Reducing Computational Complexity of Gating Procedures Using Sorting Algorithms

Viet Duc Nguyen and Tim Clausen
Digital Signal Processing and System Theory
Christian-Albrechts-Universität zu Kiel
Kiel, Germany
Email: vng@tf.uni-kiel.de, tic@tf.uni-kiel.de

Abstract—Gating is an important part in many data association algorithms, specifically tracking algorithms. Its purpose is the preselection of suitable measurements or observations respectively in order to avoid unlikely measurement-to-track associations. As such, it can be essential for a low computational load. A very common gating method is individual ellipsoidal gating incorporating the Mahalanobis distance. On the one hand it is a reliable gating method used in many different tracking techniques, e.g., the global nearest neighbor method or the Multi-Hypothesis Tracking approach, but on the other hand it can be also very time consuming due to the inversions of covariance matrices related to all the measurement-to-track pairs.

In this paper a Gating-by-Sorting approach is proposed to accelerate the gating procedure significantly. It uses sorting algorithms to carry out coarse gating followed by a second ellipsoidal gating step. The gating approach is incorporated into a multi-hypothesis tracker and applied to a real radar dataset. Results are obtained by measuring the computational load within tracking processes. They show that the proposed gating algorithm outperforms classical ellipsoidal gating by orders of magnitude.

I. INTRODUCTION

Tracking systems are widely used in many applications, e.g., air traffic control, navigation radar, or anti-submarine-warfare. Depending on the sensors and on the environmental conditions, tracking systems have to face different challenges: Noisy measurements, missing contacts, or even phantom contacts. As such, tracking always involves data association problems in which numerous object-tracks have to be associated with numerous contacts correctly.

Implementations of well known tracking algorithms such as the global nearest neighbour (GNN) approach, the Multi-Hypothesis Tracking (MHT) approach, or the Joint Integrated Probabilistic Density Association (JIPDA) use gating to avoid unlikely contact-to-track associations [1]–[3]. The main idea of gating is to form a gate about the predicted measurement and only contacts that satisfy the gating condition, i.e. fall within the gate, are considered for further steps within the tracking process. So gating prevents unnecessary computation steps and saves computational load.

One very common gating method is ellipsoidal gating (EG). Its main principle is as follows:

Given a measurement prediction \( \hat{y}_i \) of the \( i \)th track and a measurement \( y_j \) of the contact \( j \), both values are afflicted with uncertainty due to prediction uncertainty and measurement noise, respectively. This uncertainty is represented by their residual covariance matrix \( S \). First, the Mahalanobis distance \( d_M \) is calculated as

\[
d_M^2 = \left(\hat{y}_i - y_j\right)^\top S^{-1} \left(\hat{y}_i - y_j\right). \tag{1}
\]

Only if \( d_M \) meets the requirement

\[
d_M^2 \leq G, \tag{2}
\]

with \( G \) being the gate size, the contact \( j \) is considered to fall within the gate. Compared to the Euclidean distance

\[
d_E^2 = \left(\hat{y}_i - y_j\right)^\top \left(\hat{y}_i - y_j\right), \tag{3}
\]

the Mahalanobis distance takes the residual covariance matrix into account. So as an advantage, EG considers the uncertainty within the gating process. Hence, it is a very common and reliable gating method which is used within many tracking-system implementations. As can be seen in (1), calculating the Mahalanobis distance requires a matrix inversion which is expensive in terms of computational load. Furthermore, as EG is used as individual gating, each track-to-contact or hypothesis-to-contact (MHT) combination has to be gated, i.e., \( m \cdot n \) individual gating processes and matrix inversions are to be conducted, with \( m \) being the number of tracks and \( n \) being the number of contacts. The number of gating processes remains large even if less computationally demanding but also less reliable methods, e.g., rectangular gating as described in [1], are used.

