A Track Correlation Algorithm for Multi-Sensor Integration

Michitaka Kosaka,* Shoji Miyamoto,† and Hirokazu Ihara‡
Hitachi Ltd., Asao-ku Kawasaki, Japan

A track correlation algorithm in a multi-sensor integration system for a surveillance mission is proposed. The performance of such track correlation processing is strongly dependent not only on the target state estimation error distribution but also on the target spatial density distribution. Therefore, a track correlation problem is formulated as the likelihood ratio test problem which can take both target state estimation error distribution and target state spatial density distribution into consideration. From this formulation, the correlation algorithm for on-line processing is derived by modifications and approximations. Through analytical evaluations and simulation studies, it is shown that the proposed algorithm is superior to the conventional nearest neighbor algorithm.

I. Introduction

IN the multi-sensor integration for surveillance missions discussed by C.L. Bowman,[1] sensor-level track files are selected to provide the measurements for a multi-sensor integration algorithm in response to computational and timing requirements, as well as system reliability and flexibility. Track correlation processing and track filtering are indispensable for such multi-sensor integration.[1-3] This paper deals with a track correlation algorithm suitable for integrating multi-sensor track files.

There are two important requirements for the track correlation problem in multi-sensor track file integration, where each sensor track file usually has mutli-target-track data. One of the requirements is the low frequency of the correlation error and the other is the high speed of the correlation processing. Because of these requirements, the nearest neighbor algorithm (or the gate algorithm) or the maximum likelihood correlation algorithm[1] based only on the target state estimation error are most popular for multi-sensor track file integration. However, track correlation error depends on target state spatial density distribution, as well as target state estimation error. From this point of view, the track correlation problem should be formulated as the likelihood ratio testing which can take both target state estimation error and target state spatial density distribution into consideration.

Based on this formulation of the track correlation problem, a new track correlation algorithm suitable for multi-sensor track file integration is presented. The algorithm has been derived through some approximations and modifications of the log-likelihood ratio function in the likelihood ratio testing formulation. Next, the optimization technique of threshold parameters in correlation algorithms is discussed in order to obtain the minimum correlation error. Finally, analytical evaluations and simulation studies are carried out to show the feasibility of the proposed approach.

II. Multi-Sensor Integration System

Multi-Sensor Integration System Model

The multi-sensor integration system for target tracking discussed in this paper can be modeled as Fig. 1. In this model, each sensor site has a sensor such as radar or sonar, an estimator such as Kalman filter or α-β tracker, and a track file for storing target track data. The notations in Fig. 1 are as follows:

$X_i(t)$ = the ith target state vector
$Y_{ij}(t)$ = the jth sensor's observation of the ith target
$\hat{X}_{ij}(t)$ = the ith target state estimate (track data) at jth sensor site
$\hat{X}_i(t)$ = the target state estimate in the multi-sensor integration system
t = time

$\hat{X}_i(t)$ is obtained using the track filter[3] or by linear combination of $\hat{X}_{ij}(t)$ as

$$\hat{X}_i(t) = \sum_{j=1}^{M} c_j \hat{X}_{ij}(t) \tag{1}$$

where M is the number of track files in a multi-sensor tracking system. In order to integrate the multi-sensor track files, a track correlation is indispensable as preprocessing for track filtering or track combination.

Track Correlation Problem Statement

Sensor site R_1 is assumed to have n_1 tracks in its track file and sensor site R_2 to have n_2 tracks in its track file. Then, the track correlation problem can be formulated as follows:

Track Correlation Problem

$$\hat{X}_{i1}(t) \in R_1, \quad 1 \le i \le n_1$$

$$\hat{X}_{j2}(t) \in R_2, \quad 1 \le j \le n_2$$

and, $\hat{X}_{i1}(t)$ and $\hat{X}_{j2}(t)$ are assumed to be n-dimensional track vectors. Then, find $\hat{X}_{j*2}(t)$ in R_2 such that it has the same target track as $\hat{X}_{i1}(t)$. Here, $j*$ is satisfied with $0 \le j* \le n_2$, and $j* = 0$ implies that there is no same target track as $\hat{X}_{i1}(t)$ in R_2. $\hat{X}_{i1}(t)$, $\hat{X}_{j2}(t)$ are denoted as

$$\hat{X}_{i1}(t) = \left[x_{i1}^1(t), \ldots, x_{i1}^n(t)\right]$$

$$\hat{X}_{j2}(t) = \left[x_{j2}^1(t), \ldots, x_{j2}^n(t)\right]$$

where $x_{i1}^k(t)$, $x_{j2}^k(t)$ are position or velocity parameters and so on. Furthermore, $Z_{ij}(t)$ is defined as the difference between $\hat{X}_{i1}(t)$ and $\hat{X}_{j2}(t)$, that is,

$$Z_{ij}(t) = \hat{X}_{i1}(t) - \hat{X}_{j2}(t) = \left[z_{ij}^1(t), \ldots, z_{ij}^n(t)\right] \tag{2}$$

Received Sept. 3, 1985; revision received Jan. 2, 1986.

