Efficient Data Association for 3D Passive Sensors: If I Have Hundreds of Targets and Ten Sensors (or More)

Shuo Zhang  
Dept. of Electrical and Computer Engineering  
University of Connecticut  
Storrs, CT 06269-2157  
Email: shuo.zhang@engr.uconn.edu

Yaakov Bar-Shalom  
Dept. of Electrical and Computer Engineering  
University of Connecticut  
Storrs, CT 06269-2157  
Email: ybs@engr.uconn.edu

Abstract—This paper considers the passive-sensor data association problem based on multi-dimensional assignment (MDA). The 5-D algorithm has been shown to be effective for solving the MDA problem. The bottleneck of the S-D algorithm lies in its cost computation, which consumes about 95%-99% of the CPU times. Since the number of costs in the MDA problem increases exponentially with the number of sensors, the S-D algorithm becomes quite inefficient when a large number of sensors are used. We propose an efficient data association technique, “S$_0$-D+Seq(2-D)” algorithm, which decomposes the original problem to an S$_0$-dimensional assignment and several 2-dimensional assignments. The S$_0$-D+Seq(2-D) algorithm yields a total number of costs which only increases quadratically with the number of sensors. Simulation results show that the S$_0$-D+Seq(2-D) algorithm achieves a significant reduction in CPU times compared to the S-D algorithm with similar association qualities.

Keywords: Data association, passive sensor, multiple dimension assignment (MDA), S-D algorithm.

I. INTRODUCTION

Data association is a crucial task in many surveillance systems. In general, the goal of data association is to partition the set of measurements across sensors into a number of subsets, in which the measurements are either from the same target (i.e., having the identical origin) or false alarms. In tracking problems, data association is carried out before filtering (e.g., Kalman filtering). Consequently, the quality of data association is critical to the overall tracking performance.

Data association becomes especially challenging if the sensors are passive and measure line of sight (LOS) angles only for the targets. Measurements from multiple sensors have to be associated to determine the full positions of the targets. The brute force approach, i.e., enumerating all possible combinations and choosing the most likely one, is computationally prohibitive even for a moderate size problem. For example, the total number of combinations for a scenario of 20 targets and 2 sensors (assuming no missed detections or false alarms) is $20! = 2.4 \times 10^{18}$. A practical approach is to formulate the multisensor data association as a multiple dimensional assignment (MDA) problem [2] and then employ (constrained) optimization techniques to obtain the optimal assignment. When the number of sensors is greater than or equal to three (i.e., $S \geq 3$), the MDA is known to be NP hard. While a number of suboptimal techniques have been proposed, the Lagrangian relaxation based approaches [10], [12] have been shown to be superior to others (e.g., branch and bound, row-column heuristic) for their excellent balance between the accuracy and the efficiency. The relaxation technique in [7] is termed as the S-D (assignment) algorithm. In [13] an extended approach of determining the top $m$ assignments (as opposed to only the best one) has been obtained by using Murty’s ranking algorithm [8].

Prior to the optimization step in the S-D algorithm, the first step is to calculate the candidate association costs. It has been reported [15], [1] that this cost-calculation step consumes 95%-99% of the CPU time. Consequently, when the number of targets is large, a direct use of the S-D algorithm can become quite inefficient. Thus, for the large-scale problem clustering techniques [6] are applied before carrying out the S-D algorithm. By employing the clustering (or partitioning), the original large problem is reduced to a number of smaller subproblems, which can be solved efficiently by the S-D algorithm.

However, even with the clustering, the CPU time of the S-D algorithm increases drastically when a large number of sensors are used. The reason is that the number of costs to be computed increases exponentially with the number of sensors, although this computational burden can be mitigated by employing the gating technique [2], [5] to remove unlikely candidate associations. Note that it is not uncommon to have a large number of passive sensors since some of them have lower costs and are easier for deployment, e.g., infrared or CCD cameras. Aiming at the large-scale data association, that is, when a large number of targets are in presence and many sensors are used, we propose an efficient data association technique: “S$_0$-D+Seq(2-D)” algorithm. This algorithm first decomposes the original problem to an S$_0$-dimensional assignment and several 2-dimensional assignments.

