3D People Counting Demo Configuration Parameters

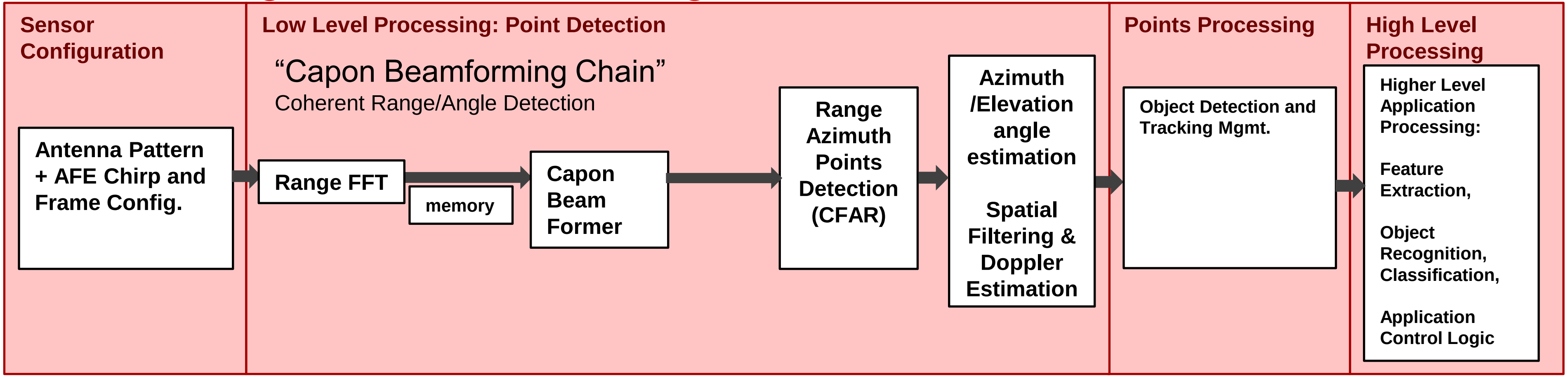
Configuration parameters set in the .cfg file

- To the right is an example of the default .cfg file for the 3D people counting demo.
- Each line represents a command line interface message that configures several parameters.
- The spreadsheet file 68xxChirpParams_Classifierdemo.xlsx provides some information on each value on each command line.
- This document elaborates specifically on:
  - The detection algorithm configuration parameters specified by CFAR and angle estimation configuration commands, highlighted in RED. **The values we choose to use in these set of configurations are the best empirical values we obtained from testing in open office/hallway/living room environment**.
  - The antenna pattern and board related parameters, highlighted in GREEN.
- The group tracker configuration parameters specified by the tracking commands, highlighted in BLUE, are elaborated in the customization guide document.

Configuration File: mmw_classifierDemo_68xx.cfg

```
sensorStop
flushCfg
dfeDataOutputMode 1
channelCfg 15 7 0
adcCfg 2 1
adcbufCfg -1 0 1 1 1
lowPower 0 0
profileCfg 0 60.75 30.00 25.00 59.10 1184274 0 54.7125 1   96 2950.00 0 0 36
chirpCfg 0 0 0 0 0 0 0 1
chirpCfg 1 1 0 0 0 0 0 2
chirpCfg 2 2 0 0 0 0 0 4
frameCfg 0 2  96  0 55.00 1 0
dynamicRACfarCfg -1 4 4 2 4 8 16 4 4 4.00 4.50 0.50 1 1
staticRACfarCfg -1 4 4 2 4 8 16 4 6 8.00 13.00 0.30 0 0
dynamicRangeAngleCfg -1 0.75 0.0010 1 0
dynamic2DAngleCfg -1 3 0.0300 1 0 1 0.50 0.85 8.00
staticRangeAngleCfg -1 1 8 2
antGeometry0 0 -1 -2 -3 -2 -3 -4 -5 -4 -5 -6 -7
antGeometry1 -1 -1 -1 -1 0 0 0 0 -1 -1 -1 -1
antPhaseRot 1 1 1 1 1 1 1 1 1 1 1 1
fovCfg -1 70.0 20.0
compRangeBiasAndRxChanPhase 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
staticBoundaryBox -4.5 4.5 1 5.5 -1.5 1.5
boundaryBox -5 5 0.5 6 -2 2
gatingParam 3 1.5 1.5 2 0
stateParam 3 3 10 40 5 600
allocationParam 400 800 0.1 20 0.5 20
maxAcceleration 0.1 0.1 0.1
trackingCfg 1 2 1000 20 67 105 100 90
sensorStart
```

