*Track-to-Track Association and Bias Removal

Lawrence D. Stone*, Mark L. Williams*, Thy M. Tran*

stone@metsci.com, williams@metci.com, ttran@metsci.com

Metron, Inc.
11911 Freedom Drive, Suite 800
Reston, VA 20190
703-787-8700

31 October 2001

ABSTRACT

This paper develops methods for associating two sets of sensor tracks in the presence of missing tracks and translation bias. Key results include (1) Extension of the Maximum *A Posteriori* Probability method of matching tracks to use feature information as well as kinematic information; (2) translation bias removal techniques that are computationally tractable for large numbers of tracks, and effective in the presence of missing tracks. These methods were evaluated by Monte Carlo simulation. The experimental results indicate that the Maximum *A Posteriori* Probability method with its adaptive threshold achieves close to its best performance for matching tracks without an additional threshold adjustment.

1. INTRODUCTION

This paper extends the method for finding the Maximum *A Posteriori* Probability (MAP) assignment described in [1] and [2] to include feature information in addition to position and velocity information. In the track-to-track association problem that we consider, we assume that there are two independent sensor systems. Each system has produced a set of tracks for the objects that they have detected. The number of tracks produced by each sensor need not be the same. Our goal is to identify which tracks from sensor 1 are associated with which tracks from sensor 2 and which tracks are not associated with any other track. The method for doing this is to find the extended MAP assignment and use this assignment to associate the sensor tracks. This extended MAP assignment assumes that any potential bias between the coordinates reported by the two sensors have been removed. Common methods for removing bias are discussed in [1], [2], and [3]; in this paper we present a correlation-based method for removing translation bias in which the correlation is efficiently calculated using Fourier transform techniques.

Section 2 derives the method for computing the posterior probability of an assignment being correct. In this section, we also show that when the target state is position and velocity, our formulation of the MAP agrees with the formulation in [1] and [2]. Section 3 discusses how one can find the MAP assignment by formulating it as a linear programming problem. In section 4 we present the results of simulation runs that test how well the extended MAP approach performs when measured by the number of correct matches produced. The results in this section show that using feature information can substantially improve the performance of the MAP assignment. More interestingly, the results in this section indicate that the MAP method provides close to its best matching performance without adding an additional threshold, contrary to the results in [1].

* stone@metsci.com; phone 1 703 787-8700; fax 1 703 787-3518; http://www.metsci.com; Metron, Inc, 11911 Freedom Drive, Reston, VA, USA 20190-5602

Section 5 presents our correlation-based method for estimating and removing the relative translation bias between two sets of tracks. The technique is applied to simulation data provided by XonTech and Photon Research Associates, respectively.

2. MAXIMUM A POSTERIOR (MAP) ASSIGNMENT

The discussion in this section assumes that bias between the two sets of tracks has been removed. The track assignment method described below is an extension of the Maximum *A Posteriori* Probability Association method given in [1]. The extension consists primarily of accounting for the sensor dependent features associated with each track. To do this we extend the target state to include target type and matching feature variables such as length, spin rate, precession angle, or precession rate. We envision that the two sensors will produce measurements, such as range and bearing measurements, which will be used to estimate the position and velocity of the targets. In addition, the sensors will provide other feature measurements. Features that can be measured by both sensors are called matching features. The remaining sensor dependent feature measurements can be used to determine target type.

2.1 Prior Distributions.

The target state is represented by a 3-tuple $(\mathbb{X},\mathbb{Y},\mathbb{K})$ where

$$\begin{aligned} \mathrm{X} &= \text{ target's postion and velocity} \\ \mathrm{Y} &= \text{ vector of matching feature variables} \\ \mathrm{K} &= \text{ target type} \end{aligned}$$

Let

$$\begin{aligned} \mathbf{E} &= \text{ position-velocity space} \\ \mathbf{Y} &= \text{ space of the matching feature variables} \\ \mathbf{K} &= \{1,\ldots,K\} \text{ be the space of the } K \text{ target types.} \end{aligned}$$

We assume there are a random number *N* of true objects to be tracked with mean number = $\bar{\nu} > 0$. . (We do not consider false targets.) Given that *N* is known, the objects are spatially distributed as *N* independent draws from a probability distribution with density $p^0(x)$ on **E**. The density function for the expected number of true objects is

$$\lambda(x) \equiv \bar{\nu}\, p^0(x) \text{ for } x \in \mathbf{E}.$$

Our assumptions follow those given in [1] except that we do not require the number of objects to be Poisson distributed.

We assume that the prior distribution on $(\mathbb{Y},\mathbb{K})$ is independent of $\mathbb{X}$, the target's position and velocity. Let β^0 be the prior "density" function for $(\mathbb{Y},\mathbb{K})$. The prior on $(\mathbb{X},\mathbb{Y},\mathbb{K})$ is given by

$$\alpha^0(x,y,k) = p^0(x)\beta^0(y,k) \text{ for } (x,y,k) \in \mathbf{E}\times\mathbf{Y}\times\mathbf{K}$$

2.2 Sensor Tracks.

There are two sets of tracks, *I* sensor 1 tracks and *J* sensor 2 tracks. The tracks have been produced independently by the two sensors. Each track consists of a set of measurements and a resulting posterior distribution on $(\mathbb{X},\mathbb{Y},\mathbb{K})$ at the present time *t*.

The feature values may have influenced the choice of which measurements are associated with a track, but they do not directly affect the position-velocity state estimate. In this paper we will not consider problems that may be caused by mis-association of measurements with tracks.

Each track is represented by a probability distribution on the target's state at time *t*. Let

$\{\alpha_i^1; i=1,\ldots,I\}$ be the posterior distributions of the tracks produced by sensor 1 at time t

$\{\alpha_j^2; j=1,\ldots,J\}$ be the posterior distributions of the tracks produced by sensor 2 at time t.

