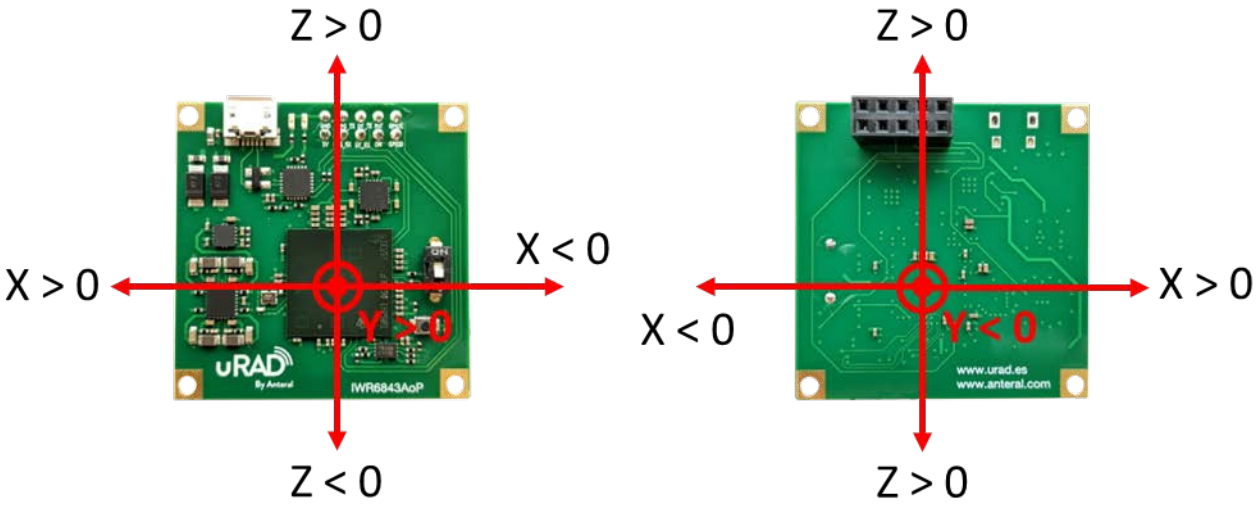
3D people counting

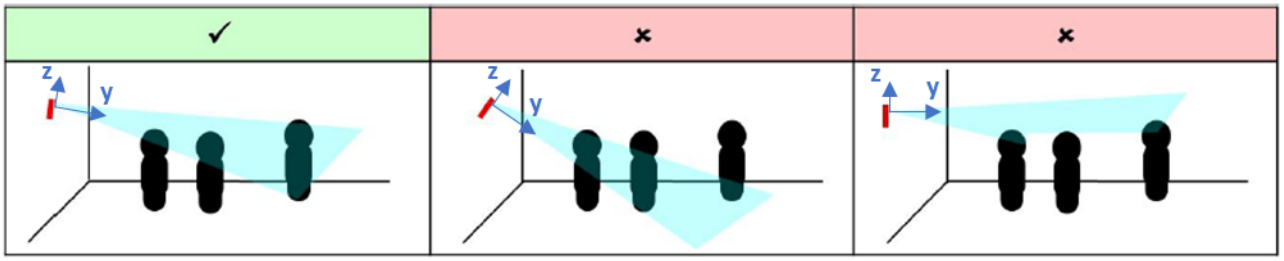
Sensor position

For 3D people detection, uRAD radar sensor should be positioned high enough to be above the top of tracked objects and with a slight down tilt. The aim is to position the EVM so that the antenna beam can encompass the area of interest. If the down tilt is too severe, noise from ground clutter would increase and the effective sensing area would decrease. If there is no down tilt, counting performance would be worse for cases in which one person is in line with and shielded by another person.

Radar orientation should be:

- X axis parallel to ground (±70 degrees horizontal field of view)
- Z axis with a certain tilt (±70 degrees vertical field of view)
- Y axis pointing to the heads of the people to detect.





Radar can be positioned with the micro-USB connector upward or downward. Just take into account the orientation of the X and Z axis in the configuration parameters.

