# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

# Object Tracking under High Correlation for Kalman & α – β Filter

D. M. Akbar Hussain and Zaki Ahmed*
*Department of Electronic Systems*
*Esbjerg Institute of Technology*
*Aalborg University Esbjerg, Denmark*
*akbar@aaue.dk*
* *zaki424@hotmail.com*

**Abstract.** The investigation presented here compares the advantage of using a Kalman filter as opposed to an $\alpha - \beta$ filter for multi-target tracking systems. The former is often used to speed up the computation time. However, it is shown here that due to the difficulty of data association the benefits are not as great as might be expected when correlation is high. Extensive analyses are performed by selecting various scenarios where the correlation factor affects the performance of $\alpha - \beta$ filter compared with Kalman filter. The research also investigate a new framework of forming a tree clustering scheme to prune false target-measurement pairing introduced when multiple targets are in the same vicinity.

**Key words:** Filtering, Kalman, $\alpha - \beta$, Filter, Target Tracking, State Estimation.

## 1. Introduction

During the past two decades the improved technology available for surveillance systems has generated a great deal of interest in algorithms capable of tracking large number of objects using information from one or more sensors like radar, sonar etc. Typical sensor systems, such as radar, obtain data returns corrupted with noise from true targets and possibly from other objects. In general the tracking problem requires processing of incoming data to produce accurate position and velocity estimates [1-3]. There are two types of uncertainties involved with this incoming data, first the position inaccuracy, as the measurements are corrupted by noise, and second the measurement origin since there may be uncertainty as to which measurement originates from which target [4, 5]. These uncertainties lead to a data association problem and the tracking performance depends not only on the measurement noise but also upon the uncertainty in the measurement origin [6-8]. Therefore, in a multi-target environment extensive computation may be required to establish the correspondence between measurements and tracks at each radar scan [9, 10]. After the data association process, tracks are normally updated using either standard Kalman or $\alpha - \beta$ filter [11-13]. Also tracks whose statistics deviate from the assumed model and shown to be following the same target are normally eliminated [14, 15]. Kalman or

$\alpha - \beta$ filters can be ideal choice for a single target case where one noisy measurement is obtained at each radar scan. In the multi-target tracking case, an unknown number of measurements are received at each radar scan and assuming no false measurements, each one has to be associated with an existing or new tracking filter. When the targets are well apart from each other then forming a measurement prediction ellipse around a track to associate the correct measurement with that track is a standard technique [16]. When targets are near to each other, more than one measurement may fall within the prediction ellipse of a filter and prediction ellipses of different filters may interact. The number of measurements accepted by a filter will therefore be quite sensitive in this situation to the accuracy of the prediction ellipse. Several approaches may be used for this situation [17, 18], one of which is called the Track Splitting Filter algorithm (explained later in the text). In this research as mentioned earlier we extend our investigation and introduce a tree based clustering framework to prune the excessive tracks generated at ambiguity time. A typical recursive multi-target tracking system is shown in figure 1. The algorithm is implemented using an AMD Athlon 64, 2.2 GHz microprocessor on a standard PC for convenience. However, real implementation should be on a much powerful processor for example on a DSP for faster computation.

## 2. Problem Statement

### 2.1 The Data Association Problem

In a general multiple target tracking situation, as explained in the introduction, a number of measurements from an unknown number of targets are received at each radar scan. Typically the radar sensor measures target position in polar co-ordinates, that is in range r and bearing $\theta$. Measurement inaccuracy can normally be modeled as additive zero-mean uncorrelated Gaussian noise on r and $\theta$, with given variances $\sigma_r^2$ and $\sigma_\theta^2$ respectively. The multiple targets tracking problem requires each measurement received at every radar scan to be associated with an existing or new target track. The associated observations are then incorporated in a state estimation algorithm or initialization procedure to produce updated track position and velocity estimates. Basically there are two fundamental approaches to deal with the data association problem, first there is the deterministic approach in which the most likely of several "candidate" associations are formed and then treated as if they were certain, ignoring the fact that this may not necessarily be true. The results of the deterministic association are then used in a standard state estimation algorithm. The Nearest Neighbour Filter (NNF) and the Track Splitting Filter (TSF) are two common examples of the deterministic approach. The second method is a probabilistic model, based on a Bayesian approach, which computes the probabilities of individual associations and state estimations are obtained with associated probabilities. The multiple Hypothesis Tracking (MHT) approach, where a number of hypotheses are generated and evaluated as more data is received, and the Joint Probabilistic Data Association Filter (JPDAF) are examples of algorithms which use the Bayesian approach. The fundamental difference between the deterministic approach and the Bayesian approach is that the latter explicitly takes into account the uncertainty in the measurement origin. The Bayesian approach is the optimal solution to the data association problem, but it is computationally very expensive. The JPDAF method is a suboptimal modification of the Bayesian approach and is simpler since it does not require storage of information regarding hypotheses at previous time instants.

Here we will be concerned with the implementation of a deterministic approach, the TSF algorithm.

