4. Implementation Details

4.1. Group Tracker

Tracking algorithm is implemented as a library. Application task creates an algorithm instance with configuration parameters that describe sensor, scenery, and behavior of radar targets. Algorithm is called once per frame from Application Task context. It is possible to create multiple instances of group tracker.
The figure below explains the steps algorithm goes during each frame call. Algorithm inputs measurement data in Polar coordinates (range, angle, Doppler), and tracks objects in Cartesian space. Therefore we use Extended Kalman Filter (EKF) process.

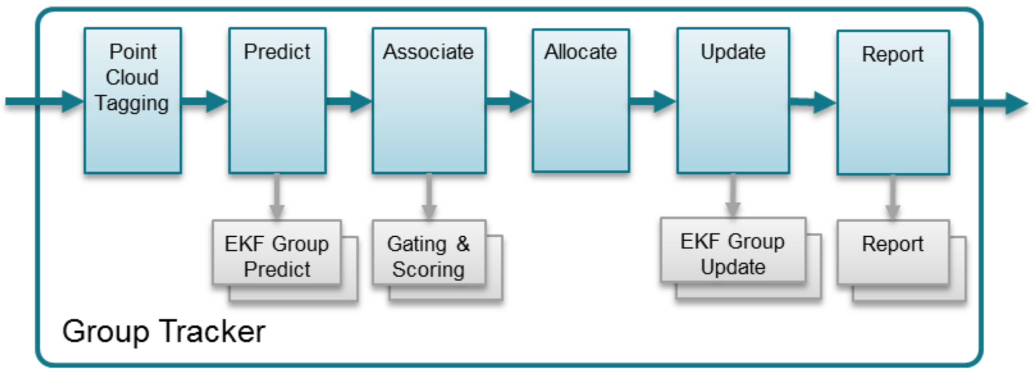

Figure 8. Group Tracking Algorithm

Point cloud input is first tagged based on scene boundaries. Some points may be get tagged as "outside the boundaries", and will be ignored in association and allocation processes.
Predict function estimates tracking group centroid for time n based on state and process covariance matrices estimated at time n-1. We compute a-priori state and error covariance estimations for each trackable object. At this step we also compute measurement vector estimations.
Association function allows each tracking unit to indicate whether each measurement point is "close enough" (gating), and if it is, to provide the bidding value (scoring). Point is assigned to a highest bidder.
Points not assigned, are going through an Allocate function. During the Allocation process, points are first joined into a sets based on their proximity in measurement coordinates. Each set becomes a candidate for allocation decision. It has to pass multiple tests to become a new track. Once passed, the new tracking unit is allocated.
During Update step, tracks are updated based on the set of associated points. We compute the innovation, Kalman gain, and a-posteriori state vector and error covariance. In addition to classic EKF, the error covariance calculation includes group dispersion in measurement noise covariance matrix.
The Report function queries each tracking unit and produces the algorithm output.

4.2. Configuration Parameters

The configuration parameters are used to configure Tracking algorithm. They shall be adjusted to match customer usage case based on particular scenery and targets characteristics. Parameters are divided into mandatory, and optional (advanced). Mandatory parameters are described below.

4.2.1. Mandatory Configuration Parameters

Table 1. Mandatory Configuration Parameters

Parameter	Default, TM	Default, PC	Dim	Description
maxNumPoints	250	250	-	Maximum Number of Detection Points per frame
maxNumTracks	20	20	-	Maximum Number of Targets to track at any given time
stateTrackingVectorType	2DA	2DA	-	2D={x,y,vx,vy} **2DA={x,y,vx,vy,ax,ay} Currently, this is the only supported option** 3D={x,y,z,vx,vy,vz} 3DA={x,y,z,vx,vy,vz,ax,ay,az}
initialRadialVelocity	-20	0	m/s	Expected target radial velocity at the moment of detection
maxRadialVelocity	N/A	N/A	m/s	Maximum absolute radial velocity reported by sensor. This shall match sensor chirp configuration
radialVelocityResolution	N/A	N/A	m/s	Minimal non-zero radial velocity reported by sensor. This shall match sensor chirp configuration
maxAccelerationX	0	2	m/s^2	Maximum targets acceleration. Used to compute processing noise matrix
maxAccelerationY	20	2	m/s^2	Maximum targets acceleration. Used to compute processing noise matrix
deltaT	N/A	N/A	ms	Frame Rate. This shall match sensor chirp configuration
verbosityLevel	NONE	NONE	-	A bit mask representing levels of verbosity: NONE | WARNING | DEBUG | ASSOCIATION DEBUG | GATE_DEBUG | MATRIX DEBUG

4.2.2. Advanced parameters

Advanced parameters are divided into few sets. Each set can be omitted, and defaults will be used by an algorithm. Customer is expected to modify needed parameters to achieve better performance.

Scenery Parameters

This set of parameters describes the scenery. It allows user to configure the tracker with expected boundaries, and areas of static behavior. User can define up to 2 boundary boxes, and up to 2 static boxes. Boxes coordinates are in meters, sensor is assumed at (0, 0) of Cartesian (X, Y) space.

