A Set Based Approach to Target Tracking in Clutter

F. K. Fletcher  D. J. Kershaw
Maritime Operations Division  Science Policy Division
Defence Science and Technology Organisation
PO Box 1500, Edinburgh SA 5111, Australia
Fiona.Fletcher@dsto.defence.gov.au  David.Kershaw@defence.gov.au

Abstract – In this paper, a set based estimation approach to target tracking in clutter is considered. To ensure a guaranteed bound on target location is maintained at all times, a track split approach using all feasible measurements is adopted. This approach suffers from an exponential growth in the number of feasible target location bounds over time, even with modest clutter densities. An overbounding procedure is introduced to group sets of bounds in a similar locale into a single bound. Above a clutter density threshold, this provides a more computationally efficient option while ensuring a guaranteed bound is maintained.

1 Introduction

When undertaking target tracking, it is essential for the estimate of target state to be accurate and robust. This is particularly true when tracking in clutter as any error may quite easily be propagated for a number of steps before being discovered. For many military applications, it is often more important to have a robust bound on target location rather than a possibly inaccurate point estimate (or distribution) of the target location. Forming a robust bound on target location when tracking in a cluttered environment is the focus of this paper.

Most conventional algorithms for target tracking in clutter are based on the Kalman Filter (KF) [1, 2] and are subject to the KF assumptions of known Gaussian distributions for noises and initial estimates and linear (or linearisable) models for target motion. The optimal Bayesian approach to tracking in clutter is to track split; that is, to update using each feasible association of measurement with track. The optimal Bayesian approach suffers from exponential growth in the number of tracks, and suboptimal approaches are used in practice. Pruning of tracks with low probability is often performed to maintain a more manageable number of tracks. Nearest neighbour (NN) and strongest neighbour (SN) associations may also be used, but neglect all feasible measurements other than the nearest and strongest respectively. A more advanced technique is that of the Probabilistic Data Association (PDA) Filter where the Gaussian mixture probability density function of the optimal track split is approximated by a single Gaussian distribution.

None of these algorithms will give a guaranteed bound on target location as the estimate is either a Gaussian distribution or a Gaussian mixture, which, although arbitrarily small, will have a positive target state probability density over all space. Gating may be applied to the track split approach to gain an approximate bound, but as soon as measurements are neglected (NN, SN, Pruning) or approximated by a single measurement (PDA), the bound obtained is inaccurate. The assumption of Gaussian distributions for noises and initial estimates also limits the applicability of these techniques.

A more modern approach to target tracking in clutter uses Particle Filter (PF) techniques [5, 6, 7]. These assume a model for target motion (linearity is not required) and the noises and initial estimate may have any distribution, but these distributions must be known. The set of feasible measurements are used to calculate the weights for each particle, and these weights are used to resample the particle distribution. The estimate is a distribution for the target state and, provided enough particles are used, an approximate bound on target location may be obtained by bounding the set of particles. The advantage of the PF is that it works for both linear and non-linear target motions. The disadvantage is the requirement to know the distribution of noises and initial estimates.

Set based estimation [3, 13, 14] is an alternative approach to tracking where estimates are a bound on target location. This bound is a guaranteed bound provided the target dynamics are linear and are modelled accurately and the noise processes satisfy a known bound. No assumptions other than boundedness are made about the distribution of the noise processes.

Saleem et al [12] use a set based track split approach for tracking in clutter. At each step, a guaranteed bound on target location is obtained, but as for the Bayesian track split,
this approach suffers from exponential growth in the number of feasible tracks over time. A more complicated attempt at using a set based estimator for tracking in clutter is considered by Best et al [4]. They investigate the problem of tracking multiple targets in clutter and use an approach analogous to that used for Multiple Hypothesis Tracking. The weighting of an estimate is directly proportional to the size of the estimate after update. Pruning is used to avoid exponential growth of the number of estimates. However, the concept of pruning undermines our premise of a guaranteed bound on target location. Any estimate that may contain the true target state must be retained and propagated through to the next time step.

