CONTENTS

PREFACE

This Master Thesis was initiated at Tallinn University of Technology, Embedded AI Research Lab and the topic was proposed by leader of the lab, Mairo Leier.

The work is about improving the detection and tracking capabilities of the developed radar-camera sensor fusion system. Main goal is to optimize relevant configuration parameters of the radar sensor in order to maintain a reliable traffic monitoring by accurately detecting and tracking vehicles in smart city environments.

Here, I would like to express my sincere gratitude to my supervisors - researcher Uljana Reinsalu and research scientist Mairo Leier for their guidance, help and support throughout this period. I would also like to thank engineers Henry Juhanson and Karl Janson for their assistance in the data collection and processing phases.

Keywords: *mmwave radar, object detection, vehicle tracking, traffic monitoring*

List of figures

List of tables

List of abbreviations and symbols

AD – Autonomous Driving

ADAS – Advanced Driver Assistance Systems

ADC – Analog-to-Digital Converter

BS – Background Subtraction

CI – Critical Infrastructure

CIS – Critical Infrastructure Security

CLI – Command Line Interface

CW – Continuous Wave

DBSCAN – Density Based Spatial Clustering of Applications with Noise

DSP – Digital Signal Processors

EHF – Extremely High Frequency

EKF – Extended Kalman Filter

EVM – Evaluation Module

FD – Frame Differencing

FFT – Fast Fourier Transform

FMCW – Frequency Modulated Continuous Wave

FOV – Field of View

NAR – Nuisance Alarm Rate

RF – Radio Frequency

ROI – Region of Interest

SNR – Signal Noise Ratio

SVM – Support Vector Machine

UAV – Unmanned Aerial Vehicle

1. INTRODUCTION

Overview

Critical infrastructures (CI) are organizational and physical structures and facilities that are essential for ensuring the social and economic security of a given nation or community. Failure or degradation of these assets can lead to sustained supply shortages, significant disturbance of public safety, or other dramatic consequences [1].

Ensuring the security of CI is one of the primary issues in present times. Nowadays, industrial and public sectors whose operability is essential in terms of the well-being of a large number of people, such as telecommunication and energy assets, banking and transportation facilities and etc. are exposed to various natural (earthquakes, landslides, flooding, etc.) and intentional (thefts, vandalism, terrorism, etc.) threats. Although, experts are addressing the issue of Critical Infrastructure Security (CIS) by developing different methods, approaches and organizing relevant events and publications, most of these attempts fail to address the CIS issue as an interdisciplinary and universal problem. For handling this issue more efficiently, it needs to be examined in an integrated manner both at the organizational and technological levels and focusing on digital (cyber) as well as physical security [2].

In the field of applying physical security systems for different assets, many researches have been done; various sensors, devices, technologies and complex security systems are developed and currently being used. For ensuring efficient physical security, it is key to have accurate data on objects moving around and/or inside the given facility. This data can be used by relevant units of the security departments per purpose and can play an important role in preventing potential threats or mitigating their effects.

Currently, advances and different approaches used in radio detection and ranging (radar) hardware technology have made it possible to detect objects accurately and get reliable data on object recognition by applying this technology. Because of their ability to function in harsh working conditions, cost-effectiveness and accuracy in object detection, radar technology-based detection systems are more preferred in comparison to others, such as systems based on lidar and imaging. Moreover, radars have a longer detection range and can provide multiple returns per azimuth [3]. While efficiency of some widely-used perimeter security methods, e.g. CCTV cameras and fibre optic detection systems can be limited due to external natural factors and affected by mistaken alarms, on the other hand, radars show better performance regardless of surrounded conditions and are able to keep the nuisance alarm rate (NAR) at a low

level. They can be deployed both on and off-site. Offsite placed radars can still communicate with the rest of the entire onsite system. This makes earlier alerting possible and gives security servicemen plenty of time to respond to different situations before they escalate [4].

Even though radar-based systems are the most preferred tools for object detection, in some situations their capabilities can be insufficient. For example, generally, radars provide reliable measurements of moving objects regardless of the weather conditions but stationary objects in the radar's field of view further complicate the task. Also, the tracking of humans can be especially hard since people's behaviours are often completely unpredictable. Some 24 GHz radars used for traffic monitoring have drawbacks such as lower angular resolution than a camera, lower range/velocity performance vs. higher frequency ranges and etc. [5]. For compensating above mentioned drawbacks and increasing the productivity of the object detection system, a combined method of using an extra sensor(s) with the radar can be tested.

Motivation

As it is stated already, ensuring the physical security of CI is very crucial in present times. Providing an accurate object detection system for these assets is very beneficial in terms of their persistent operability. Applying an approach of radar based fusion technology for this purpose will take the overall safety to higher levels. Reliable data on objects detected in a given range by the system will allow the related units to avoid undesirable situations and help CI planners for better productivity.

Considering the above-mentioned perspectives, relevant research was conducted and the thesis was written with great enthusiasm.

Research objectives

Main research objectives can be summarized as follows:

- Analysing existing solutions and identifying modern trends in the field of radar based object detection technologies;
- Developing the methodology of optimizing the radar tracker for better detection and tracking performance;
- Performing tests, analysing results;
- Identifying shortcomings and providing suggestions for further improvements.

Thesis structure

Contents of the thesis's chapters are briefly described below.

The introduction part provides information about the aim of this work, objectives and motivation of the research and a brief description of the thesis structure.

The literature review part contains theoretical background for object detection, classification and tracking concepts and provides detailed information about proposed solutions, developed projects and current advances of the sector. The chapter also includes information on radar and camera sensor fusion techniques and their applications.

Chapter 3 provides the theoretical basis of radar technology, a brief analysis of their working principle and specifications of the radar sensor used for this work. Hardware used for the data collection is also described in the chapter.

Chapter 4 describes the procedure of configuring the radar, radar processing levels, radar tracker, its parameters and the process of tuning them.

Chapter 5 provides information about preparations for measurements, selection of test environments and settings, implementation and application of MATLAB graphical user interface (GUI). Chapter also includes analysis of acquired results and the process of verifying the accuracy of the developed methodology. Identification of parameters that affect the accuracy and overall productivity and limitations of the system is discussed as well.

In chapter 6 overall work is summarized and suggestions for further improvements to the system are provided.

2. LITERATURE REVIEW AND BACKGROUND

In this chapter, the overview of the literature and existing approaches are discussed. Analysis of the literature that explains previously developed solutions which are closely related to the objective of this thesis helps to understand the basis of developing radar-based object detection and tracking system.

2.1 Object detection and visual classification

Object detection is a computer vision technique that handles identifying and locating instances of semantic objects of a certain class in videos or images. This method of localization and identification can be applied to determine objects in a given scene, count them, track their precise locations, and label these identified objects accurately [6].

This technology has been widely used in various real-world applications, such as autonomous driving, robot vision, intelligent video surveillance and in many other practical industrial productions and scientific researches. With the growing number of security related issues, demand for fast and accurate object detection is also increased in the last years. This rise is also reflected in the number of publications associated with "object detection" from 1998 to 2018 as shown in Figure 1 below.

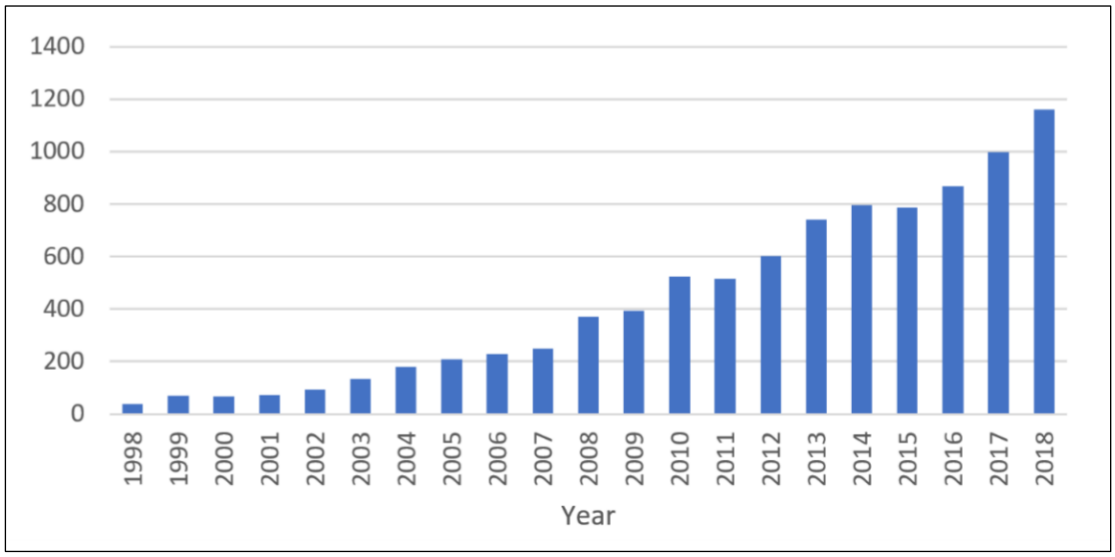


Figure 1. Growing number of publications in object detection from 1998 to 2018 [7]

Although many solutions have been proposed by researchers in the past decades, there are still some challenges in object detection, such as motion blur, video defocus and etc. [7], [8].

Object detection is in close relationship with object classification, semantic segmentation and instance segmentation techniques. These relations can be expressed as follows [9]:

- Object classification is a process of identifying the category of objects in a given visual scene.
- Object detection identifies the category of objects and then locates them with bounding boxes.
- Semantic segmentation is a process of predicting the category of each pixel, and it does not include distinguishing the object instances.
- Instance segmentation predicts both the category of each pixel and object instances in a visual scene.

In general, application of object detection can be examined in two - dedicated and generic object detection classifiers. The details are shown in Figure 2 [9].

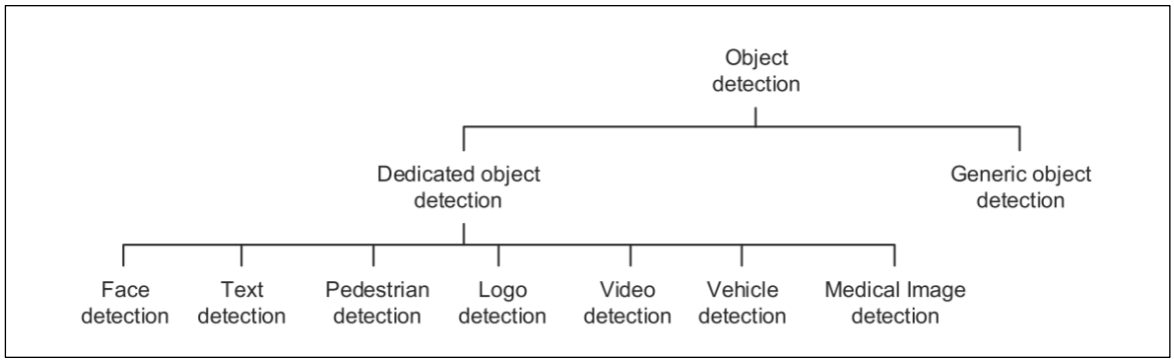


Figure 2. Object detection classification [9]

According to [10], object detection can be grouped into motion-based and appearance-based object detection methods. Motion-based detection methods include the identification of the velocity, acceleration, movement direction and trajectory of the object. On the other hand, appearance-based approaches mainly focus on the features like colour, edge, shape, size and any other static features. Main motion-based detection methods are Background Subtraction (BS), Frame Differencing (FD) and Optical Flow method. Appearance-based methods can be classified as Template Matching and Feature Extraction. Object Classification is the next step after object detection. In this process, detected object is differentiated from other objects in the scene based on four physical features - shape and size, colour, texture (surface) and motion.

In the object detection process, the main goal is to detect a foreground object in a given frame. In this context, a foreground object is a desired object and it is distinct from the stationary background. This distinctiveness can be due to the appearance or local motion

of the object. It can change from frame to frame. Background objects are objects in a given frame that are part of the stationary background.

BS is one of the most popular and repeatedly used techniques for detecting objects. This method is performed on the current frame and then foreground object is detected. Background modelling is the first step in the BS process. The next step involves the Background Update processor that compares the current frame with the previous frame to detect the object. These procedures are described in Figure 3 below. Exact construction and proper implementation of the background model are essential in terms of getting an accurate object detection outcome. Further steps of video surveillance applications such as object recognition and tracking are greatly dependent on the quality of detection [11], [12].

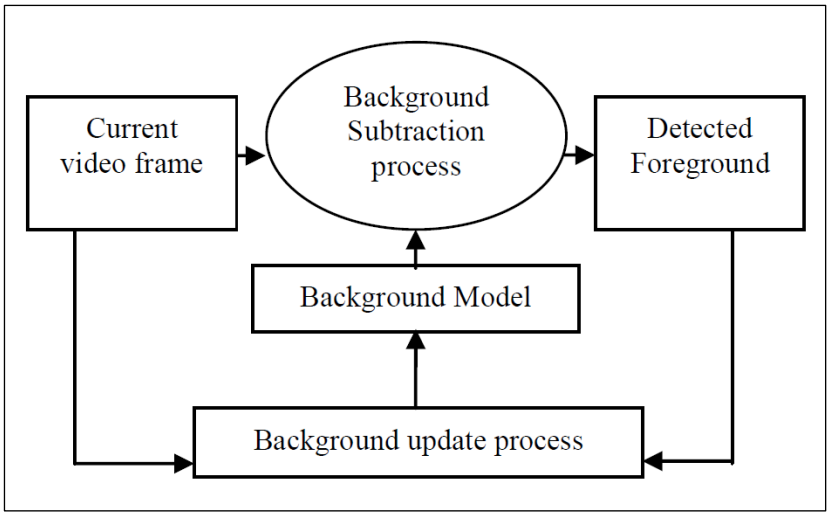


Figure 3. Block diagram of Background Subtraction process [11]

In [13], a BS algorithm known as Mixture of Gaussians was used to get faster detection and efficient prediction of objects if it is a human or not in a surveillance video. System architecture starts with the Video Segmentation phase for an input video. Following steps are BS, Post Processing, connected components labelling, Feature Extraction and Classification. The output of the process is identified class for a particular object. A set of simple and efficient features are extracted and provided to Support Vector Machine (SVM). SVM is a supervised learning model widely used for analysing data for classification and regression analysis. The performance of the system is tested with various kernels of SVM and also for K Nearest Neighbour Classifier with its different distance metrics. Efficiency of the system is evaluated by using statistical tests and experiments that resulted in average F measure of 86,925%.

2.2 Radar based object detection systems

A large number of active object detection systems available today are solely or partly radar based. This technology has a long history of application and it has developed from being a means of providing an early warning system of airborne attacks during WWII to the level of portable and adaptable systems that can support a wide variety of distant sensing applications. Undoubtedly, there are many differences between target tracking in airborne applications and vehicle tracking for active safety systems. For automotive active safety and detection systems, distances to the objects of interest are generally in a range of few tens of meters, while this distance is tens of kilometres for airborne radar applications to track aircrafts [14].

Radar based object detection systems can be implemented for various purposes. Some of these applications are examined, development principles and efficiency of developed projects are discussed in the following paragraphs.

Advanced Driver Assistance Systems (ADAS) and Autonomous Driving (AD) systems are one of the most common application fields of radars. This is also a very popular research topic due to its possibility of expanding the area of application. ADAS and AD systems are mostly assisted with different sensor kits including camera, lidar, radar and etc. Compared to others, radars are capable of operating in a wide range of weather and lighting conditions. In this type of applications, more than one objects exist in the field of view (FOV) of the radar, and the aim of it is to detect and track each relevant object. To handle this task efficiently, authors of [15] applied a non-linear motion model for the objects that are tracked by the ego vehicle. Grid-based DBSCAN was used to cluster multiple target detections per object. Also, a simple and efficient data association approach (JPDAF and UKF - proven scenarios for multiple object tracking) was adopted. The overall structure is verified according to simulation tests developed by PreScan.

Ensuring road safety, protecting pedestrians and avoiding traffic accidents are considered as main goals of ADAS scientists. For addressing this problem, it is crucial to develop a system that controls a vehicle by monitoring and identifying objects moving around it with sensors on-board [16], [17]. Authors of [16] proposed a solution by using mmwave radar sensor. This is an on-road object detection method taking into consideration time series of radar data. Process starts with extraction of features from the 79,5 GHz mmwave radar information, e.g. velocity, distance, and signal power by time windows. Then, the mean and variance were calculated for each feature and each window. Using a time window, made capturing the changes of object specific signals over time possible. On-road experiments were carried out with a radar-equipped vehicle to evaluate the classification performance. According to the gathered results, proposed

features significantly contribute to the accurate identification and developed method achieved 10% performance improvement compared to conventional (camera-based) approach.