Research sponsored under ONR N00014-07-1-0131 and ARO W911NF-06-1-0467 grants. 14th International Conference on Information Fusion, Chicago, IL, July 2011.
the original problem to a (fixed) $S_0$-dimensional assignment and $S - S_0$ 2-dimensional assignments. Then the former is solved by using the S-D algorithm and the latter is solved by a successive use of the (modified) Auction algorithm [10]. In Simulations, a scenario of 300 targets and 10 passive sensors with LOS measurements (i.e., azimuth and elevation) has been tested. Each sensor may receive more than 300 measurements due to false alarms. The CPU time for this scenario is about 6 seconds by using the $S_0$-D+Seq(2-D) algorithm.

The paper is organized as follows. In Section II the MDA problem is formulated for the 3D passive sensors. In Section III the iterative least squares (ILS) technique is presented for target position estimation using the LOS measurements. In Section IV the dihedral-angle based clustering technique is discussed. The proposed $S_0$-D+Seq(2-D) algorithm is given in Section V. The S-D and $S_0$-D+Seq(2-D) algorithms are compared in Section VI based on a large-scale localization problem. Finally, conclusions and future work are given in Section VII.

II. FORMULATION OF THE MDA PROBLEM

Assume that there are $S$ passive sensors in a 3D space, with known positions $p_s = [x_s, y_s, z_s]^T$ $(s = 1, \ldots, S)$. The sensors are assumed to be synchronized. For a given target, each sensor provides its LOS measurement: azimuth angle and elevation angle, namely,

\[ z_{is} = h(x, p_s) + w_s \quad s = 1, \ldots, S \tag{1} \]

where $i_s$ is the measurement index in sensor $s$, $x = [x, y, z]^T$ denotes the target’s position, $w_s$ is zero-mean white Gaussian measurement noise with covariance $R_s$ and

\[ h(x, p_s) = \begin{bmatrix} \alpha_s \\ \varepsilon_s \end{bmatrix} = \begin{bmatrix} \tan^{-1} \left( \frac{y - y_s}{x - x_s} \right) \\ \tan^{-1} \left( \frac{z - z_s}{\sqrt{(x - x_s)^2 + (y - y_s)^2}} \right) \end{bmatrix} \tag{2} \]

Each sensor may receive a number of such measurements from multiple targets, as well as false alarms. An $S$-tuple of measurements $Z_{i_1i_2...i_S}$ consisting of one measurement from each sensor, represents a possible association, that is, the measurements $Z_{i_1i_2...i_S}$ are assumed to originate from the same target. Since a target may not be detected by every sensor, a dummy measurement is added to each sensor with index 0, to represent the missed detection. If there is only one nondummy measurement in a $S$-tuple, this nondummy measurement is deemed to be a false alarm. For each $S$-tuple there is an associated cost $c_{i_1i_2...i_S}$, which is given by the negative log-likelihood ratio [2]

\[ c_{i_1i_2...i_S} = -\ln \frac{\Lambda(Z_{i_1i_2...i_S} | \mathbf{x})}{\Lambda(Z_{i_1i_2...i_S} | \emptyset)} \tag{3} \]

The numerator in (3) represents the likelihood that the $S$-tuple of measurements $Z_{i_1i_2...i_S}$ originate from the same target with position $\mathbf{x}$, namely,

\[ \Lambda(Z_{i_1i_2...i_S} | \mathbf{x}) = \prod_{s=1}^{S} [1 - P_{D,s}]^{-u(i_s)} [P_{D,s}p(z_{is} | \mathbf{x})]^{u(i_s)} \tag{4} \]

where $P_{D,s}$ is the detection probability of sensor $s$, $u(i_s)$ is an indicator function, defined as

\[ u(i_s) = \begin{cases} 0 & \text{if } i_s = 0 \\ 1 & \text{otherwise} \end{cases} \tag{5} \]

and $p(z_{is} | \mathbf{x})$ is given by

\[ p(z_{is} | \mathbf{x}) = \frac{1}{2\pi R_s^{-1/2}} \exp \left( -\frac{1}{2} \left| z_{is} - h(x, p_s) \right|^2 / R_s \right) \tag{6} \]

The denominator in (3) represents the likelihood that all the measurements in the $S$-tuple are false alarms, namely,

\[ \Lambda(Z_{i_1i_2...i_S} | \emptyset) = \prod_{s=1}^{S} \lambda^{u(i_s)} \tag{7} \]

where $\lambda$ denotes the spatial density [2] of the false alarms.

