MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY

AN ALGORITHM FOR MULTIPLE TARGET TRACKING AND DATA CORRELATION

C.B. CHANG
L.C. YOUENS
Group 32

TECHNICAL REPORT 643

13 JUNE 1983

Approved for public release; distribution unlimited.

LEXINGTON MASSACHUSETTS

ABSTRACT

An algorithm for multiple target tracking and data correlation is described. A general description of the problem and solution is first given. More specific discusssions on tracking with a passive infrared sensor then follow. An example is presented to illustrate the trade-off between algorithm complexity, performance, and processing requirements.

TABLE OF CONTENTS

I. INTRODUCTION

Multiple target tracking is a classical problem with both civilian and military applications. Air traffic control is a notable civilian application area. Military applications range from air, ocean, and ground surveillance to missile defense. Due to a drastic increase in target density in recent years, this subject area has been a center of discussions in the open literatures [1]-[6] and company reports [7]-[10]. Interested readers can find special sessions in the conference proceedings of recent IEEE Conferences on Decision and Control and several articles for specific applications in the IEEE Transactions on Aerospace and Electronic Systems.

Motivated by the application for exoatmospheric Ballistic Missile Defense (BMD) in an extremely high target density environment, we have studied and obtained an algorithm for multiple target tracking. References [7]-[10] contain specific algorithms for exo-BMD applications. Due to the nature of this problem it is often referred to as the scan-to-scan correlation (SSC) problem in the BMD community. Although our algorithm bears this specific application in mind, its concept is rather general and it can be easily extended for a general multiple target tracking and data correlation application. Our algorithms shares some features with those in the references, it also has some unique characteristics.

We will point out differences between our algorithm and those of [7]-[10] whenever it is appropriate. In this section, we will briefly review the open literatures in this area. Reference [1] is a recent and complete description of a multiple target tracking algorithm. Our mild reservations about this paper are (1) it does not consider the effect of limited sensor resolution and (2) it attempts to model every stage of the target-tracking process such as the a priori target distribution and the probability of a given number of detections occuring where in reality, these probabilities may only be vaguely known. Reference [2] is a survey article References [3] and [4] discuss the problem of track maintenance in a dense target (or cluttered) environment. What is missing is a critical stage of the process, track initiation. The subject of track initiation is covered in [1] and [5]. Reference [6] discusses a probabilistic data association scheme which can be shown to be a special case of the algorithm discussed in [3]-[4]. We will point out more specifics related to those approachs in the next sections.

This report is organized as follows. The next section will give a very general discussion of the hypothesis tree approach to the multiple tracking problem. Section 3 contains details of our algorithm. This algorithm attempts to realize the approach discussed in the section 2 whenever it is determined feasible. An example illustrating the algorithm discussed in this report is given in Section 4. A summary is given at the Section 5. Two appendices, discussing the polynomial fit formulaes and the batch estimator used in this report, are given at the end.

II. GENERAL DISCUSSION

The multiple target tracking problem can be divided into two phases. The first phase is track initiation and the second phase is track maintenance. They are discussed individually below.

2.1 Track Initiation

Consider the case of a scanning sensor. The first and second scan produce N_1 and N_2 detections, respectively. The problem is to associate the two sets of detections to form $\min(N_1,N_2)$ number of track files. Notice that we have assumed that $N_1 \neq N_2$. This can be caused by (1) imperfect detection and resolution, (2) emergence of new targets in the second scan, and (3) targets leaving the sensor field of view before the second scan. In the following, an approach for track initiation with k scans of data is described.

Let Z_k denote all the measurements (N) collected during k-th scan, i.e.,

$$Z_k = \{\underline{z}_1(k),\ \underline{z}_2(k), \ldots, \underline{z}_N(k)\} \qquad (2.1)$$

Let Z^k denote the set of measurements up to and including the k-th scan, i.e.,

$$Z^k = \{Z_i;\ i=1,\ldots,\ k\} \qquad (2.2)$$

For simplicity, we assume that N is the number of detections for all Z_k's. Assume also that the sensor has perfect target detection. When this is not true, one has to enumerate more hypotheses to account for all possibilities. With Z^k, there can be N^k combinations of measurement sequences and each measurement sequence represents a possible track. Let each possible combination be denoted by a hypothesis, $H_{m_k}(k)$ which is defined by

$$H_{m_k}(k) = \{\underline{z}_{n_1}(1), \underline{z}_{n_2}(2), \ldots, \underline{z}_{n_k}(k)\} \qquad (2.3)$$

Suppose that a tracking filter is applied to process each possible measurement sequence. The a posteriori hypothesis probability of $H_{m_k}(k)$ being true can be computed recursively using

$$P(H_{m_k}(k)/Z^k) = \frac{p(\underline{z}_{n_k}(k)/H_{m_{k-1}}(k-1), Z^{k-1})}{p(z_{n_k}(k)/Z^{k-1})} P(H_{m_{k-1}}(k-1)/Z^{k-1}) \qquad (2.4)$$

where $p(z_{n_k}(k)/H_{m_{k-1}}(k-1), Z^{k-1})$ is the probability density of the residual from the tracking filter using $H_{m_{k-1}}(k-1)$ and $z_{n_k}(k)$*. The above equation can be derived as a

*A more parametric approach for modelling this probability density function is given in Refs. [1], [2] and [5] in which situations including a priori target distribution and the probability of a given number of detections were also considered.

special case of the results presented in [13],[14]. The final set of tracks (total N) can be chosen as those N feasible hypotheses with the largest hypothesis probabilities, i.e.,

$$\max \left\{P(H_{m_k}(k)/Z^k);\ m_k = 1,\ldots,N^k\right\} \qquad (2.5)$$

$$\left\{N;\quad H_{m_k}(k)\in\mathcal{F}\right\}$$

where the feasible set, $\mathcal{F}$, is the restriction that each measurement at a given time can be used only once, i.e.,

$$\mathcal{F} = \left\{H_{m_k}(k):\quad H_{i_k}(k) \cap H_{j_k}(k) = \phi \text{ for } i \neq j\right\}$$

The computational requirement of the above method is clearly non-trivial. In fact, the above optimization problem defines a N-dimensional assignment problem. A well known solution to the 2-dimensional assignment problem is the Hungarian algorithm [15]. To the best of the authors' knowledge, the N-dimensional extension of the Hungarian algorithm is not yet available.

In many applications, one may be able to pre-cluster the detections so that search over the entire set of detections is not necessary. Other physical constraints can sometimes be imposed to reduce the search requirements depending upon given systems and applications.

A similar approach using a maximum likelihood method was described in [5] in which the multidimensional search problem was reduced to a 0-1 integer programming problem.

2.2 Track Continuation

Once track files have been established, the computational requirement is greatly reduced. This is because for each track file one is only required to search the "admissible" region dictated by the covariance of the filter residual process.

We note that a slightly modified method of the track inititation algorithm discussed in 2.1 can be applied to the track maintenance problem. That is, one establishes a new hypothesis for each detection resident in the admissible region. This procedure results in an exponentially growing number of track files. One can inhibit the growing memory and computational requirement by selecting a tree depth and conducting a global search for a set of feasible tracks having the highest hypothesis probabilities (eqs. (2.5), (2.6)). Another approach is to combine a set of "most likely hypotheses" growing out of the same track file using the weighted sum of state estimates with the hypothesis probabilities as weighting factors. This second approach is the basis of the Bayesian tracker presented by Singer et. al. [3], [4]. If the depth is equal to one, i.e., one combines all admissible detections at each scan, then one obtains the probabilistic data association filter of Bar-Shalom and Tse [6]. We exmphasize however, that the approaches of [3], [4], and [6] are suitable for tracking in a cluttered environment and do not directly address the multiple target tracking issue.

The concept discussed above constitutes the basis of the hypothesis tree approach to multiple target tracking. To exactly implement the above algorithm however, will result in excessively high computational requirements. For example, finding the optimum solution of a N-dimensional assignment problem (for N being large) is unpractical. A suboptimal but computationally more feasible solution is therefore desirable. In the next section, we present analgorithm which is developed specifically for ballistic missile defense application with a passive infrared sensor. Several concepts discussed however, are useful for a larger class of multiple target tracking and data correlation problems. These concepts will be identified as we move along in discussion.

III. ALGORITHM DESCRIPTION

3.1 Introduction and Basic Assumptions

Consider the situation that there are N_1 and N_2 detections in the first and second scans (or called "frames" for an optical sensor), respectiely. In a simple problem for which the target motion is insignificant between two scans or the relative motion among targets is small (such that the target pattern is preserved), then one can apply a two-dimensional assignment method for correlating measurements of these two scans.* Entries of the assignment matrix can be that of eq. (2.4) with k=2. For Gaussian measurement vectors, one may use the weighted distances

$$\ell_{ij} = (\underline{z}_i(1)-\underline{z}_j(2))^T(R_i(1)+R_j(2))^{-1}(\underline{z}_i(1)-\underline{z}_j(2)) \qquad (3.1)$$

as entries where $\underline{z}_i(k)$ is the i-th measurement of the k-th scan with measurement covariance $R_i(k)$. Once measurement of two scans have been correlated, a velocity vector can be established making the correlation with measurements of the third frame somewhat easier. For an optical sensor measuring line-of-sight angles, this velocity vector will initially be limited to the angle domain while a radar sensor can give a three dimensional velocity vector.