Configuration parameters

Configuration is loaded from the **.cfg* file. This is an example of a **.cfg* file for the 3D people counting application. Each line represents a command line interface message that configure several parameters.

```
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 394758 0 54.71 1 96 2950.00 2 1 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 1 0
dynamicRACfarCfg -1 4 4 2 2 8 12 4 8 5.00 8.00 0.40 1 1
staticRACfarCfg -1 6 2 2 2 8 8 6 4 8.00 15.00 0.30 0 0
dynamicRangeAngleCfg -1 0.75 0.0010 1 0
dynamic2DAngleCfg -1 1.5 0.0300 1 0 1 0.30 0.85 8.00
staticRangeAngleCfg -1 0 8 8
antGeometry0 -1 -1 0 0 -3 -3 -2 -2 -1 -1 0 0
antGeometry1 -1 0 -1 0 -3 -2 -3 -2 -3 -2 -3 -2
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
boundaryBox -4 4 0 8 0 3
staticBoundaryBox -3 3 0.5 7.5 0 3
presenceBoundaryBox -3 3 0.5 7.5 0 3
sensorPosition 2 0 15
gatingParam 3 2 2 3 4
stateParam 3 3 12 500 5 6000
allocationParam 20 100 0.1 20 0.5 20
trackingCfg 1 2 800 30 46 96 55
maxAcceleration 0.1 0.1 0.1
sensorStart
```

Lines starting by % are comments and it has to be a final line empty after `sensorStart`.

Black lines correspond with the configuration of the signal chirp. Chirp defines the antenna configuration, frequency band, frame rate, range and velocity resolution, maximum unambiguous range and velocity, etc. The values of this line can be obtain using the *mmWave Demo Visualizer* by Texas Instruments. Read the general user manual of uRAD Industrial to learn about this tool. Additional information of each of these parameters can be found in the document *standard_mmwave_ssdk_commands_TI.pdf* by Texas instruments included with this packet. **Default values are recommended**.

Red lines define configuration parameters of the CFAR detection algorithm of the point cloud. **Default values are recommended**. Additional information can be found in the document *3D_people_counting_detection_layer_tuning_guide.pdf* by Texas instruments.

Green lines define the antenna geometry and board related parameters. Do not change default values of `antGeometry0`, `antGeometry1` and `antPhaseRot`. Additional information can be found in the document *PeoplecountingDemo_ConfigDetails_TI.pdf*

- fovCfg [-1] [AzimuthFoV] [ElevationFoV]

  Defines the Field of View of the radar. Maximum FoV supported by uRAD IWR6843AoP radar is 70 in each field which defines +/- 70 degrees. Reduce the elevation field of view around 20 degrees. This parameter has big impact on memory usage because it can highly reduce the number of detected points.

- compRangeBiasAndRxChanPhase

  This parameter is for compensating the bias (calibration). This set of range and phase compensation parameters can be left by default. For higher accuracy, the set of parameters can be obtained with a simple experiment and using the *mmWave Demo Visualizer*. In the mmWave SDK document is explained how to obtain them.

Blue lines define the group tracker configuration parameters. Sensor detects a dense point cloud in each frame.

- boundaryBox [LeftWall] [RightWall] [BackWall] [FrontWall] [Ceil] [Ground]

  This sets boundaries where tracks can exist. Only points inside the box will be used by the tracker. Each value denotes an edge of the 3D cube. According to radar axis, these walls would be [-X] [X] [-Y] [Y] [-Z] [Z] being the axis origin the center of the radar. Detected points of the point cloud can exit outside this boundary but cannot outside the field of view previously defined. Take into account radar position.

- staticBoundaryBox [LeftWall] [RightWall] [BackWall] [FrontWall] [Ceil] [Ground]

  This defines the zone in which tracks are expected to become static for a long time. Each value denotes an edge of the 3D cube in meters. When a person is tracked and enters this box, if the person stops, the track is not lost. According to radar axis, these walls would be [-X] [X] [-Y] [Y] [-Z] [Z] being the axis origin the center of the radar. Take into account radar position.

- presenceBoundaryBox [LeftWall] [RightWall] [BackWall] [FrontWall] [Ceil] [Ground]

  The presence module in the tracker layer determines if any target exists within the occupancy area specified by the presence boundary box. Each value denotes an edge of the 3D cube in meters. When a person is tracked and enters this box, if the person stops, the track is not lost. According to radar axis, these walls would be [-X] [X] [-Y] [Y] [-Z] [Z] being the axis origin the center of the radar. Take into account radar position.

