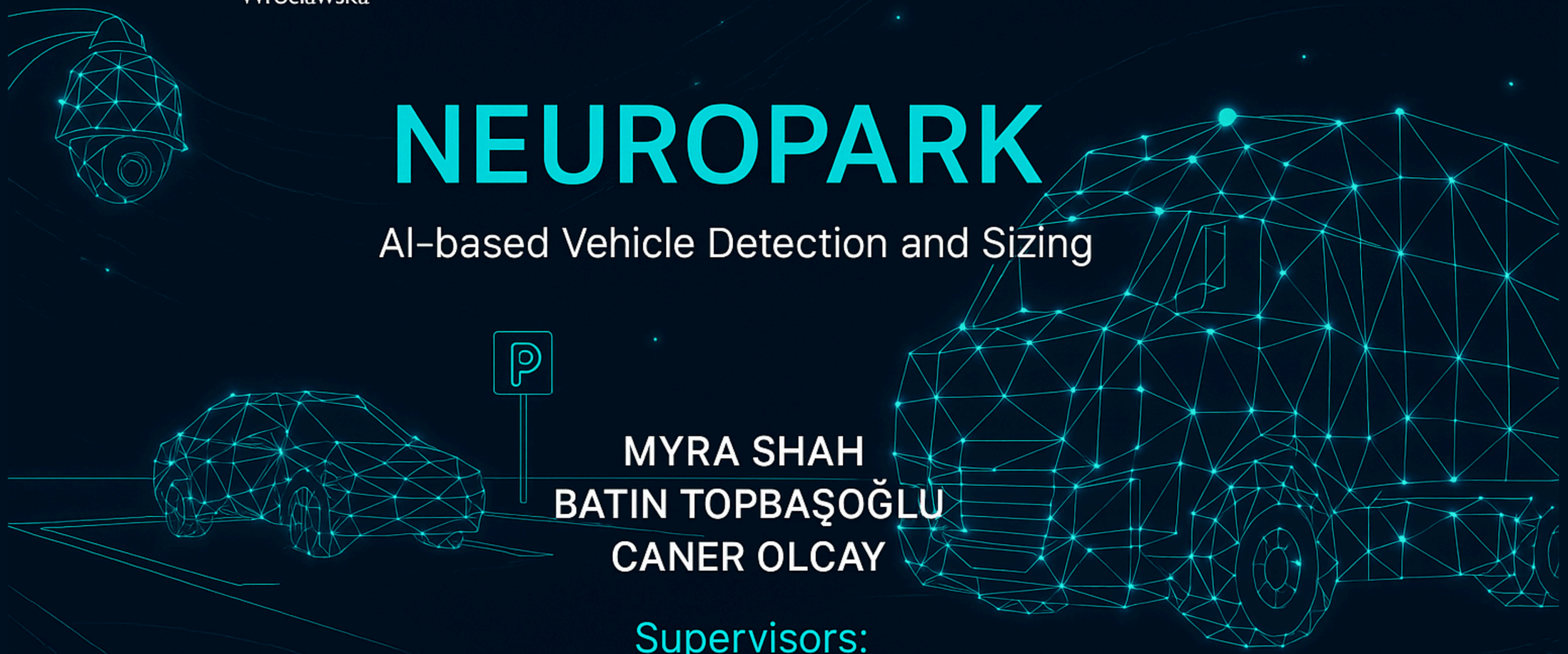
neurosoft

Politechnika Wroclawska

NEUROPARK

AI-based Vehicle Detection and Sizing



MYRA SHAH
BATIN TOPBAŞOĞLU
CANER OLCAY

Supervisors:

Paweł Mrowka Paweł Kaczmarek

INTRODUCTION

GOAL

This project aims to transform 2D traffic camera images into interpretable 3D vehicle scenes.

Main Idea

The system provides spatial understanding of cars, buses, trucks, and motorcycles with LIDAR.

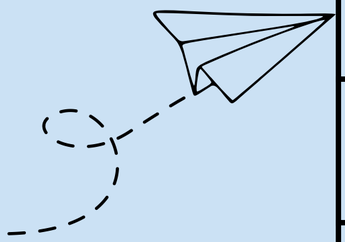
What it focuses

It focuses on vehicle detection, depth estimation, and 3D bounding box generation using a single RGB image.

What is it for ?

It is designed for smart traffic systems, parking area monitoring, and urban mobility analytics.

TOOLS & FRAMEWORKS USED



ultralytics.YOLOv8x	2D object detection
Midas	Monocular depth estimation
Open3D	Point cloud creation, 3D box fitting, scene visualization
NumPy, PyTorch	Mathematical computation, PCA, transformations
OpenCV	Lane detection, Canny & Hough transform
Matplotlib	Result visualization and image saving
scipy.spatial.transform	3D rotations and bounding box alignment
os, PIL.Image	File operations and image loading

Scene Processing Steps

Object Detection (YOLOv8x)

Detects vehicles: car, truck, bus, motorcycle and outputs 2D bounding boxes with class & confidence