The tracking problem discussed in this report is

*This is similar to a two sensor measurement correlation problem discussed in [11].

more complicated than that described above. Target motion, density and limited sensor resolution are such that target patterns are not preserved in successive scans. In this case, one must apply more knowledge about the target motion dynamics and use more scans of data to identify a string of successive measurements representing the same target.

In the following two subsections, we will discuss the problem of track initiation and continuation individually. An overall description is given in Table 3.1.

3.2 Track Initiation

The most crucial and difficult part of the multiple target tracking problem is track initiation. Specifically for the optical sensor tracking problem, the following factors further complicate the issues:

(1) The target angular velocities vary over a wide range of values making the acceptance cell on the second frame large resulting in a large number of false correlations.

(2) With angle-only (passive receiver) measurements, the target range estimate can not be readily obtained making impossible the use of precise ballistic equations of motion as target dynamics.

In the case of radar tracking, the first point above may still be true depending on target velocity and data rate. The second point above is at least partially true since at the initial stage, a large number of track files are false, using

TABLE 3.1

ALGORITHM DESCRIPTION

Functions	Description
Initiation	
No. of Scans	5-8
First Frame Angular Velocity	A Priori Knowledge for some targets, recursively search for parallel targets.
Prediction	0th - 3rd order polynomial
Correlation	(1) Track split (2) Chi-square test (3) Pattern match test
Continuation	
Prediction	Target equation of motion or polynomial dynamics
Search Bin Size	Filter covariance and model error analysis
Correlation	(1) Track split (2) Pattern match test

the exact target dynamics at this time is very time consuming. For these reasons, we use

(1) a general/parallel search scheme for reducing the number of false correlations and

(2) a polynomial function in each angle domain as target dynamics to simplify calculations.

Furthermore, a sliding window scheme is employed to initiate tracks for new detections in each frame. It is important to note that this is an iterative rather than recursive method and data from many frames must be saved.

The track initiation logic is illustrated in Fig. 3.1. We use the following steps to illustrate the general and parallel search scheme.

(1) For a given detection in the first frame (called an initiator), draw an acceptance region centered at this detection. Detections on the second frame which fall into this region form potential tracks. The size of the initial acceptance region is determined by the maximum target angular velocity. Usually a large number of potential tracks result for one given initiator.

(2) Apply the straight line extrapolation scheme (see Appendix A) to extend all potential tracks into the third frame. The size of the acceptance region at the third frame is determined by model and measurement errors of the linear extrapolation. This acceptance region size is usually much smaller than that of the first step above.

(3) Apply a second order polynomial (see Appendix A) to extend tracks into the fourth frame. Similarly, the acceptance region size further reduces. Tracks which do not receive a detection in their acceptance region will be dropped. Tracks receiving multiple measurements in their acceptance region will be split.

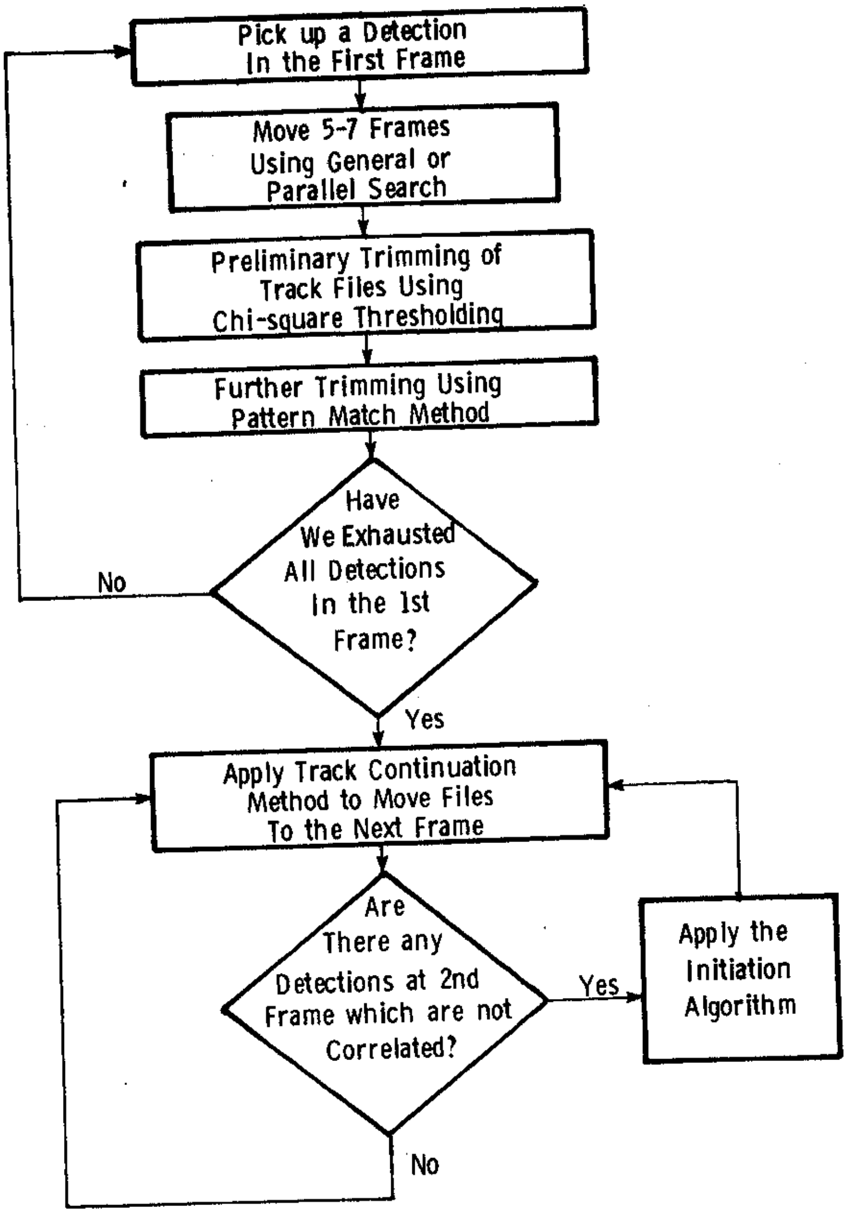

Pick up a Detection In the First Frame
Move 5-7 Frames Using General or Parallel Search
Preliminary Trimming of Track Files Using Chi-square Thresholding
Further Trimming Using Pattern Match Method
Have We Exhausted All Detections In the 1st Frame?
No
Yes
Apply Track Continuation Method to Move Files To the Next Frame
Are There any Detections at 2nd Frame which are not Correlated?
Yes
Apply the Initiation Algorithm
No

Fig. 3.1. Track initiation logic.

(4) Continue to a total of 5 to 8 frames depending upon the target density and scenario. At the end of this stage, usually only a few potential tracks remain. Final choices of tracks are selected using a global polynomial fit. Those tracks with weighted residuals (Chi-squares) below a threshold will all be retained. This completes the general search for an initiator.

(5) Go back to the first frame, assuming that targets in the neighborhood of the initiator will travel in nearly the same direction; one therefore only has to search for detections in the successive frames in parallel with the track(s) established with this initiator. This step greatly reduces the computational and memory requirements. This step is called parallel search.

(6) Once all parallel tracks have been found, go back to the first frame, find another detection which has not been included in any tracks to use as a new initiator for general search.

(7) Repeat until all detections of the first frame have been exhausted.

