Robust & Reliable Sensing & Perception Systems

Thesis Report

University Supervisor: Michiel Klifman

Company Supervisor: Dr. Eng. Alexandru Forrai

Hassan Hotait

June 28, 2023

Preface

I am honored to present this thesis report, which represents the culmination of my research efforts. I am immensely grateful to Dr. Eng. Alexandru Forrai for his invaluable guidance and support throughout this academic endeavor. Dr. Forrai's expertise, dedication, and mentorship have played a pivotal role in shaping my understanding of the subject matter.

I extend my heartfelt appreciation to Dr. Forrai for his unwavering support which have fostered a collaborative environment conducive to intellectual growth. I would also like to thank Mr. Michiel Klifman and Mr. Frank Berndsen for their patience and support.
This thesis report is the result of extensive research and reflection, and I hope it contributes to the existing knowledge in the field of perception systems. I am grateful for the opportunity to have worked under Dr. Forrai's mentorship, and I am confident that his influence will have a lasting impact on my academic and professional journey.

Contents

Term Definitions

SMOKE Single Monocular 3D Object Detection Via Key-point estimation. 3D Detection/Classification algorithm with Pose Estimation abilities.

YOLOv3 (You Only Look Once, Version 3) is a real-time object detection algorithm that identifies specific objects. The YOLO machine learning algorithm uses features learned by a deep convolutional neural network to detect an object.

DSGN2 Deep Stereo Geometry Network, which is a currently state-of-the-art stereo based algorithm for 3D object detection.
Ground-truth information that is known to be real or true, provided by direct observation and measurement as opposed to information provided by inference.

KITTI Vision Benchmark Suite, A project of Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago. Datasets are captured by driving around the mid-size city of Karlsruhe, in rural areas and on highways. Up to 15 cars and 30 pedestrians are visible per image. Besides providing all data in raw format, benchmarks are extracted, and an evaluation metric is provided for tasks such as stereo, optical flow, visual odometry, 3D object detection and 3D tracking.

SimCenter Prescan physics-based simulation platform by Siemens used in the automotive industry for development of Advanced Driver Assistance Systems (ADAS) that are based on sensor technologies such as radar, laser/LiDAR, camera, and GPS. Considering physics-based sensor models and urban traffic related scenarios as well as Simcenter Amesim, considering a 15DOF vehicle dynamics model.

AP Average-Precision metric used for evaluation of object detectors and other ML applications.

COCO dataset, meaning Common Objects In Context, is a set of challenging, high quality datasets for computer vision, mostly state-of-the-art neural networks. This name is also used to name a format used by those datasets.

DNN stands for Deep Neural Network. It is a type of artificial neural network composed of multiple layers of interconnected nodes, or neurons. DNNs are typically used for tasks such as pattern recognition, image and speech recognition, natural language processing, and many other machine learning tasks. They are designed to automatically learn hierarchical representations of data through a process known as deep learning.

SimCenter Amesim is a software tool developed by Siemens Digital Industries Software for system-level simulation. It is used for modeling and simulating the behavior of complex engineering systems, such as automotive powertrains, aircraft systems, and industrial processes. SimCenter Amesim offers a wide range of libraries and components to model physical domains like mechanical, electrical, thermal, and fluid systems, allowing engineers to analyze the dynamic behavior and performance of these systems.

1 Management Summary

This report presents a comprehensive study of robust reliable sensing and perception Systems, focusing on their design, verification, and validation in computer simulation. One objective of this study is to provide a framework in which these systems can be evaluated. This framework encompasses various metrics and methodologies to assess the performance and reliability of these systems creating a base for future certification and is integrated into SimCenter Prescan.

The report begins by highlighting the importance of robust perception systems for regulation. These systems play a crucial role in maintaining safe autonomous driving. By leveraging advanced sensing technologies and intelligent algorithms, robust perception systems enable machines to perceive and interpret their surroundings accurately and act appropriately even in harsh conditions.

To evaluate the robustness of perception systems, an assessment framework is proposed. It provides a systematic approach to identify strengths, weaknesses, and areas for improvement, ultimately leading to the development of more robust and reliable systems. The framework allows verification of requirements from the EU guidelines on robust and trustworthy AI such as indiscrimination and technical robustness.

These investigations showed that YOLOv3 did not demonstrate any discriminatory behavior. Since the race of the pedestrian did not affect the performance metrics. However, the size of the object did influence the performance. Smaller pedestrians such as children were more vulnerable as they were less likely to be detected.

In addition to that, the technical robustness investigation resulted in another intuitive conclusion. As the uncertainty due to perturbations increased, the performance of the perception system deteriorated. Even though this expected, its useful in determining the operational design domain of the system. In other words, in what range of conditions and specifications does the perception system satisfy the performance requirements

Perception systems which are aware of spatial information were researched and designed. Autonomous vehicles need detailed spatial information for safety, information regarding the object pose and dimensions allows collision avoidance and path planning. As Yolov3 is insufficient for this, 3D object detection algorithms such as SMOKE and DSGN2 were investigated in addition to depth estimation via stereo vision. The implemented solutions (SMOKE and Stereo Vision) did not meet requirements for object localization accuracy. However, DSGN2 is recommended for implementation as there are strong indicators which makes it an ideal choice for the perception stack. DSNG2 is a stereo vision-based algorithm which is the highest scoring method on the benchmarks for 3D Object Detection. The code is also open source. By doing so its possible to gain a competitive edge in the industry. This management summary provides a high-level overview of the reports key findings and recommenda-

tions. For a more detailed understanding, its encouraged to read the full report.

2 Introduction

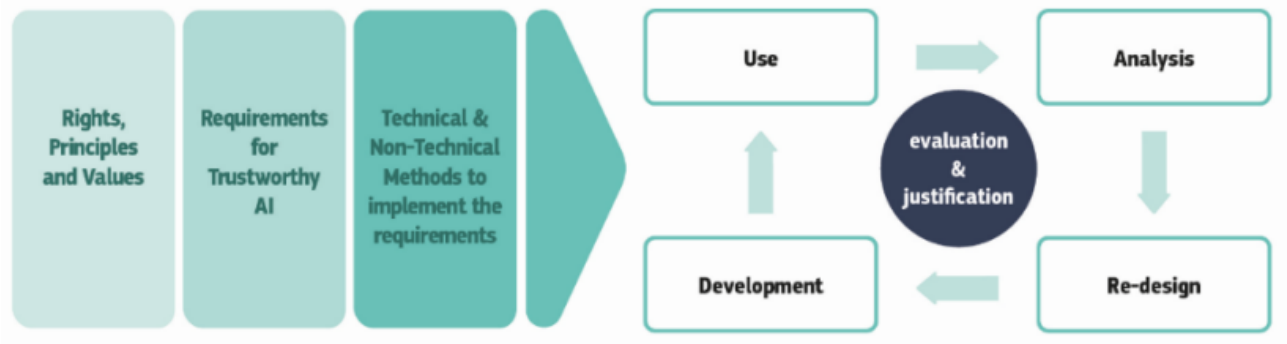


Figure 1: Realization Of Trustworthy and Reliable AI

The project aims to develop methodologies for evaluation and robustness assessments of perception systems. Perception systems refer to the sensors and algorithms that enable a vehicle to perceive and understand its surrounding environment. These systems are responsible for gathering data from various sensors, processing that data to extract meaningful information about the surrounding objects, road conditions, and relevant traffic elements. Providing the vehicle with real-time awareness of its surroundings allowing it to make informed decisions and navigate safely.

Robustness in the context of this project refers to the ability of a system, such as a perception system, to maintain its functionality and performance even in the presence of unexpected or challenging conditions. It is an important factor in ensuring the safety and reliability of autonomous systems.

In this project we focus on camera-based perception systems. These systems use AI/Learning-enabled algorithms. Such algorithms acquire knowledge and enhance performance through learning from data. Models such as deep neural networks e.g., DNNS are used to analyze large datasets, identify patterns, and adapt their behavior accordingly. They undergo a training phase where they adjust their parameters to minimize errors and optimize performance. Learning-enabled systems aim to generalize their knowledge beyond the training data, enabling them to make accurate predictions or decisions in new situations.

It is a strong interest and market need for effective methods and efficient tools to verify and validate the safe behavior of perception systems in complex environments. As such methods play a key role in the development and certification process of ADAS and autonomous driving systems.

Robustness assessments are in line with currently applicable standards such as ISO21448. Testing the perception module against different kinds of uncertainties creates harsh conditions for sensors and algorithms. Therefore, we

investigate **how to assess the robustness of vision-based perception systems for self-driving applications?**

A systematic and unified approach is followed by adopting data formats and metrics used in the industry, creating a base for future standardization and certification. In this project, a robustness assessment framework for camera-based perception systems is proposed and implemented in SimCenter Prescan. SimCenter Prescan is a physics-based simulation platform by Siemens used in the automotive industry for development of Advanced Driver Assistance Systems (ADAS) that are based on sensor technologies such as radar, laser/LiDAR, camera, and GPS. Considering physics-based sensor models and urban traffic related scenarios as well as Simcenter Amesim (Vehicle Dynamics Simulator), considering a 15DOF vehicle dynamics model.

The algorithms used in the perception systems assessment are Yolov3 and SMOKE. This report relies heavily on the work done during my internship. The objects detectors investigated during the internship (Yolov3 and SMOKE) are now part of the perception stack.

Unlike the internship period where all investigations were done offline; recording raw data, then post processing it separately in python to get the KITTI labels, test object detector on images recorded, then perform evaluation based on predictions and labels. The current framework done during the thesis integrates the labels generations pipeline, perception stack and the evaluation into SimCenter Prescan. This accelerates validation and verification of the perception module.

Perception systems which are aware of spatial information were researched and designed. Autonomous vehicles need detailed spatial information for safety, information regarding the object pose and dimensions allows collision avoidance and path planning. As 2D object detectors are insufficient for this, 3D object detection algorithms such as SMOKE and DSGN2 were investigated in addition to depth estimation via stereo vision. There is a particular interest in, **what the practical challenges of implementing Stereo Vision in driving scenarios are. In addition to the different techniques for obtaining spatial information, whether monocular or stereo-based and how they compare in terms of performance.**

3 Robustness Assessment Framework

How to assess the robustness of 3D Object Detection for self-driving applications?

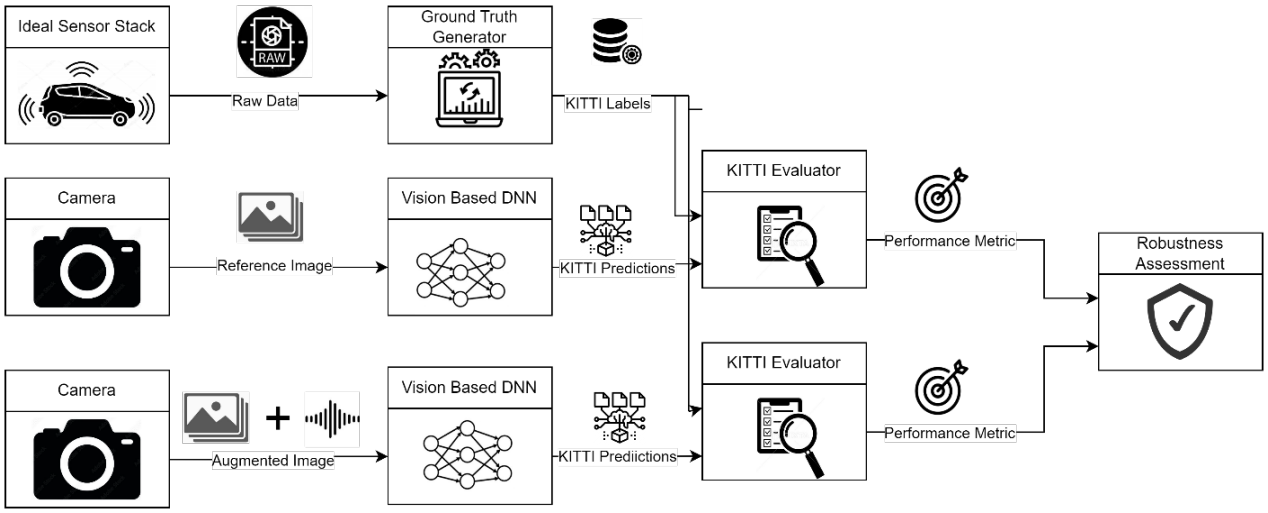


Figure 2: Robustness Framework

Let us assume that in case of camera-based sensing and perception system, misdetections are associated with a failure. The number of misdetections depend on the operating conditions of the sensors. Therefore, the reference image in Fig 2 shall be representative capturing a wide variety of operating conditions (e.g. illumination conditions, weather conditions, etc.).

Therefore, it is relevant to investigate the robustness of the system, e.g. how the number of failures are changing, when the sensing system is exposed to an augmented image shown in Fig 2. The reference image is augmented by aleatoric and parametric uncertainty.

Uncertainty refers to the degree of ambiguity or lack of information contained within an image. Image uncertainty can arise from various factors, including noise, occlusions, image quality, variations in lighting conditions, complex backgrounds, or inherent limitations of the imaging system. The difference between parametric and aleatoric uncertainties lies in the cause of uncertainty. Where uncertainty due to hardware faults is defined as aleatoric while other conditions such as lighting and fog are considered parametric uncertainties. Some examples are shown in the following sections.

Robustness assessment of camera-based perception systems requires the definition of a reference data set, an augmented data set, a distance metric to assess the similarity between the reference data set and augmented data set as well as a performance metric such as the precision of the network.

The relationship between the performance metric and uncertainty can be seen as robustness. Deterioration of performance metric as uncertainty increases represents low robustness. While constant performance as uncertainty in-

creases represents high robustness.

Synthetic data is generated in SimCenter Prescan. This allows the definition and parametrization of an autonomous driving test scenario in a flexible way. For vehicles, different models and colors can be selected, while for humans: gender, race and age can be chosen.

Furthermore, environmental and illumination conditions specific for the operational design domain can be easily specified. This parametrization allows us to generate an augmented dataset which includes the uncertainty shown in Fig 2. The test scenario with all its configurations is referred to as an experiment.

Data generated by a computer simulation can be seen as synthetic data. This encompasses most applications of physical modeling, such as music synthesizers or flight simulators. The output of such systems approximates the real thing but is fully algorithmically generated (Nowruzi et al., 2019) .