*Researcher, Systems Development Laboratory. Member AIAA.
†Senior Researcher, Systems Development Laboratory.
‡Deputy General Manager, Systems Development Laboratory.

By using $Z_{ij}(t)$, the above problem can be reduced to the following hypothesis testing problem.

Hypothesis Testing in Track Correlation

Let H_0 and H_1 be the following events:

H_0: the event where $\hat{X}_{i1}$, $\hat{X}_{j2}$ are tracks of the same target, that is,

$$Z_{ij}(t) = \boldsymbol{v} \tag{3}$$

H_1: the event where $\hat{X}_{i1}$, $\hat{X}_{j2}$ are tracks of different targets, that is,

$$Z_{ij}(t) = \boldsymbol{m} + \boldsymbol{v} \tag{4}$$

then, the track correlation problem is reduced to detecting the acceptance of H_0 or H_1 from $Z_{ij}(t)$.

In the above formulation, $\boldsymbol{m}$ and $\boldsymbol{v}$ are n-dimensional vectors as

$$\boldsymbol{m} = (m_1, \ldots, m_n)$$
$$\boldsymbol{v} = (v_1, \ldots, v_n)$$

The term $\boldsymbol{v}$ is an estimation error vector due to the target state estimation error of R_1, R_2, and $\boldsymbol{m}$ is a vector of the difference between the ith target and the jth target.

In a multi-target environment, n_1 or n_2 is usually a large number. Therefore, there are two important requirements for the track correlation problem such as the low frequency of correlation error and the high speed of correlation processing. From these reasons, the nearest neighbor algorithm (or the gate algorithm) is the most popular approach to this problem, which is given as follows.

Nearest Neighbor Algorithm

If $(|z_{ij}^1(t)| < e_1) \cap \ldots \cap (|z_{ij}^n(t)| < e_n)$, then accept H_0; otherwise accept H_1.

III. New Track Correlation Algorithm

Correlation Problem Formulation Using Likelihood Ratio Testing

Generally speaking, in a multi-target environment, it is difficult to distinguish or identify tracks of each sensor site when the target spatial density distribution is dense. On the other hand, it is easy to distinguish each track when the target spatial density distribution is sparse. Therefore, the track correlation problem is strongly dependent on the target spatial density distribution, as well as the target state estimation error distribution.

Consider the multi-target tracking environment in an air surveillance system. Targets are aircraft, so that aircraft velocities and altitudes are more similar than their horizontal positions. Therefore, the spatial density distribution of velocity or altitude is more dense around $m_i = 0$ compared to that of a horizontal position parameter. Considering such a difference in each parameter's spatial density distribution, it is possible to formulate the track correlation problem for a multi-sensor integration system as the well-known likelihood ratio testing problem.

First, we introduce the following two conditional probabilities: 1) $P(Z|H_0)$: the conditional probability of Z under H_0, which depends on the target state estimation error; 2) $P(Z|H_1)$: the conditional probability of Z under H_1, which depends on the target state estimation error and the target spatial density distribution, that is,

$$P(Z|H_0) = \frac{1}{\sqrt{2\pi^n |R|}} \exp[-1/2 \;\; Z^t R^{-1} Z] \tag{5}$$

$$P(Z|H_1) = \frac{1}{\sqrt{2\pi^n |R|}} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \exp\big[-1/2 \;\; (Z - \boldsymbol{m})^t R^{-1}(Z - \boldsymbol{m}) \cdot f(\boldsymbol{m})\,\mathrm{d}\boldsymbol{m}\big] \tag{6}$$

where R is the covariance matrix of a vector Z.

Next, the likelihood ratio $L(Z)$ is defined by

$$L(Z) = P(Z|H_1)/P(Z|H_0) \tag{7}$$

Then, using the likelihood ratio testing theory, the track correlation problem is reduced to detect the acceptance of H_1 or H_0 from Z_{ij} according to the following relationship:

$$\text{Accept } H_1 \text{ if } L(Z_{ij}) > C_{e'}$$
$$\text{Accept } H_0 \text{ if } L(Z_{ij}) < C_{e'}$$

where $C_{e'}$ is the threshold of likelihood ratio testing.