Depth Estimation

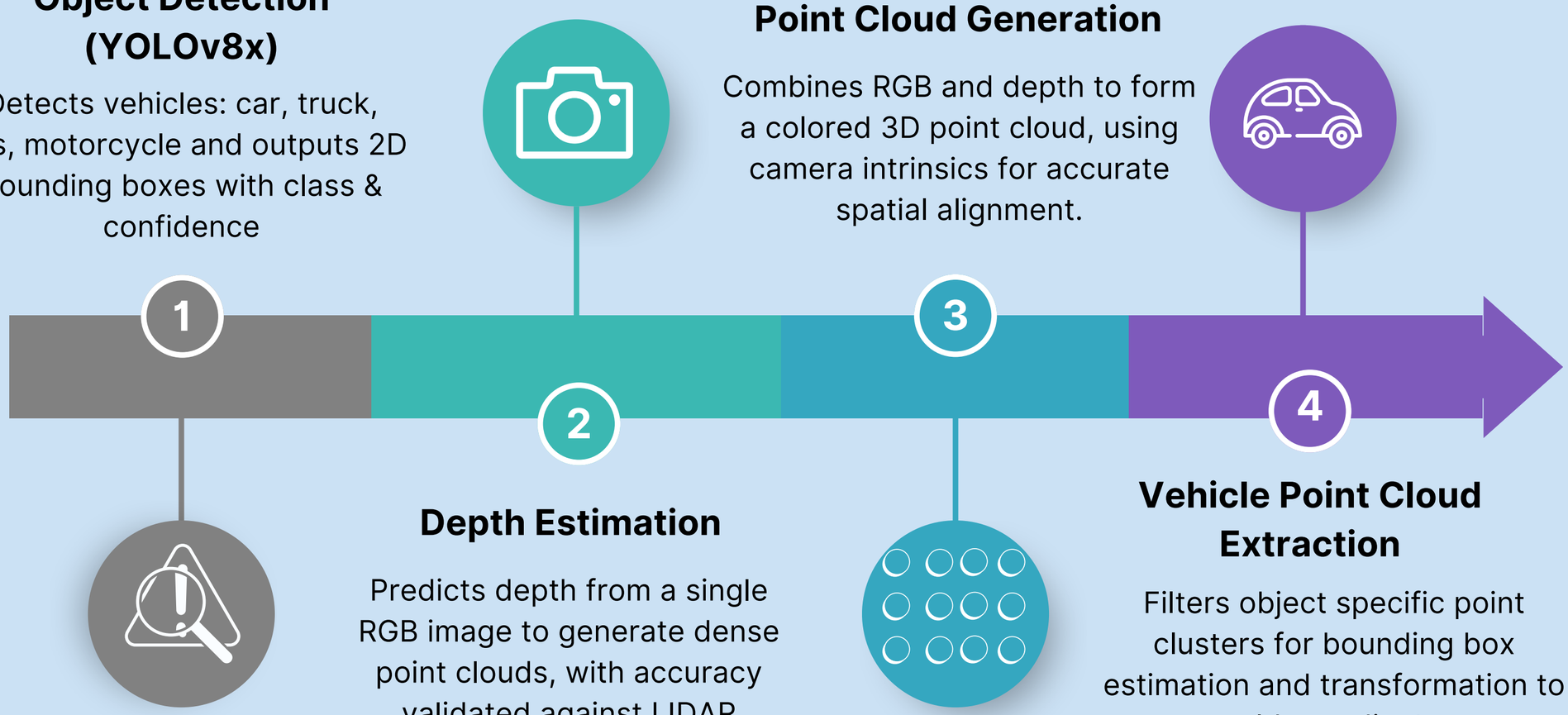
Predicts depth from a single RGB image to generate dense point clouds, with accuracy validated against LIDAR measurements.

Point Cloud Generation

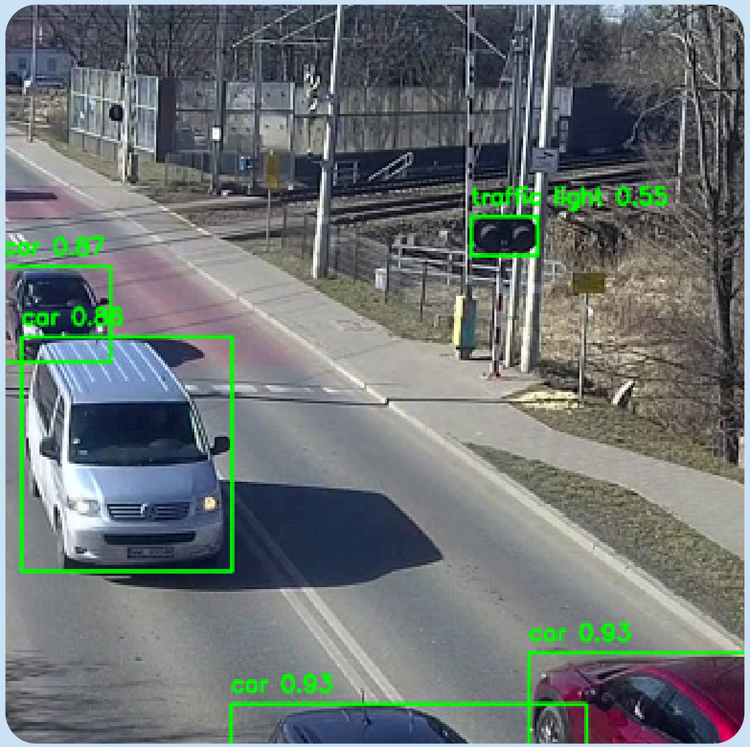
Combines RGB and depth to form a colored 3D point cloud, using camera intrinsics for accurate spatial alignment.