Another radar based safety system is described in [17]. This is a new approach of using a remote sensor to provide multifunctional safety in a vehicle. The system includes a remote sensor located adjacent to the rear corner of the vehicle. This sensor is configured to detect objects by transmitting and receiving radar waves at a predefined angle, and within a predefined range. A control module is set to receive signals from the sensor and determine the velocity, severity of impact and likelihood of the object impacting the vehicle through a calculated approach vector of a detected object. In such a state, the control module starts to compare the severity of this impact with the pre-determined threshold value, and also configures an impact algorithm to initialize safety systems of the vehicle upon the object crossing a determined distance threshold.

As already indicated above, radar technology has been widely applied in surveillance systems. In [18], authors designed and implemented a lighter, safer and smaller radar system which is suitable for using to monitor public places. They tested the system that detects a single target to show the accessibility of this technology to further applications. For making possible to use it in public, system is designed to fulfil all the relevant safety requirements. The system uses two coffee cans for each transmit and receive antennas and fabrication of RF components into a single RF board. Since this RF board supports higher power transmission, it leads to better overall performance. In advance, power supply, voltage regulator and power distribution parts supporting the power consuming components are implemented in the single PCB. Developed detection system is capable of targeting class 1 UAVs which are popularly adopted to people and maneuvered for recreational purposes and also can detect big objects such as car and a person. Experiments were carried out in different settings (listed as versions in the paper) and in indoor and outdoor circumstances. It can be concluded that during those experiments the maximum capability of the radar to detect a vehicle was around 50 m, a person was around 30 m, and a drone was around 17 m. However, this capability is expected to be improved by taking some actions. Possible ways of getting better results are reducing the noise by using improved hardware or applying noise-reducing algorithms in the postprocessing phase, also placing the system in location where ground and clutter interference is minor can be beneficial.

2.3 Radar and camera sensor fusion

A crucial task for safe and reliable surveillance technologies is to possess an accurate and fast detection and tracking capabilities. In order to achieve this, different sensors and devices can be deployed together or separately. Most widely used technologies are radar sensors, cameras and lidars. Each of these technologies has different advantages and disadvantages, and they need to be applied by considering both kinds of characteristics.

Compared to radar and lidar, camera provides more detailed and rich semantic data of the objects, similar to that by the human eye, but it does not preserve accurate depth information. Cameras are also sensitive to light and weather conditions, and they show low detection accuracy especially for the range and velocity detections [19].

On the other hand, lidar provides depth information and a 3D view of the object's surroundings, but it is also sensitive to weather conditions, such as fog, heavy rain or snow. Lidars are more expensive compared to cameras and radar sensors and it is an important factor for many automotive and industrial projects [19].

An important advantage of radars over camera and lidar-based systems is that radars are capable of penetrating non-metal objects such as glass, plastic, clothing and they can operate robustly in various environmental conditions (rain, dust, smoke and etc.) since they detect objects by emitting radio signals and analysing this echo in the reflected signal. FMCW radars can work in complete darkness or bright daylight (radars are not affected by glare) as well. Compared to ultrasound-based devices, radars usually have a longer range and faster time of transit for their signals [20]. However, mmwave radar measurements are limited in terms of angular and spatial resolution and they can possibly have false detections due to noise [21].

It is obvious that for fulfilling all relevant requirements for efficient object detection and tracking, separate usage of these technologies is insufficient due to above listed drawbacks. Owing to these limitations, a combination of radar and camera sensor - sensor fusion technology is developed to increase target detection performance levels and guarantee higher accuracy while reducing detection noise [21].

Table 1 below provides a general overview of the evaluation of camera and radar technologies and radar-camera sensor fusion.

Table 1. Evaluation of camera, radar and radar-camera sensor fusion technologies performance [19], [20]

	Camera	**Radar**	**Radar-camera sensor fusion**
Distance estimation	Fair	Good	Good
Range of visibility	Fair	Good	Good
Object localization	Fair	Good	Good
Classification	Good	Limited	Good
Velocity (Radial)	Limited	Good	Good
Velocity (Lateral)	Fair	Limited	Good
Angle estimation	Good	Fair	Good
Functionality in poor weather conditions	Limited	Good	Good
Functionality in poor lighting	Fair	Good	Good
Cost	Good	Good	Good

In [22], authors present a collaborative fusion method to get the optimal balance between vehicle detection accuracy and computational efficiency. They used fusion of mmwave radar and monocular camera for on-road vehicle detection and tracking. The system was assessed with a real-world dataset collected by the designed intelligent vehicle platform (Kaufu-II). First, mmwave radar detects the potential vehicles and provides a region of interest (ROI) to the image sequences gathered by the camera. Then, the vision processing module starts to identify the vehicle inside the ROI provided by the radar by generating a square boundary in the image frames and employing active counter method. If this method fails, it is considered as a false alarm of mmwave radar and vision processing module eliminates this detection. Radar processing module also generates a trajectory by using the radar data. Generated two trajectories of the radar and camera are further compared to verify and confirm if the detection and tracking results are valid. The test results show that the developed system can achieve a 92.36% detection rate and 0% false alarm rate under a real-world dataset.

In [23], a mmwave radar and an onboard camera were used to develop a sensor fusion system for a forward collision warning system in vehicles. Fusion technology was applied in order to compensate the deficiencies caused by relying on a single sensor and to

improve frontal object detection rates. While tracking the object for removing non-object noise, DBSCAN and particle filter algorithms were implemented in the radar detection subsystem. Also, system is able to identify an object in front of a vehicle as one of three main categories (car, pedestrians and motorcycles) with the two-stage vision recognition subsystem. Data obtained from two subsystems was integrated. A radial basis function neural network was used by the spatial alignment to learn the conversion relationship between the distance data by the radar and the coordinate data in the image. Then, a neural network was applied for object matching. Experiments were carried out in three different environmental conditions (daytime, night-time, and rainy-day) to test the performance of the proposed method. According to the results, proposed sensor fusion system achieved the detection rates around of 90,5% and the false alarm rates around of 0,6%. These rates are better than those obtained by a single sensor system (radar subsystem – 66,6% and image subsystem – 67,8%).

2.4 Object tracking fundamentals and methods review

Object/target tracking is a type of data processing used to interpret the environment based on observations from one or more sensors and estimating the state of the object that is present in the scene [24]. This sensor can be any measuring device that is used for collecting data about objects in the environment, such as radar, lidar, camera, infrared sensor, microphone, ultrasound and etc. [25].

Figure 4 describes the steps of multi-object tracking process. Process includes localizing multiple moving objects using a camera, assignation of unique identity to each detected object, and formation of motion trajectories of objects based on these identities.

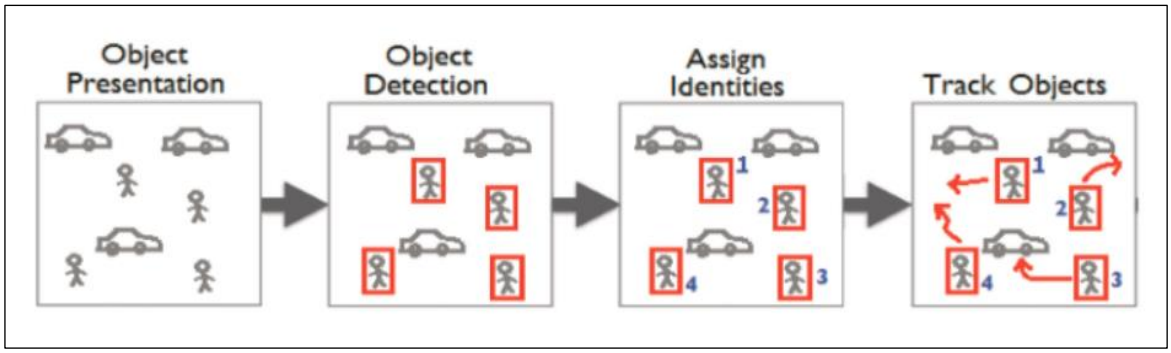


Figure 4. Multi-object tracking steps [26]

Most of the modern tracking methods follow the "Tracking by Detection" scheme where objects are found and located in the scene and then corresponding tracklets (position of the object in the next frame) of them are determined [27].

An object tracking system is composed of an object or objects to be tracked, a sensor which measures some aspect of the object, a signal and data processor, as depicted in Figure 5.

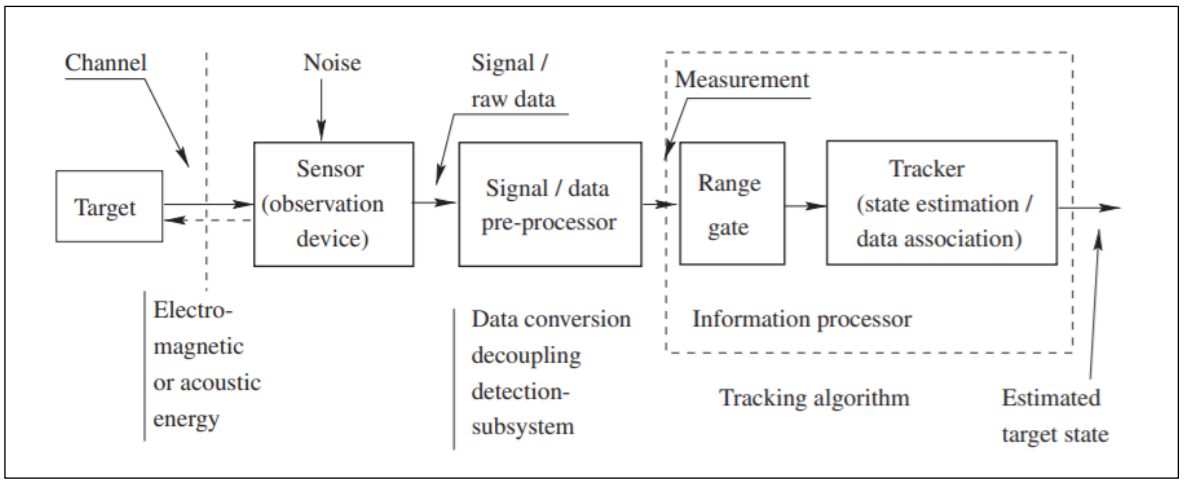


Figure 5. Object tracking system block diagram [25]

Typically, main goals of the tracking process are determining the number of objects, their identities and states, positions, speed ranges, features and variations due to geometric changes such as pose and etc. For example, for a radar tracking system of aircraft, the goal is to determine the number of aircraft in a region under observation, their types-e.g. military or commercial, identities, velocities and positions, all based on data obtained from a radar. Some of main application fields of object tracking are air space monitoring, video surveillance, weather monitoring, ADAS and etc. [25].

An important challenge in object tracking is the fragmentation issue. It occurs when detection of objects is missing in some frames and therefore fragmented trajectory is generated. One solution is eliminating or reducing the number of missing detection frames by improving the detection algorithm. Another way is using appearance and localization features of objects. Since objects in the frame have a different location and different appearance relative to each other, based on these features it is possible to build an algorithm which can track all objects in a series of frames. Based upon motion and appearance features, trajectory fragmentation of one object can be related to the other fragmentation of that object and the trajectory path can be completed [26]. This process is described in Figure 6.

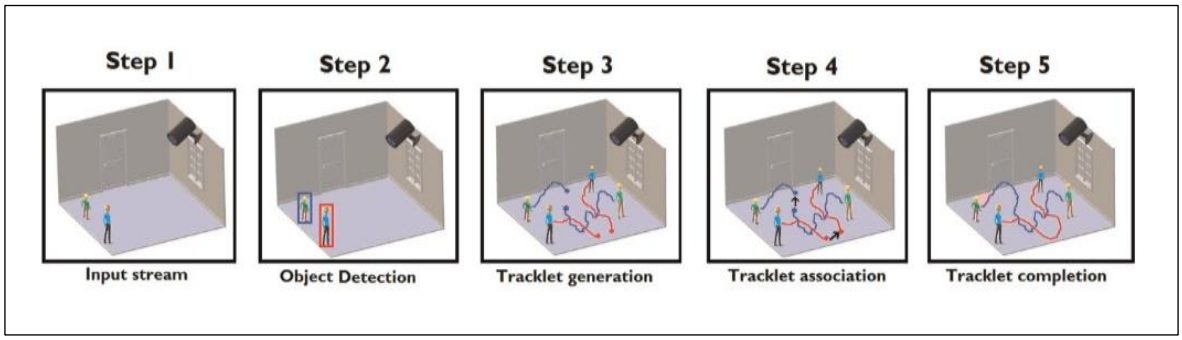


Figure 6. Fragmentation removal process [26]

Understanding human activity from video data has gained great importance in last few years and now it is one of the most significant research areas in computer vision technology. This task is important due to the role it is playing in fighting against crime and terrorism, public safety and efficient management of accidents. There are three main steps in visual analysis: detecting interested moving objects, tracking these objects from each and every frame to frame, and analysing object tracks to determine their behaviour. The powerful PCs, accessibility of top quality and low-cost cameras and the expanding need for automated video analysis technologies have created a huge interest in object tracking algorithms [28].

Object tracking is also described and affected by different factors such as illumination variation, co-ordinates matching, environmental issues, variety of tracked objects, pose variation, occlusion, motion blur etc. According to [10], tracking is classified into three parts: Point Tracking, Kernel Tracking and Silhouette-based object tracking. Each method is explained and their advantages and shortcomings are listed. All methods have their own advantages but one single method is not able to deal with all the problems. Since object tracking is an integral part of video surveillance systems, it is necessary to review all existing methods and approaches in order to solve existing challenges and the paper is important in terms of helping to understand all the basics to design an efficient tracking algorithm that can be tested in different situations.

In [29], authors described the main concepts of object tracking. A single Moving Object Tracking method was tested with one person moving inside the room and Multiple Moving Object Tracking was tested with a scene which has a moving car and two people. Multi-object tracking system is divided into three parts; visual tracking, track management and online model learning.

Two methodologies were used to track the objects: first depends on correspondence matching and second based on a distinct tracking. Main limitations of the research are that it’s not useful where higher keyframes are required for object detection and it’s not suitable for track moving objects in denser environments such as crowds of people.

2.4 Conclusion

Different techniques on object detection, classification and tracking are described in the above paragraphs. Various algorithms for each method have been tested, experiments carried out and results of works are described. It can be stated that, selection of the overall approach, proper hardware and detection algorithm are the most essential tasks for developing a system. Since each algorithm and method is unique, they can act as intended only if their capabilities are enough for the given conditions and the detection of targeted objects.

ADAS and AD systems are among the fastest-growing segments in automotive electronics due to the steadily increasing adoption of industry-wide quality and safety standards. This growth is also reflected in development of modern technologies used for object detection systems. Application of the radar based detection systems to maintain safe autonomous driving of smart vehicles is a very popular research topic nowadays. Radar and camera sensor fusion technology is considered as a novel and more efficient approach for on-road object detection purposes. Some of these projects and experimentations were examined and described in this chapter.

On the other hand, application of sensor fusion technique for object detection and tracking in surveillance systems has not been proposed according to conducted literature review.

Some of the researches described in the chapter state limitations of the process of using radar for detecting objects. Main concerns are about low detection quality due to noise level, sensitivity of used sensors, some difficulties related to detection of humans and analysis of human movement, financial cost of used technologies and so on. To address these issues, defining all the necessary specifications of the used sensor and the developed hardware, and data collecting phase with it is done by considering those limitations and examining previously tested technologies and methods.

Since most of the object-detection systems are developed for autonomous driving systems, developing a system by using sensor fusion approach will attempt to ensure a high level of security for the physical structures and facilities of vital importance.

3. HARDWARE

3.1 mmWave radars

mmWave radars are radars operating in the millimeter-wave band. Generally, the millimeter wave refers to an electromagnetic wave in the frequency domain of 30–300 GHz (wavelength of 1–10 mm) [30]. This band of spectrum is also known as the extremely high frequency (EHF) band by the International Telecommunication Union (ITU). A mmWave system that operates at 76–81 GHz frequency range (corresponding wavelength is around 4 mm) is able to detect movements that are as small as a millimeter fraction [31].

Main factors that make usage of mmWave sensors efficient:

- Ability to sense through different materials such as clothing, drywall and plastic and etc.
- Capability of forming compact beam with 1° angular accuracy.
- Ability to be focused and steered using standard optical methods.
- Ability to distinguish multiple nearby objects [32].

Based on the transmitted signal waveform, radars can be classified into the following categories [33], [34], [35]:

1. Pulsed radars - they emit high power and high-frequency pulsed signals towards the target object. Before sending another pulse, it waits for receiving an echo signal from the object. The range and resolution of the radar depend on pulse repetition frequency. Pulsed radars use the Doppler shift method.

2. CW (Continuous Wave) radar - instead of pulses, electromagnetic radiation is emitted without any interruption. The echo signal is received and processed simultaneously. These radars don’t determine the distance of the target but rather the rate of change of range by measuring the Doppler shift of the received signal. CW radars are mainly applied for speed measurement.

3. FMCW (Frequency-Modulated Continuous Wave) radars - a special class of radar sensors that transmit a frequency-modulated signal but constant in amplitude. Signal transmission is done continuously for measuring the range, angle and velocity. The working principle differs from pulsed-radar systems, which transmit short pulses periodically as shown in Figure 7.