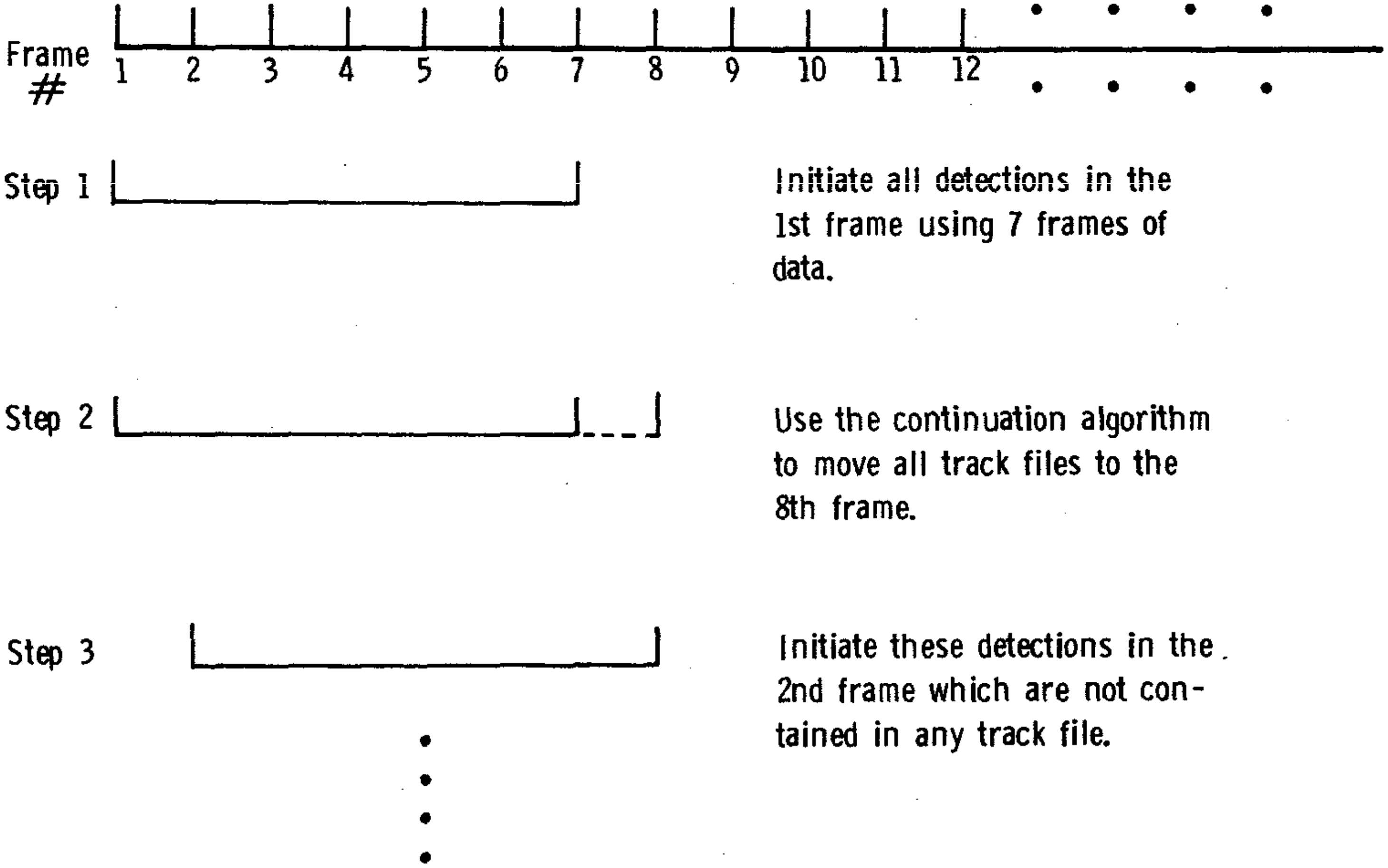
Once the initial correlation described above has been completed, one can trim track files by applying a n-th order polynomial (n is determined by a particular application) fit to measurements of a file and reject those files with excessively high "chi-square" values. A third correlation method listed in Table 3.1 under track initiation is called "pattern match test". This test is the same as the one used in the track continuation. We therefore defer its explanation until the next subsection.

The above procedure is applied in a sliding window fashion so that measurements not included in the track file are used to initiate new tracks. This is illustated in Fig. 3.2.

3.3 Track Continuation

The track initiation process correlates measurements over 5 to 8 frames to produce track files using data of all frames (therefore a smoothing process). The track continuation stage can be a traditional prediction, correlation and updating process. As shown in Table 3.1, the prediction step may use either target equations of motion or polynomial dynamics for reducing the computational burden. In the exoatmospheric BMD application, we have found that the use of a precision tracking filter with the complete target equations of motion greatly enhances the performance. We have implemented both the extended Kalman filter and the batch filter described in [16] for track continuation. The batch filter is also briefly reviewed in Appendix B for the purpose of quick reference.

We use Figure 3.3 to illustrate some typical situations encountered in the track continuation process. As a track moves along, if multiple detections are encounted, the track is split (case 1). If no detections are found for several frames in a row, the track is dropped (cases 1 and 2). There may also be situations for which a track is split and then merged (case 3).



Step 1: Initiate all detections in the 1st frame using 7 frames of data.

Step 2: Use the continuation algorithm to move all track files to the 8th frame.

Step 3: Initiate these detections in the 2nd frame which are not contained in any track file.

Fig. 3.2. Sliding window initiation logic.

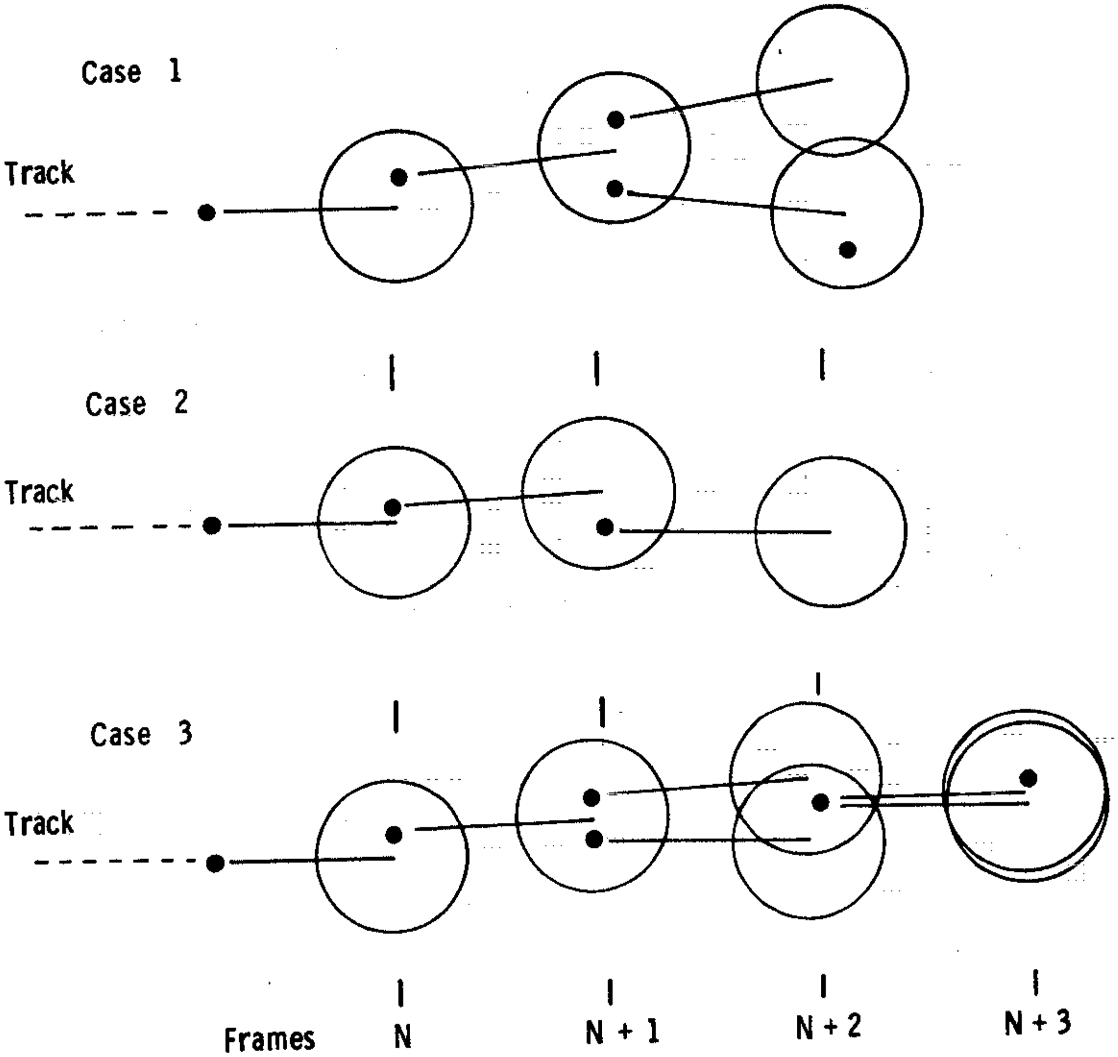


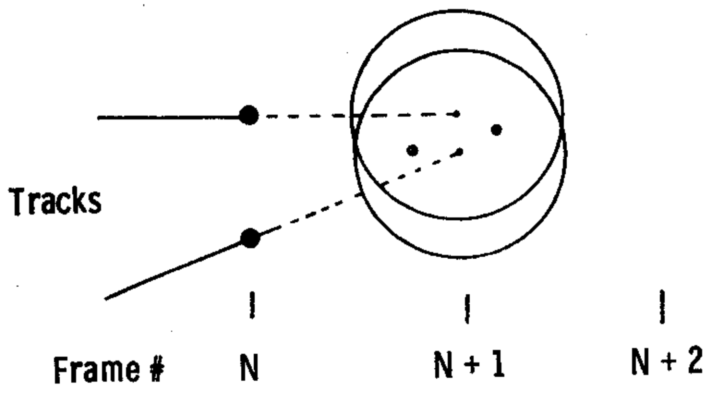
Fig. 3.3. Typical situations encountered in track continuation.

Ambiguities may arise when a number of track files have overlapping acceptance regions and share the same detections. This is illustrated in Figure 3.4 for the case where two tracks share two measurements. Problems of this kind are similar to the assignment problem in operations research. The optimal (in the sense of minimum sum of weighted distances) solution is usually obtained with a so-called Munkres' algorithm (see [11] and [15]). If one attempts to resolve this ambiguity at the frame where it is encountered, this is called immediate conflict resolution. Since the track formation problem is really a multiple dimensional assignment problem (eq. (2.5)), a more reliable decision can be obtained by deferring decisions until further measurements have been received. This is called deferred conflict resolution. A tradeoff for these methods is computation vs performance. We have implemented both the immediate conflict resolution method and the one frame deferred conflict resolution method. Later numerical results will compare the performance of these two methods.

