Karlsruher Institut für Technologie

Institut für Technik der Informationsverarbeitung (ITIV)

Institutsleitung
Prof. Dr.-Ing. Dr. h. c. J. Becker (Sprecher)
Prof. Dr.-Ing. E. Sax
Prof. Dr. rer. nat. W. Stork

Masterarbeit Nr. ID-3493

von Herrn cand. el. Faraz **Maqsud**

Distributed Data Source Fusion for Automated Driving Functionalities

Beginn:	13.06.2024
Abgabe:	13.12.2024
Betreuer:	M.Sc. David Kraus Institut für Technik der Informationsverarbeitung M.Sc. Luca Seidel Institut für Technik der Informationsverarbeitung
Korreferent:	Prof. Dr.-Ing. Jürgen Becker Institut für Technik der Informationsverarbeitung

Hauptreferent: Prof. Dr.-Ing. Eric Sax

Ich versichere wahrheitsgemäß, die Arbeit selbstständig verfasst, alle benutzten Quellen und Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde, sowie die Satzung des KIT zur Sicherung guter wissenschaftlicher Praxis in der jeweils gültigen Fassung beachtet zu haben.

Karlsruhe, den 12. Dezember 2024

Faraz Maqsud

Abstract

The advancement of technology has brought automated vehicles to the forefront of the automotive industry, emphasizing their growing significance. These vehicles depend on various sensors to understand their surroundings, including cameras, LiDAR, and Radar. Despite their capabilities, sensors on Onboard units (OBUs) face challenges such as blind spots and occlusions, leading to incomplete environmental perception. This limited awareness poses safety risks to drivers, passengers, and pedestrians. To overcome the challenges, integrating external sensors from Roadside units (RSUs) can enhance the OBU sensor's field of view and provide a more holistic understanding of the environment.

This thesis aims to develop a fusion framework using mid-level (feature-level) fusion to enhance perception and decision-making by combining sensor data from OBU and RSU. The framework addresses the challenge of aligning data from different viewpoints through precise transformations and feature matching, enabling effective fusion for object detection, classification, and tracking.

A key component of this research is the integration of static RSU cameras and dynamic OBU cameras using the ROIF. The ROIF utilizes homography transformations and feature matching to map bounding boxes detected by the OBU camera onto the RSU camera's view, ensuring spatial alignment. These aligned bounding boxes are fused, resulting in a more accurate and continuous representation of the detected objects, even as they move across different viewpoints.

A proof-of-concept prototype validates this approach, demonstrating that the fusion framework significantly enhances object detection accuracy and tracking performance compared to using only OBU camera data. By effectively combining OBU and RSU data, the proposed ROIF improves road perception and offers a reliable solution for automated driving systems.

Zusammenfassung

Der technologische Fortschritt hat automatisierte Fahrzeuge in den Vordergrund der Automobilindustrie gerückt und ihre wachsende Bedeutung unterstrichen. Diese Fahrzeuge sind auf verschiedene Sensoren angewiesen, um ihre Umgebung zu erfassen, darunter Kameras, LiDAR und Radar. Trotz ihrer Fähigkeiten stehen die Sensoren der On-Board-Units (OBUs) vor Herausforderungen wie toten Winkeln und Verdeckungen, was zu einer unvollständigen Wahrnehmung der Umgebung führt. Diese eingeschränkte Wahrnehmung stellt ein Sicherheitsrisiko für Fahrer, Passagiere und Fußgänger dar. Um diese Probleme zu überwinden, kann die Integration externer Sensoren von straßenseitigen Einheiten (RSUs) das Sichtfeld der OBU-Sensoren erweitern und ein ganzheitlicheres Verständnis der Umgebung ermöglichen.

Ziel dieser Arbeit ist es, ein Fusionsverfahren auf mittlerer Ebene (Feature-Level) zu entwickeln, um die Wahrnehmung und Entscheidungsfindung durch die Kombination von Sensordaten von OBU und RSU zu verbessern. Das Framework befasst sich mit der Herausforderung, Daten aus verschiedenen Blickwinkeln durch präzise Transformationen und Feature-Matching abzugleichen, um eine effektive Fusion zur Objekterkennung, -klassifizierung und -verfolgung zu ermöglichen.

Eine Schlüsselkomponente dieser Forschung ist die Integration von statischen RSU-Kameras und dynamischen OBU-Kameras unter Verwendung des ROIF. ROIF nutzt Homographie-Transformationen und Feature-Matching, um von der OBU-Kamera erkannte Bounding Boxes auf die Ansicht der RSU-Kamera abzubilden und so eine räumliche Ausrichtung zu gewährleisten. Diese ausgerichteten Bounding Boxes werden verschmolzen, was zu einer genaueren und kontinuierlichen Darstellung der erkannten Objekte führt, auch wenn sie sich über verschiedene Standpunkte hinweg bewegen.

Ein Proof-of-Concept-Prototyp validiert diesen Ansatz und zeigt, dass der Fusionsrahmen die Genauigkeit der Objekterkennung und die Verfolgungsleistung im Vergleich zur ausschließlichen Verwendung von OBU-Kameradaten deutlich erhöht. Durch die effektive Kombination von OBU- und RSU-Daten verbessert das vorgeschlagene ROIF die Straßenwahrnehmung und bietet eine zuverlässige Lösung für automatisierte Fahrsysteme.

Contents

1. Introduction

The rapid advancement of autonomous driving technologies offers great potential to improve transportation safety, efficiency, and accessibility. Autonomous vehicles (AVs) are expected to reduce human error and decrease traffic congestion by reacting faster to road conditions and optimizing routes for smoother traffic flow [1]. However, existing systems face some challenges in handling different driving scenario, where unpredictability and sudden changes can result in catastrophic failures. This is highlighted by multiple incidents in recent years where automated systems failed to correctly identify hazards in time to prevent accidents. The recent accident back in 2017, where a Tesla Model S crashed into an 18-wheel tractor trailer, leading to the driver's death [2]. Similarly, in 2018, a self-driving Uber vehicle fatally struck a pedestrian, becoming the first recorded instance of an automated car causing a pedestrian's death [3], which led to Uber halting further research [4]. In both accidents, the systems did not detect the obstacles in time to stop, revealing critical shortcomings in autonomous vehicle perception.

Autonomous driving typically relies on a wide range of sensors, each with its own specifications and functionalities, such as cameras, Light Detection and Ranging (LiDAR), ultrasonic sensors, and Radio Detection and Ranging (Radar). These sensors, when used independently, have some limitations and drawbacks. For example, cameras struggle at night and with glare, LiDAR may face difficulties during fog or rain, and Radar can have challenges detecting fine details or distinguishing between different objects. These limitations affect the vehicle's ability to accurately perceive the urban environment, where pedestrians cyclists and other vehicles may unexpectedly cross its path [5][6].

On the other hand, the integration of Vehicle-to-Everything (V2X) communication marks a significant milestone in the advancement of urban mobility and transportation system [7], which allows automated vehicles to receive information from connected infrastructure, vehicles, and cloud-based systems. The advancement in the data rate from LTE (up to 1 Gbit/s) to 5G (up to 20 Gbit/s) [8] and the introduction of ultra-reliable low latency communication (uRLLC), designed for vehicle networking, give modern vehicles the ability to exchange data reliably and quickly. This allows nearby vehicles to share information about potential sources of danger and road conditions, thereby increasing road safety. These improvements are important to potentially reducing fatal road accidents and ensuring safer roads for all.

With the integration of multiple onboard and external sensors generating a substantial amount of redundant information, the challenge lies in how to efficiently process this data while maximizing the unique strengths of each sensor. Effective data fusion techniques

are essential to filter out redundant information, reduce data overload, and ensure that the system operates with high accuracy and reliability.

1.1. Problem statement

Despite improvements in autonomous driving technology, current systems still need help maintaining consistent reliability and safety in complex and dynamic driving environments. A key limitation arises from the dependence on individual sensors like cameras, LiDAR, and Radar, each of which has its limitations, such as sensitivity to adverse weather conditions, lighting variations, etc. These limitations can result in inaccurate object detection, classification, and tracking, ultimately affecting the vehicle's ability to make safe decisions [5][6].

The potential of external data sources, such as vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication, needs to be more utilized, limiting the system's situational awareness. These external data provide valuable information that can significantly enhance situational awareness beyond what the vehicle's onboard sensors can perceive. The fusion of OBU sensors with RSU sensors can provide a more accurate understanding of the driving environment, improving the system's ability to respond to challenging scenarios. However, the full potential of external sensor fusion is often not realized, leaving a gap in the robustness of current autonomous driving systems, which could lead to increased risk of accidents and potential legal liabilities [9].

Aligning data from the RSU and OBU sensors is challenging due to the occlusions and partial visibility both sensors often experience. This alignment challenge becomes more pronounced when attempting to fuse data from these two perspectives, as precise alignment is crucial for ensuring that objects are consistently detected and tracked across both views. Even when sufficient data is available, aligning and fusing this data becomes more complicated, as it involves processing information from two distinct sources with differing perspectives, where misalignment could lead to false detections and incorrect fusion.

Furthermore, early fusion techniques, where raw sensor data is combined at the input stage, exacerbate these complexities. It significantly increases the amount of data to be processed. The large volume of raw data from different sources leads to high computational costs, longer processing times, and greater demands on memory storage [10]. This overload not only complicates the fusion process but also reduces the efficiency of the system, making it more difficult to process and respond to real-time driving scenarios. Thus, the need for more efficient sensor fusion techniques that can overcome these challenges is important for improving both the performance, reliability and robustness of automated driving systems.

1.2. Thesis Structure

Chapter 2 presents an overview of the sensor systems used in autonomous driving technology and explains the relevant fusion concepts, including levels of fusion, object detection network and V2X. Chapter 3 discusses related work in the field of sensor fusion and object detection. Chapter 4 defines the formal system requirements and develops the conceptual approaches used to address these requirements. Chapter 5 introduces the tools and methods for implementing these concepts, along with a detailed description of the fundamental software architecture and fusion framework. Chapter 6 evaluates the framework's performance by testing the algorithm in different scenarios and analyzing its effectiveness and capabilities. Finally, Chapter 7 wraps up the thesis by summarizing the findings and offering potential future enhancements to the proposed system.

2. Fundamentals of Sensor Fusion

Sensor fusion plays an important role in the development of automated driving. It refers to the process of integrating data from multiple sensors to provide a more accurate, reliable, and comprehensive understanding of the environment. This enhanced perception improves safety and drives progress in autonomous driving systems. This chapter delves into different levels of automation along with sensor technologies, further more different fusion approaches such as low, mid, high and as well as hybrid fusion are discussed. It also examines the integration of V2X communication systems, which enhance situational awareness by facilitating data exchange between OBU and RSU sensors. The discussion highlights the pivotal role of these technologies in advancing autonomous driving capabilities.

2.1. Overview of Autonomous Driving System and its Sensor Technology

2.1.1. Automation in Driving System

In the EU, more than twenty thousand road accident fatalities are caused every year due to human negligence [11]. Drivers most commonly fail to act when they should. Increased safety is one of the important goal for the adoption of autonomous vehicles and advanced driver assistance systems (ADAS) [12]. In 2010, a project group introduced a preliminary framework for classifying different level of vehicle automation and driver assistance systems [13]. In parallel, the National Highway Traffic Safety Administration (NHTSA) in the United States explored similar classification related to vehicle automation [14]. In 2014, the Society of Automotive Engineers (SAE) published the SAE J3016 standard [15], which officially outlines the levels of automation [16], it covers a spectrum of automation levels, from basic driver assistance to fully autonomous driving. At lower levels, features like adaptive cruise control, lane-keeping assistance, and automatic emergency braking supports the driving in different driving scenarios. As the levels of automation increase, vehicles begin to make more decisions independently, such as navigating, avoiding obstacles, and interpreting traffic signals. The goal of driving automation is to achieve fully autonomous vehicles that can operate safely and efficiently without involving any human control. The levels of driving automation outlined by SAE J3016B [16] can be defined as follows:

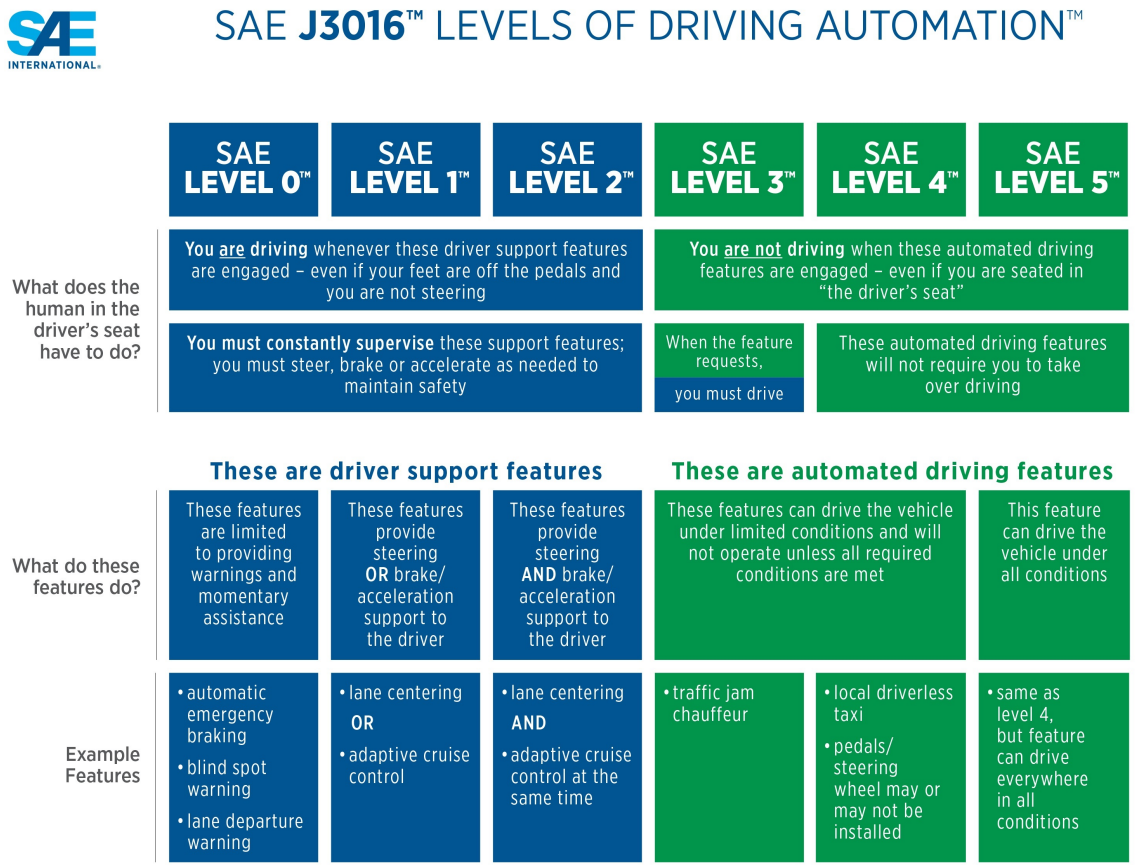

Figure 2.1.: SAE J3016 levels of Driving automation [16]

.

Level 0 - No Automation:

The driver has complete control over the vehicle, including steering, acceleration, braking, and decision-making. While the vehicle may be equipped with certain warning features, these systems only provide alerts and do not take control of the vehicle. All decisions remain the driver's responsibility. Examples of such features include Lane Departure Warning (LDW), Blind Spot Detection (BSD), and Forward Collision Warning (FCW) [17].

Level 1 - Driver Assistance:

The vehicle can assist with one feature at a time, such as steering or acceleration/braking. The driver must remain in control and stay alert to monitor the environment. Such features include Adaptive Cruise Control (ACC) and Lane Keeping Assistance (LKA) [17].

Level 2 - Partial Automation:

This Level is characterized as "hands off" automation, allowing the vehicle to manage both steering and acceleration/braking simultaneously [17]. However, the driver must remain seated in the driver's position and be prepared to take control when needed. Features like Tesla's Autopilot fall under this category [17].

Level 3 - Conditional Automation:

This level is described as "eyes off" automation, in which the system is capable of handling all driving capabilities and also make decision based on driving scenario for themselves such driving around a stopping vehicle or driving in congested areas. However, the driver needs to stay in the driver's seat and be ready to take over the wheel when the car asks for it. The example of it is Audi's Traffic Jam Pilot system [17].

Level 4 - High Automation:

This level is also known as "mind off" automation, allowing vehicle to take fully control of the vehicle. Systems can perform all driving tasks autonomously in defined conditions or geofenced areas where the driver's attention is not needed, and they can engage in other activities. However, outside of these areas, the driver may still need to take control. An example of Level 4 Waymo's self-driving taxis in select urban areas [17].

Level 5 - Full Automation:

This level represents the highest level of the autonomous driving spectrum, where the system fully controls the vehicle in all driving scenarios and road conditions. There is no need for a steering wheel, pedals, or any driving control, as the vehicle is fully capable of always driving itself. All persons in the vehicle are passengers. This represents the goal of autonomous vehicle development [17].

2.1.2. Sensor Technology

The most important thing in autonomous driving in environment perception [18], which would not be possible without using sensor technology. This sensor technology enable vehicles to perceive and understand their environment, different types of sensors are installed in the vehicle to gather data about the surrounding helping vehicle to detect obstacles, identify lanes, recognize traffic signs, and predict the behaviour of pedestrians and other vehicles. Primary sensors installed in vehicle includes camera, Radar, LiDAR and ultrasonic sensor.

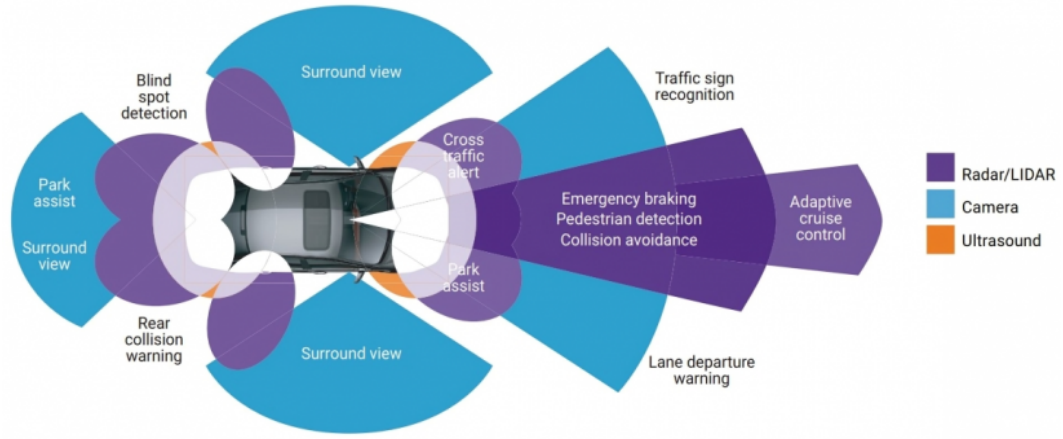

Figure 2.2.: Sensors used by the ADAS [10]

2.1.2.1. Camera

The camera play an important role in visualizing the environment. They have become a vital component in a vehicles due to their ability to capture visual information with detailed information about the environment . It is one of the most reliable sensor as compared to other sensors as it visually differentiate between different objects, with a key advantage of being able to detect color and textures [10]. This capability allows autonomous vehicles to recognize features like road signs and traffic lights. Cameras work by capturing passive light to produce a digital image of their surroundings [19]. They can identify both stationary and moving objects within their field of view and gather various types of data, such as spatial details, shape, size, dynamic information (tracking motion across consecutive frames).