In this paper the usage of ellipsoidal gating as the only gating method in a tracking algorithm is denoted by pure ellipsoidal gating (PEG). In the following, a two-step approach is introduced to overcome the disadvantage of high computational load of PEG without decreasing its reliability. The approach is denoted by Gating-by-Sorting (GbS) as it takes advantage of sorting algorithms to accelerate the gating procedure and to lower computational complexity. GbS is integrated into an existing MHT algorithm. Tracking results are obtained by applying the algorithm to a real radar dataset presented in [4]. Evaluation is based on computation time and the number of gating processes of PEG and GbS.

The structure of this paper is as follows: Section II describes the MHT algorithm. In Section III, gating in general, and the GbS algorithm, are explained along with complexity considerations. Section IV contains simulation results. Further aspects...
of GbS are discussed in Section V. Finally, Section VI gives a conclusion and an outlook.

Furthermore the terms contact, measurement and observation are used synonymously.

II. MULTI-HYPOTHESIS TRACKING

A. General Concept

MHT is a widely known approach for tracking in cluttered environments. There are different algorithmic implementations of the MHT approach [5]. The one realized in this paper is explained in [6] in detail. It is based on sequential state estimation realized by Kalman filtering.

In principle MHT creates a tree of differently weighted hypotheses about a target state. New hypotheses are generated by associating existing ones to new contacts. The main idea is that the decision, whether a contact is target-originated or a false contact, is postponed until sufficient information is available. In doing so, MHT can generate target-originated tracks in spite of missing contacts or highly cluttered environments. To account for measurement noise and to incorporate kinematic models of targets, MHT includes the Kalman filter [7]. A nonlinear variant of the Kalman filter is necessary due to the nonlinear relation between target state and radar measurements. As the number of contacts within one scan may be high, the number of hypotheses increases quickly over time. Hence, different methods such as gating, pruning, and merging are necessary to limit the number of hypotheses and tracks [8].

B. Kalman Filter

1) Kinematic State Model and Measurement: Tracking is done in the 2-dimensional Cartesian plane. In a state space representation the state vector is defined as

\[ \mathbf{x}_k = [p_x, v_x, p_y, v_y]^T, \]  

with \( p \) denoting the position in Cartesian coordinates and \( v \) denoting the velocity for the corresponding dimensions. Assuming a linear system dynamic, the system equation is

\[ \mathbf{x}_k = \mathbf{A} \mathbf{x}_{k-1} + \mathbf{w}_{k-1}. \]  

The system matrix \( \mathbf{A} \) is chosen so that it matches the Nearly Constant Velocity (NCV) model given in [9]. The process noise, representing deviations from the assumed kinematic model, is represented by the white Gaussian noise \( \mathbf{w}_{k-1} \).

The measurement vector \( \mathbf{y}_k \) comprises the bearing \( \varphi \) and the distance \( r \) between receiver and potential target:

\[ \mathbf{y}_k = [r, \varphi]^T. \]

The nonlinear relation between \( \mathbf{x}_k \) and \( \mathbf{y}_k \) is given by the function \( h \). White Gaussian noise \( \mathbf{v}_k \) is added to model measurement noise:

\[ \mathbf{y}_k = h(\mathbf{x}_k) + \mathbf{v}_k. \]

The output function \( h \) transforms Cartesian coordinates into polar coordinates.

2) Filtering: Process noise \( \mathbf{w}_{k-1} \) and measurement noise \( \mathbf{v}_k \) are considered within the Kalman filter. Its estimation process is divided into two steps:

a. Time update
b. Measurement update

The first step calculates an a priori estimation \( \hat{\mathbf{x}}_{k|k-1} \) of the target state \( \mathbf{x}_k \) using the NCV model. As indicated by the index \( k|k-1 \), this step considers all measurements up to time step \( t_{k-1} \). Afterwards, the measurement-update step calculates the a posteriori estimation \( \hat{\mathbf{x}}_{k|k} \) using all measurements up to time step \( t_k \). In this work the Unscented Kalman filter [10] (UKF) is used due to the nonlinearity of \( h \).