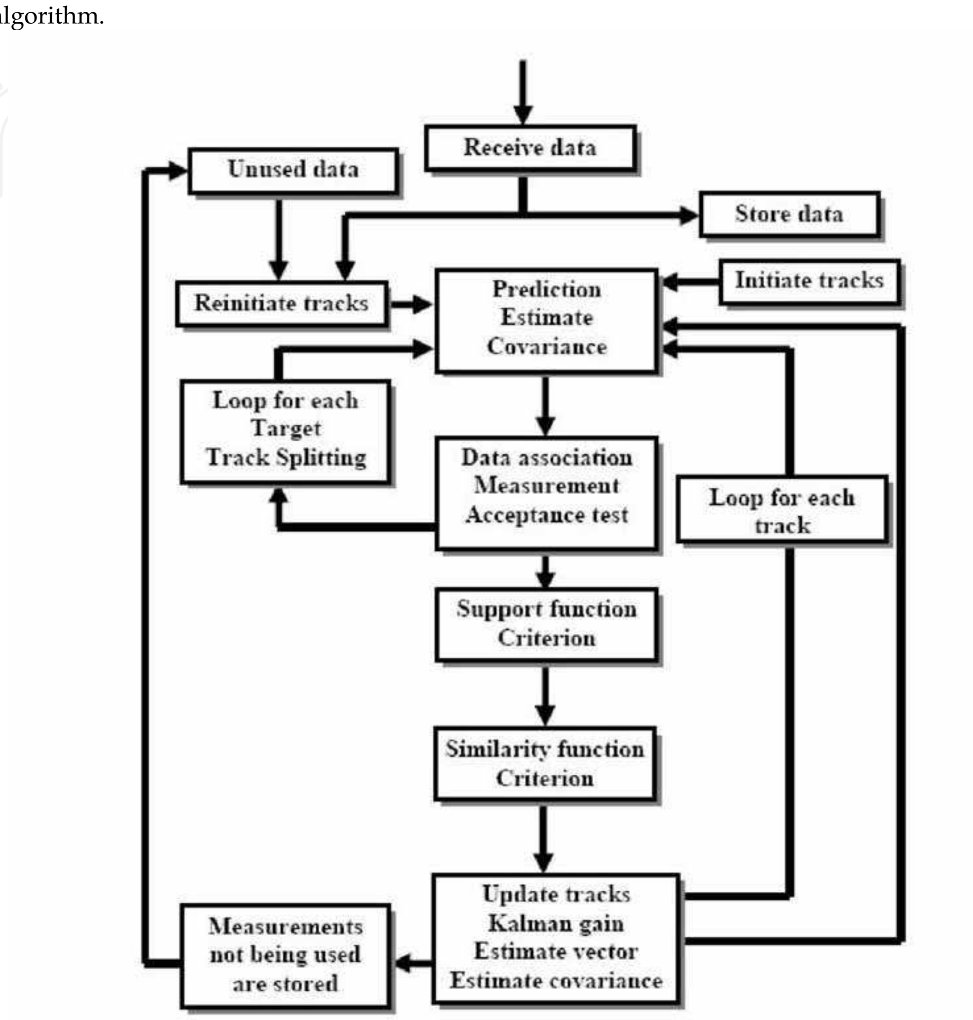


Fig. 1. Recursive Multi-Target Tracking System

### 2.2 State Estimation

For the estimation of target position and velocity from the track data, it is common to use a recursive Kalman filtering algorithm, [17, 18]. This requires a model for the motion of the target being tracked and one often assumed a constant velocity target with random acceleration description given by;

$$\underline{x}_{n+1} = \mathbf{\Phi}\underline{x}_n + \mathbf{\Gamma}\underline{w}_n \tag{1}$$

and the corresponding measurement $\underline{z}_{n+1}$ is given by

$$\underline{z}_{n+1} = \mathbf{H}\underline{x}_{n+1} + \underline{\nu}_{n+1} \tag{2}$$

The state vector, the state transition matrix, the excitation and the measurement matrix are respectively,

$$\underline{x}_{n+1}^T = (x \quad \dot{x} \quad y \quad \dot{y})_{n+1} \tag{3}$$

$$\mathbf{\Phi} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

$$\mathbf{\Gamma} = \begin{bmatrix} \Delta t^2/2 & 0 \\ \Delta t & 0 \\ 0 & \Delta t^2/2 \\ 0 & \Delta t \end{bmatrix} \tag{5}$$

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{6}$$

Here $\Delta t$ is the sampling interval and corresponds to the time interval, assumed uniform, at which radar measurement data is received. The acceleration noise $\underline{w}_n$ is zero mean Gaussian and independent in each Cartesian co-ordinate with covariance $\mathbf{Q}_n$. The measurement noise $\underline{\nu}_n$ is assumed Gaussian with covariance $\mathbf{R}_n$. The Cartesian coordinates of the measurements are obtained from the polar coordinates ($r$, $\theta$) received by radar through a non-linear transformation.

$$x = r\ Cos\ \theta \tag{7}$$

$$y = r\ Sin\ \theta \tag{8}$$

This transformation results in the measurement errors on the Cartesian coordinates being non-Gaussian distributed, but under the reasonable assumption that the measurement errors on the polar coordinates are small compared with the true target coordinates ($r$, $\theta$), it can be shown that the Cartesian errors are bivariate Gaussian random variables [19] with zero mean and covariance $\mathbf{R}_n$ given by

$$\mathbf{R}_n = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix} \tag{9}$$

where,

$$\sigma_x^2 = \sigma_r^2\ Cos^2\theta + \sigma_\theta^2\ r^2\ Sin^2\theta \tag{10}$$

$$\sigma_y^2 = \sigma_r^2\ Sin^2\theta + \sigma_\theta^2\ r^2\ Cos^2\theta \tag{11}$$

$$\sigma_{xy} = 0.5\ (\sigma_r^2 - r^2\ \sigma_\theta^2)\ Sin2\theta \tag{12}$$

Although the analysis in [19] is approximate, it is the only way of avoiding a non-linear tracking filter, because if the measurement errors on the Cartesian coordinates are non-Gaussian distributed the optimum tracking filter in these coordinates is non-linear. On the other hand tracking can be performed in polar coordinates to avoid the correlation introduced by the non-linear transformation but it leads to large dynamic errors when a linear model for target motion is used, even simple constant velocity tracks appear non-linear in polar coordinates and artificial acceleration components are generated. This problem does not arise if tracking is performed in Cartesian coordinates. When the target position measurement errors are correlated, a fourth order full Kalman filter is the optimal tracker in that it minimizes the mean squared error between the estimated and the actual states, and it accounts for the cross-correlation term in the measurement covariance matrix. The standard Kalman filter equations for estimating the position and velocity are;

$$\hat{x}_{n+1/n} = \mathbf{\Phi}\underline{\hat{x}}_n \tag{13}$$

$$\underline{\hat{x}}_{n+1} = \mathbf{\Phi}\underline{\hat{x}}_{n+1/n} + \mathbf{K}_{n+1}\underline{\nu}_{n+1} \tag{14}$$

$$\mathbf{K}_{n+1} = \mathbf{P}_{n+1/n}\mathbf{H}^T\mathbf{B}_{n+1}^{-1} \tag{15}$$

$$\mathbf{P}_{n+1/n} = \mathbf{\Phi}\mathbf{P}_n\mathbf{\Phi}^T + \mathbf{\Gamma}\mathbf{Q}_n\mathbf{\Gamma}^T \tag{16}$$

$$\mathbf{B}_{n+1} = \mathbf{R}_{n+1} + \mathbf{H}\mathbf{P}_{n+1/n}\mathbf{H}^T \tag{17}$$

$$\mathbf{P}_{n+1} = (\mathbf{I} - \mathbf{K}_{n+1}\mathbf{H})\mathbf{P}_{n+1/n} \tag{18}$$

$$\underline{\nu}_{n+1} = \underline{z}_{n+1} - \mathbf{H}\underline{\hat{x}}_{n+1/n} \tag{19}$$

where $\mathbf{\Phi}$ is the assumed target motion model of eqn. 1, $\mathbf{K}_{n+1}$ is the filter gain, $\underline{\nu}_{n+1}$ the innovations, $\mathbf{H}$ is the measurement matrix, $\mathbf{P}_{n+1}$ is the state covariance matrix, $\mathbf{B}_{n+1}$ is the covariance of the innovations, $\mathbf{\Gamma}$ is the excitation matrix, $\mathbf{R}_n+_1$ the assumed measurement noise covariance matrix, and $\mathbf{Q}_n$ the filter acceleration noise matrix. The measurement noise matrix $\mathbf{R}_n$ and filter acceleration noise matrix $\mathbf{Q}_n$ are parameters which must be estimated in a practical situation. Clearly when $\mathbf{R}_n$ is not diagonal, the recursive updating of the innovation covariance matrix $\mathbf{B}_{n+1}$ and the state covariance matrices $\mathbf{P}_{n+1}$ and $\mathbf{P}_{n+1/n}$ using the above equations introduces off-diagonal terms. When the measurement errors in each co-ordinate are independent that is $\mathbf{R}_n$ is diagonal, the Kalman filter may be decoupled into two optimal tracking filters, known as $\alpha - \beta$ filters [20]. This filter configuration simplifies the computational requirements considerably, because the states relating to each of the two co-ordinates can be estimated independently. The equations for an $\alpha - \beta$ filter to estimate x-position and velocity are;

$$\hat{x}_{n+1} = x_{n+1/n} + \alpha_{n+1}(z_{x_{n+1}} - x_{n+1/n}) \tag{20}$$

$$\hat{\dot{x}}_{n+1} = \dot{x}_{n+1/n} + (\beta_{n+1}/\Delta t)(z_{x_{n+1}} - x_{n+1/n}) \tag{21}$$