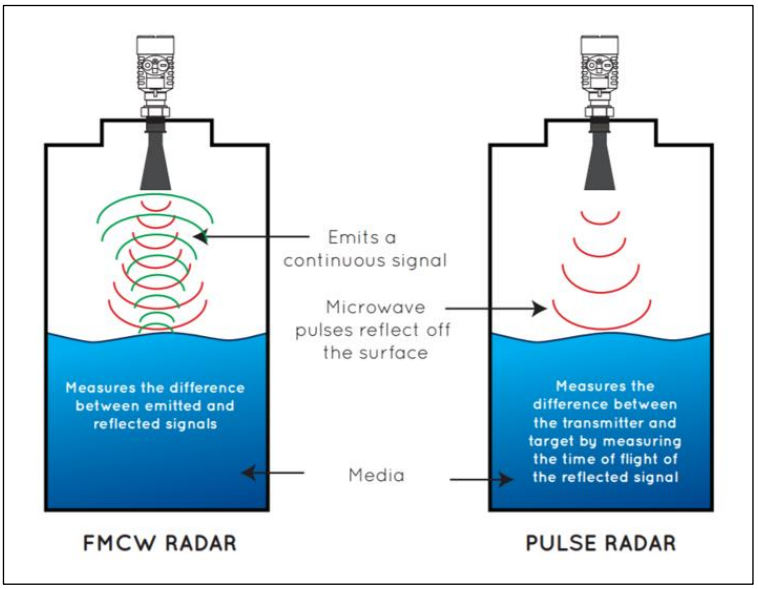


Figure 7. Differences between working principles of FMCW and pulse radars [36]

The main advantage of the FMCW radar is that evaluation phase takes place without a time out for reception and therefore, the observation results are continuously available.

Electromagnetic signal that is transmitted by radar is also called a chirp. In the chirp used in FMCW radars, the frequency increases linearly with time [31].

3.2 General working principles

A complete mmWave radar system is composed of receiver (RX antenna) and transmitter (TX antenna); radio frequency (RF) components; analog components such as clocking; and digital components such as analog-to-digital converters (ADCs), microcontrollers (MCUs) and digital signal processors (DSPs) [31]. Block diagram of radar components is described in Figure 5.

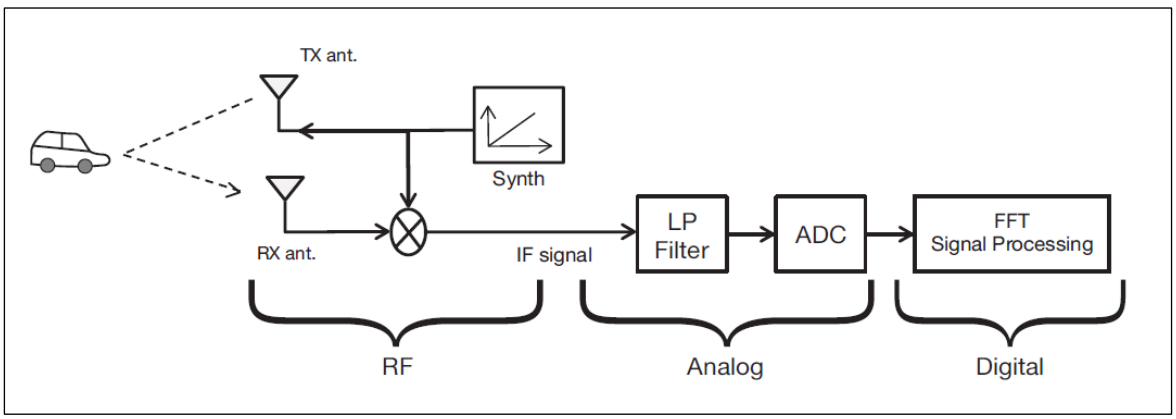


Figure 8. Block diagram of radar components [31]

Operating principle of these components can be briefly explained according to the above described block diagram. First, a signal (chirp) is generated by the synthesizer (synth) circuit. Then TX antenna sends out a chirp signal. Reflection of this signal by a detected object sends a reflected chirp and it is captured by RX antenna. Frequency mixer produces an IF (intermediate frequency) signal from a combination of two – RX and TX signals. Mixer has three ports in total: two inputs and one output. For two sinusoidal inputs (x_1 and x_2) an output sinusoid (x_{out}) is generated. These 2 sinusoidal input signals x_1 and x_2 are shown below.

$$x_1 = \sin(\omega_1 t + \Phi_1) \quad (1)$$

$$x_2 = \sin(\omega_2 t + \Phi_2) \quad (2)$$

Instantaneous frequency and phase of x_{out} output is equal to the difference of instantaneous frequencies and phases of two input signals:

$$x_{out} = \sin[(\omega_1 - \omega_2)t + (\Phi_1 - \Phi_2)] \quad (3)$$

At the end, for an object that is located at a d distance from the radar, the intermediated signal will be the following sine wave:

$$A\sin(2\pi f_0 t + \Phi_0) \quad (4)$$

where $f_0 = S2d/c$ and $\Phi_0 = 4\pi d/\lambda$. f_0 is the start frequency, S slope of the chirp, d the distance to the detected object, c is the speed of light and Φ_0 initial phase of IF signal.

Then the IF signal is digitized by ADC, and FFT (Fast Fourier transform) is performed on the ADC data. This signal consists of multiple tones, the location of peaks in the frequency spectrum (frequency of each tone) is directly proportional to the distance of the corresponding objects as described in Figure 9 [37].

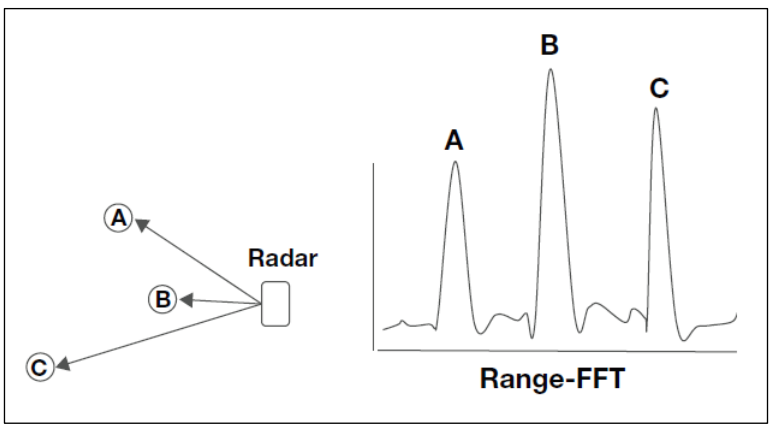


Figure 9. The locations of peak frequencies and corresponding object ranges [38]

3.3 AWR1843BOOST Evaluation Module

Since the project mainly focuses on monitoring traffic and performing detection, tracking and classification procedures for predefined vehicle classes on various traffic situations, the main selection criteria included possible detection ranges, range resolution, relative velocity detection and velocity resolution, possibility of using in robust environments, software and data processing options and market price. Initial performance requirements for the radar are up to 35 meters of measurement range, 0,2 meters of range resolution, more than 22 m/s (79 km/h) radial velocity detection, around 0,76 m/s (2,7 km/h) radial velocity resolution, 45° FOV and 15 FPS frame rate.

Selected radar hardware for the project is the AWR1843BOOST Evaluation Module provided by Texas Instruments [39]. It successfully meets above mentioned requirements and is fully capable of operating for intended measurement purposes. The module is described in Figure 10 below.

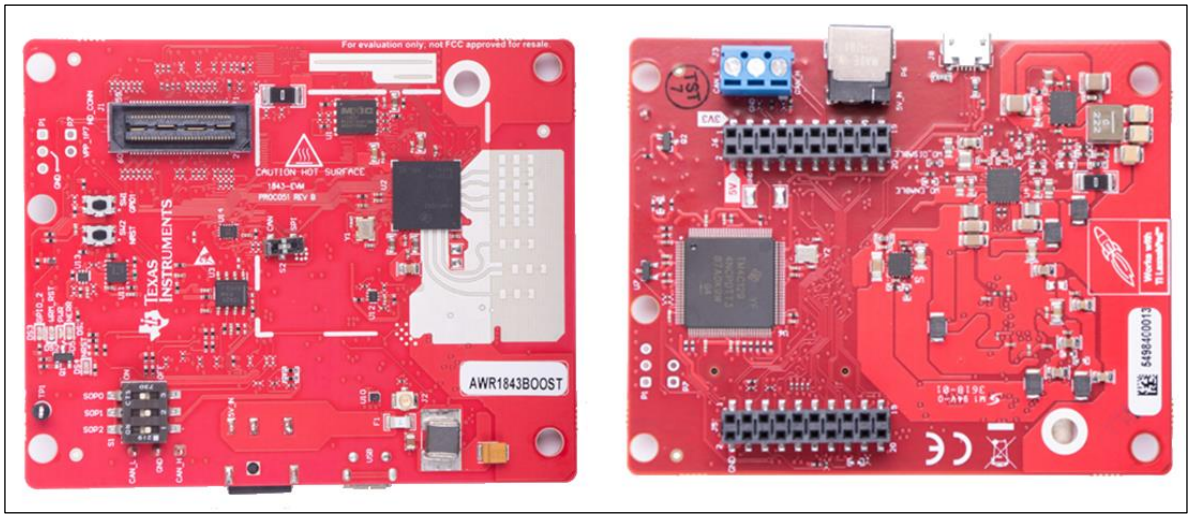


Figure 10. Front and rear sides of the EVM

Main features and components of the AWR1843BOOST EVM are given below and the block diagram is described in Figure 11.

- The module is provided with AWR1843 integrated single-chip FMCW radar sensor. This sensor operates in the 76-GHz to 81-GHz coverage with 4 GHz available bandwidth, designed via TI's low-power 45-nm RFCMOS process and enables unprecedented levels of integration in an extremely small form factor. AWR1843 has 25 Msps ADC sampling rate, CAN, CAN-FD, I2C, QSPI, SPI, UART interfaces and 2 MB memory. Operating temperature is in 40 to 125 °C range. The sensor is considered to be an efficient solution for self-monitored, low-power, ultra-accurate radar systems in the automotive space [40], [41].

- Power management circuit which provides 5V input voltage to all the required supply rails [20].
- Board has our onboard receive channels and three transmit channels.
- Onboard XDS110 that provides a JTAG interface, UART1 for loading the radar configuration to the sensor, and UART2 to send the object data back to the PC [20].

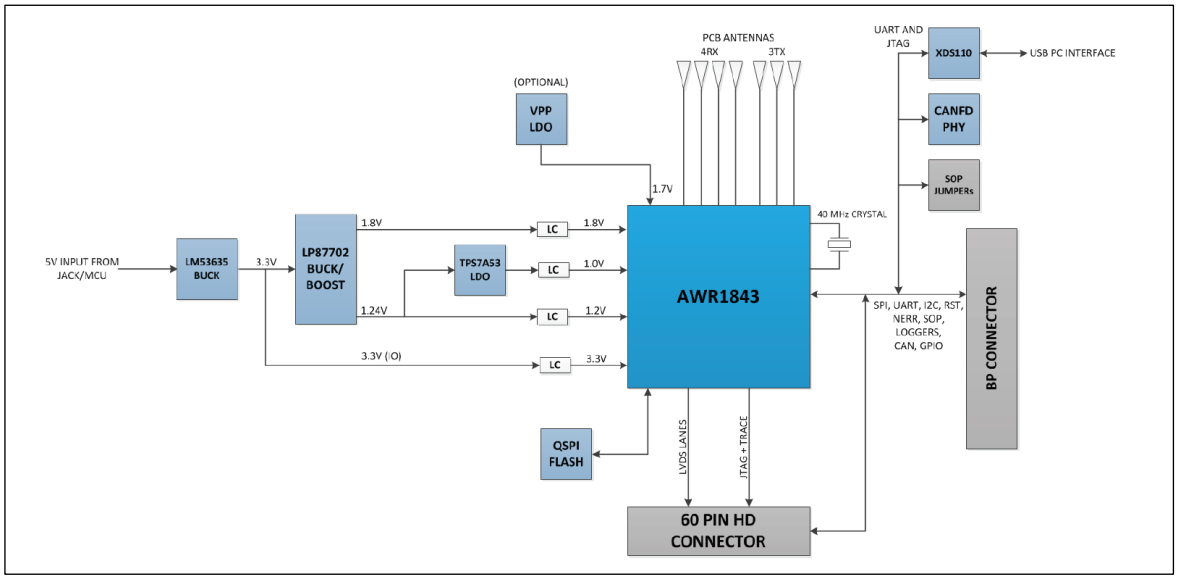


Figure 11. AWR1843BOOST EVM block diagram [20]

AWR1843BOOST EVM is supported with mmWave tools and software packages including mmWave Software Development Kit (SDK), mmWave Studio, mmWave Sensing estimator and mmWave Demo Visualizer.

The mmWave Software Development Kit is a unified software platform provided by Texas Instruments. SDK has relevant tools to provide easy setup and fast out-of-box access to the project evaluation and development. mmWave SDK also has industrial and automotive toolboxes that are open-source software examples and documentation for developing a project and to better understand the application performance [42].

mmWave Studio is a free platform that presents tools for making different experiments with the out-of-box demo. It provides post-processing and visualization of ADC data, and MATLAB based post-processing examples [43].

mmWave Sensing estimator cloud is a GUI-based online tool for configuring the radar according to given parameters [44].

mmWave Demo Visualizer is a GUI-based online tool for real-time plotting of observed point cloud and making relevant configuration changes according to application need [45].

3.4 Data acquisition system

Hardware used for the data acquisition is composed of AWR1843 radar, NVIDIA Jetson Nano and MIPI camera. NVIDIA Jetson Nano is a small, powerful computer running neural networks. It is mostly used in image classification, object detection, segmentation, and speech processing applications. The platform runs in 5 W, has local storage for data saving and is capable of connecting to internet [46].

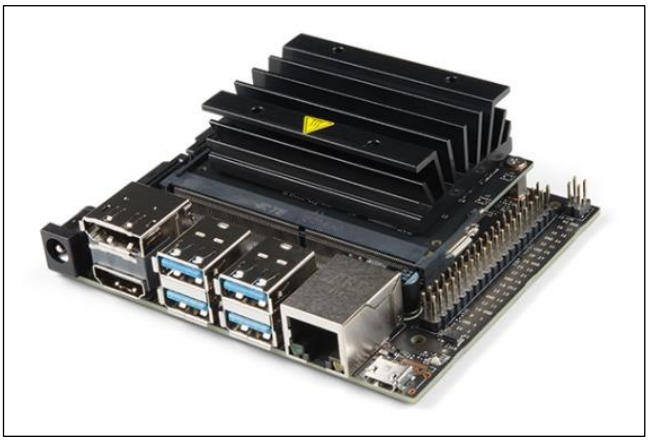


Figure 12. NVIDIA Jetson Nano board

Used camera supports 45° FOV, 3264 x 3264 pixels resolution and 15 FPS frame rate. Radar and camera sensors both are connected to the Jetson Nano central unit. Camera is connected to the board with MIPI CSI 2 high-bandwidth interface cable, which allows data and power transfer, while radar is connected with a USB cable. Radar and Jetson Nano are powered externally with a power source of 2.5A at 5V. Hardware connectivity diagram and internal structure of the hardware are shown in Figures 13 and 14 below.