Camera sensors are affordable and widely accessible, however they face some limitations that can affect their performance, such as unfavorable weather conditions and the computational demands of processing the data they capture[19]. For instance, heavy rain or fog can reduce visibility and affect the environment perception. Additionally, the processing of high-resolution images and video such as 1080p which are installed in automated vehicle for advanced object detection [20] can demand high performance hardware, as standard hardware can lead to longer processing times. As technology advances these limitations will continue to improve.

2.1.2.2. LiDAR

LiDAR, which stands for Light Detection and Ranging, is used to measure the distance to objects and generate a detailed, three-dimensional point cloud of the surroundings. Unlike other sensors that offer a limited field of view, LiDAR provides a 360° map around the vehicle [21], providing comprehensive coverage of its environment. It works

by emitting a pulse of light mostly in the form of laser and then measuring the pulse that returns after reflecting off an object. It calculates the time duration between the emitted pulse and the reflected pulse, allowing it to determine the distance to the object [19]. This time-of-flight measurement generates 3D point cloud data, which forms a detailed 3D representation of the environment. Each point in the point cloud represent a location where a laser pulse reflected off an object.

LiDAR plays a crucial role in the automotive industry, where it works alongside other sensors to provide detailed information about a vehicle's surroundings. Its precision and ability to generate high-resolution data make it a preferred choice for autonomous vehicle manufacturers, including companies like Google, Uber, and Toyota [21].Although LiDAR offers significant benefits, it also has limitations that prevent it from being used as a standalone solution. For instance, its performance can be compromised in different weather conditions, such as fog and rain, which affects its ability to accurately detect objects and surroundings. Furthermore its relatively more expensive as compared to other sensor which limits its practicality in some applications [5].

2.1.2.3. Radar

Radar technology uses electromagnetic (radio) waves to measure the distance, position and velocity of the surrounding objects [19]. It transmits radio waves that reflect off objects, returning as echoes. Radar analyze these echoes and determine the distance to the object, its angle relative to the transmitter, and its movement. Due to their robustness, as they operate using radio waves which are not much effected by diverse weather conditions, Radar sensors are highly reliable and can be integrated with other sensors for accurate environmental perception. They are available in different variants, such as short-range and long-range to perform a range of functions. Short-range Radar sensors are typically used for tasks like blind-spot monitoring, lane-keeping assistance, and parking aids, while long-range Radar sensors are essential for adaptive cruise control and emergency braking systems [21].

Radar systems have certain limitations that can affect their performance in specific scenarios. One key limitation is their restricted depth perception, which can lead to false identification of objects. This occurs when Radar misinterprets environmental factors like birds or surface reflections as actual objects. Additionally, Radar struggles to detect small or thin objects because they may not produce enough of a signal to be recognized, potentially causing important objects to be overlooked [6].

2.1.2.4. Ultrasonic sensor

Ultrasonic sensors are widely used for object detection and distance measurement in the automotive application. They transmit ultrasonic waves, which reflect off nearby objects and return to the sensor. By measuring the time it takes for the waves to travel

to the object and back, the sensor can determine the distance [19]. These sensors are capable of measuring short-range distances, typically from centimeters to few meters. Instead of light wave these sensors depend on ultrasonic waves which make them highly reliable in various weather conditions, making them ideal for applications such as parking assistance and collision avoidance. However, these sensors can be significantly affected by disturbances in sound waves as they rely on the propagation of sound to detect objects. Sound waves travel through a medium, and any changes in the environment such as humidity, temperature or other can effect its capabilities [10][19].

2.2. Types of Fusion

Sensor fusion in autonomous driving involves integrating data from multiple sensors to improve the overall perception and decision-making capabilities of the vehicle. Different types of fusion techniques are used to combine sensor inputs at different levels based on the processing stage at which the fusion occurs, allowing for more accurate and reliable environmental understanding. These levels of fusion are generally categorized as low, mid, and high. Low-level fusion focuses on fusing raw data to create a single representation before any significant processing [20]. In mid-level fusion, features are extracted from the raw data and then fusion process take place [22], whereas in high level fusion each sensor individually process their data and then fuse the final decision [23]. The quality of decision-making regarding the system's state can be enhanced by performing mid-level and high-level fusions concurrently to improve the final decision [24].

2.2.1. Low Level Fusion

Low-level fusion, also known as early fusion, involves combining raw data from individual sensors before any processing takes place [20]. Each sensor generates data that contains information about the environment but may be incomplete or noisy when considered independently. By fusing the raw data from all sensors, the system takes advantage of the strengths of each sensor, creating a more detailed and comprehensive representation of the environment. After this fusion process, the combined dataset is then processed for further analysis, such as object detection, classification, or decision-making.

Although low-level fusion offers the advantage of providing comprehensive detail, leading to more accurate object detection and localization, it also has some drawbacks. One of the main disadvantages is that it requires significant computational power to process the large amounts of raw data generated by multiple sensors [20]. This can result in increased processing times and may require more advanced hardware. Furthermore, because the raw data is typically large, it demands more storage space, which can be a challenge in systems with limited memory capacity [25].

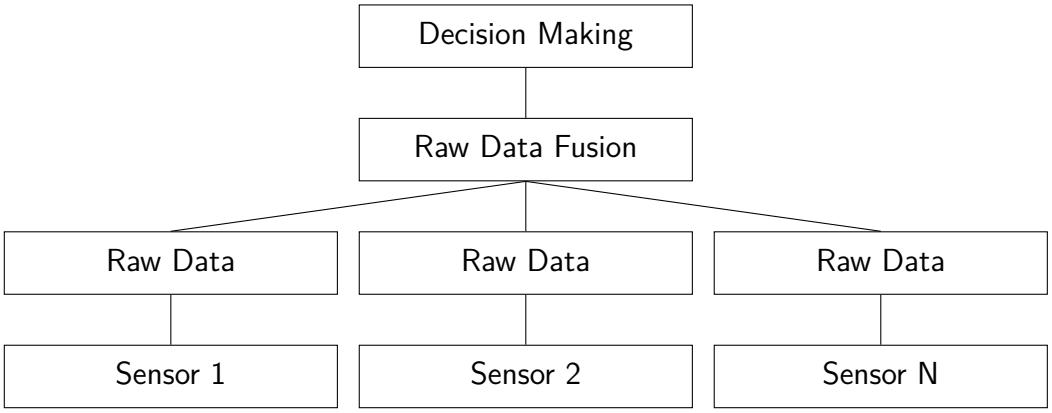

Figure 2.3.: Low Level Fusion [25]

2.2.2. Mid Level Fusion

Mid-level fusion also known as feature level fusion in which key features (such as edges, shape, texture or object attributes) are extracted from the raw data [22]. As illustrated in figure 2.4 each individual sensor (sensor 1, sensor 2,..., sensor N) collect raw data from their respective environments at the most basic level. Depending on their features and intended uses, these sensors may gather various kinds of data. After that, each sensor's raw data is run via a feature extraction procedure. The aim of feature extraction is to simplify and improve the interpretability of the raw data by locating and separating the most pertinent traits or patterns. To ensure only important and meaningful data is retained, each sensor goes through a separate feature extraction procedure. The extracted features from each sensor are then integrated in the feature fusion stage. Feature fusion combines the different features of multiple sensors into a single representation [22]. This stage is essential because it takes advantage of the complimentary nature of data from several sensors, thereby increasing the system's robustness and accuracy. Following feature fusion, the resulting information serves as input for the decision making stage, this final stage employs advanced algorithms, such as machine learning or statistical approaches, to assess the combined characteristics and draw educated conclusions or predictions.

2.2.3. High Level Fusion

High-level fusion, also known as decision-level fusion, occurs after individual sensors have processed their data independently and made decisions or classifications [23]. In this approach, each sensor first performs its own analysis and makes a decision about the environment, such as object detection, classification, or distance measurement. These individual decisions are then fused to make a final decision about the environment. By relying on the processed decision rather than raw data or features, high-level fusion allows for easier integration of multiple sensor types and can be more computationally efficient.

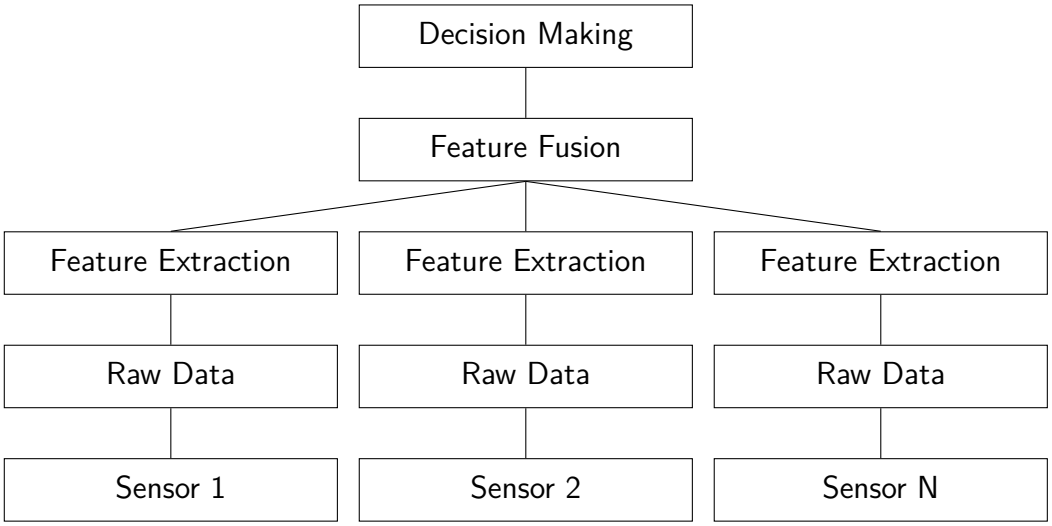

Figure 2.4.: Mid-Level Fusion [25]

This approach is especially useful in systems where sensors operate independently or when combining the output of different algorithms. Since each sensor performs its own analysis, the system can easily combine the output decision data, making it more flexible for integrating additional sensors into the existing system. This approach also requires less processing time, as the system only needs to fuse the final decisions rather than processing raw data or features from each sensor. Although high-level fusion approaches are often preferred because they have lower complexity compared to low-level fusion and mid-level fusion methods but they can result in insufficient information, as it discards classifications with lower confidence values, especially in situations with overlapping obstacles [26].

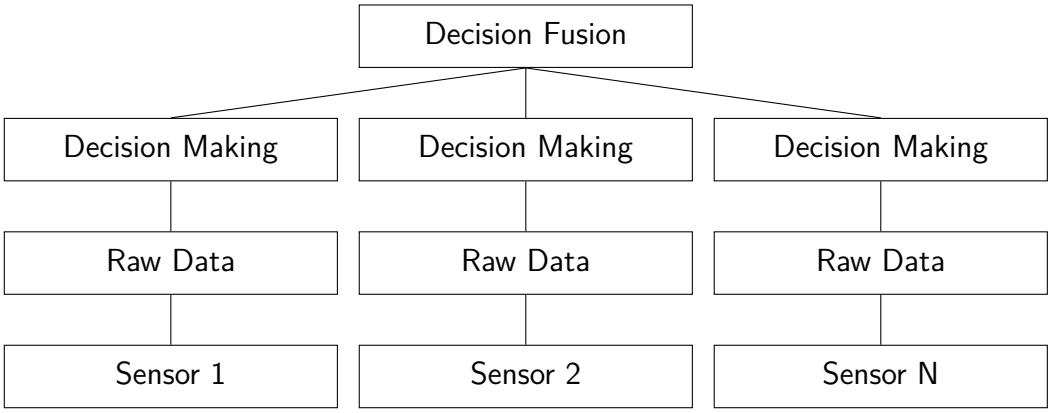

Figure 2.5.: High Level Fusion [25]

2.2.4. Hybrid Fusion

Hybrid fusion, also known as multi-level fusion, involves combining two or more levels of fusion, such as low-level, mid-level, or high-level fusion, to enhance system performance as a specific type of fusion may have some advantage or disadvantage over other. This allows for a flexible fusion strategy that integrates raw data, extracted features, and decision-level information based on the system's specific requirements [27]. Hybrid fusion architectures seem to have the advantages of integrating multiple level of fusion architectures. However, their practical implementation can become quite complex. This complexity arises when fusion is required across various sensor types, which may be spread across different communication channels and hardware systems [28]. Moreover, ensuring synchronization between different sensors and maintaining data transmission latencies can add further difficulties [28]. Therefore, the design and implementation of hybrid fusion systems requires proper planning to ensure the system operates efficiently without being overly complicated.

2.2.5. Comparison

Each sensor fusion level has its own advantages and disadvantages, depending on the conditions and system requirements. Low-level fusion deals with raw data, offering precision and detailed information but requiring significant higher processing time and computational synchronization. Mid-level fusion balances data processing and complexity by extracting and combining features, improving system efficiency, though it may lose some raw data detail. High-level fusion operates on decisions, depending on each individual sensor outputs. Table 2.1 provides a comparison between these fusion levels low, mid, high and hybrid as explained in the previous sections, highlighting their respective advantages and disadvantages.

Table 2.1.: Fusion Techniques: Advantages and Disadvantage

Fusion	Advantage	Disadvantage
Low Level	- Provide detailed information. - Support complex learning algorithms. - Preserve raw information.	- Computationally expensive. - Challenging to synchronize sensors operating at different rates. - Sensitive to noise, can impact fusion results.
Mid Level	- Reduce data volume by extracting features. - Easier to integrate heterogeneous sensor. - Computationally efficient.	- Possibility of missing some raw information. - Requires well alignment of features before fusion. - Errors if extracted features are inconsistent.
High level	- Easily integrates decisions from multiple independent sensors. - Reduces system complexity - Computational simplest.	- Relies on independent sensors capabilities. - Difficult to trace origin of errors - Limited adaptability to new conditions.
Multi Level	- Flexible in terms of fusing raw data, feature or decisions. - More robust by integrating data at different abstraction levels. - Can accommodate a wide range of sensor types.	- Advance system architecture to handle multiple fusion levels. - Can introduce delay in the system. - Demands advanced hardware and software capabilities.

2.3. Object Detection Network

In automated driving, object detection is one of the most crucial tasks for identifying vehicles, pedestrians, cyclists, and obstacles in the environment. By accurately detecting these elements, the systems gain a comprehensive view of the surrounding scene and make informed decisions, such as adjusting speed, changing lanes, or stopping when required. Effective object detection not only enables the system to respond to changes in the environment but it also improves the overall safety and comfort of passengers, ensuring smoother navigation and reducing the risk of accidents. Among the numerous detection algorithms available, YOLOv3 has been widely recognized for its impressive balance of speed and accuracy of detection, making it particularly suitable for automated driving application.

2.3.1. YOLOv3

The YOLOv3 is a real-time, object detection network that uses a feature extraction and multiple detection heads to accurately identify objects in both images and videos [29]. The algorithm's name "YOLO" (You Only Look Once) represents the key foundation behind its design, it processes an image just once in order to generate predictions about the objects within it. Unlike earlier object identification models which required numerous passes over an image using techniques like Region Proposal Networks (RPNs) to search for objects in different parts of the image, YOLO's approach involves analyzing the entire image in a single forward pass of the neural network, making use of efficient 1×1 convolutions that ensure the prediction map size aligns with that of the feature map [30]. This efficient process improves both speed and accuracy, allowing the fully connected layer to take advantage of a compact feature representation before generating detection predictions, which ultimately minimizes the chance of missing the object in a scene.

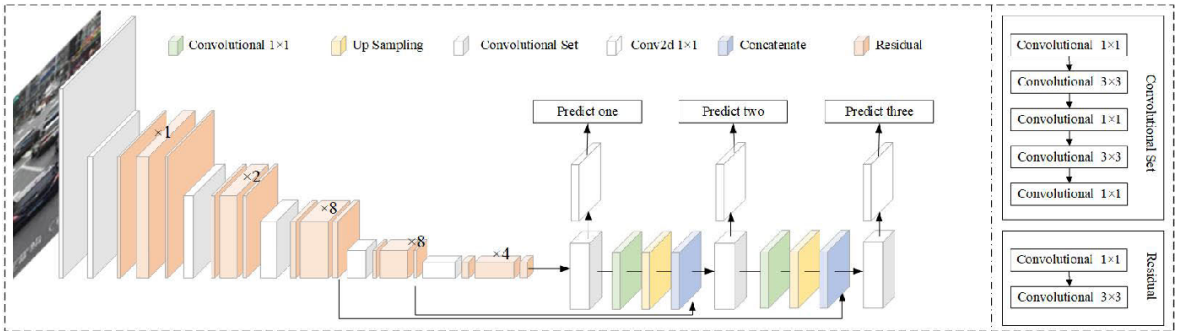

Figure 2.6.: Structure of YOLOv3 [30]

2.3.2. How YOLOv3 works

The YOLOv3 object detection model utilizes a deep learning Convolutional Neural Network (CNN) to analyze input images and generate predictions based on various feature maps. To standardize the data fed into the neural network, the input image is resized to a fixed dimension, typically either 416x416 or 608x608 pixels [31]. A key component of YOLOv3 known as Darknet-53, which is integral to the feature extraction process. This architecture consists of multiple convolutional layers which allows it to learn different complex patterns and representations directly from raw image data and systematically extract the features from the raw image, utilizing residual connections to facilitate gradient flow and improve the efficiency of the learning process [29]. Furthermore YOLOv3 has ability to perform multi-scale detection, utilizing three detection heads that function at different scales. Each head is designed to detect objects of different sizes. The smallest objects are identified using the most detailed feature map from the deeper layers, medium-sized objects are recognized through a mid-level feature map, and larger objects are detected with the coarsest feature map derived from the shallower

layers. This multi-scale technique enables YOLOv3 to identify objects of all sizes inside an image. YOLOv3 employs predefined anchor boxes, which are rectangles of different shapes and sizes, to facilitate object detection in an image. For every anchor box, the algorithm predicts three primary attributes [29]:

- Intersection over union (IoU)
- Anchor box offsets
- Class probability

2.3.3. IoU and Anchor Box Offsets

The IoU score, the first attribute, measures the overlapping between the predicted bounding box and the real bounding box, indicating how closely the prediction matches an actual object in the image. A higher IoU score (as it ranges from [0, 1]) suggests a more precise alignment, helping the algorithm identify boxes that likely contain objects. Bounding boxes with an IoU score above a specified threshold are considered to represent the same object. This threshold ensures that only highly overlapping boxes, which are likely to correspond to the same object, are fused, thus preventing false positives from merging. The IoU can be calculated as the area of overlap of the bounding boxes over the area of union.

$$\text{IoU} = \frac{\text{Area of Intersection}}{\text{Area of Union}}$$

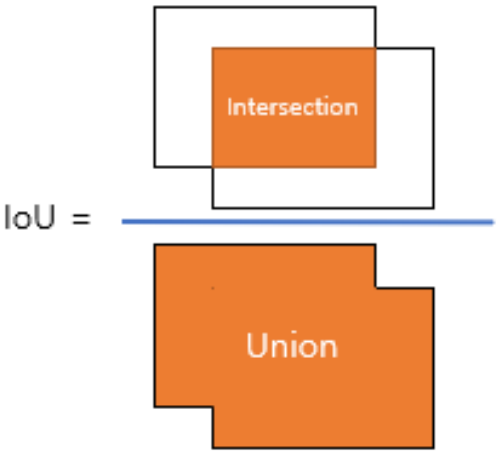

Figure 2.7.: Intersection over Union [32]

The second feature involves the anchor box offsets, which modify the box's position to align more precisely with the object's actual location within the grid cell. These

offsets derive from the anchor's original position and enhance the placement for improved detection accuracy. Finally, the class probability assigns a label to each anchor box, indicating which class the identified object probably falls into. This probability is crucial for the model to classify objects, such as differentiating between a car and a pedestrian, and is vital for recognizing multiple classes in a single image.