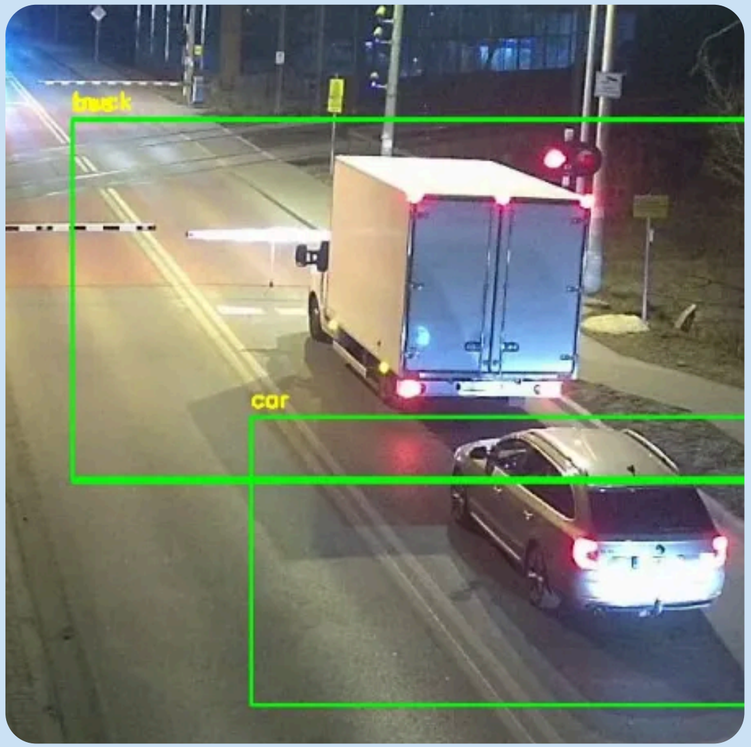
Vehicle Point Cloud Extraction

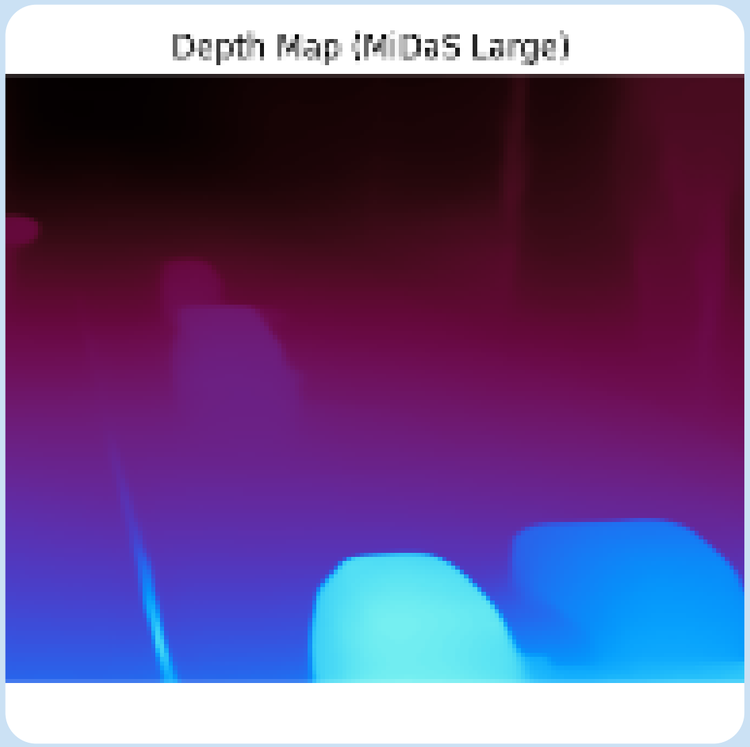
Filters object specific point clusters for bounding box estimation and transformation to world coordinates.