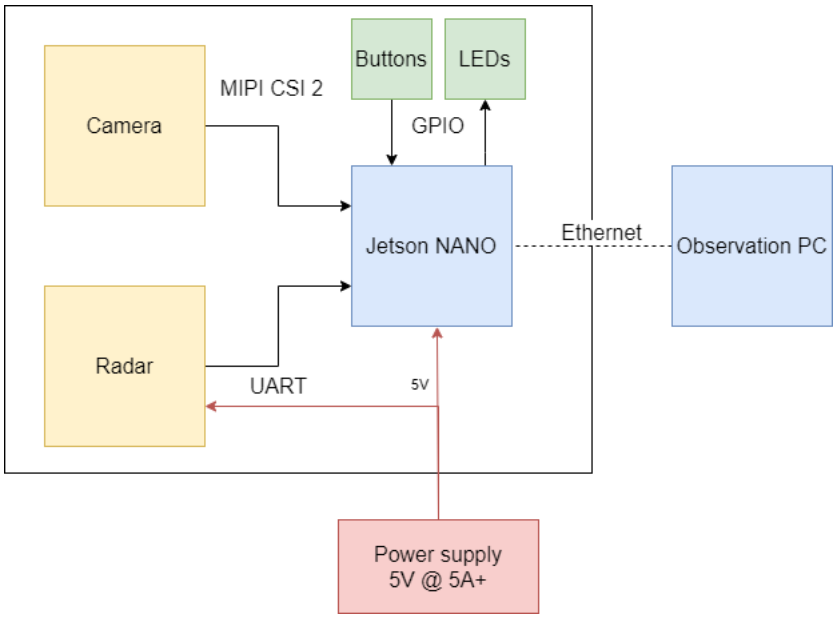


Figure 13. Hardware connectivity diagram

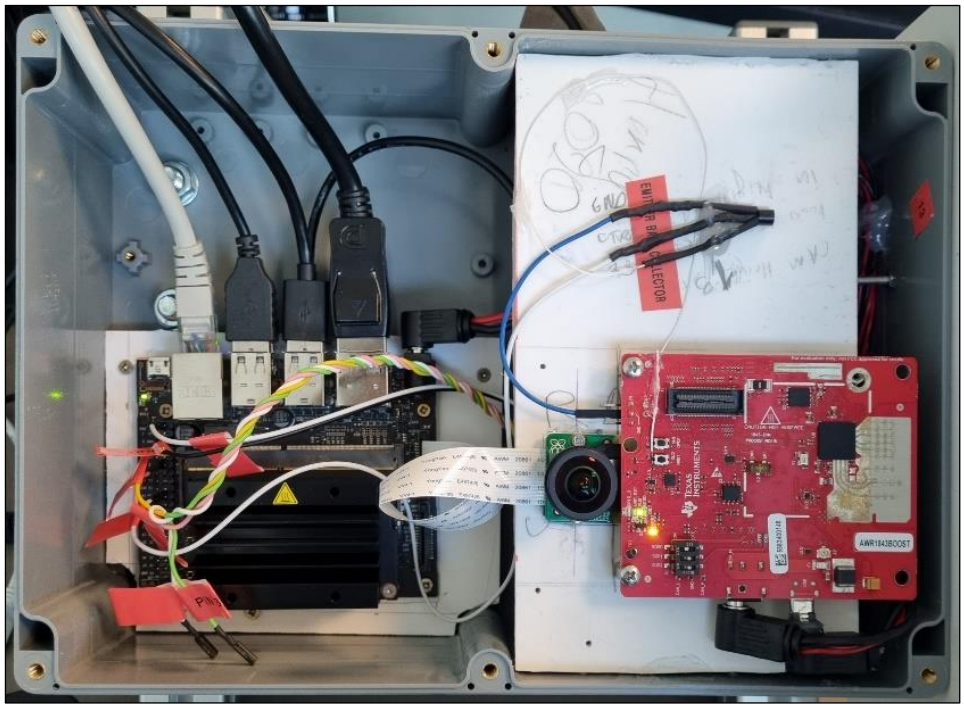


Figure 14. Internal structure of the hardware

ROS (Robot Operating System) is being used for data processing - saving the collected data from sensors to the BAG file and playing it back when needed. The system displays the radar data as a point cloud using its visualization tool. Process is described in Figure 15 below.

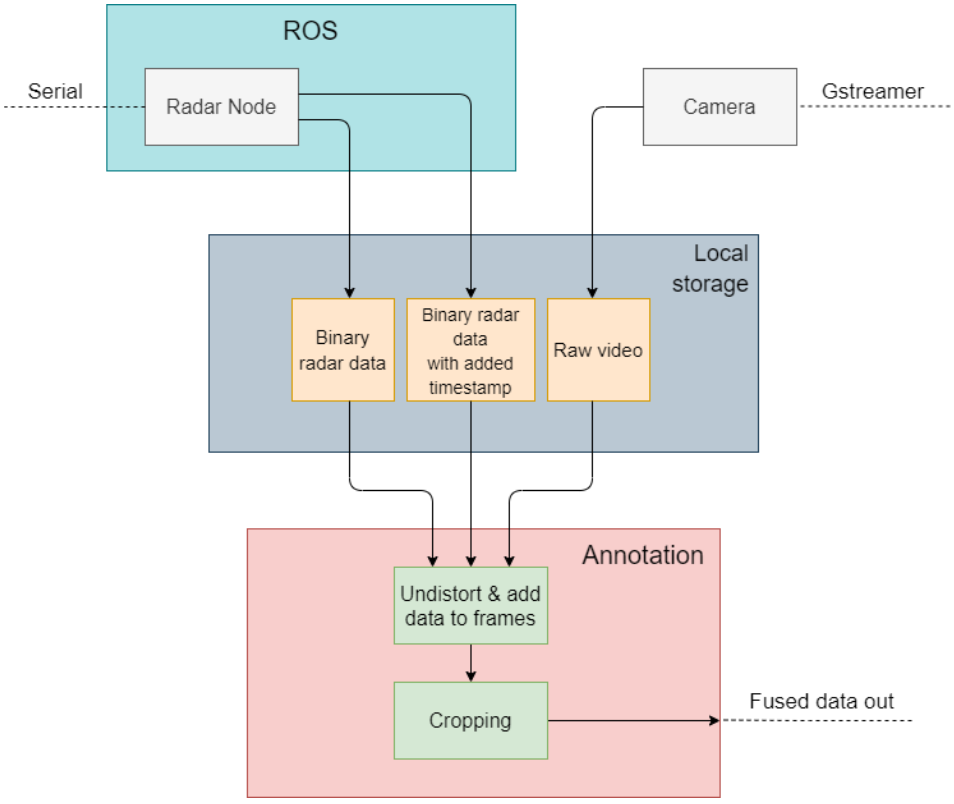


Figure 15. Block diagram of ROS flow

4. DEVELOPMENT

4.1 Configuring the radar

Configuring the radar is a process of defining parameters that will make the radar perform according to intended requirements. mmWave SDK provides different tools for effective configuration and assessment procedures. Out of Box Demo from Industrial Toolbox was used to specify sensing characteristics of the radar, tune and assess them and visualize the results of the detection by using real-time plots interfaces [45], [47]. General diagram of connections between the EVM and Demo Visualizer is described in Figure 16 below.

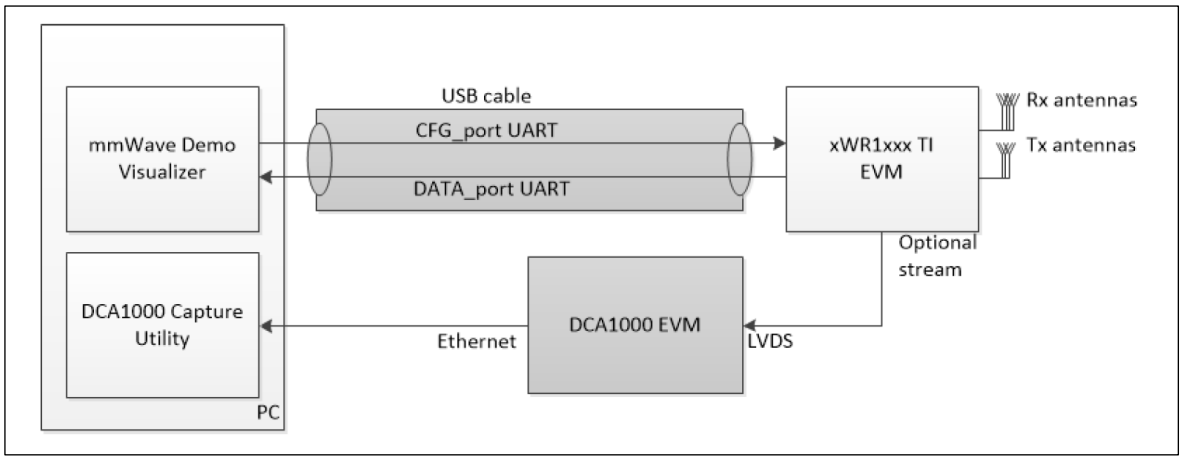


Figure 16. Connectivity between mmWave Demo Visualizer and EVM [48]

Main part of the GUI is configure tab, which is composed of setup details, scene selection and plot selection sections (Figure 17). After selecting the proper platform, SDK version and antenna configuration, desirable configuration – scene classifier has to be chosen from the relevant drop-down menu. This menu reflects the parameter that a user is mostly concerned about. GUI offers 3 possible scene classifiers and tunes the system towards the selected one. They are Best Range Resolution, Best Velocity Resolution and Best Range. Best Range classifier is chosen as a desirable scene classifier since the goal of the project is to detect and track vehicles moving from medium to long ranges in various traffic situations.

Transmit chirp configuration and RF transceiver performance, antenna array design, processing power, available memory and key configuration parameters affect overall radar performance. Some of these key parameters are briefly described below [20], [49].

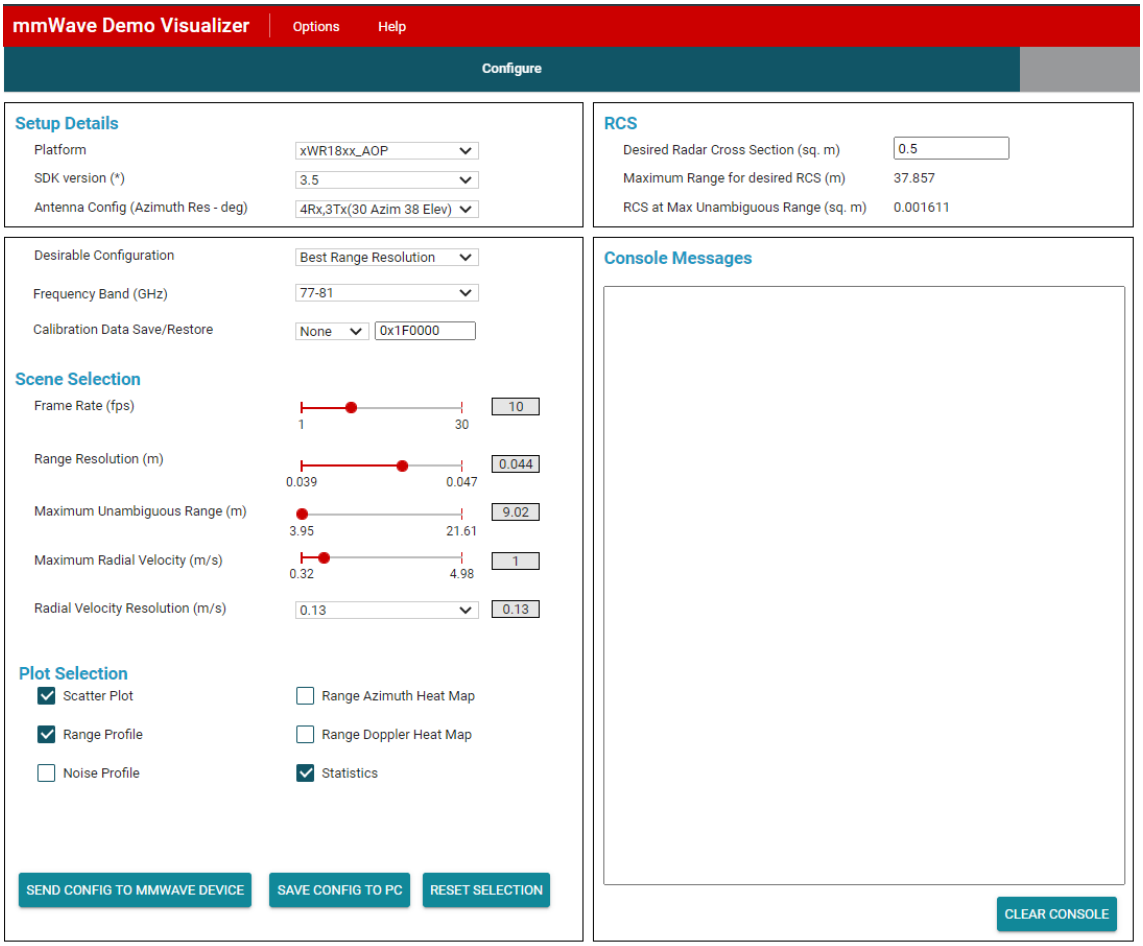


Figure 17. Configure tab of the mmWave Demo Visualizer

Maximum Range - the maximum beat frequency in the de-chirped signal detected in the RF transceiver and signal-to-noise ratio (SNR) threshold of the received signal determine the maximum practical range value. Maximum Unambiguous range is selected based on the farthest distance expected to detect objects. 35 meters of maximum unambiguous range is defined for the configuration used for the process of tuning the radar tracker. Selecting lower values for this parameter results in fewer but finer options for the range resolution value.

Range resolution - this value is determined by the bandwidth of the chirp frequency sweep and it is selected based on the minimum range difference expected to be distinguished by the detector between two or more individual objects. The higher values for chirp bandwidth correspond to fine range resolution capability. 0,586 m range resolution is determined for the used radar configuration. Selecting finer values for this parameter provides lower values and fewer options for maximum radial velocity.

Maximum velocity – in the low-level processing chain the radial velocity is measured and then maximum unambiguous velocity is determined by the chirp repetition time within one frame. While configuring the radar, the possible maximum velocity of the

objects in the measurement range must be considered and this value must be selected according to that. 23,05 m/s (82,98 km/h) is the defined maximum unambiguous velocity value for the radar configuration. Selecting lower values for this parameter results in finer velocity resolution capability for the radar.

Velocity resolution - this value is determined by the total chirping time in a frame and it is the minimum velocity difference expected to be distinguished by the detector between two or more individual objects. Defined value for this parameter is 0.37 m/s (1.332 km/h). The longer chirping time provides finer velocity resolution options.

Field of View (FOV) - this value is the sweep of angles that determines an observable area where radar transceiver can effectively detect targets. Typically, FOV is separately specified for the elevation and azimuth.

Angular resolution - this value is determined by the number and geometry of the TX and RX antennas and it is a minimum angular difference that can be distinguished by the detector between two or more individual objects which are in the same range and have the same velocity. Typically, angular resolution is separately specified for the elevation and azimuth.

Frame rate - this is the rate at which the collected radar data is shipped out from the radar. Since the camera used for the data acquisition system works at 15 FPS frame rate, the same rate is also applied for the radar while configuring it. Therefore, frame duration corresponds to 66,667 milliseconds.

Second part of the visualizer is the plots tab. Provided plot options are X-Y and 3D Scatter Plots, Doppler Range Plot, Range Profile, Noise Profile, Range-Azimuth Heatmap, Range-Doppler Heatmap and statistics.

The final radar configuration (Appendix 1 – AWR1843configuration.cfg) was created as a result of the development process which includes taking into consideration above stated parameters, applying various observation methods by using provided plots, conducting relevant indoor and outdoor tests and comparing acquired results.

4.2 Implementation

Implementation process consists of three radar processing layers: Front End Processing, Low Level Processing and High Level Processing layers. They are implemented on the radar hardware and visualization and scene interpretation phases are done in MATLAB-based GUI at the host computer as shown in Figure 18 [50].

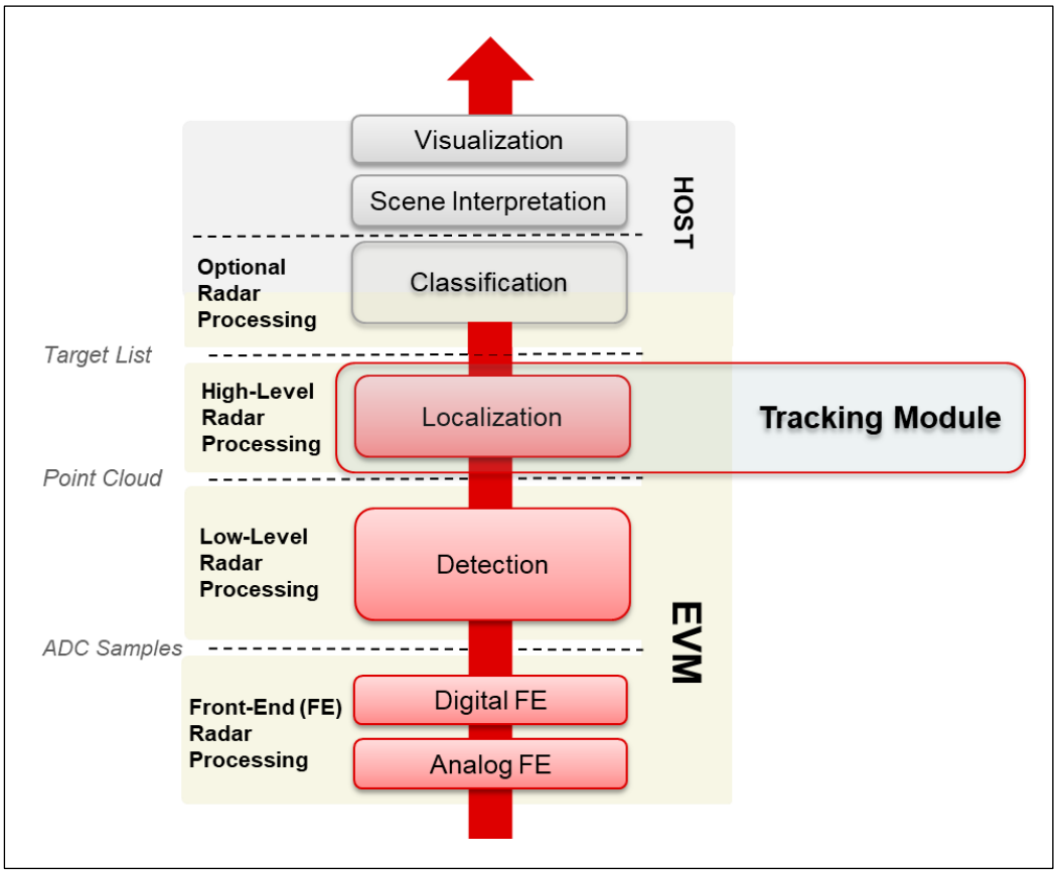


Figure 18. Radar processing layers and traffic module in overall chain [50]

Detection, tracking and visualization signal-processing chain is composed of range processing, Doppler processing, Range-Doppler detection algorithm and angle estimation blocks (Figure 19) [5].

