

Frame Building Algorithm for Electronically Scanned Array Radar

Benjamin J. Slocumb and Michael E. Klusman, III

Numerica Corporation, PO Box 271246, Ft. Collins, CO 82527, USA

ABSTRACT

Advanced tracking algorithms such as multiple frame assignment (MFA) and multiple hypothesis tracking (MHT) require the formation of a “frame of data” to input measurements into the tracking system. A “frame” is a collection of measurements in which a target should appear at most once. For some sensor types, the frame definition is straightforward: all measurements in “one scan” of the antenna across the surveillance area compose a frame of data. However, for electronically scanned array (ESA) radar, the beam pointing is agile and the radar may point the beam in a sequence of overlapping positions. If the data from the sequence of dwells are merged into one frame, duplicate measurements may result from targets in the overlap regions. But restricting each frame to be one dwell has negative consequences because it causes an incomplete representation of closely-spaced targets within each frame. This paper presents a new algorithm for the formation of frames of data for ESA radar systems. The algorithm uses a series of gating tests to determine which radar dwells may be merged together. For overlapping beams, a selection technique is developed that minimizes the number of redundant measurements that appear in any given frame. A summary of tracking performance results attained when using the algorithm is provided.

Keywords: Frame Building, Multiple Frame Assignment, Multiple Hypothesis Tracking, Electronically Scanned Array Radar, Local Tracker

1. INTRODUCTION

One can generally partition the approaches to multiple target tracking into those based on a *single frame* of data and those based on *multiple frames* of data. In the former case, data associations decisions are irrevocable and are made immediately as data arrives. In the latter case, decisions are held in abeyance until more data arrive. The advanced multi-frame methods such as MHT¹ and MFA^{2,3} require a pre-processing algorithm that creates a *frame of data* at the input to the tracking system. The performance of the frame building algorithm has a critical impact on the operation of an MHT/MFA tracker. If measurements are improperly organized into frames, then violations in the multi-hypothesis logic can result and the tracker will perform poorly. For some sensor types, the frame definition is straightforward: all measurements in “one scan” of the antenna across the surveillance area constitute a frame of data. However, for electronically scanned array (ESA) radar, the beam pointing is agile and radar may point the beam in a sequence of overlapping positions. This presents a problem for the framing process because duplicate measurements may result from the merging of data from the sequence of radar dwells. At the same time, the performance of the MHT/MFA tracker is greatly enhanced when measurements for all targets (or at least all closely-spaced targets in a given region) are included in the input frame. Thus, there are competing requirements on the ESA radar frame building algorithm.

Frame building algorithms for sensor applications are based on some basic requirements imposed by the tracker. A *proper frame of data* is a set of measurements over an observation time with the following requirements: (i) each target should appear at most once in the data set; (ii) the frame has an associated sensor field of view (FOV) over the observation time; (iii) each track has an associated score for the frame, and the score is based on three cases: the associated target was seen in the frame, the target was in the FOV and should have been seen but was not, and the target was out of the FOV and should not have been seen. A *maximal proper frame* is one in which a measurement for every target appears, or should appear, in the frame. A basic approach to frame building is to add measurements, as they arrive, to the first data frame (called the *open frame*) until a repeated measurement on a target might exist. At that point, one closes the first frame and shifts all frames. A new open (first) frame is created, and all new data are subsequently added to this frame. Then the process is continued.

For this paper, we focus solely on the process of *constructing* the frames. A separate and additional process can be employed that provides dwell-related track-specific information by comparing existing tracks to the beam point angles

Other author information: Send correspondence to Email: bjslocumb@numerica.us, Telephone: 970-419-8343.

within the frame. Targets that are well outside the field of view (FOV) of dwells within the frame can be scored with a very low P_d . For targets within the FOV, and for which an estimate of the radar cross section is maintained, the expected P_d can be computed. Additionally, if the beam point is at a low elevation angle, then using clutter map information and the transmit power one can compute a clutter P_{fa} value. Since these calculations involve the use of track-specific data, we view this additional process as part of the track scoring process* and not part of the frame builder. However, since beam pointing and dwell information are used in each, the steps are closely related.

We identify a *local tracker* as one that is in-the-loop with the sensor and resource manager, and its tracks drive the radar beam pointing operation. A local tracker is differentiated from a *network tracker* that only receives measurements and does not directly support the sensor operation. When implementing a frame builder for an ESA radar local tracker, several challenges arise. First, an additional *time* constraint exists as to when to close the open frame. Since beam pointing of the radar depends on track state updates, the frame builder must quickly provide new data to keep the system “cycling.” Thus, holding onto data for extended periods of time (for the purpose of optimizing the data frame) is prohibited. Second, in scenarios with closely spaced objects, the ESA radar will often point the beam in a series of overlapping positions as to “cover” all the objects. Thus, if the measurements in these overlapping beams are to be put into the same frame, a mechanism for removing the repeated measurements is required.

Given the challenges just describe, the objective in this paper is to formulate a new approach to frame building for ESA radar sensors. The process of determining when to close the first frame and shift the frames is based on four criteria: (i) comparison of beam pointing angles, (ii) matching of dwell group indices, (iii) checking of the open frame time duration, and (iv) checking the estimated time of the next expected beam report. Once the open frame is closed and shifted, the measurements in the candidate frame are processed to remove/mitigate repeated measurements that may have entered because of overlapping beams. Section 2 describes the ESA radar processing loop and the relationship of the frame builder to other components. Section 3 describes three separate frame building algorithms for the ESA radar application. Section 4 summarizes simulation results attained with a software implementation of the frame building algorithm, and Section 5 summarizes the paper.