Processing Chain Elements High Level Overview



1. The chain described above is a high level view of the functional split of the software.
2. The chirp configuration parameters dictate the analog front end configuration, the frame configuration, and the Range FFT size.
3. The capon beamformer works on the per frame antenna data to estimate the virtual antenna covariance matrices for each range bin. This behavior is dictated by the antenna configuration, and the chirp and frame configurations.
4. The range azimuth heat map is then formed and a range-azimuth peak detection algorithm is applied. It is here that parameters specified by `cfarCfg` and `doaCfg` affect the range-azimuth detection behavior of the chain.
5. After peaks are detected in the range and azimuth heat map, then, for each detected point, further processing is done to estimate the azimuth/elevation angle, and then extract the Doppler spectrum, which is then searched for Doppler spectrum peaks, further refining the list of detected points. `doaCfg` also contains parameters that configure this behavior.
6. Step 4 and 5 will be done for both dynamic scene (static information removed) and static scene separately.
7. After the Doppler search/estimation is done, then, for the given radar frame, there is a list of detected points. Each point has a range, angle of arrival, and a doppler value. This is the so called "point cloud", which is used by the group tracker for object detection and tracking. The configuration of the group tracker is specified in other documents, but the behavior of the initial object detection, and the subsequent tracking, and deallocation of the tracks, can be controlled by those parameters specified by the `trackingCfg` command.

Dynamic Scene CFAR configuration parameters (1)

dynamicRACfarCfg													
<cfarMethod>	<cfarDiscardLeft1>	<cfarDiscardRight1>	<cfarDiscardLeft2>	<cfarDiscardRight2>	<refWinSize1>	<refWinSize2>	<guardWinSize1>	<guardWinSize2>	<rangeThre>	<angleThre>	<sidelobThr>	<Enable2ndPass>	<dynamicFlag>
6	4	4	2	4	8	16	4	4	4	4.5	0.5	1	1
6: 2-pass range-azimuth CFAR	samples discarded on the left (range)	samples discarded on the right(range)	amples discarded on the left(angle	samples discarded on the right(angle)	range ref win size	Angle ref win size	range ref guard size	Angle ref guard size	treshold	treshold	sidelobThr	Enable2ndPass	dynamicFlag

- cfarMethod:
  This is the CFAR method hardcoded for 3D people counting demo. It is a 2-pass CFAR on range-angle heatmap, with the first pass in Range domain, and second pass in Azimuth domain to confirm the detections from the first pass. Both passes use CFAR-CASO (cell average, smaller of). This parameter CANNOT BE CHANGED.
- Search Boundary Parameters: cfarDiscardLeft1, cfarDiscardRight1, cfarDiscardLeft2, and cfarDiscardRight2
  These parameters determine the edges of the CFAR search domain in both the range and the azimuth dimensions.
  - cfarDiscardLeft1:
    This is the number of samples on the beginning side that will NOT be included in CFAR search in the first pass (Range domain). Depending on the chirp configuration and derived inter-bin range resolution, this setting dictate the range near the sensor that will be excluded from the detection, typically due to the close range high noise. For the example of default iwr6843 chirp configuration, value of 4 in this field implies the 25cm within the sensor will not have any detection from the CFAR.
  - cfarDiscardRight1:
    This is the number of samples on the ending side that will NOT be included in CFAR search in the first pass (Range domain). Depending on the chirp configuration and derived inter-bin range resolution, this setting dictate the range furthest away from the sensor that will be excluded from the detection, typically range boundary of the scene. For the example of default iwr6843 chirp configuration, value of 4 in this field implies the furthest 25cm from the sensor will not have any detection from the CFAR.
  - cfarDiscardLeft2 and cfarDiscardRight2:
    These 2 parameters are similar to cfarDiscardLeft1 and cfarDiscardRight1, but rather in the Azimuth domain which is the second pass of the CFAR. This set of parameters is somewhat redundant with another parameter in the `doaCfg` settings called `doaSearchRange`. They both effectively do the same thing and so they should be configured to be consistent with each other.