2.3.4. Class confidence and Box confidence score

Every bounding box has an center (x,y), width (w), height (h) and box confidence score value This confidence score tells us two things, how likely it is that an object (like a car, person, or animal) is in the box, and how well the box fits the object. YOLOv3 breaks the image into a grid, and each square in the grid makes its own predictions about objects it might contain. Each prediction includes the bounding box coordinates (x, y, w, h), normalized to fit within the grid cell, and 20 probabilities that represent different object classes [30]. This means each grid cell can guess what type of object it's looking at (like a dog or car) and how certain it is. To get a final confidence score for each box, YOLOv3 multiplies the confidence score of the box itself by the probability of the object class it thinks it contains. This way, each box has a score showing the likelihood it contains a specific object, making it easier to choose the best predictions. When all predictions are made, only the boxes with high confidence scores (usually above 0.25) are kept as final results. YOLOv3 uses a technique called Non-Maximum Suppression (NMS) to keep just one bounding box for each detected object, removing overlapping boxes to avoid duplicates [33][34]. Choosing the right confidence threshold can be tricky, too high and chances of missing small or less clear objects, too low and it could detect too many irrelevant things. YOLOv3 handles this by using multiple confidence levels for each object in the final output, so we don't have to adjust the confidence level manually. This approach helps balance the detection accuracy and keeps the process efficient.

2.3.5. Metrics for Object detection

Object detection is evaluated using metrics such as accuracy and precision. Accuracy measures how correctly objects are detected; it is the fraction of successfully detected objects (true positives) among the total number of objects in the dataset, which includes both true positives and false negatives [35]. On the other hand, precision measures how accurately the detected items are identified. It focuses on detection quality by determining how many of the detected objects are correct [35]. Both accuracy and precision are calculated using specific formulas.

$$\text{Accuracy} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.1}$$

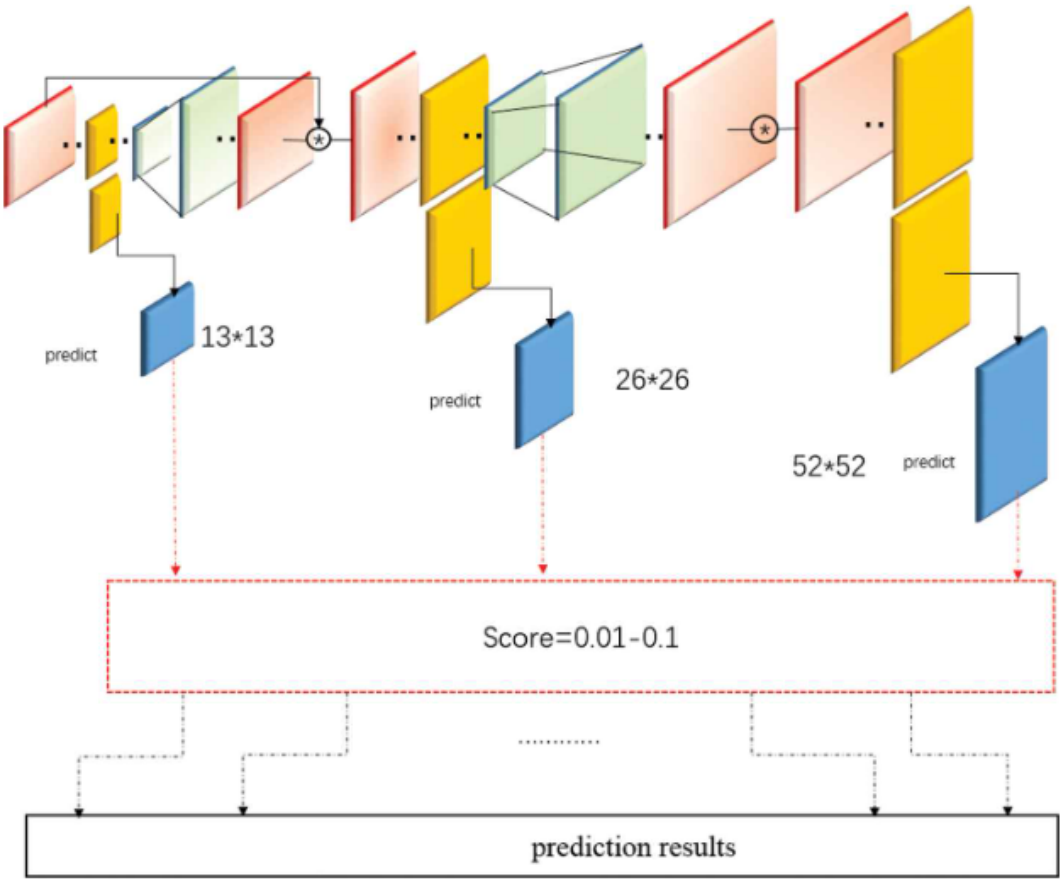

Figure 2.8.: YOLOv3 Predictions Above 0.1 Yield Large Outputs and Low Accuracy [33]

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.2}$$

Here:

- **True Positives (TP)** : Objects correctly identified.
- **False Positives (FP)** : Non-existent objects that are incorrectly identifies.
- **False Negatives (FN)** : Real objects that are failed to detect.

Accuracy and precision are two important ways to evaluate how well object detection models work, but they serve different purposes. Accuracy gives a general idea of how effective the model is, but it might not be beneficial when the dataset is unbalanced. Precision, however, is essential in critical areas like automated driving, where avoiding false positives is essential for safety and reliability [35].

2.4. Homography Transformation

Homography transformation is a mathematical concept used to relate two images of the same environment taken from different perspectives or angles. It defines how points from one image can be mapped onto corresponding points in the other through a projective transformation [36]. The first step involves selecting corresponding points in each image; these points should lie on the same plane in the scene and be identifiable in both images. The next step involves obtaining the intrinsic and extrinsic parameters of each camera. The intrinsic parameters include properties such as the focal length and optical center of the camera, while the extrinsic parameters describe the camera's position and orientation in the scene, capturing its perspective and angle. Once these parameters are known, they are used to compute the homography matrix, which establishes the relationship between the two images, enabling accurate alignment and mapping between them.

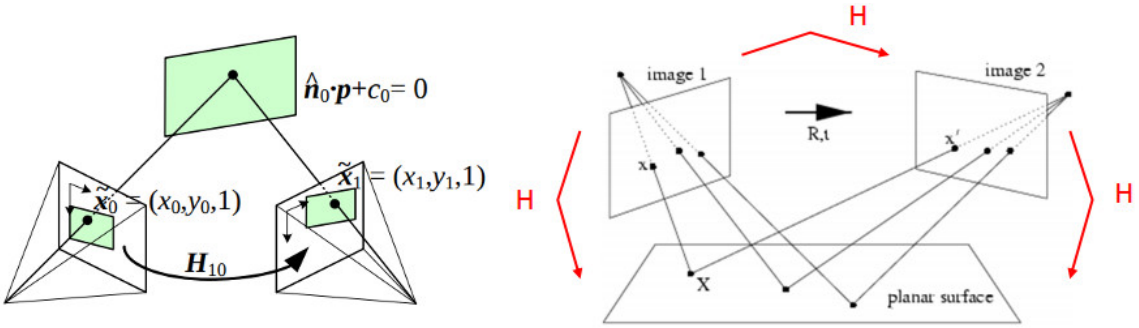

Figure 2.9.: A Planar Surface viewed by Two Cameras Positions [36]

For any point (x, y) in the first image, the homography transformation maps it into the corresponding point (x', y') in the second image using a 3×3 homography matrix H:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.3}$$

Expanding this matrix equation:

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.4}$$

where h_{ij} are the elements of the homography matrix H, and w' is a scale factor in homogeneous coordinates.

To retrieve the coordinates in Cartesian form, we divide by w':

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}} \tag{2.5}$$

$$y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}} \tag{2.6}$$

To determine the values of H, we need at least four pairs of corresponding points, $(x, y) \leftrightarrow (x', y')$, between the two images. For each pair, we can set up two equations based on the above transformations.

Given a point correspondence $(x, y) \leftrightarrow (x', y')$, the following two equations are derived:

$$x'(h_{31}x + h_{32}y + h_{33}) = h_{11}x + h_{12}y + h_{13} \tag{2.7}$$

$$y'(h_{31}x + h_{32}y + h_{33}) = h_{21}x + h_{22}y + h_{23} \tag{2.8}$$

These equations can be expanded and rearranged into a linear system:

$$\begin{cases} xh_{11} + yh_{12} + h_{13} - x'h_{31}x - x'h_{32}y - x'h_{33} = 0 \\ xh_{21} + yh_{22} + h_{23} - y'h_{31}x - y'h_{32}y - y'h_{33} = 0 \end{cases} \tag{2.9}$$

For four points, this results in eight equations, which are sufficient to solve for the eight unknowns variables (since one of the elements of H is typically set to 1 for scaling).

The system of linear equations for multiple points can be written in matrix form as:

$$A\mathbf{h} = 0 \tag{2.10}$$

where A is a matrix that contains the coordinates of the points, and $\mathbf{h}$ is a vector that holds the elements of the homography matrix. To solve this equation, we can apply methods like Singular Value Decomposition (SVD) to find the solution that minimizes error, providing us with the homography matrix H. Once H is obtained, it can be used to map any point from one image to the corresponding point in the other image [36][37].

2.4.1. Metric for Homography Transformation

To verify whether the points are maps accurately onto the other perspective, the re-projection error is calculated. The re-projection error evaluates the accuracy of the homography transformation by comparing the bounding box points in one view with the transformed points in the other view using the homography matrix [38]. The re-projection error is determined as follows:

$$\text{Re-projection Error} = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}_i' - H\mathbf{x}_i\|_2 \tag{2.11}$$

where

- $\mathbf{x}_i$: Original 2D points in the OBU image (in homogeneous coordinates $[x, y, 1]$).
- $\mathbf{x}_i'$: Actual observed 2D points in the RSU image.
- H: Homography matrix.
- $H\mathbf{x}_i$: Transformed points from the OBU view to the RSU using H, normalized to $[x'/w, y'/w, 1]$.
- $\|\cdot\|_2$: Euclidean distance between the observed and transformed points.
- N: Total number of point correspondences.

2.5. Oriented FAST and Rotated BRIEF (ORB)

The Oriented FAST and Rotated BRIEF (ORB) algorithm is a robust and reliable feature detection and description method particularly used for object recognition, image matching, and structure-from-motion [39]. ORB combines two widely used techniques: FAST (Features from Accelerated Segment Test) for keypoint detection and BRIEF (Binary Robust Independent Elementary Features) for feature description [39].

keypoints and feature descriptors, two essential components of feature-based image processing. Keypoints are recognizable and recurring areas in a picture, like edges, corners, or blobs, that are distinguished by changes in intensity. These dots, which are distinguished by their spatial coordinates, scale, and orientation, are essential for recognizing and following visual elements across several images [39]. Conversely, feature descriptors are numerical depictions that encode the local visual information surrounding key spots. Despite changes in scale, rotation, and illumination, feature descriptors enable effective comparison and matching by succinctly and robustly expressing the main points [39]. Together, keypoints and descriptors form the foundation for reliable image analysis and correspondence.

The initial step in ORB's keypoint detection process involves the FAST algorithm, a corner detection method that identifies keypoints by analyzing the intensity variations of pixels within a circular neighborhood surrounding a candidate pixel, it works by selecting a pixel and comparing its intensity to that of the pixels around it in a circle. If a significant number of adjacent pixels are brighter or darker than the candidate pixel, it is deemed a corner and thus a keypoint [40][39]. FAST is known for its computational simplicity and speed, making it highly convenient for real-time applications [39]. However, it lacks scale and rotational invariance, which can affect its robustness in dynamic environments. To address this, ORB enhances FAST by computing the orientation of each keypoint using the intensity centroid method, enabling rotational invariance [39].

The next step is BRIEF which is used in ORB for feature description. It creates a concise and effective descriptor by comparing the brightness of pixel pairs in a little region surrounding the keypoint [39][40]. Despite being computationally efficient and light, BRIEF is intrinsically rotation-sensitive. By matching its sample pattern to the orientation of the keypoint, ORB adjusts BRIEF, guaranteeing rotational invariance and enhancing robustness [39].

Figure 2.10.: Features Matching [40]

ORB is a key component of feature-level fusion in sensor fusion, where strong feature matching and detection are required to align data from multiple sensors. The image 2.10 shows the feature matching between two same images using the ORB algorithm. Keypoints have been detected in both images, and the matching features are connected by lines. Each line represents a matched feature pair between the two images, indicating the same area. The visualized lines demonstrate how ORB establishes link between the features, which are essential for aligning the two perspectives or performing transformations like homography. Because of its effectiveness and dependability, ORB is a top option for applications that need for precise feature correspondence, even in the face of difficult circumstances like motion blur, occlusions, or changing lighting [40].

2.6. Vehicle to Everything (V2X)

The advancement of connected car technology is leading the world into a new era where seamless interaction between vehicles, drivers, and their surroundings has become standard [41]. This shift is enabling vehicles to communicate with each other, which is facilitated by V2X. The term "Vehicle-to-Everything" refers to the ability of vehicles to communicate with their external environment, including other vehicles, pedestrians, infrastructure, and networks and can be described as Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), Vehicle-to-pedestrian (V2P) and Vehicle-to-Network (V2N). This technology facilitates the exchange of information, enabling vehicles to share data about their speed, location, road conditions, and potential hazards. This communication helps provide timely warnings to drivers, enhances traffic management, improves road safety, and improves overall driving experience. V2X communication with other vehicle sensing technologies marks a significant milestone in the advancement of urban mobility and transportation system [7].

Figure 2.11.: Vehicle to Everything (V2X) [41]

There are two primary options for enabling V2X communications, Dedicated Short Range Communication (DSRC) and Cellular Network technologies. Generally DSRC typically refers to a wireless technology developed for automotive and intelligent transportation system (ITS) applications, enabling short-range information exchange between RSU and OBU [42]. On the other hand, cellular networks provide extensive coverage and high capacity, with each cell containing a base station that facilitates communication with neighboring cells, supported by a well-established infrastructure [42].It has ability to support a large number of connections, making it particularly useful populated urban areas, especially with 5G technology. This capability allows for a wider range

of communication applications, including vehicle safety enhancements, efficient traffic management and infotainment services [42].

A major reason for the growth of the global V2X communication system market is the increasing demand for autonomous vehicles. Car manufacturers around the world are working harder to develop and improve self-driving technologies [43]. This trend is driven by the potential for enhanced road safety, improved traffic management, and increased efficiency in transportation systems.

2.6.1. DSRC

The DSRC system is based on a set of IEEE and SAE standards [44]. It operates over dedicated radio spectrum bands, which vary across North America, Europe, and Japan, leading to compatibility issues between these regions. The original V2X communication system utilizes Wireless-Direct Local Area Network (W-DLAN) technology, enabling direct communication between V2V and traffic V2I. When two V2X transmitters are within range, they form a vehicular ad-hoc network, eliminating the need for external communication infrastructure. Figure 2.12 provides an overview of the DSRC spectrum bands allocated by the U.S. FCC and industry Canada in North America, the Electronic Communications Committee of the European Conference of Postal and Telecommunications Administrations (CEPT) in Europe, and the Ministry of Internal Affairs and Communications (MIC) in Japan [42].

DSRC can be used for either one-way or two-way data exchange and operates in the 5.9 GHz band and typically ranges up to several hundred meters and supports high data rates, generally up to 27 Mbps [45], with low latency making it ideal for enhancing safety in transportation. By facilitating V2V and V2I communication, DSRC helps create a more responsive and interconnected vehicular system, contributing to reduced accidents and improved traffic flow.

Although DSRC offers a reliable communication channel between RSU and OBU it has certain limitations. One major drawback is its short range, which limits communication over longer distances. This means that vehicles must remain within the coverage area, and if they are traveling at high speeds, the connection between the OBU and the RSU will only last for a short distance [42]. Furthermore in much traffic or congested area where there are many RSUs and OBUs, DSRC can experience communication congestion. Such limitation does not apply to cellular network technologies, where base stations provide coverage over much larger areas compared to DSRC.

2.6.2. Cellular Network

The shortcomings of DSRC and recent advancements in Cellular Network technologies have motivated the research community to explore cellular-based V2X communications

Region	Band (MHz)	Channelization	In-use or allocated	Applications	Standard	Scope
North America	902-928[a]	Uplink/downlink channels for active and backscatter systems [12]	In-use	Electronic toll collection and commercial and non-commercial vehicle applications [13]	ASTM E2158-01	Physical (PHY) layer
					[14]	Data link layer
	5850-5925	One 10 MHz control channel, six 10 MHz service channels, and one 5 MHz channel (held in reserve) [15]	Allocated	Road safety, passenger infotainment, manufacturer services, and vehicle traffic optimization applications [5]–[7]	IEEE 1609.0	System architecture
					IEEE 1609.2	Security services
					IEEE 1609.3	Logical link control (LLC), network, and transport layers
					IEEE 1609.4	Multi-channel operation
					IEEE 1609.11	Electronic payment
					IEEE 1609.12	Identifier allocation
					IEEE 802.11-2012	PHY and medium access control (MAC) layers
Europe	5470-5725[b] (ITS-G5C)	Dynamic frequency selection (DFS) of a 10 MHz or 20 MHz service channel [16]	Allocated	ITS applications based on V2I communications [16]	ETSI EN 302 571	Requirements for operation in the 5855-5925 MHz band
					ETSI EN 300 674-1	Requirements for operation in the 5795-5815 MHz band
					ETSI ES 202 663 (ITS-G5)	PHY and MAC layers
	5795-5815	Four 5 MHz channels [17]	In-use	Road transport and traffic telematics [18]	ETSI EN 302 665	Communication architecture
					ETSI EN 302 636-3	Network architecture
					ETSI EN 302 636-4-1	Geographical routing functionality
					ETSI TS 102 636-4-2	Geographical routing based on ITS-G5
	5855-5925	One 10 MHz control channel and six 10 MHz service channels [19]	Allocated	Non-safety applications [5855-5875 MHz (ITS-G5B)], safety applications [5875-5905 MHz (ITS-G5A)], and future ITS applications (5905-5925 MHz) [19]	ETSI EN 302 636-6-1	Transmission of IPv6 packets using geographical routing
					ETSI EN 302 636-5-1	Transport layer
					ETSI EN 302 637-2	Format and handling of cooperative awareness messages (CAMs)
					ETSI EN 302 637-3	Format and handling of decentralized environmental notification messages (DENMs)
Japan	755.5-764.5	Single channel [20]	Allocated	Safety applications [20]	ARIB STD-T109	PHY, data link, application, and IVC-RVC layers[c]
	5770-5850[d]	Seven uplink and seven downlink 5 MHz channels [21]	In-use	Toll collection, passenger entertainment, and information provisioning regarding road conditions, local events, and emergent disasters [22], [23]	ARIB STD-T55	PHY, data link, and application layers[e]
					ARIB STD-T75	
					ARIB STD-T88	Application sub-layer for deployment of multiple DSRC applications based on ARIB STD-T75
					ARIB STD-T110	Application interface for deploying non-IP applications based on ARIB STD-T88 and ARIB STD-T75

Figure 2.12.: DSRC Bands Spectrum in America, Europe and Japan [42]

[44]. Cellular Networks provide extensive coverage and high capacity, with each cell containing a base station that facilitates communication with neighboring cells, supported by a well-established infrastructure. They supports very high mobility of vehicles up to 350 km/h [46]. Car manufacturers such as BMW have relied on the cellular networks to provide their vehicles with communication services, mainly targeting infotainment applications and a few V2I-based safety applications [42].