The MDA problem is formulated as follows [7]

\[ \min_{\rho_{i_1i_2...i_S}} \sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_2} \cdots \sum_{i_S=0}^{n_S} c_{i_1i_2...i_S} \rho_{i_1i_2...i_S} \tag{8} \]

subject to

\[ \sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_2} \cdots \sum_{i_S=0}^{n_S} \rho_{i_1i_2...i_S} = 1; \quad i_1 = 1, 2, \ldots, n_1 \]

\[ \sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_2} \cdots \sum_{i_S=0}^{n_S} \rho_{i_1i_2...i_S} = 1; \quad i_2 = 1, 2, \ldots, n_2 \]

\[ \vdots \]

\[ \sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_2} \cdots \sum_{i_S-1=0}^{n_S-1} \rho_{i_1i_2...i_S} = 1; \quad i_S = 1, 2, \ldots, n_S \tag{9} \]

where $\rho_{i_1i_2...i_S} \in \{0, 1\}$. Thus, the goal is to find $\{\rho_{i_1i_2...i_S}\}$, i.e., a partition of the total measurements that minimizes the global cost, subject to the constraints that each measurement is associated with one and only one measurement (including the dummy measurement) in each sensor. When $S = 2$, this MDA problem can be solved exactly by using the modified Auction algorithm [10]. In the general case, i.e., $S > 2$, this problem is NP hard and can only be solved suboptimally. The S-D algorithm [7], which is based on the Lagrangian relaxation, has been shown to be an effective technique to solve this general MDA problem.

III. POSITION ESTIMATION VIA ITERATIVE LEAST SQUARES

Since the target position $\mathbf{x}$ in (4) is unknown, it is substituted by its estimate $\hat{x}$ obtained from the $S$-tuple of measurements $Z_{i_1i_2...i_S}$. While there are a number of methods to obtain $\hat{x}$, the iterative least squares (ILS) technique [3] is preferred since it is easy to implement (no Hessian involved) and provides a
(approximate) covariance matrix for its estimate at the same time.

Assume that there are \( n \) nondummy measurements in \( Z_{i_1, i_2, \ldots, i_S} \) \( (2 \leq n \leq S) \) and we stack them to form an augmented vector \( \mathbf{z} \). Then, the ILS estimate in the \( j \)th iteration can be written as

\[
\hat{x}^{j+1} = \hat{x}^j + [(H^j)' R^{-1} H^j]^{-1} (H^j)' R^{-1} [\mathbf{z} - h(\hat{x}, \mathbf{p})] \tag{10}
\]

where \( R = \text{diag}([R_1, \ldots, R_n]^2) \), \( \hat{z} = [z'_1, \ldots, z'_n]' \), \( h(\hat{x}, \mathbf{p}) = [h(\hat{x}, p_1)', \ldots, h(\hat{x}, p_n)']' \) and

\[
H^j = \frac{\partial h(\hat{x}, \mathbf{p})}{\partial \mathbf{x}} \bigg|_{\mathbf{x} = \hat{x}^j} \tag{11}
\]

is the Jacobian matrix of the stacked measurement vector evaluated at \( \hat{x}^j \). In this case, the Jacobian matrix is

\[
H = \begin{bmatrix} H_1' & \cdots & H_n' \end{bmatrix}' \tag{12}
\]

where

\[
H_i = \begin{bmatrix} \frac{\partial \alpha_i}{\partial x} & \frac{\partial \alpha_i}{\partial y} & \frac{\partial \alpha_i}{\partial z} & \frac{\partial \alpha_i}{\partial \epsilon} \\ \frac{\partial \epsilon_i}{\partial x} & \frac{\partial \epsilon_i}{\partial y} & \frac{\partial \epsilon_i}{\partial z} & \frac{\partial \epsilon_i}{\partial \epsilon} \end{bmatrix} \tag{13}
\]

and \( \| \cdot \| \) denotes the Euclidean norm.

To start the ILS recursion an initial estimate \( \hat{x}^0 \) is required, which is given by [9]

\[
\hat{x}^0 = \frac{y_2 - y_1 + x_1 \tan \alpha_1 - x_2 \tan \alpha_2}{\tan \alpha_1 - \tan \alpha_2} \tag{20}
\]

\[
\hat{y}^0 = \frac{\tan \alpha_1 (y_2 + \tan \alpha_2 (x_1 - x_2)) - y_1 \tan \alpha_2}{\tan \alpha_1 - \tan \alpha_2} \tag{21}
\]

\[
\hat{z}^0 = z_1 + \tan \epsilon_1 \frac{|y_1 - y_2| \cos \alpha_2 + (x_2 - x_1) \sin \alpha_2}{\sin(\alpha_1 - \alpha_2)} \tag{22}
\]

which has made use of the first two measurements.