Dynamic Scene CFAR configuration parameters (2)

dynamicRACfarCfg													
<cfarMethod>	<cfarDiscardLeft1>	<cfarDiscardRight1>	<cfarDiscardLeft2>	<cfarDiscardRight2>	<refWinSize1>	<refWinSize2>	<guardWinSize1>	<guardWinSize2>	<rangeThre>	<angleThre>	<sidelobThr>	<Enable2ndPass>	<dynamicFlag>
6	4	4	2	4	8	16	4	4	4	4.5	0.5	1	1
6: 2-pass range-azimuth CFAR	samples discarded on the left (range)	samples discarded on the right(range)	amples discarded on the left(angle	samples discarded on the right(angle)	range ref win size	Angle ref win size	range ref guard size	Angle ref guard size	treshold	treshold	sidelobThr	Enable2ndPass	dynamicFlag

- Range-Azimuth CFAR search window parameters: refWinSize1 and refWinSize2, guardWinSize1, guardWinSize2
  CFAR search, in both range and azimuth dimensions, uses a sliding window to calculate the local noise floor, for comparison to the cell under test (CUT). These parameters determine the size and shape of the window for the range domain and the azimuth domain.
  - refWinSize1 and refWinSize2:
    This is the CFAR search window size in number of samples, for the first (Range Domain) and second (angle domain) pass. The value of 16 is empirically best choice and user may want to use this as default setting.
  - guardWinSize1:
    This is the CFAR guard window size in number of samples in Range domain. When calculating the detection threshold for the cell under test (CUT), the left and right guardWinSize1 of samples will be excluded from noise accumulation. If we have a target with an area of 0.5m^2 reflecting radar energy, then we don't want samples in those area being counted as noise samples to raise the detection threshold. This way there will be richer point cloud detection out of CFAR. For the example of the default iwr6843 chirp configuration, value of 3 in this field implies the 19cm on each side of the CFAR CUT that should not be counted into noise power for CFAR detection. User should adjust the setting based on the chirp configuration and derived inter-bin resolution, as well as the typical target size within the scene. These settings have been already adjusted for the desired effect on people as intended targets with the provided chirp configuration.
  - guardWinSize2:
    Similar to guardWinSize1, this is the CFAR guard window size in number of samples in Azimuth domain. Again, this has been empirically adjusted for the desired effect on people as the intended targets of interest with the provided chirp configuration.

Dynamic Scene CFAR configuration parameters (3)

dynamicRACfarCfg													
<cfarMethod>	<cfarDiscardLeft1>	<cfarDiscardRight1>	<cfarDiscardLeft2>	<cfarDiscardRight2>	<refWinSize1>	<refWinSize2>	<guardWinSize1>	<guardWinSize2>	<rangeThre>	<angleThre>	<sidelobThr>	<Enable2ndPass>	<dynamicFlag>
6	4	4	2	4	8	16	4	4	4	4.5	0.5	1	1
6: 2-pass range-azimuth CFAR	samples discarded on the left (range)	samples discarded on the right(range)	amples discarded on the left(angle	samples discarded on the right(angle)	range ref win size	Angle ref win size	range ref guard size	Angle ref guard size	treshold	treshold	sidelobThr	Enable2ndPass	dynamicFlag

- Range-Azimuth CFAR Threshold Parameters: rangeThre, angleThre
  These thresholds are defined as ratios of the local noise floor to the cell under test(CUT), above which a detected point is declared. The lower the thresholds, the more false detections will occur on random noise, object sidelobes, and other spurious environmental effects. The higher the thresholds, the fewer points from the objects of interest are detected.
  - rangeThre: This is the relative threshold for the first pass (Range domain). After noise power are calculated based on the refWinSize1 and guardWinSize1, the final detection threshold will be calculated by noisePower * rangeThre. If the power of CUT is greater than the threshold, it will be declared as detected point. The threshold in range domain settings impact detection performing significantly. The lower the threshold, the more low confident detections may show up. User would see more noisy points around the target, more multipath ghost targets forming between the real target and strong reflectors such as metal beams and walls, and more angle and Doppler errors from the low confident point. On the other hand, if the threshold is too high, user will lose detected point that potentially carry valuable information about the target's angle/size and Doppler. Lost of these information would affect the performance of the following modules such as tracker and classifier, which rely heavily on the statistics/distribution of the Range/angle/Doppler information from detected points from a target.
  - angleThre: Similar to rangeThre, this is the relative threshold for the second pass (angle domain). And similarly, setting of this threshold also affect the overall performance of the system. If set too low, user will see angle spread from the target (the rainbow effect around the target), which may mislead tracker. If set too high, we again lose valuable information.
  - SidelobeThr: This is used in combining with the 2nd pass search. If 2nd pass is enabled, the algorithm will try to confirm the range-domain detected points in the angle domain using CFAR-CASO. If it is not confirmed by CFAR, we will also check whether the range-domain detected point is a local maxima in the angle domain. If yes and its power exceeds the threshold*maxP in the same range bin, we will confirm it as detected points. If 2nd pass is disabled, the CFAR search will be skipped, only confirming the point by checking the local maxima and exceeding the threshold*maxP in the same range bin.
- Enable2ndPass: indicating 2nd pass CFAR confirmation is enabled or not
- dynamicFlag: flag indicating the CFAR is for dynamic or static scene detection