For each target we assume that the posterior distribution on $\mathbb{X}$ is independent of the posterior on $(\mathbb{Y}, \mathbb{K})$ so that the distribution on target state factors as follows:

$$\begin{aligned} \alpha_i^1(x,y,k) &= p_i^1(x)\beta_i^1(y,k) \\ \alpha_j^2(x,y,k) &= p_j^2(x)\beta_j^2(y,k). \end{aligned} \tag{1}$$

In general the number of tracks produced by each sensor are not equal, i.e., $I \neq J$. It will often be true that sensor 1 is tracking some objects not detected (tracked) by sensor 2, and sensor 2 is tracking some objects not detected by the sensor 1.

2.3 Assignment Probability

Following [1] we let $A(I,J)$ be the set of possible assignments between sensor 1 tracks and sensor 2 tracks. Specifically, let

$$A(I,J) = \text{set of all one-to-one mappings from subsets of } \{1,\ldots,I\} \text{ to subsets of } \{1,\ldots,J\}.$$

If $a \in A(I,J)$, then its domain, $\mathbf{Dom}(a) \subseteq \{1,\ldots,I\}$, its range, $\mathbf{Rng}(a) \subseteq \{1,\ldots,J\}$, and $\#(\mathbf{Dom}(a)) = \#(\mathbf{Rng}(a))$ where $\#(S)$ is the number of elements in the set *S*. Note that some tracks from sensor 1 may not be assigned to any tracks from sensor 2, and some tracks from sensor 2 may not be assigned to any tracks from sensor 2.

The following calculation of assignment probability is an approximation that is exact when the targets are stationary. Let Z_i^1 and Z_j^2 be the measurements associated to sensor 1 track *i* and sensor 2 track *j* respectively, and let $\mathcal{L}(Z_i^1 \mid x,y,k)$ and $\mathcal{L}(Z_j^2 \mid x,y,k)$ be the likelihoods of these observations given $(\mathbb{X},\mathbb{Y},\mathbb{K}) = (x,y,k)$. Define

$P_d^1(x,y,k) =$ the probability that sensor 1 will detect a target in state (x,y,k)

$P_d^2(x,y,k) =$ the probability that sensor 2 will detect a target in state (x,y,k).

The likelihood functions $\mathcal{L}(Z_i^1 \mid x,y,k)$ and $\mathcal{L}(Z_j^2 \mid x,y,k)$ contain factors of $P_d^1(x,y,k)$ and $P_d^2(x,y,k)$ respectively. The posteriors can be written as follows

$$\begin{aligned} p_i^1(x)\beta_i^1(y,k) &= \frac{1}{D_i^1}\mathcal{L}\left(Z_i^1 \mid x,y,k\right)p^0(x)\beta^0(y,k) \\ p_j^2(x)\beta_j^2(y,k) &= \frac{1}{D_j^2}\mathcal{L}\left(Z_j^2 \mid x,y,k\right)p^0(x)\beta^0(y,k) \end{aligned} \tag{2}$$

where

$$\begin{aligned} D_i^1 &= \sum_{k=1}^{K}\int\int\mathcal{L}\left(Z_i^1 \mid x,y,k\right)p^0(x)\beta^0(y,k)dxdy \\ D_j^2 &= \sum_{k=1}^{K}\int\int\mathcal{L}\left(Z_j^2 \mid x,y,k\right)p^0(x)\beta^0(y,k)dxdy. \end{aligned}$$

Let $S^1(i)$ denote sensor 1 target *i*, and $S^2(j)$ denote sensor 2 target *j*. The likelihood that the sets of observations Z_i^1 and Z_j^2 are produced by $S^1(i)$ given that $S^1(i) = S^2(j)$ is

$$\hat{\Lambda}(i,j) = \sum_{k=1}^{K}\int\int\mathcal{L}\left(Z_i^1 \mid x,y,k\right)\mathcal{L}\left(Z_j^2 \mid x,y,k\right)p^0(x)\beta^0(y,k)dxdy$$

$$= D_i^1 D_j^2 \sum_k \iint \frac{p_i^1(x)\beta_i^1(y,k)p_j^2(x)\beta_j^2(y,k)}{p^0(x)\beta^0(y,k)} dxdy \,. \tag{3}$$

We need to specify the prior (association) probability that $S^1(i) = S^2(j)$. If we knew the number N of targets present (detected or not), then we would choose $1/N$ as this probability. We do know the expected number $\bar{\nu}$. So we will take

$$\begin{aligned} \gamma(i,j) &\equiv \text{ prior probability that } \mathrm{S}^1(i) = S^2(j) \\ &= 1/\bar{\nu}. \end{aligned} \tag{4}$$

We shall also assume that $\bar{\nu} \geq 1$ so that $\gamma(i,j)$ is a probability. Cases where $\bar{\nu} < 1$ are not of much interest for track-to-track association, so this not a restrictive assumption.

The likelihood $\Lambda(i,j)$ that Z_i^1 and Z_j^2 are produced by $S^1(i)$ *and* that $S^1(i) = S^2(j)$ is

$$\Lambda(i,j) = \hat{\Lambda}(i,j)\gamma(i,j) = D_i^1 D_j^2 \sum_k \iint \frac{p_i^1(x)\beta_i^1(y,k)p_j^2(x)\beta_j^2(y,k)}{\bar{\nu}\, p^0(x)\beta^0(y,k)} dxdy \,. \tag{5}$$

The likelihood $\Lambda(i,0)$ that sensor 1 target i is *not* associated to any sensor 2 target is the likelihood that sensor 1 measurements Z_i^1 are from target i and that this target is not detected by sensor 2. Specifically

$$\Lambda(i,0) = D_i^1 \sum_k \int\int p_i^1(x)\beta_i^1(y,k)\left(1 - P_d^2(x,y,k)\right) dxdy \,. \tag{6}$$

Similarly the likelihood that sensor 2 target j is *not* associated to any sensor 1 target is

$$\Lambda(0,j) = D_j^2 \sum_k \int\int p_j^2(x)\beta_j^2(y,k)\left(1 - P_d^1(x,y,k)\right) dxdy \,. \tag{7}$$