The subscript in \( R_i \) is the index of the nondummy measurement and is not the sensor index. It is different from that of the previous section. This holds for other variables.

IV. CLUSTERING

In the S-D algorithm, the most expensive step is computing the association costs, which consumes 95%-99% of the CPU time [15], [1]. From (8) the total number of costs to be calculated is

\[
n_c = \prod_{s=1}^{S} (n_s + 1) \tag{23}
\]

For simplicity, assuming that the number of measurements is \( n_0 \) for every list (sensor), then

\[
n_c = (n_0 + 1)^S \tag{24}
\]

Consequently, a large \( n_0 \) is unfavorable for the efficiency of the S-D algorithm.

The clustering technique is used to reduce a large-size problem to a number of smaller subproblems, which can be solved independently. The clustering algorithm groups measurements based on a distance metric. For the 3D passive sensors, an effective metric is the so-called dihedral angle [6]. The dihedral angle is defined as the angle between two planes, a target plane and a reference plane (see Fig. 1). The target plane passes through two sensors and one LOS measurement from either of these two sensors, while the reference plane is the XY plane in the 3D Cartesian space (assuming both of the sensors are located on the XY plane). Given two LOS measurements from two different sensors, if these measurements originate from the same target, then the two dihedral angles, one for each LOS measurement, would be close to each other. As a result, clustering the dihedral angles leads to clustering the respective LOS measurements.

![Figure 1. Clustering using dihedral angles.](image1)

The dihedral angle \( \varphi \) for a LOS measurement \( [\alpha, \epsilon]' \) from sensor A located at the origin with reference to another sensor
Example, assuming the number of sensors increases, the CPU time of the algorithm using the dihedral angles is given in Table I.

The dihedral angles have to be computed in pairwise between each pair of sensors. A summary of the clustering algorithm using the dihedral angles is given in Table I.

Table I

<table>
<thead>
<tr>
<th>FOR sensor s = 1 : S</th>
<th>S - 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOR sensor j = s + 1 : S</td>
<td></td>
</tr>
<tr>
<td>Cluster measurements in sensors s and j using the dihedral angles</td>
<td></td>
</tr>
<tr>
<td>END FOR</td>
<td></td>
</tr>
<tr>
<td>Find the measurements that have not yet been clustered for the use of the next iteration</td>
<td></td>
</tr>
<tr>
<td>END FOR</td>
<td></td>
</tr>
</tbody>
</table>

The dihedral angle can be also utilized in gating [2] to prune unlikely associations. If a candidate association fails in the gating test, there is no need to compute its cost, i.e., an infinitely large cost is assigned to it. For a given cluster, the calculation of the candidate costs is recursive. Beginning at $z_i$ in list 1, we take one measurement from each list at a time. If the measurement falls outside the gate defined by the previous measurements in the tuple, this measurement is incorporated in the tuple, which advances to the next list. The cost of the tuple is only evaluated at the last list when a full tuple is achieved. For example, assuming the current list is $m$ and the current association tuple is $Z_{i_1...i_{m-1}}$, if $z_i$ passes the gating test then it is added and form $Z_{i_1...i_m}$, otherwise all the subsequent candidate associations starting with $Z_{i_1...i_m}$ are discarded. Consequently, the CPU time spent in the cost computation can be saved via the gating process.

V. $S_0$-D+Seq(2-D) Algorithm

For a small number of sensors, the $S$-D algorithm (along with the clustering technique if the number of targets is large) is able to perform in real time. However, when the number of the sensors increases, the CPU time of the $S$-D algorithm increases drastically, which is impossible for the $S$-D algorithm to operate in real time. We can see from (24) the total number of costs increases exponentially with $S$. For example, assuming $n_0$ in (24) is 7 (in each cluster we expect a small number of $n_0$), the numbers of costs for the $S$-D algorithm are shown in Table II for different numbers of sensors. When $S = 10$, the total number of costs is over one billion$^3$ for a single cluster.