2. ESA RADAR LOCAL TRACKER PROCESSING LOOP

The processing loop of an ESA radar local tracker is shown in Figure 1. Each time the “processing command” passes from the resource manager to the sensor to the tracker, one loop is completed. The time to complete this one loop is called a *resource period*, and at the end of the resource period the command is passed back to the resource manager. A typical resource period is very small (fraction of a second). The resource manager may commit multiple radar dwells to the sensor within one cycle; however, once these dwells are committed they cannot be retracted because of changing priorities. Altering of dwell priorities can only happen at the end on one complete resource period.

The data flow within the resource period is shown in Figure 1. Sensor measurement data are passed from the sensor to a report database. A data control function pulls current data from the database and passes it to the tracker. In a given resource period, a collection of dwell reports generated in that period is passed to the tracker. Multiple target detections may exist within each dwell report (i.e., measurements in one radar beam), and multiple dwell reports may be contained within the input data array.

The frame builder receives the input data array and must decide which frame (the current open frame or a new frame) the measurements should be inserted. The decision is based on a set of tests and logic rules. When the tests and rules indicate that measurements should be placed into a new frame, the current open frame is closed and all the frames are shifted. Thus, Frame 1 represents the current open frame, while Frames $\{2, \dots, N\}$ represent the other closed frames that remain within the Window of the multi-frame tracker. Measurements in Frame $N + 1$ are typically discarded.

Using the frames produced by the frame builder, track updates are formed. The updates are reported to a track database used by the resource manager to generate dwell requests; this drives the inner loop of the radar system. The track database requires no measurement data, hence one can periodically update the track states without committing to the closure of Frame 1 (i.e., the concept of partial frames could be used).

*Depending on the implementation, track-specific information such as P_d may already be available from the resource manager and beam scheduler, which optimizes the beam point to acquire measurements on specific targets.

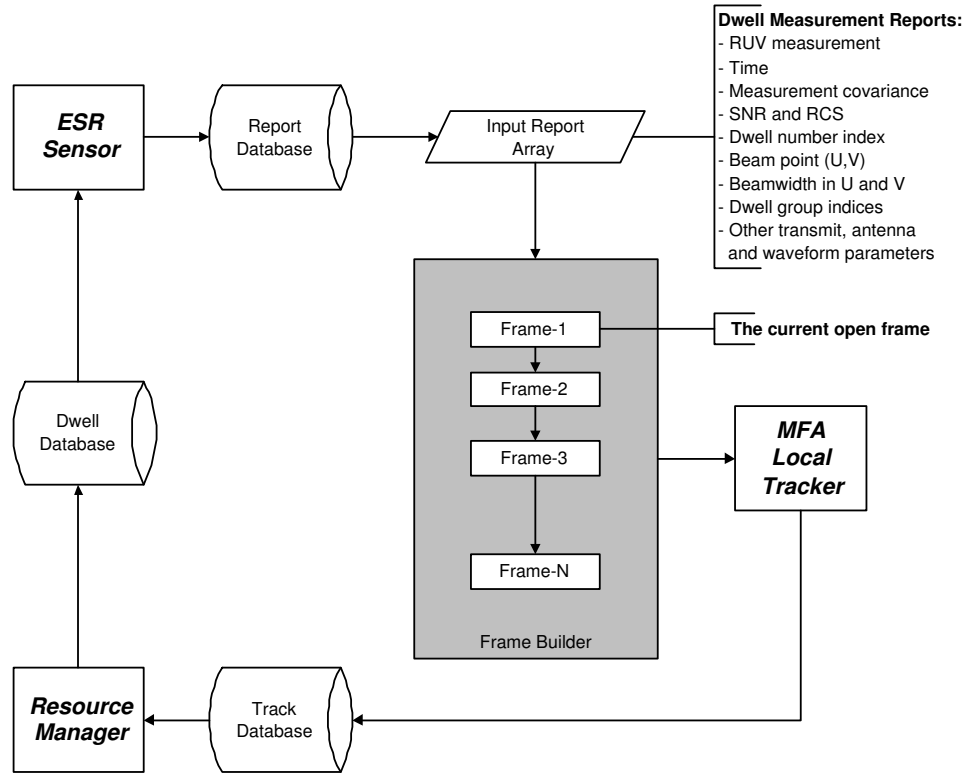


Figure 1. Diagram of the ESA radar local tracker processing loop and the placement of the frame building function within the loop.

The resource manager uses the data in the track database to prioritize targets, schedule radar dwells, and determine the beam point for each dwell. A resource limiter may also be employed; after the resource manager generates a request list of dwells, the limiter may cut that list based on the dwell and rescan time required in the list of dwells.

In Figure 1, note the callout of the contents of the input data message to the frame builder. It includes typical data such as the sine-space RUV measurements, time, and the SNR and RCS. Also included are two components that are useful in frame building: the UV beam point angles, and a data field called *dwell group indices*. The latter defines the local track number indices for targets that were expected in the given radar dwell. As will be discussed later, this index set is useful in the frame building process.

3. LOCAL TRACKER FRAME BUILDING ALGORITHM

In this section, we present three different ESA radar frame building algorithms for application with an multi-frame tracker:

- *Beam is a frame.* Each dwell report (one radar beam) constitutes a frame of data.
- *Framing within a resource period.* All dwell reports within one resource period are considered for merging into the open frame. At the end of the resource period, the open frame is closed and all frames are shifted. Time management constraints within the framing process are not an issue for this version because the frame is automatically closed after each period.
- *Framing across resource periods.* The frame builder can use data from dwells acquired over several resource periods (i.e., the open frame is not necessarily closed after one period). In this version, time management constraints becomes an issue because one cannot hold the current open frame too long – otherwise, the tracker will become deprived of new data and the track state estimates will become stale.