The data generated includes the images captured by the camera, labels generated by the ground truth labels generator subsystem and predictions generated and is used for robustness assessment of the perception system. Ground-truth is information that is known to be real or true, provided by direct observation and measurement as opposed to predictions provided by inference. In the framework both the labels and predictions are in KITTI format described in next section.

KITTI is a Computer Vision Benchmark Suite, A project of Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago to collect real world driving data. Besides providing all data in raw format, benchmarks are extracted, and an evaluation metric is provided for tasks such as stereo, optical flow, visual odometry, 3D object detection and 3D tracking. KITTI is also the term used to refer to the format in which the data is stored.

Here the importance of SimCenter Prescan is highlighted, it allows us to obtain the true information synthetically when real world data is difficult to obtain.

The subsystems of the framework shown in Fig 2 are described in detail in the next section to provide an insight on the labels, predictions, performance metric and uncertainty.

3.1 Ground Truth Labels Generator

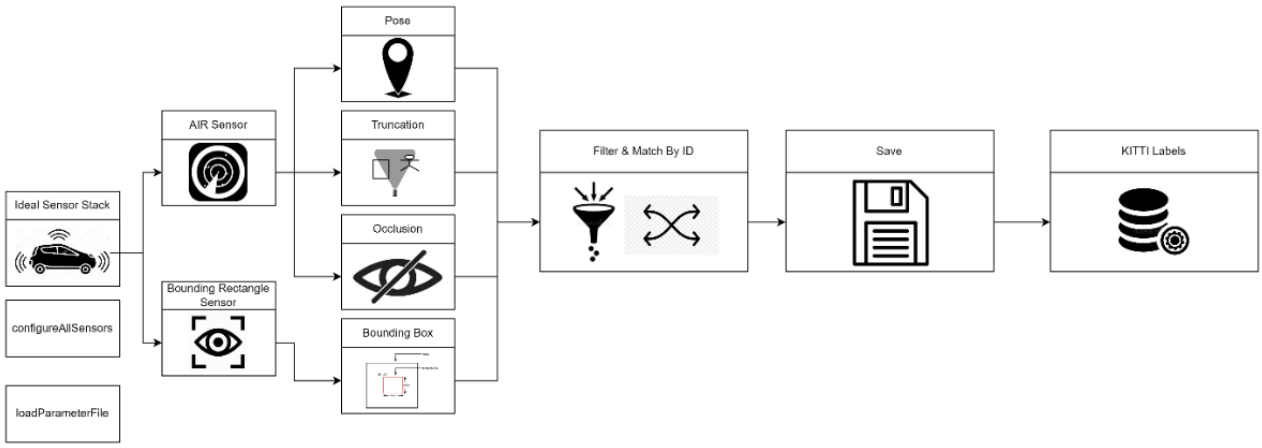


Figure 3: Ground Truth Labels Generator

The ground truth labels generator requires Prescan AIR Sensor (Actor Information Receiver) and BRS Sensor (Bounding Rectangle Sensor) referred to in the Figure 3 as Ideal Sensor Stack. These sensors obtain the ground truth or true information and are therefore referred to as Ideal.

Configuration of sensors in the ideal stack is critical to accurate labels generation. This configuration is handled by configureAllSensors which is executed before running the experiment. This would match the FOV (Field Of View) of both sensors and sets AIR sensor range. loadParameterFile is also executed before the experiment is run to load the experiment object dimensions.

```
#Values    Name      Description
----------------------------------------------------------------------------
   1    type         Describes the type of object: 'Car', 'Van', 'Truck',
                     'Pedestrian', 'Person_sitting', 'Cyclist', 'Tram',
                     'Misc' or 'DontCare'
   1    truncated    Float from 0 (non-truncated) to 1 (truncated), where
                     truncated refers to the object leaving image boundaries
   1    occluded     Integer (0,1,2,3) indicating occlusion state:
                     0 = fully visible, 1 = partly occluded
                     2 = largely occluded, 3 = unknown
   1    alpha        Observation angle of object, ranging [-pi..pi]
   4    bbox         2D bounding box of object in the image (0-based index):
                     contains left, top, right, bottom pixel coordinates
   3    dimensions   3D object dimensions: height, width, length (in meters)
   3    location     3D object location x,y,z in camera coordinates (in meters)
   1    rotation_y   Rotation ry around Y-axis in camera coordinates [-pi..pi]
   1    score        Only for results: Float, indicating confidence in
                     detection, needed for p/r curves, higher is better.
```

Figure 4: KITTI Data Format

3.1.1 Pose

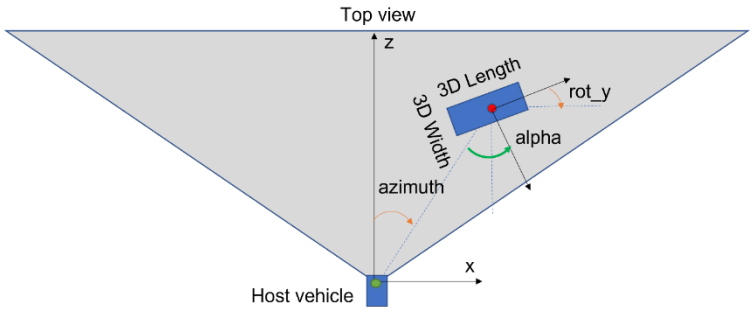


Figure 5: KITTI Coordinate System Convention

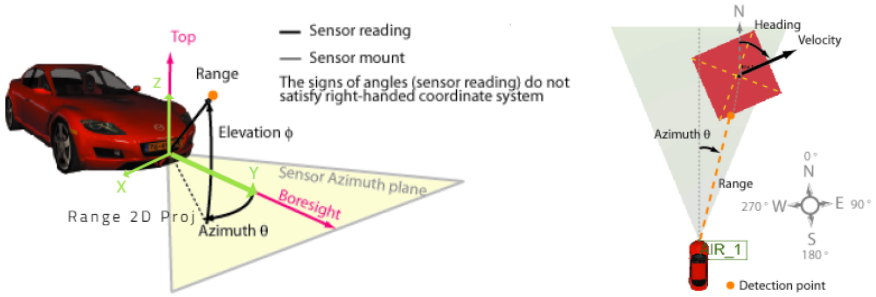


Figure 6: AIR Sensor Operating Principle

Algorithm 1: Pose

Input: sensorRange, sensorAzimuth , sensorElevation, sensorHeading
Output: X, Y, Z, Alpha, Rot
Procedure: *Use Prescan Actor Information Receiver Sensor to get position and orientation of detected objects.*

$xyProj = \text{sensorRange} \times \cos(\text{sensorElevation});$
$X \leftarrow xyProj \times \sin(\text{sensorAzimuth});$
$Y \leftarrow xyProj \times \cos(\text{sensorAzimuth});$
$Z \leftarrow \text{sensorRange} \times \sin(\text{sensorElevation});$
$Rot \leftarrow \text{sensorHeading};$
$Alpha \leftarrow \text{sensorAzimuth} + \text{sensorHeading};$

3.1.2 Bounding Box

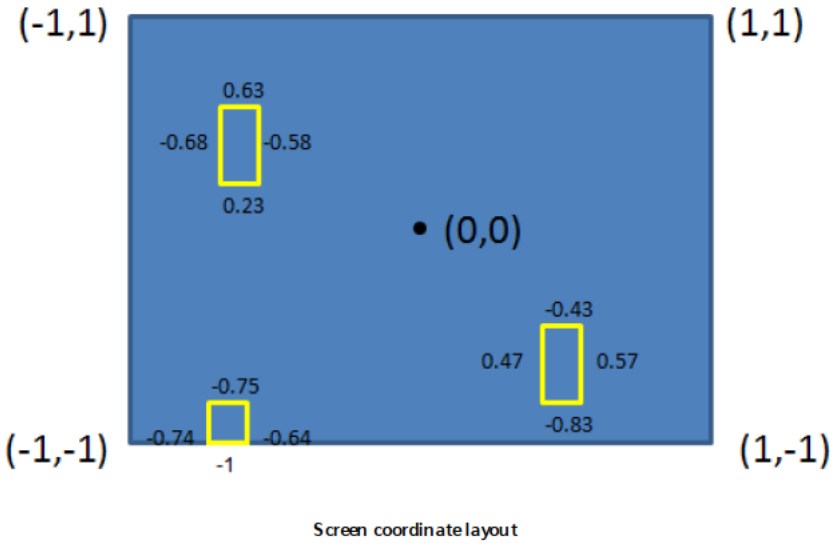


Figure 7: Screen Coordinate Layout

The bounding rectangle sensor provides bounding rectangle information for each Object ID: Left, Right, Bottom and Top coordinates are provided. If an actor is in view the coordinate values are displayed in values between -1 and 1. The coordinates are scaled to the image dimensions according to the algorithm described below.

Algorithm 1: Bounding Box

Input: leftIn , rightIn, topIn, bottomIn, imgWidth, imgHeight
Output: leftOut, rightOut, topOut, bottomOut
Procedure: *Scale the rectangle coordinates given between 0-1 from Prescan Bounding Rectangle Sensor to their respective image coordinates.*

$leftScaled = (leftIn + 1)/2$
$rightScaled = (rightIn + 1)/2$
$bottomScaled = (bottomIn + 1)/2$
$topScaled = (topIn + 1)/2$
$leftOut \leftarrow leftScaled \times$ imgWidth
$rightOut \leftarrow rightScaled \times$ imgWidth
$bottomOut \leftarrow (1 - bottomScaled) \times$ imgHeight
$topOut \leftarrow (1 - topScaled) \times$ imgHeight

3.1.3 Truncation

In object detection, truncation refers to objects that are only partially visible within the image frame, extending beyond the image boundaries or being cut off by the frame's edge.

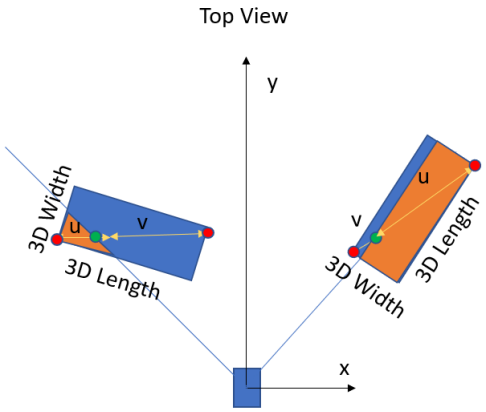


Figure 8: Truncation 2D Representation

Algorithm 1: Truncation

Input: X, Y, lengthObject,widthObject,rotGlobal, sensorFOV,sensorRange
Output: truncation
Procedure: *Get intersection point of rectangle representing object with sensor field of view. Use rectangle corner points to get the distances u and v. Truncation is the ratio of object within FOV over complete object.*

Get Rectangle Corner Points
N = Get Number of Corner Points within FOV

if $N = 0$ **then**
    $truncation \leftarrow 1$;

else if $N = 4$ **then**
    $truncation \leftarrow 0$;

else
    Get coordinates of diagnoal with vertex outside FOV.

    Find intersection point of FOV with diagonal.

    Given the coordinates, compute u and v.

    $truncation \leftarrow \frac{u}{u+v}$;
end

3.1.4 Occlusion

In object detection, occlusion occurs when objects of interest are partially or fully hidden by other objects or elements in the scene.

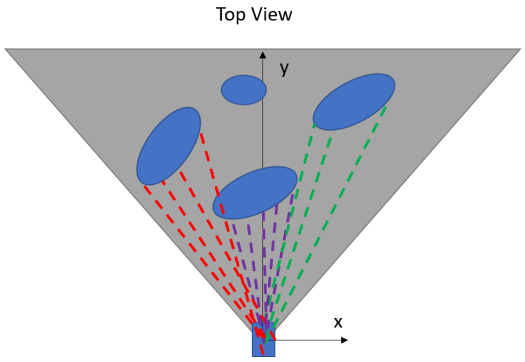


Figure 9: Occlusion 2D Representation

Algorithm 1 Occlusion

Input: X, Y, lengthObject,widthObject,rotGlobal, sensorFOV,sensorRange
Output: occlusion
Procedure: *Get the number of rays that hit the object with and without occlusion. The ratio and the thresholds are used to assign to occlusion level*

```
N = Number Of Rays
  n = 0
  m = 0

for i = 1 to N do
    // Perform operations within the loop instructions
    if Ray ∩ Object @ No Occlusion then
        m + =1
    end
    if Ray ∩ Object @ Occlusion then
        n + =1
    end
end

occlusionRatio = n/m
if occlusionRatio < 0.05 then
    occlusion ← 0
    ▷ Fully Visible

else if 0.05 < occlusionRatio < 0.5 then
    occlusion ← 1
    ▷ Partly Occluded

else if 0.5 < occlusionRatio < 0.75 then
    occlusion ← 2
    ▷ Largely Occluded

else
    occlusion ← 3
    ▷ Unkown
end
```

3.1.5 Limitations

The ground truth labels generator has some limitations when it comes to the accuracy of truncated occluded objects.

The detection type configuration of the AIR sensor is set to object 3D center. The detection point must be within the sensor FOV for the object to be detected. Thus, some truncated objects are not detected and hence the ground truth label for that object is not collected. Such a case is shown in Fig 10, where part of the box is in the FOV, but the center of the box is outside.

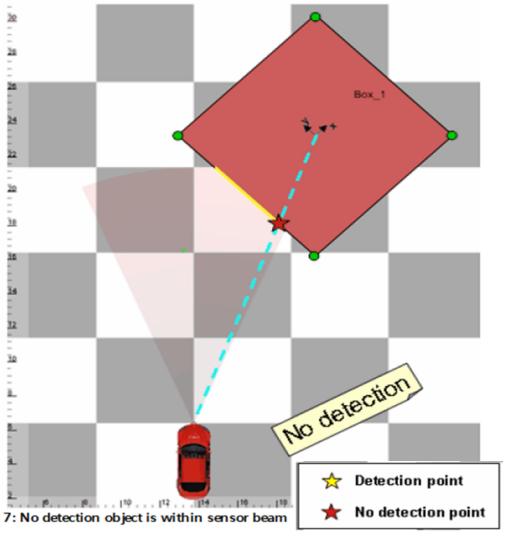


Figure 10: Effect of Detection Type on AIR Sensor behaviour

The algorithms for computing occlusion and truncation assume only a 2D representation of the object. It does not take in account occlusion and truncation in the height dimension. Also, the objects are represented as rectangles as shown in Fig 8 for truncation and ellipses as shown in Fig 9. These assumptions affect the accuracy of truncation and occlusion.