C. Data Association and Hypotheses Weighting

It is not known a priori whether a contact is target-originated or clutter. Hence all contacts are considered in the process of hypothesis forming [8]. Given a set \( X_{k-1} = \{ \hat{x}_{k-1|k-1}^1, \hat{x}_{k-1|k-1}^2, \ldots, \hat{x}_{k-1|k-1}^{m_{k-1}} \} \) of \( m_{k-1} \) hypothesis states at time step \( t_{k-1} \), subsequent hypothesis states \( X_k \) are obtained by associating each hypothesis of \( X_{k-1} \) with each of the \( n_k \) contacts at \( t_k \) of the set \( Y_k = \{ y_k^1, y_k^2, \ldots, y_k^{n_k} \} \).

The weight of a hypothesis state \( i \) represents its probability of being the actual target state. First, one has to calculate the preliminary weights \( \tilde{\omega} \) [11]:

\[ \tilde{\omega}^i_{k-1} \propto f_c \mathcal{N}(\mathbf{y}_k^i; h_{UT}(\hat{\mathbf{x}}_{k|k-1}^i), \mathbf{S}_k^i), \quad 1 \leq i \leq m_k, \]

\[ \omega^i_{k-1} = \left( 1 - P_d \right), \quad j = 0, \]

\[ \omega^i_{k-1} \leq \omega^i_{k-1} \leq \omega^i_{k-1} \]

\[ j = 0 \] considers the hypothesis of no contacts belonging to the target. \( h_{UT}(\hat{\mathbf{x}}_{k|k-1}^i) \) is the result of the unscented transformation of the prediction \( \hat{\mathbf{x}}_{k|k-1}^i \) of the Kalman filter. \( \mathbf{S} \) denotes the innovation covariance or residual covariance respectively calculated by the Kalman filter during the measurement update step. \( P_d \) denotes the assumed probability of detection of the targets and is determined distance-dependent according to [4]. \( f_c \) represents the clutter density calculated adaptively by means of density contributions [12]. \( \mathcal{N} \) denotes the normal distribution.

The final weights are obtained by normalizing the preliminary weights:

\[ \omega^i_k = \frac{\omega^i_{k-1}}{\sum_{j=0}^{n_k+1} \sum_{i=1}^{m_k} \omega^j_k}. \]  

As indicated by (8), the number of hypotheses increases by the factor \( n_k+1 \) at each time step. Computation might not be feasible anymore. Thus different techniques of hypothesis reduction such as gating, pruning, and merging are used [6]. In addition, confirmation and deletion of tracks as part of the track management of MHT is carried out by means of Sequential Track Extraction [13].
As can be seen in the given description of MHT, many calculation steps concern hypothesis states of a track. Especially data association does not refer to measurement-to-track but to measurement-to-hypothesis associations. Hence, in the following, the focus is on the processing of hypotheses. It should be mentioned that the methods given in the following section can be applied to non-MHT approaches as well.

III. GATING

A. Ellipsoidal Gating

Considering (1) and (2) as well as the denotation given in (8), the final gating condition ends up to

$$\begin{align*}
[y_k^j - \hat{y}_k^j]^T (S_k^{ij})^{-1} [y_k^j - \hat{y}_k^j] & \leq G, \quad (10)
\end{align*}$$

with

$$\hat{y}_k^j = h_{UT}(\hat{x}_{k|k-1}) \quad (11)$$

being the predicted measurement. As the Mahalanobis distance takes the covariance into account, it automatically regards the uncertainty of the measurement and its prediction. As a result the gate volume is, in general, limited by an ellipse as shown in Fig. 1a. So even measurements which are far away in terms of Euclidean distance may fall within the gate while the ones which are closer to the prediction may not fall within the gate. The exact shape of the ellipse and its size depend on the composition of the innovation covariance matrix \(S\). In this paper the gate size \(G\) corresponds to an association gate probability of 99\% which is a common value in the literature, e.g. [2], [14].

Only if a measurement falls within the ellipse, further association-related calculations are done. If it does not satisfy the gating condition, the measurement-to-hypothesis association is considered to be too unlikely. In this case, further processing steps like measurement update, weight calculation, eventually hypotheses merging, and pruning do not need to be carried out, thus saving computation time and memory.