- sensorPosition [Height] [AzimuthTilt] [ElevationTilt]
  - Height: height of sensor from the floor.
  - AzimuthTilt: horizontal rotation of sensor. This must be set to 0.
  - ElevationTilt: vertical rotation of sensor with respect to ground. Sensor should be tilted between 10-20 degrees. If micro-USB connector is upward (axis Z > 0 upward), the value must be positive, if connector is downward (axis Z < 0 upward), the values must be negative.

- gatingParam [Gain] [LengthLimit] [WidthLimit] [HeightLimit] [VelocityLimit]

  The gating parameters determine the maximum volume and velocity of a tracked object. These parameters are used to associate detected points with tracked people in the scene. Points detected near a tracked person's centroid, but beyond the limits set by these parameters will not be included in the set of points that make up the tracked object. There are 4 parameters:

Parameter	Default	Dimensions	Description
Gain	3	m	Gating gain
Length Limit	2	m	Gating limit in length
Width Limit	2	m	Gating limit in width
Height Limit	3	m	Gating limit in height
Velocity Limit	4	m/s	Gating limit in velocity.

  Gain is defined as the amount the track dimensions can grow to create the gating function. Each dimension, X, Y, Z, and Velocity can be multiplied by the gain to generate a gating function for the track, which is the space where new points can be associated with the target. Length limit, width limit, height limit and velocity limit also serve to limit the size and velocity of the ellipsoid.

- stateParam [det2activeThre] [det2freeThre] [active2freeThre] [static2freeThre] [exit2freeThre] [sleep2freeThre]

  The state transition parameters determine the state of a tracking instance. Any tracking instance can be in one of three states: FREE, DETECT, or ACTIVE. Instances in ACTIVE state produce tracks. Once per frame, each instance will get a HIT (have one or more points associated with the instance), or a MISS (have zero points associated with the instance). The state transition parameters are described in the table below:

Parameter	Default	Dim	Description
det2activeThre	3	-	In DETECT state, how many consecutive HIT events needed to transition to ACTIVE state.
det2freeThre	3	-	In DETECT state, how many consecutive MISS events needed to transition to FREE state.
active2freeThre	12	-	In ACTIVE state and NORMAL condition, how many consecutive MISS events needed to transition to FREE state.
static2freeThre	500	-	In ACTIVE state and STATIC condition, how many consecutive MISS events needed to transition to FREE state.
exit2freeThre	5	-	In ACTIVE state and EXIT condition, how many consecutive MISS events needed to transition to FREE state.
sleep2freeThre	6000	-	In ACTIVE state and sleep condition, how many misses are needed to transition to FREE state.

A target in STATIC condition is not moving, and a target in NORMAL condition is moving. A target in EXIT is near the edge of the detection zone.

- allocationParam [SNRThre] [SNRObscuredThre] [VelocityThre] [PointsThre] [maxDistanceThre] [maxVelocityThre]

  The allocation parameters are used to detect new people in the scene. When detected points are not associated with existing tracked people, allocation parameters are used to cluster these remaining points. Allocation parameters are then used to determine if a cluster qualifies as a person. MaxDistanceThre and maxVelThre determine when the candidate point can be added to the cluster (allocation set). SNR Threshold, Velocity Threshold, and Points Threshold determine when an allocation set becomes a track.

Parameter	Default	Dim	Description
SNRThre	20	-	Minimum total SNR for the allocation set, linear sum of power ratios.
SNRObscuredThre	100	-	Minimum total SNR for the allocation set. If allocation set is behind an existing track, linear sum of power ratios.
VelocityThre	0.1	m/s	Minimum radial velocity of the allocation set centroids
PointsThre	20	-	Minimum number of points in the allocation set

maxDistanceThre	0.5	m^2	Maximum squared distance between candidate and centroid to be part of the allocation set
maxVelocityThre	20	m/s	Maximum velocity difference between candidate and centroid to be part of the allocation set

- trackingCfg [TrackerEnabled] [DefaultConfiguration] [MaxPoints] [MaxTracks] [MaxRadialVelocity] [RadialVelocityResolution] [FrameTime]

