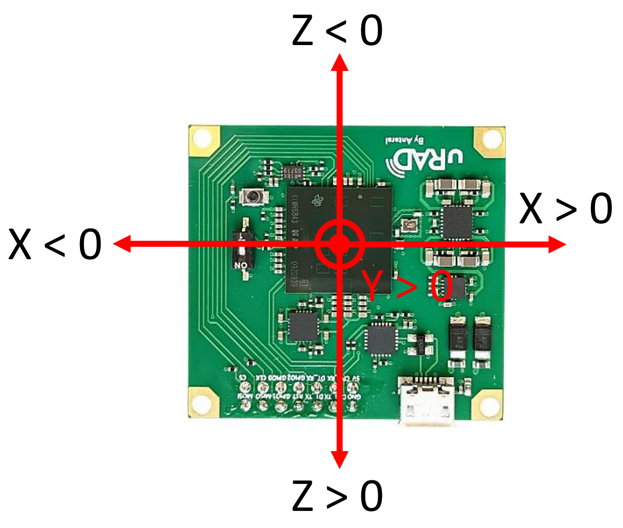
3D people counting

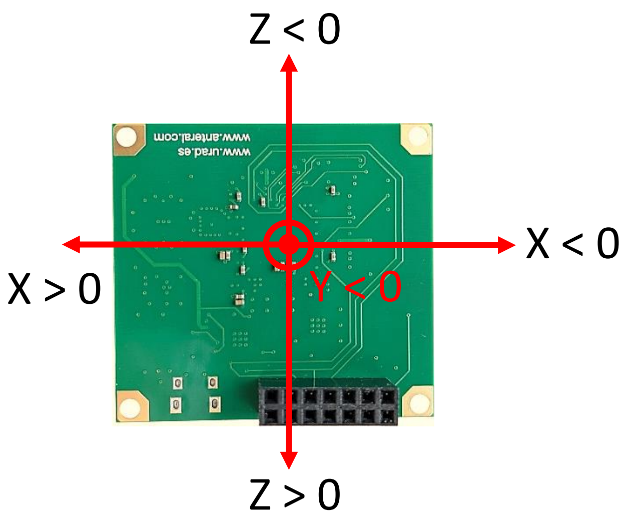
Sensor position

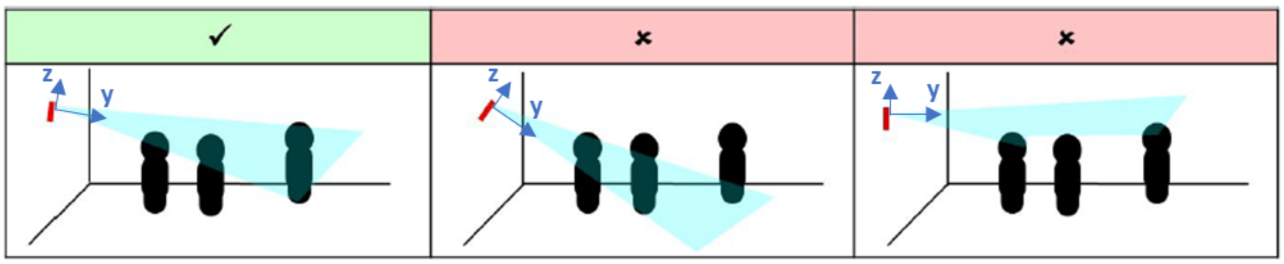
For 3D people detection, uRAD radar sensor should be positioned high enough to be above the top of tracked objects and with a slight down tilt. The aim is to position the EVM so that the antenna beam can encompass the area of interest. If the down tilt is too severe, noise from ground clutter would increase 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 (±80 degrees horizontal field of view)
- Z axis with a certain tilt (±80 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.

```
%WALL MOUNTED
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 0
staticBoundaryBox -2 2 1 4 -2 2
boundaryBox -2.5 2.5 0.5 5 -2.5 2.5
sensorPosition 2 0 15
gatingParam 3 2 2 2 4
stateParam 3 3 6 300 5 1000
allocationParam 40 100 0.1 20 0.5 20
trackingCfg 1 2 800 30 46 96 55
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.

Red lines define configuration parameters of the CFAR detection algorithm of the point cloud. We recommend do not change the default values. Additional information can be found in the document *PeoplecountingDemo_ConfigDetails_TI.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.

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

  This sets boundaries where static points can be used by the tracker and tracks are allowed to become static. 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.

- 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.