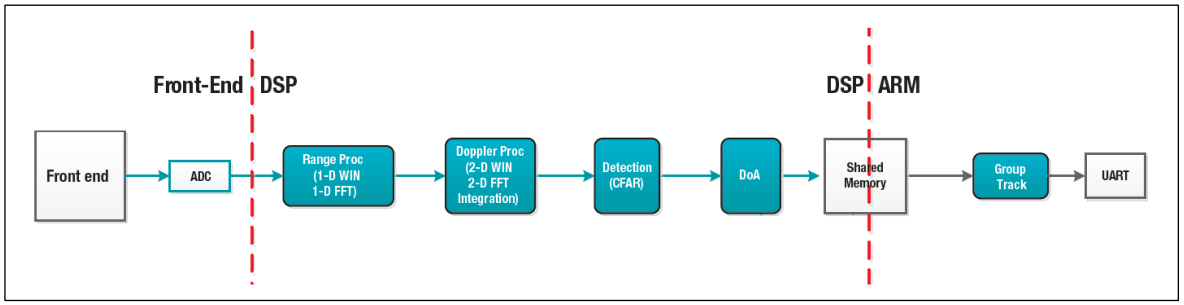


Figure 19. Detection, tracking and visualization signal-processing chain [5]

4.2.1 High level radar processing

High level radar processing layer deals with providing an effective tracking capability to the EVM. Traffic monitoring requires detection of various real-world targets that also can present multiple fluctuating reflection points. Therefore, tracking module is applied for tracking point cloud (a group of points) that belongs to a particular object instead of tracking a single individual point. Input point cloud data from the low level processing layer is copied from the shared memory and sent to the tracker of high level processing layer which is implemented in ARM R4F core of the AWR1843 radar sensor (Figure 20) [50].

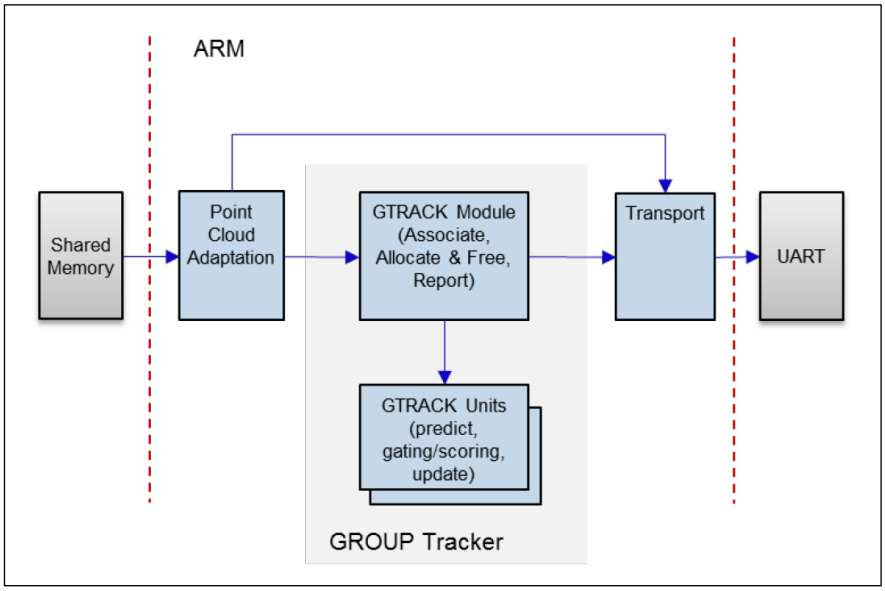


Figure 20. High level processing flow [50]

Group tracker is composed of two sub-layers: a module layer where points from input point cloud are attempted to be associated with a tracking unit and non-associated points go for allocation stage and a unit layer where Extended Kalman Filter (EKF) is used for each track for prediction and estimation the group properties [50].

4.2.2 Radar tracking module

Working principle of the radar tracker is based on providing localization information to the classification layer by processing point clouds coming from detection layer. Typically, this information is a trackable object with properties such as velocity, range, position, density of point cloud, physical dimensions and etc., and used for making a relevant identification decision in the classification layer. Functional block diagram describing the

main steps of the group tracking algorithm and general group tracking approach are described in Figures 21 and 22 respectively [51].

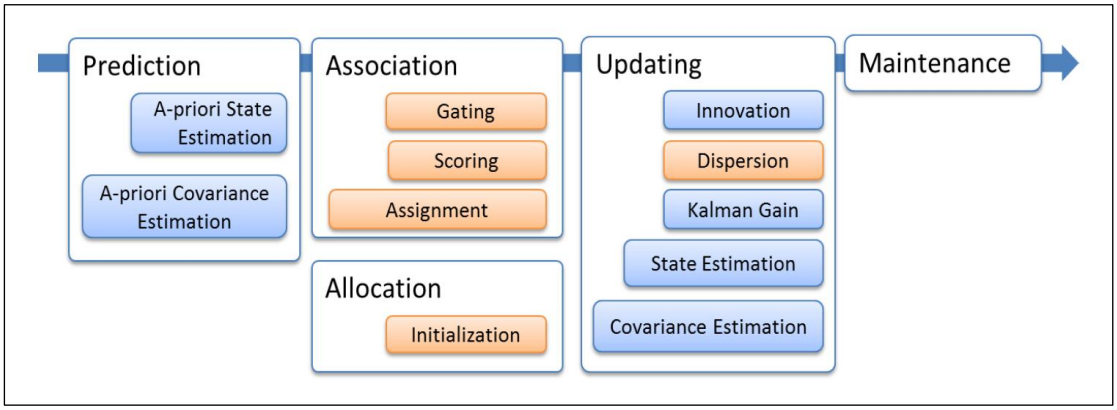


Figure 21. Group tracking block diagram [52]

Classical EKF operations are described in blue blocks and blocks of additional steps for support multipoint grouping are given in orange. A brief explanation of these blocks is provided according to Figure 22 [53].

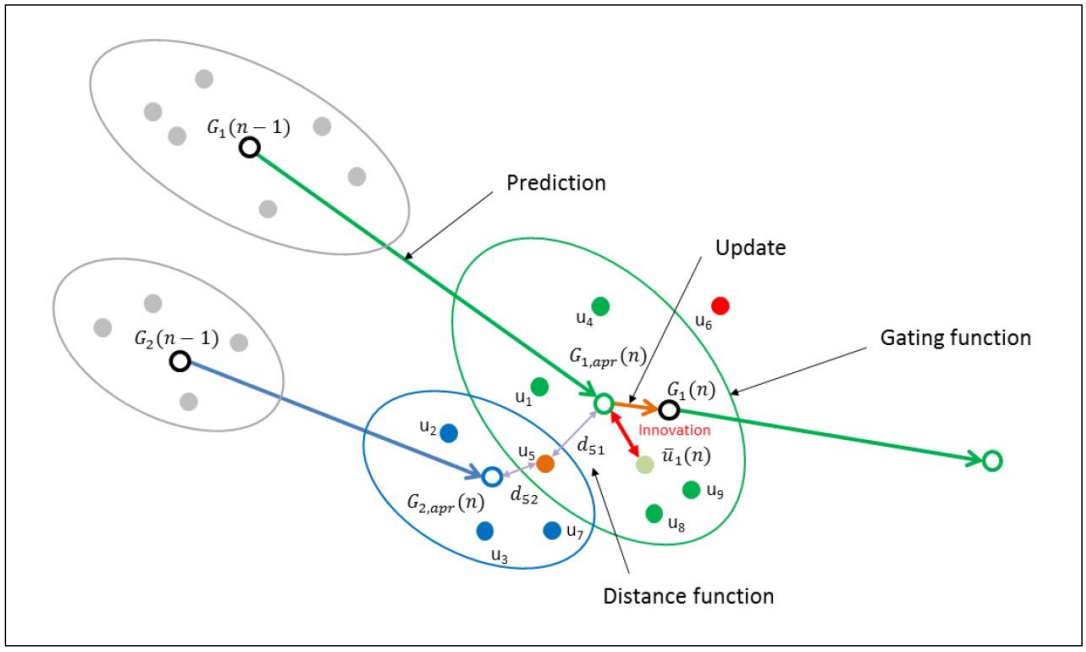


Figure 22. Group tracking approach [53]

In the figure above, tracking process of two objects is described. G_1(n-1) and G_2(n-1) are centroids of two tracks belong to that objects at n-1 time instance. Process is explained step by step below [51], [53].

Prediction step – in this step, the centroids of tracks are predicted for time n based on model. These centroids are marked with $G_{1,apr}$(n) and $G_{2,apr}$(n).

Association step – in this step, gates around the predicted centroids are constructed by a defined gating function. Next to that, obtained measurement vectors set [u_1,…u_9] must be associated with a relevant track. After determining the distance between the measurement vectors and predicted centroids, vectors are assigned to the track that is in the closest distance. Briefly, [u_1, u_4, u_5, u_8, u_9] vectors were assigned to the green track and [u_2, u_3, u_7] vectors were assigned to the blue track as a result of the gating test. In the figure, d_{51} is the distance from vector u_5 to the green tack's centre, and d_{52} is the distance between vector u_5 and the blue track's centre. Since, this step checks target manoeuvre, dispersion of the group and measurement noise while forming a gate, the u_5 measurement vector was assigned to the green track as a result of relevant tests and applied likelihood function in scoring phase before a bidding score is assigned to it. These tests check if the vector is in measurement limits of the particular track and include determination and comparison of velocity, distance and etc. parameters.

Allocation step – in this step, a new tracker is allocated for leftover measurement vectors that are not associated with any track. Process includes relevant checking procedures and making an allocation decision according to the results. Measurements sets that pass the qualifying criteria of the test are assigned to the new track, otherwise, they are ignored.

Updating step – in this step, mean u(n) value of associated points is determined. In the figure, this value is labelled as $u_1(n)$. The difference between $u_1(n)$ mean of the measurement vectors and $G_{1,apr}(n)$ predicted centroid is called innovation and it is a measure of the amount of differences between the tracker's predictions and real observations. The track centroid is updated to $G_1(n)$ by using this innovation measure.

Maintenance step – in this step, the state of the track that is not active any more can be changed or the track can be completely removed (de-allocated).

4.3 Group tracker configuration

In this subchapter basis of tuning the radar tracker is discussed and selection procedure of the parameters of tracker commands, their role in overall process and how they affect the results are described. Group tracking algorithm is composed of the steps described in Figure 23 and it is provided with configuration parameters that cover the sensor geometry, scenery, behaviours and various features of targeted objects. EKF process is applied within a tracking process since the group tracking algorithm tracks objects in Cartesian space but input data is entered in Polar coordinates [50].

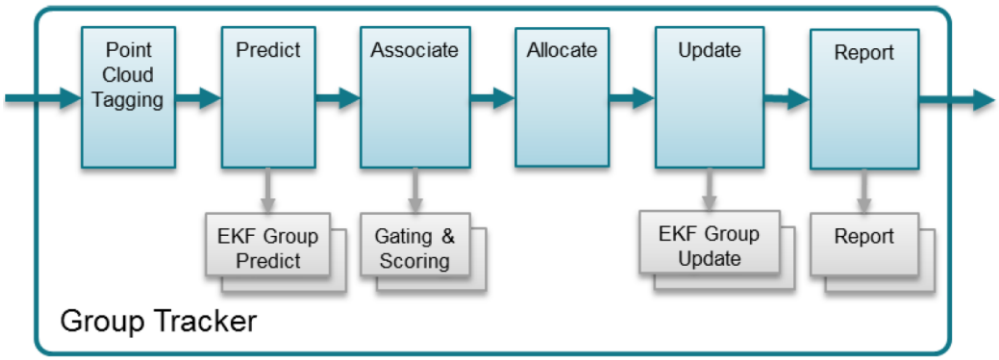


Figure 23. Group tracking algorithm [50]

In order to find the most optimal radar tracker performance for tracking the targeted objects in predefined measurement settings, more than 40 experiments were conducted in various environments. Each measurement set focused on a preselected configuration parameter and the most suitable value was determined according to the results of measurements.

Measurements were taken in various locations, different times of day and weather conditions, traffic situations such as heavy and easy-going traffic, with different hardware mounting settings and developed radar configurations. Selected spots for measurements were the electric pole facing the road next to it and by mounting the device from 2- and 3,2-meters height, and the bridge, where device was mounted from 8,1 meters height facing the road underneath.

Process of identifying the final tracker commands based on preparatory measurements is described in following paragraphs and the detailed analysis of measurement results are provided in the next chapter.

Configuration parameters are classified as mandatory and optional (advanced) parameters. Main parameter sets are scenery parameters, gating parameters, allocation parameters, state transition parameters and maximum acceleration parameters.

4.3.1 Scenery parameters

Scenery parameters are used to define the boundaries of the area where the radar tracker is expected to operate and where it will perform static behaviour. Radar sensor's position and angular orientation are defined by scenery parameters as well. These parameters also affect how the output data will be visualized in GUI. Tracker doesn't use the measurement points that are detected outside the defined boundaries. Before starting to configure this parameter, dimensions of measurement space, mounting

position of the sensor, proper horizontal and vertical tilt angles were estimated, found and according to the results of initial preparatory measurements these values were further optimized and finally scenery parameters were determined.

Following spaces are defined in the sensor mounting geometry as shown in Figure 24.

World Space W: {X_w, Y_w, Z_w} a Cartesian coordinate system, the origin is O at ground level.

Tracker Space T: {x_t, y_t, z_t} a Cartesian coordinate system, the origin is at sensor.

Point Cloud Space P: {r, φ, θ, $\dot{r}$} a Spherical coordinate system, the origin is considered to be the centre of the radar sensor's antenna virtual array. Here, *r* - radial distance, φ - azimuth angle, θ - elevation angle and $\dot{r}$ - radial velocity of the detected point with respect to the sensor axis [53].

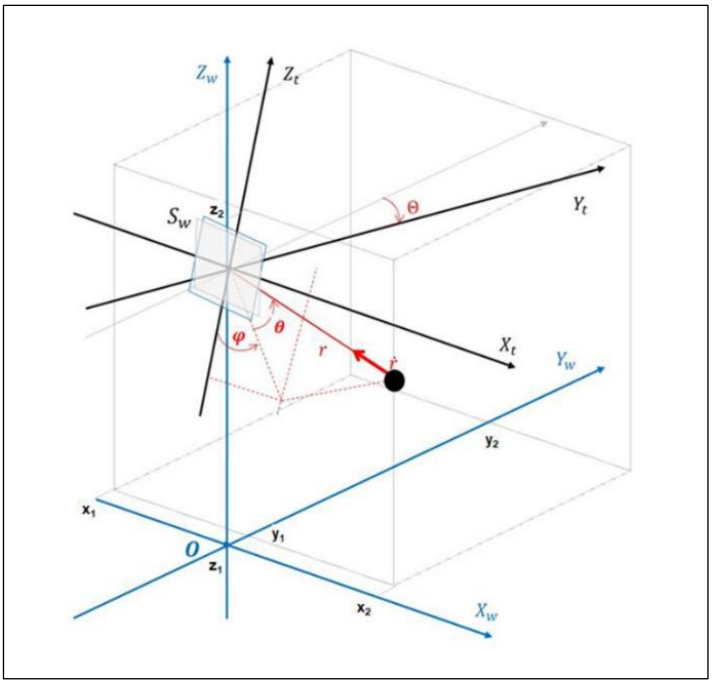


Figure 24. Sensor mounting geometry [53]

Sensor location is S_w = {0, 0, H}, where H is the height of the radar sensor from the ground. This value is specified in the World Cartesian Space. Depending on the measurement spots, H was defined as 2, 3,2, 3,5, 8,1 and etc.

In case sensor is tilted by an θ angle clockwise about the X_w axis, then the rotation matrix $R_x(\theta)$ is

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\theta & sin\theta \\ 0 & -sin\theta & cos\theta \end{bmatrix}$$

Radar tracker operates in a Cartesian coordinate system, output point cloud data is reported in Spherical coordinate system and tracker output data is visualized in World space. Therefore, transformation between these spaces is required in some occasions within a tracking procedure [53].