Static Scene CFAR configuration parameters (1)

staticRACfarCfg													
<cfarMethod>	<cfarDiscardLeft1>	<cfarDiscardRight1>	<cfarDiscardLeft2>	<cfarDiscardRight2>	<refWinSize1>	<refWinSize2>	<guardWinSize1>	<guardWinSize2>	<rangeThre>	<angleThre>	<sidelobThr>	<Enable2ndPass>	<dynamicFlag>
6	4	4	2	4	8	16	4	6	8	13	0.3	0	0
6: 2-pass range-azimuth CFAR	samples discarded on the left (range)	samples discarded on the right(range)	amples discarded on the left(angle	samples discarded on the right(angle)	range ref win size	Angle ref win size	range ref guard size	Angle ref guard size	treshold	treshold	sidelobThr	Enable2ndPass	dynamicFlag

- Static scene CFAR configuration has the same fields (and descriptions) as the dynamic scene CFAR configuration, with the following field having different values:
  - guardWinSize2
  - rangeThre
  - angleThre
  - SidelobeThr
  - Enable2ndPass
  - dynamicFlag

Dynamic Scene Range-Angle Heatmap Estimation Configuration

dynamicRangeAngleCfg			
<azimuthSearchStep>	<rangeAngleDiagonlLoading>	<rangeAngleEstMethod>	<DopplerEstMethod>
0.75	0.001	1	0
Azimuth search granularity	Range-angle domain diagonl loading	1: range azimuth, the only method support now	1: single peak search in Doppler domain, only method supported.

- Range-Angle heatmap estimation uses Copan beamforming in range-azimuth domain for dynamic scene. The configurations are detailed below:
  - azimuthSearchStep: this parameter defines the search resolution (inter-bin resolution). Due to memory limitation, it should not be set to value smaller than 0.75. In addition, because this parameter defines the Azimuth inter-bin resolution, user should also adjust guardWinSize2 in cfarCfg when changing this parameter. This parameter is also antenna pattern dependent.
  - gamma: this is the diagonal loading parameter for constructing spatial covariance matrix for Capon beamforming for range-angle heatmap estimation. 0.001 is the best empirical setting we obtained from testing. We would strongly suggest not changing the setting. This parameter, by affecting the spatial covariance matrix calculation, also affects the azimuth angle spectrum calculation for each range bin.
  - rangeAngleEstMethod: The C implementation currently only support the detection method of range-azimuth heatmap based CFAR detection, and on detected the detected [rangeBin, azimuthBin] pair, 2D azimuth-elevation heatmap for elevation estimation. We have found this method is better suitable for 68xx ISK wallmount application.
  - DopplerEstMethod: This C implementation only support single peak search in Doppler domain for Doppler estimation. We noticed CFAR search in Doppler domain in the indoor environment is very noise and not desirable in terms of performance.

Static Scene Range-Angle Heatmap Estimation Configuration

staticRangeAngleCfg		
<staticProcEnabled>	<staticAzimStepDeciFactor>	<staticElevStepDeciFactor>
1	8	2
Static scene processing enabled	Azimuth domain decimation factor	Elevation domain decimation factor

- Range-Angle heatmap estimation uses Bartlett beamforming in range-azimuth-elevation domain for static scene. The configurations are detailed below:
  - staticProcEnabled: enable the static scene processing by setting this flag to 1. If there is no need of static scene processing, one can disable it to save MIPS.
  - staticAzimStepDeciFactor and staticElevStepDeciFactor: the static scene processing uses lower angular search resolution than the dynamic scene processing. Since several buffers and parameters are reused, we only specify the decimation factor here.