Where $x_{n+1/n}$ and $\dot{x}_{n+1/n}$ are the predicted position and velocity respectively, that is $\underline{x}_{n+1/n} = \mathbf{\Phi}\,\hat{\underline{x}}_n$ for the full Kalman filter, and $z_{x_{n+1}}$ is the x-component of the measurement vector $\underline{z}_{n+1}$. The Kalman gain for the $\alpha - \beta$ filter is;

$$\mathbf{K}_{n+1} = (\alpha_{n+1} \quad \beta_{n+1}/\Delta t)^T \tag{22}$$

where $\alpha_{n+1}$ and $\beta_{n+1}$ are the gain coefficients. The estimation error covariance can be shown to be given by

$$\mathbf{P}_{n+1} = (\sigma_x^2)_{n+1} \begin{bmatrix} \alpha_{n+1} & \beta_{n+1}/\Delta t \\ \beta_{n+1}/\Delta t & \delta_{n+1}/\Delta t^2 \end{bmatrix} \tag{23}$$

Where $(\sigma_x^2)_{n+1}$ is the variance in the error of the $(n+1)^{th}$ x-position measurement. Recurrence relations for $\alpha$, $\beta$ and $\delta$ can easily be obtained using equations 13 to 19.

## 3. Track Splitting Algorithm

Tracking of a single target, in the ideal situation where one noisy measurement is obtained at each radar scan, can be achieved using standard Kalman filter techniques. In the multi-target case, an unknown number of measurements are received at each radar scan and, assuming no false measurements, each measurement has to be associated with an existing or new target tracking filter. The standard approach for associating a measurement with an already established track is to form an ellipse or gate, around the predicted position measurement [16] and if a measurement falls inside the ellipse it is normally used to update the track. When targets are near to each other, then more than one measurement may fall within the prediction ellipse of a filter. For example, if n measurements occur inside a prediction ellipse, then the filter branches or splits into n tracking filters. This approach is known as the track splitting algorithm [16]. The multiple target tracking algorithm using a track splitting filter consists of the following three basic modules, track initialization, track continuation and the track pruning as a means of controlling the explosion in the number of tracks due to measurement ambiguity [21].

### 3.1 Track Initialization

One of the basic requirements for any filtering algorithm is satisfactory "initialization" of the filter that is providing the initial estimate vector and the initial covariance matrix of the estimate vector. The filter initialization becomes more important in the multi-target environment where additional algorithms such as those for data association are involved. The measurement acceptance test, which is an essential part of the algorithm, is also affected by a poor guess for the initialization of the covariance matrix of the estimate. The initial state is usually assumed to be a normally distributed random variable [17] that is;

$$\underline{x}_0 \sim N[\underline{\hat{x}}_{0/0}, \mathbf{P}_{0/0}] \tag{24}$$

The scheme for track initialization is quite simple. A measurement m is considered to be unused if it has not been used to update a track. This measurement is then stored for possible correlation with another measurement n arriving at a later scan, usually the next scan, and if the actual distance between these two measurements m and n is less than a distance threshold $\delta_d(m,n)$ a new track is initialized. A maximum speed $V_{max}$ is assumed for a target, thus the maximum distance that can be traveled in j time step is $V_{max}\ j\ \Delta t$, *where* $\Delta t$ is the time interval between two scans. The distance measurement noise variance $\sigma_T^2(m,k)$ of measurement m at time k is obtained from the measurement noise variances on the (x, y) coordinates as;

$$\sigma_T^2(m,k) = \sigma_x^2(m,k) + \sigma_y^2(m,k) \tag{25}$$

The distance threshold $\delta_d(m,n)$ between measurements m and n, j time step apart is therefore taken as;

$$\delta_d(m,k) = V_{max}\ j\ \Delta t + \sigma_T(m,k) + \sigma_T(n,k+j) \tag{26}$$

When the above test is satisfied the most recent measurement is taken as the initial track position estimate. The initial velocity of the track is taken by dividing the difference between the two measurements m and n with the time elapsed between the two scans. Thus the x co-ordinate velocity estimate is given by;

$$\hat{\dot{x}}_0 = \frac{z_x(k+j,\ n)\ -\ z_x(k,\ m)}{j\ \Delta t}. \tag{27}$$

The initial velocity estimate for the y co-ordinate can be similarly derived. The choice of the covariance matrix of the estimate should be such that the expected value of the estimation errors achieved by the filter match the filter calculated covariance that is;

$$E[(x_0 - \hat{x}_{0/0})(x_0 - \hat{x}_{0/0})^T] = \mathbf{P}_{0/0} \tag{28}$$

The Kalman gain or the weighting given to the predicted estimate is directly proportional to the covariance of the estimate. This means that an optimistic (very accurate) covariance at the initial stage will produce a low gain with the result that a small weighting is given to incoming measurements, which normally results in large errors in the initial estimates. At the other extreme a very high value of gain may have a bad effect on tracking accuracy since a high weighting is placed on the noisy measurements. The covariance of the measurement noise $\mathbf{R}_n$ can be taken as an initial uncertainty of the target position, thus, supposing the initial covariance matrix for a two dimensional Kalman filter is

$$\mathbf{P}_{0/0} = \begin{bmatrix} \sigma_x^2 \begin{pmatrix} \alpha & \beta/\Delta t \\ \beta/\Delta t & \delta/\Delta t^2 \end{pmatrix} & \sigma_{xy} \begin{pmatrix} \alpha & \beta/\Delta t \\ \beta/\Delta t & \delta/\Delta t^2 \end{pmatrix} \\ \sigma_{xy} \begin{pmatrix} \alpha & \beta/\Delta t \\ \beta/\Delta t & \delta/\Delta t^2 \end{pmatrix} & \sigma_y^2 \begin{pmatrix} \alpha & \beta/\Delta t \\ \beta/\Delta t & \delta/\Delta t^2 \end{pmatrix} \end{bmatrix} \tag{29}$$

where $\alpha = 1$, $\beta = 1$, and $\delta = 2$ are the gain coefficients [22]. For a decoupled $\alpha - \beta$ filter it follows that the above covariance matrix becomes the block diagonal matrix.