3.2 Vision Based DNN

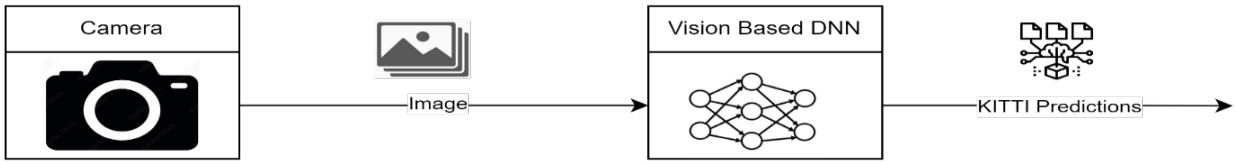


Figure 11: Vision Based DNN

Currently the vision-based perception stack includes:

- YOLOv3
- SMOKE

Perceptions algorithms are available in Python and are integrated into the experiment via Matlab Simulink. Each of the object detectors has a different implementation and therefore each has its own pros and cons. Some of the differences are shown below:

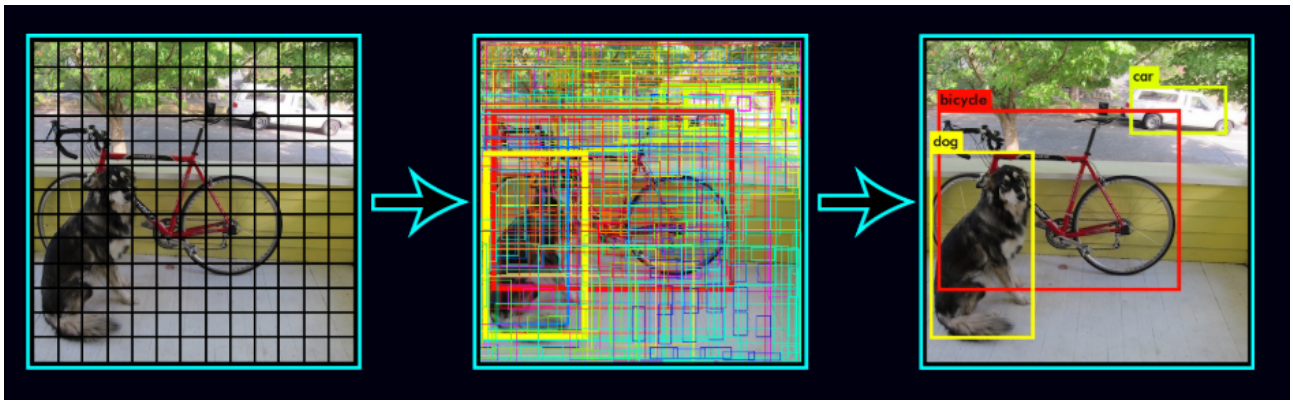


Figure 12: YOLOv3 Grid Based Approach

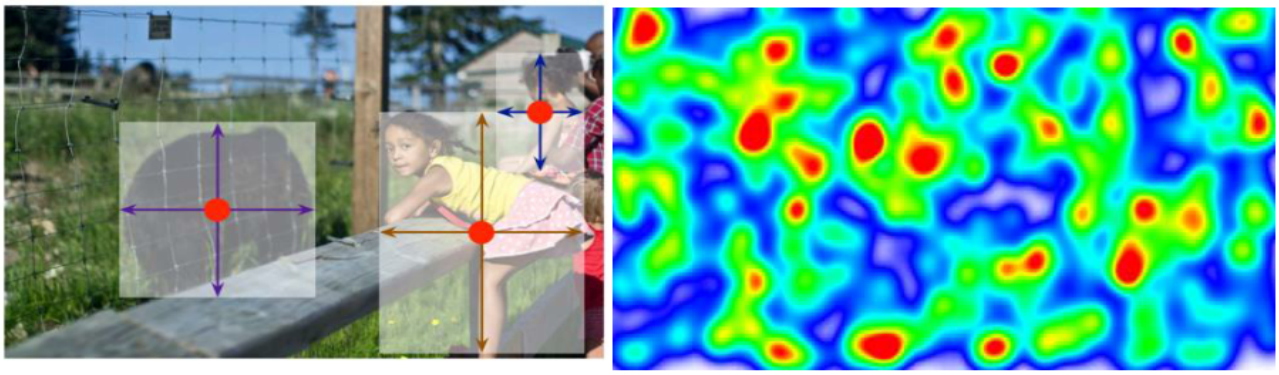


Figure 13: SMOKE Objects As Points Approach

Yolov3 is an object detection algorithm that operates in a single pass through an input image to detect and classify objects. It divides the image into a grid and predicts bounding boxes and class probabilities for each grid cell. Yolov3 uses anchor boxes of different sizes to handle objects of various scales. It also utilizes feature maps from different layers to detect objects at different scales and resolutions.

To improve accuracy, Yolov3 employs multiple scales during training and testing, allowing it to detect small and large objects effectively. It utilizes a Darknet-53 architecture, which is a deep convolutional neural network, for feature extraction.

Yolov3 uses a combination of bounding box regression and non-maximum suppression (NMS) to refine the predicted boxes and eliminate duplicate detections. It achieves impressive performance in terms of both accuracy and speed (Redmon and Farhadi, 2018).

2D Object detectors such as YOLOv3 are insufficient for autonomous driving. They lack information about object pose and dimensions. This limits collision avoidance and path planning capabilities. Autonomous vehicles need detailed spatial information for safety. Therefore, SMOKE was investigated to seek spatial information via 3D detection. Appendix X answers internship research question How SMOKE predicts object 3D Position?. Refer to it for the internal workings of SMOKE.

SMOKE is specifically designed for tasks related to autonomous driving. Therefore, it focuses on classes directly relevant to these scenarios, such as pedestrians, cars, and cyclists. While YOLOv3 model is trained on COCO (Common Objects in Context) dataset that contains a wide range of object classes commonly encountered in everyday scenes.

3.3 Performance Evaluator

As shown in the frameworks for evaluation and robustness in Fig 2, the performance metric is essential for evaluation. This describes how good or bad the perception system is behaving. Properties such as object difficulty and object neighboring class are used for a fairer evaluation are considered. In addition to that, common metrics used in the industry and KITTI benchmarks are extracted in the Performance Evaluation subsystem shown in Fig 2. To further investigate robustness, uncertainty is defined and then computed using the MATLAB computer vision toolbox.

3.3.1 Object Difficulty

Algorithm 1: difficulty

Input: imgHeight,boxHeight,truncation,occlusion
Output: difficulty
Procedure: *Categorize Objects based on the boxHeight, truncation and occlusion thresholds adopted by KITTI*
$occlusion = 0$ Fully Visible
$occlusion = 1$ Partly Occluded
$occlusion = 2$ Largely Occluded
$occlusion = 3$ Unknown

imgHeightKitti = 375

if $\frac{boxHeight}{imgHeight} >= \frac{40}{imgHeightKitti}$ *and truncation* $<= 0.15$ *and occlusion* $\in [0]$ **then**
$difficulty \leftarrow Easy;$

else if $\frac{boxHeight}{imgHeight} >= \frac{25}{imgHeightKitti}$ *and truncation* $<= 0.3$ *and occlusion* $\in [0, 1]$ **then**
$difficulty \leftarrow Moderate;$

else if $\frac{boxHeight}{imgHeight} >= \frac{25}{imgHeightKitti}$ *and truncation* $<= 0.5$ *and occlusion* $\in [0, 1, 2]$ **then**
$difficulty \leftarrow Hard;$

else
$difficulty \leftarrow Ignored;$
end

3.3.2 Object Neighbouring Class

Figure 14: YOLOv3 misclassifies car as a van

Neighbouring class is a concept introduced in the KITTI vision suite. It aims to contribute to a fairer evaluation of object detectors. Neighbouring classes are classes which are visually similar, such as cars and vans. It's likely object detectors will misclassify such objects as shown in Fig 14.

While it is important to distinguish between these classes, the impact of confusing a car with a van does not significantly affect the behavior of the self-driving vehicle. While both vehicles differ in size and shape, their functional behavior on the road is generally similar. Therefore, in the context of autonomous driving, the distinction between these classes may not have a substantial impact on path planning and decision making processes. Thus, the evaluation must not penalize for misclassifying neighboring classes.

3.3.3 Performance Metrics

Performance metrics play a crucial role in evaluating the effectiveness and accuracy of object detectors such as YOLOv3 and SMOKE. Object detection is a fundamental task in computer vision that involves identifying and localizing objects of interest within images or video frames. To assess the performance of object detectors, various metrics are utilized to quantify their detection capabilities and provide insights into their strengths and weaknesses.

3.3.4 Common Industry Metrics

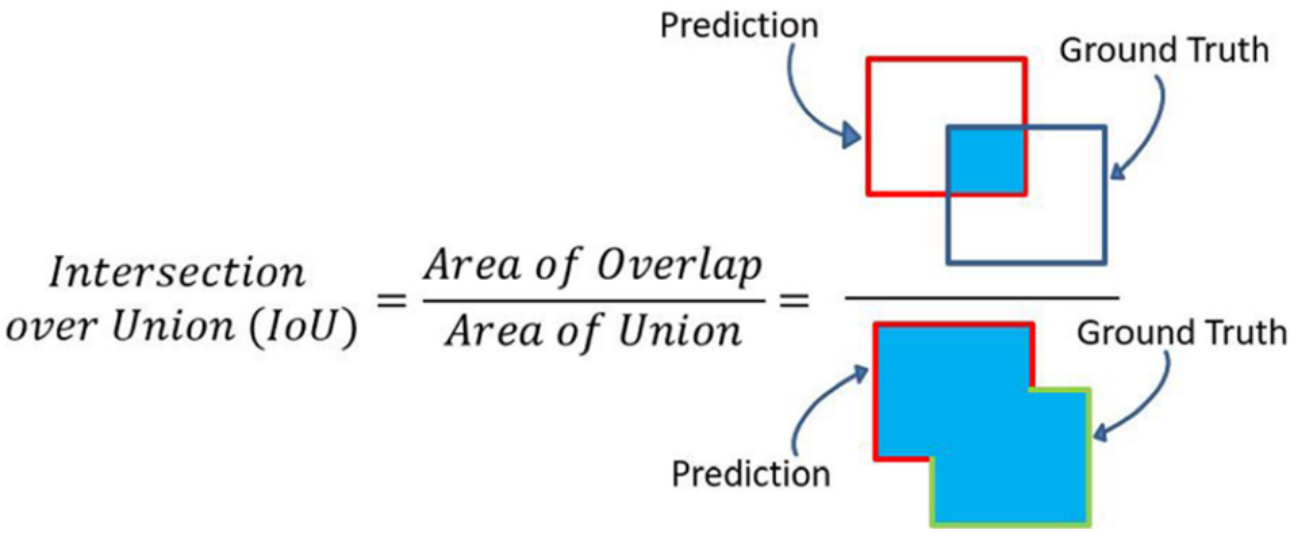


Figure 15: Intersection Over Union Metric

The IoU (Intersection Over Union) metric is widely used to evaluate object detectors. It measures the overlap between predicted and ground truth bounding boxes, indicating the alignment and localization accuracy. A higher IoU score signifies more accurate object detection, while a lower score suggests potential issues. IoU is employed with a threshold to classify detections as true positives or false positives, enabling the calculation of precision and recall. In the evaluation framework the IoU threshold defined is 50 percent. Overall, IoU serves as a standardized measure to compare different detectors, assess their detection quality, and make informed decisions about model selection. For the prediction to be considered a true or good prediction also referred to as True Positive it is not sufficient to solely satisfy the IoU threshold, the class of predicted objects must match with the ground truth as well.

2D Object detectors are also assessed based on their performance in terms of precision, recall, and overall accuracy shown in Fig 16. Precision helps to understand the detector's ability to minimize false positives, while recall reflects its capability to minimize false negatives. Balancing both precision and recall is crucial to achieving accurate and comprehensive object detection. Accuracy provides a general measure of the detector's correctness, considering both true positives and true negatives. Refer to Fig 16 for definitions of True Positives (TP), False Positives (FP), and False Negatives (FN).

Its worth noting that:

$$Accuracy(Acc) = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

However for object detection, TN = 0. Hence the altered formula in Fig 16.

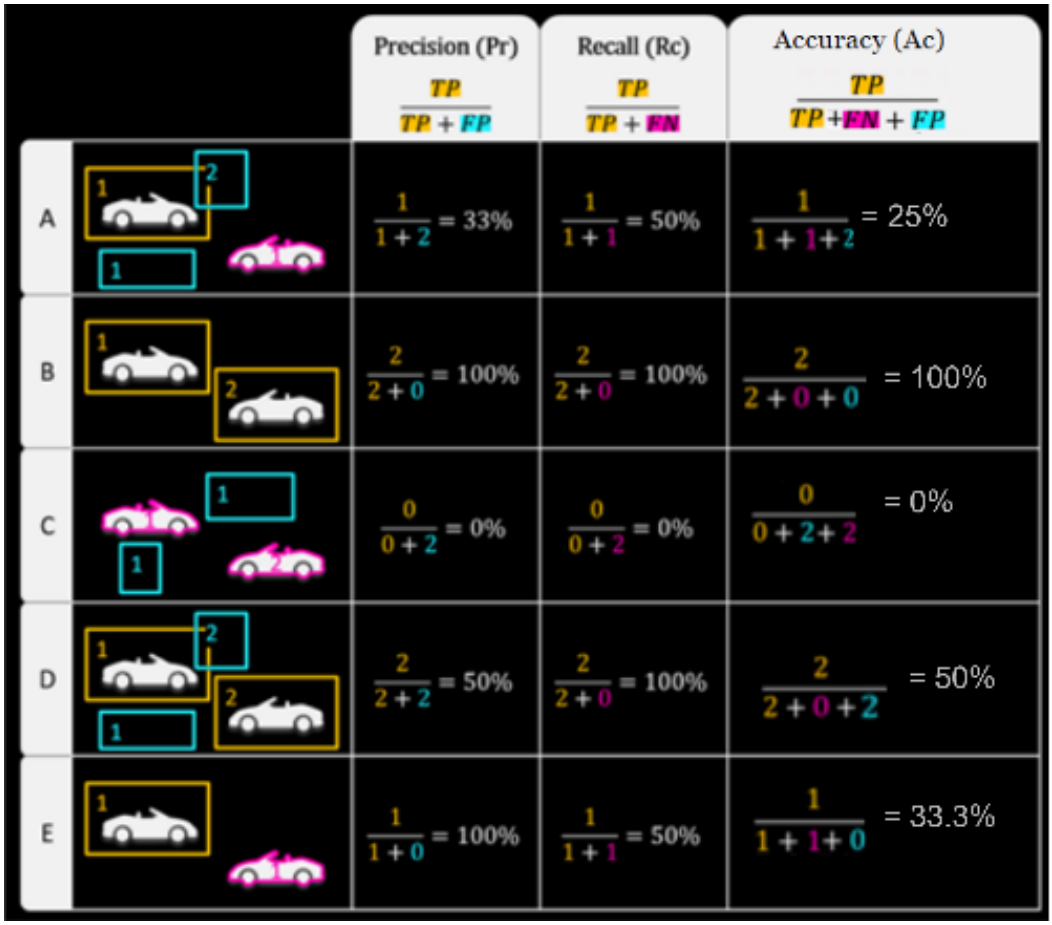


Figure 16: Common Metrics For Object Detection Evaluation

In addition to the precision, recall and accuracy metrics the average precision AP is also used to evaluate the performance of object detectors such as YOLOv3 and SMOKE. This metric was adopted by PASCAL VOC Challenge in 2007. The PASCAL Visual Object Classes (VOC) challenge is a benchmark in visual object category recognition and detection, providing the vision and machine learning communities with a standard dataset of images and annotation, and standard evaluation procedures. Organized annually from 2005 to present (2009), the challenge and its associated dataset has become accepted as the benchmark for object detection (Everingham et al., 2010).