Table 2. Scenery Parameters

Parameter	Default, TM	Default, PC		Description
numBoundaryBoxes	1U	0	-	Number of boundary boxes defined. Points outside boundary box will be ignored
boundaryBox[2]	{-1.f, 12.f,15.f,75.f},	{0, 0, 0, 0},	m	(left, right, bottom, top}

	{0, 0, 0, 0}	{0, 0, 0, 0}		
numStaticBoxes	1U	0	-	Number of static boxes defined. Targets inside static box are allowed to persist as static
staticBox[2]	{0.f, 11.f,19.f,50.f}, {0, 0, 0, 0}	{0, 0, 0, 0}, {0, 0, 0, 0}	m	(left, right, bottom, top}

Measurement Standard Deviation Parameters

This set of parameters is used to estimate standard deviation of the reflection point measurements.

Table 3. Measurements Standard Deviation Parameters

Parameter	Default, TM	Default, PC		Description
LengthStd	1/3.46	1/3.46	m	Expected standard deviation of measurements in target length dimension
WidthStd	1/3.46	1/3.46	m	Expected standard deviation of measurements in target width dimension
DopplerStd	1.f	1.f	m/s	Expected standard deviation of measurements of target radial velocity

Typically, the uniform distribution of reflection points across target dimensions can be assumed. In such cases, standard deviation can be computed as below.

$$\boldsymbol{\sigma = \frac{b-a}{\sqrt{12}}}$$

For example, for the targets of 1m wide, standard deviation can be configured as $\frac{1}{\sqrt{12}}$.

Allocation Parameters

The reflection points reported in point cloud are associated with existing tracking instances. Points that don't get associated are subjects for the allocation decision. Each candidate point is clustered into an allocation set. To join the set, each point needs to be within maxDistance and maxVelThre from the set's centroid. Once the set is formed, it has to have more than setPointsThre members, and pass the minimal velocity and SNR thresholds.

Table 4. Allocation Parameters

Parameter	Default, TM	Default, PC	Dim	Description
setSNRThre	-1.f	150.f	-	Minimum total SNR for the allocation set, linear sum of power ratios
setSNRObscThre	-1.f	250.f	-	Minimum total SNR for the allocation set, linear sum of power ratios, when obscured by another target
setVelThre	1.f	0.1f	m/s	Minimum radial velocity of the allocation set centroid
setPointsThre	3U	5U	-	Minimum number of points in the allocation set
maxDistanceThre	4.f	1.f	m^2	Maximum squared distance between candidate and

				centroid to be part of the allocation set
maxVelThre	2.f	2.f	m/s	Maximum velocity difference between candidate and centroid to be part of the allocation set

State Transition Parameters

Each tracking instance can be in either FREE, DETECT, or ACTIVE state. Once per frame the instance can get HIT (have non-zero points associated to a target instance) or MISS (no points associated) event.

Once in FREE state, the transition to DETECT state is made by the allocation decision. See section **Error! Reference source not found.** for the allocation decision configuration parameters.
Once in DETECT state, we use det2active threshold for the number of consecutive hits to transition to ACTIVE state, or det2free threshold of number of consecutive misses to transition back to FREE state.
Once in ACTIVE state, the handling of the MISS (no points associated) is as follow:

- If the target is in the "static zone" AND the target motion model is close to static then the assumption is made that the reason we don't have detection is because we removed them as "static clutter". In this case, we increment the miss count, and use static2free threshold to "extend the life expectation" of the static targets.
- If the target is outside the static zone, then the assumption is made that the reason we didn't get the points is that target is exiting. In this case, we use exit2free threshold to quickly free the exiting targets.
- Otherwise, (meaning target is in the "static zone", but has non-zero motion in radial projection) we assume that the reason of not having detections is that target got obscured by other targets. In this case, we continue target motion according to the model, and use active2free threshold.

Table 5. State Transitions Parameters

Parameter	Default, TM	Default, PC	Dim	Name
det2activeThre	3U	10U	-	In DETECT state; how many consecutive HIT events needed to transition to ACTIVE state
det2freeThre	3U	5U	-	In DETECT state; how many consecutive MISS events needed to transition to FREE state
active2freeThre	5U	10U	-	In ACTIVE state and NORMAL condition; how many consecutive MISS events needed to transition to FREE state
static2freeThre	5U	100U	-	In ACTIVE state and STATIC condition; how many consecutive MISS events needed to transition to FREE state
exit2freeThre	5U	5U	-	In ACTIVE state and EXIT condition; how many consecutive MISS events needed to transition to FREE state

Gating Parameters

Gating parameters set is used in association process to provide a boundary for the points that can be associated with a given track. These parameters are target-specific.

Table 6. Gating Function Parameters

Parameter	Default, TM	Default, PC	Dim	Description
Volume	16.f	2.f		Gating volume
LengthLimit	12.f	2.f	m	Gating Limit in length
WidthLimit	8.f	2.f	m	Gating Limit in width
VelocityLimit	0.f	0.f	m/s	Gating Limit in velocity (0 – no limit)

Gating volume can be estimated as the volume of the Ellipsoid, computed as

$$V = \frac{4\pi}{3} abc$$

where a, b, and c are the expected target dimensions in range (m), angle (rad), and doppler (m/s).

For example, consider a vehicle as a radar target. For the vehicle center, we could want to reach +/- 4m in range ($a = 8$), +/- 3 degree in azimuth ($b = 6\pi/180$), and +/- 2m/s in radial velocity ($c = 4$), resulting in volume about 16.

In addition to setting the volume of the gating ellipsoid, the limits can be imposed to protect ellipsoid from overstretching. The limits are the function of the geometry and motion of the expected targets. For example, setting WidthLimit to 8m will not allow the gating function to stretch beyond 8m in width.