Of the three methods, obviously the first is the simplest in terms of the implementation. However, pursuit of the second and third approaches are motivated by the following problems with the first approach:

- If there are multiple targets in distinctly different parts of the sky, then the radar will be required to constantly switch the beam around to cover the targets. Depending on how the scheduler operates, the radar may not provide enough dwells on one target to fulfill the required number of frames to initiate a track. In other words, the initiation window may become filled with too many frames from other regions and not enough from the new target to allow the tracker to initiate a track. By merging the beams into frames, one is more likely to provide enough frames with measurements to initiate the track. However, there is no guarantee that all targets will be initiated because of dependencies on (i) the number of new targets in different regions and (ii) the behavior of the scheduler. Nevertheless, frame building can significantly reduce the problem.
- In scenarios with multiple closely spaced objects, the ESA radar may be required to put out a sequence of overlapping beams to cover the objects. Here, when using each beam as a frame, we experience several problems. Measurements in separate beams (from different closely-spaced objects) can be allowed to associate to the same track even though they should not. That is, because the measurements are in separate frames, the tracker has no way to prevent the incorrect association of measurements from two or more different objects. Tracks using different combinations of object measurements become initiated and tend to diverge away from the actual targets. The problem is exacerbated by *unbalanced association scores* where some tracks are updated more often than others and hence get better scores. This leads to competition for the measurements, and in some instances the tracks with more updates “steal” the measurements from the tracks with less updates because their states have become better. The problem is mitigated by having one measurement per target within a frame for those targets that are closely spaced. A frame builder that can merge the overlapping beams while eliminating the repeated measurements is required to solve this problem.

3.1. Algorithm 1: dwell is a frame

The first version of the frame building is very simple. In a given radar resource period, one or more dwells of measurements will be passed into the frame builder (i.e., the input array will contain several dwells of data). If there are K dwells in the input array, then K frames of data are formed. The k th frame of data includes all the measurements from the k th dwell. After the measurements are placed into frames, all K frames are passed to the MFA tracker, and the tracker will process these frames sequentially (in time order). When a new input array is passed in on the next resource period, the process is repeated.

3.2. Algorithm 2: frame building within a resource period

In this version of the frame building algorithm, we attempt to merge dwell reports collected during one resource period into a smaller set of frames. Since a resource period is typically of very short duration, time constraints on how long to keep the frame open can be ignored and *only beam pointing information* is used in the framing process. Suppose there are K dwell reports in the input array of dwells and, without loss of generality, they are indexed in time order by $k \in \{1, \dots, K\}$. The goal is to attempt to merge some or all of these reports into N frames, where $1 \leq N \leq K$, such that

$$\begin{aligned} \text{Frame 1 set of dwell indices} &= \mathcal{F}_1 = \{1, 2, \dots, m\}, \text{ for } 1 \leq m; \\ \text{Frame 2 set of dwell indices} &= \mathcal{F}_2 = \{m + 1, m + 2, \dots, n\}, \text{ for } 1 \leq m < n; \\ &\text{continued this way for Frames 3 through } N - 1; \\ \text{Frame } N \text{ set of dwell indices} &= \mathcal{F}_N = \{r + 1, r + 2, \dots, K\}, \text{ for } 1 \leq m < n < r \leq K. \end{aligned}$$

At the completion, all closed frames $\{\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_N\}$ are sent to the MFA tracker for processing. Note that for this implementation we only build frames with time-ordered dwells. For example, we do not allow dwells $\{1, 3, 5\}$ to form one frame and dwells $\{2, 4, 6\}$ to form another. A variation of the algorithm to allow non-time ordered groupings of dwells is a topic for continued research.

The flow diagram for the implementation of Algorithm 2 is shown in Figure 2. The algorithm sequentially processes (in time order) the dwells obtained in the input array for the current resource period. Each dwell is tested using coarse and fine beam point tests to determine if it can be merged into the current open frame. After the framing is complete, then the measurements in each formed frame are processed using a selection algorithm to remove the redundant measurements due to beam overlap. The two parts are described in detail next.

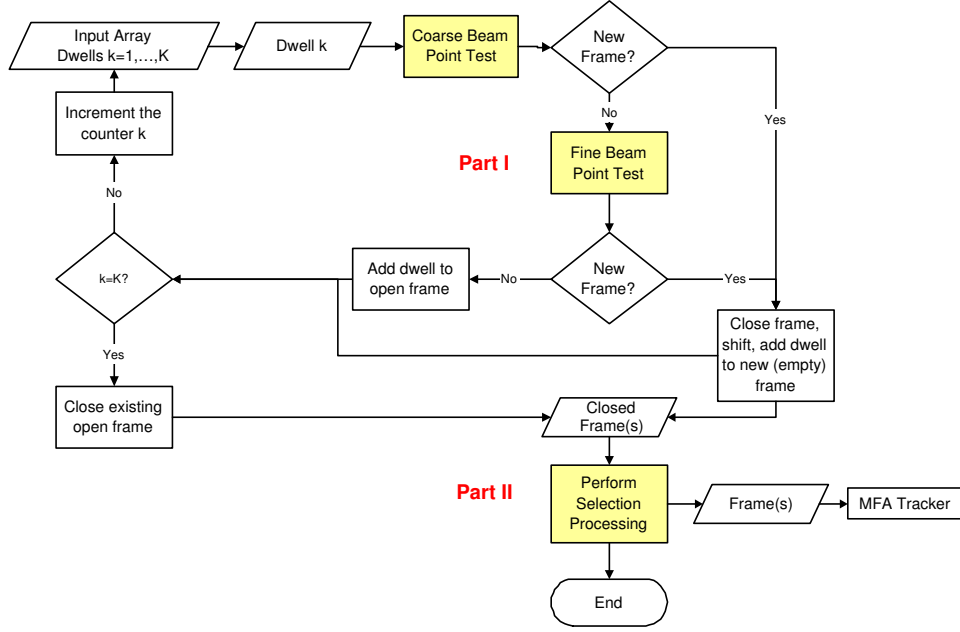


Figure 2. Flow diagram for the Version 2 algorithm.