This paper is an initial investigation of the use of set based estimation for tracking in clutter. The set based track split algorithm is extended to include an additional step where sets of feasible estimates in a similar location are overbounded by an ellipsoid to give a single overbounded estimate. This has the potential to form a robust bound on target location without the need to exhaustively compute all possible target tracks.

The aim of this paper is to provide an approach for reducing the computational load of calculating robust bounds for target location whilst still providing a satisfactorily tight overbound. Comparing results between set based and conventional KF based approaches is not attempted due to the difficulty in specifying appropriate performance measures for such a comparison (the set based approach in this paper guarantees a bound on target location provided the target motion model is correct, conventional Kalman Filtering does not). The Particle Filter is a potential candidate that may be used for comparison, provided the particle distributions have a tight bound. Such a comparison is beyond the scope of the current paper but will be investigated further at a later stage.

Section 2 gives a brief overview of track splitting using set theoretic estimation, and results are presented for a one-dimensional example. Section 3 discusses a modification to this algorithm to improve computational efficiency. Results are presented and compared with the basic track split approach in Section 4. Section 5 includes a discussion on higher dimensionality, and Section 6 concludes the paper.

2 Track Split Approach for Set Based Tracking

We assume the target trajectory and measurement models satisfy the system

\[ x_{k+1} = F_k x_k + v_k \]  
\[ z_k = H_k x_k + w_k \]

where \( x_k \) and \( z_k \) are target states and measurements respectively, and \( F_k \) describes the transition between target states and \( H_k \) converts target states to measurements. The process and measurement noises, \( v_k \) and \( w_k \) respectively, have bounds \( v_k \in \Omega_Q(k) \) and \( w_k \in \Omega_R(k) \) on the values they may take. An initial set estimate of the target state, \( \Omega(0|0) \), is assumed known (this may be the entire space).

2.1 General Algorithm

Assume that at some time \( T_k \), there are \( N_k \) a posteriori estimates of the target state, \( \Omega'(k|k), i = 1, ..., N_k \). Each of these estimates is predicted forward to give a priori estimates of the target state at time \( T_{k+1} \).

\[ \Omega'(k+1|k) = \Omega'(k+1|k) \cap \Omega_R(k+1), \]

for \( i = 1, ..., N_k \), where the \( \cap \) above denotes a Minkoski sum. In this equation, \( \Omega_R(k+1) \) is the projection of \( \Omega'(k|k) \) to time \( T_{k+1} \).

Assume \( M \) measurements have been received. Each tracker is then updated with any feasible measurements

\[ \tilde{\Omega}'(k+1|k+1) = \Omega'(k+1|k) \cup \Omega(l), \]

for \( i = 1, ..., N_k \) and \( j = 1, ..., M \) and \( l = 1, ..., N_{k+1} \), where \( N_{k+1} \leq MN_k \) and \( \Omega(l)(k+1) \) is the measurement error bound constructed around the \( j \)th measurement.

2.2 Ellipsoidal Bounds

For computational ease we use ellipsoidal bounds throughout the remainder of this paper. An ellipsoid is denoted by \( \varepsilon(a, A) \), where \( a \) is the centre of the ellipsoid and the matrix \( A \) describes its shape.

\[ \varepsilon(a, A) = \{ x : (x - a)'A^{-1}(x - a) \leq 1 \}. \]

The process and measurement noises are bounded by ellipsoids, \( v_k \in \varepsilon(0, Q_k) \) and \( w_k \in \varepsilon(0, R_k) \) respectively, as is the initial state, \( \Omega(0|0) = \varepsilon(\tilde{x}_0, \tilde{P}_0) \).

The prediction phase follows equation (3) to generate \( \Omega'(k+1|k) \), however, since in general this is not an ellipsoid, it is overbounded by an ellipsoid

\[ \tilde{\Omega}'(k+1|k) = \varepsilon(\tilde{m}'(p), \tilde{P}'(p)) \supseteq \Omega'(k+1|k). \]

The centre, \( m'(p) \), and matrix, \( P'(p) \), defining the ellipse are functions of a free parameter \( p \) which may be used to optimise some aspect of the ellipsoid, for example minimising the volume [8, 9].