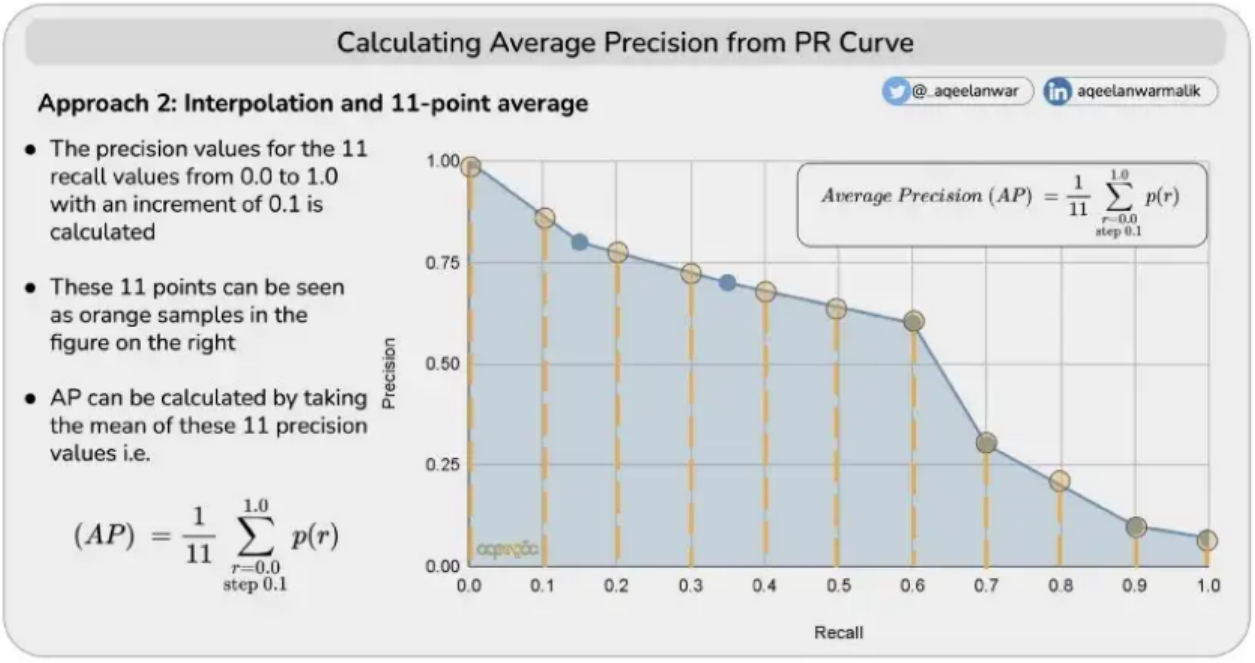


Figure 17: Average Precision from PR Curve; by interpolation and 11-point average

3.3.5 KITTI Object Detection Benchmarks

KITTI used the AP at 11 recall points described in the earlier section as a benchmark for 2D object detection. However, the paper Disentangling Monocular 3D Object Detection was published proving the inaccuracy of the AP metric at 11 recall points. Currently, KITTI uses the AP at 40 recall points following the recommendation of the paper (Simonelli et al., 2019) which ignored the precision value at recall equal to 0. The 2D benchmark AP at 40 recall points is represented as $2DAP_{R40}$.

Figure 18: $2DAP_{R40}$ Evaluation Perspective

The AP can be used for 3D detection evaluation as well. In this case, the IoU metric represents the intersection of volumes and not intersection of areas shown in Fig 15. This metric takes in account the object pose (position and orientation) in the 3D world as well the object dimensions. The 3D benchmark AP at 40 recall points is represented as $3DAP_{R40}$.

Figure 19: $3DAP_{R40}$ Evaluation Perspective

Similarly, this can be used for bird-eyes view evaluation. Here the IoU represents the intersected area like the case of 2D object detection. The metric takes in account the pose (position and orientation), however it excludes the effect of the predicted object position in height dimension. The bird-eyes view benchmark AP at 40 recall points is represented as $BEVAP_{R40}$.

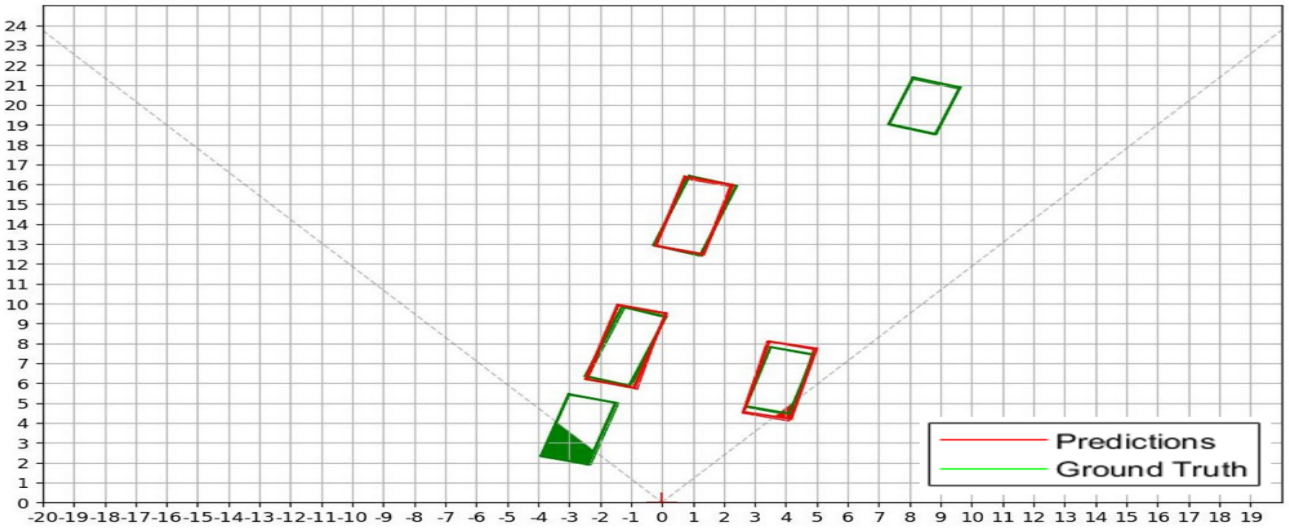


Figure 20: $BEVAP_{R40}$ Evaluation Perspective

Average orientation similarity (AOS) is another metric evaluating solely the predicted orientation of the object. The objects are set on top of each other; thus the predicted object pose is not taken in account in this metric. Similar to $BEVAP_{R40}$ and $2DAP_{R40}$ the IoU here represents the intersection of the areas of the rectangles.

3.3.6 Measuring Uncertainty

There has been significant research in the computer vision community on image uncertainty metrics. Our robustness framework refers to A Comprehensive Evaluation Framework for Deep Model Robustness (Guo et al., 2022) and uses it as guide in measuring image uncertainty.

Two metrics from the guide are selected:

- Average l_p Distortion (ALD_p)

  Most uncertainties cause additive p-norm adversarial perturbations (e.g., p $\in$ 0, 1, ∞). To measure the visual perceptibility of the data, ALD_p is used as the average normalized p distortion:

$$ALD_p = \frac{1}{m} \sum_{i=1}^{m} \frac{||x_{adv}^{(i)} - x^{(i)}||_p}{||x^{(i)}||} \tag{2}$$

  where m denotes the number of adversarial examples that attack successfully, and the smaller ALD_p is, the more imperceptible the adversarial example is.

  The p 1-norm is defined as the maximum absolute column sum of a matrix:

$$ALD_1 = \max_{1 \leq j \leq n} \sum_{i=1}^{n} |a_{ij}| \tag{3}$$

  The p 2-norm is more difficult to characterize, but it can be shown that is the square root of the maximum eigenvalue of the inner product of $A^T A$:

$$ALD_2 = \max_{1 \leq i \leq n} \sqrt{\lambda_i (A^T A)} \tag{4}$$

  The p ∞-norm of a matrix is defined as the maximum absolute sum of the matrix rows:

$$ALD_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^{n} |a_{ij}| \tag{5}$$

  As a remark, the norms mentioned above are normalized in such a way that they are independent of image size as well as number of layers.

- Average Structural Similarity (ASS)

  To evaluate the imperceptibility of adversarial examples, we further use SSIM which is effective to measure human visual perception. SSIM is the most used metric to evaluate the structure similarity between two images. It separates the task of similarity measurement into three comparisons: luminance, contrast, and structure.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(\sigma_x y + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{6}$$

  where:
  μ_x and μ_y : mean of input images x and y
  σ_x and σ_y: covariance of input images x and y
  σ_{xy}: covariance of input images x and y
  c_1 and c_2: constants
  Thus, ASS can be defined as the average of SSIM for the complete dataset:

$$ASS = \frac{1}{m}\sum_{i=1}^{m} SSIM(x_{adv}^{(i)}, x^{(i)}) \tag{7}$$

  where m denotes the number of successful adversarial examples, and the higher ASS is, the more imperceptible the adversarial example is.

  Experimental results including parametric and aleatoric uncertainties generated in SimCenter Prescan favored the ASS metric for measuring data uncertainty. The ASS metric reflected the uncertainty due to fog in the range 10 - 60 m in a wide uncertainty range shown in Fig 22. Unlike ALD_∞ that only observes a narrow uncertainty range as shown in Fig 21 . Furthermore, the computer vision toolbox in Matlab provides a module for computing the SSIM. Making ASS an ideal choice for the framework. The framework defines uncertainty μ = 1 - ASS.