We propose an efficient data association technique, called $S_0$-D+Seq(2-D) algorithm, for the case where more than 3 sensors are used. This algorithm consists of two steps, the $S_0$-D step and the Seq(2-D) step, which are presented next.

1. $S_0$-D step: This step solves a standard MDA problem with the dimension of $S_0$ using the $S$-D algorithm, that is, the data association is performed among $S_0$ sensors ($S_0 < S$). While the minimum value of $S_0$ is 2, practically $S_0$ should be at least 3 to achieve quality associations. This is due to the ghosting problem [2] of the passive sensors. The use of more sensors can mitigate this ghosting effect. A large gate (or no gating) is recommended for the $S_0$-D step to prevent discarding some less likely (due to noises) but real associations. The quality of this $S_0$-dimensional assignment is critical to the subsequent associations. Thus, the $S_0$ sensors should be chosen carefully, e.g., choosing the sensors with good geometry (in terms of GDOP), or small measurement errors.

2. Seq(2-D) step: This step solves a series of 2-D assignments sequentially using the modified Auction algorithm [10]. The number of the 2-D assignments is $S - S_0$. After the $S_0$-D step, the $S_0$-tuple association results are available. Then, take a new list from the remaining $S - S_0$ lists and formulate a 2-D assignment between the previous association results and the measurements in this new list. After the 2-D assignment, the length of each association is incremented by one, i.e., becoming a $S_0 + 1$-tuple. Next, take another list from the remaining $S - S_0 - 1$ lists and solve another 2-D assignment, and so on. In the end, after carrying out $S - S_0$ 2-D assignments, each association is in a full tuple, i.e., a $S$-tuple. The $S_0$-D+Seq(2-D) algorithm is summarized in Table III, assuming the $S_0$-D step chooses the first $S_0$ lists, i.e., $s = 1, 2, \ldots, S_0$. An illustration of the $S_0$-D+Seq(2-D) algorithm is shown in Fig. 3 for $S_0 = 3$.

Table II

<table>
<thead>
<tr>
<th>No. of sensors $S$</th>
<th>No. of costs $n_c$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>4,096</td>
</tr>
<tr>
<td>7</td>
<td>2,097,152</td>
</tr>
<tr>
<td>10</td>
<td>1,073,700,000</td>
</tr>
</tbody>
</table>

Table III

<table>
<thead>
<tr>
<th>$S_0$-D+Seq(2-D) Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) $S_0$-D step: Solve the $S_0$-D assignment using the $S$-D algorithm and obtain the $S_0$-tuple association results $(i_1, \ldots, i_{S_0})$.</td>
</tr>
<tr>
<td>2) Seq(2-D) step: FOR $n = S_0 + 1 : S$</td>
</tr>
<tr>
<td>Construct the 2-D association between the previous association results $(i_1, \ldots, i_{n-1})$ and the measurements ${z_{i_n}}$ in list $n$;</td>
</tr>
<tr>
<td>Solve the 2-D assignment using the modified Auction algorithm and obtain the $n$-tuple results $(i_1, \ldots, i_n)$.</td>
</tr>
<tr>
<td>END FOR</td>
</tr>
</tbody>
</table>

Similarly to (24), the number of costs in the $S_0$-D step is $(n_0 + 1)^{S_0}$. In the Seq(2-D) step, among the $S - S_0$ 2-D assignments the largest number of costs occur at the last 2-D
assignment, which in the worst case is \((S - 1)n_0 + 1\). Consequently, an upper bound of the total number of costs of the \(S_0\)-D+Seq(2-D) algorithm is given by

\[
n'_c = (n_0 + 1)^S_0 + (S - S_0)(n_0 + 1)((S - 1)n_0 + 1)
\]

which increases quadratically with \(S\). In Table IV the values of this upper bound are shown for different number of sensors. The association quality of this \(S_0\)-D+Seq(2-D) algorithm is evaluated next in Simulation Results.