It is obvious that the calculation of the Mahalanobis distance is mainly responsible for the computation time of an EG-procedure. In the case of PEG the inversion \((S \rightarrow S^{-1})\) has to be conducted for each measurement-to-hypothesis pairing \(ij\) individually as the residual covariance matrix \(S^{ij}\) is different for each pairing. As a result, the computation time \(T_{\text{PEG}}\) is

$$T_{\text{PEG}} \in O(m \cdot n), \quad (12)$$

with \(m\) being the number of hypotheses and \(n\) being the number of contacts. The so called big \(O\) notation \(O(.)\) characterizes the time complexity.

PEG is a precise gating method, but in practical application the spatial extension of the ellipsoidal gate is much smaller than the spatial extension of all contacts as indicated in Fig. 1b. Hence, calculating the Mahalanobis distance for precise gating is actually unnecessary for most of the contact-to-hypothesis pairings, as most pairings could be gated by less precise gating methods. This fact can be utilized by using rectangular gating [1] which is described in the next section.

B. Rectangular Gating

As the precise ellipsoidal gate covers only a very small area within the complete tracking area, one may conduct a less precise but fast gating procedure which in this case is rectangular gating (RG). Using the two-dimensional NCV model, the RG condition can be expressed as

$$|\tilde{p}_{x,k}^j - \tilde{p}_{x,k}^i| < l_x \quad \land \quad |\tilde{p}_{y,k}^j - \tilde{p}_{y,k}^i| < l_y, \quad (13)$$

with \(\tilde{p}_{x,k}^i, \tilde{p}_{y,k}^i\) being the elements (xy-coordinates) of the predicted state vector \(\tilde{x}_{k|k-1}^j\) and \(\tilde{p}_{x,k}^j, \tilde{p}_{y,k}^j\) being the Cartesian coordinates of the corresponding contact \(j\). This condition is illustrated in Fig. 1b by the rectangle. As can be seen in (13), no matrix inversion is required and hence the calculations are faster than in case of PEG. As RG is only considered to be a coarse gating procedure, contacts passing this rectangular gate should be gated by the more precise ellipsoidal gating. So RG can be seen as a preliminary gating step. To ensure that RG does not influence the result of PEG, the limits \(l_x\) and \(l_y\) which determine the rectangular gate size should be set appropriately large [1].

Although RG prevents PEG from calculating the Mahalanobis distance for very unlikely measurement-to-hypothesis pairings, still each measurement-to-hypothesis pairing \(ij\) has to be checked for the condition given in (13). This results again in a time complexity of \(O(m \cdot n)\), though the final computation time will likely be less.
C. Gating-by-Sorting

1) Concept: The main idea of Gating-by-Sorting (GbS) is to highly accelerate RG by reformulating the condition in (13) and hence being able to apply more efficient algorithms. For now, to provide better readability, the reformulation of (13) is applied to dimension \( x \) only:

\[
\hat{p}_{x,k}^i - l_x < \tilde{p}_{x,k}^i < \hat{p}_{x,k}^i + l_x. \tag{14}
\]

With the lower bound

\[
b_{low}^i = \hat{p}_{x,k}^i - l_x, \tag{15}
\]

and the upper bound

\[
b_{up}^i = \hat{p}_{x,k}^i + l_x, \tag{16}
\]

(14) can be written as

\[
b_{low}^i < \tilde{p}_{x,k}^i < b_{up}^i. \tag{17}
\]

\( b_{low} \) and \( b_{up} \) determine the edges of the stripes shown in Fig. 1b which can be considered as a gate with a fixed gate size defined by \( l_x \). So, for each hypothesis one has to find all contacts which lie within the boundaries \( b_{low} \) and \( b_{up} \). Searching for these contacts within an arbitrary contact list \( Y \) would require checking every contact for each hypothesis. But using a contact list \( Y_{sorted}^x \) in which the contacts are sorted along the dimension \( x \), only two contacts have to be found:

1) The first contact \( j_{low} \) with \( \hat{p}_{x,k}^{j_{low}} < b_{low} \).
2) the last contact \( j_{up} \) with \( \hat{p}_{x,k}^{j_{up}} < b_{up} \).