Parameter	Default	Dim	Description
TrackerEnabled	1	-	Enables or disables the tracker. 1 is enabled.
DefaultConfiguration	2	-	Do not change default configuration.
MaxPoints	800	-	Maximum number of detection points per frame for tracker to consider.
MaxTracks	30	-	Maximum number of targets to track at any given time.
MaxRadialVelocity	46	m/s*10	Maximum absolute radial velocity reported by sensor (does not affect signal chain).
RadialVelocityResolution	96	m/s*1000	Minimal non-zero radial velocity reported by the sensor (does not affect signal chain).
FrameTime	55	ms	**Ensure the frame time matches the frame period of the frameCfg command (5^{th} parameter).**

- maxAcceleration [Xdirection] [Ydirection] [Zdirection]

  These parameters specify the maximum amount that the acceleration can change in the lateral, longitudinal and vertical directions between frame periods. As an example, we are interested in modeling the position, velocity and acceleration of a target and we choose to use a 3D constant acceleration model. The assumption is that for each discrete time step the position and velocity of the object can potentially change however the acceleration would remain constant. However, in reality there

is no guarantee that the targets acceleration will remain constant as there could be several unknown external un-modeled factors that can alter this assumption.

These max acceleration parameters try to model this deviation from our assumed motion model. Setting a large value will imply that we don't trust our motion model and hence the Prediction step is less reliable. Setting a low value implies that our motion model is very accurate and we do not expect much uncertainty. There are 4 parameters:

Parameter	Default	Dimensions	Description
Xdirection	0.1	m/s^2	Maximum amount that the target acceleration is expected to change in the X-direction between time-periods
Ydirection	0.1	m/s^2	Maximum amount that the target acceleration is expected to change in the Y-direction between time-periods
Zdirection	0.1	m/s^2	Maximum amount that the target acceleration is expected to change in the Z-direction between time-periods

Fine Motion Detection

In the 3D people counting demo, a processing mode is added to detect and track static people with fine motions (standing, sitting still, sleeping, etc.).

Two new parameters have to be added to the chirp config file to enable this mode.

- fineMotionCfg [subFrameIdx] [fineMotionProcEnabled] [fineMotionObservationTime] [fineMotionProcCycle] [fineMotionDoplerThrIdx]

Parameter	Default	Description
subFrameIdx	-1	Always set to -1
fineMotionProcEnabled	1	0: Fine motion processing disabled. 1: Fine motion processing enabled.
fineMotionObservationTime	1.0	Total observation time (in seconds) is utilized to compute the number of frames combined and the number of chirps selected from each frame.
fineMotionProcCycle	10	The processing cycle of the fine-motion mode (in frames). The algorithm runs the multi-frame block once at every N frame.

fineMotionDoplerThrIdx	2	The Doppler threshold (in terms of the number of one-sided Doppler bins) is utilized when filtering the dynamic points in the fine-motion mode.

- bpmCfg [subFrameIdx] [fineMotionProcEnabled] [fineMotionObservationTime] [fineMotionProcCycle] [fineMotionDoplerThrIdx]

Parameter	Default	Description
subFrameIdx	-1	Always set to -1
enabled	1	Enable/disable the BPM mode. Set to 0 to disable (i.e., all TX antennas are in TDM mode) and 1 to enable (i.e., two TX antennas are in BPM mode and 1 TX antenna is in TDM mode).
chirp0Idx	0	If BPM is enabled in previous argument, this is the chirp index for the first BPM chirp. It will have phase 0 on both TX antennas (TXA+ , TXB+). Note that the chirpCfg command for this chirp index must have both TX antennas enabled. If BPM is disabled, this argument has no impact on the demo.
chirp1Idx	2	If BPM is enabled, this is the chirp index for the second BPM chirp. It will have phase 0 on TXA and phase 180 on TXB (TXA+ , TXB-). Note that the chirpCfg command for this chirp index must have both TX antennas enabled. If BPM is disabled, this argument has no impact on the demo.

Additional information can be found in document *3D_people_counting_tracker_layer_tuning_guide.pdf.*

Configuration tips

- **Too Few People counted**

There are multiple parameters that affect when a person can be positively identified and tracked. First, make sure that the chirp configuration that the device is configured to acquire data with extends to the range at which we want to detect people/targets and that the <boundaryBox> is set accordingly.

If there are reflection points coming from the person but the tracker is unable to allocate a track, then consider changing the track allocation parameters shown in Table 12. However, it should be kept in mind that lowering these parameters increases the chances of false detection.