Assuming that the assignment likelihoods for pairs of tracks are independent, we can calculate the assignment probability for $a \in A(I,J)$ as follows:

$$\begin{aligned} \mathbf{Pr}\{a\} &= C\left(\prod_{i \in \mathbf{Dom}(a)} \Lambda\left(i,a(i)\right)\right)\left(\prod_{\substack{i=1 \\ i \notin \mathbf{Dom}(a)}}^{I} \Lambda(i,0)\right)\left(\prod_{\substack{j=1 \\ j \notin \mathbf{Rng}(a)}}^{J} \Lambda(0,j)\right) \\ &= C\prod\nolimits_{i=1}^{I} D_i^1 \prod\nolimits_{j=1}^{J} D_j^2 \left(\prod_{i \in \mathbf{Dom}(a)} L\left(i,a(i)\right)\right)\left(\prod_{\substack{i=1 \\ i \notin \mathbf{Dom}(a)}}^{I} L(i,0)\right)\left(\prod_{\substack{j=1 \\ j \notin \mathbf{Rng}(a)}}^{J} L(0,j)\right) \end{aligned} \tag{8}$$

where C is a normalizing constant and $L(i,j)$ is defined as follows:

$$L(i,j) = \begin{cases} \sum_k \int\int \frac{p_i^1(x)\beta_i^1(y,k)p_j^2(x)\beta_j^2(y,k)}{\bar{\nu}\, p^0(x)\beta^0(y,k)} dydx & \text{if } (i,j) \in \{1,\ldots,I\} \times \{1,\ldots,J\} \\ \sum_k \int\int p_i^1(x)\beta_i^1(y,k)\left(1 - P_d^2(x,y,k)\right) dxdy & \text{if } i \in \{1,\ldots,I\} \text{ and } j = 0 \\ \sum_k \int\int p_j^2(x)\beta_j^2(y,k)\left(1 - P_d^1(x,y,k)\right) dxdy & \text{if } i = 0 \text{ and } j \in \{1,\ldots,J\}. \end{cases} \tag{9}$$

Assuming $L(i,0) > 0$ for all i and $L(0,j) > 0$ for all j, equation (8) may be written as

$$\mathbf{Pr}\{a\} = C' \prod_{i \in \mathbf{Dom}(a)} \ell\left(i,a(i)\right) \tag{10}$$

where

$$C' = C \prod_{i=1}^{I} D_i^1 L(i,0) \prod_{j=1}^{J} D_j^2 L(0,j)$$

and

$$\ell(i,j) = \frac{L(i,j)}{L(i,0)L(0,j)} \text{ for } (i,j) \in \{1,\ldots,I\} \times \{1,\ldots,J\} .$$

2.4 Special Cases.

In this subsection we consider a number of special cases of (9) and compare these results to those in [1].

2.4.1 Target State is Position-Velocity, Gaussian Posterior Distributions

First, suppose that the target state is $\mathbb{X}$, the target's 6-dimensional position and velocity in E^6. In this case equation (9) becomes

$$L(i,j) = \begin{cases} \int_{E^6} \frac{p_i^1(x) p_j^2(x)}{\bar{\nu} p^0(x)} dx & \text{if } (i,j) \in \{1,\ldots,I\} \times \{1,\ldots J\} \\ \int_{E^6} p_i^1(x)\left(1 - P_d^2(x)\right) dx & \text{if } i \in \{1,\ldots,I\} \text{ and } j = 0 \\ \int_{E^6} p_j^2(x)\left(1 - P_d^1(x)\right) dx & \text{if } i = 0 \text{ and } j \in \{1,\ldots,J\} \end{cases} \quad (11)$$

and equation (10) stays as before.

One can check that equations (8) and (11) above are equivalent to equations (8) – (10) of reference [1] and that the results presented in [1] for calculating probability of track association are a special case of the results presented here.

Continuing with the case where the target state is $\mathbb{X}$, assume further that the posterior target state distributions are Gaussian and that p^0 equals a constant ρ over the region of interest. We also assume that P_d^1 and P_d^2 do not depend on *x*. Since P_d^1 and P_d^2 are constant they do not affect the posterior distributions. Let

$\left\{p_i^1; i = 1,\ldots,I\right\}$ be the set of sensor 1 tracks at time t

$\left\{p_j^2; j = 1,\ldots,J\right\}$ be the set of sensor 2 tracks at time t,

where

x_i^1 and V_i^1 are the mean and covariance of p_i^1

x_j^2 and V_j^2 are the mean and covariance of p_j^2.

Now (11) becomes

$$L(i,j) = \begin{cases} (\bar{\nu}\rho)^{-1} \left[\det\left(2\pi\left(V_i^1 + V_j^2\right)\right)\right]^{-1/2} \exp\left(-\frac{1}{2}\chi_{ij}^2\right) & \text{if } (i,j) \in \{1,\ldots,I\} \times \{1,\ldots,J\} \\ 1 - P_d^2 & \text{if } i \in \{1,\ldots,I\} \text{ and } j = 0 \\ 1 - P_d^1 & \text{if } i = 0 \text{ and } j \in \{1,\ldots,J\} \end{cases} \quad (12)$$

where

$$\chi_{ij}^2 = \left(x_i^1 - x_j^2\right)^T \left(V_i^2 + V_j^2\right)^{-1} \left(x_i^1 - x_j^2\right).$$

This is the same as equation (18) in reference [1].

2.4.2 Target Type, No Matching Features, Gaussian Position-Velocity Posterior Distributions

Suppose that the two sensors produce feature measurements, but there are no matching features so that the target state space is $(\mathbf{E},\mathbf{K})$. Define

$$\varphi^0(k) = \text{prior probability that } \mathbb{K}=k \text{ for } k=1,\ldots,K .$$

Then the prior distribution becomes

$$\alpha^0(x,k) = p^0(x)\varphi^0(k) \text{ for } (x,k)\in \mathbf{E}\times\mathbf{K}$$

Suppose there are *M* sensor 1 features and *N* sensor 2 features. Let

$z_m^1(i)$ be the observed value of the *m*th sensor 1 feature for sensor 1 target *i* at time *t*

$z_n^2(j)$ be the observed value of the *n*th sensor 2 feature for sensor 2 target *j* at time *t*

For this discussion we will assume that the sensor features are independent of the kinematic state, *x*. Let

$$\mathcal{L}_m^1(z\,|\,k) = \mathbf{Pr}\{\text{observed value of } m\text{th sensor 1 feature} = z\,|\,\mathrm{K}=k\}$$
$$\mathcal{L}_n^2(z\,|\,k) = \mathbf{Pr}\{\text{observed value of } n\text{th sensor 2 feature} = z\,|\,\mathrm{K}=k\}.$$

We correspondingly assume that the kinematic measurements are independent of target type.