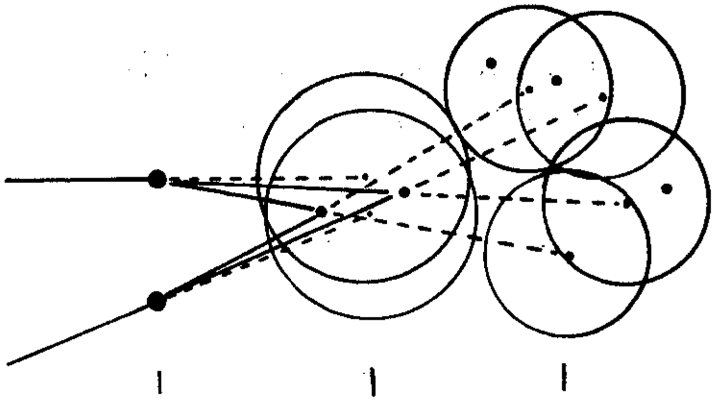
We use Fig. 3.5 to further illustrate the conflict resolution methods. In Fig. 3.5a two track files with measurements up to the N-1 frame are extrapolated to Nth frame and found to compete for measurements a and b. To resolve this conflict immediately is to first form a distance matrix

TR-643(3.4)

▲ Immediate Resolution



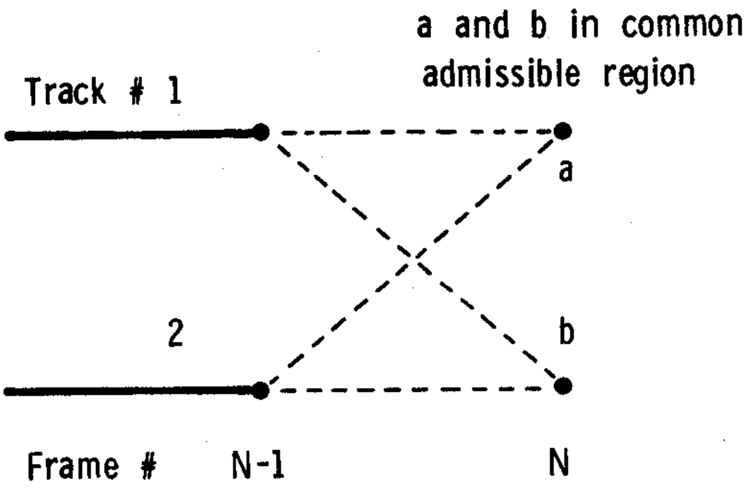
▲ Delayed Resolution



+ Using Interpolated Estimate to Resolve the Ambiguity of the Previous Frame

Fig. 3.4. Track continuation ambiguity resolution.

TR-643(3.5a)

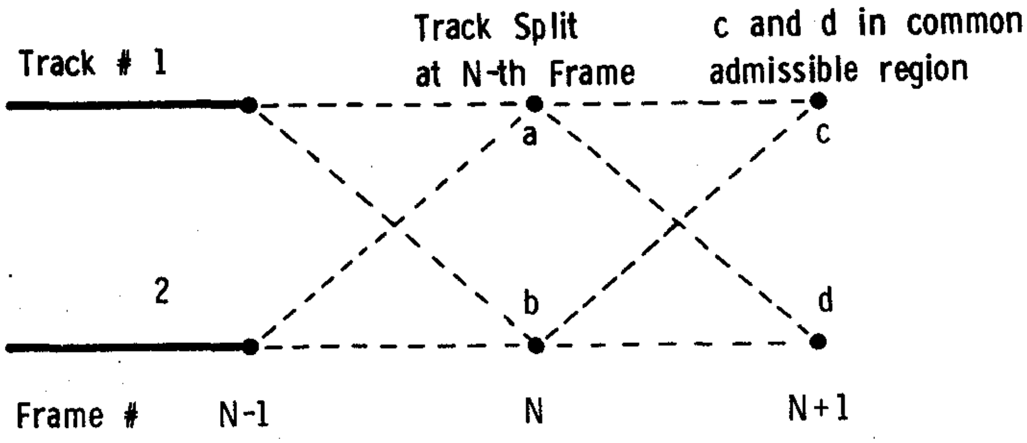


Pattern Match Matrix:

$$\begin{array}{cc} & \text{Measurements} \\ & \begin{array}{cc} a & b \end{array} \\ \text{Track \#} \begin{array}{c} 1 \\ 2 \end{array} & \begin{bmatrix} D_{1a} & D_{1b} \\ D_{2a} & D_{2b} \end{bmatrix} \end{array}$$

Fig. 3.5(a). Illustration of ambiguity resolution technique: immediate resolution method.

TR-643(3.5b)



Two-Layered Pattern Match Matrix:

For Measurement c

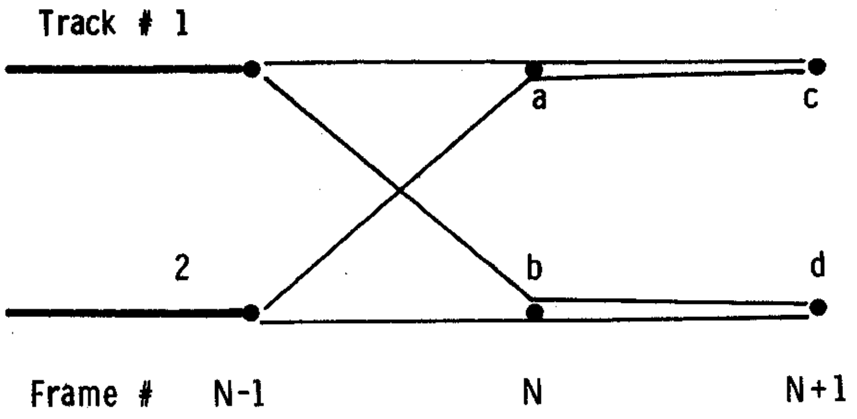
$$\begin{array}{ccc} & \multicolumn{2}{c}{\text{Measurements}} \\ & a & b \\ 1 & D_{1ac} & D_{1bc} \\ 2 & D_{2ac} & D_{2bc} \end{array}$$

For Measurement d

$$\begin{array}{ccc} & \multicolumn{2}{c}{\text{Measurements}} \\ & a & b \\ & D_{1ad} & D_{1bd} \\ & D_{2ad} & D_{2bd} \end{array}$$

Fig. 3.5(b). Illustration of ambiguity resolution technique: multiple-layered pattern matching.

TR-643(3.5c)



Pattern Match Matrix (To resolve measurements of N-th frame using measurements up to N+1st frame)

Measurements

$$\begin{array}{cc} & \begin{array}{cc} a & \quad\quad b \end{array} \\ \begin{array}{c} 1 \\ 2 \end{array} & \begin{bmatrix} X_{1a} & X_{1b} \\ X_{2a} & X_{2b} \end{bmatrix} \end{array}$$

X_{ij}: The smallest residual of track i going through measurement j (of N-th frame) to end at the N+1st frame.

Fig. 3.5(c). Illustration of ambiguity resolution technique: an one-frame deferred resolution method.

shown in Fig. 3.5a. For example, D_{1a} is the weighted distance between the predicted location of the first track and measurement a. The "optimum" solution is the pair of D_{ij}'s that gives the minimum sum (see for example, [11]). This is the "immediate" resolution method.

If one simply splits all tracks and moves to the N+1st frame (Fig. 3.5b) and assumes that there are again two measurements (c and d) falling into the common admissible region, there can be a total of eight possible tracks formed. The "optimum" approach for this problem is to first form a "two-layered" matrix with dimension (2x2x2). The entries of this matrix are the posteriori hypothesis probabilities shown in (2.4). The "optimum" solution is the pair of entires with minimum sum. Two factors will have to be considered: (1) the computation of (2.4) is tedious especially when the numbers of track files and measurements involved are large and (2) an efficient algorithm (equivalent to the Hungarian Method for searching the optimum solution) has not been found. For the above reasons, we device a suboptimal procedure which is particularly efficient when the numbers of track files and measurements are large. Notice that there are two possible tracks to go from track file #1 through measurement a of the Nth frame, namely (1,a,c) and (1,a,d) (see Fig. 3.5c). We first select a track among these two with the smallest residual. This procedure is repeated for all track files and all measurements in the Nth frame. The residuals at the Nth frame (a smoothed residual) of all selected tracks are used to form the distance matrix (Fig. 3.5c). The final selection is the pair of tracks with the minimum sum of entries of the distance matrix. This is the one-frame deferred resolution method. Notice that the one frame deferred method is analogous to the one-step lagged fixed-lag smoothing.

Notice that the above is attempting to reduce a N-dimensional (N=3 for this case) assignment problem to a 2-dimensional assignment problem. This procedure although suboptimal, is straightforward to implement and gives better performance than the immediate resolution method (see section 4).

The deferred ambiguity resolution method can be implemented in a sliding window fashion, i.e., one uses data in the past and future for resolving the conflict of the current frame. This method is also used by the track initiation process right after the "Chi-square" test (see Section 3.2 and Table 3.1). It is referred to as the pattern match test in Table 3.1.