Simplified Correlation Algorithm

In the correlation processing using likelihood ratio testing, the acceptance of H_0 or H_1 can be determined by using the above hypothesis testing. However, integral calculations are needed for carrying out the hypothesis testing. Therefore, the processing time is not short enough to implement it on the on-line processing system. This problem can be solved by modifications and approximations based on the following assumptions. First, each component's estimation error is assumed to be independent in order to derive an effective correlation algorithm. It is also assumed that z^i is a zero-mean Gaussian white noise with a covariance of σ_i and the prob-

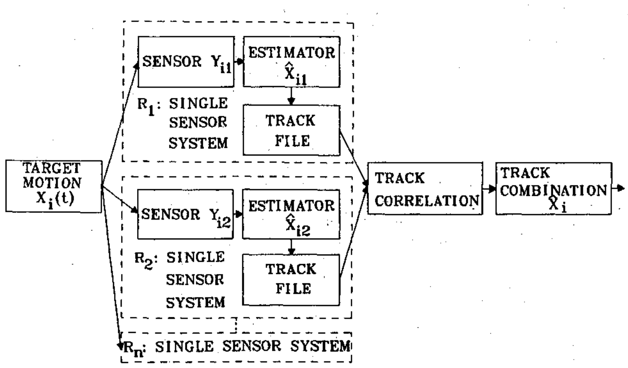

Fig. 1 Functional model in a multi-sensor tracking system.

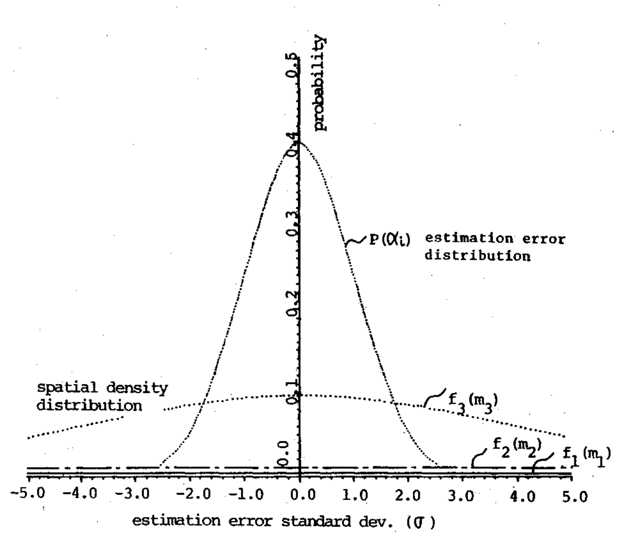

Fig. 2 Estimation error distribution and spatial density distribution.

ability density function $p_i(z^i)$ of z^i is given by

$$p_i(z^i) = \frac{1}{\sqrt{2\pi}\,\sigma_1} e^{-(z^i)^2/2\sigma_i^2} \tag{8}$$

Next, the state spatial density distribution function $f(\boldsymbol{m})$ of $\boldsymbol{m}$ is assumed to be resolved into each component spatial density distribution function $f_i(m_i)$ of m_i, that is,

$$f(\boldsymbol{m}) = f_1(m_1) \cdot f_2(m_2) \cdot \cdots \cdot f_n(m_n) \tag{9}$$

Then, the target spatial density distribution functions can be assumed to be specified as in Fig. 2. Each component spatial density distribution function $f_i(\boldsymbol{m}_i)$ is assumed to be normalized by the state estimation error standard deviation σ_i. In Fig. 2, for example, $i = 1$ represents a position distance parameter, $i = 2$ represents an altitude distance parameter, and $i = 3$ represents a velocity difference parameter.

In the following, the subscript ij of Z_{ij}, z_{ij}^k is omitted for simplicity. From the above assumption, $P(Z|H_0)$ and $P(Z|H_1)$ can be computed by the product of each parameter's conditional probability $P_i(z^i|H_0), P_i(z^i|H_1)$, that is,

$$P(Z|H_0) = \prod_{i=1}^{n} P_i(z^i|H_0) \tag{10}$$

$$P(Z|H_1) = \prod_{i=1}^{n} P_i(z^i|H_1) \tag{11}$$

where $P_i(z^i|H_0), P_i(z^i|H_1)$ are the conditional probabilities of z^i under H_0 or H_1. From the relationship of the Eqs. (10) and (11), the likelihood ratio function $L(Z)$ can be given by the product of each parameter's likelihood ratio function $L_i(z^i)$:

$$L(Z) = \prod_{i=1}^{n} L_i(z^i) \tag{12}$$

$$L_i(z^i) = P_i(z^i|H_1)/P_i(z^i|H_0) \tag{13}$$

By introducing log-likelihood ratio functions $T(Z), T_i(z^i)$ such as

$$T(Z) = \ln L(Z) \tag{14}$$

$$T_i(z^i) = \ln L_i(z^i) \tag{15}$$

the hypothesis testing in track correlation can be carried out by using the following relationship:

$$\text{If } T(Z) = \sum_{i=1}^{n} T_i(z^i) > C_e, \text{ then accept } H_1$$

$$\text{If } T(Z) = \sum_{i=1}^{n} T_i(z^i) < C_e, \text{ then accept } H_0$$

It is notable that the component spatial density distribution functions $f_1(m_1), \ldots, f_n(m_n)$ are different from each other as in Fig. 2. Therefore, the log-likelihood ratio functions $T_1(z^1), T_2(z^2), \ldots$ can be shown as in Fig. 3. From Fig. 3, it is easy to see that the log-likelihood ratio function of each parameter has a different value at $z = 0$, and a position parameter is superior to an altitude parameter or a velocity parameter for deciding $Z \in H_0$ or $Z \in H_1$, if component spatial density distribution functions are assumed to be as in Fig. 2. Moreover, $L_i(z^i)$ is the monotone increasing function, and the following relationship is obvious near $z^i = 0$:

$$f_1(m_1) < f_2(m_2) < \cdots < f_n(m_n) \tag{16}$$

that is,

$$L_1(z^1) < L_2(z^2) < \cdots < L_n(z^n) \tag{17}$$

Then, each parameter's $T_i(z^i)$ can be given as in Fig. 3, and $T_i(z^i)$ has the minimum value at $z^i = 0$. By defining the minimum value of $T_i(z^i)$ as $T_i^{\min}$, we can obtain the following relationship:

$$T_1^{\min} < T_2^{\min} < \cdots < T_n^{\min} \tag{18}$$

and

$$T(Z) = \sum_{i=1}^{n} T_i(z^i) > T_1(z^1) + \sum_{i=2}^{n} T_i^{\min} \tag{19}$$

Introducing $C_e(i)$ such that

$$C_e(i) = C_e - \sum_{j=i+1}^{n} T_j^{\min} \tag{20}$$

if $T_1(z^1) > C_e(1)$, then $Z \in H_1$. This fact shows that if $T_1(z^1) > C_e(1)$, only one detection is needed for deciding to accept H_1 or H_0 instead of n detections. This feature is very important for the detection problem such as the track correlation problem in multi-sensor track file integration, where most of all detections are accepted by the event of H_1. In the case of $T_1(z^1) < C_e(1)$, the following test is performed.

$$\text{If } \sum_{i=1}^{k} T_i(z^i) > C_e(k), \text{ then } Z \in H_1 \tag{21}$$

If $Z \notin H_1$, then we increase k, that is, $k = k + 1$. If $T(Z) < C_e$, this track is a candidate of the same target. And after checking up all tracks in the track file, the track which minimizes the log-likelihood ratio $T(Z)$ is chosen as the correlated track. Next, in order to avoid too much calculation for the log-likelihood ratio function, $T_i(z^i)$ is approximated as in Fig. 4, that is,

$$T_i(z^i) = c_j^i \quad \text{if } d_j^i < |z^i| < d_{j+1}^i \tag{22}$$

where $j = 1, \ldots, M_a$, and c_j^i is a constant. From these approximations, the value of $T_i(z^i)$ can be obtained easily by a table-looking-up method. Due to these approximations, the calculated $T(Z)$ is not the true log-likelihood ratio value, but this approximation error due to these approximations of log-likelihood ratio functions is small enough to accept this proposed algorithm in the multi-sensor integration system such as a surveillance system. The feature of this proposed algorithm

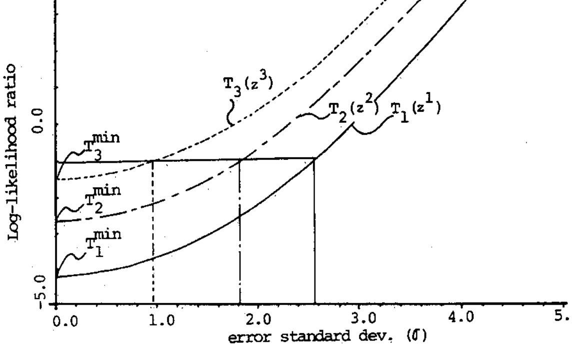

Fig. 3 Log-likelihood ratio curve.

is summarized as follows: 1) the detection criterion is the approximated log-likelihood ratio; 2) the event of H_1 can be detected by few steps.