Suppose that the detection probabilities of the two sensors depend *k* but not on *x*. Then we may compute the posterior on target type given the sensor 1 feature observations by

$$\varphi_i^1(k) = \frac{\varphi^0(k)P_d^1(k)\prod_{m=1}^{M}\mathcal{L}_m^1\left(z_m^1(i)\,|\,k\right)}{\sum_{k'=1}^{K}\varphi^0(k')P_d^1(k')\prod_{m=1}^{M}\mathcal{L}_m^1\left(z_m^1(i)\,|\,k'\right)} \text{ for } k=1,\ldots,K , \tag{13}$$

and the posterior on target type given the sensor 2 feature observations by

$$\varphi_j^2(k) = \frac{\varphi^0(k)P_d^2(k)\prod_{n=1}^{N}\mathcal{L}_n^2\left(z_n^2(j)\,|\,k\right)}{\sum_{k'=1}^{K}\varphi^0(k')P_d^2(k')\prod_{n=1}^{N}\mathcal{L}_n^2\left(z_n^2(j)\,|\,k'\right)} \text{ for } k=1,\ldots,K . \tag{14}$$

Then equation (9) becomes

$$L(i,j) = \begin{cases} \int \frac{p_i^1(x)p_j^2(x)}{\bar{\nu}\,p^0(x)}dx \sum_k \frac{\varphi_i^1(k)\varphi_j^2(k)}{\varphi^0(k)} & \text{if } (i,j)\in\{1,\ldots,I\}\times\{1,\ldots,J\} \\ \sum_k \varphi_i^1(k)\left(1-P_d^2(k)\right) & \text{if } i\in\{1,\ldots,I\} \text{ and } j=0 \\ \sum_k \varphi_j^2(k)\left(1-P_d^1(k)\right) & \text{if } i=0 \text{ and } j\in\{1,\ldots,J\}. \end{cases} \tag{15}$$

In the case where the posteriors on position-velocity are Gaussian and $p^0(x)=\rho$, equation (15) becomes

$$L(i,j) = \begin{cases} \left(\bar{\nu}\rho\right)^{-1}\left[\det\left(2\pi\left(V_i^1+V_j^2\right)\right)\right]^{-1/2} e^{-\chi_{ij}^2/2}\sum_k \frac{\varphi_i^1(k)\varphi_j^2(k)}{\varphi^0(k)} & \text{if } (i,j)\in\{1,\ldots,I\}\times\{1,\ldots,J\} \\ \sum_k \varphi_i^1(k)\left(1-P_d^2(k)\right) & \text{if } i\in\{1,\ldots,I\} \text{ and } j=0 \\ \sum_k \varphi_j^2(k)\left(1-P_d^1(k)\right) & \text{if } i=0 \text{ and } j\in\{1,\ldots,J\} \end{cases} \tag{16}$$

3. FINDING THE MAP ASSIGNMENT

We now discuss how to find the assignment $a^* \in A(I,J)$ such that

$$\mathbf{Pr}\{a^*\} = \max_{a \in A(I,J)} \mathbf{Pr}\{a\}.$$

The linear programming approach to finding a^* outlined below may be interpreted as finding assignments which minimize thresholded association costs. In the terminology set forth in [1], the MAP algorithm may thus be viewed as an adaptive thresholding approach.

3.1 Adaptive Thresholding

For an assignment $a \in A(I,J)$, define ξ_a as follows:

$$\xi_a(i,j) = \begin{cases} 1 \text{ if } a \text{ assigns sensor 1 track } i \text{ to sensor 2 track } j \\ 0 \text{ otherwise} \end{cases} \quad \text{for } (i,j) \in \{1,\ldots,I\} \times \{1,\ldots,J\}.$$

Let

$$C_{ij} = -2\ln\left(\ell(i,j)\right) \text{ for } (i,j) \in \{1,\ldots,I\} \times \{1,\ldots,J\}. \tag{17}$$

The factor of 2 in the definition of C_{ij} is placed there for convenience in dealing with Gaussian distributions. Finding a^* is equivalent to finding ξ^* that solves the following integer programming problem:

$$\begin{aligned} \text{Minimize} \quad & \sum_{i=1}^{I} \sum_{j=1}^{J} C_{ij}\xi(i,j) \\ \text{Subject to} \quad & \xi(i,j) \in \{0,1\} \text{ for all } (i,j) \in \{1,\ldots,I\} \times \{1,\ldots,J\} \\ & \sum_{j=1}^{J} \xi(i,j) \le 1 \text{ for all } i \in \{1,\ldots,I\} \\ & \sum_{i=1}^{I} \xi(i,j) \le 1 \text{ for all } j \in \{1,\ldots,J\}. \end{aligned} \tag{18}$$

3.1.1 Linear Programming Formulation

In order to convert this to a linear programming problem and apply the JVC algorithm (reference [4]), we extend the definition of ξ to $(i,j) \in \{1,\ldots,I\} \times \{J+1,\ldots,J+I\}$ as follows

$$\xi(i, J+i) = \begin{cases} 1 \text{ if sensor 1 track } i \text{ is not assigned to any sensor 2 track} \\ 0 \text{ otherwise} \end{cases}. \tag{19}$$