Transformation from Spherical *P: {r, φ, θ}* to Cartesian *T: {x_t, y_t, z_t}* system:

$$x_t = rCos(\theta)Sin(\varphi),\ y_t = rCos(\theta)Sin(\varphi),\ z_t = rSin(\theta)$$

Transformation from tracker space *T: {x_t, y_t, z_t}* to World space *W: {x_w, y_w, z_w}* for tracker output visualization:

$$\begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} = R_x(\theta) \begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ H \end{bmatrix}$$

Transformation from World space *W: {x_w, y_w, z_w}* to tracker space *T: {x_t, y_t, z_t}* for boundary box conversions:

$$\begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} = R_x(-\theta) \left(\begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ H \end{bmatrix} \right)$$

Scenery parameters used in radar configuration and a process of determining their values are briefly explained below [50], [51], [52].

Boundary Box – it is used to define the boundaries of the space in which the radar tracker operates and a track can exist. *boundaryBox* CLI command is given below.

boundaryBox X_{min}, X_{max}, Y_{min}, Y_{max}, Z_{min}, Z_{max}

X and Y define horizontal and vertical distances from the radar sensor respectively and Z values define height of the radar sensor from the ground. They all are measured in meters and defined with respect to the O origin of the World coordinate system. Selected values for boundaryBox parameters are given in Table 2 below.

Table 2. boundaryBox parameters

boundaryBox	**X_{min}**	**X_{max}**	**Y_{min}**	**Y_{max}**	**Z_{min}**	**Z_{max}**
	-10	10	2	35	-8.1	2

Values were set by considering the total width of the road, maximum unambiguous range capability of the sensor and height of the spot where the device was mounted.

Static Boundary Box – it is used to define the area where detected targets are expected to persist static for some time. *staticBoundaryBox* CLI command is given below.

staticBoundaryBox X_{min}, X_{max}, Y_{min}, Y_{max}, Z_{min}, Z_{max}

X and Y define horizontal and vertical distances from the radar sensor respectively and Z values define height of the radar sensor from the ground. They all are measured in meters and defined with respect to the O origin of the World coordinate system. Selected values for staticBoundaryBox parameters are given in Table 3 below.

Table 3. staticBoundaryBox parameters

staticBoundaryBox	**X_{min}**	**X_{max}**	**Y_{min}**	**Y_{max}**	**Z_{min}**	**Z_{max}**
	-7	7	2	30	-8.1	2

From Table 1 and 2, it can be seen that static boundary boxes are smaller than boundary boxes. The reason for that is to make it possible for the tracker to determine if the tracks have gone static or the object has actually exited the area bounded by boundary box values. If this is the case, tracker can quickly delete those tracks.

Position of the EVM sensor with respect to X, Y, Z orientation axes is described in Figure 25 below.

Figure 25. EVM sensor position with respect to X, Y, Z orientation axes

Sensor position and orientation – it is used to define the orientation and position of the radar sensor. *sensorPosition* CLI command is given below.

sensorPosition sensorHeight, azimTilt, elevTilt

sensorHeight is the height from the ground to radar sensor in meters, azimTilt defines the azimuth tilt angle around the Z_w axis and elevTilt defines tilt angle of the radar sensor around X_w axis in degrees. Selected values for sensorPosition parameters are given in Table 4 below.

Table 4. sensorPosition parameters

sensorPosition	**sensorHeight**	**azimTilt**	**elevTilt**
	8.1	0	18

Figure 26 below is a screenshot taken from GUI and shows visualization of scenery parameters.

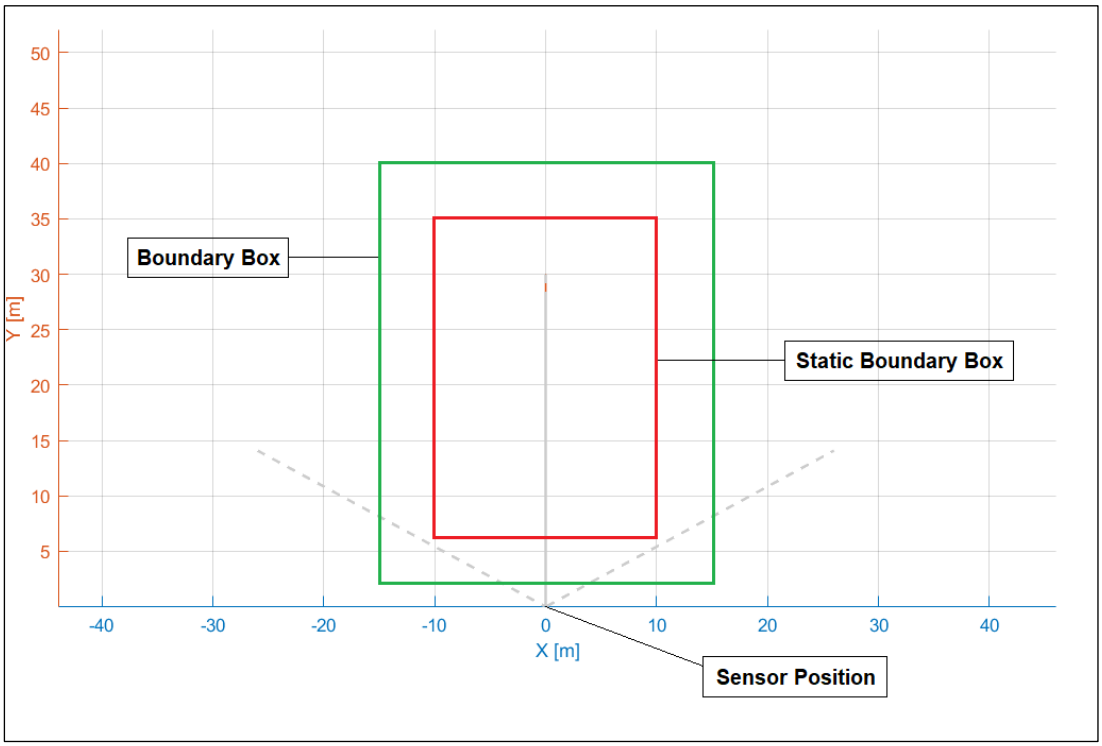


Figure 26. Visualization of the defined scenery parameters in GUI

Dimensions of boundary boxes and the proper mounting position of the radar sensor play an important role in achieving accurate tracking performance. Improper selection (e.g. too closely located boundaries, radar mounting with wrong vertical angle and etc.) of these values can result in late removal and disappearance of some tracks on the interface as shown in Figure 27.

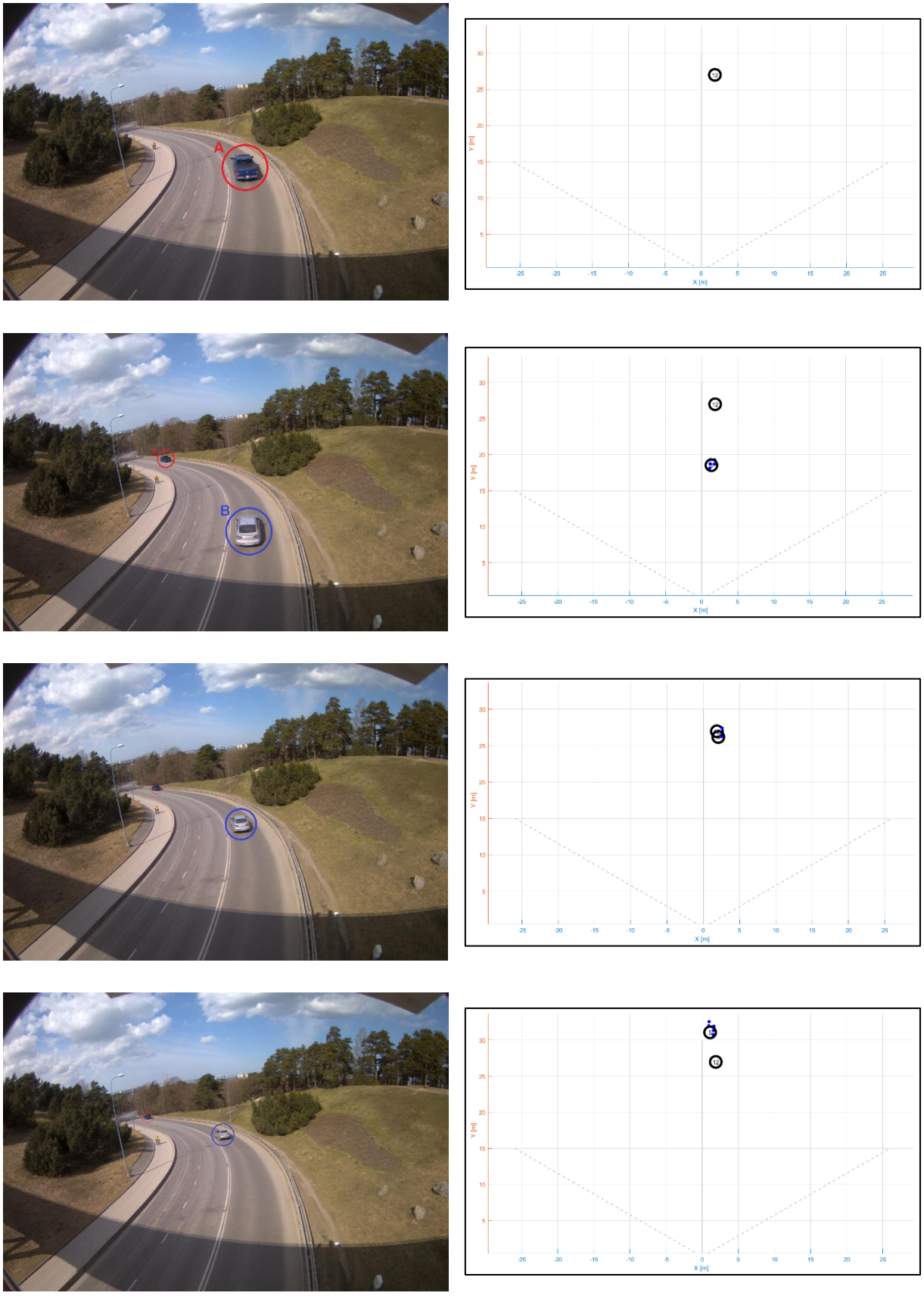


Figure 27. A situation where a track belongs to vehicle A does not exit the defined range on time, and causes mix-up with the track belongs to vehicle B

4.3.2 Gating parameters

Gating parameters are used in association step to form a boundary or gate around the predicted centroid of detected points. The points that can be associated with a particular track are determined by this formed boundary around the track centroid. *gatingParam* CLI command is given below.

gatingParam Volume, Limit-Length, Limit-Width, Limit-Height, Limit-Velocity

These commands determine the maximum volume, dimensions - width, length and height limits, and the maximum velocity of a tracked object. Points that are detected beyond the given limits are not involved in making up tracks. Gain defines the track gating limits and it is estimated as the volume of the ellipsoid. Width, length, height and velocity limits are multiplied by the gain value and determine a gating function for the track [51]. These values are set by considering physical dimensions of objects that need to be tracked, in meters. Selected values for gatingParam parameters are given in Table 5 below.

Table 5. gatingParam parameters

gatingParam	**Volume**	**Limit-Length**	**Limit-Width**	**Limit-Height**	**Limit-Velocity**
	16	8	3	3	0

Gating volume (gain) can be computed with $V = \frac{4\pi}{3}abc$ formula, where a is expected dimension of targeted object in range (m), b in an angle (rad), and c in doppler (m/s) [50]. For instance, if a target object is a car and located ± 3 meters far from the sensor (a=6), ± 3 degree in azimuth (b = 6π/180), and ± 4 m/s in radial velocity (c=8). As a result, volume will be approximately 21. According to results of tests taken with different values, 16 was set as a value of volume.

Finding the most optimal values for physical limits parameters is the critical part of setting the process and directly affects the tracking quality. Tests show that, giving too small values for these limits results in formation of more than one tracks for a single object (Figure 28) and too higher values result in allocating only one track for multiple closely located or moving objects (Figure 29).

Limits for physical dimensions were formed by considering the most realistic values for targeted objects dimensions and optimized based on several test results. Since the biggest vehicle that regularly passes through the measurement area was MAN Lion's City GL/A40 bus with 18,75 x 2,5 x 2,98 m dimensions [54] and the smallest one was

a car with around 4,2 x 1,9 x 1,8 m dimensions, final dimension limits are set 8 m for width, 3 m for length and 3 m for height. 0 was set for velocity limit since 0 defines no limit for the particular parameter.

Figure 28. Two tracks for a single object, caused by too small dimension limits

Figure 29. One track for two objects, caused by too higher dimension limits

4.3.3 Allocation parameters

Points from detected point cloud that don't get associated with any existing tracking instances become subjects for the allocation procedure. Each of these points is clustered into a set of allocation candidates and this is determined by allocation parameters. A candidate point must be within a maximum distance and maximum velocity threshold range from the centroid of allocation set. These sets must have more points than points estimated for points threshold (pointsThre) parameter and pass SNR and maximum velocity thresholds [50], [51], [52]. *allocationParam* CLI command is given below.

allocationParam snrThre, snrThreObscured, velocityThre, pointsThre,
maxDistanceThre, maxVelThre

Flow chart of the allocation procedure is described in Figure 30.

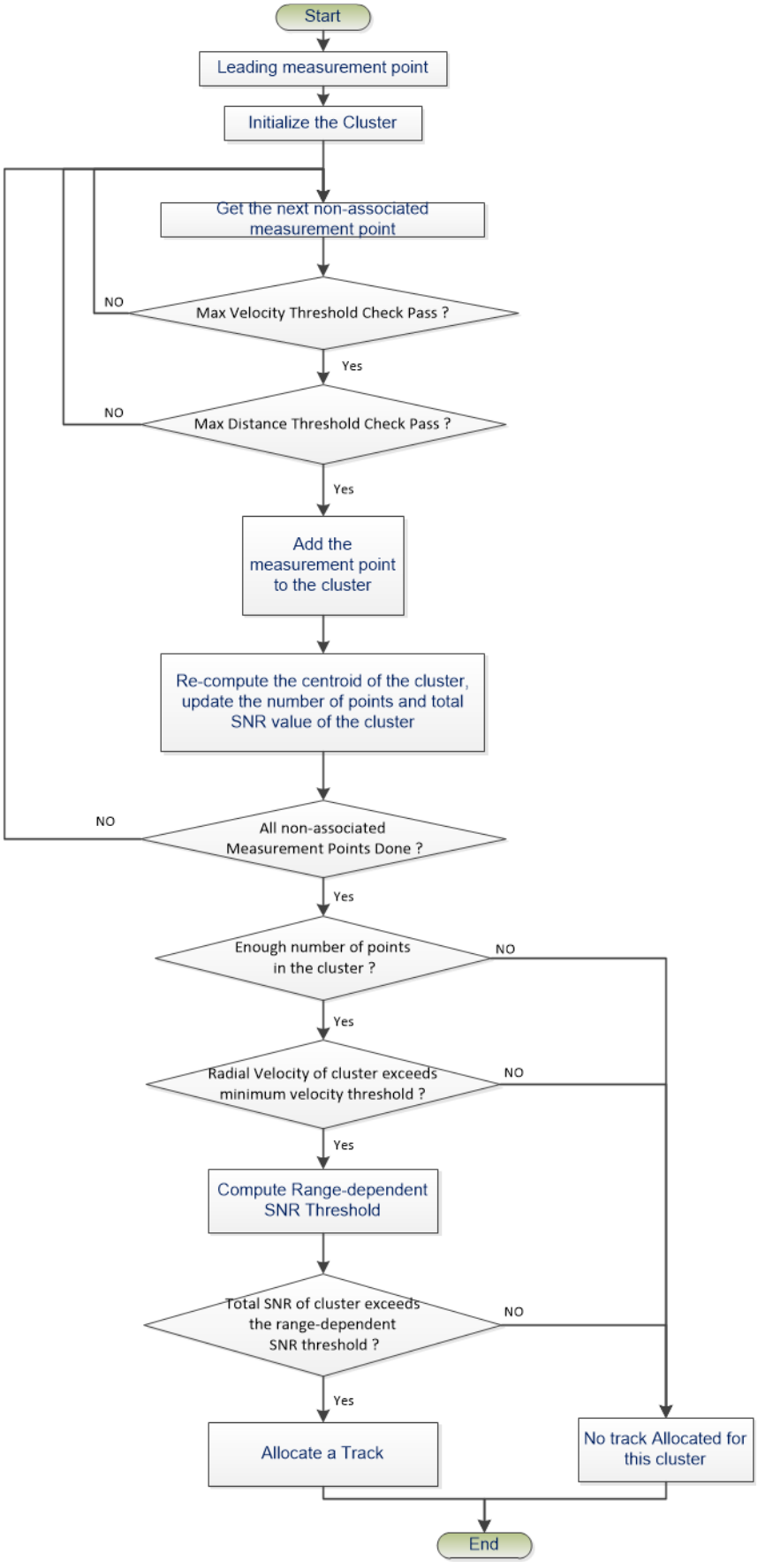


Figure 30. Allocation procedure flow chart [53]

Selected values for allocationParam parameters are given in Table 6 below.