2D Angle Estimation Configuration for Dynamic Scene

dynamic2DAngleCfg							
<elevSearchStep>	<angleDiagonlLoading>	<maxNpeak2Search>	<peakExpSamples>	<elevOnly>	<sideLobThr>	<peakExpRelThr>	<peakExpSNRThr>
3	0.03	1	0	1	0.5	0.85	8
Elevation search granularity	Angle domain diagonl loading	1: the only value support now	0: single peak search in Doppler domain, only value supported.	1: evelation estimation only, only method supported.	sidelobe threshold	peak expansion threshold	peak SNR threshold for peak expansion

- elevSearchStep : this parameter defines the search resolution (inter-bin resolution). Due to memory limitation, it should not be set to value smaller than 1. This parameter is also antenna pattern dependent.
- angleDiagonlLoading: this is the diagonal loading parameter for constructing spatial covariance matrix for Capon beamforming for azimuth-elevation heatmap estimation. 0.03 is the best empirical setting we obtained from testing. We would strongly suggest not changing the setting. This parameter, by affecting the spatial covariance matrix calculation, also affects the angle spectrum calculation for each range bin, as well as the Doppler spectrum estimation. Changing it could cause bigger angle and Doppler estimation errors.
- maxNpeak2Search: Current version of C implementation only support single peak search in the elevation domain. This hook enables future support of multi-peak search to enrich the point cloud. The following parameter is related to possible future support of multi-peak search:
  - sideLobThr: The relative threshold (comparing to the global maxima) to declare a local peak as detected point.
- peakExpSamples: Current version of C implementation does not support peak expansion around the peak. This hook enables future support of peak-expansion search to enrich the point cloud. The following parameter are all related to possible future support of peak-expansion:
  - peakExpRelThr: The relative threshold to include the neighbor of the peak as detected point.
  - peakExpSNRThr: The SNR threshold to allow peak expansion for the detected peak.
- elevOnly: Although we calculate the 2D azimuth-elevation heatmap, we only perform elevation detection in the current C implementation.

FoV Configuration

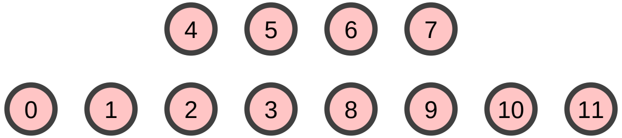
fovCfg	
<AzimuthFoV>	<ElevationFoV>
70	20
Azimuth FoV	Elevation FoV

- FoV configuration used in the system. Please note that these parameters also have big impact on memory usage. The current build of the implementation only support maximum FoV defined in the table above. User needs to adjust the local heap memory to be able to support wider FoV.

Antenna Geometry and Phase Rotation from the Board

antGeometry0 and 1											
<virtAntIdx 0>	<virtAntIdx 1>	<virtAntIdx 2>	<virtAntIdx 3>	<virtAntIdx 4>	<virtAntIdx 5>	<virtAntIdx 6>	<virtAntIdx 7>	<virtAntIdx 8>	<virtAntIdx 9>	<virtAntIdx 10>	<virtAntIdx 11>
0	-1	-2	-3	-2	-3	-4	-5	-4	-5	-6	-7
-1	-1	-1	-1	0	0	0	0	-1	-1	-1	-1

- Antenna geometry parameter 0 and 1 defines the physical location of the virtual antennas in azimuth and elevation domain. In the above example of 68xx ISK, antGeometry1 indicates there are 2 elevation rows (specified by 0 and -1). And antGeometry0 indicates the physical location of the virtual antenna on each elevation row. The physical pattern define by the above table is shown as below:



antPhaseRot											
<virtAntIdx 0>	<virtAntIdx 1>	<virtAntIdx 2>	<virtAntIdx 3>	<virtAntIdx 4>	<virtAntIdx 5>	<virtAntIdx 6>	<virtAntIdx 7>	<virtAntIdx 8>	<virtAntIdx 9>	<virtAntIdx 10>	<virtAntIdx 11>
1	1	1	1	1	1	1	1	1	1	1	1

- antPhaseRot defines the phase rotation introduced in the board design. Should be set to 1 if no rotation exists.

Phase Bias Compensation Parameter

compRangeBiasAndRxChanPhase												
<rangeBias>	<virtAntIdx 0>	<virtAntIdx 1>	<virtAntIdx 2>	<virtAntIdx 3>	<virtAntIdx 4>	<virtAntIdx 5>	<virtAntIdx 6>	<virtAntIdx 7>	<virtAntIdx 8>	<virtAntIdx 9>	<virtAntIdx 10>	<virtAntIdx 11>
0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0	1, 0

- This set of range and phase compensation parameters should be the output from TI's MMW SDK OOB demo. Please refer to the procedure defined in MMW SDK for how to obtain them. If user decide to not compensating the bias, the default value in the above table should be set in the configuration file.