IV. Threshold Optimization Technique

Optimization Criteria

In correlation processing for multi-sensor track file integration, the threshold optimization is very important in order to minimize the correlation error. There are two errors in the hypothesis testing described in Sec. II, that is,

1) Type 1 error: does not correlate to the same target-track, E_1;

2) Type 2 error: correlates different target-tracks, E_2.

Furthermore, in correlation processing, there are three correlation errors as follows:

Case 1: When the same target-track of $\hat{X}_{i1}(t)$ exists in R_2

Error 1: does not correlate, E_A

Error 2: correlates to different target-tracks, E_B

Case 2: When the same target-track of $\hat{X}_{i1}(t)$ does not exist in R_2

Error 3: correlates to different target-tracks, E_C where E_1, E_2, E_A, E_B, E_C are the probabilities of each event and

$$E_A = E_1 \tag{23}$$

$$E_C \doteq 1 - (1 - E_2)^{n_2} \tag{24}$$

If it is assumed that the probability of Case 1 is P_a and that of Case 2 is P_b, then the probability of the correlation error can be calculated by

$$C_E = P_a \cdot (E_A + E_B) + P_b \cdot E_C \tag{25}$$

Threshold parameters in correlation algorithms can be optimized so that C_E has the minimum value.

Optimization of the Nearest Neighbor Algorithm

According to the nearest neighbor algorithm which is described in Sec. II, it is easy to see that

$$E_1 = 1 - \prod_{i=1}^{n} E_{1i} \tag{26}$$

$$E_2 = \prod_{i=1}^{n} E_{2i} \tag{27}$$

where

$$E_{1i} = \int_{-e_i}^{e_i} p_i(z^i|H_0)\,dz^i \tag{28}$$

$$E_{2i} = \int_{-e_i}^{e_i} p_i(z^i|H_1)\,dz^i \tag{29}$$

e_i is a threshold parameter in the nearest neighbor algorithm. If more than two tracks are detected to be H_0, choose that track which minimizes the difference of position parameter z^p_{ij*}. Then, E_B can be calculated by

$$E_B = (1 - E_1)\left[1 - (1 - P_E)^{n_2 - 1}\right] \tag{30}$$

P_E is the probability of $z^p_{ij} < z^p_{ij*}$ and can be given by

$$P_E = E_2 \cdot \frac{1}{E_{2p}} \cdot \int_{-e_p}^{e_p} p_p(a|H_0) \int_{-a}^{a} p_p(b|H_1)\,db\,da \tag{31}$$

where $j*$ is the index of the same target-track in R_2. From above equations, C_E is a function of threshold parameters $e_i, i = 1, \ldots, n$. Therefore, the threshold parameter optimization problem can be reduced to minimizing $C_E(e_1, \ldots, e_n)$. Therefore, a nonlinear optimization method is applied to this problem. If $p_i(z^i|H_0), p_i(z^i|H_1)$ are differentiable, then the well-known modified Newton-Raphson method can be applied. Defining the vector e as $e = (e_1, \ldots, e_n)$, the optimization of thresholds can be carried out by using the following equations:

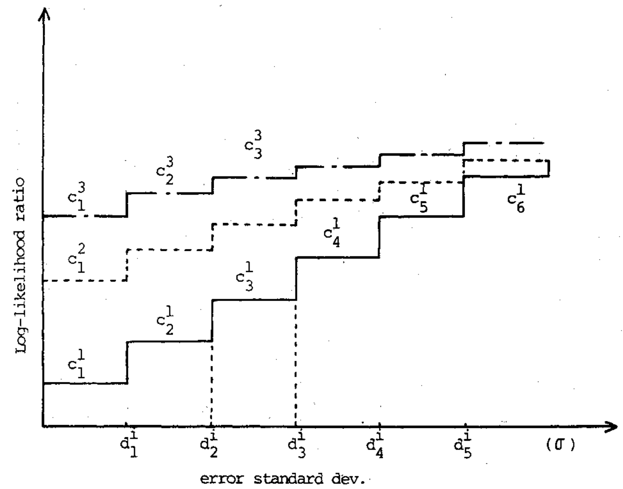

Fig. 4 Approximation of log-likelihood ratio curve.