Cellular-V2X provides two modes, in-coverage mode and out-coverage mode. In in-coverage mode, a vehicle is within the cellular network's range, which allows the nearby base station to handle direct message exchanges. This setup helps ensure reliable communication, as the base station can effectively manage data flow and bandwidth allocation [46], while in out-coverage mode the vehicle operates independently, without relying on cellular coverage. it autonomously manages its resources to communicate with other vehicles or infrastructure. This flexibility ensures that vehicles can still maintain connectivity even in areas with poor cellular signal, enhancing the overall robustness of V2X communications [46].

Cellular V2X provides high network capacity to handle large bandwidth demands and wide coverage. This helps vehicles stay connected to base stations for longer

periods, reducing the frequency of switching between them compared to connections with RSUs [47]. Furthermore, its mature technology simplifies implementation and speeds up the deployment of V2X communications.Despite of these advantages, there are some limitations that affect the ability of cellular technology to support reliable V2X communications. One major concern is the centralized control of Cellular Networks, which requires all vehicular data to pass through a base station before reaching its intended recipient, this introduces delays in vehicular networking that are unacceptable for critical safety applications [47].

3. State of Art

In recent years, much work has been done to improve sensor fusion. Researchers have explored different levels of fusion playing a key role in making fusion systems more effective, particularly to enhance object detection and environment perception. V2X integration has also become an important aspect of sensor fusion, with recent studies highlighting how important it is for enabling seamless communication between RSUs and OBUs. These advancements show the growing need for distributed sensors and reliable communication to make object detection and perception systems more robust in automotive applications.

3.1. Sensor Fusion

In [48], a low-level fusion approach is presented, combining data from a laser scanner and a camera sensor by integrating the measurements from both sensors, Sensor information is maintained and more precise data is provided, resulting in a reduction of the signal-to-noise ratio compared to sensors operating individually, thereby improving detection accuracy.

Similarly in a previous study, a framework for feature-level sensor fusion was presented to detect targets in dynamic environments with limited communication [49]. This framework capture low-dimensional features from several infrared sensors placed at different angles and exposed to different light conditions. These extracted features are then combined into clusters using the agglomerative hierarchical clustering algorithm [50] to improve the detection of moving targets.

High level sensor fusion is also applied in straightforward two-sensor fusion systems [51] in which Radar and infrared sensor are fused to enhance vehicle detection. Furthermore laser scanner and stereo vision sensor are fused using high level fusion by Labayrade [52] to introduce collision mitigation system, however, in this case, stereo camera confirms the significance of the objects identified by the laser scanner.

Along with these, there have been numerous implementations of hybrid fusion in various domains, showcasing its ability to combine the advantages of each level to improve the accuracy of sensor integration. In [27] a generic sensor processing architecture for driver assistance was proposed by Naab which consist of both low-level fusion module and high level fusion module. Similarly In [28], Scheunert outlines a multi-level fusion

architecture developed as part of the European project PEeVENT. A key component of this architecture is the perception memory object (PMO), which is available at every stage of data processing. Depending on the object type and model, fusion with the PMO can be performed at various levels [28].

3.1.1. V2X in Sensor Fusion

[53] provides a probabilistic paradigm for improving cooperative perception with V2X data. It focuses on combining data from vehicles and infrastructure to improve object detection and scene understanding, while also tackling issues such as communication delays and data consistency. The approach is intended to maximize perception performance within real-world restrictions.Similarly UniV2X, presented by [54], is an integrated solution for cooperative autonomous driving that combines vehicle and infrastructure data. It stresses spatial and temporal synchronization for cross-view fusion while addressing issues such as bandwidth restrictions and communication delay. The proposed method enhances planning and lowers transmission costs while retaining strong perception and mapping skills.

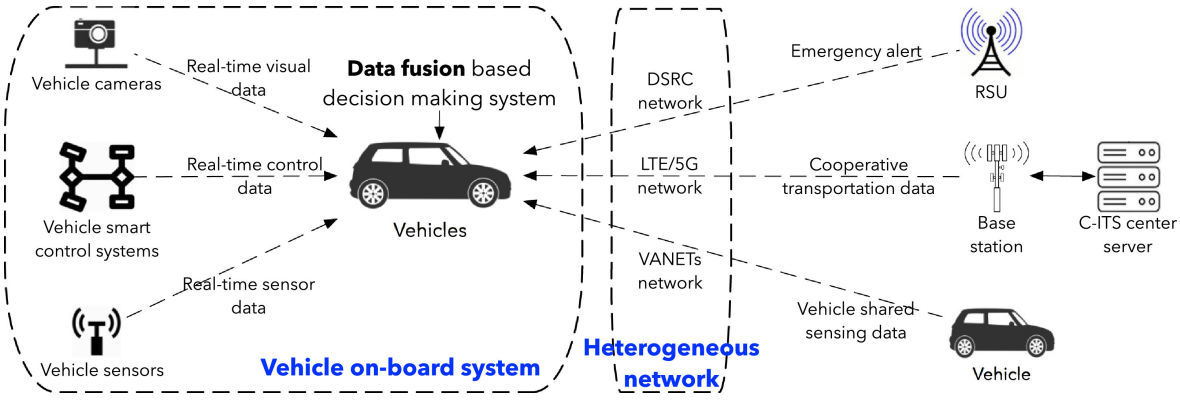

Figure 3.1.: Data Fusion in V2X Communication Networks [55]

3.2. Object Detection

Object detection has been a key focus in the development of perception systems, particularly in automotive applications. Early methods relied on handcrafted features, such as Histograms of Oriented Gradients (HOG), combined with machine learning techniques like SVM to detect objects such as pedestrians and vehicles [56]. These approaches laid the foundation for more advanced methods but were often limited in handling complex environments. Algorithms like YOLO and Faster R-CNN brought real-time performance and improved accuracy, making them suitable for dynamic scenarios [57][58].

In recent years, object detection has increasingly been integrated into multi-sensor setups, such as combination of camera and LiDAR data has shown a good performance and improve reliability in detecting objects by using complementary sensor information [59]. Additionally, the integration of RSUs and OBUs in distributed sensor systems has demonstrated potential in expanding detection coverage and ensuring robustness and reliability in challenging conditions [60].

In sensor fusion applications, where multiple sensors or camera views are used to observe the same scene from different perspectives, ORB has shown significant advantages in feature matching and data alignment. [61] demonstrated the use of ORB in monocular visual odometry, where good feature matching was critical for calculating camera motion and generating accurate maps in real-time applications. Their experiment demonstrated ORB's robustness in managing issues such as motion blur and changing illumination conditions, making it perfect for use in autonomous driving. These developments showcase the ongoing progress in sensor fusion, object detection and alignment methods are improving to meet the need for accurate and reliable perception in difficult real world situations.

4. RSU-OBU Integration Framework (ROIF)

4.1. System Requirements

An automated driving system has specific technical requirements that must be met to ensure effective implementation and reliable operation in various environmental conditions. These requirements encompass a variety of hardware and software components that work together to facilitate accurate perception, address limitations, and enable effective control. Additionally, some requirements shall be implemented to enhance the system's performance and safety and those that could be considered to improve functionality and adaptability in diverse scenarios. The system can achieve its objectives and optimize its overall performance by concentrating on these requirements. The following system requirements are essential for the successful implementation and operation of an automated driving system:

No	**Statement**	**Type**
RQ1	The System **Must** be able to fuse detection from different sources	Functional
RQ2	The system **Must** integrate with external data sources (V2X) to improve decision-making.	Functional
RQ3	The system **Must** be designed for aligning the data from different sources	Functional
RQ4	The System **Shall** avoid sending redundant data to processor and handle uncertainties	Non-Functional
RQ5	The system **Could** be scalable to accommodate additions of new sensors and features without significant redesign	Misc.

Table 4.1.: System Requirements

4.2. ROIF Meaning

ROIF is based on a feature fusion method that integrates data features from multiple sources, such as cameras on the RSU and OBU to improve object detection and enhance

environmental perception. This framework ensures that the data is properly aligned to enable fusion from different perspectives, overcoming challenges such as viewpoint variation, occlusions, and dynamic environments.

Figure 4.1 illustrates ROIF, where two cameras, one on RSU and the other on OBU, capture the same environment from different perspectives. Each camera independently detects objects, generating bounding boxes around each detected object. To provide a comprehensive and unified understanding of the surroundings, these bounding boxes are fused after necessary alignment, ensuring that data from both perspectives are accurately fused. This fusion process enhances object detection accuracy, precision and addresses challenges such as partial visibility and occlusions. After fusion, the system continuously tracks the objects across frames, maintaining their identity and location even the scenarios where objects might temporarily disappear from one camera's field of view. This robust tracking capability ensures reliable perception, making the ROIF well-suited for applications in automated driving and traffic monitoring systems.

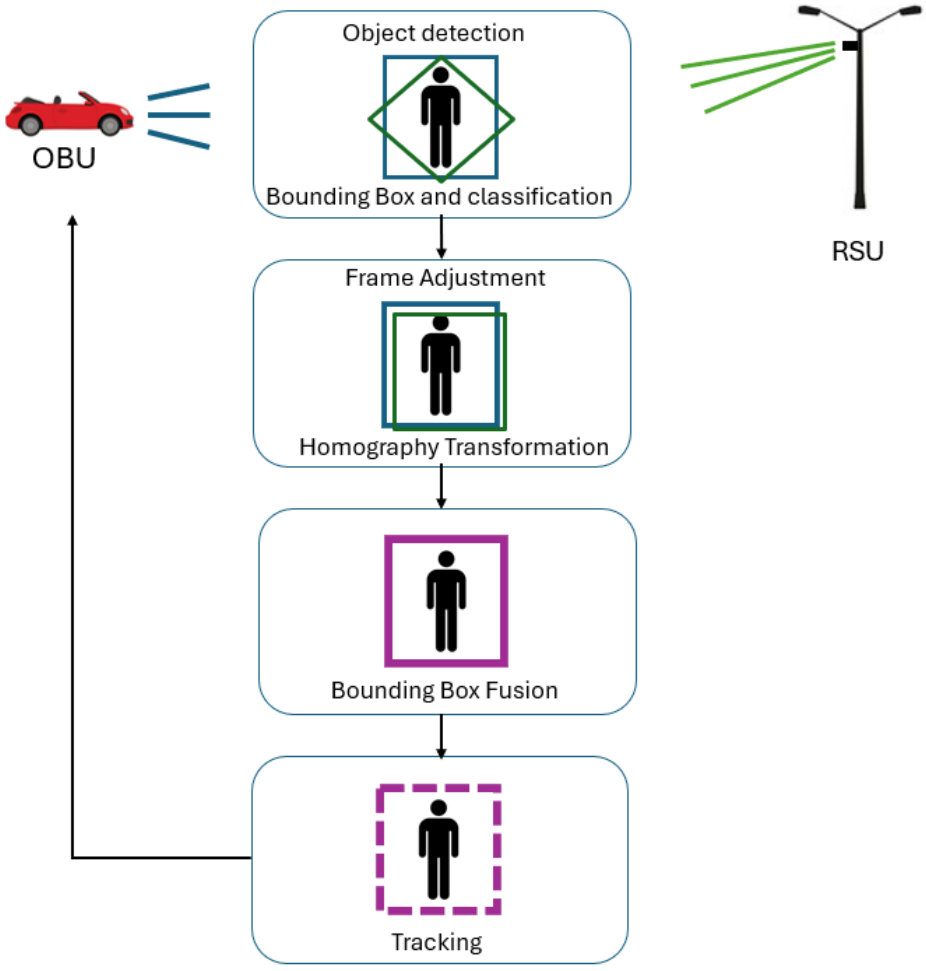

Figure 4.1.: ROIF

4.3. OBU and RSU Camera System

The OBU and RSU camera systems are essential components of the ROIF, these two systems collaborate to provide complementary perspectives that help create a clearer and more accurate view of the environment.

The OBU camera, installed on a moving vehicle, offers a dynamic and constantly changing view of the environment. Its primary function is to capture real-time data from the vehicle's immediate surroundings, enabling the detection of objects such as pedestrians, vehicles, and obstacles with high precision. This real-time data is important for making instantaneous driving decisions. However, the dynamic nature of the OBU camera introduces several challenges in which objects may temporarily leave the camera's field of view due to its limited coverage, resulting in data loss. Additionally, occlusions caused by other vehicles or infrastructure can obstruct visibility, complicating object detection and tracking. Perspective fluctuations and distortions arising from the vehicle's movement further add to the complexity of accurately localizing and interpreting objects [62].

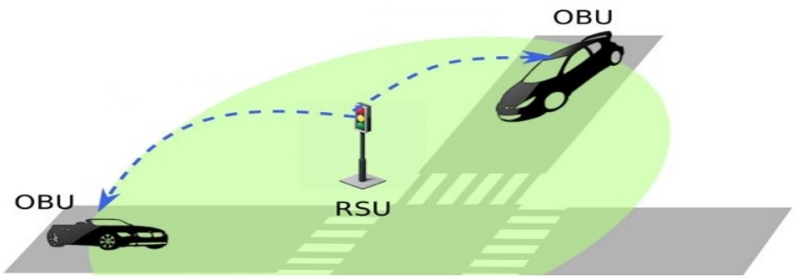

Figure 4.2.: OBU and RSU System [62]

The RSU camera, installed in a fixed position such as on a traffic pole or along the roadside, provides a stable, wide-angle view of the surrounding environment. Its primary function is to capture top-down perspective of the road, enabling the detection of vehicles, pedestrians, and other objects over a much larger area than the OBU camera. This fixed positioning allows the RSU to continuously monitor a wide region without the challenges of movement induced distortion, providing consistent coverage over time [62]. However, the RSU camera also faces its own set of challenges. Since it does not have the flexibility to adapt to dynamic changes in the environment, its coverage is limited to specific fixed locations. While it can detect stationary objects and monitor traffic flow, it is not capable of tracking moving objects in the same way as the OBU camera. Furthermore, its static field of view can leave gaps in detection, especially when objects move outside the camera's coverage or when the environment changes rapidly. Despite these limitations, the RSU camera plays a important role in offering extended range, stable perception that complements the dynamic, close-range detection of the OBU camera. When combined, the RSU's broad coverage and the OBU's adaptability provide a more complete and reliable understanding of the road environment [62].

4.4. Object Detection

In ROIF, the OBU and RSU cameras detects objects within their respective fields of view based on their positions as shown in figure 4.3. Each camera identifies and locates objects in the frame such as vehicles, pedestrians, or obstacles in the environment and then create a boundary around these detected objects, commonly referred to as a bounding box. The bounding box determines the spatial extent of each identified object, allowing for precise positioning and size determination within the image frame.

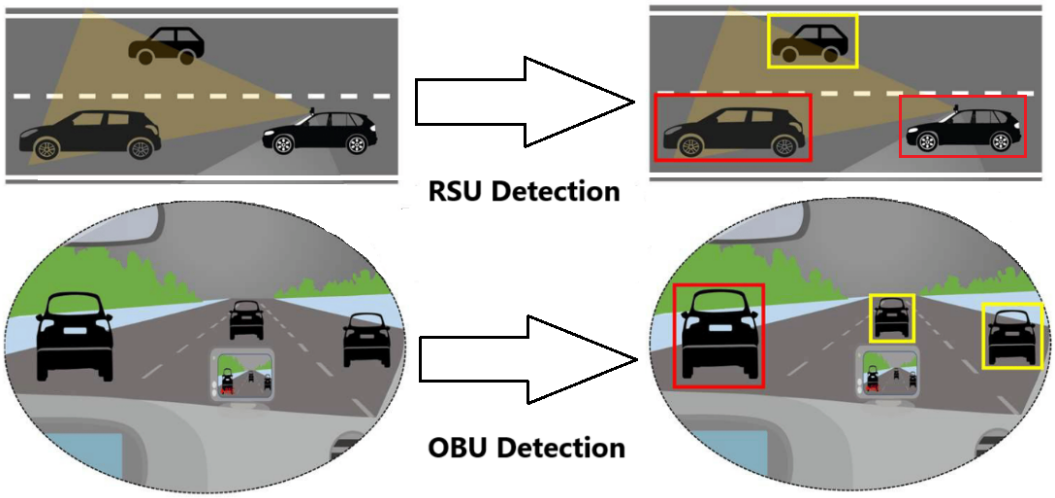

Figure 4.3.: OBU and RSU Object Detection [63]

The object detection process, as depicted in Figure 4.4, begins with a detection phase where the system continuously scans for objects within a given frame. If no object is detected, the system re-evaluates by returning to the detection phase, ensuring robust handling of dynamic or noisy inputs. Upon detecting an object, the process diverges into two parallel paths to perform complementary tasks.

The first path focuses on predicting the type of the detected object. Once the type is predicted, the system classifies the object by organizing it into predefined categories based on its features and attributes. The classification step differentiates between objects of similar appearance but varying dimensions, such as different vehicle types or sizes. The second path addresses the spatial characteristics of the object by predicting its position within the frame. Using this information, the system determines the precise location of the object and places a 2D bounding box around it. This bounding box not only visualizes the object's presence but also assigns it a unique identifier (ID), which is also useful for tasks such as object tracking and further analysis.

After object detection phase we have bounding boxes where each bounding box is assigned an unique ID, along with a label and a confidence score, which indicates the likelihood of the object being correctly identified in the frame. Both the OBU and RSU cameras

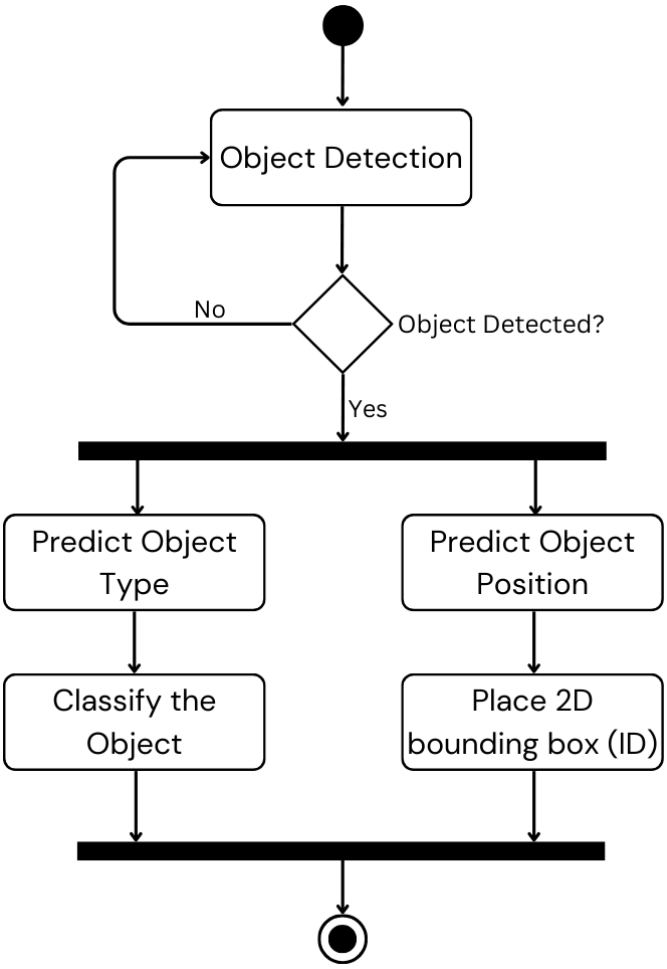

Figure 4.4.: Activity Chart for Object Detection and Classification

perform the object detection and classification process independently, ensuring that data from each perspective is processed in isolation. This method ensure that the system accurately identify object positions within the context of each camera's viewpoint, laying the groundwork for the necessary transformations when aligning data from different perspectives. The separation of processing also allows for greater flexibility in combining data while preserving the unique context of each camera's view.