Parameter Type	Field	Tuning Guide
Allocation Params	SNRThre	Further the distance of the target from the radar, the lower the SNR of the detected points. Lowering the SNR threshold will lower the cumulative SNR that is required for an allocation set to be considered as a track.
Allocation Params	PointsThre	Fewer points are needed for an allocation set to be considered as a track if the pointsThre is lowered. Lowering this threshold might be beneficial for longer ranges as fewer.
Allocation Params	VelocityThre	Lowering the velocity threshold will allow a person making very tiny movements to still be tracked.

- **Too Many People Counted (Ghosts)**

There could be several reasons that the tracker is detecting too many people in the scene. One likely cause is multipath reflections (radar Energy reflected from a person being reflected again from a wall, ceiling, floor or some object) which will cause the tracker to produce a false detection. These false detections are called ghosts. Ghosts can be caused by multiple phenomena, but there are a few parameters that can be tuned to minimize or remove ghosting.

Parameter Type	Field	Tuning Guide
Scenery Params	boundaryBox	Start by properly setting the scenery parameters. Many ghosts caused by multipath reflections from the wall will appear outside of the detection area. These can be immediately ruled out if the scenery parameters are properly set.

Allocation Params	SNRThre	Ghosts will usually have detection points with lower SNR.
Allocation Params	PointsThre	Lowering the velocity threshold will allow a person making very tiny movements to still be tracked.
State Params	det2actThre	Increase the amount of time the ghost has to exist before its state is changed to ACTIVE state. Useful if the Ghost only appears momentarily.
State Params	det2FreeThre static2FreeThre active2FreeThre axit2FreeThre	Lowering these thresholds, the tracks will be freed faster.

Items # 2,3,4 in table can be used to reduce false detections. Ghosts will usually have fewer points with lower SNRs. Increasing the Points Threshold and SNR Threshold will stop the tracker from allocating these clusters as tracks. Increasing the <det2actThre> threshold will increase the amount of time the ghost has to exist before it is promoted to ACTIVE state. In the case where a ghost only appears momentarily, this can stop the tracker from tracking it.

If changing these parameters fails to stop ghosts from appearing, consider changing the parameters in item #5 i.e. <det2FreeThre>, <static2FreeThre>, <acitive2FreeThre>, <exit2FreeThre>. By lowering these thresholds, tracks will be freed faster, so the ghost will be tracked for a shorter period of time. However, lowering these may reduce the ability of the tracker to properly maintain a track on a real target, especially in a cluttered environment.

Resolving Multiple People When Close Together

In some situations, the tracker may allocate one track to two or more people. This is likely to happen when people are near each other, and walking at the same pace in the same direction (a fairly common occurrence). Other times, the tracker may give one person multiple tracks.

To prevent these situations, consider changing the following parameters: All of these parameters will affect how points in the point cloud get distributed into different tracks.

Parameter Type	Field	Tuning Guide
Allocation Params	maxDistanceThre	The Allocation parameters will affect how tracks are created. Detected points are clustered, and each cluster can become a tracked person. Lowering these thresholds will force the point clusters that become tracked people to be smaller, increasing the differentiation between people, while raising them will allow the point clusters to be larger, decreasing differentiation between people.
Allocation Params	maxVelThre	
Allocation Params	Gain	These parameters determine how close a measurement needs to be to be counted as part of an existing track. Lowering these values will help the tracker to separate the individual point clouds if multiple people will be walking near each other. However setting these values too low will result in allocating multiple tracks for a single person
Gating Params	Limit-Length	
Gating Params	Limit-Width	
Gating Params	Limit-Velocity	

Tracks from stationary people (that were once moving) and disappearing

At times, it may happen that a tracker was successfully tracking a moving person but when the person stopped moving (e.g. sat down, standing very still) the track got de-allocated. This could be due to multiple reasons.

First, make sure that the static Boundary box is configured correctly i.e. the stationary target is within the static Boundary box.

The maximum lifespan for static targets inside the static box is determined by the state transition parameters <static2freeThre> and <Sleep2FreeThre>. Try increasing the values of these parameters which will help in keeping the track alive for a longer time duration.

If the particular use-case scenarios are likely to have people sitting very still or stationary then it is recommended to tune the detection layer parameters for increased motion sensitivity (see document *3D_people_counting_tracker_layer_tuning_guide.pdf*).