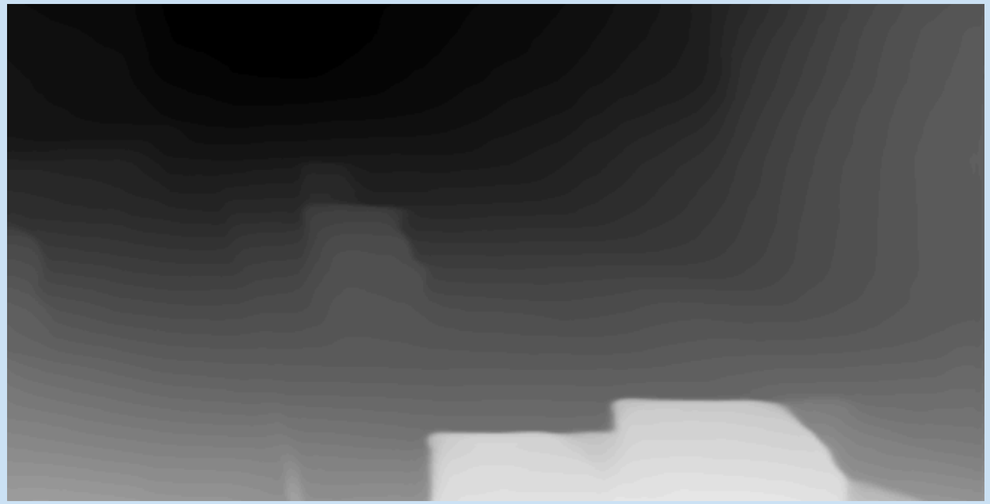


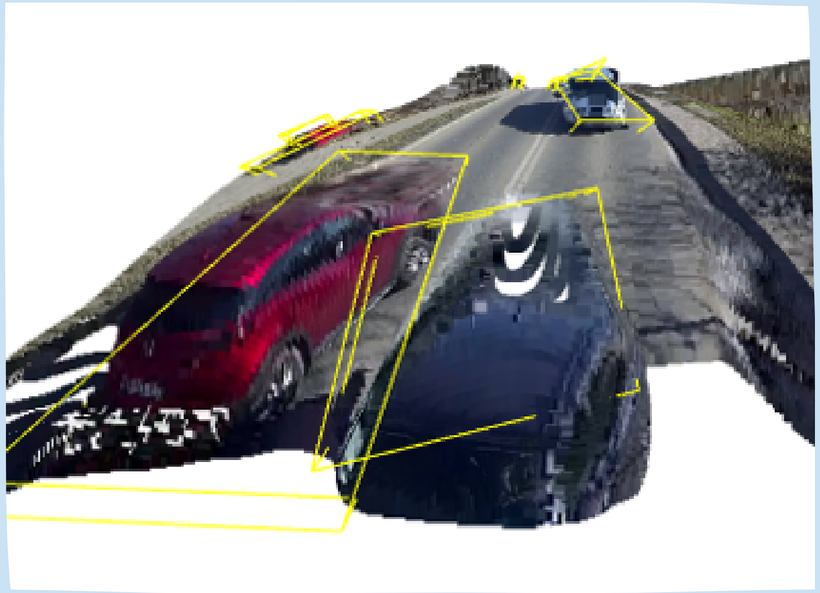
Initial YOLO and Depth outputs

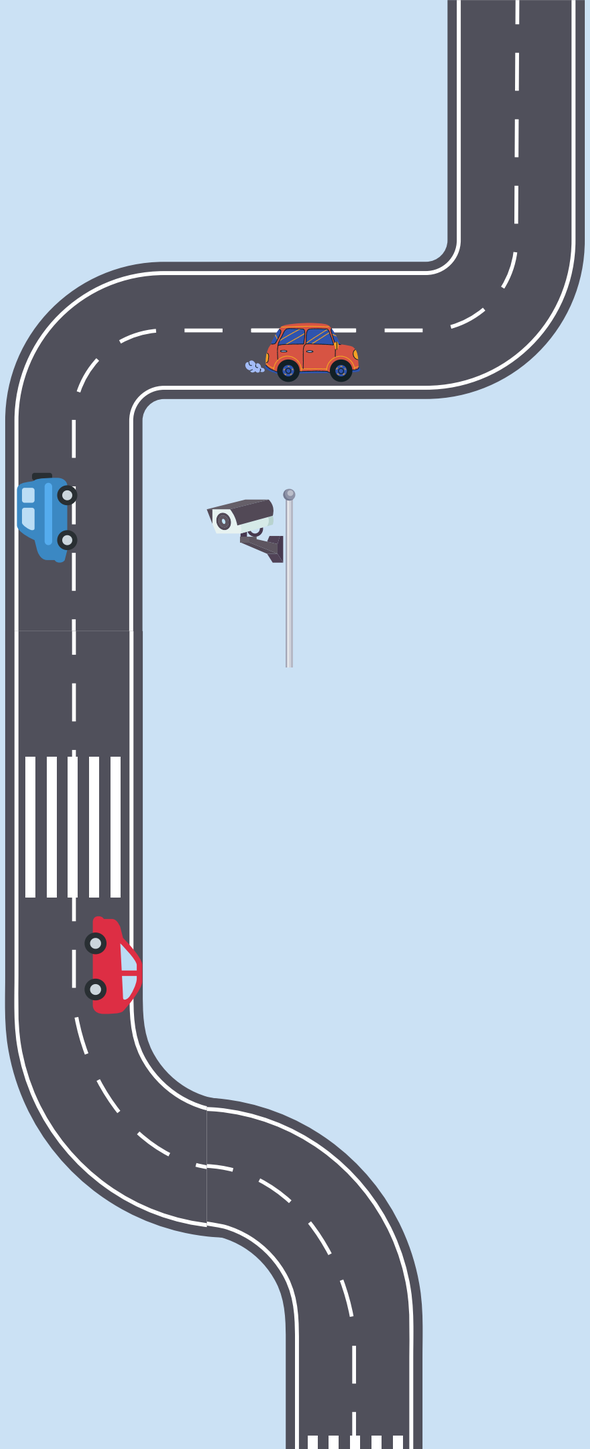


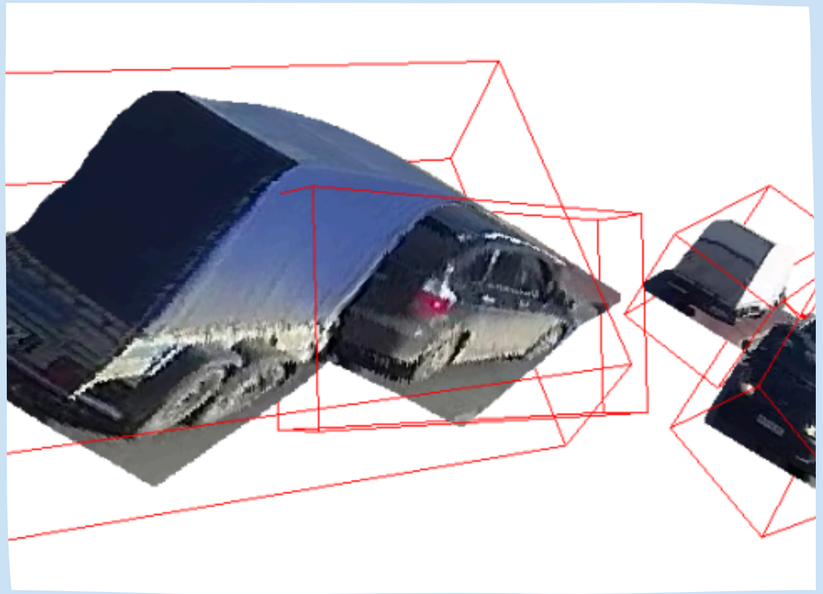


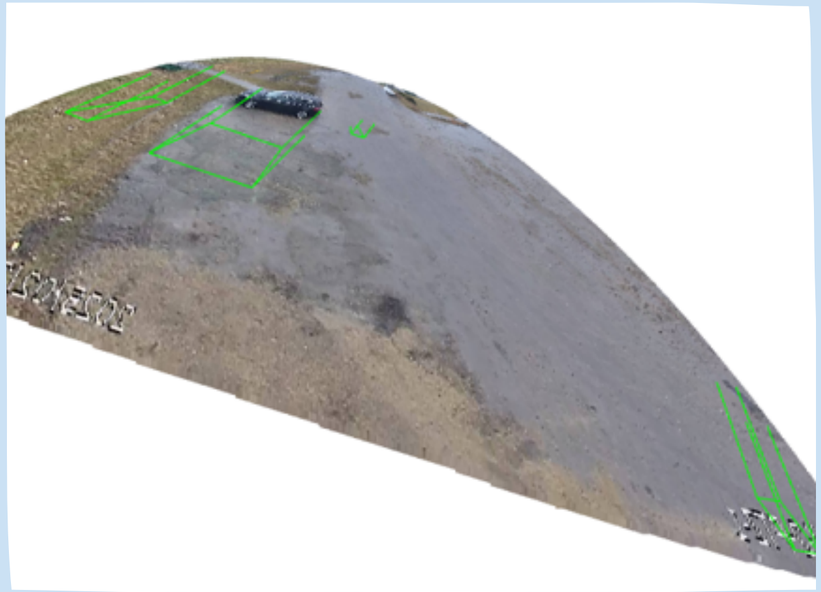


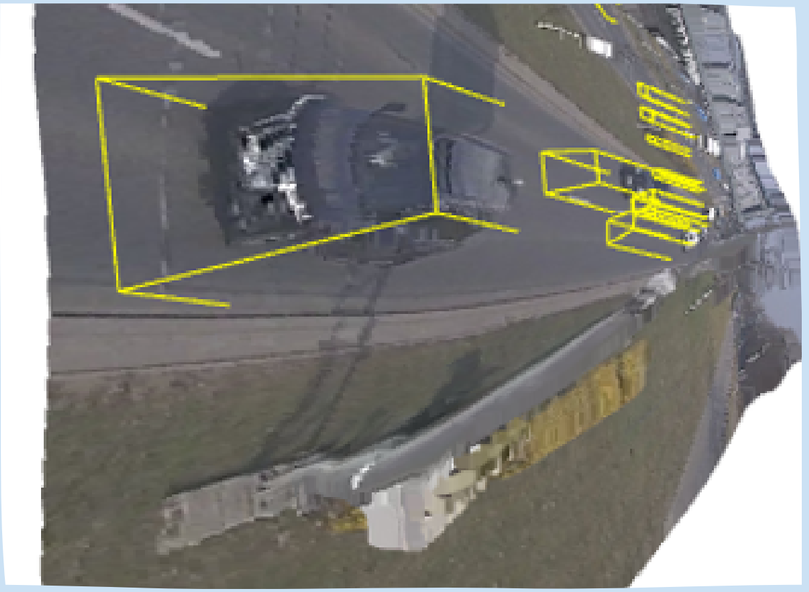












3D Bounding Box

Steps of Construction