4.5. Bounding Box Transformation

Once we have the bounding boxes our next step in to perform fusion of these bounding boxes. However, due to the differing viewpoints of the cameras, directly applying bounding box fusion is challenging, we have to align them to one frame. To align the bounding boxes accurately, we use a homography transformation also known as

perspective transformation. In this process, the bounding box from the OBU (which may shift as the camera is not static) is transformed to align with the RSU bounding box. This transformation allows us to overlay the projected bounding box from the OBU onto the RSU, positioning it approximately where the detected object appears in the top down view of RSU camera.

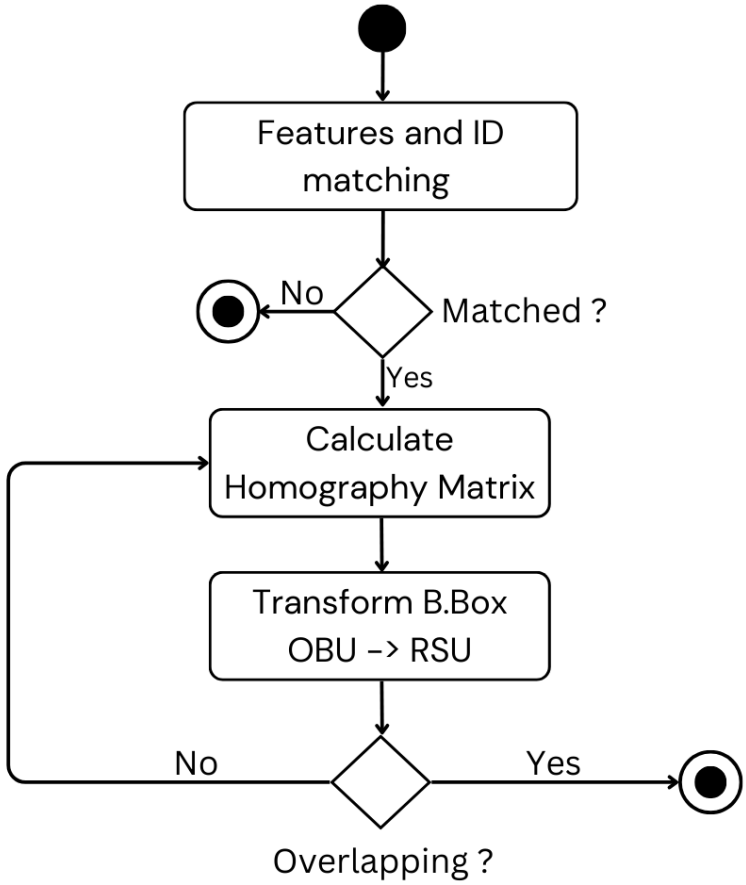

Figure 4.5.: Activity Chart for Homography Transformation

To perform the homography transformation, the source and destination images are first defined. In this case the source image corresponds to the OBU, while the destination image represents the fixed RSU. The corner points of the bounding box around the detected object in both views are identified to ensure that the bounding box from the OBU view precisely overlays the corresponding object in the RSU. This alignment is achieved by matching the ID and features based on the color and texture of the detected object to its counterpart in the other view. if the feature and ID are not matched the system stop the process and If a match is successfully established, the system proceeds to calculate the homography matrix.

Following the computation of the homography matrix, the system applies it to transform the OBU bounding box into the RSU perspective. This transformation ensures that the object's position and dimensions in the OBU view are accurately represented within the RSU coordinate frame. Once the transformation is completed, the system checks for overlapping bounding boxes between the transformed OBU bounding boxes and the RSU bounding boxes. If no overlapping is detected, the process returns to the calculation phase where homography matrix is recalculated for refinement. However, if overlapping is identified, it confirms that the two cameras are correctly detecting the same object. This step lays the groundwork for further tasks such as fusion, object tracking and decision-making based on both views data.

4.6. Bounding Boxes Fusion

Once both scenes are aligned, the system combines the bounding boxes from each camera. This fusion step helps to create a unified and more accurate representation of the detected objects in the environment, overcoming the limitations of individual viewpoints. If the OBU camera records a portion of an object that the RSU camera fully detects, the fused bounding box will encompass the entire object, lowering the possibility of redundant or incomplete detections.

As illustrated in figure 4.6, the first step involves calculating the ϵ which is referred as IoU [32] of OBU transformed and RSU bounding boxes to determine if they should be merged or not. If the ϵ is less than 10%, the overlap is deemed insignificant, and the bounding boxes are not fused, as they likely represent detections of different objects. However, if the ϵ falls within the range of 10% to 100%, the bounding boxes are considered for fusion, as they likely correspond to the same object observed from different perspectives. For these overlapping bounding boxes, weights are computed based on their respective confidence scores. These weights reflect the trustworthiness of each detection, considering factors such as precision and accuracy.

Using the assigned weights, a weighted average is calculated for the bounding box coordinates x, y, as well as its width and height [64]. Since the x, y, w, and h of each bounding box in both OBU and RSU are already known, the averaging can be performed by calculating the mean values of these parameters. We define each bounding box i with the following parameters:

- x_i: x-coordinate of the center
- y_i: y-coordinate of the center
- w_i: width
- h_i: height
- s_i: confidence score for box i

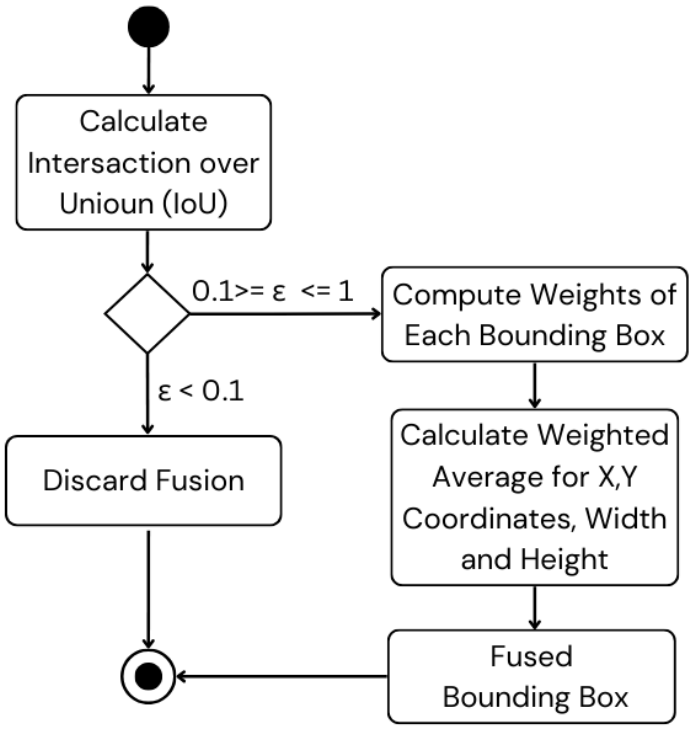

Figure 4.6.: Bounding Boxes Fusion

Given N bounding boxes, the weighted coordinates (x, y), width, and height of the fused bounding box can be computed as follows:

$$x_{\text{fused}} = \frac{\sum_{i=1}^{N} s_i \cdot x_i}{\sum_{i=1}^{N} s_i}$$

$$y_{\text{fused}} = \frac{\sum_{i=1}^{N} s_i \cdot y_i}{\sum_{i=1}^{N} s_i}$$

$$w_{\text{fused}} = \frac{\sum_{i=1}^{N} s_i \cdot w_i}{\sum_{i=1}^{N} s_i}$$

$$h_{\text{fused}} = \frac{\sum_{i=1}^{N} s_i \cdot h_i}{\sum_{i=1}^{N} s_i}$$

where:

- x_{fused} and y_{fused} represent the center coordinates of the fused bounding box,

- w_{fused} and h_{fused} represent the width and height of the fused bounding box.

This ensures that the fused bounding box accurately represents the detected object by balancing contributions from both OBU and RSU perspectives. The center and dimensions of the fused box are determined in a way that reflects the most reliable representation of the object. Finally, a new fused bounding box is generated, combining information from both perspectives to ensure a more robust and accurate detection.

4.7. Object Tracking

Object tracking is a one of the most important part of ROIF, ensuring accurate and consistent monitoring of objects over time. As each detected object is assigned a unique ID, enabling the system to distinguish between different objects, even when their features are similar. This ID-based approach ensures continuity in tracking, preventing confusion when objects move, overlap, or reappear in subsequent frames. The ID is assigned based on the vehicle's relative position within the frame, making it easier to reliably track the vehicle over time.

4.7.1. State Prediction

To ensure accurate and consistent tracking of objects across frames, a of state prediction algorithm is utilized. The algorithm will predict the future state of a detected object based on its current state using a motion model, such as the constant velocity or constant acceleration model. When new observations are received, the filter updates the predicted state by incorporating these observations, thereby refining the estimate of the object's true state.

4.7.2. Data Association

To associate detected objects with existing tracks, the cost metric is applied. This requires calculating a cost matrix based on the IoU metric, which helps identify the best match between detected objects and predicted objects. The tracking algorithm can be explained in figure 4.7. For each detection track pair, the IoU is computed, this information is organized into a 2D matrix, where rows correspond to detections and columns to tracks. The detections in the current frame are represented as $D = \{D_1, D_2, \ldots, D_n\}$ and the tracks prediction are represented as $T = \{T_1, T_2, \ldots, T_m\}$.

For each pair of detection D_i and track T_j we need to calculate the IoU

$$\text{IoU}(D_i, T_j) = \frac{\text{Area of Overlap}(D_i, T_j)}{\text{Area of Union}(D_i, T_j)}$$

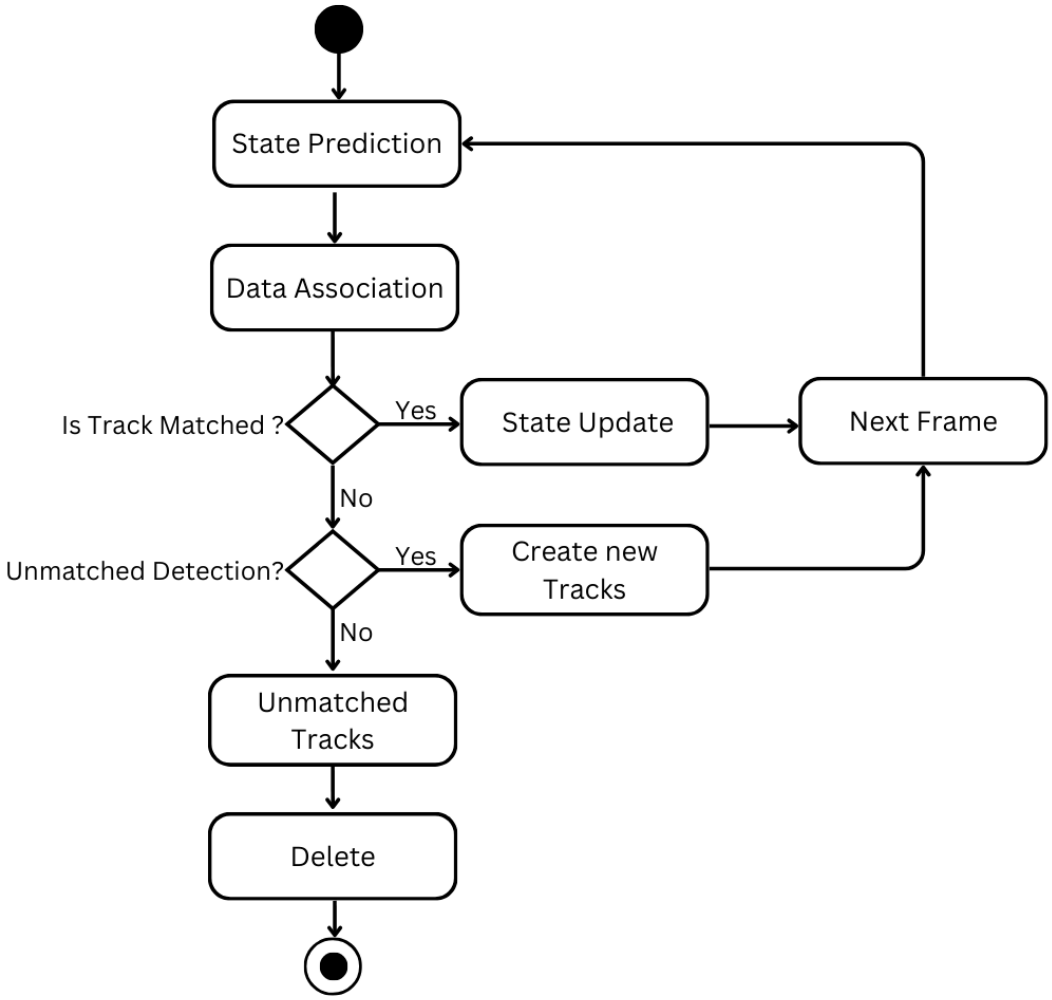

Figure 4.7.: Activity Chart for Object Tracking

The result will be a value in the range $[0, 1]$, where 0 indicates no overlap and 1 indicates complete overlap.

Based on detection and track we can compute cost matrix C, where C is an $n \times m$ matrix (with n detections and m tracks). Each element of the matrix $C(i, j)$ represents the cost of associating detection D_i with track T_j.

Since a higher IoU indicates a better match, invert the IoU values to create the cost matrix:

$$C(i, j) = 1 - \text{IoU}(D_i, T_j)$$

Using the cost matrix, the most optimal matches between detected objects and predicted tracks are selected. It minimizes the overall cost, ensuring the most accurate associations between detections and tracks. Once the algorithm performs the matching, each track is classified as either a matched detection, an unmatched detection, or an unmatched track.

4.7.3. Track Management

Track management is responsible for maintaining accurate and consistent information about tracked objects over time. It involves updating existing tracks, creating new ones, and deleting tracks when necessary, all based on the results of data association. The primary goal of track management is to ensure that each detected object is correctly assigned to its corresponding track. If an object is not on the correct track, the system compares it with existing tracks and reassigns it as needed. The process is divided into three main parts: matched tracks, new matched tracks, and unmatched tracks,

A matched track signifies that a newly observed track has successfully aligned with a previously predicted track, allowing their positions and velocities to be updated and refined, the system proceeds to the state update phase. Here, the track's state is refined by integrating the predicted state with the observed state, thereby improving the accuracy of the track. Once updated, the system progresses to process the next frame, and the cycle repeats.

If a detected object cannot be matched to an existing track during data association, the system determines whether it represents an unmatched detection. For unmatched detections, the system creates a new track, initializing the tracking process for the newly identified object. Conversely, if a track remains unmatched (i.e., it does not correspond to any current detections), it is flagged as an unmatched track. These unmatched tracks are subjected to a deletion criterion, which removes obsolete or irrelevant tracks to optimize system performance and maintain efficiency.

This tracking method ensures continuous and accurate object tracking by integrating predictive modeling, data association, and adaptive track management. By handling unmatched detections and efficiently removing unmatched tracks, the system maintains a streamlined and robust tracking pipeline that supports dynamic environments and different scenarios.

4.8. Data Processing within OBU

Each OBU and RSU camera independently performs object detection, placing bounding boxes around detected objects within their respective fields of view. A DSRC link facilitates real-time communication between the OBU and RSU, enabling efficient data transfer as shown in figure 4.8. Through this link, the RSU transmits its object detection

results, including IDs, classifications, confidence levels, bounding box coordinates, and trajectory predictions, to the OBU.

Once the RSU's data is received, the processing is performed within the OBU. During this step, the system matches the corresponding features between the RSU and OBU bounding boxes to identify the same objects captured from different perspectives. A transformation aligns the OBU data to the RSU's perspective. After this alignment, the bounding boxes from both sources are fused. Once the fused data is generated, the OBU uses it to make informed decisions, such as detecting potential hazards or navigating complex environments. The OBU continuously tracks the fused objects to maintain situational awareness, ensuring a seamless flow of information for autonomous decision-making and improved vehicle safety.

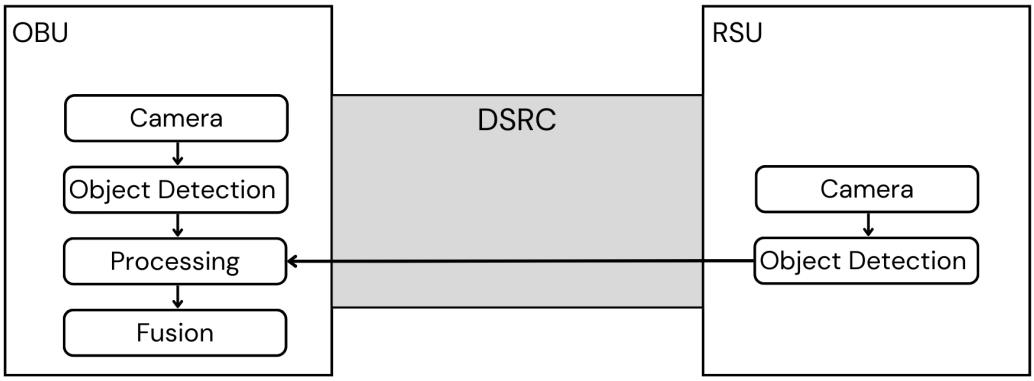

Figure 4.8.: Data Processing within OBU

5. Implementation

The implementation phase translates ROIF into actionable and integrated software modules. This architecture incorporates multiple layers where sensor inputs from cameras are processed in real-time, fused through feature fusion techniques as described in the earlier sections. The software is also structured to handle V2X communication through DSRC protocol for data exchange between OBU and RSU.

5.1. System Setup and Configuration

Figure 5.1.: Experimental Setup at ITIV KIT

The experimental setup shown in figure 5.1 includes a scaled down automated truck on a setup named as CAVIL (Collaborative Automated Vehicle-in-the-Loop), which operates on a conveyor belt to simulate a controlled driving environment. OBU is equipped with a front-facing camera, providing a continuous forward view of the scene ahead as shown in figure 5.2. Additionally, a stationary camera is installed on a pole above the conveyor belt referred as RSU, offering a top down, bird's eye perspective of the same environment as

shown in figure 5.3. This dual camera setup enables simultaneous detection of the scene from both viewpoints OBU and RSU which enhances spatial awareness and supports improved accuracy in tasks like object detection and tracking.

Figure 5.2.: OBU Camera view

Figure 5.3.: RSU Camera view

The OBU vehicle is powered by a 12V battery and is equipped with a Raspberry Pi 5, which functions as the Telematics Control Unit (TCU), and two NVIDIA Jetson Orin Nano SoCs (System-on-Chip), which serve as redundant Electronic Control Units (ECUs). These components collectively manage the OBU's steering and velocity based on sensor data and control algorithms. Various sensors, including a distance sensor and an onboard Intel Depth Camera D455, provide input to help the OBU navigate the conveyor belt. The RSU consists of an Intel Depth Camera D455 mounted on a pole, serving as an infrastructure-based sensor to provide an additional bird's eye perspective. This perspective supplements the OBU front camera by enhancing situational awareness and aiding in precise vehicle positioning on the track, thereby improving system reliability.