The ellipsoid prediction, \( \tilde{\Omega}'(k+1|k) \), is then used in the update equation (7) instead of \( \Omega'(k+1|k) \). Again, the result, \( \tilde{\Omega}'(k+1|k+1) \), is not an ellipsoid but may be overbounded by one

\[ \tilde{\Omega}'(k+1|k+1) = \varepsilon(\tilde{m}'(q), \tilde{P}'(q)) \supseteq \Omega'(k+1|k+1). \]

The centre, \( m'(q) \), and matrix, \( P'(q) \), are functions of the free parameter \( q \) which may be used to minimise the volume of the ellipsoid [8, 9].
posed for this overbounding is described in the next section. Provide a hard bound on target location. The process proceeds because the only requirement of the tracking process is to overbound the space. Rather than using all of these sets individually in the tracking process, instead it is possible to overbound the target obeys an approximately constant velocity motion state is shown by an asterisk (*)..

Figure 1 shows all feasible estimates generated using the track split approach after 20 steps. Recall that the estimates are sets, represented in this case by ellipses. The true target state is shown by an asterisk (*).

There are 512 feasible ellipses illustrated in this figure. Observe that all of these are overlapping in the same region of space. Rather than using all of these sets individually in the tracking process, instead it is possible to overbound the set of overlapping ellipses at each stage, and use this overbound as the estimate for the next stage. This is possible because the only requirement of the tracking process is to provide a hard bound on target location. The process proposed for this overbounding is described in the next section.

\[ a_j \in \varepsilon(b, B) \supseteq \varepsilon(a_i, A_i) \oplus \varepsilon(0, A_j) \] (8)

The ellipse \( \varepsilon(b, B) \) is the minimum volume overbound of the vector sum of \( \varepsilon(a_i, A_i) \) and \( \varepsilon(0, A_j) \), as calculated for the prediction stage [8, 9]. Note that \( a_i = b \).

Although it is not strictly true that \( \varepsilon(a_i, A_i) \) and \( \varepsilon(a_j, A_j) \) intersect if this property is satisfied, they must be very close. To ensure strict intersection, we would require that \( a_j \in \varepsilon(a_i, A_i) \oplus \varepsilon(0, A_j) \), which is more difficult to ascertain.

We define an indicator function with entries

\[ \gamma_{ij} = \begin{cases} 1 & \text{if } a_j \in \varepsilon(b, B) \\ 0 & \text{otherwise} \end{cases} \] (9)

Clearly, \( \gamma_{ii} = 1 \) for all \( i \). This indicator function may be calculated for each \((i, j)\) pairing for \( j > i \), noting that symmetry may be used to imply the result for \( j < i \).

For each \( i \), the set of intersecting ellipses is \( S_i = \{ j : \gamma_{ij} = 1 \} \). Starting with \( i = 1 \), the first group is then \( \phi_1 = S_1. \) Then iterating through from \( i = 2, ..., N \), with \( l \) groups from the \((i - 1)^{th} \) iteration, if no elements of \( S_i \) are found in any existing group \( \phi_{j_1}, ..., \phi_{j_l} \), then \( S_i \) becomes a new group, \( \phi_{j_{l+1}} = S_i \). Otherwise, if elements of \( S_i \) are found in groups \( \phi_{j_{1}}, ..., \phi_{j_{l}} \), then these groups merge to form a single group, \( \phi_{j_{l+1}} = S_i \). Otherwise, if elements of \( S_i \) are found in groups \( \phi_{j_{1}}, ..., \phi_{j_{l}} \), then these groups merge to form a single group, \( \phi_{j_{l+1}} = S_i \). Otherwise, if elements of \( S_i \) are found in groups \( \phi_{j_{1}}, ..., \phi_{j_{l}} \), then these groups merge to form a single group, \( \phi_{j_{l+1}} = S_i \). Otherwise, if elements of \( S_i \) are found in groups \( \phi_{j_{1}}, ..., \phi_{j_{l}} \), then these groups merge to form a single group, \( \phi_{j_{l+1}} = S_i \). Otherwise, if elements of \( S_i \) are found in groups \( \phi_{j_{1}}, ..., \phi_{j_{l}} \), then these groups merge to form a single group, \( \phi_{j_{l+1}} = S_i \). Otherwise, if elements of \( S_i \) are found in groups \( \phi_{j_{1}}, ..., \phi_{j_{l}} \), then these groups merge to form a single group, \( \phi_{j_{l+1}} = S_i \). Otherwise, if elements of \( S_i \) are found in groups \( \phi_{j_{1}}, ..., \phi_{j_{l}} \), then these groups merge to form a single group, \( \phi_{j_{l+1}} = S_i \). Otherwise, if elements of \( S_i \) are found in groups \( \phi_{j_{1}}, ..., \phi_{j_{l}} \), then these groups merge to form a single group, \( \phi_{j_{l+1}} = S_i \).