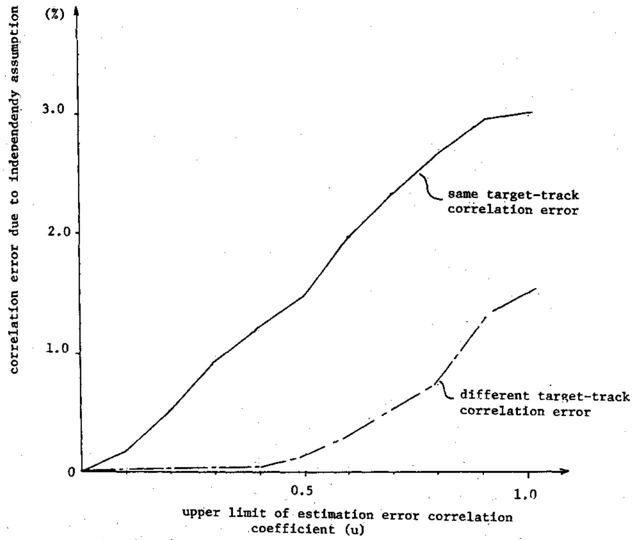

Fig. 5 Influence of independency assumption to correlation error (an example).

$$e^{(k+1)} = e^{(k)} - wG^{-1}g \tag{32}$$

$$G = \left.\frac{\partial^2 C_E}{\partial e_i \partial e_j}\right|_{e=e(k)} \tag{33}$$

$$g = \left.\frac{\partial C_E}{\partial e_i}\right|_{e=e(k)} \tag{34}$$

where k is the iteration index and w is satisfied with $C_E(e^{(k)}) > C_E[e^{(k+1)}]$.

Optimization of the Proposed Algorithm

In order to evaluate C_E of the proposed algorithm, two probabilities are defined as

$$p_{ij} = P\left(T_i(z^i) = c^i_j|\text{same target}\right)$$

$$q_{ij} = P\left(T_i(z^i) = c^i_j|\text{different targets}\right)$$

then

$$P\left(T_1 = C^1_{a_1}, \ldots, T_n = C^n_{a_n} \mid \text{same target}\right) = \prod_{i=1}^{n} p_{ia_i}$$

$$P\left(T_1 = C^1_{b_1}, \ldots, T_n = C^n_{b_n} \mid \text{different targets}\right) = \prod_{i=1}^{n} q_{ib_i}$$

Using the above probabilities, E_1 can be given by probabilities such as $C^1_{a_1} + \cdots + C^n_{a_n} > C_e$, that is,

$$E_1 = U\left(\prod_i p_{ia_i}\right) \left(\sum_{i=1}^{n} C^i_{a_i} > C_e\right) \tag{35}$$

Similarly, E_2 can be given by

$$E_2 = U\left(\prod_i q_{ib_i}\right) \left(\sum_{i=1}^{n} C^i_{b_i} < C_e\right) \tag{36}$$

Furthermore, E_B can be calculated by the following equation:

$$E_B = \sum_{C_j < C_e} P\left[T(Z) = C_j \mid Z \in H_0\right] \times \left(1 - \left\{1 - P\left[T(Z) < C_j \mid Z \in H_1\right]\right\}^{n_2 - 1}\right) \tag{37}$$

where j is the index of a correlated different target. From the above probabilities, the optimal threshold C_e can be obtained by minimizing C_E.

Numerical Evaluation

Numerical evaluation is carried out based on the following conditions:

$$P_i\left(z^i \mid H_0\right) = \frac{1}{\sqrt{2\pi}} e^{-z^{i2}/2} \tag{38}$$

$$P_i\left(z^i \mid H_1\right) = \frac{1}{\sqrt{2\pi}\,\sigma_i} e^{-z^{i2}/2\sigma_i^2} \tag{39}$$

$$N = 100,\ n = 4,\ P_a, P_b = 1/2$$

Evaluation case 1: $\sigma_1 = 30.0$, $\sigma_2 = 15.0$, $\sigma_3 = 8.0$, $\sigma_4 = 5.0$
Evaluation case 2: $\sigma_1 = \sigma_2 = \sigma_3 = \sigma_4 = 15.0$

Under the above conditions, E_A, E_B, E_C, and C_B are calculated by the threshold optimization techniques described here, and the results are shown in Table 1. These results show that the proposed algorithm is superior to the nearest neighbor algorithm because the correlation error probability of the proposed algorithm is less than that of the nearest neighbor algorithm.

V. Consideration of the Assumption of Independency

A simplified algorithm can be derived based on the assumption of independency of each component's estimation error. However, according to Bar-Shalom,[7] each component's estimation error is not independent. This means that the matrix R^{-1} in the Eq. (6) is not a diagonal matrix, so that the proposed algorithm based on the assumption of independency of each component's estimation error seems to be unrealistic. However, the proposed algorithm is still effective to the multi-sensor track file integration because of the following reasons.