3.2.1. Part I: frame determination

The framing algorithm operates by stepping through the dwell list and attempting to merged the k th dwell into the open frame. Let K' , where $1 \leq K' < K$, denote number of dwells that have been merged into the open frame. We will next determine if dwell $K' + 1$ can be merged into the open frame. Now define the following parameters:

- r_k^{\min} = the minimum range of the measurements obtained in the k th dwell.
- r_k^{\max} = the maximum range of the measurements obtained in the k th dwell.
- \bar{U}_k = the U pointing angle of the k th dwell.
- \bar{V}_k = the V pointing angle of the k th dwell.
- $U_k^{(i)}$ = the U -coordinate angle of the i th measurement in the k th dwell.
- $V_k^{(i)}$ = the V -coordinate angle of the i th measurement in the k th dwell.
- Θ_{U_k} = the beamwidth in the U dimension on the k th dwell.
- Θ_{V_k} = the beamwidth in the V dimension on the k th dwell.

Test 1: coarse beam overlap test. To assess whether dwell $k = K' + 1$ can be merged into the open frame, we conduct the following gating tests:

$$|\bar{U}_k - \bar{U}_n| < \gamma_U \cdot \Theta_{U_k}, \quad \forall n \in \{1, 2, \dots, K'\} \quad (1)$$

$$|\bar{V}_k - \bar{V}_n| < \gamma_V \cdot \Theta_{V_k}, \quad \forall n \in \{1, 2, \dots, K'\} \quad (2)$$

where $(\gamma_U, \gamma_V) \in [0, 1]$ are angular distance thresholds (used to select a beamwidth fraction). We also apply a range gate test to ensure that the two dwells are within the same spatial region of each other,

$$r_k^{\max} > r_n^{\min} - \gamma_r, \quad \forall n \in \{1, 2, \dots, K'\} \quad (3)$$

$$r_k^{\min} < r_n^{\max} + \gamma_r, \quad \forall n \in \{1, 2, \dots, K'\} \quad (4)$$

where $\gamma_r = v_{\max} \cdot |t_k - t_n| + \epsilon$ determines the maximum expected range difference between targets in the two dwells. Here, $\epsilon = \sqrt{G_0} \cdot \sigma_{r_k}$, with G_0 being the desired Chi-square parameter for one degree of freedom, and

$$\sigma_{r_k} = \max_i \sigma_{r_k^{(i)}} \quad (5)$$

If these inequalities hold for any of the values of n , then we pass the dwell k data to the second test. If not, then dwell k can be directly merged into the open frame.

Test 2: fine beam overlap test. If the coarse gate test passes for any value of n , then we apply the following finer gating test,

$$\left(\frac{(\bar{U}_k - \bar{U}_n)^2}{\Theta_{U_k}^2} + \frac{(\bar{V}_k - \bar{V}_n)^2}{\Theta_{V_k}^2} \right)^{1/2} < 1 - \gamma_\Theta, \quad \forall n \in \{1, 2, \dots, K'\} \quad (6)$$

where $\gamma_\Theta \in [0, 1]$ determines the fraction-of-beamwidth overlap that is allowed for merging. If the inequality holds for any value of n , then **merging is rejected**: we close the open frame, shift the frames, and add the k th dwell to a new (empty) open frame. If the inequality fails for all values of n , then dwell k can be added to the current open frame.

After dwell $k = K' + 1$ is evaluated for merging (it was placed either into the an existing open frame with other dwells or it was placed into a new open frame), we increment the count and evaluate if dwell $k = K' + 2$ can be merged. The process ends after all K dwells are evaluated.

3.2.2. Part II: selection processing

The purpose of this component of the algorithm is to mitigate the number of redundant measurements from overlapping beams within any of the frames formed in the current resource period. The process is accomplished by identifying measurements in overlap regions of overlapping beams, and then applying *selection processing* to extract a subset of measurements.

To test for redundant measurement, we construct a test for each measurement in the k th beam to find if it falls within the region of the n th beam,

$$\left((U_k^{(i)} - \bar{U}_n)^2 + (V_k^{(i)} - \bar{V}_n)^2 \right)^{1/2} < \Theta_n^{\max} + \dot{\theta}_{\max} \cdot |t_k - t_n| + \sqrt{G_0 \text{tr}(\mathbf{R}_k^{(i)})} \quad (7)$$

where $(U_k^{(i)}, V_k^{(i)})$ are the UV components of the i th measurement in the k th dwell, and Θ_{\max} is the maximum scaled beamwidth over both the U and V angles for the beams in the current open frame,

$$\Theta_n^{\max} = \alpha_0 \cdot \max\{\Theta_{U_n}, \Theta_{V_n}\} \quad (8)$$

Here, $\alpha_0 \geq 1$ is a user-defined inflation parameter. The righthand side of (7) includes “expansions” to the beam regions. The term $\dot{\theta}_{\max} \cdot |t_k - t_n|$ accounts for target motion, where the maximum expected angular rate for targets in the open frame is

$$\begin{aligned} \dot{\theta}_{\max} &= v_{\max}/r_{\min} \\ r_{\min} &= \min_{m \in \{1, \dots, K'\}} \{r_m^{\min}\} \end{aligned} \quad (9)$$

Also, $\sqrt{G_0 \text{tr}(\mathbf{R}_k^{(i)})}$ accounts for measurement noise, where the parameter G_0 is a Chi-square parameter with two degrees of freedom and where $\mathbf{R}_k^{(i)}$ is the covariance matrix of the (U, V) component of i th measurement from the k th dwell.