- 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	2	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	6	-	In ACTIVE state and NORMAL condition, how many consecutive MISS events needed to transition to FREE state.
static2freeThre	300	-	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	1000	-	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	40	-	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 (5th parameter).**

Configuration tips

- **Too Few Detections**

There are multiple parameters that affect when a person can be positively identified and tracked. First, make sure you are using a chirp that extends to the range at which you want to detect people. Then consider changing these three parameters:

1) Allocation Parameters – SNR Threshold
2) Allocation Parameters – Points Threshold
3) Allocation Parameters – Velocity Threshold

At a distance, the reflected signal will be weaker. This will decrease the SNR of the target points, and potentially reduce the size of the point cloud. By lowering the Points threshold, fewer points are needed to allocate a track to the cluster. Lowering the SNR threshold will lower the cumulative SNR required for an allocation set to be considered a track. Lowering the Velocity threshold will allow a person to make very tiny movements and be tracked. Be careful, as lowering these parameters increases the chance of false detection.

- **Too Many Detections (Ghosts)**

Sometimes, multipath reflections (radar energy reflected from a person being reflected again from a wall or some object) 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 many tools in the people counting software to minimize or completely remove ghosting.

Start by properly setting the scenery parameters. Many ghosts caused by multipath reflections will appear outside of the detection area. These can be immediately ruled out if the scenery parameters are properly set.

In cases where the ghost appears in the valid area, you may try changing other parameters:

1) Allocation Parameters – SNR Threshold
2) Allocation Parameters – Points Threshold
3) State Transition Parameters – Det2Active Threshold
4) State Transition Parameters – Det2Free, Static2Free, Active2Free, exit2Free

The first three parameters in the list 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 Det2Active 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 the 4th bullet. 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 for 2 or more people. This is likely to happen when they 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:

1) Allocation – maxDistanceThre
2) Allocation – maxVelThre
3) Gating – Gain
4) Gating – Length Limit
5) Gating – Width Limit
6) Gating – Velocity Limit

All of these parameters will affect how points in the point cloud get distributed into different tracks.

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.

Gating parameters are used to associate points with tracks that already exist. By lowering these thresholds, points will have to be closer to a track to be counted as part of that track. Increasing these parameters will allow a track to take on points farther away. In situations where multiple people will be walking near each other, keeping these values low will make it easier for the algorithm to separate their individual point clouds.

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**

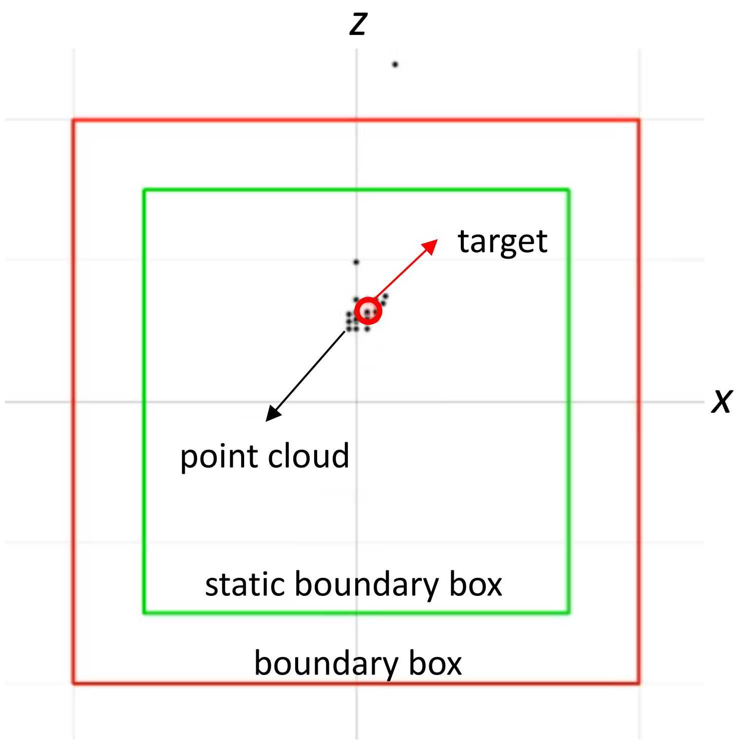
Each line of this file contains the information of how each detected point of the point cloud is associated to the tracks. There is one column for each point with the value of the target ID associated at each point. If the point is not associated to any target, the value is:

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).

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).



z
target
x
point cloud
static boundary box
boundary box