PCA (Principal Component Analysis):

Applied to the vehicle's point cloud to find dominant directions and orientation.

Covariance Matrix & Eigen Decomposition:

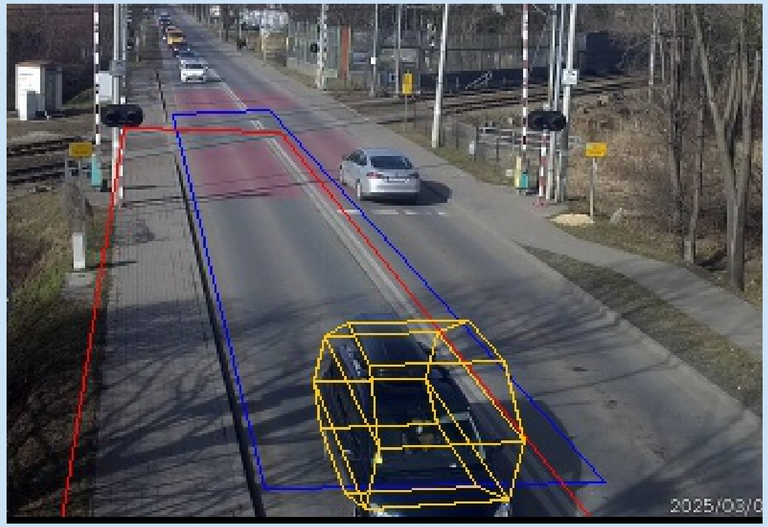
Used to calculate rotation matrix from the shape distribution of the object.

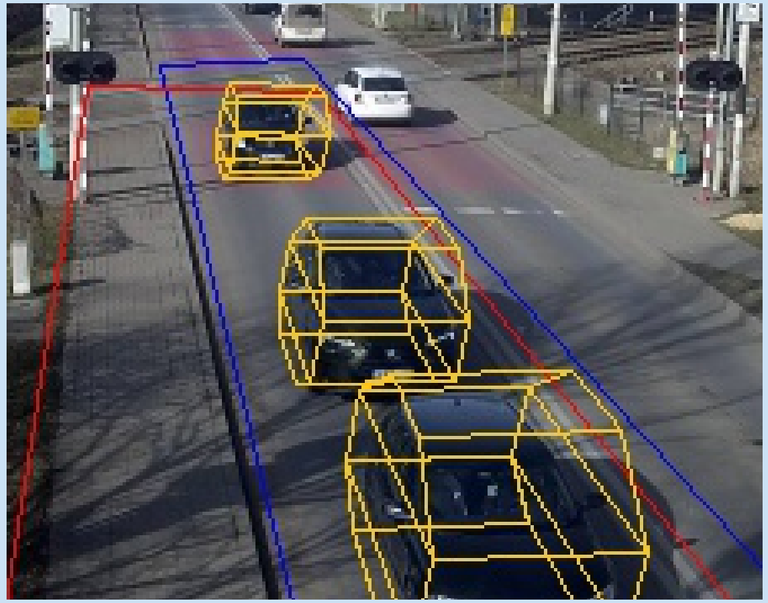
Dimension Estimation:

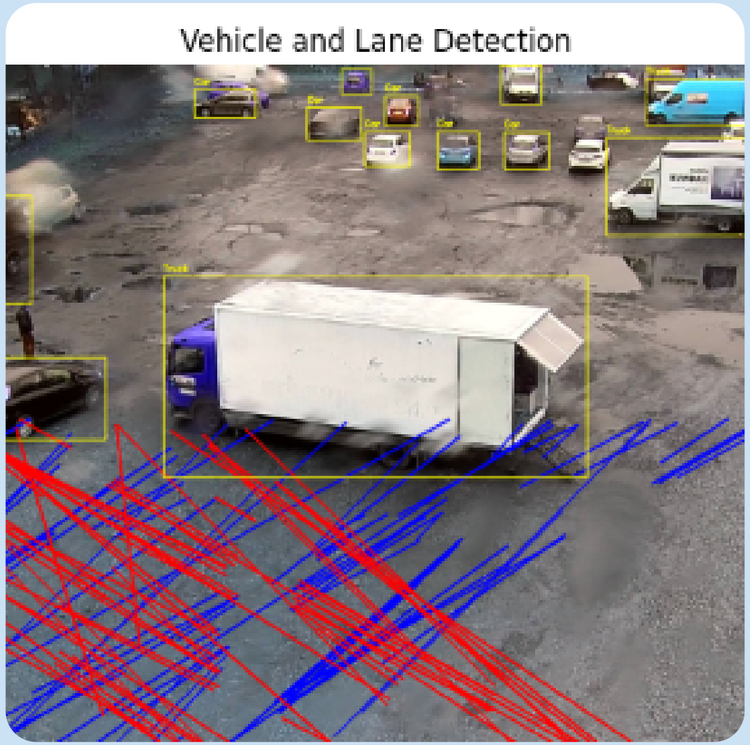
Box size (W × H × L) calculated from spatial spread along principal axes.

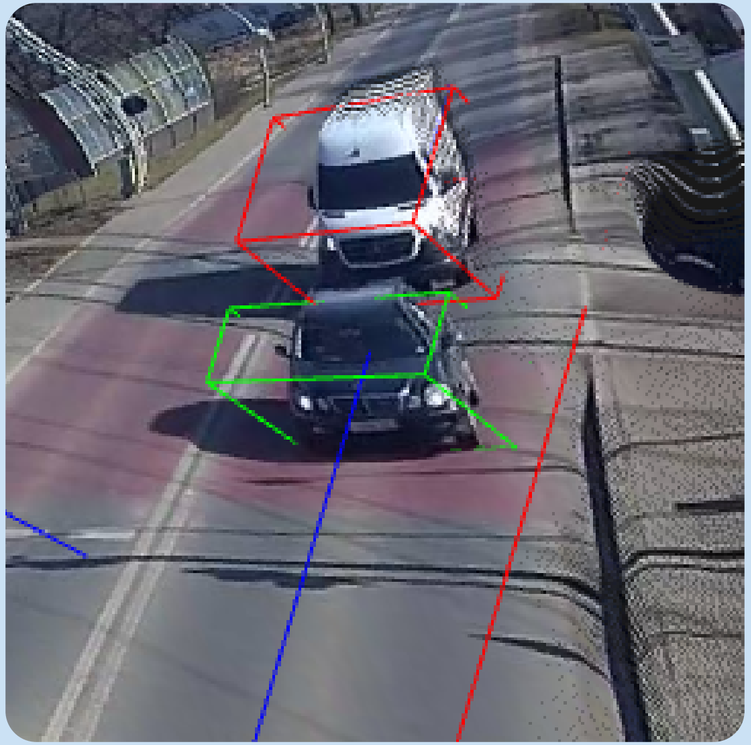
SVD (Singular Value Decomposition):

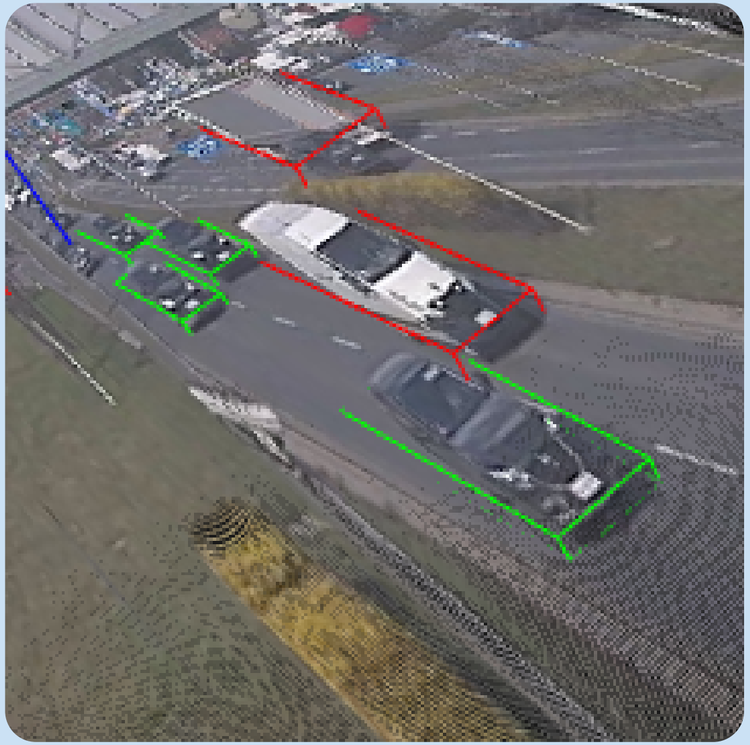
Ensures orthogonality of rotation matrix when converting to world space.






Vehicle and Lane Detection





Coordinate Transformation & Validation

Coordinate Transformation

3D boxes and point clouds are first generated in camera coordinates, then transformed to world space using a 4×4 extrinsic matrix. SVD refines rotation to align with real world axes.