All contacts of the sorted list which lie between these two contacts are automatically within the boundaries and do not need to be checked further (for one dimension). In addition, searching for these two contacts within a sorted list can be implemented very efficiently.

2) Implementation: In the practical application, contacts are not gated in a single- but in a multi-dimensional space (e.g., two dimensions in this paper). Hence GbS includes further steps to deal with multi-dimensional gating: Firstly, the concept given in Section III-C1 is applied to all dimensions separately. For each dimension \( D \) the result will be a contact list \( Y_{gated}^D \) containing only contacts which have passed the gate of the corresponding dimension. Accordingly, the intersection set of all the lists contains only contacts which have passed the gating for all dimensions. Only these few remaining contacts will then be gated using ellipsoidal gating as a final step. Basically, GbS can be seen as a two-step approach which first uses accelerated rectangular gating with fixed gate sizes. Very unlikely contact-to-hypotheses pairings are sorted out quickly before they are gated by the precise but slower individual ellipsoidal gating. In contrast PEG uses precise individual ellipsoidal gating even for every, and as such, also for very unlikely, pairings. A block diagram of the GbS-algorithm is given in Fig. 2.

3) Complexity: A precise computation-time analysis of the GbS approach is hardly feasible since it strongly depends on the choice of the search and sorting algorithm and on the computation performance of the hardware. However, a general complexity estimation can be given if assumptions about the search and sorting algorithms are made: An efficient sorting algorithm is used and searching is carried out by a binary search algorithm. The sorting algorithm used in this paper is Quicksort whose average computation time \( T_{sort}(n) \) can be assumed to be

\[
T_{sort}(n) \in \mathcal{O}(n \ln(n)), \tag{18}
\]

with \( n \) being the number of elements to be sorted.2 Furthermore the binary searching computation time \( T_{search}(n) \) follows

\[
T_{search}(n) \in \mathcal{O}(\ln(n)), \tag{19}
\]

with \( n \) being the size of the sorted list through which is searched [15]. In this case, \( n \) is the number of contacts. Combining \( T_{sort} \) and \( T_{search} \) as well as considering the fact that searching has to be conducted for each of the \( m \) hypotheses individually the total computation time is

\[
T_{tot}(m, n) \in \mathcal{O}((m + n) \ln(n)). \tag{20}
\]

So GbS can be assumed to take linearithmic time which is an improvement compared to the classical PEG approach’s complexity of \( \mathcal{O}(m \cdot n) \) (see Section III-A).

---

2 In the worst case Quicksort uses \( n^2/2 \) comparisons [15]. This is the unlikely event of the elements being already sorted.
IV. SIMULATIONS AND RESULTS

GbS is implemented in an existing MHT algorithm [12] using MATLAB and applied to a radar dataset denoted by PITAS11. Tracking is performed twice, once using classical ellipsoidal gating and once using GbS. To compare the results, not only the total computation time is measured but also the computation times for the different components (e.g., searching, sorting) are analyzed. In addition, the number of calls of the different steps are counted.

A. Radar Dataset

The PITAS11 dataset is a radar dataset. It was obtained using a navigation radar during a sea trial in the Eckernförde Bight in the Baltic Sea in northern Germany in 2011. The rotation speed of the radar is 24 rpm. The duration of the scenario is 40 min resulting in 960 scans. The number of contacts per scan is about 100. All contacts of all scans are shown in Fig. 4. The main goal of the sea trial was to determine the tracking system’s ability for small-target tracking [4]. Here, the focus is on the comparison of GbS with classical ellipsoidal gating in terms of computation time within a realistic environment.

The trial was conducted within the scope of the project PITAS (Ger.: Piraterie- und Terror-Abwehr auf Seeschiffen, Engl.: Pirate and Terrorist Aversion System). Further information about PITAS and the sea trial (other sensors such as sonar and camera, more results, threat analysis) are to be found in [4] and [16].

B. Results

The tracks of the complete simulation result are shown in Fig. 4. In terms of tracks the usage of GbS does not differ from PEG. However, the computation time for the complete tracking procedure is 2665 sec using GbS while using PEG the tracking algorithm needs 6123 sec.