### 3.2 Track Continuation

As shown in figure 1, following the initial track formation incoming observations are considered for the continuation of existing tracks. The continuation procedure consists of prediction, measurement association and state estimation (i.e. updating). At each radar scan, the target position is predicted using eq. 13 and the uncertainty associated (eq. 19) with this is used to place a measurement acceptance ellipse around the predicted position. If the dynamics of the assumed target model is correct then each measurement from that particular target will fall inside the predicted ellipse (measurement acceptance ellipse). However, at times incorrect measurements (that are not original returns from that particular target) may have been used to update the target. In such case the target dynamic does not remain correct and true measurement may fall outside the prediction ellipse. On the other hand, when targets are very close together, more than one measurement may fall within the prediction ellipse of a particular target. Therefore, one has to resolve such situations through various data association techniques [17, 18]. The track splitting filter algorithm is one such technique in which the filter is allowed to split into the total number of measurements inside the ellipse [16]. This approach assumes that all the measurements falling inside the ellipse are equally probable for that particular target; therefore, all of them are used to update its state. Once the filter determines that a measurement has fallen inside its prediction ellipse, it uses a measurement acceptance test and if the test is satisfied then that particular measurement is used for update. The measurement acceptance criterion uses a simple test i.e., if the dimension of the measurement vector $\mathbf{Z}_n$ is M, then the norm $\mathbf{d_n^2}$ of the innovation vector $\underline{\nu}_n$ at scan n for a filter is given by;

$$\mathbf{d_n^2} = \underline{\nu}_\mathbf{n}^\mathbf{T} \mathbf{B_n^{-1}} \underline{\nu}_\mathbf{n} \tag{30}$$

where the M-dimensional Gaussian probability density for the innovation is;

$$f(\underline{\nu}) = \frac{e^{\frac{-d^2}{2}}}{(2\pi)^{\frac{M}{2}} \sqrt{|\mathbf{B_n}|}} \tag{31}$$

with $\mathbf{B_n}$ being the innovations covariance matrix for the specific filter and $|\mathbf{B_n}|$ its determinant. Provided that the filter model for the track dynamics is accurate and that all the measurements used to update the track did indeed originate from one particular target, the quantity $\mathbf{d_n^2}$ is a sum of squares of M-independent zero mean and unit standard deviation Gaussian random variables. Thus $\mathbf{d_n^2}$ will have a $\chi^2$ distribution with M degrees of freedom. The measurement acceptance criterion for a track is thus defined that if $\mathbf{d_n^2}$ is less than a threshold $\mathbf{J}^2$ (with some known probability) then that particular measurement at scan n can be used for update [22].

### 3.3 Support Function Criterion

A mechanism for restricting the excess tracks that originate from track splitting under measurement ambiguity is needed and one possibility is the use of the track support function. The likelihood function of a track is the measure of the probability of the track accepting a sequence of measurements in n scans. It is given by [16];

$$\Lambda = \prod_{i=1}^{n} f(\underline{\nu}_i) \tag{32}$$

$$= \left( \prod_{i=1}^{n} \frac{1}{(2\pi)^{M/2}\sqrt{|\mathbf{B}|}} \right) \left( e^{-\frac{1}{2}\sum_{i=1}^{n} d_i^2} \right) \tag{33}$$

The natural logarithm of the second part of the above equation is called the modified log-likelihood function (or support function $S_n$) [17]. The support function $S_n$ is given by;

$$S_n = -\frac{1}{2}\sum_{i=1}^{n} d_i^2 \tag{34}$$

and it can be calculated recursively from

$$S_{n+1} = S_n - \frac{1}{2}\underline{\nu}_{n+1}^T \mathbf{B}_{n+1}^{-1} \underline{\nu}_{n+1} \tag{35}$$

If the support function of a track is smaller than a threshold value, it may not represent a true target in the sense that the measurements it has been using are inconsistent with the assumed target motion. This is used as a criterion in the track-splitting algorithm to terminate a track. The summation of the norm $\mathbf{d_n^2}$ for n scans is $\chi^2$ distributed with N degrees of freedom where $N = n \times M$. Therefore the support function $S_n$ is also $\chi^2$ distributed with N degrees of freedom. If we wish to define a threshold T for a $\chi^2$ distribution so that the probability of the variable exceeding the threshold is $P_{rT}$ then we can either use $\chi^2$ tables or, if the degree of freedom is large, use the approximate relationship [23];

$$T = N + T_g\sqrt{(2N)} \tag{36}$$

which relates the equivalent threshold $T_g$ for a Gaussian distribution of unit variance to that of T for the $\chi^2$ distribution. For example if $N = 30$, $T_g = 2.327$ for a threshold probability of $P_{rT}$ equal to 0.01, and the above formula yields $T = 48.025$ compared with the value 50.892

from the $\chi^2$ table. For the implementation of the support function criterion, assuming that $M = 2$ and thus $N = 2 \times n$, then $T_n$ the threshold for the support function $S_n$ is given by the following relationship.

$$T_n^{'} = -(n + T_g\sqrt{(n)}) \tag{37}$$

### 3.4 Similarity Criterion

To further reduce the number of filters obtained when using the track-splitting algorithm the similarity criterion is used to ensure that no more than one filter is tracking the same target [24, 25]. The similarity criterion provides a measure for the nearness of two tracking filter position estimates and if this is below a certain threshold one filter can be eliminated. Any two tracks i and j are deemed to be similar if

$$(\underline{\hat{x}}_i - \underline{\hat{x}}_j)^T \mathbf{P}^{-1} (\underline{\hat{x}}_i - \underline{\hat{x}}_j) \leq D_{th} \tag{38}$$

where $\underline{\hat{x}}_i$ and $\underline{\hat{x}}_j$ are the two filter state estimate vectors. **P** is a weighting matrix chosen to be the sum of the covariance matrices of the two filters with off diagonal elements set to zero and $D_{th}$ is the chosen threshold. As the estimates $\underline{\hat{x}}_i$ and $\underline{\hat{x}}_j$ are correlated **P** will not be the true covariance of $(\underline{\hat{x}}_i - \underline{\hat{x}}_j)$ but in many instances a reasonable estimate. When the two tracks are similar, one obviously wishes to keep the best supported track. The support function of track is proportional to the life of the track; therefore if two tracks with different track life are compared then the track with a longer life may incorrectly be eliminated. To prevent such a situation the support function of a track is normalized where the normalized support function, is [26]

$$S_{normalize} = -\frac{S_n + n}{\sqrt{n}} \tag{39}$$

Similarity pruning as described above may lead to a different elimination of tracks in a multi-target scenario, depending upon the order in which similarity calculations between tracks are made.

### 3.5 Clustering Scheme

Here we discuss some additional modifications using clustering methods to reduce the number of similarity calculations so that further speed up can be achieved. If targets are far from each other then only one measurement will be accepted by each tracking filter, assuming that there are no false measurements. When track splitting occurs it means that at least two targets are close to each other. At subsequent scans a tree is formed for each target. Only one branch of this tree corresponds to the real target, the others are false tracks and they will usually be eliminated by either the similarity or support function tests.