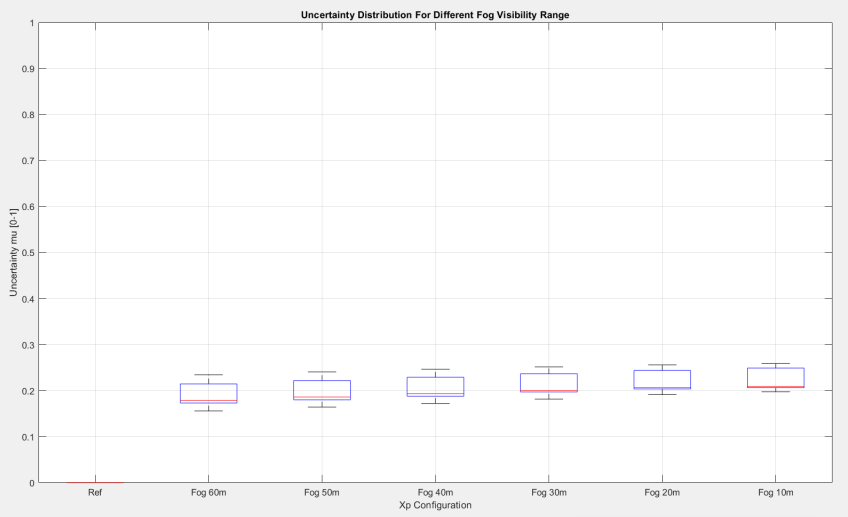


Figure 21: ALD_∞ Uncertainty at different fog visibility range

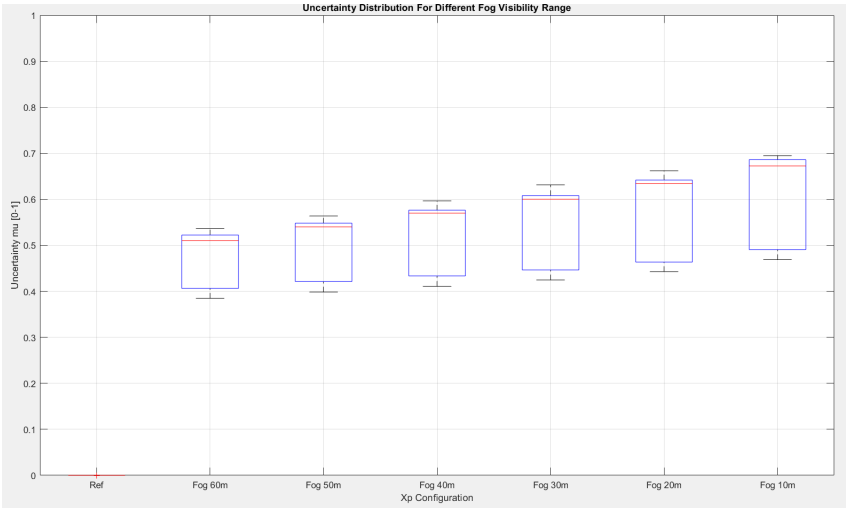
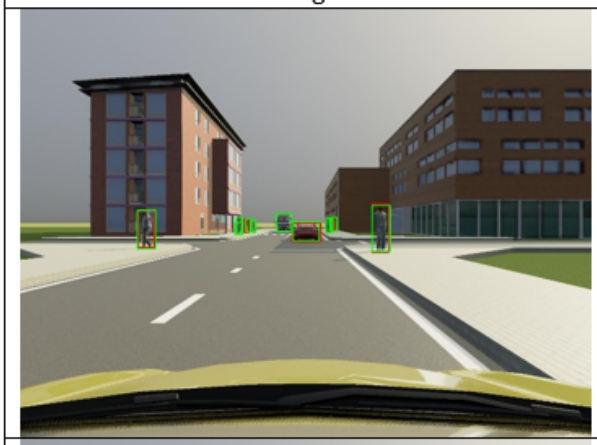
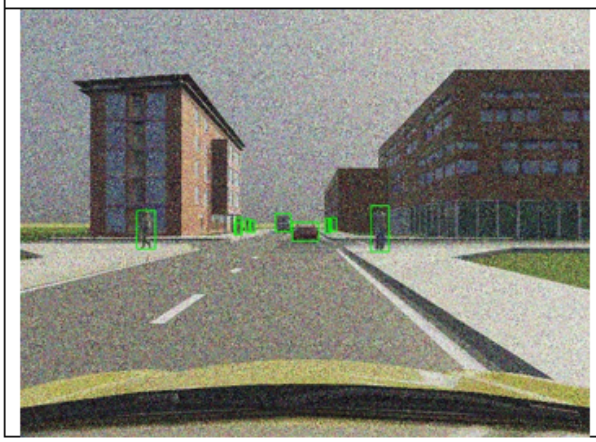
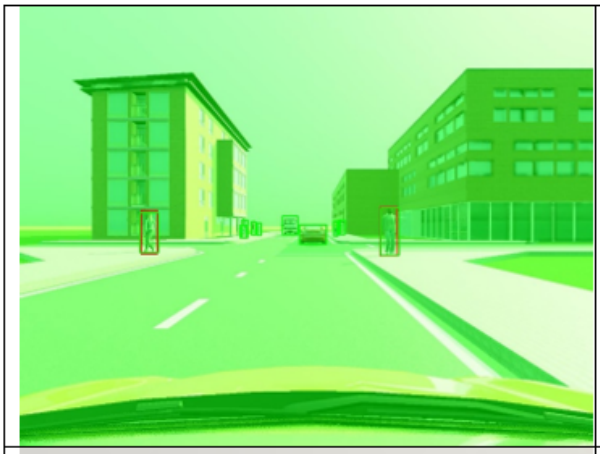
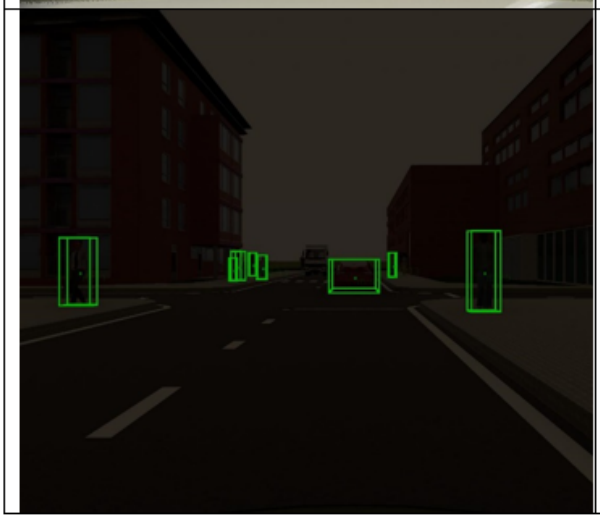


Figure 22: μ Uncertainty at different fog visibility range

3.3.7 Aleatoric and Parametric Uncertainties

Image	Uncertainty	Info
	None (Reference Image)	None
	Aleatoric Uncertainty (Row Address Fault Image)	A row address fault occurs when there is an error or malfunction in accessing or reading out a specific row of pixels in the image sensor. This fault can result in various image artifacts or anomalies
	Aleatoric Uncertainty (Random Noise Fault)	Random noise can be added to the image using the normal distribution mean and variance.

	Aleatoric Uncertainty (Color Fault Image)	A color fault can be injected to the image by adding a shift value to one or more of the RGB color channels.
	Parametric Uncertainty (Image with Fog Uncertainty)	Fog visibility is configured in SimCenter Prescan by setting the max distance in which the scene is visible due to fog.
	Parametric Uncertainty (Image with Illumination Uncertainty)	Illumination is configured by Prescan.

3.3.8 SMOKE and YOLOv3 Performance Comparison

SMOKE and YOLOv3 performance metrics are compared on both the car and pedestrian class. YOLOv3 outperforms SMOKE on the 2D AP Object detection metric described earlier. It should be noted that YOLOv3 is trained on the COCO dataset is expected to outperform SMOKE which trained on the KITTI dataset due to the larger dataset size and greater object variety.

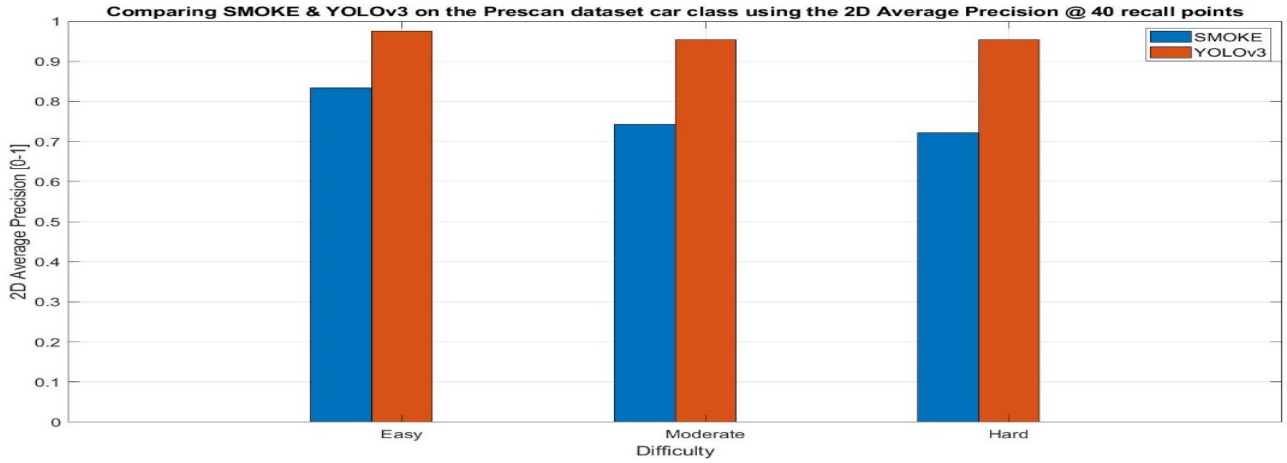


Figure 23: 2D AP Performance Comparison SMOKE and YOLOv3 for car class.

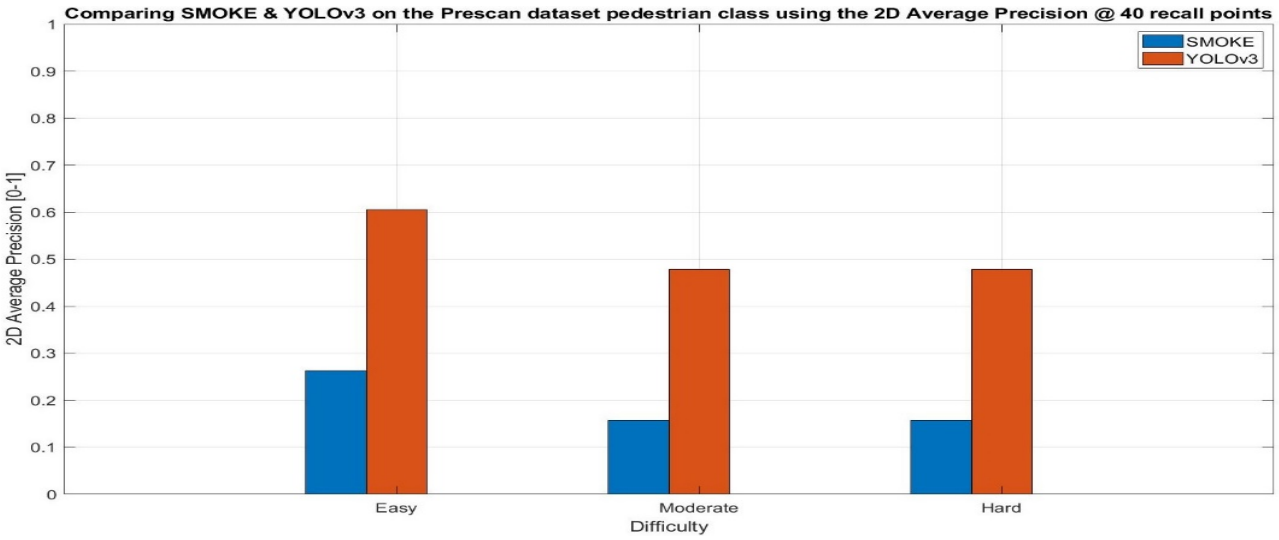


Figure 24: 3D AP Performance Comparison SMOKE and YOLOv3 for pedestrian class.

3.3.9 YOLOv3 Robustness Results

According to the ethics guidelines for trustworthy AI (EU-study/report, 2019) the development, deployment and use of AI systems should meet the seven key requirements for trustworthy AI: (1) human agency and oversight, (2) technical robustness and safety, (3) privacy and data governance, (4) transparency, (5) diversity, non-discrimination, and fairness, (6) environmental and societal well-being and (7) accountability.

Therefore, one of the robustness investigations presented hereby, are related to technical robustness and safety considering diversity, non-discrimination, and fairness. In this sense, the robustness of the perception system is studied, when different vulnerable road users, which belong to different age, gender and ethnic categories are considered. In Simcenter Prescan many different vulnerable road users are already predefined.

Another investigation is carried, when different illumination conditions (dawn and dusk) as well as different weather conditions (fog) are considered. The reference data set is gathered considering daytime illumination and normal weather (no precipitations and no particulates). Here we test YOLOv3 against different vulnerable road users (VRU) in line with discrimination requirement for trustworthy AI from EU guidelines. Very little to no performance loss is observed when different male pedestrians are placed in the experiment. However, a correlation between object size and performance is noticed.This is confirmed by replacing the pedestrians in the experiment with the child which is of smaller size, and a significant loss in performance is shown below.

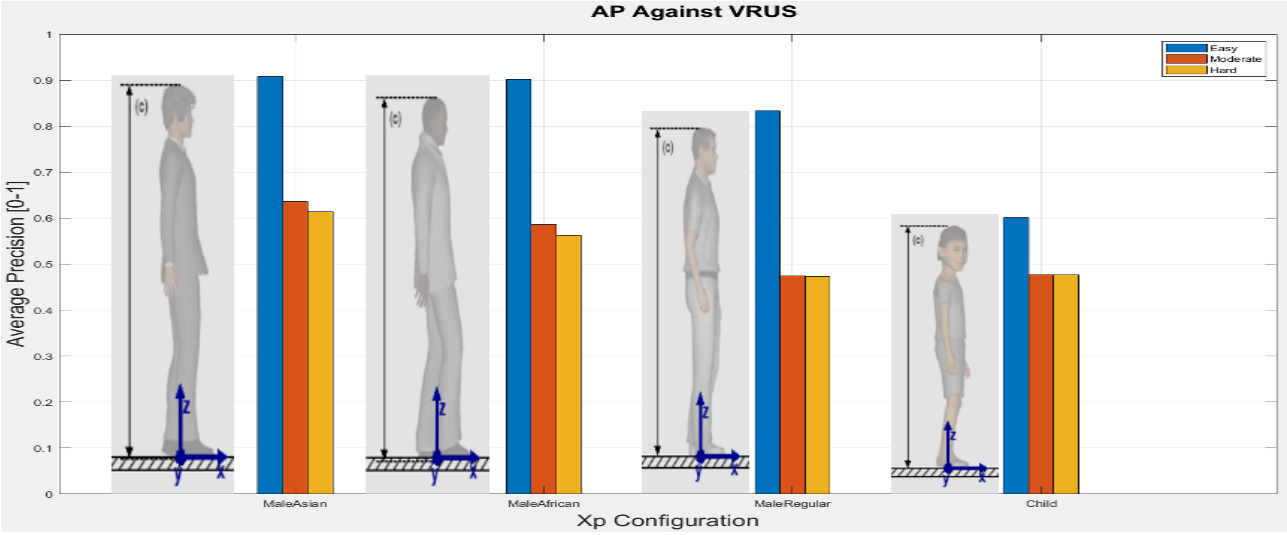


Figure 25: Yolov3 results against different VRUs.

When investigating technical robustness, as uncertainty increases the performance of Yolov3 deteriorated. This observed for uncertainty due to fog and random noise as shown below.

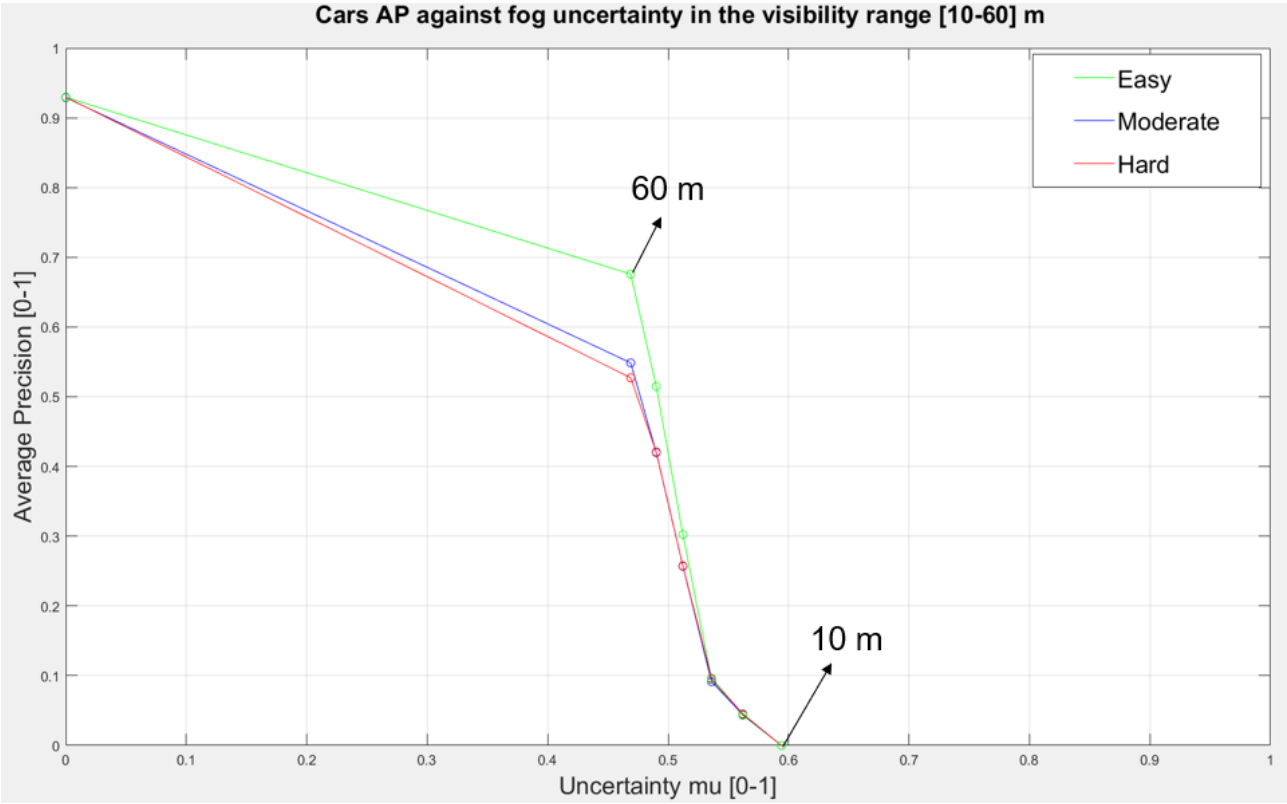


Figure 26: Yolov3 results against Fog.