For the remainder of the $(i,j) \in \{1,\ldots,I\} \times \{J,\ldots,J+I\}$, we let $\xi(i,j) \in [0,1]$ be arbitrary. This part of the definition of ξ will not enter into the solution as will be seen below. We also extend the cost function to $\{1,\ldots,I\} \times \{J+1,\ldots,J+I\}$ as follows

$$\begin{aligned} C_{ij} &= -2\ln\left(\ell(i,j)\right) && \text{for } (i,j) \in \{1,\ldots,I\} \times \{1,\ldots,J\} \\ &= 0 && \text{for } i \in \{1,\ldots,I\} \text{ and } j = J+i \\ &= \infty && \text{otherwise.} \end{aligned} \tag{20}$$

From equation (10), which gives the probability for assignment *a*, one can verify that if *a* does not assign sensor 1 track *i* to any sensor 2 track *j*, then this is equivalent to multiplying the product in equation (10) by a factor of 1. Specifically, since *i* is not in the domain of *a*, it is not included in the product. This is of course equivalent to adding 0 to the logarithm of the product. Thus the extended cost definition in (20) produces a suitable cost function, i.e., it is –2 times the log of the assignment probability plus a constant term that applies to all assignments. The costs of ∞ in (20) will not enter into an optimal solution since such a solution will never contain an assignment with cost ∞.

As noted in [1], finding a^* is equivalent to finding ξ^* to solve the following linear programming problem, and in this form the problem may be solved efficiently using the JVC algorithm described in [4].

$$\begin{aligned}
&\text{Minimize} && \sum_{i=1}^{I}\sum_{j=1}^{J+I} C_{ij}\xi(i,j) \\
&\text{Subject to} && \xi(i,j)\in[0,1] \text{ for all } (i,j)\in\{1,\ldots,I\}\times\{1,\ldots,J+I\} \\
& && \sum_{j=1}^{I+J}\xi(i,j)=1 \text{ for all } i\in\{1,\ldots,I\} \\
& && \sum_{i=1}^{I}\xi(i,j)\le 1 \text{ for all } j\in\{1,\ldots,J+I\}.
\end{aligned} \tag{21}$$

To illustrate this method more concretely, let us return to the special case considered in section 2.4 where the state space is position-velocity and the target track posteriors are Gaussian. For this case the cost function *C* becomes

$$C_{ij} = \chi_{ij}^2 - A_{ij} \text{ for } (i,j)\in\{1,\ldots,I\}\times\{1,\ldots,J\} \tag{22}$$

where

$$A_{ij} = -\ln\left[\left(\overline{\nu}\rho(1-P_d^1)(1-P_d^2)\right)^2 \det\left(2\pi(V_i^1+V_j^2)\right)\right] \tag{23}$$

From (19) - (21) we see that a MAP assignment will never assign *i* to *j* when $\chi_{ij}^2 > A_{ij}$ because it can obtain a lower cost (namely 0) by not assigning *i* to *j*. In the terminology of [1], A_{ij} is an adaptive assignment threshold. Equation (9.17) of [6] produces this same adaptive thresholding scheme for the special case considered in this section. Reference [6] also considers the possibility of false alarms and correlation in the covariance of the tracks. This adaptive threshold is in contrast to the fixed threshold used in the standard chi-squared assignment procedure (see [5] or section 2.1 of [1]). In the fixed threshold procedure, A_{ij} is replaced by a constant A_0 that does not depend on *i* or *j*.

3.1.2 Detection Probability Considerations

Suppose sensor 2 cannot detect sensor 1 target *i*. How does the formulation above respond to this situation? If sensor 1 can not detect sensor 2 target *i*, this means

$$\sum_k \iint p_i^1(x)\beta_i^1(y,k)P_d^2(x,y,k)dxdy = 0\,.$$

Since the likelihood function $\mathcal{L}(Z_j^2 \mid x,y,k)$ contains the factor $P_d^2(x,y,k)$, we know that

$$\mathcal{L}\left(Z_j^2 \mid x,y,k\right)=0 \text{ whenever } p_i^1(x)\beta_i^1(y,k)>0\,.$$

From this we have that

$$\begin{aligned}
L(i,j) &= \sum_k \iint \frac{p_i^1(x)\beta_i^1(y,k)\mathcal{L}\left(Z_j^2\mid x,y,k\right)p^0(x)\beta^0(y,k)}{\overline{\nu}\,p^0(x)\beta^0(y,k)D_j^2}\,dydx \\
&= \sum_k \iint \frac{p_i^1(x)\beta_i^1(y,k)\mathcal{L}\left(Z_j^2\mid x,y,k\right)}{\overline{\nu}D_j^2}\,dydx \\
&= 0.
\end{aligned}$$

From this it follows that

$$\ell(i,j)=0 \text{ and } C_{ij}=\infty \text{ for } j=1,\ldots,J\,. \tag{24}$$

From (24) we can see that the MAP assignment will never assign a sensor 2 target to sensor 1 target *i*. Similarly if sensor 1 cannot detect sensor 2 target *j*, it will never be assigned to a sensor 1 target.

3.2 Threshold Adjustment

One issue that arises immediately in the standard chi-squared assignment algorithm is the problem of choosing a threshold. Simulation results in [1] show that the performance of the chi-squared algorithm is quite sensitive to this choice. Theoretically, then, one benefit of using the MAP algorithm is that it removes the necessity of choosing such a threshold. However, in [1], the authors found that they could improve the association performance of the MAP algorithm by adding a positive additional threshold A' in the range of 3 to 5 to the adaptive assignment threshold A_{ij} given in equation (23). This produces revised costs $C'_{ij} = C_{ij} - A'$. The effect of adding a fixed (positive) threshold to A_{ij} is to make the assignment obtained from the costs C'_{ij} more likely to assign sensor 1 tracks to sensor 2 tracks. When this was done in [1], they obtained improved results.

The authors of [1] explained this improvement by noting that we should not expect a method that maximizes the posterior association probability to maximize the expected number of correct matches. To support this statement, we give a brief indication of why this is so. The association probability computed in equation (8) is the product of pairwise association probabilities. The MAP is the set of assignments of sensor 1 to sensor 2 targets plus null matches (i.e., no assignments) that maximizes this product. Each factor in the product is the probability that a particular assignment or null match in the set is correct. By contrast, the expected number of correct matches for this set of assignments is computed by summing the probability of each assignment or null match being correct over all members of the set. Thus in one case we are maximizing the product of probabilities. In the other we are seeking to maximize the sum of these probabilities. We should not expect the same set of assignments to maximize both. We conjecture that the ability of the MAP method to allow null matches is the reason that the MAP assignment does so well in fraction of correct matches.