### Table IV

Numbers of Costs of the \(S_0\)-D+Seq(2-D) Algorithm for Different Numbers of Sensors (\(n_0 = 7\))

<table>
<thead>
<tr>
<th>No. of sensors (S)</th>
<th>No. of costs (n'_c)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>688</td>
</tr>
<tr>
<td>7</td>
<td>1,888</td>
</tr>
<tr>
<td>10</td>
<td>4,096</td>
</tr>
</tbody>
</table>

We consider a localization problem using the LOS measurements. The numbers of the passive sensors used are 4, 7 and 10. The sensors are located in a circle of radius 5km centered at (5, 5)km in the XY plane, with equal angle separations. The measurement noise standard deviation is 1mrad for both azimuth and elevation angle. All the sensors are assumed to have the same accuracy, detection probability \(P_D\) and false alarm rate \(P_F\). The number of targets is 300 and their positions are randomly placed in the 3D Cartesian space, where the ranges of the X, Y, Z coordinates are 0-10km, 1-10km, 5-10km, respectively. There is no prior information assumed for the number of targets. The sensor-target geometry is shown in Fig. 4 for the case of 10 sensors. The number of Monte Carlo runs is 20. Given an association tuple, if there is only one nondummy measurement, then it is deemed to represent a false alarm, otherwise it falls into one of the following 3 categories (similar to [1]):

1. Completely correct (CC) association: The measurements in an association tuple have identical origin and there is no dummy measurement associated.
2. Partially correct (PC) association: There are at least 2 measurements with common origin, and the rest may be from different origins or dummy measurements.
3. Completely incorrect (CI) association: In an association tuple, there does not exist a pair of measurements that come from the same origin.

Each CC or PC association corresponds to a detected target (DT). The DT is defined as the origin that appears most in an association tuple, and the number of times that the DT appears is referred as the detection index (DI). The detected targets are a subset of the total targets (TT).

Given an association tuple, if the number of the nondummy measurements is no less than a threshold \(T_H\) \((T_H > 1)\), then this association is accepted, otherwise it is rejected. To quantify the quality of the accepted associations, we introduce four metrics: fraction of correct associations, fraction of missed associations, fraction of duplicated associations and fraction of purity, which are defined below:

- **Fraction of correct associations (FCA):**
  \[
  FCA = \frac{N_{CC} + N_{PC}}{N_{HH} + N_{PC} + N_{CI}}
  \]

- **Fraction of missed associations (FMA):**
  \[
  FMA = \frac{N_{TT} - N_{DT}}{N_{TT}}
  \]

- **Fraction of duplicated associations (FDA):**
  \[
  FDA = \frac{N_{CC} + N_{PC}}{N_{DT}}
  \]

- **Fraction of purity (FP):**
  \[
  FP = \frac{N_{DT}}{N_T}
  \]

where \(\overline{DT}\) denotes the average detection index. Note that only \(N_{TT}\) is independent of the threshold \(T_H\). In the following we consider two examples with different \(P_D\) and \(P_F\). For both examples we use \(S_0 = 3\) for the \(S_0\)-D+Seq(2-D) algorithm and \(T_H = 3\) as the nondummy threshold.

\[^4\]\(N_X\) represents the number of \(X\), e.g., \(N_{DT}\) denotes the number of detected targets.
A. Example 1: $P_D = 1$, $P_F = 0$ (ideal case)

In this scenario each sensor receives exactly 300 LOS measurements. The dihedral angle based clustering technique from Section IV is employed. For each cluster the $S$-$D$ algorithm and the $S_0$-$D$+$Seq$(2-$D$) algorithm are used to perform the association. The algorithms are coded in C++ and run on an Intel 2 Duo-2.66 GHz computer. The CPU times for these two algorithms are compared in Table V. The improvement factors of $S_0$-$D$+$Seq$(2-$D$) over $S$-$D$ are about 3 and 130 for 4 sensors and 7 sensors, respectively. As discussed in Section V the advantage of the $S_0$-$D$+$Seq$(2-$D$) algorithm is prominent when a large number of sensors are used. For the case of 10 sensors, the $S$-$D$ algorithm requires huge memories that exceed the computer capacity and no results are obtained. The CPU times of the $S_0$-$D$+$Seq$(2-$D$) algorithm can be further reduced by introducing parallelization to process different clusters concurrently.

The association qualities of these two algorithms are compared in Table VI. In general the quality differences between $S$-$D$ and $S_0$-$D$+$Seq$(2-$D$) are not significant compared to their differences in CPU times. In terms of either correct associations or duplicated associations (from FCA and FDA), which one of $S$-$D$ and $S_0$-$D$+$Seq$(2-$D$) is better depends on the number of sensors used. The $S$-$D$ algorithm has less missed associations (from FMA), while the $S_0$-$D$+$Seq$(2-$D$) algorithm yields purer associations (from FP). With the increase of the number of sensors, both the missed associations and the purities decline for both $S$-$D$ and $S_0$-$D$+$Seq$(2-$D$) algorithms.