Table 1 An example of analytical correlation error evaluation

	Algorithm	E_A	E_B	E_C	C_E
Case 1	N.N[a]	0.0564	0.0217	0.0762	0.0771
	L.R.T[b]	0.0503	0.0123	0.0683	0.0654
Case 2	N.N	0.0257	0.0117	0.0430	0.0402
	L.R.T	0.0246	0.0039	0.0346	0.0315

[a]N.N = nearest neighbor. [b]L.R.T = likelihood ratio test.

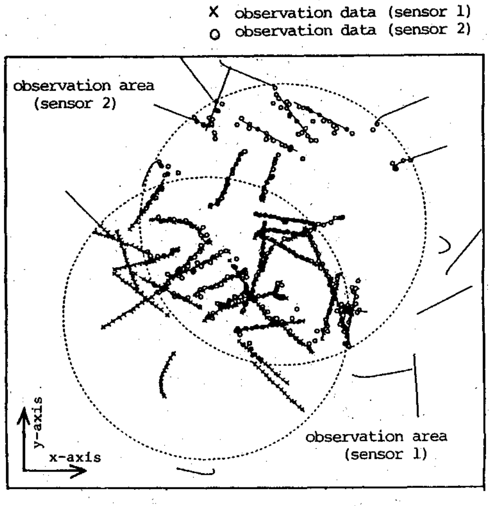

Fig. 6 Multi-target motion and sensor's observation in a simulation study.

R^{-1} in Eq. (6) is a real symmetric matrix so that R^{-1} can be diagonalized by an orthogonal matrix T, that is, $R' = T^t R^{-1} T$. Therefore, $Z' = TZ$, whose components are independent of each other, can be obtained. Z' can be used instead of Z in the proposed algorithm. In the case that R is a constant matrix, the use of Z' does not make the correlation processing time too large. On the other hand, in the case that the matrix R is a time-dependent matrix, the use of Z' makes the processing time too large because T is a time-dependent matrix and needs to be calculated in each hypothesis testing. Therefore, in this case, this method is unrealistic when applied to multi-sensor integration, which needs the high speed of the correlation processing.

If the component errors are not independent, the derived simplified algorithm is an approximation of the likelihood ratio testing. In order to show the feasibility of the proposed algorithm even in the case that the assumption is not valid, a simple simulation study was carried out. In this simulation, the following conditions are assumed:

1) A target state is three-dimensional and the covariance matrix R is given by

$$R = \begin{bmatrix} 1 & u_1 & u_2 \\ u_1 & 1 & u_3 \\ u_2 & u_3 & 1 \end{bmatrix} \tag{40}$$

The correlation coefficients u_i of each component's estimation error are assumed to be distributed uniformly on $(0, u)$, $u < 1.0$.

2) Each component's spatial density distribution is assumed to be a uniform distribution on $(0, 7.0\sigma_i)$, $\sigma_i = 1.0$.

The likelihood ratio testing results under the condition of $u_i = 0$, which corresponds to the proposed algorithm, were compared with the true likelihood ratio testing results. A simulation result is summarized in Fig. 5. The influence of independency assumption to correlation error is dependent on the upper limit u of a uniform distribution. Under the assumed condition, the correlation error due to the assumption

Table 2 A comparison of correlation processing performance in a simulation study

Algorithm	Ratio of S	Ratio of throughput in-decision processing
Nearest neighbor ($n=2$)	1.00	1.00
Nearest neighbor ($n=5$)	0.73	1.07
Likelihood ratio testing	0.38	0.98

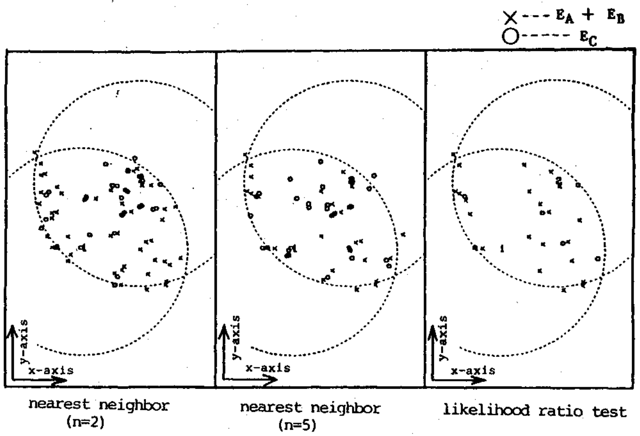

Fig. 7 Spots where correlation error occurred.

of independency is less than 3.0% even in the case of $u = 1.0$. The correlation error due to the independency assumption depends on various factors such as the distribution of correlation coefficient u_i, components' spatial density distributions and so on. However, from the above simulation results, the proposed algorithm seems to be effective even if the independency assumption is not valid.