The discussion and results in [1] indicate that MAP method will be substantially sub-optimal without an additional threshold. However, contrary to the experimental MAP results in [1], the Monte Carlo results presented in the following section indicate that the MAP algorithm provides near optimal matching performance without adding an additional threshold A'. Specifically, the fraction of correct matches obtained using a zero threshold adjustment $(A' = 0)$ differs from the fraction of correct matches obtained at the (experimentally determined) optimum threshold adjustment by at most 1%. Thus, while the MAP does not provide optimal performance without an additional threshold, it still performs in a near optimal fashion.

4. EVALUATION OF EXTENDED MAP METHOD BY MONTE CARLO SIMULATION

In this section, we give examples that indicate that the extended MAP method provides near optimal matching performance without the need for manually selecting a fixed threshold. In addition, these examples show the improvement that one can obtain using feature data as well as position-velocity information. To do this we perform a simulation very similar to the one described in section 2.3 of [1].

4.1 Description of Simulation

Following [1] we simulate the posterior distributions of 40 tracks. We take the detection probability to be $P_d^1 = p_1$ for sensor 1 and $P_d^2 = p_2$ for sensor 2 for all target states. For each of the 40 tracks we make an independent draw with success probability p_1 to determine if it is detected by sensor 1 and another independent draw with probability of success p_2 to determine if it is detected by sensor 2.

The target state space is 6D position-velocity and target type. There are 5 target types. The track posteriors in position-velocity space are Gaussian. We specify base covariances V_1 and V_2 for sensor 1 and sensor 2 tracks respectively. The target density $\rho\bar{v}$ was chosen to equal 5 targets per 3σ hyper volume corresponding to the covariance matrix $V_1 + V_2$. The true position of each target was chosen from the uniform density ρ over a region of volume *H* such that $\rho\,\bar{v}H = 40$. For each target, the type was chosen independently from a uniform distribution on $\{1,\ldots,5\}$. To obtain the posterior position-velocity distribution of a sensor 1 target we randomly varied the lengths of the axes of the base covariance V_1 over plus or minus 10% and randomly rotated the axes over plus or minus 1 degree to produce the posterior covariance. The mean was obtained by taking a draw from the Gaussian distribution with mean 0 and this covariance and adding this draw to the actual target position. The same procedure was followed for sensor 2 targets using the base covariance V_2.

4.2 Description of Features

Each sensor measures one feature, and the features do not match, e.g., length for sensors 1 and temperature for sensor 2. The feature information can be used determine target type which will assist in matching sensor 1 and sensor 2 targets. We use the following simplified model for feature values to illustrate the potential value that feature information might add in pairing targets.

Let

$$z(1,i) = \text{measured value of the sensor 1 feature for target } i$$
$$z(2,j) = \text{measured value of the sensor 2 feature for target } j,$$

where

$$z(1,i) \sim N(\mu_{1k},\sigma_1^2) \text{ and } z(2,j) \sim N(\mu_{2k},\sigma_2^2) \tag{25}$$

For each sensor 1 or sensor 2 target, we drew a sample value of the measured sensor 1 or sensor 2 feature from the distribution for its type. We used these values to compute the posterior on target type according to equations (13) and (14) above. Let

$$\begin{aligned} SNR(1) &= \min_{k_1 \neq k_2} \frac{(\mu_{1k_1} - \mu_{1k_2})^2}{\sigma_1^2},\ SNR(2) = \min_{k_1 \neq k_2} \frac{(\mu_{2k_1} - \mu_{2k_2})^2}{\sigma_2^2} \\ SNR &= \min\left\{SNR(1), SNR(2)\right\}. \end{aligned} \tag{26}$$

We use SNR as simple measure of the degree of separation in the features.

4.3 Performance of Extended MAP

Taking $p_1 = p_2 = 0.9$, we performed 100 replications of the simulation described above. For each replication, we used the extended MAP algorithm to find the MAP match. We then paired sensor 1 and sensor 2 targets according to this match and scored the results. For each replication, the maximum possible number of correct matches equals the number of targets detected by at least one sensor. If sensor 1 target *i* is correctly matched to sensor 2 target *j* that counts as one correct match. If sensor 1 target *i* is correctly not matched with any sensor 2 target that is a correct match, and if sensor 2 target *j* is correctly not matched with any sensor 1 target that is a correct match. For each replication, the number of correct matches divided by number of detected targets is the fraction of correct matches. Figure 1 below shows the fraction of correct matches averaged over the 100 replications for a number of cases.