Once the groups have been defined, it remains to overbound each group of ellipses by a single ellipse. Consider a group of ellipses \( \{ \varepsilon(c_i, C_i) \} \) which must be overbounded together. We require a minimum volume overbounding ellipse \( \varepsilon(p, P) \) for the union of the group of ellipses such that

\[ \varepsilon(p, P) \supseteq \bigcup_{i=1}^{n} \varepsilon(c_i, C_i) \] (10)

Although it is possible to find an overbounding ellipse for groups of ellipses, the minimum volume criterion makes
this is a very complicated problem, for which an analytical solution has not been found by the authors.

A numerical algorithm for overbounding a set of points by an ellipsoid was described by Pronzato and Walter [10, 11]. This algorithm may be used to generate the overbound provided a set of points describing the group of ellipses can be defined. Choosing points on the boundary of each ellipse is not sufficient as the overbound may not completely contain all of the ellipses and thus does not maintain the guaranteed bound on target location. Even using small increments over each ellipse boundary is not sufficient from this point of view, and is also undesirable because of the computational effort required.

Instead, the points should be outside the ellipses to maintain a guaranteed bound, and there should be as few points as possible for each ellipse to maintain computational efficiency. One way of doing this is to overbound each ellipse with a polygon, and use the vertices of the polygon in the numerical algorithm. An octagon is a good choice of polygon for this purpose as while it has only few (eight) vertices, it provides a relatively close outer bound for each ellipse, and may be defined easily because of its symmetry.

We consider an ellipse centred at the origin with major and minor radii respectively \( r_1 \) and \( r_2 \). The equation of the tangent line through this point can easily be constructed, and the vertices calculated as a function of \( a \), while it has only few (eight) vertices, it provides a relatively close outer bound for each ellipse, and may be defined easily because of its symmetry.

Asume there is a tangent at some parameterised point on the ellipse given by \((r_1 \cos a, r_2 \sin a)\). The equation of the tangent line through this point can easily be constructed, and the vertices calculated as a function of \( a \) as the intersection of this tangent with the already defined sides of the octagon, \( x = r_1 \) and \( y = r_2 \). A minimum perimeter criterion is used to choose the value of \( a \), it can easily be shown that \( a^* = \arctan \sqrt{r_2 / r_1} \). The vertices in the first quadrant are

\[
(v_1) = \left( r_1, \frac{r_2(1 - \cos a^*)}{\sin a^*} \right) \quad (11)
\]

\[
(v_2) = \left( \frac{r_1(1 - \sin a^*)}{\cos a^*}, r_2 \right) \quad (12)
\]

These may be mirrored through the \( x \) and \( y \) axes using the symmetry of the octagon and ellipse to obtain all eight vertices. For arbitrary ellipses, the reverse rotation and translation operations may be applied to the set of vertices to obtain an octagonal overbound for the original ellipse.

A set of eight vertices, \( V_i \), is obtained for each of the \( n \) ellipses in the group. The numerical ellipsoidal overbounded procedure is then applied to the set, \( V \), of all vertices in the group, \( V = \bigcup_{i=1}^{n} V_i \). This results in an ellipse that completely contains the entire group of ellipses. The set of these overbounding ellipses (one for each group) becomes the estimate, and is used for the prediction phase of the next iteration of the algorithm. This will be referred to as the Track Split with Overbounding (TSO) algorithm.