B. Example 2: $P_D = 0.98$, $P_F = 10^{-5}$

This $P_F$ corresponds to an average 15 false alarms for each sensor, where the number of the false alarms is assumed to be Poisson distributed. The comparison of the CPU times of the $S$-$D$ and $S_0$-$D$+$Seq$(2-$D$) algorithms is shown in Table VII, which is similar to Table V. Although the detection probability is 0.98, a larger value of $P_D$ ($P_D < 1$) is recommended to use in (4). This is due to a phenomenon to be called “association splitting”, in which a CC or PC association is divided into two or more PC associations, that provide an overall lower cost. This association splitting will result in duplications as well as a less accurate target position estimate. Furthermore, since a split association has less nondummy measurements, it is more likely to be rejected, that is, it may fail to pass the nondummy threshold $T_H$, which will lead to missed associations. The use of a larger $P_D$ will penalize incomplete associations and prevent the association from splitting. A similar phenomenon called “track splitting” was observed in [1] for track-to-track associations.

The association qualities are shown in Table VIII. The qualities in terms of correct associations (FCA), missed associations (FMA) and purity (FP) degrade slightly compared to Example 1 (ideal case) for both $S$-$D$ and $S_0$-$D$+$Seq$(2-$D$) algorithms. However, there is a significant degradation in missed associations (FMA) for these two algorithms, which happens mainly due to two reasons. The first reason is that the use of a large $P_D$ may overpenalize the incomplete associations, discarding some real incomplete associations due to $P_D < 1$. The other reason lies in the clustering step. The clustering algorithm in Section IV relies most on the detections of the first sensor which initiates the clustering process, that is, if a target is undetected in the first sensor, it becomes less likely that the measurements of this target will exist in the same cluster, which can lead to missed associations.

---

### Table V
**CPU Times Comparison (Example 1)**

<table>
<thead>
<tr>
<th>No. of sensors</th>
<th>Algorithm</th>
<th>CPU Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>$S$-$D$</td>
<td>2.97</td>
</tr>
<tr>
<td>4</td>
<td>$S_0$-$D$+$Seq$(2-$D$)</td>
<td>0.94</td>
</tr>
<tr>
<td>7</td>
<td>$S$-$D$</td>
<td>425.03</td>
</tr>
<tr>
<td>7</td>
<td>$S_0$-$D$+$Seq$(2-$D$)</td>
<td>3.24</td>
</tr>
<tr>
<td>10</td>
<td>$S$-$D$</td>
<td>--</td>
</tr>
<tr>
<td>10</td>
<td>$S_0$-$D$+$Seq$(2-$D$)</td>
<td>6.51</td>
</tr>
</tbody>
</table>

### Table VI
**Comparison of Association Qualities (Example 1)**

<table>
<thead>
<tr>
<th>No. of sensors</th>
<th>Algorithm</th>
<th>FCA</th>
<th>FMA</th>
<th>FDA</th>
<th>FP</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>$S$-$D$</td>
<td>97.6%</td>
<td>2.1%</td>
<td>3.5%</td>
<td>91.5%</td>
</tr>
<tr>
<td>4</td>
<td>$S_0$-$D$+$Seq$(2-$D$)</td>
<td>98.6%</td>
<td>3.1%</td>
<td>3.7%</td>
<td>93.9%</td>
</tr>
<tr>
<td>7</td>
<td>$S$-$D$</td>
<td>98.5%</td>
<td>1.2%</td>
<td>5.9%</td>
<td>81.1%</td>
</tr>
<tr>
<td>7</td>
<td>$S_0$-$D$+$Seq$(2-$D$)</td>
<td>98.2%</td>
<td>1.9%</td>
<td>4.1%</td>
<td>87.5%</td>
</tr>
</tbody>
</table>

### Table VII
**CPU Times Comparison (Example 2)**

<table>
<thead>
<tr>
<th>No. of sensors</th>
<th>Algorithm</th>
<th>CPU Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>$S$-$D$</td>
<td>2.98</td>
</tr>
<tr>
<td>4</td>
<td>$S_0$-$D$+$Seq$(2-$D$)</td>
<td>0.96</td>
</tr>
<tr>
<td>7</td>
<td>$S$-$D$</td>
<td>425.44</td>
</tr>
<tr>
<td>7</td>
<td>$S_0$-$D$+$Seq$(2-$D$)</td>
<td>3.21</td>
</tr>
<tr>
<td>10</td>
<td>$S$-$D$</td>
<td>--</td>
</tr>
<tr>
<td>10</td>
<td>$S_0$-$D$+$Seq$(2-$D$)</td>
<td>6.49</td>
</tr>
</tbody>
</table>