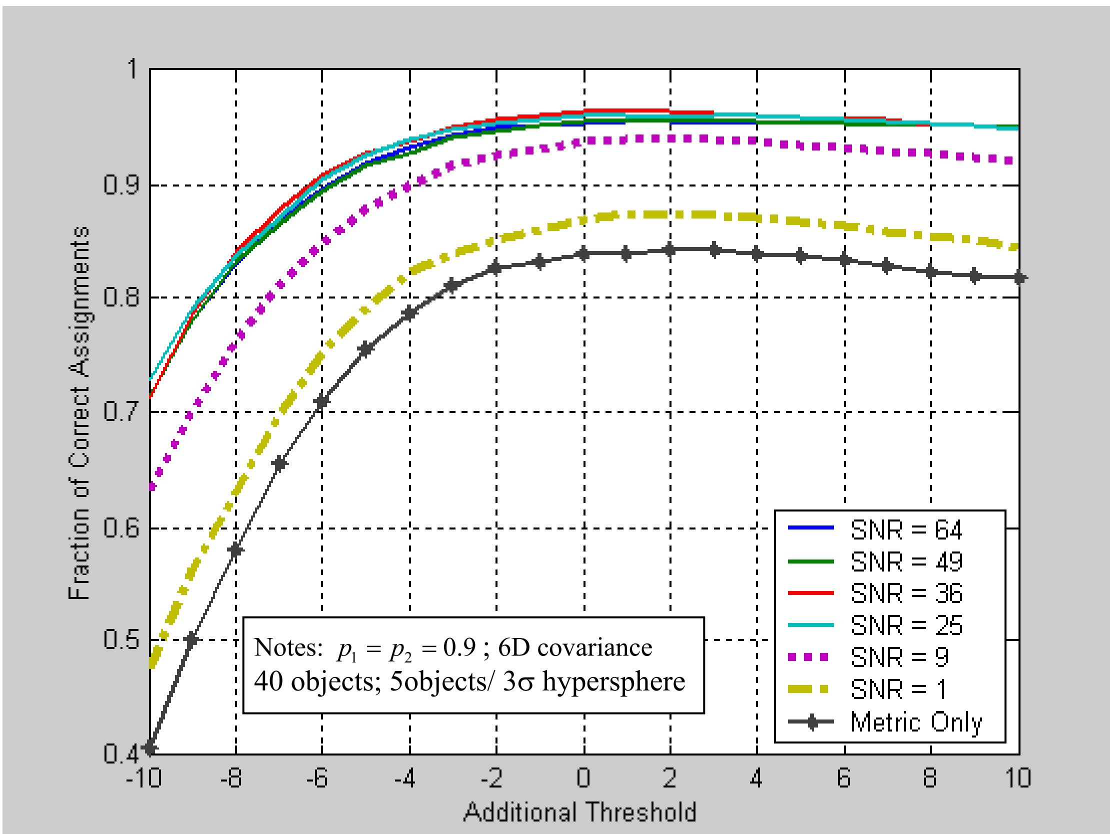

Figure 1: Improvement in Association Probability from Using Feature Information

Let us concentrate on the case where the additional threshold is 0. In this case, we are computing the MAP as it is described in section 2 and using the definitions of the likelihood functions in equation (16). The black (lowest) line in Figure 1 shows the fraction of correct matches that we obtain from using only the metric information. It is 0.83 for this situation. The lines above it show the improvement that is obtained when feature information is added. As the SNR (defined in (26)) increases, the fraction of correct matches increases to 0.96.

We have performed the above simulations (using metric data only) for a range of number of targets, target density $\bar{\nu}$, detection probabilities p_1 and p_2, and position-velocity dimensions used for matching. In all of these cases, the maximum percentage of correct matches occurs at a threshold adjustment A' between –2 and +2. In most situations, the difference in the percentage of correct matches obtained from using $A' = 0$ and the optimum adjustment is very small. Thus, contrary to the results in [1], our results indicate that the MAP algorithm does not require adding an additional threshold to obtain near optimal performance.

5. BIAS REMOVAL VIA CORRELATION

In this paper we assume that bias is present only in the object state position estimates, and that this bias is translational in nature (i.e. no rotational or scaling biases). Many methods for removing translation bias are well-known to the multiple sensor data fusion community. A partial list includes centroiding and singleton matching; for an overview of these and other methods, see [1] and [2].

A popular alternative to centroiding and singleton matching that has been proposed is to consider all pairwise matches. The difference between the coordinates of the matched pair is used as a possible bias. Given I sensor 1 targets and J sensor 2 targets, this requires $I \times J$ iterations. Each of the $I \times J$ biases considered must then be evaluated (if one were simply to average across the whole set then one has recovered the centroiding method in an inefficient manner). One possible evaluation is to perform a MAP association for each of these possible biases and to select the bias estimate for which the MAP association has the least cost. A common result from the various implementations of this method is that it is very computationally demanding and not feasible when the number of detected targets is significantly larger than 10. We will refer to this bias removal technique as pairwise matching.

Another technique for removing sensor bias for small numbers of tracks is the direct optimization method described in [2]. This method begins by enumerating all possible track association hypotheses. Then, for a given association hypothesis a, the relative sensor bias estimate $\tau(a)$ is computed as the average difference between the position coordinates of the associated tracks. Each association hypothesis and relative sensor bias estimate is evaluated by performing a MAP association using the modified cost function $C(i, j, \tau(a)) = \chi^2_{ij}(\tau(a)) - A_{ij}$, where

$$\chi^2_{ij}(\tau(a)) = \left(x_i^1 - x_j^2 - [\tau(a)\ \ 0]^T\right)^T \left(V_i^2 + V_j^2\right)^{-1} \left(x_i^1 - x_j^2 - [\tau(a)\ \ 0]^T\right).$$

The association hypothesis and bias estimate with the lowest MAP association cost is selected as the solution. Since the number of hypotheses grows factorially with the number of tracks, this method is computationally demanding when the number of targets is greater than 7. An evaluation of the direct optimization method by Monte Carlo simulation for a small number of targets may be found in [2].

The translation bias removal technique we describe subsequently is computationally tractable for large numbers of targets and is effective in the presence of missing tracks. This technique is borrowed from the computer vision community [7] and exploits the computational efficiency of the Fast Fourier Transform (FFT) for doing correlation; hence, we will refer to this bias removal algorithm as FFT correlation.

Suppose that we have tracks from two sensors that we need to correlate. The first step in FFT correlation is to form discrete approximations of the target densities for each set of tracks. To illustrate, suppose posteriors on position for each track are specified by a 3 dimensional distribution. In this case, a discrete approximation for the sensor 1 target densities will be a 3 dimensional grid containing N^3 cells, with each cell having a value equal to the target density in that cell. This is computed by summing the probability density in that cell over all targets in the grid. Here, N is a parameter constrained by the available computational resources. A similar discrete approximation may be constructed for the sensor 2 target densities.