Both the OBU and RSU process their respective camera streams independently, performing object detection and tracking for each perspective. The results from both cameras

are then sent to the OBU, where the fusion of object detection data takes place. The OBU integrates this information, including object IDs, classifications, and bounding box coordinates, and adjusts the vehicle's position and steering accordingly. Communication between the RSU and the OBU is managed via a DSRC interface as shown in figure 5.4. The Raspberry Pi 5, functioning as the TCU, oversees telematics and network management, serving as the primary interface for receiving data from the RSU [65]. The received data, including object IDs, classifications, and bounding box coordinates, is processed by the NVIDIA Jetson Orin Nano to execute the necessary control actions. DSRC facilitates low-latency, reliable communication, ensuring the OBU can promptly respond to real-time updates.

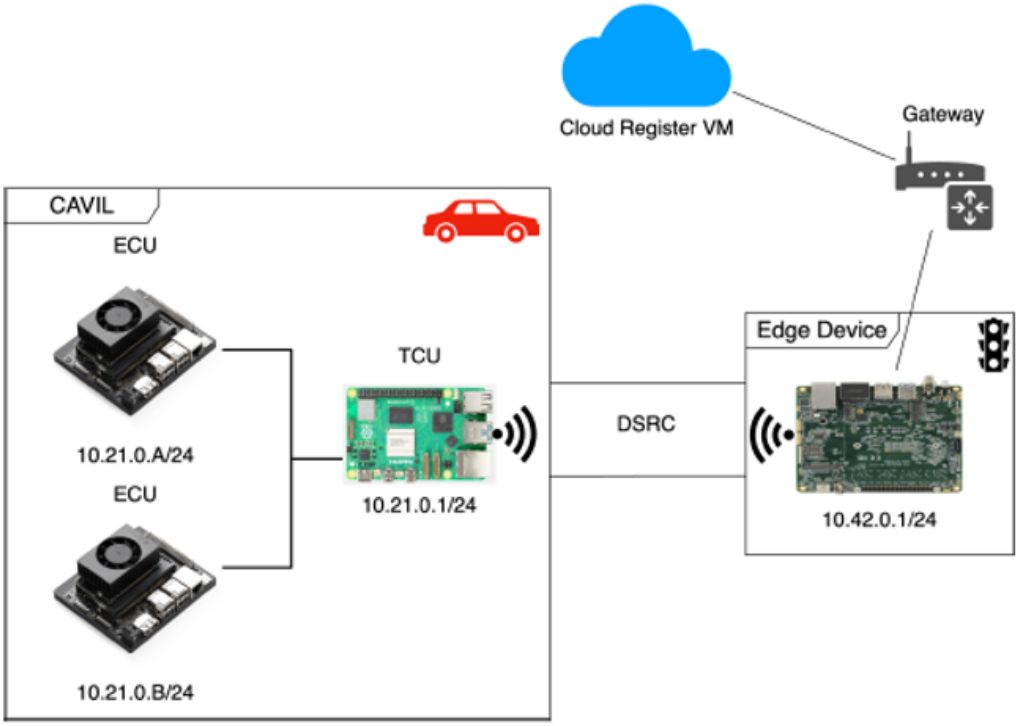

Figure 5.4.: Communication between OBU and RSU through DSRC [65]

5.1.1. ROS 2 and NORA

The implementation of this project was carried out using ROS 2 and NORA as the core frameworks for managing the software architecture. ROS 2 (Robot Operating System 2) is a powerful framework designed to build modular and distributed systems, particularly for applications that require real-time data processing and integration of multiple sensors. ROS 2 core communication model consists of nodes and topics. Nodes are independent processes that either publish data to or subscribe to topics. These topics act as data channels, facilitating efficient message transmission across different system

components [65]. On the other hand, the NORA framework, built on top of ROS 2, offers enhanced flexibility for managing node configurations and dependencies. NORA provides a dynamic approach for configuring ROS 2 nodes, where nodes do not directly publish or subscribe to topics. Instead, they describe their dependencies and providers within a configuration file. NORA then handles the dynamic configuration of the nodes, ensuring that all dependencies are met and allowing the system to be optimized for performance. This abstraction simplifies node management, as it eliminates the need for manual configuration, providing a scalable and efficient solution for complex systems [65].

During the design phase, each camera was configured to publish data to a dedicated topic within a container, with the system automatically managing the dependencies and configurations of each node. This setup ensured clear separation of data streams from each camera, preventing data overlap and allowing individual handling of each camera's data based on its unique perspective. The RSU and OBU cameras each had their own topics for publishing image data in real time. The system dynamically managed the subscription to these topics in separate containers, allowing each data stream to be processed independently. This enabled the system to processing necessary for transformations, object detection, and other system operations on each feed for efficient system performance.

5.2. Object Detection and Classification

The object detection pipeline harnesses the power of the YOLOv3 pre-trained model, which efficiently detects objects in an image. The process begins with the loading the pre-trained weights of YOLOv3 model, which have been trained on large datasets such as COCO (Common Objects in Context) and ImageNet. These weights provide the model with a strong foundation, enabling it to detect a wide range of objects [66]. Before the input image is fed into the YOLOv3 network, it undergoes a preprocessing stage. This includes resizing the image to the standard input size of 416×416 pixels for YOLOv3, and normalizing the pixel values to the range [0,1]. This normalization aids in stabilizing the training and inference processes by reducing the variance in input data. The normalized image is then converted into a blob format, a compact representation suitable for efficient processing by the deep learning framework. This blob is then passed through the YOLOv3 model, which impressively performs feature extraction and object detection in a single forward pass. The model outputs three key results: the bounding box coordinates and associated object IDs for each detected object, a confidence score indicating the likelihood of accurate detection, and a class probability distribution specifying the predicted class of each detected object.

To restrict the detection to one bounding box per object instead of multiple bounding boxes for the same object, NMS was utilized. NMS ensures that for each detected object, only the bounding box with the highest confidence score is retained, while overlapping

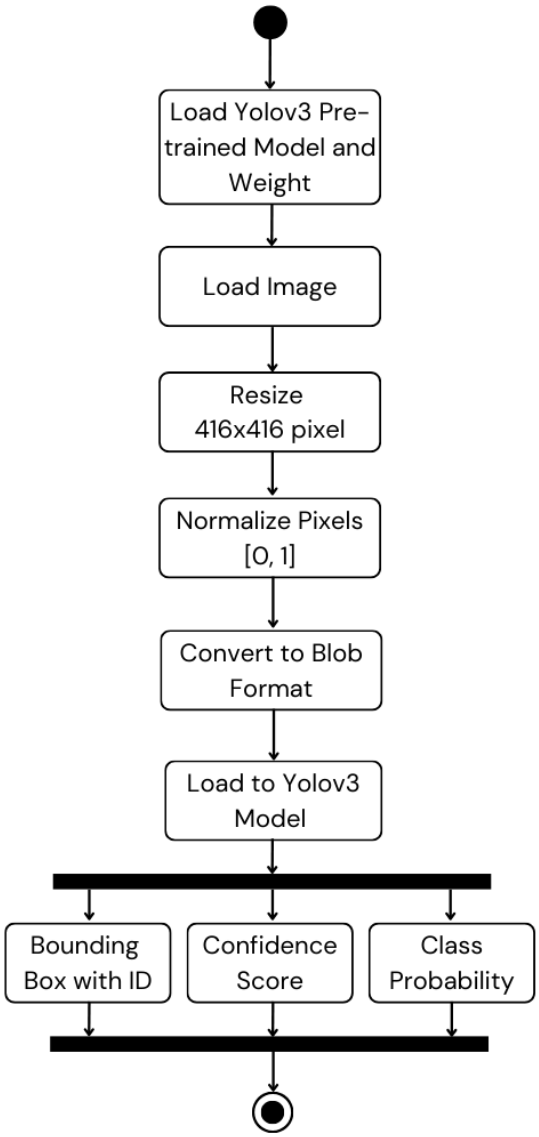

Figure 5.5.: Object Detection using Yolov3

boxes with significant IoU are removed. This technique refines the detection results, ensuring that each object is represented by a single, accurate bounding box.

5.3. Features Matching and Transformation

The implementation of feature matching and homography transformation in this system begins with detecting keypoints in the images captured by the OBU and RSU cameras using the ORB algorithm. Keypoints are distinctive features, such as texture and color that are reliably identifiable across different perspectives [39]. We restrict our keypoints to be detected within the bounding box of each perspective (to avoid any false detection). Once detected, a conditional check ensures that a sufficient number of keypoints exist in both images. If no keypoints are detected, the algorithm terminates for that frame. If keypoints are present, the ORB algorithm extracts descriptors for each keypoint. These descriptors are numerical depictions of the neighborhood around each keypoint, useful for comparing features between RSU and OBU images.

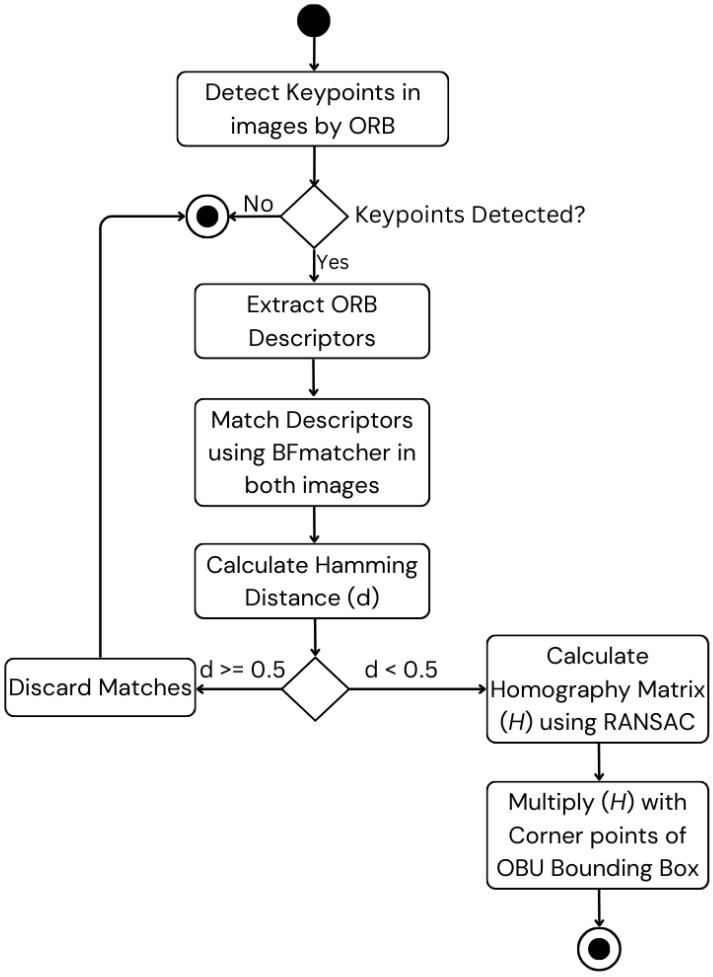

Figure 5.6.: Algorithm for Feature Matching and Transformation

The next step involves matching the extracted descriptors from the OBU and RSU images using a brute-force matching algorithm (BFMatcher) [67]. The similarity of the descriptors is measured using the Hamming distance, which quantifies the number of differing bits between two binary descriptors. Matches are filtered based on a predefined threshold, where only matches with a Hamming distance (d) less then 0.5 are retained. Matches with distance greater then or equal to 0.5 are discarded as they are deemed unreliable for subsequent calculations. From the filtered matches, a homography matrix (H) is calculated using the RANSAC (Random Sample Consensus) algorithm. RANSAC helps identify the best transformation matrix while excluding outliers, ensuring a robust estimation of the homography [68]. The homography matrix is then applied to the corner points of the bounding box detected in the OBU image. By multiplying these corner points (in homogeneous coordinates) with the homography matrix, the bounding box is transformed into the RSU perspective. The transformed bounding box aligns with the RSU perspective and is ready for further process of bounding box fusion.

5.4. Bounding Box Fusion

As the transformed OBU and RSU bounding boxes are now aligned in the same perspective, the bounding box fusion can be implemented. Figure 5.7 illustrates the bounding boxes from different views, which are now ready to be fused. As shown in Figure 5.7(a) and 5.7(b), the bounding boxes represent the RSU and OBU views from individual cameras, while Figure 5.7(c) depicts the transformed OBU (green) bounding box mapped onto the RSU view. The IoU is calculated for each pair of bounding boxes from the RSU and transformed OBU. The IoU score quantifies the overlap between these two bounding boxes, and only pairs with an IoU score greater than or equal to 10% are considered for fusion. To compute the IoU, the intersection points between the two bounding boxes are first determined, which are calculated as follows:

$$x_{\text{min, inter}} = \max(x_{\text{min, red}}, x_{\text{min, green}}) \tag{5.1}$$

$$y_{\text{min, inter}} = \max(y_{\text{min, red}}, y_{\text{min, green}}) \tag{5.2}$$

$$x_{\text{max, inter}} = \min(x_{\text{max, red}}, x_{\text{max, green}}) \tag{5.3}$$

$$y_{\text{max, inter}} = \min(y_{\text{max, red}}, y_{\text{max, green}}) \tag{5.4}$$

The intersection area or overlapping between bounding boxes does not exist, i.e., when $x_{\text{min, inter}} \geq x_{\text{max, inter}}$ or $y_{\text{min, inter}} \geq y_{\text{max, inter}}$, the IoU is zero. Similarly for the union the total area covered by both bounding boxes is calculated as:

$$\text{Area of Union} = \text{Area of red box} + \text{Area of green box} - \text{Area of Intersection} \tag{5.5}$$

After calculating the IoU score and identifying bounding box pairs with a threshold greater than 10%, the next step is to perform bounding box fusion using weighted averaging.

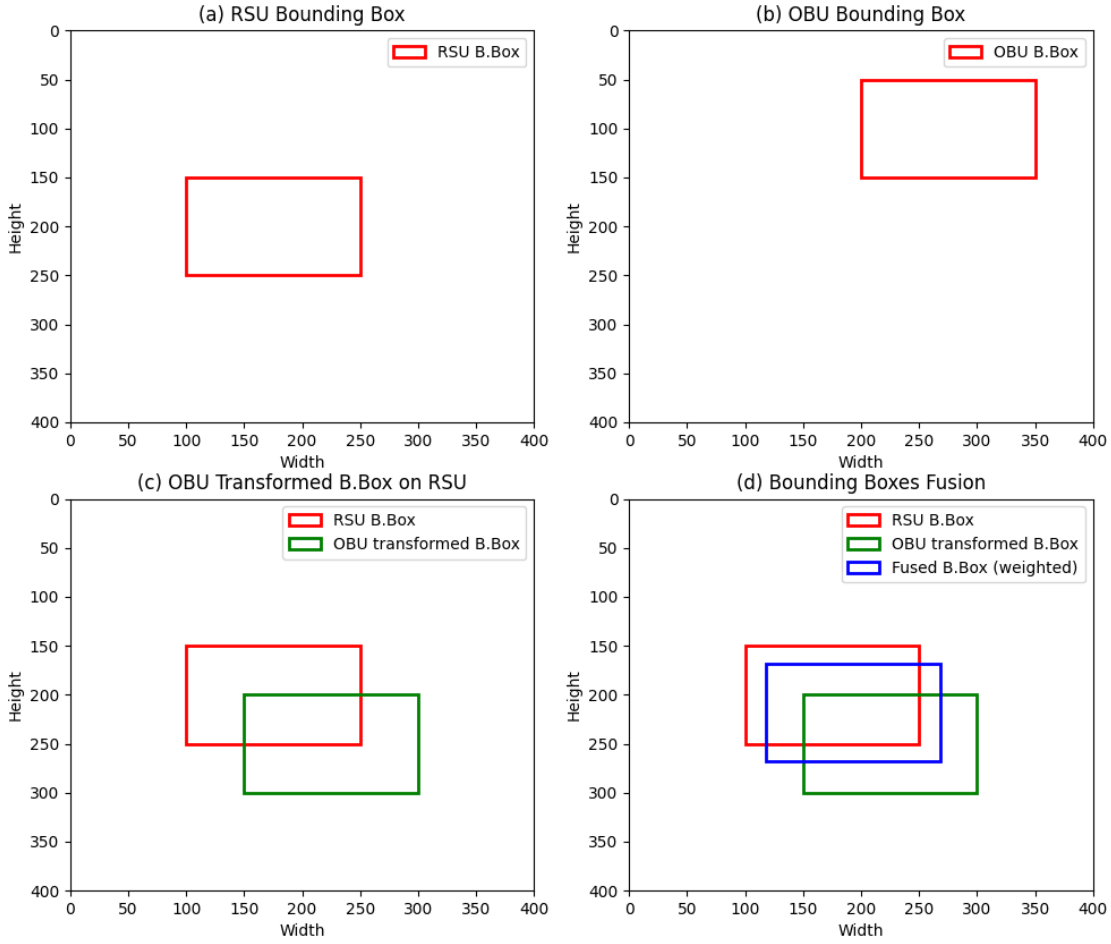

Figure 5.7.: Plots for Bounding Boxes Fusion

The weight assigned to each bounding box is proportional to its confidence score. The top-left and bottom-right corners of the fused bounding box are computed through weighted averaging, ensuring that the resulting bounding box reflects the contributions of both sources. More importance is given to the bounding box with the higher confidence score, leading to a more precise and reliable fusion outcome. The top-left corner ($x_{\text{fused\_tl}}, y_{\text{fused\_tl}}$) of the fused bounding box is calculated as:

$$x_{\text{fused\_tl}} = \frac{x_{\text{red\_tl}} \cdot w_{\text{red}} + x_{\text{green\_tl}} \cdot w_{\text{green}}}{\text{total\_weight}} \tag{5.6}$$

$$y_{\text{fused\_tl}} = \frac{y_{\text{red\_tl}} \cdot w_{\text{red}} + y_{\text{green\_tl}} \cdot w_{\text{green}}}{\text{total\_weight}} \tag{5.7}$$

Similarly the bottom-right corner ($x_{\text{fused\_br}}, y_{\text{fused\_br}}$) of the fused bounding box is calculated as:

$$x_{\text{fused\_br}} = \frac{x_{\text{red\_br}} \cdot w_{\text{red}} + x_{\text{green\_br}} \cdot w_{\text{green}}}{\text{total\_weight}} \tag{5.8}$$

$$y_{\text{fused\_br}} = \frac{y_{\text{red\_br}} \cdot w_{\text{red}} + y_{\text{green\_br}} \cdot w_{\text{green}}}{\text{total\_weight}} \tag{5.9}$$

With the top-left and bottom-right corners determined, we calculate the width and height of the fused bounding box as following:

$$\text{Width} = x_{\text{fused\_br}} - x_{\text{fused\_tl}} \tag{5.10}$$

$$\text{Height} = y_{\text{fused\_br}} - y_{\text{fused\_tl}} \tag{5.11}$$

The result of the weighted averaging will produce a new fused bounding box, as illustrated in Figure 5.7(d) in blue. The dimensions of this fused bounding box, including its width and height, are derived from the weighted contributions of the original red and green bounding boxes, representing the RSU and transformed OBU views.

6. Results and Discussion

To test our implementation, two additional vehicles were placed in front of the OBU, as shown in Figure 6.1. These vehicles were visible in both the camera views. Since the RSU camera was static, it consistently captured both vehicles in its field of view. However, the OBU camera being dynamic, alternated between capturing both vehicles and occasionally only one.