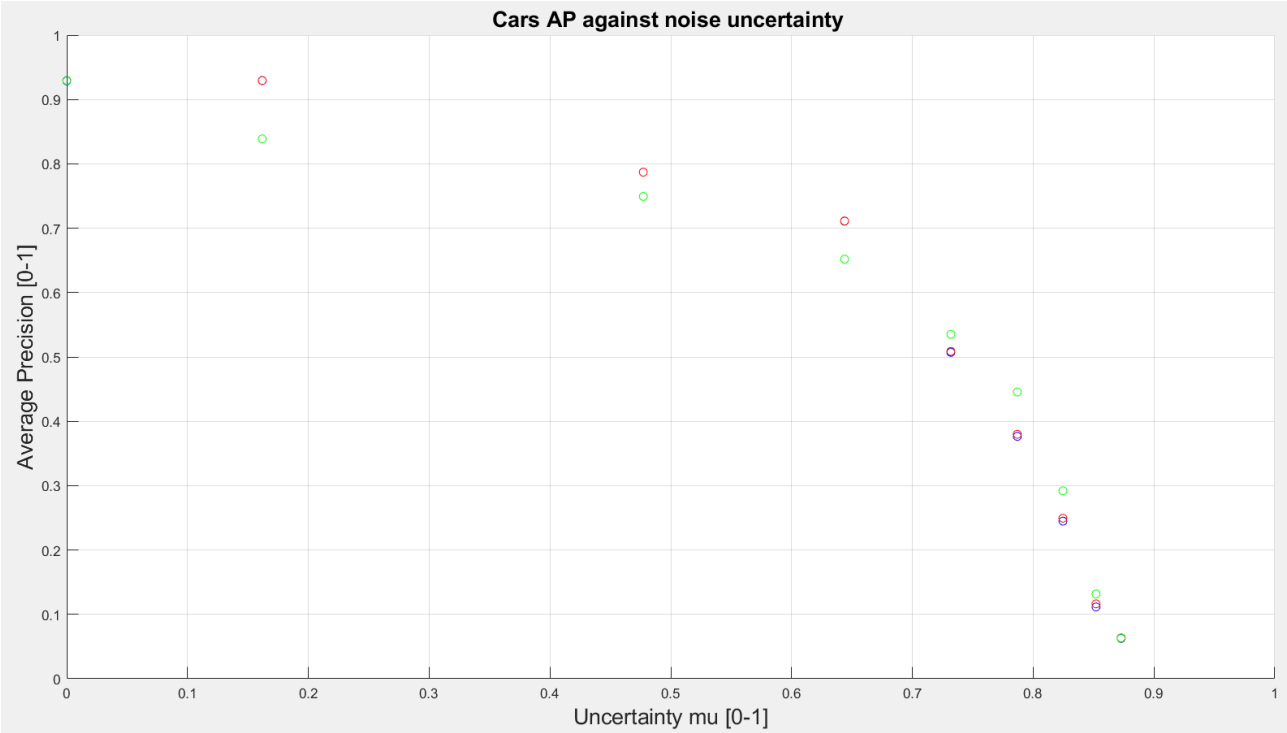


Figure 27: Yolov3 results against Noise.

4 Stereo Vision Depth Estimation

What are the practical challenges of implementing Stereo Vision in driving scenarios?

2D Object detectors such as YOLOv3 are insufficient for autonomous driving. They lack information about object pose and dimensions. This limits collision avoidance and path planning capabilities. Autonomous vehicles need detailed spatial information for safety.

Stereo vision, also known as binocular vision, obtains spatial information by leveraging the disparity between corresponding points in a pair of stereo images captured from slightly different viewpoints. Disparity refers to the positional difference or shift of corresponding points between the two images. By comparing the differences in pixel locations between the left and right images, stereo vision algorithms can estimate the depth or distance of objects in the scene. This depth information provides valuable spatial cues that can greatly assist YOLOv3 and SMOKE. By combining the depth information from stereo vision with object detectors, algorithms can accurately localize objects in 3D space, estimate their size, and distance from the camera.

This additional depth information enhances the understanding of the scene geometry, improves the robustness of object detection in complex scenarios, and enables more precise localization and tracking of objects, particularly in applications like autonomous driving or robotics, where accurate 3D perception is crucial.

In this project, two prototypes of stereo vision algorithms were developed and tested. However, none of them result in reliable accuracy. The following section walks through the theoretical background behind each of the prototypes and the challenges faced.

4.1 Stereo Vision Depth Estimation

A common methodology is shown below for achieving stereo vision involves image acquisition, undistortion, rectification, stereo matching, and point cloud generation.

The process begins with an input image captured by a stereo camera setup, which consists of two cameras positioned slightly apart to mimic the human binocular vision. The first step is undistortion, where lens distortions are corrected in both images to ensure accurate measurements. Next, rectification is performed to align the images along a common epipolar line, simplifying subsequent computations. Stereo matching is then applied, where corresponding points in the two rectified images are matched to establish correspondences between the left and right views. Finally, triangulation is employed to determine the depth information by calculating the disparities between corresponding points and applying geometry to obtain the three-dimensional positions in

the scene. This sequential process enables stereo vision systems to reconstruct a scene's depth and provide a more comprehensive understanding of the environment.

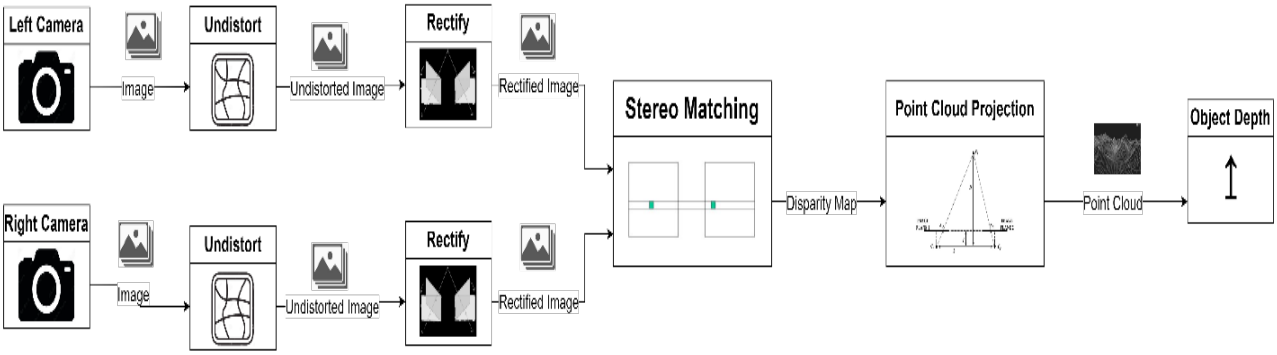


Figure 28: Common Methodology For Stereo Vision

4.1.1 Distortion

Figure 29: Before and After Distortion

Given the radial and tangential distortion coefficients, k and p respectively:

$$D = [k_1 \quad k_2 \quad k_3 \quad p_1 \quad p_2]$$

Radial distortion is represented as

$$x_{distorted} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{8}$$

$$y_{distorted} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{9}$$

While tangential distortion is represented as:

$$x_{distorted} = x(2p_1 xy + p_2(r^2 + 2x^2)) \tag{10}$$

$$y_{distorted} = y(2p_1 xy + p_2(r^2 + 2x^2)) \tag{11}$$

The images are then undistorted using the inverse of the equations above.

4.1.2 Rectification

The process of image rectification involves computing two homographies that can be applied to a pair of images to make them parallel. A homography can be seen as a matrix or mask representing a transformation to be applied to the image.

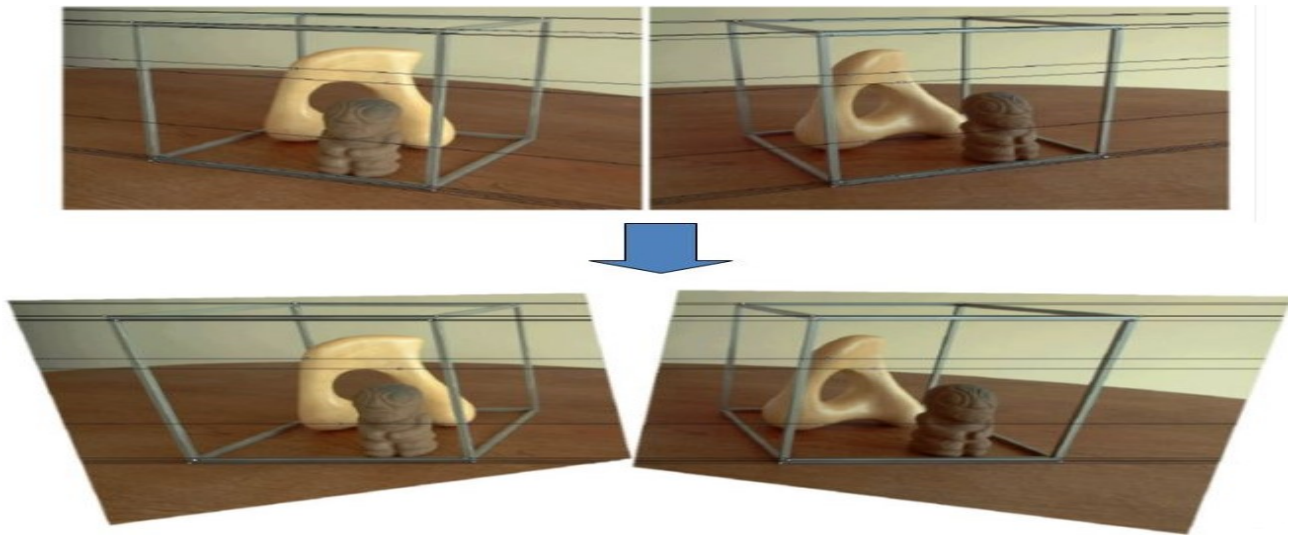


Figure 30: Image Rectification

The approach to rectification involves decomposing each homography into a projective and affine component. Then find the projective component that minimizes a well-defined projective distortion criterion. Then further decompose the affine component of each homography into a pair of simpler transforms, one designed to satisfy the constraints for rectification, the other is used to further reduce the distortion introduced by the projective component (Loop and Zhang, 1999).

An interesting case of epipolar geometry occurs when the image planes are parallel to each other. The projection of a point in left image must be located on the epipolar line of the second image. When the image planes are parallel this forces that point in the left image and point in the right image to share the same y-coordinate. Consequently, there exists a very straightforward relationship between the corresponding points as shown Fig 31. Understanding epipolar geometry allows us to rectify the images to exploit the geometric constraint resulting in reliable point correspondence.

4.1.3 Stereo Matching

Given the geometric constraint imposed by the rectification process, a point x in one image generates a line in the other on which its corresponding point x must lie. We see that the search for correspondences is thus along that line.

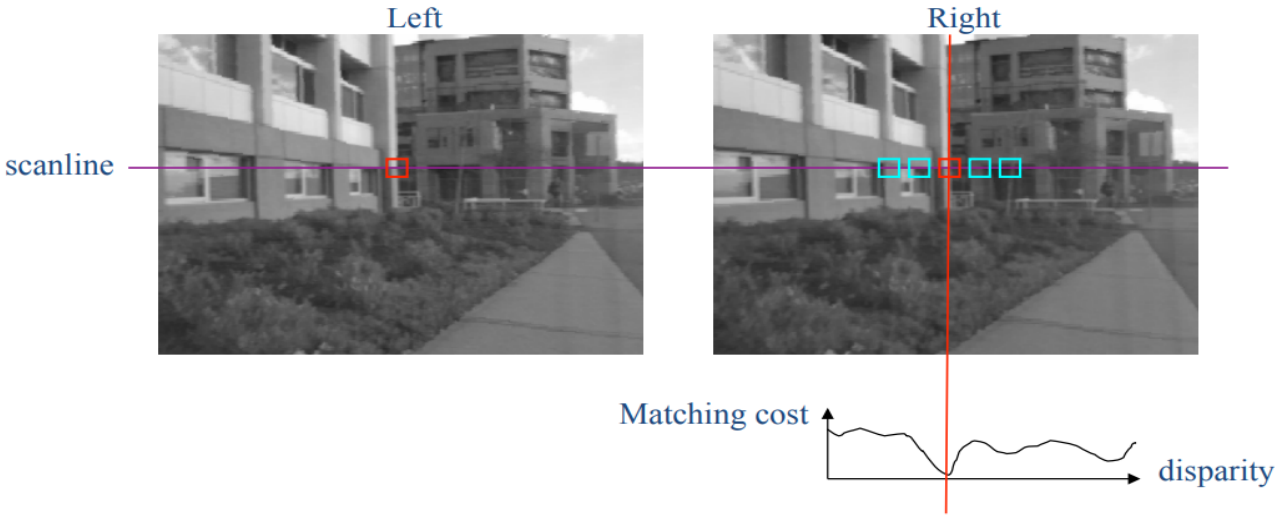


Figure 31: Stereo Matching via Semi-Global Matching

The following is a description of the cv2.StereoSGBM() function in OpenCV for computing disparities in stereo images (Hirschmuller, 2008):

- Calculate matching cost for each pixel based on intensity values and different methods.
- Apply smoothness constraint to penalize changes in neighboring disparities and calculate aggregated cost.
- Use Dynamic Programming and SGM to optimize disparity and avoid streaking artifacts.
- Apply iterative refinement to initial disparity image for improved accuracy and use a hierarchical approach to reduce computation time.