Tables I and II show detailed times for the different computing steps within the gating processes. The computation time holds for the complete tracking procedure and includes effort for profiling. The numbers in brackets show the computation time for different dimensions or indicate the number of calls to be doubled respectively. Of course, as Table II shows the results for PEG, many steps are not listed as they only occur in GbS.

Even though GbS carries out more steps, the computation time for the complete procedure is much lower compared to PEG. The reason is that PEG conducts significantly more ellipsoidal gating steps than GbS. In GbS the most time consuming step is the calculation of the intersection set which causes more than half of the computation time. PEG needs 3692 sec and hence caused about 60% of the overall time. GbS causes only 10% of the entire processing time (266 sec). So, GbS is about 14 times faster than PEG and reduces the complete computing time by 56%.

Table I: GbS simulation results. Time values are rounded to whole seconds. (%)-values refer to percentage of total computation time.

<table>
<thead>
<tr>
<th>GbS steps</th>
<th>Time / sec</th>
<th>%</th>
<th>#Calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sorting Contacts</td>
<td>&lt; 1</td>
<td>0.04</td>
<td>960 (×2)</td>
</tr>
<tr>
<td>Determin Boundaries</td>
<td>3 (2, 1)</td>
<td>0.11</td>
<td>≈ 622 000 (×2)</td>
</tr>
<tr>
<td>Searching Contacts</td>
<td>9 (6, 3)</td>
<td>0.38</td>
<td>≈ 622 000 (×2)</td>
</tr>
<tr>
<td>Find Intersection</td>
<td>173</td>
<td>6.49</td>
<td>≈ 622 000</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>38</td>
<td>1.43</td>
<td>≈ 622 000</td>
</tr>
<tr>
<td>Ellipsoidal Gating</td>
<td>42</td>
<td>1.58</td>
<td>≈ 580 000</td>
</tr>
<tr>
<td><strong>Sum</strong></td>
<td><strong>266</strong></td>
<td><strong>9.98</strong></td>
<td></td>
</tr>
</tbody>
</table>
Table II: PEG simulation results.

<table>
<thead>
<tr>
<th>PEG steps</th>
<th>Time / sec</th>
<th>%</th>
<th>#Calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ellipsoidal Gating</td>
<td>3692</td>
<td>60.30</td>
<td>≈ 49,304,000</td>
</tr>
<tr>
<td>Sum</td>
<td>3692</td>
<td>60.30</td>
<td>-</td>
</tr>
</tbody>
</table>

V. DISCUSSION

The results in Section IV-B show a significant acceleration of the gating procedure using GbS compared to classical ellipsoidal gating. In this section further aspects are considered.

A. Matrix Inversion

As ellipsoidal gating requires a matrix inversion. One possibility to reduce processing time is to implement more efficient matrix inversion algorithms. On the one hand this could reduce the computation time for PEG, but on the other hand in PEG the number of matrix inversion calculations is much higher than in case of GbS. In addition, GbS would also benefit from a better matrix inversion. Hence it can be assumed that a better implementation of the matrix inversion will not make PEG faster than GbS but possibly will only reduce the gap between the two methods a little bit.

B. Fixed Gate Size

As described in (14), fixed gate sizes for every dimension are needed for GbS. In order to ensure that GbS gates as precise as PEG, these gate sizes must be set large enough to prevent them from “cutting” any ellipses of any contact-to-measurement pairing. In this paper, the sizes are set large enough but the resulting area is still much smaller than the entire spatial extension of the contacts and, hence, GbS works efficiently. There are two possible cases in which this would not work: Either the spatial extension of the ellipses covers a high percentage of tracking space or their maximum extension can not be estimated a priori with sufficient precision. In both cases the fixed gate size has to be set sufficiently large that the resulting area would cover a lot of space compared to the entire tracking space which would make GbS less effective. This aspect is visualized in Fig. 5.

