assumption, the velocity vector, V , can be evaluated with any two consecutive frames, using the expression,

$$V_{jk} = L_k - L_j \quad 1.$$

Vector, L , represents the two dimensional blip coordinates, x and y . The indices i, j , and k represents the appropriate frames in which the coordinates are taken. On the other hand, the acceleration vector, A_{ijk} , is evaluated with three frames, using the expression,

$$\begin{aligned} A_{ijk} &= V_{jk} - V_{ij} \\ &= L_k - 2L_j - L_i \end{aligned} \quad 2.$$

In the expressions given in equations (1 - 2), the notation convention adopted earlier is used. The acceleration vector, A_{ijk} , consists of $n_1 n_2 n_3$ coordinates, each of which is evaluated using one of the possible combinations of the indices, i, j and k . It is of interest to note that the vectors, L, V and A , are described by 1, 2, and 3, subscripts, respectively. This fact indicates that whereas, the velocity vector can be described by only 2 frames, the acceleration vector must be described with the use of 3 frames.

Since this analysis deals with the 'angle-only' or the two dimensional model for target tracking, the acceleration vector, A , has two components, one along each axis in the Cartesian system of coordinates. As a result, the evaluation of the acceleration vector, A_{ijk} , can be conducted with the aid of two feed-forward neural network modules, as follows

$$A_{ijk} = \sqrt{(a_{ijk}^x)^2 + (a_{ijk}^y)^2} \quad 3.$$

Parameters, a_{ijk}^x and a_{ijk}^y , represent the components of the acceleration vector. Equation (3) can be evaluated with the use of a single feed-forward neural networks. However, for real time on board analysis, two neural networks module are recommended.

Probability Mapping with Feed-Forward Neural Networks

The kinetic model used in this analysis is based on the acceleration association probability. Having a priori determined a mean acceleration value, \bar{A} , for the targets of interest, the normal probability density function for the acceleration vector gives,

$$P = e^{-(A-\bar{A})^2} \quad 4.$$

The vector \mathbf{P} will be referred to as the acceleration association probability vector. Again, in expression (4), the notation convention adopted earlier is used. The acceleration association probability vector \mathbf{P} , consists of $n_1 n_2 n_3$ coordinates P_{ijk} , each of which is evaluated using one of the possible combinations of the indices, i, j and k . Evaluating the acceleration association probability coordinates can be accomplished real time with the aid of a feed-forward neural network module.

Track Association using Recurrent Neural Networks

The recurrent neural network is merely a single layer feed-forward network with feedback connections of the network output channels bridging its input channels. The dynamics of the recurrent network is equivalent to the evaluation of an equation of the type, described by the following equation:

$$t_{ijk}^{s+1} = g \left(\sum_{l=1}^{n_1 n_2 n_3} w_{lm} t_{m,ijk}^s \right) + p_{ijk} \quad 5.$$

where the variables; w_{lm} represents the network weights, $P_{m,ijk}$ the input acceleration probabilities, t_{ijk}^s the track association vector and the function, g , represents the nonlinear activation or threshold function used in this analysis. Once again, the notation convention adopted earlier is used in equation (5). In this report the threshold function takes the form of a piecewise linear sigmoid function.

The very first thing to be defined when using the recurrent neural network is the network weights. Typically, the network weights are determined by differentiating the network energy, E , or cost function with respects to the acceleration association probability. The shape and state of this energy function is determined by the network initial input and the interconnection constraints and strengths of its neurons, as follows:

$$E = \beta \sum_{ijk} \sum_{i'j'k'} t_{ijk} t_{i'j'k'} \chi(\delta_{ii'} \delta_{jj'} \delta_{kk'}) + \alpha \sum_{ijk} p_{ijk} t_{i'j'k'} \quad 6.$$

Here t_{ijk} represents the tracks, p_{ijk} defines the probability of the ijk track, δ_{ij} is the Kronecker Delta function, α and β are the optimization parameters and χ is the Exclusive OR function. The Exclusive OR function, χ , is defined as follows

$$\chi(x, y, z) = \begin{cases} 1; & \text{if } \begin{cases} x, y = 1, & z \neq 1 \\ x, z = 1, & y \neq 1 \\ y, z = 1, & x \neq 1 \end{cases} \\ 0; & \text{otherwise} \end{cases} \quad 7.$$

When equation (7) is twice differentiated with respect to $t_{m,ijk}$, the appropriate weights to the recurrent neural network is found in the form,

$$w_{lm} = -\beta [\chi(\delta_{ii'}, \delta_{jj'}, \delta_{kk'})]_{l,ijk}^{m,i'j'k'} \quad 8.$$

The range of values for β used in this analysis is: $0 < \beta < 0.50$. Once activated the network converges toward a minimum of the energy function in which some neurons are on and others are off. The set of neurons that are on at convergence represents the group of tracks that are part of a final solution. Within the context of the multiple target tracking problem, the network weights also enforces the constraints of having one corresponding blip per frame per track.