Figure 6.1.: Experimental Setup at ITIV KIT

6.1. Object Detection by Individual Camera

Object detection was performed independently by each camera (OBU and RSU). Using YOLOv3, each camera detected objects and placed bounding boxes around them, assigning a classification label, an ID, and a confidence score, as illustrated in Figure 6.2. The confidence scores reflected the accuracy of the detections, varying based on the quality of the detected features.

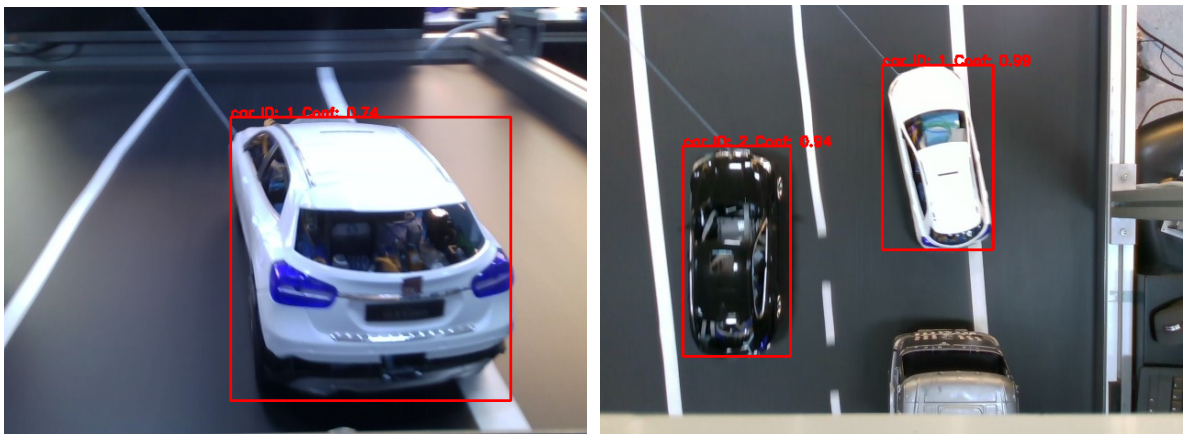

Figure 6.2.: (Left) Object Detection by OBU; (Right) Object Detection by RSU.

The performance of YOLOv3 was evaluated for both camera views using standard object detection metrics, including detection accuracy and precision. The RSU camera, with its stable placement and comprehensive field of view, achieved high confidence scores and demonstrated reliable performance. Precision was observed to be perfect, with no false positives or wrong classification and the system also exhibited excellent accuracy, correctly detecting the placed objects. As shown in Figure 6.2 (right), the RSU camera successfully detected the black car (ID=2) with a confidence score of 94% and the white car (ID=1) with a confidence score of 99%, ensuring robust and accurate object detection.

In comparison, the dynamic nature of the OBU camera introduced occasional challenges. These included missing one of the vehicles or producing lower confidence scores when detecting objects at oblique angles or during rapid movements. As a result, slight fluctuations in accuracy and confidence scores were observed for the OBU camera. For instance, as illustrated in Figure 6.2 (left), the OBU camera detected only one object with a lower confidence score of 74% as compared to RSU due to its dynamic motion.

Table 6.1.: Performance metrics for YOLOv3 Object Detection using RSU and OBU Cameras.

Camera View	Average Precision (%)	Average Accuracy (%)
Top Camera	100.0	37.0
CAVIL Camera	100.0	91.0

6.2. Feature Matching and Transformation

In figure 6.3 some misalignment were observed in cases where object features were less distinct, leading to false mismatches. These mismatches occurred because features were matched indiscriminately across the entire image without considering bounding boxes. This lack of specificity resulted in errors that hindered accurate homography computation and the reliable overlay of bounding boxes. In contract figure 6.4 shows a more refined technique by emphasizing feature matching inside comparable bounding boxes. Isolating features inside bounding boxes reduces the incidence of false matches, allowing for more precise tracking and alignment. This focused strategy guarantees that only the characteristics of objects that are relevant are matched, resulting in a strong correlation between OBU and RSU cameras perspectives.

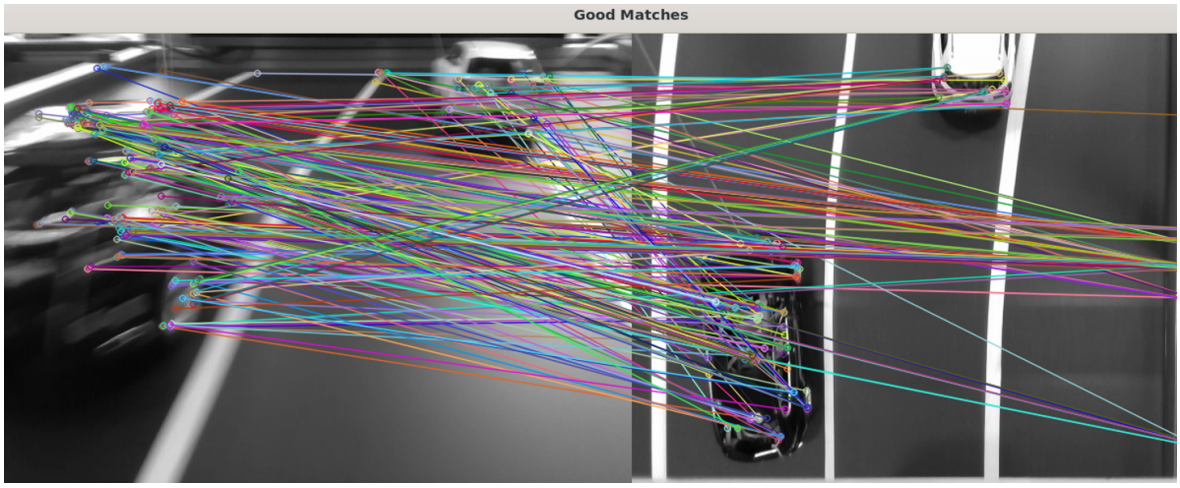

Figure 6.3.: Feature Matching Across Entire Frame

The feature matching technique successfully transformed and aligned 90% of the bounding boxes between the OBU camera view and the RSU camera view by matching corresponding features based on color and texture, as illustrated in figure 6.6. This feature matching ensured that bounding boxes from the OBU view were accurately projected onto the RSU view, aligning precisely with the objects positions. Additionally, it highlights how this approach effectively reduces errors in object correspondence by overlaying the bounding

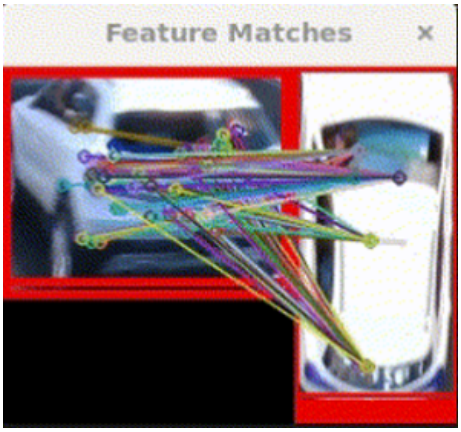

Figure 6.4.: Feature Matching Inside Corresponding Bounding Boxes

boxes in their exact positions, even when objects were only partially visible in one of the views.

The overlapping accuracy was validated using ground-truth data. The re-projection error was distributed over the range from 2 pixels to 7 pixels with an average of 4.3 pixels in transformed bounding box coordinates as shown in figure 6.5. This validates that the bounding box from the OBU is correctly mapped and aligned with its corresponding location in the RSU view. The low error number tells that the homography transformation is accurately aligning bounding boxes across two views.

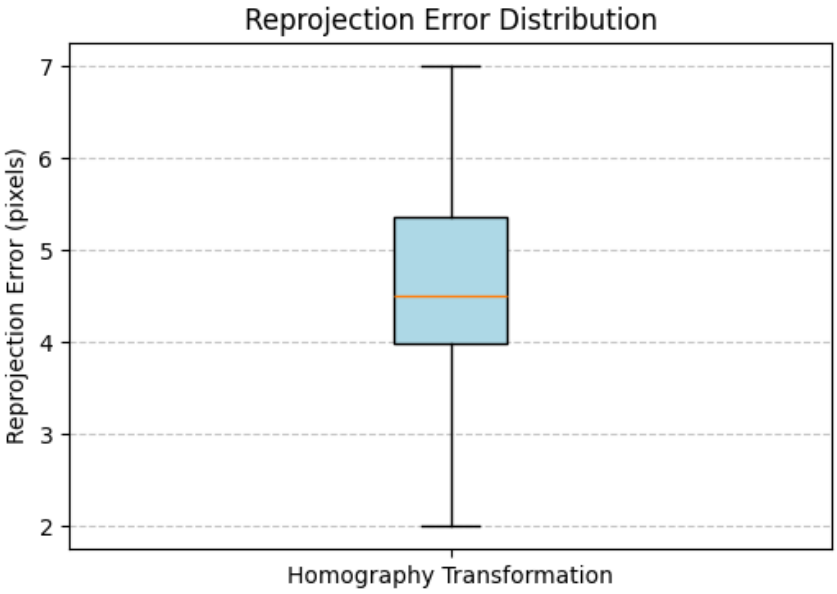

Figure 6.5.: Homography Re-projection Error

6.3. Fusion Process

The fusion process effectively combines bounding boxes from the OBU camera (transformed view) and the RSU camera view. As shown in Figure 6.6, the blue bounding box represents the final fused result, derived from two sources: the green bounding box from the transformed OBU view and the red bounding box from the RSU camera view. The fusion algorithm uses a weighting mechanism based on the confidence scores of each detection, ensuring that the fused bounding box leans closer to the source with the higher confidence score. This weighting approach enhances the accuracy and robustness of the fused detections.

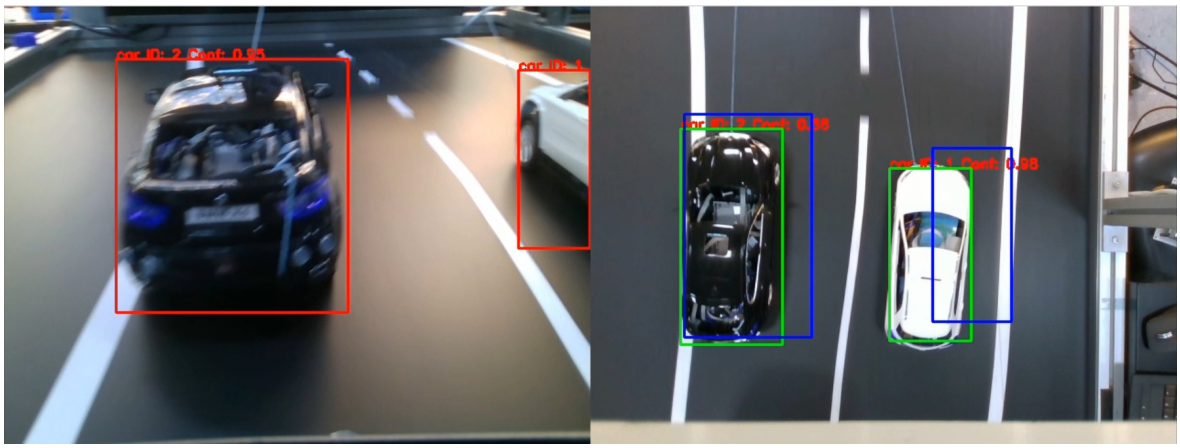

Figure 6.6.: Fused Bounding Box

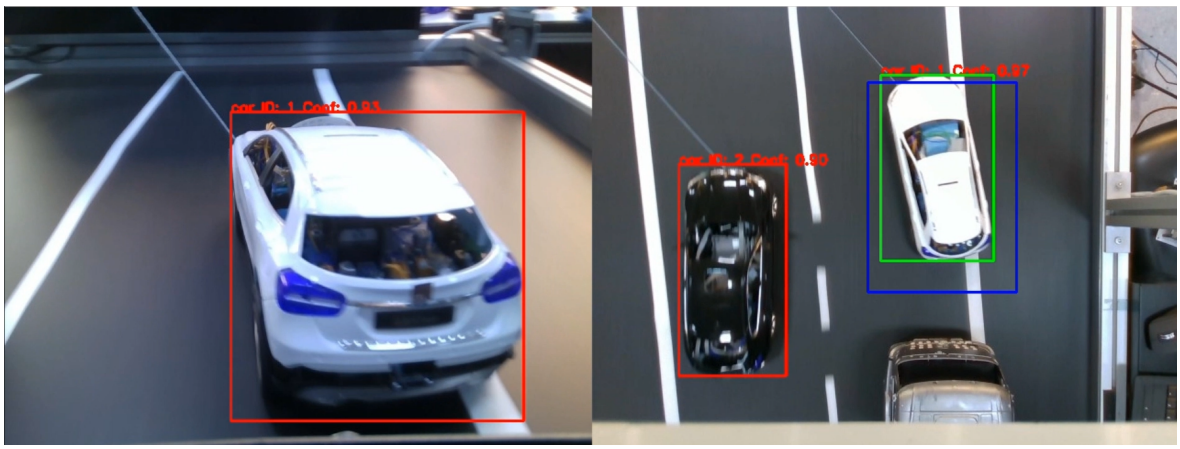

Figure 6.7.: Fused Bounding Box

In specific scenarios, such as when the OBU camera detects only one object while the RSU camera detects both as shown in figure 6.7, the system overlays the OBU detection onto the RSU camera view for fusion. For the second object, which is clearly detected

by the RSU camera, the system relies solely on top view without fusion. This ensures precise detection, even when one camera has incomplete information.

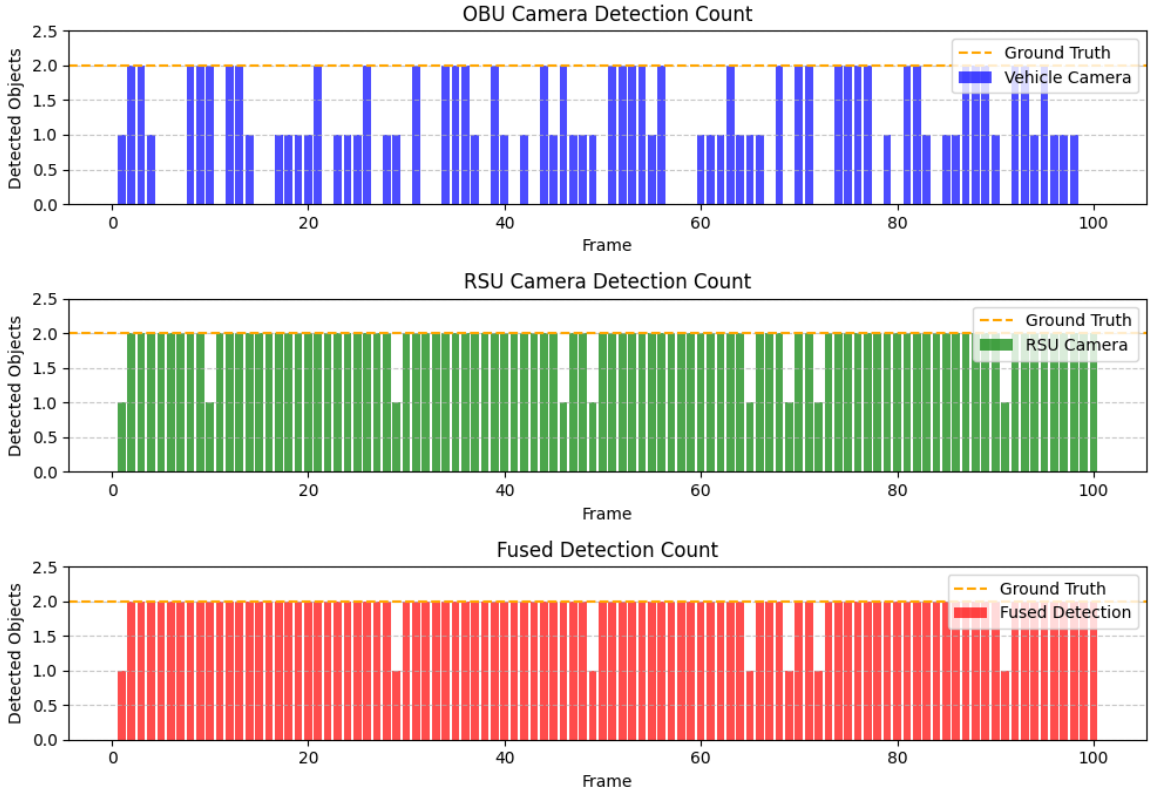

Figure 6.8.: Object Detection Results Across 100 Frames

The results 6.8 indicate that this adaptive fusion strategy achieves high reliability across various scenarios. As illustrated in figures 6.6 and 6.7 the fused bounding boxes are accurately aligned in most cases, delivering improved localization compared to utilizing a single camera independently. The bounding box fusion not only improved detection stability but also significantly enhanced accuracy over individual detections from each camera.

Figure 6.8 further highlights the disparity between the performance of the OBU and RSU cameras in isolation. The object detection accuracy of the OBU camera was limited to 37% due to its dynamic movement, leading to intermittent detection of one, both, or no objects. In contrast, the RSU camera achieved a much higher accuracy of 91%, benefiting from its static position, with both cameras maintaining a precision of 100% by correctly identifying and classifying the detected objects. By integrating data from both camera views, the fusion strategy successfully minimized errors, reduced false positives, and boosted overall detection accuracy to 93%, demonstrating a significant improvement in vehicle detection from 37% to 93%.

Furthermore, weighted averaging based on detection confidence ensured that the fused bounding boxes consistently represented the objects across both perspectives. This approach effectively handled variations in detection confidence and ensured accurate alignment of the bounding boxes. The fusion method also proved robust against moderate occlusions and dynamic bounding box movements, maintaining an average IoU of 0.72. These results underscore the potential of the adaptive fusion strategy to deliver reliable performance even in challenging scenarios, making it a practical solution for enhancing object detection in real world applications.

7. Conclusion and Future Work

7.1. Conclusion

This master's thesis presented a framework for multi-camera data fusion aimed at enhancing automated driving functionalities. The system integrates data from a moving OBU camera and a fixed RSU camera providing a birds eye view of the same scene. The proposed architecture utilizes mid-level fusion, also known as feature-level fusion, which processes extracted features instead of raw sensor data or final decision outputs. This approach minimizes the transmitted data volume, enabling efficient communication without delays while preserving sufficient information for timely and reliable decision making.

The framework successfully detected objects with each camera, leveraging the complementary strengths of both perspectives. The RSU camera provided a broader, static perspective, making it ideal for monitoring large areas and capturing objects beyond the vehicle's immediate field of view. Conversely, the OBU camera offered a dynamic, close-up perspective, essential for detecting objects at closer range and tracking them during vehicle movement. To achieve seamless integration, the system employed homography transformation, a technique that maps one image onto another, to accurately align object bounding boxes from both cameras. This step enabled the fusion of data into a unified representation, enhancing the robustness and reliability of object detection by addressing challenges such as partial occlusions, dynamic camera perspectives, and visibility loss caused by vehicle movement. This method enhanced object detection robustness and reliability by addressing challenges such as dynamic camera perspectives, partial occlusions, and visibility loss caused by vehicle movement. Moreover, it ensured the system could operate effectively in diverse scenarios, providing consistent and accurate results even in challenging environments.

Overall the developed framework effectively demonstrates the potential of feature-level fusion for improving perception in automated driving. Integrating data from both OBU and RSU cameras addresses critical challenges such as data efficiency, real-time processing, and robust object detection and tracking. This work is a foundation for future advancements in cooperative perception systems, paving the way for safer and more intelligent automated vehicles.