3.1 Variations on the Overbounding Procedure

Ad hoc variants of this algorithm may be used, two such variants are described here.

First, note that when there are only a small number of tracks, the track split approach is relatively efficient (this is demonstrated in the next section). Thus, we can define a threshold for the number of tracks. Below this threshold, the track split approach is used, but when the number of tracks exceeds the threshold, the overbound on intersect algorithm is employed. The results in the next section use a threshold of 5 tracks. This is referred to as the Track Split with Overbounding if \( n \) tracks (TSO > \( n \)) algorithm, where \( n \) is the threshold for the number of tracks.

A second variant employs a slight variation in the grouping stage of the algorithm. Note that by the construction of the grouping process, the TSO algorithm groups ellipses even if they are “close”. In this case, it would often be better to treat the ellipses separately. This variant changes the grouping method so that ellipses are grouped together if \( a_i \in \varepsilon(b, B) \supseteq \varepsilon(a_i, A_j) \) and \( 0 < c < 1 \) is the “overlap threshold”. We now define some

\[
\eta_{ij} = \begin{cases} 1 & \text{if } a_i \in \varepsilon(b, B) \supseteq \varepsilon(a_i, A_j) \\ 0 & \text{otherwise} \end{cases}
\]

(13)

in a similar manner to \( \gamma_{ij} \). Note that the problem is no longer symmetric so that \( \eta_{ij} \) must be calculated for each \((i, j)\) pairing. Any \((i, j)\) pairing such that \( \eta_{ij} = 0 \) will be grouped together (even if \( \eta_{ji} = 1 \)). The grouping algorithm described in the previous section is simply the specific case of this algorithm with \( c = 1 \). This variant will be referred to as the Track Split with Overbounding - Overlap Threshold \( c \) (TSO-OT) algorithm. In the results, an overlap threshold of \( c = 1/3 \) is used.

4 Results

The example used in Section 2.3 has been repeated using the overbounding procedures described in the previous section. The results are illustrated in Figure 2. The TSO and TSO-OT algorithms result in almost identical with a single estimate at the 20th iteration, depicted in a solid line. The TSO > 5 algorithm had four estimates at the 20th iteration, these are depicted by a dash-dot line. Note that the overbounded estimates do not completely contain the track split estimates. However, at the time when the overbounds

---

\footnote{The octagon bounding was suggested by Dr. Jane Perkins, Maritime Operations Division, DSTO}
were generated, they completely contained the track split estimates.

Consider a set of ellipsoidal estimates from a track split approach at time $k$ and assume they are overbounded by a single ellipse, the overbounded estimate. The prediction of the overbounded estimate is guaranteed to contain the exact prediction forward of these track split estimates to step $k + 1$ as in equation (3), but may not completely contain the ellipsoidal approximation of these estimates given in equation (6) (although there should not be much discrepancy at this stage). The real discrepancy occurs in the measurement update stage, where the intersection of the measurement error ellipses and the predictions are approximated by ellipses as in equation (7). In this stage, the overbound will completely contain the intersection of the measurement error ellipse with the exact prediction as in (4) (the real feasible region rather than an overbound), but is unlikely to completely contain the ellipse overbounds of the track split estimates. These errors will propagate through time, but both are guaranteed to contain the actual target state.

4.1 Monte Carlo Simulations

The same model parameters as used in Section 2.3 have been used for a Monte Carlo simulation to compare the performance of each of the algorithms: Track Split (TS), Track Split with Overbounding (TSO), Track Split with Overbounding if $> 5$ tracks ($\text{TSO} > 5$), and the Track Split with Overbounding with Overlap Threshold $\frac{1}{2}$ ($\text{TSO-OT} \frac{1}{2}$). The number of clutter points satisfied a Poisson distribution with clutter densities $\lambda = 0.001$ to 0.007 points per metre in steps of 0.001. These clutter points were uniformly distributed across the space, and a new clutter map generated at each iteration. For each clutter density, 200 Monte Carlo runs of 20 tracking steps were performed.