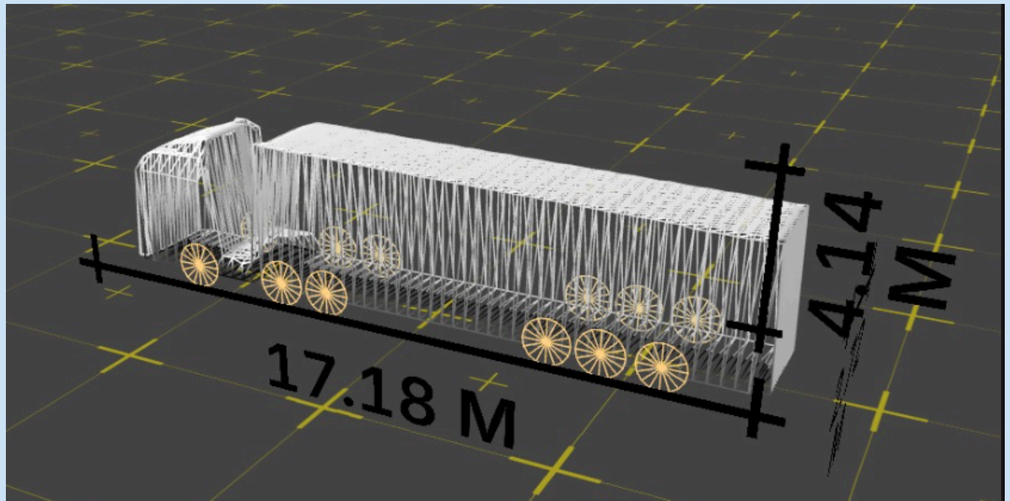
LIDAR Based Validation

Reference sizes come from a LIDAR based motion capture system. Predicted 3D boxes are compared using Euclidean distance for size and center error calculation.

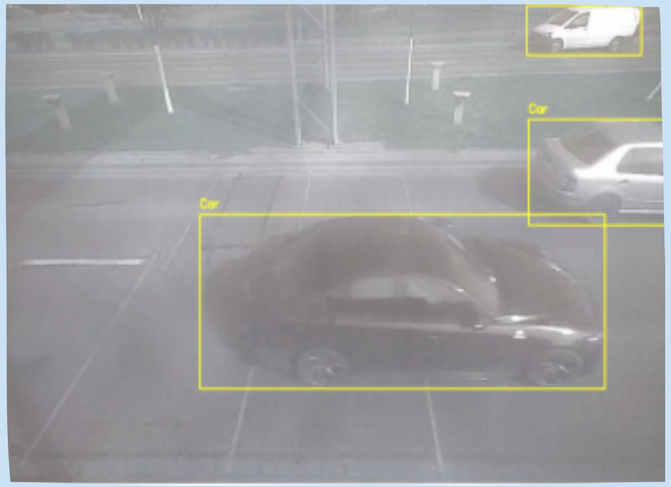
Result:

Very small size errors confirm high physical accuracy of the system's predictions.

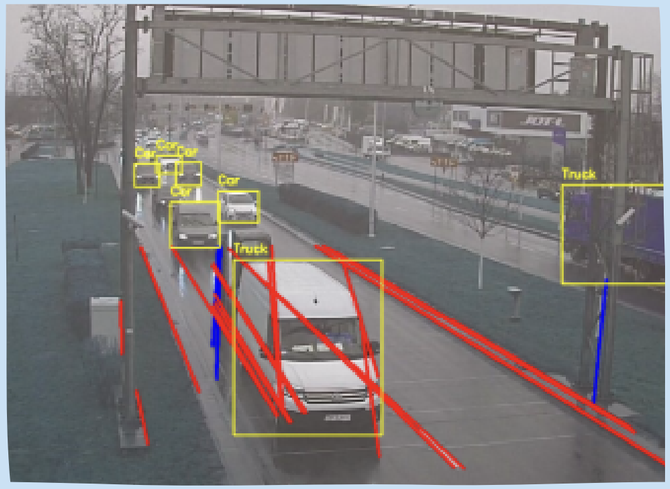

```
Box 0:
  Predicted size: [        6.43          3.49          2.71]
  GLB reference : [        6.42          2.49          2.58]
  Size error    : [        0.01          1.00          0.13]
```



Some Examples of Error Checking




```
Box 0:
  Predicted size: [      6.62        3.73        0.3]
  GLB reference : [      5.65        1.99        1.51]
  Size error    : [      0.97        1.74        1.21]
```



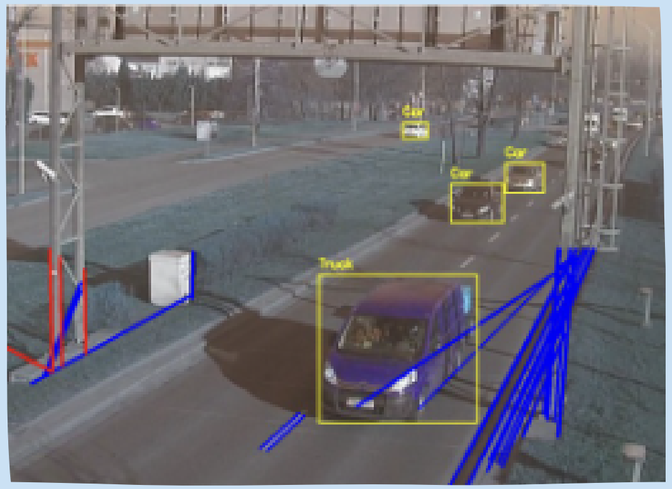

```
Box 0:
  Predicted size: [      8.41        2.78        0.8]
  GLB reference : [     12.21           3        2.2]
  Size error    : [       3.8        0.21        1.4]

Box 1:
  Predicted size: [     11.71        4.34       0.38]
  GLB reference : [     12.21           3        2.2]
  Size error    : [       0.5        1.34       1.81]

Box 2:
  Predicted size: [       3.9        1.35       0.38]
  GLB reference : [     12.21           3        2.2]
  Size error    : [      8.32        1.65       1.82]

Box 3:
  Predicted size: [      3.95        1.53       0.31]
  GLB reference : [     12.21           3        2.2]
  Size error    : [      8.26        1.47       1.89]

Box 4:
  Predicted size: [      3.03        1.45       0.26]
  GLB reference : [     12.21           3        2.2]
  Size error    : [      9.18        1.55       1.94]
```