Next we define indexing notation that allows us to identify specific measurements in the overlap region defined by the test in (7). Let I_k represent the set of all measurement indices from dwell k . Next, define the following subsets of I_k :

$$\begin{aligned} I_k(n) &= \text{set of all measurement indices from dwell } k \text{ that pass test (7) relative to dwell } n. \\ I_k(\tilde{n}) &= I_k \setminus I_k(n) = \text{set of all measurement indices from dwell } k \text{ that fail test (7) relative to dwell } n. \end{aligned}$$

In essence, measurements $I_k(n)$ are those measurement indices from dwell k that fall within the beamwidth of the n th beam (corrected for motion and measurement noise). Figure 3a shows an example for two overlapping beams where three regions (A_1, A_2, A_3) are defined,

- Region A_1 : measurements from beam-1 but not within the beam-2 footprint, with indices $I_1(\tilde{2})$.
- Region A_2 : measurements from beam-2 but not within the beam-1 footprint, with indices $I_2(\tilde{1})$.
- Region A_3 : measurements form both beams in the overlap footprint, with indices $I_1(2) \cup I_2(1)$.

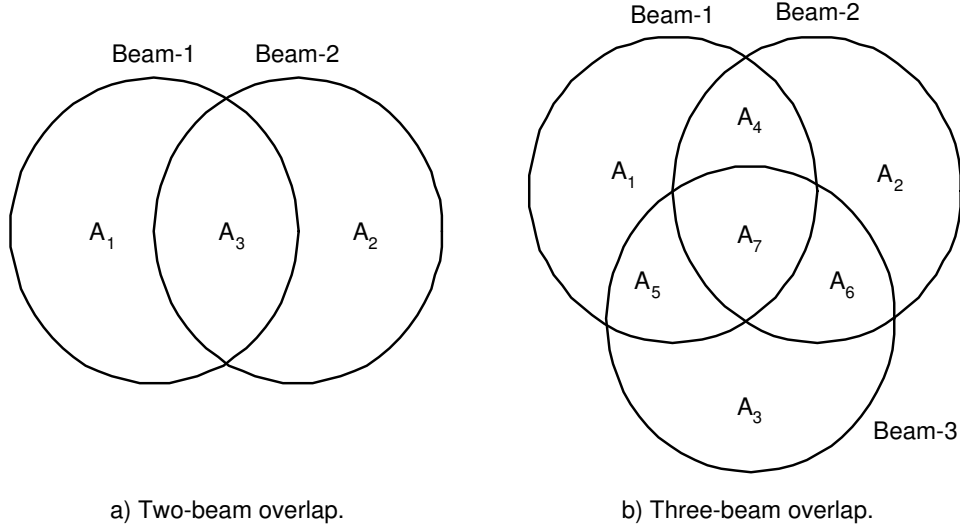


Figure 3. Examples of a) two-beam overlap and b) three-beam overlap.

Next, we extend the index notation to cases where measurements from dwell k fall into multiple other beams. The following examples show the indices for three and four overlapping beams, and the notation extends to an arbitrary M overlapping beams in the obvious way:

$$I_k(m, n) = I_k(m) \cap I_k(n) \quad (11)$$

$$I_k(m, \tilde{n}) = I_k(m) \cap I_k(\tilde{n}) \quad (12)$$

$$I_k(m, n, p) = I_k(m) \cap I_k(n) \cap I_k(p) \quad (13)$$

$$I_k(m, n, \tilde{p}) = I_k(m) \cap I_k(n) \cap I_k(\tilde{p}) \quad (14)$$

Figure 3b shows an example for three overlapping beams where seven regions (A_1, \dots, A_7) are defined. We summarize three of the seven regions here:

Region A_1 : measurements from beam-1 but not within the footprints of 2 or 3, with indices $I_1(\tilde{2}, \tilde{3})$.

Region A_4 : measurements from beams 1 and 2 but not within the beam-3 footprint, with indices $I_1(2, \tilde{3}) \cup I_2(1, \tilde{3})$.

Region A_7 : measurements from all three beams within the overlap footprint, with indices $I_1(2, 3) \cup I_2(1, 3) \cup I_3(1, 2)$.

The concepts generalize to an arbitrary number of overlapping beams, with the number of individual regions growing. Further, it is possible that, depending on the beam size and positions, not all intersections will occur in a given realization.