The similarity test checks the nearness of two tracks so that non-interfering trees cannot be similar to each other. Therefore, only tracks which belong to interfering trees need to be compared to detect similar tracks. Interfering trees can be detected as follows. When a track is initiated it is given a unique number as its identity and when a track splits its identity is passed to all new branches. If any two branches from two different trees have been formed by accepting the same measurements, then it means these trees are interfering with each other. Therefore, the clustering algorithm first finds the branches which have used the same measurements, which are called interfering branches. These branches are placed in different groups (i.e. tracks which have used measurement-0 will be in the first group, those which have used measurement-1 will be in the second group and so on). Then the identities of the tracks in different groups are checked. If there are two tracks in two different groups with the same identity these two groups are merged to form a cluster. This procedure is known as clustering algorithm A. A somewhat similar approach has been investigated in reference [27] to form clusters and distributing them on parallel processing architecture. Now if only the interfering branches are considered instead of interfering trees, then each group becomes a cluster. This approach has been used by Reid [28] to implement his multiple hypotheses tracking algorithm for multiple target tracking. This clustering method has also been implemented and is called clustering algorithm B, results of these two algorithms are discussed in the next section.

## 4. Analysis

As mentioned earlier, when the track splitting filter algorithm is used, the tracking filter splits into branches which are equal to the number of measurements found within the predicted acceptance ellipse. This means the shape of the measurement ellipse is very important in the case of neighboring or crossing targets. A four crossing target scenario is considered for our investigation as shown in figure 2 for low and high correlation factors of 0.1 and 0.9 respectively. These four targets are moving from a fixed location with same velocity and they cross each other after 30 seconds. The correlation factor is approximately kept constant throughout the run (100 Seconds) by maintaining the relative positions of the target and the platform (sensor is on-board a ship). To obtain the two correlation factors only the initial position of the platform is changed and all other parameters remain the same. The scenario was run with ten different random seeds; table 1 gives the number of average tracks present for the two filters with low and high correlation factors. As anticipated, because of the inferior measurement prediction ellipse, $\alpha - \beta$ filter has considerably more branching near the crossing point (30th seconds) for the high correlation factor. For low correlation the numbers of branches are almost the same. The fact of the matter is, this increased branching requires extra overhead computation for data association and track maintenance. The clustering schemes Algorithm A and B for similarity criterion produced relatively low number of branches because of the more structured and intelligent techniques as shown in table 1. We selected a number of similar scenarios to compare the speed-up between the two filters, the computation ratio for Kalman vis a v $\alpha - \beta$ filter is in the order of 1 to 7 approximately.

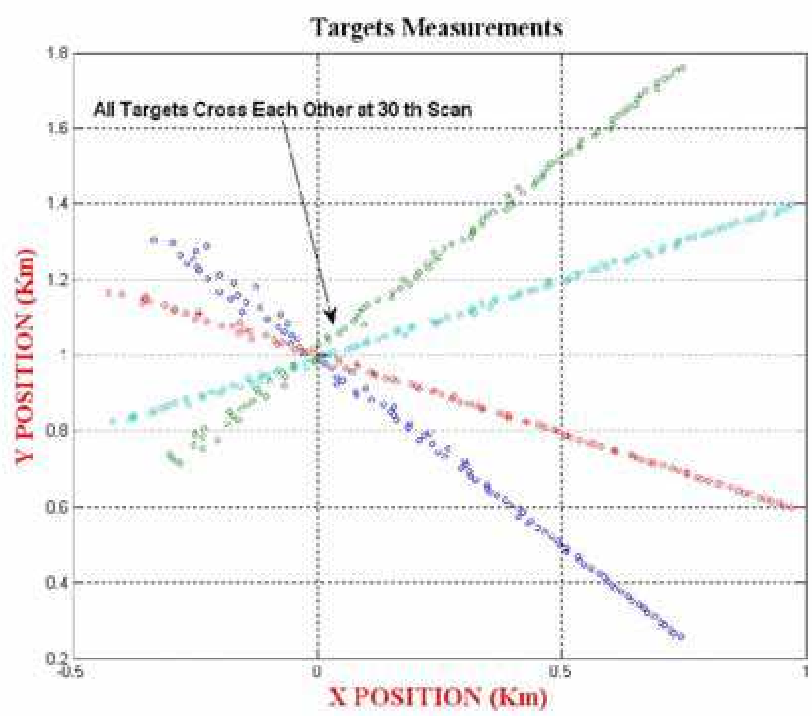


Fig. 2. Crossing Target Scenario

| | CORRELATION | | | | | |
|---|---|---|---|---|---|---|
| | Kalman Filter | | $\alpha - \beta$ Filter | | Clustering | |
| Scan No | Low (0.1) | High (0.9) | Low (0.1) | High (0.9) | A | B |
| 22 | 4 | 4 | 4 | 4 | 4 | 4 |
| 23 | 4 | 4 | 4 | 4 | 4 | 5 |
| 24 | 4 | 4 | 4 | 8 | 6 | 6 |
| 25 | 4 | 5 | 5 | 10 | 7 | 7 |
| 26 | 5 | 10 | 6 | 30 | 27 | 29 |
| 27 | 7 | 24 | 7 | 40 | 32 | 35 |
| 28 | 12 | 29 | 13 | 51 | 44 | 47 |
| 29 | 15 | 32 | 18 | 45 | 38 | 40 |
| 30 | 16 | 28 | 19 | 42 | 38 | 38 |
| 31 | 16 | 26 | 20 | 42 | 36 | 40 |
| 32 | 14 | 17 | 19 | 35 | 30 | 31 |
| 33 | 13 | 12 | 18 | 30 | 27 | 29 |
| 34 | 10 | 9 | 15 | 22 | 19 | 19 |
| 35 | 8 | 5 | 10 | 15 | 12 | 11 |
| 36 | 6 | 4 | 7 | 13 | 10 | 11 |
| 37 | 4 | 4 | 6 | 12 | 8 | 8 |
| 38 | 4 | 4 | 5 | 9 | 9 | 8 |
| 39 | 4 | 4 | 4 | 7 | 7 | 7 |
| 40 | 4 | 4 | 4 | 4 | 4 | 4 |

Table 1. Average Number of Tracks

Figure 3 shows the average speed-up plotted against various numbers of targets. It can be seen from figure 3 that as the ambiguity increases, the speed-up deteriorates to a ratio of 1 to 3. Therefore, the advantage of using an $\alpha - \beta$ filter under high correlation is not really great. One of the main reasons for excessive branching, in the case of $\alpha - \beta$ filter under high correlation, is due to the shape of the prediction ellipse as shown in figure 4 which is almost like a circle around the predicted position of the track. However, the shape of prediction ellipse in case of Kalman filter is like a true ellipse aligned in the direction of the target heading. For our second part of investigation the scenario geometry was modified and the measurement data was generated corresponding to one of the target before crossing (30th seconds) and the second after the crossing as shown in figure 5, duration of the tracking is 100 seconds. In this investigation we want to find how effective is the support function criterion for the two filters when correlation is high. Tables 2 and 3 show the initial angles when the target starts its motion and the intersection angles when it changes its direction. The correlation ranges during tracking period for these angles are also shown. The idea is to feed different kinds of data to analyze filter's behavior. Both filters Kalman and $\alpha - \beta$ filters were used to track these scenarios for the two values of correlation. Basically each run consists of 10 iterations and a new seed is selected for every iteration. The support function and the measurement acceptance values are obtained during these iterations and finally the average is computed.