Parameter Type	Field	Tuning Guide
Scenery Params	staticBoundaryBox	Make sure that the stationary target is within the static boundary boxes
State Params	static2FreeThre sleep2FreeThre	These thresholds are only applicable to static targets within the Static zone. Larger these values the longer the track will persist inside the static boundary box. If a static target disappears, then try increasing these values

Tracks from moving people are disappearing

Sometimes a track corresponding to a moving person might disappear. One likely cause for this is that the person/target is occluded by other objects in the scene and hence there won't be any reflection points detected from the target. The performance of the tracker is dependent on the continuity of the tracking process and occlusion of targets might cause a track to disassociate from its target. In these cases very little can be done. Most likely, these occlusions would lead to temporary short-time errors, and once the target is visible again, a track will be re-allocated.

Sometimes the track might get de-allocated if the reflected points from the target do not get associated with its track (i.e. they fall outside the gating function of the track). In these cases, increasing the values of the Gating Parameters can be considered. However, as mentioned in Section 4.3, setting the gating parameters too high can result in difficulty in resolving multiple close-by people.

A track from a moving person can also disappear, if the motion dynamics of the target violate the assumptions made in the tracker motion model. For instance, in the 3D constant acceleration model, we expect the acceleration of the target to be constant. If a moving target abruptly changes acceleration (e.g. makes a sharp turn), the motion model might not adequately capture this and can result in the target track being lost. As discussed in Section 3.5, max acceleration parameters can be used to model these deviations from the assumed motion model. Setting these parameters to larger values will imply that our motion model has more uncertainty and the EKF will give comparatively less weightage to the Prediction step.

Parameter Type	Field	Tuning Guide
Max Acceleration Params	maxAcceleration	These max acceleration parameters try to model the deviation from our assumed 3D constant acceleration motion model. If a track from a moving target is disappearing then increasing this value can be considered. Setting a large value will imply that we don't trust our motion model and hence the Prediction step is less reliable.
Gating Params	Gain Limit-depth Limit-width Limit-height	These parameters determine how close a measurement needs to be to be counted as part of an existing track. If a track from a moving target is disappearing then increasing these values can be considered. However, keep in mind that setting the gating parameters too high can result in difficulty in resolving multiple close-by people.

Results

Python scripts save in the *output_files* folders three different .txt files (**set first the corresponding configPort_name, dataPort_name and configFile_name in the scripts**).

- **PointCloud.txt**

Each line of this file contains the information of the point cloud in each frame. It saves consecutively in each column

[range (m)] [azimuth (deg)] [elevation (deg)] [doppler (m/s)] [snr (dB)]

of every detected point of the point cloud. At the end, the timestamp is included, as the number of seconds since January 1, 1970, 00:00:00 (UTC).

Range, azimuth and elevation can be transformed to XYZ coordinates of the radar with the following formulas:

x = range * cos(elevation) * sin(azimuth)
y = range * cos(elevation) * cos(azimuth)
z = range * sin(elevation)

- **Targets.txt**

Each line of this file contains the information of the tracked targets in each frame. It saves consecutively in each column the track identifier, position (m), velocity (m/s) and acceleration (m/s^2) of each track:

[target ID] [posX] [posY] [posZ] [velX] [velY] [velZ] [accX] [accY] [accZ]

At the end, the timestamp is included, as the number of seconds since January 1, 1970, 00:00:00 (UTC).

- **TargetsIndex.txt**

Each line of this file contains the information of how each detected point of the point cloud is associated to the tracks. There is one column for each point with the value of the target ID associated at each point. If the point is not associated to any target, the value is:

253: point not associated, SNR too week.
254: point not associated, located outside boundary of interest
255: point not associated, considered as noise.

At the end, the timestamp is included, as the number of seconds since January 1, 1970, 00:00:00 (UTC).

- **TargetsHeight.txt**

Each line of this file contains the information of the estimated minimum and maximum height of each target together with the tack identifier:

[target ID] [maxZ (m)] [minZ (m)]

At the end, the timestamp is included, as the number of seconds since January 1, 1970, 00:00:00 (UTC).

Take into account that the tracking algorithm tends to underestimate the height. You can find more information in the document *3D_people_counting_demo_implementation_guide.pdf.*

Graphical User Interface

The Graphical User Interface (GUI) allows to visualize in real time the point cloud, the tracks as well as the defined static and dynamic boundary box. The axis in the GUI is aligned with the axis of the radar. Radar is located at the origin of the coordinates (0,0).