4.1.4 Point Cloud Projection

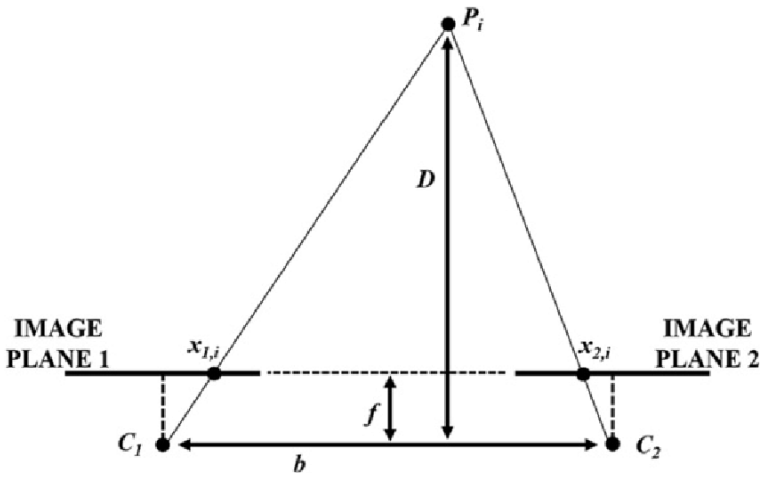


Figure 32: Depth By Triangulation

Using the similar triangles formed by the vertices C1, C2 and Pi and X1,X2 and Pi the Depth is obtained via triangulation:

$$D = \frac{f * b}{X2 - X1} \tag{12}$$

Given the object depth Z/D, point on the image plane (x,y), and the Camera Matrix, the projection (X,Y,Z) is computed using:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = K^{-1} \begin{pmatrix} Z * x \\ Z * y \\ Z \end{pmatrix} \tag{13}$$

4.2 Object Localization with Stereo Vision and Object Detector

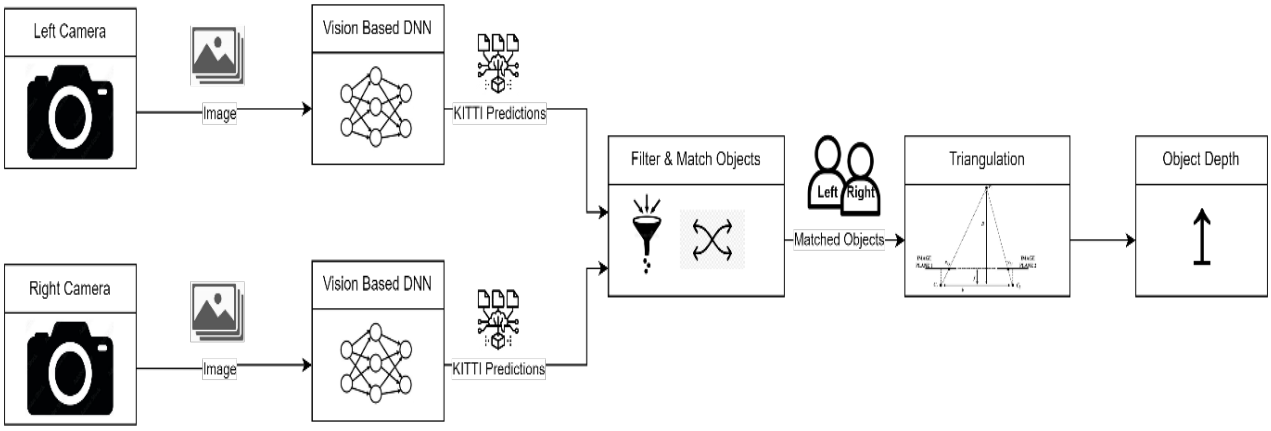


Figure 33: Methodology For Stereo Vision

Due the challenges faced with the previous method, an alternative solution where only the object depth is obtained and not a depth map/point cloud was investigated. Yolov3 is used to detect objects in both left and right frames. Only objects detected in both left and right images are considered due to nature of triangulation. The objects are then matched based on their bounding box coordinates and scene keypoint features. Finally, the depth is obtained by calculating the disparity of the matched objects and triangulation.

4.2.1 Object Correspondance with Object Detector and Feature Keypoints

Image keypoint features represent interesting points in the scene that are invariant to perspective. A keypoint can be viewed as a point of interest in the 3D World, where every keypoint has a pair of point coordinates. The point coordinates represent the 2D Image Plane coordinates for the left and right images.

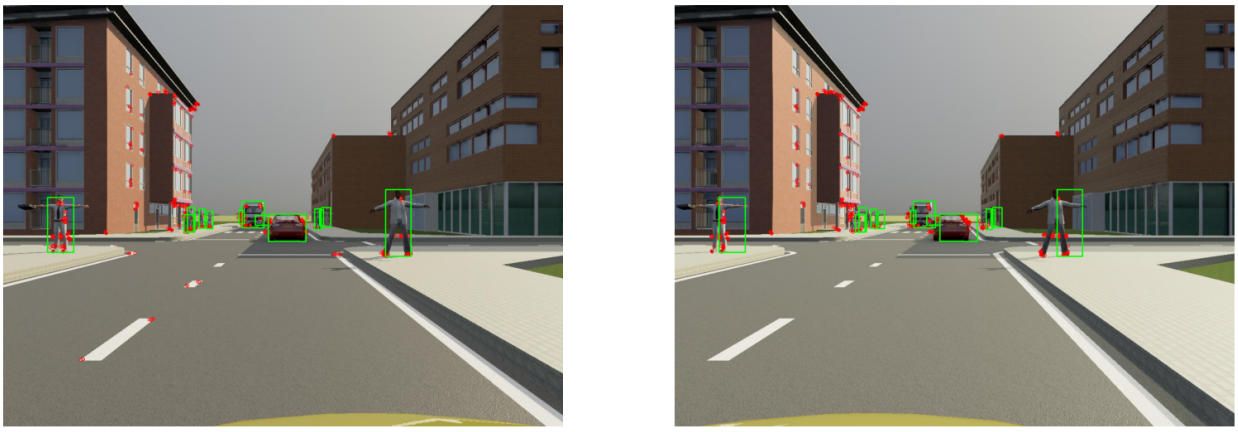


Figure 34: Left and Right Image Features Visualized Alongside Ground Truth Bbox

Figure 35: Object Correspondance via matched keypoint in left and right images

We match detected object in left and right images by checking if the key points left point is within a bounding rectangle and similarly for the key points right point. If both conditions are satisfied we associate the objects according to their bounding rectangles.

This alone isnt robust enough to match objects very close to each or that have a small bounding rectangle, we can see in the Fig 34 that some key points can be within more than one bounding rectangle.

To account for that, we iterate over all the key points, where every key point votes for a potential match. Finally, we select only unique matches with the highest votes.

4.2.2 Object Localization Experimental Results

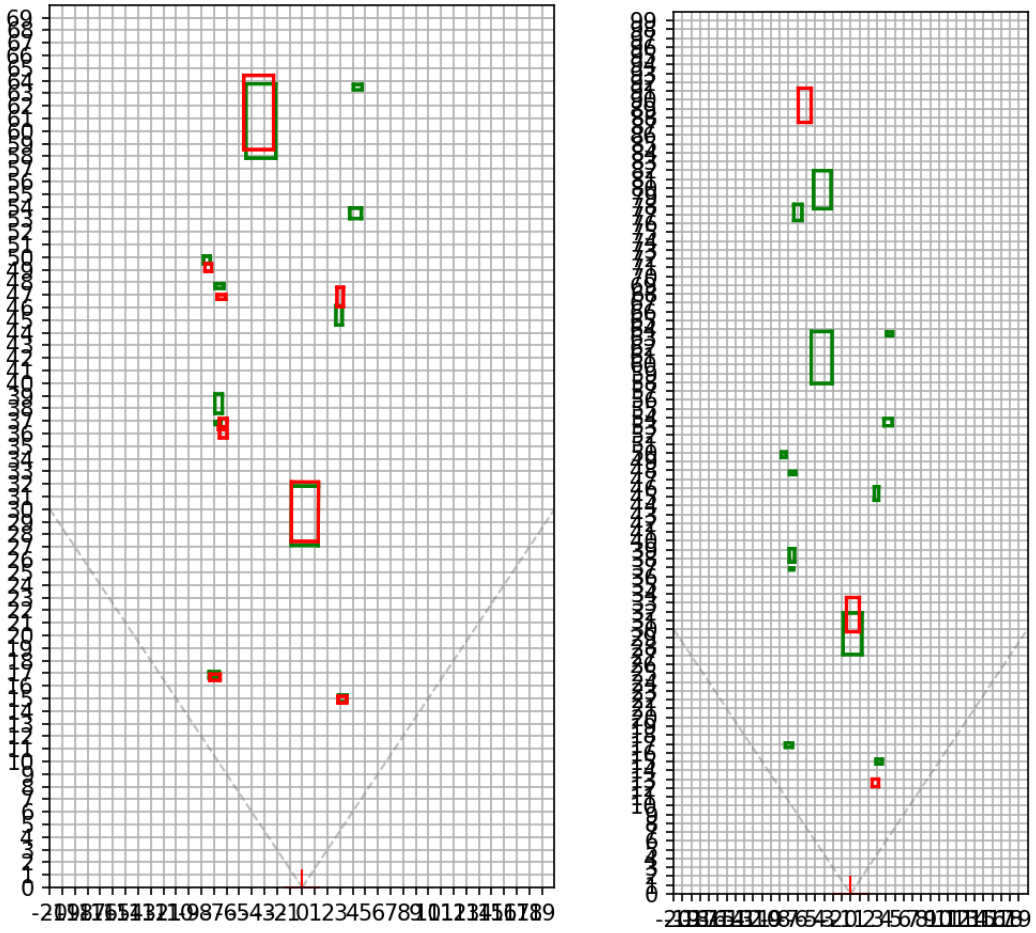


Figure 36: Left image shows object localization calculated with groundtruth boxes while the right images shows object localization calculated with predicted boxes.

In Fig 36 the left image demonstrates that the object localization method with stereo vision returns good approximates of the object pose. Method works up to a sufficient range around 60 meters, but larger errors are observed specially with small objects such as pedestrians.

The disparity was calculated according to the ground truth bounding boxes. This is to independently validate the triangulation and projection methods in equations 12 and 13. However, to deploy this in practice we cant rely on ground truth and therefore must use the predicted bounding boxes by YOLOv3.

When the predicted bounding boxes are used for computing the disparity, its observed that the localization is not as accurate as shown in the right image of Fig 36. This is due to the high sensitivity of the triangulation formula to disparity. A small error in disparity leads to a large error margin for depth.

4.3 Stereo Vision Practical Challenges

The main challenge encountered with the implementation of a stereo vision algorithm was in the image rectification subsystem shown in Fig 28. As described in the Stereo Matching section, image rectification is critical to reliable stereo matching, as it imposes a geometric constraint that facilitates the Semi-Global Matching algorithm. Without this constraint, the matching algorithm performs poorly. The implementation of this step was using OpenCV module using functions such as cv2.stereoRectify.

Due to the unsatisfying results obtained using the conventional way of approaching stereo vision, a workaround was proposed that used object detector to match the objects in left and right images. The matching is executed by matching the bounding boxes in left and right images, using feature key points in the scene. However, in practice this required near perfect IoU (metric described in 4.3.3.1 Common Industry Metrics) for accurate depth estimation. Which is not the case with YOLOv3, which is confirmed in (Redmon and Farhadi, 2018) as it's mentioned:This indicates that YOLOv3 is a very strong detector that excels at producing decent boxes for objects. However, performance drops significantly as the IOU threshold increases indicating YOLOv3 struggles to get the boxes perfectly aligned with the object. As a result, since the bounding boxes are not perfectly aligned, theres always an error in the computation of the disparity which results in a high error margin given equation 12.

However, even if the workaround method resulted in accurate depth estimation, it would not have been an ideal choice. This method requires the system to predict objects in both left and right images which is a computational drawback. Also, object depth can only be obtained for objects that are detected in both images. This suggests that some objects are ignored when they are only detected in one image due to occlusion or other factors, therefore such a method will perform worse on detection/classification compared with using one camera only.

The challenges encountered here lead us to look for answers in the industry and review what is currently considered state-of-the-art in stereo vision.

5 Comparison of Monocular and Stereo Based Object Detectors

What are the differences between monocular and stereo vision perception systems in terms of accuracy?

Due to our inability to develop a reliable stereo vision algorithm, this section refers to the weakness of SMOKE discovered during internship project, and the theoretical results from the research papers of SMOKE and DSGN2 to answer the research question. DSGN2 stands for Deep Stereo Geometry Network, which is a currently state-of-the-art stereo based algorithm for 3D object detection. Its important to note that the conclusion in this section does not generalize to all monocular and stereo methods for object detection but is specific to SMOKE and DSGN2.

It is shown how SMOKE is not suitable for deployment in new scenarios, as it requires training assisted by mean and standard of deviation of the depth distribution of the dataset (which is not available during deployment). In addition to that, the performance of DSGN2 is compared with that of SMOKE to determine which method outperforms the other both on 2D and 3D detection.