### Table VIII
**Comparison of Association Qualities (Example 2)**

<table>
<thead>
<tr>
<th>No. of sensors</th>
<th>Algorithm</th>
<th>FCA</th>
<th>FMA</th>
<th>FDA</th>
<th>FP</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>$S$-$D$</td>
<td>97.3%</td>
<td>4.8%</td>
<td>3.7%</td>
<td>91.7%</td>
</tr>
<tr>
<td>4</td>
<td>$S_0$-$D$+$Seq$(2-$D$)</td>
<td>98.2%</td>
<td>6.6%</td>
<td>3.8%</td>
<td>93.0%</td>
</tr>
<tr>
<td>7</td>
<td>$S$-$D$</td>
<td>98.3%</td>
<td>4.0%</td>
<td>6.3%</td>
<td>80.4%</td>
</tr>
<tr>
<td>7</td>
<td>$S_0$-$D$+$Seq$(2-$D$)</td>
<td>97.5%</td>
<td>5.3%</td>
<td>4.3%</td>
<td>86.1%</td>
</tr>
<tr>
<td>10</td>
<td>$S$-$D$</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>10</td>
<td>$S_0$-$D$+$Seq$(2-$D$)</td>
<td>97.1%</td>
<td>5.1%</td>
<td>5.4%</td>
<td>85.9%</td>
</tr>
</tbody>
</table>

**Remark:** Similarly to the $n$-scan pruning approach [4] used in the dynamic association, one can also apply a sequential $S_0$-dimensional assignment to this static association problem, that is, solve the $S_0$-$D$ assignment on sensors $1 : S_0$, then make a hard decision on sensor 1 and solve the $S_0$-$D$ assignment on $2 : S_0 + 1$, etc. However, compared to $S_0$-$D$+$Seq$(2-$D$), at each step the $S_0$-$D$ assignment (assuming $S_0 > 2$) is more
costly than the 2-D assignment for both cost evaluations and optimizations. In terms of association performance, from the above results we can see that even the S-D assignment is similar to $S_0$-D+Seq(2-D), thus the possible improvement of the sequential $S_0$-D assignment over $S_0$-D+Seq(2-D) is quite limited.

VII. CONCLUSIONS AND FUTURE WORK

This paper presented an efficient data association technique, $S_0$-D+Seq(2-D) algorithm, for 3D passive sensors. The passive-sensor data association is a challenging problem, since the line of sight (LOS) measurements from the passive sensors only provide partial knowledge of a target position. The assignment-based methods have been shown to be very effective for data association, where the data association is first formulated as a multiple dimension assignment (MDA) problem and then solved (suboptimally) by using the Lagrangian-relaxation based S-D algorithm. The bottleneck of the S-D algorithm lies in the cost computation, which consumes about 95%-99% of the CPU times. The number of costs to be evaluated in the MDA problem increases exponentially about 95%-99% of the CPU times. The number of costs to be evaluated in the MDA problem increases exponentially with the number of lists (sensors), which renders the S-D algorithm quite inefficient when a large number of sensors are used. The proposed $S_0$-D+Seq(2-D) algorithm has a total number of costs increasing quadratically with the number of sensors. As a result, it reduces the number of costs drastically in comparison with the S-D algorithm. For 7 sensors the $S_0$-D+Seq(2-D) algorithm achieves a CPU time reduction of 2 orders of magnitude compared to the S-D algorithm. A large-scale problem of 300 hundred targets and 10 sensors was tested and finished in about 6 seconds by using the $S_0$-D+Seq(2-D) algorithm. The CPU times can be further reduced by introducing parallelization to process different clusters concurrently. The S-D and $S_0$-D+Seq(2-D) algorithms have similar association qualities in terms of the four quality metrics: fraction of correct associations (FCA), fraction of missed associations (FMA), fraction of duplicated associations (FDA) and fraction of purity (FP). Future work will consider applying the $S_0$-D+Seq(2-D) algorithm to multiple cameras for 3D object tracking.

REFERENCES