Association Probabilities to Track Files Mapping

Once activated the recurrent neural network converges toward a minimum of the energy function, E , in which some neurons are on and others are off. The set of neurons that are on at convergence represents the

group of tracks that are part of a final solution. At convergence the track vector, t_{ijk} , of $n_1 n_2 n_3$ coordinates will have values clustered around the values of zero and one. Only the coordinates with values that are clustered in the upper range are of interest. The next step in the evaluation process is to identify the coordinates with values that are clustered in the region of unity and represents these tracks in terms of targets located in frames 1, 2, and 3.

TECHNICAL RESULTS

When properly initiated the simulated real time multiple target tracking accepts data in the form of 3 consecutive blip-frame data sets. The data is processed and track files are developed. A FORTRAN Code was developed to simulated the functions of the two feed-forward neural networks, the recurrent neural network and the set of expert rules modules. The code for multiple target tracking is program according to the flow chart illustrated in Figure 4. Results from observing the tracking algorithm at work with four objects on a 256x256 focal array plane over 30 frames are illustrated in Figures 6. As indicated by these results the multiple target-tracking algorithm is capable of tracing very complex movements, even when there are crossover in motions.

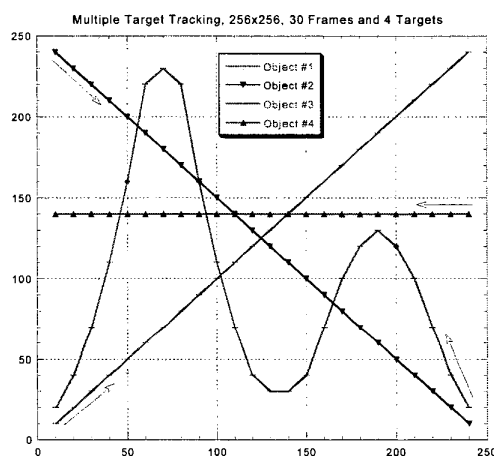


Fig. 6. Results from tracking four targets.

In principle the number of targets that can be tracked with this algorithm is unlimited. However, in reality, the number of targets is dictated by the number of neurons, which in turn is constrained by hardware requirements. Software simulation results showed that the MTT algorithm is capable of tracking an arbitrary number of targets very efficiently. The program was tested and debugged for use in the tracking of sets of multiple targets; ranging from 3 to 14. Preliminary results indicated that once the average acceleration of the targets is adequately evaluated, track files can be developed with 100% accuracy.

REFERENCES

1. R.E. Bethel, and G.J. Paras, A PDF Multisensor Multitarget Tracker. *IEEE Log No. T-AES/34/1/00185*
2. D.R. Zahiriak, D.L. Sharpin and T.W. Fields, A Hardware-Efficient, Multirate, Digital Channelized Receiver Architecture. *IEEE Log No. T-AES/34/1/00184*
3. P.E. Pace, B.H. Nishimura, W.M. Morris and R.E. Surratt, Effectiveness Calculations in Captive-Carry HIL Missile Simulator Experiments. *IEEE Log No. T-AES/34/1/00183*
4. E. Mazor, A. Averbuch, Y. Bar-Shalom and J. Dayan, Interacting Multiple Model Methods in Target Tracking: A Survey. *IEEE Log No. T-AES/34/1/00182*
5. B. Armstrong and B.S. Holeman, Target Tracking with a Network of Doppler Radars. *IEEE Log No. T-AES/34/1/00176*
6. R. K. Saha and K. C. Chang, An Efficient Algorithm for Multisensor Track Fusion. *IEEE Log No. T-AES/34/1/00189*

7. I Ishii, Y. Nakabo and M. Ishikawa, Target Tracking Algorithm for 1ms Visual Feedback System Using Massively Parallel Processing. *IEEE Int. Conf. Robotics and Automation (Minneapolis, 1996.4.25)/Proc. IEEE Int. Conf. Robotics and Automation*, pp.2309-2314
8. J.F. Puztaszeri, P.E. Rensing and T.M. Liebling, 1996. Tracking elementary particles near their primary vertex: a combinatorial approach. *Journal of Global Optimization*, 9: 41-64.