After the indices of measurements in each of the overlap regions have been identified, the selection processing can be accomplished. Let I be the set of all measurement indices in the frame that are retained after the selection processing. For the two-beam overlap example, the set of retained measurement indices are the following,

$$I = I_{A_1} \cup I_{A_2} \cup I_{A_3} \quad (15)$$

where

$$I_{A_1} = I_1(\tilde{2}) \quad (16)$$

$$I_{A_2} = I_2(\tilde{1}) \quad (17)$$

$$I_{A_3} = \text{select}(I_1(2), I_2(1)) \quad (18)$$

The select operator[†] works as follows:

- If $|I_k(n)| > |I_n(k)|$, then $I_{A_3} = I_k(n)$ and measurements $I_n(k)$ are discarded.
- Elseif $|I_k(n)| < |I_n(k)|$, then $I_{A_3} = I_n(k)$ and measurements $I_k(n)$ are discarded.
- Otherwise, the same number of measurements from each of the two beams falls into the overlap region. When this happens, we fall back to a second selection criteria. We set I_{A_3} to the set of indices for which the sum of the SNR over all measurements is the greatest. If the sum of the SNR over the two beams is identical (an unrealistic outcome), then we would arbitrarily select either of the set of measurements.

The selection operator keeps the measurements from the beam with the most measurement in the selection region. The reason for doing this is that we want to retain as many of the measurements as possible that potentially represent targets, but not the redundant ones. The fall-back criteria of the maximum sum of the SNR values gives the set with the best overall quality of measurements.

For the case of three beams, the retained measurements are determined from the seven regions defined in Figure 3:

$$I = \bigcup_{n=1}^7 I_{A_n} \quad (19)$$

where

$$I_{A_1} = I_1(\tilde{2}, \tilde{3}) \quad (20)$$

$$I_{A_4} = \text{select}(I_1(2, \tilde{3}), I_2(1, \tilde{3})) \quad (21)$$

$$I_{A_7} = \text{select}(I_1(2, 3), I_2(1, 3), I_3(1, 2)) \quad (22)$$

The other indices follow similarly from these three. The select operator extends in the obvious way for I_{A_7} where the selection is conducted over three sets: retain the set out of the three with the most measurements in the overlap region, and in the case of a tie use the set with the highest accumulated SNR.

The implementation of selection algorithm for cases of more than three overlapping beams follows similarly to the two described cases. The output of the selection algorithm is a frame of measurements with indices I , where the set has been determined via equation (19), or a similar one for a larger number of overlapping beams.

3.3. Algorithm 3: frame building across multiple resource periods

The third implementation of the algorithm is more advanced in that it seeks to merged dwell reports received over multiple resource periods. That is, the frames can be formed using dwells received in different resource periods. To implement this version, *time management constraints* are integrated into the Version 2 algorithm.

Figure 4 shows a diagram of the Version 3 algorithm. After the array of dwell reports is received for the current resource period, each of the dwells is processed in sequential order. The first check on the input dwell is using a time-management test (Part I). After this test, a beam point or dwell group test (Part II) is applied. After all of the dwells in the input array are processed, a second time management test (Part III) is applied. Finally, when each frame is closed and sent to the tracker, then selection processing (Part IV) is applied. Each of these parts of the Version 3 algorithm is described in detail next.

3.3.1. Part I: time-management frame build test-1

The first test that is implemented in the Version 3 algorithm is a time-based test. We check to see if the time of the k th dwell being considered for merge with the current open frame exceeds an upper limit relative to the minimum time of the existing dwells in the open frame,

$$t_k > t_{\min} + T_{\text{frame}}^{\max} \quad (23)$$

where

$$t_{\min} = \min_{m \in \{1, 2, \dots, K'\}} \{t_m\} \quad (24)$$

[†]Note that $|I|$ is the cardinality of set I , i.e., the number of the items in set I .