Let $M_1(x)$ and $M_2(x)$ be the discrete approximations for the sensor 1 and sensor 2 target densities, respectively. For each translation τ in the discrete space, define the correlation $\Gamma(\tau)$ to be

$$\Gamma(\tau) = \sum_x M_1(x-\tau) M_2(x). \tag{27}$$

To compute Γ note that the convolution of M_1 and M_2 at τ is defined as

$$M_1 * M_2(\tau) = \sum_x M_1(\tau - x) M_2(x). \tag{28}$$

From (27) - (28), we see that correlation is convolution with an appropriate "flip" or rotation. Keeping this in mind, and using a calculation similar to one which shows that the Fourier transform of $M_1 * M_2$ is the product of the Fourier transforms of M_1 and M_2, we see that the correlation Γ may be computed efficiently as follows:

$$\Gamma(\tau) = \text{IFFT}\left[\overline{\text{FFT}(M_1)} \times \text{FFT}(M_2)\right](\tau). \tag{29}$$

Here, FFT and IFFT denote the Fast Fourier Transform and the Inverse Fast Fourier Transform, and $\overline{\text{FFT}(M_1)}$ is the complex conjugate of $\text{FFT}(M_1)$. The estimated translation bias is the value of τ^* which maximizes Γ.

6. EVALUATION OF FFT CORRELATION BY MONTE CARLO SIMULATION

The FFT correlation method was applied to simulation data provided by XonTech and Photon Research Associates.

6.1 Description of Simulation

Photon Research Associates (PRA) simulated ground truth trajectories for 8 objects. Using high fidelity tracking simulations they produced 100 Monte Carlo runs for two independent tracking systems. The sensor track for each object consisted of feature as well as metric information, and there was a translational bias in the position estimates. There were 4 object types. Sensor 1 did not detect one type 25% of the time, and sensor 2 did not detect one type 100% of the time. Objects were also localized in space by type.

Given the accuracy of the two sensors and the strong feature data, the bias removal problem for 8 objects is a trivial one. To provide a challenging test for our bias removal algorithms, we added 40 additional objects to each one of the original PRA and XonTech data sets. The objects were added in a manner that reflected the original statistics and retained the flavor of the PRA-Xontech data.

6.2 Performance of FFT Correlation

Metron used the PRA-XonTech data sets to test the FFT correlation technique against two common algorithms for bias removal: centroiding and pairwise matching. After removing the bias, the extended MAP method was used to find the assignments. In these evaluations, the metric information consisted of position estimates, and the feature information consisted of one non-matching feature. The sensor detection probabilities for type k, $P_d^1(k)$ and $P_d^2(k)$, were calculated as the percentage of time objects of type k were seen by sensor 1 or sensor 2 in the original 100 Monte Carlo runs.

Table 1 shows the performance of the bias removal techniques. For reference, the no bias removal case is included. The columns labeled 'Metric' and 'Metric + Features' indicate whether metric information or metric and feature information were used to calculate the MAP assignment. In this table, if sensor 1 track i is correctly matched with sensor 2 track j, that counts as a correct match. The fraction of correct object-to-object matches is defined as the number of correct matches divided by the number of objects detected by both sensors. The percent of correct object-to-object matches reported in the table is the average over the 100 runs.

Percent of Object-to-Object Matches				
Bias Removal Technique	**8 Objects**		**48 Objects**	
	Metric	Metric + Features	Metric	Metric + Features
None	14	51	6	29
Centroiding	0	0	0	0
Pairwise Matching	97	98	NA	NA
FFT Correlation	91	97	95	98

The reason for the poor performance of centroiding is that the PRA-XonTech data has all the ingredients that make centroiding fail: objects are localized by type and, in many cases, objects of one type are seen by only one sensor. Though the table indicates that pairwise matching is slightly superior to FFT correlation in the 8 objects case, the computational costs of pairwise matching renders it too computationally demanding when dealing with significantly more than 10 objects. In our tests, the pairwise matching method required about 1 hour to estimate the translation bias for a single Monte Carlo run containing 48 objects. Hence there are no results for pairwise matching in the 48 objects case. In contrast, the FFT correlation method was able to estimate the translation bias in about 10 seconds. Thus, FFT correlation is not constrained by the number of objects, and is able to produce similar quality results in either the 8 objects or 48 objects case.

Conclusions

This paper has shown how to extend the MAP association technique of [1] to include feature data and has provided examples of the improvement that one can obtain in matching tracks from two sensors when using feature as well as position-velocity data. In addition, we have provided a more direct proof of the formula for calculating the association probability in equations (9) and (10).

The simulated examples presented in this paper indicate that using no threshold adjustment provides almost optimal matching performance from the MAP algorithm. This seems a substantial benefit over the standard non-adaptive chi-squared thresholding scheme, but it does require estimates of ρ, P_d^1, and P_d^2. Simulation results in [1] show that the performance of the chi-squared non-adaptive matching method is quite sensitive to the choice of threshold.

Our experiments indicate that FFT correlation provides a viable method for removing translational bias between two sets of tracks. In particular, this technique is robust to (i) unequal number of tracks and (ii) large numbers of targets. As our experiments indicate, centroiding fails in the presence of (i) and pairwise matching is not feasible in the face of (ii).

REFERENCES

[1] S. Mori and C-Y Chong, "Effects of Unpaired Objects and Sensor Biases on Track-to-Track Association: Problems and Solutions" in *Proceedings of MSS Sensor and Data Fusion 2000*, Volume I.

[2] S. Mori and C-Y Chong, "BMD Mid-Course Object Tracking: Track Fusion Under Asymmetric Conditions" in *Proceedings of the National Symposium on Sensor and Data Fusion*, 2001.

[3] Mark Williams and Thy Tran, "Bias Removal for Bayesian Handover," Metron Memorandum, 25 July 2001 case (6A64-001).

[4] D. A. Canstañon, "New Assignment Algorithms for Data Association" in *Proc SPIE, Signal and Data Processing of Small Targets*, Vol 1698, 1992.

[5] O. E. Drummond, D. A. Canstañon, and M. S. Bellovin, "Comparison of 2-D Assignment Algorithms for Sparse, Rectangular, Floating Point, Cost Matrices," *Proceedings of the SDI Panels on Tracking*, Issue 4, 1990.

[6] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems,* Artech House, Boston, 1999.

[7] Lisa Gottesfeld Brown, "A Survey of Image Registration Techniques," ACM Computing Surveys, Vol. 24, No. 4, 1992.