Fig. 3. Kalman & Alpha-Beta Filter Speed-Up Comparison

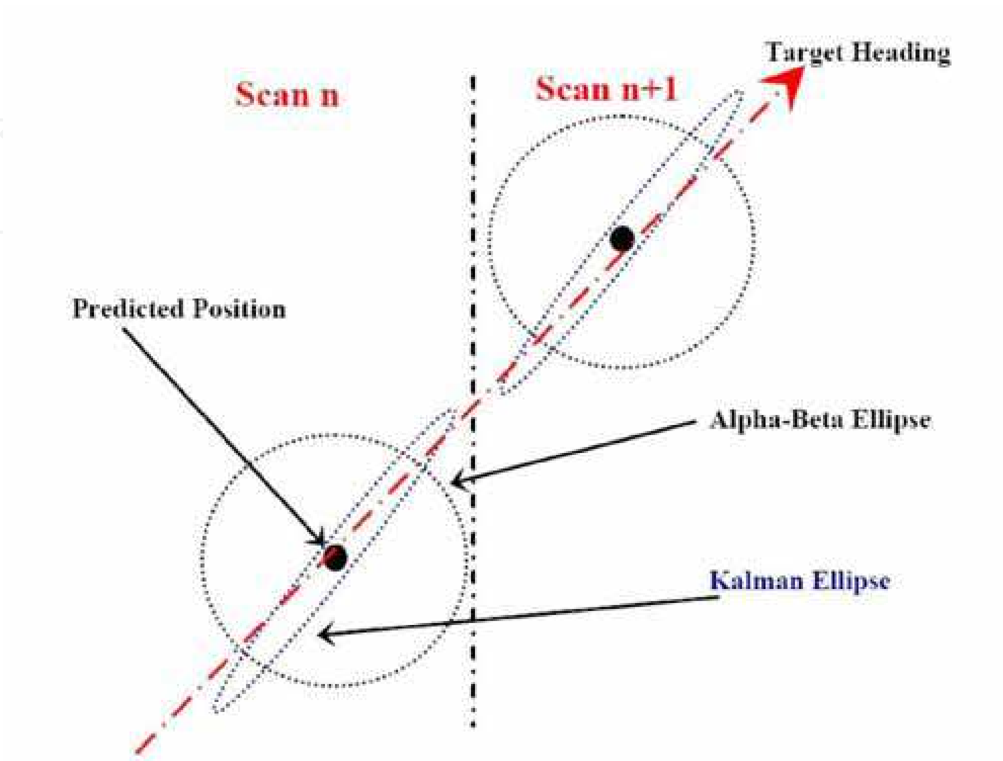


Fig. 4. Prediction Ellipse for Kalman & Alpha-Beta Filter

Figure 6 shows the value of these parameters with low correlation factors corresponding to table 2 and it can be seen that after initial track formation the target is following its path. This actually means that the measurement acceptance criterion remains less than a given probability threshold value, which in our case is 99 %. At the intersection point where a new target appears the track is lost, which should be the case by a tracking filter as the measurement from the second target will fall outside the prediction ellipse. However, in figure 7 where the correlation factor is high (table 3), the behavior of the two filters is totally different. Kalman filter is consistent by losing the track at the intersection point but $\alpha - \beta$ filter kept on tracking the target assuming it is the best supported track.

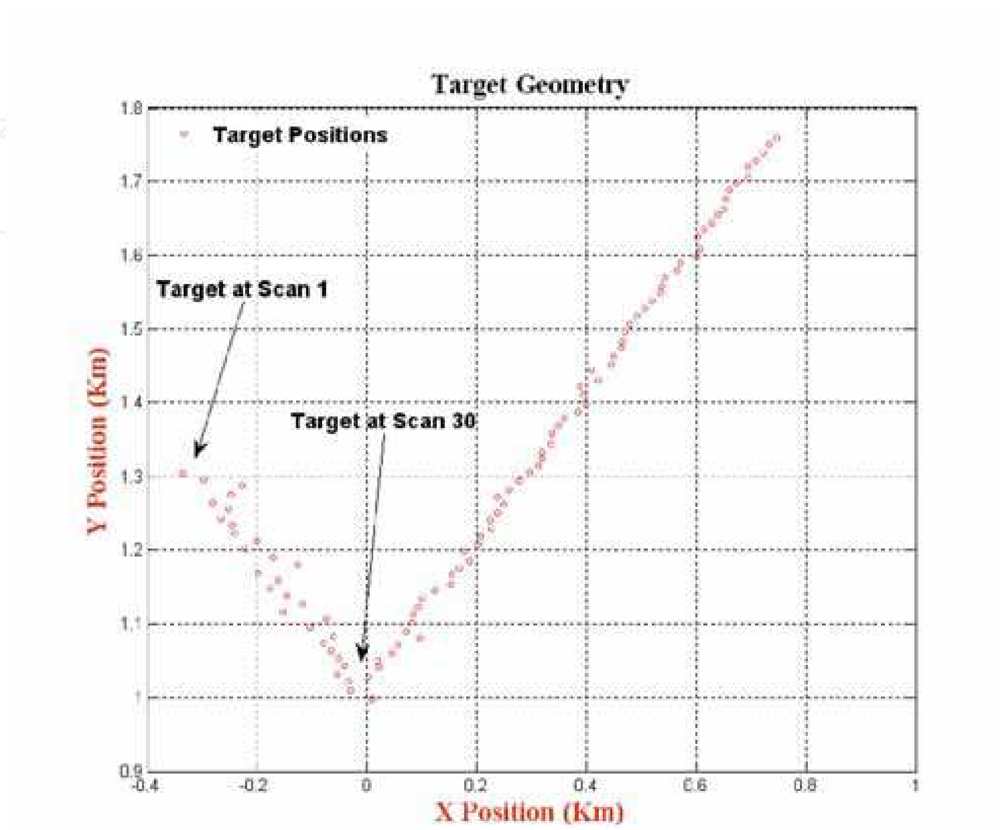


Fig. 5. Scenario Geometry

| Scan No | Initial Angle | Intersection Angle | Correlation Range | Average Correlation |
|---|---|---|---|---|
| 1 | 45° | 90° | 0.02 - 0.20 | 0.17 |
| 2 | 90° | 135° | 0.01 - 0.18 | 0.05 |
| 3 | 90° | 180° | 40.02 - 0.50 | 0.30 |
| 4 | 45° | 135° | 40.02 - 0.33 | 0.27 |