Table 6. allocationParam parameters

allocation Param	snrThre	snrThre Obscured	velocity Thre	points Thre	max Distance Thre	maxVel Thre
	50	70	1	5	5	40

SNR threshold value defines the amount of minimum total SNR value of the allocation set-candidate. Summing SNR values of all the allocation set members in linear scale forms the total SNR value of the set [51]. This value is optimized at 50, as results have shown that lower SNR values resulted in false detections – ghosts caused by multipath reflections.

SNR Obscured Threshold value defines the amount of minimum total SNR of the allocation set when obscured by another target. In case an existing track is directly located between the sensor and a track that has a similar Doppler to an existing track, then this track is considered as obscured track [51]. This issue was seen often in measurements taken in a measurement spot, where the radar sensor was facing the road from the side and vehicles were passing in front of it. Similar issue didn't happen in measurements taken from the bridge, as there, radar was mounted perpendicular to the road where vehicles were coming towards and moving away from it. The most optimal value for this parameter is considered to be 70.

Velocity threshold defines the minimum limit for the velocity of the centroid of allocation set in m/s. velocityThre parameter is set as 1 m/s for the final configuration. Setting such a low value for the velocity threshold parameter helped the tracker to continuously track objects, especially vehicles that move very slowly and make tiny movements [51], [53].

Points threshold defines the minimum amount of points in the allocation set, this is regulated by estimating the minimum possible amount of points of a potential target [51]. Measurements show that, most of the times, minimum number of points for an object is 5.

Maximum Distance Threshold value defines the maximum squared distances (in m^2) between a candidate point and a centroid of an allocation set [51]. The most optimal value for this parameter is decided to be 5 m^2, after some measurements taken with higher maxiDistanceThre values. Setting lower values resulted in properly located tracks for detected objects.

Maximum Velocity Threshold defines the maximum velocity difference (in m/s) between a candidate point and centroid of the allocation set [51]. This difference is set as 40 m/s for the configuration. This helped to get more accurate tracks for point clusters and measurements with higher maxVelThre values caused less differentiation between closely located or moving objects.

4.3.4 State Transition parameters

State transition parameters determine the transition between 3 possible states of a track instance. These are FREE, DETECT and ACTIVE states. Two types of events can happen for a tracking instance within a frame: HIT or MISS events. HIT event refers to a situation where a tracking instance has non-zero detection points associated and in a MISS event, a tracking instance is not associated with any points [50]. *stateParam* CLI command is given below.

stateParam det2actThre, det2freeThre, active2freeThre, static2freeThre, exit2freeThre, sleep2freeThre

Flow diagram of the process is described in Figure 31 below and explanation of the transitions is given based on the diagram.

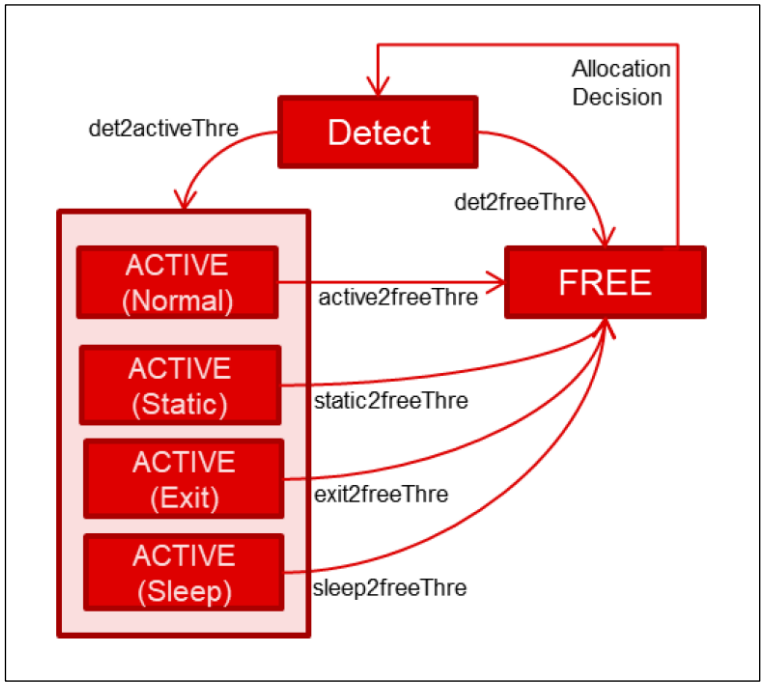


Figure 31. State transition flow [53]

Transition from FREE to DETECT state is done according to relevant allocation decision. Transition from DETECT to ACTIVE state is realized when consecutive HIT events happen in an amount of det2active threshold. Transition from DETECT to FREE state is realized when consecutive MISS events happen in an amount of det2free threshold. Transition from ACTIVE to FREE depends on condition of a target in ACTIVE state. It can be in any of Normal, Exit, Sleep or Static conditions. For these state transitions static2free, exit2free, sleep2free, active2free threshold values are used [51], [52].

Flow charts describing these transitions are given Figures 32 and 33.

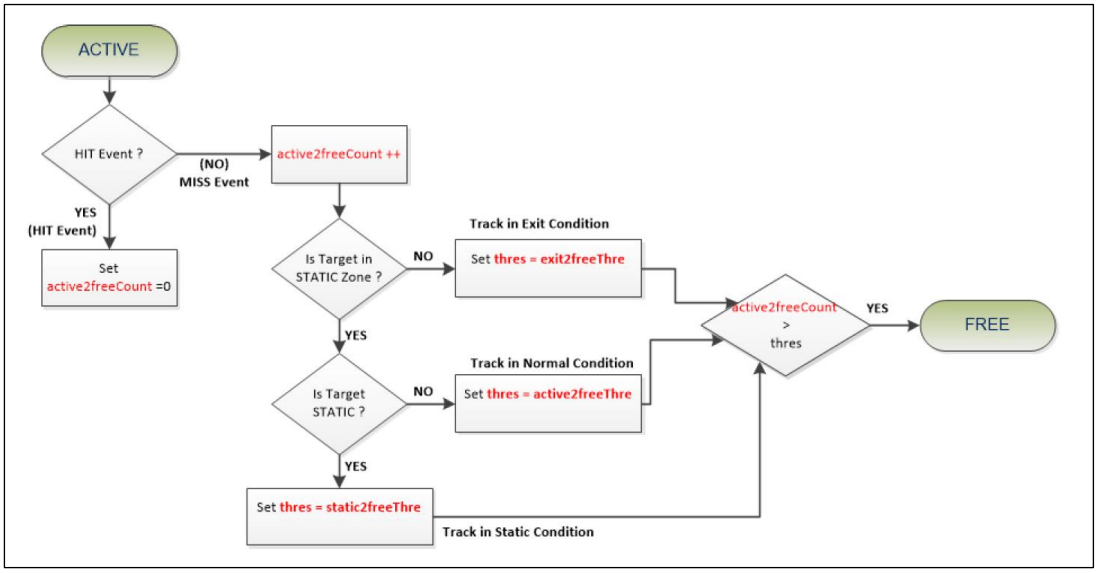


Figure 32. ACTIVE to FREE state transition flow diagram [53]

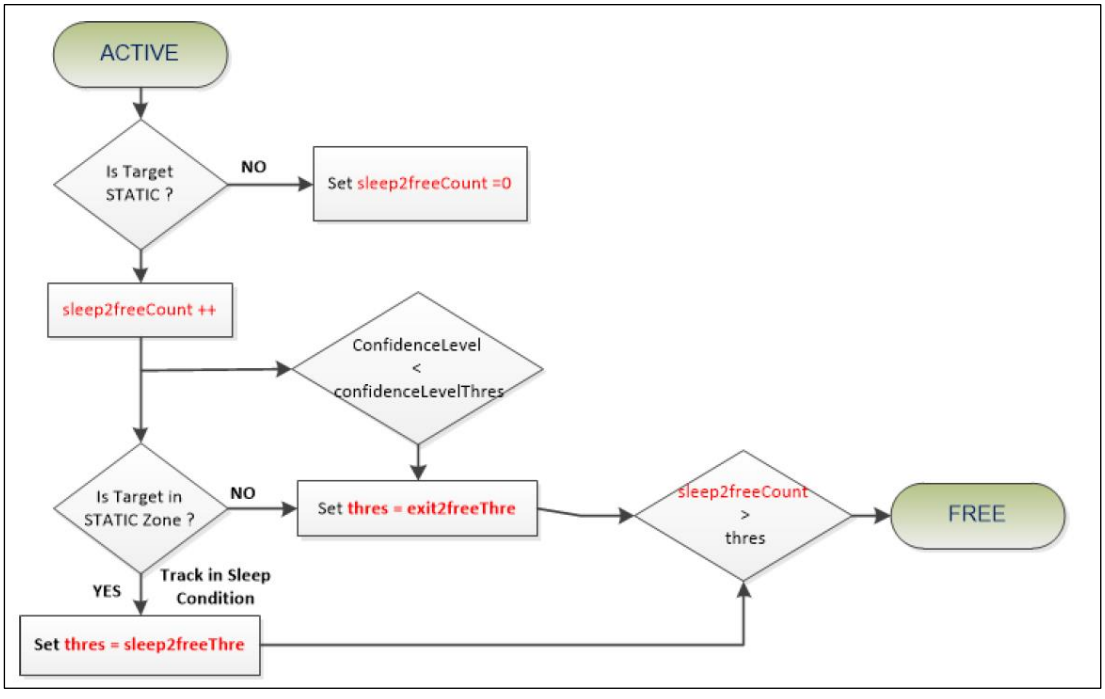


Figure 33. ACTIVE (in Sleep condition) to FREE state transition flow diagram [53]

Selected values for stateParam parameters are given in Table 7 below.

Table 7. stateParam parameters

stateParam	**det2 act Thre**	**det2 free Thre**	**active2 free Thre**	**static2 free Thre**	**exit2 free Thre**	**sleep2 free Thre**
	13	10	10	500	10	500

det2actThre determines the amount of continuous HIT events for transition from DETECT to ACTIVE state. Selected amount for this threshold is 13 as giving lower values resulted in having false detections and tracks since it required less time for a ghost to exist.

det2freeThre determines the amount of continuous MISS events for transition from DETECT to FREE state. 10 was given to this parameter after comparing tests made with higher and lower values.

active2freeThre determines the amount of continuous MISS events for transition from ACTIVE state and normal condition to FREE state. Optimized value is 10 for this parameter.

static2freeThre determines the amount of continuous MISS events for a static target in a static zone for transition from ACTIVE state to FREE state. Larger values for this parameter lead to keeping the track active for a longer time in the static boundary box. Considering how fast was the traffic in the measurement area, parameter is set to 500.

exit2freeThre determines the amount of continuous MISS events for a target outside the static zone for transition from ACTIVE state to FREE state. 10 was given as an optimal value for this parameter since small exit2free threshold values make ghosts to be tracked only for very short times. This also helped the tracker to delete the tracks which exit static boundary box. Smaller values make the proper operation of tracker more difficult according to the test results.

sleep2freeThre determines the amount of maximum possible time a target can be static. There is a specific counter that gets reset when a dynamic point is associated. Selected value is 500 for this parameter.

4.3.5 Maximum Acceleration parameters

Maximum acceleration parameters define the maximum possible value that the acceleration can be changed in X (lateral), Y (longitudinal) and Z (vertical) directions in m/s^2 [51]. *maxAcceleration* CLI command is given below.

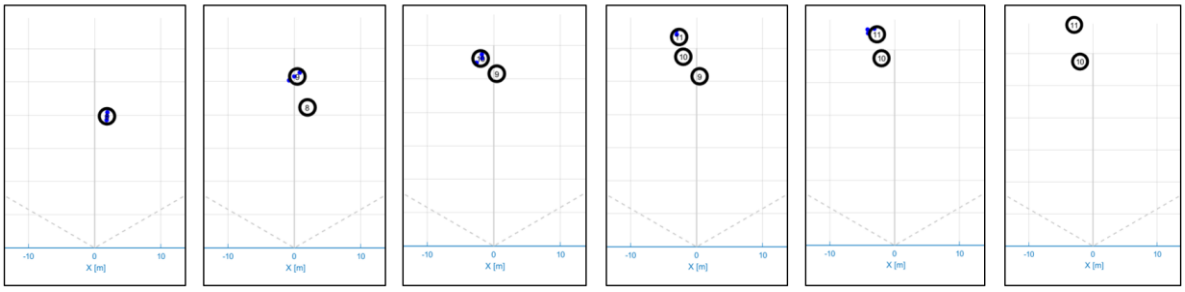
maxAcceleration max X acc., max Y acc., max Z acc

Selected values for maxAcceleration parameters are given in Table 8 below.

Table 8. maxAcceleration parameters

maxAcceleration	**max X acc.**	**max Y acc.**	**max Z acc.**
	50	50	0.1

In some measurements, it was observed that, when vehicles make sudden changes their velocity or directions, deviation from the assumed acceleration motion model happens. This means, track allocated for that object, can be lost or gone into fragmentation. In a situation described in Figure 34, as a result of wrong acceleration limits (maxAcceleration 0,1 0,1 0,1) and also improper mounting of the device, vehicle reaching to the curved part of the road was tracked uneasily by the tracker and it split a single track into multiple tracks.



1 2 3 4 5 6

Figure 34. Fragmentation issue due to low acceleration limits

This kind of situations make keeping constant acceleration values impossible. Therefore, maxAcceleration parameter is used to define maximum limits for possible changes. In case the motion model and the prediction step are less reliable, larger values for maxAcceleration parameter are set. Limits can be set to lower values if less amount of deviation is observed [51]. Measurements were done with higher and lower values for acceleration limits and also without giving this command at all, and the final decision was made on keeping the values as shown in Table 7 above to get better accuracy in tracking performance.

4.3.6 Tracker Configuration parameters

Tracker configuration parameters define activation and deactivation of the group tracker and are used to set and configure the maximum amount of points, tracks, radial velocity related features, and frame rate [51]. *trackingCfg* CLI command is given below.

trackingCfg enable, paramSet, maxNumPoints, maxNumTracks, maxRadialVelocity, radialVelocityResolution, deltaT

Selected values for trackingCfg parameters are given in Table 9 below.

Table 9. trackingCfg parameters

trackingCfg	**enable**	**ParamSet**	**Max Num Points**	**Max Num Tracks**	**Max Radial Velocity**	**Radial Velocity Resolution**	**deltaT**
	1	2	200	16	230	370	66,667

Enable is used to enable and disable the group tracker, therefore can take 2 values: 1 or 0. This value is set to 1 for all configurations.

paramSet is used to define hardcoded configurations. Platform provides 5 configurations [0,1,…4] in total. Measurements taken with stateParam 2 had better results.

maxNumPoints defines the maximum number of detection points to input to tracker per frame. 200 was selected as the most realistic number for this value.

maxNumTracks defines the maximum number of targets (tracks) that can be allocated by tracker per frame. 16 was selected as a maximum number of tracks per frame since it was the number of targets for detection with the highest possibility.

maxRadialVelocity defines the maximum absolute velocity that is reported by the sensor from the detection layer. Estimated by multiplying the maximum radial velocity value determined in sensor chirp configuration with 10 (23,05 x 10 = 230 m/s).

radialVeocityResolution defines the radial velocity resolution reported by the sensor in millimeter/sec. Estimated by multiplying the radial velocity resolution value determined in sensor chirp configuration with 1000 (0,37 x 1000 = 370 mm/s).

deltaT defines the frame rate in millisecond. It is determined to be 66,667 msec in the sensor chirp configuration.

5. EXPERIMENTS

5.1 Data collection process

Data collection process starts with setting up hardware and software platforms for the intended measurement requirements and includes multiple steps till getting visual and radar ready for the analysis. First step is flashing the EVM with the proper binary image from TI Industrial Toolbox by using the UniFlash tool [55]. Traffic monitoring lab from the mmWave Industrial Toolbox 4.7.0 was applied as a development platform within the data collection process. Hardware is considered ready for making the experiment after uploading the developed configuration file to the radar and defining test specific parameters such as duration of each measurement, the number and the names of configurations to run in a single attempt. This is done by running Python startup script on Jetson NANO Ubuntu 18.04.5 LTS with ROS Melodic. The next step of the process is mounting the hardware on a selected measurement spot. The mounting height, horizontal and vertical angles must be measured and estimated beforehand and specified in the sensorPosition command of the configuration file. After mounting the device and connecting it with the battery, a relevant button can be pressed for starting the measurement process. More detailed description of the test environment is given in the next subchapter. Device runs the camera for video recording and the radar module for detection and tracking at the same time and both processes continue for a time interval specified in the script. The output of the measurement is the radar and the video data taken in corresponding time period. MATLAB based GUI was used to visualize the radar data and it requires the input data to be in text format. Therefore, HxD editor was used for the conversion from a binary format in which radar data is taken to the text format. Detailed description of the GUI is given in relevant subchapter. In order to assess the performance, radar and camera data was streamed together and various analysis were done on both outputs for providing results of the work.