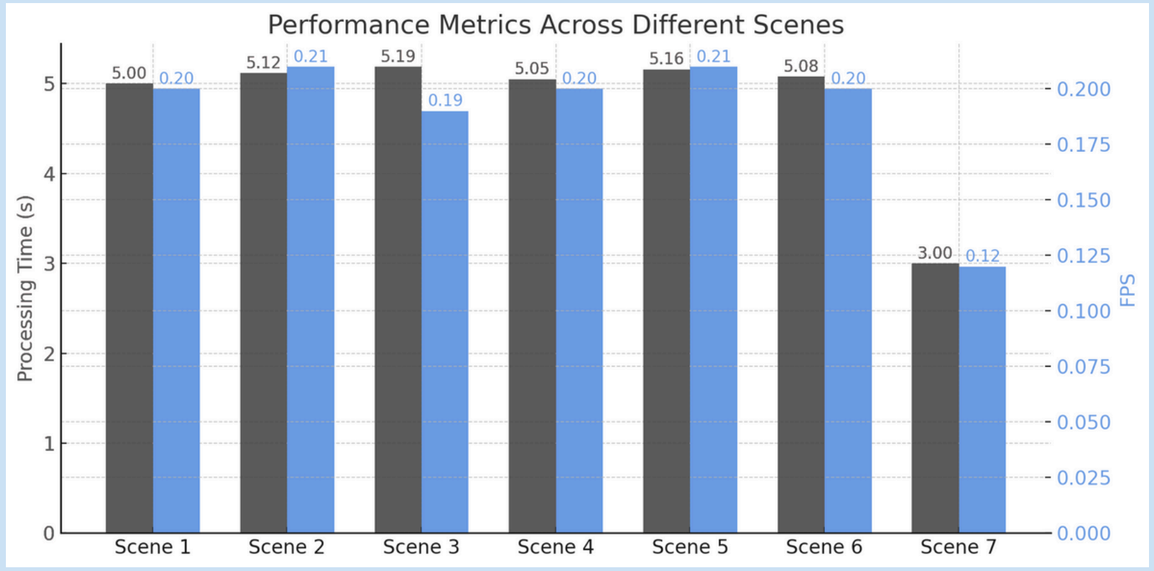


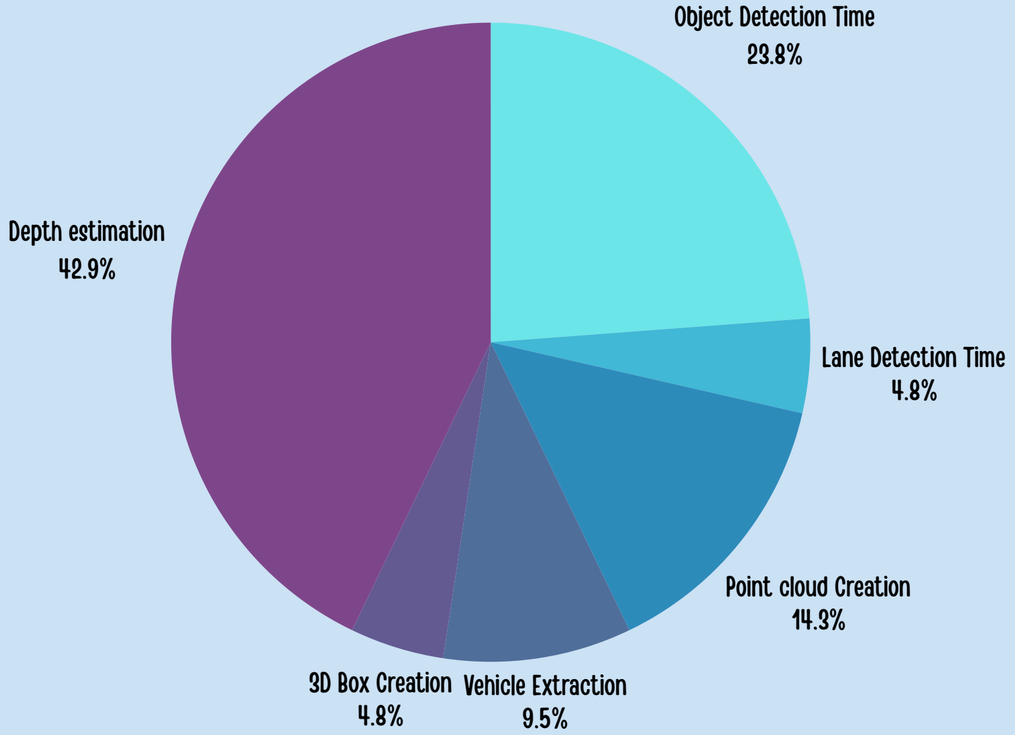

```
Box 0:
  Predicted size: [       9.1        2.29       0.11]
  GLB reference : [      8.42         2.1       2.02]
  Size error    : [      0.68        0.18       1.91]

Box 1:
  Predicted size: [      5.66        1.25        0.1]
  GLB reference : [      8.42         2.1       2.02]
  Size error    : [      2.76        0.85       1.92]

Box 2:
  Predicted size: [         3        1.14       0.07]
  GLB reference : [      8.42         2.1       2.02]
  Size error    : [      5.42        0.96       1.95]
```

Processing Performance (CPU Test)



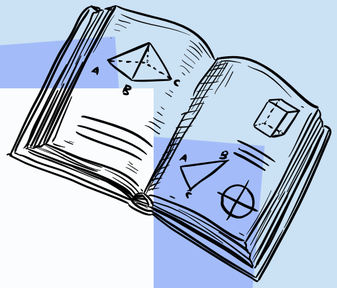


WHAT WE HAD

- CCTV images and input data
- Camera Calibrations
- Approximate direction of approach
- Example results

WHAT WE WANTED

- Real time 3D vehicle body estimation
- Output should include location, orientation, and dimensions
- Workable on low resource devices like Raspberry Pi
- Fully automated from 2D camera input to 3D output



THANK YOU!

We appreciate your attention

Wrocław University of Science and Technology

neurosoft