Table 2. Low Correlation

| Scan No | Initial Angle | Intersection Angle | Correlation Range | Average Correlation |
|---|---|---|---|---|
| 1 | 45° | 90° | 0.90 - 0.99 | 0.99 |
| 2 | 90° | 135° | 0.97 - 0.99 | 0.99 |
| 3 | 90° | 180° | 0.93 - 0.99 | 0.99 |
| 4 | 45° | 135° | 0.98 - 0.99 | 0.99 |

Table 3. High Correlation

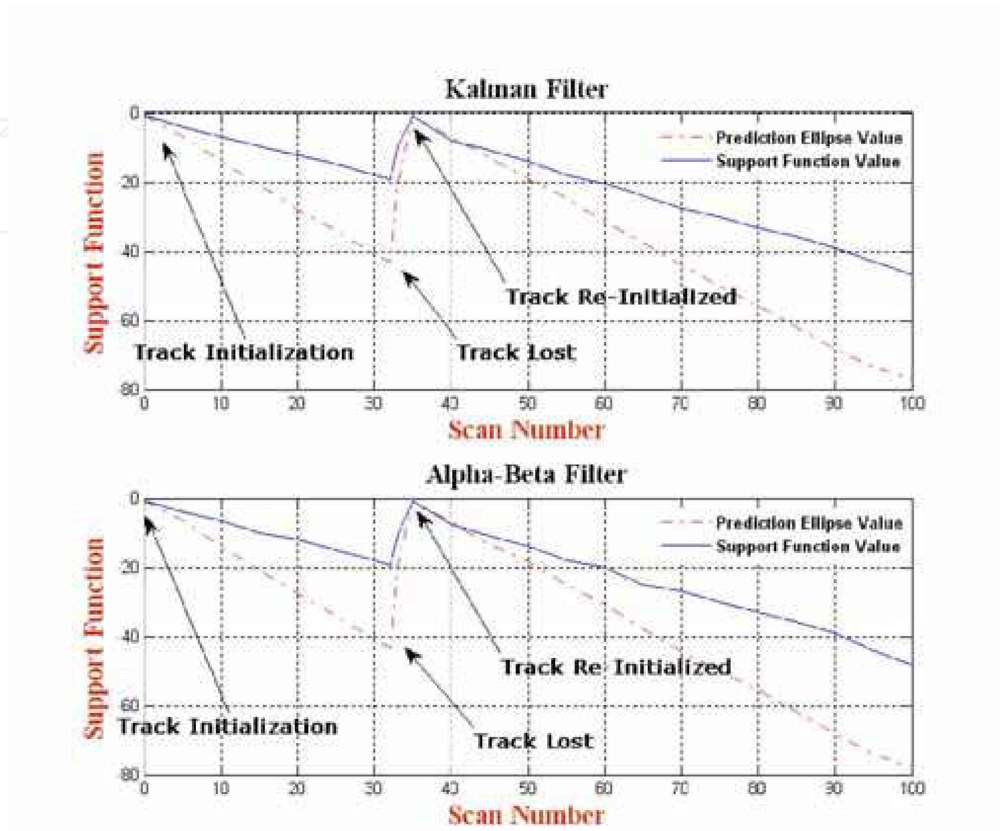


Fig. 6. Low Correlation Behaviour

## 5. Conclusion

In this paper we have compared the relative merits of the optimal Kalman filter with the sub-optimal $\alpha - \beta$ filter, we introduced tree clustering schemes for pruning the false target-measurement pairing to keep the excessive track explosion under control at the time of ambiguity. The interesting point for investigation is the amount of correlation during tracking period. Much investigation has been carried out in determining the position error accuracy of these two filters that reveals there is not much difference between the two estimates. However, one aspect which has not been given much attention in the past is the shape of the prediction ellipse under different correlation factors. It has been demonstrated by our investigation that in multi-target environments containing neighboring as well as crossing targets, more branching occur in the case of the $\alpha - \beta$ filter due to the shape of the prediction ellipse. It has been shown that in a high correlation scenario, the de-coupled $\alpha - \beta$ filter is more likely to accept unrealistic measurements compared with the Kalman filter. Therefore, the speed of computation when using an $\alpha - \beta$ filter in a multi-target scenario is not high as one would predict from single target considerations. In fact it was found to be only 3 to 4 times faster than a standard Kalman filter for crossing target scenarios containing up to 10 targets. The clustering techniques which are more structured and intelligent therefore, in the case of $\alpha - \beta$ filter less branching is observed, although numbers are not very significant however, the techniques have the premise to produce better results for such scenarios. Further, one important aspect must be kept in mind that number of branches depends on couple of factors, for example the time the target cross each other, their angle of intersection at the time of crossing and the order in which similarity criterion is executed. In future we would like to carry out our research work for more in-depth analyses of these two filters considering the results obtained in our investigation here plus with more realistic scenarios.

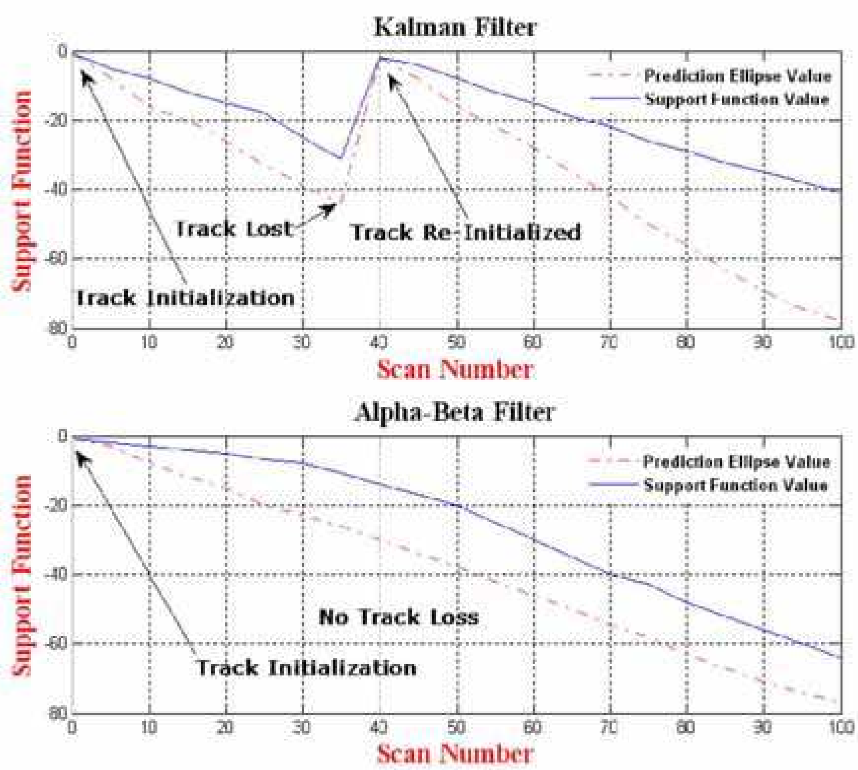


Fig. 7. High Correlation Behaviour

## 6. References


Glen W. Mabey and Jacob Gunther: A Robust Motion-Estimation Algorithm for Multiple-Target Tracking at Close Proximity Based on Hexagonal Partitioning, July 2003, Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance AVSS '03, Publisher: IEEE Computer Society.