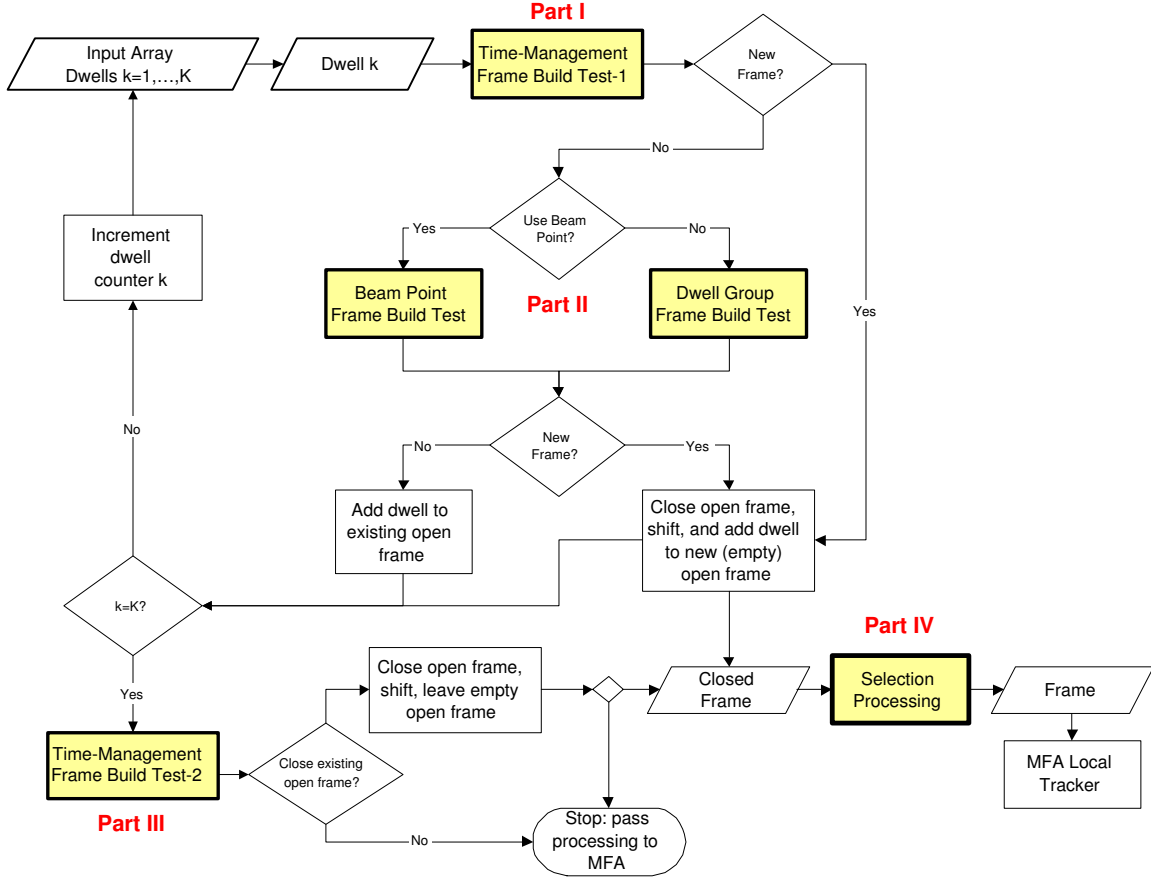


Figure 4. Flow diagram of the Version 3 Frame Building Algorithm.

The value of $T_{\text{frame}}^{\text{max}}$ is a user-defined parameter. If the inequality in (23) holds, then the current open frame is immediately closed, and the k th dwell is inserted into a new open frame.

If the inequality (23) does not hold, then we proceed to the next part of the algorithm to consider whether the k th dwell can be added to the current open frame.

3.3.2. Part II: beam point and dwell group frame build tests

The second test implemented in the Version 3 algorithm has two components: a *beam point test* (like the Version 2 algorithm), and a *dwell group test*. Only one of the two components is executed for a given dwell, and the selected one is based on a time test. If the present dwell is significantly different in time from other dwells in the open frame, then the beam-point test is deemed unreliable and a dwell group test must be used.

To determine which of the components to use for a given dwell, we first compute some parameters. A *beam point allowance time* is computed based on the worst-case angular target motion relative to the beamwidth,

$$T_{bpa} = \frac{\Theta_{\text{max}}}{\dot{\theta}_{\text{max}}} \quad (25)$$

where is

$$\Theta_{\text{max}} = \max_{n \in \{1, \dots, K'\}} \Theta_n^{\text{max}} \quad (26)$$

and Θ_n^{\max} is defined in (8). The maximum angular velocity $\dot{\theta}_{\max}$ is defined in (9). Upper and lower bounds are then applied to this value relative to the duration of the radar resource period T_{rp} ,

$$T_{bpa} = \max\{\min\{T_{bpa}, n_2 \cdot T_{rp}\}, n_1 \cdot T_{rp}\} \quad (27)$$

where (n_1, n_2) are user-selected integers. If the following condition holds,

$$t_k < \min_{m \in \{1, \dots, K'\}} \{t_m\} + T_{bpa} \quad (28)$$

then we decide that target motion is not significant relative to the antenna point angle and that the beam point test can be applied. However, if (28) does not hold, then we apply the dwell group test instead (see Figure 4).

Beam point test. When it is determined using (28) that target motion is not significant, then we apply the beam point test. The test is implemented using first the coarse angle gate test and then the fine angle gate test shown in the Version 2 algorithm. If the amount of overlap between the k th dwell and all of other dwells in the current open frame is not significant, then the k th dwell is added to the current open frame.

Dwell group test. When it is determined using (28) that the beam point test cannot be used, we use a test based on the dwell group indices associated with each dwell. The concept with this approach is to determine if any of the dwell group indices associated with dwell k match any of the indices associated with the current open frame. If they match, then one can expect that a measurement from a target in the current dwell already exists in the open frame; therefore, the open frame should be closed and the a new open frame with the current dwell should be started. If not (i.e., there is no match), then it should be acceptable to merge dwell k into the current open frame.

Let $\iota_k^{(i)}$ be the i th dwell group index associated with the k th dwell, where $i \in \{1, 2, \dots, N_{DG_k}\}$ and N_{DG_k} is the total number of dwell group indices associated with the k th dwell. If K' frames have been added to the current open frame, then the set of all dwell group indices associated with the open frame is given by

$$\Upsilon_{1:K'} = \bigcup_{m=1}^{K'} \bigcup_{i=1}^{N_{DG_m}} \iota_m^{(i)} \quad (29)$$

Also define the following set of dwell group indices for the current dwell $k = K' + 1$,

$$\Upsilon_k = \bigcup_{i=1}^{N_{DG_k}} \iota_k^{(i)} \quad (30)$$

Now we compute the intersection of these two sets,

$$\mathcal{I} = \Upsilon_{1:K'} \cap \Upsilon_k \quad (31)$$

If $\mathcal{I} = \emptyset$, then the current beam can be merged into the open frame. Otherwise, the open frame must be closed, the frames shifted, and the current dwell inserted into a new open frame. An alternate rule is to compute $|\mathcal{I}|/|\Upsilon_{1:K'}|$, and if this ratio is below a specified threshold then the current dwell can be merged (i.e., we accept duplicate measurements up to a specific percentage). However, this is a less restrictive rule and we prefer to implement the rule that requires $\mathcal{I} = \emptyset$ for merging.

3.3.3. Part III: time-management frame build test-2

After all the dwells in the current input array have been processed, a non-empty open frame may remain. In this case a second time-management test is applied. This test is needed only when the frame builder is running within a synchronous (single-processor) simulation environment; in the event that the frame builder is running on an asynchronous computer within an ESA radar platform where self-monitoring of the system time is possible, then we expect that this test can be ignored.