3.4 Track File Smoothing

The track initiation algorithm described above is a smoothing algorithm because it uses all past data in a batch processing mode. All past data will therefore have to be stored for this purpose. If one exactly implements eq. (2.4) for track initiation, this process can be recursive although the number of tracks can grow out of hand rather quickly. The track continuation algorithm using the one frame deferred resolution method is a one step lagged *fixed-lag smoothing* process; one is only required to store the immediate past set of measurements.

In the exoatmospheric BMD application where a passive optical (Infrared) sensor is used for target tracking,a conventional recursive filter (e.g., the extended Kalman filter) may not work satisfactorily, [16]. Instead, a maximum likelihood estimator based batch estimator is found to give near optimum performance (see [16] and also Appendix B of this report for the batch algorithm). This estimator however requires that all measurements be saved. Suppose that all past measurements have been saved, then one can further edit track file measurements when the batch filter is being used to process these track files. This process is illustrated in Fig. 3.6. A hypothesis test is applied to measurements of a track file with respect to the estimated trajectory. When a residual is too large, that particular measurement is rejected and a new measurement is chosen. This procedure indeed works

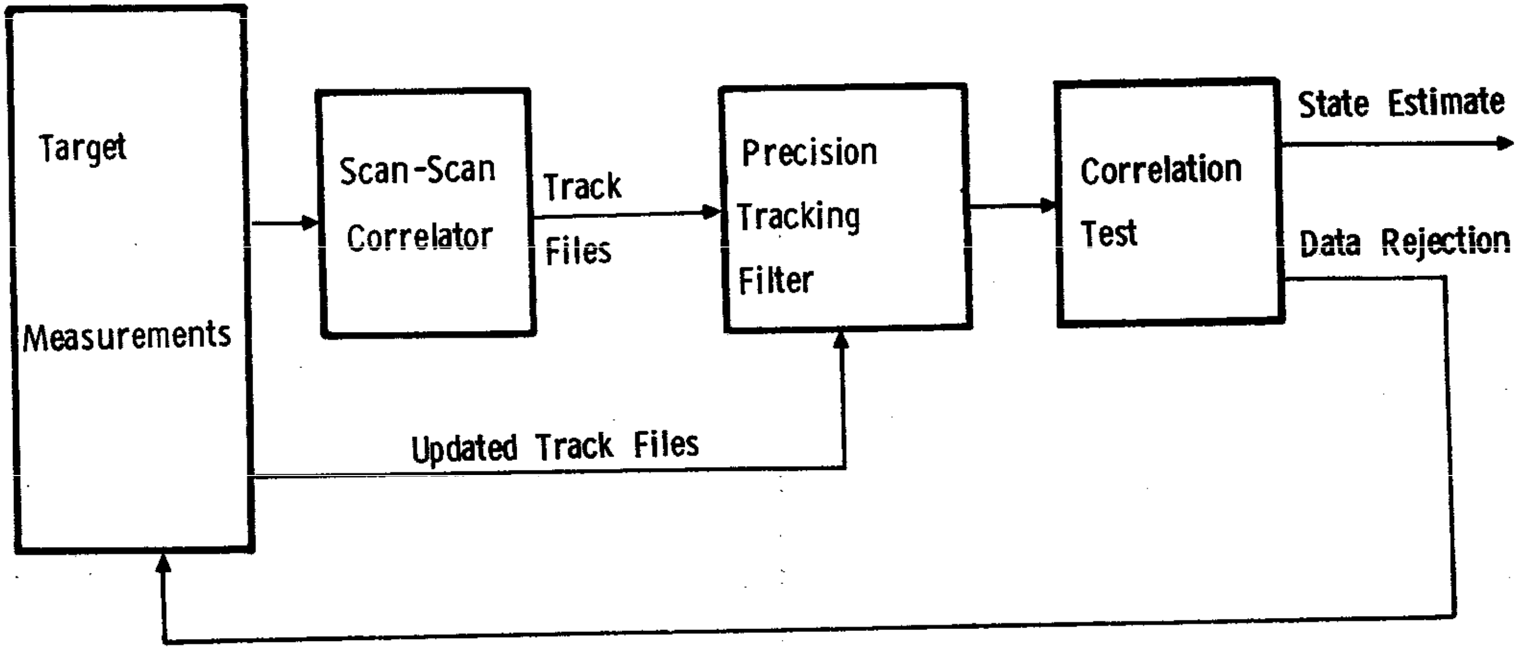


Fig. 3.6. Track file smoothing.

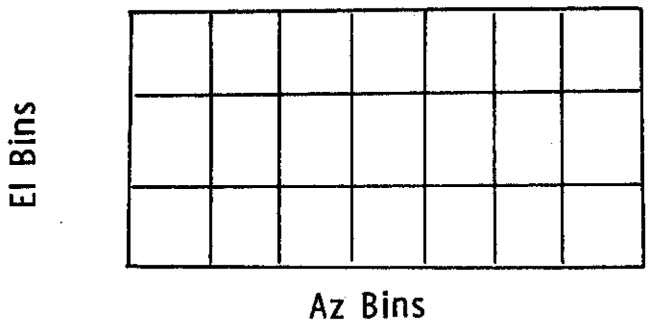
well as will be illustrated in the numerical results section.

3.5 A Partitioned Field of View Implementation

From the discussion of previous sections, it is clear that the solution to the multiple target tracking problem is not a problem requiring sophisticated mathematical manipulations, rather involving simple calculations and large scale data and file management schemes. When it is to be implemented on a digital computer, the problem of indexing measurements in a track file can be a real challenge for programmers.

In this section, we briefly describe an implementation scheme which is found to be computationally efficient. The sensor field of view (FOV) is first partitioned into bins (Figure 3.7). In the exoatmospheric BMD application, these will specifically be bins in azimuth and elevation. The bin size should not be smaller than the maximum target motion between two observation and should be at least a few standard deviations of prediction error. When measurements of a bin are being track initiated, only measurements in the bin itself and the bordering bins for the subsequent measurement frame are searched. In the track continuation mode, only those track files with their last measurement residing in the current bin and its immediate neighbors are used for processing. For the target density

TR-643(3.7)



- Partition the FOV Window into Az and El Bins.

- Each Bin May Be Handled By An Independent Processor Which Initiates Tracks in Its Own Bin And Maintains Track Files Which Has the Last Measurement Residing In Its Bin And Immediate Neighbors.

- A Monitor/Supervisor Processor Handles Track Files Crossing Bin Boundaries.

Fig. 3.7. Partitioned field of view implementation.

considered in the BMD application, this method significantly reduces memory access time. Furthermore, if one stores all measurements and track files on disk, then the core memory will only have to be large enough to hold those of a few bins.

Another advantage of the partitioned field of view approach is that it is especially suitable for implementation with dedicated multiple parallel processors. In this case, each bin may be handled by an independent processor and a monitor/supervisor processor can be assigned to handle tracks crossing bin boundaries.

3.6 Summary

In this section, we have described a multiple target tracking algorithm with specific applications to ballistic missile defense. Features of this algorithm are summarized below.

(1) The track initiation process uses a general/parallel search procedure which can substantially reduce processing time.

(2) The track initiation applied to the exoatmospheric BMD problem requires 5-7 frames of measurements. It is applied in a sliding window fashion to handle the changing scene and crossing traffic problem.

(3) A one frame deferred conflict resolution scheme is presented for resolving the problem of multiple track files competing for several measurements.

(4) If all or part of the past measurements are being saved, a concept using a precision filter for further track file editing is described.

(5) A partitioned field of view implementation scheme, with potential for multiple processor implementation, is discussed.

IV. EXAMPLE

In this section, we illustrate our results using an example. This example represents a typical high target density environment in BMD systems.

The exoatmospheric BMD system concept calls for an infrared sensor deployed on a probe vehicle flying on a ballistic trajectory to observe and track incoming targets. In the example presented in this section, a complex consisting of approximately 600 targets are being observed with 20 frames of measurements with ten seconds of time between frames.

The target true angle data at each frame are first processed by a sensor/signal processor (SSP) simulation to generate simulated angle measurements. This simulation takes into account 1) target intensity variations, 2) background and receiver interference, and 3) effects due to limited sensor resolution. Targets may therefore be detected in one frame but missed in the next. Several closely spaced targets may not be mutually resolved and therefore result in fewer number of measurements than targets. Measurements containing unresolved targets are usually biased with a standard deviation determined by all targets involved. Some resolved targets may also contain excessively high measurement errors due to interference introduced by nearby targets. A detailed description of these effects can be found in [17].

Tracking performance is evaluated using a track file consistency measure. We first illustrate some typical track files using Table 4.1. The top row gives frame number. The left column gives track file (TF) identification number. The entries are target identification numbers. Only 12 frames of data are shown for illustration. Track file (TF) #100 contains target 20 throughout 12 frames shown. This is a well-defined, consistant track file. TF #101 contains target 22. It begins at frame 4 and misses the target at frame 7. TF #102 and 103 should be examined together. Targets 31 and 32 form an unresolved closely spaced target cluster at frame 1, 2, 3, 4, and 7. When these two targets begin to get resolved at frame 5, 6, and beyond frame 7, these two track files have successfully tracked them. TF #104 and 105 show a similar situation except that they have failed to consistently track the CSO splitting (see frame 8 and beyond). Unresolved measurements containing more than two targets can also be found in the threat data examined.