7.2. Future Work

Building upon the contributions of this thesis, several promising directions for future research can significantly enhance the proposed framework's capabilities and extend its applicability in real-world scenarios. One such direction is the integration of multiple sensor modalities, such as LiDAR, Radar, and ultrasonic sensors, alongside the existing cameras. These sensors offer complementary data that can address specific limitations of vision-based systems. For example, LiDAR can provide accurate 3D spatial information, Radar can enhance object detection and tracking in adverse weather conditions, and ultrasonic sensors can improve short-range detection. Combining these modalities through multi-sensor fusion would create a more comprehensive perception system capable of operating robustly under diverse environmental conditions, inspiring a new era of intelligent transportation systems.

An advanced approach to fusion could involve adopting a multi-layer fusion architecture that combines mid-level and high-level fusion techniques. Mid-level fusion can continue to provide efficient feature integration, while high-level fusion can aggregate outputs from different sensor modalities to enhance decision-making. This hierarchical fusion structure would enable the system to process feature-level data while incorporating decision-level insights for more adaptive and context-aware behavior.

Future research could investigate the implementation of fused object tracking to improve object detection and tracking robustness. The system can identify objects even if they leave the camera's field of vision for a few frames. If available, this can be accomplished by combining data from many sources, such as temporal data from consecutive frames, motion prediction models, and extra sensory inputs. Kalman filters, deep learning-based trackers, and optical flow algorithms can forecast an object's trajectory and reposition it when it appears in the frame. This development would increase the system's stability in dynamic conditions and provide continuous object tracking, instilling confidence in its reliability.

Improving the communication infrastructure between OBU and RSU is another critical area for future work. Advanced communication protocols, such as V2X technologies, including V2I and V2N, can be explored for faster, more reliable, and low-latency data transmission. Employing 5G or emerging 6G technologies can ensure high-speed communication with minimal delays, even in high-traffic scenarios.

A. Appendix

Abbreviations

Abbreviation	Meaning
ADAS	Advanced driver assistance systems
ACC	Adaptive Cruise Control
AV	Autonomous vehicles
BFMatcher	Brute-Force matcher
BRIEF	Binary Robust Independent Elementary Feature
BSD	Blind spot detection
BSM	Basic Safety Messages
CAM	Cooperative Awareness Messages
CAVIL	Collaborative Automated Vehicle-in-the-Loop
CEPT	Conference of Postal and Telecommunications Administrations
CNN	Convolutional Neural Network
COCO	Common Objects in Context
DSRC	Dedicated short range communication
ECU	Electronic Control Unit
FAST	Features from Accelerated Segment Test
FCW	Forward Collision Warning
HOG	Histograms of Oriented Gradient
IoU	Intersection over union
LiDAR	Light Detection and Ranging
LDW	Lane Departure Warning
LKA	Lane keeping Assistance
NHTSA	National Highway Traffic Safety Administration
NMS	Non-Maximum Suppression
OBU	Onboard Unit
ORB	Oriented FAST and Rotated BRIEF
PMO	Perception memory object
Radar	Radio Detection and Ranging
RANSAC	Random Sample Consensus
ROIF	RSU-OBU Integration Framework
RSU	Roadside Unit
RPN	Region Proposal Networks
ROS2	Robot Operating System 2
SAE	Society of Automotive Engineers
SVD	Singular Value Decomposition

SoC	System-on-Chip
TCU	Telematics Control Unit
uRLLC	ultra-reliable low latency communication
V2X	Vehicle-to-Everything
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
YOLOv3	You Only Look Once, Version 3

Bibliography

[1] K. Othman. "Exploring the implications of autonomous vehicles: a comprehensive review". In: *Innov. Infrastruct. Solut.* 7.2 (2022). DOI: 10.1007/s41062-022-00763-6.

[2] The Guardian. *Tesla driver dies in first fatal crash while using autopilot mode*. Accessed: 2024-09-25. 2016. URL: https://www.theguardian.com/technology/2016/jun/30/tesla-autopilot-death-self-driving-car-elon-musk.

[3] S. Maki and A. Sage. *Self-driving Uber car kills Arizona woman crossing street*. (2018). URL: https://www.reuters.com/article/us-autos-selfdriving-uber/woman-dies-in-arizona-after-being-hit-by-uber-self-driving-car-idUSKBN1GV296.

[4] M. Bergen and E. Newcomer. *Uber halts autonomous car tests after fatal crash in arizona*. (2018). URL: https://www.bloomberg.com/news/articles/2018-03-19/uber-autonomous-car-involved-in-fatal-crash-in-arizona.

[5] Andrew Kahn. "Elon Musk thinks Tesla's approach to self-driving cars is better than Lidar". In: *The Verge* (2022). Accessed: 2024-10-03. URL: https://www.theverge.com/23776430/lidar-tesla-autonomous-cars-elon-musk-waymo.

[6] RGBSI. *Sensor Fusion in Autonomous Driving Systems Part 2: Sensors Explained*. Accessed: October 03, 2024. 2020. URL: https://blog.rgbsi.com/sensor-fusion-autonomous-driving-systems%20part-2.

[7] Car2X. *Wie Kommunikationstechnik Unfälle komplett verhindern könnte*. (2023). URL: https://www.adac.de/rund-ums-fahrzeug/ausstattung-technik-zubehoer/assistenzsysteme/car2x/.

[8] H. Fattah. *5G LTE Narrowband Internet of Things (NB-IoT)*. 1st. CRC Press, 2018. DOI: 10.1201/9780429455056.

[9] Man Zhou, Weina Niu, and Xin Liu. "Sensor Fusion for the Safety of Automated Driving Systems". In: *Sensors* 21.6 (2021). DOI: 10.3390/s21062140.

[10] Tong Zhang. "Sensor Fusion and Multi-Sensor Data Integration for Enhanced Perception in Autonomous Vehicles". In: *Science Times* (2024). Accessed: 2024-10-03. URL: https://www.sciencetimes.com/articles/50773/20240617/sensor-fusion-and-multi-sensor-data-integration-for-enhanced-perception-in-autonomous-vehicles.htm.

[11] European Commission. *2023 Figures Show Stalling Progress in Reducing Road Fatalities in Too Many Countries*. Accessed: 2024-09-25. URL: https://transport.ec.europa.eu/news-events/news/2023-figures-show-stalling-progress-reducing-road-fatalities-too-many-countries-2024-03-08_en.

[12] R. M. Castelino et al. "Improving the Accuracy of Pedestrian Detection in Partially Occluded or Obstructed Scenarios". In: *2020 10th International Conference on Advanced Computer Information Technologies (ACIT)*. 2020, pp. 834–838. DOI: 10.1109/ACIT49673.2020.9208877.

[13] T. Gasser. "Ergebnisse der Projektgruppe Automatisierung: Rechtsfolgen zunehmender Fahrzeugautomatisierung". In: *5. Tagung Fahrerassistenz*. Munich, Germany, May 2012. URL: https://www.bast.de/DE/Publikationen/Berichte/unterreihe-f/2013-2012/f83.html.

[14] Stephen P. Wood et al. "The Potential Regulatory Challenges of Increasingly Autonomous Motor Vehicles". In: *Santa Clara Law Review* 52.4 (2012), p. 1423. URL: https://digitalcommons.law.scu.edu/lawreview/vol52/iss4/9.

[15] SAE International. *Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems*. Tech. rep. SAE J 3016. SAE International, Jan. 2014. URL: https://www.sae.org/standards/content/j3016_201806/.

[16] SAE International. *SAE J3016™: Updated Levels of Driving Automation*. Accessed: 2024-09-25. URL: https://www.sae.org/blog/sae-j3016-update.

[17] Synopsys. *What are the 6 Levels of Autonomous Driving?* Accessed: 2024-09-25. URL: https://www.synopsys.com/blogs/chip-design/autonomous-driving-levels.html.

[18] Scott Drew Pendleton et al. "Perception, Planning, Control, and Coordination for Autonomous Vehicles". In: *Machines* 5.1 (2017). ISSN: 2075-1702. DOI: 10.3390/machines5010006. URL: https://www.mdpi.com/2075-1702/5/1/6.

[19] S. Campbell et al. "Sensor Technology in Autonomous Vehicles: A review". In: *2018 29th Irish Signals and Systems Conference (ISSC)*. 2018. DOI: 10.1109/ISSC.2018.8585340.

[20] F. Camara et al. "Pedestrian Models for Autonomous Driving Part I: Low-Level Models, From Sensing to Tracking". In: *IEEE Transactions on Intelligent Transportation Systems* 22.10 (Oct. 2021), pp. 6131–6151. DOI: 10.1109/TITS.2020.3006768.

[21] Itransition. *Autonomous Vehicle Sensors: Key Technologies, and Challenges*. https://www.itransition.com/blog/autonomous-vehicle-sensors. Accessed: 2024-10-03. 2022.

[22] Manish Sharma, Mayur Dhanaraj, Srivallabha Karnam, et al. "YOLOrs: Object Detection in Multimodal Remote Sensing Imagery". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 14 (2021), p. 1497. DOI: 10.1109/jstars.2020.3041316.

[23] D. Ballabio, R. Todeschini, and V. Consonni. "Recent Advances in High-Level Fusion Methods to Classify Multiple Analytical Chemical Data". In: *Data Handling in Science and Technology*. Ed. by Marina Cocchi. Vol. 31. Elsevier, 2019, pp. 129–155. ISBN: 9780444639844. DOI: 10.1016/B978-0-444-63984-4.00005-3. URL: https://www.sciencedirect.com/science/article/pii/B9780444639844000053.

[24] P. Ghamisi et al. "Multisource and Multitemporal Data Fusion in Remote Sensing: A Comprehensive Review of the State of the Art". In: *IEEE Geoscience and Remote Sensing Magazine* 7.1 (Mar. 2019), pp. 6–39. DOI: 10.1109/MGRS.2018.2890023.

[25] Essam Debie et al. "Multimodal Fusion for Objective Assessment of Cognitive Workload: A Review". In: *IEEE Transactions on Cybernetics* 51.3 (2021), pp. 1542–1555. DOI: 10.1109/TCYB.2019.2939399.

[26] D.J. Yeong et al. "Sensor and Sensor Fusion Technology in Autonomous Vehicles: A Review". In: *Sensors* 21.6 (2021), p. 2140. DOI: 10.3390/s21062140. URL: https://doi.org/10.3390/s21062140.

[27] K. Naab. "Sensorik- und Signalverarbeitungsarchitekturen für Fahrerassistenz und Aktive Sicherheit". In: *Aktive Sicherheit durch Fahrerassistenz*. Garching, Germany, Mar. 2004.

[28] Ullrich Scheunert et al. "Early and Multi Level Fusion for Reliable Automotive Safety Systems". In: *2007 IEEE Intelligent Vehicles Symposium*. 2007, pp. 196–201. DOI: 10.1109/IVS.2007.4290114.

[29] MathWorks. *Getting Started with YOLO v3*. URL: https://de.mathworks.com/help/vision/ug/getting-started-with-yolo-v3.html.

[30] Qichao Mao et al. "Mini-YOLOv3: Real-Time Object Detector for Embedded Applications". In: *IEEE Access* 7 (2019), pp. 133529–133538. URL: https://api.semanticscholar.org/CorpusID:203567178.

[31] Jason Nataprawira et al. "Pedestrian Detection Using Multispectral Images and a Deep Neural Network". In: *Sensors* 21 (Apr. 2021), p. 2536. DOI: 10.3390/s21072536.

[32] Adrian Rosebrock. *Intersection over Union (IoU) for Object Detection*. Accessed: 2024-11-11. 2016. URL: https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/.

[33] Lin Zheng Chun et al. "YOLOv3: Face Detection in Complex Environments". In: *International Journal of Computational Intelligence Systems* 13.1 (2020). Received 20 June 2020, Accepted 28 July 2020, Available Online 17 August 2020., pp. 1153–1160. DOI: 10.2991/ijcis.d.200805.002.

[34] A. Neubeck and L. Van Gool. "Efficient Non-Maximum Suppression". In: *Pattern Recognition Letters* 29.3 (2006), pp. 356–360. DOI: 10.1016/j.patrec.2007.02.003.

[35] Hua Zhang and Hongbo Lu. "Evaluation Metrics for Object Detection and Their Relationship". In: *Journal of Computer Science and Technology* 35.4 (2020), pp. 763–775. DOI: 10.1007/s11390-020-9612-1.

[36] OpenCV Documentation. *Homography Transformations Tutorial*. Accessed: 2024-10-26. URL: https://docs.opencv.org/4.x/d9/dab/tutorial_homography.html#lecture_16.

[37] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. 2nd. Cambridge, UK: Cambridge University Press, 2004.

[38] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. 2nd. Cambridge University Press, 2003. ISBN: 978-0521540513.

[39] Eric Rublee et al. "ORB: An Efficient Alternative to SIFT or SURF". In: *IEEE International Conference on Computer Vision*. IEEE. 2011, pp. 2564–2571.

[40] Yinggang Xie et al. "Fast Target Recognition Based on Improved ORB Feature". In: *Applied Sciences* 12.2 (2022). ISSN: 2076-3417. DOI: 10.3390/app12020786. URL: https://www.mdpi.com/2076-3417/12/2/786.

[41] HAAS Alert. *V2X Explainer: What You Need to Know About Vehicle-to-Everything Technology*. Accessed: October 3, 2024. 2021. URL: https://www.haasalert.com/blog/v2x-explainer.

[42] Khadige Abboud, Hassan Aboubakr Omar, and Weihua Zhuang. "Interworking of DSRC and Cellular Network Technologies for V2X Communications: A Survey". In: *IEEE Transactions on Vehicular Technology* 65.12 (2016), pp. 9457–9470. DOI: 10.1109/TVT.2016.2591558.

[43] Business Wire. *Global Vehicle-to-Everything (V2X) Communication System Market 2019-2023*. Accessed: 2024-10-05. 2019. URL: https://www.businesswire.com/news/home/20190702005425/en/Global-Vehicle-to-Everything-V2X-Communication-System-Market-2019-2023.

[44] J. B. Kenney. "Dedicated short-range communications (DSRC) standards in the United States". In: *Proceedings of the IEEE* 99 (2011), pp. 1162–1182. DOI: 10.1109/JPROC.2011.2132790.

[45] Sandhiya Reddy Govindarajulu and Elias A. Alwan. "Range Optimization for DSRC and 5G Millimeter-Wave Vehicle-to-Vehicle Communication Link". In: *2019 International Workshop on Antenna Technology (iWAT)*. 2019, pp. 228–230. DOI: 10.1109/IWAT.2019.8730597.

[46] Shanzhi Chen et al. "LTE-V: A TD-LTE-based V2X solution for future vehicular network". In: *IEEE Internet of Things Journal* 3.6 (Dec. 2016), pp. 997–1005.

[47] Sohan Gyawali et al. "Challenges and Solutions for Cellular Based V2X Communications". In: *IEEE Communications Surveys & Tutorials* 23.1 (2021), pp. 222–255. DOI: 10.1109/COMST.2020.3029723.

[48] H. Takizawa, K. Yamada, and T. Ito. "Vehicles detection using sensor fusion". In: *IEEE Intelligent Vehicles Symposium, 2004*. 2004, pp. 238–243. DOI: 10.1109/IVS.2004.1336388.

[49] Yue Li et al. "Feature level sensor fusion for target detection in dynamic environments". In: *Proceedings of the American Control Conference* 2015 (July 2015), pp. 2433–2438. DOI: 10.1109/ACC.2015.7171097.

[50] M. Angelo et al. *Property-based Hierarchical Clustering of Peers using Mobile Agent for Unstructured P2P Systems*. 2016.

[51] R. Mobus and U. Kolbe. "Multi-target multi-object tracking, sensor fusion of radar and infrared". In: *IEEE Intelligent Vehicles Symposium, 2004*. 2004, pp. 732–737. DOI: 10.1109/IVS.2004.1336475.

[52] R. Labayrade, C. Royere, and D. Aubert. "A collision mitigation system using laser scanner and stereovision fusion and its assessment". In: *IEEE Proceedings. Intelligent Vehicles Symposium, 2005.* 2005, pp. 441–446. DOI: 10.1109/IVS.2005.1505143.

[53] Mao Shan et al. "A Novel Probabilistic V2X Data Fusion Framework for Cooperative Perception". In: *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. 2022, pp. 2013–2020. DOI: 10.1109/ITSC55140.2022.9922251.

[54] Jingkang Li, Hanhan Wang, Xiaodan Xu, et al. "End-to-End Autonomous Driving through V2X Cooperation". In: *arXiv preprint arXiv:2404.00717* (2024). URL: https://ar5iv.org/html/2404.00717.

[55] Han QIU et al. "An efficient key distribution system for data fusion in V2X heterogeneous networks". In: *Information Fusion* 50 (2019), pp. 212–220. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2019.02.002. URL: https://www.sciencedirect.com/science/article/pii/S1566253518306948.

[56] Navneet Dalal and Bill Triggs. "Histograms of oriented gradients for human detection". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. IEEE. 2005, pp. 886–893.

[57] Joseph Redmon et al. "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.

[58] Shaoqing Ren et al. "Faster R-CNN: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems* 28 (2015), pp. 91–99.

[59] Luca Caltagirone et al. "Lidar-camera fusion for road detection using fully convolutional neural networks". In: *Robotics and Autonomous Systems* 111 (2019), pp. 125–131.

[60] Wei Zhang, Tao Li, and Xiaoming Chen. “Distributed sensor fusion for autonomous driving: Road-side unit and on-board unit collaboration”. In: *IEEE Transactions on Intelligent Transportation Systems* (2022).

[61] Christian Forster, Massimiliano Pizzoli, and Davide Scaramuzza. “SVO: Fast Semi-Direct Monocular Visual Odometry”. In: *IEEE International Conference on Robotics and Automation (ICRA 2014)*. IEEE, 2014, pp. 15–22. DOI: 10.1109/ICRA.2014.6907067.

[62] A. Bhawiyuga et al. “A Wi-Fi based Electronic Road Sign for Enhancing the Awareness of Vehicle Driver”. In: *Journal of Physics: Conference Series* 801.1 (2017), p. 012085. DOI: 10.1088/1742-6596/801/1/012085.

[63] Jinuk Park et al. “Deep Learning-Based Stopped Vehicle Detection Method Utilizing In-Vehicle Dashcams”. In: *Electronics* 13.20 (2024). ISSN: 2079-9292. DOI: 10.3390/electronics13204097. URL: https://www.mdpi.com/2079-9292/13/20/4097.

[64] Roman Solovyev, Weijie Wang, and Tatiana Gabruseva. “Weighted Boxes Fusion: Ensembling Boxes for Object Detection Models”. In: *arXiv preprint arXiv:1910.13302* (2019). URL: https://arxiv.org/abs/1910.13302.

[65] O. Goczol. “Realtime Edge-Discovery for Transient Interoperability Between Vehicles and IoT-Devices; Echtzeitfähige Edge-Discovery zur Nutzung einer flüchtigen Interoperabilität zwischen Fahrzeugen und IoT-Endgeräten”. Master’s thesis. Karlsruhe Institute of Technology, 2024.

[66] “Real Time Object Detection Using YOLOv4”. In: *International Journal for Research in Applied Science and Engineering Technology (IJRASET)* (). URL: https://www.ijraset.com/research-paper/real-time-object-detection-using-yolov4.

[67] OpenCV contributors. *OpenCV: BFMatcher*. https://docs.opencv.org/4.x/d5/d6f/tutorial_feature_flann_matcher.html. 2024.

[68] Martin A Fischler and Robert C Bolles. “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”. In: *Communications of the ACM*. Vol. 24. 6. ACM, 1981, pp. 381–395.

List of Tables

List of Figures