The need for the second time test can be explained as follows. To prevent problems where data within the open frame grows stale, the frame builder algorithm needs to monitor the system time. When the time window in which the frame remains open becomes too large, the algorithm can simply close the frame and send the data to the tracker. But this is a problem when the algorithm has no ability to monitor the system time[‡]. If the radar sensor is running with a very long update period, it could be several seconds before a new dwell of data is provided to the frame builder (and therefore the “command” is passed to the frame building function). In that case, the data that has remained in the open frame will have become stale (i.e., it would have been best to close the frame and use the data to update tracks, thus keeping the track states more up to date). To prevent this problem, the frame builder must *estimate* the next expected time it will receive a dwell of data. If that estimated time is too far into the future, then after all dwells in the input array from the current resource period have been processed it should close the open frame and process the data. If the estimated time to receive new data is soon enough, then the open frame can remain without being closed.

The implementation of the second time test can be described as follows. Suppose there are N tracks in the track database. Let $t_{\text{update}}^{(n)}$ represent the last update time of the n th track. Let $\hat{T}_{\text{revisit}}^{(n)}$ represent the estimated revisit time of the n th track (derived by the resource manager). Using these time parameters, the next expected dwell for the n th track will be received at time

$$t_{\text{next}}^{(n)} = t_{\text{update}}^{(n)} + \hat{T}_{\text{revisit}}^{(n)} \quad (32)$$

From this, we compute the expected time the next dwell report will be received,

$$t_{\text{next}}^{\min} = \min_{n \in \{1, \dots, N\}} \{t_{\text{next}}^{(n)}\} \quad (33)$$

Now let t_k be the time of the k th dwell that is in the current open frame, with $k \in \{1, 2, \dots, K'\}$, and K' is the total number of dwells being held in the open frame. Let t_1 be the minimum time[§] in the open frame. If T_{frame}^{\max} is the maximum amount of time we allow a frame to remain open, then if

$$t_{\text{next}}^{\min} > t_1 + T_{\text{frame}}^{\max} \quad (34)$$

we decide that the frame will remain open too long before new data will arrive and we close the current open frame and send it to the MFA tracker.

3.3.4. Part IV: selection processing

The last part of the Version 3 algorithm is selection processing, where the measurements in the closed frame are processed to remove potential duplicates due to overlapping beams. The algorithm implementation is the same as the Part II component of the Version 2 algorithm; refer to that description for details.

4. RESULTS SUMMARY

The ESA radar frame building algorithm has been implemented in software and evaluated in a target tracking simulator. The software implementation allows for each of the three versions of the frame builder described in Section 3 to be selected. To characterize the frame building performance, we provide here a qualitative description of the MFA tracker metrics when using frame builder Algorithm 2 versus Algorithm 1 (the results for Algorithms 2 and 3 were similar). The evaluation scenario involved a number of closely spaced objects, thus representing challenging conditions.

- *Tracker completeness.* This metric assesses whether or not the tracker has one track per object in the surveillance environment. Using Algorithm 1 the tracker significantly under-represented the object count because too few tracks were initiated given the incomplete frames. Using Algorithm 2, complete frames were received and the optimal tracker completeness level was achieved.

[‡]This is a particular problem when the algorithm is running in a single-processor event-based simulation environment.

[§]Assuming the dwells are indexed in time order, then t_1 is the minimum time. If time-order indexing is not used, then we would need to compute the minimum time over the set.

- *Spurious and redundant tracks.* These metrics assess whether the tracker has created extra tracks that either associate to an object that is already being tracked (redundant) or to no object at all (spurious). When using Algorithm 1, the tracker produces some redundant and spurious tracks because the incomplete frames allow the initiation algorithm to start tracks using measurements from multiple objects. The tracks tend to diverge and are eventually dropped thus making the track picture look messy. When using Algorithm 2, the redundant/spurious tracks are not initiated and the track picture looks clean.
- *Track accuracy.* The track accuracy performance is slightly improved when using Algorithm 2 instead of Algorithm 1. The improvements are realized because the data association performance is better, thus tracks are updated with appropriate measurements at a higher rate.

In summary, we found the tracking performance to be far superior when using the multi-dwell frame building algorithm as compared to a single beam-is-a-frame algorithm. Significant benefits were realized in a CSO environment.

5. FRAME BUILDING ALGORITHM SUMMARY

A specialized frame building algorithm is required to integrate an MHT or MFA tracker as *local tracker* within ESA radars. The unique challenge for ESA radar frame building is the potential for beam overlap and therefore the duplication of measurements from the same target. This paper has presented techniques for building frames based on beam point, beamwidth, dwell group index, and time of dwell information. Three solutions to the framing problem have been developed. The first is a very simple solution that considers each dwell as an independent frame. This solution has several drawbacks, as discussed at the beginning of Section 3. The second version allows for frame building using dwells from within a single radar resource period. The third version of the algorithm allows for frame building across resource periods. An implementation of the algorithm in software has been achieved, and the simulation results indicate substantial improvements in tracking performance when using the second and third frame building algorithms.

REFERENCES

1. S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*, Artech House, Norwood, MA, 1999.
2. A. B. Poore, "Multidimensional assignment formulation of data association problems arising from multitarget tracking and multisensor data fusion," *Computational Optimization and Applications* **3**, pp. 27–57, 1994.
3. A. B. Poore and A. J. R. III, "A new Lagrangian relaxation based algorithm for a class of multidimensional assignment problems," *Computational Optimization and Applications* **8**, pp. 129–150, 1997.