Based upon the above illustration, we now describe a "target oriented" scoring (performance evaluation) scheme. For a given target, we first identify all track files containing it. For example, using Table 4.1 for target number 31, one finds that both track files 102 and 103 contain this target. A track file containing this target most often is

TABLE 4.1

ILLUSTRATION OF TYPICAL TRACK FILES

Frame # TF #	1	2	3	4	5	6	7	8	9	10	11	12
100	20	20	20	20	20	20	20	20	20	20	20	20
101	0	0	0	22	22	22	0	22	22	22	22	22
102	31 32	31 32	31 32	31 32	31	31	31 32	31	31	31	31	31
103	31 32	31 32	31 32	31 32	32	32	31 32	32	32	32	32	32
104	26 27	26 27	26 27	26 27	26	26	26 27	27	27	27	27	27
105	26 27	26 27	26 27	26 27	27	27	26 27	26	26	26	26	26

Entries are target ID's.

assigned to represent this target. In Table 4.1, track files 100, 101, 102, 103, 104, and 105 are assigned to represent targets 20, 22, 31, 32, 27, and 26, respectively. It is clear that a track file can sometimes be assigned to more than one targets because of the closely spaced target effect. A performance for each "target" can then be evaluated using the assigned track files. Suppose that the number of true occurrences of each target in Table 4.1 is the same as the table shows, then we conclude that targets 20, 22, 31, 32, 26, and 27 each meet 100%, 100%, 100%, 100%, 83.33% and 83.33% performance, respectively. Take target 26 as an example. Target 26 has appeared for 12 frames. The track file assigned to represent target 26 is track file number 105 which contains target 26 for 10 frames. This means that the performance on tracking target 26 is 10/12 = 83.33%.

We apply the above scoring scheme to tracks generated for the 600 target case described earlier, the results are shown in Fig. 4.1. The horizontal axis gives performance criterion described in the previous paragraph and the vertical axis gives the percentage of targets satisfing a given criterion. Again using data of Table 4.1 for the purpose of illustration, 4 out of 6 targets meet the 100% performance criterion and all targets meet the 80% criterion. We use results shown in Fig. 4.1 to compare the three tracking algorithms de-

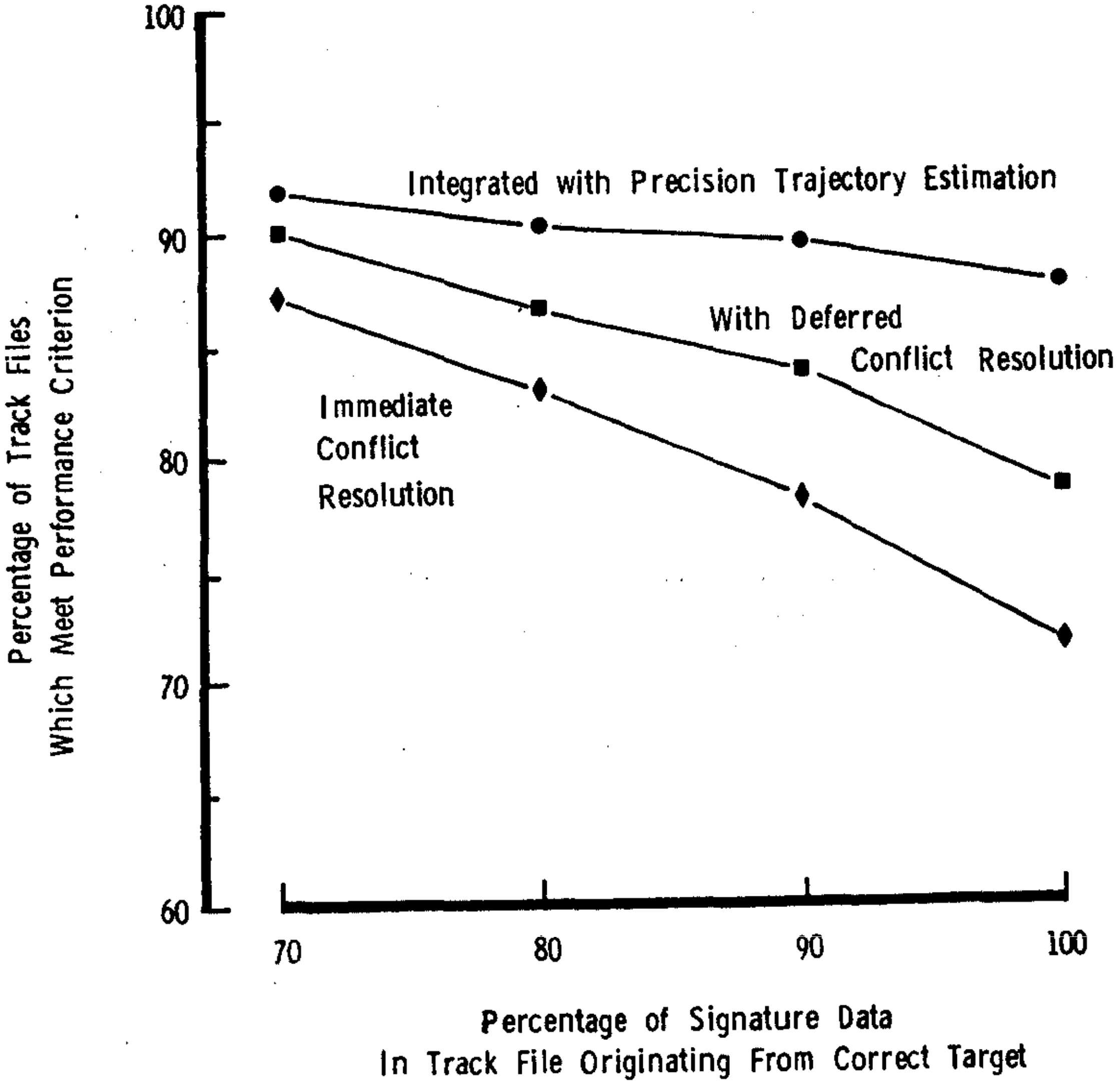


Fig. 4.1. Illustration of tracking performance.

scribed in the previous section. Clearly, the most sophisticated algorithm gives the best result. Also notice that the biggest gain in using a sophisticated algorithm is at the 100% performance criterion level while the differences at the lower criterion region grow smaller. This is because the precision trajectory estimation begins by using track files produced by the correlation algorithm. In order for the precision estimation to succeed, track files presented by the correlation algorithm will have to be reasonably consistent(e.g., > 50%). The gain shown in Fig. 4.1 for the case with track file smoothng is obtained largely by upgrading the track files satisfying 60% - 70% performance criteria.

In Table 4.2, we compare the processing time of these three methods. Notice that they are compared on a relative basis and the total time includes both correlation and precision tracking.

TABLE 4.2

PROCESSING TIME COMPARISON

	CPU Time (normalized)
Immediate Resolution	1
Delayed Resolution	1.2
With Track File Smoothing	1.8

Notes: Processing time includes correlation and precision estimation.

V. SUMMARY

In this report, we have described an algorithm for multiple target tracking and data correlation in a dense target environment. Although some of our discussions were centered around the tracking problem with a passive infrared sensor, the approach represented by this report is applicable in a general situation.

The significant features of our approach include: 1) the use of a general/parallel search for track initiation and 2) the one frame deferred ambiguity resolution method. A partitioned field of view implementation scheme, with potential for multiple processor implementation, was also discussed.

APPENDIX A: Some Useful Results in Fitting Polynomials to a Set of Noisy Measurements

The techniques of polynomial-fit have found wide acceptance in applications. In this Appendix, we present a brief analysis of some polynomial-fit formulas which are relevent to trajectory tracking and prediction problems. The polynomial predictor used in the scan-to-scan correlation algorithm can be derived using these results.

A.1 General Results

Consider a set of polynomials of variable t (denoting time in our applications) represented by $f_o(t)$, $f_1(t),\ldots,f_K(t)$. Assume that this set of polynomials can adequately represent a function y(t) in the form of a weighted linear sum, i.e.,

$$y(t) = \sum_{k=0}^{K} a_k f_k(t). \tag{A.1}$$

Given a set of noise corrupted discrete time measurements of y(t)

$$z(t_n) = y(t_n) + \xi_n;\ n=1,\ldots,N \tag{A.2}$$

where ξ_n is an uncorrelated non-stationary zero mean noise sequence with variance $\sigma_n{}^2$. The objective is to find a set of a_k's giving the "optimal" estimate of y(t) from $z(t_n)$, n=1,...,N. If the weighted-least-square estimator is used, then the optimal a_k's are those minimizing

$$J = \sum_{n=1}^{N} \frac{1}{\sigma_n^{\ 2}} (z(t_n) - y(t_n))^2 \qquad (A.3)$$

Let

$$\underline{a} = \begin{bmatrix} a_o \\ \cdot \\ \cdot \\ a_k \end{bmatrix} \quad \text{and} \quad \underline{f} = \begin{bmatrix} f_o \\ \cdot \\ \cdot \\ f_k \end{bmatrix} \qquad (A.4)$$