5.1 Monocular Limitation in SMOKE

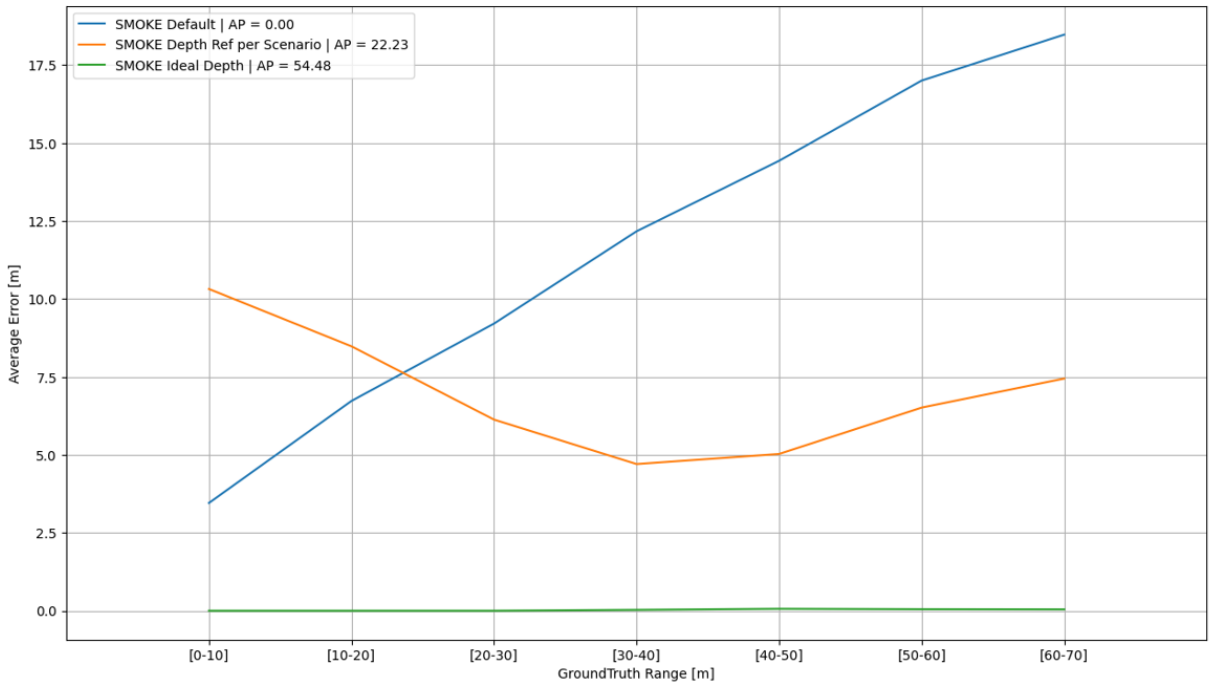


Figure 37: Depth Estimation Evaluation for different methods on a dataset from Prescan.

The default SMOKE depth estimation parameters (mean, std) from the KITTI dataset do not generalize. The average depth error (blue) is too high over the complete range causing very poor detection/classification performance. Refer to Appendix A equation 3 for SMOKEs depth estimation.

Therefore, the depth estimation parameters (mean, std) were calculated from the Prescan Scenario, exactly like it was calculated for KITTI. The method adopting depth parameters based on the complete scenario (orange) surprisingly does not perform well either.

To reflect on that, we asked ourselves how come SMOKE depth estimation error is in the range 0-5 meters on KITTI dataset with KITTI depth stats but performs poorly on Prescan dataset with Prescan depth stats with an estimation error range 5-10 m? As illustrated in Fig 38 and Fig 37.

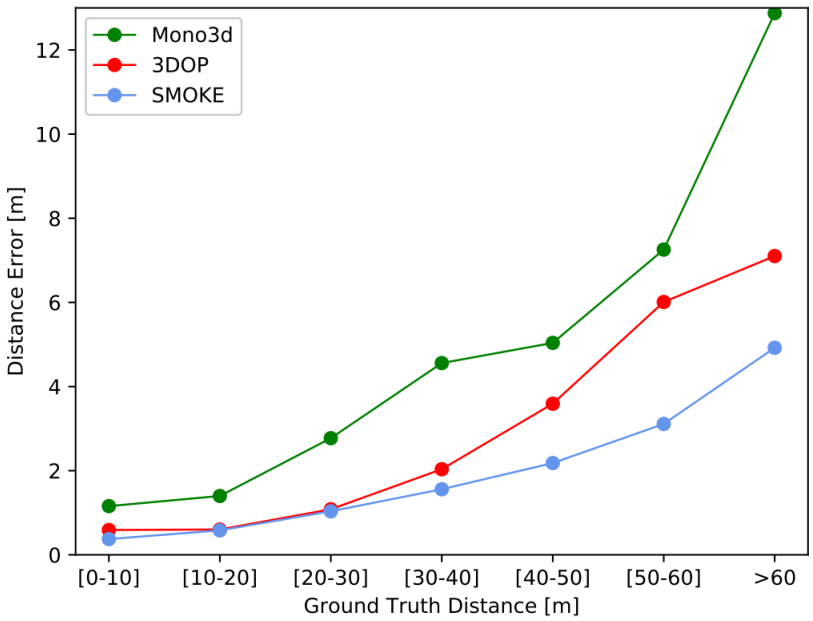


Figure 38: SMOKE Distance Error as a function ground truth distance on the KITTI dataset.

To answer that we dig deeper into depth equation and figure out how the model learns to predict the relative depth. During training we provide the absolute depth via the ground truth labels. Therefore, for the model to learn relative depth prediction, it must compute relative depth in the loss function (during training) using:

$$\delta_z = \frac{Z - \mu_z}{\sigma_z} \tag{14}$$

where:
δ_z: depth offset [unitless]; predicted by model
Z: depth in meters
μ_z: mean of depth distribution in dataset
σ_z: standard of deviation of depth distribution in dataset

This means that the depths stats guide the model to learn depth offset δ_z during training and expects the same depth stats for deployment. Since our model is trained on KITTIs depth stats but uses Prescan depth stats for deployment,

it performs poorly. Since training SMOKE is out of scope of this internship assignment, the authors were contacted to confirm the hypothesis.

This dependency on the depth stats appears to be the main limitation of SMOKE, even if we retrain SMOKE with the relevant depth stats and get the results mentioned in the paper, what will happen when SMOKE is deployed in a new scenario with different depth distribution that we do not have?

SMOKE will perform poorly, like it did for the default implementation of SMOKE on the Prescan dataset as shown in Fig 37. To enable reliable SMOKE deployment, we need to eliminate the depth estimation dependency on the depth stats, by using another method for depth estimation such as Lidar or Stereo Vision.

5.2 Comparing SMOKE and DSGN2 on the KITTI dataset

Here the methods are compared based on their detection/classification ability as well as their depth estimation ability.

SMOKE outperforms DSGN2 on the 2D APR40 metric. As explained in section 3.3.2.2 this metric independently evaluates detection/classification. This demonstrates that the objects as points approach shown in Fig 13 is a good method for detection/classification as it competes with current state-of the art methods. Its also concluded that stereo based methods do not necessarily outperform monocular methods for detection/classification, as shown in Fig 39.

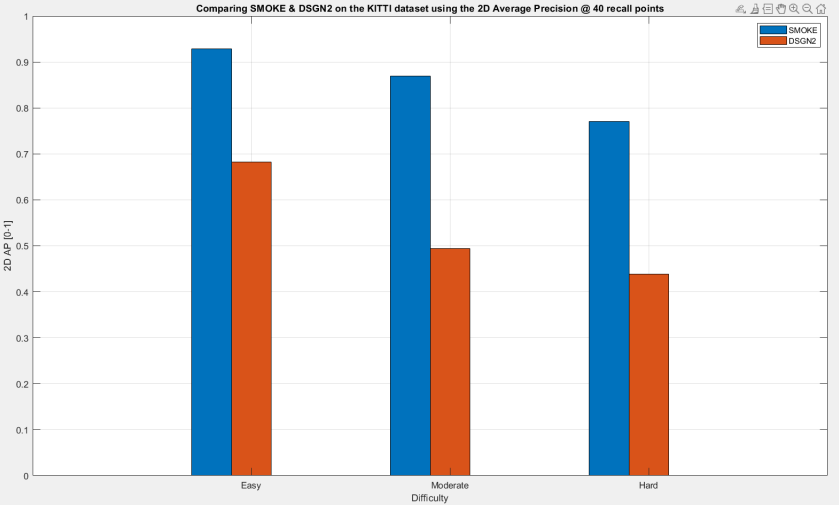


Figure 39: 2D AP Evaluation Comparison

However, comparing the methods while taking the pose, orientation, and object dimensions in account as explained in section 3.3.2.2, DSGN2 outperforms SMOKE on the on the 3D APR40.

Its not surprising that a stereo-based method outperforms a monocular method in object localization, considering the inherent challenges involved in 3D re-

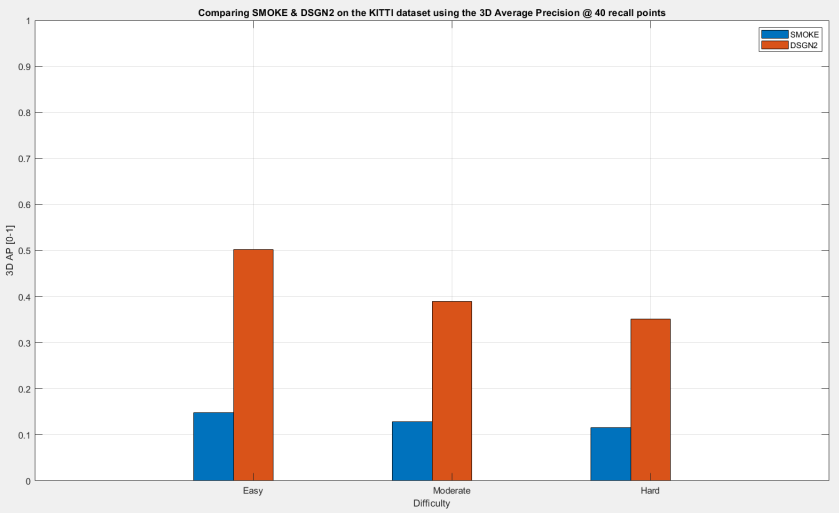


Figure 40: 3D AP Evaluation Comparison

construction. Stereo vision leverages the presence of two cameras, to extract depth information and create a 3D representation. This additional dimension enables more accurate localization of objects in the environment. On the other hand, monocular methods rely solely on a single camera, which lacks depth perception. Although advancements have been made in monocular depth estimation, it remains a fundamentally more challenging task compared to stereo-based approaches. Therefore, given the intricacies of 3D reconstruction, the superior performance of stereo-based methods in object localization is to be expected.

6 Conclusion

How to design, verify and validate robust camera-based sensing and perception sub-systems into autonomous vehicles?

Verification and validation of robust camera-based sensing and perception sub-systems is possible via computer simulation and synthetic data. The framework developed during this project is implemented in SimCenter Prescan. The framework allows verification of requirements from the EU guidelines on robust and trustworthy AI such as indiscrimination and technical robustness. By setting up an autonomous driving scenario in simulation with various vulnerable road users from all races and genders, the discriminatory behavior of the perception system is tested. While for technical robustness, the perception system is tested in various weather conditions such as fog and other sources of uncertainty such as hardware faults and noise. In this framework, the KITTI data format is adopted for the labels and predictions.

In this project, rapid prototypes of stereo vision algorithms were designed, developed, and tested in simulation. Several challenges with stereo vision were encountered. First, difficulty with image rectification is highlighted with its effect on accuracy of depth estimation via stereo vision. Then, the triangulation method in section 5.1.4 displayed high sensitivity towards the disparity, where a, error of a few pixels result in large error margin for depth estimation.

Our investigations of stereo vision did not result in the desired functionality of accurate localization via the camera-based system.

With advancements in technology, such as high-resolution cameras and improved processing power, stereo vision systems have become highly reliable, enabling precise navigation and interaction in various applications, from autonomous vehicles to robotics. Its wide adoption speaks to its effectiveness in providing accurate depth estimation in todays technological landscape. Therefore, its still considered a potential solution. Given more time to investigate the challenges encountered with image rectification, its possible to achieve accurate depth estimation via stereo vision.

The monocular method SMOKE demonstrated good detection/classification abilities. However, its reliability is highly dependent on the depth estimation branch of the model. While the depth estimation branch, suffers from limitations for deployment in practice. SMOKE requires retraining for specific scenarios. Its also necessary to have the dataset depth distribution when deploying. This does not meet our requirements. Robust perception systems aim to generalize their knowledge beyond the training data, enabling them to make accurate predictions or decisions in new situations.

SMOKE can still be considered a potential solution based on the sensor setup. As our focus was vision-based systems, Lidars were not considered. However,

SMOKE is an ideal solution in a camera-lidar setup. As SMOKEs depth estimation can be greatly improved by relying on the lidar point cloud. Or even the stereo point cloud. Therefore, we emphasize the importance of further investigating stereo vision. Compared with Lidar its a much more cost-effective sensing approach. However, the challenges with image rectification must be explored.

7 Recommendation

An open-source algorithm DSGN2 currently considered state-of-the-art is recommended to be investigated as it is the highest-scoring method on the KITTI benchmarks.

Both SMOKE and a DSGN2,monocular and stereo based methods respectively, were compared in terms of performance of detection/classification and depth estimation. Here its important to note that the conclusion is specific to SMOKE and DSGN2 and does not generalize to all monocular and stereo-based methods. Based on the comparison, its concluded that stereo based methods do not necessarily outperform monocular methods for detection/classification as shown in Fig 39.

However, with respect to depth estimation, its not surprising that a stereo-based method outperforms a monocular method. Considering the inherent challenges involved in 3D reconstruction. Stereo vision leverages the presence of two cameras, to extract depth information and create a 3D representation. This additional dimension enables more accurate localization of objects in the environment. On the other hand, monocular methods rely solely on a single camera, which lacks depth perception. Although advancements have been made in monocular depth estimation, it remains a fundamentally more challenging task compared to stereo-based approaches.

Therefore, given the intricacies of 3D reconstruction, the superior performance of stereo-based methods in object localization is to be expected.

8 Bibliography

EU-study/report (2019). Ethics guidelines for trustworthy ai. In *https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai*.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338.

Guo, J., Wei Bao, J. W., Yuqing Ma, X. G., Gang Xiao, A. L., Jian Dong, X. L., and Wu, W. (2022). A comprehensive evaluation framework for deep model robustness.

Hirschmuller, H. (2008). Stereo processing by semiglobal matching and mutual information. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 30), pages 328 – 341.

Loop, C. and Zhang, Z. (1999). Computing rectifying homographies for stereo vision. Technical Report MSR-TR-99-21.

Nowruzi, F., Kapoor, P., Kolhatkar, D., Hassanat, F., Laganiere, R., and Rebut, J. (2019). How much real data do we actually need: Analyzing object detection performance using synthetic and real data. In *arXiv preprint*, number arXiv:1907.07061.

Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement.

Simonelli, A., Samuel Rota Bulo, L. P., Manuel Lopez-Antequera, Peter Kontschieder, M. R., and of Trento, U. (2019). Disentangling monocular 3d object detection.

9 Appendices