David Hall: Lectures in Multi-sensor Data Fusion and Target Tracking, March 2001, Book, Publisher: Artech House, Inc.

Lawrence D. Stone, Thomas L. Corwin, and Carl A. Barlow: Bayesian Multiple Target Tracking, August 1999 Book Publisher: Artech House, Inc.

Peter Nillius, Josephine Sullivan, and Stefan Carlsson: Multi-Target Tracking - Linking Identities using Bayesian Network Inference, June 2006, Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 CVPR '06, Publisher: IEEE Computer Society.

Ahmet G. Pakfiliz and Murat Efe: Multi-Target Tracking in Clutter with Histogram Probabilistic Multi-Hypothesis Tracker, August 2005, Proceedings of the 18th International Conference on Systems Engineering ICSENG '05, Publisher: IEEE Computer Society.

Junji Satake and Takeshi Shakunaga: Multiple Target Tracking by Appearance-Based Condensation Tracker using Structure Information, August 2004, Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 3 - Volume 03 ICPR '04 , Publisher: IEEE Computer Society.

Hyong-suk Kim, Hong-rak Son, Young-jae Lim and Jae-chul Chung: Target Tracking via Region-Based Confidence Computation with the CNN-UM, December 2002, Proceedings of the Third IEEE Pacific Rim Conference on Multimedia: Advances in Multimedia Information Processing PCM '02, Publisher: Springer-Verlag.

Duncan Smith and Sameer Singh: Approaches to Multi-sensor Data Fusion in Target Tracking, A Survey, December 2006, IEEE Transactions on Knowledge and Data Engineering, Volume 18 Issue 12, Publisher: IEEE Educational Activities Department.

Bum-Jik Lee, Jin-Bae Park and Young-Hoon Joo: IMM Algorithm Using Intelligent Input Estimation for Maneuvering Target Tracking, May 2005, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Volume E88-A Issue 5 Publisher: Oxford University Press.

QingE Wu, Tuo Wang, YongXuan Huang and JiSheng Li: Application of Fuzzy Automata on Target Tracking, Proceedings of the Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007) Volume 02, Pages: 437441, 2007, ISBN:0-7695-2874-0.

David Choi and Benjamin Roy: A Generalized Kalman Filter for Fixed Point Approximation and Efficient Temporal-Difference Learning, April 2006 Discrete Event Dynamic Systems, Volume 16 Issue 2, Publisher: Kluwer Academic.

Jon Whittle and Johann Schumann: Automating the implementation of Kalman filter algorithms, December 2004, ACM Transactions on Mathematical Software (TOMS), Volume 30 Issue 4 Publisher: ACM Press.

R. Vasquez and J. Mayora: Estimation of motion and position of a rigid object using a sequence of images (tridimensional Kalman filter approach), April 1991, Proceedings of the 19th annual conference on Computer Science CSC '91 Publisher: ACM Press.

Jae-Chern Yoo and Young-Soo Kim: Alpha-beta-tracking index (?-?-?) tracking filter, January 2003, Signal Processing, Volume 83 Issue 1 Publisher: Elsevier North-Holland, Inc.

P. R. Kalata and K. M. Murphy: Alpha-Beta Target Tracking and Track Rate Variations, Proceedings of the 29th Southeastern Symposium on System Theory (SSST '97) SSST '97, Publisher: IEEE Computer Society.

P. L. Smith and G. Buechler: A branching algorithm for discriminating and tracking multiple objects, IEEE Transactions Automatic Control, Vol. AC-20, February 1975, pp 101-104.

Y. Bar-Shalom and T. E. Fortmann: Tracking and Data Association, Academic Press, Inc. 1988.

S. S. Blackman: Multiple Target Tracking with Radar Applications, Artech House, Inc. 1986.

A. Farina and F. A. Studer: Radar Data Processing Volume 1, Research Studies Press Ltd, 1986.

A. W. Bridgewater: Analysis of second and third order steady state tracking filters, AGARD conference proceedings No. 252, Moterey, CA, Oct. 1978, pp 9-1 to 9-11.

D. P. Atherton, E. Gul, A. Kountzeris and M. Kharbouch: Tracking Multiple Targets using parallel processing ", Proc. IEE, Part D, No. 4, July 1990, pp 225-234.

D. P. Atherton, D. M. Akbar Hussain and E. Gul: Target tracking using transputers as parallel processors, 9th IFAC symposium on identification and system parameter estimation, Budapest Hungary July 1991.

Abramowitz M and Stegun I. A. (Eds): A Handbook of Mathematical Functions, Dover N. Y. 1972.

D. M. Akbar Hussain, David Hicks, Daniel Ortiz-Arroyo, Shaiq A. Haq, Zaki Ahmed: A Case Study: Kalman and Alpha-Beta Computation under High Correlation, published in the proceedings of the IAENG International Conference on Software Engineering, Hong Kong 19 21 March, 2008.

D. M. Akbar Hussain, Zhenyu Yang, Shaiq A Haq, Zaki Ahmed, M. Zafar Ullah Khan: A Novel Technique to Avoid Similarity Criterion Calculations in a MultiProcessor Environment, Springer: Advances and Innovation in System, Computing Science and Software Engineering 2008.

Kharbouch M. M. and D. P. Atherton: A Transputer Implementation for Multiple Target Tracking, Microprocessors and Microsystems, April 1989, Pages: 188 - 194.

E. Gul: An Investigation of Target Tracking on Transputers, D. Phil thesis 1989, University of Sussex, Brighton U.K.

Reid, D. B.: An Algorithm for Tracking Multiple Targets, IEEE Transactions on Automatic Control, Vol. AC-24, pp 843 - 854, December 1979.


**Advances in Computer Science and IT**
Edited by D M Akbar Hussain

ISBN 978-953-7619-51-0
Hard cover, 420 pages
**Publisher** InTech
**Published online** 01, December, 2009
**Published in print edition** December, 2009

The book presents some very interesting and excellent articles for this divergent title. The 22 chapters presented here cover core topics of computer science such as visualization of large databases, security, ontology, user interface, graphs, object oriented software developments, and on the engineering side filtering, motion dynamics, adaptive fuzzy logic, and hyper static mechanical systems. It also covers topics which are combination of computer science and engineering such as meta computing, future mobiles, colour image analysis, relative representation and recognition, and neural networks. The book will serve a unique purpose through these multi-disciplined topics to share different but interesting views on each of these topics.

**How to reference**
In order to correctly reference this scholarly work, feel free to copy and paste the following:

D. M. Akbar Hussain and Zaki Ahmed (2009). Object Tracking under High Correlation for Kalman & α - β Filter, Advances in Computer Science and IT, D M Akbar Hussain (Ed.), ISBN: 978-953-7619-51-0, InTech, Available from: http://www.intechopen.com/books/advances-in-computer-science-and-it/object-tracking-under-high-correlation-for-kalman-and-alpha-beta-filter

INTECH
open science | open minds

**InTech Europe**
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

**InTech China**
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821