Then the estimate of $\underline{a}$, $\underline{\hat{a}}$, is

$$\underline{\hat{a}} = \left[\sum_{n=1}^{N} \frac{1}{\sigma_n^{\ 2}} \underline{f}(t_n)\, \underline{f}^T(t_n)\right]^{-1} \left[\sum_{n=1}^{N} \frac{1}{\sigma_n^2} z(t_n)\, \underline{f}(t_n)\right] (A.5)$$

The covariance of $\underline{\hat{a}}$, P, is

$$P = \left[\sum_{n=1}^{N} \frac{1}{\sigma_n^{\ 2}} \underline{f}(t_n)\, \underline{f}^T(t_n)\right]^{-1} \qquad (A.6)$$

Consider a special set of polynomials

$$\begin{aligned} f_O(t) &= 1 \\ f_1(t) &= t \qquad (A.7) \\ &\ \vdots \\ f_K(t) &= t^K/K! \end{aligned}$$

Then,

$$F(t_n) = \begin{bmatrix} 1 \\ t_n \\ \cdot \\ \cdot \\ \cdot \\ t_n{}^K/K! \end{bmatrix} \begin{bmatrix} 1 & t_n & \cdots\cdots & t_n{}^K/K! \end{bmatrix}$$

$$= \begin{bmatrix} 1 & t_n & \cdots\cdots\cdots & t_n{}^K/K! \\ t_n & & & \\ \cdot & & & \\ \cdot & & & \\ t_n{}^K/K! & & \cdots\cdots\cdots & t_n{}^{2K}/(K!)^2 \end{bmatrix} \tag{A.8}$$

and

$$\underline{\hat{a}} = \left[\sum_{n=1}^{N} \frac{1}{\sigma_n{}^2} F(t_n) \right]^{-1} \left[\sum_{n=1}^{N} \frac{1}{\sigma_n{}^2} z(t_n) \, \underline{f}(t_n) \right] \tag{A.9}$$

$$P = \left[\sum_{n=1}^{N} \frac{1}{\sigma_n{}^2} F(t_n) \right]^{-1} \tag{A.10}$$

A.2 Results for a Second Order Polynomial With Uniform Time Samples and Stationary Noise Sequences

Consider the following conditions

1. $f_K(t) = f_2(t) = \frac{t^2}{2}$

2. Time samples are taken uniformly and t=0 corresponds to the center of the data batch. This implies

$$t_n = (n-1)T - \frac{(N-1)T}{2} \tag{A.11}$$

where T is the intersample spacing.

3. $\sigma_n = \sigma$ for all n's

Also, using the following identities,

1. $$\sum_{n=1}^{N} ((n-1)T - \frac{(N-1)T}{2}) = 0$$

2. $$\sum_{n=1}^{N} ((n-1)T - \frac{(N-1)T}{2})^2 = \frac{T^2}{12} (N-1)N(N+1) \qquad (A.12)$$

3. $$\sum_{n=1}^{N} ((n-1)T - \frac{(N-1)T}{2})^3 = 0$$

4. $$\sum_{n=1}^{N} ((n-1)T - \frac{(N-1)T}{2})^4 = \frac{T^4(N-1)N(N+1)}{240} (3N^2-7)$$

Then Equations (A.9) and (A.8) become

$$P=\sigma^2 \begin{bmatrix} \frac{3}{4} \frac{(3N^2-7)}{(N-2)N(N+2)} & 0 & \frac{-30}{T^2(N-2)N(N+2)} \\ 0 & \frac{12}{T^2(N-1)N(N+1)} & 0 \\ \frac{-30}{T^2(N-1)N(N+1)} & 0 & \frac{720}{T^4(N-2)(N-1)N(N+1)(N+2)} \end{bmatrix}$$

and (A.13)

$$\begin{bmatrix} \hat{a}_0 \\ \\ \hat{a}_1 \\ \\ \hat{a}_2 \end{bmatrix} = \frac{1}{\sigma^2} P \begin{bmatrix} \sum_{n=1}^{N} z(t_n) \\ \\ \sum_{n=1}^{N} T(n-1-\frac{(N-1)}{2}) \quad z(t_n) \\ \\ \sum_{n=1}^{N} T^2(n-1-\frac{(N-1)}{2})^2 \quad z(t_n)/2 \end{bmatrix} \qquad (A.14)$$

If one repeats the above exercise by choosing K=1, one will obtain

$$P = \sigma^2 \begin{bmatrix} \frac{1}{N} & 0 \\ \\ 0 & \frac{12}{T^2(N-1)N(N+1)} \end{bmatrix} \qquad (A.15)$$

$$\begin{bmatrix} \hat{a}_0 \\ \\ \hat{a}_1 \end{bmatrix} = \frac{1}{\sigma^2} P \begin{bmatrix} \sum_{n=1}^{N} z(t_n) \\ \\ \sum_{n=1}^{N} T(n-1-\frac{(N-1)}{2}) \quad z(t_n) \end{bmatrix} \qquad (A.16)$$

A.3 Applications to Position, Velocity and Acceleration Estimation

Let p_o,v_o, and a_o denote the position, velocity, and acceleration, respectively, of a moving object at t=0, then its position at an arbitrary time t is

$$p(t) = p_o + v_o t + \frac{1}{2} a_o t^2 \qquad (A.17)$$

If $p(t_n)$, n=1,..., N denote a set of noisy measurements of $p(t_n)$, the Equations (A.13), (A.14), (A.15), and (A.16) can be directly applied for estimating p_o, v_o, and a_o. Equation (A.13) and (A.14) correspond to a constant acceleration model while Equations (A.15) and (A.16) corresponds to a constant velocity model. With p_o, v_o, and a_o, estimates at time t are obtained by using

$$\begin{bmatrix} \hat{p}(t) \\ \hat{v}(t) \\ \hat{a}(t) \end{bmatrix} = \Phi \ (t) \begin{bmatrix} \hat{p}_o \\ \hat{v}_o \\ \hat{a}_o \end{bmatrix} \qquad (A.18)$$

where

$$\Phi(t) = \begin{bmatrix} 1 & t & t^2/2 \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix} \qquad (A.19)$$

and the covariance becomes

$$P(t) = \Phi(t) \ P \ \Phi^T(t) \qquad (A.20)$$

The results for a constant velocity model can be obtained accordingly.

APPENDIX B: On the Multiple and Single Stage Iterative Least Square Estimators

The batch state estimator based on the Maximum Likelihood or the Weighted Minimum Mean Square error criterion for the angle only tracking application was discussed in detail in [16]. In this appendix, we briefly state this algorithm in section B.1. It is called a multiple stage iterative algorithm because it attempts to minimize residuals with respect to several (all) measurements. If one selects to minimize with respect to the most recent measurement only while holding the estimate obtained with all previous measurements constant, it is called the single stage iterative algorithm. This algorithm is discussed in section B.2. Several forms of iterative filters can be found in [18].

B.1 A Multiple Stage Iterative Least Square Estimator

Consider the following state and measurement equations:

$$\dot{\underline{x}} = f(\underline{x}) \qquad (B.1)$$

$$\underline{z}_k = \underline{h}(\underline{x}_k) + \underline{v}_k; \; k=1,\ldots,N \qquad (B.2)$$

where $\underline{x}$ is the state vector, $\underline{z}_k$ is the measurement vector, and $\underline{v}_k$ is the measurement noise vector with Gaussian distribution with zero mean and known covariance. The current time is denoted by the index k=N. We state the estimator equations without derivation.

Let $\hat{\underline{x}}^{k}_{N/N}$ denote the k-th iteration of the estimate of $\underline{x}_N$, then

$$\hat{\underline{x}}^{k+1}_{N/N} = \hat{\underline{x}}^{k}_{N/N} + P^{k}_{N/N} \left[\sum_{n=1}^{N} G^{k^T}_{n} H^{k^T}_{n} R^{-1}_{n} (\underline{z}_n - \underline{h}(\hat{\underline{x}}^{k}_{n/N})) \right] \qquad \text{(B.3)}$$

$${P^{k}_{N/N}}^{-1} = \sum_{n=1}^{N} G^{k^T}_{N} H^{k^T}_{n} R^{-1}_{n} H^{k}_{n} G^{k}_{n} \qquad \text{(B.4)}$$

where

$G^k_n \quad = \quad {\phi^k_n}^{-1} G^k_{n+1}$; $n=N-1, N-2, \ldots, 1$

$G^k_N \quad = \quad I$ (an identity matrix)

$\phi^k_n \quad = \quad$ linearized transition matrix

$F^k_s \quad = \quad$ the Jacobian matrix of $\underline{f}(\hat{\underline{x}}^{k}_{s/N})$

$H^k_n \quad = \quad$ the Jacobian matrix of $\underline{h}(\hat{\underline{x}}^{k}_{n/N})$

$\hat{\underline{x}}^{k}_{n/N} \quad = \quad$ result of integrating $\dot{\underline{x}}=\underline{f}(\underline{x})$ backward from t_N to t_n using $\underline{x}_N = \hat{\underline{x}}^{k}_{N/N}$.

The iteration is terminated when $\hat{\underline{x}}^{k+1}_{N/N}$ and $\hat{\underline{x}}^{k}_{N/N}$ are sufficiently close. The notation $\hat{\underline{x}}_{i/j}$ denotes the estimate of $\underline{x}_i$ based upon data $\underline{z}_1, \ldots, \underline{z}_j$.