C. Intersection

Calculating the intersection set is the most “expensive” step as can be seen in Table I. It takes more time the more contacts the gated-contact lists \(Y_{gated}^{D}\) (see Section III-C2) contain and the more dimensions exist. So, its implementation should be analyzed and improved to further accelerate the gating procedure. Furthermore one could sort and search in one dimension only and afterwards the resulting contacts are processed sequentially only, thus avoiding intersection. This may accelerate the whole gating procedure depending on the number of contacts per gated list and the number of dimensions.

D. MATLAB-Implementation

As already mentioned, the algorithms are implemented using MATLAB-script language. The advantage of MATLAB is that algorithms can be easily realized and hence results can be obtained quickly. As a drawback the execution time is, in general, slower compared to implementations using compiled programming languages such as C/C++. However, GbS reduces the computation time by reducing the number of ellipsoidal gating-process calls (see Table I and II). While the computation time is influenced by the choice of the programming language, the number of calls is not. As such, when using GbS instead of PEG the execution time is expected to be reduced independent of the choice of the programming language. However, the amount of time which can be saved is likely to depend on the language.

E. Exactness of Data Association

Any kind of gating procedure produces four different result cases:
1) True positive: target-originated contacts fall within the gate.
2) True negative: clutter-contacts fall outside the gate.
3) False positive: clutter-contacts fall within the gate.
4) False negative: target-originated contacts fall outside the gate.

Since GbS is a two-step approach which includes ellipsoidal gating as the second step, the resulting gate size can only be equal or smaller than the gate size of PEG. It is equal if the ellipsoidal gate lies completely within the rectangular gate and it is smaller if the rectangle cuts the ellipse (see Fig. 5). In this case, the resulting gate shape is the intersection of the rectangle and the ellipse. Considering this fact, the number of true positive ($TP$), true negative ($TN$), false positive ($FP$) and false negative ($FN$) cases can be considered to be:

$$TP_{GbS} \leq TP_{PEG} \quad (21)$$
$$TN_{GbS} \geq TN_{PEG} \quad (22)$$
$$FP_{GbS} \leq FP_{PEG} \quad (23)$$
$$FN_{GbS} \geq FN_{PEG}. \quad (24)$$

If the fixed gate sizes of GbS are chosen appropriately (see Section V-B), $\leq$ and $\geq$ can be replaced by $=.$

VI. CONCLUSION AND OUTLOOK

In this paper, a new approach for gating is presented. This approach, namely Gating-by-Sorting, is implemented into a tracking system using an MHT-algorithm applied to a real radar dataset and analyzed in terms of computation time. It has been shown that GbS accelerates the gating process significantly. The main concept of GbS is to conduct a first coarse but fast gating step using sorting algorithms and binary search. Hence, very unlikely contacts are sorted out and the following slower ellipsoidal gating step does not need to be carried out as often. Furthermore, complexity considerations indicate that the advantage of GbS compared to the pure usage of ellipsoidal gating even increases when the number of contacts or hypotheses increases.

To further verify these results, the approach needs to be applied to other datasets. For example, GbS may be applied on simulated scenarios with different numbers of contacts to analyze the effect of GbS holistically.

So far, GbS is only used within the hypothesis generation of MHT. Since gating is also used in track and hypothesis merging, GbS should be tested for this purpose as well.

Since sorting is an essential part of GbS, other efficient sorting algorithms such as Merge Sort or Heap Sort [17] are to be considered for GbS. Furthermore, as searching is also a part of GbS, the potential advantage of Hash functions may be analyzed [15]. However it is to be considered that these steps take only very little execution time.

As mentioned before, GbS is a two-step approach with the first step using a fixed gate size. Only the second step, i.e., individual ellipsoidal gating, uses the variable gate sizes depending on the covariances of the measurement and the prediction. As pointed out in Section V-B, the usage of these fixed limits may be disadvantageous in certain scenarios. So the main goal of the future work is to develop a method which actually uses the covariances of the measurements and the predictions within one single step, but which is still as fast as GbS.

ACKNOWLEDGMENT

This work was made possible by Thomas Lehmann and Carsten Reiter from Raytheon Anschütz GmbH who organized the sea trial. The authors would like to thank Christoph Norrenbrock for helpful discussions and support.

This work was supported by the Federal Ministry of Economics and Technology (Germany).

REFERENCES