5.1.1 Test case description

More than 40 Measurements were taken in 2 different locations with different settings within the work. Selected spots for mounting the hardware were the electric pole next to the Akadeemia tee and Raja roads intersection, facing the road in North-East direction from 2- and 3,2-meters height, and the Nõmme bridge, facing the Ehitajate tee road beneath in North-East and South directions from 8,1 meters height.

Initial measurements were taken in the first spot and results were used to form general understanding of tracking concept and learn basis tracking commands and their functionality. Since the radar mounted in this spot was facing the traffic from the side of the road (not perpendicular to the moving vehicles), it was difficult to test all the parameters efficiently and analyse the quality of the tracking performance especially in terms of the maximum range capabilities of the sensor in tracking context (width of the road is around 10 meters). Therefore, further tests were conducted from the top of bridge where measurements were more suitable for optimizing and evaluating the tracking performance (Figure 35). Measurements were taken from both sides of the bridge (Figure 36).

Figure 35. Bridge where measurements were taken

Figure 36. Mounted hardware on both sides of the bridge

Estimating the correct vertical mounting angle of the device is very important part of the sensor geometry settings and it directly affects the radar performance. Maximum unambiguous range defined for the used radar sensor is 35 meters and mounting height of the sensor is 8,1 meters. By considering these values and natural landscape of the road, 18 degrees was found to be the most optimal value for the vertical tilt.

Tracking performance was evaluated according to results of measurements taken from the road in the South of the bridge, since the road on this side is more straight compared to other side and it made an accurate assessment of all relevant parameters possible. Measurement setup layout and a screenshot taken from the acquired data is described in Figures 37 and 38 respectively.

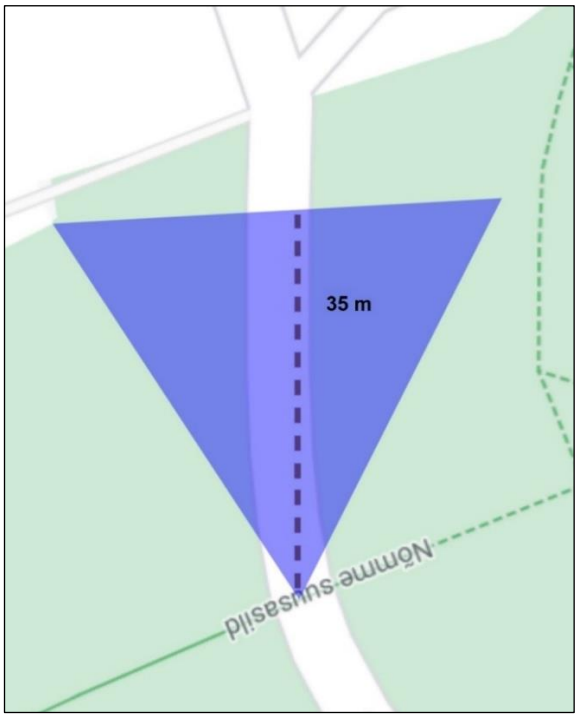


Figure 37. Measurement setup layout

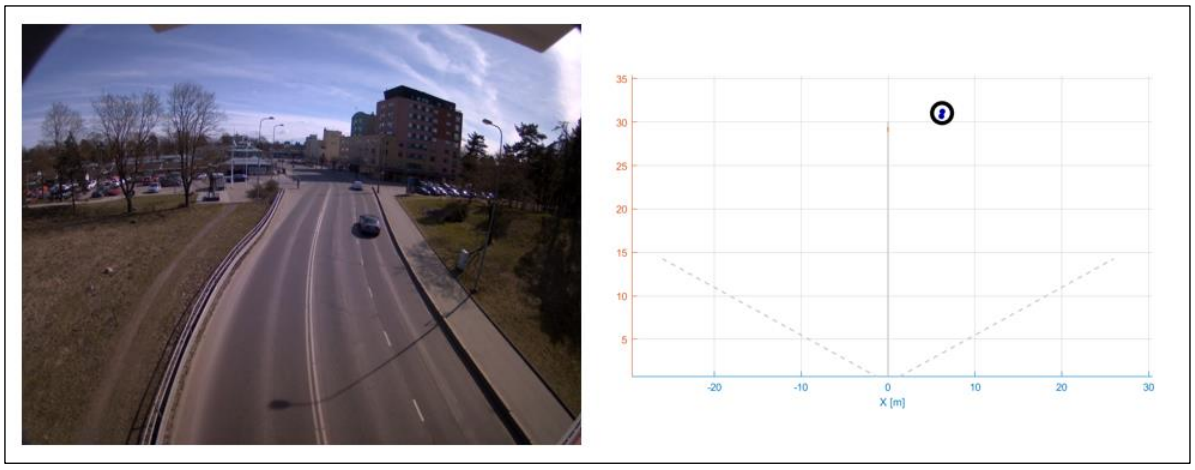


Figure 38. Visual data and corresponding radar output

5.1.2 Graphical User Interface

MATLAB based GUI downloaded from TI Industrial Toolbox was used for visualizing the radar output. After running the relevant script, a setup window appears which allows the user to select the needed visualizer mode, data and configuration files, sensor information and other visualizer options. GUI has 2 visualizer modes: real time mode for parsing the incoming data stream for visualization and play back for uploading recorded data in text format and streaming it via interface. Visualizer interface is described in Figure 39 below.

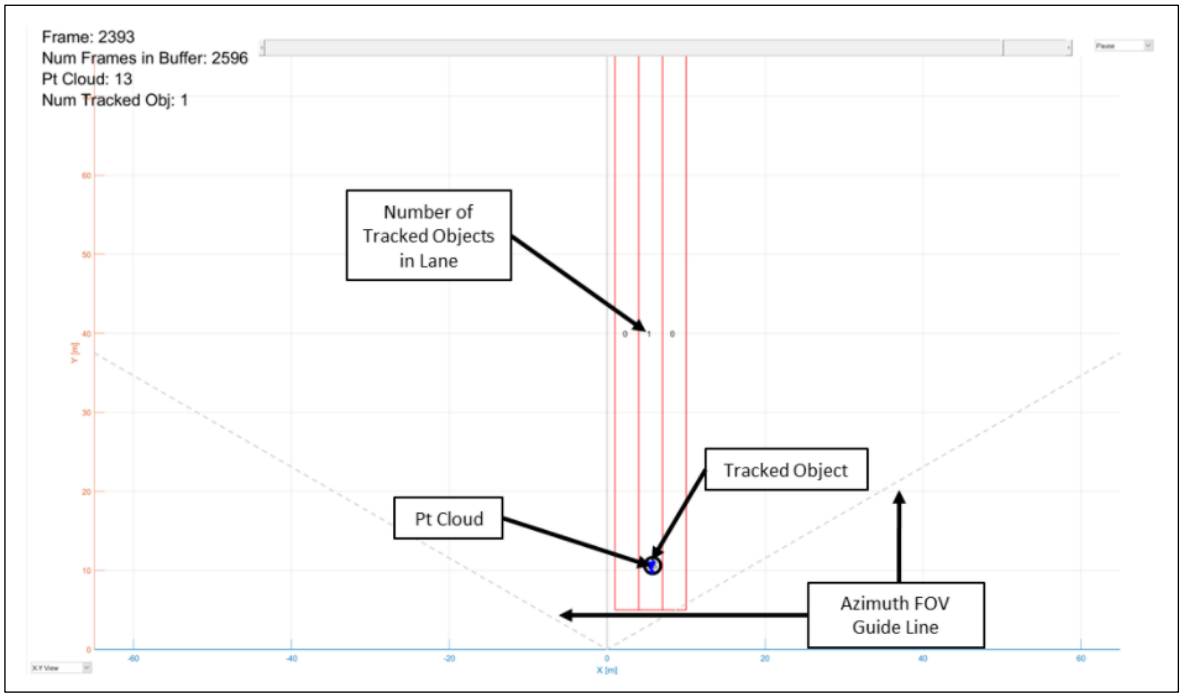


Figure 39. Graphical User Interface [56]

Tracked objects are represented by black circles and each has an ID number. They are formed according to the used configuration file for the particular measurement around point cloud of each detected target.

5.2 Performance

Detection and tracking performance were evaluated manually by streaming visual and radar data simultaneously taken from 4 minutes long measurement. Evaluation process includes determining the number of detected, tracked and perfectly tracked objects separately and estimating the overall tracking performance per each pre-defined vehicle classes.

5.2.1 Detection and tracking reliability

Detection and tracking reliability of the measurement is summarized in Table 10 below.

Lane 1 indicates the left side of the road, where vehicles are coming towards the device, and lane 2 refers to the right side, two traffic lanes of the road where vehicles are moving away from the device.

Table 10. Detection and tracking reliability results

Lane	Number of vehicles	Number of detected vehicles	Detecting reliability, %	Number of tracked vehices	Tracking reliability, %
1	31	31	100	30	96
2	32	31	96	29	93
Total	**63**	**62**	**98**	**59**	**95**

According to the results, it can be concluded that vast majority of vehicles from pre-defined targeted vehicle classes are detected and they are associated with relevant tracks as an output of applied tracking commands.

5.2.2 Tracking precision

Overall tracking precision of the measurement is summarized in Table 11.

Perfect tracking refers to the formation of properly located and continuous (non-broken) tracks around point clouds of detected vehicles in 25 meters. Perfect tracking excludes situations where a single object is reflected with multiple tracks and late removal of tracks belonging to the objects that exit defined measurement ranges.

Table 11. Tracking precision results

Lane	Number of tracked vehicles	Number of perfectly tracked vehicles	Perfect tracking reliability, %
1	30	25	83
2	29	24	82
Total	**59**	**49**	**83**

According to the table, some tracks couldn’t qualify to be considered as a perfect tracking sample. In order to determine this kind of situations, radar data was examined for each vehicle class separately and results are provided in Table 12.

Table 12. Tracking performance evaluation for defined vehicle classes

	Small vehicle	Bus/Truck	Van	Motorcycle/ Bicycle
Number of vehicles on 2 lanes	56	1	4	2
Number of detected vehicles	56	1	4	1
Detection reliability, %	**100**	**100**	**100**	**50**
Number of tracked vehicles	54	1	4	0
Tracking reliability, %	**96**	**100**	**100**	**0**
Number of perfectly tracked vehicles	45	0	4	0
Perfect tracking reliability, %	**83**	**0**	**100**	**0**

5.2.3 Results analysis and limitations

In general, the overall detection and tracking reliability of the developed radar configuration can be considered to be satisfactory. Although it can be confirmed that, almost all targeted vehicle classes were successfully detected and tracked, tracking precision results for some vehicles were not high enough according to Tables 11 and 12.

Regarding detection performance, only 1 of 63 passed vehicles was not detected by the radar. As this vehicle was identified to be a motorcycle, the possible reasons for the situation can be either loss of frame due to inconsistent data sent by the object or play back mode's insufficiency to interpret certain character combinations.

According to the results, 3 of 62 detected vehicles were not tracked at all. These objects were 2 cars (small vehicles) and 1 motorcycle. This has happened when cars couldn't be differentiated from the closest vehicle by the sensor. As the motorcycle was reflected with the point cloud that has less than 5 points, it wasn't associated with a track in the end. The situation is given in Figure 40.

Detailed examination of results taken from all measurements, revealed that perfect tracking capabilities for large vehicles are poor in most cases. The main reason for this issue is a huge amount of dispersion of points in point clouds belonging to large vehicles. This resulted in having multiple tracks for a single target. Despite some attempts for mitigating the number of such cases by optimizing gating parameters described in the previous chapter, the issue couldn't be solved completely and it affected the overall performance. This problem has seen in the tracking of the single bus passed during the measurement. The situation is described in Figure 41.

Another problem that decreased the number of perfect tracking cases is observed broken tracks for some targets. For some objects, when they are getting further away from the radar (mostly after 25 meters distance) and reaching the limits of defined boundary boxes, their tracks are split into two or three. Although measurements were conducted with considering the most suitable sensor geometry parameters, this issue was observed few times during the recording.

Two frequently seen issues also affected the results of final measurement, and perfect tracking cases were observed in only 83% of all tracked vehicles.

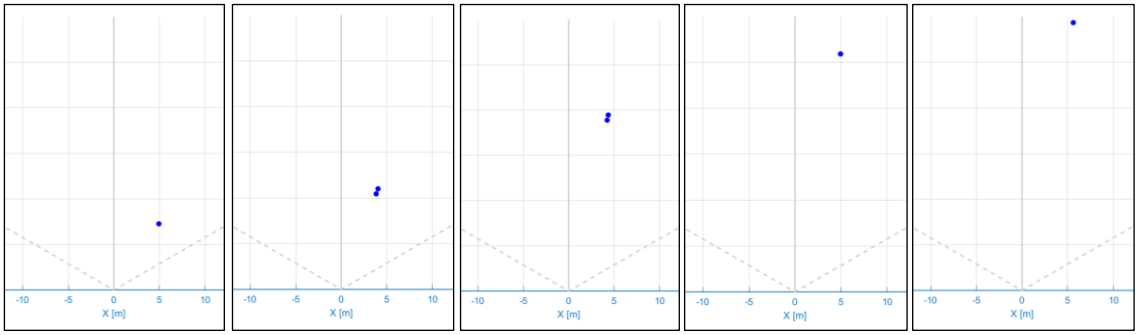


Figure 40. Motorcycle that was detected with less than 5 points in the point cloud

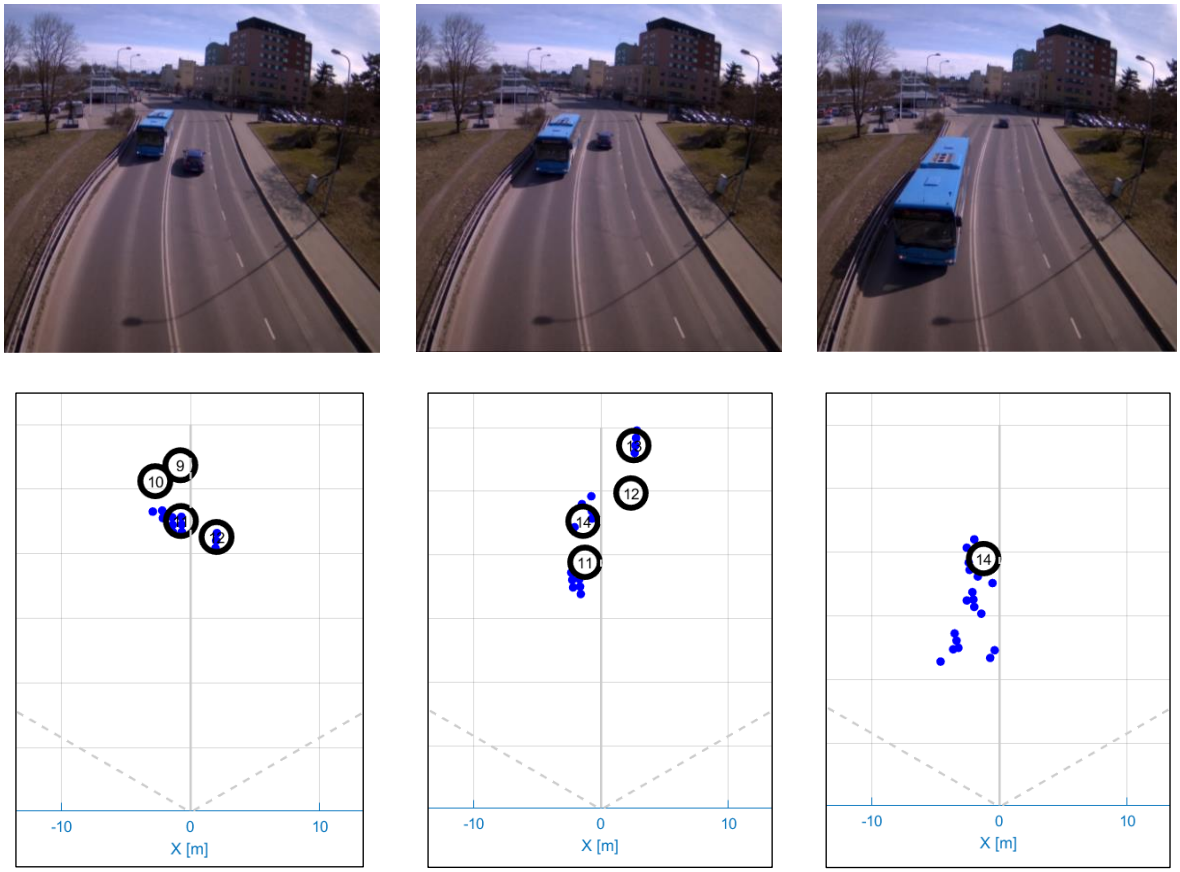


Figure 41. Problems observed during the detection and tracking of a bus