Figure 3 shows the mean CPU time required for each run of each algorithm. It can clearly be seen that for clutter densities above 0.004/m, the overbounding procedures significantly outperform the track split approach in terms of computational efficiency. All of the overbounding algorithms perform comparably in terms of computational efficiency.

Figure 4 shows the mean value of an upper bound on the area contained by the estimates after 20 steps for each algorithm and each clutter density. The upper bound is calculated by overbounding the entire set of estimates by a single ellipse in the same way as performed in the overbounding algorithm, overbounding each estimate by an octagon, and overbounding the set of all vertices by a single ellipse using the numerical algorithm. Note that this will not necessarily give an accurate estimate of the area contained by the estimates, especially where there are disjoint sets of estimates. However, it does give an indication of the relative size of the estimates between the different algorithms. As shown in this figure, the overbounding algorithms give larger estimate areas than the track split. The best performing overbounded algorithm is the $\text{TSO} > 5$ algorithm in regards to area.

Finally, Figure 5 gives an indication of the mean number of estimates after 20 iterations for each algorithm. The overbounding algorithms have an almost constant number of tracks after the 20 steps, while the track split approach shows the effect of exponential growth with clutter density (as expected).

These results show that the overbounded algorithms perform significantly better in terms of computational efficiency than the track split approach for increasing clutter density. The clutter density threshold, where the overbounded approach starts to computationally outperform the track split occurs at approximately 0.004/m. However, there is a trade-off between the area contained by estimates and the computational efficiency, as the overbounded estimates contain more area than the track split approach. It is recommended that the $\text{TSO} > 5$ algorithm be used as its computational efficiency is around the same as the other overbounding algorithms, with a significant decrease in the estimate area.

5 Discussion

The method described here may be extended to higher dimensions in a relatively straightforward manner. The grouping procedure may be used as it is, but the overbounding procedure must be adapted slightly. The octagonal overbounds used for ellipses must be generalised to polygons in higher dimensions, but the ideas used here may be adopted for higher dimensions.

It is worth noting that this algorithm is likely to be applicable only in the degenerate measurement case, that is where $\dim(z) < \dim(x)$ where $z$ and $x$ are used to denote a measurement and target state respectively. For the exam-
ple used here, where states are \([r, \dot{r}]\) and measurements are range only, the measurements do not give a direct indication of \(\dot{r}\). The measurement error region is an interval in the \(r\)-direction, but is projected through all \(\dot{r}\). Similarly, in a higher dimensional case, the corresponding components of measurement and estimate will have to match closely, but those components of the estimate that are not measured will be projected through all space.

In contrast, when \(\dim(z) = \dim(x)\), there is more chance of incorrect tracks becoming infeasible as the measurement and prediction must coincide relatively closely in all dimensions. The number of tracks should not blow out to the extent of the degenerate case, except maybe in the case of very high clutter density. This has not yet been explored due to the complexity of the intersection overbounding in the measurement update stage (7), but is worth further investigation.

The technique described in this paper may also be used for tracking multiple targets in clutter. The only difference would be that each set must be propagated forward using the relevant propagation model for every target that may be located in the set.

6 Conclusion

This paper has looked at the problem of using set based estimation to form a robust bound on target location when tracking in clutter. Initially, a simple track split approach was used, but it soon became evident that such an approach was not efficient, even for small clutter densities, as many estimates had significant overlapping regions. An overbounding step was introduced into the algorithm after the update stage to reduce the number of estimates, while maintaining the guaranteed bound on target location. The overbounding procedure works by grouping together estimates that are close, overbounding each estimate by an octagon, and using the set of all octagon vertices from a group in a numerical ellipsoidal overbounding algorithm [10, 11].

Monte Carlo simulation results suggest that there is a modest clutter density threshold, above which the overbounding algorithm presents a significantly more computationally efficient option than the straight track split approach. A particularly good implementation was to invoke the algorithm when the number of feasible estimates exceeded a threshold. However, there is a trade-off for the computational efficiency attained using this overbounding step, as the area contained by the estimate increases as a result of overbounding.

The algorithm presented here is specific to a one-dimensional target measurement and two-dimensional target state estimate (for example range and range-rate), but the ideas may be extended to higher dimensions.
References