We make the following remarks:

1) The above algorithm is a realization of the maximum likelihood estimator with Gaussian measurement noise process. It is well-known that the maximum likelihood estimate is asymptotically efficient and Gaussian and approaches the Cramer-Rao bound.

2) The $P_{N/N}$ of (B.4) is an approximate expression for the covariance of $\hat{\underline{x}}_{N/N}$. The $P_{N/N}$ evaluated at the true state is the Cramer-Rao lower bound on the covariance of $\hat{\underline{x}}_{N/N}$. Since $\hat{\underline{x}}_{N/N}$ approaches the true state with probability one, $P_{N/N}$ also approaches the Cramer-Rao bound with probability one.

3) Notice that the inverse of $P_{N/N}$ is the Fisher's information matrix. The invertibility of the information matrix is tied to the observabilility of the system, see for example [19].

There are many application areas for this algorithm. One important application area is track initiation. Since the initial covariance and state estimates are not generally given a priori, the above algorithm can obtain the best estimates based on the first N measurement vectors and then proceed to use $\hat{\underline{x}}_{N/N}$ and $P_{N/N}$ as the initial state and covariance estimates, respectively. This method is sometimes referred to as the information matrix approach for filter initiation.

B.2 A Single Stage Iterative Least Square Estimator

The above results were given without derivations. Interested readers can consult [16] for details. Notice that the estimator (eqs. (B.3)) is a weighted combination of all previous measurements. This requires storing of all past measurements. The well-known extended Kalman filter only requires storing the most recent measurements. However it may result in biased estimates. If indeed past measurements can not be stored, one can extend the result of Section B.1 to a single stage iterative estimator. It can also be shown that the extended Kalman filter is only the first iteration of the single stage iterative estimator.

Let $\hat{\underline{x}}_{N/N-1}$ denote the estimate of $\underline{x}_N$ based upon all the measurements up to and including $\underline{z}_{N-1}$. Upon receiving the new measurement $\underline{z}_N$, the optimum estimate at time t_N, $\hat{\underline{x}}_{N/N}$, is the $\underline{x}_N$ minimizing

$$J = (\underline{z}-\underline{h}(\underline{x}_N))^T R_N^{-1}(\underline{z}_N-\underline{h}(\underline{x}_N)) + (\hat{\underline{x}}_{N/N-1} - \underline{x}_N)^T P_{N/N-1}(\hat{\underline{x}}_{N/N-1} - \underline{x}_N) \quad (B.5)$$

where R_N and $P_{N/N-1}$ are covariances of $\underline{z}_N$ and $\hat{\underline{x}}_{N/N-1}$, respectively. Following similar derivations of [16], one obtains the following iterative algorithm.

$$\hat{\underline{x}}^{k+1}_{N/N} = \hat{\underline{x}}^{k}_{N/N} + P^{k}_{N/N}\; H^{k}_{n} R^{-1}_{n} (\underline{z}_N - \underline{h}(\hat{\underline{x}}^{k}_{N/N})) + P^{-1}_{N/N-1}(\hat{\underline{x}}_{N/N-1} - \hat{\underline{x}}^{k}_{N/N}) \tag{B.6}$$

$${P^{k}_{N/N}}^{-1} = P^{-1}_{N/N-1} + H^{k}_{N} R_N^{-1} {H^{k}_{N}}^{T} \tag{B.7}$$

where H^k_N is the Jacobian matrix of $\underline{h}(\cdot)$. Equation (B.7) can be re-written as

$$P^{k}_{N/N} = P_{N/N-1}\left[I - H_N^{T} (H_N^{k} P_{N/N-1} {H_N^{k}}^{T} + R_N^{k})^{-1} H_N^{k} P_{N/N-1}\right] \tag{B.8}$$

using the Matrix Inversion Lemma.

Notice that if we choose the initial guess for $\hat{\underline{x}}^0_{N/N}$ as $\hat{\underline{x}}_{N/N-1}$ and stop after the first iteration, we have obtained the extended Kalman filter equation.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Keh-Ping Dunn for technical discussions throughout this work, to Dr. S. D. Weiner for reviewing the manuscript, and to Chris Tisdale for preparing the manuscript.

REFERENCES

[1] D. B. Reid, "An Algorithm for Tracking Multiple Targets," IEEE Trans. on Automatic Control AC-24, 843 (1979).

[2] Y. Bar-Shalom, "Tracking Methods in a Multitarget Environment," IEEE Trans. on Automatic Control AC-23, 618 (1978).

[3] R. A. Singer, R. G. Sea, and K. B. Housewright, "Derivation and Evaluation of Improved Tracking Filters for Use in Dense Multi-target Environments," IEEE Trans. on Automatic Control IT-20, 423 (1974).

[4] R. A. Singer and R. G. Sea, "New Results in Optimizing Surveillance System Tracking and Data Correlation Performance in Dense Multitarget Environments," IEEE Trans. on Automatic Control AC-18, 571, (1973).

[5] C. L. Morefield, "Application of 0-1 Integer Programming to Multitarget Tracking Problems," IEEE Trans. on Automatic Control AC-22, 302 (1977).

[6] Y. Bar-Shalom and E. Tse, "Tracking in a Cluttered Environment With Probabilistic Data Association," Automatica 11, 451 (1975).

[7] "Compliation of FAS Data Processor Functional and Algorithm Performance Descriptions, Adendum," McDonnell Douglas Astronautics Company Report No. MDC G8641, Huntington Beach, CA (December 1980).

[8] "TBE Scan-to-Scan Algorithm," Special Technical Report No. SD80-BMDATC-2482, Teledyne Brown Engineering, Huntsville, AL (November 1980).

[9] R. P. Wishner, F. X. Lanzinger, H. M. Mesivala, and S. Mori, "Advanced Techniques for Multi-Target Track Formation," Final Report for Contract No. DASG-80-0039, Advanced Information and Decision Systems, Mountain View, CA (April 1981).

[10] "Algorithm Development and Evaluation For the FAS IGT Program, Vol. I, Algroithm Description and Performance," Nichols Research Corporation Report No. NRC-TR-82-011 (30 April 1982).

[11] C. B. Chang and L. Youens, "Measurement Correlation for Multiple Sensor Tracking in a Dense Target Environment," Technical Report 549, Lincoln Laboratory, M.I.T. (20 January 1981), DTIC AD-A098001.

[12] C. B. Chang and J. A. Tabaczynski, "Application of Adaptive Estimation to Target Tracking," Technical Report 598, Lincoln Laboratory, M.I.T. (13 January 1982), DTIC AD-A112036.

[13] C. B. Chang and M. Athans, "State Estimation for Discrete Systems with Switching Parameters," IEEE Trans. on Aero and Elec. Syst. AES-14, 418 (1978).

[14] M. Athans and C. B. Chang, "Adaptive Estimation and Parameter Identification Using Multiple Model Estimation Algorithms," Technical Note 1976-28, Lincoln Laboratory, M.I.T. (23 June 1976), DDC AD-A028510/6.

[15] B. Kreko, Linear Programming (American Elsevier Publishing Company, New York, 1968).

[16] C. B. Chang, "Optimal State Estimation of Ballistic Trajectories with Angle-Only Measurements," Technical Note 1979-1, Lincoln Laboratory, M.I.T. (24 January 1979), DDC AD-A067354/1.

[17] C. B. Chang and K. P. Dunn, "A Functional Model for the Closely Spaced Object Resolution Process," Technical Report TR-611, Lincoln Laboratory, M.I.T. (20 May 1982), DTIC AD-A116890.

[18] A. H. Jazwinski, Stochastic Processes and Filtering Theory (Academic Press, New York, 1970).

[19] T. S. Lee, K. P. Dunn, and C. B. Chang, "On Observability and Unbaised Estimation of Nonlinear Systems," Proceedings of 10th IFIP Conference on Systems Modeling and Optimization (Springer-Verlag, New York, N.Y. 1982).

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
ESD-TR-83-017

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE *(and Subtitle)*
An Algorithm for Multiple Target Tracking
and Data Correlation

5. TYPE OF REPORT & PERIOD COVERED
Technical Report

6. PERFORMING ORG. REPORT NUMBER
Technical Report 643

7. AUTHOR(s)
Chaw-Bing Chang and Leonard C. Youens

8. CONTRACT OR GRANT NUMBER(s)
F19628-80-C-0002

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Lincoln Laboratory, M.I.T.
P.O. Box 73
Lexington, MA 02173-0073

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
Program Element Nos. 63304A
and 63308A

11. CONTROLLING OFFICE NAME AND ADDRESS
Ballistic Missile Defense Program Office
Department of the Army
5001 Eisenhower Avenue
Alexandria, VA 22333

12. REPORT DATE
13 June 1983

13. NUMBER OF PAGES
60

14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*
Electronic Systems Division
Hanscom AFB, MA 01731

15. SECURITY CLASS. *(of this report)*
Unclassified

15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

algorithm
multiple target tracking
data correlation
infrared sensor
measurement correlation

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

An algorithm for multiple target tracking and data correlation is described. A general description of the problem and solution is first given. More specific discussions on tracking with a passive infrared sensor then follow. An example is presented to illustrate the trade-off between algorithm complexity, performance, and processing requirements.