VI. Simulation Study

Simulation Model

The target motion model in each axis of a three-dimensional Cartesian coordinate system can be approximated by the following equations:

$$x(t+dt) = x(t) + \dot{x}(t)\,dt + \ddot{x}(t)\,dt^2/2 \qquad (41)$$

$$\dot{x}(t+dt) = \ddot{x}(t)\,dt + \dot{x}(t) \qquad (42)$$

A simulation study of correlation processing in a multi-sensor tracking system which can be modelled as in Fig. 1 was carried out. It was assumed that 100 targets are generated in the area of Fig. 6, and each sensor site observes and estimates targets' states during a ten-second period. The observation error of each sensor site is given by additive Gaussian white noise with a zero mean. The well-known α-β tracker is used in the sensor site 1 (R_1) and the Kalman filter is used in the sensor site 2 (R_2) in Fig. 1. Correlation processing for multi-sensor track file integration is performed for a 100-second period.

An Example of the Simulation Study

The performance of the following three correlation algorithms is compared by simulation study:

1) Nearest neighbor algorithm ($n = 2$, position parameter);

2) Nearest neighbor algorithm ($n = 5$, position, height, velocity parameter);

3) Likelihood ratio test algorithm.

Fig. 6 shows an example of multi-target motion and observation, and Fig. 7 shows the spots where correlation errors occurred. The symbol "x" shows a type 1 error and "o" shows a type 2 error. In addition, the correlation error rate and the throughput of the hypothesis test per one correlation are compared in Table 2. The correlation error rate S is defined by $S = (N_1 + N_2)/N$, where N_1 is the number of type 1 error occurrences, N_2 is the number of type 2 error occurrences, and N is the total number of correlation processings. The ratio of S in Table 2 means the relative ratio, for example, the ratio between S of likelihood ratio testing and S of nearest neighbor ($n = 2$) in the case of likelihood ratio test. The throughputs are also compared by the ratio of each algorithm's throughput in the hypothesis test processing. These results show that the proposed algorithm is almost the same as the nearest neighbor algorithm in the processing speed but can reduce the frequency of correlation error occurrences. Therefore, the proposed algorithm is better for multi-sensor track file integration than the well-known nearest neighbor algorithm.

VII. Conclusion

A track correlation algorithm for multi-sensor track file integration has been developed by applying the log-likelihood ratio testing theory and simplifying it through modifications and approximations. Furthermore, the threshold optimization technique has been discussed in order to minimize the probability of correlation error.

It is considered that the concept of this correlation algorithm can be applied to various detection problems which need high-speed and highly accurate processing, for example, the star identification problem in an attitude determination system using a star sensor.

Acknowledgments

The authors would like to thank Mr. T. Usui and Mr. M. Katoh of Ohmica Works, Hitachi Ltd., for their invaluable technical effort. Special acknowledgment is due to Dr. J. Kawasaki, general manager of the Systems Development Laboratory, Hitachi Ltd., for his encouragement of this research.

References

[1] Bowman, C.L., "Multi-Sensor Integration for Defensive Control Surveillance," *Proceedings of NAECON'79*, 1979, pp. 176-184.

[2] Farina, A. and Pardini, S., "Survey of Radar Data Processing Technique in Air Traffic Control and Surveillance System," *IEE Proceedings*, Vol. 27, no. 3, 1980, pp. 190-204.

[3] Thomas, H.W., et al, "Tracking in a Multi-Radar Environment," *IEE Proceedings*, Vol. 123, no. 3, 1976, pp. 191-194.

[4] Bar-Shalom, Y., "Tracking Methods in a Multi-Target Environment," *IEEE Transactions on Automatic Control*, Vol. AC-23, 1978, pp. 618-626.

[5] Sage, A.P. and Melsa, J.L., *Estimation Theory with Application to Communication and Control*, New York, McGraw-Hill, 1970.

[6] Fortmann, T.E. et al., "Detection Thresholds for Multi-Target Tracking in Clutter," *Proceedings of IEEE Conference on Decision and Control*, 1981, pp. 1421-1424.

[7] Bar-Shalom, Y., "On the Track-to-Track Correlation Problem," *IEEE Transactions on Automatic Control*, Vol. AC-26, no. 2, 1981, pp. 571-572.